Modelo de detección de Fraude

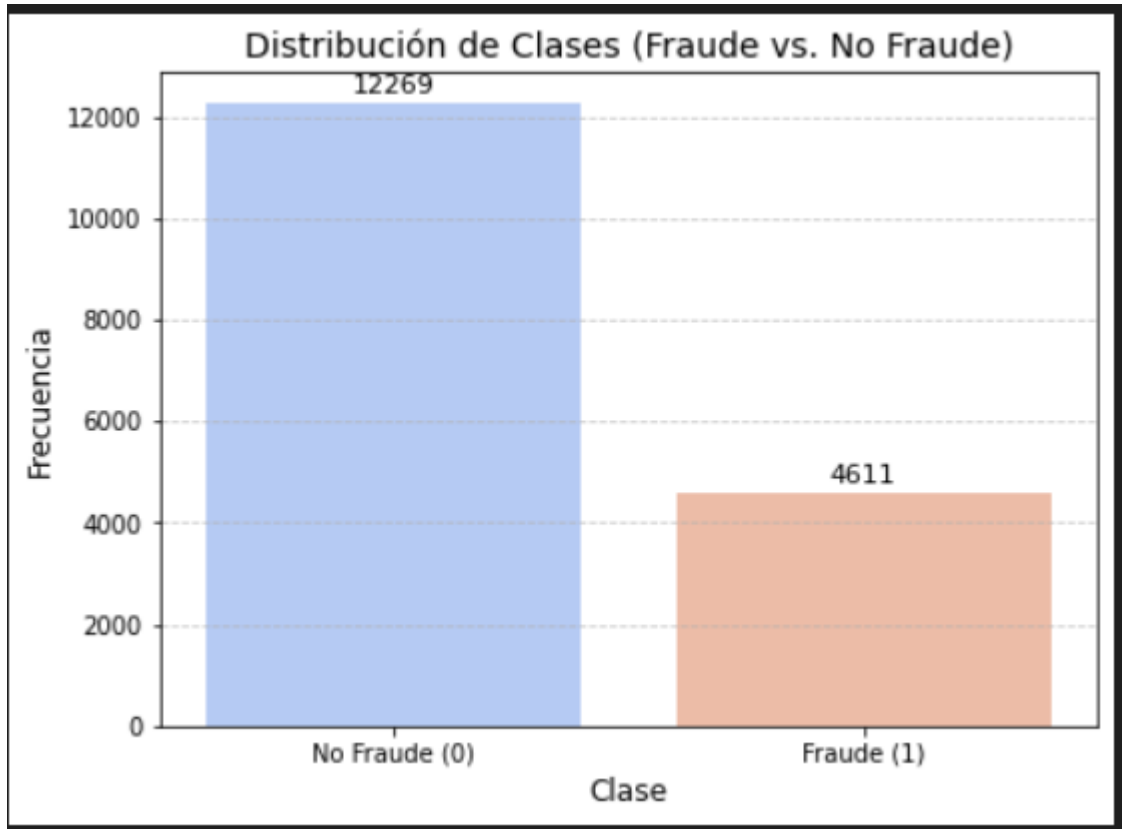
Andrés Julián Jurado Castaño

Introducción

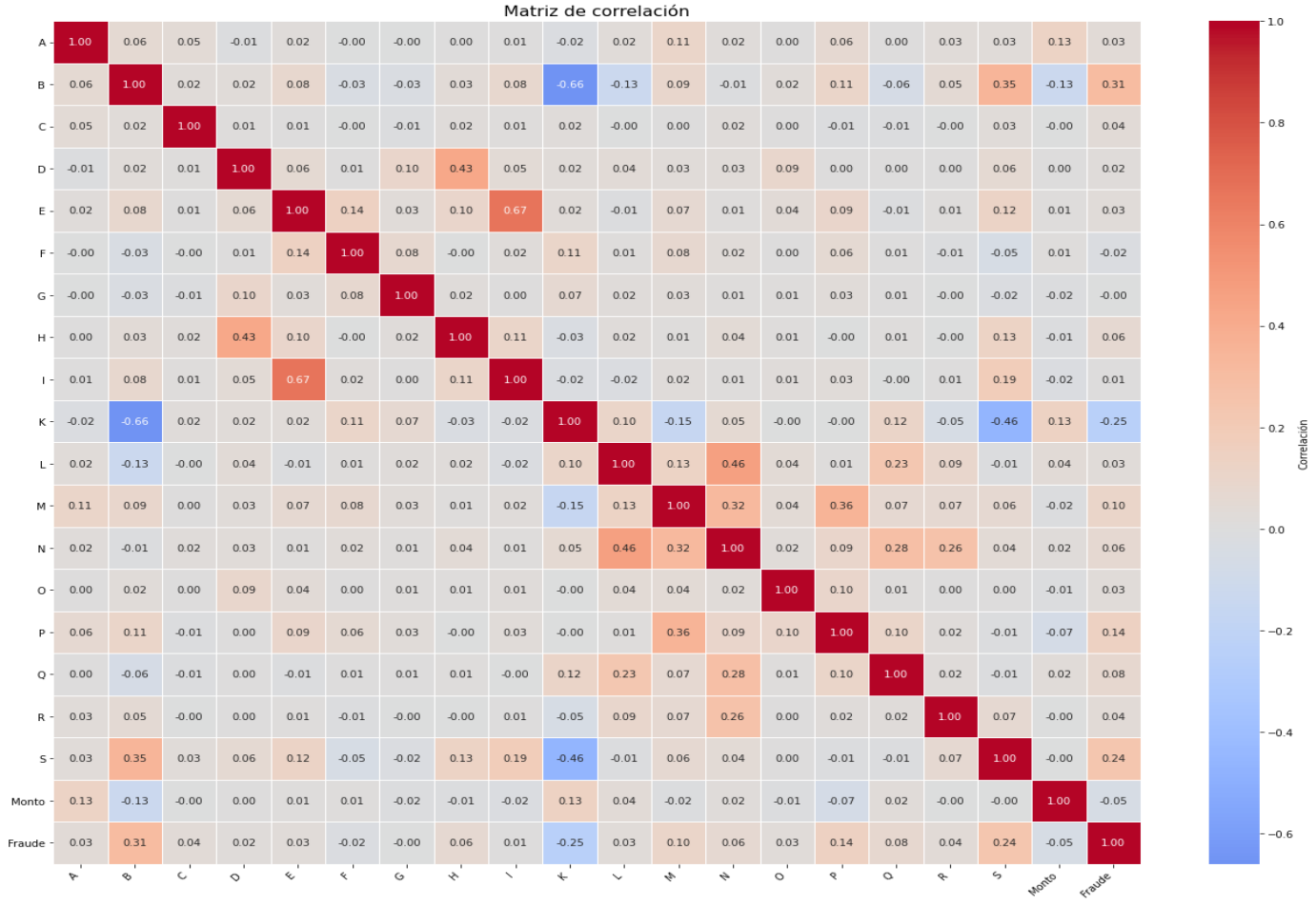
- El data set presenta (como es habitual) un desbalanceo en la variable a predecir (Fraude).
- Las features predictoras no poseen un nombre especifico para entender en contexto lo que significa cada una, por lo que se debe hacer un análisis exploratorio para explotar la capacidad predictora de cada una.
- Mas allá de usar técnicas clásicas de Machine Learning para maximizar la capacidad de predicción y usar métricas de evaluación para estas, debe tenerse en cuenta la ganancia esperada para influir en el entrenamiento del modelo y para el calculo del umbral de predicción, de esta forma no existe ambigüedad en el manejo de este umbral y en los resultados.

