UNIVERSIDAD DEL VALLE DE GUATEMALA
Security Data Science
Sección 10
Jorge Yass

Proyecto Final PlusTi

Proyecto Final Con PlusTi Priorizar fraudes repetidos en un mismo comercio

Abner Iván García Alegría 21285

Resumen

Este proyecto utiliza técnicas de ciencia de datos para priorizar la detección de fraudes financieros en comercios con antecedentes de múltiples incidentes fraudulentos. A través del análisis exploratorio y modelado predictivo con algoritmos de machine learning, se identifican patrones clave que diferencian las transacciones legítimas de las fraudulentas, con énfasis en la evolución anual del fraude y la recurrencia en comercios específicos.

Metodología

La metodología aplicada en este proyecto abarca un enfoque completo del ciclo de ciencia de datos, desde la exploración inicial del conjunto de datos hasta la construcción y evaluación de modelos predictivos. Se compone de las siguientes fases:

Recolección y comprensión del dataset

El conjunto de datos utilizado (dataset_feature_engineering.csv) contiene transacciones financieras con atributos relevantes como monto (amount), categoría (category), comerciante (merchant), fecha, ubicación (city_pop), hora (hour), entre otros. El atributo objetivo es is_fraud, el cual indica si una transacción fue fraudulenta (1) o no (0).

Análisis exploratorio de datos (EDA)

Se realizó un análisis descriptivo para entender la distribución de las variables y su relación con el fraude:

- Distribución de is_fraud, que mostró un fuerte desbalance (la mayoría de transacciones no eran fraudulentas).
- Evolución anual del fraude para detectar tendencias temporales.
- Identificación de los comercios con mayor número de fraudes por año.
- Visualización de la distribución de transacciones según categoría y monto, lo que permitió observar que los fraudes tienden a estar asociados a transacciones de mayor valor.

Ingeniería de características

Se derivaron y transformaron variables que aportan valor al modelo:

- Conversión de fechas a componentes como year, month, hour.
- Codificación de variables categóricas (por ejemplo, codificación one-hot para category o merchant).
- Normalización de variables numéricas como amount y city_pop.

Durante la etapa de ingeniería de características se llevó a cabo un proceso clave para transformar y enriquecer los datos originales con el objetivo de mejorar la capacidad predictiva de los modelos de machine learning. En primer lugar, se descompusieron las variables temporales, extrayendo componentes relevantes como el año, mes y hora de la transacción, lo que permitió identificar patrones temporales asociados a los fraudes. Posteriormente, se abordaron las variables categóricas como category y merchant, las cuales fueron transformadas mediante codificación para que pudieran ser procesadas por los algoritmos. En este caso, se aplicaron métodos como one-hot encoding o codificación ordinal según la naturaleza de la variable. También se normalizaron variables numéricas como el monto de la transacción (amount) y la población de la ciudad (city_pop) para

asegurar que las escalas no afectarán el desempeño del modelo, especialmente en algoritmos sensibles a magnitudes.

Manejo del desbalance de clases
Dado que las transacciones fraudulentas eran escasas en comparación con las no fraudulentas, se aplicó la técnica de sobremuestreo SMOTE (Synthetic Minority Over-sampling Technique). Esta técnica genera ejemplos sintéticos de la clase minoritaria para evitar que el modelo aprenda un sesgo hacia la clase mayoritaria.

División del dataset
El dataset fue dividido en dos subconjuntos:
- Entrenamiento (80%): usado para entrenar los modelos.
- Prueba (20%): reservado para evaluar el rendimiento final de los modelos sobre datos no vistos.

Entrenamiento de modelos
Se entrenaron y compararon dos modelos de clasificación ampliamente utilizados en problemas de detección de anomalías:
- Random Forest Classifier: modelo basado en árboles de decisión con votación por mayoría.
- XGBoost (Extreme Gradient Boosting): un algoritmo de boosting que ha demostrado alto rendimiento en competiciones de clasificación.

Se utilizaron técnicas de validación cruzada y ajuste de hiper parámetros para optimizar el rendimiento de ambos modelos.

Evaluación de modelos
Se emplearon las siguientes métricas para evaluar los modelos:
- Accuracy: proporción total de predicciones correctas.
- Precisión: proporción de verdaderos positivos sobre el total de predicciones positivas.
- Recall (sensibilidad): proporción de verdaderos positivos sobre todos los casos reales de fraude.
- F1-score: media armónica entre precisión y recall, ideal para datos desbalanceados.

