**Comparación de algoritmos de DL frente a algoritmos de ML clásicos**

Caso de Estudio: Predicción del incumplimiento de pago de préstamos bancarios garantizados por la U.S. Small Business Administration

|  |  |  |
| --- | --- | --- |
| Titulación:  Máster Universitario en Inteligencia Artificial  Curso académico  2021 - 2023 | Alumno: Juan Alex Castro Gumiel  D.N.I: 5074733  Director de TFM: José Antonio Lagares Rodríguez | Convocatoria:  Cuarto Periodo |

**Índice**

[1. Introducción 5](#_Toc148476256)

[2. Marco Teórico 6](#_Toc148476257)

[2.1 Conceptos Clave 6](#_Toc148476258)

[2.2 Investigaciones Previas 10](#_Toc148476259)

[2.3 Hipótesis de Investigación 10](#_Toc148476260)

[3. Ámbito de la Aplicación 11](#_Toc148476261)

[4. Estado del Arte 12](#_Toc148476262)

[5. Justificación de la Propuesta 16](#_Toc148476263)

[5.1 Objetivo General 16](#_Toc148476264)

[5.2 Objetivos Específicos 17](#_Toc148476265)

[6. Descripción de la Propuesta 18](#_Toc148476266)

[7. Desarrollo de la Propuesta 19](#_Toc148476267)

[7.1 Análisis Exploratorio de Datos 19](#_Toc148476268)

[7.1.1 Comprensión de los Datos 19](#_Toc148476269)

[7.1.2 Variables Categóricas Nominales 22](#_Toc148476270)

[7.1.3 Variables Categóricas Numerales 24](#_Toc148476271)

[7.1.4 Variables Categóricas Temporales 26](#_Toc148476272)

[7.1.5 Variables Numéricas Discretas 28](#_Toc148476273)

[7.1.6 Variables Numéricas Continuas 29](#_Toc148476274)

[7.2 Preprocesamiento de Datos 30](#_Toc148476275)

[7.2.1 Ingeniería de Características 31](#_Toc148476276)

[7.2.2 Imputación de Valores Faltantes 32](#_Toc148476277)

[7.2.3 Detección de Valores Atípicos 34](#_Toc148476278)

[7.2.4 Escalar Variables Numéricas 35](#_Toc148476279)

[7.2.5 Codificar Variables Categóricas 36](#_Toc148476280)

[7.2.6 Selección de Características 37](#_Toc148476281)

[7.2.7 Resultados del Preprocesamiento 38](#_Toc148476282)

[7.2.8 Partición de Datos 42](#_Toc148476283)

[7.3 Modelos de ML para Clasificación 43](#_Toc148476284)

[7.3.1 Métricas de Evaluación 44](#_Toc148476285)

[7.3.2 Logistic Regression 44](#_Toc148476286)

[7.3.3 K-Nearest Neighbors 49](#_Toc148476287)

[7.3.4 Decision Tree Classifier 53](#_Toc148476288)

[7.3.5 Random Forest Classifier 57](#_Toc148476289)

[7.3.6 XGBoost 61](#_Toc148476290)

[7.4 Modelos de Deep Learning 65](#_Toc148476291)

[8. Resultados 66](#_Toc148476292)

[9. Conclusiones 67](#_Toc148476293)

**Índice de tablas**

[**Tabla 1** Performance metric for XGBoost algorithm on the loan dataset. 12](#_Toc148476294)

[**Tabla 2** Evaluation metrics comparison of the four techniques. 13](#_Toc148476295)

[**Tabla 3** Loan default prediction model results. 14](#_Toc148476296)

[**Tabla 4** The result of using three important features. 15](#_Toc148476297)

[**Tabla 5** Comparación de los mejores resultados obtenidos en los papers. 15](#_Toc148476298)

[**Tabla 6** Diccionario de Datos. 19](#_Toc148476299)

[**Tabla 7** Clasificación NAICS basada en sus dos primeros dígitos. 20](#_Toc148476300)

[**Tabla 8** Clasificación de las variables del Conjunto de Datos. 21](#_Toc148476301)

[**Tabla 9** Análisis de Variables Categóricas Nominales. 22](#_Toc148476302)

[**Tabla 10** Análisis de Variables Categóricas Numerales. 24](#_Toc148476303)

[**Tabla 11** Análisis del Target Leakage de la variable ChgOffDate. 27](#_Toc148476304)

[**Tabla 12** Análisis de Variables Numéricas Discretas. 28](#_Toc148476305)

[**Tabla 13** Análisis de Variables Numéricas Continuas. 29](#_Toc148476306)

[**Tabla 14** Análisis del Target Leakage de la variable ChgOffPrinGr. 30](#_Toc148476307)

[**Tabla 15** Características que fueron eliminadas. 31](#_Toc148476308)

[**Tabla 16** Resumen de la transformación de características. 32](#_Toc148476309)

[**Tabla 17** Detalle de variables con valores faltantes. 33](#_Toc148476310)

[**Tabla 18** Resultados de aplicar Desviación Estándar y Z-score. 34](#_Toc148476311)

[**Tabla 19** Resultados de aplicar Rango Intercuartílico. 35](#_Toc148476312)

[**Tabla 20** Umbral de Varianza y Análisis de Información Mutua. 38](#_Toc148476313)

[**Tabla 21** Tarjeta de Datos Preprocesados. 39](#_Toc148476314)

[**Tabla 22** Clasificación de las variables del conjunto de datos preprocesado. 39](#_Toc148476315)

[**Tabla 23** Hiperparametros óptimos de la Regresión Logística. 46](#_Toc148476316)

[**Tabla 24** Métricas de Evaluación de la Regresión Logística. 47](#_Toc148476317)

[**Tabla 25** Hiperparametros óptimos de K-NN. 50](#_Toc148476318)

[**Tabla 26** Métricas de Evaluación de K-NN. 51](#_Toc148476319)

[**Tabla 27** Hiperparametros óptimos del Árbol de Decision. 54](#_Toc148476320)

[**Tabla 28** Métricas de Evaluación del Árbol de Decision. 55](#_Toc148476321)

[**Tabla 29** Hiperparametros óptimos de Random Forest. 58](#_Toc148476322)

[**Tabla 30** Métricas de Evaluación de Random Forest. 59](#_Toc148476323)

[**Tabla 31** Hiperparametros óptimos de XGBoost. 62](#_Toc148476324)

[**Tabla 32** Métricas de Evaluación de XGBoost. 63](#_Toc148476325)

**Índice de figuras**

[**Figura 1** Distribución de la Variable Objetivo MIS\_Status 23](#_Toc148476326)

[**Figura 2** Distribución de la Variable UrbanRural 24](#_Toc148476327)

[**Figura 3** Distribución de Sectores de la Variable NAICS 25](#_Toc148476328)

[**Figura 4** Comparación de Variables Categóricas Temporales 26](#_Toc148476329)

[**Figura 5** Distribución de la Variable Term 28](#_Toc148476330)

[**Figura 6** Distribución de la Variable GrAppv 29](#_Toc148476331)

[**Figura 7** Distribución de las Variables Categóricas Nominales. 40](#_Toc148476332)

[**Figura 8** Distribución de las Variables Categóricas Ordinales. 40](#_Toc148476333)

[**Figura 9** Distribución de las Variables Categóricas Binarias. 41](#_Toc148476334)

[**Figura 10** Distribución de las Variables Numéricas Discretas. 41](#_Toc148476335)

[**Figura 11** Distribución de las Variables Numéricas Continuas. 42](#_Toc148476336)

[**Figura 12** Importancia de Características de Regresión Logística. 46](#_Toc148476337)

[**Figura 13** Validación Cruzada de la Regresión Logística. 47](#_Toc148476338)

[**Figura 14** Matriz de Confusion de la Regresión Logística. 48](#_Toc148476339)

[**Figura 15** Curva ROC y AUC de la Regresión Logística. 48](#_Toc148476340)

[**Figura 16** Matriz de Correlación de las variables. 50](#_Toc148476341)

[**Figura 17** Validación Cruzada de K-NN. 51](#_Toc148476342)

[**Figura 18** Matriz de Confusion de K-NN. 52](#_Toc148476343)

[**Figura 19** Curva ROC y AUC de K-NN. 52](#_Toc148476344)

[**Figura 20** Importancia de Características del Árbol de Decision. 54](#_Toc148476345)

[**Figura 21** Validación Cruzada del Árbol de Decision. 55](#_Toc148476346)

[**Figura 22** Matriz de Confusion del Árbol de Decision. 56](#_Toc148476347)

[**Figura 23** Curva ROC y AUC del Árbol de Decision. 56](#_Toc148476348)

[**Figura 24** Importancia de Características de Random Forest. 58](#_Toc148476349)

[**Figura 25** Validación Cruzada de Random Forest. 59](#_Toc148476350)

[**Figura 26** Matriz de Confusion de Random Forest. 60](#_Toc148476351)

[**Figura 27** Curva ROC y AUC de Random Forest. 60](#_Toc148476352)

[**Figura 28** Importancia de Características de XGBoost. 62](#_Toc148476353)

[**Figura 29** Validación Cruzada de XGBoost. 63](#_Toc148476354)

[**Figura 30** Matriz de Confusion de XGBoost. 64](#_Toc148476355)

[**Figura 31** Curva ROC y AUC de XGBoost. 64](#_Toc148476356)

1. Introducción
2. Marco Teórico

Este Marco Teórico proporciona un sólido fundamento conceptual y teórico para el estudio comparativo de algoritmos de Machine Learning (ML) con algoritmos de Deep Learning (DL) en el contexto de la predicción del incumplimiento de pago de préstamos bancarios garantizados por la U.S. SBA, se establece la base necesaria para comprender la naturaleza y la relevancia de los enfoques de IA en la toma de decisiones financieras y crediticias.

2.1 Conceptos Clave

Préstamo Bancario

Un préstamo bancario es un acuerdo financiero en el que una institución financiera presta una cantidad de dinero a un individuo, empresa u organización, conocido como prestatario, con la obligación de devolver esa cantidad junto con intereses en un plazo acordado.

Aprobación de Préstamos

La aprobación de préstamos se refiere al proceso mediante el cual una institución financiera realiza la evaluación de una solicitud de préstamo presentada por individuos o entidades, con el objetivo de determinar si son elegibles para recibir financiamiento, el proceso implica la consideración de diversos factores financieros para tomar decisiones informadas.

Incumplimiento de Pagos

El incumplimiento de pagos de un préstamo, también conocido como "default" o "impago", se refiere a la situación en la cual un prestatario no cumple con las obligaciones acordadas en un contrato de préstamo, estas obligaciones suelen incluir el pago regular de cuotas o intereses en las fechas especificadas en el contrato; el incumplimiento puede manifestarse en forma de retrasos en los pagos, pagos incompletos o la falta total de pago.

Garantizar Préstamos

El proceso de garantizar préstamos bancarios se refiere a proporcionar una seguridad o respaldo financiero adicional para un préstamo otorgado por una institución financiera a un individuo o entidad, esta garantía se utiliza para mitigar el riesgo del prestamista y asegurarse de que, en caso de que el prestatario no cumpla con sus obligaciones de pago, el prestamista tenga un recurso adicional para recuperar el dinero prestado.

Inteligencia Artificial (IA)

La Inteligencia Artificial (IA) se refiere a la simulación de procesos de inteligencia humana mediante el uso de sistemas informáticos, matemáticas y algoritmos; es una rama de la informática que se centra en el desarrollo de programas y sistemas capaces de realizar tareas que normalmente requieren de inteligencia humana, como el razonamiento, la resolución de problemas, el aprendizaje, la percepción y la toma de decisiones.

Machine Learning (ML)

El Machine Learning (Aprendizaje Automático) es una subdisciplina de la IA que se centra en el desarrollo de algoritmos y modelos informáticos que permiten a las computadoras aprender y mejorar su rendimiento en tareas específicas a través de la experiencia y el análisis de datos, en lugar de ser programadas explícitamente para cada tarea; en esencia, el Machine Learning se basa en la idea de que las computadoras pueden aprender patrones y tomar decisiones basadas en datos sin ninguna intervención humana directa.

Aprendizaje Supervisado

El Aprendizaje Supervisado es un enfoque dentro del campo del Machine Learning en el que se entrena un modelo utilizando un conjunto de datos de entrenamiento que contiene ejemplos etiquetados; en el aprendizaje supervisado, el objetivo principal es que el modelo aprenda a mapear las entradas (características) a las salidas (etiquetas) de manera que pueda hacer predicciones o tomar decisiones en nuevos datos no etiquetados.

Aprendizaje Supervisado para Regresión

El Aprendizaje Supervisado para Regresión es una subcategoría del aprendizaje supervisado en el campo del Machine Learning que se utiliza cuando el objetivo principal consiste en predecir un valor numérico o continuo en función de las características de entrada; en la regresión supervisada se busca predecir un valor numérico específico.

Aprendizaje Supervisado para Clasificación

El Aprendizaje Supervisado para Clasificación es una rama del Machine Learning en la que se entrena un modelo utilizando un conjunto de datos de entrenamiento que contiene ejemplos etiquetados, con el objetivo de que el modelo pueda predecir a qué clase o categoría pertenece una nueva observación en función de sus características de entrada.

Aprendizaje No Supervisado

El Aprendizaje No Supervisado es un enfoque dentro del campo del Machine Learning en el que un algoritmo o modelo se entrena utilizando un conjunto de datos que no contiene etiquetas o categorías predefinidas para las observaciones; a diferencia del aprendizaje supervisado, en el aprendizaje no supervisado, el objetivo principal es descubrir patrones, estructuras ocultas o relaciones entre datos sin la guía de etiquetas previamente conocidas.

Aprendizaje No Supervisado para Agrupación

El Aprendizaje No Supervisado para Agrupación se utiliza cuando el objetivo principal es dividir un conjunto de datos en grupos o clústeres, de modo que las observaciones dentro de cada grupo sean más similares entre sí que con las observaciones en otros grupos; este proceso de agrupación se realiza sin el uso de etiquetas o categorías predefinidas, y el algoritmo busca encontrar patrones en los datos que permitan formar grupos coherentes.

Aprendizaje por Refuerzo

El Aprendizaje por Refuerzo es un enfoque dentro del campo del Machine Learning que se centra en el desarrollo de algoritmos y modelos de IA que aprenden a tomar decisiones secuenciales óptimas en un entorno interactivo; a diferencia del aprendizaje supervisado, donde se proporcionan ejemplos etiquetados, y del aprendizaje no supervisado, que se enfoca en patrones de datos no etiquetados, el aprendizaje por refuerzo se basa en la interacción del agente de aprendizaje con su entorno y el concepto de recompensa.

Deep Learning (DL)

El Deep Learning (Aprendizaje Profundo) es una subdisciplina de la IA que se centra en el desarrollo de algoritmos y modelos de aprendizaje automático basados en redes neuronales artificiales profundas; estas redes neuronales profundas están diseñadas para imitar la estructura y el funcionamiento del cerebro humano, con capas de neuronas artificiales interconectadas que procesan y transforman datos de entrada en representaciones cada vez más abstractas y sofisticadas, las cuales permiten que el modelo pueda aprender representaciones jerárquicas de los datos a medida que fluyen a través de las capas.

Redes Neuronales Profundas

Las Redes Neuronales Profundas son un tipo de modelo de Machine Learning inspirado en la estructura y funcionamiento del cerebro humano, se caracterizan por tener múltiples capas de unidades computacionales llamadas neuronas artificiales o perceptrones, a medida que los datos fluyen a través de las capas, las redes neuronales profundas aprenden y representan características cada vez más abstractas y complejas de los datos de entrada.

Retropropagación

La Retropropagación (backpropagation) es un algoritmo esencial en el entrenamiento de redes neuronales artificiales, especialmente en redes neuronales profundas, su función principal es ajustar los pesos y sesgos de las conexiones entre neuronas para minimizar el error de salida de la red neuronal durante el proceso de entrenamiento; la retropropagación es una técnica de optimización que utiliza el descenso de gradiente para actualizar los parámetros de la red y mejorar su capacidad para hacer predicciones precisas.

Descenso de Gradiente

El Descenso de Gradiente (Gradient Descent) es un algoritmo de optimización utilizado en el aprendizaje automático y en la resolución de problemas de optimización en general, su principal objetivo es encontrar el mínimo de una función o superficie de error al ajustar los parámetros de un modelo de manera iterativa; el descenso de gradiente se basa en la idea de que, en la vecindad de un mínimo local de una función, el valor del gradiente (la derivada de la función) indica la dirección en la que se debe mover para alcanzar ese mínimo.

Ciencia de Datos

La Ciencia de Datos es un campo interdisciplinario que se enfoca en el estudio y la extracción de conocimiento a partir de datos, combina estadística, informática, matemáticas y conocimientos de dominio para analizar y comprender conjuntos de datos, identificar patrones, tomar decisiones basadas en datos, realizar predicciones mediante ML y generar información útil para la toma de decisiones en una variedad de campos y aplicaciones.

Análisis Descriptivo

El Análisis Descriptivo es una fase fundamental en el análisis de datos que se centra en la descripción y resumen de un conjunto de datos, con el objetivo de comprender sus características principales, patrones y tendencias; esta fase proporciona una visión general de los datos antes de realizar análisis más avanzados o modelado estadístico y ayuda a los científicos de datos a obtener una comprensión inicial de los datos que están analizando.

Análisis Predictivo

El Análisis Predictivo es un enfoque dentro de la ciencia de datos que utiliza técnicas estadísticas y de Machine Learning para hacer predicciones o proyecciones sobre eventos futuros o resultados basados en datos históricos y patrones identificados en esos datos; la ideas se trata de utilizar información pasada para prever lo que podría suceder en el futuro.

Análisis Prescriptivo

El Análisis Prescriptivo es un enfoque avanzado dentro de la ciencia de datos que se centra en proporcionar recomendaciones y soluciones óptimas para la toma de decisiones basadas en datos; a diferencia del análisis descriptivo, que se enfoca en describir lo que ha ocurrido en el pasado, y del análisis predictivo, que se enfoca en predecir eventos futuros, el análisis prescriptivo va un paso más allá al proporcionar recomendaciones y orientación sobre qué acciones tomar para lograr un resultado deseado u optimizar un objetivo específico.

2.2 Investigaciones Previas

Aplicaciones de IA en Finanzas

Estudios previos han demostrado fehacientemente la elevada eficacia de diversos algoritmos de IA, incluyendo redes neuronales profundas, en aplicaciones financieras y crediticias, como la detección de fraudes y la predicción de riesgos crediticios.

Comparaciones de Algoritmos de IA

La literatura académica y la industria financiera han realizado comparaciones entre algoritmos de ML clásicos y DL en diversas aplicaciones, estos estudios han arrojado resultados variados y han destacado la importancia de la adecuada elección de un algoritmo en función del problema específico para conseguir los resultados esperados.

Gap de Investigación

A pesar de la creciente adopción de técnicas de IA en la industria financiera, existe una falta de estudios específicos que comparen de manera exhaustiva los algoritmos de DL con los algoritmos de ML clásicos en el contexto del impago de préstamos bancarios; este TFM aborda esta laguna de investigación y busca complementar y proporcionar una comprensión más profunda de la eficacia relativa de estos enfoques en un escenario financiero real.

2.3 Hipótesis de Investigación

Con base en el Marco Teórico, se formulan las siguientes hipótesis de investigación:

H1: Los algoritmos de Machine Learning serán más interpretables y requerirán menos recursos computacionales en comparación con los algoritmos de Deep Learning.

H2: Los algoritmos de Deep Learning superarán a los algoritmos de Machine Learning en términos de precisión en la predicción de incumplimiento de pago de préstamos.

Este Marco Teórico sienta las bases para el análisis comparativo de los algoritmos de IA (tanto de ML como de DL) para la predicción de impago de préstamos en la SBA, brindando una comprensión sólida de los conceptos clave, teorías relevantes y antecedentes de investigación en el campo; estos fundamentos son esenciales para la formulación de hipótesis y el diseño de experimentos que se llevarán a cabo en este TFM.

1. Ámbito de la Aplicación

El contexto de este Trabajo de Fin de Máster (TFM) se centra en la comparación de algoritmos de Machine Learning (ML) clásicos frente a algoritmos de Deep Learning (DL), aplicados al caso de estudio de la “Predicción del incumplimiento de pago de préstamos bancarios garantizados por la U.S. Small Business Administration (SBA)”.

La predicción del incumplimiento de pago de préstamos bancarios resulta de vital importancia en el ámbito financiero, el propósito principal de este estudio es mejorar el proceso de toma de decisiones relacionado con la aprobación de préstamos en base a los datos históricos existentes de la U.S. SBA optimizando la precisión y eficiencia.

