UNIVERSIDAD RICARDO PALMA

ESCUELA DE POSGRADO

MAESTRIA EN CIENCIA DE LOS DATOS

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

"Clasificación de aceptación de campañas para una entidad financiera, usando random forest con datos balanceados y datos no balanceados"

Autor: Bach. Cárdenas Garro José Antonio

Asesor: Mg. Salinas Flores Jesús

LIMA – PERÚ
2019

JURADO

Mg. Renzo Bustamante Avanzini
Presidente

Mg. Enver Tarazona Vargas
Jurado

Mg. Alfredo León Aguilar
Jurado

AGRADECIMIENTOS

Por la Vida

A Dios y a mis Padres

Por el esfuerzo realizado, por su paciencia y apoyo constante:

Mi Madre, Abuelita y novia

Por creer en mí y alentarme siempre a seguir adelante:

Dr. Erwin Kraenau Espinal

Mg. Ofelia Roque

Mg. Jesús Salinas

Y a cada una de las personas que me apoyaron, colaboraron y contribuyeron para hacer posible la realización de este trabajo.

INDICE

Listado de tablas y figuras

Listado de tablas:

Listado de figuras:

Resumen

En este trabajo de tesis se planteó abordar un enfoque de modelamiento de aprendizaje supervisado de clasificación mediante el modelo de random forest, se utilizó la librería h2o, que permitió tener una comparación de los modelos planteados dando un balanceo de la variable respuesta (*target*) y sin balancear y asimismo ejecutar en un menor tiempo estos modelos, puesto que la librería trabaja en procesamiento en paralelo, también realizar el *tuning* de parámetros del modelo de random forest y compararlos mediante los indicadores de Área Bajo la Curva (AUC), especificidad y sensibilidad.

Los datos a utilizar pertenecen a una entidad financiera en el mes de abril del 2018, donde la variable target es la aceptación de una campaña de tarjeta de crédito.

Los principales resultados obtenidos fueron para el caso del indicador AUC, los 4 modelos planteados obtuvieron similar indicador alrededor de 0.75, en el indicador de especificidad, los mejores modelos fueron los que trabajaron con datos desbalanceados, en el indicador de sensibilidad, los mejores modelos fueron los que trabajaron con datos balanceados. Dado el interés del negocio se escogió un modelo con datos balanceados y con mejor desempeño en la sensibilidad.

Palabras Claves: Aprendizaje supervisado de clasificación, target, random forest, balanceo, AUC, especificidad, sensibilidad.

Abstract

This thesis proposes to focus on a modeling approach of supervised learning using the random forest model. The h2o library was used, which allowed us to have a comparison of the proposed models giving a balance of the response variable (target) and without balancing, while executing these models in a shorter period since this library works in parallel processing. Additionally, it performs the tuning of parameters of the random forest model and compares them using the indicators of Area under the Curve (AUC), specificity and sensitivity.

The data to be used belongs to a financial institution in the month of April 2018, where the target variable is the acceptance of a credit card campaign.

The main results obtained were for the AUC indicator, the 4 proposed models obtained a similar indicator around 0.75, in the specificity indicator. In the specificity indicator, the best models were those that worked with unbalanced data; in the sensitivity indicator, the best models were those who worked with balanced data. Therefore, due the interest of the business, a model with balanced data and with better performance in sensitivity was chosen.

Keywords: Supervised learning of classification, target, random forest, balancing, AUC, specificity, sensitivity.

INTRODUCCIÓN

La presente tesis se refiere al uso de técnicas de modelamiento para clasificación, como lo es el modelo de random forest, el cual se usará mediante una librería (h2o) que fue creado para open source, en este caso se empleó el software R, esta librería tiene como una de sus grandes virtudes, lo que es el trabajo en paralelo; es decir, dividir el funcionamiento del algoritmo utilizado en varios cores del computador y así tener una estimación de los resultados en un tiempo menor, asimismo tiene dentro de sus funciones para los algoritmos incluir como parámetro si se va a balancear o no en base a la target, el cual ayudará a tener una comparación de las mismas.

La investigación aportará a la entidad financiera, a identificar, quién de sus futuros clientes potenciales (leads) aceptarían la campaña de ventas de tarjeta de crédito propuesta por el área de CRM campañas y así poder obtener mejores ganancias y por ende un ahorro para la empresa puesto que se definirá que canales de atención se dirigirán a los mejores leads y que otros canales a los que el modelo refiera que no aceptarán la campaña.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción del Problema

Xie, Li, Ngai & Ying (2008) y Hitzia (2017), mencionan que uno de los problemas más comunes en el ámbito financiero y empresarial es obtener un buen método de clasificación de la target o variable respuesta es decir etiquetar a un sujeto u objeto mediante ciertas características que lo describen, dependiendo mucho del objetivo que tiene trazado la empresa, para otorgar una campaña a un individuo que quizá no desee, generará un gasto y este puede ser evitado, creando un buen modelo predictivo y que indique probabilísticamente si acepta o no la campaña y así la institución financiera generaría un ahorro y quizá posteriormente invertir ese ahorro en futuros proyectos. Si se clasifica a un sujeto u objeto, a partir de parámetros o patrones medidos u observados, esa clasificación está asociado a un error, esto lleva a pensar en una metodología probabilística, que permite medir o cuantificar el error asociado, pues ahora se analizará y se hará la siguiente pregunta: ¿Qué tipo de método de clasificación se empleará?

1.2. Formulación del problema

Esta investigación va orientado a los modelos del aprendizaje supervisado para variable de respuesta cualitativo es decir un modelo de clasificación usando el algoritmo de random forest, donde la aplicación dará respuesta a las siguientes preguntas:

1.2.1. Problema general:

¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera?

1.2.2. Problemas específicos:

a. ¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador del Área Bajo la Curva (AUC)?
b. ¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de especificidad?
c. ¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de sensibilidad?

1.3. Importancia y Justificación del Estudio

El estudio realizado ayudará a la entidad financiera a una mejor toma de decisiones para la priorización de leads en la aceptación de una campaña de tarjeta de crédito asimismo la importancia de un modelo de clasificación mediante técnicas de balanceo.

1.4. Delimitación del Estudio

Los datos que van a ser analizados son de una entidad financiera que ofrece productos financieros vía telefónica u otros canales, para el año 2018 en el mes de abril.

1.5. Objetivos de la Investigación

1.5.1. Objetivo general

Comparar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera.

1.5.2. Objetivo específicos

- ✓ Evaluar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador del Área Bajo la Curva (AUC).
- ✓ Evaluar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de especificidad.
- ✓ Evaluar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de sensibilidad.

CAPÍTULO II: MARCO TEÓRICO

2.1. Marco histórico

Antecedentes Nacionales

Sindia (2016), partió del problema ¿Cuáles son los principales factores que influyen en los accidentes de tránsito fatales en Lima Metropolitana? mediante los resultados de la importancia de variables de los modelos de minería de datos usando algoritmos como: random forest, boosting, y árbol de decisiones CART, donde se encontró con el problema de la data desbalanceada, y solo uso el método tradicional para el balanceo de datos mediante la técnica SMOTE, para darle solución a este inconveniente y así utilizar bien sus algoritmos y obtener resultados confiables.

Medina & Ñique (2017), mencionan en su artículo la aplicación del modelo de aprendizaje supervisado como el random forest como una extensión de los árboles de clasificación, detalla la metodología del random forest como el conjunto de árboles mediante el algoritmo CART que aleatoriamente y bajo votación del conjunto se clasifica al individuo, la aplicación se realizó con una base de datos libres otorgada por Kaggle, y en los software libres R y Python.

En el estudio realizado por Hitzia, detalla el problema que se sufre actualmente en todos los bancos que el *CHURN* o fuga de clientes, el cual presenta un desbalanceo de datos y compara dos técnicas para obtener el mejor pronóstico, una de ellas el de regresión logística y un modelo de *machine learning* como lo es el de random forest y compara indicadores para medir la eficiencia de ambas técnicas (Hitzia, 2017).

Antecedentes Internacionales

Moreno (2001), demostró que las técnicas de minería de datos se están utilizando para la obtención de patrones en los datos y la extracción de información valiosa en el campo de la Ingeniería del Software, para dicha investigación, se utilizó árboles de decisión con las diferentes características del desarrollo de software, en la predicción de riesgos de mantenimiento en módulos de programa.

Alfaro, Gamez & Garcia (2002), detallan el método boosting como aquel conjunto de datos que servirá para el aprendizaje del modelo va a ser capaz de predecir mediante un clasificador una nueva observación, donde va a depender su buena predicción de la calidad del método que se ha utilizado y asimismo de la dificultad de aprendizaje de los datos donde se busca que el clasificador que se obtenga trate de superar la regla que por defecto busca encontrar algún patrón de comportamiento para tratar así de conseguir esa ventaja. El método boosting es un método que busca aumentar la precisión del clasificador utilizado precisamente sacando lo máximo de la ventaja conseguida y de esta manera el método de clasificación utilizado como una función para producir un clasificador que obtenga una precisión alta del conjunto de entrenamiento utilizado.

Heras, Rodríguez & Hernández (2003), plantean una aproximación de la clasificación de asegurados en una cartera de compañía de seguro de automóvil, para lo cual se ha utilizado la clasificación del algoritmo de Máquina de Soporte Vectorial (SVM), para su solución del caso con una eficiente selección de factores de riesgo que se presenta en los asegurados de la cartera, la primera selección se elige a los que tienen mayor riesgo de siniestralidad, uno de los indicadores que establece dicho criterio es la tasa de

clasificación, se plantea también, técnicas del *machine learning* que son herramientas que se establecen con algoritmos genéticos.

A su vez, se desarrolla de manera similar el proceso con el modelo de regresión logística como parte de las conclusiones, los resultados más alentadores son los que muestra la SVM. La problemática de la investigación parte de problemas en la clasificación con múltiples atributos, para lo cual se asigna valores que sirve como máquina de clasificación (red neuronal, algoritmo genético, entre otros) como parte de la selección se ve que toman los grupos de asegurados que presentan o no siniestro en el periodo de un año.

Cardona (2004), aplica árboles de decisión para modelos de riesgo en instituciones financieras según lo regula la Superintendencia Bancaria en Colombia. Los árboles de decisión sirven como herramienta para el cálculo de probabilidades de incumplimiento en crédito, esto se define como pérdida potencial y se debe a la incapacidad de la contraparte de cumplir sus obligaciones, esto lleva a la necesidad de cuantificar la pérdida. En el proceso de la investigación, se define dos tipos de modelos para predecir la probabilidad de incumplimiento de pago, los cuales son: modelo de otorgamiento, en este modelo de iniciación, contiene 6 nodos terminales o 6 variables de interés para esa investigación, en los cuales permite identificar 6 perfiles de riesgo, los cuales son: muy bueno, bueno, regular, malo y muy malo, mientras que el segundo tipo de modelo es el comportamiento, con el que se controla la maduración del crédito.

Para esta investigación se utilizaron los arboles de decisión binarios, que son métodos no paramétricos. Los árboles de decisión se presentan como herramienta efectiva para la predicción de incumplimiento, y esto ayuda a una mejora en la provisión de la entidad financiera, y el grado de morosidad del lead, en resumen el artículo nos habla de la importancia del método de árboles de decisión para la clasificación de leads morosos, la

cual este método es muy utilizado en la banca precisamente por la facilidad de entendimiento de la técnica la cual utiliza técnicas lógicas y de correlación para las ramificaciones y la toma de decisiones para el momento de clasificar a los clientes del banco.

Muñoz (2006), refiere el algoritmo de boosting construye clasificadores asignando pesos a los ejemplos o bases de entrenamiento de forma adaptativa o que se modifican mediante el aprendizaje donde las iteraciones se va construyendo un clasificador madre que trata de compensar los errores cometidos por los otros clasificadores usados, para lograr este objetivo se utiliza un conjunto de ponderaciones cuyos pesos son modificados por cada iteración, donde se aumenta los pesos si está mal clasificado y se decrece los pesos si están bien clasificados con esto conseguiremos mejores resultados en las regiones donde han fallado los anteriores clasificadores.

El boosting con reponderación puede utilizar todos los ejemplos que se han ponderado para construir todos los clasificadores o también podría usarse el boosting con resampling que es remuestreo ponderado donde este último hay una gran probabilidad de que aparezca en la muestra los ejemplos con mayor peso.

En todos los casos mencionados el algoritmo base utilizado se encuentra con el conjunto de datos de entrenamiento con distintos pesos, donde cada nuevo clasificador individual utilizado se concentra en la clasificación de los ejemplos donde hay mayor problema de clasificación o que han sido mal clasificados por los clasificadores utilizados.

Una de las deficiencias del boosting según cuenta algunos autores es que tienen problemas con datos que presentan ruido o error de asignación de etiquetas de la clase de estudio.

López (2008), detalla la importancia de trabajar con datos balanceados puesto ayuda a la precisión del pronóstico otorgado por un clasificador, en la técnica de balanceo utilizo el dos enfoques: el sobre-muestreo y el sub-muestreo y comparó los resultados mediante indicadores como sensibilidad y la precisión global del modelo.

Xie, Li, Ngai & Ying (2008), plantearon una investigación donde resaltan la importancia de *CHURN* o fuga de clientes en los bancos, puesto a los bajos recursos que cuenta el banco estudiado, encontrar un modelo de *CHURN* es primordial para ese país, donde se encuentran un target desbalanceado y plantean un método de aprendizaje de mejora llamado (IBRF), su principal virtud de esta metodología es que da una mejora en las características o atributos del modelo y van iterativamente aprendiendo, cambiando la distribución de clases y poniendo mayores sanciones a la clasificación errónea de la clase minoritaria.

