UNIVERSIDAD RICARDO PALMA

ESCUELA DE POSGRADO

MAESTRÍA EN CIENCIA DE LOS DATOS

Tesis para optar el Grado Académico de Maestro en Ciencia de los Datos

"Modelo de pronóstico de rendimiento académico de alumnos en los cursos del programa de estudios básicos de la Universidad Ricardo Palma usando algoritmos de Machine Learning**"**

Autor: Bach. Vega García, Javier Fernando

Asesor: Mg. Salinas Flores, Jesús Walter

LIMA- PERÚ

2019

PÁGINA DEL JURADO

El Jurado Examinador para la evaluación de la sustentación de la presente tesis, se encuentra integrado por los siguientes miembros:

1. Presidente : Mg. José Antonio Cárdenas Garro
2. Miembro : Mg. Mirko Jerber Rodríguez Mallma
3. Miembro : Mg. Alfredo León Aguilar
4. Asesor : Mg. Jesús Walter Salinas Flores
5. Representante de la EPG : Mg.

DEDICATORIA

A mi esposa,
a mi hijo,
a mi pequeño amor de historieta,
y principalmente a Dios,
simplemente, gracias.

AGRADECIMIENTO

El autor expresa su especial gratitud al Mg. Jesús Salinas Flores, por su apoyo como asesor en la presente investigación, así como su dedicación y paciencia en absolver cada inquietud o duda en el desarrollo de la misma.

Muchas gracias, a todos los profesores de la Maestría en Ciencias de los Datos de la Universidad Ricardo Palma por el conocimiento adquirido y a mis amigos de estudio por el compañerismo y el apoyo desinteresado.

Gracias, a cada una de las personas que de una u otra manera me apoyaron, colaboraron o impulsaron para poder finalizar este proyecto de tesis.

ÍNDICE DE CONTENIDO

LISTADO DE TABLAS

LISTADO DE FIGURAS

RESUMEN Y PALABRAS CLAVE

En la sociedad actual, el acceso a la educación es un derecho que genera la expectativa de que los estudiantes con un alto rendimiento académico tendrán mejores oportunidades laborales que aquellos con un rendimiento académico normal o inferior. La identificación de oportunidades de mejora educativa es crucial para el desarrollo de la sociedad.

El objetivo de esta investigación es efectuar predicciones, mediante el uso de algoritmos de *Machine Learning*, con la finalidad de identificar con anticipación a los estudiantes que tienen alta probabilidad de obtener un bajo rendimiento académico en cualquiera de los 13 cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma en Perú, y como consecuencia, poder implementar estrategias que los ayude a tener mejores resultados en dichos cursos.

Se presenta la implementación, análisis y comparación de tres algoritmos de *Machine Learning*: Redes Neuronales Artificiales (RNA), *Gradient Boosting Machine* (GBM) y *XGBoosting*; con los cuales se pretende determinar el rendimiento académico a través del pronóstico de la cantidad de estudiantes aprobados y desaprobados para cada curso.

Palabras Clave: Rendimiento académico, *Machine Learning*, Red Neuronal Artificial, *Boosting*, Ensamble, Pronósticos.

ABSTRACT AND KEYWORDS

In today's society, the access to the education is a right that generates the expectation that students with high academic performance will have better job opportunities than those with normal or lower academic performance. The identification of opportunities for educational improvement is crucial for the development of our society.

The objective of this research is to make predictions, through the use of Machine Learning algorithms, to identify in advance the students with high probability of obtaining a low academic performance in any of the 13 courses of the Basic Studies Program at the Ricardo Palma University in Peru, and as a consequence, to be able to implement strategies that help them to achieve better results in those courses.

The implementation, analysis and comparison of three Machine Learning algorithms is presented: Artificial Neural Networks (RNAs), Gradient Boosting Machine (GBM) and XGBoosting; with which it are intended to determine the academic performance of students through the forecast of the number of students who will fail or pass each course.

Keywords: Academic performance, Machine Learning, Artificial Neural Networks, Boosting, Ensemble, Forecasting.

INTRODUCCIÓN

El rendimiento académico del estudiante, es un tema que preocupa a cualquier sociedad, se reconoce como un derecho fundamental el tener acceso a la educación, y se asume como consecuencia que los estudiantes con mejor rendimiento académico tendrán mejores oportunidades laborales, entonces, conocer con anticipación a los estudiantes que tienen la probabilidad de obtener un bajo rendimiento académico es fundamental para poder asistirlos en el camino hacia mejores perspectivas de futuro.

La Universidad Ricardo Palma, constituida en la ciudad de Lima en Perú, ofrece 19 carreras profesionales, y todos los alumnos siguen un programa común de cursos denominado Programa de Estudios Básicos (PEB) que consta de 13 cursos, los cuales se dictan durante los primeros 3 ciclos de cada carrera profesional.

Si bien es cierto que los factores que conforman el rendimiento académico son de diversa índole (social, personal, económico, psicológicos, etc.), también es cierto que tal acopio de datos no existe en la Universidad Ricardo Palma. Sin embargo, con las notas de los alumnos desde el ciclo 2015-1 hasta el ciclo 2019-0, y con todos aquellos datos que pudiesen estar registrados en el Centro de Computo de la Universidad Ricardo Palma se preparó un *dataset* para cada curso del PEB.

La metodología CRISP-DM fue la que se utilizó para el proceso de minería de datos y los algoritmos de *Machine Learning* para el modelado, con la finalidad de realizar una predicción con alta precisión de exactitud que anticipe el desempeño académico de un estudiante, identificar aquellos con bajo rendimiento y seleccionarlos para que reciban tutoría académica y de esta manera puedan mejorar su desempeño.

La presente investigación está compuesta por 5 capítulos:

- Capítulo I, donde se explica el problema y los objetivos que se pretenden conseguir.
- Capítulo II, comprende el marco histórico, el marco teórico, variables, investigaciones relacionadas y las hipótesis.

- Capítulo III, describe la metodología aplicada en la presente investigación, como son el tipo, el método, el diseño, las técnicas y los instrumentos.

- Capítulo IV, comprende la ejecución de la metodología CRISP-DM para realizar el pre-procesamiento y limpieza de datos, el tratamiento y desenvolvimiento de los modelos, los resultados obtenidos y el análisis correspondiente.

- Capítulo V, donde se expresan las conclusiones y recomendaciones.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción del problema

El análisis de los datos es un proceso mediante el cual usamos algoritmos para descubrir e informar sobre patrones influyentes en los datos. El objetivo es obtener conocimiento (información) que, a menudo, afectan las decisiones. Los datos son una medida de información histórica, por lo que, por definición, los análisis se encargan de observar y estudiar los datos históricos. (Abbott, 2014, pág. 3)

Hay innumerables decisiones que las empresas y los entes estatales toman todos los días y se pueden llevar acciones de mejora mediante la utilización de los patrones encontrados como ayuda para obtener tendencias y llevar a tomar decisiones para obtener un beneficio. (Abbott, 2014, pág. 2)

De acuerdo a lo mostrado por el Diccionario de la Real Academia Española (RAE), el concepto del vocablo “reprobar” proviene de la lengua latín, delimitándose al hecho de no tener las suficientes habilidades, competencias o conocimientos sobre un tema en particular. Se asocia principalmente al ámbito educativo, aunque también puede usarse para calificar las cualidades de una persona. En ambos casos, tanto en el ámbito académico como social, conlleva una connotación negativa pues es indicador de que, la persona no ha obtenido la calificación de aprobado en una asignatura (examen, curso o materia) u opinión favorable por parte de una tercera persona respectivamente.

Los componentes para estudiar la desaprobación universitaria de los estudiantes son muy variados y de campos muy diversos, debido a ello existen un sinfín de investigaciones sobre el tema sin que ninguna de ellas logre juntar y encontrar las causas de un insuficiente rendimiento académico, pero, irónicamente es un tema por el cual se mide el nivel educacional de todas las instituciones educativas, independiente del nivel educacional a donde se dirigen. (García Ortiz, López de Castro Machado, & Rivero Frutos, 2014)

Los estudios generales son parte de la formación del estudiante universitario, donde se busca la integración y el desarrollo de sus habilidades, pero en un marco de reflexión, de pensamiento crítico y de vínculo con otras ciencias, comprendiendo que todo se encuentra relacionado como parte de un contexto mayor.

El Programa de Estudios Básicos de la Universidad Ricardo Palma, consta de 3 ciclos académicos, durante los cuales se brinda al alumnado un conjunto de cursos para que puedan desarrollar un razonamiento crítico y completo, integrando las competencias profesionales con las competencias éticas y actitudinales.

1.2. Formulación del problema

La presente investigación, con la ayuda de los algoritmos de *Machine Learning*, tiene como objetivo predecir la cantidad de alumnos que aprobarán y desaprobarán los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma, por lo mencionado, la importancia de la presente investigación se orienta a sentar las bases para el desarrollo de una herramienta que, en un corto plazo, permita conocer con anticipación el número de grupos que deberá tener un curso, del Programa de Estudios Básicos, para el ciclo siguiente de una manera más eficiente y precisa que la actual, contribuyendo en el extremo de mejorar la precisión de la programación académica en lo concerniente a aulas, docentes y horarios, además de disminuir la utilización de horas hombre en dicha programación.

La información histórica nos indica que, durante el semestre 2015-1, había una cantidad de 6,336 registros que correspondían a alumnos matriculados en los cursos del Programa de Estudios Básicos (PEB).

Al realizar un análisis más exhaustivo pudimos observar que, dicha cantidad de registros pertenece a 1,360 alumnos. También se obtuvo que 4,695 registros tenían condición de aprobados los cuales pertenecían a 1,259 alumnos y 1,641 registros tenían situación de desaprobados correspondiendo a 789 alumnos, por lo tanto, existen alumnos que aprueban y desaprueban cursos del PEB simultáneamente en el mismo periodo académico.

Entonces, se puede indicar que, el porcentaje total de aprobados y desaprobados en los 13 cursos del Programa de Estudios Básicos, fue del 74% y 26% respectivamente.

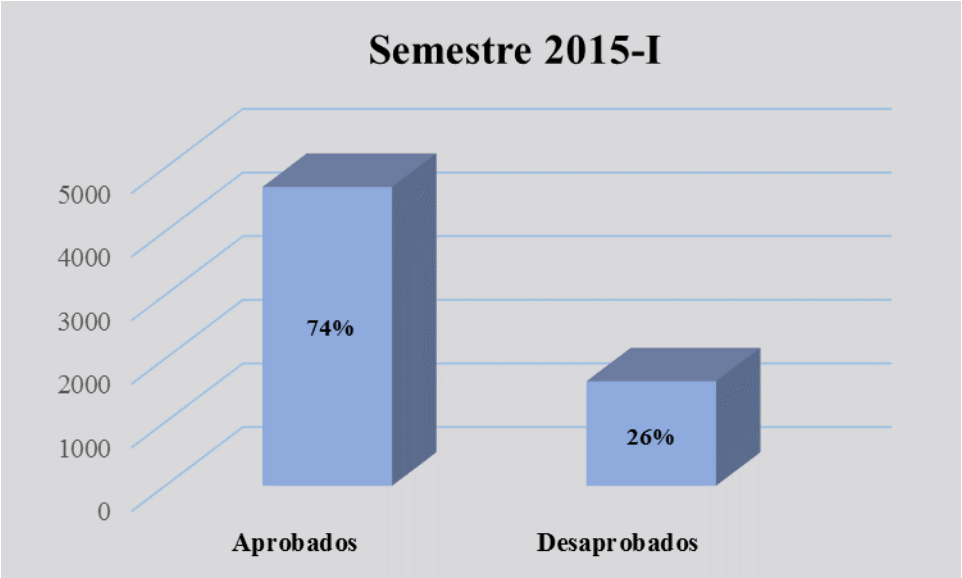


Figura 01: Distribución de los Aprobados y Desaprobados en los cursos del Programa de Estudios Básicos durante el Semestre 2015-I.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

El objetivo es hallar conocimiento ignorado, original y nuevo, que permita a la Universidad Ricardo Palma tener una valoración que brinde diferencia con respecto a su competencia. Para ello se necesitan las técnicas que permitan obtener nuevo conocimiento y a partir de allí, hallar la información más importante para poder adelantarse a las expectativas de los alumnos.

De acuerdo a lo expuesto, la pretensión general de la presente investigación es, utilizar los algoritmos de *Machine Learning* con la finalidad de predecir la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.

1.2.1. Problema general.

En atención a lo anterior se formuló la siguiente interrogante general: ¿Cómo hacer uso de los algoritmos de *Machine Learning* para predecir la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma?

1.2.2. Problemas específicos.

Se formularon las siguientes interrogantes específicas:

A. ¿Cómo utilizar el algoritmo de Redes Neuronales Artificiales (RNA) para predecir la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma?

B. ¿Cómo utilizar el algoritmo *Boosting* para predecir la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma?

C. ¿Cuál de los dos algoritmos brindará mejores pronósticos?

1.3. Importancia y justificación del estudio

La importancia y la justificación de la investigación se sustentan en lo siguiente:

A. Se busca optimizar el modelo de predicción actual (el cual no existe).

B. Se busca disminuir los niveles de desaprobación del alumnado en los cursos del Programa de Estudios Básicos (PEB).

C. Se busca establecer una característica adicional en la evaluación docente.

   La primera opción sería el reflejo del desempeño de todos los alumnos que el docente tenga a su cargo en un determinado curso, y estaría determinada por el promedio de la nota final de los alumnos versus el promedio de la proyección de la nota final en base a la primera evaluación.

   La segunda opción sería enviar la encuesta de evaluación solo a aquellos alumnos con posibilidades de aprobar el curso, los alumnos serían seleccionados en base a la primera evaluación del respectivo curso.

D. Se busca crear una Unidad de Apoyo al Alumno, compuesto por psicólogos y docentes, que organice las acciones y los medios para evitar que el alumno sea desaprobado en un determinado curso del PEB.

E. Se busca una toma de decisiones más asertiva con respecto al uso del personal docente en tareas de tutoría, asesoramiento y apoyo al alumnado.

F. Se busca evitar que haya deserción del alumnado debido a un bajo rendimiento académico en los cursos del PEB, porque la Universidad Ricardo Palma dejaría de

percibir ingresos de un determinado alumno, en promedio, por los siguientes cuatro años.

G. La Universidad tiene el deber de formar al alumno de una manera integral, para ello la relación universidad-alumno debe ser estrecha, el alumno debe sentir que la Universidad se preocupa por él. Una manera de hacerlo es que al detectar la posibilidad de que el alumno pueda tener un bajo rendimiento en un determinado curso, la universidad le ofrezca su apoyo mediante talleres de tutoría y de prácticas dirigidas en forma gratuita.

H. Se propone que alumnos a partir de sexto ciclo de cada carrera, previa evaluación y selección, serán los encargados de los talleres de prácticas dirigidas y como contraprestación recibirán un descuento en sus pensiones de pago de hasta 2 armadas.

Se eligieron los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma, porque dichos cursos son llevados por la totalidad de los alumnos que ingresan, a diferencia de los cursos de cada carrera que son llevados exclusivamente por los alumnos de cada carrera, teniendo como consecuencia una mayor cantidad de registros en el *dataset*.

A continuación, acciones adicionales y complementarias que deberían implementarse al concluir la presente investigación:

A. Establecer un mecanismo para determinar el número de grupos que deberá tener un determinado curso en el ciclo siguiente.

B. Implementar una programación de horarios y docentes en cada curso y sus respectivos grupos, que busque maximizar el uso de las aulas y minimizar el cruce de horario de los alumnos.

Como consecuencia se obtendrá un ahorro de horas-hombre y recursos en la implementación de los 2 ítems anteriores y los cambios descritos tendrán como resultado la percepción, por parte del alumnado, de una mejora en el servicio.

1.4. Delimitación del estudio

La Universidad Ricardo Palma proporcionará las bases de datos que contienen el historial del rendimiento académico del alumnado (las notas de todos los cursos, incluyendo los 13 del Programa de Estudios Básicos, se encuentran expresadas en la escala del 0 al 20), así como la correspondiente información sociodemográfica. Sin embargo, dicha información no se encuentra unificada ni pertenece a una única fuente. Por lo tanto, se tendrá que implementar un cronograma suplementario, donde se deberá considerar un tiempo adicional y a la vez necesario, para poder transformar y unificar los datos en un único *dataset*.

Existe una dificultad en el acceso a fuentes de información que hayan investigado y desarrollado una investigación de características similares en el Perú, por lo tanto se tomará como base anteriores trabajos de investigación del ámbito regional.

1.5. Objetivos de la investigación

1.5.1. Objetivo general.

El objetivo general es el siguiente:

- Pronosticar la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma mediante el uso de algoritmos de *Machine Learning*.

1.5.2. Objetivos específicos.

Los objetivos específicos que podríamos citar son los siguientes:

- Determinar la efectividad del uso del algoritmo de Redes Neuronales Artificiales (RNA) para pronosticar la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.

- Determinar la efectividad del uso del algoritmo *Boosting* para pronosticar la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.

- Especificar y evaluar la Tasa de Acierto de ambos algoritmos.

CAPÍTULO II: MARCO TEÓRICO

2.1. Marco Histórico

2.1.1. Rendimiento Académico.

Desde el primer momento en que se empezó a impartir educación, aparecen dos consecuencias naturales: la primera, que el estudiante termine sus estudios; la segunda, que el educando los abandone.

De la primera, los beneficios que se obtienen se trasladan no solo a la persona, sino a la familia, al centro de estudios, a la institución en la que trabaja y por último al país. La segunda origina todo lo contrario, por ello es importante para cualquier centro de estudios preguntarse: ¿Cuál es el motivo de un estudiante para abandonar sus estudios?, o en otras palabras ¿Cuáles son las causas de la deserción en el centro de estudios?. (SINEACE, 2013, pág. 14)

Al realizar la investigación, se encontró una infinidad de trabajos relacionados con el tema que intentan encontrar una explicación sobre los motivos que originan la deserción, los cuales podrían clasificarse en: psicológicos, económicos, sociológicos, organizacionales y de interacciones. (Himmel, 2002, pág. 96)

En relación con el tipo de deserción, se puede dividir en: voluntaria e involuntaria. La primera está relacionada con una decisión personal del alumno, mientras que la segunda está vinculada al centro de estudios y sus normas institucionales (académicas y/o administrativas). (Himmel, 2002, pág. 95)

En las universidades privadas la deserción es un problema muy grave, debido a que ocasiona problemas financieros a futuro, a la vez, puede percibirse como una mala gestión administrativa por parte de la universidad y generar una impresión negativa de la imagen de la universidad.

Por el contrario, una alta tasa de graduados en relación al número de ingresantes, es un indicador de éxito, aceptado global e históricamente y debería ser considerado como prioridad en la gestión universitaria.

2.1.2. Machine Learning.

Aristóteles (384-322 a.C.), podría ser considerado la primera persona que inicio el camino hacia la Inteligencia Artificial (IA), cuando se propuso a estudiar, codificar y explicar ciertas formas de razonamiento deductivo que decidió denominar silogismos. (Ponce Cruz, 2010, pág. 1)

En su libro "*Leviatán*" de 1651, Thomas Hobbes sugirió la idea de un "animal artificial", argumentando "Qué es en realidad el corazón sino un resorte; y los nervios qué son, sino diversas fibras; y las articulaciones sino varias ruedas que dan movimiento al cuerpo entero". (Russell & Norvig, 2010, pág. 6)

En 1847 George Boole (1815–1864), desarrollo matemáticamente los detalles de la lógica proposicional o booleana, basado en la idea de la lógica formal de los filósofos de la antigua Grecia. En 1879, Gottlob Frege (1848–1925) creo la lógica de primer orden, mediante la inclusión de objetos y relaciones en la lógica de Boole; y Alfred Tarski (1902–1983) desarrollo la teoría de referencia donde muestra la forma de relacionar los objetos en una Lógica a las entidades en el universo finito. (Russell & Norvig, 2010, págs. 7-8)

Alan Turing (1950), presenta la Prueba de Turing, en la cual se pretende ofrecer una especificación de la inteligencia de forma satisfactoria y operacional. Turing delimitó un comportamiento inteligente de modo que, en cualquier acción cognitiva se pueda obtener una eficacia en la dimensión humana. Turing formuló la prueba donde, una persona interrogase a un terminal por medio de un teletipo; si la persona no tenía la capacidad de poder identificar fehacientemente si, la computadora o la persona respondía el cuestionario por medio del teletipo, entonces, el desempeño de la computadora debería considerarse aprobado. (Russell & Norvig, 2010, pág. 2)

John McCarthy (1956), con el apoyo de Marvin Minsky, Claude Shannon y Nathaniel Rochester, logra promover para el verano de 1956, la conferencia de Darthmouth (inicialmente un taller con una duración de 2 meses), la cual es calificada como el acontecimiento donde surge el ámbito de la Inteligencia Artificial (IA), debido a que, Minsky hace uso por primera vez del término "*Artificial Intelligence*" durante la duración de la conferencia. (Russell & Norvig, 2010, pág. 17)

En 1958 John McCarthy postulo un sistema llamado "*Advice Taker*", donde las matemáticas proposicionales se utilizan a manera de un lenguaje para simbolizar y usar la inteligencia. El inconveniente era que, en lugar de ser programado se le tenía que indicar qué debía realizar. (Ponce Cruz, 2010, pág. 2)

Arthur Samuel de IBM, en el año 1959, especifica al aprendizaje automático como: "Campo de estudio que le da a las computadoras la capacidad de aprender sin ser programado explícitamente". Samuel es coautor de un algoritmo de autoaprendizaje basado en el juego de damas, porque es un juego simple donde se necesita establecer una estrategia para poder ganar, dicha estrategia podría ser aprendida por su algoritmo. (Bell, 2015, pág. 2)

Durante los años sesenta e inicios de los setenta la IA pudo crecer en diversos programas, uno de los cuales era el "*General Problem Solver*" (GPS) desarrollado por Newell, Show y Simon. (Ponce Cruz, 2010, pág. 2)

Alrededor de 1980, se pudo desarrollar programas con una mayor capacidad y con el conocimiento esencial para emular la capacidad de expertos en diversas labores. (Ponce Cruz, 2010, pág. 2)

DeJong, en 1981, propone la noción del "*Explanation Based Learning*" (EBL), mediante la cual, un terminal indaga en una fuente de información (el *dataset* para el entrenamiento) a partir de la cual, formula patrones que utiliza para eliminar la información menos relevante. (Russell & Norvig, 2010, pág. 799)

El once de mayo del año 1997, *Deep Blue* un programa desarrollado por IBM fue capaz de vencer a Garry Kasparov (campeón mundial vigente de ajedrez de ese año). (Ponce Cruz, 2010, pág. 2)

Ese mismo año, Tom M. Mitchell, Director de Aprendizaje Automático en la Universidad Carnegie Mellon, detalla al aprendizaje automático como:

> Un programa de computadora aprende de la experiencia (E) con respecto a una clase de tareas (T) y una medida de rendimiento (P), si su rendimiento en las tareas en T, medidas por P, mejora con la experiencia E. (Bell, 2015, pág. 2)

Dicho enunciado, quizás uno de las más citados, tiene como novedad la inclusión de 3 elementos que ayudan en la definición: Tarea (T, puede ser una o varias), Experiencia (E) y Rendimiento (P, por la sigla en inglés de *Performance*). Entonces, a mayor experiencia **E** en la ejecución de la tarea **T** tiene como resultado un aumento del rendimiento **P**. (Bell, 2015, pág. 2)

La ciencia de la computación empezó su historia hace más de 60 años y durante todo ese tiempo los recursos se han centrado en los algoritmos, actualmente algunos ensayos sobre Inteligencia Artificial (IA), han encontrado que los recursos deben focalizarse en los datos en vez de preocuparse del algoritmo a utilizar, debido al incremento de información (texto, imagen, voz, audio, etc.) en bases de datos a nivel mundial. En el mismo sentido crítico, personajes importantes del estudio de IA han manifestado su desacuerdo sobre el camino que ha seguido esta, por el empeño de seguir mejorando aplicaciones que se emplean para una tarea específica, en vez de ayudar a crear nuevas aplicaciones. (Russell & Norvig, 2010, pág. 27)

En los próximos años, aumentar y/o mejorar las destrezas y/o habilidades de las maquinas autónomas así como de las funciones del *software*, serán el impulso que guiaran las investigaciones en Inteligencia Artificial. (Ponce Cruz, 2010, pág. 2)

2.1.3. Minería de Datos.

La definición fue establecida en los años 80 por los desarrolladores de bases de datos y se indica como el proceso de descubrimiento de importantes patrones desconocidos, mediante la exploración de grandes volúmenes de información. (Rodriguez Pacheco, 2015, pág. 46)

Los profesionales de diferentes disciplinas (economistas, estadísticos e ingenieros) siempre han tenido presente la idea de que existen patrones en los datos y que al usarlos se podría realizar predicciones automáticas sobre actividades y hábitos cotidianos. Aproximadamente cada 20 meses a nivel mundial, la información recolectada en diferentes medios de almacenamiento se está duplicando lo que origina un nicho de oportunidades para explorar, encontrar y confirmar patrones. Dicho crecimiento ha tenido como consecuencia que, la minería de datos ha aumentado su importancia a nivel empresarial porque permite tomar decisiones de una manera más asertiva. (Witten, Frank, Hall, & Pal, 2017, pág. 5)

Actualmente asistimos a un boom de datos, pero no se ha aprovechado de forma útil los beneficios que ellos nos brindan, pero las organizaciones más grandes han utilizado dicho potencial para su beneficio, y lo han hecho utilizando algoritmos de *Machine Learning* (aprendizaje automático) para encontrar en los datos información útil para sus propios fines. (Witten, Frank, Hall, & Pal, 2017, págs. 4-5)

2.1.3. Métodos de Ensamble.

En un principio los algoritmos de *Boosting* se aplicaron para problemas de clasificación y posteriormente abarcaron los problemas de regresión. Todo dio comienzo con el algoritmo *AdaBoost* y evoluciona hasta la "*stochastic gradient boosting machine*" de Friedman, la cual comprende problemas de clasificación y regresión. (Kuhn & Johnson, 2013, págs. 203-204)

En 1990, dos trabajos sobre el algoritmo *Boosting* fueron publicados por separado. Uno fue realizado por Robert Shapire, el otro, por Yoav Freund. En 1995, ambos unieron esfuerzos y publicaron el algoritmo *Add a Boost* o llamado también *Addabost*. (Abbott, 2014, pág. 316)

Tras el éxito del algoritmo *AdaBoost* algunos intelectuales establecieron conexión entre el algoritmo con la función de pérdida, con la regresión logística y con el modelado aditivo y expusieron que el algoritmo *Boosting* puede representarse como un esquema que disminuye la pérdida exponencial. Este concepto se aplicó y logró mejorar las perspectivas de los problemas de clasificación, y a su vez se extendió a los problemas de regresión. (Kuhn & Johnson, 2013, pág. 204)

Friedman logró obtener un algoritmo refinado, escueto y muy flexible para problemas de índoles muy disimiles, al cual llamo "*gradient boosting machine*". Está compuesto por los siguientes elementos primordiales: una función de pérdida (en un problema de regresión sería el error cuadrado) y una técnica de modelado débil (*weak learner*) (para el mismo problema utilizaríamos el árbol de regresión); el algoritmo de Friedman trata de hallar un modelo adicional, el cual, cuando se junte con el modelo débil logre disminuir la función de pérdida; el proceso se itera tantas veces como haya indicado la persona. (Kuhn & Johnson, 2013, pág. 204)

Los algoritmos *Bagging* y *Boosting* dieron inicio a los métodos de ensamble, dichos algoritmos surgieron al mismo tiempo que el algoritmo de árbol de decisión. A partir de su aparición, otros punto de vista han visto la luz para los métodos de ensamble, donde los algoritmos con mayor desarrollo y éxito son: *random forests*, *stochastic gradient boosting* y conjuntos heterogéneos. (Abbott, 2014, pág. 320)

2.1.4. Redes Neuronales Artificiales.

Alrededor del 335 a.C., Aristóteles anoto que: "De todos los animales, el hombre tiene el cerebro más grande en proporción a su tamaño". Sin embargo, a mediados del siglo XVIII recién el cerebro fue reconocido como la base de la conciencia. Antes, los órganos candidatos eran el corazón y el bazo. (Russell & Norvig, 2010, pág. 10)

En 1861 Paul Broca (1824–1880) demostró que existen áreas del cerebro específicamente responsables de funciones cognitivas. Por ese entonces, era de conocimiento general que el cerebro se encontraba constituido por neuronas (también denominadas células nerviosas); Camillo Golgi (1843–1926) en el año de 1873, explico un modo mediante el cual se podía observar a las células nerviosas de forma individual. Santiago Ramón y Cajal (1852–1934) la volvió a utilizar para realizar estudios acerca del modo en que las neuronas se organizan en el cerebro. Entre el año 1936 y el año 1938, Nicolas Rashevsky se valió de modelos matemáticos para el estudio del sistema nervioso. (Russell & Norvig, 2010, pág. 10)

El cerebro humano y su funcionamiento fue intentado explicar, alrededor de 1943, por Walter Pitts y Warren McCulloch, mediante una simple red de neuronas conectadas entre sí, para poder examinar operaciones lógicas (y, o, no, etc.) y en donde, cada una tiene el estado de "desactivada" o "activada", con un cambio a "activada" que sucede basándose en la estimulación de las neuronas que se encuentran a su alrededor. (Russell & Norvig, 2010, pág. 16)

Según Donald Hebb (1949) y su concepto denominado "*The Organization of Behavior*" (regla de actualización simple), la fortaleza de la conexión entre neuronas puede modificarse. Su gobierno, ahora llamado aprendizaje Hebbiano, sigue siendo un modelo influyente hasta hoy. (Russell & Norvig, 2010, pág. 16)

En 1950, en Harvard, Marvin Minsky conjuntamente con Dean Edmonds ensamblaron la "*Stochastic Neural Analog Reinforcement Calculator*" (SNARC), que es una red de 40 neuronas artificiales, conformada por 3000 tubos de vacío. (Russell & Norvig, 2010, pág. 16)

Una simple red neuronal artificial, denominada perceptrón, fue construida en 1959, por el psicólogo Frank Rosenblatt (Universidad de Corell) y constaba de 400 fotoceldas que se enlazaban al azar con 512 unidades tipo neurona. El propósito general era instruir acerca de las propiedades básicas en sistemas inteligentes, sin ahondar en muchos detalles relacionados con determinadas circunstancias o con omisiones en relación a determinados seres vivos. (Ponce Cruz, 2010, págs. 6-8)

El trabajo de Winograd y Cowan (1963) visualizó la manera cómo un enorme conjunto de elementos podrían simbolizar colectivamente una concepción en particular, con un incremento correspondiente a la robustez y al paralelismo. Bernie Widrow (entre 1960 y 1962), logró mejorar los algoritmos de aprendizaje de Hebb y llamó a sus redes *Adaline* (*Adaptive Linear Element*), también Frank Rosenblatt (1962) logró mejorarlas mediante los perceptrones. El teorema de convergencia del perceptrón (Block et al., 1962) dice que el algoritmo de aprendizaje puede adecuar las fortalezas de conexión de un determinado perceptrón, de forma que coincida con cualquier dato de entrada, siempre y cuando la coincidencia exista. (Russell & Norvig, 2010, pág. 10)

El algoritmo de *back-propagation* (propagación hacia atrás) para redes de múltiples capas, fue la responsable del resurgimiento de las redes neuronales artificiales a mediados de los 70 (Werbos en 1974), y con mayor ímpetu en los años 80 (Parker en 1985), la cual fue descubierta por primera vez por Bryson y Ho en 1969. (Russell & Norvig, 2010, pág. 22)

La publicación "*Parallel Distributed Processing*" de Rumelhart y McClelland del año 1986 difundió resultados de trabajos que aplicaban el *back-propagation* (retro propagación o propagación hacia atrás), lo que permitió el redescubrimiento de dicho algoritmo. (Russell & Norvig, 2010, pág. 24)

En los años ochenta, Tuevo Kohonen (Universidad de Helsinki) propuso un algoritmo compuesto por dos variantes, cada una construye una representación gráfica basándose en las cualidades semejantes de los datos de ingreso proporcionados, la diferencia radica

en la dimensionalidad del gráfico. En estas redes el aprendizaje es de tipo *offline*, o es lo mismo establecer que, hay la primera etapa denominada de aprendizaje, en la cual se establecen los pesos de las conexiones y la segunda etapa denominada de funcionamiento. (Ponce Cruz, 2010, pág. 240)

J. Hopfield fundamentó, durante los ochenta, la red recurrente, que está conformada por un conjunto de neuronas (n) en donde la salida de cada una de ellas (cada neurona) es aprovechado como retroalimentación en las entradas, excepto para su misma entrada, lo cual le otorga estabilidad. La idea de Hopfield es similar a la propuesta por Jordan así como la propuesta por Elman. (Ponce Cruz, 2010, pág. 242)

En 1998 LeCun, Bottou, Bengio y Haffner desarrollaron LeNet, que es una red neural convolucional (CNN, por las siglas en inglés de *Convolutional Neural Networks*), la cual tiene su bases en el "neocognitron" formulado en 1980 por Fukushima. (Witten, Frank, Hall, & Pal, 2017, pág. 461)

En los años siguientes aparecen las redes LSTM (*Long Short Term Memory*), las máquinas de Boltzmann, las redes autoencoders y recientemente las "*Generative Adversarial Networks*" (Redes Antagónicas Generativas o GAN).

2.2. Investigaciones relacionadas con el tema

Se ha logrado encontrar y revisar diversos Trabajos de Investigación cuyo tema es similar al que se pretende desarrollar, los más relevantes son los siguientes:

- Fischer, E. (2012), alumno de la Universidad de Chile (Santiago de Chile, Chile), en su tesis titulada: "*Modelo para la automatización del proceso de determinación de riesgo de deserción en alumnos universitarios*" tuvo como objetivo: Desarrollar un indicador que permitía clasificar en forma automática a los alumnos con mayor riesgo de deserción de las carreras de Ingeniería de la Universidad de Las Américas, para trabajar un programa docente y de tutorías, focalizado con el objeto de mejorar los índices de retención.

  El proceso de minería de datos, siguiendo la metodología CRISP-DM adoptada en la investigación, se organizó en seis fases, cada una de ellas interactúan entre sí de forma iterativa. Los algoritmos aplicados fueron los de Redes Neuronales Artificiales, Árboles de decisión y algoritmo K-medianas, con los cuales se intenta

predecir el comportamiento de los estudiantes, basándose en los siguientes factores: puntaje promedio obtenido en la Prueba de Selección Universitaria (PSU), promedio de notas obtenido en la enseñanza media, edad del estudiante al ingresar y su sexo.

El conjunto de datos de pruebas fue utilizado para calcular la exactitud de los algoritmos, y dado que dentro de los límites de la investigación no fue posible conseguir datos completos y confiables, para evitar los problemas se propuso una metodología para afrontar investigaciones de minería de datos educativa.

La tesis finaliza con las siguientes conclusiones:

- Los algoritmos utilizados para desarrollar los modelos predictivos fueron Redes Neuronales Artificiales, algoritmo K-medianas y Árboles de Decisión.
- El algoritmo de Redes Neuronales Artificiales fue el modelo con mejor desempeño para agrupar al alumnado, pero los valores obtenidos no permitían validar positivamente el modelo.
- El *dataset* no contenía la información necesaria para desarrollar un mejor modelo predictivo.
- El investigador propuso actividades y datos (variable) que debían ser implementados en investigaciones posteriores.
- El investigador utilizó la metodología CRISP-DM para la recopilación, análisis y preparación del *dataset*.

- Acosta, P. et all (2011), alumno de la Universidad Nacional de Ingeniería (Lima, Perú), en su tesis titulada: "*Predicción del rendimiento académico en la educación Superior usando minería de datos y su comparación con Técnicas estadísticas*" tuvo como objetivo: La utilización de técnicas de minerías de datos y técnicas estadísticas aplicadas a la universidad peruana, que permitiría al estudiante predecir su rendimiento académico en un curso en que se deseaba inscribir en un nuevo ciclo.

  La metodología empleada consistió en aplicar modelos predictivos (redes neuronales artificiales, regresión logística y regresión múltiple), para que el

estudiante universitario pudiese deducir su rendimiento académico de cada curso en que deseaba inscribirse. Para ello primero se realizó una selección de las variables predictoras, en base a la experiencia de los autores en la cátedra universitaria y haciendo un confrontación de dichas variables con las usadas en trabajos de investigación publicados y relacionados al tema. Se usó la base de datos académica de los alumnos de una especialidad (previamente preparados) y el currículo correspondiente, y mediante un programa en Java se obtuvo los datos para las variables seleccionadas.

Se aplicaron las técnicas de redes neuronales artificiales de Retropropagación (*back-propagation*) y de regresión logística a 7 cursos de la especialidad de Ingeniería Química de la Universidad Nacional de Ingeniería (UNI), correspondientes a los períodos académicos comprendidos entre el 1993-1 y el 2010-2.

La tesis finaliza con las siguientes conclusiones:

- El *dataset* contenía la información necesaria de los ciclos pasados, para realizar un modelo de pronóstico adecuado.
- El algoritmo de la red neuronal artificial permitió predecir si un alumno aprobaría un determinado curso con aciertos superiores a 70% y errores cuadráticos medios inferiores a 0.14.
- Las variables críticas fueron: "promedio ponderado acumulado" y "antigüedad en años del alumno".

- Pacco, R. (2015)), alumno de la Universidad Peruana Unión (Lima, Perú), en su tesis titulada: "*Análisis Predictivo Basado en Redes Neuronales no Supervisadas Aplicando Algoritmo de K-medias y CRISP-DM para Pronóstico de Riesgo de Morosidad de los Alumnos en la Universidad Peruana Unión*" tuvo como objetivo: Implementar el análisis predictivo, basado en redes neuronales artificiales no supervisadas, aplicando el algoritmo de K-medias y CRISP-DM, para ayudar mucho en la predicción de riesgo de morosidad de los alumnos de la Universidad Peruana Unión, año 2015.

Para lograr el objetivo planteado se utilizó la metodología CRISP-DM, la cual consistía en un grupo de deberes descritos en 4 fases de abstracción: fase, tarea genérica, tarea especializada, e instancia de proceso, los cuales están jerárquicamente organizados que van desde lo genérico hasta lo más detallado.

Dicha metodología incorporaba un modelo y una guía, establecidos en seis fases, algunas de las cuales son bidireccionales, es decir que permiten revisar fases anteriores de forma parcial o total.

La tesis finaliza con las siguientes conclusiones:

- Se logró obtener la segmentación de los Clústeres, adicionalmente se ha obtenido la integración de cada herramienta utilizada en el proyecto de forma correcta.

- Después de la creación del modelo de *Clustering* y de la elaboración del *Business Analytics* (BA), se pudo perfeccionar la toma de decisiones, mediante el manejo dinámico de los reportes y de la herramienta misma.

- Se logró identificar a estudiantes de la misma universidad que tenían características similares a las de los Clústeres hallados.

- Laura, L. (2016), alumno de la Universidad Católica de Santa María (Arequipa, Perú), en su tesis titulada: "*Estudio Comparativo de Técnicas No Supervisadas de Minería de Datos para Segmentación de Alumnos*" tuvo como objetivo: Realizar una investigación comparativa de técnicas no supervisadas de minería de datos para segmentación de alumnos de la misma Universidad.

  Al conservar una perspectiva más profunda con respecto a los objetivos del proyecto y porque es la metodología mayoritariamente usada en minería de datos, se eligió a la metodología CRISP-DM.

  Para la segmentación académica, se implementó en R el algoritmo de *Clustering* particional K-medias y se logró obtener una mejor eficacia de agrupamiento en las

medidas internas, en las distancias intra-clúster e inter-clúster y en el coeficiente de silueta.

La tesis finaliza con las siguientes conclusiones:

- El *dataset* contenía la información académica del segundo semestre del año 2014 y que correspondía al alumnado de Ingeniería de Sistemas.
- Para realizar el análisis comparativo de la segmentación académica, se utilizaron K-medias, PAM y *Clustering* jerárquico aglomerativo obteniendo diferentes resultados entre ellos.
- El coeficiente de silueta, fue la métrica evaluadora de los resultados obtenidos por los algoritmos citados en el párrafo anterior, con lo cual se determinó que el algoritmo K- medias implementaba agrupaciones de mejor calidad, por lo tanto la segmentación académica determino tres niveles (básico, intermedio y avanzado) para el reforzamiento de los alumnos.

- Hernández, J. (2015), alumno del Tecnológico Nacional de México (La Paz, Baja California Sur, México), en su tesis titulada: "*Modelo de Minería de Datos para Identificación de Patrones que Influyen en el Aprovechamiento Académico*" tuvo como objetivo: Diseñar y generar un modelo de minería de datos para la identificación de patrones de comportamiento relacionados con el desempeño académico de alumnos en una institución de educación media superior.

  La metodología empleada consistió en la implementación del modelo CRISP-DM, con la cual se organizó el proceso de la minería de datos en las 6 etapas descritas en el modelo, los cuales interactúan entre ellas de forma iterativa y se caracterizan por poseer fases más específicas y minuciosas. Los modelos aplicados fueron los de Árboles de decisión, RNA y algoritmo K-medianas, con la finalidad de poder pronosticar el comportamiento del alumnado.

  La tesis finaliza con las siguientes conclusiones:

- La veracidad de los modelos fue calculada en base al *dataset* obtenido, y los resultados predictivos fueron positivos.

- El modelo de predicción más efectivo fue el algoritmo RNA, con un mejor comportamiento respecto al árbol de decisión y algoritmo K-medianas.

- La toma de decisiones implementada, estuvo orientada a la programación de los aspectos docente, psicopedagógico y técnico-administrativo para evitar la posibilidad de rezago estudiantil.

- Ordoñez, K. (2013), alumno de la Universidad Técnica Particular de Loja (Loja, Ecuador), en su tesis titulada: "*Aplicación de técnicas de minería de datos para predecir la deserción de los estudiantes de primer ciclo de la Modalidad Abierta y a Distancia de la UTPL*" tuvo como objetivo: crear un modelo predictivo que permitiese conocer cuáles eran las posibles causas por lo que un alumno decidía abandonar sus estudios universitarios, a través del análisis de las características de los estudiantes desertores de primer ciclo de la Modalidad Abierta y a Distancia de la UTPL

  La metodología CRISP-DM, creada en 1996, se utilizó para formular el modelo en términos de un proceso jerárquico describiendo tareas en cuatro niveles (de general a específico) y en cada uno se detallan las tareas de forma ordenada y apropiada.

  Se aplicaron las técnicas de *Clustering*, árboles de decisión y reglas de asociación, con las cuales se logró un modelo que prediga la deserción de los estudiantes, usando un *dataset* de la institución educativa y obtener como resultado un modelo útil.

  La tesis finaliza con las siguientes conclusiones:

  - Los resultados hallados en las diferentes carreras analizadas muestran que los campos: estado civil y género tienen una influencia nula en la decisión de un estudiante para desertar. Por el contrario, aquellas características que integran la productividad académica obtenida por el alumno, tienen un peso importante cuando deciden tomar dicha decisión.

- La técnica de *Clustering* fue la que brindo resultados más eficaces respecto de las otras dos, porque se pudo obtener las principales características de una posible deserción.

- Gutiérrez, J. (2012), alumno de la Universidad Autónoma de Manizales (Manizales, Colombia), en su tesis titulada: "*Descubrimiento de Conocimientos en la Base de Datos Académica de la Universidad Autónoma de Manizales Aplicando Redes Neuronales*" tuvo como objetivo: generar conocimiento útil para encontrar posibles causas del problema de la deserción estudiantil de la Universidad Autónoma de Manizales a partir de la inmensa información académica generada por los sistemas transaccionales de la universidad.

  La etapa uno de la metodología propuso obtener investigaciones sobre el problema de la deserción académica y problemas similares relacionados con la educación superior, en el contexto internacional y nacional; en la fase dos se logró obtener información académica de las diversas fuentes transaccionales (sistemas) de la Universidad Autónoma de Manizales; y en la última etapa mediante técnicas de algoritmos genéticos, RNA y árboles de decisión (definidas después del proceso de extracción) se realizó el análisis de la información.

  La tesis finaliza con las siguientes conclusiones:

  - Los resultados indicaron que los estudiantes más propensos a la deserción académica provenían en gran proporción de regiones alejadas de la Universidad Autónoma de Manizales.

  - Los resultados obtenidos por medio del análisis debían ser utilizados en proyectos de definición de estrategias que promuevan como objetivo la permanencia de estudiantes de esta población altamente vulnerable a la deserción.

  - Se identificó a los programas que presentaban mayor número de deserciones como aquellos de más larga duración, por esta razón era necesario tener presente

dentro de los futuros programas académicos el riesgo latente de deserción a mayor duración del programa

- Muñoz, J. (2014), alumno de la Universidad Politécnica de Cartagena (Cartagena, Colombia), en su tesis titulada: "*Estudio de la predictibilidad del rendimiento académico de alumnos en asignaturas de segundo curso*" tuvo como objetivo: Crear una red neuronal artificial de clasificación con capacidad para predecir potenciales abandonos, alumnos que tenían grandes posibilidades de abandonar la carrera universitaria que iniciaron. Ello podría ayudar a la universidad para tomar medidas de refuerzo sobre tales alumnos y así, evitar este desenlace.

  La metodología empleada para la resolución del problema mediante RNA fue, en general, independiente de las condiciones del asunto en discusión, por lo tanto el primer paso fue recoger y preparar los datos, el segundo paso fue separar la data en tres grupos (entrenamiento, validación y test), posteriormente se realizó el entrenamiento del modelo y ajustar los criterios de entrenamiento.

  Los resultados obtenidos en la data de entrenamiento y validación, fueron analizados mediante la magnitud de los pesos y por 3 análisis de sensibilidad diferentes con el fin de conocer la confiabilidad del modelo de clasificación para poder predecir el potencial de abandono o no abandono de un determinado alumno.

  La tesis finaliza con las siguientes conclusiones:

  - El resultado del algoritmo de predicción fue malo para determinar la nota de un alumno.

  - El resultado del algoritmo de predicción fue bueno para clasificar el abandono de la totalidad del alumnado, pero malo para grupos pequeños.

  - Dicha divergencia fue explicada por la escasa información relevante que contenía el *dataset*.

- El problema de fondo de la investigación se afrontaba a partir de la perspectiva psicológica, social y humanista, pero con la perspectiva de la Ciencia de los Datos ese enfoque ha cambiado.

- Villamarín, J. (2017), alumno de la Universidad Autónoma de Occidente (Santiago de Cali, Colombia), en su tesis titulada "*Análisis de la Deserción Estudiantil en la FCECEP utilizando Machine Learning específicamente Mapas auto organizados de Kohonen*" tuvo como objetivo: Analizar la deserción estudiantil en las carreras tecnológicas de ingeniería en la institución educativa FCECEP con un Mapa Auto-Organizado de Kohonen determinando las variables de mayor influencia de este fenómeno y las características comunes de los diferentes grupos de riesgos de deserción.

  La metodología empleada consistió en un método mixto que pretendía integrar la recolección de datos (cuantitativo), comprobar supuestos (evidencia numérica) y el análisis estadístico (patrones de comportamiento cualitativos), para poder percibir y deducir los factores de alarma para que un alumno pueda desertar.

  La tesis finaliza con las siguientes conclusiones:

  - Como procedimiento previo a la utilización de los Mapas Auto Organizados de Kohonen se tuvo que realizar un pre-procesamiento de los datos para poder obtener un óptimo *dataset*.

  - Fueron identificadas las variables que tenían una incidencia preponderante en la deserción del alumno, así como las características en común que tienen los diversos grupos para desertar.

  - La RNA (Mapas Auto Organizados de Kohonen) implementada, demostró ser una opción adicional para analizar datos de deserción por medio de la traficación en 2D y 3D.

- Aftab, J. (2017), alumno de *Capital University of Science and Technology* (Islamabad, Pakistán), en su tesis titulada: "*Student Retention in Higher Education Institutions*" tuvo como objetivo: identificar las variables, en el contexto local, que originan la deserción en *Capital University of Science and Technology* (CUST).

  Mediante la investigación se logró identificar ciertos factores como causa principal de desgaste en universidades extranjeras. Dichos factores con algunos nuevos fueron evaluados para determinar la importancia de su participación en la deserción del estudiante del CUST y en qué medida.

  El conjunto de datos se recopiló desde el registro del CUST y constaba de atributos independientes que se obtuvieron de las categorías de datos académicos, demográficos y preuniversitarios. Algunos atributos se utilizaron en investigaciones anteriores porque jugaban un papel importante en la deserción de los estudiantes y otros eran nuevos, y se deseaba comprobar si juegan un papel detrás del desgaste de los estudiantes.

  La tesis finaliza con las siguientes conclusiones:

  - En los experimentos se utilizó WEKA (*software* para realizar el análisis de la información), al realizar el trabajo de clasificación se basaron en el árbol de decisión y para la evaluación, se utilizó el método "10-K *fold cross-validation*" (validación cruzada con 10 repeticiones), donde K, es el número de subconjuntos en que se dividen los datos.

  - Para la verificación de los cursos más influyentes detrás de la deserción de los estudiantes, se preparó un archivo de datos de experimentos que consistía en Grado del curso de programación de computadora, Introducción a la calidad de la computadora, Grado del cálculo y Grado inglés del primer semestre. Las técnicas aplicadas a los datos revelaban los resultados: programación de computadora (CP), Cálculo (Cal-I) e introducción a la computadora (ITC) eran cursos "efectivos" para predecir el desgaste de los estudiantes.

  - Para verificar el impacto del pedagogo sobre el desgaste de los estudiantes preparamos otro archivo para experimentos que incluía nombre del pedagogo de todos los cursos mencionados. Luego se aplicó la extracción de datos y algunos

enfoques estadísticos sobre el resultado revelaba que la programación de computadoras y el cálculo La metodología del docente tuvo un impacto leve en la deserción de los estudiantes.

- Para verificar el impacto sobre el rendimiento estudiantil, recopilamos datos de 163 estudiantes a los que se le ofreció tutoría, los resultados revelaban que el rendimiento de los estudiantes que recibieron tutoría era ligeramente mejor que el de otros estudiantes.

2.3. Estructura teórica y científica que sustenta el estudio

2.3.1. Aspectos Académicos.

2.3.1.1. La Educación Superior.

El 2009 se realizó la "Conferencia Mundial sobre Educación Superior" organizada por la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) y se menciona en la Declaración Final, preámbulo (cuarto párrafo), que: "En ningún otro momento de la historia ha sido más importante que ahora la inversión en los estudios superiores, por su condición de fuerza primordial para la construcción de sociedades del conocimiento integradoras y diversas, y para fomentar la investigación, la innovación y la creatividad".

En concordancia con dicha declaración, el artículo N° 013 de la Constitución Política del Perú del año 1993 expresa lo siguiente: "La educación tiene como finalidad el desarrollo integral de la persona humana" y se amplía el concepto con el artículo N° 09 de la Ley General de Educación, que señala siguiente:

> (...) formar personas capaces de lograr su realización ética, intelectual, artística, cultural, afectiva, física, espiritual y religiosa, promoviendo la formación y consolidación de su identidad y autoestima y su integración adecuada y crítica a la sociedad para el ejercicio de su ciudadanía en armonía con el entorno, así como el desarrollo de sus capacidades y habilidades para vincular su vida con el mundo del trabajo y para afrontar los incesantes cambios en la sociedad y el conocimiento.

2.3.1.2. Calidad y Rendimiento Académico.

Cuando un joven destina su tiempo y dinero en el servicio de Educación Superior, y desea determinar los beneficios de dicha inversión, le toma años averiguarlo. Debido a que, el alumno espera a terminar su carrera para comenzar a trabajar, en ese momento observa si el sueldo y su lugar de labores, concuerdan con lo que aspiraba recibir en contraprestación al esfuerzo que le produjo culminar sus estudios. (SINEACE, 2013, pág. 59)

En ese sentido, la educación es un proceso mediante el cual se busca el desarrollo completo del individuo, prepararlo para enfrentar la vida y prepararlo para un desarrollar labores de calidad. Si la calidad en educación evidencia un alto estándar, entonces su aportación al progreso social, económico y cultural del Perú será determinante, porque existe un fuerte vínculo entre desarrollo educativo y las posibilidades de alejarse de las carencias. (SINEACE, 2013, pág. 23)

De acuerdo a lo mostrado por el Diccionario de la Real Academia Española (RAE), el concepto del vocablo reprobación tiene su origen en el término latino “reprobatĭo” que se refiere a la acción y efecto de reprobar, y su origen en latín es “reprobāre”, y conceptualiza el hecho de no aprobar. Para entender con exactitud lo que representa reprobación, hay que conocer qué se entiende con aprobar. Se trata de otro concepto que proviene del latín: “aprobare”; es calificar como bueno a alguien o algo. Reprobar es no alcanzar una calificación positiva, es lograr una calificación que no es suficiente para cumplir con un determinado objetivo.

Terminar los estudios superiores es el objetivo de la mayoría de estudiantes de pregrado, si se tiene niveles de desaprobación altos, entonces la posibilidad de cumplir el objetivo es bajo. Por consiguiente, la mejora del bienestar individual y social será más difícil de alcanzar. (SINEACE, 2013, págs. 15-16)

El rendimiento académico puede ser alto o bajo, en otras palabras, alcanzar una calificación superior o inferior con respecto a un determinado límite; pero hay una ligera diferencia, la frontera entre una calificación superior o inferior es variable, es definida por cada institución educativa, podría ser 11 o 12 o 14 si la escala de calificación es vigesimal u 70, 75, 80 u 85 si la escala de calificación en centesimal. En la presente investigación la escala de calificación es vigesimal y la frontera es 10.5, es

decir menores a 10.5 tienen bajo rendimiento académico y mayores o iguales a 10.5 tienen alto rendimiento académico.

2.3.1.3. Universidad Ricardo Palma.

La Universidad Ricardo Palma (URP), fue fundada oficialmente como universidad particular el 01 de julio de 1969 de acuerdo a lo establecido en el Decreto Ley N° 017723, su reconocimiento se encuentra establecido en la ley N° 023733, artículo N° 097, inciso N° 030 y el visto bueno para el ejercicio de sus actividades fue otorgado por el Consejo Nacional de la Universidad Peruana mediante la resolución N° 0307 en el año 1971.

La URP tiene su local central ubicado en Av. Benavides 5440, Urb. Las Gardenias, distrito de Santiago de Surco, provincia y departamento de Lima, donde se ofrecen 19 carreras profesionales y se han construido pabellones para cada una de las 8 facultades, los cuales cuentan con salones de clase, aulas para laboratorios, bibliotecas especializadas y salas de conferencias; también se ubica una edificación para funciones administrativas y otra para la atención médica.

2.3.1.4. Programa de Estudios Básicos.

El Programa de Estudios Básicos (PEB) se implementó en la Universidad Ricardo Palma a partir del semestre 2006-II mediante ACU (Acuerdo de Consejo Universitario) Nº 01962-2006 emitido el 18 de julio de 2006; a través del cual se imparten a los alumnos ingresantes cursos desde el primer ciclo y que siguen la siguiente proporción:

- Primer Ciclo: 60% cursos del PEB y 40% cursos de carrera.
  Los 5 cursos a desarrollar son: Taller de Método de Estudio Universitario, Taller de Comunicación Oral y Escrita I, Matemática, Inglés I y Actividades Artísticas y Deportivas.

- Segundo Ciclo: 50% cursos del PEB y 50% cursos de carrera.
  Los 5 cursos que se dictan son: Psicología General, Lógica y Filosofía, Formación Histórica del Perú, Taller de Comunicación Oral y Escrita II e Inglés II.

- Tercer Ciclo: 40% cursos del PEB y 60% cursos de carrera.
  Los últimos 3 cursos que se desarrollan son: Recursos Naturales y Medio Ambiente, Realidad Nacional e Historia de la Civilización.

La desaprobación ha sido estudiada de diferentes maneras, pero siempre a posteriori, es decir se ha efectuado el análisis después de conocerse los resultados de las evaluaciones. Mediante las técnicas de *Machine Learning*, se pretende predecir la posibilidad de desaprobación de un estudiante y tomar acciones correctivas a priori.

2.3.2. Machine Learning (ML).

También es conocido como Aprendizaje Automático (AA) o Aprendizaje de Máquina.

El aprendizaje automático es un sub campo de la informática (pertenece a la IA, que a su vez es una disciplina de la informática), mediante el cual se realiza la investigación y la creación de algoritmos que puedan encontrar patrones contenidos en un conjunto de datos. (Rodriguez Pacheco, 2015, pág. 47)

Mediante los algoritmos del aprendizaje automático se extrae información (patrones) de diversas fuentes de datos, por lo tanto, el soporte tecnológico de la minería de datos es el aprendizaje automático. (Witten, Frank, Hall, & Pal, 2017, pág. 6)

El aprendizaje automático desarrolla investigaciones para mejorar el rendimiento o aprendizaje de las computadoras en función de los datos. Asignar un diagnóstico a un paciente según sus características observadas (género, presión arterial, presencia o ausencia de ciertos síntomas, etc.) al compararlas con otros pacientes, es un problema representativo de aprendizaje automático; o implementar un programa para que la computadora pueda reconocer de forma automática códigos postales de correo escritos a mano al compararlos con el conjunto de ejemplos que ha aprendido. (Han, Kamber, & Pei, 2012, pág. 24)

En términos generales, el aprendizaje automático se puede dividir en varias categorías; las de mayor arraigo son dos: el supervisado y el no supervisado. (Rodriguez Pacheco, 2015, pág. 47)

2.3.2.1. Categorías en Machine Learning

A. Aprendizaje (Método) Supervisado o *Supervised Learning* o Aprendizaje Predictivo

   Los algoritmos (técnicas o métodos) de aprendizaje supervisado intentan construir un patrón a partir de datos conocidos previamente, lo que nos permite inferir

(predecir o clasificar) el valor de salida de los nuevos datos, conociendo solo sus características. (Garreta & Moncecchi, 2013, pág. 20)

Normalmente, los datos se componen de un conjunto de observaciones. Cada observación posee un conjunto de diversas características (variables o campos) llamadas predictores, y queremos predecir un resultado (valor de salida, respuesta, atributo objetivo, etiqueta o target) a los cuales los representamos como instructores, porque los algoritmos aprenden de ellos. (Rodriguez Pacheco, 2015, pág. 47)

El objetivo final de la función creada por el algoritmo es la extrapolación de su comportamiento hacia nuevas observaciones, es decir, la predicción para la toma de decisiones. Esta predicción corresponde al valor de salida de un algoritmo de aprendizaje supervisado que podría denominarse: (Rodriguez Pacheco, 2015, pág. 47)

1. Regresión

   Cuando la respuesta es un valor que pertenece a un conjunto continuo (números en la recta numérica real, ejemplo: predecir el ancho del pétalo de una flor), entonces, estamos tratando de resolver un problema de regresión (el término fue acuñado por Francis Galton). (Garreta & Moncecchi, 2013, pág. 20)

2. Clasificación

   Cuando la respuesta es una categoría que pertenece a una clase (conjunto discreto, ejemplo: una lista de especies de flores), entonces, enfrentamos un problema de clasificación. (Garreta & Moncecchi, 2013, pág. 20)

   Los modelos de clasificación generalmente generan dos tipos de predicciones. Al igual que los modelos de regresión, los algoritmos de clasificación producen una predicción de valor continuo, que generalmente tiene forma de probabilidad (es decir, los valores pronosticados de la pertenencia a una determinada clase para cualquier muestra individual se encuentran entre 0 y 1 y suman 1). Además de la predicción continua, los algoritmos de clasificación generan una clase predicha, que se presenta en forma de una categoría discreta. Para la mayoría de las aplicaciones prácticas, se requiere una predicción de categoría discreta para tomar una decisión. (Kuhn & Johnson, 2013, pág. 247)

Podemos dividir el proceso del Aprendizaje Supervisado en dos etapas: (Rodriguez Pacheco, 2015, pág. 48)

1. Modelado o Entrenamiento y Prueba

   Es la etapa donde comenzamos con datos en bruto que se utilizarán para entrenar el modelo y continuamos con la definición de las variables que se aprovecharan para construir el modelo, con la posibilidad de reducirlas o transformarlas. (Rodriguez Pacheco, 2015, pág. 48)

   Procedemos a entrenar el modelo, y finalmente llevamos a cabo la evaluación. Es importante tener en cuenta que el entrenamiento, la construcción y la validación del modelo forman un proceso iterativo dirigido a lograr el mejor modelo posible, lo que significa que es posible que tengamos que regresar a un paso anterior para hacer ajustes. (Rodriguez Pacheco, 2015, pág. 48)

   Posteriormente de haber obtenido los modelos y haberlos ejecutado en la Data de Prueba, se aplican Indicadores de Performance a cada uno, para determinar cuál de los modelos es el que presenta los mejores indicadores.

2. Predicción

   Ya tenemos el modelo y una serie de nuevas observaciones. Usando el modelo que construimos y probamos, se ejecuta una predicción para nuevos datos y se generan los resultados. (Rodriguez Pacheco, 2015, pág. 48)

Como referencia, algunos ejemplos de modelos de aprendizaje supervisado son: (Rodriguez Pacheco, 2015, pág. 48)

1. Regresión Lineal o Regresión

   a) Regresión lineal simple

   b) Regresión lineal múltiple

2. Regresión No Lineal

   a) Regresión exponencial

   b) Regresión logarítmica

c) Regresión polinómica

3. Regresión Logística
4. Redes Neuronales Artificiales
5. Árboles de Decisión
    a) Algoritmo CART.
    b) Algoritmo Chaid
    c) Algoritmo C5.0
6. Máquina de Soporte Vectorial (SVM)
7. Naive Bayes

B. Aprendizaje (Método) No Supervisado o *Unsupervised Learning* o algoritmos no supervisados

El objetivo del aprendizaje no supervisado es describir las asociaciones y relaciones en un conjunto de datos. La diferencia fundamental del aprendizaje supervisado es que los datos no tienen un valor de salida, por lo que no tiene respuesta que predecir y más bien trata de encontrar estructuras de datos por su relación. (Rodriguez Pacheco, 2015, pág. 48)

En contraste con el aprendizaje supervisado, en el que el valor del modelo se deriva principalmente de la predicción, en el aprendizaje no supervisado, los hallazgos obtenidos durante la fase de modelado podrían ser suficientes para cumplir el propósito. (Rodriguez Pacheco, 2015, pág. 49)

Por ejemplo, supongamos que tiene un conjunto de datos compuesto por correos electrónicos y desea agruparlos por temas. Podemos usar como criterio diferenciador, por ejemplo, las diferentes palabras usadas en cada uno de los asuntos. (Garreta & Moncecchi, 2013, pág. 20)

Como ejemplos de modelos tenemos:

1. Agrupamiento o *Clustering*

a) Algoritmo K-*means* o K-medias.

b) Algoritmo K-*medoids* o K-medoides o algoritmo PAM.

c) Algoritmo CLARA.

d) Algoritmo KNN o K-*Nearest Neighbors* (vecinos más cercanos).

2. Asociación o Reglas de Asociación

3. Redes Neuronales Artificiales

Después de haber obtenido los modelos tenemos el problema de no disponer de una respuesta verdadera que sepamos a priori, por lo que es mucho más complicado poder determinar qué medida de eficiencia se utilizará.

C. Aprendizaje Semi-supervisado o *Semi-supervised Learning*

Variedad de algoritmos donde se utilizan ejemplos etiquetados y no etiquetados para que un modelo pueda aprender. Desde un determinado punto de vista, los ejemplos rotulados se utilizan para determinar los patrones del grupo y los ejemplos sin rotular se usan para refinar los linderos entre los grupos. En un problema de clasificación binaria, podemos imaginar que el grupo de ejemplos positivos pertenecen a una clase y los restantes a la clase de los ejemplos negativos. (Han, Kamber, & Pei, 2012, pág. 25)

D. Aprendizaje Activo o *Active Learning*

Perspectiva que otorga libertad a los programadores para que desarrollen un rol activo para que un modelo aprenda. Por ejemplo, se puede solicitar a una persona (generalmente un experto en el tema) que clasifique un registro, del total de ejemplos no etiquetados, la finalidad es mejorar las características del modelo mediante la trasmisión de conocimiento de parte de los expertos, el problema radica en determinar cuántos registros serán etiquetados por los expertos. (Han, Kamber, & Pei, 2012, pág. 25)

A continuación se muestra algunas técnicas asociadas al tipo de tarea que realizan:

Tabla 01: Algoritmos de *Machine Learning* asociados al tipo de tarea.

Nombre	Predictivo		Descriptivo		
	Clasificación	Regresión	Agrupación	Reglas de Asociación	Correlación Factorización
Redes Neuronales Artificiales	**X**	**X**	**X**		
Árbol de Decisión CART	**X**	**X**			
Regresión Lineal y Logarítmica		**X**			
Regresión Logística	**X**			**X**	
K-means			**X**		
Análisis Componentes Principales					**X**
Algoritmos Genético y Evolutivo	**X**	**X**	**X**	**X**	**X**
Máquinas de Soporte Vectorial	**X**	**X**	**X**		

Fuente: (Han, Kamber, & Pei, 2012). Elaboración: Propia, 2019

2.3.3. Minería de Datos o Data Mining.

Busca encontrar respuestas a problemas ocultos en las bases de datos. El diagnostico de dichos problemas son los llamados patrones o patrones de comportamiento o patrones estructurales, que son obtenidos mediante algoritmos de *Machine Learning*. En el mundo actual, centrado en los requerimientos del cliente, la necesidad de encontrar patrones para brindar mejores servicios es el motor que empuja el negocio. (Witten, Frank, Hall, & Pal, 2017, pág. 5)

La minería de datos es un proceso semiautomático (pero debería ser automático) mediante el cual se intenta encontrar, en los datos, patrones que otorguen ventajas diferenciadoras con respecto a la competencia, los cuales deben ser de fácil comprensión y disponibles para diversos fines. (Witten, Frank, Hall, & Pal, 2017, pág. 6)

La minería de datos, como una sucesión de pasos para la obtención de nuevo conocimiento, utiliza matemáticas, estadística, inteligencia artificial, bases de datos, reconocimiento de patrones o aprendizaje automático. De hecho, a veces algunos de estos términos se consideran sinónimos, lo que de hecho es incorrecto. El siguiente

diagrama ilustra algunas disciplinas involucradas en el proceso de minería de datos: (Rodriguez Pacheco, 2015, pág. 46)

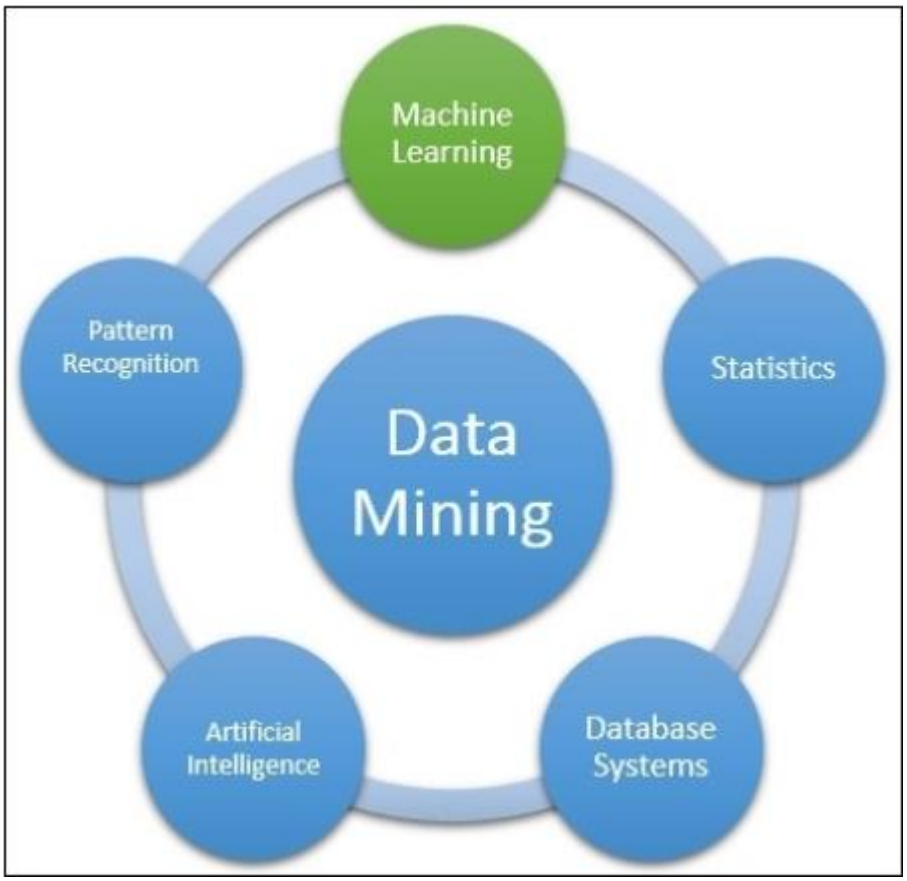


Figura 02: Disciplinas involucradas en el proceso de minería de datos.
Fuente: (Rodriguez Pacheco, 2015, pág. 46)

2.3.4. Métodos o Técnicas de Ensamble.

Teniendo en cuenta la dimensión y la calidad del *dataset*, podemos obtener respuestas más fiables uniendo las respuestas de distintos algoritmos. *Bagging*, *Boosting* y *Stacking* son métodos del *Machine Learning* que obtienen una respuesta basada en, una combinación de respuestas provenientes de distintos modelos (denominados modelos débiles o *weak models*). A dicho proceso, se le llama ensamble o ensamble de modelos y generalmente la eficacia en la predicción de su respuesta es más elevada que el de un solo modelo que lo compone y se puede emplear tanto a problemas de clasificación como a problemas de predicción. (Witten, Frank, Hall, & Pal, 2017, pág. 480)

Sin embargo, la desventaja que comparten cada uno es que, son complejos de examinar y poder determinar que factor fue determinante para perfeccionar la respuesta. (Witten, Frank, Hall, & Pal, 2017, pág. 481)

Hay varias formas de combinar las respuestas provenientes de los modelos. En el caso de problemas de regresión, los resultados de cada modelo se promedian, y para problemas de clasificación se promedian las probabilidades de clase, se puede utilizar el promedio aritmético o el promedio ponderado (el peso de cada modelo está relacionado con la importancia del mismo); para los problemas de clasificación, se utiliza el voto mayoritario, es decir, la clase emitida por cada modelo es un voto y la clase que obtenga la mayoría de los votos se convierte en la respuesta, se puede utilizar la suma aritmético o la suma ponderada. (Witten, Frank, Hall, & Pal, 2017, pág. 482)

2.3.4.1. Boosting.

Desde hace más de 20 años los algoritmos de *Bagging*, *Boosting* y *Stacking* han obtenido rendimientos que han sorprendido a los investigadores, que al estudiar el motivo, han hallado que el algoritmo *Boosting*, probablemente el más eficaz de los métodos de ensamble, se encuentra fuertemente ligado con los fundamentos estadísticos de los modelos aditivos, logrando mejorar su performance con la renovación del mismo. (Witten, Frank, Hall, & Pal, 2017, pág. 480)

El algoritmo *Boosting* se encuentra relacionado con la teoría del aprendizaje de Kearns y Valiant, la cual indica que, una relación de clasificadores débiles se pueden acoplar para originar un clasificador superior con una tasa de error menor que la de los clasificadores que lo formaron. (Kuhn & Johnson, 2013, pág. 204)

Como principio básico el algoritmo *Boosting* intenta encontrar modelos que se puedan integrarse entre ellos. En primer lugar, busca las similitudes entre las respuestas de cada uno de los modelos que conforman el ensamble, para ello combina todas las respuestas mediante la votación, si el problema es de clasificación, o mediante el promedio, si el problema es de predicción. En segundo lugar, utiliza modelos de igual naturaleza. En tercer lugar, la construcción de los modelos es repetitivo, con la finalidad de que cada nuevo modelo este afectado por la performance de los anteriores, de tal forma que siempre se busque disminuir los casos erróneos. En cuarto lugar, cada uno de los modelos tiene un peso diferente, el cual se encuentra relacionado con su desempeño. (Witten, Frank, Hall, & Pal, 2017, pág. 487)

En la figura siguiente se puede observar que al principio se determina un *peso* igual a cada registro de la data. Como primer paso, se genera el primer modelo y se compara la

salida que arroja el modelo con la respuesta de la data de entrenamiento; las comparaciones que son correctas tienen como consecuencia una disminución del *peso*, y en las incorrectas el *peso* se incrementa, y por lo tanto obtenemos una relación de registros con *peso* pequeño o fáciles y una relación de registros con *peso* elevado o difíciles. Para el segundo paso y los subsiguientes, se genera un modelo que se enfoca en mejorar las salidas de los registros difíciles del modelo anterior, es decir, busca que, al comparar las salidas del nuevo modelo con los registros difíciles del modelo anterior, esta cantidad disminuya; al realizar la nueva comparación el *peso* de un registro puede disminuir o aumentar y como consecuencia el registro se puede volver más fácil o más difícil. A partir del segundo paso, el *peso* de un determinado registro refleja la complejidad de predicción del mismo, o en que magnitud la salida hallada ha sido correcta o incorrecta. La categoría de registro difícil permite establecer modelos que se integran para intentar hallar una salida más idónea. (Witten, Frank, Hall, & Pal, 2017, pág. 487)

Tabla 02: Diagrama del algoritmo *Boosting*.

Generación del Modelo

Asigne igual peso a cada instancia de entrenamiento.
Para cada una de las iteraciones:
  Aplique el algoritmo de aprendizaje al conjunto de datos ponderado y almacene el modelo resultante.
  Calcule el error **ê** del modelo en el conjunto de datos ponderado y almacene el error.
  Si **ê** es igual a cero, o **ê** es mayor o igual a 0.5:
    Terminar la generación del modelo.
  Para cada instancia en el conjunto de datos:
    Si la instancia se clasificó correctamente por modelo:
      Multiplique el peso de la instancia por: **ê** / (1 - **ê**).
  Normalizar el peso de todas las instancias.

Clasificación

Asigne un peso de cero a todas las clases.
Para cada uno (o menos) de los modelos ***t*** :
  Agregue: **- *log* (ê / (1 - ê))** al peso de la clase predicho por modelo.
Clase de retorno con mayor peso.

Fuente: (Witten, Frank, Hall, & Pal, 2017, pág. 488)

¿En qué magnitud el *peso* de cada registro ha de variarse al finalizar cada paso? Se puede contestar, con respecto al error genérico y de la siguiente manera:

$$Peso \leftarrow Peso \times \frac{e}{(1-e)}$$

donde, se considera que ***e*** es el error genérico al comparar las salidas con las respuestas de los datos (porción de un valor comprendido entre 1 y 0); sin embargo el *peso* solo se actualiza para los registros con una comparación correcta, de lo contrario el *peso* no se modifica. Por supuesto que, al finalizar la actualización del peso de todos los registros, se aplica una normalización con el fin de que la suma se mantenga inalterada. El nuevo peso de cada registro se fracciona entre la sumatoria de todos ellos y se multiplica por la sumatoria de los pesos previos, teniendo como consecuencia el incremento del peso de los registros difíciles y la disminución del peso de los registros fáciles. (Witten, Frank, Hall, & Pal, 2017, pág. 488)

Cuando el error genérico es mayor o igual al valor de 0.5, o cuando el valor es 0, entonces el proceso iterativo se detiene y el último modelo generado no es considerado y se procede a eliminarlo. (Witten, Frank, Hall, & Pal, 2017, pág. 488)

El algoritmo *Boosting* genera una serie de modelos y entrega una salida final, dicha salida es producto de la combinación ponderada de todos los modelos utilizados. Para poder determinar el peso de cada modelo, se tiene que observar el desempeño del modelo, por lo tanto un modelo con un ***e*** próximo a 0 le debería tocar un peso elevado, y un modelo con un ***e*** próximo a 0.5 le debería tocar un peso minimizado. (Witten, Frank, Hall, & Pal, 2017, pág. 488)

Para obtener una salida final, se realiza la sumatoria de los pesos de todos los modelos por cada una de las categorías, y se selecciona la de mayor puntaje. (Witten, Frank, Hall, & Pal, 2017, pág. 488)

El algoritmo *Boosting* puede reducir el error genérico de una forma sorprendente. Si al ejecutar más modelos logramos reducir el error genérico, necesariamente no va a revelar ni explicar el comportamiento de nuestra data. Podemos intentar resolver nuestra inquietud basándonos en la eficacia del modelo en sus salidas, por lo tanto el margen estará definido por la diferencia entre la confianza estimada para la clase verdadera y la de la clase predicha más probable que no sea la clase verdadera. Mientras el margen sea más grande, entonces el modelo será más confiable. (Witten, Frank, Hall, & Pal, 2017, pág. 490)

Lo transcendental de los algoritmos *Boosting*, es que se crea basándose en una serie de modelos débiles que no son buenos para todo el conjunto de datos, pero sí para una parte del mismo, de tal manera que cada modelo logra incrementar el rendimiento de todo el conjunto de modelos. Como ejemplo podemos tener modelos de árboles de decisión muy simples para una casuística de clase binaria y obtener excelentes resultados. (Witten, Frank, Hall, & Pal, 2017, pág. 490)

Los algoritmos *Boosting* producen resultados muy satisfactorios en datos nuevos, sin embargo, están expuestos a sufrir "*overfitting*" (sobreajuste), porque puede ocurrir que, al generar un algoritmo *Boosting* y al comparar su precisión con uno de los algoritmos que lo componen, esta resulte inferior que la del modelo individual. (Witten, Frank, Hall, & Pal, 2017, pág. 490)

2.3.4.2. Stacking.

La idea del *Stacking* es entrenar un metamodelo (metaclasificador o *meta-classifier*) cuyas entradas son las respuestas (salidas o predicciones) de distintos algoritmos. Como ejemplo tenemos, un problema de clasificación donde las técnicas de modelado débiles (*weak models*) son: KNN, regresión logística (RL) y SVM, y como metamodelo la Red Neuronal Artificial (RNA). Entonces, la RNA tendrá como entradas las salidas de cada uno de los *weak models* y devolverá como salida la predicción combinada. (Witten, Frank, Hall, & Pal, 2017, págs. 497-498)

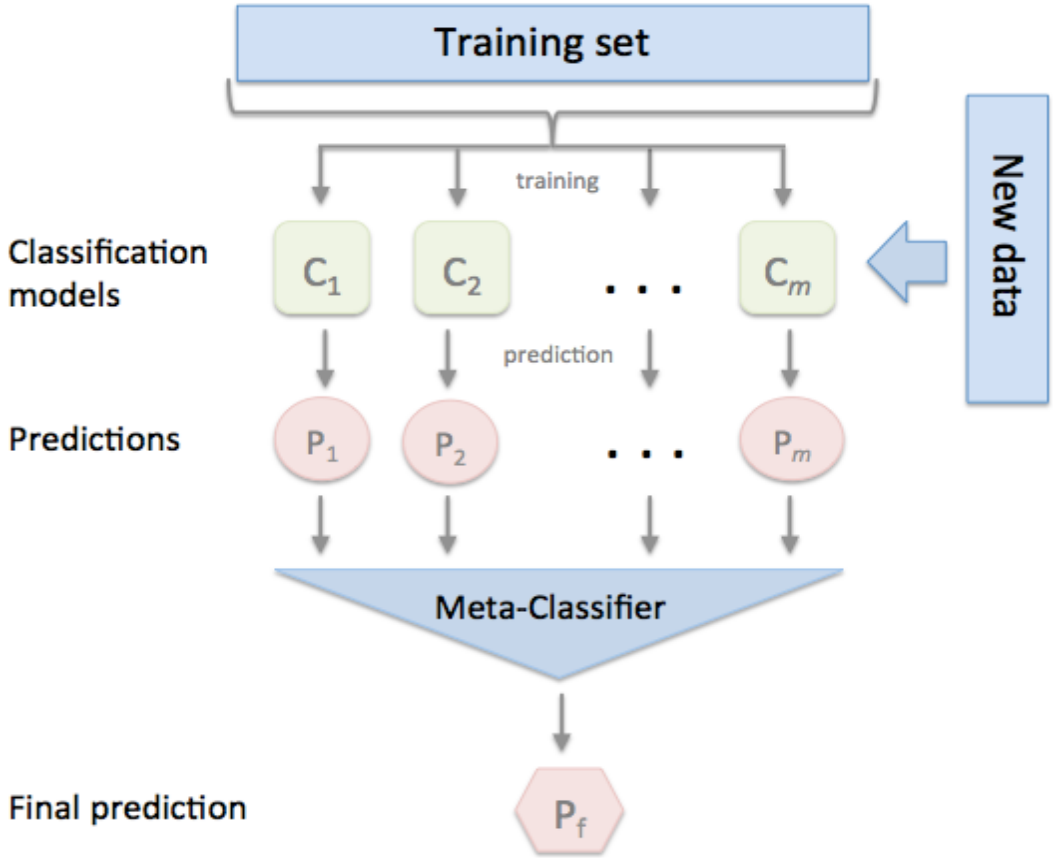


Figura 03: Diagrama del algoritmo *Stacking*.
Fuente: (Raschka, 2014)

2.3.5. Redes Neuronales Artificiales.

2.3.5.1. Neurona Biológica.

De acuerdo a lo mostrado por el Diccionario de la Real Academia Española (RAE), el vocablo neurona tiene su origen en el término griego neûron cuyo concepto es ‘cuerda’ o ‘nervio’.

Poder comprender el cerebro se ha facilitado gracias al trabajo pionero de Ramón y Cajál (1911), al conceptualizar cada neurona como la unidad básica de la estructura del cerebro, una similitud sería un grano de arena en la viga de concreto de un edificio. Simbólicamente, la puertas lógica del chip de silicio es más rápida que la neurona en un orden comprendido entre 5 y 6 veces; los eventos lógicos se producen en nanosegundos, en tanto que los eventos neuronales suceden en milisegundos. No obstante, el cerebro equilibra la velocidad de operación relativamente lenta de una neurona con un número verdaderamente asombroso de neuronas (células nerviosas) y sus interconexiones masivas entre ellas, dando como resultado una estructura enormemente eficiente denominada cerebro. (Haykin, 2009, pág. 6)

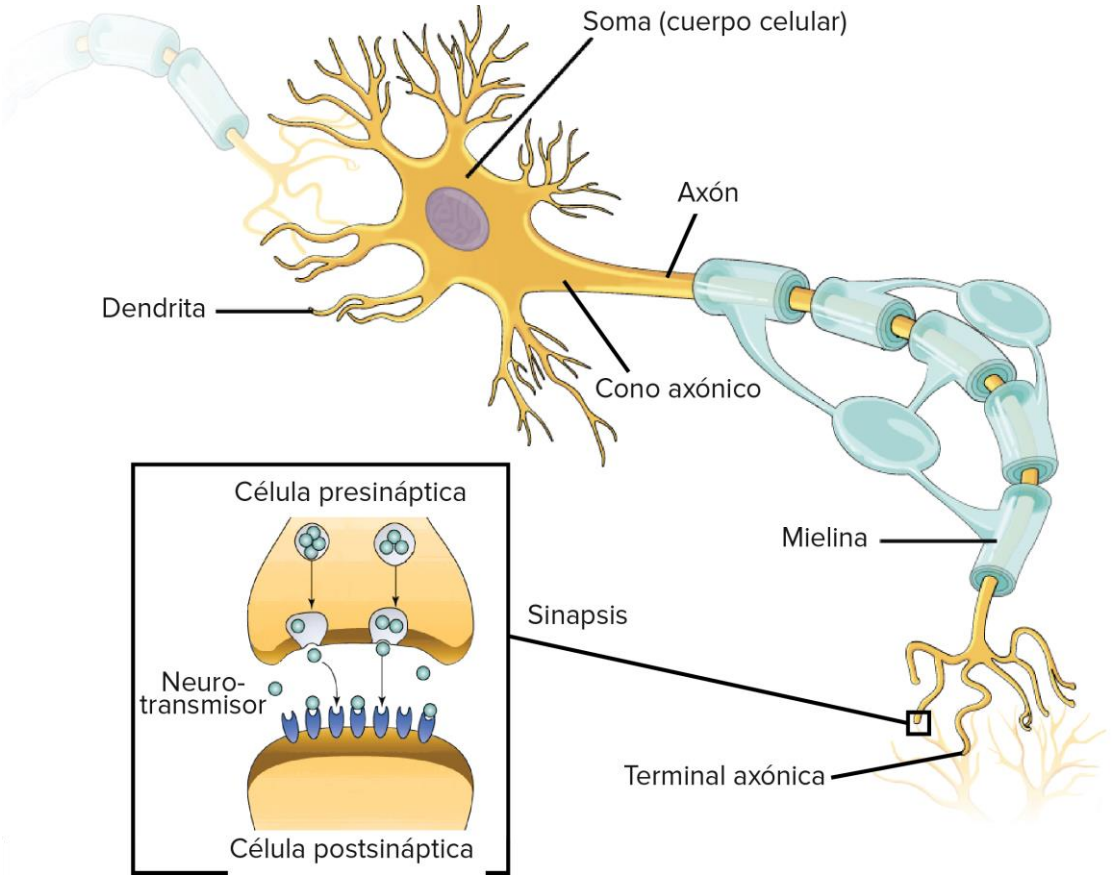


Figura 04: Estructura de la Neurona Humana o Biológica.
Fuente: (Ponce Cruz, 2010, pág. 194)

La célula llamada a transformar la información es la neurona biológica que acoge, por medio de sus dendritas, los impulsos o señales provenientes de neuronas que están a su alrededor y envía a través del axón, señales emitidas en su propio cuerpo. (Ponce Cruz, 2010, pág. 193)

Cada neurona se encuentra conformada por los siguientes elementos primordiales:

A. Dendritas

   Son tejidos de aspecto tubular, ramificados a semejanza de un árbol y habitualmente son las que admiten los estímulos. (Ponce Cruz, 2010, pág. 194)

B. Soma o Cuerpo de la Célula o Pericarion Central

   Contiene el ADN, los organelos y excepcionalmente admiten los estímulos. (Ponce Cruz, 2010, pág. 194)

C. Axón o Cilindro Eje

   Es un tejido de aspecto tubular más grande que las dendritas puede emplearse como conector o como transmisor; en su parte final se ramifica y cada una de ellas termina

en una especie nuez llamada bulbo o terminal axónico. (Ponce Cruz, 2010, págs. 194-195)

D. Sinapsis

Es la zona ubicada entre el bulbo de una neurona (etapa pre sináptica) y principalmente la dendrita (etapa pos sináptica) de la neurona siguiente. (Ponce Cruz, 2010, pág. 195)

La capacidad de comunicarse es una cualidad fundamental de la neurona. Las señales de entrada son percibidas por las dendritas y el soma; el soma las armoniza, las une y envía señales de salida; dichas señales son: eléctricas y químicas. La señal eléctrica, es un impulso eléctrico generado por la neurona y transita por todo el largo del Cilindro Eje, mientras la señal química es la señal que, viaja en la sinapsis entre una neurona y la siguiente. (Ponce Cruz, 2010, pág. 193)

Para construir una analogía de la actividad sináptica con las RNA debemos suponer que, las señales percibidas por la sinapsis de cada neurona son los ingresos de datos a la RNA; las cuales son simplificadas (compensadas o atenuadas) por medio del parámetro denominado peso, el cual está relacionado con la sinapsis. Cada señal de ingreso podría ocasionar que la neurona sea estimulada (sinapsis excitatoria o sinapsis con peso positivo) o sea inhibida (sinapsis inhibitoria o sinapsis con peso negativo). (Ponce Cruz, 2010, págs. 193-194)

Tabla 03: Analogía entre Neurona Biológica y Artificial.

Neurona Biológica	**Neurona Artificial**
Dendrita	Entrada
Soma	Neurona
Sinapsis	Peso / Conexión
Axón	Salida

Fuente: (Ponce Cruz, 2010). Elaboración: Propia, 2019

Es un panorama donde la neurona solo tiene dos únicas opciones y tiene que elegir obligatoriamente entre esas, es decir se activara o no se activara. Si la ponderación en la suma de los ingresos de datos para una determinada neurona da como resultado un valor mayor o igual que el valor de su umbral, entonces, como consecuencia se produce una salida (se activa la neurona). La actividad del sistema nervioso afecta directamente el

desenvolvimiento de la emisión de señales. Situaciones como la fatiga, ausencia de oxígeno y la existencia de anestésicos, entre otros afectan directamente a las sinapsis. La habilidad de la neurona al poder ajustar las señales se considera un medio de aprendizaje. (Ponce Cruz, 2010, pág. 194)

En la etapa pos sináptica de una neurona, la dendrita puede recibir cientos de sinapsis (impulso pos sináptico o PPS). Para conocer el resultado se realiza una suma algebraica de los impulsos pos sinápticos inhibitorios (PPSI) con los impulsos pos sináptico excitadores (PPSE). (Ponce Cruz, 2010, pág. 196)

2.3.5.2. Neurona Artificial.

Más adecuadamente conocido como Red Neuronal Artificial (RNA), también es llamado Red Neural o Inducción Neuronal.

Las RNA se encuentran explicadas a modo de una estructura similar al sistema nervioso de los seres vivos, constituidos en base a una gran cantidad de unidades de procesamiento ensamblados, cada uno de ellos con un determinado peso. Cada unidad se denomina neurona, la cual recibe señales de las otras unidades y envía una salida escalar simple, la cual está sujeta a la información que se encuentre disponible en dicha unidad, almacenada de forma doméstica o que proviene de las uniones y sus respectivos pesos. La cantidad de conexiones determina la realización de funciones complejas o simples. (Ponce Cruz, 2010, pág. 198)

Puede la RNA poseer cualquier estructura, pero, las capas están ubicadas en un determinado lugar de acuerdo a la función que realicen en dicha estructura. La información de entrada es considerada como la capa primera, y la información de salida u output se considera como la última. Las capas ocultas o internas son aquellas donde no se visualiza información de entrada o de salida. Al no efectuar ningún tipo de procesamiento la información de entrada no se considera como una capa. (Ponce Cruz, 2010, pág. 199)

Cada neurona (o unidad de proceso de la RNA) tiene una simple y única labor: recoger la información de otras neuronas o de fuentes externas, procesar dicha información y emitir una salida que se despliegue a otras neuronas. (Ponce Cruz, 2010, pág. 199)

Tabla 04: Analogía de funcionamiento entre Neurona Biológica y Artificial.