La SBA fue creada en 1953 con el principio de colaborar a las pequeñas empresas en el mercado crediticio de EE. UU., las pequeñas empresas han sido una fuente principal de creación de empleo, por lo tanto, fomentar la formación y el crecimiento de pequeñas empresas tiene beneficios sociales al crear oportunidades laborales y reducir el desempleo. Existieron varias historias de éxito de empresas emergentes que recibieron garantías de préstamo de la SBA, como FedEx y Apple Computer, sin embargo, también existieron historias de empresas que han incumplido con sus préstamos garantizados por la SBA.

(Mickel & Taylor, 2018)

El propósito principal de este TFM es evaluar y comparar el rendimiento de algoritmos de ML tradicionales (como regresión logística y árboles de decisión) de frente a algoritmos de DL (como redes neuronales profundas) en la predicción del incumplimiento de pago de préstamos bancarios, identificando ventajas y desventajas de cada enfoque en términos de precisión, interpretabilidad y eficiencia, proporcionado recomendaciones prácticas para mejorar el proceso de toma de decisiones en relación con la aprobación de préstamos.

En el presente TFM nos centraremos en el desarrollo y evaluación de modelos de ML y DL para la predicción del incumplimiento de pagos en la SBA, el alcance planificado incluye:

* La recopilación y análisis de datos históricos de préstamos de la SBA.
* Implementación y entrenamiento de modelos de ML y DL.
* Evaluación comparativa de los modelos utilizando métricas de rendimiento.
* Discusión de resultados y recomendaciones.

1. Estado del Arte

En el presente capítulo se proporciona una visión integral de la investigación y avances relevantes en el campo de la predicción del impago de préstamos bancarios; se destacan estudios previos, enfoques metodológicos y tecnologías utilizadas en investigaciones relacionadas con la aplicación de algoritmos de Inteligencia Artificial (IA) en este dominio.

Paper 1: Predicting Bank Loan Default with Extreme Gradient Boosting

La predicción del incumplimiento de préstamos es uno de los problemas más importantes y críticos que enfrentan los bancos, ya que tiene un efecto enorme sobre las ganancias; aunque existen muchos métodos tradicionales para extraer información sobre una solicitud de préstamo, la mayoría de estos métodos parecen tener un rendimiento deficiente, ya que se tiene conocimiento de aumentos en el número de préstamos incobrables.

En esta investigación, se utiliza el algoritmo XGBoost para la predicción del incumplimiento de pago de los préstamos, la predicción se basa en préstamos de un banco importante considerando datos tanto de la solicitud de préstamo como de la demografía del solicitante, también se presenta importantes métricas de evaluación, proporcionando una base efectiva para la aprobación de créditos de préstamos con el fin de identificar clientes riesgosos de una gran cantidad número de solicitudes de préstamos utilizando modelos predictivos.

El resultado más relevante de este estudio fue determinar la importancia de las características que ayudan al clasificador a predecir correctamente el incumplimiento del préstamo. En la Tabla 1 se observan las métricas de rendimiento obtenidas por el algoritmo XGBoost (luego de encontrar los parámetros óptimos) en el conjunto de datos de préstamos.

**Tabla 1**  
Performance metric for XGBoost algorithm on the loan dataset.

|  |  |
| --- | --- |
| **Metric** | **Score (%)** |
| Accuracy | 79 |
| Precision | 97 |
| Recall | 79 |
| F1-Score | 87 |

*Nota.* (Odegua, Predicting Bank Loan Default with Extreme Gradient Boosting, 2002)

Paper 2: A study on predicting loan default based on the random forest algorithm

Con el avance del comercio electrónico las plataformas de préstamos en línea P2P han traído oportunidades para los empresarios, pero al mismo tiempo, también se enfrentan al riesgo de impago de los préstamos de los usuarios; por lo tanto, basándose en el algoritmo Random Forest, este artículo construye un modelo de predicción de incumplimiento de préstamos en vista de los datos de préstamos de usuarios del mundo real.

Se adopta el método SMOTE para abordar el problema del desequilibrio de clases en el conjunto de datos (98.47% vs 1.53%), y luego se llevan a cabo una serie de operaciones como la limpieza de datos (115,000 instancias), ingeniería de características (15 variables seleccionadas) y la reducción de dimensionalidad; los resultados experimentales muestran que el algoritmo Random Forest supera a la regresión logística, el árbol de decisión y otros algoritmos de aprendizaje automático en la predicción de muestras predeterminadas.

El experimento en este articulo muestra en la Tabla 2 que el algoritmo de Random Forest tiene un rendimiento sobresaliente respecto a los otros tres algoritmos testeados en la predicción del impago de los préstamos y tiene una gran capacidad de generalización.

**Tabla 2**  
Evaluation metrics comparison of the four techniques.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rank** | **Classifier** | **Accuracy (%)** | **AUC (%)** | **F1-Score (%)** | **Recall (%)** |
| 1 | Random Forest | 98 | 98 | 98 | 98 |
| 2 | Decision Tree | 95 | 96 | 96 | 95 |
| 3 | SVM | 75 | 76 | 75 | 76 |
| 4 | Logistic Regression | 73 | 73 | 74 | 73 |

*Nota.* (Lin Zhu, 2019)

Paper 3: Loan Default Prediction Model Using Sample, Explore, Modify, Model, and Assess

El propósito de este estudio fue proporcionar una investigación exhaustiva y desarrollar un modelo para predecir el impago de préstamos, para hacer frente a este problema, se realizó la revisión de la literatura para estudiar los factores que conducen a este problema, además, estos estudios revisados se centraron en la aplicación de técnicas de extracción de datos para la predicción y clasificación de los incumplimientos de los préstamos.

Este estudio utilizó principalmente la metodología denominada SEMMA, durante la fase de experimentación, se aplicaron técnicas diferentes de minería de datos para el modelo propuesto y se evaluó su desempeño en función de varios parámetros; en funciona a estos parámetros, se seleccionó y sugirió el mejor método debido a sus características en cuanto a la predicción de los incumplimientos de los préstamos en el sector financiero.

Entre otras técnicas y luego de la normalización de los datos, se empleó redes neuronales para el desarrollo del modelo propuesto, la red neuronal empleada estaba compuesta por 11 neuronas en la capa de entrada (como variables independientes), mientras que la primera capa oculta tiene 9 neuronas y la segunda capa oculta tiene 5 neuronas, además, la última capa de salida tiene 2 neuronas como clasificadores del modelo propuesto. La arquitectura del modelo se basó en el Perceptrón Multicapa (MLP), con el 80% del conjunto de datos para el entrenamiento y el 20% para fines de validación del modelo. Los resultados finales se muestran en la Tabla 3 donde se observa un rendimiento estable de MLP.

**Tabla 3**  
Loan default prediction model results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy (%)** | **Sensitivity (%)** | **Specificity (%)** | **Error Rate (%)** |
| Decision Tree | 79.8 | 78.8 | 80.0 | 21.2 |
| Logistic Regression | 80.9 | 98.3 | 42.7 | 19.1 |
| Neural Network | 83.1 | 83.4 | 80.9 | 16.9 |

*Nota.* (Hafiz Ilyas Tariq, 2019)

Paper 4: Predicting Default Risk on Peer-to-Peer Lending Imbalanced Datasets

En este estudio se utilizan varios esquemas de Machine Learning para predecir el riesgo de incumplimiento de los préstamos P2P, también se analizan mecanismos de resampling para procesar conjuntos de datos desequilibrados; esto debido a que los conjuntos de datos desequilibrados son bastante comunes en el mundo real, como el fraude con tarjetas de crédito en transacciones; lamentablemente, los datos desequilibrados no son compatibles con los esquemas normales de aprendizaje automático, en el contexto de préstamos, los modelos sin ningún método adaptativo se centrarían en aprender el pago normal, sin embargo, la característica de la clase minoritaria es fundamental en el negocio.

Durante la experimentación se utilizaron métodos de resampling, que hacen que los conjuntos de datos se equilibren al cambiar la clase de distribución, se aplicaron el submuestreo que hace que la clase más grande alcance un tamaño cercano al de la clase más pequeña, y el sobremuestreo que hace que la clase pequeña alcance un tamaño cercano al de la clase más grande, para este caso se utilizó también la SMOTE.

En los resultados del experimento, el submuestreo aleatorio muestra el mejor rendimiento en diferentes clasificadores; después de realizar el preprocesamiento y la selección de características, el esquema propuesto puede aumentar la precisión de la predicción del riesgo de incumplimiento de pagos, en la Tabla 4 se muestran los mejores resultados.

**Tabla 4**The result of using three important features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Random Under-sampling** | **Accuracy (%)** | **Recall (%)** | **F1-Score (%)** | **G-mean** |
| Random Forest | 63.93 | 60.88 | 42.92 | 62.81 |
| Neural Networks | 63.56 | 66.46 | 44.83 | 64.57 |
| Logistic Regression | 63.24 | 66.15 | 44.49 | 64.25 |

*Nota.* (Yen-Ru Chen, 2021)

Comparación de los mejores resultados obtenidos en los Papers

En este capítulo se ha proporcionado una base sólida para la comprensión de los avances y desafíos en la predicción del incumplimiento de pagos de préstamos bancarios, que servirá como referencia esencial para la justificación del estudio comparativo de algoritmos de IA en el contexto establecido, a manera de resumen en la Tabla 5 se presentan los papers abordados con los mejores modelos y resultados obtenidos en cada uno.

**Tabla 5**  
Comparación de los mejores resultados obtenidos en los papers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper** | **Model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-Score (%)** |
| Predicting Bank Loan Default with Extreme Gradient Boosting | XGBoost | 79 | 97 | 79 | 87 |
| A study on predicting loan default based on the random forest algorithm | Random Forest | 98 | - | 98 | 98 |
| Loan Default Prediction Model Using Sample, Explore, Modify, Model, and Assess | Neural Network | 83 | 83 | 81 | - |
| Predicting Default Risk on Peer-to-Peer Lending Imbalanced Datasets | Random Forest | 64 | - | 61 | 43 |

1. Justificación de la Propuesta

La industria financiera, en particular el sector bancario, se enfrenta a retos significativos en la gestión de riesgos relacionados con la concesión de préstamos, lo que implica la necesidad de una evaluación exhaustiva del riesgo crediticio; la capacidad de predecir el incumplimiento de pago de préstamos es esencial para la toma de decisiones financieras efectivas y la protección de los intereses de las instituciones bancarias y de los prestatarios.

La Inteligencia Artificial (IA) ha emergido como una herramienta poderosa para abordar estos desafíos, y en particular, los Algoritmos de Machine Learning (ML) y Deep Learning (DL) han demostrado un gran potencial en la predicción del incumplimiento de pago en diferentes contextos financieros, tal como se lo ha revisado en el Estado del Arte.

La justificación de esta propuesta de TFM se basa en varios argumentos importantes:

Mejora en la Precisión Predictiva: Los algoritmos de IA, especialmente los de DL, tienen la capacidad de modelar relaciones complejas y no lineales en los datos, lo que puede traducirse en una mayor precisión en la predicción del incumplimiento de pago de préstamos, esta precisión es esencial para la gestión efectiva de riesgos financieros.

Eficiencia Operativa: La automatización de procesos de toma de decisiones mediante algoritmos de IA puede mejorar significativamente la eficiencia operativa de las instituciones financieras, acelerando la evaluación crediticia y reduciendo el tiempo de la revisión manual.

Ventaja Competitiva: Las instituciones financieras que adoptan soluciones basadas en IA de vanguardia pueden ganar una considerable ventaja competitiva al ofrecer servicios más personalizados y basados en datos, lo que puede atraer a clientes y prestatarios.

Contribución a la Investigación: Este TFM contribuirá tanto a la investigación académica como a la práctica en el sector financiero al proporcionar una evaluación de la eficacia de los algoritmos de ML tradicionales en comparación con los algoritmos de DL en el contexto de la predicción del incumplimiento de pago de préstamos garantizados por la SBA.

5.1 Objetivo General

El objetivo general de este TFM es evaluar y comparar el desempeño de algoritmos de Aprendizaje Automático Clásico (Machine Learning) con algoritmos de Aprendizaje Profundo (Deep Learning) en la tarea de predicción del incumplimiento de pago de préstamos bancarios en el contexto de la U.S. Small Business Administration (SBA).

Este estudio busca determinar cuál de estos enfoques es más eficaz para mejorar la precisión en la identificación de préstamos de alto riesgo y, por lo tanto, contribuir a la toma de decisiones más informadas y la mitigación de riesgos financieros en el ámbito de la SBA.

5.2 Objetivos Específicos

Para lograr el objetivo general, se plantean los siguientes objetivos específicos:

Análisis y Preprocesamiento de Datos

Analizar el conjunto de datos histórico de préstamos, incluyendo datos financieros y crediticios de las empresas solicitantes, y datos de aprobación o incumplimiento de pago.

Realizar el preprocesamiento de datos que incluya la limpieza de datos, la eliminación de valores atípicos, la imputación de datos faltantes y la normalización de atributos.

Selección de Algoritmos y Parámetros

Seleccionar cuidadosamente algoritmos de ML y de DL que sean los más representativos y relevantes para la tarea de predicción del incumplimiento de pago de préstamos.

Ajustar y optimizar los parámetros de los algoritmos seleccionados mediante técnicas de búsqueda de hiperparámetros con el objetivo de maximizar su rendimiento predictivo.

Implementación y Experimentos

Implementar los algoritmos seleccionados y entrenar los modelos utilizando los datos preparados en el objetivo específico de Análisis y Preprocesamiento de los datos.

Realizar experimentos exhaustivos que involucren técnicas de validación cruzada para evaluar el rendimiento de los modelos seleccionados en términos de las métricas de exactitud, precisión, sensibilidad, F1-score y área bajo la curva ROC (AUC).

Análisis y Comparación de Resultados

Analizar detalladamente los resultados de los experimentos para evaluar la eficacia relativa de los algoritmos de DL y ML en la predicción del incumplimiento de pago de préstamos.

Identificar las fortalezas y debilidades de cada enfoque y proporcionar conclusiones fundamentadas sobre cuál de ellos es más adecuado para el caso de la SBA.

1. Descripción de la Propuesta

La descripción de la propuesta se basa en un enfoque comparativo entre algoritmos de ML y DL para evaluar su desempeño en la predicción de impago de préstamos; este estudio se encuadra dentro de un diseño experimental que implica la implementación y evaluación de varios algoritmos, con el objetivo de determinar cuál es más eficaz en el contexto de la SBA.

Análisis de Datos

El análisis de datos se llevará a cabo utilizando los conjuntos de datos históricos disponibles de los préstamos gestionadas por la SBA, estos datos incluyen información financiera y crediticia de las empresas solicitantes, así como el estado de pago de los créditos.

Preprocesamiento

Los datos recopilados se someterán a un riguroso preprocesamiento que incluirá la limpieza de datos, la eliminación de valores atípicos y la imputación de datos faltantes; también serán codificadas las variables categóricas y se estandarizarán las variables numéricas.

Algoritmos de ML y DL

Se seleccionarán cuidadosamente los algoritmos de ML clásicos y de DL que serán objeto de comparación en este estudio, los parámetros de estos algoritmos se ajustarán mediante técnicas de búsqueda avanzada de hiperparámetros para optimizar su rendimiento.

Experimentación y Resultados

Se realizarán experimentos exhaustivos utilizando técnicas de validación cruzada para evaluar el rendimiento de los algoritmos seleccionados, las métricas de evaluación incluirán exactitud, precisión, sensibilidad, F1-score y área bajo la curva ROC (AUC); los resultados obtenidos serán esenciales para la comparación de los algoritmos de ML y DL, y así poder establecer cuál de los enfoques aplicados resulta el óptimo para realizar la predicción.

Software y Herramientas

La implementación de los algoritmos y la realización de experimentos se llevarán a cabo en Google Colab y Visual Studio Code, utilizando el lenguaje Python y librerías ampliamente reconocidas en la IA como Pandas, Numpy, Matplotlib, Scikit-Learn, TensorFlow y Keras.

1. Desarrollo de la Propuesta

7.1 Análisis Exploratorio de Datos

7.1.1 Comprensión de los Datos

La comprensión de los datos es un paso fundamental en el proceso de investigación que nos permite obtener información valiosa sobre el conjunto de datos en estudio, en este TFM exploramos el conjunto de datos obtenido de la plataforma Kaggle[[1]](#footnote-1) (ampliamente reconocida para competiciones de ciencia de datos y conjuntos de datos públicos) que se centra en la predicción del incumplimiento de pago de préstamos bancarios garantizados por la SBA (Should This Loan be Approved or Denied?, 2021)[[2]](#footnote-2), para determinar si un préstamo puede ser aprobado o denegado, la comprensión adecuada de los datos es vital para el éxito del análisis comparativo de algoritmos de Machine Learning y Deep Learning.

El conjunto de datos se compone de registros históricos de préstamos respaldados por la SBA, que incluyen información sobre los prestatarios, las características de los préstamos y su estado de cumplimiento o incumplimiento; se cuenta con un importante volumen de datos de 899164 instancias y 27 variables, en la Tabla 6 se describen todas las variables.

**Tabla 6**  
Diccionario de Datos.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Tipo de Dato** | **Descripción** |
| LoanNr\_ChkDgt | Text | Identificador - Clave principal |
| Name | Text | Nombre del prestatario |
| City | Text | Ciudad del prestatario |
| State | Text | Estado del prestatario |
| Zip | Text | Código postal del prestatario |
| Bank | Text | Nombre del banco |
| BankState | Text | Estado del banco |
| NAICS | Text | Código del sistema de clasificación de la industria de América del Norte |
| ApprovalDate | Date/Time | Fecha de emisión del compromiso de la SBA |
| ApprovalFY | Text | Año fiscal del compromiso |
| Term | Number | Plazo del préstamo en meses |
| NoEmp | Number | Número de empleados de la empresa |
| NewExist | Text | 1 = Negocio existente, 2 = Nuevo negocio |
| CreateJob | Number | Número de trabajos creados |
| RetainedJob | Number | Número de trabajos retenidos |
| FranchiseCode | Text | Código de franquicia: (00000 o 00001) = Sin franquicia |
| UrbanRural | Text | 1 = Urbano, 2 = Rural, 0 = Indefinido |
| RevLineCr | Text | Línea de crédito renovable: Y = Si, N = No |
| LowDoc | Text | Programa de préstamos: Y = Si, N = No |
| ChgOffDate | Date/Time | La fecha en que un préstamo se declara en mora |
| DisbursementDate | Date/Time | Fecha de desembolso |
| DisbursementGross | Currency | Monto Bruto desembolsado |
| BalanceGross | Currency | Saldo Bruto pendiente |
| MIS\_Status | Text | Estado del préstamo cancelado = CHGOFF, Pagado en su totalidad = PIF |
| ChgOffPrinGr | Currency | Importe cancelado |
| GrAppv | Currency | Importe bruto del préstamo aprobado por el banco |
| SBA\_Appv | Currency | Monto garantizado del préstamo aprobado por la SBA |

La variable NAICS (Sistema de Clasificación de la Industria de América del Norte) es un sistema de clasificación jerárquico de 2 a 6 dígitos utilizado por las agencias estadísticas federales para clasificar los establecimientos comerciales para la recopilación, el análisis y la presentación de información; en la Tabla 7 se detalla la descripción en base a los dos primeros dígitos de la clasificación NAICS que representan el sector económico.

**Tabla 7**  
Clasificación NAICS basada en sus dos primeros dígitos.

|  |  |
| --- | --- |
| **Sector** | **Descripción** |
| 11 | Agricultura, silvicultura, pesca y caza |
| 21 | Minería, explotación de canteras, y extracción de petróleo y gas |
| 22 | Utilidades |
| 23 | Construcción |
| 31-33 | Manufactura |
| 42 | Comercio al por mayor |
| 44-45 | Comercio minorista |
| 48-49 | Transporte y almacenamiento |
| 51 | Información |
| 52 | Finanzas y seguros |
| 53 | Bienes inmuebles y alquiler y arrendamiento |
| 54 | Servicios profesionales, científicos y técnicos |
| 55 | Gestión de sociedades y empresas |
| 56 | Servicios administrativos y de apoyo y gestión de residuos y remediación |
| 61 | Servicios educativos |
| 62 | Asistencia sanitaria y asistencia social |
| 71 | Artes, entretenimiento y recreación |
| 72 | Servicios de alojamiento y alimentación |
| 81 | Otros servicios (excepto administración pública) |
| 92 | Administración pública |

La variable NewExist representa si el negocio es un negocio existente (en existencia por más de 2 años) o es un nuevo negocio (en existencia por menos de o igual a 2 años).

