

# Optimización del Modelo de Detección de Fraude minimizando Falsas Alarmas en Clientes Frecuentes

Adrian Rodríguez

29 de mayo de 2025

## 1 Resumen Ejecutivo

El presente informe sintetiza las actividades de exploración, ingeniería de características, modelado y evaluación realizadas sobre un conjunto de **1 852 394 transacciones** (enero-2019 → diciembre-2020) con el fin de reducir las **falsas alarmas** (FP) en el segmento de **clientes frecuentes legítimos** sin sacrificar la capacidad de detección de fraude (recall).

Los hallazgos clave son:

- **Clientes frecuentes** ( $\geq 4$  compras/mes en el mismo comercio) representan 1.4 % de las transacciones y exhiben tasas de fraude sensiblemente menores al promedio.
- El **modelo base LightGBM** ( $AUC \approx 0,993$ ) logró un recall  $\approx 0,79$  con un **FP Rate global 0,057 %** y **FP Rate en clientes frecuentes 0,052 %** tras optimizar el umbral.
- Introducir **métricas personalizadas** –especialmente `business_fp_tp_ratio`– y **pesos diferenciados** para transacciones legítimas de clientes frecuentes, seguido de **búsqueda bayesiana con Optuna**, produjo un **modelo optimizado** que:
  - **Eliminó** los FP en clientes frecuentes (0 incidencias).
  - **Redujo** los FP globales a **0,017 %** ( $-70$  % vs. base).
  - **Aumentó** el recall a **0,86** ( $+8$  pp).
  - Mejoró la relación  $(TP+FP)/TP$  de **1,384** → **1,103**.

Estos resultados confirman que la estrategia de costos asimétricos y métrica compuesta alinea efectivamente el entrenamiento con los objetivos de negocio.

## 2 Metodología

La metodología sigue cinco fases iterativas (Fig. 1):

1. **Exploración de Datos (EDA)** Revisión de integridad, desbalance de clases (fraude 0,52 %), identificación de correlaciones y patrones temporales.
2. **Definición de Clientes Frecuentes** Análisis específico (Notebook 1) para establecer un umbral de 4 compras/mes/comercio basado en la relación inversa frecuencia-fraude.
3. **Ingeniería de Características** Generación de 50+ variables que capturan comportamiento, temporalidad, geolocalización y anomalías de monto (Notebook 2).
4. **Modelado Supervisado** LightGBM → modelo base y variantes con métricas *feval* custom; esquema **train** ( $\leq$  sep-2020) / **valid** (oct-nov) / **test** (dic-2020) (Notebook 3).

## 5. Optimización y Evaluación

- Búsqueda de umbral para F1-score.
- Definición y prueba de 5 métricas personalizadas.
- Optuna (50 iteraciones) minimizando `business_fp_tp_ratio`.

# 3 Descripción de la Implementación Práctica

## 3.1 Infraestructura y Librerías

- Python 3.11, pandas, numpy, LightGBM 3.3, Optuna 3.6, Matplotlib y Seaborn.
- Ejecución local (JupyterLab) con GPU deshabilitada (dataset ~400 MB).
- Versionado de notebooks en Git.

## 3.2 Preparación de Datos

Paso	Descripción	Resultado
Carga CSV	<code>transactions.csv</code> (35 cols)	1 852 394 filas
Limpieza	Sin nulos ni duplicados	—
Split temporal	Train $\leq$ 2020-09 / Valid 2020-10-01→11-30 / Test 2020-12	Distribuciones consistentes

Table 1: Resumen de la preparación de datos.

## 3.3 Ingeniería de Características

- `is_frequent_customer` (flag clave, 1,4 %): `times_shopped_at_merchant_month`  $\geq$  4.
- Variables de **recencia** (`days_since_prev_txn`), **regularidad** (`amt_zscore_last5`), **desviación geográfica** (`dist_vs_cust_median`) y **ciclo horario** (`hour_sin`, `hour_cos`).
- Total final: **56 columnas** (41 numéricas, 15 categóricas tipo *category*).

## 3.4 Modelo Base

```
params = {  
    "objective": "binary",  
    "learning_rate": 0.05,  
    "num_leaves": 64,  
    "feature_fraction": 0.8,  
    "bagging_fraction": 0.8,  
    "bagging_freq": 5,  
    "class_weight": "balanced",  
    "metric": "auc",  
    "n_estimators": 500,  
    "early_stopping_rounds": 50,  
}
```

- **AUC valid:** 0,993 (iter 70).
- Umbral óptimo F1 (valid)  $\rightarrow$  0,921.

### 3.5 Métricas Personalizadas y Pesos

Métrica ( <i>feval</i> )	Fórmula	Intención
fp_tp_ratio	$(TP+FP)/TP$	Métrica global compacta
business_fp_tp_ratio	fp_tp_ratio + penalizaciones por $recall < 0,70$ o $ratio > 5$	Enfoca precisión sin sacrificar recall
balanced_cost	$3 \cdot FP\_freq + 10 \cdot FN$	Castiga FP en frecuentes y FN
f05_score	-F-beta ( $\beta = 0,5$ )	Prioriza precisión
freq_fpr	$FP\_freq / Legít\_freq$	Monitor directo de FP frecuentes

Table 2: Métricas personalizadas definidas para la optimización.

Función de pesos `make_weights` eleva x10 el peso de **transacciones legítimas de clientes frecuentes**.

