Modelo predictivo para la detección de fraudes financieros y estimación del riesgo crediticio en tiempo real en el sector bancario

Esteban Daza Alzate

Asesor

Sixyel Jeyson Castañeda Coronado

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2025

Nota de Aceptación

_____

Nombre Director de Trabajo de Grado

_____ _____

Jurado Jurado

2025

Resumen

El proyecto desarrolla un modelo predictivo para la detección de fraude financiero en tiempo real y generar una estimación del riesgo de no pago en solicitudes de crédito, basado en el contexto operativo de un banco de Colombia. Por temas de cumplimiento de la normativa de protección de datos, se utilizan bases de datos sintéticas, que imitan la composición de los datos de información crediticia y transaccional.

Este modelo está centrado en la identificación de patrones a partir de datos previos, utilizando secciones avanzadas de análisis y algoritmos. El proceso metodológico abarca la etapa de preparación de datos, el ejercicio de análisis exploratorio de datos, ventana de validez cruzada, para asegura robustez y desempeño del modelo. Otros hitos para asegurar una implementación robusta que permita la evaluación del desempeño mediante métricas especializadas como precisión, sensibilidad y especificidad. Además, el enfoque no solo apunta a la detección de fraudes en tiempo real, sino que ha de estimar la probabilidad de que una solicitud crediticia se convierta en una obligación desatendida; de este modo, el proyecto propone el proceso de aprobación y gestión del riesgo para las áreas de impacto. Resultados para fortalecer asegurar la seguridad operativa y la toma de decisiones en un banco de Colombia.

***Palabras clave*:** Predicción, fraude, riesgo, analítica, seguridad.

Abstract

This project develops a predictive model for real-time financial fraud detection and the estimation of default risk in credit applications, based on the operational context of a Bank in Colombia. To comply with data protection regulations, synthetic databases are used, replicating the structure of credit and transactional information.

The model focuses on identifying patterns from historical data through advanced analytical techniques and supervised classification algorithms. The methodological process includes data preparation, exploratory data analysis, and cross-validation to ensure robustness and model performance. Additional stages are dedicated to evaluating performance with specialized metrics such as accuracy, sensitivity, and specificity. Beyond real-time fraud detection, the approach also estimates the probability that a credit application will become a defaulted obligation. In this way, the project strengthens credit approval and risk management processes within the institution. The expected results aim to reinforce operational security and decision-making at a Bank in Colombia, while providing a tool that can be applied to other financial products and replicated by different institutions in the sector.

Keywords**:** Prediction, Fraud, Risk, Analytics, Security.

Contenido

Introducción 9

Descripción del Problema 10

Planteamiento del Problema 11

Sistematización del Problema 12

Justificación 13

Objetivos 14

- Objetivo General 14
- Objetivos Específicos 14

Marco de Referencia 15

- Estado del Arte 15
- Marco Contextual 15
- Marco Conceptual 16
- Marco Normativo 17

Metodología 19

- Método 19
- Tipo de Estudio 19
- Recolección de Datos 19

Generación y Validación de la Base de Datos 21

- Principios de la Data Sintética 21
- Generación de Variables 22
- Construcción de la Variable Objetivo Mora/Riesgo 23
- Tamaño del Dataset y Estructura Final 24

Resultados Esperados.....25

Metodología Aplicada.....26

Flowchart 28

Preparación de Variables..... 28

Resultados y Evaluación del Modelo.....30

Evaluación del Modelo 32

Evidencia Técnica del Desarrollo del Modelo 33

Preparación de Datos y Preprocesamiento 34

Entrenamiento, Búsqueda de Hiperparámetros y Selección del Modelo 35

Selección del Umbral Operativo 35

Evaluación Final del Modelo 36

Gráficas Generadas Automáticamente 37

Inferencia y Producción 42

Conclusiones.....43

Recomendaciones 45

Referencias.....46

Lista de Tablas

Tabla 1 *Variables Nivel de Importancia* 30

Lista de Figuras

Figura 1 *Matriz de Confusión* 37

Figura 2 *Curva ROC* 38

Figura 3 *Curva Precisión–Recall (PR-AUC)* 39

Figura 4 *Matriz de Confusión* 40

Figura 5 *Curva de Lift por Deciles* 41

Introducción

Actualmente, el entorno financiero global se encuentra en constante transformación debido al crecimiento acelerado de las transacciones digitales y a la creciente complejidad de las amenazas cibernéticas. Esta situación ha incrementado la exposición de las entidades bancarias a diversos tipos de fraudes financieros, lo que se constituye en uno de los principales retos para su sostenibilidad y credibilidad. (Aguilar Antonio, 2021).

La progresiva digitalización de los servicios financieros ha acarreado una mayor vulnerabilidad a delitos como el fraude bancario y el incumplimiento crediticio. Estos fenómenos impactan no solo la rentabilidad de las instituciones, sino también su reputación y la confianza de los usuarios. (Patino Orozco, 2021; Carrión-Barco et al., 2021).

Descripción del Problema

Tradicionalmente, la detección de fraudes se ha basado en sistemas de reglas estáticas que, aunque resultaron útiles en su momento, muestran limitaciones ante los patrones dinámicos y sofisticados empleados por los defraudadores modernos. (Carrión-Barco et al., 2021). Los sistemas tradicionales, fundamentados en reglas fijas, exhiben limitaciones significativas frente a los cambios dinámicos en los patrones de comportamiento de los defraudadores y la diversidad de perfiles crediticios. (de la Rosa Rodríguez, 2021).

Planteamiento del Problema

En consecuencia, las instituciones financieras se ven compelidas a adoptar nuevas estrategias sustentadas en tecnologías emergentes como el machine learning para reforzar su capacidad de respuesta. El aprendizaje automático posibilita la identificación de anomalías y patrones sospechosos mediante el análisis masivo de datos históricos, ofreciendo una alternativa eficiente y escalable para contrarrestar el fraude en tiempo real. (Gutierrez Portela et al., 2023). De igual manera, su aplicación se ha extendido a la predicción del riesgo crediticio, facilitando la estimación de la probabilidad de que una obligación devenga en mora o incumplimiento. (Borrero-Tigreros & Bedoya-Leiva, 2020).

En este marco, el presente trabajo se propone desarrollar un modelo predictivo que identifique, con un elevado grado de precisión, solicitudes de crédito con alto riesgo de impago, así como transacciones potencialmente fraudulentas. El propósito es optimizar los procesos internos de un banco en Colombia y perfeccionar la toma de decisiones en la gestión del riesgo financiero.

Sistematización del Problema

¿Cómo puede un modelo predictivo identificar, con un elevado grado de precisión, solicitudes de crédito con alto riesgo de impago en un banco en Colombia?

¿De qué manera es posible detectar transacciones potencialmente fraudulentas en tiempo real a partir del análisis masivo de datos históricos?

¿Qué técnicas de machine learning resultan más eficaces para anticipar patrones de fraude y riesgo crediticio en el contexto de un banco en Colombia?

Justificación

La naturaleza compleja y volátil del entorno financiero contemporáneo demanda modelos capaces de adaptarse con rapidez a nuevas amenazas y patrones, lo que posiciona al machine learning como una solución viable e indispensable. Diversas investigaciones han comprobado la eficacia de algoritmos como Random Forest y XGBoost en tareas de clasificación y detección de anomalías, tanto en el ámbito del fraude como en la predicción del riesgo crediticio. (Espinosa Zúñiga, 2020; Su et al., 2023). Estos modelos generan indicadores de riesgo más robustos y personalizados, facilitando la toma de decisiones fundamentadas en las áreas de análisis y aprobación. Adicionalmente, la implementación de estas soluciones tecnológicas fortalece la capacidad operativa de los bancos ante escenarios de morosidad, permitiendo priorizar los casos de mayor riesgo y optimizar los recursos disponibles. (Fierro Torres et al., 2022; Gil-Vera Quintero-López, 2021).

Desde esta perspectiva, el presente proyecto trasciende una mera propuesta técnica para configurarse como una respuesta estratégica a los desafíos de un sistema financiero cada vez más digitalizado, competitivo y expuesto a riesgos emergentes.