Respecto a la variable LowDoc, para poder procesar préstamos de manera más eficiente, se implementó un programa de “Préstamo LowDoc” en el que se pueden procesar préstamos de menos de $150,000 mediante una solicitud de una página, “Y” indica préstamos con una solicitud de una página y “N” indica préstamos con más información adjunta a la solicitud, existen otros valores que serían posibles errores de entrada de datos.

La variable objetivo es MIS\_Status, que nos indica el estado de cada préstamo, cuando se encuentra en mora/impago es “CHGOFF” y cuando esta pagado en su totalidad es “PIF”.

Para una comprensión más profunda y estructurada de nuestro conjunto de datos, es esencial clasificar las variables en variables categóricas y variables numéricas, esta distinción es crucial, ya que influirá en el enfoque de análisis que aplicaremos a cada tipo de variable, la Tabla 8 resume esta clasificación para una visión estructurada de los datos.

**Tabla 8**  
Clasificación de las variables del Conjunto de Datos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categórica** | | | **Numérica** | |
| **Nominal** | **Numeral** | **Temporal** | **Discreta** | **Continua** |
| LoanNr\_ChkDgt | Zip | ApprovalDate | Term | DisbursementGross |
| Name | NAICS | ApprovalFY | NoEmp | BalanceGross |
| City | NewExist | ChgOffDate | CreateJob | ChgOffPrinGr |
| State | FranchiseCode | DisbursementDate | RetainedJob | GrAppv |
| Bank | UrbanRural |  |  | SBA\_Appv |
| BankState |  |  |  |  |
| RevLineCr |  |  |  |  |
| LowDoc |  |  |  |  |
| MIS\_Status |  |  |  |  |

7.1.2 Variables Categóricas Nominales

Las variables LoanNr\_ChkDgt, Name, City y Bank tienen un elevado número de valores distintos (como se observa en la Tabla 9), lo que significa que estas variables cuentan con una elevada cardinalidad. Estas variables no aportarían a los modelos predictivos.

**Tabla 9**  
Análisis de Variables Categóricas Nominales.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Nulos** | **Distintos** | **Distintos (%)** |
| LoanNr\_ChkDgt | 0 | 899164 | 100 |
| Name | 14 | 779583 | 86.7 |
| City | 30 | 32581 | 3.6 |
| Bank | 1559 | 5802 | 0.6 |
| State | 14 | 51 | 0 |
| BankState | 1566 | 56 | 0 |
| RevLineCr | 4528 | 18 | 0 |
| LowDoc | 2582 | 8 | 0 |
| MIS\_Status | 1997 | 2 | 0 |

La elevada cardinalidad ocurre cuando en una variable categórica existe un gran número de categorías o etiquetas distintas; esta situación plantea ciertas problemáticas al entrenar modelos de ML, porque el modelo intenta ajustarse demasiado a los datos, memorizando las categorías en lugar de aprender patrones generales (sobreajuste), agrega complejidad al modelo, el cual puede volverse menos robusto frente a variaciones en los datos por la gran cantidad de categorías. Es importante considerar estrategias como la consolidación de categorías similares, la selección de características o reducción de dimensionalidad.

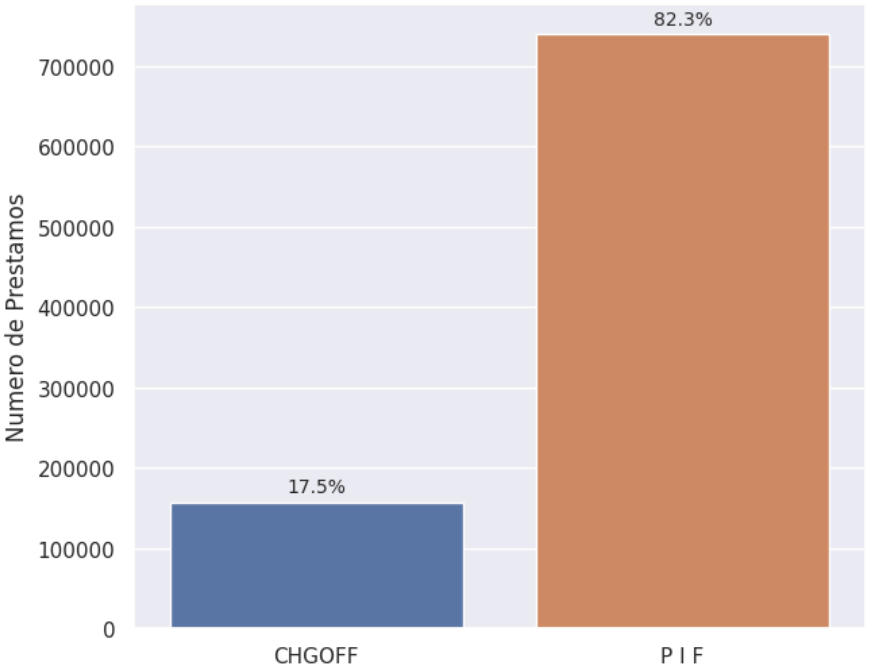
Las variables State y BankState tiene un número coherente de categorías distintas, y un número razonable (en relación al tamaño del dataset) de valores nulos, como se puede observar en la Tabla 9. Estas variables podrían aportar a los modelos predictivos.

La variable RevLineCr cuenta con 18 categorías distintas, como se observa en la Tabla 9, sin embargo, en el diccionario de datos de la Tabla 6 se menciona que debería tener únicamente dos categorías (“Y” y “N”), analizando las categorías existentes se evidenció que cuenta con categorías como “0” y “T” (entre otras) que sumadas son el 30.9% del total; tiene un número razonable de valores nulos. Esta variable luego de preprocesar las categorías no definidas y tratar los valores nulos podría aportar a los modelos predictivos.

La variable LowDoc tiene 8 categorías distintas, acorde a la Tabla 9, sin embargo, en el diccionario de datos de la Tabla 6 se menciona que debería tener solamente dos categorías (“Y” y “N”), analizando las categorías existentes se pudo evidenciar que tiene categorías como “0”, “C” y “S” (entre otras) pero en un porcentaje menor, sumadas son solo el 0.6%; tiene un número razonable de valores nulos. Esta variable luego de preprocesar las categorías no definidas y tratar los valores nulos podría aportar a los modelos predictivos.

La variable objetivo MIS\_Status tiene las dos categorías establecidas en el diccionario de datos de la Tabla 6 (para créditos impagos “CHGOFF” y para créditos pagados “PIF”), tiene también un valor razonable de valores nulos. Como se observa en la Figura 1, existe un desbalanceo de clases, lo que hace sentido debido a que en su mayoría los préstamos bancarios son pagados en su totalidad y son menores los casos de incumplimiento de pago.

**Figura 1***Distribución de la Variable Objetivo MIS\_Status*



El desbalanceo de clases ocurre en problemas de clasificación cuando las clases que se deben predecir no están representadas de manera equitativa en el conjunto de datos, si no se aborda adecuadamente esta situación, un modelo de ML puede tener un alto sesgo hacia la clase mayoritaria y puede no ser efectivo en la detección de la clase minoritaria de interés. Para abordar el desbalanceo de clases, se puede utilizar diversas estrategias, como: Sobremuestreo (generar más instancias de la clase minoritaria, replicando instancias existentes o generando instancias sintéticas), Submuestreo (eliminar el número de instancias de la clase mayoritaria para igualar la proporción), la elección de la estrategia es importante para asegurar que el modelo aprenda de manera equitativa de todas las clases.

7.1.3 Variables Categóricas Numerales

La variable Zip tiene un elevado número de valores distintos, es decir elevada cardinalidad, como se observa en la Tabla 10. Esta variable no aportaría a los modelos predictivos.

**Tabla 10**  
Análisis de Variables Categóricas Numerales.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Nulos** | **Distintos** | **Distintos (%)** |
| Zip | 0 | 33611 | 3.7 |
| FranchiseCode | 0 | 2768 | 0.3 |
| NAICS | 0 | 1312 | 0.1 |
| NewExist | 136 | 3 | 0 |
| UrbanRural | 0 | 3 | 0 |

La variable UrbanRural tiene las tres categorías establecidas en su diccionario de datos, no tiene ningún valor nulo; esta variable podría aportar a los modelos predictivos.

**Figura 2***Distribución de la Variable UrbanRural*

Gráfico, Gráfico de barras

Descripción generada automáticamente

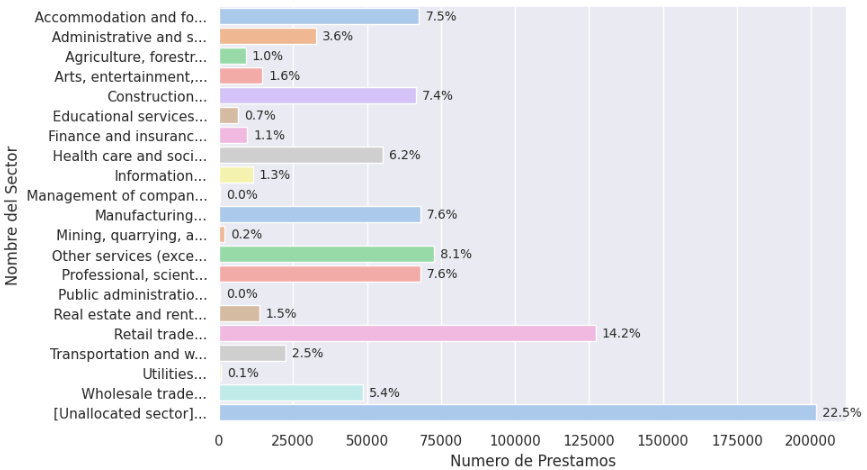
La variable NewExist tiene 3 categorías distintas, acorde a la Tabla 10, sin embargo, en el diccionario de datos de la Tabla 6 se menciona que debería tener solamente dos categorías (“1” y “2”), analizando los datos se pudo evidenciar que la tercera categoría es “0”, con un porcentaje mínimo de solo el 0.1%; tiene un número mínimo de valores nulos. Esta variable luego de preprocesar y tratar los valores nulos podría aportar a los modelos predictivos.

La variable FranchiseCode en función al diccionario de datos de la Tabla 6, donde se menciona que, si el código de franquicia es “00000” o “00001” se considera sin franquicia, analizando los datos se observa que el 94.2% serían códigos sin franquicia, es decir, se tiene un elevado grado de uniformidad. Esta variable no aportaría a los modelos predictivos.

El elevado grado de uniformidad es una situación en la que la mayoría o todos los valores de esa variable son prácticamente idénticos o muy similares entre sí, es decir, no hay mucha variabilidad en los valores de la variable, son casi uniformes u homogéneos; resulta ser problemático en ML ya que la variable no proporciona información discriminativa, se busca la variabilidad en los datos para identificar patrones, relaciones o diferencias significativas.

La variable NAICS como se observa en la Tabla 10 cuenta con 1312 categorías distintas, sin embargo, de acuerdo a la Tabla 7 se reagrupan los datos en 20 categorías distintas, con un 22.5% de datos que no pertenecen a ningún sector industrial, con los cuales creamos una nueva categoría de “Sector no Asignado” como se observa en la Figura 3. Esta variable luego de transformar y preprocesar sus datos podría aportar a los modelos predictivos.

**Figura 3***Distribución de Sectores de la Variable NAICS*



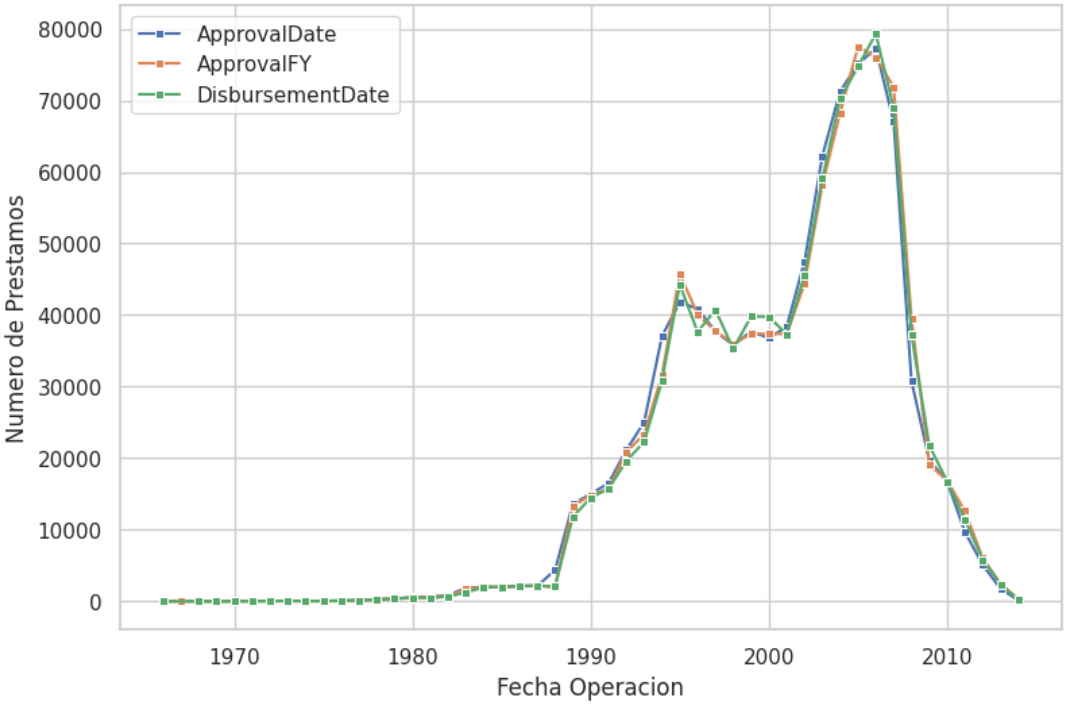
7.1.4 Variables Categóricas Temporales

La variable ApprovalDate se encuentra en el rango de años entre 1966 y 2014, no tiene ningún valor nulo, esta variable podrían aportar a los modelos predictivos.

La variable ApprovalFY se encuentra en el rango de años entre 1966 y 2014, de acuerdo al diccionario de datos de la Tabla 6 esta variable es el año fiscal del compromiso, vale decir equivaldría a extraer el año de la variable ApprovalDate, como se observa en la Figura 6, su comportamiento es muy similar. Esta variable no aportaría a los modelos predictivos.

La variable DisbursementDate se encuentra en el rango de años de 1966 y 2014, tiene 2368 valores nulos (0.3%), y como se observa en la Figura 4, su comportamiento es similar a las anteriores variables analizadas. Esta variable no aportaría a los modelos predictivos.

**Figura 4***Comparación de Variables Categóricas Temporales*



La variable ChgOffDate indica cuando un préstamo se declara en mora, por lo tanto, se trata de un Target Leakage (fuga de información de la variable objetivo) debido a que la variable contiene información posterior a la ocurrencia del evento que se desea predecir, vale decir que, si esta variable tiene una fecha “Nula” entonces el préstamo no estaría en un estado de mora o impago (en un 99.76%), y viceversa (con 96.97%), como se observa en la Tabla 11. Esta variable no debe ser considerada para los modelos predictivos.

**Tabla 11**  
Análisis del Target Leakage de la variable ChgOffDate.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha Mora** | **Impago** | **Cantidad** | **Frecuencia** |
| No Nula | N | 4927 | 3.03 |
| No Nula | S | 157772 | 96.97 |
| Nula | N | 734682 | 99.76 |
| Nula | S | 1783 | 0.24 |

El Target Leakage (fuga de información de la variable objetivo) se refiere a una situación en la que la información de la variable objetivo (la que se está tratando de predecir) se filtra de alguna manera en las características utilizados para entrenar un modelo. La fuga de información es problemática porque puede dar lugar a modelos que parecen muy precisos en el entrenamiento, pero que fallan en la generalización a nuevos datos, ya que se basan en información incorrecta o que no está disponible al momento de realizar la predicción.

7.1.5 Variables Numéricas Discretas

Las variables CreateJob y RetainedJob tienen una moda de 0, valor que se repite en un 70% y 49% respectivamente, tendencia muy pronunciada en el 0, tal como se observa en la Tabla 12, en ambos casos su valor máximo se encuentra bastante alejado de la mediana. Esta variable luego de detectar valores atípicos podría aportar a los modelos predictivos.

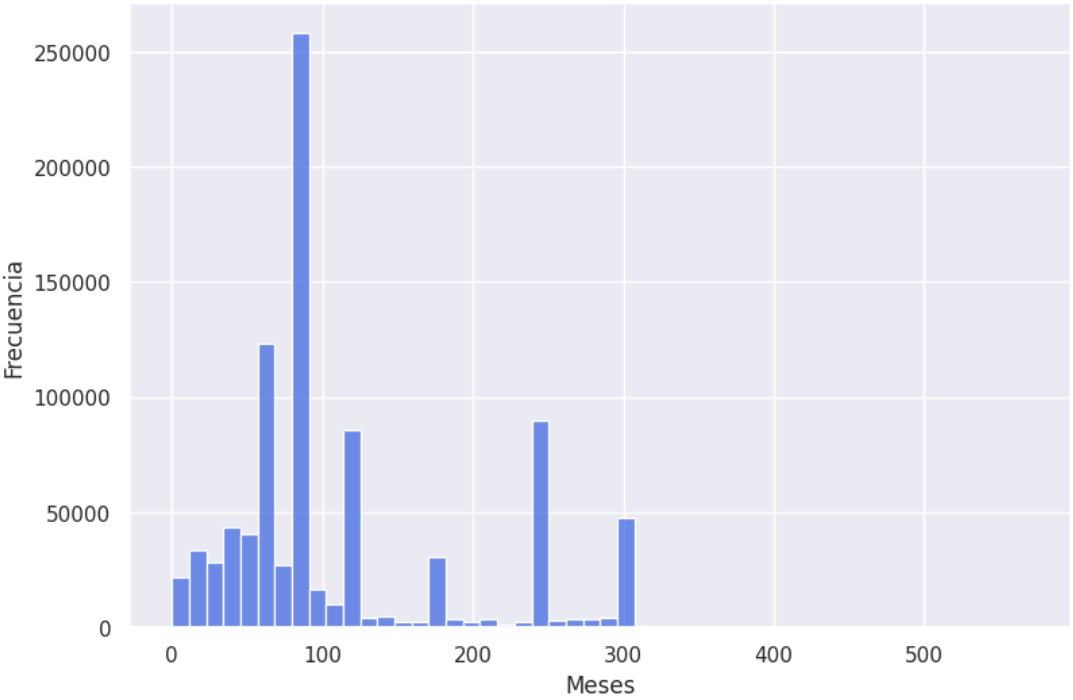
La variable NoEmp con una media de 11.4, mediana de 4 y moda de 1 (17.2%), tiene una distribución más dispersa que las dos anteriores variables, como se observa en la Tabla 12. Esta variable luego de detectar valores atípicos podría aportar a los modelos predictivos.

**Tabla 12**  
Análisis de Variables Numéricas Discretas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **Media** | **Mediana** | **Moda** | **Moda (%)** | **Mínimo** | **Máximo** |
| Term | 110.8 | 84.0 | 84 | 25.6 | 0 | 569 |
| NoEmp | 11.4 | 4.0 | 1 | 17.2 | 0 | 9999 |
| CreateJob | 8.4 | 0.0 | 0 | 70.0 | 0 | 8800 |
| RetainedJob | 10.8 | 1.0 | 0 | 49.0 | 0 | 9500 |

La variable Term representa el plazo en meses del préstamo, su mediana y moda es de 84 meses (25.6% de repeticiones), su media es de 110.8, y su rango esta entre 0 y 569 meses, rango mucho más compacto que las anteriores tres variables analizadas, tal como se muestra en la Figura 5. Esta variable podría aportar bastante a los modelos predictivos.

**Figura 5***Distribución de la Variable Term*



7.1.6 Variables Numéricas Continuas

La variable DisbursementGross tiene medidas razonables tal como se puede observar en la Tabla 13. Esta variable podría aportar a los modelos predictivos.

