Customer Shopping Behaviour Analysis

1. Project Overview

In today's competitive retail environment, understanding customer purchasing behaviour is essential for improving sales performance and customer satisfaction. This project focuses on analyzing customer shopping data to identify trends related to demographics, product categories, discounts, subscriptions, and customer feedback.

The analysis was performed using Python for data processing, PostgreSQL for structured querying, and Power BI for interactive visualization.

2. Problem Statement

Retail businesses often collect large volumes of customer transaction data, but without proper analysis, this data cannot be effectively used for decision-making. The challenge is to transform raw customer shopping data into meaningful insights that help understand:

- Customer spending behaviour
- Product performance
- Impact of discounts and subscriptions
- Revenue contribution by different customer segments

3. Objectives of the Project

The main objectives of this project are:

- To clean and preprocess customer shopping data using Python
- To store and analyze the data using SQL queries in PostgreSQL
- To extract meaningful business insights from customer transactions
- To visualize trends and patterns using Power BI dashboards

4. Dataset Description

The dataset used in this project contains **3,900 customer purchase records**.

Dataset Details

- **Total Rows:** 3,900
- **Total Columns:** 18

Important Columns

- Customer ID
- Age
- Gender
- Item Purchased
- Product Category
- Purchase Amount
- Location
- Shipping Type
- Discount Applied
- Subscription Status
- Review Rating
- Previous Purchases

The dataset represents customer transactions across multiple product categories and customer segments.

5. Data Preparation and Cleaning (Python – VS Code)

5.1 Dataset Loading

The dataset was loaded into the Python environment using the Pandas library. The data consists of customer demographics, purchase behavior, and transactional details required for analysis.

```
df.head()
```

Python

```
Rows, Cols: (3900, 18)
```

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo	Fortnightly
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	Cash	Fortnightly
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	Credit Card	Weekly
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	PayPal	Weekly
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	PayPal	Annually

5.2 Data Cleaning and Preprocessing

Column names were standardized using snake case to ensure consistency. Irrelevant columns were removed, and categorical values were checked for uniformity.

```python
# Cell 3 – Clean column names (snake_case)

# Make all column names lowercase
df.columns = df.columns.str.lower()

# Replace spaces with underscores
df.columns = df.columns.str.replace(' ', '_')

# Fix specific column names if needed
df = df.rename(columns={
    'purchase_amount_(usd)': 'purchase_amount'  # your file uses this format
})

df.head()
```

Python

	customer_id	age	gender	item_purchased	category	purchase_amount	location	size	color	season	review_rating	subscription_status	shipping_type	discount_applied	promo_code_u
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	

```python
# --- Renaming columns to snake_case for PostgreSQL compatibility ---

df.columns = (
    df.columns
    .str.strip()                # remove extra spaces
    .str.lower()                # convert to lowercase
    .str.replace(' ', '_')      # replace spaces with underscore
    .str.replace('(', '', regex=False)  # remove (
    .str.replace(')', '', regex=False)  # remove )
)

# Show new column names
df.columns
```

[5]

```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'promo_code_used', 'previous_purchases',
       'payment_method', 'frequency_of_purchases'],
      dtype='object')
```

5.3 Dataset Structure and Data Types

The structure of the dataset was examined using df.info() to verify data types, non-null counts, and memory usage.

```python
print("=== df.info() ===")
df.info()
```

```
=== df.info() ===
<class 'pandas.core.frame.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   object
 3   item_purchased          3900 non-null   object
 4   category                3900 non-null   object
 5   purchase_amount         3900 non-null   int64
 6   location                3900 non-null   object
 7   size                    3900 non-null   object
 8   color                   3900 non-null   object
 9   season                  3900 non-null   object
 10  review_rating           3863 non-null   float64
 11  subscription_status     3900 non-null   object
 12  shipping_type           3900 non-null   object
 13  discount_applied        3900 non-null   object
 14  promo_code_used         3900 non-null   object
 15  previous_purchases      3900 non-null   int64
 16  payment_method          3900 non-null   object
 17  frequency_of_purchases  3900 non-null   object
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

5.4 Statistical Summary of Numerical Features

Descriptive statistics were generated using df.describe() to understand the distribution of numerical variables such as age, purchase amount, and review ratings.

