PRUEBA TECNICA ANALISIS TECNICO

Sebastián Restrepo Betancur

Tabla de Contenidos:

Preguntas para resolver:

1. Trazar el flujo de datos y explicar bajo cuál criterio seleccionó el modelo final.

2. Proponga con qué frecuencia deben actualizarse los datos y por qué.

3. Diseñar una arquitectura ideal y los recursos necesarios para desplegar su propuesta (Opcional).

Nota: Al referirse a modelo final se entiende como producto de datos que son 2 elementos muy distintos, en este caso la propuesta seleccionada fue un producto de datos(dashboard en power bi) esto debido a que no se posee información en mayor cuantía que amerite un modelo analítico, en el guithub se puede acceder debido al tamaño del mismo a las carpeta donde queda alojado las bases_de_datos, el conjunto csv con la solución, un video del tablero en power bi y por último el tablero desarrollado.

Entregables:

A continuación, se menciona cada archivo que dan respuesta a la prueba técnica:

- El dashboard donde se hace el análisis exploratorio de los data sets tipo parquet, con beneficios posibles desde punto de vista de negocio, junto a este se encuentra el tablero de alertas donde se encuentra toda la información de los clientes que poseen una mala práctica transaccional.

- La base en csv con nombre ejercicio de los clientes con la información Cuenta_de_usuario,fecha_transado,cantidad_de_transacciones,cantidad_transada,minimo_transado,maximo_transado todo esto siendo el insumo del tablero de alertas
- El script-Informe técnico desarrollado en el proyecto con la libreria sparky para correr en caso de que sea necesario volver a cargar la información, o mirar un análisis univariado simple.
- El PDF detallado con cada literal anteriormente mencionado, el presente escrito.

Para acceder a los links se puede acceder por medio del github.

FLUJO DE DATOS:

Para definir el flujo de datos se hace la necesidad de como en cada producto de datos hacer procesos de ETL en caso de ser necesario, esto por ser insumo importante dentro de cualquier tipo de análisis, a continuación se muestra algunos temas exploratorios entre los que se buscan el tipo de información que se tiene en cada base de datos, las dimensiones y la cantidad de na´s; en este proceso se descargaron los archivos parquet el día 22/07/2023 en caso de modificaciones:

DIMENSIONES: hay un total de 10758500 registros para la base sample_data_0006_part_00 y 10758418 registros para la base sample_data_0007_part_00 (1) ambas con un total de 8 caracteristicas.

EN GENERAL EL ESQUEMA PARA LAS DOS TABLAS:

```
root
 |-- merchant_id: string (nullable = true)
 |-- _id: string (nullable = true)
 |-- subsidiary: string (nullable = true)
 |-- transaction_date: timestamp (nullable = true)
 |-- account_number: string (nullable = true)
 |-- user_id: string (nullable = true)
 |-- transaction_amount: decimal(24,8) (nullable = true)
 |-- transaction_type: string (nullable = true)
```

DE IGUAL MANERA ANALISIS DE NA O VALORES FALTANTES, AMBOS CON LOS MISMOS RESULTADOS:

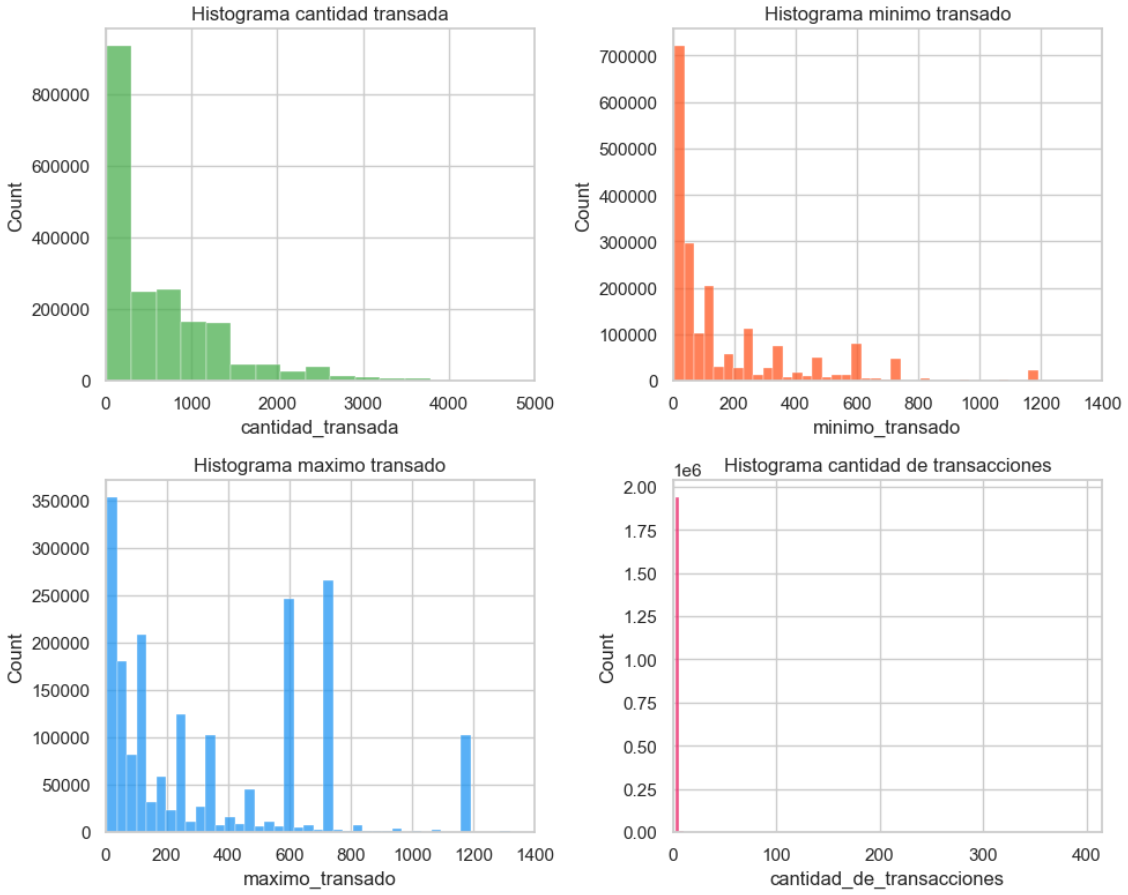
```
+-----------+---+----------+--------------+-------+------------------+----------------+
|merchant_id|_id|subsidiary|account_number|user_id|transaction_amount|transaction_type|
+-----------+---+----------+--------------+-------+------------------+----------------+
|          0|  0|         0|             0|      0|                 0|               0|
+-----------+---+----------+--------------+-------+------------------+----------------+
```

Continuando con el proceso dado que las dos tablas son similares se hace un analisis univariado de cada uno de los elementos del mismos, estos mas detallados en el script, después como parte importante de este y como cumplen el hecho de ser similares en su estructura se concatenan uno debajo del otro, este con la finalidad de que se pueda hacer un analisis final del mismo, ya con las tablas juntas es simplemente correr el siguiente query que construye una nueva tabla la cual agrupa por fecha y USER_ID, la cantidad de transacciones hechas ese mismo dia, el total que forman transacción original, cuanto es lo mínimo que transo en esa fecha y lo máximo, el siguiente es el producto del siguiente query:

```
WITH t1 AS (SELECT *,CAST(concat(YEAR(transaction_date),'-
',MONTH(transaction_date),'-
',DAY(transaction_date)) AS string) AS fecha FROM alertas) \

SELECT user_id AS Cuenta_de_usuario,fecha AS fecha_transado,C
OUNT(*) AS cantidad_de_transacciones,SUM(transaction_amount)
AS cantidad_transada,
MIN(transaction_amount) AS minimo_transado,
MAX(transaction_amount) AS maximo_transado  FROM t1
GROUP BY fecha_transado,Cuenta_de_usuario HAVING cantidad_de_
transacciones > 1 ORDER BY cantidad_de_transacciones DESC
```

En teoria lo que arroja el query y la idea central de la concluión de este es arrojar un data.frame que en este caso es de 1959712 registros con la que se deja un prescedente de detección, control y ajuste de MPT(Malas practicas transaccionales).
A continuación unos histogramas que describen el comportamiento de las mismas distribuciones:



Vemos que en general parecieran ser varias misturas con máximos y minimos bastantes bajos, vale destacar que sobresale la cantidad de transacciones donde por 2 transacciones se genera la alerta.