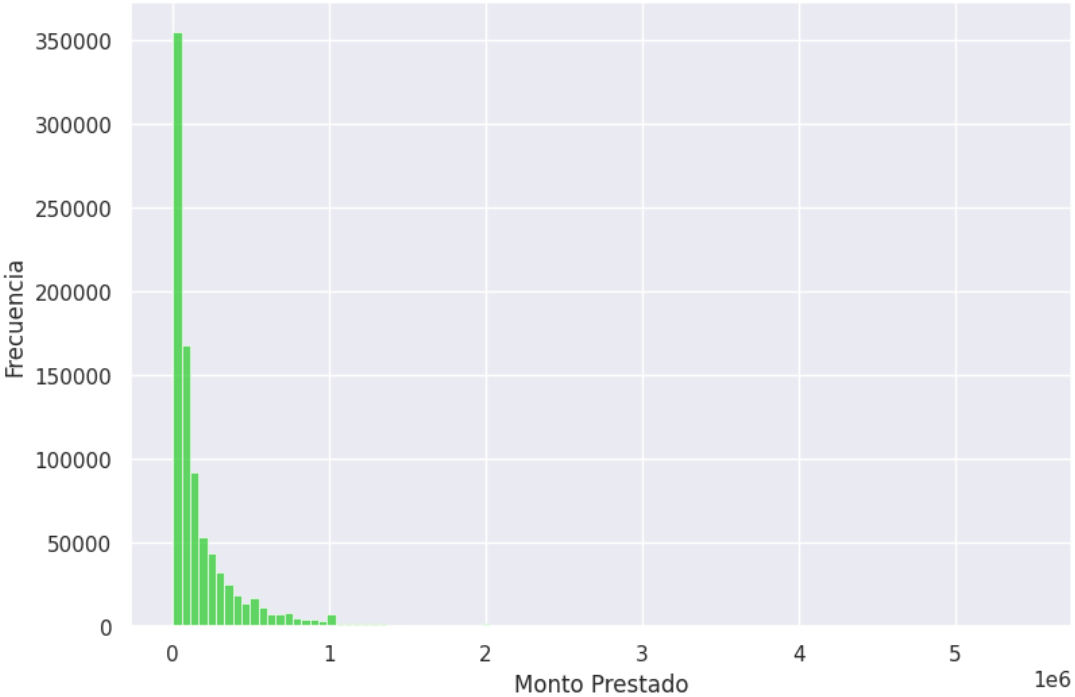
**Tabla 13**  
Análisis de Variables Numéricas Continuas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Media** | **Mediana** | **Moda** | **Mínimo** | **Máximo** |
| DisbursementGross | 201154.0 | 100000.0 | 50000 | 0 | 11446325 |
| BalanceGross | 3.0 | 0.0 | 0 | 0 | 996262 |
| GrAppv | 192687.0 | 90000.0 | 50000 | 200 | 5472000 |
| SBA\_Appv | 149488.8 | 61250.0 | 25000 | 100 | 5472000 |

La variable BalanceGross tiene 899150 valores en cero, que son prácticamente el 100%, tiene un total grado de uniformidad. Esta variable no aportaría a los modelos predictivos.

La variable GrAppv tiene medidas razonables tal como se puede observar en la Tabla 13, adicionalmente, se ha ploteado su distribución como se puede ver en la Figura 6. Esta variable luego de detectar valores atípicos podría aportar a los modelos predictivos.

**Figura 6***Distribución de la Variable GrAppv*



La variable SBA\_Appv tiene medidas razonables como se puede observar en la Tabla 13. Esta variable luego de detectar valores atípicos podría aportar a los modelos predictivos.

La variable ChgOffPrinGr indica el monto de la mora, que se determinaría después de saber si el préstamo está impago, por lo tanto, se trata de un Target Leakage, podemos saber que el préstamo está impago (Tabla 14) si el monto de la mora no es cero (96.99%) y viceversa (99.67%). Esta variable no debe ser considerada para los modelos predictivos.

**Tabla 14**  
Análisis del Target Leakage de la variable ChgOffPrinGr.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mora Cero** | **Impago** | **Cantidad** | **Frecuencia** |
| N | N | 4884 | 3.01 |
| N | S | 157128 | 96.99 |
| S | N | 734725 | 99.67 |
| S | S | 2427 | 0.33 |

7.2 Preprocesamiento de Datos

El Preprocesamiento de Datos es un conjunto de técnicas y pasos que se aplican a los datos crudos o sin procesar para prepararlos y limpiarlos antes de utilizarlos en análisis de datos o en la construcción de modelos de ML, su objetivo principal es asegurarse de que los datos sean de alta calidad, estén en un formato adecuado y sean apropiados para la tarea específica que se va a realizar. El preprocesamiento de datos es una parte crítica en proyectos de ML, ya que la calidad de los datos y su preparación adecuada tienen un impacto significativo en la efectividad y el rendimiento de los modelos resultantes.

7.2.1 Ingeniería de Características

La Ingeniería de Características es el proceso de identificar, crear y transformar características a partir de los datos brutos con el fin de representar de manera efectiva las relaciones y patrones en los datos, este proceso implica seleccionar las características más relevantes, eliminar las irrelevantes o redundantes, y transformar las características de manera que sean más adecuadas para el modelado, su objetivo principal es mejorar la capacidad de un modelo de ML para hacer predicciones y tomar decisiones informadas.

Eliminar Características

Luego de concluir con el Análisis Exploratorio de Datos (EDA) y como un paso importante en el preprocesamiento de datos, se realizó la eliminación de características que no aportarán significativamente al modelo, debido a su elevada cardinalidad, redundancia, elevada uniformidad y fuga de información, se detallan las mismas en la Tabla 15.

**Tabla 15**  
Características que fueron eliminadas.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Detalle** | **Justificación** |
| LoanNr\_ChkDgt | Tiene un 100% de valores distintos | Elevada Cardinalidad |
| Name | Tiene un 86.7% de valores distintos | Elevada Cardinalidad |
| City | Cuenta con 32581 ciudades diferentes | Elevada Cardinalidad |
| Zip | Cuenta con 33611 códigos postales | Elevada Cardinalidad |
| Bank | Cuenta con 5802 bancos diferentes | Elevada Cardinalidad |
| ApprovalFY | Comportamiento similar a ApprovalDate | Característica Redundante |
| FranchiseCode | Tiene 94.2% de códigos Sin Franquicia | Elevada Uniformidad |
| CghOffDate | Se registra de manera posterior al evento | Fuga de Información |
| DisbursementDate | Comportamiento similar a ApprovalDate | Característica Redundante |
| BalanceGross | Tiene prácticamente 100% de valores cero | Elevada Uniformidad |
| ChgOffPrinGr | Se registra de manera posterior al evento | Fuga de Información |

Transformar Características

La transformación de características es un proceso esencial para mejorar la calidad de los datos, hacer que estos sean más adecuados para los algoritmos de ML y permitir que los modelos capturen relaciones y patrones de manera más efectiva, el resumen de todas las transformaciones que fueron realizadas se describe en la Tabla 16.

**Tabla 16**  
Resumen de la transformación de características.

|  |  |
| --- | --- |
| **Variable** | **Proceso** |
| DisbursementGross | Convertir de tipo de dato Currency a Entero |
| DisbursementGross | Cambiar el nombre de la variable a GrDisburs |
| GrAppv | Convertir de tipo de dato Currency a Entero |
| GrAppv | Cambiar el nombre de la variable a GrApprov |
| SBA\_Appv | Convertir de tipo de dato Currency a Entero |
| SBA\_Appv | Cambiar el nombre de la variable a ApprovSBA |
| ApprovalDate | Crear la variable AppYear con el año respectivo |
| ApprovalDate | Crear la variable AppMonth con el mes respectivo |
| ApprovalDate | Eliminar la variable, el día no es un dato relevante |
| NAICS | Categorizar los datos acorde al Sector respectivo |
| NAICS | Cambiar el nombre de la variable a Sector |
| NewExist | Clasificar datos a 0 y 1, y NaN para indefinidos |
| NewExist | Convertir de tipo de dato Float a Entero |
| RevLineCr | Clasificar valores "Y" y "T" a 1 (se asumen True) |
| RevLineCr | Clasificar valores "N" a 0 (se asumen False) |
| RevLineCr | Clasificar valores diferentes de 1 y 0 a NaN |
| RevLineCr | Convertir de tipo de dato Object a Entero |
| RevLineCr | Cambiar el nombre de la variable a RevLine |
| LowDoc | Clasificar valores "Y" a 1 (se asumen True) |
| LowDoc | Clasificar valores "N" a 0 (se asumen False) |
| LowDoc | Clasificar valores diferentes de 1 y 0 a NaN |
| LowDoc | Convertir de tipo de dato Object a Entero |
| MIS\_Status | Clasificar valores "CHGOFF" a 1 (Default) |
| MIS\_Status | Clasificar valores "P I F" a 0 (No Default) |
| MIS\_Status | Convertir de tipo de dato Object a Entero |
| MIS\_Status | Cambiar el nombre de la variable a Default |
| State y BankState | Crear DifState, si los Estados son iguales es 0 y si no 1 |
| Term | Crear Secured, 1 si es mayor igual a 240 y 0 si es menor |
| SBA\_Appv y GrAppv | Crear SecuredSBA, porcentaje de SBA\_Appv / GrAppv |
| SecuredSBA | Convertir de tipo de dato Float a Entero |

7.2.2 Imputación de Valores Faltantes

La Imputación de Valores Faltantes es un proceso que implica rellenar o estimar valores para las observaciones en un conjunto de datos que tienen valores faltantes o ausentes, los cuales son comunes en conjuntos de datos del mundo real debido a diversas razones, como errores de entrada, fallas en la recolección de datos o simplemente la falta de información para ciertas observaciones; la imputación es importante para asegurarse de que el conjunto de datos esté completo y que se puedan utilizar algoritmos de Machine Learning de manera efectiva. La Tabla 17 describe las variables y los valores faltantes del conjunto de datos.

**Tabla 17**  
Detalle de variables con valores faltantes.

|  |  |
| --- | --- |
| **Variable** | **Valores Faltantes** |
| State | 14 |
| BankState | 1564 |
| NewExist | 1169 |
| RevLine | 4568 |
| LowDoc | 4514 |
| Default | 1956 |

Imputación Univariante

La imputación univariante es una técnica de imputación de valores faltantes que se centra en reemplazar los valores faltantes en una variable específica utilizando solo la información de esa variable en lugar de utilizar información de otras variables; para tal, propósito se puede imputar usando la media o mediana para variables numéricas, y la moda para variables categóricas, también se puede utilizar valores constantes. Para nuestras variables State y BankState se ha utilizado la imputación mediante la moda (Estado más repetido).

Imputación Multivariante

La imputación multivariante es una técnica de imputación de valores faltantes que tiene en cuenta las relaciones y dependencias entre múltiples variables en un conjunto de datos, considera múltiples variables al realizar la imputación (identificando variables relacionadas), esto permite una imputación más precisa y realista, ya que se aprovechan las relaciones entre las variables para estimar los valores faltantes; para tal propósito se puede imputar usando regresión o vecinos cercanos para variables numéricas, y algún clasificador para variables categóricas. Para las variables NewExist, RevLine y LowDoc se ha utilizado RandomForest para imputar sus valores, las variables independientes utilizadas fueron: AppYear, AppMonth, Term, NoEmp, UrbanRural, GrDisburs, GrApprov y ApprovSBA.

La eliminación de valores faltantes se refiere a eliminar filas o columnas que contienen valores faltantes en un conjunto de datos, esto se hace con el propósito de limpiar los datos y garantizar que las observaciones restantes estén completas y no tengan valores faltantes; la eliminación de valores faltantes se utiliza en situaciones específicas en las que no es factible o apropiado imputar o rellenar los valores faltantes. En nuestro caso para la variable objetivo Default se eliminaron las instancias con valores faltantes considerando que tiene una mínima cantidad respecto al volumen total de las instancias del conjunto de datos.

7.2.3 Detección de Valores Atípicos

La Detección de Valores Atípicos es el proceso de identificar observaciones en un conjunto de datos que se desvían significativamente del comportamiento esperado o de la mayoría de las otras observaciones, estos valores atípicos son inusuales en relación con el resto de los datos y pueden ser el resultado de errores, eventos raros o patrones inusuales en los datos. Los valores atípicos pueden distorsionar estadísticas descriptivas, como la media y la varianza lo que puede llevar a conclusiones erróneas, por lo que es importante elegir una técnica adecuada acorde a la naturaleza de los datos y el problema específico.

Desviación Estándar y Z-score

Estas técnicas se basan en la idea de que los valores atípicos son observaciones que se desvían significativamente de la media del conjunto de datos. La desviación estándar es una medida de la dispersión de los datos en torno a la media, los valores atípicos suelen estar lejos de la media y, por lo tanto, pueden identificarse mediante la comparación de la distancia entre un valor y la media. El Z-Score es una medida que cuantifica cuántas desviaciones estándar un valor está lejos de la media. Se establece un umbral para definir qué valores se considerarán atípicos en función del z-score, por ejemplo, con un umbral de Z-score de ±3, los valores por encima o debajo de estos valores se considerarán atípicos.

Los resultados de aplicar las técnicas de Desviación Estándar y Z-score en las variables numéricas con un umbral de 3, se encuentran detallados en la Tabla 18.

**Tabla 18**  
Resultados de aplicar Desviación Estándar y Z-score.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Media** | **Desviación Estándar** | **Límite Máximo** | **Valores Atípicos** |
| Term | 110.9 | 78.9 | 347.6 | 183 |
| NoEmp | 11.4 | 73.8 | 232.8 | 1,449 |
| CreateJob | 8.4 | 236.8 | 718.9 | 676 |
| RetainedJob | 10.8 | 237.3 | 722.6 | 674 |
| DisbursementGross | 201,627.2 | 287,822.9 | 1,065,095.8 | 19,260 |
| GrAppv | 193,085.0 | 283,448.4 | 1,043,430.2 | 19,605 |
| SBA\_Appv | 149,800.1 | 228,571.8 | 835,515.6 | 18,049 |

Rango Intercuartílico (IQR)

La técnica del rango intercuartílico (IQR) es un método estadístico comúnmente utilizado para la detección de valores atípicos en un conjunto de datos, se basa en la distribución de los datos y se utiliza para identificar valores que se encuentran por encima o por debajo de umbrales definidos por los cuartiles, el IQR se calcula de la siguiente manera:

1. Calcular el primer cuartil (Q1), que es el valor en el que el 25% de los datos se encuentra por debajo.
2. Calcular el tercer cuartil (Q3), que es el valor en el que el 75% de los datos se encuentra por debajo.
3. Calcular el rango intercuartílico (IQR) que es la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1): IQR = Q3 - Q1.
4. Definir los dos umbrales:
   * Límite inferior = Q1 − 1.5 × IQR
   * Límite superior = Q3 + 1.5 × IQR.
5. Cualquier valor que se encuentre por debajo del límite inferior o por encima del límite superior se considera un valor atípico.

El rango intercuartílico es una técnica robusta para la detección de valores atípicos, ya que no se ve afectado por valores extremos en la misma medida que otros métodos; el factor de 1.5 en la definición de los límites puede variar según el contexto y las preferencias, y se puede ajustar si se desea una detección más o menos estricta de valores atípicos.

Los resultados de aplicar la técnica de Rango Intercuartílico en las variables numéricas con un factor de escala de 4.5, se encuentran detallados en la Tabla 19.

**Tabla 19**  
Resultados de aplicar Rango Intercuartílico.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **IQR** | **Límite Inferior** | **Límite Superior** | **Valores Atípicos** |
| Term | 60.0 | -210.0 | 390.0 | 39 |
| NoEmp | 8.0 | -34.0 | 46.0 | 33,305 |
| CreateJob | 5.0 | -20.5 | 29.5 | 9,890 |
| RetainedJob | 7.0 | -29.5 | 40.5 | 14,837 |
| DisbursementGross | 196,500.0 | -841,750.0 | 1,123,250.0 | 17,023 |
| GrAppv | 190,000.0 | -820,000.0 | 1,080,000.0 | 18,227 |
| SBA\_Appv | 153,750.0 | -670,625.0 | 866,875.0 | 16,952 |

7.2.4 Escalar Variables Numéricas

El Escalamiento de Variables Numéricas es un proceso común en el preprocesamiento de datos utilizado en análisis de datos y modelado de ML, consiste en transformar las variables numéricas para que tengan una escala común o similar, esta transformación se realiza con el objetivo de mejorar el rendimiento de los modelos y asegurar que las características contribuyan de manera equitativa al proceso de modelado.

Estandarización

En este enfoque, se resta la media de la variable y se divide por la desviación estándar, lo que lleva a que la variable tenga una media de 0 y una desviación estándar de 1, la estandarización es útil cuando se asume que los datos siguen una distribución normal y cuando se desean escalas comunes para todas las características. Estandarizar variables depende del modelo a utilizar, como algoritmos que utilizan descenso de gradiente en regresiones o redes neuronales; algunos algoritmos, como los árboles de decisión, no se ven afectados por la escala de las variables, por lo que no necesitan escalamiento.

Normalización

En este enfoque, los valores de las variables se escalan al rango de 0 a 1, la normalización es útil cuando se desea que todas las variables tengan la misma escala y se encuentren en el rango (generalmente) de 0 a 1, lo que puede ser beneficioso para algoritmos que utilizan medidas de distancia o para evitar que características con escalas muy diferentes dominen el modelo. Normalizar variables depende del modelo a utilizar, como algoritmos que utilizan medidas de distancia, como Vecinos Cercanos, K-Means y Redes Neuronales con funciones de activación basadas en distancias, como la función Sigmoide.

La elección entre estandarización y normalización depende del problema específico y la distribución de los datos, en muchos casos, se aplican ambas técnicas y se evalúa cuál funciona mejor para un conjunto de datos y un algoritmo en particular. En nuestro caso se experimentará con el escalamiento de variables según el algoritmo de ML que se aplique.

7.2.5 Codificar Variables Categóricas

La Codificación de Variables Categóricas es un proceso que se utiliza para convertir variables categóricas en un formato numérico que pueda ser utilizado por algoritmos de ML. Las variables categóricas son aquellas que representan categorías en lugar de valores numéricos, existen varias técnicas de codificación de variables categóricas, y la elección de la técnica depende del tipo de variable categórica y del algoritmo de ML que se va a utilizar.

Codificación de Etiqueta (Label Encoding)

En esta técnica, cada categoría de la variable codificada recibe una etiqueta numérica única, por ejemplo, si se tiene una variable categórica de "colores" con categorías como "rojo," "verde," y "azul," se podrían codificar las categorías como 0, 1, y 2, respectivamente. Aplicamos esta técnica para las variables categóricas State, BankState y Sector.

Codificación One-Hot (One-Hot Encoding)

En esta técnica, se crea una columna binaria para cada categoría, cada columna representa la presencia o ausencia de una categoría, es útil para variables categóricas nominales donde no existe un orden natural entre las categorías, por ejemplo, en el caso de "colores," tendríamos las columnas: "rojo," "verde," y "azul," con valores binarios 1 o 0 para indicar la presencia de cada color. Aplicamos esta técnica para la variable UrbanRural, se crean las variables Urban y Rural, pero se elimina la columna específica para los valores indefinidos, la eliminación se realiza para evitar la multicolinealidad en el conjunto de datos.

Codificador Ordinal

La codificación ordinal es una técnica de codificación de variables categóricas que se utiliza cuando las categorías tienen un orden o jerarquía natural, es especialmente útil para variables categóricas ordinales, donde existe una relación de orden lógica entre las categorías, pero no se pueden representar simplemente mediante etiquetas numéricas. Aplicamos esta técnica a la variable categórica AppYear ordenada según al año respectivo.

7.2.6 Selección de Características

La selección de características implica la elección de un subconjunto de variables disponibles en un conjunto de datos, mientras se descartan otras, su objetivo principal es identificar y retener las características más relevantes y significativas para una tarea específica, al tiempo que se eliminan las características redundantes o irrelevantes, la selección de características es importante para mejorar el rendimiento de los modelos, reducir la dimensionalidad, prevenir el sobreajuste y mejorar la interpretabilidad.

Umbral de Varianza

El método de Umbral de Varianza (Variance Threshold) es una técnica de selección de características que se utiliza para eliminar características con una varianza por debajo de un cierto umbral, la idea subyacente es que las características con una varianza muy baja, es decir, aquellas que apenas cambian en el conjunto de datos, aportan poca información útil para la construcción de modelos de aprendizaje automático, por lo tanto, estas características pueden eliminarse sin afectar significativamente el rendimiento del modelo.

Análisis de Información Mutua

El Análisis de Información Mutua (Mutual Information Analysis) es una técnica de selección de características que se utiliza para medir la relación entre las características (variables independientes) y la variable objetivo en un problema de ML, su objetivo es identificar qué características tienen una alta dependencia o relación con la variable objetivo, es útil especialmente con datos que contienen características categóricas; cuanto mayor sea la información mutua, mayor será la relación entre la característica y la variable objetivo.

El resultado de aplicar los métodos de Umbral de Varianza y Análisis de Información Mutua al conjunto de datos se puede observar en la Tabla 20; donde se evidencia que la variable CreateJob no tiene ninguna relación con la variable objetivo en el Análisis de Información Mutua y la variable RetainedJob tiene puntuaciones bastantes mínimas en ambos análisis; como conclusión, las dos variables mencionadas serán eliminadas del conjunto de datos.

