**Diplomado de Ciencia de Datos**

**Introducción:**

Este proyecto se basa en un dataset de la plataforma Kaggle que proporciona un conjunto de datos para una aplicación de pagos móviles con 1,048,575 registros. Incluye información sobre el remitente y destinatario de una transacción, así como si las transacciones están marcadas como fraude o no fraude. El análisis de esta información es crucial para la prevención y detección de fraudes.

Este trabajo forma parte del proyecto de investigación "Sistemas escalables y eficientes en recursos para análisis de big data", financiado por la Fundación del Conocimiento (subvención: 20140032) en Suecia.

El dataset incluye información relevante acerca de:

- Tipo de transacción

- Cantidades transaccionadas

- Indicación de si es una operación fraudulenta

Además, muestra los saldos de las cuentas y cómo pueden influir en la detección de fraudes.

**Objetivos y Tareas:**

- Cargar los datos en un modelo de base de datos relacional.

- Realizar actividades de preprocesamiento de los datos para obtener un conjunto de datos limpio y bien estructurado, facilitando tareas de predicción y clasificación.

- Realizar un análisis exploratorio de datos para obtener metricas sobre los datos recopilados.

- Utilizar el dataset para desarrollar un análisis con Python que permita realizar predicciones, clasificaciones y otras manipulaciones de datos.

- Basándonos en el análisis, decidir qué modelo de machine learning es el más adecuado para su manipulación.

-Establecer conclusiones con los resultados obtenidos a lo largo del análisis.

**Análisis exploratorio de datos:**

Para la selección de la base de datos, nos planteamos las siguientes preguntas:

- ¿Puede el machine learning (ML) ser una solución para el problema?

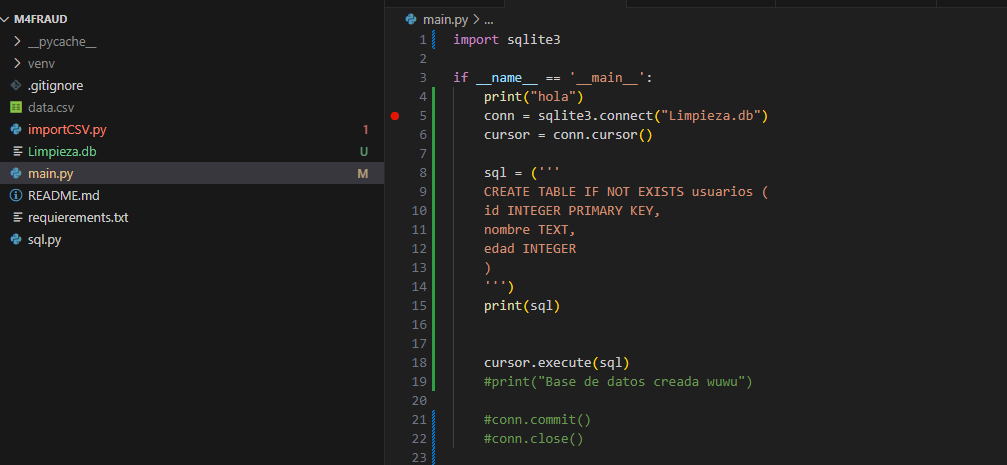
- ¿Tenemos todos los datos necesarios?

- ¿Hay suficientes datos para desarrollar un algoritmo?

- ¿Los datos se han recopilado de manera adecuada?

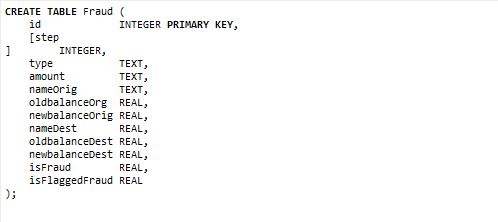
- ¿Los datos se han almacenado en el formato correcto?

