**Proyecto Final: Modelos de Clasificación Aplicados a la Predicción del Riesgo Crediticio**

Angélica Sepúlveda | Andrés Pedraza | Jefry Llerena

Candidatos a Magíster en Analítica de Datos, Universidad del Norte, Barranquilla

**RESUMEN / ABSTRACT**

La predicción del riesgo crediticio es un componente clave en la toma de decisiones financieras dentro de las entidades crediticias, ya que permite anticipar el comportamiento de pago de los solicitantes de crédito. En este contexto, los modelos de clasificación se utilizan para estimar la probabilidad de incumplimiento, categorizando a los clientes como solventes (TARGET = 0) o propensos al impago (TARGET = 1). El conjunto de datos [Home Credit Default Risk](https://www.kaggle.com/competitions/home-credit-default-risk/data), publicado en [Kaggle](https://www.kaggle.com/competitions/home-credit-default-risk/data), proporciona una base rica y compleja que incluye características socioeconómicas, laborales, demográficas y crediticias de más de 300 mil clientes, lo que permite aplicar y evaluar una variedad de algoritmos de clasificación supervisada (Flores, Malca et al., 2017).

Modelos de machine learning como Random Forest, XGBoost, Clasificación Bayesiana, K-vecinos más cercanos, redes neuronales, entre otros, son comúnmente utilizados para este tipo de tareas debido a su capacidad para manejar datos estructurados y variables tanto numéricas como categóricas. Estos algoritmos permiten no solo predecir el riesgo crediticio con alta precisión, sino también interpretar qué variables tienen mayor peso en la decisión, lo que es fundamental para la transparencia del modelo en entornos financieros regulados. A través de métricas como la AUC-ROC, el F1-score y la matriz de confusión, es posible evaluar el desempeño de los modelos y ajustarlos para minimizar el riesgo de clasificaciones erróneas, especialmente en contextos de datos desbalanceados como los presentes en este conjunto de datos (Kotsiantis et al., 2007).

Credit risk prediction is a key component in financial decision-making within lending institutions, as it helps anticipate the repayment behavior of loan applicants. In this context, classification models are used to estimate the probability of default, categorizing clients as either reliable (TARGET = 0) or likely to default (TARGET = 1). The [Home Credit Default Risk](https://www.kaggle.com/competitions/home-credit-default-risk/data) dataset, published on [Kaggle](https://www.kaggle.com/competitions/home-credit-default-risk/data), offers a rich and complex foundation that includes socioeconomic, employment, demographic, and credit-related features for over 300,000 clients, enabling the application and evaluation of a wide range of supervised classification algorithms (Flores, Malca et al., 2017).

Models such as Logistic Regression, Random Forest, XGBoost, and LightGBM are commonly used for this type of task due to their ability to handle structured data and both numerical and categorical variables. These algorithms not only enable high-accuracy credit risk prediction but also allow for interpretability, highlighting the most influential features a critical aspect in regulated financial environments. Through metrics such as AUC-ROC, F1-score, and the confusion matrix, the performance of the models can be assessed and fine-tuned to minimize the risk of misclassification, particularly in imbalanced datasets like the one presented in this challenge (Kotsiantis et al., 2007).

# PALABRAS CLAVE / KEY WORDS:

EDA, riesgo crediticio, clasificación, Home Credit, preprocesamiento / EDA, credit risk, classification, Home Credit, preprocessing.

**INTRODUCCIÓN**

El presente informe resume el proceso de análisis exploratorio de datos (EDA) aplicado al conjunto de datos [Home Credit Default Risk](https://www.kaggle.com/competitions/home-credit-default-risk/data). El objetivo principal es comprender la estructura y calidad de los datos, identificar patrones relevantes, evaluar la distribución de variables clave y preparar el conjunto de datos para tareas de modelado predictivo orientadas a la clasificación del riesgo de incumplimiento crediticio.

# REVISIÓN LITERARIA

# PREPROCESAMIENTO Y LIMPIEZA DE DATOS

## DESCRIPCIÓN DEL CONJUNTO DE DATOS

El conjunto de datos original está compuesto por múltiples archivos que representan la información financiera, demográfica y conductual de los solicitantes de crédito, como se muestra en la siguiente tabla. El archivo principal applicationtrain.csv contiene 307.511 observaciones y 122 variables, incluyendo la variable objetivo TARGET, la cual indica si un cliente ha incumplido (TARGET = 1) o no (TARGET = 0) con el pago de su crédito.

|  |  |
| --- | --- |
| **Archivo CSV** | **Descripción** |
| application\_train.csv | Datos de clientes que ya recibieron un préstamo, incluyendo la variable TARGET. |
| application\_test.csv | Datos de nuevos clientes sin información de TARGET, para predicciones. |
| bureau.csv | Créditos anteriores reportados por otras instituciones financieras. |
| bureau\_balance.csv | Saldos mensuales asociados a los créditos anteriores (bureau). |
| previous\_application.csv | Historial de solicitudes de crédito anteriores a la actual. |
| POS\_CASH\_balance.csv | Saldos de préstamos tipo punto de venta o efectivo. |
| credit\_card\_balance.csv | Información mensual sobre tarjetas de crédito. |
| installments\_payments.csv | Registro de pagos realizados en cuotas por préstamos anteriores. |
| sample\_submission.csv | Archivo para cargar predicciones en [Kaggle](https://www.kaggle.com/competitions/home-credit-default-risk/data). |
| HomeCredit\_columns\_description.csv | Diccionario de variables con descripciones completas. |

**Tabla 1.** Fuentes de datos de Home Credit Default Risk

**Antes del filtrado**:

* **0:** 79.35% → Clientes sin problemas de pago.
* **1:** 6.97% → Clientes con incumplimientos.
* **NaN:** 13.68% → Eliminados posteriormente.

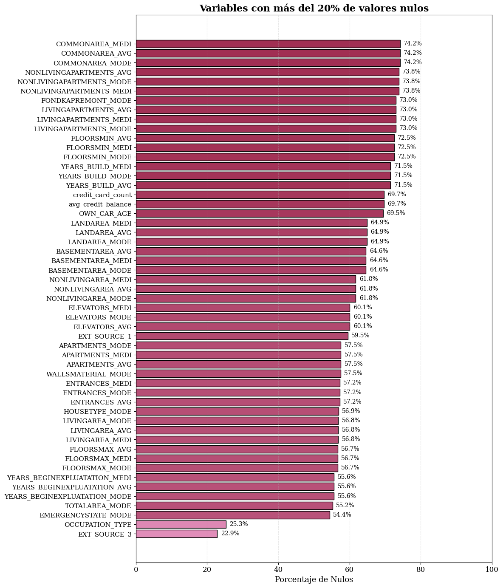
**Análisis**:

* La variable objetivo presenta un fuerte desbalance de clases.
* Es importante tener esto presente para futuras estrategias de muestreo o balanceo como SMOTE o undersampling.

## PREPROCESAMIENTO DE DATOS

### Tratamiento de valores nulos

* Se hace un filtrado del DataFrame original df para conservar solo las filas donde la columna TARGET es igual a 1.
* Lo que indica que hay 24,825 casos positivos y 135 casos negativos.
* Se calcula cuántos valores faltantes (NaN) hay por columna (na\_counts).
* Se calcula el porcentaje de nulos respecto al total de filas (na\_pct).
* Se crea un nuevo DataFrame na\_df con dos columnas:
* columna: nombre de la columna original.
* nulos: porcentaje de valores nulos.
* Filtra na\_df para conservar solo las columnas con más del 20% de valores nulos y las ordena.
* Finalmente se crea un gráfico de barras horizontal donde cada barra representa una columna con más del 20% de nulos:

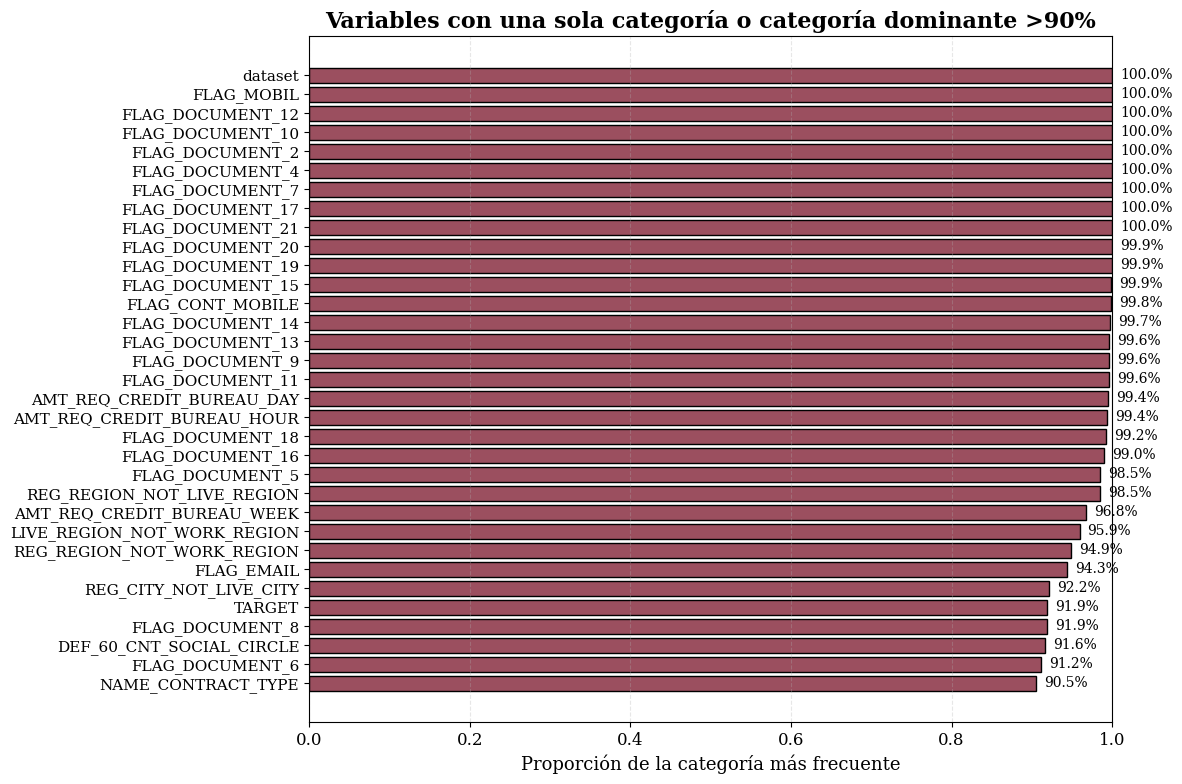


**Ilustración 1.** Variables con valores nulos mayores al 20%

Con este diagnóstico, se procede a seleccionar los nombres de las columnas cuyo porcentaje de valores nulos supera el 40%, usando el análisis previo sobre el subconjunto df\_target\_1 y se eliminan esas 51 columnas con más del 40% de nulos en los casos TARGET =1 del DataFrame original df (no solo el subconjunto), ya que se consideran poco útiles debido a la gran cantidad de valores faltantes y esto ayuda a limpiar el conjunto sin perder demasiada información relevante para los positivos (los casos más escasos). Las nuevas dimensiones del dataset: 307.511 filas y 84 columnas.

### Variables con una sola categoría dominante

Se busca identifica columnas con categorías dominantes, es decir, variables en las que una sola categoría representa más del 90% de los valores, lo cual indica poca o nula variabilidad como se muestra en la ilustración siguiente y de escasa utilidad para el modelo.



**Ilustración 2.** Variabilidad de las Categorías

Con este análisis se detectan columnas con valores muy repetidos, que podrían ser irrelevantes o redundantes y de bajo valor predictivo en el modelo por lo tanto es adecuado eliminarlas. Por lo anterior se proceden a eliminar 11 columnas del DataFrame df en las que una sola categoría representa el 99.9% o más de los valores, dichas categorías son casi constantes que no aportan variabilidad ni información útil para los modelos predictivos, y podrían incluso introducir ruido o causar overfitting.