**Tabla 20**  
Umbral de Varianza y Análisis de Información Mutua.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Umbral de Varianza** | |  | **Análisis de Información Mutua** | |
| NoEmp | 0.0001 |  | CreateJob | 0.0000 |
| RetainedJob | 0.0006 |  | RetainedJob | 0.0004 |
| CreateJob | 0.0007 |  | NoEmp | 0.0008 |
| GrDisburs | 0.0018 |  | Rural | 0.0059 |
| AppYear | 0.0158 |  | LowDoc | 0.0121 |
| Term | 0.0189 |  | GrApprov | 0.0122 |
| ApprovSBA | 0.0189 |  | ApprovSBA | 0.0126 |
| GrApprov | 0.0213 |  | GrDisburs | 0.0128 |
| SecuredSBA | 0.0333 |  | RevLine | 0.0242 |
| BankState | 0.0822 |  | NewExist | 0.0249 |
| State | 0.0913 |  | Secured | 0.0379 |
| AppMonth | 0.0934 |  | AppMonth | 0.0408 |
| Rural | 0.1033 |  | State | 0.0446 |
| LowDoc | 0.1087 |  | BankState | 0.0459 |
| Sector | 0.1128 |  | Sector | 0.0699 |
| Secured | 0.1378 |  | SecuredSBA | 0.0785 |
| RevLine | 0.1837 |  | DifState | 0.0791 |
| NewExist | 0.2026 |  | Urban | 0.0899 |
| DifState | 0.2492 |  | AppYear | 0.0926 |
| Urban | 0.2495 |  | Term | 0.1557 |

Adicionalmente, realizamos la Eliminación de Valores Duplicados que es relevante para mejorar la eficiencia del modelo, garantiza que las instancias de datos sean únicas y representativas, valores duplicados pueden introducir sesgo en los resultados del modelo.

7.2.7 Resultados del Preprocesamiento

Tarjeta de Datos Preprocesados

Luego de completar el preprocesamiento de datos, el conjunto de datos ha sido refinado y se encuentra preparado para ser utilizado en nuestros análisis y aplicación de modelos de Machine Learning; como se puede observar en la Tabla 21, las características han sido transformadas, codificadas y seleccionadas aplicando estrategias acordes con nuestras necesidades, y los valores faltantes han sido tratados de manera adecuada y efectiva.

**Tabla 21**  
Tarjeta de Datos Preprocesados.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Tipo de Dato** | **Valores Nulos** |
| State | int64 | 0 |
| BankState | int64 | 0 |
| DifState | int64 | 0 |
| Sector | int64 | 0 |
| AppYear | int64 | 0 |
| AppMonth | int64 | 0 |
| Term | int64 | 0 |
| NoEmp | int64 | 0 |
| Secured | int64 | 0 |
| NewExist | int64 | 0 |
| Urban | int64 | 0 |
| Rural | int64 | 0 |
| RevLine | int64 | 0 |
| LowDoc | int64 | 0 |
| GrDisburs | int64 | 0 |
| GrApprov | int64 | 0 |
| ApprovSBA | int64 | 0 |
| SecuredSBA | int64 | 0 |
| Default | int64 | 0 |

Tabla de Variables Preprocesadas

Para una comprensión más profunda y estructurada del conjunto de datos preprocesado, es fundamental volver a clasificar las variables en categóricas y numéricas, esta clasificación nos permitirá volver a identificar las características que contiene categorías discretas y las características que contienen valores numéricos, la Tabla 22 resume la clasificación para una visión clara de la estructura del conjunto de datos preprocesado.

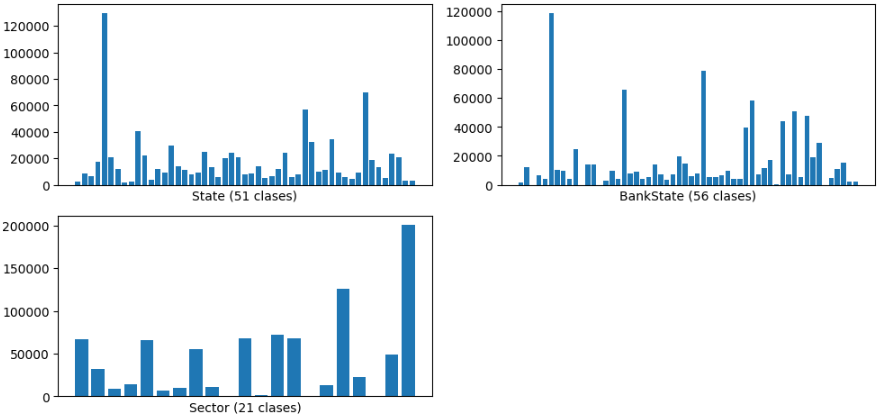
**Tabla 22**  
Clasificación de las variables del conjunto de datos preprocesado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categórica** | | | **Numérica** | |
| **Nominal** | **Ordinal** | **Binaria** | **Discreta** | **Continua** |
| State | AppYear | DifState | Term | GrDisburs |
| BankState | AppMonth | Secured | NoEmp | GrApprov |
| Sector |  | NewExist | SecuredSBA | ApprovSBA |
|  |  | Urban |  |  |
|  |  | Rural |  |  |
|  |  | RevLine |  |  |
|  |  | LowDoc |  |  |
|  |  | **Default** |  |  |

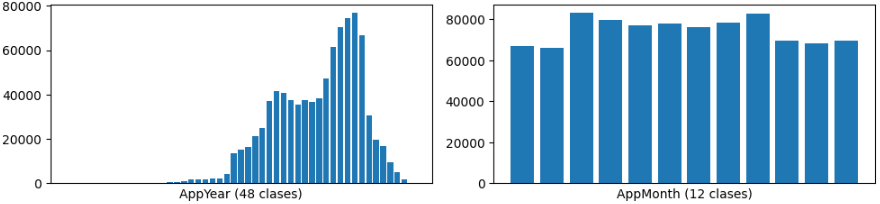
Visualización de Variables Preprocesadas

Finalmente, procederemos a explorar visualmente todas las variables obtenidas del preprocesamiento de datos para comprender mejor su distribución, relaciones entre características y cualquier patrón emergente, esta visualización nos ayudará a tener una perspectiva más profunda del conjunto de datos resultante antes de comenzar con la experimentación de los algoritmos de Machine Learning y Deep Learning.

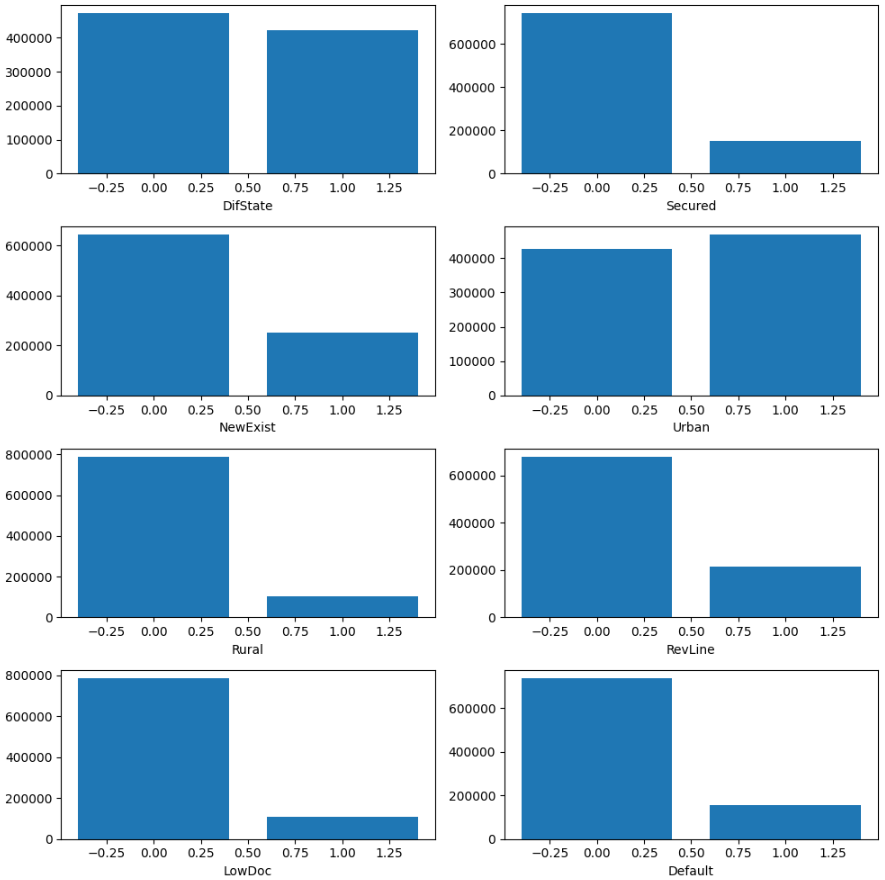
**Figura 7***Distribución de las Variables Categóricas Nominales.*



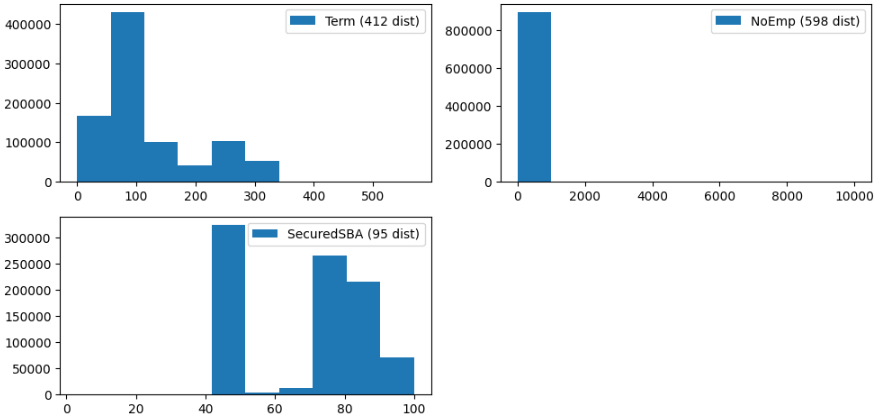
**Figura 8***Distribución de las Variables Categóricas Ordinales.*



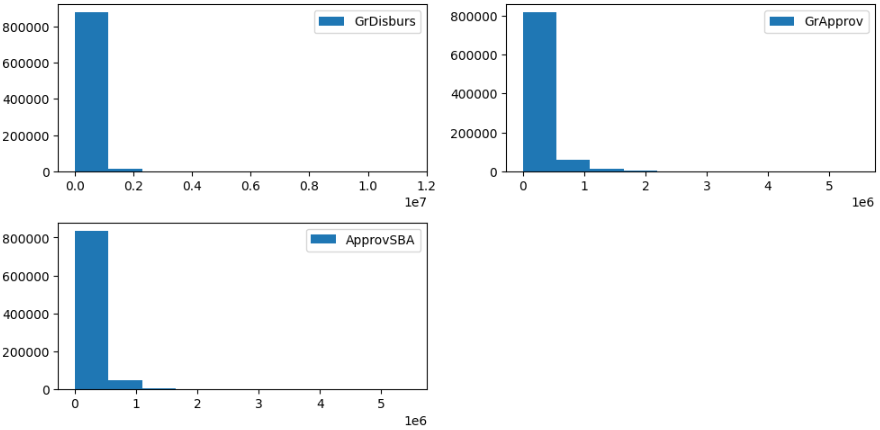
**Figura 9***Distribución de las Variables Categóricas Binarias.*



**Figura 10***Distribución de las Variables Numéricas Discretas.*



**Figura 11***Distribución de las Variables Numéricas Continuas.*



7.2.8 Partición de Datos

La partición de datos es un paso esencial en el proceso de entrenamiento y evaluación de modelos de ML, consiste en dividir el dataset preprocesado en conjuntos más pequeños con diferentes propósitos, en tal sentido se ha realizado las siguientes particiones:

**Conjunto de Entrenamiento (Training)**

Este conjunto se utiliza para entrenar el modelo de ML, vale decir que el modelo se ajusta a estos datos durante el proceso de entrenamiento; para nuestro caso de estudio se ha reservado el 70% del conjunto de los datos para entrenamiento (629414 instancias).

**Conjunto de Validación (Val)**

Este conjunto se utiliza para ajustar hiperparámetros y realizar ajustes adicionales en el modelo, ayuda a evitar el sobreajuste y permite la selección de modelos; para tal propósito, se ha reservado el 15% del conjunto de los datos para validación (134875 instancias).

**Conjunto de Prueba (Test)**

Este conjunto se utiliza para evaluar el rendimiento del modelo después de que se ha entrenado y validado, proporciona una medida imparcial del rendimiento del modelo en datos no vistos; se ha reservado 15% de los datos para las pruebas (134875 instancias).

Adicionalmente, utilizamos el conjunto de datos de entrenamiento para generar datos balanceados, que pueden ser importantes para mejorar el rendimiento del modelo, reducir el sesgo, mejorar la generalización, mitigar el sobre ajuste y mejorar la interpretabilidad; generamos un conjunto de datos aplicando submuestreo (219654 instancias) y otro aplicando sobremuestreo utilizando la técnica de SMOTE (1033686 instancias).

La idea principal de **SMOTE** (Técnica de Sobremuestreo Sintético de la Clase Minoritaria) es crear nuevas instancias que estén "cerca" de las instancias de la clase minoritaria existentes, en lugar de simplemente duplicar ejemplos existentes; realiza estos pasos:

1. Selecciona una instancia de la clase minoritaria existente.
2. Elige uno o más vecinos cercanos a esta instancia, los vecinos se seleccionan de manera que estén dentro de un rango definido.
3. Genera instancias sintéticas entre la instancia seleccionada y sus vecinos, estas nuevas instancias se crean mediante la interpolación de las características de las instancias originales y se añaden al conjunto de datos.

7.3 Modelos de ML para Clasificación

Los modelos de Machine Learning para clasificación son algoritmos diseñados para predecir la clase o categoría a la que pertenece una instancia de datos en función de sus características, la clasificación es parte del aprendizaje supervisado, lo que significa que estos modelos se entrenan en datos etiquetados, donde la etiqueta o clase de cada instancia de datos es conocida previamente, el objetivo es que un modelo aprenda a asignar automáticamente una etiqueta a nuevas instancias de datos no etiquetados.

Los modelos de ML para clasificación se utilizan en una amplia gama de aplicaciones, como detección de diversos fraudes bancarios, diagnóstico médico, clasificación de imágenes, recomendación de productos y demás. La elección del modelo depende de la naturaleza de los datos y los objetivos específicos del problema de clasificación, cada modelo tiene sus propias ventajas y desventajas, y se selecciona en función al problema en cuestión.

7.3.1 Métricas de Evaluación

En problemas de clasificación, existen diversas métricas de evaluación que permiten medir el rendimiento y la calidad de un modelo de Machine Learning, estas métricas proporcionan información sobre cómo el modelo se comporta en términos de precisión y otros aspectos.

**Exactitud (Accuracy)**: Mide la proporción de predicciones correctas sobre el total de predicciones, es muy utilizada, pero puede ser engañosa en datos desequilibrados.

**Precisión (Precision)**: Se refiere a la proporción de predicciones positivas correctas sobre el total de predicciones positivas, es útil cuando se quiere minimizar los falsos positivos.

**Sensibilidad (Recall)**: Mide la proporción de verdaderos positivos sobre el total de muestras verdaderamente positivas, es útil cuando se quiere minimizar los falsos negativos.

**F1-Score**: Es una métrica que combina precision y recall en una sola puntuación, es útil cuando se desea encontrar un equilibrio entre la precision y el recall.

**Matriz de Confusión (Confusion Matrix)**: Es una tabla que muestra la cantidad de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN), es bastante útil para comprender el rendimiento del modelo en detalle.

**Área bajo la Curva ROC (AUC)**: La curva ROC es una representación gráfica del rendimiento del modelo en función de la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR). El AUC mide la capacidad del modelo para distinguir entre clases.

7.3.2 Logistic Regression

La Regresión Logística es un algoritmo de ML utilizado para la clasificación de datos en dos o más categorías, a pesar de su nombre, no se utiliza para problemas de regresión (predicción de valores numéricos), sino para estimar la probabilidad de que una instancia de datos pertenezca a una de las clases disponibles; funciona de la siguiente manera:

1. **Probabilidad de Clase**: Para realizar una clasificación binaria, la regresión logística estima la probabilidad de que una instancia de datos pertenezca a la clase positiva.
2. **Función Logística (Sigmoide)**: Para modelar la probabilidad, la regresión logística utiliza la función sigmoide. La función sigmoide transforma un valor entre 0 y 1, que se interpreta como la probabilidad de pertenecer a la clase positiva.
3. **Entrenamiento**: En el entrenamiento, se ajustan los coeficientes (pesos) del modelo para minimizar la diferencia entre las probabilidades estimadas y las etiquetas reales del conjunto de entrenamiento; esto se hace utilizando una función de costo, como la entropía cruzada (cross-entropy) o el logaritmo negativo de la verosimilitud.
4. **Umbral de Decisión**: Se utiliza un umbral de decisión (suele ser 0.5 en clasificación binaria) para tomar una decisión sobre la clase predicha: si es mayor o igual al umbral, se predice la clase positiva, de lo contrario, se predice la clase negativa.
5. **Clasificación Multiclase**: La regresión logística se puede extender para problemas de clasificación multiclase utilizando técnicas como “One-vs-All” o "Softmax".

La regresión logística es un modelo lineal (asume que la relación entre características y la probabilidad de clasificación es aproximadamente lineal), a pesar de su simplicidad, es muy utilizada dada su interpretabilidad y eficacia en una variedad de problemas de clasificación; sin embargo, en problemas no lineales, pueden ser necesarios algoritmos más avanzados.

Entre las **ventajas** de la regresión logística podemos mencionar: su interpretabilidad (produce coeficientes para cada característica que indican el impacto de esa característica en la probabilidad de pertenecer a una clase en particular), tiene buen rendimiento en clasificación binaria cuando las clases son linealmente separables, es un modelo computacionalmente eficiente y rápido de entrenar y aplicar, funciona bien en conjuntos de datos de tamaño moderado a grande, tiende a tener una baja probabilidad de sobreajuste.

Algunas **desventajas** de la regresión logística serian: no es adecuada para problemas altamente no lineales sin transformaciones previas de características, cuando hay un desequilibrio significativo entre las clases puede producir resultados sesgados hacia la clase mayoritaria, en datasets con muchas características, puede no funcionar tan eficazmente y puede sufrir problemas de multicolinealidad y puede ser sensible a valores atípicos.

Se ha aplicado estandarización al conjunto de datos de entrenamiento, considerando que, la regresión logística, funcionan mejor con datos estandarizados, ya que asumen una distribución normal; en ese sentido, se han estandarizado todas a las variables numéricas: Term, NoEmp, SecuredSBA, GrDisburs, GrApprov y ApprovSBA.

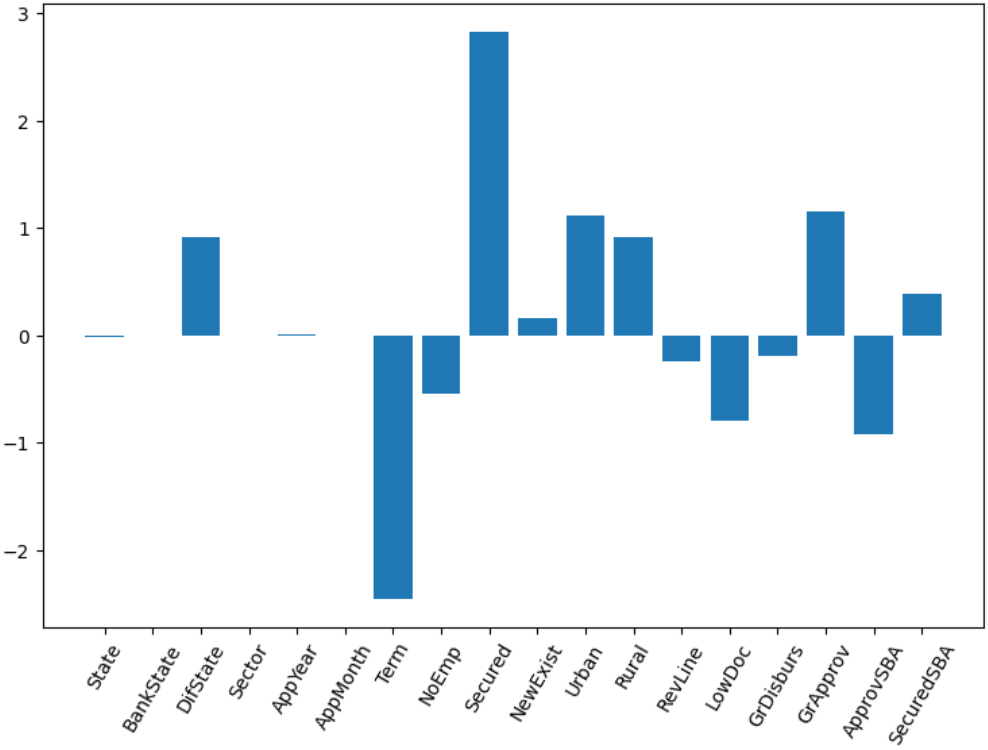
Para realizar el **ajuste del modelo** de regresión logística utilizamos GridSearch, que es una estrategia eficaz y eficiente para encontrar la configuración de hiperparámetros óptima y mejorar el rendimiento del modelo, los hiperparametros óptimos se detallan en la Tabla 23, se ha conseguido un accuracy de **0.8556** utilizando los hiperparametros óptimos.

