UNIVERSIDAD DEL VALLE DE GUATEMALA

Departamento de Ciencias de la Computación

Security Data Science

Sección 10

Ing. Jorge Yass

Proyecto - Métricas custom para reducción de falsos positivos en clasificación binaria fraude con enfoque a comercios con historial riesgoso

Elías Alberto Alvarado Raxón – 21808

Guatemala, 02 de junio de 2025

Índice

Resumen

En el presente proyecto se tiene como objetivo el poder optimizar un modelo de LightGBM capaz de clasificar una transacción como legítima o fraudulenta, teniendo alto cuidado en optimizar la detección de transacciones fraudulentas en comercios previamente identificados como riesgosos. Para poder realizar esto se implementaron distintos enfoques utilizando distintas estrategias de ingeniería de características y evaluaciones personalizadas, todas orientadas a mejorar la sensibilidad del modelo ya que el dataset tiene un gran desbalance de clases.

Luego de un análisis exploratorio exhaustivo del conjunto de datos, se comprendió la distribución de las variables y su significado dentro del dataset. Con toda esta información se pudieron diseñar nuevas variables como la tasa histórica de fraude por comercio y categoría, lo cuál busca enriquecer la representación de un riesgo dentro de una transacción. Posteriormente, se construyeron tres funciones personalizadas que tienen como fin abordar distintas estrategias que puedan mejorar la detección del modelo.

Cada una de las funciones, con parámetros optimizados, lograron entregar distintos resultados. Unos mejores que otros, pero todas permitieron al modelo ajustar su función de pérdida y sus parámetros internos como los pesos, lo cuál hizo que se obtuvieron modelos buenos y malos en sentido al objetivo del proyecto.

Los resultados demostraron que la función basada en el *recall*, la cuál integraba análisis sobre comercios riesgosos, fue la más efectiva en cumplir con el objetivo del proyecto. Logrando un recall de 93.83% en comercios riesgosos, además que brindó una tasa de falsos positivos bastante prometedora. Asegurando que el enfoque tomado en dicha función personalizada fue la ideal para lograr obtener un modelo realmente bueno para la clasificación de transacciones fraudulentas en comercios con historial riesgoso.

Metodología

Análisis exploratorio de datos (EDA)

El conjunto de datos ya se sabía que contaba con un desbalance de clases bastante alto. Por lo que un EDA era esencial no solo para saber el comportamiento que tienen las variables, sino para saber qué pasos se podrían tomar para mejorar la entrada de datos del modelo.
Se realizaron análisis estadísticos y visuales de la distribución de las variables, en especial a la variable objetivo. Donde se pudo determinar que menos del 0.2% de las transacciones eran fraudulentas, confirmando el extremo cuidado que se debe de tener con la creación del modelo para evitar sesgos.

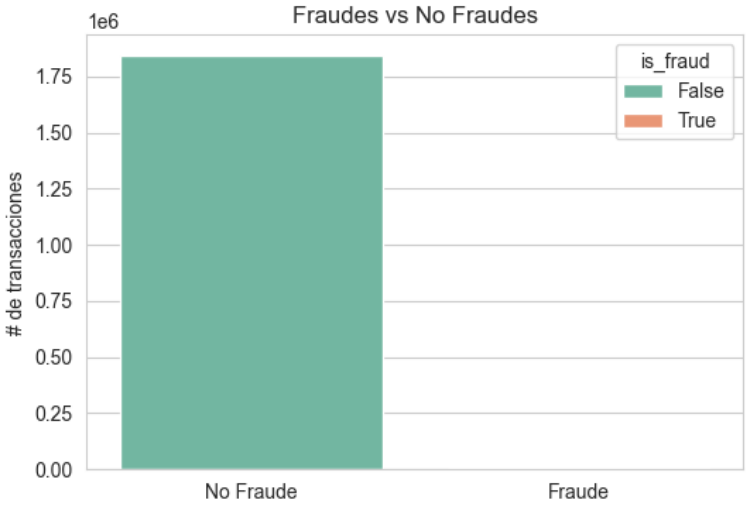

Gráfico 1. Comparación entre fraudes y no fraudes

Además, se observó que el dataset contaba con una alta cantidad de datos atípicos en una variable con suma importancia y que seguramente tendrá un alto significado para nuestro modelo. Los cuales son un riesgo ya que podrían sesgar nuestro modelo y provocar que no genere los pesos correctos. La variable es la cantidad de dinero relacionada a la transacción.