Neurona Biológica	Neurona Artificial
Conexiones sinápticas	Conexiones ponderadas Potencial de la neurona
Efectividad de las sinapsis	Peso de las conexiones
Efecto excitador o inhibidor de una conexión	Signo del peso de una conexión
Efecto combinado de las sinapsis	Función de propagación o de red
Activación *origina* Tasa de Disparo	Función de activación *origina* Salida

Fuente: (Ponce Cruz, 2010). Elaboración: Propia, 2019

Cada RNA tiene los siguientes componentes:

A. Neuronas (conjuntos de procesadores o unidades de procesamiento)

   Los enlaces de conexión (colección de sinapsis), cada uno caracterizado por un peso o fuerza propios. Concretamente, una señal $\mathbf{x_j}$ para la entrada de la sinapsis **j** que llega a la neurona **k** se multiplica por el peso sináptico $\boldsymbol{W_{kj}}$. Hay que tener cuidado en la forma como se detallan los subíndices del peso sináptico $\boldsymbol{W_{kj}}$. El primer subíndice hace referencia a la neurona, y el segundo subíndice indica la entrada de la sinapsis a la que corresponde el peso. A diferencia del peso de una sinapsis en el cerebro, el peso sináptico de una neurona artificial puede estar en un rango que incluye valores negativos y positivos. (Haykin, 2009, pág. 10)

B. Conexiones entre neuronas

   La suma ponderada de las señales de entrada, cuya ponderación depende de las fortalezas sinápticas de la neurona, componen una combinación lineal. (Haykin, 2009, pág. 10)

C. Función de Activación o Salida para cada neurona

   Tiene como finalidad limitar el espacio de respuesta. También se le conoce como función de aplastamiento, en razón de que aplasta (limita) el rango de amplitud permitido de la señal de salida a algún valor finito. Comúnmente, dicho rango de amplitud normalizado se escribe como el intervalo cerrado [0,1], o, alternativamente, [-1,1]. (Haykin, 2009, págs. 10-11)

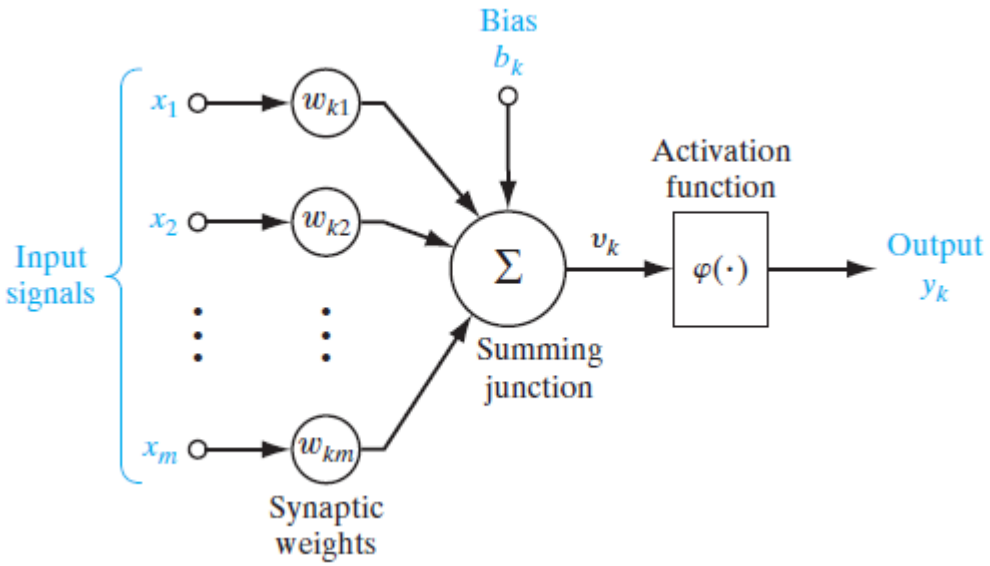


Figura 05: Modelo no lineal de una neurona, etiquetada como ***k***.
Fuente: (Haykin, 2009, pág. 11)

El modelo neuronal de la figura anterior también incluye un sesgo aplicado externamente, indicado por $\mathbf{b_k}$, que posee la habilidad de aumentar o disminuir la entrada neta de la función de activación, en función de si el valor es positivo o negativo. (Haykin, 2009, pág. 11)

En términos matemáticos, podemos describir la neurona ***k*** representada en la figura anterior, escribiendo el par de ecuaciones:

$$v_k = \sum_{j=1}^{m} w_{kj} x_j$$

$$y_k = \varphi(v_k + b_k)$$

donde:

$\mathbf{x_1, x_2, \ldots, x_m}$ son las señales de entrada;

$\mathbf{w_{k1}, w_{k2}, \ldots, w_{km}}$ son los respectivos pesos sinápticos de la neurona ***k***;

$\boldsymbol{v_k}$ es la salida de la combinación lineal debido a las señales de entrada;

$\mathbf{b_k}$ es el sesgo o -bias;

$\boldsymbol{\varphi}$ es la función de activación;

$\mathbf{y_k}$ es la señal de salida de la neurona. (Haykin, 2009, pág. 11)

Se denomina excitación si el peso es positivo, de lo contrario se denomina inhibición. (Ponce Cruz, 2010, pág. 199)

2.3.5.3. Funciones de activación.

Entre las más utilizadas encontramos a:

A. Función de activación escalón

Relacionado con neuronas binarias, tenemos que tener en cuenta que, si en una determinada neurona la sumatoria de sus datos de ingreso da como resultado un valor igual o mayor que su umbral, entonces, el resultado de la función será 1; lo inverso nos otorgara como resultado 0 (o -1). (Ponce Cruz, 2010, pág. 200)

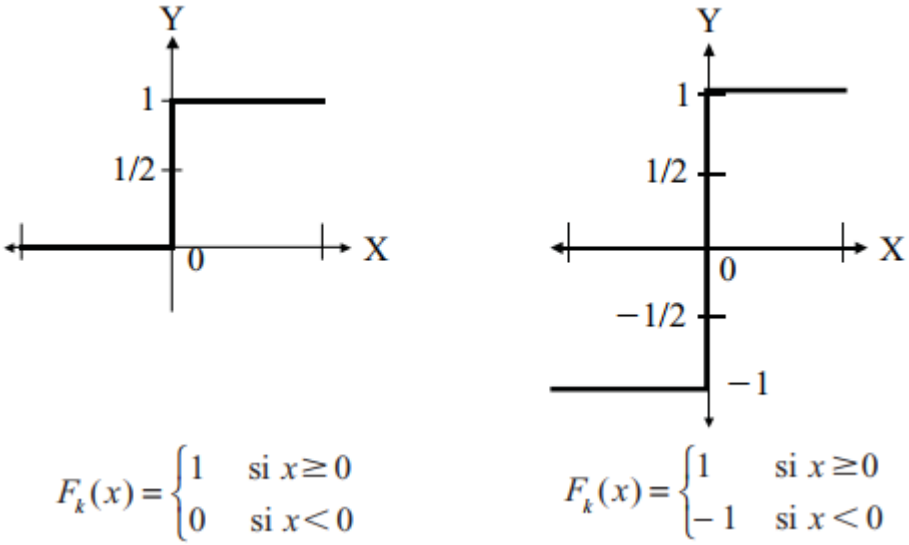


Figura 06: Función de activación escalón.
Fuente: (Ponce Cruz, 2010, pág. 200)

B. Función lineal o identidad y la función mixta

El comportamiento de la función identidad se encuentra determinada por el enunciado siguiente:

$$F_k(u) = u$$

Para la función mixta, hay que considerar que, cuando en una determinada neurona la sumatoria de sus datos de ingreso da como resultado un valor menor al límite mínimo, el cual ha sido definido con anterioridad en conjunto con el límite máximo, entonces, el resultado de la función será 0 (o -1); si la sumatoria es igual o mayor al límite máximo, entonces, el resultado de la función será 1, caso contrario, el resultado de la función estará basado en una formula lineal que incluye dicha sumatoria. (Ponce Cruz, 2010, pág. 200)

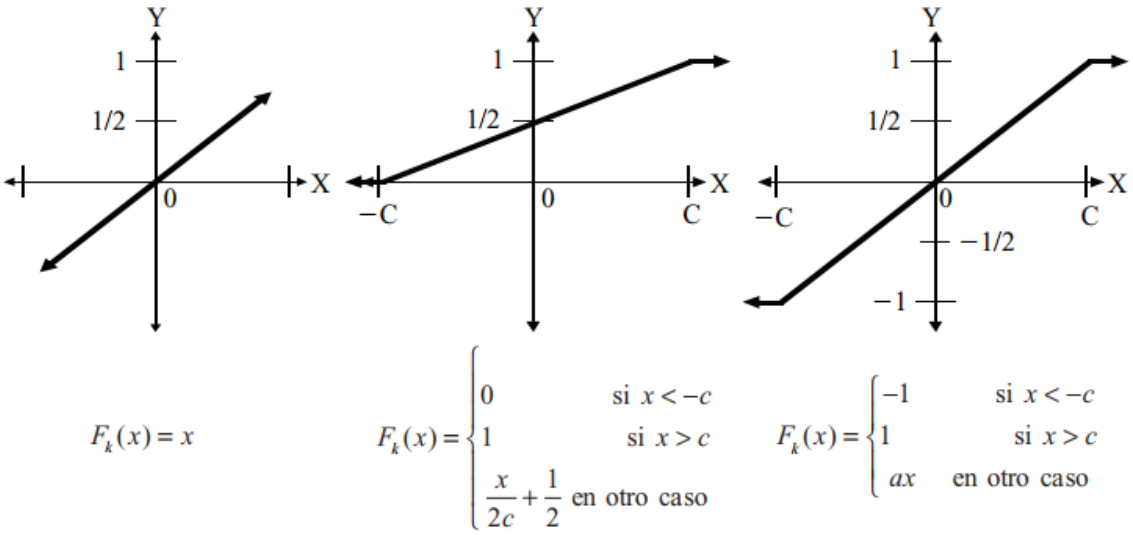


Figura 07: Función lineal y función mixta.
Fuente: (Ponce Cruz, 2010, pág. 200)

C. Función de activación tangente hiperbólica

Se emplea cuando se muestran suaves diferencias en los valores positivos y negativos de la señal a clasificar. Es de las funciones más utilizadas para los entrenamientos supervisados, como retropropagación del error. Hay que tener precaución al emplearla previo a la saturación entre los umbrales positivos y negativos, de lo contrario generara mayoritariamente valores de 1 y -1. (Ponce Cruz, 2010, pág. 201)

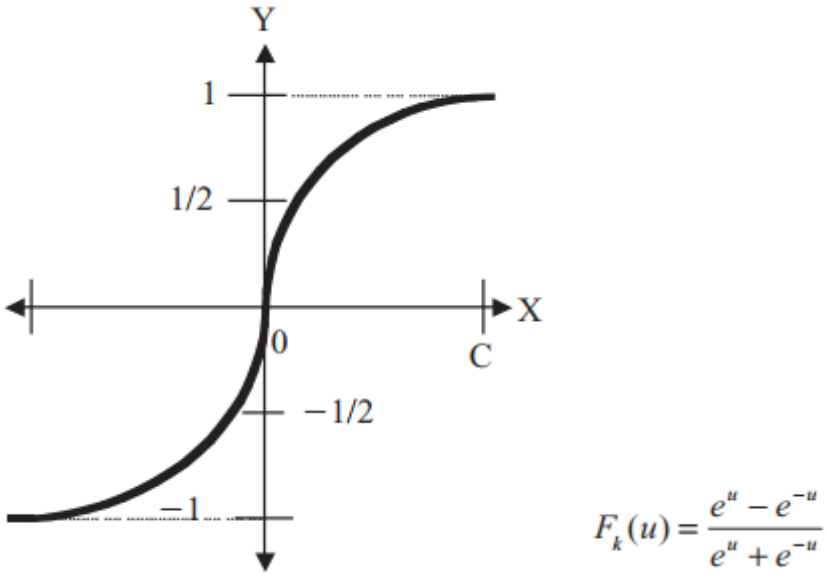


Figura 08: Función tangente hiperbólica.
Fuente: (Ponce Cruz, 2010, pág. 201)

D. Función de activación sigmoidal

La función otorga una cuantía cercana a cualquiera de los extremos de su escala, la cual se encuentra entre 0 y 1, lo que confiere como consecuencia que, en la mayor cantidad de ocurrencias, el resultado de la función se encuentra cercano a los límites inferior o superior de la curva. Por ejemplo, si tenemos una pendiente empinada, entonces los resultados de la función serán similares a los arrojados por la función escalón. Pero es importante indicar que, la derivada de esta función, en números altos enteros, es positiva y próxima a cero, adicionalmente su mayor cuantía se produce en el momento en que **x = 0**. (Ponce Cruz, 2010, pág. 201)

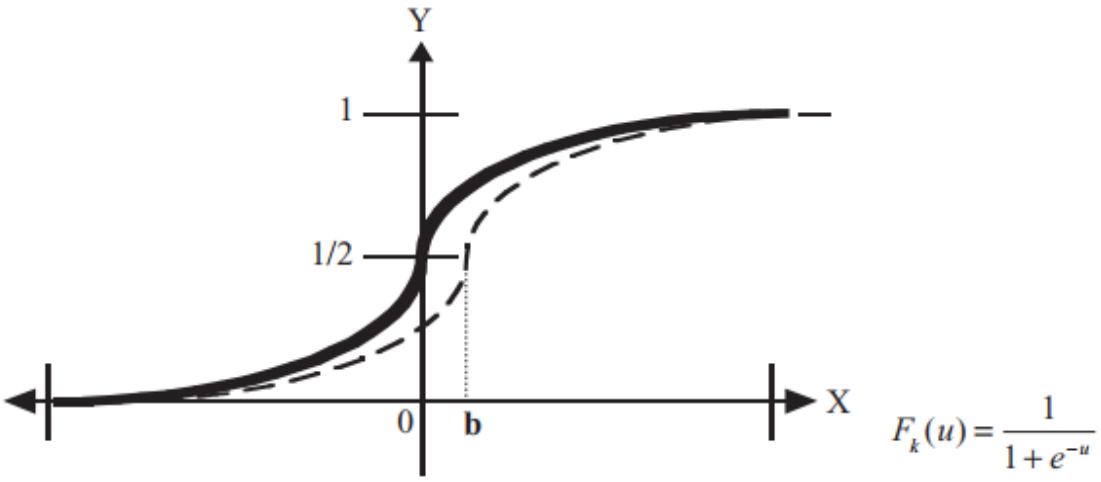


Figura 09: Función sigmoidal.
Fuente: (Ponce Cruz, 2010, pág. 201)

E. Función de activación de Gauss

Si no se desea utilizar la función sigmoidal, se puede utilizarse este tipo de función con una única capa de neuronas en los niveles ocultos. (Ponce Cruz, 2010, pág. 202)

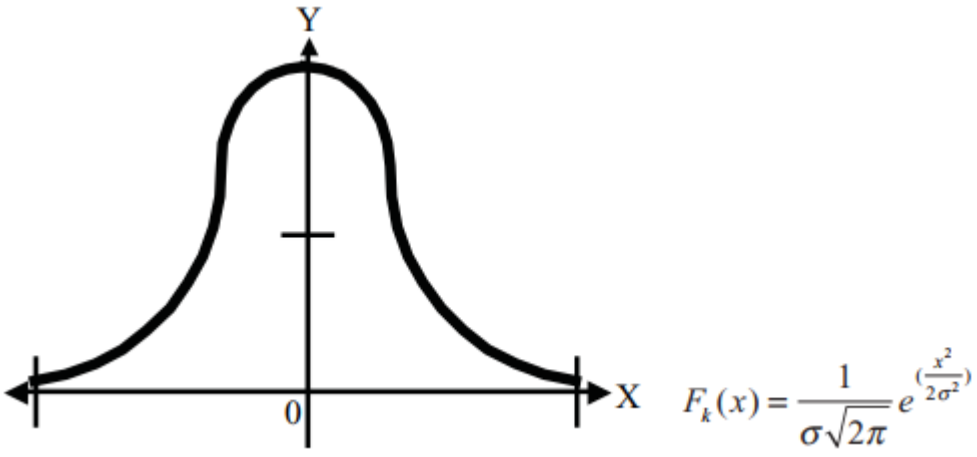


Figura 10: Función gaussiana.
Fuente: (Ponce Cruz, 2010, pág. 202)

2.3.5.4. Topologías de las RNA.

Es conocido también como *Network Architectures* o Conexión entre Neuronas.

Las neuronas son los componentes principales de una RNA que se conectan entre ellas por medio de sus uniones para procesar y compartir información. En la siguiente figura se observa un croquis simple de una RNA, que presenta una estructura de 3 capas y está relacionada con la topología denominada *feed-forward* debido a que la información fluye hacia delante. (Ponce Cruz, 2010, pág. 202)

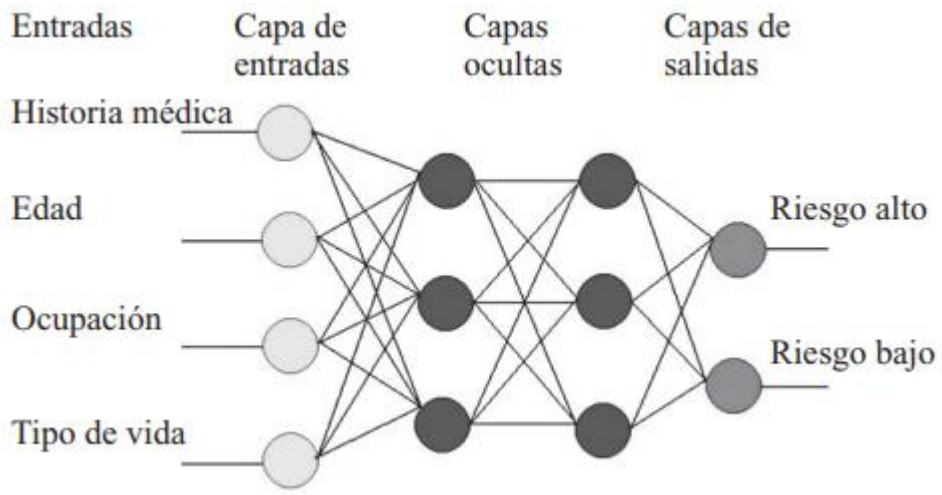


Figura 11: Diagrama del esquema básico de una Red Neuronal Artificial (RNA).
Fuente: (Ponce Cruz, 2010, pág. 202)

El modelo neuronal y la arquitectura de la red determinan cómo la arquitectura va a convertir la información de los *inputs* en información de *outputs*. (Ponce Cruz, 2010, pág. 203)

Las condiciones en las cuales se estructuran las neuronas de una RNA están estrechamente ligado al algoritmo de aprendizaje utilizado en el entrenamiento de ella. Por lo tanto, podemos hablar de algoritmos de aprendizaje (reglas) utilizados en el diseño de redes neuronales artificiales como estructurados. En general, podemos identificar tres clases fundamentalmente diferentes de arquitecturas de red (estructuras): (Haykin, 2009, pág. 21)

A. Redes *feedforward* de una sola capa (*Single-Layer Feedforward Networks*)

   En una RNA estratificada, las neuronas se organizan formando niveles o capas. Para el croquis más básico de una red estratificada, deberíamos tener un nivel de entrada que se proyecta directamente en el nivel de salida, pero no al revés. Dicho de otra

manera, es una RNA de tipo *feedforward* rigurosamente y se designa como red de una capa, y cuyo significado de "capa única" hace referencia a la capa de salida. No debemos contar el nivel de entrada porque allí no se ejecuta ningún cálculo. (Haykin, 2009, pág. 21)

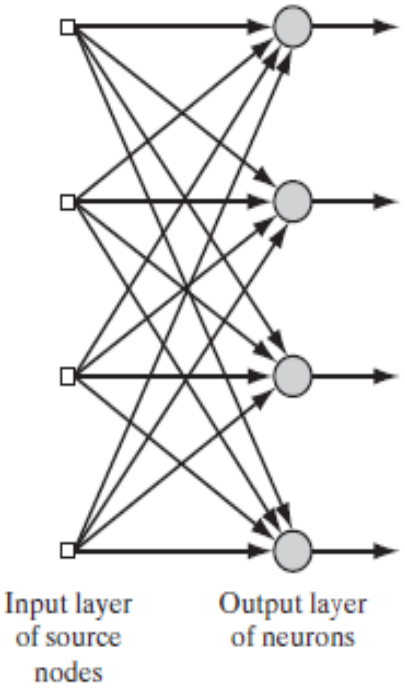


Figura 12: Red *feedforward* con 1 capa de neuronas, 4 neuronas tanto en la capa de entrada como en la capa de salida.
Fuente: (Haykin, 2009, pág. 21)

B. Redes *feedforward* de múltiples capas (*Multilayer Feedforward Networks*)

Se distingue por estar representado por una o más capas ocultas, y los nodos de cálculo denominados como neuronas ocultas o unidades ocultas; el término "oculto" se refiere al hecho de que esta parte de la red neuronal artificial no se ve directamente desde la entrada o la salida de la red. Las unidades ocultas tienen la obligación de meterse entre la entrada externa y la salida de red de alguna manera útil. En un sentido bastante flexible, la red adquiere una perspectiva global a pesar de su conectividad local, debido al conjunto adicional de conexiones sinápticas y la dimensión adicional de las interacciones neuronales. (Haykin, 2009, pág. 22)

Las neuronas o nodos en la capa de entrada, proporcionan los componentes para el vector de entrada (patrones de activación), dicho vector conforma la señal de entrada que se aplica a las neuronas (nodos de cálculo) de la capa segunda (en otras palabras, la capa oculta primera). Al producirse las señales de salida de esta capa se utilizaran como entradas en la tercera capa, y así sucesivamente. En general, cada

capa de neuronas recibe de entradas únicamente las señales de salida de la anterior capa. El conjunto de señales de salida de las neuronas de la última capa significa la respuesta de la RNA a los componentes del vector de entrada (capa primera). La red *feedforward* con **m** nodos de origen, $\mathbf{h_1}$ neuronas para la primera capa oculta, $\mathbf{h_2}$ neuronas en la segunda capa escondida y **q** neuronas para la capa de respuesta (salida), se conoce como una red $\mathbf{m\text{-}h_1\text{-}h_2\text{-}q}$. Como ejemplo, la figura siguiente se conoce como una red 10-4-2 porque tiene 10 nodos de origen, 4 neuronas ocultas y 2 neuronas de salida. (Haykin, 2009, pág. 22)

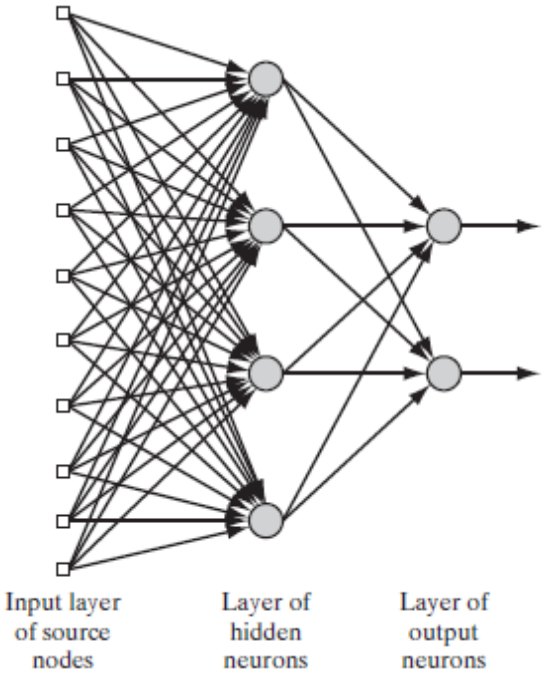


Figura 13: Red *feedforward* de múltiples capas totalmente acopladas, teniendo una capa de salida y una capa oculta.
Fuente: (Haykin, 2009, pág. 22)

Se dice que la RNA de la figura anterior está completamente conectada, porque, cada nodo de cada capa está interconectado a cada nodo de la capa delantera inmediata. Sin embargo, si faltan algunos de los enlaces de comunicación (conexiones sinápticas) de la red, decimos que la red está parcialmente conectada. (Haykin, 2009, pág. 23)

C. Redes recurrentes (*Recurrent Networks*)

La RNA recurrente se diferencia de la RNA predictiva porque posee por lo menos un circuito de retroalimentación. Como ejemplo tenemos la RNA recurrente que consiste en una sola capa de neuronas donde cada neurona entrega su señal de salida a las entradas del resto de neuronas, como se ilustra en la siguiente figura. En dicha

estructura, no hay bucles de auto-retroalimentación en la red, adicionalmente tampoco tiene neuronas ocultas; *self-feedback* describe un contexto donde la salida de la neurona alimenta su propia entrada. (Haykin, 2009, pág. 23)

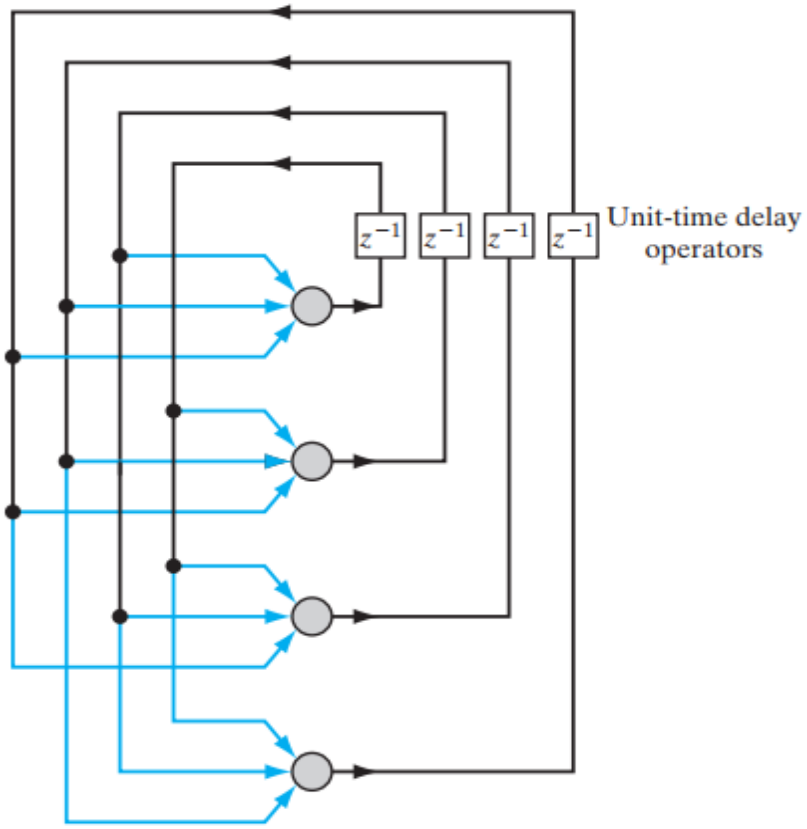


Figura 14: Red recurrente sin bucles de auto-retroalimentación y sin ninguna neurona oculta.
Fuente: (Haykin, 2009, pág. 23)

En la figura siguiente ilustramos otra clase de redes recurrentes con neuronas ocultas. Las conexiones de retroalimentación que se muestran se originan en las neuronas de salida y en las escondidas. (Haykin, 2009, pág. 23)

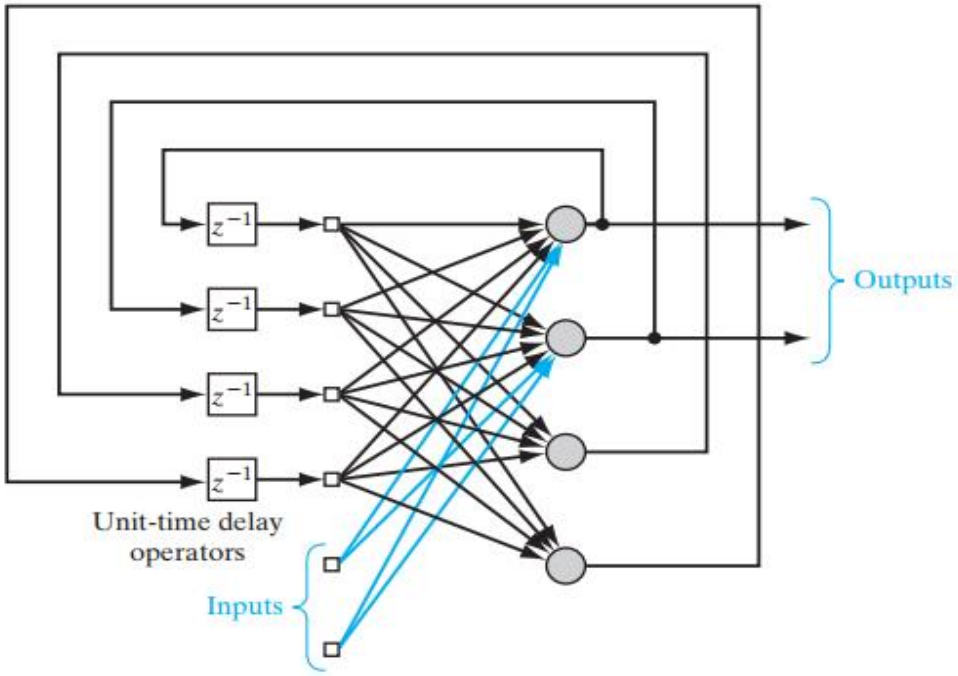


Figura 15: Red recurrente con bucles de auto-retroalimentación y con una capa de neuronas ocultas.
Fuente: (Haykin, 2009, pág. 24)

La disposición de los bucles de retroalimentación, en cualquiera de las 2 figuras anteriores, tiene un profundo impacto en la capacidad de aprendizaje de la red y en su rendimiento. Además, los bucles de retroalimentación implican el uso de ramas particulares compuestas de elementos de retraso en la unidad de tiempo (denotados por z-1), que dan como resultado una conducta dinámico no lineal, suponiendo que en la red neuronal artificial hay presencia de unidades no lineales. (Haykin, 2009, pág. 23)

2.3.5.5. Entrenamiento de las RNA.

Es el mecanismo mediante el cual se configura una RNA con la finalidad que, los datos de ingreso originen los resultados esperados por intermedio del robustecimiento de sus uniones. Un modo de implementar lo anteriormente descrito es establecer unos pesos iniciales definidos anteriormente; otra forma es utilizar algoritmos de retroalimentación y padrones de aprendizaje con la finalidad de descubrir los pesos que mejor se ajusten. (Ponce Cruz, 2010, pág. 203)

A. Aprendizaje Supervisado

   Requiere la disponibilidad de un objetivo o respuesta deseada para la realización de un mapeo específico de entrada-salida para minimizar una función de costo de interés. (Haykin, 2009, pág. 45)

B. Aprendizaje No Supervisado o Aprendizaje Sin Supervisión

   El aprendizaje de un mapeo de entrada-salida se efectúa por medio de una incesante interacción con el ambiente para reducir un indicador de rendimiento. (Haykin, 2009, pág. 36)

C. Aprendizaje Reforzado o con Fortalecimiento

   No hay un instructor o crítico externo que supervise el proceso de aprendizaje. Más bien, se prevén medidas independientes de la calidad para la representación de la red, así como que los parámetros libres de la RNA se encuentren optimizados con respecto a esa medida. (Haykin, 2009, pág. 37)

El aprendizaje supervisado se basa en la disponibilidad de una muestra de entrenamiento de ejemplos etiquetados, y cada uno de ellos está conformado por una señal de ingreso (estímulo) y su correspondiente objetivo (respuesta). En la práctica, encontramos que la recopilación de ejemplos etiquetados es una tarea costosa y que requiere mucho tiempo, especialmente cuando se trata de problemas de aprendizaje a gran escala; típicamente, por lo tanto, encontramos que los ejemplos etiquetados son escasos. Por otro lado, el aprendizaje no supervisado se basa únicamente en ejemplos no etiquetados, que consisten simplemente en un conjunto de señales de entrada o estímulos, para los cuales generalmente hay un suministro abundante. A la luz de estas realidades, existe un gran interés en otra categoría de aprendizaje: el aprendizaje semisupervisado, que emplea una muestra de capacitación que consta de ejemplos etiquetados y no etiquetados. El desafío en el aprendizaje semisupervisado, es diseñar un sistema de aprendizaje que tenga una escala razonablemente buena para que su implementación sea prácticamente factible cuando se trate de problemas de clasificación de patrones a gran escala. (Haykin, 2009, pág. 45)

El aprendizaje de refuerzo se encuentra definido entre los aprendizajes supervisado y no supervisado. Funciona mediante interacciones continuas entre un sistema de aprendizaje (agente) y el entorno. El sistema de aprendizaje realiza una acción y aprende de la respuesta del entorno a esa acción. En efecto, el rol del docente en el aprendizaje supervisado es reemplazado por un crítico, por ejemplo, que se integra en la maquinaria de aprendizaje escala. (Haykin, 2009, págs. 45-46)

2.3.6. CRISP-DM.

En 1996 fue establecida la metodología denominada *Cross Industry Standard Process for Data Mining* (CRISP-DM), y en la actualidad, es una de las metodologías más usadas debido a que provee un enfoque de mayor profundidad en relación a los objetivos que se desean alcanzar en un determinado proyecto. (Rodriguez Pacheco, 2015, pág. 58)

La metodología nos brinda una secuencia de seis etapas, cada una de ellas con relaciones de interdependencia con las otras y con su propia relación de tareas que se encuentran representadas en cuatro categorías escalonadas de abstracción, las cuales abarcan desde un nivel genérico a un nivel de detalle. (Rodriguez Pacheco, 2015, pág. 58)

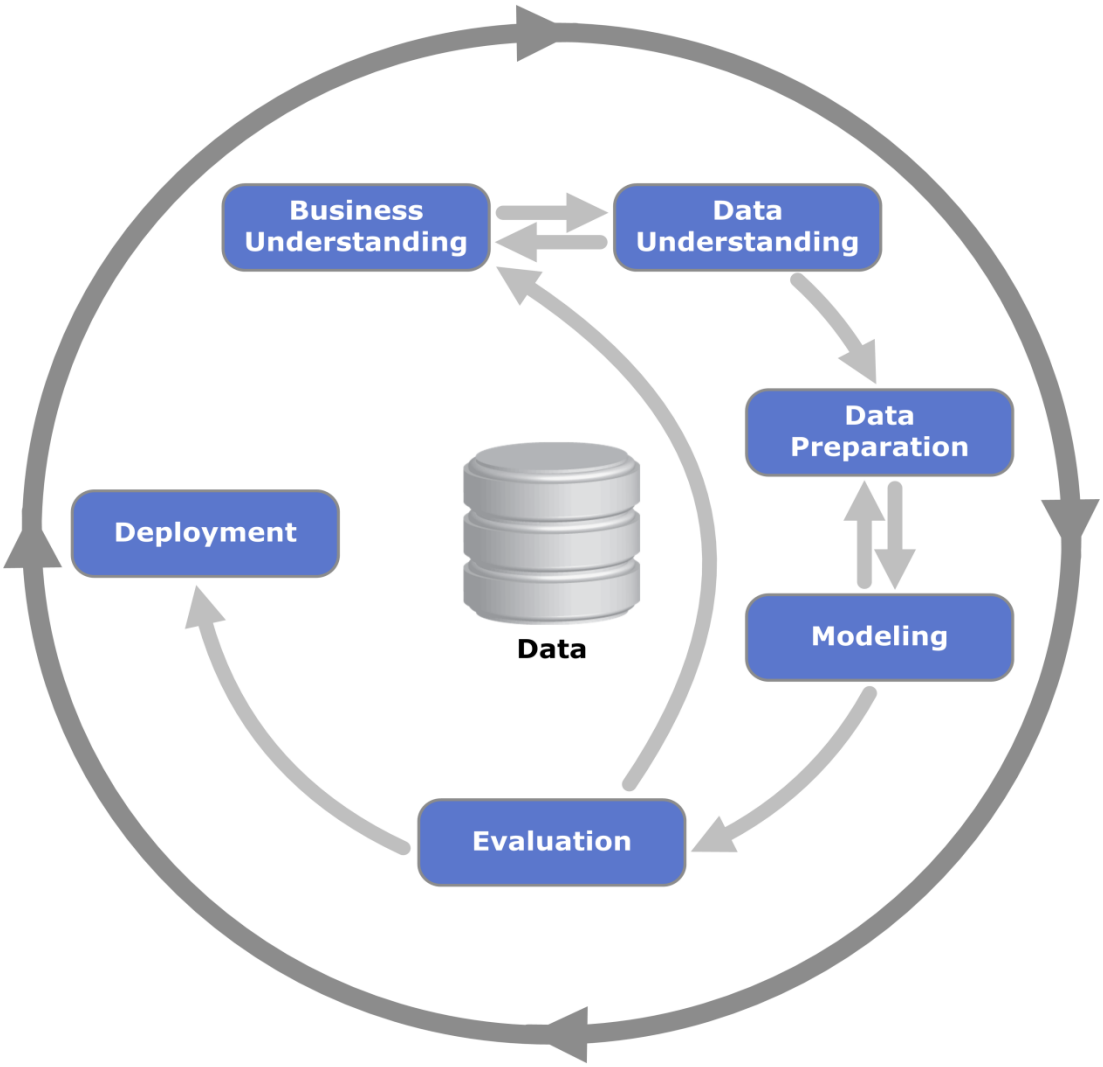


Figura 16: Secuencia de la metodología CRISP-DM.
Fuente: (Rodriguez Pacheco, 2015, pág. 59)

A continuación se describirá cada etapa:

A. Comprensión del Negocio (*Business understanding*)

   Se busca entender cómo funciona la empresa, mediante el conocimiento de sus objetivos; conocer cuál es el problema a resolver, mediante un análisis pormenorizado; determinar una posible solución y presentar un plan de trabajo. (Rodriguez Pacheco, 2015, pág. 58)

B. Comprensión de los datos (*Data understanding*)

   Se realiza la recopilación de los datos que estén vinculados al problema que se pretende resolver; se les investiga buscando conocerlos, describirlos, valorar su calidad, asegurar la consistencia de su información e identificar las relaciones entre ellos. (Rodriguez Pacheco, 2015, pág. 58)

C. Preparación de los datos (*Data preparation*)

   Se construye el *dataset* con la finalidad de poder solucionar el problema planteado; para ello se debe seleccionar a los datos después de haber sido transformados, limpiados, estructurados (derivación de datos) y unificados en un proceso con un alto nivel de calidad. (Rodriguez Pacheco, 2015, pág. 58)

D. Modelado (*Modeling*)

   Se seleccionan las técnicas de modelado que serán aplicadas al *dataset*; cada una de ellas será calibrada, aplicada, evaluada y reajustada; posteriormente se evaluaran todas las técnicas de modelado bajo un mismo criterio técnico, el cual debe haber sido seleccionado con anterioridad. (Rodriguez Pacheco, 2015, pág. 58)

E. Evaluación (*Evaluation*)

   Se evalúa el modelo desde la óptica de poder encontrar una respuesta satisfactoria al problema planteado; asimismo se revisa el procedimiento en la construcción y selección del modelo y si se encuentra alineado con la solución del problema planteado. (Rodriguez Pacheco, 2015, pág. 58)

F. Despliegue (*Deployment*)

   Se implementa el modelo mediante una estrategia que defina cuál será el plan de despliegue, cuál será el plan de monitoreo y mantenimiento, a quienes se informara los resultados obtenidos y como se difundirá. (Rodriguez Pacheco, 2015, pág. 58)

2.4. Definición de términos básicos

A continuación se presentan los términos básicos:

Dato

Si un pediatra desea saber el peso de un bebe y efectuamos la medición, la consecuencia, el valor de la medición efectuada es un dato, a mayor abundamiento Laudon & Laudon (2012) señalan: "los datos son flujos de elementos en bruto que representan los eventos que ocurren en las organizaciones o en el entorno físico antes de ordenarlos e interpretarlos en una forma que las personas puedan comprender y usar" (pág. 15).

Tabla

Una tabla consiste en una cuadrícula de columnas (atributo, característica, propiedad, cualidad, campo, factor, etiqueta, *feature*, *attribute*, *property*, *field*) y filas (registro, instancia, ejemplo, objeto, *instance*, *sample*, *record*) de datos. (Laudon & Laudon, 2012, pág. 214)

La fila representa a la entidad. La entidad puede ser: persona, lugar, cosa o evento y de la cual recolectamos y conservamos información. La columna es cada una de las características que define a la entidad. Tomemos como ejemplo a las columnas de la tabla CURSO: ID del alumno, Código del Curso y Notas Obtenidas; el contenido de cada una de ellas describe el rendimiento de cada alumno para un determinado curso. (Laudon & Laudon, 2012, pág. 210)

El conjunto de tablas relacionadas entre sí, tiene como finalidad formar una base de datos. La tabla CURSO del párrafo anterior se podría agrupar con las tablas: histórico de datos personales del alumno e histórico financiero del alumno, con las cuales podría crearse una base de datos de estudiantes. (Laudon & Laudon, 2012, pág. 210)

***Dataset* o Colección de Datos**

Es una tabla que guarda información histórica y actualizada, la cual podría ser potencialmente relevante para realizar: análisis para la búsqueda de patrones. Es donde se agrupa y normaliza la información proveniente de diferentes bases de

datos proveniente de sistemas diversos y/o de sitios Web. (Laudon & Laudon, 2012, pág. 222)

Información

Son los datos que se han sido transformados de una manera reveladora y útil para los analistas y que son el primer paso para obtener conocimiento. (Laudon & Laudon, 2012, pág. 15)

Patrón o Tendencia

También conocidos como patrones o patrón de comportamiento o patrones estructurales.

Es una secuencia de eventos (auditivos, gestuales, gráficos, etc.) los cuales se edifican bajo un determinado criterio, que podría ser: (Cadenas, 2015)

A. Patrones de repetición.
   Es donde los diferentes eventos se despliegan de forma periódica teniendo en cuenta su estructura.
B. Patrones de recurrencia.
   Es donde la frecuencia con que la cual se despliegan los eventos se modifica, por lo tanto se procura hallar el criterio de formación, en otras palabras, poder inferir el evento siguiente en base al patrón de los eventos preliminares.

Algoritmo

También es conocido como Técnica de computadora. Son aquellos que precisan con exactitud la secuencia (relación de instrucciones) a adoptar para provocar una solución automatizada, altamente estructurada y muy rápida, como consecuencia de la información contenida en el *dataset*, de la capacidad de los procesadores y del *software* utilizado en la resolución de la labor. (Laudon & Laudon, 2012, pág. 461)

***Machine Learning* (ML)**

Laudon & Laudon (2012) señalan que es una tecnología relacionada con la Inteligencia Artificial (IA) que: "se enfoca en los algoritmos y métodos estadísticos que permiten a las computadoras 'aprender' al extraer reglas y patrones de conjuntos masivos de datos y realizar predicciones sobre el futuro" (pág. 438).

Aprendizaje Supervisado

Los algoritmos (técnicas o métodos) de aprendizaje supervisado intentan construir un patrón a partir de datos conocidos previamente, lo que nos permite inferir (predecir o clasificar) el valor de salida de los nuevos datos, conociendo solo sus características. (Garreta & Moncecchi, 2013, pág. 20)

En ocasiones, el aprendizaje supervisado también se denomina modelado predictivo. Los algoritmos primarios de modelado predictivo son la clasificación para variables objetivo categóricas o la regresión para variables objetivo continuas. (Abbott, 2014, pág. 5)

Aprendizaje No Supervisado

También es a veces llamado modelado descriptivo, no tiene una variable objetivo. Las entradas se analizan y agrupadas en función de la proximidad de los valores de entrada entre sí. A cada grupo o clúster se le asigna una etiqueta para indicar a qué grupo pertenece un registro. En algunas aplicaciones, como en el análisis de clientes, el aprendizaje no supervisado es denominado segmentación debido a la función de los modelos (segmentar clientes en grupos). (Abbott, 2014, pág. 5)

Clasificación

Laudon & Laudon (2012) puntualizan que:

> La clasificación reconoce los patrones que describen el grupo al que pertenece un elemento, para lo cual se examinan los elementos existentes que hayan sido clasificados y se infiere un conjunto de reglas. Por ejemplo, las empresas como las compañías de tarjetas de crédito o las telefónicas se preocupan por la pérdida de clientes estables. La clasificación ayuda a descubrir las características de los clientes con probabilidades de dejar de serlo y puede proveer un modelo para ayudar a los gerentes a predecir quiénes son esos clientes, de modo que puedan idear campañas especiales para retenerlos. (pág. 225).

Los Modelos Lineales Generalizados, permiten que una variable de respuesta categórica (nominal o alguna transformación de la misma) pueda establecer vínculos con una agrupación de variables con características predictoras. Los modelos de

regresión de Poisson y logístico están circunscritos en los modelos lineales generalizados. (Han, Kamber, & Pei, 2012, págs. 599-600)

Boosting

Es un método de ensamble de árboles de clasificación. Construye un modelo de clasificación simple y se asignan pesos iguales a cada observación. Luego de esta primera predicción, les reduce el peso a los casos bien clasificados, mientras les aumenta el peso a los errores registrados. A continuación, elabora otro modelo de clasificación débil y las observaciones que fueron mal predichas tienen mayor énfasis en su siguiente clasificación, debido al aumento de su peso. El ciclo puede repetirse "n" veces. El resultado final se da a partir del promedio ponderado de la predicción de todos los modelos. El algoritmo está diseñado para que se componga de modelos de clasificación débiles y los más usados son los árboles de decisión. Por lo general, la precisión del *Boosting* es más alta que la precisión de un único árbol de decisión o del ensamble de árboles *Bagging*. (Abbott, 2014, págs. 316-317)

RNA

Laudon & Laudon (2012) en su libro sobres sistemas de información indican que:

> Las redes neuronales se utilizan para resolver problemas complejos y malentendidos, para los que se han recolectado grandes cantidades de datos. Buscan patrones y relaciones en cantidades masivas de datos cuyo análisis sería demasiado complicado y difícil para un humano. Las redes neurales descubren este conocimiento mediante el uso de *hardware* y *software* que se asemejan a los patrones de procesamiento del cerebro biológico o humano. Las redes neurales "aprenden" patrones de grandes cantidades de datos al escudriñar los datos, buscar relaciones, crear modelos y corregir una y otra vez los propios errores del modelo. (pág. 436).

Desaprobado

De acuerdo al Diccionario de la Real Academia Española (RAE), reprobar se circunscribe al hecho de no dar por bueno o suficiente a algo o a alguien; o de no tener las suficientes habilidades, competencias o conocimientos sobre un tema, asignatura o examen.

Rendimiento Académico

El rendimiento académico es alto o bajo, dependiendo si se alcanza una calificación superior o inferior con respecto a un determinado límite o frontera, el cual es definido por la institución educativa; por ejemplo, la frontera podría ser 11 o 12 si la escala de calificación es vigesimal.

2.5. Fundamentos teóricos que sustentan las hipótesis

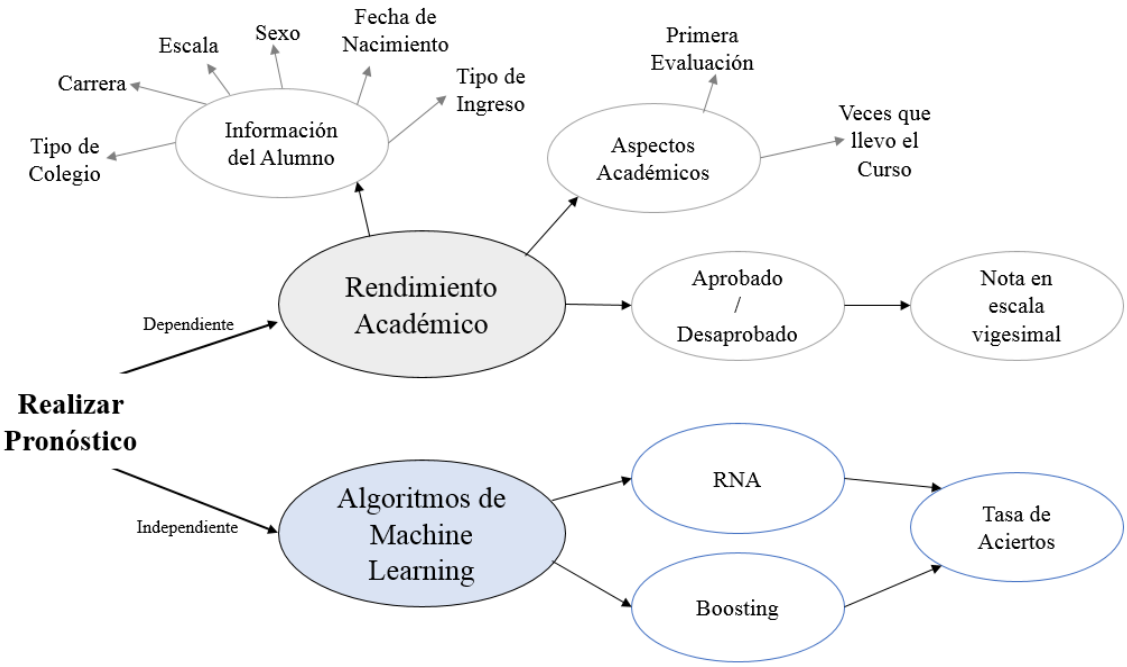


Figura 17: Diagrama de los fundamentos teóricos que sustentan las hipótesis.
Elaboración: Propia, 2019

2.6. Hipótesis

2.6.1. Hipótesis general.

Hemos formulado como hipótesis general la siguiente:

- Si se aplican los algoritmos de *Machine Learning*, entonces, se incrementará la eficiencia en la predicción de la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.

2.6.2. Hipótesis específicas.

Y como hipótesis específicas podríamos citar lo siguiente:

- Si se aplica el algoritmo de Redes Neuronales Artificiales (RNA), entonces se incrementará la eficiencia en la predicción de la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.

- Si se aplica el algoritmo *Boosting*, entonces se incrementará la eficiencia en la predicción de la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.

- Al comparar el nivel de precisión de ambos algoritmos se puede identificar el más óptimo para la problemática planteada.

2.7. Variables

Hernández, Fernández & Baptista (2014) en su libro sobre metodología indican que:

> (...) los diseños correlacionales-causales pueden limitarse a establecer relaciones entre variables sin precisar sentido de causalidad o pretender analizar relaciones causales. Cuando se limitan a relaciones no causales, se fundamentan en planteamientos e hipótesis correlacionales; del mismo modo, cuando buscan evaluar vinculaciones causales, se basan en planteamientos e hipótesis causales. (pág. 157).

Tabla 05: Variables Empleadas.

Tipo	**Variable**	**Definición Conceptual**	**Dimensiones**	**Indicadores**	**Instrumento**
Independiente	Algoritmos de *Machine Learning*	Programa de computación que calificará el rendimiento académico de un alumno	Algoritmos de RNA Algoritmos *Boosting*	Tasa de Aciertos	*Softwares* libres R y Python MS-Excel
Dependiente	Rendimiento Académico de los Alumnos en el Programa de Estudios Básicos de	Cantidad de alumnos con Rendimiento Académico aprobado y desaprobado	Rendimiento Académico en la Universidad (Nota por cada vez que ha llevado el	Nota de cada alumno en escala vigesimal, Número de veces que ha llevado el	Tablas de Datos proporcionadas por del Centro de Computo de la Universidad Ricardo Palma.

	la Universidad Ricardo Palma	curso) Nivel socio-económico Sexo Edad Permanencia en la Universidad Ricardo Palma	curso y Promedio ponderado. Escala de pago Masculino o Femenino Fecha de nacimiento Ciclo donde ha estudiado y número de aprobaciones	(*Dataset* preparado y elaborado por el tesista)

Elaboración: Propia, 2019

CAPÍTULO III: MARCO METODOLÓGICO

3.1. Tipo, método y diseño de la investigación

Según la naturaleza que abarca la información (tipo de datos) que se acopio para responder al problema, la investigación tiene un **enfoque de tipo Cuantitativo**, porque de acuerdo a lo definido por Hernández, Fernández & Baptista (2014) se empleó: "la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías" (pág. 4).

Por lo tanto, se buscó resolver un determinado problema, entonces el **tipo de investigación es aplicado.**

La presente investigación desarrolló un **método con alcance descriptivo y correlacional:**

- Descriptivo simple, porque permitió mostrar de forma evidente la ocurrencia de las características de una o más variables en la data de investigación (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2014, pág. 155)

- Correlacional, justificado según Hernández, Fernández & Baptista (2014) porque se buscó: "conocer la relación o grado de asociación que exista entre dos o más conceptos, categorías o variables en una muestra o contexto en particular. En ocasiones sólo se analiza la relación entre dos variables, pero con frecuencia se ubican en el estudio vínculos entre tres, cuatro o más variables" (pág. 93).

La presente investigación desarrolló un **diseño de tipo no experimental**, en concordancia con el siguiente concepto:

> En un estudio no experimental no se genera ninguna situación, sino que se observan situaciones ya existentes, no provocadas intencionalmente en la investigación por quien la realiza. En la investigación no experimental las variables independientes ocurren y no es posible manipularlas, no se tiene control directo sobre dichas variables ni se puede influir en ellas, porque ya se

produjeron, de la misma forma que las respectivas consecuencias. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2014, pág. 152).

3.2. Población y muestra

3.2.1. Población.

Se construyó con todas las notas de aquellos alumnos matriculados en los cursos del Programa de Estudios Básicos desde el ciclo 2015-1 hasta el ciclo 2019-0, obteniendo un total de 9,118 alumnos y 574,283 calificaciones.

3.2.2. Muestra.

Un mayor número de filas de información nos permitirá elaborar modelos estables. Específicamente cuando se desea realizar un modelo de clasificación, cada clase de la variable respuesta debe poseer una cantidad mínima de 1,000 registros, en cada uno de los conjuntos de datos donde se realizarán el entrenamiento y la evaluación del modelo. En líneas generales, se busca reducir el tamaño de la población para evitar los problemas en el tiempo de modelamiento, pero si el equipo de cómputo en conjunto con el *software* puede maniobrar el tamaño de toda la población, entonces no hay ninguna razón para reducirla. (Abbott, 2014, págs. 128-129)

Por lo tanto, para el pronóstico de rendimiento no se optó por escoger a un subgrupo de la población, por el contrario, se seleccionó a toda la población, por lo tanto, no hubo un diseño muestral.

Sin embargo, el diseño metodológico, para los modelos de *Machine Learning*, nos indica que, la población de estudio se debe dividir en 2 porciones: una denominada "train" que tiene como finalidad elaborar un modelo (70% de la población); y una para realizar las pruebas de performance del modelo obtenido (30% de la población) denominada "test".

3.3. Técnicas e instrumentos de recolección de datos

En trato directo con el Centro de Computo de la Universidad Ricardo Palma, se logró recabar directamente la información de acuerdo a las características que les solicitamos; la cual se encuentra conformada por los siguientes archivos de datos:

A. Archivo de notas

B. Archivo del tipo de evaluación

C. Archivo de planes curriculares

D. Archivo de alumnos

E. Archivo de carreras

F. Archivo de modalidad de ingreso

Las acciones que se realizaron para el procedimiento de recolección de datos fueron:

- Se solicitó el Archivo de notas, a partir del cual se obtuvo una relación de alumnos, dicha relación se entregó al Centro de Computo de la Universidad Ricardo Palma.

  Con el *software* Python se generó el archivo denominado "relacion.csv" que contiene los códigos de los alumnos mencionados en el párrafo anterior. Los comandos utilizados se encuentran en el Anexo 05 de la presente investigación.

- Se coordinó con el Centro de Cómputo para que nos entregaran la información socioeconómica correspondiente a los alumnos que se encontraban en la relación del punto anterior.

- De acuerdo al punto anterior se proporcionaron 5 archivos de datos adicionales.

A todos los archivos de datos se les practicó un análisis exploratorio y una depuración, con la finalidad de obtener un *dataset* con las mejores características.

Como se puede observar la técnica que se utilizó fue la revisión de las bases de datos y el instrumento fue el *dataset* obtenido a partir de cada una de las base de datos.

3.4. Descripción de procedimientos de análisis de datos

Los procedimientos de análisis fueron realizados en:

- El *software* R, dado que es un *software* libre y muy potente para el análisis estadístico y la generación de modelos.

- El *software* Python, por ser un *software* libre y muy versátil para la transformación de las estructuras de los archivos de datos.

- MS Excel, por la mejor visualización gráfica de la información y para la obtención de las tablas de equivalencias.

CAPÍTULO IV: RESULTADOS Y ANÁLISIS DE RESULTADOS

4.1. Resultados

4.1.1. Comprensión de los datos.

En esta etapa se realizó un análisis interno a cada uno de los archivos de datos que nos proporcionó el Centro de Computo de la Universidad Ricardo Palma.

4.1.1.1. Archivo de carreras.

A. Conjunto inicial de datos

   Denominado "Carrera.csv", el cual contiene las características de todas las carreras de pre-grado. Se encuentra conformado por 19 registros y las siguientes columnas:

   1. Código de la carrera
   2. Nombre de la carrera

B. Exploración de los datos

   Se utilizó MS Excel para revisar y explorar las carreras que se ofrecen en la Universidad Ricardo Palma.

C. Resultados Iniciales

   Se pudo observar que el Programa de Estudios Básicos, se presenta como una carrera con el código de identificación "500".

D. Limpieza de datos

   Se eliminó el registro cuyo código es igual a "500", porque no es una carrera.

E. Construcción de datos

   Se generó el nuevo archivo con 18 carreras o registros, denominado "carrera_a.csv".

F. Verificación de calidad de datos

El archivo de carreras, después del pre procesamiento y limpieza, contiene la siguiente información:

Tabla 06: Carreras en la Universidad Ricardo Palma.

Carrera	
Código	Descripción
11	Arquitectura
21	Biología
25	Medicina Humana
27	Medicina Veterinaria
31	Economía
32	Administración y Gerencia
33	Contabilidad y Finanzas
34	Administración de Negocios Globales
35	Turismo, Hotelería y Gastronomía
38	Marketing Global y Administración Comercial
41	Psicología
46	Derecho
51	Traducción e Interpretación
61	Ingeniería Civil
62	Ingeniería Electrónica
63	Ingeniería Industrial
66	Ingeniería Informática
68	Ingeniería Mecatrónica

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Con el *software* R se revisó las características internas del nuevo archivo, comprobándose que no presentaba ninguna inconsistencia. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

4.1.1.2. Archivo de modalidad de ingreso.

A. Conjunto inicial de datos

Contiene las características de la modalidad de ingreso a la universidad por parte de los alumnos. Se encuentra conformado por 21 registros y las siguientes columnas:

1. Código de la modalidad de ingreso

2. Descripción de la modalidad de ingreso

B. Exploración de los datos

Se utilizó MS Excel para revisar y explorar las diferentes modalidades de ingreso que ofrece la Universidad Ricardo Palma.

C. Resultados Iniciales

No se observó ninguna inconsistencia.

D. Limpieza de datos

No fue necesario realizar la limpieza de los datos.

E. Construcción de datos

Se conservó el archivo de modalidad de ingreso denominado “Ingreso.csv”.

F. Verificación de calidad de datos

El archivo de modalidad de ingreso contiene la siguiente información:

Tabla 07: Modalidad de ingreso en la Universidad Ricardo Palma.

Modalidad de Ingreso	
Código	Descripción
00	Alumno Libre
01	Bachillerato
02	Bachillerato Escolar Internacional
03	Beca 18 - Pronabec
04	Cepurp Ciclo Regular
05	Cepurp Aptitud Académica
06	Cepurp Escolares 5to de Secundaria
07	Cepurp Especial Escolares Ago-Ene
08	Cepurp Intensivo Enero-Marzo
09	Deportistas
10	Diplomáticos
11	Graduado y/o Titulado
12	Primeros Puestos
13	Examen de Aptitud Académica
14	Cobertura Aptitud Académica
15	Examen General de Admisión
16	Cobertura Examen General
17	Examen Promocional
18	Cobertura Examen Promocional

19	Traslado Externo
20	Cobertura Traslado Externo

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Con el *software* R se revisó las características internas del archivo original, comprobándose que no presentaba ninguna inconsistencia. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

4.1.1.3. Archivo de alumnos.

A. Conjunto inicial de datos

Denominado "Alumnos.csv", el cual contiene las características sociodemográficas, únicamente, de los alumnos que se encuentran en el archivo de notas. Se encuentra conformado por 9,118 registros y las siguientes columnas:

1. Código del alumno
2. Código de la carrera
3. Sexo
4. Fecha de nacimiento
5. Modalidad de ingreso
6. Escala de pago
7. Nombre del colegio de procedencia
8. Tipo del colegio de procedencia ('P' = Particular / 'E' = Estatal)

B. Exploración de los datos

Con el *software* R se revisó las características internas del archivo original, los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

MS Excel se utilizó para revisar y explorar los alumnos que han seguido los cursos del Programa de Estudios Básicos para el periodo comprendido desde el ciclo 2015-1 hasta el ciclo 2019-0.

Con respecto al sexo del alumnado, su distribución es como sigue:

Tabla 08: Distribución del alumnado según sexo.

Sexo		**Cantidad**
Código	Descripción	
M	Masculino	4,688
F	Femenino	4,430
		9,118

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

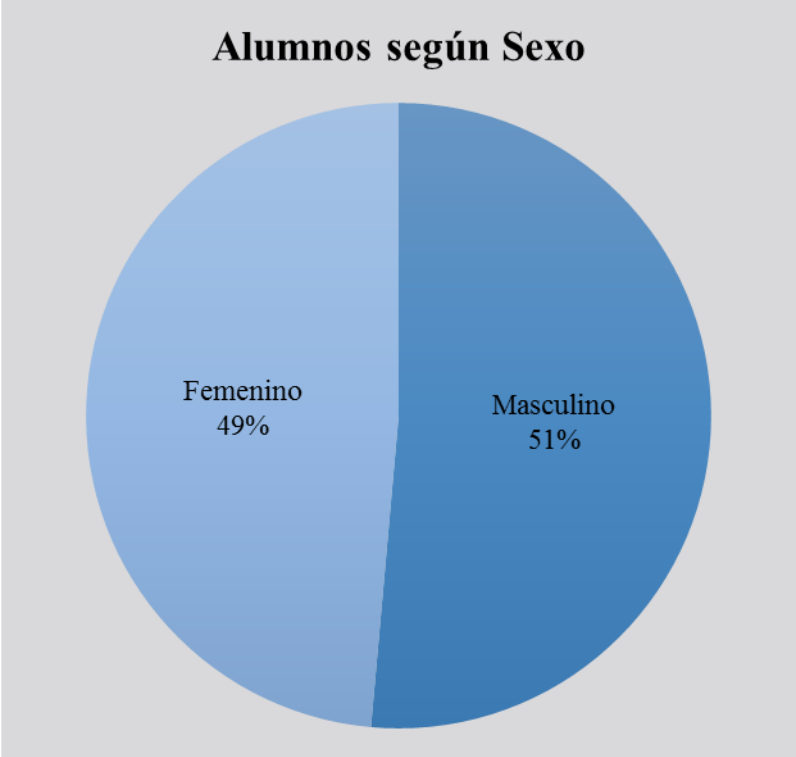


Figura 18: Distribución del alumnado según sexo.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se pudo observar, que el alumnado femenino es ligeramente superior al masculino.
Con respecto a la carrera de pre grado que el estudiante ha elegido, la distribución es la siguiente:

Tabla 09: Distribución del alumnado según carrera.

Carrera		**Cantidad**
Código	Descripción	
11	Arquitectura	1,343

21	Biología	196
25	Medicina Humana	1,026
27	Medicina Veterinaria	316
31	Economía	135
32	Administración y Gerencia	533
33	Contabilidad y Finanzas	248
34	Administración de Negocios Globales	512
35	Turismo, Hotelería y Gastronomía	167
38	Marketing Global y Administración Comercial	245
41	Psicología	571
46	Derecho	369
51	Traducción e Interpretación	681
61	Ingeniería Civil	1,276
62	Ingeniería Electrónica	133
63	Ingeniería Industrial	864
66	Ingeniería Informática	276
68	Ingeniería Mecatrónica	227
		9,118

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Figura 19: Distribución del alumnado según carrera.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se logró observar, que las carreras cuya cantidad de alumnos es mayor a 1,000 estudiantes son Arquitectura, Ingeniería Civil y Medicina Humana.

Con respecto a la escala de pago del alumno, su distribución se observa a continuación:

Tabla 10: Distribución del alumnado según escala de pago.

Escala de Pago			Cantidad
Código	Matricula	Cuota	
A23	300.00	1,350.00	5,414
A33	300.00	1,250.00	2,024
A13	300.00	2,100.00	882
A38	300.00	1,450.00	336
A28	300.00	1,550.00	275
A18	300.00	2,600.00	144
A99	-	-	43
			9,118

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

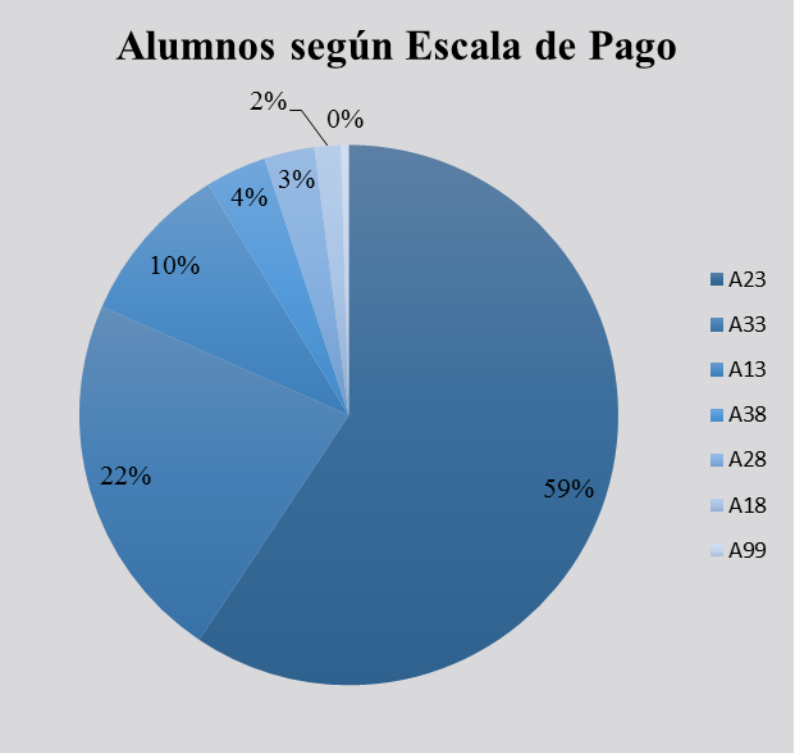


Figura 20: Distribución del alumnado según escala de pago.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede apreciar a cuarenta y tres (43) alumnos que tienen como escala de pago el código "A99", el cual es asignado a los estudiantes extranjeros y/o de intercambio internacional.

Con respecto a la modalidad de ingreso, la distribución del alumnado es la siguiente:

Tabla 11: Distribución del alumnado según modalidad de ingreso.

Modalidad de Ingreso		Cantidad
Código	Descripción	
00	Alumno Libre	43
01	Bachillerato	3
02	Bachillerato Escolar Internacional	8
03	Beca 18 - Pronabec	72
04	Cepurp Ciclo Regular	386
05	Cepurp Aptitud Académica	305
06	Cepurp Escolares 5to de Secundaria	83
07	Cepurp Especial Escolares Ago-Ene	26
08	Cepurp Intensivo Enero-Marzo	186
09	Deportistas	20
10	Diplomáticos	3
11	Graduado y/o Titulado	61
12	Primeros Puestos	56
13	Examen de Aptitud Académica	3,335
14	Cobertura Aptitud Académica	64
15	Examen General de Admisión	2,829
16	Cobertura Examen General	79
17	Examen Promocional	1,271
18	Cobertura Examen Promocional	47
19	Traslado Externo	240
20	Cobertura Traslado Externo	1
		9,118

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

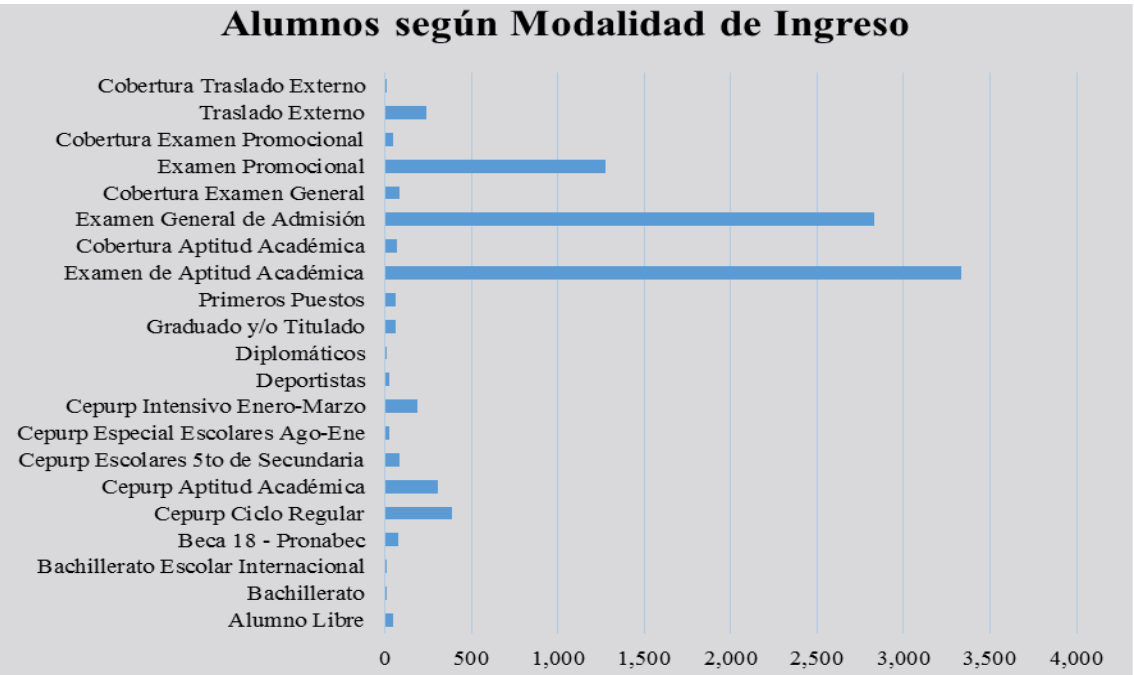


Figura 21: Distribución del alumnado según modalidad de ingreso.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

En la categoría denominada “Alumno Libre” (cuyo código es “00”) hay 43 participantes, los cuales son estudiantes de intercambio internacional que estuvieron matriculados en los cursos, siempre y cuando que, el curso se encuentre ubicado en el semestre equivalente al de su universidad de origen.

Con respecto al colegio de procedencia, se precisó que su distribución:

Tabla 12: Distribución del alumnado según tipo de colegio de procedencia.

Tipo de Colegio		**Cantidad**
Código	Descripción	
E	Estatal	1,961
P	Particular	7,114
N	Null	43
		9,118

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

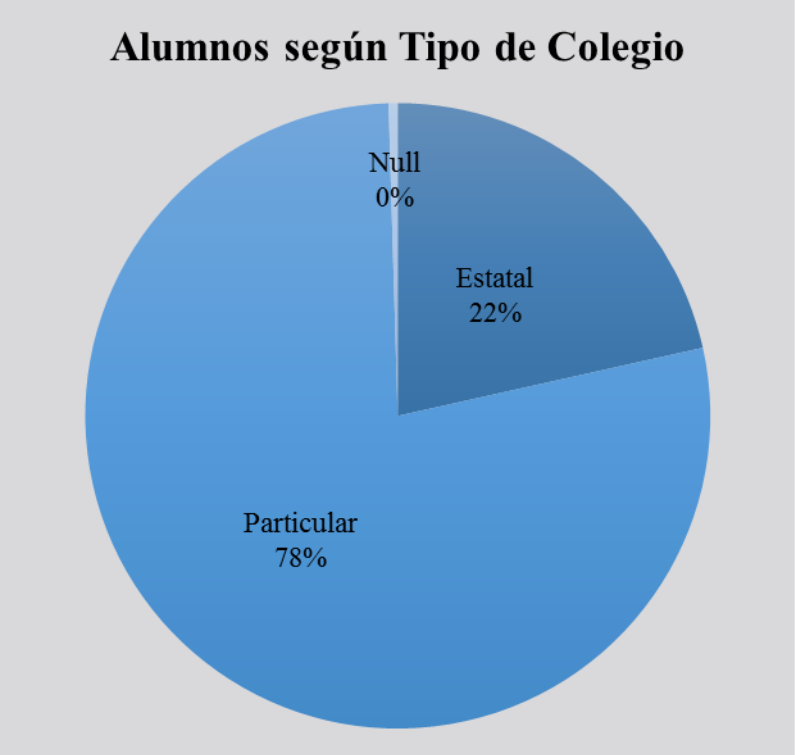


Figura 22: Distribución del alumnado según tipo de colegio de procedencia.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se pudo determinar, que en el Tipo de Colegio sin identificar (con descripción “*Null*” y cuyo código es “N”) hay 43 participantes, y que a mayor abundamiento son los mismos estudiantes de intercambio identificados anteriormente y que a su vez no tienen escala definida.

C. Resultados Iniciales

Se pudo observar que al realizar la normalización de la información que nos proporcionó el Centro de Computo de la Universidad Ricardo Palma, en la columna denominada "Nombre del Colegio de procedencia", nos hemos topado con 2,213 registros de los cuales una proporción mayor al 20% se encontraban duplicados, una variabilidad tan alta tiene como consecuencia la poca o nula utilidad de dicha columna.

D. Limpieza de datos

Se tuvo que eliminar la columna denominada "Nombre del Colegio de procedencia" debido a que contenía una variabilidad muy alta.

Existían 43 registros en la categoría de "Alumno Libre" (con código "00") y que son los mismos que figuran sin escala y sin tipo de colegio de procedencia, los cuales representaban el 0.47% del total, por lo tanto, se procedió a eliminar a los 43 registros identificados.

E. Construcción de datos

Se generó el nuevo archivo de alumnos denominado "alumnos_a.csv", con 9,075 registros.

F. Verificación de calidad de datos

Con el *software* R se revisó las características internas del nuevo archivo, comprobándose que no presentaba ninguna inconsistencia con los resultados previos, el único inconveniente fue que 18 fechas tenían valores perdidos (NA o con contenido Nulo), por lo que se realizó su respectiva imputación.

Tabla 13: Características de la estructura interna final del archivo de alumnos.

alu_cod	car_cod	sexo	nacio	ing_cod	escala	col_tipo
Length: 9,075	11: 1,343	F: 4,400	Min. : 10/03/1957	13: 3,335	A13: 882	E: 1,961
Class : character	61: 1,276	M: 4,675	1st Qu.: 02/05/1997	15: 2,829	A18: 144	P: 7,114
Mode : character	25: 1,026		Median : 13/09/1998	17: 1,271	A23: 5,414	
	63: 863		Mean : 26/03/1998	04: 386	A28: 275	
	51: 651		3rd Qu.: 16/12/1999	05: 305	A33: 2,024	
	41: 569		Max. : 03/05/2004	19: 240	A38: 336	
	(Other): 3,347		NA's : 18	(Other): 709		

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Después de la eliminación de los datos faltantes, se generó un nuevo archivo de alumnos denominado “alumnos_b.csv”, que conservó la cantidad de registros.

Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

4.1.1.4. Archivo de planes curriculares.

A. Conjunto inicial de datos

Denominado “Planes.csv”, el cual contiene las características de todos los cursos en los diferentes planes curriculares de cada carrera. Se encuentra conformado por 4,840 registros y las siguientes columnas:

1. Código de la carrera
2. Código del plan curricular

   Los códigos de los Planes Curriculares vigentes para el periodo de la presente investigación corresponden a: “50”, “51” y “52”.

3. Código del curso
4. Mascara del curso

   Visualización alfanumérica del curso, donde la primera letra corresponde al departamento académico donde se encuentra asignado el curso. La letra "E" corresponde al "Programa de Estudios Básicos".

5. Nombre del curso
6. Ciclo donde se dicta el curso
7. Tipo de Curso: (‘O’ = Obligatorio / ‘E’ = Electivo)
8. Número de créditos del curso
9. Horas de Teoría del curso
10. Horas de Practica del curso

11. Horas de Laboratorio del curso

12. Horas de Taller del curso

13. Prerrequisito, Código del curso que tiene que aprobar

14. Prerrequisito, Número de créditos aprobados

B. Exploración de los datos

Para la etapa de exploración, depuración y filtrado se utilizó el *software* R que generó un archivo de planes denominado "planes_a.csv" con 553 registros.

La cantidad de registros disminuyo debido a que, solo se consideró los cursos del Programa de Estudios Básicos que se encontraban en los planes curriculares con código: "50", "51" o "52". Se seleccionó únicamente 4 columnas, eliminando las columnas restantes porque no aportaban información relevante.

Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

C. Resultados Iniciales

Al realizar un análisis previo con el *software* R, no se consideró la columna denominada "Código del plan curricular", porque se filtró los registros correspondientes a los planes curriculares vigentes. Tampoco se consideró la columna de carrera, porque el contenido de los cursos del PEB es igual indistintamente de la carrera.

Se utilizó MS Excel para revisar los registros duplicados lo que dio como resultado que la cantidad de 553 registros se redujo a 53, y pudimos observar la duplicidad de cursos debido a una incorrecta digitación en la "mascara", como se puede observar en la siguiente muestra:

Tabla 14: Muestra de la relación de cursos duplicados.

Curso	
Mascara	Descripción
EB 0001	Actividades Artísticas y Deportivas
EB0011	Actividades Artísticas y Deportivas

EB-0011	Actividades Artísticas y Deportivas
E B0001	Actividades Artísticas y Deportivas

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Debemos señalar que la cantidad de cursos en el PEB es de 13, por lo tanto habían 40 registros cuya mascara fue digitada y/o generada erróneamente, lo cual representa el 75.5 % del total, es decir de cada 4 registros 3 han sido ingresados de forma incorrecta.

Basándonos en los 53 registros se tuvo que generar una tabla de equivalencias y una tabla con los cursos del plan curricular del PEB.

D. Limpieza de datos

Después de la uniformización, la tabla quedo conformada por 13 registros, que corresponden a los cursos del plan curricular del PEB.

E. Construcción de datos

Se generó el nuevo archivo de planes curriculares denominado “planes_b.csv”, conteniendo 13 registros.

De forma manual y con la ayuda de MS Excel fue necesario generar una tabla de equivalencias, denominada “equi_cursos.csv”, conformada por 53 registros y 2 columnas; la primera columna contiene las distintas mascaras en una cantidad equivalente al número de veces que aparezca el código del curso, la segunda columna contiene el código del curso, como se puede apreciar en el ejemplo siguiente:

Tabla 15: Muestra de la tabla de equivalencias de cursos.

Equivalencias	
Mascara	Código
EB 0001	0001
EB0011	0001
EB-0011	0001
E B0001	0001

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

También se elaboró un diagrama de la malla curricular de los cursos del Programa de Estudios Básicos, cuya utilidad es fundamental para la construcción del *dataset* para cada curso.

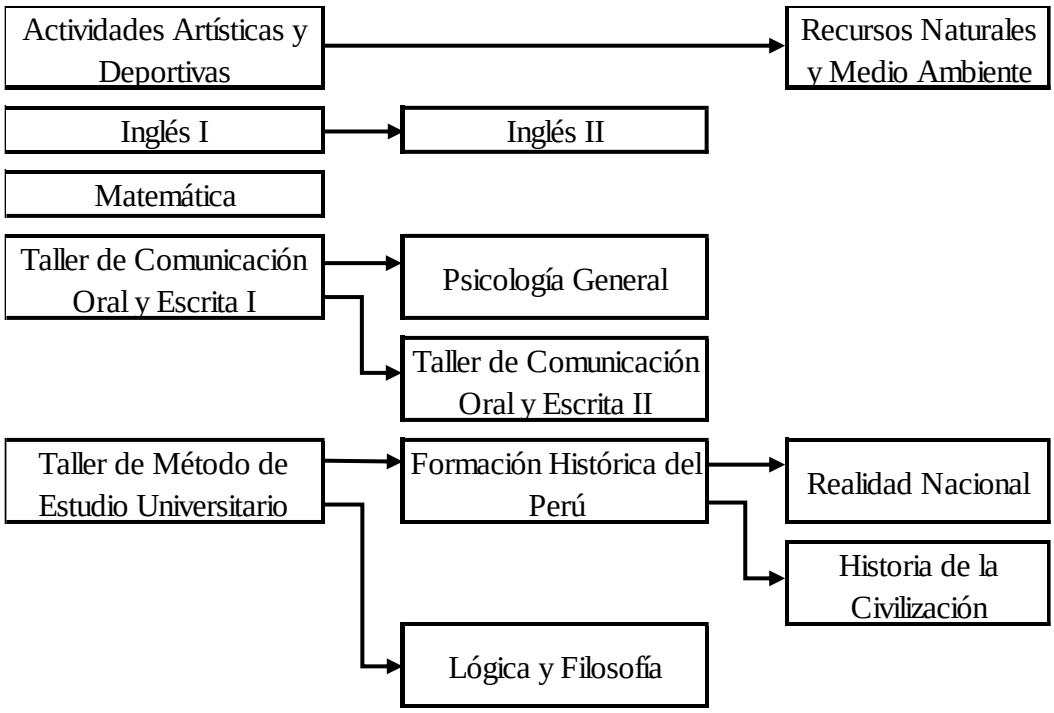


Figura 23: Diagrama de la estructura de la malla curricular del Programa de Estudios Básicos.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

F. Verificación de calidad de datos

El archivo de planes curriculares, después del pre procesamiento y limpieza, está conformado únicamente por la información concerniente a los cursos del PEB, como se muestra a continuación:

Tabla 16: Plan curricular del Programa de Estudios Básicos.

Curso			
Código	Mascara	Descripción	Pre Requisito
0001	EB 0001	Actividades Artísticas y Deportivas	
0002	EB 0002	Taller de Método de Estudio Universitario	
0003	EB 0003	Taller de Comunicación Oral y Escrita I	
0004	EB 0004	Matemática	
0005	EB 0005	Inglés I	
0006	EB 0006	Psicología General	0003
0007	EB 0007	Lógica y Filosofía	0002
0008	EB 0008	Taller de Comunicación Oral y Escrita II	0003
0009	EB 0009	Inglés II	0005
0010	EB 0010	Formación Histórica del Perú	0002

0011	EB 0011	Recursos Naturales y Medio Ambiente	0001
0012	EB 0012	Realidad Nacional	0010
0013	EB 0013	Historia de la Civilización	0010

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Con el *software* R se revisó las características internas del nuevo archivo, comprobándose que no presentaba ninguna inconsistencia. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

4.1.1.5. Archivo del tipo de evaluación.

A. Conjunto inicial de datos

Contiene las distintas evaluaciones que un alumno puede obtener durante el desarrollo de un curso. Se encuentra conformado por 21 registros y las siguientes columnas:

1. Código del tipo de evaluación
2. Descripción del tipo de evaluación

B. Exploración de los datos

Se utilizó MS Excel para revisar y explorar los diferentes tipos de evaluaciones mediante las cuales los docentes de la Universidad Ricardo Palma califican el desempeño académico de los alumnos.

C. Resultados Iniciales

No se observó ninguna inconsistencia.

D. Limpieza de datos

No fue necesario realizar la limpieza de los datos.

E. Construcción de datos

Se conservó el archivo de tipos de evaluaciones denominado “Evaluacion.csv”.

F. Verificación de calidad de datos

El archivo de tipos de evaluaciones contiene la siguiente información:

Tabla 17: Tipos de evaluaciones en la Universidad Ricardo Palma.

Tipo de Evaluación	
Código	Descripción
ASP	Asistencia y Puntualidad
AVP	Avance del Proyecto
CPX	Concurso de Proyecto
CTL	Control Laboratorio
DCA	Demostración del Aprendizaje
EXP	Exposición
EXV	Expovitrina
FIN	Final
INF	Informe
INF	Informe Final
INT	Informe Taller
LAB	Laboratorio
LAM	Lamina
NPA	Nota Participación
PAD	Participación Activa
PAR	Parcial
PRA	Práctica
PRO	Proyecto
PRT	Práctica Teórica
PTL	Práctica Taller
PYF	Proyecto Final
PYL	Proyecto de Laboratorio
PYT	Proyecto Taller
SUP	Sustentación Proyecto
SUS	Sustitutorio
TLR	Taller
TMO	Trabajo Monográfico
TRA	Trabajo de Investigación
TRP	Trabajo Práctico

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Con el *software* R se revisó las características internas del archivo original, comprobándose que no presentaba ninguna inconsistencia. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

4.1.1.6. Archivo de notas.

A. Conjunto inicial de datos

Denominado “Notas.csv”, el cual contiene las notas de los alumnos desde el ciclo 2015-1 hasta el ciclo 2019-0. Se encuentra conformado por 574,283 registros y las siguientes columnas:

1. Semestre académico, el cual indica cuando el alumno asistió al curso
2. Mascara del curso

   El Centro de Computo asignó a la columna el nombre de “Código del curso”, lamentablemente las características correspondían a la “Mascara del curso”.

3. Fórmula de cálculo para obtener la nota final del curso
4. Tipos de evaluación
5. Grupo donde se matriculo
6. Nota obtenida en el tipo de evaluación
7. Código del alumno
8. Nota (o promedio) final del curso

   Calificación en escala vigesimal, para ser considerado aprobado en el curso la nota final debe ser mayor o igual a 11. La nota 99 corresponde al alumno que No Se Presentó (NSP).

B. Exploración de los datos

Con el *software* R se revisó las características internas del archivo original, comprobándose que no presentaba ninguna inconsistencia, asimismo se generó un resumen de notas. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 04 de la presente investigación.

Tabla 18: Características de la estructura interna del archivo que contiene las notas de todos los cursos.

Sem..	Notas	Cod..	Curso	Formula	
20181 :	94,232	EB 0004:	56,864	(((PRA1+PRA2+PRA3)/3)+PAR1+FIN1)/3	79,930
20171 :	87,569	EB 0002:	44,645	(PRA1+PRA2+PRA3+PRA4)/4 :	65,387
20172 :	84,572	EB 0006:	41,140	(PAR1+FIN1+(PRA1+PRA2+PRA3+PRA4)/3)/3	64,091
20182 :	81,800	EB 0011:	35,907	((PRA1+PRA2+PRA3+PRA4)/4+PAR1+FIN1)/3	48,127
20162 :	70,913	EB 0007:	34,867	(PRA1+PRA2+PRA3+PRA4+PRA5)/5	47,943
20161 :	68,624	EB 0010:	33,397	(TRA1+PAR1+FIN1+2*((PRA1+PRA2+PRA3+I	41,581
(Other):	86,573	(Other):	327,463	(Other) :	227,224

Eval		Grupo		Nota		Cod..	Alumno	Nota.1	
PRA1 :	81,491	01:	23,356	0.00 :	51,755	Length:	574,283	11:	79,307
PRA2 :	81,418	02:	22,572	14.00 :	50,645	Class :	character	12:	74,223
PRA3 :	81,370	03:	20,567	15.00 :	48,766	Mode :	character	13:	68,576
PRA4 :	62,500	04:	19,179	16.00 :	45,220			14:	62,276
FIN1 :	52,002	05:	18,663	13.00 :	43,063			15:	55,268
PAR1 :	52,001	06:	18,046	12.00 :	42,729			16:	41,804
(Other):	163,501	(Other):	451,900	(Other):	292,105			(Other):	192,829

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se utilizó MS Excel para revisar y explorar el resumen del archivo de notas, generado con el *software* R, el cual contenía el acumulado de los diferentes tipos de evaluaciones agrupados por semestre y año.

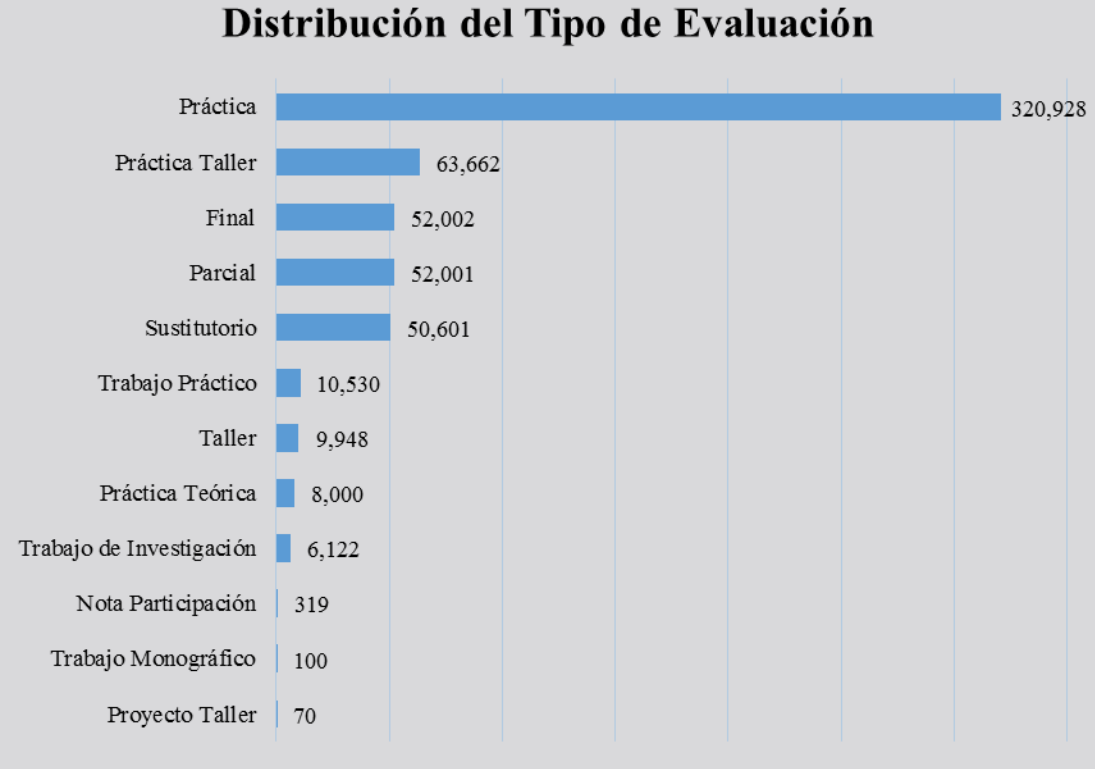


Figura 24: Distribución del Tipo de Evaluación en el archivo que contiene las notas de todos los cursos.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

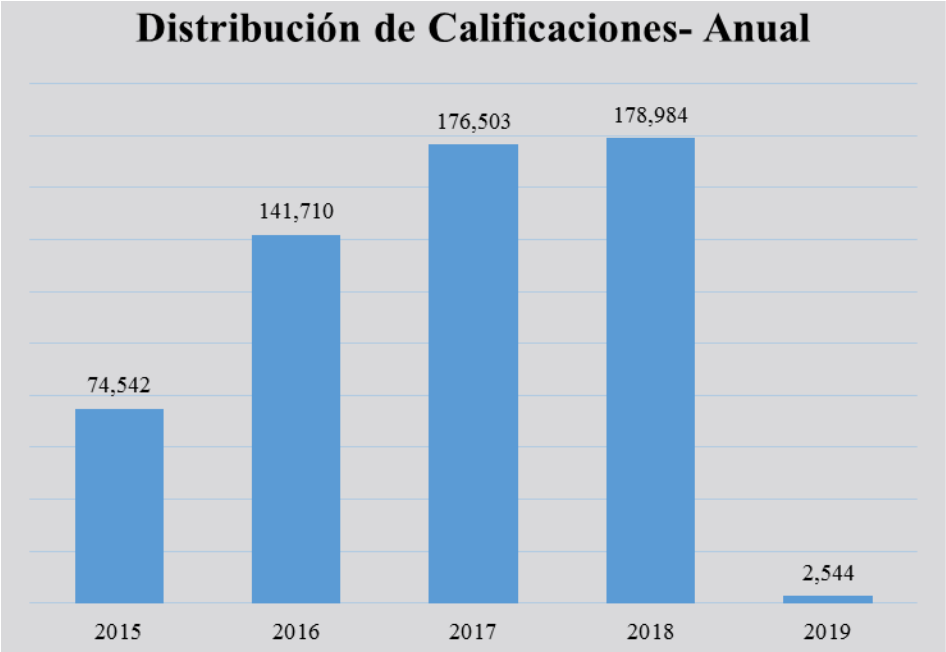


Figura 25: Distribución de las Calificaciones por año en el archivo que contiene las notas de todos los cursos.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para poder transformar el archivo de notas se procedió a trabajar con el *software* Python. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 05 de la presente investigación.

Como primer paso, se procedió a generar el archivo denominado “notas_1.csv”, el cual contenía la misma cantidad de registros y el número de columnas se redujo a 6 pero, con un elemento diferenciador, la columna denominada “mascara del curso” fue transformada a “código del curso” con ayuda del archivo “equi_cursos.csv” elaborado en la sección de: Comprensión de los datos del Archivo de planes curriculares.

Acto seguido, se generó la división del archivo “notas_1.csv” en trece archivos, correspondiendo cada uno a un determinado curso del Programa de Estudios Básicos (PEB); adicionalmente se construyeron dos archivos por cada curso, el primero contenía los diferentes tipos de evaluaciones a las cuales se sometió el alumno durante el dictado del curso y el segundo contenía el histórico de los diferentes tipos de evaluaciones del curso (los tipos de evaluaciones establecidas en el curso en cada ciclo y por cada grupo).

C. Resultados Iniciales

Para una mejor comprensión del procedimiento utilizado, la explicación está basada en el curso del PEB denominado: “Actividades Artísticas y Deportivas” con código “0001”; a mayor abundamiento debemos indicar que el archivo que contiene las notas del curso se denominó “0001.csv”.

Con el *software* R se revisó las características internas del archivo de notas, comprobándose que no presentaba ninguna inconsistencia, asimismo se visualizó la frecuencia de los tipos de evaluaciones. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 06 de la presente investigación.

El archivo de los diferentes tipos de evaluaciones en el curso “Actividades Artísticas y Deportivas”, denominado “0001_eva.csv”, contiene la siguiente información:

Tabla 19: Tipos de evaluaciones en el curso “0001”.

Código de Evaluación
PRA1
PRA2
PRA3
PRA4
PTL1
PTL2
PTL3
PTL4

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Fue necesario determinar el código equivalente por cada tipo de evaluación, por lo tanto, mediante MS Excel se analizó la información de la tabla de “tipos de evaluaciones“ en conjunto con el archivo de los tipos de evaluación histórica (denominado “0001_hist.csv”) y se generó un archivo denominado “0001_equi.csv”, conformada por 8 registros y 2 columnas; la primera columna contiene los diferentes tipos de evaluación hallados en el curso “Actividades Artísticas y Deportivas”, y la segunda columna contiene el código del tipo de evaluación equivalente, lo cual puede visualizarse a continuación:

Tabla 20: Tabla de equivalencia para los tipos de evaluaciones en el curso "0001".

Equivalencias	
Original	Nuevo
PRA1	PRA1
PRA2	PRA2
PRA3	PRA3
PRA4	PRA4
PTL1	PRA1
PTL2	PRA2
PTL3	PRA3
PTL4	PRA4

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

D. Limpieza de datos

No fue necesario realizar la limpieza de los datos.

E. Construcción de datos

Para construir el archivo de notas en el formato necesario, se procedió a trabajar con el *software* Python. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 05 de la presente investigación.

En primer lugar se realizó la generación del archivo de notas con el tipo de evaluación estandarizado denominado "0001_1.csv", que contenía la misma cantidad de registros y su estructura es la siguiente:

Tabla 21: Estructura Inicial del archivo de notas del curso "0001".

Estructura Inicial				
Semestre	Alumno	Evaluación	Calificación	Promedio Final
2015-1	A	Parcial	10	10
2015-1	A	Practica 1	9	10
2015-1	A	Practica 2	11	10
2015-1	A	Final	9	10
2015-2	A	Parcial	12	12
2015-2	A	Practica 1	11	12
2015-2	A	Practica 2	13	12
2015-2	A	Final	11	12

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Lamentablemente, dicha estructura no es óptima para realizar predicciones, por lo tanto, tuvo que ser transformada a la siguiente estructura:

Tabla 22: Estructura Final del archivo de notas del curso "0001".

Estructura Final							
Semestre	Alumno	Promedio Final	Ha llevado el curso	Evaluaciones			
				Parcial	Practica 1	Practica 2	Final
2015-1	A	10	2 veces	10	9	11	9
2015-2	A	11	2 veces	12	11	13	11

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Con dicha estructura se tuvieron que generar 2 archivos:

- Un archivo denominado "0001_2.csv" que contiene todos los registros y se utilizó para fines estadísticos.
- Un archivo denominado "0001_3.csv" que contiene solo los registros de los alumnos con rendimiento académico aprobado y desaprobado (nota o promedio final del curso en un rango comprendido entre 01 y 20) y se utilizó para la construcción del *dataset*.

F. Verificación de calidad de datos

Con el *software* R se revisó las características internas del archivo denominado "0001_2.csv", comprobándose que no presentaba ninguna inconsistencia. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 06 de la presente investigación.

Tabla 23: Características de la estructura interna final del archivo que contiene las notas del curso "0001".

semestre		**alu_cod**		**eva_fin**		**veces**		**estado**	
20171 :	1,561	Length:	9,365	Min. :	1.00	Min. :	1.000	A:	7,984
20161 :	1,548	Class :	character	1st Qu.:	14.00	1st Qu.:	1.000	D:	568
20181 :	1,528	Mode :	character	Median :	15.00	Median :	1.000	N:	813
20162 :	1,131			Mean :	21.99	Mean :	1.212		
20172 :	993			3rd Qu.:	17.00	3rd Qu.:	1.000		
20151 :	924			Max. :	99.00	Max. :	7.000		
(Other):	1,680								

PRA1		**PRA1**		**PRA3**		**PRA4**	
Min. :	0.00	Min. :	0.00	Min. :	0.00	Min. :	0.00

1st Qu.:	12.00	1st Qu.:	13.00	1st Qu.:	12.00	1st Qu.:	12.00
Median :	15.00	Median :	15.00	Median :	15.00	Median :	15.00
Mean :	13.14	Mean :	13.46	Mean :	13.19	Mean :	13.29
3rd Qu.:	16.00	3rd Qu.:	16.00	3rd Qu.:	17.00	3rd Qu.:	17.00
Max. :	20.00	Max. :	20.00	Max. :	20.00	Max. :	20.00
NA's :	2.00	NA's :	9.00	NA's :	29.00	NA's :	49.00

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Con respecto a la frecuencia de la nota final de los estudiantes en el curso "0001", se detectó la distribución siguiente:

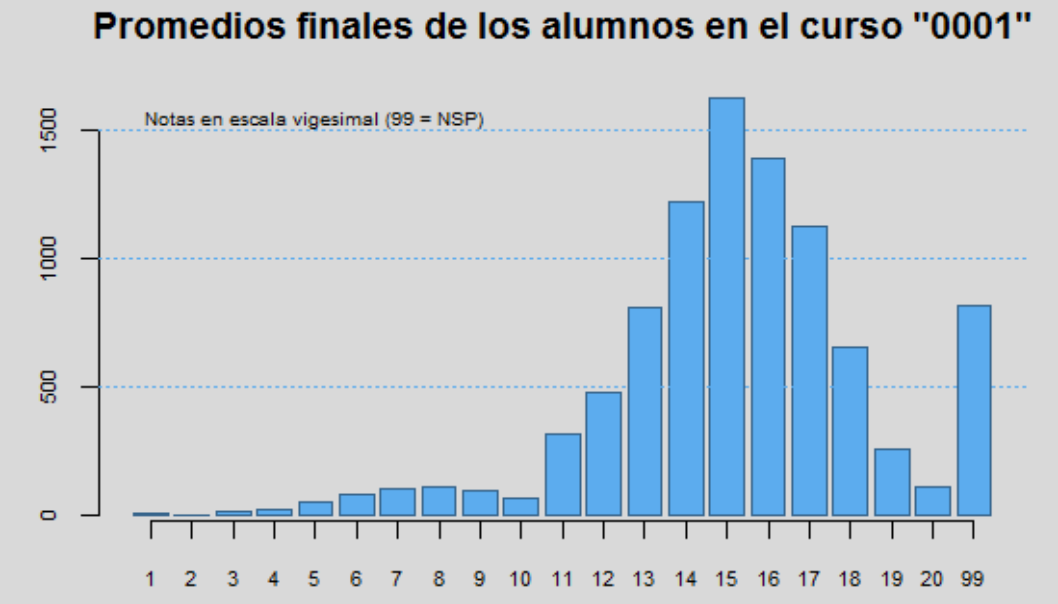


Figura 26: Histograma de la nota final de los alumnos en el curso "0001".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Con respecto al rendimiento académico, se pudo observar lo siguiente:

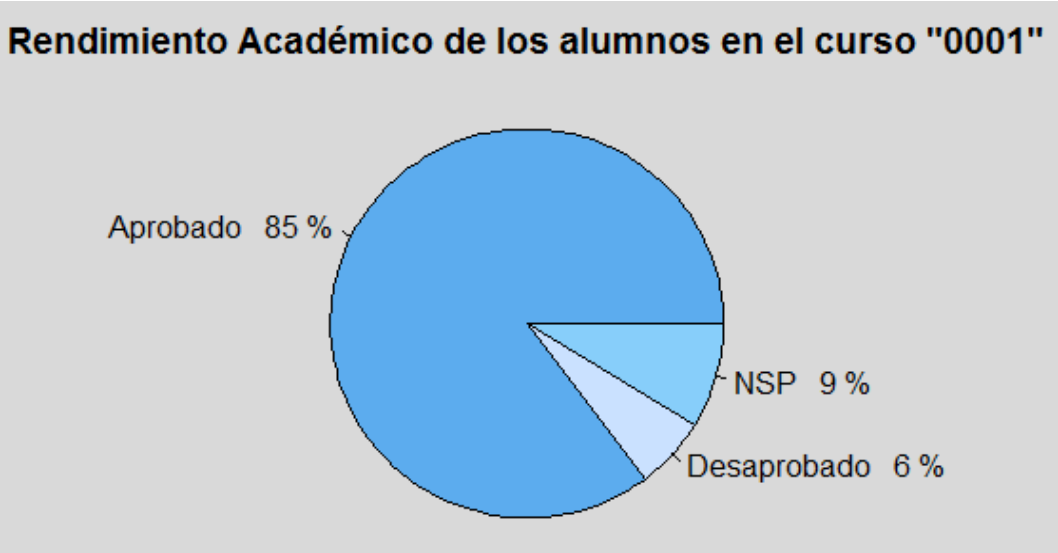


Figura 27: Distribución del rendimiento académico del alumnado en el curso "0001".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

También con el *software* R se revisó las características internas de los archivos denominados "0001_1.csv" y "0001_3.csv", comprobándose que ninguno de ellos presentaba alguna inconsistencia. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 06 de la presente investigación.

G. Comprensión de los cursos faltantes

Con el *software* R se realizó el mismo procedimiento descrito en los pasos anteriores y se revisó las características internas de los archivos de los cursos restantes, comprobándose que ninguno de ellos presentaba alguna inconsistencia. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 06 y en el Anexo 11 de la presente investigación.

4.1.1.7. Archivo de cursos.

Para una mejor comprensión del procedimiento utilizado, la explicación está basada en el curso del PEB denominado: "Actividades Artísticas y Deportivas" con código "0001"; a mayor abundamiento debemos indicar que los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 07 de la presente investigación.

A. Conjunto inicial de datos

El archivo denominado "0001_3.csv" contiene las características de los alumnos con rendimiento académico aprobado y desaprobado en el curso cuyo código es el: "0001". Se encuentra conformado por 8,552 registros y las siguientes columnas:

1. Semestre académico, el cual indica cuando el alumno asistió al curso
2. Código del alumno
3. Nota (o promedio) final del curso
4. Número de veces que estuvo matriculado en el curso para poder aprobarlo, incluyendo la ocasión donde lo aprobó.
5. Estado del rendimiento académico: ('Si' = Aprobó / 'No' = Desaprobó)
6. Nota obtenida en la primera práctica
7. Nota obtenida en la segunda práctica

8. Nota obtenida en la tercera práctica

9. Nota obtenida en la cuarta práctica

Se debe tener en consideración que, para cada curso que fue analizado, a partir de la sexta columna los tipos de evaluaciones son distintos, en concordancia con las particularidades del mismo curso.

B. Exploración de los datos

Se utilizó el *software* R para revisar y realizar la verificación de la cantidad de registros en los siguientes ámbitos:

- Si la cantidad de notas del archivo original denominado "0001_1.csv" coincide con la cantidad de notas en las diferentes evaluaciones del archivo denominado "0001_2.csv".

- Si la cantidad de registros no vacíos (con notas) del archivo denominado "0001_3.csv" coincide con la cantidad de notas no vacías de aprobados y desaprobados del archivo denominado "0001_2.csv".

- Si la cantidad de registros vacíos (sin notas) del archivo denominado "0001_3.csv" coincide con la cantidad de notas vacías de aprobados y desaprobados del archivo denominado "0001_2.csv".

Adicionalmente se debe comprobar si el curso contenía valores perdidos (NA o con contenido Nulo) y determinar si el curso comprendía entre sus diferentes evaluaciones, aquella evaluación denominada "Sustitutorio" (código "SUS").

C. Resultados Iniciales

Se determinó que la consistencia en la cantidad de registros para los tres ámbitos descritos en la sección anterior arrojó resultados correctos.

