Santiago de Cali, 30 de Mayo del 2024

Doctor
Diego Luis Linares Ospina
Director Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias
Pontificia Universidad Javeriana de Cali

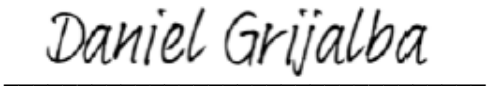
Asunto: Presentación para evaluación del proyecto aplicado

Cordial Saludo,

Con el fin de cumplir con los requisitos exigidos por la Universidad para optar por el título de Magíster en Ciencia de Datos, nos permitimos presentar a su consideración el proyecto denominado "**Machine Learning aplicado a estudios de otorgamiento de créditos en presencia del desbalanceo de clase**", el cual fue realizado por el estudiante Daniel Felipe Grijalba Gonzalez con código 8979773 pertenecientes a la Maestría en Ciencia de Datos, bajo la dirección de Andrés Felipe Ochoa Muñoz y codirección de Isabel Cristina García Arboleda.

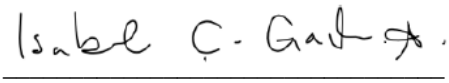
El suscrito director del Proyecto Aplicado autoriza para que se proceda a hacer la evaluación de este proyecto, toda vez que ha revisado cuidadosamente el documento y avala que ya se encuentra listo para ser presentado y sustentado oficialmente.

Atentamente,



Daniel Felipe Grijalba Gonzalez
C.C. 1.144.085.787 de Cali

Andrés Felipe Ochoa Muñoz
C.C. 1.151.941.460 de Cali



Isabel Cristina García Arboleda
C.C. 43.109.746 de Bello (Antioquia)

FICHA RESUMEN
PROYECTO APLICADO – MAESTRÍA EN CIENCIA DE DATOS

TÍTULO: Machine Learning aplicado a estudios de otorgamiento de créditos en presencia del desbalanceo de clases.

1. ÁREA DE TRABAJO: Predicción y Clasificación.
2. TIPO DE PROYECTO (Aplicado, Innovación, Investigación): Aplicado
3. ESTUDIANTE(S): Daniel Felipe Grijalba Gonzalez
4. CORREO ELECTRÓNICO: dafegrijalba@javerianacali.edu.co
5. DIRECCIÓN Y TELEFONO: Calle 60ª # 119C - 84
6. DIRECTOR: Andrés Felipe Ochoa Muñoz
7. VINCULACIÓN DEL DIRECTOR: Profesor Hora Catedra
8. CORREO ELECTRÓNICO DEL DIRECTOR: andresochoa7788@gmail.com
9. CO-DIRECTOR (Si aplica): Isabel Cristina García Arboleda
10. CORREO ELECTRÓNICO DEL CO-DIRECTOR: isabel.garcia@javerianacali.edu.co
11. GRUPO O EMPRESA QUE LO AVALA (Si aplica): No Aplica
12. OTROS GRUPOS O EMPRESAS: No Aplica
13. PALABRAS CLAVE (al menos 5): Credit Score, Machine Learning, Clasificación, Otorgamiento, Riesgo.
14. FECHA DE INICIO: Junio 2023
15. FECHA DE FINALIZACIÓN: Junio 2024
16. RESUMEN:

El otorgamiento de créditos y servicios financieros juega un papel fundamental en la economía y la vida de las personas. La toma de decisiones precisas y efectivas en cuanto a la aprobación o rechazo de solicitudes de créditos es crucial para mantener un equilibrio entre la rentabilidad y la gestión de riesgos. Los modelos de Machine Learning se han convertido en herramientas poderosas para ayudar en la toma de decisiones en diversas áreas, y el sector bancario no es la excepción, estos ayudan tener una mejor precisión y mejoran la eficiencia de los procesos, lo que les permitiría reducir los riesgos asociados al otorgamiento de crédito y optimizar la asignación de recursos.

Pontificia Universidad
JAVERIANA
Cali

Machine Learning aplicado a estudios de otorgamiento de créditos en presencia del desbalanceo de clases

Daniel Felipe Grijalba Gonzalez

Proyecto Aplicado para optar al título de:
Magister en Ciencia de Datos

Director:
Andrés Felipe Ochoa Muñoz

Codirectora:
Isabel Cristina García Arboleda

Pontificia Universidad Javeriana de Cali
Facultad de Ingeniería y Ciencias
Maestría en Ciencia de Datos
Santiago de Cali
Mayo del 2024

Contenido

Lista de Figuras

Lista de Tablas

1. Introducción

En el sector bancario, el otorgamiento de créditos y servicios financieros juega un papel fundamental en la economía y la vida de las personas. La toma de decisiones precisas y efectivas en cuanto a la aprobación o rechazo de solicitudes de créditos es crucial para mantener un equilibrio entre la rentabilidad y la gestión de riesgos.

En los últimos años, los modelos de Machine Learning se han convertido en herramientas poderosas para ayudar en la toma de decisiones en diversas áreas, y el sector bancario no es la excepción. Sin embargo, uno de los desafíos más comunes que enfrentan los modelos es el desbalanceo de clases. Este puede tener un impacto significativo en la capacidad predictiva de los modelos, debido a que tienden a sesgarse hacia la clase dominante y presentar dificultades para detectar los casos de interés minoritario. Esto puede resultar en la pérdida de oportunidades de negocio, así como en un aumento del riesgo crediticio.

En este trabajo de grado, se abordo el desafío del desbalanceo de clases en el otorgamiento de créditos en el sector bancario mediante la aplicación de modelos de Machine Learning. Para ello, se utilizaron tres conjuntos de datos públicos de Kaggle, donde, cada uno de ellos presentó una tasa de otorgamiento diferente a las otras. El objetivo principal de este trabajo fue evaluar y comparar diferentes enfoques y técnicas de manejo del desbalanceo de clases, como el *Oversampling*, el *Undersampling* y el SMOTE, en la construcción de modelos de Machine Learning para el otorgamiento de créditos bancarios. Se busco identificar el enfoque más adecuado que permitiera mejorar la capacidad predictiva y reducir el sesgo del desbalanceo de clases, contribuyendo así a una gestión más eficiente del riesgo y una mayor rentabilidad para las instituciones financieras.

2. Definición del problema

2.1. Planteamiento del Problema

En la actualidad, la evaluación del riesgo de un cliente al solicitar un nuevo préstamo es uno de los principales problemas que enfrenta la industria crediticia. Las entidades bancarias deben estar totalmente seguras de que los clientes poseen la capacidad de realizar los pagos oportunamente para no tener pérdidas. En el pasado, se han utilizado procedimientos empíricos para hacer esto, cómo revisar los informes de crédito, determinar los ingresos y revisar el historial laboral entre otras cosas. Sin embargo, estos procedimientos pueden ser demasiado engorrosos, costosos y demorados, aparte, ralentizan el proceso, por lo cual se vuelve ineficiente y poco fiable.

A lo largo de los años, el problema del otorgamiento de crédito ha sido abordado mediante el desarrollo de técnicas y modelos estadísticos conocidos como "Credit Scoring". Estos métodos surgieron en la década de 1960 y se centran en proporcionar a los prestamistas herramientas para evaluar de manera más precisa la elegibilidad de los clientes para recibir un préstamo, especialmente en el caso de créditos de consumo [8]. Estos enfoques, basados en análisis históricos y clasificaciones, buscan minimizar el error al asignar préstamos a los clientes, logrando así una mayor eficiencia y confiabilidad tanto para los usuarios como para las entidades financieras [9].

La aplicación de algoritmos de Machine Learning son muy útiles en la evaluación crediticia. Estos permiten procesar y analizar grandes volúmenes de datos y su vez detectar tendencias o patrones que podrían pasar desapercibidos con enfoques tradicionales, permitiendo así predecir si un cliente posee un perfil de riesgo alto (no realiza los pagos a tiempo) o bajo (realiza los pagos a tiempo). Con lo cual, las entidades bancarias pueden tomar mejores decisiones en función de otorgar el crédito o no a cada cliente en particular, de esta forma se puede reducir el riesgo y optimizar la asignación de recursos.

Además, es importante considerar el tema del desequilibrio de clases en la calificación crediticia. En muchos casos, el porcentaje de clientes que cumple con sus obligaciones de pago puede ser significativamente mayor que el porcentaje de clientes que incumple [10, 11, 12], este desequilibrio puede hacer que el modelo se desvíe hacia el grupo mayoritario, lo que afectará la precisión y confiabilidad de las predicciones [13].

2.2. Formulación del problema

En este trabajo de grado se desarrollaron una serie de modelos de machine learning, los cuales ayudaron a responder las siguientes preguntas:

- ¿Cuál algoritmo de machine learning ofrece un mejor rendimiento para predecir si al cliente se le otorga un crédito o no, en presencia de clases desbalanceadas?
- ¿Cuáles son las variables más relevantes para la predicción de riesgo crediticio?
- ¿Cuáles son las técnicas más efectivas para abordar el desbalanceo de clases en la predicción de riesgo crediticio utilizando modelos de machine learning?
- ¿Mejora el desempeño de los algoritmos de machine learning frente a las metodologías tradicionales?
- A partir de los algoritmos de machine learning. ¿Se puede encontrar un modelo que sea estable bajo varios niveles de desbalanceo?
- ¿El tuning de hiperparámetros ayuda en el desempeño de los algoritmos de machine learning?

3. Objetivos del proyecto

3.1. Objetivo general

Desarrollar un modelo de otorgamiento de crédito a partir de técnicas de aprendizaje automático y teniendo en cuenta el desbalanceo de clases.

3.2. Objetivos específicos

1. Abordar el desequilibrio de clases mediante las técnicas de Oversampling, Undersampling y SMOTE.
2. Ajustar los modelos de Regresión Logística, Árboles de Decisión, Random Forest y XGBoost para predecir y clasificar a los clientes con un perfil de riesgo bajo, identificando aquellos que tienen una alta probabilidad de adquirir un producto con la entidad financiera.
3. Realizar la búsqueda de parámetros óptimos para los algoritmos.
4. Evaluar el rendimiento de los modelos planteados mediante los indicadores de desempeño.
5. Construir un dashboard para la visualización de los resultados obtenidos en el estudio.

3.3. Justificación

Las investigaciones en este campo han demostrado la efectividad del Machine Learning en la evaluación crediticia. Por ejemplo, en el artículo [14], realizaron un estudio sobre ratios financieros y análisis discriminante para predecir la bancarrota corporativa, encontrando un modelo altamente preciso para predecir la bancarrota, alcanzando un 94 % de accuracy en una muestra inicial y un 95 % de accuracy en varias muestras secundarias. Por otra parte, en [15] utilizaron el enfoque de máquinas de soporte vectorial (SVM) para desarrollar un modelo de scoring crediticio. Los resultados revelaron que el modelo basado en SVM demostró una notable competividad en términos de precisión en la clasificación, logrando un *accuracy* superior al 70 %.

Las anteriores investigaciones demuestran cómo el Machine Learning puede ser aplicado en la evaluación crediticia con resultados prometedores, dado que el otorgamiento de créditos en el sector financiero es un proceso fundamental para garantizar la sostenibilidad y rentabilidad de estas entidades, con estos algoritmos se puede mejorar la evaluación del riesgo crediticio y la toma de decisiones, lo que resulta en una selección más precisa de clientes y una reducción de pérdidas financieras.

Por otra parte, las empresas financieras se enfrentan a otro desafío particular: cada una posee una calidad de cartera, lo que se traduce en retos distintos a la hora de evaluar el riesgo crediticio. Para abordar esta realidad compleja, es necesario considerar diversos escenarios de desbalanceo, que reflejen la variabilidad en el porcentaje de clientes que presentan dificultades en el cumplimiento de sus obligaciones de pago. Esto puede compararse con la realidad, donde existen empresas con una proporción menor de clientes en situación de riesgo crediticio y otras con una cartera más sólida y confiable. Al tener en cuenta estos diferentes escenarios de desbalanceo, se podrá desarrollar un enfoque más adaptado y efectivo en la evaluación crediticia, que permita tomar decisiones informadas y mitigar los riesgos asociados.

Las entidades financieras se beneficiarían al contar con un modelo de evaluación crediticia más preciso y eficiente, lo que les permitiría reducir los riesgos asociados al otorgamiento de crédito y optimizar la asignación de recursos. Por otro lado, los clientes también se verían favorecidos, ya que un proceso de evaluación crediticia más objetivo y preciso aumentaría sus posibilidades de obtener créditos en condiciones favorables, facilitando así el acceso a financiamiento para proyectos personales y comerciales.

4. Marco Teórico y Antecedentes

En el presente capítulo se realiza un revisión literaria de las metodologías estadísticas que se llevaron acabo para el desarrollo de los objetivos del estudio, se definen los conceptos relacionados al problema de interés y se recopilan trabajos donde se han abordado temáticas similares al problema objetivo de estudio.

4.1. Marco Conceptual

4.1.1. Credit scoring

El credit scoring se refiere a todas las metodologías y modelos estadísticos utilizados por los prestamistas para evaluar y otorgar crédito, principalmente en el ámbito del consumo. Estas técnicas tienen como objetivo determinar quién es elegible para recibir crédito, cuánto crédito se le puede otorgar y bajo qué condiciones.

Estas técnicas permite evaluar el riesgo asociado con prestar dinero a un cliente específico. Un prestamista debe tomar dos tipos de decisiones: en primer lugar, decidir si otorgar o no crédito a un nuevo solicitante, y en segundo lugar, cómo tratar a los clientes existentes, lo cual incluye decisiones como aumentar o no su límite de crédito [8].

El objetivo principal de estas técnicas es clasificar a los clientes como morosos o no morosos a través de un análisis de riesgo crediticio más efectivo, logrando una mayor precisión en los parámetros de los modelos utilizados. Para lograr esto, se emplea la información de los clientes con el fin de identificar las relaciones existentes entre sus características y la calidad de su historial crediticio, es decir, determinar si su historial ha sido considerado como "bueno" o "malo" [8, 16].

4.2. Marco Teórico

4.2.1. Modelos Lineales Generalizados

El Modelo Lineal Generalizado o GLM (por sus siglas en inglés) surgió del trabajo realizado por [17]. Este análisis entra en juego, cuando la variable de respuesta dada las covariables

no se distribuye normal, es decir la distribución de los errores no es normal [18].

Los GLM son definidos en términos de un conjunto de variables aleatorias independientes $Y_1, ..., Y_n$, cada una con una distribución de la familia exponencial y las siguientes propiedades [19]:

1. La distribución de cada Y_i tiene la forma canónica y depende de un solo parámetro θ_i:

$$f(y_i; \theta_i) = exp[y_i b_i(\theta_i) + c_i(\theta_i) + d(y_i)] \tag{4-1}$$

   donde $b_i(\theta_i)$, $c_i(\theta_i)$ son funciones conocidas y $b_i(\theta_i)$ es el parámetro natural de la distribución.

2. La distribución de todos los $Y_i's$ tiene la misma forma, por ende no son necesarios los subíndices i en b,c y d. Por lo tanto, la función de densidad de probabilidad conjunta de $Y_1, ..., Y_n$ sería:

$$f(y_1, ..., y_n; \theta_1, ..., \theta_n) = \prod_{i=1}^{n} exp\left[y_i b(\theta_i) + c(\theta_i) + d(y_i)\right] = exp\left[\sum_{i=1}^{n} y_i b(\theta_i) + \sum_{i=1}^{n} c(\theta_i) + \sum_{i=1}^{n} d(y_i)\right] \tag{4-2}$$

Los parámetros θ_i no interesan de manera directa (ya que puede haber uno para cada observación). Para la especificación del modelo, nos interesa un conjunto más pequeño de parámetros $\beta_1, ..., \beta_p$ (donde $p < n$). Supongamos que $E(Y_i) = \mu_i$, donde μ_i es una función de θ_i. Para un GLM hay una transformación de μ_i:

$$g(\mu_i) = \mathbf{x}_i^T \vec{\beta}$$

Donde $g(.)$ es una función monótona llamada función de enlace, $\vec{\mathbf{x}_i}$ es el vector de valores observados para las variables explicativas de tamaño $p \times 1$ y $\vec{\beta}$ es el vector de parámetros de tamaño $p \times 1$. Cayuela [20] define que la función de enlace es la encargada de la linealizar la relación entre el valor medio de la variable respuesta y la variable independiente mediante la transformación del valor medio de la variable respuesta. La solución de máxima verosimilitud para estimar los parámetros en los modelos lineales generalizados implica el uso de mínimos cuadrados generalizados de manera iterativa [18].

4.2.2. Regresión Logística

El modelo de regresión logística es un GLM para modelar datos con respuestas binarias [21]. Consideremos un conjunto de datos que contienen n observaciones $(y_i, x_i) : i = 1, ..., n$,

donde y_i es la variable aleatoria binaria:

$$Y = \begin{Bmatrix} 1 & Si\ el\ resultado\ es\ exito \\ 0 & Si\ el\ resultado\ no\ es\ exito \end{Bmatrix}$$

para el i-ésimo individuo y x_i es su vector de variables explicativas asociado con probabilidad $Pr(Y_i = 1) = \pi_i$, su distribución de probabilidad se puede escribir en forma de la familia exponencial, tal como se ve en 4-3.

$$f_Y(y_i) = exp\left[\frac{y_i log\left(\frac{\pi_i}{1-\pi_i}\right) - (-log(1-\pi_i))}{1} + 0\right] \tag{4-3}$$

Utilizando el parámetro natural $b_i(\theta_i) = log(\pi_i/(1-\pi_i)$, obtenemos la siguiente fórmula:

$$logit(\pi_i) = log\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{x}_i^t\vec{\beta} \tag{4-4}$$

equivalentemente se obtiene:

$$\pi_i = logistic(\mathbf{x}_i^t\vec{\beta}) = \frac{exp(\mathbf{x}_i^t\vec{\beta})}{1 + exp(\mathbf{x}_i^t\vec{\beta})} \tag{4-5}$$

4.2.3. Estimación mediante los mínimos cuadrados ponderados iterativos

Según [19, 21], los mínimos cuadrados no son apropiados cuando la distribución de la variable respuesta (Y) no es continua, en los modelos lineales generalizados la estimación de los parámetros se realiza bajo la estructura del método de máxima verosimilitud, teniendo en cuenta que este método de estimación tiene un relación muy estrecha con los mínimos cuadrados ponderados iterativo. Partiendo del hecho de que se cuenta con un conjunto de variables aleatorias $Y_1, Y_2, ..., Y_n$, las cuales tienen la misma distribución, pero están indexadas bajo diferentes parámetro θ_i y cumplen con todas las propiedades de los modelos lineales generalizados, se tiene que para cada Y_i la función de log-verosimilitud es :

$$l_i = y_i b(\theta_i) + c(\theta_i) + d(y_i) \tag{4-6}$$

donde las funciones $b(.)$, $c(.)$ y $d(.)$ son conocidas, además se tiene que:

$$E(Y_i) = \mu_i = \frac{-c^{'}(\theta_i)}{b^{'}(\theta_i)} \tag{4-7}$$

$$V(Y_i) = \frac{b''(\theta_i)c'(\theta_i) - c''(\theta_i)b'(\theta_i)}{(b'(\theta_i))^3} \tag{4-8}$$

$$g(\mu_i) = \mathbf{x}_i^T \vec{\beta} = \eta_i \tag{4-9}$$

donde x_i es un vector con elementos x_{ij}, $j = 1, ...p$. Ahora, la log-verosimilitud para todos los $Y_i^{'} s$ es:

$$l = \sum_{i=1}^{n} l_i = \sum_{i=1}^{n} y_i b(\theta_i) + \sum_{i=1}^{n} c(\theta_i) + \sum_{i=1}^{n} d(y_i)$$

Para obtener los estimadores de máxima verosimilitud para el parámetro β_j necesitamos usar la regla de la cadena para la diferenciación:

$$\frac{\partial l}{\partial \beta_j} = U_j = \sum_{i=1}^{n} \left[\frac{\partial l_i}{\partial \beta_j} \right] = \sum_{i=1}^{n} \left[\frac{\partial l_i}{\partial \theta_i} . \frac{\partial \theta_i}{\partial \mu_i} . \frac{\partial \mu_i}{\partial \beta_j} \right] \tag{4-10}$$

Por lo tanto, la función score obtenida de (4-10), es:

$$U_j = \sum_{i=1}^{n} \left[\frac{(y_i - \mu_i)}{var(Y_i)} x_{ij} \left(\frac{\partial \mu_i}{\partial \eta_i} \right) \right] \tag{4-11}$$

La matrix de varianzas-covariazas de U_j's es:

$$J_{jk} = E[U_j U_k]$$

la cual forma la matriz de información J. De (4-11)

$$J_{jk} = E \left\{ \sum_{i=1}^{n} \left[\frac{Y_i - \mu_i}{var(Y_i)} x_{ij} \left(\frac{\partial \mu_i}{\partial \eta_i} \right) \right] \sum_{l=1}^{n} \left[\frac{Y_l - \mu_l}{var(Y_l)} x_{lk} \left(\frac{\partial \mu_l}{\partial \eta_l} \right) \right] \right\}$$

$$= \sum_{i=1}^{n} \frac{E\left[(Y_i - \mu_i)^2\right]}{\left[var(Y_i)\right]^2} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 \tag{4-12}$$

porque $E\left[(Y_i - \mu_i)(Y_l - \mu_l)\right] = 0$ para $i \neq 1$ como los $Y_i^{'} s$ son independientes. Utilizando $E\left[(Y_i - \mu_i)^2\right] = var(Y_i)$, (4-12) se puede simplificar a:

$$J_{jk} = \sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 \tag{4-13}$$

El algoritmo de optimización Newton-Raphson, para el método scoring se generaliza a

$$\mathbf{b}^{(m)} = \mathbf{b}^{(m-1)} + \left[J^{(m-1)}\right]^{-1} \mathbf{U}^{(m-1)}, \tag{4-14}$$

donde $\mathbf{b}^{(m)}$ es un vector de estimaciones de los parámetros $\beta_1, ..., \beta_p$ en la m-ésima interacción. En la ecuación (4-14), $\left[J^{(m-1)}\right]^{-1}$ Es la inversa de la matriz de información con elementos J_{jk} dada por (4-13), y $\mathbf{U}^{(m-1)}$ es el vector de elementos dado por (4-11), todos evaluados en $\mathbf{b}^{(m-1)}$. Si ambos lados de la ecuación (4-14) son multiplicados por $J^{(m-1)}$ obtenemos:

$$\left[J^{(m-1)}\right] \mathbf{b}^{(m)} = \mathbf{b}^{(m-1)} + \left[J^{(m-1)}\right] \mathbf{U}^{(m-1)}, \tag{4-15}$$

Desde (4-13), J puede ser escrito como:

$$\mathbf{J} = \mathbf{X}^T \mathbf{W} \mathbf{X}$$

Donde $\mathbf{W}$ es la matriz diagonal $n \times n$ con elementos:

$$w_{ii} = \frac{1}{var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 \tag{4-16}$$

La expresión en el lado derecho de (4-15) es el vector con elementos:

$$\sum_{k=1}^{p} \sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 b_k^{(m-1)} + \sum_{i=1}^{n} \frac{(y_i - \mu_i)x_{ij}}{var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i}\right)$$

evaluado en $\mathbf{b}^{(m-1)}$. Esto se deduce de la ecuación (4-13) y (4-11). Así, el lado derecho de la ecuación (4-15) se puede escribir como:

$$\mathbf{X}^T \mathbf{W} \mathbf{z}$$

donde $\mathbf{z}$ tiene elementos:

$$z_i = \sum_{k=1}^{p} x_{ik} b_k^{(m-1)} + (y_i - \mu_i) \left(\frac{\partial \eta_i}{\partial \mu_i}\right) \tag{4-17}$$

con μ_i y $\partial\eta_i/\partial\mu_i$ evaluadas en $\mathbf{b}^{(m-1)}$.

Por lo tanto, la ecuación iterativa (4-15), se puede escribir como:

$$\mathbf{X}^T\mathbf{W}\mathbf{X}\mathbf{b}^{(m)} = \mathbf{X}^T\mathbf{W}\mathbf{z} \tag{4-18}$$

Se puede observar que posee la misma forma que las ecuaciones normales obtenidas para un modelo lineal obtenido por mínimos cuadrados ponderados, con la excepción que se debe resolver iterativamente ya que, $\mathbf{z}$ y $\mathbf{W}$ dependen de $\mathbf{b}$.

4.2.4. Árboles de Decisión

Uno de los métodos más eficientes para generar clasificadores a partir de datos es mediante la construcción de árboles de decisión, estos árboles representan un modelo jerárquico de aprendizaje supervisado que utiliza divisiones recursivas en nodos de decisión para identificar regiones locales [1]. Estos árboles son no paramétricos, lo que significa que no asumen una forma paramétrica específica para la densidad de las clases.

El proceso de construcción de un árbol de decisión sigue una estrategia de arriba hacia abajo, buscando una solución en un espacio de búsqueda. Esto garantiza la obtención de un árbol relativamente simple (no necesariamente el más simple). Cada nodo del árbol corresponde a una prueba de atributos, y las ramas salientes representan los posibles resultados de dicha prueba. En la Figura **4-1**, se observa un árbol de decisión simple para la clasificación de dos clases con dos atributos de entrada.

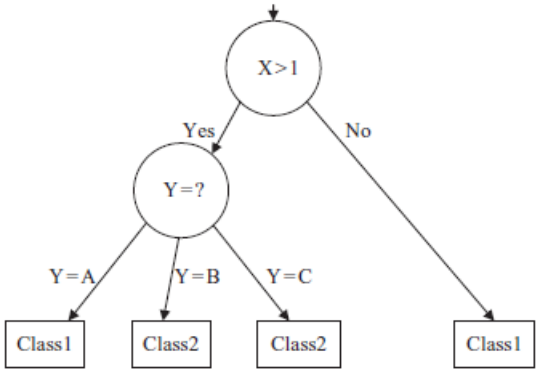


Figura 4-1.: Representación gráfica del árbol de decisión, fuente: [1].

Unos de los parámetros más comunes de este algoritmo son:

- max_depth: Este parámetro establece la profundidad máxima que puede tomar el árbol. A mayor profundidad, se aumenta el riesgo de sobreajuste del modelo.

- max_features: Este parámetro ayuda a controlar el número máximo de variables que se consideran al buscar la mejor división en cada nodo del árbol. Al tomar un menor número de variables, se puede ayudar a evitar el sobreajuste del modelo.
- ccp_alpha: Este parámetro ayuda a controlar la complejidad del árbol mediante la poda de nodos, evitando el sobreajuste.
- min_samples_split: Este parámetro determina el número mínimo de registros requeridos para dividir un nodo del árbol.

4.2.5. Random Forest

El algoritmo *Random Forest* fue propuesto por [22], esta es una técnica de aprendizaje supervisado que se basa en la generación de múltiples árboles de decisión a partir de un conjunto de datos de entrenamiento. Estos árboles trabajan de manera individual y sus resultados se combinan para formar un modelo único y más robusto [2]. El proceso de construcción de cada árbol consta de dos etapas:

- Se crea un conjunto de árboles de decisión utilizando el conjunto de datos. Cada árbol se construye con un subconjunto aleatorio de variables, representando un número menor m de predictores seleccionados, donde m es menor que el número total de predictores M.
- Cada árbol crece hasta su máxima extensión.

Cada árbol producido por el algoritmo *Random Forest* incluye un conjunto de observaciones seleccionadas aleatoriamente mediante el método de *bootstrap*. Las observaciones no utilizadas en la construcción de cada árbol, también conocidas como "out of the bag", se emplean para validar el modelo. Las salidas de todos los árboles se combinan en una salida final Y mediante una regla de ensamblado, que generalmente consiste en calcular el promedio cuando las salidas son numéricas, o realizar un conteo de votos cuando las salidas son categóricas, esta idea se puede observar en la Figura **4-2**.

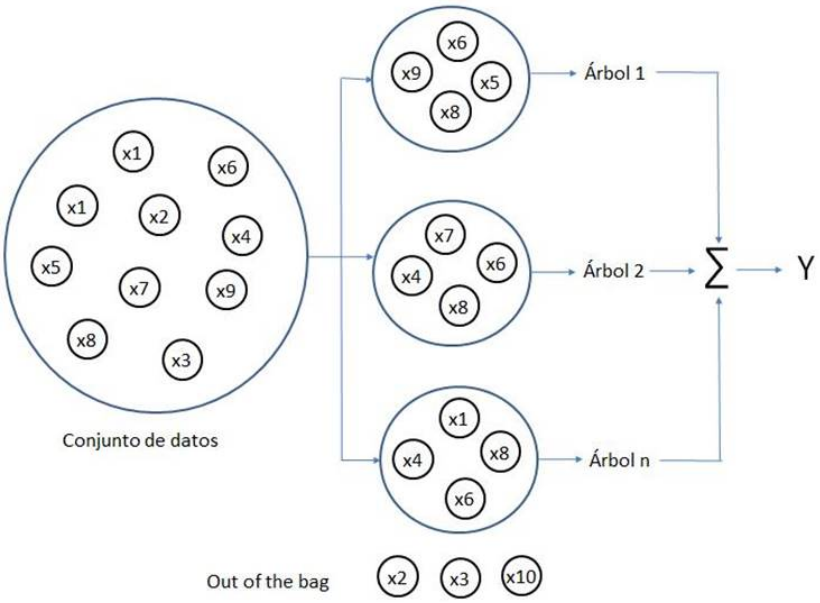


Figura 4-2.: Representación gráfica del algoritmo *Random Forest*, fuente: [2].

