

Diplomatura en Ciencia de
Datos y Análisis
Avanzado

Sistema de Detección de Fraude con Tarjetas de Crédito usando Machine Learning

Trabajo Final - 2025

Arenas, Diego; Díaz, Augusto; Galermes,
Joaquín; Palazón, Agustina; Telis, Mónica;
Vidable, Ignacio

Octubre, 2025

Índice

- Resumen Ejecutivo..... 2
- Definición del Problema y Relevancia 3
- Datos y Metodología..... 4
 - Exploración de Datos (EDA) 4
 - Procesamiento y Metodología 4
- Modelado y Evaluación 5
 - Algoritmos probados 5
 - Metodología de entrenamiento..... 5
 - Resultados comparativos 6
 - Modelo seleccionado..... 6
- Impacto en el Negocio..... 8
 - Beneficios esperados 8
 - Riesgos y limitaciones 9
 - Costos del proyecto 9
 - Escalabilidad y próximos pasos 10
- Conclusiones y Recomendaciones 10
 - Conclusiones principales 10
 - Recomendaciones 11

Resumen Ejecutivo

El fraude con tarjetas de crédito representa una de las principales amenazas para el sistema financiero y para la confianza de los clientes en los servicios digitales. Detectar estas operaciones fraudulentas en tiempo real es un desafío crítico, ya que la gran mayoría de las transacciones son legítimas y el porcentaje de fraude es muy bajo, lo que genera un fuerte desbalance de clases.

El presente proyecto propone el desarrollo de un **sistema de detección de fraude basado en técnicas de Machine Learning**, utilizando el dataset de transacciones de tarjetas de crédito ampliamente reconocido en la comunidad académica. La solución incluye un proceso integral: desde la exploración y limpieza de los datos, la ingeniería de variables y el entrenamiento de modelos, hasta la implementación de un dashboard de monitoreo de resultados.

El impacto esperado en el negocio es significativo: **reducción de pérdidas económicas por transacciones fraudulentas, aumento en la eficiencia de los equipos de fraude operativo y mejora en la confianza de los usuarios finales**. El sistema se diseñó con enfoque escalable, de modo que pueda adaptarse a escenarios productivos reales con altos volúmenes de datos.

Entre los resultados clave obtenidos se destacan:

- **Modelos probados:** Regresión Logística, Random Forest, XGBoost y LightGBM.
- **Métrica principal de evaluación:** AUC-ROC, por su capacidad de balancear recall y precisión en contextos de desbalance.
- **Modelo final seleccionado:** El modelo seleccionado fue XGBoost, con AUC-ROC = 0.983 y AUC-PR = 0.878. Ajustando el umbral de decisión por costo esperado (FN=100, FP=1) se obtuvo Recall = 0.888, Precision = 0.664 y F1 = 0.760, con TP=87, FP=44, FN=11, TN=56.820. Este ajuste reduce fraudes no detectados a un nivel operativo razonable para el negocio.

Repositorio público:

[GitHub - Grupo G](#)

Video Youtube:

[Grupo G - Video Youtube](#)

Definición del Problema y Relevancia

El fraude con tarjetas de crédito constituye una de las principales preocupaciones de bancos, emisores y fintechs a nivel global. Según estadísticas internacionales, las pérdidas por fraude superan miles de millones de dólares anuales, lo que afecta no sólo la rentabilidad de las instituciones financieras, sino también la confianza de los usuarios en el sistema de pagos.

El desafío central es detectar transacciones fraudulentas en tiempo real con alta precisión y sin generar un exceso de falsos positivos. Esto implica un problema complejo: por un lado, el número de fraudes es extremadamente bajo en comparación con el total de transacciones ($<0,2\%$ en datasets típicos); por otro, los patrones de fraude evolucionan constantemente, lo que requiere sistemas adaptativos y con capacidad de actualización continua.

La motivación de este proyecto surge de la necesidad de contar con herramientas basadas en ciencia de datos y machine learning que permitan identificar transacciones sospechosas de manera temprana, reduciendo pérdidas económicas y aumentando la efectividad de los equipos de monitoreo.

