Informe Análisis de Fraude

10.12.2020

—

Isaí Piña Torres

Analista de datos

Tabla de contenido

Descripción General 2

Objetivos 4

Metodología 4

Selección de variables 5

Limpieza de datos 6

Preprocesamiento de datos 7

Balanceo de datos 8

Matriz de Correlación 10

Análisis exploratorio de datos 11

Entrenamiento e implementación del modelo 12

Evaluación del modelo 13

Conclusiones y sugerencias 16

Descripción General

La empresa XX, es una empresa dedicada a la venta de productos agrícolas 100% online. Durante la pandemia se han aumentado significativamente las ventas por medio de su página web, generando también un incremento elevado en intentos de fraude al momento de comprar los productos. Por tal motivo se hace necesario contar con un modelo de machine Learning que permita basado en los datos históricos de transacciones, identificar de manera inmediata el nivel de fraude que pueda tener una transacción y ser rechazada o aprobada.

Datos

El archivo transacciones.xls contiene más de 500.000 registros históricos con las siguientes

columnas:

- Country: País de compra
- Order id: Código de la orden
- Creation date: Fecha y hora de creación de la orden
- Fecha: Fecha de realización de la orden
- Mes: número del mes de la orden
- Merchant: Código del sitio de venta
- Account Id: Código de la cuenta de cliente
- Reference: Referencia de la venta
- Description: Respuesta de la venta
- Transaction type: Tipo de transacción
- Status: Estado de la transacción
- Response code: Código de respuesta de la transacción
- Bank Response: Código de respuesta del banco
- Message Response Error: Mensaje de respuesta del Banco

- Payment Method: Método de pago
- Authorization code: Código de Autorización
- Payment Model: Modelo Bancario del Pago
- Transaction Origin: Método de origen del envío de la transacción
- Accreditation model: Modelo de Acreditación por el banco
- Days to deposit: Número de días en que se ve reflejado un depósito
- BIN Bank: Nombre del Banco de la compra
- Country Bank: País del banco de la compra
- Visible number: Número de la tarjeta de pago
- Card type: Tipo de tarjeta
- Transaction currency: Moneda en que se hizo el pago
- Valor: Valor de la compra
- Processing currency: Moneda en la que se procesa el pago
- Processing value: Valor final de la compra en la moneda de procesamiento del pago
- Franchise: Franquicia del pago.

Objetivo:

Implementar una solución basada en machine Learning que aprueba las transacciones legítimas y rechace las transacciones fraudulentas de manera automática, a partir de los datos históricos de las transacciones.

Metodología

Para abordar este problema creamos un flujo de trabajo de datos de la siguiente manera:

- **Configuración y carga de datos**: Cargamos la base de datos de registros de transacciones al notebook.
- **Limpieza y preprocesamiento de datos:** buscamos la existencia de valores nulos, consistencia en el tipo de datos y el valor de los datos de cada columna, eliminación de columnas irrelevantes y transformación de los datos de tipo objeto a tipo entero y de tipo categórico a tipo entero.
- **Entrenamiento del modelo:** Se seleccionaron dos modelos de clasificación para este estudio: Random Forest Classifier y Tree Decision Classifier. Se dividió el dataset en tres subsets: entrenamiento (80%), validación (10%) y prueba (10%). Una vez entrenado los modelos, se comprobaron las métricas relacionadas a sus resultados, 'Accuracy' de cada subset, Recall del Modelo, Precisión del Modelo, F1-Score del Modelo, el Área bajo la curva (AUC), su respectiva matriz de confusión, y las curvas ROC de cada modelo empleado.

El estudio se realizó mayoritariamente en Jupyter Notebook usando Python y sus librerías como Kernel central en un entorno local. Además de ello, se usó Power BI para la exploración de datos y realización de gráficas, Excel para la visualización de los valores de datos y de sus categorías, PowerPoint para la comunicación de los resultados y Word junto a Google Docs para la realización de este informe. El notebook junto con los datasets originales y producidos por el estudio, serán subidos a Github para incorporarse como parte de un repositorio privado, sin usar el nombre de Truora en ningún lugar.

Se realizó el estudio en una computadora de escritorio Hp modelo 2010 con Intel Core 2 Duo E8400 con 3.00GHz de velocidad de procesador, 4GB de RAM, gráficos generados por Intel G41 Express Chipset y 298GB de almacenamiento en Disco.