El algoritmo ofrece varias ventajas destacadas [23]:

- Versatilidad en su aplicación: Puede utilizarse tanto para clasificación como para predicción. En el caso de clasificación, cada árbol emite un "voto" por una clase y el modelo final selecciona la clase con mayor número de votos entre todos los árboles. En predicción, el resultado del modelo es el promedio de las salidas de todos los árboles.
- Entrenamiento sencillo: A pesar de su rendimiento comparable a técnicas más complejas, el modelo es más fácil de entrenar.
- Eficiencia y precisión: Presenta un rendimiento altamente eficiente y es particularmente preciso en conjuntos de datos grandes.
- Manejo de múltiples predictores: Puede manejar cientos de predictores sin necesidad de excluir ninguno y, además, es capaz de estimar la importancia de cada predictor. Por esta razón, también se utiliza en técnicas de reducción de dimensionalidad.

Unos de los parámetros más comunes de este algoritmo son:

- n_estimators: Este parámetro determina la cantidad de árboles de decisión que se crearan en el random forest. A un mayor número de árboles se obtendrá un modelo mas robusto preciso, sin embargo, aumenta el tiempo de entrenamiento.
- max_depth: Este parámetro establece la profundidad máxima que pueden tomar los árboles en el bosque. A mayor profundidad, se aumenta el riesgo de sobreajuste del modelo.

- max_features: Este parámetro ayuda a controlar el número máximo de variables que se consideran para la división en los árboles individuales. Al tomar un menor número de variables, se puede ayudar a evitar el sobreajuste del modelo.
- min_samples_split: Este parámetro determina el número mínimo de registros requeridos para realizar una división adicional en el nodo del árbol.
- bootstrap: Este parámetro indica si se realiza el muestreo bootstrap al crear cada árbol del bosque.
- criterion: Este parámetro determina la medida utilizada para evaluar la calidad de una división en los árboles.

4.2.6. XGBoost

El algoritmo *XGBoost* (Extreme Gradient Boosting) propuesto por [24], es una técnica de aprendizaje supervisado también basada en árboles de decisión, el cual utiliza el principio del *boosting*. Este consiste en un proceso de ensamblado secuencial de árboles de decisión. En este enfoque, cada árbol se construye para aprender de los resultados de los árboles previos y corregir los errores cometidos, siguiendo un proceso llamado "gradiente descendente" [2], en la Figura **4-3** se puede observar dicho proceso. Este proceso continúa hasta que ya no sea posible corregir más errores, lo que conduce a la creación de un modelo con un mayor poder predictivo y una mayor estabilidad en los resultados.

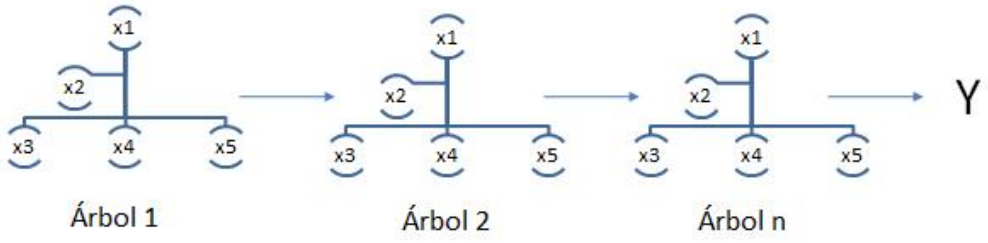


Figura 4-3.: Representación gráfica del algoritmo *XGBoost*, fuente: [2].

El algoritmo *XGBoost* funciona de la siguiente manera [2]:

1. Se obtiene un árbol inicial F_0 para predecir la variable objetivo "y", el resultado se asocia con un residual $(y - F_0)$.
2. Se obtiene un nuevo árbol h_1 que ajusta al error del paso previo.
3. Los resultados de F_0 y h_1 se combinan para obtener el árbol F_1, donde el error cuadrático medio de F_1 será menor que el de F_0:

$$F_1(x) = F_0(x) + h_1(x) \tag{4-19}$$

4. Este proceso se sigue iterativamente hasta que el error es minimizado lo más posible de la siguiente forma:

$$F_m(x) = F_{m-1}(x) + h_m(x) \tag{4-20}$$

Este algoritmo tiene varias ventajas, como su capacidad para manejar grandes bases de datos con muchas variables, manejar valores faltantes y producir resultados muy precisos. También ofrece una excelente velocidad de ejecución. Sin embargo, también tiene algunas limitaciones, como el consumo de recursos informáticos al trabajar con bases de datos muy grandes, la necesidad de ajustar correctamente los parámetros y la limitación de que solo puede procesar datos numéricos.
Unos de los parámetros más comunes de este algoritmo son:

- n_estimators: Este parámetro determina la cantidad de árboles se crearán en el ensamble. A un mayor número de árboles se obtendrá un modelo más robusto preciso, sin embargo, aumenta el tiempo de entrenamiento.
- max_depth: Este parámetro establece la profundidad máxima que pueden tomar los árboles en el ensamble. A mayor profundidad, se aumenta el riesgo de sobreajuste del modelo.
- learning_rate: Este parámetro controla la tasa de aprendizaje y contribución de cada árbol al modelo final. Una mayor tasa indica un ajuste más rápido.
- min_child_weight: Este parámetro controla la complejidad de los árboles mediante una suma mínima de pesos de las muestras requeridas en la hoja de cada árbol.
- subsample: Este parámetro controla el número de registros que se utilizaran para entrenar cada árbol.
- gamma: Este parámetro ayuda a controlar la poda de un árbol mediante un umbral mínimo.
- colsample_bytree: Este parámetro controla el número de variables que se utilizaran para entrenar cada árbol.

4.2.7. Métricas de desempeño

Por lo general, la evaluación del desempeño de los modelos implica analizar su rendimiento en conjuntos de datos de prueba. Para evaluar la capacidad de los ajustes realizados y comparar diferentes modelos, se utilizan diversas métricas, como *accuracy*, *precision*, *recall*, la especificidad, la matriz de confusión y la curva ROC. Estas medidas son indicadores efectivos para cuantificar el rendimiento y comparar los resultados obtenidos.

Matriz confusión

Una manera integral de presentar los resultados de la evaluación de clasificaciones binarias es a través del uso de matrices de confusión [25]. Esto implica la comparación de las clases reales con las predichas. En la Figura **4-4**, se puede observar un ejemplo de una matriz de confusión. En donde, las columnas de la matriz representan el conteo de las clases estimadas, mientras que las filas representan el conteo de las clases reales.

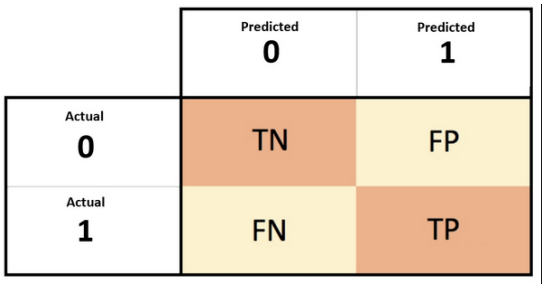


Figura 4-4.: Representación gráfica de la matriz de confusión para el caso binario, fuente [3].

La matriz de confusión se divide en cuatro celdas, las cuales nos permiten el cálculo de diferentes indicadores de desempeño:

- Verdaderos positivos (TP): Representa los casos en los que el modelo predijo correctamente la clase positiva.
- Verdaderos negativos (TN): Representa los casos en los que el modelo predijo correctamente la clase negativa.
- Falsos positivos (FP): Representa los casos en los que el modelo predijo incorrectamente la clase positiva cuando la clase real era negativa.
- Falsos negativos (FN): Representa los casos en los que el modelo predijo incorrectamente la clase negativa cuando la clase real era positiva.

Recall

La métrica mide la capacidad del modelo para detectar correctamente los casos positivos de una clase. Un valor alto de recall indica que el modelo tiene una baja tasa de falsos negativos, este se calcula dividiendo el número de verdaderos positivos (TP) entre la suma de verdaderos positivos y falsos negativos (FN).

$$Recall = \frac{TP}{TP + FN} \tag{4-21}$$

Precision

Es una métrica que se centra en la proporción de casos positivos que el modelo clasifica correctamente, es decir, mide la exactitud de las predicciones positivas realizadas por un modelo. Se calcula dividiendo el número de verdaderos positivos (TP) entre la suma de verdaderos positivos y falsos positivos (FP).

$$Precision = \frac{TP}{TP + FP} \tag{4-22}$$

Accuracy

Es una métrica que mide la proporción de predicciones correctas realizadas por un modelo en relación con el total de predicciones realizadas. Matemáticamente, se calcula dividiendo el número de predicciones correctas (TP y TN) entre el número total de ejemplos en el conjunto de datos.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4-23}$$

Esta proporciona una medida general del rendimiento del modelo al evaluar su capacidad para predecir correctamente todas las clases. Sin embargo, puede ser engañoso en casos donde hay un desequilibrio en la distribución de las clases.

F1-Score

El F1-Score es una métrica que combina tanto la precisión como el recall para proporcionar una medida equilibrada del rendimiento del modelo. El F1-score se calcula como la media armónica de la *precision* y el *recall*:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4-24}$$

Se suele utilizar cuando se desea tener en cuenta tanto los falsos positivos como los falsos negativos en la evaluación del rendimiento del modelo. Proporciona una medida equilibrada entre el *precision* y el *recall*, y es especialmente útil cuando hay un desequilibrio entre las clases o cuando tanto el *precision* como el *recall* son igualmente importantes.

ROC y AUC-ROC

La curva ROC (Receiver Operating Characteristic) es otra herramienta común utilizada con clasificadores binarios, esta representa la tasa de verdaderos positivos frente (*recall*) a la tasa de falsos positivos [4]. Por lo tanto, la curva ROC traza el *recall* frente a 1 – especificidad.

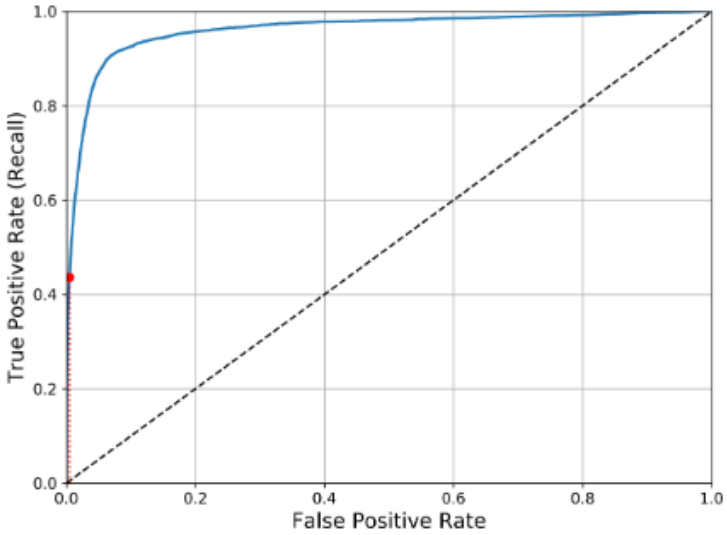


Figura 4-5.: Representación gráfica de la curva roc, fuente: [4].

El AUC-ROC (Area Under the Receiver Operating Characteristic curve) representa el área bajo la curva ROC y proporciona una medida numérica del rendimiento global del modelo. Esta métrica resume la capacidad del modelo para clasificar correctamente tanto los casos positivos como los casos negativos en diferentes umbrales de decisión. El valor del AUC-ROC varía entre 0 y 1, a un mayor valor indica un mejor rendimiento de clasificación del modelo.

4.2.8. Randomized Search

A diferencia de la búsqueda en cuadrícula (*Grid Search*), que prueba todas las combinaciones posibles de los hiperparámetros, la técnica del *Randomized Search* evalúa un número determinado de combinaciones aleatorias. En lugar de explorar exhaustivamente todo el espacio de búsqueda de hiperparámetros, esta técnica selecciona aleatoriamente un valor para cada hiperparámetro en cada iteración [4]. Esto permite una búsqueda más eficiente, especialmente cuando el espacio de búsqueda de hiperparámetros es grande y no es práctico evaluar todas las combinaciones posibles.

En la Figura **4-6**, podemos observar como se tiene el espacio paramétrico y como aleatoriamente, se van probando distintas combinaciones de hiperparámetro.

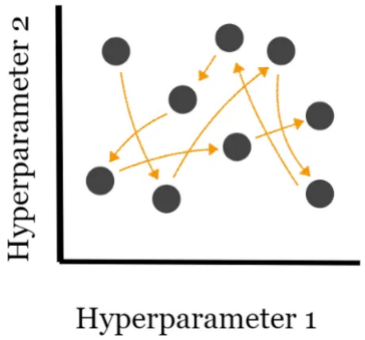


Figura 4-6.: Representación gráfica de combinaciones aleatorias de hiperparámetros para un *Random Search*, fuente: [5].

Este enfoque tiene unos beneficios:

- Si se ejecuta la búsqueda aleatoria durante 1.000 iteraciones, este enfoque explora 1.000 valores diferentes para cada hiperparámetro.
- Tienes más control sobre el presupuesto de cómputo que se desea asignar a la búsqueda de hiperparámetros, simplemente estableciendo el número de iteraciones.

4.2.9. Técnicas de balanceo

El desbalanceo de clases ocurre cuando una clase tiene muchos más registros que otra clase en el conjunto de datos, en la Figura **4-7** se puede observar un claro ejemplo de clases desbalanceadas. Este puede ser un gran problema, dado que a los algoritmos de ML tienden a tener dificultades para aprender patrones de las clases minoritarias. Para abordar este problema, existen diversas técnicas. Entre las más conocidas y utilizadas para el manejo del desbalanceo de clases se encuentran el *undersampling*, *oversampling* y SMOTE [13, 26, 27, 28, 29].

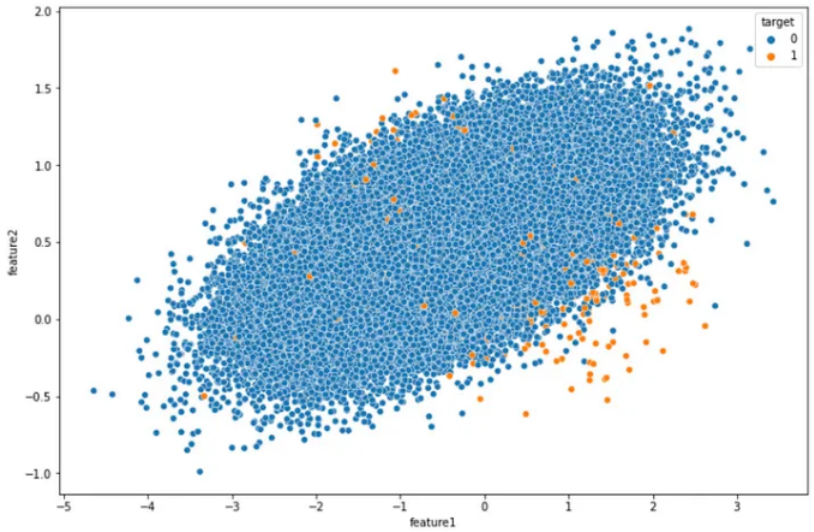


Figura 4-7.: Representación gráfica del desbalanceo de datos, fuente: [6].

Oversampling

La técnica de *oversampling* (sobremuestreo) es una de las estrategias más simples para manejar el desbalanceo de clases, esta consiste en aumentar la cantidad de registros de la clase minoritaria mediante un muestreo aleatorio con repetición para igualarla a la cantidad de registros en la clase mayoritaria. En la Figura **4-8** se observa como aumentan los registros de la clase minoritaria.

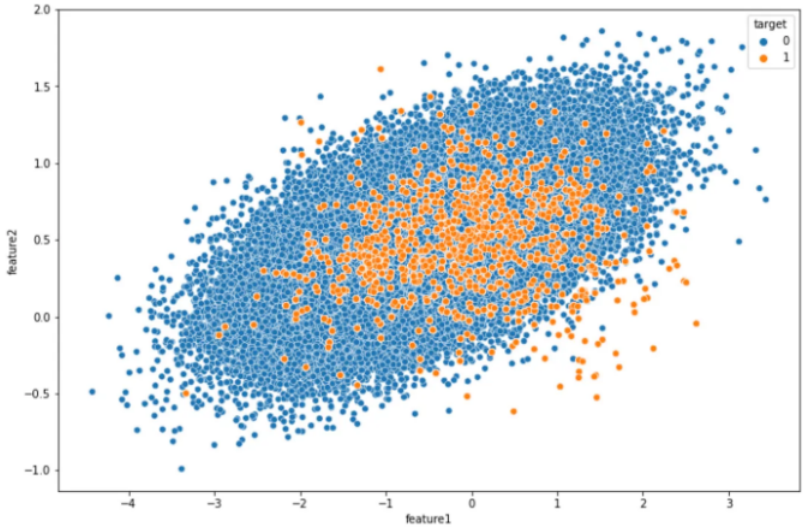


Figura 4-8.: Representación gráfica de la técnica del *oversampling*, fuente: [6].

Undersampling

El *undersampling* (submuestreo) consiste en reducir la cantidad de registros de la clase mayoritaria. Esto se puede lograr eliminando registros al azar de la clase mayoritaria o realizando un muestreo aleatorio de la misma. En la Figura **4-9**, se puede observar cómo disminuye la cantidad de registros de la clase mayoritaria para igualarla a la clase minoritaria.

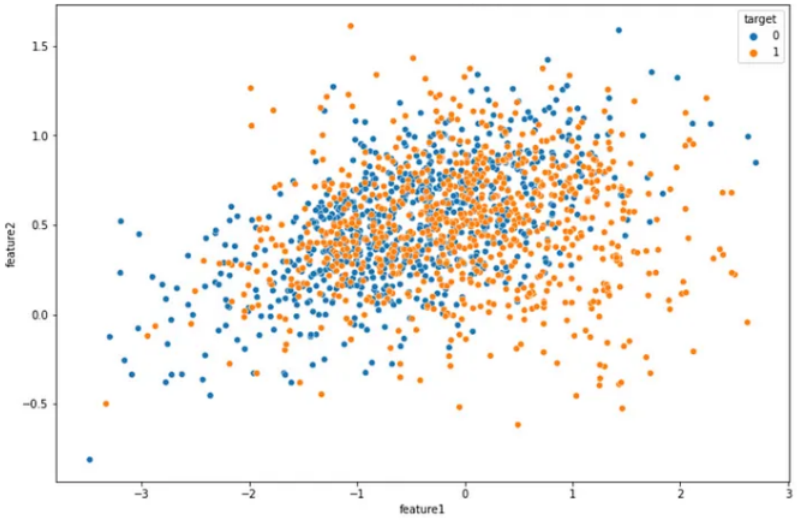


Figura 4-9.: Representación gráfica de la técnica del *undersampling*, fuente: [6].

SMOTE

La Técnica de Sobremuestreo de Minorías Sintéticas (SMOTE) es una técnica estadística para aumentar el número de casos en tu conjunto de datos de manera equilibrada, esta fue propuesta por [30] en 2002. Desde su publicación, esta técnica ha demostrado ser muy exitosa en la aplicación de diferentes dominios para contrarrestar el desequilibrio de clases. Esta técnica funciona generando nuevas clases sintéticas a partir de los casos minoritarios existentes

Las muestras sintéticas se generan tomando la diferencia entre el vector de características x_i en consideración y su vecino más cercano x_{zi}, luego se multiplica esta diferencia por un número aleatorio en el intervalo [0; 1] denominado α y se agrega al vector de características en consideración. Esto provoca la selección de un punto aleatorio a lo largo del segmento de línea entre dos características específicas [31], el nuevo registro o instancia se forma con la siguiente ecuación:

$$x_{new} = x_i + \alpha(x_{zi} - x_i) \tag{4-25}$$

En la Figura **4-10**, se puede observar lo anterior, donde los puntos rojos serían el x_{new} y los dos puntos amarillos serían su vecino más cercano. Al igual que las técnicas mencionadas anteriormente, este enfoque solo se centra en una de las clases, en este caso, en la clase minoritaria.

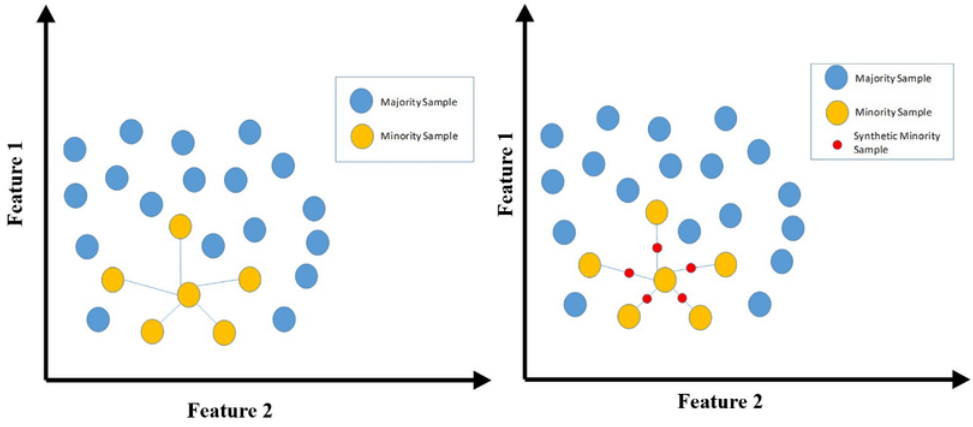


Figura 4-10.: Representación gráfica de la técnica del SMOTE, fuente: [7].

4.2.10. Test de U Mann-Whitney

El test de U Mann-Whitney es una prueba estadística no paramétrica utilizada para realizar comparaciones entre la distribución de dos grupos independientes y determinar si existe una diferencia significativa entre ambos grupos o provienen de una misma población. Esta prueba es el equivalente a la prueba T de muestras independientes, sin embargo, a diferencia de esta última, el test de U de Mann-Whitney no asume normalidad en los datos [32, 33, 34].

El test plantea la siguiente hipótesis:

$$H_0 : \ F(X) = G(Y) \quad vs \quad H_1 : \ F(X) \neq G(Y)$$

en donde $F(X)$ es la distribución de la primer muestra y $G(Y)$ es la distribución de la segunda muestra.

4.2.11. Test Chi-Cuadrado

El test chi-cuadrado o χ^2, es una prueba estadística no paramétrica utilizada para evaluar la independencia o asociación entre dos variables categóricas o grupos mediante tablas de contingencia, bajo la hipótesis de que no existe ningún tipo de relación significativa entre ambas variables categóricas [35, 36, 37].

Se plantea la siguiente hipótesis:

$$H_0: \ F(X) \ y \ G(Y) \ son \ independientes \ \ vs \ \ H_1: \ F(X) \ y \ G(Y) \ no \ son \ independientes$$

en donde $F(X)$ y $G(Y)$ corresponden a las distribuciones de cada grupo o variable categórica.

4.3. Antecedentes

Modelo de Scoring para la segmentación de clientes morosos usando minería de datos en una empresa de cobranzas del Perú [16]

En este trabajo de grado, se desarrolló un Modelo de Scoring con el objetivo de segmentar a los clientes morosos con más de 180 días de mora. El autor utilizó técnicas de minería de datos y análisis predictivo para reducir los costos mediante la reasignación de carteras de cobranza en una empresa de cobranzas en Perú. El modelo se basó en datos históricos recopilados durante el período de junio a septiembre de 2020, y se seleccionó el grupo de clientes con más de 180 días de mora debido a su baja efectividad de recuperación.

Aplicando la metodología CRISP-DM, construyeron cuatro modelos de clasificación: K-Nearest Neighbors, Árbol de Decisión, SVM y Regresión Logística. Después de evaluar los modelos, se seleccionó el modelo de Regresión Logística debido a su alta precisión, con un *accuracy* del 98,4 % y un AUC del 81,51 % durante los cuatro meses de análisis.

Finalmente, se propuso una nueva distribución en la asignación de la cartera, donde se asignó el 20 % de la gestión al Call Interno y el 80 % al Call Externo. Los clientes con mayores probabilidades de pago fueron asignados al Call Interno. Logrando demostrar que la efectividad de pago del Call Interno fue del 6,46 % en promedio mensual durante los cuatro meses, en comparación con el 0,84 % del Call Externo. Además, se logró un mayor volumen de clientes que pagaron en el Call Interno, con un promedio mensual de 192 clientes más en comparación con el Call Externo.

Analysis and comparison of machine learning classification models applied to credit approval [26]

En este artículo los autores tuvieron como objetivo principal analizar diversos algoritmos de Machine Learning para predecir de manera precisa el otorgamiento de crédito y encontrar el mejor modelo que les permitiera encontrar unas reglas óptimas de decisiones para el otorgamiento de créditos al Banco Alemán utilizando también técnicas para el desbalanceo de clases.

Los resultados obtenidos revelaron que el modelo más efectivo fue el de *Gradient Boosting*, alcanzando un *accuracy* del 83,71 %. Estos hallazgos respaldaron la viabilidad y eficacia de la implementación de algoritmos de machine learning en la toma de decisiones de otorgamiento de crédito, y sientan las bases para continuar explorando nuevas estrategias y enfoques que permitan optimizar aún más el proceso.

Predicción de adquisición de un préstamo personal bancario a través del canal de televentas utilizando el algoritmo Random Forest [27]

En este trabajo, el autor buscó crear un modelo de clasificación binaria para predecir la aceptación o el rechazo de préstamos por parte de los clientes durante llamadas telefónicas. Para lograrlo, se empleó el algoritmo *Random Forest* y se aplicaron las técnicas de balanceo *undersampling* y SMOTE para obtener los mejores resultados. Estas estrategias permitieron identificar a los clientes con mayor probabilidad de adquirir el producto, optimizando así las gestiones para priorizar las ventas.

El modelo se desarrolló utilizando una muestra de datos recopilados durante seis meses (de marzo a agosto de 2017) de clientes que poseían al menos una tarjeta de crédito y un préstamo preaprobado con la entidad bancaria. En total, la base de datos abarcó 991.619 registros de clientes.

Como resultado, la implementación del algoritmo ayudó a incrementar las ventas en las campañas y cumplir con las metas mensuales establecidas. Además, al comparar las dos técnicas de balanceo utilizadas, se encontró que el *undersampling* obtuvo mejores resultados en términos de sensibilidad.

Modelo de scoring para crédito de consumo en una entidad del sector solidario [28]

En este trabajo se presentaron diferentes modelos de clasificación, los cuales fueron comparados mediante las métricas AUC, *recall*, *precision*, *accuracy* y *F1 Score*, con el fin de obtener el mejor modelo que permita calcular la probabilidad e incumplimiento de un cliente para nuevos otorgamientos.

Para esto, se utilizaron una base de datos de una entidad del sector solidario, la cual incluía 5.974 créditos de consumo. Se entrenaron varios modelos de machine learning con tecnicas de balanceo para determinar la probabilidad de incumplimiento, encontrando así que, el modelo de *Light Gradient Boosting Machine* obtuvo el mejor rendimiento, con un AUC de 0,7550, *recall* de 0,7111 y *precisio*n de 0,1587. Además, lograron identificar las variables más importantes para inferir la probabilidad de incumplimiento en los créditos de consumo,

las cuales eran la antigüedad, los activos y pasivos totales, los ingresos y egresos mensuales, el valor del crédito, el plazo, el nivel de estudio, el estado civil, la forma de pago y la edad.

Loan default prediction using decision trees and random forest: A comparative study [38]

En este artículo, los autores se enfocaron en el uso de dos algoritmos de machine learning, el Árboles de Decisión y el *Random Forest*, para desarrollar un modelo de otorgamiento de préstamos. El objetivo principal fue identificar si una persona, según ciertas características, tiene una alta probabilidad de incumplir un préstamo, lo que podría ser valioso para entidades financieras como Lending Club.

Para lograr este objetivo, los autores utilizaron un conjunto de datos de Lending Club, el cual contenía aproximadamente 2.2 millones de registros sobre préstamos otorgados entre 2007 y 2015. Este conjunto de datos incluye información sobre el estado de los préstamos y las tasas de interés. Los autores incluyeron en su metodología la preparación de datos, eliminación de campos con datos faltantes, el análisis exploratorio de datos (EDA) para estudiar las relaciones y patrones en los datos, y la construcción de modelos de predicción utilizando los algoritmos de Árboles de Decisión y *Random Forest*, en donde dividieron los datos en conjuntos de entrenamiento y prueba para evaluar el rendimiento de los modelos. Los resultados mostraron que el *Random Forest* superó al Árbol de Decisión, con un *accuracy* del 80 % frente al 73 %, respectivamente. Esto sugiere que el *Random Forest* es una opción más efectiva para predecir el incumplimiento de préstamos en el conjunto de datos suministrado.