El objetivo concreto de este trabajo es desarrollar, evaluar e implementar un modelo de machine learning capaz de:

1. Analizar transacciones históricas de tarjetas de crédito.
2. Aprender patrones de comportamiento asociados al fraude.
3. Predecir, con un nivel de precisión elevado, la probabilidad de que una transacción entrante sea fraudulenta.

Desde el punto de vista estratégico, el proyecto aporta un valor clave al negocio al:

- Reducir costos operativos y pérdidas financieras.
- Aumentar la eficiencia de los procesos de monitoreo de fraude.
- Mejorar la confianza de los clientes en la seguridad de los sistemas financieros digitales.

Detectar fraude implica priorizar no dejar pasar transacciones fraudulentas (FN) aun a costa de revisar algunas legítimas (FP). Por eso, además de comparar modelos por AUC-PR, optimizamos el umbral con una función de costo simple: costo (FN) = 100 y costo (FP) = 1. El umbral óptimo resultó 0.065, que minimiza la pérdida esperada y eleva la cobertura de fraude.

Datos y Metodología

Para el desarrollo del proyecto se utilizó el dataset público “Credit Card Fraud Detection” (<https://www.kaggle.com/mlg-ulb/creditcardfraud/data>), que contiene transacciones realizadas con tarjetas de crédito en Europa durante dos días de septiembre de 2013.

- Cantidad de registros: 284.807 transacciones.
- Cantidad de variables: 31 columnas (28 componentes obtenidos por PCA, más Time, Amount y la variable objetivo Class).
- Variable objetivo:
 - 0 = transacción legítima (99.83% de los casos).
 - 1 = transacción fraudulenta (0.17% de los casos).

Este dataset presenta un desbalance de clases extremo, lo que representa uno de los principales desafíos para el modelado, ya que los algoritmos tienden a sesgarse hacia la clase mayoritaria.

Exploración de Datos (EDA)

- La variable Amount presenta una distribución muy sesgada, con la mayoría de las transacciones en montos bajos.
- La variable Time refleja la secuencia temporal de las operaciones, útil para estudiar patrones de fraude en función del horario.
- Los 28 componentes principales (V1–V28) resultan de una reducción de dimensionalidad por PCA aplicada para proteger la confidencialidad de los datos.

Procesamiento y Metodología

Se aplicaron los siguientes pasos:

1. Limpieza y validación de datos: chequeo de valores faltantes (no presentes en este dataset).
2. Normalización de variables: ajuste en Amount y Time para mejorar la estabilidad del modelo.
3. Muestreo y balanceo: técnicas de *undersampling*, *oversampling* y *SMOTE* para abordar el desbalance.

4. Ingeniería de variables: creación de features derivados y normalización temporal.
5. Metodología CRISP-DM: se siguió un ciclo iterativo de comprensión del negocio, comprensión de los datos, preparación, modelado, evaluación y despliegue.

Modelado y Evaluación

Para abordar el problema de detección de fraude, se entrenaron y compararon distintos algoritmos de Machine Learning, seleccionados por su desempeño en contextos de clasificación binaria con fuerte desbalance de clases.





Algoritmos probados

1. **Regresión Logística:** modelo base, interpretable y rápido de entrenar.
2. **Árboles de Decisión:** modelo que segmenta el espacio de decisión en reglas simples, ayuda a entender relaciones no lineales y detectar interacciones entre variables.
3. **Random Forest:** algoritmo de ensamble, robusto frente a ruido y desbalance.
4. **XGBoost:** técnica de boosting altamente eficiente, recomendada para problemas con gran desbalance.
5. **LightGBM:** alternativa de boosting optimizada para datasets grandes y alta velocidad.

Metodología de entrenamiento

- Validación cruzada estratificada para preservar la proporción de clases en cada fold.
- Métricas de evaluación:
 - AUC-ROC: mide la capacidad del modelo para separar fraudes de transacciones legítimas.
 - Recall (sensibilidad): porcentaje de fraudes correctamente detectados.
 - Precision: proporción de alertas que realmente son fraude.
 - F1-score: balance entre recall y precision.