Selección de variables

Como variable objetivo, se escogió la variable 'Responde code' debido a que especificaba la razón de que una transacción fuese rechazada o aprobada y cuantas de esas transacciones eran rechazadas por el software anti-fraude. Se consideraron solo los valores de 'APPROVED' y 'ANTIFRAUD REJECTED' no solo porque son las categorías a las cuales deseamos que el algoritmo clasifique los nuevos datos de transacciones, sino también debido a que la gran mayoría de los datos se dividían en estos dos valores. Para entrenar el modelo, se cambiaron estos valores categóricos de 'APPROVED' y 'ANTIFRAUD REJECTED' por 0 y 1, respectivamente.

Como predictores, se escogieron las siguientes variables:

	Variables	Número de registros	Tipo de datos
1	hour	409250 non-null	int64
2	day	409250 non-null	int64
3	month	409250 non-null	int64
4	year	409250 non-null	int64
5	Order_Id	409250 non-null	int64
6	Account_Id	409250 non-null	int64
7	Valor	409250 non-null	int64
8	Transaction_type	409250 non-null	int32
9	Payment_method	409250 non-null	int32
10	Transaction_origin	409250 non-null	int32
11	BIN_Bank	409250 non-null	int32
12	Country_BIN_ISO	409250 non-null	int32
13	Card_type	409250 non-null	int32
14	Visible_number	409250 non-null	int32

Para seleccionarlas estas variables como predictores tuvimos en cuenta que cumpliesen 2 requisitos:

1. No fuesen variables eliminables en la parte de limpieza de datos. (Esto se explicará mejor más adelante).
2. Existiese algo de asociación entre cada una de ellas y la variable objetivo durante el análisis exploratorio de datos.

NOTA: Las columnas de 'hour', 'year', 'month' y 'day' fueron una segregación de la columna 'Creation Date'.

Limpieza de datos

Los siguientes pasos fueron los utilizados para la limpieza de los datos:

1. Una vez se observaron las características del Dataset, se borraron las columnas que no son relevantes para nuestro análisis por las siguientes razones:

- Country: Todos los valores son Colombia

- Days to deposit : La mayoría de valores son nulos y tiene 14 cuyos valores son 0.0.

- Description : 'XYZ agradece su compra' tiene la mayoría de los valores, lo cual no parece ser decisivo para determinar si una transacción es fraudulenta o no.

- Transaction currency : Todos los valores son "Cop", por tanto, ya sabemos que las transacciones se realizan en pesos colombianos y no necesitamos tenerla presente.

- Accreditation model: Todos los valores son "Intermediate".

- Merchant Id: Es el mismo sitio de venta.

- Bank Response Code: Esta columna está llena de datos y valores aleatorios que muy poco brindan información sobre la transacción.

- Message response error: La mayoría de los valores de esta columna están vacíos.

- Payment Model: La mayoría de valores son 'Gateway'.

- Reference: La columna de referencia tiene una alta cardinalidad y puede evitar que el modelo se desarrolle de manera correcta.

- Status: Para nuestro propósito, repite la misma función que la variable target: Response code, siendo esta última más específica para lo que deseamos predecir.

- Authorization Code: La ausencia de código de autorización, nos demuestra que la transacción fue detectada fraudulenta. Esa misma función ya la cumple la variable target, además que esta variable solo es una consecuencia de la decisión tomada para permitir o no la transacción, no la causa de la decisión, por lo que se deja afuera del modelo.

- Processing currency: Sabemos que la mayoría de las columnas son 'COP' a excepción de unas cuantas donde hay valores nulos.

- Processing value: Tiene los mismos valores que la columna de valor, pero en formato float y, además, le faltan valores.

- dif_val: Se creó esta columna como una forma de ver si 'valor' y 'Processing value' compartían los mismos datos. Una vez se comprobó que sí, se eliminó esta columna.

- Franchise: ya existen estos datos en la columna 'Payment method'.

2. Posteriormente, se borraron todos los registros que no tuviesen los valores 'APPROVED' y 'ANTIFRAUD REJECTED' en la columna de 'Response code'.

3. Se borraron todos los valores nulos del dataset, una vez se hubo comprobado que era la mejor forma de tratar con los valores nulos, puesto que no eran muchos registros y la mayoría eran nulos en casi todas las columnas.

