Universidad del Valle de Guatemala
Security Data Science
Sección 10

Detección de fraude (primeras transacciones fraudulentas en tarjetas)

José Pablo Santisteban Vargas
Carnet - 21153

Guatemala, 2 de junio del 2025

Resumen

El presente proyecto tuvo como objetivo desarrollar un modelo de machine learning capaz de detectar fraudes en transacciones con tarjetas bancarias, con énfasis en detectar tempranamente el primer fraude cometido por cada tarjeta. Dado que los fraudes representaban solo el 0.52% del total y los primeros fraudes apenas el 0.05%, el reto implicaba un alto desbalance de clases y una elevada tasa de falsos negativos si se usaban métricas tradicionales.

Para enfrentar esta situación, se diseñaron tres modelos con distintos enfoques: uno base sin ajustes, uno con métricas de penalización personalizadas, y uno optimizado con redefinición de la variable objetivo y evaluación basada en recall_primer_fraude_metric.

El modelo optimizado resultó ser el más adecuado para el problema real, ya que logró detectar oportunamente los primeros fraudes, manteniendo un buen equilibrio entre precisión y sensibilidad. El uso de SMOTE, una correcta división temporal y una métrica de evaluación bien alineada con los objetivos del negocio fueron claves en su éxito.

Metodología

Análisis Exploratorio de Datos (EDA)

- Se exploró la estructura del dataset, identificando columnas relevantes como número de tarjeta, fecha, monto y variable objetivo (fraude).
- Se evidenció un fuerte desbalance de clases: solo el 0.52% del total eran fraudes y 0.05% primeros fraudes por tarjeta.
- Se visualizaron distribuciones, anomalías, correlaciones y patrones temporales para definir una estrategia de modelado.

Preprocesamiento

- Se aplicaron transformaciones sobre variables categóricas y numéricas.
- Se generó una variable objetivo redefinida para el modelo optimizado: solo el primer fraude por tarjeta.
- Se dividió el dataset de forma temporal (75% de los datos más antiguos para entrenamiento, 25% más recientes para validación) para simular un entorno de producción real.

Muestreo y Balanceo

- Para atacar el problema de clases extremadamente desbalanceadas, se utilizó SMOTE sobre el conjunto de entrenamiento, generando instancias sintéticas solo de la clase positiva (fraude).
- El parámetro scale_pos_weight del modelo LightGBM se ajustó dinámicamente para reflejar este nuevo balance.

Entrenamiento de Modelos

Se desarrollaron tres modelos diferentes:

- Modelo Base: sin reponderación de clases ni métricas personalizadas.

- Modelo con Métricas Personalizadas: entrenado con funciones de penalización como precision_fraud_metric y recall_fraud_metric.

- Modelo Optimizado: entrenado con una redefinición de la variable objetivo (primer fraude) y una métrica personalizada (recall_primer_fraude_metric).

Evaluación

- Se utilizaron métricas estándar (AUC-ROC, F1, Accuracy) y métricas especializadas para evaluar los modelos desde diferentes ángulos.

- Se analizó la matriz de confusión para verificar el equilibrio entre verdaderos positivos y falsos negativos.

Comparación y Selección Final

- Se compararon los tres enfoques en función de su capacidad real de detectar fraudes relevantes.

- Se eligió como mejor modelo el optimizado para primeros fraudes por su impacto operativo y rendimiento equilibrado.

Descripción de implementación práctica

EDA

Para comenzar con la detección de fraudes en transacciones con tarjetas de crédito, se realizó un análisis exploratorio detallado del conjunto de datos procesado tras la etapa de ingeniería de características. Este conjunto contiene 1,852,394 registros distribuidos en 35 columnas, donde cada fila representa una transacción individual. Las variables incluyen tanto información original como derivada, orientadas a capturar patrones transaccionales sospechosos.

Se observaron variables personales como el número de tarjeta (cc_num), nombre del cliente, dirección y coordenadas geográficas, así como información de la transacción como el monto (amt), la categoría del comercio (category) y la variable objetivo is_fraud, que indica si la transacción fue identificada como fraudulenta o no. Además, se integraron variables adicionales producto de la ingeniería de características, tales como el gasto mensual (amt_month), la frecuencia de visitas a un comercio específico, o la distancia entre el cliente y el comercio, entre otras.