5. Metodología

A continuación se presenta la metodología empleada para encontrar el algoritmo de ML más adecuado para el otorgamiento de créditos cuando se tiene el problema del desbalanceo de clases.

5.1. Conjuntos de datos

En este proyecto, se trabajó con tres conjuntos de datos públicos disponibles en la plataforma Kaggle. El primer conjunto de datos, denominado Credit Approval [39], contiene información sobre los préstamos aprobados por un banco, con un total de 9 campos y 4.521 registros. El segundo conjunto de datos, Credit Risk Analysis [40], proporciona información sobre los solicitantes de préstamos y sus características, con el objetivo de determinar su probabilidad de incumplimiento. Este consta de 12 campos y 32.581 registros. Por último, el tercer conjunto de datos, Credit Risk Customers [41], incluye información sobre el otorgamiento de tarjetas de crédito en el año 1994, con un total de 21 campos y 1.000 registros. Además, se empleó Python [42] como lenguaje de programación principal, junto con las bibliotecas Pandas [43], NumPy [44], Scikit-learn [45], XGBoost [24], Imbalanced-learn [46] y Statsmodels [47].

5.2. Preparación de los conjuntos de datos

Para el entendimiento de los conjuntos de datos, se llevó a cabo la homologación de los nombres de los campos, con el objetivo de entender a fondo cada uno de ellos y así, facilitar su posterior manejo y análisis. Además, se realizó un tratamiento y homologación de las variables categóricas, asegurando un entendimiento de estas categorías y una correcta clasificación.

Una vez completada esta etapa inicial, se procedió a realizar un análisis descriptivo entre las variables presentes en cada base de datos y la variable objetivo, aparte, un análisis de correlaciones y truncamiento de valores atípicos mediante los cuantiles. Por último, se llevó a cabo una pre-selección de variables. Se utilizaron los test de U Mann-Whitney con las variables continuas y el test Chi-Cuadrado con las variables categóricas, marcando las variables donde el $valor - p$ de los test fuera mayor a 0,05, esto con el fin de excluir en las

iteraciones aquellas variables que no mostraron una asociación significativa con la variable objetivo.

5.3. Definición espacio paramétrico

Se definieron los espacios paramétricos para los modelos *XGBoost*, *Random Forest* y Árbol de Decisiones, con los cuales se llevó a cabo una búsqueda óptima de parámetros utilizando la técnica del *Randomized Search*. Este proceso implicó la especificación de rangos y valores permitidos para cada hiperparámetro de los modelos.

5.4. Definición niveles de balanceo

Se implementaron diferentes niveles de balanceo, abarcando desde las tasas de desequilibrio propio en cada conjunto de datos hasta alcanzar un nivel de balanceo equitativo del 50/50 para la variable objetivo.

5.5. Modelación

Para el modelamiento, se desarrollo una función llamada *balance_and_train_multiple_models*. Esta función recibe como input el conjunto de datos, el nivel de balanceo y el método de balanceo.

La función comienza dividiendo el conjunto de datos en los conjuntos de *train* y *test*, representando el 70 % y 30 %, respectivamente. Luego, se aplicó un método de balanceo, considerando el método y tasa específica de balanceo previamente establecida. Posteriormente, se definieron los modelos base del *XGBoost*, *Random Forest*, Árbol de Decisiones y la Regresión Logística, con los cuales se realizó una pre selección de variables en función de su importancia y así reducir el número de variables del conjunto de train.

Con la nueva selección de variables, se procedió a entrenar los modelos y buscar los mejores hiperparámetros para cada uno con ayuda de la búsqueda aleatoria. Finalmente, se llevó a cabo una evaluación de cada modelo, registrando las métricas de *accuracy*, *precision*, *recall*, *F1 Score* y AUC. Además, se buscó el punto de corte óptimo para cada iteración y se volvió a evaluar cada modelo para obtener una comparación más precisa y un mejor desempeño de los modelos.

La función se aplicó a los tres conjuntos de datos con el propósito de obtener resultados representativos y comparables, teniendo en cuenta una variedad de tasas de balanceo.

5.6. Dashboard

Con el objetivo de proporcionar una presentación efectiva y accesible de los resultados derivados de la aplicación de la función *balance_and_train_multiple_models* en los tres conjuntos de datos, y también facilitar la elección de la mejor combinación entre el modelo, método y tasa de balanceo, se diseñó un dashboard en Power BI [48] para comparar el rendimiento de los modelo.

6. Resultados

En este capítulo, se presentará la homologación de los campos, el análisis descriptivo de cada conjunto de datos, la definición de los hiperparámetros y el espacio paramétrico de cada modelo, las tasas de balanceo establecidas y los resultados obtenidos en cada iteración.

6.1. Credit Approval

6.1.1. Conjunto de datos

En la Tabla **6-1**, se observan los campos que conforman el primer conjunto de datos y su correspondiente descripción. Este conjunto contiene información sociodemográfica y financiera de los clientes, la edad, el estado civil, el nivel educativo, la tenencia de vivienda, entre otros. Además, se incluyen detalles relacionados con los préstamos, como el saldo del cliente, la duración del crédito, entre otros.

Tabla 6-1.: Campos del conjunto de datos credit approval.

Campo	Descripción
age	Edad
job	Tipo de trabajo
marital	Estado civil
education	Nivel educativo
balance	Saldo
housing	Tenencia de casa
duration	Duración del préstamo
campaign	Campañas
approval	Estado de aprobación

6.1.2. Homologación de los campos

Realizando un pre procesamiento inicial en el primer conjunto de datos, se llevó acabo un re nombramiento con el fin de facilitar la manipulación y reducir los posibles errores relacionados con caracteres especiales. En la Tabla **6-2**, se observan los nombres originales y su nuevo nombre.

Tabla 6-2.: Renombramiento de los campos del conjunto de datos: credit approval.

Campo	Nuevo nombre
age	edad
job	trabajo
marital	estado_civil
education	nivel_educativo
balance	saldo
housing	ten_casa
duration	duracion
campaign	campanas

6.1.3. Análisis exploratorio

Después de realizar el proceso de re nombramiento de los campos, se llevaron a cabo algunas correcciones y transformaciones de los datos. En particular, se identificaron la presencia de valores negativos en ciertos campos, como la edad y el saldo. Para abordar este problema, se tomó la decisión de convertir todos los valores negativos en positivos. Adicionalmente, se decidió dejar aquellos clientes cuya edad superará los 18 años. Por último, se realizó una truncamiento de los valores extremos para la edad, el saldo y la duración del préstamo. Este proceso se llevó a cabo utilizando el percentil 0,99 de cada uno de estos.

En la Tabla **6-3**, se pueden apreciar que la edad promedio de los clientes es de alrededor de 41 años, con un rango que va desde 19 hasta 74 años. Los saldos de las cuentas bancarias muestran una variabilidad significativa, con un saldo promedio de aproximadamente 1.477 dolares y un saldo máximo de 71.188 dólares. El 57 % de los clientes son propietarios de una vivienda. La duración promedio de los préstamos es de aproximadamente 260 días, con una variación que se extiende desde 4 días hasta 1.275 días. En cuanto a la campaña de marketing, se observa que el promedio de contactos realizados es de 2,79, con un rango que va desde 1 hasta 50 contactos.

Tabla 6-3.: Estadísticas descriptivas de los campos numéricos del conjunto de datos: credit approval.

	Edad	Saldo	Tenencia de Casa	Duración	Campaña
Promedio	41	1.477	0,57	260	2,79
Desviación std	10	2.989	0,50	240	3
Mínimo	19	0	0	4	1
25º Percentil	33	132	0	104	1
Mediana	39	479	1	185	2
75º Percentil	49	1.490	1	329	3
Máximo	74	71.188	1	1.275	50

Por otra parte, en la Figura **6-1**, se observa que en el estado civil presenta tres valores, en donde, aproximadamente el 62 % (2.782) de los clientes son casados, el 27 % (1.191) de ellos son solteros y por último, el 12 % (526) de los clientes están divorciados.

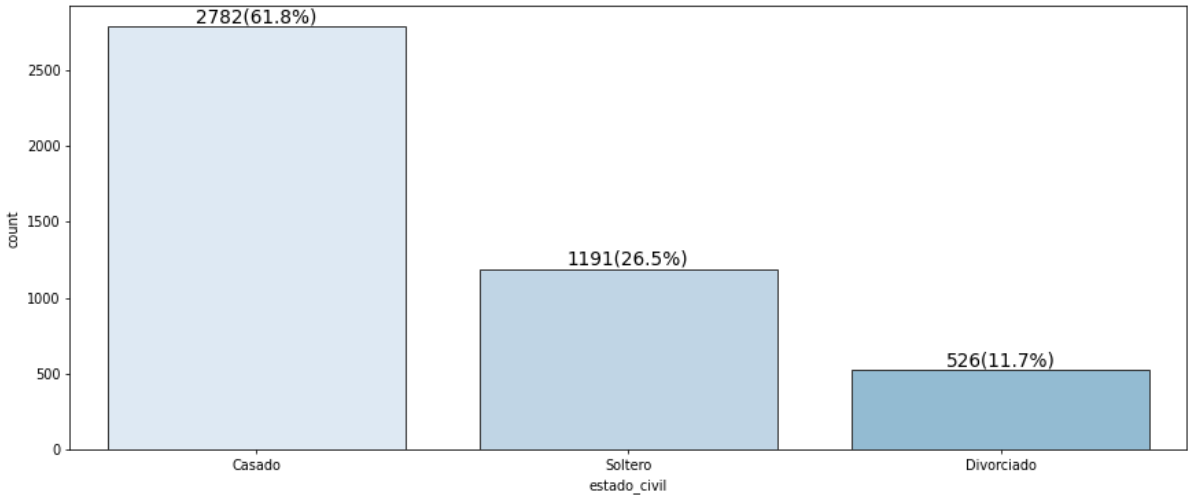


Figura 6-1.: Categorías del estado civil del conjunto de datos: credit approval.

De igual manera, en la Figura **6-2**, podemos apreciar que aproximadamente, el 81 % (3.638) de los clientes poseen un nivel educativo entre secundaria y pos secundaria, también hay un 15 % (676) de los clientes que solo terminaron la primaria.

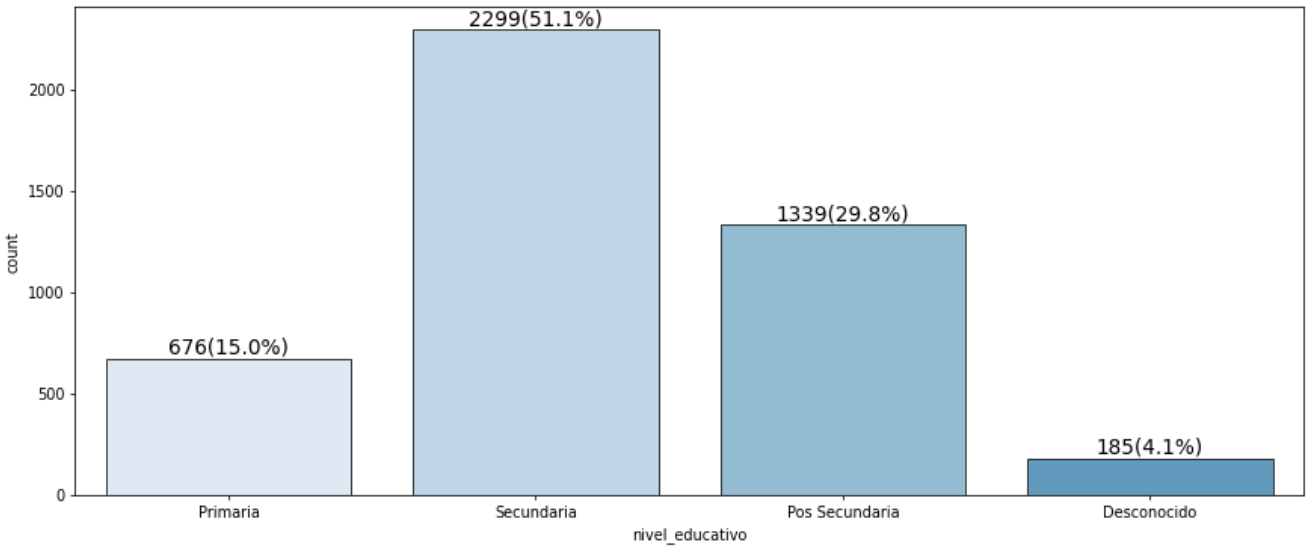


Figura 6-2.: Categorías del nivel educativo del conjunto de datos: credit approval.

Por último, en la Figura **6-3**, podemos observar las diferentes ocupaciones de los clientes.

Los cuatro trabajos con mayor presencia son el *manager*, con el 21.0 % de los clientes, el obrero con el 21.0 %, el técnico con el 17.0 % y el administrador con el 10.6 %. Sin embargo, también se puede apreciar la presencia de estudiantes, mucamas, pensionados, independientes y desempleados, entre otros.

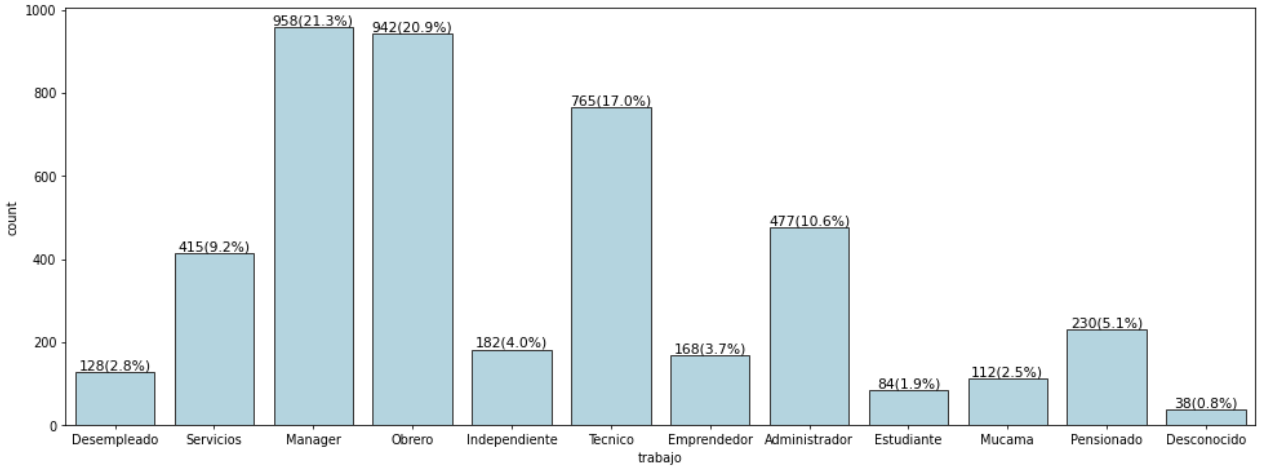


Figura 6-3.: Categorías de la ocupación del conjunto de datos: credit approval.

6.1.4. Preselección de variables

Después de aplicar los test de Mann-Whitney U y chi-cuadrado en los campos del conjunto de datos Credit Approval, en las Tablas **6-4** y **6-5** se pueden observar los estadísticos de ambos test y su correspondiente $valor - p$, en donde, el saldo, la duración, el trabajo, el estado civil, el nivel educativo y la tenencia de casa arrojaron un $valor - p$ menor que $0,05$ por lo cual, se sugiere que tiene distribuciones diferentes en relación con el otorgamiento de crédito.

Tabla 6-4.: Resultados del test mann-whitney u en los campos numéricos del conjunto de datos: credit approval.

Variable	Estadístico U	Valor p
Edad	99967950	0,2586
Saldo	88522400	<0,0001
Duración	3800915	<0,0001

Tabla 6-5.: Resultados del test chi-cuadrado en los campos categóricos del conjunto de datos: credit approval.

Variable	Estadístico χ^2	Valor p
Trabajo	68,79	<0,0001
Estado Civil	19,16	<0,0001
Nivel Educativo	15,68	0,0013
Tenencia de Casa	49,16	<0,0001

6.2. Credit Risk Analysis

6.2.1. Conjunto de datos

En la Tabla **6-6**, se observan los campos que conforman el segundo conjunto de datos y su descripción. Este conjunto también contiene información sociodemográfica y financiera de los clientes, como el ingresos, la edad y el estado de la vivienda. Además, se incluyen detalles relacionados con los préstamos, como tasas de interés del crédito, montos solicitados, marca de incumplimientos previas y el propósitos de los préstamos.

Tabla 6-6.: Campos del conjunto de datos: credit risk analysis.

Campo	Descripción
id	Identificación
age	Edad
income	Ingreso
home	Estado de la vivienda
emp_length	Antigüedad en el trabajo
intent	Propósito del préstamo
amount	Valor del préstamo
rate	Tasa de interés
status	Estado de aprobación
percent_income	Ingreso sobre el valor del préstamo
default	Marca de incumplimiento previa
cred_length	Historial crediticio

6.2.2. Homologación de los campos

De igual manera, para el segundo conjunto de datos, se realizó también un pre procesamiento inicial del segundo conjunto de datos, se realizó el re nombramiento con el fin de facilitar la manipulación y reducir los posibles errores relacionados con caracteres especiales. En la Tabla **6-7**, se pueden observar los nuevos nombres del conjunto de datos.

Tabla 6-7.: Renombramiento de los campos del conjunto de datos: credit risk analysis.

Campo	Nuevo nombre
id	identificacion
age	edad
income	ingreso
home	estado_vivienda
emp_length	antiguedad_empleo
intent	proposito_prestamo
amount	monto_prestamo
rate	tasa
status	target
percent_income	porc_prest_ingre
default	incumplimiento_previo
cred_length	historial_crediticio

6.2.3. Análisis exploratorio

Posterior al re nombramiento de los campos, se tomó la decisión de conservar únicamente los registros que contenían información completa, ya que la imputación de datos relacionados con los préstamos no sería apropiada, dado que cada préstamo se adapta a las circunstancias específicas de cada cliente. Además, se realizó una truncamiento de los valores extremos para la edad y antigüedad en el empleo, esto se hizo utilizando el percentil 0.99 de ambas variables. Por otro lado, en las Tablas **6-8** y **6-9**, se observa que la edad de los clientes varía entre 20 y 50 años, con una edad promedio de 27 años. En cuanto a los ingresos, estos abarcan desde los 4.000 hasta 6.000.000 dólares, con un ingreso promedio de alrededor de 66.649 dólares.

La antigüedad en el empleo promedio es de 4 años, y el monto promedio de los préstamos solicitados es de aproximadamente 9.656 dólares, con tasas de interés que oscilan entre 5,42 % y 23,22 %. Además, en promedio, los préstamos representan el 17 % de los ingresos de los clientes. Por último, alrededor del 18 % de los clientes han tenido algún incumplimiento previo.

Tabla 6-8.: Estadísticas descriptivas de los campos numéricos del conjunto de datos: credit risk analysis, parte 1.

	Edad	Ingresos	Antigüedad en Empleo	Monto del Préstamo
Promedio	27,65	66.649	4,76	9.656
Desviación std	5,89	62.356	3,93	6.329
Mínimo	20	4.000	0	500
25º Percentil	23	39.480	2	5.000
Mediana	26	55.956	4	8.000
75º Percentil	30	80.000	7	12.500
Máximo	50	6.000.000	18	35.000

Tabla 6-9.: Estadísticas descriptivas de los campos numéricos del conjunto de datos: credit risk analysis, parte 2.

	Tasa	% Préstamo/Ingreso	Incumplimientos	Historial Crediticio
Promedio	11,04	0,17	0,18	5,79
Desviación std	3,23	0,11	0,38	4,04
Mínimo	5,42	0	0	2
25º Percentil	7,90	0,09	0	3
Mediana	10,99	0,15	0	4
75º Percentil	13,48	0,23	0	8
Máximo	23,22	0,83	1	30

En la Figura **6-4**, podemos obervar como es el estado de la vivienda de los clientes, se aprecia que la mayoría de los clientes, aproximadamente el 51 %, vive en alquiler, un 41 % tiene su vivienda hipotecada y solamente un 7.7 %, posee vivienda propia.

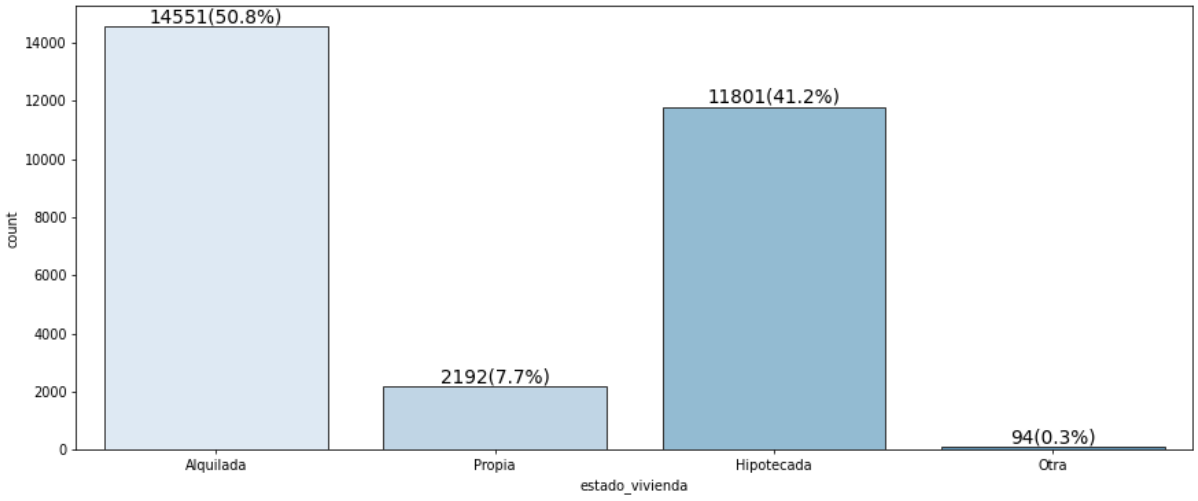


Figura 6-4.: Categorías del estado de la vivienda del conjunto de datos: credit risk analysis.

Por otra parte, en la Figura **6-5**, podemos apreciar el motivo o propósito por el cual el cliente está solicitando el crédito. Los clientes buscan el crédito para temas de educación, salud, emprendimiento, motivos personales y unificación de deudas, representando un 20 %, 18.5 %, 17.5 %, 17 % y 15.9 %, respectivamente.

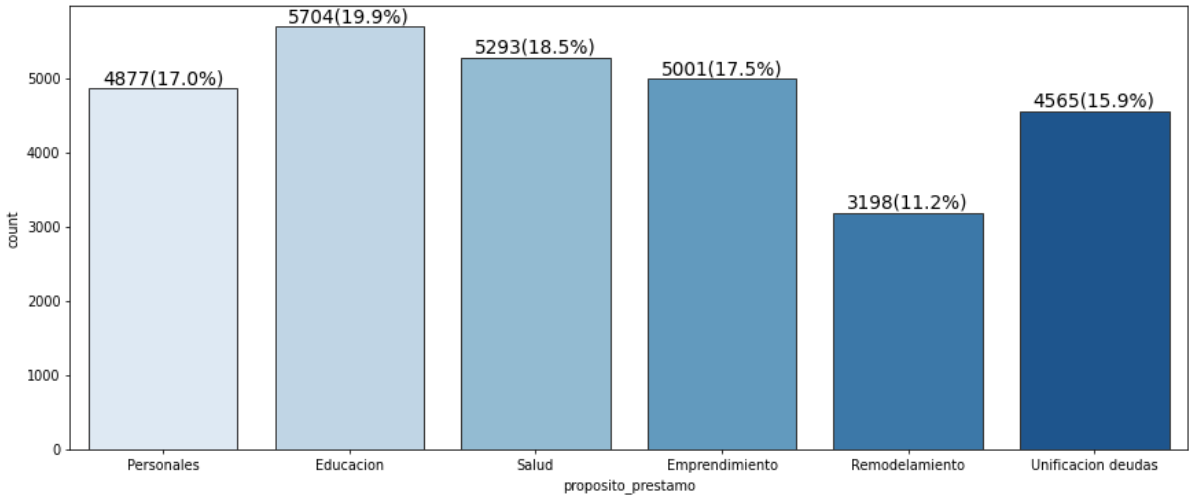


Figura 6-5.: Categorías del propósito del préstamo del conjunto de datos: credit risk analysis.

6.2.4. Preselección de variables

De igual manera, en las Tablas **6-10** y **6-11**, se pueden apreciar los resultados obtenidos de los test no paramétricos luego de aplicarlos en los campos del conjunto de datos Credit Risk Analysis, en donde, las nueve variables obtuvieron un $valor - p$ menor al 0,05, por lo cual, se sugiere que todas las variables poseen una distribución diferente en relación con el otorgamiento del crédito.

Tabla 6-10.: Resultados del test mann-whitney u en los campos numéricos del conjunto de datos: credit risk analysis.

Variable	Estadístico U	Valor p
Edad	7287069700	<0,0001
Ingresos	9542192100	<0,0001
Antigüedad en Empleo	7913714100	<0,0001
Monto del Préstamo	6052139200	<0,0001
Tasa de Interés	3794549450	<0,0001
Historial Crediticio	7189279400	<0,0001

Tabla 6-11.: Resultados del test chi-cuadrado en los campos categóricos del conjunto de datos: credit risk analysis.

Variable	Estadístico χ^2	Valor p
Estado de Vivienda	1652	<0,0001
Propósito del Préstamo	465	<0,0001
Incumplimiento Previo	947	<0,0001

6.3. Credit Risk Customers

6.3.1. Conjunto de datos

En la Tabla **6-12**, se observan los campos del último conjunto de datos. Este conjunto de datos también contienen información sociodemográfica y financiera de los clientes, como el ingresos, la edad, el estado civil, el nivel educativo, la tenencia de vivienda y propiedades, el número de personas a cargo, el tipo de trabajo, entre otros. Además, se incluyen detalles relacionados con los préstamos, como tasas de crédito, montos solicitados, la tasa asociada a la cuota del préstamo, cantidad de créditos existentes y propósitos de los préstamos.

Tabla 6-12.: Campos del conjunto de datos: credit risk customers.

Campo	Descripción
checking_status	Estado de la cuenta corriente
duration	Duración en meses
credit_history	Historial crediticio
purpose	Propósito del crédito
credit_amount	Valor del crédito
savings_status	Estado de la cuenta de ahorros
employment	Antigüedad laboral
installment_commitment	Tasa de la cuota
personal_status	Estado civil
other_parties	Otros deudores
residence_since	Residencia actual en años
property_magnitude	Propiedades
age	Edad
other_payment_plans	Otros planes de pago
housing	Tipo de vivienda
existing_credits	Numero de créditos
job	Tipo de trabajo
num_dependents	Número de personas a cargo
own_telephone	Teléfono
foreign_worker	Trabajador extranjero
class	Estado de aprobación

6.3.2. Homologación de los campos

De igual manera, como parte del pre procesamiento inicial del tercer conjunto de datos, se llevó a cabo el re nombramiento con el objetivo de simplificar la manipulación y disminuir posibles errores. En la Tabla **6-13**, se detallan los nuevos nombres del conjunto de datos.

Tabla 6-13.: Renombramiento de los campos del conjunto de datos credit risk customers.

Campo	Nuevo nombre
checking_status	estado_cuenta_cheques
duration	duracion_prestamo_meses
credit_history	historial_crediticio
purpose	proposito_credito
credit_amount	monto_credito
savings_status	estado_ca
employment	antiguedad_empleo
installment_commitment	tasa_cuota
personal_status	estado_civil
other_parties	otro_deudor
residence_since	antiguedad_residencia
property_magnitude	propiedades
age	edad
other_payment_plans	otros_metodos_pago
housing	tipo_vivienda
existing_credits	num_creditos
job	perfil_trabajo
num_dependents	num_personas_a_cargo
own_telephone	tenencia_telefono
foreign_worker	trabajador_extranjero
class	target

6.3.3. Análisis exploratorio

En la Tabla **6-14**, se observa que la duración de los prestamos de los clientes varían desde 4 hasta 72 meses, con una duración promedio de aproximadamente 20 meses. El monto del crédito también muestra variabilidad, con un promedio de alrededor de 3.271 marcos alemanes. La tasa de la cuota promedio es de 2,97, con tasas que van desde 1 hasta 4. Respecto a la antigüedad de residencia, en promedio, los clientes tienen aproximadamente 2,85 años en sus lugares de residencia. Por otra parte, en la Tabla **6-15**, se aprecia que la edad promedio de los clientes es de 35 años, con un rango amplio de edades desde 19 hasta 75 años. El 25 % de los clientes tiene un solo crédito, aunque el 75 % posee hasta cuatro créditos. En promedio

los clientes tienen 1 persona a cargo.