Preprocesamiento de datos

Como parte del preprocesamiento de datos se realizaron las siguientes acciones:

1. Se renombraron las columnas restantes del dataset, eliminando cualquier espacio en blanco en sus nombres, para evitar un problema al colocarlos dentro del modelo. Algunos modelos son sensibles a nombre de categorías con espacios en blanco, por lo que se sustituyó por un guión bajo.

2. Se cambió el tipo de dato de la columna 'Creation Date', la cual pasó de tipo objeto a tipo datetime.

3. Se crearon las columnas 'hour', 'day', 'month' y 'year' a partir de la columna 'Creation Date' y se borró esta última.

4. Se transformaron las variables categóricas a través de 'Labelencoder', asignando así, un número entero a cada valor de cada variable categórica. De esta manera, se pudo hacer

una matriz de correlación y se evitó problemas de tipo de datos al incluir el dataset a los modelos.

5. Se renombró la variable objetivo 'Response code' por 'Target'

Balanceo de datos

La diferencia entre las clases de la variable objetivo es del 39%. Es decir, hay más registros (81076, para ser exactos) en la clase de transacciones legítimas (0) que en la clase de transacciones rechazadas como fraudulentas (1).

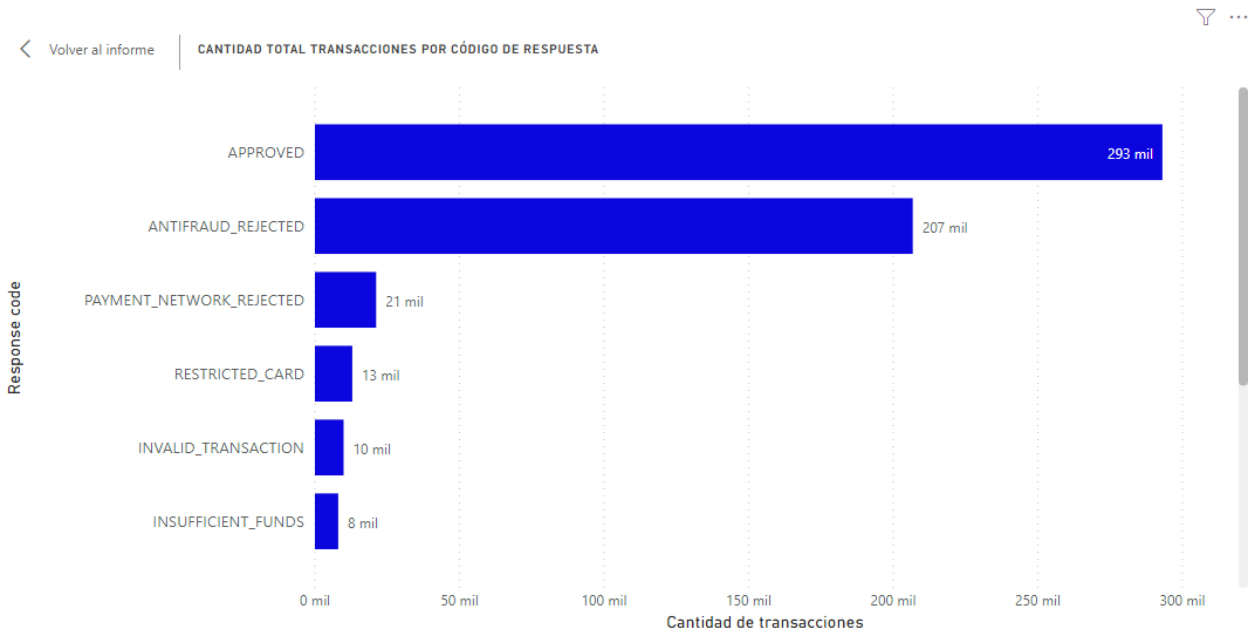

Figura 1. Cantidad de transacciones por legitimidad. Fuente: Datos de Truora 2020.

Por lo tanto, se hizo el balanceo de esta variable a partir de **Undersampling**, que es el proceso en el que aleatoriamente se eliminan algunas observaciones de la clase que tiene mayores registros, con el fin de igualar los registros de ambas clases. De esta manera, ambas clases quedaron con 204625 registros.

Hay dos razones principales por las que se toma muy en cuenta la práctica de balanceo de datos:

1. Se quiere evitar que el modelo entrenado, prediga valores basado en un sesgo de entrenamiento y se incline más por una clase que por otra al tener que predecir.
2. Se aclaren correlaciones entre variables que podían verse oscurecidas por un dataset desproporcionado.