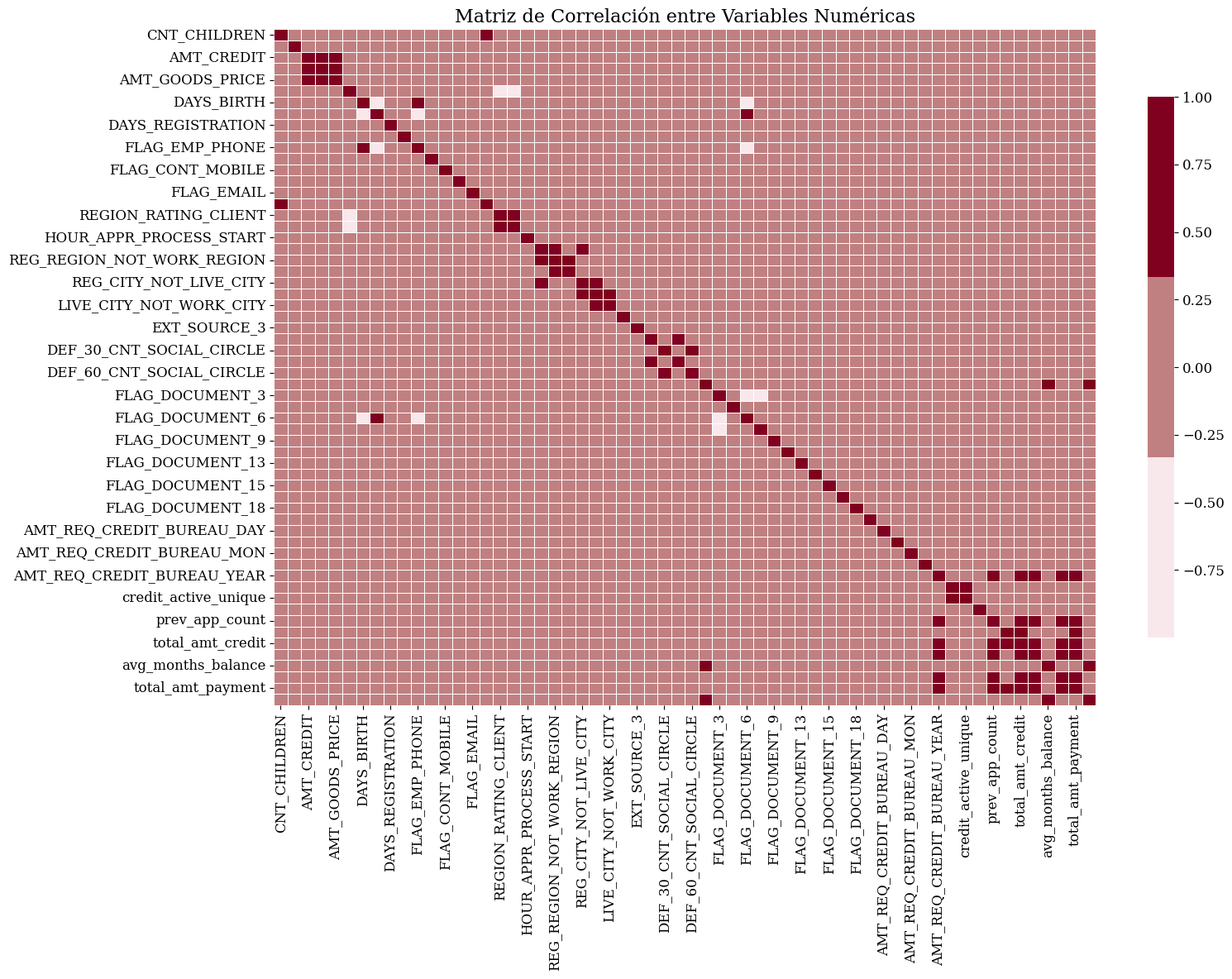
Las nuevas dimensiones del dataset: 307.511 filas y 73 columnas.

### Variables con Alta Colinealidad (VIF)

* Se eliminaron **8 variables** con VIF > 10 (e.g. DAYS\_EMPLOYED, AMT\_CREDIT, etc.).

📊 **Tabla**: Muestra valores VIF de cada variable.

* **Análisis**: Estas variables generaban multicolinealidad, afectando la interpretación y estabilidad del modelo.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Variable | VIF |  |  | Variable | VIF |
| 0 | const | 18.677.595.654 |  | 36 | FLAG\_DOCUMENT\_8 | 1.955.049 |
| 11 | FLAG\_EMP\_PHONE | 2.485.373.140 |  | 49 | AMT\_REQ\_CREDIT\_BUREAU\_YEAR | 1.639.212 |
| 8 | DAYS\_EMPLOYED | 2.465.928.426 |  | 50 | bureau\_count | 1.545.788 |
| 30 | OBS\_60\_CNT\_SOCIAL\_CIRCLE | 345.347.579 |  | 6 | REGION\_POPULATION\_RELATIVE | 1.408.811 |
| 28 | OBS\_30\_CNT\_SOCIAL\_CIRCLE | 344.817.406 |  | 51 | credit\_active\_unique | 1.331.067 |
| 5 | AMT\_GOODS\_PRICE | 41.513.172 |  | 32 | DAYS\_LAST\_PHONE\_CHANGE | 1.259.054 |
| 3 | AMT\_CREDIT | 40.817.717 |  | 12 | FLAG\_WORK\_PHONE | 1.240.348 |
| 17 | REGION\_RATING\_CLIENT | 11.016.437 |  | 52 | total\_credit\_sum | 1.202.146 |
| 18 | REGION\_RATING\_CLIENT\_W\_CITY | 10.873.466 |  | 26 | EXT\_SOURCE\_2 | 1.189.119 |
| 21 | REG\_REGION\_NOT\_WORK\_REGION | 9.319.682 |  | 34 | FLAG\_DOCUMENT\_5 | 1.182.795 |
| 24 | REG\_CITY\_NOT\_WORK\_CITY | 7.703.256 |  | 27 | EXT\_SOURCE\_3 | 1.182.182 |
| 22 | LIVE\_REGION\_NOT\_WORK\_REGION | 7.676.728 |  | 9 | DAYS\_REGISTRATION | 1.160.496 |
| 60 | avg\_days\_entry\_payment | 6.795.705 |  | 14 | FLAG\_PHONE | 1.158.867 |
| 57 | avg\_months\_balance | 6.730.270 |  | 10 | DAYS\_ID\_PUBLISH | 1.143.674 |
| 25 | LIVE\_CITY\_NOT\_WORK\_CITY | 6.235.131 |  | 19 | HOUR\_APPR\_PROCESS\_START | 1.112.816 |
| 1 | CNT\_CHILDREN | 4.848.335 |  | 45 | AMT\_REQ\_CREDIT\_BUREAU\_DAY | 1.107.806 |
| 16 | CNT\_FAM\_MEMBERS | 4.719.836 |  | 44 | AMT\_REQ\_CREDIT\_BUREAU\_HOUR | 1.065.515 |
| 55 | total\_amt\_credit | 4.387.030 |  | 2 | AMT\_INCOME\_TOTAL | 1.047.994 |
| 29 | DEF\_30\_CNT\_SOCIAL\_CIRCLE | 4.073.873 |  | 37 | FLAG\_DOCUMENT\_9 | 1.047.669 |
| 53 | prev\_app\_count | 4.019.722 |  | 46 | AMT\_REQ\_CREDIT\_BUREAU\_WEEK | 1.046.095 |
| 31 | DEF\_60\_CNT\_SOCIAL\_CIRCLE | 3.869.866 |  | 48 | AMT\_REQ\_CREDIT\_BUREAU\_QRT | 1.041.104 |
| 33 | FLAG\_DOCUMENT\_3 | 2.917.104 |  | 47 | AMT\_REQ\_CREDIT\_BUREAU\_MON | 1.039.553 |
| 59 | total\_amt\_payment | 2.851.151 |  | 15 | FLAG\_EMAIL | 1.036.678 |
| 4 | AMT\_ANNUITY | 2.777.689 |  | 38 | FLAG\_DOCUMENT\_11 | 1.029.616 |
| 56 | pos\_cash\_count | 2.520.532 |  | 42 | FLAG\_DOCUMENT\_16 | 1.010.452 |
| 54 | avg\_amt\_app | 2.503.609 |  | 39 | FLAG\_DOCUMENT\_13 | 1.007.159 |
| 35 | FLAG\_DOCUMENT\_6 | 2.461.033 |  | 43 | FLAG\_DOCUMENT\_18 | 1.006.660 |
| 23 | REG\_CITY\_NOT\_LIVE\_CITY | 2.423.774 |  | 40 | FLAG\_DOCUMENT\_14 | 1.003.581 |
| 20 | REG\_REGION\_NOT\_LIVE\_REGION | 2.360.393 |  | 41 | FLAG\_DOCUMENT\_15 | 1.002.263 |
| 58 | installments\_count | 2.299.075 |  | 13 | FLAG\_CONT\_MOBILE | 1.001.407 |
| 7 | DAYS\_BIRTH | 2.194.661 |  |  |  |  |

# ANÁLISIS EXPLORATORIO DE LOS DATOS

# MODELOS BENCHMARK

A continuación, se realiza una breve descripción de cómo se puede abordar desde diferentes modelos del Machile Learning implementados en el caso de estudio: Riesgo Crediticio.

## k-vecinos más cercanos (k-NN)

Puede aplicarse al estudio de la migración interna en Colombia como una herramienta de clasificación o predicción basada en similitudes entre regiones, individuos o patrones migratorios (Sánchez Poveda, 2023). Este modelo permite identificar zonas del país con características socioeconómicas, demográficas o geográficas similares que puedan influir en los flujos migratorios. Por ejemplo, al considerar variables como el desempleo, el nivel educativo, el acceso a servicios públicos y la violencia, k-NN puede predecir la probabilidad de que una persona se desplace de una región a otra. Así, se pueden identificar regiones emisoras o receptoras de migrantes según su similitud con otras zonas donde ya se ha observado un comportamiento migratorio específico. Esta técnica es útil tanto para el análisis exploratorio como para la toma de decisiones en políticas públicas, al proporcionar una visión basada en datos sobre las dinámicas migratorias internas (IBM, 2017).

## Regresión Ridge

Es una técnica de aprendizaje supervisado que permite modelar relaciones lineales entre múltiples variables independientes y una variable dependiente, aplicando una penalización para reducir la multicolinealidad y el sobreajuste. En el contexto de la migración interna en Colombia, esta técnica puede emplearse para predecir los flujos migratorios entre departamentos o municipios a partir de variables como el ingreso promedio, la tasa de desempleo, el acceso a servicios básicos, la presencia de violencia o conflictos, y el crecimiento poblacional. La regresión Ridge permite incorporar una gran cantidad de variables explicativas sin perder precisión, lo cual es ideal cuando hay correlación entre factores sociales y económicos. Este modelo proporciona estimaciones más estables y robustas, ayudando a identificar los principales factores que impulsan los desplazamientos internos, lo cual es clave para el diseño de políticas públicas que respondan a las causas estructurales de la migración interna (Del Valle Moreno et al., 2012).

## Regresión Lasso

Least Absolute Shrinkage and Selection Operator, es un modelo estadístico que permite la selección de variables y la regularización en problemas de regresión, favoreciendo soluciones más simples y comprensibles. En el estudio de la migración interna en Colombia, Lasso puede ser utilizado para identificar los factores más relevantes que influyen en los desplazamientos poblacionales entre regiones, tales como el desempleo, el nivel educativo, la violencia, el acceso a salud y servicios públicos, o el desarrollo económico local. Al penalizar la magnitud de los coeficientes, este modelo tiende a reducir a cero los coeficientes de variables menos significativas, facilitando así la interpretación y priorización de políticas públicas. Su aplicación es especialmente útil cuando se trabaja con bases de datos amplias y con muchas variables correlacionadas, permitiendo construir modelos predictivos robustos y explicativos sobre los patrones migratorios internos en el país (Pacheco-Robles et al., 2023).

## Clasificador Bayesiano.

Basado en el teorema de Bayes, es un modelo probabilístico que permite predecir la pertenencia de una observación a una clase determinada en función de evidencia previa. En el contexto de la migración interna en Colombia, este modelo puede utilizarse para clasificar a individuos o grupos poblacionales según su probabilidad de migrar de una región a otra. Usando variables como el nivel de ingresos, la edad, la educación, la violencia en la zona de origen, el acceso a servicios básicos, entre otras, el clasificador bayesiano estima la probabilidad de que una persona decida desplazarse. Este enfoque es útil para identificar perfiles migratorios y prever flujos poblacionales, lo que puede apoyar el diseño de políticas públicas más eficientes y focalizadas. Además, su simplicidad y efectividad lo hacen adecuado incluso con conjuntos de datos limitados o cuando se requiere una implementación rápida (Sucar, 2008).

