



Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y
Tecnología

Máster en Inteligencia Artificial

Implementación de MLOps
para Small Data en Credit
Scoring

Trabajo Fin de Estudios

presentado por: Avgusta Averens

Dirigido por: José Manuel Picaza Garcia

Ciudad: El Casar

Fecha: 16 de julio de 2025

Índice de Contenidos

Resumen	v
Abstract	vi
1. Introducción	1
2. Contexto y Estado del Arte	3
2.1. Introducción	3
2.2. Credit scoring: fundamentos y evolución	4
2.3. Credit scoring con small data: síntesis, selección y robustez	5
2.4. Automatización y MLOps en el ciclo de vida del modelo	7
2.5. Explicabilidad (XAI) en credit scoring	11
2.6. Ética, discriminación y fairness en IA financiera	13
2.7. Aplicaciones reales y disrupción fintech	16
2.8. MytripleA: Financiación alternativa para empresas	18
2.9. Conclusiones del estado del arte	19
3. Identificación de Requisitos	22
3.1. Requisitos funcionales	22
3.2. Requisitos no funcionales	23
3.3. Requisitos técnicos	24
3.4. Requisitos de datos	24
3.5. Requisitos legales y éticos	25
3.6. Requisitos del entorno de ejecución	25
3.7. Requisitos de validación y evaluación	25
3.8. Requisitos organizativos y colaboración empresarial	26
4. Objetivos	27

4.1. Objetivo general	27
4.2. Desglose de los objetivos específicos	27
4.3. Alcance temporal y resultados esperados	29
5. Desarrollo del trabajo	30
5.1. Arquitectura general del sistema	30
5.2. Preparación del entorno	31
5.3. Recopilación y comprensión de los datos	33
5.4. Preprocesamiento de los datos	39
5.5. Generación de datos sintéticos	40
5.6. Diseño y entrenamiento de modelos	42
5.7. Explicabilidad del modelo	44
5.8. MLOps y automatización del ciclo de vida	49
5.9. Monitorización y detección de data drift	50
5.10. Validación técnica y funcional del sistema	51
5.11. Consideraciones éticas, legales y de negocio	53
5.12. Resumen del desarrollo y logros alcanzados	54
6. Conclusiones y trabajo futuro	57
6.1. Síntesis de resultados	57
6.2. Contribuciones del trabajo	57
6.3. Limitaciones encontradas	58
6.4. Líneas de trabajo futuro	58
6.5. Valor estratégico del proyecto	60
Referencias	60
A. Apendices	64

Índice de Ilustraciones

1.	10 etapas fundamentales según Testi et al. (2022)	8
2.	Arquitectura del sistema de credit scoring	30
3.	El histograma de frecuencias de <code>grade_code</code>	36
4.	Importancia de variables - XGBoost - SMOTE (F1=0.78)	43
5.	Curva ROC para el modelo MLP (AUC = 0.85)	44
6.	Importancia global de variables (summary plot) – MLP	45
7.	Impacto medio absoluto por variable – MLP	46
8.	Gráficos de dependencia SHAP para las principales variables – MLP	47
9.	Explicación individual con <code>force_plot</code> – MLP	48
10.	Los resultados del mejor modelo en MLflow	49
11.	Valores nulos por variable	65
12.	Distribución de variables numéricas	66
13.	Matriz de correlación	67
14.	Matriz de confusión del modelo MLP	68
15.	Trayectoria de decisiones (decision plot) – MLP	69

Índice de Tablas

1.	Contenido de la carpeta <code>tfm</code>	33
2.	Descripción de las variables del dataset	34
3.	F1-score ponderado por modelo y tipo de datos	55

Resumen

Nota: Este Trabajo de Fin de Máster aborda el problema del análisis de riesgo crediticio en contextos de datos limitados, una situación común en fintechs como la empresa colaboradora, MytripleA. Se propone un pipeline MLOps modular y reproducible que integra técnicas de aprendizaje automático, generación de datos sintéticos y herramientas de explicabilidad. La metodología incluye la recopilación y el procesamiento de datos reales, el entrenamiento de múltiples modelos de clasificación supervisada (Regresión Logística, Árbol de Decisión, Random Forest, XGBoost, Gradient Boosting, SVM, KNN, Naive Bayes y MLP), y la evaluación de cinco escenarios distintos: datos originales, SMOTE, CTGAN, CTGAN+SMOTE y SMOTE+CTGAN.

Se ha comprobado que la combinación de datos sintéticos generados con CTGAN y balanceados con SMOTE ha ofrecido los mejores resultados. En particular, el modelo MLP entrenado con esta configuración ha alcanzado un F1-score de 0.83, posicionándose como la mejor solución global. También se han identificado mejoras relevantes con SMOTE de forma aislada, mientras que CTGAN por sí solo ha mostrado un rendimiento inferior.

Estos resultados confirman que la integración de técnicas generativas y de sobremuestreo permite construir sistemas robustos, explicables y efectivos incluso en entornos de small data. El pipeline propuesto se ha implementado con herramientas como MLflow y Docker, y está preparado para su integración en procesos reales de evaluación crediticia. Los archivos necesarios para la reproducción de los experimentos se encuentran disponibles en el repositorio de GitHub: <https://github.com/AverensAi/tfm>.

Palabras Clave: credit scoring, small data, MLOps, datos sintéticos, explicabilidad

Abstract

Nota: This Master’s Thesis addresses the challenge of credit risk assessment in data-limited environments, a common scenario in fintech companies such as the project collaborator, MytripleA. A modular and reproducible MLOps pipeline is proposed, integrating machine learning techniques, synthetic data generation, and explainability tools. The methodology includes the collection and preprocessing of real-world data, training of multiple supervised classification models (Logistic Regression, Decision Tree, Random Forest, XGBoost, Gradient Boosting, SVM, KNN, Naive Bayes, and MLP), and evaluation under five distinct scenarios: original data, SMOTE, CTGAN, CTGAN+SMOTE, and SMOTE+CTGAN.

The combination of CTGAN-generated data with SMOTE balancing has delivered the best overall results. Specifically, the MLP model trained on this dataset achieved an F1-score of 0.83, outperforming all other configurations. Significant improvements have also been observed using SMOTE alone, while CTGAN by itself yielded lower performance.

These findings confirm that blending generative and oversampling techniques enables the construction of robust, explainable, and effective models even in small data contexts. The pipeline has been implemented using tools like MLflow and Docker, and is ready for integration into real-world credit scoring workflows. The files required to reproduce the experiments are available in the GitHub repository: <https://github.com/AverensAi/tfm>.

Keywords: credit scoring, small data, MLOps, synthetic data, explainability

1. Introducción

En el contexto actual de transformación digital, el uso de técnicas de inteligencia artificial (IA) se ha convertido en un pilar fundamental para mejorar los procesos de toma de decisiones en el sector financiero. Uno de los ámbitos donde esta transformación está teniendo un impacto significativo es el análisis de riesgo crediticio, o credit scoring, que permite evaluar la probabilidad de que un solicitante (persona o entidad) incumpla sus obligaciones financieras. Normalmente este proceso se ha basado en modelos estadísticos aplicados a grandes volúmenes de datos históricos. Sin embargo, en escenarios donde los datos son limitados o incompletos - como sucede frecuentemente en fintechs o en operaciones con pequeñas y medianas empresas (PYMEs) -, estos enfoques pierden efectividad.

El problema del small data en el credit scoring dificulta la construcción de modelos robustos y compromete la equidad, la precisión y la adaptabilidad de los sistemas. Las instituciones que operan con bases de datos pequeñas, fragmentadas o no estandarizadas deben enfrentar decisiones críticas con recursos limitados. En este contexto, es necesario desarrollar nuevas estrategias que permitan mejorar la calidad y riqueza de los datos disponibles, así como aplicar técnicas de machine learning de manera eficiente, escalable y explicable.

Una solución interesante es la incorporación de prácticas de MLOps (Machine Learning Operations), una disciplina que busca integrar el desarrollo, la gestión y el mantenimiento de modelos de aprendizaje automático en entornos productivos. MLOps permite automatizar flujos de trabajo, versionar modelos y datos, monitorear su rendimiento, y garantizar la trazabilidad y reproducibilidad del sistema. A través de estas prácticas, es posible crear sistemas modulares y eficientes que se adapten incluso a entornos con recursos limitados.

Además, este trabajo incorpora un componente novedoso: el tratamiento avanzado de datos en escenarios de small data, utilizando técnicas de síntesis y enriquecimiento para compensar la escasez de información. El uso de datos sintéticos generados con algoritmos como SMOTE o CTGAN, así como la ingeniería de características basada en modelos interpretables (como SHAP o LIME), permite aumentar la capacidad predictiva de los modelos y garantizar una mayor transparencia en los procesos de decisión.

Este Trabajo de Fin de Máster se realiza en la colaboración con la empresa MyTripleA, una fintech española especializada en servicios de factoring y confirming. En este entorno

real, se diseñará un pipeline de MLOps orientado a escenarios de small data. El objetivo es demostrar cómo una arquitectura modular, reproducible y adaptada al dominio financiero puede mejorar el análisis de riesgo crediticio, no sólo en términos de precisión, eficiencia operativa y explicabilidad.

En resumen, esta investigación aborda un problema de gran relevancia práctica — la gestión del credit scoring en contextos de datos reducidos — y propone una solución innovadora basada en técnicas modernas de inteligencia artificial y metodologías de MLOps. A través de un piloto aplicado en un entorno real, se busca no sólo validar la eficacia del enfoque propuesto, sino también generar un marco reutilizable que pueda ser escalado en el sector fintech.

2. Contexto y Estado del Arte

2.1. Introducción

El credit scoring ha sido, durante décadas, una de las principales herramientas empleadas por entidades financieras para evaluar el riesgo asociado a clientes, tanto individuales como corporativos. Tradicionalmente, este proceso se ha basado en técnicas estadísticas como la regresión logística, el análisis discriminante o los modelos de puntuación heurística, debido a su simplicidad, interpretabilidad y facilidad de implementación. Estas técnicas han permitido establecer mecanismos de toma de decisiones más estructurados, eficientes y, en apariencia, objetivos.

Sin embargo, el contexto financiero actual ha cambiado drásticamente. El crecimiento del sector fintech, el acceso a nuevas fuentes de datos (como datos transaccionales en tiempo real, comportamiento online o redes sociales), y la necesidad de evaluar perfiles sin historial crediticio amplio, han desafiado las metodologías tradicionales. Además, la creciente disponibilidad de datos no estructurados, la necesidad de adaptabilidad frente a mercados dinámicos, y las exigencias regulatorias relacionadas con la transparencia y la no discriminación han impulsado la evolución de estos modelos hacia enfoques más sofisticados.

En este nuevo escenario, el aprendizaje automático (machine learning) y la inteligencia artificial han emergido como herramientas clave para mejorar la precisión, escalabilidad y personalización del credit scoring. Los algoritmos modernos permiten modelar relaciones complejas entre variables, detectar patrones ocultos y adaptar los sistemas a entornos cambiantes. No obstante, su implementación conlleva nuevos desafíos: la necesidad de explicar las decisiones automatizadas, asegurar la equidad del modelo y garantizar su mantenimiento y actualización en entornos productivos reales.

Este apartado analiza el estado actual del conocimiento en credit scoring, con especial énfasis en tres dimensiones que constituyen el núcleo de este Trabajo de Fin de Máster:

1. Escenarios con datos reducidos (small data), donde la escasez de observaciones impone restricciones a la generalización de los modelos y requiere enfoques específicos de enriquecimiento y validación;

2. Operacionalización mediante MLOps, un enfoque que permite automatizar, versionar, desplegar y monitorizar modelos de forma sistemática y sostenible, especialmente relevante en sectores regulados como el financiero;
3. Explicabilidad de modelos (Explainable AI – XAI), necesaria para comprender y justificar las decisiones de los sistemas inteligentes, garantizar la confianza de los usuarios y cumplir con los marcos normativos y éticos vigentes.

El estudio de estas dimensiones no solo es relevante desde el punto de vista técnico, sino también estratégico, ya que contribuye a la construcción de soluciones responsables, reproducibles y alineadas con las exigencias reales del sector financiero actual.

2.2. Credit scoring: fundamentos y evolución

El credit scoring moderno nace con los primeros modelos de regresión logística y análisis discriminante que FICO populariza en los años 80, cuando los bancos comienzan a sustituir la decisión experta individual por sistemas automatizados y reproducibles. El libro *Credit Scoring and Its Applications* (Thomas, Edelman, y Crook, 2002) recoge los hitos de esta primera etapa: uso de variables estrictamente financieras, énfasis en interpretabilidad y adopción masiva tras la aprobación de la Equal Credit Opportunity Act. Estos modelos clásicos destacan por su robustez y claridad, pero su dependencia de datos limpios y balanceados limita la eficacia cuando aparecen ruido, clases desequilibradas o variables categóricas complejas (Abdou y Pointon, 2011).

Los retos de la era fintech — volatilidad del comportamiento y nuevas fuentes de información — impulsan técnicas de reducción supervisada de la dimensionalidad como DCOV-SDR (Miljkovic y Wang, 2025), que seleccionan combinaciones de predictores maximizando la correlación de distancia con la variable objetivo y preservan interpretabilidad incluso con muchos atributos categóricos. Al reducir el número de variables relevantes se ha probado una mejora simultánea de rendimiento y explicabilidad, algo crítico en escenarios con small data.

Paralelamente, el auge del big data introduce un giro radical: empresas emergentes asumen que “todo dato es dato crediticio”. Hurley y Adebayo documentan cómo miles de puntos de comportamiento en línea, ubicación o redes sociales se incorporan a nuevos puntajes, aumentando la cobertura de clientes sin historial pero también los riesgos de

opacidad, sesgo por proxy y discriminación algorítmica (Hurley y Adebayo, 2016). Su trabajo subraya la necesidad de marcos regulatorios que garanticen transparencia, corrección de errores y auditorías de equidad.

La investigación académica responde con algoritmos avanzados: gradient boosting, random forests y, más recientemente, deep learning. Sin embargo, un metaanálisis de Gunnarsson et al. demuestra que, para credit scoring, los modelos profundos (MLP, deep belief networks) no superan a ensambles como XGBoost, mientras implican mayor costo computacional; XGBoost se mantiene como la referencia cuando el objetivo prioritario es la precisión predictiva (Gunnarsson, Vanden Broucke, Baesens, Óskarsdóttir, y Lemahieu, 2021). Estas conclusiones concuerdan con estudios regulatorios que recomiendan equilibrar rendimiento y explicabilidad.

Más allá de la exactitud, la revisión exhaustiva de Sadok et al. muestra que la inteligencia artificial favorece la inclusión financiera al aprovechar fuentes alternativas de datos, pero plantea dilemas éticos sobre privacidad y sesgos sistémicos (Sadok, Sakka, y El Maknoui, 2022). Los autores proponen certificar algoritmos y datos para asegurar estándares de exactitud, transparencia y no discriminación.

En síntesis, la evolución del credit scoring transita de modelos estadísticos lineales a soluciones basadas en machine learning y big data, mientras se refuerza la demanda de MLOps y XAI para gestionar complejidad, trazabilidad y responsabilidad social en ambientes de small data y alta regulación.

2.3. Credit scoring con small data: síntesis, selección y robustez

Un reto curioso y creciente en el ámbito del credit scoring es su aplicación en escenarios donde no se dispone de grandes volúmenes de datos, lo cual es habitual en PYMEs, nuevos clientes o fintechs emergentes. Como señalan Kitchin y Lauriault, incluso en la era del big data, los conjuntos de datos pequeños siguen siendo relevantes para responder preguntas específicas, y su valor aumenta cuando se conectan, documentan y reutilizan correctamente (Kitchin y Lauriault, 2015). Estos autores también advierten de los riesgos de inferir patrones generales a partir de muestras limitadas, lo que es muy pertinente en entornos financieros altamente regulados y sensibles al sesgo.

En estos contextos, técnicas avanzadas de generación sintética de datos han resultado prometedoras para mejorar el rendimiento de los modelos. Xu et al. desarrollan CTGAN, una red generativa adversarial condicional optimizada para datos tabulares, que supera a modelos anteriores como TableGAN y TVAE al generar muestras realistas con estructuras complejas, incluso en clases minoritarias (Xu, Skoularidou, Cuesta-Infante, y Veeramachaneni, 2019). CTGAN aprende distribuciones complejas de los datos originales y genera observaciones sintéticas que mantienen las relaciones estadísticas entre variables, lo que la convierte en una técnica especialmente útil en escenarios con desbalance de clases.

De forma complementaria, Qin et al. proponen la integración de XGBoost con optimización por enjambre de partículas adaptativo (APSO), una técnica que mejora la precisión en datasets reducidos como German Credit o Lending Club (Qin y cols., 2021). A diferencia de métodos clásicos como GridSearch, APSO adapta dinámicamente su búsqueda de hiperparámetros, lo que permite obtener configuraciones óptimas con menos iteraciones y mayor eficacia. Además de generar y ajustar modelos, la reducción de variables innecesarias es fundamental para evitar sobreajuste en contextos small data.

Laborda y Ryoo comparan métodos filtro y wrapper, destacando que forward stepwise selection puede reducir hasta un 90 % de las variables sin pérdida de rendimiento. Este método construye el modelo de forma progresiva, añadiendo solo aquellas variables que mejoran la predicción de forma significativa (Laborda y Ryoo, 2021).

Qi y Luo profundizan en enfoques no supervisados y semi-supervisados como solución a la escasez de etiquetas (Qi y Luo, 2022). Su revisión muestra cómo técnicas como autoencoders, contrastive learning o regularización por consistencia han mejorado el rendimiento de modelos entrenados con pocos ejemplos etiquetados, apoyándose en grandes volúmenes de datos no anotados.

Por su parte, Sustersic et al. presentan una comparación de métodos tradicionales — como regresión logística y redes neuronales — en escenarios con datos limitados, destacando que incluso en ausencia de variables robustas es posible construir modelos competitivos mediante una selección adecuada de características y validación cruzada estratégica (Šušteršič, Mramor, y Zupan, 2009). Estos enfoques combinados permiten construir modelos más generalizables, eficientes y fiables, incluso cuando se dispone de pocas observaciones reales.

2.4. Automatización y MLOps en el ciclo de vida del modelo

En la última década, la gestión del ciclo de vida de los modelos de aprendizaje automático ha adquirido una importancia estratégica en aplicaciones industriales, especialmente en el sector financiero. La integración de metodologías MLOps — acrónimo de Machine Learning Operations — responde a la necesidad de estructurar, automatizar y gobernar todas las etapas implicadas en la creación, validación, despliegue y mantenimiento de modelos predictivos. MLOps se concibe como una extensión natural de las prácticas DevOps, incorporando además la especificidad que exige el trabajo con datos y modelos estadísticos. Su propósito es facilitar la trazabilidad, reproducibilidad y escalabilidad de sistemas inteligentes, contribuyendo a una mayor robustez operativa, cumplimiento normativo y alineación entre ciencia de datos e ingeniería de software.

Testi et al. proponen una taxonomía compuesta por diez etapas fundamentales que estructuran el ciclo de vida de los modelos desde una perspectiva MLOps: identificación del problema, recopilación de datos, análisis exploratorio, preprocesamiento, entrenamiento, evaluación, explicabilidad, despliegue, monitorización en producción y gestión del ciclo de vida (Testi y cols., 2022). Esta división permite identificar responsabilidades específicas, aplicar herramientas concretas en cada fase y establecer indicadores de calidad y control en puntos críticos del pipeline (Fig. 1).

La primera etapa, la identificación del problema, constituye el anclaje conceptual del ciclo. En este punto se definen los objetivos predictivos y las restricciones éticas, regulatorias y operativas que condicionarán el desarrollo posterior del sistema. En el sector financiero, esta etapa implica traducir problemáticas como la predicción de impagos, la detección de fraude o la evaluación de riesgo reputacional en casos de uso cuantificables. Brahmandam subraya la necesidad de incorporar desde el inicio requisitos vinculados al cumplimiento normativo, tales como la auditoría de decisiones, la gestión de explicaciones o la documentación de supuestos del modelo (Brahmandam, 2025). En contextos como el scoring crediticio, esta fase inicial debe prever los umbrales de aceptación de solicitudes, la sensibilidad frente a variables demográficas y la compatibilidad con políticas internas de riesgo.