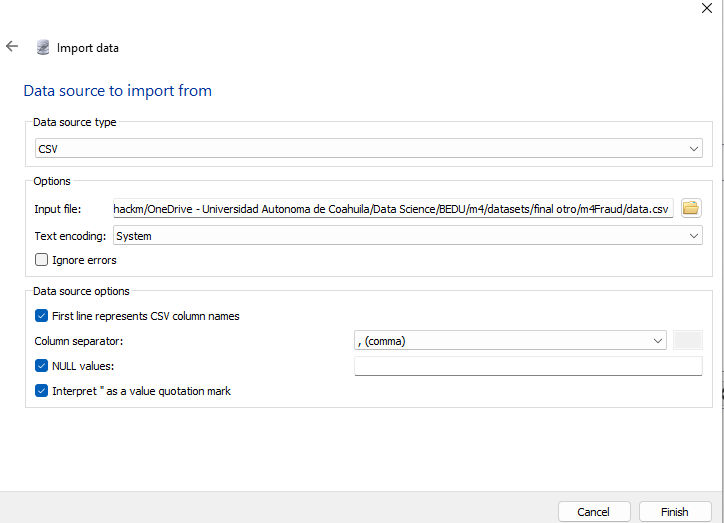
- ¿Se garantiza el acceso a la información?

Utilizando las tecnologías python y sqlite se implementa la creación de la base de datos como se muestra enseguida:

Para confirmar dicha creación se utiliza el archivo main one ejecutamos el programa haciendo la conexión a la base de datos y creamos una tabla de prueba para verificar que la conexión fue exitosa como se muestra en la captura anterior.

Seguido a ello ingresamos a sqlite y creamos la base de acorde a el archivo ‘data’ que estaremos limpiando:

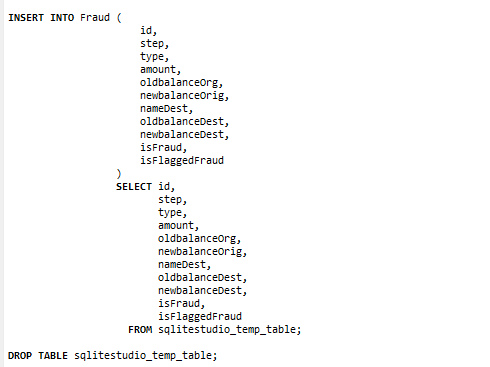


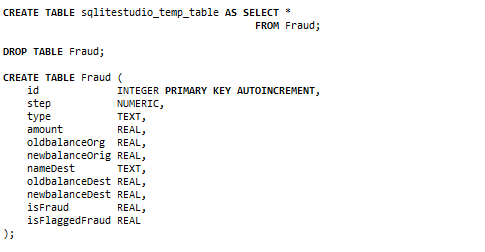
Una vez creada la tabla cargamos el archivo especificando la ruta en donde se encuentra el csv

De forma exitosa el archivo es cargado rellenando así todos los campos que se encuentran dentro del csv

**Limpieza de datos**

En sqlite no se tiene una sentencia para eliminar una columna, no existe el famoso drop, por lo que es necesario copiar los campos de la tabla omitiendo la columna que se necesita borrar como se muestra enseguida:



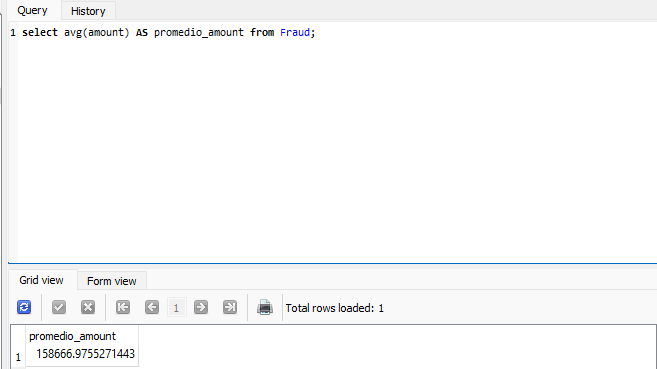


Se borra la columna nameOrigin ya que se considera innecesaria para el tratado de datos debido a que solo son claves genéricas que sirven como identificadores por sí mismas en cada movimiento.