Resultados comparativos

	AUC_ROC	AUC_PR	RECALL	PRECISION	F1	TN	FP	FN	TP
 LOGISTIC REGRESSION	0.970952	0.712809	0.908163	0.055452	0.104521	55348	1516	9	89
 DECISION TREE	0.856949	0.543970	0.714286	0.760870	0.736842	56842	22	28	70
 XGBOOST	0.983255	0.877673	0.846939	0.864583	0.855670	56851	13	15	83
 RANDOM FOREST	0.967009	0.864970	0.765306	0.949367	0.847458	56860	4	23	75
LIGHTGBM	0.768613	0.004236	0.846939	0.004691	0.009330	39253	17611	15	83

Modelo seleccionado

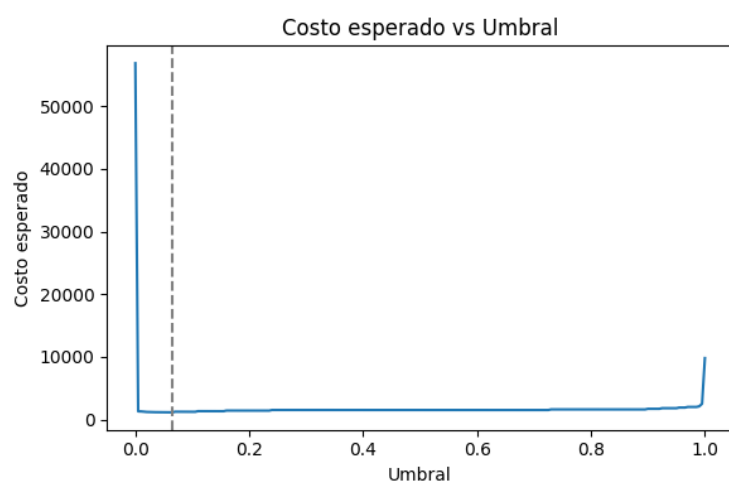
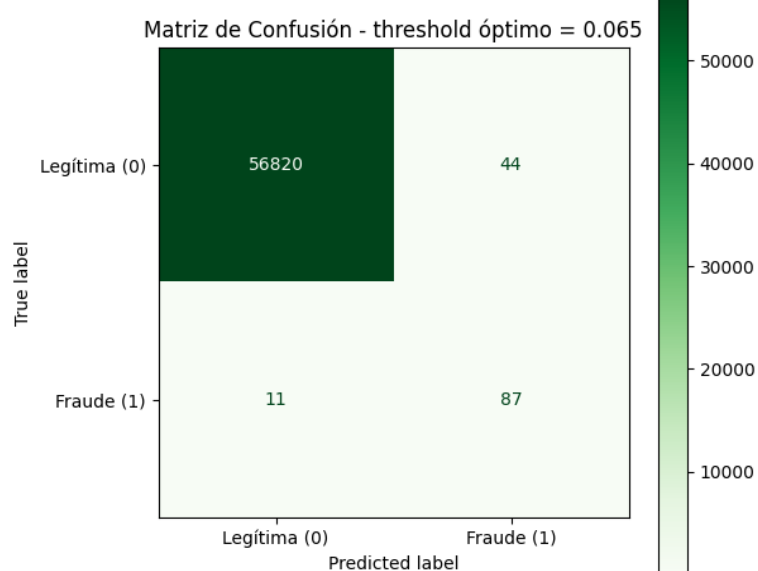
