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: 3900
-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, Snipping Type)
- Missing Data: 37 values in the Review Rating column

3. Exploratory Data Analysis using Python

We began with data preparation and cleaning in Python

- **Data Loading**: Import the dataset using pandas
- **Initial exploration**: Used df.head(), df.info() to check structure and df.describe() for summary statistics

```
[2]  df.info()
     ✓ 0.1s
...  <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.head(10)
```

0.1s Open 'df' in Data Wrangler — Python

	# Customer ID	# Age	Gender	Item Purchased	Category
0	1	55	Male	Blouse	Clothing
1	2	19	Male	Sweater	Clothing
2	3	50	Male	Jeans	Clothing
3	4	21	Male	Sandals	Footwear
4	5	45	Male	Blouse	Clothing
5	6	46	Male	Sneakers	Footwear
6	7	63	Male	Shirt	Clothing
7	8	27	Male	Shorts	Clothing
8	9	26	Male	Coat	Outerwear
9	10	57	Male	Handbag	Accessories

10 rows x 18 cols 10 per page Page 1 of 1

- **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 the age_group column by binning customer ages into four groups (0-24 as young adult, 25-34 as adult, 35-54 as middle age, 55-100 as senior).
    - Created the purchase_frequency_days column from the purchase data
- Data consistency Check: Verified if discount_applied and promo_code_used were redundant; dropped Promo_code_used as all users that applied discounts used promo codes.
- **Database Integration:** Connected Python to PostgreSQL and loaded the cleaned DataFrame into the database for SQL Analysis.

4. Data Analysis using SQL

We performed structured analysis in PostgresSQL to answer jey business questions:

1. **Revenue by Gender:** Compared total revenue generated by male vs female customers using these SQL codes

```
select gender, sum(purchase_amount) as total_purchase
from customer
group by gender
order by sum(purchase_amount) desc;
```

	gender text	total_purchase numeric
1	Male	157890
2	Female	75191

2. **High-spending Discount users-** identified customers who used discounts but still spent above the average purchase amount.

```
select customer_id, purchase_amount
from customer
where discount_applied = 'Yes'
and purchase_amount > (select avg(purchase_amount) from customer);
```

	customer_id bigint	purchase_amount bigint
1	2	64
2	3	73
3	4	90
4	7	85
5	9	97
6	12	68

Total rows: 839 Query complete 00:00:00.352

3. **Top 5 Products by Rating:** identified products with the highest average review ratings.

```
select item_purchased, round(avg(review_rating)::numeric,2) as "avg_review_rating"
from customer
group by item_purchased
order by avg(review_rating) desc
limit 5;
```

	item_purchased text	avg_review_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**: analyzed and compared the average purchase amounts between standard and express shipping options.

```
select shipping_type, round(avg(purchase_amount)::numeric, 2) as "avg_purchase"
from customer
where shipping_type in ('Standard', 'Express')
group by shipping_type
order by avg_purchase desc;
```

	shipping_type text	avg_purchase numeric
1	Express	60.48
2	Standard	58.46

5. **Subscription Impact Analysis**: evaluated whether subscribed customers spend more by comparing the average purchase amounts and total revenue between subscribers and non-subscribers.

```
select subscription_status,
count(customer_id) as total_customers,
round(avg(purchase_amount)::numeric, 2) as  avg_spend,
round(sum(purchase_amount)::numeric, 2) as  total_revenue
from customer
where subscription_status in ('Yes', 'No')
group by subscription_status
order by total_customers desc;
```

	subscription_status text	total_customers bigint	avg_spend numeric	total_revenue numeric
1	No	2847	59.87	170436.00
2	Yes	1053	59.49	62645.00

6. **Discount Influence Analysis**: identified the top 5 products with the highest percentage of purchases made using discounts.

```
select item_purchased,
100* sum(case when discount_applied = 'Yes' then 1 else 0 end)/count(*) as
discount_rate
from customer
group by item_purchased
order by discount_rate desc
limit 5;
```

	item_purchased text	discount_rate bigint
1	Hat	50
2	Sneakers	49
3	Coat	49
4	Sweater	48
5	Pants	47

7. **Customer Segmentation**: classified customers as New, Returning, or Loyal based on total previous purchases and counted each segment.

```
with segment as (
select case when previous_purchases = 1 then 'New'
                    when previous_purchases between 2 and 10 then 'Returning'
                    else 'Loyal'
               end as customer_segment,
customer_id
from customer)
select customer_segment, count(customer_id) as no_of_customers
from segment
group by customer_segment
order by no_of_customers desc;
```

	customer_segment text	no_of_customers bigint
1	Loyal	3116
2	Returning	701
3	New	83