Tabla 6-14.: Estadísticas descriptivas de los campos numéricos del conjunto de datos: credit risk customers, parte 1.

	Duración Préstamo	Monto Crédito	Tasa Cuota	Antigüedad Residencia
Promedio	20,90	3.271	2,97	2,85
Desviación std	12,06	2.822	1,12	1,10
Mínimo	4	250	1	1
25º Percentil	12	1.365	2	2
Mediana	18	2.319	3	3
75º Percentil	24	3.972	4	4
Máximo	72	18.424	4	4

Tabla 6-15.: Estadísticas descriptivas de los campos numéricos del conjunto de datos: credit risk customers, parte 2.

	Edad	Num. Créditos	Num. Pers. a Cargo
Promedio	35,55	1,41	1,16
Desviación std	11,38	0,58	0,36
Mínimo	19	1	1
25º Percentil	27	1	1
Mediana	33	1	1
75º Percentil	42	2	1
Máximo	75	4	2

En la Tabla **6-16**, se destaca que la mayoría de los clientes no posee una cuenta de cheques. En cuanto al historial crediticio, más de la mitad de los clientes cuenta con un historial crediticio existente y pagado. Respecto al propósito de los créditos, aproximadamente el 28 % de las solicitudes están destinadas a la compra de radio o televisión. Por otra parte, se observa que la mayoría de los clientes tienen ingresos mensuales inferiores a 100 marcos alemanes y una antigüedad laboral de 1 a 4 años. Además, la mayoría de los clientes son hombres solteros. En lo que respecta a la tenencia de propiedades, se destaca que la propiedad más común entre los clientes es el carro, y en su mayoría, no utilizan otros métodos de pago. Por último, se puede apreciar que la vivienda propia es la preferida por la mayoría de los clientes. En cuanto al perfil de trabajo, la mayoría de los clientes se clasifican como calificados, y la gran mayoría no son trabajadores extranjeros.

Tabla 6-16.: Estadísticas descriptivas de los campos categóricos del conjunto de datos: credit risk customers.

Variable	Valores Únicos	Categoría principal	Frecuencia
Estado Cuenta Cheques	4	Sin cuenta	394
Historial Crediticio	5	Existente pagado	530
Propósito Crédito	9	Radio - Tv	280
Estado Ca	5	Menor 100	603
Antigüedad Trabajo	4	1 - 4	339
Estado Civil	3	Hombre soltero	548
Otro Deudor	3	Ninguno	907
Propiedades	4	Carro	332
Otros Métodos Pago	3	Ninguno	814
Tipo Vivienda	3	Propia	713
Perfil Trabajo	4	Calificado	630
Trabajador Extranjero	2	No	963

6.3.4. Preselección de variables

Por último, luego de aplicar los test al conjunto de datos datos Credit Risk Customers, con un nivel de significancia de 0,05, en la Tabla **6-17** y **6-11**, se identificaron las variables que resultaron ser significativas para predecir si se otorga o no un crédito a un cliente. Entre ellas esta la duración del préstamo, la edad, el monto del crédito, la tasa de la cuota, el estado de cuenta de cheques, el historial crediticio, el propósito del crédito, el estado civil, el estado de la cuenta de ahorros, la antigüedad en el trabajo, otro deudor, la tenencia de propiedades, el tipo de vivienda, y si es un trabajador extranjero.

Tabla 6-17.: Resultados del test mann-whitney u en los campos numéricos del conjunto de datos: credit risk customers.

Variable	Estadístico U	Valor p
Duración del Préstamo	7799550	<0,0001
Monto del Crédito	9348000	0,0059
Tasa de Cuota	9588950	0,0199
Antigüedad de Residencia	10468050	0,9358
Edad	11983300	0,0004
Número de Créditos	11027200	0,1348
Número de Personas a Cargo	10525000	0,9242

Tabla 6-18.: Resultados del test chi-cuadrado en los campos categóricos del conjunto de datos: credit risk customers.

Variable	Estadístico χ^2	Valor p
Estado de Cuenta Cheques	123,72	<0,0001
Historial Crediticio	61,69	<0,0001
Propósito del Crédito	31,07	<0,0001
Estado Civil	9,61	0,0222
Estado Actual del C.A.	36,1	<0,0001
Antigüedad en el Trabajo	18,37	0,001
Otro Deudor	6,65	0,0361
Propiedades	23,72	<0,0001
Tipo de Vivienda	18,2	<0,0001
Perfil de Trabajo	1,89	0,5966
Tenencia de Teléfono	1,17	0,2789
Trabajador Extranjero	5,82	0,0158

6.4. Tasas de otorgamiento de los conjuntos de datos

La Tabla **6-19** se presentan las tasas de otorgamiento correspondientes a cada uno de los conjuntos de datos. Cabe destacar que cada conjunto presenta su propia tasa de otorgamiento, detallando así las variaciones entre ellas.

Tabla 6-19.: Tasa de otorgamiento de los tres conjuntos de datos.

	Tasa de otorgamiento	
Conjuntos de datos	Se otorga	No se otorga
Credit Approval	12 %	88 %
Credit Risk Analysis	78 %	22 %
Credit Risk Customers	70 %	30 %

6.5. Definición espacio paramétrico

Para llevar acabo la de optimización de los modelos, se definieron conjuntos específicos de hiperparámetros para cada uno de los algoritmos. Para el algoritmo *XGBoost*, se exploraron y ajustaron hiperparámetros tales como la profundidad del árbol, el número de estimadores, la tasa de aprendizaje, el mínimo peso de la hoja, la submuestra, el valor gamma y la tasa de muestreo de columnas. En la Tabla **6-20**, se pueden apreciar los valores asociados a los parámetros.

Tabla 6-20.: Definición de los hiperparámetros para el algoritmo *XGBoost*.

Hiperparámetro	Valor Asignado
max_depth	Rango (1-200, cada 5)
n_estimators	Rango (10-500, cada 20)
learning_rate	Rango (0-0.55, cada 0.05)
min_child_weight	Rango (10-550, cada 20)
subsample	Rango (0-1, cada 0.025)
gamma	Rango (0-1, cada 0.05)
colsample_bytree	Rango (0.3-0.7, cada 0.05)

En el caso del *Random Forest*, en la Tabla **6-21**, se pueden observar los parámetros y el rango de valores utilizados. En este caso, se ajustaron parámetros como el número de estimadores, min. de muestras para división, min. de muestras por hoja, máx. características, profundidad máxima, bootstrap y criterio. Con respecto al Árbol de Decisión, se consideraron factores como el costo mínimo de la poda, la división mínima de muestras, la máxima cantidad de características y la profundidad máxima, los valores asociados a estos parámetros se pueden observar en la Tabla **6-22**.

Tabla 6-21.: Definición de los hiperparámetros para el algoritmo *Random Forest*.

Hiperparámetro	Valor Asignado
n_estimators	Rango (5-500, cada 20)
min_samples_split	[2, 4, 6, 8, 10, 15, 20, 30, 40]
min_samples_leaf	[10, 30, 60, 90, 100, 150, 200]
max_features	[sqrt, auto, 0.25, 0.3, 0.5, 0.75, 1.0]
max_depth	Rango (1-300, cada 5)
bootstrap	[True, False]
criterion	[gini, entropy]

Tabla 6-22.: Definición de los hiperparámetros para el algoritmo Árbol de Decisión.

Hiperparámetro	Valor Asignado
ccp_alpha	Rango (0-1, cada 0.02)
min_samples_split	[10, 15, 20, 30, 40]
max_features	[sqrt, auto, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.75, 1.0]
max_depth	Rango (1-300, cada 5)

Por último, en la Regresión Logística, se tuvo en cuenta el tipo de penalización, el parámetro de regularización y el tipo de solución, como se detalla en la Tabla **6-23**. Estas conFigura-

ciones se utilizaron en la búsqueda de hiperparámetros a través de validación cruzada y la búsqueda aleatoria.

Tabla 6-23.: Definición de los hiperparámetros para la Regresión Logística.

Hiperparámetro	Valor Asignado
penalty	[l1, l2]
C	[0.001, 0.01, 0.1, 1]
tol	[0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1]
solver	[liblinear]

6.6. Definición tasas de balanceo

Dado que cada uno de los conjuntos de datos presentaba su propio nivel de desequilibrio, se utilizaron tasas de balanceo específicas para cada una de las bases. En la Tabla **6-24**, se pueden observar las tasas definidas para cada base de datos.

Tabla 6-24.: Definición de las tasas de balanceo a utilizar para cada conjunto de datos.

Base de Datos	Tasas de Balanceo
Credit Approval	Sin balanceo, 0.2, 0.25, 0.3, 0.35, ..., 0.8, 0.85, 0.9
Credit Risk Analysis	Sin balanceo, 0.29, 0.3, 0.35, ..., 0.8, 0.85, 0.9
Credit Risk Customers	Sin balanceo, 0.45, 0.5, 0.55, ..., 0.8, 0.85, 0.9

6.7. Modelación

En el Anexo A se puede observar la función *balance_and_train_multiple_models*, una herramienta diseñada para manejar el desequilibrio en los datos y aplicar múltiples modelos de machine learning. En términos generales, la función comienza dividiendo la base de datos original en conjuntos de *train* y *test*. Luego, aborda el desbalanceo utilizando la técnica seleccionada. A continuación, se definen y configuran los modelos, se seleccionan las variables más importantes, se entrena cada modelo con estas variables, se ajustan sus parámetros, y finalmente, se evalúa el rendimiento de cada modelo.

6.7.1. Credit Approval

Al aplicar la función al conjunto de datos credit approval sin realizar ningún tipo de balanceo, en la Tabla **6-25**, podemos observamos que los modelos muestran variaciones significativas en sus métricas de rendimiento. El modelo *XGBoost* destaca por su capacidad para identificar correctamente a aquellos a quienes se les debería otorgar el crédito, como se refleja en su alto *recall* del 0.82, aunque con un *precision* más bajo (0.29). Con este equilibrio, se obtuvo un *F1 Score* del 0.43 y un AUC del 0.85, indicando un rendimiento general sólido. Por otro lado, el *Random Forest* exhibe el mejor *precision* (0.38) entre los modelos, junto con un *F1 Score* del 0.47 y un AUC de 0.84. El Árbol de Decisiones muestra un rendimiento moderado, con un AUC del 0.70, el cual sugiere un rendimiento aceptable. Finalmente, la Regresión Logística presenta un rendimiento equilibrado, con un *precision* del 0.35, un *recall* del 0.74, y un *F1 Score* de 0.47. Su AUC de 0.85 es comparable al de *XGBoost*, indicando un rendimiento sólido en la clasificación.

Tabla 6-25.: Resultados de modelos sin balanceo en el base del test del conjunto de datos credit approval.

Model	Accuracy	precision	recall	F1 Score	AUC
XGBoost	0.72	0.29	0.82	0.43	0.85
Random Forest	0.82	0.38	0.62	0.47	0.84
Árbol de Decisión	0.77	0.28	0.53	0.37	0.70
Regresión Logística	0.79	0.35	0.74	0.47	0.85

Logistic Regression

Luego de utilizar las técnicas de balanceo con las diferentes tasas de balanceo y la Regresión Logística, en la Figura **6-6**, podemos observar que a medida que aumenta la tasa de balanceo, el *accuracy* del modelo se encuentra por encima del 0.75 con los tres métodos del balanceo, sin embargo, a partir de la tasa de balanceo del 70 %, el *accuracy* empieza a caer con el SMOTE, en comparación a los otros dos métodos continuaron estables.

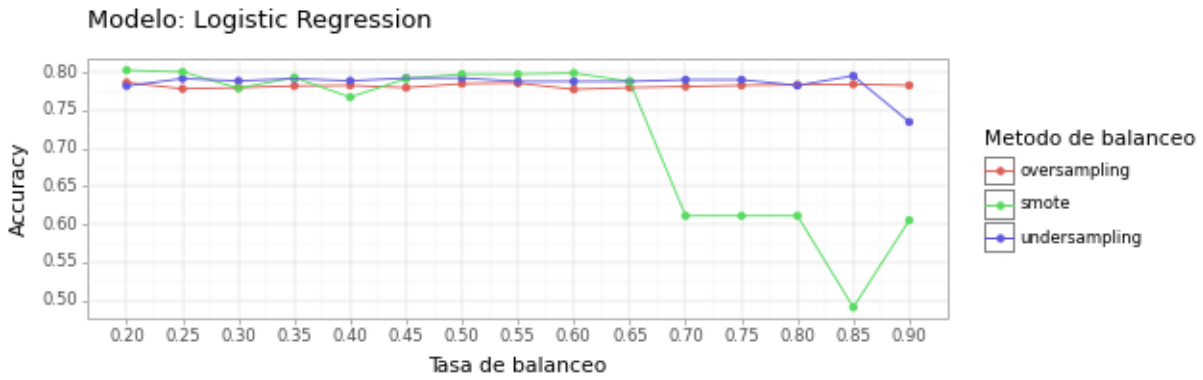


Figura 6-6.: Métrica *accuracy* de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

De igual manera, al observar la Figura **6-7**, podemos observar que el AUC sigue el mismo comportamiento del *accuracy*, sin embargo, con una tasa de balanceo del 60 %, el AUC con los tres métodos de balanceo es de aproximadamente del 0.85, sin embargo, por encima del 65 %, el AUC con el SMOTE decae nuevamente, entre el 0.55 y el 0.6.

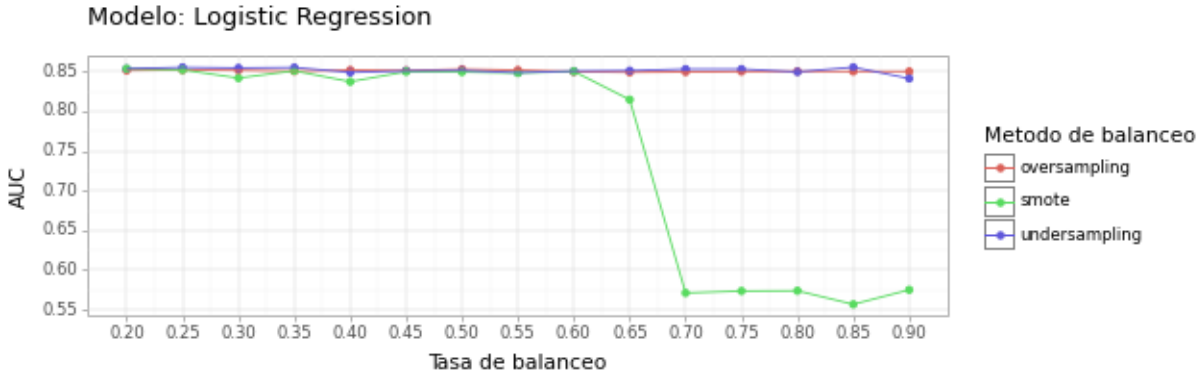


Figura 6-7.: Métrica AUC de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Por otra parte, en la Figura **6-8**, podemos apreciar que igual que las métricas anteriores, el *F1 Score* se comporta de la misma manera, en donde, hasta una tasa de balanceo del

65 %, en donde el *F1 Score* esta entre el 0.43 y el 0.53, sin embargo, a partir de una tasa de balanceo del 70 %, la métrica decae nuevamente con el SMOTE.

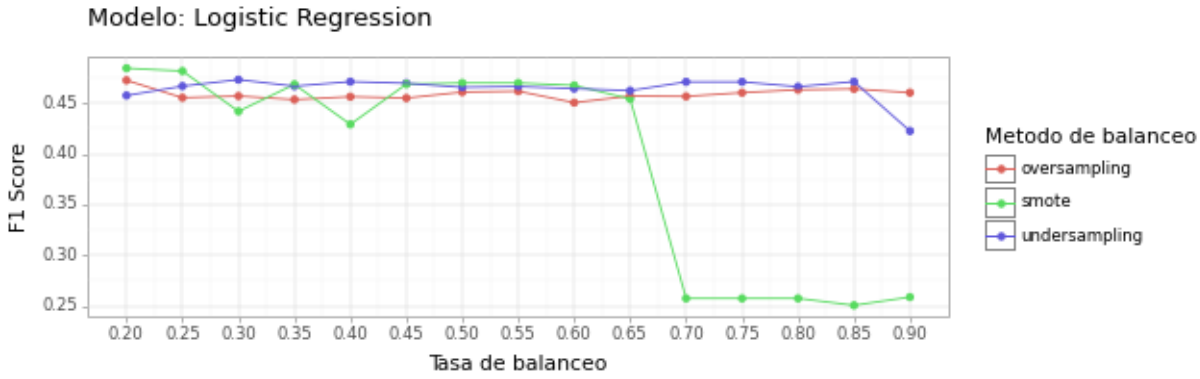


Figura 6-8.: Métrica *F1 Score* de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Árbol de Desición

Con la aplicación del Árbol de Decisiones con las diferentes tasas de balanceo, en la Figura **6-9**, podemos observar como el *accuracy* con el *undersampling* y unas tasas de balanceo del 30 % y del 35 %, se presentó una caída aproximadamente del 0.75 al 0.13. Sin embargo, a partir de una tasa del 40 %, el *accuracy* se mantiene estable con los tres métodos, aunque, el *oversampling* y el SMOTE presentan un mejor desempeño, por encima del 0.75.

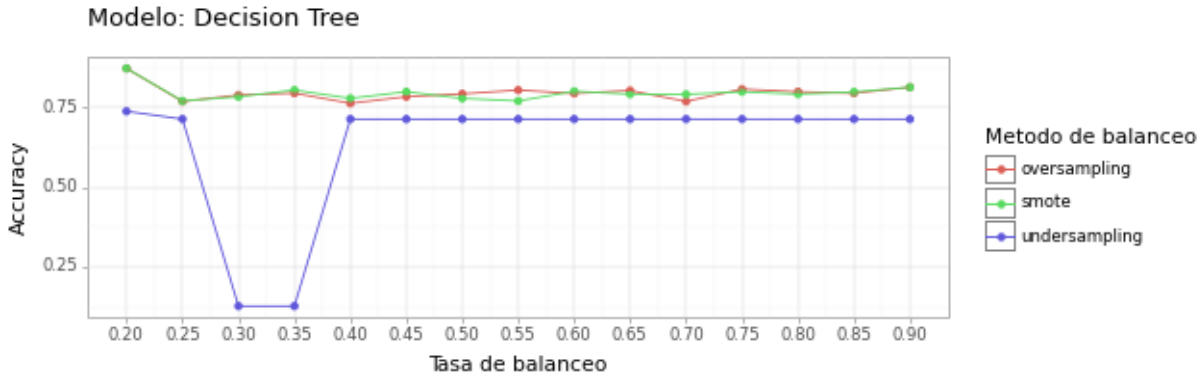


Figura 6-9.: Métrica *accuracy* del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Por otra parte, en la gráfica **6-10**, podemos observar como se presenta una variabilidad en el AUC dependiendo de la tasa de balanceo y del método de balanceo, en donde, con una tasa de balanceo del 20 %, el *oversampling* y *undersampling* presentaron un AUC del 0.5,

por otro lado, con tasas del 30 % y 35 %, el *undersampling* también presentó un AUC del 0.5, mientras que con los otros dos métodos, el AUC esta entre el 0.67 y el 0.75. A partir de una tasa de balanceo del 40 %, el *undersampling* presenta un desempeño estable, con un AUC entre el 0.7 y 0.75, seguido del SMOTE y por último el *oversampling*.

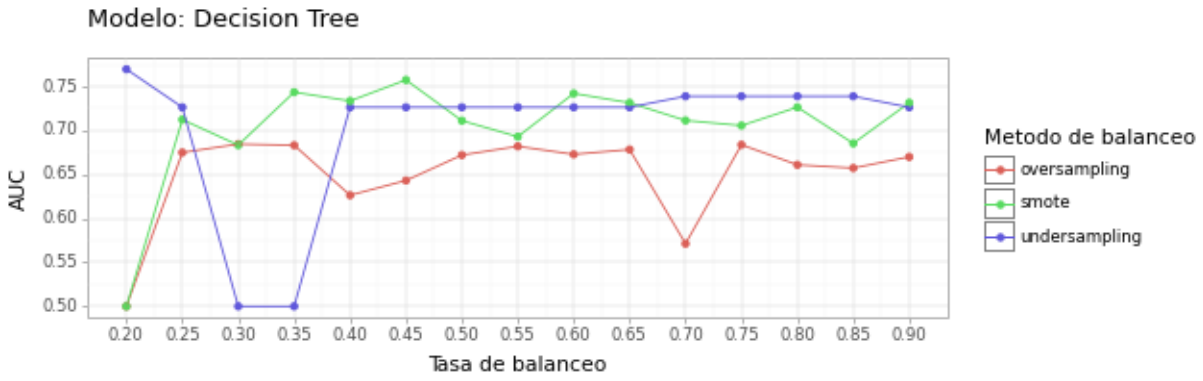


Figura 6-10.: Métrica AUC del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Por último, en la Figura **6-11** podemos observar que con una tasa de balanceo del 20 %, el *oversampling* y el SMOTE presentaron el peor desempeño, sin embargo, a partir de esta tasa de balanceo se presenta un mejor desempeño, en donde el *F1 Score* esta alrededor del 0.25 y 0.4. El *undersampling*, presenta el mejor desempeño, a partir de una tasa de balanceo del 40 %, con un *F1 Score* del 0.4.

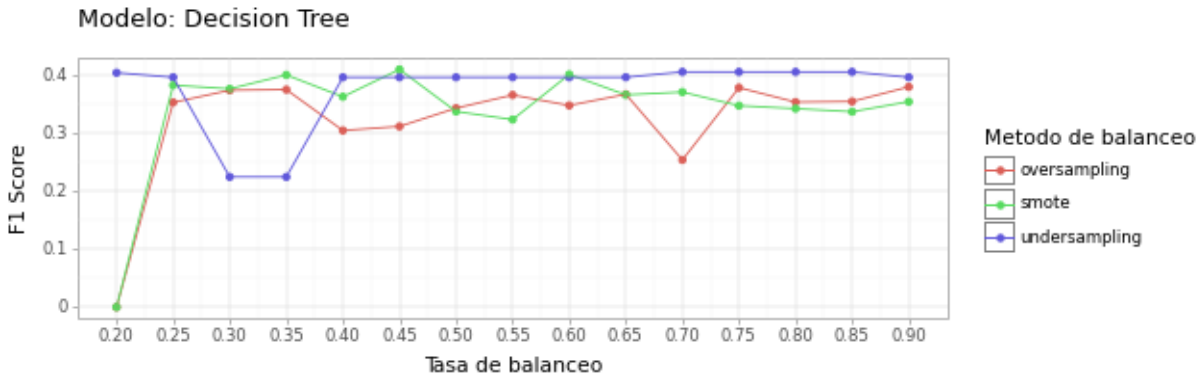


Figura 6-11.: Métrica *F1 Score* del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Random Forest

En la Figura **6-12**, podemos observar el resultado del *accuracy* de los modelos con las diferentes tasas de balanceo y los métodos, en donde, podemos apreciar que a medida que se

aumenta la tasa de balanceo, el *accuracy* empieza a decaer cuando se utiliza el *undersampling*. Sin embargo, con el *oversampling* y el SMOTE, el *accuracy* se mantiene estable, además, a medida que va aumenta la tasa de balanceo, el *accuracy* crece poco a poco y se mantiene encima del 0.82.

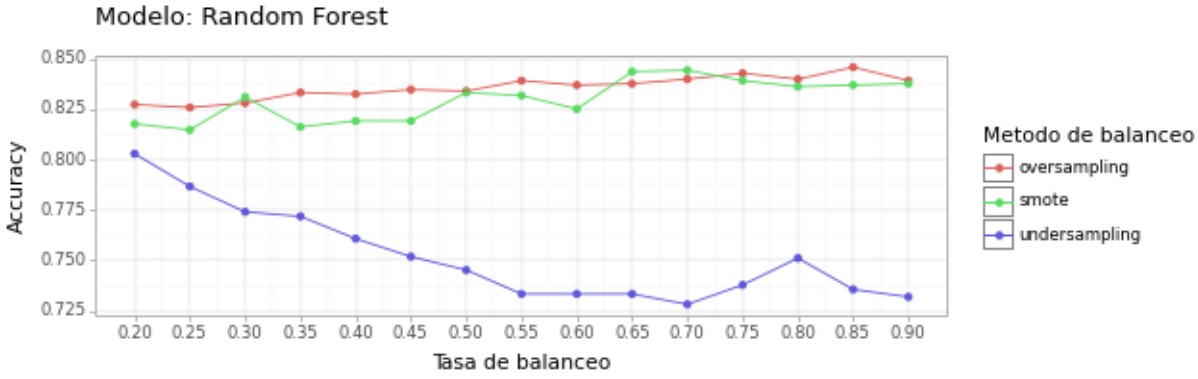


Figura 6-12.: Métrica *accuracy* del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Con respecto al AUC, en la Figura **6-13** podemos observar que con el SMOTE, se tiene una mayor variabilidad en la métrica a medida que se aumenta la tasa de balanceo, en donde, se va del 0.8 hasta el 0.85. Por otro lado, el *oversampling* el *undersampling* presentan una mayor estabilidad a partir de una tasa de balanceo del 40 %, en donde el AUC se mantiene entre el 0.82 y 0.83.

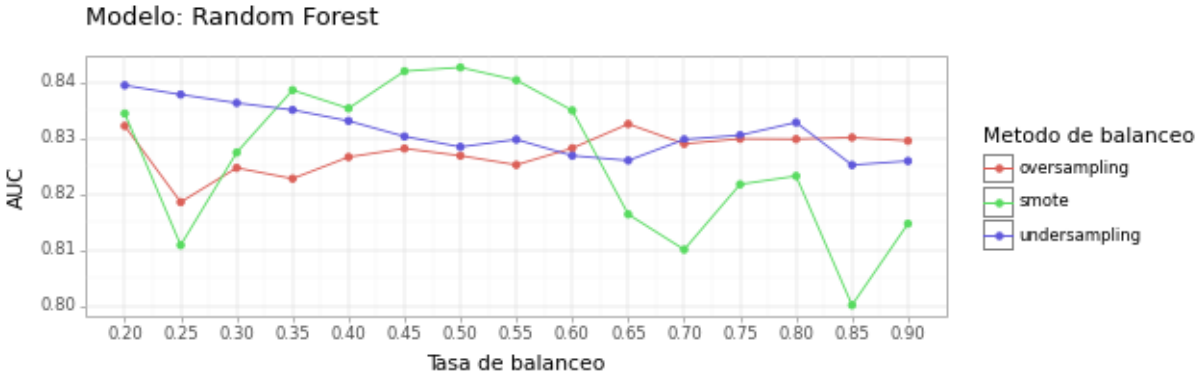


Figura 6-13.: Métrica AUC del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Por último, el *F1 Score* mostró un variabilidad similar al AUC, este variando entre el 0.42 y 0.48. Se observa que el *undersampling* y tasas de balanceo superiores al 20 % resultaron en un *F1 Score* inferior a los otros métodos de balanceo. Por otra parte, el *oversampling* y

el SMOTE, con tasas de balanceo entre el 30 % y el 80 %, presentaron un comportamiento estable y muy similar entre sí.

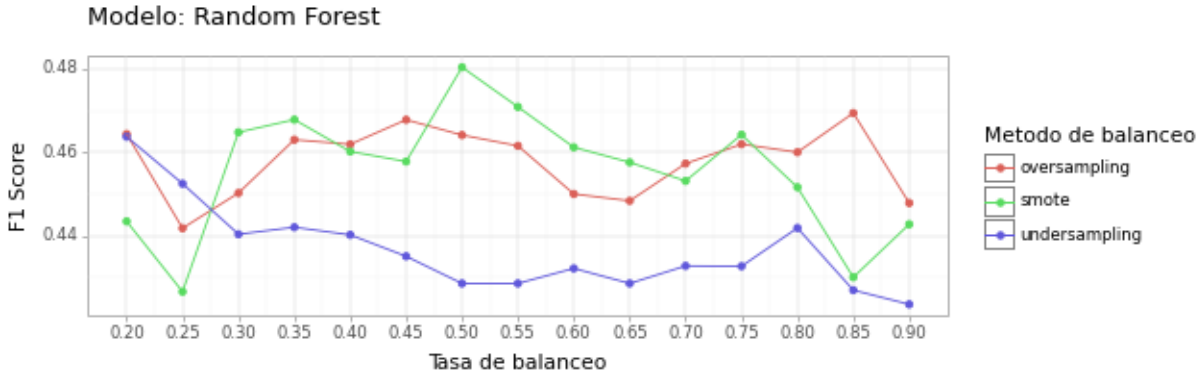


Figura 6-14.: Métrica *F1 Score* del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

XGBoost

Con la aplicación del *XGBoost* con las diferentes tasas de balanceo, en la Figura **6-15**, podemos observar los resultados obtenidos en cuanto al *accuracy*. Se observa que, de manera general, tanto el *oversampling* como el *undersampling* tienden a mantener un *accuracy* estable, aproximadamente entre 0.75 y 0.82. Este comportamiento es muy similar al encontrado en la aplicación del Árbol de Decisiones y el *Random Forest*. Por otro lado, el *undersampling* produce resultados inferiores, y a medida que se aumenta la tasa de balanceo, el *accuracy* tiende a disminuir.