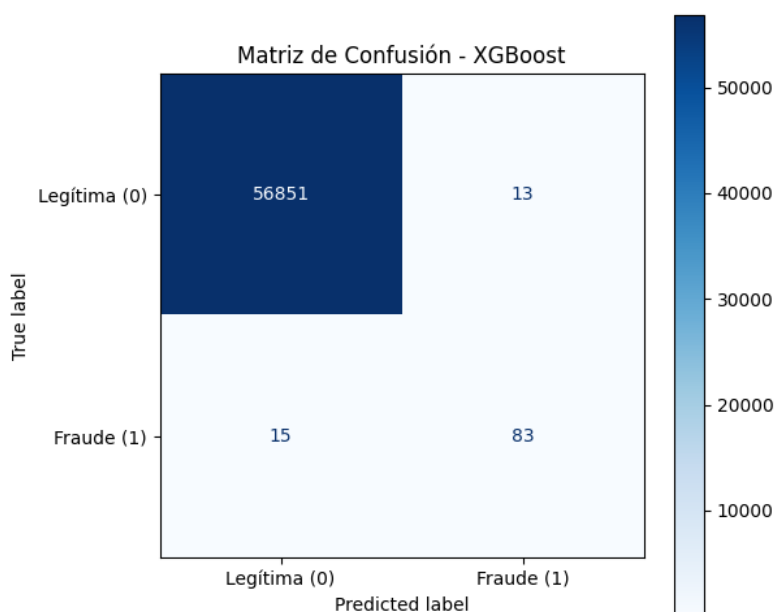
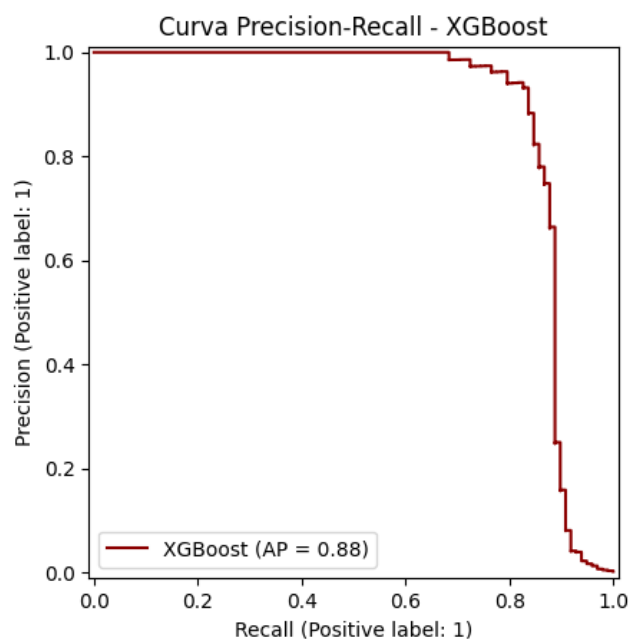
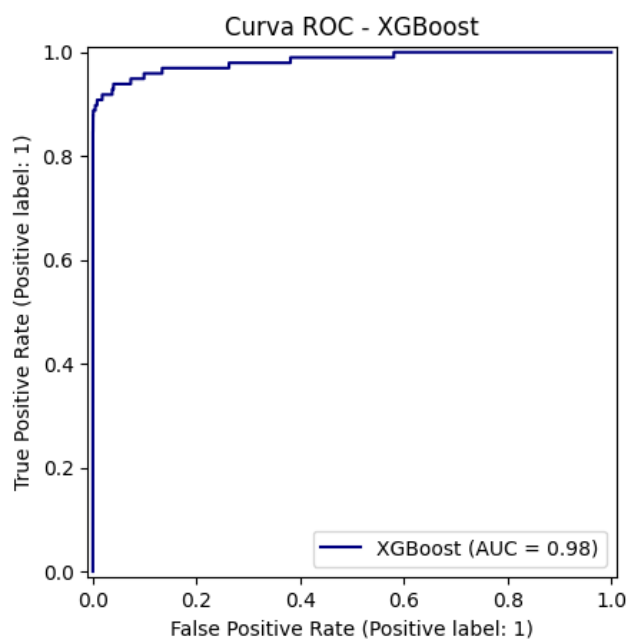
El modelo final elegido debido a su desempeño fue XGBoost, que alcanzó un AUC-PR de 0,878 y un AUC-ROC de 0,983, recall (84%) y precisión (86%) minimizando tanto los falsos negativos (fraudes no detectados) como los falsos positivos (transacciones legítimas clasificadas como fraude). Random Forest y LightGBM presentaron resultados competitivos, aunque levemente inferiores en AUC-PR.

