Customer Shopping Behavior Analysis

1. Project Overview

This project analyzes 3,900 customer transactions to understand shopping behavior. It covers spending patterns, customer types, popular products, and subscription habits to help make better business decisions.

2. Dataset Summary

- Rows: 3,900
- Columns: 18
- Key Features:
  - Customer details: 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 and `.describe()` for summary statistics.

```
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 (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
```

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

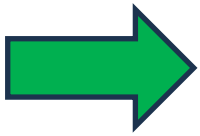
	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Pro Co Us
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	3900	39
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2	6	2	
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No	Free Shipping	No	
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847	675	2223	22
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN	NaN	NaN	N
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN	NaN	NaN	N
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN	NaN	NaN	N
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN	NaN	NaN	N
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN	NaN	NaN	N
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN	NaN	NaN	N
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN	NaN	NaN	N

- **Missing Data Handling:** Checked for null values and imputed missing values in the `Review Rating` column using the median rating of each product category.

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

```
df.isnull().sum()

Customer ID                0
Age                        0
Gender                     0
Item Purchased             0
Category                   0
Purchase Amount (USD)      0
Location                   0
Size                       0
Color                      0
Season                     0
Review Rating             37
Subscription Status        0
Shipping Type              0
Discount Applied           0
Promo Code Used            0
Previous Purchases         0
Payment Method             0
Frequency of Purchases     0
dtype: int64
```



```
df.isnull().sum()

Customer ID                0
Age                        0
Gender                     0
Item Purchased             0
Category                   0
Purchase Amount (USD)      0
Location                   0
Size                       0
Color                      0
Season                     0
Review Rating              0
Subscription Status        0
Shipping Type              0
Discount Applied           0
Promo Code Used            0
Previous Purchases         0
Payment Method             0
Frequency of Purchases     0
dtype: int64
```

- **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.

```
# Create a new Column age_group

bins = [0,25,40,60,100]
labels = ['Young Adult', 'Adult', 'Middle-Aged', 'Senior']
df["age_group"] = pd.cut(df["age"], bins= bins, labels= labels, include_lowest= True)
```

```
df[["age", "age_group"]].head(10)
```

	age	age_group
0	55	Middle-Aged
1	19	Young Adult
2	50	Middle-Aged
3	21	Young Adult
4	45	Middle-Aged
5	46	Middle-Aged
6	63	Senior
7	27	Adult
8	26	Adult
9	57	Middle-Aged

- **Data Consistency Check:** Verified if `discount_applied` and `promo_code_used` were redundant; dropped `promo_code_used`.
- **Database Integration:** Connected Python script to MySQL and loaded the cleaned DataFrame into the database for SQL analysis.

```
# Create Column purchase_frequency_days

frequency_mapping = {
    "Weekly" :7,
    "Bi-Weekly" : 14,
    "Fortnightly" : 14,
    "Monthly" : 30,
    "Every 3 Months" : 90,
    "Quarterly" : 90,
    "Annually" : 365
}

df["purchase_frequency_days"] = df["frequency_of_purchases"].map(frequency_mapping)
```

```
df[["purchase_frequency_days", "frequency_of_purchases"]].head(10)
```

	purchase_frequency_days	frequency_of_purchases
0	14	Fortnightly
1	14	Fortnightly
2	7	Weekly
3	7	Weekly
4	365	Annually
5	7	Weekly
6	90	Quarterly
7	7	Weekly
8	365	Annually
9	90	Quarterly

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.

gender	revenue
Male	157890
Female	75191

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

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

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

item_purchased	avg_review_rating
Gloves	3.861
Sandals	3.844
Boots	3.819
Hat	3.801
Skirt	3.785

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

shipping_type	avg_purchase_amount
Express	60.48
Standard	58.46

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

subscription_status	total_customers	average_spend	total_revenue
No	2847	59.87	170436
Yes	1053	59.49	62645

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

item_purchased	percentage_purchase
Hat	50.00
Sneakers	49.66
Coat	49.07
Sweater	48.17
Pants	47.37

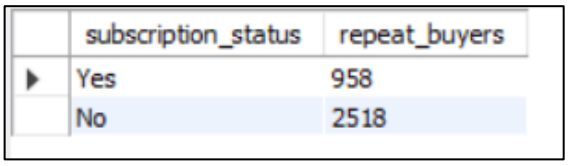
7. **Customer Segmentation** – Classified customers into New, Returning, and Loyal segments based on purchase history.

