Detección de fraude en cuentas bancarias (BAF): Un pipeline tabular con umbral operativo (FPR=5%) y análisis de fairness

Carlos Alberto Mentado Reyes, Fernanda Díaz Gutiérrez, José Eduardo Puentes Martínez, and Raymundo Iván Díaz Alejandre

Instituto Tecnológico y de Estudios Superiores de Monterrey

Abstract. El reto *Bank Account Fraud* de Kaggle consiste en predecir fraudes bancarios a partir de datos sintéticos y restricciones éticas. Presentando principalmente dificultades con desbalanceo de clases, variables categóricas, la presencia de atributos protegidos y diferentes disparidades presentes en varios conjuntos de datos. Para abordar el reto se realizó un análisis exploratorio de datos que reveló una gran cantidad de datos faltantes que se abordó con imputación de datos, baja cardinalidad de variables categóricas que propició la implementación de one-hot-encoding, necesidad de escalar los datos con StandardScaler. Con estas condiciones se entrenó un modelo LightGBM con una división temporal de los datos en la que los meses 0-5 fueron usados para entrenar, el mes 6 para validar y el mes 7 para pruebas, obteniendo así resultados con AUC de 0.89, un recall mayor a 50% manteniendo un FPR de 5%. Sin embargo se muestran indicios de imparcialidad en el modelo al comparar los valores de FPR de las personas mayores a las personas jóvenes.

Keywords: Fraude bancario, Tabular ML, LightGBM, Umbral operativo, FPR, Fairness

1 Introducción

Bank Account Fraud es un reto que se encuentra en Kaggle; consiste en una serie de conjuntos de datos sintéticos, creados con Inteligencia Artificial generativa, basándose en una base de datos real sobre fraude bancario. Es un reto que tiene como objetivo poner a prueba la capacidad de un modelo de clasificación de dar un buen resultado de *recall* (porcentaje de verdaderos positivos) mientras se mantiene un *FPR* menor o igual al 5% (FPR es una métrica que indica la cantidad de falsos positivos o "fallas" que tuvo el modelo al realizar predicciones). Las principales dificultades u obstáculos que se presentan en este reto son las siguientes:

- **Desbalanceo de clase:** En la variante *Base*, la tasa de fraude ronda el 0.8–1% (aprox. 99 no fraude por cada 1 fraude).
- **Variables categóricas:** Los conjuntos de datos contienen no solo características numéricas, sino que también hay características categóricas (texto)

y se tiene que tomar una decisión sobre cuál sería la mejor forma de tratar con ellas.

- **Atributos protegidos:** Esto se refiere a que hay algunas características que describen a una persona y que, por razones legales y/o éticas, no pueden ser completamente utilizadas para entrenar al modelo, pues podría crear un sesgo hacia algún sexo, grupo étnico o por edad.
- **Valores faltantes:** No todas las filas de los conjuntos de datos cuentan con información completa, hay algunos valores faltantes que se representan con un -1
- **División de los datos:** Usamos los meses 0–5 para entrenamiento, el mes 6 para validación y el mes 7 para pruebas, respetando la temporalidad.

Además, los conjuntos de datos también cuentan con diferentes disparidades que están más presentes en alguna de las variantes proveídas que en otras:

- **Disparidad de grupos:** Representada como $P[A = a] \neq \frac{1}{N}$ *La probabilidad de que un registro sea parte de un grupo no es igual a la probabilidad equitativa de todos los grupos.* Quiere decir que en el conjunto de datos hay mucha más representación de un grupo proveniente de los atributos protegidos que de los demás.
- **Disparidad de prevalencia:** Representada como $P[Y] \neq P[Y|A = a]$ *La probabilidad de que una muestra sea igual a una etiqueta de la variable objetivo no es igual a la probabilidad de que una muestra de un grupo sea igual a una etiqueta de la variable objetivo.* Esto quiere decir que algún grupo del conjunto de datos tiene muchas más etiquetas positivas a comparación de otros, esto puede causar un sesgo en los resultados del modelo.
- **Disparidad de separabilidad:** Representada como $P[X, Y] \neq P[X, Y|A = a]$ *La probabilidad de que la correlación de X y Y de el resultado de una etiqueta no es igual a que la correlación de X y Y de un grupo en especifico.* Es decir, la correlación entre las características de una muestra y la variable objetivo es mayor en algunos grupos que otros