Cárdenas-Montes (2010), menciona que existen metodologías que se basan en la combinación de modelos con el objetivo de la mejora de la precisión de los modelos de clasificación, teóricamente estos métodos son llamados métodos de ensamble puesto que combinan los resultados de diferentes técnicas para obtener una sola estimación. El método *bagging* (*bootstrap agregation*) es uno de los más conocidos gracias a que reduce la varianza y ayuda a reducir un problema común en diferentes métodos el sobreajuste.
El método *bagging* consiste en promediar varios modelos en general en el cual obtenemos un mejor ajuste o score que cuando utilizamos un modelo de algoritmo de aprendizaje, donde la idea principal es obtener un remuestreo de los datos y obtener las estimaciones de los datos de entrenamiento que están siendo re-muestreados. Al obtener estos scores

se corrigen aquellas predicciones que tienen sesgo como asimismo como los que tienen variabilidad alta.

Considerar que si el algoritmo predice datos categóricos, ya sea nominalmente u ordinalmente el resultado será otorgado por mayoría de la clase que domina, mientras tanto si se trabaja con la variable respuesta cuantitativa este se realizara con el promedio de todas las predicciones obtenidas con los otros métodos.

Algo que aconseja el autor del paper en mención es que el método *bagging* ofrece una mejor precisión de los clasificadores individuales que se ha utilizado, sobre todo si el modelo tiene un sobreajuste pero no obtendría un buen resultado para modelos que tienen un alto underfit o bias.

Asimismo el modelo es más robusto puesto que el compuesto reduce la variabilidad de los modelos de clasificación usados individualmente, y el valor agregado de este método es muy bueno para pequeñas variaciones de los datos de entrenamiento.

El autor da como ejemplo es que por ejemplo el *bagging* produce mejora significativa en el método de árboles de decisión pero no en los de vecinos más cercanos (KNN), el bootstrap utilizado en los datos de entrenamiento son aproximadamente un 63.2%, pero considerar que si se usa conjuntos no pequeños de muestras, el modelo final habría utilizado todos los datos, es decir una observación podría estar en más de la mitad de las muestras *bootstrap*.

Maude (2010), menciona la definición de *bagging* proviene por el acrónimo bootstrap agregating, el cual significa un agregado de remuestreos, la idea esencial es construir N clasificadores como base, donde cada uno de ellos va a utilizar el mismo método de clasificación (algoritmo), la única diferencia es que utilizaran distintas bases de entrenamiento.

Cada base de datos de entrenamiento se va a obtener bajo un remuestreo con reemplazo (CR), con un porcentaje determinado del conjunto de datos original. Una vez utilizado el algoritmo en todas las bases de entrenamiento se realiza las predicciones y a partir de estas se toma como resultado aquella clase con más votos (variables cualitativas) y el promedio si estas son variables cuantitativas. Para que la predicción obtenida mediante el método *bagging*, este debe de ser sensible a las variaciones lo recomendable es que sean pequeñas del conjunto de entrenamiento por ejemplo los arboles de decisión.

Una de las desventajas del *bagging* es precisamente que solo se enfoca en reducir la variabilidad (varianza) más no la componente de bias. En su documento menciona otros tipos de métodos como el AdaBoost o Boosting que contemplan precisamente la componente mencionada anteriormente.

Sánchez (2014), presentó un método de balanceo para tratar de dar una solución al problema de clasificación multi-instancia, cuyo objetivo es encontrar un modelo matemático que permita clasificar con el menor error posible, donde la variable respuesta o target esta desbalanceada. Su estudio radica principalmente en utilizar un algoritmo nuevo para dar solución al desbalanceo de datos, el método empleado es el de MISMOTE, la cual consiste en modificar la distribución de los datos que son utilizados al entrenamiento, replicando los datos de las clase minoritaria para equilibrar los datos de la variable de interés, obtuvo en sus resultados un buen pronóstico por ende un menor error de clasificación, por lo que ayuda a entender no necesariamente los métodos clásicos de balanceo son los únicos para dar solución al problema de desbalanceo.

Puente, López & Cruz (2014), presentan en su investigación métodos rápidos de procesamiento para clasificación en conjuntos de datos no balanceados, detallan el gran

problema que tienen los modelos de clasificación para poder predecir la categoría de interés de la target cuando esta esta desbalanceada, puesto los modelos tradicionales para el aprendizaje supervisado siempre otorga un pronóstico erróneo, puesto el algoritmo aprende de la categoría mayoritaria, la metodología planeada en la técnica de balanceo como los enlaces Tomek, cuyo tiempo de ejecución es muy reducido con respecto a otras técnicas de balanceo, los resultados son comparados mediante el AUC.

Beltrán (2015), plantea metodologías mediante métodos estadísticos para el *Credit Scoring* y así mejorar la toma de decisiones al otorgar préstamos en una entidad bancaria. Basándose en modelos de Basilea II y III la cual exige probabilidades de default más confiables mediante calidad explicativa, predictiva y discriminante, las métodos utilizados recaen en la minería de datos, como la combinación de clasificadores (modelos propuestos para modelar) y así otorgar un pronóstico más certero, otra metodología planteada en su tesis es la de incluir costos al modelo de clasificación, es decir cuánto es el peso de clasificar mal a un buen pagador y viceversa, y por último la implementación de estos métodos en diferentes lenguajes uno de ellos que destaca es el JAVA.

Hermosilla (2015), enfoca el problema en la creación de la mejor target o variable respuesta para poder utilizar el mejor algoritmo de clasificación y así generar mayor rentabilidad en la industria de telecomunicaciones, las variables a utilizar siempre han sido sociodemográficos y comerciales de sus clientes pero quieren introducir data no estructurada como por ejemplo las redes sociales y así poder innovar en el mundo de telefonía.

Hassinger (2015), presentó una aplicación de *machine learning* aplicado a los accidentes de tráfico, uno de los algoritmos planteados es el de árboles de decisión por su fácil interpretación de reglas que presenta la salida de este algoritmo y así encontrar patrones que pueden ser interpretables para las razones de accidentes de tránsito, obtuvo buenos resultados comparando los distintos árboles que planteo utilizando indicadores como el AUC, sensibilidad entre otros.

Randa, López & Garach (2015), presentaron un modelo de predicción de gravedad de lesiones en un accidente de tránsito mediante clasificadores bayesianos, las variables que usó fueron: número de vehículos implicados, condiciones de la superficie, velocidad, patrón de accidente, número de direcciones y obtuvo como resultado que el uso de los conjuntos de datos balanceados, usando la técnica de **balanceo**: *oversampling* con redes bayesianas mejoraron la clasificación de la gravedad de lesiones en un accidente de tráfico, donde la técnica de balanceo *oversampling* fue la solución para tratar datos desbalanceados.

Pardo (2015), utilizó la técnica de *bagging* que es una combinación de árboles y en promedio te otorga por mayoría de votos una respuesta estimada que precisamente es la base del algoritmo de random forest.

Cánovas, Alonso & Gomariz (2016), detallan como utilizaron random forest como clasificador de imágenes y explica las ventajas que tiene este algoritmo para la estimación interna de exactitud mediante validación cruzada. Utilizaron coberturas del suelo con áreas de entrenamiento compuesta por varios pixeles (la cual se descompone en una

matriz de números que otorga estos pixeles), la idea del estudio es modificar los parámetros del algoritmo para que la predicción obtenga una mejor acierto.

De Juan (2017), pudo identificar un algoritmo que obtenga una mayor precisión o exactitud para la detección de patrones de aquellos clientes que no realizan el pago de sus tarjetas de créditos, el estudio utilizó metodologías mediante el aprendizaje supervisado y no supervisado, así como también técnicas para el desbalanceo de datos, adicionalmente para ver cuál es el algoritmo que mejor precisión obtiene, para comparar los modelos utilizó indicadores como precisión o accuracy, sensibilidad, AUC, el tiempo de ejecución en cada modelo y asimismo la complejidad en una futura implementación.

Arnejo (2017), detalla en su investigación el gran problema de trabajar modelos con clases desbalanceadas, puesto genera que el clasificador empleado aprenda más de la categoría con mayor proporción en la target y así otorgue pronósticos erróneos, la aplicación del trabajo se concentra en el problema de fuga de clientes y en las metodologías de balanceos de datos para dar solución al problema antes descrito.

2.2. Investigaciones relacionadas con el tema

El algoritmo planteado por Leo Breiman en el año 2001 fue la combinación de árboles de decisión mediante el algoritmo de CART independientes generados a partir de un vector de muestreo aleatorio que usa la misma distribución para todos los árboles de estudio (bagging).

El término random forest se toma de la primera propuesta en el año 1995, este algoritmo está considerado como un clasificador bastante preciso. Trabaja bien aunque haya datos perdidos y ofrece un método para la interacción de las variables (Breiman, 2001).

En el 2008 el Ing. Arturo López Pineda presentó un problema de balanceo de datos en el cual detalla como definición al conjunto de datos está desbalanceado si una clase está representada por un número grande de datos, mientras que la otra clase se representa por muy pocas instancias (López, 2008).

Hitzia, (2017), presentó la técnica de *machine learning* de random forest para el análisis de la fuga de clientes en un banco, en el estudio tomó solo como referencia las bondades de la técnica, no atacando el problema a investigar, razón suficiente para tener en consideración el desarrollo de la problemática planteada en la presente investigación.

2.3. Estructura teórica y científica que sustenta el estudio

2.3.1. Random forest

Es una técnica mejorada de *bagging*, que ayuda a obtener una precisión más alta en la clasificación mediante la incorporación de aleatoriedad en la construcción de cada clasificador individual. Esta aleatorización puede introducirse en la partición del espacio (construcción del árbol), así como en la muestra de entrenamiento.

El algoritmo random forest, a diferencia de *bagging* introduce de forma aleatoria en cada nodo una cantidad de p variables de todas la originales, y de estas selecciona la mejor para realizar la partición.

Se presenta a continuación el proceso del algoritmo:

1) Selecciona individuos al azar (usando muestreo con reemplazo) para crear diferentes conjuntos de datos.
2) Al crear los árboles se eligen variables al azar en cada nodo del árbol, dejando crecer el árbol en profundidad (sin podar).
3) Crea un árbol de decisión con cada set de datos, obteniendo diferentes árboles, ya que cada conjunto contiene diferentes individuos y diferentes variables.

4) Predice los nuevos datos usando el "voto mayoritario", donde clasificará como "positivo" si la mayoría de los arboles predicen la observación como positiva. (Breiman, 2001).

Asimismo el algoritmo de random forest es uno de los algoritmos de clasificación más usados puesto una de sus mejores virtudes es que aporta una estimación interna de exactitud mediante una forma de validación cruzada, aportando conocimiento sobre el procedimiento moderno de trabajo en paralelo que es precisamente la distribución en cores del computador para grandes volúmenes de datos.

2.3.1.1. Estimación del error con random forest

Se define la tasa de error fuera de muestra (OOBi) de una observación, como el error obtenido al ser clasificada por los árboles del bosque construidos sin su intervención es decir dejando fuera a la muestra no modelada.

La estimación OOB del error es el promedio de todos los OOBi para todas las observaciones del conjunto de datos es mejor estimador que el error aparente. Parecida a la estimación por validación cruzada la medida se puede extrapolar al problema de regresión describiéndola en términos del Error Cuadrático Medio (ECM).

2.3.2. Desbalanceo de Datos

El objetivo de un algoritmo de clasificación es intentar aprender un separador o clasificador, que pueda distinguir los dos clases de la target. Hay muchas maneras de hacerlo, basadas en varias suposiciones matemáticas o estadísticas como se muestra en la siguiente figura:

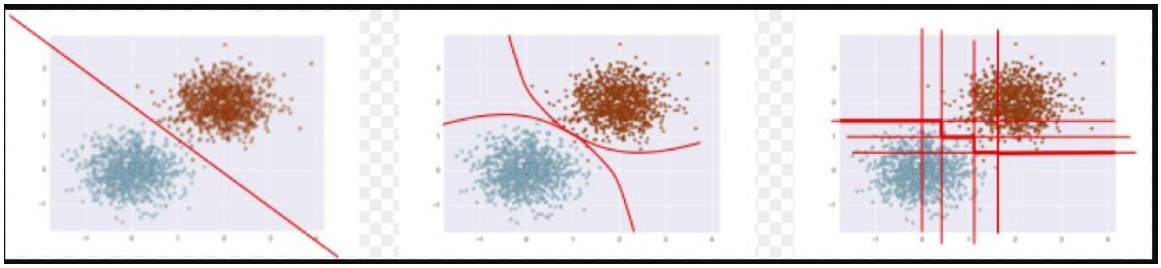


Figura N°1: Clasificadores posibles para separar 2 categorías de la target
Fuente: (Fawcett, 2016)

Pero cuando se comienza a trabajar con datos reales, una de las primeras observaciones que resalta es la desigualdad de proporción de las dos clases de la target, como se muestra a continuación:

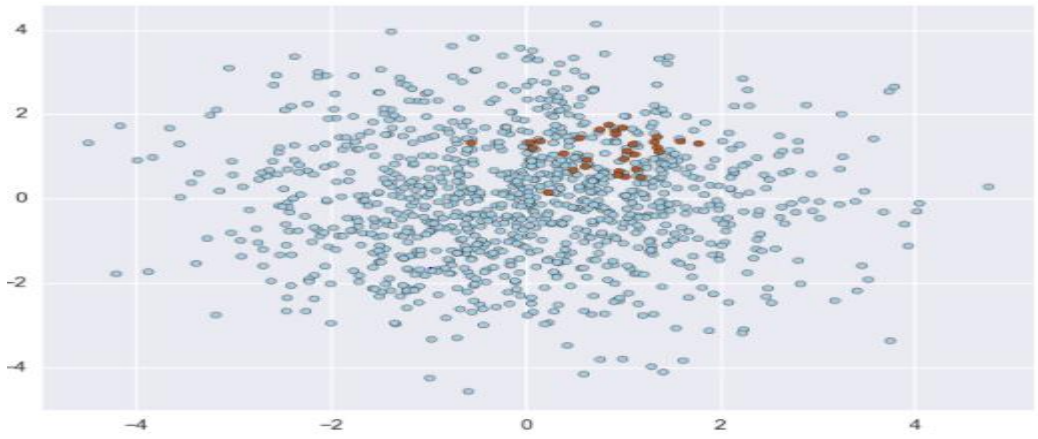


Figura N°2: Clases desproporcionada de la target
Fuente: (Fawcett, 2016)

El problema principal es que estas clases están desbalanceadas: los puntos rojos son superados en gran medida por el azul.

