RappiCard

Challenge DS

Índice

Introducción

- Construcción de un modelo que sea capaz de detectar Defraudadores.

- Contamos con un dataset de 2,697 registros y 16 columnas.

- Observamos que tenemos variable objetivo “fraude” con un desbalance evidente teniendo 97% de nuestro conjunto como No fraudulentas y solo el 3% como fraude.

- El objetivo es encontrar el mejor método para poder obtener mejores resultados en nuestro modelo.

Introducción al modelo

Regresión Logística

Análisis Exploratorio:

- 1. Carga de datos y librerías necesarias.
  - pandas: Usada para manejar y analizar datos.
  - numpy: Proporciona soporte para arrays y matrices
  - json: Para manejar datos en formato JSON.
  - sklearn.model_selection: Contiene funciones para dividir los datos en grupos de entrenamiento y prueba, una parte crucial en el desarrollo de modelos de machine learning.
  - sklearn.preprocessing: Proporciona herramientas para el preprocesamiento de datos.
- 2. Procesamiento de los datos.
  - Incluye el uso de OneHotEncoder y LabelEncoder, que son técnicas para convertir variables categóricas en un formato numérico parte de las buenas prácticas en nuestro modelo.

- 3. Modelado.
  - para este punto empleamos sklearn.linear_model para el uso de nuestro modelo de regresión logística.

- 4. Entrenamiento del Modelo.

  - train_test_split de sklearn.model_selection usado para dividir los datos en conjuntos de entrenamiento y prueba, esencial para validar la eficacia del modelo.

  - El modelo es entrenado usando el método .fit, indicativo de la construcción y ajuste del modelo a los datos de entrenamiento.

Metodología del Modelo

Regresión Logística

Dentro de este modelo se probaron 3 técnicas en las cuales se pretende ir corrigiendo el desbalance de nuestro dataset.

Como primera iteración generamos instancia de la regresión logística “LogisticRegression” y se utilizó “cross_val_score” esta función evalúa un puntaje utilizando la validación cruzada. Aquí, usamos accuracy como la métrica de scoring para evaluar el rendimiento del modelo.

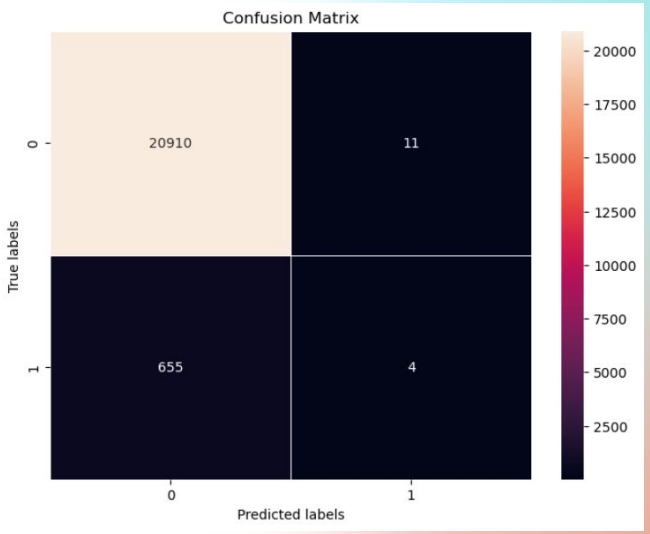
Conclusión de primer iteración:

Alta Precisión Promedio con un Accuracy aproximadamente 96.91% esto puede ser engañoso y muy peligroso por los falsos positivos y sobre todo el desbalanceo que tenemos en el Dataset. Vamos a seguir análizando

Conclusión de primer iteración:

Falsos Negativos (FN): 655 cuentas fraudulentas, clasificándolas erróneamente como no fraudulentas.

Con estos resultados observamos que el modelo no está siendo favorable y debemos corregir el desbalance

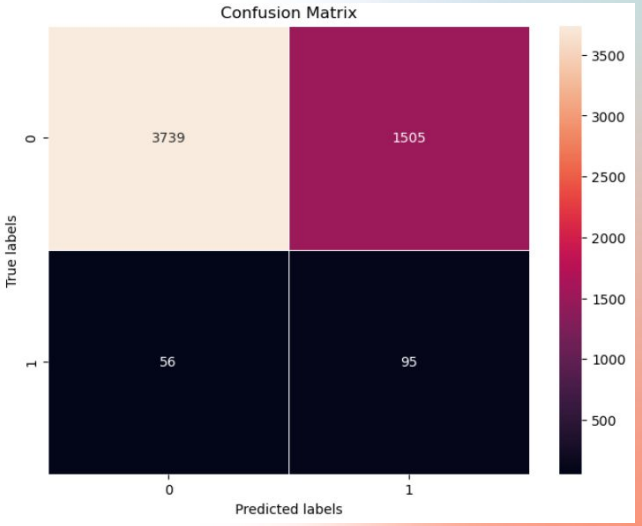


Metodología del Modelo

Regresión Logística

	precision	recall	f1-score	support
0	0.99	0.71	0.83	5244
1	0.06	0.63	0.11	151
accuracy			0.71	5395
macro avg	0.52	0.67	0.47	5395
weighted avg	0.96	0.71	0.81	5395