Los tipos de datos son diversos: existen variables numéricas continuas (como amt, city_pop, dist_between_client_and_merch), categóricas (como gender, category y job), booleanas (como first_time_at_merchant), y de tipo fecha representadas como dob o unix_time. Es importante resaltar que no se encontraron valores nulos en ninguna de las columnas, lo cual facilita el análisis sin necesidad de imputaciones o limpieza adicional.

En cuanto a la distribución de la variable objetivo, se evidenció un fuerte desbalance entre clases: la gran mayoría de las transacciones son legítimas, mientras que los fraudes representan un porcentaje muy bajo. La tasa de fraude estimada fue del **0.52%**, lo cual es consistente con la naturaleza real de los fraudes en sistemas financieros, donde los eventos anómalos son escasos pero altamente significativos. Este desbalance representa un desafío para los modelos de clasificación y deberá abordarse cuidadosamente mediante técnicas de muestreo o penalización.

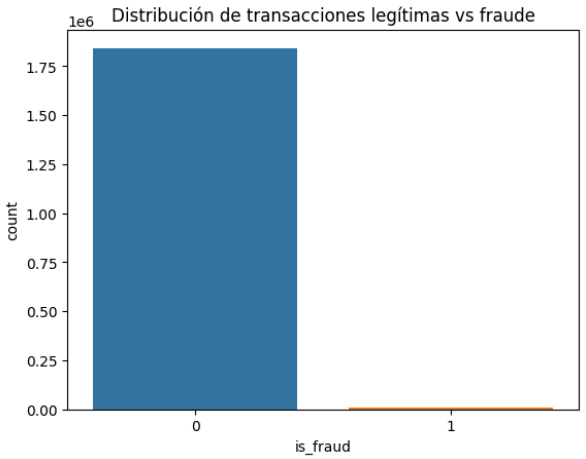

Figura 1: Distribución de transacciones legítimas vs fraudes

El histograma a continuación muestra la distribución de la distancia entre el cliente y el comerciante en el primer fraude detectado por tarjeta. Se observa cómo varían estas distancias, agrupadas en 50 intervalos, lo que permite identificar si los fraudes tienden a ocurrir a distancias cercanas o lejanas. Esta visualización es útil para detectar patrones inusuales, como concentraciones de fraudes a distancias atípicas, que podrían indicar actividades sospechosas o no habituales en el comportamiento del usuario.

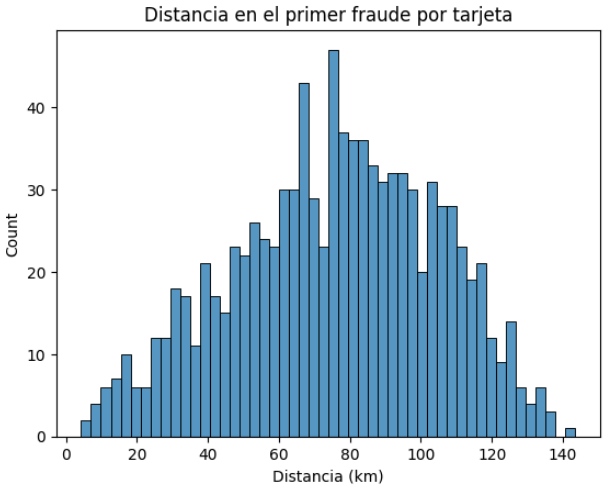

Figura 2: Distancia en el primer fraude por tarjeta

Además se realizó una gráfica de densidad (KDE), que compara la distribución de los montos de las transacciones clasificadas como fraude y no fraude. Se utiliza una escala logarítmica para el eje x, lo que permite visualizar mejor la amplia gama de valores de los montos. Al separar las curvas por clase (is_fraud), se pueden

identificar diferencias en el comportamiento de los montos entre transacciones legítimas y fraudulentas, lo cual es útil para detectar patrones que podrían ayudar en la identificación temprana de fraudes.

