Customer Shopping Behaviour Analysis

1. Introduction

Customer behaviour analysis plays a vital role in understanding how users interact with products, services, and brands. In today's competitive retail and e-commerce environment, companies rely heavily on data-driven insights to improve customer satisfaction, increase loyalty, and maximize revenue.

This project focuses on analysing customer shopping and service behaviour using Python. The dataset includes customer demographics, purchase details, product categories, payment methods, subscription status, review ratings, and purchase frequency. The aim is to clean, transform, analyse, and store this data in a structured format to generate meaningful business insights.

The entire analysis was carried out using:

MS Excel

Python (Pandas, NumPy)

MY SQL

Power Bi

2. Dataset Overview

- Rows: 3,900

- Columns: 18

- Key Features

: - Customer demographics (Age, Gender, Location, Subscription Status)

: - Purchase details (Item Purchased, Category, Purchase Amount, Season, Size, Colour)

: - Shopping behaviour (Discount Applied, Promo Code Used, Previous Purchases, Frequency of Purchases, Review Rating, Shipping Type)

- Missing Data: 37 values in Review Rating column

3. Data Analysis Using Python

1 Data Loading: Load excel dataset of CSC format into python with the help of pandas

2 Initial Exploration: Used df.info() to check structure and .describe() for summary statistics.

```python
# help getting statistical information of numerical columns
df.describe()
```

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3863.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.750065	25.351538
std	1125.977353	15.207589	23.685392	0.716983	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.800000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

3 Missing Null Values: Checked for null values and imputed missing values in the Review rating column using the median rating of each product category.

```python
# filling null values
df['Review Rating'] = df.groupby('Category')['Review Rating'].transform(lambda x: x.fillna(x.median()))
```

4 Column Standardisation: Changed column name into snake case for better readability and documentation.

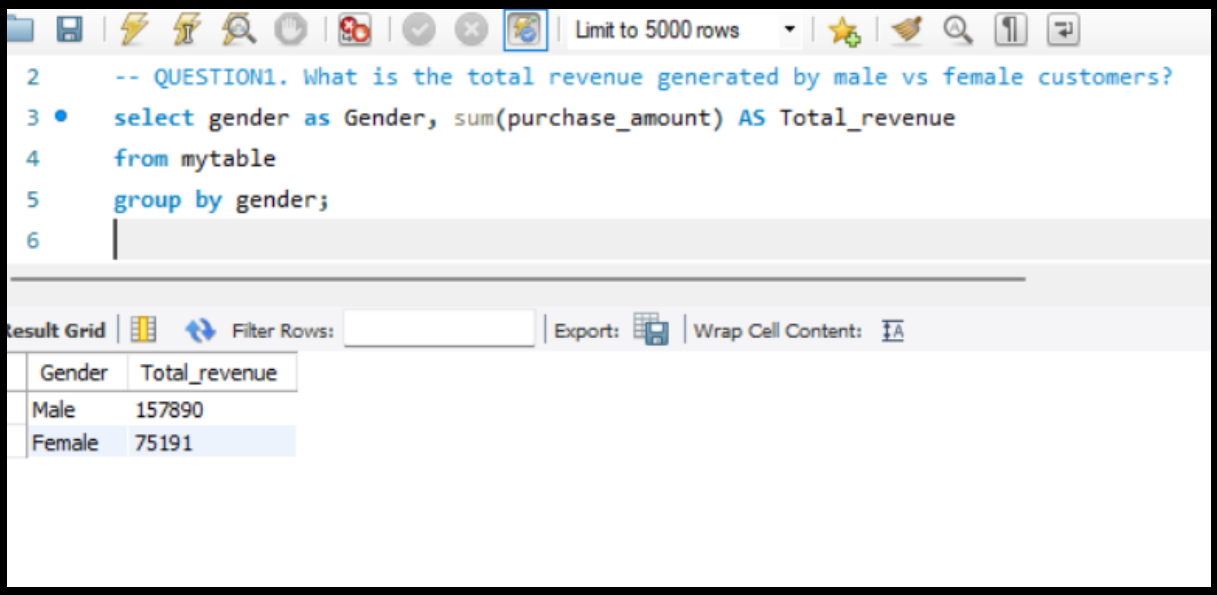
5 Remove Duplicate Columns: Verified columns “discount applied” and “promo code used” have same data and removed column “promo code used”.

6 Database Integration: Connected Python script to PostgreSQL and loaded the cleaned data frame into the database for SQL analysis.