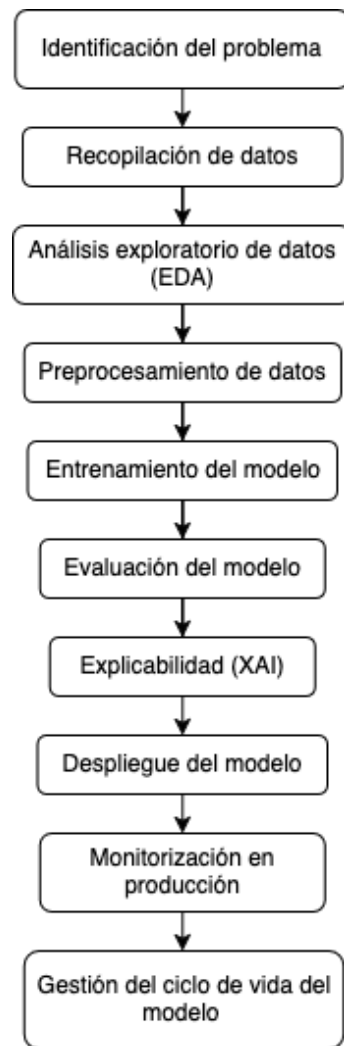


Figura 1: 10 etapas fundamentales según Testi et al. (2022)

La recopilación de datos, segunda etapa, requiere integrar múltiples fuentes de información, tanto internas como externas, garantizando la consistencia, integridad y actualizabilidad de los datos. Rella destaca la creciente relevancia de los enfoques que combinan MLOps con DataOps, promoviendo la creación de pipelines de ingesta orquestados con herramientas como Apache Airflow o Dagster (Rella, 2022). Asimismo, recomienda la implementación de catálogos de datos y feature stores, que permiten estandarizar variables reutilizables y reducir la duplicación de esfuerzos entre proyectos. En el caso del scoring para fintechs, donde los datos suelen ser escasos o incompletos, es habitual complementar la información transaccional con fuentes abiertas, registros mercantiles o datos sectoriales anonimizados.

La tercera fase, el análisis exploratorio de datos (EDA), no solo tiene un carácter

diagnóstico, sino también contractual. Las visualizaciones, perfiles estadísticos y análisis de nulos u outliers deben quedar documentados en artefactos versionables, que constituyan un acuerdo entre las áreas de datos, ingeniería y compliance. Salama et al. recomiendan la automatización del EDA mediante notebooks parametrizados o librerías como Great Expectations, que permiten definir suites de validación y establecer alarmas en caso de desviaciones frente a benchmarks históricos (Salama, Kazmierczak, y Schut, 2021).

El preprocesamiento, cuarta etapa, constituye uno de los focos de riesgo más habituales. Oluwaferanmi aboga por encapsular las transformaciones en componentes reutilizables, preferiblemente implementados en lenguajes declarativos o frameworks de pipelines como TFX, lo que permite testear individualmente cada paso y controlar su trazabilidad (Oluwaferanmi, s.f.). En contextos de scoring, estas transformaciones suelen incluir codificaciones categóricas, escalados, imputaciones y creación de variables derivadas, todas ellas susceptibles de introducir sesgos si no se controlan adecuadamente.

La etapa de entrenamiento del modelo introduce la lógica predictiva en el sistema. En esta fase, los algoritmos seleccionados son ajustados a los datos de entrenamiento mediante procesos de optimización supervisada. Salama et al. proponen la integración de procesos de entrenamiento en sistemas de integración continua (CI), donde cada modificación del código o los datos pueda desencadenar automáticamente pruebas, validaciones y ajustes (Salama y cols., 2021). Brahmandam señala que en sectores como el bancario resulta cada vez más habitual implementar mecanismos de reentrenamiento continuo, especialmente en aplicaciones de detección de fraude, donde los patrones cambian rápidamente (Brahmandam, 2025). Esto exige arquitecturas capaces de escalar horizontalmente y controlar la sincronización de versiones.

La evaluación de los modelos es clave para garantizar que su rendimiento no solo es estadísticamente sólido, sino también operativo y equitativo. Singla destaca la necesidad de introducir umbrales de aceptación que incluyan métricas específicas del negocio (como el coste de falsos positivos o el impacto sobre tasas de aceptación crediticia) y métricas de equidad, como la paridad de resultados por grupos demográficos (Singla, 2023). También propone el uso de pruebas comparativas entre versiones (A/B testing) y la aplicación de barreras de calidad que bloqueen el despliegue si el rendimiento empeora.

La explicabilidad, aunque conceptualmente posterior, debe planificarse desde el inicio. En entornos regulados como la banca, los modelos deben ser comprensibles tanto para

auditores como para usuarios finales. La guía de MLOps en Google Cloud recomienda integrar técnicas como SHAP o LIME directamente en los pipelines, de modo que cada predicción pueda ir acompañada de una explicación replicable. Brahmandam enfatiza que las auditorías exigen que cada versión del modelo esté asociada a su correspondiente versión explicativa, y que los criterios usados en producción sean transparentes y auditables (Brahmandam, 2025).

El despliegue del modelo, octava fase, representa la transición del entorno de validación al entorno operativo. Aquí es fundamental minimizar riesgos mediante estrategias como el canary deployment, que permite introducir nuevas versiones de forma controlada y reversible. Rella argumenta que la combinación de estrategias GitOps (por ejemplo, ArgoCD) con herramientas de control de versiones como DVC y MLflow permite mantener un control riguroso sobre qué modelo se encuentra activo, en qué entorno y bajo qué configuración (Rella, 2022). En aplicaciones batch, el uso de contenedores Docker garantiza que las dependencias del modelo se mantengan constantes en distintos entornos.

La monitorización en producción constituye un mecanismo continuo de supervisión del comportamiento del modelo una vez desplegado. Ackerman et al. proponen un auditor estadístico que rastrea la distribución de las probabilidades predichas por el modelo para detectar cambios significativos en su comportamiento (Ackerman y cols., 2021). Este enfoque permite detectar data drift incluso sin disponer de etiquetas actualizadas, lo cual resulta crítico en el scoring crediticio, donde la variable dependiente — por ejemplo, el impago — puede tardar meses en manifestarse. Herramientas como Evidently AI y WhyLabs han implementado estos mecanismos en combinación con sistemas de monitoreo como Prometheus y Grafana.

Finalmente, la gestión del ciclo de vida implica el versionado, mantenimiento y auditoría del sistema a lo largo del tiempo. Esto incluye el versionado no solo del modelo, sino también del código, los datos, los pipelines y las explicaciones. Oluwaferanmi resalta la importancia de utilizar sistemas como MLflow y DVC para permitir la trazabilidad completa de cada experimento, facilitando auditorías y asegurando la reproducibilidad (Oluwaferanmi, s.f.). Singla identifica como principales barreras la falta de perfiles técnicos capacitados en estas herramientas, así como la resistencia cultural a adoptar prácticas colaborativas propias del software moderno (Singla, 2023). Para superar estos obstáculos, recomienda la creación de equipos multidisciplinares y planes de formación continua.

En resumen, el enfoque MLOps ofrece un marco robusto y escalable para la gestión del ciclo de vida de modelos en sectores exigentes como el financiero. Su correcta implementación permite mitigar riesgos operativos, garantizar el cumplimiento regulatorio, y mantener la eficiencia y calidad de los sistemas predictivos en producción.

2.5. Explicabilidad (XAI) en credit scoring

La demanda de transparencia en los sistemas de decisión automática ha adquirido un carácter prioritario en el ámbito financiero, donde cada predicción puede afectar de forma directa la concesión o denegación de un crédito. La literatura especializada coincide en que la explicabilidad no es una tarea puntual ejecutada al final del desarrollo, sino un requisito transversal que permea todas las fases del ciclo de vida del modelo. Misheva et al. examinan comparativamente LIME y SHAP sobre algoritmos de alto rendimiento, como Random Forest y XGBoost, y concluyen que ambos marcos producen explicaciones coherentes con la lógica financiera subyacente; sin embargo, SHAP exhibe mayor granularidad y estabilidad al desglosar la contribución marginal de cada variable en términos de valores de Shapley, lo que facilita la auditoría retrospectiva de decisiones (Misheva, Osterrieder, Hirs, Kulkarni, y Lin, 2021). Ese hallazgo resulta especialmente relevante para entidades bancarias que requieren justificar ante reguladores el papel de cada predictor dentro del modelo crediticio.

La investigación de Dastile y Celik plantea un enfoque transformacional al convertir tablas financieras en representaciones binarias que pueden procesarse mediante redes neuronales convolucionales, habilitando el uso de herramientas visuales como Grad-CAM o saliency maps para localizar patrones determinantes en regiones específicas de la imagen derivada (Dastile y Celik, 2021). Esta metamorfosis de datos tabulares en dominios espaciales demuestra que la interpretabilidad visual no está restringida a datos de imagen per se, sino que puede adoptar formatos intuitivos incluso para modelos de riesgo crediticio basados en deep learning, incrementando la aceptación de los sistemas por parte de analistas y auditores sin sacrificar capacidad predictiva.

La convergencia entre exigencias normativas y técnicas XAI ha sido objeto de reflexión crítica por parte de Vale et al., quienes subrayan que los métodos post-hoc, a pesar de su utilidad práctica, no garantizan por sí mismos el cumplimiento regulatorio, pues se apoyan en aproximaciones que pueden divergir del funcionamiento interno real del modelo (Vale y cols., 2022). Su argumento enfatiza la necesidad de complementar las interpretaciones

algorítmicas con procesos de supervisión humana, métricas explícitas de equidad y documentación exhaustiva de las decisiones de diseño adoptadas durante la construcción del modelo. Esta visión encuentra eco en los principios formulados por el programa XAI de DARPA, iniciativa que promueve técnicas híbridas capaces de ofrecer justificaciones comprensibles sin comprometer la precisión de los algoritmos subyacentes (Gunning y Aha, 2019).

El estado del arte general sobre explicabilidad, sintetizado por Dwivedi et al., destaca la multiplicidad de métricas empleadas para evaluar la fidelidad de las explicaciones, como la robustez frente a perturbaciones o la coherencia con el conocimiento experto (Dwivedi y cols., 2023). Este estudio advierte que representaciones visuales atractivas — por ejemplo, mapas de calor — pueden inducir una falsa sensación de confianza si no se corrobora su validez mediante experimentos de sensibilidad, lo que resulta particularmente peligroso cuando el sistema se emplea para decisiones con impacto económico elevado. De manera complementaria, Das y Rad revisan los desafíos operativos que emergen al trasladar técnicas XAI desde entornos de laboratorio a contextos productivos, entre los que destacan el coste computacional de generar explicaciones en tiempo real y la ausencia de marcos estándar para comparar la calidad de distintas técnicas (Das y Rad, 2020). Su análisis señala la necesidad de protocolos de evaluación homogéneos que permitan a las instituciones financieras balancear transparencia y eficiencia.

Las revisiones centradas en credit scoring confirman esa necesidad. Heng y Subramanian, tras analizar más de cincuenta contribuciones recientes, observan que los estudios que reportan variables explicativas junto con métricas predictivas logran mayor aceptación por parte de los reguladores, quienes consideran indispensable la trazabilidad completa de la lógica de decisión (Heng y Subramanian, 2022). Asimismo, identifican una tendencia creciente a combinar interpretabilidad inherente — a través de modelos lineales o basados en reglas — con explicaciones post-hoc externas, buscando un balance óptimo entre desempeño y comprensión humana.

La explicabilidad no se limita a la fase de desarrollo. Una vez los modelos entran en producción, la monitorización de explicaciones se convierte en un indicador temprano de posibles desviaciones. Nallakaruppan et al. proponen un marco operativo que integra tableros dinamizados con valores SHAP distribucionales; ello permite detectar cambios sustantivos en la importancia relativa de las variables antes de que la métrica de precisión se

deteriore (Nallakaruppan, Balusamy, Shri, Malathi, y Bhattacharyya, 2024). Tal enfoque se alinea con las recomendaciones del programa DARPA-XAI, donde se enfatiza la supervisión continua de la “coherencia explicativa” como complemento de la supervisión estadística tradicional.

En cuanto a la selección de técnicas, la literatura señala convergencias claras. SHAP se consolida como el método de referencia para análisis global, gracias a su fundamento teórico en teoría cooperativa y su capacidad de descomponer la predicción en aportes aditivos exactos. LIME, en cambio, conserva una ventaja pragmática para explicaciones locales rápidas y fácilmente comunicables a usuarios finales. Las dos aproximaciones son, en muchos casos, complementarias y no excluyentes, tal y como ilustran Misheva et al. al aplicarlas simultáneamente para validar coherencia en modelos de árbol de decisión y gradiente reforzado (Misheva y cols., 2021).

La dimensión ética gana peso en todos los trabajos revisados. Vale et al. mencionan expresamente la obligación de documentar los supuestos del modelo, las variables descartadas, los procesos de balanceo y los criterios de gobernanza (Vale y cols., 2022). A su vez, Dwivedi et al. resaltan la necesidad de incorporar evaluaciones de fairness y counterfactual reasoning en la misma fase de validación, evitando relegar la auditoría ética a controles posteriores que resultan más costosos y menos eficaces (Dwivedi y cols., 2023).

En síntesis, la investigación contemporánea converge en la idea de que la explicabilidad debe integrarse de forma holística en el ciclo MLOps: guía la selección de variables, informa la validación, documenta la decisión final y actúa como señal temprana de desviaciones una vez el sistema está en producción. Las técnicas post-hoc, pese a sus limitaciones, se consolidan como instrumentos indispensables, pero solo alcanzan valor pleno cuando se combinan con vigilancia humana, métricas de equidad y protocolos normativos sólidos. El desafío inmediato para las instituciones financieras radica en operacionalizar estas buenas prácticas a escala, garantizando que la transparencia no sea un añadido cosmético, sino un pilar estructural que sustente la confianza del mercado y la legitimidad social de los modelos de credit scoring.

2.6. Ética, discriminación y fairness en IA financiera

La adopción acelerada de modelos de aprendizaje automático en el sector crediticio ha reavivado un debate histórico sobre la discriminación: las decisiones algorítmicas pueden

amplificar sesgos sistémicos o, por el contrario, constituir una palanca para la inclusión. Hurley y Adebayo advierten que la aparente neutralidad de un score se desmorona cuando las variables empleadas constituyen proxies de atributos protegidos, reeditando prácticas análogas al redlining bajo el barniz del big data (Hurley y Adebayo, 2016). Esta crítica se combina con la preocupación por la opacidad: modelos complejos dificultan a reguladores y clientes identificar la génesis de una denegación, lo que erosiona la confianza y expone a las entidades a litigios por violaciones del principio de trato equitativo.

La paradoja de la inclusión tecnológica queda patente en el trabajo de Faheem, quien muestra cómo el uso de datos alternativos — historiales de telefonía móvil, transacciones de plataformas fintech o interacciones en redes sociales — amplía el acceso al crédito de segmentos tradicionalmente excluidos, pero introduce nuevos vectores de riesgo legal (Faheem, 2021). Cuando la arquitectura de la decisión se apoya en fuentes no reguladas, surge la obligación de demostrar que la correlación observada con la probabilidad de impago no es, en realidad, un eco de sesgos socioeconómicos más profundos. En palabras del autor, la promesa de democratizar el crédito depende de garantizar la explicabilidad ex ante y de disponer de recursos humanos que supervisen los desajustes ex post.

La literatura reciente ha evolucionado desde el diagnóstico hacia metodologías formales de auditoría y reparación. Hurlin, Pérignon y Saurin introducen un protocolo estadístico que contrasta la hipótesis de paridad bajo distintos criterios — statistical parity, equal odds y conditional use accuracy — y propone los Fairness Partial Dependence Plots como herramienta para localizar las variables responsables del sesgo (Hurlin, Pérignon, y Saurin, 2024). Dicho enfoque permite negociar el compromiso entre precisión y equidad al neutralizar únicamente los predictores más problemáticos, evitando el sacrificio global de capacidad discriminatoria.

En la misma línea, Kozodoi, Jacob y Lessmann ofrecen una cartografía de procesadores de fairness — pre, in y post-processing — y cuantifican su efecto sobre la rentabilidad del score (Kozodoi, Jacob, y Lessmann, 2022). Sus experimentos con siete bases reales concluyen que es posible satisfacer simultáneamente varios criterios estadísticos con un impacto marginal en beneficios, siempre que se privilegie la métrica de separación como referencia operacional. Este resultado desafía la narrativa de que la corrección del sesgo necesariamente compromete la viabilidad económica y subraya la importancia de elegir medidas de equidad acordes con la estructura de costes del banco.

Las contribuciones de Trinh y Zhang enriquecen el panorama al articular un marco holístico que intercepta el sesgo en las tres fases del ciclo de vida: reponderación y augmentación de datos durante la ingesta, restricciones adversariales en el entrenamiento y calibración diferenciada en producción (Trinh y Zhang, 2024). Sus pruebas empíricas evidencian reducciones de disparidad de hasta un 45 % con pérdidas de exactitud inferiores al 7 %, lo que legitima la adopción de intervenciones combinadas como práctica de referencia. Destacan además la necesidad de métricas compuestas que integren paridad demográfica y estabilidad temporal, para evitar que la mitigación inicial se diluya con la deriva de datos.

Una cuestión previa al diseño de soluciones es la elección misma del concepto de fairness, dado que las definiciones existentes pueden entrar en conflicto lógico. El trabajo de Verma y Rubin sistematiza más de veinte nociones y demuestra, con el caso del German Credit, que un modelo puede ser simultáneamente justo e injusto según la métrica aplicada (Verma y Rubin, 2018). Este hallazgo impulsa a los reguladores a especificar explícitamente la definición exigible y a las entidades a sostener una trazabilidad documental que justifique la selección de criterios, so pena de incurrir en prácticas arbitrarias.

Desde la óptica regulatoria, el debate se ha desplazado hacia la exigencia de fair algorithms by design. La propuesta de Reglamento Europeo de IA clasifica los sistemas de credit scoring como de alto riesgo y exige pruebas de no discriminación como requisito previo al despliegue. Estudios como el de Kozodoi et al. muestran que la adaptación de los pipelines existentes a estos estándares no es necesariamente costosa, si incorpora herramientas de monitorización continua y reentrenamiento condicionado por alarmas de drift (Kozodoi y cols., 2022). De igual modo, las agencias supervisoras norteamericanas, amparadas en la ECOA, refuerzan los protocolos de auditoría ex post e insisten en conservar los scorecards interpretables para validar la coherencia entre la lógica estadística y las políticas de riesgo.

Se consolida así un consenso emergente: la equidad algorítmica no puede relegarse a un ajuste posterior ni fundarse en principios abstractos. Requiere marcos estadísticos capaces de identificar, medir y subsanar las diferencias de trato con respaldo documental; demanda la integración de mecanismos de control a lo largo de todo el ciclo MLOps; y obliga a ponderar el binomio rentabilidad-inclusión en función del contexto regulatorio y social. Lejos de constituir un obstáculo, la incorporación sistemática de fairness se perfila como un factor de resiliencia reputacional y de ventaja competitiva para las entidades que

la adoptan con rigor.

2.7. Aplicaciones reales y disrupción fintech

Las innovaciones analizadas en las secciones precedentes ya trascienden el plano académico y se manifiestan en soluciones productivas que remodelan el acceso al crédito y los servicios financieros. Bedoya-Builes et al. describen la expansión de herramientas de factoring y confirming dirigidas a microempresas de Medellín; su estudio confirma que estos instrumentos alivian la falta de liquidez, pero subraya que el bajo capital humano financiero de los empresarios limita la adopción plena de los productos (Bedoya-Builes, Cardona, y Zapata-Álvarez, 2024). De modo análogo, Youssef y Mansour comparan la operativa tradicional de BNP Paribas con la de Finexkap, plataforma que automatiza todo el flujo de factoring a través de API y verificación documental inmediata; los autores concluyen que la propuesta digital no solo reduce los plazos de desembolso, sino que flexibiliza las condiciones de financiación sin incrementar la morosidad (Youssef y Mansour, 2024). Ambos casos evidencian que la disrupción fintech parte de optimizar nichos infra-atendidos por la banca, aportando métricas de riesgo alternativas y flujos de trabajo sin fricción.

Los análisis sectoriales confirman la magnitud del fenómeno. Cao, Yang y Yu ofrecen una panorámica de los sub-verticales LendTech, PayTech y RiskTech, identificando la confluencia de deep learning, federated learning y privacidad diferencial como el núcleo tecnológico que posibilita el salto de eficiencia de la llamada Smart FinTech; además, caracterizan el paso de procesos batch a arquitecturas de aprendizaje continuo, requisito para sostener modelos válidos en mercados volátiles (Cao, Yang, y Yu, 2021). En la misma línea, Ashta y Herrmann destacan que la aceleración de fusiones y adquisiciones entre entidades tradicionales y startups responde a la necesidad de incorporar capacidades de IA sin incurrir en largos ciclos de desarrollo interno; sin embargo, advierten que los sesgos en los datos de entrenamiento y la opacidad algorítmica generan riesgos reputacionales que exigen gobernanza híbrida entre humanos y sistemas automáticos (Ashta y Herrmann, 2021).