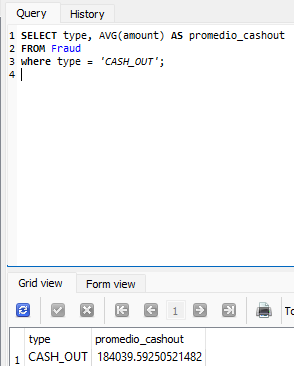
**Reporte de datos**

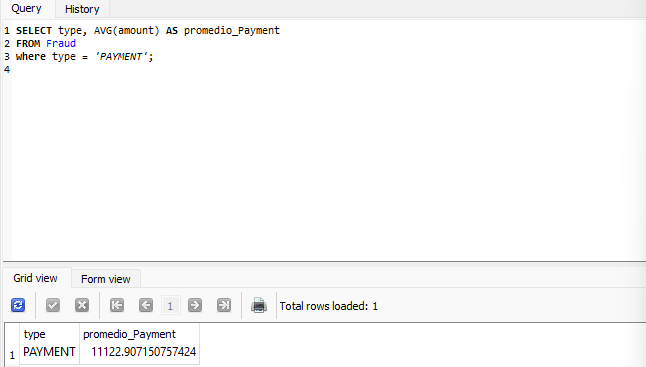
Ahora que se tiene la base de datos limpia buscaremos trabajar con la información que se tiene.

1. Obtendremos el promedio de las cantidades que se está trabajando en la tabla Fraud:



1. Buscaremos el promedio de las personas que tienen como tipo de transacción ‘cashout’ en la columna ‘type’:



1. Buscaremos el promedio de las personas que tienen como tipo de transacción ‘payment’ en la columna ‘type’:

Al analizar el dataset, identificamos columnas que no aportaban información y que contenían datos alfanuméricos que actuaban como claves de productos. Estos datos entorpecían el procesamiento y afectaban al modelo. Por tanto, eliminamos estas columnas marcadas como nombre de origen y destino. También eliminamos la columna "Step", que resultaba completamente irrelevante para nuestro análisis. Además, notamos que los datos de la última columna "Is Fraud" eran de tipo booleano, lo cual consideramos fundamental para el modelo de machine learning más adecuado.

Primero, verificamos que no existían valores nulos en este dataset. Posteriormente, codificamos la columna "TYPE", que contenía los valores "PAYMENT", "TRANSFER" y "CASH\_OUT", convirtiéndolos en valores numéricos. Reorganizamos las columnas para un mejor funcionamiento del dataset y normalizamos la escala de los datos utilizando el comando "MinMaxScaler". También dividimos los datos en conjuntos de entrenamiento y prueba.

Estas métricas nos ayudarán a comprender la eficacia del modelo en la detección de transacciones fraudulentas. El objetivo del modelo de aprendizaje automático es predecir si una transacción dada es un fraude real o no. El modelo clasifica las transacciones en dos categorías: fraude (clase positiva) y no fraude (clase negativa).

Para evaluar el rendimiento del modelo, utilizaremos las siguientes métricas:

1. Puntuación F-beta: Combina precisión y recall, permitiendo ponderar más precisión o recall según se necesite.

2. Puntuación AUC-ROC (Área Bajo la Curva de Característica Operativa del Receptor):Evalúa la capacidad del modelo para distinguir entre las clases positiva y negativa.

3. Recall (Recuperación): Mide cuántas observaciones positivas se han clasificado correctamente, es decir, la tasa de verdaderos positivos.

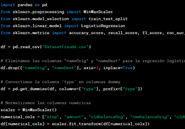
4. Precisión Promedio: Resume la curva de precisión-recall, ofreciendo una evaluación completa del rendimiento del modelo en la detección de fraudes.