```
=== df.describe() (numeric) ===
```

	customer_id	age	purchase_amount	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

5.5 Categorical Data Overview

Categorical columns such as gender, category, shipping type, and payment method were analyzed to understand customer preferences.

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

	customer_id	age	gender	item_purchased	category	purchase_amount	location	size	color	season	review_rating	sub
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	

```
=== Missing values per column ===
review_rating    37
dtype: int64
```

subscription_status	shipping_type	discount_applied	promo_code_used	previous_purchases	payment_method	frequency_of_purchases
3900	3900	3900	3900	3900.000000	3900	3900
2	6	2	2	NaN	6	7
No	Free Shipping	No	No	NaN	PayPal	Every 3 Months
2847	675	2223	2223	NaN	677	584
NaN	NaN	NaN	NaN	25.351538	NaN	NaN
NaN	NaN	NaN	NaN	14.447125	NaN	NaN
NaN	NaN	NaN	NaN	1.000000	NaN	NaN
NaN	NaN	NaN	NaN	13.000000	NaN	NaN
NaN	NaN	NaN	NaN	25.000000	NaN	NaN
NaN	NaN	NaN	NaN	38.000000	NaN	NaN
NaN	NaN	NaN	NaN	50.000000	NaN	NaN

5.6 Missing Value Analysis

Missing values were identified in the review_rating column and handled appropriately to avoid bias in analysis.