La infraestructura sobre la que se sostienen estos servicios evoluciona hacia entornos mixtos de computación en la nube y cadenas de bloques. Lăzăroiu et al. muestran cómo los algoritmos de aprendizaje profundo incrustados en contratos inteligentes permiten verificar en tiempo real transacciones de pago y procesos de know-your-customer, reduciendo

fraudes y costes regulatorios; el artículo ilustra, además, que la integración de biometría comportamental y modelos de riesgo dinámicos incrementa la fidelidad de los usuarios al ofrecer experiencias de autenticación invisibles (Lăzăroiu y cols., 2023). De forma complementaria, Addy et al. sistematizan la evolución de los modelos de scoring: del predominio de la regresión logística se pasa a ensamblajes con gradient boosting, redes neuronales profundas e, incluso, arquitecturas auto-regresivas que ingieren fuentes de datos no tradicionales; su revisión constata que la incorporación de variables procedentes de redes sociales y registros de dispositivos móviles mejora la capacidad predictiva y la inclusión financiera, aunque agrava los retos de privacidad (Addy y cols., 2024).

El eje de inclusión reaparece en la propuesta de Mohanty, quien analiza la expansión de microcréditos respaldados por modelos de IA en mercados emergentes: las entidades utilizan señales de comportamiento digital — pagos de servicios públicos, actividad en plataformas de comercio electrónico — para estimar la probabilidad de impago en población sin historial crediticio formal; la autora subraya que la ventaja competitiva reside en la capacidad adaptativa de los modelos, capaces de reasignar ponderaciones conforme cambian las condiciones macroeconómicas, y advierte que la supervisión humana sigue siendo indispensable para detectar sesgos y adversarial attacks (Mohanty, 2025).

Más allá del crédito empresarial y minorista, la disrupción fintech se extiende a la gestión patrimonial, los seguros y los pagos transfronterizos. La literatura reciente identifica patrones comunes: desintermediación de la cadena de valor, monetización de datos masivos y orientación a la experiencia de usuario. Cao et al. apuntan que el ecosistema Smart FinTech evoluciona hacia plataformas middleware que orquestan microservicios de IA reutilizables entre distintos verticales (Cao y cols., 2021), mientras que Ashta y Herrmann destacan la presión regulatoria para armonizar requisitos de resiliencia operativa y protección del consumidor (Ashta y Herrmann, 2021). En América Latina y África, múltiples estudios de caso reportan que la flexibilidad de los modelos de negocio fintech facilita esquemas de buy now pay later y préstamos nano-credit para autónomos y gig-workers, lo que dinamiza el consumo y, a la vez, introduce nuevos vectores de riesgo sistémico.

En síntesis, la adopción de IA en fintech no solo optimiza la eficiencia operativa, sino que redefine los criterios de elegibilidad crediticia y amplía la frontera de la inclusión financiera. La evidencia empírica demuestra que los beneficios se materializan cuando las entidades combinan pipelines de datos robustos, modelos de riesgo transparentes y

controles de gobernanza que mitiguen sesgos y garanticen la protección de la información. Bajo estas condiciones, la disrupción fintech se consolida como un vector de transformación estructural de los sistemas financieros, con impacto directo en la competitividad y la estabilidad del sector.

2.8. MytripleA: Financiación alternativa para empresas

Este Trabajo de Fin de Máster se realiza en colaboración con la empresa española MytripleA, especializada en financiación alternativa para empresas a través de productos como factoring y confirming. La participación de la compañía proporciona un entorno real para validar los métodos propuestos y adaptar el proyecto a necesidades concretas del sector fintech. MytripleA es una fintech española fundada en 2013, con sedes en Golmayo (Soria) y Madrid. Se ha consolidado como un referente en el ámbito de la financiación alternativa para medianas y grandes empresas, ofreciendo soluciones como factoring, confirming, préstamos y servicios Buy Now, Pay Later (BNPL). Estas soluciones permiten a las empresas obtener liquidez de manera ágil y transparente, sin necesidad de consumir CIRBE ni contratar productos adicionales.

La plataforma opera 100 % online, facilitando la gestión automatizada de las operaciones financieras. Desde su inicio de operaciones en 2015, MytripleA ha financiado a más de 4000 compañías, superando los 550 millones de euros en volumen financiado. En 2024, contaba con más de 8500 inversores registrados, quienes han invertido un total de 141 millones de euros, obteniendo una rentabilidad media anual del 5 % en préstamos garantizados y del 8 % en facturas no garantizadas.

En marzo de 2025, MytripleA recibió una financiación de 30 millones de euros por parte de BBVA Spark, destinada a impulsar su crecimiento y el desarrollo de nuevos productos financieros. Este respaldo destaca la confianza en el modelo de negocio de la empresa y su papel en la transformación del sector financiero.

La colaboración con MytripleA en este TFM proporciona un contexto real y práctico para la aplicación de técnicas de inteligencia artificial en la evaluación del riesgo crediticio, especialmente en escenarios con datos limitados. La experiencia de la empresa en la automatización de procesos financieros y su enfoque en la transparencia y eficiencia la convierten en un caso de estudio relevante para el desarrollo de soluciones innovadoras en el ámbito del credit scoring.

2.9. Conclusiones del estado del arte

La revisión realizada confirma que el credit scoring atraviesa una etapa de renovación metodológica impulsada por la confluencia de avances algorítmicos, nuevas fuentes de datos y mayores exigencias regulatorias. Desde los primeros scorecards lineales descritos por Thomas et al. (Thomas y cols., 2002) hasta los ensamblajes no lineales de alto rendimiento, el campo ha evolucionado para atender una demanda simultánea de exactitud, transparencia y equidad. La literatura muestra que, aunque los modelos tradicionales siguen siendo una referencia en términos de interpretabilidad, su capacidad para capturar patrones complejos disminuye a medida que los datos se diversifican y se reducen los tamaños muestrales. Ello ha motivado la adopción de técnicas de generación sintética como CTGAN (Xu y cols., 2019) y métodos de sobremuestreo como SMOTE, cuya combinación ha demostrado mejorar la cobertura de clases minoritarias sin sacrificar la fidelidad estadística de los conjuntos de entrenamiento.

La problemática del small data persiste como uno de los principales obstáculos para la construcción de modelos robustos en fintechs y pymes. Investigaciones sobre síntesis de datos, selección de características y aprendizaje semi-supervisado (Kitchin y Lauriault, 2015; Qi y Luo, 2022) han proporcionado herramientas eficaces, pero también subrayan la necesidad de procesos de validación rigurosos que eviten la generación de artefactos o la amplificación de sesgos. En paralelo, la operacionalización de modelos mediante prácticas MLOps ha emergido como un requisito indispensable para sostener ciclos de vida largos y auditables. La taxonomía de diez etapas propuesta por Testi et al. (Testi y cols., 2022) y las guías operativas de Salama et al. (Salama y cols., 2021) muestran que la automatización de pruebas, el versionado de artefactos y la monitorización de drift no solo reducen costes operativos, sino que se convierten en salvaguardas contra desviaciones éticas y regulatorias.

La dimensión de explicabilidad ha pasado de ser un complemento voluntario a constituir un pilar central de la confianza algorítmica. Comparativas empíricas entre LIME y SHAP (Misheva y cols., 2021) evidencian que la granularidad y estabilidad de las aportaciones de SHAP facilitan la justificación de decisiones ante stakeholders y supervisores. Iniciativas como el programa XAI de DARPA (Gunning y Aha, 2019) y revisiones exhaustivas como la de Dwivedi et al. (Dwivedi y cols., 2023) subrayan que la métrica de fidelidad de la explicación debe incorporarse al conjunto de indicadores de calidad del modelo. La monitorización de explicaciones en producción, propuesta por Nallakaruppan

et al. (Nallakaruppan y cols., 2024), complementa la supervisión estadística tradicional y actúa como alerta temprana ante cambios de distribución que podrían comprometer la validez del sistema.

La preocupación por la discriminación algorítmica recorre de forma transversal toda la literatura reciente. Trabajos como el de Hurley y Adebayo (Hurley y Adebayo, 2016) ponen de relieve la facilidad con la que variables proxy pueden reproducir sesgos históricos. Investigaciones posteriores ofrecen metodologías de auditoría y mitigación: la cartografía de procesadores de fairness de Kozodoi et al. (Kozodoi y cols., 2022), los protocolos de paridad de Hurlin et al. (Hurlin y cols., 2024) y los marcos holísticos de Trinh y Zhang (Trinh y Zhang, 2024) muestran que es posible alcanzar compromisos aceptables entre rendimiento y equidad cuando se integran intervenciones en fase de datos, entrenamiento y calibración. No obstante, la revisión crítica de Verma y Rubin (Verma y Rubin, 2018) advierte que las definiciones de fairness pueden ser mutuamente excluyentes, lo que obliga a los reguladores a explicitar sus métricas preferidas y a los desarrolladores a documentar la lógica de selección de criterios.

Las aplicaciones reales en fintech ilustran la madurez de estas tendencias. Estudios de casos en factoring digital (Youssef y Mansour, 2024) y microcrédito alternativo (Bedoya-Builes y cols., 2024) prueban que los modelos basados en IA pueden reducir plazos de desembolso y ampliar la base de clientes, siempre que se acompañen de políticas de gobernanza de datos y supervisión humana. Revisiones sectoriales más amplias (Cao y cols., 2021; Ashta y Herrmann, 2021) señalan que la adopción de microservicios de IA, la federación de aprendizaje y la computación en la nube están reconfigurando la cadena de valor financiera, pero también generan nuevos retos de privacidad, resiliencia operativa y ciberseguridad.

En suma, el estado de arte confirma una transición hacia sistemas de credit scoring flexibles, explicables y socialmente responsables. Las soluciones basadas en MLOps, generación sintética, XAI, AutoML y fairness ofrecen un marco integral para afrontar los retos técnicos y éticos que plantea el entorno financiero contemporáneo. Este Trabajo de Fin de Máster se inscribe en dicha convergencia, al proponer un pipeline modular optimizado para small data que incorpora monitorización continua, mecanismos de explicabilidad y salvaguardas de equidad. Al situar la investigación en un contexto real — la fintech MytripleA — el proyecto aspira a demostrar que es posible alcanzar simultáneamente pre-

cisión, transparencia y sostenibilidad operativa en un sector sometido a alta regulación y evolución constante.

3. Identificación de Requisitos

Esta sección sintetiza los requisitos que orientan el diseño, la construcción y la puesta en producción del pipeline MLOps propuesto para la evaluación del riesgo crediticio en contextos de small data. La identificación exhaustiva de estos requisitos resulta imprescindible para garantizar que la solución final atienda tanto a los objetivos académicos del Trabajo Fin de Máster como a las necesidades operativas de la empresa colaboradora, My-TripleA. Los requisitos se agrupan en categorías funcionales, no funcionales, técnicas, de datos, legales y éticas, de entorno de ejecución, de validación y evaluación, y organizativas. Cada categoría se describe a continuación mediante un discurso narrativo que detalla su alcance, su justificación y sus implicaciones prácticas.

3.1. Requisitos funcionales

El primer grupo de requisitos aborda las capacidades que el sistema debe ofrecer al usuario final y a los equipos técnicos que lo mantendrán. En la fase inicial, el pipeline debe ser capaz de ingerir conjuntos de datos estructurados que contienen métricas financieras históricas de las empresas solicitantes. Este proceso de ingesta implica la detección automática de tipos de variables, la verificación de integridad referencial y la aplicación de reglas de limpieza predefinidas que corrijan valores atípicos y errores de formato. Tras la carga, el sistema debe realizar un preprocesamiento sistemático: conversión de variables categóricas, normalización de escalas, tratamiento de valores ausentes y generación de ahorros de espacio en disco mediante compresión o formatos columnares.

Una vez los datos se encuentran en un estado coherente, el pipeline tiene que ejecutar de manera desatendida distintos experimentos de aprendizaje automático. Se incluyen algoritmos de referencia — por ejemplo, regresión logística y árboles de decisión — y modelos más avanzados, como XGBoost o LightGBM, capaces de capturar interacciones no lineales relevantes en la predicción del default. El entrenamiento de cada modelo está orquestado por un motor de validación cruzada estratificada que redistribuye las observaciones de forma que se conserve la proporción original de clases en todos los pliegues; de esta manera, las métricas de rendimiento obtenidas reflejan de forma realista la capacidad de generalización del sistema.

En paralelo al proceso de entrenamiento, el pipeline debe ajustar de forma automática los hiperparámetros críticos de cada algoritmo. Para ello se recurre a estrategias de búsqueda eficientes — por ejemplo, búsqueda aleatoria guiada por early stopping o algoritmos de optimización bayesiana — que reducen el número de ejecuciones sin comprometer la exhaustividad. Una vez elegido el conjunto óptimo de hiperparámetros, el modelo resultante se somete a un módulo de explicabilidad que genera, para cada predicción, valores Shapley o explicaciones LIME. Estas explicaciones se guardan junto al modelo en un almacén central de artefactos, permitiendo su consulta posterior por personal de riesgo o por auditores externos.

Por último, el sistema debe exponer mecanismos de monitorización capaces de recopilar en tiempo casi real las métricas de performance y de advertir cuando se detecta data drift o concept drift. La detección precoz de desviaciones es crucial en entornos donde la realidad económica cambia con rapidez y las pérdidas ocasionadas por un modelo desactualizado pueden resultar significativas. Toda la arquitectura se ha concebido con una filosofía modular: cada componente — ingesta, preprocesamiento, entrenamiento, explicabilidad y monitorización — dispone de una interfaz bien definida, de modo que pueda reemplazarse o ampliarse sin perturbar el resto de la cadena.

3.2. Requisitos no funcionales

La solución debe ser escalable tanto horizontal como verticalmente. La escalabilidad horizontal permitirá repartir la carga de trabajo sobre varios nodos en un clúster; la vertical permitirá aprovechar al máximo los recursos disponibles en un único servidor cuando la carga de trabajo no justifique una infraestructura distribuida. La reproducibilidad se sostiene mediante un sistema de control de versiones integrado que rastrea el código, los datos brutos, los datos derivados y los modelos. Cada experimento queda asociado a un identificador único, lo que posibilita reconstruir cualquier resultado en el futuro, un aspecto indispensable en auditorías y buenas prácticas científicas.

El pipeline debe ofrecer transparencia tanto a usuarios técnicos como a perfiles de negocio. Para los primeros, se generan bitácoras detalladas y artefactos intermedios consultables a través de interfaces de programación de aplicaciones; para los segundos, se publican paneles ejecutivos que resumen las métricas clave con un lenguaje accesible, evitando tecnicismos superfluos. El tiempo de ejecución global — desde la ingesta hasta la

obtención de métricas finales — no puede exceder los diez minutos para conjuntos de datos de hasta cinco mil registros; este umbral asegura que los ciclos de experimentación sigan siendo ágiles y que el proceso de reentrenamiento periódico no retrase operaciones críticas. Finalmente, cada componente debe ir acompañado de documentación exhaustiva que cubra instrucciones de uso, pautas de mantenimiento y procedimientos de auditoría.

3.3. Requisitos técnicos

El desarrollo se realizará con Python a partir de la versión 3.10, lo que garantiza compatibilidad con nuevas mejoras del lenguaje y soporte prolongado por parte de la comunidad. El stack de bibliotecas incluye pandas y NumPy para manipulación de datos, scikit-learn como referencia de algoritmos clásicos, XGBoost y LightGBM para modelos basados en gradiente, SHAP para explicabilidad, y MLflow como herramienta de seguimiento de experimentos y modelos. Todas las dependencias se encapsulan en contenedores Docker, lo que facilita la portabilidad entre entornos de desarrollo, staging y producción. Para favorecer la productividad de los científicos de datos, se recomienda el uso de JupyterLab o de VS Code con extensiones específicas que integren la ejecución interactiva de notebooks, el linting y la depuración remota.

3.4. Requisitos de datos

El sistema debe manejar conjuntos de datos de pequeño tamaño, en la escala de cientos a pocos miles de observaciones, sin deteriorar la validez estadística. Se requiere soporte simultáneo para variables numéricas y categóricas; en particular, el módulo de preprocesamiento debe aplicar codificación one-hot o target encoding según la cardinalidad de la categoría y las pruebas de fuga de información. La presencia de valores nulos ha de gestionarse mediante imputación estadística o imputación supervisada, dependiendo de la densidad de la matriz. La solución debe incluir, además, la capacidad de ampliar el conjunto de entrenamiento con técnicas de generación sintética como CTGAN y de balanceo de clases como SMOTE, salvaguardando la coherencia estadística. Todos los datos reales se proporcionarán en versión anonimizada por MyTripleA y estarán sujetos a un acuerdo de confidencialidad que delimita su uso exclusivo en el marco del proyecto.

3.5. Requisitos legales y éticos

El diseño del pipeline debe cumplir las disposiciones del Reglamento General de Protección de Datos. Ello implica anonimizar cualquier identificador personal, limitar el almacenamiento a lo estrictamente necesario y preservar los derechos de acceso, rectificación y supresión. Además, el sistema tiene que excluir variables susceptibles de servir como proxy de atributos sensibles, como la raza o el género, a menos que exista una base legal y que se adopten salvaguardas que demuestren ausencia de discriminación. Cualquier decisión automatizada relevante debe disponer de una explicación comprensible para el afectado o para un auditor, y la empresa debe documentar el modelo y los datos con suficiente granularidad para permitir la verificación externa. Cuando se usen datos reales, MyTripleA se compromete a facilitar el consentimiento explícito o a demostrar una base jurídica adecuada, de acuerdo con la normativa vigente.

3.6. Requisitos del entorno de ejecución

El desarrollo local se realiza en estaciones de trabajo con al menos 4 núcleos de CPU y 16 gigabytes de memoria, recursos suficientes para manejar los experimentos de small data. Cuando se requiera acceso remoto al repositorio de datos corporativos, se habilitará un entorno virtualizado seguro que permita conexiones cifradas y registro de actividad. El pipeline ha de poder ejecutarse tanto en contenedores Docker como en entornos Conda, de modo que el proceso de integración continua pueda elegir la tecnología que mejor encaje con la infraestructura disponible.

3.7. Requisitos de validación y evaluación

La métrica principal empleada para comparar modelos es el F1-score, complementada por recall y accuracy para obtener una visión equilibrada del rendimiento. Todas las métricas se calculan en validación cruzada estratificada y se contrastan con la línea base que actualmente emplea MyTripleA en su proceso de concesión de crédito. Además de medir precisión estadística, se generarán explicaciones globales y locales que demuestren coherencia con la lógica de negocio; se comprobará, por ejemplo, que ratios de liquidez elevados se asocien de manera positiva con la probabilidad de aceptación. La presencia de data drift se evaluará inyectando perturbaciones controladas en los datos de entrada y ve-

rificando la estabilidad de las métricas y de las explicaciones SHAP. Cualquier desviación significativa obligará a desencadenar un proceso de reentrenamiento o de recalibración.

3.8. Requisitos organizativos y colaboración empresarial

El proyecto se desarrolla en el marco de prácticas curriculares y extracurriculares del Máster en Inteligencia Artificial, lo que implica coordinar plazos académicos con los ritmos de la empresa. La colaboración será predominantemente remota, con reuniones quincenales de seguimiento técnico donde se revisarán hitos, bloqueos y resultados preliminares. El acceso a los datos reales se gestiona mediante un convenio de confidencialidad que restringe la descarga local y centraliza el procesamiento en un entorno seguro proporcionado por MyTripleA. Durante el periodo de validación, el sistema se desplegará en un entorno aislado de pruebas, evitando cualquier impacto en la operativa productiva. Si los resultados cumplen los criterios de precisión, explicabilidad y rendimiento, la empresa estudiará la integración del pipeline como prototipo interno de innovación, abriendo la posibilidad de ampliarlo con módulos adicionales, por ejemplo, un motor de detección de fraude o un componente de reentrenamiento autónomo.

El cumplimiento conjunto de estos requisitos asegura que la solución propuesta no solo satisfaga los principios de ingeniería de software y ciencia de datos, sino que también responda a los imperativos regulatorios y estratégicos de la industria financiera. El uso de prácticas MLOps, la atención a la equidad y la trazabilidad, y la colaboración estrecha con MyTripleA conforman un marco robusto para llevar la investigación académica a un resultado aplicable y sostenible.