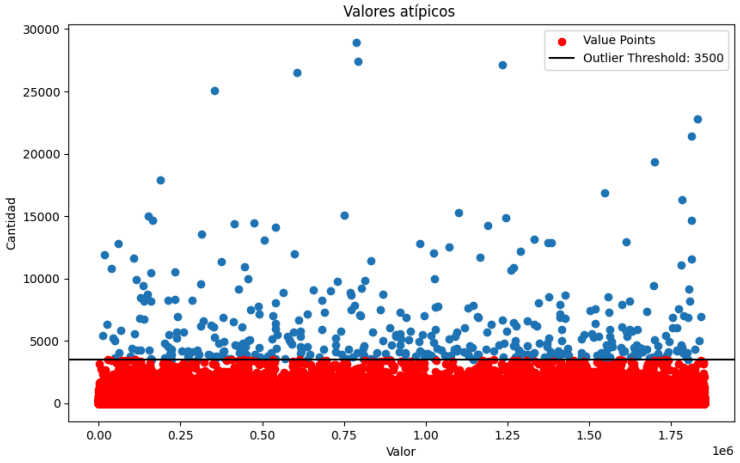

Gráfico 2. Comportamiento de la variable amount

Se puede observar cómo esta variable empieza a tener un comportamiento atípico desde un valor de 3500, aproximadamente. Donde la cantidad de registros con un valor mayor al mencionado, eran solamente de 386. Esa cantidad comparada con el tamaño del dataset no es realmente significativa. Por lo que se tomó la decisión de eliminar estos registros.

El resto del EDA se puede observar en el repositorio que se encontrará en el apartado de Anexos.

Ingeniería de características

Pensando en la función a optimizar, se incorporaron nuevas variables al dataset para ayudar al modelo tener acceso a variables importantes con el fin de mejorar su comportamiento y que dichas variables entreguen contexto histórico.

Tasa de transacciones por usuario por mes, día y hora

Esta variable histórica busca determinar cuando el usuario realiza más transacciones de las que haría normalmente. Sumamente importante para detectar si una transacción es fraudulenta.

Tiempo entre la transacción actual y la última realizada por el usuario

Cuando una transacción se realiza con un delta de tiempo muy corto con la última transacción, es muy probable que se trate de una transacción fraudulenta. Esta variable busca entregar una métrica clave para identificar transacciones automatizadas, o bien, transacciones realizadas muy rápidamente.

Es categoría o ubicación nueva para el usuario

Aunque es similar a la variable *fist_time_at_merchant*, esta variable busca entregar mayor información. Donde se determina si el usuario es nuevo en cierta categoría o ubicación. Para asegurar que el registro también contempla temas como herencia de categoría a comercio.

Tasa de fraudes detectados en comercio y categoría

Esta variable busca darle al modelo información si el comercio o la categoría tiene o no un historial riesgoso. Aportando información clave para que el modelo pueda luego determinar bajo un umbral, definido por mí (percentil 75), si el comercio es realmente riesgoso o no.

Separación de datos

Para la separación de entrenamiento y test se fraccionó el dataset por tiempo. El conjunto de datos para test se utilizó en diciembre de 2020. Debido a que dentro de un entorno real se busca que el modelo se comporte bien para iteraciones futuras sin fuga de información

Métricas personalizadas

Penalización según la tasa de falsos positivos

Esta función busca penalizar el ratio de falsos positivos que se está teniendo por cada vez que se encuentra un falso positivo. Buscando minimizar la cantidad de falsos positivos.

F1-Score penalizando por falsos positivos

Esta función busca evaluar el desempeño del modelo en términos de F1, pero penalizando los falsos positivos. Buscando minimizar la cantidad de falsos positivos.

Costo para minimizar falsos positivos y negativos

Esta función busca penalizar tanto los falsos positivos y negativos. Ya que en un entorno real, tanto los falsos positivos y negativos representan un costo en la vida real al ser clasificados así. Por lo que se busca minimizarlos utilizando pesos personalizados para cada uno de los casos.

Optimización de función objetivo

La función objetivo es la detección de fraude en comercios con historial riesgoso. Para alcanzar este objetivo se diseñaron tres funciones las cuales fueron implementadas como métricas de validación como funciones de pérdida dentro del modelo.

Recall comercios riesgosos

Esta función busca maximizar el recall únicamente para transacciones realizadas en comercios riesgosos.

F1-Score comercios riesgosos

Esta función penaliza el F1-Score con mayor peso cuando existe un falso positivo o negativo en un comercial riesgoso.