Objetivos

Objetivo General

Diseñar un modelo predictivo basado en machine learning que permita detectar fraudes financieros en tiempo real y estimar el riesgo de impago en solicitudes de crédito dentro de un banco en Colombia.

Objetivos Específicos

Analizar los métodos actuales utilizados en la detección de fraudes y evaluación de riesgo crediticio en el sector bancario colombiano.

Seleccionar las variables más relevantes a partir del análisis exploratorio de datos históricos de solicitudes de crédito y comportamiento transaccional.

Desarrollar un modelo predictivo utilizando algoritmos como Random Forest y XGBoost, entrenado con datos anonimizados para un banco en Colombia.

Evaluar el desempeño del modelo mediante métricas como precisión, sensibilidad, especificidad y área bajo la curva ROC.

Establecer recomendaciones para la implementación del modelo en procesos de decisión crediticia y monitoreo de transacciones.

Marco de Referencia

Estado del Arte

El sistema financiero actual enfrenta un entorno progresivamente desafiante debido a la transformación digital, el aumento de las transacciones electrónicas y la creciente sofisticación de los métodos de fraude. Estos factores han expuesto a las instituciones bancarias a un riesgo operacional considerable, afectando tanto sus resultados financieros como la confianza del público. (Aguilar Antonio, 2021; Patino Orozco, 2021).

Los métodos tradicionales de detección de fraude y evaluación del riesgo crediticio, basados en reglas estáticas, se muestran insuficientes frente a las estrategias dinámicas de los defraudadores actuales (de la Rosa Rodríguez, 2021). Aunque estos sistemas fueron útiles en su momento, carecen de flexibilidad y resultan superados por patrones de comportamiento en constante evolución. Asimismo, el incremento en el volumen de solicitudes de crédito y transacciones exige soluciones más automatizadas y adaptativas. (Milián Gómez & Rodríguez Corría, 2021).

Marco Contextual

Dado que el presente proyecto se desarrolla en un banco de Colombia, su implementación contribuiría directamente a reducir la tasa de morosidad y a detectar oportunamente transacciones inusuales, generando valor tangible para la entidad. Asimismo, se espera que esta iniciativa sirva como referente para otras instituciones financieras en Colombia que persiguen modernizar sus procesos de evaluación y control del riesgo. (Borrero-Tigreros & Bedoya-Leiva, 2020; Chakri et al., 2023).

En este contexto, el machine learning se erige como una alternativa poderosa para abordar problemas complejos de clasificación y predicción, especialmente en la identificación de

operaciones anómalas o riesgosas. (Gutierrez Portela et al., 2023; Quirumbay Yagual et al., 2022).

Modelos como Random Forest y XGBoost han demostrado ser eficaces en la detección de fraudes y en la predicción de la probabilidad de incumplimiento crediticio. (Espinosa Zúñiga, 2020; Rao et al., 2023). Además, investigaciones recientes subrayan la relevancia de desarrollar soluciones explicables, capaces no solo de proporcionar un porcentaje de riesgo, sino también de esclarecer las razones por las cuales una solicitud es clasificada como riesgosa. (ben Jabeur et al., 12 2023; Su et al., 2023).

Estos modelos aportan un valor estratégico a las instituciones financieras al posibilitar una toma de decisiones basada en evidencia y análisis riguroso. Por otra parte, el uso de modelos predictivos no se circunscribe a evitar pérdidas por impagos, sino que también optimiza los procesos internos del banco al priorizar las solicitudes de crédito según su nivel de riesgo, mejorando los tiempos de respuesta y la asignación de recursos. (Fierro Torres et al., 2022).

Marco Conceptual

Fraude financiero: Se refiere a prácticas ilegales destinadas a obtener beneficios económicos mediante engaño en los sistemas financieros. Su impacto resulta crítico para la reputación y estabilidad de las entidades bancarias. (Aguilar Antonio, 2021; Gutierrez Portela et al., 2023).

Riesgo crediticio: Constituye la probabilidad de que un cliente incumpla con el pago de una obligación financiera. Su predicción efectiva es esencial para evitar pérdidas en las instituciones de crédito. (Borrero-Tigreros & Bedoya-Leiva, 2020; Fierro Torres et al., 2022).

Machine learning: Es una técnica de inteligencia artificial que permite crear modelos capaces de aprender de los datos y mejorar con el tiempo sin programación explícita. (Suazo 13 Galdames, 2023; Chakri et al., 2023).

Random Forest: Algoritmo basado en árboles de decisión que combina múltiples predictores para aumentar la precisión. Es útil en tareas de clasificación como la detección de fraude y morosidad. (Espinosa Zúñiga, 2020).

XGBoost: Algoritmo de aprendizaje automático por gradiente que optimiza el rendimiento en clasificación y predicción de riesgos. Resulta eficaz ante datos desbalanceados. (Rao et al., 2023; ben Jabeur et al., 2023).

Modelos explicables (Explainable AI): Facilitan la interpretación de las decisiones de un modelo predictivo, aportando transparencia y confianza en contextos financieros. (Su et al., 2023; ben Jabeur et al., 2023).

Análisis exploratorio de datos (EDA): Es la fase inicial del análisis de datos, enfocada en visualizar, comprender y preparar la información para su uso en modelos predictivos. (Chakri et al., 2023; Sahoo et al., 2019).

Detección en tiempo real: Consiste en identificar transacciones o eventos sospechosos al momento de ocurrir, permitiendo una acción rápida para mitigar el impacto del fraude. (Gutierrez Portela et al., 2023).

Marco Normativo

Ley 1581 de 2012, por la cual se dictan disposiciones generales para la protección de datos personales, constituye el marco de referencia principal en el tratamiento de información sensible en el país. (Congreso de Colombia, 2012).

Ley 1266 de 2008, denominada Ley de Habeas Data financiero, regula la administración de datos personales en centrales de riesgo y el reporte de información crediticia. (Congreso de Colombia, 2008).

Ley 1328 de 2009, mediante la cual se dictan normas en materia financiera, de seguros, del mercado de valores y se regula la protección al consumidor financiero. (Congreso de Colombia, 2009).

Circular Básica Jurídica de la Superintendencia Financiera de Colombia, que establece lineamientos para la gestión del riesgo operativo y de fraude, incluyendo el Sistema de Administración de Riesgo Operativo (SARO). (Superfinanciera, 2017).

SARLAFT (Sistema de Administración del Riesgo de Lavado de Activos y de la Financiación del Terrorismo), regulado por la Superintendencia Financiera, que establece mecanismos para la detección de operaciones inusuales y el monitoreo de riesgos asociados. (Superfinanciera, 2017).

Como complemento internacional, se tiene en cuenta la norma ISO/IEC 27001, referente a la gestión de seguridad de la información, adoptada como estándar global para la protección de activos de información. (Viguri Cordero, 2021).

Metodología

El desarrollo del presente proyecto se estructura en cinco fases metodológicas orientadas a construir un modelo predictivo capaz de detectar fraudes financieros en tiempo real y estimar el riesgo de impago en solicitudes de crédito. La metodología integra técnicas de análisis exploratorio de datos con algoritmos de machine learning de alto rendimiento. Recolección y preparación de los datos: Se recopila información histórica anonimizada de un banco de Colombia, que abarca registros de solicitudes de crédito, características sociodemográficas y comportamiento financiero. Estos datos se someten a procesos de limpieza, transformación y estructuración para garantizar su calidad y utilidad analítica. (Chakri et al., 2023).

Método

Entrenamiento del modelo de machine learning. Se implementan algoritmos supervisados como Random Forest y XGBoost, reconocidos por su eficacia en tareas de clasificación y detección de anomalías en el sector financiero. (Espinosa Zúñiga, 2020; Rao et al., 2023). El modelo se entrena con una partición del conjunto de datos y se valida mediante técnicas como la validación cruzada.

Tipo de Estudio

El enfoque corresponde a un estudio aplicado con alcance predictivo.

Recolección de Datos

Análisis exploratorio de datos (EDA) Se efectúa un análisis exploratorio para identificar patrones relevantes, valores atípicos y correlaciones entre variables. Esta etapa permite comprender la estructura de los datos y seleccionar los atributos más significativos para el modelo. (Sahoo et al., 2019).