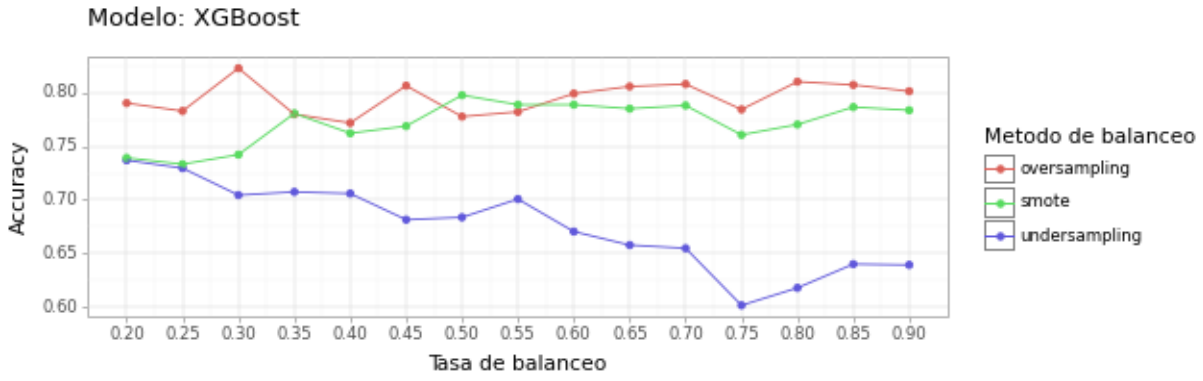


Figura 6-15.: Métrica *accuracy* del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

En cuanto al AUC, en la Figura **6-16**, podemos observar que tanto el *undersampling* como el SMOTE, con unas tasas de balanceo entre el 30 % y el 80 %, el AUC presenta una pequeña

variación, en donde se encuentra entre el 0.82 y el 0.84. En este caso, el *oversampling* presenta un desempeño menor, sin embargo, con los tres métodos de balanceo, el AUC se encuentra por encima del 0.7, indicando un buen desempeño de los modelos.

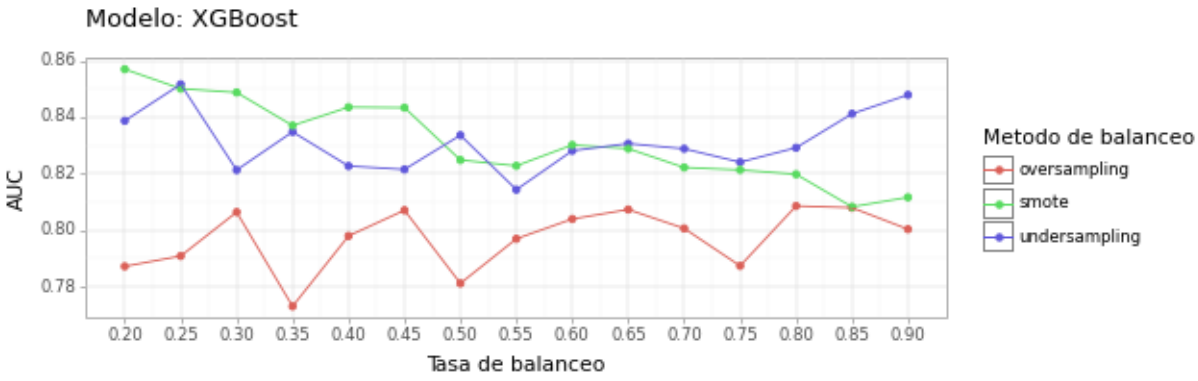


Figura 6-16.: Métrica AUC del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Por último, en la Figura **6-17**, se puede apreciar los resultados obtenidos del *F1 Score* con las diferentes tasas y métodos de balanceo. Se observa que el SMOTE, con una tasa de balanceo entre el 30 % y el 70 %, muestra un mejor desempeño que el *oversampling* y el *undersampling*. Sin embargo, al aumentar la tasa de balanceo entre el 75 % y el 85 %, el *F1 Score* tiende a disminuir. Por otro lado, el *undersampling* con tasas de balanceo bajas presenta un desempeño aceptable, pero a medida que se aumenta la tasa, el *F1 Score* comienza a decaer. De manera similar, el *oversampling* con tasas bajas de balanceo presenta más variabilidad en la métrica, y a partir de una tasa de balanceo del 50 %, el *F1 Score* muestra una tendencia ascendente.

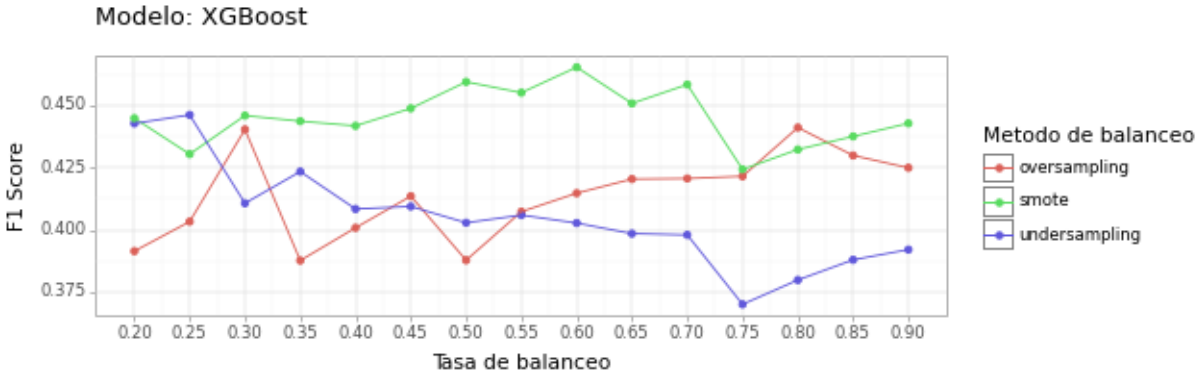


Figura 6-17.: Métrica *F1 Score* del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit approval.

Según los resultados obtenidos, se observó que el desempeño de los modelos varía según el método de balanceo y tasa de balanceo. En general, los modelos *XGBoost*, *Random Forest* y la Regresión Logística presentaron un buen rendimiento en términos del *accuracy*, *F1 Score* y AUC. Sin embargo, la elección del método de balanceo y la tasa específica influye en los resultados. Para el *XGBoost*, la técnica SMOTE con una tasa de balanceo entre el 25 % y el 70 %, mostró ser efectivo, obteniendo altos valores en las métricas. En cuanto al *Random Forest*, el balanceo mediante el *oversampling* con tasas por encima del 25 %, también arrojó muy buenos resultados. También, cabe resaltar que, la Regresión Logística con el *oversampling* y *undersampling* con las diferentes tasas de balanceo, fue el modelo que presentó la mejor estabilidad en las métricas. Por último, el Árbol de Decisiones con tasas de balanceo superiores al 40 % mostró estabilidad, sin embargo, presentó menores métricas que el resto de modelos.

6.7.2. Credit Risk Analysis

En la Tabla **6-26**, se presentan los resultados de los modelos aplicados al conjunto de datos credit risk analysis sin la implementación de ningún tipo de balanceo. En donde, el modelo *Random Forest* lidera con un alto *accuracy* (0.85), seguido de cerca por *XGBoost* (0.81), superando ambos al Árbol de Decisiones y a la Regresión Logística. En cuanto al *precision*, *Random Forest* y *XGBoost* lideran nuevamente, mientras que la Regresión Logística exhibe el *precision* más bajo. En relación con el *recall*, se observa que la Regresión Logística presenta el valor más alto, seguido por *Random Forest* y *XGBoost*. Además, el *Random Forest* muestra un *F1 Score* muy aceptable. Con lo cual, el *Random Forest* presenta un rendimiento global sólido y superior al resto de los modelos.

Tabla 6-26.: Resultados de modelos sin balanceo en el base del test del conjunto de datos credit risk analysis.

Model	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	0.81	0.55	0.75	0.63	0.87
Random Forest	0.85	0.63	0.75	0.69	0.89
Árbol de Desición	0.81	0.55	0.71	0.62	0.79
Logistic Regression	0.77	0.48	0.78	0.60	0.85

Logistic Regression

Al aplicar el modelo de Regresión Logística con distintas tasas y métodos de balanceo, en la Figura **6-18**, podemos observar los diferentes valores del *accuracy* obtenidos. Se puede apreciar que, en general, con tasas de balanceo del 30 % y 35 %, los tres métodos proporcionaron un *accuracy* superior a 0.76. Sin embargo, con tasas de balanceo del 40 % y el 55 % ,se registra una disminución en la métrica cuando se utiliza la técnica de SMOTE. A medida

que la tasa de balanceo aumenta, esta métrica también lo hace, mientras que con los otros dos métodos, el *accuracy* es mayor.

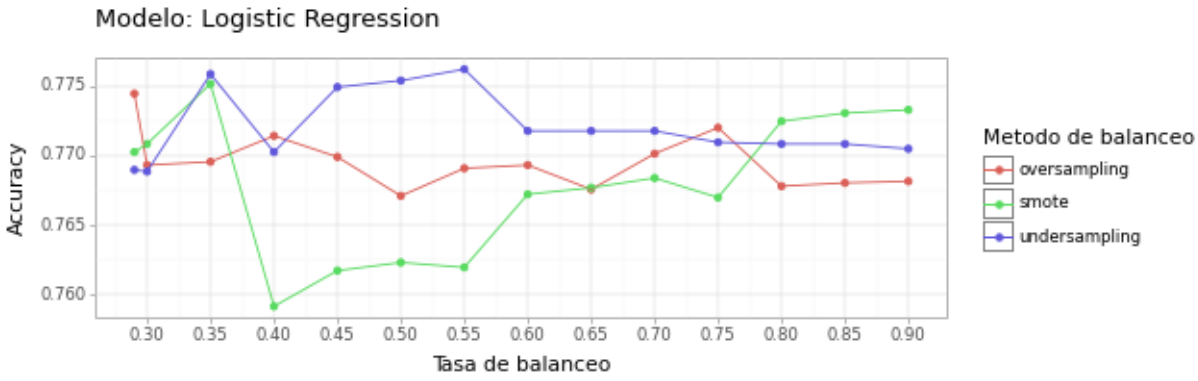


Figura 6-18.: Métrica *accuracy* de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Por otra parte, en la Figura **6-19**, se observan los valores del AUC obtenidos. Con tasas de balanceo del 30 % y 35 %, los tres métodos de balanceo presentan un buen desempeño, con un AUC superior a 0.85. Sin embargo, a partir de una tasa del 40 %, la técnica de SMOTE experimenta una disminución en el AUC, manteniéndose alrededor del 0.82 y 0.83. En cuanto al *oversampling* y *undersampling*, se observa un AUC constante y superior a 0.85 para todas las tasas de balanceo.

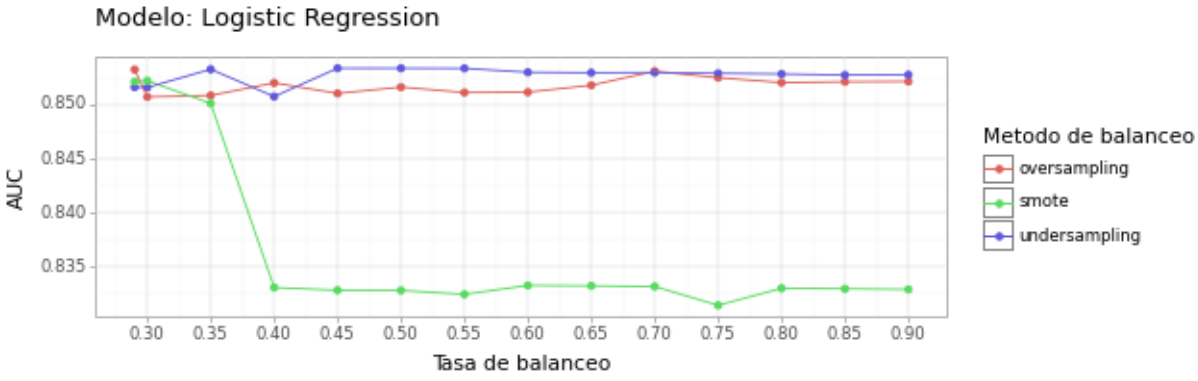


Figura 6-19.: Métrica AUC de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Por último, en la Figura **6-20**, se muestran los valores obtenidos del *F1 Score*. Podemos observar que se presenta el mismo comportamiento al del AUC, donde, con una tasa de balanceo del 30 % y 35 %, los tres métodos de balanceo muestran un buen desempeño. Sin embargo, a partir de una tasa de balanceo del 40 %, la técnica del SMOTE presenta una

disminución en el *F1 Score*. Por otro lado, tanto el *oversampling* como el *undersampling* mantienen un *F1 Score* constante y superior a 0.59.

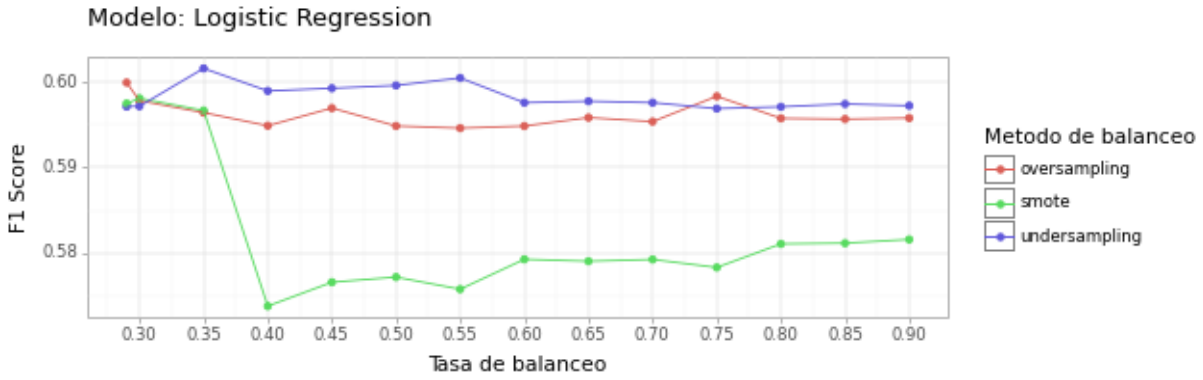


Figura 6-20.: Métrica *F1 Score* de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Árbol de Desición

Con la aplicación del Árbol de Decisiones con las diferentes tasas y métodos de balanceo, en la Figura **6-21**, se observa que el *undersampling* no presentó ningún tipo de variación, independientemente de la tasa de balanceo utilizada, con un valor constante del 0.81. por otra parte, el *oversampling* muestro una variación intermitente con diferentes tasas de balanceo. Sin embargo, el SMOTE mantiene un comportamiento constante con tasas de balanceo inferiores al 50 %, pero a partir de esta tasa, se observa una caída en la métrica seguida de variaciones.

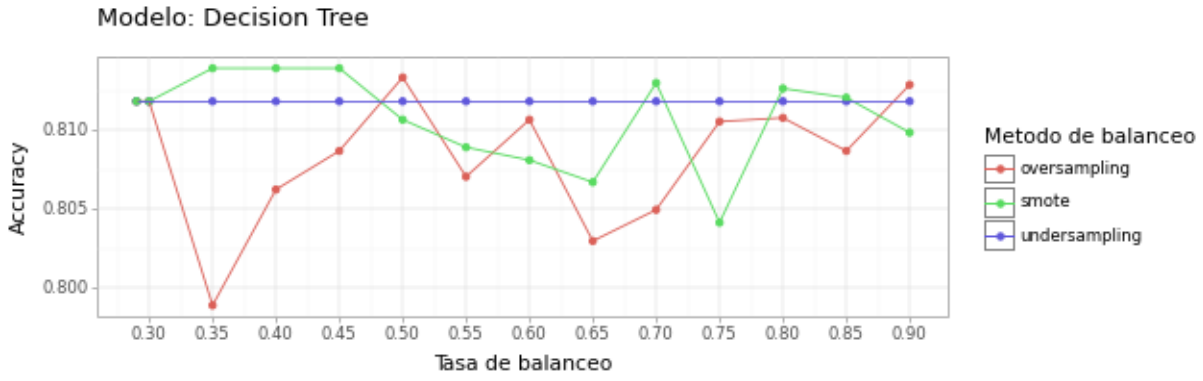


Figura 6-21.: Métrica *accuracy* del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

En la Figura **6-22**, se puede observar que con el *undersampling* se presentó un caso similar al del *accuracy*, manteniéndose constante independientemente de la tasa de balanceo aplicada,

con un AUC de 0.79. En cuanto al SMOTE, se obtuvo un AUC de 0.79 con tasas de balanceo entre el 30 % y el 45 %; sin embargo, con tasas de balanceo iguales o superiores al 50 %, los resultados del AUC fueron superiores a 0.81. Por último, el *oversampling* presentó un AUC cercano a 0.82 con tasas de balanceo entre el 35 % y el 60 %, pero con tasas más altas, se observa una disminución en la métrica.

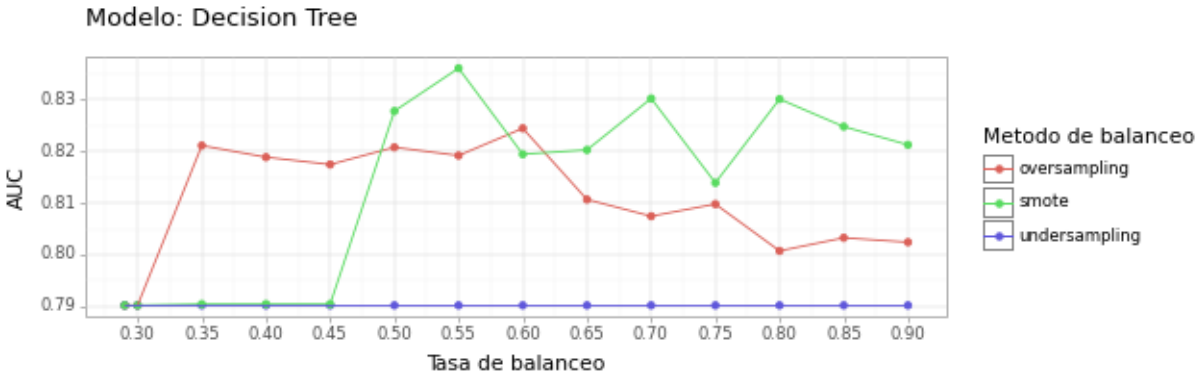


Figura 6-22.: Métrica AUC del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Por último, en la Figura **6-23**, se observan los resultados obtenidos del *F1 Score*. En donde, el *undersampling* exhibió un comportamiento constante similar a las dos métricas anteriores, con un valor constante de 0.625. Por otro lado, el SMOTE mostró valores altos en la métrica con tasas de balanceo entre el 30 % y el 55 %, sin embargo, con tasas superiores, se presentó una caída en la métrica. En contraste, el *oversampling* presentó valores por encima de 0.61 con tasas de balanceo inferiores al 65 %, pero con tasas más altas, los valores obtenidos fueron menores con excepción cuando se utilizo una tasa de balanceo del 75 %.

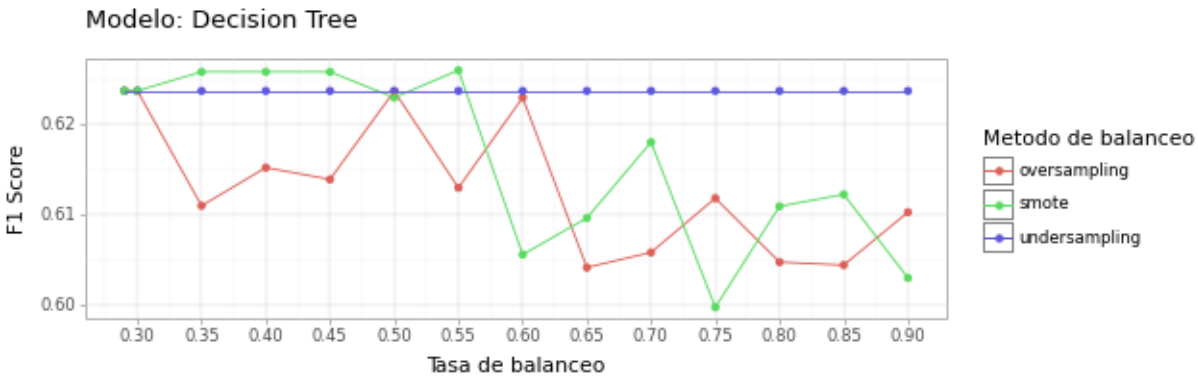


Figura 6-23.: Métrica *F1 Score* del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Random Forest

En la Figura **6-24**, podemos observar lo resultados obtenidos del *accuracy* luego de la aplicación del *Random Forest* con las diferentes tasas y métodos de balanceo. Podemos apreciar que, con una tasa de balanceo del 30 %, los tres métodos de balanceo presentaron un comportamiento similar. Sin embargo, a medida que aumenta la tasa de balanceo, el *undersampling* muestra una disminución en el *accuracy*. Por otro lado, tanto el *oversampling* como el SMOTE muestran un crecimiento constante en el *accuracy* a medida que aumenta la tasa de balanceo.

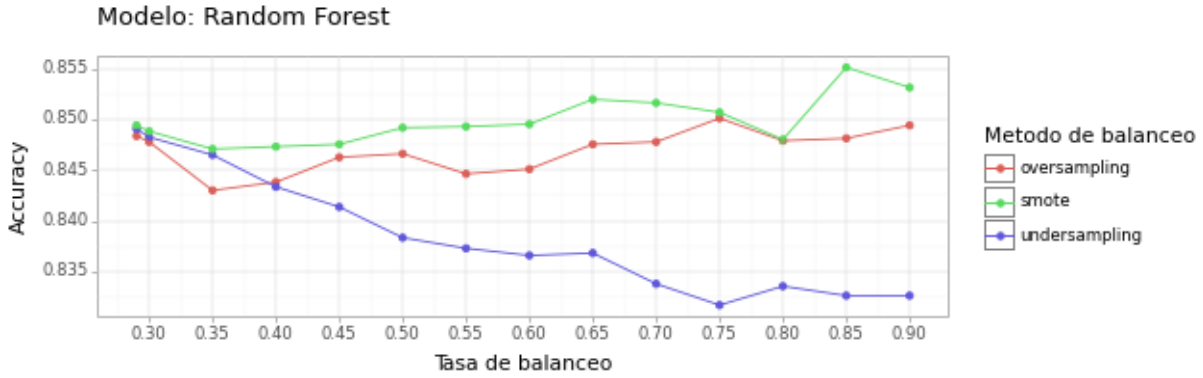


Figura 6-24.: Métrica *accuracy* del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Por otra parte, en la Figura **6-25**, podemos apreciar los resultados obtenidos del AUC. En donde, con una tasa de balanceo del 30 %, los tres métodos de balanceo presentaron un comportamiento similar, lo cual también se refleja en el *accuracy*. Sin embargo, al aumentar las tasas de balanceo, tanto el *oversampling* como el SMOTE mostraron una disminución en la métrica. Por otra parte, el *oversampling* experimenta una caída mucho menor y se mantiene por encima de un AUC del 0.885, excepto cuando se aplica una tasa de balanceo del 75 %, donde el AUC desciende hasta 0.882.

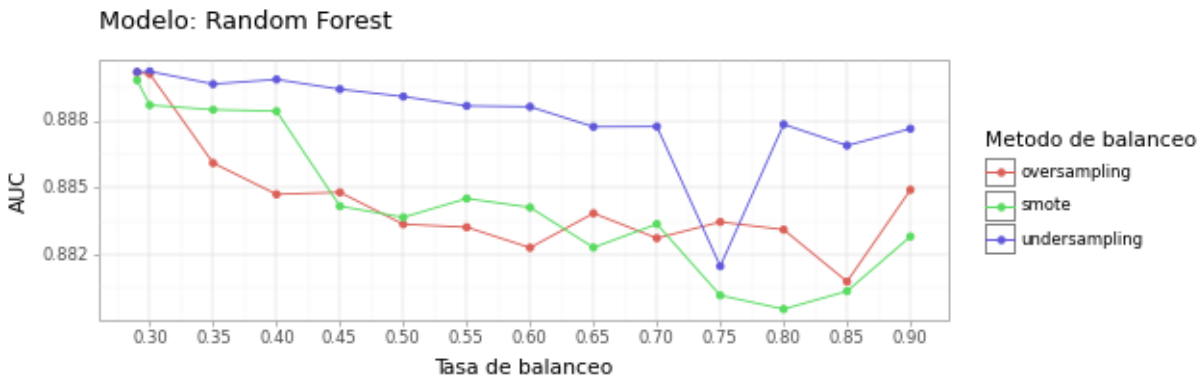


Figura 6-25.: Métrica AUC del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Por último, en la Figura **6-26**, podemos apreciar los resultados obtenidos del *F1 Score*. Podemos observar que, con tasas de balanceo entre el 30 % y el 40 %, los tres métodos de balanceo presentaron un comportamiento similar. Sin embargo, al aumentar las tasas de balanceo, tanto el *undersampling* como el SMOTE mostraron una disminución, aunque, el SMOTE se recupera nuevamente con tasas de balanceo del 85 % y 90 %. Por otro lado, el *oversampling* presenta el mejor desempeño, a medida que se aumenta la tasa de balanceo, el *F1 Score* se mantiene constante y muestra un crecimiento gradual.

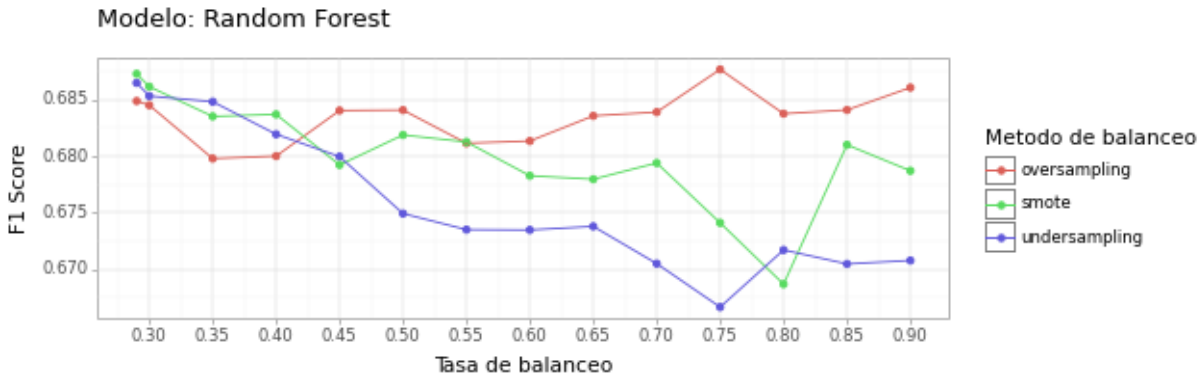


Figura 6-26.: Métrica *F1 Score* del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

XGBoost

Con la aplicación de *XGBoost* con las diferentes tasas y métodos de balanceo, en la Figura **6-27**, podemos observar los resultados obtenidos del *accuracy*. Se puede apreciar que el SMOTE presenta un aumento leve con tasas de balanceo inferiores o iguales al 50 %, sin embargo, con tasas por encima de este punto, el *accuracy* se mantiene constante alrededor

del 0.85. Por otra parte, el *oversampling* muestra una caída en el *accuracy* hasta alcanzar una tasa de balanceo del 75 %, donde aumenta y se mantiene constante. Del mismo modo, el *undersampling* presenta una caída a medida que se aumentan las tasas de balanceo, pero con una tasa del 90 %, el *accuracy* vuelve a aumentar.

