Modelo de identificación de Mala práctica transaccional

**### Objetivo**

\* El objetivo de la prueba es idear una solución para identificar transacciones que evidencian un

comportamiento de Mala Práctica Transaccional, empleando un producto de datos. Adicional,

describir la solución y detallar cómo incorporar el producto de datos en un marco operativo.

**### Datos relevantes**

\* Se entiende como una Mala Práctica Transaccional, un comportamiento donde se evidencia un

uso de los canales mal intencionado; como Fraccionamiento transaccional.

\* Fraccionamiento transaccional: esta es una mala práctica que consiste en fraccionar una transacción grande en varias pequeñas, estas transacciones se caracterizan por estar en una misma ventana de tiempo que suele ser 24 horas y tienen como origen o destino la misma cuenta o cliente.

**### Diccionario**

\* \_id: identificador único del registro

\* merchant\_id: código único del comercio o aliado.

\* subsidiary: código único de la sede o sucursal.

\* transaction\_date: fecha de transacción en el Core financiero.

\* account\_number: número único de cuenta.

\* user\_id: código único del usuario dueño de la cuenta desde donde se registran las transacciones.

\* transaction\_amount: monto de la transacción (en moneda ficticia).

\* transaction\_type: naturaleza de la transacción (crédito o débito).

Codigo:

Herramientas utilizadas: Pyspark, Pandas, matplotlib, seabor entre otrsas.

Se crea una función que instale y cargue las librerías que necesitamos si no están y si ya están simplemente que avise que ya están, luego se importan las librerías.

Se optimiza el cuadreno para que tenga mayor disponibilidad de la ram y utilice toda la ram que tenemos.

Se Inicia el EDA

Luego se crea un pipeline de descarga y lectura de los archivos fuentes de AWS de S3 que tome los datos desde la fuente y los cargue en el código. Si hay que unir los archivos se unen y se eliminan duplicados.

De aquí por capacidad de procesamiento se trabaja con una sola base, la cual contiene 6456562 datos y 1570006 de clientes unicos

Luego miramos con cuantos clientes únicos inicialmente estaríamos trabajando.

Arreglamos los posibles errores de las columnas en caso de que tengan y para ello creamos una función llamada corregir\_columna.

Vemos los tipos de variables que trae cada columna.

Eliminado duplicados y transformamos al formato que necesitamos algunas columnas y creamos columnas nuevas que vamos necesitando para el análisis.

Se hacen analisis descriptivos, búsqueda de correlaciones y patrones, visualización de outliers(boxplots, scatterplots, etc).

Posible selección de variables.

Como resultado se optiene conocimiento de la calidad de los datos, de las principales variables a usar en el modelo y que nos servirá para hacer el modelo.

Entrenamiento del modelo.

En esta parte utilizaremos herramientas como scikit-learn, PySpark, MLlib, DBScan, etc

En esta parte como tarea se tiene

* Dividir los datos (train/test o train/validation).
* Entrenar el modelo elegido (p. ej.: regresión, clasificación, clustering, etc.).
* Ajustar hiperparámetros (si procede).

Se obtendrá como resultado un modelo entrenado y validado, listo para predecir/inferir sobre nuevos datos. En este caso se utiliza DBSCAN para intentar clusterizar los clientes con esos valores atípicos que buscamos.

Notas: el EDA quedo semi automático en las conclusiones, eso quiere decir que si se cambia la data generara conclusiones distintas para ese tipo de data y queda fácil de realizar informe automático para que se descargue, se genere ppt y se envié a los distintos correos autorizados. .

Del Eda se pudieron obtener las siguientes conclusiones:

De la base que manejamos se tienen como:

* transaction\_type: DEBITO: 5155241 y CREDITO: 1301321
* El tiempo promedio general para esos clientes que realizan entre 2 y n transacciones es de 8.56 minutos
* El tiempo promedio general para aquellos que realizan desde una transacción en adelante es de 85.08 minutos.
* El tiempo promedio para aquellos que realizan mas de 2 transacciones de 43.64 minutos entre transacción pero para ver mejor y mas a detalle dividimos los montos entre mayores e iguales a 3000 dólares y menores a 3000 dólares.
* De los datos podemos ver que para las transacciones totales menores a 3000 dólares, el tiempo promedio entre transacciones es de 44.32 minutos. y que el 50% de estas transacciones se realizaron con una diferencia de tiempo entre ellas menor a 5.79 minutos.
* De los datos podemos ver que para las transacciones totales mayores e iguales a 3000 dólares, el tiempo promedio entre transacciones es de 13.67 minutos. y que el 50% de estas transacciones se realizaron con una diferencia de tiempo entre ellas menor a 1.47 minutos.
* Vemos que el principal merchant\_id es el siguente: '817d18cd3c31e40e9bff0566baae7758' el cual tiene más de 4360340 transaciones de comercios aliados. siendo esta la principal
* Las 20 principales subsidiary, donde la subsidiary '{subsidiary\_name}' tiene más de {max\_transactions} transacciones siendo la principal subsidiary.
* De la data general vemos que el monto máximo de la transacción es de 4624.79 dólares, y que el mínimo es de 5.94 dólares y la media es de 191.39 dólares con una desviacion general puede llegar a ser de 240.90.
* De la data general tenemos que Dado que una persona utiliza Nequi y tiene transacciones en Nequi, la probabilidad de que el número de transacciones sea 1 es de 92.91%.
* La probabilidad de que sus transacciones sean 2.0 transacciones es del 5.84%.
* La probabilidad de que reciba una o dos transacciones en un día es del 98.75%.
* La probabilidad de que el número de transacciones sea mayor o igual a 3 es del 1.25%.
* El número promedio de transacciones generales en un día por cliente es de: 1.09

Conclusiones:

Se utiliza el algoritmo DBSCAN , para le detención de fraudes de valores atípicos

\* esto bajo el criterio de que los datos no vienen etiquetados, por lo tanto, este deberá ser un modelo no supervisado. Y tratar de identificar esos clientes que posiblemente quieran realizar malas prácticas transaccionales

\* La frecuencia en la cual se debería ejecutar el modelo para ir viendo que clientes sobrepasan las 3 transferencias debería ser hora a hora, dado que el análisis anterior me indica que para las transacciones totales menores a 3000 dólares, el tiempo promedio entre transacciones es de 44.32 minutos. y que el 50% de estas transacciones se realizaron con una diferencia de tiempo entre ellas menor a 5.79 minutos y para las transacciones totales mayores e iguales a 3000 dólares, el tiempo promedio entre transacciones es de 13.67 minutos. y que el 50% de estas transacciones se realizaron con una diferencia de tiempo entre ellas menor a 1.47 minutos., en este caso podemos usar mejor Azure Synapse, dado que es muy conveniente para analizar y manejar información en tiempo real de macrodatos dado que esta herramienta maneja lo mejor de SQL  y Spark e integración de ETL.