En nuestro caso, se hizo más por la primera razón que por la segunda, porque como veremos en la próxima sección, la diferencia entre las matrices de correlación de las variables del dataset desbalanceado y normalizado es casi nula.

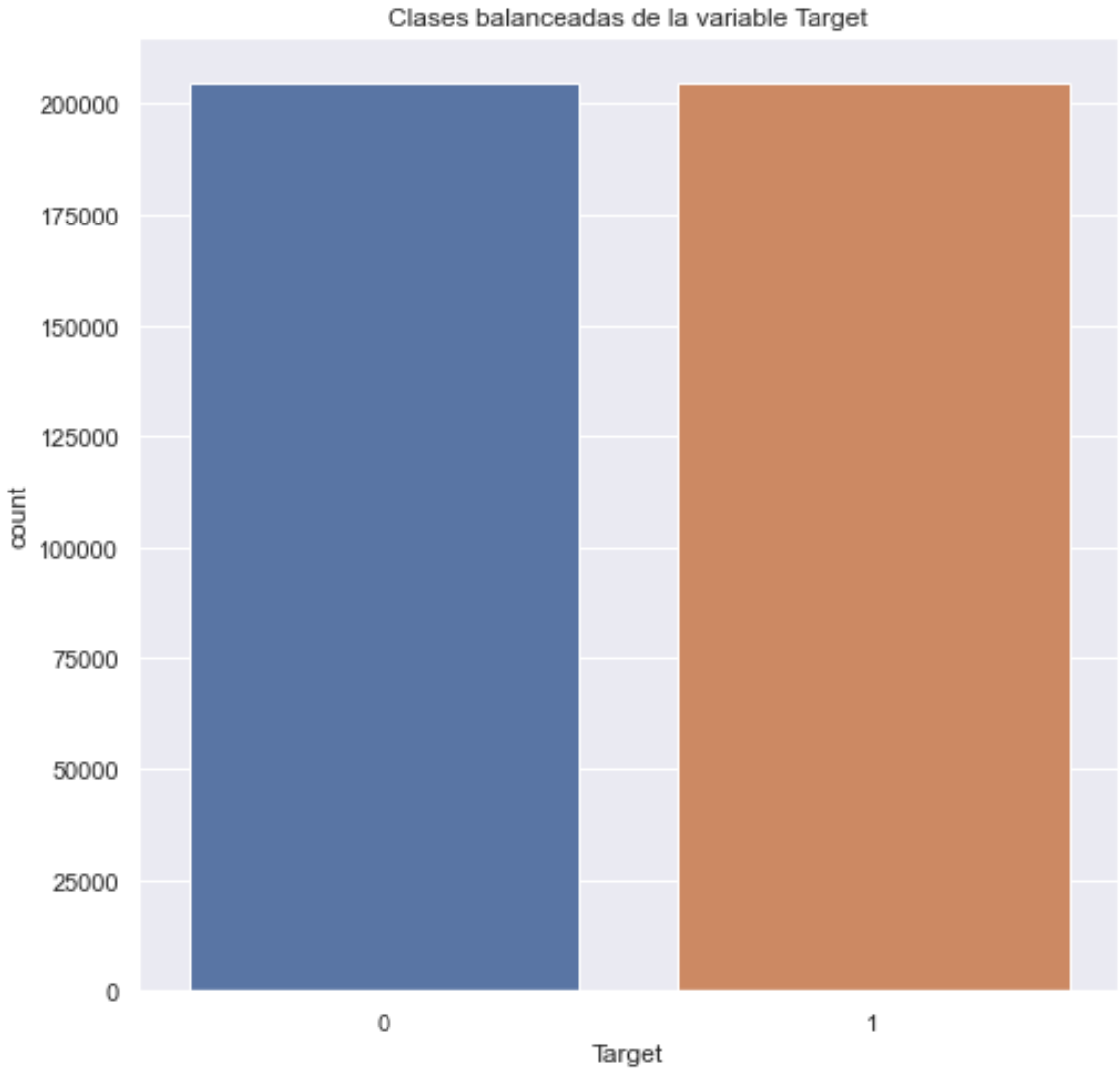

Figura 2. Cantidad de transacciones por legitimidad del dataset normalizado. Fuente: Datos de Truora 2020.

Matriz de Correlación

Las matrices de correlación son utilizadas para conocer asociaciones entre las variables que componen un dataset.