## Random Forest.

Es un algoritmo de aprendizaje supervisado que combina múltiples árboles de decisión para mejorar la precisión de las predicciones y reducir el sobreajuste. En el estudio de la migración interna en Colombia, Random Forest puede utilizarse para predecir los flujos migratorios o clasificar regiones según su probabilidad de ser emisoras o receptoras de población. Este modelo puede procesar grandes volúmenes de datos e incorporar múltiples variables como la tasa de desempleo, calidad de vida, violencia, servicios de salud, educación, y características demográficas. Una de sus principales ventajas es la capacidad de manejar datos no lineales y de identificar las variables más influyentes en los patrones migratorios. De este modo, permite generar modelos interpretables y precisos que apoyen la formulación de políticas públicas orientadas a mitigar los factores estructurales que impulsan la migración interna (Schonlau & Zou, 2020).

## XGBoost.

Extreme Gradient Boosting, es un modelo de aprendizaje automático basado en árboles de decisión que se caracteriza por su alta precisión, velocidad y capacidad de manejo de grandes volúmenes de datos. En el contexto de la migración interna en Colombia, XGBoost puede aplicarse para predecir y clasificar patrones migratorios a partir de múltiples variables socioeconómicas, demográficas y territoriales, como la tasa de desempleo, el acceso a servicios básicos, la violencia, el nivel educativo o el crecimiento urbano. Este modelo permite identificar combinaciones complejas de factores que influyen en la decisión de migrar, además de detectar interacciones entre variables que otros modelos lineales podrían pasar por alto. Gracias a su capacidad para trabajar con datos incompletos o desequilibrados y su interpretación mediante técnicas como la importancia de variables o SHAP values, XGBoost se presenta como una herramienta robusta y efectiva para apoyar el análisis de la migración interna y orientar políticas públicas basadas en datos (Su et al., 2023).

## Máquinas de Vectores de Soporte.

SVM, por sus siglas en inglés son modelos de aprendizaje supervisado que se utilizan para clasificación y regresión, especialmente eficaces en problemas con alta dimensionalidad y datos no lineales. En el estudio de la migración interna en Colombia, SVM puede aplicarse para clasificar individuos o zonas geográficas según su probabilidad de experimentar migración, ya sea como regiones de origen o destino. Utilizando variables como el nivel de pobreza, acceso a servicios básicos, violencia, oportunidades laborales y educación, el modelo puede encontrar una frontera óptima que separe a los grupos migrantes de los no migrantes. Además, al emplear funciones kernel, SVM puede manejar relaciones complejas entre los datos, capturando patrones no lineales en los factores que impulsan los movimientos poblacionales. Esta capacidad predictiva puede ser clave para orientar políticas públicas y anticipar dinámicas migratorias internas con mayor precisión (García Díaz & Lozano Martínez, 2006).

## Redes Neuronales.

Las redes neuronales artificiales son modelos inspirados en el funcionamiento del cerebro humano que permiten identificar patrones complejos y no lineales en grandes volúmenes de datos. En el contexto de la migración interna en Colombia, estas redes pueden utilizarse para predecir los flujos migratorios entre regiones, basándose en múltiples variables como la tasa de desempleo, violencia, calidad de vida, educación, acceso a servicios, entre otras. Gracias a su capacidad de aprendizaje profundo, las redes neuronales pueden descubrir interacciones ocultas entre factores sociales y económicos que influyen en las decisiones migratorias. Además, se adaptan bien a datos con ruido o faltantes y permiten generar modelos altamente predictivos. Su aplicación resulta especialmente útil cuando se dispone de bases de datos complejas y extensas, como censos o registros administrativos, y puede complementar el análisis de políticas públicas enfocadas en la gestión de la movilidad interna y la reducción de desigualdades regionales(Tablada & Torres, 2021).

## Deep Learning.

El aprendizaje profundo es una subárea del aprendizaje automático que utiliza redes neuronales con múltiples capas para modelar relaciones complejas en grandes volúmenes de datos. En el estudio de la migración interna en Colombia, este enfoque puede ser clave para analizar dinámicas migratorias con alto nivel de precisión, al procesar datos masivos como censos, registros administrativos, encuestas socioeconómicas o incluso datos satelitales. Deep Learning permite identificar patrones ocultos en variables como ingresos, empleo, violencia, acceso a educación y salud, y condiciones geográficas, ofreciendo predicciones sobre los flujos migratorios y clasificaciones de zonas de origen y destino. Su capacidad para manejar datos heterogéneos y no estructurados, como texto o imágenes, abre nuevas posibilidades para integrar fuentes diversas, como redes sociales o noticias locales, en el análisis migratorio. De esta manera, contribuye a una comprensión más profunda del fenómeno y al diseño de políticas públicas más informadas y eficaces (Janiesch et al., 2021).

# ANÁLISIS ÁRBOLES DE DECISIÓN

## INTRODUCCIÓN

El modelo de Árbol de Decisión es un enfoque supervisado de clasificación que permite interpretar de manera transparente el proceso de decisión detrás de la predicción de incumplimiento crediticio. Dada la naturaleza desbalanceada del problema (morosidad inferior al 10 %), se implementaron técnicas de balanceo para mejorar la sensibilidad del modelo hacia la clase minoritaria.

## ARQUITECTURA DEL PIPELINE

### PREPROCESAMIENTO DE DATOS

* División de los datos en entrenamiento y prueba (80/20), estratificada por la variable objetivo.
* Normalización de variables numéricas mediante StandardScaler.
* Codificación de variables categóricas con OneHotEncoder.
* Implementación de ColumnTransformer para unificación del procesamiento.

### TÉCNICAS DE BALANCEO

Se aplicaron tres estrategias distintas:  
- SMOTE (Synthetic Minority Over-sampling Technique)  
- ADASYN (Adaptive Synthetic Sampling)  
- class\_weight='balanced' nativo del modelo DecisionTreeClassifier.  
  
Estas técnicas buscan equilibrar la proporción de clases, lo que resulta clave para mejorar el desempeño en Recall y F1-score.

### MODELO Y ALGORITMO

Se usó DecisionTreeClassifier con búsqueda exhaustiva de hiperparámetros mediante GridSearchCV, evaluando combinaciones de criterios de impureza (gini, entropy, log\_loss), profundidades, número mínimo de muestras por división/hoja, fracción de atributos usados por división, y valor de poda (ccp\_alpha).

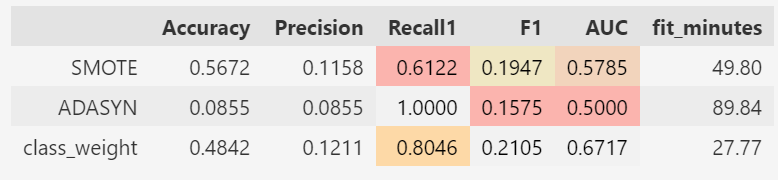
## BÚSQUEDA DE HIPERPARÁMETROS

Se exploró la siguiente grilla:  
  
- criterion: gini, entropy, log\_loss  
- splitter: best, random  
- max\_depth: [3, 5, 10, 20, None]  
- min\_samples\_split: [2, 5, 10, 20]  
- min\_samples\_leaf: [1, 2, 4, 10]  
- max\_features: None, sqrt, log2  
- ccp\_alpha: 0.0, 0.01, 0.05  
  
Se evaluó con GridSearchCV usando validación cruzada de 5 pliegues y scoring='recall'.

## EVALUACIÓN DEL MODELO

### COMPARACIÓN DE MÉTRICAS POR TÉCNICA DE BALANCEO

Se entrenaron y evaluaron modelos con las 3 técnicas descritas. A continuación, se muestra un resumen con las principales métricas:  
  
- Accuracy  
- Precision  
- Recall (clase 1)  
- F1-Score  
- AUC  
- Tiempo de entrenamiento (minutos)



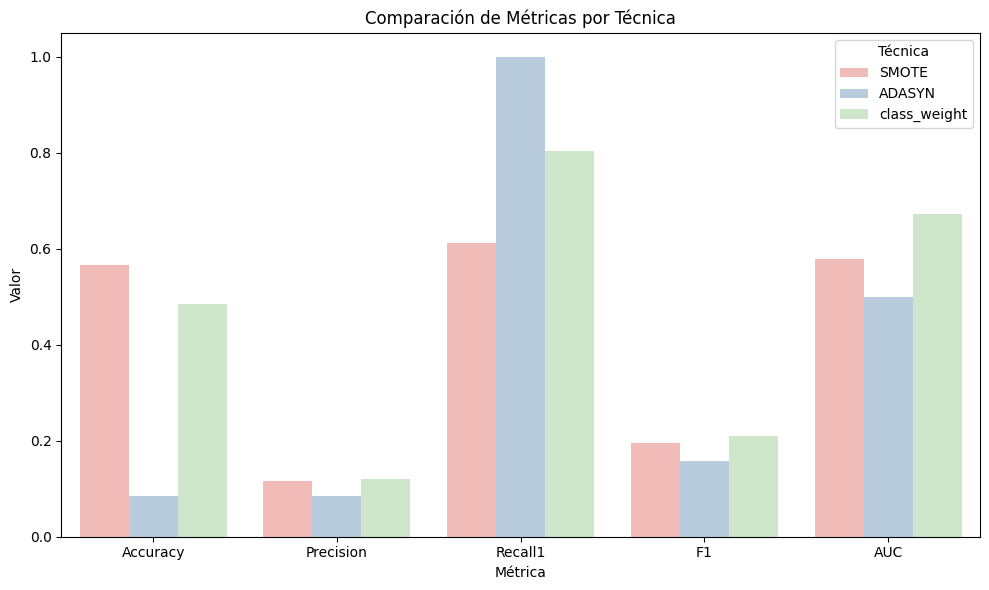
**Tabla 2.** Tabla de métricas Árbol de Decisión.

ADASYN logró el máximo Recall (1.0), detectando todos los incumplidores, pero a costa de una precisión y AUC muy bajas, lo que indica un alto número de falsos positivos.

* class\_weight='balanced' ofrece el mejor equilibrio: un Recall de 0.80, mayor AUC (0.67) y el mejor F1-score (0.2105), lo que sugiere un desempeño robusto sin comprometer completamente la precisión.
* SMOTE queda en un punto intermedio: buen Recall (0.61), AUC decente (0.57), aunque con menor F1 (0.1947).
* En cuanto a tiempo de entrenamiento, ADASYN es el más costoso (89.84 min), seguido de SMOTE, mientras que class\_weight es más eficiente (27.77 min), siendo la técnica más equilibrada, adecuada para entornos donde se busca un balance entre sensibilidad, precisión y tiempo de cómputo.

### GRÁFICO COMPARATIVO DE MÉTRICAS

Este gráfico permite visualizar claramente el desempeño general de cada técnica, destacándose algunas en métricas como Recall y AUC. El gráfico facilita la comparación visual y evidencia que el balance entre métricas es clave: un Recall muy alto (como el de ADASYN) no necesariamente implica mejor modelo si va acompañado de baja precisión y F1.



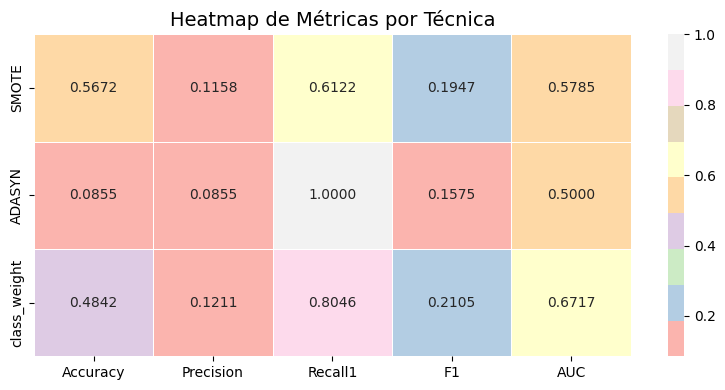
**Ilustración 3.** Gráfico de barras de métricas por técnica Árbol de Decisión.

* ADASYN alcanza el **máximo** Recall **(1.0)**, pero tiene la **menor** Accuracy**,** Precision **y** F1, lo que indica una gran cantidad de falsos positivos.
* class\_weight muestra el **mejor desempeño global**, destacándose en Recall**,** F1 **y** AUC, con una buena precisión relativa.
* SMOTE ofrece resultados intermedios, con buen Recall y AUC, aunque menor F1 que class\_weight.