Como segunda iteración usamos la técnica de Balanceo de clases con “class_weight='balanced” entrenando al modelo ponderado en el conjunto de datos de entrenamiento , esto para tratar de ajustar los pesos de las clases.



Conclusión de segunda iteración:

Desbalance en el Recall: Aunque el modelo tiene una alta tasa de detección de fraudes (recall para la clase 1), también tiene un alto porcentaje de falsos positivos.

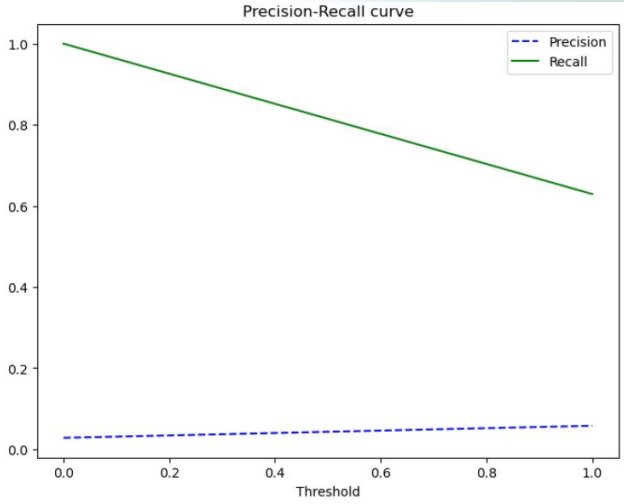
Metodología del Modelo

Regresión Logística

	precision	recall	f1-score	support
0	0.99	0.70	0.82	5244
1	0.06	0.63	0.11	151
accuracy			0.70	5395
macro avg	0.52	0.67	0.46	5395
weighted avg	0.96	0.70	0.80	5395

Como tercera iteración usamos la técnica SMOTE (Synthetic Minority Over-sampling Technique). utilizada para desequilibrio de clases en problemas de clasificación.

El objetivo es resolver el desbalance que hemos estado viendo, atacando la clase menor



Conclusión de tercera iteración:

Clase 0 (No Fraude):Recall del 70% las no fraudulentas se están clasificando erróneamente como fraudulentas falsos positivos.

Clase 1 (Fraude) : Precisión: Baja (6%), lo que implica que de todas las transacciones que el modelo identifica como fraudulentas, solo un pequeño porcentaje lo son realmente. Recall: Alto (63%), lo cual es positivo ya que muestra que el modelo puede identificar una gran proporción de todas las cuentas fraudulentas reales.

Curva ROC y el área bajo la curva (AUC) para obtener una medida más clara de la capacidad del modelo para distinguir entre las clases.

Con este gráfico se intenta observar la curva de precisión y recall donde vemos que cambian a medida que ajustamos el umbral para la clasificación de una clase minorista

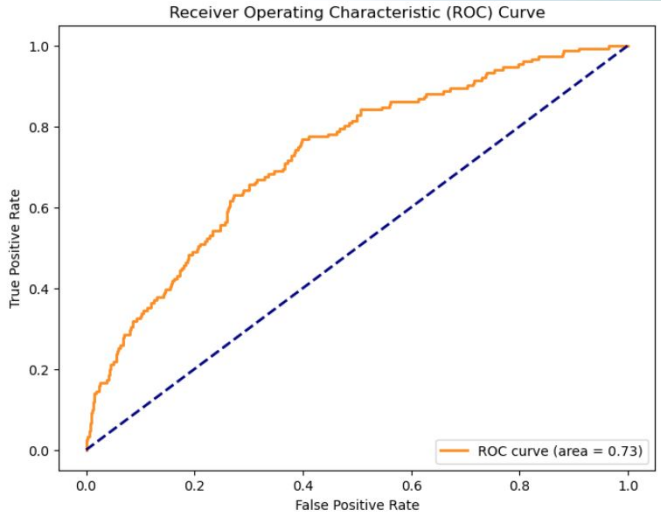
Metodología del Modelo

Regresión Logística

Como cuarta iteración usamos la técnica logistic_model_weighted. Este método entrena el modelo en el conjunto de datos de entrenamiento X_train con las etiquetas de clase y_train