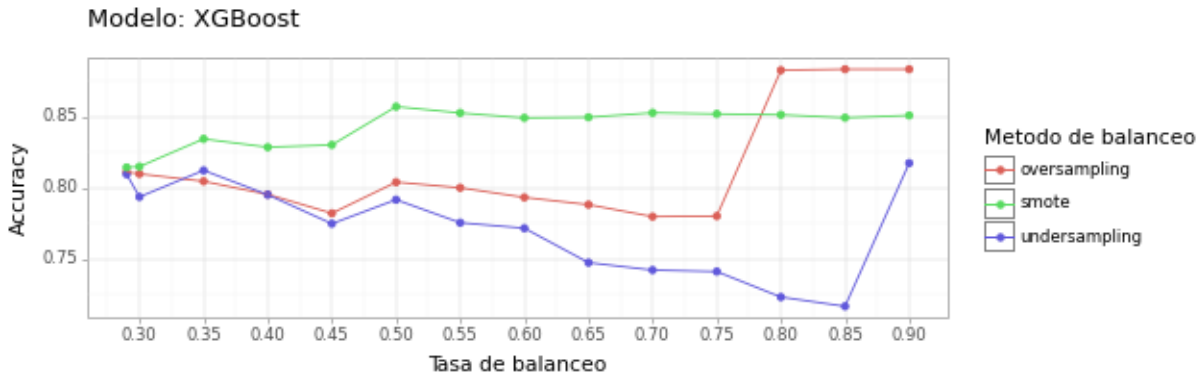


Figura 6-27.: Métrica *accuracy* del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

En la Figura **6-28**, podemos observar que el SMOTE, independientemente de la tasa de balanceo aplicada, muestra un comportamiento constante en el AUC, con un valor aproximado de 0.87. Por otro lado, el *undersampling* presenta un AUC constante hasta una tasa de balanceo del 65 %, donde experimenta una leve caída, sin embargo, con una tasa de balanceo del 90 %, el AUC aumenta nuevamente. De manera similar, el *oversampling* muestra una disminución en el AUC, seguida de una estabilización hasta una tasa de balanceo del 80 %, donde se observa un aumento nuevamente.

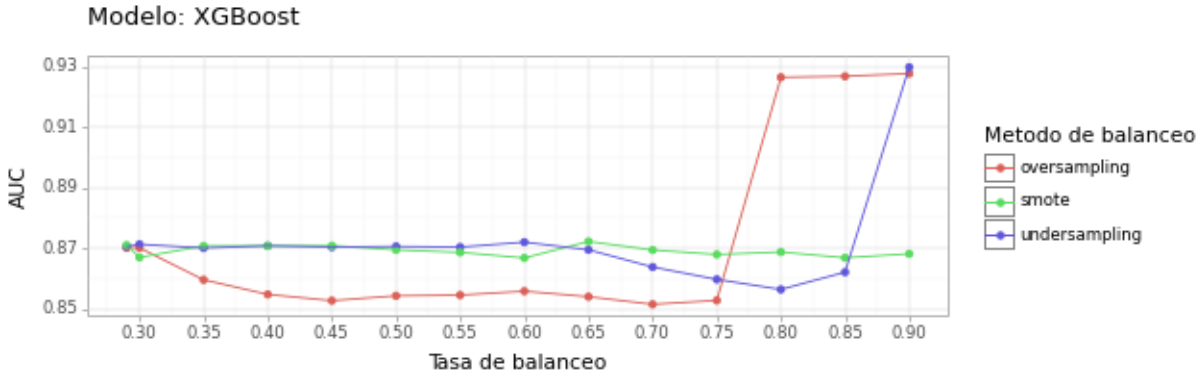


Figura 6-28.: Métrica AUC del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Por último, en la Figura **6-29**, podemos observar que el SMOTE presenta un comportamiento casi constante en el *F1 Score*, independientemente de la tasa de balanceo utilizada. Por otra

parte, tanto el *oversampling* como el *undersampling* muestran una disminución en el *F1 Score* a medida que se aumenta la tasa de balanceo. Sin embargo, con tasas de balanceo más altas, se observa nuevamente un aumento en el *F1 Score*

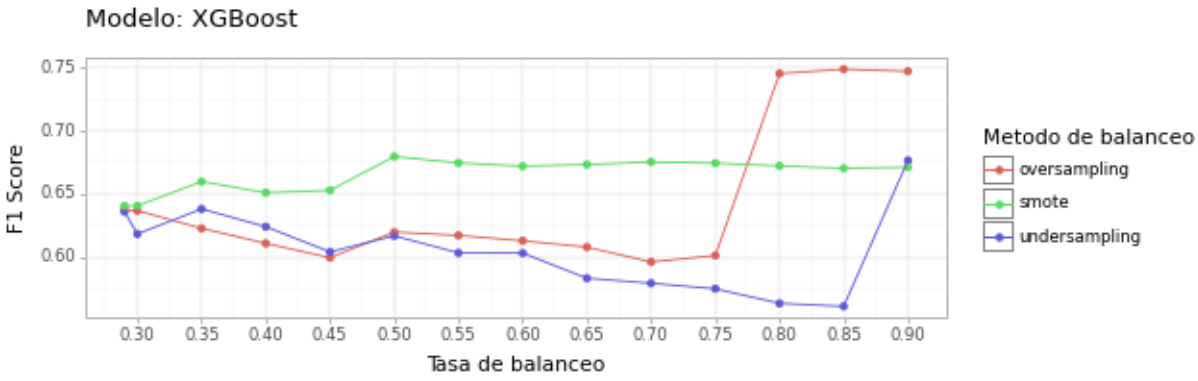


Figura 6-29.: Métrica *F1 Score* del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk analysis.

Los resultados obtenidos revelaron que el rendimiento de los modelos puede variar según la técnica y la tasa de balanceo utilizadas en el conjunto de datos credit risk analysis. Se observó que la Regresión Logística con *oversampling* y *undersampling* presentó un buen desempeño, independientemente de la tasa de balanceo utilizada. Por otro lado, el *Random Forest* mostró un desempeño más constante con tasas de balanceo bajas, menores o iguales al 40 % o 45 %, utilizando los tres métodos de balanceo. No obstante, el *XGBoost* con el SMOTE presentó el mejor desempeño en comparación con los otros dos métodos de balanceo, donde las métricas no sufrieron cambios tan notorios independientemente de la tasa de balanceo utilizada. Por lo tanto, cualquiera de estas combinaciones presentadas podría ser considerada como candidata principal para abordar el problema de otorgamiento

6.7.3. Credit Risk Customers

En la Tabla **6-27**, se presentan los resultados obtenidos al aplicar la función al conjunto de datos credit risk customers sin utilizar ningún tipo de balanceo. El modelo *XGBoost* presentó el mejor *accuracy*, a pesar de mostrar limitaciones en el *precision* con un valor del 0.46. Sin embargo, logró el mejor *F1 Score* y un AUC muy aceptable. Por otro lado, el *Random Forest* mostró un *accuracy* superior en comparación con el *XGBoost*, pero su *precision* también fue moderado. Finalmente, el Árbol de Decisiones y la Regresión Logística presentaron resultados similares.

Tabla 6-27.: Resultados de modelos sin balanceo en el base del test del conjunto de datos credit risk customers.

Model	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	0.66	0.46	0.82	0.59	0.72
Random Forest	0.73	0.55	0.58	0.57	0.76
Árbol de Desición	0.66	0.45	0.55	0.49	0.69
Logistic Regression	0.68	0.48	0.73	0.58	0.78

Logistic Regression

Con la aplicación de la Regresión Logística y diferentes tasas y métodos de balanceo, en la Figura **6-30**, podemos observar que el *undersampling* no demostró el mejor rendimiento, ya que a medida que se aumentaba la tasa de balanceo, el *accuracy* comenzaba a disminuir. En comparación, el SMOTE mostró una incremento en el *accuracy* a medida que se incrementaba la tasa de balanceo. Por otro lado, el *oversampling* obtuvo un buen desempeño con tasas de balanceo iguales o inferiores al 60 %, sin embargo, por encima de esta tasa de balanceo, el *accuracy* comenzó a disminuir nuevamente.

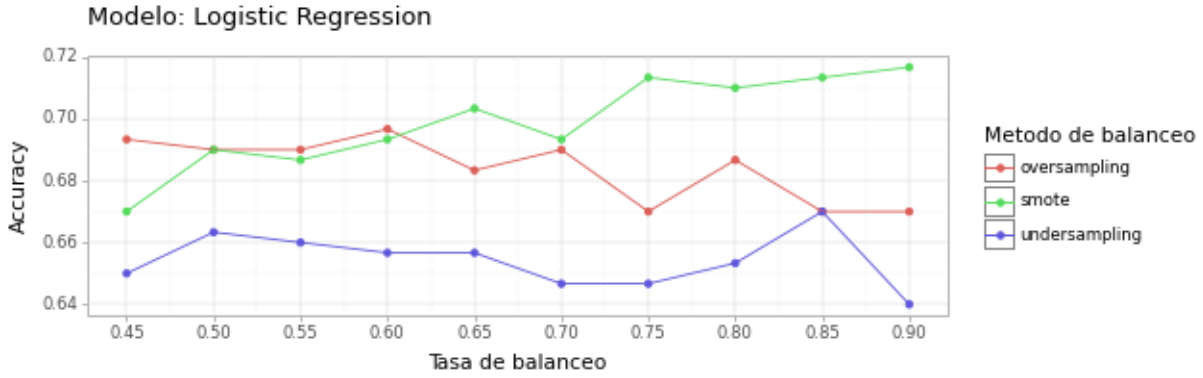


Figura 6-30.: Métrica *accuracy* de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Por otra parte, en la Figura **6-31**, observamos que, en general, el *oversampling* mostró un desempeño sólido y constante con tasas de balanceo iguales o inferiores al 85 %, donde el AUC fue superior a 0.76. Sin embargo, tanto el *undersampling* como el SMOTE presentaron una disminución en el AUC a medida que se aumentaba la tasa de balanceo, siendo el *undersampling* ligeramente más efectivo que el SMOTE.

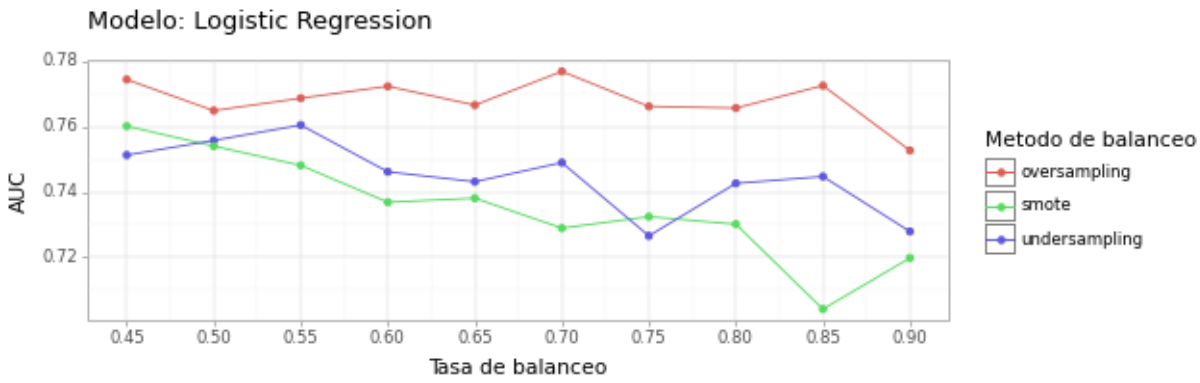


Figura 6-31.: Métrica AUC de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Igualmente, en la Figura **6-32**, podemos apreciar que el *oversampling* mostró un desempeño moderado, superando al *undersampling* y al SMOTE. Sin embargo, a medida que aumentaba la tasa de balanceo, el SMOTE exhibió una disminución en el *F1 Score*. Por otro lado, el *undersampling* mostró un desempeño aceptable.

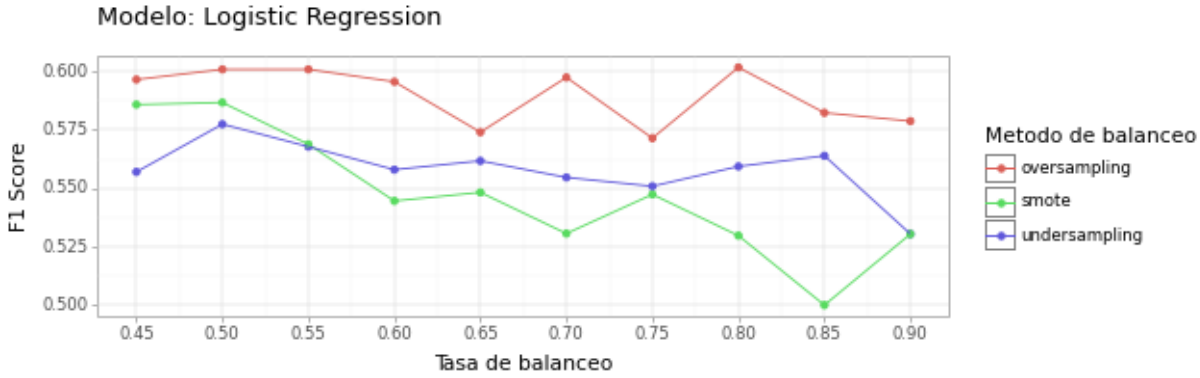


Figura 6-32.: Métrica *F1 Score* de la Regresión Logística, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Árbol de Desición

En la Figura **6-33**, se muestran los resultados del *accuracy* obtenidos con el Árbol de Decisiones y las diferentes tasas y métodos de balanceo. Tanto el *oversampling* como el *undersampling* presentaron un desempeño constante, con pequeñas variaciones en el *accuracy*. Por otro lado, el SMOTE mostró un desempeño bajo con tasas de balanceo entre el 50 % y el 60 % en contraste con los otros dos métodos. Sin embargo, con tasas de balanceo superiores o iguales al 65 %, el SMOTE mostró un desempeño similar a los otros dos métodos.

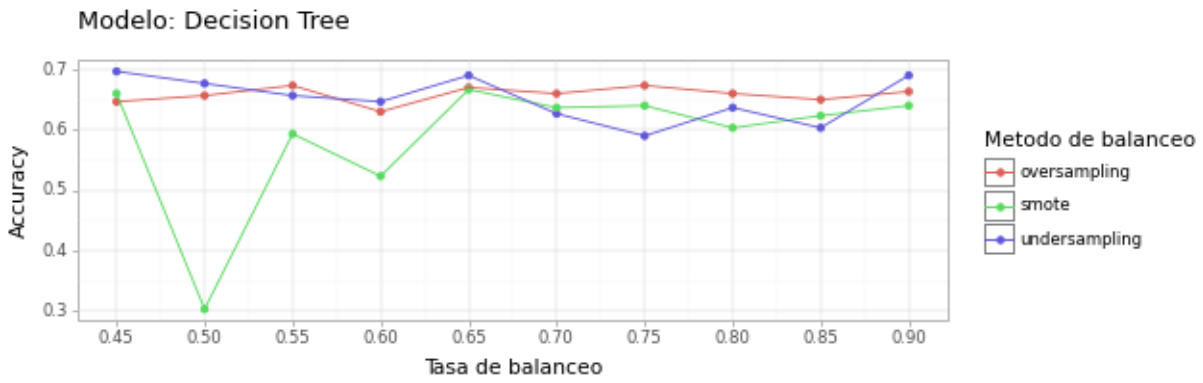


Figura 6-33.: Métrica *accuracy* del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Por otra parte, en la Figura **6-34**, observamos que tanto el *oversampling* como el SMOTE, con tasas de balanceo entre el 50 % y el 60 %, mostraron un AUC por debajo de 0.65. Sin embargo, el *undersampling*, con tasas de balanceo entre el 50 % y el 65 %, presentó el mejor AUC, con un valor por encima de 0.65. Con tasas de balanceo entre el 70 % y el 90 %, los tres métodos de balanceo mostraron resultados similares.

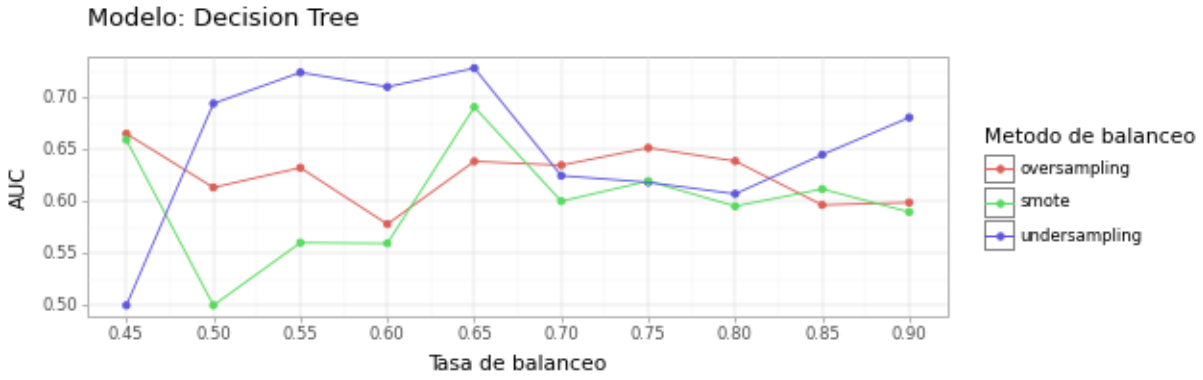


Figura 6-34.: Métrica AUC del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Por último, podemos observar en la Figura **6-35**, que el *undersampling* con una tasa de balanceo del 45 %, presentó el peor desempeño posible, sin embargo, los otros dos métodos, presentaron un *F1 Score* por encima del 0.4. Por otra parte, con una tasa de balanceo igual o superior al 50 %, podemos apreciar que con los tres métodos de balanceo, se obtuvieron valores similares en el *F1 Score*.

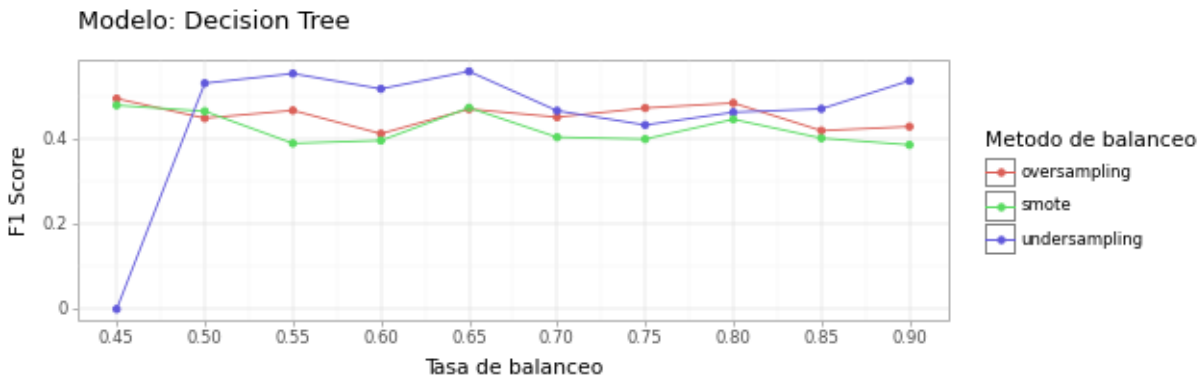


Figura 6-35.: Métrica *F1 Score* del Árbol de Decisiones, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Random Forest

Tras la aplicación del *Random Forest* con los tres métodos y las tasas de balanceo, en la Figura **6-36** podemos apreciar que se presenta una notable variabilidad en los valores del *accuracy* según el método y la tasa de balanceo empleados, en comparación con los modelos anteriores. Por otra parte, utilizando *oversampling* y tasas de balanceo del 60 %, 65 %, 75 %, 80 %, 85 % y 90 %, se obtuvo un *accuracy* superior al 0.71. Además, con tasas de balanceo del 45 % y 55 %, los tres métodos de balanceo presentaron un *accuracy* entre el 0.71 y 0.73.

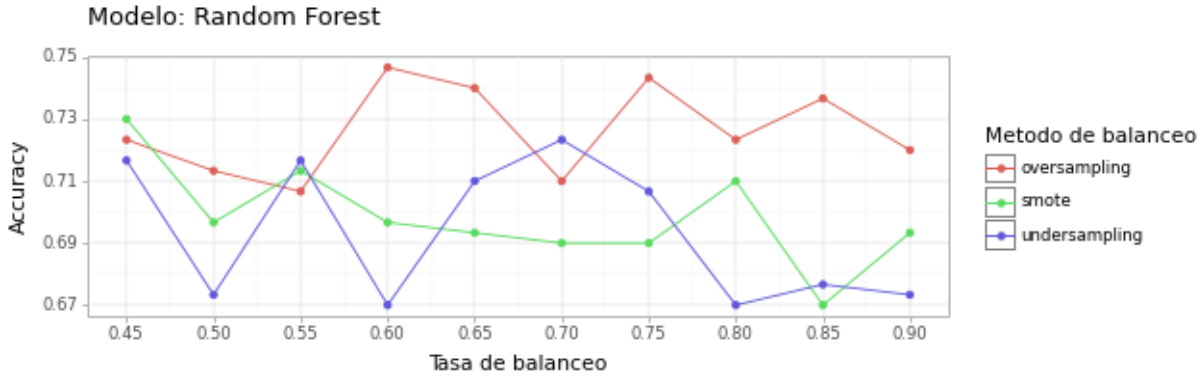


Figura 6-36.: Métrica *accuracy* del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

En la Figura **6-37**, se observa que tanto el *oversampling* como el *undersampling* exhibieron un AUC superior a 0.72, independientemente de la tasa de balanceo utilizada. Del mismo modo, el SMOTE, con tasas de balanceo entre el 45 % y el 55 %, mostró un rendimiento similar a los otros dos métodos, con un AUC superior a 0.71. Sin embargo, al aumentar la tasa de balanceo, se evidenció una disminución en la métrica de rendimiento.

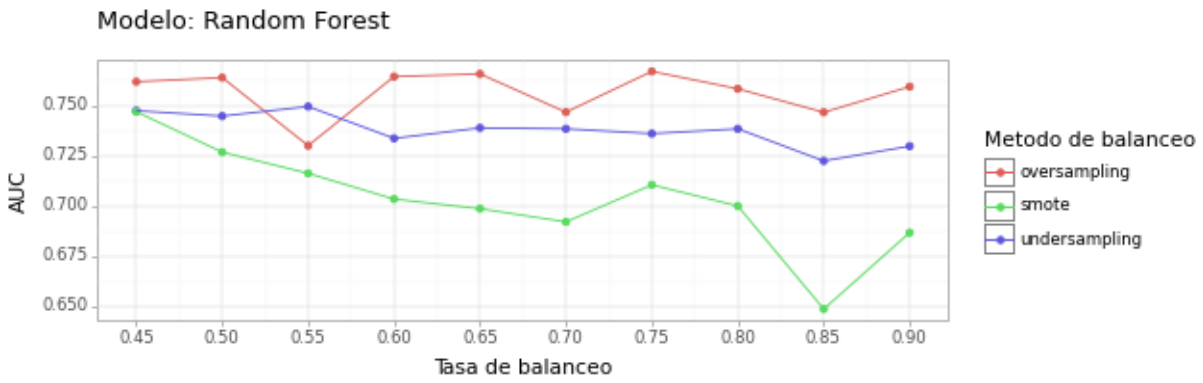


Figura 6-37.: Métrica AUC del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Por otra parte, en la Figura **6-38**, se observa que, con tasas de balanceo iguales o inferiores al 60 %, los tres métodos de balanceo muestran un rendimiento similar en el *F1 Score*. Sin embargo, al aumentar la tasa de balanceo, el SMOTE experimenta una disminución la métrica en comparación con el *oversampling* y el *undersampling*, los cuales se mantienen estables con un *F1 Score* superior a 0.53.

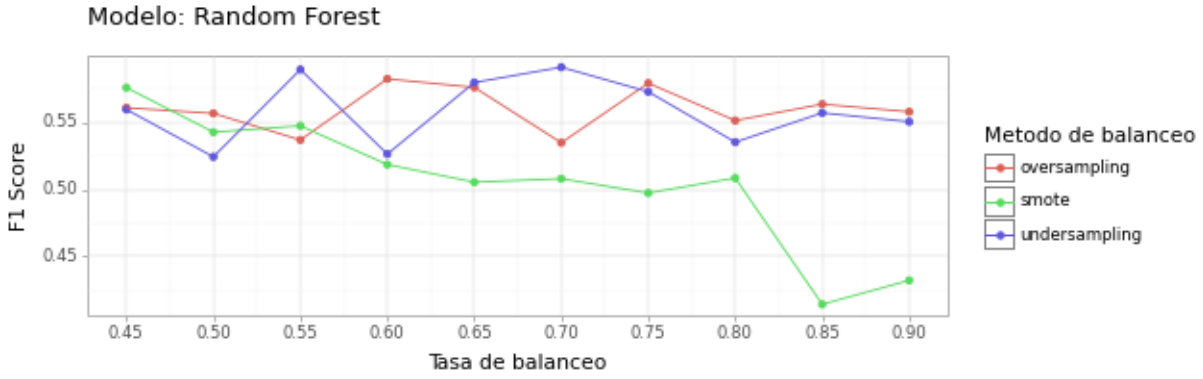


Figura 6-38.: Métrica *F1 Score* del *Random Forest*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

XGBoost

Con la aplicación del *XGBoost* y la implementación de diferentes tasas y métodos de balanceo, en la Figura **6-39** se puede observa que, con tasas de balanceo entre el 45 % y el 65 %, tanto el *oversampling* como el *undersampling* muestran un *accuracy* aceptable. Sin embargo, a medida que se aumenta la tasa de balanceo, tiende a disminuir ligeramente. Por otro lado, con tasas de balanceo superiores al 65 %, ambos métodos de balanceo experimentan una disminución en el *accuracy*, en donde, con una tasa del 85 %, el *undersampling* exhibe el

peor desempeño. En comparación, el SMOTE, con una tasa de balanceo del 45 %, presenta un desempeño aceptable. Sin embargo, al aumentar la tasa de balanceo, al igual que los métodos anteriores, se observa una disminución en la métrica.

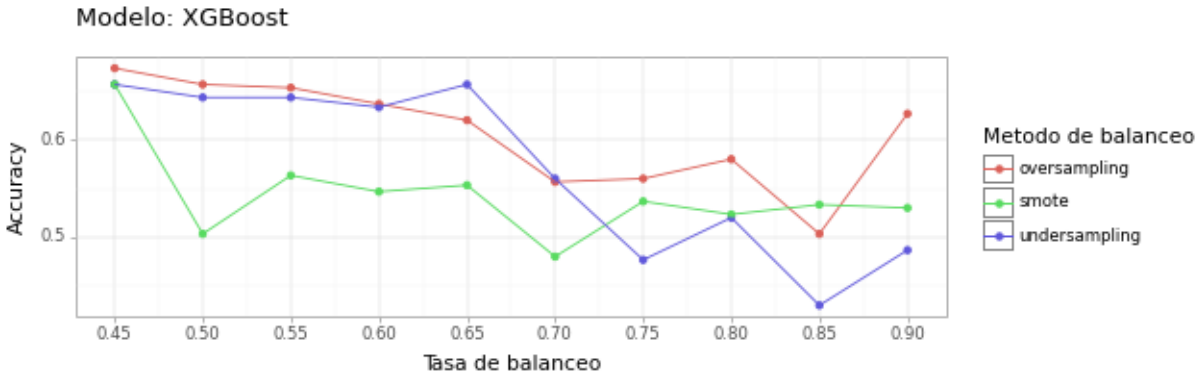


Figura 6-39.: Métrica *accuracy* del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

En la Figura **6-40**, se observa que tanto el *oversampling* como el *undersampling* muestran un AUC constante, independientemente de la tasa de balanceo utilizada, con valores entre 0.7 y 0.75. En contraste, el SMOTE, con una tasa de balanceo del 45 %, exhibe un AUC superior al 0.7. Sin embargo, a medida que se incrementa la tasa de balanceo, se observa una disminución en el AUC, llegando incluso a valores por debajo de 0.6.

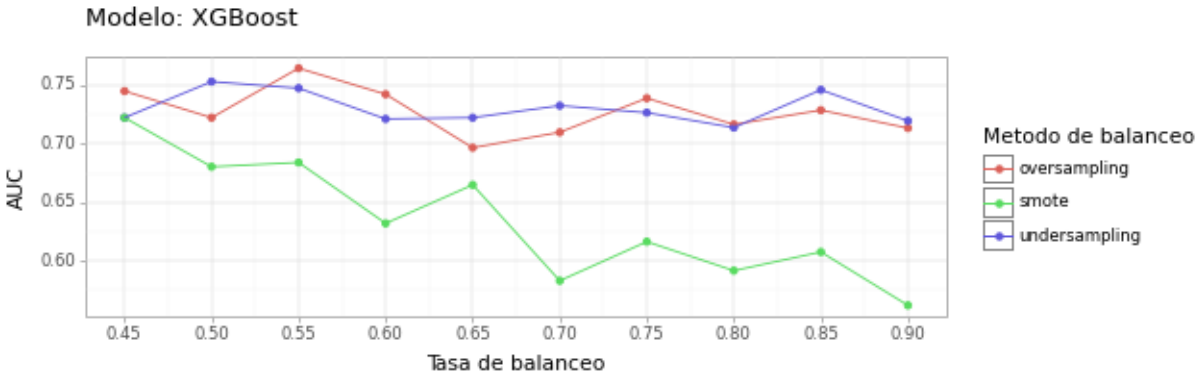


Figura 6-40.: Métrica AUC del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Por otra parte, en la Figura **6-41**, se puede observar que el *F1 Score* presenta un comportamiento similar al *accuracy*. Con tasas de balanceo entre el 45 % y el 65 %, tanto el *oversampling* como el *undersampling* presentaron un desempeño aceptable, con un *F1 Score* superior a 0.56. Sin embargo, conforme se incrementa la tasa de balanceo, se observa una

disminución. Del mismo modo, el SMOTE muestra un desempeño similar a los otros dos métodos de balanceo solo con una tasa del 45 %, con tasas superiores a esta, se evidencia una disminución en la métrica.

