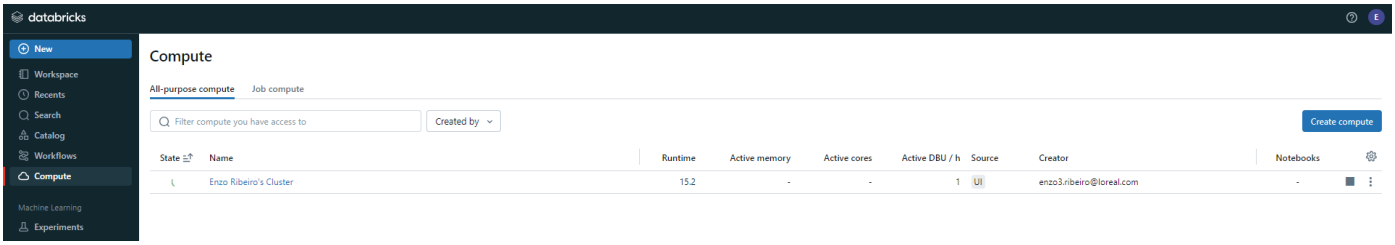
MVP – Sprint Engenharia de Dados

Aluno: Enzo Sant'Anna Ribeiro

Iremos fazer o MVP dessa disciplina no Databricks Community Edition, com o objetivo de importar uma fonte de dados, criar um ETL para esse processo de importação e depois fazer as consultas SQL para responder as perguntas foco do trabalho.

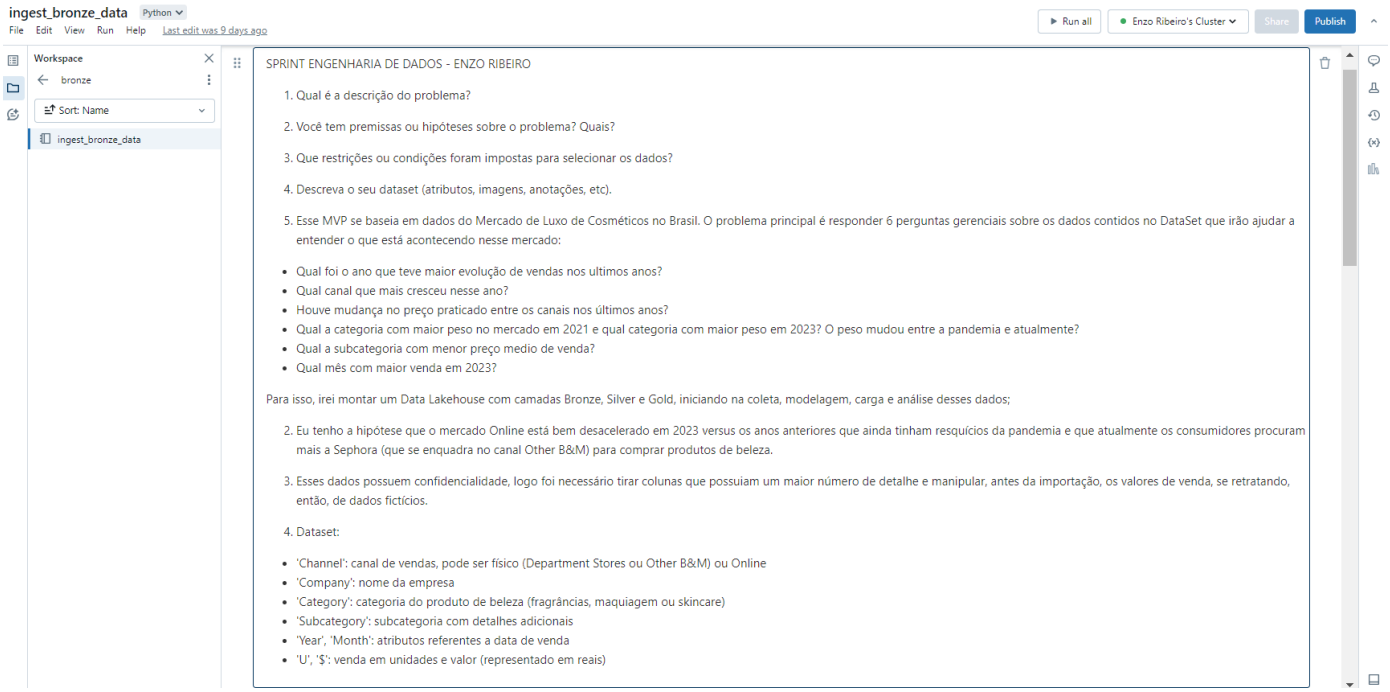
Em primeiro lugar, temos que criar um Cluster, onde os notebooks e as os bancos de dado serão armazenados. Para isso, criamos um cluster com a versão 15.2 do Databricks, uma versão recente que conta com Spark 3.5.0. Como estamos na versão gratuita da plataforma, o Cluster sai do ar após um tempo de inatividade.

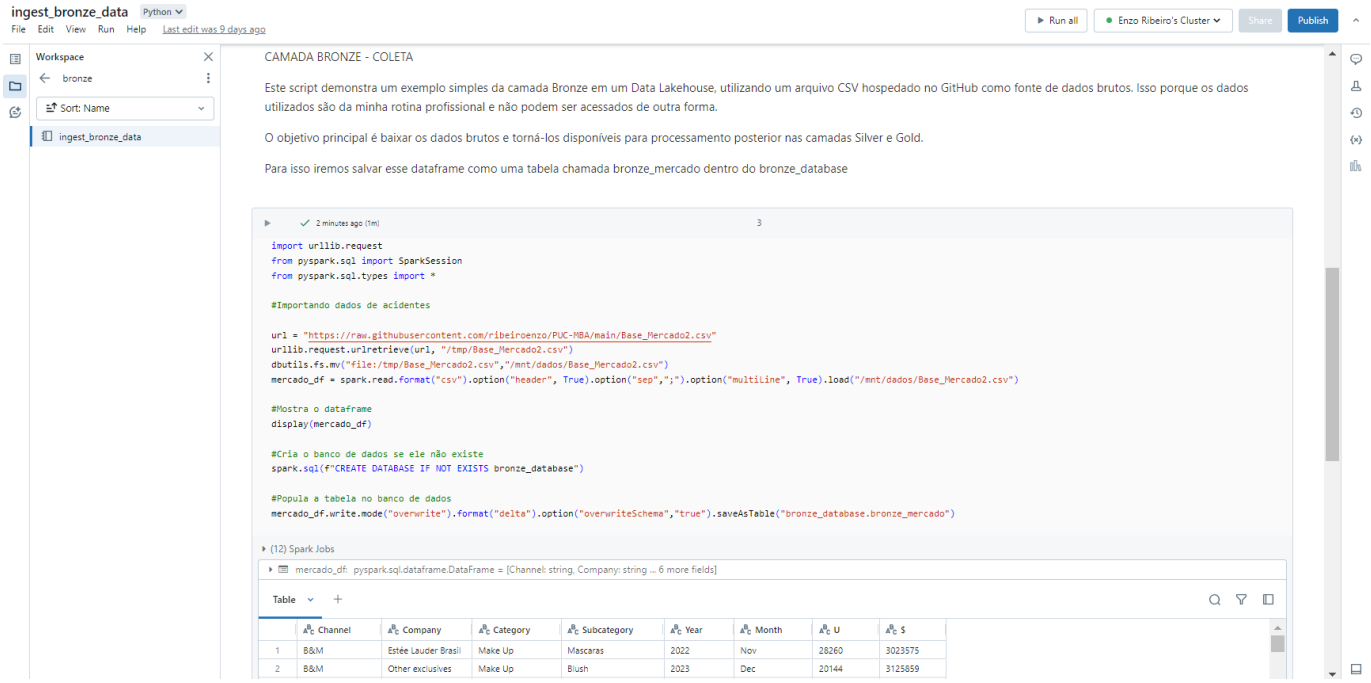


Após criar o cluster, criei as pastas bronze, silver e gold, para organizar onde cada um dos notebooks estará presente, de acordo com o objetivo de cada uma das camadas do Data Lake. Vamos iniciar pelo notebook da camada bronze:

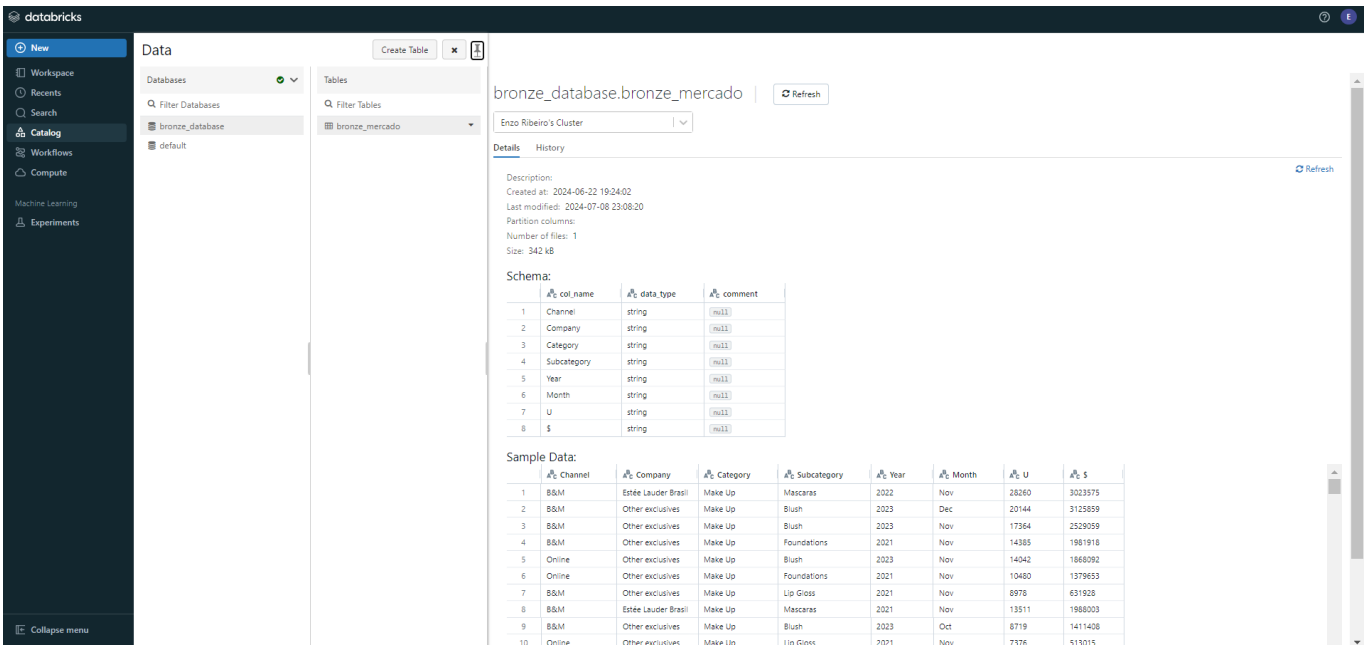
1- CAMADA BRONZE

Dentro da pasta bronze, irei criar o notebook ingest_bronze_data para introduzir o problema analisado e fazer a ingestão dos dados iniciais.





Após a ingestão dos dados, a tabela foi salva dentro do bronze_database como bronze_mercado. Essa tabela será consultada para o início da camada Silver. Com isso, a tabela fica disponível para visualização e para entender o formato de cada coluna dentro da seção Catalog.



2- CAMADA SILVER

Com os dados da camada bronze salvos em uma tabela do database, a próxima etapa é fazermos a transformação e check de qualidade desses dados, para posteriormente os carregá-los na camada gold para consulta do usuário e do time de Business Intelligence.

Para isso, dentro da pasta “Silver”, criamos o notebook transform_silver_data, que será usado como pipeline dos dados nesse estudo.

transform_silver_data Python

File Edit View Run Help Last edit was 4 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

- create_dim_channel
- create_dim_date
- transform_silver_data

CAMADA SILVER - TRANSFORMAÇÃO E CARGA

Este script demonstra o processamento de dados na camada Silver, utilizando como base os dados brutos da camada Bronze.

Os dados são limpos, enriquecidos e transformados para torná-los mais confiáveis e úteis para análise.

Para isso iremos salvar esse dataframe como uma tabela chamada silver_mercado dentro do silver_database

08:14 PM (<1s) 2

```
from pyspark.sql import SparkSession

# Inicializa a SparkSession (se ainda não estiver inicializada)
spark = SparkSession.builder.appName("Camada Silver").getOrCreate()

# Carrega os dados da tabela Bronze em um DataFrame
bronze_mercado_df = spark.table("bronze_database.bronze_mercado")
```

bronze_mercado_df: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 6 more fields]

08:14 PM (1s) 3

```
# Registra o DataFrame como uma view temporária para usar em consultas SQL
bronze_mercado_df.createOrReplaceTempView("bronze_mercado")

# Consulta SQL para obter os nomes e tipos das colunas
resultado_schema = spark.sql("DESCRIBE TABLE bronze_mercado")

# Exibe o resultado
resultado_schema.show()
```

resultado_schema: pyspark.sql.dataframe.DataFrame = [col_name: string, data_type: string ... 1 more field]

```
+-----------+---------+-------+
|   col_name|data_type|comment|
+-----------+---------+-------+
|    Channel|   string|   NULL|
|    Company|   string|   NULL|
|   Category|   string|   NULL|
|Subcategory|   string|   NULL|
|       Year|   string|   NULL|
```

transform_silver_data Python

File Edit View Run Help Last edit was 4 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

- create_dim_channel
- create_dim_date
- transform_silver_data

08:14 PM (2s) 4

```
bronze_mercado_df.head()
```

(1) Spark Jobs

```
Row(Channel='B&M', Company='Estée Lauder Brasil', Category='Make Up', Subcategory='Mascaras', Year='2022', Month='Nov', U='28260', $='3023575')
```

08:14 PM (2s) 5

```
from pyspark.sql.functions import col, when, lit, to_date, to_timestamp

# Mapeamento de meses (pode continuar usando o seu dicionário 'meses')

# Cria o dataframe silver_mercado que será manipulado nessa camada
silver_mercado_df = bronze_mercado_df

# Cria a coluna 'Month_Num' usando expressões when()
silver_mercado_df = silver_mercado_df.withColumn(
    "Month_Num",
    when(col("Month") == "Jan", "01")
    .when(col("Month") == "Feb", "02")
    .when(col("Month") == "Mar", "03")
    .when(col("Month") == "Apr", "04")
    .when(col("Month") == "May", "05")
    .when(col("Month") == "Jun", "06")
    .when(col("Month") == "Jul", "07")
    .when(col("Month") == "Aug", "08")
    .when(col("Month") == "Sep", "09")
    .when(col("Month") == "Oct", "10")
    .when(col("Month") == "Nov", "11")
    .when(col("Month") == "Dec", "12")
    .otherwise(None)  # Valor padrão se o mês não for encontrado
)

silver_mercado_df.show(5)
```

(1) Spark Jobs

silver_mercado_df: pyspark.sql.dataframe.DataFrame

```
Channel: string
Company: string
Category: string
```

transform_silver_data Python

File Edit View Run Help Last edit was 5 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

- create_dim_channel
- create_dim_date
- transform_silver_data

08:14 PM (1s) 6 Python

```
import pyspark.sql.functions as F

# Cria a coluna 'Date' no formato YYYY-MM-DD (Corrigido)
silver_mercado_df = silver_mercado_df.withColumn(
    "Date",
    F.date_format(F.expr("make_date(Year, Month_Num, 1)"), "MM/dd/yyyy")
)
silver_mercado_df.show(5)
```

(1) Spark Jobs

silver_mercado_df: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 8 more fields]

```
+-------+--------------------+--------+-----------+----+-----+-----+--------+----------+
|Channel|             Company|Category|Subcategory|Year|Month|    U|      $|Month_Num|      Date|
+-------+--------------------+--------+-----------+----+-----+-----+--------+----------+
|    B&M|Estée Lauder Brasil|Make Up|   Mascaras|2022|  Nov|28260|3023575|       11|11/01/2022|
|    B&M|   Other exclusives|Make Up|      Blush|2023|  Dec|20144|3125859|       12|12/01/2023|
|    B&M|   Other exclusives|Make Up|      Blush|2023|  Nov|17364|2529059|       11|11/01/2023|
|    B&M|   Other exclusives|Make Up|Foundations|2021|  Nov|14385|1981918|       11|11/01/2021|
| Online|   Other exclusives|Make Up|      Blush|2023|  Nov|14042|1868092|       11|11/01/2023|
+-------+--------------------+--------+-----------+----+-----+-----+--------+----------+
only showing top 5 rows
```

08:14 PM (1s) 7

```
# Remove as colunas antigas (opcional)
silver_mercado_df = silver_mercado_df.drop("Year", "Month_Num", "Month")

silver_mercado_df = silver_mercado_df.withColumn("Date", to_date(col("Date"), "MM/dd/yyyy"))

silver_mercado_df.show(5)
```

