

# Customer Shopping Behavior Analysis

## Overview

<i>Project Name</i>	Customer Shopping Behavior Analysis.
<i>Created by</i>	Devvrat Tiwari
<i>Tools Used:</i>	<ul style="list-style-type: none"><li>• Python (pandas)</li><li>• PostgreSQL</li><li>• Power BI</li><li>• Jupyter Notebook</li></ul>
<i>Background</i>	Write a brief history of the project, starting with the rationale or problem it set out to solve.
<i>Objectives</i>	<ul style="list-style-type: none"><li>• Understand Customer Behavior.</li><li>• Segment Customers.</li><li>• Analyze Subscription Impact.</li><li>• Evaluate Product Performance.</li><li>• Build a BI Dashboard.</li></ul>

## 1. Project Overview

This project focuses on analyzing customer shopping behavior using transactional data from 3,900 purchases across multiple product categories. The objective is to uncover meaningful insights into customer spending patterns, segmentation, product preferences, and subscription behavior. These insights help guide strategic decisions in marketing, sales, and customer retention.

## 2. Dataset Summary

- Rows: 3,900
- Columns: 18

### Key Feature Categories

- **Customer Demographics:**  
Age, Gender, Location, Subscription Status
- **Purchase Information:**  
Item Purchased, Category, Purchase Amount, Season, Size, Color
- **Shopping Behavior:**  
Discount Applied, Promo Code Used, Previous Purchases, Purchase Frequency,  
Review Rating, Shipping Type
- **Missing Data:**  
37 missing values in the *Review Rating* column

## 3. Exploratory Data Analysis (Python - Jupyter Notebook)

### Data Preparation & Cleaning

- Loaded the dataset using **pandas**
- Used `df.info()` and `df.describe()` for initial data understanding

- Identified missing values and **imputed** Review Rating using the **median rating per product category**
- Standardized column names into **snake\_case** for readability

```
[10]: #creating a column age_group
labels = ['Young Adult', 'Adult', 'Middle-aged', 'Senior']
df['age_group'] = pd.qcut(df['age'], q=4, labels = labels)
```

```
[11]: df[['age', 'age_group']].head(10)
```

## Feature Engineering

- Created **age\_group** by binning customer ages
- Generated **purchase\_frequency\_days** from purchase timestamps
- Checked redundancy between **discount\_applied** and **promo\_code\_used**  
→ Dropped *promo\_code\_used* after confirming overlap

```
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)
```

```
13]: df[['purchase_frequency_days', 'frequency_of_purchases']].head(10)
```

## Database Integration

- Connected Python with **PostgreSQL**
- Uploaded cleaned DataFrame into PostgreSQL for deeper SQL-based business analysis

# 4. SQL-Based Business Analysis (PostgreSQL)

Key analytical queries included:

- **Revenue by Gender** - Compared total spending between male and female customers

	gender text	revenue numeric
1	Female	75191
2	Male	157890

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

- **High-Spending Discount Users** - Identified customers who used discounts yet exceeded average spend
- **Top 5 Products by Rating** - Ranked items with the highest average review scores

	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

- **Shipping Type Comparison** - Compared average spend across Standard vs. Express shipping

- **Subscribers vs. Non-Subscribers** – Analyzed revenue and average spend differences

	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

- **Discount-Dependent Products** – Listed top 5 products heavily dependent on discounts

	item_purchased text	discount_rate numeric
1	Hat	50.00
2	Sneakers	49.00
3	Coat	49.00
4	Sweater	48.00
5	Pants	47.00

- **Customer Segmentation** – Classified customers into New, Returning, and Loyal groups

	customer_segmentation text	number of customer bigint
1	Loyal	3116
2	New	83
3	Returning	701

	age_group text	total_revenue numeric
1	Young Adult	62143
2	Middle-aged	59197
3	Adult	55978
4	Senior	55763

- **Top Products per Category** – Identified top 3 most purchased products within each category
- **Repeat Buyers & Subscription Correlation** – Checked if customers with >5 purchases tend to subscribe
- **Revenue by Age Group** – Calculated each age group's contribution to total revenue

	item_rank bigint	category text	item_purchased text	total_orders bigint
1	1	Accessori...	Jewelry	171
2	2	Accessori...	Sunglasses	161
3	3	Accessori...	Belt	161
4	1	Clothing	Blouse	171
5	2	Clothing	Pants	171
6	3	Clothing	Shirt	169
7	1	Footwear	Sandals	160
8	2	Footwear	Shoes	150
9	3	Footwear	Sneakers	145
10	1	Outerwear	Jacket	163
11	2	Outerwear	Coat	161