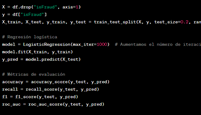
Estas métricas nos ayudarán a evaluar qué tan bien el modelo de aprendizaje automático identifica transacciones fraudulentas y cómo equilibra la precisión y el recall al hacerlo.

**Modelo de Regresión Logística:**

A pesar del desequilibrio en los datos, se realizó un análisis utilizando el modelo de regresión logística. Los resultados mostraron una eficiencia baja, probablemente debido al desequilibrio en la columna "isFraud". Aunque inicialmente parecía menos adecuado, decidimos evaluarlo para obtener insights.

El código utilizado fue el siguiente:







Inicialmente, consideramos que la regresión logística no sería una opción viable debido a la naturaleza booleana de la columna que indica si es fraude o no. Sin embargo, decidimos llevarla a cabo para evaluar los resultados obtenidos. Se realizaron pruebas y, en una selección al azar, obtuvimos un rendimiento de 0.54.

Los resultados de la prueba de Regresión Logística fueron los siguientes:

- Exactitud (Accuracy): 1.00

- Recall: 0.08

- F1-Score: 0.15

- AUC-ROC: 0.54

Estos resultados indican una eficiencia notablemente baja del modelo, debido al abrumador sesgo hacia operaciones sospechosas no fraudulentas en la columna "isFraud" (99.89%), en comparación con las transacciones detectadas como fraude (0.1089%). Este sesgo afecta negativamente el rendimiento del modelo.

Dado que tenemos un conjunto de datos desequilibrado, las Curvas Características de Operación del Receptor (AUC-ROC) no son tan informativas en este caso, ya que se puede obtener una curva aparentemente buena simplemente adivinando que cada caso no es fraude.

Puntuación de precisión del modelo: La forma más sencilla de analizar el rendimiento es a través de la precisión, que mide cuántas observaciones, tanto positivas como negativas, fueron clasificadas correctamente. Sin embargo, en problemas desequilibrados, la precisión no es una métrica adecuada, ya que se puede obtener una puntuación alta simplemente clasificando todas las observaciones como la clase mayoritaria. Por ejemplo, en nuestro caso, al clasificar todas las transacciones como no fraudulentas, podríamos obtener una precisión de más del 0.9. Lo cual resulto revelador en el siguiente modelo.

**Modelo de Árboles de decisión.**

Al comenzar el modelo, esperábamos mejorar considerablemente los resultados obtenidos en el modelo de regresión. Por lo que decidimos comenzar a llevar a cabo el modelo con 100 árboles de decisión para evaluar la capacidad de cómputo y los resultados, ya que como vimos en clase se considera el min numero para obtener resultados confiables.







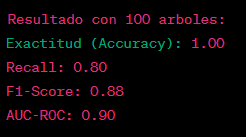
Resultado con 100 arboles.

Exactitud (Accuracy): 1.00

Recall: 0.80

F1-Score: 0.88

AUC-ROC: 0.90



Como podemos ver los resultados fueron diametralmente opuestos al modelo de regresión, lo cuál nos indica que es un modelo confiable para atacar este tipo de dataset. A pesar de que obtuvimos una exactitud de 1 ya que se trata de una serie de datos académicos que no tiene un comportamiento real podemos ver que el modelo obtuvo una evaluación AUC-ROC del 0.90 Por lo que podemos inferir que este sería el modelo ideal a utilizar para atacar este dataset y poder definir si los patrones que tiene el dataset pueden determinar si existen casos de Fraude y que comportamiento tienen.