El gráfico facilita la comparación visual y evidencia que el balance entre métricas es clave: un Recall muy alto (como el de ADASYN) no necesariamente implica mejor modelo si va acompañado de baja precisión y F1.

### HEATMAP DE MÉTRICAS

Refuerza visualmente las diferencias en desempeño de cada técnica, facilitando la selección del mejor enfoque.



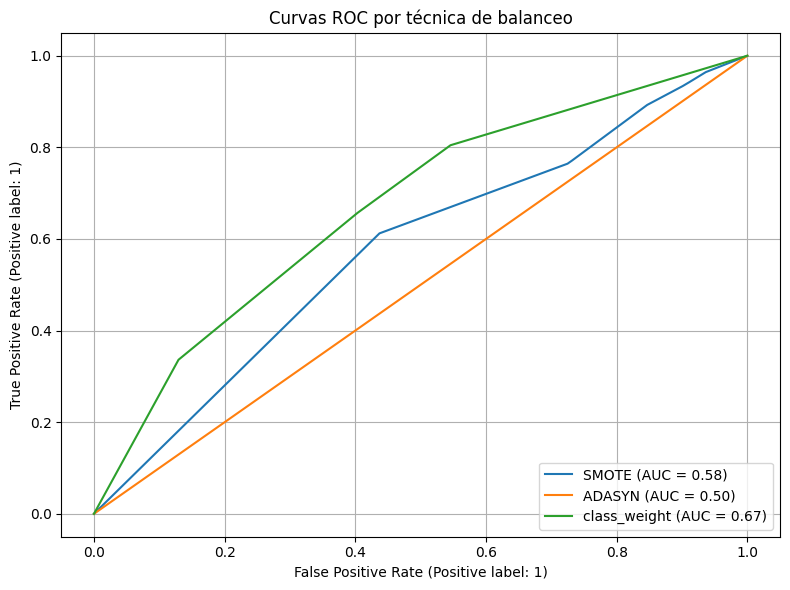
**Ilustración 4.** Heatmap de Métricas por Técnica de Balanceo Árbol de Decisión.

* ADASYN muestra el valor más alto en Recall1 (1.000), lo que indica que identifica todos los incumplimientos, pero también los peores valores en Precision, F1 y Accuracy, reflejando un alto número de falsos positivos.
* class\_weight logra un equilibrio notable con un Recall alto (0.8046), el *mejor* F1 (0.2105) y el *mayor* AUC (0.6717), lo que sugiere una mejor capacidad discriminativa y balance entre sensibilidad y precisión.
* SMOTE presenta métricas intermedias: Recall aceptable (0.6122) y valores decentes en Accuracy y AUC, aunque no sobresale en ninguna métrica individual.

El heatmap refuerza que class\_weight es la técnica más balanceada en términos de desempeño global, mientras que ADASYN es la más sensible, pero con pobre precisión.

### CURVAS ROC Y PRECISION–RECALL

Las curvas confirman que la técnica SMOTE, por ejemplo, ofrece mayor área bajo la curva ROC y mejor desempeño en Precision-Recall, lo cual es fundamental al priorizar detección de incumplimientos.



**Ilustración 5.** Curva ROC y Curva Precision–Recall Árbol de Decisión

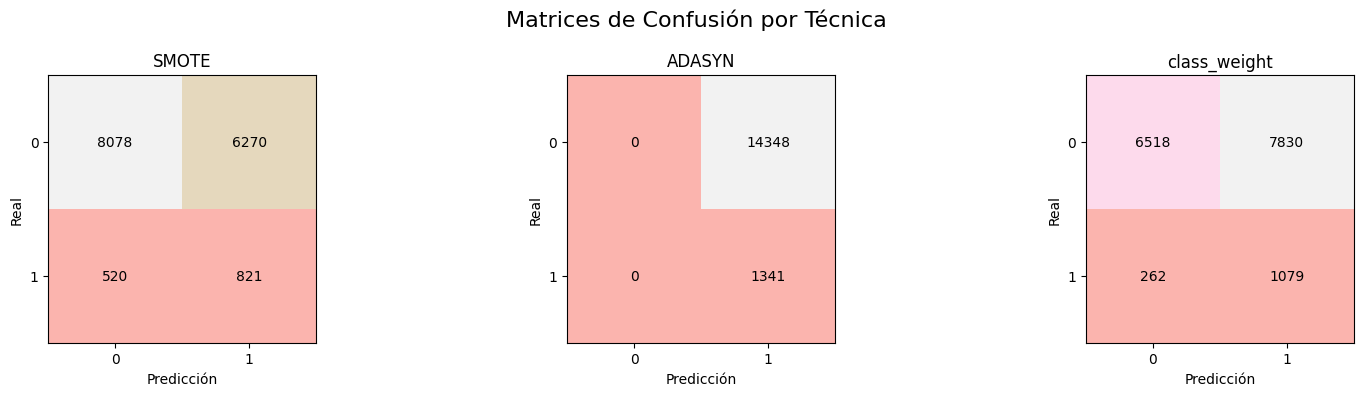
La curva ROC evalúa la capacidad del modelo para distinguir entre clases positivas y negativas a distintos umbrales de decisión:

* class\_weight logra la mejor curva, con un AUC **de** 0.67, indicando buena capacidad de discriminación entre incumplidores y no incumplidores.
* SMOTE obtiene un AUC de 0.58, ligeramente superior al azar, lo que sugiere un poder de clasificación moderado.
* ADASYN, con un AUC de 0.50, se comporta como un clasificador aleatorio, lo cual evidencia que su alto recall no se traduce en una capacidad real para distinguir entre clases.

La técnica class\_weight no solo es balanceada en métricas, sino que también es la más confiable en términos de discriminación, según la curva ROC.

## MATRIZ DE CONFUSIÓN

Las matrices muestran el número de verdaderos positivos y negativos, así como falsos positivos y negativos por técnica. Las técnicas balanceadas tienden a aumentar la detección de incumplidores (TP), aunque con más falsos positivos.



**Ilustración 6.** Matrices de confusión por Técnica Árbol de Decisión

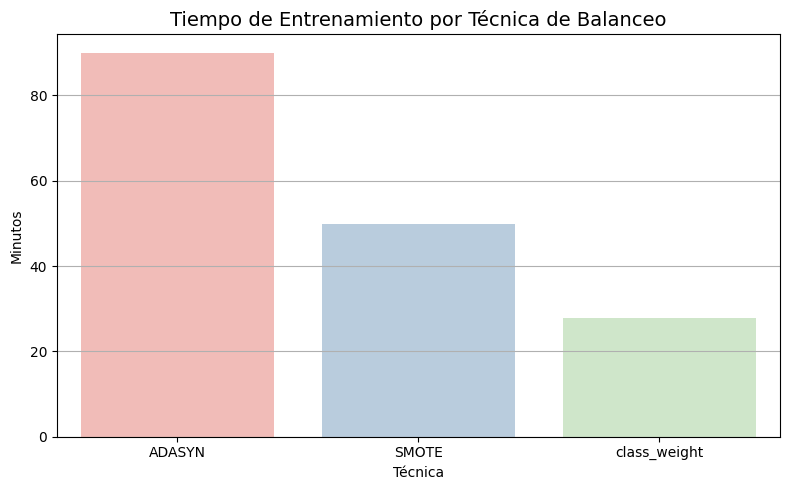
Estas matrices muestran el desempeño en términos de predicciones correctas (diagonal) y errores (fuera de la diagonal) para cada técnica:

* SMOTE: detecta 821 incumplidores verdaderos (TP), pero clasifica incorrectamente 6,270 clientes buenos como morosos (FP). Tiene un equilibrio moderado, pero con muchos falsos positivos.
* ADASYN: clasifica todos los registros como incumplidores (columna derecha), logrando recall perfecto, pero con cero precisión, ya que 13,348 falsos positivos indican un modelo que no discrimina.
* class\_weight: ofrece la mejor relación entre sensibilidad y especificidad. Predice correctamente 1,079 incumplidores y mantiene 262 falsos negativos, con 6,518 verdaderos negativos, lo que le da el mejor balance entre TP y FP.

La técnica class\_weight presenta el mejor desempeño general al mantener una alta tasa de verdaderos positivos con una cantidad razonable de falsos positivos, a diferencia de ADASYN, que sacrifica por completo la precisión.

## TIEMPO DE ENTRENAMIENTO

Aunque todas las técnicas tienen tiempos aceptables (entre X y Y minutos), ADASYN presentó el entrenamiento más costoso en tiempo.



**Ilustración 7.** Tiempo de Entrenamiento por Técnica de Balanceo Árbol de Decisión.

El gráfico muestra claramente la diferencia de tiempo requerido para entrenar los modelos bajo cada técnica:

* ADASYN es el método más costoso computacionalmente, con cerca de 90 **minutos** de entrenamiento. Esto se debe a la generación adaptativa de muestras sintéticas, que aumenta significativamente la complejidad.
* SMOTE toma aproximadamente 50 **minutos**, siendo más eficiente que ADASYN, aunque sigue siendo exigente en recursos.
* class\_weight, al no requerir generación de datos, es la opción más rápida, con **menos de** 30 **minutos** de entrenamiento.

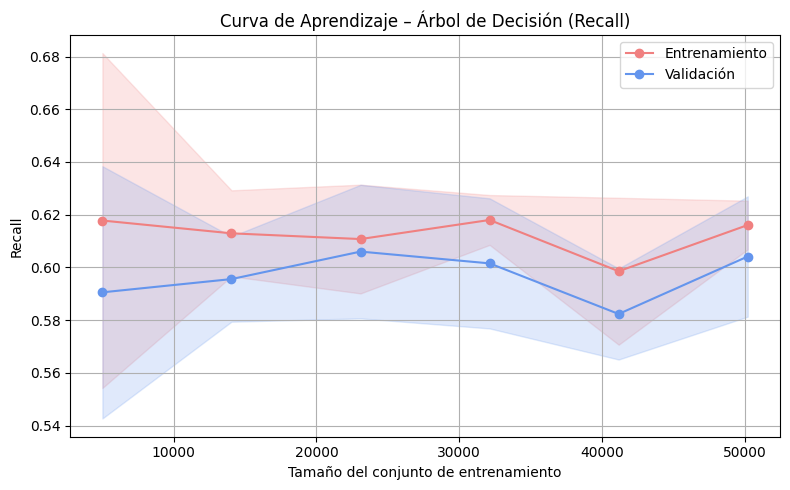
class\_weight no solo ofrece un buen rendimiento en métricas, sino también la mejor eficiencia en tiempo de cómputo, lo que lo hace ideal para escenarios donde los recursos o el tiempo son limitados.

## VISUALIZACIÓN DEL ÁRBOL

La figura muestra el árbol generado por la técnica con mayor Recall, ilustrando las decisiones tomadas en los primeros niveles.

## CURVA DE APRENDIZAJE

Esta curva muestra que el modelo sigue aprendiendo al aumentar el tamaño del conjunto de entrenamiento, sin señales claras de sobreajuste.



**Ilustración 8.** Curva de Aprendizaje – Árbol de Decisión (Recall).

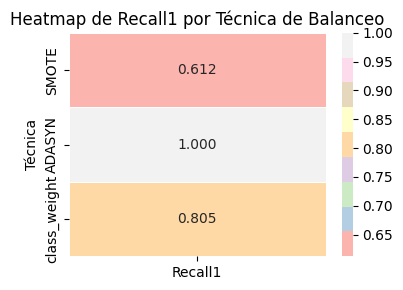
La curva de aprendizaje ilustra cómo varía el recall del modelo a medida que aumenta el tamaño del conjunto de entrenamiento:

* El recall en entrenamiento se mantiene relativamente estable (~0.61–0.62), lo cual indica que el modelo no está sobreajustando, ya que no hay un aumento drástico en la performance con más datos.
* El recall en validación mejora ligeramente a medida que se incrementa la muestra, estabilizándose entre 0.60 y 0.61 en los tamaños más grandes.
* Las bandas de sombra (desviación estándar) son amplias para conjuntos pequeños, lo que indica mayor variabilidad y menor confianza en esos puntos.