## 5. Power BI Dashboard

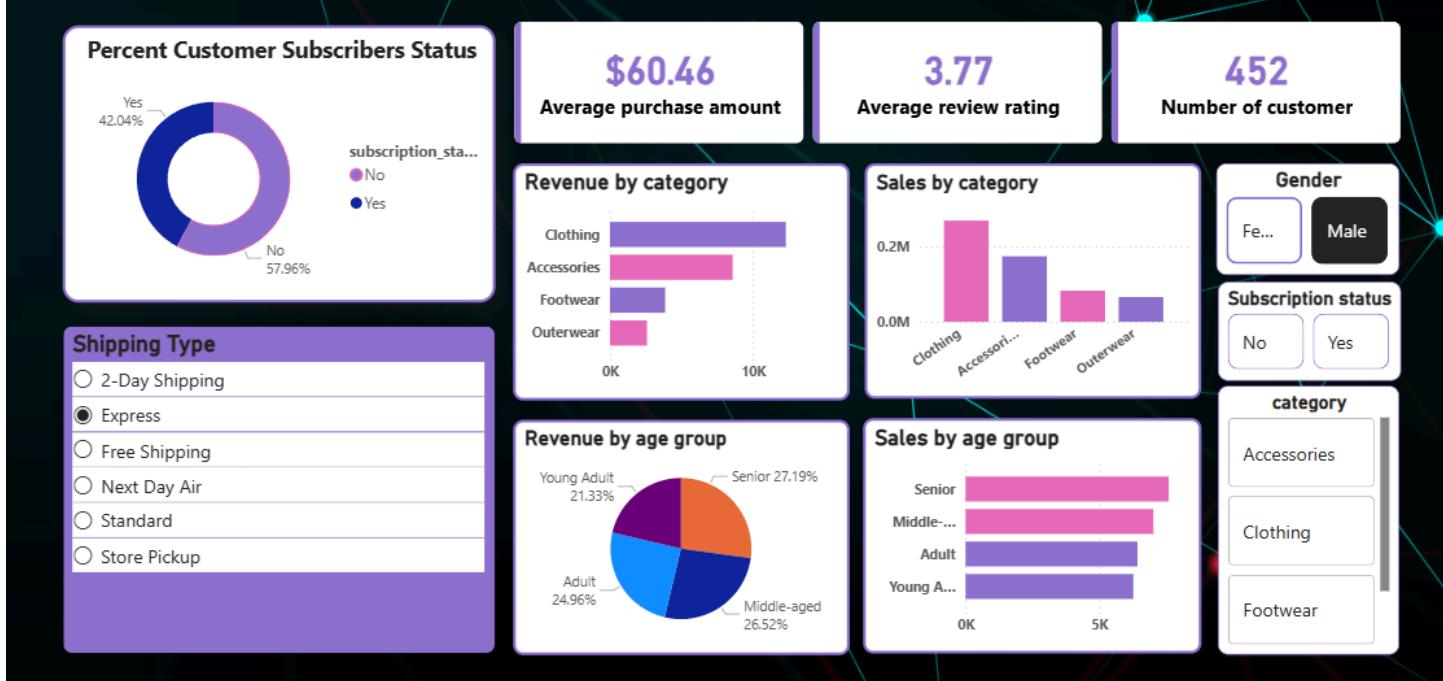
Created an interactive dashboard covering:

- Revenue and sales trends
- Customer segmentation distribution
- Subscription-based performance
- Product category insights
- Average spend by shipping type, age group, and gender
- Discount patterns and rating analysis

This dashboard allows stakeholders to visually explore business trends and take data-driven actions.



# Customer Behaviour Analysis



## 6. Business Recommendations

### 1. Boost Subscription Adoption

Introduce exclusive benefits (priority shipping, offer previews, loyalty points) to increase subscription conversions.

### 2. Strengthen Loyalty Programs

Reward customers with high purchase frequency to convert them into long-term “Loyal” customers.

### 3. Optimize Discount Strategy

Analyze discount-heavy products and optimize pricing to protect profit margins while maintaining sales volume.

### 4. Highlight High-Performing Products

Promote top-rated and best-selling products in digital campaigns to maximize ROI.

### 5. Enable Targeted Marketing

Focus marketing strategies toward:

- High-revenue age groups
- Frequent express-shipping users
- Customer segments with high purchase potential