Análisis exploratorio de los datos

- Desbalanceo de la variable fraude



Correlación de las variables

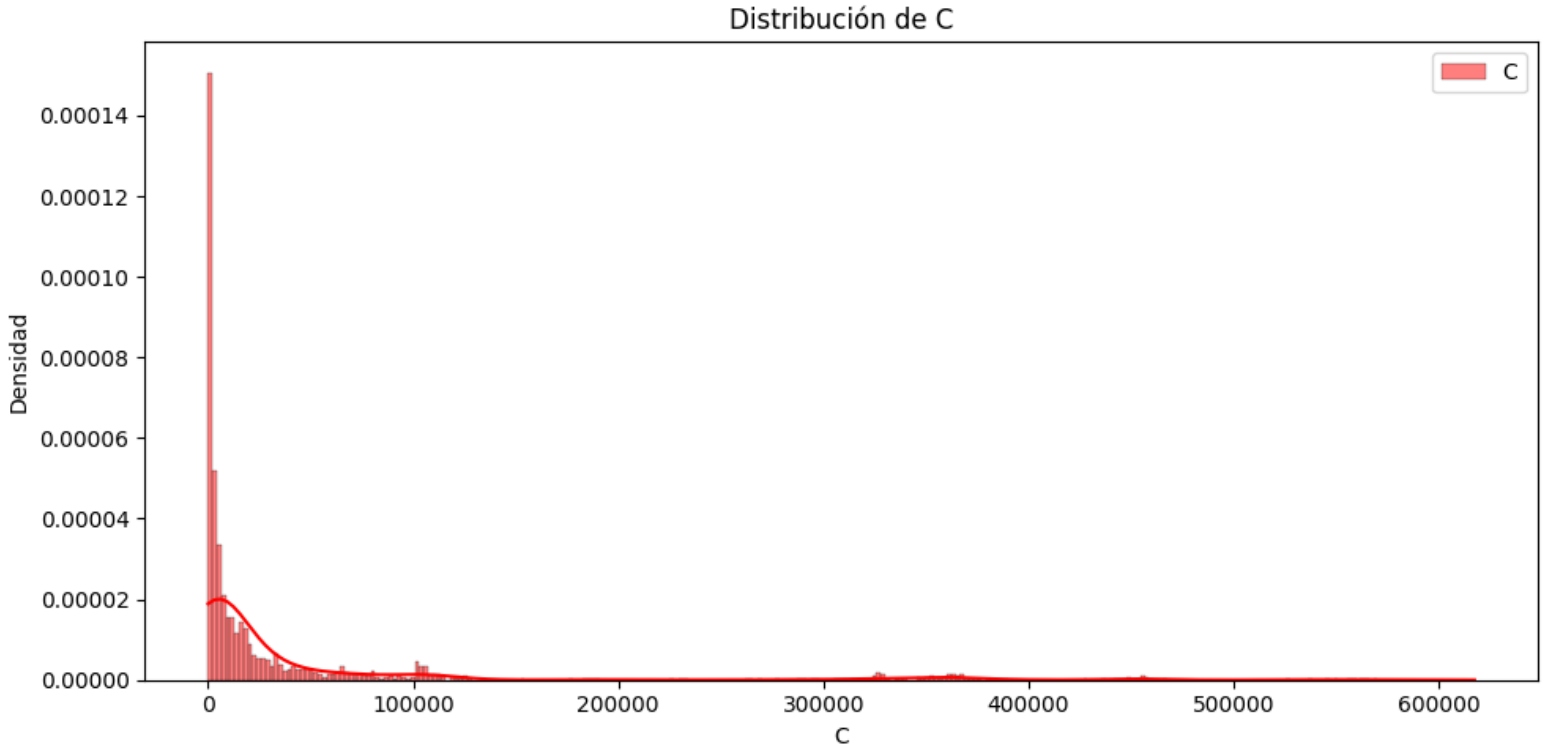


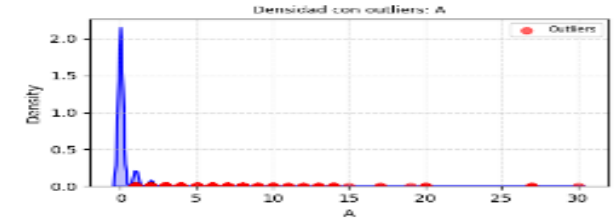
No se observan correlaciones fuertes que lleven a pensar problemas de multicolinealidad.
La Variable K tiene alta cantidad de datos faltantes pero los datos que existen que tiene buena correlación con otras variables.

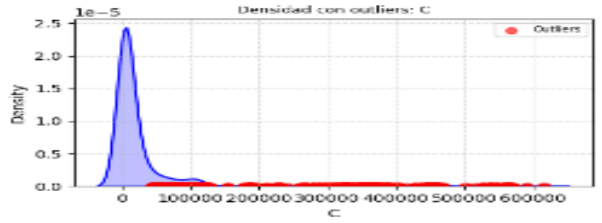
Muchas de las variables contienen distribuciones sesgadas a hacia la derecha, el modelo a usar, puede manejar bien este tipo de variables con estas distribuciones, luego de hacer pruebas, no hubo una mejora significativa al aplicar logaritmo natural sobre la variable monto o con alguna otra, por lo que se usan sus versiones originales.

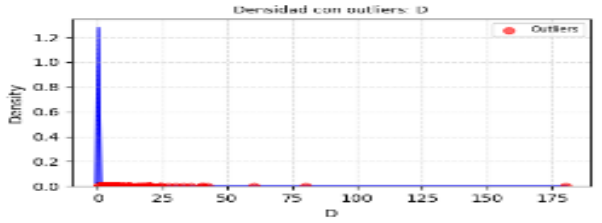
A su vez los modelos pueden manejar las escalas originales de los datos.

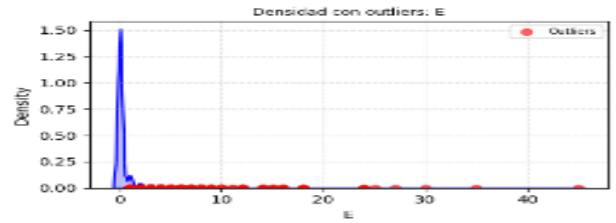
La columna ‘C’ no tenia una cantidad significante de valores faltantes, por lo que debido a su distribución se imputó con la mediana.