Como fue mencionado, estas disparidades están presentes en las diferentes variantes de conjutos de datos proporcionados por el diseño del reto, con el objetivo de probar al máximo la capacidad de un modelo de clasificación de no crear un sesgo, la explicación de la repartición de las disparidades en el reto está descrita de la siguiente forma:

Table 1. Resumen de variantes del conjunto de datos Bank Account Fraud

Variante	**Descripción**
Base	Conjunto de datos que más se asemeja al conjunto original.
I	Mayor disparidad de grupos.
II	Mayor disparidad de prevalencia.
III	Mejor separabilidad para un grupo específico.
IV	Mayor disparidad de prevalencia en los datos de entrenamiento (meses 1 a 5).
V	Mejor separabilidad para un grupo en los datos de entrenamiento (meses 1 a 5).

Las columnas de los conjuntos de datos son similares para todas las variantes de estos, cuentan con una variable objetivo llamada 'fraud_bool', que puede tomar un valor de 0 o 1, en el que 0 significa que el registro no es un fraude y 1 para expresar que sí es un fraude. Las características númericas son: income, name_email_similarity, prev_address_months_count, current_address_months_count, customer_age, days_since_request, intended_balcon_amount, zip_count_4w, velocity_6h, velocity_24h, velocity_4w, bank_branch_count_8w, date_of_birth_distinct_emails_4w, credit_risk_score, email_is_free, phone_home_valid, phone_mobile_valid, bank_months_count, has_other_cards, proposed_credit_limit, foreign_request, session_length_in_minutes, keep_alive_session, device_distinct_emails_8w, device_fraud_count, month. Todas estas describen diferentes fenómenos de los conjuntos de datos que pueden ser descritos de forma numérica. Por otro lado, las características categóricas son: payment_type, employment_status, housing_status, source, device_os. Sin embargo, es importante mencionar que las variantes III y V de conjuntos de datos cuentan con dos variables numéricas adicionales, denominadas como X1 y X2, cuyo objetivo es agregar sesgo por disparidad de separabilidad en las variantes de conjuntos de datos ya mencionadas.

Como fue mencionado anteriormente, el objetivo de este reto es que al entrenar un modelo de clasificación automático con estas dificulatadas se logre obtener la calificación de *recall* más alta posible, manteniendo siempre una razón de fallo *FPR* menor o igual al 5% en este caso no se hace uso de la métrica de precisión, pues el desbalanceo de clases hace que sea irrelevante para los resultados del modelo, en su lugar, se hará uso de la curva ROC, que muestra la tasa de verdaderos positivos frente a los falsos positivos obtenidos en la prueba del modelo.

2 EDA

El análisis descriptivo de los datos es una herramienta que ayuda a entender como se comportan los datos de un conjunto de datos, a pesar de que en la descripción del reto ya se encuentra una pequeña descripción de lo que se encontrará en las muestras, es importante indagar aún más para tener una visión clara de como abordar el problema.

Gracias a este análisis se descubrió que la columna prev_address_months_count cuenta con el 70% de sus registros marcados como dato faltante (-1), por lo que se tendrá que tomar la decisión de si borrar la columna, borrar los registros faltantes o realizar una imputación para evitar perder la mayor cantidad de información posible. Este ánalisis también arroja la cardinalidad de las variables categóricas, la cuál no resulta ser muy alta, pues la más alta cardinalidad de las pocas variables categóricas existentes es de 9, esto hace que realizar one-hot-encoding no sea una opción imposible, pues no añadiría muchas más columnas de las ya existentes. También se descubrió que los rangos de los datos de las variables numéricas pueden ser muy grandes en ciertos puntos de los conjuntos de datos, por lo que resulta evidente que realizar un escalado de los datos ayudará a que el modelo los procese de la forma más eficiente posible.

Las gráficas de dispersión resultaron ser irrelevantes para el análisis realizado, pues la variable objetivo al ser un dato booleano, no permite ver un verdadero comportamiento, además, analizar el comportamiento del conjunto de datos columna por columna puede provocar que se pierda la imagen completa del problema, sobre todo considerando la existencia de los atributos protegidos que no permiten que todo el análisis se base en los datos personales de los clientes.