Algunos ejemplos claros donde radica principalmente este problema es:

- ✓ Alrededor del 2% de las cuentas de tarjetas de crédito son defraudadas por año. (La mayoría de los dominios de detección de fraude están muy desbalanceados).

- ✓ El examen médico para una afección generalmente se realiza en una gran población de personas sin esta afección, para detectar una pequeña minoría que lo acompaña (por Ejemplo, La prevalencia del VIH en EE. UU. Es aproximadamente de 0,4%).
- ✓ Las fallas de la unidad de disco son aproximadamente 1% por año.
- ✓ Las tasas de defectos de producción en fábrica normalmente se ejecutan alrededor de 0.1%.

Los algoritmos convencionales o tradicionales a menudo están sesgados hacia la clase mayoritaria porque sus funciones de pérdida intentan optimizar cantidades tales como la tasa de error, sin tener en cuenta la distribución de datos.

En el peor de los casos, los ejemplos de minorías se tratan como valores atípicos de la clase mayoritaria e ignorada. El algoritmo de aprendizaje simplemente genera un clasificador trivial que clasifica cada ejemplo como la clase mayoritaria.

La solución según Fawcett (2016), para este problema es:

1) Equilibre el conjunto de entrenamiento de alguna manera:
   - Sobre muestra de la clase de la minoría.
   - Dé muestras de la clase mayoritaria.
   - Sintetiza nuevas clases de minorías.

2) Replicar los ejemplos de minorías y cambie a un marco de detección de anomalías.
3) En el nivel del algoritmo, o después de él:
   - Ajuste el peso de la clase (costos de clasificación errónea).
   - Ajusta el umbral de decisión.
   - Modifique un algoritmo existente para que sea más sensible a las clases raras.

4) Construya un algoritmo completamente nuevo para obtener buenos resultados en datos desequilibrados.

Las técnicas más comunes que se usa en estos tipos de problemas son: *oversampling* y *undersampling*.

Los enfoques más fáciles requieren pocos cambios en los pasos de procesamiento, y simplemente implican ajustar los conjuntos de ejemplos hasta que estén equilibrados.

2.3.2.1. Oversampling

La técnica replica aleatoriamente instancias minoritarias (clase menor en proporción de la target) para aumentar su población y así equilibrar a la clase mayoritaria, en la siguiente figura claramente se entiende el procedimiento:

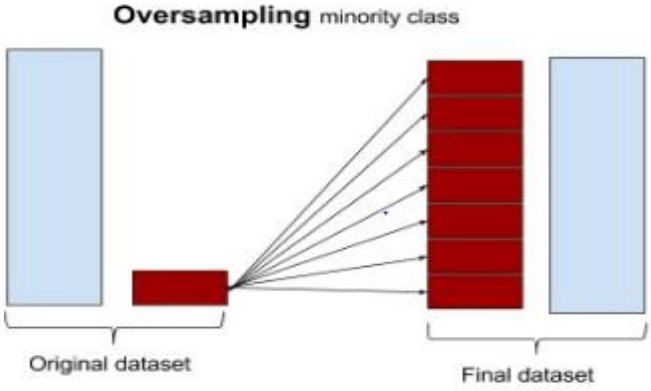


Figura N°3: Funcionamiento del oversampling
Fuente: (Fawcett, 2016)

2.3.2.2. Undersampling

A comparación de la técnica anterior, este proceso realiza lo contrario es decir aleatoriamente reduce la clase de la mayoría hasta completar la clase minoritaria y así equilibrar las muestras para el entrenamiento del modelo de *machine learning*.

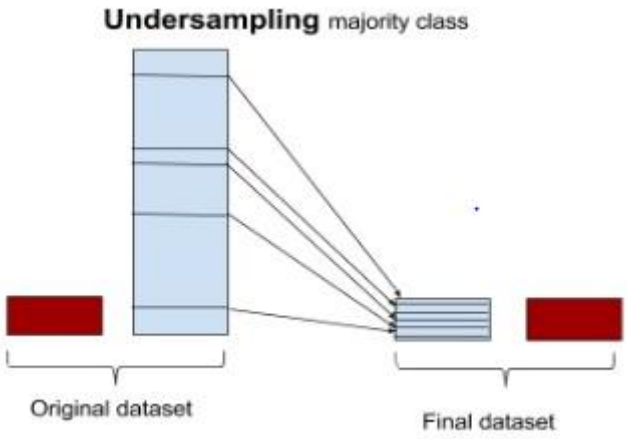


Figura N°4: Funcionamiento del Undersampling
Fuente: (Fawcett, 2016)

Existen otras técnicas recién creadas y pocos explorados para el problema de desbalanceo de datos, el más conocido es SMOTE que detallaremos su funcionamiento de esta técnica en el siguiente punto:

2.3.2.3. SMOTE

SMOTE (Técnica de Sobremuestreo de Minorías Sintéticas) consiste en la síntesis de elementos para la clase minoritaria, basados en los que ya existen. Funciona eligiendo al azar un punto de la clase minoritaria y calcula los k vecinos más cercanos para este punto. Los puntos sintéticos se agregan entre el punto elegido y sus vecinos.

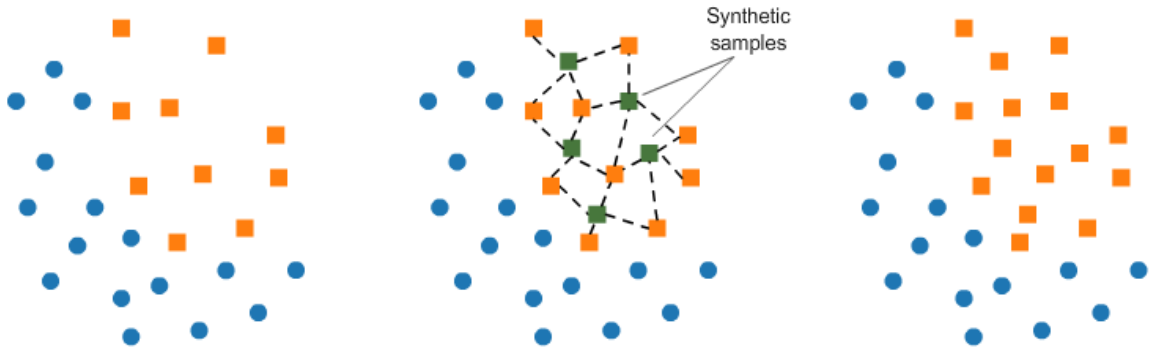


Figura N°5: Funcionamiento del SMOTE
Fuente: (Fawcett, 2016)

2.3.3. Librería H2O

La Libreria H2O es open source, desarrollada en JAVA que trabaja con algoritmos de *machine learning*, en plataformas para procesar grandes volúmenes de datos como Spark y Hadoop, las principales ventajas son:

1) La velocidad, sus principales enfoques son:
    - El tiempo es valioso
    - En memoria es más rápido
    - Distribuido es más rápido.
    - Alta velocidad y precisión
2) Sin muestreo
    - Escala a big data
    - Acceder a los enlaces de datos
    - Utilizar todos los datos sin muestreo.
3) Interfaz de usuario interactiva
    - Modelado basado en web con H2O Flow
    - Comparación de modelos
4) Innovador
    - Suite de algoritmos de *machine learning* de última generación.
    - Aprendizaje profundo y conjuntos

Los principales algoritmos que presenta son los siguientes:

Análisis Estadístico

• Modelos Lineales Generalizados (GLM)

• Modelos proporcionales de Cox

• Regla de Bayes ingenuo o Naïve Bayes

Ensambles

• Random forest

• Árboles distribuidos

• Gradient Boosting Machine

• Ensambles de modelos

Deep Learning

• Redes Neuronales

• Auto redodificación

• Detección de anomalías

• Variables profundas

Clustering

• K-Medias (K-Means)

Reducción de dimensionalidad

• Análisis de componentes principales

• Modelos de rango bajo generalizados

Optimización

• ADMM

• L-BFGS (método cuasi Newton)

• Gradiente del descenso estocástico

Datos transformados

• Transformación Log

Fuente: (Landry, 2018)

2.4. Definición de términos básicos

- ✓ **Modelo de Clasificación:** Modelos que utilizan como variable dependiente a una variable categórica y variables independientes de cualquier tipo de variable

para intentar separar las clases de la target mediante un modelo estadístico o un modelo de *machine learning*.

- ✓ **Default:** Variable de interés en la base de datos también llamada variable dependiente o target.
- ✓ **Leads:** Persona natural que puede ser potencial cliente de la entidad financiera, en la que se aplica una determinada campaña para aceptación de uno de los productos que cuenta la entidad.
- ✓ **Balanceado:** Los datos desbalanceados normalmente se refieren a un problema de clasificación donde las clases de la target no están representadas por igual. **Fuente: (Brownlee, 2015)**
- ✓ **Tabla de clasificación:** La tabla de clasificación muestra la distribución de valores observados y estimados. Los valores observados son los valores reales y los valores estimados se obtienen a partir del modelo de clasificación.

Observado		Pronosticado		
		A		Porcentaje
		1	0	correcto
A	1	a	b	a/(a+b)
	0	c	d	d/(c+d)
Porcentaje Global				(a+d)/(a+b+c+d)

Figura N°6: Tabla de clasificación
Fuente: (Fernández, 2016)

- ✓ **Sensibilidad:** a/(a+b), indica la capacidad que tiene un modelo para clasificar correctamente la categoría de interés de la variable dependiente. (Fernández, 2016)
- ✓ **Especificidad:** d/(c+d), indica la capacidad que tiene un modelo para clasificar correctamente la categoría que no es de interés de la variable dependiente. (Fernández, 2016)

- ✓ **AUC de la Curva ROC:** La curva ROC (Receiver Operating Characteristic), indica que cuanto más alejada este de la diagonal principal mejor es el método de diagnóstico, ya que la curva ROC ideal sería la que con una especificidad de 1 tuviera una sensibilidad de 1, y cuanto más cercana esté a dicha diagonal peor será el método de diagnóstico. Pérez (2015) menciona que la diagonal principal es la que corresponde al peor test de diagnóstico y tiene un área bajo la curva de 0.5. Adicionalmente se ha establecido los siguientes los intervalos para los valores de la curva ROC:

  - [0.5 - 0.6>: Test malo
  - [0.6 - 0.75>: Test regular
  - [0.75 - 0.9>: Test bueno
  - [0.9 - 0.97>: Test muy bueno
  - [0.97 - 1>: Test excelente

  Se plantean las siguientes hipótesis:

  - H0: el área bajo la curva ROC es igual a 0.5
  - H1: el área bajo la curva ROC no es igual a 0.5

  Si se rechaza Ho asociado a un p-valor implica que el modelo ajustado es el adecuado. (Pérez, 2015)

- ✓ **LogLoss:** Mide el rendimiento de un modelo de clasificación, donde la entrada de predicción es un valor de probabilidad del modelo, el indicador mide que tan cercana en promedio esta la probabilidad a la clase real de la variable respuesta. (Buja, Stuetzle & Shen ,2005).

2.5. Fundamentos teóricos que sustentan las hipótesis

López (2008), detalla la importancia de trabajar con datos balanceados puesto ayuda a la precisión del pronóstico otorgado por un clasificador o algoritmo de clasificación.

2.6. Hipótesis

2.6.1. Hipótesis general

El algoritmo de random forest para datos balanceados proporcionará mejores indicadores que para datos no balanceados en el área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera.

2.6.2. Hipótesis específicas

- ✓ Existen diferencias entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados del área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera utilizando el indicador del Área Bajo la Curva (AUC).
- ✓ Existen diferencias entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados del área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera utilizando el indicador de especificidad.
- ✓ Existen diferencias entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados del área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera utilizando el indicador de sensibilidad.

2.7. Variables

La variable dependiente o default Propensión: Indica si el lead acepta o no la campaña.

Las variables independientes o Drivers:

X1: Indica que segmento pertenece el lead.

X2: Indica que tipo de evaluación seguirá el expediente del lead.

X3: Indica la edad del lead.

X4: Indica si la tarjeta de crédito ofrecida es VISA o no.

X5: Línea de crédito de la tarjeta de crédito.

X6: Indica a que grupo de ejecución pertenece el lead.

X7: Indica que propenso es el lead para ser contactado.

X8: Indica que tan propenso es el lead para aceptar el producto.

X9: Número de veces que se ha llamado al lead.

X10: Indica que tan recurrente es el lead en la campaña asignada.

X11: Indica la calidad de lead, el menor valor indica mejor calidad de lead.

III. MARCO METODOLÓGICO

3.1. Tipo, método y diseño de la investigación

La presente investigación fue de corte transversal porque se tomó la muestra de leads en un momento determinado (abril de 2018), es explicativa porque ayudó a describir los datos de los leads permitiendo identificar las variables que caracterizan de manera significativa los leads que no aceptarán la campaña de ventas propuesta por el área de CRM Campañas.

3.2. Población y muestra

El universo fue la población de leads de la entidad financiera que posee una campaña de ventas propuesta por el área de CRM, para la aceptación de la campaña de tarjeta de crédito.

3.3. Técnicas e instrumentos de recolección de datos

El datawarehouse de la entidad bancaria es una base de datos donde se encuentra la información sociodemográfica, socioeconómica y financiera de los leads del mes de cierre de abril del 2018 de todos los leads que posee una campaña de ventas propuesta por el área de CRM, para la aceptación de la campaña de tarjeta de crédito.

3.4. Descripción de procedimientos de análisis

En el estudio se evalúan los indicadores de AUC, sensibilidad y especificidad para los modelos propuestos utilizando datos desbalanceados mediante la Liberia h2o del software R.

