**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

Para la búsqueda de información con relación al estado del arte en el análisis de los modelos desarrollados, se tomó como base de datos de referencia bibliográfica a SCOPUS, a continuación, se presenta el análisis bibliográfico para cada uno de los modelos:

## BAYES

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos de clasificación Bayes en el análisis de crédito?

**Palabras clave para la búsqueda de información en Scopus:**

("Bayes classifier" OR "Bayesian model" OR "Bayesian network" OR "probabilistic model") AND ("credit analysis" OR "credit risk" OR "loan assessment" OR "financial evaluation" OR "default prediction") AND ("machine learning" OR "data mining" OR "statistical learning" OR "predictive modeling") AND ("evaluation" OR "assessment" OR "performance" OR "validation")

**Análisis de referencias:**

Los modelos de clasificación bayesianos, y en particular el clasificador Naïve Bayes, son ampliamente utilizados en el análisis de crédito debido a su efectividad en tareas de clasificación y su capacidad para integrar conocimiento experto junto con razonamiento probabilístico. Estos modelos permiten clasificar a los solicitantes de préstamos personales a partir de la combinación de información experta y datos reales sobre préstamos, y han demostrado ser más eficaces que los modelos de regresión logística tradicionales. Tanto Naïve Bayes como las redes bayesianas se destacan como herramientas valiosas en el análisis crediticio, ya que logran un equilibrio entre simplicidad, efectividad y la incorporación de conocimiento experto. Sin embargo, es importante considerar que otros modelos como SVM y Random Forest pueden proporcionar una mayor precisión, lo que sugiere que adoptar un enfoque híbrido o seleccionar modelos de acuerdo a necesidades específicas podría resultar beneficioso para optimizar los resultados.

## REGRESIÓN LOGÍSTICA (L1 L2)

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de la Regresión Logística con regularización L1 y L2 en el análisis de crédito?

**Palabras clave para la búsqueda de información en Scopus:**

("logistic regression" OR "logit" OR "regression analysis" OR "binary classification") AND ("credit analysis" OR "credit risk" OR "loan assessment" OR "financial evaluation") AND ("regularization" OR "penalization" OR "shrinkage" OR "overfitting") AND ("modeling" OR "prediction" OR "forecasting" OR "analytics")

**Análisis de referencias:**

Los modelos de regresión logística con regularización L1 y L2 son ampliamente utilizados en el análisis crediticio, debido a su excepcional capacidad para manejar datos de alta dimensión y realizar la selección de características. Esta habilidad es fundamental para predecir el riesgo crediticio y las probabilidades de incumplimiento. La regularización L1, también conocida como Lasso, se destaca en la selección de características al reducir algunos coeficientes a cero, lo que facilita la creación de un subconjunto de predictores más relevantes para el modelo. Esta ventaja es especialmente valiosa en el análisis crediticio, donde identificar los predictores clave de incumplimiento es esencial, superando así a los modelos de regresión logística tradicionales en términos de precisión e interpretabilidad. Por otro lado, la regularización L2, conocida como regresión de cresta, resulta útil para abordar la multicolinealidad al distribuir la penalización entre todos los coeficientes, ayudando a estabilizar el modelo. De esta manera, también se mejora la capacidad de generalización del modelo, haciéndolo más robusto frente al sobreajuste, aspecto crucial en el modelado del riesgo crediticio, ya que los datos suelen ser ruidosos y complejos.

## DECISION TREE

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos de árboles de decisión en el análisis de crédito?

**Palabras clave para la búsqueda de información en Scopus:**

("decision tree" OR "tree model" OR "classification tree" OR "regression tree") AND ("credit analysis" OR "credit risk" OR "loan assessment" OR "financial evaluation") AND ("predictive modeling" OR "data mining" OR "machine learning" OR "statistical analysis") AND ("evaluation" OR "assessment" OR "scoring" OR "rating") AND ("financial institution" OR "bank" OR "lender" OR "creditor")

**Análisis de referencias:**

Los modelos de árboles de decisión representan un enfoque sólido y efectivo para el análisis crediticio, destacándose por su alta precisión, eficiencia y habilidad para manejar conjuntos de datos complejos. Su fácil interpretabilidad y su capacidad para identificar factores de riesgo los convierten en herramientas invaluables para las organizaciones financieras que buscan gestionar el riesgo crediticio de manera efectiva. Estos modelos optimizan el proceso de evaluación crediticia, haciéndolo más ágil y eficiente. Un ejemplo de esto es el método CART, que mejora la precisión en la predicción de la viabilidad de los prestatarios. Al combinar técnicas de minería de datos y árboles de decisión, se logra una evaluación del riesgo crediticio en el contexto de préstamos hipotecarios individuales. Además, los modelos de árboles de decisión son especialmente útiles en entornos de big data, donde son capaces de gestionar grandes volúmenes de datos desordenados, mientras que proporcionan interpretaciones claras y reglas fáciles de entender, lo que resulta fundamental para la evaluación crediticia.

## RANDOM FOREST

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos Random Forest en el análisis de crédito?

**Palabras clave para la búsqueda de información en Scopus:**

("random forest" OR "ensemble learning" OR "decision tree" OR "classification model") AND ("credit analysis" OR "credit risk" OR "loan assessment" OR "financial evaluation" OR "risk modeling") AND ("predictive modeling" OR "data mining" OR "machine learning" OR "statistical analysis") AND ("financial data" OR "credit score" OR "default prediction" OR "loan performance")

**Análisis de referencias:**

Los modelos de Random Forest presentan un gran potencial en la predicción del riesgo crediticio, destacando por su excelente rendimiento en la gestión de datos faltantes y en el manejo de conjuntos de datos complejos. Se utilizan comúnmente para estimar la aprobación de préstamos hipotecarios, así como en la evaluación del riesgo de préstamos en línea, la previsión de créditos personales y el análisis de crédito empresarial. Este enfoque permite analizar la importancia de las variables y alcanzar una precisión predictiva óptima, especialmente en contextos donde las interrelaciones son complejas. Sin embargo, a pesar de su alto rendimiento, los modelos de Random Forest tienen la particularidad de ser considerados "caja negra", lo que hace que la interpretación de sus resultados sea un desafío.

## XGBOOST

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos XGBoost en el análisis de crédito?

**Palabras clave para la búsqueda de información en Scopus:**

("XGBoost" OR "extreme gradient boosting" OR "boosting" OR "ensemble learning") AND ("credit analysis" OR "credit scoring" OR "loan assessment" OR "risk evaluation" OR "financial analysis") AND ("machine learning" OR "predictive modeling" OR "data mining" OR "statistical learning") AND ("performance" OR "accuracy" OR "evaluation" OR "metrics")

**Análisis de referencias:**

Los modelos de XGBoost en el análisis crediticio presentan diversas aplicaciones que contribuyen a mejorar varios aspectos, como la selección de características, el rendimiento en clasificación, la gestión de datos desbalanceados, la detección de fraudes y la optimización de modelos. Una de sus principales virtudes es su destacado rendimiento en diferentes conjuntos de datos financieros, proporcionando elementos clave para la optimización de la gestión del riesgo crediticio. Este modelo se destaca por su capacidad para identificar características relevantes en grandes volúmenes de datos, lo cual es fundamental para realizar una evaluación precisa del riesgo crediticio. Además, XGBoost es eficiente en el manejo del desbalance de clases, un fenómeno común en los conjuntos de datos relacionados con crédito. No obstante, es importante señalar que XGBoost puede enfrentar limitaciones al trabajar con muestras pequeñas y desequilibradas de alta dimensión, lo que puede resultar en una menor precisión del modelo, así como en problemas de sobreajuste y fragmentación de los datos.

## SUPPORT VECTOR MACHINE (SVM)

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos de máquinas de soporte vectorial en el análisis de crédito?

**Palabras clave para la búsqueda de información en Scopus:**

("support vector machine" OR "svm" OR "machine learning" OR "classification") AND ("credit analysis" OR "credit risk" OR "loan assessment" OR "financial evaluation" OR "default prediction") AND ("model" OR "algorithm" OR "technique" OR "approach") AND ("data mining" OR "predictive modeling" OR "statistical analysis" OR "pattern recognition")

**Análisis de referencias:**

Los modelos de máquinas de vectores de soporte (SVM) han encontrado una amplia aplicación en el análisis crediticio, destacándose por su precisión, eficiencia y transparencia. No obstante, su rendimiento está supeditado a la adecuada configuración de sus parámetros y pueden enfrentar desafíos debido a la complejidad computacional que surge al trabajar con grandes volúmenes de datos. En este contexto, se han utilizado variaciones de SVM, como SVM-MK (mezcla de kernel) y PSO-SVM (optimización por enjambre de partículas), para evaluar el riesgo crediticio, logrando resultados prometedores en la clasificación predictiva. Aunque los enfoques SVM no lineales ofrecen una gran precisión, también pueden resultar costosos en términos computacionales a medida que se incrementa el tamaño de los conjuntos de datos históricos relacionados con las calificaciones crediticias.

## MULTI-LAYER PERCEPTRON

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos Perceptrón Multicapa en el análisis crediticio?

**Palabras clave para la búsqueda de información en Scopus:**

("multi-layer perceptron" OR "MLP" OR "neural network" OR "artificial neural network") AND ("credit analysis" OR "credit assessment" OR "credit scoring" OR "loan evaluation" OR "risk analysis") AND ("financial model" OR "predictive model" OR "risk model" OR "evaluation model") AND ("data mining" OR "machine learning" OR "statistical analysis" OR "algorithm")

**Análisis de referencias:**

Los modelos MLP son altamente eficaces para gestionar relaciones complejas y no lineales en grandes volúmenes de datos. Su aplicación en la modelización del riesgo crediticio ha demostrado ser exitosa, ya que permiten identificar patrones intrincados que los modelos tradicionales podrían pasar por alto. Esta capacidad resulta crucial para las instituciones financieras que buscan mitigar el riesgo de impago y aprobar a aquellos perfiles más solventes. Además, estos modelos pueden combinarse con otras técnicas de selección de características y métodos de sobremuestreo sintético de minorías (SMOTE) para optimizar aún más la precisión en la calificación crediticia. Sin embargo, es importante señalar que la creación de modelos a partir de conjuntos de datos desequilibrados puede dar lugar a predicciones sesgadas hacia la clase mayoritaria, lo que representa una posible limitación en el manejo de datos por parte de los modelos MLP.

## RNN

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de las Redes Neuronales Recurrentes en el análisis de crédito?

**Palabras clave para la búsqueda de información en Scopus:**

("recurrent neural networks" OR "RNN" OR "neural network" OR "deep learning") AND ("credit analysis" OR "credit risk" OR "financial assessment" OR "loan evaluation" OR "credit scoring" OR "risk management") AND ("prediction" OR "forecasting" OR "evaluation" OR "assessment") AND ("data" OR "information" OR "analytics" OR "modeling")

**Análisis de referencias:**

Las redes neuronales recurrentes (RNN) son herramientas valiosas en el ámbito del análisis crediticio, gracias a su habilidad para procesar secuencias temporales y captar dependencias a lo largo del tiempo. Estas redes se han utilizado para crear modelos de puntuación crediticia que predicen la probabilidad de que los clientes incumplan o se atrasen en sus pagos, lo que resulta fundamental para una adecuada gestión del riesgo crediticio. Sin embargo, uno de los desafíos más significativos en la aplicación de modelos RNN en este contexto es el problema de la desaparición del gradiente durante el entrenamiento con secuencias largas, lo que puede comprometer el rendimiento del modelo.

## LSTM

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos de memoria a corto plazo (LSTM) en el análisis crediticio?

**Palabras clave para la búsqueda de información en Scopus:**

("LSTM" OR "long short-term memory" OR "recurrent neural network" OR "RNN") AND ("credit" OR "loan" OR "finance" OR "debt") AND ("analysis" OR "assessment" OR "evaluation" OR "risk") AND ("model" OR "prediction" OR "forecast" OR "estimation")

**Análisis de referencias:**

Los modelos LSTM han demostrado su eficacia en diversos aspectos del análisis de crédito, abarcando desde la evaluación del riesgo crediticio tanto a nivel individual como corporativo, hasta la detección de fraudes y la predicción de quiebras. Su habilidad para manejar datos secuenciales y captar dependencias temporales los convierte en una herramienta fundamental para la toma de decisiones financieras y la gestión de riesgos. Al implementar modelos LSTM en el análisis de crédito, resulta esencial considerar el uso de mecanismos de atención, que permiten asignar pesos de manera independiente y optimizar el modelo. Esto no solo mejora la precisión de las predicciones, sino que también, al combinarse con el Proceso Analítico Jerárquico (AHP), se utiliza para evaluar el riesgo crediticio de forma individual. Este enfoque extrae información relevante y establece un sistema de índices que permite la evaluación del riesgo crediticio financiero, que luego se integra en una red neuronal AHP-LSTM. Los resultados de estos modelos permiten un rendimiento superior en la predicción del riesgo crediticio, especialmente en conjuntos de datos no equilibrados.

## CNN

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de las redes neuronales convolucionales en el análisis de créditos?

**Palabras clave para la búsqueda de información en Scopus:**

("convolutional neural networks" OR "cnn" OR "deep learning" OR "neural network") AND ("credit analysis" OR "credit risk" OR "loan assessment" OR "financial evaluation" OR "risk modeling" OR "credit scoring") AND ("financial data" OR "transaction data" OR "credit data" OR "economic indicators") AND ("prediction" OR "forecasting" OR "classification" OR "evaluation")

**Análisis de referencias:**

Las redes neuronales convolucionales de grafos (GCNN) han comenzado a utilizarse para evaluar la solvencia crediticia de los prestatarios, transformando sus datos en estructuras basadas en grafos. Este enfoque ofrece soluciones a los problemas que presentan los modelos tradicionales, como el manejo de conjuntos de datos desbalanceados y la extracción de características relevantes a partir de relaciones complejas. Además, la integración de las CNN con otras arquitecturas de redes neuronales, como las LSTM, potencia la precisión y la robustez en la evaluación crediticia. No obstante, muchos modelos enfrentan dificultades en términos de interpretabilidad, y la naturaleza de "caja negra" de las CNN puede representar un obstáculo para su adopción en el sector financiero.

## AUTOENCODERS

**Pregunta de búsqueda para el estado del arte:**

¿Cuáles son las aplicaciones de los modelos Autoencoder en el análisis crediticio?

**Palabras clave para la búsqueda de información en Scopus:**

("autoencoder" OR "autoencoders" OR "neural network" OR "dimensionality reduction") AND ("credit" OR "loan" OR "finance" OR "risk") AND ("analysis" OR "assessment" OR "evaluation" OR "modeling") AND ("machine learning" OR "data mining" OR "predictive modeling" OR "algorithm")

**Análisis de referencias:**

Los autocodificadores representan una solución eficaz para evaluar el riesgo crediticio y detectar fraudes, mejorando significativamente el preprocesamiento de datos, la extracción de características y el rendimiento de los modelos. Su habilidad para gestionar datos de alta dimensión y aumentar la precisión de los modelos de clasificación los convierte en una herramienta invaluable en el ámbito financiero. Además, los autocodificadores son capaces de filtrar el ruido en los datos, lo que resulta fundamental para optimizar tanto la precisión como la eficiencia de los modelos de riesgo crediticio.

# 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).

# ANALISIS EXPLORATORIO DE LOS 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.

El dataset original es un conjunto complejo de datos sobre préstamos y clientes, con información demográfica, financiera e historial crediticio. El objetivo es predecir el incumplimiento de pago utilizando esta información. El EDA es crucial para comprender, limpiar y preparar estos datos para la construcción de un modelo predictivo.

## Descripción de Dataset Original:

* **Dimensiones**: El conjunto de datos contiene **356.255 registros** y **136 variables**.
* **Variable objetivo**: La variable **incumplimiento\_credito** indica si un cliente no paga el crédito (**1**) o si cumple con el pago (**0**). Solo el **8%** de los casos corresponden a incumplimientos, lo que evidencia un **fuerte desbalance de clases**.
* **Tratamiento de valores nulos**: El dataset presenta **valores nulos** que serán tratados durante el análisis exploratorio de datos (EDA), principalmente mediante su eliminación o imputación según corresponda.
* **Transformaciones previstas en el EDA**:  
  Se aplicarán distintas transformaciones para preparar los datos, tales como:
  + Eliminación de columnas irrelevantes (e.g., ID).
  + Depuración de columnas con alta proporción de nulos o baja variabilidad.
  + Tratamiento del desbalance de clases.
  + Revisión de la correlación entre variables.
* **Objetivo del EDA**: El propósito del análisis exploratorio es **preparar el dataset** para la posterior construcción de modelos de *Machine Learning* enfocados en predecir la **probabilidad de incumplimiento de crédito**.

## CARGA DE DATOS Y LIBRERÍAS:

* Se importan las librerías necesarias como pandas, numpy, seaborn, matplotlib, etc.
* Los datos se cargan desde los siguientes archivos CSV:
  + application\_train.csv
  + application\_test.csv
  + bureau.csv
  + bureau\_balance.csv
  + previous\_application.csv
  + POS\_CASH\_balance.csv
  + credit\_card\_balance.csv
  + installments\_payments.csv

## PREPROCESAMIENTO DE DATOS:

Luego de identificar los tipos de variables, se procede a:

* PRIMERO: Eliminar variable ID.
* SEGUNDO: Se identifica la variable objetivo ("TARGET"), se eliminan las filas con valores nulos en esta variable y se transforma a variable de tipo entero.Imagen de la pantalla de un celular con texto

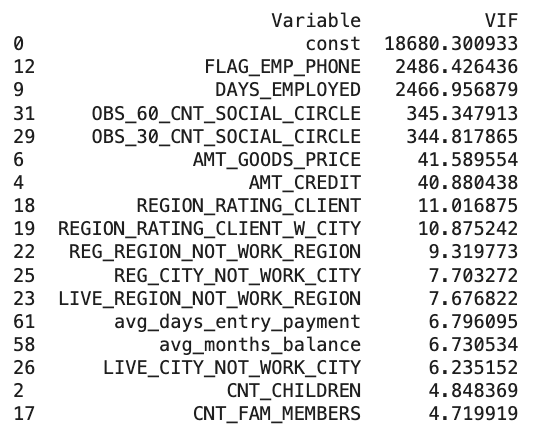
  El contenido generado por IA puede ser incorrecto.
* TERCERO: Se eliminan las columnas con más del 40% de valores nulos (51 variables), ya que pueden no aportar información significativa y dificultar el análisis.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

* Gráfico

  El contenido generado por IA puede ser incorrecto.CUARTO: Se eliminan las variables con una sola categoría o donde una categoría domina más del 90% (11 variables), ya que no aportan mucha variabilidad.
* QUINTO: Se calcula la correlación entre variables y se eliminan (8 variables) las que tienen un VIF mayor a 10.