Mediante al análisis de los valores perdidos se obtuvieron los porcentajes de datos perdidos por fila y por columna inferiores al 30%; también se determinó que el curso "no" contenía el tipo de evaluación denominada "Sustitutorio".

Debemos tener presente que, si hubiésemos encontrado alguna columna con porcentaje mayor o igual al 30%, entonces dicha columna debió haber sido eliminada.

Se debe tener en consideración que, si en el curso analizado, se encuentra la evaluación denominada "Sustitutorio", dicha nota se debe reemplazar en la evaluación denominada "Parcial" (código "PAR") o en la evaluación denominada "Final" (código "FIN"); el criterio de evaluación es el siguiente: la nota del "Sustitutorio" reemplaza la nota mínima a seleccionar entre el "Parcial" y el "Final". Adicionalmente se debe eliminar la columna "Sustitutorio" y generar una nueva columna que contenga los valores 1 (el alumno "Si" ha sido evaluado en el "Sustitutorio") o 0 (el alumno "No" ha sido evaluado en el "Sustitutorio").

D. Limpieza de datos

El curso contenía valores perdidos, pero ninguna columna superaba el 30% de datos perdidos, por lo tanto no se realizó ninguna eliminación.

No se encontró la evaluación denominada "Sustitutorio", por lo tanto no se realizó ningún cambio.

Se debe tener en consideración que, para cada curso, la limpieza de datos es única, teniendo coherencia con las particularidades del mismo.

E. Construcción de datos

Después de la limpieza de los datos, se generó un nuevo archivo de notas denominado "0001_4.csv", que conservó la cantidad de registros.

Tabla 24: Características de la estructura interna del archivo de notas del curso "0001" preparado para generar el dataset.

semestre		**alu_cod**		**eva_fin**		**veces**		**aprobo**	
20171 :	1,445	Length:	8,252	Min. :	1.00	Min. :	1.000	No:	568
20161 :	1,419	Class :	character	1st Qu.:	13.00	1st Qu.:	1.000	Si:	7,984
20181 :	1,408	Mode :	character	Median :	15.00	Median :	1.000		
20162 :	1,047			Mean :	14.67	Mean :	1.129		
20172 :	896			3rd Qu.:	17.00	3rd Qu.:	1.000		
20151 :	858			Max. :	20.00	Max. :	7.000		
(Other):	1,479								

PRA1		PRA1		PRA3		PRA4	
Min. :	0.00	Min. :	0.00	Min. :	0.00	Min. :	0.00
1st Qu.:	13.00	1st Qu.:	14.00	1st Qu.:	13.00	1st Qu.:	14.00
Median :	15.00	Median :	15.00	Median :	15.00	Median :	16.00
Mean :	14.39	Mean :	14.74	Mean :	14.44	Mean :	14.56
3rd Qu.:	16.00	3rd Qu.:	16.00	3rd Qu.:	17.00	3rd Qu.:	17.00
Max. :	20.00	Max. :	20.00	Max. :	20.00	Max. :	20.00
NA's :	2.00	NA's :	8.00	NA's :	24.00	NA's :	46.00

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

F. Verificación de calidad de datos

El nuevo archivo de notas, después del pre procesamiento y limpieza, contiene únicamente la información necesaria para la creación del *dataset* donde el curso se encuentre involucrado, lo cual se puede apreciar en el esquema de construcción de cada *dataset* para los cursos del PEB.

G. Comprensión de los cursos faltantes

Con el *software* R se preparó la estructura interna de todos los cursos en base al procedimiento descrito en los pasos anteriores. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 07 y en el Anexo 12 de la presente investigación.

4.1.2. Preparación de los datos.

Como resultado de la etapa de comprensión de datos, se obtuvo como insumo un archivo de alumnos y 13 archivos de cursos, a partir de los cuales se construyó un *dataset* por cada curso del Programa de Estudios Básicos, como se puede apreciar en el diagrama siguiente:

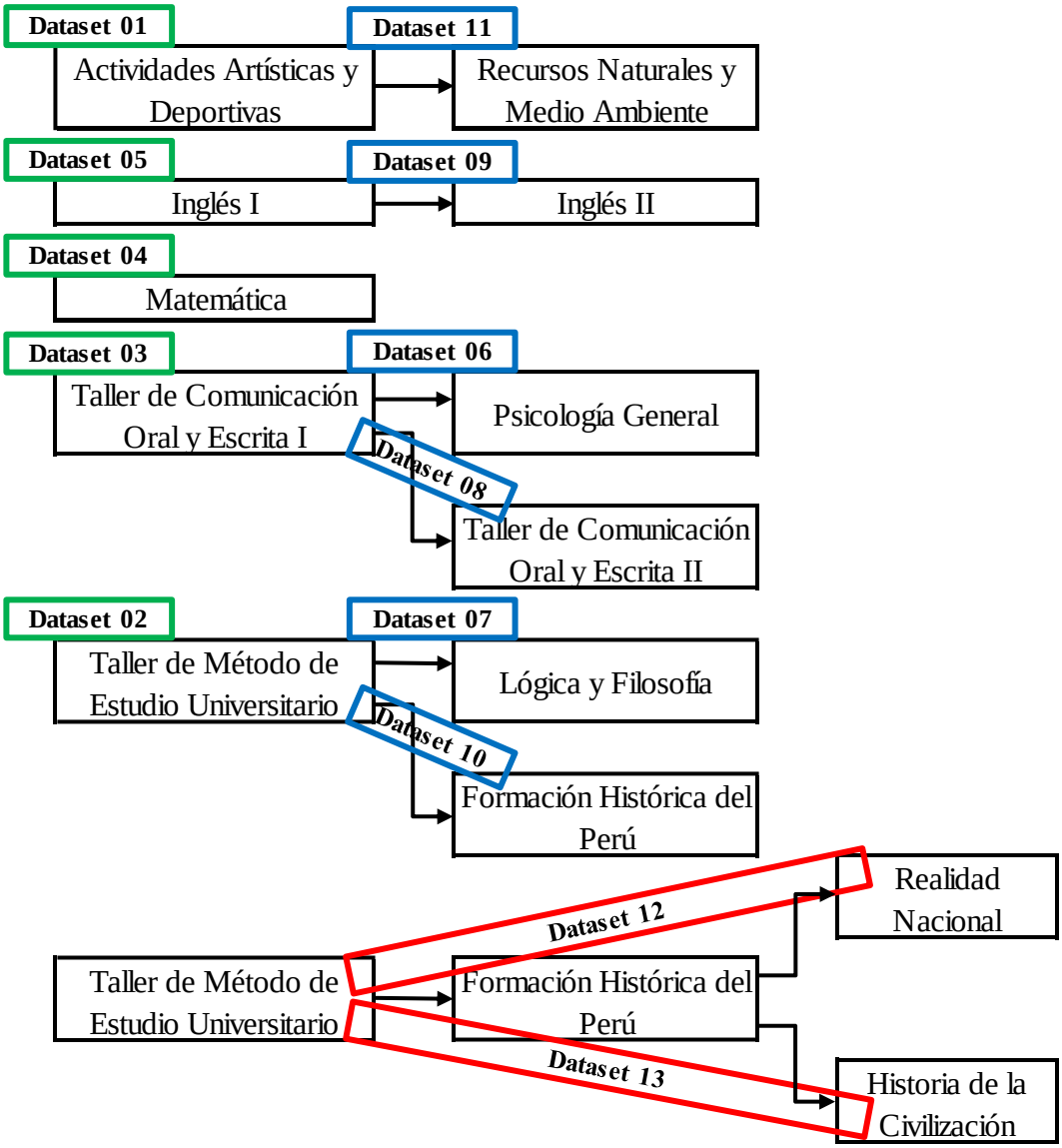


Figura 28: Diagrama del esquema de construcción de cada *dataset* para los cursos del Programa de Estudios Básicos.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

4.1.2.1. Procedimiento.

Para una mejor comprensión de la secuencia adoptada, la explicación está basada en el curso del PEB denominado: "Actividades Artísticas y Deportivas" con código "0001" y que generó el *dataset* denominado "DS_0001.csv".

A. Selección e Integración de los datos

Se decidió, para la creación del *dataset*, incluir todas las características del archivo de alumnos denominado "alumnos_b.csv" y con respecto a los atributos del archivo del curso denominado "0001_4.csv" se decidió incluir las siguientes columnas:

1. Semestre académico, el cual indica cuando el alumno asistió al curso

2. Código del alumno

3. Número de veces que estuvo matriculado en el curso para poder aprobarlo, incluyendo la ocasión donde lo aprobó.

4. Estado del rendimiento académico: (‘Si’ = Aprobó / ‘No’ = Desaprobó)

5. Nota obtenida en la primera evaluación o práctica

   Dicha evaluación coincide con un periodo de tiempo comprendido entre el 25% y el 30% del desarrollo del curso, por lo tanto, al tener la predicción del desempeño académico del alumno con bastante anticipación, es posible tomar las acciones correctivas para la mejora de su rendimiento.

Al realizarse la fusión de ambas tablas, teniendo como identificador (atributo en común) el “código del alumno”, se obtuvo como resultado un *dataset* donde se integraron las columnas de ambas tablas, pero, sin la columna del identificador (por el principio de anonimato), y con respecto a las filas permanecieron aquellos registros que se encontraban en ambas tablas.

El *dataset* inicial del curso “Actividades Artísticas y Deportivas”, estuvo conformado por 8,539 filas y 10 columnas, podemos observar sus características en:

Tabla 25: Características de la estructura interna inicial del *dataset* del curso “0001”.

car_cod		sexo		nacio		ing_cod		escala	
11:	1,313	F:	4,217	Min. :	08/04/1957	13:	3,188	A13:	855
61:	1,170	M:	4,322	1st Qu.:	07/06/1997	15:	2,625	A18:	134
25:	989			Median :	29/09/1998	17:	1,228	A23:	5,102
63:	816			Mean :	03/05/1998	04:	376	A28:	253
51:	630			3rd Qu.:	22/12/1999	05:	302	A33:	1,905
41:	536			Max. :	03/05/2004	08:	182	A38:	290
(Other):	3,085					(Other):	638		

col_tipo		semestre_1		veces_1		aprobo		PRA1_1	
E:	1,830	20171 :	1,444	Min. :	1.000	No:	568	Min. :	0.00
P:	6,709	20161 :	1,419	1st Qu.:	1.000	Si:	7,971	1st Qu.:	13.00
		20181 :	1,400	Median :	1.000			Median :	15.00
		20162 :	1,047	Mean :	1.129			Mean :	14.39
		20172 :	895	3rd Qu.:	1.000			3rd Qu.:	16.00
		20151 :	858	Max. :	7.000			Max. :	20.00

(Other): 1,476 NA's : 1.00

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se debe tener en consideración que, si en la creación del *dataset*, se necesitó incorporar información de los cursos que figuran como prerrequisito, entonces, de cada curso que es prerrequisito se debió incluir las siguientes columnas:

1. Semestre académico, el cual indica cuando el alumno asistió al curso
2. Código del alumno
3. Número de veces que estuvo matriculado en el curso para poder aprobarlo, incluyendo la ocasión donde lo aprobó.
4. Nota (o promedio) final del curso

B. Limpieza de datos

Se procedió a revisar y efectuar la verificación de la calidad del *dataset* en los siguientes ámbitos:

1. Verificación de valores perdidos (NA)

   Mediante al análisis de los valores perdidos se obtuvo los porcentajes de datos perdidos por fila y por columna. Si hubiésemos encontrado alguna columna con porcentaje mayor o igual a 30%, entonces dicha columna debió haber sido eliminada. Posteriormente se realizó la imputación de los valores perdidos, siempre y cuando la eliminación de la columna no haya desaparecido a todos los valores perdidos.

2. Revisión de valores atípicos (*Outliers*).

   El análisis de los *outliers* se realiza a nivel de columnas que contienen datos con valores numéricos. Si se descubre alguna columna con porcentaje menor a 5%, entonces se pudo eliminar dicha columna.

3. Revisión de predictores nzv (*near zero variance*).

   El análisis de los datos con variancia cero o casi cero se realiza a nivel de columnas, y si se localiza alguna, entonces, se puede eliminar dicha columna.

4. Creación de la matriz de correlación

   Mediante al análisis de la matriz de correlación se visualiza las relaciones entre las variables numéricas. Si se detecta alguna columna con un índice de correlación mayor a 0.75 (columnas altamente correlacionadas), entonces se pudo eliminar dicha columna.

Se procedió a revisar y efectuar los 4 análisis mencionados en el *dataset* del curso "Actividades Artísticas y Deportivas", y se obtuvieron los porcentajes de datos perdidos por fila y por columna inferiores al 30%; también se determinó que el curso "no" contenía valores atípicos, ni predictores nzv, ni columnas con el índice de correlación mayor a 0.75.

C. Construcción de datos

El *dataset* contiene una columna cuyos valores son de tipo fecha y para su procesamiento, en la mayoría de modelos, es necesaria una transformación, es decir generar una nueva columna que es una derivación de la columna inicial.

Para este caso en particular se creó una columna que contenía la diferencia estandarizada entre la fecha de nacimiento y el semestre académico (periodo académico que indica cuando llevó el curso); al mismo tiempo se eliminó las columnas "fecha de nacimiento" y "semestre académico".

Se debe tener en consideración que, si el *dataset*, contiene información de los cursos que figuran como prerrequisito, entonces, por cada curso que es prerrequisito se debió efectuar la misma operación descrita en el párrafo anterior.

D. Verificación de calidad de datos

El archivo denominado "DS_0001.csv", contiene el *dataset* final del curso "Actividades Artísticas y Deportivas", conformado por 8,539 filas y 9 columnas, como se puede apreciar a continuación:

Tabla 26: Características de la estructura interna final del *dataset* del curso "0001".

car_cod		sexo		ing_cod		escala		col_tipo	
11:	1,313	F:	4,217	13:	3,188	A13:	855	E:	1,830

61:	1,170	M:	4,322	15:	2,625	A18:	134	P:	6,709
25:	989			17:	1,228	A23:	5,102		
63:	816			04:	376	A28:	253		
51:	630			05:	302	A33:	1,905		
41:	536			08:	182	A38:	290		
(Other):	3,085			(Other):	638				

veces_1		aprobo		PRA1_1		sem_dif_1	
Min. :	1.000	No:	568	Min. :	0.00	Min. :	1.00
1st Qu.:	1.000	Si:	7,971	1st Qu.:	13.00	1st Qu.:	13.00
Median :	1.000			Median :	15.00	Median :	16.00
Mean :	1.129			Mean :	14.39	Mean :	17.69
3rd Qu.:	1.000			3rd Qu.:	16.00	3rd Qu.:	19.00
Max. :	7.000			Max. :	20.00	Max. :	135.00

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se utilizó el *software* R para ejecutar los procedimientos indicados en los párrafos anteriores para lograr el *dataset* final. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 08 de la presente investigación.

E. Formateo de datos para el modelado

La estructura interna final del *dataset* se debió modificar porque contiene variables alfanuméricas que fueron transformadas a *dummies*; cada variable convertida a *dummy* generó tantas columnas como clases tenía la variable, como consecuencia, se procedió a eliminar una de las columnas *dummy* de cada una de las variables, con la finalidad de obtener de cualquiera de las clases de cada variable una importancia equivalente; la estructura interna final y modificada, de 13 columnas, se observa a continuación:

Tabla 27: Características de la estructura interna final del *dataset* del curso "0001" después de la transformación.

Campo		Información que contiene
'data.frame'	:	8539 obs. of 13 variables:
$ car_cod	:	Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1 ...
$ sexoM	:	int 1 0 0 0 0 1 0 0 0 1 ...
$ ing_cod	:	Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6 ...
$ escalaA18	:	int 0 0 0 0 0 0 0 0 0 0 ...
$ escalaA23	:	int 1 0 1 1 1 1 1 1 1 1 ...
$ escalaA28	:	int 0 0 0 0 0 0 0 0 0 0 ...
$ escalaA33	:	int 0 1 0 0 0 0 0 0 0 0 ...
$ escalaA38	:	int 0 0 0 0 0 0 0 0 0 0 ...
$ col_tipoP	:	int 1 1 1 1 1 0 1 1 1 1 ...

```
$ veces_1     :  int  1 1 1 1 1 1 1 1 1 1 ...
$ aprobo      :  Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
$ PRA1_1      :  int  14 16 18 16 13 17 14 13 12 14 ...
$ sem_dif_1   :  num  12 12 15 12 12 15 12 12 12 12 ...
```

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

A mayor abundamiento, debemos indicar que una variable ficticia o *dummy*, es en esencia una variable dicotómica que se encuentra asociada al número 1 y al número 0, que representan al valor "verdadero" y al valor "falso" respectivamente. Si deseamos transformar, a variables ficticias, una variable que contiene una cantidad de clases como "n", entonces, se crearán "n" columnas de variables ficticias, cada una conteniendo los números 1 y 0. (Abbott, 2014, pág. 107)

A continuación, se dividió aleatoriamente el *dataset* en 2 partes: una porción del *dataset* denominada "train" donde se realizó el entrenamiento del modelo (70% de la población) y la porción restante del *dataset* denominada "test" donde se ejecutó la evaluación del modelo (30% de la población).

Se observó que, la muestra de entrenamiento denominada "train" se encontraba desbalanceada, es decir la proporción entre "Aprobó" y "Desaprobó" superaba la relación de 9 a 1, por lo que se realizó un proceso de balanceo de los datos usando el algoritmo SMOTE (genera datos artificiales que están sustentados en datos similares del conjunto de variables de la muestra minoritaria) únicamente en "train".

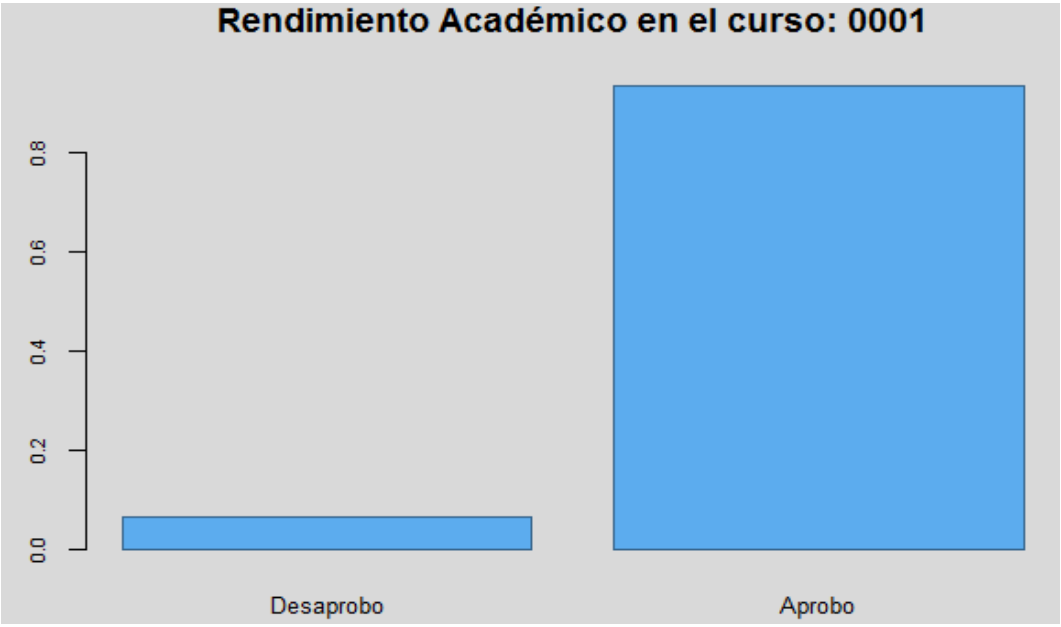


Figura 29: Distribución de la variable dependiente en la muestra de entrenamiento "train" del curso "0001".

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se debe tener en cuenta que para cada *dataset*, fue obligatorio hacer la transformación de las variables alfanuméricas y eliminar una columna de cada conversión, realizar la división del *dataset* en "train" y "test", pero, la realización del balanceo de datos está relacionada con las particularidades propias de cada *dataset*.

Se utilizó el *software* R para ejecutar los procedimientos indicados en la presente sección con la finalidad de preparar el *dataset* para el modelado. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 08 de la presente investigación.

F. Preparación de los cursos faltantes

Con el *software* R se generó el *dataset* de cada curso, siguiendo el mismo procedimiento descrito en los pasos anteriores para cada uno. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 08 y en el Anexo 13 de la presente investigación.

4.1.2.2. Sumario.

Nuestro objetivo es que, en todos los cursos deberíamos poder distinguir a los alumnos con alto y bajo rendimiento académico o en otras palabras a los estudiantes aprobados y desaprobados respectivamente, por lo tanto, nos encontramos frente a un problema de Clasificación.

A. Estructura del *dataset* desde el curso "0001" hasta el curso "0005"

Cada *dataset* posee las siguientes columnas:

1. Código de la carrera
2. Sexo ('F' = Femenino / 'M' = Masculino)
3. Código de la modalidad de ingreso
4. Escala de pago
5. Tipo del colegio de procedencia ('P' = Particular / 'E' = Estatal)
6. Primera Nota obtenida en el curso

7. Número de veces que estuvo matriculado en el curso para poder aprobarlo, incluyendo la ocasión donde lo aprobó.

8. Diferencia estandarizada entre la fecha de nacimiento y el periodo académico que indica cuando llevó el curso

9. Estado del rendimiento académico: (‘Si’ = Aprobó / ‘No’ = Desaprobó), es la variable dependiente.

A continuación, mostramos los primeros 10 registros de cada *dataset* desde el curso “0001” hasta el curso “0005”:

Tabla 28: Visualización de los primeros registros del *dataset* del curso “0001”.

car_cod	sexo	ing_cod	escala	col_tipo	PRA1_1	veces_1	sem_dif_1	**Aprobó**
11	M	6	A23	P	14	1	12	**Si**
33	F	6	A33	P	16	1	12	**Si**
11	F	6	A23	P	18	1	15	**Si**
11	F	6	A23	P	16	1	12	**Si**
11	F	6	A23	P	13	1	12	**Si**
11	M	6	A23	E	17	1	15	**Si**
11	F	6	A23	P	14	1	12	**Si**
11	F	6	A23	P	13	1	12	**Si**
11	F	6	A23	P	12	1	12	**Si**
11	M	6	A23	P	14	1	12	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 29: Visualización de los primeros registros del *dataset* del curso “0002”.

car_cod	sexo	ing_cod	escala	col_tipo	PRA1_1	veces_1	sem_dif_1	**Aprobó**
11	M	6	A23	P	13	1	12	**Si**
33	F	6	A33	P	16	1	12	**Si**
11	F	6	A23	P	17	1	15	**Si**
11	F	6	A23	P	17	1	12	**Si**
11	F	6	A23	P	18	1	12	**Si**
11	M	6	A23	E	15	1	15	**Si**
11	F	6	A23	P	15	1	12	**Si**
11	F	6	A23	P	14	1	12	**Si**
11	F	6	A23	P	9	1	12	**Si**
11	M	6	A23	P	11	1	12	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 30: Visualización de los primeros registros del *dataset* del curso "0003".

car_cod	sexo	ing_cod	escala	col_tipo	PRA1_1	veces_1	sem_dif_1	**Aprobó**
11	M	6	A23	P	19	1	12	**Si**
33	F	6	A33	P	17	1	12	**Si**
11	F	6	A23	P	11	1	15	**Si**
11	F	6	A23	P	13	1	12	**Si**
11	F	6	A23	P	15	1	12	**Si**
11	M	6	A23	E	13	1	15	**Si**
11	F	6	A23	P	13	1	12	**Si**
11	F	6	A23	P	14	1	12	**Si**
11	F	6	A23	P	12	1	12	**Si**
11	M	6	A23	P	13	1	12	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 31: Visualización de los primeros registros del *dataset* del curso "0004".

car_cod	sexo	ing_cod	escala	col_tipo	PRA1_1	veces_1	sem_dif_1	**Aprobó**
11	M	6	A23	P	14	1	12	**Si**
33	F	6	A33	P	17	1	12	**Si**
11	F	6	A23	P	15	1	15	**Si**
11	F	6	A23	P	19	1	12	**Si**
11	F	6	A23	P	6	1	12	**Si**
11	M	6	A23	E	12	1	15	**Si**
11	F	6	A23	P	15	1	12	**Si**
11	F	6	A23	P	12	2	12	**No**
11	F	6	A23	P	10	2	13	**Si**
11	F	6	A23	P	0	2	12	**No**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 32: Visualización de los primeros registros del *dataset* del curso "0005".

car_cod	sexo	ing_cod	escala	col_tipo	PRA1_1	veces_1	sem_dif_1	**Aprobó**
11	M	6	A23	P	16	1	18	**Si**
11	F	6	A23	P	15	1	22	**Si**
11	F	6	A23	P	14	1	19	**Si**
11	F	6	A23	P	15	1	19	**Si**
11	M	6	A23	E	20	1	22	**Si**
11	F	6	A23	P	12	1	18	**Si**
11	F	6	A23	P	20	1	19	**Si**
11	F	6	A23	P	9	1	18	**Si**
11	M	6	A23	P	13	1	19	**Si**
41	F	6	A33	P	14	1	18	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

B. Estructura del *dataset* desde el curso “0006” hasta el curso “0011”

Cada *dataset* posee las siguientes columnas:

1. Código de la carrera
2. Sexo (‘F’ = Femenino / ‘M’ = Masculino)
3. Código de la modalidad de ingreso
4. Escala de pago
5. Tipo del colegio de procedencia (‘P’ = Particular / ‘E’ = Estatal)
6. Nota (o promedio) final del curso pre-requisito
7. Número de veces que estuvo matriculado en el curso pre-requisito para poder aprobarlo, incluyendo la ocasión donde lo aprobó.
8. Diferencia estandarizada entre la fecha de nacimiento y el periodo académico que indica cuando aprobó el curso pre-requisito
9. Primera Nota obtenida en el curso actual
10. Número de veces que estuvo matriculado en el curso actual para poder aprobarlo, incluyendo la ocasión donde lo aprobó.
11. Diferencia estandarizada entre la fecha de nacimiento y el periodo académico que indica cuando llevó el curso actual
12. Estado del rendimiento académico del curso actual: (‘Si’ = Aprobó / ‘No’ = Desaprobó), es la variable dependiente.

A continuación, mostramos los primeros 10 registros de cada *dataset* desde el curso “0006” hasta el curso “0011”:

Tabla 33: Visualización de los primeros registros del *dataset* del curso "0006".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	PRA1_2	veces_2	sem_dif_2	**Aprobó**
11	M	6	A23	P	14	1	12	17	1	15	**Si**
33	F	6	A33	P	16	1	12	20	1	11	**Si**
11	F	6	A23	P	14	1	15	14	1	14	**Si**
11	F	6	A23	P	12	1	12	15	1	11	**Si**
11	F	6	A23	P	14	1	12	13	1	11	**Si**
11	M	6	A23	E	14	1	15	6	1	14	**Si**
11	F	6	A23	P	13	1	12	5	1	16	**Si**
11	F	6	A23	P	14	1	12	9	1	11	**Si**
11	F	6	A23	P	12	1	12	12	1	13	**Si**
11	M	6	A23	P	14	1	12	11	1	11	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 34: Visualización de los primeros registros del *dataset* del curso "0007".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	PRA1_2	veces_2	sem_dif_2	**Aprobó**
11	M	6	A23	P	15	1	12	7	2	11	**No**
11	M	6	A23	P	15	1	12	10	2	13	**Si**
33	F	6	A33	P	15	1	12	10	1	11	**Si**
11	F	6	A23	P	15	1	15	11	1	14	**Si**
11	F	6	A23	P	17	1	12	14	1	11	**Si**
11	F	6	A23	P	18	1	12	10	1	11	**Si**
11	M	6	A23	E	15	1	15	15	1	14	**Si**
11	F	6	A23	P	14	1	12	13	1	11	**Si**
11	F	6	A23	P	15	1	12	7	1	11	**Si**
11	M	6	A23	P	15	1	12	7	1	11	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 35: Visualización de los primeros registros del *dataset* del curso "0008".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	PRA1_2	veces_2	sem_dif_2	**Aprobó**
11	M	6	A23	P	14	1	12	9	1	17	**Si**
11	F	6	A23	P	14	1	15	18	1	20	**Si**
11	F	6	A23	P	12	1	12	14	1	17	**Si**
11	F	6	A23	P	14	1	12	11	1	17	**Si**
11	M	6	A23	E	14	1	15	14	1	20	**Si**
11	F	6	A23	P	13	1	12	12	1	16	**Si**
11	F	6	A23	P	14	1	12	11	1	17	**Si**
11	F	6	A23	P	12	1	12	12	1	16	**Si**
11	M	6	A23	P	14	1	12	13	1	17	**Si**
41	F	6	A33	P	11	1	12	10	3	17	**No**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 36: Visualización de los primeros registros del *dataset* del curso "0009".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	PRA1_2	veces_2	sem_dif_2	**Aprobó**
11	M	6	A23	P	15	1	18	15	1	17	**Si**
11	F	6	A23	P	15	1	22	16	1	22	**Si**
11	F	6	A23	P	15	1	19	13	1	19	**Si**
11	F	6	A23	P	16	1	19	14	1	19	**Si**
11	M	6	A23	E	17	1	22	14	1	22	**Si**
11	F	6	A23	P	15	1	18	11	1	17	**Si**
11	F	6	A23	P	19	1	19	20	1	19	**Si**
11	M	6	A23	P	12	1	19	8	1	18	**Si**
41	F	6	A33	P	16	1	18	11	1	17	**Si**
61	M	6	A23	P	15	1	21	8	1	20	**No**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 37: Visualización de los primeros registros del *dataset* del curso "0010".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	PRA1_2	veces_2	sem_dif_2	**Aprobó**
11	M	6	A23	P	15	1	12	15	1	11	**Si**
33	F	6	A33	P	15	1	12	16	1	11	**Si**
11	F	6	A23	P	15	1	15	14	1	14	**Si**
11	F	6	A23	P	17	1	12	17	1	11	**Si**
11	F	6	A23	P	18	1	12	20	1	11	**Si**
11	M	6	A23	E	15	1	15	8	1	14	**Si**
11	F	6	A23	P	14	1	12	9	1	11	**Si**
11	F	6	A23	P	15	1	12	14	1	11	**Si**
11	F	6	A23	P	13	1	12	16	1	11	**Si**
11	M	6	A23	P	15	1	12	20	1	11	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 38: Visualización de los primeros registros del *dataset* del curso "0011".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	PRA1_2	veces_2	sem_dif_2	**Aprobó**
11	M	6	A23	P	14	1	12	15	1	12	**Si**
33	F	6	A33	P	16	1	12	15	1	12	**Si**
11	F	6	A23	P	18	1	15	6	1	15	**Si**
11	F	6	A23	P	16	1	12	15	1	12	**Si**
11	F	6	A23	P	14	1	12	14	1	12	**Si**
11	M	6	A23	E	17	1	15	10	1	15	**Si**
11	F	6	A23	P	14	1	12	9	1	10	**Si**
11	F	6	A23	P	15	1	12	8	1	10	**Si**
11	F	6	A23	P	14	1	12	15	1	10	**Si**
11	M	6	A23	P	16	1	12	9	1	11	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

C. Estructura del *dataset* del curso "0012" y el curso "0013"

Cada *dataset* posee las siguientes columnas:

1. Código de la carrera
2. Sexo ('F' = Femenino / 'M' = Masculino)
3. Código de la modalidad de ingreso
4. Escala de pago
5. Tipo del colegio de procedencia ('P' = Particular / 'E' = Estatal)
6. Nota (o promedio) final del requisito del curso pre-requisito
7. Número de veces que estuvo matriculado en el requisito del curso pre-requisito para poder aprobarlo, incluyendo la ocasión donde lo aprobó.

8. Diferencia estandarizada entre la fecha de nacimiento y el periodo académico que indica cuando aprobó el requisito del curso pre-requisito

9. Nota (o promedio) final del curso pre-requisito

10. Número de veces que estuvo matriculado en el curso pre-requisito para poder aprobarlo, incluyendo la ocasión donde lo aprobó.

11. Diferencia estandarizada entre la fecha de nacimiento y el periodo académico que indica cuando aprobó el curso pre-requisito

12. Primera Nota obtenida en el curso actual

13. Número de veces que estuvo matriculado en el curso actual para poder aprobarlo, incluyendo la ocasión donde lo aprobó.

14. Diferencia estandarizada entre la fecha de nacimiento y el periodo académico que indica cuando llevó el curso actual

15. Estado del rendimiento académico del curso actual: ('Si' = Aprobó / 'No' = Desaprobó), es la variable dependiente.

A continuación, mostramos los primeros 10 registros del *dataset* correspondiente al curso "0012" y el curso "0013":

Tabla 39: Visualización de los primeros registros del *dataset* del curso "0012".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	eva_fin_2	veces_2	sem_dif_2	PRA1_3	veces_3	sem_dif_3	**Aprobó**
11	M	6	A23	P	15	1	12	14	1	11	14	1	12	**Si**
33	F	6	A33	P	15	1	12	14	1	11	9	1	12	**Si**
11	F	6	A23	P	15	1	15	13	1	14	14	1	14	**Si**
11	F	6	A23	P	17	1	12	16	1	11	15	1	12	**Si**
11	F	6	A23	P	18	1	12	12	1	11	13	1	12	**Si**
11	M	6	A23	E	15	1	15	12	1	14	14	1	15	**Si**
11	F	6	A23	P	14	1	12	13	1	11	14	1	12	**Si**
11	F	6	A23	P	15	1	12	14	1	11	15	1	12	**Si**
11	F	6	A23	P	13	1	12	17	1	11	15	1	11	**Si**
11	M	6	A23	P	15	1	12	16	1	11	13	1	12	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 40: Visualización de los primeros registros del *dataset* del curso "0013".

car_cod	sexo	ing_cod	escala	col_tipo	eva_fin_1	veces_1	sem_dif_1	eva_fin_2	veces_2	sem_dif_2	PRA1_3	veces_3	sem_dif_3	**Aprobó**
11	M	6	A23	P	15	1	12	14	1	11	15	1	13	**Si**
33	F	6	A33	P	15	1	12	14	1	11	13	1	13	**Si**
11	F	6	A23	P	15	1	15	13	1	14	11	1	15	**Si**
11	F	6	A23	P	17	1	12	16	1	11	17	1	13	**Si**
11	F	6	A23	P	18	1	12	12	1	11	20	1	13	**Si**
11	M	6	A23	E	15	1	15	12	1	14	14	1	16	**Si**
11	F	6	A23	P	14	1	12	13	1	11	16	1	13	**Si**
11	F	6	A23	P	15	1	12	14	1	11	16	1	13	**Si**
11	F	6	A23	P	13	1	12	17	1	11	14	1	13	**Si**
11	M	6	A23	P	15	1	12	16	1	11	12	1	13	**Si**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

D. Deducción

Se puede notar que, a pesar de ser 13 *datasets*, sus características internas permiten que se concentren en 3 grupos. El procedimiento y las instrucciones que se aplicaron a cada uno de los 13 *datasets* es exactamente igual, pero las características internas de todos los *datasets* que pertenecen a un determinado grupo deberían ser similares entre sí; podemos tomar como ejemplo que, 13 es la cantidad de columnas de cada *dataset*, preparado para el modelado, desde el curso "0001" hasta el curso "0005".

4.1.3. Modelado.

Como resultado de la etapa de preparación de datos se obtuvo 13 *datasets*, cada uno de los cuales corresponde a un curso del Programa de Estudios Básicos, a los cuales se les implementó diferentes técnicas de modelado para elaborar patrones de predicción que al ser evaluados mediante indicadores se pudo determinar cuál fue la mejor técnica de modelado para cada uno de los cursos.

4.1.3.1. Técnicas de modelado.

En cada *dataset* se aplicaron las siguientes:

A. Red Neuronal Artificial (RNA)

B. *Gradient Boosting Machine* (GBM)

C. *XGBoosting*

D. Ensamble

E. *Stacking*

4.1.3.2. Parámetros de las técnicas de modelado.

Los parámetros en cada técnica de modelado se describen a continuación:

A. Red Neuronal Artificial (RNA)

Dada la amplitud de librerías sobre el tema de redes neuronales artificiales, se tuvo que decidir por la librería denominada "nnet", que es de las más empleadas, en la cual se utilizaron los siguientes parámetros:

1. Número de neuronas ocultas (size)

   Determina la cantidad de neuronas que se van a utilizar en una única capa oculta, que pueden variar de de 0 a "*n*". Para determinar la cantidad de pesos se usa la fórmula:

$$(i \times h) + h + (h \times o) + o$$

   donde:

   i son las neuronas de entrada;
   h son las neuronas ocultas;
   o es la salida

   Si por ejemplo, tenemos 4 datos de entrada, dos nodos ocultos y tres salidas, entonces aplicando la formula obtendremos 19 pesos (4 * 2 + 2 + 2 * 3 + 3).

2. Decay

   Afecta el algoritmo de optimización que evita un sobreajuste del modelo, su valor óptimo se encuentra entre 0 y 1 (también se indica que podría ser 0.5). También es conocido como la penalización de la tasa de aprendizaje.

3. Linout

   Si el parámetro es Falso, la función de activación que se seleccionará será la función logística, caso contrario será la función de identidad (Regresión).

4. Trace

   Evitamos la visualización de las iteraciones cuando el valor es Falso.

Lo principal en el uso de la librería "nnet", es su facilidad de manejo en contraprestación con la rigidez que demuestra en, no poder seleccionar la función de activación, que si bien es cierto, se realiza con la función logística no puede escogerse la función tangente hiperbólica, a pesar de que ambas pertenecen a la clase de funciones sigmoideas; en, no poder seleccionar un algoritmo de entrenamiento diferente a BFGS (algoritmo de Broyden–Fletcher–Goldfarb–Shanno), como podría ser el *backpropagation*; y en, no poder definir más de 1 capa de neuronas ocultas.

B. *Gradient Boosting Machine* (GBM)

Se utilizó la librería denominada "gbm" (*Generalized Boosted regression Modeling*) y se usaron los siguientes parámetros:

1. Número de árboles (n.trees)

   También es conocido como el número de iteraciones. El incrementar dicho número tiene como consecuencia la disminución del error, pero, puede originar un sobreajuste del modelo.

2. Profundidad de cada árbol (Interaction.depth)

   Altura o cantidad de niveles del árbol (a partir de un tronco principal o nivel cero - 0). A mayor número de divisiones, mayor la cantidad total de nodos (vértices o elementos del árbol) y mayor la cantidad de nodos terminales, los cuales se siguen por las siguientes formulas:

$$Nodos_{Arbol} = 2^{n+1} - 1$$

$$Nodos_{Terminal} = 2^{n}$$

   donde:

   n es el nivel de la división, considerando desde el nivel 0.

3. Tasa de Aprendizaje o *Learning Rate* (*Shrinkage*)

   Su valor varía de 0 a 1, su valor óptimo se encuentra entre 0.01 y 0.3. Un valor alrededor de 1.0 puede originar un sobreajuste del modelo, pero, un valor menor que 0.01 tiene como consecuencia un modelo con un paulatino aprendizaje.

Generalmente, se define el parámetro con anterioridad y se van variando la cantidad de árboles, teniendo en consideración que, con un valor pequeño se obtiene mejores resultados, pero, se debe utilizar un elevado número de árboles.

4. Mínimo número de observaciones en el nodo hijo o terminal (n.minobsinnode)

   El valor por defecto es 10, a menos que la muestra de entrenamiento sea ínfima, entonces puede variar dicho valor a 5 o 3.

5. Función de perdida (Distribution)

   Al tratarse de un problema de clasificación la función a utilizar será Bernoulli o Adaboost, siendo más apropiada la primera de ellas.

6. Verbose

   Evitamos la visualización de las iteraciones cuando el valor es Falso.

C. *XGBoosting*

Se utilizó la librería denominada "xgboost" (*eXtreme Gradient BOOSTing*), la cual mediante la implementación de la computación en paralelo es más rápida que otros algoritmos *Boosting* y en la cual se emplearon los siguientes parámetros:

1. Número de árboles (nrounds)

   También es conocido como el número de iteraciones. El incrementar dicho número tiene como consecuencia la disminución del error, pero, puede originar un sobreajuste del modelo.

2. Profundidad de cada árbol (max_depth)

   Altura o cantidad de niveles del árbol (a partir de un tronco principal o nivel cero - 0). A mayor número de divisiones, mayor la cantidad total de nodos y mayor la cantidad de nodos terminales.

3. Tasa de Aprendizaje o *Learning Rate* (eta)

   Su valor varía de 0 a 1, su valor óptimo se encuentra entre 0.01 y 0.3. Un valor alrededor de 1.0 puede originar un sobreajuste del modelo, pero, un valor menor

que 0.01 tiene como consecuencia un modelo con un paulatino aprendizaje. Generalmente, se define el parámetro con anterioridad y se van variando la cantidad de árboles, teniendo en consideración que, con un valor pequeño se obtiene mejores resultados, pero, se debe utilizar un elevado número de árboles.

4. Tasa de regularización o de penalización (gamma)

   Su valor varía desde 0 (sin regularización) a $\propto$, el incrementar la tasa tendrá como consecuencia un modelo más fortificado contra el sobreajuste.

5. Columnas en cada árbol (colsample_bytree)

   Su valor varía de 0 a 1, su valor óptimo se encuentra entre 0.5 y 0.9, fiscaliza la cantidad de columnas (características) proporcionadas a cada árbol.

6. Peso de los hijos (min_child_weight)

   Si el peso del nodo padre es mayor que la cantidad determinada en este parámetro, entonces, no se realiza la división para evitar un sobreajuste del modelo.

7. Porcentaje de la muestra (subsample)

   Su valor varía de 0 a 1, su valor óptimo se encuentra entre 0.5 y 0.8, fiscaliza la cantidad de filas (observaciones) proporcionadas a cada árbol

8. Función de perdida (Objective)

   Al tratarse de un problema de clasificación la función a utilizar será "binary:logistic" (regresión logística para clasificación binaria), la cual retorna las probabilidades de clase.

9. Evaluación del modelo (eval_metric)

   Al tratarse de un problema de clasificación, la métrica que se utilizara es "error" (Tasa de error de clasificación binaria), la cual está definida como la cantidad de casos erróneos dividido por la cantidad total de casos.

10. Verbose

    Evitamos la visualización de las iteraciones cuando el valor es 0.

D. Ensamble

   Es una técnica democrática, porque considera los resultados de las predicciones de todas las técnicas de modelado que la conforman. Al tratarse de una clasificación, debemos combinar las probabilidades de las predicciones de cada técnica de modelado, pudiendo escoger entre el promedio aritmético simple o el promedio ponderado.

E. *Stacking*

   Es una técnica de combinación, porque los resultados de las predicciones de todas las técnicas de modelado que la conforman son utilizados como información de ingreso para el metamodelo (que es en realidad otra técnica de modelado), el cual brindará como salida la predicción definitiva del modelo.

F. Interfaz de trabajo

   Se utilizó el paquete Caret (*Classification And REgression Training*), el cual es una librería concentradora, porque se tiene acceso a múltiples métodos de regresión, clasificación y ensamble mediante una única instrucción, de modo tal, que al invocar el método en conjunto con sus hiperparámetros, "caret" se encarga de llamar a la librería del método, transferir los valores de los hiperparámetros y devolver el modelo final; adicionalmente permite la transformación, estandarización y/o normalización de los datos así como la división de los datos en muestras de entrenamiento y evaluación y la respectiva certificación con el uso de diferentes técnicas de validación cruzada.

4.1.3.3. Procedimiento.

Para una mejor comprensión de la secuencia adoptada, la explicación está basada en el curso del PEB denominado: "Actividades Artísticas y Deportivas" con código "0001" y con su *dataset* denominado "DS_0001.csv".

Como cuestión previa debemos señalar que, se ha seleccionado al indicador "*Accuracy*" como medida de calidad que nos brindara información sobre el grado de exactitud que tiene el modelo con respecto a las respuestas verdaderas o correctas.

A. Selección de las mejores variables

Se realiza una selección de las variables más representativas en vez de trabajar con la totalidad de las variables, con la finalidad de hacer más simple el modelo (principio de parsimonia o navaja de *Ockham* u *Occam* o innovación progresiva).

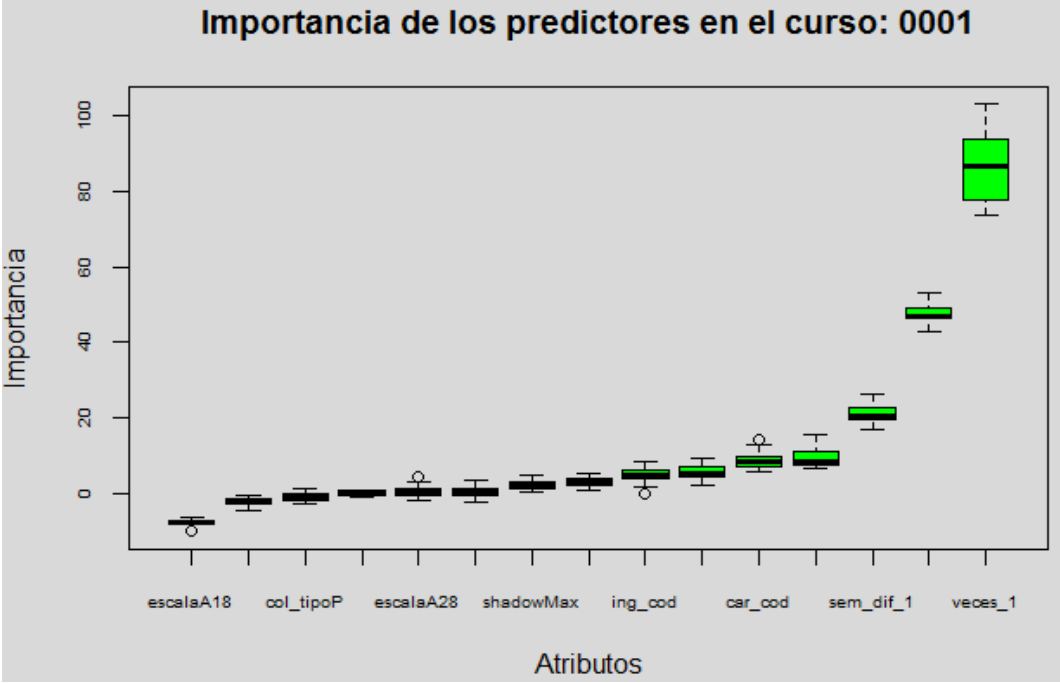


Figura 30: Visualización de resultados de la selección de las variables predictoras del *dataset* del curso "0001".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

B. Optimización de los hiperparámetros (*Tuning* del modelo)

A continuación, se detalla las diversas combinaciones de hiperpárametros utilizados en la configuración de cada una de las técnicas de modelado, las cuales se implementaron en la muestra de entrenamiento "train", la cual fue objeto de una validación cruzada de 10 iteraciones en 3 ocasiones y la estandarización de sus datos numéricos.

1. Red Neuronal Artificial (RNA)

Para cada prueba, se definió con valor Falso los parámetros "trace" y "linout".

En la primera prueba los hiperparámetros para “size” y “decay” no se utilizaron, y se obtuvo los siguientes resultados:

Tabla 41: Visualización de resultados de la primera prueba con la librería “nnet” en la muestra de entrenamiento “train” del *dataset* del curso “0001”.

Prueba Nº 1		
size	decay	*Accuracy*
1	0.0000	0.7171062
1	0.0001	0.6944567
1	0.1000	0.8400778
3	0.0000	0.8120346
3	0.0001	0.8186633
3	0.1000	0.8508528
5	0.0000	0.8534493
5	0.0001	0.8189761
5	0.1000	0.8509835

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la segunda prueba se indicó que los hiperparámetros “size” y “decay” debían tener 5 valores aleatorios cada uno, por lo tanto, la propia librería selecciono dichos valores, y se obtuvo los siguientes resultados:

Tabla 42: Visualización de resultados de la segunda prueba con la librería “nnet” en la muestra de entrenamiento “train” del *dataset* del curso “0001”.

Prueba Nº 2		
size	decay	*Accuracy*
1	0.0000	0.8518515
1	0.0001	0.8558298
1	0.0010	0.8603664
	•••	
5	0.0001	0.8591691
5	0.0010	0.8596376
5	0.0100	0.8594055
	•••	
9	0.0010	0.8526981
9	0.0100	0.8528313
9	0.1000	0.8567774

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la tercera prueba el número de neuronas en “size” iba a variar entre 3 y 10, y el hiperparámetro “decay” iba a considerar los valores de 0.1, 0.2, 0.3, 04 y 0.5; se obtuvo los siguientes resultados:

Tabla 43: Visualización de resultados de la tercera prueba con la librería "nnet" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 3		
size	decay	*Accuracy*
3	0.1000	0.8529525
3	0.2000	0.8473286
	•••	
6	0.4000	0.8460127
6	0.5000	0.8466100
7	0.1000	0.8536664
	•••	
10	0.1000	0.8564139
10	0.2000	0.8491242
10	0.3000	0.8469685
10	0.4000	0.8467291
10	0.5000	0.8462503

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la cuarta prueba el número de neuronas en "size" iba a variar entre 1 y 3, y el hiperparámetro "decay" iba a considerar los valores de 0, 0.05 y 0.005; se obtuvo los siguientes resultados:

Tabla 44: Visualización de resultados de la cuarta prueba con la librería "nnet" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 4		
size	decay	*Accuracy*
1	0.0000	0.8579799
1	0.0050	0.8576150
1	0.0100	0.8572562
	•••	
2	0.0450	0.8537850
2	0.0500	0.8539062
3	0.0000	0.8591627
3	0.0050	0.8589306
	•••	
3	0.0400	0.8523560
3	0.0450	0.8542646
3	0.0500	0.8521188

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede observar que el mejor *Accuracy* se obtuvo en la segunda prueba.

2. *Gradient Boosting Machine* (GBM)

Para cada prueba, se definió al parámetro "verbose" con valor Falso y al parámetro "distribution" se definió con la función de perdida llamada Bernoulli.

En la primera prueba ningún hiperparámetros se utilizó, y se obtuvo los siguientes resultados:

Tabla 45: Visualización de resultados de la primera prueba con la librería "gbm" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 1				
shrinkage	interaction.depth	n.minobsinnode	n.trees	*Accuracy*
0.1	1	10	50	0.8683245
0.1	2	10	50	0.8727839
0.1	3	10	50	0.8866854
0.1	1	10	100	0.8712656
0.1	2	10	100	0.8849687
0.1	3	10	100	0.8967507
0.1	1	10	150	0.8738615
0.1	2	10	150	0.8926291
0.1	3	10	150	0.9037760

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la segunda prueba se indicó que todos los hiperparámetros debían tener 2 valores aleatorios cada uno, por lo tanto, la propia librería selecciono dichos valores, y se obtuvo los siguientes resultados:

Tabla 46: Visualización de resultados de la segunda prueba con la librería "gbm" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 2				
shrinkage	interaction.depth	n.minobsinnode	n.trees	*Accuracy*
0.1	1	10	50	0.8730462
0.1	2	10	50	0.8768763
0.1	1	10	100	0.8732886
0.1	2	10	100	0.8902793

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la tercera prueba el número de árboles en "n.trees" iba a considerar los valores de 50 y 51, el hiperparámetro "interaction.depth" iba a considerar los valores de 2, 3 y 4 para cada árbol; la tasa de aprendizaje "shrinkage" se definió con el valor de 0.1; y el mínimo número de observaciones en el nodo hijo "n.minobsinnode" se seleccionó en 10; se obtuvo los siguientes resultados:

Tabla 47: Visualización de resultados de la tercera prueba con la librería "gbm" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 3				
shrinkage	interaction.depth	n.minobsinnode	n.trees	*Accuracy*
0.1	2	10	50	0.8774719
0.1	3	10	50	0.8906382
0.1	4	10	50	0.8969802
0.1	2	10	51	0.8773524
0.1	3	10	51	0.8906390
0.1	4	10	51	0.8969806

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la cuarta prueba el número de árboles en "n.trees" iba a variar entre 140 y 160, el hiperparámetro "interaction.depth" iba a considerar los valores de 5, 6, 7 y 8 para cada árbol; la tasa de aprendizaje "shrinkage" se definió con el valor de 0.05 y el mínimo número de observaciones en el nodo hijo "n.minobsinnode" se seleccionó en 10; se obtuvo los siguientes resultados:

Tabla 48: Visualización de resultados de la cuarta prueba con la librería "gbm" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 4				
shrinkage	interaction.depth	n.minobsinnode	n.trees	Accuracy
0.05	5	10	140	0.9083475
0.05	6	10	140	0.9111023
0.05	7	10	140	0.9113387
		•••		
0.05	8	10	149	0.9131334
0.05	5	10	150	0.9085882
0.05	6	10	150	0.9116992
		•••		
0.05	6	10	160	0.9126550
0.05	7	10	160	0.9134944
0.05	8	10	160	0.9151704

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede observar que el mejor *Accuracy* se obtuvo en la cuarta prueba.

3. *XGBoosting*

Para cada prueba, se definió al parámetro "verbose" con valor 0, al parámetro "objective" se definió con la función de pérdida denominada "binary:logistic" y "eval_metric" se definió con "error".

En la primera prueba ningún hiperparámetros se utilizó, se obtuvo los siguientes resultados:

Tabla 49: Visualización de resultados de la primera prueba con la librería "xgboost" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 1							
eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	*Accuracy*
0.3	1	0	0.6	1	0.50	50	0.8734541
0.3	1	0	0.6	1	0.75	50	0.8727650
				•••			
0.3	2	0	0.6	1	1.00	100	0.9101494
0.3	2	0	0.8	1	0.50	100	0.9085746
				•••			
0.3	3	0	0.8	1	1.00	150	0.9172731
0.4	3	0	0.6	1	0.50	150	0.9129825
0.4	3	0	0.6	1	0.75	150	0.9147165
0.4	3	0	0.6	1	1.00	150	0.9154129
0.4	3	0	0.8	1	0.50	150	0.9131237
0.4	3	0	0.8	1	0.75	150	0.9135138
0.4	3	0	0.8	1	1.00	150	0.9159132

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la segunda prueba se indicó que todos los hiperparámetros debían tener 2 valores aleatorios cada uno, por lo tanto, la propia librería selecciono dichos valores, y se obtuvo los siguientes resultados:

Tabla 50: Visualización de resultados de la segunda prueba con la librería "xgboost" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Prueba Nº 2							
eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	*Accuracy*
0.3	1	0	0.6	1	0.50	50	0.8767619
0.3	1	0	0.6	1	1.00	50	0.8744868
0.3	1	0	0.8	1	0.50	50	0.8762793
				•••			
0.4	2	0	0.8	1	1.00	50	0.9109768
0.3	1	0	0.6	1	0.50	100	0.8875318
				•••			
0.4	2	0	0.6	1	1.00	100	0.9194715
0.4	2	0	0.8	1	0.50	100	0.9146848
0.4	2	0	0.8	1	1.00	100	0.9200723

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la cuarta prueba el número de árboles en “nrounds” iba a variar entre 48 y 51, el hiperparámetro “max_depth” iba a considerar los valores de 1, 2 y 3 para cada árbol; la tasa de aprendizaje “eta” se definió con el valor de 0.3; la tasa de regularización “gamma” se definió con el valor de 0; el hiperparámetro “colsample_bytree” se definió con el valor de 0.8; el peso de los hijos “min_child_weight” se definió con el valor de 1; y el porcentaje de la muestra “subsample” se seleccionó en 1; se obtuvo los siguientes resultados:

Tabla 51: Visualización de resultados de la tercera prueba con la librería “xgboost” en la muestra de entrenamiento “train” del *dataset* del curso “0001”.

Prueba Nº 3							
eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	*Accuracy*
0.3	1	0	0.8	1	1.00	48	0.8750846
0.3	2	0	0.8	1	1.00	48	0.9052377
0.3	3	0	0.8	1	1.00	48	0.9155302
0.3	1	0	0.8	1	1.00	49	0.8746062
0.3	2	0	0.8	1	1.00	49	0.9058342
0.3	3	0	0.8	1	1.00	49	0.9155289
0.3	1	0	0.8	1	1.00	50	0.8753248
0.3	2	0	0.8	1	1.00	50	0.9054767
0.3	3	0	0.8	1	1.00	50	0.9155297
0.3	1	0	0.8	1	1.00	51	0.8749655
0.3	2	0	0.8	1	1.00	51	0.9059554
0.3	3	0	0.8	1	1.00	51	0.9157678

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Para la tercera prueba el número de árboles en “nrounds” iba a variar entre 52 y 55, el hiperparámetro “max_depth” iba a variar entre 4 y 8 para cada árbol; la tasa de aprendizaje “eta” se definió con el valor de 0.2; la tasa de regularización “gamma” se definió con el valor de 0; el hiperparámetro “colsample_bytree” se definió con el valor de 0.8; el peso de los hijos “min_child_weight” se definió con el valor de 1; y el porcentaje de la muestra “subsample” se seleccionó en 1; se obtuvo los siguientes resultados:

Tabla 52: Visualización de resultados de la cuarta prueba con la librería “xgboost” en la muestra de entrenamiento “train” del *dataset* del curso “0001”.

Prueba Nº 4							
eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	*Accuracy*
0.2	4	0	0.8	1	1.00	52	0.9189949
0.2	5	0	0.8	1	1.00	52	0.9241444

0.2	6	0	0.8	1	1.00	52	0.9260585
				•••			
0.2	8	0	0.8	1	1.00	53	0.9279701
0.2	4	0	0.8	1	1.00	54	0.9193538
				•••			
0.2	6	0	0.8	1	1.00	55	0.9271342
0.2	7	0	0.8	1	1.00	55	0.9274935
0.2	8	0	0.8	1	1.00	55	0.9283294

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede observar que el mejor *Accuracy* se obtuvo en la cuarta prueba.

4. Ensamble

   En su construcción se seleccionaron las técnicas de modelado: Red Neuronal Artificial (RNA), *Gradient Boosting Machine* (GBM) y *XGBoosting* con sus parámetros de configuración de mejor performance entre las diferentes pruebas efectuadas en los puntos anteriores.

5. *Stacking*

   Para su construcción se seleccionaron las técnicas de modelado: Red Neuronal Artificial (RNA), *Gradient Boosting Machine* (GBM) y *XGBoosting*.

   En cada prueba se seleccionó como metamodelo a la técnica de modelado denominada regresión logística, porque es la más habitual y simple de utilizar.

   Se realizaron dos pruebas: la primera, donde los hiperparámetros de cada técnica de modelado debían tener valores aleatorios, por lo tanto, la propia librería selecciono dichos valores; la segunda, donde se utilizaron los valores de mejor performance en cada técnica de modelado obtenidos en los puntos anteriores.

Tabla 53: Visualización de resultados de las pruebas con el método "*Stacking*" en la muestra de entrenamiento "train" del *dataset* del curso "0001".

	Exactitud obtenida
Prueba 1	0.9195792
Prueba 2	0.9221963

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede observar que el mejor *Accuracy* se obtuvo en la segunda prueba.

Se debe tener en cuenta que, para cada *dataset*, se tiene que implementar las respectivas pruebas por cada técnica de modelado con la finalidad de obtener la mejor versión de cada una de ellas. Debemos tener en consideración que los valores de los hiperparámetros para cada técnica de modelado están relacionados con las particularidades propias de cada *dataset*.

Los comandos utilizados en esta sección y sus respectivos resultados se encuentran en el Anexo 09 de la presente investigación.

C. Elección de parámetros de cada técnica de modelado

A continuación, se muestran los valores del indicador "*Accuracy*" en la muestra de entrenamiento "train" y en la muestra de evaluación "test" que fueron obtenidos por las diferentes pruebas de cada técnica de modelado en el curso del PEB denominado "Actividades Artísticas y Deportivas".

Tabla 54: Exactitud obtenida por los modelos de prueba de cada técnica de modelado implementada en la muestra de entrenamiento "train" y en la muestra de evaluación "test" del curso "0001".

	Exactitud obtenida en el Train			**Exactitud obtenida en el Test**		
	RNA	GBM	*XGBoosting*	RNA	GBM	*XGBoosting*
Prueba 1	0.8534493	0.9037760	0.9172731	0.8461538	0.8734869	0.8973057
Prueba 2	0.8603664	0.8902793	0.9200723	0.8438110	0.8578680	0.8906677
Prueba 3	0.8564139	0.8969806	0.9157678	0.8379539	0.8594299	0.8887153
Prueba 4	0.8591627	0.9151704	0.9283294	0.8426396	0.8797345	0.8797345

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Observando la tabla podemos concluir que la mejor técnica de modelado fue el *XGBoosting*, a mayor abundamiento debemos indicar que, los parámetros de cada técnica de modelado se seleccionan basándose en el mejor valor del indicador "*Accuracy*" en la muestra de evaluación "test".

1. Red Neuronal Artificial (RNA)

Se puede observar que el mejor *Accuracy* en la muestra de evaluación "test" se obtuvo en la primera prueba, por lo tanto, se utilizaron dichos valores para definir los parámetros de la Red Neuronal Artificial definitiva, obteniendo lo siguiente:

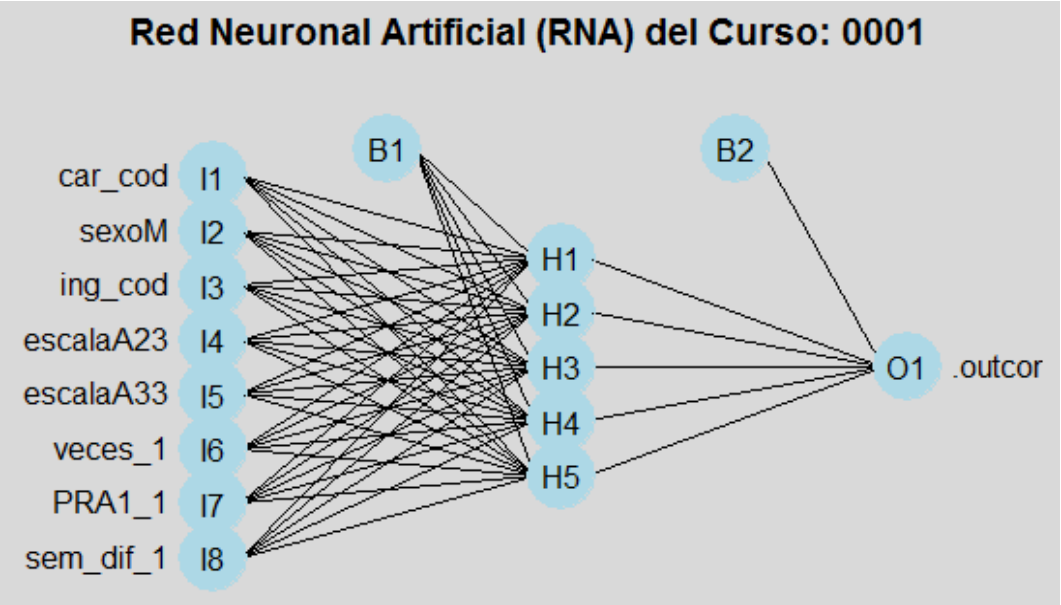


Figura 31: Grafico de la estructura de la Red Neuronal Artificial definitiva obtenida con la librería "nnet" en el *dataset* del curso "0001".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

En resumen, la Red Neuronal Artificial obtenida, es un perceptrón multicapa con una arquitectura 8-5-1 y 51 pesos, donde 5 es la cantidad de neuronas ocultas en una única capa; se definió como función de activación a la función logística y el algoritmo de aprendizaje es el método de optimización numérica denominado BFGS.

Los parámetros "trace" y "linout" fueron definidos con valor Falso; el parámetro "size" tiene el valor de 5; y el parámetro "decay", que es un mecanismo de regularización para evitar el sobreajuste, toma el valor de 0.

2. *Gradient Boosting Machine* (GBM)

Se puede observar que el mejor *Accuracy* en la muestra de evaluación "test" se obtuvo en la cuarta prueba, por lo tanto, se utilizaron dichos valores para definir los parámetros del *Gradient Boosting Machine* definitivo.

El parámetro "verbose" fue definido con valor Falso; el parámetro "distribution" se definió con la función de perdida Bernoulli; el parámetro "n.trees" tiene el valor de 160; el parámetro "interaction.depth" toma el valor de 8; el parámetro "shrinkage", que es la tasa de aprendizaje, acoge el valor de 0.05; y el parámetro "n.minobsinnode" se seleccionó en 10.

3. *XGBoosting*

Se puede observar que el mejor *Accuracy* en la muestra de evaluación "test" se obtuvo en la primera prueba, por lo tanto, se utilizaron dichos valores para definir los parámetros del *XGBoosting* definitivo.

El parámetro "verbose" fue definido con valor 0; el parámetro "objective" se definió con la función de perdida binary:logistic; el parámetro "eval_metric" se definió con "error"; el parámetro "nrounds" tiene el valor de 150; el parámetro "max_depth" toma el valor de 3; el parámetro "eta", que es la tasa de aprendizaje, acoge el valor de 0.3; el parámetro "gamma", que es la tasa de regularización, fue definido con el valor de 0; el parámetro "colsample_bytree", tiene el valor de 0.8; y los parámetro "min_child_weight" y "subsample" adoptaron el valor de 1.

4. Tabla de resumen

Tabla 55: Relación de los parámetros de cada técnica de modelado para la implementación en el *dataset* del curso "0001".

Nombre del Modelo	**Parámetros**	
	Nombre	Valor
Red Neuronal Artificial (RNA)	size	5
	decay	0
	trace	FALSO
	linout	FALSO
Gradient Boosting Machine (GBM)	shrinkage	0.05
	interaction.depth	8
	n.minobsinnode	10
	n.trees	160
	verbose	FALSO
	distribution	Bernoulli
XGBoosting	eta	0.3
	max_depth	3
	gamma	0
	colsample_bytree	0.8
	min_child_weight	1
	subsample	1
	nrounds	150
	verbose	0
	objective	binary:logistic

	eval_metric	error

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

5. Importancia de las variables

Tabla 56: Importancia de las variables en cada técnica de modelado en la muestra de entrenamiento "train" del *dataset* del curso "0001".

Variable RNA	% RNA	Variable GBM	% GBM	Variable XGB	% XGB
veces_1	23.27%	veces_1	16.88%	veces_1	48.30%
car_cod	21.42%	PRA1_1	16.00%	PRA1_1	25.73%
sem_dif_1	15.13%	sem_dif_1	11.65%	sem_dif_1	10.28%
ing_cod	13.56%	sexoM	11.63%	car_cod	8.61%
PRA1_1	11.46%	car_cod	11.54%	ing_cod	5.80%
escalaA33	8.97%	escalaA33	10.92%	sexoM	0.49%
escalaA23	3.72%	ing_cod	10.79%	escalaA33	0.49%
sexoM	2.47%	escalaA23	10.59%	escalaA23	0.31%

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

En el cuadro anterior se visualiza la importancia de cada característica dentro de las técnicas de modelado: Red Neuronal Artificial (RNA), *Gradient Boosting Machine* (GBM) y *XGBoosting*; pudiéndose apreciar que la variable "veces_1" (número de veces que estuvo matriculado en el curso actual para poder aprobarlo, incluyendo la ocasión donde lo aprobó) es la más influyente seguida de "PRA1_1" (primera nota obtenida en el curso actual) y de "sem_dif_1" (diferencia estandarizada entre la fecha de nacimiento y el periodo académico que indica cuando aprobó el curso pre-requisito).

6. Interpretabilidad del modelo

A continuación procederemos a interpretar las variables de la técnica de modelado *XGBoosting* en la muestra de entrenamiento "train" del curso "Actividades Artísticas y Deportivas"; a mayor abundamiento debemos indicar que las tres primeras variables representan en conjunto una importancia del 85%.

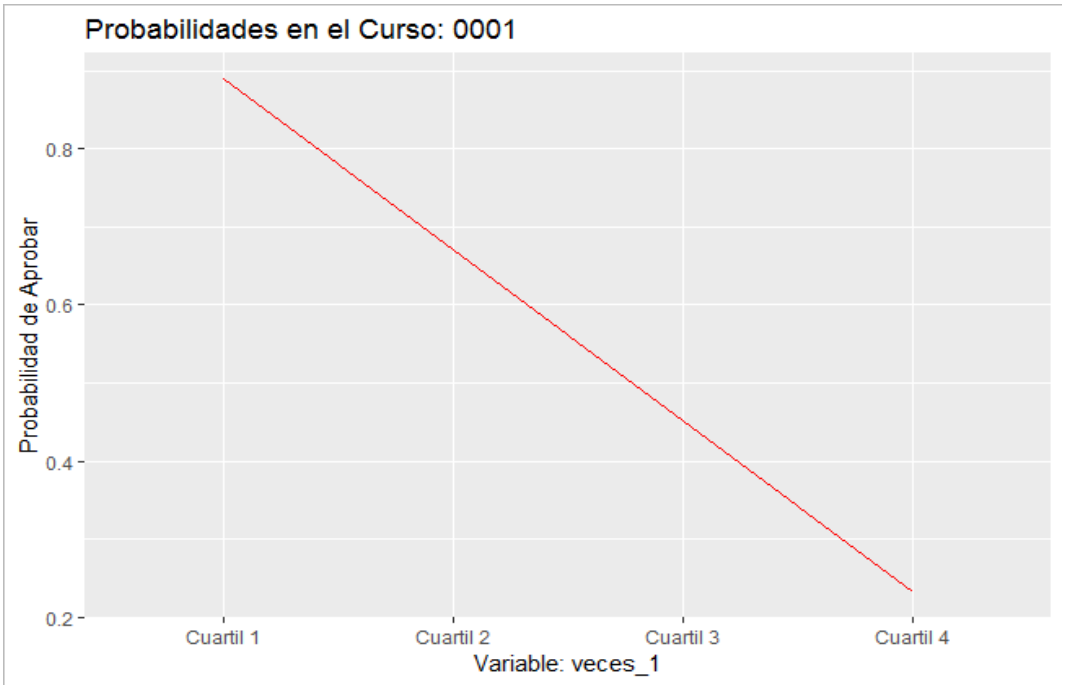


Figura 32: Interpretabilidad de la variable denominada "veces_1" en la muestra de entrenamiento "train" del curso "0001", según la librería "xgboost".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede apreciar que, el alumno posee mayor probabilidad de salir aprobado mientras se encuentre inscrito en el curso la menor cantidad de veces.

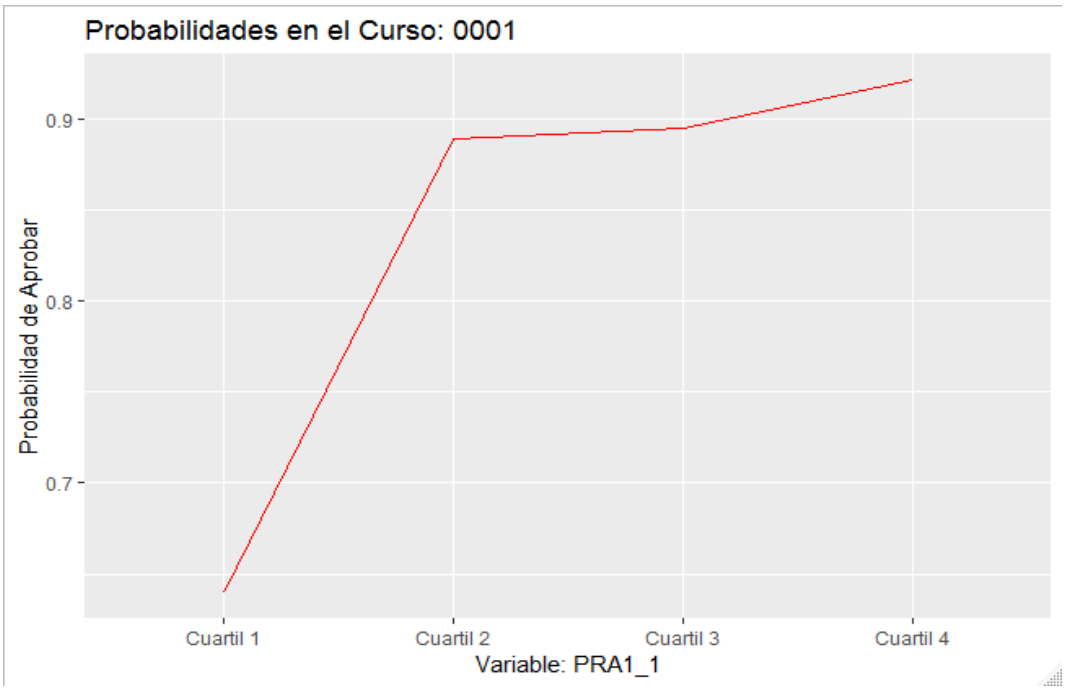


Figura 33: Interpretabilidad de la variable denominada "PRA1_1" en la muestra de entrenamiento "train" del curso "0001", según la librería "xgboost".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se dedujo que, una mayor calificación del alumno en su primera evaluación, tiene como consecuencia que, será mayor las probabilidades de salir aprobado.

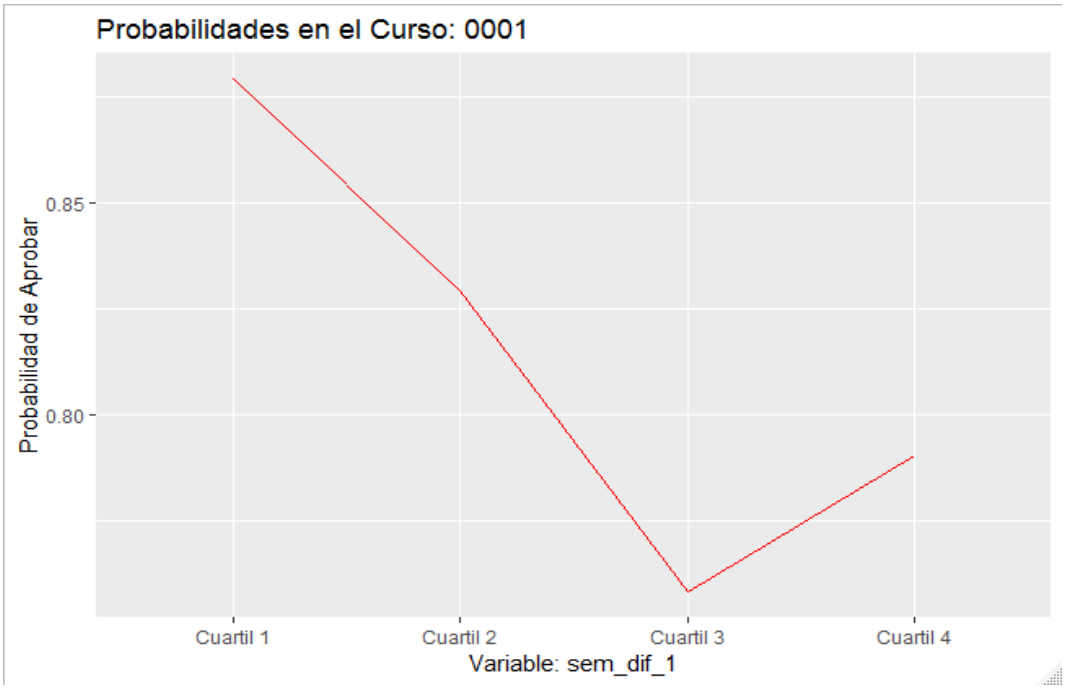


Figura 34: Interpretabilidad de la variable denominada "sem_dif_1" en la muestra de entrenamiento "train" del curso "0001", según la librería "xgboost".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se observa que, mientras más joven sea el alumno, entonces, su probabilidad de salir aprobado será mayor, hasta llegar a un mínimo, después del cual la probabilidad vuelve a subir; el motivo puede estar relacionado con alumnos que retoman sus estudios, o con alumnos del programa de estudios por experiencia laboral para profesionales.

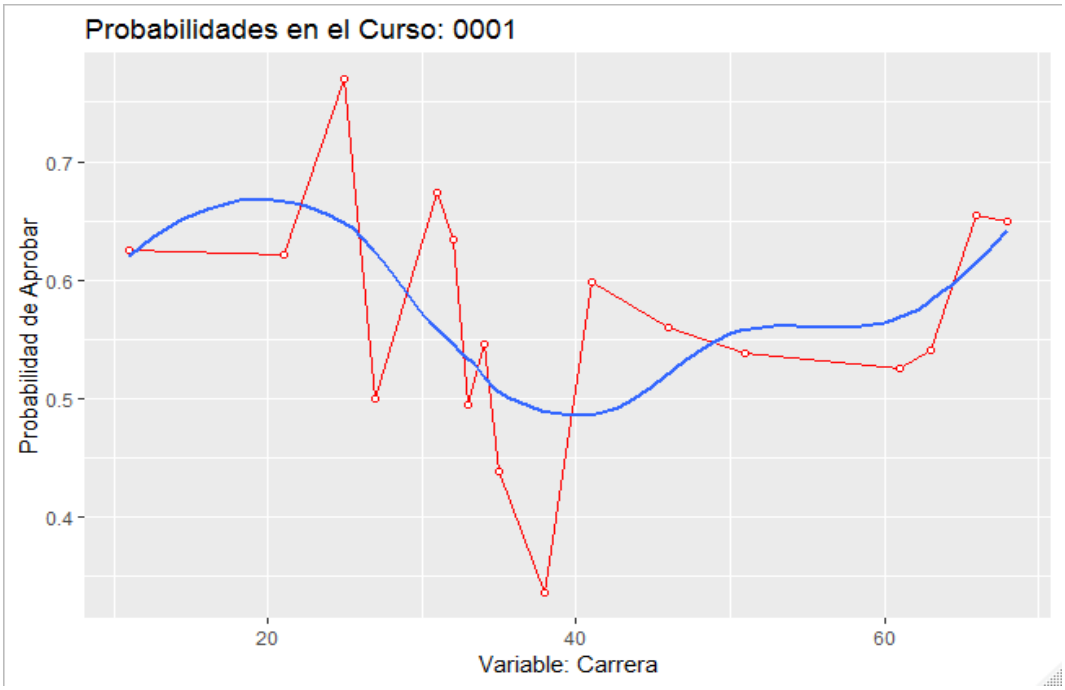


Figura 35: Interpretabilidad de la variable denominada "carrera" en la muestra de entrenamiento "train" del curso "0001", según la librería "xgboost".

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se observa que, mientras el alumno pertenezca a las primeras carreras (las cuales corresponden a Arquitectura y Medicina) o a las últimas carreras (correspondientes a carreras de Ingeniería), entonces, su probabilidad de salir aprobado será mayor.

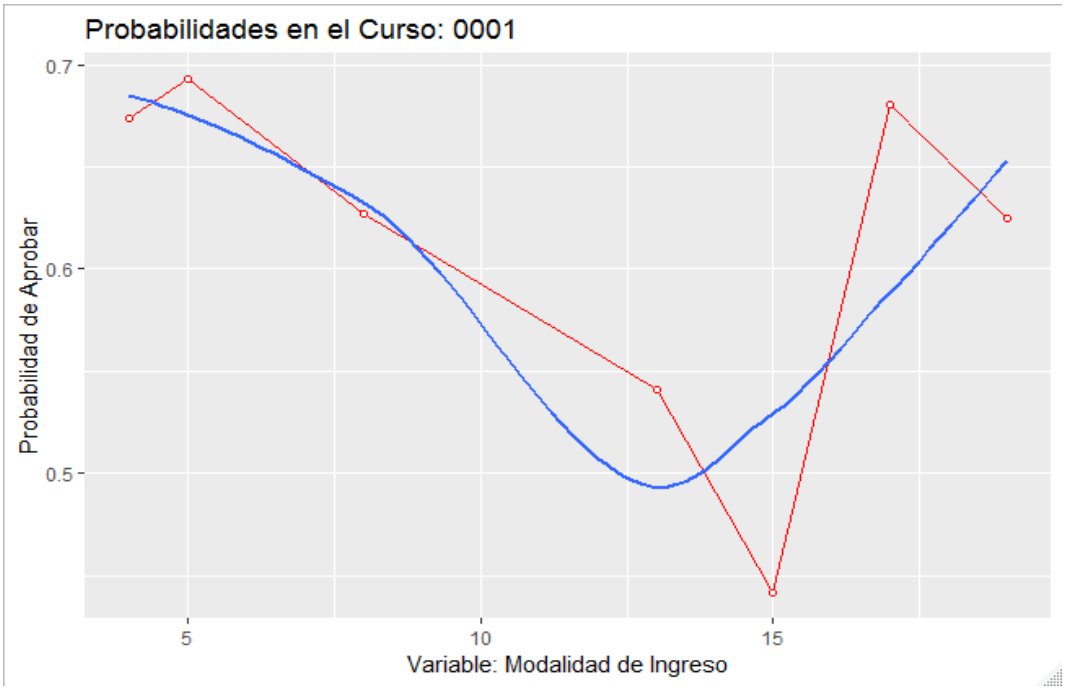


Figura 36: Interpretabilidad de la variable denominada "modalidad de ingreso" en la muestra de entrenamiento "train" del curso "0001", según la librería "xgboost".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se observa que, si el alumno ingreso a la Universidad Ricardo Palma a través de una modalidad de ingreso que se encuentre alrededor del código "15" (la cual corresponde a Examen General de Admisión), entonces, su probabilidad de salir aprobado será menor.

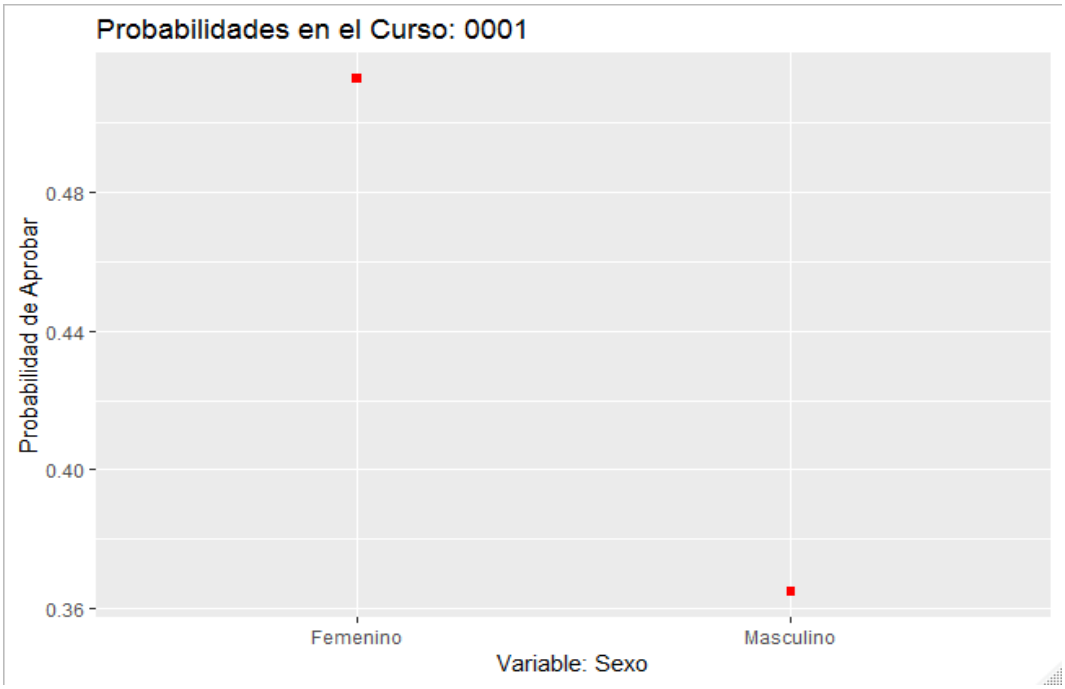


Figura 37: Interpretabilidad de la variable denominada "sexo" en la muestra de entrenamiento "train" del curso "0001", según la librería "xgboost".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede observar que, si el alumno pertenece al sexo Femenino, entonces, su probabilidad de salir aprobado será mayor.

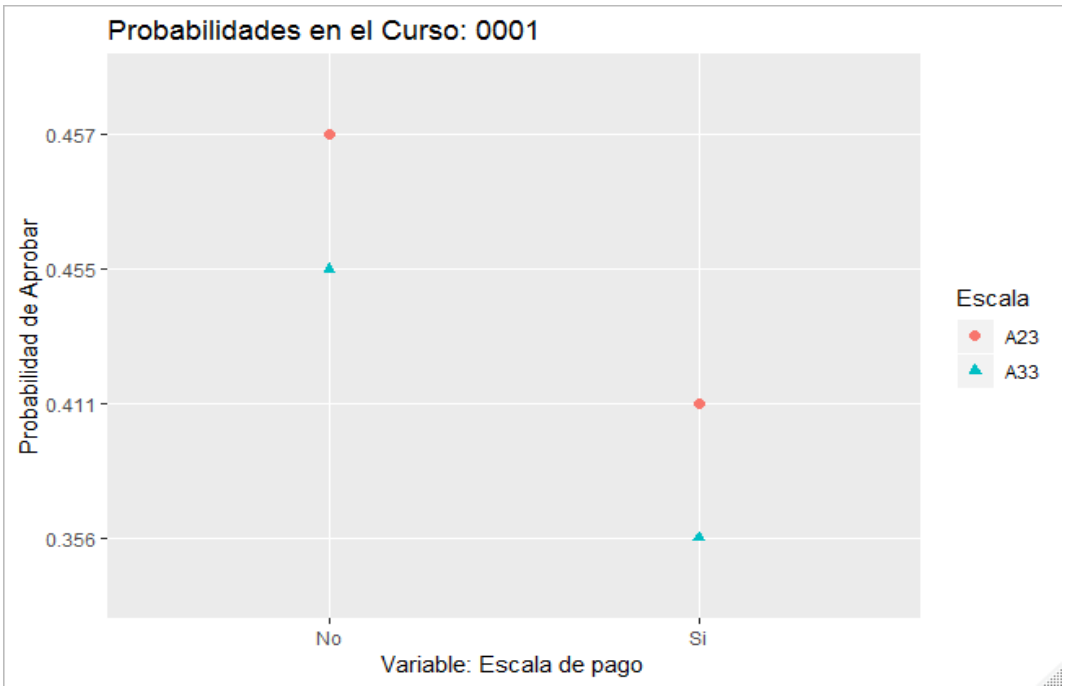


Figura 38: Interpretabilidad de la variable denominada "Escala de pago" en la muestra de entrenamiento "train" del curso "0001", según la librería "xgboost".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede notar que, si el alumno pertenece a la escala de pago denominada "A23", entonces, su probabilidad de salir aprobado será mayor que si pertenece a la escala de pago denominada "A33".

7. Ensamble

Para la técnica de ensamble se seleccionó el promedio aritmético simple con la finalidad de combinar las probabilidades de las predicciones de cada una de las técnicas de modelado independiente.

8. *Stacking*

A continuación, se muestran los valores del indicador "*Accuracy*" en la muestra de entrenamiento "train" y en la muestra de evaluación "test" que fueron obtenidos por las diferentes pruebas de la técnica de modelado denominada *Stacking* en el curso del PEB denominado "Actividades Artísticas y Deportivas".

Tabla 57: Exactitud obtenida por los modelos de prueba de la técnica de modelado *Stacking* implementada en la muestra de entrenamiento "train" y en la muestra de evaluación "test" del curso "0001".

	Exactitud obtenida	
	Train	Test
Prueba 1	0.9195792	0.9004295
Prueba 2	0.9221963	0.8949629

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede observar que el mejor *Accuracy* en la muestra de evaluación "test" se obtuvo en la primera prueba, por lo tanto, se utilizaron dichos valores para definir los parámetros del *Stacking* definitivo.

El siguiente gráfico muestra la valoración interna cuando se aplicó la técnica de modelado *Stacking* en la estimación del entrenamiento, es decir, muestra la variabilidad de cada uno de sus componentes cuando fueron implementados en la muestra de entrenamiento "train", sobresaliendo el componente interno denominado *XGBoosting*.

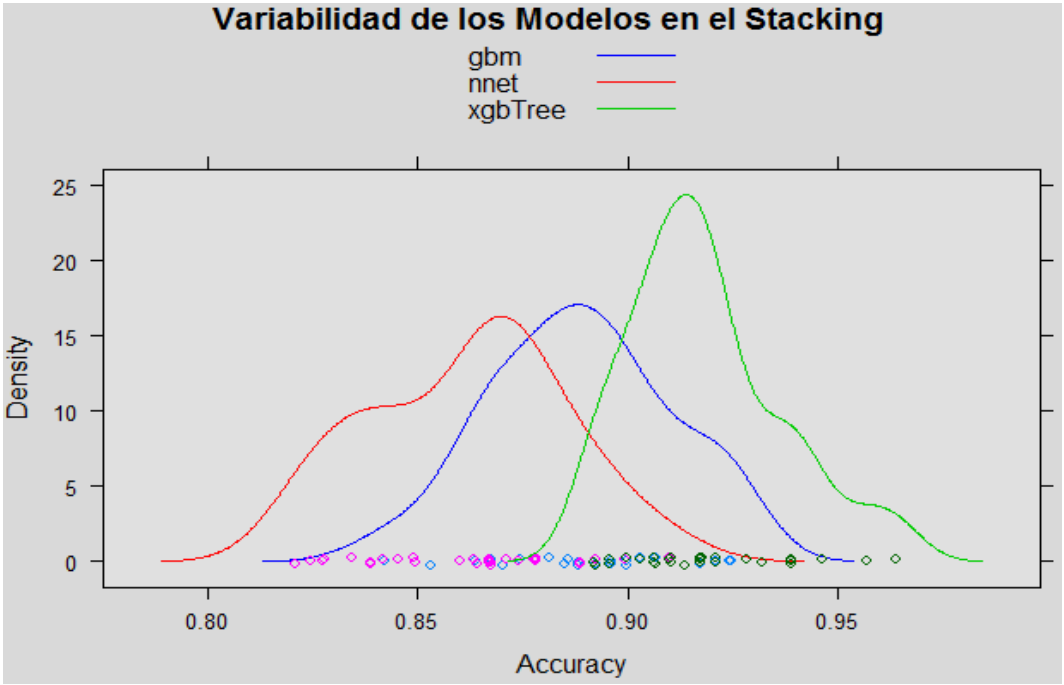


Figura 39: Evaluación interna de la técnica de modelado *Stacking* implementada en la muestra de entrenamiento "train" del curso "0001", según variabilidad mínima.
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

A continuación se aprecia la correlación interna obtenida por la técnica de modelado *Stacking* cuando fue implementada en la muestra de entrenamiento "train":

Tabla 58: Correlación interna de la técnica de modelado *Stacking* implementada en la muestra de entrenamiento "train" del curso "0001".

	nnet	gbm	xgbTree
nnet	1	0.8554661	0.6357500
gbm	0.8554661	1	0.7377189
xgbTree	0.6357500	0.7377189	1

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se debe tener en cuenta que para cada *dataset*, se debe implementar cada técnica de modelado en su mejor versión y determinar cuál de ellas tiene mejor performance.

Con el *software* R se generó el *dataset* de cada curso, siguiendo el mismo procedimiento descrito en los pasos anteriores para cada uno. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 09 y en el Anexo 14 de la presente investigación.

D. Implementación del modelo

La tabla siguiente muestra el indicador de exactitud (*Accuracy*) alcanzado por el modelo seleccionado de cada técnica de modelado cuándo se implementó en la muestra de entrenamiento "train" del curso "0001", donde se puede observar que la técnica de modelado denominada "*Stacking*" obtuvo el mejor índice de exactitud.

Tabla 59: Exactitud obtenida por el modelo seleccionado de cada técnica de modelado implementada en la muestra de entrenamiento "train" del curso "0001".

Nombre del Modelo	**Exactitud**
Red Neuronal Artificial (RNA)	0.8559437
Gradient Boosting Machine (GBM)	0.9140922
XGBoosting	0.9191135
Ensamble	0.8963831
Stacking	0.9195222

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

La finalidad de un modelo predictivo es descubrir un patrón que permita vaticinar un resultado, en este caso en particular, es poder pronosticar si el alumno va a tener un desempeño académico alto o bajo en el curso del PEB denominado "Actividades Artísticas y Deportivas".

E. Aplicación del modelo

El resultado de la implementación de cada una de las técnicas de modelado en la muestra de entrenamiento "train" se designa como modelo o patrón. Cada patrón debe ser aplicado en la muestra de evaluación denominada "test" del curso "Actividades Artísticas y Deportivas", mediante lo cual se obtienen los siguientes indicadores:

Tabla 60: Indicadores obtenidos por el modelo seleccionado de cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0001".

Nombre del Modelo	**Indicador obtenido en el Test**			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.8492776	0.8573818	0.7352941	0.7963380
Gradient Boosting Machine (GBM)	0.8805154	0.8929318	0.7058824	0.7994071
XGBoosting	0.8922296	0.9067336	0.6882353	0.7974844
Ensamble	0.8793440	0.8912589	0.7117647	0.8015118
Stacking	0.9047247	0.9213718	0.6705882	0.7959800

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se puede distinguir que los patrones de las técnicas de modelado denominadas Red Neuronal Artificial (RNA), Ensamble y *Stacking* consiguieron los mejores indicadores de validación, pero específicamente en el indicador denominado "*Accuracy*" sobresalió el modelo *Stacking*.

Se debe tener en cuenta que para cada *dataset*, se realizó la aplicación del patrón de cada técnica de modelado en la muestra de evaluación "test" y se tuvo que analizar los resultados obtenidos en los indicadores.

F. Evaluación del modelo

A continuación se aprecia la comparación del patrón de cada técnica de modelado, según los valores que fueron obtenidos por el indicador "*Accuracy*" en la muestra de evaluación "test" del curso "Actividades Artísticas y Deportivas":

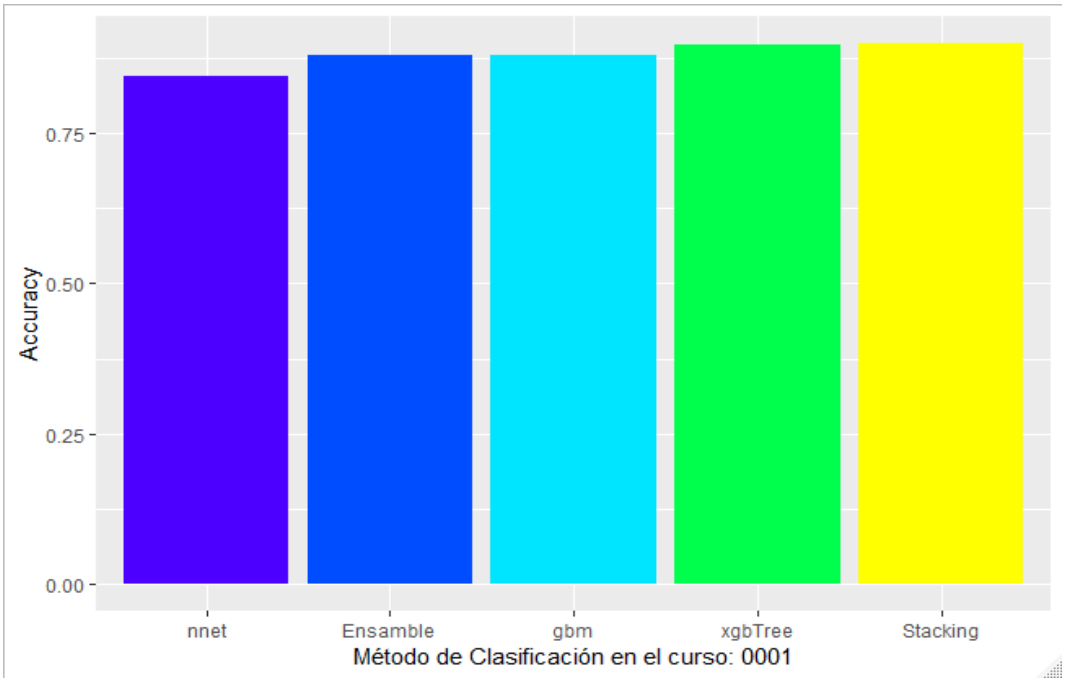


Figura 40: Comparación de las técnicas de modelado aplicadas en la muestra de evaluación "test" del curso "0001", según el indicador "*Accuracy*".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se concluyó que la mejor técnica de modelado resulto ser el *Stacking*, que a su vez estuvo compuesta por la combinación del *Gradient Boosting Machine* (GBM), la Red Neuronal Artificial (RNA) y el *XGBoosting* siendo esta última la que tuvo la menor variabilidad interna durante el entrenamiento.

Se debe tener en cuenta que para cada curso y su respectivo *dataset*, determinar la mejor técnica de modelado estuvo basado en la realización de una comparación detallada de los valores emitidos por los indicadores al aplicar cada patrón en la muestra de evaluación "test".

Se utilizó el *software* R para generar cada uno de los modelos con información del *dataset* del curso con código "0001", visualizar sus resultados y compararlos. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 10 de la presente investigación.

G. Modelado del *dataset* de cada uno de los cursos faltantes

Con el *software* R se procesó el *dataset* de cada curso, siguiendo el mismo procedimiento descrito en los pasos anteriores. Los comandos utilizados y sus respectivos resultados se encuentran en el Anexo 10 y en el Anexo 15 de la presente investigación.

4.1.3.4. Sumario.

Cada uno de los 13 *datasets* ha recibido el mismo procedimiento para obtener los mejores resultados en el modelado, un resumen de las características de cada técnica de modelado lo detallaremos a continuación.

Con respecto a los parámetros de la técnica de modelado denominada Red Neuronal Artificial (RNA), las características internas han permitido una división en 3 grupos:

A. Características de la RNA desde el curso "0002" hasta el curso "0005"

Cada RNA en este grupo posee características similares a las mostradas por la RNA del curso "0001", pero sus valores varían de acuerdo a cada *dataset*.

B. Características de la RNA desde el curso "0006" hasta el curso "0011"

La RNA obtenida, es un perceptrón multicapa con una arquitectura 15-5-1 y 86 pesos, donde 5 es la cantidad de neuronas ocultas en una única capa; se definió como función de activación a la función logística y el algoritmo de aprendizaje es el método de optimización numérica denominado BFGS. Los parámetros "trace" y

"linout" fueron definidos con valor Falso; los parámetros "size" y "decay" toman distintos valores de acuerdo a cada *dataset*.