CAPÍTULO IV: RESULTADOS Y ANÁLISIS DE LOS RESULTADOS

4.1. Resultados

En este apartado se muestran los resultados obtenidos, acompañados de su respectiva explicación y un análisis profundo. Se pueden utilizar tablas y gráficas para reportar los resultados, si esto facilita su comprensión.

En primer lugar se obtuvieron las estadísticas de resumen de la base de datos para poder entender el comportamiento de las variables:

Tabla N° 1
Estadísticas de resumen de los datos

Nombre	Tipo	Nulos	Promedio	disp	Mediana	mad	min	max	nlevs
ID	integer	-	35,744.50	20,636.95	35,745	26,497.03	1	71,488	-
X1	factor	-	-	0.55	-	-	1,316	31,897	6
X2	factor	-	-	0.22	-	-	389	55,506	5
X3	integer	-	41.41	11.45	40	13.34	1	97	-
X4	factor	-	-	0.30	-	-	21,194	50,294	2
X5	integer	-	8,345.62	11,813.72	3,100	3,113.46	700	100,000	-
X6	factor	-	-	0.84	-	-	231	11,287	16
X7	factor	-	-	0.51	-	-	161	35,291	6
X8	factor	-	-	0.46	-	-	729	38,711	5
X9	integer	-	7.14	6.05	5	4.45	1	58	-
X10	integer	-	2.76	0.77	3	-	1	4	-
X11	integer	-	1.95	1.06	2	1.48	1	6	-
FLAG_APROBADO	factor	-	-	0.09	-	-	6,596	64,892	2

Fuente: Elaboración Propia

- Disp: desviación estándar.
- Mad: desviación media absoluta.
- Min: mínimo valor.
- Max máximo valor
- Nlevs: número de niveles.

Se puede observar en la Tabla **N° 1**, se cuenta con 11 drivers o variables independientes y la target (FLAG_APROBADO), en las variables cuantitativas, la Edad (X3) tiene una mediana de 40 años ese decir el 50% de los datos se encuentra alrededor de los 40 años y la media de la edad es de 41 años por lo que se sospecha que tiende a una distribución simétrica, la línea de crédito en promedio es de 8,345 soles, y su mediana es de 3,100

soles con lo que se llega a sospechar que tiende a una distribución asimétrica, el número de llamadas el promedio es de 7 llamadas y su mediana es de 5 llamadas, mientras tanto las variables cualitativas el flujo de riesgos tiene 5 niveles, la prioridad de ejecución tiene 16 niveles.

Para la modelación se recodificó cada variable respecto a la target, para poder mantener una estabilidad en el modelo propuesto, para ello se utilizó un árbol de decisión mediante el algoritmo CHAID para la recodificación como se muestra a continuación:

Por ejemplo, la recodificación de la variable Edad (X3) :

Variable Edad(X3)

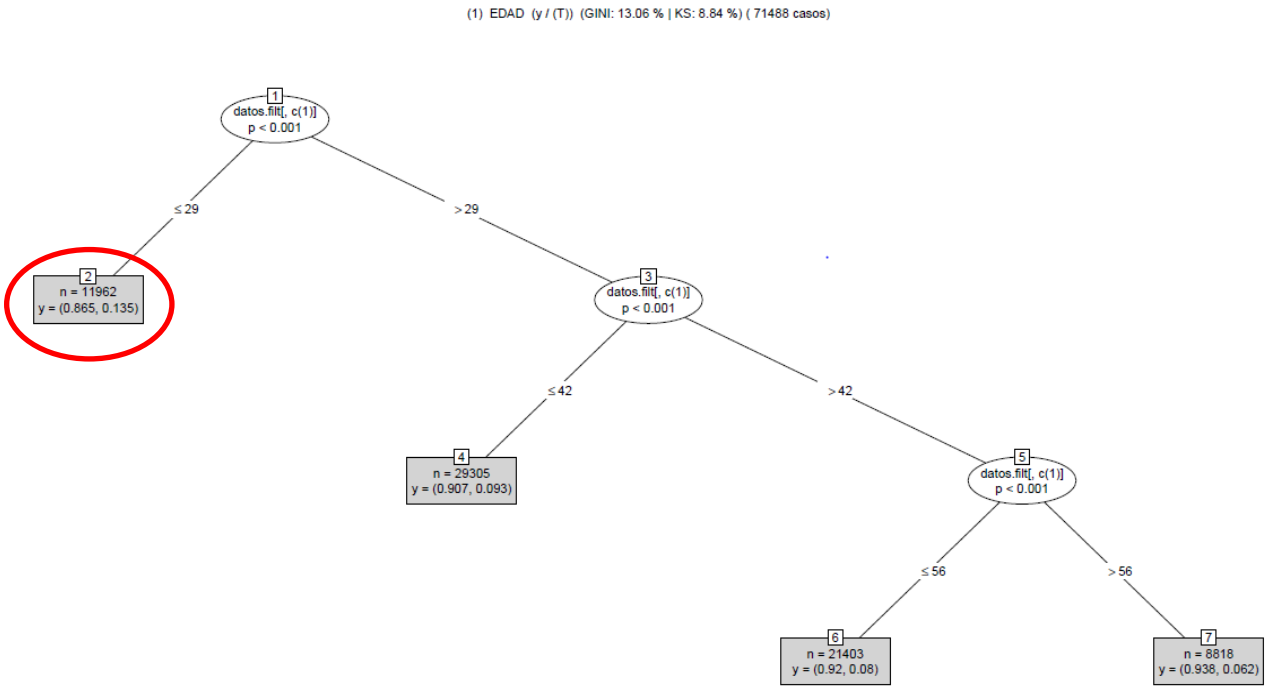


Figura N°7: Árbol de decisión para la variable Edad
Fuente: Elaboración Propia

La utilidad de los árboles de decisión es que ordena cada variable ya sea cuantitativa o cualitativa en grupos con propensión hacia la categoría de interés, por ejemplo el nodo encerrado en un círculo rojo nos muestra en primer lugar la cantidad de casos en el nodo (n= 11,962) , las probabilidades de cada categoría por ejemplo la probabilidad del nodo ser cero o no tomar la oferta (0.865) y la probabilidad de ser uno o tomar la oferta (0.135)

, donde esté último se tomará en cuenta para el orden de la variable categorizada, es decir a la Edad menor igual a 29 años tomara valor "4" puesto que de todos los nodos del árbol, es el que tiene mayor probabilidad de tomar la oferta y así sucesivamente se recodificara cada nodo.

La variable edad categorizada es la siguiente:

Tabla N° 2
Categorización de la variable Edad(X3)

Edad	Categoría	Operaciones	Casos "1"	% de Target
> 56	1	8,818	546	6.2%
<42 - 56]	2	21,403	1,713	8.0%
<29 - 42]	3	29,305	2,725	9.3%
<= 29	4	11,962	1,612	13.5%

Fuente: Elaboración Propia

El resto de variables seguirán el proceso indicado y se adjuntara en los anexos los arboles de decisión CHAID (parámetro de corte para split y merge al 95% de confianza) asimismo se presenta las tablas resumen de todas variables categorizadas mediante la metodología ya explicada:

Tabla N° 3
Categorización de la variable Segmento(X1)

Segmento	Categoría	Operaciones	Casos "1"	% de Target
[2,5]	1	9,705	412	4.2%
[3,6]	2	20,997	1,655	7.9%
[1,4]	3	40,786	4,529	11.1%

Fuente: Elaboración Propia

Tabla N° 4
Categorización de la variable Marca de Tarjeta de Crédito(X4)

TC	Categoría	Operaciones	Casos "1"	% de Target
Visa	1	50,294	3,805	7.6%
Otra	2	21,194	2,791	13.2%

Fuente: Elaboración Propia

Tabla N° 5
Categorización de la variable Flujo de Riesgo(X2)

Flujo Riesgo	Categoría	Operaciones	Casos "1"	% de Target
Bajo	1	12,035	386	3.2%
Alto	2	59,453	6,210	10.4%

Fuente: Elaboración Propia

Tabla N° 6
Categorización de la variable Línea de Crédito(X5)

Línea Crédito	Categoría	Operaciones	Casos "1"	% de Target
>17,200	1	11,630	423	3.6%
<8,200 - 17,200]	2	10,487	606	5.8%
<2,100 - 8,200]	3	19,004	1,490	7.8%
<1,100 - 2,100]	4	16,446	1,771	10.8%
<= 1,100	5	13,921	2,306	16.6%

Fuente: Elaboración Propia

Tabla N° 7
Categorización de la variable Prioridad de Ejecución(X6)

Prioridad	Categoría	Operaciones	Casos "1"	% de Target
[6,7,14,15,16]	1	21,675	958	4.4%
[4,5,11,12,13]	2	20,594	1,563	7.6%
[3,8]	3	11,859	1,164	9.8%
[2,9,10]	4	8,526	953	11.2%
1	5	8,834	1,958	22.2%

Fuente: Elaboración Propia

Tabla N° 8
Categorización de la variable Cod Propensión de Contacto(X7)

Pro - Contacto	Categoría	Operaciones	Casos "1"	% de Target
2	1	35,291	2,367	6.7%
[1,3]	2	11,288	982	8.7%
5	3	9,104	1,144	12.6%
4	4	15,805	2,103	13.3%

Fuente: Elaboración Propia

Tabla N° 9
Categorización de la variable Cod Propensión de Aceptación(X8)

Pro - Aceptaciór	Categoría	Operaciones	Casos "1"	% de Target
no	1	32,777	2,721	8.3%
si	2	38,711	3,875	10.0%

Fuente: Elaboración Propia

Tabla N° 10
Categorización de la variable Números de llamada(X9)

N° Llamada	Categoría	Operaciones	Casos "1"	% de Target
> 12	1	8,748	572	6.5%
<3 - 12]	2	27,873	2,220	8.0%
<3 - 9]	3	11,084	1,074	9.7%
<1 - 3]	4	14,325	1,488	10.4%
<= 1	5	9,458	1,242	13.1%

Fuente: Elaboración Propia

Tabla N° 11
Categorización de la variable Recurrencia(X10)

Recurrencia	Categoría	Operaciones	Casos "1"	% de Target
> 1	1	61,681	4,268	6.9%
<= 1	2	9,807	2,328	23.7%

Fuente: Elaboración Propia

Tabla N° 12
Categorización de la variable Propensión(X11)

Propensión	Categoría	Operaciones	Casos "1"	% de Target
2	1	24,448	1,259	5.1%
3	2	17,521	1,872	10.7%
1	3	29,519	3,465	11.7%

Fuente: Elaboración Propia

Posteriormente se procedió a partición muestral estratificada en base a la target para garantizar que se mantenga la proporción de la target en los datos de entrenamiento y evaluación de los datos, los datos se dividió en un 80% para los datos de entrenamiento y un 20% para los datos de evaluación, es decir ver el desempeño del modelo realizado,

esta partición muestral se realizó mediante la librería h2o, la cual se emplea a los largo de la investigación para realizar los modelos y compararlos (ver parte muestral del código del anexo)

Se observa la proporción de las categorías de la target

Tabla N° 13
Tabla de contingencia de la target para el Train

FLAG_APROBADO	Cantidad	% del Total
0	51972	90.9%
1	5209	9.1%

Fuente: Elaboración Propia

Tabla N° 14
Tabla de contingencia de la target para los datos de evaluación

FLAG_APROBADO	Cantidad	% del Total
0	12920	90.3%
1	1387	9.7%

Fuente: Elaboración Propia

Claramente comparando la Tabla 13 y 14 se mantiene la proporción original de la target, tanto para los datos de entrenamiento (Train) como para la de evaluación (ver modelado de la data mediante random forest del código del anexo)

Ejecución de los Modelos:

Se plantearon 4 modelos:

- Modelo 1: se modeló sin que la target este balanceado y utilizando los parámetros del random forest por defecto.
- Modelo 2: se modeló balanceando en base a la target mediante un tipo de balanceo de *oversampling* (SMOTE) y utilizando parámetros del random forest por defecto.
- Modelo 3: se modeló sin balancear en base a la target y utilizando un *tuning* de parámetros mediante un *grid search* para encontrar los mejores parámetros del random forest.

- Modelo 4: se modeló balanceando en base a la target y utilizando un *tuning* de parámetros mediante un *grid search* para encontrar los mejores parámetros del random forest.

Se procederá a realizar el primer modelo de random forest, se modeló sin que la target este balanceado y utilizando los parámetros del random forest por defecto.

El primer modelo otorga los siguientes indicadores para el conjunto de datos de evaluación.

Tabla N° 15
Indicadores del modelo 1 para los datos de evaluación

Indicador	Valor
AUC	75.70%
especificidad	85.00%
sensibilidad	47.20%

Fuente: Elaboración Propia

El segundo modelo de random forest, se modelará balanceando en base a la target mediante un tipo de balanceo de oversampling (SMOTE) y utilizando en el parámetro de balanceo de clases (*balance_classes*) **TRUE** y los otros parámetros se utilizará por defecto.

El segundo modelo otorga los siguientes indicadores para el conjunto de datos de evaluación.

Tabla N° 16
Indicadores del modelo 2 para los datos de evaluación

Indicador	Valor
AUC	75.30%
especificidad	67.40%
sensibilidad	67.30%

Fuente: Elaboración Propia

El tercer modelo de random forest, se modeló sin balancear en base a la target pero utilizando un *tuning* de parámetros mediante un *grid search* para encontrar los mejores parámetros del random forest.

Para el *grid search* se creó una lista con los parámetros a cambiar:

Tabla N° 17
Tabla de parámetros para el *grid search* del modelo 3

ntrees	30	**max_depth**	1 al 30
	50		
	100		
	150	**nbins**	6
	200		4
	250		3
sample_rate	0.3	**mtries**	2 al 11
	0.7		
	0.4		

Fuente: Elaboración Propia

ntrees: esta opción especifica el número de árboles para construir en el modelo.
max_depth: esto especifica la profundidad máxima a la que se construirá cada árbol.
nbins: la opción especifica el número de nodos que se incluirán en el árbol.
mtries: esta opción especifica el número de variables para seleccionar aleatoriamente en cada árbol.
sample_rate: esta opción se utiliza para especificar la frecuencia de muestreo de las observaciones (sin reemplazo).