**Tabla 23**  
Hiperparametros óptimos de la Regresión Logística.

|  |  |  |
| --- | --- | --- |
| **Hiperparametro** | **Valor** | **Detalle** |
| c | 1 | Inversa de la fuerza de regularización |
| penalty | l2 | Especifica la norma de la penalización |

La interpretación de la **Importancia de las Características** en la regresión logística refleja cómo cada variable contribuye a la predicción de la probabilidad de pertenecer a una clase en particular, el signo indica la dirección de la relación, como se observa en la Figura 12.

**Figura 12***Importancia de Características de Regresión Logística.*



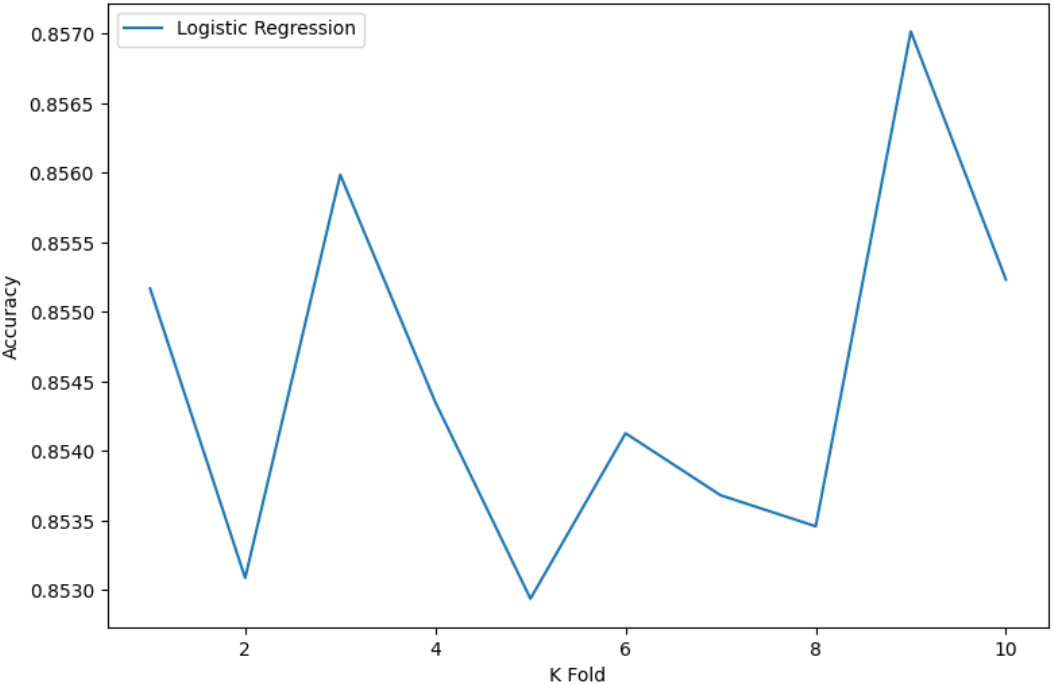
Se ha aplicado la Eliminación de Características Recursivas (REF), que es una técnica para mejorar el rendimiento de un modelo al identificar y eliminar sistemáticamente las variables menos importantes; seleccionamos 9 variables que dieron un accuracy de 0.8479.

Se ha experimentado con un Umbral Personalizado, modificando el umbral de decisión establecido de 0.5, consiguiendo que con un umbral de 0.4 se logre mejorar el accuracy obteniendo 0.8610, que llega a ser el mejor resultado obtenido en este modelo.

Se ha testeado el modelo con los conjuntos de datos balanceados, obteniendo accuracy de 0.1774 y 0.1771 en los datos de submuestreo y sobremuestreo respectivamente.

La Validación Cruzada, es una técnica fundamental en ML que se utiliza para evaluar el rendimiento de un modelo predictivo, su objetivo principal es estimar la capacidad de generalización del modelo, los resultados obtenidos se encuentran en la Figura 13.

**Figura 13***Validación Cruzada de la Regresión Logística.*



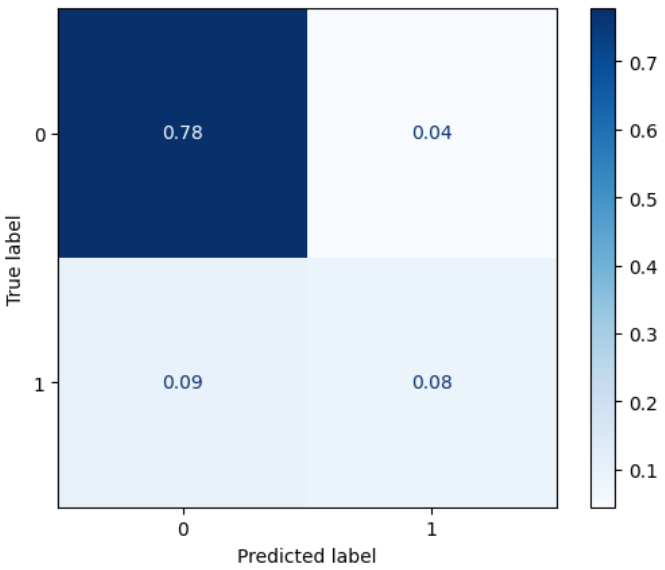
Las mejores Métricas de Evaluación obtenidas aplicando los hiperparametros óptimos encontrados y utilizando el umbral personalizado se detallan en la Tabla 24.

**Tabla 24**  
Métricas de Evaluación de la Regresión Logística.

|  |  |  |  |
| --- | --- | --- | --- |
| **Logistic Regression** | | | |
| Accuracy | Precision | Recall | F1-score |
| 0.8612 | 0.7716 | 0.7076 | 0.7318 |

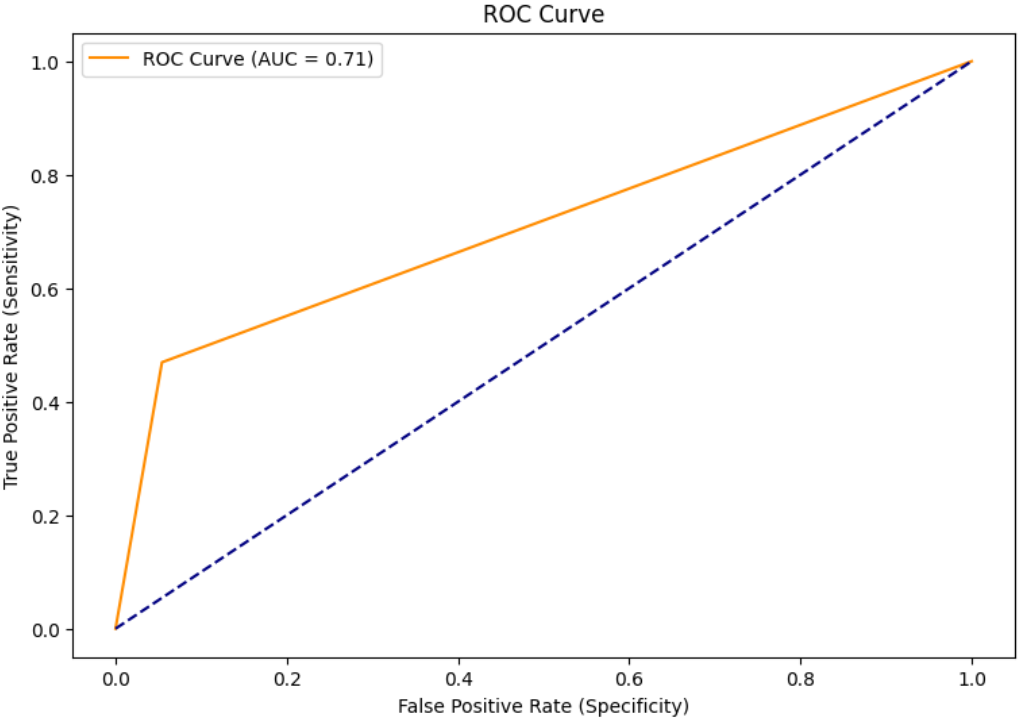
La Matriz de Confusión resultante del modelo optimizado se muestra en la Figura 14.

**Figura 14***Matriz de Confusion de la Regresión Logística.*



El Área bajo la curva ROC (AUC) del modelo optimizado se encuentra en la Figura 15.

**Figura 15***Curva ROC y AUC de la Regresión Logística.*



7.3.3 K-Nearest Neighbors

K-Nearest Neighbors (K-NN) es un algoritmo de ML utilizado para clasificación y regresión, es basado en instancias, lo que significa que no crea un modelo explícito durante el entrenamiento, sino que realiza predicciones basadas en la similitud de las instancias de datos de entrenamiento con el punto de datos a predecir; funciona de la siguiente manera:

1. Elección de k: Se selecciona un valor para k, que representa el número de vecinos más cercanos que se utilizarán para tomar una decisión de clasificación o regresión.
2. Medición de la Similitud: Para cada punto de datos de prueba, se calcula la distancia entre ese punto y todos los puntos de datos de entrenamiento, las medidas de distancia comunes incluyen la distancia Euclidiana o la distancia de Manhattan.
3. Selección de Vecinos: Se seleccionan los k puntos de entrenamiento más cercanos al punto de prueba en función de la distancia calculada.
4. Clasificación o Regresión: En el caso de clasificación, se realiza un conteo de las clases de los k vecinos más cercanos y se asigna al punto de prueba la clase más común entre esos vecinos.; en regresión, se promedian los valores de los k vecinos más cercanos y se asigna ese valor al punto de prueba.

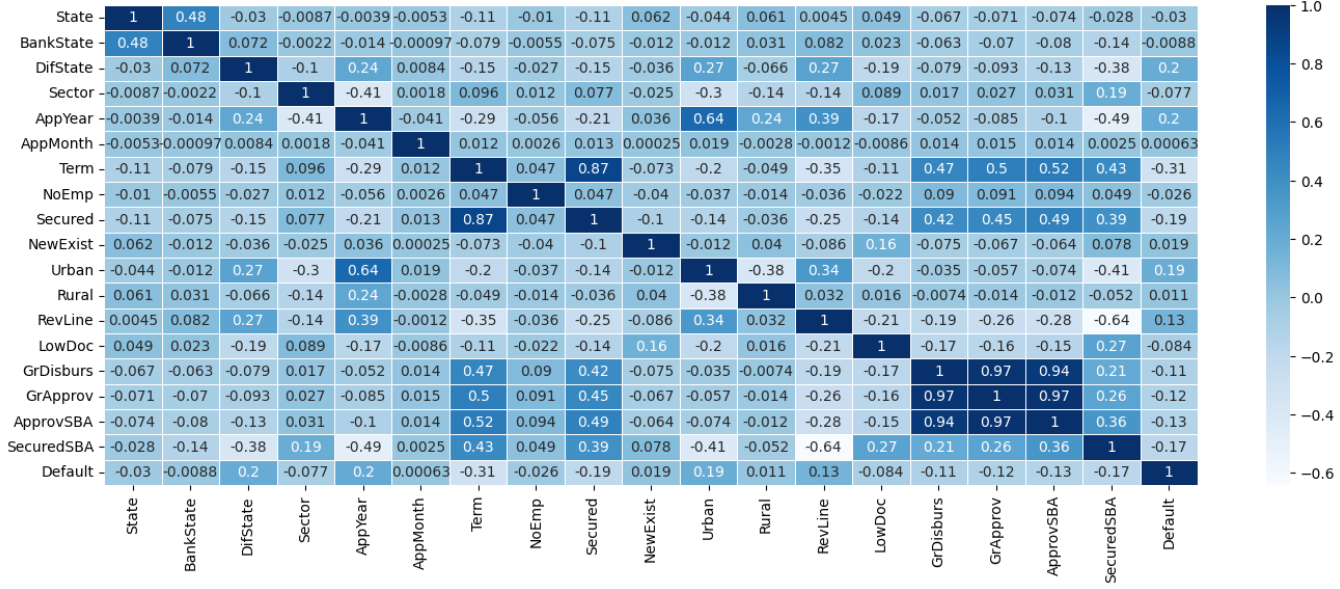
Las ventajas que tiene K-NN son: es un algoritmo simple de entender y de implementar, se adapta bien a datos no lineales, puede manejar problemas de clasificación y regresión, no hace suposiciones sobre la distribución de los datos, lo que lo hace robusto en diferentes situaciones y sus predicciones pueden ser fácilmente explicadas y comprendidas.

Las desventajas que tiene K-NN son: es sensible a la escala de las variables, por lo que es importante estandarizar o normalizar los datos antes de usarlo, la elección de k puede afectar a su rendimiento (un valor incorrecto de k puede llevar a subajuste o sobreajuste). La elección de k y el preprocesamiento de datos son factores críticos para su éxito.

Se ha aplicado normalización al conjunto de datos de entrenamiento, considerando que, algoritmos sensibles a la escala, como K-NN y Redes Neuronales Artificiales que tienen funciones de activación basadas en distancias (como la función Sigmoide), pueden beneficiarse de la normalización; en tal sentido, se han normalizado las variables numéricas: Term, NoEmp, SecuredSBA, GrDisburs, GrApprov y ApprovSBA.

La Matriz de Correlación es una tabla o matriz que muestra los coeficientes de correlación entre múltiples pares de variables. El Coeficiente de Correlación de Pearson mide la relación lineal entre dos variables numéricas, puede tomar valores entre -1 (correlación negativa perfecta) y 1 (correlación positiva perfecta), un valor cercano a 0 indica una correlación débil; en la Figura 16 mostramos la correlación de las variables del dataset.

**Figura 16***Matriz de Correlación de las variables.*



El Factor de Inflación de la Varianza (VIF) proporciona un índice que mide hasta qué punto la varianza de un coeficiente se incrementa a causa de la colinealidad (VIF = 1 / (1 - R2)).

* VIF = 1: Las variables no están correlacionadas.
* VIF < 5: Las variables tienen una correlación moderada, se pueden quedar en el modelo.
* VIF > 5: Las variables están altamente correlacionadas, deben desaparecer del modelo.

Aplicamos el Factor de Inflación de la Varianza (VIF) a las variables predictoras del conjunto de datos, se obtuvieron índices menores a 5, excepto las variables GrDisburs, GrApprov y ApprovSBA que tenían índices de 20.1, 51.5 y 29.4 respectivamente; considerando lo elevado de los índices estas variables fueron eliminadas para entrenar este modelo.

Para realizar el **ajuste del modelo** de K-NN utilizamos GridSearch, para encontrar la configuración de hiperparámetros óptimos, los resultados se detallan en la Tabla 25, se ha conseguido un accuracy de **0.8620** utilizando los hiperparametros encontrados.

**Tabla 25**  
Hiperparametros óptimos de K-NN.

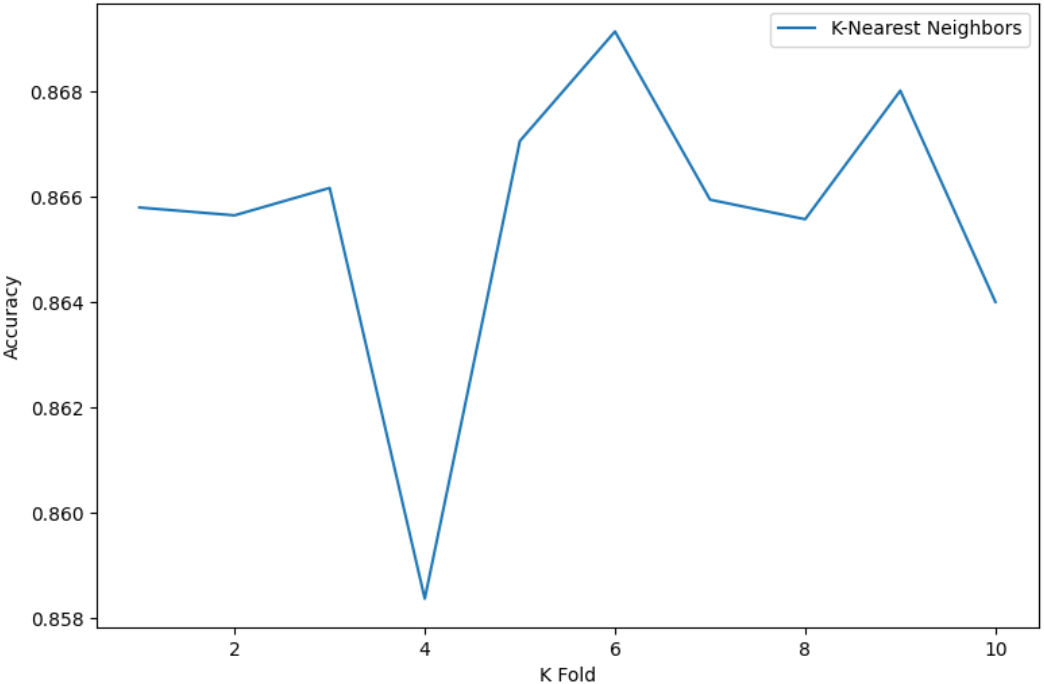
|  |  |  |
| --- | --- | --- |
| **Hiperparametro** | **Valor** | **Detalle** |
| metric | manhattan | Métrica que se utilizará para calcular la distancia |
| n\_neighbors | 10 | Número de vecinos que se utilizarán |
| weights | uniform | Función de peso utilizada en la predicción |

Para la Selección de Características del modelo de K-NN utilizaremos la prueba Chi-Cuadrado (prueba estadística utilizada para determinar si existe una asociación significativa entre dos variables categóricas en una tabla de contingencia), que implica la selección de un subconjunto de variables más informativas para mejorar el rendimiento; las variables elegidas fueron: State, DifState, Sector, AppYear, Term, Secured, Urban, RevLine, LowDoc y SecuredSBA, logrando un accuracy de 0.8809, el mejor resultado del modelo.

Se ha testeado el modelo con los conjuntos de datos balanceados, obteniendo accuracy de 0.2733 y 0.4963 en los datos de submuestreo y sobremuestreo respectivamente.

Los resultados obtenidos en la Validación Cruzada se encuentran en la Figura 17.

**Figura 17***Validación Cruzada de K-NN.*



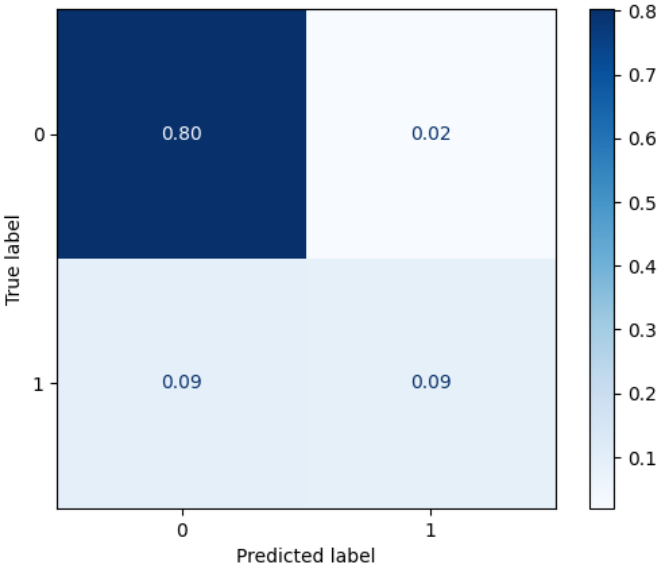
Las Métricas de Evaluación (detalladas en la Tabla 26) fueron obtenidas aplicando los hiperparametros óptimos encontrados, eliminando variables con índice VIF muy elevado y considerando únicamente las variables elegidas en la selección de características, entrenando el modelo con los conjuntos de datos de entrenamiento y validación.

