

Customer Shopping Behavior Analysis

Table of Contents

1.	<i>Project Overview</i>	3
2.	<i>Dataset Summary</i>	3
3.	<i>Exploratory Data Analysis using Python</i>	3
4.	<i>Data Analysis using SQL (Business Transactions)</i>	5
5.	<i>Dashboard In Power BI</i>	9
6.	<i>Business Recommendations</i>	10

Customer Shopping Behavior Analysis

1. Project Overview

This analytics project examines customer shopping behavior using transactional data from 3,900 purchase records across multiple product categories. The analysis focuses on identifying key trends in spending patterns, customer segmentation, product performance, and subscription behavior to generate actionable insights that support data-driven business decision-making.

2. Dataset Summary

- The dataset consists of 3,900 transaction records across 18 structured variables capturing customer demographics, purchase characteristics, and behavioral indicators.
- Key features include customer attributes such as age, gender, location, and subscription status, along with detailed purchase information covering product categories, item selection, purchase amounts, seasonal trends, sizes, and colour preferences.
- Behavioral metrics capture engagement and purchasing patterns, including discount usage, promotional activity, purchase frequency, historical buying behavior, customer review ratings, and shipping preferences.
- A small proportion of missing values was identified within the review rating variable and addressed during the data preparation phase to ensure data completeness and analytical accuracy.

3. Exploratory Data Analysis using Python

Exploratory data analysis and preprocessing were conducted using Python, primarily leveraging the pandas library to ensure data quality and analytical readiness.

- **Data Loading**

Imported the dataset using `Pandas`.

- **Initial Exploration**

Used `df.info()` to check structure and `df.describe()` for summary statistics.

```
#Summary Statisitcs
```

```
df.describe(include='all')
```

✓ [79] 53ms

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000

```
df.info()
```

✓ [78] 97ms

```
<class 'pandas.DataFrame'>
```

```
RangeIndex: 3900 entries, 0 to 3899
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	Customer ID	3900 non-null	int64
1	Age	3900 non-null	int64
2	Gender	3900 non-null	str
3	Item Purchased	3900 non-null	str
4	Category	3900 non-null	str
5	Purchase Amount (USD)	3900 non-null	int64
6	Location	3900 non-null	str
7	Size	3900 non-null	str
8	Color	3900 non-null	str
9	Season	3900 non-null	str

```
#Summary Statisitcs
```

- **Missing Data Handling**

Checked for null values and imputed missing values in the “Review Rating” column using the median rating of each product category.

- **Column Standardization**

Reamed columns to **snake case** for better readability and documentation.

- **Feature Engineering**

Created **age_group** column by binning customer ages.

Created **purchase_frequency_days** column from purchase data.

- **Data Consistency Check**

Verified if **discount_applied** and **promo_code_used** were redundant, dropped **promo_code_used**.

- **Database Integration**

Connected Python script to PostgreSQL and loaded the cleaned DataFrame into the database for SQL analysis.

Connecting This notebook to MySQL Workbench

```
from sqlalchemy import create_engine
import pandas as pd

# Connection details
username = "root"
password = "Sohaib%40123" # replace with your actual MySQL password
host = "localhost"
database = "Customer_Shopping_Behavior_Analysis" # or whatever DB you created in Workbench

# Create the engine
engine = create_engine(f"mysql+mysqlconnector://{username}:{password}@{host}/{database}")

#Loading Dataset into MySQL Workbench

table_name = "Customer"
df.to_sql(table_name, engine, if_exists='replace', index=False)