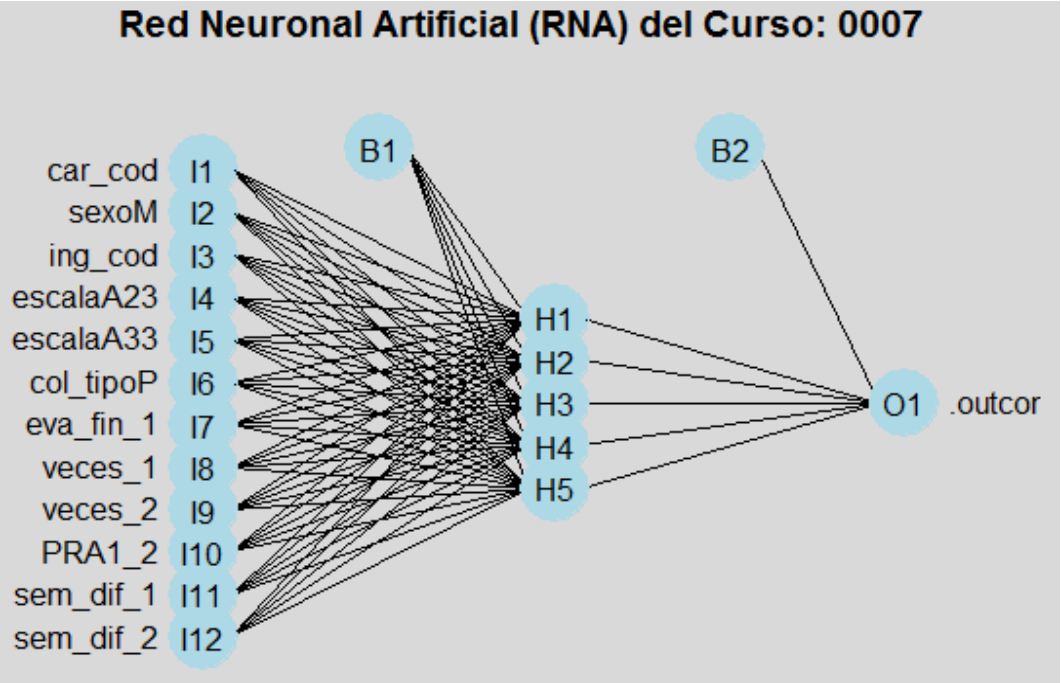


Figura 41: Gráfico de la estructura de la Red Neuronal Artificial definitiva obtenida con la librería "nnet" en el *dataset* del curso "0007".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Cada RNA en este grupo posee características similares a las mostradas en el gráfico de la RNA del curso "0007", pero sus valores varían de acuerdo a cada *dataset*.

C. Características de la RNA del curso "0012" y del curso "0013"

La RNA obtenida, es un perceptrón multicapa con una arquitectura 18-7-1 y 141 pesos, donde 7 es la cantidad de neuronas ocultas en una única capa; se definió como función de activación a la función logística y el algoritmo de aprendizaje es el método de optimización numérica denominado BFGS. Los parámetros "trace" y "linout" fueron definidos con valor Falso; los parámetros "size" y "decay" toman distintos valores de acuerdo a cada *dataset*.

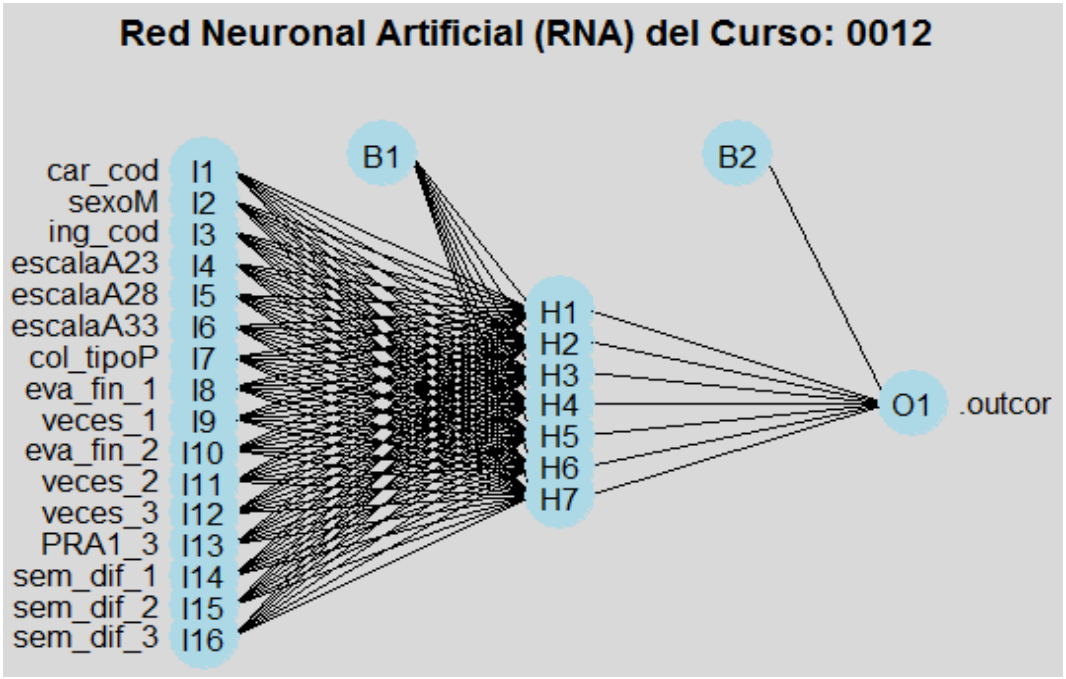


Figura 42: Gráfico de la estructura de la Red Neuronal Artificial definitiva obtenida con la librería "nnet" en el *dataset* del curso "0012".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Cada RNA en este grupo posee características similares a las mostradas en el gráfico de la RNA del curso "0012", pero sus valores varían de acuerdo a cada *dataset*.

Con respecto a los parámetros de la técnica de modelado denominada *Gradient Boosting Machine* (GBM), son los mismos para cada uno de los 13 *datasets*.

El parámetro "verbose" fue definido con valor Falso; el parámetro "distribution" se definió con la función de perdida Bernoulli; los parámetros "n.trees, "interaction.depth", "shrinkage" y el parámetro "n.minobsinnode" adquieren distintos valores de acuerdo a cada *dataset*.

Con respecto a los parámetros de la técnica de modelado denominada *XGBoosting*, son los mismos para cada uno de los 13 *datasets*.

El parámetro "verbose" fue definido con valor Falso; el parámetro "distribution" se definió con la función de perdida Bernoulli; los parámetros "n.trees, "interaction.depth", "shrinkage" y "n.minobsinnode" adquieren distintos valores de acuerdo a cada *dataset*.

El parámetro "verbose" fue definido con valor 0; el parámetro "objective" se definió con la función de perdida binary:logistic; el parámetro "eval_metric" se definió con "error"; los parámetros "nrounds", "max_depth", "eta", "gamma", "colsample_bytree",

"min_child_weight" y "subsample" adoptaron distintos valores de acuerdo a cada *dataset*.

Con respecto a los parámetros de la técnica de modelado denominada Ensamble, se seleccionó el promedio aritmético simple de las predicciones para cada uno de los 13 *datasets*.

Con respecto a los parámetros de la técnica de modelado denominada *Stacking*, se seleccionó como metamodelo a la técnica de modelado denominada regresión logística para cada uno de los 13 *datasets*.

4.1.4. Evaluación de los otros cursos.

A. En la muestra de entrenamiento

A continuación, se muestra el indicador de exactitud (*Accuracy*) que se obtuvo al implementar cada técnica de modelado en la muestra de entrenamiento "train" de cada uno de los cursos adicionales.

Tabla 61: Exactitud obtenida por cada técnica de modelado implementada en la muestra de entrenamiento "train" desde el curso "0002" al curso "0005".

Nombre del Modelo	**Exactitud obtenida en el Train**			
	Curso 02	Curso 03	Curso 04	Curso 05
Red Neuronal Artificial (RNA)	0.8838881	0.9021553	0.8210626	0.8728750
Gradient Boosting Machine (GBM)	0.8924208	0.9051481	0.8382120	0.8835765
XGBoosting	0.8889887	0.9052485	0.8436823	0.8842004
Ensamble	0.8884325	0.9041840	0.8343190	0.8802173
Stacking	0.8904812	0.9042933	0.8416317	0.8841640

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 62: Exactitud obtenida por cada técnica de modelado implementada en la muestra de entrenamiento "train" desde el curso "0006" al curso "0009".

Nombre del Modelo	**Exactitud obtenida en el Train**			
	Curso 06	Curso 07	Curso 08	Curso 09
Red Neuronal Artificial (RNA)	0.8845501	0.8747073	0.9328083	0.9105168
Gradient Boosting Machine (GBM)	0.8894481	0.8843864	0.9318478	0.9067753
XGBoosting	0.8952193	0.8882958	0.9312795	0.9040252
Ensamble	0.8897391	0.8824632	0.9319786	0.9071058
Stacking	0.8973741	0.8949339	0.9337324	0.9143563

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 63: Exactitud obtenida por cada técnica de modelado implementada en la muestra de entrenamiento "train" desde el curso "0010" al curso "0012".

Nombre del Modelo	Exactitud obtenida en el Train			
	Curso 10	Curso 11	Curso 12	Curso 13
Red Neuronal Artificial (RNA)	0.9065064	0.8904465	0.9277922	0.9070807
Gradient Boosting Machine (GBM)	0.9146077	0.8866635	0.9367614	0.9155540
XGBoosting	0.9248077	0.8902166	0.9460986	0.9352297
Ensamble	0.9153073	0.8891089	0.9368840	0.9192882
Stacking	0.9254346	0.8911148	0.9444345	0.9350487

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se pudo determinar que, la técnica de modelado denominada *Stacking* consiguió los mejores indicadores de validación en 7 ocasiones (incluyendo el curso con código "0001"), seguida por el *XGBoosting* en 5 ocasiones y para completar el *Gradient Boosting Machine* (GBM) en 1 ocasión.

B. En la muestra de evaluación

A continuación se muestra las tablas con los indicadores obtenidos al aplicar el patrón de cada técnica de modelado en la muestra de evaluación "test" de cada uno de los cursos adicionales.

Tabla 64: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0002".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.8877060	0.9523810	0.5888031	0.7705920
Gradient Boosting Machine (GBM)	0.8945742	0.9636591	0.5752896	0.7694744
XGBoosting	0.8956044	0.9582289	0.6061776	0.7822033
Ensamble	0.8956044	0.9607352	0.5945946	0.7776649
Stacking	0.8935440	0.9615706	0.5791506	0.7703606

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 65: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0003".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.9112819	0.9664148	0.6382536	0.8023342
Gradient Boosting Machine (GBM)	0.9102340	0.9685139	0.6216216	0.7950677
XGBoosting	0.9119804	0.9689337	0.6299376	0.7994356
Ensamble	0.9119804	0.9693535	0.6278586	0.7986061
Stacking	0.9123297	0.9672544	0.6403326	0.8037935

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 66: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0004".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.8247934	0.8495954	0.7907129	0.8201542
Gradient Boosting Machine (GBM)	0.8371901	0.8719657	0.7894048	0.8306853
XGBoosting	0.8385675	0.8729177	0.7913669	0.8321423
Ensamble	0.8382920	0.8691099	0.7959451	0.8325275
Stacking	0.8432507	0.8786292	0.7946370	0.8366331

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 67: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0005".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.8877327	0.9458128	0.6839506	0.8148817
Gradient Boosting Machine (GBM)	0.8953998	0.9528501	0.6938272	0.8233386
XGBoosting	0.8970427	0.9556650	0.6913580	0.8235115
Ensamble	0.8926616	0.9521464	0.6839506	0.8180485
Stacking	0.8943045	0.9493315	0.7012346	0.8252830

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 68: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0006".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.8860518	0.9398221	0.7419355	0.8408788
Gradient Boosting Machine (GBM)	0.8784299	0.9256934	0.7517532	0.8387233
XGBoosting	0.8894817	0.9330194	0.7727910	0.8529052
Ensamble	0.8917683	0.9356358	0.7741935	0.8549147
Stacking	0.8963415	0.9455782	0.7643759	0.8549771

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 69: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0007".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.8803285	0.9493938	0.6818182	0.8156060
Gradient Boosting Machine (GBM)	0.8791553	0.9499209	0.6757576	0.8128393
XGBoosting	0.8912788	0.9546653	0.7090909	0.8318781
Ensamble	0.8885413	0.9578281	0.6893939	0.8236110
Stacking	0.8904967	0.9609910	0.6878788	0.8244349

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 70: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0008".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.9363167	0.9755776	0.6754386	0.8255081
Gradient Boosting Machine (GBM)	0.9334481	0.9709571	0.6842105	0.8275838
XGBoosting	0.9368904	0.9768977	0.6710526	0.8239752
Ensamble	0.9351692	0.9742574	0.6754386	0.8248480
Stacking	0.9386116	0.9716172	0.7192982	0.8454577

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 71: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0009".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.9254420	0.9831461	0.6609442	0.8220451
Gradient Boosting Machine (GBM)	0.9123751	0.9625468	0.6824034	0.8224751
XGBoosting	0.9169869	0.9606742	0.7167382	0.8387062
Ensamble	0.9208301	0.9662921	0.7124464	0.8393692
Stacking	0.9254420	0.9672285	0.7339056	0.8505670

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 72: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0010".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.9112545	0.9616725	0.6957447	0.8287086
Gradient Boosting Machine (GBM)	0.9084308	0.9556994	0.7063830	0.8310412
XGBoosting	0.9160952	0.9656546	0.7042553	0.8349549
Ensamble	0.9185155	0.9676456	0.7085106	0.8380781
Stacking	0.9221460	0.9736187	0.7021277	0.8378732

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 73: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0011".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.8860828	0.9405738	0.6851385	0.8128562
Gradient Boosting Machine (GBM)	0.8887695	0.9467213	0.6750630	0.8108921
XGBoosting	0.8860828	0.9412568	0.6826196	0.8119382
Ensamble	0.8882321	0.9433060	0.6851385	0.8142223
Stacking	0.8871574	0.9398907	0.6926952	0.8162930

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 74: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0012".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.9279509	0.9648391	0.7060932	0.8354661
Gradient Boosting Machine (GBM)	0.9259070	0.9672229	0.6774194	0.8223211
XGBoosting	0.9391926	0.9737783	0.7311828	0.8524806
Ensamble	0.9356157	0.9719905	0.7168459	0.8444182
Stacking	0.9407256	0.9737783	0.7419355	0.8578569

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 75: Indicadores obtenidos por cada técnica de modelado aplicada en la muestra de evaluación "test" del curso "0013".

Nombre del Modelo	Indicador obtenido en el Test			
	Accuracy	Sensibilidad	Especificidad	Curva ROC
Red Neuronal Artificial (RNA)	0.9009095	0.9428044	0.7537797	0.8482921
Gradient Boosting Machine (GBM)	0.9071326	0.9477245	0.7645788	0.8561517
XGBoosting	0.9310675	0.9587946	0.8336933	0.8962439
Ensamble	0.9186213	0.9538745	0.7948164	0.8743455
Stacking	0.9301101	0.9581796	0.8315335	0.8948565

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se observó que en la mayoría de cursos el patrón de la técnica de modelado denominada *Stacking* consiguió los mejores indicadores de validación, seguida por *XGBoosting* y posteriormente el Ensamble y la Red Neuronal Artificial con resultados similares entre ellas.

4.1.5. Despliegue.

La finalidad de un modelo predictivo es descubrir un patrón que nos permita vaticinar un resultado, pero dicho conocimiento debe convertirse en acciones al interior de la Universidad Ricardo Palma, su implementación debe comprender las etapas siguientes:

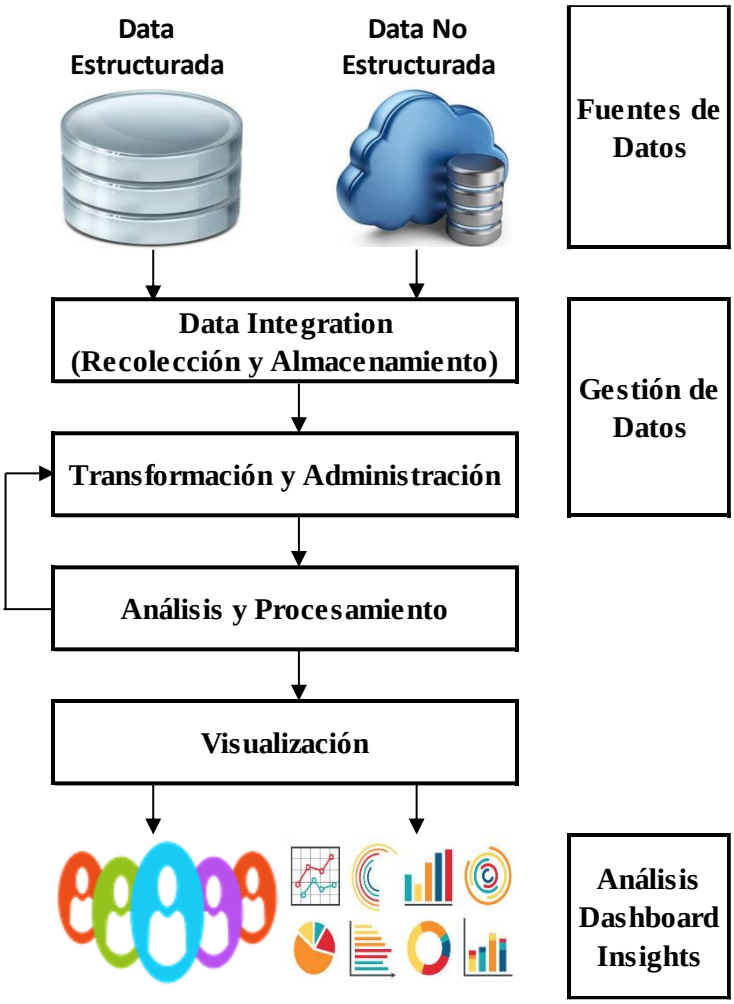


Figura 43: Diagrama de la estructura de las etapas del Despliegue.
Elaboración: Propia, 2019

Cada una de las etapas para la planificación del despliegue comprende lo siguiente:

A. Ingreso o fuentes de datos

En una primera fase son los registros provenientes de los sistemas transaccionales de la Universidad Ricardo Palma. Posteriormente debería considerarse la posibilidad de almacenar información proveniente de las redes sociales.

B. Gestión de datos

También conocida como gestión del almacenamiento de los datos estructurados y no estructurados.

Es la manera de cómo se organiza y gestiona el conjunto de datos que ingresa, independientemente del origen, formato, velocidad, tamaño o frecuencia y que se debe almacenar en sistemas de archivos distribuidos (se recomienda el *Framework*

Hadoop, donde el almacenamiento de archivos tiene la denominación de HDFS - *Hadoop Distributed File System*). También se encarga de la administración de los usuarios y/o grupo de usuarios.

C. Análisis y procesamiento de datos

Descubrir información útil para la empresa mediante el uso de modelos predictivos para descubrir patrones. Para cumplir con dicho objetivo se puede utilizar herramientas desarrolladas por *software* libre, de pago o en la nube.

El modelo que se desarrolló en la presente investigación, es una herramienta paralelizable, por lo tanto se puede dividir el flujo de datos en varios nodos y utilizar Spark para procesarlo mediante el API para el *software* R; también se podría mejorarlo mediante la utilización de librerías propias de *big data* como MapReduce o Spark ML.

D. Visualización

Es la etapa de la visualización de los resultados de los modelos, explotando los beneficios por el conocimiento obtenido incluyéndolos en la toma de decisiones de la Universidad Ricardo Palma. La manera más habitual de observarlos es a través de reportes, *dashboards* o tableros en tiempo real con la finalidad de encontrar *insights*. Adicionalmente se debe realizar un seguimiento al comportamiento de los resultados encontrados, mediante el monitoreo de un tablero de control.

4.2. Análisis de resultados

El desempeño del modelo esta operacionalizado por el valor obtenido en el indicador *Accuracy*, que es la tasa de acierto de la matriz de consistencia y representa la proporción entre el número de predicciones correctas (las cuales pueden ser positivas o negativas) con el número total de predicciones (cantidad de observaciones).

En las siguientes tablas se muestra por cada curso el valor del mejor *Accuracy* y la técnica de modelado a la que pertenece, los cuales fueron obtenidos cuando se aplicó cada técnica de modelado en la muestra de evaluación “test” de cada curso.

Tabla 76: Mejor técnica de modelado basado en *Accuracy*, según aplicación en la muestra de evaluación "test" desde el curso "0001" al curso "0005".

Nombre del Modelo	Modelo con mejor Accuracy en el Test				
	Curso 01	Curso 02	Curso 03	Curso 04	Curso 05
Red Neuronal Artificial (RNA)					
Gradient Boosting Machine (GBM)					
XGBoosting		0.8956044			0.8970427
Ensamble		0.8956044			
Stacking	0.9004295		0.9123297	0.8432507	

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 77: Mejor técnica de modelado basado en *Accuracy*, según aplicación en la muestra de evaluación "test" desde el curso "0006" al curso "0009".

Nombre del Modelo	Modelo con mejor Accuracy en el Test			
	Curso 06	Curso 07	Curso 08	Curso 09
Red Neuronal Artificial (RNA)				0.9254420
Gradient Boosting Machine (GBM)				
XGBoosting		0.8912788		
Ensamble				
Stacking	0.8963415		0.9386116	0.9254420

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 78: Mejor técnica de modelado basado en *Accuracy*, según aplicación en la muestra de evaluación "test" desde el curso "0010" al curso "0013".

Nombre del Modelo	Modelo con mejor Accuracy en el Test			
	Curso 10	Curso 11	Curso 12	Curso 13
Red Neuronal Artificial (RNA)				
Gradient Boosting Machine (GBM)		0.8887695		
XGBoosting				0.9310675
Ensamble				
Stacking	0.9221460		0.9407256	

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Puede advertirse que, en la mayoría de cursos el patrón de la técnica de modelado denominada *Stacking* consiguió los mejores resultados en el indicador de validación denominado "*Accuracy*", seguida por *XGBoosting,* y posteriormente Ensamble con Red Neuronal Artificial.

La consecuencia del análisis de los resultados en cada curso permitió determinar que pronosticar el desempeño académico alto o bajo de un alumno en cualquiera de los 13 cursos del PEB es posible.

A continuación se despliega, por cada curso, la cantidad de alumnos aprobados y desaprobados reales versus los pronósticos, los cuales fueron obtenidos por la mejor técnica de modelado en la muestra de evaluación "test" de cada curso.

Tabla 79: Cantidad de alumnos aprobados y desaprobados (reales versus pronosticados), según aplicación en la muestra de evaluación "test" desde el curso "0001" al curso "0013".

Cantidad de Alumnos obtenidos en el Test	**Real**		**Predicción**		**Diferencias**			
	Aprobó	Desaprobó	Aprobó	Desaprobó	Aprobó		Desaprobó	
Curso 01	2,391	170	2,195	111	196	8.2%	59	34.7%
Curso 02	2,394	518	2,302	300	92	3.8%	218	42.1%
Curso 03	2,382	481	2,308	303	74	3.1%	178	37.0%
Curso 04	2,101	1,529	1,846	1,215	255	12.1%	314	20.5%
Curso 05	1,421	405	1,349	284	72	5.1%	121	29.9%
Curso 06	1,911	713	1,807	545	104	5.4%	168	23.6%
Curso 07	1,897	660	1,817	455	80	4.2%	205	31.1%
Curso 08	1,515	228	1,480	153	35	2.3%	75	32.9%
Curso 09	1,068	233	1,050	154	18	1.7%	79	33.9%
Curso 10	2,009	470	1,956	330	53	2.6%	140	29.8%
Curso 11	1,464	397	1,378	271	86	5.9%	126	31.7%
Curso 12	1,678	279	1,634	207	44	2.6%	72	25.8%
Curso 13	1,626	463	1,558	385	68	4.2%	78	16.8%
	23,857	**6,546**	**22,680**	**4,713**	**1,177**	**4.9%**	**1,833**	**28.0%**

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

La consecuencia de la obtención de la cantidad de alumnos aprobados y desaprobados en cada curso, permitió determinar que es posible pronosticarlas en cualquiera de los 13 cursos del PEB.

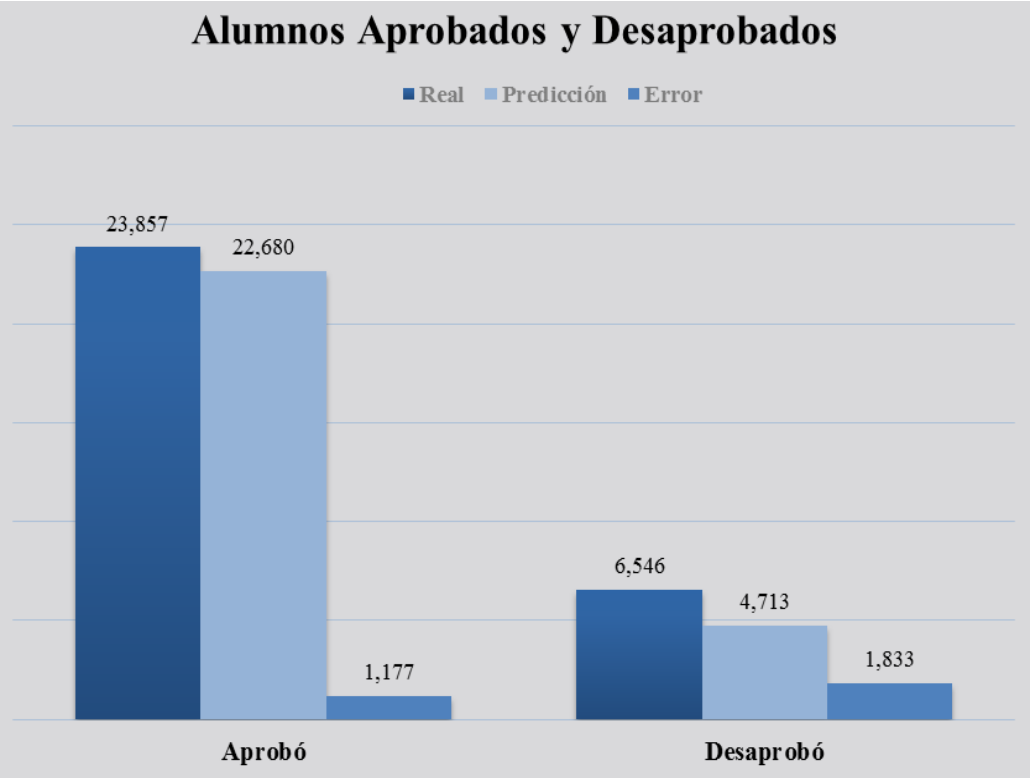


Figura 44: Gráfico de la cantidad de alumnos aprobados y desaprobados (reales versus pronosticados), según aplicación en la muestra de evaluación "test" desde el curso "0001" al curso "0013".
Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

El corolario de la presente investigación es el siguiente:

1. En relación a la metodología

   El método de trabajo que se utilizó para el desarrollo de la presente investigación fue la Metodología CRISP-DM, su utilización fue muy útil para obtener ideas y poder estructurar un camino donde fue sencillo completar cada etapa, no por la complejidad de la investigación, sino por la meta que se debía alcanzar en cada una.

2. En relación a la calidad de los datos

   Cada modelo brinda una exactitud que se encuentra íntimamente relacionada con el conjunto de datos para las pruebas (*Dataset* de cada curso), y de acuerdo a los resultados obtenidos es altamente probable que los datos de entrada eran suficientemente confiables para realizar predicciones.

   Lamentablemente la calidad de los datos para poder extraer patrones fue deficiente, si bien es cierto que la información fue extraída de sistemas de información que se utilizan continuamente y que minimizan la aparición de errores en el procesamiento de notas, también es cierto que, la estructura de esos mismos datos no podía usarse ni para la generación de un *dataset* donde se pueda realizar una simple predicción, por lo que se tuvo que realizar un arduo y extenso pre procesamiento (establecer, catalogar, reorganizar, escoger y limpiar cada archivo de datos).

3. Con respecto a la hipótesis general

   Los modelos predictivos extraen patrones de los datos históricos transaccionales y el objetivo es evaluar la probabilidad de que un alumno tenga un rendimiento académico aprobado o desaprobado, es decir buscamos que al menos un modelo haya podido cumplir con los criterios de éxito del problema planteado.

Las técnicas de modelado (algoritmos de *Machine Learning*) que se aplicaron en la muestra de entrenamiento "train" y en la muestra de evaluación "test" para cada uno de los cursos del Programa de Estudios Básicos (PEB), obtuvieron indicadores de exactitud muy buenos para ambas muestras, por lo tanto, cada algoritmo de *Machine Learning* predijo con alta precisión la cantidad de alumnos aprobados y desaprobados en cada curso.

Por lo tanto, se demostró que si es posible aplicar un modelo de *Machine Learning* para la determinación del rendimiento académico.

4. Con respecto a las hipótesis específicas

   4.1. Una de las técnicas de modelado que se aplicó a cada uno de los cursos del Programa de Estudios Básicos (PEB) fue el algoritmo de Redes Neuronales Artificiales (RNA) de manera independiente y como parte de una unión de técnicas de modelado.

   Se observa en la Tabla 60 y desde la Tabla 64 a la Tabla 75 que, se obtuvo indicadores de exactitud muy buenos en cada curso, por lo tanto, el algoritmo de Redes Neuronales Artificiales (RNA) predijo con alta precisión la cantidad de alumnos aprobados y desaprobados.

   4.2. Otra de las técnicas de modelado que se aplicó a cada uno de los cursos del Programa de Estudios Básicos (PEB) fue el algoritmo *Boosting* (en dos métodos denominados *Gradient Boosting Machine* (GBM) y *XGBoosting*) de manera independiente y como parte de una unión de técnicas de modelado.

   En la Tabla 60 y desde la Tabla 64 a la Tabla 75 que, se aprecia que se obtuvo indicadores de exactitud muy buenos en cada curso, por lo tanto, el algoritmo de *Gradient Boosting Machine* (GBM) y el algoritmo de *XGBoosting*, predijeron con alta precisión la cantidad de alumnos aprobados y desaprobados.

   4.3. En las siguientes tablas se observan, únicamente, los indicadores obtenidos cuando se aplicaron las técnicas de modelado denominadas Red Neuronal Artificial (RNA) y *Boosting*, en la muestra de evaluación "test" de cada curso.

Tabla 80: Indicador *Accuracy* obtenido por cada técnica de modelado en forma individual, aplicada en la muestra de evaluación "test" desde el curso "0001" al curso "0005".

Nombre del Modelo	**Accuracy obtenida en el Test**				
	Curso 01	Curso 02	Curso 03	Curso 04	Curso 05
Red Neuronal Artificial (RNA)	0.8461538	0.8877060	0.9112819	0.8247934	0.8877327
Gradient Boosting Machine (GBM)	0.8797345	0.8945742	0.9102340	0.8371901	0.8953998
XGBoosting	0.8973057	0.8956044	0.9119804	0.8385675	0.8970427

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 81: Indicador *Accuracy* obtenido por cada técnica de modelado en forma individual, aplicada en la muestra de evaluación "test" desde el curso "0006" al curso "0009".

Nombre del Modelo	**Accuracy obtenida en el Test**			
	Curso 06	Curso 07	Curso 08	Curso 09
Red Neuronal Artificial (RNA)	0.8860518	0.8803285	0.9363167	0.9254420
Gradient Boosting Machine (GBM)	0.8784299	0.8791553	0.9334481	0.9123751
XGBoosting	0.8894817	0.8912788	0.9368904	0.9169869

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Tabla 82: Indicador *Accuracy* obtenido por cada técnica de modelado en forma individual, aplicada en la muestra de evaluación "test" desde el curso "0010" al curso "0013".

Nombre del Modelo	**Accuracy obtenida en el Test**			
	Curso 10	Curso 11	Curso 12	Curso 13
Red Neuronal Artificial (RNA)	0.9112545	0.8860828	0.9279509	0.9009095
Gradient Boosting Machine (GBM)	0.9084308	0.8887695	0.9259070	0.9071326
XGBoosting	0.9160952	0.8860828	0.9391926	0.9310675

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Se determinó que para predecir el rendimiento académico en cada curso del PEB, el mejor algoritmo de *Machine Learning* fue el modelo *XGBoosting* en 11 ocasiones y en 1 ocasión fueron los modelos *Gradient Boosting Machine* (GBM) y Red Neuronal Artificial (RNA). Entonces podemos afirmar que el nivel de precisión del algoritmo *Boosting* es más óptimo que el del algoritmo de RNA.

Debemos tener en cuenta que, cuando se realizó la comparación con otros algoritmos de *Machine Learning*, su nivel de precisión a pesar de ser bueno, no siempre fue el mejor.

5. Con respecto a los aspectos económicos

Teniendo como sustento la Tabla 10, en la cual se puede visualizar la distribución de alumnos según su escala de pago así como los montos de la matrícula y de la cuota por cada una, se procedió a eliminar a los estudiantes extranjeros y/o de intercambio internacional, porque ellos no están incluidos en el análisis de la presente investigación; obteniéndose lo siguiente:

Tabla 83: Distribución del alumnado según escala de pago, con el valor de la matrícula y de la cuota (en Soles).

Escala	**Cantidad**	**%**	**Cuota**		**Matricula**
			Normal	Ponderada	
A23	5,414	59.66%	1,350.00	805.39	300.00
A33	2,024	22.30%	1,250.00	278.79	
A13	882	9.72%	2,100.00	204.10	
A38	336	3.70%	1,450.00	53.69	
A28	275	3.03%	1,550.00	46.97	
A18	144	1.59%	2,600.00	41.26	
	9,075	**100.00%**	**1,716.67**	**1,430.19**	

Fuente: Centro de Computo de la Universidad Ricardo Palma. Elaboración: Propia, 2019

Debemos considerar que en cada semestre se abonan 5 cuotas y una matrícula, por lo tanto, un alumno abona por semestre la cantidad de 7,450.95 soles, siempre y cuando consideremos el promedio ponderado, si consideramos el promedio aritmético dicho abono se incrementa a 8,883.35 soles por semestre.

De acuerdo a la Figura 01, hay 789 alumnos desaprobados en el semestre 2015-I, si consideramos, que dicho número de alumnos abandonan los estudios, que les resta un promedio de 8 semestres para culminar sus estudios y que utilizamos el promedio ponderado, la Universidad Ricardo Palma dejaría de percibir el monto total de 47´030,396.40 soles si consideramos el promedio aritmético dicho total se incrementa a 56’071,705.20 soles.

En la presente investigación se demostró cada una de las hipótesis planteadas, por lo tanto, aspirar a que los alumnos no obtengan un rendimiento académico negativo es

posible y como consecuencia no abandonaran sus estudios y la Universidad Ricardo Palma podría disminuir su proyección de pérdidas por retiro de los estudiantes, como mínimo, en un 50%, lo que representaría un aproximado de 23.5 millones de soles.

El implementar el modelo descrito en la presente investigación en conjunto con las medidas correctivas de ayuda al alumnado, tiene como efecto un impacto económico positivo en beneficio de la Universidad Ricardo Palma.

5.2. Recomendaciones

Brindamos las siguientes recomendaciones:

1. Con respecto a los archivos de datos

    1.1. Creemos que el Programa de Estudios Básicos no debería formar parte del archivo de datos de Carreras, debería formar parte de un archivo de "Áreas Temáticas", donde figure junto a Matemáticas, Humanidades, Idiomas o denominaciones similares, que permitiría un mejor manejo de los cursos por cada una de las áreas.

    1.2. Debería conformarse un archivo de datos de colegios, porque al realizar la normalización de la información que nos proporcionó el Centro de Computo de la Universidad Ricardo Palma, nos encontramos inicialmente con 2,213 registros de los cuales una proporción mayor al 20% estaban duplicados, cuyos errores eran principalmente fallas ortográficas.

    Dicho archivo debería contener, como mínimo, el código (el cual debería estar en concordancia con el código establecido por el Ministerio de Educación para cada institución educativa), el nombre, el tipo y su ubicación geográfica.

    1.3. Se debería elaborar un registro electrónico de las notas, correspondiente a la etapa escolar del alumno que postula a la universidad, lo cual hubiera proporcionado características adicionales para la elaboración del modelo.

    1.4. El archivo que contiene los tipos de evaluaciones debería ser revisado, de forma tal que, el concepto de cada evaluación debería ser único para evitar la ambigüedad que suscita las descripciones de los tipos de evaluación. Esto se

fundamenta en el hecho de que, para cada *dataset* de la presente investigación, se ha tenido que realizar una tabla de equivalencias para los tipos de evaluaciones en cada curso.

1.5. El archivo de notas es un archivo de datos transaccional típico, pero cuya estructura debe ser estudiada y transformada antes de poder elaborar un *dataset* que permita realizar predicciones con cierto grado de exactitud.

Debe ser corregida la estructura del archivo, en el extremo de modificar las características de la columna "Mascara del curso" por las de "Código del curso".

Sin embargo, la duplicidad de los tipos de evaluaciones en un mismo curso impide obtener datos estadísticos de una manera óptima, para realizarlo se debe pasar necesariamente por una etapa intermedia de estandarización.

1.6. En el archivo de planes curriculares el número de cursos erróneos del Programa de Estudios Básicos es ligeramente mayor a 75%, un índice demasiado alto, motivo por el cual debería elaborarse un algoritmo para la generación de la máscara, con la finalidad de automatizarlo y disminuir el porcentaje de error.

2. Con respecto al Centro de Computo

2.1. Debería emplear un equipo de personas destinado a verificar la calidad de la información, así como la creación y mantenimiento de un diccionario de datos, de haber contado con dicho personal muchas de las inconsistencias encontradas no debieron haber aparecido.

2.2. Implementar una herramienta para el procedimiento de captura de características sobresalientes en el instante que el alumno postule a la universidad, para facilitar la elaboración de información de mejor calidad con la finalidad de tener una mejor toma de decisiones.

2.3. Realizar la implementación de los algoritmos hallados, para corroborar en nuevos datos la validación positiva de los modelos para reforzar la prevención de un bajo rendimiento académico antes de que ocurra y valerse de esa

predicción temprana para poder retener a los alumnos mediante programas de asesoría o tutoría.

2.4. Se debería crear un área de analítica, dedicada exclusivamente a la búsqueda de *insights* que permita a la Universidad diferenciarse de sus competidores.

3. Con respecto a investigaciones futuras

3.1. Se propone el desarrollo de nuevos modelos predictivos que incorporen un mayor conjunto de variables con el objetivo de analizar esta misma problemática u otros problemas similares como la deserción estudiantil o el *churn* de pago de la pensión académica.

3.2. También se ha identificado que los datos contenidos en el aula virtual (tiempo de permanencia, horario de uso, frecuencia de ingreso, etc.) podrían ser material para desarrollar modelos predictivos en la búsqueda de conocer mejor a los alumnos y brindarles una mejor atención.

3.3. Asimismo, se puede señalar que falta un uso más adecuado de los datos que brinda la marcación digital de los docentes, los cuales se podrían utilizar para un control más adecuado, así como establecer un modelo relacional entre el rendimiento académico, el horario de clase y el profesor.

3.4. Adicionalmente se debería construir un *Dashboard*, porque cuando se despliegue el modelo en producción este generará información que debería alimentar un tablero de control, el cual permitiría visualizar si las acciones correctivas están brindando los resultados esperados.

REFERENCIAS BIBLIOGRÁFICAS

Abbott, D. (2014). *Applied Predictive Analytics.* Indianapolis, Indiana, USA: John Wiley & Sons, Inc.

Bell, J. (2015). *Machine Learning: Hands-On for Developers and Technical Professionals.* Indianapolis, Indiana, USA: John Wiley & Sons, Inc.

Cadenas, G. (6 de noviembre de 2015). *Smartick, matematicas a un click*. Recuperado el 8 de septiembre de 2018, de Series y Patrones: https://www.smartick.es/blog/matematicas/recursos-didacticos/series-y-patrones/

García Ortiz, Y., López de Castro Machado, D., & Rivero Frutos, O. (15 de febrero de 2014). *EDUMECENTRO, 6(2), 272-278*. Recuperado el 08 de septiembre de 2018, de Estudiantes universitarios con bajo rendimiento académico, ¿qué hacer?: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2077-28742014000200018&lng=es&tlng=es

Garreta, R., & Moncecchi, G. (2013). *Learning scikit-learn: Machine Learning in Python.* Birmingham, United Kingdom: Packt Publishing Ltd.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques - 3rd ed.* Waltham, Massachusetts, USA: Morgan Kaufmann Publishers.

Haykin, S. (2009). *Neural networks and learning machines - 3rd ed.* New Jersey, USA: Prentice Hall.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2014). *Metodología de la Investigación - 6ta ed.* México D.F., México: Mc Graw Hill Education.

Himmel, E. (2002). Modelo de análisis de la deserción estudiantil en la educación superior. *Calidad en la Educación* (17), 91-108. doi:https://doi.org/10.31619/caledu.n17.409

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* New York, USA: Springer.

Laudon, K., & Laudon, J. (2012). *Sistemas de Informacion Gerencial - 12va ed.* México D.F., México: Prentice Hall.

Ponce Cruz, P. (2010). *Inteligencia artificial con aplicaciones a la ingenieria.* México D.F., México: Alfaomega Grupo Editor.

Raschka, S. (30 de abril de 2014). *Mlxtend (machine learning extensions).* Recuperado el 9 de abril de 2019, de StackingClassifier: http://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/

Rodriguez Pacheco, E. (2015). *Unsupervised Learning with R.* Birmingham, United Kingdom: Packt Publishing Ltd.

Russell, S., & Norvig, P. (2010). *Artificial Intelligence A Modern Approach - 3rd ed.* New Jersey, USA: Prentice Hall.

SINEACE. (2013). *Educación Superior en el Perú: Retos para el aseguramiento de la calidad.* Lima, Perú: Impresión Arte Perú S.A.C.

Witten, I., Frank, E., Hall, M., & Pal, C. J. (2017). *Data Mining, Practical Machine Learning Tools and Techniques - 4th ed.* Cambridge, Massachusetts, USA: Morgan Kaufmann Publishers.

ANEXOS

Anexo 01: Declaración de Autenticidad (según formato adjunto).

Anexo 02: Autorización de consentimiento para realizar la investigación (según formato adjunto).

Anexo 03: Matrices Adicionales

Anexo 03.1: Matriz de Consistencia.

Problema	Objetivo	Hipótesis	Marco Teórico	Metodología y Variables
Problema general: ¿Cómo hacer uso de los algoritmos de *Machine Learning* para predecir la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma? **Problemas específicos:** A. ¿Cómo utilizar el algoritmo	**Objetivo general:** Pronosticar la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma mediante el uso de algoritmos de *Machine Learning*. **Objetivos específicos:** A. Determinar la efectividad del uso	**Hipótesis general:** Si se aplican los algoritmos de *Machine Learning*, entonces se incrementará la eficiencia en la predicción de la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma. **Hipótesis específicas:** A. Si se aplica el algoritmo de Redes	**Antecedentes:** **A nivel Internacional:** **Fischer, E. (2012)**. *Modelo para la automatización del proceso de determinación de riesgo de deserción en alumnos universitarios*. Tesis para obtener el grado académico de Magíster en Tecnologías de la Información. Departamento de Ciencias de la Computación, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile. Santiago de Chile, Chile. **Hernández, J. (2015)**. *Modelo de Minería de Datos para Identificación de Patrones que Influyen en el Aprovechamiento Académico.* Tesis para obtener el grado académico de Maestro en Sistemas Computacionales. Maestría en Sistemas Computacionales, División de Estudios de Posgrado e Investigación, Instituto Tecnológico de La Paz, Tecnológico Nacional de México. La	**Tipo de Enfoque:** Cuantitativo y Aplicado **Método:** Alcance Descriptivo y Correlacional **Tipo de Diseño:** No experimental

de Redes Neuronales Artificiales (RNA) para predecir la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma?	del algoritmo de Redes Neuronales Artificiales (RNA) para pronosticar la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.	Neuronales Artificiales (RNA), entonces se incrementará la eficiencia en la predicción de la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.	Paz, Baja California Sur, México. **Aftab, J. (2017).** *Student Retention in Higher Education Institutions*. Tesis para obtener el grado académico de Master of Science in Computer Science, Department of Computer Science. Capital University of Science and Technology (CUST). Islamabad, Pakistán. **A nivel nacional:**	**Variables:** **Variable independiente:** Algoritmos de *Machine Learning* **Indicador:** Tasa de Aciertos
B. ¿Cómo utilizar el algoritmo *Boosting* para predecir la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma?	B. Determinar la efectividad del uso del algoritmo *Boosting* para pronosticar la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.	B. Si se aplica el algoritmo *Boosting*, entonces se incrementará la eficiencia en la predicción de la cantidad de alumnos aprobados y desaprobados en los cursos del Programa de Estudios Básicos de la Universidad Ricardo Palma.	**Acosta, P. et all (2011).** *Predicción del rendimiento académico en la educación Superior usando minería de datos y su comparación con Técnicas estadística.* Tesis para obtener el grado académico de Maestro en Ciencias con mención en Ingeniería de Sistemas. Sección de Postgrado, Facultad de Ingeniería Industrial y de Sistemas, Universidad Nacional de Ingeniería. Lima, Perú. **Pacco, R. (2015).** *Análisis Predictivo Basado en Redes Neuronales no Supervisadas Aplicando algoritmo de K-medias y CRISP-DM para Pronóstico de Riesgo de Morosidad de los Alumnos en la Universidad Peruana Unión.* Tesis para obtener el grado académico de Magíster en Ingeniería de Sistemas con mención en Dirección y Gestión de	**Variable dependiente:** Rendimiento Académico de los Alumnos en el Programa de Estudios Básicos de la Universidad Ricardo Palma **Indicadores:** Nota de cada alumno en escala vigesimal, Número de veces que ha llevado el curso y Promedio ponderado. Escala de pago Masculino o
C. ¿Cuál de los dos algoritmos	C. Especificar y evaluar la Tasa de	C. Al comparar el nivel de precisión		

brindará mejores pronósticos?	Acierto de ambos algoritmos.	de ambos algoritmos se puede identificar el más óptimo para la problemática planteada.	Tecnologías de la Información. Maestría en Ingeniería de Sistemas, Escuela de Posgrado Universidad Peruana Unión. Lima, Perú. **Bases Teóricas:** *Machine Learning*. Redes Neuronales Artificiales (RNA). *Boosting*. **Definición de términos:** Algoritmo. RNA. *Boosting*. **Bibliografía:** Metodología de la Investigación, 6ta edición (Hernández Sampieri, Roberto, 2014) Neural networks and learning machines - 3rd ed. (Haykin, S., 2009) Unsupervised Learning with R (Rodriguez Pacheco, E., 2015)	Femenino Fecha de nacimiento Ciclo donde ha estudiado y número de aprobaciones

Elaboración: Propia, 2019

Anexo 03.2: Matriz de Validación del Instrumento.

Validación del Instrumento: Base de Datos			
Dato a encontrar	**Utilidad**		**Motivo**
Código del Alumno	✘		
Carrera	✓	Obtener la relación entre Carrera y Nota Final del Curso	Determinar la influencia de la variable para obtener un rendimiento académico aprobatorio
Sexo	✓	Obtener la relación entre Sexo y Nota Final del Curso	Determinar la influencia de la variable para obtener un rendimiento académico aprobatorio
Modalidad de Ingreso	✓	Obtener la relación entre Modalidad de Ingreso y Nota Final del Curso	Determinar la influencia de la variable para obtener un rendimiento académico aprobatorio
Escala de Pago	✓	Obtener la relación entre Escala de Pago y Nota Final del Curso	Determinar la influencia de la variable para obtener un rendimiento académico aprobatorio
Fecha de Nacimiento	✘		
Nombre del Colegio	✘		
Tipo de Colegio	✓	Obtener la relación entre Tipo de Colegio y Nota Final del Curso	Determinar la influencia de la variable para obtener un rendimiento académico aprobatorio
Nota Final del Curso Pre-Requisito	✓	Obtener la relación entre Nota Final del Curso Pre-Requisito y Nota Final del Curso	Determinar la influencia de la Nota Final del Curso Pre-Requisito en la obtención de un rendimiento académico aprobatorio
Primera Evaluación	✓	Obtener la relación entre la Primera Evaluación y Nota Final del Curso	Determinar la influencia de la Primera Evaluación del Curso en la obtención de un rendimiento académico aprobatorio
Otras Evaluaciones	✘		
Nota Final del Curso	✓	Conocer el Rendimiento Académico del alumno: Aprobado o Desaprobado	Es el objetivo a evaluar

Elaboración: Propia, 2019

Anexo 04: Script del algoritmo en R para la comprensión de cada uno de los Archivos de Datos.

Anexo 05: Script del algoritmo en Python para la transformación del Archivo de Notas.

Anexo 06: Script del algoritmo en R para la comprensión de cada curso del Programa de Estudios Básicos (PEB).

Asimismo, se visualizarán únicamente los resultados correspondientes al *dataset* del curso con el código "0001" denominado "Actividades Artísticas y Deportivas".

Anexo 07: Script del algoritmo en R para la preparación de cada curso del Programa de Estudios Básicos (PEB).

Asimismo, se visualizarán únicamente los resultados correspondientes al *dataset* del curso con el código "0001" denominado "Actividades Artísticas y Deportivas".

Anexo 08: Script del algoritmo en R para la creación del *dataset* de cada curso del Programa de Estudios Básicos (PEB).

Asimismo, se visualizarán únicamente los resultados correspondientes al *dataset* del curso con el código "0001" denominado "Actividades Artísticas y Deportivas".

Anexo 09: Script del algoritmo en R para el *tuning* de los modelos de prueba de cada técnica de modelado para cada curso del Programa de Estudios Básicos (PEB).

Asimismo, se visualizarán únicamente los resultados correspondientes al *dataset* del curso con el código "0001" denominado "Actividades Artísticas y Deportivas".

Anexo 10: Script del algoritmo en R para la generación de los modelos de cada curso del Programa de Estudios Básicos (PEB).

Asimismo, se visualizarán únicamente los resultados correspondientes al *dataset* del curso con el código "0001" denominado "Actividades Artísticas y Deportivas".

Anexo 11: Resultados de la comprensión de cada curso del Programa de Estudios Básicos (PEB).

Exclusivamente se visualizarán los resultados desde el curso con el código "0002" hasta el curso con el código "0013".

Anexo 12: Resultados de la preparación de cada curso del Programa de Estudios Básicos (PEB).

Exclusivamente se visualizarán los resultados desde el curso con el código "0002" hasta el curso con el código "0013".

Anexo 13: Resultados de la creación del *dataset* de cada curso del Programa de Estudios Básicos (PEB).

Exclusivamente se visualizarán los resultados desde el curso con el código "0002" hasta el curso con el código "0013".

Anexo 14: Resultados del *tuning* de los modelos de prueba de cada técnica de modelado para cada curso del Programa de Estudios Básicos (PEB).

Exclusivamente se visualizarán los resultados desde el curso con el código "0002" hasta el curso con el código "0013".

Anexo 15: Resultados de la generación de los modelos de cada curso del Programa de Estudios Básicos (PEB).

Exclusivamente se visualizarán los resultados desde el curso con el código “0002” hasta el curso con el código “0013”.

Anexo 04: Script del algoritmo en R para la comprensión de cada uno de los Archivos de Datos.

Procederemos a leer y revisar el contenido de cada uno de los Archivos de Datos.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())

# Uso de Librerías
library(VIM)
```

---

1. Archivo de Datos: Carreras

Verificación de la estructura interna del nuevo archivo.

```
data <- read.csv("carrera_a.csv",
                 colClasses = c("factor", "factor"),
                 col.names = c("car_cod", "car_des"))
str(data)
```

```
## 'data.frame':    18 obs. of  2 variables:
##  $ car_cod: Factor w/ 18 levels "11","21","25",..: 1 2 3 4 5 6 7 8 9 10 ..
##  $ car_des: Factor w/ 18 levels "Administración de Negocios Globales",..:
3 4 14 15 7 2 5 1 18 13 ...
```

```
list(data)
```

```
## [[1]]
##    car_cod                                    car_des
## 1       11                                Arquitectura
## 2       21                                    Biología
## 3       25                              Medicina Humana
## 4       27                          Medicina Veterinaria
## 5       31                                     Economía
## 6       32                    Administración y Gerencia
## 7       33                      Contabilidad y Finanzas
## 8       34          Administración de Negocios Globales
## 9       35              Turismo,Hotelería y Gastronomía
## 10      38 Marketing Global y Administración Comercial
## 11      41                                   Psicología
## 12      46                                     Derecho
## 13      51                  Traducción e Interpretación
## 14      61                             Ingeniería Civil
## 15      62                  Ingeniería Electrónica
## 16      63                   Ingeniería Industrial
## 17      66                  Ingeniería Informática
## 18      68                  Ingeniería Mecatrónica
```

---

2. Archivo de Datos: Modalidad de Ingreso

Visualización de la estructura interna del archivo original.

```
data <- read.csv("Ingreso.csv",
                 colClasses = c("factor", "factor"),
                 col.names = c("ing_cod", "ing_des"))
str(data)
```

```
## 'data.frame':    21 obs. of  2 variables:
##  $ ing_cod: Factor w/ 21 levels "00","01","02",..: 1 2 3 4 5 6 7 8 9 10 ..
##  $ ing_des: Factor w/ 21 levels "ALUMNO LIBRE",..: 1 2 3 4 6 5 7 8 9 14 ..
```

```
list(data)
```

```
## [[1]]
##    ing_cod                              ing_des
## 1       00                         ALUMNO LIBRE
## 2       01                         BACHILLERATO
## 3       02     BACHILLERATO ESCOLAR INTERNACIONAL
## 4       03                   BECA 18 - PRONABEC
## 5       04                  CEPURP CICLO REGULAR
## 6       05               CEPURP APTITUD ACADEMICA
## 7       06 CEPURP ESCOLARES DEL 5TO DE SECUNDARIA
## 8       07 CEPURP ESPECIAL PARA ESCOLARES AGO-ENE
## 9       08           CEPURP INTENSIVO ENERO-MARZO
## 10      09                          DEPORTISTAS
## 11      10                         DIPLOMATICOS
## 12      11                GRADUADO Y/O TITULADO
## 13      12                     PRIMEROS PUESTOS
## 14      13           EXAMEN DE APTITUD ACADEMICA
## 15      14           COBERTURA APTITUD ACADEMICA
## 16      15            EXAMEN GENERAL DE ADMISION
## 17      16              COBERTURA EXAMEN GENERAL
## 18      17                    EXAMEN PROMOCIONAL
## 19      18           COBERTURA EXAMEN PROMOCIONAL
## 20      19                     TRASLADO EXTERNO
## 21      20           COBERTURA - TRASLADO EXTERNO
```

---

3. Archivo de Datos: Alumnos

Visualización de la estructura interna del archivo original.

```
data <- read.csv("alumnos.csv",
                 colClasses = c("character", "factor", "factor", "Date",
                                "factor", "factor", "factor", "factor"),
                 col.names = c("alu_cod", "car_cod", "sexo", "nacio",
                               "ing_cod", "escala", "nom_cole", "tipo_cole"))
str(data)
```

```
## 'data.frame':    9118 obs. of  8 variables:
##  $ alu_cod  : chr  "201510001" "201510002" "201510003" "201510004" ...
##  $ car_cod  : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1..
##  $ sexo     : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio    : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod  : Factor w/ 21 levels "00","01","02",..: 7 7 7 7 7 7 7 7 7 7 .
```

```
##  $ escala   : Factor w/ 7 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3 3
##  $ nom_cole : Factor w/ 1939 levels "-","0009 JOSE MARIA ARGUEDAS",..: 184
4 1852 1325 1036 1261 1480 1789 1578 921 1822 ...
##  $ tipo_cole: Factor w/ 3 levels "E","N","P": 3 3 3 3 3 1 3 3 3 3 ...
```

```
summary(data)
```

```
##    alu_cod             car_cod     sexo         nacio
##  Length:9118        11     :1343   F:4430   Min.   :1957-03-10
##  Class :character   61     :1276   M:4688   1st Qu.:1997-04-27
##  Mode  :character   25     :1026            Median :1998-09-08
##                     63     : 864            Mean   :1998-03-23
##                     51     : 681            3rd Qu.:1999-12-14
##                     41     : 571            Max.   :2004-05-03
##                     (Other):3357            NA's   :19
##     ing_cod      escala                    nom_cole    tipo_cole
##  13     :3335   A13 : 882   TRILCE             : 294   E:1961
##  15     :2829   A18 : 144   PAMER              : 234   N:  43
##  17     :1271   A23 :5414   SACO OLIVEROS      : 228   P:7114
##  04     : 386   A28 : 275   MANUEL POLO JIMENEZ: 201
##  05     : 305   A33 :2024   TRILCE MARSANO     : 132
##  19     : 240   A38 : 336   SACO OLIVEROS S.J.M: 112
##  (Other): 752   NULL:  43   (Other)            :7917
```

```
head(data, 10)
```

```
##      alu_cod car_cod sexo      nacio ing_cod escala
## 1  201510001      11    M 1998-07-21      06    A23
## 2  201510002      33    F 1998-06-27      06    A33
## 3  201510003      11    F 1997-12-16      06    A23
## 4  201510004      11    F 1998-01-02      06    A23
## 5  201510005      11    F 1998-05-12      06    A23
## 6  201510007      11    M 1997-09-13      06    A23
## 7  201510008      11    F 1998-03-27      06    A23
## 8  201510009      11    F 1998-03-06      06    A23
## 9  201510010      11    F 1998-09-14      06    A23
## 10 201510011      11    M 1998-07-19      06    A23
##                     nom_cole tipo_cole
## 1      TRILCE DE SAN MIGUEL         P
## 2            TRILCE MARSANO         P
## 3                   PIO XII         P
## 4       MANUEL POLO JIMENEZ         P
## 5                     PAMER         P
## 6  SAGRADO CORAZON DE BELEN         E
## 7                   SOR INÉS         P
## 8         SAN JOSE DE CLUNY         P
## 9                   LA SALLE         P
## 10                    TRILCE         P
```

Verificación de la estructura interna del nuevo archivo.

```
data <- read.csv("alumnos_a.csv",
                 colClasses = c("character", "factor", "factor",
                                "Date", "factor", "factor", "factor"),
                 col.names = c("alu_cod", "car_cod", "sexo",
                               "nacio", "ing_cod", "escala", "col_tipo"))
str(data)
```

```
## 'data.frame':    9075 obs. of  7 variables:
##  $ alu_cod : chr  "201510001" "201510002" "201510003" "201510004" ...
##  $ car_cod : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1 ..
##  $ sexo    : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio   : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6 ..
##  $ escala  : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3 3
##  $ col_tipo: Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
```

```
summary(data)
```

```
##    alu_cod             car_cod     sexo         nacio
##  Length:9075        11     :1343   F:4400   Min.   :1957-03-10
##  Class :character   61     :1276   M:4675   1st Qu.:1997-05-02
##  Mode  :character   25     :1026            Median :1998-09-13
##                     63     : 863            Mean   :1998-03-26
##                     51     : 651            3rd Qu.:1999-12-16
##                     41     : 569            Max.   :2004-05-03
##                     (Other):3347            NA's   :18
##     ing_cod     escala     col_tipo
##  13     :3335   A13: 882   E:1961
##  15     :2829   A18: 144   P:7114
##  17     :1271   A23:5414
##  04     : 386   A28: 275
##  05     : 305   A33:2024
##  19     : 240   A38: 336
##  (Other): 709
```

```
head(data, 10)
```

```
##      alu_cod car_cod sexo      nacio ing_cod escala col_tipo
## 1  201510001      11    M 1998-07-21      06    A23        P
## 2  201510002      33    F 1998-06-27      06    A33        P
## 3  201510003      11    F 1997-12-16      06    A23        P
## 4  201510004      11    F 1998-01-02      06    A23        P
## 5  201510005      11    F 1998-05-12      06    A23        P
## 6  201510007      11    M 1997-09-13      06    A23        E
## 7  201510008      11    F 1998-03-27      06    A23        P
## 8  201510009      11    F 1998-03-06      06    A23        P
## 9  201510010      11    F 1998-09-14      06    A23        P
## 10 201511011      11    M 1998-07-19      06    A23        P
```

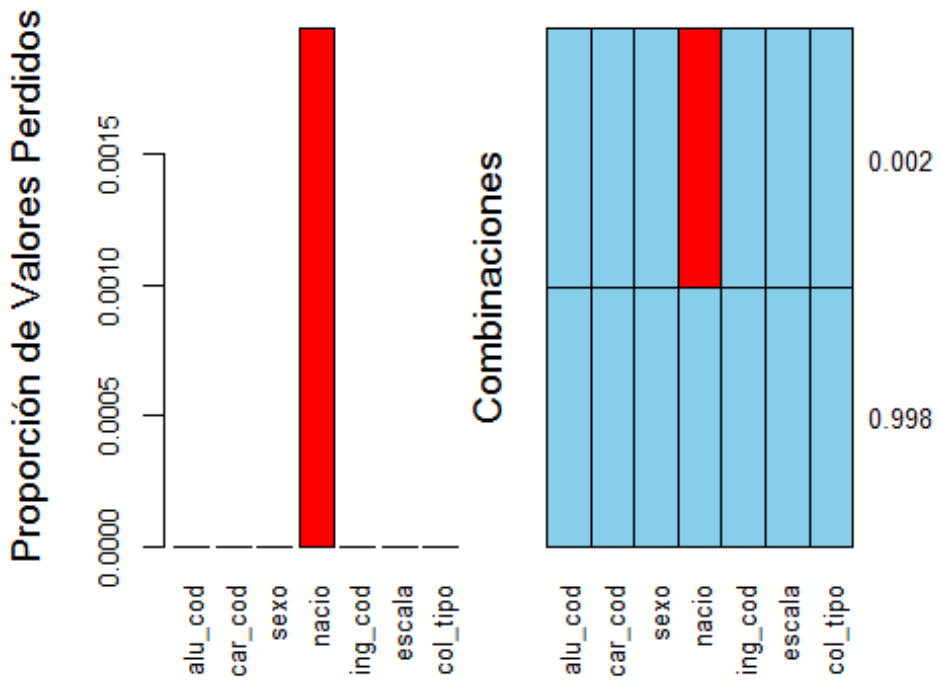
```
# Columnas donde hay datos faltantes
(miss_col = which(colSums(is.na(data)) != 0))
```

```
## nacio
##     4
```

```
# Filas donde hay datos faltantes
(miss_row = which(rowSums(is.na(data)) != 0, arr.ind = T))
```

```
##  [1] 431 432 434 438 439 440 443 444 446 447 448 452 454 457 458 460 463
## [18] 464
```

```
# Donde se encuentran los datos faltantes
summary(aggr(data,
             bars = F,
             numbers = T,
             ylabs = c("Proporción de Valores Perdidos", "Combinaciones"),
             cex.axis = 0.8, cex.numbers = 0.8,
             gap = 2))
```




```
##
##  Missings per variable:
##  Variable Count
##   alu_cod     0
##   car_cod     0
##      sexo     0
##     nacio    18
##   ing_cod     0
##    escala     0
##  col_tipo     0
##
##  Missings in combinations of variables:
##   Combinations Count    Percent
##  0:0:0:0:0:0:0  9057 99.8016529
##  0:0:0:1:0:0:0    18  0.1983471
```

```r
# Proceso para eliminar los datos faltantes
data_1     <- data
data_1[,4] <- as.factor(data_1[,4])
data_1     <- kNN(data_1, k = 10)
data_1     <- data_1[, 0:dim(data)[2]]
data_1[,4] <- as.Date(data_1$nacio)

# Comprobación de eliminación de datos faltantes
head(data[c(miss_row),], 10)
```

```
##       alu_cod car_cod sexo nacio ing_cod escala col_tipo
## 431 201511146      68    M  <NA>      03    A23        E
## 432 201511150      11    M  <NA>      03    A23        P
## 434 201511211      61    M  <NA>      03    A23        E
## 438 201511281      61    M  <NA>      03    A23        P
## 439 201511288      66    F  <NA>      03    A23        P
## 440 201511306      61    F  <NA>      03    A23        P
## 443 201511321      11    F  <NA>      03    A23        E
## 444 201511331      41    F  <NA>      03    A33        P
```

```
## 446 201511338      61    F  <NA>      03    A23        E
## 447 201511339      61    M  <NA>      03    A23        E
```

```
head(data_1[c(miss_row),], 10)
```

```
##       alu_cod car_cod sexo      nacio ing_cod escala col_tipo
## 431 201511146      68    M 1999-01-30      03    A23        E
## 432 201511150      11    M 1999-01-12      03    A23        P
## 434 201511211      61    M 1998-02-22      03    A23        E
## 438 201511281      61    M 1994-07-30      03    A23        P
## 439 201511288      66    F 1997-03-18      03    A23        P
## 440 201511306      61    F 1998-12-08      03    A23        P
## 443 201511321      11    F 1997-12-19      03    A23        E
## 444 201511331      41    F 1998-03-01      03    A33        P
## 446 201511338      61    F 1997-10-10      03    A23        E
## 447 201511339      61    M 1999-01-30      03    A23        E
```

```
str(data_1)
```

```
## 'data.frame':    9075 obs. of  7 variables:
##  $ alu_cod : chr  "201510001" "201510002" "201510003" "201510004" ...
##  $ car_cod : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1 ..
##  $ sexo    : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio   : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6 ..
##  $ escala  : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3 3
##  $ col_tipo: Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
```

```
summary(data_1)
```

```
##    alu_cod             car_cod     sexo         nacio
##  Length:9075        11     :1343   F:4400   Min.   :1957-03-10
##  Class :character   61     :1276   M:4675   1st Qu.:1997-05-02
##  Mode  :character   25     :1026            Median :1998-09-12
##                     63     : 863            Mean   :1998-03-26
##                     51     : 651            3rd Qu.:1999-12-14
##                     41     : 569            Max.   :2004-05-03
##                     (Other):3347
##     ing_cod     escala     col_tipo
##  13     :3335   A13: 882   E:1961
##  15     :2829   A18: 144   P:7114
##  17     :1271   A23:5414
##  04     : 386   A28: 275
##  05     : 305   A33:2024
##  19     : 240   A38: 336
##  (Other): 709
```

```
# Grabación del nuevo archivo de alumnos
write.csv(data_1,
          file = "alumnos_b.csv",
          row.names = FALSE)
```

---

4. Archivo de Datos: Planes Curriculares

Exploración, depuración y filtrado del archivo original.

```r
data <- read.csv("Planes.csv")

# Convirtiendo las columnas necesarias a factores
data[, 1] <- as.factor(data[, 1])
data[, 6] <- as.factor(data[, 6])
str(data)
```

```
## 'data.frame':    4840 obs. of  14 variables:
##  $ CCARR_CODIGO        : Factor w/ 19 levels "11","21","25",..: 8 8 8 8 8
8 8 8 8 8 ...
##  $ CCURR_CODIGO        : Factor w/ 5 levels "50  ","51  ",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ CCURS_CODIGO        : Factor w/ 580 levels "0001","0002",..: 109 120 12
6 129 269 271 272 1 258 259 ...
##  $ CCURS_MASCARA       : Factor w/ 3155 levels "AC-A001","AC-E001",..: 116
3 1164 1166 1168 2471 2472 2473 1136 1171 1172 ...
##  $ CCURS_NOMBRE        : Factor w/ 2137 levels "Abastecimiento de Agua",..
: 1832 1746 1233 1161 50 596 365 6 1530 727 ...
##  $ CCURS_NCICLO        : Factor w/ 14 levels "1","2","3","4",..: 1 1 1 1 1
1 1 2 2 2 ...
##  $ CCURS_TIPOCURSO     : Factor w/ 4 levels "0","E","O","P": 3 3 3 3 3 3 3
3 3 3 ...
##  $ FCURS_CREDITO       : num  2 2 4 3 4 3 3 1 3 3 ...
##  $ ICURS_HRSTEORIA     : int  0 0 2 2 2 2 2 0 2 3 ...
##  $ ICURS_HRSPRACTICA   : int  0 0 4 2 4 2 2 2 2 0 ...
##  $ ICURS_HRSLABORATORIO: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ICURS_HRSTALLER     : int  4 4 0 0 0 0 0 0 0 0 ...
##  $ CPRECURS_CODIGO     : Factor w/ 393 levels "_   ","0001",..: 1 1 1 1 1
1 1 1 1 85 ...
##  $ FCURS_CREDREQTOTALES: int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
# Generando tabla de la columna 2 (Planes Curriculares)
# Podemos observar los distintos Planes Curriculares
# Los Planes Curriculares vigentes son: “50”, “51” y “52”
table(data[, 2])
```

```
##
## 50   51   52   53   A
## 1587  355 1539   82 1277
```

```r
# Generando tabla de la columna 4 (Mascara del Curso)
# Cada primera letra corresponde a un departamento Academoa
# La letra "E" corresponde al departamento del "Programa de Estudios Básicos"
table(substr(data[, 4], 1, 1))
```

```
##
##   A   C   D   E   I   M   N   P   T
## 478 870 205 682 881 660 315 200 549
```

```r
# Eliminando las columnas y filas innecesarias
data_1 <- data[substr(data[, 4], 1, 1) == "E", c(1:5, 13, 14)]
data_1 <- data_1[strtrim(data_1[, 2], 1) != "A", ]
data_1 <- data_1[strtrim(data_1[, 2], 2) != "53", ]

# Verificando Planes Curriculares
table(data_1[, 2])
```

```
##
## 50   51   52   53   A
##  195  121  237    0    0
```

```r
# Verificando Areas Academicas
table(substr(data_1[, 4], 1, 1))
```

```
##
##   E
## 553
```

```r
# Verificando el Prerrequisito en la Cantidad de créditos aprobados
table(data_1[, 7])
```

```
##
##   0
## 553
```

```r
# En vista del resultado anterior eliminamos dicha columna
# Tambien eliminamos la columna de curricula, debido a que hemos filtrado los registros que necesitamos
# Tambien eliminamos la columna de carrera, porque el contenido de cada curso del PEB es igual indistintamente de la carrera
data_1 <- data_1[, c(3:6)]

# Cambiamos el nombre de las columnas
names(data_1) <- c("cur_cod", "mascara", "cur_nom", "cur_pre_cod")

# Revisando su estructura
str(data_1)
```

```
## 'data.frame':    553 obs. of  4 variables:
##  $ cur_cod    : Factor w/ 580 levels "0001","0002",..: 109 120 126 129 1 2 58 259 263 12 13 ...
##  $ mascara    : Factor w/ 3155 levels "AC-A001","AC-E001",..: 1163 1164 1166 1168 1136 1171 1172 1175 1146 1147 ...
##  $ cur_nom    : Factor w/ 2137 levels "Abastecimiento de Agua",..: 1832 1746 1233 1161 6 1530 727 777 1569 1564 ...
##  $ cur_pre_cod: Factor w/ 393 levels "_   ","0001",..: 1 1 1 1 1 1 85 1 1 139 ...
```

```r
head(data_1, 10)
```

```
##    cur_cod mascara                                  cur_nom cur_pre_cod
## 1      101  EB0101  Taller de Método de Estudio Universitario        _
## 2      102  EB0102      Taller de Comunicación Oral y Escrita        _
## 3     1032 EB01032                              Matemática I        _
## 4      104  EB0104                                    Lógica        _
## 8     0001 EB 0001       Actividades Artísticas y Deportivas        _
## 9      202  EB0202                         Psicología General        _
## 10     203  EB0203                                   Filosofía      104
## 11    2060 EB02060              Formación Histórica del Perú        _
## 17    0011 EB 0011       Recursos Naturales y Medio Ambiente        _
## 18    0012 EB 0012                        Realidad Nacional     2060
```

```r
# Grabación del nuevo archivo de planes curriculares
write.csv(data_1,
          file = "planes_a.csv",
          row.names = FALSE)
```

Verificación de la estructura interna del nuevo archivo de planes curriculares.

```
data <- read.csv("planes_b.csv",
                 colClasses = c("factor", "factor", "character", "factor"))
str(data)
```

```
## 'data.frame':    13 obs. of  4 variables:
##  $ cur_cod    : Factor w/ 13 levels "0001","0002",..: 1 2 3 4 5 6 7 8 9 10
##  $ mascara    : Factor w/ 13 levels "EB 0001","EB 0002",..: 1 2 3 4 5 6 7
8 9 10 ...
##  $ cur_nom    : chr  "Actividades Artísticas y Deportivas" "Taller de Méto
do de Estudio Universitario" "Taller de Comunicación Oral y Escrita I" "Matem
ática" ...
##  $ cur_pre_cod: Factor w/ 6 levels "","0001","0002",..: 1 1 1 1 1 4 3 4 5
3 ...
```

```
list(data)
```

```
## [[1]]
##    cur_cod mascara                                    cur_nom cur_pre_cod
## 1     0001 EB 0001       Actividades Artísticas y Deportivas
## 2     0002 EB 0002 Taller de Método de Estudio Universitario
## 3     0003 EB 0003   Taller de Comunicación Oral y Escrita I
## 4     0004 EB 0004                                Matemática
## 5     0005 EB 0005                                  Inglés I
## 6     0006 EB 0006                     Psicología General        0003
## 7     0007 EB 0007                     Lógica y Filosofía        0002
## 8     0008 EB 0008  Taller de Comunicación Oral y Escrita II        0003
## 9     0009 EB 0009                                 Inglés II        0005
## 10    0010 EB 0010             Formación Histórica del Perú        0002
## 11    0011 EB 0011       Recursos Naturales y Medio Ambiente        0001
## 12    0012 EB 0012                      Realidad Nacional        0010
## 13    0013 EB 0013              Historia de la Civilización        0010
```

Visualización de la estructura interna del archivo de equivalencias.

```
data <- read.csv("equi_cursos.csv",
                 colClasses = c("factor", "factor"))
str(data)
```

```
## 'data.frame':    53 obs. of  2 variables:
##  $ mascara: Factor w/ 53 levels "E B0001","E B0003",..: 9 35 8 1 18 47 46
28 21 42 ...
##  $ cur_cod: Factor w/ 13 levels "0001","0002",..: 1 1 1 1 10 10 10 10 13 1
3 ...
```

```
head(data, 10)
```

```
##    mascara cur_cod
## 1  EB 0001    0001
## 2   EB0011    0001
## 3  EB-0011    0001
## 4  E B0001    0001
## 5  EB 0010    0010
## 6  EB02060    0010
## 7   EB0206    0010
## 8  EB 0206    0010
```

```
## 9  EB 0013    0013
## 10  EB0105    0013
```

---

5. Archivo de Datos: Tipo de Evaluación

Visualización de la estructura interna del archivo original.

```
data <- read.csv("Evaluacion.csv",
                 colClasses = c("factor", "character"),
                 col.names = c("eva_cod", "eva_des"))
str(data)
```

```
## 'data.frame':    29 obs. of  2 variables:
##  $ eva_cod: Factor w/ 28 levels "ASP","AVP","CPX",..: 1 2 3 4 5 6 7 8 9 9
##  $ eva_des: chr  "Asistencia y Puntualidad" "Avance del Proyecto" "Concurs
o de Proyecto" "Control Laboratorio" ...
```

```
list(data)
```

```
## [[1]]
##    eva_cod                      eva_des
## 1      ASP     Asistencia y Puntualidad
## 2      AVP          Avance del Proyecto
## 3      CPX         Concurso de Proyecto
## 4      CTL          Control Laboratorio
## 5      DCA Demostración del Aprendizaje
## 6      EXP                   Exposición
## 7      EXV                  Expovitrina
## 8      FIN                        Final
## 9      INF                      Informe
## 10     INF                Informe Final
## 11     INT               Informe Taller
## 12     LAB                  Laboratorio
## 13     LAM                       Lamina
## 14     NPA           Nota Participación
## 15     PAD         Participación Activa
## 16     PAR                      Parcial
## 17     PRA                     Práctica
## 18     PRO                     Proyecto
## 19     PRT             Práctica Teórica
## 20     PTL              Práctica Taller
## 21     PYF               Proyecto Final
## 22     PYL      Proyecto de Laboratorio
## 23     PYT              Proyecto Taller
## 24     SUP        Sustentación Proyecto
## 25     SUS                 Sustitutorio
## 26     TLR                       Taller
## 27     TMO          Trabajo Monográfico
## 28     TRA     Trabajo de Investigación
## 29     TRP             Trabajo Práctico
```

---

6. Archivo de Datos: Notas

Visualización de la estructura interna del archivo original de todos los cursos.

```
data <- read.csv("notas.csv",
                 colClasses = c("factor", "factor", "factor", "factor",
                                "factor", "factor", "character", "factor"))
str(data)
```

```
## 'data.frame':    574283 obs. of  8 variables:
##  $ Sem..Notas : Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1
1 ...
##  $ Cod..Curso : Factor w/ 29 levels "E B0001","E B0003",..: 4 4 4 4 19 19
19 19 19 19 ...
##  $ Fórmula    : Factor w/ 54 levels "(((((PRA1+PRA2+PRA3+PRA4+NPA1)/5)*2)+
PAR1+FIN1+TRA1)/5)",..: 51 51 51 51 18 18 18 18 18 18 ...
##  $ Eval       : Factor w/ 29 levels "FIN1  ","NPA1  ",..: 14 15 16 17 14 1
5 16 17 22 23 ...
##  $ Grupo      : Factor w/ 95 levels "01","02","03",..: 61 61 61 61 52 52 5
2 52 52 52 ...
##  $ Nota       : Factor w/ 23 levels "0.00","1.00",..: 1 1 1 1 8 7 4 2 17 3
##  $ Cod..Alumno: chr  "201512432" "201512432" "201512432" "201512432" ...
##  $ Nota.1     : Factor w/ 22 levels "00","01","02",..: 22 22 22 22 10 10 1
0 10 10 10 ...
```

```
summary(data)
```

```
##    Sem..Notas      Cod..Curso
##  20181  :94232   EB 0004: 56864
##  20171  :87569   EB 0002: 44645
##  20172  :84572   EB 0006: 41140
##  20182  :81800   EB 0011: 35907
##  20162  :70913   EB 0007: 34867
##  20161  :68624   EB 0010: 33397
##  (Other):86573   (Other):327463
##                                                 Fórmula            Eval
##  (((PRA1+PRA2+PRA3)/3)+PAR1+FIN1)/3                 : 79930   PRA1   : 81491
##  (PRA1+PRA2+PRA3+PRA4)/4                            : 65387   PRA2   : 81418
##  (PAR1+FIN1+(PRA1+PRA2+PRA3+PRA4)/3)/3              : 64091   PRA3   : 81370
##  ((PRA1+PRA2+PRA3+PRA4)/4+PAR1+FIN1)/3              : 48127   PRA4   : 62500
##  (PRA1+PRA2+PRA3+PRA4+PRA5)/5                       : 47943   FIN1   : 52002
##  (TRA1+PAR1+FIN1+2*((PRA1+PRA2+PRA3+PRA4)/4))/5: 41581   PAR1   : 52001
##  (Other)                                            :227224   (Other):163501
##      Grupo             Nota        Cod..Alumno            Nota.1
##  01     : 23356   0.00   : 51755   Length:574283      11     : 79307
##  02     : 22572   14.00  : 50645   Class :character   12     : 74223
##  03     : 20567   15.00  : 48766   Mode  :character   13     : 68576
##  04     : 19179   16.00  : 45220                      14     : 62276
##  05     : 18663   13.00  : 43063                      15     : 55268
##  06     : 18046   12.00  : 42729                      16     : 41804
##  (Other):451900   (Other):292105                      (Other):192829
```

```
head(data_1, 10)
```

```
##    cur_cod mascara                                    cur_nom cur_pre_cod
## 1      101  EB0101  Taller de Método de Estudio Universitario           _
## 2      102  EB0102      Taller de Comunicación Oral y Escrita           _
## 3     1032 EB01032                              Matemática I             _
```

```
## 4     104  EB0104                                  Lógica        _
## 8     0001 EB 0001      Actividades Artísticas y Deportivas        _
## 9     202  EB0202                  Psicología General             _
## 10    203  EB0203                           Filosofía             104
## 11    2060 EB02060           Formación Histórica del Perú        _
## 17    0011 EB 0011      Recursos Naturales y Medio Ambiente       _
## 18    0012 EB 0012                   Realidad Nacional            2060
```

Anexo 05: Script del algoritmo en Python para la transformación del Archivo de Notas.

Procederemos a revisar el Archivo de Notas con la finalidad de poder desdoblarlo en 13 archivos, cada uno de cuales corresponde a un curso del PEB.

```
# Uso de Librerías
import csv
```

---

1. Extracción de alumnos del Archivo de Notas

Procederemos a revisar el Archivo de Notas y generaremos una relación única de códigos de alumnos.

```
# Obteniendo la relación de codigos de alumnos
lista = []
with open("Notas.csv", "r") as f:
    reader = csv.reader(f, delimiter=',')
    next(reader)
    for row in reader:
        if (row[6] not in lista):
            lista.append(row[6])

# Generando archivo de códigos de alumnos
lista.sort()
fo = open("relacion.csv", "w")
fo.write("Codigo_Alumno"+ "\n")

for i in lista:
    fo.write(i + "\n")

fo.close()
```

---

2. Lectura y estandarización de cursos del Archivo de Notas

Procederemos a estandarizar la codificación de los cursos del Archivo de Notas y generaremos un nuevo archivo.

```
# Levantando el archivo de equivalencias
equivale = {}
with open("equi_cursos.csv", "r") as f:
    reader = csv.reader(f, delimiter=',')
    next(reader)
    for row in reader:
        # añado un nuevo key al diccionario
        equivale[row[0]] = row[1]

# Generando el archivo de notas con el codigo del curso estandarizado
fo = open("notas_1.csv", "w")
```

```
fo.write("semestre,cur_cod,eva_cod,eva_nota," +
         "alu_cod,eva_fin,grupo"+ "\n")

with open("Notas.csv", "r") as f:
    reader = csv.reader(f, delimiter=',')
    next(reader)
    for row in reader:
        cur_cod = equivale[row[1]]
        fo.write(row[0] + "," + cur_cod + "," + row[3].strip() + "," +
                 row[5] + "," + row[6] + "," + row[7] + "," + row[4] + "\n")

fo.close()
```

---

3. Dividir el Archivo de Notas en archivo por curso

Procederemos a dividir el Archivo de Notas en tantos archivos como cursos.
Adicionalmente generaremos un archivo auxiliar de los tipos de evaluacion en cada curso.
Tambien generaremos un archivo auxiliar conteniendo el histórico de los tipos de evaluacion en cada curso por semestre y por grupo.

```
for i in range(1, 14):
    file = str(i).zfill(4)
    lista = []
    donde = []

    # Generando archivo por cada curso
    fo = open(file + ".csv", "w")
    fo.write("semestre,eva_cod,eva_nota," +
             "alu_cod,eva_fin"+ "\n")
    with open("notas_1.csv", "r") as f:
        reader = csv.reader(f, delimiter=',')
        next(reader)
        for row in reader:
            if (str(i).zfill(4) == row[1]):
                tempo = row[0] + "," + row[6] + "," + row[2]
                if (tempo not in donde):
                    donde.append(tempo)
                if (row[2] not in lista):
                    lista.append(row[2])
                fo.write(row[0] + "," + row[2] + "," +
                         row[3] + "," + row[4] + "," + row[5] + "\n")
    fo.close()

    # Generando el archivo de tipos de evaluacion por curso
    lista.sort()
    fo = open(file + "_eva.csv", "w")
    fo.write("eva_cod"+ "\n")
    for j in lista:
        fo.write(j + "\n")
    fo.close()

    # Generando  archivo de tipos de evaluacion por curso, semestre y grupo
    donde.sort()
```

```
    fo = open(file + "_hist.csv", "w")
    fo.write("semestre,grupo,eva_cod"+ "\n")
    for j in donde:
        fo.write(j + "\n")
    fo.close()
```

---

4. Lectura y estandarización de evaluaciones en cada archivo por curso

Procederemos a revisar el Archivo de Notas y generaremos una relación única de tipos de evaluación.

```
for i in range(1, 14):
    file = str(i).zfill(4)
    notas = {}

    # Levantando el archivo de equivalencias
    equivale = {}
    with open(file + "_equi.csv", "r") as f:
        reader = csv.reader(f, delimiter=',')
        next(reader)
        for row in reader:
            # añado un nuevo key a los diccionarios
            equivale[row[0]] = row[1]
            notas[row[1]] = "NA"

    # Generando el archivo de notas con el tipo de evaluación estandarizado
    fo = open(file + "_1.csv", "w")
    fo.write("semestre,eva_new,eva_nota," +
             "alu_cod,eva_fin"+ "\n")
    with open(file + ".csv", "r") as f:
        reader = csv.reader(f, delimiter=',')
        next(reader)
        for row in reader:
            eva_cod = equivale[row[1]]
            fo.write(row[0] + "," + eva_cod + "," +
                     row[2] + "," + row[3] + "," + row[4] + "\n")
    fo.close()

    k = 1
    actual = None
    # Generando el archivo de alumnos por curso
    fo = open("temporal.csv", "w")
    fo.write("semestre,alu_cod" + "\n")
    with open(file + "_1.csv", "r") as f:
        reader = csv.reader(f, delimiter=',')
        next(reader)
        for row in reader:
            if (actual != row[3]):
                if (k == 0):
                    fo.write(cadena + "\n")
                else:
                    k = 0
                cadena = row[0] + "," + row[3]
                actual = row[3]
        fo.write(cadena + "\n")
```

```
fo.close()

# Obteniendo cuantas veces aparece el alumno
lista = {}
with open("temporal.csv", "r") as f:
    reader = csv.reader(f, delimiter=',')
    next(reader)
    for row in reader:
        if (row[1] in lista):
            lista[row[1]] = lista[row[1]] + 1
        else:
            lista[row[1]] = 1
fo.close()

cadena = ""
for j in notas:
    cadena = cadena + "," + j
k = 1
actual = None
# Generando el archivo por curso transpuesto completo
# Adicionamos una columna para el estado del rendimiento académico
fo = open(file + "_2.csv", "w")
fo.write("semestre,alu_cod,eva_fin,veces,estado" + cadena + "\n")
with open(file + "_1.csv", "r") as f:
    reader = csv.reader(f, delimiter=',')
    next(reader)
    for row in reader:
        if (actual != row[3]):
            if (k == 0):
                for j in notas:
                    cadena = cadena + "," + notas[j]
                fo.write(cadena + "\n")
                for j in notas:
                    notas[j] = "NA"
            else:
                k = 0
            if (row[4] == "99"): ## NSP
                estado = 'N'
            elif (int(row[4]) <= 10): ## Desaprobado
                estado = 'D'
            else: ## Aprobado
                estado = 'A'
            cadena = row[0] + "," + row[3] + "," + row[4] + "," +\
                     str(lista[row[3]]) + "," + estado
            actual = row[3]
        notas[row[1]] = row[2]
    for j in notas:
        cadena = cadena + "," + notas[j]
    fo.write(cadena + "\n")
fo.close()

# Obteniendo cabecera del archivo
with open(file + "_2.csv", "r") as f:
    reader = csv.DictReader(f, delimiter=',')
    headers = reader.fieldnames

# Obteniendo nueva cabecera
pos_est = headers.index("estado")
cadena = ""
```

```
    for j in headers:
        if (j != "estado"):
            cadena = cadena + j + ","
        else:
            cadena = cadena + "aprobo" + ","
    cadena = cadena[0:-1]

    # Generando el archivo por curso transpuesto sin NSP
    fo = open(file + "_3.csv", "w")
    fo.write(cadena + "\n")
    with open(file + "_2.csv", "r") as f:
        reader = csv.reader(f, delimiter=',')
        next(reader)
        for row in reader:
            if (row[pos_est] != "N"):
                cadena = ""
                for j in range(len(row)):
                    if (j != pos_est):
                        cadena = cadena + row[j] + ","
                    else:
                        if (row[pos_est] == "A"):
                            cadena = cadena + "Si" + ","
                        else:
                            cadena = cadena + "No" + ","
                cadena = cadena[0:-1]
                fo.write(cadena + "\n")
    fo.close()

from os import remove
remove("temporal.csv")
```

Anexo 06: Script del algoritmo en R para la comprensión de cada curso del Programa de Estudios Básicos (PEB).

Procederemos a revisar los Archivos de notas de cada curso del Programa de Estudios Básicos (PEB), con la finalidad de comprender su contenido.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")
```

Para la comprensión del Curso con código: "0002" en adelante, los resultados se visualizarán en el Anexo 11.

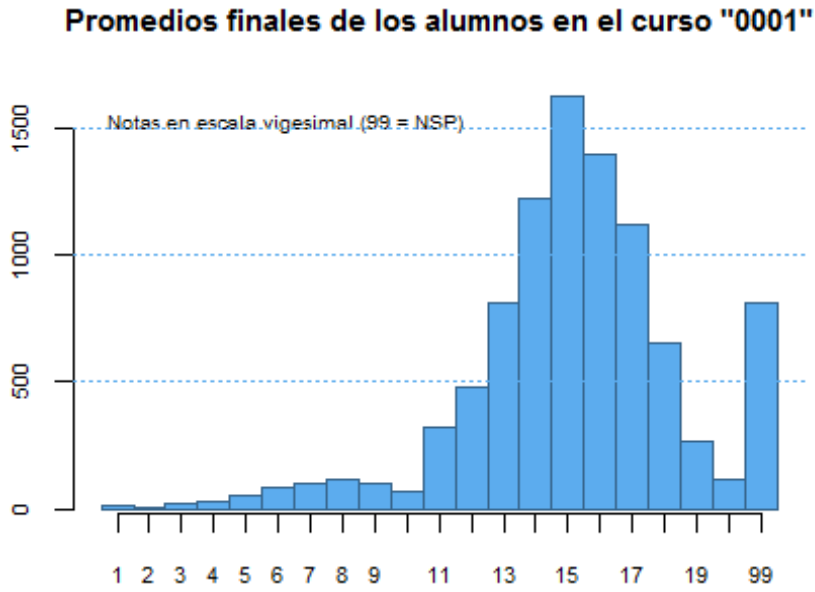
---

1. Archivo del Curso con código: "0001"

Visualización de la estructura interna del curso denominado: "Actividades Artísticas y Deportivas".

```
file = "0001"
## [1] "***** Inicio - Curso: 0001*****"

##############################
# 1.0 Lectura y Preparacion de los datos
##############################
## [1] "El curso original"

data <- read.csv(paste(file, ".csv", sep=""))
str(data)

## 'data.frame':    37371 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 8 levels "PRA1","PRA2",..: 5 6 7 8 5 6 7 8 5 6 ...
##  $ eva_nota: num  0 0 0 0 0 0 0 0 16 19 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201510876 201510
876 201510876 201510876 201512175 201512175 ...
##  $ eva_fin : int  99 99 99 99 99 99 99 99 17 17 ...

# Generando tabla de la columna 2 (Mascara del Curso)
print(table(data[, 2]))

## PRA1 PRA2 PRA3 PRA4 PTL1 PTL2 PTL3 PTL4
## 4957 4950 4941 4954 4406 4406 4395 4362

##############################
# Visualización de la estructura interna del curso con el tipo de evaluación
estandarizado.
##############################
## [1] "El curso con el tipo de evaluación estandarizado"

data <- read.csv(paste(file, "_1.csv", sep=""))
str(data)
```

```
## 'data.frame':    37371 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 4 levels "PRA1","PRA2",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ eva_nota: num  0 0 0 0 0 0 0 0 16 19 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201510876 201510
876 201510876 201510876 201512175 201512175 ...
##  $ eva_fin : int  99 99 99 99 99 99 99 99 17 17 ...

# Visualizando los primeros 10 registros
print(head(data, 10))

##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151    PRA1        0 201512432      99
## 2     20151    PRA2        0 201512432      99
## 3     20151    PRA3        0 201512432      99
## 4     20151    PRA4        0 201512432      99
## 5     20151    PRA1        0 201510876      99
## 6     20151    PRA2        0 201510876      99
## 7     20151    PRA3        0 201510876      99
## 8     20151    PRA4        0 201510876      99
## 9     20151    PRA1       16 201512175      17
## 10    20151    PRA2       19 201512175      17

##############################
# Visualización de la estructura interna del curso, con el tipo de evaluación
estandarizado y con la nueva estructura.
##############################
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"

cadena = paste(" de los alumnos en el curso ", paste('"', file, '"', sep=""),
sep="")
data  <- read.csv(paste(file, "_2.csv", sep=""))

# Convirtiendo las columnas necesarias a factores
data[, 1] <- as.factor(data[, 1])
data[, 2] <- as.character(data[, 2])
str(data)

## 'data.frame':    9365 obs. of  9 variables:
##  $ semestre: Factor w/ 11 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ alu_cod : chr  "201512432" "201510876" "201512175" "201512275" ...
##  $ eva_fin : int  99 99 17 15 16 19 16 12 99 15 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 2 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 3 3 1 1 1 1 1 1 3 1 ...
##  $ PRA1    : num  0 0 16 12 14 18 17 9 0 14 ...
##  $ PRA2    : num  0 0 19 15 16 19 17 14 0 15 ...
##  $ PRA4    : num  0 0 18 17 17 18 15 15 0 16 ...
##  $ PRA3    : num  0 0 16 16 16 19 15 10 0 14 ...

# Grafico de la frecuencia de los promedios finales de los alumnos
barplot(table(data[, 3]),
        cex.axis = 0.7, space = 0,  # para el rmd
        col = "steelblue2", border = "steelblue4", axis.lty = 1, cex.names =
0.7,
        main = list(paste("Promedios finales", cadena, sep=""), font = 2, cex
= 0.95))
legend("topleft", bty = "n", cex = 0.6,
       legend = "Notas en escala vigesimal (99 = NSP)")
abline(h = c(500, 1000, 1500, 2000), col = "steelblue2", lty = 3)
```




```
# Generar descripción en función a la columna 5 (estado)
describe <- paste(c("Aprobado","Desaprobado","NSP"), " ",
                  round(prop.table(table(data[, 5]))*100), "%")

# Distribución de los promedios finales de los alumnos
pie(table(data[, 5]),
    labels = describe,
    radius = 1,
    col = c("steelblue2", "lightsteelblue1", "lightskyblue"),
    main = list(paste("Rendimiento Académico", cadena, sep=""), font = 2, cex
= 0.95))
```


```
# Visualizando los primeros 5 registros
print(head(data, 5))
```

```
##   semestre   alu_cod eva_fin veces estado PRA1 PRA2 PRA4 PRA3
## 1    20151 201512432      99     2      N    0    0    0    0
## 2    20151 201510876      99     1      N    0    0    0    0
## 3    20151 201512175      17     1      A   16   19   18   16
## 4    20151 201512275      15     1      A   12   15   17   16
## 5    20151 201512356      16     1      A   14   16   17   16
```

```
##############################
# Visualización de la estructura interna del curso, con el tipo de evaluación
estandarizado, con la nueva estructura y solo con alumnos aprobados.
##############################
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"

data <- read.csv(paste(file, "_3.csv", sep=""))
str(data)

## 'data.frame':    8552 obs. of  9 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ alu_cod : int  201512175 201512275 201512356 201511917 201511830 201512
645 201511856 201512841 201512317 201510373 ...
##  $ eva_fin : int  17 15 16 19 16 12 15 17 12 14 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  16 12 14 18 17 9 14 15 11 15 ...
##  $ PRA2    : num  19 15 16 19 17 14 15 17 13 16 ...
##  $ PRA4    : num  18 17 17 18 15 15 16 19 14 16 ...
##  $ PRA3    : num  16 16 16 19 15 10 14 17 11 10 ...

print(head(data, 5))

##   semestre   alu_cod eva_fin veces aprobo PRA1 PRA2 PRA4 PRA3
## 1    20151 201512175      17     1     Si   16   19   18   16
## 2    20151 201512275      15     1     Si   12   15   17   16
## 3    20151 201512356      16     1     Si   14   16   17   16
## 4    20151 201511917      19     1     Si   18   19   18   19
## 5    20151 201511830      16     1     Si   17   17   15   15

## [1] "***** Fin de Ejecución *****"
```

Anexo 07: Script del algoritmo en R para la preparación de cada curso del Programa de Estudios Básicos (PEB).

Procederemos a revisar los Archivos de Datos para obtener una mejor visualización de su contenido.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
```

Para la preparación del Curso con código: "0002" en adelante, los resultados se visualizarán en el Anexo 12.

---

1. Archivo del Curso con código: "0001"

Preparación de la estructura interna del curso denominado: "Actividades Artísticas y Deportivas".