**Tabla 26**  
Métricas de Evaluación de K-NN.

|  |  |  |  |
| --- | --- | --- | --- |
| **K-Nearest Neighbors** | | | |
| Accuracy | Precision | Recall | F1-score |
| 0.8896 | 0.8561 | 0.7329 | 0.7737 |

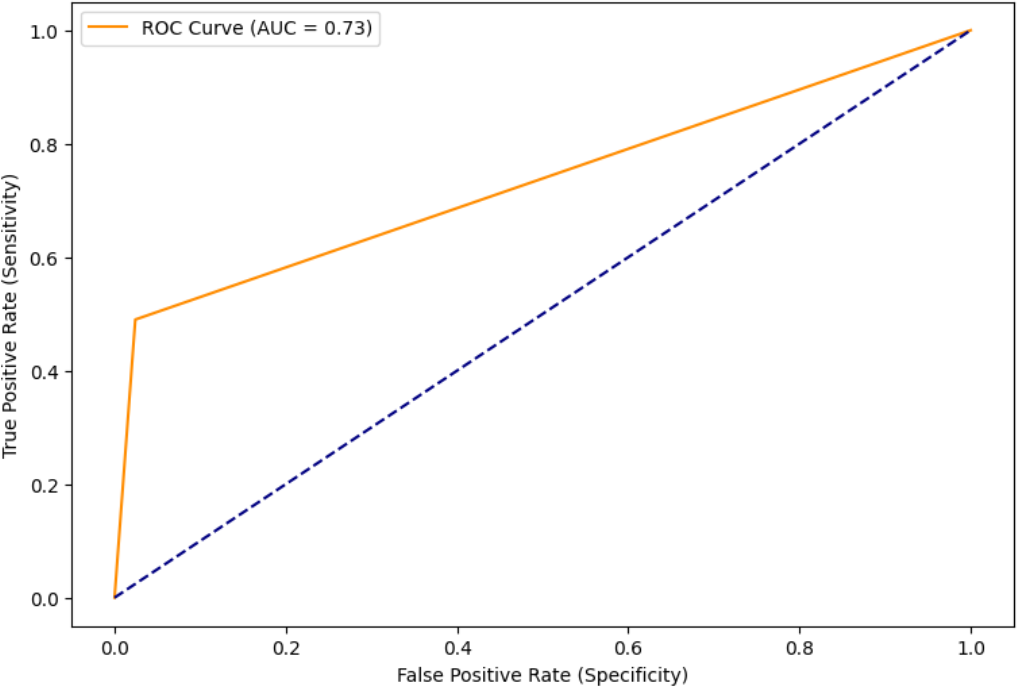
La Matriz de Confusión resultante del modelo optimizado se muestra en la Figura 18.

**Figura 18***Matriz de Confusion de K-NN.*



El Área bajo la curva ROC (AUC) del modelo optimizado se encuentra en la Figura 19.

**Figura 19***Curva ROC y AUC de K-NN.*



7.3.4 Decision Tree Classifier

Un Decision Tree Classifier (Clasificador de Árbol de Decisión) es un algoritmo de ML supervisado utilizado para la clasificación de datos, funciona creando un árbol de decisiones que divide los datos en grupos más homogéneos, lo que permite tomar decisiones basadas en reglas si-entonces; su funcionamiento es de la siguiente manera:

1. División de los Datos: Comienza dividiendo el conjunto de datos en dos o más subconjuntos, cada división se basa en una característica que se considera la más relevante en ese punto para separar las clases objetivo de manera más efectiva.
2. Selección de la Característica Óptima: Busca la característica que proporciona la mejor división de los datos, generalmente utilizando métricas como la ganancia de información, la impureza de Gini o el error cuadrático medio.
3. Creación de Nodos: Se crea un nodo en el árbol para representar la división, cada nodo representa una decisión basada en una característica.
4. Recursión: El proceso de división y creación de nodos se repite en cada subconjunto resultante, formando una estructura de árbol jerárquica, este proceso continúa hasta que se cumplan ciertos criterios de parada, como una profundidad máxima del árbol o un número mínimo de muestras en los nodos hoja.
5. Nodos Hoja: Los nodos hoja son los nodos terminales del árbol y representan las etiquetas de clase finales, cada nodo hoja puede estar asociado con una clase.

Las ventajas que tiene Decision Tree Classifier son: es fácilmente interpretable y se puede visualizar, las decisiones se toman de manera lógica y se pueden explicar claramente, puede manejar datos tanto numéricos como categóricos sin necesidad de preprocesamiento exhaustivo, son rápidos en la etapa de predicción, ya que involucran una serie de decisiones sencillas y ayudan a identificar las características más importantes en el conjunto de datos.

Las desventajas que tiene Decision Tree Classifier son: tienden a sobreajustarse a los datos de entrenamiento especialmente si se crean árboles profundos, pequeños cambios en los datos de entrenamiento pueden llevar a árboles de decisión significativamente diferentes, pueden ser ineficientes en la clasificación de datos con clases desequilibradas, pueden no encontrar siempre la mejor solución global, ya que toman decisiones locales en cada nodo y pueden no funcionar tan bien en problemas de regresión como en clasificación.

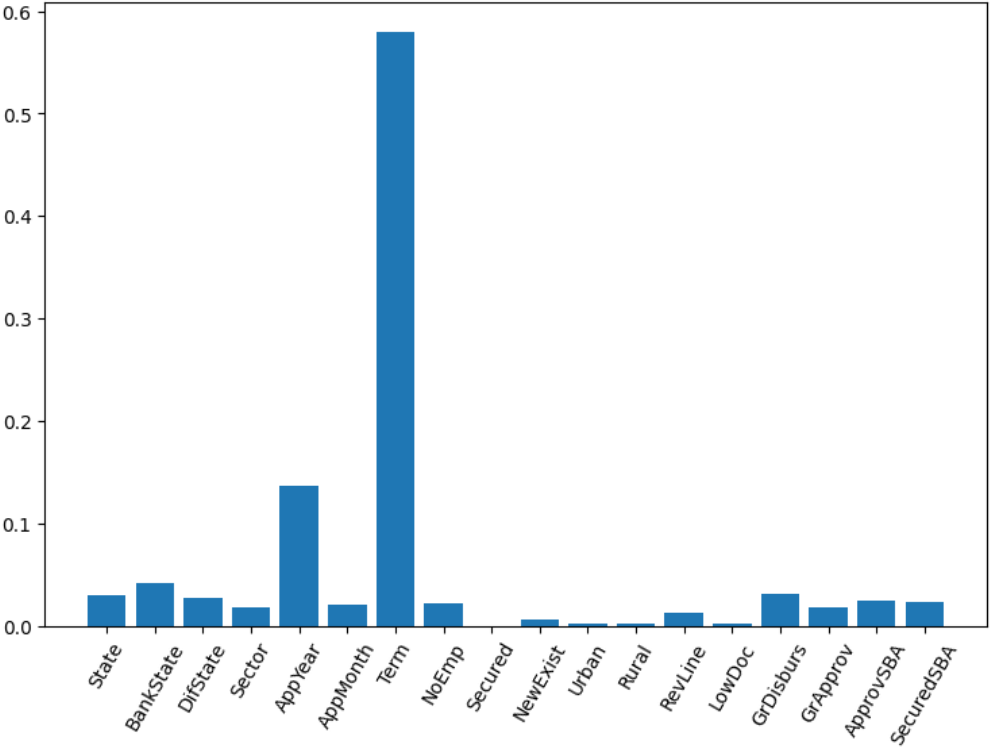
Para realizar el **ajuste del modelo** de Árbol de Decision utilizamos GridSearch, para encontrar la configuración de hiperparámetros óptimos, los resultados se detallan en la Tabla 27, se ha logrado un accuracy de **0.9376** utilizando los hiperparametros encontrados.

**Tabla 27**  
Hiperparametros óptimos del Árbol de Decision.

|  |  |  |
| --- | --- | --- |
| **Hiperparametro** | **Valor** | **Detalle** |
| criterion | entropy | La función para medir la calidad de una división |
| max\_depth | 20 | La profundidad máxima del árbol |

La Importancia de las Características en el Árbol de Decision refleja cómo cada variable contribuye a la predicción de pertenencia de cada clase, como se observa en la Figura 20.

**Figura 20***Importancia de Características del Árbol de Decision.*

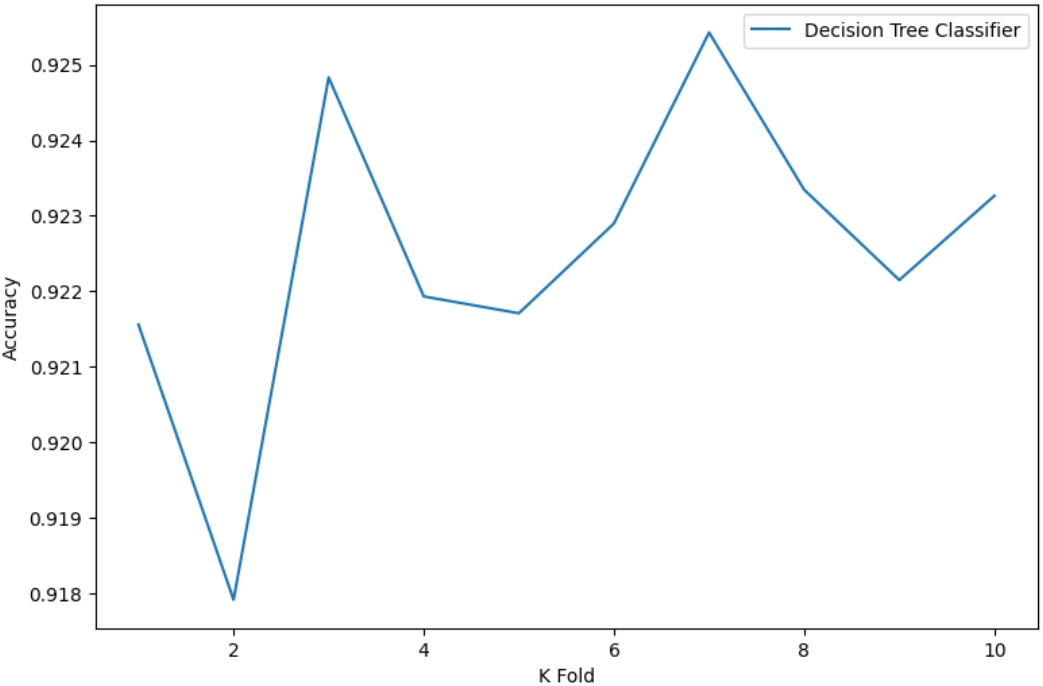


Se ha aplicado la Eliminación de Características Recursivas (RFE), que es una técnica para mejorar el rendimiento de un modelo al identificar y eliminar sistemáticamente las variables menos importantes; las variables seleccionadas fueron: State, BankState, DifState, AppYear, Term, NoEmp, GrDisburs, ApprovSBA y SecuredSBA, logrando un accuracy de 0.9379, siendo el mejor resultado obtenido por el modelo.

Se ha testeado el modelo con los conjuntos de datos balanceados, obteniendo accuracy de 0.8904 y 0.8926 en los datos de submuestreo y sobremuestreo respectivamente.

Los resultados obtenidos en la Validación Cruzada se encuentran en la Figura 21.

**Figura 21***Validación Cruzada del Árbol de Decision.*



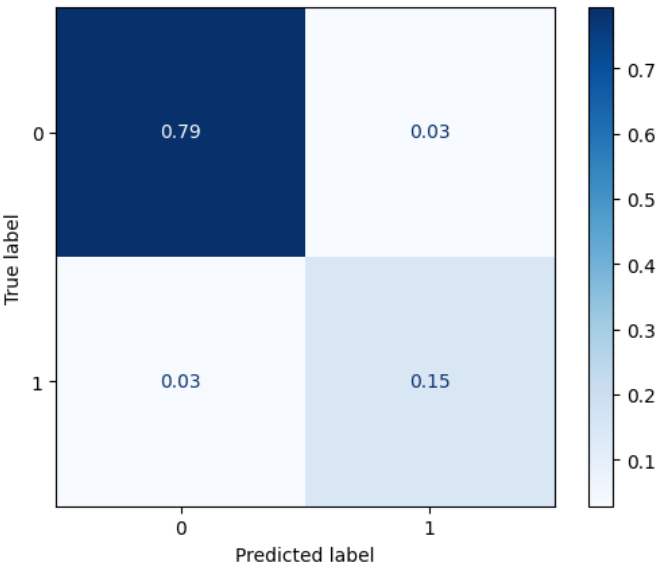
Las Métricas de Evaluación (detalladas en la Tabla 28) fueron obtenidas aplicando los hiperparametros óptimos encontrados, considerando únicamente las variables elegidas en RFE, entrenando el modelo con los conjuntos de datos de entrenamiento y validación.

**Tabla 28**  
Métricas de Evaluación del Árbol de Decision.

|  |  |  |  |
| --- | --- | --- | --- |
| **Decision Tree Classifier** | | | |
| Accuracy | Precision | Recall | F1-score |
| 0.9389 | 0.8979 | 0.8912 | 0.8945 |

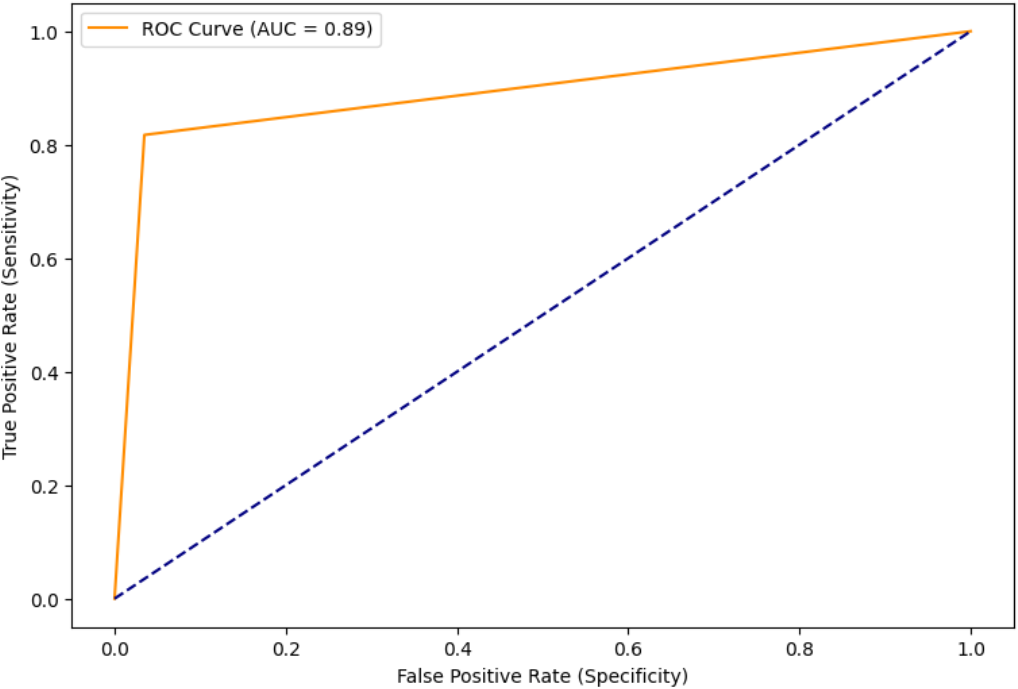
La Matriz de Confusión resultante del modelo optimizado se muestra en la Figura 22.

**Figura 22***Matriz de Confusion del Árbol de Decision.*



El Área bajo la curva ROC (AUC) del modelo optimizado se encuentra en la Figura 23.

**Figura 23***Curva ROC y AUC del Árbol de Decision.*



7.3.5 Random Forest Classifier

Random Forest Classifier es un algoritmo de ML que se basa en la construcción de múltiples árboles de decisión (un bosque) y es ampliamente utilizado para tareas de clasificación, es un poderoso algoritmo que combina múltiples árboles de decisión para mejorar la precisión y la generalización, es adecuado para una amplia gama de aplicaciones y se considera uno de los enfoques más sólidos en ML; su funcionamiento es de la siguiente manera:

1. Creación de Múltiples Árboles: Inicia creando un conjunto de árboles de decisión, cada árbol se entrena con una porción aleatoria de datos de entrenamiento, esto se conoce como "bootstrap sampling" (muestreo por reemplazo), lo que significa que se seleccionan muestras aleatorias del conjunto de entrenamiento con reemplazo.
2. Selección de Características Aleatorias: En cada división de un árbol, el algoritmo no considera todas las características disponibles, en cambio, se selecciona un subconjunto aleatorio de características para la división, esto se conoce como "feature bagging" (muestreo de características).
3. Votación Mayoritaria: Una vez que se han construido todos los árboles, cuando se realiza una predicción para una nueva instancia, cada árbol en el bosque emite una predicción, luego, se realiza una votación mayoritaria para determinar la clase final de la instancia, la clase que obtiene la mayoría de votos se toma como la predicción.

Las ventajas que tiene Random Forest Classifier son: tiene la capacidad de generalizar bien a partir de los datos de entrenamiento, lo que reduce el riesgo de sobreajuste en comparación con árboles de decisión individuales, tiende a producir predicciones precisas y robustas en una variedad de conjuntos de datos y problemas, puede manejar conjuntos de datos con un gran número de características y variables categóricas, sin necesidad de mucho preprocesamiento, proporciona una medida de la importancia de las características, lo que ayuda a identificar cuáles características son más informativas para la clasificación y tiene la capacidad de manejar datos ruidosos y valores atípicos.

Las desventajas que tiene Random Forest Classifier son: construir múltiples árboles y combinar sus resultados puede hacer que Random Forest sea más lento en el entrenamiento que un solo árbol de decisión, aunque Random Forest es más interpretable que algunos algoritmos de ML, como redes neuronales, todavía puede ser menos interpretable que un solo árbol de decisión y requiere la afinación de hiperparámetros, como el número de árboles y la profundidad máxima, para lograr el mejor rendimiento.

La idea del bagging en el contexto de Random Forest, es que, al utilizar muestras aleatorias en la construcción de cada modelo, se reduce la varianza y el riesgo de sobreajuste, al promediar o votar entre múltiples modelos, se obtiene una predicción más robusta y precisa, esto es especialmente útil cuando se trabaja con datos ruidosos o propensos al sobreajuste.

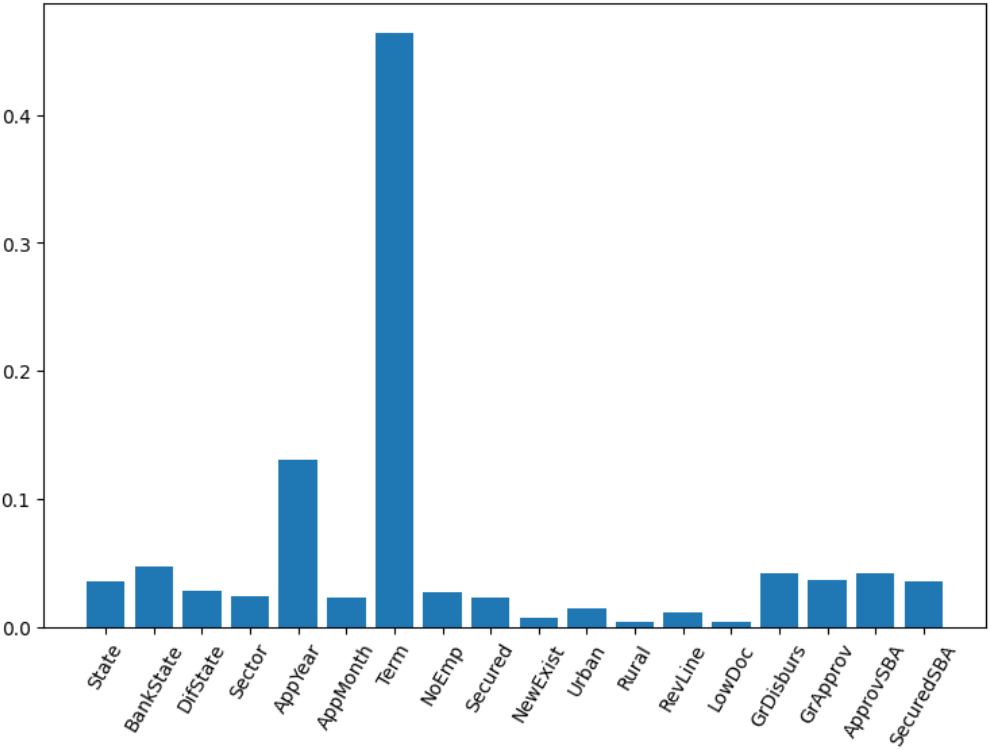
Para realizar el **ajuste del modelo** de Random Forest utilizamos GridSearch, para encontrar la configuración de hiperparámetros óptimos, los resultados se detallan en la Tabla 29, se ha conseguido un accuracy de **0.9479** utilizando los hiperparametros encontrados.

**Tabla 29**  
Hiperparametros óptimos de Random Forest.

|  |  |  |
| --- | --- | --- |
| **Hiperparametro** | **Valor** | **Detalle** |
| criterion | entropy | La función para medir la calidad de una división |
| max\_depth | 20 | La profundidad máxima del árbol |

La Importancia de las Características en Random Forest refleja cómo cada variable contribuye a la predicción de pertenencia de cada clase, como se observa en la Figura 24.

