TP1 - Análisis de Plataforma de Ventas Online

Aprendizaje de máquina I

Domenje, Carlos R.

Una plataforma de ventas online nos contrata para que realicemos un modelo que nos permita detectar un posible fraude dada cierta operación para ello contamos con un dataset

https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset

Utilizando los modelos de clasificación vistos hasta el momento generar un notebook que permita de ser posible resolver el problema que nos está planteando el cliente.

Descripción de las columnas.

A continuación se describen brevemente las columnas que intervienen en el dataset.

- **step:** representa una unidad de tiempo donde 1 step equivale a 1 hora
- **type:** tipo de transacción en línea
- **amount:** el importe de la transacción
- **nameOrig:** cliente que inicia la transacción
- **oldbalanceOrg:** saldo antes de la transacción
- **newbalanceOrig:** saldo después de la transacción
- **nameDest:** destinatario de la transacción
- **oldbalanceDest:** saldo inicial del destinatario antes de la transacción
- **newbalanceDest:** el nuevo saldo del destinatario después de la transacción
- **isFraud:** transacción fraudulenta

Para este trabajo se tomará la columna **isFraud** como nuestra variable a predecir. Es decir, ante un nuevo dato de entrada queremos clasificar si la transacción que se esta realizando es fraudulenta o no.

Análisis de datos

Para el análisis de los datos se utilizaron las siguientes herramientas:

- Función **describe** para el análisis estadístico de los datos.
- Verificación de datos nulos en cada una de las columnas.
- Información del tipo de datos que contiene el dataset.
- Verificación gráfica de la distribución de las variables.
- Detección de Outliers a través de gráficos de histograma, QQ-Plot y Boxplot.
- Transformación de variables, utilización de PowerTransformer y QuantileTransformer para normalizar las columnas.
- Análisis gráfico de las variables del dataset para tener una mejor aproximación de sus valores.
- Análisis de la variable de salida. Verificación del balance de los datos.
- Gráfico de correlación de las variables.
- Transformación de la variable categórica TYPE en variables numéricas, anexadas a las columnas del dataset.

Con estas diferentes metodologías utilizadas, se pudo quitar del dataset aquellas variables que no presentan relevancia en la decisión. Tenemos dos columnas que representan los nombres de las cuentas de destino y origen donde no tenemos aportes a la información para determinar si es o no fraudulenta. Además tenemos la variable isFlaggedFraud que tiene el valor 1 en todas las filas por lo cual también se decidió eliminar.

Por otro lado, la variable categórica TYPE, contiene 5 categorías, que sí son relevantes para el análisis ya que el fraude está fuertemente relacionado al tipo de transacción que se está realizando. Para ello, se decidió transformar las variables en columnas, pasando de una variable categórica a 5 numéricas.

Por último se analizó la variable Target **isFraud**, la cual posee un desbalance muy grande frente a las detecciones fraudulentas vs a las que no lo son.

Al realizar este análisis, se va a tener en cuenta en las predicciones este dato para así tomar medidas para mitigar posibles errores en los algoritmos a utilizar.

Separación de variables

Se utiliza la función **train_test_split** con la opción de stratify en TRUE. Esta opción permite distribuir de igual forma la variable de salida en función de la cantidad de datos de entrenamiento y testeo, por

lo cual, nos aseguramos que al tener el target desbalanceado, podamos tener en los dos sets, la variable de salida en sus dos estados.

Por último, podemos chequear esta distribución utilizando el **value_count** para contar las cantidades.

Regresión Logística

El primer algoritmo utilizado es Regresión Logística, donde se pudo observar que al aplicarlo directamente con los datos previamente separados obtuvimos una precisión del 96% pero si evaluamos la matriz de confusión, vemos que se equivocó mucho en la cantidad de datos positivos, es decir tenemos muchos falsos negativos. Si vemos en la métrica recall, tenemos que para la detección de 1 es del 44% frente al 99.9% de la detección de 0.

Luego, se decide aplicar un Under sampleo, lo cual baja la cantidad de datos para balancear la aparición de datos minoritarios y se vuelve a aplicar regresión obteniendo un mejor resultado en la detección de verdaderos positivos y verdaderos negativos. Esto además hace que disminuya significativamente la precisión, ya que ahora tenemos más errores en la detección de los 0 de la salida.

Contrario a lo anterior, se aplica un Sobre Sampleo y se aplica regresión, donde obtenemos resultado similares al under sampling.

Otra de las pruebas que se realizó, es utilizar regresión logística con el parámetro **class_weight** en **balanced** donde la tarea que realiza es el balance de datos y vemos que los resultados son similares a la aplicación de los dos métodos anteriores.

Por último, se decide realizar una regresión logística aplicando el método antes analizado de **QuantileTransform** donde obtenemos valores mejorados que los métodos anteriores aunque teniendo en cuenta la precisión que se mantiene en valores bajos.

Decision Tree

Se implementa la función de árbol de decisión, donde podemos obtener métricas mejoradas con respecto a la aplicación de regresión lineal pero con la desventaja que el algoritmo tiene un tiempo de procesamiento mucho mayor a la aplicación de regresión.

Random Forest

Con la implementación del algoritmo de random forest, logramos mejorar en pequeñas cantidades al algoritmo de Decision Tree, también con la desventaja del tiempo de procesamiento de los datos.

PyCaret

Por último, se utilizó PyCaret para verificar que el análisis anterior sea correcto en cuanto a los valores obtenidos. Además, se pudo ver que hay muchos algoritmos que se implementan en esta librería que no se pudieron probar ya que en las simulaciones nunca llegan a converger por lo que se decidió quitarlos y solo quedarse con aquellos que si lo hacen.

Vemos que en primer lugar nos recomienda utilizar RandomForest, luego DecisionTree y en quinto lugar regresión logística lo cual coincide con el análisis realizado previamente.

Conclusión

Mediante este trabajo practico se pudieron aplicar los conceptos vistos durante las clases de la materia. Se pudieron utilizar las diferentes funciones para realizar el análisis de los datos y poder prepararlos para la aplicación de los algoritmos de ML.

En cuanto a la aplicación de los diferentes algoritmos, se pudo apreciar la gran diferencia en procesamiento en la aplicación de regresión versus random forest y decision tree, pero a su vez también se pudo apreciar el mejoramiento en las métricas.