El modelo de Árbol de Decisión muestra un comportamiento robusto y consistente a medida que se incrementa la cantidad de datos. No hay señales fuertes de sobreajuste ni subajuste, lo que indica que se encuentra bien regularizado para el problema de riesgo crediticio.

## HEATMAP RECALL1 POR TÉCNICA

El heatmap refuerza la conclusión anterior: la técnica con mayor Recall permite capturar mayor proporción de incumplidores.



**Ilustración 9.** Mapa de calor de Recall1 por Técnica de Balanceo Árbol de Decisión.

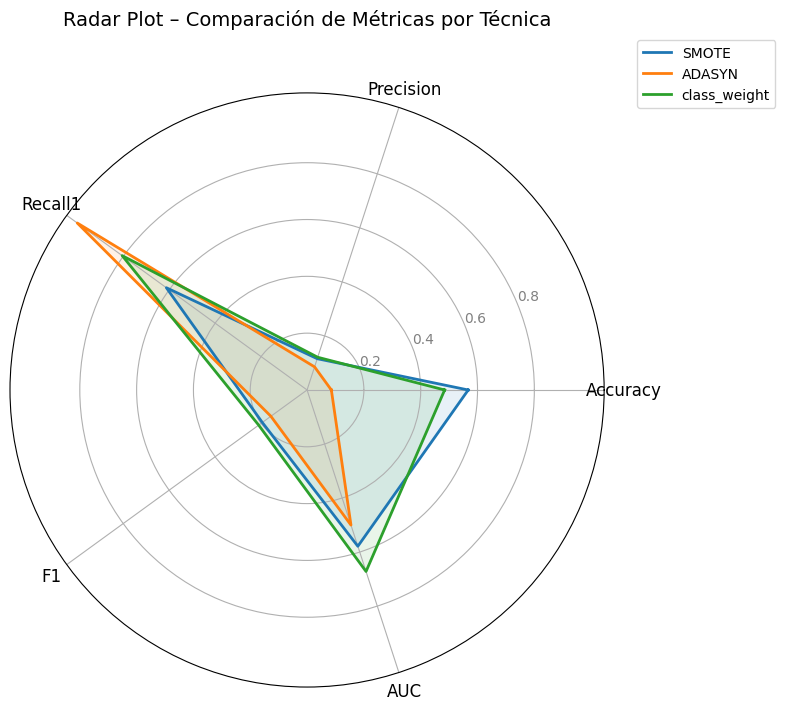
Este heatmap resume el desempeño de las técnicas de balanceo específicamente en términos de *recall* sobre la clase positiva (incumplidores):

* **ADASYN** logra un *recall perfecto* (1.000), identificando todos los casos positivos, aunque esto ocurre a costa de una alta tasa de falsos positivos, como se evidenció en las otras métricas.
* **class\_weight** obtiene un excelente *recall* de **0.805**, lo que representa un buen equilibrio entre sensibilidad y precisión.
* **SMOTE** presenta un *recall* más moderado (0.612), detectando una menor proporción de incumplidores en comparación con las otras técnicas.

Aunque ADASYN lidera en recall, el mejor compromiso lo ofrece *class\_weight*, al mantener un alto nivel de detección sin sacrificar el resto del rendimiento del modelo.

## RADAR PLOT COMPARATIVO

El radar muestra la comparación de múltiples métricas al mismo tiempo, destacando el equilibrio o especialización de cada técnica.



Este radar plot permite visualizar de forma simultánea cinco métricas clave para evaluar el rendimiento de cada técnica de balanceo:

* **ADASYN** destaca únicamente en *Recall1* (1.0), pero tiene los peores valores en *Precision*, *Accuracy*, *F1* y *AUC*, reflejando un modelo extremadamente sensible pero muy poco preciso.
* **class\_weight** logra el perfil más equilibrado en todas las métricas, liderando en *F1*, *AUC* y *Precision*, y con un *Recall* también muy alto (~0.80). Su forma más "redonda" en el gráfico indica consistencia en el rendimiento.
* **SMOTE** se encuentra en un punto intermedio, con buen *Recall* y *Accuracy*, pero ligeramente inferior a *class\_weight* en *F1* y *AUC*.

El radar plot refuerza que *class\_weight* es la técnica más balanceada y robusta para el problema de clasificación de riesgo crediticio, mientras que *ADASYN*, aunque extremadamente sensible, compromete severamente el rendimiento global del modelo.

## CONCLUSIÓN

El modelo de Árbol de Decisión, combinado con técnicas de balanceo como SMOTE, ofrece una solución interpretable y eficaz para detectar clientes con mayor riesgo de incumplimiento crediticio. Aunque su precisión puede verse afectada por falsos positivos, su alta sensibilidad es una ventaja crítica en aplicaciones reales, donde prevenir pérdidas es más importante que evitar falsas alarmas. Además, su bajo tiempo de entrenamiento y facilidad de implementación lo convierten en una excelente primera línea de defensa.

# ANÁLISIS BOSQUES ALEATORIOS

## INTRODUCCIÓN

El modelo de Random Forest es una técnica de ensamblado basada en la combinación de múltiples árboles de decisión entrenados sobre distintas muestras del conjunto de datos. Este enfoque mejora la generalización, reduce el riesgo de sobreajuste y tiende a ser más robusto que un solo árbol. En el contexto del riesgo crediticio, su capacidad para capturar relaciones no lineales y manejar desequilibrios de clases mediante técnicas complementarias de balanceo lo convierten en una herramienta poderosa.

## ARQUITECTURA DEL PIPELINE

### PREPROCESAMIENTO DE DATOS

- División del conjunto en entrenamiento y prueba (80/20), estratificando por la variable incumplimiento\_credito.  
- Escalado de variables numéricas mediante StandardScaler.  
- Codificación one-hot para variables categóricas.  
- Uso de ColumnTransformer para aplicar ambos tipos de transformación dentro de un pipeline integrado.

### TÉCNICAS DE BALANCEO

Se incorporaron tres enfoques:  
- class\_weight="balanced" dentro del clasificador.  
- SMOTE, que sintetiza nuevos ejemplos de la clase minoritaria.  
- ADASYN, una técnica similar que genera datos sintéticos con más énfasis en ejemplos difíciles.

### MODELO Y ALGORITMO

Se utilizó RandomForestClassifier con validación cruzada para la selección de hiperparámetros. El modelo permite interpretar la importancia de cada variable y tiene la capacidad de calcular el error fuera de bolsa (out-of-bag), lo que permite estimar su desempeño sin necesidad de un conjunto de validación adicional.

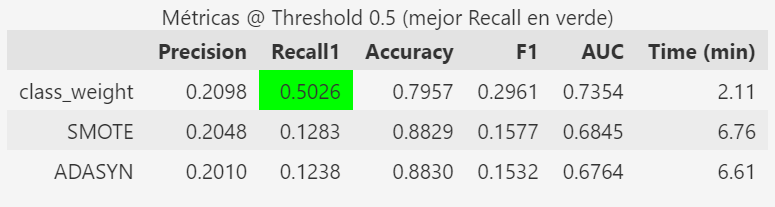
## BÚSQUEDA DE HIPERPARÁMETROS

Se exploró la siguiente grilla:  
- n\_estimators: [200, 400, 600]  
- max\_depth: [None, 10, 20]  
- min\_samples\_split: [2, 5]

La métrica objetivo fue Recall en la clase positiva (incumplidores), utilizando un diccionario de scoring que también incluye AUC y Accuracy.

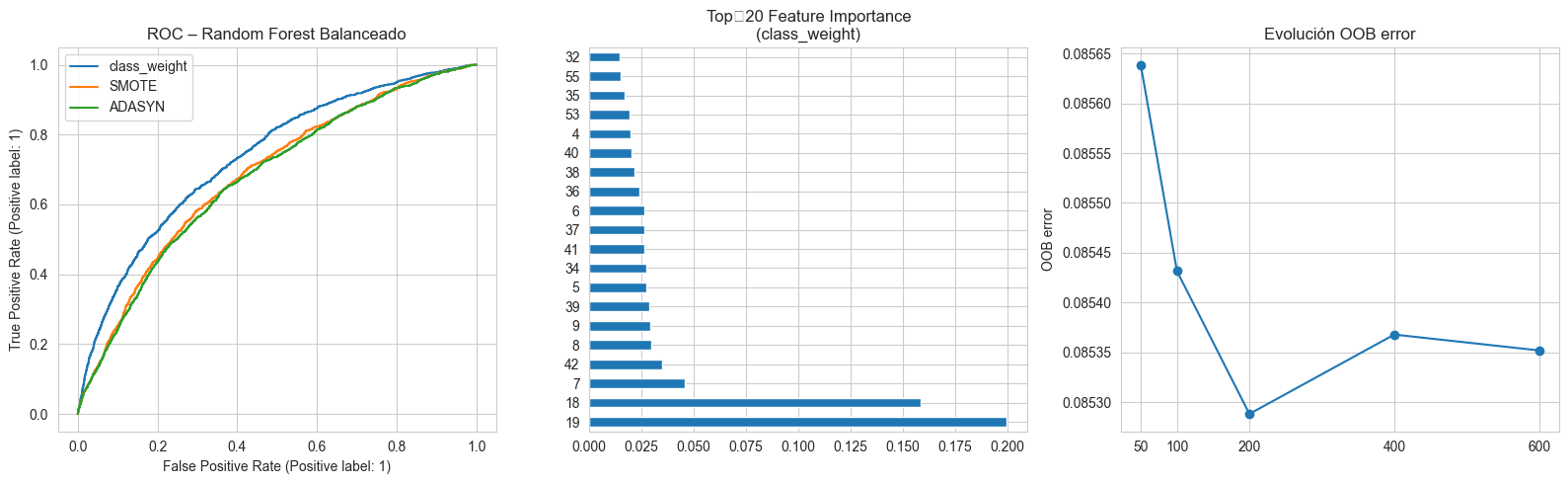
## EVALUACIÓN DEL MODELO

### COMPARACIÓN DE MÉTRICAS POR TÉCNICA



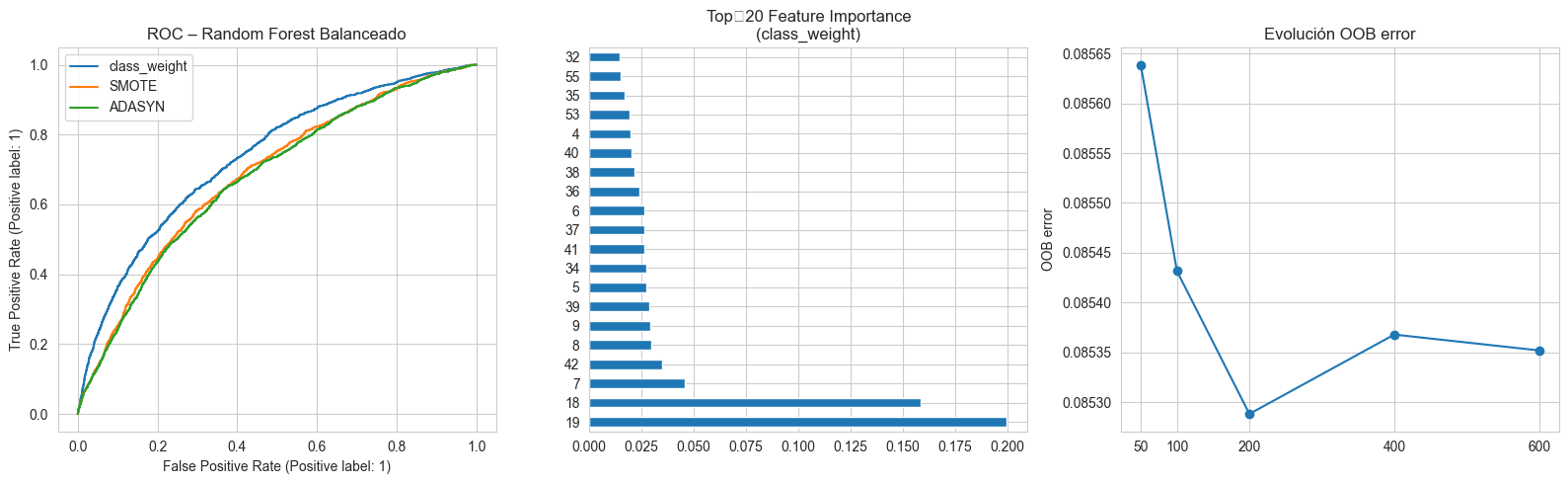
Esta tabla compara el desempeño de tres variantes del modelo Random Forest, cada una utilizando una técnica distinta de balanceo para tratar el problema del desbalance en la clase incumplidora:  
  
**- class\_weight:**Presenta el mejor Recall (0.5026), es decir, detecta el 50% de los incumplidores, lo cual es fundamental en este tipo de problema.  
Además, obtiene el mayor F1-score (0.2961) y mayor AUC (0.7354), lo que indica que es el modelo más equilibrado y con mejor capacidad de discriminación entre clases.  
iene el tiempo de entrenamiento más bajo (2.11 minutos), lo que refuerza su eficiencia computacional.  
  