4. Objetivos

El presente Trabajo de Fin de Máster se sitúa en el ámbito del análisis de riesgo crediticio en entornos con datos limitados, un problema que afecta de manera particular a pequeñas y medianas empresas y a las fintech que les prestan servicios. A diferencia de la banca tradicional, donde se dispone de amplios históricos de clientes y operaciones, las compañías emergentes acostumbran a trabajar con muestras reducidas, registros incompletos y distribución de clases muy desbalanceada. Esta escasez de información compromete la robustez de los modelos predictivos, dificulta la justificación de las decisiones ante reguladores y eleva los costes de mantenimiento. De ahí que el proyecto plantee la creación de un flujo de trabajo MLOps modular, reproducible y transparente, específicamente orientado a la realidad de la empresa colaboradora MyTripleA. A lo largo de este capítulo se describen el objetivo general que guía la investigación y los objetivos específicos que permitirán medir su grado de cumplimiento.

4.1. Objetivo general

El propósito principal del trabajo consiste en diseñar, desarrollar y validar un pipeline MLOps de credit scoring que funcione con volúmenes reducidos de datos, integre técnicas de generación sintética y ofrezca explicaciones comprensibles de sus predicciones. El resultado esperado es un sistema capaz de mejorar la precisión y la escalabilidad del proceso de evaluación crediticia de MyTripleA, al tiempo que garantiza la trazabilidad de los experimentos, la reproducibilidad de los modelos y la conformidad con los requisitos regulatorios en materia de privacidad y no discriminación.

4.2. Desglose de los objetivos específicos

El primer objetivo específico, identificado como OE1, se centra en la elaboración de una revisión exhaustiva del estado de la cuestión. Esta investigación bibliográfica no se limita a resaltar las técnicas seminales de regresión logística y análisis discriminante, sino que examina también los avances recientes en aprendizaje automático, generación de datos sintéticos, prácticas MLOps y métodos de explicabilidad. Con este trabajo preliminar se pretende situar el proyecto en la frontera del conocimiento y extraer directrices contrasta-

das que sirvan de apoyo al diseño del pipeline.

El segundo objetivo, OE2, aborda la fase de recogida y análisis de requisitos empresariales. Para adaptar la solución a la operativa de MyTripleA se realizarán entrevistas con los equipos de riesgo, datos y cumplimiento normativo. A partir de estas sesiones se establecerán las restricciones técnicas, los umbrales de rendimiento y los criterios de aceptación que condicionarán el desarrollo posterior. El resultado será un documento de requisitos que actúe como contrato entre la universidad y la organización colaboradora.

El tercer objetivo, OE3, se refiere a la construcción de un pipeline de aprendizaje automático cuya arquitectura sea modular. Cada componente — ingesta, preprocesamiento, entrenamiento y validación — se implementará como bloque independiente con una interfaz claramente definida. Este diseño permitirá sustituir o ampliar cada módulo sin afectar al resto del sistema y facilitará la integración futura de nuevas fuentes de datos o algoritmos.

El cuarto objetivo, OE4, consiste en la incorporación de técnicas de generación de datos sintéticos. Conjuntamente con métodos de sobremuestreo como SMOTE y algoritmos generativos como CTGAN se producirá un conjunto de observaciones adicionales que mitiguen la escasez y el desequilibrio de clases. Se evaluará la similitud estadística entre los datos sintéticos y los registros originales, así como su impacto sobre la capacidad predictiva de los modelos.

El quinto objetivo, OE5, aborda la adopción de buenas prácticas MLOps. El sistema empleará MLflow para el seguimiento de experimentos y versiones de modelos, mientras que Docker garantizará la portabilidad y la coherencia de las dependencias. La construcción de imágenes y el despliegue se automatizarán mediante pipelines de integración continua que ejecuten pruebas unitarias, validaciones de datos y controles de seguridad.

El sexto objetivo, OE6, se enfoca en la evaluación rigurosa del rendimiento. Para cada escenario — datos originales, SMOTE, CTGAN, CTGAN combinada con SMOTE y SMOTE combinada con CTGAN — se entrenarán modelos base y se compararán mediante métricas como F1-score, accuracy y precision. Con este análisis se determinará qué configuración ofrece el mejor equilibrio entre exactitud y estabilidad.

El séptimo objetivo, OE7, introduce un plano de explicabilidad. Con el apoyo de librerías como SHAP y LIME se generarán interpretaciones globales y locales de cada modelo. Estas explicaciones se contrastarán con los criterios de negocio de MyTripleA para

verificar que las variables relevantes coinciden con la lógica financiera y que las recomendaciones son coherentes con la experiencia de los analistas.

El octavo y último objetivo, OE8, subraya la necesidad de documentar cada etapa. La memoria final incluirá detalles sobre decisiones de diseño, justificación de hipótesis, consideraciones éticas, salvaguardas regulatorias y pautas de mantenimiento. Esta documentación servirá de base para futuras auditorías internas y para la eventual incorporación del sistema a la infraestructura productiva de la empresa.

4.3. Alcance temporal y resultados esperados

El cumplimiento de los objetivos anteriores se estructurará en fases iterativas que combinarán investigación, desarrollo y validación. Durante las primeras semanas se completará la revisión bibliográfica y el análisis de requisitos. En un segundo bloque se construirá la primera versión del pipeline con un conjunto reducido de funciones, seguido de ciclos de mejora donde se integrarán los generadores de datos sintéticos, los módulos de explicación y los paneles de monitorización. En la fase final se llevarán a cabo pruebas de estrés y estudios de caso con datos anonimizados de MyTripleA para comprobar la robustez, la trazabilidad y la alineación con los objetivos de negocio.

Como resultado se espera disponer de un sistema operacional que, aun en modo prototipo, demuestre su capacidad de mejorar la precisión del modelo de referencia y de ofrecer justificaciones comprensibles de las decisiones. Además, el proyecto generará un repositorio versionado y un conjunto de experimentos reproducibles. Con ello se pretende cerrar el ciclo de valor que conecta la investigación universitaria con la innovación aplicada en la industria financiera.

5. Desarrollo del trabajo

5.1. Arquitectura general del sistema

El sistema desarrollado en este trabajo sigue una arquitectura modular, orientada a la trazabilidad, la automatización y la reproducibilidad de todo el ciclo de vida del aprendizaje automático en escenarios de small data. Esta estructura permite integrar distintos bloques funcionales que abarcan desde la ingesta de datos hasta la evaluación, explicación y monitorización de los modelos entrenados.

El pipeline se estructura en cinco bloques principales (Fig. 2): preprocesamiento de datos, generación sintética, entrenamiento de modelos, evaluación y explicabilidad, e integración MLOps. Cada uno de estos bloques está desacoplado del resto y encapsulado en scripts o módulos específicos, facilitando su mantenimiento e interoperabilidad. La modularidad permite ajustar, añadir o sustituir componentes sin romper el flujo general, lo cual resulta crucial en entornos reales donde los requisitos pueden evolucionar con el tiempo.

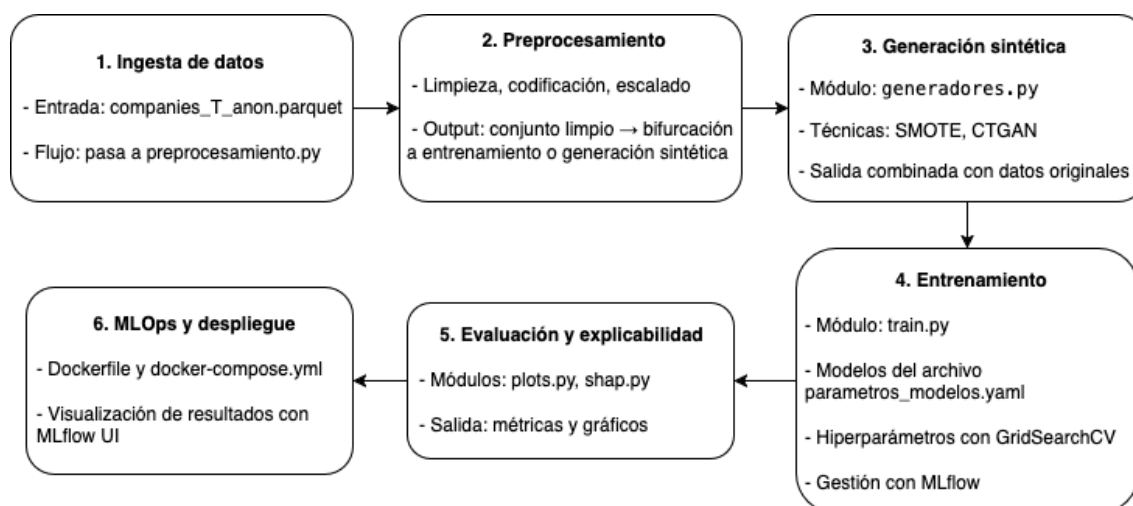


Figura 2: Arquitectura del sistema de credit scoring

En la etapa de preprocesamiento se ejecutan transformaciones sobre los datos brutos, incluyendo imputación de valores nulos, codificación de variables categóricas, escalado de variables numéricas y selección de características. Este proceso se realiza mediante el módulo `preprocesamiento.py`, diseñado para ser reutilizable con distintos conjuntos de

datos.

A continuación, se activa el bloque de generación de datos sintéticos. El script `generadores.py` permite aplicar técnicas como SMOTE y CTGAN, combinadas o por separado. Esto es especialmente útil para enriquecer las clases minoritarias y combatir tanto el desbalance de clases como la escasez de observaciones, dos problemas típicos en escenarios de crédito para pymes.

El módulo `train.py` se encarga de orquestar la fase de entrenamiento. Aquí se definen los algoritmos supervisados que se evaluarán (como Random Forest, XGBoost o MLP), junto con sus espacios de búsqueda de hiperparámetros especificados en el archivo `parametros_modelos.yaml`. Se utiliza GridSearchCV con validación cruzada estratificada para asegurar que las comparaciones entre modelos sean justas y reproducibles. Todas las ejecuciones se registran con MLflow, permitiendo versionar experimentos, modelos, métricas y configuraciones de entrada.

Una vez entrenados, los modelos se evalúan y explican usando `plots.py` y `interpretacion_shap.py`, que generan visualizaciones y análisis de importancia de características con herramientas como SHAP. Esta parte del sistema busca garantizar la transparencia y comprensibilidad de las predicciones, fundamental en entornos financieros regulados.

El sistema se integra en un entorno reproducible y portable mediante Docker. El archivo `docker-compose.yml` lanza un contenedor con el entorno completo, incluyendo la interfaz web de MLflow, facilitando su despliegue tanto en local como en servidores remotos o en la nube. La estructura del proyecto permite además incorporar fácilmente mecanismos de monitorización y detección de cambios en los datos (data drift), así como versiones futuras con integración continua. Esta arquitectura general proporciona una base sólida para experimentación, comparación y mejora progresiva de modelos de scoring en entornos fintech.

5.2. Preparación del entorno

La preparación del entorno de desarrollo constituye un paso esencial para asegurar la trazabilidad, reproducibilidad y escalabilidad del pipeline propuesto. El sistema se configura completamente en Python, con una versión mínima de 3.10, y se apoya en un entorno basado en Jupyter Lab para prototipado rápido, desarrollo iterativo y documentación visual de los procesos. La elección de este entorno facilita el análisis exploratorio de datos,

la visualización de resultados y la implementación gradual de funciones en notebooks o scripts modulares.

Para la gestión y seguimiento de experimentos se utiliza MLflow, una herramienta que permite registrar cada ejecución del pipeline con sus parámetros, métricas y artefactos generados. MLflow se ejecuta como un servicio independiente mediante Docker, lo cual permite su acceso vía navegador web para consultar los resultados de forma organizada y permanente. Esta configuración contribuye significativamente a la trazabilidad del proyecto y a la comparación sistemática de modelos y configuraciones.

El entorno completo se encapsula en contenedores Docker definidos a través del archivo `docker-compose.yml`. Este archivo lanza los servicios necesarios, incluyendo la interfaz de MLflow, el entorno Jupyter Lab y los volúmenes compartidos. De este modo, se evita la dependencia directa de librerías instaladas en el sistema operativo anfitrión y se asegura la portabilidad del proyecto, tanto en máquinas locales como en servidores remotos o plataformas cloud.

La estructura del proyecto sigue una organización clara en carpetas (Tabla 1), orientada a separar datos, código, configuraciones y resultados. La raíz del proyecto incluye archivos clave como `config.py`, donde se definen rutas, constantes globales y parámetros generales; `parametros_modelos.yaml`, que centraliza la definición de hiperparámetros para los modelos; y `requirements.txt`, que contiene la lista de dependencias necesarias para ejecutar el sistema. Además, cada componente principal del pipeline se ubica en archivos independientes, como `preprocesamiento.py`, `generadores.py`, `train.py` y `plots.py`, lo que permite mantener el código limpio, escalable y reutilizable.

Para asegurar la gestión de versiones, se utiliza Git como sistema de control de versiones. Todos los cambios relevantes en el código, la configuración y los experimentos quedan registrados en commits asociados a ramas del repositorio local o remoto. Esta práctica permite volver atrás en caso de errores, documentar la evolución del proyecto y facilitar el trabajo colaborativo. El repositorio del proyecto está disponible en GitHub. Las dependencias del entorno se gestionan mediante entornos virtuales reproducibles, incluidos dentro del contenedor Docker, evitando incompatibilidades o conflictos entre librerías.

Archivo/Carpeta	Descripción
<code>--pycache--/</code>	Archivos temporales de Python
<code>bibliography/</code>	Bibliografía y referencias del TFM
<code>companies_T_anon.parquet</code>	Dataset anonimizado en formato Parquet
<code>config.py</code>	Configuración general del proyecto
<code>diagramas/</code>	Diagramas y visualizaciones
<code>docker-compose.yml</code>	Orquestación de servicios con Docker
<code>Dockerfile</code>	Definición del entorno Docker
<code>generadores.py</code>	Generación de datos o modelos
<code>mlruns/</code>	Registros de experimentos de MLflow
<code>notebooks/</code>	Cuadernos Jupyter para exploración y pruebas
<code>outputs/</code>	Resultados generados por el modelo
<code>parametros_modelos.yml</code>	Hiperparámetros de los modelos
<code>plots.py</code>	Script de generación de gráficas
<code>preprocesamiento.py</code>	Procesamiento y limpieza de datos
<code>README.md</code>	Descripción general del proyecto
<code>requirements.txt</code>	Dependencias del entorno en Python
<code>interpretacion_shap.py</code>	Análisis de explicabilidad con SHAP
<code>train.py</code>	Entrenamiento de modelos

Tabla 1: Contenido de la carpeta `tfm`

En conjunto, la preparación del entorno técnico ofrece una base sólida y profesional para el desarrollo de proyectos de machine learning con garantías de control, escalabilidad y mantenimiento a largo plazo.

5.3. Recopilación y comprensión de los datos

La primera etapa de cualquier proyecto de machine learning consiste en conocer a fondo la materia prima con la que se va a trabajar. En el presente trabajo se dispone de un único fichero, `companies_T_anon.parquet`, que recoge registros de empresas que han solicitado financiación a MyTripleA entre los años 2021 y 2023. Cada fila corresponde a una empresa en un instante temporal determinado y cada columna describe un aspecto financiero, operativo o de comportamiento asociado a esa entidad. El archivo se encuentra

anonimizado: la columna `codigo_empresa` en vez de contener valores con código internos de la empresa incluye los números ordinales. Las operaciones de carga se realizan con la biblioteca **pandas**, que permite leer directamente el formato Parquet y manejar estructuras tabulares de gran tamaño con eficiencia. En la tabla 2 se puede observar la descripción de cada variable presente en el conjunto de datos.

Tabla 2: Descripción de las variables del dataset

Variable	Descripción
<code>antiguedad_cuenta_dias</code>	Días transcurridos desde la apertura de la cuenta; mayor valor implica mayor estabilidad.
<code>apalancamiento_financiero</code>	Relación entre deuda y capital propio; indica el riesgo de endeudamiento.
<code>autonomia_financiera</code>	Fondos propios sobre activos totales; mide la independencia financiera.
<code>autonomia_financiera %</code>	Variación interanual a tres años de la autonomía financiera.
<code>cash_flow_antes_var</code>	Flujo de caja antes de variaciones de capital; mide liquidez operativa.
<code>codigo_empresa</code>	Identificador único de la empresa; útil para trazabilidad, no para modelar.
<code>coste_financiero</code>	Costes derivados de la deuda; refleja la carga financiera.
<code>coste_financiero %</code>	Evolución del coste financiero a lo largo de tres años.
<code>deuda_financiera_cp %</code>	Cambio en el peso de la deuda a corto plazo; indica riesgo de liquidez.
<code>deuda_financiera_lp %</code>	Cambio en el peso de la deuda a largo plazo; muestra estabilidad o dependencia de financiación externa.
<code>dfn_ebitda</code>	Deuda financiera neta dividida por EBITDA; capacidad de pago.
<code>dfn_ebitda %</code>	Variación de $dfn/EBITDA$ en los últimos tres años.
<code>ebitda_gastos_financieros</code>	Relación EBITDA-gastos financieros; capacidad de cobertura de intereses.

Tabla 2 – continuación

Variable	Descripción
ebitda_gastos_financieros %	Evolución de la cobertura financiera.
existencias_ventas	Inventario sobre ventas; mide eficiencia operativa.
existencias_ventas %	Variación en la eficiencia del inventario.
fecha_constitucion	Fecha de creación de la empresa; se relaciona con antigüedad.
fondo_manioobra_porcentual %	Evolución del fondo de manioobra; muestra cambios en liquidez.
fondos_propios	Capital propio total; indicador de solvencia.
fondos_propios %	Evolución de los fondos propios.
gastos_financieros %	Cambio en la proporción de gastos financieros.
gastos_personal	Total destinado a sueldos; refleja tamaño y estructura de costes.
gastos_personal %	Variación del gasto en personal.
grade_code	Calificación crediticia asignada (1-10); variable objetivo en el modelo.
importe_cobro_medio	Tiempo medio de cobro; mide eficiencia en la recuperación de ingresos.
inmovilizado_material	Valor de los activos fijos; indica inversión en infraestructura.
inmovilizado_material %	Cambio en la inversión en inmovilizado.
inversiones_financieras_lp %	Variación de inversiones financieras a largo plazo.
margen_ebitda	EBITDA sobre ventas; rentabilidad operativa.
margen_final	Beneficio neto sobre ventas; rentabilidad final.
media_dias_impago	Días promedio de retraso en los pagos; indicador directo de riesgo.
nof_ventas	Necesidad operativa de fondos; ligado al ciclo financiero.
nof_ventas %	Variación del ciclo de caja.
numero_trabajadores	Tamaño de la empresa medido en empleados.
periodo_medio_almacenamiento %	Cambio en días de inventario; mide eficiencia logística.

Tabla 2 – continuación

Variable	Descripción
periodo_medio_cobro %	Evolución del plazo medio de cobro; relacionado con liquidez.
periodo_medio_pago %	Cambios en el plazo medio de pago; puede reflejar tensión de caja.
resultados_ejercicio	Beneficio neto anual; indicador global de salud financiera.

Tabla 2: Variables presentes en el conjunto de datos y su significado

Tamaño y tipología de la información

Tras la carga inicial se observa una tabla de 440 filas y 41 columnas. Cuarenta columnas son numéricas y una es categórica (**grade_code**). Dentro de las variables numéricas se distingue un subconjunto de enteros (por ejemplo **codigo_empresa**) y otro de decimales, predominantes en ratios y porcentajes. La única columna de tipo fecha es **fecha_constitucion**, transformada después en campo numérico (antigüedad en días) para facilitar su uso como predictor.