Densidad con outliers: A
Outliers
Density
A


Densidad con outliers: C
1e−5
Outliers
Density
C


Densidad con outliers: D
Outliers
Density
D


Densidad con outliers: E
Outliers
Density
E


Densidad con outliers: F
Outliers
Density
F


Densidad con outliers: G
Outliers
Density
G


Densidad con outliers: H
Outliers
Density
H

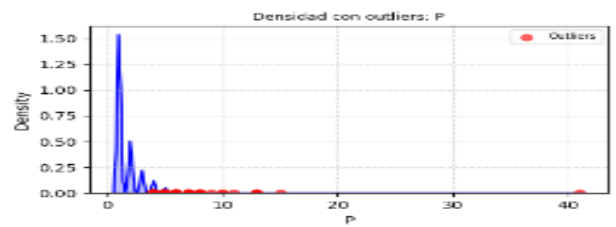

Densidad con outliers: I
Outliers
Density
I

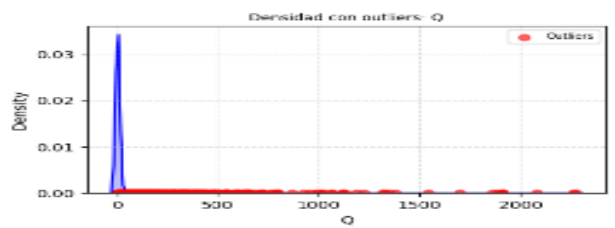

Densidad con outliers: L
Outliers
Density
L

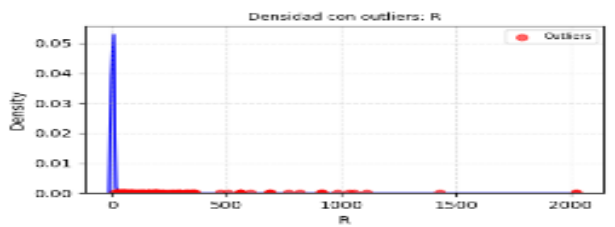

Densidad con outliers: M
Outliers
Density
M


Densidad con outliers: N
Outliers
Density
N

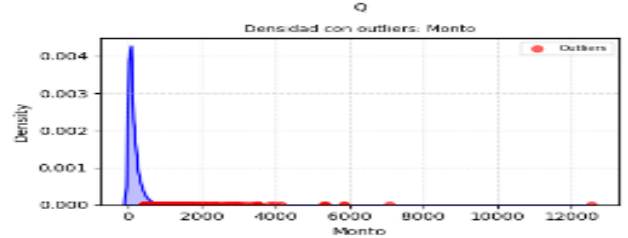

Densidad con outliers: O
Outliers
Density
O


Densidad con outliers: P
Outliers
Density
P


Densidad con outliers: Q
Outliers
Density
Q

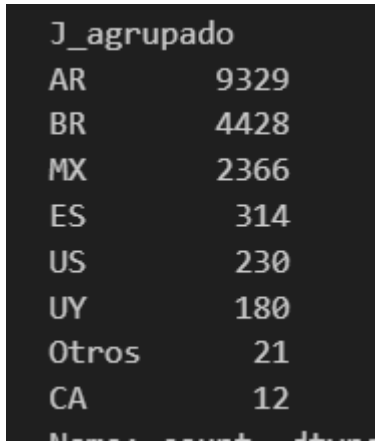

Densidad con outliers: R
Outliers
Density
R


Densidad con outliers: S
Outliers
Density
S


Densidad con outliers: Monto
Outliers
Density
Monto

Transformación de datos

- La columna ‘J’, que representa los países, contiene algunas categorías con muy pocas observaciones. Para evitar la creación de variables dummies con poca representación y no caer en sobredimensión del dataset, estas categorías con poca presencia se agruparán en una nueva categoría denominada ‘Otros’ al aplicar el onehotencoder
- Específicamente los países que contengan menos de 10 observaciones



J_agrupado	
AR	9329
BR	4428
MX	2366
ES	314
US	230
UY	180
Otros	21
CA	12

Debido a la falta de contexto respecto a las variables, la columna ‘K’ será eliminada por tener cerca del 80% de valores faltantes.

Modelos propuestos

- Se entrenan y se comparan 3 modelos:
- - Xgboost con función de ganancia personalizada
- - Xgboost con Optimización Bayesiana y función personalizada
- - Random Forest con Optimización Bayesiana
- Se hace una partición del 70% de los datos de entrenamiento y el 30% para evaluación.

Funciones Personalizada

- Teniendo en cuenta que la necesidad principal es maximizar las ganancias se usará una función personalizada para la evaluación de los modelos la cual se define como:

$$\textbf{Ganancia} = \sum_{i \in \text{Aprobadas}} [(0.25 \times M_i) \cdot \mathbb{I}(y_{\text{verdaderos},i} = 0) - (1.0 \times M_i) \cdot \mathbb{I}(y_{\text{verdaderos},i} = 1)]$$

El modelo con la mejor Ganancia Total es el que logra el **mejor equilibrio** entre:

- **Maximizar** la aprobación de transacciones legítimas (ganancia).
- **Minimizar** la aprobación de transacciones fraudulentas (pérdida).