Una de las gráficas más relevantes es la relación temporal entre los meses del estudio y la cantidad de fraudes detectados.

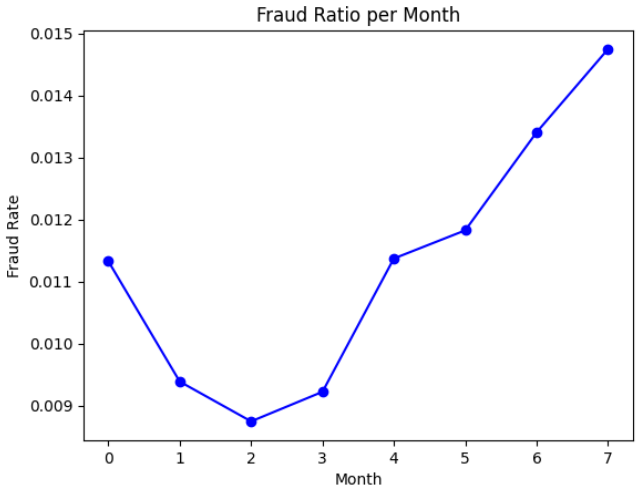

Fig. 1. Razón de fraudes detectados por mes

Esta gráfica hace evidente la razón por la que el reto se basa en los meses del estudio, se ve una elevación de casos positivos en los meses que están destinados

a ser validación y pruebas, de esta forma se pueden probar los sesgos del modelo que se obtuvieron durante el entrenamiento.

El análisis exploratorio de los datos también arroja la presencia de outliers en las muestras de los datos, sin embargo, se consideró que tratar con estos datos puede ser perjudicial, pues estos otorgan una imágen completa y real del comportamiento de los datos.

3 Preprocesamiento

3.1 Imputación y tratamiento de faltantes

Previo al ajuste de imputadores, los valores codificados como faltantes (-1) en columnas numéricas (prev_address_months_count, current_address_months_count, intended_balcon_amount, session_length_in_minutes, bank_months_count, device_distinct_emails_8w) se transformaron a NaN. La imputación numérica se realizó con mediana global y la categórica con la moda (most_frequent). Todo el preprocesamiento se *ajustó solo con train (0–5)* y se proyectó a validación (6) y prueba (7).

3.2 Escalamiento

Se aplicó StandardScaler a las variables numéricas para estabilizar el entrenamiento del clasificador basado en gradiente.

3.3 Codificación categórica

Se utilizó One-Hot Encoding con la opción de ignorar categorías desconocidas para las variables categóricas (payment_type, employment_status, housing_status, source, device_os). El ajuste del codificador se realizó solo con train y se proyectó sobre validación y prueba.

3.4 Balanceo y objetivo operativo

No se aplicó sobre-muestreo entre meses para respetar la secuencia temporal. Se empleó class_weight en LightGBM para compensar el desbalance y se fijó el umbral operativo en validación por cuantil de los negativos para asegurar FPR = 5%. Posteriormente, se evaluó en prueba usando ese mismo umbral y, adicionalmente, se reportó el recall en prueba a FPR exacto del 5% (método basado en la curva, sin cambiar el punto operativo).

3.5 Subtexto X

Variables como `source`, `device_os`, `housing_status` muestran alta cardinalidad; evitamos one-hot masivo y preferimos hacer pruebas con target/WOE encoding con **encajonado por fold temporal** para prevenir fuga.

4 Metodología de modelado

4.1 Modelos exploratorios y descartados

Red Neuronal (MLP) con SMOTE. Se entrenó una red neuronal densa sobre variables numéricas estandarizadas y variables categóricas por one-hot, aplicando SMOTE para balancear las clases. El entrenamiento se realizó con partición aleatoria y umbral fijo de 0.5. Los resultados reportados fueron altos, pero posteriormente se identificaron problemas metodológicos que invalidan esas métricas como evidencia comparativa.

Table 2. Resultados de la NN con SMOTE (partición aleatoria y umbral 0.5)

Métrica	Validación (promedio por batch)	Prueba
Accuracy	0.92–0.93	0.9277
Recall	0.90–0.95	0.9337

Motivos de descarte (riesgo de fuga y protocolo no comparable).