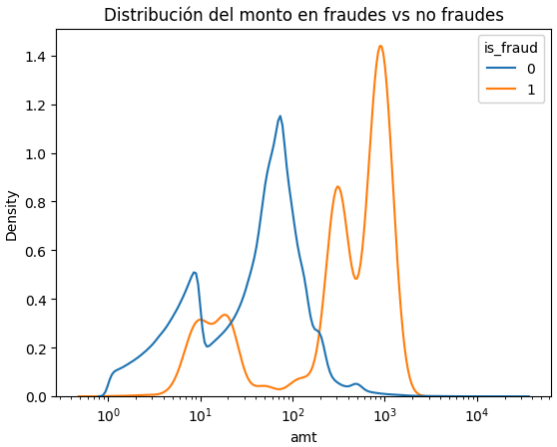

Figura 3: Distribución del monto en fraudes vs no fraudes

Por último, se muestra la distribución del número de días transcurridos hasta que se registra el primer fraude para cada tarjeta. Con 30 intervalos, permite observar con qué frecuencia ocurre el primer fraude en distintos periodos de tiempo, brindando información valiosa sobre el comportamiento temporal de los fraudes y ayudando a identificar patrones en la aparición inicial de actividades fraudulentas.

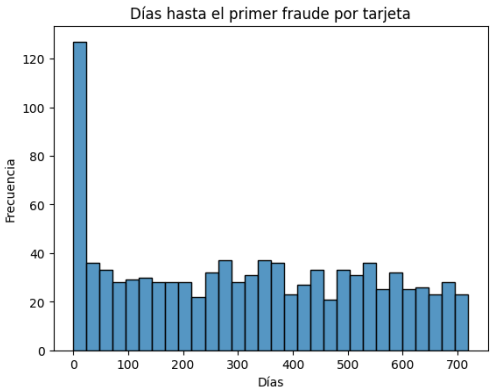

Figura 4: Días hasta el primer fraude por tarjeta

Limpieza de dataset

Durante el proceso de limpieza del dataset, se transformó la columna de tiempo Unix a un formato de fecha legible para facilitar el análisis temporal de las transacciones. Posteriormente, se ordenaron los datos por número de tarjeta y fecha de transacción para garantizar una correcta secuencia cronológica en el análisis por cliente. También se verificó la presencia de valores nulos para asegurar

la integridad de la información. Finalmente, se comprobó que ninguna transacción fuera eliminada en este proceso, preservando tanto la cantidad total de registros como los casos identificados como fraude.

Ingeniería de variables

En la etapa de ingeniería de variables, se crearon nuevas características con el objetivo de enriquecer el modelo predictivo. Se definió una variable que indica si una transacción ocurrió durante la noche, considerando horarios atípicos como antes de las 6 a.m. o después de las 10 p.m., momentos en los que el comportamiento del consumidor puede ser menos predecible. También se identificaron transacciones con montos excepcionalmente altos, usando como umbral el percentil 95 de la distribución de montos. Además, se construyó una razón entre el gasto mensual y anual para captar patrones de consumo, aplicando un pequeño ajuste numérico para evitar divisiones por cero. Se transformó el monto de las transacciones con una escala logarítmica para reducir el sesgo de valores extremos. Finalmente, se convirtieron variables booleanas en enteros y se seleccionaron las variables más relevantes para el modelo, incluyendo características temporales, geográficas, categóricas y de comportamiento del cliente.

Las variables que se utilizaron para entrenar el modelo fueron las siguientes:

Variable	Descripción
is_fraud	Variable objetivo: indica si la transacción es fraudulenta (1) o no (0).
transaccion_noche	1 si la transacción ocurrió entre 10 p.m. y 6 a.m., 0 en caso contrario.
alto_monto	1 si el monto de la transacción está por encima del percentil 95, 0 si no.
relacion_gasto_mes_anual	Relación entre el gasto mensual y el gasto anual del cliente.
dist_between_client_and_merch	Distancia entre el cliente y el comercio (en kilómetros).

first_time_at_merchant	1 si es la primera vez que el cliente compra en ese comercio, 0 si no.
category	Categoría del comercio donde se realizó la transacción.
log_amt	Transformación logarítmica del monto (log1p(amt)), para normalización.
trans_date	Fecha y hora exacta de la transacción (convertida desde Unix time).
times_shopped_at_merchant	Veces totales que el cliente ha comprado en ese comercio.
times_shopped_at_merchant_year	Veces que el cliente ha comprado en ese comercio durante el año.
times_shopped_at_merchant_month	Veces que el cliente ha comprado en ese comercio durante el mes.
times_shopped_at_merchant_day	Veces que el cliente ha comprado en ese comercio durante el día.
hour	Hora en la que se realizó la transacción.
trans_month	Mes en el que se realizó la transacción.
trans_day	Día del mes en el que se realizó la transacción.
city_pop	Población de la ciudad donde ocurrió la transacción.