Tabla N° 18
Resultados del *grid search* del modelo 3

max_depth	mtries	nbins	ntrees	sample_rate	modelo_id	logloss
9	5	6	30	0.3	modelo_99	0.2638
8	6	6	30	0.3	modelo_128	0.264
9	7	6	30	0.3	modelo_159	0.26401
8	7	6	30	0.3	modelo_158	0.26411
9	4	6	30	0.3	modelo_69	0.2643

Fuente: Elaboración Propia

Como se puede observar en la Tabla N° 18, el modelo ganador es la iteración número 99, la cual obtiene un menor logloss que es un indicador que mide en promedio la cercanía de la probabilidad respecto a la clase de la target, con los siguientes parámetros:

- ✓ **ntrees:** 30
- ✓ **max_depth:** 9
- ✓ **nbins:** 6
- ✓ **mtries:** 5
- ✓ **sample_rate:** 0.3

Se procederá a ejecutar el tercer modelo con los parámetros obtenidos:

El tercer modelo de random forest, se modeló sin que el target esté balanceado y utilizando los parámetros obtenidos en la Tabla N° 18 del modelo ganador del *grid search*.

El tercer modelo otorga los siguientes indicadores para el conjunto de datos de evaluación.

Tabla N° 19
Indicadores del modelo 3 para los datos de evaluación

Indicador	Valor
AUC	75.70%
especificidad	91.30%
sensibilidad	38.60%

Fuente: Elaboración Propia

El cuarto modelo de random forest, se modeló balanceando en base al target, pero utilizando un *tuning* de parámetros mediante un *grid search* para encontrar los mejores parámetros del random forest con la siguiente sentencia:

Para el *grid search* se usó la lista creada ya en el modelo 3 (ver Tabla N° 17)

Al ejecutar el grid search nos otorga las iteraciones de todas las combinaciones posibles de los parámetros ingresados y se ordenó los resultados de acuerdo a un indicador (LOG-LOSS) de menor a mayor:

Tabla N° 20
Resultados del *grid search* del modelo 4

max_depth	mtries	nbins	ntrees	sample_rate	modelo_id	logloss
8	9	6	30	0.3	modelo_277	0.2635
9	5	6	30	0.3	modelo_99	0.2638
9	6	4	30	0.3	modelo_390	0.2639
8	11	6	30	0.3	modelo_309	0.26395
8	6	6	30	0.3	modelo_128	0.264

Fuente: Elaboración Propia

Como se puede observar en la Tabla N° 20, el modelo ganador es la iteración número 277 con los siguientes parámetros:

- ✓ **ntrees:** 30
- ✓ **max_depth:** 8
- ✓ **nbins:** 6
- ✓ **mtries:** 9
- ✓ **sample_rate:** 0.3

Luego se procedió a realizar el cuarto modelo con los parámetros obtenidos:

El cuarto modelo de random forest, se modelará con el target balanceado y utilizando los parámetros obtenidos en el Tabla N° 20 del modelo ganador del *grid search*.

El cuarto modelo otorga los siguientes indicadores para el conjunto de datos de evaluación.

Tabla N° 21
Indicadores del modelo 4 para los datos de evaluación

Indicador	Valor
AUC	75.30%
especificidad	55.30%
sensibilidad	79.10%

Fuente: Elaboración Propia

Comparación de modelos mediante indicadores

1) Comparación del *Out-of-bag* (OOB) de la muestral train y evaluación

Tabla N° 22
Comparación del OOB en la muestra train y evaluación

Modelo 1		
Indicador	Train	Evaluación
Error Class	33.27%	34.82%
AUC	75.20%	75.71%

Modelo 2		
Indicador	Train	Evaluación
Error Class	22.44%	35.88%
AUC	86.34%	75.31%

Modelo 3		
Indicador	Train	Evaluación
Error Class	35.34%	34.82%
AUC	76.08%	75.71%

Modelo 4		
Indicador	Train	Evaluación
Error Class	30.21%	35.88%
AUC	78.97%	75.31%

Fuente: Elaboración Propia

Como se observa en la Tabla N° 22, presenta el error de clasificación en los 4 modelos planteados para el OOB de la muestra de train y evaluación, donde el indicador es muy cercano para los 4 modelos, en el caso del AUC varia aproximadamente entre 1% y 12% el indicador por lo que no presenta una alta variabilidad en los indicadores.

2) Importancia de variables:

Una de las virtudes del modelo de random forest es el de importancia de variables

Tabla N° 23
Importancia de variables del modelo 1

Variable	Importancia Relativa	Porcentaje
X9	5,510.50	16.7%
X10	5,418.98	16.4%
X6	5,380.85	16.3%
X5	4,057.05	12.3%
X3	3,180.73	9.6%
X7	3,046.14	9.2%
X11	1,831.08	5.5%
X1	1,786.63	5.4%
X4	1,382.45	4.2%
X8	785.23	2.4%
X2	689.32	2.1%

Fuente: Elaboración Propia

Como se observa en la Tabla N° 23, las 4 principales variables son:

- ✓ X9: Número de veces que se ha llamado al lead.
- ✓ X10: Indica que tan recurrente es el lead en la campaña asignada.
- ✓ X6: Indica a que grupo de ejecución pertenece el lead.
- ✓ X5: Línea de crédito de la tarjeta de crédito.

Tabla N° 24
Importancia de variables del modelo 2

Variable	Importancia Relativa	Porcentaje
X6	54,591.90	18.0%
X10	50,101.67	16.5%
X5	39,515.83	13.0%
X9	36,341.18	12.0%
X7	30,197.75	10.0%
X3	22,868.34	7.5%
X1	18,164.51	6.0%
X11	17,070.15	5.6%
X2	16,622.32	5.5%
X4	10,990.84	3.6%
X8	6,549.98	2.2%

Fuente: Elaboración Propia

Como se observa en la Tabla N° 24, las 4 principales variables son:

- ✓ X6: Indica a que grupo de ejecución pertenece el lead.
- ✓ X10: Indica que tan recurrente es el lead en la campaña asignada.
- ✓ X5: Línea de crédito de la tarjeta de crédito.
- ✓ X9: Número de veces que se ha llamado al lead.

Tabla N° 25
Importancia de variables del modelo 3

Variable	Importancia Relativa	Porcentaje
X10	2,815.74	48.3%
X6	1,992.75	19.7%
X5	1,326.74	9.6%
X9	1,242.57	8.3%
X7	796.93	4.9%
X3	690.42	4.4%
X1	475.45	1.8%
X11	399.56	1.7%
X4	281.53	0.5%
X8	145.11	0.4%
X2	104.87	0.4%

Fuente: Elaboración Propia

Como se observa en la Tabla N° 25, las 4 principales variables son:

- ✓ X10: Indica que tan recurrente es el lead en la campaña asignada.
- ✓ X6: Indica a que grupo de ejecución pertenece el lead.
- ✓ X5: Línea de crédito de la tarjeta de crédito.
- ✓ X9: Número de veces que se ha llamado al lead.

Tabla N° 26
Importancia de variables del modelo 4

Variable	Importancia Relativa	Porcentaje
X10	49,129.45	48.7%
X6	14,910.06	20.2%
X7	7,681.10	10.6%
X5	7,443.39	7.8%
X9	5,354.12	3.7%
X11	3,996.18	2.8%
X1	3,825.79	2.6%
X3	2,273.12	2.6%
X2	1,713.87	0.5%
X4	675.62	0.3%
X8	428.28	0.2%

Fuente: Elaboración Propia

Como se observa en la Tabla N° 26, las 4 principales variables son:

- ✓ X10: Indica que tan recurrente es el lead en la campaña asignada.
- ✓ X6: Indica a que grupo de ejecución pertenece el lead.
- ✓ X7: Indica que propenso es el lead para ser contactado.
- ✓ X5: Línea de crédito de la tarjeta de crédito.

3) Matriz de confusión de las clases estimadas y la target

Tabla N° 27
Matriz de confusión del modelo 1

Predicción	FLAG_APROBADO	Casos
0	0	10,981
0	1	733
1	0	1,939
1	1	654

Fuente: Elaboración Propia

Tabla N° 28
Matriz de confusión del modelo 2

Predicción	FLAG_APROBADO	Casos
0	0	8,705
0	1	454
1	0	4,215
1	1	933

Fuente: Elaboración Propia

Tabla N° 29
Matriz de confusión del modelo 3

Predicción	FLAG_APROBADO	Casos
0	0	11,791
0	1	852
1	0	1,129
1	1	535

Fuente: Elaboración Propia

Tabla N° 30
Matriz de confusión del modelo 4

Predicción	FLAG_APROBADO	Casos
0	0	7,146
0	1	290
1	0	5,774
1	1	1,097

Fuente: Elaboración Propia

De los cuales obtendremos los indicadores propuestos para esta investigación:

4) Indicadores de comparación

Tabla N° 31
Comparación de Indicadores de los modelos propuestos

Indicador	Modelo 1	Modelo 2	Modelo 3	Modelo 4
AUC	75.70%	75.30%	75.70%	75.30%
especificidad	85.00%	67.40%	91.30%	55.30%
sensibilidad	47.20%	67.30%	38.60%	79.10%

Fuente: Elaboración Propia

4.2. Análisis de resultados o discusión de los resultados

Como se observa en la Tabla N° 30, los 4 modelos plateados mediante el modelo de random forest, comparando en primer lugar balanceo y luego *tuning* de parámetros del modelo se detalla lo siguiente:

4.2.1. Indicador AUC

Los 4 modelos propuestos obtuvieron un indicador AUC alrededor de 0.75, donde según (Pérez, 2015), el cual plantea los siguientes intervalos para medir la calidad de predicción del modelo:

- ✓ siguientes los intervalos para los valores de la curva ROC:
  - [0.5 - 0.6>: Test malo
  - [0.6 - 0.75>: Test regular
  - [0.75 - 0.9>: Test bueno
  - [0.9 - 0.97>: Test muy bueno
  - [0.97 - 1>: Test excelente

Los 4 modelos se encuentran en una calidad Buena.

4.2.2. Indicador de especificidad

En base a este indicador existen diferencias en los resultados, este indicador evalúa la predicción de la clase sobre la categoría que no es de interés, en este caso el cero(0), para los modelos 1 y 3 se obtuvieron un mayor indicador, esto es esperable puesto los modelos que se plantearon solo aprendieron de la categoría que presenta mayor clase, es decir sobre la categoría de los ceros, siendo el modelo 3, encontrando los mejores parámetros del modelo de random forest mediante el *grid search*, donde se obtuvo un mejor acierto sobre este indicador.

4.2.3. Indicador de sensibilidad

En base a este indicador existen diferencias en los resultados, este indicador evalúa la predicción de la clase sobre la categoría que es de interés, en este caso el uno (1), para los modelos 2 y 4 se obtuvieron un mayor indicador, esto es esperable puesto se balanceo la target y los modelos que se plantearon aprendieron de ambas categorías, siendo el modelo 4, encontrando los parámetros mediante el *grid search* del modelo de random forest donde se obtuvo un mejor acierto sobre este indicador.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

En la investigación se presentaron 4 modelos, el primer modelo usando la data por defecto, es decir sin balancear la target, el segundo modelo balanceando en base a la target, un tercer modelo sin balancear en base a la target y encontrando los mejores parámetros mediante un *grid search* y un cuarto modelo balanceando en base a la target y encontrando los mejores parámetros mediante un *grid search*, el cual mediante indicadores se procedió a comparar y hacer el análisis respectivo.

1) En el caso de AUC todos los modelos presentaron alrededor de un 0.75, por lo que las predicciones son buenas y los mejores fueron el modelo 1 y modelo 3.
2) En el caso de especificidad mide la predicción de la clase que no es de interés, es decir la del cero, los mayores indicadores lo obtuvieron el modelo 1 (85%) y modelo 3 (91.3%) dándose el mejor cuando se mejoró los parámetros mediante el *grid search* es decir el modelo 3.
3) En el caso de sensibilidad mide la predicción de la clase que es de interés, es decir la del uno, los mayores indicadores lo obtuvieron el modelo 2 (67.3%) y modelo 4 (79.1%), los indicadores resultaron mayores cuando se mejoró los parámetros del random forest mediante el *grid search*, es decir el modelo 4.
4) Para el presente estudio interesa ver más sobre la calidad de predicción sobre los unos puesto para el negocio de Banca, le interesa ver una mejor predicción sobre los que aceptaron la campaña de tarjeta de crédito para así tener un mejor performance y decisión sobre los leads, por lo que el modelo 4 cumple con tener una mayor sensibilidad.

5) Las recodificaciones de las variables mediante una técnica de árboles de decisión nos ayudó a tener una mejor estabilidad de cada variable y convertirlas a una variable categórica ordinal (acorde al porcentaje de uno de la target), balanceando la target y encontrando los mejores parámetros para el modelo de random forest, se pudo obtener un mejor performance sobre la categoría de interés, la cual nos ayudó a obtener un mejor indicador de sensibilidad y así poder tomar una mejor estrategia para el caso de negocio de Banca.

Recomendaciones

1) Entrenar los modelos respecto a un indicador que mida las probabilidades como lo es el AUC en el *grid search*, nos ayudaría a encontrar un mejor performance para este indicador.
2) Un alto valor de la especificidad ayudará a conocer a los leads que no toman la campaña ofrecida por la entidad, por lo que se recomienda hacer un estudio más profundo para estos leads.
3) Un alto valor de la sensibilidad ayudará hacer la gestión más eficiente y priorizar a los leads más propensos.
4) Comparar otros algoritmos, ayudará a encontrar mejores indicadores.