Evaluación del modelo. El desempeño del modelo se evalúa utilizando métricas como precisión, sensibilidad, especificidad y área bajo la curva ROC (AUC), con el objetivo de determinar su capacidad para discriminar entre solicitudes confiables y riesgosas. (ben Jabeur et al., 2023).

Estimación del riesgo y simulación operativa. Se aplica el modelo en escenarios simulados para estimar la probabilidad de impago en nuevas solicitudes de crédito. Además, se analiza su comportamiento en condiciones de operación en tiempo real, evaluando su posible integración en los procesos internos del banco. (Gutierrez Portela et al., 2023).

Generación y Validación de la Base de Datos

En el contexto bancario no se puede extraer una base con información real, hay temas de privacidad, reserva bancaria, gobierno de datos y en general reglas internas que son muy estrictas, para poder desarrollar el proyecto en un entorno académico, se trabajó con una base de datos sintética, construida con cierta lógica, para que se parezca a lo que normalmente ve en un proceso de originación y análisis de riesgo, el objetivo real era probar el enfoque completo, end-to-end el preprocesamiento, el entrenamiento, la evaluación, cómo se define un umbral, cómo se generan artefactos del modelo y cómo se hace scoring con datos nuevos, demostrar que el pipeline funciona técnica y conceptualmente.

Principios de la Data Sintética

La generación de la base se hizo con unas reglas generales necesarias. Coherencia interna que las variables tengan sentido entre sí. Por ejemplo, que la cuota tenga relación con el monto y el plazo, que el ratio cuota, ingreso no sea exagerado, la edad y la ocupación no tengan sentido.

Realismo estadístico: que los rangos y distribuciones sean plausibles, ingresos con cola a la derecha, montos también con cola a la derecha, el score acotado.

Relaciones “más o menos causales”: no pretendiendo un modelo perfecto, pero sí que existan correlaciones típicas en riesgo crediticio, cuota, ingreso más alta suele aumentar la probabilidad de mora, score más alto tiende a bajar el riesgo.

Reproducibilidad y trazabilidad: que el proceso se pudiera repetir, con semilla random_state y reglas claras.

Se armó un diccionario de datos con el esquema del dataset, se definió el tipo de variable numérica, categórica, fecha, rangos esperados y restricciones básicas como no negatividad, integridad de fechas, categorías válidas, etc. También se incluyeron algunas variables derivadas

tipo ratios o componentes temporales. En cuanto a variables, la base incluyó cosas coherentes con un proceso de crédito, como score_crediticio, ingresos_mensuales, valor_credito, plazo_meses, cuota_estimada, cuota_ingreso, edad, dependientes, ocupacion, producto y además variables de contexto como ubicación o segmento, porque son variables que suelen estar en un proceso de originación y después aparecen en el análisis de importancia del modelo.

Generación de Variables

Edad: se generó en un rango adulto, pero no uniforme, se concentró más en edades laborales activas.

Dependientes: se generó como discreta, con mayor probabilidad en 0–2 y menor probabilidad en valores altos. también consistencia, edades más bajas tienden a tener menos dependientes, aunque no es una regla rígida.

Ocupación / vínculo: se generó como categórica con proporciones plausibles, empleado, independiente, pensionado, etc. Esta variable no se usa para aproximación de perfiles de ingresos o estabilidad, en un escenario real estas variables se deberían revisar con criterios de equidad, cumplimiento, etc.

Ingresos mensuales: se modelaron con distribución asimétrica cola derecha, que es lo típico, además se condicionaron parcialmente por ocupación/segmento, por ejemplo, medias diferentes, para que no parezca que todos ganan igual.

Score crediticio: se generó en un rango acotado similar a escalas conocidas y se relacionó suavemente con estabilidad e ingresos.

Valor del crédito: se generó también con cola derecha y con dependencia parcial de ingresos y score, la lógica es: montos altos son más probables en clientes con mayor capacidad o mejor perfil.

Plazo: se generó como discreto en rangos típicos y una relación suave con el valor del crédito, montos más altos tienden a plazos mayores.

La cuota estimada se derivó usando valor del crédito, plazo y una tasa simulada o aproximación de tasa efectiva mensual, la idea era que la cuota no fuera un número cualquiera, sino una consecuencia del crédito.

Luego se calculó cuota_ingreso = cuota_estimada / ingresos_mensuales. Esta variable es usada en riesgo porque aproxima la presión financiera y en el dataset sintético ayuda a amarrar el comportamiento de la variable objetivo.

Se hicieron validaciones para evitar valores imposibles negativos o extremos sin sentido, cuando salían casos atípicos, se restringían con topes, en el pipeline se reforzó con winsorización.

Se agregó una fecha de solicitud y se derivaron mes, año y trimestre, en un entorno real esto sirve para estacionalidad, drift, aquí en la base sintética es más estructura para mostrar que el pipeline puede trabajar con fechas y crear features temporales.

Construcción de la Variable Objetivo Mora/Riesgo

La etiqueta mora, no mora se genera totalmente aleatoria, entonces el modelo aprende poco o aprende cosas raras, por eso la variable objetivo se generó de forma controlada, con un enfoque probabilístico.

Primero se calculó un “score de riesgo latente” combinando señales típicas:

Score bajo = aumenta riesgo.

Cuota, ingreso alto = aumenta riesgo.

Ingresos bajos vs monto = aumenta riesgo.

Montos altos y plazos largos = aumentan riesgo, pero con peso moderado.

Ajustes de contexto producto, segmento con menos peso, para que exista variación.

El score se transformó en probabilidad con una función tipo logística para mantener 0–1.

Se calibró la prevalencia final para que el evento fuera minoritario, porque en riesgo normalmente no se tienes50/50 de morosos vs no morosos, se ajustó el intercepto o el nivel del muestreo hasta llegar a un desbalance razonable, esto importa porque métricas como PR-AUC y la curva precision-recall dependen bastante del desbalance.

Tamaño del Dataset y Estructura Final

La base se definió con un tamaño manejable para experimentar en Python, alrededor de 20.000 registros y 36 variables iniciales, luego en preprocesamiento, las categóricas se expanden con One-Hot Encoding, entonces internamente el número de features crece.

Para evitar que la base fuera solo números, se aplicaron controles de validación. Rangos para edad dentro de límites, ingresos positivos, montos y plazos plausibles, cuota y cuota, ingreso sin valores imposibles. Coherencia lógica para ingresos vs cuota vs cuota, ingreso evitando escenarios absurdos, también coherencia básica entre producto y montos, plazos cuando aplicaba y edad vs ocupación, por ejemplo, no pensionados en cantidad en edades bajas. Validaciones estadísticas se revisaron distribuciones asimetría en ingresos y montos, correlaciones esperadas cuota-ingreso vs riesgo, score vs riesgo y el desbalance final.

Resultados Esperados

Se espera que el modelo desarrollado identifique, con un alto grado de precisión, las solicitudes de crédito con mayor probabilidad de impago, así como transacciones potencialmente fraudulentas. La herramienta asigna una probabilidad de riesgo a cada caso, permitiendo priorizar la gestión y tomar decisiones más acertadas desde las áreas de crédito y seguridad bancaria. Se proyecta una mejora significativa en la eficiencia de los procesos mediante la reducción de falsos positivos y negativos, así como una respuesta más efectiva ante eventos de riesgo. Por otro lado, el modelo contribuye a disminuir la tasa de morosidad del banco al identificar tempranamente a clientes con comportamiento financiero riesgoso. A nivel institucional, se anticipa que se fortalecerá la seguridad operativa de un banco de Colombia, se mejorará la experiencia del cliente y se sentarán las bases para implementar sistemas de análisis automatizado. Finalmente, se prevé que el modelo facilite la toma de decisiones basada en datos y fomente una cultura de prevención, con herramientas integrables de manera efectiva en los flujos internos de trabajo del banco.