Costo ponderado del riesgo

Esta función busca calcular un costo ponderado, dándole un mayor costo cuando se trata de un comercio con historial riesgoso.

Evaluación de modelos

Una vez con los modelos entrenados, se debía de realizar una evaluación para determinar el modelo que entregó un mejor desempeño. Para ello se evaluaron métricas clave como:

- F1-Score
- Recall general
- Recall para comercios riesgosos

- Ratio de falsos positivos

Bajo este estudio fue fácil comparar de forma objetiva qué modelo maximiza la detección de transacciones fraudulentas en comercios con historial riesgoso.

Implementación

Preparación del dataset

La ingeniería de característica incluyó añadir varias variables nuevas. Las cuales se detallan a continuación:

- trans_per_day, trans_per_hour, trans_per_month: Para cada una de estas variables la clave fue primero agrupar el conjunto de datos por la variable *cc_num*, que representa a un usuario, y luego ya por la granularidad de tiempo que se desea. Para luego aplicar una transformación de tipo *count*.
- time_since_last_trans: Para poder lograr un valor representativo y coherente, para esta variable primero se ordenó el dataset por la variable de tiempo *unix_time* y luego se aplicó un agrupado por la variable *cc_num* para luego calcular la diferencia entre el registro actual con el anterior.
- first_time_at_category, first_time_at_city, first_time_at_state: Para esta variable se calculó si un *cc_num* ya tenía una transacción asociada a la categoría, ciudad o estado.
- rate_by_merchant, rate_by_category: Para esta variable se realizó un estudio asociando un comercio o categoría si anteriormente ya tenía o no transacciones fraudulentas registradas.

Luego de haber añadido las nuevas variables, se procedió a convertir las variables booleanas a valores de 0 o 1, para luego separar el dataset completo en dos. El dataset para test está compuesto de registros que datan del mes de diciembre de 2020, el resto (registros anteriores a dicha fecha) pertenecen al conjunto de datos de entrenamiento.

Utilizando la librería *RobustScaler* se escalaron variables monetarias y de comportamiento, y se convirtieron variables categóricas a tipo string para poder realizar un *LabelEncoder*, pero solo a variables que no tenían muchos datos distintos para evitar la creación de tantos nuevos valores. Finalmente, se eliminaron variables irrelevantes.

Modelo base

Para lograr un modelo base se realizó un *GridSearch* el cuál buscaba optimizar el F1-Score probando distintos valores para *n_estimators*, *learning_rate* y *num_leaves*. Con esto se determinó que los mejores parámetros eran:

- learning_rate: 0.05
- n_estimators: 1000
- num_leaves: 70

Donde, entregó los siguientes resultados:

- AUC-ROC (Entrenamiento): 1.0000
- AUC-ROC (Test): 0.9370

- F1-Score (Entrenamiento): 1.0000
- F1-Score (Test): 0.9159

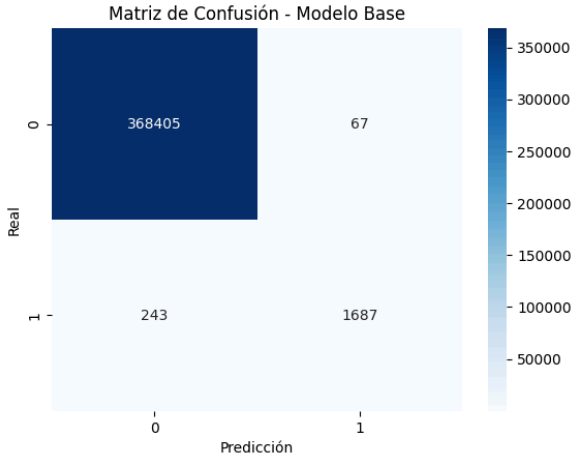

Gráfico 3. Matriz de confusión - Modelo base

Métricas personalizadas

Penalización según la tasa de falsos positivos

La implementación de esta métrica busca comparar la clase real con la clase predicha por el modelo, para luego calcular cuántos falsos y verdaderos positivos existen, con lo que se calcula el ratio de falsos positivos con el fin de retornar la métrica como un false, ya que busca minimizar su valor.

F1-Score penalizando por falsos positivos

Igual que la anterior. Se calculan los verdaderos y falsos positivos, y también los falsos negativos. Utilizando los valores anteriores se calcula la precisión y el recall del modelo con el fin de obtener el F1-Score. Al tener estos valores se calcula un factor de penalización basado en la proporción de falsos positivos, y se ajusta el F1-Score utilizando la penalización calculada. Finalmente se devuelve el valor como True, ya que busca maximizarse.

