Customer Shopping Behavior Analysis

1. Project Objective

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. Data Overview

- 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

Cleaned and prepared data in Python to build a strong analytical foundation:

- **Data Loading:** Imported the dataset using pandas.

- **Initial Exploration:** Used df.info()to check structure and .describe()for summary statistics.

```
# to check the summary of statistics of all the columns
df.describe(include='all')
```

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	3
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	

Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
3863.000000	3900	3900	3900	3900	3900.000000	3900	3900
NaN	2	6	2	2	NaN	6	7
NaN	No	Free Shipping	No	No	NaN	PayPal	Every 3 Months
NaN	2847	675	2223	2223	NaN	677	584
3.750065	NaN	NaN	NaN	NaN	25.351538	NaN	NaN
0.716983	NaN	NaN	NaN	NaN	14.447125	NaN	NaN
2.500000	NaN	NaN	NaN	NaN	1.000000	NaN	NaN
3.100000	NaN	NaN	NaN	NaN	13.000000	NaN	NaN
3.800000	NaN	NaN	NaN	NaN	25.000000	NaN	NaN
4.400000	NaN	NaN	NaN	NaN	38.000000	NaN	NaN
5.000000	NaN	NaN	NaN	NaN	50.000000	NaN	NaN

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

```
# for checking the sum of missing values in the data
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
```

```python
# Impute missing review ratings by filling NaNs with the median rating of each category
df["Review Rating"] = df.groupby("Category")["Review Rating"].transform(lambda x: x.fillna(x.median()))
```

```python
# lets check whether the null values filled.
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.

```python
# lets convert all the the column names to snake case
# it makes the code cosistent, easier to read and most importantly with snake casing
# it is easier to reference code in SQL or Python without running into errors
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(" ","_")
```

```python
df.columns
```

```
Index(['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', 'previous_purchases',
       'payment_method', 'frequency_of_purchases'],
      dtype='object')
```

```python
# Renaming a column name
df = df.rename(columns = {'purchase_amount_(usd)':'purchase_amount'})
```

- **Feature Engineering:**
  - Created **age_group** column by binning customer ages.
  - Created **purchase_frequency_days** column from purchase data.

```python
# create a column age_group
# creating labels for qcut
labels=["Young Adult","Adult","Middle-aged","Senior"]

# qcut > it will split the ages into four equal sized groups based on
# the data distribution and assigns the labels we define
df["age_group"]=pd.qcut(df["age"],q=4,labels=labels)
```

```python
# create column purchase_frequency_days
# map frequency labels to numeric day values
frequency_mapping={
    'Fortnightly': 14,
    'Weekly': 7,
    'Monthly': 30,
    'Quarterly': 90,
    'Bi-Weekly': 14,
    'Annually': 365,
    'Every 3 Months': 90
}

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

- **Data Consistency Check:** Verified if discount_appliedand promo_code_used were redundant; dropped promo_code_used.

```python
# check if all values in discount_applied and promo_code_used columns are exactly the same
(df["discount_applied"]==df["promo_code_used"]).all()
```

```
np.True_
```

```python
# since both the columns have exactly the same data we can remove any of the column
df=df.drop("promo_code_used",axis=1)
```

- **Database Integration:** Connected Python script to MYSQL and loaded the cleaned DataFrame into the database for SQL analysis.

```
pip install pymysql sqlalchemy
```

```python
from sqlalchemy import create_engine

# MySQL connection
username = "root"
password = "Mdarif%4027"
host = "localhost"
port = "3306"
database = "customer_behavior"

engine = create_engine(f"mysql+pymysql://{"root"}:{"Mdarif%4027"}@{"localhost"}:{"3306"}/{"customer_behavior"}")

# Write DataFrame to MySQL
table_name = "customer"   # choose any table name
df.to_sql(table_name, engine, if_exists="replace", index=False)

# Read back sample
pd.read_sql("SELECT * FROM customer LIMIT 5;", engine)
```

	customer_id	age	gender	item_purchased	category	purchase_amount	location	size	color	season	review_rating	subscription_status	shipping_type	disc
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	

4. Data Analysis using MYSQL

Performed structured analysis in MYSQL to answer key business questions:

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

Result Grid Filter Rows:

gender	total_revenue
Male	157890
Female	75191

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

Result Grid Filter Rows: Export:

customer_id	gender	discount_applied	purchase_amount
2	Male	Yes	64
3	Male	Yes	73
4	Male	Yes	90
7	Male	Yes	85
9	Male	Yes	97
12	Male	Yes	68
13	Male	Yes	72
16	Male	Yes	81
20	Male	Yes	90

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

Result Grid Filter Rows:

item_purchased	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.