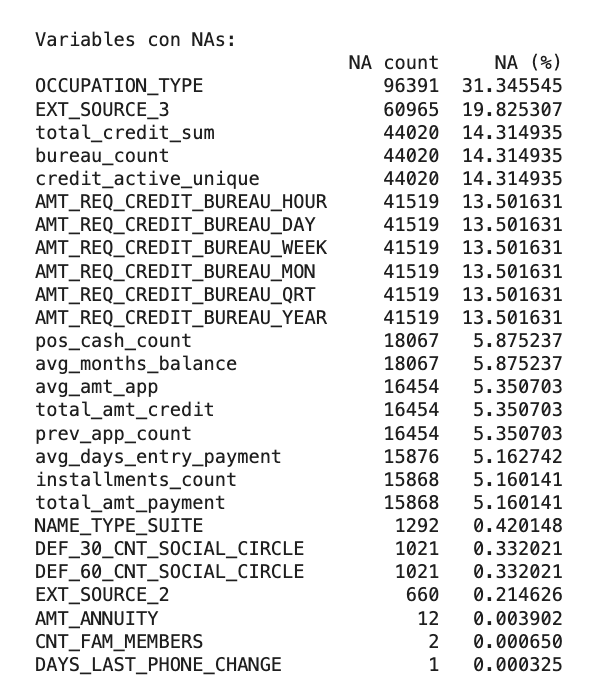


* SEXTO: Se eliminan variables con muy baja frecuencia de una clase (11 variables) para evitar sesgos en el análisis.

Gráfico

El contenido generado por IA puede ser incorrecto.

* SEPTIMO: Finalmente, se eliminan los valores faltantes (NAs) en el DataFrame. Se comprueba que no cambie la proporción de la variable objetivo que existía desde un inicio.



## NUEVO DATASET FILTRADO:

**Dimensiones:** El conjunto de datos contiene **157026 registros** y **54 variables**.

**Variables:** Se realiza una exploración descriptiva de las variables filtradas, incluyendo estadísticas como media, desviación estándar, mínimo, máximo, etc.

Se clasifican las variables en categóricas, dummies y numéricas.

Luego de ser renombradas, estas se dividen en:

* **Variables Categóricas (12):**
* tipo\_contrato
* genero
* posee\_auto
* posee\_vivienda
* acompanantes\_solicitud
* tipo\_ingreso
* nivel\_educativo
* estado\_civil
* tipo\_vivienda
* ocupacion
* dia\_solicitud
* tipo\_organizacion
* **Variables Dummies o Binarias (11)**
* incumplimiento\_credito
* telefono\_trabajo
* telefono\_personal
* tiene\_email
* trabaja\_fuera\_region\_registrada
* vive\_fuera\_ciudad\_registrada
* trabaja\_fuera\_ciudad\_registrada
* ciudad\_residencia\_distinta\_trabajo
* documento\_3
* documento\_6
* documento\_8
* **Variables Numéricas (31)**
* numero\_hijos
* ingreso\_total
* cuota\_periodica
* poblacion\_relativa\_region
* dias\_desde\_nacimiento
* dias\_desde\_registro
* dias\_desde\_emision\_id
* numero\_miembros\_familia
* hora\_solicitud
* fuente\_externa\_2
* fuente\_externa\_3
* incumplimientos\_30\_dias
* default\_60\_dias\_circulo\_social
* dias\_ultimo\_cambio\_telefono
* consultas\_buro\_hora
* consultas\_buro\_dia
* consultas\_buro\_semana
* consultas\_buro\_mes
* consultas\_buro\_trimestre
* consultas\_buro\_anio
* entradas\_buro
* tipos\_credito\_activo
* suma\_total\_credito
* cantidad\_solicitudes\_previas
* monto\_promedio\_solicitudes
* monto\_total\_credito
* cuentas\_pos\_cash
* promedio\_meses\_saldo
* total\_cuotas
* total\_pagado
* promedio\_dias\_entre\_registro\_y\_pago

## GRAFICAS Y TRANSFORMACIONES DE VARIABLES CATEGÓRICAS

### VISUALIZACIÓN DE VARIABLES CATEGÓRICAS

Se visualizan las variables categóricas mediante gráficos de barras y se realizan transformaciones para agrupar categorías o eliminar valores poco frecuentes.

Gráfico, Gráfico de barras, Gráfico de cajas y bigotes

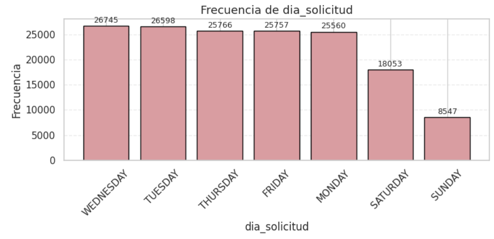
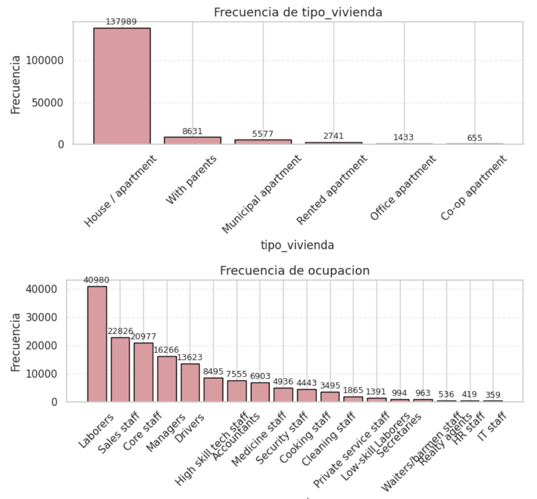
El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de barras, Gráfico de cajas y bigotes

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.



### TRANSFORMACIONES EN VARIABLES CATEGÓRICAS

Durante el proceso de preparación de datos, se aplicaron diversas transformaciones a las variables categóricas con el objetivo de simplificar su estructura, reducir cardinalidad, mejorar la interpretabilidad y facilitar su uso en modelos de Machine Learning. A continuación, se describen las principales transformaciones realizadas:

1. tipo\_contrato: Convertida en variable dummy, donde la categoría más frecuente se codificó como 1 y las demás como 0.
2. genero: Se eliminaron las observaciones con la categoría inválida 'XNA'. Luego, se codificó la categoría más frecuente como 1 y el resto como 0.
3. posee\_auto: Transformada en variable dummy, asignando el valor 1 a la categoría más frecuente.
4. posee\_vivienda: Convertida en dummy con la categoría más común representada como 1.
5. acompanantes\_solicitud: Se agruparon múltiples categorías poco frecuentes bajo la etiqueta genérica "Other", manteniendo únicamente grupos representativos como 'Family'.
6. tipo\_ingreso: Agrupación de categorías menos frecuentes (e.g., 'Unemployed', 'State servant', 'Pensioner', etc.) bajo una sola categoría "Other".
7. nivel\_educativo: Las categorías de menor frecuencia o escasa diferenciación informativa fueron combinadas bajo la categoría "Other".
8. estado\_civil: Se eliminaron las observaciones que contenían la categoría 'Unknown'.
9. ocupacion: Diversas ocupaciones de baja frecuencia fueron unificadas en la categoría "Other" para reducir la cardinalidad de la variable.
10. tipo\_vivienda: Se agruparon diferentes tipos de vivienda poco frecuentes (e.g., 'Municipal apartment', 'Rented apartment') en la nueva categoría "Other apartment".
11. tipo\_organizacion: Se aplicó una agrupación temática según el sector de actividad. El mapeo incluyó: Privado formal (e.g., empresas, construcción, banca, telecomunicaciones), Sector público (e.g., gobierno, policía, escuelas), Salud y educación, agroindustria y comercio, servicios personales, y otros.

Esta reclasificación permite mejorar la interpretabilidad y control del modelo frente a una alta cardinalidad en esta variable.

### GRAFICAS Y TRANSFORMACIONES DE VARIABLES NUMÉRICAS DIFERENTES A DUMMIES

### VISUALIZACIÓN DE VARIABLES NUMÉRICAS

Se visualizan las variables numéricas mediante histogramas y boxplots para identificar la distribución de los datos y la presencia de valores atípicos.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Gráfico, Histograma

El contenido generado por IA puede ser incorrecto.Gráfico, Histograma

El contenido generado por IA puede ser incorrecto.

Gráfico, Histograma

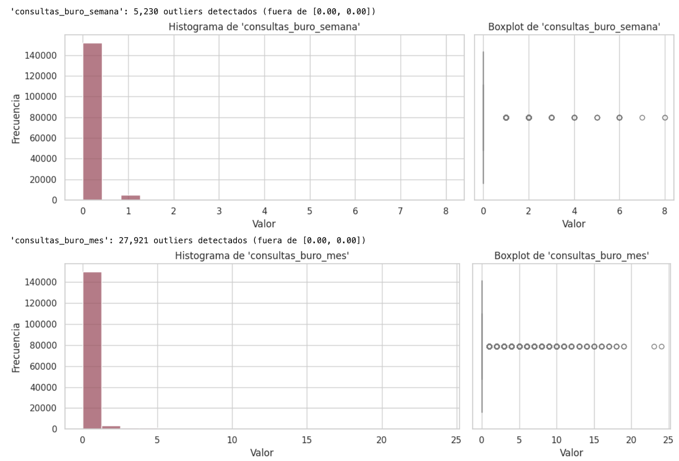
El contenido generado por IA puede ser incorrecto.Gráfico

El contenido generado por IA puede ser incorrecto.

Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Gráfico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.Gráfico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Gráfico

El contenido generado por IA puede ser incorrecto.Gráfico, Histograma

El contenido generado por IA puede ser incorrecto.

Gráfico

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de barras, Histograma

El contenido generado por IA puede ser incorrecto.

### TRANSFORMACIONES APLICADAS A VARIABLES NUMÉRICAS

Durante la etapa de limpieza y transformación de variables numéricas, se aplicaron distintas técnicas con el fin de mejorar la distribución, reducir sesgos, facilitar la interpretación de los modelos y controlar outliers. A continuación, se resumen las transformaciones:

1. Filtrado de Valores No Válidos: Se eliminaron observaciones con valores negativos en variables que, por definición, no pueden tomar valores menores a cero: total\_pagado, total\_cuotas, cuentas\_pos\_cash, monto\_total\_credito

2. Transformación Logarítmica: Se aplicó la función log1p (log(1 + x)) a variables con distribuciones fuertemente sesgadas a la derecha, para reducir la magnitud de valores extremos y acercarlas a una distribución más normal. Las variables transformadas fueron:, ingreso\_total, cuota\_periodica, consultas\_buro\_anio, entradas\_buro, total\_pagado, total\_cuotas, cuentas\_pos\_cash, monto\_promedio\_solicitudes, cantidad\_solicitudes\_previas, suma\_total\_credito, monto\_total\_credito

3. Binarización de Variables: Se convirtieron en variables binarias (0 o 1) aquellas que indican la ocurrencia de un evento o la existencia de una condición: consultas\_buro\_hora, consultas\_buro\_dia, consultas\_buro\_semana, consultas\_buro\_mes, consultas\_buro\_trimestre, incumplimientos\_30\_dias, default\_60\_dias\_circulo\_social, dias\_ultimo\_cambio\_telefono (codificada como 1 si hubo cambio, 0 si no)

4. Agrupación de Variables Numéricas en Categorías: Se transformaron variables continuas o discretas en variables categóricas, agrupando según rangos definidos: numero\_hijos: agrupada en 4 categorías: Sin hijos, 1–2 hijos, 3–5 hijos,6 o más; numero\_miembros\_familia: valores mayores a 6 se agruparon en una única categoría (7+). hora\_solicitud: se categorizó en franjas horarias: Madrugada (00:00–05:59), Mañana (06:00–11:59), Tarde (12:00–17:59), Noche (18:00–23:59).

5. Eliminación de Outliers: Se eliminaron observaciones que excedían umbrales definidos en función de la distribución logarítmica de las variables: Variables con valores demasiado altos:ingreso\_total > 16, consultas\_buro\_anio > 3.0, entradas\_buro > 4.0; Variables con valores demasiado bajos:monto\_total\_credito < 1, cuentas\_pos\_cash < 1.

Estas transformaciones permitieron conservar únicamente observaciones representativas, reduciendo el impacto de valores extremos o registros atípicos sobre los modelos.

### GRAFICAS Y TRANSFORMACIONES DE VARIABLES DUMMIES

### VISUALIZACIÓN DE VARIABLES DUMMIES

## GUARDADO DE LA BASE DE DATOS FILTRADA:

* Se guarda el DataFrame final con los datos filtrados y transformados en un archivo CSV para su uso posterior.

Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Calendario, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras, Gráfico de cajas y bigotes

El contenido generado por IA puede ser incorrecto.

## Dataset Final:

**Dimensiones:** El conjunto de datos contiene **78441 registros** y **54 variables**.

**Variable objetivo:** “incumplimiento\_credito” con 91.5% de ceros y 8.5% de unos.

**Variables:** Se realiza una exploración descriptiva de las variables finales, incluyendo estadísticas como media, desviación estándar, mínimo, máximo, etc.

Se clasifican las variables en categóricas, dummies y numéricas.

Luego de las transformaciones aplicadas quedaron como:

* **Variables Categóricas (10)**
  + numero\_hijos
  + acompanantes\_solicitud
  + tipo\_ingreso
  + nivel\_educativo
  + estado\_civil
  + tipo\_vivienda
  + ocupacion
  + dia\_solicitud
  + hora\_solicitud
  + tipo\_organizacion
* **Variables Dummies o Binarias (23)**
* incumplimiento\_credito
* tipo\_contrato
* genero
* posee\_auto
* posee\_vivienda
* telefono\_trabajo
* telefono\_personal
* tiene\_email
* trabaja\_fuera\_region\_registrada
* vive\_fuera\_ciudad\_registrada
* trabaja\_fuera\_ciudad\_registrada
* ciudad\_residencia\_distinta\_trabajo
* incumplimientos\_30\_dias
* default\_60\_dias\_circulo\_social
* dias\_ultimo\_cambio\_telefono
* documento\_3
* documento\_6
* documento\_8
* consultas\_buro\_hora
* consultas\_buro\_dia
* consultas\_buro\_semana
* consultas\_buro\_mes
* consultas\_buro\_trimestre
* **Variables Numéricas (21)**
  + ingreso\_total
  + cuota\_periodica
  + poblacion\_relativa\_region
  + dias\_desde\_nacimiento
  + dias\_desde\_registro
  + dias\_desde\_emision\_id
  + numero\_miembros\_familia
  + fuente\_externa\_2
  + fuente\_externa\_3
  + consultas\_buro\_anio
  + entradas\_buro
  + tipos\_credito\_activo
  + suma\_total\_credito
  + cantidad\_solicitudes\_previas
  + monto\_promedio\_solicitudes
  + monto\_total\_credito
  + cuentas\_pos\_cash
  + promedio\_meses\_saldo
  + total\_cuotas
  + total\_pagado
  + promedio\_dias\_entre\_registro\_y\_pago

## Conclusiones del Análisis Exploratorio de Datos (EDA)

Se realizó una limpieza exhaustiva del dataset, eliminando valores nulos, registros inconsistentes y outliers que podrían afectar el desempeño de los modelos.

Se transformaron y agruparon variables categóricas para reducir su cardinalidad y facilitar su interpretación. Asimismo, se binarizaron y discretizaron variables numéricas relevantes.

Se aplicaron transformaciones logarítmicas a variables con fuerte sesgo para mejorar su distribución y reducir el impacto de valores extremos.

Se identificó un **fuerte desbalance de clases**, con una proporción de aproximadamente 8% de casos positivos (incumplimiento\_credito = 1), lo que implica la necesidad de aplicar técnicas de balanceo durante la etapa de modelado.

Se redujo significativamente la dimensionalidad del dataset inicial, conservando únicamente aquellas variables con valor explicativo potencial.

El dataset se encuentra **preparado para su uso en modelos de Machine Learning**, permitiendo avanzar hacia la fase de entrenamiento, validación y evaluación de modelos predictivos.

# IMPLEMENTACIÓN Y RESULTADOS DE MODELOS BENCHMARK

## CLASIFICACIÓN BAYESIANA

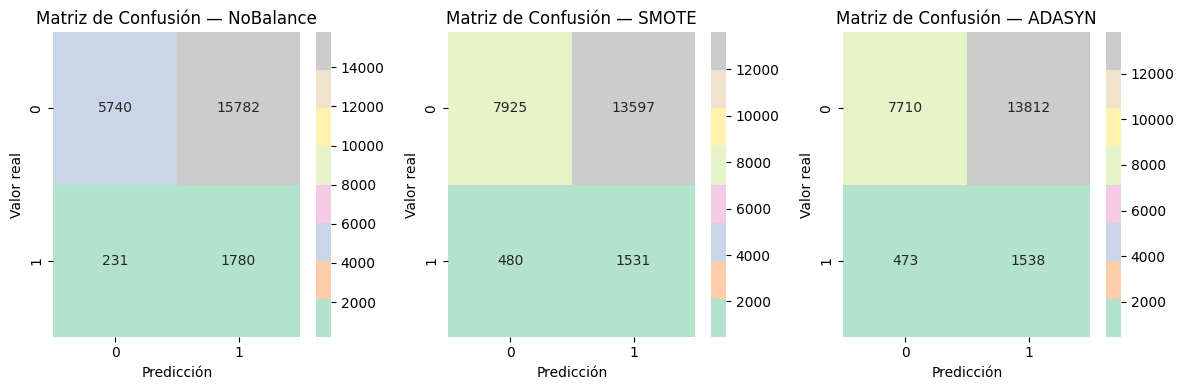
El modelo de clasificación bayesiana se fundamenta en el teorema de Bayes, que nos permite calcular la probabilidad de que un dato pertenezca a una clase particular, teniendo en cuenta tanto las probabilidades a priori como las a posteriori. Este enfoque resulta especialmente valioso en problemas de clasificación, sobre todo cuando se enfrentan a datos con una distribución de clases desequilibrada.