REFERENCIAS BIBLIOGRÁFICAS

Alfaro Cortez, Esteban; Gamez Martinez, Matias; Garcia Rubio, Noelia;. (2002). *Una Revisión de los Métodos de Agregación de Clasificadores.* Plaza de la Universidad, s/n. 02071 Albacete.: Universidad de Castilla-La Mancha.

Arnejo Calviño, H. (2017). *Métodos para la mejora de predicciones en clases desbalanceadas en el estudio de bajas de clientes (CHURN).* España.

Beltrán Pascual, M. (2015). *Diseño e implementación de un nuevo clasificador de préstamos bancarios a través de la minería de datos.* Madrid.

Breiman, L. (2001). *Random Forest.* California: Statistics Department.

Brownlee, J. (Agosto de 2015). *Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset*. Obtenido de http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Buja, A., Stuetzle, W., & Shen, Y. (2005). *Loss Functions for Binary Class Probability Estimation and Classification: Structure and Applications.* Pennsylvania.

Cánovas García, F., Alonso Sarría, F., & Gomariz Castillo, F. (2016). *MODIFICACIÓN DEL ALGORITMO RANDOM FOREST PARA SU EMPLEO EN.* Málaga.

Cárdenas-Montes, M. (2010). *Técnica de Bagging.* Sevilla: Dpto. Ciencias de la Computación e Inteligencia Artificial.

Cardona Hernández, P. A. (2004). *"Aplicación de árboles de decisión en modelos de riesgo crediticio".* Colombia: Revista Colombiana de Estadística.

Cortez, E. A. (2006). *Combinación de clasificadores mediante el método boosting, una aplicación a la predicción del fracaso empresarial en España.* Madrid: Castilla La Mancha.

De Juan de Llano, I. (2017). *Análisis y optimización de algoritmos de clasificación supervisada sobre operaciones impagadas en tarjetas de créditos.* Madrid.

Fawcett, T. (25 de Agosto de 2016). *Datos Desbalanceados*. Recuperado el 25 de Noviembre de 2017, de https://svds.com/learning-imbalanced-classes/

Fernández Vásquez, R. (2016). *Regresión bayesiana con enlaces asimétricos para la clasificación de clientes con propensión a caer en mora en una entidad bancaria.* Lima: Escuela de Posgrado UNAM.

Garzon, V. (2008). *Aplicación de técnicas de Induscción de árboles de Decisión a problemas de clasificación mediante el uso de WEKA.* Bogota, Colombia: Fundación Universitaria Konrad Lorenz.

Gonzalez, L. (2008). *Máquinas I- SVCR con salidad probabilisticas.*

Hassinger Rodriguez, M. (2015). *GRADO DE MÁSTER EN INGENIERÍA Y TECNOLOGÍA DE SISTEMAS SOFTWARE.* Valencia.

Heras Martínez, A., Rodríguez-Piñero Piedad, T., & Hernández-March, J. (2003). *"Un Análisis Comparativo de una SMV y un Modelo Logit en un problema de clasificación de Asegurados".* Madrid, España: Universidad de Complutense.

Hermosilla Martelli, G. (2015). *Mejoramiento de un modelo de targeting de clientes de telefonía móvil usando análisis de redes sociales y minería de datos.* Santiago de Chile.

Hitzia, G. M. (2017). *Predicción de fuga de clientes en una corredora de seguros utilizando regresión logística y el algoritmo de random forest.* Lima.

Landry, M. (2018). *Machine Learning with R and H2O.* United States of America: H2O.

López Pineda, A. (2008). *Algoritmos de balanceo de clases en problemas de clasificación binaria de conjuntos altamente desproporcionados.* Mexico.

Maude, J. (2010). *Combinación de Clasificadores:Construcción de Características e Incremento de la Diversidad.* Universidad de Burgos.

Medina Merino, Rosa ; Ñique Chacón, Carmen;. (2017). *Bosques aleatorios como extensión de los árboles de clasificación con los programas R y Python.* Lima.

Moreno García María, Q. L. (2001). *"Aplicación de técnicas de minería de datos en la Construcción y Validación de modelos predictivos y Asociativos a partir de especificaciones de requisitos de software".* Real: Universidad de Salamanca.

Muñoz Martinez, G. (2006). *Clasificación mediante conjuntos.* MADRID.

Murillo, S. N. (2010). *Crédito al Consumo, La Estadística aplicada a un problema de Riesgo Crediticio.* Mexico.

Pardo Aguilar, C. (2015). *Minería de datos y combinación de regresores.* España.

Pérez J., M. (2015). *Bayesian Asymmetric Logit Model for Detecting Risk Factors in Motors Ratemaking.* Principado de Asturias: Facultad de Ciencias, Universidad de Oviedo.

Puente-Maury, L., López-Chau, & Cruz-Santos, W. (2014). *Método rápido de preprocesamiento para clasificación en conjuntos de datos no balanceados.* Mexico.

Randa, M., Lopez, G., & Garach, L. (2015). *Bayes classifiers for imbalanced traffic accidents data sets.*

Rodríguez, O. (2010). *Un Aprendizaje Supervisado: Árboles de Decisión.*

Samuel D. Pacheco Leal, L. G. (2005). *El clasificador de Naive Bayes en la extracción de conocimiento de bases de datos.*

Sánchez Tarragó, D. (2014). *Algoritmos para la clasificación multinstancia.* España.

Sindia, T. T. (2016). *Identification of Main Factors and Variables Describing the Quantity and Distribution of Fatal Vehicular Accidents in Metropolitan City of Lima Using data Mining Techniques: Random Forest, Boosting, Decision Trees.* Lima.

Vargas, M. P. (s.f.). *Clasificador Bayesiano usando Distribución Normal Multivariado para predecir el riesgo acadmeico de pregrado en la Universidad de Colombia.*

Vigo, G. (2010). *Método de calsificación para evaluar el riesgo crediticio: Una comparación.* Lima, Perú: Universidad Nacional Mayor de San Marcos.

Xie, Y., Li, X., Ngai, E., & Ying, W. (2008). *Customer churn prediction using improved balanced random forests.* Hong Kong: Department of Management and Marketing,.

Anexos

1) Código en R sobre el modelado
2) Recodificación de las variables mediante Arboles de decisión:

```
rm(list=ls()) ## eliminar archivos guardados en memoria

######################################################################
########### -- TESIS DE MAESTRIA -- ################################
######################################################################
####### Autor: Lic. José Cárdenas  ###################################
######################################################################

### -- 0) Direccionar a la base de datos

### -- 1) Librerías a usar ####

library(sqldf) ## librería para usar comandos sql
library(h2o) ## librería para trabajar la tesis
library(tidyverse) ## librería para tratamiento de data
library(ggplot2) ## librería para gráficos
library(mlr)

### -- 2) Datos a Utilizar ####

datos_R   <- read.csv(file = "data.csv", header = TRUE)

# colocar los tipos de datos a las categóricas

datos_R$SEGMENTO <- as.factor (datos_R$SEGMENTO)
datos_R$FLUJO_RIESGO <- as.factor (datos_R$FLUJO_RIESGO)
datos_R$MARCA_TC <- as.factor (datos_R$MARCA_TC)
datos_R$PRIORIDAD_EJECUCION <- as.factor (datos_R$PRIORIDAD_EJECUCION)
datos_R$COD_PROP_CONTACTO <- as.factor (datos_R$COD_PROP_CONTACTO)
datos_R$COD_PROP_ACEPTA <- as.factor (datos_R$COD_PROP_ACEPTA)
datos_R$FLAG_APROBADO <- as.factor (datos_R$FLAG_APROBADO)
#levels(datos_R$FLAG_APROBADO) <- c("Desaprobado","Aprobado")

# estadísticas de las variables principales
estadísticas <- summarizeColumns(datos_R)
#write.csv(estadisticas,"estadisticas.csv",row.names = F)

## se discretiza las variables mediante arboles de decision Chaid

datos_R <- sqldf("select case when SEGMENTO in (2,5) then 1
                    when SEGMENTO in (3,6) then 2 else 3 end X1,
                    case when FLUJO_RIESGO in (1) then 1 else 2 end X2,
                    case when EDAD <= 29 then 4
                    when EDAD<= 42 then 3
                    when EDAD <= 56 then 2 else 1 end X3,
                    MARCA_TC AS X4,
                    case when LINEA_CREDITO <= 1100 then 5
                    when LINEA_CREDITO <= 2100 then 4
                    when LINEA_CREDITO <= 8200 then 3
                    when LINEA_CREDITO <= 17200 then 2 else 1 end X5,
                    case when PRIORIDAD_EJECUCION in (1) then 5
                    when PRIORIDAD_EJECUCION in (6, 7, 14, 15, 16) then
1
                    when PRIORIDAD_EJECUCION in (4,5,11,12,13) then 2
                    when PRIORIDAD_EJECUCION in (3,8) then 3
                    when PRIORIDAD_EJECUCION in (2,9,10) then 4 end X6,
                    case when COD_PROP_CONTACTO in (1,3) then 2
                    when COD_PROP_CONTACTO in (2) then 1
                    when COD_PROP_CONTACTO in (5) then 3
                    else 4 end X7,
```

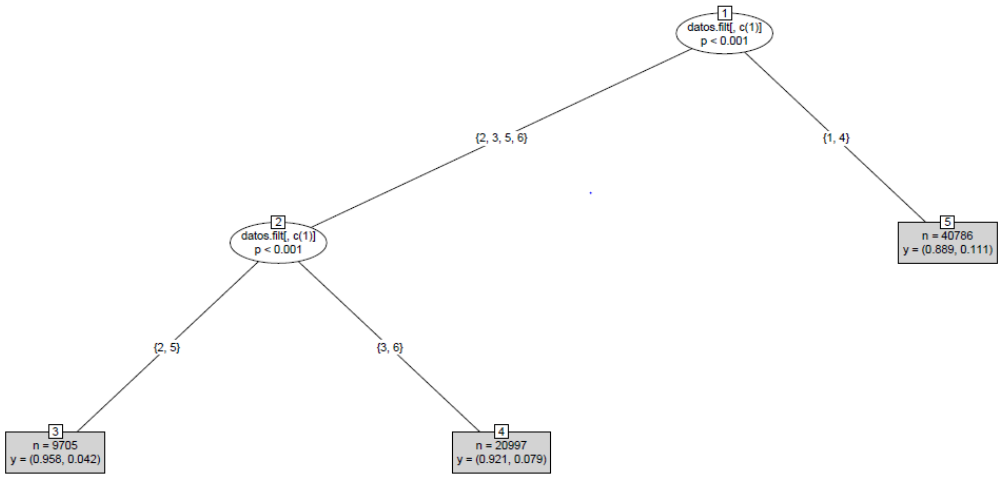
```
                    case when COD_PROP_ACEPTA in (2) then 2 else 1 end
X8,
                    case when NRO_LLAMADA <= 1 then 5
                    when NRO_LLAMADA <= 3 then 4
                    when NRO_LLAMADA <= 9 then 2
                    when NRO_LLAMADA <= 12 then 1
                    when NRO_LLAMADA > 12 then 3 end X9,
                    case when RECURRENCIA <= 1 then 2 else 1 end X10,
                    case when PROPENSION <= 1 then 3
                    when PROPENSION <= 2 then 1 else 2 end X11,
                    FLAG_APROBADO
                    from datos_R")


write.csv(datos_R,"data_recodificadaf.csv",row.names = F)


input_name <- 'data_recod'

## Categorizacion de las variables ##

datos_R[,1:ncol(datos_R)]                                              <-
lapply(datos_R[,1:ncol(datos_R)],as.factor)

#save(datos_R, file = paste(input_name,".RData", sep = ""))
#load(file = paste(input_name,".RData", sep = ""))

### -- 3) Iniciar  h2o conexion ####

h2o.init()
h2o.init(nthreads = -1, max_mem_size = "10G")
## nthreads: -1 indica que se empleen todos los cores disponibles.
## max_mem_size: Máxima memoria disponible para el cluster.

# Se eliminan los datos del cluster por si ya había sido iniciado.
h2o.removeAll()

datos_h2o <- as.h2o(x = datos_R, destination_frame = "datos_h2o")

### -- 4) Analisis descriptivo de la Datos ####

# Dimensiones del set de datos
h2o.dim(datos_h2o)

# Nombre de las columnas
h2o.colnames(datos_h2o)

# obtener un análisis rápido que muestre el tipo de datos
h2o.describe(datos_h2o)

# Se crea una tabla con el número de observaciones de cada tipo.
tabla_muestra <- as.data.frame(h2o.table(datos_h2o$FLAG_APROBADO))
tabla_muestra

ggplot(data = tabla_muestra,
       aes(x = FLAG_APROBADO, y = Count, fill = FLAG_APROBADO)) +
  geom_col() +
  scale_fill_manual(values = c("gray50", "orangered2")) +
  theme_bw() +
  labs(x = "Aprobacion", y = "Número de observaciones",
       title = "Distribución de la variable Aprobacion") +
```

```
  theme(legend.position = "none")


## -- Partición Muestral -- ##

# Separación de las observaciones en conjunto de entrenamiento y
evaluación.
# En los ejemplos de GBM y deep learning se repetirá la separación, pero
en
# tres conjuntos en lugar de dos.
particiones <- h2o.splitFrame(data = datos_h2o, ratios = c(0.8), seed =
123)
datos_train_h2o      <-  h2o.assign(data  =  particiones[[1]],  key  =
"datos_train_H2O")
datos_test_h2o       <-  h2o.assign(data  =  particiones[[2]],  key  =
"datos_test_H2O")

### -- 3) Modelado de la Datos ####

## -- Modelado de los datos mediante Random forest -- ###

tabla <- h2o.table(datos_train_h2o$FLAG_APROBADO)
tabla

tablat <- h2o.table(datos_test_h2o$FLAG_APROBADO)
tablat

cantidad_0_train <- tabla[1,2] ## Cantidad de Ceros en la Train
cantidad_1_train <- tabla[2,2] ## Cantidad de Unos en la Train

n <- ncol(datos_train_h2o) # Numero de columnas en la Train

## primer Modelo de Random forest sin balancear
m1   <-   h2o.randomForest(1:(n-1),   n,   datos_train_h2o,   model_id
="RF_defaults", seed=12345)

h2o.performance(model = m1, newdata = datos_test_h2o)
predicciones_m1 <- h2o.predict(object = m1, newdata = datos_test_h2o)


## Segundo Modelo de Random forest con balanceo ##
m2   =   h2o.randomForest(1:(n-1),   n,   datos_train_h2o,   model_id
="RF_balanced", seed=12345,
                      balance_classes = TRUE)

predicciones_m2 <- h2o.predict(object = m2, newdata = datos_test_h2o)
h2o.performance(model = m2, newdata = datos_test_h2o)


## Tercer Modelo de random forest tuning ##

## Tuning de parametros del random forest

grid_space <- list()
grid_space$ntrees <- c(30,50,100,150,200,250)
grid_space$max_depth <- seq(1,30)
grid_space$nbins <- c(6, 4, 3)
grid_space$mtries <- c(2,3,4,5,6,7,8,9,10,11)
grid_space$sample_rate <- c(0.3, 0.7, 0.4)

# Entrenamiento con los parametros para data sin balancear
```