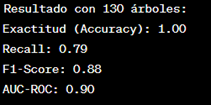
Posteriormente a modo de experimento realizamos una prueba con 130 árboles en donde obtuvimos el siguiente resultado:

Exactitud (Accuracy): 1.00

Recall: 0.79

F1-Score: 0.88

AUC-ROC: 0.90

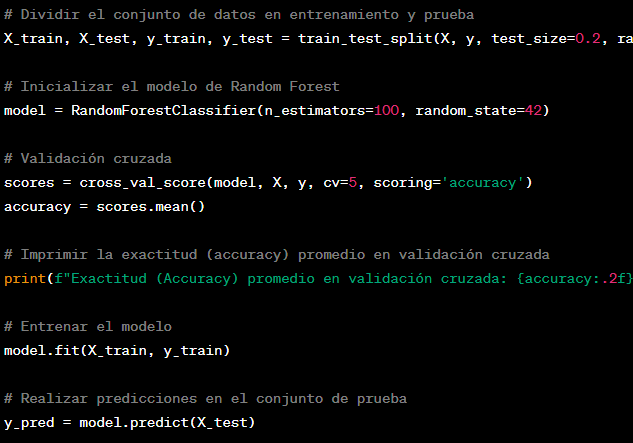


Como podemos ver el modelo no mejoro al realizarse con mayor cantidad de árboles de decisión, lo cual nos llamó la atención ya que esperábamos que mejorara, consideramos que tiene que ver el desequilibrio de los datos y es necesario referirnos a la gráfica de dispersión para ver qué tan separados están los puntos o si están muy están muy lejos del comportamiento sigmoidal.

**Validación cruzada**

Posteriormente procedimos a realizar la validación cruzada del modelo para corroborar el desempeño del mismo y verificar su rendimiento.





Se probaron distintas combinaciones de datos con los siguientes resultados:

1.

Validación cruzada(100 arboles):

* Exactitud (Accuracy) promedio en validación cruzada: 0.59
* Exactitud (Accuracy) en el conjunto de prueba: 1.00
* Recall: 0.35
* F1-Score: 0.49
* AUC-ROC: 0.68

2.

Validación cruzada(150 arboles):

* Exactitud (Accuracy) promedio en validación cruzada: 0.59
* Exactitud (Accuracy) en el conjunto de prueba: 1.00
* Recall: 0.36
* F1-Score: 0.50
* AUC-ROC: 0.68

3.

Con los siguientes hiperparámetros:

n\_estimators = 100

max\_depth = 20

min\_samples\_split = 2

min\_samples\_leaf = 2

* Exactitud (Accuracy) promedio en validación cruzada: 0.59
* Exactitud (Accuracy) en el conjunto de prueba: 1.00
* Recall: 0.29
* F1-Score: 0.45
* AUC-ROC: 0.65

4.

Con los siguientes hiperparámetros:

n\_estimators = 100

max\_depth = 20

min\_samples\_split = 5

min\_samples\_leaf = 5

* Exactitud (Accuracy) promedio en validación cruzada: 0.61
* Exactitud (Accuracy) en el conjunto de prueba: 1.00
* Recall: 0.26
* F1-Score: 0.40
* AUC-ROC: 0.63

5.

Con los siguientes hiperparámetros:

n\_estimators = 100

max\_depth = 30

min\_samples\_split = 2

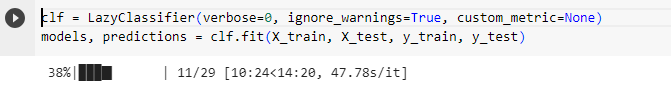
min\_samples\_leaf = 2

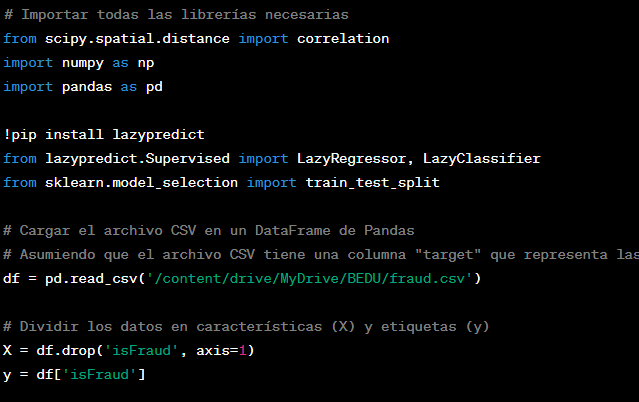
* Exactitud (Accuracy) promedio en validación cruzada: 0.59
* Exactitud (Accuracy) en el conjunto de prueba: 1.00
* Recall: 0.32
* F1-Score: 0.47
* AUC-ROC: 0.66