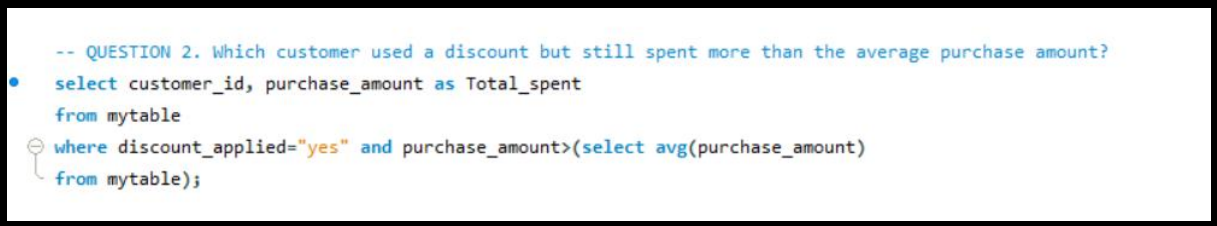
```python
from sqlalchemy import create_engine

username = "root"
password = "india%4012345"    # @ replaced with %40
host = "127.0.0.1"
port = "3306"
database = "customer_shopping"

engine = create_engine(
    f"mysql+pymysql://{username}:{password}@{host}:{port}/{database}"
)

df.to_sql("mytable", engine, if_exists="replace", index=False)
```

4. Data analysis with MY SQL

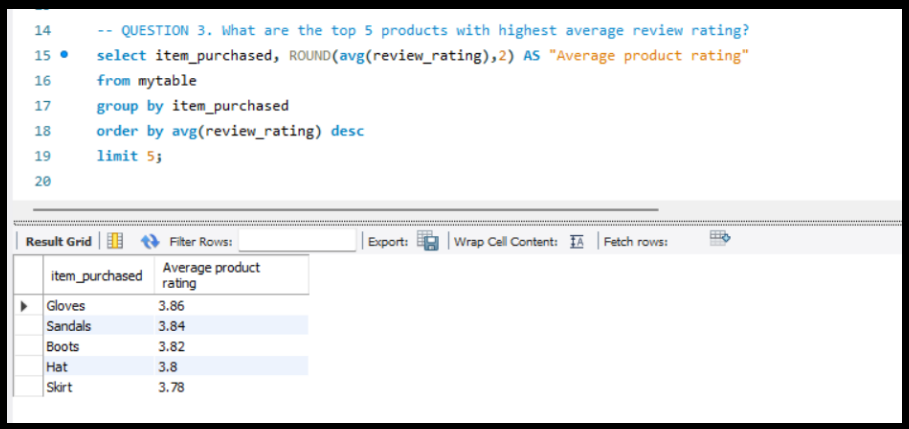
1. Total revenue by gender – Run a simple my sql query , in which we get total revenue generated by each gender.



2. High-Spending Discount Users – Identified customers who used discounts but still spent above the average purchase amount.



3. Top 5 products Identified – top 5 products with highest average review rating.



4. Shipping Type Comparision – Compare the average purchase amount between strandard and express shipping

```
21    -- QUESTION 4. Compare the average purchase amonut between strandard and express shipping ?
22    SELECT shipping_type, ROUND(avg(purchase_amount),2)
23    FROM mytable
24    WHERE shipping_type IN ("standard","express")
25    GROUP BY shipping_type;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

shipping_type	ROUND(avg(purchase_amount),2)
Express	60.48
Standard	58.46

5. Subscriber vs. Non-Subscribers – Compared average spend and total revenue across subscription status.

```
27    -- QUESTION 5. Do subsribed customers spend more? compare average spend and total revenue between subsribed and non subsribed.
28    SELECT subscription_status, COUNT(customer_id), ROUND(AVG(purchase_amount),2) AS Average_spend, SUM(purchase_amount) AS Total_spend
29    FROM mytable
30    GROUP BY subscription_status;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

subscription_status	COUNT(customer_id)	Average_spend	Total_spend
Yes	1053	59.49	62645
No	2847	59.87	170436

6. Discount-Dependent Products – Identified 5 products with the highest percentage of discounted purchases.

```
32    -- QUESTION 6. Which 5 products have the highest percentage of purchase with discount applied?
33    SELECT item_purchased, ROUND(SUM(CASE WHEN discount_applied= "yes" THEN 1 ELSE 0 END)/COUNT(*)*100 ,2) AS dis
34    FROM mytable
35    GROUP BY item_purchased
36    ORDER BY dis DESC
37    LIMIT 5;
38
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

item_purchased	dis
Hat	50.00
Sneakers	49.66
Coat	49.07
Sweater	48.17
Pants	47.37

7. Customer Segmentation – Classified customers into New, Returning, and Loyal segments based on purchase history.

```sql
-- QUESTION 7. Segment customers into new, returing and loyal based on their total number of privious purchases and show the
-- count of each segment?
WITH customer_type AS(
SELECT customer_id, case
WHEN previous_purchases = 1 THEN "New"
WHEN previous_purchases BETWEEN 2 AND 10 THEN "Returning"
ELSE "Loyal" END AS customer_segment
FROM mytable
)
SELECT customer_segment, COUNT(customer_id) AS "Number of customers"
FROM customer_type
GROUP BY customer_segment;
```

customer_segment	Number of customers
Loyal	3116
Returning	701
New	83

8. Top 3 Products per Category – Listed the most purchased products within each category.