Esta la función explicita que se usará para escoger el umbral, teniendo en cuenta también la capacidad de este para hacer buenas predicciones respecto al fraude.

$$\mathbf{g_i} = \frac{\partial \mathbf{L_i}}{\partial \eta_i} = \begin{cases} 0.25 \cdot M_i \cdot p_i & \text{si } y_i = 0 \text{ (No Fraude)} \\ -1.0 \cdot M_i \cdot (1 - p_i) & \text{si } y_i = 1 \text{ (Fraude)} \end{cases}$$

$$\mathbf{h_i} = \frac{\partial^2 \mathbf{L_i}}{\partial \eta_i^2} = \begin{cases} 0.25 \cdot M_i \cdot p_i(1 - p_i) & \text{si } y_i = 0 \text{ (No Fraude)} \\ 1.0 \cdot M_i \cdot p_i(1 - p_i) & \text{si } y_i = 1 \text{ (Fraude)} \end{cases}$$

$$\mathbf{Ganancia_{total}} = \left(0.25 \sum_{i \in \text{TN}} M_i\right) - \left(1.0 \sum_{i \in \text{FN}} M_i\right)$$

Símbolo	Descripción
$\mathbf{M_i}$	Monto de la Transacción. Utilizado como peso para el impacto económico.
$\mathbf{y_i}$	Etiqueta Verdadera binaria (0 : No Fraude, 1 : Fraude).
η_i	Log-Odds (Predicción Bruta). La variable que el modelo optimiza.
$\mathbf{p_i}$	Probabilidad Predicha de fraude ($p_i = \sigma(\eta_i)$).
$\mathbb{I}(\cdot)$	Función Indicadora. Devuelve 1 si la condición es cierta.
$\mathcal{L}_i$	Pérdida Económica ($\mathcal{L}_i = -\mathbf{Ganancia}_i$).
$\mathbf{g_i}$	Gradiente ($\frac{\partial \mathcal{L}_i}{\partial \eta_i}$). Indica la dirección del error.
$\mathbf{h_i}$	Hessiano ($\frac{\partial^2 \mathcal{L}_i}{\partial \eta_i^2}$). Determina la curvatura y estabilidad.
TN	Verdaderos Negativos en fraude (Legítimo aprobado).
FN	Falsos Negativos (Fraude aprobado).

Xgboost Función personalizada

- Para ambos modelos xgboost se usa la **Ganancia Máxima Esperada** y se descarta Logloss dado el problema de negocio. Así, la ganacia se define como:
- La **utilidad neta** calculada al evaluar el costo de los errores (Falsos Negativos) y el beneficio de los aciertos (Verdaderos Negativos en el conjunto de prueba) utilizando el monto real de la transacción.
- Por lo tanto el supuesto se define como, cuesta 4 veces más dejar pasar un fraude que rechazar una verdadera transacción.
- Esta se puede definir como una función de perdida personalizada, que usa el hessiano y el gradiente especifico de la función para la construcción de los arboles previamente definido en las diapositivas anteriores siguiendo el método de convergencia del Xgboost basado en el gradiente y el Hessiano dado que la función es diferenciable.
- Además, logra darle peso a las transacciones fraudelantas balanceando los datos sin incurrir en parámetros de balanceo o técnicas SMOTE

Random forest con Optimización Bayesiana

Este modelo debido a su naturaleza de creación de arboles, utiliza métricas de pureza (Gini o Entropía),estas no permiten usar la función de ganancia para entrenar, pero si usa la función de ganancia total para la evaluación y decisión del random forest para maximizar la rentabilidad.

Se usa optimización bayesiana ya que es computacionalmente mas barato que usar (*Grid Search*).