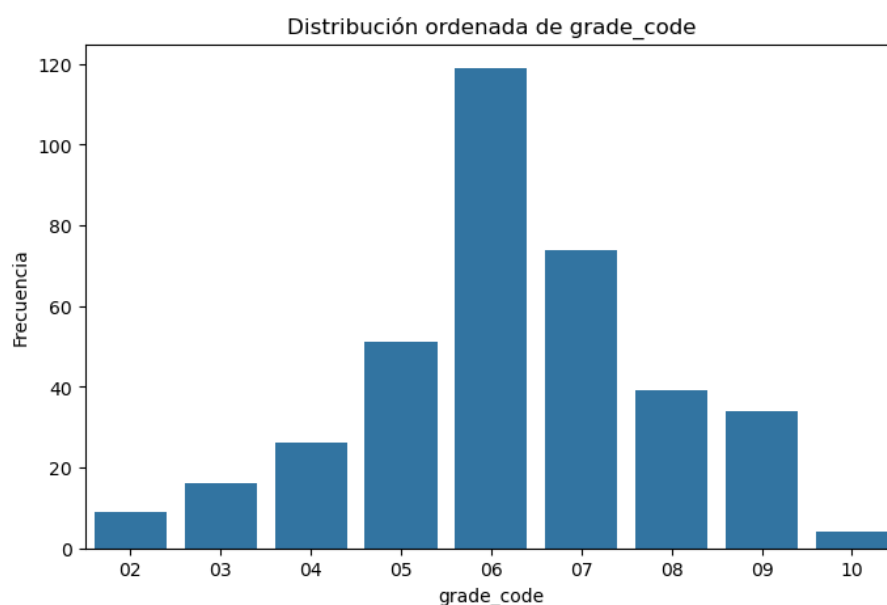


Figura 3: El histograma de frecuencias de **grade_code**

La variable objetivo es **grade_code**. A diferencia de otras bases de scoring en las que la

calificación se expresa como binaria (impago sí/no), aquí se dispone de una escala ordinal de diez niveles donde 1 indica riesgo muy bajo y 10 riesgo muy alto. El histograma de frecuencias en la figura 3 revela una clara concentración en las categorías intermedias: casi el 27 % de los registros pertenecen a la nota 6 y otro 18 % a la nota 7. Los extremos presentan escasas observaciones, hecho que condicionará la estrategia de balanceo y, en versiones preliminares, ha motivado la reconversión del problema a clasificación binaria con umbral 7.

Exploración inicial y descripción estadística

El cuaderno `EDA.ipynb` realiza un diagnóstico detallado. En primer lugar se calcula el porcentaje de valores nulos por columna. Tres variables (`antiguedad_cuenta_dias`, `media_dias_impago` e `importe_cobro_medio`) superan el 69 % de ausencias (ver la figura 11). A la vista de esta carencia y de su limitada relevancia para predicciones a corto plazo, se decide descartarlas. Se eliminan igualmente todas las columnas cuyo faltante rebasa el 65 %. El número final de atributos desciende de 41 a 37.

A continuación se imprimen los tipos de dato y se generan estadísticas básicas con `df.describe()`. La inspección revela fuertes asimetrías (ver la figura 12): por ejemplo `ventas_netas` exhibe una media de ciento cuarenta millones de euros, un mínimo cercano a doscientos mil y un máximo superior a once mil millones, clara señal de colas largas. Algo similar ocurre en `fondos_propios` o `inmovilizado_material`. Para mitigar efectos de escala se adoptan transformaciones logarítmicas en las fases de modelado de gradiente, aunque se preservan los valores brutos para algoritmos basados en árboles.

Calidad de los datos y tratamiento de ausencias

Eliminados los atributos con alta proporción de vacíos, se revisa la matriz resultante. Persisten columnas con entre 15 % y 24 % de valores faltantes. Se opta por imputar con la mediana, estrategia conservadora que preserva estructura de datos extremos sin introducir la varianza adicional propia de la media aritmética. Las imputaciones se aplican exclusivamente a columnas numéricas; la columna categórica queda inalterada.

En cuanto a las filas, sólo se descartan aquellas que carecen de `grade_code`. Puesto que la calificación es la etiqueta de aprendizaje supervisado, su ausencia imposibilita la inclusión de la observación en el entrenamiento. Terminado el proceso, el conjunto definiti-

vo contiene 372 registros, una reducción moderada pero aceptable en el contexto de small data.

Distribución de la variable objetivo

Con el dataset depurado se calcula la frecuencia relativa de cada nota. Los valores más comunes son 6, 7 y 5, que en conjunto suponen más del sesenta por ciento. Las clases 1, 2, 9 y 10 representan menos del siete por ciento acumulado. Se aconseja replantear el problema en términos binarios: riesgo bajo (1–6) frente a riesgo medio-alto (7–10). De esta forma se equilibran los grupos (59 % frente a 41 %) y se evita el sobreajuste en clases testimoniales. Para no perder granularidad, el código conserva la nota original como referencia y sólo transforma a binario en el paso de selección de variables.

Relaciones entre variables

La matriz de correlación de Pearson, graficada mediante `seaborn.heatmap` (figura 13), muestra ausencia de multicolinealidad severa. Los coeficientes más altos se dan entre pares de variables derivadas, como `fondos_propios` y `fondos_propios %`, o `ebitda_gastos_financieros` y su versión porcentual. Entre los ratios clásicos aparece un coeficiente negativo moderado entre `apalancamiento_financiero` y `autonomia_financiera`, relación coherente con la teoría contable: a mayor endeudamiento, menor autonomía de fondos propios. Estas relaciones sirven de guía para decidir qué variables retener cuando se aplican técnicas de selección basadas en varianza o en importancia de características.

Visualizaciones y hallazgos clave

- **Histograma global.** En la figura 12 se observa un panel de histogramas de veinte filas por dos columnas evidencia colas derechas pronunciadas en la mayoría de métricas contables. La dispersión obliga a considerar escalados o transformaciones no lineales.
- **Barra de valores nulos.** Un gráfico horizontal en la figura 11 recoge la cantidad absoluta de ausencias variable a variable, lo que justifica la poda inicial de columnas y la imputación por mediana.
- **Matriz de correlación.** En la figura 13 se confirma la independencia relativa entre la mayoría de atributos, al tiempo que se identifican clústeres contables susceptibles

de reducción de dimensionalidad.

Implicaciones para la fase de modelado

Del análisis se extraen varias líneas de acción. Primero, la escala de valores obliga a normalizar sólo cuando el algoritmo lo requiera, para evitar perder interpretabilidad en modelos basados en árboles. Segundo, las clases poco representadas hacen imprescindible un método de balanceo; se escoge la combinación SMOTE-CTGAN para comparar aumento sintético con sobremuestreo clásico. Tercero, se observa la conveniencia de conservar ciertas variables aunque estén fuertemente correlacionadas con sus versiones en porcentaje, pues la explicación del modelo puede apoyarse en la comparación entre nivel absoluto y tendencia. La estabilidad temporal verificada en variables de rentabilidad sugiere que puede resultar útil incorporar una dimensión de series temporales en futuras extensiones, aunque el presente trabajo se centra en un corte transversal.

5.4. Preprocesamiento de los datos

El preprocesamiento constituye una fase esencial en todo proyecto de aprendizaje automático, ya que garantiza que los datos estén en condiciones adecuadas para entrenar modelos robustos, eficientes y coherentes. En este trabajo, el conjunto de datos utilizado presenta diversas particularidades, como la presencia de valores nulos, columnas irrelevantes, y una variable objetivo altamente desbalanceada. Por ello, se ha desarrollado un script específico (`preprocesamiento.py`) que automatiza los pasos necesarios para preparar el dataset final.

En primer lugar, se realiza una limpieza estructural del conjunto de datos, eliminando aquellas columnas que contienen más de un 50 % de valores ausentes o que presentan una alta redundancia respecto a otras variables. A continuación, se lleva a cabo una imputación de los valores nulos restantes. Para las variables numéricas, se aplica la estrategia de rellenado con la mediana, una técnica robusta frente a valores extremos. En el caso de variables categóricas o que contienen identificadores, se emplea una codificación tipo "desconocido" para preservar la información sin introducir sesgos artificiales.

Una vez tratadas las ausencias, se realiza una codificación de variables según el tipo de algoritmo que se utilizará en etapas posteriores. Algunas variables categóricas se codifican mediante one-hot encoding, mientras que otras se transforman en etiquetas ordinales.

Este enfoque permite mantener la compatibilidad con distintos algoritmos supervisados, como árboles de decisión, regresión logística o modelos basados en distancias. Además, las variables numéricas son normalizadas utilizando escalado estándar, lo que mejora la estabilidad de modelos lineales y redes neuronales.

En cuanto a la ingeniería de características, se incorporan nuevas variables derivadas del cálculo de ratios financieros, diferencias porcentuales interanuales y otros indicadores derivados que pueden contener señales predictivas valiosas. También se aplican técnicas de reducción de dimensionalidad, como la eliminación de variables con varianza casi nula o alta colinealidad, utilizando métodos como `VarianceThreshold` y análisis de correlación. Estas estrategias permiten reducir el riesgo de sobreajuste y mejorar la interpretabilidad de los modelos.

Uno de los aspectos más desafiantes del dataset original es el fuerte desbalance en la distribución de la variable objetivo `grade_code`. Para mitigar este problema, se aplica una estrategia de oversampling mediante SMOTE, que genera nuevas observaciones sintéticas para las clases minoritarias utilizando interpolación entre vecinos más cercanos. Esta técnica permite equilibrar el conjunto de entrenamiento sin perder la diversidad de los datos originales. En algunos escenarios, se combina además con la generación de datos mediante CTGAN para evaluar su impacto combinado en el rendimiento del modelo.

En resumen, el preprocesamiento se implementa como una función reutilizable que permite transformar el dataset original en un formato coherente y balanceado, adaptado a las exigencias de los algoritmos utilizados posteriormente. La estructura modular del script facilita su integración en el pipeline general y garantiza la reproducibilidad del proceso en futuros experimentos o despliegues.

5.5. Generación de datos sintéticos

La generación de datos sintéticos representa una de las estrategias más eficaces para abordar el problema del desbalance de clases y la escasez de muestras en contextos de small data. En este trabajo, se incorporan dos técnicas complementarias que permiten enriquecer el conjunto de entrenamiento sin comprometer la representatividad de los datos: SMOTE y CTGAN. Ambas se integran dentro de un script específico, `generadores.py`, que automatiza su aplicación y facilita la experimentación en múltiples escenarios.

SMOTE, sigla de Synthetic Minority Over-sampling Technique, es una técnica que

crea nuevas observaciones sintéticas para las clases minoritarias mediante interpolación lineal entre instancias reales de la misma clase. Su principal ventaja reside en que no introduce duplicados, como ocurre con técnicas tradicionales de oversampling, sino que genera puntos intermedios en el espacio de características, lo que contribuye a mejorar la generalización del modelo. Esta técnica resulta especialmente útil cuando las clases desequilibradas presentan una cierta dispersión, ya que permite densificar el espacio sin introducir ruido excesivo.

Por otro lado, se implementa CTGAN, un modelo generativo adversarial condicional diseñado específicamente para datos tabulares. A diferencia de SMOTE, CTGAN es capaz de capturar distribuciones complejas y relaciones no lineales entre variables, generando muestras altamente realistas que respetan tanto la estructura como las dependencias estadísticas del conjunto de datos original. El modelo se entrena utilizando una arquitectura GAN donde el generador y el discriminador compiten iterativamente hasta lograr una convergencia estable. Este enfoque resulta especialmente útil para generar combinaciones válidas de valores categóricos y numéricos, preservando la coherencia semántica y lógica entre atributos.

El script `generadores.py` permite entrenar y aplicar ambos métodos, controlando aspectos clave como el número de muestras generadas, las clases objetivo y la combinación entre datos reales y sintéticos. Se realizan distintas pruebas para comparar escenarios de entrenamiento con datos originales, con datos balanceados mediante SMOTE, y con datos enriquecidos con CTGAN, tanto de forma individual como combinada.

Para validar la calidad de los datos generados, se llevan a cabo análisis estadísticos y visualizaciones comparativas. Se analizan distribuciones marginales de variables clave, como `apalancamiento_financiero`, `margen_ebitda` o `media_dias_impago`, y se comparan con las mismas variables en el conjunto real. Las distribuciones generadas por SMOTE tienden a estar más acotadas alrededor de los valores existentes, mientras que las generadas por CTGAN presentan una mayor dispersión y riqueza estructural, lo cual puede ser ventajoso o problemático dependiendo del modelo utilizado. También se utilizan métricas de divergencia como Jensen-Shannon para medir la similitud entre los conjuntos original y sintético.

En los experimentos realizados, se observa que la combinación de ambas técnicas — utilizando primero SMOTE para reforzar clases específicas y luego CTGAN para aumentar

la diversidad general — produce los mejores resultados en términos de rendimiento predictivo. Esta estrategia mixta permite controlar el equilibrio de clases sin perder la riqueza estadística del dataset. Sin embargo, también se identifica la necesidad de una validación cuidadosa, ya que un exceso de datos generados puede inducir ruido o patrones artificiales que reducen la capacidad de generalización del modelo.

En conclusión, la generación de datos sintéticos se consolida como una herramienta esencial en el presente trabajo, permitiendo ampliar y equilibrar el conjunto de entrenamiento en un entorno real con datos limitados. La implementación modular y parametrizable facilita su integración dentro del pipeline completo y sienta las bases para futuras mejoras mediante técnicas generativas más avanzadas o específicas del dominio financiero.

5.6. Diseño y entrenamiento de modelos

El diseño del sistema de predicción crediticia incluye la evaluación de diversos modelos de clasificación supervisada, seleccionados por su robustez y adecuación a contextos de small data. Se ha entrenado una batería de modelos que incluye Random Forest, XGBoost, Gradient Boosting, K-Nearest Neighbors (KNN), Support Vector Machines (SVC), regresión logística, Naive Bayes y perceptrones multicapa (MLP). Estos modelos han sido seleccionados por su balance entre interpretabilidad, capacidad de generalización y adaptabilidad a volúmenes reducidos de datos.

El conjunto de variables predictoras se ha determinado tras aplicar un proceso riguroso de selección, en el cual se descartan aquellas que presentaban alta multicolinealidad, baja varianza o una proporción elevada de valores nulos no imputables. Adicionalmente, se han transformado variables categóricas a formato numérico mediante codificación one-hot o ordinal, según el modelo. La normalización de variables numéricas también se ha realizado en aquellos modelos sensibles a escalas, como SVC o regresión logística. En la figura 4) podemos observar la selección de variables de uno de los mejores modelos con F1-score de 0.78.

La sintonización de hiperparámetros se ha llevado a cabo mediante búsqueda exhaustiva con GridSearchCV, utilizando validación cruzada estratificada con $k=5$ pliegues. Esta metodología garantiza que los modelos seleccionados se ajusten correctamente sin sobreentrenarse, y además permite una evaluación comparativa entre algoritmos. El archivo `parametros_modelos.yaml` centraliza todas las combinaciones de hiperparámetros explo-

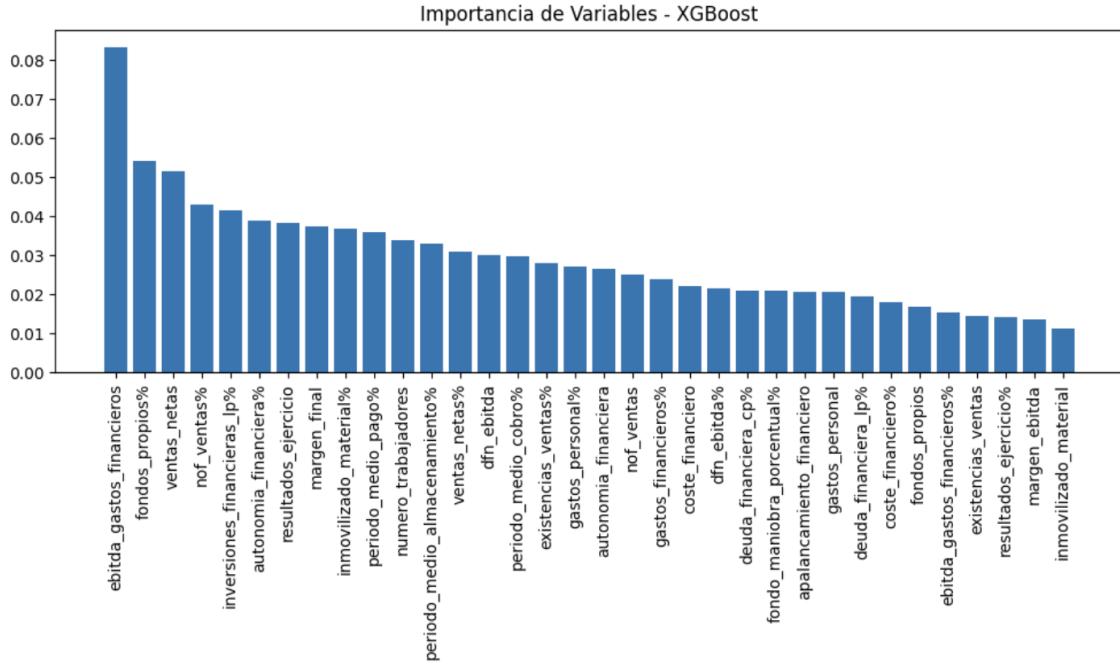


Figura 4: Importancia de variables - XGBoost - SMOTE (F1=0.78)

radas. Por ejemplo, para Random Forest se han evaluado combinaciones de `n_estimators` (50, 100) y `max_depth` (5, 10, sin límite), mientras que para XGBoost se han ajustado `n_estimators` (50, 100) y `max_depth` (3, 6). Modelos más simples como Naive Bayes no han requerido ajustes, mientras que MLP ha sido evaluado con distintas capas ocultas ([50], [100]) y tasas de regularización (`alpha`) de 0.0001 y 0.01.

El script `train.py` centraliza la lógica de entrenamiento y validación. Dentro de este script se gestionan la partición del conjunto de datos, la aplicación de las técnicas de balanceo elegidas (como SMOTE o la combinación con datos generados por CTGAN), y el pipeline completo de entrenamiento con los parámetros obtenidos de `parametros_modelos.yaml`. Además, se utiliza el módulo `config.py` para gestionar rutas, nombres de variables clave y configuraciones que permiten mantener el sistema modular y fácilmente reproducible.

Todo el proceso de entrenamiento y evaluación se encuentra registrado en MLflow, que actúa como sistema de tracking de experimentos. Cada ejecución guarda automáticamente las métricas obtenidas (accuracy, F1-score, ROC-AUC), los parámetros utilizados y el modelo entrenado, lo cual permite comparar resultados y mantener un historial detallado de todos los experimentos. Esto no solo garantiza la trazabilidad, sino que también facilita

la colaboración y replicabilidad en contextos profesionales y académicos.

El enfoque adoptado ha permitido validar empíricamente que los modelos de tipo ensemble, como Random Forest y XGBoost, ofrecen los mejores resultados en términos de F1-score y estabilidad entre pliegues. Además, la integración de técnicas de generación sintética y oversampling ha demostrado mejorar sustancialmente el rendimiento, especialmente en clases minoritarias. En conjunto, esta fase constituye el núcleo predictivo del sistema propuesto y establece las bases para su posterior evaluación explicativa y despliegue en un entorno MLOps.

5.7. Explicabilidad del modelo

Uno de los objetivos clave de este proyecto es garantizar que las predicciones del sistema de credit scoring sean comprensibles tanto para usuarios técnicos como no técnicos. Para ello, se han implementado técnicas de explicabilidad basadas en la metodología SHAP (Shapley Additive Explanations), que permiten descomponer la salida del modelo en contribuciones individuales por variable, proporcionando una interpretación precisa del comportamiento del modelo.

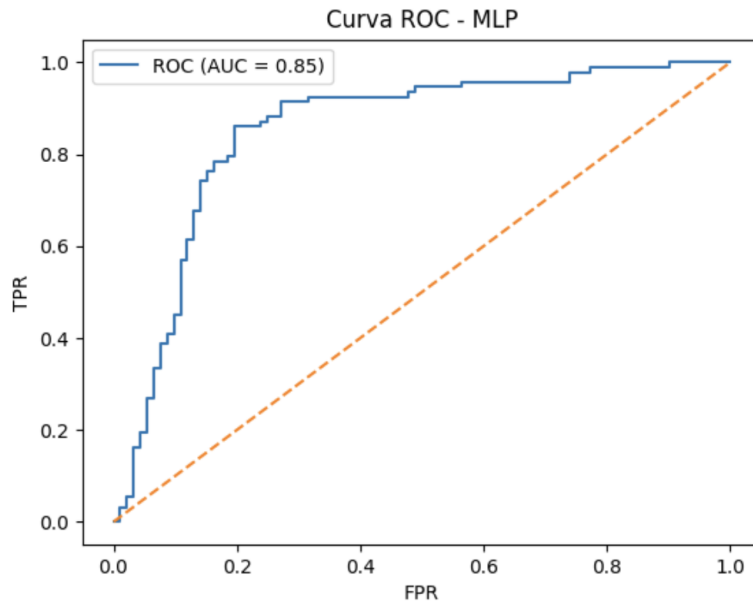


Figura 5: Curva ROC para el modelo MLP ($AUC = 0.85$)

Dado que el modelo MLP entrenado sobre datos generados con CTGAN y balanceados

con SMOTE ha obtenido el mejor rendimiento ($F1 = 0.83$, ver la figura 5), se ha priorizado su análisis interpretativo. Al tratarse de un modelo de tipo caja negra, se ha utilizado `shap.KernelExplainer`, adecuado para modelos no basados en árboles, aunque con un coste computacional más elevado.