8. **Top Products per Category**: determined the top 3 most purchased products within each category.

```
with cte1 as (
select item_purchased, sum(purchase_amount) as total_amount, category
from customer
group by item_purchased, category),
cte2 as(
select category, item_purchased, total_amount,
row_number() over(partition by category order by total_amount desc) as rn
from cte1
)
select category, item_purchased,total_amount, rn
from cte2
where rn <4
order by category, total_amount desc;
```

	category text	item_purchased text	total_amount numeric	rn bigint
1	Accessories	Jewelry	10010	1
2	Accessories	Sunglasses	9649	2
3	Accessories	Belt	9635	3
4	Clothing	Blouse	10410	1
5	Clothing	Shirt	10332	2
6	Clothing	Dress	10320	3
7	Footwear	Shoes	9240	1
8	Footwear	Sandals	9200	2
9	Footwear	Boots	9018	3
10	Outerwear	Coat	9275	1
11	Outerwear	Jacket	9249	2

9. **Repeat Buyer Subscription Analysis**: assessed whether customers with more than 5 previous purchases are more likely to subscribe.

```
select  subscription_status, count(customer_id) as customer_count
from customer
where previous_purchases > 5
group by subscription_status;
```

	subscription_status text	customer_count bigint
1	No	2518
2	Yes	958

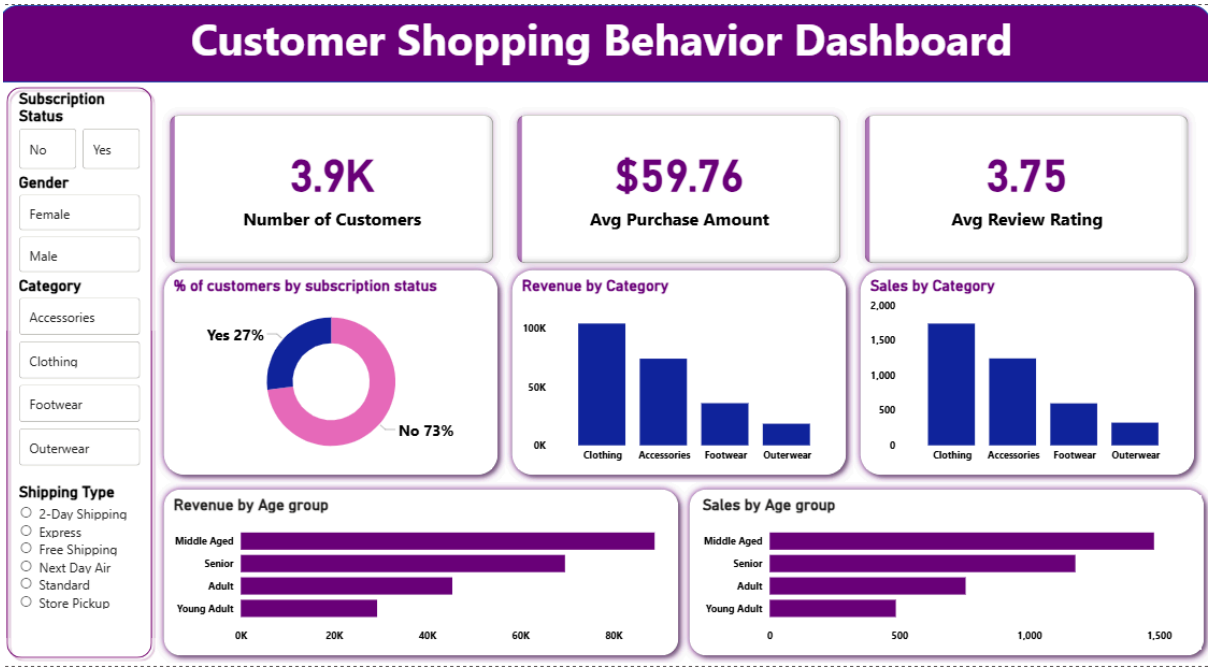
10. **Age-Group Revenue Analysis**: calculated the revenue contribution of each customer age group.

```
select age_group, sum(purchase_amount) as total_revenue
from customer
group by age_group
order by total_revenue desc
```

	age_group text	total_revenue numeric
1	Middle Aged	88833
2	Senior	69590
3	Adult	45400
4	Young Adult	29258

5. Data Presentation using Power BI

Finally, i built an interactive dashboard using power bi to present insights visually



6. Business Recommendation

- **Boost Subscription(Currently at 27%)**- Fewer customers (both loyal and returning) are subscribing, this is possibly due to unattractive benefits of subscribing or poor communications with respect to the benefits of subscribing
- **Customer Loyalty Programs -** Reward repeat buyers to move them into the “Loyal” segment
- **Focus on Clothing Sales** - Clothing generates the highest revenue and sales; prioritizing promotions, inventory, and marketing for this category can maximize ROI.
- **Focused Advertisements -** Advertisement/ Marketing efforts should be focused on top-selling products within each category (e.g., Accessories: Sunglasses, Belt; Clothing: Blouse, Shirt, Dress; Footwear: Shoes, Sandals, Boots; Outerwear: Coat, Jacket) and also on high review ratings commodities (e.g., Gloves, Sandals, Boots, Hat, Skirt) to boost visibility to the right audience and increase sales.
- **Target Middle-Aged and Senior Segments -** These age groups contribute the most to revenue; tailored campaigns and personalized offers can further increase engagement and spending.