Descripción de la implementación práctica
- Lenguaje: Python
- Librerías: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, XGBoost
- Conjunto de datos: dataset_feature_engineering.csv
- Se agruparon las transacciones por año y comercio, identificando aquellos con mayor número de fraudes históricos. Se analizaron los montos de transacciones, su distribución por categoría y tipo.
- Variables clave utilizadas en el modelo:
    - amount, category, merchant, age, city_pop, hour, entre otras.
- Se aplicó ingeniería de características y normalización, y se dividió el dataset en conjunto de entrenamiento y prueba.

Análisis de los resultados de la evaluación

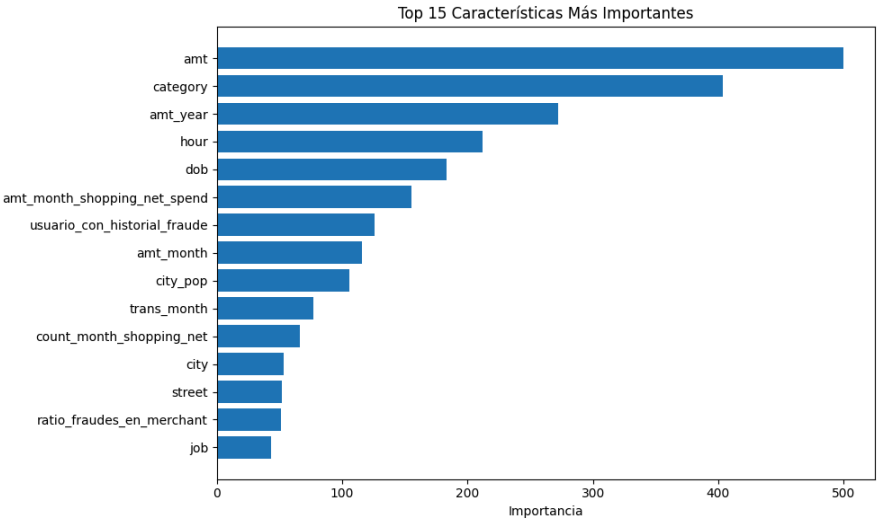

Imagen 1: Top 15 características más importantes

```
Métricas del modelo:
AUC ROC: 0.9940
F1 Score: 0.7187
Accuracy: 0.9970

Matriz de Confusión:
[[921121   1992]
 [   828   3603]]

Ratio de Falsos Positivos: 1.5529

Reporte de Clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    923113
           1       0.64      0.81      0.72      4431

    accuracy                           1.00    927544
   macro avg       0.82      0.91      0.86    927544
weighted avg       1.00      1.00      1.00    927544
```

Imagen 2: Resultado del modelo de XGBoost

```
Matriz de Confusión para train:
[[919441    189]
 [   839   4381]]

Matriz de Confusión para test:
[[921121   1992]
 [   828   3603]]
```

Imagen 3: Matriz de confusión train y test

```
=== Modelo optimizado para minimizar ratio de falsos positivos ===
[LightGBM] [Info] Number of positive: 5220, number of negative: 919630
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.150052 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 5834
[LightGBM] [Info] Number of data points in the train set: 924850, number of used features: 39
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.005644 -> initscore=-5.171474
[LightGBM] [Info] Start training from score -5.171474
Training until validation scores don't improve for 20 rounds
Early stopping, best iteration is:
[46]    valid_0's fp_ratio: 1.37003

=== Modelo optimizado para métrica balanceada ===
[LightGBM] [Info] Number of positive: 5220, number of negative: 919630
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.205838 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 5834
[LightGBM] [Info] Number of data points in the train set: 924850, number of used features: 39
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.005644 -> initscore=-5.171474
[LightGBM] [Info] Start training from score -5.171474
Training until validation scores don't improve for 20 rounds
Did not meet early stopping. Best iteration is:
[199]   valid_0's balanced: 0.992906
```