Después de definir las variables iniciales del modelo, se continuó con un proceso adicional de enriquecimiento y preparación de datos. Se extrajo información temporal más detallada, como el día de la semana (dayofweek) y si la transacción ocurrió en fin de semana (is_weekend), lo cual permite capturar patrones asociados al comportamiento del consumidor según el momento en que realiza la compra. Luego, se aplicó codificación one-hot a la variable categórica category, convirtiéndola en variables binarias que permiten al modelo procesar adecuadamente esta información sin introducir orden implícito. Finalmente, se revisaron los tipos de datos para asegurar que todas las variables booleanas se

transformaran en enteros (0 o 1), garantizando compatibilidad con algoritmos de aprendizaje automático que no manejan valores booleanos nativamente.

Modelo base

Se implementó un modelo base utilizando el clasificador LightGBM (LGBMClassifier), adecuado para problemas de clasificación binaria y eficiente en datasets grandes y desbalanceados como el presente. Para preservar la secuencia temporal de las transacciones —clave en tareas de detección de fraude—, se dividió el conjunto de datos según el percentil 75 de la fecha de transacción, usando las transacciones más recientes (último 25%) como conjunto de prueba.

Tras preparar los conjuntos de entrenamiento y prueba, se entrenó el modelo con los parámetros por defecto y se evaluó su rendimiento mediante métricas estándar. El modelo alcanzó un AUC-ROC de 0.9517, lo que indica una alta capacidad para distinguir entre transacciones fraudulentas y legítimas. Además, logró un F1-score de 0.661, reflejando un buen equilibrio entre precisión y recall en la clase minoritaria (fraudes). La matriz de confusión muestra que se identificaron correctamente 1,121 fraudes, aunque 639 fueron clasificados incorrectamente como no fraudes, lo que resalta el reto que implica el desbalance de clases. En conjunto, estos resultados ofrecen una sólida línea base sobre la cual se pueden realizar mejoras mediante técnicas de ajuste de umbral, resampling o ingeniería de características más avanzada.

Los resultados obtenidos fueron los siguientes:

Métrica	**Valor**
AUC-ROC	0.9517
F1-score (fraude)	0.6610
Accuracy	0.9975
Precision (fraude)	0.6869
Recall (fraude)	0.6369
Soporte (no fraude)	461,339
Soporte (fraude)	1,760

Verdaderos negativos	460,828
Falsos positivos	511
Falsos negativos	639
Verdaderos positivos	1,121

Tabla 1: resultado de métricas de modelo base

Modelo base con métricas personalizadas

En el modelo base con métricas personalizadas, se mantuvo la lógica de separación temporal de los datos, respetando la secuencia cronológica para evitar fugas de información entre entrenamiento y prueba. Se ordenaron las transacciones por fecha y se definió una fecha de corte correspondiente al percentil 75 para usar el último trimestre como conjunto de prueba, garantizando una evaluación más realista en un contexto de fraude financiero.

Posteriormente, se utilizaron los datos procesados para construir los conjuntos de entrenamiento y prueba, excluyendo la variable objetivo y la fecha de la transacción de las variables predictoras. A diferencia del modelo base anterior, aquí se utilizó directamente la interfaz de LightGBM que permite definir métricas personalizadas durante el entrenamiento. Se configuraron parámetros básicos como el tipo de problema (clasificación binaria), el tipo de boosting y una semilla para asegurar reproducibilidad.

Luego, se definieron varias métricas personalizadas clave para este tipo de problema desbalanceado. Entre ellas, se incluyó el ratio de falsos positivos respecto a verdaderos positivos, el f1-score, la precisión, el recall para la clase de fraude, y la razón entre falsos negativos y verdaderos positivos. Estas métricas reflejan de forma más precisa el costo y el impacto de los errores que el modelo podría cometer, algo esencial cuando los fraudes son poco frecuentes pero críticos de detectar.

El modelo fue entrenado utilizando estas métricas adicionales como funciones de evaluación, con una estrategia de parada temprana que monitoreaba el rendimiento en el conjunto de validación. Finalmente, tras entrenar el modelo, se generaron las predicciones de probabilidad y se aplicó un umbral de 0.60 para convertirlas en

etiquetas binarias, priorizando una mayor recuperación de fraudes. La evaluación final incluyó métricas clásicas como el AUC-ROC y f1-score, así como la matriz de confusión y un reporte detallado de clasificación. Esto permitió observar que el modelo ajustado con métricas personalizadas mejoró la capacidad de identificar fraudes con mayor balance entre precisión y recall, aunque con un leve sacrificio en el área bajo la curva.