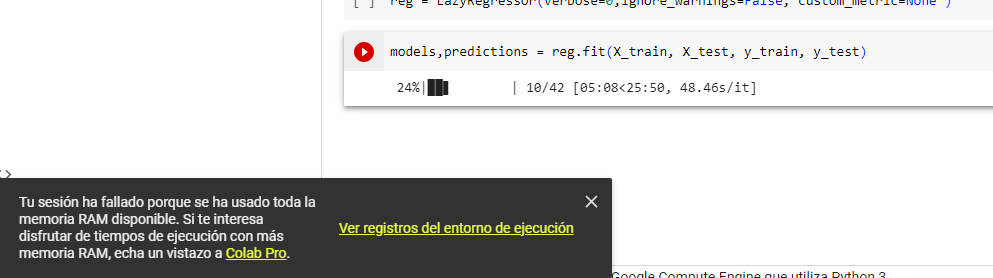
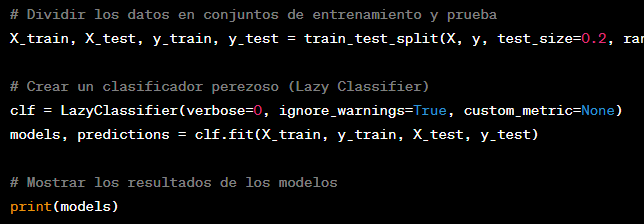
En general, los resultados sugieren que el modelo con la configuración actual no es muy efectivo para identificar los casos positivos (fraudes) en el conjunto de datos, ya que el recall es relativamente bajo en todas las configuraciones probadas. Podría ser beneficioso explorar diferentes valores de hiperparámetros o incluso probar otro modelo.

**Lazy predict**

Por último intentamos utilizar el algoritmo de lazy predict pero debido al tamaño del data set solo compilaba el 38% del mismo en el mejor de los casos antes de parar el proceso y me menciona que es por motivos de procesamiento de RAM. Notamos una dificultad a la hora del análisis del modelo de Parallel build trees, requería demasiada capacidad computacional con la cual no contabamos.







**Conclusiones:**

- La precisión es útil para evaluar el rendimiento del modelo, pero no en problemas desequilibrados.

- Es necesario abordar el desequilibrio en los datos para obtener resultados precisos y significativos en la detección de fraudes.

-Fue de vital importancia el preprocesamiento de los datos

-El modelo de Regresión logística, no es adecuado para este dataset ya que los datos están desequilibrados y el rendimiento es muy bajo.

-El rendimiento del modelo de Árboles de decisión es el mejor que encontramos y tiene un gran rendimiento.

-Añadir mas arboles al modelo de Árboles de decisión no mejora el modelo

-El tamaño del dataset dificulto y retardo considerablemente la manipulación y el procesamiento de datos

-Debido al desbalanceo de datos, y a la calificación de exactitud que obtuvimos podemos concluir que no es un data set confiable para predecir o modelar de madera adecuada el comportamiento de movimientos de Fraude en la vida real.

-Los resultados obtenidos con Validación cruzada sugieren que el modelo con la configuración actual no es muy efectivo para identificar los casos positivos en el conjunto de datos, ya que el recall es relativamente bajo en todas las configuraciones probadas. Podría ser beneficioso explorar diferentes valores de hiperparámetros o incluso probar otro modelo.