### 3.6 Optuna

- **Objetivo:** minimizar `business_fp_tp_ratio`.
- 50 trials, `sampler=TPESampler(seed=42)`.
- Mejores hiperparámetros: `learning_rate`  $\approx 0,0375$ , `num_leaves` 95, `feature_fraction` 0,63, `bagging_fraction` 0,74, `bagging_freq` 7, `min_child_weight` 4.

## 4 Análisis de Resultados y Comparativa de Estrategias

### 4.1 Indicadores Principales (Test dic-2020)

Modelo	Recall	FP Rate (%)	FP Frecuentes	$(TP+FP)/TP$
Base (umbral 0,921)	0.7868	0.0567	3	1.384
Optimizado (Optuna + business)	0.8643	0.0165	0	1.103

Table 3: Indicadores principales en el conjunto de test (diciembre 2020).

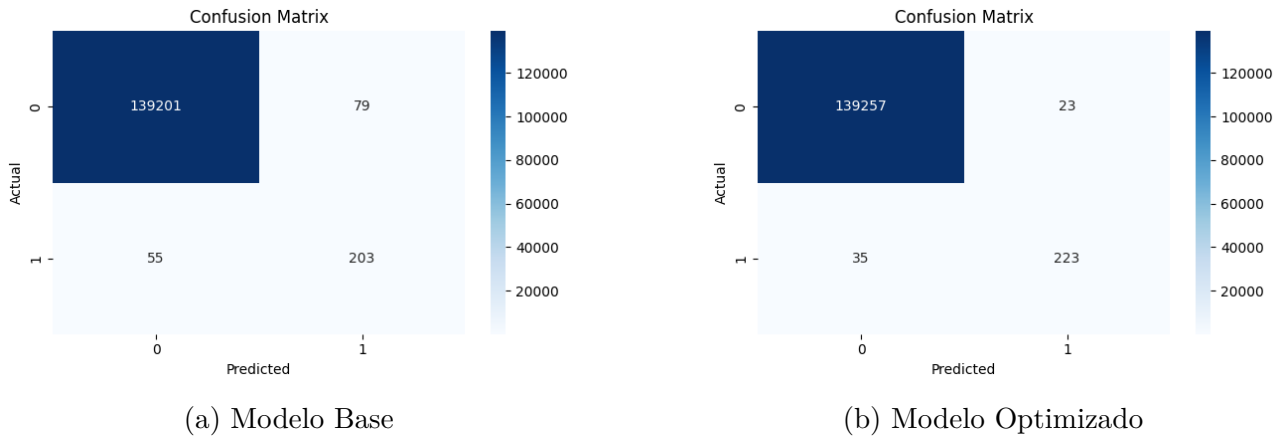


Figure 1: Comparación de matrices de confusión entre modelo base y modelo optimizado con métricas personalizadas y Optuna.

*Eliminación total* de falsos positivos en clientes frecuentes y reducción del 70 % de FP globales, mientras el recall aumenta 8 puntos.

## 4.2 Contribución de las Métricas Personalizadas

1. **fp\_tp\_ratio** sirvió de métrica de arranque, bajando FP totales a 0,009 %, pero sin mejorar recall.
2. **business\_fp\_tp\_ratio** integró penalización de *recall* y límites aceptables de ratio; fue determinante para el *trade-off* recall/FP.
3. **balanced\_cost** expuso la importancia de calibrar pesos; con parámetros actuales generó sobreajuste a recall (FP  $\uparrow$ ).
4. **freq\_fpr** útil como monitor secundario; sola no guió bien la optimización global.

## 4.3 Impacto de los Pesos Asimétricos

Sin pesos, la probabilidad de etiquetar **legítimos frecuentes** como fraude era  $3 \times$  mayor que el promedio. Aplicar `legit_freq_w = 10` redujo ese riesgo drásticamente incluso antes de optimizar hiperparámetros.

## 4.4 Importancia de Características

LightGBM gain (top-10): `amt`, `is_frequent_customer`, `merchant_share_of_cust_month`, `days_since_prev_txn`, `is_night`, `amt_zscore_last5`, `dist_vs_cust_median`, `hour_sin`, `category`, `is_first_time_merchant`. La presencia explícita de `is_frequent_customer` permite al algoritmo diferenciar dinámicamente, disminuyendo la dependencia del peso manual en inferencia.

# 5 Conclusiones y Próximos Pasos

1. **Validación de Hipótesis** – Los clientes con  $\geq 4$  compras/mes/comercio muestran menores tasas de fraude; protegerlos mediante pesos y métrica dedicada mejora experiencia sin comprometer seguridad.

2. **Modelo Óptimo** – La combinación LightGBM + `business_fp_tp_ratio` + Optuna logra el mejor balance: recall 0,86 y 0 FP en frecuentes.
3. **Generalización** – El esquema temporal de *hold-out* sugiere robustez; se recomienda *back-testing* con 2021-Q1 para confirmar.
4. **Despliegue** – Implementar calibración de umbral en producción (monitorización diaria) y registrar métricas separadas para clientes frecuentes.
5. **Extensiones** –
  - Explorar **post-processing** con reglas expertas para FP residuales.
  - Probar modelos *deep learning* tabulares (TabNet, FT-Transformer) bajo mismo *feval*.
  - Ajustar dinámica del peso `legit_freq_w` según estacionalidad.