Customer Shopping Behavior Analysis

Case Study:

- Tools: Jupyter Notebook, MySQL, PowerBI
- GitHub Repository: Customer-Behavior-Analysis

Project Overview

This project analyzes customer shopping behavior using transactional data from 3900 purchases across various product categories. The goal is to uncover insights into spending patterns, customer segments, product references, and subscription behavior to guide strategic business decisions.

Data Summary

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

Exploratory Data Analysis using Python

- Data Loading: Imported the dataset using Pandas.
- Initial Exploration: Used .info( ) to check structure and .describe( ) for summary statistics.
- 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.
- Database Integration: Connected Python script to MySQL and loaded the cleaned DataFrame into the database for SQL analysis.

```
df.describe()
```

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3863.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.750065	25.351538
std	1125.977353	15.207589	23.685392	0.716983	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.800000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

Data Analysis using SQL

Performed structured analysis in MySQL to answer key business questions:

1. **Revenue by Gender** – Compared total revenue generated by Male vs. Female customers.

   SQL Query:

```
SELECT
gender,
SUM(purchase_amount) AS revenue
FROM customer
GROUP BY 1;
```

   Result Snapshot:

gender	revenue
Male	157890
Female	75191

   Insights:
   Male customers generate ~68% of total revenue, more than 2x the revenue of female customers.

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

   SQL Query:

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

   Result Snapshot:

customer_id	purchase_amount	avg_purchase_amount
2	64	59.7644
3	73	59.7644
4	90	59.7644
7	85	59.7644
9	97	59.7644
12	68	59.7644
13	72	59.7644
16	81	59.7644

Result 2

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

   SQL Query:

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

   Result Snapshot:

	product	avg_product_rating
▶	Gloves	3.86
	Sandals	3.84
	Boots	3.82
	Hat	3.8
	Handbag	3.78

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

   SQL Query:

```
SELECT
shipping_type,
ROUND(AVG(purchase_amount),2) AS avg_purchase_amount
FROM customer
WHERE shipping_type IN ('Standard', 'Express')
GROUP BY 1
ORDER BY 2 DESC
;
```

   Result Snapshot:

	shipping_type	avg_purchase_amount
▶	Express	60.48
	Standard	58.46

   Insights:
   Customers who choose Express shipping spend ~3.5% more per order than standard shipping customers.

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

SQL Query:

```
SELECT
CASE WHEN subscription_status = 'Yes' THEN 'Subcribers'
     WHEN subscription_status = 'NO' THEN 'Non-Subscribers'
     END AS subscription_status,
SUM(purchase_amount) AS total_purchase_amount,
ROUND(AVG(purchase_amount),2) AS avg_purchase_amount
FROM customer
GROUP BY 1
ORDER BY 3 DESC
;
```

Result Snapshot:

	subscription_status	total_purchase_amount	avg_purchase_amount
▶	Non-Subscribers	170436	59.87
	Subcribers	62645	59.49

6. **Discount Dependent Products** – Quantified customer price sensitivity by measuring what percentage of purchases relied on discounts, helping determine how dependent revenue is on promotions.

SQL Query:

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

Result Snapshot:

	product	discount_percentage
▶	Hat	50.00
	Sneakers	49.66
	Coat	49.07
	Sweater	48.17
	Pants	47.37

Insights:
If Discount % is high -> pricing power is weak.
If Discount % is low -> strong brand & margins.

7. **Customer Segmentation** – Segmented customers based on frequency to distinguish loyal from new and returning customers.

SQL Query:

```
SELECT
CASE WHEN previous_purchases < 5 THEN 'New'
     WHEN previous_purchases BETWEEN 5 AND 10 THEN 'Returning'
     WHEN previous_purchases > 10 THEN 'Loyal'
     ELSE 'Unknown' END AS customer_segment,
COUNT(DISTINCT customer_id) AS num_customers
FROM customer
GROUP BY customer_segment
ORDER BY 1
;
```

Result Snapshot:

	customer_segment	num_customers
▶	Loyal	3116
	New	337
	Returning	447

8. **Top 3 Products per Category** – Identified the top-selling SKUs within each category to enable focused promotion, optimized shelf space, and reduced inventory risk by emphasizing high-velocity products.