Además de las métricas estándar, se aplicó un análisis de optimización de umbral para reflejar el impacto económico de los errores de clasificación. Se definió un esquema donde un Falso Negativo (FN) tiene un costo 100 veces mayor que un Falso Positivo (FP).

El barrido de thresholds determinó un umbral óptimo de 0,065, con un costo esperado mínimo de 1.144. En este punto, el modelo logra un mejor equilibrio entre detectar fraudes y no sobrerrecargar al equipo con falsos positivos.

Con el threshold óptimo (0,065), el modelo alcanzó:

- TP = 87, FP = 44, FN = 11, TN = 56.820.
- Precision = 0,664, Recall = 0,888, F1-score = 0,760.



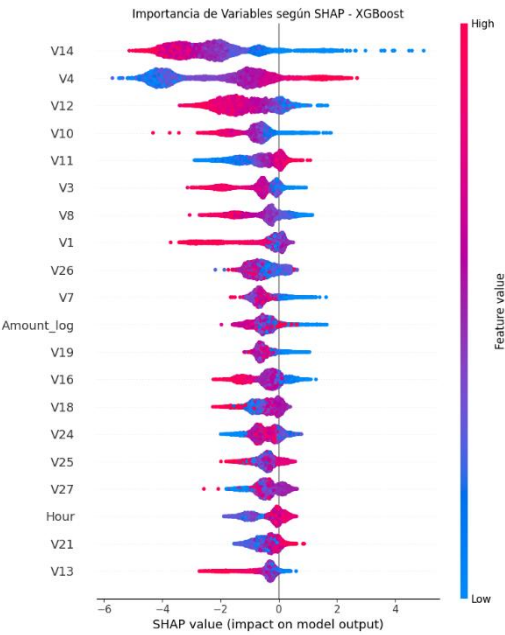
En comparación:

- Baseline (0,5): Precision = 0,865, Recall = 0,847, FN = 15, FP = 13.
- Óptimo (0,065): Precision = 0,664, Recall = 0,888, FN = 11, FP = 44.

La decisión de reducir el umbral incrementa los falsos positivos, pero mejora la detección de fraudes, lo cual resulta más valioso desde la perspectiva de negocio.

	Escenario	Threshold	TP	FP	FN	TN	Precision	Recall	F1	AUC_ROC	AUC_PR
0	Baseline	0.5	83	13	15	56851	0.8646	0.8469	0.8557	0.9833	0.8777
1	Óptimo por costo	0.065	87	44	11	56820	0.6641	0.8878	0.7598	0.9833	0.8777

A su vez, se aplicó el método SHAP (SHapley Additive exPlanations) para analizar la importancia de las variables en el modelo XGBoost. Las características más influyentes fueron V14, V4, V12, V10 y V11. Aunque las variables provienen de una transformación PCA y no son interpretables directamente, el análisis confirma que el modelo se apoya en un conjunto diverso de factores y no en una sola variable.



Impacto en el Negocio

La implementación de un sistema de detección de fraude basado en Machine Learning tiene un impacto directo y medible en los resultados financieros y en la confianza de los clientes.

Beneficios esperados

1. Reducción de pérdidas económicas: al detectar de manera temprana transacciones fraudulentas, se evita que las instituciones absorban los costos asociados.
2. Eficiencia operativa: disminuye la carga manual de revisión, ya que el sistema prioriza las operaciones con mayor probabilidad de fraude.
3. Mejora en la experiencia del cliente: al reducir falsos positivos, menos usuarios legítimos se ven afectados por bloqueos o rechazos innecesarios.
4. Escalabilidad: el modelo puede adaptarse a volúmenes crecientes de transacciones sin incrementar proporcionalmente los costos operativos.

Riesgos y limitaciones

- Desbalance de clases: la baja proporción de fraudes reales puede generar métricas engañosas si no se eligen correctamente.
- Drift de datos: los patrones de fraude cambian en el tiempo; el modelo requiere monitoreo y reentrenamiento regular.
- Labels demorados: la confirmación de fraude puede tardar días o semanas, lo que afecta la calibración y el monitoreo en tiempo real.
- Restricciones de integración: los sistemas de origen y las regulaciones de privacidad (ej. PII) pueden limitar la velocidad de adopción.