```
file = "0001"
## [1] "***** Inicio - Curso: 0001*****"

##############################
# 1.0 Levantando Archivos
##############################
base <- read.csv(paste(file, "_1.csv", sep=""))
base[, 1] <- as.factor(base[, 1])
base[, 4] <- as.character(base[, 4])
str(base)

## 'data.frame':    37371 obs. of  5 variables:
##  $ semestre: Factor w/ 11 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 4 levels "PRA1","PRA2",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ eva_nota: num  0 0 0 0 0 0 0 0 16 19 ...
##  $ alu_cod : chr  "201512432" "201512432" "201512432" "201512432" ...
##  $ eva_fin : int  99 99 99 99 99 99 99 99 17 17 ...

print(summary(base))

##     semestre    eva_new        eva_nota       alu_cod
##  20171  :6229   PRA1:9363   Min.   : 0.00   Length:37371
##  20161  :6192   PRA2:9356   1st Qu.:12.00   Class :character
##  20181  :6106   PRA3:9336   Median :15.00   Mode  :character
##  20162  :4500   PRA4:9316   Mean   :13.27
##  20172  :3964               3rd Qu.:17.00
##  20151  :3661               Max.   :20.00
##  (Other):6719
##     eva_fin
##  Min.   : 1.00
##  1st Qu.:14.00
##  Median :15.00
##  Mean   :21.99
##  3rd Qu.:17.00
##  Max.   :99.00
```

```
base_t <- read.csv(paste(file, "_2.csv", sep=""))
base_t[, 1] <- as.factor(base_t[, 1])
base_t[, 2] <- as.character(base_t[, 2])
str(base_t)
```

```
## 'data.frame':    9365 obs. of  9 variables:
##  $ semestre: Factor w/ 11 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512432" "201510876" "201512175" "201512275" ...
##  $ eva_fin : int  99 99 17 15 16 19 16 12 99 15 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 2 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 3 3 1 1 1 1 1 1 3 1 ...
##  $ PRA1    : num  0 0 16 12 14 18 17 9 0 14 ...
##  $ PRA2    : num  0 0 19 15 16 19 17 14 0 15 ...
##  $ PRA4    : num  0 0 18 17 17 18 15 15 0 16 ...
##  $ PRA3    : num  0 0 16 16 16 19 15 10 0 14 ...
```

```
print(summary(base_t))
```

```
##     semestre      alu_cod            eva_fin          veces
##  20171  :1561   Length:9365        Min.   : 1.00   Min.   :1.000
##  20161  :1548   Class :character   1st Qu.:14.00   1st Qu.:1.000
##  20181  :1528   Mode  :character   Median :15.00   Median :1.000
##  20162  :1131                      Mean   :21.99   Mean   :1.212
##  20172  : 993                      3rd Qu.:17.00   3rd Qu.:1.000
##  20151  : 924                      Max.   :99.00   Max.   :7.000
##  (Other):1680
##  estado        PRA1            PRA2            PRA4            PRA3
##  A:7984   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D: 568   1st Qu.:12.00   1st Qu.:13.00   1st Qu.:12.00   1st Qu.:12.00
##  N: 813   Median :15.00   Median :15.00   Median :15.00   Median :15.00
##           Mean   :13.14   Mean   :13.46   Mean   :13.29   Mean   :13.19
##           3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:17.00   3rd Qu.:17.00
##           Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##           NA's   :2       NA's   :9       NA's   :49      NA's   :29
```

```
data <- read.csv(paste(file, "_3.csv", sep=""))
data[, 1] <- as.factor(data[, 1])
data[, 2] <- as.character(data[, 2])
str(data)
```

```
## 'data.frame':    8552 obs. of  9 variables:
##  $ semestre: Factor w/ 11 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512175" "201512275" "201512356" "201511917" ...
##  $ eva_fin : int  17 15 16 19 16 12 15 17 12 14 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  16 12 14 18 17 9 14 15 11 15 ...
##  $ PRA2    : num  19 15 16 19 17 14 15 17 13 16 ...
##  $ PRA4    : num  18 17 17 18 15 15 16 19 14 16 ...
##  $ PRA3    : num  16 16 16 19 15 10 14 17 11 10 ...
```

```
print(summary(data))
```

```
##     semestre      alu_cod            eva_fin          veces
##  20171  :1445   Length:8552        Min.   : 1.00   Min.   :1.000
##  20161  :1419   Class :character   1st Qu.:13.00   1st Qu.:1.000
##  20181  :1408   Mode  :character   Median :15.00   Median :1.000
```

```
##  20162  :1047                      Mean   :14.67   Mean   :1.129
##  20172  : 896                      3rd Qu.:17.00   3rd Qu.:1.000
##  20151  : 858                      Max.   :20.00   Max.   :7.000
##  (Other):1479
##  aprobo         PRA1            PRA2            PRA4            PRA3
##  No: 568   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  Si:7984   1st Qu.:13.00   1st Qu.:14.00   1st Qu.:14.00   1st Qu.:13.00
##            Median :15.00   Median :15.00   Median :16.00   Median :15.00
##            Mean   :14.39   Mean   :14.74   Mean   :14.56   Mean   :14.44
##            3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:17.00   3rd Qu.:17.00
##            Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##            NA's   :2       NA's   :8       NA's   :46      NA's   :24
```

```
##############################
# 2.0 Verificacion de cantidad de Registros
##############################
## [1] "Verificacion de cantidad de Registro..."
```

```
# Registros en el archivo "base" (original de notas), que es el equivalente a la cantidad de notas
dim(base)[1]
```

```
## [1] 37371
```

```
# Verificando la cantidad de registros en el archivo "base_t" (transpuesto de notas)
p1 = sapply(base_t, function(x) sum(!is.na(x)))
p2 = sapply(base_t, function(x) sum(is.na(x)))
dim(base_t)[1]
```

```
## [1] 9365
```

```
dim(base_t)[1] == (p1 + p2)
```

```
## semestre  alu_cod  eva_fin    veces   estado     PRA1     PRA2     PRA4
##     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
##     PRA3
##     TRUE
```

```
# La Cantidad de notas del archivo original coincide con la cantidad de notas en las diferentes evaluaciones de "base_t" ??
p3 = sapply(base_t[6:length(base_t)], function(x) sum(!is.na(x))) # Registros no vacios (con notas)
dim(base)[1] == sum(p3)
```

```
## [1] TRUE
```

```
#----------------------------
# 2.1 Verificacion en el "data"
#----------------------------
# Verificando la cantidad de registros en el archivo de notas "data" (transpuesto y sin NSP)
d1 = sapply(data, function(x) sum(!is.na(x)))
d2 = sapply(data, function(x) sum(is.na(x)))
dim(data)[1]
```

```
## [1] 8552
```

```
dim(data)[1] == (d1 + d2)
```

```
## semestre  alu_cod  eva_fin    veces   aprobo     PRA1     PRA2     PRA4
##     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
##     PRA3
##     TRUE
```

```
# La Cantidad de registros coincide con cantidad de registros del archivo "ba
se_t" ??
dim(data)[1] == sum(base_t$estado == "A" | base_t$estado == "D")
```

```
## [1] TRUE
```

```
# Para los aprobados
# La Cantidad de registros no vacios (con notas) coincide con la cantidad de
notas no vacias del archivo "base_t" ??
p4 = sapply(base_t[base_t$estado == "A", 6:length(base_t)], function(x) sum(!
is.na(x)))
d3 = sapply(data[data$aprobo == "Si", 6:length(data)], function(x) sum(!is.na
(x)))
d3 == p4
```

```
## PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE
```

```
# La Cantidad de registros vacios (sin notas) coincide con la cantidad de not
as vacias del archivo "base_t" ??
p5 = sapply(base_t[base_t$estado == "A", 6:length(base_t)], function(x) sum(i
s.na(x)))
d4 = sapply(data[data$aprobo == "Si", 6:length(data)], function(x) sum(is.na(
x)))
d4 == p5
```

```
## PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE
```

```
# Para los desaprobados
# La Cantidad de registros no vacios (con notas) coincide con la cantidad de
notas no vacias del archivo "base_t" ??
p6 = sapply(base_t[base_t$estado == "D", 6:length(base_t)], function(x) sum(!
is.na(x)))
d5 = sapply(data[data$aprobo == "No", 6:length(data)], function(x) sum(!is.na
(x)))
d5 == p6
```

```
## PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE
```

```
# La Cantidad de registros vacios (sin notas) coincide con la cantidad de not
as vacias del archivo "base_t" ??
p7 = sapply(base_t[base_t$estado == "D", 6:length(base_t)], function(x) sum(i
s.na(x)))
d6 = sapply(data[data$aprobo == "No", 6:length(data)], function(x) sum(is.na(
x)))
d6 == p7
```

```
## PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE
```

```
# La Cantidad de registros coincide con la cantidad de notas de los aprobados
del archivo "base" ??
print((d3 + d4 + d5 + d6) == (p4 + p5 + p6 + p7))
```

```
## PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE

##############################
# 3.0 Verificacion si existe examen sustitutorio
##############################
## [1] "Verificando si existe examen sustitutorio..."

# Existe el examen sustitutorio en el archivo de notas "data" ???
col_sus = "SUS" == colnames(data)
print(col_sus)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

#---------------------------
# 3.1 Cambio de la nota del Parcial/Final por la nota del Sustitutorio
#---------------------------
# Donde estan ubicadas las columnas del Parcial (PAR), Final (FIN) y Sustitut
orio (SUS)
pos_par = which("PAR" == colnames(data))
pos_fin = which("FIN" == colnames(data))
pos_sus = which(col_sus == TRUE)

# Generamos vectores con nuevos contenidos
vec_sus = ifelse(data[,col_sus] <= 20 & !is.na(data[,col_sus]), 1, 0)
vec_par = ifelse((data[,col_sus] <= 20 & data[,pos_par] <= data[,pos_fin] & !
is.na(data[,col_sus])),
                 data[,col_sus], data[,pos_par])
vec_fin = ifelse((data[,col_sus] <= 20 & data[,pos_par] > data[,pos_fin] & !i
s.na(data[,col_sus])),
                 data[,col_sus], data[,pos_fin])

# Reemplazamos en "data" los nuevos contenidos
data[pos_sus] = vec_sus
data[, pos_sus] <- as.factor(data[, pos_sus])
data[pos_par] = vec_par
data[pos_fin] = vec_fin


##############################
# 4.0 Verificacion de valores perdidos (NA)
##############################
## [1] "Verificacion de valores perdidos (NA)..."

# Filas con valores perdidos
miss_row = which(rowSums(is.na(data))!=0, arr.ind=T)
length(miss_row) # Para ver cuantas filas tienen valores perdidos

## [1] 65

# Porcentaje de filas con valores perdidos
length(miss_row) * 100 / dim(data)[1]

## [1] 0.7600561

# Columnas con valores perdidos
miss_col = which(colSums(is.na(data)) !=0)
str(miss_col) # Para ver las columnas con valores perdidos
```

```
##  Named int [1:4] 6 7 8 9
##  - attr(*, "names")= chr [1:4] "PRA1" "PRA2" "PRA4" "PRA3"
```

```
# Porcentaje de columnas con valores perdidos
if (length(miss_col) == 1) {
   (miss_col_por = 100 * sum(is.na(data[,miss_col])) / dim(data)[1])
}else{
   (miss_col_por = 100 * colSums(is.na(data[,miss_col])) / dim(data)[1])
}
```

```
##       PRA1       PRA2       PRA4       PRA3
## 0.02338634 0.09354537 0.53788587 0.28063611
```

```
# Hay alguna columna que supera el 30% de datos perdidos ???
menos_col = miss_col[which(miss_col_por >= 30)]
print(menos_col)
```

```
## named integer(0)
```

```
# En el resultado se observa que NO/SI hay columna con porcentaje mayor o igu
al a 30
if (length(menos_col) == 0) {
  # Entonces, el nuevo archivo de notas es "data"
  new_file = data
} else {
  # Entonces, se procede a eliminar las columnas con porcentaje mayor o igual
a 30
  new_file = data[,-menos_col]
}
```

```
################################
# 5.0 Grabación del archivo final del curso
################################
write.csv(new_file,
          file = paste(file, "_4.csv", sep=""),
          row.names = FALSE, quote = FALSE)

print(paste("*****", " Fin de Ejecución ", "*****", sep=""))
```

```
## [1] "***** Fin de Ejecución *****"
```

Anexo 08: Script del algoritmo en R para la creación del dataset de cada curso del Programa de Estudios Básicos (PEB).

Procederemos a generar los Dataset a partir de los Archivos de notas de cada curso del Programa de Estudios Básicos (PEB).

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")
paleta <- colorRampPalette(c("dodgerblue", "white", "dodgerblue4"))

# Uso de Librerías
library(dplyr)
library(VIM) # imputacion
library(DMwR) # Densidad Local
library(caret) # nzv
library(corrplot)
```

Para la creación del dataset del Curso con código: "0002" en adelante, los resultados se visualizarán en el Anexo 13.

---

1. Creación del DataSet número 1.

Generación de la estructura interna del primer DataSet.
Formado por:
1. *"Actividades Artísticas y Deportivas", con código: "0001"*

```
# Levantando el archivo de alumnos una única vez
data_alu <- read.csv("alumnos_b.csv",
                     colClasses = c("character", "factor", "factor",
                                    "Date", "factor", "factor", "factor"))

file = "0001"
## [1] "***** Inicio - Curso: 0001*****"

##############################
# 1.0 Levantando el archivo del curso
##############################
base_01 <- read.csv(paste(file, "_4.csv", sep=""))
base_01[, 1] <- as.factor(base_01[, 1])
base_01[, 2] <- as.character(base_01[, 2])

# Eliminando columnas extras
base_01 <- base_01[, c(1:5, which(colnames(base_01) == "PRA1"))]
base_01 <- base_01[, -which(colnames(base_01) == "eva_fin")]

# Cambiando nombre a columnas
tmp_col_names <- colnames(base_01)
for(h in tmp_col_names)
{
   tmp_col_names[which(tmp_col_names == h)] <- paste(h, "_1", sep="")
```

```r
}
tmp_col_names[which(tmp_col_names == "alu_cod_1")] <- "alu_cod"
tmp_col_names[which(tmp_col_names == "aprobo_1")]  <- "aprobo"
colnames(base_01) <- tmp_col_names

# Generando el DataSet
D_01 <- inner_join(data_alu, base_01, by = "alu_cod")
D_01 <- D_01[, -(which(names(D_01) == "alu_cod"))]
str(D_01)
```

```
## 'data.frame':    8539 obs. of  10 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 11 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1    : int  14 16 18 16 13 17 14 13 12 14 ...
```

```r
##############################
# 2. Pre-Procesamiento de los datos
##############################
```

```
## [1] "Verificacion de valores perdidos (NA)..."
```

```r
#---------------------------
# 2.1 Verificacion de valores perdidos (NA)
#---------------------------
# Filas con valores perdidos
miss_row = which(rowSums(is.na(D_01)) != 0, arr.ind = T)
length(miss_row) # Para ver cuantas filas tienen valores perdidos
```

```
## [1] 1
```

```r
# Porcentaje de filas con valores perdidos
length(miss_row) * 100 / dim(D_01)[1]
```

```
## [1] 0.01171097
```

```r
# Columnas con valores perdidos
miss_col = which(colSums(is.na(D_01)) != 0)
str(miss_col) # Para ver las columnas con valores perdidos
```

```
##  Named int 10
##  - attr(*, "names")= chr "PRA1_1"
```

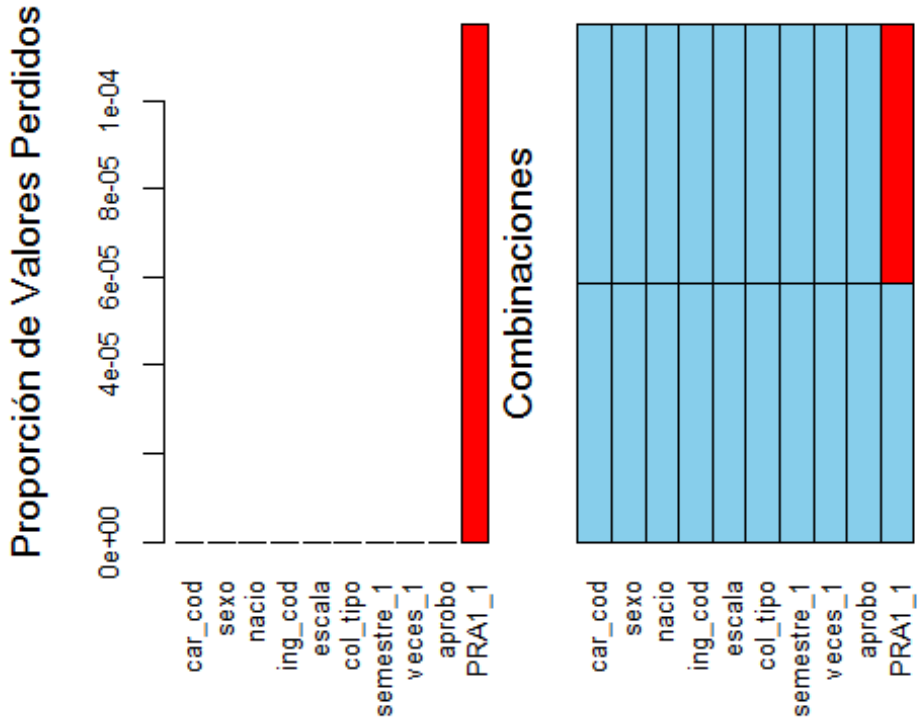
```r
# Porcentaje de columnas con valores perdidos
if (length(miss_col) == 1) {
   (miss_col_por = 100 * sum(is.na(D_01[,miss_col])) / dim(D_01)[1])
} else {
   (miss_col_por = 100 * colSums(is.na(D_01[,miss_col])) / dim(D_01)[1])
}
```

```
## [1] 0.01171097
```

```r
# Hay alguna columna que supera el 30% de datos perdidos ???
h = miss_col[which(miss_col_por >= 30)]
print(h)
```

```
## named integer(0)
```

```r
# En el resultado se observa que NO/SI hay columna con porcentaje mayor o igu
al a 30
if (length(h) == 0) {
   # Entonces, el nuevo archivo de notas es "data"
   D_01 = D_01
} else {
   # Entonces, se procede a eliminar las columnas con porcentaje mayor o igua
l a 30
   D_01 = D_01[,-h]
   miss_col = which(colSums(is.na(D_01)) !=0)
}

#----------------------------
# 2.1.1 Imputacion de valores perdidos (NA)
#----------------------------
# Donde se encuentran los datos faltantes
summary(aggr(D_01,
             bars = F,
             numbers = T,
             ylabs = c("Proporción de Valores Perdidos", "Combinaciones"),
             cex.axis = 0.8, cex.numbers = 0.8,
             gap = 2))
```




```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
```

```
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##     veces_1     0
##      aprobo     0
##      PRA1_1     1
##
##  Missings in combinations of variables:
##         Combinations Count     Percent
##  0:0:0:0:0:0:0:0:0:0  8538 99.98828903
##  0:0:0:0:0:0:0:0:0:1     1  0.01171097
```

```
# Preparar DataSet para imputar los datos faltantes
D_02       <- D_01

# Proceso para imputar los datos faltantes
D_02 <- VIM::kNN(D_02, k = 5)
D_02 <- D_02[, 0:dim(D_01)[2]]

# Comprobación de eliminación de datos faltantes
print(head(D_01[c(miss_row), miss_col], 10))
```

```
## [1] NA
```

```
print(head(D_02[c(miss_row), miss_col], 10))
```

```
## [1] 14
```

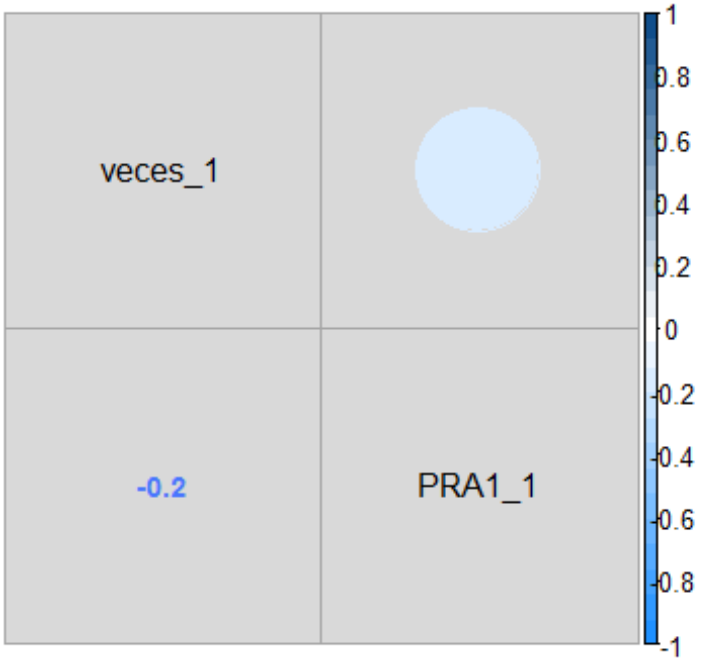
```
#----------------------------
# 2.2 Revisar data atipicos (Outliers)
#     Densidad Local
#----------------------------
## [1] "Verificando Outliers..."

# Obteniendo columnas con valores numericos
tmp_col_names <- sapply(D_02, is.numeric)

data_tmp = D_02[D_02[which(colnames(D_02) == "aprobo")]=="Si", tmp_col_names]
h = as.numeric(rownames(data_tmp))
indicador = lofactor(data_tmp, k = 10)

# Porcentaje de elementos outliers
100 * length(indicador) / dim(D_02)[1]
```

```
## [1] 93.34817
```

```
# Se decidio quedarse con los outliers y trabajar la data completa

#----------------------------
# 2.3 Identification of nzv (near zero variance) predictors
#----------------------------
## [1] "Verificando nzv..."

indicador <- nearZeroVar(D_02, saveMetrics = TRUE)
#  Se ha detectado datos con variancia cero o casi cero ??
FALSE == indicador$nzv
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```r
# No / SI Porque, Segun los resultados Todas las variables tienen valor False
/True

if (length(which(TRUE == indicador$nzv)) == 0) {
   # Entonces, el nuevo archivo de notas es "data"
   D_02 <- D_02
} else {
   # si hay poca variabilidad para que la mantienes, hay que eliminarla
   indicador <- nearZeroVar(D_02)
   data_tmp <- D_02[, -indicador]
   dim(data_tmp)
}

#----------------------------
# 2.4 La matriz de correlación
#     Relaciones entre las variables numéricas
#----------------------------
## [1] "Verificando correlacion..."

# Indices de correlacion
data_tmp <- D_02[,tmp_col_names]
indicador <- cor(data_tmp)
print(indicador)

##           veces_1    PRA1_1
## veces_1  1.000000 -0.197051
## PRA1_1  -0.197051  1.000000

# Grafico
corrplot(indicador, type = "upper", bg = "gray85",
         col = paleta(25),
         tl.pos = "d", addgrid.col = "gray65",
         method ="circle", cl.length = 11, tl.cex = 1, tl.col = "black", )
corrplot(indicador, type = "lower", bg = "gray85",
         col = "royalblue1",
         tl.pos = "n", addgrid.col = "gray65",
         method = "number", diag = FALSE, cl.pos = "n", add = TRUE, number.ce
x = 0.9)
```




```
# Detecta la alta correlacion en la matriz triangular superior
summary(indicador[upper.tri(indicador)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.1971 -0.1971 -0.1971 -0.1971 -0.1971 -0.1971
```

```
(h <- sum(abs(indicador[upper.tri(indicador)]) > .999))
```

```
## [1] 0
```

```
colnames(data_tmp)[h]
```

```
## character(0)
```

```
# detectar aquellas con correlacion > 0.75 (columnas altamente correlacionada
s)
h <- findCorrelation(indicador, cutoff = .75)
if (length(h) > 0) {
   # extraigo las columnas con correlacion > 0.75
   colnames(data_tmp)[h]

   # eliminando
   data_tmp <- D_02[,-(which(colnames(D_02) == colnames(data_tmp)[h]))]
}
```

```
#----------------------------
# 2.5 Calcular diferencia de periodos entre fechas
#----------------------------
## [1] "Generando dataset..."
```

```
# Para el alumno
pos_nac = which("nacio" == colnames(D_02))
# Para el primer curso
pos_sem = which("semestre_1" == colnames(D_02))

# Generamos vector con la diferencia en años entre la fecha de nacimiento y e
l semestre donde aprobo el primer curso
sem_dif_1 = as.integer(substr(D_02[, pos_sem], 1, 4)) - year(D_02[, pos_nac])
```

```r
sem_dif_1 = sem_dif_1 * 3 + as.integer(substr(D_02[, pos_sem], 5, 5))
sem_dif_1 = sem_dif_1 - min(sem_dif_1) + 1

# Agregar vector de diferencia y Eliminar fecha de nacimiento y semestre
D_02 = cbind(D_02[,-c(pos_nac, pos_sem)], sem_dif_1)
str(D_02)
```

```
## 'data.frame':    8539 obs. of  9 variables:
##  $ car_cod  : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1 .
..
##  $ sexo     : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ ing_cod  : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6 .
..
##  $ escala   : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3 3
...
##  $ col_tipo : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1   : int  14 16 18 16 13 17 14 13 12 14 ...
##  $ sem_dif_1: num  12 12 15 12 12 15 12 12 12 12 ...
```

```r
##############################
# 3.0 Grabación del DataSet final del curso
##############################
write.csv(D_02,
          file = paste("DS_", file, ".csv", sep=""),
          row.names = FALSE, quote = FALSE)
```

```
## [1] "***** Fin de Ejecución *****"
```

Anexo 09: Script del algoritmo en R para el tuning de los modelos de prueba de cada técnica de modelado para cada curso del Programa de Estudios Básicos (PEB).

Generacion del modelo para cada uno de los Archivos de Datos.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")

# Uso de Librerías
library(caret)
library(caretEnsemble)
library(dummies)
library(Boruta) # Selección de variables
library(DMwR) # SMOTE
library(devtools) # Para visualizar la RNA
library(dplyr) # Para resumenes
source_url('https://gist.githubusercontent.com/fawda123/7471137/raw/466c1474d
0a505ff044412703516c34f1a4684a5/nnet_plot_update.r')
```

Para la generacion del modelo para el Curso con código: "0002" en adelante, los resultados se visualizarán en el Anexo 14.

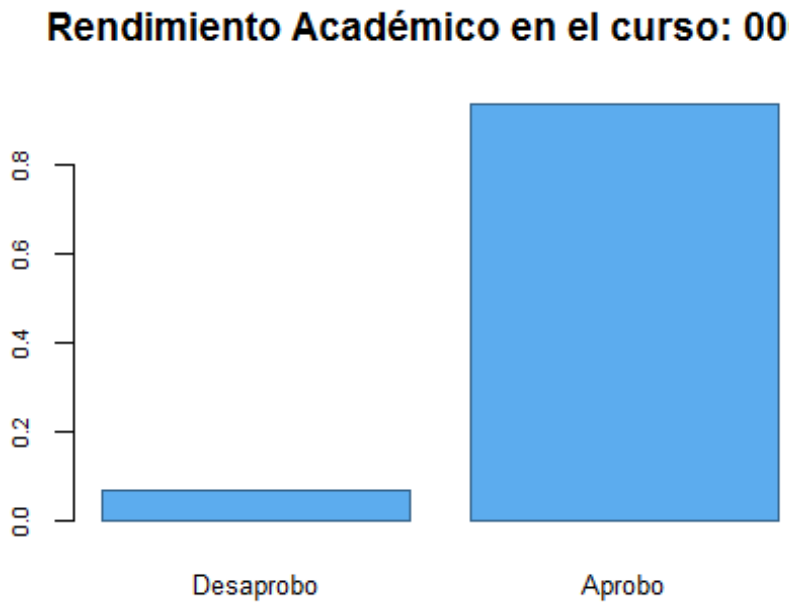
---

1. Archivo del Curso con código: "0001"

Generacion del modelo para el curso denominado: "Actividades Artísticas y Deportivas".

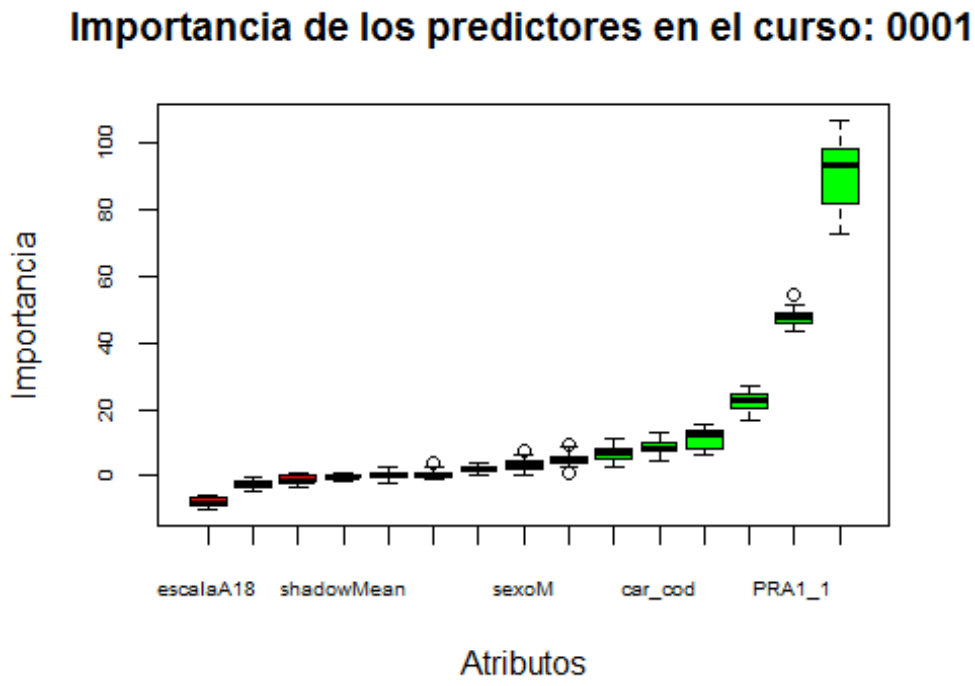
```
##############################
# 0.0 Definiendo variables generales del modelo
##############################
# Variables relacionadas con el Dataset
val_target  <- "aprobo"

# Variables relacionadas con los métodos
models_met <- c("nnet", "gbm", "xgbTree")
models_nom <- c("Red Neuronal Artificial (RNA)",
                "Gradient Boosting Machine (GBM)",
                "XGBoosting")

# Definiendo nombre(s) para cada método
val_nom_metodo <- c("metodo_rna", "metodo_gbm", "metodo_xgb")

# Definiendo tuneGrid(s) para cada método
val_tunegrid_A <- c("rna_grid", "gbm_grid", "xgb_grid")
val_tunegrid_B <- c("grid_rna", "grid_gbm", "grid_xgb")
# Definiendo el tuneLength para cada método
val_tunelen <- c(5, 2, 2)
```

```r
# Definiendo la estandarizacion para datos numericos
val_preproc  <- "range"
# Definiendo la metrica de calculo
val_metrica = "Accuracy"

k=1
val_formula <- paste(val_target, "~ .") # "aprobo ~ ."
file = formatC(k, width = 4, flag = "0")
## [1] "***** Inicio - Curso: 0001*****"

##############################
# 1.0 Lectura y Preparacion de los datos
##############################
data <- read.csv(paste("DS_", file, ".csv", sep=""))

# Convirtiendo a Variables Dummies
aux = colnames(data)
data <- dummy.data.frame(data, names = c("sexo", "col_tipo"))
data <- data[, -(which(aux == "sexo"))]
data <- data[, -(which(aux == "col_tipo"))]

data <- dummy.data.frame(data, names = c("escala"))
data <- data[, -(which(aux == "escala"))]
str(data)

## 'data.frame':    8539 obs. of  13 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1   : int  14 16 18 16 13 17 14 13 12 14 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...

# Selección de muestra de entrenamiento (70%) y de Validación (30%)
set.seed(123)
aux  <- createDataPartition(data[, val_target], p = 0.7, list = FALSE)
data_train <- data[ aux, ] # Datos training
data_test  <- data[-aux, ] # Datos testing

#---------------------------
# 1.1 Balanceo de los datos
#---------------------------
# Gráfico donde muestra las diferencias entre las proporciones
barplot(prop.table(table(data_train[, val_target])),
        cex.axis = 0.7,
        col = "steelblue2", border = "steelblue4", cex.names = 0.8,
        names.arg = c("Desaprobo", "Aprobo"),
        main = list(paste("Rendimiento Académico en el curso: ", file, sep=""
), font = 2))
```




```
# Para obtener la tabla de frecuencia de la Variable Dependiente
table(data_train[, val_target])

##   No   Si
##  398 5580

h = prop.table(table(data_train[, val_target])) * 100
print(h)

##        No        Si
##  6.657745 93.342255

#----------------------------
# 1.1.1 Proceso del balanceo de los datos
#----------------------------
# Si los resultados estan en una proporcion mayor/menor de 9 a 1, entonces:
#  se necesita realizar un balanceo de Datos
if (h[1] <= 10) {
   h = round(20 / (prop.table(table(data_train[, val_target]))[1] * 100)) * 1
00
   set.seed(789)
   aux <- SMOTE(as.formula(val_formula), data = data_train,
                perc.over = h, # porcentaje de oversampling
                perc.under = 100) # porcentaje de undersampling
   table(aux[, val_target])
   data_train <- aux
}

#----------------------------
# 1.2 Seleccion de variables
#----------------------------
# Se selecciona las variables basandose en la data original,
#  antes de dividirlas en training y testing
set.seed(111)
aux <- Boruta(as.formula(val_formula),
              data = data, doTrace = 0)
plot(aux, cex.axis = 0.6,
     cex.lab = 1, xlab = "Atributos", ylab = "Importancia",
```

```
     main = list(paste("Importancia de los predictores en el curso: ", file,
sep=""), font = 2))
```




```
h = names(aux$finalDecision[aux$finalDecision == "Confirmed"])
val_formula <- paste(paste(val_target, "~"),
                     paste(h[!h %in% val_target], collapse = " + "))


##############################
# 2.0 Generando Modelos
##############################
#---------------------------
# 2.1 Generando dataframe para ejecucion de los metodos
#---------------------------
models_nro <- length(models_met)
# Generando el dataframe para ejecutar de los metodos
modelos_resu <- matrix(c(models_nom,
                         models_met,
                         val_tunelen),
                       nrow = models_nro,
                       ncol = 3,
                       byrow = FALSE,
                       dimnames = list(c(seq(1, models_nro)),
                                       c("Nombre", "Metodo", "TuneLength")))
aux          <- matrix(c(val_nom_metodo,
                         val_tunegrid_A,
                         val_tunegrid_B),
                       nrow = models_nro,
                       ncol = 3,
                       byrow = FALSE,
                       dimnames = list(NULL,
                                       c("Nom_Metodo", "Tunegrid_A", "Tunegri
d_B")))
modelos_resu <- cbind(modelos_resu,
                      as.data.frame(aux, stringsAsFactors = F),
                      stringsAsFactors = F)
```

```r
#----------------------------
# 2.1.1 Agregando filas para ensamble y stacking
#----------------------------
# Modificando el dataframe para almacenar los parametros de los metodos
modelos_resu = rbind(modelos_resu, modelos_resu[models_nro,])
modelos_resu[models_nro+1, c(1:6)] <- c("Ensamble", "Ensamble", models_nro, "
metodo_ens", "", "")
modelos_resu = rbind(modelos_resu, modelos_resu[models_nro,])
modelos_resu[models_nro+2, c(1:6)] <- c("Stacking", "Stacking", "", "metodo_s
tk", "stk_grid", "grid_stk")

# Renombrando filas
rownames(modelos_resu) <- NULL

#----------------------------
# 2.2 Generando el dataframe para almacenar los indicadores de las pruebas
#----------------------------
# Para las pruebas se necesita el accuracy
methods_nro = models_nro + 1
aux          <- matrix(rep(0, methods_nro*2*4),
                       nrow = methods_nro*2,
                       ncol = 4,
                       dimnames = list(c(paste0(models_met[seq(1, models_nro)
], "_train"),
                                         "stk_train",
                                         paste0(models_met[seq(1, models_nro)
], "_test"),
                                         "stk_test"),
                                       c(paste0("Prueba ", seq(1, 4)))))
modelos_accu <- as.data.frame(t(aux))

#----------------------------
# 2.2 Generando Entrenamiento
#----------------------------
# El entrenamiento es de 30 muestras (3*10)
set.seed(456)
val_ctrl <- trainControl(method  = "repeatedcv",
                         repeats = 3,
                         savePredictions = 'final',
                         classProbs = TRUE,
                         number  = 10)

#----------------------------
# 2.3 Generando los Modelos
#----------------------------
#----------------------------
# 2.3.1 Creación de los Modelos de RNA
#----------------------------
#----------------------------
# 2.3.1.1 Generando los Modelos
#----------------------------
# Definiendo el tuneGrid para cada método
rna_grid <- expand.grid(size = seq(from = 3, to = 10, by = 1),
                        decay = seq(from = 0.1, to = 0.5, by = 0.1))
grid_rna <- expand.grid(size = seq(1,3,1),
                        decay = seq(0, 0.05, 0.005))

i = 1
```

```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0001
"

# Prueba sin hiperparametros
set.seed(789)
metod1 <- train(as.formula(val_formula),
                data       = data_train,
                method     = modelos_resu[i, 2],
                trace      = F,
                lineout    = F)

# Prueba con hiperparametros no definidos
set.seed(789)
metod2 <- train(as.formula(val_formula),
                data       = data_train,
                method     = modelos_resu[i, 2],
                preProcess = val_preproc,
                trControl  = val_ctrl,
                tuneLength = as.integer(modelos_resu[i, 3]),
                metric     = val_metrica,
                trace      = F,
                lineout    = F)

# Pruebas con hiperparametros definidos
for(h in 1:(dim(modelos_resu)[2]-4))
{
   set.seed(789)
   metodo <- train(as.formula(val_formula),
                   data       = data_train,
                   method     = modelos_resu[i, 2],
                   preProcess = val_preproc,
                   trControl  = val_ctrl,
                   tuneGrid   = eval(parse(text = modelos_resu[i, h+4])),
                   metric     = val_metrica,
                   trace      = F,
                   lineout    = F)
   assign(paste("metod", h+2, sep=""), metodo)
}

for(j in 1:4)
{
   if (k==1) assign(paste("rna", j, sep=""), get(paste("metod", j, sep="")))
   assign("aux", get(paste("metod", j, sep="")))
   # Evaluación de la perfomance del modelo en Train
   modelos_accu[j, i] <- max(aux$results$Accuracy)

   # Evaluación de la perfomance del modelo en Test
   metodo_clas       <- predict(aux, newdata = data_test)
   # Calcular el % de acierto (accuracy) o tasa de exactitud
   modelos_accu[j, i + methods_nro] <- mean(data_test[, val_target] == metodo
_clas)
}

#----------------------------
# 2.3.1.2 Comparación de los modelos
#----------------------------
aux = which(modelos_accu[,i + methods_nro] == max(modelos_accu[,i + methods_n
ro]))
```

```r
#----------------------------
# 2.3.1.3 Generación del modelo Final
#----------------------------
assign(modelos_resu[i,4], get(paste("metod", aux, sep=""))) #metodo_rna <- me
tod2
plot.nnet(get(modelos_resu[i,4])$finalModel,
          nid=F, # standard illustration
          main=paste(models_nom[i], "del Curso:", file))

#----------------------------
# 2.3.2 Creación de los Modelos de GBM
#----------------------------
#----------------------------
# 2.3.2.1 Generando los Modelos
#----------------------------
# Definiendo el tuneGrid para cada método
gbm_grid <- expand.grid(n.trees           = seq(50,51,1), # numero de arboles
                        interaction.depth = c(2, 3, 4),   # numero de niveles
de profundidad de cada iteracion
                        shrinkage         = 0.1,          # Tasa de aprendiza
je
                        n.minobsinnode    = 10)           # numero minimo de
observaciones en el nodo hijo
grid_gbm <- expand.grid(n.trees           = seq(140,160,1),
                        interaction.depth = seq(5,8,1),
                        shrinkage         = 0.05,
                        n.minobsinnode    = 10)

i = 2
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
01"

# Prueba sin hiperparametros
set.seed(789)
metod1 <- train(as.formula(val_formula),
                data       = data_train,
                method     = modelos_resu[i, 2],
                verbose    = F,
                distribution = "bernoulli")

# Prueba con hiperparametros no definidos
set.seed(789)
metod2 <- train(as.formula(val_formula),
                data       = data_train,
                method     = modelos_resu[i, 2],
                preProcess = val_preproc,
                trControl  = val_ctrl,
                tuneLength = as.integer(modelos_resu[i, 3]),
                verbose    = F,
                distribution = "bernoulli")

# Pruebas con hiperparametros definidos
for(h in 1:(dim(modelos_resu)[2]-4))
{
   set.seed(789)
   metodo <- train(as.formula(val_formula),
                   data       = data_train,
                   method     = modelos_resu[i, 2],
                   preProcess = val_preproc,
```

```
                    trControl  = val_ctrl,
                    tuneGrid   = eval(parse(text = modelos_resu[i, h+4])),
                    metric     = val_metrica,
                    verbose    = F,
                    distribution = "bernoulli")
   assign(paste("metod", h+2, sep=""), metodo)
}

for(j in 1:4)
{
   if (k==1) assign(paste("gbm", j, sep=""), get(paste("metod", j, sep="")))
   assign("aux", get(paste("metod", j, sep="")))
   # Evaluación de la perfomance del modelo en Train
   modelos_accu[j, i] <- max(aux$results$Accuracy)

   # Evaluación de la perfomance del modelo en Test
   metodo_clas       <- predict(aux, newdata = data_test)
   # Calcular el % de acierto (accuracy) o tasa de exactitud
   modelos_accu[j, i + methods_nro] <- mean(data_test[, val_target] == metodo
_clas)
}

#----------------------------
# 2.3.2.2 Comparación de los modelos
#----------------------------
aux = which(modelos_accu[,i + methods_nro] == max(modelos_accu[,i + methods_n
ro]))

#----------------------------
# 2.3.2.3 Generación del modelo Final
#----------------------------
assign(modelos_resu[i,4], get(paste("metod", aux, sep="")))


#----------------------------
# 2.3.3 Creación de los Modelos de XG
#----------------------------
#----------------------------
# 2.3.3.1 Generando los Modelos
#----------------------------
# Definiendo el tuneGrid para cada método
xgb_grid <- expand.grid(nrounds          = seq(48,51,1), # numero de arboles
                        max_depth        = c(1, 2, 3),   # numero de niveles
de profundidad de cada iteracion
                        eta              = c(0.3),       # Tasa de aprendizaj
e de 0 a 1 (0.01 a 0.3)
                        gamma            = c(0),         # Tasa de regulariza
cion (penalizacion)
                        colsample_bytree = c(0.8),       # Columnas en cada a
rbol
                        min_child_weight = c(1),         # peso de los hijos
                        subsample        = c(1))         # Porcentaje de la m
uestra (default 50%)
grid_xgb <- expand.grid(nrounds          = seq(52,55,1),
                        max_depth        = seq(4,8,1),
                        eta              = 0.2,
                        gamma            = c(0),
                        colsample_bytree = c(0.8),
                        min_child_weight = c(1),
```

```
                      subsample        = c(1))

i = 3
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0001"

# Prueba sin hiperparametros
set.seed(789)
metod1 <- train(as.formula(val_formula),
                data       = data_train,
                method     = modelos_resu[i, 2],
                verbose    = 0,
                objective  = "binary:logistic",
                eval_metric = "error")

# Prueba con hiperparametros no definidos
set.seed(789)
metod2 <- train(as.formula(val_formula),
                data       = data_train,
                method     = modelos_resu[i, 2],
                preProcess = val_preproc,
                trControl  = val_ctrl,
                tuneLength = as.integer(modelos_resu[i, 3]),
                verbose    = 0,
                objective  = "binary:logistic",
                eval_metric = "error")

# Pruebas con hiperparametros definidos
for(h in 1:(dim(modelos_resu)[2]-4))
{
   set.seed(789)
   metodo <- train(as.formula(val_formula),
                   data       = data_train,
                   method     = modelos_resu[i, 2],
                   preProcess = val_preproc,
                   trControl  = val_ctrl,
                   tuneGrid   = eval(parse(text = modelos_resu[i, h+4])),
                   metric     = val_metrica,
                   verbose    = 0,
                   objective  = "binary:logistic",
                   eval_metric = "error")
   assign(paste("metod", h+2, sep=""), metodo)
}

for(j in 1:4)
{
   if (k==1) assign(paste("xgb", j, sep=""), get(paste("metod", j, sep="")))
   assign("aux", get(paste("metod", j, sep="")))
   # Evaluación de la perfomance del modelo en Train
   modelos_accu[j, i] <- max(aux$results$Accuracy)

   # Evaluación de la perfomance del modelo en Test
   metodo_clas       <- predict(aux, newdata = data_test)
   # Calcular el % de acierto (accuracy) o tasa de exactitud
   modelos_accu[j, i + methods_nro] <- mean(data_test[, val_target] == metodo
_clas)
}

#----------------------------
# 2.3.3.2 Comparación de los modelos
```

```r
#----------------------------
aux = which(modelos_accu[,i + methods_nro] == max(modelos_accu[,i + methods_n
ro]))

#----------------------------
# 2.3.3.3 Generación del modelo Final
#----------------------------
assign(modelos_resu[i,4], get(paste("metod", aux, sep="")))


#----------------------------
# 2.3.4 Creación de los Modelos de Stacking
#       Haciendo ensamble de modelos con otros modelos
#----------------------------
#----------------------------
# 2.3.4.1 Generando los Modelos
#----------------------------
# Definiendo el tuneGrid para cada método
# Hiperparametros solo con TuneLength definidos
stk_grid = list(
   nnet = caretModelSpec(method = modelos_resu[1, 2],
                         preProcess = val_preproc,
                         tuneLength = as.integer(modelos_resu[1, 3]),
                         metric     = val_metrica,
                         trace      = F,
                         lineout    = F),
   gbm  = caretModelSpec(method = modelos_resu[2, 2],
                         preProcess = val_preproc,
                         tuneLength = as.integer(modelos_resu[2, 3]),
                         metric     = val_metrica,
                         verbose    = F,
                         distribution = "bernoulli"),
   xgbTree  = caretModelSpec(method = modelos_resu[3, 2],
                             preProcess = val_preproc,
                             tuneLength = as.integer(modelos_resu[3, 3]),
                             metric     = val_metrica,
                             verbose    = 0,
                             objective  = "binary:logistic",
                             eval_metric = "error")
)

# Hiperparametros definidos con los valores de los modelos
grid_stk = list(
   nnet = caretModelSpec(method = modelos_resu[1, 2],
                         preProcess = if(is.null(metodo_rna$preProcess)) NULL
else val_preproc,
                         tuneGrid   = metodo_rna[["bestTune"]],
                         metric     = metodo_rna[["metric"]],
                         trace      = F,
                         lineout    = F),
   gbm  = caretModelSpec(method = modelos_resu[2, 2],
                         preProcess = if(is.null(metodo_gbm$preProcess)) NULL
else val_preproc,
                         tuneGrid   = metodo_gbm$bestTune,
                         metric     = metodo_gbm$metric,
                         verbose    = F,
                         distribution = "bernoulli"),
   xgbTree  = caretModelSpec(method = modelos_resu[3, 2],
                             preProcess = if(is.null(metodo_xgb$preProcess))
```

```
NULL else val_preproc,
                              tuneGrid   = metodo_xgb$bestTune,
                              metric     = metodo_xgb$metric,
                              verbose    = 0,
                              objective  = "binary:logistic",
                              eval_metric = "error")
)

j = dim(modelos_resu)[1]
i = j - 1
## [1] "Ejecutando el Modelo: Stacking en el curso: 0001"

# Pruebas con hiperparametros definidos
for(h in 1:2)
{
   set.seed(123)
   stacking_list <- caretList(# Si usamos formula :
      as.formula(val_formula),
      data       = data_train, # dataset donde se aplica la formula
      tuneList = eval(parse(text = modelos_resu[j, h+4])),
      trControl = val_ctrl)
   assign(paste("metod", h+2, sep=""), stacking_list)
   assign(paste("metod", h, sep=""), caretEnsemble(stacking_list))

   assign("metodo", get(paste("metod", h, sep="")))
   # Evaluación de la perfomance del modelo en Train
   modelos_accu[h, i] <- max(metodo$ens_model$results$Accuracy)

   # Evaluación de la perfomance del modelo en Test
   metodo_clas       <- predict(metodo, newdata = data_test)
   # Calcular el % de acierto (accuracy) o tasa de exactitud
   modelos_accu[h, i + methods_nro] <- 1 - mean(data_test[, val_target] == me
todo_clas)
   if (k==6) modelos_accu[h, i + methods_nro] <- mean(data_test[, val_target]
== metodo_clas)
}

#----------------------------
# 2.3.4.2 Comparación de los modelos
#----------------------------
aux = which(modelos_accu[,i + methods_nro] == max(modelos_accu[,i + methods_n
ro]))

#----------------------------
# 2.3.4.3 Generación del modelo Final
#----------------------------
assign("stacking_list", get(paste("metod", aux+2, sep="")))
assign(modelos_resu[j,4], get(paste("metod", aux, sep="")))


#----------------------------
# 2.4 Almacenando Modelos
#----------------------------
# Guarda los modelos (objetos) en un archivo de datos de R
if (k==1) save(modelos_resu, modelos_accu,
               rna1, rna2, rna3, rna4,
               gbm1, gbm2, gbm3, gbm4,
               xgb1, xgb2, xgb3, xgb4,
               val_target, modelos_resu, val_ctrl, data_train, data_test,
```

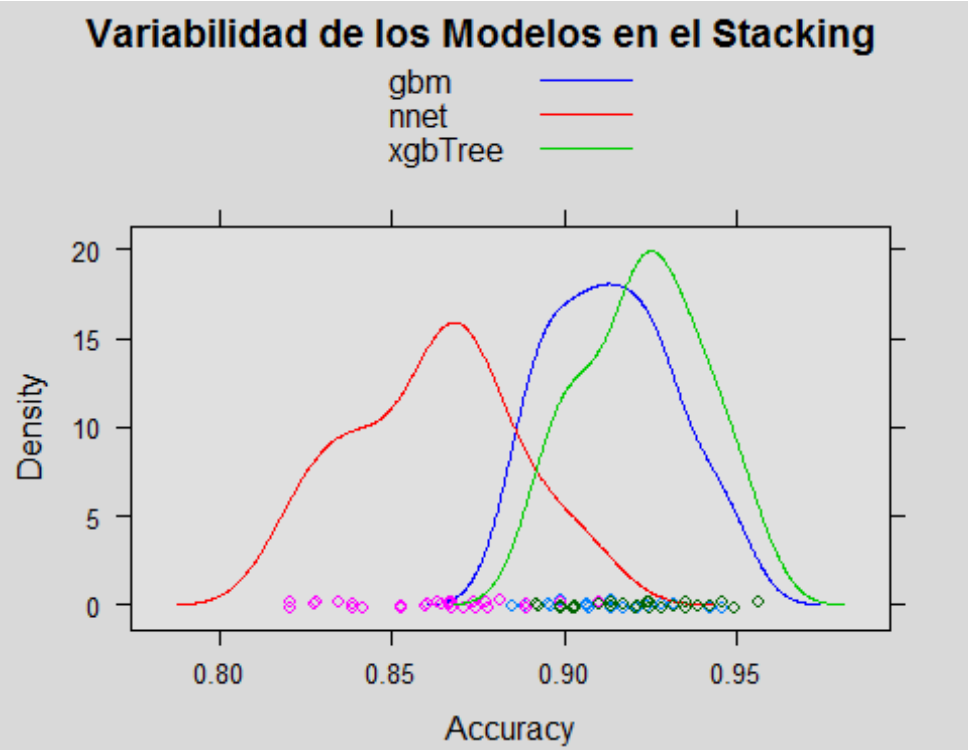
```
              metod1, metod2, metod3, metod4,
              metodo_rna, metodo_gbm, metodo_xgb, metodo_stk, stacking_list,
              file = paste("modelos_", file, ".RData", sep=""))

modelos_resu = modelos_resu[, 1:4]
save(val_target, modelos_resu, data_test,
     metodo_rna, metodo_gbm, metodo_xgb, metodo_stk,
     file = paste("tuning_DS_", file, ".RData", sep=""))

# Grabando archivo de indicadores
write.csv(modelos_accu,
          file = paste("tuning_", file, ".csv", sep=""),
          quote = FALSE)



##############################
# 3.0 Comparacion de la Importancia de las Variables en los Modelos
##############################
#---------------------------
# 3.1 En los modelos independientes
#---------------------------
# Modelo Red Neuronal Artificial (RNA)
h = cbind(varImp(get(modelos_resu[1, 4]))$importance, # Obtengo Peso incremen
tal
          varImp(get(modelos_resu[1, 4]), scale = F)$importance) # Obtengo %
colnames(h) = c("Peso_RNA_1", "Peso_RNA_2")
h$Variable_RNA = row.names(h)
aux <- h[with(h, order(-h[1])), c(3,1:2)]

# Modelo Gradient Boosting Machine (GBM)
h = cbind(varImp(get(modelos_resu[2, 4]), useModel = F)$importance[1], # Obte
ngo Peso
          varImp(get(modelos_resu[2, 4]), useModel = F, scale = F)$importance
[1]) # Obtengo %
colnames(h) = c("Peso_GBM_1", "Peso_GBM_2")
h$Peso_GBM_2 = h$Peso_GBM_2 / sum(h$Peso_GBM_2) * 100
h$Variable_GBM = row.names(h)
aux <- cbind(aux, h[with(h, order(-h[1])), c(3,1:2)])

# Modelo XGBoosting
h = cbind(varImp(get(modelos_resu[3, 4]))$importance, # Obtengo Peso
          varImp(get(modelos_resu[3, 4]), scale = F)$importance * 100) # Obte
ngo %
colnames(h) = c("Peso_XGB_1", "Peso_XGB_2")
h$Variable_XGB = row.names(h)
aux <- cbind(aux, h[with(h, order(-h[1])), c(3,1:2)])
rownames(aux) <- NULL

## [1] "Importancia de las variables en los modelos: "

print(aux)

##   Variable_RNA   Peso_RNA_1   Peso_RNA_2 Variable_GBM Peso_GBM_1
## 1      veces_1 100.00000000 97.902730582      veces_1 100.000000
## 2       PRA1_1   1.31086467  1.290810287       PRA1_1  85.959254
## 3      car_cod   0.25209315  0.254323860    sem_dif_1  16.763211
## 4    sem_dif_1   0.21525906  0.218265050        sexoM  16.473704
## 5    escalaA33   0.13741253  0.142057043      car_cod  15.180066
```

```
## 6      ing_cod   0.12010713  0.125115892    escalaA33   5.260137
## 7        sexoM   0.05273367  0.059160506      ing_cod   3.147837
## 8    escalaA23   0.00000000  0.007536781    escalaA23   0.000000
##   Peso_GBM_2 Variable_XGB   Peso_XGB_1 Peso_XGB_2
## 1   16.88447      veces_1 100.00000000 44.4709478
## 2   16.00063       PRA1_1  61.81845300 27.6347006
## 3   11.64486    sem_dif_1  26.22958533 11.9417019
## 4   11.62664      car_cod  18.36880585  8.4754718
## 5   11.54520      ing_cod  12.26302021  5.7831106
## 6   10.92076    escalaA33   1.25403930  0.9286734
## 7   10.78779        sexoM   0.03172816  0.3896923
## 8   10.58964    escalaA23   0.00000000  0.3757017

# Grabando Importancia de las variables
write.csv(as.data.frame(aux),
          file = paste("importancia_", file, ".csv", sep=""),
          row.names = FALSE,
          quote = FALSE)

#----------------------------
# 3.1.1 Interpretabilidad de las principales variables en el mejor metodo
#----------------------------
# Determinando el mejor Modelo
i = which.max(apply(modelos_accu[(models_nro+2):((models_nro+1)*2-1)], 2, max
))

# Determinando las variables a utilizar del modelo
h = aux[, (1+(3*(i-1)))]

# Evaluación de la perfomance del mejor modelo en Train
assign("aux", get(modelos_resu[i, 4]))
metodo_prob <- predict(aux,
                       newdata = data_train,
                       type = "prob")[,2]

#----------------------------
# 3.1.1.2 Probabilidad en cuartiles
#----------------------------
# Se realiza una division en cuartiles de la data y en cada division se
#  calcula el promedio de la probabilidad del modelo
# Generando la interpretabilidad por cada variable por cuartiles
for(j in 1:length(h))
{
   # Con la probabilidad del metodo
   aux <- data.frame(x = data_train[, h[j]],
                     y = metodo_prob,
                     clase = 4)

   if (h[j] == "car_cod" || h[j] == "ing_cod") {
      metodo <- aux %>%
         select(x, y) %>%
         group_by(x) %>%
         summarize(y = mean(y), n = n()) %>%
         filter(n >= round(dim(data_train)[1] * 0.01))

      # Grafico de Probabilidades
      aux = ggplot(data = metodo,
                   aes(x = x,
                       y = y)) +
```

```
         geom_line(colour="red")  +
         geom_point(size=1.5, shape=21, fill="white", colour="red") +
         stat_smooth(method = 'loess', formula ='y ~ x', level = 0.1, se = F)
+
         labs(title = paste0("Probabilidades en el Curso: ", file),
              y = "Probabilidad de Aprobar",
              x = paste0("Variable: ",
                         if (h[j] == "car_cod") "Carrera" else "Modalidad de
Ingreso")) +
         theme(legend.position = "none")

   } else if (substr(h[j], 1, 4) == "sexo") {
      aux$x = factor(round(aux$x), levels = c(0, 1), labels = c("Femenino", "
Masculino"))
      metodo <- aux %>%
         select(x, y) %>%
         group_by(x) %>%
         summarize(y = mean(y), n = n())

      # Grafico de Probabilidades
      aux = ggplot(data = metodo,
                   aes(x = x,
                       y = y)) +
         geom_line(colour="red")  +
         geom_point(size=1.5, shape=15, colour="red") +
         stat_smooth(method = 'loess', formula ='y ~ x', level = 0.1, se = F)
+
         labs(title = paste0("Probabilidades en el Curso: ", file),
              y = "Probabilidad de Aprobar",
              x = "Variable: Sexo") +
         theme(legend.position = "none")

   } else if (substr(h[j], 1, 6) == "escala") {

   } else if (substr(h[j], 1, 7) == "coltipo") {
      aux$x = factor(round(aux$x), levels = c(0, 1), labels = c("Estatal", "P
articular"))
      metodo <- aux %>%
         select(x, y) %>%
         group_by(x) %>%
         summarize(y = mean(y), n = n())

      # Grafico de Probabilidades
      aux = ggplot(data = metodo,
                   aes(x = x,
                       y = y)) +
         geom_path(colour="red")  +
         geom_point(size=1.5, shape=2, fill="white", colour="red") +
         labs(title = paste0("Probabilidades en el Curso: ", file),
              y = "Probabilidad de Aprobar",
              x = "Variable: Tipo de Colegio") +
         theme(legend.position = "none")

   } else {
      metodo = data.frame(limite=quantile(aux$x, probs = seq(0, 1, 0.25)))
      aux$clase = ifelse (aux$x <= metodo[4,1], 3, aux$clase)
      aux$clase = ifelse (aux$x <= metodo[3,1], 2, aux$clase)
      aux$clase = ifelse (aux$x <= metodo[2,1], 1, aux$clase)
      metodo <- aux %>%
```

```
            select(clase, y) %>%
            group_by(clase) %>%
            summarize(y = mean(y))

         # Grafico de Probabilidades
         aux = ggplot(data = metodo,
                      aes(x = clase,
                          y = y)) +
            geom_line(colour="red")  +
            labs(title = paste0("Probabilidades en el Curso: ", file),
                 y = "Probabilidad de Aprobar",
                 x = paste0("Variable: ", h[j])) +
            xlim(c("Cuartil 1", "Cuartil 2", "Cuartil 3", "Cuartil 4")) +
            theme(legend.position = "none")
      }
      if (substr(h[j], 1, 6) != "escala") print(aux)
   }
   # Grafico de Probabilidades de las escalas
   i = 1
   for(j in 1:length(h))
   {
      if (substr(h[j], 1, 6) == "escala") {
         aux  <- data.frame(x = data_train[, h[j]],
                            y = metodo_prob)
         aux$x = factor(round(aux$x), levels = c(0, 1), labels = c("No", "Si"))
         aux  <- matrix(c(substr(h[j], 7, 20),
                          "Si",
                          round(mean(aux[aux$x=="Si", "y"]),3),
                          substr(h[j], 7, 20),
                          "No",
                          round(mean(aux[aux$x=="No", "y"]),3)),
                        nrow = 2,
                        ncol = 3,
                        byrow = TRUE,
                        dimnames = list(NULL,
                                        c("Escala", "Clase", "Probabilidad")))
         if (i == 1) {
            i = 2
            metodo <- as.data.frame(aux, stringsAsFactors = F)
         } else {
            metodo <- rbind(metodo,
                            as.data.frame(aux, stringsAsFactors = F),
                            stringsAsFactors = F)
         }
      }
   }

   # Grafico de Probabilidades de las escalas
   if (i == 2) {
      aux = ggplot(data = metodo,
                   aes(x = Clase,
                       y = Probabilidad,
                       by = Escala)) +
         geom_point(aes(color = Escala, shape = Escala), size = 2) +
         labs(title = paste0("Probabilidades en el Curso: ", file),
              y = "Probabilidad de Aprobar",
              x = "Variable: Escala de pago")
      print(aux)
   }
```

```r
#----------------------------
# 3.2 En el ensamble y en el stacking
#----------------------------
# NO es posible la comparacion, debido a que es una combinacion de modelos

##############################
# 4.0 Exploración de los modelos
##############################
#----------------------------
# 4.1 Comparación de los modelos en el stacking
#----------------------------
## [1] "Caracteristicas internas del modelo Stacking"

compara_stack <- resamples(stacking_list)

if (k==1) {
   i = sort(apply(compara_stack$values[2:(2*models_nro+1)], 2, IQR))

   # Calculando la menor variabilidad
   h = names(i[which(i == min(i[1:models_nro]))])
   print(paste("Menor Variabilidad:",
               modelos_resu[which(modelos_resu[,2] == strsplit(h, "~")[[1]][1
]), 1]))
   # Graficamente
   aux = densityplot(compara_stack,
                     metric = "Accuracy",
                     par.settings = list(superpose.line = list(col = c("blue"
, "red", "green3")),
                                         background = list(col = c("gray85"))
,
                                         panel.background = list(col = c("gra
y88"))),
                     auto.key = TRUE,
                     main = "Variabilidad de los Modelos en el Stacking")
   print(aux)
}
```




```
#----------------------------
# 4.2 Comparación de la correlacion entre los modelos en el stacking
#----------------------------
## [1] "Correlacion:"

print(modelCor(compara_stack))

##              nnet       gbm   xgbTree
## nnet    1.0000000 0.7783568 0.5510740
## gbm     0.7783568 1.0000000 0.8023699
## xgbTree 0.5510740 0.8023699 1.0000000

## [1] "***** Fin de Ejecución *****"
```

Anexo 10: Script del algoritmo en R para la generación de los modelos de cada curso del Programa de Estudios Básicos (PEB).

Procederemos a generar los indicadores de los modelos en cada Curso, para poder realizar una evaluación de los mismos.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")

# Uso de Librerías
library(caret)
library(gmodels) # CrossTable
library(InformationValue) # ks_stat
library(caTools) # colAUC
library(ModelMetrics) # logLoss
library(lattice) # splom
```

Para la generación de indicadores para el Curso con código: "0002" en adelante, los resultados se visualizarán en el Anexo 15.

---

1. Archivo del Curso con código: "0001"

Generación de los indicadores de los modelos en el curso denominado: "Actividades Artísticas y Deportivas".

```
# Variables relacionadas con los métodos
val_compara <- c("RMSE", "% Error", "Accuracy", "Sensibilidad", "Especificida
d",
                 "Precision", "Coe. K-S", "Curva ROC", "Coe. Gini", "logLoss"
,
                 "Accuracy_train")

k=1
file = formatC(k, width = 4, flag = "0")
## [1] "***** Inicio - Curso: 0001*****"

##############################
# 1.0 Lectura de los datos
##############################
load(paste("tuning_DS_", file, ".RData", sep=""))
models_nro <- dim(modelos_resu)[1]

##############################
# 2.0 Generando Modelos
##############################
#---------------------------
# 2.1 Generando el dataframe para los metodos
#---------------------------
# Modificando el dataframe para almacenar los indicadores de los metodos
aux          <- matrix(rep(0, models_nro*length(val_compara)),
```

```
                      nrow = models_nro,
                      ncol = length(val_compara),
                      dimnames = list(NULL,
                                      val_compara))
modelos_resu <- cbind(modelos_resu,
                      as.data.frame(aux),
                      stringsAsFactors = F)
aux          <- matrix(rep(0, models_nro*4),
                      nrow = models_nro,
                      ncol = 4,
                      dimnames = list(NULL,
                                      c("Real_P", "Real_N", "Pred_P", "Pred_
N")))
modelos_resu <- cbind(modelos_resu,
                      as.data.frame(aux),
                      stringsAsFactors = F)

#----------------------------
# 2.3 Generando los Indicadores de los Modelos
#----------------------------
for(i in 1:models_nro)
{
   print(paste("Ejecutando el Modelo: ", modelos_resu[i,1], " en el curso: ",
file, sep=""))
   if (i==1) {
      assign("metodo", get(modelos_resu[i,4]))
      models_list <- list(metodo)
   } else if (i==2 || i==3 || i==5) {
      assign("metodo", get(modelos_resu[i,4]))
      if (i!=5) models_list <- append(models_list, list(metodo))
   } else {
      for(j in 1:as.integer(modelos_resu[i, 3])) {
         if (j==1) {
            metodo_prob <- models_prob[[j]]
            aux = max(models_list[[j]]$results$Accuracy)
         } else {
            metodo_prob <- metodo_prob + models_prob[[j]]
            aux = aux + max(models_list[[j]]$results$Accuracy)
         }
      }
   }

   # 2.3.0 Asignar accuracy del metodo
   if (i==1 || i==2 || i==3) {
      modelos_resu[i,15] <- max(metodo$results$Accuracy)
   } else if (i==4) {
      modelos_resu[i,15] <- aux / as.integer(modelos_resu[i, 3])
   } else {
      modelos_resu[i,15] <- max(metodo$ens_model$results$Accuracy)
   }

   # 2.3.1 Predecir la probabilidad de ocurrencia de la prediccion
   if (i==1 || i==2 || i==3 || i==5) {
      metodo_prob <- predict(metodo,
                             newdata = data_test,
                             type = "prob")

      if (i==1) models_prob <- list(metodo_prob[,2])
      if (i!=1 && i!=5) models_prob <- append(models_prob, list(metodo_prob[,
```

```
2]))
   } else {
      metodo_prob <- metodo_prob / as.integer(modelos_resu[i, 3])
   }

   # 2.3.2 Evaluación de la perfomance del modelo
   if (i==1 || i==2 || i==3 || i==5) {
      metodo_clas <- predict(metodo,
                             newdata = data_test)
   } else {
      metodo_clas <- as.factor(ifelse(metodo_prob > 0.5, "Si", "No"))
   }

   # 2.3.3 Matriz o Tabla de Clasificacion
   CrossTable(x = data_test[, val_target],
              y = metodo_clas,
              prop.t = FALSE,
              prop.c = FALSE,
              prop.chisq = FALSE,
              dnn = c("Real", "Prediccion"))

   # Obtencion de Sensibilidad, Especificidad y Precision
   aux <- caret::confusionMatrix(metodo_clas,
                                 data_test[, val_target],
                                 dnn = c("Predicción", "Real"),
                                 positive="Si")
   modelos_resu[i, 8] <- aux$byClass[1]
   modelos_resu[i, 9] <- aux$byClass[2]
   modelos_resu[i,10] <- aux$byClass[5]

   modelos_resu[i,16] <- aux$table[3] + aux$table[4]
   modelos_resu[i,17] <- aux$table[1] + aux$table[2]
   modelos_resu[i,18] <- aux$table[4]
   modelos_resu[i,19] <- aux$table[1]

   aux   <- as.numeric(data_test[, val_target]) - as.numeric(metodo_clas)
   modelos_resu[i, 5] <- sqrt(mean(aux^2))

   modelos_resu[i, 7] <- mean(data_test[, val_target] == metodo_clas)

   modelos_resu[i, 6] <- mean(data_test[, val_target] != metodo_clas)

   modelos_resu[i,11] <- ks_stat(as.numeric(data_test[, val_target]),
                                 as.numeric(metodo_clas),
                                 returnKSTable = F)

   modelos_resu[i,12] <- colAUC(as.numeric(metodo_clas),
                                as.numeric(data_test[, val_target]),
                                plotROC = TRUE)
   abline(0, 1, col = "red", lty = 2)
   text(0.5, 0, pos = 3, cex = 0.8, font = 2,
        paste("Modelo: ", modelos_resu[i, 1], sep=""))

   modelos_resu[i,13] <- 2*modelos_resu[i, 8] - 1

   modelos_resu[i,14] <- logLoss(as.numeric(data_test[, val_target]),
                                 as.numeric(metodo_clas))
}
```

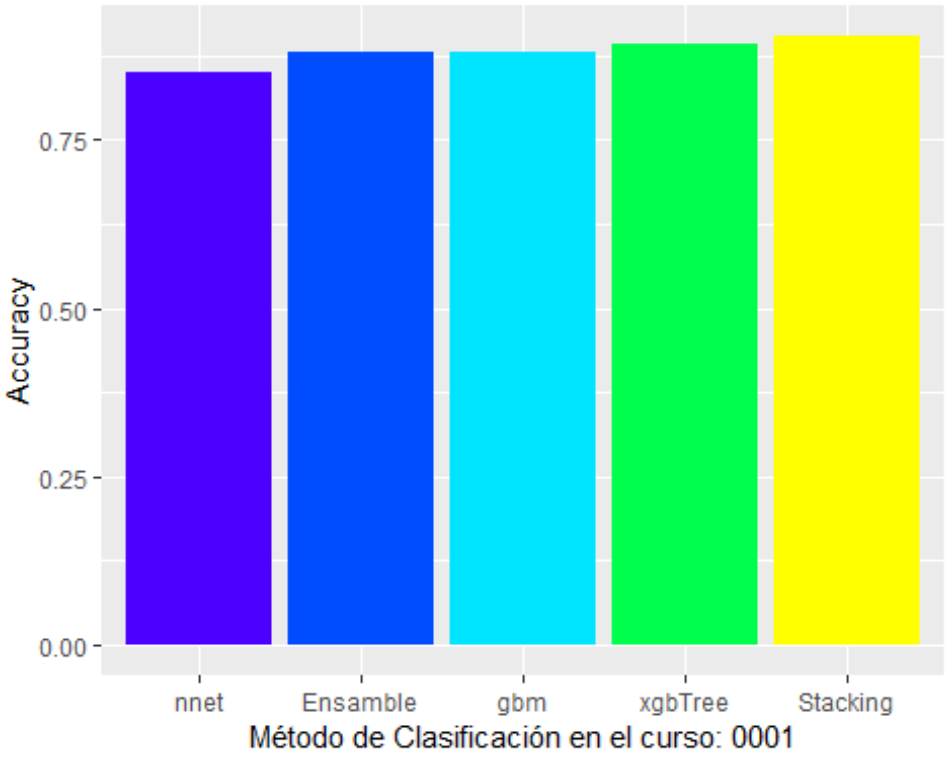
```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0001
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
01"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0001"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0001"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0001"
```

```r
#-----------------------------------------
# 2.6 Grabando archivo de indicadores
#-----------------------------------------
write.csv(modelos_resu[, c(1, 7, 8, 9, 12, 15:19)],
          file = paste("modelo_", file, ".csv", sep=""),
          row.names = FALSE, quote = FALSE)

############################
# 3.3 Comparación de los modelos
############################
test_compara <- which(val_compara == "Accuracy") # es 3
#---------------------------
# 3.3.1 Comparación de los resultados por un coeficiente
#---------------------------
# Grafico de comparacion segun el indicador
test_metodos <- modelos_resu[, 4+test_compara]
test_resulta <- data.frame(modelos_resu[, 2], test_metodos)

# El mejor metodo resulto ser:
print(modelos_resu[which(test_metodos == max(test_metodos)), 2])
```

```
## [1] "Stacking"
```

```r
aux <- ggplot(data = test_resulta,
              aes(x = reorder(modelos_resu[, 2], test_metodos),
                  y = test_metodos)) +
   geom_bar(stat = "identity",
            fill = topo.colors(models_nro)) +
   labs(x = paste("Método de Clasificación", " en el curso: ", file, sep=""),
```

```
y = val_compara[test_compara])
```



Anexo 11: Resultados de la comprensión de cada curso del Programa de Estudios Básicos (PEB) (PEB).

Procederemos a revisar los Archivos de notas de cada curso del Programa de Estudios Básicos (PEB), con la finalidad de comprender su contenido.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")
```

Para la comprensión del Curso con código: "0001" sírvase revisar el Anexo 06.

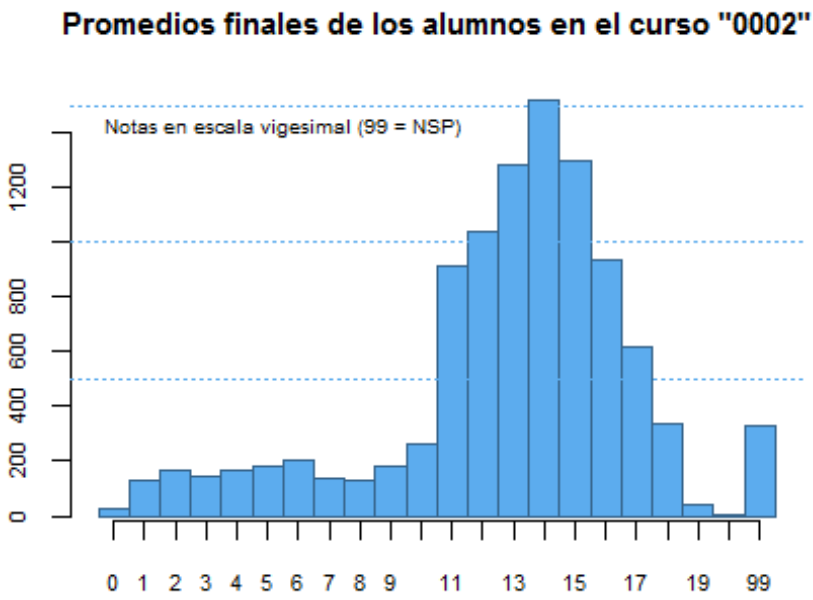
---

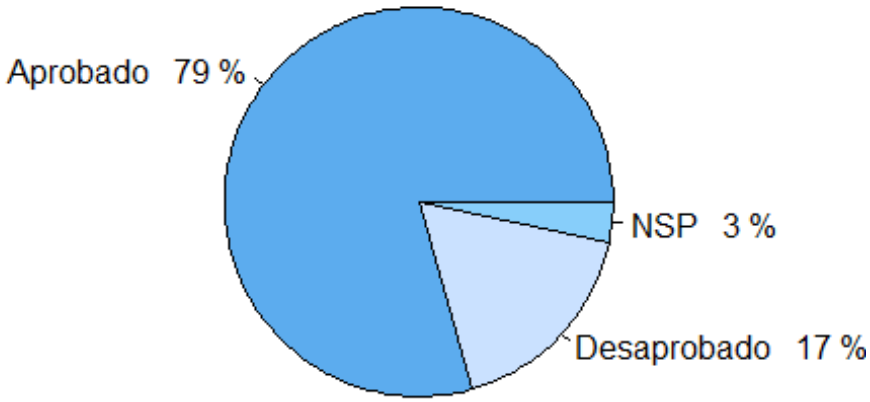
1. Visualización de la estructura interna de los Cursos

0002: "Taller de Método de Estudio Universitario".
0003: "Taller de Comunicación Oral y Escrita I".
0004: "Matemática".
0005: "Inglés I".
0006: "Psicología General".
0007: "Lógica y Filosofía".
0008: "Taller de Comunicación Oral y Escrita II".
0009: "Inglés II".
0010: "Formación Histórica del Perú".
0011: "Recursos Naturales y Medio Ambiente".
0012: "Realidad Nacional".
0013: "Historia de la Civilización".

```
## [1] "***** Inicio - Curso: 0002*****"
##
## [1] "El curso original"
## 'data.frame':    59917 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 16 levels "PRA1","PRA2",..: 5 6 7 8 11 12 5 6 7 8 .
##  $ eva_nota: num  15 14 11 1 5 10 15 15 13 14 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201512432 201512
432 201510876 201510876 201510876 201510876 ...
##  $ eva_fin : int  9 9 9 9 9 9 15 15 15 15 ...
##
## PRA1 PRA2 PRA3 PRA4 PTL1 PTL2 PTL3 PTL4 PYT1 PYT2 TLR1 TLR2 TLR3 TLR4 TRP1
## 5056 5063 5063 5029 4941 4941 4941 4941   35   35 4855 4855   35   35 5063
## TRP2
## 5029
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    59917 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 6 levels "PRA1","PRA2",..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ eva_nota: num  15 14 11 1 5 10 15 15 13 14 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201512432 201512
432 201510876 201510876 201510876 201510876 ...
```

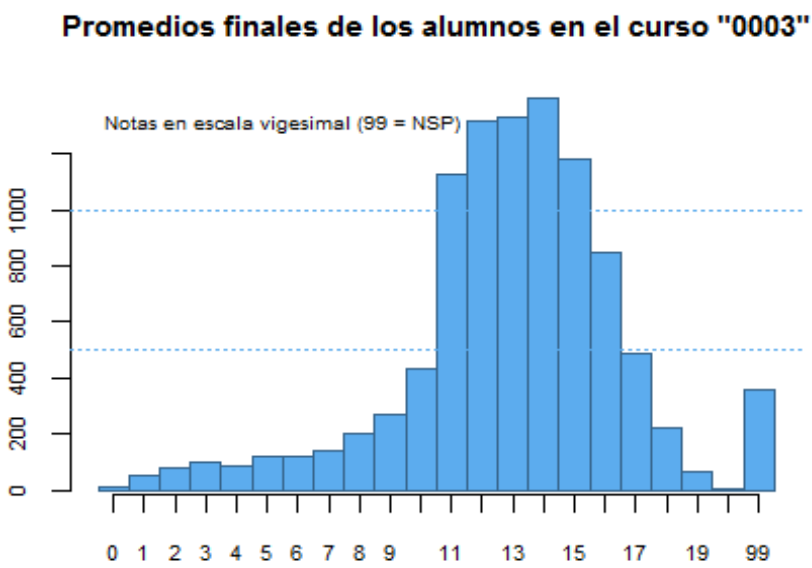
```
##  $ eva_fin : int  9 9 9 9 9 9 15 15 15 15 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151    PRA1       15 201512432       9
## 2     20151    PRA2       14 201512432       9
## 3     20151    PRA3       11 201512432       9
## 4     20151    PRA4        1 201512432       9
## 5     20151    PRA5        5 201512432       9
## 6     20151    PRA6       10 201512432       9
## 7     20151    PRA1       15 201510876      15
## 8     20151    PRA2       15 201510876      15
## 9     20151    PRA3       13 201510876      15
## 10    20151    PRA4       14 201510876      15
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    10039 obs. of  11 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1..
##  $ alu_cod : chr  "201512432" "201510876" "201512136" "201512049" ...
##  $ eva_fin : int  9 15 99 14 16 12 14 12 15 10 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 2 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 1 3 1 1 1 1 1 1 2 ...
##  $ PRA1    : num  15 15 0 9 15 15 14 12 17 11 ...
##  $ PRA2    : num  14 15 0 10 16 11 15 15 15 5 ...
##  $ PRA4    : num  1 14 0 17 10 9 12 11 17 11 ...
##  $ PRA5    : num  5 16 0 14 18 11 13 11 14 8 ...
##  $ PRA3    : num  11 13 0 16 17 15 15 14 11 13 ...
##  $ PRA6    : num  10 14 0 15 17 12 14 12 16 13 ...
```

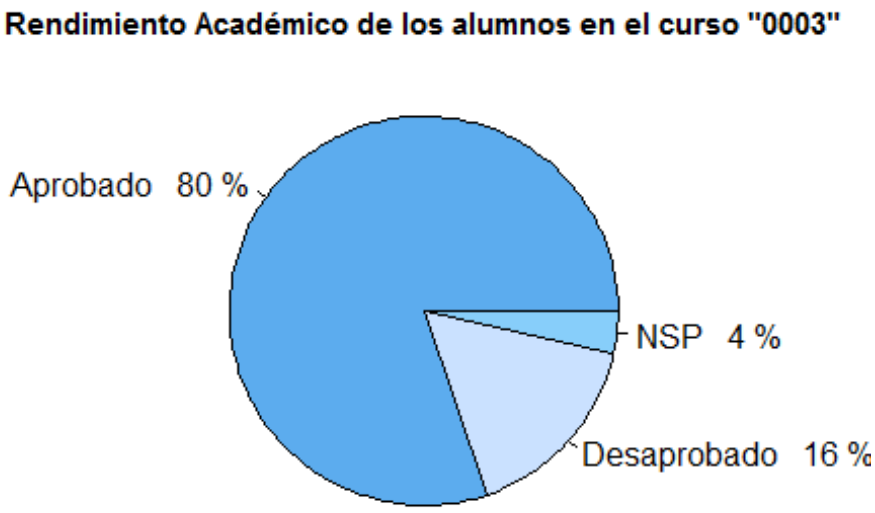





```
##   semestre   alu_cod eva_fin veces estado PRA1 PRA2 PRA4 PRA5 PRA3 PRA6
## 1    20151 201512432       9     2      D   15   14    1    5   11   10
## 2    20151 201510876      15     1      A   15   15   14   16   13   14
## 3    20151 201512136      99     1      N    0    0    0    0    0    0
## 4    20151 201512049      14     1      A    9   10   17   14   16   15
## 5    20151 201512175      16     1      A   15   16   10   18   17   17
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    9710 obs. of  11 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ alu_cod : int  201512432 201510876 201512049 201512175 201512706 201512
275 201512356 201511917 201511830 201512645 ...
##  $ eva_fin : int  9 15 14 16 12 14 12 15 10 18 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 2 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 2 2 2 2 2 2 2 1 2 ...
##  $ PRA1    : num  15 15 9 15 15 14 12 17 11 18 ...
##  $ PRA2    : num  14 15 10 16 11 15 15 15 5 18 ...
##  $ PRA4    : num  1 14 17 10 9 12 11 17 11 18 ...
##  $ PRA5    : num  5 16 14 18 11 13 11 14 8 18 ...
##  $ PRA3    : num  11 13 16 17 15 15 14 11 13 18 ...
##  $ PRA6    : num  10 14 15 17 12 14 12 16 13 18 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 PRA2 PRA4 PRA5 PRA3 PRA6
## 1    20151 201512432       9     2     No   15   14    1    5   11   10
## 2    20151 201510876      15     1     Si   15   15   14   16   13   14
## 3    20151 201512049      14     1     Si    9   10   17   14   16   15
## 4    20151 201512175      16     1     Si   15   16   10   18   17   17
## 5    20151 201512706      12     1     Si   15   11    9   11   15   12
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0003*****"
##
## [1] "El curso original"
## 'data.frame':    42167 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 9 levels "PRA1","PRA2",..: 5 6 7 8 9 5 6 7 8 9 ...
##  $ eva_nota: num  12 12 11 12 14 0 0 0 0 0 ...
##  $ alu_cod : int  201510876 201510876 201510876 201510876 201510876 201512
136 201512136 201512136 201512136 201512136 ...
##  $ eva_fin : int  12 12 12 12 12 99 99 99 99 99 ...
##
## PRA1 PRA2 PRA3 PRA4 PTL1 PTL2 PTL3 PTL4 PTL5
```

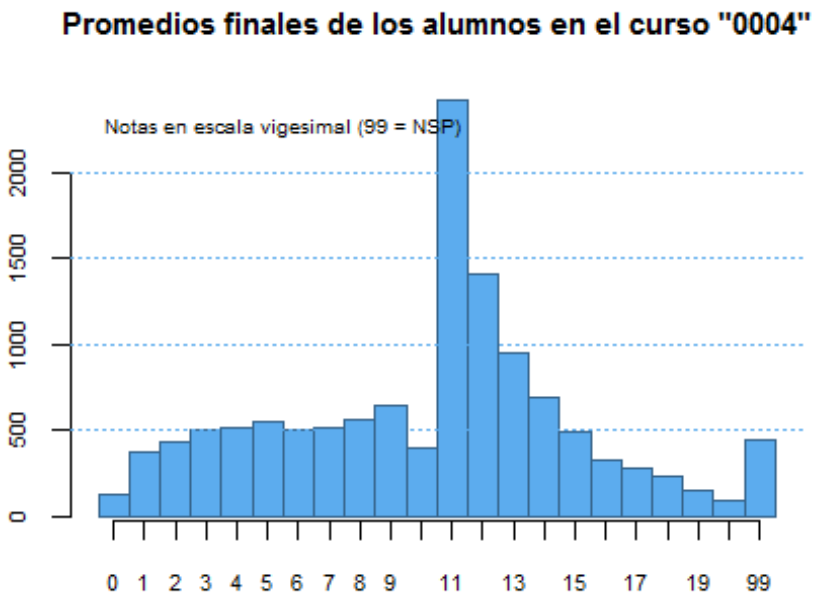
```
## 5040 5042 5042 5042 4891 4891 4888 4891 2440
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    42167 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 5 levels "PRA1","PRA2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ eva_nota: num  12 12 11 12 14 0 0 0 0 0 ...
##  $ alu_cod : int  201510876 201510876 201510876 201510876 201510876 201512
136 201512136 201512136 201512136 201512136 ...
##  $ eva_fin : int  12 12 12 12 12 99 99 99 99 99 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151    PRA1       12 201510876      12
## 2     20151    PRA2       12 201510876      12
## 3     20151    PRA3       11 201510876      12
## 4     20151    PRA4       12 201510876      12
## 5     20151    PRA5       14 201510876      12
## 6     20151    PRA1        0 201512136      99
## 7     20151    PRA2        0 201512136      99
## 8     20151    PRA3        0 201512136      99
## 9     20151    PRA4        0 201512136      99
## 10    20151    PRA5        0 201512136      99
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    9933 obs. of  10 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
##  $ alu_cod : chr  "201510876" "201512136" "201512049" "201512175" ...
##  $ eva_fin : int  12 99 11 17 13 15 13 16 16 17 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 3 1 1 1 1 1 1 1 ...
##  $ PRA5    : num  14 0 12 18 14 18 18 16 16 17 ...
##  $ PRA1    : num  12 0 12 11 12 16 13 15 15 15 ...
##  $ PRA2    : num  12 0 9 17 15 9 8 15 14 19 ...
##  $ PRA4    : num  12 0 12 20 12 14 15 14 17 17 ...
##  $ PRA3    : num  11 0 12 19 12 17 13 18 16 17 ...
```

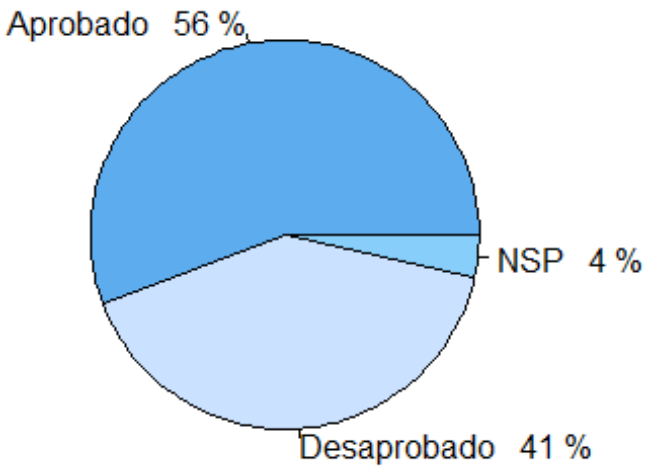





```
##   semestre   alu_cod eva_fin veces estado PRA5 PRA1 PRA2 PRA4 PRA3
## 1    20151 201510876      12     1      A   14   12   12   12   11
## 2    20151 201512136      99     1      N    0    0    0    0    0
## 3    20151 201512049      11     1      A   12   12    9   12   12
## 4    20151 201512175      17     1      A   18   11   17   20   19
## 5    20151 201512706      13     1      A   14   12   15   12   12
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    9576 obs. of  10 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ alu_cod : int  201510876 201512049 201512175 201512706 201512275 201512
356 201511917 201511830 201512645 201512056 ...
##  $ eva_fin : int  12 11 17 13 15 13 16 16 17 12 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA5    : num  14 12 18 14 18 18 16 16 17 12 ...
##  $ PRA1    : num  12 12 11 12 16 13 15 15 15 11 ...
##  $ PRA2    : num  12 9 17 15 9 8 15 14 19 12 ...
##  $ PRA4    : num  12 12 20 12 14 15 14 17 17 12 ...
##  $ PRA3    : num  11 12 19 12 17 13 18 16 17 12 ...
##   semestre   alu_cod eva_fin veces aprobo PRA5 PRA1 PRA2 PRA4 PRA3
## 1    20151 201510876      12     1     Si   14   12   12   12   11
## 2    20151 201512049      11     1     Si   12   12    9   12   12
## 3    20151 201512175      17     1     Si   18   11   17   20   19
## 4    20151 201512706      13     1     Si   14   12   15   12   12
## 5    20151 201512275      15     1     Si   18   16    9   14   17
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0004*****"
##
## [1] "El curso original"
## 'data.frame':    89576 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 13 levels "FIN1","PAR1",..: 1 2 3 4 5 6 13 1 2 3 ..
##  $ eva_nota: num  0 9 13 0 0 0 88 2 2 5 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201512432 201512
432 201512432 201510876 201510876 201510876 ...
##  $ eva_fin : int  4 4 4 4 4 4 4 3 3 3 ...
##
##  FIN1  PAR1  PRA1  PRA2  PRA3  PRA4  PRA5  PRA6  PRT1  PRT2  PRT3  PRT4
```

```
## 11906 11906 12444 12445 12445 12432  2393  1287   103   103   103   103
##  SUS1
## 11906
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    89576 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 9 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 6 9 1 2 3
##  $ eva_nota: num  0 9 13 0 0 0 88 2 2 5 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201512432 201512
432 201512432 201510876 201510876 201510876 ...
##  $ eva_fin : int  4 4 4 4 4 4 4 3 3 3 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151     FIN        0 201512432       4
## 2     20151     PAR        9 201512432       4
## 3     20151    PRA1       13 201512432       4
## 4     20151    PRA2        0 201512432       4
## 5     20151    PRA3        0 201512432       4
## 6     20151    PRA4        0 201512432       4
## 7     20151     SUS       88 201512432       4
## 8     20151     FIN        2 201510876       3
## 9     20151     PAR        2 201510876       3
## 10    20151    PRA1        5 201510876       3
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    12548 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ alu_cod : chr  "201512432" "201510876" "201512136" "201512049" ...
##  $ eva_fin : int  4 3 2 11 14 13 11 11 15 15 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 2 2 1 1 1 1 1 1 ...
##  $ PRA1    : num  13 5 0 10 10 14 17 4 16 20 ...
##  $ FIN     : num  0 2 0 9 13 13 7 8 15 15 ...
##  $ PRA2    : num  0 5 0 8 16 17 6 10 15 14 ...
##  $ PRA4    : num  0 4 0 17 11 15 2 11 7 16 ...
##  $ PRA3    : num  0 4 0 13 19 11 16 7 20 14 ...
##  $ PRA5    : num  NA 7 NA 9 NA 13 10 13 NA 6 ...
##  $ SUS     : num  88 88 88 13 88 88 88 88 88 88 ...
##  $ PRA6    : num  NA 6 NA NA NA NA 8 13 NA NA ...
##  $ PAR     : num  9 2 5 7 15 12 16 13 14 15 ...
```

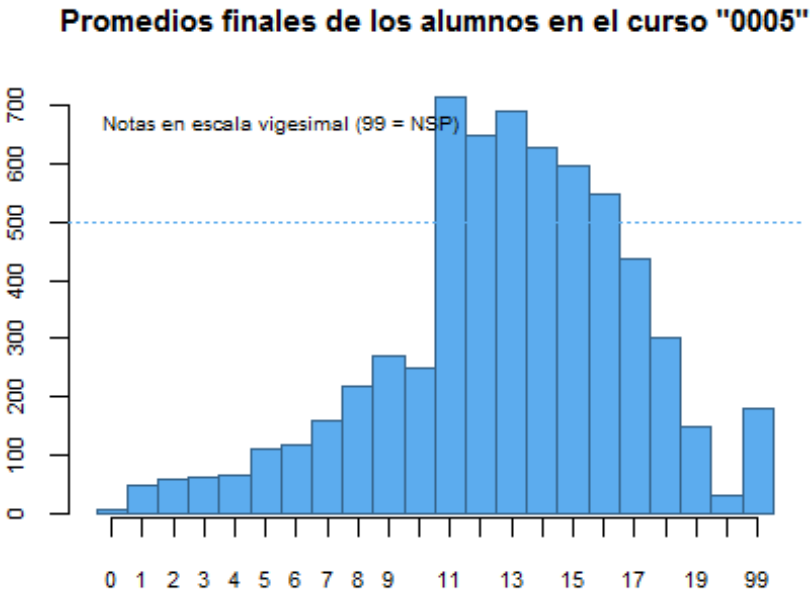





```
##   semestre   alu_cod eva_fin veces estado PRA1 FIN PRA2 PRA4 PRA3 PRA5 SUS
## 1    20151 201512432       4     2      D   13   0    0    0    0   NA  88
## 2    20151 201510876       3     1      D    5   2    5    4    4    7  88
## 3    20151 201512136       2     1      D    0   0    0    0    0   NA  88
## 4    20151 201512049      11     1      A   10   9    8   17   13    9  13
## 5    20151 201512175      14     1      A   10  13   16   11   19   NA  88
##   PRA6 PAR
## 1   NA   9
## 2    6   2
## 3   NA   5
## 4   NA   7
## 5   NA  15
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    12102 obs. of  14 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ alu_cod : int  201512432 201510876 201512136 201512049 201512175 201512
706 201512275 201512356 201511917 201511830 ...
```

```
##  $ eva_fin : int  4 3 2 11 14 13 11 11 15 15 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 1 1 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  13 5 0 10 10 14 17 4 16 20 ...
##  $ FIN     : num  0 2 0 9 13 13 7 8 15 15 ...
##  $ PRA2    : num  0 5 0 8 16 17 6 10 15 14 ...
##  $ PRA4    : num  0 4 0 17 11 15 2 11 7 16 ...
##  $ PRA3    : num  0 4 0 13 19 11 16 7 20 14 ...
##  $ PRA5    : num  NA 7 NA 9 NA 13 10 13 NA 6 ...
##  $ SUS     : num  88 88 88 13 88 88 88 88 88 88 ...
##  $ PRA6    : num  NA 6 NA NA NA NA 8 13 NA NA ...
##  $ PAR     : num  9 2 5 7 15 12 16 13 14 15 ...
##    semestre   alu_cod eva_fin veces aprobo PRA1 FIN PRA2 PRA4 PRA3 PRA5 SUS
## 1    20151 201512432       4     2     No   13   0    0    0    0   NA  88
## 2    20151 201510876       3     1     No    5   2    5    4    4    7  88
## 3    20151 201512136       2     1     No    0   0    0    0    0   NA  88
## 4    20151 201512049      11     1     Si   10   9    8   17   13    9  13
## 5    20151 201512175      14     1     Si   10  13   16   11   19   NA  88
##   PRA6 PAR
## 1   NA   9
## 2    6   2
## 3   NA   5
## 4   NA   7
## 5   NA  15
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0005*****"
##
## [1] "El curso original"
## 'data.frame':    31217 obs. of  5 variables:
##  $ semestre: int  20161 20161 20161 20161 20161 20161 20161 20161 20161...
##  $ eva_cod : Factor w/ 6 levels "PRA1","PRA2",..: 1 2 3 4 6 1 2 3 4 5 ...
##  $ eva_nota: num  19 13 17 15 20 13 9 11 15 13 ...
##  $ alu_cod : int  201611022 201611022 201611022 201611022 201611022 201611
857 201611857 201611857 201611857 201611857 ...
##  $ eva_fin : int  17 17 17 17 17 12 12 12 12 12 ...
##
## PRA1 PRA2 PRA3 PRA4 PRA5 TRP1
## 6270 6272 6272 6272 5704  427
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    31217 obs. of  5 variables:
##  $ semestre: int  20161 20161 20161 20161 20161 20161 20161 20161 20161...
##  $ eva_new : Factor w/ 5 levels "PRA1","PRA2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ eva_nota: num  19 13 17 15 20 13 9 11 15 13 ...
##  $ alu_cod : int  201611022 201611022 201611022 201611022 201611022 201611
857 201611857 201611857 201611857 201611857 ...
##  $ eva_fin : int  17 17 17 17 17 12 12 12 12 12 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20161    PRA1       19 201611022      17
## 2     20161    PRA2       13 201611022      17
## 3     20161    PRA3       17 201611022      17
## 4     20161    PRA4       15 201611022      17
## 5     20161    PRA5       20 201611022      17
## 6     20161    PRA1       13 201611857      12
## 7     20161    PRA2        9 201611857      12
## 8     20161    PRA3       11 201611857      12
## 9     20161    PRA4       15 201611857      12
## 10    20161    PRA5       13 201611857      12
```

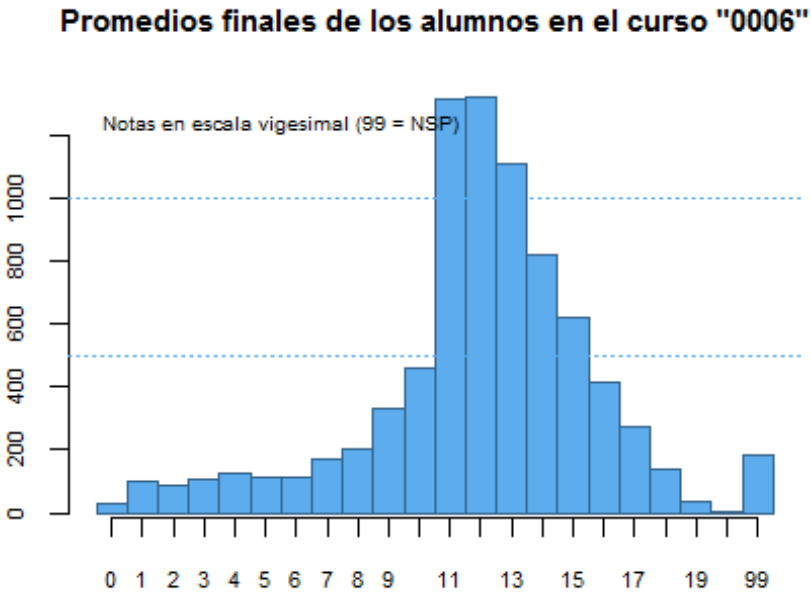
```
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    6272 obs. of  10 variables:
##  $ semestre: Factor w/ 9 levels "20161","20162",..: 1 1 1 1 1 1 1 1 1 ..
##  $ alu_cod : chr  "201611022" "201611857" "201610308" "201610173" ...
##  $ eva_fin : int  17 12 19 11 18 11 11 12 11 4 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 2 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 2 ...
##  $ PRA5    : num  20 13 18 13 20 13 9 9 14 0 ...
##  $ PRA1    : num  19 13 20 14 19 14 11 11 10 6 ...
##  $ PRA2    : num  13 9 18 11 18 6 7 11 11 1 ...
##  $ PRA4    : num  15 15 19 15 17 12 15 11 7 12 ...
##  $ PRA3    : num  17 11 19 0 18 9 11 17 11 0 ...
```

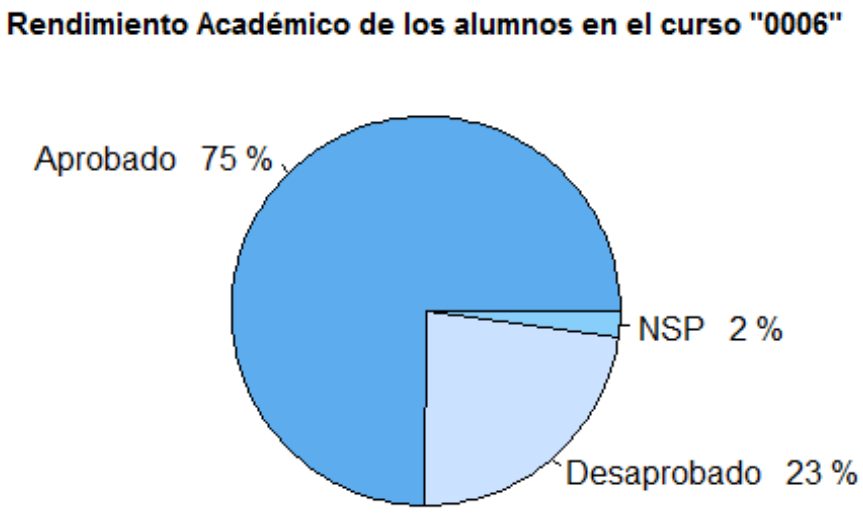



