Customer Shopping Behaviour Analysis

1.Project Overview

This project analyzes customer purchasing patterns using Python, SQL, and Power BI to uncover insights that help businesses improve marketing strategies, customer segmentation, and sales performance. The workflow includes data cleaning, exploratory analysis, SQL-based business queries, and a fully interactive Power BI dashboard for visualization.

2. Dataset Summary

The dataset used in this project contains detailed information about customer shopping behaviour. It includes 3,900 rows and 18 columns, covering demographic details, purchase characteristics, product categories, payment preferences, and customer engagement metrics.

Dataset Size

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

3.Exploratory Data Analysis (EDA)

The Exploratory Data Analysis was performed in Python using **Pandas**, and **SQLAlchemy**. The goal was to understand the structure, quality, and patterns in the customer shopping dataset before performing SQL-based business analysis and dashboard creation.

3.1. Importing Libraries & Loading the Dataset

The analysis begins by loading essential Python libraries:

- **pandas** for data manipulation
- **pymysql & SQLAlchemy** for database upload
- **read_csv()** to load the raw dataset

```
df = pd.read_csv("customer_shopping_behavior.csv")

print(df.head(10))

print(df.columns)
```

Outcome

- First 10 rows were inspected to understand data structure.
- Verified that the dataset contains **3,900 rows × 18 columns**.

3.2. Understanding Dataset Structure

The following commands were used:

```
print(df.info())

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

Key Insights

- **info()** revealed data types: mix of numerical (age, purchase amount, review rating) and categorical (gender, category, payment method).
- **describe()** showed statistical summaries for both numerical and categorical variables.
- Identified missing values in **Review Rating**.

3.3. Missing Value Treatment

A category-wise median imputation was performed for missing review ratings:

```
df['Review Rating'] = df.groupby('Category')['Review Rating'].transform(lambda x: x.fillna(x.median()))
```

Reason

- Different product categories may have different rating behaviours.
- Median is more robust to outliers compared to mean.

Result

- All missing review ratings were successfully filled.

3.4. Column Standardization

For consistency, all column names were converted to lowercase and underscores:

```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
df = df.rename(columns={'purchase_amount_(usd)': 'purchase_amount'})
```

Outcome

- Clean and SQL-friendly column names (e.g., *purchase_amount*, *shipping_type*, *subscription_status*).

3.5. Creating New Features

a) Age Group Segmentation

```
labels = ['Young Adult','Adult','Middle-age','Senior']
df['age_group'] = pd.qcut(df['age'], 4, labels=labels)
```

- Customers were distributed evenly into 4 age segments.
- Useful for demographic analysis and visualization.

b) Purchase Frequency (Numeric Conversion)

Converted frequency labels into numeric days:

```
frequency_mapping = {
                        'Fortnightly': 14,
                        'Weekly': 7,
                       'Monthly': 30,
                       'Quarterly': 90,
                       'Bi-Weekly': 14,
                       'Annually': 365,
                       'Every 3 Months': 90
                                }
df['purchase_frequency_days'] = df['frequency_of_purchases'].map(frequency_mapping)
```

Outcome

- Enables measurable insights into customer shopping habits.
- Helpful for churn prediction and behavioural segmentation.

---

c) Checking Discount vs Promo Code Consistency

```
(df['discount_applied'] == df['promo_code_used']).all()
 df = df.drop('promo_code_used', axis=1)
```

Result

- Both columns were identical, meaning whenever a promo code was used, a discount was applied.
- The redundant column *promo_code_used* was removed.

---

6. Final Dataset Preparation & Upload to MySQL

The cleaned dataset was uploaded to the MySQL database using SQLAlchemy:

```
engine = create_engine("mysql+pymysql://root:root@localhost:3306/customer_behavior")
df.to_sql('customer', con=engine, if_exists='replace', index=False)
```

Outcome

- A clean, structured, analysis-ready version of the dataset was stored in the MySQL database for running SQL queries and business analytics.

4.Data Analysis using SQL

After cleaning and uploading the dataset into MySQL, several business questions were answered using SQL queries. Each query uncovers important customer trends, revenue insights, and behavioural patterns.

1.Total Revenue by Gender

```
SELECT gender, SUM(purchase_amount) AS revenue
FROM customer    GROUP BY gender;
```

Insight

This query calculates total revenue generated from **male vs female** customers.
It helps understand which gender segment contributes more to overall sales.

2. Customers Who Used Discounts & Spent Above Average

```
SELECT customer_id, purchase_amount
 FROM customer
WHERE discount_applied = 'Yes'
AND purchase_amount > (SELECT AVG(purchase_amount) FROM customer);
```

Insight

Identifies **high-value customers** who still used discounts.
These customers:

- Are price-aware but still willing to spend more
- Can be targeted with premium discount campaigns

3.Top 5 Highest Rated Products

```
SELECT item_purchased,
ROUND(AVG(review_rating), 2) AS average_rating
FROM customer
GROUP BY item_purchased
ORDER BY average_rating DESC LIMIT 5;
```

Insight

Reveals items with the highest customer satisfaction.
These items can be:

- Promoted as “Top Rated”
- Featured in recommendation engines

4.Average Purchase Amount by Shipping Type

```
SELECT shipping_type,
ROUND(AVG(purchase_amount), 2) AS average_amount
FROM customer
WHERE shipping_type IN ('Standard','Express')
GROUP BY shipping_type;
```

Insight

Shows whether **Express** shipping customers spend more than **Standard** shipping customers.
Useful for:

- Shipping cost strategy
- Premium delivery offerings

5. Do Subscribed Customers Spend More?

