Technical Challange Autoforce | eCommerce Events Analysis Report (*updated*)

**Consumer Behavior and Product Performance in Online Cosmetics Store (Oct. 2019 - Feb. 2020)*

Overview

This report presents a comprehensive analysis of consumer behavior and product performance in an online cosmetics store, using data collected from October 2019 to February 2020. Each section outlines the objectives of the analysis, includes the SQL query used, and discusses how the results can be utilized along with associated usage.

The analyses are categorized based on their objectives and the insights they provide. The insights gained from these analyses can drive strategic decision-making in marketing, sales, inventory management, and customer relationship management.

a. Objective

This report presents an in-depth analysis of consumer behavior and product performance in the cosmetics e-commerce, utilizing the **eCommerce Events History in Cosmetics Shop** dataset from **Kaggle**. The this files contains behavior data for 5 months **(Oct 2019 – Feb 2020)** from a medium cosmetics online store, and are focus to provide insights for marketing and sales strategies, with an emphasis on *Business Intelligence* (BI) analytics.

b. Dataset

The complete dataset contains user event records, including product views, additions to cart, removals, and purchases. Each row in the file represents an event. All events are related to products and users. Each event is like many-to-many relation between products and users.

The downloadable dataset comprises five ".csv" files, each one with *9 columns* and related to a specific month, as mentioned, spanning from *October 2019 to January 2020.*

- The *table description* follows this structure and organization:

Property	Description	Type
`event_time`	Time when event happened at (in UTC).	*time*
`event_type`	Four kinds of event: view, cart, purchase, remove from cart	*string*
`product_id`	ID of a product.	*integer*
`category_id`	Product's category ID.	*integer*
`category_code`	Product's category taxonomy, code name (can be missing).	*string*
`brand`	Downcased string of brand name (can be missing).	*string*
`price`	Float price of a product. Present.	*float*
`user_id`	Permanent user ID.	*integer*
`user_session`	Temporary user's session ID. Changed after pauses.	*string*

After downloading all the data from Kaggle, a data engineering stage was set up for analysis. The data was then transferred to a **Google Cloud Storage** bucket in Lakehouse, which was created specifically for this challenge. Subsequently, this dataset underwent extensive manipulation and transformation within a **BigQuery Studio** project, aptly named *Technical Challenge Autoforce*.

- ***Update 27.12:*** In a new phase of the *Technical Challenge*, to fill in missing information about the category and types of products in this dataset, we were asked to cross-reference product identifiers with the Makeup API to obtain data about products and their respective categories. An application was developed in `Node.js` in the GCP environment, using **Google Cloud Function** to retrieve the dataset from the endpoint products.json and convert it into CSV format.

Using **Google Cloud** tools and servers is the best option because it combines the resources of a *data lake* and a *data warehouse* into a single platform for storing, processing, and analyzing structured and unstructured data. This integration facilitates efficient storage, processing, and analysis of diverse datasets, encompassing both structured and unstructured data forms. Moreover, **Google Cloud's** seamless interoperability with **Looker Studio** significantly enhances its utility, enabling the creation of sophisticated, insightful dashboards essential for the visualization and interpretation of critical business intelligence.

c. Methodology

To conduct the analyses, such as **Business Intelligence Analysis**, as mentioned earlier, the following steps were required:

1. Setting up a secure data storage environment (*data lake*) for the Tables.
2. Uploading the 5 tables stored in the data lake through the *Technical Challenge Autoforce* project to be utilized and managed by the `dataset_cs_ecommerce` data warehouse in **BigQuery**.

3. Selecting and querying data using standard SQL language, making use of BigQuery's robust analytics engine to extract actionable insights, thus aligning with the goals of the technical challenge.
4. Utilize the **Google Colab** tool to access this dataset and conduct a more comprehensive descriptive analysis. This enabled not only *exploratory analysis* of the provided dataset, but also the identification of *trends*, *patterns* and the generation of initial *business insights*.

Therefore, SQL was as used for initial queries and Python for more complex analyses and some first data visualizations.

- Additional information on data engineering can be found in the documentation located in the `1.data_engineering` subdirectory of this **Technical Challange project**.

Exploratory Data Analysis (EDA) | Python for Powerful Descriptions

Descriptive Analysis

Enhanced analytical depth and precision of your dataset was eminently achieved. We could encapsulate and elucidate various facets of our data, encompassing the most frequently viewed, added to cart, or purchased products, the zenith of shopping activities, the mean pricing of products, and the prevalent brands.

