Universidad de Buenos Aires
Facultad de Ingeniería

Ciencia de Datos - Cátedra Martinelli

Trabajo Práctico N.º 2: “Resolución de Consultas con Spark”

2do Cuatrimestre 2025

Alumno:

- Castro, Martín

Introducción

En el presente trabajo se utilizará **Apache Spark** para ejemplificar cómo realizar un procesamiento distribuido de datos y afrontar los desafíos que implica un análisis de este tipo. A partir del dataset utilizado en el trabajo anterior, repetiremos algunas consultas para observar las diferencias entre Pandas y Spark.
Además, se realizarán nuevas consultas para seguir explorando el dataset provisto.

Desarrollo

Como primer paso, se realizó la estandarización de los archivos CSV a utilizar. Este proceso consistió en transformar las columnas de tipo *string*, convirtiendo su contenido a mayúsculas o minúsculas, según resultara más conveniente, y eliminando los espacios en blanco adicionales que podrían generar inconvenientes en el desarrollo del trabajo.

En relación a los *outliers*, se decidió no descartarlos, ya que no se identificaron registros significativamente alejados de la media que pudieran afectar el análisis.

Finalmente, con el fin de comprender en mayor profundidad el dataset, se llevaron a cabo las consultas sugeridas por la cátedra:

Consulta 1

¿Cuál es el estado que más descuentos tiene en total? y en promedio? Supongan que de una dirección del estilo: 3123 Alan Extension Port Andrea, MA 26926, “MA” es el estado.

Hipótesis:

1) Solo se consideraron las *‘orders’* con estado *‘completed’*, ya que son las que completan todo el flujo del proceso: compra y recepción del pedido.
2) Se asume que todas las *‘shipping_address’* contienen un código de estado seguido de un número (pueden tener otra información pero como mínimo debe tener esto), es decir, son de la forma:
   Estado número
   Esto es necesario para poder parsear las direcciones.
3) Se descartaron las filas con *shipping_address* nulo ya que se consideró que no se sabía la dirección real o no se cargó el dato.

4) Se descartaron las filas con '*shipping_address*' = *"undefined"* dado que los estados eran de USA y no existe un estado llamado *"undefined"*. Se consideró que las filas con este valor hacían referencia a que no se tenía el dato o no se cargo.

Conclusiones:

Observamos que el estado que tiene mayor cantidad de descuento en total es ***TE***. Este resultado es el mismo que obtuvimos en el trabajo práctico de pandas.

Por otra parte, el estado con mayor descuento promedio es ***HI***, nuevamente nos dio el mismo resultado que en el trabajo práctico anterior.

Obs: En esta ocasión no se realizó el análisis del top 5 como en el tp pasado ya que se quería aprovechar el mejor rendimiento de *reduce* sobre *takeOrdered* y solamente nos quedamos el top 1 como marcaba la consigna.

Consulta 2

¿Cuáles son los 5 códigos postales más comunes para las órdenes con estado 'Refunded'? ¿Y cuál es el nombre más frecuente entre los clientes de esas direcciones?

Hipótesis:

1) Solo se consideraron las '*orders'* con estado '*refunded'* ya que lo pedía la consigna.
2) Eliminamos los códigos postales nulos ya que no nos aportaba información para la consulta que queremos hacer.
3) Eliminamos los nombres nulos por el mismo motivo. Si lo dejamos podría alterar los resultados.
4) Eliminamos los nombre *UNDEFINED* porque damos por sentado que no se llama nadie de esa forma y que es una forma de indicar que no se tiene esa información.

Conclusión:

Esta consulta dio como resultado que la persona con más pedidos en los códigos postales más frecuente es **Jessica** con 5 pedidos. Sin embargo, hay varias personas que también hicieron 5 pedidos por lo que este resultado puede variar. En

nuestro caso, dejamos que la función *reduce* decida quién será la persona que aparezca como top 1.

Por otro lado, también hay varios códigos postales que pueden ocupar el puesto 5to puesto, por lo que dependiendo de cuál se elija el resultado también puede variar.

En el desarrollo de esta consulta se noto diferencias entre los resultados de la consulta en Spark y Pandas, esto es debido a la forma en que las distintas funciones eligen la quinta posición. Para este caso, dejamos que la función *takeOrdered* elija el top 5.

Sí se quisiera que en Spark y Pandas tuvieran el mismo resultado a la consulta entonces se debería dar un segundo criterio de ordenamiento como por ejemplo quedarnos con los códigos postales más grandes.

Consulta 3

Para cada tipo de pago y segmento de cliente, devolver la suma y el promedio expresado como porcentaje, de clientes activos y de consentimiento de marketing. Se valora que el output de la consulta tenga nombres claros y en español.