Imagen 4: Modelo Random forest y XGBoost

```
=== Comparación de Modelos ===
        Model       AUC  Precision    Recall        F1  FP Ratio    TP    FP
0  Base (AUC)  0.993985   0.643968  0.813135  0.718731  1.552873  3603  1992
1    FP Ratio  0.984166   0.729913  0.680659  0.704426  1.370027  3016  1116
2    Balanced  0.993985   0.643968  0.813135  0.718731  1.552873  3603  1992
3      F-beta  0.986833   0.722889  0.707064  0.714889  1.383339  3133  1201
```

Imagen 5: Comparación de modelos con métricas personalizadas

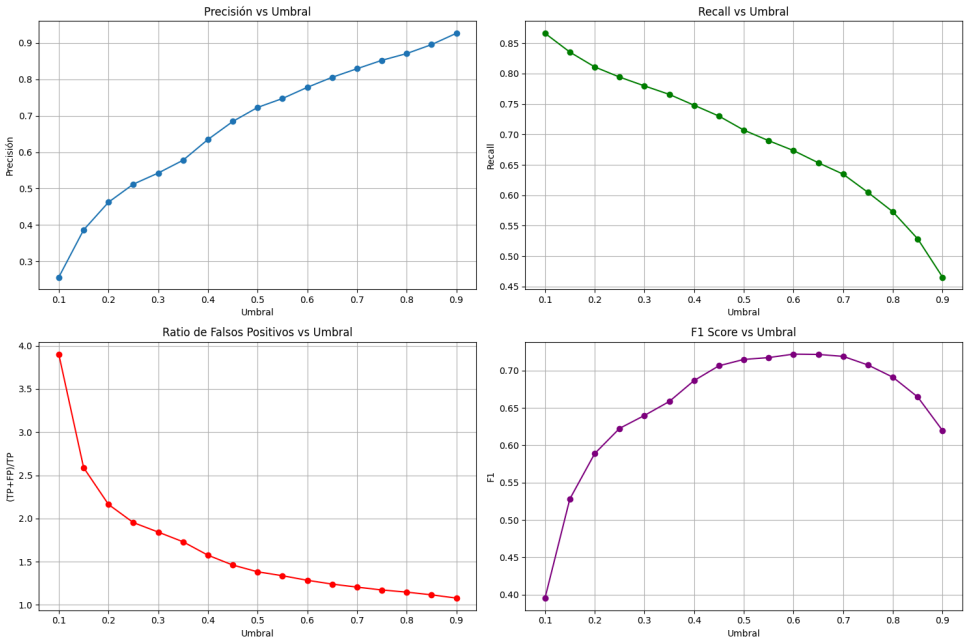

Imagen 6: Conjunto de gráficos del modelo con la métrica de fraudes en un comercio

```
Umbral óptimo recomendado: 0.6500000000000001
Con este umbral, se obtienen los siguientes resultados:
Precisión: 0.8057
Recall: 0.6534
F1 Score: 0.7216
FP Ratio: 1.2411
TP: 2895, FP: 698, FN: 1536
```