Given the intricate nature and complexity of the dataset, we had opted to undertake a specialized descriptive analysis focusing on consumer behavior and product performance within the cosmetics e-commerce sector. This analysis was facilitated through **Google Colab**, utilizing the robustness of *Python* and its extensive range of libraries. This approach enabled us to harness the full potential of the data, thereby deriving meaningful insights and strategic business intelligence.

- ***Update 27.12:*** For the next phase of the *Technical Challenge*, a random correlation of product identifiers was performed between the datasets, establishing a mapping between the `id` of the new **Makeup API** set and the `product_id` of the **eCommerce Events History in Cosmetics Shop** set. This resulted in a random correspondence between the identifiers, meaning each `id` in the **Makeup API** set will be linked to a random `product_id` in the **eCommerce Events History in Cosmetics Shop** set.

Location of Analysis:

- The files `.ipynb` and `.pdf` containing these analyses are located in the `2.data_analysis` subdirectory of this Project. Alternatively, the **Google Colab** notebook can be accessed directly via this Link here.

Business Intelligence Analysis | SQL Queries for Custom Insights

This analyze presents a comprehensive analysis of consumer behavior and product performance in an online cosmetics store, covering the *period from October 2019 to February 2020*. By leveraging advanced analytics techniques, the report aims to unearth patterns, trends, and actionable insights from complex datasets.

In the dynamic realm of online retail, the role of a marketing business intelligence analyst is pivotal in steering strategic decisions through data-driven insights.

Such analyses are indispensable for a marketing business intelligence analyst, as they illuminate aspects like customer preferences, purchasing behavior, product popularity, and overall market dynamics. The insights derived from this report are expected to inform crucial marketing strategies, optimize customer engagement, and drive business growth, ensuring that the company remains competitive and responsive to consumer needs in a constantly evolving digital marketplace.

- ***Update 27.12:*** All new queries used in the new stage of the challenge, with the crossing of product category information, are the next section to the queries already in the first study.

Below are selected **SQL queries** focusing on key metrics and some of the same analyses already performed in **Python**.

1. Data Types and Missing Values

a. Column Data Types

Objective: To understand the structure of the dataset by examining the data types of each column. **SQL Query**:

```
 SELECT column_name, data_type
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'all_events_2019_to_2020'
```

Usage: This information is essential for data preprocessing and ensuring that the appropriate data types are used for analysis.

b. Missing Values by Column

Objective: Identifying columns with missing data, crucial for data cleaning and integrity checks. **SQL Query**:

```
SELECT
    COUNT(*) - COUNT(event_time) AS missing_event_time,
    COUNT(*) - COUNT(event_type) AS missing_event_type,
    COUNT(*) - COUNT(product_id) AS missing_product_id,
    COUNT(*) - COUNT(category_id) AS missing_category_id,
    COUNT(*) - COUNT(category_code) AS missing_category_code,
    COUNT(*) - COUNT(brand) AS missing_brand,
    COUNT(*) - COUNT(price) AS missing_price,
    COUNT(*) - COUNT(user_id) AS missing_user_id,
    COUNT(*) - COUNT(user_session) AS missing_user_session
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
```

Usage: Helps in deciding whether to fill, ignore, or drop missing values, impacting the accuracy of further analysis.

2. Descriptive Statistics and Event Distribution

a. Descriptive Statistics for Price Column

Objective: Provides summary statistics for product prices. **SQL Query**:

```
SELECT
    AVG(price) AS mean_price,
    MIN(price) AS min_price,
    MAX(price) AS max_price,
    STDDEV(price) AS stddev_price
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
```

Usage: Useful in understanding pricing strategy and identifying outliers or abnormal pricing structures.

b. Distribution of Event Types

Objective: Analysis of different event types (e.g., views, adds to cart, purchases) in the dataset. **SQL Query**:

```
SELECT event_type, COUNT(*) AS frequency
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`

GROUP BY event_type
```

Usage: Helps in understanding user behavior and the popularity of different types of interactions with the products.