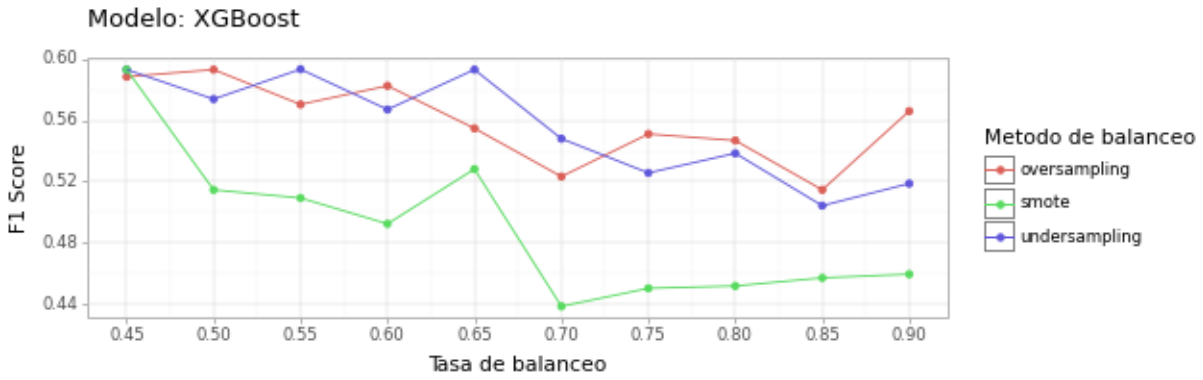


Figura 6-41.: Métrica *F1 Score* del *XGBoost*, por tasa y método de balanceo en la base del test del conjunto de datos: credit risk customers.

Los resultados obtenidos revelan varias observaciones significativas. La Regresión Logística con *oversampling* y tasas de balanceo menores o iguales al 60 %, presentó un rendimiento aceptable en comparación con los otros dos métodos de balanceo. Por otro lado, el Árbol de Decisiones con *undersampling* y tasas de balanceo entre el 50 % y el 65 % demostró un desempeño más sólido en comparación con los otros métodos y tasas de balanceo. En cuanto al Random Forest, tanto el *oversampling* como el *undersampling* pueden ser opciones viables. Por último, *XGBoost* con tasas de balanceo inferiores o iguales al 65 %, junto con *oversampling* y *undersampling*, mostró un desempeño aceptable, sin embargo, sus métricas fueron inferiores a las obtenidas con la Regresión Logística. Por lo tanto, la Regresión Logística con *oversampling* y tasas de balanceo inferiores o iguales al 60 % se perfila como la primera candidata para abordar el problema de otorgamiento, como segunda opción, podrían considerarse tanto el Árbol de Decisiones como *XGBoost*.

A continuación, se observa el diseño del tablero creado, en donde, se permite seleccionar cualquiera de los tres conjuntos de datos disponibles, junto con la combinación de modelo de machine learning y método de balanceo. Este proporciona resultados detallados de las métricas evaluadas con diferentes tasas de balanceo aplicadas. De esta manera, se obtiene una comparación global del desempeño del modelo, método y tasa de balanceo elegidos, lo que permite al usuario final identificar la combinación más adecuada según sus necesidades. De manera general, el tablero sirve como herramienta para que el analista o usuario elija fácilmente la combinación óptima de modelo, método y tasa de balanceo para cada conjunto de datos en función de las métricas de desempeño.

Comparación del desempeño de los modelos de machine learning bajo diferentes métodos de balanceo.

En este tablero se presentan los resultados de la evaluación de cuatro modelos de Machine Learning aplicados a tres conjuntos de datos. Cada modelo ha sido sometido a tres métodos de balanceo diferentes, con variadas tasas de balanceo, con el objetivo de analizar su desempeño y efectividad en el otorgamiento de crédito con datos desbalanceados.

Conjunto

Credit Approval | Credit Risk | Credit Risk Customers

Modelo

Decision Tree | Logistic Regression | Random Forest | XGBoost

Método Balanceo

Oversampling | Smote | Undersamplig

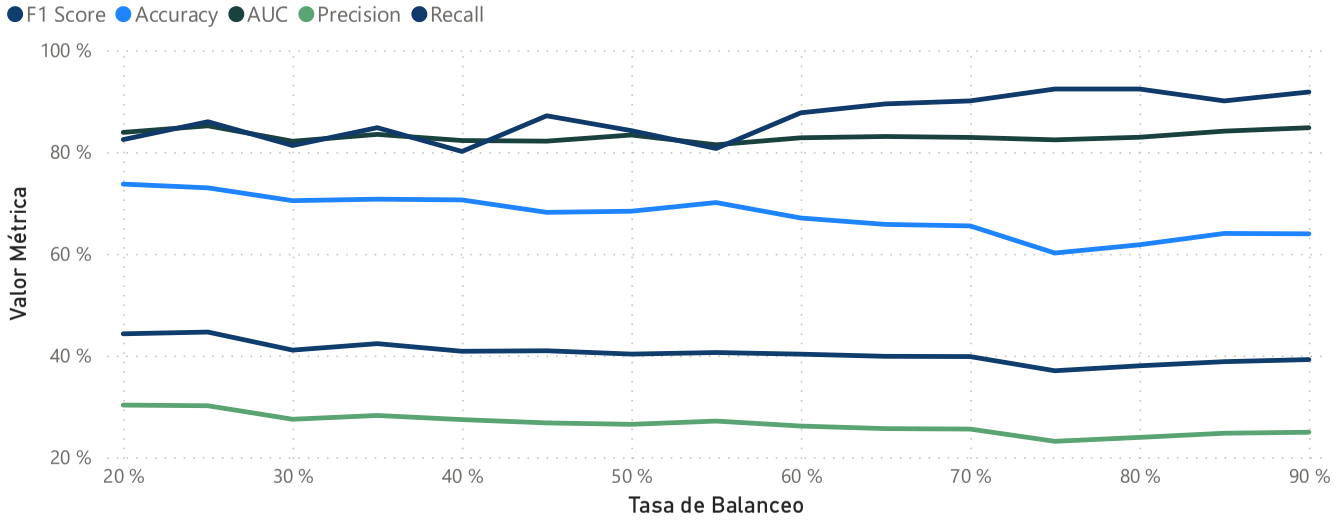
Tasa de Otorgamiento 11 %

Total de modelos estimados 15

Desempeño del modelo en el conjunto de datos sin aplicar ningún método ni tasa de balanceo

Precision	F1_Score	AUC	Recall	Accuracy
28,99 %	42,81 %	84,92 %	81,87 %	72,30 %

Comparación de las métricas de desempeño por Tasa de Balanceo



Métricas de desempeño

Tasa de Balanceo	Accuracy	AUC	F1 Score	Precision	Rec
20 %	73,70 %	83,87 %	44,27 %	30,26 %	82,4
25 %	72,96 %	85,16 %	44,61 %	30,12 %	85,9
30 %	70,44 %	82,13 %	41,06 %	27,47 %	81,2
35 %	70,74 %	83,48 %	42,34 %	28,21 %	84,8
40 %	70,59 %	82,27 %	40,83 %	27,40 %	80,1
45 %	68,15 %	82,15 %	40,93 %	26,75 %	87,1
50 %	68,37 %	83,36 %	40,28 %	26,47 %	84,2
55 %	70,07 %	81,43 %	40,59 %	27,11 %	80,7
60 %	67,04 %	82,81 %	40,27 %	26,13 %	87,7
65 %	65,78 %	83,06 %	39,84 %	25,63 %	89,4
70 %	65,48 %	82,88 %	39,79 %	25,54 %	90,0
75 %	60,15 %	82,40 %	37,00 %	23,13 %	92,4
80 %	61,78 %	82,91 %	37,98 %	23,90 %	92,4
85 %	64,00 %	84,12 %	38,79 %	24,72 %	90,0
90 %	63,93 %	84,79 %	39,20 %	24,92 %	91,8

Por último, se creo un repositorio público en GitHub denominado **MCD-Javeriana-Cali** (link-repositorio). En este repositorio se encuentran cinco archivos:

- balance_and_train_multiple_models: Función creada para las iteraciones.
- Tablero resultados.pbit: Plantilla del tablero diseñado.
- Preview - Tablero Resultados.pdf: Pdf del tablero que muestra el diseño.
- Resultados_modelos_cojuntos_de_datos.xlsx : Resultados de todas las iteraciones con las diferentes tasas y métodos de balanceo.
- Resultados_modelos_cojuntos_de_datos_sin_balanceo.xlsx: Resultados de todas las iteraciones con los diferentes métodos de balanceo sin aplicar ninguna tasa de balanceo.

7. Conclusiones y recomendaciones

En este trabajo se ha estudiado y buscado el mejor modelo para el otorgamiento de crédito cuando la tasa de otorgamiento no está completamente balanceada, utilizando las métricas adecuadas para evaluar el rendimiento de los modelos bajo diferentes escenarios de balanceo. Este análisis se realizó usando modelos de machine learning, métodos de balanceo y diferentes tasas de balanceo.

El tratamiento adecuado de los datos antes de construir cualquier modelo es una etapa fundamental en el proceso de modelamiento. Este pre-procesamiento incluye diversas técnicas, como la limpieza de datos para manejar valores atípicos y valores faltantes, la normalización para estandarizar escalas, y la categorización de variables. Además, de la selección cuidadosa de variables o características para mejorar la eficiencia y la capacidad de los modelos.

Se desarrolló la función balance_and_train_multiple_models, la cuál tuvo como objetivo principal proporcionar una herramienta capaz de aplicar un enfoque sistemático y eficiente en la evaluación de múltiples modelos en diversos conjuntos de datos, utilizando distintos métodos y tasas de balanceo. Gracias a esta función, se logró estimar 184 modelos para el conjunto de datos credit approval, 172 modelos para el conjunto de datos credit risk analysis y, finalmente, 124 modelos para el conjunto de datos credit risk customers. En total, se estimaron 480 modelos con el fin de identificar el mejor modelo con el método y tasa de balanceo más adecuada para cada base de datos.

La Regresión Logística, el *XGBoost* y el *Random Forest* demostraron ser consistentemente sólidos en todas las bases de datos, exhibiendo mejoras notables en las métricas cuando se implementaron estrategias de balanceo de clases. En el conjunto de datos credit approval, el *XGBoost* con el SMOTE con una tasa de balanceo entre el 25 % y el 75 % se destacan como las mejores opciones, teniendo un balance muy aceptable en sus métricas. En el conjunto de datos credit risk analysis, el *Random Forest* con el SMOTE y tasas de balanceo menores o iguales al 40 % emerge como líder, mostrando un equilibrio óptimo, como segunda opción, se presenta el *XGBoost* con el SMOTE, independientemente de la tasa de balanceo. Para el conjunto de datos credit risk customers, la Regresión Logística con tasas de balanceo inferiores o iguales al 60 %, teniendo en cuenta el *oversampling*, es la primera candidata por su gran desempeño.

Un aspecto destacado es la importancia del balanceo de clases, donde se observa una mejora general en las métricas de rendimiento al aplicar técnicas de balanceo. Sin embargo, es crucial encontrar un equilibrio adecuado en la tasa de balanceo, ya que aumentos excesivos pueden tener un impacto negativo en el desempeño del modelo, dado que se podría estar incurriendo en un sobre ajuste del modelo, como se pudo apreciar en varios casos.

En resumen, se recomienda un enfoque personalizado, considerando la naturaleza específica de cada conjunto de datos. La implementación exitosa de modelos predictivos no solo depende del algoritmo seleccionado, sino también de la estrategia de balanceo de clases adecuada y del conocimiento del experto de negocio.

Recomendaciones

Para futuros estudios, se sugiere ampliar las características en el análisis, incorporando elementos adicionales como el perfil de morosidad de los clientes, los saldos de los diversos productos, el número de transacciones realizadas en un periodo determinado, los ingresos y gastos mensuales, y la relación del cliente con otras entidades bancarias, entre otras. Estas nuevas variables ayudarían al conjunto de datos, permitiendo la identificación de características relevantes que podrían mejorar la precisión del modelo, reduciendo así la dependencia exclusiva de técnicas de balanceo de clases.

Además, para complementar la evaluación de los modelos, se propone una base de datos adicional con información más actual posible, realizando una prueba conocida como Out of Time (OOT). Realizar esta prueba permitiría evaluar cómo se comporta el modelo frente a datos recientes, proporcionando una validación adicional de su robustez a lo largo del tiempo. También se sugiere explorar una variedad más amplia de modelos de machine learning para construir un conjunto champion challenger más completo o exhaustivo, permitiendo una comparación más compleja y una mejor selección de modelos.

A. Anexo: Función de modelamiento

```
def balance_and_train_multiple_models(df, balance_ratio, balancing_method):


    # Particion dataframe de test y train
    train_df, test_df = train_test_split(df, test_size=0.3, random_state=42)

    X_train = train_df.drop('target', axis=1)
    y_train = train_df['target']
    X_test = test_df.drop('target', axis=1)
    y_test = test_df['target']



    if balancing_method == 'over':
        oversampler = RandomOverSampler(sampling_strategy = balance_ratio,
                                        random_state=42)
        X_train_resampled, y_train_resampled = oversampler.fit_resample(X_train, y_train)

    elif balancing_method == 'under':
        undersampler = RandomUnderSampler(sampling_strategy = balance_ratio,
                                          random_state=42)
        X_train_resampled, y_train_resampled = undersampler.fit_resample(X_train, y_train)

    elif balancing_method == 'smote':
        smote = SMOTE(sampling_strategy = balance_ratio,
                      random_state=42)
        X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

    elif balancing_method == 'ninguno':
        X_train_resampled = X_train.copy()
        y_train_resampled = y_train.copy()


    tasa_target = round(y_train.value_counts(normalize=True)[1],2)
```

```
35     tasa_target_resam = round(y_train_resampled.value_counts(normalize=
       True)[1],2)
36
37
38     param_scale_pos = np.ceil(y_train_resampled.shape[0] /
       y_train_resampled.sum()).astype(int)
39     param_scale_pos
40
41     models = [
42         ('XGBoost', xgb.XGBClassifier(objective= 'binary:logistic', seed
       =42, scale_pos_weight = param_scale_pos)),
43         ('Random Forest', RandomForestClassifier(random_state=42,
       class_weight='balanced')),
44         ('Decision Tree', DecisionTreeClassifier(random_state=42,
       class_weight='balanced')),
45         ('Logistic Regression', LogisticRegression(random_state=42,
       class_weight='balanced', max_iter = 1000))
46     ]
47
48     scoring = {'balanced_accuracy': 'balanced_accuracy',
49                'recall': 'recall',
50                'f1':'f1',
51                'neg_log_loss': 'neg_log_loss',
52                'precision': 'precision'}
53
54     results = [] #lista para almacenar los resultados de los modelos
55     selected_features = []  # Lista para almacenar las variables
       seleccionadas
56     parameters = []  # Lista para almacenar los par metros
57
58     for model_name, base_model in models:
59         if model_name == 'XGBoost':
60             param_dist = {
61                 'max_depth': range (1, 200, 5),
62                 'n_estimators': range(10, 500,20),
63                 'learning_rate': np.arange(0,0.55,0.05),
64                 'min_child_weight' : range (10, 550, 20),
65                 'subsample': np.arange(0,1,0.025),
66                 'gamma':  np.arange(0,1,0.05),
67                 'colsample_bytree':np.arange(0.3,0.7,0.05)
68             }
69
70
71             model = RandomizedSearchCV(base_model,
72                                        param_distributions=param_dist,
73                                        n_iter=100,
74                                        scoring = scoring,
75                                        refit='balanced_accuracy',
```

```
                                    n_jobs=-1,
                                    cv=20,
                                    random_state=42)

        elif model_name == 'Random Forest':
            param_dist = {
                'n_estimators'      : np.arange(5,500,20),
                'min_samples_split' : [2, 4, 6,8, 10, 15, 20, 30, 40],
                'min_samples_leaf'  : [10, 30, 60, 90, 100, 150, 200],
                'max_features'      : ['sqrt','auto',0.25, 0.3, 0.5, 0.75,
 1.0],
                'max_depth'         : range (1, 300, 5),
                'bootstrap'         : [True, False],
                'criterion'          :['gini', 'entropy']
            }
            model = RandomizedSearchCV(base_model,
                                    param_distributions=param_dist,
                                    n_iter=100,
                                    #scoring='f1',
                                    refit = 'balanced_accuracy',
                                    n_jobs=-1,
                                    cv=20,
                                    random_state=42)

        elif model_name == 'Decision Tree':
            param_dist = {
                'ccp_alpha':np.arange(0,1,0.02),
                'min_samples_split': [ 10, 15, 20, 30, 40],
                'max_features': ['sqrt','auto',0.4,0.45, 0.5,
0.55,0.6,0.65, 0.75, 1.0],
                'max_depth': range (1, 300, 5)
            }
            model = RandomizedSearchCV(base_model,
                                    param_distributions=param_dist,
                                    n_iter=100,
                                    #scoring='f1',
                                    refit = 'balanced_accuracy',
                                    n_jobs=-1,
                                    cv=20,
                                    random_state=42)

        elif model_name == 'Logistic Regression':
            param_dist = {
                'penalty': ['l1', 'l2'],
                'C': [0.001, 0.01, 0.1, 1],
                'tol': [0.000001,0.00001,0.0001,0.001, 0.01, 0.1, 1],
                'solver': ['liblinear'],
            }
```

```
122             model = RandomizedSearchCV(base_model,
123                                        param_distributions=param_dist,
124                                        n_iter=10,
125                                        scoring='balanced_accuracy',
126                                        n_jobs=-1,
127                                        cv=20,
128                                        random_state=42)
129
130
131
132
133         # Ajusa modelo 1
134
135
136         # Usar el modelo ajustado para la selecci n de caracter sticas
137         selector = SelectFromModel(estimator = base_model)
138         selector.fit(X_train_resampled, y_train_resampled)
139
140         X_train_selected = selector.transform(X_train_resampled)
141         X_test_selected  = selector.transform(X_test)
142
143         # variables seleccionadas
144         selected_features = list(X_train_resampled.columns[selector.
    get_support()])
145
146         # Entrenar y evaluar el modelo con las caracter sticas
    seleccionadas
147         model.fit(X_train_selected, y_train_resampled)
148
149         # Obtener los mejores par metros
150         parameters = model.best_params_ if hasattr(model, 'best_params_')
    else None
151
152
153
154         y_t_prob = model.predict_proba(X_train_selected)
155         y_t_prob = pd.DataFrame(y_t_prob[:, 1]).iloc[:,0]
156
157         y_pred = model.predict(X_test_selected)
158         y_prob = model.predict_proba(X_test_selected)
159         y_prob = pd.DataFrame(y_prob[:, 1]).iloc[:,0]
160
161
162         #Metricas
163
164         accuracy  = accuracy_score(y_test, y_pred)
165         precision = precision_score(y_test, y_pred)
166         recall    = recall_score(y_test, y_pred)
```

```
167         f1        = f1_score(y_test, y_pred)
168         auc_score = roc_auc_score(y_test, y_prob)
169 
170         #punto de corte
171         threshold = Find_Optimal_Cutoff(y_train_resampled, y_t_prob)
172 
173         y_pred_pc = np.where(y_prob > threshold, 1, 0)
174 
175         accuracy_pc  = accuracy_score(y_test, y_pred_pc)
176         precision_pc = precision_score(y_test, y_pred_pc)
177         recall_pc    = recall_score(y_test, y_pred_pc)
178         f1_pc        = f1_score(y_test, y_pred_pc)
179 
180 
181 
182         results.append({
183             'Model'           : model_name,
184             'Balance_Method'  : balancing_method,
185             'Balance_Ratio'   : balance_ratio,
186             'Target_Ratio'    : tasa_target,
187             'New_Target_Ratio'    : tasa_target_resam,
188 
189             'Accuracy'        : accuracy,
190             'Precision'       : precision,
191             'Recall'          : recall,
192             'F1_Score'        : f1,
193             'AUC'             : auc_score,
194             'Threshold'       : threshold,
195             'Accuracy_Th'     : accuracy_pc,
196             'Precision_Th'    : precision_pc,
197             'Recall_Th'       : recall_pc,
198             'F1_Score_Th'     : f1_pc,
199             'Variables'       : selected_features,
200             'Parametros'      : parameters
201         })
202 
203     result_df = pd.DataFrame(results)
204     return result_df
205 
206 
207 # Define las tasas de balanceo que deseas probar
208 balance_ratios = [0.2,0.25,0.3]
209 
210 # Define los m todos de balanceo que deseas probar
211 balancing_methods = ['over', 'under', 'smote']
212 
213 # Crea una lista para almacenar los resultados de todas las combinaciones
214 all_results = []
```

```
215
216
217 # Itera sobre las tasas de balanceo y los m todos de balanceo
218 for balance_ratio in balance_ratios:
219     for balancing_method in balancing_methods:
220         # Llama a la funci n balance_and_train_multiple_models
221         results_1 = balance_and_train_multiple_models(df_selecc,
    balance_ratio, balancing_method)
222         # Agrega los resultados a la lista
223         all_results.append(results_1)
224
225 # Combina todos los resultados en un solo DataFrame
226 final_results_1 = pd.concat(all_results, ignore_index=True)
```

B. Anexo: Tablas completas de las métricas de los modelos

Credit Approval

Tabla B-1.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 1)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.20	0.79	0.31	0.53	0.39	0.79
Random Forest			0.83	0.38	0.59	0.46	0.83
Decision Tree			0.87	0.00	0.00	0.00	0.50
Logistic Regression			0.79	0.34	0.75	0.47	0.85
XGBoost	under	0.20	0.74	0.30	0.82	0.44	0.84
Random Forest			0.80	0.35	0.67	0.46	0.84
Decision Tree			0.74	0.28	0.70	0.40	0.77
Logistic Regression			0.78	0.33	0.73	0.46	0.85
XGBoost	smote	0.20	0.74	0.30	0.82	0.44	0.86
Random Forest			0.82	0.36	0.57	0.44	0.83
Decision Tree			0.87	0.00	0.00	0.00	0.50
Logistic Regression			0.80	0.36	0.73	0.48	0.85
XGBoost	ove	0.25	0.78	0.31	0.58	0.40	0.79
Random Forest			0.83	0.37	0.54	0.44	0.82
Decision Tree			0.77	0.27	0.50	0.35	0.67
Logistic Regression			0.78	0.33	0.73	0.46	0.85
XGBoost	under	0.25	0.73	0.30	0.86	0.45	0.85
Random Forest			0.79	0.34	0.70	0.45	0.84
Decision Tree			0.71	0.27	0.74	0.40	0.73
Logistic Regression			0.79	0.35	0.72	0.47	0.86
XGBoost	smote	0.25	0.73	0.30	0.80	0.43	0.85
Random Forest			0.81	0.35	0.54	0.43	0.81
Decision Tree			0.77	0.29	0.56	0.38	0.71
Logistic Regression			0.80	0.36	0.73	0.48	0.85

Tabla B-2.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 2)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.30	0.82	0.37	0.55	0.44	0.81
Random Forest			0.83	0.38	0.56	0.45	0.82
Decision Tree			0.79	0.30	0.50	0.37	0.68
Logistic Regression			0.78	0.33	0.73	0.46	0.85
XGBoost	under	0.30	0.70	0.27	0.81	0.41	0.82
Random Forest			0.77	0.32	0.70	0.44	0.84
Decision Tree			0.13	0.13	1.00	0.22	0.50
Logistic Regression			0.79	0.35	0.75	0.47	0.85
XGBoost	smote	0.30	0.74	0.31	0.82	0.45	0.85
Random Forest			0.83	0.39	0.58	0.46	0.83
Decision Tree			0.78	0.30	0.51	0.38	0.68
Logistic Regression			0.78	0.33	0.69	0.44	0.84
XGBoost	over	0.35	0.78	0.30	0.55	0.39	0.77
Random Forest			0.83	0.39	0.57	0.46	0.82
Decision Tree			0.80	0.31	0.49	0.38	0.68
Logistic Regression			0.78	0.33	0.71	0.45	0.85
XGBoost	under	0.35	0.71	0.28	0.85	0.42	0.83
Random Forest			0.77	0.32	0.71	0.44	0.84
Decision Tree			0.13	0.13	1.00	0.22	0.50
Logistic Regression			0.79	0.35	0.72	0.47	0.86
XGBoost	smote	0.35	0.78	0.33	0.69	0.44	0.84
Random Forest			0.82	0.37	0.64	0.47	0.84
Decision Tree			0.81	0.33	0.51	0.40	0.74
Logistic Regression			0.79	0.35	0.72	0.47	0.85

Tabla B-3.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 3)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.40	0.77	0.30	0.60	0.40	0.80
Random Forest			0.83	0.39	0.57	0.46	0.83
Decision Tree			0.76	0.24	0.41	0.31	0.63
Logistic Regression			0.78	0.33	0.72	0.46	0.85
XGBoost	under	0.40	0.71	0.27	0.80	0.41	0.82
Random Forest			0.76	0.31	0.74	0.44	0.83
Decision Tree			0.71	0.27	0.74	0.40	0.73
Logistic Regression			0.79	0.35	0.74	0.47	0.85
XGBoost	smote	0.40	0.76	0.31	0.74	0.44	0.84
Random Forest			0.82	0.37	0.61	0.46	0.84
Decision Tree			0.78	0.29	0.50	0.36	0.73
Logistic Regression			0.77	0.31	0.69	0.43	0.84
XGBoost	over	0.45	0.81	0.34	0.54	0.41	0.81
Random Forest			0.83	0.40	0.57	0.47	0.83
Decision Tree			0.78	0.26	0.39	0.31	0.64
Logistic Regression			0.78	0.33	0.73	0.46	0.85
XGBoost	under	0.45	0.68	0.27	0.87	0.41	0.82
Random Forest			0.75	0.31	0.75	0.44	0.83
Decision Tree			0.71	0.27	0.74	0.40	0.73
Logistic Regression			0.79	0.35	0.73	0.47	0.85
XGBoost	smote	0.45	0.77	0.32	0.74	0.45	0.84
Random Forest			0.82	0.37	0.60	0.46	0.84
Decision Tree			0.80	0.33	0.55	0.41	0.76
Logistic Regression			0.79	0.35	0.73	0.47	0.85

Tabla B-4.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 4)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.50	0.78	0.30	0.56	0.39	0.78
Random Forest			0.83	0.39	0.57	0.46	0.83
Decision Tree			0.79	0.29	0.43	0.34	0.67
Logistic Regression			0.79	0.34	0.73	0.46	0.85
XGBoost	under	0.50	0.68	0.26	0.84	0.40	0.83
Random Forest			0.75	0.30	0.75	0.43	0.83
Decision Tree			0.71	0.27	0.74	0.40	0.73
Logistic Regression			0.79	0.35	0.71	0.47	0.85
XGBoost	smote	0.50	0.80	0.35	0.68	0.46	0.82
Random Forest			0.83	0.40	0.61	0.48	0.84
Decision Tree			0.78	0.27	0.44	0.34	0.71
Logistic Regression			0.80	0.35	0.71	0.47	0.85
XGBoost	over	0.55	0.78	0.31	0.59	0.41	0.80
Random Forest			0.84	0.40	0.54	0.46	0.83
Decision Tree			0.81	0.31	0.44	0.37	0.68
Logistic Regression			0.79	0.34	0.73	0.46	0.85
XGBoost	under	0.55	0.70	0.27	0.81	0.41	0.81
Random Forest			0.73	0.29	0.79	0.43	0.83
Decision Tree			0.71	0.27	0.74	0.40	0.73
Logistic Regression			0.79	0.34	0.73	0.47	0.85
XGBoost	smote	0.55	0.79	0.34	0.70	0.46	0.82
Random Forest			0.83	0.39	0.59	0.47	0.84
Decision Tree			0.77	0.26	0.43	0.32	0.69
Logistic Regression			0.80	0.35	0.71	0.47	0.85