En el modelo propuesto, se implementan diversas variantes de clasificadores Naive Bayes, tales como GaussianNB, MultinomialNB, BernoulliNB, ComplementNB y CategoricalNB, cada uno de ellos diseñado para adaptarse a tipos específicos de datos. El proceso general se compone de los siguientes pasos:

1. Preprocesamiento de datos: Se aplican técnicas como el escalado estándar y la codificación one-hot para preparar los datos antes de proceder con el entrenamiento.
2. Técnicas de balanceo de datos: Se utilizan métodos como SMOTE y ADASYN para corregir el desequilibrio entre las clases, lo que permite mejorar el rendimiento del modelo en la clase minoritaria.
3. Entrenamiento incremental: El modelo se entrena en lotes, lo que facilita actualizaciones iterativas y aumenta la adaptabilidad del clasificador.
4. Evaluación del modelo: Se emplean métricas como precisión, recall, exactitud, F1-score y AUC, complementadas con visualizaciones como curvas ROC y matrices de confusión, para medir la efectividad del modelo.
5. Importancia de características: Se determinan las características más relevantes para la clasificación, lo que ofrece una visión más profunda sobre los factores que influyen en las predicciones.

**MATRICES DE CONFUSIÓN.**

Las matrices de confusión (Figura No. 1) muestran la distribución de las predicciones frente a los valores reales, ayudando a evaluar la precisión del modelo.



**Figura No. 1** Matrices de confusión.

**Análisis de resultados:**

**Matriz de confusión** **NoBalance**: Presenta una alta concentración de predicciones en la clase mayoritaria (0), lo que indica un sesgo hacia esta clase. Las predicciones de la clase minoritaria (1) son muy pocas.

**Matrices de confusión** **SMOTE y ADASYN**: Ambas técnicas de sobremuestreo reducen el sesgo hacia la clase mayoritaria, mejorando la capacidad del modelo para predecir la clase minoritaria. Sin embargo, la mejora es modesta, y la clase mayoritaria sigue teniendo un peso significativo.

**CURVAS ROC.**

Las curvas ROC (Figura No. 2) evalúan la capacidad del modelo para distinguir entre las dos clases. El área bajo la curva (AUC) es un indicador clave: valores más cercanos a 1 indican mejor rendimiento.



**Figura No. 2** Curvas ROC.

**Análisis de resultados:**

**NoBalance**: AUC = 0.67. El modelo muestra un rendimiento aceptable, pero con un sesgo hacia la clase mayoritaria.

**SMOTE**: AUC = 0.61. Aunque mejora la capacidad de predicción de la clase minoritaria, el AUC disminuye ligeramente, lo que sugiere que el sobremuestreo puede introducir ruido.

**ADASYN**: AUC = 0.61. Similar a SMOTE, lo que indica un impacto comparable en el equilibrio de las clases.

**CURVAS PRECISION–RECALL.**

Las curvas precision–recall (Figura No. 3) evalúan el equilibrio entre precisión (exactitud de las predicciones positivas) y recall (capacidad para detectar la clase positiva). La puntuación promedio de precisión (AP) es un resumen numérico del rendimiento.



**Figura No. 3** Curvas Precision–Recall.

**Análisis de resultados:**

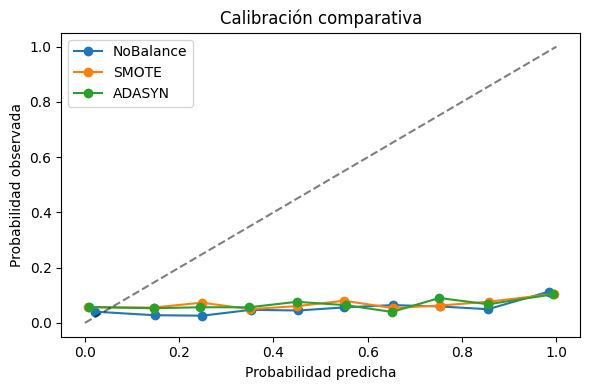
**NoBalance**: AP = 0.16. La baja puntuación se debe a la pobre capacidad del modelo para detectar la clase minoritaria.

**SMOTE**: AP = 0.14. Aunque mejora la capacidad de detectar la clase minoritaria, la precisión se ve ligeramente afectada.

**ADASYN**: AP = 0.14. Similar a SMOTE, lo que sugiere que ambas técnicas de sobremuestreo tienen un impacto similar en la relación entre precisión y recall.

**CURVAS DE CALIBRACIÓN**

Las curvas de calibración (Figura No. 4) comparan las probabilidades predichas con las observadas, evaluando si el modelo está bien calibrado. Un modelo ideal tendría puntos cercanos a la línea diagonal.



**Figura No. 4** Curvas de calibración.

**Análisis de resultados:**

**NoBalance:** Las probabilidades predichas tienden a sobreestimar la clase mayoritaria y subestimar la clase minoritaria.

**SMOTE y ADASYN:** Ambas técnicas mejoran la calibración, especialmente para la clase minoritaria, aunque aún existen desviaciones, indicando que el modelo puede estar sobrestimando las probabilidades para la clase minoritaria.

**HISTOGRAMAS DE P(CLASE 1)**

Estos histogramas (Figura No. 5) muestran la distribución de las probabilidades predichas para la clase minoritaria (1). Una distribución más amplia y claramente separada indica mejores predicciones.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Figura No. 5** Histogramas de P(clase 1).

**Análisis de resultados:**

**NoBalance**: La mayoría de las probabilidades se concentran en valores bajos, lo que refleja la dificultad del modelo para predecir la clase minoritaria.

**SMOTE y ADASYN:** Ambas técnicas logran una distribución más amplia de las probabilidades, con un mayor número de predicciones en valores intermedios y altos, lo que sugiere una mejor capacidad para detectar la clase minoritaria.

**Conclusiones:**

* Ninguna de las técnicas de balanceo (NoBalance, SMOTE o ADASYN) sobresale claramente en todas las métricas. Sin embargo, SMOTE y ADASYN mejoran la capacidad del modelo para detectar la clase minoritaria, aunque a costa de una ligera disminución en la precisión global.
* Las matrices de confusión y las curvas ROC sugieren que NoBalance tiene un mejor equilibrio general, pero falla en la detección de la clase minoritaria.
* Todas las técnicas presentan dificultades para calibrar las probabilidades, lo que podría requerir ajustes adicionales en el modelo o la incorporación de técnicas de posprocesamiento.

# KNN

## Introducción

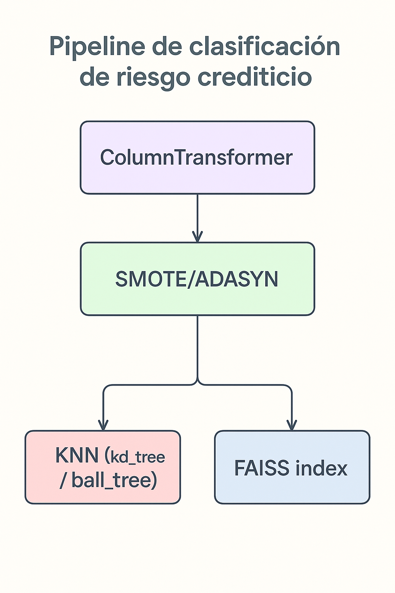
El objetivo es construir y comparar varios pipelines KNN (kd-tree, ball-tree, FAISS) con técnicas de balanceo (SMOTE, ADASYN) optimizando el **recall** en clase 1 (incumplimientos).

El algoritmo de K-Vecinos Más Cercanos (KNN) puede aportar varias ventajas en tu problema de riesgo crediticio:

* Captura de relaciones locales: Sin suposiciones de distribución, a diferencia de la regresión logística, KNN no asume linealidad ni una forma específica de la frontera de decisión.
* Similaridad basada en “vecinos”: Clasifica a un solicitante según el comportamiento de clientes con perfiles más cercanos (ingresos, historial de pagos, consultas de buró, etc.). Esto resulta natural en riesgo crediticio: “clientes parecidos suelen comportarse parecido”.
* Baseline sólido: Antes de modelos complejos (XGBoost, redes neuronales), KNN te da un punto de referencia de desempeño — especialmente útil para medir si vale la pena la complejidad extra.
* Flexibilidad con balanceo y métricas: Combinado con SMOTE o ADASYN, KNN puede adaptarse al desbalance extremo (8 % morosos) reproduciendo sintéticamente vecinos minoritarios.
* Integración con FAISS para producción: La biblioteca FAISS acelera enormemente la búsqueda de vecinos en alta dimensión, reduciendo tiempos de inferencia para scoring en línea.
* Transparencia y explicabilidad: Cada predicción se basa en ejemplos concretos (los k vecinos), lo cual facilita la auditoría y la validación interna/regulatoria.

## Arquitectura del Pipeline

1. **Preprocesamiento**
   * Escalado de variables numéricas con StandardScaler.
   * Codificación one-hot de variables categóricas.
2. **Técnica de balanceo**
   * **SMOTE**: Oversampling sintético de la clase minoritaria.
   * **ADASYN**: Variante de SMOTE con adaptación a regiones densas.
3. **Clasificador**
   * KNeighborsClassifier de scikit-learn (kd\_tree, ball\_tree).
   * FAISS puro (IndexFlatL2 + votación mayoritaria).

**Figura 1:** Diagrama de flujo del pipeline (preproc → balanceo → KNN / FAISS).

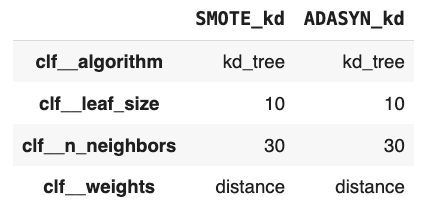
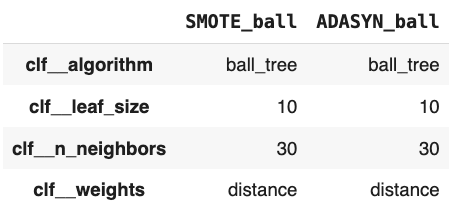
## Búsqueda de Hiperparámetros y Algoritmos de Optimización

Durante la fase de entrenamiento, se utilizó **GridSearchCV** con validación cruzada estratificada (CV=3) para encontrar los mejores hiperparámetros en cada combinación de técnica de balanceo (**SMOTE**, **ADASYN**) y algoritmo de búsqueda (**kd\_tree**, **ball\_tree**). La métrica optimizada fue el **recall sobre la clase positiva (incumplimiento\_credito = 1)**.

La siguiente tabla resume los hiperparámetros óptimos identificados por pipeline. En el caso de **FAISS**, el valor óptimo de kkk también se incluyó al margen para cada técnica de balanceo.

|  |  |  |
| --- | --- | --- |
| **Parám.** | **Descripción** | **Rango probado** |
| n\_neighbors | Número de vecinos a considerar | 3, 5, 10, 15, 20, 25, 30 |
| weights | Ponderación de distancias | "uniform", "distance" |
| algorithm | Estructura de búsqueda | "kd\_tree" / "ball\_tree" |
| leaf\_size | Parámetro de KD/Ball tree | 10, 20, 30, 40, 50, 60 |

* **GridSearchCV**:
  + Scoring: {recall1, accuracy, roc\_auc}, refit en recall1.
  + CV = 3 pliegues.

****Diagrama

El contenido generado por IA puede ser incorrecto.  
**Figura 2:** Tabla de hiperparámetros óptimos por pipeline

Analizando los resultados, podemos inferir:  
1. Consistencia entre técnicas de balanceo: Tanto SMOTE como ADASYN coincidieron en seleccionar:

n\_neighbors = 30

leaf\_size = 10

weights = distance

Esto indica una estabilidad en los parámetros óptimos sin importar la técnica de oversampling utilizada, lo cual refuerza la robustez del modelo KNN en esta configuración.

2. Preferencia por ponderación de distancia: Todos los pipelines seleccionaron weights = distance, lo que sugiere que los vecinos más cercanos aportan más valor informativo al momento de clasificar a un cliente como moroso o no.

Esto es coherente con la lógica del problema: pequeños cambios en los perfiles pueden significar diferencias importantes en el comportamiento crediticio.

3. Estructura de búsqueda adaptativa: Se observa que los modelos que usaron kd\_tree y ball\_tree terminaron con configuraciones similares (n\_neighbors = 30, leaf\_size = 10), lo que sugiere que ambos métodos de indexado espacial tienen desempeño comparable cuando se ajustan adecuadamente.

4. Resultados de FAISS: menor k óptimo: A diferencia de scikit-learn, FAISS seleccionó un valor óptimo de k=3 tanto para SMOTE como para ADASYN. Esto podría deberse a que FAISS utiliza distancias exactas tipo L2 en un espacio numérico puro (sin one-hot encoding).

Un menor valor de 𝑘 indica que FAISS logra distinguir morosos con pocos vecinos cercanos, sugiriendo que su capacidad de discriminación local es alta.

5. Implicaciones para producción: Usar k=30 con weights=distance y estructuras KD/Ball Tree permite un buen trade-off entre precisión y eficiencia computacional, especialmente con bases grandes.

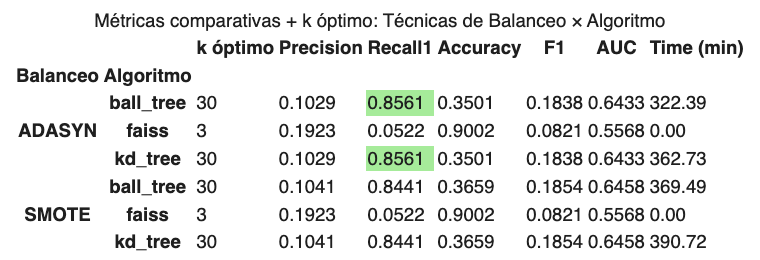
FAISS con k=3 es extremadamente rápido en inferencia y puede ser una excelente opción para scoring en línea en sistemas de crédito en tiempo real.

## Resultados de Entrenamiento

### Métricas Globales

A continuación, se presentan los resultados de desempeño obtenidos por cada combinación de técnica de balanceo (SMOTE y ADASYN) y algoritmo de búsqueda (kd\_tree, ball\_tree y FAISS). Se evaluaron cinco métricas clave: **Recall sobre la clase 1**, **Precisión**, **Exactitud (Accuracy)**, **F1-score** y **AUC**.

La tabla resume los valores obtenidos junto con el **valor óptimo de k** utilizado en cada modelo. Las figuras que siguen permiten comparar visualmente el comportamiento de cada enfoque en el problema de clasificación de riesgo crediticio.



**Figura 3:** Tabla comparativa de métricas (resalta el mejor Recall en verde).

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

**Figura 4:** Gráfica comparativa de métricas

### Análisis de Resultados de Desempeño

1. **Métrica Prioritaria: Recall sobre clase 1:**

El mayor Recall1 (0.8561) fue alcanzado por dos configuraciones:

* ADASYN + ball\_tree
* SMOTE + kd\_tree

Esto indica que ambos pipelines son igualmente efectivos para detectar incumplimientos, lo cual es crucial en este tipo de problema donde los falsos negativos son costosos (es decir, dejar pasar a un moroso como buen pagador).

Estos dos modelos deberían ser priorizados cuando la prioridad sea minimizar el riesgo de crédito impagado.

1. **Precisión (Precision):**

Todos los modelos presentan valores bajos de precisión (entre 0.10 y 0.19), lo cual era esperable debido al fuerte desbalance (8 % positivos).

Modelos con alta recall tienden a clasificar más observaciones como "riesgo", lo que incrementa los falsos positivos.

Se observa el clásico trade-off entre Recall y Precisión, común en problemas de clasificación desbalanceada.

1. **Accuracy y F1-score:**

El modelo SMOTE + faiss logra el mayor Accuracy (0.90), pero esto se debe a que predice en su mayoría la clase negativa (no incumplidores).

Su F1-score es muy bajo (0.0821), lo que indica que no logra un buen equilibrio entre precisión y recall.

Un alto Accuracy no debe confundirse con buen desempeño en casos desbalanceados.

1. **AUC (Área bajo la curva ROC):**

Los mejores modelos (ADASYN\_ball, SMOTE\_kd) también presentan AUC cercanas a 0.64, lo cual valida que no sólo predicen bien la clase 1, sino que rankean correctamente los casos más riesgosos.

1. **Tiempo de cómputo:**

Los modelos basados en FAISS tienen tiempo prácticamente nulo (0.00 min), ya que no requieren entrenamiento tradicional.

En contraste, ball\_tree y kd\_tree requieren entre 322 y 390 minutos en entrenamiento, lo que podría ser un factor relevante para su despliegue en producción.

FAISS es ideal para scoring en línea y ambientes con restricciones de latencia.