3. Product and User Diversity

a. Diversity of Products, Users, and Categories

Objective: Insights into the range and variety of products, users, and product categories. **SQL Query**:

```
SELECT
    COUNT(DISTINCT product_id) AS unique_products,
    COUNT(DISTINCT user_id) AS unique_users,
    COUNT(DISTINCT category_id) AS unique_categories
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
```

Usage: This analysis can guide inventory and category management based on user and product diversity.

b. Events by Category

Objective: Frequency of events segmented by product categories. **SQL Query**:

```
 SELECT category_id,
    SUM(CASE WHEN event_type = 'view' THEN 1 ELSE 0 END) AS views,
    SUM(CASE WHEN event_type = 'cart' THEN 1 ELSE 0 END) AS carts,
    SUM(CASE WHEN event_type = 'purchase' THEN 1 ELSE 0 END) AS purchases,
    SUM(CASE WHEN event_type = 'remove_from_cart' THEN 1 ELSE 0 END) AS removed_from_cart
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`

GROUP BY category_id
```

Usage: Important for understanding which categories are most engaging or need more marketing efforts.

4. Brand Analysis

a. Events by Brand

Objective: Analyzing user interaction (views, purchases, etc.) with different brands. **SQL Query**:

```
 SELECT brand,
    SUM(CASE WHEN event_type = 'view' THEN 1 ELSE 0 END) AS views,
    SUM(CASE WHEN event_type = 'cart' THEN 1 ELSE 0 END) AS carts,
    SUM(CASE WHEN event_type = 'purchase' THEN 1 ELSE 0 END) AS purchases,
    SUM(CASE WHEN event_type = 'remove_from_cart' THEN 1 ELSE 0 END) AS removed_from_cart
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`

WHERE brand IS NOT NULL
GROUP BY brand
```

Usage: Provides insights into brand popularity and user preferences, informing brand-related marketing strategies.

b. Average Price by Brand

Objective: Insight into the pricing strategy of different brands. **SQL Query**:

```
 SELECT brand, AVG(price) AS average_price
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`

WHERE brand IS NOT NULL
GROUP BY brand
```

Usage: Helps in understanding how brand positioning affects pricing and consumer perception.

5. Conversion Rates and Product Popularity

a. Conversion Rate from Views to Purchases

Objective: Measures the effectiveness of turning product views into actual purchases. **SQL Query**:

```
 SELECT total_purchases / total_views * 100.0 AS conversion_rate
FROM views, purchases
```

Usage: Crucial for evaluating the effectiveness of product presentation and marketing strategies.

b. Top 10 Most Viewed Products

Objective: Identifies the most popular products based on views. **SQL Query**:

```
 SELECT product_id, COUNT(*) AS total_views
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`

WHERE event_type = 'view'
GROUP BY product_id
ORDER BY total_views DESC
LIMIT 10
```

Usage: Useful for inventory planning and prioritizing marketing efforts on popular products.

6. Shopping Cart and User Session Patterns

a. Shopping Cart Activity

Objective: Analyzing user behavior in terms of adding to and removing from the cart. **SQL Query**:

```
 SELECT event_type, COUNT(*) AS cart_events
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
WHERE event_type IN ('cart', 'remove_from_cart')
GROUP BY event_type
```

Usage: Offers insights into consumer decision-making and potential pain points in the purchasing process.

b. User Session Patterns

Objective: Insights into user engagement and session durations. **SQL Query**:

```
 SELECT user_session, COUNT(DISTINCT user_id) AS unique_users, COUNT(*) AS total_events, SUM(CASE WHEN event_type = 'purchase' THEN 1
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
GROUP BY user_session
ORDER BY total_events DESC
LIMIT 10
```

Usage: Useful for understanding user engagement patterns and optimizing the website for better user experience.

7. User Interaction and Engagement

a. Popular Product Categories by User Interaction

Objective: Identifies which categories engage users the most. **SQL Query**:

```
 SELECT category_code, COUNT(DISTINCT user_id) AS unique_users, COUNT(*) AS total_interactions
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
WHERE event_type IN ('view', 'cart', 'purchase')
GROUP BY category_code
ORDER BY total_interactions DESC
LIMIT 10
```

Usage: Key for understanding which product categories are most engaging to users, informing category-specific marketing and product placement strategies.

b. Weekly User Engagement

Objective: Analysis of user activity patterns over a week. **SQL Query**:

```
 SELECT EXTRACT(WEEK FROM event_time) AS week, COUNT(DISTINCT user_id) AS weekly_users
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
GROUP BY week
ORDER BY week
```

Usage: Vital for planning weekly marketing campaigns and understanding when users are most active.

c. Net Promoter Score (NPS) Calculation

Objective: A measure of customer loyalty and satisfaction. **SQL Query**:

```
 SELECT promoters, detractors, (promoters - detractors) AS nps
FROM NPS_CTE
```

Usage: Crucial for measuring customer satisfaction and loyalty, guiding improvements in customer service and product offerings.