# Test the connection
```

✓ [88] 185ms

3900

4. Data Analysis using SQL (Business Transactions)

The cleaned dataset was analyzed using PostgreSQL to extract actionable business insights through structured SQL queries. The analysis focused on revenue drivers, customer behavior patterns, product performance, and subscription impact.

Key analytical objectives included:

- Evaluating total revenue distribution by gender to identify demographic spending trends

Result Grid			Filter Rows:	Search	Export:
Gender	Total Revenue				
Male	157890				
Female	75191				
Result 2					

- Identifying high-value customers who utilized discounts while maintaining above-average purchase amounts

Result Grid			Filter Rows:	Search	Export:
customer_id	purchase_amount				
2	64				
3	73				
4	90				
7	85				
9	97				
12	68				
13	72				
16	81				
20	90				
22	62				
24	88				
29	94				
32	79				
33	67				
35	91				
37	69				
40	60				
41	76				
43	100				
44	69				
55	94				
57	73				
Customer 3					
Action Output					
	Time	Action	Response		
3	12:00:15	SELECT customer_id, purchase_amount FROM Customer WHERE discount_applied = 'Yes'...	839 row(s) returned		

- Determining top-performing products based on average customer review ratings

Result Grid

Filter Rows:

Search

Export:

Fetch rows:

Top 5 Produc...

Average Review Rating

Gloves

3.86

Sandals

3.84

Boots

3.82

Hat

3.8

Skirt

3.78

Result 4

Action Output

Time

Action

Response

4

12:01:13

SELECT

item_purchased AS 'Top 5 Products',

ROUND(AVG(review_rating), 2) AS 'Average Review...

5 row(s) returned

- Comparing purchase value across shipping methods to assess fulfillment preferences

Result Grid

Filter Rows:

Search

Export:

Shipping Type	Average Purchase Amo...
Express	60.48
Standard	58.46

- Analyzing spending behavior and revenue contribution between subscribers and non-subscribers

Result Grid

Filter Rows:

Search

Export:

Subscription Stat...	Customer Count	Total Revenue	Average Purchase Amo...	
No	2847	170436	59.87	
Yes	1053	62645	59.49	

- Identifying products with the highest dependency on promotional discounts

Result Grid

Filter Rows:

Search

Export:

Fetch rows:

Products with highest P...	Discount_Rate	
Hat	50.00	
Sneakers	49.66	
Coat	49.07	
Sweater	48.17	
Pants	47.37	

Result 7

Action Output

	Time	Action	Response
<div><div></div></div> 7	12:02:41	SELECT item_purchased AS 'Products with highest POP', ROUND(100 * SUM(CASE	WH... 5 row(s) returned

- Segmenting customers into New, Returning, and Loyal groups based on historical purchase activity

Result Grid

Filter Rows:

Search

Export:

customer_segmn...	Customers_cou...	
Loyal	3116	
Returning	701	
New	83	

- Highlighting top-selling products within each product category