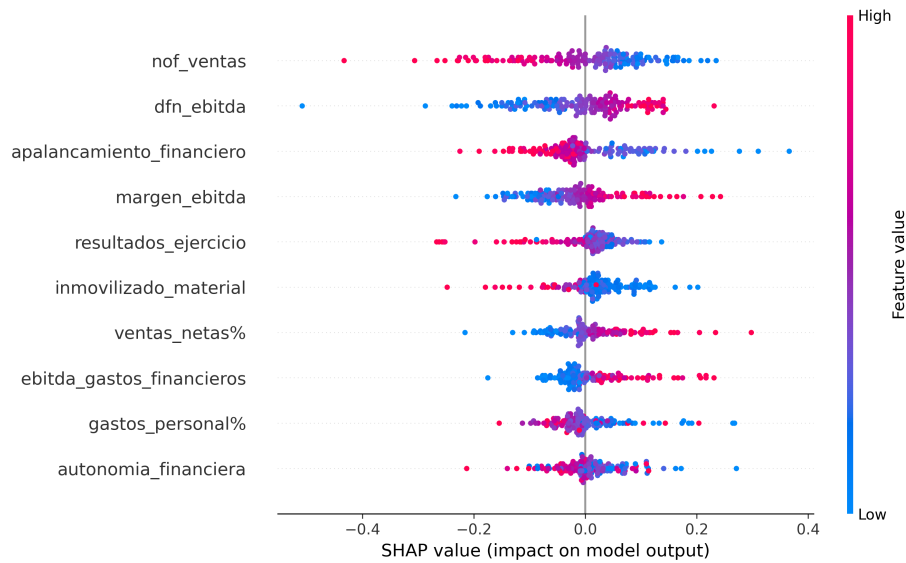


Figura 6: Importancia global de variables (summary plot) – MLP

La Figura 6 muestra el gráfico de resumen SHAP, que representa el impacto individual de cada variable sobre las predicciones del modelo en función de su valor. Cada punto corresponde a una observación del conjunto de datos, posicionada según cuánto contribuye esa variable específica a aumentar o disminuir la probabilidad de clasificación positiva. El eje horizontal indica el valor SHAP, es decir, la magnitud y dirección del impacto en la predicción, mientras que el color refleja el valor original de la variable, desde azul (valores bajos) hasta rojo (valores altos).

Se observa que las variables `nof_ventas`, `dfn_ebitda`, `apalancamiento_financiero` y `margen_ebitda` son las que más influyen en la salida del modelo, presentando una alta dispersión de valores SHAP. En particular, valores altos de `nof_ventas` (en rojo) tienden a aumentar las predicciones, lo que indica que un mayor número de ventas se asocia a menor riesgo. De forma similar, valores bajos de `dfn_ebitda` (en azul), que reflejan un bajo nivel de deuda neta en relación al EBITDA, también empujan la predicción hacia una clasificación positiva. Por el contrario, un alto `apalancamiento_financiero` reduce la predicción del modelo, sugiriendo que una estructura excesivamente apalancada es per-

cibida como más riesgosa. Finalmente, un mayor `margen_ebitda` tiene un efecto positivo claro, en línea con la lógica financiera de que una mayor rentabilidad operativa reduce el riesgo crediticio. Este tipo de visualización permite interpretar de manera intuitiva tanto la importancia global como el comportamiento local de cada variable, y es especialmente útil para detectar patrones, relaciones no lineales y posibles interacciones implícitas.

En la Figura 7, se presenta la importancia promedio de cada variable, calculada como la media del valor absoluto de SHAP. Esta visualización corrobora que `nof_ventas` y `dfn_ebitda` son los factores más determinantes a nivel global.

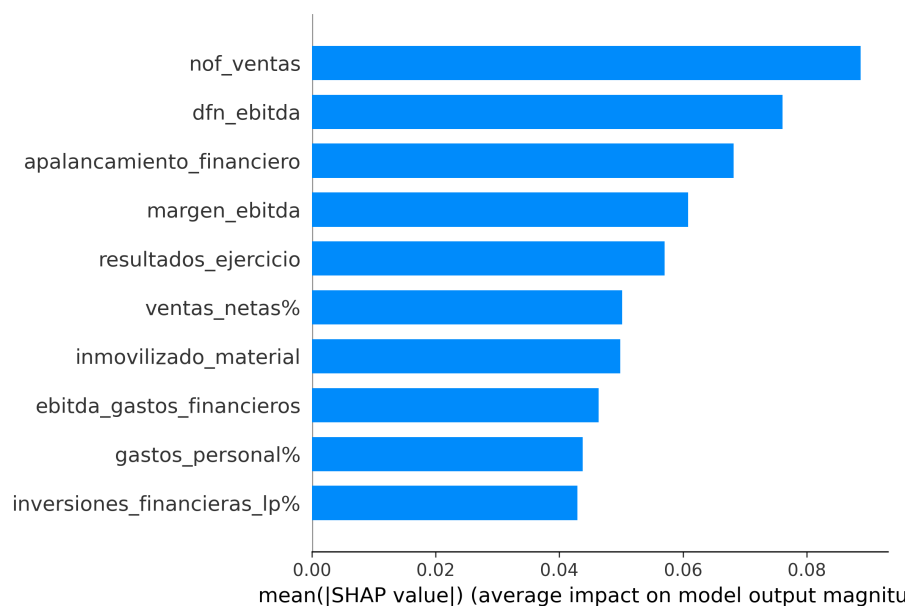


Figura 7: Impacto medio absoluto por variable – MLP

La Figura 8 presenta una serie de gráficos de dependencia SHAP que permiten analizar cómo varía el impacto de cada variable sobre la predicción del modelo en función de su valor, y cómo estas relaciones se ven influidas por la interacción con una segunda variable. En cada gráfico, el eje horizontal representa el valor original de la característica principal, mientras que el eje vertical muestra el valor SHAP asociado, es decir, su impacto individual en la salida del modelo. El color de los puntos refleja el valor de una segunda variable que interactúa con la principal, permitiendo detectar relaciones cruzadas y no lineales.

Por ejemplo, se aprecia una relación inversa entre `nof_ventas` y su valor SHAP: a mayor número de ventas, mayor contribución positiva a la clasificación favorable. En contraste, valores bajos de `dfn_ebitda` tienden a aumentar la predicción, lo que indica que una

menor deuda neta respecto al EBITDA es percibida como un factor de menor riesgo. También se observan efectos negativos cuando `apalancamiento_financiero` es elevado, ya que esto incrementa el riesgo crediticio estimado. En el caso de `ventas_netas%` o `gastos_financieros%`, el gráfico revela interacciones evidentes, donde el efecto de una variable depende de los niveles de otra, lo cual es clave para entender el comportamiento del modelo más allá de simples relaciones lineales.

Estos gráficos de dependencia resultan especialmente útiles para identificar umbrales, regiones de mayor sensibilidad del modelo y efectos combinados entre variables, mejorando tanto la interpretabilidad como la posibilidad de ajustar políticas basadas en reglas derivadas del modelo.

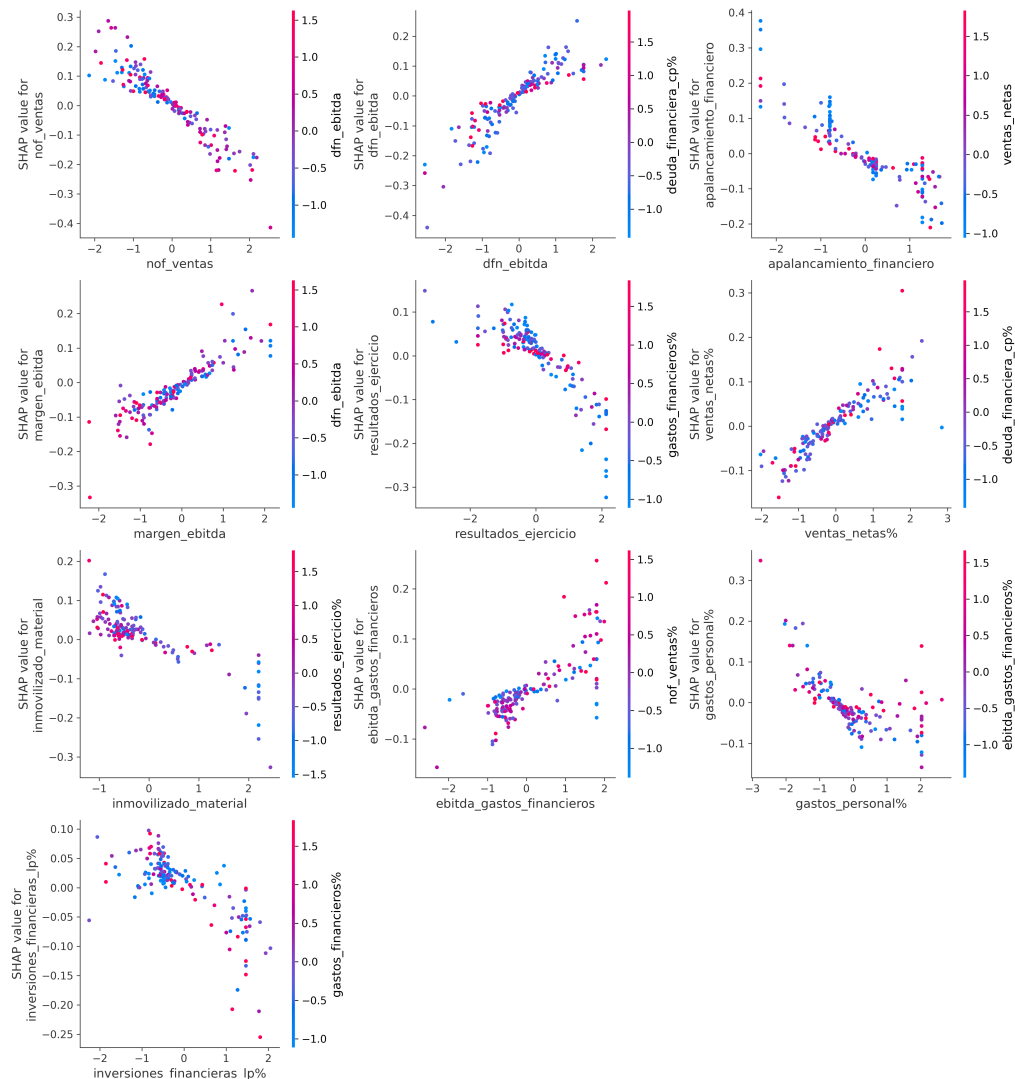


Figura 8: Gráficos de dependencia SHAP para las principales variables – MLP

Asimismo, la Figura 9 presenta una explicación individual generada con `force.plot`, que muestra cómo ciertas características específicas de una observación han contribuido positiva o negativamente a su clasificación final. Este tipo de visualizaciones es especialmente útil para decisiones individuales y trazabilidad de casos concretos.

Finalmente, la Figura 15 muestra un gráfico de trayectoria de decisión (decision plot), que permite entender cómo se van acumulando los efectos de las distintas variables en cada instancia para llegar a la predicción final. Esta herramienta facilita el análisis detallado del razonamiento interno del modelo.

Este sistema de explicabilidad contribuye de forma decisiva a la transparencia del modelo, facilita auditorías internas o regulatorias y permite alinear las decisiones automatizadas con las políticas crediticias de la entidad. Además, actúa como puente entre analistas técnicos y perfiles de negocio, favoreciendo la confianza y adopción del sistema.

Complementariamente, en el fichero `explicacion_codigo.ipynb`, disponible en el repositorio de GitHub, se presenta un análisis paralelo de interpretabilidad para el modelo XGBoost entrenado con SMOTE ($F1 = 0.78$), utilizando `shap.TreeExplainer`. Esta herramienta, diseñada específicamente para modelos basados en árboles, permite una estimación más eficiente y precisa de las contribuciones SHAP, con menor coste computacional y mayor interpretabilidad directa que los modelos tipo caja negra.

En contraste, el análisis aplicado al modelo MLP requiere el uso de `shap.KernelExplainer` o `PermutationExplainer`, los cuales estiman las contribuciones mediante técnicas de muestreo, con mayor sensibilidad al ruido, mayor tiempo de cómputo y ciertas limitaciones en la estabilidad de los valores explicativos. Por ello, el análisis del modelo XGBoost actúa como punto de comparación útil para validar la coherencia de los patrones identificados en el modelo neuronal y aporta mayor robustez al proceso global de explicación.

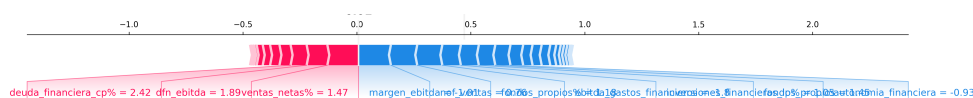


Figura 9: Explicación individual con `force_plot` – MLP

5.8. MLOps y automatización del ciclo de vida

El desarrollo de este proyecto ha seguido principios fundamentales del enfoque MLOps con el objetivo de garantizar la trazabilidad, reproducibilidad y mantenimiento continuo del sistema de credit scoring. El diseño modular del pipeline permite la separación clara entre las etapas de preprocesamiento, generación de datos sintéticos, entrenamiento, evaluación y explicación del modelo, facilitando la gestión de versiones y la incorporación de mejoras iterativas sin afectar al sistema completo.

Una pieza clave en esta arquitectura ha sido el uso de MLflow (ver la figura 10) como herramienta de gestión de experimentos. A través de su interfaz y funcionalidades de tracking, se registran automáticamente los hiperparámetros utilizados en cada experimento, las métricas obtenidas (como F1-score, ROC-AUC y precisión), así como los artefactos generados, incluyendo los modelos entrenados. Estos elementos quedan almacenados en la carpeta mlruns, lo que permite comparar versiones de forma rigurosa y recuperar configuraciones anteriores cuando sea necesario.

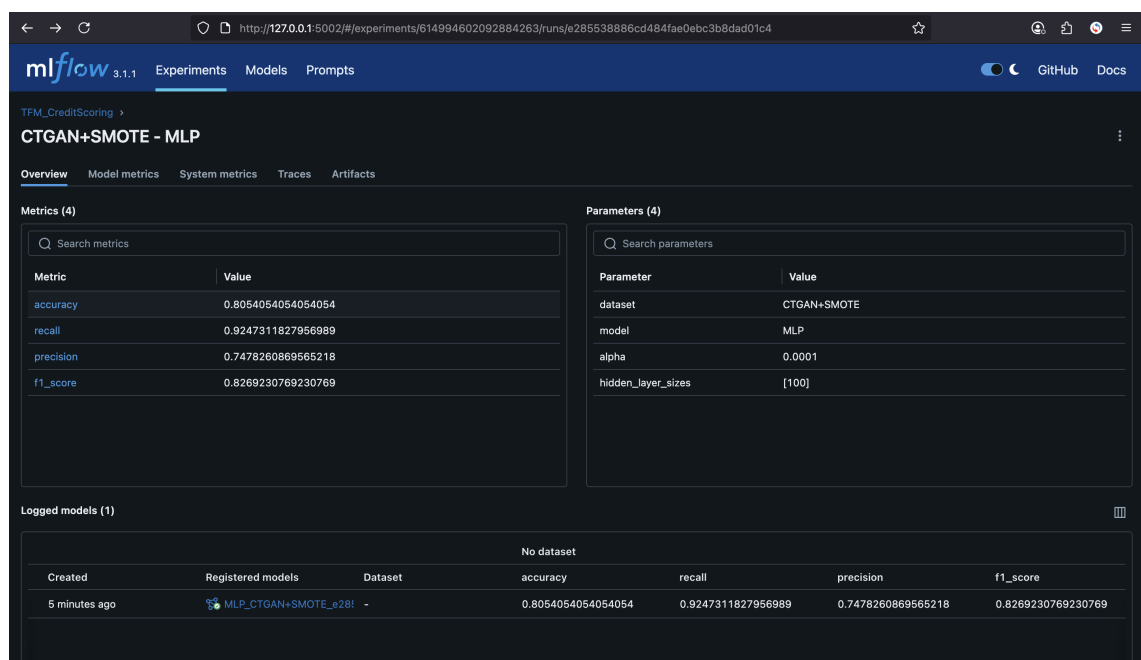


Figura 10: Los resultados del mejor modelo en MLflow

Para asegurar la reproducibilidad completa del entorno, se ha optado por una solución basada en contenedores Docker. En el archivo Dockerfile se especifica la imagen base, los

paquetes requeridos y las configuraciones necesarias para ejecutar el pipeline. Este contenedor incluye la instalación de dependencias listadas en `requirements.txt`, y permite iniciar sesiones de entrenamiento, análisis o inferencia en un entorno controlado. Complementariamente, el archivo `docker-compose.yml` orquesta la ejecución del contenedor y expone el servidor de MLflow en el puerto 5001, accesible desde entornos locales o remotos. Los volúmenes montados garantizan la persistencia de los registros de experimentos y modelos, incluso tras detener o reiniciar el contenedor.

El diseño modular del sistema, reforzado por el uso del archivo `config.py` para centralizar parámetros, rutas y configuraciones, permite modificar el comportamiento del pipeline sin alterar el código principal. Esto facilita el mantenimiento, la extensión del sistema y su adaptación a nuevos datasets, tareas de clasificación o contextos empresariales.

En conjunto, estas decisiones reflejan buenas prácticas de MLOps que van más allá del simple entrenamiento de modelos. El sistema está preparado para escalar, ser auditado y desplegado en producción, manteniendo siempre una trazabilidad total de las decisiones tomadas, los datos utilizados y los resultados obtenidos.

5.9. Monitorización y detección de data drift

Una vez desplegado un sistema de credit scoring basado en aprendizaje automático, uno de los desafíos clave es garantizar su validez en el tiempo. El fenómeno conocido como data drift, o deriva de datos, se refiere a los cambios en la distribución de las variables de entrada o en la relación entre las variables y la variable objetivo. Este cambio puede afectar gravemente al rendimiento del modelo, volviéndolo obsoleto o incluso perjudicial si no se detecta a tiempo.

En el presente trabajo no ha sido posible realizar una validación empírica del comportamiento del modelo ante nuevos datos reales, ya que las prácticas en la empresa colaboradora MytripleA finalizaron antes de poder acceder a registros futuros. Sin embargo, se propone una metodología para incorporar un sistema de supervisión activa que permita identificar indicios de data drift una vez que el modelo se encuentre en producción.

Una técnica común para detectar drift es comparar las distribuciones estadísticas de los datos de entrenamiento con los datos que llegan en producción. Para ello pueden emplearse métricas como el estadístico de Kolmogórov-Smirnov para variables numéricas, o la divergencia de Jensen-Shannon para comparar distribuciones categóricas. En modelos más

avanzados, pueden entrenarse clasificadores que intentan distinguir entre datos antiguos y nuevos; si logran hacerlo con alta precisión, es señal de que existe un cambio sustancial en las características de entrada.

Además de la comparación de distribuciones, se recomienda realizar un seguimiento periódico de las métricas de rendimiento del modelo (como F1-score, precisión o recall), siempre que exista retroalimentación o etiquetas disponibles en tiempo diferido. Una caída sistemática en estas métricas puede ser síntoma de concept drift, es decir, cambios en la relación entre entradas y la variable objetivo.

Para mitigar los efectos del drift, se pueden definir umbrales que activen mecanismos de alerta o reentrenamiento automático del modelo. Por ejemplo, si se detecta que la distribución de una variable crítica ha cambiado más de un determinado porcentaje respecto al entrenamiento, el sistema puede enviar una notificación al equipo de datos. Alternativamente, se puede establecer un calendario regular de reentrenamiento (por ejemplo, mensual o trimestral), integrando datos nuevos acumulados en ese período.

Como parte del trabajo futuro, se propone integrar librerías especializadas como evidently, River o alibi-detect, que permiten monitorizar de forma continua tanto el drift de variables como la estabilidad de las predicciones del modelo. Estas herramientas pueden integrarse fácilmente en un entorno MLOps basado en MLflow y Docker, como el desarrollado en este proyecto, y generar reportes automáticos para el equipo técnico.