Metodología Aplicada

Primero, ingesta del histórico consolidado. Revisión de tipos, duplicados, y no se usa campos de texto libre/PII que no sumaban o estorbaban. Luego, imputación de faltantes con mediana, que es más robusta que el promedio cuando hay cola pesada. Y winsorización p1–p99 para recortar outliers sin cortar la variable; la idea es que un valor extremo no distorsione todo.

Después, categóricas a One-Hot. Ya que el modelo no entiende strings. Se aplicó umbral de frecuencia mínima para agrupar categorías raras y evitar explosión de columnas, además de limitar cardinalidad. Aparte, de la fecha de solicitud saqué año, mes, trimestre. Asumiendo que lo temporal a veces tiene peso en comportamiento de pago.

Con eso, split estratificado 70/30. Proporción de mora preservada en train y test para que no se desconfigure el problema. En modelado, LR, RF, XGB con Random/Grid Search y StratifiedKFold como validación. El scorer fue AUC-ROC si el objetivo es rankear, tiene sentido.

Calibración con CalibratedClassifierCV cuando el pipeline lo permite. Luego, la selección de umbral se hizo por tres enfoques:

F1 (equilibrio precisión–recall).

constraints (si negocio pide mínimo X de recall).

y costo (ponderando FN vs FP) si aplica.

Para este piloto quedó 0.29. Con ese número, se pasó a evaluación final en el bloque de prueba, métricas, matriz de confusión, curvas ROC/PR, y la curva por deciles/KS para verificar que el riesgo de mora efectivamente se apile arriba.

Al cierre, se guardan los artefactos: preprocessor.pkl y best_model.pkl, cada uno con su checksum SHA-256 por integridad del modelo y se deja el entorno listo para scoring sin internet

dependencias fijadas, sin librerías de visualización innecesarias, posibilidad de instalar desde wheels internos. El flujo operativo final toma un CSV nuevo, valida columnas, aplica el mismo preprocesamiento, calcula predicción de probabilidad, compara contra 0.29 y genera predicciones.csv con id, score, predicción. Sencillo de usa y repetible.

Flowchart

Este flowchart muestra, de forma simple, cómo pasé de un histórico de créditos a una señal preventiva que prioriza casos con mayor probabilidad de caer en mora y que también puede levantar patrones atípicos cercanos a fraude. La idea es dejar de reaccionar tarde el pipeline toma los datos, los limpia, entrena modelos, calcula probabilidades y produce un archivo con id, score y bandera listo para gestión. Todo con controles de trazabilidad y seguridad para que cada corrida sea consistente. Es la radiografía del proceso de datos crudos a decisión operativa.

Ingesta y control de calidad. Se carga el histórico (base_modelo_mora.csv) y se hace una sanidad básica tipos de datos, duplicados, exclusión de texto libre o PII que no aporta. Aquí también se detecta el separador, se unifican formatos y se registran avisos de columnas faltantes.

Preparación de Variables

Numéricas: imputación de faltantes con mediana y winsorización para evitar que outliers distorsionen.

Categóricas One-Hot Encoding con manejo de categorías

Temporales derivación de año,mes y trimestre desde fecha solicitud para capturar estacionalidad.

Partición estratificada. Se separa train/test (70/30) manteniendo la proporción de mora. Esto permite evaluar de forma justa y evita que el modelo aprenda una distribución distinta a la que verá en operación.

Entrenamiento y selección. Se prueban Regresión Logística, Random Forest y XGBoost con validación cruzada estratificada y búsqueda de hiperparámetros. El criterio principal es AUC-ROC y se revisa estabilidad entre pliegues. Cuando aplica, se calibran probabilidades para que sean interpretables $0.70 \approx 70\%$.

Umbral operativo. A partir del score continuo se define un umbral. Primero se busca el máximo F1, si negocio exige mínimos de precisión/recall, se ajusta. También puede considerarse un esquema de FN/FP. En el piloto quedó 0.29 por buen equilibrio cobertura–calidad.

Evaluación y ranking. En el test se reportan AUC, precisión, recall, F1 y matriz de confusión al umbral elegido. Además, KS/deciles para confirmar que el riesgo se concentra arriba, eso valida la priorización por ranking, se gestiona del decil 1 hacia abajo.

Artefactos y gobernanza. Se guardan preprocessor.pkl y best_model.pkl con checksum SHA-256, dependencias fijadas y evidencia de métricas. El runtime de scoring es mínimo, sin librerías de visualización.

Scoring en producción. Con un archivo nuevo de nuevos créditos, se valida el esquema, se aplican las mismas transformaciones y se calcula predict_proba. Se compara contra el umbral y se genera predicciones.csv con id, score, pred. Eso es lo que consume operación para priorizar.

Con eso, el diagrama amarra el flujo completo: datos → preparación → modelo → decisión → archivo accionable. Deja claro dónde están los controles validación, calibración, gobernanza) y cómo se convierte un histórico en una lista priorizada que sí sirve en el día a día.

Resultados y Evaluación del Modelo

El modelo alcanzó un AUC de 0.71, no es perfecto, pero sí demuestra una buena capacidad para diferenciar entre los créditos que probablemente van a caer en mora y los que no. El modelo sí entiende el patrón general, no está adivinando.

Luego, con la técnica de Permutation Importance, se mira qué variables fueron las que más peso tuvieron y acá se empieza a poner interesante porque uno ve que el modelo sí está priorizando lo que en la práctica uno también miraría.

Tabla 1

Variables Nivel de Importancia

Variable	Importancia (%)	Interpretación técnica
12 – score_crediticio	46.7%	Fue la variable más fuerte, y tiene sentido. Este puntaje mide el riesgo histórico del cliente, cuando ese score baja, la probabilidad de que caiga en mora sube mucho.
9 – ingresos_mensuales	25.0%	Refleja directamente la capacidad de pago. Si la persona gana poco, pues es obvio que se complica más para cumplir.
3 – valor_credito	14.4%	Es el monto total de la obligación. Mientras más alto, más presión financiera y por ende más riesgo.
8 – plazo_meses	9.7%	A más meses, más exposición al impago. Es simple, pero el modelo también lo captó.
124 – producto	2.1%	Algunos productos concentran más mora que otros, como libre inversión o vehículo.
128 – ciudad/departamento	0.8%	Muestra que las condiciones geográficas o socioeconómicas igual afectan. No es lo mismo un cliente en Bogotá que uno en una zona más pequeña.

Variable	Importancia (%)	Interpretación técnica
83 – ocupacion	0.7%	Los que tienen trabajo estable o son pensionados tienden a cumplir más. Los independientes o desempleados, menos.
16 – cuota_ingreso	0.6%	Básicamente, cuánto representa la cuota sobre el ingreso. Si es muy alto ese porcentaje, hay riesgo.
5 – edad	0.55%	Los extremos (muy jóvenes o muy mayores) son más inestables.
7 – dependientes	0.53%	Entre más personas dependen de los ingresos del cliente, más carga económica tiene.

Nota. Top Variables nivel de importancia.

En realidad, solo con estas diez variables se explica casi todo el poder predictivo del modelo y no es casualidad, la mayoría están ligadas a la capacidad de pago del cliente, que sigue siendo el eje del riesgo crediticio.

Ya si uno lo ve en detalle, el score_crediticio fue el de mayor puntaje. Después, los ingresos y el valor del crédito marcan el tono del análisis, mientras que las variables categóricas, como producto o ciudad, ajustan un poco el contexto. Digamos que el modelo logra equilibrar lo cuantitativo con lo descriptivo, sin volverse caótico.

La verdad es que me parece coherente si uno tiene un historial malo, pocos ingresos y un crédito grande, pues el riesgo de caer en mora no es ninguna sorpresa.

También me parece interesante que el modelo capta cosas pequeñas, como el tipo de producto o la región, que uno normalmente no ve tan directo en el día a día, pero que sí influyen. Por ejemplo, hay zonas donde los clientes tienden a retrasarse más, o ciertos productos que son

más sensibles. En fin, no es solo capacidad de pago, también hay una especie de comportamiento típico que el modelo logró detectar.

En términos generales, las variables numéricas ingresos, score, valor del crédito, cuota se llevaron la mayor parte del peso y las categóricas ayudaron más a segmentar. El modelo no solo aprendió a predecir, sino que también da pistas sobre por qué un cliente podría entrar en mora.