CRITERIO SELECCIONADO PARA EL MODELO FINAL:

Frente al modelo seleccionado se propone dos soluciones como producto de datos, haciendo la aclaración que no siempre un producto de datos es lo mismo que un modelo

analítico; en este caso no me pareció necesario y no había features para hacer un modelo predictivo para detección de las mismas, y siendo esta mi recomendación para la necesidad propuesta para este caso se presentaron dos propuestas de la cual se seleccionó la primera:

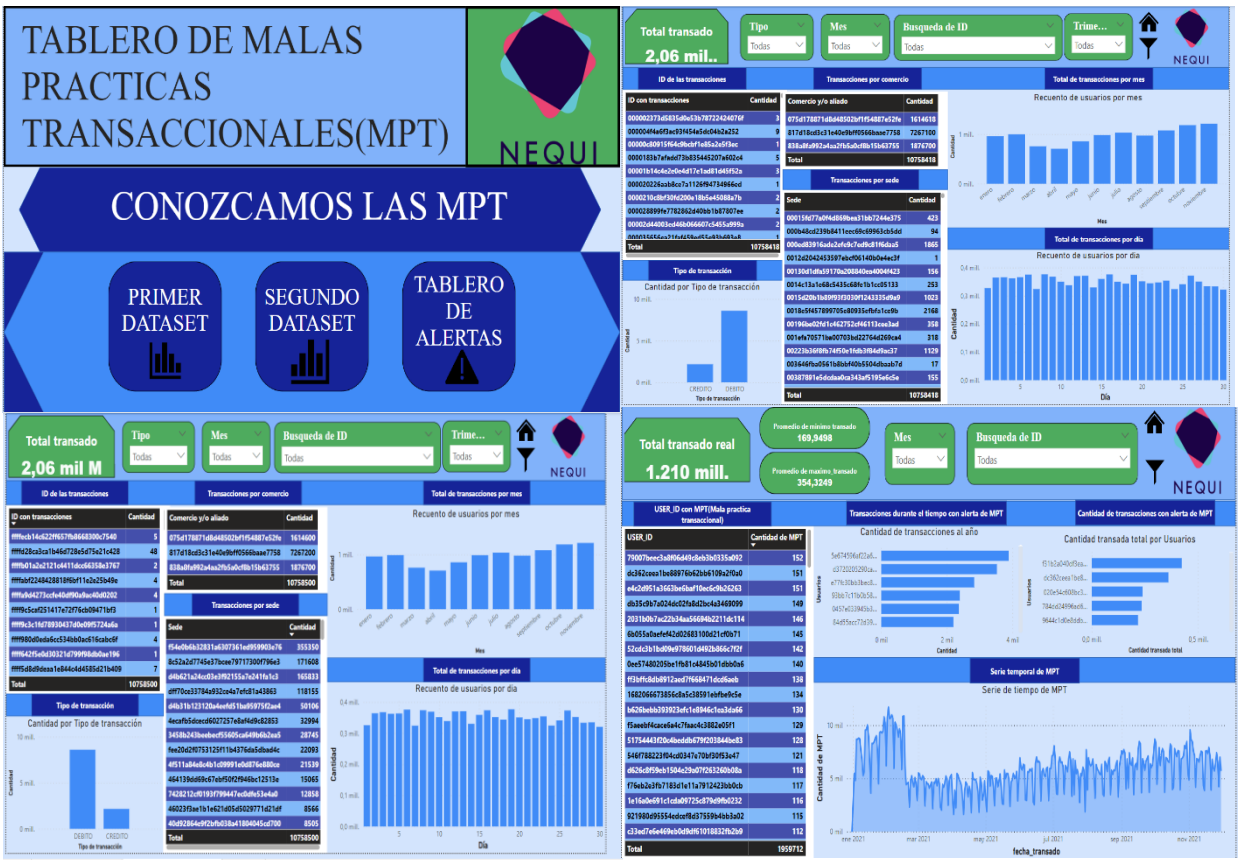
- Primero es la creación de un dashboard bajo 3 tableros, el primero y el segundo un análisis univariado y bivariado de algunas variables de suma importancia tales como la cantidad de USER_ID, los tipos de transacción, las transacciones por sede, la cantidad de comercios junto con la cantidad de transacciones hechas en el mismo, y otros tipos de recuentos que pueden solventar necesidades de negocio y toma de decisiones, entre otros, por último un tablero de alertas que muestran cuantas veces una cuenta repitió una transacción en un día mostrando alertas por cantidades transadas y que pueden ser de utilidad ya sea para mandar alertas a comercios debido a posible detección de mala práctica transaccional siendo esta la propuesta desarrollada.
- La segunda es la creación de una API en cualquier framework(Por la dimensión de escala Python seria mas optimo) ya sea en FLASK por su conexión fácil con SPARK pero este tendría distintos problemas ya sea mantener el control, subir los datos, y demandaría más capacidades pero permitiría ser más autónomos en la detección de MPT.