customer_segment	no_of_customers
Loyal	3116
Returning	701
New	83

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

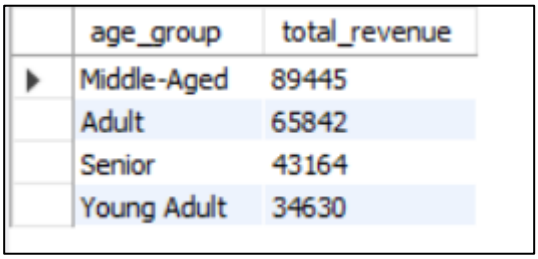
item_rank	category	item_purchased	total_orders
1	Accessories	Jewelry	171
2	Accessories	Sunglasses	161
3	Accessories	Belt	161
1	Clothing	Blouse	171
2	Clothing	Pants	171
3	Clothing	Shirt	169
1	Footwear	Sandals	160
2	Footwear	Shoes	150
3	Footwear	Sneakers	145
1	Outerwear	Jacket	163
2	Outerwear	Coat	161

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



subscription_status	repeat_buyers
Yes	958
No	2518

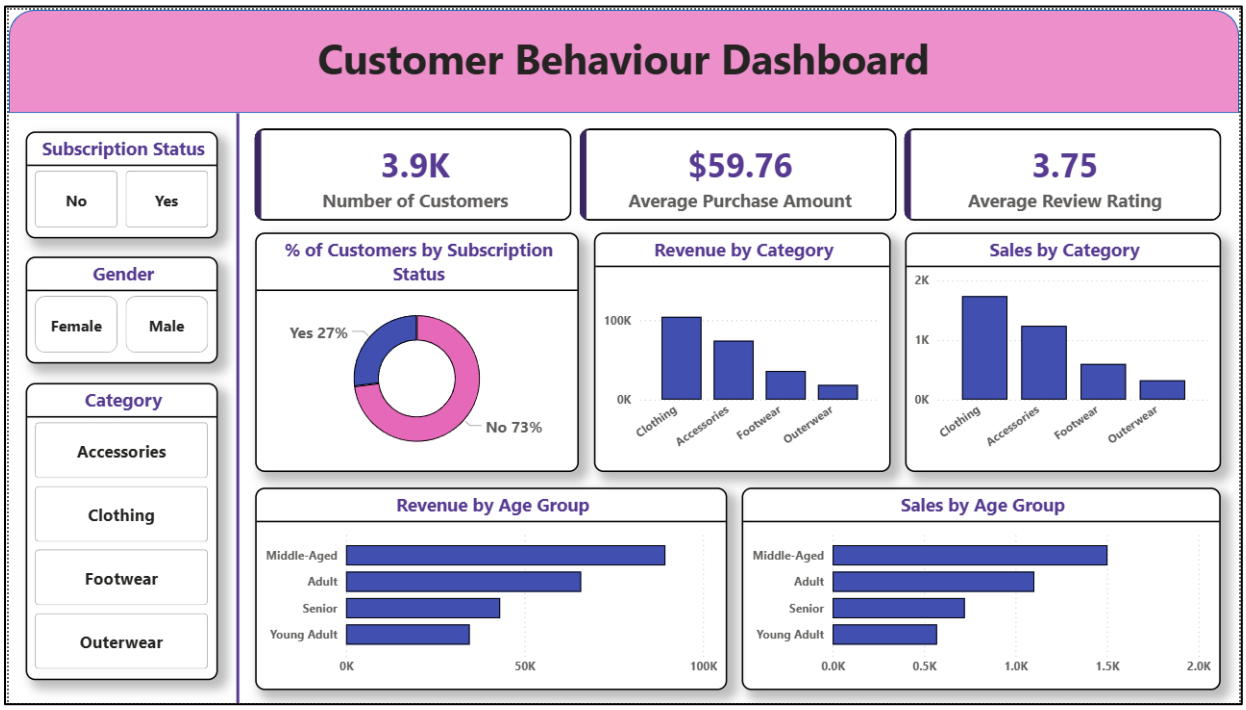
10. **Revenue by Age Group** – Calculated total revenue contribution of each age group.



age_group	total_revenue
Middle-Aged	89445
Adult	65842
Senior	43164
Young Adult	34630

5. Dashboard in Power BI

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



6. Business Recommendations

- **Boost Subscriptions** – Highlight exclusive benefits to encourage more users to subscribe.
- **Customer Loyalty Programs** – Reward repeat buyers to move them into the “Loyal” segment.
- **Review Discount Policy** – Use discounts wisely to improve sales without reducing profit margins.
- **Product Positioning** – Highlight top-rated and best-selling products in campaigns.
- **Targeted Marketing** – Focus efforts on high-revenue age groups and express-shipping users.