Result Grid | Filter Rows:

shipping_type	avg_purchase_amt
Express	60.48
Standard	58.46

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

Result Grid | Filter Rows: | Export: | Wrap

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

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

Result Grid | Filter Rows:

item_purchased	discount_percentage
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.

Result Grid | Filter Rows:

customer_segment	total_counts
Loyal	3116
Returning	701
New	83

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

Result Grid | Filter Rows: | Export: | W

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

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

Result Grid | Filter Rows:

subscription_status	repeat_buyers
Yes	958
No	2518

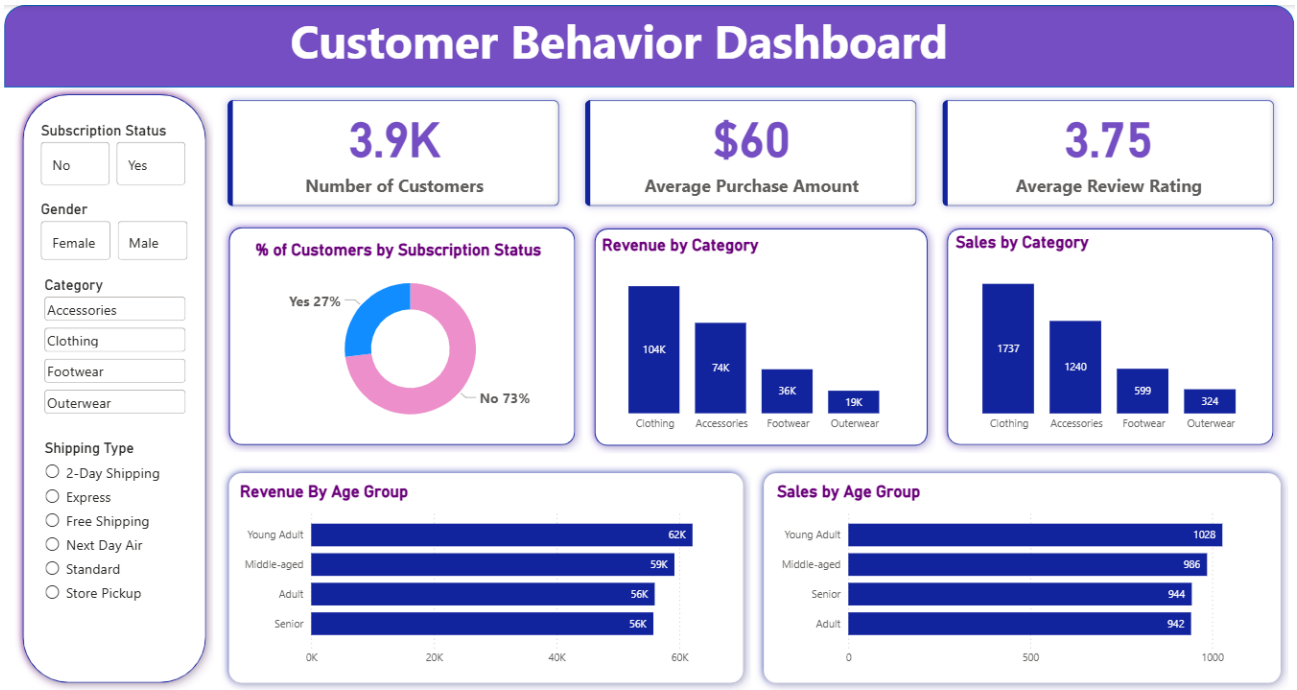
10. **Revenue by Age Group** – Calculated total revenue contribution of each age group.

Result Grid | Filter Rows:

age_group	total_revenue
Senior	55763
Adult	55978
Middle-aged	59197
Young Adult	62143

5. Dashboard in Power BI

Finally, as the **final step** of my end-to-end data analysis project, I built an interactive **Power BI dashboard** to bring all insights together. It highlights key **KPIs** such as revenue, customer growth, and category performance through clear and engaging visual storytelling. The dashboard transforms raw data into meaningful insights that empower smarter, **data-driven business decisions**.



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.