A continuación, las gráficas de las matrices de correlación del dataset desbalanceado y normalizado, respectivamente:

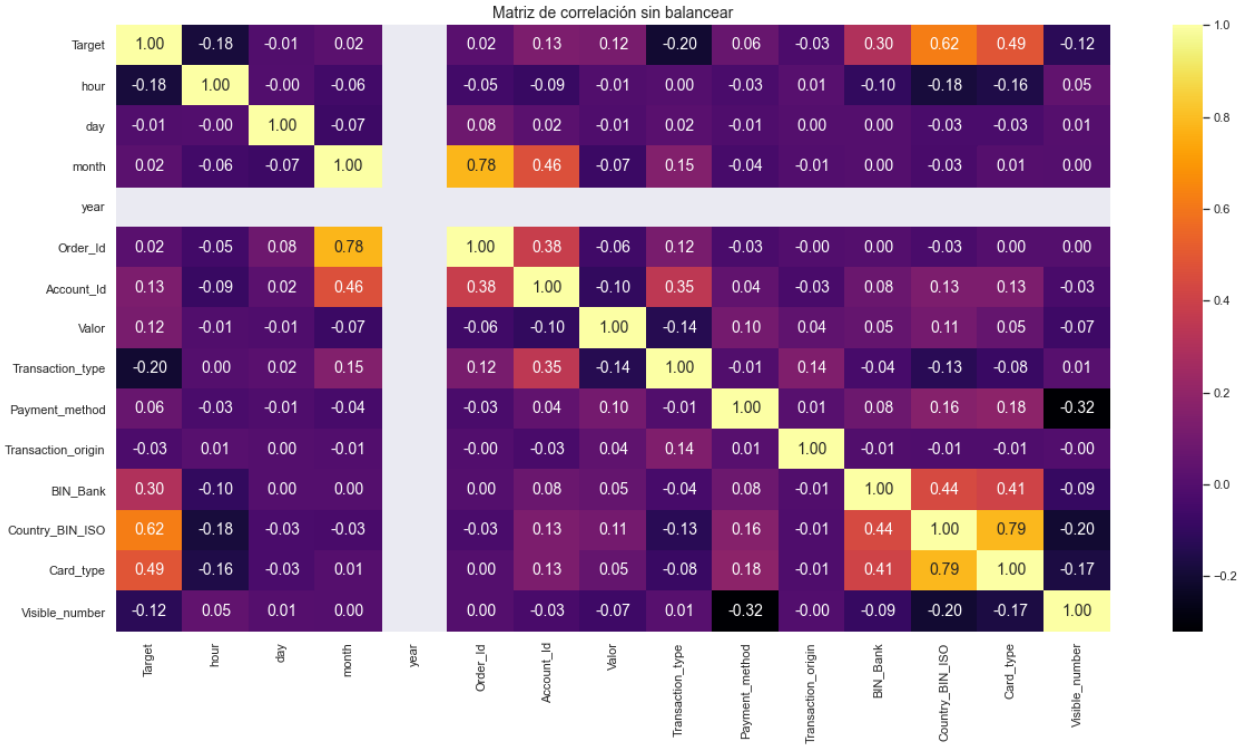

Figura 3. Matriz de correlación del dataset desbalanceado. Fuente: Datos de Truora 2020.