Evaluación del Modelo

Se entrenaron tres algoritmos: Regresión Logística, Random Forest y XGBoost. Partición estratificada 70/30, porque la base está desbalanceada cae menos gente en mora que la que paga bien, normal en crédito. Para evaluar, AUC-ROC como métrica principal, además se miró estabilidad entre pliegues con cross-validation que no se caiga en un fold y se infle en otro y reproducibilidad usando la semilla en 42.

Cuando aplica, calibré probabilidades para que un 0.70 signifique, más o menos, 70% real. No siempre se puede, pero cuando sí, mejora interpretación. El umbral operativo salió en 0.29, buscando un equilibrio razonable entre precisión y recall, o sea, el F1 decente sin irse a los extremos.

Números puntuales holdout:

AUC-ROC ≈ 0.715 separa con criterio.

Precision ≈ 0.38 de 10 alertas, ~4 sí terminan morosas.

Recall ≈ 0.13 capta una parte, si negocio pide más cobertura, toca mover umbral o enriquecer variables.

F1 ≈ 0.19 y ~0.34 en el punto óptimo durante la búsqueda de umbral.

Matriz de confusión (0.29): TN≈5057, FP≈164, FN≈678, TP≈101.

Deciles / KS: los primeros deciles concentran mayor tasa de mora. Esto es clave, justifica gestionar de arriba para abajo.

El modelo ordena bien. Para operación eso vale oro, porque la priorización es medio juego ganado. Y si negocio exige, digamos, un recall mínimo, se ajusta el umbral sabiendo que hay trade-off.

Evidencia Técnica del Desarrollo del Modelo

Para esta parte básicamente organicé todo como un proyecto de Python, pero tratando de que fuera reproducible, lo trabajé en VS Code, la estructura quedó así, aunque no todo salió perfecto desde el inicio: config.py, data_prep.py, features.py, train.py, evaluate.py, inference.py, utils.py y una carpeta artefactos donde se guardan los modelos, los pkl, que después uno necesita y no sabe dónde dejó. El dataset principal que usé fue base_modelo_mora.csv y ya para las pruebas finales de scoring usé nuevos_creditos.csv, que al comienzo no estaba contemplado usar.

Sobre el flujo, el pipeline quedó siguiendo una línea lógica, aunque varias veces tuve que devolverme. Primero validación del esquema y tipos, luego, imputación de faltantes, winsorización entre p1 y p99 , después la codificación One-Hot, pero con límite de cardinalidad porque si no el modelo se vuelve un complicado, después la partición estratificada 70/30, que es lo típico y ahí sí entrenamiento. Probé Regresión Logística, Random Forest y XGBoost, cada uno con su búsqueda de hiperparámetros, no todo quedó perfecto desde la primera corrida, pero cuando aplicaba, también hice calibración de probabilidades y luego seleccioné el umbral operativo mirando F1, recall y una función de costo que armé para que tuviera sentido con el problema. Al final, todo lo que había que guardar quedó en: preprocessor.pkl, best_model.pkl y algunos metadatos que sirven para el scoring.

Sobre los fragmentos de código, acá solo dejé unos pedazos representativos para no llenar esto de archivos. Todo lo demás está en los anexos, ahí se puede ver el flujo real del proyecto.

Preparación de Datos y Preprocesamiento

El preprocesamiento, terminó siendo una mezcla de varias cosas que tocaba hacer sí o sí. Primero lo básico imputar lo que venía en blanco, después la winsorización para los valores que se iban a las nubes, p1–p99, ya luego todo el tema de las categorías, que quedó con one-hot, aunque tuve que controlar la cardinalidad, taambién metí unas derivaciones temporales y excluí cualquier columna Información de Identificación Personal PII, eso lo dejé por fuera desde el comienzo para evitar enredos de tratamiento de datos.

```
num_steps = [("imputer", SimpleImputer(strategy="median"))]
if getattr(cfg, "USE_WINSORIZE", False):
    num_steps.append(("winsor", Winsorizer(lo=cfg.WINSOR_LO,
hi=cfg.WINSOR_HI)))
cat_pipe = Pipeline([
    ("imputer", SimpleImputer(strategy="most_frequent")),
    ("ohe", OneHotEncoder(
        handle_unknown="infrequent_if_exist",
        min_frequency=cfg.OHE_MIN_FREQ,
        sparse_output=False,
        dtype=np.float32
    )),
])
```

El conjunto final procesado fue almacenado en preprocessor.pkl, y registro de nombres de variables en feature_names.json.

Entrenamiento, Búsqueda de Hiperparámetros y Selección del Modelo

En train.py básicamente dejé montada una búsqueda aleatoria con RandomizedSearchCV, probé tres tipos de modelos Regresión Logística, Random Forest y XGBoost. Cada uno con sus hiperparámetros ahí medio ajustados.

Todo eso se evaluó usando AUC, con validación cruzada estratificada para que no quedara desequilibrado y el script va probando combinaciones hasta que encuentra algo o al menos lo suficientemente bueno para no seguir dándole vueltas.

best_plain = fit_and_select_model(X_train, y_train, pre, use_smote=False)

best_smote = fit_and_select_model(X_train, y_train, pre, use_smote=cfg.USE_SMOTE_PIPELINE)

best = best_plain if best_plain["score"] >= best_smote["score"] else best_smote

logger.info(f"Mejor modelo: {best['name']} con AUC-ROC CV={best['score']:.4f}")

El mejor modelo, una vez salió elegido, lo volví a entrenar con todo el conjunto de entrenamiento completo, deespués vino la parte de la calibración, que solo la apliqué cuando realmente tenía sentido y casi siempre usé isotonic regression.

Cuando ya quedó listo guardé el artefacto final como best_model.pkl, que es básicamente el modelo calibrado y empaquetado para después al hacer scoring.

Selección del Umbral Operativo

En thresholds.json quedó todo ese registro de los umbrales que probé los de F1, los de las restricciones, los de costo básicamente opciones que se fueron comparando. El sistema al final terminó escogiendo el 0.50, que es como el umbral safe.

Aunque, si uno se va solo por F1, el mejor era 0.29. Lo dejé anotado ahí porque igual sirve para entender cómo se comporta el modelo.

{
  "threshold_final": 0.5,
  "by_f1": { "threshold": 0.29, "f1": 0.342, "precision": 0.253, "recall": 0.528 },
  "by_constraints": { "threshold": 0.5, "f1": -1.0, "precision": 0.0, "recall": 0.0 }
}

Aunque el sistema al final se quedó con el umbral 0.50 más que todo por las constraints, porque tampoco había mucho margen ahí, la verdad es que el **0.29** se comporta mejor. Tiene un F1 más equilibrado y cubre más casos, así que en las recomendaciones lo menciono tal cual. No es que contradiga al sistema ni nada, pero digamos que muestra un panorama más completo de lo que realmente podría funcionar mejor.

Evaluación Final del Modelo

Los resultados del archivo metrics.json son los siguientes:

AUC-ROC: 0.7152

AUC-PR: 0.2795

Precisión (positiva): 0.3811

Recall: 0.1296

F1: 0.1934

Accuracy: 0.8596

Matriz de confusión real (umbral 0.50):

Figura 1

Matriz de Confusión

	Pred 0	Pred 1
Real 0	5057	164
Real 1	678	101

Estos valores muestran que el modelo ordena bastante bien. El AUC quedó en 0.7152, que no es perfecto, pero sí habla de que prioriza bien los casos que tienen más probabilidad de caer en mora, el modelo entiende el ranking, aunque el corte final no es completo.