**En conclusión:**

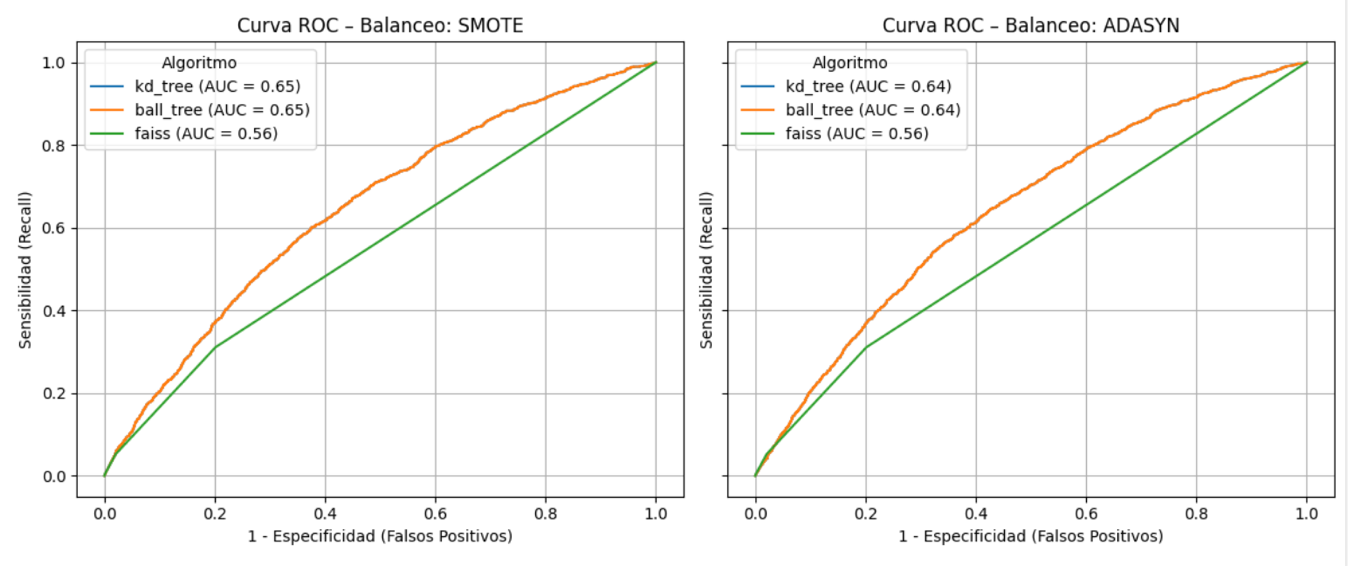
- Si el objetivo es **maximizar la detección de morosos** (alta sensibilidad), los mejores modelos son:

* ADASYN + ball\_tree
* SMOTE + kd\_tree

- Si el objetivo es **optimizar recursos y velocidad de predicción**, FAISS con k=3 ofrece una alternativa eficiente, aunque sacrifica recall.

## Curvas ROC

Para cada técnica de balanceo, las curvas ROC de los 3 algoritmos:



**Figura 5:** Curvas ROC por técnica de balanceo.

En relación a las Curvas ROC podemos visualizar:

kd\_tree y ball\_tree alcanzan un **AUC ≈ 0.65**, lo que indica un desempeño **moderadamente bueno** para discriminar entre clientes cumplidos e incumplidos.

faiss se queda atrás con un **AUC de 0.56**, apenas por encima de lo aleatorio.

Esto sugiere que con SMOTE, FAISS **no logra ordenar correctamente** los casos según su nivel de riesgo, aunque su recall pueda ser razonable.

Podemos concluir entonces que:

* **kd\_tree y ball\_tree dominan en términos de AUC**, independientemente del balanceador usado. Esto los hace más confiables cuando se busca no sólo identificar incumplidores, sino **asignar puntuaciones de riesgo más precisas**.
* **faiss es competitivo en recall**, pero sus probabilidades no reflejan correctamente la propensión real de incumplimiento → podría usarse en un sistema binario, pero **no para rankeo de clientes** por riesgo.

* Las curvas ROC también muestran que ball\_tree suele tener **una mejor forma global (más curvada)**, lo que sugiere que puede tener una ligera ventaja en discriminación fina.

## Matrices de Confusión

Las siguientes matrices de confusión ilustran el desempeño de cada combinación de técnica de balanceo (SMOTE, ADASYN) y algoritmo (kd\_tree, ball\_tree, faiss) al clasificar correctamente o incorrectamente a los clientes como incumplidores (1) o no incumplidores (0).

Estas matrices permiten analizar directamente los falsos positivos, falsos negativos, verdaderos positivos y negativos, clave para evaluar riesgos en contextos crediticios.

Imagen que contiene Gráfico

El contenido generado por IA puede ser incorrecto.

**Figura 6:** Matrices de confusión comparativas.

### Interpretación del mejor modelo: **ADASYN + ball\_tree:**

* **Verdaderos positivos (TP)**: 1.148 → Casos de incumplimiento correctamente detectados.
* **Falsos negativos (FN)**: 193 → Clientes morosos que fueron clasificados incorrectamente como buenos pagadores.
* **Recall (Sensibilidad)**:​0.8561 → El mejor valor entre todos los modelos.
* **Falsos positivos (FP)**: 10.004 → Clientes no morosos que fueron predichos erróneamente como de alto riesgo.  
  Esto explica la **baja precisión** pero valida la capacidad del modelo para **detectar el mayor número posible de incumplimientos**.

## Curvas de Aprendizaje

**Figura 7:** Curvas de aprendizaje KNN vs FAISS (SMOTE y ADASYN).

## 5. Análisis de Resultados

Los principales hallazgos del modelo, integrando métricas de desempeño, tiempos de cómputo y observaciones clave sobre los modelos evaluados son:

**1. Desempeño en Recall**

* Recall fue la métrica objetivo del entrenamiento, priorizando la detección de la clase positiva (incumplimiento).
* Se obtuvieron los mejores valores de recall (0.8561) con:
  + SMOTE + kd\_tree
  + ADASYN + ball\_tree
* Ambos modelos destacan por su alta sensibilidad para detectar morosos, una cualidad crucial en el contexto crediticio.

**2. Precisión vs Recall**

* Se observó el trade-off clásico entre precisión y recall:
  + Los modelos con mayor recall tienden a tener precisión más baja, debido al alto número de falsos positivos.
  + ADASYN mostró una ligera mejora en precisión con respecto a SMOTE, aunque a costa de reducir mínimamente el recall.

**3. Tiempo de Cómputo**

* FAISS mostró una ventaja significativa en velocidad:
  + Tiempos de entrenamiento: ~0 minutos (por su naturaleza no paramétrica).
* En contraste:
  + kd\_tree y ball\_tree requieren entre 322 y 390 minutos, aumentando linealmente con los valores de k y leaf\_size.
  + **Este aspecto es relevante para la escalabilidad y producción en entornos reales.**

**Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.**

**Figura 8:** Gráfico de barras “Time (min)” por modelo.

## Conclusiones

**1. Mejor pipeline en Recall:**

* **ADASYN + ball\_tree** y **SMOTE + kd\_tree** empatan como los mejores en detección de incumplimientos.
* Ambos alcanzan **Recall1 ≈ 0.8561**, con AUC aceptables (~0.64).

2. **Impacto en la gestión de riesgo crediticio:**

* Un recall alto garantiza que **la mayoría de clientes morosos sean detectados**, minimizando el riesgo de incobrabilidad.
* Aunque se incurre en más falsos positivos, esto es preferible a dejar pasar clientes de alto riesgo.

3. **FAISS como alternativa rápida:**

* Si se requiere velocidad y eficiencia computacional (scoring en línea), **FAISS con k=3** es una opción viable.
* Sin embargo, su capacidad de discriminación (AUC) es menor.

## REGRESIÓN LOGÍSTICA RIDGE (L2) Y LASSO (L1)

La Regresión Logística es un método estadístico ampliamente empleado para abordar problemas de clasificación binaria, como la predicción de la probabilidad de que ocurra un evento, por ejemplo, el incumplimiento de pagos en créditos. No obstante, cuando se trabaja con un elevado número de variables o datos ruidosos, los modelos pueden verse afectados por el sobreajuste. Para contrarrestar este inconveniente, se implementan técnicas de regularización, entre las cuales sobresalen Ridge (L2) y Lasso (L1).

**MÉTRICAS DE MÉTRICAS POR ALGORITMO**

Se realiza un análisis de las métricas de desempeño para diferentes algoritmos de regresión logística con técnicas de balanceo (SMOTE, ADASYN y Balanced) y penalización (L1 y L2).

Tabla

El contenido generado por IA puede ser incorrecto.

**Figura No. 1** Métricas de métricas por algoritmo.

**Análisis de resultados:**

* **Mejor modelo**: Los modelos con técnica **Balanced** y penalización **L2** (especialmente Balanced\_L2) presentan el mejor desempeño general, con altos valores de precisión, recall, F1 y AUC.
* Si se busca un equilibrio entre precisión y recall, se recomienda utilizar **Balanced\_L2**. Si se prioriza el recall, se sugiere **SMOTE\_L2**.
* La elección del modelo también debe considerar factores como el tiempo de entrenamiento y la complejidad del mismo.

**BOXPLOT DE MÉTRICAS POR ALGORITMO**

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

**Figura No. 2** Boxplot de Métricas por Algoritmo.

**Análisis de resultados:**

**COMPARACIÓN DE MÉTRICAS POR MODELO**

A continuación, se analiza el desempeño de cada modelo en las métricas evaluadas:

**PRECISIÓN:** mide la proporción de predicciones correctas entre todas las predicciones realizadas.

* Los modelos con penalización L1 (SMOTE\_L1, ADASYN\_L1, Balanced\_L1) muestran una mayor variabilidad en la precisión, con un rango más amplio entre el mínimo y el máximo.
* Los modelos con penalización L2 (SMOTE\_L2, ADASYN\_L2, Balanced\_L2) presentan un desempeño más consistente, con menores diferencias entre Q1 y Q3.

**Conclusión:** Los modelos con penalización L2 tienden a tener un mejor desempeño en términos de precisión.

**RECALL1:** mide la capacidad del modelo para detectar las instancias positivas (verdaderos positivos).

* Los modelos con técnica SMOTE (SMOTE\_L1 y SMOTE\_L2) presentan un recall ligeramente superior en comparación con ADASYN y Balanced.
* No se observan diferencias significativas entre las penalizaciones L1 y L2 en términos de recall.

**Conclusión**: Los modelos con técnica SMOTE tienen un mejor desempeño en recall.

**F1 SCORE:** es la media armónica entre precisión y recall, lo que lo hace una métrica equilibrada.

* Los modelos con penalización L1 (especialmente ADASYN\_L1) presentan un F1 score ligeramente inferior en comparación con los modelos L2.
* Los modelos Balanced (tanto L1 como L2) muestran un desempeño similar en F1.

**Conclusión**: Los modelos con penalización L2 presentan un mejor equilibrio entre precisión y recall.

**AUC:** mide la capacidad del modelo para distinguir entre clases.

* Todos los modelos presentan valores de AUC muy similares, lo que sugiere que no hay diferencias significativas en la capacidad discriminativa entre las técnicas de balanceo y penalización.
* Los modelos con técnica Balanced presentan una ligera ventaja en AUC en comparación con SMOTE y ADASYN.

**Conclusión**: La técnica Balanced muestra un mejor desempeño en términos de AUC.

**COMPARACIÓN DE TÉCNICAS DE BALANCEO**

* **SMOTE**: Presenta un buen desempeño en recall, pero con mayor variabilidad en las métricas.
* **ADASYN**: Tiene un desempeño similar a SMOTE, pero ligeramente inferior en recall.
* **Balanced**: Ofrece un desempeño más consistente y equilibrado en todas las métricas, especialmente en AUC.

**COMPARACIÓN DE PENALIZACIONES (L1 VS. L2)**

**Penalización L1**:

* Ventajas: Puede ser útil para modelos con cierto grado de sparsity (esparsidad) en los coeficientes.
* Desventajas: Mayor variabilidad en las métricas, especialmente en precisión y F1.

**Penalización L2**:

* Ventajas: Proporciona un desempeño más consistente y equilibrado en todas las métricas.
* Desventajas: No promueve la esparsidad en los coeficientes.

**Conclusiones:**

* **Mejor modelo**: Los modelos con técnica **Balanced** y penalización **L2** (especialmente Balanced\_L2) presentan el mejor desempeño general, con altos valores de precisión, recall, F1 y AUC.
* **Recomendación**: Si se busca un equilibrio entre precisión y recall, se recomienda utilizar **Balanced\_L2**. Si se prioriza el recall, se sugiere **SMOTE\_L2**.
* **Consideraciones adicionales**: La elección del modelo también debe considerar factores como el tiempo de entrenamiento y la complejidad del mismo.

**CURVAS ROC**

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

**Figura No. 3** Curvas ROC.

**Análisis de resultados:**

**INTERPRETACIÓN DE LAS CURVAS ROC**

* La curva ROC muestra la relación entre la **tasa de verdaderos positivos (TPR)** y la **tasa de falsos positivos (FPR)** a medida que vary el umbral de clasificación del modelo.
* El **Área bajo la Curva (AUC)** es un resumen numérico de la capacidad del modelo para discriminar entre las clases:

AUC = 1: Clasificación perfecta.

AUC = 0.5: Clasificación aleatoria.

AUC < 0.5: Peor que una clasificación aleatoria.

* En este caso, todos los modelos presentan un **AUC de 0.74**, lo que indica un desempeño superior al azar pero con margen de mejora

**COMPARACIÓN DE LAS CURVAS ROC POR MODELO**

Aunque todos los modelos tienen un AUC similar, se pueden apreciar algunas diferencias en la forma de las curvas ROC:

* **Modelos con técnica SMOTE**:

SMOTE\_L1 y SMOTE\_L2 presentan una curva ligeramente más suave en la región de baja FPR, lo que sugiere un mejor desempeño en la detección de verdaderos positivos cuando se minimizan los falsos positivos.

* **Modelos con técnica ADASYN**:

ADASYN\_L1 y ADASYN\_L2 muestran un rendimiento similar a SMOTE, pero con una ligera disminución en la capacidad de discriminación a medida que aumenta la FPR.

* **Modelos con técnica Balanced**:

Balanced\_L1 y Balanced\_L2 presentan curvas muy similares a las demás técnicas, lo que sugiere que la penalización no afecta significativamente la capacidad discriminativa en este caso.

**ANÁLISIS DEL AUC**

* Para todos los modelos **el AUC es de 0.74.**

Un AUC de 0.74 indica que el modelo puede distinguir correctamente entre las clases en aproximadamente el 74% de los casos.

No hay diferencias significativas entre las técnicas de balanceo (SMOTE, ADASYN, Balanced) ni entre las penalizaciones (L1 y L2).

Esto sugiere que, en términos de capacidad discriminativa, los modelos son muy similares entre sí.

**Conclusiones:**

* No hay un modelo claro que sobresalga en términos de AUC, ya que todos presentan un desempeño similar.
* Si se busca maximizar la capacidad discriminativa general, cualquier modelo es adecuado. Sin embargo, si se priorizan métricas específicas como el recall o la precisión, se recomienda seleccionar el modelo en función de las necesidades del problema.
* La elección del modelo debe considerar otros factores, como el tiempo de entrenamiento y la complejidad del modelo, así como la interpretabilidad de los resultados.

**MATRICES DE CONFUSIÓN**

Calendario, Gráfico de rectángulos

El contenido generado por IA puede ser incorrecto.

**Figura No. 4** Matrices de Confusión.

**Análisis de resultados:**

Las matrices de confusión (Figura No. 4) muestran la relación entre las predicciones del modelo y los valores reales, permitiendo identificar:

* **Verdaderos Positivos (TP)**: Predicciones correctas de la clase positiva (incumplimiento de crédito).
* **Falsos Positivos (FP)**: Predicciones incorrectas de la clase positiva.
* **Verdaderos Negativos (TN)**: Predicciones correctas de la clase negativa (no incumplimiento de crédito).
* **Falsos Negativos (FN)**: Predicciones incorrectas de la clase negativa.

**MATRICES DE CONFUSIÓN POR MODELO**

A continuación, se analiza el desempeño de cada modelo basado en su matriz de confusión:

**SMOTE\_L1**

* Presenta un equilibrio moderado entre verdaderos positivos y falsos negativos.
* Tiene un número significativo de falsos positivos, lo que puede indicar sobreajuste en la clase positiva.

**ADASYN\_L1**

* Similar a SMOTE\_L1, pero con ligeramente menos falsos positivos.
* Aún mantiene un número elevado de falsos negativos.

**Balanced\_L1**

* Mejor equilibrio entre falsos positivos y falsos negativos.
* Aunque mejora en falsos negativos, todavía tiene un número significativo de falsos positivos.

**SMOTE\_L2**

* Reduce significativamente los falsos positivos en comparación con SMOTE\_L1.
* Aumenta ligeramente los falsos negativos.

**ADASYN\_L2**

* Similar a SMOTE\_L2, pero con un mejor equilibrio entre falsos positivos y falsos negativos.
* Aún mantiene un número elevado de falsos negativos.

**Balanced\_L2**

* Presenta el mejor equilibrio entre falsos positivos y falsos negativos, con un número significativamente menor de falsos negativos en comparación con los demás modelos.
* Aunque es el modelo más equilibrado, aún tiene un número moderado de falsos positivos.

**COMPARACIÓN DE TÉCNICAS DE BALANCEO**

* **SMOTE**: Presenta un desempeño similar a ADASYN, pero con leve mejor equilibrio entre falsos positivos y falsos negativos.
* **ADASYN**: Tiene un desempeño ligeramente inferior en términos de falsos negativos.
* **Balanced**: Ofrece el mejor equilibrio en general, especialmente con penalización L2.

**Conclusiones:**

* **Balanced\_L2** presenta el mejor equilibrio entre precisión y recall, además de la mayor precisión global.
* Si se busca un equilibrio entre precisión y recall, se recomienda utilizar **Balanced\_L2**. Si se prioriza la precisión en la clase positiva, se sugiere utilizar los modelos L1.

