# Customer Shopping Behavior Analysis

## 1. Project Overview

This project analyzes customer shopping behavior using transactional data from 3,900 purchases across various product categories. The goal is to uncover insights into spending patterns, customer segments, product preferences, and subscription behavior to guide strategic business decisions.

## 2. Dataset Summary

- Rows: 3,900
- Columns: 18
- Key Features:
- Customer demographics (Age, Gender, Location, Subscription Status)
- Purchase details (Item Purchased, Category, Purchase Amount, Season, Size, Color)
- Shopping behavior (Discount Applied, Promo Code Used, Previous Purchases, Frequency of Purchases, Review Rating, Shipping Type)
- Missing Data: 37 values in Review Rating column

## 3. Exploratory Data Analysis using Python

We began with data preparation and cleaning in Python:

- **Data Loading**: Imported the dataset using pandas.
- **Initial Exploration**: Used df.info() to check structure

```
 #   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 (USD)   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
 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)
```

- **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**: Renamed 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.

## 4. Data Analysis using SQL (Business Transactions)

We performed structured analysis in MySQL to answer key business questions:

1. **Revenue by Gender** – Compared total revenue generated by male vs. female customers.

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

**Output:**

| | gender | revenue |
|---|---|---|
| ▶ | Male | 157890 |
| | Female | 75191 |

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

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

**Output:**

| 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 |

3. **Top 5 Products by Rating** – Found products with the highest average review ratings.

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

**Output:**

| item_purchased | Average_Product_Rating |
|---|---|
| Gloves | 3.86 |
| Sandals | 3.84 |
| Boots | 3.82 |
| Hat | 3.8 |
| Skirt | 3.78 |

4. **Shipping Type Comparison –** Compared average purchase amounts between Standard and Express shipping.

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

**Output:**

| | shipping_type | avg_price |
|---|---|---|
| ▶ | Express | 60.48 |
| | Standard | 58.46 |

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

```
SELECT subscription_status, COUNT(customer_id) Total_customers,ROUND(AVG(purchase_amount), 2) Avg_spend,
SUM(purchase_amount) Total_revenue
FROM customer
GROUP BY subscription_status
ORDER BY Total_revenue , Avg_spend DESC;
```

**Output:**

| | subscription_status | Total_customers | Avg_spend | Total_revenue |
|---|---|---|---|---|
| ▶ | Yes | 1053 | 59.49 | 62645 |
| | No | 2847 | 59.87 | 170436 |

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

```
SELECT item_purchased,
ROUND(100 * SUM(CASE WHEN discount_applied = 'Yes' THEN 1 ELSE 0 END) / COUNT(*), 2) AS discount_rate
FROM customer
GROUP BY item_purchased
ORDER BY discount_rate DESC LIMIT 5;
```

**Output:**

| | item_purchased | discount_rate |
|---|---|---|
| | 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
SELECT
  CASE
    WHEN previous_purchases = 1 THEN 'New'
    WHEN previous_purchases BETWEEN 2 AND 10 THEN 'Returning'
    WHEN previous_purchases >= 10 THEN 'Loyal'
  END AS customer_segment,
  COUNT(*) AS segment_count
FROM customer
GROUP BY customer_segment;
```

**Output:**

| customer_segment | segment_count |
|---|---|
| Loyal | 3116 |
| Returning | 701 |
| New | 83 |

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

```sql
with RankedProducts AS (SELECT category, item_purchased, count(customer_id) AS total_orders,
    ROW_NUMBER() OVER (PARTITION BY category ORDER BY count(customer_id) DESC) AS item_rank
    FROM customer
  GROUP BY category, item_purchased
)
SELECT item_rank, category, item_purchased, total_orders FROM RankedProducts
WHERE item_rank <= 3 ORDER BY category, item_rank;
```

**Output:**

| item_rank | category | item_purchased | total_orders |
|---|---|---|---|
| 1 | Accessories | Jewelry | 171 |
| 2 | Accessories | Sunglasses | 161 |
| 3 | Accessories | Belt | 161 |
| 1 | Clothing | Blouse | 171 |
| 2 | Clothing | Pants | 171 |
| 3 | Clothing | Shirt | 169 |
| 1 | Footwear | Sandals | 160 |
| 2 | Footwear | Shoes | 150 |
| 3 | Footwear | Sneakers | 145 |
| 1 | Outerwear | Jacket | 163 |
| 2 | Outerwear | Coat | 161 |

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