**Figura 24***Importancia de Características de Random Forest.*

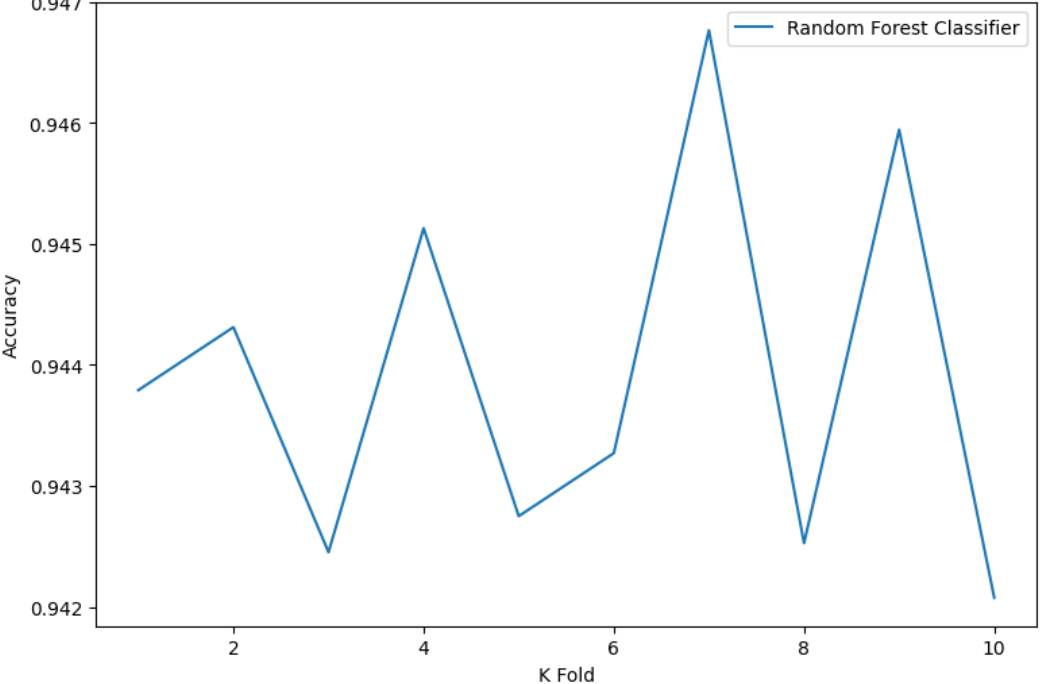


Se ha aplicado la Eliminación de Características Recursivas (RFE), que es una técnica para mejorar el rendimiento de un modelo al identificar y eliminar sistemáticamente las variables menos importantes; las variables seleccionadas fueron: State, BankState, Sector, AppYear, Term, GrDisburs, GrApprov, ApprovSBA y SecuredSBA, logrando un accuracy de 0.9493, siendo el mejor resultado obtenido por el modelo.

Se ha testeado el modelo con los conjuntos de datos balanceados, obteniendo accuracy de 0.9093 y 0.9149 en los datos de submuestreo y sobremuestreo respectivamente.

Los resultados obtenidos en la Validación Cruzada se encuentran en la Figura 25.

**Figura 25***Validación Cruzada de Random Forest.*



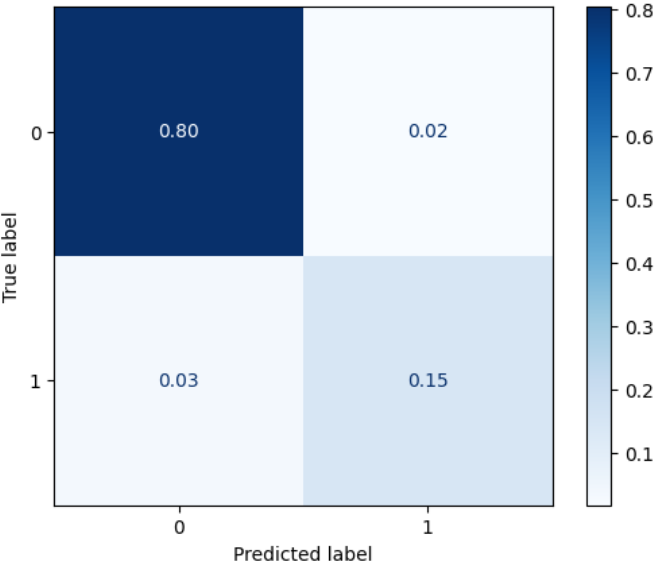
Las Métricas de Evaluación (detalladas en la Tabla 30) fueron obtenidas aplicando los hiperparametros óptimos encontrados, considerando únicamente las variables elegidas en RFE, entrenando el modelo con los conjuntos de datos de entrenamiento y validación.

**Tabla 30**  
Métricas de Evaluación de Random Forest.

|  |  |  |  |
| --- | --- | --- | --- |
| **Random Forest Classifier** | | | |
| Accuracy | Precision | Recall | F1-score |
| 0.9494 | 0.9241 | 0.8987 | 0.9107 |

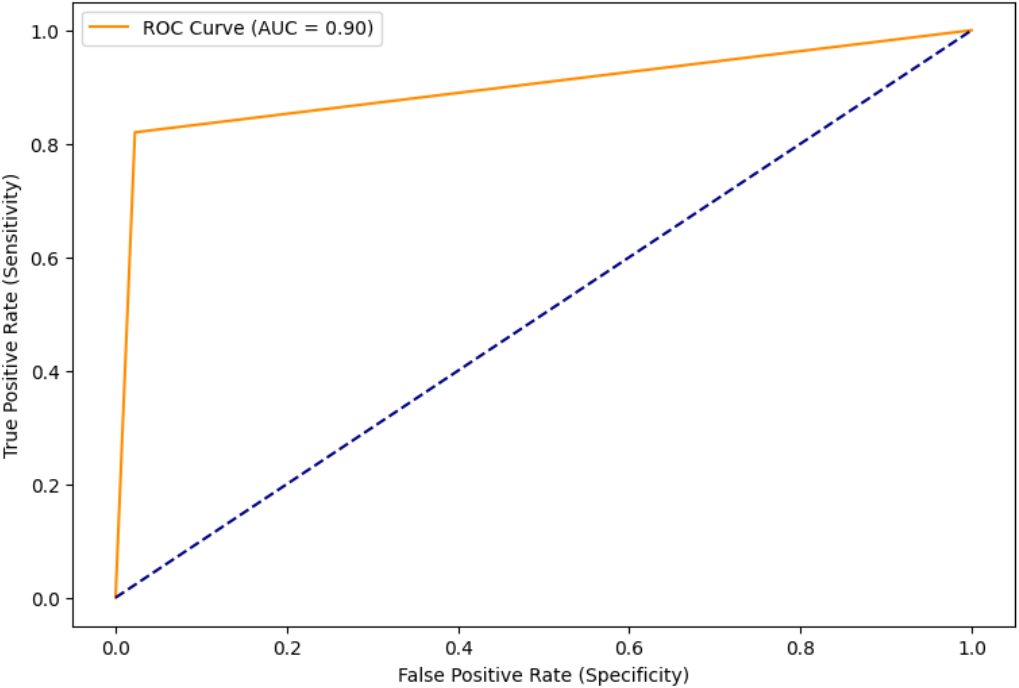
La Matriz de Confusión resultante del modelo optimizado se muestra en la Figura 26.

**Figura 26***Matriz de Confusion de Random Forest.*



El Área bajo la curva ROC (AUC) del modelo optimizado se encuentra en la Figura 27.

**Figura 27***Curva ROC y AUC de Random Forest.*



7.3.6 XGBoost

XGBoost (Extreme Gradient Boosting) es un algoritmo de ML que se basa en la técnica de boosting (esta técnica se basa en la idea de construir múltiples modelos débiles y combinar sus predicciones para formar un modelo más fuerte) para mejorar la precisión en tareas de clasificación y regresión, es ampliamente utilizado y se destaca por su eficacia y rendimiento; su funcionamiento es de la siguiente manera:

1. Construcción de Árboles Débiles: Crea un conjunto de árboles de decisión, pero a diferencia de Random Forest, que construye árboles en paralelo, XGBoost los construye secuencialmente; comienza con un árbol débil y luego ajusta árboles adicionales para corregir los errores de predicción del modelo anterior.
2. Ponderación de Muestras: Durante el entrenamiento de cada árbol, se asigna un peso a cada muestra del conjunto de datos de entrenamiento, las muestras que se predijeron incorrectamente por los árboles anteriores tienen un mayor peso, lo que las hace más importantes en el siguiente árbol.
3. Cálculo de Errores Residuales: Calcula los errores residuales en función de las predicciones actuales y los verdaderos valores objetivos; luego, entrena un nuevo árbol para ajustar estos errores residuales.
4. Regularización: Utiliza técnicas de regularización para evitar el sobreajuste, incluyendo la poda de árboles, la penalización de la complejidad de los árboles y la limitación de la profundidad de los árboles.
5. Combinación de Modelos: Las predicciones de todos los árboles se combinan al final mediante una suma ponderada, esto da como resultado la predicción final.

Las ventajas que tiene XGBoost son: es altamente eficiente y rápido (es conocido por su velocidad de entrenamiento y predicción, lo que lo hace adecuado para conjuntos de datos grandes), tiende a lograr un alto rendimiento en términos de precisión en comparación con otros algoritmos de ML, las técnicas de regularización incorporadas ayudan a evitar el sobreajuste y a crear modelos más generales, tiene la capacidad de manejar automáticamente valores faltantes en los datos y proporciona información sobre la importancia de las características, lo que facilita la interpretación del modelo.

Las desventajas que tiene XGBoost son: a pesar de su eficacia, requiere una afinación cuidadosa de hiperparámetros, como la profundidad máxima del árbol y la tasa de aprendizaje, para lograr su mejor rendimiento, puede requerir más memoria que otros algoritmos, especialmente si se trabaja con conjuntos de datos muy grandes y los modelos basados en árboles tienden a ser menos interpretables que los modelos lineales, lo que puede dificultar la explicación de las predicciones.

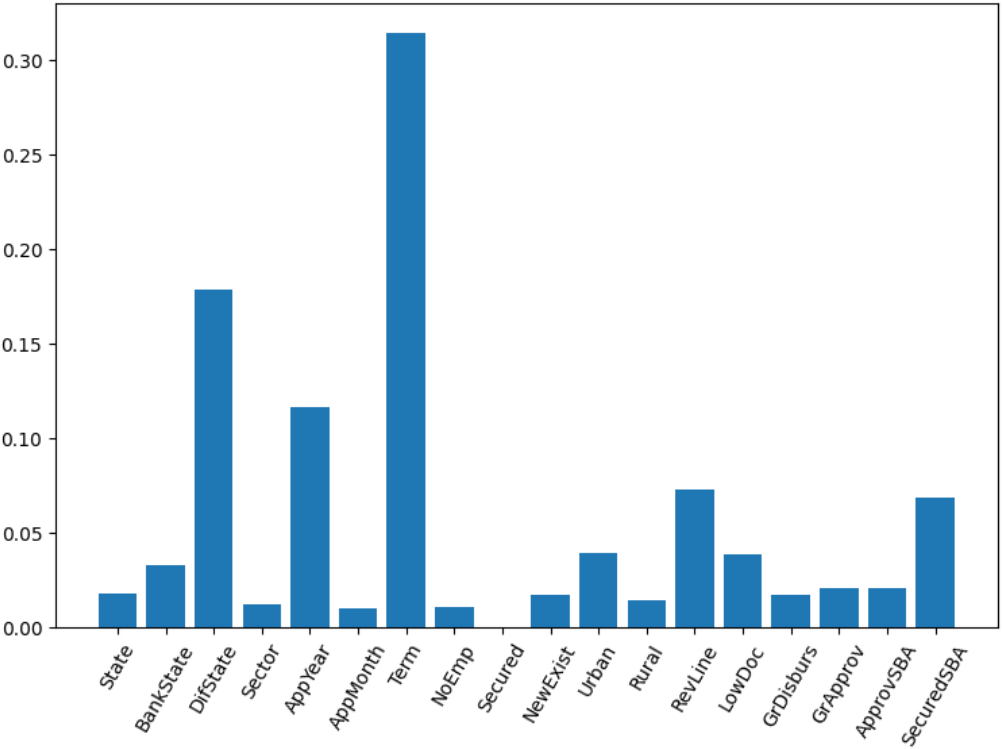
Para realizar el **ajuste del modelo** de XGBoost utilizamos GridSearch, para encontrar la configuración de hiperparámetros óptimos, los resultados se detallan en la Tabla 31, se ha conseguido un accuracy de **0.9562** utilizando los hiperparametros encontrados.

**Tabla 31**  
Hiperparametros óptimos de XGBoost.

|  |  |  |
| --- | --- | --- |
| **Hiperparametro** | **Valor** | **Detalle** |
| learning\_rate | 0.2 | Reduce la contribución de cada árbol |
| max\_depth | 10 | La profundidad máxima del árbol |

La Importancia de las Características en XGBoost refleja cómo cada variable contribuye a la predicción de pertenencia de cada clase, como se observa en la Figura 28.

**Figura 28***Importancia de Características de XGBoost.*

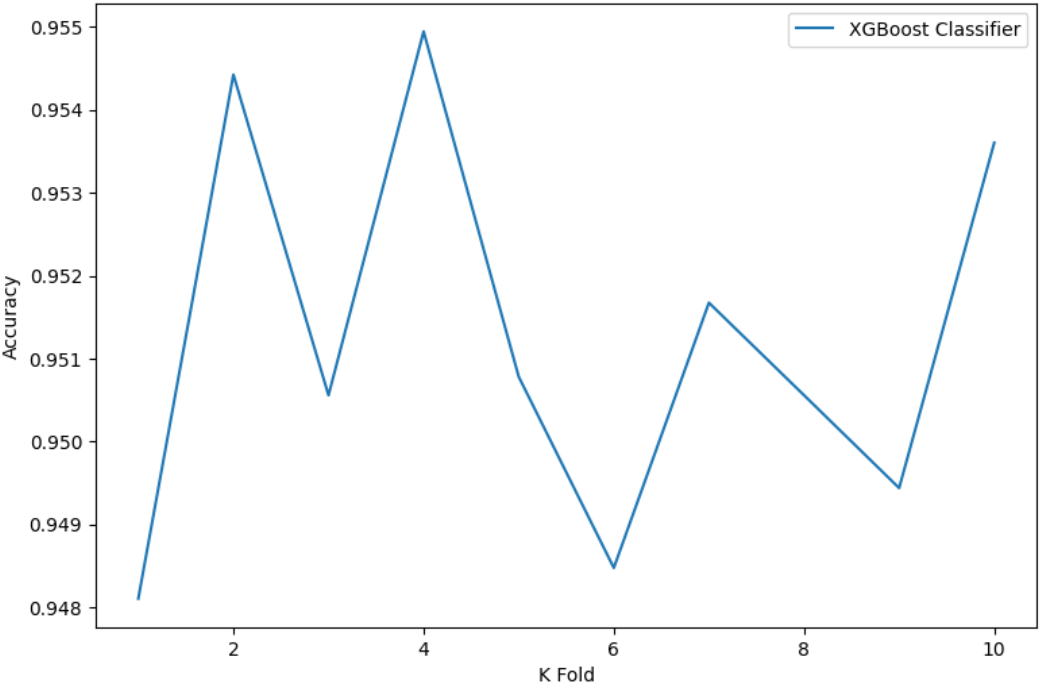


Se ha aplicado Eliminación de Características Recursivas (RFE), para mejorar el rendimiento de un modelo al identificar y eliminar sistemáticamente las variables menos importantes; las variables seleccionadas fueron: BankState, DifState, AppYear, Term, Urban, RevLine, LowDoc, ApprovSBA y SecuredSBA, logrando 0.9554 de accuracy.

Se ha testeado el modelo con los conjuntos de datos balanceados, obteniendo accuracy de 0.9246 y 0.9312 en los datos de submuestreo y sobremuestreo respectivamente.

Los resultados obtenidos en la Validación Cruzada se encuentran en la Figura 29.

**Figura 29***Validación Cruzada de XGBoost.*



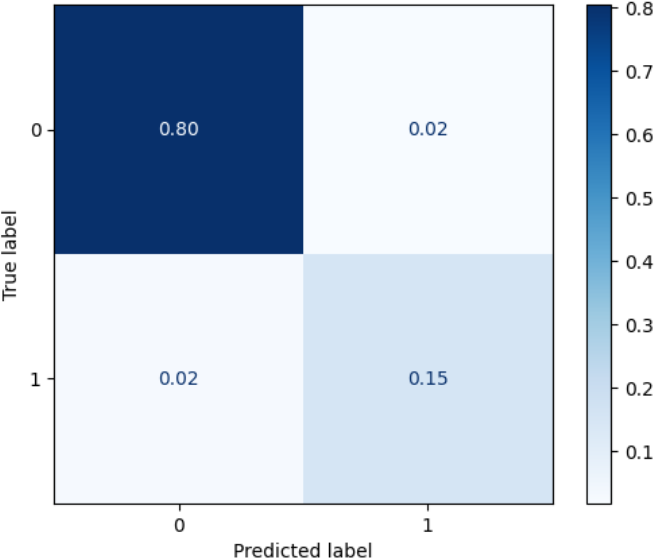
Las Métricas de Evaluación (detalladas en la Tabla 32) fueron obtenidas aplicando los hiperparametros óptimos encontrados, considerando todas las variables, entrenando el modelo definitivo con los conjuntos de datos de entrenamiento y validación.

**Tabla 32**  
Métricas de Evaluación de XGBoost.

|  |  |  |  |
| --- | --- | --- | --- |
| **XGBoost** | | | |
| Accuracy | Precision | Recall | F1-score |
| 0.9566 | 0.9311 | 0.9185 | 0.9246 |

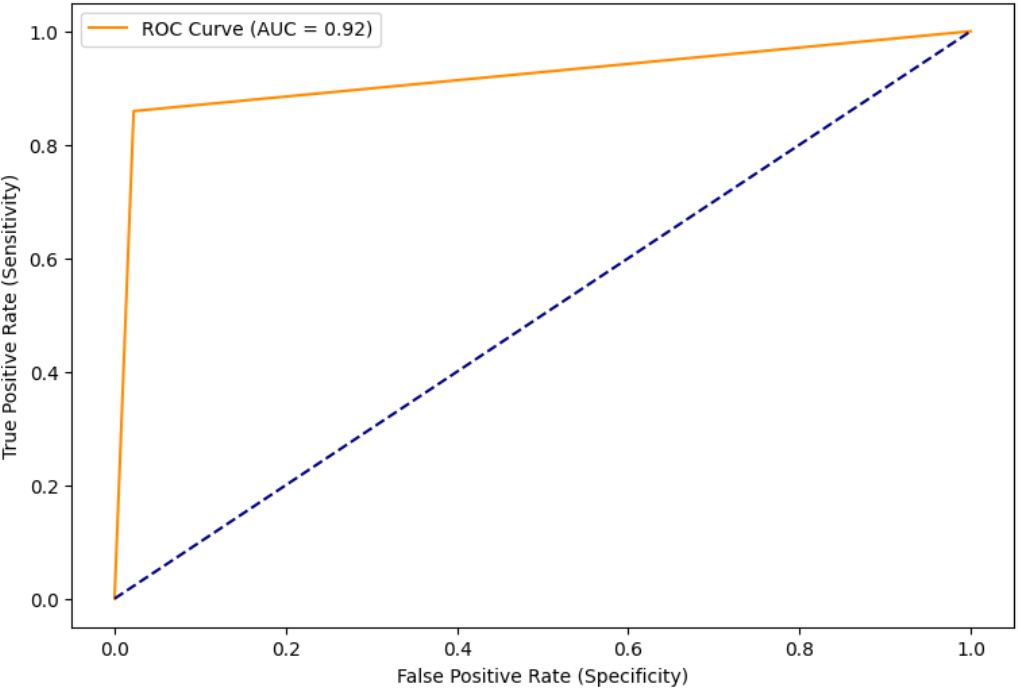
La Matriz de Confusión resultante del modelo optimizado se muestra en la Figura 30.

**Figura 30***Matriz de Confusion de XGBoost.*



El Área bajo la curva ROC (AUC) del modelo optimizado se encuentra en la Figura 31.

**Figura 31***Curva ROC y AUC de XGBoost.*



7.4 Modelos de Deep Learning

1. Resultados
2. Conclusiones

1. www.kaggle.com [↑](#footnote-ref-1)
2. www.kaggle.com/datasets/mirbektoktogaraev/should-this-loan-be-approved-or-denied [↑](#footnote-ref-2)