(1) Spark Jobs

silver_mercado_df: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 5 more fields]

```
+-------+--------------------+--------+-----------+-----+-------+----------+
|Channel|             Company|Category|Subcategory|    U|      $|      Date|
+-------+--------------------+--------+-----------+-----+-------+----------+
|    B&M|Estée Lauder Brasil|Make Up|   Mascaras|28260|3023575|2022-11-01|
|    B&M|   Other exclusives|Make Up|      Blush|20144|3125859|2023-12-01|
```

transform_silver_data Python

File Edit View Run Help Last edit was 5 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

create_dim_channel

create_dim_date

transform_silver_data

08:14 PM (1s) 8 Python

```
from pyspark.sql.functions import col, regexp_replace

# 1. Converter coluna "U" para inteiro
silver_mercado_df = silver_mercado_df.withColumn("U", col("U").cast("int"))

# 2. Converter coluna "$" para double (decimal)
silver_mercado_df = silver_mercado_df.withColumn("$", col("$").cast("double"))

silver_mercado_df.head()
```

(1) Spark Jobs

silver_mercado_df: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 5 more fields]

```
Row(Channel='B&M', Company='Estée Lauder Brasil', Category='Make Up', Subcategory='Mascaras', U=28260, $=3023575.0, Date=datetime.date(2022, 11, 1))
```

08:14 PM (2s) 9

```
from pyspark.sql.functions import col, upper, trim

# 1. Padronizar Colunas de String (Caixa Alta, Remoção de Espaços)
silver_mercado_df = silver_mercado_df.withColumn("Channel", upper(trim(col("Channel"))))
silver_mercado_df = silver_mercado_df.withColumn("Company", upper(trim(col("Company"))))
silver_mercado_df = silver_mercado_df.withColumn("Category", upper(trim(col("Category"))))
silver_mercado_df = silver_mercado_df.withColumn("Subcategory", upper(trim(col("Subcategory"))))

# 2. Tratando Valores Ausentes (se houver, substituir por "Unknown")
silver_mercado_df = silver_mercado_df.fillna({"Channel": "Unknown", "Company": "Unknown", "Category":"Unknown", "Subcategory":"Unknown"})

silver_mercado_df.head()
```

(1) Spark Jobs

silver_mercado_df: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 5 more fields]

```
Row(Channel='B&M', Company='ESTÉE LAUDER BRASIL', Category='MAKE UP', Subcategory='MASCARAS', U=28260, $=3023575.0, Date=datetime.date(2022, 11, 1))
```

transform_silver_data Python

File Edit View Run Help Last edit was 6 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

create_dim_channel

create_dim_date

transform_silver_data

08:14 PM (2s) 10 Python

```
# 3. Validação de Dados (adicione verificações para a qualidade dos dados)
# Filtrar as linhas onde 'U' (unidades) seja menor que 0 OU '$' (valor total) seja menor que 'U'
silver_mercado_df_validação = silver_mercado_df.filter((col("U") <= 0) & (col("$") <= col("U")))

silver_mercado_df_validação.show(20)
```

(1) Spark Jobs

silver_mercado_df_validação: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 5 more fields]

```
| ONLINE|    SHISEIDO BRASIL|SKINCARE|                LIPS|  0|  -34.0|2023-01-01|
| ONLINE|    SHISEIDO BRASIL|SKINCARE|          BC FIRMING|-1|-1366.0|2023-04-01|
|    B&M|PRESTIGE COSMETICOS| MAKE UP|           LIP LINER|  0|  -39.0|2021-07-01|
| ONLINE|PRESTIGE COSMETICOS| MAKE UP|           EYE LINER|-1|  -209.0|2023-04-01|
| ONLINE|       UNIVERSE S.A|SKINCARE|      MIXED SKINCARE|  0|  -860.0|2023-05-01|
| ONLINE|       UNIVERSE S.A|SKINCARE|      MIXED SKINCARE|  0| -1530.0|2023-08-01|
| ONLINE|        LVMH BRASIL|SKINCARE|SUN BLOCK LOW/MID...|  0|    -5.0|2021-08-01|
| ONLINE|  OTHER EXCLUSIVES|SKINCARE|           AFTER SUN|  0|   -30.0|2022-04-01|
|    B&M|PRESTIGE COSMETICOS|SKINCARE|       BODY CARE SET|  0|   -42.0|2021-02-01|
|    B&M|             SISLEY|SKINCARE|      ACNE TREATMENT|  0|-1427.0|2023-11-01|
| ONLINE|             SISLEY|SKINCARE|          BC OTHERS|  0|  -212.0|2022-12-01|
| ONLINE|             SISLEY|SKINCARE|WHITENING/DARK SPOTS|  0|-1230.0|2023-12-01|
| ONLINE|             AMYRIS|SKINCARE|      MIXED SKINCARE|  0|  -197.0|2023-06-01|
|    B&M|       UNIVERSE S.A|SKINCARE|         OTHER CARE|  0|   -23.0|2023-11-01|
|    B&M|    L'ORÉAL BRASIL|SKINCARE|       MULTIBENEFIT|-1|  -199.0|2022-04-01|
| ONLINE|    L'ORÉAL BRASIL| MAKE UP|            POWDERS|-1|  -279.0|2022-10-01|
| ONLINE|    L'ORÉAL BRASIL| MAKE UP|            POWDERS|-1|  -231.0|2023-03-01|
| ONLINE|    L'ORÉAL BRASIL| MAKE UP|        HIGHLIGHTER|-1|  -299.0|2023-04-01|
+-------+-------------------+--------+-------------------+---+-------+----------+
only showing top 20 rows
```

08:14 PM (<1s) 11

```
#Observamos que o dataframe tem algumas linhas com dados sem qualidade para a análise que queremos fazer. Possivelmente, as linhas com valores negativos em Unidades ou Valor de Venda são devoluções de produtos comprados em meses anteriores. Como queremos apresentar na camada gold apenas os dados referentes a venda daquele mês e essas devoluções podem interferir no diagnóstico, iremos tirar essas linhas

# Retirar as linhas onde 'U' (unidades) seja menor que 0 OU '$' (valor total) seja menor que 'U'
silver_mercado_df = silver_mercado_df.filter((col("U") >= 0) & (col("$") >= col("U")))
```

silver_mercado_df: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 5 more fields]

transform_silver_data Python

File Edit View Run Help Last edit was 7 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

create_dim_channel

create_dim_date

transform_silver_data

08:14 PM (5s) 12 Python

```
# Como estou ligado ao negócio e sei que as colunas de Category e Channel são bem específicas e deveriam retratar as 4 categorias do mercado de luxo (Men Fragrances, Women Fragrances, Skincare e Make Up) e os 3 canais (Online, Department Stores e Other B&M, que são as outras lojas físicas sem ser lojas de departamento), respectivamente. Vamos checar se alguma coluna pode ter vindo com falta de qualidade ou escrita errada

# Seleciona as colunas "Category" e "Channel" e remove duplicatas
category_distintos = silver_mercado_df.select("Category").distinct()
channel_distintos = silver_mercado_df.select("Channel").distinct()

# Mostra os valores distintos
category_distintos.show()
channel_distintos.show()
```

(4) Spark Jobs

category_distintos: pyspark.sql.dataframe.DataFrame = [Category: string]

channel_distintos: pyspark.sql.dataframe.DataFrame = [Channel: string]

```
+----------------+
|        Category|
+----------------+
|  MEN FRAGRANCES|
|         MAKE UP|
|WOMEN FRAGRANCES|
|        SKINCARE|
+----------------+

+-----------------+
|          Channel|
+-----------------+
|        OTHER B&M|
|DEPARTMENT STORES|
|              B&M|
|           ONLINE|
+-----------------+
```

transform_silver_data Python

File Edit View Run Help Last edit was 7 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

create_dim_channel

create_dim_date

transform_silver_data

08:14 PM (2s) 13 Python

```
# A coluna Category veio exatamente de acordo com a expectativa, porém a coluna Channel veio com um valor indevido, onde Other B&M e B&M significam a mesma coisa e devemos substituir "B&M" por "Other B&M" na coluna "Channel", para ficar com os 3 canais corretos para essa análise. Na visão dos canais, Online significa tudo que é comprado nos websites, Department Stores o canal composto por Renner, Riachuelo e C&A, e Other B&M os outros, sendo eles as lojas da Sephora e das Perfumarias Seletivas.

silver_mercado_df = silver_mercado_df.withColumn(
    "Channel",
    when(col("Channel") == "B&M", "OTHER B&M").otherwise(col("Channel"))
)

channel_distintos = silver_mercado_df.select("Channel").distinct()
channel_distintos.show()
```

