Customer Shopping Behavior Analysis Project

1. 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 and .describe() for summary statistics.

```
# df.describe() .....only gives summary statistics of numerical columns
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	Promo Code Used	3
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	3900	3900	
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2	6	2	2	
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No	Free Shipping	No	No	
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847	675	2223	2223	
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN	NaN	NaN	NaN	
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN	NaN	NaN	NaN	
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN	NaN	NaN	NaN	
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN	NaN	NaN	NaN	
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN	NaN	NaN	NaN	
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN	NaN	NaN	NaN	
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN	NaN	NaN	NaN	

Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
3900	3900	3900.000000	3900	3900
2	2	NaN	6	7
No	No	NaN	PayPal	Every 3 Months
2223	2223	NaN	677	584
NaN	NaN	25.351538	NaN	NaN
NaN	NaN	14.447125	NaN	NaN
NaN	NaN	1.000000	NaN	NaN
NaN	NaN	13.000000	NaN	NaN
NaN	NaN	25.000000	NaN	NaN
NaN	NaN	38.000000	NaN	NaN
NaN	NaN	50.000000	NaN	NaN

```
[6]: 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
```

- Missing Data Handling: Checked for null values and imputed missing values in the Review Ratingcolumn using the median rating of each product category. (I used groupby('category') to split the category into diff buckets/categories of 'Category' column and then apply the median of respective buckets/categories to the missing values 'Review Rating' column).
- Column Standardization: Renamed columns to snake case for better readability and documentation.
- Feature Engineering:
    - Created age_group column by binning customer ages. (This division/feature engineering is useful for marketing or understanding customer behavior).
    - Created purchase_frequency_days column from purchase data. (The current dataset has frequency of purchases (weekly, annually, etc...) col that is in text form. For analysis text form is harder than numerical form. I have mapped the 'freq_of_purchases' col with the freq_mapping dict to create a new col which represents the fre_of_purchases col in numbers rather than text).
- Data Consistency Check: Verified if discount_applied and promo_code_used were redundant; dropped promo_code_used.

```
# 3) Check if discount_applied and promo_code_used are redundant. Evidently when a promo code is used, a discount is applied hence both indicate 'yes'.
#    But a company can still give discounts without the use of any promo code such as seasonal discounts, clearance discounts, etc. Hence, I'm trying to
#    see if these 2 columns differ in any instance. Basically trying to find if promo_code=No and discount_applied=Yes, then both columns are needed. If
#    if both the columns are same throughout, then they are reducndant and I will then eliminate a column.

df[['discount_applied', 'promo_code_used']].head(10)
(df['discount_applied'] == df['promo_code_used']).all()
```

```
True
```

- Database Integration: Connected Python script to PostgreSQL using sqlalchemy and loaded the cleaned DataFrame into the database for SQL analysis.

4. Data Analysis using SQL (Business Transactions)

Performed structured analysis in PostgreSQL to answer key business questions:

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

	gender text	revenue numeric
1	Female	75191
2	Male	157890

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

	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	99

Total rows: 839 Query complete 00:00:

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

	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

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

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

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

	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

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

	item_purchased text	discount_rate numeric
1	Hat	50.00
2	Sneakers	49.66
3	Coat	49.07
4	Sweater	48.17
5	Pants	47.37

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

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

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

	subscription_status text	repeat_buyers bigint
1	No	2518
2	Yes	958

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

	item_rank bigint	category text	item_purchased text	total_orders bigint
1	1	Accessories	Jewelry	171
2	2	Accessories	Sunglasses	161
3	3	Accessories	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

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

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

5. Power BI Dashboard

Built an interactive dashboard in Power BI to present the Business insights visually.