SECURITY DATA SCIENCE
SECCIÓN - 10
Manuel Rodas, 21509

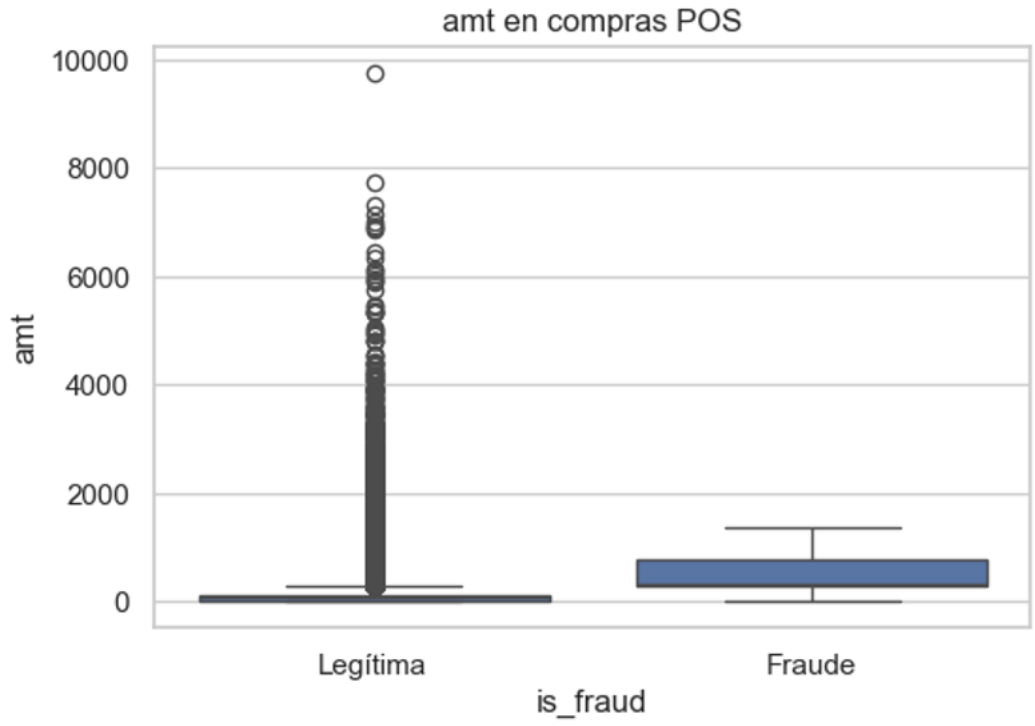
Minimización de Falsos Positivos en Compras Presenciales

Objetivo del Proyecto

El objetivo principal fue desarrollar un modelo de machine learning capaz de detectar fraudes en transacciones presenciales, minimizando la cantidad de falsos positivos (transacciones legítimas clasificadas erróneamente como fraude). Este enfoque es crucial para evitar fricciones en la experiencia del cliente, sin comprometer la capacidad del sistema para detectar fraudes reales.

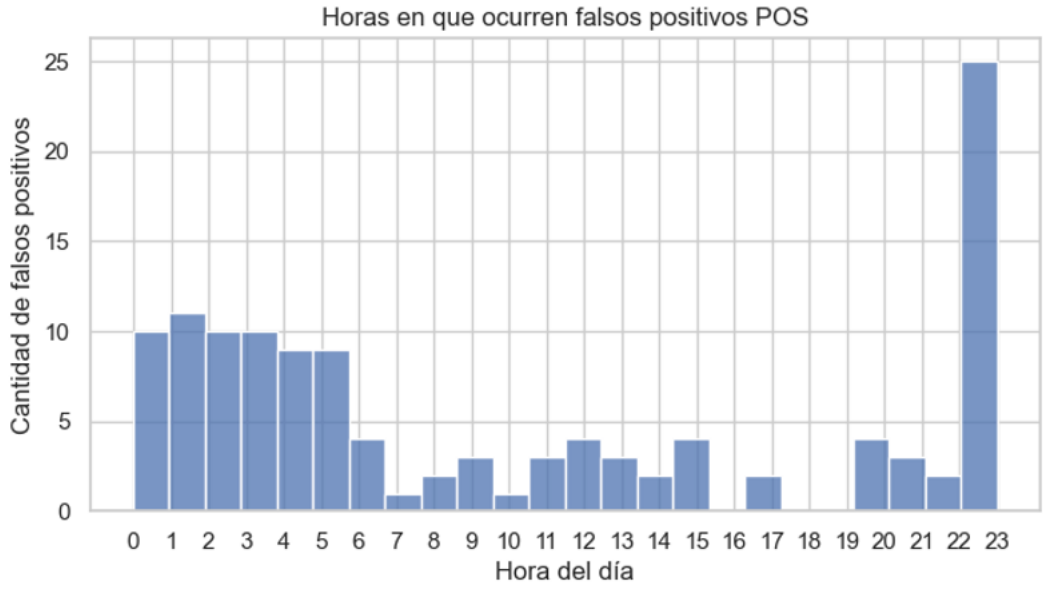
Análisis Exploratorio (EDA)