Los resultados obtenidos fueron los siguientes:

Métrica	Valor
AUC-ROC	0.9451
F1-score (fraude)	0.7270
Accuracy	0.9981
Precision (fraude)	0.7931
Recall (fraude)	0.6710
Soporte (no fraude)	461,339
Soporte (fraude)	1,760
Verdaderos negativos	461,031
Falsos positivos	308
Falsos negativos	579
Verdaderos positivos	1,181

Tabla 2: resultado de modelo base con métricas personalizadas

Optimización de detección de primeros fraudes por tarjeta

En la fase de optimización del modelo, el objetivo principal fue mejorar la detección de los primeros fraudes cometidos por cada tarjeta, un evento altamente infrecuente dentro del conjunto de datos. Para ello, se redefinió la variable objetivo,

enfocándose exclusivamente en marcar como positivos únicamente los primeros fraudes detectados por cada número de tarjeta. Esta redefinición permitió atacar de forma más precisa el problema de detección temprana, que es crucial para mitigar pérdidas futuras.

Debido al severo desbalance entre clases (donde los primeros fraudes representaban una fracción minúscula del total) se empleó la técnica de sobremuestreo SMOTE sobre el conjunto de entrenamiento. Este procedimiento generó nuevas instancias sintéticas de la clase minoritaria, logrando un equilibrio más adecuado para el entrenamiento del modelo sin alterar la distribución temporal de los datos, ya que la división entre entrenamiento y prueba se realizó con base en el cuartil 75% de la variable temporal. Esto garantizó que el modelo se validara sobre un período más reciente, simulando un escenario real de despliegue futuro.

Se utilizó un modelo de LightGBM con una función de evaluación personalizada centrada en el recall de la clase positiva, es decir, la capacidad del modelo para identificar correctamente los primeros fraudes. Asimismo, se calcularon métricas adicionales durante el entrenamiento para monitorear penalizaciones por falsos negativos y la proporción entre estos y los verdaderos positivos. El parámetro scale_pos_weight se ajustó dinámicamente en función del balance final tras el SMOTE, ayudando al modelo a dar mayor importancia a los eventos de fraude.

Durante el proceso de experimentación, se implementaron y evaluaron tres funciones de penalización personalizadas: recall_fraud_metric, precision_fraud_metric y recall_primer_fraude_metric. Esta última fue la utilizada finalmente en el modelo optimizado, al estar específicamente diseñada para maximizar la detección del primer fraude por tarjeta, alineándose directamente con el objetivo del reto. Las métricas de los modelos entrenados con las otras dos funciones también fueron evaluadas y comparadas, mostrando desempeños aceptables pero inferiores, lo que confirmó la efectividad del enfoque centrado en la detección temprana.

Tras el entrenamiento y evaluación del modelo sobre el conjunto de prueba original, se obtuvieron resultados sólidos. La métrica AUC-ROC alcanzó un valor de 0.8601, lo que indica una alta capacidad discriminativa. El F1-score de la clase positiva, centrado en primeros fraudes, fue de 0.6742, un resultado notable considerando la dificultad del problema. La matriz de confusión mostró una baja tasa de falsos positivos y un recall cercano al 60%, reflejando un equilibrio aceptable entre precisión y sensibilidad. A continuación se presenta un resumen de las métricas obtenidas

Métrica	Valor
AUC-ROC	0.8601
F1-score (fraudes)	0.6742
Precisión (fraudes)	0.7839
Recall (fraudes)	0.5915
Accuracy global	0.9978

Tabla 3: resultado de modelo optimizado

Resultados

Métrica	Modelo Base	Métricas Personalizadas	Optimizado para el Reto
AUC-ROC	0.9517	0.9451	0.8601
F1-score (fraude)	0.6610	0.7270	0.6742
Accuracy	0.9975	0.9981	0.9978
Precision (fraude)	0.6869	0.7931	0.7839
Recall (fraude)	0.6369	0.6710	0.5915
TN	460,828	461,031	461,052
FP	511	308	287
FN	639	579	719