SQL Query:

```
WITH CTE AS(
    SELECT
    item_purchased AS product,
    category,
    COUNT(*) AS total_orders,
    DENSE_RANK() OVER(PARTITION BY category ORDER BY COUNT(item_purchased) DESC) AS rnk
    FROM customer
    GROUP BY 1, 2
)
SELECT
product,
category,
total_orders,
rnk
FROM CTE
WHERE rnk <= 3
ORDER BY category, rnk
;
```

Result Snapshot:

Result Grid | Filter Rows: | Export

	product	category	total_orders	rnk
▶	Jewelry	Accessories	171	1
	Sunglasses	Accessories	161	2
	Belt	Accessories	161	2
	Scarf	Accessories	157	3
	Blouse	Clothing	171	1
	Pants	Clothing	171	1

Result 9 ×

Output

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

   SQL Query:

```
SELECT
CASE WHEN subscription_status = 'Yes' THEN 'subscribers'
          WHEN subscription_status = 'No' THEN 'non_subscribers'
          END AS customer_type,
COUNT(DISTINCT customer_id) AS customer_count
FROM customer
WHERE previous_purchases > 5
GROUP BY 1;
```

   Result Snapshot:

	customer_type	customer_count
▶	non_subscribers	2518
	subscribers	958

10. **Revenue by Age Group** – Segmented revenue by customer age group to identify and guide targeted marketing, personalization and product positioning.

    SQL Query:

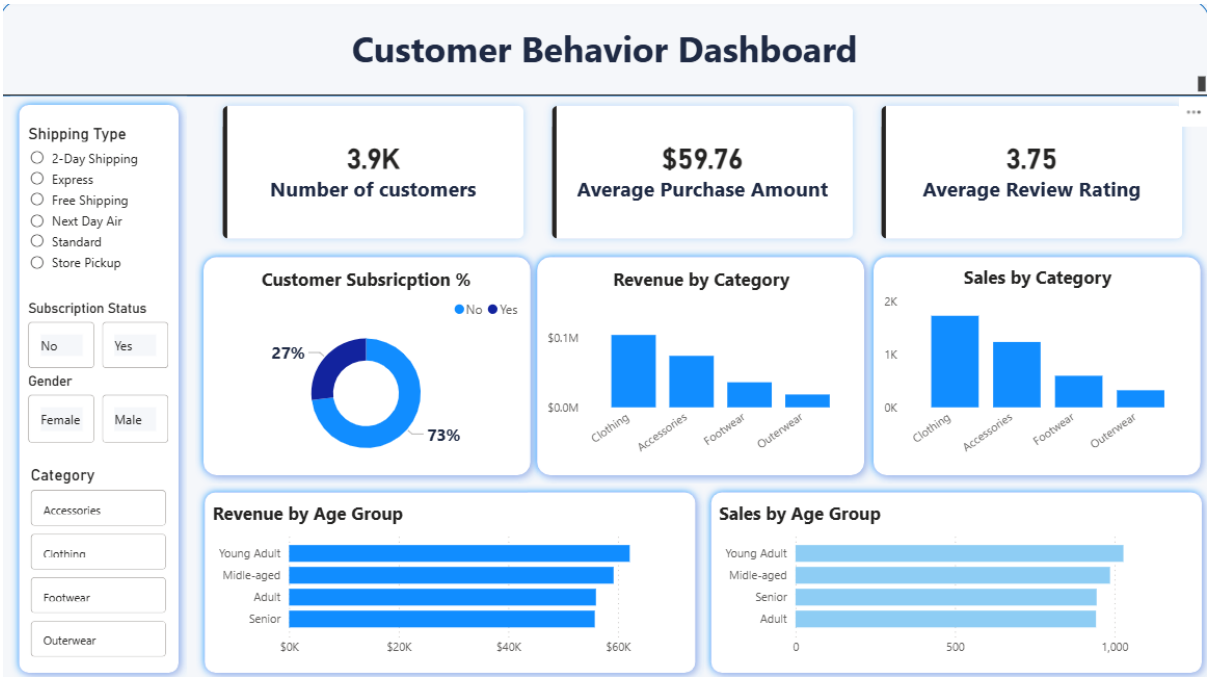
```
SELECT
age_group,
SUM(purchase_amount) AS revnue,
ROUND(SUM(purchase_amount) * 100/ (SELECT SUM(purchase_amount) FROM customer), 2) AS perc_of_contribution
FROM customer
GROUP BY age_group
ORDER BY perc_of_contribution DESC
;
```

    Result Snapshot:

	age_group	revnue	perc_of_contribution
▶	Young Adult	62143	26.66
	Midle-aged	59197	25.40
	Adult	55978	24.02
	Senior	55763	23.92

    Insights:
    Young Adults drive the highest revenue -> marketing dollars should go there.

Dashboard in Power BI



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.