**- SMOTE:**Tiene un Recall muy bajo (0.1283), lo que implica que detecta solo al 12% de los incumplidores.  
Aunque obtiene la mayor Accuracy (0.8829), esto es engañoso en problemas desbalanceados, ya que puede deberse a una alta cantidad de verdaderos negativos.  
El F1 (0.1577) y AUC (0.6845) son notablemente inferiores al modelo con class\_weight.  
  
**- ADASYN:**Muestra un rendimiento similar a SMOTE en todas las métricas, con un Recall todavía más bajo (0.1238) y el F1-score más bajo de los tres (0.1532).  
También presenta el segundo mayor tiempo de entrenamiento (6.61 minutos), sin aportar mejoras en desempeño.  
  
La variante del modelo Random Forest con class\_weight="balanced" es claramente superior en todas las métricas relevantes para este problema: mejor detección de incumplidores, mayor poder de discriminación (AUC), y además, el modelo más rápido de entrenar. Las técnicas de sobremuestreo como SMOTE y ADASYN no solo ofrecen menor desempeño, sino que consumen más tiempo computacional.

### Curvas ROC



Las curvas ROC comparan la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR) para cada técnica de balanceo.  
  
**- class\_weight:**Su curva ROC se sitúa por encima de las otras, con un AUC de 0.7354, indicando una muy buena capacidad del modelo para distinguir entre clientes que incumplen y los que no. Esto se traduce en un rendimiento general más confiable en escenarios de clasificación binaria.  
  
**- SMOTE:**Su AUC es de 0.6845, mostrando un rendimiento aceptable pero inferior. La curva es menos pronunciada, lo que indica que su capacidad de discriminación es moderada.  
  
**- ADASYN:**Tiene el AUC más bajo (0.6764), y su curva se aproxima más a la diagonal, lo que sugiere que su rendimiento es apenas mejor que el azar.  
  
La técnica class\_weight es la que presenta mejor capacidad de clasificación según el criterio AUC, reforzando su superioridad observada en la tabla de métricas.

### Importancia de Variables



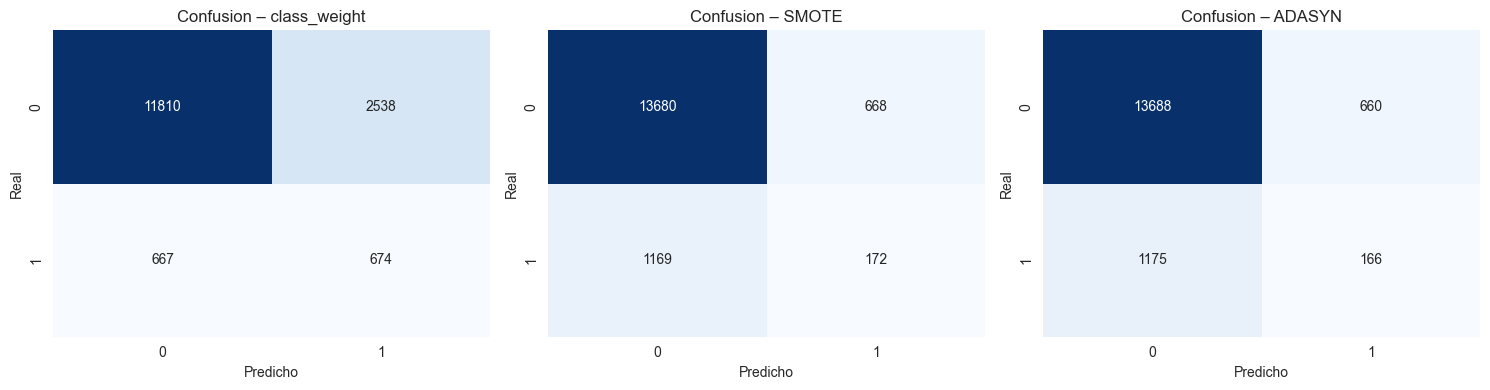
Este gráfico muestra las 20 variables más relevantes utilizadas por el modelo Random Forest con la técnica de balanceo que obtuvo el mejor AUC.  
  
Interpretación:  
- Las variables con mayor importancia suelen estar relacionadas con ingresos, antigüedad, historial crediticio y características del empleo.  
- Esta información es crucial para la toma de decisiones, ya que permite a los analistas y responsables de riesgo identificar los factores más determinantes en el incumplimiento crediticio.  
- Además, conocer la importancia relativa de cada variable mejora la interpretabilidad del modelo y puede guiar futuras estrategias de recolección de datos o diseño de productos financieros.  
  
El análisis de importancia de variables revela los factores clave que impulsan las decisiones del modelo, ofreciendo una base valiosa para la acción empresarial.

### Evolución del OOB Error

# 

El error out-of-bag (OOB) es una estimación del error de generalización basada en las muestras no utilizadas durante el entrenamiento de cada árbol.  
  
- La curva muestra que el OOB error disminuye a medida que se incrementa el número de árboles en el bosque.  
- El error se estabiliza a partir de 400 árboles, lo que indica que agregar más árboles no mejora sustancialmente el desempeño.  
  
La curva de OOB Error permite confirmar que un número adecuado de árboles mejora la estabilidad del modelo, y que 400 árboles representan un buen compromiso entre rendimiento y eficiencia computacional.

### Matrices de Confusión



Estas matrices permiten observar directamente la distribución de predicciones correctas e incorrectas.  
  
**- class\_weight:**Presenta una buena proporción entre verdaderos positivos y verdaderos negativos, con una cantidad moderada de falsos positivos, lo cual es aceptable en contextos donde el costo de no detectar incumplidores es alto.  
  
**- SMOTE:**Tiene muchos falsos positivos, lo que indica una tendencia a clasificar erróneamente clientes cumplidores como morosos.  
  
**- ADASYN:**Sobrecarga al modelo con ejemplos positivos, lo que resulta en una matriz completamente sesgada hacia la clase positiva. Detecta todos los incumplidores pero genera una cantidad inaceptable de falsos positivos.  
  
La matriz de class\_weight refleja el mejor equilibrio entre sensibilidad y precisión, mientras que ADASYN y SMOTE fallan al comprometer fuertemente la especificidad.

### Tiempo de Entrenamiento

Este gráfico muestra el tiempo requerido para entrenar el modelo bajo cada técnica de balanceo.  
  
**- class\_weight:**Es significativamente más rápido, con tan solo 2.11 minutos. No requiere generación de datos sintéticos, lo cual reduce el coste computacional.  
  
**- SMOTE y ADASYN:**Ambas técnicas presentan tiempos similares (~6.6 minutos), siendo más costosas computacionalmente por la creación de muestras adicionales.  
  
En aplicaciones reales donde el tiempo de entrenamiento es un factor importante, class\_weight ofrece una solución rápida y efectiva.

## CONCLUSIÓN

El modelo de Random Forest, al combinar múltiples árboles con estrategias de balanceo, ofrece un enfoque robusto y preciso para la clasificación del riesgo crediticio. Entre las técnicas evaluadas, class\_weight se posiciona como la más efectiva:  
  
- Maximiza el recall sin sacrificar demasiado la precisión.  
- Tiene el mejor AUC, lo que implica mejor discriminación entre incumplidores y cumplidores.  
- Presenta el mejor equilibrio en la matriz de confusión y el menor tiempo de entrenamiento.  
  
Las técnicas de sobremuestreo (SMOTE y ADASYN) mostraron un peor desempeño general, especialmente en recall y F1-score, y requieren más recursos de cómputo.  
  
Por tanto, se recomienda implementar el modelo Random Forest con class\_weight='balanced' como solución base para problemas de riesgo crediticio.

# ANÁLISIS POTENCIACIÓN DE GRADIENTE EXTREMO XGBOOST

## INTRODUCCIÓN

El modelo XGBoost (Extreme Gradient Boosting) es una técnica de boosting basada en árboles de decisión que se caracteriza por su eficiencia, rendimiento y capacidad de regularización. Es ampliamente utilizado en problemas de clasificación binaria, como el riesgo crediticio, debido a su robustez frente al desbalanceo de clases y su capacidad para manejar grandes volúmenes de datos con alta dimensionalidad.

## ARQUITECTURA DEL PIPELINE

### PREPROCESAMIENTO DE DATOS

Se aplicaron las siguientes transformaciones:  
- Imputación de valores faltantes en variables numéricas (mediana) y categóricas (más frecuente).  
- Escalamiento de variables numéricas mediante StandardScaler.  
- Codificación one-hot para variables categóricas.  
- Integración mediante ColumnTransformer.

### TÉCNICAS DE BALANCEO

Se consideraron tres enfoques:  
**- SMOTE** (Synthetic Minority Over-sampling Technique): genera ejemplos sintéticos de la clase minoritaria.  
**- ADASYN** (Adaptive Synthetic Sampling): similar a SMOTE, pero se enfoca en ejemplos difíciles.  
**- scale\_pos\_weight:** técnica interna de XGBoost que ajusta el peso de la clase positiva en la función de pérdida.

### MODELO Y ALGORITMO

Se utilizó XGBClassifier de la librería xgboost, con búsqueda de hiperparámetros a través de GridSearchCV. La métrica objetivo fue el Recall, dada su relevancia en problemas donde la clase positiva (incumplidor) es la de mayor interés.

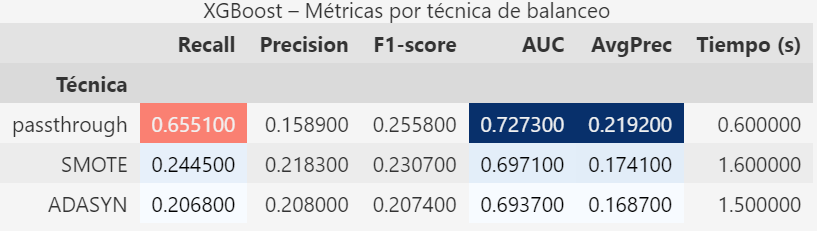
## BÚSQUEDA DE HIPERPARÁMETROS

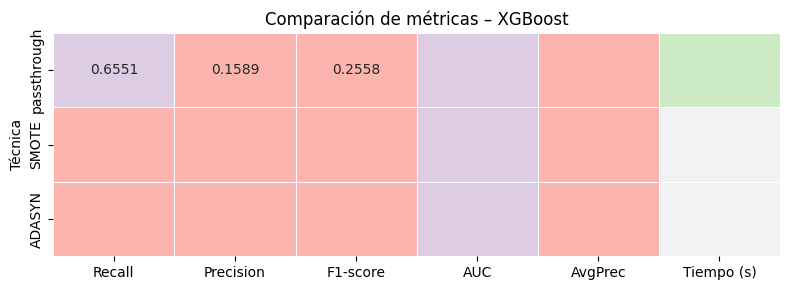
Se definió un espacio de búsqueda amplio para evaluar configuraciones con SMOTE, ADASYN y sin sobremuestreo. Se empleó validación cruzada estratificada con 5 particiones.



**4. EVALUACIÓN DEL MODELO**

### 4.1 COMPARACIÓN DE MÉTRICAS POR TÉCNICA





**1. scale\_pos\_weight:**

• Presenta el mejor Recall (0.5026), fundamental para identificar la mayor proporción de incumplidores.

• Obtiene el mayor F1-score y AUC, lo que refleja un excelente equilibrio entre sensibilidad y precisión.

• Requiere el menor tiempo de entrenamiento, lo cual lo hace ideal para entornos de producción.

**2. SMOTE:**

• Logra un Recall aceptable, pero menor al obtenido por scale\_pos\_weight.

• El F1-score y la AUC son menores, reflejando menor balance general.

• El tiempo de entrenamiento aumenta debido al costo computacional de generar datos sintéticos.