- Preprocesamiento ajustado con todo el conjunto antes de dividir (get dummies y StandardScaler sobre el total). Esto introduce fuga de información desde validación y prueba hacia entrenamiento.
- SMOTE aplicado antes de la división aleatoria, generando ejemplos sintéticos con información de todo el dataset y "contaminando" los splits.
- Partición aleatoria estratificada en lugar de respetar la división temporal (meses 0–5 entrenan, 6 valida, 7 prueba), por lo que no se evalúa el drift temporal del reto.
- Umbral fijo de 0.5 y métricas agregadas por batch, no se controló el punto operativo de FPR = 5% ni se reportaron métricas ROC/PR coherentes con el objetivo del reto (recall a FPR $< 5\%$).
- La combinación de los puntos anteriores explica los valores "demasiado buenos" de accuracy/recall: provienen de datos procesados con fuga y de un protocolo de evaluación distinto al del pipeline final.

En síntesis, aunque la NN con SMOTE arrojó números elevados, el procedimiento no es comparable con el pipeline validado (división temporal, preprocesamiento ajustado solo en train, ausencia de sobre-muestreo entre meses y fijación de umbral por cuantil para asegurar FPR = 5%). Por ello, se descartó como evidencia principal y no se incluyó en la comparación final, ya que al evaluar ajustando estos valores se obtuvieron métricas peores a las obtenidas con LigthGBM.

4.2 División temporal

Entrenamiento: meses 0–5; Validación: mes 6; Prueba: mes 7. No se mezclaron meses para preservar el drift temporal.

4.3 Modelos y regularización

Se empleó LightGBM como modelo principal, ya que permite manejar desbalance de clases mediante class_weight y ofrece buen rendimiento en variables mixtas (numéricas y categóricas). Se usaron hiperparámetros de regularización (num_leaves, min_child_samples, feature_fraction, bagging_fraction, lambda_l2) y early stopping.

4.4 Selección de umbral

El umbral operativo se fijó en validación mediante el cuantil de los puntajes negativos que asegura FPR = 5%. Ese mismo umbral se aplicó en test. También, reportamos el recall en test al 5% de FPR leído de la curva ROC (métrica de reporte), sin modificar el punto operativo.

4.5 Bias/Varianza y ajuste

La cercanía entre AUC de validación (0.89) y de prueba (0.89), junto con un AP mayor en prueba (0.211 vs. 0.176 en validación), indican baja varianza y ausencia de sobreajuste (fit adecuado).

5 Resultados

Para entender más el problema y cómo abordarlo, el equipo comenzó realizando un análisis exploratorio de los datos. En este, pudimos encontrar más detalles del conjunto de datos con el que se trabajará durante el reto.

5.1 Resultados en Validación

En la Tabla 3 se muestran los resultados del modelo en el conjunto de validación. El AUC de 0.89 indica buena capacidad de discriminación. Con un umbral operativo que asegura un 5% de FPR, el modelo alcanza un recall del 52%.

Table 3. Métricas en Validación

Métrica	Valor
AUC	0.890
AP (PR-AUC)	0.176
FPR (umbral operativo)	0.050
Recall (umbral operativo)	0.524

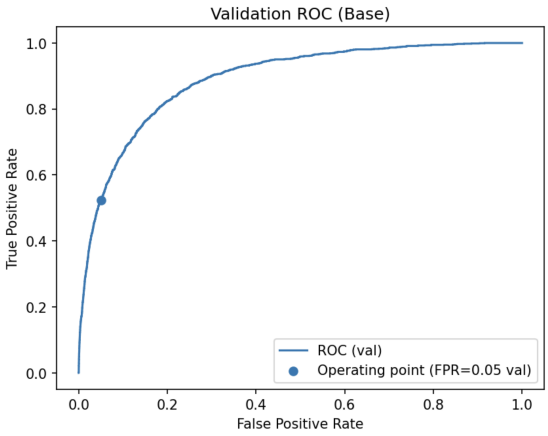

Fig. 2. Curva ROC en validación con umbral operativo (FPR=5%).

5.2 Resultados en Test

Aplicando el umbral obtenido en validación al conjunto de prueba, se observa que el AUC se mantiene en 0.89 y el AP mejora (0.211). El modelo no muestra sobreajuste, ya que el desempeño en test es consistente. Además, el FPR baja a 4.2% y el recall sube a 53%. Si se ajusta el umbral nuevamente al 5% de FPR, el recall sube hasta 56%. Se emplearon hiperparámetros con regularización (lambda_l2, num_leaves, min_child_samples, feature/bagging_fraction) y early stopping para controlar varianza. La similitud de AUC entre validación (0.89) y prueba (0.89) sugiere ajuste adecuado (sin sobreajuste).