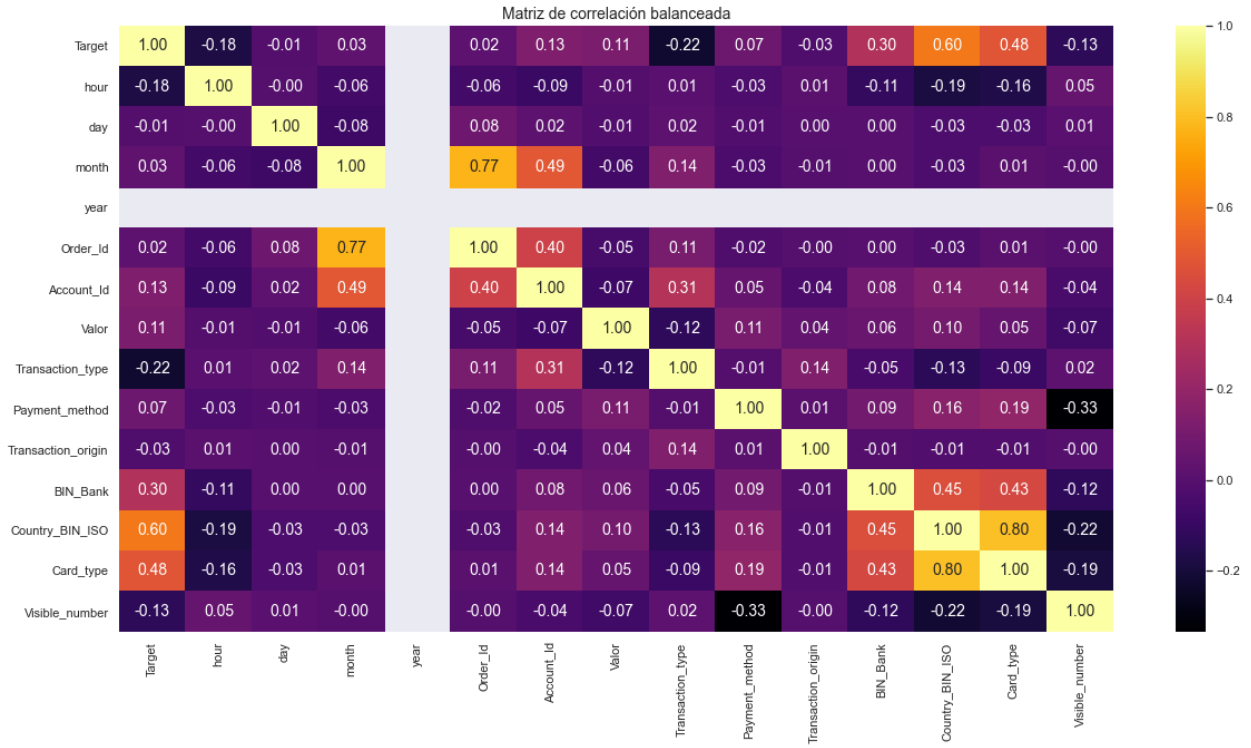

Figura 4. Matriz de correlación del dataset normalizado. Fuente: Datos de Truora 2020.

Observamos que las columnas 'Bin Bank' (Nombre del banco en el que se hizo la transacción), 'Contry_Bink_ISO' (País del banco) y 'Card Type' (Si es débito o crédito) tienen una alta correlación con las transacciones fraudulentas. Así mismo, existe una ligera asociación negativa de 'Transaction_type' y 'hour' con respecto a las transacciones de fraude. Esto podría indicar que estas variables se relacionan con las transacciones legítimas.

Como recalcamos anteriormente, la diferencia entre los datasets en la correlación de variables fue casi inexistente, pero se justifica el balanceo del dataset para evitar problemas en la implementación del modelo.

Análisis exploratorio de datos

Se realizó el análisis exploratorio de datos en el Software Power BI, por lo que esta sección no se verá reflejada en el notebook adjunto a este informe. Sin embargo, se adjuntará un reporte de este análisis en formato PDF exportado desde Power BI para su apreciación.

Entrenamiento del modelo

Para este estudio se escogieron dos algoritmos de aprendizaje automático:

1. Decision tree Classifier model
2. Random Forest Classifier model

La razón por la cual se escogieron fue por su simplicidad en términos de comprensión de los conceptos asociados a estos modelos, aparte de su vasto uso y popularidad en la comunidad. Pero esto no quiere que decir que carezcan de robustez. Ambos son modelos computacionalmente poderosos.

Para ambos clasificadores se dividió el dataset en 2 partes: entrenamiento (80%), y test (20%). No usaremos un dataset de validación debido a que validaremos el modelo a partir de validación cruzada de K-folds. Recordemos que la validación cruzada de K-folds consiste en dividir el conjunto de datos del entrenamiento en k-folds (en nuestro caso, escogeremos 3), y entonces hacer predicciones y evaluarlas en cada fold usando un modelo entrenado en los folds restantes (NO en el mismo fold). Por lo tanto, usaremos la función cross_val_score() de scikit-learn para evaluar ambos modelos con 3 folds.

Desde esta perspectiva, se podría pensar que se hace innecesario evaluar el modelo con los test datasets creados. Sin embargo, la metodología usada en este estudio, fue analizar y evaluar las predicciones de los modelos en cada fold y al final, se usó el dataset de prueba para contrastarlo con lo obtenido en cada fold.