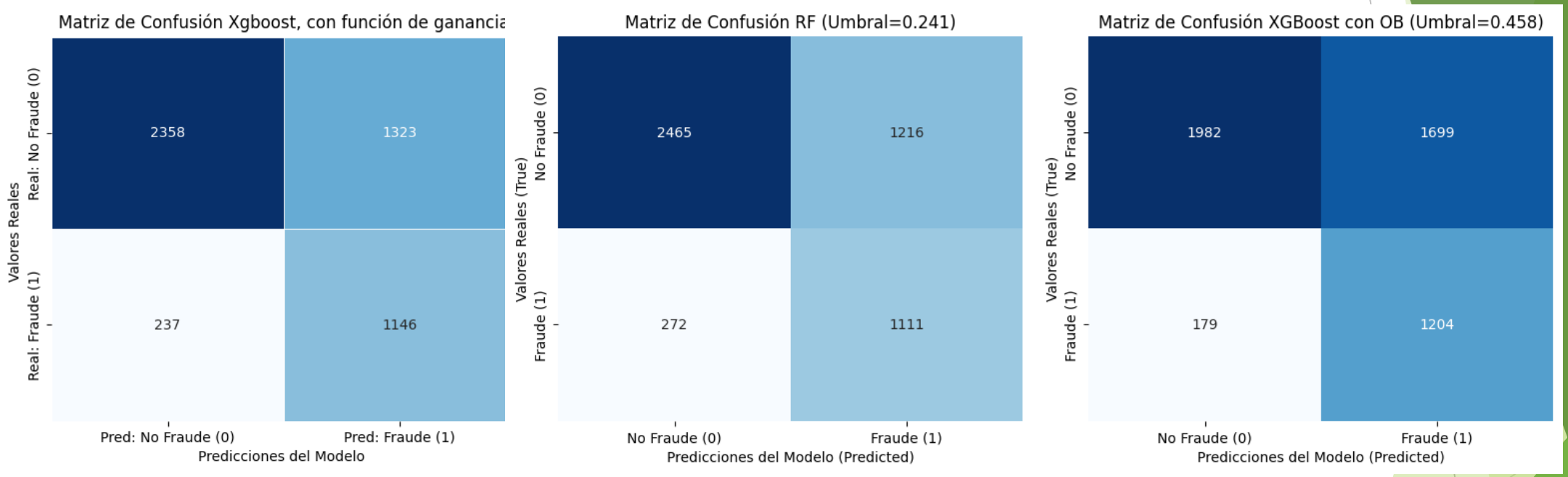
El RF no entrena directamente con la ganancia, sino que entrena para un excelente AUC, y luego usa la Ganancia como una métrica de calibración final para encontrar el umbral de negocio ideal.

Xgboost con Optimización Bayesiana

- El objetivo aquí es encontrar los hiperparámetros que produzcan el mejor modelo estadístico, usando la eficiencia de la Optimización
- La OB maximiza el AUC para **seleccionar los HPs**. El modelo que usa esos HPs se entrena a continuación, pero el proceso de entrenamiento es el que necesita la función de personalizada de perdida
- La OB define la métrica de optimización para el entrenamiento final. Una vez que la OB encuentra la mejor "estructura" de HPs (por AUC), el modelo se entrena usando la Ganancia personalizada como función de pérdida. Esto asegura que los árboles se construyan para optimizar directamente la rentabilidad, no solo el AUC.