100% 22:99				
Result Grid				
Filter Rows:		Search	Export:	
product_ra...	category	item_purchased	Total_revenue	
1	Clothing	Blouse	10410	
2	Clothing	Shirt	10332	
3	Clothing	Dress	10320	
1	Accessories	Jewelry	10010	
2	Accessories	Sunglasses	9649	
3	Accessories	Belt	9635	
1	Outerwear	Coat	9275	
2	Outerwear	Jacket	9249	
1	Footwear	Shoes	9240	
2	Footwear	Sandals	9200	
3	Footwear	Boots	9018	
Result 9				
Action Output				
	Time	Action	Response	
9	12:04:01	with Products_rank as (select category, item_purchased, sum(purchase_amount) as Total_Revenue,...	11 row(s) returned	

- Assessing the relationship between repeat purchasing behavior and subscription adoption

100% 6:118

Result Grid Filter Rows: Search Export:

subscription_status	count_of_custom...
No	2583
Yes	980

Result 11

Action Output

	Time	Action	Response
11	12:04:47	SELECT subscription_status, COUNT(*) AS count_of_customers FROM Customer WHERE	previ... 2 row(s) returned

- Measuring revenue contribution across defined age groups

100% 31:131

Result Grid Filter Rows: Search Export:

Age Group	Total_revenue	Revenue_Contributi...
Yound Adult	62143	26.66
Middle-aged	59197	25.40
Adult	55978	24.02
Senior	55763	23.92

Result 12

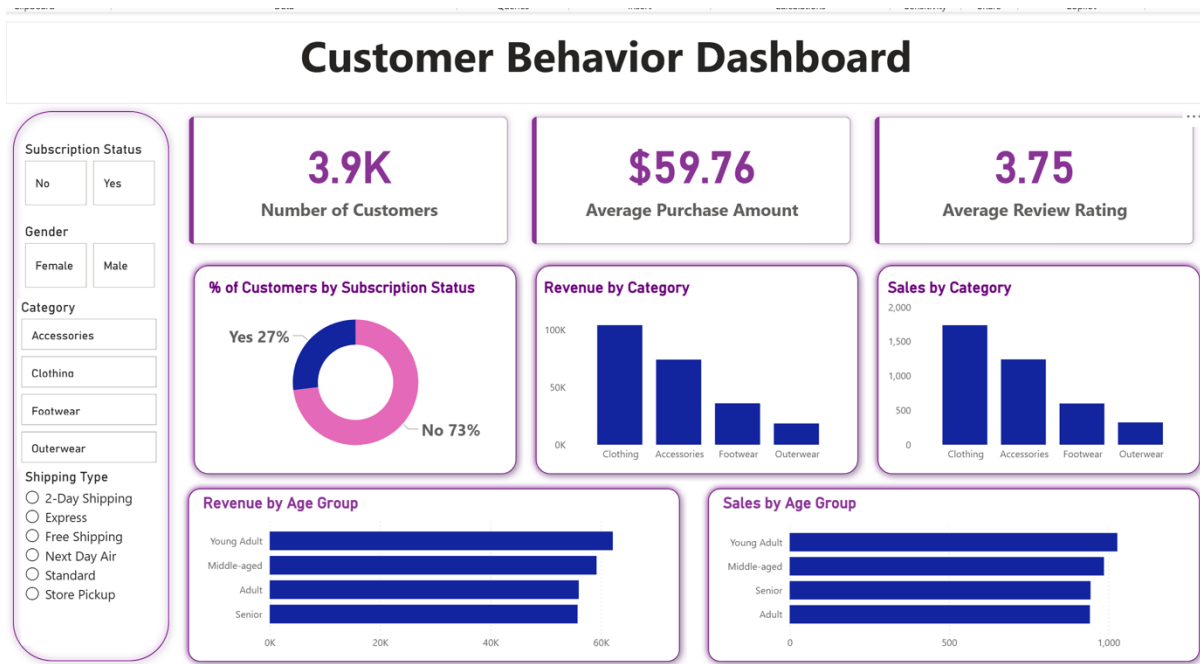
Action Output

	Time	Action	Response
12	12:05:09	SELECT age_group AS 'Age Group', SUM(purchase_amount) AS Total_revenue, ROUND(100 * S...	4 row(s) returned

5. Dashboard In Power BI

An interactive Power BI dashboard was developed to visualize key findings from the analysis, enabling intuitive exploration of customer trends, revenue performance, product insights, and segmentation outcomes.

The dashboard integrates dynamic filters and visual KPIs to support data-driven decision-making and stakeholder reporting.



6. Business Recommendations

Based on the analytical findings, several strategic initiatives were identified to improve revenue performance, customer retention, and marketing effectiveness.

- **Strengthen Subscription Programs:** Introduce exclusive incentives and personalized benefits to increase subscription adoption and recurring revenue.
- **Implement Customer Loyalty Strategies:** Develop reward-based programs targeting repeat buyers to transition customers into high-value loyal segments.
- **Optimize Discount Strategies:** Reevaluate promotional policies to balance short-term sales growth with long-term profit margins.
- **Enhance Product Positioning:** Prioritize the promotion of top-rated and high-performing products across marketing channels to maximize conversion potential.
- **Deploy Targeted Marketing Campaigns:** Focus outreach efforts on high-revenue customer segments and purchasing behaviors to improve campaign efficiency and ROI.