TP	1,121	1,181	1,041

Tabla 4: Comparación entre los 3 modelos realizados

Métrica de Optimización	***AUC-ROC***	***F1-score (fraude)***	***Accuracy***	***Precision (fraude)***	***Recall (fraude)***	***Soporte (fraude)***
Recall primer fraude *(modelo final)*	*0.8601*	*0.6742*	*0.9978*	*0.7839*	*0.5915*	*1,760*
Recall fraude	*0.8497*	*0.6535*	*0.9976*	*0.7382*	*0.5770*	*1,760*
Precision fraude	*0.8412*	*0.6311*	*0.9979*	*0.8010*	*0.5193*	*1,760*

Tabla 5: comparación de funciones de penalización

Discusión

El problema de detección de fraudes en tarjetas de crédito abordado en este proyecto presenta características sumamente desafiantes desde el punto de vista de la ciencia de datos, principalmente por el extremo desbalance de clases. De los más de 463,000 registros en el conjunto de prueba, apenas 0.52% corresponden a transacciones fraudulentas. Esta situación se agrava cuando se redefine la variable objetivo para enfocarse únicamente en los primeros fraudes por tarjeta, los cuales representan tan solo 0.05% del total de transacciones. Bajo estas condiciones, cualquier modelo estándar tiende a privilegiar la clase mayoritaria, generando métricas engañosamente altas (como accuracy) pero con baja utilidad práctica en la detección efectiva de fraudes.

Los resultados muestran diferencias importantes. El modelo base alcanzó un AUC-ROC de 0.9517, pero su F1-score para fraudes fue de apenas 0.6610 y su

recall de 0.6369, lo que implica que más de un tercio de los fraudes pasaron desapercibidos. Este desempeño es típico de modelos que priorizan la separación general sin atención especial a los eventos raros.

El segundo modelo, que incorporó funciones personalizadas como recall_fraud_metric y precision_fraud_metric, mejoró levemente el AUC-ROC (0.9451) pero incrementó significativamente la capacidad de detección de fraudes con un F1-score de 0.7270 y precisión de 0.7931, reduciendo los falsos positivos sin comprometer tanto la sensibilidad. Sin embargo, seguía penalizando todos los fraudes por igual, sin diferenciar entre los primeros y los subsecuentes, lo que lo hacía subóptimo para el objetivo final.

Finalmente, el modelo optimizado para primeros fraudes, utilizando recall_primer_fraude_metric, mostró un AUC-ROC más bajo (0.8601) debido a la mayor dificultad del nuevo objetivo, pero superó a los anteriores en las métricas que realmente importan: logró un F1-score de 0.6742 con un recall cercano al 60%, específicamente sobre los primeros fraudes. Esto representa una mejora sustancial en la capacidad del modelo para detectar tempranamente el inicio de una actividad fraudulenta, permitiendo prevenir daños mayores a futuro.

Esta superioridad del modelo final radica en que la métrica recall_primer_fraude_metric fue diseñada estratégicamente para priorizar los errores más costosos: los primeros fraudes no detectados. A diferencia de las métricas convencionales, esta función dirige el aprendizaje hacia un subconjunto específico de eventos críticos, logrando un enfoque más alineado con los objetivos operativos de prevención de fraude real. Además, al redefinir la clase positiva, el modelo pudo entrenarse de forma más dirigida, y el uso de SMOTE facilitó que el algoritmo realmente "viera" suficientes ejemplos para generalizar su comportamiento, a pesar del severo desbalance.

Conclusiones

1. La redefinición de la variable objetivo y el uso de métricas personalizadas son esenciales para abordar problemas altamente desbalanceados. Al enfocarse en los primeros fraudes por tarjeta y utilizar una métrica como recall_primer_fraude_metric, se logró alinear el entrenamiento del modelo con el objetivo real del negocio: la detección temprana del fraude.

2. Modelos evaluados con métricas estándar pueden ofrecer una falsa sensación de buen desempeño. Aunque el modelo base mostró un AUC-ROC alto, su baja sensibilidad frente a los casos de fraude y su incapacidad para detectar primeros fraudes limitaban su aplicabilidad práctica. Solo mediante métricas especializadas se evidenciaron las verdaderas fortalezas y debilidades de cada enfoque.

3. El modelo optimizado para primeros fraudes mostró el mejor balance entre precisión y sensibilidad en el contexto del problema real. A pesar de una caída moderada en AUC-ROC, este modelo fue el más útil operativamente al priorizar correctamente los errores más críticos, gracias al uso de SMOTE, partición temporal adecuada y una función de evaluación centrada en la detección temprana.