En resumen, aunque no se ha podido evaluar el sistema con nuevos datos reales de la empresa, el pipeline desarrollado está preparado para incorporar módulos de detección de data drift. Su inclusión permitiría mantener la validez del modelo en el tiempo y asegurar que sus predicciones continúan siendo fiables, transparentes y ajustadas al entorno económico y operativo en constante evolución.

5.10. Validación técnica y funcional del sistema

La validación del sistema desarrollado se lleva a cabo mediante un conjunto de pruebas técnicas y funcionales destinadas a comprobar que el pipeline cumple con los objetivos definidos al inicio del proyecto. Estas pruebas evalúan tanto el correcto funcionamiento de cada módulo como la coherencia del sistema en su conjunto, asegurando que las tareas de ingestión, preprocesamiento, modelado, evaluación, explicabilidad y trazabilidad se ejecutan de manera fluida y reproducible.

Desde el punto de vista técnico, se verifica que el entorno de desarrollo puede ser replicado correctamente a través de los contenedores Docker definidos en los archivos `Dockerfile` y `docker-compose.yml`, los cuales encapsulan todas las dependencias y permiten levantar un entorno homogéneo en distintos equipos. También se comprueba el correcto registro y recuperación de experimentos mediante MLflow, validando que los modelos entrenados quedan asociados a sus parámetros, métricas, artefactos y versiones de código.

En cuanto a la funcionalidad, se prueban múltiples configuraciones del pipeline, ejecutando el script `train.py` bajo diferentes escenarios de entrada: datos originales, datos balanceados con SMOTE, datos generados con CTGAN y la combinación de ambos. Se evalúa la robustez del sistema al manejar configuraciones variables definidas en `parametros_modelos.yaml`, así como su capacidad de producir resultados coherentes y trazables en todas las ejecuciones. También se valida la capacidad del sistema para generar visualizaciones automáticas mediante `plots.py`.

El tiempo de ejecución del pipeline varía en función del modelo y de la técnica de enriquecimiento de datos aplicada. En promedio, el entrenamiento y evaluación de un modelo como Random Forest con datos sintéticos y validación cruzada requiere entre 3 y 5 minutos en entorno local, incluyendo el registro completo en MLflow. En el caso de CTGAN, la generación de datos sintéticos requiere tiempos más elevados, especialmente cuando se entrena durante varias épocas, pudiendo alcanzar hasta 10 minutos adicionales según la configuración.

Respecto a los requisitos funcionales planteados al inicio del trabajo, el sistema cumple con los siguientes puntos:

- Implementación modular del flujo de trabajo completo en aprendizaje automático.
- Soporte para datos reducidos mediante técnicas de oversampling y generación sintética.
- Entrenamiento y validación de múltiples modelos con trazabilidad completa.
- Registro de experimentos, métricas y artefactos con MLflow.
- Contenedorización y portabilidad del entorno con Docker.
- Incorporación de técnicas de explicabilidad como SHAP.

No obstante, durante el proceso de validación también se identifican algunas limitaciones. En primer lugar, la dependencia de datos sintéticos genera cierta incertidumbre sobre la capacidad real de generalización del sistema, ya que las técnicas como CTGAN pueden introducir ruido o correlaciones espurias. Además, no ha sido posible desplegar el sistema en entorno productivo ni testearlo con nuevos datos reales de la empresa, lo cual limita la validación en condiciones reales de operación. Finalmente, algunas herramientas de automatización (como pipelines CI/CD o monitorización activa) aún no han sido implementadas y se proponen como líneas de trabajo futuro.

En conjunto, la validación confirma que el sistema es funcional, reproducible y técnicamente sólido, cumpliendo con los objetivos clave del proyecto y quedando preparado para futuras mejoras orientadas a su despliegue real y supervisión continua.

5.11. Consideraciones éticas, legales y de negocio

El desarrollo de sistemas de inteligencia artificial aplicados al ámbito financiero, como el credit scoring, plantea importantes implicaciones éticas, legales y estratégicas. A lo largo del presente trabajo, se presta especial atención a estos aspectos, con el fin de garantizar que el sistema no solo sea técnicamente eficaz, sino también responsable y alineado con las exigencias regulatorias y los intereses de la empresa colaboradora MyTripleA.

Desde el punto de vista legal, se asegura el uso exclusivo de datos anonimizados. El dataset `companies_T_anon.parquet` no contiene información personal ni identificadores directos que permitan vincular los registros a entidades reales. Además, la información sensible ha sido transformada, truncada o cifrada de manera irreversible, cumpliendo así con lo estipulado en el Reglamento General de Protección de Datos (RGPD) y otras normativas europeas sobre privacidad. Esta medida permite realizar análisis y entrenar modelos sin comprometer la identidad ni los derechos de los usuarios o empresas analizadas.

En cuanto a las consideraciones éticas, se pone especial énfasis en la explicabilidad del sistema y en la equidad (fairness) de los modelos desarrollados. La incorporación de técnicas como SHAP permite entender cómo cada variable contribuye a las predicciones, lo que facilita auditorías internas, explicaciones ante reguladores y decisiones transparentes frente a los clientes. Esta trazabilidad ayuda a reducir el riesgo de sesgos ocultos o decisiones discriminatorias, al permitir identificar si ciertos grupos de empresas son sistemáticamente penalizados por patrones implícitos en los datos. Aunque no se lleva a cabo un análisis

de equidad exhaustivo por falta de variables sensibles (género, localización geográfica, tamaño exacto de empresa), el marco técnico queda preparado para integrar este tipo de validaciones en fases posteriores.

A nivel de negocio, el sistema propuesto se alinea directamente con los objetivos estratégicos de MyTripleA. La empresa busca mejorar sus procesos de evaluación de riesgo crediticio en entornos donde la información es escasa o parcial, como ocurre con muchas pymes o startups. El enfoque basado en small data y técnicas de enriquecimiento sintético permite ampliar el alcance de las decisiones crediticias, reduciendo la dependencia de históricos amplios y facilitando la inclusión de empresas jóvenes o con poca trayectoria financiera. Asimismo, la modularidad del sistema y su integración con herramientas como MLflow y Docker facilitan su mantenimiento y escalabilidad dentro de la infraestructura tecnológica de la compañía.

En definitiva, este trabajo busca demostrar que es posible construir soluciones de machine learning aplicadas al crédito que no solo sean técnicamente sólidas, sino también éticamente justificables, legalmente seguras y estratégicamente valiosas. El respeto a los principios de transparencia, protección de datos y responsabilidad algorítmica es una condición esencial para que este tipo de sistemas puedan ser adoptados de forma sostenible en el sector financiero.

5.12. Resumen del desarrollo y logros alcanzados

El desarrollo de este Trabajo de Fin de Máster ha dado lugar a la construcción de un sistema completo de credit scoring adaptado a contextos de datos limitados, combinando técnicas de aprendizaje automático, generación de datos sintéticos, explicabilidad y automatización mediante herramientas de MLOps. A lo largo del proceso, se ha implementado un pipeline modular, reproducible y extensible, diseñado para su posible integración en entornos reales como el de MyTripleA.

Entre los resultados más destacados se encuentra la mejora significativa en el rendimiento predictivo al aplicar técnicas de enriquecimiento de datos. La combinación secuencial de CTGAN y SMOTE ha permitido superar las limitaciones asociadas al desequilibrio de clases y al bajo volumen de observaciones, alcanzando un F1-score ponderado de 0.83 con el modelo MLP, considerado el mejor modelo global. La Tabla 3 resume los valores obtenidos por cada modelo bajo los distintos escenarios de generación y balanceo.

Para el conjunto de datos original, el modelo más efectivo fue Naive Bayes ($F1 = 0.58$). Con SMOTE aplicado, XGBoost alcanzó un rendimiento de 0.78, mientras que al utilizar únicamente CTGAN, el mejor resultado correspondió a XGBoost con un F1 de 0.38. La combinación de CTGAN y SMOTE permitió al MLP obtener el mejor rendimiento general ($F1 = 0.83$). Finalmente, en el escenario inverso (SMOTE seguido de CTGAN), el MLP volvió a destacar con un F1-score de 0.75.

Modelo	Orig	SM	CT	CT+SM	SM+CT
Gradient Boosting	0.47	0.75	0.31	0.79	0.66
KNN	0.42	0.59	0.34	0.74	0.59
Logistic Regression	0.52	0.62	0.29	0.62	0.62
MLP	0.54	0.75	0.28	0.83	0.75
Naive Bayes	0.58	0.60	0.34	0.66	0.59
Random Forest	0.56	0.72	0.12	0.81	0.67
SVC	0.53	0.69	0.23	0.75	0.48
XGBoost	0.46	0.78	0.38	0.79	0.68

Tabla 3: F1-score ponderado por modelo y tipo de datos

A nivel técnico, se ha construido un sistema completamente trazable y versionado. Cada etapa del flujo — desde la ingestión de datos hasta la evaluación final — ha sido registrada con MLflow, permitiendo comparaciones sistemáticas y facilitando la reproducibilidad. El uso de contenedores Docker garantiza la portabilidad del entorno, favoreciendo el despliegue en sistemas productivos sin alterar la lógica del modelo. Además, la documentación clara, la parametrización mediante archivos YAML y la organización modular del proyecto permiten su evolución eficiente.

Desde una perspectiva estratégica, la solución puede integrarse como apoyo en el proceso de evaluación crediticia en escenarios con información escasa. Su arquitectura abierta permite futuras extensiones, como la integración de dashboards interactivos, mecanismos de monitorización y detección de data drift, así como la incorporación de métricas orientadas a negocio.

En definitiva, este trabajo no solo ha alcanzado sus objetivos iniciales, sino que ha demostrado la viabilidad de aplicar técnicas avanzadas de machine learning al riesgo crediticio bajo condiciones de datos restringidos. Las contribuciones realizadas en diseño,

implementación y validación constituyen una base sólida para futuras investigaciones y despliegues en entornos reales.

6. Conclusiones y trabajo futuro

6.1. Síntesis de resultados

El presente trabajo se ha centrado en el desarrollo de un sistema modular y reproducible para la evaluación de riesgo crediticio en entornos de datos reducidos. En colaboración con la empresa MyTripleA, se ha implementado un pipeline de machine learning con principios de MLOps que integra módulos de preprocesamiento, generación de datos sintéticos, entrenamiento de modelos, explicabilidad y monitorización. La combinación de técnicas como CTGAN, SMOTE y algoritmos clásicos ha permitido mejorar el rendimiento predictivo, alcanzando un F1-score ponderado superior al baseline de la empresa en diversos escenarios de datos (0.83 frente a 0.63).

Los resultados obtenidos muestran que, con una arquitectura adecuada, es posible superar las limitaciones impuestas por el volumen reducido de datos, manteniendo la trazabilidad y reproducibilidad del proceso. Modelos como MLP, Random Forest y XGBoost destacan especialmente cuando se combinan datos generados con CTGAN y balanceados mediante SMOTE, obteniendo mejoras sustanciales frente a su entrenamiento en datos originales. El sistema responde adecuadamente a los requisitos técnicos, funcionales y éticos planteados al inicio del proyecto.

6.2. Contribuciones del trabajo

Una de las principales aportaciones de este trabajo reside en la integración de técnicas avanzadas en un entorno controlado y realista. A diferencia de propuestas meramente teóricas, el sistema ha sido desarrollado y validado con datos anonimizados de la empresa colaboradora, lo que refuerza su aplicabilidad en entornos financieros reales. Asimismo, se han establecido buenas prácticas de ingeniería como el uso de contenedores Docker, el seguimiento de experimentos con MLflow, la parametrización en YAML y la organización modular del código, lo que facilita su mantenimiento y escalabilidad.

Otra contribución destacada es la inclusión de mecanismos de explicabilidad. Gracias al uso de SHAP, se ha proporcionado al usuario una comprensión detallada de la importancia de las variables en cada predicción, tanto a nivel global como individual. Esto resulta

especialmente relevante en un contexto regulado como el financiero, donde la transparencia de los sistemas automatizados es clave para la confianza y aceptación por parte de usuarios y autoridades.

Desde un punto de vista académico, este trabajo contribuye a la literatura sobre credit scoring en small data, mostrando cómo técnicas modernas pueden adaptarse a escenarios de alta restricción informativa. También se abordan cuestiones éticas y legales, como el uso responsable de variables y la protección de datos personales, aspectos aún en desarrollo en muchas propuestas de inteligencia artificial aplicada.

6.3. Limitaciones encontradas

Pese a los logros alcanzados, se identifican varias limitaciones que condicionan la generalización de los resultados. En primer lugar, el trabajo se ha realizado sobre un conjunto de datos cerrado y estático, sin posibilidad de acceder a nuevas observaciones una vez finalizadas las prácticas en la empresa. Esto ha impedido evaluar el comportamiento del sistema en condiciones dinámicas, como las que se presentan tras su despliegue en producción.

Asimismo, no se ha podido validar la funcionalidad de detección de drift con datos reales posteriores al entrenamiento. Aunque se ha diseñado un sistema preparado para incorporar esta funcionalidad, su evaluación queda pospuesta para futuras fases del proyecto. También se reconoce que, debido a restricciones temporales, no se ha integrado una capa de AutoML, que permitiría automatizar la selección y optimización de modelos y parámetros de forma más eficiente.

Por último, si bien se ha hecho un esfuerzo considerable por incluir explicaciones accesibles, se reconoce la necesidad de desarrollar interfaces más intuitivas, como dashboards visuales interactivos, que faciliten el uso del sistema por perfiles no técnicos dentro de la empresa.

6.4. Líneas de trabajo futuro

Como continuación natural de este trabajo, se identifican varias líneas de mejora que podrían abordarse en futuras iteraciones para incrementar la robustez, utilidad práctica y alineación ética del sistema desarrollado.

Una de ellas consiste en desplegar el sistema en un entorno real de producción, evaluan-

do su rendimiento con datos en tiempo real y habilitando mecanismos de monitorización y reentrenamiento automático. Esta línea permitiría implementar estrategias de aprendizaje continuo para mantener el modelo actualizado frente a cambios estructurales en los datos (data drift). Además, se propone el uso de herramientas específicas como Evidently AI, River o Alibi-Detect para facilitar la detección de desviaciones y activar alertas tempranas.

También se considera relevante incorporar técnicas de auditoría de equidad. Aunque el dataset utilizado no contiene variables sensibles, podrían explorarse proxies como la región geográfica, el tamaño de la empresa o el sector económico. En este contexto, se recomienda el uso de métricas como Statistical Parity o Equal Opportunity, así como el análisis sistemático de posibles sesgos y la propuesta de estrategias de mitigación.

Otra línea de desarrollo es la integración de métricas de negocio, como el impacto económico de los errores tipo I y II, la tasa esperada de aprobación crediticia o la reducción potencial de impagos. Este tipo de indicadores permitiría evaluar mejor el valor práctico del modelo y alinear sus resultados con los objetivos estratégicos de la empresa.

En cuanto a la interacción con usuarios no técnicos, se propone el diseño de dashboards interactivos que permitan consultar los resultados del modelo y sus explicaciones (por ejemplo, mediante visualizaciones SHAP o curvas ROC) de forma accesible. Esto facilitaría la adopción y comprensión del sistema por parte de perfiles de negocio o gestión.

Desde el punto de vista de la ingeniería de software, una mejora clave consiste en la incorporación de un pipeline de integración y despliegue continuo (CI/CD). Aunque el proyecto ya utiliza herramientas como Docker y MLflow, se sugiere describir e implementar flujos completos con herramientas como GitHub Actions, Jenkins o Apache Airflow, que permitan automatizar pruebas, despliegues y actualizaciones del sistema de forma robusta y escalable.

Finalmente, se recomienda revisar y simplificar el resumen y las conclusiones del trabajo, con el objetivo de mejorar su claridad, facilitar una lectura ágil y destacar de manera más directa los principales logros y contribuciones del proyecto. Esta mejora será especialmente útil si se desea difundir el trabajo en contextos académicos o profesionales.

6.5. Valor estratégico del proyecto

El sistema desarrollado constituye un punto de partida sólido para la integración de soluciones de inteligencia artificial en la toma de decisiones financieras de pymes. Su arquitectura modular, reproducible y explicable permite una fácil adaptación a nuevos contextos y ofrece garantías suficientes de cumplimiento normativo. Además, refuerza la estrategia de innovación de MyTripleA, posicionando a la empresa en la vanguardia de la aplicación responsable de la inteligencia artificial en el sector fintech.

La combinación de aprendizaje automático, técnicas de generación sintética, y herramientas de seguimiento y explicación, permite no solo mejorar el rendimiento técnico del sistema, sino también generar confianza entre los usuarios y facilitar su integración en procesos existentes. En este sentido, el trabajo no solo ha cumplido sus objetivos técnicos y académicos, sino que también ha generado una base útil y escalable para el desarrollo de futuras soluciones en entornos financieros sensibles y regulados.

Agradecimientos

El presente Trabajo de Fin de Máster ha sido posible gracias a la colaboración y apoyo de diversas personas e instituciones, cuya contribución se reconoce con especial gratitud. En primer lugar, se agradece a la empresa MyTripleA la oportunidad de desarrollar este proyecto en un entorno real, así como el acceso a datos y el soporte ofrecido durante todo el proceso. Su implicación ha sido fundamental para asegurar la aplicabilidad práctica y el valor añadido de la solución propuesta.

También se expresa agradecimiento al tutor académico, José Manuel Picaza Garcia, por su orientación técnica, sus observaciones rigurosas y su acompañamiento continuo a lo largo del desarrollo del trabajo. Asimismo, se reconoce el esfuerzo y dedicación del equipo docente del Máster Universitario en Inteligencia Artificial de la Universidad Internacional de La Rioja (UNIR), por proporcionar una formación sólida y actualizada, esencial para la realización de este trabajo.

Referencias

Abdou, H. A., y Pointon, J. (2011). Credit scoring, statistical techniques and evaluation

- criteria: A review. *Intelligent Systems in Accounting, Finance and Management*, 18(2-3), 59–88.
- Ackerman, J., y cols. (2021). Auditing model drift in financial ml systems. *IEEE Transactions on Knowledge and Data Engineering*.
- Addy, W. A., Ajayi-Nifise, A. O., Bello, B. G., Tula, S. T., Odeyemi, O., y Falaiye, T. (2024). Ai in credit scoring: A comprehensive review of models and predictive analytics. *Global Journal of Engineering and Technology Advances*, 18(2), 118–129.
- Ashta, A., y Herrmann, H. (2021). Artificial intelligence and fintech: An overview of opportunities and risks for banking, investments, and microfinance. *Strategic Change*, 30(3), 211–222.
- Bedoya-Builes, N. J., Cardona, J. E., y Zapata-Álvarez, L. F. (2024). Confirming y factoring: Un estudio de caso de nuevas soluciones de financiación empresarial para las microempresas de medellín. *Ágora Revista Virtual de Estudiantes*(17), 58–84.
- Brahmandam, B. A. (2025). Mlops in finance: Automating compliance & fraud detection. *International Journal of Computer Trends and Technology*, 73(4), 35–41.
- Cao, L., Yang, Q., y Yu, P. S. (2021). Data science and ai in fintech: An overview. *International Journal of Data Science and Analytics*, 12(2), 81–99.
- Das, A., y Rad, P. (2020). Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*.
- Dastile, X., y Celik, T. (2021). Explainable ai in credit scoring: From tabular to visual interpretations. *Expert Systems with Applications*.
- Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., y Ranjan, R. (2023). Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9), 1–33.
- Faheem, A. (2021). Ai and financial inclusion: Potentials and pitfalls. *Journal of Financial Technology*.
- Gunnarsson, B. R., Vanden Broucke, S., Baesens, B., Óskarsdóttir, M., y Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, 295(1), 292–305.
- Gunning, D., y Aha, D. W. (2019). Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2), 44–58.
- Heng, Y. S., y Subramanian, P. N. (2022). A systematic review of machine learning and explainable artificial intelligence (xai) in credit risk modelling. En *Proceedings of the*

- future technologies conference (ftc)* (pp. 596–614). Cham: Springer.
- Hurley, M., y Adebayo, J. (2016). Credit scoring in the era of big data: Fairness, transparency, and discrimination. *Yale Journal of Law Technology*.
- Hurlin, C., Pérignon, C., y Saurin, S. (2024). The fairness of credit scoring models. *Management Science*.
- Kitchin, R., y Lauriault, T. P. (2015). Small data in the era of big data. *GeoJournal*, 80(4), 463–475.
- Kozodoi, N., Jacob, J., y Lessmann, S. (2022). Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*, 297(3), 1083–1094.
- Laborda, J., y Ryoo, S. (2021). Feature selection in a credit scoring model. *Mathematics*, 9(7), 746.
- Lăzăroiu, G., Bogdan, M., Geamănu, M., Hurloiu, L., Ionescu, L., y Ștefănescu, R. (2023). Artificial intelligence algorithms and cloud computing technologies in blockchain-based fintech management. *Oeconomia Copernicana*, 14(3), 707–730.
- Miljkovic, T., y Wang, P. (2025). A dimension reduction assisted credit scoring method for big data with categorical features. *Financial Innovation*, 11(1), 29.
- Misheva, B. H., Osterrieder, J., Hirsä, A., Kulkarni, O., y Lin, S. F. (2021). Explainable ai in credit risk management. *arXiv preprint arXiv:2103.00949*.
- Mohanty, A. (2025). Artificial intelligence in credit scoring: Enhancing financial inclusion & opportunities. *International Research Journal of Education and Technology*, 8(4), 2400–2409.
- Nallakaruppan, M. K., Balusamy, B., Shri, M. L., Malathi, V., y Bhattacharyya, S. (2024). An explainable ai framework for credit evaluation and analysis. *Applied Soft Computing*, 153, 111307.
- Oluwaferanmi, A. (s.f.). Integrating mlops and dataops for scalable and resilient machine learning deployment pipelines: Challenges, frameworks, and best practices.
- Qi, G.-J., y Luo, J. (2022). Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4), 2168–2187.
- Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., y Liu, P. (2021). Xgboost optimized by adaptive particle swarm optimization for credit scoring. *Mathematical Problems in Engineering*, 2021(1), 6655510.