Costos del proyecto

Tiempo:

- Lean: 8–10 semanas
- Típico: 10–14 semanas
- Enterprise: 14–18 semanas

Nota: Se estima un equipo de 2–3 personas de tiempo completo (Data Scientist, Data Engineer/MLOps, y un apoyo parcial en Software/Security). Cada FTE (Full-Time Employee) trabaja ~40 h semanales.

- Escenario lean: unas 800–1.000 horas totales de trabajo, concentradas en un equipo muy experimentado y con tareas en paralelo.
- Entrega típica: 1.200–1.600 horas, incluyendo iteraciones, revisiones con stakeholders y ajustes de acceso a datos.
- Enterprise: 1.600–2.200 horas, ya que se suman entornos múltiples, auditorías y procesos de cambio más estrictos.

Costo único (mano de obra):

- Típico: USD 30k–100k
- Enterprise: USD 80k–180k

Nota: La mayor parte se destina a trabajo de Data Engineering/MLOps (pipelines, orquestación, monitoreo) y Data Science (EDA, modelado, métricas, optimización). Una fracción menor va a desarrollo del dashboard, seguridad/gobernanza y QA/documentación.

Costo mensual (cloud/software):

- Típico: USD 400–2,300
- Enterprise: USD 2k–6k

Nota: Los costos recurrentes provienen de almacenamiento (datasets y features), cómputo (entrenamientos e inferencia batch), orquestación (Airflow/Prefect/Composer), consultas en warehouse (BigQuery/Snowflake/Postgres), monitoreo/observabilidad, y hosting del dashboard.

Escalabilidad y próximos pasos

El sistema desarrollado es de carácter prototípico, pero su arquitectura puede ampliarse hacia escenarios de producción. Entre las posibles mejoras se destacan:

- Automatización del pipeline: implementar procesos de entrenamiento y validación continua con retroalimentación de nuevos casos de fraude.
- Predicción en tiempo real: incorporar APIs de scoring para decisiones inmediatas en línea con los sistemas transaccionales.
- Monitoreo de negocio: desplegar dashboards con indicadores clave como tasa de fraude detectado, ahorro estimado y nivel de impacto en clientes

Conclusiones y Recomendaciones

El presente trabajo permitió demostrar que es factible desarrollar un sistema de detección de fraude con tarjetas de crédito mediante Machine Learning, capaz de identificar transacciones sospechosas con un alto nivel de precisión y recall, incluso en un contexto de fuerte desbalance de clases.

Conclusiones principales

1. El dataset presenta un desbalance extremo ($\approx 0,17\%$ de fraudes), lo que exige el uso de métricas adecuadas (AUC-PR, F1) y técnicas específicas de preparación.
2. Los modelos de boosting (XGBoost, LightGBM) superaron a la regresión logística y Random Forest, especialmente en AUC-PR y Recall.
3. El modelo seleccionado, XGBoost, alcanzó un AUC-PR de 0,878 y un AUC-ROC de 0,983. Con el umbral optimizado por costo (0,065), logró un Recall de 0,888 y un F1-score de 0,760, reduciendo fraudes no detectados con un volumen controlado de falsos positivos.

4. Desde la perspectiva de negocio, el sistema aporta valor tangible: menor exposición a pérdidas económicas, mayor eficiencia operativa y fortalecimiento de la confianza de los clientes.

Recomendaciones

- Monitoreo continuo del modelo: implementar alertas de data drift y performance, ya que los patrones de fraude cambian con el tiempo.
- Reentrenamiento periódico: incorporar ciclos de actualización (ej. mensual o trimestral) con nuevas transacciones y etiquetas confirmadas.
- Despliegue escalable: avanzar hacia APIs de scoring en tiempo real y dashboards de monitoreo con KPIs de negocio.
- Colaboración con equipos de fraude: ajustar thresholds y métricas según el costo real de falsos positivos vs falsos negativos.
- Cumplimiento normativo: reforzar prácticas de seguridad, encriptación y manejo de PII para garantizar conformidad con regulaciones financieras.