(2) Spark Jobs

silver_mercado_df: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 5 more fields]

channel_distintos: pyspark.sql.dataframe.DataFrame = [Channel: string]

```
+----------------+
|         Channel|
+----------------+
|       OTHER B&M|
|DEPARTMENT STORES|
|          ONLINE|
+----------------+
```

transform_silver_data Python

File Edit View Run Help Last edit was 8 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

create_dim_channel

create_dim_date

transform_silver_data

08:14 PM (5s) 14

```
from pyspark.sql.functions import col, count

# Agrupa por todas as colunas e conta as ocorrências
duplicatas = silver_mercado_df.groupBy("Channel", "Company", "Category", "Subcategory","Date").count().filter(col("count") > 1)

# Exibe as linhas duplicadas (com a contagem)
duplicatas.show(20)
```

(2) Spark Jobs

duplicatas: pyspark.sql.dataframe.DataFrame = [Channel: string, Company: string ... 4 more fields]

```
+-------+-------+--------+-----------+----+-----+
|Channel|Company|Category|Subcategory|Date|count|
+-------+-------+--------+-----------+----+-----+
+-------+-------+--------+-----------+----+-----+
```

08:14 PM (4s) 15 Python

```
# Por mais que não veio nenhuma coluna duplicada, pode ser que na recorrência de atualização elas comecem a aparecer. Como não queremos ter colunas duplicadas, vamos agrupar as linhas através das colunas dimensão para somar o U e $

from pyspark.sql.functions import col, sum, avg

# 1. Definir as colunas que identificam as duplicatas
colunas_chave = ["Channel", "Company","Category","Subcategory", "Date"]  # Adapte com as colunas relevantes

# 2. Agrupar pelas colunas chave
silver_mercado_df_agrupado = silver_mercado_df.groupBy(colunas_chave)

# 3. Aplicar agregações nas outras colunas
silver_mercado_df_agrupado = silver_mercado_df_agrupado.agg(
    sum("U").alias("SellOut_Units"),
    sum("$").alias("SellOut_Value")
)

# Exibe o resultado
silver_mercado_df_agrupado.show()
```

transform_silver_data Python

File Edit View Run Help Last edit was 8 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

silver

Sort: Name

create_dim_channel

create_dim_date

transform_silver_data

```
|  ONLINE|        LVMH BRASIL| MAKE UP|       FOUNDATIONS|2021-12-01|         2524|     677558.0|
|  ONLINE|ESTÉE LAUDER BRASIL|SKINCARE|        EYES CARE|2021-08-01|          656|     190678.0|
|OTHER B&M|   OTHER EXCLUSIVES|SKINCARE|    MIXED SKINCARE|2023-01-01|          182|      70087.0|
+--------+-------------------+--------+------------------+----------+-------------+-------------+
only showing top 20 rows
```

08:14 PM (19s) 16

```
#Mostra o dataframe
display(silver_mercado_df_agrupado)

# Salvar os dados refinados na camada Silver
spark.sql(f"CREATE DATABASE IF NOT EXISTS silver_database")
silver_mercado_df_agrupado.write.mode("overwrite").format("delta").option("overwriteSchema","true").saveAsTable("silver_database.silver_mercado")
```

(13) Spark Jobs

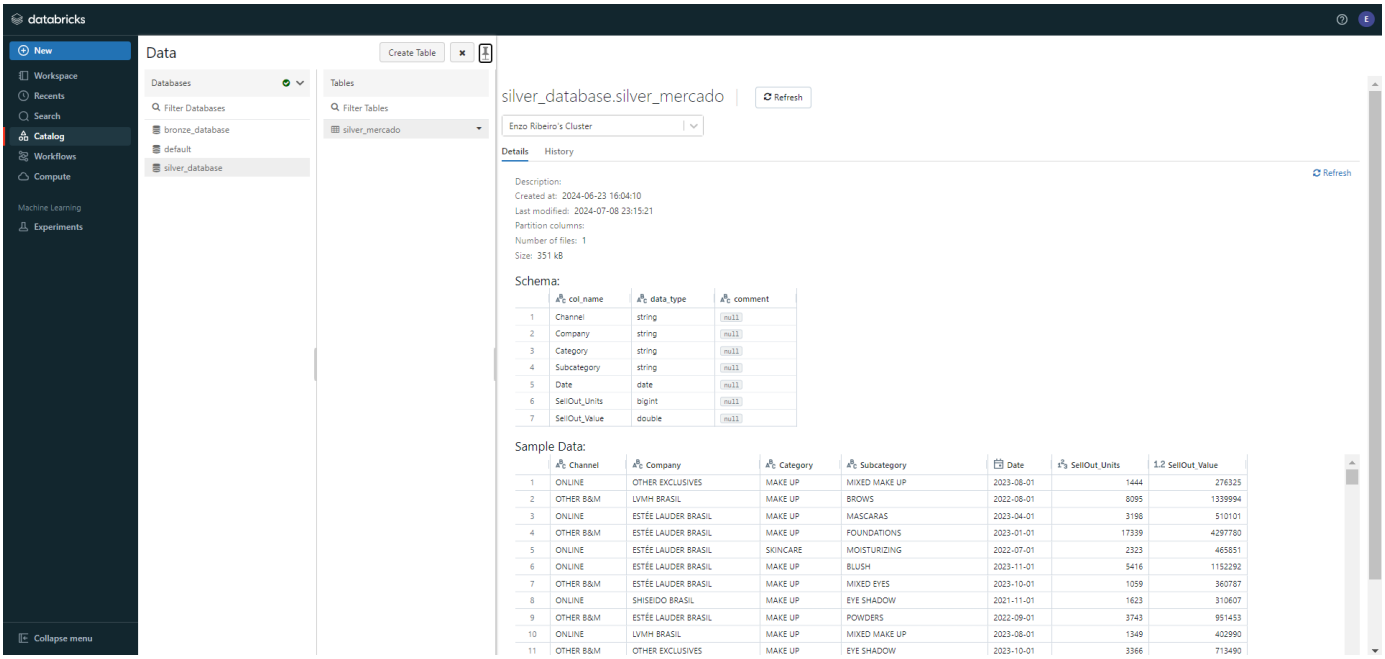
Table

	Channel	Company	Category	Subcategory	Date	SellOut_Units	SellOut_Value
1	ONLINE	OTHER EXCLUSIVES	MAKE UP	MIXED MAKE UP	2023-08-01	1444	276325
2	OTHER B&M	LVMH BRASIL	MAKE UP	BROWS	2022-08-01	8095	1339994
3	ONLINE	ESTÉE LAUDER BRASIL	MAKE UP	MASCARAS	2023-04-01	3198	510101
4	OTHER B&M	ESTÉE LAUDER BRASIL	MAKE UP	FOUNDATIONS	2023-01-01	17339	4297780
5	ONLINE	ESTÉE LAUDER BRASIL	SKINCARE	MOISTURIZING	2022-07-01	2323	465851
6	ONLINE	ESTÉE LAUDER BRASIL	MAKE UP	BLUSH	2023-11-01	5416	1152292
7	OTHER B&M	ESTÉE LAUDER BRASIL	MAKE UP	MIXED EYES	2023-10-01	1059	360787
8	ONLINE	SHISEIDO BRASIL	MAKE UP	EYE SHADOW	2021-11-01	1623	310607
9	OTHER B&M	ESTÉE LAUDER BRASIL	MAKE UP	POWDERS	2022-09-01	3743	951453
10	ONLINE	LVMH BRASIL	MAKE UP	MIXED MAKE UP	2023-08-01	1349	402990
11	OTHER B&M	OTHER EXCLUSIVES	MAKE UP	EYE SHADOW	2023-10-01	3366	713490
12	OTHER B&M	LVMH BRASIL	MAKE UP	CONCEALER	2022-04-01	2908	582102
13	ONLINE	SHISEIDO BRASIL	SKINCARE	MULTIBENEFIT	2021-12-01	1029	248185
14	ONLINE	ESTÉE LAUDER BRASIL	MAKE UP	HIGHLIGHTER	2023-06-01	588	145347
15	OTHER B&M	OTHER EXCLUSIVES	SKINCARE	TONICS/ASTRINGENT	2021-10-01	259	37144