Proponga con qué frecuencia deben actualizarse los datos y ¿por qué?

La frecuencia en la que debe actualizarse los datos para ejercer un control y detección temprano idealmente seria diario, con una franja de tiempo de diferencia de 24 horas esto con el fin de detectar esta mala práctica transaccional rápidamente, para que genere las respectivas alertas a los negocios encargados de la prevención del mismo; de manera que no se separe mucho el tiempo para que se pueda llegar a tomando decisiones sobre los castigos al usuario dueño de la cuenta donde se esta reportando este tipo de comportamiento, ahora bien en caso de que los castigos no se apliquen diario podría ajustarlo a una franja semanal corriendo el riesgo de que en el tiempo se generen varias alertas acumuladas sobre una misma cuenta.

Diseñar una arquitectura ideal y los recursos necesarios para desplegar su propuesta (Opcional).

En este caso frente a las dos propuestas presentadas contrario de la API que posiblemente demande mas recursos, esta puede ser subida a un entorno WEB de power bi rápidamente además de que al igual que tableau tiene facilidad con grandes datasets, en general aquí se descargaron los parquet de los datos pero teniendo un entorno general para recolectar datos, guardar, administrar y calendarizar tablas permitiría un trabajo más rápido al igual que con su llamado, seria necesario simplemente correr el script con la nueva información que como se mencionó anteriormente la recomendación seria recolectarla diariamente, el dashboard ya esta creado basta con alguien que lo administre y mantenga ese control con el script, los datos y el tablero, a continuación se ve visualmente como se vería tipo collage de imágenes la cual es la base de esta prueba tecnica:


TABLERO DE MALAS PRACTICAS TRANSACCIONALES(MPT)
NEQUI
CONOZCAMOS LAS MPT
PRIMER DATASET
SEGUNDO DATASET
TABLERO DE ALERTAS
Total transado
2,06 mil..
Tipo
Mes
Busqueda de ID
Trime...
Todas
ID de las transacciones
Transacciones por comercio
Total de transacciones por mes
Recuento de usuarios por mes
Transacciones por sede
Total de transacciones por dia
Recuento de usuarios por dia
Tipo de transacción
Cantidad por Tipo de transacción
Total transado
2,06 mil M
Total transado real
1.210 mill.
Promedio de mínimo transado
169,9498
Promedio de maxima transado
354,3249
USER_ID con MPT(Mala practica transaccional)
Transacciones durante el tiempo con alerta de MPT
Cantidad de transacciones con alerta de MPT
Cantidad de transacciones al año
Cantidad transada total por Usuarios
Serie temporal de MPT
Serie de tiempo de MPT
fecha_transado