---

8. Purchase Consumer Behavior and Trends

a. Average Time to Purchase

Objective: Time taken from viewing to purchasing a product. **SQL Query**:

```
 SELECT DATE(first_interaction_time) AS interaction_date,
       COUNT(*) AS users,
       AVG(time_to_purchase_minutes) AS avg_time_to_purchase
FROM TimeToPurchaseCTE
GROUP BY interaction_date
ORDER BY interaction_date
```

Usage: Offers insights into the efficiency of the sales funnel and user decision-making process.

b. Top Brands and Products Purchased

Objective: Identifies the most purchased brands and products. **SQL Query**:

```
 SELECT brand, COUNT(*) AS total_compras
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
WHERE event_type = 'purchase'
GROUP BY brand
ORDER BY total_compras DESC
```

Usage: Essential for inventory management and highlighting successful products or brands in marketing campaigns.

---

9. Cart Dynamics

a. Most Removed Items from Cart

Objective: Insights into potential issues with certain products or pricing. **SQL Query**:

```
 SELECT product_id, COUNT(*) AS total_removidos
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
WHERE event_type = 'remove_from_cart'
GROUP BY product_id
ORDER BY total_removidos DESC
LIMIT 10
```

Usage: Helps in identifying products that are often considered but not purchased, pointing to possible issues with pricing or product appeal.

b. Best Purchase-to-View Ratio Products

Objective: Efficiency of product conversion from view to purchase. **SQL Query**:

```
 SELECT product_id,
    SUM(CASE WHEN event_type = 'purchase' THEN 1 ELSE 0 END) /
    GREATEST(SUM(CASE WHEN event_type = 'view' THEN 1 ELSE 0 END), 1) AS taxa_compra_visualizacao
FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
GROUP BY product_id
ORDER BY taxa_compra_visualizacao DESC
```

Usage: Important for identifying highly effective products in terms of conversion, informing stock and marketing priorities.

---

10. Customer Lifetime Value (CLV)

Objective: To estimate the total revenue a business can expect from a single customer account throughout their relationship with the company. This metric is crucial for understanding the long-term value of customers and making informed decisions about marketing, sales, and product development.

SQL Query:

```
WITH Revenue_Per_Customer AS (
    SELECT
        user_id,
        SUM(price) AS Total_Revenue
    FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
    WHERE event_type = 'purchase'
    GROUP BY user_id
),
Average_Purchase_Value AS (
    SELECT
        AVG(Total_Revenue) AS Avg_Purchase_Value
    FROM Revenue_Per_Customer
),
Purchase_Frequency AS (
    SELECT
        user_id,
        COUNT(*) AS Num_Purchases
    FROM `technical-challenge-autoforce.dataset_cs_ecommerce.all_events_2019_to_2020`
    WHERE event_type = 'purchase'
    GROUP BY user_id
),
Average_Purchase_Frequency AS (
    SELECT
        AVG(Num_Purchases) AS Avg_Purchase_Frequency
    FROM Purchase_Frequency
),
Customer_Value AS (
    SELECT
        (SELECT Avg_Purchase_Value FROM Average_Purchase_Value) *
        (SELECT Avg_Purchase_Frequency FROM Average_Purchase_Frequency) AS Value
),
Average_Customer_Lifespan AS (
    -- This will depend on your data and might require more complex analysis.
    SELECT
        AVG(Lifespan) AS Avg_Lifespan
    FROM Customer_Lifespan_Table
)
SELECT
    (SELECT Value FROM Customer_Value) *
    (SELECT Avg_Lifespan FROM Average_Customer_Lifespan) AS CLV
```

Usage: The CLV metric is instrumental for marketing and financial forecasting. It helps in allocating marketing resources efficiently, understanding customer acquisition costs, and identifying high-value customer segments for targeted marketing campaigns. Additionally, CLV can guide decisions on customer retention strategies and product development priorities.