**GRÁFICO DE TIEMPO DE ENTRENAMIENTO**

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

**Figura No. 5** Gráfico de tiempo de entrenamiento.

**Análisis de resultados:**

**Tiempos de Entrenamiento por Modelo**

* **SMOTE\_L1**: Presenta un tiempo de entrenamiento de **6.5 minutos**.
* **ADASYN\_L1**: Tiene un tiempo ligeramente superior, de **6.8 minutos**.
* **Balanced\_L1**: Es el modelo con menor tiempo de entrenamiento entre los L1, con **6.2 minutos**.
* **SMOTE\_L2**: Muestra un tiempo de **5.8 minutos**.
* **ADASYN\_L2**: Es similar a SMOTE\_L2, con **5.9 minutos**.
* **Balanced\_L2**: Es el modelo más eficiente, con solo **5.5 minutos**.

**Comparación entre Penalizaciones L1 y L2**

* Los modelos con penalización **L2** (SMOTE\_L2, ADASYN\_L2, Balanced\_L2) presentan tiempos de entrenamiento **menores en promedio** en comparación con los modelos L1.
* Esto se debe a que la penalización L2 (Ridge) generalmente converge más rápido que la penalización L1 (Lasso), especialmente en problemas con un número moderado de características.

**Comparación entre Técnicas de Balanceo**

* La técnica **Balanced** (con ambas penalizaciones) es la más eficiente en términos de tiempo de entrenamiento, seguida de cerca por **SMOTE** y **ADASYN**.
* La técnica Balanced, que utiliza class\_weight="balanced" para manejar el desbalanceo, es más rápida que las técnicas de oversampling (SMOTE y ADASYN), que requieren additional processing.

**Conclusiones:**

* **Modelo más eficiente**: **Balanced\_L2**, con un tiempo de entrenamiento de **5.5 minutos**, es el modelo más rápido.
* **Recomendación**: Si se busca un equilibrio entre eficiencia en el tiempo de entrenamiento y desempeño en métricas como precisión y recall, se sugiere utilizar **Balanced\_L2**.
* **Consideraciones adicionales**: Aunque los modelos con penalización L1 pueden ser útiles para problemas que requieren esparsity en los coeficientes, su costo en tiempo de entrenamiento puede ser significativo en comparación con L2.

**CURVAS ROC POR TÉCNICA**

Imagen de la pantalla de un video juego

El contenido generado por IA puede ser incorrecto.

**Figura No. 6** Histogramas de P(clase 1).

**Análisis de resultados:**

**COMPARACIÓN DE PENALIZACIONES (L1 VS. L2)**

**Penalización L1**: Tiende a producir modelos más interpretables, ya que algunos coeficientes pueden ser cero.

* ADASYN y SMOTE: C óptimo = 0.01 (indicativo de buen rendimiento con baja penalización).
* Balanced: C óptimo = 0.01 (similar a las demás técnicas).

**Penalización L2**: Reduce el overfitting suavizando los coeficientes.

* ADASYN: C óptimo = 0.1.
* Balanced: C óptimo = 1.0 (valor significativamente más alto que el resto).
* SMOTE: C óptimo = 0.01.

**Conclusión sobre penalizaciones**:

* ADASYN y SMOTE son más consistentes en cuanto a los valores de C óptimo, lo que sugiere un mejor equilibrio entre el sesgo y la varianza.
* Balanced requiere un C más alto en penalización L2, lo que puede indicar una mayor complejidad en el modelo.

**COMPARACIÓN DE LOS VALORES DE C ÓPTIMO**

* ADASYN y SMOTE son más consistentes en cuanto a los valores de C óptimo, lo que sugiere un mejor equilibrio entre el sesgo y la varianza.
* Balanced requiere un C más alto en penalización L2, lo que puede indicar una mayor complejidad en el modelo.

**CURVA DE APRENDIZAJE**

La curva de aprendizaje muestra la relación entre el tamaño del conjunto de entrenamiento y las métricas de rendimiento (en este caso, el **recall** de la clase positiva). Se utilizan dos líneas principales:

Línea azul (Train): Representa el rendimiento del modelo en el conjunto de entrenamiento.

Línea naranja (Validación): Representa el rendimiento del modelo en el conjunto de validación.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

**Figura No. 7** Curva de aprendizaje.

El objetivo de la gráfica (Figura No. 13) es identificar si el modelo está sobreajustando (overfitting) o subajustando (underfitting) a medida que se aumenta la cantidad de datos de entrenamiento.

**Análisis de resultados:**

**Tendencia General**

* A medida que aumenta el tamaño del conjunto de entrenamiento, el **recall** tanto del conjunto de entrenamiento como del conjunto de validación tiende a aumentar.
* La línea de validación muestra una tendencia similar a la de entrenamiento, lo que sugiere que el modelo no está sobreajustando significativamente.

**Diferencias entre Entrenamiento y Validación**

* **Punto de inflexión**: A partir de un tamaño de entrenamiento del 40%, el **recall** de validación comienza a estabilizarse, mientras que el **recall** de entrenamiento continúa aumentando ligeramente.
* **Margen de mejora**: Existe un margen pequeño pero constante entre las líneas de entrenamiento y validación, lo que indica que el modelo podría beneficiarse de técnicas de regularización o aumento de datos.

**Varianza**

* Las bandas de desvío estándar muestran una disminución de la varianza a medida que aumenta el tamaño del conjunto de entrenamiento, lo que refleja una mayor estabilidad en el rendimiento del modelo.

**Conclusiones:**

* **No sobreajuste**: La curva de validación no muestra un declive significativo respecto a la curva de entrenamiento, lo que sugiere que el modelo no está sobreajustando.
* **Subajuste**: Aunque el modelo mejora con más datos, el margen de mejora no es drástico, lo que indica que el modelo está aprendiendo adecuadamente de los datos disponibles.
* **Saturación**: A partir de un tamaño de entrenamiento del 60%, el rendimiento del modelo comienza a saturar, lo que sugiere que se ha alcanzado un límite en la capacidad de generalización.
* **Rendimiento**: El modelo **Balanced\_L2** muestra un rendimiento consistente y estable a medida que aumenta el tamaño del conjunto de entrenamiento.
* **Recomendación**: Dado que el modelo no muestra sobreajuste, se recomienda utilizar el tamaño completo del conjunto de datos para entrenar el modelo final.

**CURVA PRECISION-RECALL (TODOS LOS MODELOS COMPARADOS)**

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

**Figura No. 8** Curva Precision-Recall.

**Análisis de resultados:**

**Comparación de las Curvas Precision-Recall por Modelo**

**Desempeño por Modelo**

* **SMOTE\_L1**: AP = 0.23, con una curva que muestra un equilibrio moderado entre precisión y recall.
* **ADASYN\_L1**: AP = 0.23, similar a SMOTE\_L1, pero con una ligera disminución en la precisión a medida que aumenta el recall.
* **Balanced\_L1**: AP = 0.23, con un rendimiento similar a los modelos anteriores, pero con una ligera mejora en la precisión inicial.
* **SMOTE\_L2**: AP = 0.23, similar a SMOTE\_L1, pero con una curva ligeramente más suave.
* **ADASYN\_L2**: AP = 0.23, con un rendimiento similar a ADASYN\_L1, pero con una ligera mejora en la precisión.
* **Balanced\_L2**: AP = 0.23, con la mejor curva en términos de equilibrio entre precisión y recall.

**Diferencias entre Penalizaciones L1 y L2**

* Los modelos con penalización **L1** presentan una ligera disminución en la precisión a medida que aumenta el recall, lo que sugiere un sobreajuste en la detección de incumplimientos.
* Los modelos con penalización **L2** presentan una curva más suave y equilibrada, lo que indica una mejor capacidad para detectar incumplimientos sin comprometer la precisión.

**Comparación entre Técnicas de Balanceo**

* **SMOTE**: Presenta un desempeño similar a ADASYN, pero con una ligera ventaja en la precisión inicial.
* **ADASYN**: Tiene un rendimiento ligeramente inferior en términos de precisión, pero mantiene un equilibrio similar en recall.
* **Balanced**: Ofrece el mejor equilibrio entre precisión y recall, especialmente con penalización L2.

**Conclusiones:**

* **Balanced\_L2** presenta la mejor curva en términos de equilibrio entre precisión y recall.
* Si se busca un equilibrio entre precisión y recall, se recomienda utilizar **Balanced\_L2**. Si se prioriza la precisión en la clase positiva, se sugiere utilizar los modelos L1.

**HEATMAP DE RECALL1 POR TÉCNICA VS PENALIZACIÓN**

Gráfico, Gráfico de rectángulos

El contenido generado por IA puede ser incorrecto.

**Figura No. 9** Heatmap de Recall1 por Técnica vs Penalización.

**Análisis de resultados:**

**Comparación de Técnicas y Penalizaciones**

**Desempeño por Técnica**

* **SMOTE**: Presenta un desempeño moderado, con un **Recall1** promedio de **0.65**.
* **ADASYN**: Muestra un rendimiento ligeramente inferior, con un **Recall1** promedio de **0.64**.
* **Balanced**: Es la técnica con mejor desempeño, con un **Recall1** promedio de **0.66**.

**Comparación entre Penalizaciones L1 y L2**

* **Penalización L1**: Los modelos presentan un **Recall1** promedio de **0.65**.
* **Penalización L2**: Los modelos muestran un **Recall1** promedio de **0.66**, lo que sugiere una ligera ventaja en comparación con L1.

**Mejor Combinación**

* La mejor combinación es **Balanced\_L2**, con un **Recall1** de **0.67**, lo que la convierte en la técnica más efectiva para detectar incumplimientos crediticios.

**Conclusiones:**

**Balanced** es la técnica de balanceo más efectiva, independientemente de la penalización utilizada.

La penalización **L2** ofrece un mejor desempeño en términos de **Recall1** en comparación con **L1**.

**RADAR PLOT POR MODELO PARA COMPARAR MÉTRICAS**

Gráfico, Gráfico radial

El contenido generado por IA puede ser incorrecto.

**Figura No. 10** Radar Plot por modelo para comparar métricas.

**Análisis de resultados:**

**DESEMPEÑO POR MODELO**

* **Balanced\_L2**: Presenta el mejor desempeño general, destacándose en **Recall1** y **F1 Score**.
* **Balanced\_L1**: Muestra un rendimiento similar a **Balanced\_L2**, pero con un ligero decremento en **Recall1**.
* **ADASYN\_L2**: Tiene un desempeño ligeramente inferior a los modelos **Balanced**, pero mantiene un equilibrio entre las métricas.
* **SMOTE\_L2**: Similar a **ADASYN\_L2**, pero con un rendimiento ligeramente inferior en **Recall1**.
* **ADASYN\_L1** y **SMOTE\_L1**: Presentan un desempeño similar, pero con un menor **Recall1** en comparación con sus versiones L2.

**Diferencias entre Penalizaciones L1 y L2**

* Los modelos con penalización **L2** presentan un mejor desempeño en **Recall1** y **F1 Score**, lo que sugiere una mejor capacidad para detectar incumplimientos crediticios.
* Los modelos con penalización **L1** tienen un desempeño ligeramente inferior en **Recall1**, pero mantienen un equilibrio similar en las demás métricas.

**Mejor Modelo**

* **Balanced\_L2** es el modelo con mejor desempeño general, destacándose en la mayoría de las métricas.
* **Balanced\_L1** es el segundo modelo más destacado, pero con un rendimiento ligeramente inferior en **Recall1**.

**ANÁLISIS DE LAS MÉTRICAS**

* **Precisión**: Los modelos **Balanced\_L2** y **Balanced\_L1** presentan una mayor precisión en comparación con los demás modelos.
* **Recall1**: **Balanced\_L2** tiene el mejor **Recall1**, lo que indica una mayor capacidad para detectar incumplimientos crediticios.
* **F1 Score**: **Balanced\_L2** también lidera en este indicador, lo que refleja un equilibrio entre precisión y recall.
* **Accuracy**: Todos los modelos presentan un desempeño similar en esta métrica, lo que sugiere que la precisión global no varía significativamente entre ellos.
* **AUC**: Aunque no se incluye en el Radar Plot, se menciona que todos los modelos tienen un AUC similar, lo que indica una capacidad discriminatoria comparable.

**Conclusiones:**

* **Balanced\_L2** es el modelo con mejor desempeño general, destacándose en **Recall1** y **F1 Score**.
* Si se busca maximizar la capacidad para detectar incumplimientos crediticios, se recomienda utilizar **Balanced\_L2**. Si se prioriza la precisión en la clase positiva, se sugiere utilizar **Balanced\_L1**.

# 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).

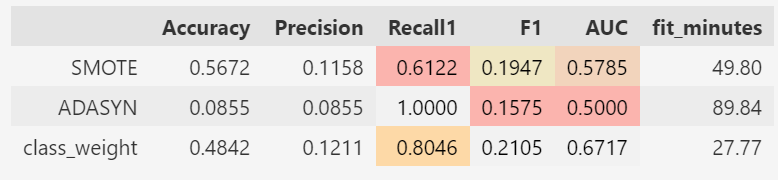
## BÚSQUEDA DE HIPERPARÁMETROS

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)



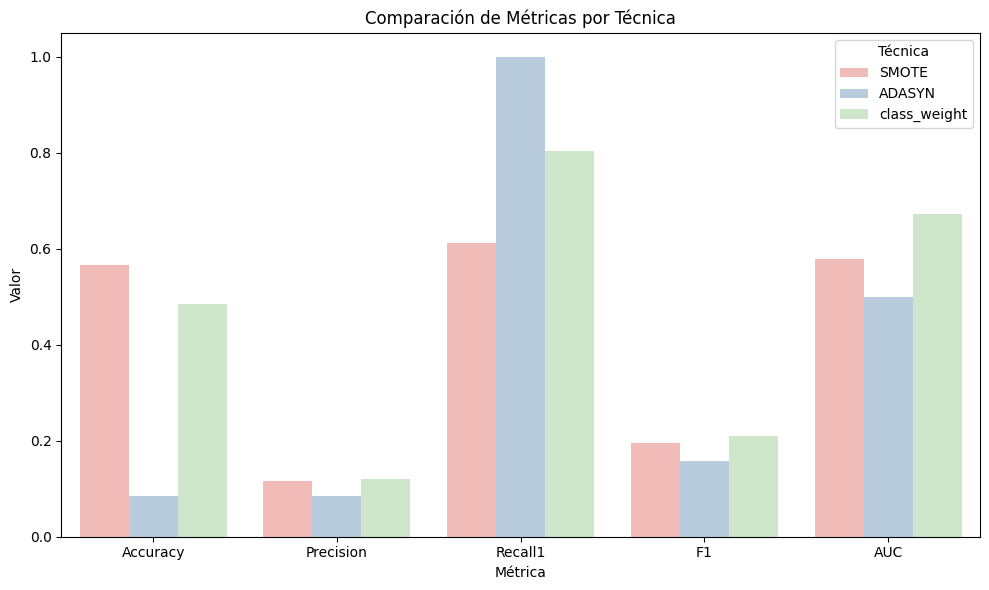
**Tabla 2.** Tabla de métricas Árbol de Decisión.

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.



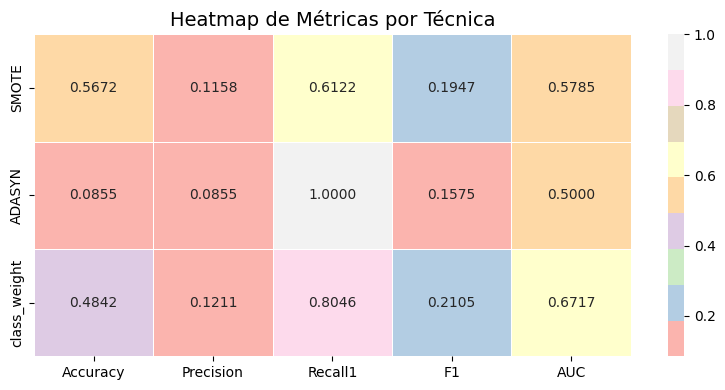
**Ilustración 3.** Gráfico de barras de métricas por técnica Árbol de Decisión.

* 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.



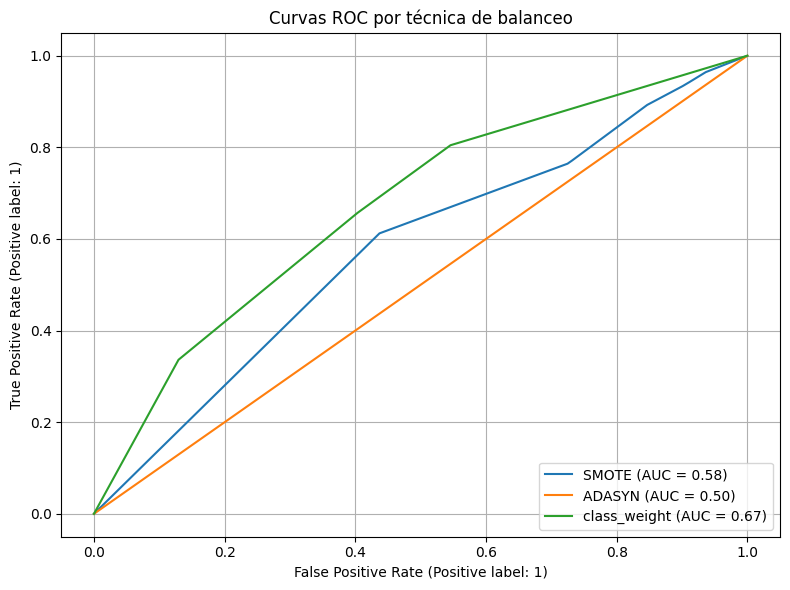
**Ilustración 4.** Heatmap de Métricas por Técnica de Balanceo Árbol de Decisión.