```
SELECT subscription_status,
COUNT(customer_id) AS customer_count,
ROUND(AVG(purchase_amount),2) AS average_spend,
SUM(purchase_amount) AS total_revenue
FROM customer
GROUP BY subscription_status;
```

Insight

Compares:

- **Average spending**
- **Total revenue**
- **Customer count**

Between **subscribed vs non-subscribed** customers.
This analysis helps justify subscription programs.

6.Top 5 Products with Highest Discount Usage

```
SELECT item_purchased,
ROUND(SUM(CASE WHEN discount_applied='Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2)
AS discount_rate FROM customer
GROUP BY item_purchased
ORDER BY discount_rate DESC LIMIT 5;
```

Insight

Identifies items that are most frequently purchased using discounts.
Useful for:

- Pricing strategies
- Identifying discount-dependent products

7. Customer Segmentation: New, Returning, Loyal

```
WITH customer_type AS (  SELECT customer_id, previous_purchases,
        CASE
              WHEN previous_purchases = 1 THEN 'New'
              WHEN previous_purchases BETWEEN 2 AND 10 THEN 'Returning'
              ELSE 'Loyal'
              END AS customer_segment
 FROM customer
     )
SELECT customer_segment,
 COUNT(*) AS number_of_customers
FROM customer_type    GROUP BY customer_segment;
```

Insight

Segments customers into:

- **New**
- **Returning**
- **Loyal**

This is useful for:

- Personalized marketing
- Retention strategies
- Customer lifecycle management

8.Top 3 Most Purchased Products per Category

```
WITH order_count AS ( SELECT item_purchased, category,
  COUNT(customer_id) AS item_count,
  ROW_NUMBER() OVER (PARTITION BY category ORDER BY COUNT(customer_id) DESC) AS row_num
   FROM customer
     GROUP BY category, item_purchased
      )
   SELECT row_num, category, item_purchased
   FROM order_count
  WHERE row_num <= 3;
```

Insight

Shows the **top 3 performing products** in every category.
Helps with:

- Inventory planning
- Cross-selling
- Category-wise marketing campaigns

9.Subscription Likelihood Among Repeat Buyers

```
 SELECT subscription_status,
     COUNT(customer_id) AS repeat_buyers
FROM customer
WHERE previous_purchases >= 5
GROUP BY subscription_status;
```

Insight

Checks if customers with **5+ previous purchases** are more likely to subscribe.
Useful for:

- Predicting subscription conversions
- Targeting repeat buyers

10. Revenue Contribution by Age Group