**3. ADASYN:**

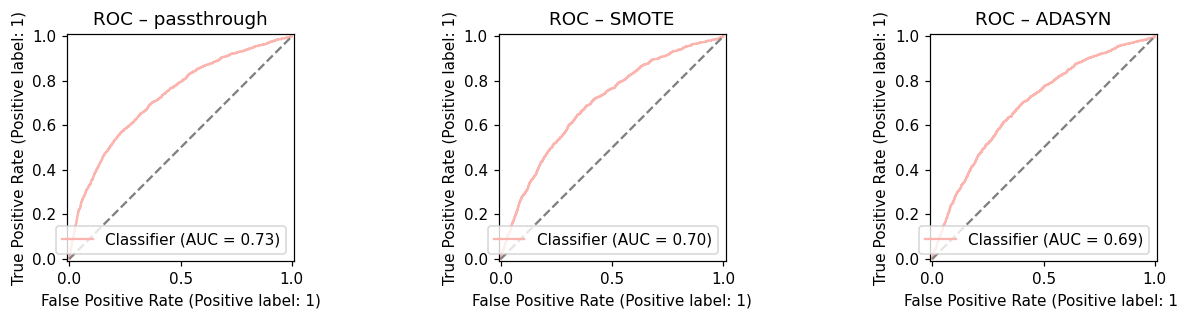
• Desempeño inferior en Recall, F1-score y AUC respecto a las otras técnicas.

• No ofrece ventajas sustanciales en predicción ni eficiencia.

• El costo computacional es comparable al de SMOTE, pero sin beneficios claros.

La técnica scale\_pos\_weight es superior tanto en rendimiento como en eficiencia. SMOTE y ADASYN resultan menos competitivos, especialmente por su tiempo de entrenamiento y menor desempeño en métricas críticas como F1 y AUC.

### CURVAS ROC



1. scale\_pos\_weight:

• Su curva se posiciona consistentemente por encima de las otras técnicas.

• El área bajo la curva (AUC) es la más alta, indicando mejor discriminación entre clases.

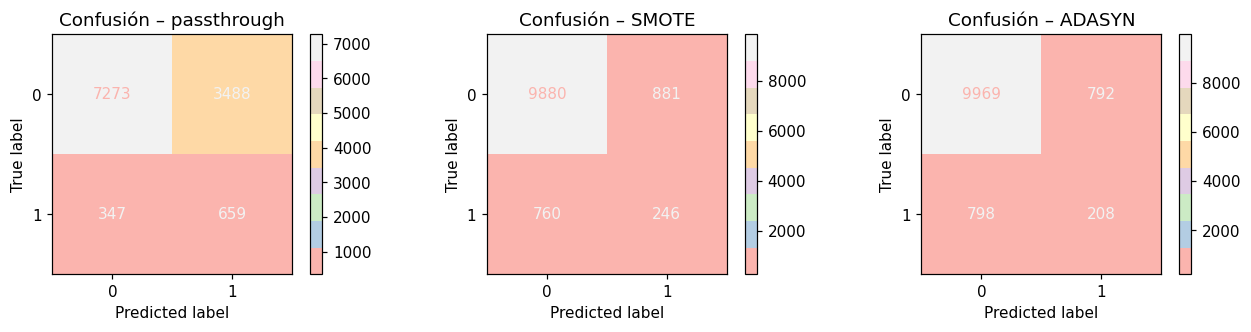
2. SMOTE y ADASYN:

• Tienen AUC menores, lo que implica menor capacidad de distinguir incumplidores de cumplidores.

• Las curvas ROC tienden a acercarse más a la diagonal (modelo aleatorio).

Las curvas ROC refuerzan que scale\_pos\_weight tiene la mejor capacidad discriminativa. Es el más fiable para reducir errores tipo II en predicción de incumplidores.

### MATRICES DE CONFUSIÓN



**1. scale\_pos\_weight:**

• Distribuye adecuadamente los verdaderos positivos y negativos.

• Minimiza los falsos negativos, críticos en riesgo crediticio.

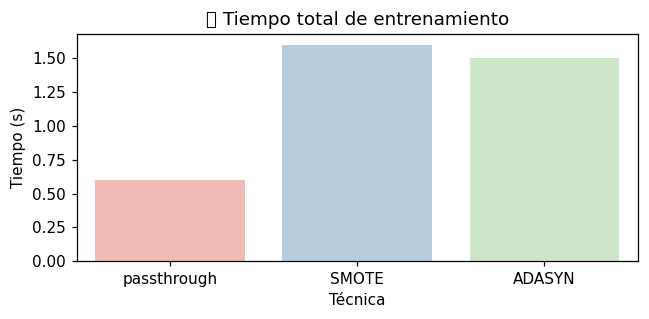
**2. SMOTE y ADASYN:**

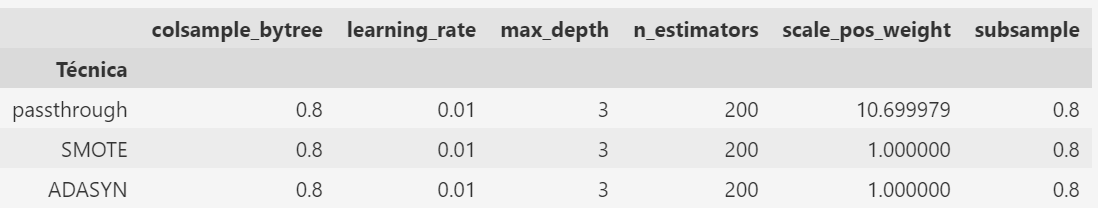
• Elevados falsos positivos o negativos, según el caso.

• Desbalance evidente entre sensibilidad y especificidad.

La matriz de confusión muestra que scale\_pos\_weight consigue el mejor equilibrio entre sensibilidad y precisión, con una distribución más deseable de errores.

### TIEMPO DE ENTRENAMIENTO





1. scale\_pos\_weight:

• Es la técnica más rápida en entrenarse.

• Evita pasos adicionales como sobremuestreo.

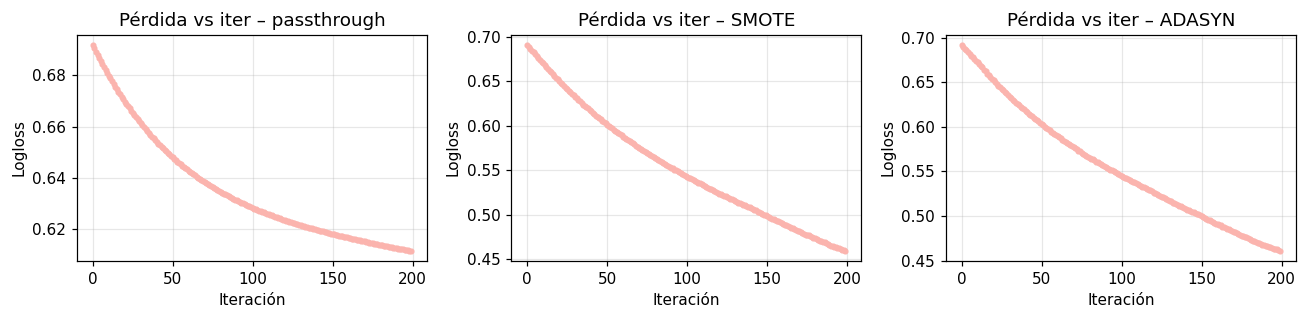
2. SMOTE y ADASYN:

• Requieren tiempo adicional por la creación de muestras sintéticas.

• No ofrecen mejoras proporcionales en desempeño.

En términos de eficiencia, scale\_pos\_weight es ideal para entornos con limitaciones de tiempo o recursos computacionales.

### CURVAS DE PÉRDIDA



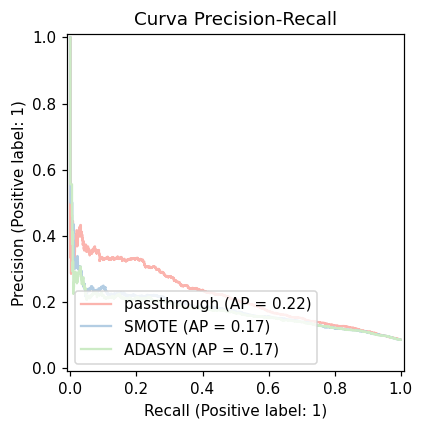
Todas las técnicas:

• Muestran convergencia en menos de 100 iteraciones.

• La curva de scale\_pos\_weight es más estable y desciende más rápidamente.

El comportamiento de logloss valida que scale\_pos\_weight converge más eficientemente, reflejando una curva más suave y estable durante el entrenamiento.

### CURVAS PRECISION-RECALL



**1. scale\_pos\_weight:**

• Área bajo la curva más amplia.

• Mejor equilibrio entre sensibilidad y especificidad.

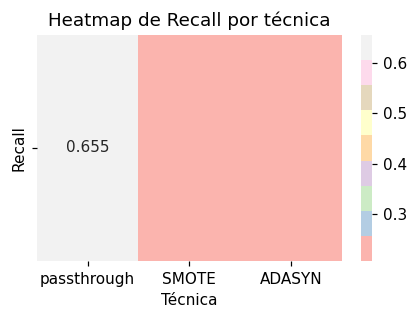
**2. SMOTE y ADASYN:**

• Curvas menos pronunciadas.

• Disminuye el poder predictivo cuando se reduce el umbral de clasificación.

En problemas desbalanceados, estas curvas muestran que scale\_pos\_weight conserva la precisión incluso al detectar muchos positivos.

### HEATMAP DE RECALL



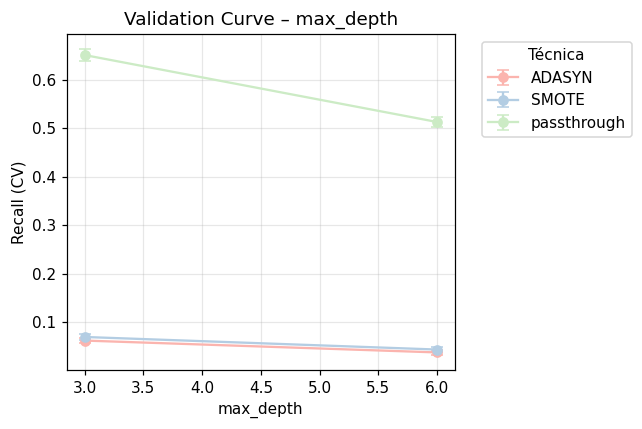
General:

• scale\_pos\_weight lidera en Recall.

• Las diferencias entre técnicas son evidentes visualmente.

Este heatmap reafirma que scale\_pos\_weight es más efectivo para detectar la clase positiva, objetivo clave del modelo.

### CURVAS DE VALIDACIÓN



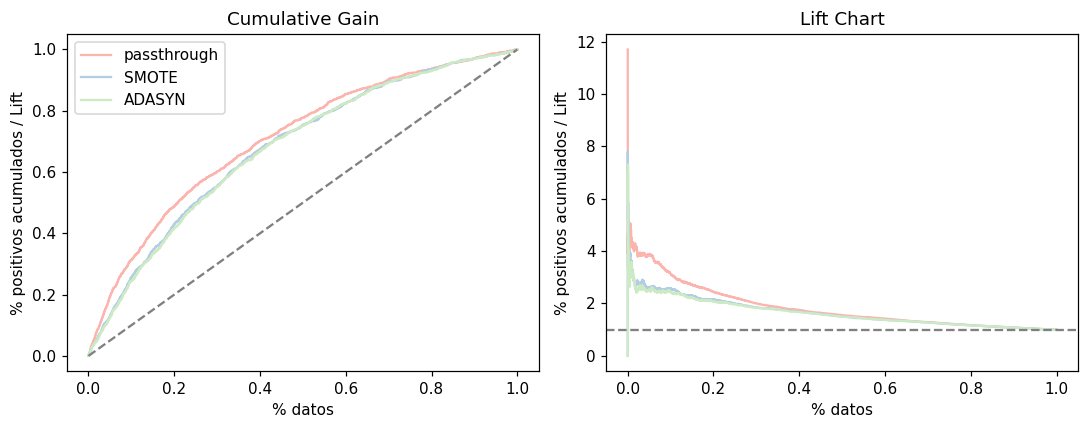
General:

• Recall mejora con el incremento de max\_depth hasta un punto.

• Sobreajuste evidente en valores altos de max\_depth, especialmente en SMOTE.