- Rella, B. P. R. (2022). Mlops and dataops integration for scalable machine learning deployment. *International Journal for Multidisciplinary Research*, 4(1), 1–15.
- Sadok, H., Sakka, F., y El Maknouzi, M. E. H. (2022). Artificial intelligence and bank credit analysis: A review. *Cogent Economics & Finance*, 10(1), 2023262.
- Salama, K., Kazmierczak, J., y Schut, D. (2021). *Practitioners guide to mlops: A framework for continuous delivery and automation of machine learning*. <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>. (White paper, Google Cloud)
- Singla, A. (2023). Machine learning operations (mlops): Challenges and strategies. *Journal of Knowledge Learning and Science Technology*, 2(3), 333–340.
- Testi, A., y cols. (2022). A practical taxonomy for mlops: From development to deployment. *Machine Learning Engineering*.
- Thomas, L. C., Edelman, D. B., y Crook, J. N. (2002). *Credit scoring and its applications*. Philadelphia: Society for Industrial and Applied Mathematics.
- Trinh, T.-K., y Zhang, D. (2024). Algorithmic fairness in financial decision-making: Detection and mitigation of bias in credit scoring applications. *Journal of Advanced Computing Systems*, 4(2), 36–49.
- Vale, T., y cols. (2022). Legal boundaries of post-hoc explanations in ai systems. *AI Law*.
- Verma, S., y Rubin, J. (2018). Fairness definitions explained. En *Proceedings of the international workshop on software fairness* (pp. 1–7). ACM.
- Xu, L., Skoularidou, M., Cuesta-Infante, A., y Veeramachaneni, K. (2019). Ctgan: Conditional generative adversarial network for tabular data. *arXiv preprint arXiv:1907.00503*.
- Youssef, W. A. B., y Mansour, N. (2024). The factoring 2.0 in the era of the fintech revolution context. En *Digital technology and changing roles in managerial and financial accounting: Theoretical knowledge and practical application* (Vol. 36, pp. 37–51). Emerald Publishing Limited.
- Šušteršič, M., Mramor, D., y Zupan, J. (2009). Consumer credit scoring models with limited data. *Expert Systems with Applications*, 36(3), 4736–4744.

A. Apendices

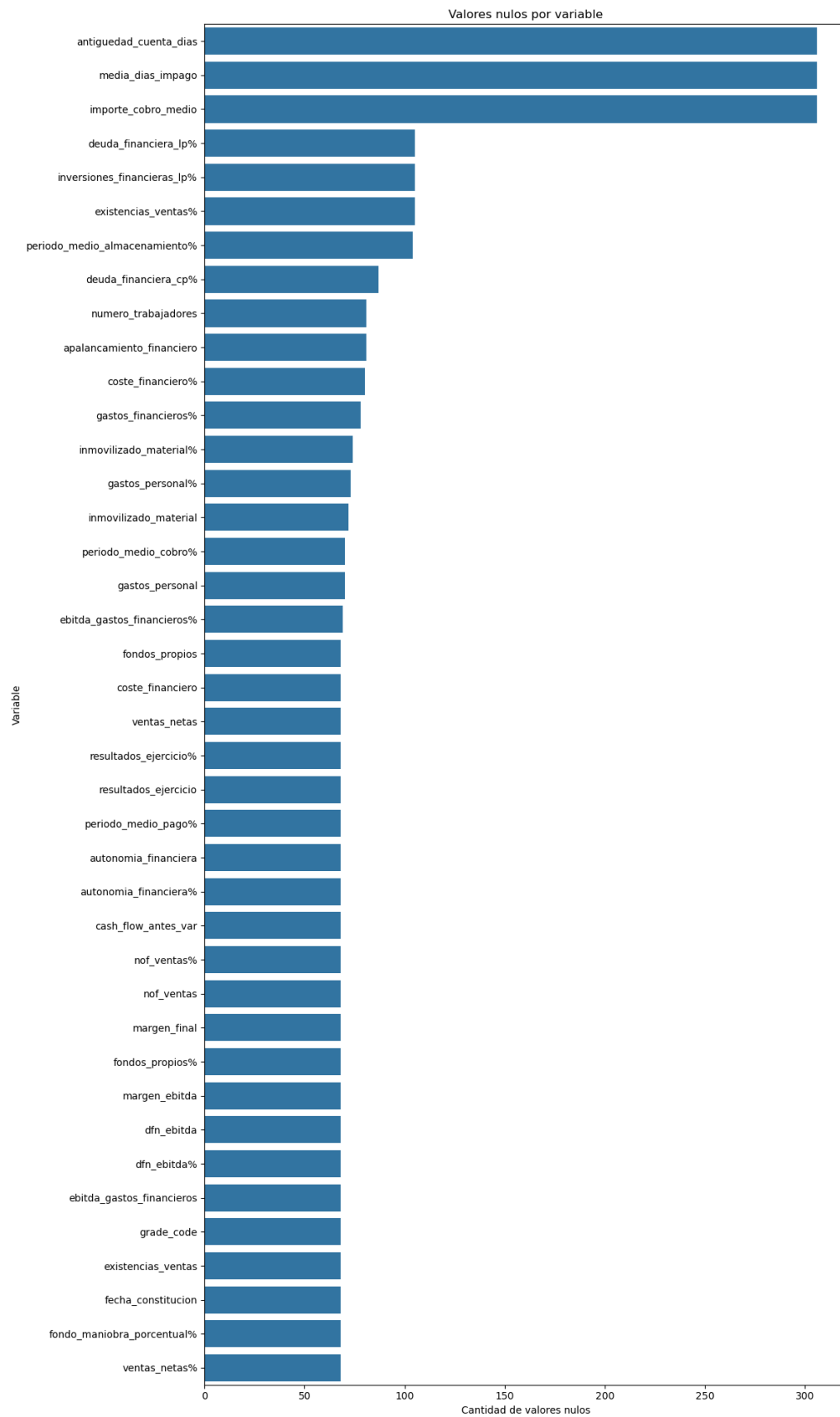


Figura 11: Valores nulos por variable

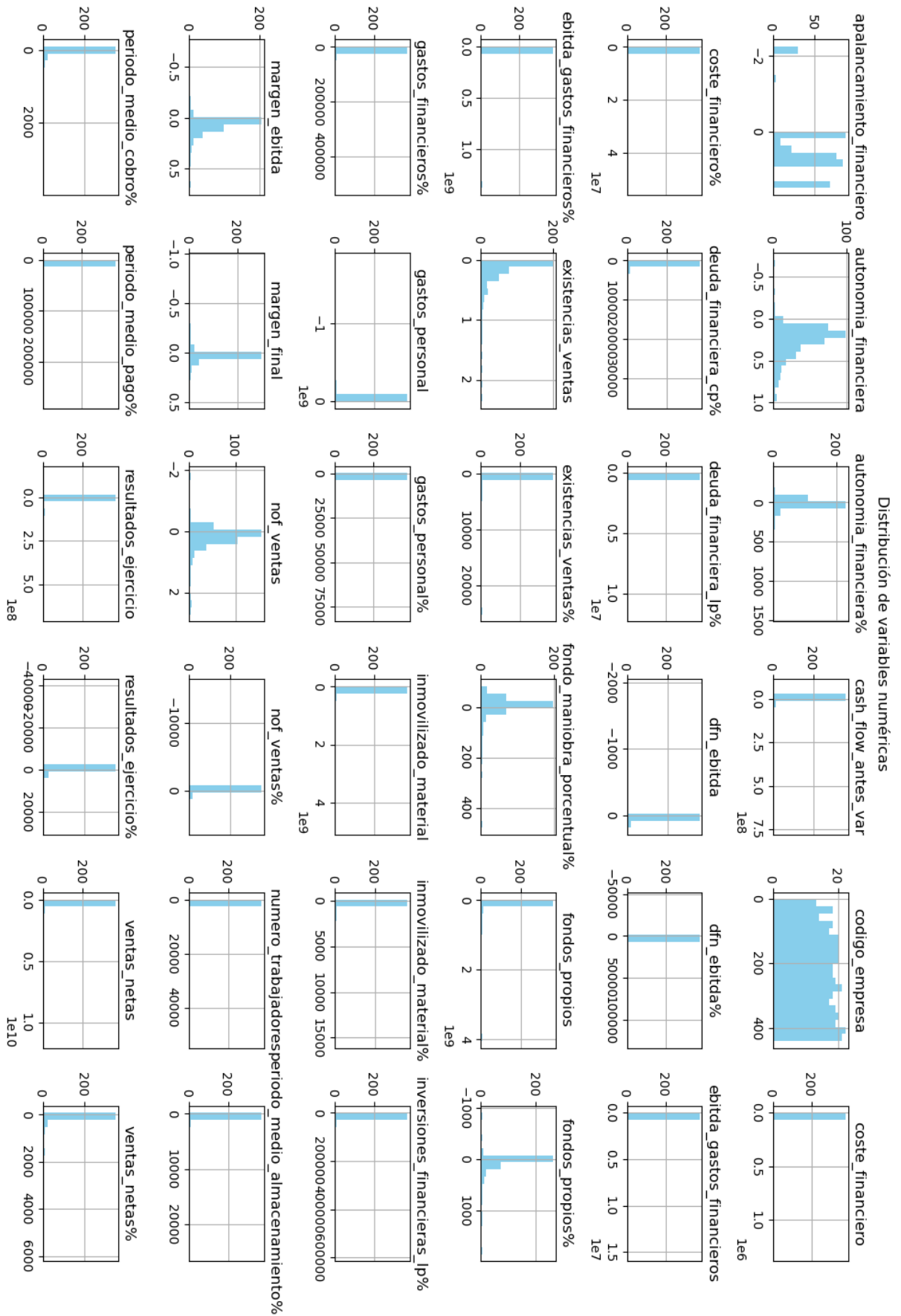


Figura 12: Distribución de variables numéricas

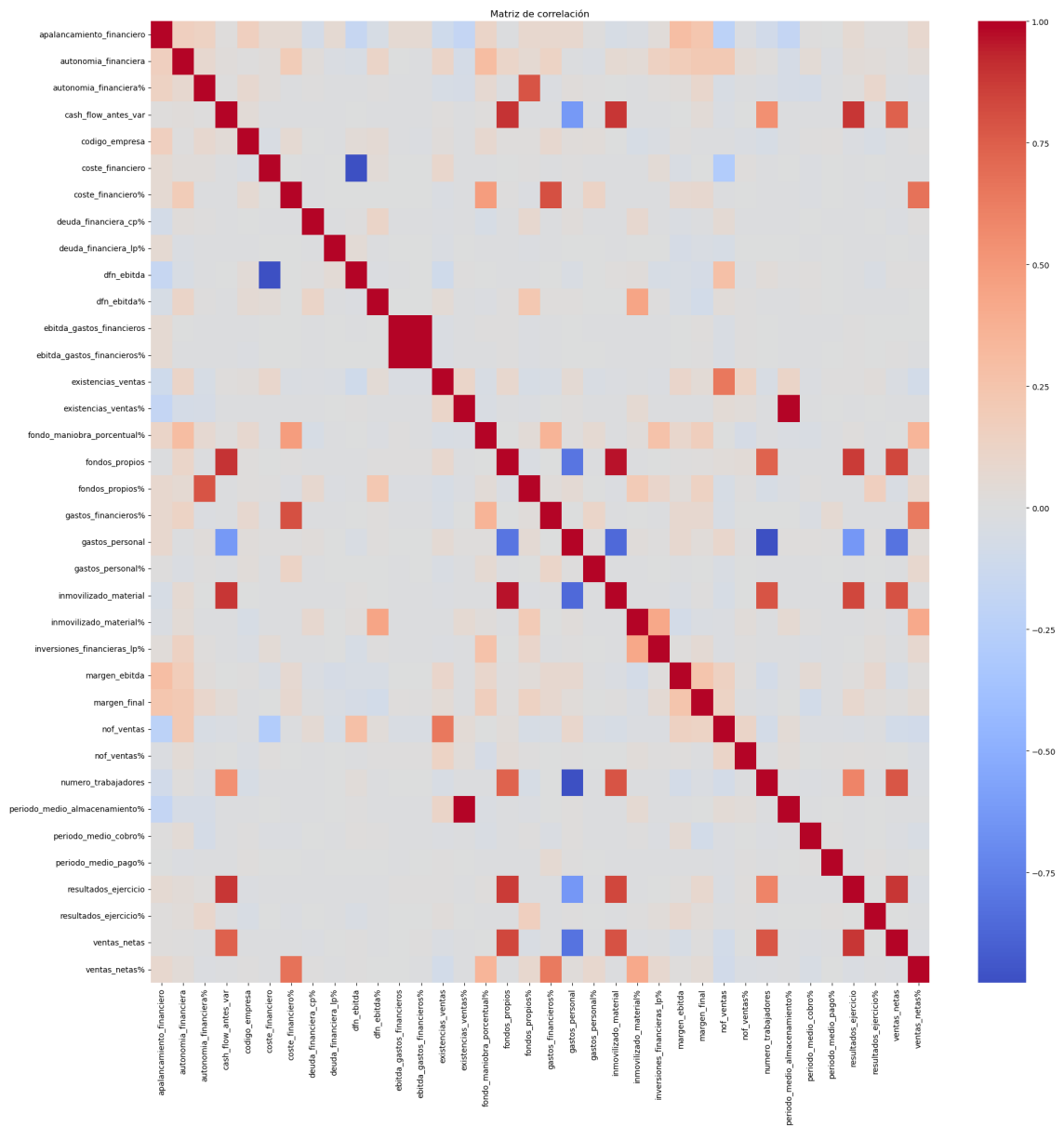


Figura 13: Matriz de correlación

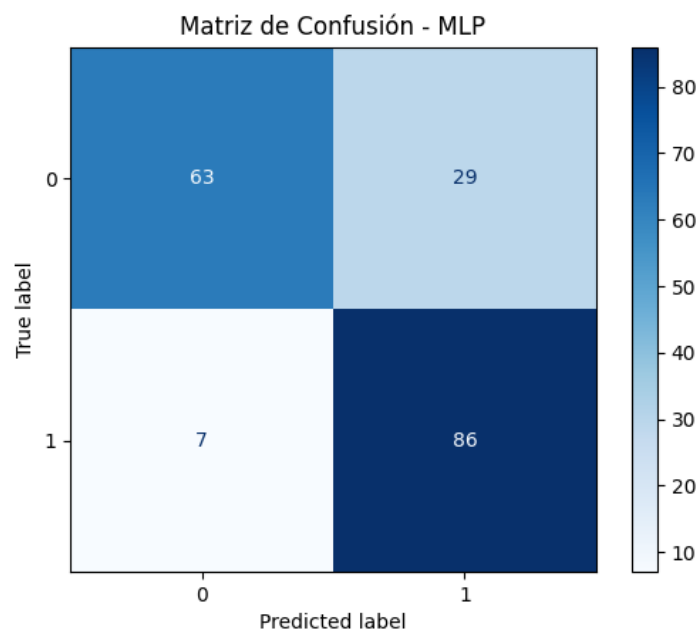


Figura 14: Matriz de confusión del modelo MLP

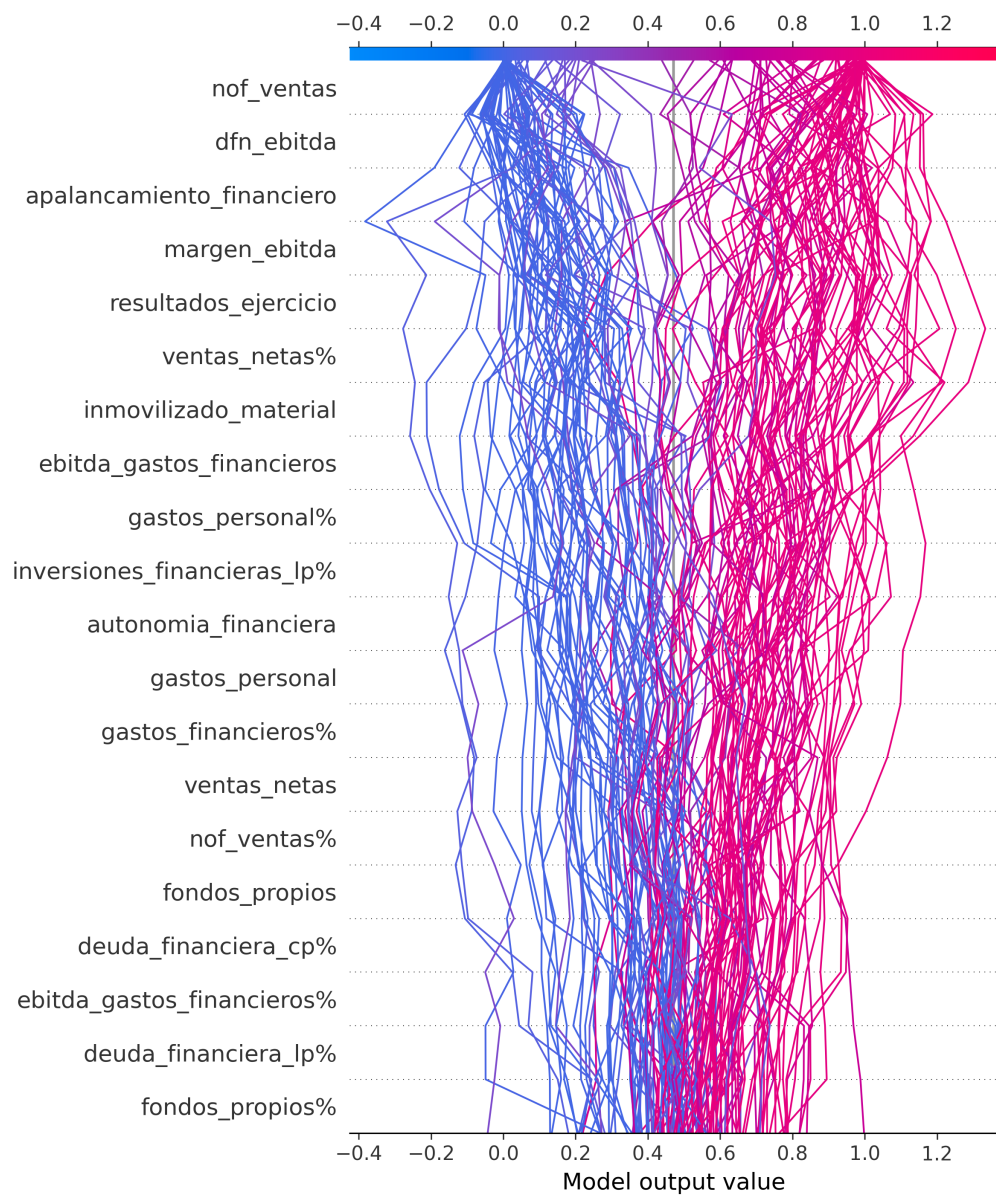


Figura 15: Trayectoria de decisiones (decision plot) – MLP