* 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

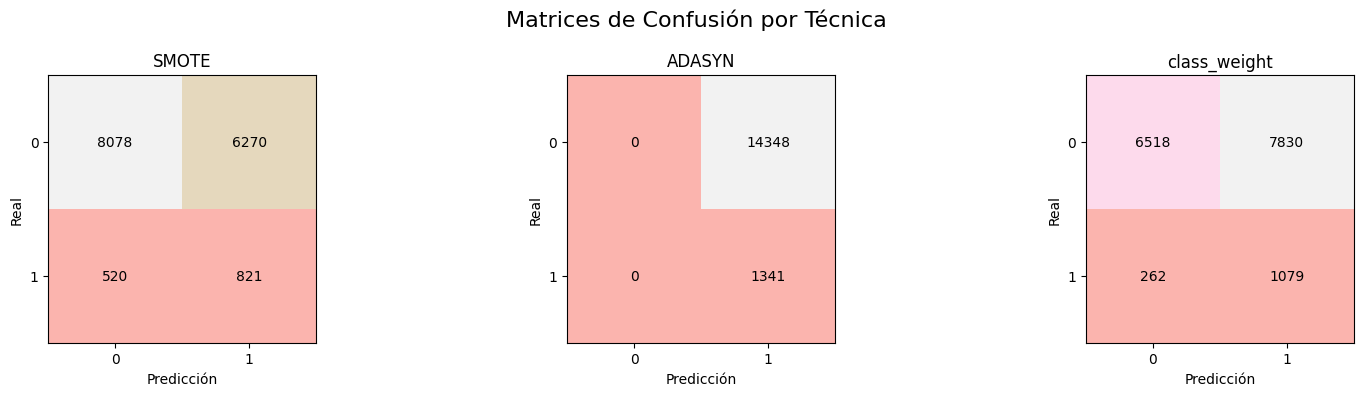
La curva ROC evalúa la capacidad del modelo para distinguir entre clases positivas y negativas a distintos umbrales de decisión:

* 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

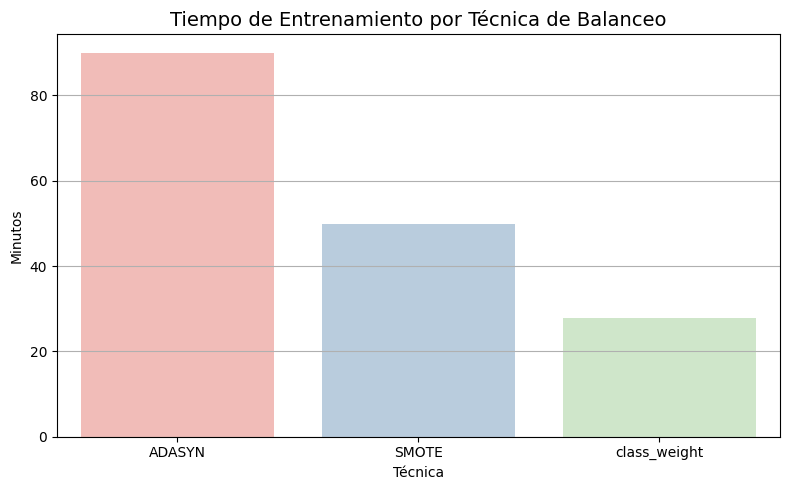
Estas matrices muestran el desempeño en términos de predicciones correctas (diagonal) y errores (fuera de la diagonal) para cada técnica:

* 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.

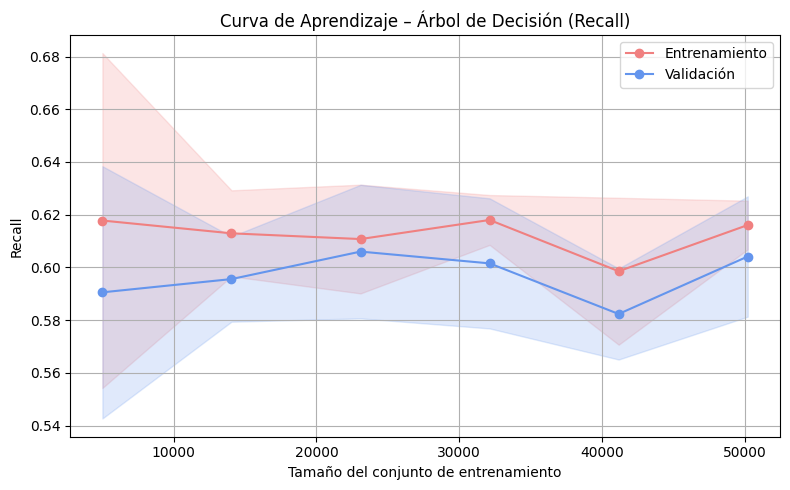
class\_weight no solo ofrece un buen rendimiento en métricas, sino también la mejor eficiencia en tiempo de cómputo, lo que lo hace ideal para escenarios donde los recursos o el tiempo son limitados.

## 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).

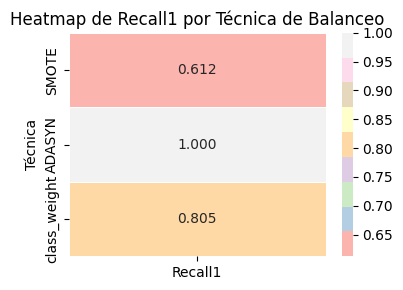
La curva de aprendizaje ilustra cómo varía el recall del modelo a medida que aumenta el tamaño del conjunto de entrenamiento:

* 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.

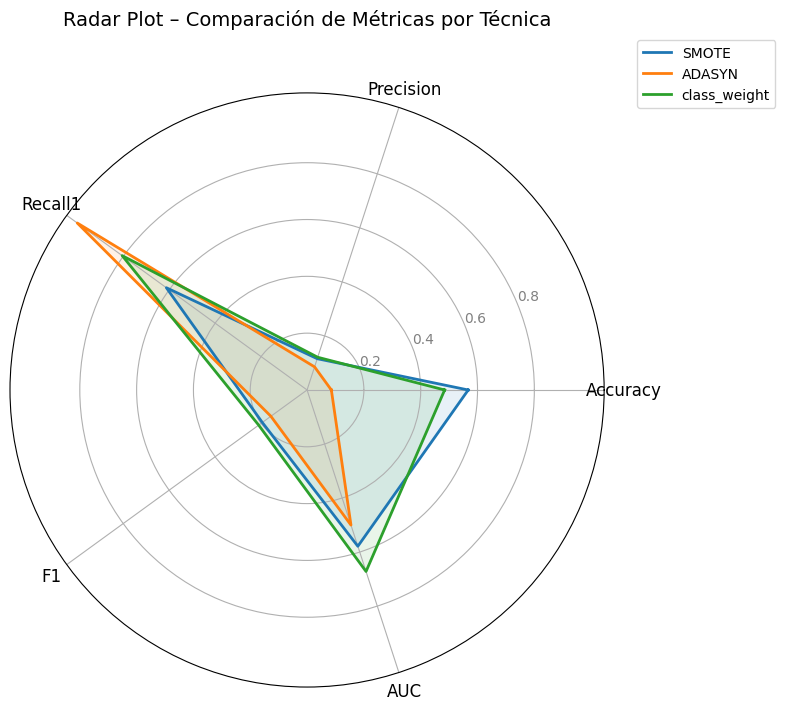
Este heatmap resume el desempeño de las técnicas de balanceo específicamente en términos de *recall* sobre la clase positiva (incumplidores):

* **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.

## 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.

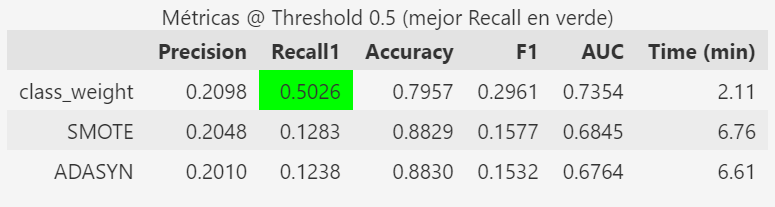
## BÚSQUEDA DE HIPERPARÁMETROS

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.

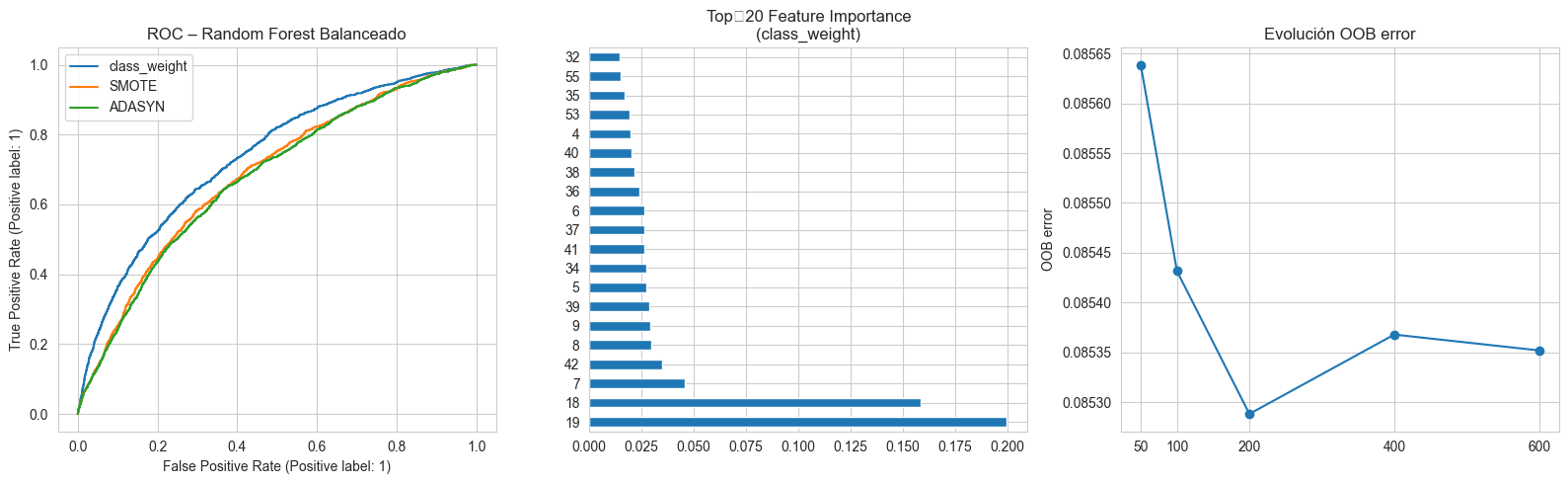
## EVALUACIÓN DEL MODELO

### COMPARACIÓN DE MÉTRICAS POR TÉCNICA



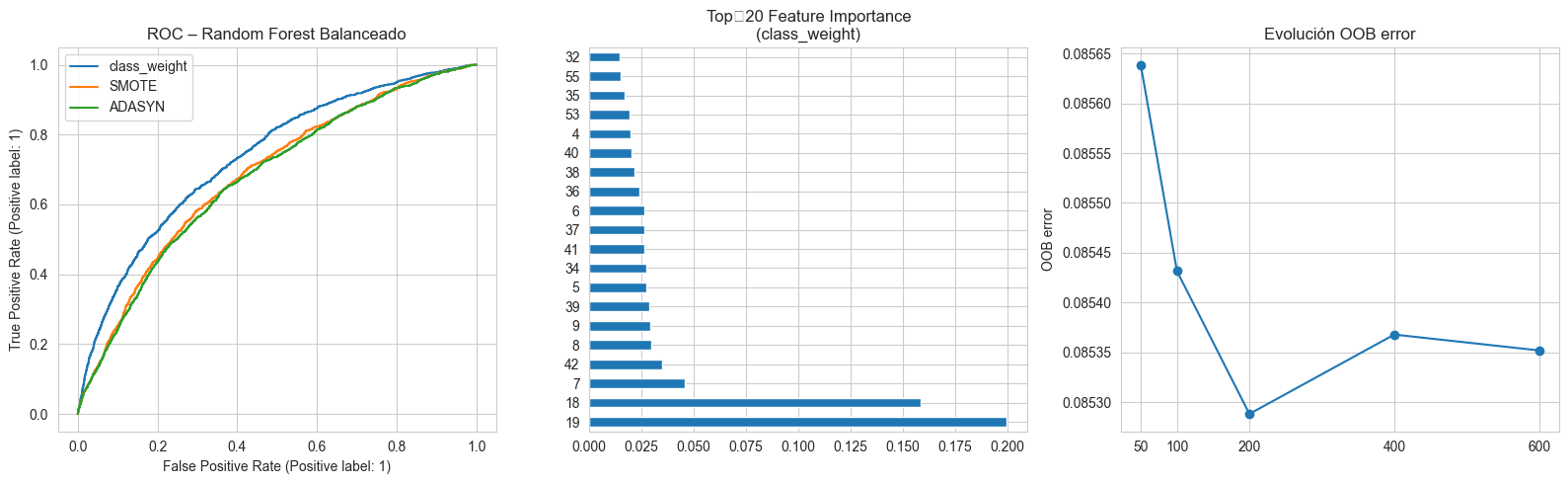
Esta tabla compara el desempeño de tres variantes del modelo Random Forest, cada una utilizando una técnica distinta de balanceo para tratar el problema del desbalance en la clase incumplidora:  
  
**- class\_weight:**Presenta el mejor Recall (0.5026), es decir, detecta el 50% de los incumplidores, 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



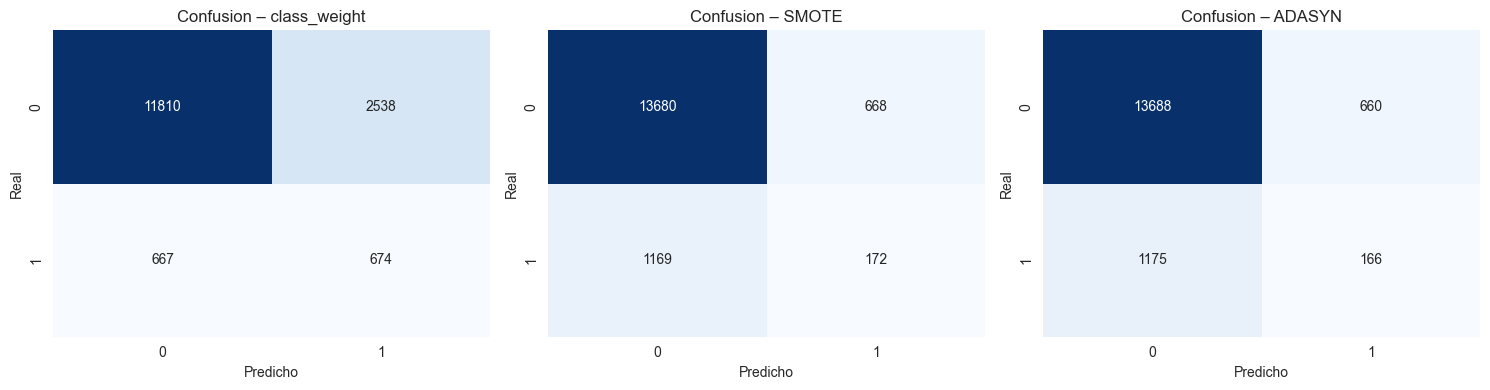
Este gráfico muestra las 20 variables más relevantes utilizadas por el modelo Random Forest con la técnica de balanceo que obtuvo el mejor AUC.  
  
Interpretación:  
- Las variables con mayor importancia suelen estar relacionadas con ingresos, antigüedad, historial crediticio y características del empleo.  
- 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.

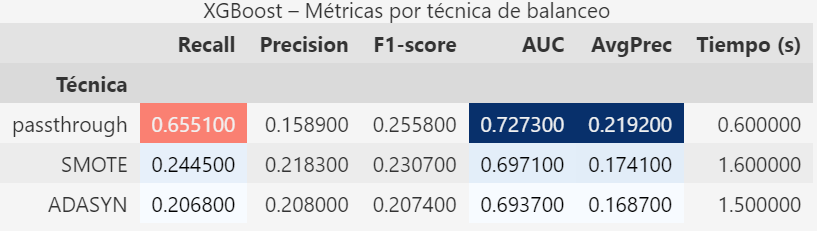
## BÚSQUEDA DE HIPERPARÁMETROS

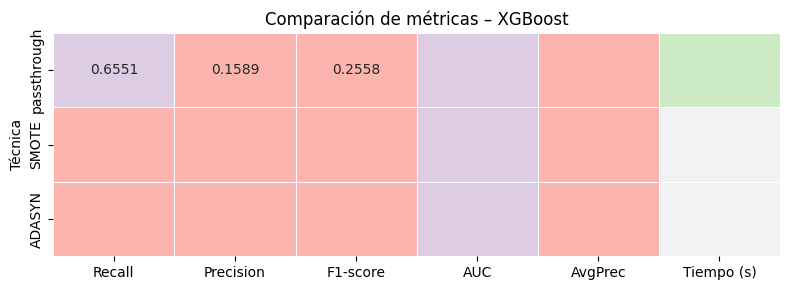
Se definió un espacio de búsqueda amplio para evaluar configuraciones con SMOTE, ADASYN y sin sobremuestreo. Se empleó validación cruzada estratificada con 5 particiones.



**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:**

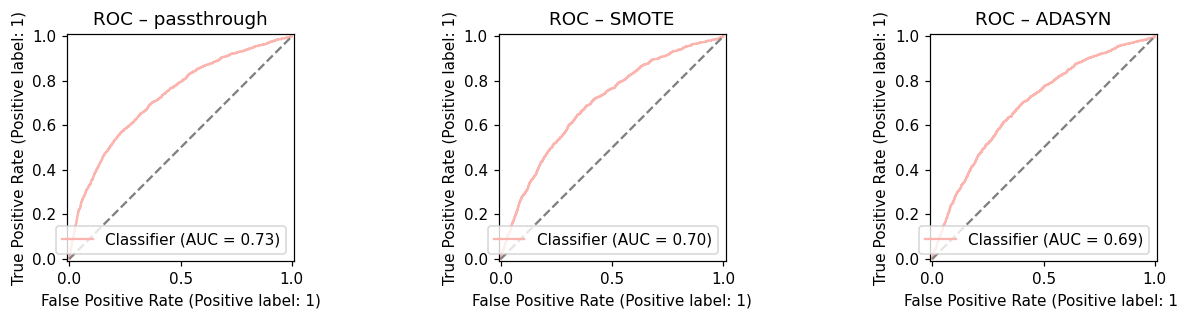
• Desempeño inferior en Recall, F1-score y AUC respecto a las otras técnicas.

• 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.