```
SELECT age_group,
      SUM(purchase_amount) AS revenue
 FROM customer
GROUP BY age_group
ORDER BY revenue DESC;
```

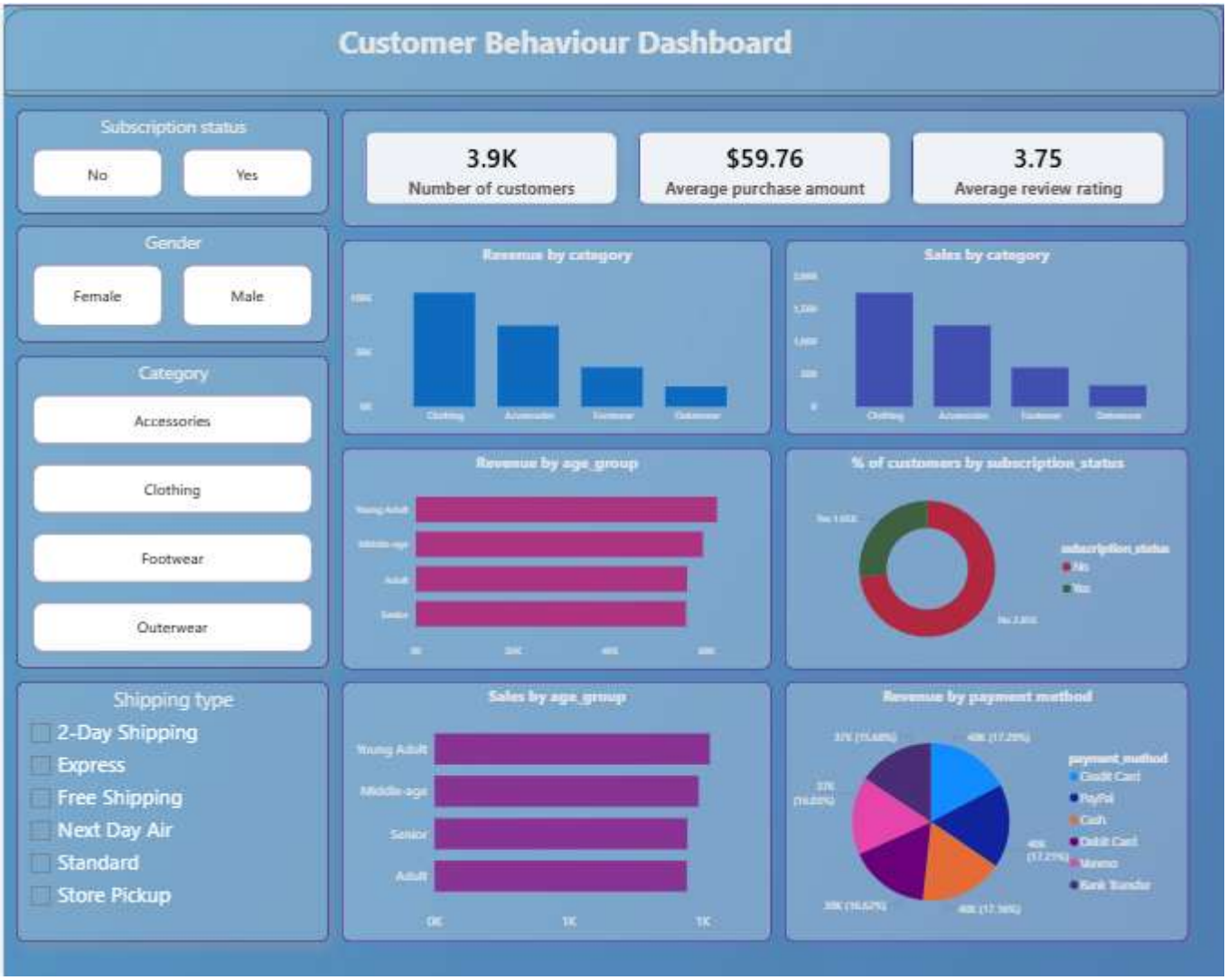
Insight

Shows which age segment contributes the most revenue.
Useful for:

- Age-targeted promotions
- Personalized recommendations

Power BI Dashboard – Customer Shopping Behaviour

The Power BI dashboard provides an interactive and visual summary of customer shopping patterns, revenue drivers, and demographic insights. It integrates the cleaned dataset and SQL outputs to present key business metrics in an easily interpretable format.



◇ Key Highlights of the Dashboard

1. Overall Business KPIs

- 3.9K Customers
- Average Purchase Amount: 59.76 USD
- Average Review Rating: 3.75

These metrics summarize the overall performance and provide a clear snapshot of customer activity.

◇ Revenue Insights

Revenue by Category

A bar chart shows that:

- **Clothing** generates the highest revenue.
- **Accessories** is the second-largest contributor.
- **Footwear and Outerwear** have lower revenue, indicating potential for improvement.

Sales Volume by Category

Confirms that categories with higher sales volume also generate higher revenue.
Useful for understanding product-level performance.

◇ Customer Demographics and Spending Patterns

Revenue by Age Group

- Young Adults and Middle-Age adults **contribute the highest revenue.**
- Seniors **contribute comparatively less.**

Shows which demographic group has the strongest purchasing power.

Sales by Age Group

Indicates purchase frequency trends across age categories.

◇ Subscription Behaviour Analysis

% of Customers by Subscription Status

A donut chart shows:

- Majority are non-subscribers
- Subscribers account for a smaller percentage

This helps evaluate the performance of subscription programs.

◇ Shipping Behaviour

Shipping type filter lets users explore:

- Standard
- Express
- 2-Day
- Free Shipping
- Next Day Air
- Store Pickup

Useful for analyzing fulfillment preferences.

◇ Payment Method Insights

A pie chart displays contribution by payment method:

- Credit Card
- PayPal
- Cash
- Debit Card
- Venmo
- Bank Transfer

This helps in planning payment partnerships and optimizing payment gateway strategies.

◇ Dashboard Filters

The dashboard includes filters for:

- Subscription Status
- Gender
- Category
- Shipping Type

These slicers allow dynamic exploration of the dataset and help stakeholders drill down into specific customer segments.

Purpose of the Dashboard

This dashboard provides a **360-degree view of customer shopping behaviour**, helping businesses identify:

- Best-performing product categories
- High-value customer segments
- Popular payment & shipping methods
- Subscription influence on spending
- Revenue distribution across demographics

It supports **data-driven decision making** across marketing, pricing, inventory, and customer retention strategies.