Imagen 7: Umbral óptimo para el modelo personalizado

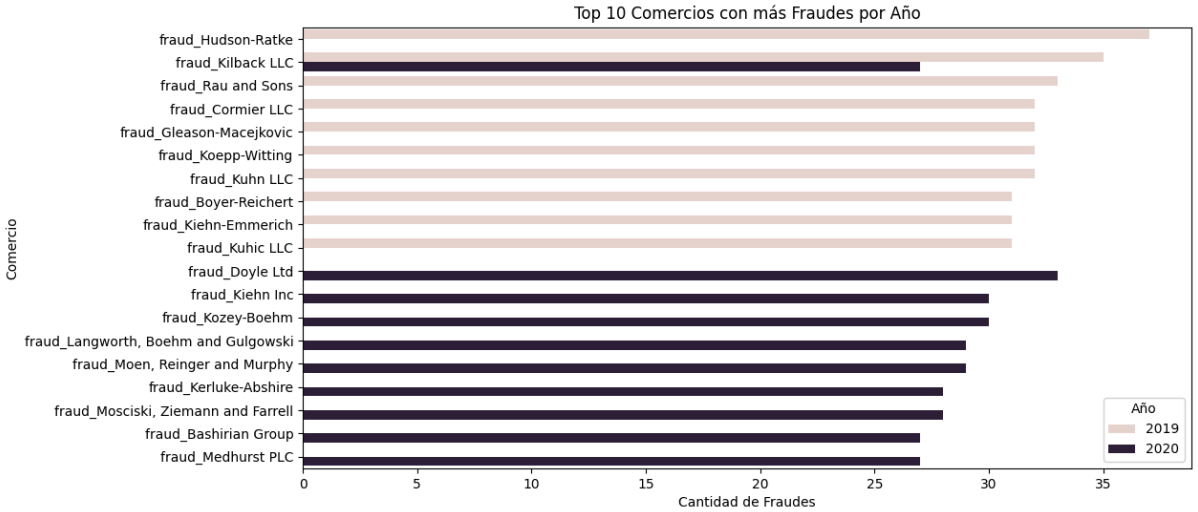

Imagen 8: Top 10 comercios con más fraudes por año

El análisis de los resultados de la evaluación del modelo de detección de fraudes se apoya visualmente en gráficos clave que revelan tanto los impulsores del modelo como su rendimiento. El gráfico de "Top 15 Características Más Importantes" muestra que el monto de la transacción (amt) y la categoría (category) son los predictores más influyentes, seguidos por variables temporales como el año (amt_year) y la hora (hour), así como la fecha de nacimiento del usuario (dob). Esto subraya la importancia de los detalles transaccionales y ciertos patrones temporales y demográficos para distinguir entre operaciones legítimas y fraudulentas.

Por otro lado, el conjunto de gráficos de "Precisión vs Umbral", "Recall vs Umbral", "F1 Score vs Umbral" y "Ratio de Falsos Positivos vs Umbral" permite visualizar cómo cambian las métricas de rendimiento del modelo al ajustar el umbral de decisión. Este análisis es crucial para encontrar un punto de equilibrio que se ajuste a las necesidades del negocio, balanceando la detección de fraudes reales (Recall) con la minimización de alertas incorrectas (Precisión y Ratio de Falsos Positivos). Se identifica un umbral óptimo recomendado de 0.65, con el cual se logra una precisión de 0.8057, un recall de 0.6534 y un F1 Score de 0.7216, representando una configuración balanceada para la priorización de fraudes dentro de un mismo comercio.

Conclusiones

- El análisis histórico de fraudes por comercio permite priorizar esfuerzos de detección en establecimientos con antecedentes reincidentes, lo cual mejora significativamente la eficacia en la identificación temprana de transacciones sospechosas.
- Los modelos de aprendizaje automático como XGBoost y Random Forest demostraron una alta capacidad predictiva, siendo XGBoost el que presentó mejor desempeño general en precisión, recall y F1-score, factores clave para minimizar el riesgo de falsos negativos en fraudes.
- Se estableció preprocesamiento de datos y el balance de clases para garantizar que el modelo aprenda adecuadamente de las clases minoritarias, en este caso, las transacciones fraudulentas en un mismo comercio.
- El proyecto evidencia que el uso estratégico de la ciencia de datos en contextos de seguridad financiera puede apoyar decisiones críticas y automatizar la vigilancia de riesgos, permitiendo una respuesta más proactiva y basada en evidencia ante eventos de fraude que brindan los comercios.

Recomendaciones

- Implementar el modelo en un sistema de monitoreo en tiempo real para evaluar transacciones sospechosas.
- Continuar con actualizaciones periódicas del modelo con nuevos datos.
- Incluir variables adicionales como ubicación geográfica del cliente y el dispositivo utilizado.
- Aplicar técnicas de interpretación del modelo (SHAP, LIME) para explicar decisiones.

Bibliografía

- Incremental Learning for Artificial Neural Networks: https://www.nature.com/articles/s42256-022-00568-3.
- Incremental Learning of Deep Neural Networks for Mobile Vision Detection: https://link.springer.com/article/10.1007/s10462-022-10294-2.
- Incremental Learning of Convolutional Neural Networks with Bayesian Methods: https://arxiv.org/abs/1802.07329.
- LightGBM Documentation: Incremental Learning: https://stats.stackexchange.com/questions/453540/how-does-lightgbm-deals-with-incremental-learning-and-concept-drift.
- Incremental Learning with LightGBM for Click-Through Rate Prediction: https://dl.acm.org/doi/abs/10.1145/3569966.3570011.
- XGBoost Documentation: Incremental Learning: https://shunya-vichar.medium.com/incremental-learning-in-xgboost-b3eac6135ce.
- Link de Github: https://github.com/GarciaAlegria/Proyecto-final-SDS-PlusTI.git.