```
##   semestre   alu_cod eva_fin veces estado PRA5 PRA1 PRA2 PRA4 PRA3
## 1    20161 201611022      17     1      A   20   19   13   15   17
## 2    20161 201611857      12     1      A   13   13    9   15   11
## 3    20161 201610308      19     1      A   18   20   18   19   19
## 4    20161 201610173      11     1      A   13   14   11   15    0
```

```
## 5    20161 201610918      18     1      A   20   19   18   17   18
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    6093 obs. of  10 variables:
##  $ semestre: int  20161 20161 20161 20161 20161 20161 20161 20161 20161...
##  $ alu_cod : int  201611022 201611857 201610308 201610173 201610918 201610
399 201610898 201611669 201611246 201610902 ...
##  $ eva_fin : int  17 12 19 11 18 11 11 12 11 4 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 2 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 1 ...
##  $ PRA5    : num  20 13 18 13 20 13 9 9 14 0 ...
##  $ PRA1    : num  19 13 20 14 19 14 11 11 10 6 ...
##  $ PRA2    : num  13 9 18 11 18 6 7 11 11 1 ...
##  $ PRA4    : num  15 15 19 15 17 12 15 11 7 12 ...
##  $ PRA3    : num  17 11 19 0 18 9 11 17 11 0 ...
##   semestre   alu_cod eva_fin veces aprobo PRA5 PRA1 PRA2 PRA4 PRA3
## 1    20161 201611022      17     1     Si   20   19   13   15   17
## 2    20161 201611857      12     1     Si   13   13    9   15   11
## 3    20161 201610308      19     1     Si   18   20   18   19   19
## 4    20161 201610173      11     1     Si   13   14   11   15    0
## 5    20161 201610918      18     1     Si   20   19   18   17   18
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0006*****"
##
## [1] "El curso original"
## 'data.frame':    55483 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 7 levels "FIN1","PAR1",..: 1 2 3 4 5 6 7 1 2 3 ...
##  $ eva_nota: num  8 9 10 14 6 16 5 12 10 17 ...
##  $ alu_cod : int  201512446 201512446 201512446 201512446 201512446 201512
446 201512446 201512451 201512451 201512451 ...
##  $ eva_fin : int  10 10 10 10 10 10 10 13 13 13 ...
##
## FIN1 PAR1 PRA1 PRA2 PRA3 PRA4 SUS1
## 7735 7734 8088 8088 8089 8089 7660
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    55483 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 6 7 1 2 3
##  $ eva_nota: num  8 9 10 14 6 16 5 12 10 17 ...
##  $ alu_cod : int  201512446 201512446 201512446 201512446 201512446 201512
446 201512446 201512451 201512451 201512451 ...
##  $ eva_fin : int  10 10 10 10 10 10 10 13 13 13 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151     FIN        8 201512446      10
## 2     20151     PAR        9 201512446      10
## 3     20151    PRA1       10 201512446      10
## 4     20151    PRA2       14 201512446      10
## 5     20151    PRA3        6 201512446      10
## 6     20151    PRA4       16 201512446      10
## 7     20151     SUS        5 201512446      10
## 8     20151     FIN       12 201512451      13
## 9     20151     PAR       10 201512451      13
## 10    20151    PRA1       17 201512451      13
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
```

```
ra"
## 'data.frame':    8089 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ alu_cod : chr  "201512446" "201512451" "201512427" "201512425" ...
##  $ eva_fin : int  10 13 13 13 11 18 13 15 16 10 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 3 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 1 1 1 1 1 1 1 1 2 ...
##  $ PRA1    : num  10 17 11 3 14 18 10 10 17 15 ...
##  $ FIN     : num  8 12 13 11 10 19 13 14 17 4 ...
##  $ PRA2    : num  14 15 13 14 13 15 20 19 12 9 ...
##  $ PRA4    : num  16 17 16 16 0 14 17 19 15 17 ...
##  $ PRA3    : num  6 17 18 16 14 12 12 19 14 17 ...
##  $ SUS     : num  5 88 88 88 14 88 88 88 88 10 ...
##  $ PAR     : num  9 10 12 16 6 19 12 14 15 3 ...
```

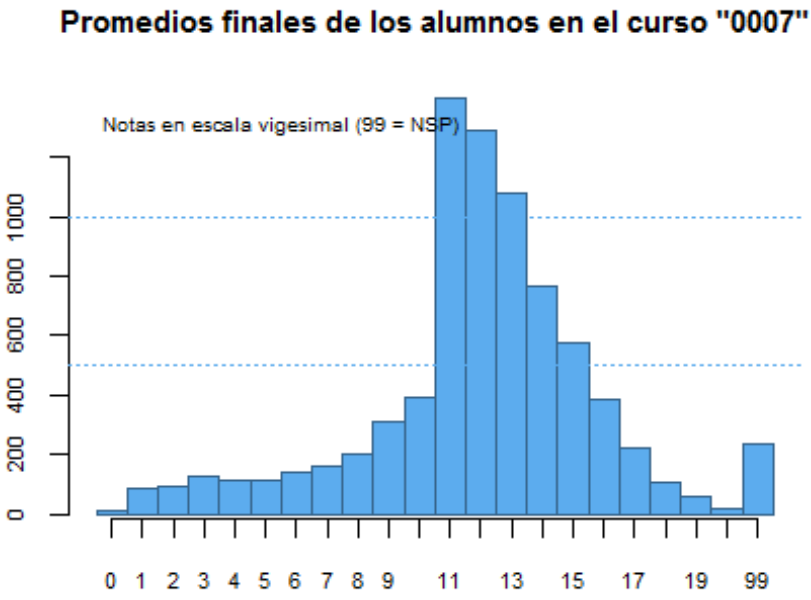





```
##   semestre   alu_cod eva_fin veces estado PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
## 1    20151 201512446      10     2      D   10   8   14   16    6   5   9
## 2    20151 201512451      13     1      A   17  12   15   17   17  88  10
## 3    20151 201512427      13     1      A   11  13   13   16   18  88  12
## 4    20151 201512425      13     1      A    3  11   14   16   16  88  16
## 5    20151 201512440      11     1      A   14  10   13    0   14  14   6
```

```
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    7907 obs. of  12 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ alu_cod : int  201512446 201512451 201512427 201512425 201512440 201512
447 201512428 201512471 201512426 201512438 ...
##  $ eva_fin : int  10 13 13 13 11 18 13 15 16 10 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 3 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 2 2 2 2 2 2 2 1 ...
##  $ PRA1    : num  10 17 11 3 14 18 10 10 17 15 ...
##  $ FIN     : num  8 12 13 11 10 19 13 14 17 4 ...
##  $ PRA2    : num  14 15 13 14 13 15 20 19 12 9 ...
##  $ PRA4    : num  16 17 16 16 0 14 17 19 15 17 ...
##  $ PRA3    : num  6 17 18 16 14 12 12 19 14 17 ...
##  $ SUS     : num  5 88 88 88 14 88 88 88 88 10 ...
##  $ PAR     : num  9 10 12 16 6 19 12 14 15 3 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
## 1    20151 201512446      10     2     No   10   8   14   16    6   5   9
## 2    20151 201512451      13     1     Si   17  12   15   17   17  88  10
## 3    20151 201512427      13     1     Si   11  13   13   16   18  88  12
## 4    20151 201512425      13     1     Si    3  11   14   16   16  88  16
## 5    20151 201512440      11     1     Si   14  10   13    0   14  14   6
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0007*****"
##
## [1] "El curso original"
## 'data.frame':    46204 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 11 levels "FIN1","PAR1",..: 1 2 7 8 9 11 1 2 7 8 ..
##  $ eva_nota: num  0 7 16 8 4 88 0 10 0 12 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201512432 201512
432 201512436 201512436 201512436 201512436 ...
##  $ eva_fin : int  5 5 5 5 5 5 5 5 5 5 ...
##
## FIN1 PAR1 PRA1 PRA2 PRA3 PRA4 PRT1 PRT2 PRT3 PRT4 SUS1
## 7458 7458 5669 5669 5669  139 2198 2198 2198  270 7278
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    46204 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 7 1 2 3 4
##  $ eva_nota: num  0 7 16 8 4 88 0 10 0 12 ...
##  $ alu_cod : int  201512432 201512432 201512432 201512432 201512432 201512
432 201512436 201512436 201512436 201512436 ...
##  $ eva_fin : int  5 5 5 5 5 5 5 5 5 5 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151     FIN        0 201512432       5
## 2     20151     PAR        7 201512432       5
## 3     20151    PRA1       16 201512432       5
## 4     20151    PRA2        8 201512432       5
## 5     20151    PRA3        4 201512432       5
## 6     20151     SUS       88 201512432       5
## 7     20151     FIN        0 201512436       5
## 8     20151     PAR       10 201512436       5
## 9     20151    PRA1        0 201512436       5
## 10    20151    PRA2       12 201512436       5
##
```

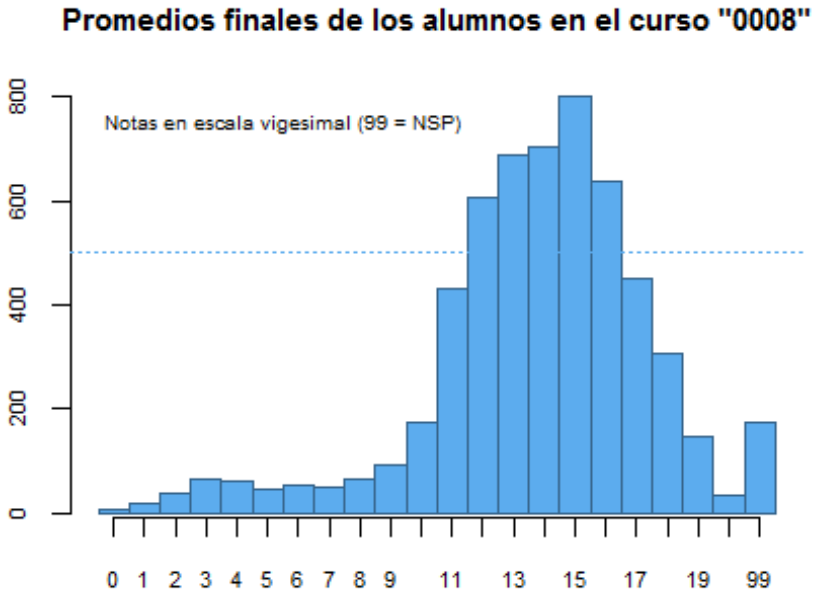
```
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    7867 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 2 2 2 2 2 2 .
##  $ alu_cod : chr  "201512432" "201512436" "201512475" "201512432" ...
##  $ eva_fin : int  5 5 12 99 11 16 18 11 9 16 ...
##  $ veces   : int  4 2 1 4 1 1 1 1 2 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 2 1 3 1 1 1 1 2 1 ...
##  $ PRA1    : num  16 0 12 0 15 18 16 10 10 17 ...
##  $ FIN     : num  0 0 17 0 14 18 18 11 6 14 ...
##  $ PRA2    : num  8 12 12 0 18 18 16 10 16 14 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  4 7 13 0 13 19 17 11 12 16 ...
##  $ SUS     : num  88 NA 88 88 88 88 88 NA 8 88 ...
##  $ PAR     : num  7 10 8 0 5 13 19 11 4 18 ...
```




```
##   semestre   alu_cod eva_fin veces estado PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
## 1    20151 201512432       5     4      D   16   0    8   NA    4  88   7
## 2    20151 201512436       5     2      D    0   0   12   NA    7  NA  10
## 3    20151 201512475      12     1      A   12  17   12   NA   13  88   8
## 4    20152 201512432      99     4      N    0   0    0   NA    0  88   0
```

```
## 5    20152 201512049      11     1      A   15  14   18   NA   13  88   5
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    7630 obs. of  12 variables:
##  $ semestre: int  20151 20151 20151 20152 20152 20152 20152 20152 20152...
##  $ alu_cod : int  201512432 201512436 201512475 201512049 201512175 201512
706 201512275 201512356 201511917 201511830 ...
##  $ eva_fin : int  5 5 12 11 16 18 11 9 16 14 ...
##  $ veces   : int  4 2 1 1 1 1 1 2 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 1 2 2 2 2 2 1 2 2 ...
##  $ PRA1    : num  16 0 12 15 18 16 10 10 17 13 ...
##  $ FIN     : num  0 0 17 14 18 18 11 6 14 14 ...
##  $ PRA2    : num  8 12 12 18 18 16 10 16 14 14 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  4 7 13 13 19 17 11 12 16 14 ...
##  $ SUS     : num  88 NA 88 88 88 88 NA 8 88 NA ...
##  $ PAR     : num  7 10 8 5 13 19 11 4 18 14 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
## 1    20151 201512432       5     4     No   16   0    8   NA    4  88   7
## 2    20151 201512436       5     2     No    0   0   12   NA    7  NA  10
## 3    20151 201512475      12     1     Si   12  17   12   NA   13  88   8
## 4    20152 201512049      11     1     Si   15  14   18   NA   13  88   5
## 5    20152 201512175      16     1     Si   18  18   18   NA   19  88  13
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0008*****"
##
## [1] "El curso original"
## 'data.frame':    22572 obs. of  5 variables:
##  $ semestre: int  20162 20162 20162 20162 20162 20162 20162 20162 20162...
##  $ eva_cod : Factor w/ 12 levels "PRA1","PRA2",..: 9 10 11 12 5 6 7 8 5 6
##  $ eva_nota: num  19 19 16 17 13 10 13 12 15 16 ...
##  $ alu_cod : int  201520493 201520493 201520493 201520493 201610280 201610
280 201610280 201610280 201610869 201610869 ...
##  $ eva_fin : int  18 18 18 18 12 12 12 12 16 16 ...
##
## PRA1 PRA2 PRA3 PRA4 PTL1 PTL2 PTL3 PTL4 TLR1 TLR2 TLR3 TLR4
## 4519 4519 4519 4519 1082 1082 1082 1082   42   42   42   42
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    22572 obs. of  5 variables:
##  $ semestre: int  20162 20162 20162 20162 20162 20162 20162 20162 20162...
##  $ eva_new : Factor w/ 4 levels "PRA1","PRA2",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ eva_nota: num  19 19 16 17 13 10 13 12 15 16 ...
##  $ alu_cod : int  201520493 201520493 201520493 201520493 201610280 201610
280 201610280 201610280 201610869 201610869 ...
##  $ eva_fin : int  18 18 18 18 12 12 12 12 16 16 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20162    PRA1       19 201520493      18
## 2     20162    PRA2       19 201520493      18
## 3     20162    PRA3       16 201520493      18
## 4     20162    PRA4       17 201520493      18
## 5     20162    PRA1       13 201610280      12
## 6     20162    PRA2       10 201610280      12
## 7     20162    PRA3       13 201610280      12
## 8     20162    PRA4       12 201610280      12
## 9     20162    PRA1       15 201610869      16
## 10    20162    PRA2       16 201610869      16
```

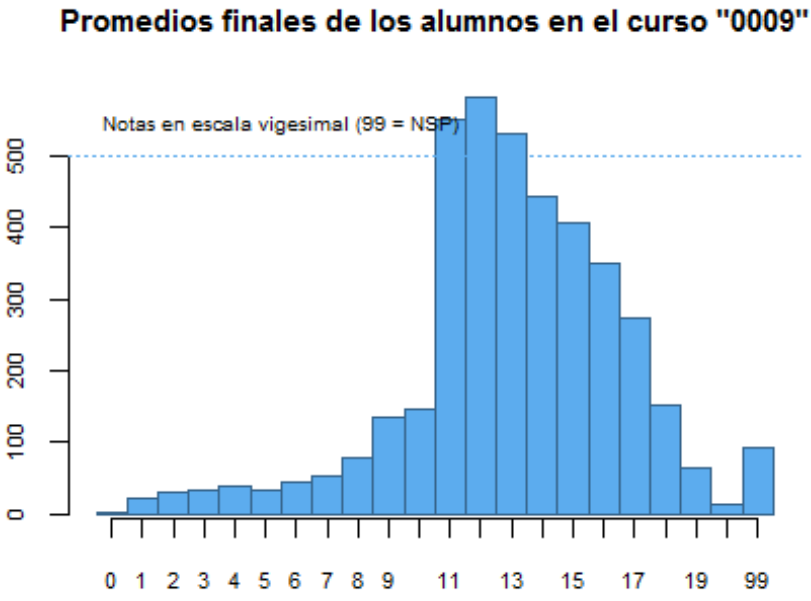
```
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    5643 obs. of  9 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 ..
##  $ alu_cod : chr  "201520493" "201610280" "201610869" "201610586" ...
##  $ eva_fin : int  18 12 16 15 13 16 9 15 16 16 ...
##  $ veces   : int  1 1 1 1 1 1 2 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 2 1 1 1 ...
##  $ PRA1    : num  19 13 15 12 13 14 11 15 15 15 ...
##  $ PRA2    : num  19 10 16 17 8 18 8 18 19 14 ...
##  $ PRA4    : num  17 12 15 17 16 16 11 13 16 16 ...
##  $ PRA3    : num  16 13 17 14 14 15 6 13 15 18 ...
```

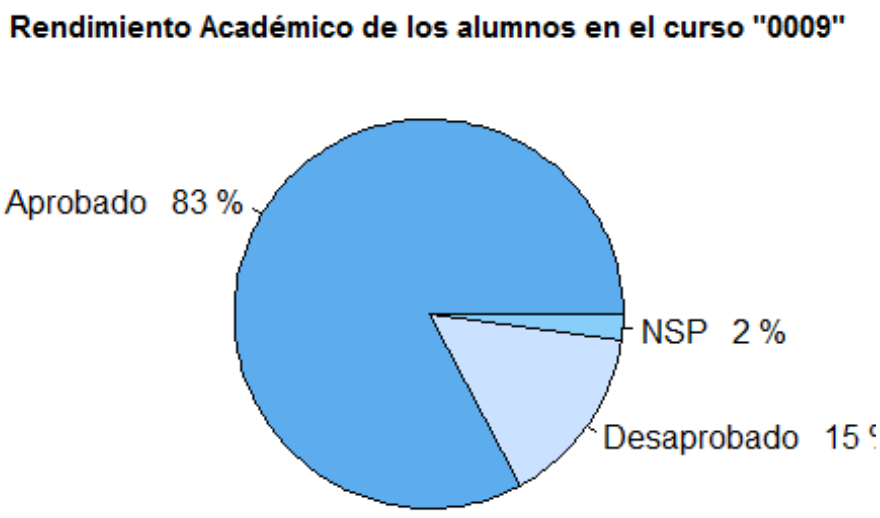



```
##   semestre   alu_cod eva_fin veces estado PRA1 PRA2 PRA4 PRA3
## 1    20162 201520493      18     1      A   19   19   17   16
## 2    20162 201610280      12     1      A   13   10   12   13
## 3    20162 201610869      16     1      A   15   16   15   17
## 4    20162 201610586      15     1      A   12   17   17   14
## 5    20162 201611022      13     1      A   13    8   16   14
##
```

```
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    5468 obs. of  9 variables:
##  $ semestre: int  20162 20162 20162 20162 20162 20162 20162 20162 20162...
##  $ alu_cod : int  201520493 201610280 201610869 201610586 201611022 201610
308 201610173 201610918 201610266 201610399 ...
##  $ eva_fin : int  18 12 16 15 13 16 9 15 16 16 ...
##  $ veces   : int  1 1 1 1 1 1 2 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 1 2 2 2 ...
##  $ PRA1    : num  19 13 15 12 13 14 11 15 15 15 ...
##  $ PRA2    : num  19 10 16 17 8 18 8 18 19 14 ...
##  $ PRA4    : num  17 12 15 17 16 16 11 13 16 16 ...
##  $ PRA3    : num  16 13 17 14 14 15 6 13 15 18 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 PRA2 PRA4 PRA3
## 1    20162 201520493      18     1     Si   19   19   17   16
## 2    20162 201610280      12     1     Si   13   10   12   13
## 3    20162 201610869      16     1     Si   15   16   15   17
## 4    20162 201610586      15     1     Si   12   17   17   14
## 5    20162 201611022      13     1     Si   13    8   16   14
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0009*****"
##
## [1] "El curso original"
## 'data.frame':    20215 obs. of  5 variables:
##  $ semestre: int  20162 20162 20162 20162 20162 20162 20162 20162 20162...
##  $ eva_cod : Factor w/ 6 levels "PRA1","PRA2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ eva_nota: num  19 19 18 19 20 20 16 18 19 20 ...
##  $ alu_cod : int  201611022 201611022 201611022 201611022 201611022 201610
308 201610308 201610308 201610308 201610308 ...
##  $ eva_fin : int  19 19 19 19 19 19 19 19 19 19 ...
##
## PRA1 PRA2 PRA3 PRA4 PRA5 TRP1
## 4072 4072 4072 4072 3916   11
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    20215 obs. of  5 variables:
##  $ semestre: int  20162 20162 20162 20162 20162 20162 20162 20162 20162...
##  $ eva_new : Factor w/ 5 levels "PRA1","PRA2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ eva_nota: num  19 19 18 19 20 20 16 18 19 20 ...
##  $ alu_cod : int  201611022 201611022 201611022 201611022 201611022 201610
308 201610308 201610308 201610308 201610308 ...
##  $ eva_fin : int  19 19 19 19 19 19 19 19 19 19 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20162    PRA1       19 201611022      19
## 2     20162    PRA2       19 201611022      19
## 3     20162    PRA3       18 201611022      19
## 4     20162    PRA4       19 201611022      19
## 5     20162    PRA5       20 201611022      19
## 6     20162    PRA1       20 201610308      19
## 7     20162    PRA2       16 201610308      19
## 8     20162    PRA3       18 201610308      19
## 9     20162    PRA4       19 201610308      19
## 10    20162    PRA5       20 201610308      19
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    4072 obs. of  10 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 1 ..
```

```
##  $ alu_cod : chr  "201611022" "201610308" "201610173" "201610918" ...
##  $ eva_fin : int  19 19 12 18 9 14 12 13 11 12 ...
##  $ veces   : int  1 1 1 1 2 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 2 1 1 1 1 1 ...
##  $ PRA5    : num  20 20 12 19 9 14 8 16 10 10 ...
##  $ PRA1    : num  19 20 11 17 12 14 14 8 8 13 ...
##  $ PRA2    : num  19 16 13 19 0 9 9 12 7 11 ...
##  $ PRA4    : num  19 19 12 18 14 17 14 13 15 16 ...
##  $ PRA3    : num  18 18 14 19 12 14 14 14 13 12 ...
```

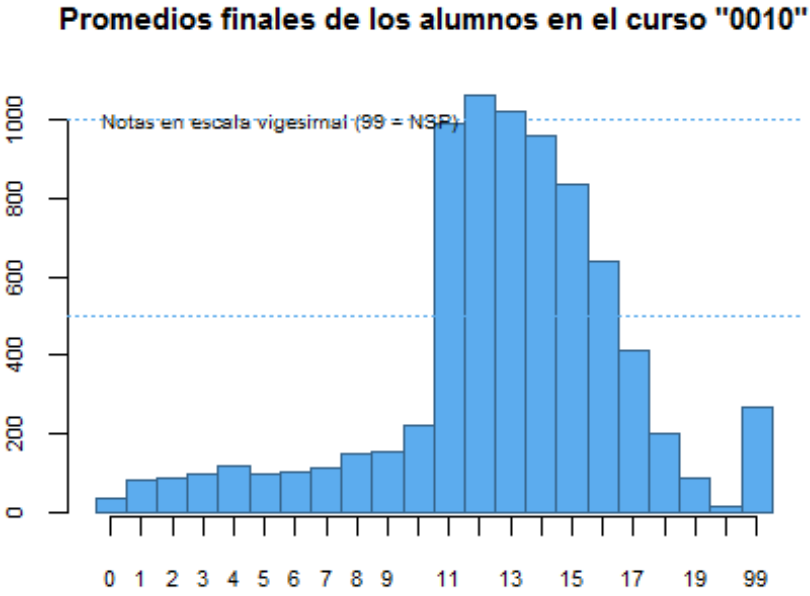


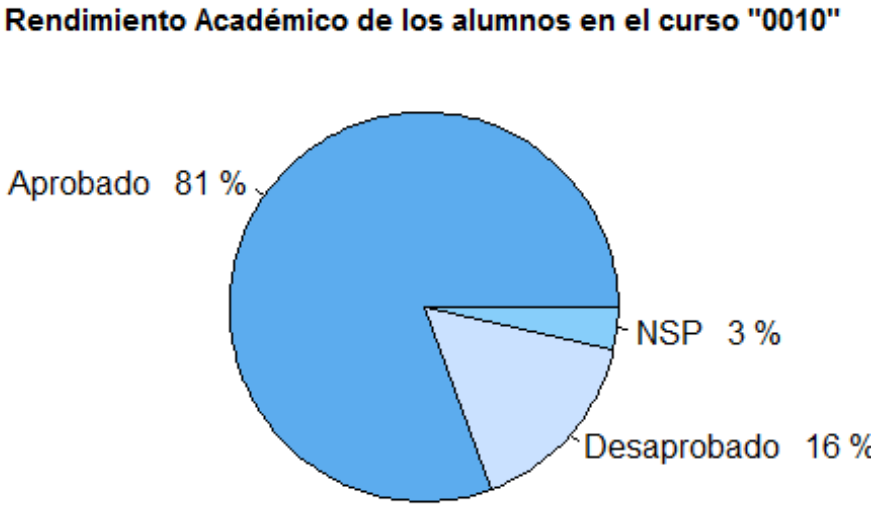



```
##   semestre   alu_cod eva_fin veces estado PRA5 PRA1 PRA2 PRA4 PRA3
## 1    20162 201611022      19     1      A   20   19   19   19   18
## 2    20162 201610308      19     1      A   20   20   16   19   18
## 3    20162 201610173      12     1      A   12   11   13   12   14
## 4    20162 201610918      18     1      A   19   17   19   18   19
## 5    20162 201610399       9     2      D    9   12    0   14   12
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    3981 obs. of  10 variables:
```

```
##  $ semestre: int  20162 20162 20162 20162 20162 20162 20162 20162 20162 ..
##  $ alu_cod : int  201611022 201610308 201610173 201610918 201610399 201611
669 201611246 201611260 201611213 201611718 ...
##  $ eva_fin : int  19 19 12 18 9 14 12 13 11 12 ...
##  $ veces   : int  1 1 1 1 2 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 1 2 2 2 2 2 ...
##  $ PRA5    : num  20 20 12 19 9 14 8 16 10 10 ...
##  $ PRA1    : num  19 20 11 17 12 14 14 8 8 13 ...
##  $ PRA2    : num  19 16 13 19 0 9 9 12 7 11 ...
##  $ PRA4    : num  19 19 12 18 14 17 14 13 15 16 ...
##  $ PRA3    : num  18 18 14 19 12 14 14 14 13 12 ...
##   semestre   alu_cod eva_fin veces aprobo PRA5 PRA1 PRA2 PRA4 PRA3
## 1    20162 201611022      19     1     Si   20   19   19   19   18
## 2    20162 201610308      19     1     Si   20   20   16   19   18
## 3    20162 201610173      12     1     Si   12   11   13   12   14
## 4    20162 201610918      18     1     Si   19   17   19   18   19
## 5    20162 201610399       9     2     No    9   12    0   14   12
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0010*****"
##
## [1] "El curso original"
## 'data.frame':    45610 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 7 levels "FIN1","PAR1",..: 1 2 3 4 5 7 1 2 3 4 ...
##  $ eva_nota: num  18 18 13 19 18 88 13 8 8 14 ...
##  $ alu_cod : int  201512444 201512444 201512444 201512444 201512444 201512
444 201512446 201512446 201512446 201512446 ...
##  $ eva_fin : int  18 18 18 18 18 18 11 11 11 11 ...
##
## FIN1 PAR1 PRA1 PRA2 PRA3 PRA4 SUS1
## 7466 7466 7725 7692 7692  259 7310
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    45610 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 7 1 2 3 4
##  $ eva_nota: num  18 18 13 19 18 88 13 8 8 14 ...
##  $ alu_cod : int  201512444 201512444 201512444 201512444 201512444 201512
444 201512446 201512446 201512446 201512446 ...
##  $ eva_fin : int  18 18 18 18 18 18 11 11 11 11 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151     FIN       18 201512444      18
## 2     20151     PAR       18 201512444      18
## 3     20151    PRA1       13 201512444      18
## 4     20151    PRA2       19 201512444      18
## 5     20151    PRA3       18 201512444      18
## 6     20151     SUS       88 201512444      18
## 7     20151     FIN       13 201512446      11
## 8     20151     PAR        8 201512446      11
## 9     20151    PRA1        8 201512446      11
## 10    20151    PRA2       14 201512446      11
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    7725 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ alu_cod : chr  "201512444" "201512446" "201512451" "201512473" ...
##  $ eva_fin : int  18 11 13 15 13 11 11 18 15 15 ...
```

```
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  13 8 15 17 9 11 13 15 13 15 ...
##  $ FIN     : num  18 13 9 15 12 11 9 20 16 14 ...
##  $ PRA2    : num  19 14 13 15 11 14 12 18 17 12 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  18 9 17 14 15 13 14 19 11 14 ...
##  $ SUS     : num  88 11 88 88 88 88 88 88 88 88 ...
##  $ PAR     : num  18 8 15 14 15 8 11 17 15 16 ...
```

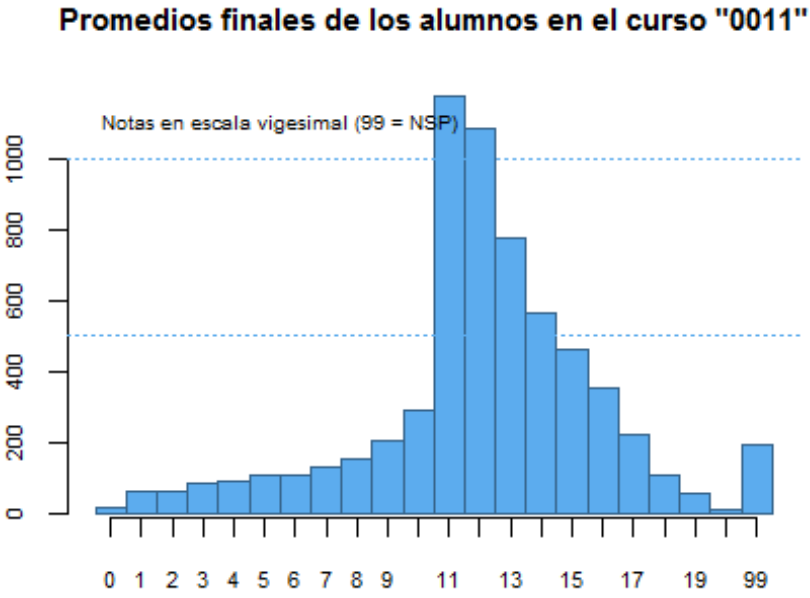


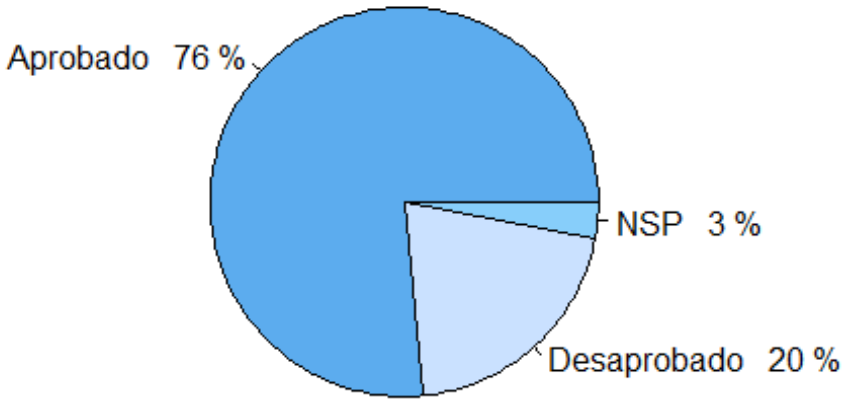



```
##   semestre   alu_cod eva_fin veces estado PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
## 1    20151 201512444      18     1      A   13  18   19   NA   18  88  18
## 2    20151 201512446      11     1      A    8  13   14   NA    9  11   8
## 3    20151 201512451      13     1      A   15   9   13   NA   17  88  15
## 4    20151 201512473      15     1      A   17  15   15   NA   14  88  14
## 5    20151 201512420      13     1      A    9  12   11   NA   15  88  15
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    7458 obs. of  12 variables:
```

```
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151 ..
##  $ alu_cod : int  201512444 201512446 201512451 201512473 201512420 201512
427 201512440 201512428 201512441 201512426 ...
##  $ eva_fin : int  18 11 13 15 13 11 11 18 15 15 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  13 8 15 17 9 11 13 15 13 15 ...
##  $ FIN     : num  18 13 9 15 12 11 9 20 16 14 ...
##  $ PRA2    : num  19 14 13 15 11 14 12 18 17 12 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  18 9 17 14 15 13 14 19 11 14 ...
##  $ SUS     : num  88 11 88 88 88 88 88 88 88 88 ...
##  $ PAR     : num  18 8 15 14 15 8 11 17 15 16 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
## 1    20151 201512444      18     1     Si   13  18   19   NA   18  88  18
## 2    20151 201512446      11     1     Si    8  13   14   NA    9  11   8
## 3    20151 201512451      13     1     Si   15   9   13   NA   17  88  15
## 4    20151 201512473      15     1     Si   17  15   15   NA   14  88  14
## 5    20151 201512420      13     1     Si    9  12   11   NA   15  88  15
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0011*****"
##
## [1] "El curso original"
## 'data.frame':    49665 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 13 levels "FIN1","NPA1",..: 1 2 3 4 5 6 7 12 13 1 .
##  $ eva_nota: num  15 13 16 14 17 16 14 88 15 14 ...
##  $ alu_cod : int  201512444 201512444 201512444 201512444 201512444 201512
444 201512444 201512444 201512444 201512430 ...
##  $ eva_fin : int  15 15 15 15 15 15 15 15 15 12 ...
##
## FIN1 NPA1 PAR1 PRA1 PRA2 PRA3 PRA4 PRT1 PRT2 PRT3 PRT4 SUS1 TRA1
## 6162  319 6162 6206 6206 6166 6166   84   84   84   84 5820 6122
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    49665 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151
##  $ eva_new : Factor w/ 9 levels "FIN","NPA","PAR",..: 1 2 3 4 5 6 7 8 9 1
##  $ eva_nota: num  15 13 16 14 17 16 14 88 15 14 ...
##  $ alu_cod : int  201512444 201512444 201512444 201512444 201512444 201512
444 201512444 201512444 201512444 201512430 ...
##  $ eva_fin : int  15 15 15 15 15 15 15 15 15 12 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151     FIN       15 201512444      15
## 2     20151     NPA       13 201512444      15
## 3     20151     PAR       16 201512444      15
## 4     20151    PRA1       14 201512444      15
## 5     20151    PRA2       17 201512444      15
## 6     20151    PRA3       16 201512444      15
## 7     20151    PRA4       14 201512444      15
## 8     20151     SUS       88 201512444      15
## 9     20151     TRA       15 201512444      15
## 10    20151     FIN       14 201512430      12
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    6290 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
```

```
##  $ alu_cod : chr  "201512444" "201512430" "201512446" "201512451" ...
##  $ eva_fin : int  15 12 10 12 14 12 14 13 14 16 ...
##  $ veces   : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 2 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  14 10 4 8 11 7 17 10 13 16 ...
##  $ FIN     : num  15 14 7 14 11 14 10 13 15 11 ...
##  $ PRA2    : num  17 10 3 9 16 12 10 14 14 14 ...
##  $ PRA4    : num  14 13 8 10 14 11 12 14 14 20 ...
##  $ PRA3    : num  16 12 8 9 12 11 18 14 15 14 ...
##  $ NPA     : num  13 14 14 18 17 13 20 15 14 18 ...
##  $ TRA     : num  15 18 12 7 17 12 20 15 15 20 ...
##  $ SUS     : num  88 88 12 88 88 88 88 88 88 88 ...
##  $ PAR     : num  16 6 9 18 14 12 11 12 14 16 ...
```

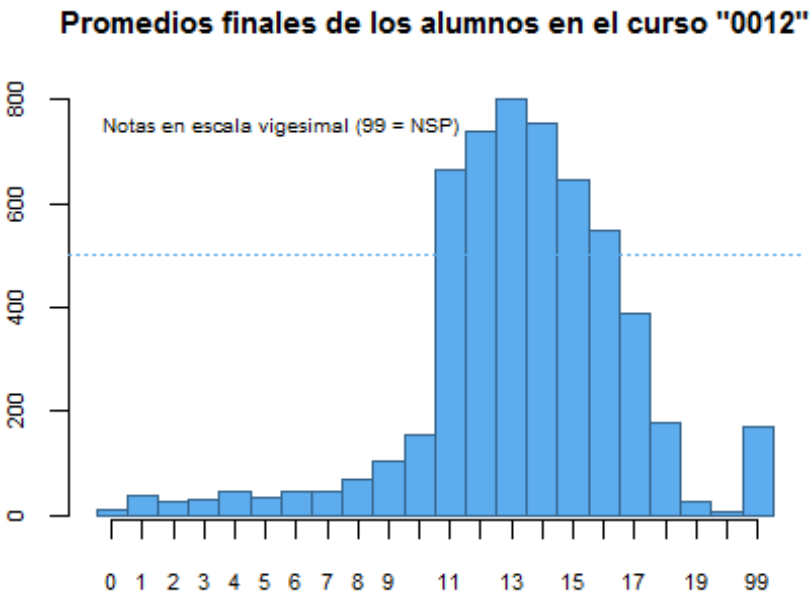


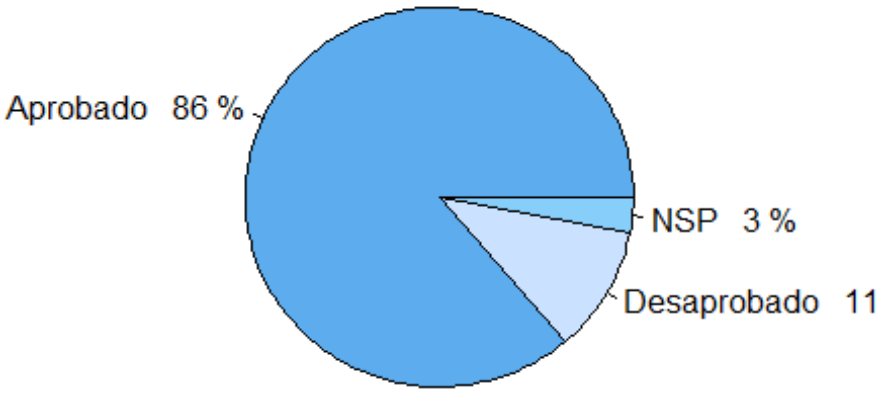



```
##   semestre   alu_cod eva_fin veces estado PRA1 FIN PRA2 PRA4 PRA3 NPA TRA
## 1    20151 201512444      15     1      A   14  15   17   14   16  13  15
## 2    20151 201512430      12     1      A   10  14   10   13   12  14  18
## 3    20151 201512446      10     2      D    4   7    3    8    8  14  12
## 4    20151 201512451      12     1      A    8  14    9   10    9  18   7
## 5    20151 201512473      14     1      A   11  11   16   14   12  17  17
##   SUS PAR
## 1  88  16
```

```
## 2  88   6
## 3  12   9
## 4  88  18
## 5  88  14
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    6100 obs. of  14 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ alu_cod : int  201512444 201512430 201512446 201512451 201512473 201512
420 201512440 201512447 201512441 201512471 ...
##  $ eva_fin : int  15 12 10 12 14 12 14 13 14 16 ...
##  $ veces   : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 1 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  14 10 4 8 11 7 17 10 13 16 ...
##  $ FIN     : num  15 14 7 14 11 14 10 13 15 11 ...
##  $ PRA2    : num  17 10 3 9 16 12 10 14 14 14 ...
##  $ PRA4    : num  14 13 8 10 14 11 12 14 14 20 ...
##  $ PRA3    : num  16 12 8 9 12 11 18 14 15 14 ...
##  $ NPA     : num  13 14 14 18 17 13 20 15 14 18 ...
##  $ TRA     : num  15 18 12 7 17 12 20 15 15 20 ...
##  $ SUS     : num  88 88 12 88 88 88 88 88 88 88 ...
##  $ PAR     : num  16 6 9 18 14 12 11 12 14 16 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 FIN PRA2 PRA4 PRA3 NPA TRA
## 1    20151 201512444      15     1     Si   14  15   17   14   16  13  15
## 2    20151 201512430      12     1     Si   10  14   10   13   12  14  18
## 3    20151 201512446      10     2     No    4   7    3    8    8  14  12
## 4    20151 201512451      12     1     Si    8  14    9   10    9  18   7
## 5    20151 201512473      14     1     Si   11  11   16   14   12  17  17
##   SUS PAR
## 1  88  16
## 2  88   6
## 3  12   9
## 4  88  18
## 5  88  14
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0012*****"
##
## [1] "El curso original"
## 'data.frame':    38618 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 9 levels "FIN1","PAR1",..: 1 2 3 4 5 6 7 8 9 1 ...
##  $ eva_nota: num  16 16 15 13 15 19 19 88 16 12 ...
##  $ alu_cod : int  201512444 201512444 201512444 201512444 201512444 201512
444 201512444 201512444 201512444 201512430 ...
##  $ eva_fin : int  16 16 16 16 16 16 16 16 16 14 ...
##
## FIN1 PAR1 PRA1 PRA2 PRA3 PRA4 PRA5 SUS1 TMO1
## 5402 5402 5507 5461 5461 5461  849 4975  100
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    38618 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 9 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 6 7 9 8 1
##  $ eva_nota: num  16 16 15 13 15 19 19 88 16 12 ...
##  $ alu_cod : int  201512444 201512444 201512444 201512444 201512444 201512
444 201512444 201512444 201512444 201512430 ...
##  $ eva_fin : int  16 16 16 16 16 16 16 16 16 14 ...
```

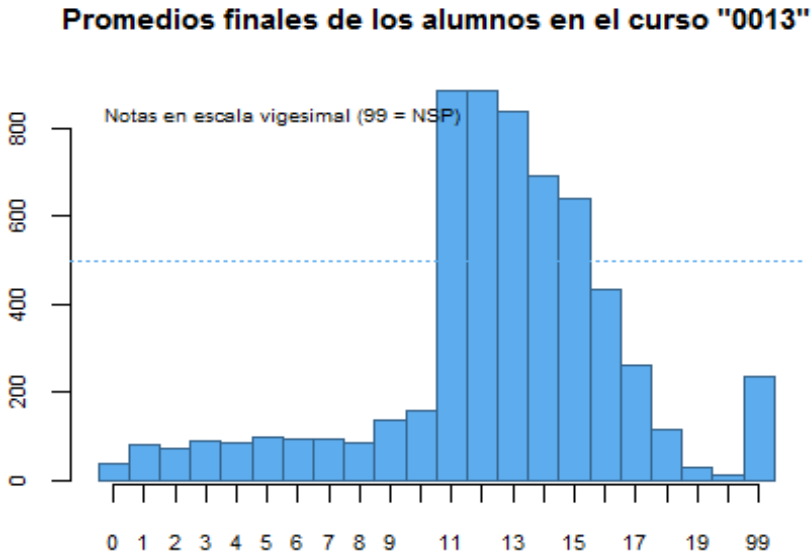
```
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151     FIN       16 201512444      16
## 2     20151     PAR       16 201512444      16
## 3     20151    PRA1       15 201512444      16
## 4     20151    PRA2       13 201512444      16
## 5     20151    PRA3       15 201512444      16
## 6     20151    PRA4       19 201512444      16
## 7     20151    PRA5       19 201512444      16
## 8     20151     SUS       88 201512444      16
## 9     20151    PRA6       16 201512444      16
## 10    20151     FIN       12 201512430      14
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    5507 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ alu_cod : chr  "201512444" "201512430" "201512446" "201512425" ...
##  $ eva_fin : int  16 14 11 14 12 16 11 11 16 11 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  15 7 12 15 10 15 12 11 15 6 ...
##  $ FIN     : num  16 12 11 12 11 18 16 12 19 0 ...
##  $ PRA2    : num  13 13 14 11 10 13 14 11 15 12 ...
##  $ PRA4    : num  19 16 17 15 15 19 14 12 15 0 ...
##  $ PRA3    : num  15 16 11 17 10 15 0 8 15 14 ...
##  $ PRA5    : num  19 17 NA 17 NA 19 NA 0 NA NA ...
##  $ SUS     : num  88 88 88 88 NA 88 9 88 NA 7 ...
##  $ PRA6    : num  16 14 17 14 14 16 0 14 16 15 ...
##  $ PAR     : num  16 15 7 14 11 14 9 8 14 12 ...
```

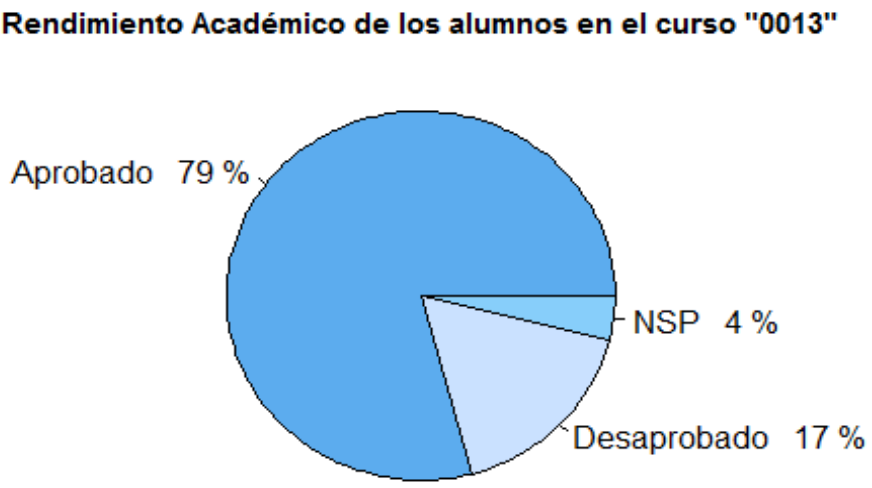





```
##   semestre   alu_cod eva_fin veces estado PRA1 FIN PRA2 PRA4 PRA3 PRA5 SUS
## 1    20151 201512444      16     1      A   15  16   13   19   15   19  88
## 2    20151 201512430      14     1      A    7  12   13   16   16   17  88
## 3    20151 201512446      11     1      A   12  11   14   17   11   NA  88
## 4    20151 201512425      14     1      A   15  12   11   15   17   17  88
## 5    20151 201512440      12     1      A   10  11   10   15   10   NA  NA
##   PRA6 PAR
## 1   16  16
## 2   14  15
## 3   17   7
## 4   14  14
## 5   14  11
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    5338 obs. of  14 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ alu_cod : int  201512444 201512430 201512446 201512425 201512440 201512
441 201512437 201512438 201512474 201512448 ...
##  $ eva_fin : int  16 14 11 14 12 16 11 11 16 11 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  15 7 12 15 10 15 12 11 15 6 ...
##  $ FIN     : num  16 12 11 12 11 18 16 12 19 0 ...
##  $ PRA2    : num  13 13 14 11 10 13 14 11 15 12 ...
##  $ PRA4    : num  19 16 17 15 15 19 14 12 15 0 ...
##  $ PRA3    : num  15 16 11 17 10 15 0 8 15 14 ...
##  $ PRA5    : num  19 17 NA 17 NA 19 NA 0 NA NA ...
##  $ SUS     : num  88 88 88 88 NA 88 9 88 NA 7 ...
##  $ PRA6    : num  16 14 17 14 14 16 0 14 16 15 ...
##  $ PAR     : num  16 15 7 14 11 14 9 8 14 12 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 FIN PRA2 PRA4 PRA3 PRA5 SUS
## 1    20151 201512444      16     1     Si   15  16   13   19   15   19  88
## 2    20151 201512430      14     1     Si    7  12   13   16   16   17  88
## 3    20151 201512446      11     1     Si   12  11   14   17   11   NA  88
## 4    20151 201512425      14     1     Si   15  12   11   15   17   17  88
## 5    20151 201512440      12     1     Si   10  11   10   15   10   NA  NA
##   PRA6 PAR
## 1   16  16
## 2   14  15
## 3   17   7
## 4   14  14
## 5   14  11
```

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0013*****"
##
## [1] "El curso original"
## 'data.frame':    35668 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_cod : Factor w/ 11 levels "FIN1","PAR1",..: 1 2 3 4 5 11 1 2 3 4 ..
##  $ eva_nota: num  11 7 11 13 12 18 12 13 13 13 ...
##  $ alu_cod : int  201512430 201512430 201512430 201512430 201512430 201512
430 201512425 201512425 201512425 201512425 ...
##  $ eva_fin : int  14 14 14 14 14 14 12 12 12 12 ...
##
## FIN1 PAR1 PRA1 PRA2 PRA3 PRA4 PRT1 PRT2 PRT3 PRT4 SUS1
## 5873 5873 5938 5939 5939   66   97   97   97   97 5652
##
## [1] "El curso con el tipo de evaluación estandarizado"
## 'data.frame':    35668 obs. of  5 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20151 20151...
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 7 1 2 3 4
##  $ eva_nota: num  11 7 11 13 12 18 12 13 13 13 ...
##  $ alu_cod : int  201512430 201512430 201512430 201512430 201512430 201512
430 201512425 201512425 201512425 201512425 ...
##  $ eva_fin : int  14 14 14 14 14 14 12 12 12 12 ...
##    semestre eva_new eva_nota   alu_cod eva_fin
## 1     20151     FIN       11 201512430      14
## 2     20151     PAR        7 201512430      14
## 3     20151    PRA1       11 201512430      14
## 4     20151    PRA2       13 201512430      14
## 5     20151    PRA3       12 201512430      14
## 6     20151     SUS       18 201512430      14
## 7     20151     FIN       12 201512425      12
## 8     20151     PAR       13 201512425      12
## 9     20151    PRA1       13 201512425      12
## 10    20151    PRA2       13 201512425      12
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra"
## 'data.frame':    6036 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 2 2 2 .
##  $ alu_cod : chr  "201512430" "201512425" "201512447" "201512471" ...
##  $ eva_fin : int  14 12 14 14 14 14 18 16 13 12 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  11 13 18 8 10 17 17 15 10 11 ...
##  $ FIN     : num  11 12 12 13 16 11 18 17 14 12 ...
##  $ PRA2    : num  13 13 12 20 15 17 20 17 15 11 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  12 11 15 20 15 16 18 18 13 14 ...
##  $ SUS     : num  18 88 14 88 88 88 88 88 88 NA ...
##  $ PAR     : num  7 13 0 14 14 13 19 14 11 11 ...
```






```
##   semestre   alu_cod eva_fin veces estado PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
## 1    20151 201512430      14     1      A   11  11   13   NA   12  18   7
## 2    20151 201512425      12     1      A   13  12   13   NA   11  88  13
## 3    20151 201512447      14     1      A   18  12   12   NA   15  14   0
## 4    20151 201512471      14     1      A    8  13   20   NA   20  88  14
## 5    20151 201512462      14     1      A   10  16   15   NA   15  88  14
##
## [1] "El curso con el tipo de evaluación estandarizado en la nueva estructu
ra y alumnos aprobados"
## 'data.frame':    5801 obs. of  12 variables:
##  $ semestre: int  20151 20151 20151 20151 20151 20151 20151 20152 20152...
##  $ alu_cod : int  201512430 201512425 201512447 201512471 201512462 201512
463 201512466 201512444 201512446 201512420 ...
##  $ eva_fin : int  14 12 14 14 14 14 18 16 13 12 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  11 13 18 8 10 17 17 15 10 11 ...
##  $ FIN     : num  11 12 12 13 16 11 18 17 14 12 ...
##  $ PRA2    : num  13 13 12 20 15 17 20 17 15 11 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  12 11 15 20 15 16 18 18 13 14 ...
##  $ SUS     : num  18 88 14 88 88 88 88 88 88 NA ...
##  $ PAR     : num  7 13 0 14 14 13 19 14 11 11 ...
##   semestre   alu_cod eva_fin veces aprobo PRA1 FIN PRA2 PRA4 PRA3 SUS PAR
```

```
## 1    20151 201512430      14     1     Si   11  11   13   NA   12  18   7
## 2    20151 201512425      12     1     Si   13  12   13   NA   11  88  13
## 3    20151 201512447      14     1     Si   18  12   12   NA   15  14   0
## 4    20151 201512471      14     1     Si    8  13   20   NA   20  88  14
## 5    20151 201512462      14     1     Si   10  16   15   NA   15  88  14
## [1] "***** Fin de Ejecución *****"
```

Anexo 12: Resultados de la preparación de cada curso del Programa de Estudios Básicos (PEB).

Procederemos a revisar los Archivos de Datos para obtener una mejor visualización de su contenido.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
```

Para la preparación del Curso con código: "0001" sirvase revisar el Anexo 07.

---

1. Preparación de la estructura interna de los Cursos

0002: "Taller de Método de Estudio Universitario".
0003: "Taller de Comunicación Oral y Escrita I".
0004: "Matemática".
0005: "Inglés I".
0006: "Psicología General".
0007: "Lógica y Filosofía".
0008: "Taller de Comunicación Oral y Escrita II".
0009: "Inglés II".
0010: "Formación Histórica del Perú".
0011: "Recursos Naturales y Medio Ambiente".
0012: "Realidad Nacional".
0013: "Historia de la Civilización".

```
## [1] "***** Inicio - Curso: 0002*****"
##
## 'data.frame':    59917 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 6 levels "PRA1","PRA2",..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ eva_nota: num  15 14 11 1 5 10 15 15 13 14 ...
##  $ alu_cod : chr  "201512432" "201512432" "201512432" "201512432" ...
##  $ eva_fin : int  9 9 9 9 9 9 15 15 15 15 ...
##     semestre     eva_new         eva_nota       alu_cod
##  20181  : 9636   PRA1:10032   Min.   : 0.00   Length:59917
##  20161  : 9306   PRA2:10039   1st Qu.:11.00   Class :character
##  20171  : 9084   PRA3:10039   Median :13.00   Mode  :character
##  20151  : 7962   PRA4:10005   Mean   :12.13
##  20172  : 6156   PRA5: 9918   3rd Qu.:16.00
##  20152  : 6138   PRA6: 9884   Max.   :20.00
##  (Other):11635
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :14.00
##  Mean   :15.37
##  3rd Qu.:15.00
##  Max.   :99.00
##
## 'data.frame':    10039 obs. of  11 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
```

```
..
##  $ alu_cod : chr  "201512432" "201510876" "201512136" "201512049" ...
##  $ eva_fin : int  9 15 99 14 16 12 14 12 15 10 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 2 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 1 3 1 1 1 1 1 1 2 ...
##  $ PRA1    : num  15 15 0 9 15 15 14 12 17 11 ...
##  $ PRA2    : num  14 15 0 10 16 11 15 15 15 5 ...
##  $ PRA4    : num  1 14 0 17 10 9 12 11 17 11 ...
##  $ PRA5    : num  5 16 0 14 18 11 13 11 14 8 ...
##  $ PRA3    : num  11 13 0 16 17 15 15 14 11 13 ...
##  $ PRA6    : num  10 14 0 15 17 12 14 12 16 13 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1606   Length:10039       Min.   : 0.00   Min.   :1.000
##  20161  :1551   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20171  :1514   Mode  :character   Median :14.00   Median :1.000
##  20151  :1327                      Mean   :15.42   Mean   :1.317
##  20172  :1026                      3rd Qu.:15.00   3rd Qu.:1.000
##  20152  :1023                      Max.   :99.00   Max.   :8.000
##  (Other):1992
##  estado        PRA1            PRA2            PRA4            PRA5
##  A:7981   Min.   : 0.0   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D:1729   1st Qu.:10.0   1st Qu.:10.00   1st Qu.:10.00   1st Qu.:11.00
##  N: 329   Median :13.0   Median :13.00   Median :13.00   Median :14.00
##           Mean   :12.2   Mean   :12.15   Mean   :12.15   Mean   :12.06
##           3rd Qu.:15.0   3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:16.00
##           Max.   :20.0   Max.   :20.00   Max.   :20.00   Max.   :20.00
##           NA's   :7                      NA's   :34      NA's   :121
##       PRA3            PRA6
##  Min.   : 0.00   Min.   : 0.00
##  1st Qu.:10.00   1st Qu.:11.00
##  Median :13.00   Median :14.00
##  Mean   :12.21   Mean   :12.04
##  3rd Qu.:16.00   3rd Qu.:16.00
##  Max.   :20.00   Max.   :20.00
##                  NA's   :155
##
## 'data.frame':    9710 obs. of  11 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512432" "201510876" "201512049" "201512175" ...
##  $ eva_fin : int  9 15 14 16 12 14 12 15 10 18 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 2 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 2 2 2 2 2 2 2 1 2 ...
##  $ PRA1    : num  15 15 9 15 15 14 12 17 11 18 ...
##  $ PRA2    : num  14 15 10 16 11 15 15 15 5 18 ...
##  $ PRA4    : num  1 14 17 10 9 12 11 17 11 18 ...
##  $ PRA5    : num  5 16 14 18 11 13 11 14 8 18 ...
##  $ PRA3    : num  11 13 16 17 15 15 14 11 13 18 ...
##  $ PRA6    : num  10 14 15 17 12 14 12 16 13 18 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1543   Length:9710        Min.   : 0.00   Min.   :1.000
##  20161  :1499   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20171  :1470   Mode  :character   Median :13.00   Median :1.000
##  20151  :1294                      Mean   :12.58   Mean   :1.285
##  20172  : 991                      3rd Qu.:15.00   3rd Qu.:1.000
##  20152  : 990                      Max.   :20.00   Max.   :8.000
##  (Other):1923
##  aprobo         PRA1            PRA2            PRA4            PRA5
##  No:1729   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
```

```
##  Si:7981   1st Qu.:11.00   1st Qu.:11.00   1st Qu.:11.00   1st Qu.:11.00
##            Median :13.00   Median :13.00   Median :13.00   Median :14.00
##            Mean   :12.61   Mean   :12.56   Mean   :12.56   Mean   :12.45
##            3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:16.00
##            Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##            NA's   :7                       NA's   :32      NA's   :103
##       PRA3            PRA6
##  Min.   : 0.00   Min.   : 0.00
##  1st Qu.:11.00   1st Qu.:11.00
##  Median :14.00   Median :14.00
##  Mean   :12.62   Mean   :12.43
##  3rd Qu.:16.00   3rd Qu.:16.00
##  Max.   :20.00   Max.   :20.00
##                  NA's   :135
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1 PRA2 PRA4 PRA5 PRA3 PRA6
## TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:4] 6 8 9 11
##  - attr(*, "names")= chr [1:4] "PRA1" "PRA4" "PRA5" "PRA6"
## named integer(0)
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0003*****"
##
## 'data.frame':    42167 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 5 levels "PRA1","PRA2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ eva_nota: num  12 12 11 12 14 0 0 0 0 0 ...
##  $ alu_cod : chr  "201510876" "201510876" "201510876" "201510876" ...
##  $ eva_fin : int  12 12 12 12 12 99 99 99 99 99 ...
##     semestre    eva_new        eva_nota       alu_cod
##  20151  :6610   PRA1:9931   Min.   : 0.00   Length:42167
##  20181  :6312   PRA2:9933   1st Qu.:11.00   Class :character
##  20161  :6241   PRA3:9930   Median :13.00   Mode  :character
##  20171  :6028   PRA4:9933   Mean   :12.09
##  20152  :4935   PRA5:2440   3rd Qu.:15.00
##  20172  :4156               Max.   :20.00
##  (Other):7885
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :15.73
##  3rd Qu.:15.00
##  Max.   :99.00
##
## 'data.frame':    9933 obs. of  10 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201510876" "201512136" "201512049" "201512175" ...
##  $ eva_fin : int  12 99 11 17 13 15 13 16 16 17 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 3 1 1 1 1 1 1 1 1 ...
##  $ PRA5    : num  14 0 12 18 14 18 18 16 16 17 ...
##  $ PRA1    : num  12 0 12 11 12 16 13 15 15 15 ...
##  $ PRA2    : num  12 0 9 17 15 9 8 15 14 19 ...
##  $ PRA4    : num  12 0 12 20 12 14 15 14 17 17 ...
##  $ PRA3    : num  11 0 12 19 12 17 13 18 16 17 ...
##     semestre      alu_cod             eva_fin          veces      estado
##  20181  :1578   Length:9933        Min.   : 0.00   Min.   :1.00   A:7970
##  20161  :1533   Class :character   1st Qu.:11.00   1st Qu.:1.00   D:1606
##  20171  :1507   Mode  :character   Median :13.00   Median :1.00   N: 357
##  20151  :1322                      Mean   :15.77   Mean   :1.29
##  20172  :1039                      3rd Qu.:15.00   3rd Qu.:1.00
##  20152  : 987                      Max.   :99.00   Max.   :9.00
##  (Other):1967
##       PRA5            PRA1            PRA2           PRA4
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.0   Min.   : 0.00
##  1st Qu.:11.00   1st Qu.:11.00   1st Qu.: 9.0   1st Qu.:11.00
##  Median :14.00   Median :14.00   Median :12.0   Median :13.00
##  Mean   :12.12   Mean   :12.83   Mean   :11.3   Mean   :12.14
##  3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:15.0   3rd Qu.:15.00
##  Max.   :20.00   Max.   :20.00   Max.   :20.0   Max.   :20.00
##  NA's   :7493    NA's   :2
##       PRA3
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :12.09
##  3rd Qu.:15.00
##  Max.   :20.00
##  NA's   :3
##
## 'data.frame':    9576 obs. of  10 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201510876" "201512049" "201512175" "201512706" ...
##  $ eva_fin : int  12 11 17 13 15 13 16 16 17 12 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA5    : num  14 12 18 14 18 18 16 16 17 12 ...
##  $ PRA1    : num  12 12 11 12 16 13 15 15 15 11 ...
##  $ PRA2    : num  12 9 17 15 9 8 15 14 19 12 ...
##  $ PRA4    : num  12 12 20 12 14 15 14 17 17 12 ...
##  $ PRA3    : num  11 12 19 12 17 13 18 16 17 12 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1525   Length:9576        Min.   : 0.00   Min.   :1.000
##  20161  :1488   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20171  :1444   Mode  :character   Median :13.00   Median :1.000
##  20151  :1290                      Mean   :12.67   Mean   :1.257
##  20172  : 985                      3rd Qu.:15.00   3rd Qu.:1.000
##  20152  : 959                      Max.   :20.00   Max.   :9.000
##  (Other):1885
##  aprobo         PRA5            PRA1            PRA2            PRA4
##  No:1606   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  Si:7970   1st Qu.:11.00   1st Qu.:12.00   1st Qu.: 9.00   1st Qu.:11.00
##            Median :14.00   Median :14.00   Median :12.00   Median :14.00
##            Mean   :12.51   Mean   :13.31   Mean   :11.72   Mean   :12.59
##            3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:15.00   3rd Qu.:16.00
##            Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##            NA's   :7212    NA's   :2
```

```
##       PRA3
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :12.54
##  3rd Qu.:16.00
##  Max.   :20.00
##  NA's   :3
##
## [1] "Verificacion de cantidad de Registro..."
## PRA5 PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:3] 6 7 10
##  - attr(*, "names")= chr [1:3] "PRA5" "PRA1" "PRA3"
## PRA5
##    6
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0004*****"
##
## 'data.frame':    89576 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 9 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 6 9 1 2 3
...
##  $ eva_nota: num  0 9 13 0 0 0 88 2 2 5 ...
##  $ alu_cod : chr  "201512432" "201512432" "201512432" "201512432" ...
##  $ eva_fin : int  4 4 4 4 4 4 4 3 3 3 ...
##     semestre        eva_new         eva_nota       alu_cod
##  20181  :13174   PRA2   :12548   Min.   : 0.00   Length:89576
##  20161  :13028   PRA3   :12548   1st Qu.: 5.00   Class :character
##  20171  :12117   PRA1   :12547   Median :10.00   Mode  :character
##  20152  :11097   PRA4   :12535   Mean   :15.61
##  20151  :10681   FIN    :11906   3rd Qu.:14.00
##  20172  :10549   PAR    :11906   Max.   :88.00
##  (Other):18930   (Other):15586
##     eva_fin
##  Min.   : 0.00
##  1st Qu.: 7.00
##  Median :11.00
##  Mean   :12.97
##  3rd Qu.:13.00
##  Max.   :99.00
##
## 'data.frame':    12548 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512432" "201510876" "201512136" "201512049" ...
##  $ eva_fin : int  4 3 2 11 14 13 11 11 15 15 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 2 2 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  13 5 0 10 10 14 17 4 16 20 ...
##  $ FIN     : num  0 2 0 9 13 13 7 8 15 15 ...
##  $ PRA2    : num  0 5 0 8 16 17 6 10 15 14 ...
```

```
##  $ PRA4    : num  0 4 0 17 11 15 2 11 7 16 ...
##  $ PRA3    : num  0 4 0 13 19 11 16 7 20 14 ...
##  $ PRA5    : num  NA 7 NA 9 NA 13 10 13 NA 6 ...
##  $ SUS     : num  88 88 88 13 88 88 88 88 88 88 ...
##  $ PRA6    : num  NA 6 NA NA NA NA 8 13 NA NA ...
##  $ PAR     : num  9 2 5 7 15 12 16 13 14 15 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1882   Length:12548       Min.   : 0.00   Min.   :1.000
##  20161  :1793   Class :character   1st Qu.: 7.00   1st Qu.:1.000
##  20171  :1731   Mode  :character   Median :11.00   Median :2.000
##  20172  :1507                      Mean   :13.04   Mean   :1.933
##  20162  :1390                      3rd Qu.:13.00   3rd Qu.:2.000
##  20152  :1365                      Max.   :99.00   Max.   :8.000
##  (Other):2880
##  estado        PRA1             FIN              PRA2
##  A:7004   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  D:5098   1st Qu.: 6.000   1st Qu.: 2.000   1st Qu.: 6.000
##  N: 446   Median :10.000   Median : 7.000   Median :10.000
##           Mean   : 9.913   Mean   : 7.286   Mean   : 9.915
##           3rd Qu.:14.000   3rd Qu.:11.000   3rd Qu.:14.000
##           Max.   :20.000   Max.   :77.000   Max.   :20.000
##           NA's   :1        NA's   :642
##       PRA4             PRA3             PRA5             SUS
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 3.000   1st Qu.: 5.000   1st Qu.: 1.000   1st Qu.:12.00
##  Median : 9.000   Median :10.000   Median : 7.000   Median :88.00
##  Mean   : 8.579   Mean   : 9.174   Mean   : 7.043   Mean   :59.36
##  3rd Qu.:13.000   3rd Qu.:13.000   3rd Qu.:12.000   3rd Qu.:88.00
##  Max.   :77.000   Max.   :20.000   Max.   :20.000   Max.   :88.00
##  NA's   :13                        NA's   :10155    NA's   :642
##       PRA6             PAR
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.: 5.000
##  Median : 7.000   Median : 9.000
##  Mean   : 6.748   Mean   : 9.081
##  3rd Qu.:12.000   3rd Qu.:13.000
##  Max.   :20.000   Max.   :77.000
##  NA's   :11261    NA's   :642
##
## 'data.frame':    12102 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512432" "201510876" "201512136" "201512049" ...
##  $ eva_fin : int  4 3 2 11 14 13 11 11 15 15 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 1 1 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  13 5 0 10 10 14 17 4 16 20 ...
##  $ FIN     : num  0 2 0 9 13 13 7 8 15 15 ...
##  $ PRA2    : num  0 5 0 8 16 17 6 10 15 14 ...
##  $ PRA4    : num  0 4 0 17 11 15 2 11 7 16 ...
##  $ PRA3    : num  0 4 0 13 19 11 16 7 20 14 ...
##  $ PRA5    : num  NA 7 NA 9 NA 13 10 13 NA 6 ...
##  $ SUS     : num  88 88 88 13 88 88 88 88 88 88 ...
##  $ PRA6    : num  NA 6 NA NA NA NA 8 13 NA NA ...
##  $ PAR     : num  9 2 5 7 15 12 16 13 14 15 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1818   Length:12102       Min.   : 0.000   Min.   :1.000
##  20161  :1731   Class :character   1st Qu.: 7.000   1st Qu.:1.000
##  20171  :1676   Mode  :character   Median :11.000   Median :2.000
```

```
##  20172  :1447                      Mean   : 9.871   Mean   :1.905
##  20162  :1334                      3rd Qu.:13.000   3rd Qu.:2.000
##  20152  :1323                      Max.   :20.000   Max.   :8.000
##  (Other):2773
##  aprobo         PRA1            FIN              PRA2
##  No:5098   Min.   : 0.00   Min.   : 0.000   Min.   : 0.00
##  Si:7004   1st Qu.: 6.00   1st Qu.: 3.000   1st Qu.: 6.00
##            Median :11.00   Median : 7.000   Median :11.00
##            Mean   :10.28   Mean   : 7.555   Mean   :10.28
##            3rd Qu.:14.00   3rd Qu.:11.000   3rd Qu.:14.00
##            Max.   :20.00   Max.   :77.000   Max.   :20.00
##            NA's   :1       NA's   :621
##       PRA4             PRA3             PRA5             SUS
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 4.000   1st Qu.: 5.000   1st Qu.: 2.000   1st Qu.:11.00
##  Median :10.000   Median :10.000   Median : 7.000   Median :88.00
##  Mean   : 8.894   Mean   : 9.512   Mean   : 7.303   Mean   :58.46
##  3rd Qu.:14.000   3rd Qu.:14.000   3rd Qu.:12.000   3rd Qu.:88.00
##  Max.   :77.000   Max.   :20.000   Max.   :20.000   Max.   :88.00
##  NA's   :12                        NA's   :9794     NA's   :621
##       PRA6            PAR
##  Min.   : 0.00   Min.   : 0.000
##  1st Qu.: 0.00   1st Qu.: 5.000
##  Median : 7.00   Median : 9.000
##  Mean   : 7.05   Mean   : 9.417
##  3rd Qu.:12.00   3rd Qu.:13.000
##  Max.   :20.00   Max.   :77.000
##  NA's   :10870   NA's   :621
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1  FIN PRA2 PRA4 PRA3 PRA5  SUS PRA6  PAR
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12]  TRUE FALSE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:6] 6 7 9 11 13 14
##  - attr(*, "names")= chr [1:6] "PRA1" "FIN" "PRA4" "PRA5" ...
## PRA5 PRA6
##   11   13
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0005*****"
##
## 'data.frame':    31217 obs. of  5 variables:
##  $ semestre: Factor w/ 9 levels "20161","20162",..: 1 1 1 1 1 1 1 1 1 ..
.
##  $ eva_new : Factor w/ 5 levels "PRA1","PRA2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ eva_nota: num  19 13 17 15 20 13 9 11 15 13 ...
##  $ alu_cod : chr  "201611022" "201611022" "201611022" "201611022" ...
##  $ eva_fin : int  17 17 17 17 17 12 12 12 12 12 ...
##     semestre    eva_new        eva_nota       alu_cod
##  20181  :7035   PRA1:6270   Min.   : 0.00   Length:31217
##  20171  :6500   PRA2:6272   1st Qu.: 9.00   Class :character
##  20172  :4925   PRA3:6272   Median :13.00   Mode  :character
##  20161  :4535   PRA4:6272   Mean   :12.08
##  20162  :4110   PRA5:6131   3rd Qu.:16.00
```

```
##  20182  :3548              Max.   :20.00
##  (Other): 564
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :15.11
##  3rd Qu.:16.00
##  Max.   :99.00
##
## 'data.frame':    6272 obs. of  10 variables:
##  $ semestre: Factor w/ 9 levels "20161","20162",..: 1 1 1 1 1 1 1 1 1 ..
.
##  $ alu_cod : chr  "201611022" "201611857" "201610308" "201610173" ...
##  $ eva_fin : int  17 12 19 11 18 11 11 12 11 4 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 2 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 2 ...
##  $ PRA5    : num  20 13 18 13 20 13 9 9 14 0 ...
##  $ PRA1    : num  19 13 20 14 19 14 11 11 10 6 ...
##  $ PRA2    : num  13 9 18 11 18 6 7 11 11 1 ...
##  $ PRA4    : num  15 15 19 15 17 12 15 11 7 12 ...
##  $ PRA3    : num  17 11 19 0 18 9 11 17 11 0 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1407   Length:6272        Min.   : 0.00   Min.   :1.000
##  20171  :1300   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20172  : 985   Mode  :character   Median :13.00   Median :1.000
##  20161  : 907                      Mean   :15.09   Mean   :1.308
##  20162  : 822                      3rd Qu.:16.00   3rd Qu.:1.000
##  20182  : 710                      Max.   :99.00   Max.   :5.000
##  (Other): 141
##  estado        PRA5            PRA1            PRA2            PRA4
##  A:4740   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D:1353   1st Qu.:10.00   1st Qu.: 9.00   1st Qu.: 9.00   1st Qu.:10.00
##  N: 179   Median :14.00   Median :13.00   Median :12.00   Median :14.00
##           Mean   :12.44   Mean   :12.04   Mean   :11.56   Mean   :12.24
##           3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:15.00   3rd Qu.:16.00
##           Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##           NA's   :141     NA's   :2
##       PRA3
##  Min.   : 0.00
##  1st Qu.: 9.00
##  Median :13.00
##  Mean   :12.12
##  3rd Qu.:16.00
##  Max.   :20.00
##
## 'data.frame':    6093 obs. of  10 variables:
##  $ semestre: Factor w/ 9 levels "20161","20162",..: 1 1 1 1 1 1 1 1 1 ..
.
##  $ alu_cod : chr  "201611022" "201611857" "201610308" "201610173" ...
##  $ eva_fin : int  17 12 19 11 18 11 11 12 11 4 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 2 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 1 ...
##  $ PRA5    : num  20 13 18 13 20 13 9 9 14 0 ...
##  $ PRA1    : num  19 13 20 14 19 14 11 11 10 6 ...
##  $ PRA2    : num  13 9 18 11 18 6 7 11 11 1 ...
##  $ PRA4    : num  15 15 19 15 17 12 15 11 7 12 ...
##  $ PRA3    : num  17 11 19 0 18 9 11 17 11 0 ...
##     semestre      alu_cod             eva_fin          veces
```

```
##  20181  :1367   Length:6093        Min.   : 0.00   Min.   :1.000
##  20171  :1261   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20172  : 958   Mode  :character   Median :13.00   Median :1.000
##  20161  : 891                      Mean   :12.63   Mean   :1.296
##  20162  : 789                      3rd Qu.:15.00   3rd Qu.:1.000
##  20182  : 687                      Max.   :20.00   Max.   :5.000
##  (Other): 140
##  aprobo         PRA5            PRA1            PRA2            PRA4
##  No:1353   Min.   : 0.00   Min.   : 0.0   Min.   : 0.0   Min.   : 0.0
##  Si:4740   1st Qu.:11.00   1st Qu.: 9.0   1st Qu.: 9.0   1st Qu.:10.0
##            Median :14.00   Median :13.0   Median :13.0   Median :14.0
##            Mean   :12.81   Mean   :12.4   Mean   :11.9   Mean   :12.6
##            3rd Qu.:16.00   3rd Qu.:16.0   3rd Qu.:15.0   3rd Qu.:16.0
##            Max.   :20.00   Max.   :20.0   Max.   :20.0   Max.   :20.0
##            NA's   :140     NA's   :2
##       PRA3
##  Min.   : 0.00
##  1st Qu.:10.00
##  Median :13.00
##  Mean   :12.47
##  3rd Qu.:16.00
##  Max.   :20.00
##
## [1] "Verificacion de cantidad de Registro..."
## PRA5 PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:2] 6 7
##  - attr(*, "names")= chr [1:2] "PRA5" "PRA1"
## named integer(0)
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0006*****"
##
## 'data.frame':    55483 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 6 7 1 2 3
...
##  $ eva_nota: num  8 9 10 14 6 16 5 12 10 17 ...
##  $ alu_cod : chr  "201512446" "201512446" "201512446" "201512446" ...
##  $ eva_fin : int  10 10 10 10 10 10 10 13 13 13 ...
##     semestre     eva_new        eva_nota       alu_cod
##  20182  :10526   FIN :7735   Min.   : 0.00   Length:55483
##  20172  : 9926   PAR :7734   1st Qu.: 9.00   Class :character
##  20162  : 9300   PRA1:8088   Median :13.00   Mode  :character
##  20181  : 6580   PRA2:8088   Mean   :19.15
##  20152  : 6202   PRA3:8089   3rd Qu.:17.00
##  20171  : 5884   PRA4:8089   Max.   :88.00
##  (Other): 7065   SUS :7660
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :12.00
##  Mean   :13.68
```

```
##  3rd Qu.:14.00
##  Max.   :99.00
##
## 'data.frame':    8089 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512446" "201512451" "201512427" "201512425" ...
##  $ eva_fin : int  10 13 13 13 11 18 13 15 16 10 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 3 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 1 1 1 1 1 1 1 1 2 ...
##  $ PRA1    : num  10 17 11 3 14 18 10 10 17 15 ...
##  $ FIN     : num  8 12 13 11 10 19 13 14 17 4 ...
##  $ PRA2    : num  14 15 13 14 13 15 20 19 12 9 ...
##  $ PRA4    : num  16 17 16 16 0 14 17 19 15 17 ...
##  $ PRA3    : num  6 17 18 16 14 12 12 19 14 17 ...
##  $ SUS     : num  5 88 88 88 14 88 88 88 88 10 ...
##  $ PAR     : num  9 10 12 16 6 19 12 14 15 3 ...
##     semestre      alu_cod             eva_fin          veces
##  20182  :1509   Length:8089        Min.   : 0.00   Min.   :1.000
##  20172  :1418   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20162  :1334   Mode  :character   Median :12.00   Median :1.000
##  20181  : 940                      Mean   :13.72   Mean   :1.456
##  20152  : 886                      3rd Qu.:14.00   3rd Qu.:2.000
##  20171  : 841                      Max.   :99.00   Max.   :6.000
##  (Other):1161
##  estado        PRA1            FIN             PRA2            PRA4
##  A:6062   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D:1845   1st Qu.: 8.00   1st Qu.: 9.00   1st Qu.: 8.00   1st Qu.:12.00
##  N: 182   Median :12.00   Median :12.00   Median :12.00   Median :15.00
##           Mean   :11.16   Mean   :11.25   Mean   :11.68   Mean   :13.17
##           3rd Qu.:15.00   3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:17.00
##           Max.   :20.00   Max.   :77.00   Max.   :20.00   Max.   :20.00
##           NA's   :1       NA's   :354     NA's   :1
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.000
##  1st Qu.: 9.00   1st Qu.:16.00   1st Qu.: 7.000
##  Median :13.00   Median :88.00   Median :10.000
##  Mean   :12.16   Mean   :66.61   Mean   : 9.749
##  3rd Qu.:16.00   3rd Qu.:88.00   3rd Qu.:13.000
##  Max.   :20.00   Max.   :88.00   Max.   :77.000
##                  NA's   :429     NA's   :355
##
## 'data.frame':    7907 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512446" "201512451" "201512427" "201512425" ...
##  $ eva_fin : int  10 13 13 13 11 18 13 15 16 10 ...
##  $ veces   : int  2 1 1 1 1 1 1 1 1 3 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 2 2 2 2 2 2 2 2 1 ...
##  $ PRA1    : num  10 17 11 3 14 18 10 10 17 15 ...
##  $ FIN     : num  8 12 13 11 10 19 13 14 17 4 ...
##  $ PRA2    : num  14 15 13 14 13 15 20 19 12 9 ...
##  $ PRA4    : num  16 17 16 16 0 14 17 19 15 17 ...
##  $ PRA3    : num  6 17 18 16 14 12 12 19 14 17 ...
##  $ SUS     : num  5 88 88 88 14 88 88 88 88 10 ...
##  $ PAR     : num  9 10 12 16 6 19 12 14 15 3 ...
##     semestre      alu_cod             eva_fin          veces
##  20182  :1471   Length:7907        Min.   : 0.00   Min.   :1.000
##  20172  :1374   Class :character   1st Qu.:11.00   1st Qu.:1.000
```

```
##  20162  :1318   Mode  :character   Median :12.00   Median :1.000
##  20181  : 912                      Mean   :11.76   Mean   :1.434
##  20152  : 879                      3rd Qu.:14.00   3rd Qu.:2.000
##  20171  : 818                      Max.   :20.00   Max.   :6.000
##  (Other):1135
##  aprobo         PRA1            FIN            PRA2            PRA4
##  No:1845   Min.   : 0.00   Min.   : 0.0   Min.   : 0.00   Min.   : 0.00
##  Si:6062   1st Qu.: 8.00   1st Qu.: 9.0   1st Qu.: 9.00   1st Qu.:12.00
##            Median :12.00   Median :12.0   Median :13.00   Median :15.00
##            Mean   :11.42   Mean   :11.5   Mean   :11.95   Mean   :13.48
##            3rd Qu.:15.00   3rd Qu.:15.0   3rd Qu.:16.00   3rd Qu.:17.00
##            Max.   :20.00   Max.   :77.0   Max.   :20.00   Max.   :20.00
##            NA's   :1       NA's   :342    NA's   :1
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.000
##  1st Qu.:10.00   1st Qu.:15.00   1st Qu.: 7.000
##  Median :14.00   Median :88.00   Median :10.000
##  Mean   :12.44   Mean   :66.12   Mean   : 9.968
##  3rd Qu.:16.00   3rd Qu.:88.00   3rd Qu.:13.000
##  Max.   :20.00   Max.   :88.00   Max.   :77.000
##                  NA's   :417     NA's   :343
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1  FIN PRA2 PRA4 PRA3  SUS  PAR
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FAL
SE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:4] 6 7 8 12
##  - attr(*, "names")= chr [1:4] "PRA1" "FIN" "PRA2" "PAR"
## named integer(0)
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0007*****"
##
## 'data.frame':    46204 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 7 1 2 3 4
...
##  $ eva_nota: num  0 7 16 8 4 88 0 10 0 12 ...
##  $ alu_cod : chr  "201512432" "201512432" "201512432" "201512432" ...
##  $ eva_fin : int  5 5 5 5 5 5 5 5 5 5 ...
##     semestre    eva_new        eva_nota       alu_cod
##  20182  :8878   FIN :7458   Min.   : 0.00   Length:46204
##  20172  :8094   PAR :7458   1st Qu.: 9.00   Class :character
##  20162  :7665   PRA1:7867   Median :13.00   Mode  :character
##  20181  :5856   PRA2:7867   Mean   :20.18
##  20171  :5010   PRA3:7867   3rd Qu.:16.00
##  20152  :4716   PRA4: 409   Max.   :88.00
##  (Other):5985   SUS :7278
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :12.00
##  Mean   :14.35
```

```
##  3rd Qu.:14.00
##  Max.   :99.00
##
## 'data.frame':    7867 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 2 2 2 2 2 2 .
..
##  $ alu_cod : chr  "201512432" "201512436" "201512475" "201512432" ...
##  $ eva_fin : int  5 5 12 99 11 16 18 11 9 16 ...
##  $ veces   : int  4 2 1 4 1 1 1 1 2 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 2 2 1 3 1 1 1 1 2 1 ...
##  $ PRA1    : num  16 0 12 0 15 18 16 10 10 17 ...
##  $ FIN     : num  0 0 17 0 14 18 18 11 6 14 ...
##  $ PRA2    : num  8 12 12 0 18 18 16 10 16 14 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  4 7 13 0 13 19 17 11 12 16 ...
##  $ SUS     : num  88 NA 88 88 88 88 88 NA 8 88 ...
##  $ PAR     : num  7 10 8 0 5 13 19 11 4 18 ...
##     semestre      alu_cod             eva_fin          veces
##  20182  :1486   Length:7867        Min.   : 0.00   Min.   :1.000
##  20172  :1349   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20162  :1285   Mode  :character   Median :12.00   Median :1.000
##  20181  : 976                      Mean   :14.34   Mean   :1.438
##  20171  : 835                      3rd Qu.:14.00   3rd Qu.:2.000
##  20152  : 802                      Max.   :99.00   Max.   :6.000
##  (Other):1134
##  estado        PRA1            FIN             PRA2            PRA4
##  A:5891   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D:1739   1st Qu.: 9.00   1st Qu.: 7.00   1st Qu.: 9.00   1st Qu.:11.00
##  N: 237   Median :12.00   Median :11.00   Median :12.00   Median :13.00
##           Mean   :11.62   Mean   :10.17   Mean   :11.71   Mean   :12.42
##           3rd Qu.:15.00   3rd Qu.:14.00   3rd Qu.:16.00   3rd Qu.:15.00
##           Max.   :20.00   Max.   :77.00   Max.   :20.00   Max.   :20.00
##                           NA's   :409                     NA's   :7458
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.: 9.00   1st Qu.:17.00   1st Qu.: 8.00
##  Median :13.00   Median :88.00   Median :11.00
##  Mean   :11.74   Mean   :67.85   Mean   :10.96
##  3rd Qu.:16.00   3rd Qu.:88.00   3rd Qu.:14.00
##  Max.   :20.00   Max.   :88.00   Max.   :77.00
##                  NA's   :589     NA's   :409
##
## 'data.frame':    7630 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 2 2 2 2 2 2 .
..
##  $ alu_cod : chr  "201512432" "201512436" "201512475" "201512049" ...
##  $ eva_fin : int  5 5 12 11 16 18 11 9 16 14 ...
##  $ veces   : int  4 2 1 1 1 1 1 2 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 1 1 2 2 2 2 2 1 2 2 ...
##  $ PRA1    : num  16 0 12 15 18 16 10 10 17 13 ...
##  $ FIN     : num  0 0 17 14 18 18 11 6 14 14 ...
##  $ PRA2    : num  8 12 12 18 18 16 10 16 14 14 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  4 7 13 13 19 17 11 12 16 14 ...
##  $ SUS     : num  88 NA 88 88 88 88 NA 8 88 NA ...
##  $ PAR     : num  7 10 8 5 13 19 11 4 18 14 ...
##     semestre      alu_cod             eva_fin          veces
##  20182  :1424   Length:7630        Min.   : 0.00   Min.   :1.000
##  20172  :1306   Class :character   1st Qu.:11.00   1st Qu.:1.000
```

```
##  20162  :1261   Mode  :character   Median :12.00   Median :1.000
##  20181  : 942                      Mean   :11.71   Mean   :1.413
##  20152  : 794                      3rd Qu.:14.00   3rd Qu.:2.000
##  20171  : 793                      Max.   :20.00   Max.   :6.000
##  (Other):1110
##  aprobo         PRA1            FIN             PRA2            PRA4
##  No:1739   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  Si:5891   1st Qu.:10.00   1st Qu.: 7.00   1st Qu.: 9.00   1st Qu.:11.00
##            Median :12.00   Median :11.00   Median :13.00   Median :13.00
##            Mean   :11.98   Mean   :10.49   Mean   :12.08   Mean   :12.67
##            3rd Qu.:15.00   3rd Qu.:14.00   3rd Qu.:16.00   3rd Qu.:15.00
##            Max.   :20.00   Max.   :77.00   Max.   :20.00   Max.   :20.00
##                            NA's   :401                     NA's   :7229
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:10.00   1st Qu.:16.00   1st Qu.: 8.00
##  Median :13.00   Median :88.00   Median :12.00
##  Mean   :12.11   Mean   :67.23   Mean   :11.31
##  3rd Qu.:16.00   3rd Qu.:88.00   3rd Qu.:14.00
##  Max.   :20.00   Max.   :88.00   Max.   :77.00
##                  NA's   :577     NA's   :401
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1  FIN PRA2 PRA4 PRA3  SUS  PAR
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALS
E
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:3] 7 9 12
##  - attr(*, "names")= chr [1:3] "FIN" "PRA4" "PAR"
## PRA4
##    9
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0008*****"
##
## 'data.frame':    22572 obs. of  5 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ eva_new : Factor w/ 4 levels "PRA1","PRA2",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ eva_nota: num  19 19 16 17 13 10 13 12 15 16 ...
##  $ alu_cod : chr  "201520493" "201520493" "201520493" "201520493" ...
##  $ eva_fin : int  18 18 18 18 12 12 12 12 16 16 ...
##     semestre    eva_new        eva_nota       alu_cod
##  20172  :5620   PRA1:5643   Min.   : 0.00   Length:22572
##  20182  :5460   PRA2:5643   1st Qu.:11.00   Class :character
##  20162  :4012   PRA3:5643   Median :14.00   Mode  :character
##  20181  :3400   PRA4:5643   Mean   :13.01
##  20171  :3384               3rd Qu.:16.00
##  20170  : 244               Max.   :20.00
##  (Other): 452
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:12.00
##  Median :14.00
##  Mean   :16.21
```

```
##  3rd Qu.:16.00
##  Max.   :99.00
##
## 'data.frame':    5643 obs. of  9 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 ..
.
##  $ alu_cod : chr  "201520493" "201610280" "201610869" "201610586" ...
##  $ eva_fin : int  18 12 16 15 13 16 9 15 16 16 ...
##  $ veces   : int  1 1 1 1 1 1 2 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 2 1 1 1 ...
##  $ PRA1    : num  19 13 15 12 13 14 11 15 15 15 ...
##  $ PRA2    : num  19 10 16 17 8 18 8 18 19 14 ...
##  $ PRA4    : num  17 12 15 17 16 16 11 13 16 16 ...
##  $ PRA3    : num  16 13 17 14 14 15 6 13 15 18 ...
##     semestre      alu_cod             eva_fin          veces
##  20172  :1405   Length:5643        Min.   : 0.00   Min.   :1.000
##  20182  :1365   Class :character   1st Qu.:12.00   1st Qu.:1.000
##  20162  :1003   Mode  :character   Median :14.00   Median :1.000
##  20181  : 850                      Mean   :16.21   Mean   :1.238
##  20171  : 846                      3rd Qu.:16.00   3rd Qu.:1.000
##  20170  :  61                      Max.   :99.00   Max.   :6.000
##  (Other): 113
##  estado        PRA1            PRA2            PRA4            PRA3
##  A:4805   Min.   : 0.00   Min.   : 0.0   Min.   : 0.00   Min.   : 0.00
##  D: 663   1st Qu.:11.00   1st Qu.:11.0   1st Qu.:12.00   1st Qu.:11.00
##  N: 175   Median :14.00   Median :14.0   Median :15.00   Median :14.00
##           Mean   :12.97   Mean   :12.9   Mean   :13.38   Mean   :12.81
##           3rd Qu.:16.00   3rd Qu.:16.0   3rd Qu.:17.00   3rd Qu.:16.00
##           Max.   :20.00   Max.   :20.0   Max.   :20.00   Max.   :20.00
##
## 'data.frame':    5468 obs. of  9 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 ..
.
##  $ alu_cod : chr  "201520493" "201610280" "201610869" "201610586" ...
##  $ eva_fin : int  18 12 16 15 13 16 9 15 16 16 ...
##  $ veces   : int  1 1 1 1 1 1 2 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 1 2 2 2 ...
##  $ PRA1    : num  19 13 15 12 13 14 11 15 15 15 ...
##  $ PRA2    : num  19 10 16 17 8 18 8 18 19 14 ...
##  $ PRA4    : num  17 12 15 17 16 16 11 13 16 16 ...
##  $ PRA3    : num  16 13 17 14 14 15 6 13 15 18 ...
##     semestre      alu_cod             eva_fin          veces
##  20172  :1354   Length:5468        Min.   : 0.00   Min.   :1.000
##  20182  :1306   Class :character   1st Qu.:12.00   1st Qu.:1.000
##  20162  : 988   Mode  :character   Median :14.00   Median :1.000
##  20171  : 830                      Mean   :13.56   Mean   :1.211
##  20181  : 822                      3rd Qu.:16.00   3rd Qu.:1.000
##  20170  :  59                      Max.   :20.00   Max.   :6.000
##  (Other): 109
##  aprobo         PRA1            PRA2            PRA4            PRA3
##  No: 663   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  Si:4805   1st Qu.:11.75   1st Qu.:11.00   1st Qu.:12.00   1st Qu.:11.00
##            Median :14.00   Median :14.00   Median :15.00   Median :14.00
##            Mean   :13.38   Mean   :13.31   Mean   :13.81   Mean   :13.22
##            3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:17.00   3rd Qu.:16.00
##            Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1 PRA2 PRA4 PRA3
```

```
## TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int(0)
##  - attr(*, "names")= chr(0)
## named integer(0)
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0009*****"
##
## 'data.frame':    20215 obs. of  5 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ eva_new : Factor w/ 5 levels "PRA1","PRA2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ eva_nota: num  19 19 18 19 20 20 16 18 19 20 ...
##  $ alu_cod : chr  "201611022" "201611022" "201611022" "201611022" ...
##  $ eva_fin : int  19 19 19 19 19 19 19 19 19 19 ...
##     semestre    eva_new        eva_nota       alu_cod
##  20182  :5235   PRA1:4072   Min.   : 0.00   Length:20215
##  20172  :4940   PRA2:4072   1st Qu.:10.00   Class :character
##  20181  :3865   PRA3:4072   Median :13.00   Mode  :character
##  20171  :2845   PRA4:4072   Mean   :12.52
##  20162  :2750   PRA5:3927   3rd Qu.:16.00
##  20180  : 224               Max.   :20.00
##  (Other): 356
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :14.77
##  3rd Qu.:15.00
##  Max.   :99.00
##
## 'data.frame':    4072 obs. of  10 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ alu_cod : chr  "201611022" "201610308" "201610173" "201610918" ...
##  $ eva_fin : int  19 19 12 18 9 14 12 13 11 12 ...
##  $ veces   : int  1 1 1 1 2 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 2 1 1 1 1 1 ...
##  $ PRA5    : num  20 20 12 19 9 14 8 16 10 10 ...
##  $ PRA1    : num  19 20 11 17 12 14 14 8 8 13 ...
##  $ PRA2    : num  19 16 13 19 0 9 9 12 7 11 ...
##  $ PRA4    : num  19 19 12 18 14 17 14 13 15 16 ...
##  $ PRA3    : num  18 18 14 19 12 14 14 14 13 12 ...
##     semestre      alu_cod             eva_fin          veces
##  20182  :1047   Length:4072        Min.   : 0.00   Min.   :1.000
##  20172  : 988   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20181  : 773   Mode  :character   Median :13.00   Median :1.000
##  20171  : 569                      Mean   :14.78   Mean   :1.263
##  20162  : 550                      3rd Qu.:15.00   3rd Qu.:1.000
##  20180  :  56                      Max.   :99.00   Max.   :5.000
##  (Other):  89
##  estado        PRA5            PRA1            PRA2            PRA4
##  A:3369   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D: 612   1st Qu.:11.00   1st Qu.: 9.00   1st Qu.:10.00   1st Qu.:11.00
```