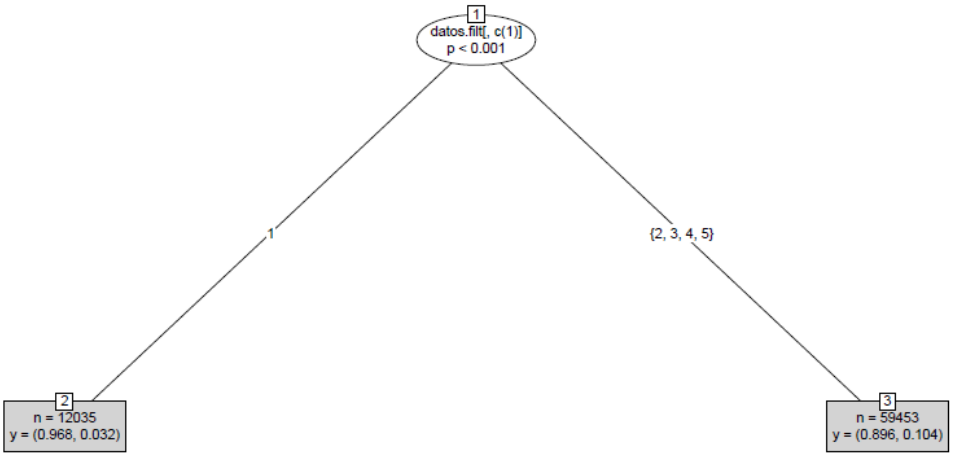
```
banca_drf_grid <- h2o.grid("randomForest",
                           grid_id="drf_grid_banca_test",
                           x=1:(n-1),
                           y=n,
                           training_frame=datos_train_h2o,
                           balance_classes = FALSE,
                           hyper_params=grid_space)
## ver parametros
banca_drf_grid

## Entrenamiento con los parametros para data balanceada
banca_drf_grid <- h2o.grid("randomForest",
                           grid_id="drf_grid_banca_test",
                           x=1:(n-1),
                           y=n,
                           training_frame=datos_train_h2o,
                           balance_classes = TRUE,
                           hyper_params=grid_space)

## ver parametros
banca_drf_grid


m3   <-   h2o.randomForest(1:(n-1),   n,   datos_train_h2o,   model_id
="RF_defaults", seed=12345,
                       balance_classes = FALSE,
                       max_depth=9,mtries=5,nbins=6,ntrees
=30,sample_rate=0.3)

predicciones_m3 <- h2o.predict(object = m3, newdata = datos_test_h2o)
h2o.performance(model = m3, newdata = datos_test_h2o)


## cuarto Modelo con tuneo con balanceo ##

m4   <-   h2o.randomForest(1:(n-1),   n,   datos_train_h2o,   model_id
="RF_balanced", seed=12345,
                       balance_classes = TRUE,
                       max_depth=8,mtries=9,nbins=6,ntrees
=30,sample_rate=0.3)

predicciones_m4 <- h2o.predict(object = m4, newdata = datos_test_h2o)
h2o.performance(model = m4, newdata = datos_test_h2o)

### -- 4) Variables Importantes por Modelo ####

## variables más importantes

h2o.varimp(m1)
h2o.varimp(m2)
h2o.varimp(m3)
h2o.varimp(m4)

# pronostico

predicciones_m1
predicciones_m2
predicciones_m3
predicciones_m4

# matriz de confusión
```

```
tabla1<-h2o.table(predicciones_m1[,1], datos_test_h2o[,12])
tabla2<-h2o.table(predicciones_m2[,1], datos_test_h2o[,12])
tabla3<-h2o.table(predicciones_m3[,1], datos_test_h2o[,12])
tabla4<-h2o.table(predicciones_m4[,1], datos_test_h2o[,12])

### -- 4) Comparación de Indicadores ####

# auc

auc1<-h2o.auc(h2o.performance(m1, newdata = datos_test_h2o))
auc2<-h2o.auc(h2o.performance(m2, newdata = datos_test_h2o))
auc3<-h2o.auc(h2o.performance(m3, newdata = datos_test_h2o))
auc4<-h2o.auc(h2o.performance(m4, newdata = datos_test_h2o))

#sensibilidad
sensibilidad1<-tabla1[4,3]/(tabla1[4,3]+tabla1[2,3])
sensibilidad2<-tabla2[4,3]/(tabla2[4,3]+tabla2[2,3])
sensibilidad3<-tabla3[4,3]/(tabla3[4,3]+tabla3[2,3])
sensibilidad4<-tabla4[4,3]/(tabla4[4,3]+tabla4[2,3])

#especificidad
especificidad1 <-tabla1[1,3]/(tabla1[1,3]+tabla1[3,3])
especificidad2 <-tabla2[1,3]/(tabla2[1,3]+tabla2[3,3])
especificidad3 <-tabla3[1,3]/(tabla3[1,3]+tabla3[3,3])
especificidad4 <-tabla4[1,3]/(tabla4[1,3]+tabla4[3,3])

## Indicadores
auc=rbind(auc1,auc2,auc3,auc4)
especificidad=rbind(especificidad1,especificidad2,especificidad3,espec
ificidad4)
SENSIBILIDAD=rbind(sensibilidad1,sensibilidad2,sensibilidad3,sensibili
dad4)

resultado=data.frame(auc,especificidad,SENSIBILIDAD)
rownames(resultado)=c("RF_Sin_Balanceo","RF_Con_Balanceo","RF_Sin_Bala
nceo_Tuneado","RF_Con_Balanceo_Tuneado")
colnames(resultado)=c("AUC","Especificidad","Sensibilidad")
resultado=round(resultado,3)
resultado

# Se apaga el cluster H2O
#h2o.shutdown(prompt = FALSE)
```

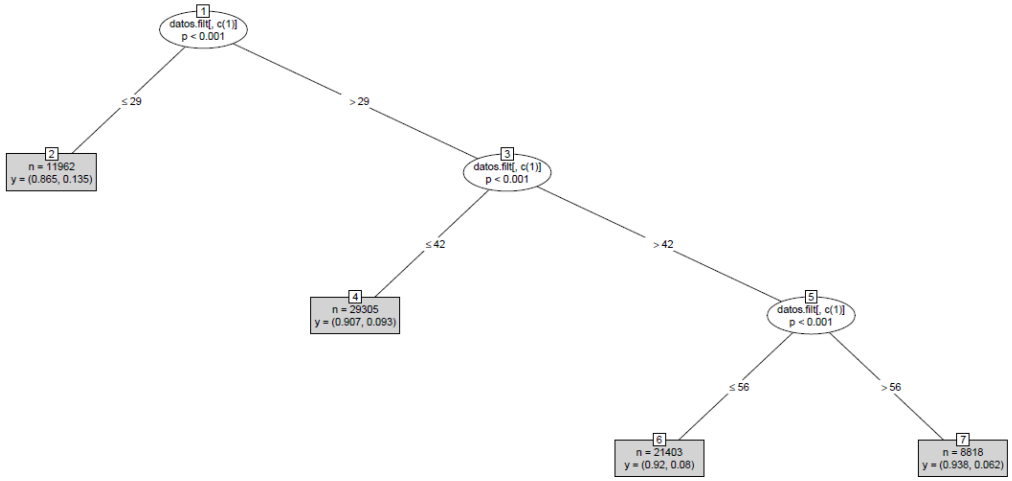
(1) SEGMENTO (y / (T)) (GINI: 14.52 % | KS: 12.79 %) (71488 casos)



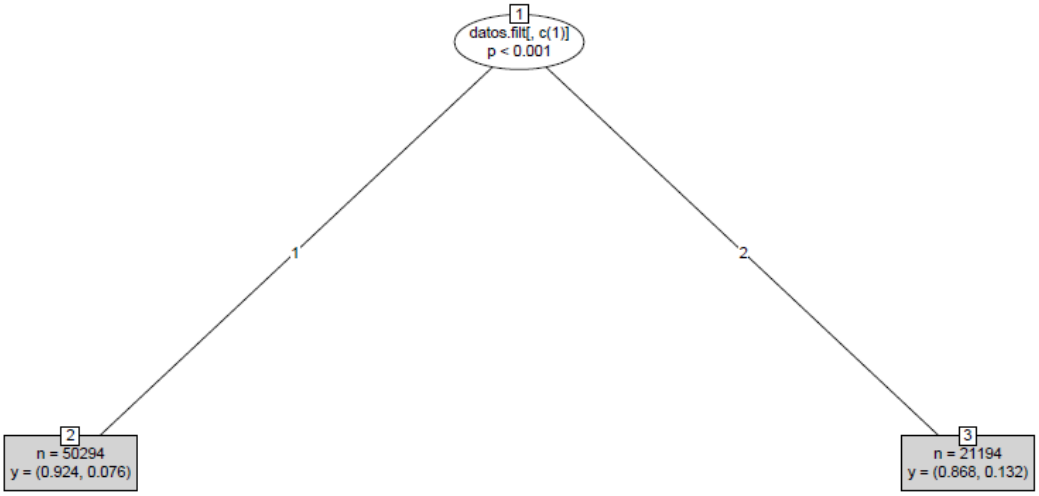
(1) FLUJO_RIESGO (y / (T)) (GINI: 12.1 % | KS: 12.1 %) (71488 casos)

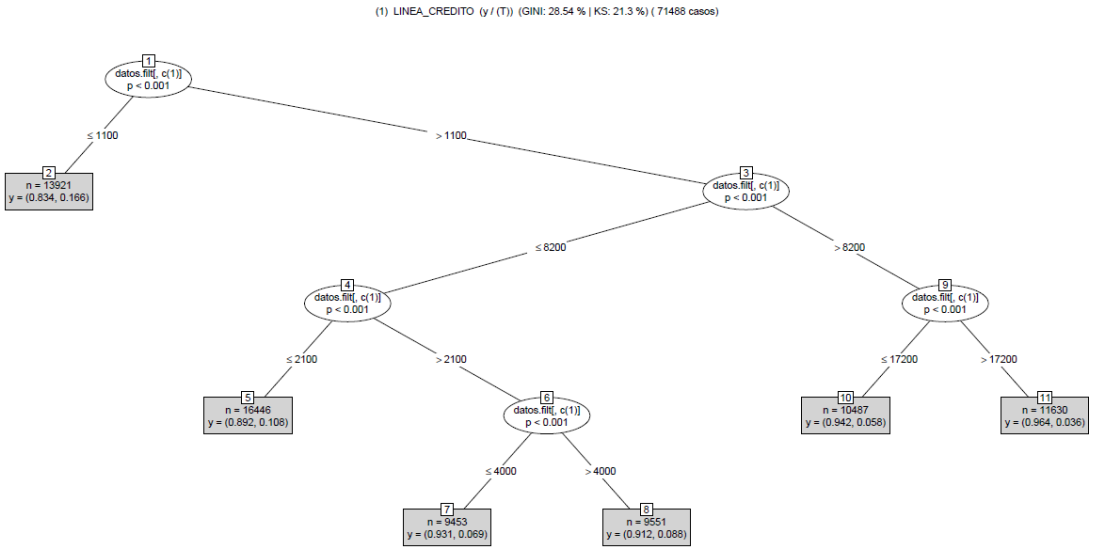


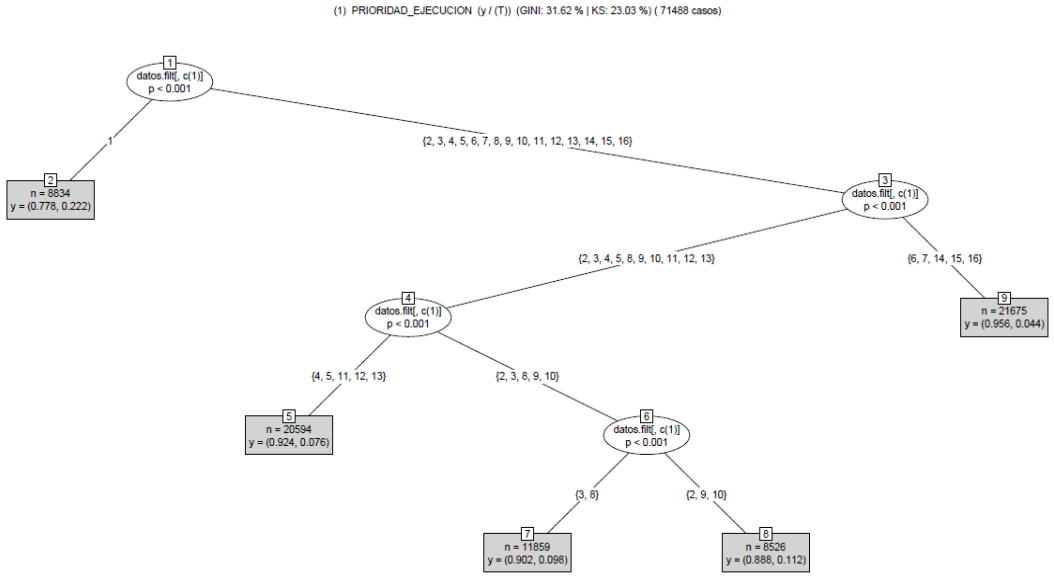
(1) EDAD (y / (T)) (GINI: 13.06 % | KS: 8.84 %) (71488 casos)



