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 and .describe() for summary statistics.

	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	
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	3900	3900	39
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	

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

- **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)

```
CREATE DATABASE customer_behavior;

USE customer_behavior;

SELECT *
FROM INFORMATION_SCHEMA.TABLES;
```

We performed structured analysis in SQL Server to answer key business questions:

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

```sql
-- Total revenue generated by male vs. female customers?

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

	gender	revenue
1	Male	157890
2	Female	75191

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

```sql
-- Discount Users Spending Above Average

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

	customer_id	purchase_amount
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	88
12	29	94
13	32	79
14	33	67

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

```
-- Top 5 Products by Average Review Rating


SELECT TOP 5
       item_purchased,
       ROUND(AVG(CAST(review_rating AS DECIMAL(5,2))), 2)
       AS [Average Product Rating]
FROM customer
GROUP BY item_purchased
ORDER BY AVG(review_rating) DESC;
```

	item_purchased	Average Product Rating
1	Gloves	3.860000
2	Sandals	3.840000
3	Boots	3.820000
4	Hat	3.800000
5	Skirt	3.780000

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

```
-- Compare Shipping Types
SELECT shipping_type,
       ROUND(AVG(purchase_amount), 2) AS avg_purchase
FROM customer
WHERE shipping_type IN ('Standard','Express')
GROUP BY shipping_type;
```

	shipping_type	avg_purchase
1	Standard	58
2	Express	60

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

```sql
-- Subscribers vs Non-Subscribers

SELECT subscription_status,
       COUNT(customer_id) AS total_customers,
       ROUND(AVG(purchase_amount),2) AS avg_spend,
       ROUND(SUM(purchase_amount),2) AS total_revenue
FROM customer
GROUP BY subscription_status
ORDER BY total_revenue DESC, avg_spend DESC;
```

	subscription_status	total_customers	avg_spend	total_revenue
1	No	2847	59	170436
2	Yes	1053	59	62645

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

```sql
-- Top 5 Products by Discount Percentage

SELECT TOP 5
       item_purchased,
       ROUND(
           100.0 * 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;
```

	item_purchased	discount_rate
1	Hat	50.000000000000
2	Sneakers	49.660000000000
3	Coat	49.070000000000
4	Sweater	48.170000000000
5	Pants	47.370000000000

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

```sql
-- Customer Segmentation

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;
```

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

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

```sql
-- Top 3 Products Within Each Category

WITH item_counts 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 item_counts
WHERE item_rank <= 3;
```

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

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

```sql
-- Repeat Buyers Subscription Behavior

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

	subscription_status	repeat_buyers
1	Yes	958
2	No	2518

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

```sql
-- Revenue by Age Group

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

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

5. Dashboard in Power BI

Finally, I 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.

Document By

Musharaf Shaik