Table 4. Métricas en Test

Métrica	Valor
AUC	0.890
AP (PR-AUC)	0.211
FPR (umbral validación)	0.042
Recall (umbral validación)	0.535
Recall (FPR=5%)	0.562

5.3 Fairness por Edad

Para evaluar imparcialidad, se utilizó la métrica Predictive Equality considerando la edad como atributo protegido. Se observa que los usuarios mayores de 50 años tienen un FPR mayor (10%) en comparación con los menores de 50 años (3%). El ratio FPR es de 0.329, lo que indica que el modelo genera más falsos positivos para adultos mayores, reflejando un sesgo que debe atenderse.

Table 5. Fairness: Predictive Equality (Edad)

Grupo	FPR
< 50 años	0.033
≥ 50 años	0.101

6 Ética y normatividad

6.1 Principios

La variable customer_age se incluyó en iteraciones iniciales y mostró alta contribución predictiva. Al ser un atributo protegido, medimos Predictive Equality por edad y observamos disparidad (FPR >=50 > FPR < 50). Documentamos el impacto de incluirla (ablation) y presentamos alternativas de mitigación.

6.2 Normatividad aplicable

Marco general de protección de datos y no discriminación: Ley Federal de Protección de Datos Personales en Posesión de los Particulares (México), LGI (no discriminación), y buenas prácticas de la industria financiera (KYC/AML). Este portafolio usa datos sintéticos de Kaggle con fines académicos.

6.3 Disparidad observada y mitigaciones

FPR ≥ 50 años: 0.101 vs FPR < 50: 0.033 (ratio=0.329). Posibles mitigaciones: Recalibración por grupo, Coste diferencial de falsos positivos por edad, Revisión de variables proxy y leakage, Umbrales condicionados sujeto a revisión legal/ética.

7 Conclusiones

7.1 Conclusiones generales

El desarrollo de este proyecto nos permitió implementar diferentes modelos de Machine Learning para un caso en el mundo real, en este caso detección de fraudes bancarios. Esto nos permitio evaluar y comparar diferentes enfoques en base a un conjunto de datos sintéticos con múltiples disparidades que pueden afectar la eficacia de los modelos si no son consideradas.

El análisis y resultados demostro que LightGBM fue la implementación mas sólida, alcanzando un AUC de 0.89 tanto en validación como en prueba. Considerando que los datos tenían un gran desbalance, el modelo pudo detectar mas de la mitad de los fraudes manteniendo el *FPR* dentro del límite del 5% que se establecio en equipo, lo que representa un buen logro en terminos de mantener un balance de falsos positivos.

Las métricas obtenidas, en particular el AUC y AP nos confirman que el modelo es capaz de distinguir patrones de fraude de una manera eficaz, igualemente la consistencia entre los resultados de validación y prueba hace ver que no existe un sobreajuste.

Pero por otra parte el análisis de Fairness, nos demuestra que existe un sesgo, el cual es esperado, que es que los clientes de mayor edad presentan una mayor tasa de falsos positivos.

Sin duda se demuestra que el uso de Machine Learning para detectar fraudes en un banco es de gran importancia, ya que permite detectar fraudes de una manera mas eficiente y con un menor costo que si fueran monitoreados individualmente, eso tambien nos permitio observar la gran importancia del Machine Learning no solo para este caso, sino para una gran cantidad de ámbitos de una forma muy diversa, desde la salud hasta el entretenimiento.