Hipótesis:

1) El promedio expresado como porcentaje se calcula como: *mean() * 100*
2) Los clientes activos representan clientes que han realizado alguna compra en el último tiempo
3) El consentimiento de marketing representa que un usuario ha aceptado que se le envíe publicidad o se usen sus datos para cuestiones de marketing.
4) El consentimiento de marketing por método de pago, por ejemplo con tarjeta de crédito, representa que acepta que le envíe publicidad sobre promociones (o similar) con ese medio de pago
5) Los clientes activos por método de pago representan que en el último tiempo han pagado con ese método o tienen alguna tarjeta asociada (en el caso de pago con tarjeta)

Conclusión:

Se puede observar que hay un gran porcentaje de clientes activos en promedio (casi del 90%).
Por otro lado, el consentimiento de marketing tiene una buena aceptación (70%).

Haciendo hincapié en los clientes activos podemos ver que la mayoría son *regulares* y no destaca ningún método de pago en particular.

El consentimiento de marketing también es mayormente aceptado por los clientes *regulares* aunque también hay muchos usuarios *budget* y *premium* que también dieron su consentimiento.

Consulta 4

Para los productos que contienen en su descripción la palabra *"stuff"* (sin importar mayúsculas o minúsculas), calcular el peso total de su inventario agrupado por marca, mostrar sólo la marca y el peso total de las 5 más pesadas.

Hipótesis:

1) Se quitan las filas las cuales tengan *'brand'* nula ya que lo que buscamos en esta consulta es ver que marcas son las que mayor peso en stock tienen.
2) Se quitan las filas la cuales tengan *'brand' = "UNDEFINED"* ya que consideramos que esta no es una marca, si no que faltan cargar esos datos o no se saben
3) Se quitan las filas que tengan la cantidad de stock o peso nulas ya que no nos aportan para calcular lo pedido.
4) Se quitan las filas que tengan la cantidad de stock o peso < 0 ya que sí es igual a 0 no nos aporta para la suma total y sí es menor a 0 se consideró que no tendría sentido y se debe a un error de tipeo.
5) Se conservaron las palabras que contienen *"stuff"* o alguna de sus variaciones como por ejemplo *"stuffs"*, ya que se interpretó que esto pedía la consigna.
6) Lo que pide la consulta es calcular el peso del inventario por marca. El cálculo se obtiene haciendo: peso de producto * stock del producto (esto por cada fila donde esté esa marca)

Conclusión:

Se puede observar que **3M** es la marca que mayor peso tiene en cuanto stock en kg, esto puede deberse al rubro que manejan, mientras que el top 3 lo completan **Adidas** y **Hasbro**.

En esta ocasión nos dio un resultado diferente a la resolución en Pandas esto puede deberse a que en el tp anterior no aplicamos la condición de:

peso total > 0
cantidad de stock > 0
Otra causa puede ser el filtrado de *“stuff”* que en Pandas se hizo con el método *contains* y en Spark: *“stufff” is in x.description*

Consulta 5

Calculen el porcentaje de productos cuyo stock es al menos 20% más alto que el stock promedio de su marca. Por ejemplo, si el stock promedio de la marca Adidas fuera 100, para los productos de dicha marca la condición será que tengan un stock mayor a 120, y luego se deberá calcular qué porcentaje del total de productos cumple con esta condición.

Hipótesis:

1) Se quitan las filas la cuales tengan *‘brand’ = “UNDEFINED”* ya que consideramos que esta no es una marca, si no que faltan cargar esos datos o no se saben
2) Se quitan las filas con *‘stock_quantity’ = None* ya que no nos aporta información valiosa a nuestra consulta
3) Se quitan las filas con *‘stock_quantity’ <= 0* ya que sí el valor es 0 no nos aporta información y si es < 0 no tendría sentido por lo que consideramos que se debe a un error en la carga
4) Es válido quedarnos solamente con el top 10 de la consulta pedida ya que el resultado de la consulta nos da muchos elementos por lo que sería un problema traerlos a memoria
5) Se asume que a lo que apunta la consulta es obtener el porcentaje de la cantidad de productos que cumplen la condición / la cantidad de productos totales

Conclusión:

Podemos observar que el promedio es de 0.40 aproximadamente. Sí vemos los distintos promedios por marca podemos notar que la mayoría de las marcas coinciden en el rango de 0.39 y 0.41.
Este resultado nos puede indicar que las marcas tienen mayor cantidad de los productos más solicitados, esto lo hacen para tener stock disponible y poder incrementar sus ventas.

Consulta 6

Obtener la cantidad de órdenes que no hayan comprado ninguno de los 10 productos más vendidos.

Hipótesis:

1) Se quitan las filas las cuales tengan *'order_id'* nulo ya que no podemos rastrear a que producto se debe la venta.
2) Se quitan las filas las cuales tengan *'product_id'* nulo ya que no podemos rastrear sobre qué producto se trata.
3) Se quitan las filas las cuales tengan *'quantity'* nulo ya que no podemos saber cuánto de ese producto se vendió.
4) Se quitan las filas las cuales tengan *'quantity'* <= 0 ya que no tendría sentido que se venda 0 o menos de un producto
5) Se interpretó que *quantity* hace referencia a la cantidad de unidades vendidas de ese producto en esa venta en particular.
6) Para calcular la cantidad de ventas de un producto se sumó la cantidad de ventas de cada producto. Es decir, si un producto aparece de la siguiente forma:

Nombre producto	Cantidad
Producto 1	7
Producto 1	5

Se va a contar que la cantidad de ventas es 12

Conclusión:

La cantidad de órdenes que no están en el top 10 de los productos más vendidos son **242369** de un total de 242457 orders. Por lo que hay **88** orders que fueron las más pedidas, este no es un número significativo con respecto al total por lo que podemos intuir que las compras son de una gran cantidad de productos y no están centralizados.