Los detalles específicos de cada secuencia de código usada para el uso de este clasificador, se puede observar en el notebook que vendrá adjunto a este informe.

En resumen:

1. Dividimos el dataset en 80% de los registros como dataset de entrenamiento y 20% como dataset de prueba.
2. Entrenamos el modelo con los datos de entrenamiento.
3. Usamos cross_val_score() para crear una validación cruzada de 3 folds en los que se tendrán 3 valores de ‘accuracy’. En realidad, estamos realizando tres validaciones del modelo simultáneamente.
4. Usamos cross_val_predict() para predecir los valores de la variable objetivo en cada fold.
5. Mostramos la matriz de confusión.
6. Usamos la función classification_report() para imprimir el resto de métricas del modelo.
7. Una vez hecho esto, se repitieron los pasos del 3 al 6 con el dataset de prueba.
8. Se imprimieron las curvas ROC y el área bajo la curva (AUC) de cada clasificador.

Evaluación de los modelos

La evaluación de un modelo de clasificación es un poco más compleja que con un modelo de regresión. Sin embargo, ambos utilizan las mismas métricas.

Para el caso de Decision Tree, el accuracy del set de prueba mejoró respecto a los folds que validaban su accuracy:

```
Test set = 0.891
 Folds = ([0.8872487, 0.88657876, 0.88772415])
```

La matriz de confusión del set de prueba también muestra mejorías:

```
Matriz de confusión normalizada del set de prueba:
[0.8815597 0.1184403]
[0.09901572 0.90098428]
```

Sin embargo, al momento de ver las métricas de clasificación, son iguales que a las del fold, lo que nos dice que es algo que puede ser estable e independiente del dataset, lo cual es bueno.

```
Métricas de Clasificación para el set de pruebas:
              precision    recall   f1-score   support

           0       0.90      0.88      0.89     41008
           1       0.88      0.90      0.89     40842
```

Lo destacable es que su recall es mejor que su precisión, lo cual para el negocio de los bancos significa que hay menos falsos negativos (transacción que dejan pasar como legítimas pero que en realidad son fraudulentas.

Por otro lado, en el caso del modelo de RandomForest, observamos que hay una ventaja bastante marcada respecto a Decision Tree:

```
Matriz de confusión normalizada:
[0.94064573 0.05935427]
[0.0995299 0.9004701]
ACCURACY OF THE MODEL:  0.9205986560781918
```

La desventaja que puede ver aquí es que la métrica de precisión está más alta que el recall.

Sin embargo, en los valores altos en los que se encuentran, no debería existir ningún inconveniente.

```
Classification metrics:
              precision    recall  f1-score   support

           0       0.90      0.94      0.92     41008
           1       0.94      0.90      0.92     40842
```

Finalmente, con las curvas ROC podemos confirmar lo que ya hemos hablado con solo darles un vistazo.

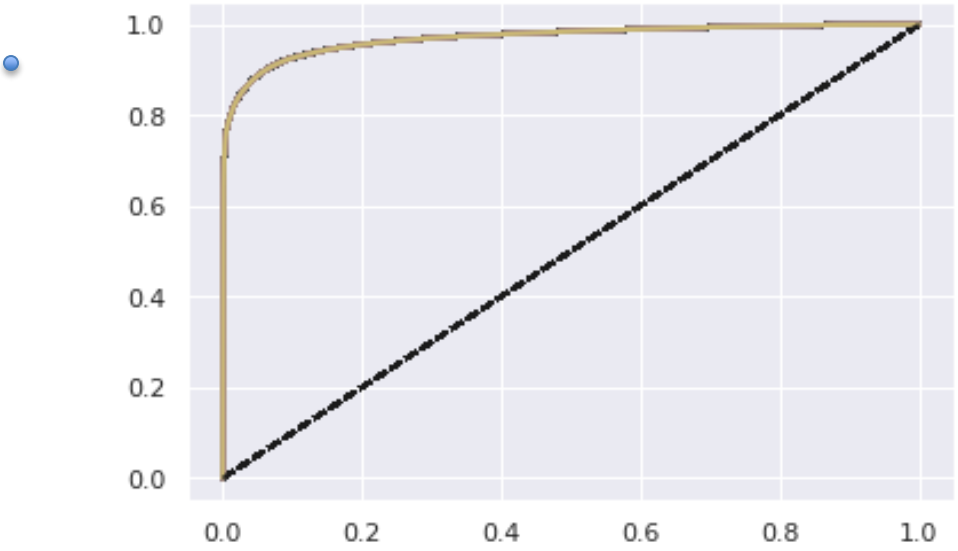

Figura 5. Curva ROC del modelo RandomForestsClassifier. Fuente: Datos de Truora 2020.