```sql
SELECT COUNT(customer_id) repeat_buyer, subscription_status
FROM customer
WHERE previous_purchases > 5
GROUP BY subscription_status;
```

**Output:**

| repeat_buyer | subscription_status |
|---|---|
| 958 | Yes |
| 2518 | No |

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

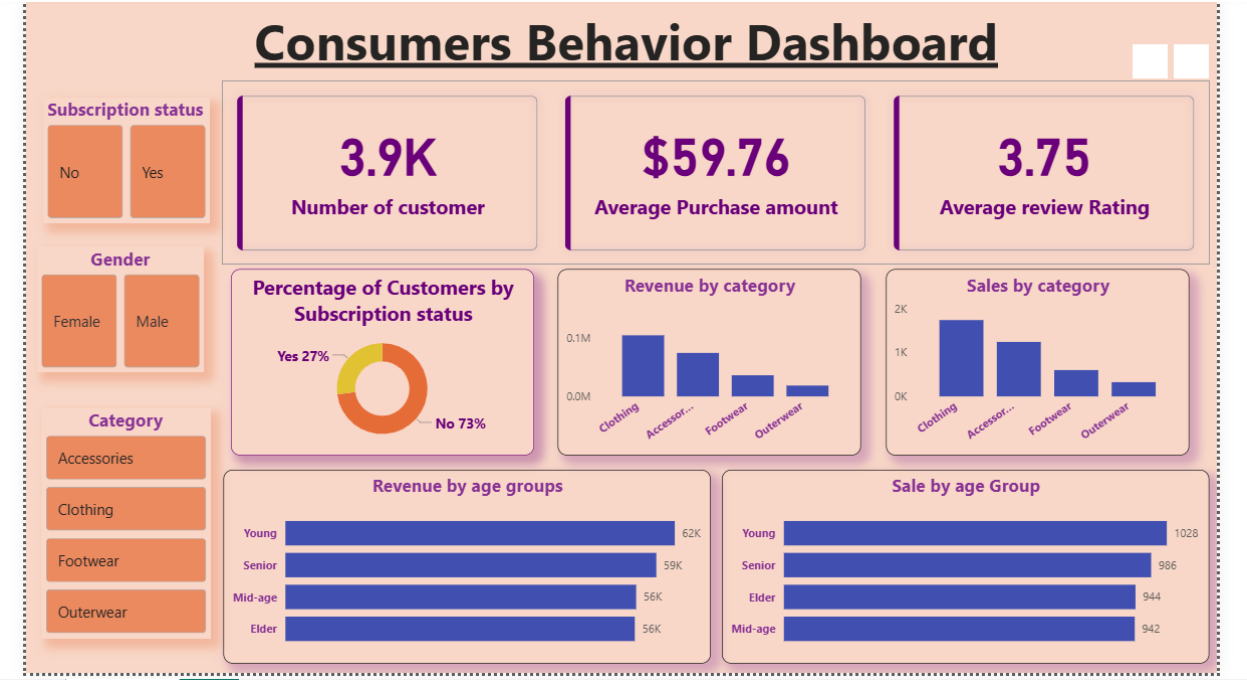
```sql
SELECT Age_group, SUM(purchase_amount) total_revenue
FROM customer
GROUP BY Age_group
ORDER BY total_revenue DESC;
```

**Output:**

| Age_group | total_revenue |
|---|---|
| Young | 62143 |
| Senior | 59197 |
| Mid-age | 55978 |
| Elder | 55763 |

## 5. Dashboard in Power BI

Finally, we built an interactive dashboard in **Power BI** to present 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.