1. Distribución del monto por tipo de transacción:



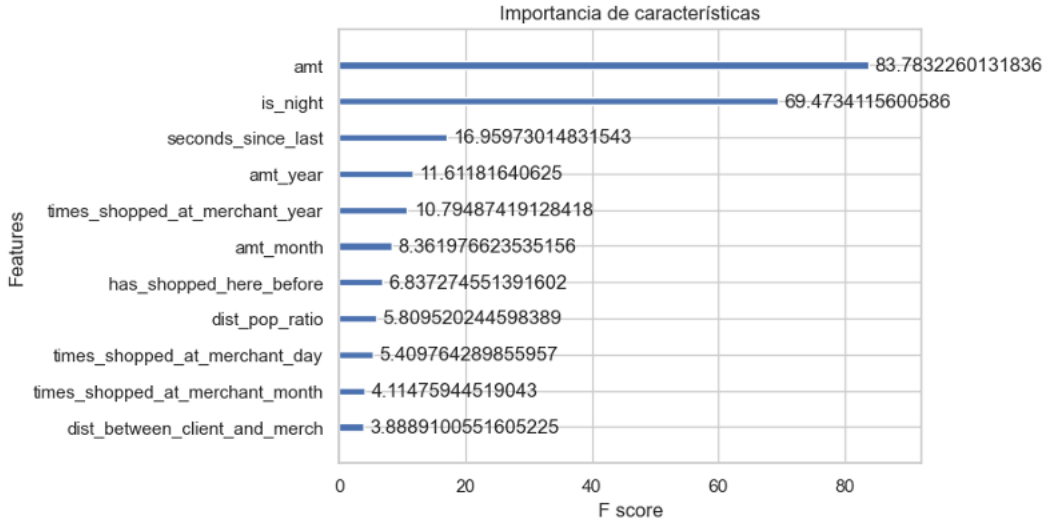
Las transacciones fraudulentas tienden a involucrar montos más altos y homogéneos. Esta observación fue clave para confirmar que el monto ('amt') debía tener alta importancia como feature en el modelo.

2. Horas en que ocurren más falsos positivos:



Los falsos positivos tienden a concentrarse en horarios poco comunes, como la madrugada o las 11 p.m., lo cual aporta información útil para ajustar las reglas o priorizar revisiones manuales.

3. Importancia de características del modelo LightGBM:



Esta gráfica confirma que variables como el monto ('amt'), el indicador nocturno ('is_night') y el tiempo desde la última compra ('seconds_since_last') son determinantes en la clasificación. Esto validó nuestras elecciones de features para el entrenamiento.

Ajuste de Hiperparámetros

Durante el entrenamiento de los modelos con LightGBM (LGBMClassifier), se realizó una optimización de hiperparámetros utilizando la librería Optuna. Esta técnica permite encontrar los valores más adecuados para mejorar el desempeño del modelo sin sobreajustar. A continuación, se describen los principales hiperparámetros optimizados:

Hiperparámetro	Descripción
learning_rate	Controla la velocidad de aprendizaje del modelo. Un valor más bajo hace el entrenamiento más lento pero generalmente más preciso. Se optimizó para encontrar el balance entre velocidad y estabilidad.
num_leaves	Determina la complejidad del árbol. A mayor cantidad de hojas, el modelo puede aprender patrones más complejos, pero también puede sobreajustar.
max_depth	Profundidad máxima de los árboles. Limita cuán complejo puede ser cada árbol individual. Ayuda a controlar el sobreajuste.
feature_fraction	Proporción de características utilizadas en cada iteración. Reduce el tiempo de entrenamiento y ayuda a evitar el sobreajuste.
bagging_fraction	Proporción de datos utilizados para entrenar cada árbol. También contribuye a la regularización.
bagging_freq	Frecuencia con la que se aplica el bagging.
n_estimators	Número total de árboles entrenados. Cuantos más árboles, mayor capacidad de aprendizaje, pero también mayor riesgo de sobreajuste si no se controla bien.
verbose	Nivel de salida durante el entrenamiento. Fue establecido como -1 para evitar salida innecesaria.

Modelado

- Modelo base: LightGBM (LGBMClassifier)
- Funciones de evaluación personalizadas:
    - fp_ratio: penaliza falsos positivos proporcionalmente
    - f1_fp_penalty: penaliza FP sobre F1
    - recall_fp_limit: maximiza recall con un límite de FP
- Optimización con Optuna sobre parámetros clave como learning_rate, num_leaves, n_estimators, etc.

Métrica	fp_ratio	f1_fp_penalty	recall_fp_limit
TP (fraudes detectados)	1632	1626	1622
FP (falsos positivos)	89	84	104
Ratio (TP+FP)/TP	1.05	1.05	1.06
Recall (class 1)	0.85	0.84	0.84
Precision (class 1)	0.95	0.95	0.94
ROC AUC	0.99907	0.99896	0.99910

Conclusiones

• El modelo 'fp_ratio' ofrece el mejor equilibrio entre detección de fraudes y baja cantidad de falsas alarmas.

• 'f1_fp_penalty' es ideal si se quiere ser más conservador con los FP.

• 'recall_fp_limit' ofrece el mayor ROC AUC, aunque con más falsos positivos.

Recomendación final

Usar 'fp_ratio' como métrica principal en producción, permitiendo adaptar la lógica empresarial a un modelo seguro, preciso y realista.