Consulta 7

Obtener los 10 países con mayor cantidad de usuarios inactivos en promedio

Hipótesis:

1) Se quitan las filas las cuales tengan *'first_name'* nulo ya que no podemos rastrear a que persona pertenece la cuenta.
2) Se quitan las filas las cuales tengan *'first_name'* = *UNDEFINED* ya que se considera que no hay personas que se llamen de esta forma.
3) Se quitan las filas las cuales tengan *'last_name'* nulo ya que no podemos rastrear a que persona pertenece la cuenta.
4) Se quitan las filas las cuales tengan *'last_name'* = *UNDEFINED* ya que se considera que no hay personas con este apellido.
5) Se quitan las filas las cuales tengan *'customer_id'* nulo ya que no podemos rastrear a que persona pertenece la cuenta.
6) Se quitan las filas las cuales tengan *'country'* nulo ya que no nos aporta información para nuestra consulta
7) Se quitan las filas las cuales tengan *'country'* = *UNDEFINED* ya que no es un país

Conclusión:

Como puede verse el país con la mayor cantidad de usuarios inactivos es **México** aunque los demás países tienen un porcentaje muy similar. La mayoría cuenta con un porcentaje de aproximadamente 0.1 usuarios inactivos.
Estas coincidencias pueden deberse a cómo fueron creados los datos y que los mismos no corresponden a datos reales por lo que la respuesta a esta consulta se observa uniforme para la mayoría de países.

Consulta 8

Obtener los 2 nombres con mayor cantidad de *helpful_votes* dentro de los 10 países con mayor cantidad de usuarios inactivo en promedio

Hipótesis:

1) Se quitan las filas las cuales tengan *'first_name'* nulo ya que no podemos rastrear a que persona pertenece la cuenta.
2) Se quitan las filas las cuales tengan *'first_name'* = *UNDEFINED* ya que se considera que no hay personas que se llamen de esta forma.

3) Se quitan las filas las cuales tengan *'last_name'* nulo ya que no podemos rastrear a que persona pertenece la cuenta.
4) Se quitan las filas las cuales tengan *'last_name' = UNDEFINED* ya que se considera que no hay personas con este apellido.
5) Se quitan las filas las cuales tengan *'customer_id'* nulo ya que no podemos rastrear a que persona pertenece la cuenta.
6) Se quitan las filas las cuales tengan *'country'* nulo ya que no nos aporta información para nuestra consulta
7) Se quitan las filas las cuales tengan *'country'* = *UNDEFINED* ya que no es un país
8) Se quitan las filas las cuales tengan *'helpful_votes'* nulo ya que no nos aporta información para nuestra consulta
9) Se quitan las filas las cuales tengan *'is_verified_purchase'* nulo ya que solo se considerarán compras verificadas
10) Se quitan las filas las cuales tengan *'is_verified_purchase'* = *False* ya que solo se considerarán compras verificadas

Conclusión:

El top 2 de la consulta fue ***Michael*** con 10305 y ***David*** con 7750.
Al ser nombres comunes en distintas partes del mundo podía esperarse un resultado de este estilo.
Sí nos enfocaremos en los países de la consulta anterior, un nombre como Michael era esperable, ya que había muchos países en el top 10 de habla inglesa (nombre común en esa lengua) y otros países como México donde ese nombre es usual.

Conclusiones

A lo largo de este trabajo se profundizó en la comprensión y el análisis del dataset provisto. Durante el proceso se identificaron diversas particularidades, entre ellas, la ausencia de una gran cantidad de *outliers*, aunque sí se detectó un gran número de registros incompletos, con valores *NaN* o *undefined*, lo que refleja problemas en la carga o disponibilidad de ciertos datos.

También pudimos observar, con la consulta 8, que algunos de los datos son muy similares debido a como fueron generados los datos.

Otro aspecto relevante fue la necesidad de aplicar un proceso de estandarización previo al análisis. Varias columnas presentaban diferencias en el uso de mayúsculas, minúsculas y espacios en blanco dentro de los *strings*, lo que

dificultaba el procesamiento directo de la información. Gracias a esta normalización, se logró garantizar una mayor coherencia en los datos y, por lo tanto, un análisis más confiable.

En conjunto, este trabajo no solo permitió responder las consultas planteadas, sino también adquirir experiencia en la detección, limpieza y preparación de datos, etapas fundamentales en cualquier proyecto de análisis.

Para finalizar, haber trabajado con Spark fue útil para entender cómo trabajar y hacer consultas en ambientes donde los datos están distribuidos y donde los procesos se hacen de manera diferente a lo que conocíamos previamente de Pandas.