El resultado ROC -AUC para este modelo es `0.9700210457841703`

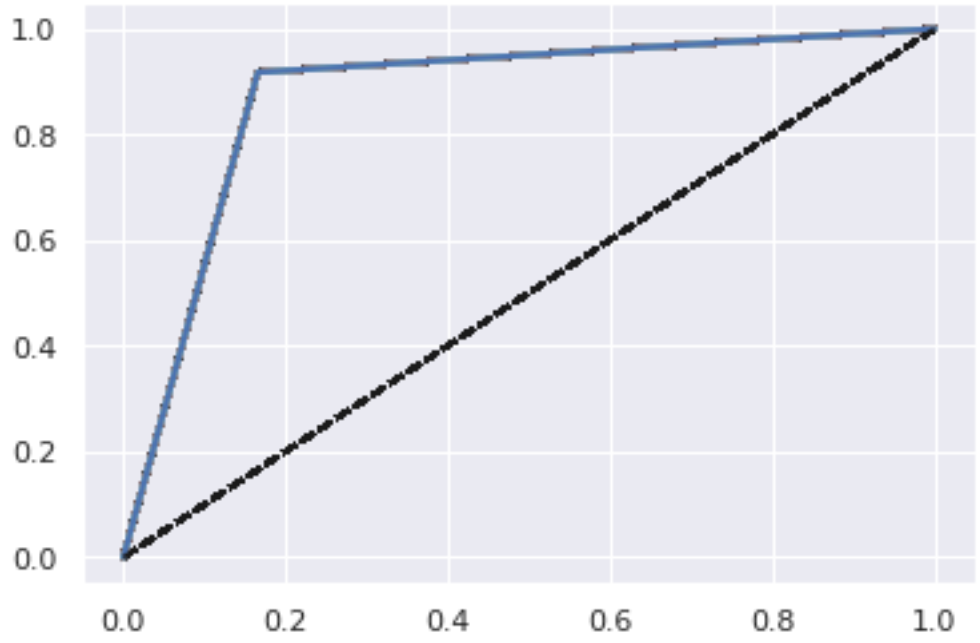

Figura 6. Curva ROC del modelo DecisionTreeClassifier. Fuente: Datos de Truora 2020.

Conclusiones y sugerencias

Analizando las métricas de ambos modelos podemos saber qué modelo se ajusta mejor que otro. Tenemos a `RandomForestClassifier` como claro ganador, aunque no por mucha ventaja.

No solo su accuracy es superior, sino también le lleva ventaja en todas las demás métricas, incluyendo el área bajo la curva AUC y la curva ROC de cada modelo.

Recordemos que la razón por la que no debemos basarnos solo en el accuracy para evaluar un modelo, es debido a que es muy susceptible a las distribuciones de datos sesgadas. Es muy probable, que el accuracy de `RandomForestClassifier` descienda significativamente, si el training set o el test set tienen más registros de una categoría, que de otra. Para esto, se puede sugerir utilizar el método de `BalancedBaggingClassifier` de la librería `imblearn`, el cual balancea automáticamente cada subset del conjunto de datos durante la implementación de cada modelo.

Es por eso, que precision y recall, se convierte en herramientas muy útiles al momento de discernir lo que sucede con el modelo. En nuestro modelo de detección de transacciones fraudulentas, conviene estar más pendiente del recall que del precision. Esto, por la misma naturaleza de la métrica, la cual, solo se enfoca en Positivos. Recall, por su parte, toma en cuenta los falsos negativos (las transacciones que el modelo considero legítimas pero que en realidad no lo son). Por lo que, un alto valor de Recall, nos da un buen alivio en cuanto a seguridad de las transacciones se refiere.

En el caso específico de este análisis, las curvas ROC confirman lo que se ha dicho. En el caso de `RandomForestClassifier` la separación de su curva con el eje y, se encuentra en una tasa de True Positives bastante elevada, y el área bajo la curva tan cercana a 1, nos da fiabilidad de ser un buen modelo. Nuevamente, la oportunidad de mejora de ambos modelos está en la eliminación de cualquier subset desbalanceado.