Market Analysis Categories with SQL Queries Using a New Dataset

As previously mentioned and thoroughly explored in **Google Colab**, once the new `products_data_makeup` table is created in the **BigQuery** project, we can utilize SQL queries to retrieve the results of the analyses conducted thus far.

To carry out an approach similar to that carried out in **Google Colab** with the creation of a dataframe containing the cross-information of the two datasets directly in **BigQuery**, using SQL, it is necessary to follow basically the same steps, but adapted to *BigQuery's SQL syntax*.

Steps to the Approach:

1. **Random Selection of IDs:** First, a set of `product_ids` is randomly selected from the `all_events_2019_to_2020` dataset, ensuring that the number of selected `IDs` matches the number of rows in the `product_data_makeup` dataset.
2. **Temporary Table Creation:** Next, you create a temporary table in **BigQuery** that would map each id from the `product_data_makeup` dataset to a random `product_id` from the `all_events_2019_to_2020` dataset.
3. **Joining Tables:** Finally, you would join (`JOIN`) the two tables using the `IDs` mapped in the temporary table, resulting in a combined table.

```
 -- Step 1: Random selection of IDs
WITH random_ids AS (
    SELECT ARRAY_AGG(product_id ORDER BY RAND() LIMIT (SELECT COUNT(*) FROM product_data_makeup)) AS ids
    FROM all_events_2019_to_2020
),
-- Step 2: Creating a temporary table with mapping
mapping_table AS (
    SELECT makeup_id, ids[OFFSET(row_number - 1)] AS mapped_product_id
    FROM (
        SELECT id AS makeup_id, ROW_NUMBER() OVER() AS row_number
        FROM product_data_makeup
    ), random_ids
    CROSS JOIN UNNEST(random_ids.ids) AS ids
)
-- Step 3: Joining the tables
SELECT m.*, n.*
FROM mapping_table
JOIN product_data_makeup m ON m.id = mapping_table.makeup_id
JOIN all_events_2019_to_2020 n ON n.product_id = mapping_table.mapped_product_id;
```

These are the same ways of querying data made in Colab with Python, but translated into SQL. Each of these following queries **provides specific insights** that **are instrumental in strategic decision-making** in areas such as marketing, sales, inventory management, and product development.

1. Product and Category Analysis

a. Distribution of Number of Purchases by Makeup Category:

Objective: To identify the most popular makeup categories based on the number of purchases.

```
 SELECT m.category, COUNT(*) AS number_of_purchases
FROM all_events_2019_to_2020 n
JOIN mapping_table map ON n.product_id = map.mapped_product_id
JOIN product_data_makeup m ON map.makeup_id = m.id
WHERE n.event_type = 'purchase'
GROUP BY m.category
ORDER BY number_of_purchases DESC;
```

Usage: This query is useful for understanding consumer preferences and demand within different makeup categories. It can help in inventory management, marketing strategy formulation, and product development.

b. Number of Unique Categories per Brand:

Objective: To find out how diverse each brand's product range is in terms of different makeup categories.

```
 SELECT m.category, COUNT(*) AS number_of_purchases
FROM all_events_2019_to_2020 n
JOIN mapping_table map ON n.product_id = map.mapped_product_id
JOIN product_data_makeup m ON map.makeup_id = m.id
WHERE n.event_type = 'purchase'
GROUP BY m.category
ORDER BY number_of_purchases DESC;
```

Usage: This query aids in assessing the product range diversity of each brand. It's valuable for market positioning analysis, competitive analysis, and for brands to identify potential areas for product line expansion.

2. Pricing and Brand Analysis

a. Average Price per Brand:

Objective: To calculate the average price of products for each makeup brand.

```
 SELECT brand, AVG(price) AS average_price
FROM product_data_makeup
GROUP BY brand
ORDER BY average_price DESC;
```

Usage: The query is crucial for pricing strategy, understanding brand positioning in the market (premium vs budget brands), and for comparative price analysis across competitors.

b. Total Orders per Brand:

Objective: To determine the total number of orders (purchases) for each brand.

```
 SELECT m.brand, COUNT(*) AS total_orders
FROM all_events_2019_to_2020 n
JOIN mapping_table map ON n.product_id = map.mapped_product_id
JOIN product_data_makeup m ON map.makeup_id = m.id
WHERE n.event_type = 'purchase'
GROUP BY m.brand
ORDER BY total_orders DESC;
```

Usage: This query helps in identifying the most popular brands in terms of sales volume. It's essential for sales analysis, forecasting, and determining marketing and sales strategies for high-performing brands.