10,000+ rows | Truncated data due to row limit | 19.05 seconds runtime

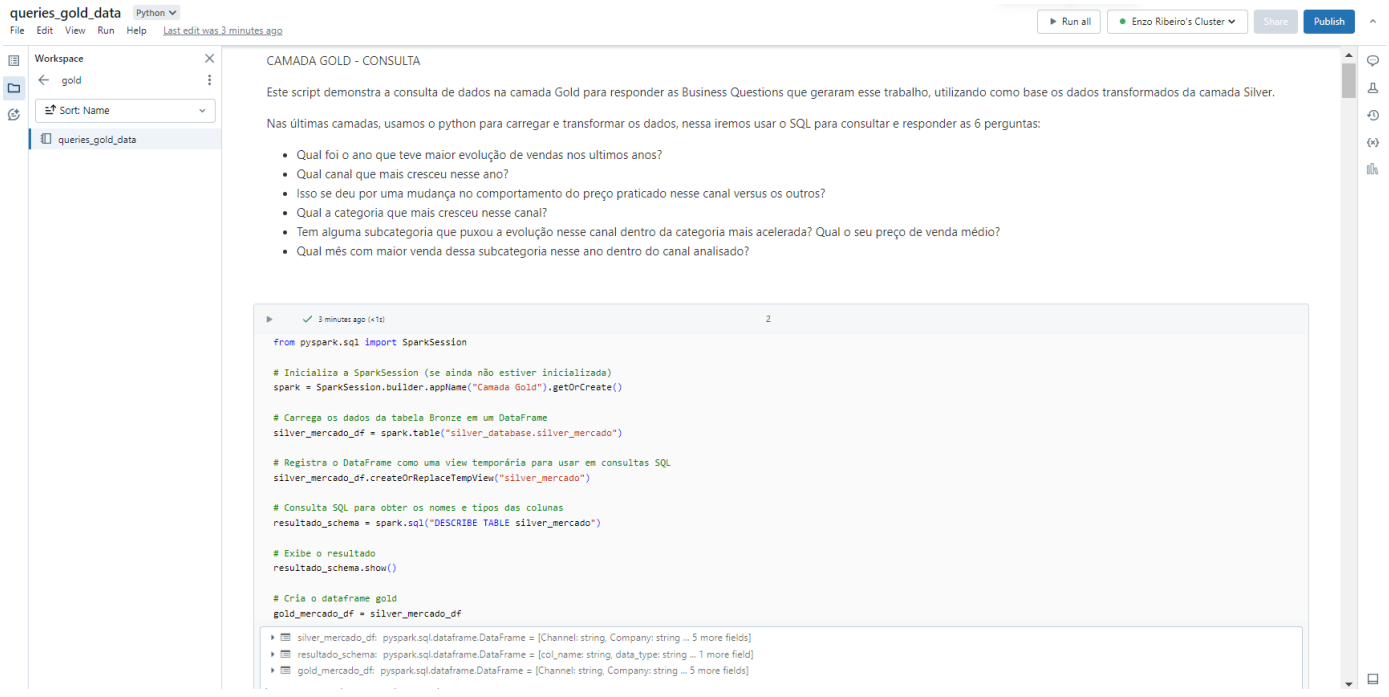
Refreshed 11 minutes ago

Após a transformação dos dados, a tabela foi salva dentro do silver_database como silver_mercado. Essa tabela será consultada para o início da camada Gold. Com isso, a tabela fica disponível para visualização e para entender o formato de cada coluna dentro da seção Catalog, agora já com os dados tratados para uma consulta do time de BI.



3- CAMADA GOLD

Com os dados da camada silver salvos em uma tabela do database, a próxima etapa é fazermos as consultas SQL para responder as 6 perguntas feitas pelo time de negócios relacionados aos dados do mercado de cosméticos de luxo no Brasil. Para isso, iremos usar o notebook queries_gold_data dentro da pasta gold do meu workspace no Databricks.



queries_gold_data Python

File Edit View Run Help Last edit was 3 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

4 minutes ago (14s) 3 SQL

```
%sql
CREATE OR REPLACE TABLE gold_vendas_por_canal_mes AS
    SELECT
        year(m.Date) AS Ano,
        month(m.Date) AS Mes,
        m.Channel,
        SUM(m.SellOut_Value) AS TotalVendas,
        SUM(m.SellOut_Value) / SUM(m.SellOut_Units) AS PrecoMedioVenda
    FROM silver_database.silver_mercado m
    GROUP BY Ano, Mes, m.Channel;
```

(11) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]

Query returned no results

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[4]`. Learn more

4 minutes ago (2s) 4

```
# Execute a consulta para visualizar a tabela criada
resultado = spark.sql("SELECT * FROM gold_vendas_por_canal_mes")

# Exibe o resultado
display(resultado)
```

(2) Spark Jobs

resultado: pyspark.sql.dataframe.DataFrame = [Ano: integer, Mes: integer ... 3 more fields]

Table

	Ano	Mes	Channel	TotalVendas	PrecoMedioVenda
1	2021	5	OTHER B&M	85810055	275.5453424485982
2	2021	3	OTHER B&M	20898939	272.6719159762542
3	2022	8	ONLINE	76648984	303.92142743854083
4	2022	11	DEPARTMENT STORES	62350407	439.8648808809939
5	2023	8	DEPARTMENT STORES	56062469	498.51917160184246
6	2021	12	OTHER B&M	174704022	287.3752274110506
7	2022	12	DEPARTMENT STORES	101871868	478.76393099009179

queries_gold_data Python

File Edit View Run Help Last edit was 4 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

4 minutes ago (12s) 5

```
%sql

CREATE OR REPLACE TABLE gold_crescimento_vendas_mensal AS

WITH VendasMensais AS (
    SELECT
        year(Date) AS Ano,
        date_format(Date, 'MMMM') AS Mes,
        Channel,
        SUM(SellOut_Value) AS TotalVendas,
        SUM(SellOut_Units) AS TotalUnidades -- Incluindo a soma das unidades vendidas
    FROM silver_database.silver_mercado
    GROUP BY Ano, Mes, Channel
)

SELECT
    m.Ano,
    m.Mes,
    m.Channel,
    m.TotalVendas,
    m.TotalUnidades, -- Incluindo a coluna TotalUnidades na tabela final
    LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Channel ORDER BY m.Ano, m.Mes) AS VendasMesAnterior,
    (m.TotalVendas - LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Channel ORDER BY m.Ano, m.Mes)) AS DiferencaMesAnterior,
    ((m.TotalVendas - LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Channel ORDER BY m.Ano, m.Mes)) / LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Channel ORDER BY m.Ano, m.Mes)) * 100 AS
    CrescimentoPercentual
FROM VendasMensais m;
```

(11) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]

Query returned no results

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[6]`. Learn more

queries_gold_data Python

File Edit View Run Help Last edit was 4 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

4 minutes ago (10s) 6

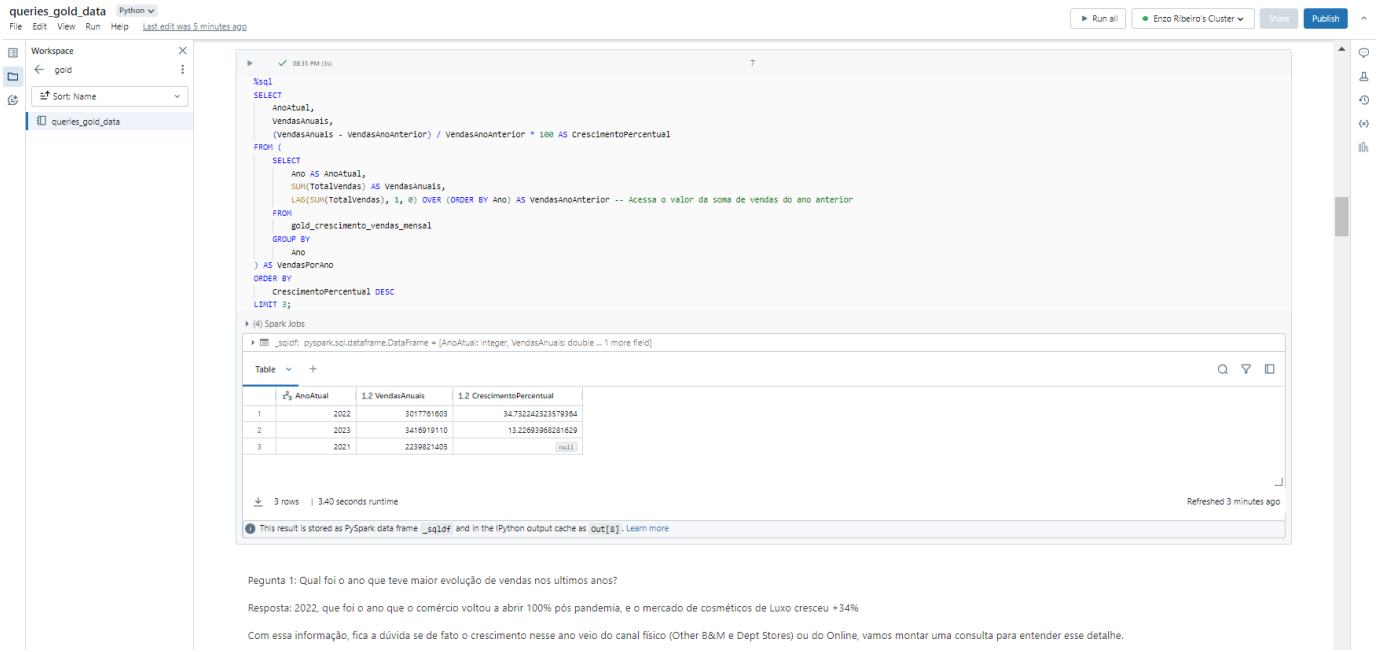
```
%sql
CREATE OR REPLACE TABLE gold_crescimento_vendas_anual AS

WITH VendasAnuais AS (
    SELECT
        year(m.Date) AS Ano,
        SUM(m.SellOut_Value) AS TotalVendasAno
    FROM silver_database.silver_mercado m
    GROUP BY Ano  -- Removido o agrupamento por Channel
)

SELECT
    v.Ano,
    v.TotalVendasAno,
    LAG(v.TotalVendasAno, 1, 0) OVER (ORDER BY v.Ano) AS VendasAnoAnterior,  -- Removido PARTITION BY Channel
    (v.TotalVendasAno - LAG(v.TotalVendasAno, 1, 0) OVER (ORDER BY v.Ano)) AS DiferencaAnoAnterior,
    ((v.TotalVendasAno - LAG(v.TotalVendasAno, 1, 0) OVER (ORDER BY v.Ano)) / LAG(v.TotalVendasAno, 1, 0) OVER (ORDER BY v.Ano)) * 100 AS CrescimentoPercentual
FROM VendasAnuais v;
```