Tabla B-5.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 5)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.60	0.80	0.33	0.56	0.41	0.80
Random Forest			0.84	0.39	0.53	0.45	0.83
Decision Tree			0.79	0.29	0.43	0.35	0.67
Logistic Regression			0.78	0.33	0.72	0.45	0.85
XGBoost	under	0.60	0.67	0.26	0.88	0.40	0.83
Random Forest			0.73	0.30	0.80	0.43	0.83
Decision Tree			0.71	0.27	0.74	0.40	0.73
Logistic Regression			0.79	0.34	0.73	0.46	0.85
XGBoost	smote	0.60	0.79	0.34	0.73	0.47	0.83
Random Forest			0.83	0.38	0.59	0.46	0.83
Decision Tree			0.80	0.32	0.53	0.40	0.74
Logistic Regression			0.80	0.35	0.70	0.47	0.85
XGBoost	over	0.65	0.81	0.34	0.56	0.42	0.81
Random Forest			0.84	0.39	0.52	0.45	0.83
Decision Tree			0.80	0.31	0.45	0.37	0.68
Logistic Regression			0.78	0.33	0.73	0.46	0.85
XGBoost	under	0.65	0.66	0.26	0.89	0.40	0.83
Random Forest			0.73	0.29	0.79	0.43	0.83
Decision Tree			0.71	0.27	0.74	0.40	0.73
Logistic Regression			0.79	0.34	0.72	0.46	0.85
XGBoost	smote	0.65	0.79	0.33	0.70	0.45	0.83
Random Forest			0.84	0.41	0.52	0.46	0.82
Decision Tree			0.79	0.30	0.47	0.37	0.73
Logistic Regression			0.79	0.34	0.70	0.45	0.81

Tabla B-6.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 6)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.70	0.81	0.34	0.55	0.42	0.80
Random Forest			0.84	0.40	0.53	0.46	0.83
Decision Tree			0.77	0.22	0.31	0.25	0.57
Logistic Regression			0.78	0.33	0.73	0.46	0.85
XGBoost	under	0.70	0.65	0.26	0.90	0.40	0.83
Random Forest			0.73	0.29	0.82	0.43	0.83
Decision Tree			0.71	0.28	0.77	0.41	0.74
Logistic Regression			0.79	0.35	0.74	0.47	0.85
XGBoost	smote	0.70	0.79	0.34	0.71	0.46	0.82
Random Forest			0.84	0.41	0.51	0.45	0.81
Decision Tree			0.79	0.30	0.49	0.37	0.71
Logistic Regression			0.61	0.17	0.53	0.26	0.57
XGBoost	over	0.75	0.78	0.32	0.62	0.42	0.79
Random Forest			0.84	0.41	0.53	0.46	0.83
Decision Tree			0.81	0.32	0.46	0.38	0.68
Logistic Regression			0.78	0.34	0.73	0.46	0.85
XGBoost	under	0.75	0.60	0.23	0.92	0.37	0.82
Random Forest			0.74	0.30	0.79	0.43	0.83
Decision Tree			0.71	0.28	0.77	0.41	0.74
Logistic Regression			0.79	0.35	0.74	0.47	0.85
XGBoost	smote	0.75	0.76	0.31	0.70	0.42	0.82
Random Forest			0.84	0.40	0.55	0.46	0.82
Decision Tree			0.80	0.30	0.42	0.35	0.71
Logistic Regression			0.61	0.17	0.53	0.26	0.57

Tabla B-7.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 7)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.80	0.81	0.35	0.59	0.44	0.81
Random Forest		0.80	0.84	0.40	0.54	0.46	0.83
Decision Tree		0.80	0.80	0.30	0.43	0.35	0.66
Logistic Regression		0.80	0.78	0.34	0.74	0.46	0.85
XGBoost	under	0.80	0.62	0.24	0.92	0.38	0.83
Random Forest		0.80	0.75	0.31	0.78	0.44	0.83
Decision Tree		0.80	0.71	0.28	0.77	0.41	0.74
Logistic Regression		0.80	0.78	0.34	0.75	0.47	0.85
XGBoost	smote	0.80	0.77	0.31	0.69	0.43	0.82
Random Forest		0.80	0.84	0.39	0.53	0.45	0.82
Decision Tree		0.80	0.79	0.29	0.43	0.34	0.73
Logistic Regression		0.80	0.61	0.17	0.53	0.26	0.57
XGBoost	over	0.85	0.81	0.34	0.57	0.43	0.81
Random Forest		0.85	0.85	0.42	0.54	0.47	0.83
Decision Tree		0.85	0.80	0.30	0.44	0.36	0.66
Logistic Regression		0.85	0.78	0.34	0.74	0.46	0.85
XGBoost	under	0.85	0.64	0.25	0.90	0.39	0.84
Random Forest		0.85	0.74	0.29	0.78	0.43	0.83
Decision Tree		0.85	0.71	0.28	0.77	0.41	0.74
Logistic Regression		0.85	0.80	0.35	0.72	0.47	0.86
XGBoost	smote	0.85	0.79	0.33	0.65	0.44	0.81
Random Forest		0.85	0.84	0.39	0.49	0.43	0.80
Decision Tree		0.85	0.80	0.29	0.40	0.35	0.73
Logistic Regression		0.85	0.49	0.15	0.67	0.25	0.56

Tabla B-8.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Approval (Parte 8)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.90	0.80	0.34	0.58	0.42	0.80
Random Forest		0.90	0.84	0.40	0.51	0.45	0.83
Decision Tree		0.90	0.81	0.33	0.45	0.38	0.67
Logistic Regression		0.90	0.78	0.34	0.73	0.46	0.85
XGBoost	under	0.90	0.64	0.25	0.92	0.39	0.85
Random Forest		0.90	0.73	0.29	0.78	0.42	0.83
Decision Tree		0.90	0.71	0.27	0.74	0.40	0.73
Logistic Regression		0.90	0.73	0.29	0.77	0.42	0.84
XGBoost	smote	0.90	0.78	0.33	0.68	0.44	0.81
Random Forest		0.90	0.84	0.39	0.51	0.44	0.81
Decision Tree		0.90	0.81	0.32	0.40	0.35	0.73
Logistic Regression		0.90	0.61	0.17	0.54	0.26	0.58

Tabla B-9.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Analysis (Parte 1)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.29	0.81	0.55	0.76	0.64	0.87
Random Forest			0.85	0.63	0.75	0.68	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.49	0.77	0.60	0.85
XGBoost	under	0.29	0.81	0.55	0.76	0.64	0.87
Random Forest			0.85	0.63	0.76	0.69	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	smote	0.29	0.81	0.55	0.76	0.64	0.87
Random Forest			0.85	0.63	0.76	0.69	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	over	0.30	0.81	0.55	0.76	0.64	0.87
Random Forest			0.85	0.63	0.76	0.68	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	under	0.30	0.81	0.55	0.76	0.64	0.87
Random Forest			0.85	0.62	0.76	0.68	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	smote	0.30	0.81	0.55	0.76	0.64	0.87
Random Forest			0.85	0.62	0.76	0.69	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85

Tabla B-10.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Analysis (Parte 2)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.35	0.80	0.54	0.74	0.62	0.86
Random Forest			0.84	0.61	0.76	0.68	0.89
Decision Tree			0.80	0.53	0.72	0.61	0.82
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	under	0.35	0.81	0.55	0.76	0.64	0.87
Random Forest			0.85	0.62	0.76	0.68	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.78	0.49	0.77	0.60	0.85
XGBoost	smote	0.35	0.83	0.60	0.74	0.66	0.87
Random Forest			0.85	0.62	0.76	0.68	0.89
Decision Tree			0.81	0.56	0.71	0.63	0.79
Logistic Regression			0.78	0.49	0.76	0.60	0.85
XGBoost	over	0.40	0.80	0.52	0.74	0.61	0.85
Random Forest			0.84	0.62	0.76	0.68	0.88
Decision Tree			0.81	0.54	0.71	0.62	0.82
Logistic Regression			0.77	0.49	0.77	0.59	0.85
XGBoost	under	0.40	0.79	0.52	0.78	0.62	0.87
Random Forest			0.84	0.61	0.77	0.68	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.79	0.60	0.85
XGBoost	smote	0.40	0.83	0.59	0.73	0.65	0.87
Random Forest			0.85	0.62	0.76	0.68	0.89
Decision Tree			0.81	0.56	0.71	0.63	0.79
Logistic Regression			0.76	0.47	0.74	0.57	0.83

Tabla B-11.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Analysis (Parte 3)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.45	0.78	0.50	0.75	0.60	0.85
Random Forest			0.85	0.62	0.76	0.68	0.88
Decision Tree			0.81	0.55	0.70	0.61	0.82
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	under	0.45	0.77	0.49	0.79	0.60	0.87
Random Forest			0.84	0.61	0.77	0.68	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.49	0.77	0.60	0.85
XGBoost	smote	0.45	0.83	0.59	0.73	0.65	0.87
Random Forest			0.85	0.63	0.74	0.68	0.88
Decision Tree			0.81	0.56	0.71	0.63	0.79
Logistic Regression			0.76	0.47	0.74	0.58	0.83
XGBoost	over	0.50	0.80	0.54	0.73	0.62	0.85
Random Forest			0.85	0.62	0.76	0.68	0.88
Decision Tree			0.81	0.56	0.71	0.62	0.82
Logistic Regression			0.77	0.48	0.78	0.59	0.85
XGBoost	under	0.50	0.79	0.52	0.77	0.62	0.87
Random Forest			0.84	0.60	0.77	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.78	0.49	0.77	0.60	0.85
XGBoost	smote	0.50	0.86	0.66	0.69	0.68	0.87
Random Forest			0.85	0.63	0.74	0.68	0.88
Decision Tree			0.81	0.55	0.72	0.62	0.83
Logistic Regression			0.76	0.47	0.74	0.58	0.83

Tabla B-12.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Analysis (Parte 4)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.55	0.80	0.53	0.74	0.62	0.85
Random Forest			0.84	0.62	0.76	0.68	0.88
Decision Tree			0.81	0.55	0.70	0.61	0.82
Logistic Regression			0.77	0.48	0.78	0.59	0.85
XGBoost	under	0.55	0.78	0.49	0.78	0.60	0.87
Random Forest			0.84	0.60	0.77	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.78	0.49	0.77	0.60	0.85
XGBoost	smote	0.55	0.85	0.65	0.70	0.67	0.87
Random Forest			0.85	0.63	0.74	0.68	0.88
Decision Tree			0.81	0.55	0.73	0.63	0.84
Logistic Regression			0.76	0.47	0.74	0.58	0.83
XGBoost	over	0.60	0.79	0.52	0.75	0.61	0.86
Random Forest			0.85	0.62	0.76	0.68	0.88
Decision Tree			0.81	0.55	0.72	0.62	0.82
Logistic Regression			0.77	0.48	0.78	0.59	0.85
XGBoost	under	0.60	0.77	0.49	0.79	0.60	0.87
Random Forest			0.84	0.60	0.77	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.49	0.78	0.60	0.85
XGBoost	smote	0.60	0.85	0.64	0.71	0.67	0.87
Random Forest			0.85	0.64	0.73	0.68	0.88
Decision Tree			0.81	0.55	0.67	0.61	0.82
Logistic Regression			0.77	0.48	0.73	0.58	0.83

Tabla B-13.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Analysis (Parte 5)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.65	0.79	0.51	0.75	0.61	0.85
Random Forest			0.85	0.63	0.75	0.68	0.88
Decision Tree			0.80	0.54	0.69	0.60	0.81
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	under	0.65	0.75	0.46	0.81	0.58	0.87
Random Forest			0.84	0.60	0.77	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.49	0.78	0.60	0.85
XGBoost	smote	0.65	0.85	0.64	0.71	0.67	0.87
Random Forest			0.85	0.65	0.71	0.68	0.88
Decision Tree			0.81	0.55	0.69	0.61	0.82
Logistic Regression			0.77	0.48	0.73	0.58	0.83
XGBoost	over	0.70	0.78	0.50	0.74	0.60	0.85
Random Forest			0.85	0.63	0.75	0.68	0.88
Decision Tree			0.80	0.54	0.69	0.61	0.81
Logistic Regression			0.77	0.48	0.77	0.60	0.85
XGBoost	under	0.70	0.74	0.45	0.81	0.58	0.86
Random Forest			0.83	0.59	0.77	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.49	0.78	0.60	0.85
XGBoost	smote	0.70	0.85	0.65	0.70	0.68	0.87
Random Forest			0.85	0.64	0.72	0.68	0.88
Decision Tree			0.81	0.56	0.69	0.62	0.83
Logistic Regression			0.77	0.48	0.73	0.58	0.83

Tabla B-14.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Analysis (Parte 6)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.75	0.78	0.50	0.76	0.60	0.85
Random Forest			0.85	0.63	0.76	0.69	0.88
Decision Tree			0.81	0.55	0.68	0.61	0.81
Logistic Regression			0.77	0.49	0.78	0.60	0.85
XGBoost	under	0.75	0.74	0.45	0.80	0.57	0.86
Random Forest			0.83	0.59	0.77	0.67	0.88
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	smote	0.75	0.85	0.65	0.70	0.67	0.87
Random Forest			0.85	0.64	0.71	0.67	0.88
Decision Tree			0.80	0.54	0.67	0.60	0.81
Logistic Regression			0.77	0.48	0.73	0.58	0.83
XGBoost	over	0.80	0.88	0.71	0.79	0.75	0.93
Random Forest			0.85	0.63	0.75	0.68	0.88
Decision Tree			0.81	0.56	0.66	0.60	0.80
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	under	0.80	0.72	0.43	0.82	0.56	0.86
Random Forest			0.83	0.59	0.78	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	smote	0.80	0.85	0.65	0.70	0.67	0.87
Random Forest			0.85	0.64	0.70	0.67	0.88
Decision Tree			0.81	0.56	0.67	0.61	0.83
Logistic Regression			0.77	0.49	0.72	0.58	0.83

Tabla B-15.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Analysis (Parte 7)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.85	0.88	0.71	0.80	0.75	0.93
Random Forest			0.85	0.63	0.75	0.68	0.88
Decision Tree			0.81	0.55	0.67	0.60	0.80
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	under	0.85	0.72	0.42	0.83	0.56	0.86
Random Forest			0.83	0.59	0.78	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	smote	0.85	0.85	0.64	0.70	0.67	0.87
Random Forest			0.86	0.66	0.71	0.68	0.88
Decision Tree			0.81	0.56	0.68	0.61	0.82
Logistic Regression			0.77	0.49	0.72	0.58	0.83
XGBoost	over	0.90	0.88	0.71	0.79	0.75	0.93
Random Forest			0.85	0.63	0.75	0.69	0.88
Decision Tree			0.81	0.56	0.67	0.61	0.80
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	under	0.90	0.82	0.55	0.87	0.68	0.93
Random Forest			0.83	0.59	0.78	0.67	0.89
Decision Tree			0.81	0.55	0.71	0.62	0.79
Logistic Regression			0.77	0.48	0.78	0.60	0.85
XGBoost	smote	0.90	0.85	0.65	0.70	0.67	0.87
Random Forest			0.85	0.65	0.71	0.68	0.88
Decision Tree			0.81	0.55	0.66	0.60	0.82
Logistic Regression			0.77	0.49	0.72	0.58	0.83

Credit Risk Customers

Tabla B-16.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Customers (Parte 1)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.45	0.67	0.48	0.77	0.59	0.75
Random Forest			0.72	0.54	0.58	0.56	0.76
Decision Tree			0.65	0.44	0.57	0.50	0.67
Logistic Regression			0.69	0.50	0.75	0.60	0.77
XGBoost	under	0.45	0.66	0.46	0.82	0.59	0.72
Random Forest			0.72	0.53	0.59	0.56	0.75
Decision Tree			0.70	0.00	0.00	0.00	0.50
Logistic Regression			0.65	0.45	0.73	0.56	0.75
XGBoost	smote	0.45	0.66	0.46	0.82	0.59	0.72
Random Forest			0.73	0.55	0.60	0.58	0.75
Decision Tree			0.66	0.45	0.52	0.48	0.66
Logistic Regression			0.67	0.47	0.77	0.59	0.76
XGBoost	over	0.50	0.66	0.46	0.82	0.59	0.72
Random Forest			0.71	0.52	0.59	0.56	0.76
Decision Tree			0.66	0.44	0.46	0.45	0.61
Logistic Regression			0.69	0.49	0.77	0.60	0.76
XGBoost	under	0.50	0.64	0.45	0.79	0.57	0.75
Random Forest			0.67	0.47	0.60	0.53	0.74
Decision Tree			0.68	0.47	0.60	0.53	0.69
Logistic Regression			0.66	0.47	0.76	0.58	0.76
XGBoost	smote	0.50	0.50	0.37	0.87	0.51	0.68
Random Forest			0.70	0.50	0.59	0.54	0.73
Decision Tree			0.30	0.30	1.00	0.47	0.50
Logistic Regression			0.69	0.49	0.73	0.59	0.75

Tabla B-17.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Customers (Parte 2)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.55	0.65	0.46	0.76	0.57	0.76
Random Forest			0.71	0.52	0.56	0.54	0.73
Decision Tree			0.67	0.46	0.47	0.47	0.63
Logistic Regression			0.69	0.49	0.77	0.60	0.77
XGBoost	under	0.55	0.64	0.45	0.86	0.59	0.75
Random Forest			0.72	0.53	0.67	0.59	0.75
Decision Tree			0.66	0.46	0.70	0.55	0.72
Logistic Regression			0.66	0.46	0.74	0.57	0.76
XGBoost	smote	0.55	0.56	0.39	0.75	0.51	0.68
Random Forest			0.71	0.53	0.57	0.55	0.72
Decision Tree			0.59	0.36	0.43	0.39	0.56
Logistic Regression			0.69	0.49	0.68	0.57	0.75
XGBoost	over	0.60	0.64	0.45	0.84	0.58	0.74
Random Forest			0.75	0.58	0.58	0.58	0.76
Decision Tree			0.63	0.40	0.43	0.41	0.58
Logistic Regression			0.70	0.50	0.74	0.60	0.77
XGBoost	under	0.60	0.63	0.44	0.79	0.57	0.72
Random Forest			0.67	0.47	0.60	0.53	0.73
Decision Tree			0.65	0.44	0.63	0.52	0.71
Logistic Regression			0.66	0.46	0.71	0.56	0.75
XGBoost	smote	0.60	0.55	0.37	0.73	0.49	0.63
Random Forest			0.70	0.50	0.54	0.52	0.70
Decision Tree			0.52	0.32	0.52	0.40	0.56
Logistic Regression			0.69	0.50	0.60	0.54	0.74

Tabla B-18.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Customers (Parte 3)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.65	0.62	0.43	0.78	0.55	0.70
Random Forest			0.74	0.57	0.58	0.58	0.77
Decision Tree			0.67	0.46	0.48	0.47	0.64
Logistic Regression			0.68	0.48	0.70	0.57	0.77
XGBoost	under	0.65	0.66	0.46	0.82	0.59	0.72
Random Forest			0.71	0.52	0.66	0.58	0.74
Decision Tree			0.69	0.49	0.65	0.56	0.73
Logistic Regression			0.66	0.46	0.73	0.56	0.74
XGBoost	smote	0.65	0.55	0.39	0.82	0.53	0.66
Random Forest			0.69	0.49	0.52	0.51	0.70
Decision Tree			0.67	0.45	0.49	0.47	0.69
Logistic Regression			0.70	0.51	0.59	0.55	0.74
XGBoost	over	0.70	0.56	0.39	0.80	0.52	0.71
Random Forest			0.71	0.52	0.55	0.53	0.75
Decision Tree			0.66	0.44	0.46	0.45	0.63
Logistic Regression			0.69	0.49	0.76	0.60	0.78
XGBoost	under	0.70	0.56	0.40	0.88	0.55	0.73
Random Forest			0.72	0.54	0.66	0.59	0.74
Decision Tree			0.63	0.41	0.54	0.47	0.62
Logistic Regression			0.65	0.45	0.73	0.55	0.75
XGBoost	smote	0.70	0.48	0.33	0.67	0.44	0.58
Random Forest			0.69	0.49	0.53	0.51	0.69
Decision Tree			0.64	0.40	0.41	0.40	0.60
Logistic Regression			0.69	0.50	0.57	0.53	0.73

Tabla B-19.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Customers (Parte 4)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.75	0.56	0.40	0.89	0.55	0.74
Random Forest			0.74	0.58	0.58	0.58	0.77
Decision Tree			0.67	0.46	0.48	0.47	0.65
Logistic Regression			0.67	0.47	0.73	0.57	0.77
XGBoost	under	0.75	0.48	0.36	0.96	0.53	0.73
Random Forest			0.71	0.51	0.65	0.57	0.74
Decision Tree			0.59	0.37	0.52	0.43	0.62
Logistic Regression			0.65	0.45	0.71	0.55	0.73
XGBoost	smote	0.75	0.54	0.35	0.63	0.45	0.62
Random Forest			0.69	0.49	0.51	0.50	0.71
Decision Tree			0.64	0.40	0.40	0.40	0.62
Logistic Regression			0.71	0.53	0.57	0.55	0.73
XGBoost	over	0.80	0.58	0.41	0.84	0.55	0.72
Random Forest			0.72	0.54	0.56	0.55	0.76
Decision Tree			0.66	0.45	0.53	0.48	0.64
Logistic Regression			0.69	0.49	0.78	0.60	0.77
XGBoost	under	0.80	0.52	0.38	0.92	0.54	0.71
Random Forest			0.67	0.47	0.63	0.54	0.74
Decision Tree			0.64	0.42	0.52	0.46	0.61
Logistic Regression			0.65	0.46	0.73	0.56	0.74
XGBoost	smote	0.80	0.52	0.35	0.65	0.45	0.59
Random Forest			0.71	0.52	0.49	0.51	0.70
Decision Tree			0.60	0.39	0.53	0.45	0.60
Logistic Regression			0.71	0.52	0.54	0.53	0.73

Tabla B-20.: Resultados de modelos con diferentes métodos de balanceo y tasas de balanceo en la base de test de Credit Risk Customers (Parte 5)

Model	Method	Ratio	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	over	0.85	0.50	0.37	0.87	0.51	0.73
Random Forest			0.74	0.57	0.56	0.56	0.75
Decision Tree			0.65	0.42	0.42	0.42	0.60
Logistic Regression			0.67	0.47	0.76	0.58	0.77
XGBoost	under	0.85	0.43	0.34	0.96	0.50	0.75
Random Forest			0.68	0.48	0.67	0.56	0.72
Decision Tree			0.60	0.40	0.58	0.47	0.64
Logistic Regression			0.67	0.47	0.70	0.56	0.74
XGBoost	smote	0.85	0.53	0.35	0.65	0.46	0.61
Random Forest			0.67	0.45	0.38	0.41	0.65
Decision Tree			0.62	0.39	0.42	0.40	0.61
Logistic Regression			0.71	0.53	0.47	0.50	0.70
XGBoost	over	0.90	0.63	0.44	0.80	0.57	0.71
Random Forest			0.72	0.54	0.58	0.56	0.76
Decision Tree			0.66	0.44	0.42	0.43	0.60
Logistic Regression			0.67	0.47	0.75	0.58	0.75
XGBoost	under	0.90	0.49	0.36	0.91	0.52	0.72
Random Forest			0.67	0.47	0.66	0.55	0.73
Decision Tree			0.69	0.49	0.59	0.54	0.68
Logistic Regression			0.64	0.44	0.67	0.53	0.73
XGBoost	smote	0.90	0.53	0.35	0.66	0.46	0.56
Random Forest			0.69	0.49	0.38	0.43	0.69
Decision Tree			0.64	0.40	0.37	0.39	0.59
Logistic Regression			0.72	0.53	0.53	0.53	0.72

Bibliografía

[1] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[2] J. J. Espinosa-Zúñiga, "Aplicación de algoritmos random forest y xgboost en una base de solicitudes de tarjetas de crédito," *Ingeniería, investigación y tecnología*, vol. 21, no. 3, 2020.

[3] "Confusion matrix," https://subscription.packtpub.com/book/data/9781838555078/6/ch06lvl1sec34/confusion-matrix.

[4] A. Geron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media, 2017.

[5] B.Chen, "A practical introduction to grid search, random search, and bayes search," https://towardsdatascience.com/a-practical-introduction-to-grid-search-random-search-and-bayes-search-d5580b1d941d, 2021, 20-05-2023.

[6] "Four oversampling and under-sampling methods for imbalanced classification using python," https://medium.com/grabngoinfo/four-oversampling-and-under-sampling-methods-for-imbalanced-classification-using-python-7304a

[7] A. Vijayvargiya, C. Prakash, R. Kumar, S. Bansal, and J. Tavares, "Human knee abnormality detection from imbalanced semg data," *Biomedical Signal Processing and Control*, vol. 66, 04 2021.

[8] M. Dabós, "Credit scoring," *Universidad de Belgrano*, pp. 1–5, 2012.

[9] Y. Liu, "The evaluation of classification models for credit scoring," *Institut für Wirtschaftsinformatik, Georg-August-Universitat Göttingen*, pp. 145–151, 2002.

[10] L. C. Caro Puerta and L. J. Rodas Zuluaga, "Modelos de aprendizaje supervisado para la clasificación de riesgo crediticio en la entidad financiera home credit," 2022.

[11] M. BeltránPascual, "Treatment of unbalanced classes with the cube method in credit scoring problems through data mining," *Cuadernos de Economía*, 2019.

[12] K. Lei, Y. Xie, S. Zhong, J. Dai, M. Yang, and Y. Shen, "Generative adversarial fusion network for class imbalance credit scoring," *Neural Computing and Applications*, vol. 32, pp. 8451–8462, 2020.

[13] V. García Jiménez *et al.*, "Distribuciones de clases no balanceadas: Métricas, análisis de complejidad y algoritmos de aprendizaje," Ph.D. dissertation, Universitat Jaume I, 2010.

[14] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The journal of finance*, vol. 23, no. 4, pp. 589–609, 1968.

[15] C.-L. Huang, M.-C. Chen, and C.-J. Wang, "Credit scoring with a data mining approach based on support vector machines," *Expert systems with applications*, vol. 33, no. 4, pp. 847–856, 2007.

[16] R. M. Pérez Ramón, "Modelo de scoring para la segmentación de clientes morosos usando minería de datos en una empresa de cobranzas del perú," 2021.

[17] J. A. Nelder and R. W. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[18] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, 2014, vol. 326.

[19] A. J. Dobson and A. G. Barnett, *An introduction to generalized linear models*. CRC press, 2018.

[20] L. Cayuela, "Modelos lineales generalizados (glm)," *Materiales de un curso del R del IREC*, 2009.

[21] X. Yan and X. Su, *Linear regression analysis: theory and computing*. World Scientific, 2009.

[22] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[23] F. Cánovas-García, F. Alonso-Sarría, F. Gomariz-Castillo, and F. Oñate-Valdivieso, "Modification of the random forest algorithm to avoid statistical dependence problems when classifying remote sensing imagery," *Computers & Geosciences*, vol. 103, pp. 1–11, 2017.

[24] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2016, pp. 785–794.

[25] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. Sebastopol, CA: O'Reilly Media, 2016.

[26] J. A. Flores, J. L. Malca, L. R. Saldarriaga, and C. S. Román, "Analysis and comparison of machine learning classification models applied to credit approval." in *SIMBig*, 2017, pp. 225–226.

[27] F. P. De la Cruz Flores, "Predicción de adquisición de un préstamo personal bancario a través del canal de televentas utilizando el algoritmo random forest," 2020.

[28] A. González Parga *et al.*, "Modelo de scoring para crédito de consumo en una entidad del sector solidario," 2022.

[29] J. Chen, A. L. Katchova, and C. Zhou, "Agricultural loan delinquency prediction using machine learning methods," *International Food and Agribusiness Management Review*, vol. 24, no. 5, pp. 797–812, 2021.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[31] "Imbalanced-learn documentation: Over-sampling," https://imbalanced-learn.org/stable/, accessed: 27-05-2023.

[32] J. H. Sundjaja, R. Shrestha, and K. Krishan, "Mcnemar and mann-whitney u tests," 2020.

[33] P. E. McKnight and J. Najab, "Mann-whitney u test," *The Corsini encyclopedia of psychology*, pp. 1–1, 2010.

[34] T. W. MacFarland, J. M. Yates, T. W. MacFarland, and J. M. Yates, "Mann–whitney u test," *Introduction to nonparametric statistics for the biological sciences using R*, pp. 103–132, 2016.

[35] M. F. Zibran, "Chi-squared test of independence," *Department of Computer Science, University of Calgary, Alberta, Canada*, vol. 1, no. 1, pp. 1–7, 2007.

[36] A. Ugoni and B. F. Walker, "The chi square test: an introduction," *COMSIG review*, vol. 4, no. 3, p. 61, 1995.

[37] M. L. McHugh, "The chi-square test of independence," *Biochemia medica*, vol. 23, no. 2, pp. 143–149, 2013.

[38] M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath, "Loan default prediction using decision trees and random forest: A comparative study," in *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1. IOP Publishing, 2021, p. 012042.

[39] "Imbalanced — credit approval." [Online]. Available: https://www.kaggle.com/datasets/enesztrk/credit-approval

[40] "Credit risk analysis." [Online]. Available: https://www.kaggle.com/datasets/nanditapore/credit-risk-analysis

[41] "Credit risk customers." [Online]. Available: https://www.kaggle.com/ds/3119852

[42] *Manual de referencia de Python.* Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[43] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.

[44] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, p. 357–362, 2020.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[46] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365

[47] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[48] "Power BI," Microsoft Corporation, 2022, https://powerbi.microsoft.com/.