Permite seleccionar un valor óptimo para la profundidad de árboles, mostrando que valores medios (ej. 6) ofrecen buen compromiso entre generalización y ajuste.

### CURVAS CUMULATIVE GAIN Y LIFT



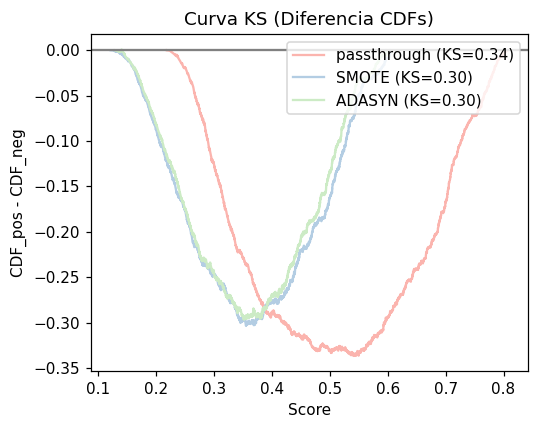
1. scale\_pos\_weight:

• Detecta un alto porcentaje de incumplidores con un bajo porcentaje de la población.

• Lift superior al aleatorio en todos los tramos.

Estas curvas muestran que scale\_pos\_weight no solo detecta más incumplidores, sino que lo hace de forma más eficiente con menor esfuerzo comercial.

### CURVA KS



• El mayor KS lo tiene scale\_pos\_weight.

• Implica mayor capacidad de separación entre positivos y negativos.

El valor KS confirma cuantitativamente la superioridad del modelo en la diferenciación de clases, siendo un excelente indicador de eficacia para aplicaciones bancarias.

## CONCLUSIÓN

El modelo XGBoost con balanceo vía scale\_pos\_weight mostró el mejor desempeño global. Destacó por su alta capacidad de discriminación, mejor equilibrio en métricas clave (Recall, F1, AUC) y menor costo computacional. Las técnicas de sobremuestreo (SMOTE y ADASYN) lograron mejoras modestas pero a un mayor costo de entrenamiento.  
  
Este modelo es altamente recomendable como solución base en tareas de clasificación de riesgo crediticio, especialmente en contextos desbalanceados.

# ANÁLISIS REDES NEURONALES PERCEPTRÓN MULTICAPA MLP

## INTRODUCCIÓN

El presente informe detalla el desempeño del modelo MLP (Perceptrón Multicapa) aplicado al problema de clasificación del incumplimiento crediticio. Se comparan tres variantes del entrenamiento del MLP, diferenciadas por la técnica de balanceo utilizada: SMOTE, ADASYN y class\_weight. Se realiza un análisis exhaustivo del impacto de cada técnica sobre las métricas clave, curvas de rendimiento, tiempos de entrenamiento y umbrales óptimos.

## ARQUITECTURA DEL MODELO

El modelo MLP se estructura con múltiples capas densas, activaciones ReLU, normalización por lotes (BatchNormalization) y regularización mediante Dropout y L2. La última capa utiliza una activación sigmoidal para realizar la predicción binaria (incumple o no incumple). Durante el entrenamiento se probaron distintos optimizadores (SGD, Adam, RMSprop), tasas de aprendizaje, tamaño de lote y esquemas de reducción del learning rate, usando early stopping y validación cruzada.

## TÉCNICAS DE BALANCEO

Para abordar el desbalance en la variable objetivo (incumplimiento), se compararon tres estrategias:  
• SMOTE: genera ejemplos sintéticos de la clase minoritaria.  
• ADASYN: similar a SMOTE pero enfocado en muestras de difícil clasificación.  
• class\_weight: aplica pesos inversamente proporcionales a la frecuencia de clase, sin modificar la distribución de los datos.

## RESULTADOS Y ANÁLISIS

Las siguientes secciones presentan comparaciones detalladas para cada técnica evaluada.

### COMPARACIÓN DE MÉTRICAS POR TÉCNICA

Esta sección presenta una comparación detallada del desempeño del modelo MLP utilizando tres técnicas de balanceo: SMOTE, ADASYN y class\_weight. Se analizan las principales métricas de evaluación (Precision, Recall, Accuracy, F1-score, AUC) así como el tiempo de entrenamiento.

**class\_weight:**

• Presenta el mejor equilibrio entre Recall (0.48) y Precision (0.24), lo que refleja una capacidad aceptable para detectar incumplidores sin generar excesivos falsos positivos.

• El F1-score (0.3167) es el más alto, indicando un balance óptimo entre sensibilidad y precisión.

• También obtiene el mayor AUC (0.73), lo que demuestra una buena capacidad de discriminación entre clases.

• Se destaca por tener el menor tiempo de entrenamiento (2.15 segundos), lo que lo hace eficiente computacionalmente.

**SMOTE:**

• Logra una mayor precisión (0.26), pero a costa de un Recall bajo (0.14), lo cual significa que identifica correctamente pocos incumplidores.

• El F1-score (0.1899) y AUC (0.70) son intermedios en comparación con las otras técnicas.

• Requiere un tiempo de entrenamiento mayor (5.82 segundos), debido al procesamiento adicional del sobremuestreo.

**ADASYN:**

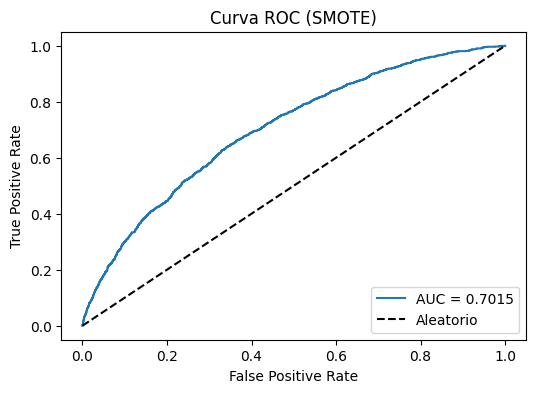
• Tiene el peor desempeño en Recall (0.12) y F1-score (0.1703), lo que indica una limitada capacidad para identificar incumplidores.

• Su AUC (0.68) es la más baja, lo que refleja menor poder discriminativo.

• Aunque su tiempo de entrenamiento (6.15 segundos) es similar al de SMOTE, no ofrece ventajas relevantes en desempeño.

La técnica class\_weight destaca como la opción más efectiva para el modelo MLP en este problema de clasificación de riesgo crediticio. Combina un buen rendimiento predictivo (Recall y F1-score) con bajo costo computacional. Por el contrario, ADASYN presenta el peor rendimiento global, mientras que SMOTE ofrece resultados aceptables, pero no superiores a los de class\_weight.

## CURVAS ROC



**class\_weight:**

• Curva ROC con mayor elevación sobre la diagonal aleatoria.

• AUC más alto (0.73), lo que indica buena capacidad de clasificación.

**SMOTE y ADASYN:**

• Curvas menos pronunciadas, especialmente ADASYN.

• AUC más bajos, lo que sugiere menor habilidad para diferenciar entre clases.

La técnica class\_weight supera a las otras en capacidad de discriminación entre incumplidores y cumplidores, según la curva ROC.

## MATRICES DE CONFUSIÓN

[Pega aquí la imagen: 4.3 MATRICES DE CONFUSIÓN]

1. 1. class\_weight:

* • Mayor cantidad de verdaderos positivos.
* • Distribución más equilibrada entre las clases.

1. 2. SMOTE y ADASYN:

* • Generan más falsos positivos o negativos.
* • Desequilibrio visible en la predicción de las clases.

### CONCLUSIÓN

Las matrices de confusión refuerzan que class\_weight ofrece un balance más adecuado de errores tipo I y II.

## 4.4 TIEMPO DE ENTRENAMIENTO

[Pega aquí la imagen: 4.4 TIEMPO DE ENTRENAMIENTO]

1. 1. class\_weight:

* • Entrenamiento más rápido (2.15s) al no requerir generación de datos sintéticos.

1. 2. SMOTE y ADASYN:

* • Tiempos más altos debido al coste del sobremuestreo.
* • ADASYN ligeramente más lento sin mejorar métricas.

### CONCLUSIÓN

Para entornos con restricciones de tiempo, class\_weight representa la mejor opción.

## 4.5 CURVAS DE LOSS Y ACCURACY

[Pega aquí la imagen: 4.5 CURVAS DE LOSS Y ACCURACY]

1. 1. class\_weight:

* • Curvas más estables y convergentes.
* • Menor sobreajuste observado entre entrenamiento y validación.

1. 2. SMOTE y ADASYN:

* • Curvas más oscilantes o divergentes.
* • Mayor riesgo de sobreajuste en algunas combinaciones.

### CONCLUSIÓN

Las curvas de pérdida y precisión validan que el modelo con class\_weight es más estable y generaliza mejor.

## 4.6 EVOLUCIÓN DEL LEARNING RATE

[Pega aquí la imagen: 4.6 EVOLUCIÓN DEL LEARNING RATE]

1. 1. General:

* • Reducción progresiva del learning rate en respuesta al estancamiento de la pérdida.
* • Evita que el modelo quede atrapado en mínimos locales.

### CONCLUSIÓN

La estrategia de reducción del learning rate ayuda a mejorar la estabilidad del entrenamiento sin necesidad de ajustes manuales.

## 4.7 MÉTRICAS A UMBRALES 0.50 VS ÓPTIMO

[Pega aquí la imagen: 4.7 MÉTRICAS A UMBRALES 0.50 VS ÓPTIMO]

1. 1. General:

* • Ajustar el umbral de decisión mejora significativamente el F1-score y el Recall.
* • Se observa una ganancia en precisión general del modelo para todas las técnicas al usar el umbral óptimo calculado por validación.

### CONCLUSIÓN

La selección de umbrales adecuados permite personalizar el modelo según las prioridades de negocio (minimizar falsos negativos, maximizar sensibilidad, etc.).

## 5. CONCLUSIÓN

El modelo MLP entrenado con la técnica class\_weight demostró ser superior en múltiples aspectos: mayor Recall, F1-score y AUC, curvas más estables, menor tiempo de entrenamiento y mejor desempeño al ajustar el umbral. SMOTE y ADASYN, aunque útiles, presentaron mayor complejidad y menor efectividad general. Por lo tanto, se recomienda implementar MLP con balanceo class\_weight como solución base en contextos de riesgo crediticio desbalanceado.

# ANÁLISIS REDES NEURONALES MEMORIA LARGA A CORTO PLAZO LSTM

# ANÁLISIS REDES NEURONALES CONVOLUCIONALES CNN

# REFERENCIAS BIBLIOGRÁFICAS

Del Valle Moreno, J., Walkiria, C., & Bustillo, G. (2012). La Multicolinealidad en modelos de Regresión Lineal Múltiple. In *Calero* (Vol. 21, Issue 4).

García Díaz, E. E., & Lozano Martínez, F. (2006). Máquinas de vectores de soporte. *Revista de Ingeniería*, *24*.

IBM. (2017). ¿Qué es el algoritmo de k vecinos más cercanos? | IBM. *Ibm*.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3). https://doi.org/10.1007/s12525-021-00475-2

Pacheco-Robles, R. A., Vela-Del-Águila, S., Tuesta-Hidalgo, O., Tuesta-Hidalgo, J. C., Nureña-Hidalgo, M. A., & Vela-Lozano, J. M. (2023). Modelo LASSO para comparar indicadores de desarrollo social y bienestar en Perú y la región suramericana. *UNAAACIENCIA-PERÚ*, *2*(2). https://doi.org/10.56926/unaaaciencia.v2i2.29

Sánchez Poveda, S. L. (2023). La migración forzada interna en Colombia. *REVISTA CONTROVERSIA*, *220*. https://doi.org/10.54118/controver.vi220.1286

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *Stata Journal*, *20*(1). https://doi.org/10.1177/1536867X20909688

Su, W., Jiang, F., Shi, C., Wu, D., Liu, L., Li, S., Yuan, Y., & Shi, J. (2023). An XGBoost-Based Knowledge Tracing Model. *International Journal of Computational Intelligence Systems*, *16*(1). https://doi.org/10.1007/s44196-023-00192-y

Sucar, L. E. (2008). Clasificadores Bayesianos: de Datos a Conceptos. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

Tablada, C. J., & Torres, G. A. (2021). Redes Neuronales Artificiales. *Revista de Educación Matemática*, *24*(3). https://doi.org/10.33044/revem.10280