```
=== Missing values per column ===
review_rating    37
dtype: int64
```

Steps Performed

1. The dataset was loaded into a Pandas DataFrame.
2. Column names were standardized for better readability.
3. Missing values in the `review_rating` column were identified and handled.
4. Data types were verified and corrected where required.
5. Descriptive statistics such as mean, minimum, maximum, and quartiles were generated.
6. The cleaned dataset was prepared for database insertion.

Python helped ensure the data was consistent and analysis-ready.

6. Database Integration and SQL Analysis (PostgreSQL)

After cleaning the data, it was uploaded to a PostgreSQL database using Python and SQLAlchemy. The table was created successfully with all 3,900 records.

The following SQL queries were executed in PostgreSQL using pgAdmin to answer key business questions.

- Revenue comparison between male and female customers

	gender text	revenue numeric
1	Female	75191
2	Male	157890

- Customers who used discounts and spent more than the average

	customer_id bigint	purchase_amount bigint
1	2	64
2	3	73
3	4	90
4	7	85
5	9	97
6	12	68
7	13	72
8	16	81
9	20	90
10	22	62
11	24	88
12	29	94
13	32	79
14	33	67
15	35	91
16	37	69
17	40	60
18	41	76
19	43	100
20	44	69
21	55	94

Total rows: 839 Query complete 00:00

- Top products based on average review rating

	item_purchased text	Average Product Rating numeric
1	Gloves	3.86
2	Sandals	3.84
3	Boots	3.82
4	Hat	3.80
5	Skirt	3.78

- Comparison of spending by shipping type

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

- Subscription vs non-subscription customer analysis

	subscription_status text	total_customers bigint	avg_spend numeric	total_revenue numeric
1	Yes	1053	59.49	62645.00
2	No	2847	59.87	170436.00

- Customer segmentation based on previous purchases

	customer_segment text	Number of Customers bigint
1	Loyal	3116
2	New	83
3	Returning	701

- Revenue contribution by different age groups

	age_group text	total_revenue numeric
1	Middle-aged	67711
2	Adult	64927
3	Young Adult	57279
4	Senior	43164

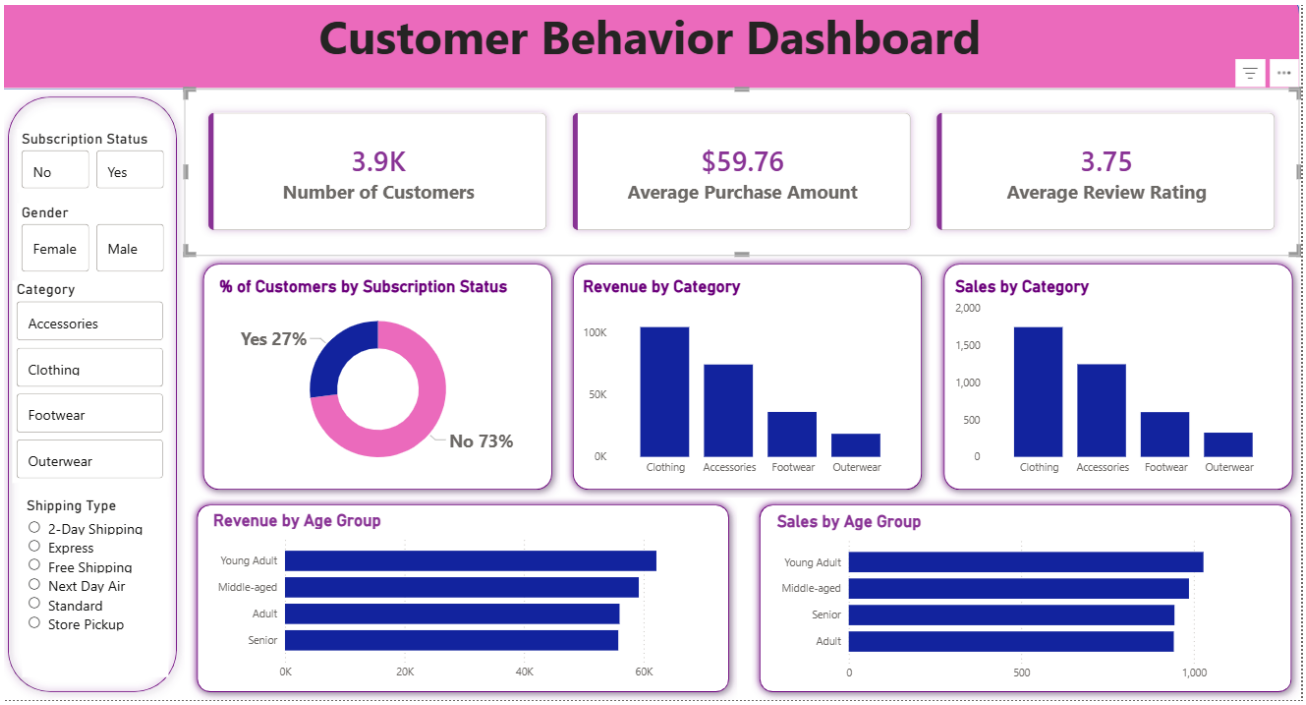
These queries helped extract structured insights directly from the database.

7. Key Insights from SQL Analysis

- Certain product categories generate significantly higher revenue.
- Customers who apply discounts still contribute strongly to overall revenue.
- Subscription customers show consistent engagement.
- Middle-aged and young adult customers contribute the highest revenue.
- Repeat customers form a valuable segment for the business.

8.Data Visualization Using Power BI

An interactive dashboard was created using Power BI to visually present insights derived from SQL analysis. The dashboard was directly connected to the PostgreSQL database, ensuring data consistency and alignment between the analytical queries and the visualizations.



Dashboard Components

- Total number of customers
- Average purchase amount
- Average customer review rating
- Revenue by product category
- Sales by category
- Revenue and sales by age group
- Subscription-based customer distribution
- Interactive filters for gender, category, and shipping type

The dashboard allows users to explore data dynamically and understand trends easily.

9. Business Insights

- Clothing and accessories are the top-performing categories.
- Discount strategies influence purchase decisions significantly.
- Younger and middle-aged customers are key revenue contributors.
- Subscription programs help improve customer retention.
- Customer reviews provide useful feedback for product performance.

10. Recommendations

Based on the analysis, the following recommendations are suggested:

- Strengthen loyalty and subscription programs.
- Target high-value age groups through focused marketing.
- Optimize discount strategies to balance revenue and profitability.
- Promote high-rated products to increase customer trust.
- Use dashboards for continuous performance monitoring.

11. Tools and Technologies Used

- **Python (Pandas, NumPy):** Data cleaning and preprocessing
- **PostgreSQL:** Data storage and SQL analysis
- **pgAdmin:** Database management
- **Power BI:** Data visualization
- **VS Code:** Development environment

12. Conclusion

This project demonstrates how customer shopping data can be effectively analyzed using Python, SQL, and Power BI. By transforming raw data into structured insights and visual dashboards, the project highlights key trends that can support data-driven business decisions and improve overall retail performance.