(12) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]

Query returned no results

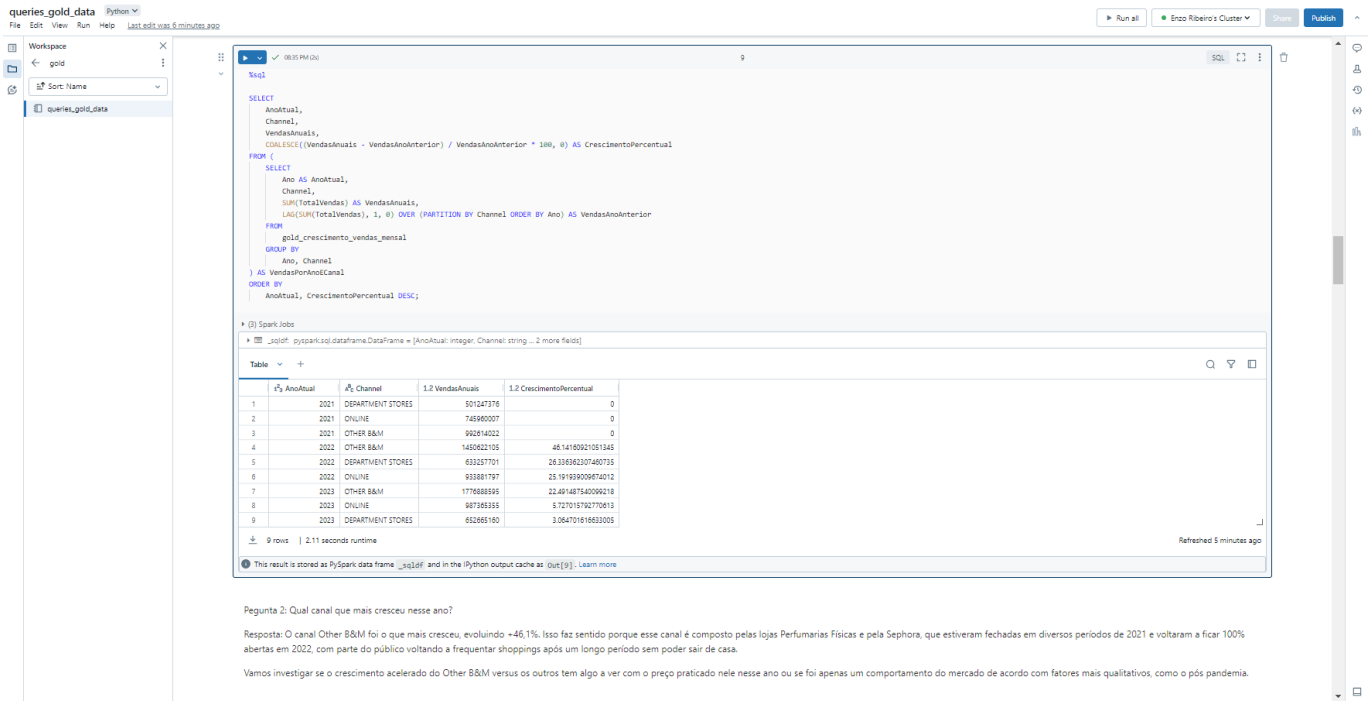
This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[7]`. Learn more



Pegunta 1: Qual foi o ano que teve maior evolução de vendas nos ultimos anos?

Resposta: 2022, que foi o ano que o comércio voltou a abrir 100% pós pandemia, e o mercado de cosméticos de Luxo cresceu +34%

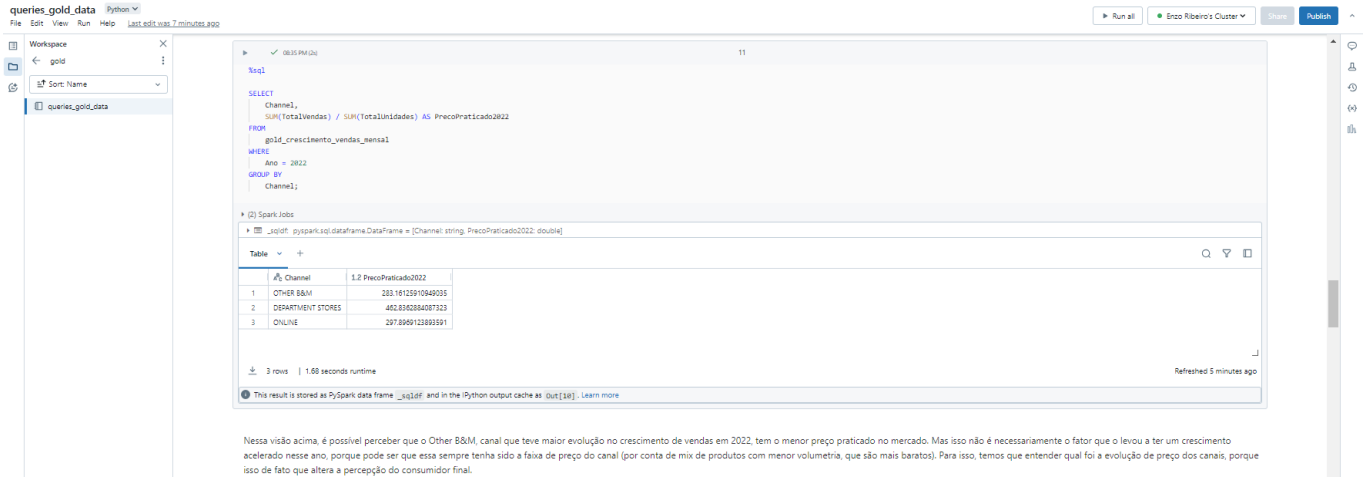
Com essa informação, fica a dúvida se de fato o crescimento nesse ano veio do canal físico (Other B&M e Dept Stores) ou do Online, vamos montar uma consulta para entender esse detalhe.



Pegunta 2: Qual canal que mais cresceu nesse ano?

Resposta: O canal Other B&M foi o que mais cresceu, evoluindo +46,1%. Isso faz sentido porque esse canal é composto pelas lojas Perfumarias Físicas e pela Sephora, que estiveram fechadas em diversos períodos de 2021 e voltaram a ficar 100% abertas em 2022, com parte do público voltando a frequentar shoppings após um longo período sem poder sair de casa.

Vamos investigar se o crescimento acelerado do Other B&M versus os outros tem algo a ver com o preço praticado nele nesse ano ou se foi apenas um comportamento do mercado de acordo com fatores mais qualitativos, como o pós pandemia.