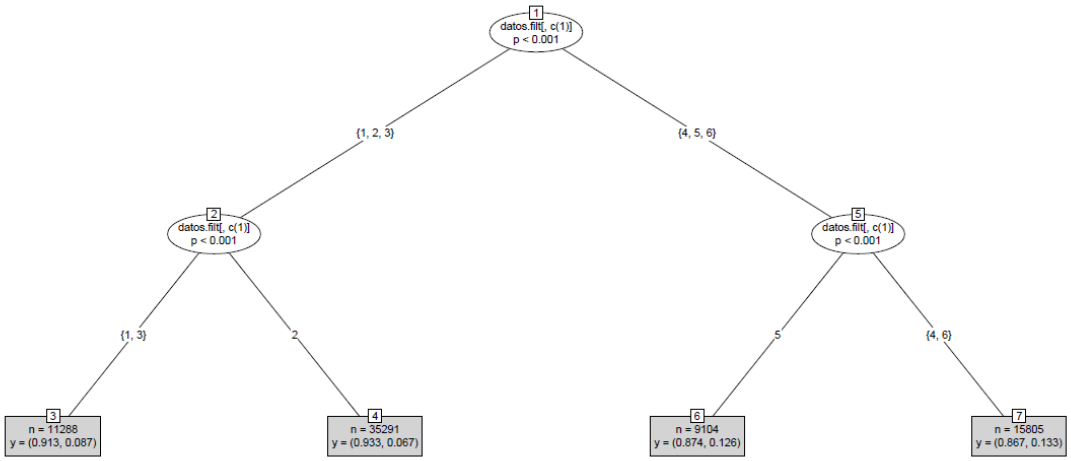
(1) MARCA_TC (y / (T)) (GINI: 13.95 % | KS: 13.95 %) (71488 casos)



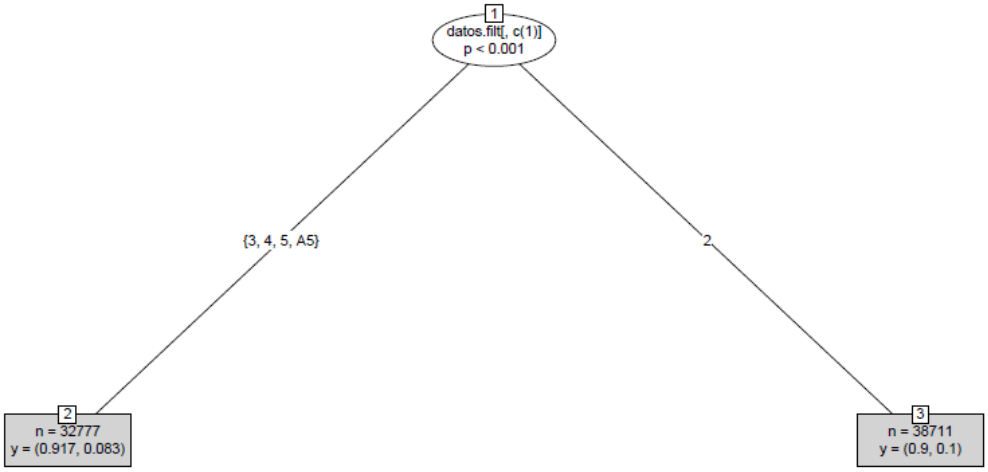

(1) LINEA_CREDITO (y / (T)) (GINI: 28.54 % | KS: 21.3 %) (71488 casos)
1
datos.fit[, c(1)]
p < 0.001
≤ 1100
> 1100
2
n = 13921
y = (0.834, 0.166)
3
datos.fit[, c(1)]
p < 0.001
≤ 8200
> 8200
4
datos.fit[, c(1)]
p < 0.001
9
datos.fit[, c(1)]
p < 0.001
≤ 2100
> 2100
≤ 17200
> 17200
5
n = 16446
y = (0.892, 0.108)
6
datos.fit[, c(1)]
p < 0.001
10
n = 10487
y = (0.942, 0.058)
11
n = 11630
y = (0.964, 0.036)
≤ 4000
> 4000
7
n = 9453
y = (0.931, 0.069)
8
n = 9551
y = (0.912, 0.088)

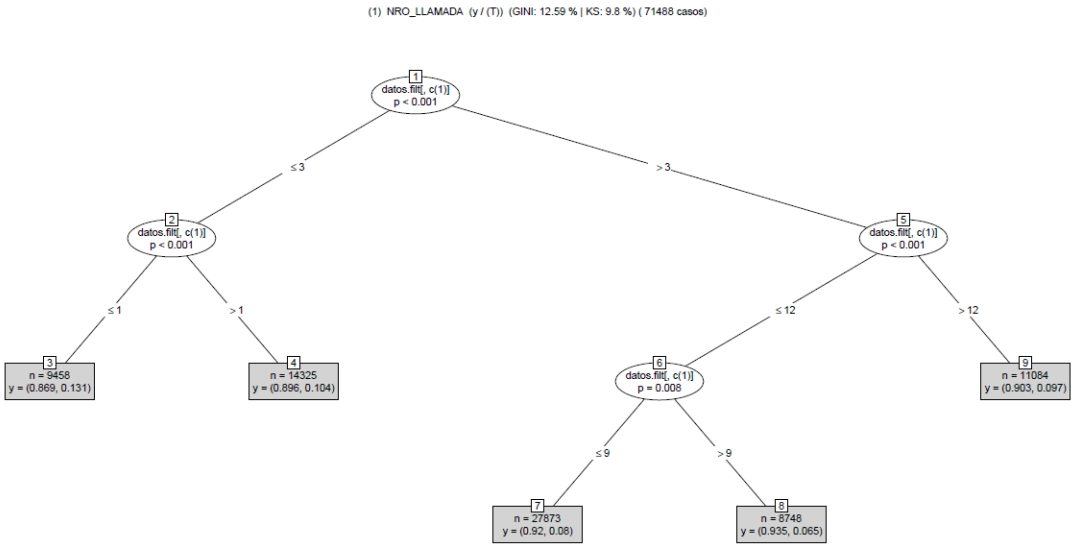

(1) PRIORIDAD_EJECUCION (y / (T)) (GINI: 31.62 % | KS: 23.03 %) (71488 casos)
1
datos.fit[, c(1)]
p < 0.001
1
{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
2
n = 8834
y = (0.778, 0.222)
3
datos.fit[, c(1)]
p < 0.001
{2, 3, 4, 5, 8, 9, 10, 11, 12, 13}
{6, 7, 14, 15, 16}
4
datos.fit[, c(1)]
p < 0.001
9
n = 21675
y = (0.956, 0.044)
{4, 5, 11, 12, 13}
{2, 3, 8, 9, 10}
5
n = 20594
y = (0.924, 0.076)
6
datos.fit[, c(1)]
p < 0.001
{3, 8}
{2, 9, 10}
7
n = 11859
y = (0.902, 0.098)
8
n = 8526
y = (0.888, 0.112)

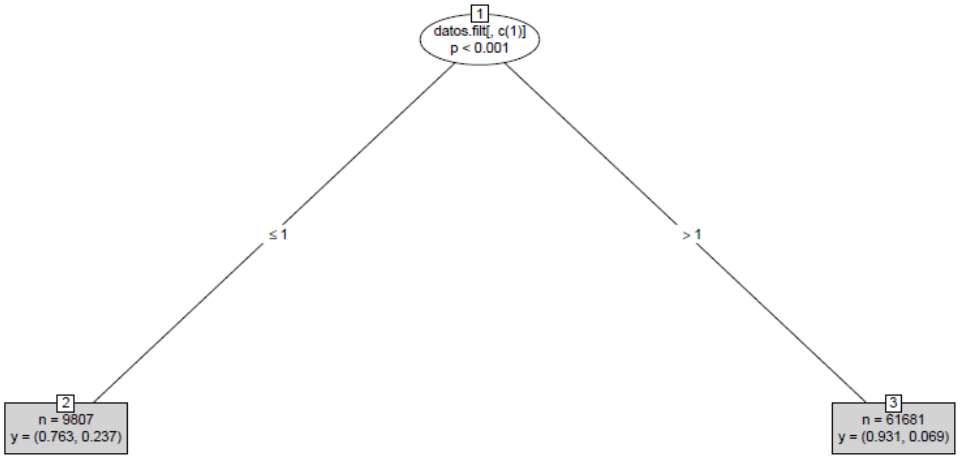
(1) COD_PROP_CONTACTO (y / (T)) (GINI: 17.95 % | KS: 15.85 %) (71488 casos)

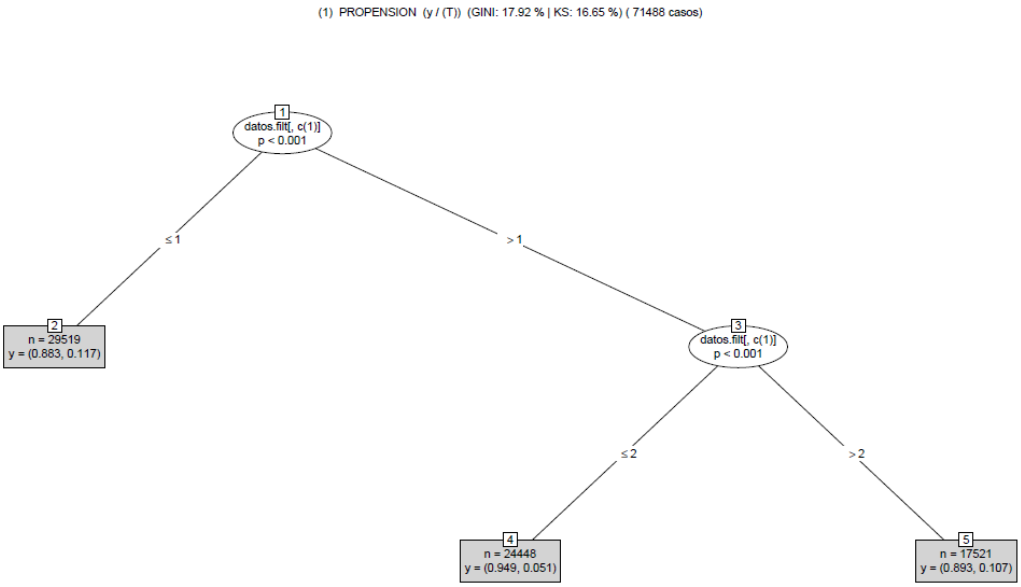


(1) COD_PROP_ACEPTA (y / (T)) (GINI: 5.06 % | KS: 5.06 %) (71488 casos)




(1) NRO_LLAMADA (y / (T)) (GINI: 12.59 % | KS: 9.8 %) (71488 casos)
1
datos.filt[, c(1)]
p < 0.001
≤ 3
> 3
2
datos.filt[, c(1)]
p < 0.001
5
datos.filt[, c(1)]
p < 0.001
≤ 1
> 1
≤ 12
> 12
3
n = 9458
y = (0.869, 0.131)
4
n = 14325
y = (0.896, 0.104)
6
datos.filt[, c(1)]
p = 0.008
9
n = 11084
y = (0.903, 0.097)
≤ 9
> 9
7
n = 27873
y = (0.92, 0.08)
8
n = 8748
y = (0.935, 0.065)


(1) RECURRENCIA (y / (T)) (GINI: 23.77 % | KS: 23.77 %) (71488 casos)
1
datos.filt[, c(1)]
p < 0.001
≤ 1
> 1
2
n = 9807
y = (0.763, 0.237)
3
n = 61681
y = (0.931, 0.069)


(1) PROPENSION (y / (T)) (GINI: 17.92 % | KS: 16.65 %) (71488 casos)
1
datos.filt[, c(1)]
p < 0.001
≤ 1
> 1
2
n = 29519
y = (0.883, 0.117)
3
datos.filt[, c(1)]
p < 0.001
≤ 2
> 2
4
n = 24448
y = (0.949, 0.051)
5
n = 17521
y = (0.893, 0.107)

Paradigma de investigación	Tipo de investigación	Problema (de acuerdo a diagnóstico)	Objetivo (primera formulación)	Hipótesis	VARIABLE	DIMENSION
		General	General	General	Variable	
CUANTITATIVA	COMPARATIVA	¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera?	Comparar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera.	El algoritmo de random forest para datos balanceados proporcionará mejores indicadores que para datos no balanceados en el área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera.	Algoritmo de clasificación de random forest para datos balanceados y datos no balanceados del área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera.	INDICADORES: AUC, especificidad, sensibilidad
		Específico		Específico		
		¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador del Área Bajo la Curva (AUC)?	Evaluar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador del Área Bajo la Curva (AUC).	Existen diferencias entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados del área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera utilizando el indicador del Área Bajo la Curva (AUC)		
		¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de especificidad?	Evaluar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de especificidad.	Existen diferencias entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados del área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera utilizando el indicador de especificidad		
		¿Qué diferencias existen entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de sensibilidad?	Evaluar el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados en el área de CRM para el acierto de leads en la aceptación de una campaña de tarjeta de crédito en una entidad financiera utilizando el indicador de sensibilidad.	Existen diferencias entre el algoritmo de clasificación de random forest para datos balanceados y datos no balanceados del área de CRM en el acierto de leads para la aceptación de la campaña de tarjeta de crédito de una entidad financiera utilizando el indicador de sensibilidad		