```sql
-- QUESTION 8. What are the top 3 most purchased products with in each category?
WITH customers AS (
SELECT category, item_purchased, COUNT(*) AS total_orders, ROW_NUMBER() OVER( PARTITION BY category ORDER BY COUNT(*)) AS item_rank
FROM mytable
GROUP BY category, item_purchased
)
SELECT category, item_purchased, item_rank, total_orders
FROM customers
WHERE item_rank <=3;
```

9. Repeat Buyers & Subscriptions – Checked whether customers with >5 purchases are more likely to subscribe

```sql
-- QUESTION 9. are custmers who are repeat buyers (more than 5 privious purchase) also like to subscribe?
SELECT subscription_status, COUNT(customer_id)
FROM mytable
WHERE previous_purchases >5
GROUP BY subscription_status;
-- QUESTION 10. What are the revenue contribution in each age group?
```

subscription_status	COUNT(customer_id)
Yes	958
No	2518

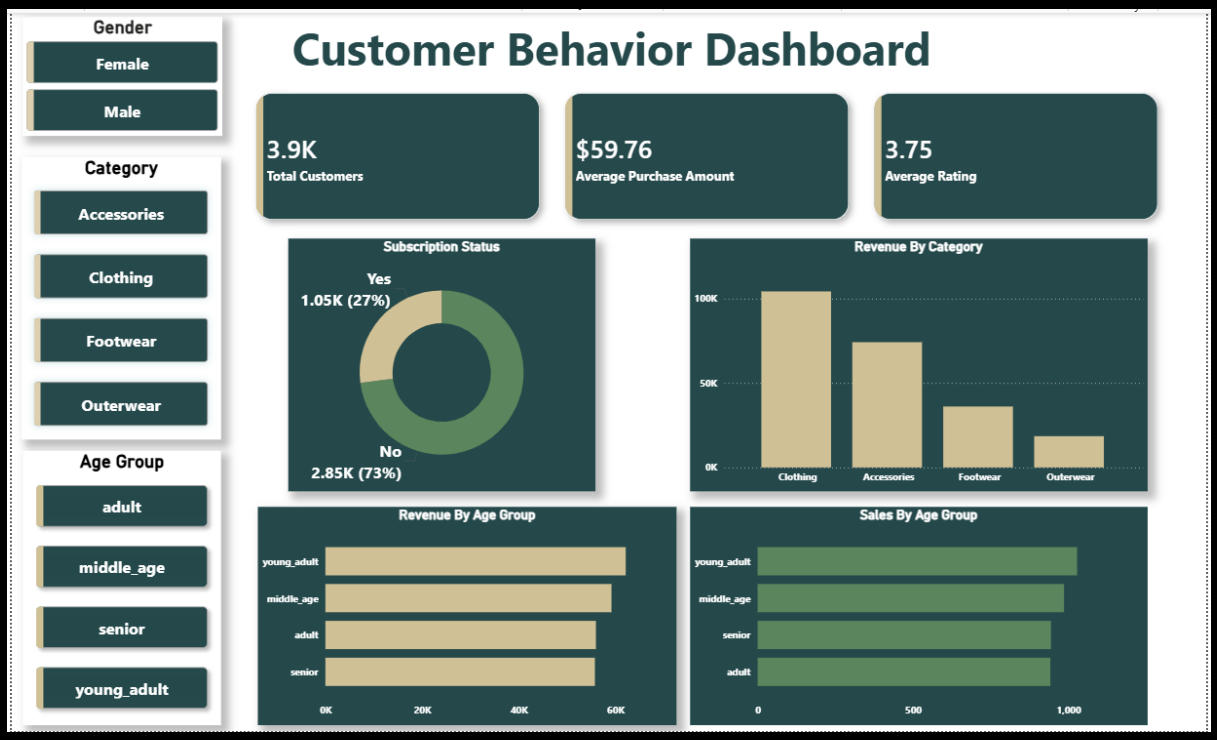
10. Revenue by Age Group – Calculated total revenue contribution of each age group

```sql
-- QUESTION 10. What are the revenue contribution in each age group?
SELECT  age_group, SUM(purchase_amount)
FROM mytable
GROUP BY age_group;
```

age_group	SUM(purchase_amount)
middle_age	59197
young_adult	62143
senior	55763
adult	55978

5. Power Bi Dashboard

Finally, we built dashboard in Power Bi to show insights visually.



6. Business Recommendations

● Boost Subscriptions – Promote exclusive benefits for subscribers.

● Customer Loyalty Programs – Reward repeat buyers to move them into the “Loyal” segment.

● Review Discount Policy – Balance sales boosts with margin control.

● Product Positioning – Highlight top-rated and best-selling products in campaigns.

● Targeted Marketing – Focus efforts on high-revenue age groups and express-shipping users