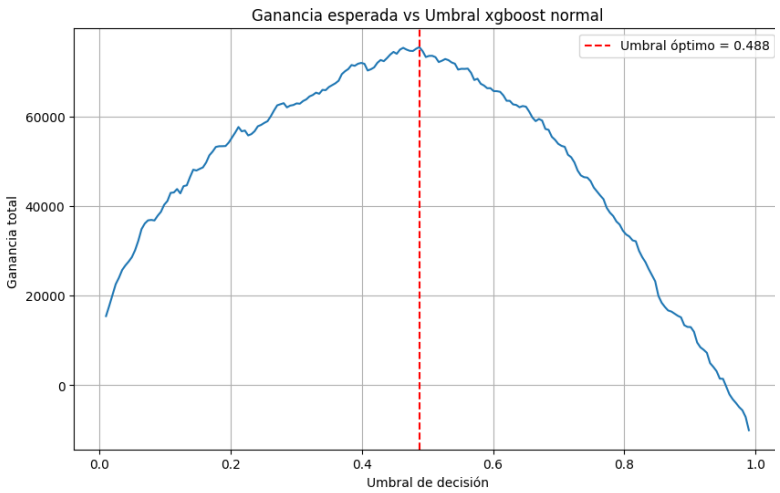
Métricas de los 3 modelos

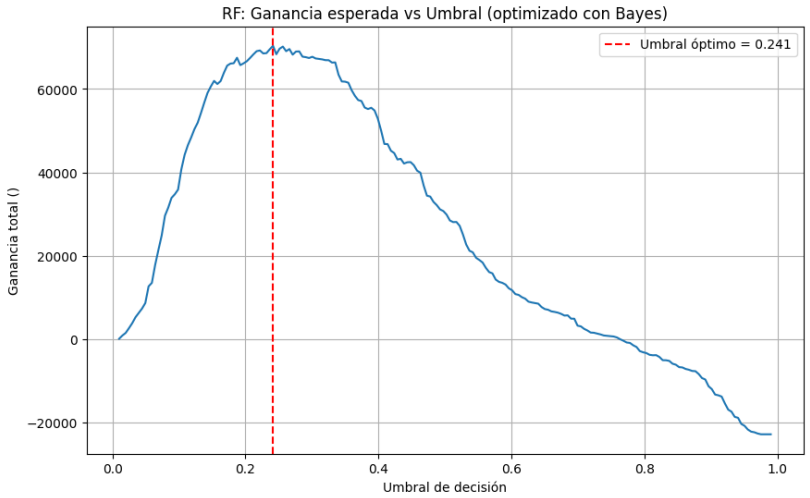
Comparación matrices de confusión

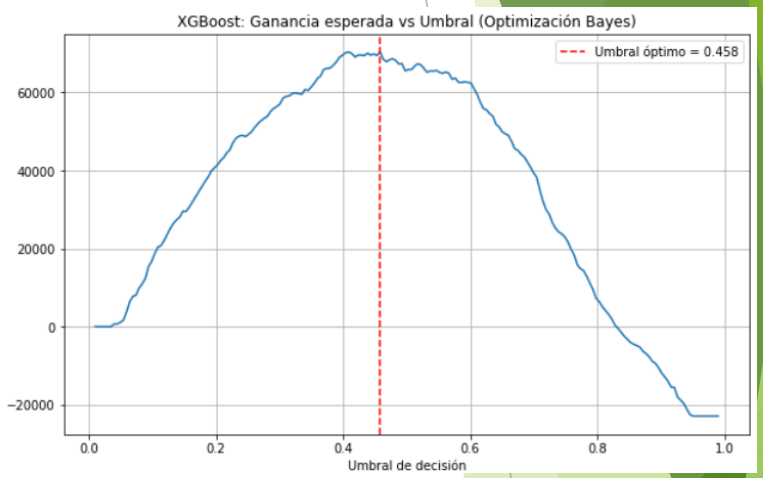


El modelo que no busca hiperparametros por AUC logra separar mejor el fraude, basado en la función de pérdidad

Ganancias para neta para la compañía en test por modelo teniendo en cuenta el umbral