Costo para minimizar falsos positivos y negativos

Luego de calcular los verdaderos y falsos positivos, y los falsos negativos se le asigna un peso a cada uno de los falsos positivos o negativos con el fin de penalizarlos. El resultado se devuelve como false ya que busca minimizarse.

Optimización de función objetivo

Recall comercios riesgosos

Teniendo un enfoque de penalizar aun más el cálculo del impacto (penalización) por un factor 10 cuando se trata de un fraude en un comercio riesgoso. Aunque se sacrifiquen los falsos positivos.

F1-Score comercios riesgosos

Busca penalizar más los errores en predicciones positivas cuando se trata de una transacción legítima pero sobre un comercio riesgoso.

Costo ponderado del riesgo

Busca asignar un peso según el tipo de error en el que cae el modelo, buscando minimizar costos operativos en un entorno real tomando en cuenta errores como falsos positivos o negativos.

Análisis de resultados de evaluación

Analizando cada uno de los modelos bajo un enfoque de:

- F1-Score
- Recall general
- Recall para comercios riesgosos
- Ratio de falsos positivos

Se obtuvieron los siguientes resultados para cada uno de los modelos entrenados con distintas métricas y evaluaciones personalizadas.

Evaluación personalizada	**F1-Score**	**Recall**	**Recall para comercios riesgosos**	**Ratio de Falsos positivos**
Recall comercios riesgosos	0.6216	0.6240	0.9383	1.6149
F1-Score comercios riesgosos	0.6816	0.5891	0.8580	1.2368
Costo ponderado del riesgo	0.6047	0.4535	0.6605	1.1026

Tabla 1. Resultados de los modelos

Teniendo en cuenta la tabla anterior se puede observar cómo cada uno de los modelos presentó resultados buenos y malos en distintas áreas.
El recall comercios riesgosos mostró un excelente rendimiento para la detección de fraudes en comercios con historial fraudulento, muy apegado al objetivo del proyecto. Demostrando una gran habilidad para clasificar correctamente una transacción fraudulenta real, aunque conlleve el aumento de falsos positivos; sin embargo, en un entorno real este sería el mejor escenario, ya que preferiblemente se estudia una transacción legítima como fraudulenta a dejar pasar una transacción fraudulenta como legítima.
El F1 penalizado entregó el mejor F1, demostrando un gran equilibrio entre la detección de fraudes y el control de errores. Dejando un enfoque más balanceado, pero teniendo un menor rendimiento en la detección de fraudes cuando el comercio tiene un historial manchado.
El costo ponderado del riesgo, aunque presentó el menor ratio de falsos positivos, su recall cayó demasiado. Dejando el recall global como el enfocado a comercios riesgosos casi como lanzar una moneda.

Como bien se mencionó, cada estrategia mostró ventajas en distintas áreas y dependiendo el enfoque del proyecto una podría ser más ventajosa que otra. Dado que el objetivo del proyecto es maximizar la detección de fraudes en comercios con historial riesgoso, la mejor opción es la estrategia basada en recall en comercios riesgosos. Pues este modelo fue capaz

de alcanzar un recall del 93%, indicando que detecta con alta precisión transacciones fraudulentas en el escenario mencionado.

Conclusiones

- Se logró implementar un modelo capaz de detectar fraudes enfocado en comercio con historial riesgoso.
- La función de evaluación personalizada basada en recall de comercios riesgosos mostró el mejor desempeño, alcanzando un recall de 93%.
- Las distintas funciones de evaluación personalizada mostraron dominio en distintos criterios, aplicables en otros entornos donde el objetivo principal sea otro.
- La ingeniería de características y la optimización de hiperparámetros fueron clave para potenciar el rendimientos de los modelos.

Referencias

X. Zhao, Y. Liu and Q. Zhao. (2024) *Improved LightGBM for Extremely Imbalanced Data and Application to Credit Card Fraud Detection*. IEEE Access, vol. 12, pp. 159316-159335, 2024, doi: 10.1109/ACCESS.2024.3487212

N. Japkowicz and S. Stephen. (2017). *The class imbalance problem: A systematic study*. Intelligent Data Analysis, vol. 6, no. 5, pp. 429–449, 2002.

G. Ke et al. (2017) *LightGBM: A highly efficient gradient boosting decision tree*. Advances in Neural Information Processing Systems, vol. 30, 2017. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Anexos

Repositorio del proyecto

Enlace