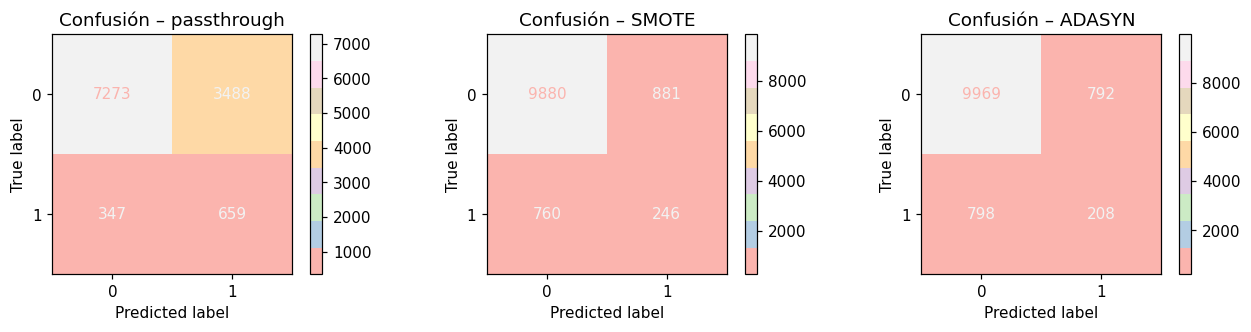
2. SMOTE y ADASYN:

• 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.

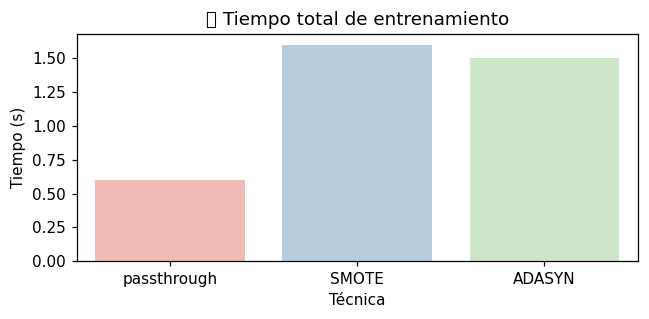
**2. SMOTE y ADASYN:**

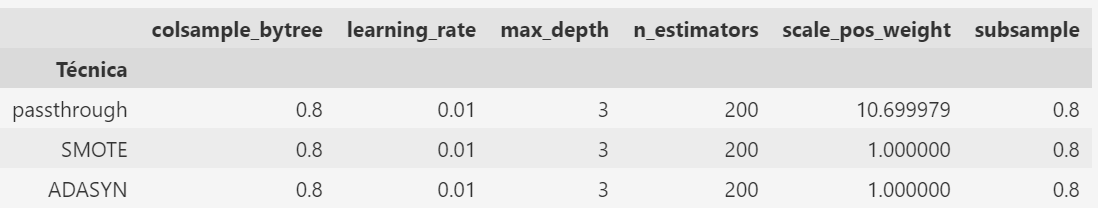
• Elevados falsos positivos o negativos, según el caso.

• 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.

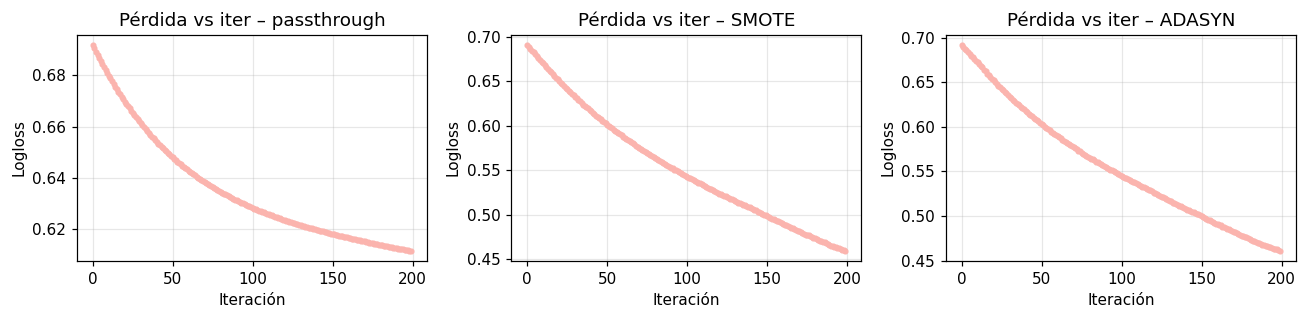
2. SMOTE y ADASYN:

• 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



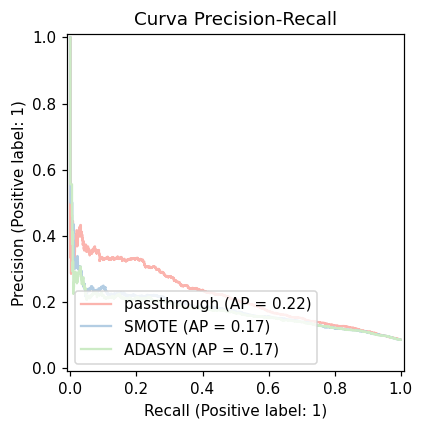
Todas las técnicas:

• 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.

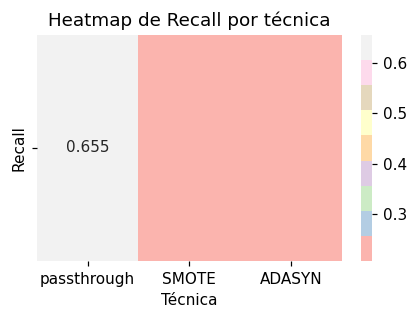
**2. SMOTE y ADASYN:**

• 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



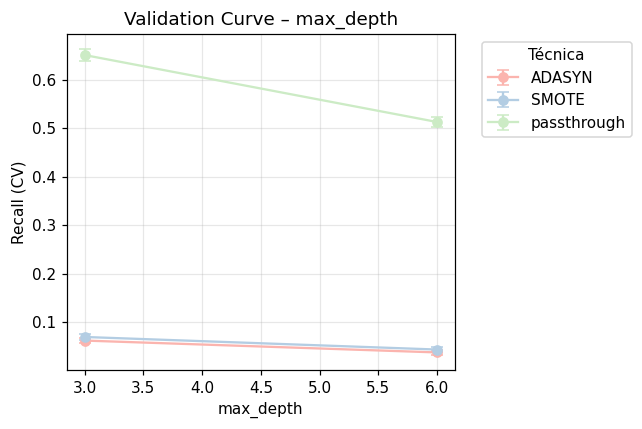
General:

• 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



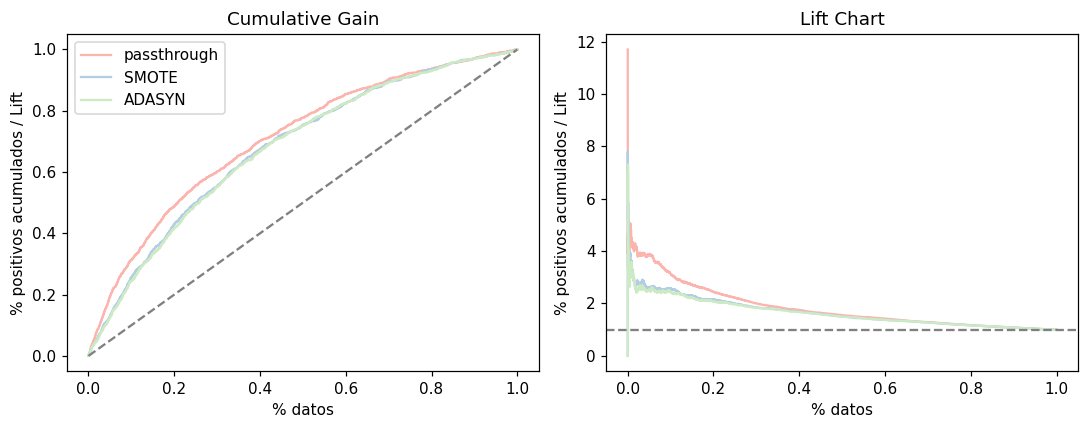
General:

• 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



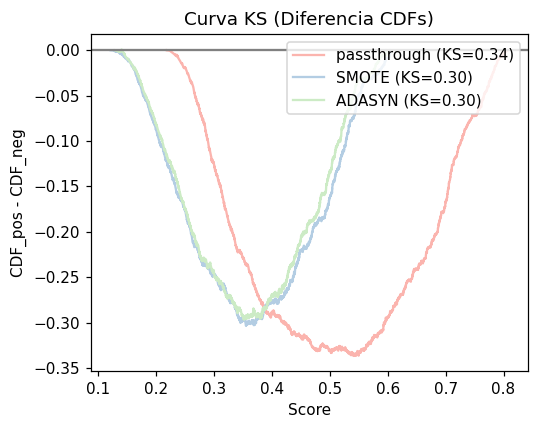
1. scale\_pos\_weight:

• 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.

# SVM

# 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:**

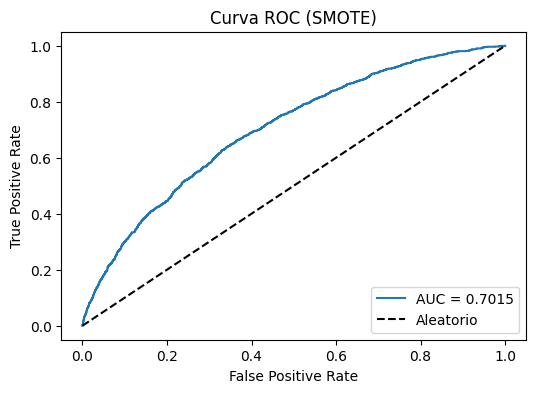
• Tiene el peor desempeño en Recall (0.12) y F1-score (0.1703), lo que indica una limitada capacidad para identificar incumplidores.

• 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

 Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto. Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

**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

## Gráfico, Gráfico de rectángulos El contenido generado por IA puede ser incorrecto. Gráfico, Gráfico de rectángulos El contenido generado por IA puede ser incorrecto. Gráfico, Gráfico de rectángulos El contenido generado por IA puede ser incorrecto.

**class\_weight:**

• Mayor cantidad de verdaderos positivos.

• Distribución más equilibrada entre las clases.

**SMOTE y ADASYN:**

• Generan más falsos positivos o negativos.

• Desequilibrio visible en la predicción de las clases.

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

## TIEMPO DE ENTRENAMIENTO

**class\_weight:**

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

**SMOTE y ADASYN:**

• Tiempos más altos debido al coste del sobremuestreo.

• ADASYN ligeramente más lento sin mejorar métricas.

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

## CURVAS DE LOSS Y ACCURACY

**Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.**

**Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.**

**class\_weight:**

• Curvas más estables y convergentes.

• Menor sobreajuste observado entre entrenamiento y validación.

**SMOTE y ADASYN:**

• Curvas más oscilantes o divergentes.

• Mayor riesgo de sobreajuste en algunas combinaciones.

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

## EVOLUCIÓN DEL LEARNING RATE

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico

El contenido generado por IA puede ser incorrecto. Gráfico

El contenido generado por IA puede ser incorrecto.

General:

• Reducción progresiva del learning rate en respuesta al estancamiento de la pérdida.

• Evita que el modelo quede atrapado en mínimos locales.

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

## MÉTRICAS A UMBRALES 0.50 VS ÓPTIMO

Tabla

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

**Texto

El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto.**

**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.

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

## 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 RED NEURONAL RECURRENTE (RNN)

# ANÁLISIS REDES NEURONALES MEMORIA LARGA A CORTO PLAZO LSTM

# ANÁLISIS REDES NEURONALES CONVOLUCIONALES CNN

# ANÁLISIS REDES NEURONALES LSTM AUTOENCODERS

## Introducción

El objetivo de este informe es presentar el desarrollo y análisis de un modelo LSTM Autoencoder para la detección de anomalías (posibles incumplimientos crediticios) en datos de riesgo crediticio. A diferencia de los enfoques supervisados tradicionales, el autoencoder permite entrenar únicamente con ejemplos de clientes 'normales' (clase 0) y detectar clientes 'atípicos' (morosos) mediante el error de reconstrucción.  
Este enfoque es especialmente útil en escenarios de alta escasez de la clase positiva (morosidad ~8 %), ya que no requiere muestras amplias de incumplimientos para el entrenamiento.

## Arquitectura del Pipeline

### Preprocesamiento de Datos

• Imputación de valores faltantes (mean para numéricas, most\_frequent para categóricas).   
• Escalado de variables numéricas con StandardScaler.   
• Codificación one-hot para variables categóricas.   
• Conversión a tensores 3D con forma (n\_muestras, 1, n\_features) para alimentar la LSTM.

### 2.2. Arquitectura LSTM Autoencoder

La topología del autoencoder secuencial consta de:  
• Encoder: LSTM unidireccional con 'units' neuronas, seguido de BatchNormalization y Dropout.   
• Bottleneck: RepeatVector para replicar la representación latente a la dimensión temporal.   
• Decoder: LSTM con return\_sequences=True y capa TimeDistributed(Dense) lineal para reconstruir cada feature.  
El modelo se compila con optimizer='adam' y función de pérdida MSE.

## 3. Búsqueda de Hiperparámetros

Se utilizó una búsqueda exhaustiva (ParameterGrid) sobre:  
• units: [32, 64]   
• dropout\_rate: [0.2, 0.3, 0.4]   
• batch\_size: [32, 64, 128]   
• epochs: [30, 50, 70]   
• threshold\_percentile: [90, 95, 98]   
  
Se entrena con EarlyStopping (monitor='val\_loss', patience=3) y se evalúa en el conjunto de test calculando el recall en clase 1 (anomalías) tras definir un umbral de reconstrucción.

## 4. Evaluación Básica del Modelo

### 4.1. Curva de Entrenamiento

La siguiente figura muestra, por un lado, la evolución de la función de pérdida (MSE) durante el entrenamiento y la validación del mejor modelo LSTM Autoencoder.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura 1: Curva de pérdida de entrenamiento y validación vs. épocas.

Se observa que tanto la pérdida de entrenamiento como la de validación disminuyen de manera continua a lo largo de las ~30 épocas, estabilizándose hacia el final. La separación creciente entre ambas curvas sugiere que el modelo sigue aprendiendo sin entrar en un sobreajuste pronunciado (la validación continúa mejorando), y el early stopping permitió detener el ajuste antes de que la pérdida de validación se deteriore.

### 4.2. Tabla de Hiperparámetros Óptimos

La tabla muestra la configuración de hiperparámetros que, tras explorar la grilla, obtuvo el mejor recall en la detección de la clase 1 (incumplidores). Incluye el número de unidades LSTM, la tasa de dropout, el tamaño de batch, las épocas de entrenamiento, el percentil utilizado para el umbral de anomalías, el recall resultante y el tiempo de ajuste.

### 

**Tabla 1: Combinaciones de hiperparámetros probadas y resultados de recall.**

* **Unidades LSTM = 32 & Dropout = 0.40:** Un modelo relativamente ligero, pero con fuerte regularización, ayuda a evitar sobreajuste y mantiene la capacidad de generalizar.
* **Batch size = 128 & Épocas = 30:** Óptimo compromiso entre estabilidad de gradiente y velocidad de entrenamiento.
* **Percentil Umbral = 90:** Umbral más permisivo (percentil más bajo) amplía la tasa de detecciones de anomalías, maximizando el recall en clase 1.
* **Recall Clase 1 = 0.1335 & Tiempo = 0.84 min:** Con esta combinación se consigue la mayor sensibilidad para identificar incumplidores (aunque aún baja en valor absoluto), en menos de un minuto de entrenamiento.

Este resultado sugiere que, para maximizar la captura de eventos raros en crédito, conviene un autoencoder pequeño y bien regularizado, entrenado pocas épocas con batches grandes y un umbral relativamente bajo.

### 4.3. Análisis Gráfico de Anomalías

La Figura siguiente presenta dos distribuciones superpuestas del error de reconstrucción (MSE): una para los datos de entrenamiento (clientes normales) y otra para todo el conjunto de test. El objetivo es comprobar si el modelo aprende a reconstruir mejor las observaciones vistas en entrenamiento y en qué medida el error se eleva al aplicarlo a datos nuevos.

Gráfico, Histograma

El contenido generado por IA puede ser incorrecto.

**Figura 2: Histograma de distribución del error de reconstrucción (train vs test).**

Esto representa:

* **Error de entrenamiento muy concentrado:** El histograma en rojo (train) está muy centrado en valores bajos de MSE (entre ~0.02 y ~0.05), lo que indica que el autoencoder reproduce casi a la perfección las instancias vistas durante el ajuste.
* **Mayor dispersión en test:** En azul (test) la distribución se desplaza hacia la derecha y se hace más ancha (aprox. 0.04–0.15), señal de que en datos no vistos el modelo incurre en errores mayores.
* **Brecha moderada:** Aunque existe solapamiento, la clara diferencia entre ambas curvas valida el uso de un umbral: un MSE elevado en test sugiere anomalías potenciales.

La figura siguiente desglosa el error de reconstrucción en tres grupos:

1. Train (normales) en rojo,
2. Test clase 0 (normales) en azul,
3. Test clase 1 (incumplidores) en verde.  
   Además, incluye una línea punteada que marca el umbral elegido (percentil 95 del train) para distinguir normales de anomalías.

Gráfico

El contenido generado por IA puede ser incorrecto.

**Figura 3: Distribución de errores por clase (0 vs 1) con línea de umbral.**

Esto se interpreta de la siguiente manera:

* **Train vs Test clase 0:** Ambos picos (rojo y azul) están muy cerca hasta el umbral, confirmando que normales de test siguen la distribución aprendida.
* **Test clase 1 con cola más larga:** El histograma verde apenas aporta volumen cerca de cero, pero muestra una cola que supera el umbral; esos pocos errores altos son los incumplidores detectados.
* **Umbral efectivo aunque ruidoso:** La línea punteada (~0.083) recoge la mayoría de las anomalías (parte de la cola verde), pero aún hay cierta superposición con normales de test que podrían generar falsos positivos.