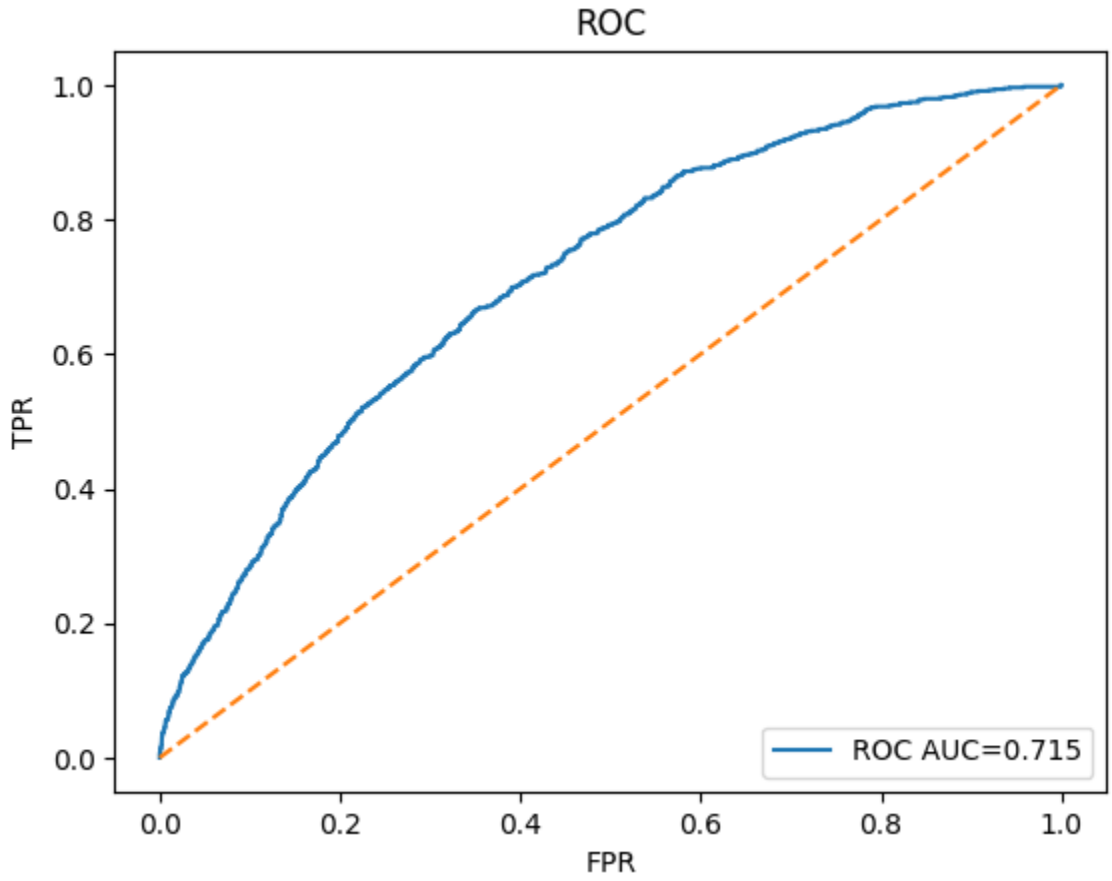
Gráficas Generadas Automáticamente

En evaluate.py se generaron varias gráficas roc.png, pr.png, confusion_matrix.png y lift_curve.png. Ayudan a ver cómo respira el modelo. Todas quedaron guardadas en la carpeta artefactos y de hecho también las metí en el reporte HTML.

```
plot_roc_pr(y_true, y_score, out_dir)
cm = confusion_matrix(y_true, y_pred, labels=[0,1])
plot_confusion(cm, out_dir)
plot_lift(y_true, y_score, out_dir)
save_json(metrics, os.path.join(out_dir, "metrics.json"))
```

Figura 2

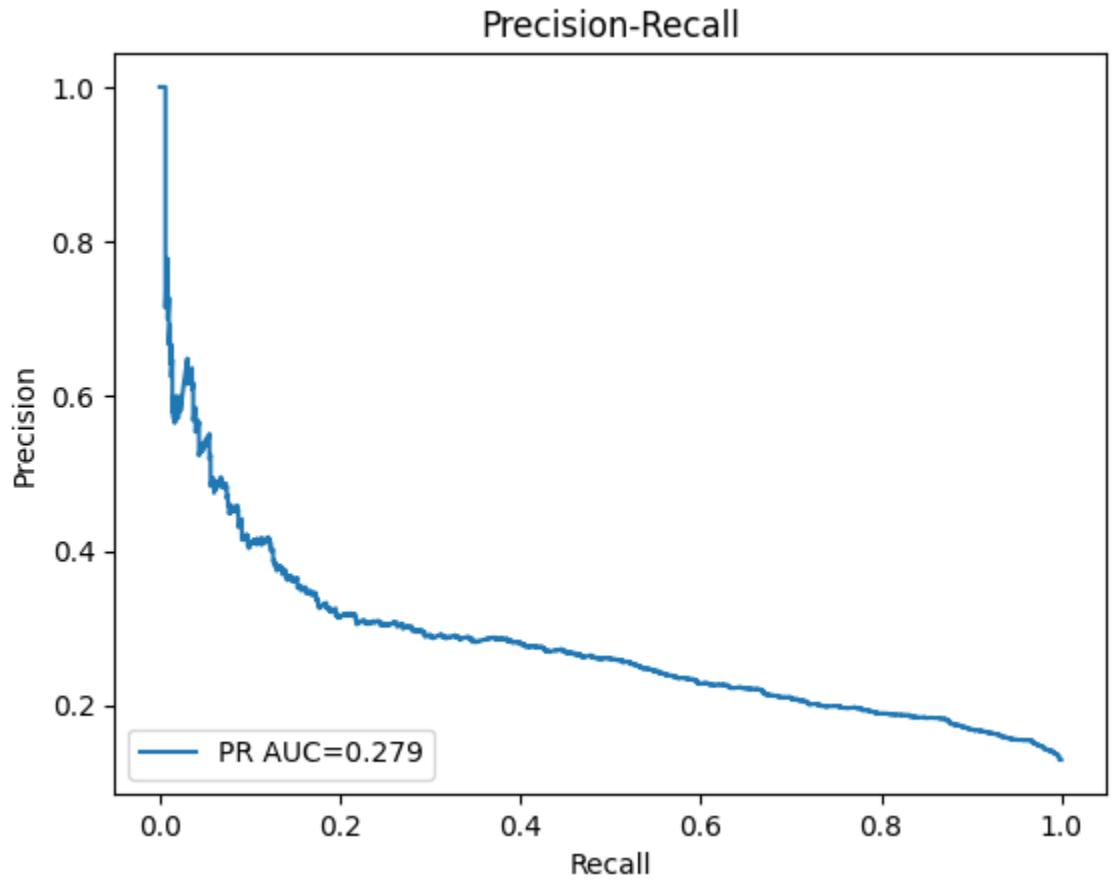
Curva ROC



La curva ROC muestra, qué tan bien el modelo distingue entre créditos buenos y los que tienen indicio de caer en mora según el umbral que uno vaya moviendo. El AUC quedó en 0.715, que l es un valor sólido, significa que en el 71.5% de las comparaciones entre un cliente moroso y uno no moroso, el modelo los ordena como debería. No es perfecto, pero se mantiene por encima de la diagonal, así que sí hay una capacidad discriminativa real, en temas de riesgo crediticio donde hay ruido, perfiles raross y de todo mezclado un AUC mayor a 0.70 está dentro de lo aceptable.