3. Consumer Behavior Analysis

a. Distribution of Makeup Categories Across Different Event Types:

Objective: To analyze how different makeup categories are interacted with across various event types (cart, purchase, remove from cart, view).

```
 SELECT m.category, n.event_type, COUNT(*) AS count
FROM all_events_2019_to_2020 n
JOIN mapping_table map ON n.product_id = map.mapped_product_id
JOIN product_data_makeup m ON map.makeup_id = m.id
GROUP BY m.category, n.event_type;
```

Usage: This query is useful for understanding consumer engagement and behavior with different product categories. Insights from this analysis can guide marketing campaigns, user experience improvements on e-commerce platforms, and inventory management.

b. Distribution of Makeup Brands Across Different Event Types:

Objective: To explore the interaction of users with different makeup brands across various event types.

```
 SELECT m.brand, n.event_type, COUNT(*) AS count
FROM all_events_2019_to_2020 n
JOIN mapping_table map ON n.product_id = map.mapped_product_id
JOIN product_data_makeup m ON map.makeup_id = m.id
GROUP BY m.brand, n.event_type;
```

Usage: The query provides insights into brand popularity and consumer interaction patterns. It's valuable for brand performance analysis, developing targeted marketing strategies, and enhancing customer engagement techniques.

Main Observations and Insights:

- **Consumer Preferences:** Insights into popular brands, products, and categories, guiding inventory and marketing strategies.
- **Purchasing Patterns:** Understanding of when and how consumers make purchases, useful for targeting promotions.
- **Product Pricing:** Analysis of pricing strategies across categories and brands.
- **Customer Behavior:** Insights into how customers interact with the shopping cart and their session patterns.
- **Conversion Efficiency:** Identification of products and brands with high conversion rates from views to purchases.
- **Time-Based Trends:** Identification of peak times for purchasing, aiding in strategic planning for customer engagement.

Overall, this dataset is quite robust for a range of BI tasks, particularly for understanding customer behavior, product performance, and sales trends. The key will be to apply the right analytical techniques to extract meaningful insights from the data you have.

Conclusions for Business Intelligence

As a complement to the **considerations and summary** of all the insights already obtained from the *Notebook Conlab analysis*, we have the following. To access all the conclusions and recommendations derived from data analysis, with a focus on **Marketing Business Intelligence**, simply click on the following link.

As a *Business Intelligence Analyst* consulting this dataset and analyzing various queries, several key conclusions can be drawn:

Understanding Customer Behavior:

- Insights into customer interactions with the website, such as most viewed and purchased products, frequency of cart activities.
- Guides user interface improvements and personalized marketing strategies.

Product Performance Analysis:

- Analysis of product views, purchases, events by category and brand reveals product performance.
- Informs inventory management and identifies areas for promotional activities.

Pricing Strategy Insights:

- Descriptive statistics for product prices and average price analysis provide insights into pricing strategy.
- Helps in competitive positioning and pricing optimization.

Conversion Rate and Sales Funnel Effectiveness:

- Examining conversion rates assesses the effectiveness of the sales funnel.
- Pinpoints areas needing optimization to improve conversion rates.

Customer Engagement and Retention:

- Analyses like weekly user engagement, session patterns, and Net Promoter Score (NPS) highlight customer engagement and satisfaction.
- Crucial for strategies to enhance customer experience and retention.

Market Dynamics and Timing:

- Analysis of purchase behavior across times and days offers insights into market dynamics.
- Guides the timing of marketing campaigns to align with peak buying times.

Customer Lifetime Value (CLV):

- CLV calculation provides a long-term perspective on customer relationships.
- Essential for strategic planning in marketing and understanding ROI in customer-related efforts.

Strategic Business Decisions:

- Provides a comprehensive view of business performance in various areas.
- Critical for data-driven decisions in marketing, sales, customer service, and business growth.

These insights offer a multi-dimensional view of the online cosmetics store's operations, enabling data-driven decisions to optimize customer experience, improve operational efficiency, and drive business growth.