Nessa visão acima, é possível perceber que o Other B&M, canal que teve maior evolução no crescimento de vendas em 2022, tem o menor preço praticado no mercado. Mas isso não é necessariamente o fator que o levou a ter um crescimento acelerado nesse ano, porque pode ser que essa sempre tenha sido a faixa de preço do canal (por conta de mix de produtos com menor volumetria, que são mais baratos). Para isso, temos que entender qual foi a evolução de preço dos canais, porque isso de fato que altera a percepção do consumidor final.

queries_gold_data Python

File Edit View Run Help Last edit was 7 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

13

```
%sql

SELECT
    AnoAtual,
    Channel,
    PrecoPraticado,
    LAG(PrecoPraticado, 1, 0) OVER (PARTITION BY Channel ORDER BY AnoAtual) AS PrecoPraticadoAnoAnterior,
    COALESCE(((PrecoPraticado - LAG(PrecoPraticado, 1, 0) OVER (PARTITION BY Channel ORDER BY AnoAtual)) / LAG(PrecoPraticado, 1, 0) OVER (PARTITION BY Channel ORDER BY AnoAtual)) * 100, 0) AS EvolucaoPercentual
FROM (
    SELECT
        Ano AS AnoAtual,
        Channel,
        SUM(TotalVendas) / SUM(TotalUnidades) AS PrecoPraticado
    FROM
        gold_crescimento_vendas_mensal
    WHERE
        Ano IN (2021, 2022)
    GROUP BY
        Ano, Channel
) AS PrecosPorAnoECanal
ORDER BY
    Channel, AnoAtual;
```

(3) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [AnoAtual: integer, Channel: string ... 3 more fields]

Table

	AnoAtual	Channel	1.2 PrecoPraticado	1.2 PrecoPraticadoAnoAnterior	1.2 EvolucaoPercentual
1	2021	DEPARTMENT STORES	401.7293696362336	0	0
2	2022	DEPARTMENT STORES	462.8362884087323	401.7293696362336	15.210966235262072
3	2021	ONLINE	257.66072518457753	0	0
4	2022	ONLINE	297.89691238993591	257.66072518457753	15.615956671688332
5	2021	OTHER B&M	264.49746511299475	0	0
6	2022	OTHER B&M	283.16125910949035	264.49746511299475	7.056322444724499

6 rows | 2.10 seconds runtime

Refreshed 6 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[11]`. Learn more

Com essa visão, vemos que o canal Department Stores aumentou em +15,2% seu preço, o Online aumentou em +15,6% e Other B&M aumentou apenas em +7,0%. Isso pode ter sido uma estratégia desse canal para atrair consumidores dos outros canais, o que possivelmente deu certo, tendo em vista que foi o canal de vendas que mais cresceu no mercado de luxo nesse ano.

Pergunta 3: Isso se deu por uma mudança no comportamento do preço praticado nesse canal versus os outros?

Resposta: Sim, esse canal foi o que teve o menor aumento de preço de um ano para o outro.

queries_gold_data Python

File Edit View Run Help Last edit was 8 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

16

```
%sql

CREATE OR REPLACE TABLE gold_crescimento_vendas_mensal AS

WITH VendasMensais AS (
    SELECT
        year(Date) AS Ano,
        date_format(Date, 'MMMM') AS Mes,
        Category,
        Subcategory,
        SUM(SellOut_Value) AS TotalVendas
    FROM silver_database.silver_mercado
    WHERE Channel = 'OTHER B&M'  -- Filtro adicionado
    GROUP BY Ano, Mes, Category, Subcategory
)

SELECT
    m.Ano,
    m.Mes,
    m.Category,
    m.Subcategory,
    m.TotalVendas,
    LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Category ORDER BY m.Ano, m.Mes) AS VendasMesAnterior,
    (m.TotalVendas - LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Category ORDER BY m.Ano, m.Mes)) AS DiferencaMesAnterior,
    ((m.TotalVendas - LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Category ORDER BY m.Ano, m.Mes)) / LAG(m.TotalVendas, 1, 0) OVER (PARTITION BY m.Category ORDER BY m.Ano, m.Mes)) * 100 AS CrescimentoPercentual
FROM VendasMensais m;
```

(7) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]

Query returned no results

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[12]`. Learn more

queries_gold_data Python

File Edit View Run Help Last edit was 8 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

17

```
%sql

SELECT
    AnoAtual,
    Category,
    VendasAnuais,
    COALESCE((VendasAnuais - VendasAnoAnterior) / VendasAnoAnterior * 100, 0) AS CrescimentoPercentual
FROM (
    SELECT
        Ano AS AnoAtual,
        Category,
        SUM(TotalVendas) AS VendasAnuais,
        LAG(SUM(TotalVendas), 1, 0) OVER (PARTITION BY Category ORDER BY Ano) AS VendasAnoAnterior
    FROM
        gold_crescimento_vendas_mensal
    GROUP BY
        Ano, Category
) AS VendasPorAnoECanal
WHERE AnoAtual = 2022  -- Filtro para o ano de 2022
ORDER BY
    AnoAtual, CrescimentoPercentual DESC;
```

(4) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [AnoAtual: integer, Category: string ... 2 more fields]

Table

	AnoAtual	Category	1.2 VendasAnuais	1.2 CrescimentoPercentual
1	2022	MAKE UP	602667904	62.92525773819469 5
2	2022	WOMEN FRAGRANC...	427489498	39.455381875241 75
3	2022	MEN FRAGRANCES	270436898	36.969826701249 06
4	2022	SKINCARE	150027805	26.366143876196794

4 rows | 3.30 seconds runtime

Refreshed 6 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[13]`. Learn more

Pergunta 4: Qual a categoria que mais cresceu nesse canal?

Resposta: Todas as categorias cresceram bastante dentro do Other B&M nesse ano, mas a que puxou o crescimento para cima foi Make Up = Maquiagem, crescendo +62,9% versus 2021. Isso deve ter acontecido porque como em 2021 as pessoas saiam pouco de casa e quando saiam usavam mascara, as vendas dessa categoria desacelerou.

queries_gold_data Python

File Edit View Run Help Last edit was 2 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

2 minutes ago (3s) 19 SQL

```sql
WITH Vendas2021 AS (
    SELECT
        Category,
        Subcategory,
        SUM(SellOut_Value) AS TotalVendas2021
    FROM
        silver_database.silver_mercado
    WHERE
        YEAR(Date) = 2021
        AND Channel = 'OTHER B&M'
        AND Category = 'MAKE UP'
    GROUP BY
        Category,
        Subcategory
),
Vendas2022 AS (
    SELECT
        Category,
        Subcategory,
        SUM(SellOut_Value) AS TotalVendas2022
    FROM
        silver_database.silver_mercado
    WHERE
        YEAR(Date) = 2022
        AND Channel = 'OTHER B&M'
        AND Category = 'MAKE UP'
    GROUP BY
        Category,
        Subcategory
)
SELECT
    v2021.Category,
    v2021.Subcategory,
    COALESCE(v2021.TotalVendas2021, 0) AS TotalVendas2021,
    COALESCE(v2022.TotalVendas2022, 0) AS TotalVendas2022,
    COALESCE(((v2022.TotalVendas2022 - v2021.TotalVendas2021) / NULLIF(v2021.TotalVendas2021, 0)) * 100, 0) AS CrescimentoPercentual -- Calcula o crescimento
FROM
    Vendas2021 v2021
LEFT JOIN
    Vendas2022 v2022 ON v2021.Category = v2022.Category AND v2021.Subcategory = v2022.Subcategory
ORDER BY
    CrescimentoPercentual DESC; -- Ordenando do maior para o menor crescimento
```

(4) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [Category: string, Subcategory: string ... 3 more fields]

Table

	Category	Subcategory	1.2 TotalVendas2021	1.2 TotalVendas2022	1.2 CrescimentoPercentual
1	MAKE UP	LIP GLOSS	7949304	23776375	199.1000847397823
2	MAKE UP	MIXED LIPS	1696654	5074406	199.0866621793504
3	MAKE UP	LIP LINER	1886036	5133663	172.1932667244951
4	MAKE UP	LIPSTICK	37849891	75447572	99.33366726203513

queries_gold_data Python

File Edit View Run Help Last edit was 3 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

3 minutes ago (2s) 20 SQL

```sql
SELECT
    Category,
    Subcategory,
    SUM(SellOut_Value) / SUM(SellOut_Units) AS PrecoMedioVenda
FROM
    silver_database.silver_mercado
WHERE
    YEAR(Date) = 2022
    AND Channel = 'OTHER B&M'
    AND Category = 'MAKE UP'
GROUP BY
    Category,
    Subcategory
ORDER BY
    PrecoMedioVenda ASC; -- Ordenando do maior para o menor crescimento;
```

