Universidad del Valle de Guatemala
Security Data Science

Rebeca Smith
20373

Métricas custom para reducción de falsos positivos en clasificación binaria – fraude

Resumen

Este trabajo implementa un sistema de detección de fraude usando LightGBM. Se inició con un análisis del dataset proporcionado con el cual, con ingeniería de características, se incorporaron variables relacionadas con el comportamiento del cliente, geolocalización y características de los comercios. Para evaluar el modelo, se implementaron métricas personalizadas que permitieron balancear la detección de fraudes. Se usó un enfoque temporal, entrenando el modelo y evaluando su rendimiento con el último trimestre. Los resultados muestran una alta capacidad de detección (recall ≈ 0.90-0.99) con una ratio de falsos positivos controlada, lo que permite encontrar un equilibrio adecuado entre seguridad y eficiencia operativa.

Metodología

La detección de fraudes financieros es un problema, donde las transacciones fraudulentas representan un porcentaje muy pequeño del total. Ante este reto, los modelos tradicionales tienden a sesgarse hacia la clase mayoritaria. Por ello, se requieren técnicas específicas tanto para la ingeniería de variables como para la evaluación del rendimiento.

LightGBM (Light Gradient Boosting Machine) es un algoritmo de aprendizaje automático basado en árboles de decisión, optimizado para velocidad y rendimiento en tareas con grandes volúmenes de datos y alto desbalance. Se adapta bien a este tipo de problemas al permitir el uso de pesos de clase y métricas personalizadas (`feval`), como la ratio de falsos positivos, que aquí se define como:

$$\frac{Ratio\ FP}{TP} = \frac{TP + FP}{FP}$$

Además del uso de métricas estándar como AUC-ROC y F1, se implementaron funciones de evaluación diseñadas para mantener altos niveles de detección de fraude mientras se reduce el número de alertas falsas.

Universidad del Valle de Guatemala
Security Data Science

Rebeca Smith
20373

Descripción de la implementación práctica

El proyecto se desarrolló con el objetivo de mejorar la detección de fraudes en transacciones financieras, priorizando la cobertura de fraudes en comercios de alto riesgo y el recall en transacciones de alto monto. Para lograr esto, se siguieron las siguientes etapas:

Análisis exploratorio de datos (EDA)

Se cargó el dataset original de transacciones y se realizó un análisis exploratorio para entender la distribución de variables clave. Se identificó el desbalance de clases y se exploraron patrones según el tipo de comercio, monto y ubicación.

Ingeniería de variables

Se crearon nuevas variables para la información disponible al modelo. Algunas de las más relevantes fueron:

- *comercio_riesgoso*: variable binaria que identifica categorías con alta tasa de fraude como misc_net, shopping_net, shopping_pos.
- *transaccion_alto_riesgo*: marca transacciones con monto mayor al percentil 90 y que además ocurrieron en comercios riesgosos.
- Variables acumuladas y de comportamiento como: veces que el cliente ha comprado en el mismo comercio (times_shopped_at_merchant), gasto mensual por categoría (amt_month_shopping_net_spend), y distancia entre cliente y comercio (dist_between_client_and_merch).

Entrenamiento del modelo base

Se entrenó un modelo usando LightGBM con:

- class_weight='balanced' para mitigar el desbalance.
- Parámetros como num_leaves, learning_rate y n_estimators ajustados.
- Evaluación estándar con métricas como AUC, F1 y recall.

```
model = LGBMClassifier(
    random_state=42,
    class_weight='balanced',
```

Rebeca Smith
20373

```
    n_estimators=500,
    learning_rate=0.05,
    max_depth=6,
    num_leaves=31
)
```

Definición y uso de métricas personalizadas

Para alinear la evaluación a los objetivos del negocio, se implementó una métrica feval personalizada que mide la ratio de falsos positivos respecto a verdaderos positivos, buscando mantener una detección de fraude >90% con menor costo operativo.

```
def ratio_falsos_positivos(y_true, preds):
    # Umbral de decisión
    y_pred = (preds > 0.5).astype(int)

    tp = np.sum((y_pred == 1) & (y_true == 1))
    fp = np.sum((y_pred == 1) & (y_true == 0))

    if tp == 0:
        ratio = float('inf')  # Penaliza si no detecta ningún fraude
    else:
        ratio = (tp + fp) / tp

    return 'ratio_fp_tp', ratio, False  # False porque menor es mejor
```

Se probaron diferentes umbrales de decisión (0.3, 0.4, 0.5, 06) para observar cómo varía el recall y la precisión, especialmente sobre las transacciones de alto riesgo.

```
for threshold in [0.3, 0.4, 0.5, 0.6]:
    y_pred_thr = (y_proba > threshold).astype(int)
    prec = precision_score(y_test, y_pred_thr)
    rec = recall_score(y_test, y_pred_thr)
    f1 = f1_score(y_test, y_pred_thr)
    print(f"Umbral: {threshold:.2f} → Precisión: {prec:.3f}, Recall:
{rec:.3f}, F1: {f1:.3f}")
```

Se evaluaron varios modelos y métricas (binary_logloss, auc, ratio_fp_tp) comparando su rendimiento en el último trimestre. El análisis identificó el modelo con mejor balance entre detección y falsos positivos.

Rebeca Smith
20373

Conclusiones

El modelo basado en LightGBM, con ajuste de hiperparámetros y la configuración class_weight='balanced', demostró un alto desempeño en la detección de fraude, manteniendo una cobertura superior al 90 % en transacciones de alto riesgo. El análisis de distintos umbrales de decisión evidenció que es posible ajustar el balance entre recall y precisión, lo cual resulta clave en sistemas donde el costo de una falsa alerta puede ser significativo. Además, mediante el uso de callbacks como *early_stopping* y *log_evaluation*, se logró evitar el sobreentrenamiento y facilitar el monitoreo del rendimiento del modelo durante el proceso de entrenamiento.

Análisis de los resultados de la evaluación, con énfasis en el comparativo de estrategias

Estrategia / Métrica	Detección (Recall)	Precisión	Ratio FP/TP
Modelo con binary_logloss	0.981	0.124	14.6
Modelo con AUC	0.983	0.137	13.2
Modelo con ratio_fp_tp	0.995	0.153	12.97

El modelo optimizado con la métrica `ratio_fp_tp` fue el que logró el mejor balance general, al mantener un recall del 99.5% con una mejor relación entre falsos positivos y verdaderos positivos.

Aunque todas las estrategias lograron altos niveles de detección, el uso de la métrica personalizada permitió reducir la cantidad de falsos positivos sin sacrificar significativamente la cobertura del fraude.