Figura 3

Curva Precisión–Recall (PR-AUC)

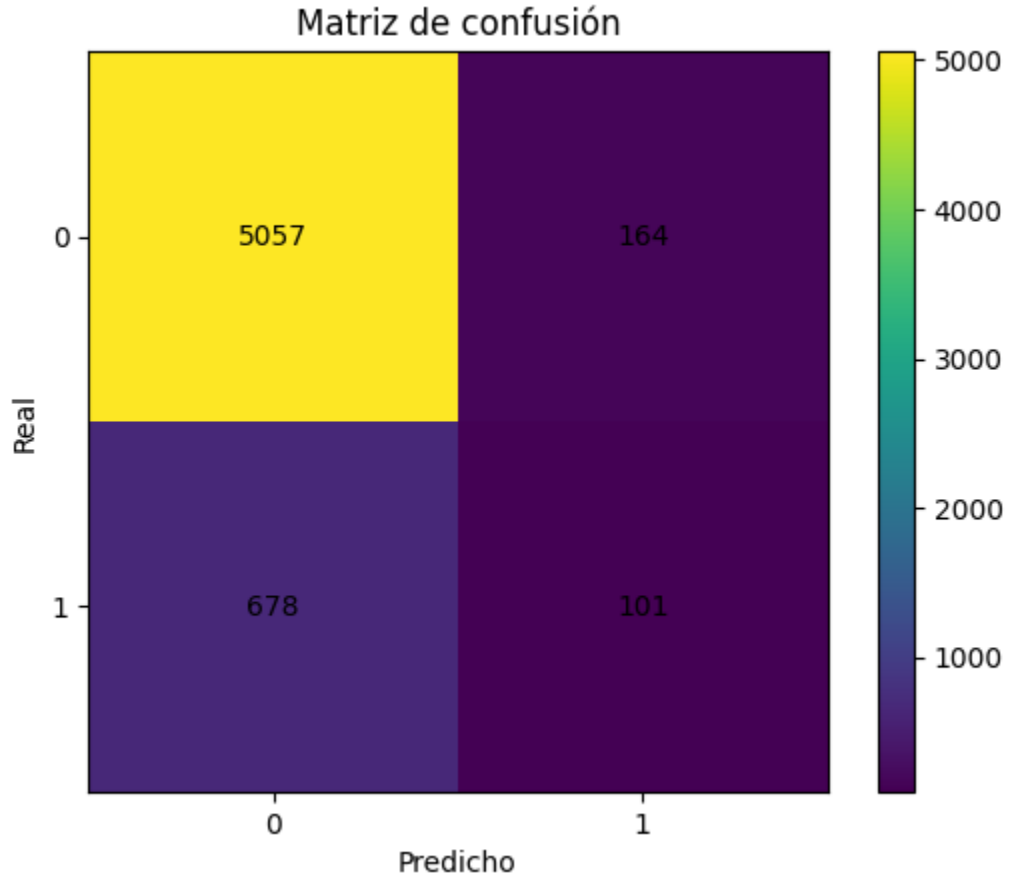


La curva PR ayuda a complementar todo porque acá el dataset está desequilibrado, la mayoría no cae en mora, eso ya se sabía, el PR-AUC salió en 0.279, que es razonable para este tipo de proporciones, se nota, además que la precisión es alta cuando el recall es bajo, sobre todo al inicio de la curva, eso quiere decir que cuando el modelo marca a alguien como riesgoso, casi siempre acertara, pero si uno empieza a tratar de econtrar más morosos subiendo el recall, la precisión cae.

Esto encaja con las métricas globales precisión 0.38 y recall 0.129 pues, refuerza el punto de que el umbral se tiene que escoger con cabeza fría según cuántos casos pueda revisar el analista del banco.

Figura 4

Matriz de Confusión



La matriz con el umbral operativo 0.50 muestra el desempeño ya directo:

TN: 5057

FP: 164

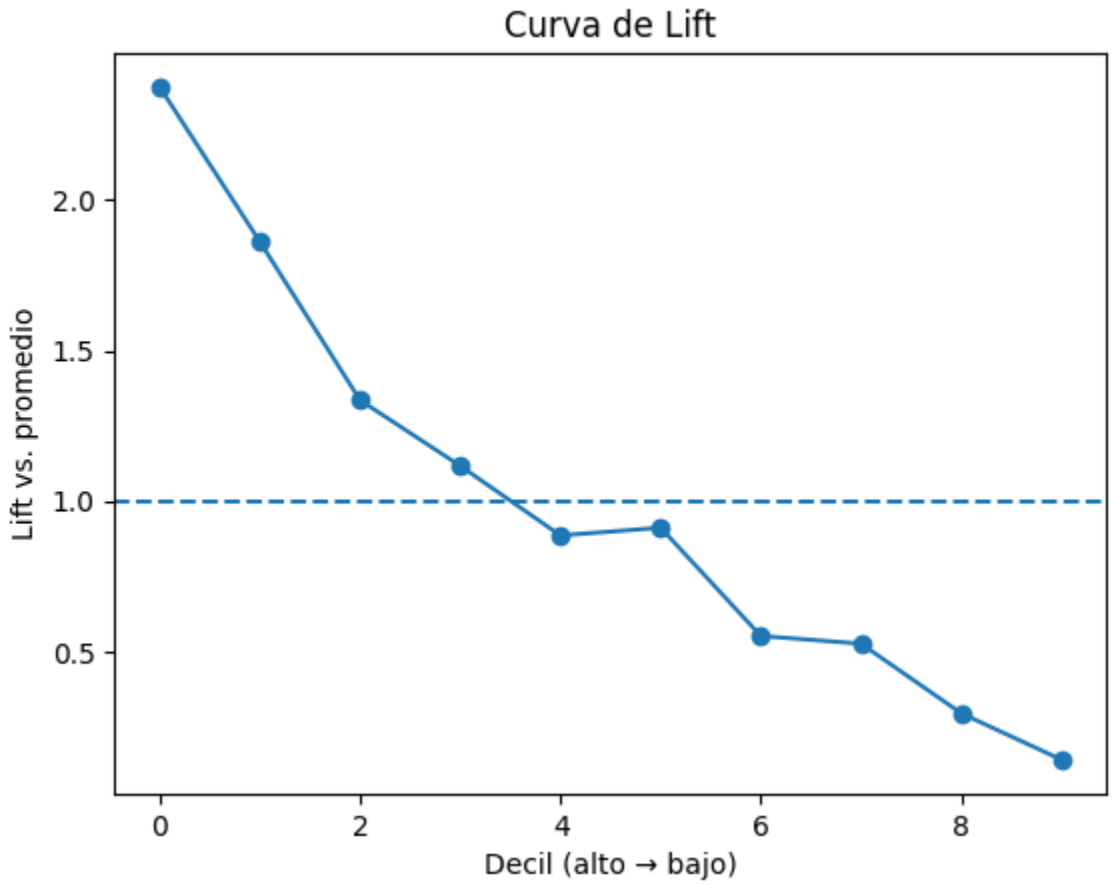
FN: 678

TP: 101

Los FN son los que más pesan , porque son clientes que sí cayeron en mora y el modelo no los vio venir, con un umbral conservador el recall queda bajito 12.9%, y eso se nota de inmediato, ahora, el lado positivo es que controla bastante bien los falsos positivos, el modelo es más cuidadoso levantando alertas. Dependiendo de la capacidad operativol por eso sugerimos el umbral 0.29 cuando haya que cubrir más casos.

Figura 5

Curva de Lift por Deciles



El gráfico de Lift deja ver qué tan bien prioriza el modelo, en el decil 0 el 10% de clientes con mayor score hay un Lift ≈ 2.3, en ese grupo la mora aparece 2.3 veces más que en el promedio, bastante bien.

Luego la curva va bajando como debe ser y ya en los deciles 6 a 9 cae por debajo de 1, lo cual es normal los primeros deciles son riesgo alto, los últimos no, eso lo confirma.

Este patrón es útil en operación, si el banco solo puede trabajar con el 20% de los casos del día, enfocarse en los primeros deciles hace que prácticamente se capture el doble de morosos con el mismo esfuerzo.

Inferencia y Producción

Se implementó un módulo de scoring (inference.py) para procesar nuevos lotes de créditos.

```
y_score = model.predict_proba(df)[:,1]
y_pred = (y_score >= threshold).astype(int)
out.to_csv(output_csv, index=False)
```

Este flujo, básicamente deja el modelo listo para usar en operación. Al final genera un predicciones.csv que ya sale tal cual, para los gestores de crédito o las mesas de riesgo, cumple justo lo que se necesita.

Conclusiones

Desarrollar este modelo fue más que solo entrenar algoritmos. Fue entender de fondo cómo los datos pueden anticipar comportamientos que normalmente detectamos tarde. El objetivo siempre fue prevenir, no solo clasificar y creo que eso se refleja en la estructura final un flujo completo que va desde la limpieza de la información hasta un resultado práctico y accionable para la operación.