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [Category: string, Subcategory: string ... 1 more field]

Table

	Category	Subcategory	1.2 PrecoMedioVenda
1	MAKE UP	MAKE UP ACCESSORI...	115.71575286490021
2	MAKE UP	LIP LINER	116.00187549429442
3	MAKE UP	LIPSTICK	131.49193159096243
4	MAKE UP	LIP GLOSS	143.64653818269696
5	MAKE UP	EYE PRIMER	151.70805739951435
6	MAKE UP	EYE LINER	154.93113091784224
7	MAKE UP	BROWS	156.14794933346822
8	MAKE UP	NAILS	157.32043204320433
9	MAKE UP	MASCARAS	162.51360723116152
10	MAKE UP	SETTING SPRAY	173.83202940665126
11	MAKE UP	PLUMPERS	174.61231959406064
12	MAKE UP	MIXED FACE	187.21629892092125
13	MAKE UP	CONCEALER	189.53672298723018
14	MAKE UP	EYE SHADOW	193.80744198701197
15	MAKE UP	BLUSH	194.34746722770674

23 rows | 1.80 seconds runtime Refreshed 2 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[30]`. Learn more

Pergunta 5: Tem alguma subcategoria que puxou a evolução nesse canal dentro da categoria mais acelerada? Qual o seu preço de venda médio?

Resposta: Na visão acima, vemos que a subcategoria que mais cresceu em 2022 dentro de Maquiagem no Other B&M foi Lip Gloss, que cresceu +199%. Em seguida, temos as subcategorias de Mixed Lips, Lip Liner e Lipstick. O que chama atenção é que a categoria de Maquiagem abrange Face, Olhos e Lábios, mas nesse ano as top 4 subcategorias de evolução no mercado são referente a Lips = Lábios. Isso faz sentido, porque em 2021, com o uso constante de máscara, os consumidores pararam de passar batom e produtos para os lábios, tendência que voltou ao mercado no fim da pandemia com forte evolução após um ano sem contra-cifra.

Além disso, observa-se que Lip Gloss tem um preço médio de venda de 143 reais, o que é considerado acessível dentro do mercado de luxo e uma das subcategorias de Maquiagem mais baratas, o que facilita o recrutamento de novas consumidoras.

queries_gold_data Python

File Edit View Run Help Last edit was 3 minutes ago

Run all | Enzo Ribeiro's Cluster | Share | Publish

Workspace

gold

Sort: Name

queries_gold_data

3 minutes ago (2s) 20 SQL

```sql
SELECT
    Category,
    Subcategory,
    SUM(SellOut_Value) / SUM(SellOut_Units) AS PrecoMedioVenda
FROM
    silver_database.silver_mercado
WHERE
    YEAR(Date) = 2022
    AND Channel = 'OTHER B&M'
    AND Category = 'MAKE UP'
GROUP BY
    Category,
    Subcategory
ORDER BY
    PrecoMedioVenda ASC; -- Ordenando do maior para o menor crescimento;
```

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [Category: string, Subcategory: string ... 1 more field]

Table

	Category	Subcategory	1.2 PrecoMedioVenda
1	MAKE UP	MAKE UP ACCESSORI...	115.71575286490021
2	MAKE UP	LIP LINER	116.00187549429442
3	MAKE UP	LIPSTICK	131.49193159096243
4	MAKE UP	LIP GLOSS	143.64653818269696
5	MAKE UP	EYE PRIMER	151.70805739951435
6	MAKE UP	EYE LINER	154.93113091784224
7	MAKE UP	BROWS	156.14794933346822
8	MAKE UP	NAILS	157.32043204320433
9	MAKE UP	MASCARAS	162.51360723116152
10	MAKE UP	SETTING SPRAY	173.83202940665126
11	MAKE UP	PLUMPERS	174.61231959406064
12	MAKE UP	MIXED FACE	187.21629892092125
13	MAKE UP	CONCEALER	189.53672298723018
14	MAKE UP	EYE SHADOW	193.80744198701197
15	MAKE UP	BLUSH	194.34746722770674

23 rows | 1.80 seconds runtime Refreshed 2 minutes ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[30]`. Learn more

Pergunta 5: Tem alguma subcategoria que puxou a evolução nesse canal dentro da categoria mais acelerada? Qual o seu preço de venda médio?

Resposta: Na visão acima, vemos que a subcategoria que mais cresceu em 2022 dentro de Maquiagem no Other B&M foi Lip Gloss, que cresceu +199%. Em seguida, temos as subcategorias de Mixed Lips, Lip Liner e Lipstick. O que chama atenção é que a categoria de Maquiagem abrange Face, Olhos e Lábios, mas nesse ano as top 4 subcategorias de evolução no mercado são referente a Lips = Lábios. Isso faz sentido, porque em 2021, com o uso constante de máscara, os consumidores pararam de passar batom e produtos para os lábios, tendência que voltou ao mercado no fim da pandemia com forte evolução após um ano sem contra-cifra.

Além disso, observa-se que Lip Gloss tem um preço médio de venda de 143 reais, o que é considerado acessível dentro do mercado de luxo e uma das subcategorias de Maquiagem mais baratas, o que facilita o recrutamento de novas consumidoras.

Com isso, conseguimos responder todas as business questions e fazer um deep dive completo nos dados de mercado disponibilizados. Com o seguinte resultado, descrito nos prints acima do Databricks:

```
Pegunta 1: Qual foi o ano que teve maior evolução de vendas nos ultimos anos?
Resposta: 2022, que foi o ano que o comércio voltou a abrir 100% pós pandemia, e o mercado de cosméticos de Luxo cresceu +34%
```

```
Pegunta 2: Qual canal que mais cresceu nesse ano?
Resposta: O canal Other B&M foi o que mais cresceu, evoluindo +46,1%. Isso faz sentido porque esse canal é composto pelas lojas Perfumarias Físicas e pela Sephora, que estiveram fechadas em diversos períodos de 2021 e voltaram a ficar 100% abertas em 2022, com parte do público voltando a frequentar shoppings após um longo período sem poder sair de casa.
```

```
Pergunta 3: Isso se deu por uma mudança no comportamento do preço praticado nesse canal versus os outros?
Resposta: Sim, esse canal foi o que teve o menor aumento de preço de um ano para o outro.
```

```
Pergunta 4: Qual a categoria que mais cresceu nesse canal?
Resposta: Todas as categorias cresceram bastante dentro do Other B&M nesse ano, mas a que puxou o crescimento para cima foi Make Up = Maquiagem, crescendo +62,9% versus 2021. Isso deve ter acontecido porque como em 2021 as pessoas saiam pouco de casa e quando saiam usavam mascara, as vendas dessa categoria desacelerou.
```

```
Pergunta 5: Tem alguma subcategoria que puxou a evolução nesse canal dentro da categoria mais acelerada? Qual o seu preço de venda médio?
Resposta: Na visão acima, vemos que a subcategoria que mais cresceu em 2022 dentro de Maquiagem no Other B&M foi Lip Gloss, que cresceu +199%. Em seguida, temos as subcategorias de Mixed Lips, Lip Liner e Lipstick. O que chama atenção é que a categoria de Maquiagem abrange Face, Olhos e Lábios, mas nesse ano as top 4 subcategorias de evolução no mercado são referente a Lips = Lábios. Isso faz sentido, porque em 2021, com o uso constante de máscara, os consumidores pararam de passar batom e produtos para os lábios, tendência que voltou ao mercado no fim da pandemia com forte evolução após um ano sem contra-cifra.
Além disso, observa-se que Lip Gloss tem um preço médio de venda de 143 reais, o que é considerado acessível dentro do mercado de luxo e uma das subcategorias de Maquiagem mais baratas, o que facilita o recrutamento de novas consumidoras.
```

```
Pergunta 6: Qual mês com maior venda dessa subcategoria nesse ano dentro do canal analisado?
Resposta: O mês com maior venda na subcategoria foi o mês 12 e em seguida o mês 11. Isso deixa claro que o Natal e a Black Friday são sazonalidades importantes para esse mercado, refletindo em ser os 2 maiores meses nas vendas de Lip Gloss dentro do Other B&M, em um ano de forte evolução da categoria de maquiagem no canal.
```

Conclui-se, portanto, que o MVP foi muito interessante para desenvolver novos skills na área de Engenharia de Dados e também fomentar uma análise questionadora sobre dados que já estava acostumado a olhar no dia a dia no trabalho. Obrigado a todos os professores dessa disciplina pelos ensinamentos e pelo acompanhamento na orientação do MVP, estou saindo dessa disciplina como um profissional mais preparado para o mercado de trabalho.