7.2 Conclusiones y aportes de los integrantes del equipo

- **Carlos Alberto Mentado Reyes**
  Observar en base a expermimentos como diferentes modelos se comportan cuando se pone a un reto de esta dificultad me ayudó, personalmente, a comprender que la solución más complicada no será necesariamente la que llegue al resultado ideal, aunque, experimentar con diferentes modelos también me ayudó a entender mejor sus posibilidades, sus requerimientos computacionales y su efectividad ante datos tan sesgados.
- **Fernanda Díaz Gutiérrez**
  Durante el reto con BAF, confirmé la importancia de diseñar un pipeline claro, robusto y eficiente acorde a los datos. En este caso, trabajé en el desarrollo del modelo principal usando LightGBM, que me permitió realmente entender cómo es que se manejan datos así de desbalanceados en la industria sin tener que recurrir a hacer oversampling y aprovechando distintas técnicas no necesariamente convencionales como primer "approach", como usar class weights, regularización y selección de umbrales para tener estabilidad en las métricas, etc. Entendí que a veces no es suficiente entrenar al modelo y ponerlo a la marcha, sino que tenemos que definir puntos operativos realistas (en este caso FPR=5%), y asegurarnos/comprobar la consistencia en la validación y el training para evitar overfitting. Además, me llevo la experiencia de revisar métricas como el fairness de los atributos protegidos (en este caso la edad), pues es importante entender el impacto que estos aspectos finalmente éticos tienen en un entorno bancario real. En conclusión, el reto me demostró cómo es que la teoría de Machine Learning es traducida a no más que una serie de decisiones de implementación y programación, que afectan directamente la confiabilidad de un modelo en la vida real, y espero poder aplicar esto en el siguiente reto para asegurar un trabajo excelente.
- **José Eduardo Puentes Martínez**
  Nuestro mundo es un mundo complejo, y tratar de entender cómo interactúan

varios elementos en los eventos que observamos es bastante interesante, a veces se pueden comprender y observar a simple vista, pero en otras ocasiones se requiere de perspectivas mucho más profundas para evaluar estos eventos, y el aprendizaje máquina demuestra cómo es que se pueden utilizar herramientas actuales para tratar de resolver y entender mejor estas interacciones. Al trabajar con estas herramientas desde sus raíces, pude entender mucho mejor cómo es que funcionan estos procesos, como la obtención de datos, entrenamiento, validación, testing, etc. y comprendo mucho mejor la importancia de esto y sus verdaderas aplicaciones y limitaciones.

- **Raymundo Iván Díaz Alejandre**
  Fue interesante observar la gran importancia del Machine Learning en casos del mundo real y la gran cantidad de recursos que pueden ser ahorrados al implementar una solución eficaz para este tipo de problemas, se me hace increíble como cualquier industria se puede beneficiar ampliamente de Machine Learning. Aprendí muchas cosas, especialmente la gran importancia de analizar los datos antes de empezar con una solución, esto puede ahorrar mucho tiempo y llevar a desarrollar una mejor solución. Estoy muy emocionado del siguiente reto a resolver y brindar una buena solución.

7.3 Contribuciones de los integrantes del equipo

- **Carlos Alberto Mentado Reyes:** Encargado de una gran parte del EDA y de los primeros intentos de modelado. Intentó con técnicas como el oversampling y el uso de varios clasificadores, permitiéndonos entender el dataset, identificar retos y limitaciones, y orientar al equipo hacia un pipeline más adecuado para el reto propuesto.
- **José Eduardo Puentes Martínez:** Participó en la fase de EDA y experimentación inicial con los datos, buscando implementar distintos enfoques de preprocesamiento/modelado. Su trabajo nos permitió descartar todas las estrategias que no fueron elegidas finalmente, y su documentación del por qué y cómo es que afectaban a las métricas fue clave para el éxito del proyecto.
- **Fernanda Díaz Gutiérrez:** Responsable de diseñar y desarrollar el pipeline final en LightGBM, implementando el preprocesamiento de los datos sin leaks, la selección de umbral a FPR=5%, el manejo del desbalance con class weights, y la integración de fairness por edad. Finalmente, trabajó en la creación y organización del repositorio, así como su documentación de uso en el README.
- **Raymundo Iván Díaz Alejandre:** Ayudó al análisis del desbalance de los datos y a la discusión de estrategias. Desarrolló un prototipo de red neuronal, que si bien no se incluye en la entrega final, fue usado para profundizar en alternativas. Diseñó y construyó la interfaz web usando React y Vite, mostrando de manera interactiva e intuitiva los resultados del modelo.

8 Referencias

Jesus, S. et al. (s.f.) *Turning the Tables: Biased, Imbalanced, Dynamic Tabular Datasets for ML Evaluation* 2DCC Faculdade de Ciências da Universidade do Porto, Portugal