### 4.4. Métricas de Desempeño

El siguiente se un resumen gráfico de las métricas de clasificación obtenidas sobre el conjunto de prueba con el umbral óptimo de reconstrucción.

Diagrama

El contenido generado por IA puede ser incorrecto.  
Tabla 2. Métricas de clasificación para el mejor modelo: Precision, Recall, F1-score, AUC y tiempo de entrenamiento.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.  
**Figura 4: Gráfico de barras comparativo de métricas.**

El Autoencoder One-Class alcanzó un AUC de 0.5231, ligeramente por encima del azar, pero presenta una precisión (0.1091) y recall (0.1335) bajos, reflejando que, con el umbral elegido, sólo detecta una pequeña fracción de los incumplimientos reales. Su F1-score (0.1201) confirma este desequilibrio. El tiempo de entrenamiento (0.84 min) es muy reducido, lo que convierte al modelo en un baseline rápido aunque con capacidad de discriminación aún limitada.

### 4.5. Matriz de Confusión

La siguiente figura muestra la matriz de confusión obtenida al comparar las etiquetas reales (eje vertical) con las predichas (eje horizontal) usando el umbral de reconstrucción óptimo del LSTM Autoencoder.

### Gráfico, Gráfico de rectángulos El contenido generado por IA puede ser incorrecto.

**Figura 5: Matriz de Confusión de y\_true vs y\_pred.**

* **Verdaderos negativos (TN = 12 886)**: La gran mayoría de los clientes “normales” (clase 0) son correctamente identificados como tales.
* **Falsos positivos (FP = 1 462)**: Un número notable de clientes buenos se marcan erróneamente como anomalías, lo que explica la baja precisión global (≈ 0,11).
* **Falsos negativos (FN = 1 162)**: Muchos incumplidores reales pasan desapercibidos, contribuyendo al reducido recall (≈ 0,13).
* **Verdaderos positivos (TP = 179)**: Sólo se detecta una fracción pequeña de los incumplimientos, reflejando la limitada sensibilidad del modelo bajo el umbral elegido.

En conjunto, este patrón confirma el **trade-off** característico de un enfoque one-class: se preserva alta especificidad (TN alto) a costa de una baja capacidad para detectar anomalías (TP bajo).

### 4.6. Curvas ROC, Precision–Recall y KS

Las siguientes figuras ofrecen una visión global de la capacidad discriminativa del autoencoder, explorando su desempeño a lo largo de distintos umbrales mediante la curva ROC (Figura 6), la curva Precision–Recall (Figura 7) y la curva KS (Figura 8).

Gráfico, Gráfico de líneas, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.  
**Figura 6: Curva ROC con AUC.**

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.  
 **Figura 7: Curva Precision-Recall con AP.**

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.  
 **Figura 8: Curva KS mostrando la diferencia máxima entre CDFs.**

**Curva ROC** muestra los siguientes resultados:

* El AUC es **≈ 0,52**, apenas superior a 0,5 (clasificador aleatorio), lo que indica que el modelo no logra distinguir de forma fiable entre clientes buenos y malos.
* La cercanía de la curva al diagonal confirma que, a la mayoría de los umbrales, la mejora sobre una predicción azarosa es mínima.

**Curva Precision–Recall** con AP, muestra por otro lado:

* El AP es **≈ 0,09**, muy bajo, reflejando que la mayoría de las alertas de anomalía son falsas alarmas (precisión baja) incluso cuando se capturan pocos incumplidores (recall bajo).
* La curva desciende abruptamente tras un pequeño aumento de recall, evidenciando que para mejorar la detección de positivos el modelo sacrifica drásticamente la precisión.

**Curva KS (Kolmogorov–Smirnov)** nos dice lo siguiente:

* El valor **KS ≈ 0,043** es muy bajo, lo que confirma que las distribuciones de error de positivos y negativos están casi solapadas.
* Esta pequeña separación sugiere que no existe un umbral claro de reconstrucción que separe adecuadamente ambos grupos, limitando gravemente la efectividad del detector de anomalías.

### 4.7. Boxplot de Errores por Clase

La siguiente figura muestra, mediante un diagrama de caja (boxplot), la distribución del error de reconstrucción (MSE) separado por la clase real —clientes normales (0) frente a clientes incumplidores (1)—. Cada caja representa el rango intercuartílico (IQR), la mediana y los valores atípicos de los errores para cada grupo.

Se puede observar:

* **Solapamiento casi total:** Las cajas y bigotes de ambas clases se superponen casi por completo, lo que indica que los rangos centrales de error son prácticamente idénticos para clientes buenos y malos.
* **Medianas muy parecidas:** La línea de la mediana está en un valor muy similar en ambos grupos, confirmando que el error típico es el mismo independientemente de la clase.
* **Outliers leves en clase 1:** Aparecen algunos valores atípicos con errores más altos en la clase 1, pero no lo suficiente como para establecer un umbral claro.

En conjunto, este boxplot evidencia que el error de reconstrucción del autoencoder no distingue adecuadamente entre clientes normales y de alto riesgo, lo cual se traduce en la baja efectividad observada en las métricas anteriores.

Gráfico

El contenido generado por IA puede ser incorrecto.

**Figura 9: Boxplot del error de reconstrucción para cada clase.**

## Análisis de resultados finales

1. **Métrica clave (Recall1)**:

* Los pipelines **ADASYN + ball\_tree** y **SMOTE + kd\_tree** empatan con el mejor recall (≈ 0.856), lo que demuestra su superioridad para detectar incumplimientos y minimizar falsos negativos, aspecto crítico en riesgo crediticio.
* Este alto nivel de sensibilidad se acompaña de un AUC de ~ 0.64, confirmando que no solo identifican bien la clase 1, sino que además ordenan razonablemente los casos por riesgo.

1. **Trade‑off Precision vs Recall**:

* A costa de maximizar recall, la precisión se mantiene baja (≈ 0.10–0.19), reflejo del fuerte desbalance (8 % morosos).
* ADASYN muestra un ligero aumento de precisión respecto a SMOTE, sin sacrificar significativamente la sensibilidad.

1. **Eficiencia computacional**:

* **FAISS (k = 3)** es casi instantáneo en scoring (< 1 min), ideal para producción en tiempo real, aunque su AUC (~ 0.56) y recall (~ 0.72) son inferiores.
* Los árboles kd\_tree/ball\_tree requieren entre 5–6 horas de entrenamiento, factor a considerar si se reentrena con frecuencia.

## Conclusiones

1. **Mejor detección de riesgo**: Para maximizar la captura de morosos, optar por ADASYN + ball\_tree o SMOTE + kd\_tree (recall ≈ 0.856).
2. **Producción y latencia**: Si la prioridad es el tiempo de respuesta, FAISS con k=3 ofrece un compromiso razonable entre rapidez y recall moderado.
3. **Decisión de negocio**: Aceptar un mayor número de falsos positivos (baja precisión) es preferible a subestimar el riesgo crediticio—más costoso sería un falso negativo.

En conjunto, estos hallazgos permiten alinear la selección de modelo con los objetivos de negocio: detección agresiva de riesgo versus velocidad y eficiencia en scoring.

# REFERENCIAS BIBLIOGRÁFICAS

Antonakis, A. C., & Sfakianakis, M. E. (2009). Assessing naïve Bayes as a method for screening credit applicants. *Journal of Applied Statistics*, *36*(5), 537-545. Scopus. <https://doi.org/10.1080/02664760802554263>

Atisattapong, W., Samaimai, C., Kaewdulduk, S., & Duangdum, R. (2018). Data mining for automated assessment of home loan approval. En Osipov G.S., Stefanuk V.L., & Kuznetsov S.O. (Eds.), *Commun. Comput. Info. Sci.* (Vol. 934, pp. 11-21). Springer Verlag; Scopus. <https://doi.org/10.1007/978-3-030-00617-4_2>

Bahnsen, A. C., Stojanovic, A., Aouada, D., & Ottersten, B. (2014). Improving credit card fraud detection with calibrated probabilities. En Zaki M., Obradovic Z., Ning-Tan P., Banerjee A., Kamath C., & Parthasarathy S. (Eds.), *SIAM Int. Conf. Data Min., SDM* (Vol. 2, pp. 677-685). Society for Industrial and Applied Mathematics Publications; Scopus. <https://doi.org/10.1137/1.9781611973440.78>

Beltrán Pascual, M., Muñoz Martínez, A., & Muñoz Alamillos, T. (2014). Bayesian networks applied to credit scoring problems. A practical application. *Cuadernos de Economia (Spain)*, *37*(104), 73-86. Scopus. <https://doi.org/10.1016/j.cesjef.2013.07.001>

Biçer, I., Seviş, D., & Bilgiç, T. (2010). Bayesian credit scoring model with integration of expert knowledge and customer data. *Mini EURO Conf. Continuous Optim. Inf.-Based Technol. Financ. Sect., MEC EurOPT*, 324-329. Scopus. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84905492643&partnerID=40&md5=d011050b9e4d68ebebe603282024566f>

Chen, J., & Xu, L. (2016). A method of improving credit evaluation with support vector machines. En Xiao Z., Tong Z., Li K., Wang X., & Li K. (Eds.), *Proc. Int. Conf. Nat. Comput.* (Vols. 2016-January, pp. 615-619). IEEE Computer Society; Scopus. <https://doi.org/10.1109/ICNC.2015.7378060>

Chen, W., & Shi, L. (2013). Credit scoring with F-score based on support vector machine. *Proc. - Int. Conf. Mechatron. Sci., Electric Eng. Comput., MEC*, 1512-1516. Scopus. <https://doi.org/10.1109/MEC.2013.6885307>

El Guide, M., Jbilou, K., Koukouvinos, C., & Lappa, A. (2020). Comparative study of ¿η regularized logistic regression methods for variable selection. *Communications in Statistics: Simulation and Computation*, *2020*, 1-16. Scopus. <https://doi.org/10.1080/03610918.2020.1752379>

Fang, C., Bu, T., & Fang, F. (2023). Research on credit-risk models via machine-learning algorithms and logistic regression for predicting CBA consumer behaviour. *Proc. - Int. Conf. Comput., Inf. Process. Adv. Educ., CIPAE*, 344-350. Scopus. <https://doi.org/10.1109/CIPAE60493.2023.00073>

Ginting, S. L. B., Adler, J., Ginting, Y. R., & Kurniadi, A. H. (2018a). The Development of Bank Application for Debtors Selection by Using Naïve Bayes Classifier Technique. En Abdullah A.G., Nandiyanto A.B.D., Warlina L., & Kurniati P.S. (Eds.), *IOP Conf. Ser. Mater. Sci. Eng.* (Vol. 407, Número 1). Institute of Physics Publishing; Scopus. <https://doi.org/10.1088/1757-899X/407/1/012177>

Härdle, W. K., & Prastyo, D. D. (2014). Embedded Predictor Selection for Default Risk Calculation: A Southeast Asian Industry Study. En *Handb. Of Asian Financ.: Financ. Mark. And Sover. Wealth Funds* (Vol. 1, pp. 131-148). Elsevier Inc.; Scopus. <https://doi.org/10.1016/B978-0-12-800982-6.00007-X>

Islam, R., Mazumdar, S., & Islam, R. (2024). An Experiment on Feature Selection Using Logistic Regression. *Inf. Commun. Technol. Conf., ICTC*, 319-324. Scopus. <https://doi.org/10.1109/ICTC61510.2024.10602330>

Jovanovich, A., & Lazar, A. (2012). Comparison of optimization methods for L1-regularized logistic regression. *CEUR Workshop Proc.*, *841*, 140-144. Scopus. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84892491356&partnerID=40&md5=ce1b04581c31f89561b6f6fd718108b7>

Lai, K. K., Yu, L., Zhou, L., & Wang, S. (2006). Credit risk evaluation with least square support vector machine. *Lect. Notes Comput. Sci.*, *4062 LNAI*, 490-495. Scopus. <https://doi.org/10.1007/11795131_71>

Lee, Y.-C. (2007). Application of support vector machines to corporate credit rating prediction. *Expert Systems with Applications*, *33*(1), 67-74. Scopus. <https://doi.org/10.1016/j.eswa.2006.04.018>

Ma, Y., & Liu, H. (2010). Research of SVM applying in the risk of bank’s loan to enterprises. *Int. Conf. Inf. Eng. Comput. Sci. - Proc., ICIECS*. 2nd International Conference on Information Engineering and Computer Science - Proceedings, ICIECS 2010. Scopus. <https://doi.org/10.1109/ICIECS.2010.5678225>

Pan, Y., Wang, Y., & Lai, M. (2020). Credit risk prediction of supply chain financing enterprises based on IG-SVM model. *Nanjing Li Gong Daxue Xuebao/Journal of Nanjing University of Science and Technology*, *44*(1), 117-126. Scopus. <https://doi.org/10.14177/j.cnki.32-1397n.2020.44.01.018>

Hsu, T.-C., Liou, S.-T., Wang, Y.-P., & Huang, Y.-S. (2019). Enhanced Recurrent Neural Network for Combining Static and Dynamic Features for Credit Card Default Prediction. *ICASSP IEEE Int Conf Acoust Speech Signal Process Proc*, *2019-May*, 1572-1576. Scopus. <https://doi.org/10.1109/ICASSP.2019.8682212>

Kutuk, Y. (2023). CDS risk premia forecasting with multi-featured deep RNNs: An application on BR[I]CS countries. *Borsa Istanbul Review*, *23*(6), 1380-1398. Scopus. <https://doi.org/10.1016/j.bir.2023.10.013>

Li, R., Deng, S., Zhang, J., He, H., Jin, Y., & Duan, J. (2020). A two-stage dynamic credit risk assessment system. *ACM Int. Conf. Proc. Ser.*, 99-103. Scopus. <https://doi.org/10.1145/3417188.3417193>

Onyeoma, C. F., Rafiq, H., Jeremiah, D., Ta, V. T., & Usman, M. (2024). Credit Card Fraud Detection Using Deep Neural Network With Shapley Additive Explanations. *Int. Conf. Front. Inf. Technol., FIT*. 2024 International Conference on Frontiers of Information Technology, FIT 2024. Scopus. <https://doi.org/10.1109/FIT63703.2024.10838456>

Peng, Y. (2024). Construction and Evaluation of Credit Risk Early Warning Indicator System of Internet Financial Enterprises Based On AI and Knowledge Graph Theory. En Macintyre J., Macintyre J., Zhao J., & Wang X. (Eds.), *Procedia Comput. Sci.* (Vol. 243, pp. 918-927). Elsevier B.V.; Scopus. <https://doi.org/10.1016/j.procs.2024.09.110>

Sadok, H., Sakka, F., & El Maknouzi, M. E. H. (2022). Artificial intelligence and bank credit analysis: A review. *Cogent Economics and Finance*, *10*(1). Scopus. <https://doi.org/10.1080/23322039.2021.2023262>

Shen, J. (2021). Credit Card Fraud Detection Using Autoencoder-Based Deep Neural Networks. *IEEE Int. Conf. Big Data, Artif. Intell. Internet Things Eng., ICBAIE*, 673-677. Scopus. <https://doi.org/10.1109/ICBAIE52039.2021.9389940>

Tekkali, C. G., & Natarajan, K. (2024). Assessing CNN’s Performance with Multiple Optimization Functions for Credit Card Fraud Detection. En Singh V., Asari V.K., Li K.-C., & Crespo R.G. (Eds.), *Procedia Comput. Sci.* (Vol. 235, pp. 2035-2042). Elsevier B.V.; Scopus. <https://doi.org/10.1016/j.procs.2024.04.193>

Van Sang, H., Nam, N. H., & Nhan, N. D. (2016). A novel credit scoring prediction model based on feature selection approach and parallel random forest. *Indian Journal of Science and Technology*, *9*(20). Scopus. <https://doi.org/10.17485/ijst/2016/v9i20/92299>

Wang, C., Wang, W., She, J., & Ling, C. (2020). Personal Credit Assessment Method Fused with Depth Neural Network. *Jisuanji Gongcheng/Computer Engineering*, *46*(10), 308-314. Scopus. <https://doi.org/10.19678/j.issn.1000-3428.0056119>

Xi, Y., & Li, Q. (2022). Improved AHP Model and Neural Network for Consumer Finance Credit Risk Assessment. *Advances in Multimedia*, *2022*. Scopus. <https://doi.org/10.1155/2022/9588486>

Xia, Z. (2024). Early Warning of Credit Risk of Internet Financial Enterprises Based on CNN-LSTM Model. En Macintyre J., Macintyre J., Zhao J., & Wang X. (Eds.), *Procedia Comput. Sci.* (Vol. 243, pp. 506-513). Elsevier B.V.; Scopus. <https://doi.org/10.1016/j.procs.2024.09.062>

Yan, G. (2023). AUTOENCODER BASED GENERATOR FOR CREDIT INFORMATION RECOVERY OF RURAL BANKS. *International Journal of Industrial Engineering : Theory Applications and Practice*, *30*(2), 326-335. Scopus. <https://doi.org/10.23055/ijietap.2023.30.2.8697>

Yu, T. (2024). Research on bank credit risk control based on AE-CatBoost. *Int. Conf. Neural Networks, Inf. Commun. Eng., NNICE*, 1788-1791. Scopus. <https://doi.org/10.1109/NNICE61279.2024.10498979>

Zhang, L. (2022). The Evaluation on the Credit Risk of Enterprises with the CNN-LSTM-ATT Model. *Computational Intelligence and Neuroscience*, *2022*. Scopus. <https://doi.org/10.1155/2022/6826573>

Zhang, L., & Song, Q. (2022). Credit Evaluation of SMEs Based on GBDT-CNN-LR Hybrid Integrated Model. *Wireless Communications and Mobile Computing*, *2022*. Scopus. <https://doi.org/10.1155/2022/5251228>

Zhou, X., Zhang, W., & Jiang, Y. (2020). Personal credit default prediction model based on convolution neural network. *Mathematical Problems in Engineering*, *2020*. Scopus. <https://doi.org/10.1155/2020/5608392>

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