El propósito de este código es entrenar un modelo que sea más efectivo para detectar la clase minoritaria en un conjunto de datos desequilibrado.

	precision	recall	f1-score	support
0	0.99	0.71	0.83	5244
1	0.06	0.63	0.11	151
accuracy			0.71	5395
macro avg	0.52	0.67	0.47	5395
weighted avg	0.96	0.71	0.81	5395



Conclusión de la cuarta iteración:

Recall para no fraudulentos: 71% (falsos positivos)

Recall para fraudulentos: 63% una mejora significancia, el modelo ahora es capaz de identificar más fraudes

Podemos observar el gráfico un AUC del 73% lo que nos dice que si bien no es el mejor modelo tiene la capacidad de distinguir relativamente bien entre la clase positiva y negativa. Mientras que la cuerva ROC esta por encima de la línea de clasificación aleatoria.

¿Por qué Regresión Logística?

XGBoost

No fraude:

- Precisión del 97%, enfocado en la clase mayor
- Recall Excelente casi 100% el modelo identifica correctamente casi todas las transacciones no fraudulentas.
- F1-Score del 98%, que es un equilibrio entre precisión y recall.

Fraude:

- Precisión 29% de todas las transacciones que el modelo identifica como fraude, menos de una tercera parte son realmente fraude.
- Recall Muy bajo 2% el modelo casi no detecta las transacciones fraudulentas.
- F1-Score Muy bajo 0.03%, debido a la baja precisión y recall.
- AUC-ROC: Moderado 67%, que es mejor que una elección aleatoria pero muestra espacio para mejora.

Conclusión:

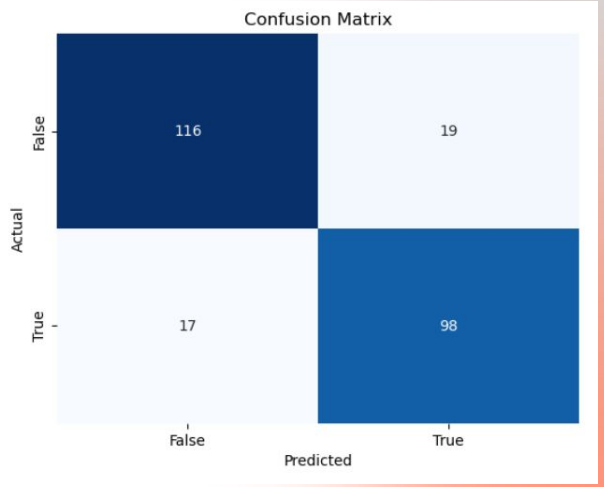
- Alta precisión en la detección de no fraudulentos (False), pero muy bajo recall para fraudulentos (True), lo que indica que detecta casi todas las transacciones legítimas, pero falla en identificar las fraudulentas.
- AUC-ROC es moderado, sugiriendo que el modelo tiene una capacidad razonable para distinguir entre clases.

```
Reporte de Clasificación:
              precision    recall  f1-score   support

       False       0.97      1.00      0.98      7850
        True       0.29      0.02      0.03       243

    accuracy                           0.97      8093
   macro avg       0.63      0.51      0.51      8093
weighted avg       0.95      0.97      0.96      8093

Matriz de Confusión:
[[7840   10]
 [ 239    4]]
AUC-ROC: 0.6709113784697649
```



¿Por qué Regresión Logística?

Random Forest

No fraude

Recall: Con un 99%, sugiere que el modelo identificó correctamente casi todas las instancias de no fraude.

Fraude

Precisión: Con un 30%, muestra que de las transacciones que el modelo predijo como fraude, solo el 30% resultaron ser fraudulentas.

Recall: Con un 9%, revela que el modelo solo detectó una pequeña proporción de todas las transacciones fraudulentas reales.

Random Forest parece ofrecer el mejor equilibrio entre precisión y recall para la clase de interés (fraudulentos) de los tres modelos, lo que lo hace más adecuado para la detección de fraudes en este conjunto de datos.

Reporte de clasificación:

	precision	recall	f1-score	support
False	0.97	0.99	0.98	5244
True	0.30	0.09	0.13	151
accuracy			0.97	5395
macro avg	0.64	0.54	0.56	5395
weighted avg	0.96	0.97	0.96	5395

RappiCard