El modelo no busca reemplazar lo que ya hacen las áreas de riesgo o credito, sino complementar desde otro ángulo. Al final, la mora no siempre es solo un tema de capacidad de pago, también hay factores de comportamiento o incluso señales de posible fraude que se mezclan. Por eso el enfoque preventivo tiene sentido.

Desde lo técnico, se logró montar un pipeline robusto, los datos pasan por validación, imputación, winsorización, codificación categórica y calibración de probabilidades, todo bajo un entorno controlado. Las métricas obtenidas AUC 0.715, precisión 0.38, recall 0.13 no son las más altas, pero demuestran que el modelo sí discrimina y ordena los casos con criterio, lo cual es un avance. Con más variables y un ciclo de reentrenamiento periódico, el rendimiento podría mejorar bastante.

Lo más valioso es que el proceso quedó automatizado y trazable. Cada corrida genera sus artefactos (best_model.pkl, preprocessor.pkl) con huellas de integridad y dependencias fijas, asegurando que el resultado sea reproducible. Eso permite que cualquier scoring futuro use exactamente el mismo modelo sin riesgos de inconsistencia.

La verdad es que el flujo quedó bastante sencillo, sin tanto enredo técnico ni pasos innecesarios. Entra un archivo nuevo, el sistema lo revisa, limpia lo que tenga que limpiar,

transforma las variables y suelta un CSV con todo calculado. Puede sonar básico, pero cuando uno lo ve funcionando se da cuenta de que ahorra bastante tiempo.

Antes tocaba hacer eso casi que manual, línea por línea y pues además del riesgo de error, ahora todo se hace en segundos. Eso deja el espacio libre para lo mirar los casos con calma, analizar por qué salen con riesgo, si tienen algo raro o si de pronto vale la pena revisarlos más a fondo.

Este proyecto demuestra que con una buena estructuración técnica y algo de curiosidad se pueden generar soluciones internas sólidas. No es un modelo final ni perfecto, pero sí una base clara para construir algo más grande.

Recomendaciones

Reentrenamiento periódico la idea es actualizar el modelo más o menos cada tres meses, o antes si el KS o los deciles empiezan a mostrar drift raro. No tiene ciencia, pero sí toca estar pendiente porque estas cosas se desajustan rápido.

Enriquecimiento de variables acá , hay bastante por hacer, se podrían meter datos transaccionales, comportamiento digital, validaciones biométricas, verificaciones de identidad, todo lo que pueda mejorar la señal sin meterse con PII.

Ajuste dinámico de umbral no hay un único corte perfecto, cuando se necesite más recall, el 0.29 funciona mejor, si la operación está apretada y no pueden revisar tanto caso, se deja el 0.50. es flexible.

Explainability integrar SHAP o Permutation Importance sería útil, sobre todo para auditores o para los comités de riesgo que siempre quieren ver el por qué detrás de cada decisión.

Gobernanza versionar todo, los artefactos (.pkl), las métricas, las dependencias (requirements.txt). Que quede rastro de cada cambio, básicamente, para mantener una auditoría continua del comportamiento del modelo.

Referencias

Aguilar Antonio, J. M. (2021). Retos y oportunidades en materia de ciberseguridad de América Latina frente al contexto global de ciberamenazas a la seguridad nacional y política exterior. *Estudios Internacionales*, 53(198).

Ben Jabeur, S., Stef, N., & Carmona, P. (2023). Bankruptcy Prediction using the XGBoost Algorithm and Variable Importance Feature Engineering. *Computational Economics*, 61(2). Borrero-Tigreros, D., & Bedoya-Leiva, O. (2020).

Predicción de riesgo crediticio en Colombia usando técnicas de inteligencia artificial. *Revista UIS Ingenierías*, 19(4).

Budholiya, K., Shrivastava, S. K., & Sharma, V. (2022). An optimized XGBoost based diagnostic system for effective prediction of heart disease. *Journal of King Saud University - Computer and Information Sciences*, 34(7).

Carrión-Barco, G., Sánchez-Chero, M.-J., del Castillo Castro, C. I., Campos Flores, F. W., & Timaná Alvarez, M. (2021). Modelo de seguridad informática para un medio de conexión pública. *Revista de La Universidad Del Zulia*, 12(32).

Chakri, P., Pratap, S., Lakshay, & Gouda, S. K. (2023). An exploratory data analysis approach for analyzing financial accounting data using machine learning. *Decision Analytics Journal*, 7.

Congreso de Colombia. (2008). *Ley 1266 de 2008*. Obtenido de funcion publica: https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=34488

Congreso de Colombia. (2009). *Ley 1328 de 2009*. Obtenido de funcion publica: https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=36841

Congreso de Colombia. (2012). *Ley 1581 de 2012*. Obtenido de funcion publica: https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981

De la Rosa Rodríguez, P. I. (2021). Aplicaciones educativas digitales y la falta de seguridad de los datos personales de sus usuarios. *RIDE Revista Iberoamericana Para La 19 Investigación y El Desarrollo Educativo*, 12(23).

Espinosa Zúñiga, J. J. (2020). Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito. Ingeniería *Investigación y Tecnología*, 21(3).

Fierro Torres, C. Á., Castillo Pérez, V. H., & Torres Saucedo, C. I. (2022). Análisis comparativo de modelos tradicionales y modernos para pronóstico de la demanda: enfoques y características. *RIDE Revista Iberoamericana Para La Investigación y El Desarrollo Educativo*, 12(24).

Gbadebo, M. O. (2025). Integrating Post-Quantum Cryptography and Advanced Encryption Standards to Safeguard Sensitive Financial Records from Emerging Cyber Threats. *Asian Journal of Research in Computer Science*, 18(4), 1–23.

Gil-Vera, V. D., & Quintero-López, C. (2021). Predicción del rendimiento académico estudiantil con redes neuronales artificiales. *Información Tecnológica*, 32(6).

Gutierrez Portela, F., Rodríguez Cárdenas, S., Patiño Ospina, L. P., & Hernandez Aros, L. (2023). Estudio de la prevención y detección de fraudes financieros a través de técnicas de aprendizaje automático. *CAFI*, 6(1).

Milián Gómez, J. F., & Rodríguez Corría, R. (2021). Aproximaciones a la contratación bancaria en relación con las cooperativas agropecuarias en Cuba: su incidencia en la seguridad alimentaria. *Boletín de La Asociación Internacional de Derecho Cooperativo*, 59.

Patino Orozco, G. A. (2021). Una comparativa de los esquemas de ciberseguridad de China y Estados Unidos. *OASIS*, 34. 20.

Quirumbay Yagual, D. I., Castillo Yagual, C., & Coronel Suárez, I. (2022). Una revisión del Aprendizaje profundo aplicado a la ciberseguridad. *Revista Científica y Tecnológica UPSE*, 9(1).

Rao, C., Liu, Y., & Goh, M. (2023). Credit risk assessment mechanism of personal auto loan based on PSO-XGBoost Model. *Complex and Intelligent Systems*, 9(2).

Sahoo, K., Samal, A. K., Pramanik, J., & Pani, S. K. (2019). Exploratory data analysis using python. *International Journal of Innovative Technology and Exploring Engineering*, 8(12).

Su, W., Jiang, F., Shi, C., Wu, D., Liu, L., Li, S., Yuan, Y., & Shi, J. (2023). An XGBoostBased Knowledge Tracing Model. *International Journal of Computational Intelligence Systems*, 16(1).

Suazo Galdames, I. (2023). *Inteligencia artificial en investigación científica*. SciComm Report.

Superintendencia Financiera de Colombia. (2017). *Circular Básica Jurídica*. Instrucciones relacionadas con la gestión del riesgo operativo y de fraude. Obtenido de Superintendencia Financiera de Colombia: https://www.superfinanciera.gov.co.

Superintendencia Financiera de Colombia. (2017). SARLAFT – *Sistema de Administración del Riesgo de Lavado de Activos y de la Financiación del Terrorismo*. Obetnido de Superintendencia Financiera de Colombia. https://www.superfinanciera.gov.co.

Viguri Cordero, J. A. (2021). Las normas ISO/IEC como mecanismos de responsabilidad proactiva en el Reglamento General de Protección de Datos. IDP. *Revista de Internet Derecho y Política*, 33.