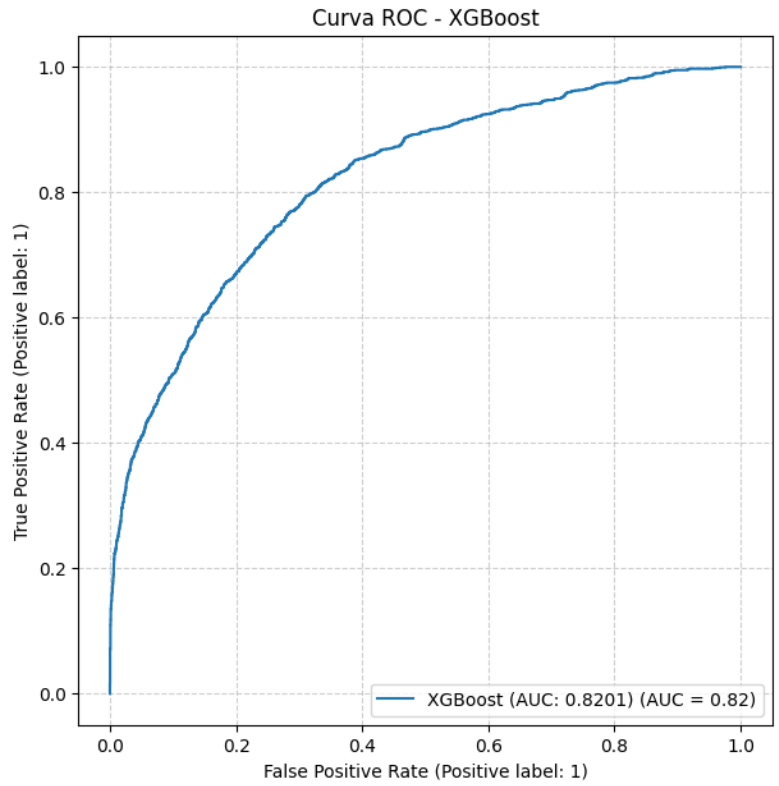


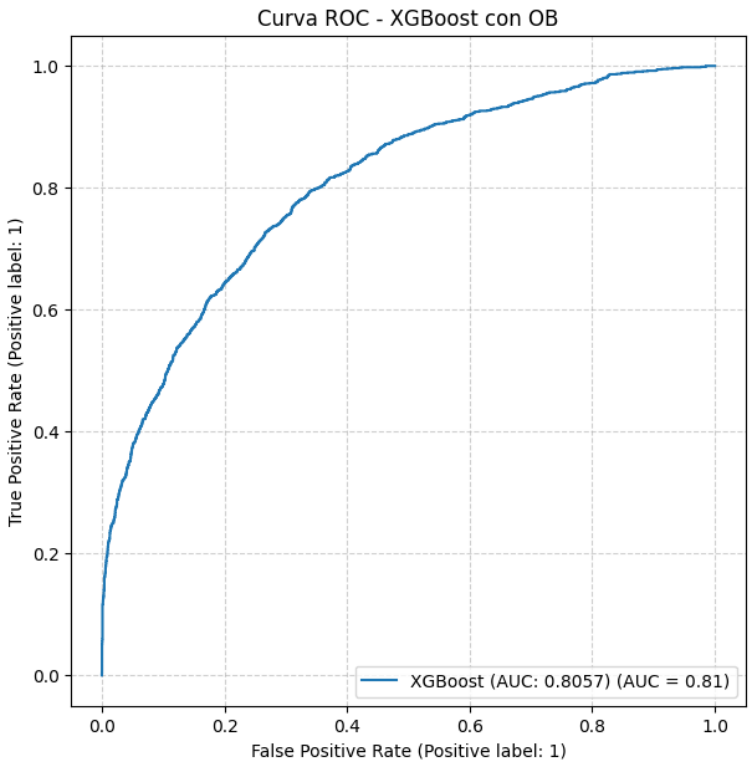


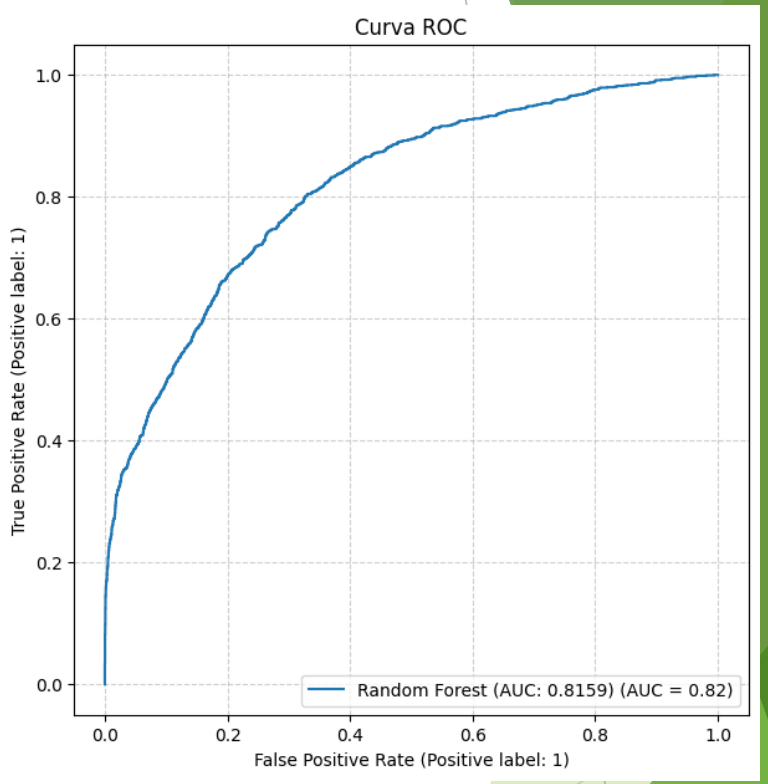
Ganancia máxima esperada: 75,446

Ganancia máxima esperada RF: 70,379, el modelo es mas sensible frente a cambios en el umbral

Ganancia máxima esperada XGBoost: 70,349







El modelo Xgboost que no contempla OB, entrena el modelo maximizando la ganancia, mas no el AUC, mientras que los otros dos usan tunnig de Hiperparametros que Maximice el AUC pero usan la función personalizada para el entrenamiento.

Conclusiones

- El modelo XGBoost que incorpora la función personalizada para minimizar la pérdida de ganancia es el de mejor desempeño, ya que optimiza directamente la métrica de negocio. Pese a que la mejora en resultados no es muy grande, el modelo es intrínsecamente eficiente en términos computacionales para la convergencia y el escalamiento, lo que lo hace ideal para incorporaciones en producción.

- Se sugiere usar este modelo mientras se mantenga el margen de ganancia, Para refinar aún más la estrategia, se pueden incorporar mejoras en la función, por ejemplo incorporar una función que logre capturar variaciones en ganancia en mercado pago en las transacciones, también podría evaluarse la perdida económica de pasar transacciones que no son fraudelentas como fraude y así optimizar el umbral de decisión, de este modo, se maximizaría la rentabilidad de las transacciones aun mas. Por último, la función de ganancia personalizada puede adaptarse a modelos más sofisticados, como una red neuronal, pero debe evaluarse el trade-off entre la posible mejora de resultados y los costos de latencia, intepretabilidad y eficiencia de incorporación en un ambiente de producción.