```
##  N:  91   Median :14.00   Median :12.00   Median :13.00   Median :14.00
##           Mean   :13.23   Mean   :11.49   Mean   :12.59   Mean   :12.92
##           3rd Qu.:16.00   3rd Qu.:14.00   3rd Qu.:16.00   3rd Qu.:16.00
##           Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##           NA's   :145
##       PRA3
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :12.38
##  3rd Qu.:15.00
##  Max.   :20.00
##
## 'data.frame':    3981 obs. of  10 variables:
##  $ semestre: Factor w/ 8 levels "20162","20170",..: 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ alu_cod : chr  "201611022" "201610308" "201610173" "201610918" ...
##  $ eva_fin : int  19 19 12 18 9 14 12 13 11 12 ...
##  $ veces   : int  1 1 1 1 2 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 1 2 2 2 2 2 ...
##  $ PRA5    : num  20 20 12 19 9 14 8 16 10 10 ...
##  $ PRA1    : num  19 20 11 17 12 14 14 8 8 13 ...
##  $ PRA2    : num  19 16 13 19 0 9 9 12 7 11 ...
##  $ PRA4    : num  19 19 12 18 14 17 14 13 15 16 ...
##  $ PRA3    : num  18 18 14 19 12 14 14 14 13 12 ...
##     semestre      alu_cod             eva_fin          veces
##  20182  :1020   Length:3981        Min.   : 0.00   Min.   :1.000
##  20172  : 971   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20181  : 750   Mode  :character   Median :13.00   Median :1.000
##  20171  : 557                      Mean   :12.86   Mean   :1.238
##  20162  : 544                      3rd Qu.:15.00   3rd Qu.:1.000
##  20180  :  52                      Max.   :20.00   Max.   :5.000
##  (Other):  87
##  aprobo         PRA5            PRA1            PRA2            PRA4
##  No: 612   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  Si:3369   1st Qu.:12.00   1st Qu.: 9.00   1st Qu.:10.00   1st Qu.:11.00
##            Median :14.00   Median :12.00   Median :13.00   Median :14.00
##            Mean   :13.52   Mean   :11.75   Mean   :12.88   Mean   :13.22
##            3rd Qu.:16.00   3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:16.00
##            Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##            NA's   :139
##       PRA3
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :12.66
##  3rd Qu.:15.00
##  Max.   :20.00
##
## [1] "Verificacion de cantidad de Registro..."
## PRA5 PRA1 PRA2 PRA4 PRA3
## TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int 6
##  - attr(*, "names")= chr "PRA5"
```

```
## named integer(0)
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0010*****"
##
## 'data.frame':    45610 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 7 1 2 3 4
...
##  $ eva_nota: num  18 18 13 19 18 88 13 8 8 14 ...
##  $ alu_cod : chr  "201512444" "201512444" "201512444" "201512444" ...
##  $ eva_fin : int  18 18 18 18 18 18 11 11 11 11 ...
##     semestre    eva_new        eva_nota       alu_cod
##  20182  :8665   FIN :7466   Min.   : 0.00   Length:45610
##  20162  :7719   PAR :7466   1st Qu.:10.00   Class :character
##  20172  :7555   PRA1:7725   Median :14.00   Mode  :character
##  20152  :5845   PRA2:7692   Mean   :22.11
##  20181  :5262   PRA3:7692   3rd Qu.:17.00
##  20171  :4704   PRA4: 259   Max.   :88.00
##  (Other):5860   SUS :7310
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :15.49
##  3rd Qu.:15.00
##  Max.   :99.00
##
## 'data.frame':    7725 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512444" "201512446" "201512451" "201512473" ...
##  $ eva_fin : int  18 11 13 15 13 11 11 18 15 15 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  13 8 15 17 9 11 13 15 13 15 ...
##  $ FIN     : num  18 13 9 15 12 11 9 20 16 14 ...
##  $ PRA2    : num  19 14 13 15 11 14 12 18 17 12 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  18 9 17 14 15 13 14 19 11 14 ...
##  $ SUS     : num  88 11 88 88 88 88 88 88 88 88 ...
##  $ PAR     : num  18 8 15 14 15 8 11 17 15 16 ...
##     semestre      alu_cod             eva_fin          veces
##  20182  :1452   Length:7725        Min.   : 0.00   Min.   :1.000
##  20162  :1290   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20172  :1262   Mode  :character   Median :13.00   Median :1.000
##  20152  : 986                      Mean   :15.49   Mean   :1.329
##  20181  : 888                      3rd Qu.:15.00   3rd Qu.:1.000
##  20171  : 784                      Max.   :99.00   Max.   :6.000
##  (Other):1063
##  estado        PRA1            FIN             PRA2            PRA4
##  A:6224   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D:1234   1st Qu.: 9.00   1st Qu.: 9.00   1st Qu.:11.00   1st Qu.:14.00
##  N: 267   Median :13.00   Median :13.00   Median :14.00   Median :16.00
##           Mean   :11.86   Mean   :11.42   Mean   :12.62   Mean   :14.75
##           3rd Qu.:16.00   3rd Qu.:15.00   3rd Qu.:17.00   3rd Qu.:17.00
##           Max.   :20.00   Max.   :77.00   Max.   :20.00   Max.   :20.00
##                           NA's   :259     NA's   :33      NA's   :7466
```

```
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:12.00   1st Qu.:88.00   1st Qu.: 8.00
##  Median :15.00   Median :88.00   Median :12.00
##  Mean   :13.19   Mean   :74.73   Mean   :11.11
##  3rd Qu.:17.00   3rd Qu.:88.00   3rd Qu.:15.00
##  Max.   :20.00   Max.   :88.00   Max.   :77.00
##  NA's   :33      NA's   :415     NA's   :259
##
## 'data.frame':    7458 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512444" "201512446" "201512451" "201512473" ...
##  $ eva_fin : int  18 11 13 15 13 11 11 18 15 15 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  13 8 15 17 9 11 13 15 13 15 ...
##  $ FIN     : num  18 13 9 15 12 11 9 20 16 14 ...
##  $ PRA2    : num  19 14 13 15 11 14 12 18 17 12 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  18 9 17 14 15 13 14 19 11 14 ...
##  $ SUS     : num  88 11 88 88 88 88 88 88 88 88 ...
##  $ PAR     : num  18 8 15 14 15 8 11 17 15 16 ...
##     semestre      alu_cod             eva_fin         veces
##  20182  :1393   Length:7458        Min.   : 0.0   Min.   :1.000
##  20162  :1260   Class :character   1st Qu.:11.0   1st Qu.:1.000
##  20172  :1213   Mode  :character   Median :13.0   Median :1.000
##  20152  : 969                      Mean   :12.5   Mean   :1.291
##  20181  : 835                      3rd Qu.:15.0   3rd Qu.:1.000
##  20161  : 760                      Max.   :20.0   Max.   :6.000
##  (Other):1028
##  aprobo         PRA1            FIN             PRA2            PRA4
##  No:1234   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  Si:6224   1st Qu.:10.00   1st Qu.: 9.00   1st Qu.:11.00   1st Qu.:14.00
##            Median :14.00   Median :13.00   Median :15.00   Median :16.00
##            Mean   :12.28   Mean   :11.83   Mean   :13.07   Mean   :15.16
##            3rd Qu.:16.00   3rd Qu.:15.00   3rd Qu.:17.00   3rd Qu.:17.00
##            Max.   :20.00   Max.   :77.00   Max.   :20.00   Max.   :20.00
##                            NA's   :252     NA's   :33      NA's   :7206
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:13.00   1st Qu.:88.00   1st Qu.: 9.00
##  Median :15.00   Median :88.00   Median :12.00
##  Mean   :13.66   Mean   :74.25   Mean   :11.51
##  3rd Qu.:17.00   3rd Qu.:88.00   3rd Qu.:15.00
##  Max.   :20.00   Max.   :88.00   Max.   :77.00
##  NA's   :33      NA's   :405     NA's   :252
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1  FIN PRA2 PRA4 PRA3  SUS  PAR
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALS
E
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:5] 7 8 9 10 12
##  - attr(*, "names")= chr [1:5] "FIN" "PRA2" "PRA4" "PRA3" ...
```

```
## PRA4
##    9
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0011*****"
##
## 'data.frame':    49665 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 9 levels "FIN","NPA","PAR",..: 1 2 3 4 5 6 7 8 9 1
...
##  $ eva_nota: num  15 13 16 14 17 16 14 88 15 14 ...
##  $ alu_cod : chr  "201512444" "201512444" "201512444" "201512444" ...
##  $ eva_fin : int  15 15 15 15 15 15 15 15 15 12 ...
##     semestre        eva_new         eva_nota       alu_cod
##  20181  :10414   PRA1   : 6290   Min.   : 0.00   Length:49665
##  20171  : 9695   PRA2   : 6290   1st Qu.: 9.00   Class :character
##  20182  : 8035   PRA3   : 6250   Median :13.00   Mode  :character
##  20172  : 7431   PRA4   : 6250   Mean   :18.45
##  20161  : 6001   FIN    : 6162   3rd Qu.:16.00
##  20162  : 4560   PAR    : 6162   Max.   :88.00
##  (Other): 3529   (Other):12261
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :12.00
##  Mean   :14.47
##  3rd Qu.:14.00
##  Max.   :99.00
##
## 'data.frame':    6290 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512444" "201512430" "201512446" "201512451" ...
##  $ eva_fin : int  15 12 10 12 14 12 14 13 14 16 ...
##  $ veces   : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 2 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  14 10 4 8 11 7 17 10 13 16 ...
##  $ FIN     : num  15 14 7 14 11 14 10 13 15 11 ...
##  $ PRA2    : num  17 10 3 9 16 12 10 14 14 14 ...
##  $ PRA4    : num  14 13 8 10 14 11 12 14 14 20 ...
##  $ PRA3    : num  16 12 8 9 12 11 18 14 15 14 ...
##  $ NPA     : num  13 14 14 18 17 13 20 15 14 18 ...
##  $ TRA     : num  15 18 12 7 17 12 20 15 15 20 ...
##  $ SUS     : num  88 88 12 88 88 88 88 88 88 88 ...
##  $ PAR     : num  16 6 9 18 14 12 11 12 14 16 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1311   Length:6290        Min.   : 0.00   Min.   :1.000
##  20171  :1221   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20182  :1009   Mode  :character   Median :12.00   Median :1.000
##  20172  : 938                      Mean   :14.47   Mean   :1.398
##  20161  : 754                      3rd Qu.:14.00   3rd Qu.:2.000
##  20162  : 570                      Max.   :99.00   Max.   :6.000
##  (Other): 487
##  estado        PRA1            FIN              PRA2            PRA4
##  A:4811   Min.   : 0.00   Min.   : 0.000   Min.   : 0.00   Min.   : 0.00
##  D:1289   1st Qu.: 9.00   1st Qu.: 7.000   1st Qu.: 8.00   1st Qu.: 7.00
##  N: 190   Median :12.00   Median :11.000   Median :12.00   Median :11.00
##           Mean   :11.49   Mean   : 9.771   Mean   :11.06   Mean   :10.48
```

```
##           3rd Qu.:15.00   3rd Qu.:13.000   3rd Qu.:15.00   3rd Qu.:15.00
##           Max.   :20.00   Max.   :77.000   Max.   :20.00   Max.   :20.00
##                           NA's   :128                      NA's   :40
##       PRA3            NPA             TRA             SUS
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.0
##  1st Qu.: 8.00   1st Qu.:12.00   1st Qu.:12.00   1st Qu.:88.0
##  Median :12.00   Median :14.00   Median :14.00   Median :88.0
##  Mean   :10.73   Mean   :13.48   Mean   :12.66   Mean   :73.5
##  3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:88.0
##  Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :88.0
##  NA's   :40      NA's   :5971    NA's   :168     NA's   :470
##       PAR
##  Min.   : 0.00
##  1st Qu.: 9.00
##  Median :12.00
##  Mean   :11.72
##  3rd Qu.:15.00
##  Max.   :77.00
##  NA's   :128
##
## 'data.frame':    6100 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512444" "201512430" "201512446" "201512451" ...
##  $ eva_fin : int  15 12 10 12 14 12 14 13 14 16 ...
##  $ veces   : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 1 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  14 10 4 8 11 7 17 10 13 16 ...
##  $ FIN     : num  15 14 7 14 11 14 10 13 15 11 ...
##  $ PRA2    : num  17 10 3 9 16 12 10 14 14 14 ...
##  $ PRA4    : num  14 13 8 10 14 11 12 14 14 20 ...
##  $ PRA3    : num  16 12 8 9 12 11 18 14 15 14 ...
##  $ NPA     : num  13 14 14 18 17 13 20 15 14 18 ...
##  $ TRA     : num  15 18 12 7 17 12 20 15 15 20 ...
##  $ SUS     : num  88 88 12 88 88 88 88 88 88 88 ...
##  $ PAR     : num  16 6 9 18 14 12 11 12 14 16 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1271   Length:6100        Min.   : 0.00   Min.   :1.000
##  20171  :1198   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20182  : 971   Mode  :character   Median :12.00   Median :1.000
##  20172  : 895                      Mean   :11.84   Mean   :1.368
##  20161  : 733                      3rd Qu.:14.00   3rd Qu.:1.000
##  20162  : 556                      Max.   :20.00   Max.   :6.000
##  (Other): 476
##  aprobo         PRA1            FIN             PRA2           PRA4
##  No:1289   Min.   : 0.00   Min.   : 0.00   Min.   : 0.0   Min.   : 0.00
##  Si:4811   1st Qu.: 9.00   1st Qu.: 8.00   1st Qu.: 9.0   1st Qu.: 8.00
##            Median :12.00   Median :11.00   Median :12.0   Median :12.00
##            Mean   :11.85   Mean   :10.08   Mean   :11.4   Mean   :10.81
##            3rd Qu.:15.00   3rd Qu.:13.00   3rd Qu.:15.0   3rd Qu.:15.00
##            Max.   :20.00   Max.   :77.00   Max.   :20.0   Max.   :20.00
##                            NA's   :126                    NA's   :39
##       PRA3            NPA             TRA             SUS
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.: 8.00   1st Qu.:12.00   1st Qu.:12.00   1st Qu.:88.00
##  Median :12.00   Median :14.00   Median :14.00   Median :88.00
##  Mean   :11.06   Mean   :13.82   Mean   :13.06   Mean   :73.04
##  3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:88.00
##  Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :88.00
```

```
##  NA's   :39      NA's   :5789    NA's   :165     NA's   :461
##       PAR
##  Min.   : 0.00
##  1st Qu.:10.00
##  Median :12.00
##  Mean   :12.08
##  3rd Qu.:15.00
##  Max.   :77.00
##  NA's   :126
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1  FIN PRA2 PRA4 PRA3  NPA  TRA  SUS  PAR
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE  TRUE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:6] 7 9 10 11 12 14
##  - attr(*, "names")= chr [1:6] "FIN" "PRA4" "PRA3" "NPA" ...
## NPA
##  11
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0012*****"
##
## 'data.frame':    38618 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 9 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 6 7 9 8 1
...
##  $ eva_nota: num  16 16 15 13 15 19 19 88 16 12 ...
##  $ alu_cod : chr  "201512444" "201512444" "201512444" "201512444" ...
##  $ eva_fin : int  16 16 16 16 16 16 16 16 16 14 ...
##     semestre       eva_new        eva_nota       alu_cod
##  20181  :8551   PRA1   :5507   Min.   : 0.00   Length:38618
##  20171  :8473   PRA2   :5461   1st Qu.:11.00   Class :character
##  20182  :6678   PRA3   :5461   Median :14.00   Mode  :character
##  20172  :5570   PRA4   :5461   Mean   :20.53
##  20161  :4731   FIN    :5402   3rd Qu.:17.00
##  20162  :3558   PAR    :5402   Max.   :88.00
##  (Other):1057   (Other):5924
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:12.00
##  Median :13.00
##  Mean   :15.73
##  3rd Qu.:15.00
##  Max.   :99.00
##
## 'data.frame':    5507 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512444" "201512430" "201512446" "201512425" ...
##  $ eva_fin : int  16 14 11 14 12 16 11 11 16 11 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  15 7 12 15 10 15 12 11 15 6 ...
```

```
##  $ FIN     : num  16 12 11 12 11 18 16 12 19 0 ...
##  $ PRA2    : num  13 13 14 11 10 13 14 11 15 12 ...
##  $ PRA4    : num  19 16 17 15 15 19 14 12 15 0 ...
##  $ PRA3    : num  15 16 11 17 10 15 0 8 15 14 ...
##  $ PRA5    : num  19 17 NA 17 NA 19 NA 0 NA NA ...
##  $ SUS     : num  88 88 88 88 NA 88 9 88 NA 7 ...
##  $ PRA6    : num  16 14 17 14 14 16 0 14 16 15 ...
##  $ PAR     : num  16 15 7 14 11 14 9 8 14 12 ...
##     semestre      alu_cod             eva_fin          veces      estado
##  20181  :1231   Length:5507        Min.   : 0.00   Min.   :1.00   A:4746
##  20171  :1221   Class :character   1st Qu.:12.00   1st Qu.:1.00   D: 592
##  20182  : 844   Mode  :character   Median :13.00   Median :1.00   N: 169
##  20172  : 806                      Mean   :15.76   Mean   :1.19
##  20161  : 673                      3rd Qu.:15.00   3rd Qu.:1.00
##  20162  : 518                      Max.   :99.00   Max.   :5.00
##  (Other): 214
##       PRA1            FIN            PRA2            PRA4
##  Min.   : 0.00   Min.   : 0.0   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:11.00   1st Qu.:10.0   1st Qu.:11.00   1st Qu.:12.00
##  Median :14.00   Median :13.0   Median :14.00   Median :15.00
##  Mean   :12.45   Mean   :12.3   Mean   :12.74   Mean   :13.25
##  3rd Qu.:15.00   3rd Qu.:16.0   3rd Qu.:16.00   3rd Qu.:17.00
##  Max.   :20.00   Max.   :20.0   Max.   :20.00   Max.   :20.00
##                  NA's   :105    NA's   :46      NA's   :46
##       PRA3            PRA5            SUS             PRA6
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:12.00   1st Qu.:12.00   1st Qu.:88.00   1st Qu.:13.75
##  Median :14.00   Median :14.00   Median :88.00   Median :15.00
##  Mean   :13.07   Mean   :12.81   Mean   :73.87   Mean   :12.83
##  3rd Qu.:16.00   3rd Qu.:17.00   3rd Qu.:88.00   3rd Qu.:16.00
##  Max.   :20.00   Max.   :20.00   Max.   :88.00   Max.   :19.00
##  NA's   :46      NA's   :4658    NA's   :532     NA's   :5407
##       PAR
##  Min.   : 0.00
##  1st Qu.:10.00
##  Median :13.00
##  Mean   :11.98
##  3rd Qu.:15.00
##  Max.   :20.00
##  NA's   :105
##
## 'data.frame':    5338 obs. of  14 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ alu_cod : chr  "201512444" "201512430" "201512446" "201512425" ...
##  $ eva_fin : int  16 14 11 14 12 16 11 11 16 11 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1    : num  15 7 12 15 10 15 12 11 15 6 ...
##  $ FIN     : num  16 12 11 12 11 18 16 12 19 0 ...
##  $ PRA2    : num  13 13 14 11 10 13 14 11 15 12 ...
##  $ PRA4    : num  19 16 17 15 15 19 14 12 15 0 ...
##  $ PRA3    : num  15 16 11 17 10 15 0 8 15 14 ...
##  $ PRA5    : num  19 17 NA 17 NA 19 NA 0 NA NA ...
##  $ SUS     : num  88 88 88 88 NA 88 9 88 NA 7 ...
##  $ PRA6    : num  16 14 17 14 14 16 0 14 16 15 ...
##  $ PAR     : num  16 15 7 14 11 14 9 8 14 12 ...
##     semestre      alu_cod             eva_fin          veces
##  20171  :1188   Length:5338        Min.   : 0.00   Min.   :1.000
```

```
##  20181  :1181   Class :character   1st Qu.:12.00   1st Qu.:1.000
##  20182  : 812   Mode  :character   Median :13.00   Median :1.000
##  20172  : 771                      Mean   :13.12   Mean   :1.161
##  20161  : 665                      3rd Qu.:15.00   3rd Qu.:1.000
##  20162  : 510                      Max.   :20.00   Max.   :5.000
##  (Other): 211
##  aprobo         PRA1            FIN             PRA2            PRA4
##  No: 592   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  Si:4746   1st Qu.:11.00   1st Qu.:11.00   1st Qu.:12.00   1st Qu.:12.00
##            Median :14.00   Median :14.00   Median :14.00   Median :15.00
##            Mean   :12.85   Mean   :12.69   Mean   :13.15   Mean   :13.67
##            3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:16.00   3rd Qu.:17.00
##            Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##                            NA's   :102     NA's   :46      NA's   :46
##       PRA3            PRA5            SUS             PRA6
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:12.00   1st Qu.:12.00   1st Qu.:88.00   1st Qu.:14.00
##  Median :14.00   Median :14.00   Median :88.00   Median :15.00
##  Mean   :13.49   Mean   :13.32   Mean   :73.58   Mean   :13.09
##  3rd Qu.:16.00   3rd Qu.:17.00   3rd Qu.:88.00   3rd Qu.:16.00
##  Max.   :20.00   Max.   :20.00   Max.   :88.00   Max.   :19.00
##  NA's   :46      NA's   :4521    NA's   :514     NA's   :5240
##       PAR
##  Min.   : 0.00
##  1st Qu.:10.00
##  Median :13.00
##  Mean   :12.36
##  3rd Qu.:15.00
##  Max.   :20.00
##  NA's   :102
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1  FIN PRA2 PRA4 PRA3 PRA5  SUS PRA6  PAR
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12]  TRUE FALSE FALSE
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:7] 7 8 9 10 11 13 14
##  - attr(*, "names")= chr [1:7] "FIN" "PRA2" "PRA4" "PRA3" ...
## PRA5 PRA6
##   11   13
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0013*****"
##
## 'data.frame':    35668 obs. of  5 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1 .
..
##  $ eva_new : Factor w/ 7 levels "FIN","PAR","PRA1",..: 1 2 3 4 5 7 1 2 3 4
...
##  $ eva_nota: num  11 7 11 13 12 18 12 13 13 13 ...
##  $ alu_cod : chr  "201512430" "201512430" "201512430" "201512430" ...
##  $ eva_fin : int  14 14 14 14 14 14 12 12 12 12 ...
##     semestre    eva_new        eva_nota       alu_cod
##  20181  :8041   FIN :5873   Min.   : 0.00   Length:35668
##  20171  :7616   PAR :5873   1st Qu.:10.00   Class :character
```

```
##  20182  :6096   PRA1:6035   Median :14.00   Mode  :character
##  20172  :5686   PRA2:6036   Mean   :21.52
##  20161  :4056   PRA3:6036   3rd Qu.:17.00
##  20162  :3378   PRA4: 163   Max.   :88.00
##  (Other): 795   SUS :5652
##     eva_fin
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :15.54
##  3rd Qu.:15.00
##  Max.   :99.00
##
## 'data.frame':    6036 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 2 2 2 .
..
##  $ alu_cod : chr  "201512430" "201512425" "201512447" "201512471" ...
##  $ eva_fin : int  14 12 14 14 14 14 18 16 13 12 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ estado  : Factor w/ 3 levels "A","D","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRA1    : num  11 13 18 8 10 17 17 15 10 11 ...
##  $ FIN     : num  11 12 12 13 16 11 18 17 14 12 ...
##  $ PRA2    : num  13 13 12 20 15 17 20 17 15 11 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  12 11 15 20 15 16 18 18 13 14 ...
##  $ SUS     : num  18 88 14 88 88 88 88 88 88 NA ...
##  $ PAR     : num  7 13 0 14 14 13 19 14 11 11 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1346   Length:6036        Min.   : 0.00   Min.   :1.000
##  20171  :1282   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20182  :1016   Mode  :character   Median :13.00   Median :1.000
##  20172  : 954                      Mean   :15.55   Mean   :1.335
##  20161  : 676                      3rd Qu.:15.00   3rd Qu.:1.000
##  20162  : 575                      Max.   :99.00   Max.   :6.000
##  (Other): 187
##  estado        PRA1            FIN             PRA2            PRA4
##  A:4782   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  D:1019   1st Qu.: 7.00   1st Qu.: 9.00   1st Qu.: 9.00   1st Qu.: 9.50
##  N: 235   Median :12.00   Median :12.00   Median :14.00   Median :13.00
##           Mean   :11.26   Mean   :11.06   Mean   :11.93   Mean   :11.43
##           3rd Qu.:16.00   3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:16.00
##           Max.   :20.00   Max.   :20.00   Max.   :20.00   Max.   :20.00
##           NA's   :1       NA's   :163                     NA's   :5873
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:12.00   1st Qu.:88.00   1st Qu.: 8.00
##  Median :15.00   Median :88.00   Median :12.00
##  Mean   :13.15   Mean   :73.93   Mean   :10.86
##  3rd Qu.:17.00   3rd Qu.:88.00   3rd Qu.:14.00
##  Max.   :20.00   Max.   :88.00   Max.   :20.00
##                  NA's   :384     NA's   :163
##
## 'data.frame':    5801 obs. of  12 variables:
##  $ semestre: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 2 2 2 .
..
##  $ alu_cod : chr  "201512430" "201512425" "201512447" "201512471" ...
##  $ eva_fin : int  14 12 14 14 14 14 18 16 13 12 ...
##  $ veces   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ PRA1    : num  11 13 18 8 10 17 17 15 10 11 ...
##  $ FIN     : num  11 12 12 13 16 11 18 17 14 12 ...
##  $ PRA2    : num  13 13 12 20 15 17 20 17 15 11 ...
##  $ PRA4    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PRA3    : num  12 11 15 20 15 16 18 18 13 14 ...
##  $ SUS     : num  18 88 14 88 88 88 88 88 88 NA ...
##  $ PAR     : num  7 13 0 14 14 13 19 14 11 11 ...
##     semestre      alu_cod             eva_fin          veces
##  20181  :1287   Length:5801        Min.   : 0.00   Min.   :1.000
##  20171  :1234   Class :character   1st Qu.:11.00   1st Qu.:1.000
##  20182  : 959   Mode  :character   Median :13.00   Median :1.000
##  20172  : 910                      Mean   :12.17   Mean   :1.303
##  20161  : 666                      3rd Qu.:15.00   3rd Qu.:1.000
##  20162  : 570                      Max.   :20.00   Max.   :6.000
##  (Other): 175
##  aprobo         PRA1            FIN            PRA2            PRA4
##  No:1019   Min.   : 0.00   Min.   : 0.0   Min.   : 0.00   Min.   : 0.00
##  Si:4782   1st Qu.: 8.00   1st Qu.: 9.0   1st Qu.:10.00   1st Qu.:11.00
##            Median :13.00   Median :12.0   Median :14.00   Median :13.00
##            Mean   :11.71   Mean   :11.5   Mean   :12.41   Mean   :12.18
##            3rd Qu.:16.00   3rd Qu.:15.0   3rd Qu.:16.00   3rd Qu.:16.00
##            Max.   :20.00   Max.   :20.0   Max.   :20.00   Max.   :20.00
##            NA's   :1       NA's   :153                    NA's   :5648
##       PRA3            SUS             PAR
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:13.00   1st Qu.:88.00   1st Qu.: 9.00
##  Median :15.00   Median :88.00   Median :12.00
##  Mean   :13.69   Mean   :73.34   Mean   :11.29
##  3rd Qu.:17.00   3rd Qu.:88.00   3rd Qu.:15.00
##  Max.   :20.00   Max.   :88.00   Max.   :20.00
##                  NA's   :373     NA's   :153
##
## [1] "Verificacion de cantidad de Registro..."
## PRA1  FIN PRA2 PRA4 PRA3  SUS  PAR
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##
## [1] "Verificando si existe examen sustitutorio..."
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALS
E
##
## [1] "Verificacion de valores perdidos (NA)..."
##  Named int [1:4] 6 7 9 12
##  - attr(*, "names")= chr [1:4] "PRA1" "FIN" "PRA4" "PAR"
## PRA4
##    9
## [1] "***** Fin de Ejecución *****"
```

Anexo 13: Resultados de la creación del dataset de cada curso del Programa de Estudios Básicos (PEB).

Procederemos a generar los Dataset a partir de los Archivos de notas de cada curso del Programa de Estudios Básicos (PEB).

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")
paleta <- colorRampPalette(c("dodgerblue", "white", "dodgerblue4"))

# Uso de Librerías
library(dplyr)
library(VIM) # imputacion
library(DMwR) # Densidad Local
library(caret) # nzv
library(corrplot)
```

Para la creación del dataset del Curso con código: "0001" sirvase revisar el Anexo 08.

---

1. Creación de los datasets de un solo curso.

Generación de la estructura interna del segundo dataset.
Formado por:
1. *"Taller de Método de Estudio Universitario", con código: "0002"*

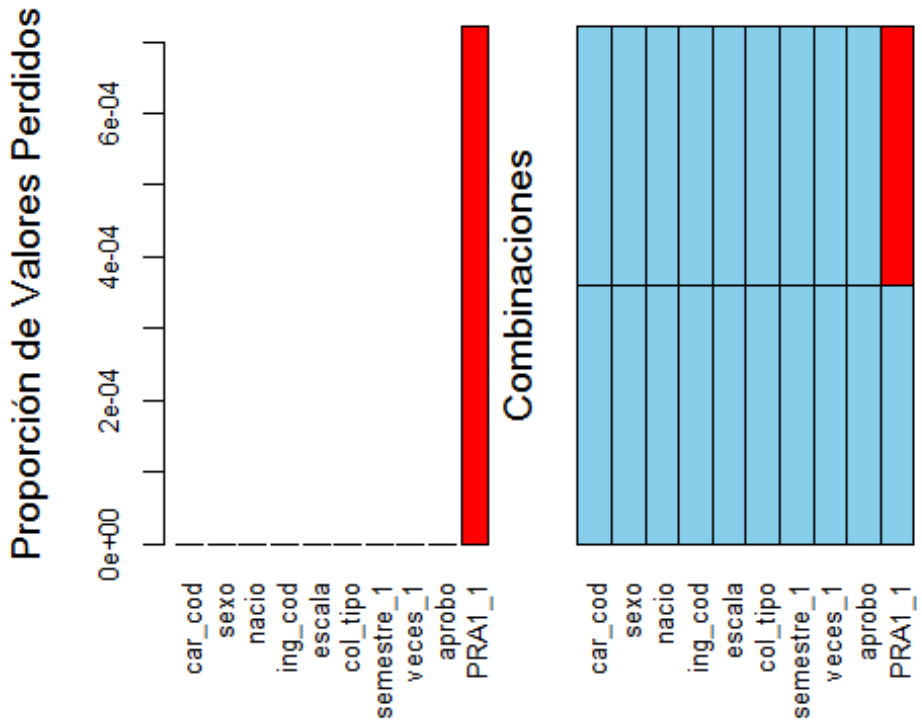
Generación de la estructura interna del tercer dataset.
Formado por:
1. *"Taller de Comunicación Oral y Escrita I", con código: "0003"*

Generación de la estructura interna del cuarto dataset.
Formado por:
1. *"Matemática", con código: "0004"*

Generación de la estructura interna del quinto dataset.
Formado por:
1. *"Inglés I", con código: "0005"*

```
## 'data.frame':    9710 obs. of  10 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
```

```
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1    : int  13 16 17 17 18 15 15 14 9 11 ...

##     car_cod     sexo         nacio                ing_cod     escala
##  11     :1421   F:4558   Min.   :1957-03-10   13     :3550   A13: 893
##  61     :1348   M:5152   1st Qu.:1997-06-02   15     :3212   A18: 139
##  25     :1032            Median :1998-09-16   17     :1359   A23:5852
##  63     : 947            Mean   :1998-05-05   04     : 425   A28: 276
##  51     : 686            3rd Qu.:1999-12-09   05     : 331   A33:2210
##  41     : 604            Max.   :2004-05-03   08     : 228   A38: 340
##  (Other):3672                                 (Other): 605
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
##  E:2060   20181  :1543   Min.   :1.000   No:1729   Min.   : 0.00
##  P:7650   20161  :1499   1st Qu.:1.000   Si:7981   1st Qu.:11.00
##           20171  :1470   Median :1.000             Median :13.00
##           20151  :1294   Mean   :1.285             Mean   :12.61
##           20172  : 991   3rd Qu.:1.000             3rd Qu.:16.00
##           20152  : 990   Max.   :8.000             Max.   :20.00
##           (Other):1923                             NA's   :7

## [1] 7

## [1] 0.07209063

##  Named int 10
##  - attr(*, "names")= chr "PRA1_1"

## [1] 0.07209063

## named integer(0)

## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```



```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##     veces_1     0
##      aprobo     0
##      PRA1_1     7
##
##  Missings in combinations of variables:
##         Combinations Count     Percent
##  0:0:0:0:0:0:0:0:0:0  9703 99.92790937
##  0:0:0:0:0:0:0:0:0:1     7  0.07209063
```

```
## [1] NA NA NA NA NA NA NA
```

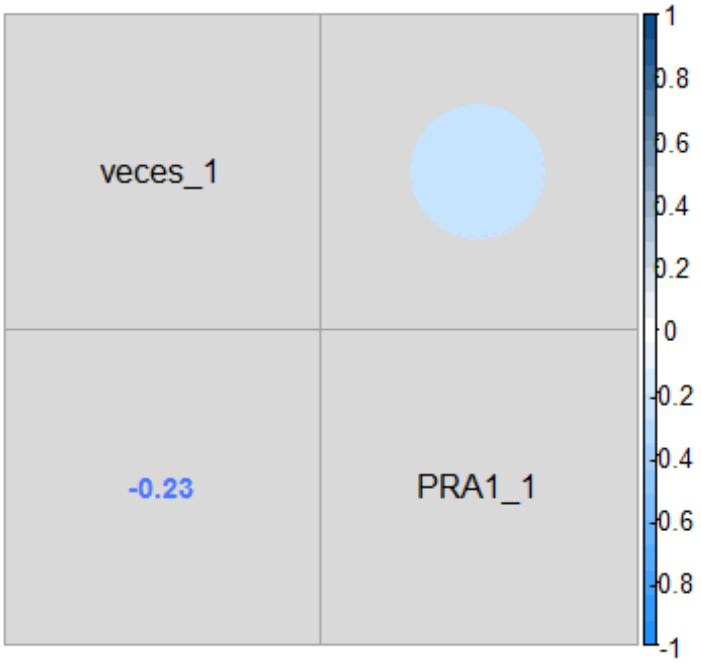
```
## [1] 14 15 12 14 12 13 12
```

```
##     car_cod     sexo         nacio                ing_cod     escala
##  11     :1421   F:4558   Min.   :1957-03-10   13     :3550   A13: 893
##  61     :1348   M:5152   1st Qu.:1997-06-02   15     :3212   A18: 139
##  25     :1032            Median :1998-09-16   17     :1359   A23:5852
##  63     : 947            Mean   :1998-05-05   04     : 425   A28: 276
##  51     : 686            3rd Qu.:1999-12-09   05     : 331   A33:2210
##  41     : 604            Max.   :2004-05-03   08     : 228   A38: 340
##  (Other):3672                                 (Other): 605
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
##  E:2060   20181  :1543   Min.   :1.000   No:1729   Min.   : 0.00
##  P:7650   20161  :1499   1st Qu.:1.000   Si:7981   1st Qu.:11.00
##           20171  :1470   Median :1.000             Median :13.00
##           20151  :1294   Mean   :1.285             Mean   :12.61
##           20172  : 991   3rd Qu.:1.000             3rd Qu.:16.00
##           20152  : 990   Max.   :8.000             Max.   :20.00
##           (Other):1923
```

```
## [1] 82.19361
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##            veces_1     PRA1_1
## veces_1  1.0000000 -0.2251368
## PRA1_1  -0.2251368  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.2251 -0.2251 -0.2251 -0.2251 -0.2251 -0.2251

## [1] 0

## character(0)

## integer(0)

## character(0)
```

```
## 'data.frame':    9547 obs. of  10 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1    : int  19 17 11 13 15 13 13 14 12 13 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  11     :1384   F:4526   Min.   :1959-08-01   13     :3474   A13: 886
##  61     :1329   M:5021   1st Qu.:1997-06-24   15     :3191   A18: 141
##  25     :1027            Median :1998-09-27   17     :1370   A23:5737
##  63     : 947            Mean   :1998-05-27   04     : 424   A28: 279
##  51     : 664            3rd Qu.:1999-12-22   05     : 326   A33:2180
##  41     : 597            Max.   :2004-05-03   08     : 218   A38: 324
##  (Other):3599                                 (Other): 544
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
```

```
##  E:2019   20181  :1525   Min.   :1.000   No:1604   Min.   : 0.00
##  P:7528   20161  :1488   1st Qu.:1.000   Si:7943   1st Qu.:12.00
##           20171  :1444   Median :1.000             Median :14.00
##           20151  :1290   Mean   :1.257             Mean   :13.31
##           20172  : 976   3rd Qu.:1.000             3rd Qu.:16.00
##           20152  : 959   Max.   :9.000             Max.   :20.00
##           (Other):1865                             NA's   :2
```

```
## [1] 2
```

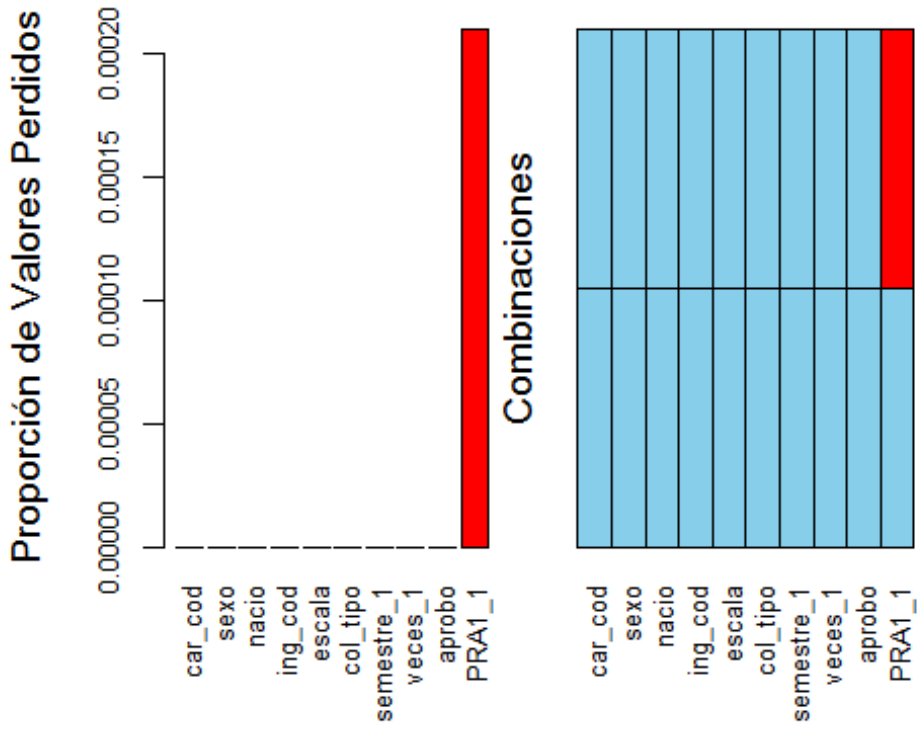
```
## [1] 0.02094899
```

```
##  Named int 10
##  - attr(*, "names")= chr "PRA1_1"
```

```
## [1] 0.02094899
```

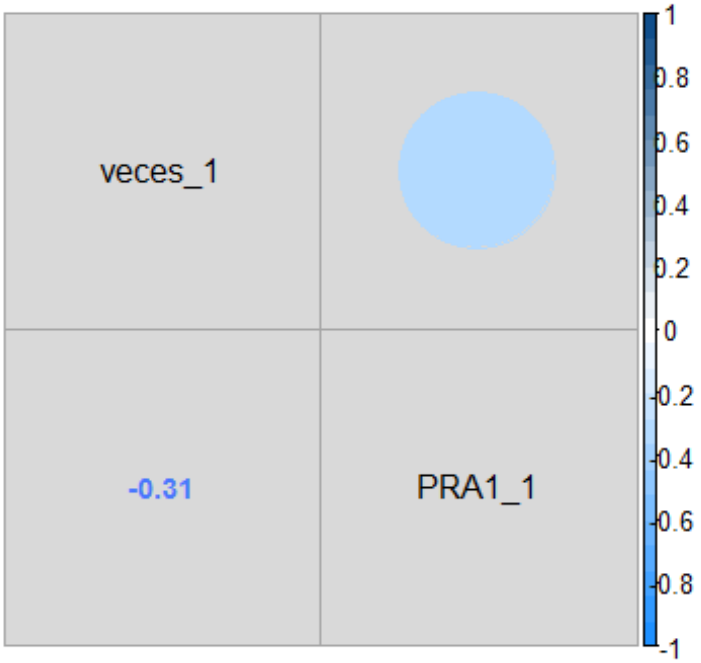
```
## named integer(0)
```

```
## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```




```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##     veces_1     0
##      aprobo     0
##      PRA1_1     2
##
```

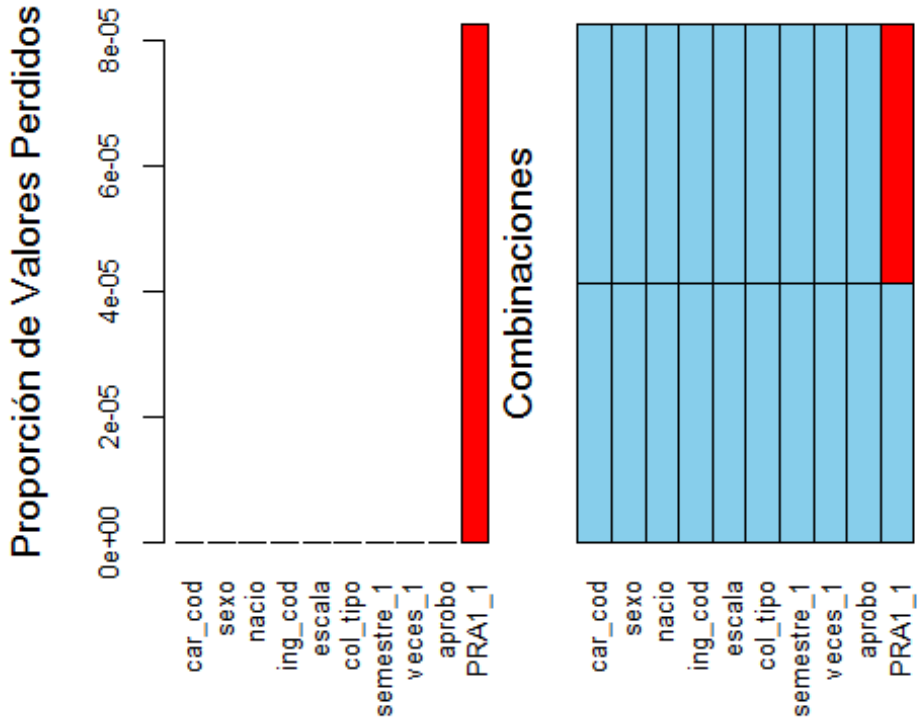
```
##  Missings in combinations of variables:
##         Combinations Count     Percent
##  0:0:0:0:0:0:0:0:0:0  9545 99.97905101
##  0:0:0:0:0:0:0:0:0:1     2  0.02094899

## [1] NA NA

## [1] 17 17

##     car_cod     sexo         nacio                ing_cod     escala
##  11     :1384   F:4526   Min.   :1959-08-01   13     :3474   A13: 886
##  61     :1329   M:5021   1st Qu.:1997-06-24   15     :3191   A18: 141
##  25     :1027            Median :1998-09-27   17     :1370   A23:5737
##  63     : 947            Mean   :1998-05-27   04     : 424   A28: 279
##  51     : 664            3rd Qu.:1999-12-22   05     : 326   A33:2180
##  41     : 597            Max.   :2004-05-03   08     : 218   A38: 324
##  (Other):3599                                 (Other): 544
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
##  E:2019   20181  :1525   Min.   :1.000   No:1604   Min.   : 0.00
##  P:7528   20161  :1488   1st Qu.:1.000   Si:7943   1st Qu.:12.00
##           20171  :1444   Median :1.000             Median :14.00
##           20151  :1290   Mean   :1.257             Mean   :13.31
##           20172  : 976   3rd Qu.:1.000             3rd Qu.:16.00
##           20152  : 959   Max.   :9.000             Max.   :20.00
##           (Other):1865

## [1] 83.19891

##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

##             veces_1     PRA1_1
## veces_1   1.0000000 -0.3083785
## PRA1_1   -0.3083785  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.3084 -0.3084 -0.3084 -0.3084 -0.3084 -0.3084

## [1] 0
```

```
## character(0)

## integer(0)

## character(0)
```

```
## 'data.frame':    12102 obs. of  10 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 1 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 2 1
...
##  $ veces_1   : int  1 1 1 1 1 1 1 2 2 2 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 1 2 1 ...
##  $ PRA1_1    : int  14 17 15 19 6 12 15 12 10 0 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  11     :1589   F:5928   Min.   :1959-08-06   13     :4489   A13:1005
##  61     :1499   M:6174   1st Qu.:1997-07-06   15     :3814   A18: 141
##  63     :1197            Median :1998-08-25   17     :1911   A23:7330
##  25     :1146            Mean   :1998-05-23   04     : 536   A28: 286
##  41     : 881            3rd Qu.:1999-11-02   05     : 427   A33:3129
##  51     : 828            Max.   :2004-05-03   08     : 265   A38: 211
##  (Other):4962                                 (Other): 660
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
##  E:2522   20181  :1818   Min.   :1.000   No:5098   Min.   : 0.00
##  P:9580   20161  :1731   1st Qu.:1.000   Si:7004   1st Qu.: 6.00
##           20171  :1676   Median :2.000             Median :11.00
##           20172  :1447   Mean   :1.905             Mean   :10.28
##           20162  :1334   3rd Qu.:2.000             3rd Qu.:14.00
##           20152  :1323   Max.   :8.000             Max.   :20.00
##           (Other):2773                             NA's   :1

## [1] 1

## [1] 0.008263097

##  Named int 10
##  - attr(*, "names")= chr "PRA1_1"

## [1] 0.008263097

## named integer(0)

## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```




```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##     veces_1     0
##      aprobo     0
##      PRA1_1     1
##
##  Missings in combinations of variables:
##         Combinations Count      Percent
##  0:0:0:0:0:0:0:0:0:0 12101 99.991736903
##  0:0:0:0:0:0:0:0:0:1     1  0.008263097
```

```
## [1] NA
```

```
## [1] 12
```

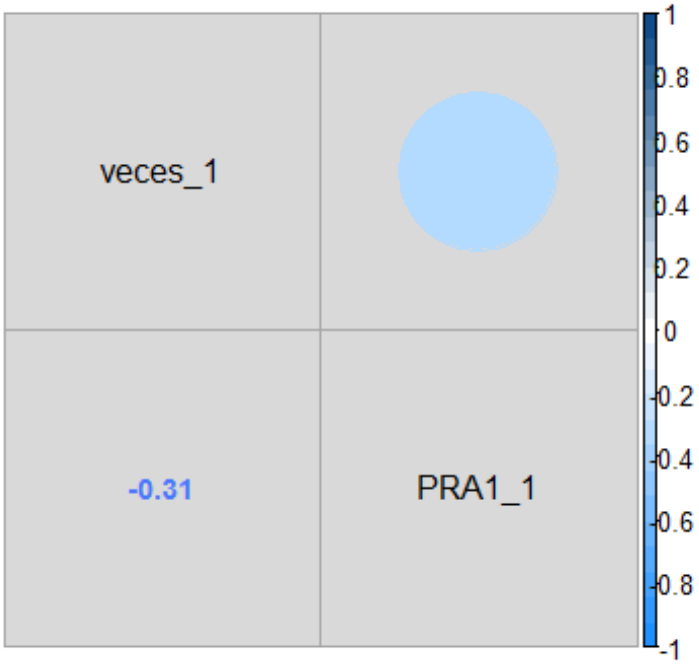
```
##     car_cod     sexo         nacio                ing_cod     escala
##  11     :1589   F:5928   Min.   :1959-08-06   13     :4489   A13:1005
##  61     :1499   M:6174   1st Qu.:1997-07-06   15     :3814   A18: 141
##  63     :1197            Median :1998-08-25   17     :1911   A23:7330
##  25     :1146            Mean   :1998-05-23   04     : 536   A28: 286
##  41     : 881            3rd Qu.:1999-11-02   05     : 427   A33:3129
##  51     : 828            Max.   :2004-05-03   08     : 265   A38: 211
##  (Other):4962                                 (Other): 660
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
##  E:2522   20181  :1818   Min.   :1.000   No:5098   Min.   : 0.00
##  P:9580   20161  :1731   1st Qu.:1.000   Si:7004   1st Qu.: 6.00
##           20171  :1676   Median :2.000             Median :11.00
```

```
##           20172  :1447   Mean   :1.905            Mean   :10.28
##           20162  :1334   3rd Qu.:2.000            3rd Qu.:14.00
##           20152  :1323   Max.   :8.000            Max.   :20.00
##           (Other):2773
```

```
## [1] 57.87473
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##            veces_1     PRA1_1
## veces_1  1.0000000 -0.3135921
## PRA1_1  -0.3135921  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.3136 -0.3136 -0.3136 -0.3136 -0.3136 -0.3136
```

```
## [1] 0
```

```
## character(0)
```

```
## integer(0)
```

```
## character(0)
```

```
## 'data.frame':    6092 obs. of  10 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 1 1 1 1 1 1 1 1 11
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 2 1 1 1 2 1 ...
##  $ nacio     : Date, format: "1998-07-21" "1997-12-16" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 3 3 3 3 3 3 3 3
5 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 1 2 2 2 2 2 ...
##  $ semestre_1: Factor w/ 9 levels "20161","20162",..: 4 5 5 5 5 4 5 4 5 4
...
```

```
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1    : int  16 15 14 15 20 12 20 9 13 14 ...
```

```
##     car_cod     sexo         nacio                ing_cod     escala
##  61     :1080   F:2678   Min.   :1959-08-01   13     :2221   A13:   3
##  11     :1034   M:3414   1st Qu.:1997-10-10   15     :1949   A18:   4
##  63     : 753            Median :1999-02-18   17     : 964   A23:4077
##  41     : 499            Mean   :1998-08-21   04     : 252   A28:  79
##  32     : 468            3rd Qu.:2000-02-26   05     : 177   A33:1613
##  34     : 425            Max.   :2004-05-03   08     : 122   A38: 316
##  (Other):1833                                 (Other): 407
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
##  E:1340   20181  :1367   Min.   :1.000   No:1353   Min.   : 0.0
##  P:4752   20171  :1261   1st Qu.:1.000   Si:4739   1st Qu.: 9.0
##           20172  : 957   Median :1.000             Median :13.0
##           20161  : 891   Mean   :1.296             Mean   :12.4
##           20162  : 789   3rd Qu.:1.000             3rd Qu.:16.0
##           20182  : 687   Max.   :5.000             Max.   :20.0
##           (Other): 140                             NA's   :2
```

```
## [1] 2
```

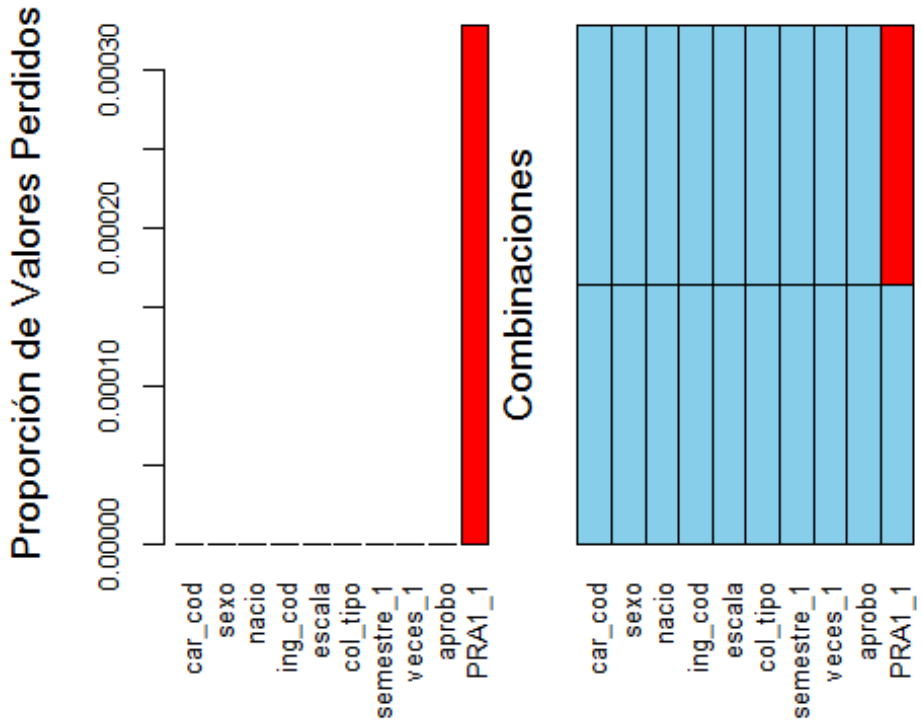
```
## [1] 0.03282994
```

```
##  Named int 10
##  - attr(*, "names")= chr "PRA1_1"
```

```
## [1] 0.03282994
```

```
## named integer(0)
```

```
## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```



```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##     veces_1     0
##      aprobo     0
##      PRA1_1     2
##
##  Missings in combinations of variables:
##         Combinations Count     Percent
##  0:0:0:0:0:0:0:0:0:0  6090 99.96717006
##  0:0:0:0:0:0:0:0:0:1     2  0.03282994
```

```
## [1] NA NA
```

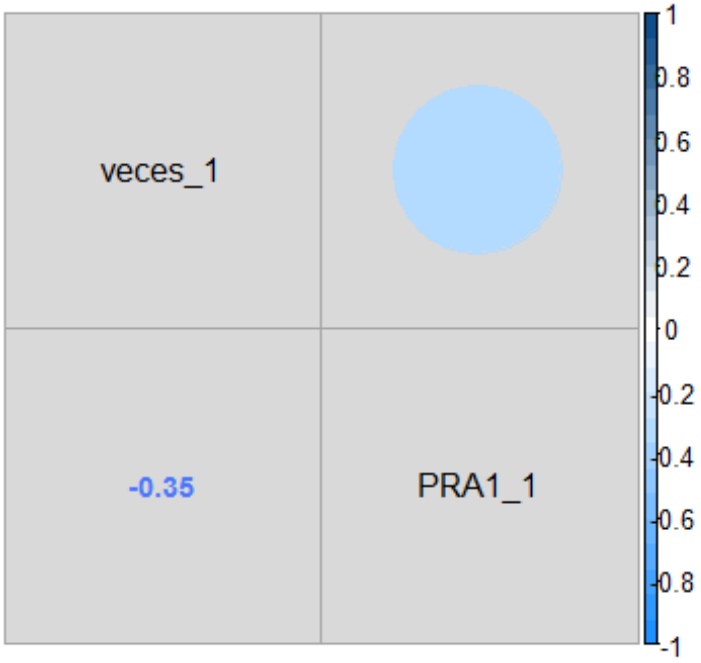
```
## [1] 14 17
```

```
##     car_cod     sexo         nacio                ing_cod     escala
##  61     :1080   F:2678   Min.   :1959-08-01   13     :2221   A13:   3
##  11     :1034   M:3414   1st Qu.:1997-10-10   15     :1949   A18:   4
##  63     : 753            Median :1999-02-18   17     : 964   A23:4077
##  41     : 499            Mean   :1998-08-21   04     : 252   A28:  79
##  32     : 468            3rd Qu.:2000-02-26   05     : 177   A33:1613
##  34     : 425            Max.   :2004-05-03   08     : 122   A38: 316
##  (Other):1833                                 (Other): 407
##  col_tipo   semestre_1      veces_1      aprobo        PRA1_1
##  E:1340   20181  :1367   Min.   :1.000   No:1353   Min.   : 0.0
##  P:4752   20171  :1261   1st Qu.:1.000   Si:4739   1st Qu.: 9.0
##           20172  : 957   Median :1.000             Median :13.0
##           20161  : 891   Mean   :1.296             Mean   :12.4
##           20162  : 789   3rd Qu.:1.000             3rd Qu.:16.0
##           20182  : 687   Max.   :5.000             Max.   :20.0
##           (Other): 140
```

```
## [1] 77.79054
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##            veces_1     PRA1_1
## veces_1  1.0000000 -0.3540495
## PRA1_1  -0.3540495  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -0.354  -0.354  -0.354  -0.354  -0.354  -0.354

## [1] 0

## character(0)

## integer(0)

## character(0)
```

---

2. Creación de los datasets de dos cursos.

Generación de la estructura interna del sexto dataset.
Formado por:
1. *"Taller de Comunicación Oral y Escrita I", con código: "0003"*
2. *"Psicología General", con código: "0006"*

Generación de la estructura interna del septimo dataset.
Formado por:
1. *"Taller de Método de Estudio Universitario", con código: "0002"*
2. *"Lógica y Filosofía", con código: "0007"*

Generación de la estructura interna del octavo dataset.
Formado por:
1. *"Taller de Comunicación Oral y Escrita I", con código: "0003"*
2. *"Taller de Comunicación Oral y Escrita II", con código: "0008"*

Generación de la estructura interna del noveno dataset.
Formado por:
1. *"Inglés I", con código: "0005"*

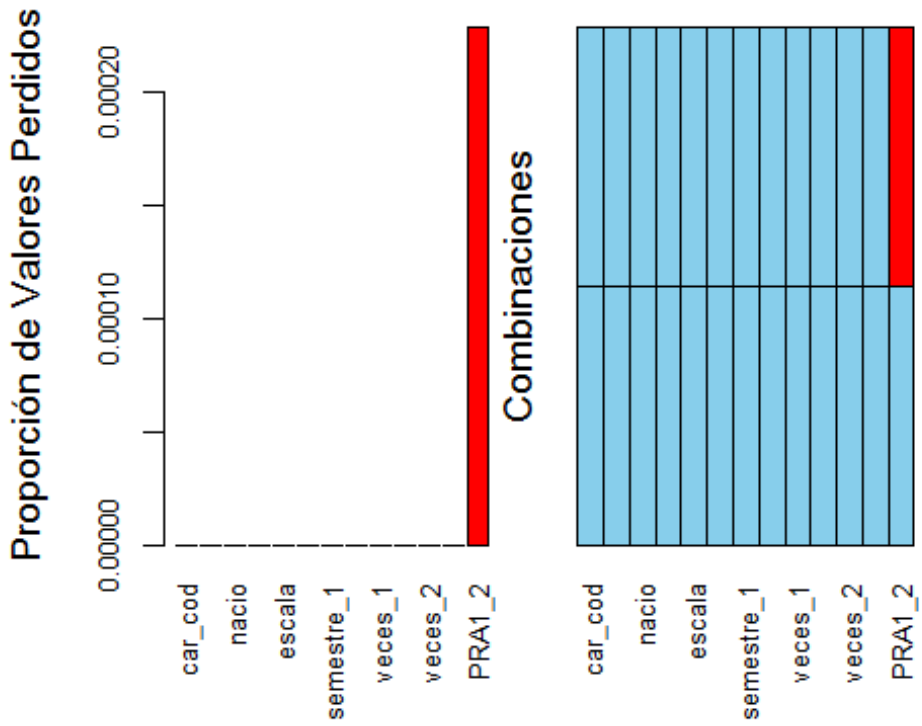
2. *“Inglés II”, con código: “0009”*

Generación de la estructura interna del décimo dataset.
Formado por:
1. *“Taller de Método de Estudio Universitario”, con código: “0002”*
2. *“Formación Histórica del Perú”, con código: “0010”*

Generación de la estructura interna del décimo primero dataset.
Formado por:
1. *“Actividades Artísticas y Deportivas”, con código: “0001”*
2. *“Recursos Naturales y Medio Ambiente”, con código: “0011”*

```
## 'data.frame':    8748 obs. of  13 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ eva_fin_1 : int  14 16 14 12 14 14 13 14 12 14 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 12 levels "20151","20152",..: 6 2 2 2 2 2 7 2 4 2
...
##  $ veces_2   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2    : int  17 20 14 15 13 6 5 9 12 11 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  61     :1226   F:4230   Min.   :1959-08-06   13     :3098   A13: 875
##  11     :1183   M:4518   1st Qu.:1997-05-28   15     :2834   A18:   5
##  63     : 888            Median :1998-07-17   17     :1445   A23:5530
##  25     : 880            Mean   :1998-04-10   04     : 340   A28:  18
##  41     : 601            3rd Qu.:1999-09-14   05     : 317   A33:2261
##  32     : 588            Max.   :2004-05-03   08     : 195   A38:  59
##  (Other):3382                                 (Other): 519
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E:1800   20161  :1643   Min.   : 0.00   Min.   :1.000   20182  :1657
##  P:6948   20171  :1522   1st Qu.:11.00   1st Qu.:1.000   20172  :1561
##           20151  :1348   Median :13.00   Median :1.000   20162  :1456
##           20181  :1199   Mean   :12.92   Mean   :1.336   20181  :1013
##           20162  : 938   3rd Qu.:15.00   3rd Qu.:1.000   20171  : 910
##           20152  : 932   Max.   :20.00   Max.   :9.000   20152  : 899
##           (Other):1166                                   (Other):1252
##     veces_2      aprobo        PRA1_2
##  Min.   :1.000   No:2378   Min.   : 0.00
##  1st Qu.:1.000   Si:6370   1st Qu.: 8.00
##  Median :1.000             Median :12.00
##  Mean   :1.513             Mean   :11.09
##  3rd Qu.:2.000             3rd Qu.:15.00
##  Max.   :6.000             Max.   :20.00
##                            NA's   :2
```

```
## [1] 2

## [1] 0.02286237

##  Named int 13
##  - attr(*, "names")= chr "PRA1_2"

## [1] 0.02286237

## named integer(0)

## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```




```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##     veces_2     0
##      aprobo     0
##      PRA1_2     2
##
##  Missings in combinations of variables:
##                Combinations Count     Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0  8746 99.97713763
##  0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1     2  0.02286237
```

```
## [1] NA NA
```

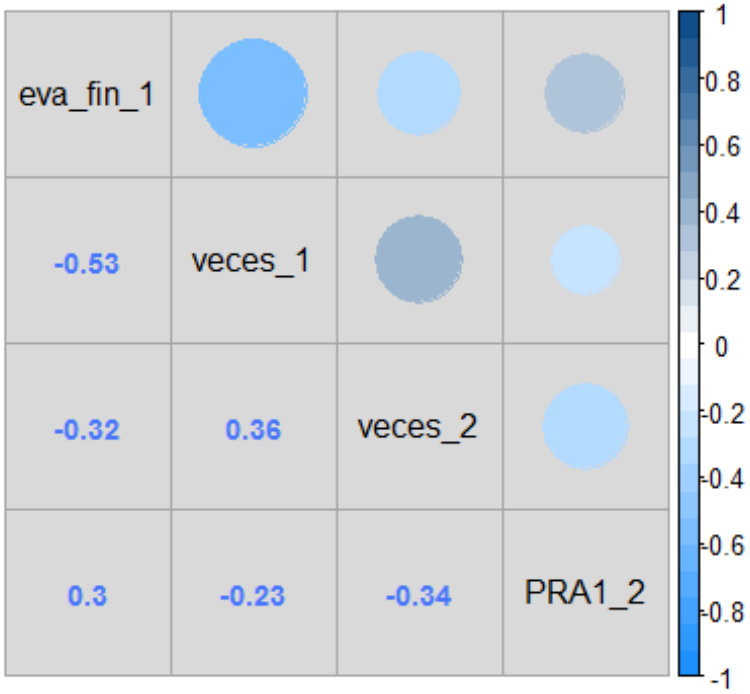
```
## [1] 7 9
```

```
##     car_cod     sexo         nacio                ing_cod     escala    
##  61     :1226   F:4230   Min.   :1959-08-06   13     :3098   A13: 875  
##  11     :1183   M:4518   1st Qu.:1997-05-28   15     :2834   A18:   5  
##  63     : 888            Median :1998-07-17   17     :1445   A23:5530  
##  25     : 880            Mean   :1998-04-10   04     : 340   A28:  18  
##  41     : 601            3rd Qu.:1999-09-14   05     : 317   A33:2261  
##  32     : 588            Max.   :2004-05-03   08     : 195   A38:  59  
##  (Other):3382                                 (Other): 519             
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2   
##  E:1800   20161  :1643   Min.   : 0.00   Min.   :1.000   20182  :1657  
##  P:6948   20171  :1522   1st Qu.:11.00   1st Qu.:1.000   20172  :1561  
##           20151  :1348   Median :13.00   Median :1.000   20162  :1456  
##           20181  :1199   Mean   :12.92   Mean   :1.336   20181  :1013  
##           20162  : 938   3rd Qu.:15.00   3rd Qu.:1.000   20171  : 910  
##           20152  : 932   Max.   :20.00   Max.   :9.000   20152  : 899  
##           (Other):1166                                   (Other):1252  
##     veces_2      aprobo        PRA1_2      
##  Min.   :1.000   No:2378   Min.   : 0.00  
##  1st Qu.:1.000   Si:6370   1st Qu.: 8.00  
##  Median :1.000             Median :12.00  
##  Mean   :1.513             Mean   :11.09  
##  3rd Qu.:2.000             3rd Qu.:15.00  
##  Max.   :6.000             Max.   :20.00  
##
```

```
## [1] 72.81664
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

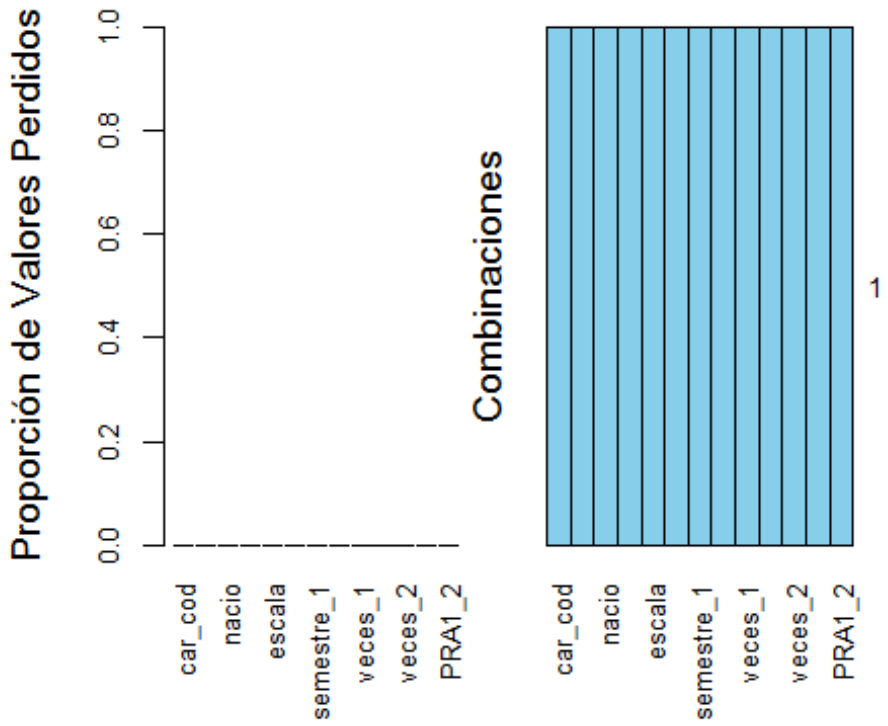
```
##            eva_fin_1    veces_1    veces_2     PRA1_2
## eva_fin_1  1.0000000 -0.5332684 -0.3163523  0.3004613
## veces_1   -0.5332684  1.0000000  0.3600006 -0.2288307
## veces_2   -0.3163523  0.3600006  1.0000000 -0.3441378
## PRA1_2     0.3004613 -0.2288307 -0.3441378  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.5333 -0.3372 -0.2726 -0.1270  0.1681  0.3600

## [1] 0

## character(0)

## integer(0)

## character(0)
```

```
## 'data.frame':    8525 obs. of  13 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 1 7 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 2 1 1 1 1 2 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-07-21" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 3 5 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 2 1 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ eva_fin_1 : int  15 15 15 15 17 18 15 14 15 15 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 12 levels "20151","20152",..: 2 4 2 2 2 2 2 2 2 2
...
##  $ veces_2   : int  2 2 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 1 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2    : int  7 10 10 11 14 10 15 13 7 7 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  11     :1285   F:4079   Min.   :1957-03-10   13     :3111   A13: 858
##  61     :1208   M:4446   1st Qu.:1997-05-14   15     :2676   A18:   7
##  25     : 865            Median :1998-07-28   17     :1349   A23:5382
```

```
##  63     : 831           Mean   :1998-03-24   04     : 340   A28:  23
##  41     : 593           3rd Qu.:1999-09-07   05     : 307   A33:2184
##  32     : 570           Max.   :2004-05-03   08     : 202   A38:  71
##  (Other):3173                                (Other): 540
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E:1742   20161  :1607   Min.   : 0.00   Min.   :1.000   20182  :1580
##  P:6783   20171  :1495   1st Qu.:12.00   1st Qu.:1.000   20172  :1473
##           20151  :1267   Median :14.00   Median :1.000   20162  :1388
##           20181  :1152   Mean   :13.12   Mean   :1.335   20181  :1060
##           20162  : 984   3rd Qu.:15.00   3rd Qu.:1.000   20171  : 914
##           20152  : 900   Max.   :19.00   Max.   :8.000   20152  : 842
##           (Other):1120                                   (Other):1268
##     veces_2      aprobo        PRA1_2
##  Min.   :1.000   No:2201   Min.   : 0.00
##  1st Qu.:1.000   Si:6324   1st Qu.: 9.00
##  Median :1.000             Median :12.00
##  Mean   :1.482             Mean   :11.74
##  3rd Qu.:2.000             3rd Qu.:15.00
##  Max.   :6.000             Max.   :20.00
##

## [1] 0

## [1] 0

##  Named int(0)
##  - attr(*, "names")= chr(0)

## numeric(0)

## named integer(0)
```



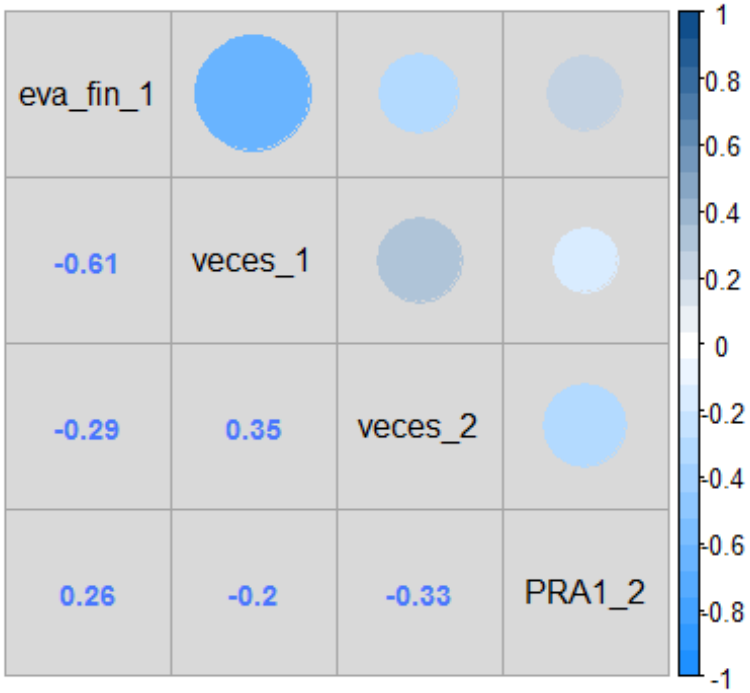

```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
```

```
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##     veces_2     0
##      aprobo     0
##      PRA1_2     0
##
##  Missings in combinations of variables:
##               Combinations Count Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0  8525     100
```

```
## [1] 74.18182
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##            eva_fin_1    veces_1    veces_2     PRA1_2
## eva_fin_1  1.0000000 -0.6120799 -0.2887470  0.2645027
## veces_1   -0.6120799  1.0000000  0.3454899 -0.1992531
## veces_2   -0.2887470  0.3454899  1.0000000 -0.3270044
## PRA1_2     0.2645027 -0.1992531 -0.3270044  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.6121 -0.3174 -0.2440 -0.1362  0.1486  0.3455
```

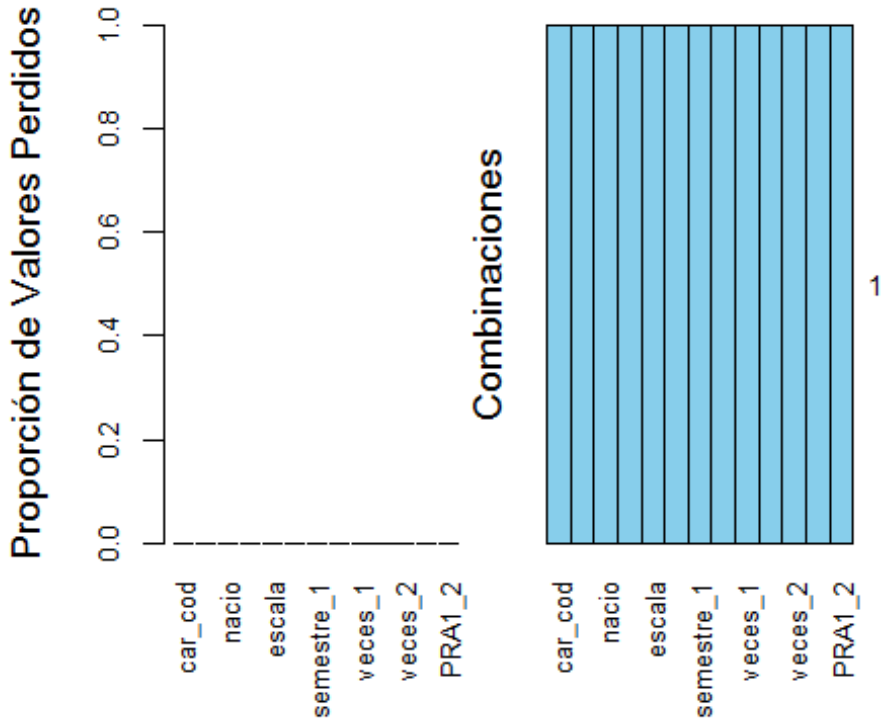
```
## [1] 0
```

```
## character(0)
```

```
## integer(0)
```

```
## character(0)
```

```
## 'data.frame':    5811 obs. of  13 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 1 1 1 1 1 1 1 1 11
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 2 1 1 1 2 1 ...
##  $ nacio     : Date, format: "1998-07-21" "1997-12-16" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 3 3 3 3 3 3 3 3
5 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 1 2 2 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ eva_fin_1 : int  14 14 12 14 14 13 14 12 14 11 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 8 levels "20162","20170",..: 4 4 4 4 4 3 4 3 4 4
...
##  $ veces_2   : int  1 1 1 1 1 1 1 1 1 3 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 1 ...
##  $ PRA1_2    : int  9 18 14 11 14 12 11 12 13 10 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  11     :1127   F:2922   Min.   :1959-08-06   13     :2166   A13: 595
##  61     : 783   M:2889   1st Qu.:1997-10-05   15     :1718   A18:   6
##  25     : 601            Median :1999-01-08   17     : 949   A23:3760
##  63     : 493            Mean   :1998-07-30   04     : 272   A28:  23
##  41     : 384            3rd Qu.:1999-12-21   05     : 210   A33:1375
##  32     : 377            Max.   :2004-05-03   08     : 152   A38:  52
##  (Other):2046                                 (Other): 344
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E:1138   20161  :1375   Min.   : 0.00   Min.   :1.000   20172  :1416
##  P:4673   20171  :1260   1st Qu.:12.00   1st Qu.:1.000   20182  :1398
##           20181  :1020   Median :13.00   Median :1.000   20162  : 986
##           20162  : 748   Mean   :13.43   Mean   :1.174   20181  : 911
##           20172  : 728   3rd Qu.:15.00   3rd Qu.:1.000   20171  : 892
##           20152  : 318   Max.   :20.00   Max.   :5.000   20170  :  71
##           (Other): 362                                   (Other): 137
##     veces_2      aprobo        PRA1_2
##  Min.   :1.000   No: 761   Min.   : 0.00
##  1st Qu.:1.000   Si:5050   1st Qu.:11.00
##  Median :1.000             Median :14.00
##  Mean   :1.224             Mean   :13.28
##  3rd Qu.:1.000             3rd Qu.:16.00
##  Max.   :6.000             Max.   :20.00
##

## [1] 0

## [1] 0

##  Named int(0)
##  - attr(*, "names")= chr(0)

## numeric(0)

## named integer(0)
```

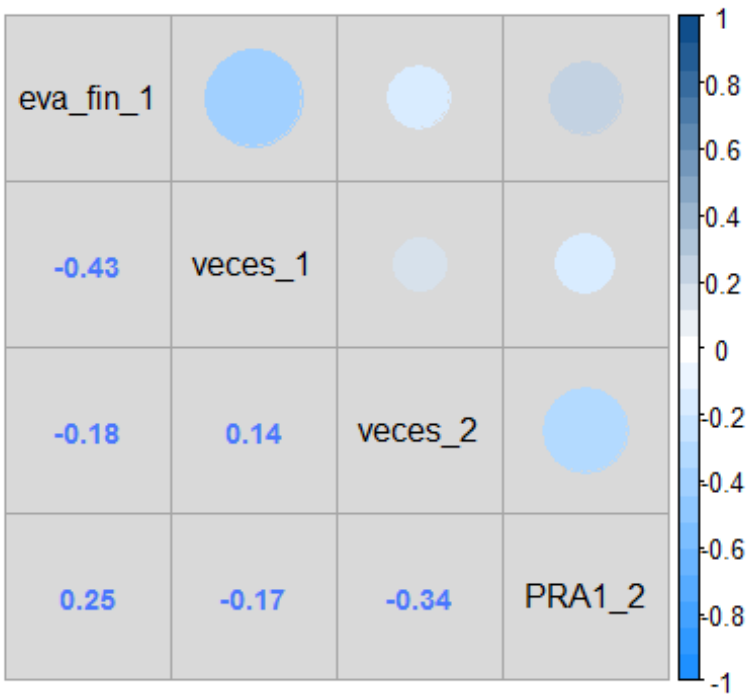



```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##     veces_2     0
##      aprobo     0
##      PRA1_2     0
##
##  Missings in combinations of variables:
##               Combinations Count Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0:0:0  5811     100
```

```
## [1] 86.90415
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

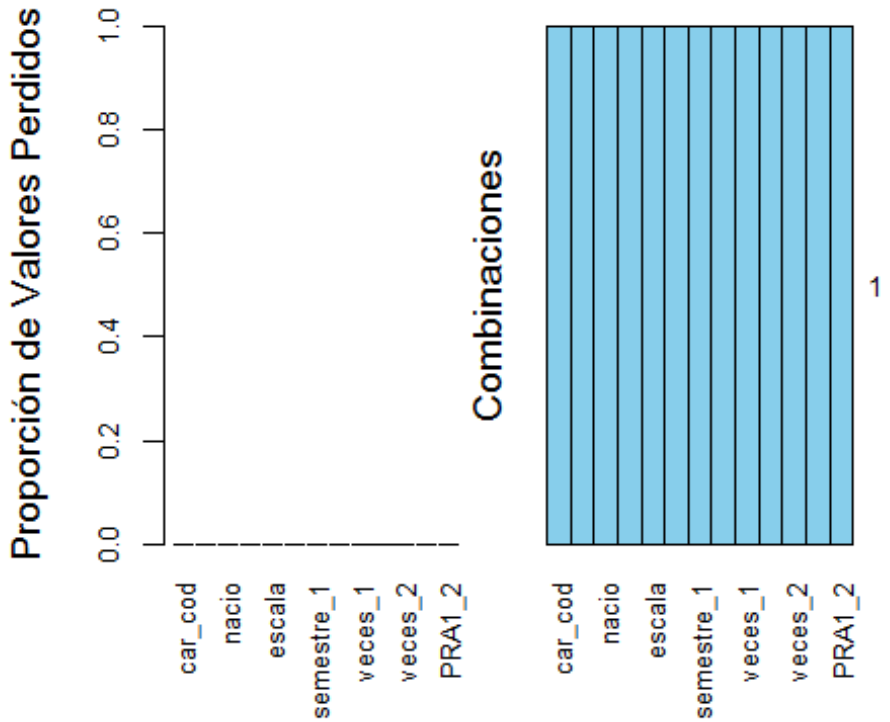
```
##            eva_fin_1    veces_1    veces_2     PRA1_2
## eva_fin_1  1.0000000 -0.4348872 -0.1839991  0.2501770
## veces_1   -0.4348872  1.0000000  0.1383693 -0.1689207
## veces_2   -0.1839991  0.1383693  1.0000000 -0.3421165
## PRA1_2     0.2501770 -0.1689207 -0.3421165  1.0000000
```




```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## -0.43489 -0.30259 -0.17646 -0.12356  0.06155  0.25018 

## [1] 0

## character(0)

## integer(0)

## character(0)
```

```
## 'data.frame':    4337 obs. of  13 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 1 1 1 1 1 1 1 11 1
4 ...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 2 1 1 2 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1997-12-16" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 3 3 3 3 3 3 3 5
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 1 2 2 2 2 2 ...
##  $ semestre_1: Factor w/ 9 levels "20161","20162",..: 4 5 5 5 5 4 5 5 4 7
...
##  $ eva_fin_1 : int  15 15 15 16 17 15 19 12 16 15 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 8 levels "20162","20170",..: 4 6 6 6 6 4 6 5 4 7
...
##  $ veces_2   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 1 ...
##  $ PRA1_2    : int  15 16 13 14 14 11 20 8 11 8 ...

##     car_cod     sexo         nacio               ing_cod     escala    
##  11     : 802   F:2083   Min.   :1959-08-06   13     :1636   A13:   1  
##  61     : 770   M:2254   1st Qu.:1997-08-14   15     :1234   A18:   3  
##  63     : 515            Median :1998-12-16   17     : 750   A23:3023  
```

```
##  32     : 383           Mean   :1998-06-16   04     : 166   A28:  12
##  41     : 378           3rd Qu.:1999-11-10   05     : 147   A33:1257
##  34     : 304           Max.   :2004-05-03   08     :  80   A38:  41
##  (Other):1185                                (Other): 324
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E: 935   20171  :1095   Min.   : 1.00   Min.   :1.000   20182  :1131
##  P:3402   20161  : 869   1st Qu.:12.00   1st Qu.:1.000   20172  :1047
##           20181  : 806   Median :13.00   Median :1.000   20181  : 836
##           20172  : 737   Mean   :13.55   Mean   :1.223   20171  : 629
##           20162  : 727   3rd Qu.:16.00   3rd Qu.:1.000   20162  : 530
##           20180  :  51   Max.   :20.00   Max.   :4.000   20180  :  59
##           (Other):  52                                   (Other): 105
##     veces_2      aprobo        PRA1_2
##  Min.   :1.000   No: 777   Min.   : 0.00
##  1st Qu.:1.000   Si:3560   1st Qu.: 9.00
##  Median :1.000             Median :12.00
##  Mean   :1.295             Mean   :11.42
##  3rd Qu.:1.000             3rd Qu.:14.00
##  Max.   :5.000             Max.   :20.00
##

## [1] 0

## [1] 0

##  Named int(0)
##  - attr(*, "names")= chr(0)

## numeric(0)

## named integer(0)
```



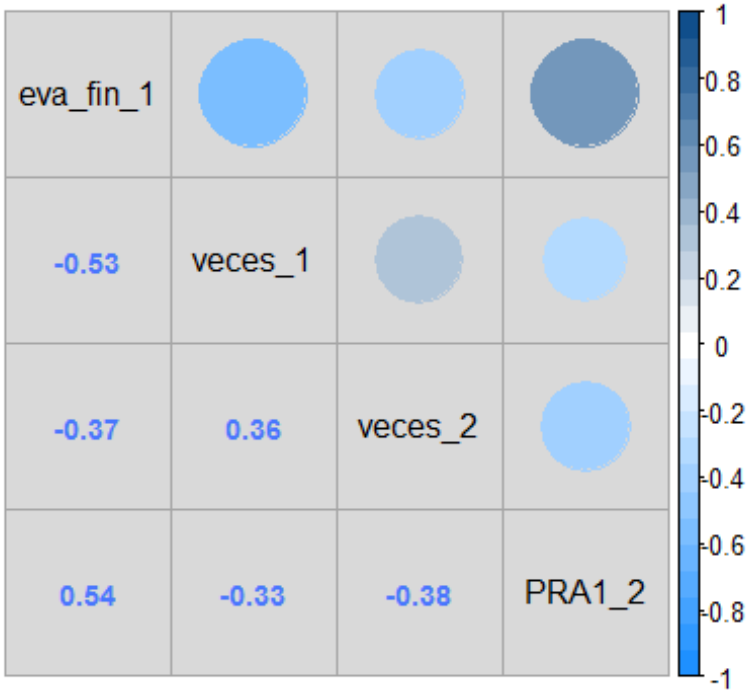

```
##
##  Missings per variable:
##    Variable Count
```

```
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##     veces_2     0
##      aprobo     0
##      PRA1_2     0
##
##  Missings in combinations of variables:
##               Combinations Count Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0  4337     100
```

```
## [1] 82.08439
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##            eva_fin_1    veces_1    veces_2     PRA1_2
## eva_fin_1  1.0000000 -0.5348693 -0.3688980  0.5417919
## veces_1   -0.5348693  1.0000000  0.3565851 -0.3298727
## veces_2   -0.3688980  0.3565851  1.0000000 -0.3781998
## PRA1_2     0.5417919 -0.3298727 -0.3781998  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.5349 -0.3759 -0.3494 -0.1189  0.1850  0.5418
```

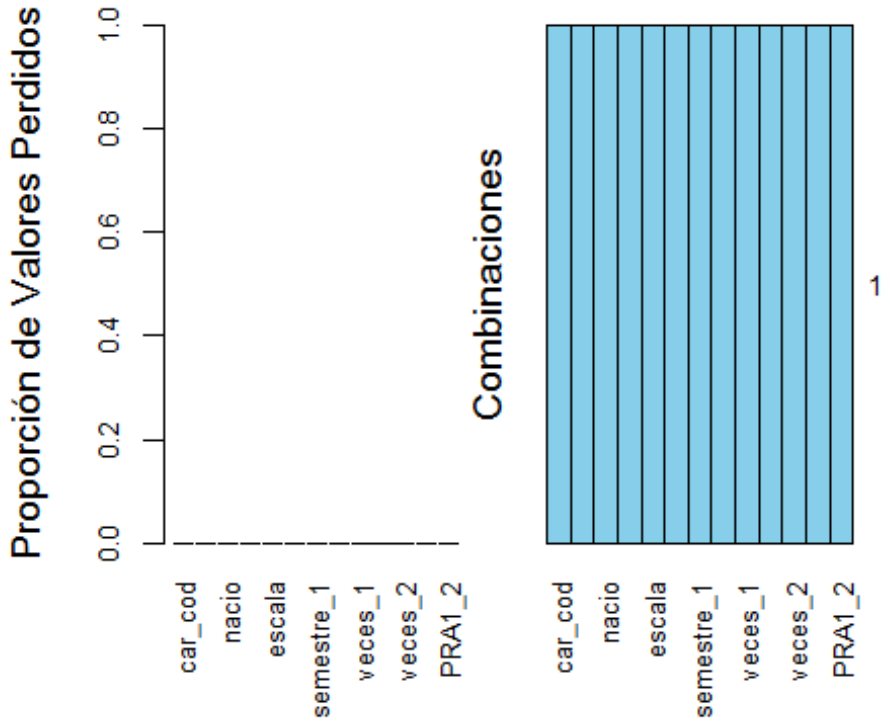
```
## [1] 0
```

```
## character(0)
```

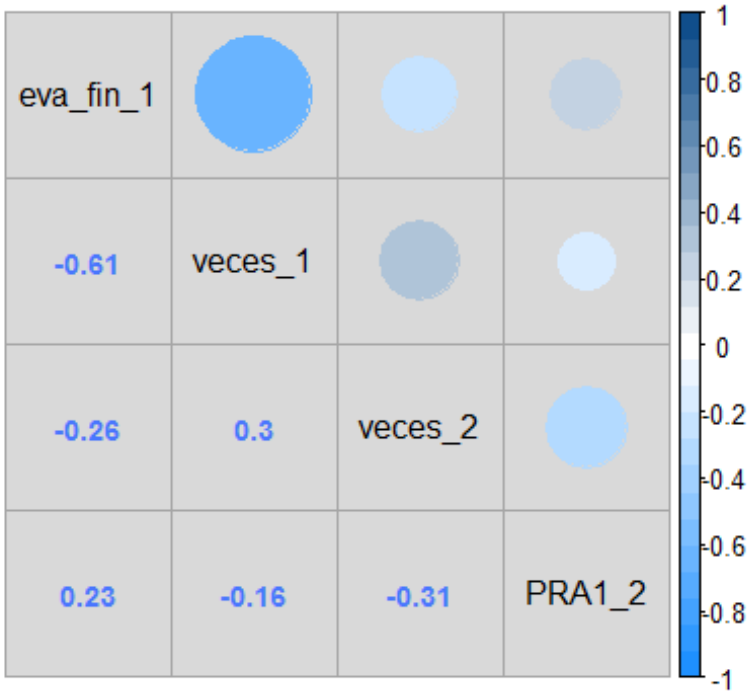
```
## integer(0)
```

```
## character(0)
```

```
## 'data.frame':    8264 obs. of  13 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ eva_fin_1 : int  15 15 15 17 18 15 14 15 13 15 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 12 levels "20151","20152",..: 2 2 2 2 2 2 2 2 2 2
...
##  $ veces_2   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2    : int  15 16 14 17 20 8 9 14 16 20 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  11     :1254   F:3991   Min.   :1957-03-10   13     :3055   A13: 860
##  61     :1153   M:4273   1st Qu.:1997-04-26   15     :2583   A18:   8
##  25     : 868            Median :1998-07-15   17     :1263   A23:5248
##  63     : 781            Mean   :1998-03-19   04     : 336   A28:  23
##  41     : 557            3rd Qu.:1999-09-10   05     : 290   A33:2034
##  51     : 546            Max.   :2004-05-03   08     : 198   A38:  91
##  (Other):3105                                 (Other): 539
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E:1704   20161  :1564   Min.   : 0.00   Min.   :1.000   20182  :1544
##  P:6560   20171  :1363   1st Qu.:12.00   1st Qu.:1.000   20162  :1399
##           20151  :1269   Median :14.00   Median :1.000   20172  :1335
##           20181  :1163   Mean   :13.15   Mean   :1.317   20152  :1028
##           20152  : 919   3rd Qu.:15.00   3rd Qu.:1.000   20181  : 941
##           20172  : 863   Max.   :20.00   Max.   :8.000   20161  : 852
##           (Other):1123                                   (Other):1165
##     veces_2      aprobo        PRA1_2
##  Min.   :1.000   No:1567   Min.   : 0.00
##  1st Qu.:1.000   Si:6697   1st Qu.: 9.00
##  Median :1.000             Median :13.00
##  Mean   :1.341             Mean   :12.04
##  3rd Qu.:1.000             3rd Qu.:16.00
##  Max.   :6.000             Max.   :20.00
##

## [1] 0

## [1] 0

##  Named int(0)
##  - attr(*, "names")= chr(0)

## numeric(0)

## named integer(0)
```



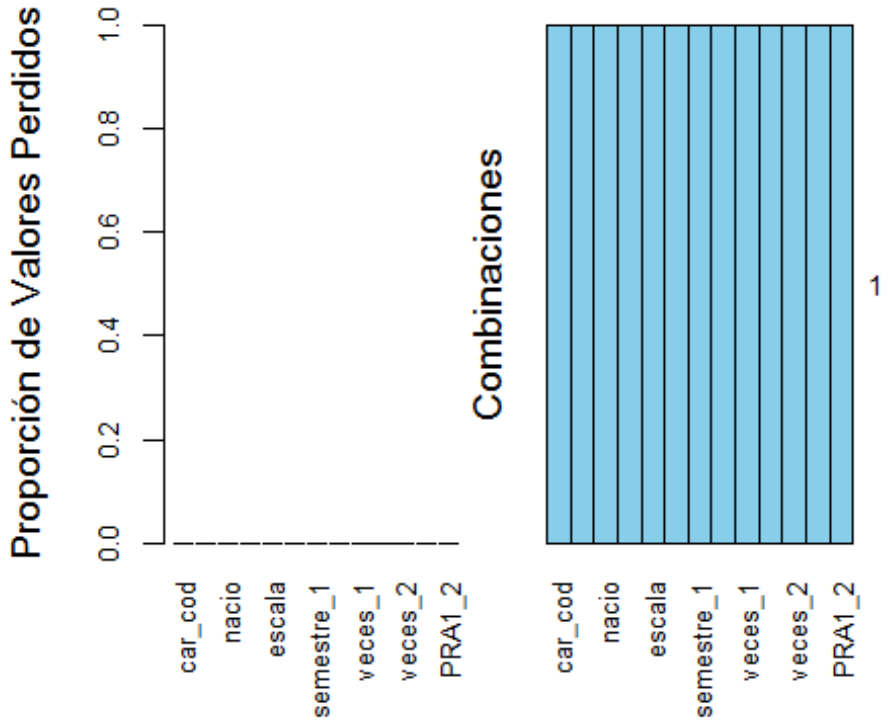

```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##     veces_2     0
##      aprobo     0
##      PRA1_2     0
##
##  Missings in combinations of variables:
##               Combinations Count Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0:0  8264     100

## [1] 81.03824

##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

##            eva_fin_1    veces_1    veces_2     PRA1_2
## eva_fin_1  1.0000000 -0.6081806 -0.2623579  0.2315146
## veces_1   -0.6081806  1.0000000  0.2979029 -0.1595277
## veces_2   -0.2623579  0.2979029  1.0000000 -0.3109697
## PRA1_2     0.2315146 -0.1595277 -0.3109697  1.0000000
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.6082 -0.2988 -0.2109 -0.1353  0.1338  0.2979

## [1] 0

## character(0)

## integer(0)

## character(0)
```

```
## 'data.frame':    6206 obs. of  13 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 11 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ eva_fin_1 : int  14 16 18 16 14 17 14 15 14 16 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 12 levels "20151","20152",..: 4 4 4 4 4 4 2 2 2 3
...
##  $ veces_2   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2    : int  15 15 6 15 14 10 9 8 15 9 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  11     :1022   F:3049   Min.   :1957-04-08   13     :2296   A13: 499
##  61     : 964   M:3157   1st Qu.:1997-02-12   15     :1871   A18:   3
##  63     : 599            Median :1998-05-09   17     : 936   A23:4100
```

```
##  25     : 502           Mean   :1997-12-09   04     : 258   A28:  23
##  41     : 417           3rd Qu.:1999-06-14   05     : 240   A33:1535
##  51     : 403           Max.   :2004-05-03   19     : 137   A38:  46
##  (Other):2299                                (Other): 468
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E:1292   20171  :1352   Min.   : 1.00   Min.   :1.000   20181  :1296
##  P:4914   20161  :1266   1st Qu.:13.00   1st Qu.:1.000   20171  :1243
##           20162  :1002   Median :15.00   Median :1.000   20182  : 969
##           20151  : 771   Mean   :14.74   Mean   :1.182   20172  : 940
##           20172  : 721   3rd Qu.:17.00   3rd Qu.:1.000   20161  : 720
##           20152  : 505   Max.   :20.00   Max.   :6.000   20162  : 566
##           (Other): 589                                   (Other): 472
##     veces_2      aprobo        PRA1_2
##  Min.   :1.000   No:1325   Min.   : 0.00
##  1st Qu.:1.000   Si:4881   1st Qu.: 9.00
##  Median :1.000             Median :12.00
##  Mean   :1.395             Mean   :11.82
##  3rd Qu.:2.000             3rd Qu.:15.00
##  Max.   :6.000             Max.   :20.00
##

## [1] 0

## [1] 0

##  Named int(0)
##  - attr(*, "names")= chr(0)

## numeric(0)

## named integer(0)
```



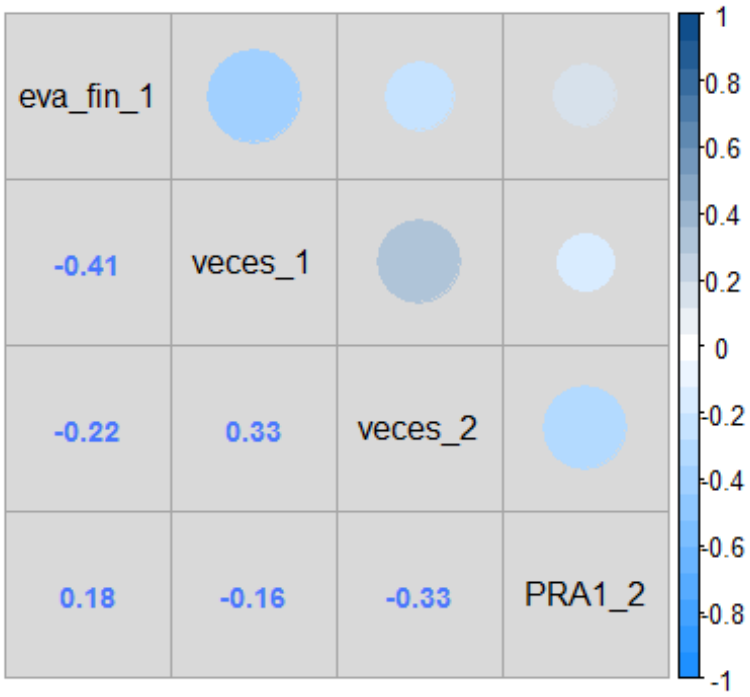

```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
```

```
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##     veces_2     0
##      aprobo     0
##      PRA1_2     0
##
##  Missings in combinations of variables:
##               Combinations Count Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0:0  6206     100
```

```
## [1] 78.64969
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##            eva_fin_1    veces_1    veces_2     PRA1_2
## eva_fin_1  1.0000000 -0.4116661 -0.2238531  0.1838307
## veces_1   -0.4116661  1.0000000  0.3310232 -0.1570386
## veces_2   -0.2238531  0.3310232  1.0000000 -0.3296078
## PRA1_2     0.1838307 -0.1570386 -0.3296078  1.0000000
```




```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.41167 -0.30317 -0.19045 -0.10122  0.09861  0.33102
```

```
## [1] 0
```

```
## character(0)
```

```
## integer(0)
```

```
## character(0)
```

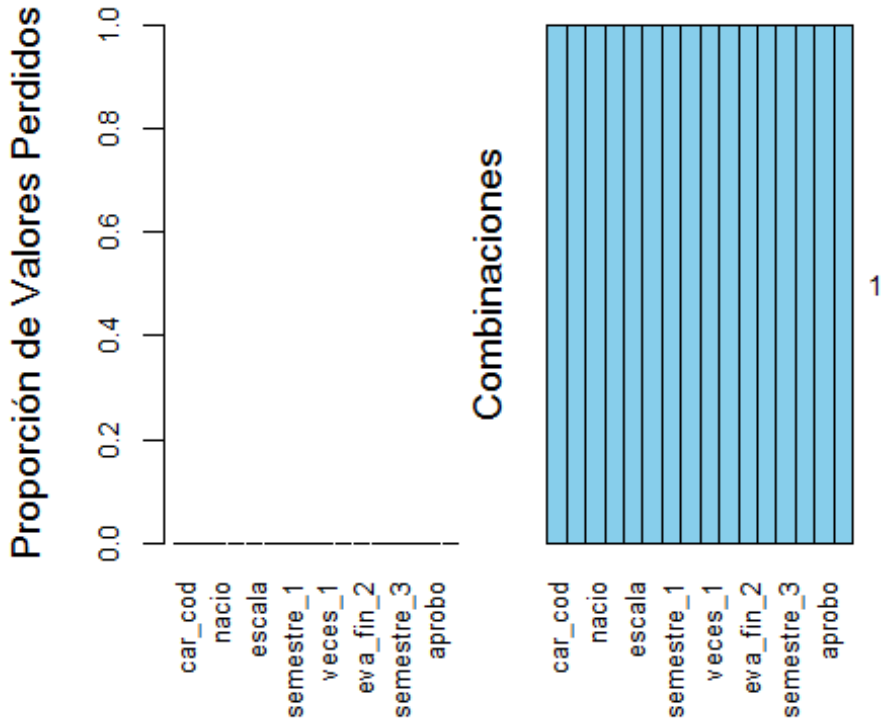
3. Creación de los datasets de tres cursos.

Generación de la estructura interna del décimo segundo dataset.
Formado por:
1. *"Taller de Método de Estudio Universitario", con código: "0002"*
2. *"Formación Histórica del Perú", con código: "0010"*
3. *"Realidad Nacional", con código: "0012"*

Generación de la estructura interna del décimo tercer dataset.
Formado por:
1. *"Taller de Método de Estudio Universitario", con código: "0002"*
2. *"Formación Histórica del Perú", con código: "0010"*
3. *"Historia de la Civilización", con código: "0013"*

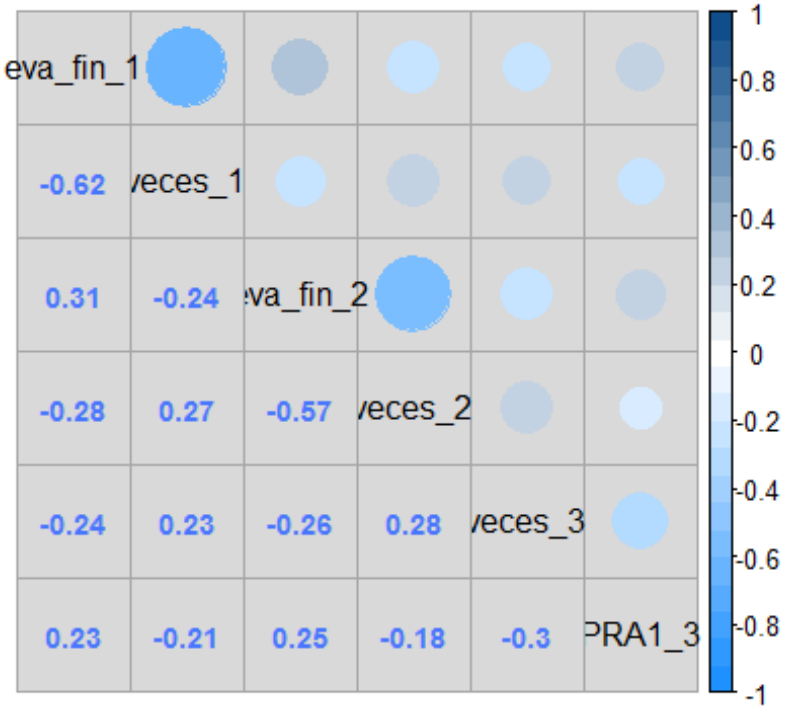
```
## 'data.frame':    6527 obs. of  16 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ eva_fin_1 : int  15 15 15 17 18 15 14 15 13 15 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 12 levels "20151","20152",..: 2 2 2 2 2 2 2 2 2 2
...
##  $ eva_fin_2 : int  14 14 13 16 12 12 13 14 17 16 ...
##  $ veces_2   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_3: Factor w/ 12 levels "20151","20152",..: 4 4 3 4 4 4 4 4 3 4
...
##  $ veces_3   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_3    : int  14 9 14 15 13 14 14 15 15 13 ...

##     car_cod     sexo         nacio               ing_cod     escala
##  11     :1061   F:3086   Min.   :1957-03-10   13     :2398   A13: 503
##  61     :1029   M:3441   1st Qu.:1997-03-07   15     :2099   A18:   5
##  63     : 640            Median :1998-05-08   17     : 955   A23:4296
##  25     : 508            Mean   :1997-12-26   04     : 280   A28:  22
##  32     : 455            3rd Qu.:1999-05-23   05     : 239   A33:1634
##  41     : 447            Max.   :2004-05-03   08     : 152   A38:  67
##  (Other):2387                                 (Other): 404
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E:1317   20161  :1372   Min.   : 0.00   Min.   :1.000   20172  :1239
##  P:5210   20171  :1210   1st Qu.:12.00   1st Qu.:1.000   20162  :1199
##           20151  :1141   Median :14.00   Median :1.000   20152  : 912
##           20152  : 815   Mean   :13.09   Mean   :1.356   20181  : 812
##           20162  : 796   3rd Qu.:15.00   3rd Qu.:1.000   20171  : 806
##           20172  : 740   Max.   :19.00   Max.   :8.000   20161  : 758
```

```
##           (Other): 453                                    (Other): 801
##    eva_fin_2        veces_2       semestre_3      veces_3      aprobo
##  Min.   : 0.00   Min.   :1.00   20171  :1541   Min.   :1.000   No: 932
##  1st Qu.:11.00   1st Qu.:1.00   20181  :1485   1st Qu.:1.000   Si:5595
##  Median :13.00   Median :1.00   20172  : 999   Median :1.000
##  Mean   :12.37   Mean   :1.39   20182  : 939   Mean   :1.231
##  3rd Qu.:15.00   3rd Qu.:2.00   20161  : 703   3rd Qu.:1.000
##  Max.   :20.00   Max.   :6.00   20162  : 584   Max.   :5.000
##                                 (Other): 276
##      PRA1_3
##  Min.   : 0.00
##  1st Qu.:11.00
##  Median :13.00
##  Mean   :12.48
##  3rd Qu.:15.00
##  Max.   :20.00
##

## [1] 0

## [1] 0

##  Named int(0)
##  - attr(*, "names")= chr(0)

## numeric(0)

## named integer(0)
```




```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
```

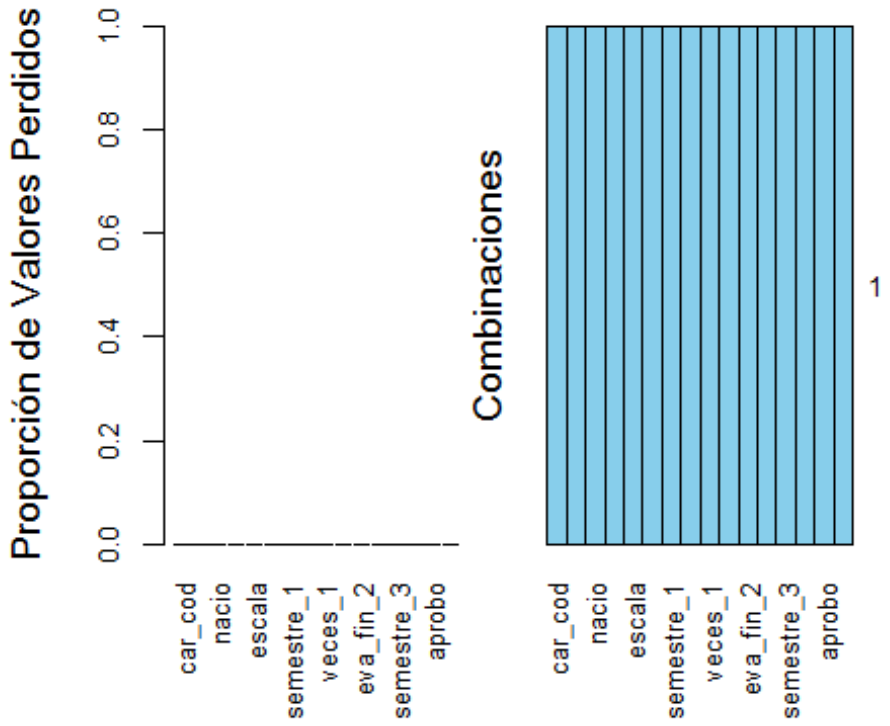
```
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##   eva_fin_2     0
##     veces_2     0
##  semestre_3     0
##     veces_3     0
##      aprobo     0
##      PRA1_3     0
##
##  Missings in combinations of variables:
##                     Combinations Count Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0  6527     100

## [1] 85.72085

##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE

##            eva_fin_1    veces_1  eva_fin_2    veces_2    veces_3
## eva_fin_1  1.0000000 -0.6163399  0.3077077 -0.2760591 -0.2359518
## veces_1   -0.6163399  1.0000000 -0.2370912  0.2663245  0.2257597
## eva_fin_2  0.3077077 -0.2370912  1.0000000 -0.5720538 -0.2579409
## veces_2   -0.2760591  0.2663245 -0.5720538  1.0000000  0.2757040
## veces_3   -0.2359518  0.2257597 -0.2579409  0.2757040  1.0000000
## PRA1_3     0.2274469 -0.2089552  0.2465748 -0.1783070 -0.3021765
##               PRA1_3
## eva_fin_1  0.2274469
## veces_1   -0.2089552
## eva_fin_2  0.2465748
## veces_2   -0.1783070
## veces_3   -0.3021765
## PRA1_3     1.0000000
```



```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.61634 -0.26700 -0.20896 -0.08902  0.23701  0.30771

## [1] 0

## character(0)

## integer(0)

## character(0)
```

```
## 'data.frame':    6968 obs. of  16 variables:
##  $ car_cod   : Factor w/ 18 levels "11","21","25",..: 1 7 1 1 1 1 1 1 1 1
...
##  $ sexo      : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 2 ...
##  $ nacio     : Date, format: "1998-07-21" "1998-06-27" ...
##  $ ing_cod   : Factor w/ 20 levels "01","02","03",..: 6 6 6 6 6 6 6 6 6 6
...
##  $ escala    : Factor w/ 6 levels "A13","A18","A23",..: 3 5 3 3 3 3 3 3 3
3 ...
##  $ col_tipo  : Factor w/ 2 levels "E","P": 2 2 2 2 2 1 2 2 2 2 ...
##  $ semestre_1: Factor w/ 12 levels "20151","20152",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ eva_fin_1 : int  15 15 15 17 18 15 14 15 13 15 ...
##  $ veces_1   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_2: Factor w/ 12 levels "20151","20152",..: 2 2 2 2 2 2 2 2 2 2
...
##  $ eva_fin_2 : int  14 14 13 16 12 12 13 14 17 16 ...
##  $ veces_2   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ semestre_3: Factor w/ 12 levels "20151","20152",..: 4 4 3 4 4 4 4 4 4 4
...
##  $ veces_3   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo    : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_3    : int  15 13 11 17 20 14 16 16 14 12 ...

##     car_cod     sexo         nacio                ing_cod     escala
##  11     :1147   F:3217   Min.   :1973-06-16   13     :2534   A13: 471
##  61     :1092   M:3751   1st Qu.:1997-01-07   15     :2247   A18:   7
##  63     : 677            Median :1998-04-22   17     :1008   A23:4602
##  32     : 495            Mean   :1997-12-02   04     : 291   A28:  23
##  25     : 478            3rd Qu.:1999-05-11   05     : 250   A33:1767
##  41     : 462            Max.   :2004-05-03   08     : 196   A38:  98
##  (Other):2617                                 (Other): 442
##  col_tipo   semestre_1     eva_fin_1        veces_1        semestre_2
##  E:1427   20161  :1526   Min.   : 0.00   Min.   :1.000   20162  :1335
##  P:5541   20171  :1313   1st Qu.:12.00   1st Qu.:1.000   20172  :1330
##           20151  :1224   Median :13.00   Median :1.000   20152  : 953
##           20152  : 912   Mean   :13.01   Mean   :1.365   20181  : 860
##           20162  : 772   3rd Qu.:15.00   3rd Qu.:1.000   20161  : 825
##           20172  : 742   Max.   :19.00   Max.   :8.000   20171  : 807
##           (Other): 479                                   (Other): 858
##    eva_fin_2        veces_2        semestre_3      veces_3      aprobo
##  Min.   : 0.00   Min.   :1.000   20181  :1593   Min.   :1.000   No:1545
##  1st Qu.:11.00   1st Qu.:1.000   20171  :1536   1st Qu.:1.000   Si:5423
##  Median :13.00   Median :1.000   20172  :1221   Median :1.000
##  Mean   :12.28   Mean   :1.416   20182  :1019   Mean   :1.431
##  3rd Qu.:15.00   3rd Qu.:2.000   20161  : 704   3rd Qu.:2.000
##  Max.   :20.00   Max.   :6.000   20162  : 652   Max.   :6.000
##                                  (Other): 243
```

```
##      PRA1_3
##  Min.   : 0.00
##  1st Qu.: 7.00
##  Median :12.00
##  Mean   :11.15
##  3rd Qu.:16.00
##  Max.   :20.00
##

## [1] 0

## [1] 0

##  Named int(0)
##  - attr(*, "names")= chr(0)

## numeric(0)

## named integer(0)
```



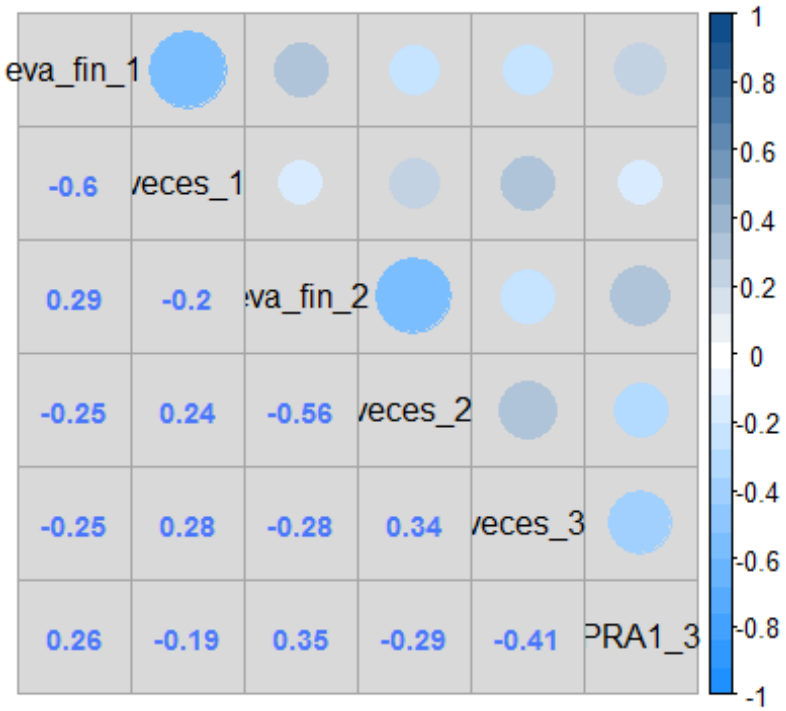

```
##
##  Missings per variable:
##    Variable Count
##     car_cod     0
##        sexo     0
##       nacio     0
##     ing_cod     0
##      escala     0
##    col_tipo     0
##  semestre_1     0
##   eva_fin_1     0
##     veces_1     0
##  semestre_2     0
##   eva_fin_2     0
##     veces_2     0
```

```
##  semestre_3     0
##     veces_3     0
##      aprobo     0
##      PRA1_3     0
##
##  Missings in combinations of variables:
##                     Combinations Count Percent
##  0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0  6968     100
```

```
## [1] 77.82721
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE
```

```
##            eva_fin_1    veces_1  eva_fin_2    veces_2    veces_3
## eva_fin_1  1.0000000 -0.5982604  0.2854545 -0.2474717 -0.2541630
## veces_1   -0.5982604  1.0000000 -0.1958588  0.2436157  0.2813069
## eva_fin_2  0.2854545 -0.1958588  1.0000000 -0.5615859 -0.2789120
## veces_2   -0.2474717  0.2436157 -0.5615859  1.0000000  0.3384954
## veces_3   -0.2541630  0.2813069 -0.2789120  0.3384954  1.0000000
## PRA1_3     0.2600241 -0.1862374  0.3481379 -0.2917783 -0.4084930
##               PRA1_3
## eva_fin_1  0.2600241
## veces_1   -0.1862374
## eva_fin_2  0.3481379
## veces_2   -0.2917783
## veces_3   -0.4084930
## PRA1_3     1.0000000
```




```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.59826 -0.28535 -0.19586 -0.08438  0.27067  0.34814
```

```
## [1] 0
```

```
## character(0)
```

```
## integer(0)
## character(0)
```

Anexo 14: Resultados del tuning de los modelos de prueba de cada técnica de modelado para cada curso del Programa de Estudios Básicos (PEB).

Generacion del modelo para cada uno de los Archivos de Datos.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")

# Uso de Librerías
library(caret)
library(caretEnsemble)
library(dummies)
library(Boruta) # Selección de variables
library(DMwR) # SMOTE
library(devtools) # Para visualizar la RNA
library(dplyr) # Para resumenes
source_url('https://gist.githubusercontent.com/fawda123/7471137/raw/466c1474d
0a505ff044412703516c34f1a4684a5/nnet_plot_update.r')
```

Para la generacion del modelo para el Curso con código: "0001" sirvase revisar el Anexo 09.

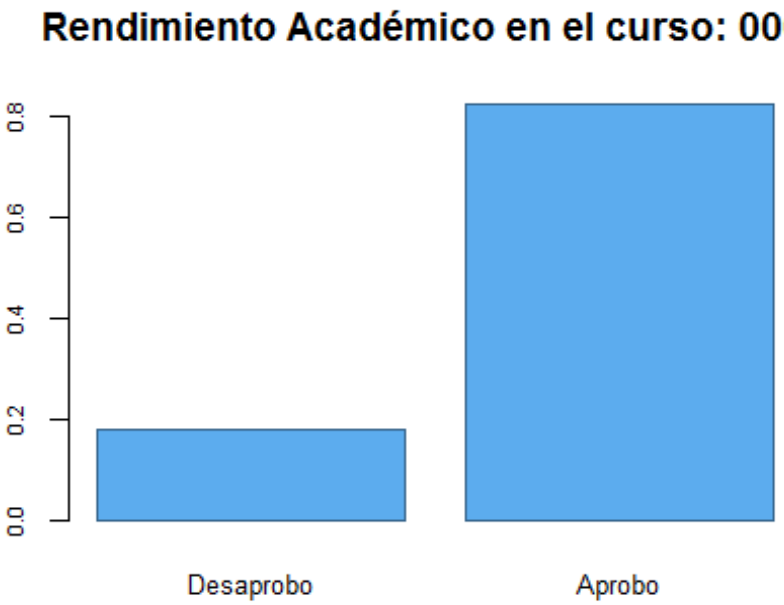
---

1. Generacion del modelo para los Cursos

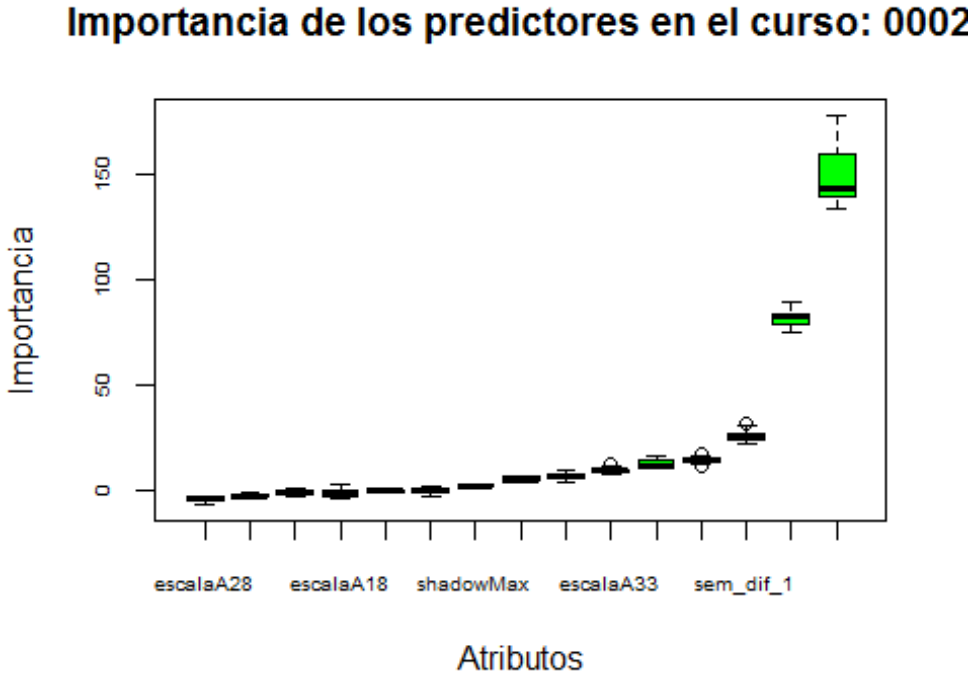
0002: "Taller de Método de Estudio Universitario".
0003: "Taller de Comunicación Oral y Escrita I".
0004: "Matemática".
0005: "Inglés I".
0006: "Psicología General".
0007: "Lógica y Filosofía".
0008: "Taller de Comunicación Oral y Escrita II".
0009: "Inglés II".
0010: "Formación Histórica del Perú".
0011: "Recursos Naturales y Medio Ambiente".
0012: "Realidad Nacional".
0013: "Historia de la Civilización".

```
## [1] "***** Inicio - Curso: 0002*****"
## 'data.frame':    9710 obs. of  13 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
```

```
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1   : int  13 16 17 17 18 15 15 14 9 11 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...
```

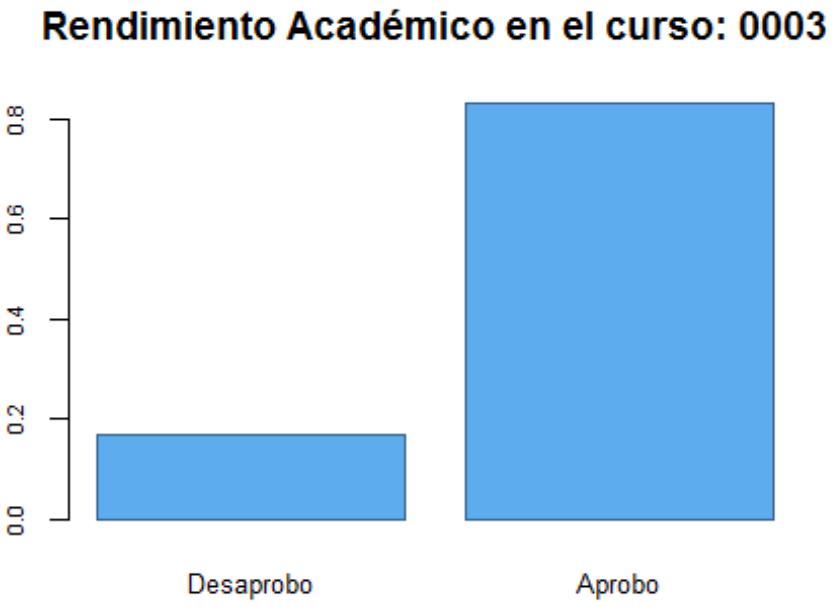



```
##
##       No       Si
## 17.81406 82.18594
```

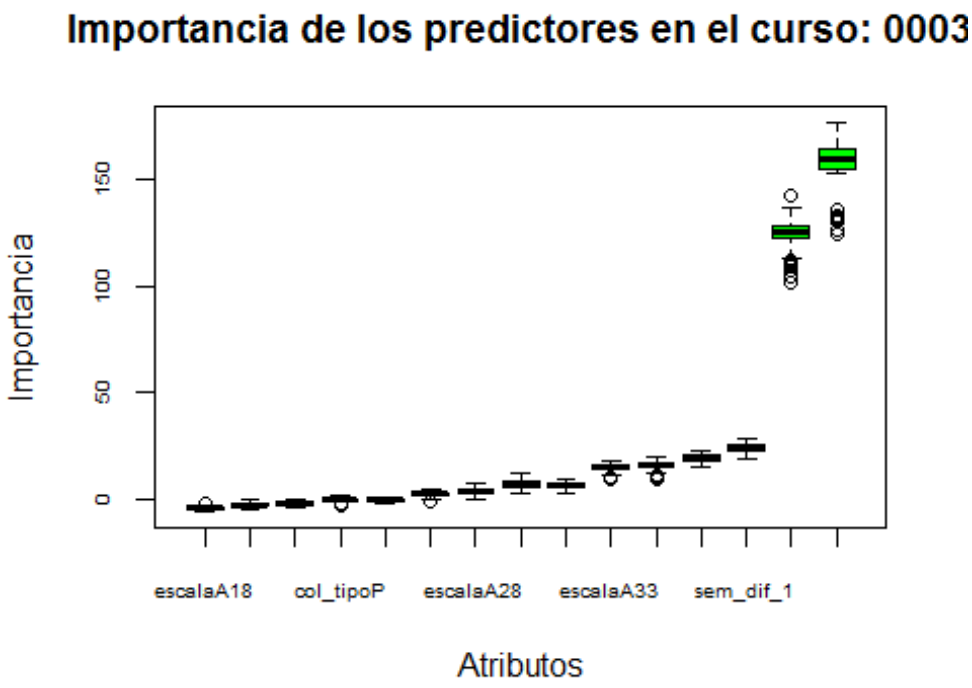



```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0002
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
02"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0002"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0002"
```

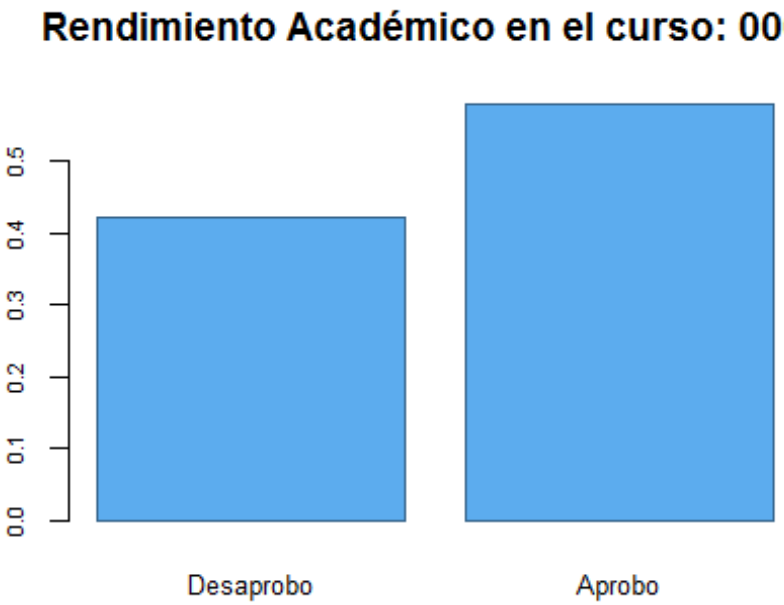
```
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0003*****"
## 'data.frame':    9547 obs. of  13 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1   : int  19 17 11 13 15 13 13 14 12 13 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...
```



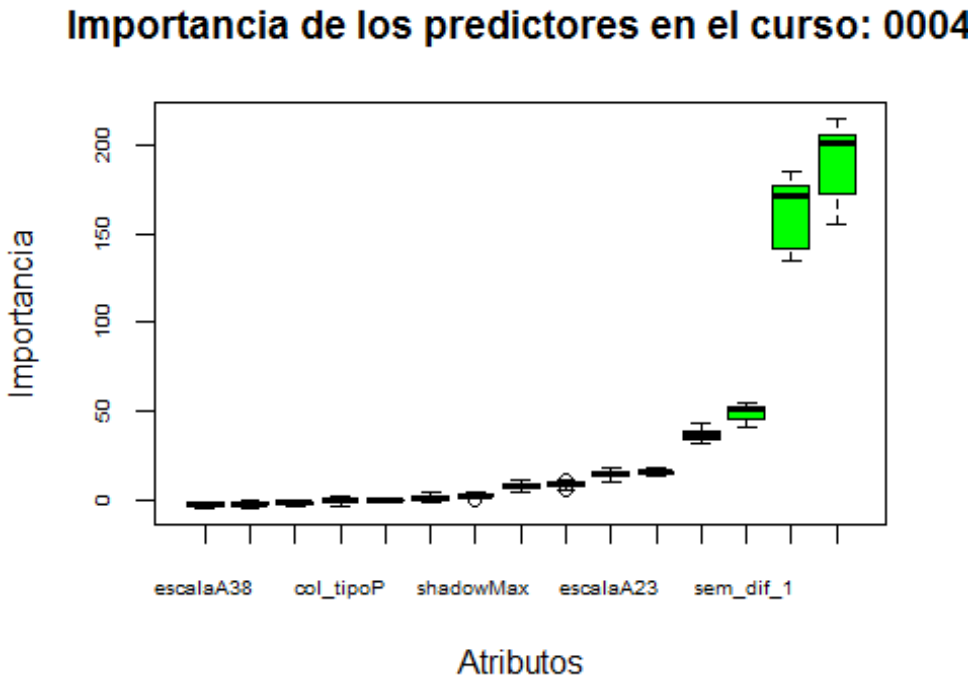

```
##
##       No       Si
## 16.80132 83.19868
```




```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0003
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
03"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0003"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0003"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0004*****"
## 'data.frame':    12102 obs. of  13 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 0 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 2 2 2 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 1 2 1 ...
##  $ PRA1_1   : int  14 17 15 19 6 12 15 12 10 0 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 13 12 ...
```

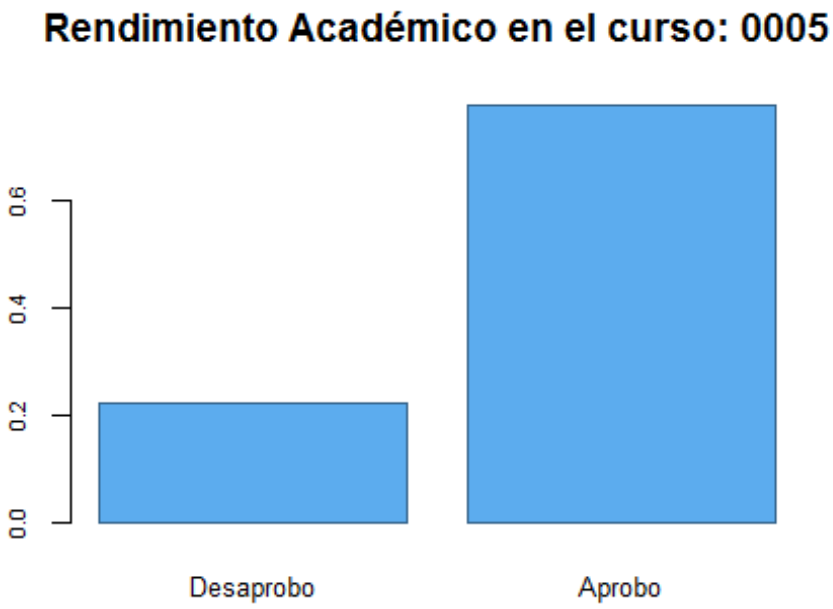



```
##
##       No       Si
## 42.12701 57.87299
```

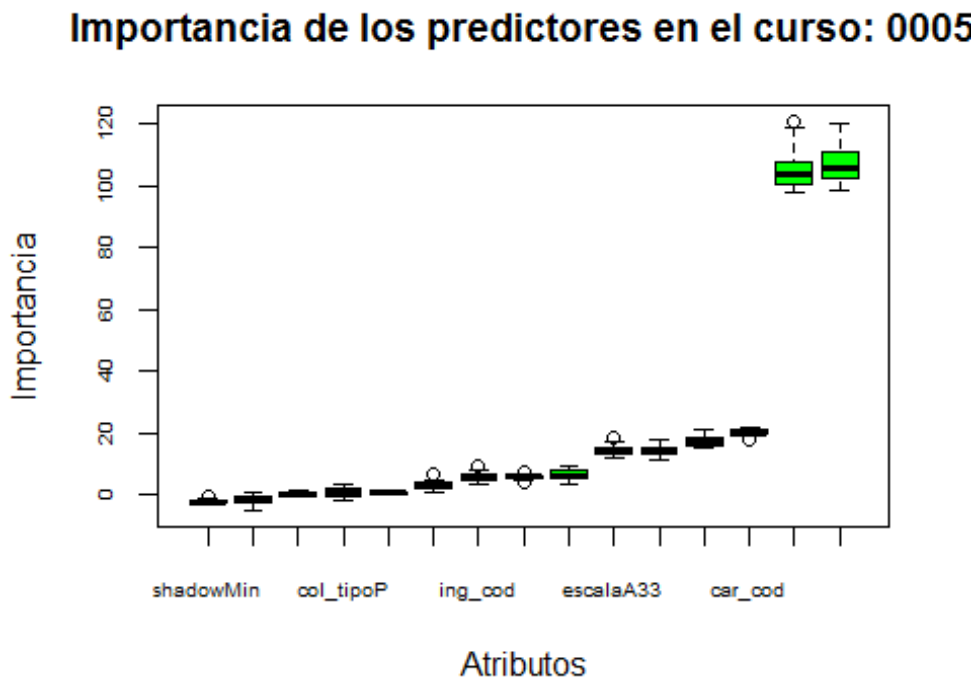



```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0004
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
04"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0004"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0004"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0005*****"
## 'data.frame':    6092 obs. of  13 variables:
##  $ car_cod  : int  11 11 11 11 11 11 11 11 11 41 ...
```

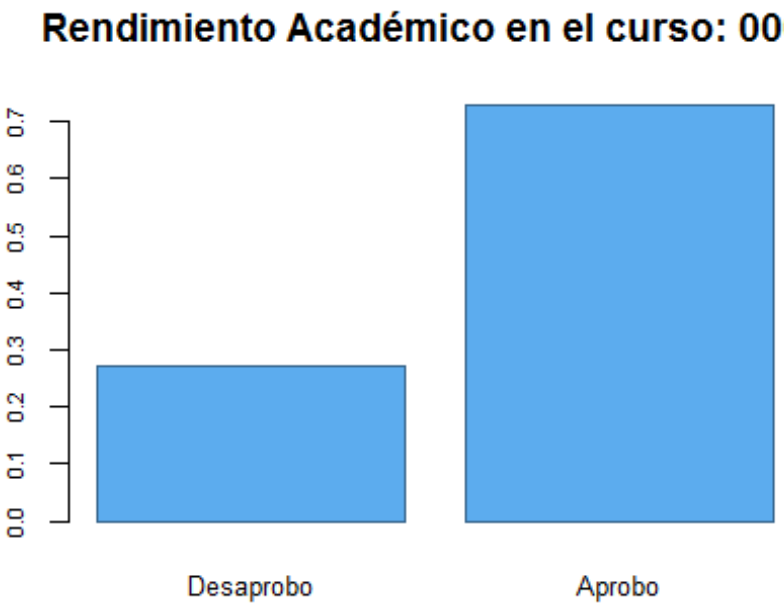
```
##  $ sexoM    : int  1 0 0 0 1 0 0 0 1 0 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 1 1 1 1 1 1 1 1 0 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 0 0 0 0 0 0 0 0 1 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 0 1 1 1 1 1 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_1   : int  16 15 14 15 20 12 20 9 13 14 ...
##  $ sem_dif_1: int  18 22 19 19 22 18 19 18 19 18 ...
## NULL
```



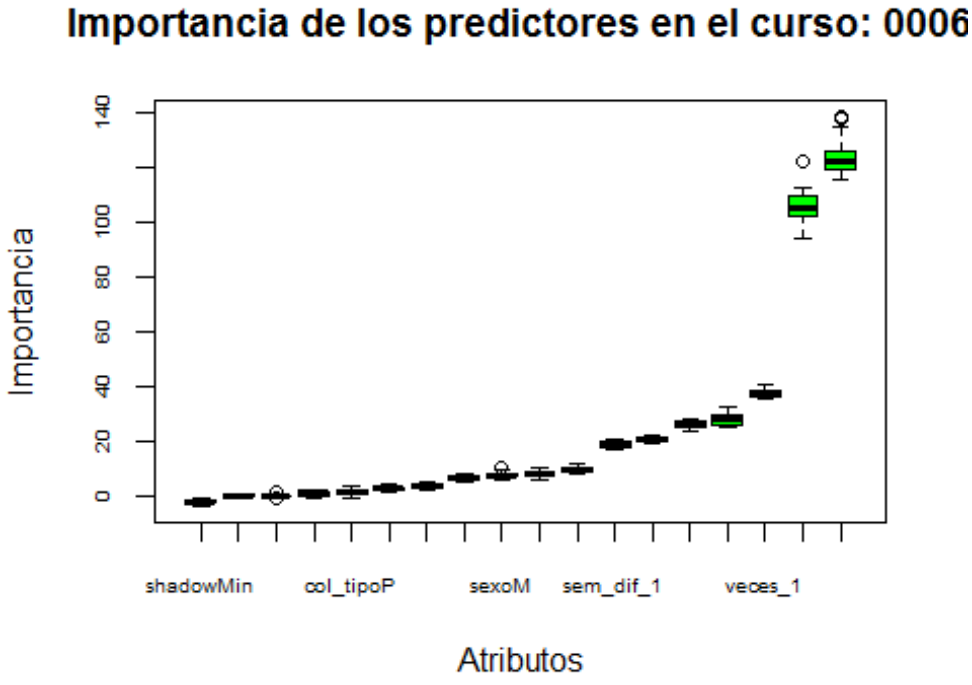

```
##
##       No       Si
## 22.22222 77.77778
```




```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0005
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
05"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0005"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0005"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0006*****"
## 'data.frame':    8748 obs. of  16 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ eva_fin_1: int  14 16 14 12 14 14 13 14 12 14 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_2  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2   : int  17 20 14 15 13 6 5 9 12 11 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...
##  $ sem_dif_2: int  15 11 14 11 11 14 16 11 13 11 ...
## NULL
```

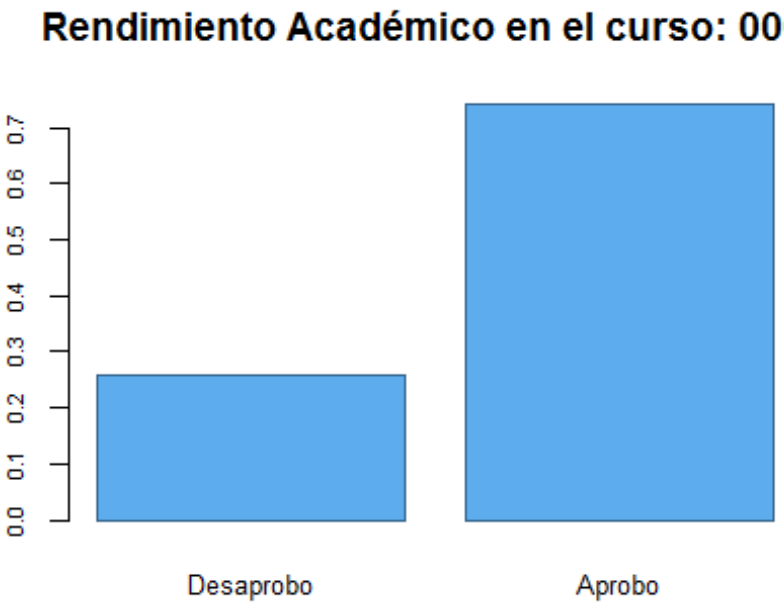



```
##
##       No       Si
## 27.18811 72.81189
```

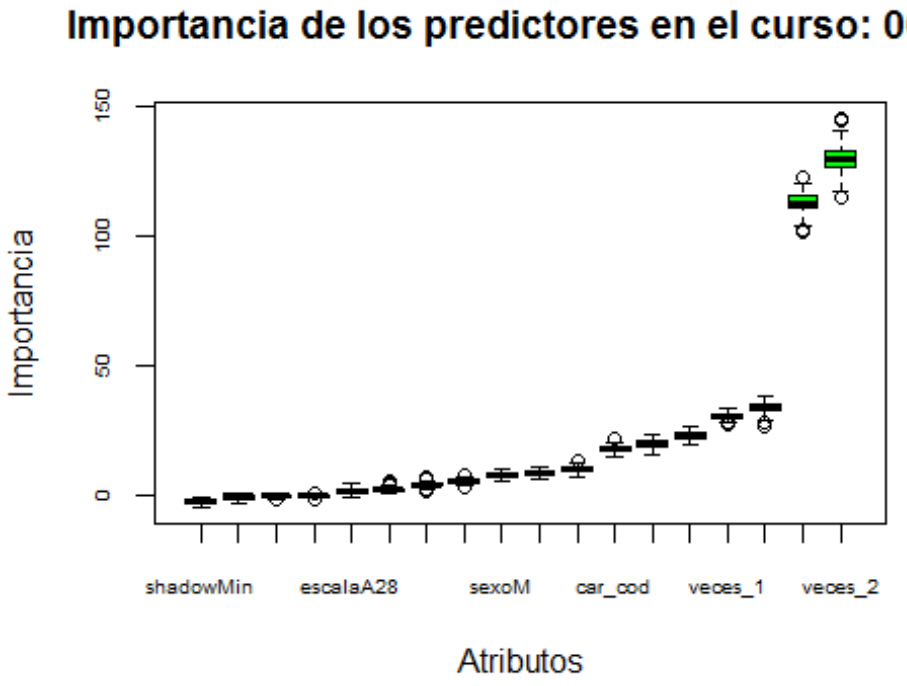



```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0006
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
06"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0006"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0006"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0007*****"
## 'data.frame':    8525 obs. of  16 variables:
##  $ car_cod  : int  11 11 33 11 11 11 11 11 11 11 ...
```

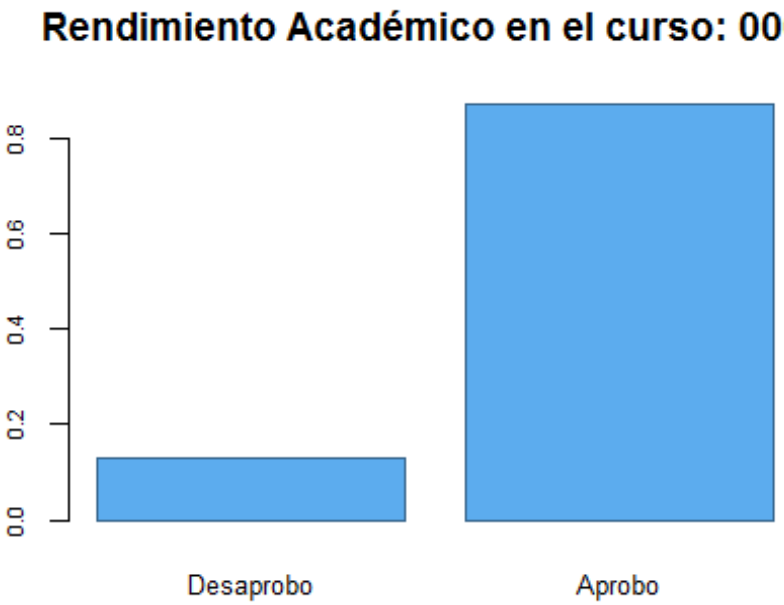
```
##  $ sexoM    : int  1 1 0 0 0 0 1 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 1 0 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 0 1 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 1 0 1 1 1 ...
##  $ eva_fin_1: int  15 15 15 15 17 18 15 14 15 15 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_2  : int  2 2 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 1 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2   : int  7 10 10 11 14 10 15 13 7 7 ...
##  $ sem_dif_1: int  12 12 12 15 12 12 15 12 12 12 ...
##  $ sem_dif_2: int  11 13 11 14 11 11 14 11 11 11 ...
## NULL
```



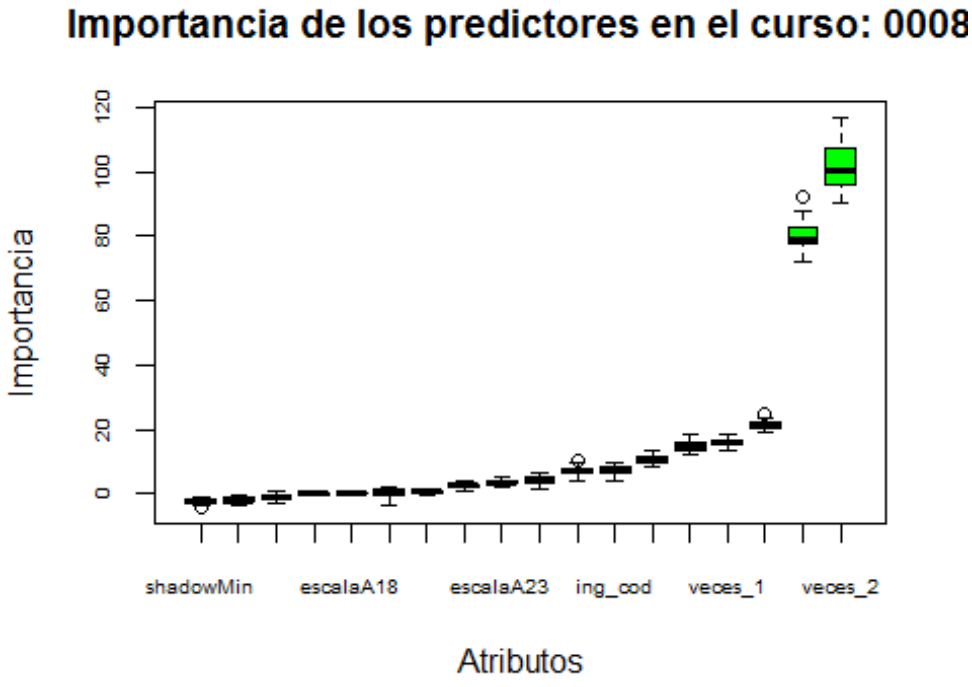

```
##
##       No       Si
## 25.82105 74.17895
```




```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0007
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
07"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0007"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0007"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0008*****"
## 'data.frame':    5811 obs. of  16 variables:
##  $ car_cod  : int  11 11 11 11 11 11 11 11 11 41 ...
##  $ sexoM    : int  1 0 0 0 1 0 0 0 1 0 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 1 1 1 1 1 1 1 1 0 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 0 0 0 0 0 0 0 0 1 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 0 1 1 1 1 1 ...
##  $ eva_fin_1: int  14 14 12 14 14 13 14 12 14 11 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_2  : int  1 1 1 1 1 1 1 1 1 3 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 1 ...
##  $ PRA1_2   : int  9 18 14 11 14 12 11 12 13 10 ...
##  $ sem_dif_1: int  12 15 12 12 15 12 12 12 12 12 ...
##  $ sem_dif_2: int  17 20 17 17 20 16 17 16 17 17 ...
## NULL
```

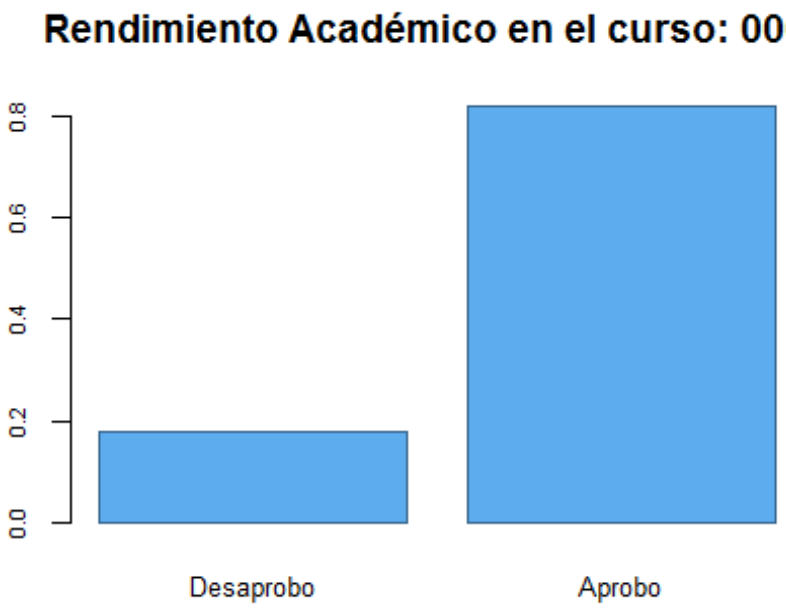



```
##
##       No       Si
## 13.10226 86.89774
```

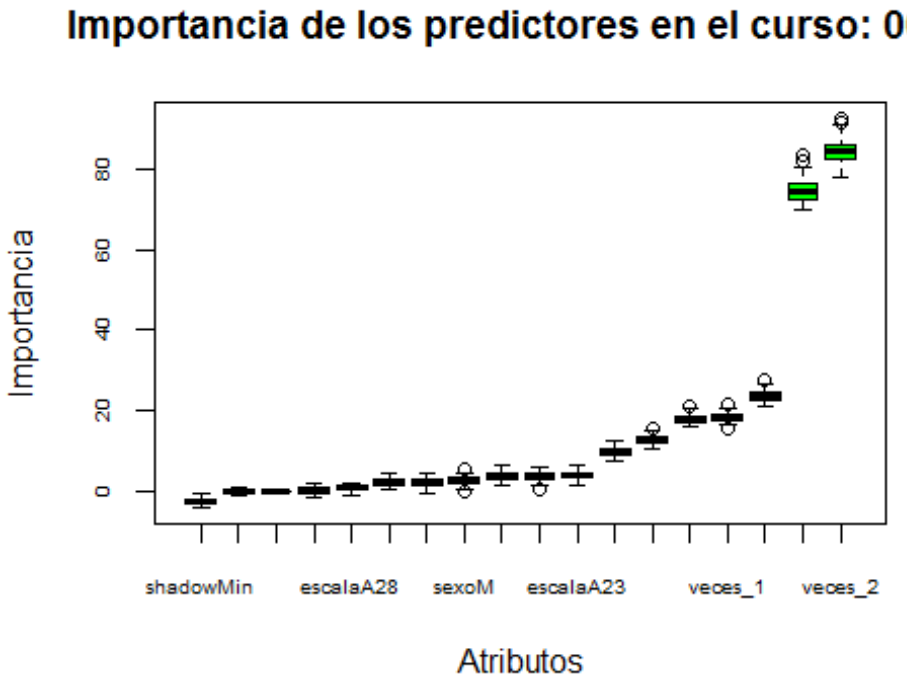



```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0008
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
08"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0008"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0008"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0009*****"
## 'data.frame':    4337 obs. of  16 variables:
##  $ car_cod  : int  11 11 11 11 11 11 11 11 41 61 ...
```

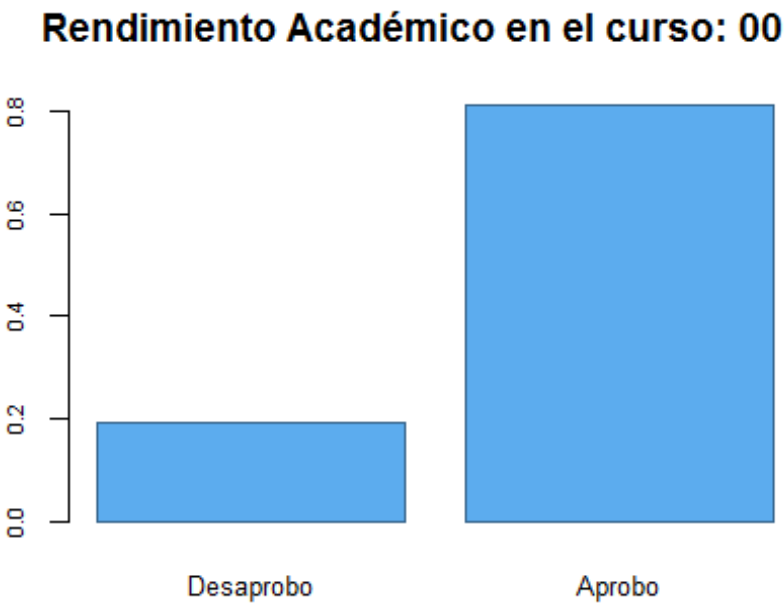
```
##  $ sexoM    : int  1 0 0 0 1 0 0 1 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 1 1 1 1 1 1 0 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 0 0 0 0 0 0 0 1 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 0 1 1 1 1 1 ...
##  $ eva_fin_1: int  15 15 15 16 17 15 19 12 16 15 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_2  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 1 ...
##  $ PRA1_2   : int  15 16 13 14 14 11 20 8 11 8 ...
##  $ sem_dif_1: int  18 22 19 19 22 18 19 19 18 21 ...
##  $ sem_dif_2: int  17 22 19 19 22 17 19 18 17 20 ...
## NULL
```



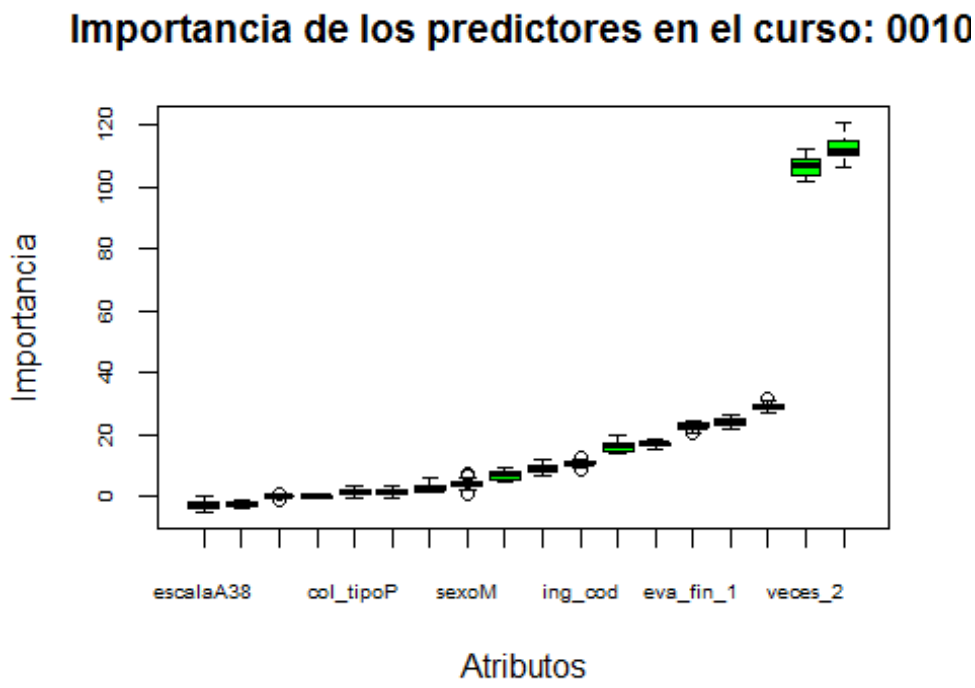

```
##
##       No       Si
## 17.91831 82.08169
```




```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0009
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
09"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0009"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0009"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0010*****"
## 'data.frame':    8264 obs. of  16 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ eva_fin_1: int  15 15 15 17 18 15 14 15 13 15 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_2  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2   : int  15 16 14 17 20 8 9 14 16 20 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...
##  $ sem_dif_2: int  11 11 14 11 11 14 11 11 11 11 ...
## NULL
```

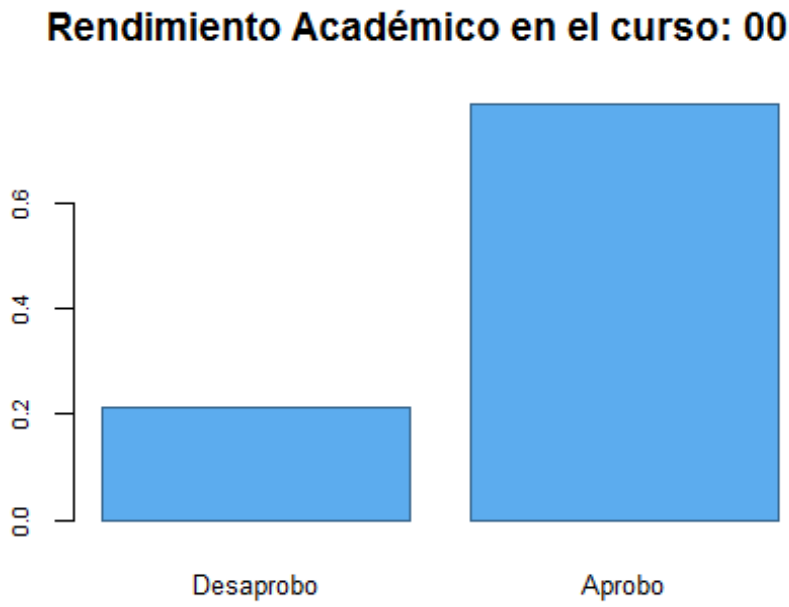



```
##
##       No       Si
## 18.96283 81.03717
```

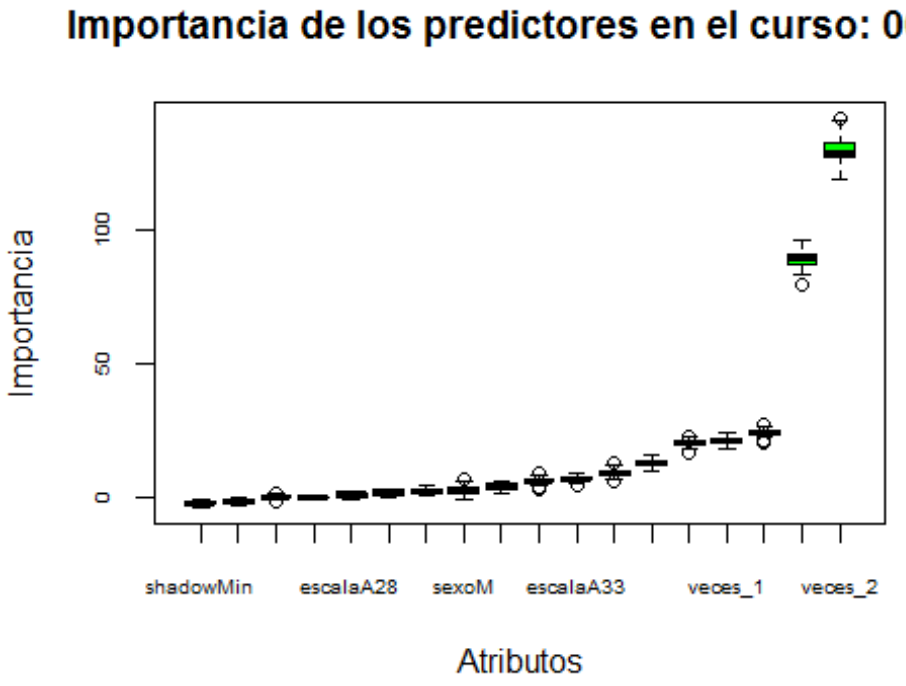



```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0010
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
10"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0010"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0010"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0011*****"
## 'data.frame':    6206 obs. of  16 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
```

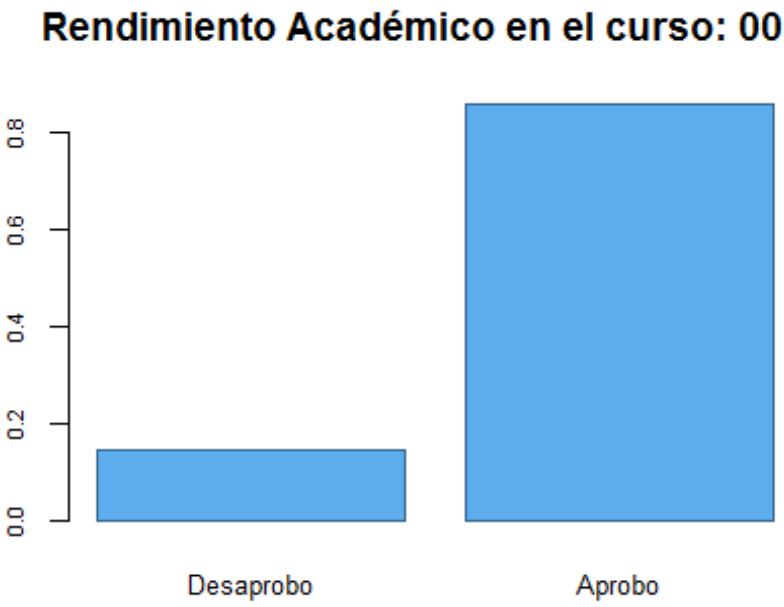
```
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ eva_fin_1: int  14 16 18 16 14 17 14 15 14 16 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_2  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_2   : int  15 15 6 15 14 10 9 8 15 9 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...
##  $ sem_dif_2: int  12 12 15 12 12 15 10 10 10 11 ...
## NULL
```



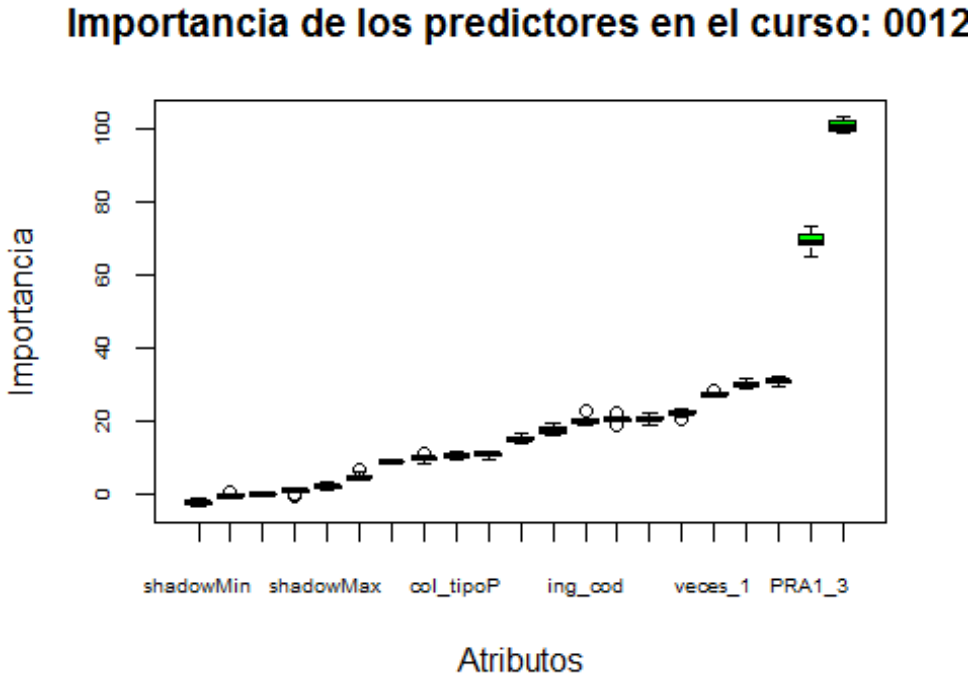

```
##
##       No       Si
## 21.35788 78.64212
```




```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0011
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
11"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0011"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0011"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0012*****"
## 'data.frame':    6527 obs. of  19 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ eva_fin_1: int  15 15 15 17 18 15 14 15 13 15 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ eva_fin_2: int  14 14 13 16 12 12 13 14 17 16 ...
##  $ veces_2  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_3  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_3   : int  14 9 14 15 13 14 14 15 15 13 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...
##  $ sem_dif_2: int  11 11 14 11 11 14 11 11 11 11 ...
##  $ sem_dif_3: int  12 12 14 12 12 15 12 12 11 12 ...
## NULL
```

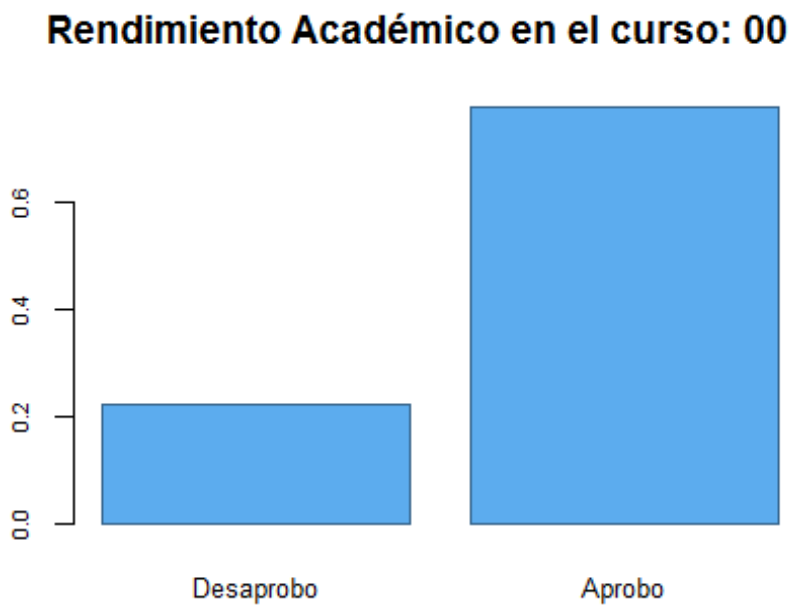



```
##
##       No       Si
## 14.28884 85.71116
```

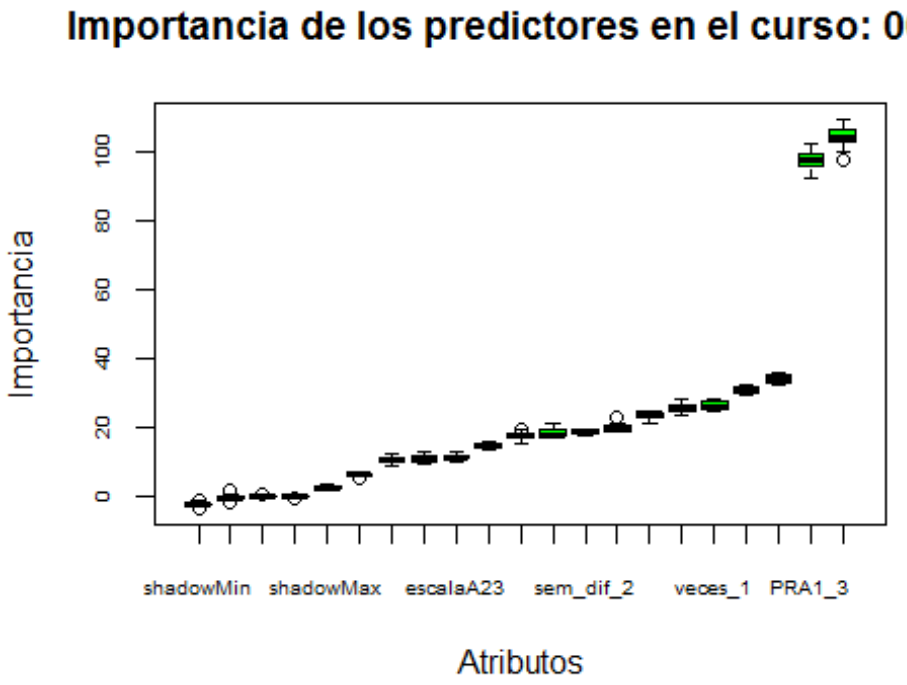



```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0012
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
12"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0012"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0012"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0013*****"
## 'data.frame':    6968 obs. of  19 variables:
##  $ car_cod  : int  11 33 11 11 11 11 11 11 11 11 ...
```

```
##  $ sexoM    : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ ing_cod  : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ escalaA18: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA23: int  1 0 1 1 1 1 1 1 1 1 ...
##  $ escalaA28: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ escalaA33: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ escalaA38: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ col_tipoP: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ eva_fin_1: int  15 15 15 17 18 15 14 15 13 15 ...
##  $ veces_1  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ eva_fin_2: int  14 14 13 16 12 12 13 14 17 16 ...
##  $ veces_2  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ veces_3  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ aprobo   : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PRA1_3   : int  15 13 11 17 20 14 16 16 14 12 ...
##  $ sem_dif_1: int  12 12 15 12 12 15 12 12 12 12 ...
##  $ sem_dif_2: int  11 11 14 11 11 14 11 11 11 11 ...
##  $ sem_dif_3: int  13 13 15 13 13 16 13 13 13 13 ...
## NULL
```




```
##
##       No       Si
## 22.17668 77.82332
```




```
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0013
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
13"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0013"
## [1] "Ejecutando el Modelo: Stacking en el curso: 0013"
## [1] "Calculando Importancia de las variables en los modelos: "
## [1] "***** Fin de Ejecución *****"
```

Anexo 15: Resultados de la generación de los modelos de cada curso del Programa de Estudios Básicos (PEB).

Procederemos a generar los indicadores de los modelos en cada Curso, para poder realizar una evaluación de los mismos.

```
# Para limpiar el workspace, por si hubiera algun dataset o informacion carga
da
rm(list = ls())
par(bg = "gray85")

# Uso de Librerías
library(caret)
library(gmodels) # CrossTable
library(InformationValue) # ks_stat
library(caTools) # colAUC
library(ModelMetrics) # logLoss
library(lattice) # splom
```

Para la preparación del Curso con código: "0001" sirvase revisar el Anexo 10.

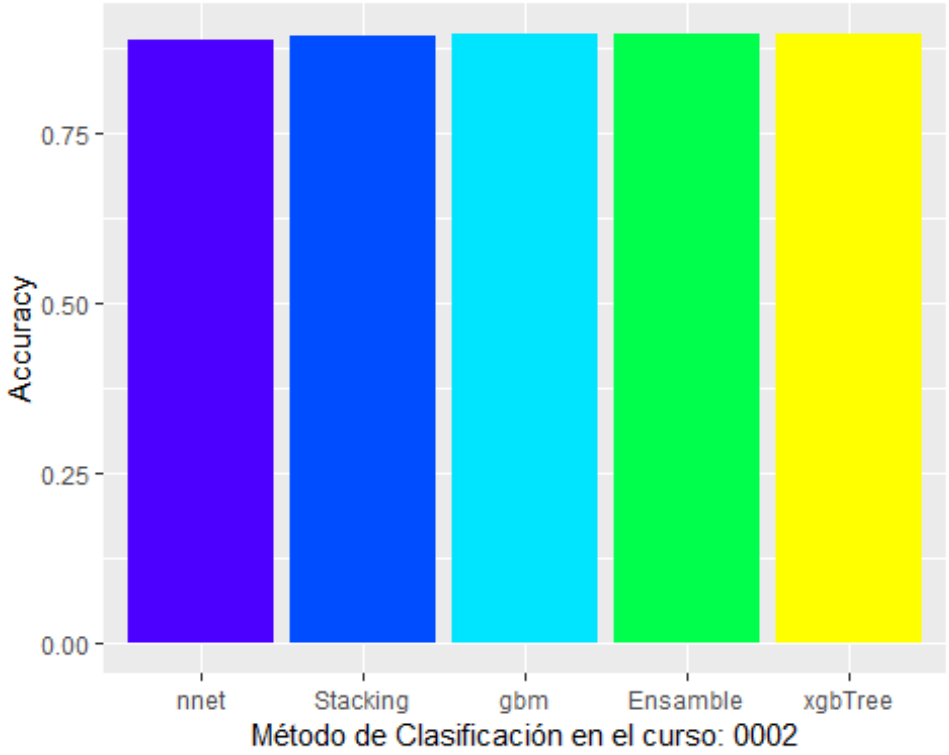
---

1. Generación de los indicadores de los modelos en los Cursos

0002: "Taller de Método de Estudio Universitario".
0003: "Taller de Comunicación Oral y Escrita I".
0004: "Matemática".
0005: "Inglés I".
0006: "Psicología General".
0007: "Lógica y Filosofía".
0008: "Taller de Comunicación Oral y Escrita II".
0009: "Inglés II".
0010: "Formación Histórica del Perú".
0011: "Recursos Naturales y Medio Ambiente".
0012: "Realidad Nacional".
0013: "Historia de la Civilización".

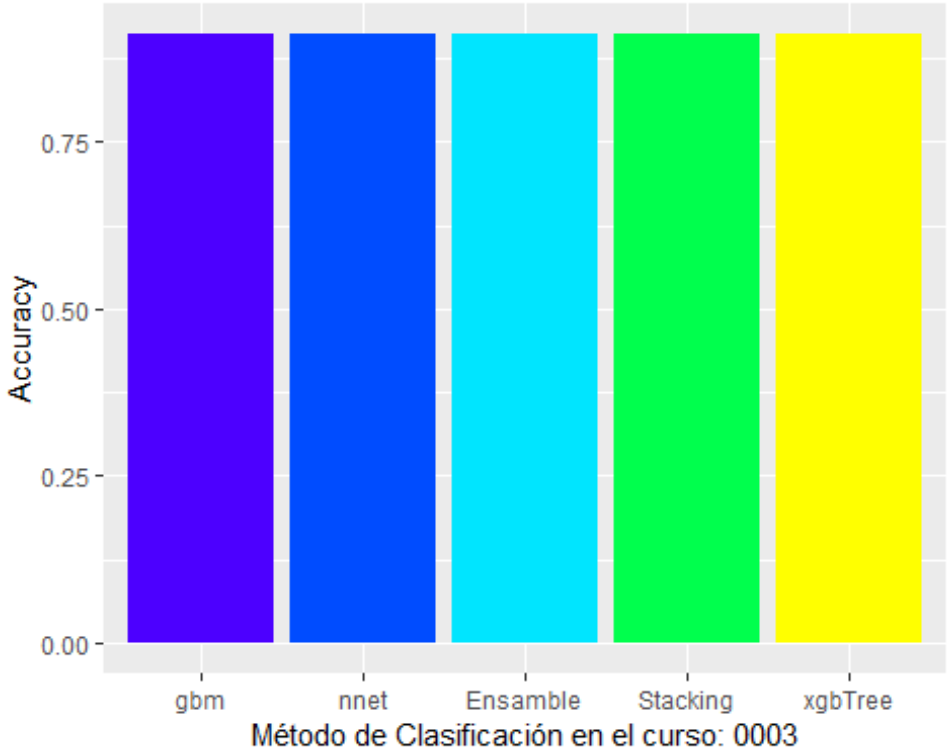
```
## [1] "***** Inicio - Curso: 0002*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0002
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
02"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0002"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0002"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0002"
```



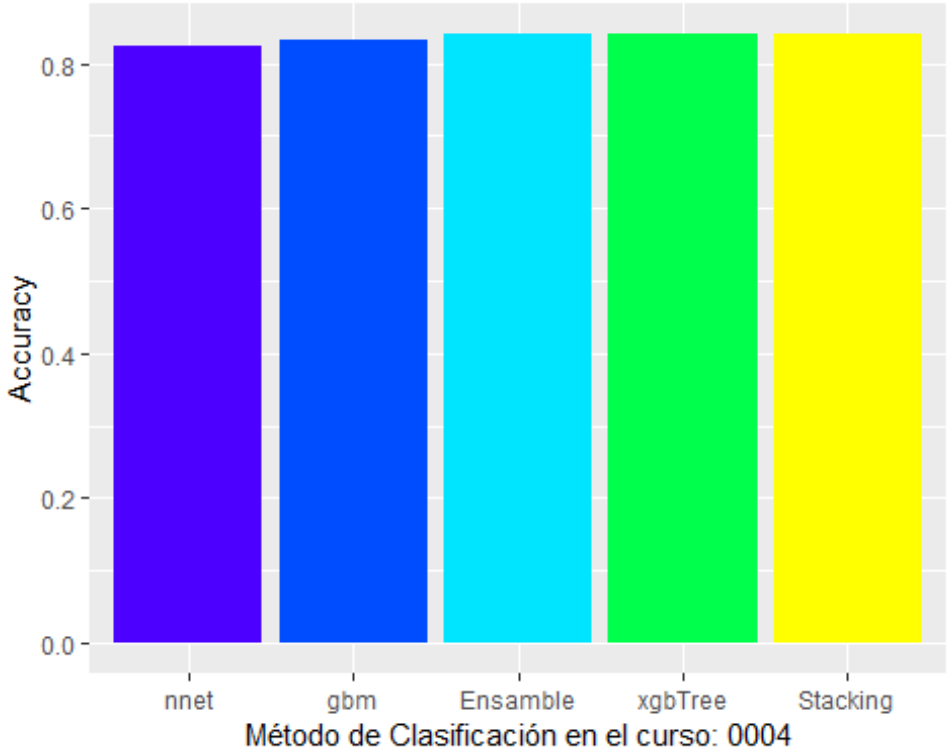

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0003*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0003
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
03"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0003"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0003"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0003"
```



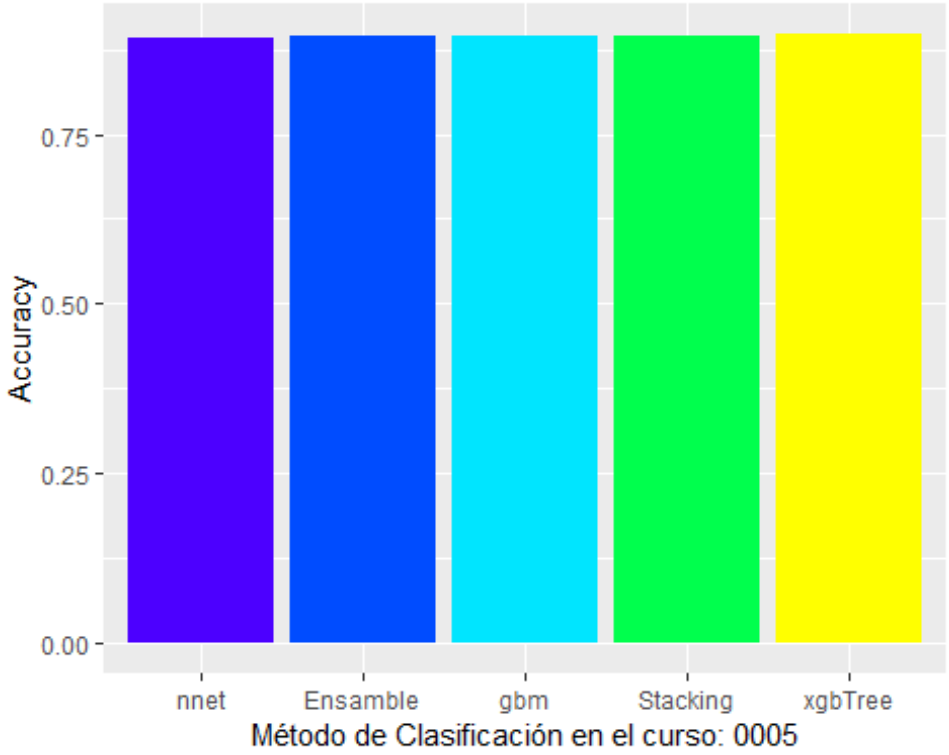

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0004*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0004
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
04"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0004"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0004"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0004"
```



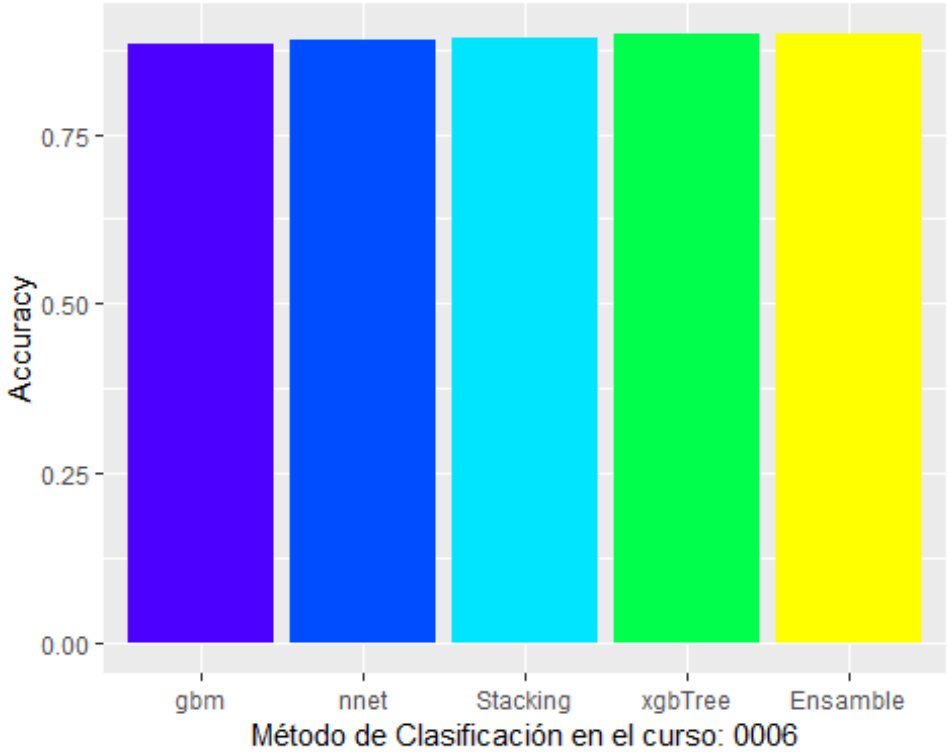

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0005*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0005
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
05"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0005"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0005"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0005"
```



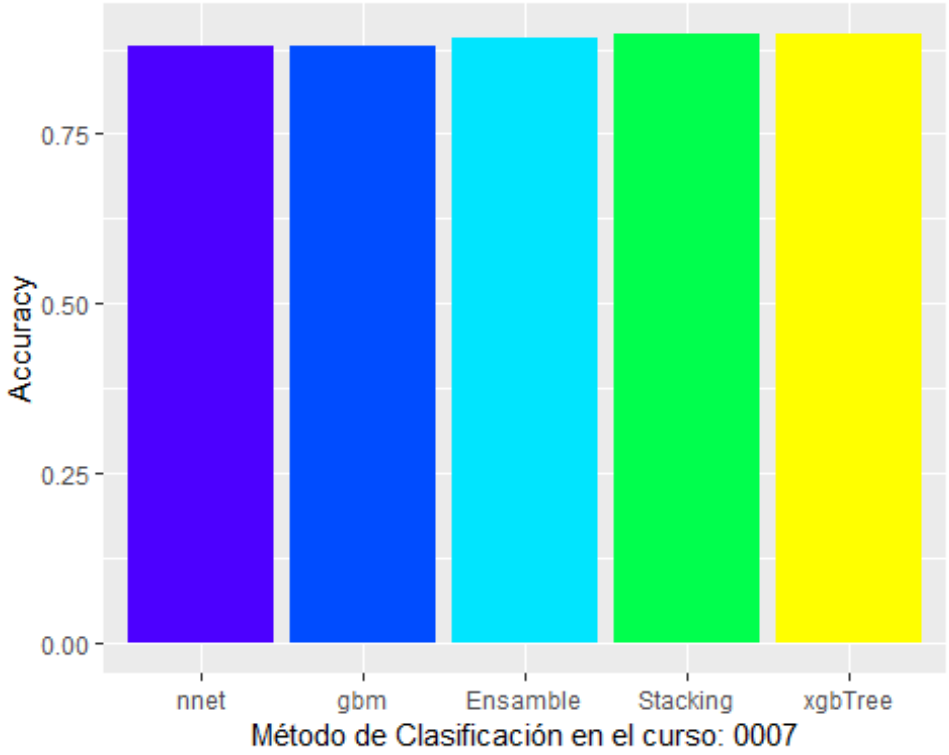

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0006*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0006
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
06"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0006"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0006"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0006"
```



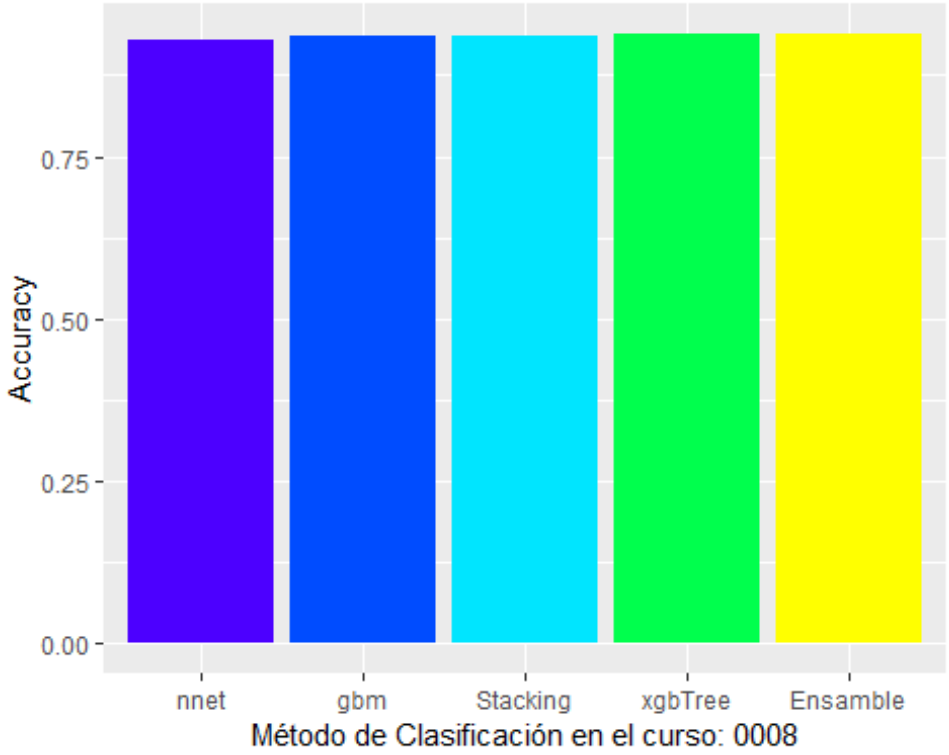

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0007*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0007
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
07"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0007"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0007"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0007"
```



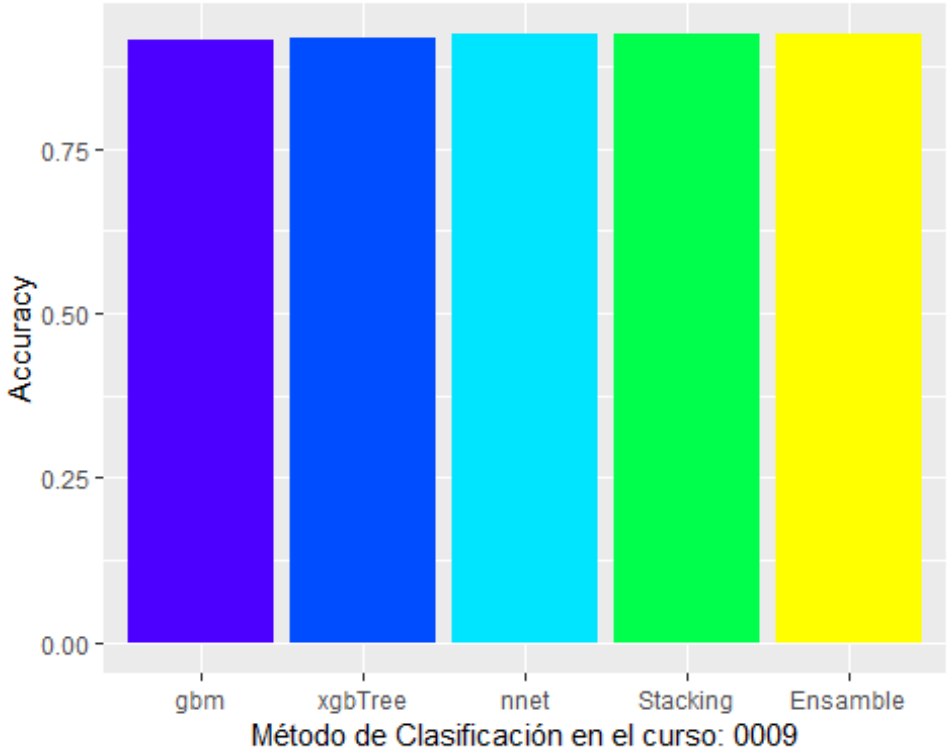

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0008*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0008
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
08"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0008"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0008"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0008"
```



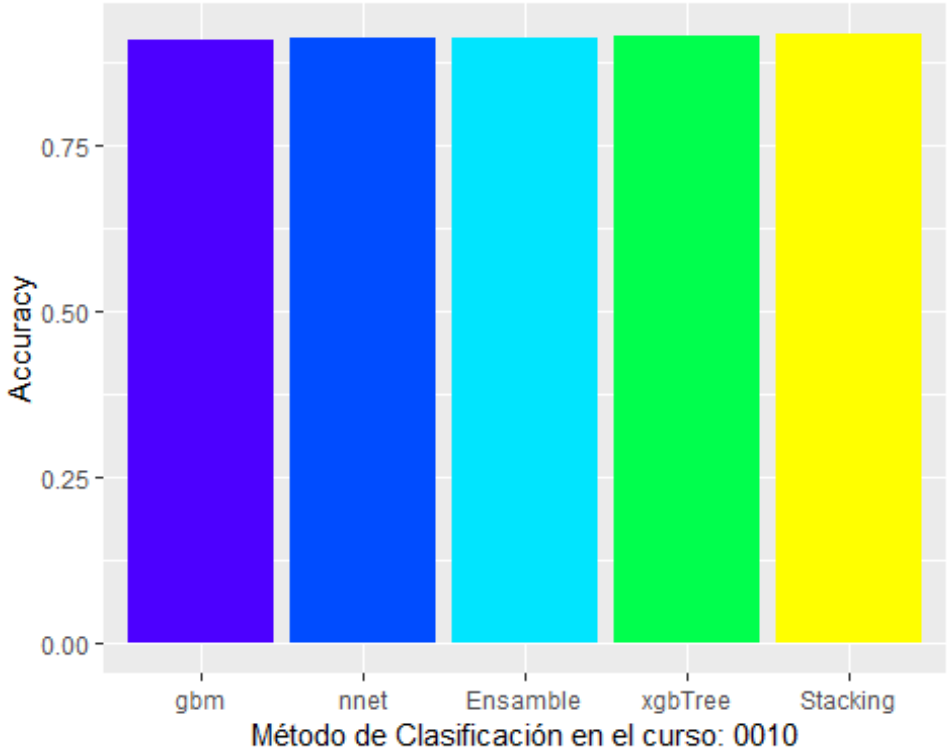

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0009*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0009
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
09"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0009"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0009"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0009"
```



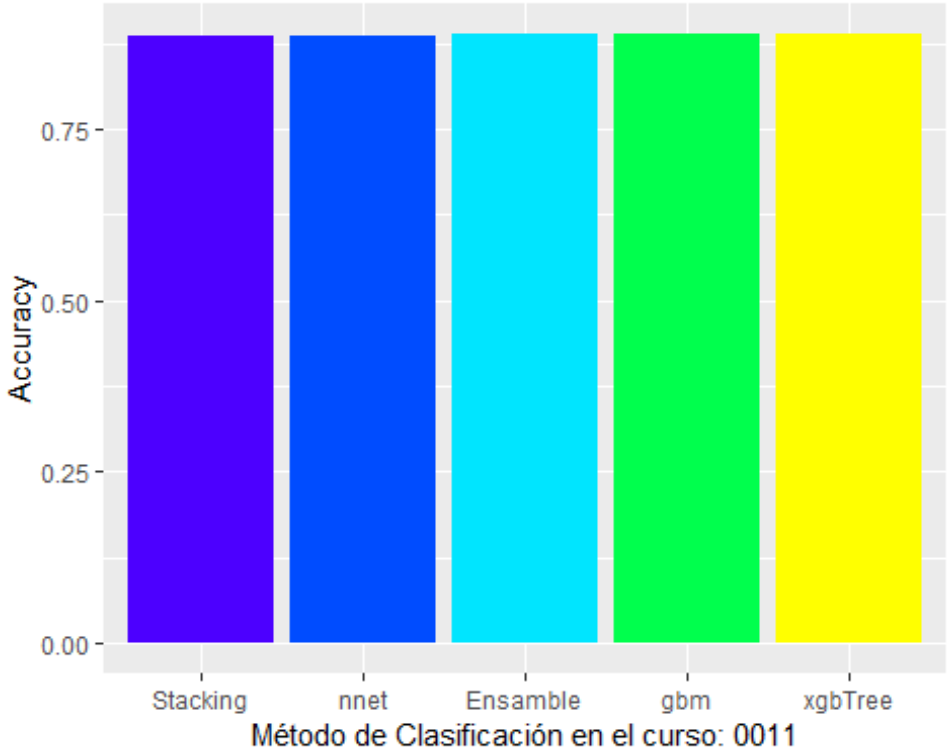

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0010*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0010
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
10"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0010"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0010"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0010"
```



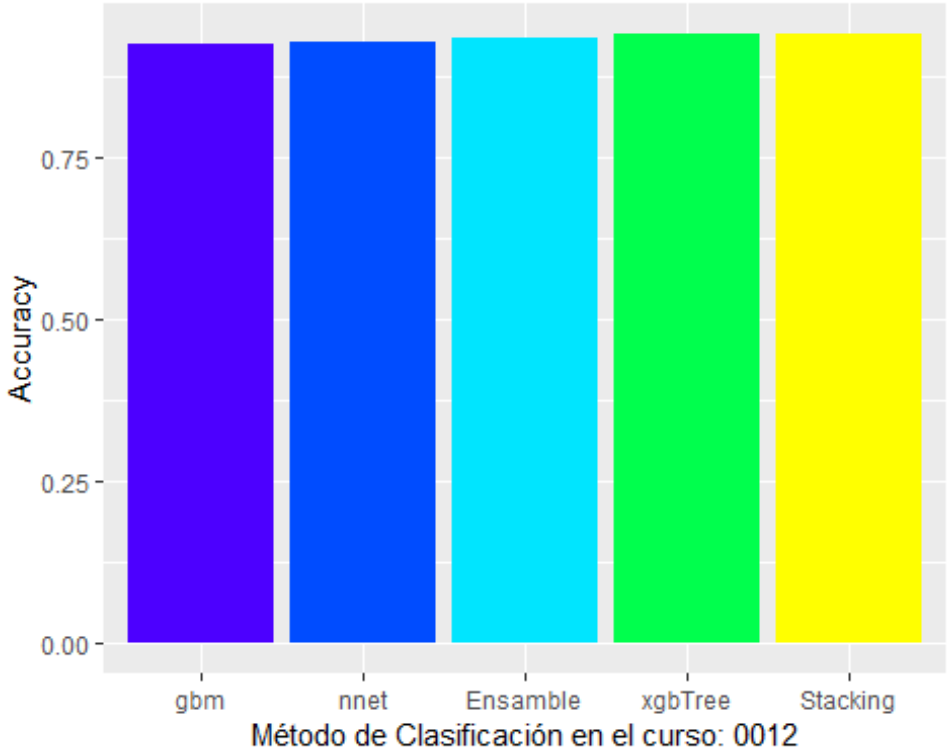

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0011*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0011
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
11"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0011"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0011"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0011"
```



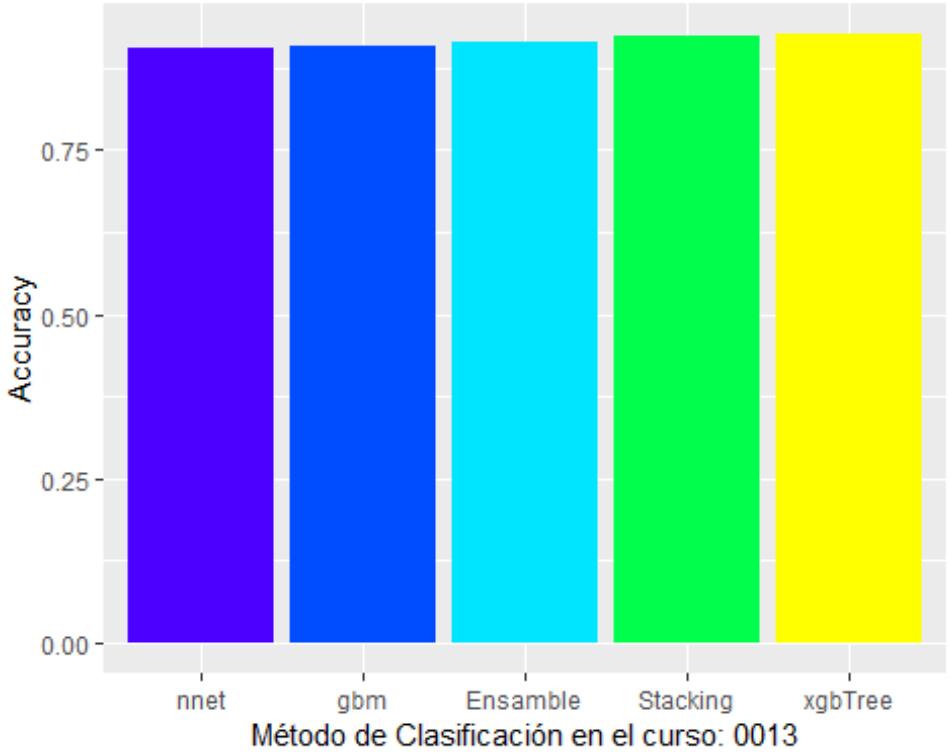

```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0012*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0012
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
12"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0012"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0012"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0012"
```




```
## [1] "***** Fin de Ejecución *****"
##
## [1] "***** Inicio - Curso: 0013*****"
## [1] "Ejecutando el Modelo: Red Neuronal Artificial (RNA) en el curso: 0013
"
## [1] "Ejecutando el Modelo: Gradient Boosting Machine (GBM) en el curso: 00
13"
## [1] "Ejecutando el Modelo: XGBoosting en el curso: 0013"
## [1] "Ejecutando el Modelo: Ensamble en el curso: 0013"
```

```
## [1] "Ejecutando el Modelo: Stacking en el curso: 0013"
```




```
## [1] "***** Fin de Ejecución *****"
```