Customer Shopping Behavior Analysis

1. Project Overview

This project focuses on analyzing customer shopping behavior using transactional data from **3,900 purchase records** across multiple product categories. The primary objective is to uncover meaningful insights into customer spending patterns, purchasing preferences, subscription behavior, and product demand.

By leveraging **Python** for data cleaning and exploratory analysis, **SQL** for structured querying, and **Power BI** for visualization, the project provides actionable insights to support strategic decision-making and **customer-centric marketing strategies**. The analysis helps businesses enhance revenue, customer satisfaction, and retention.

2. Problem Statement

A leading retail company seeks to understand **customer shopping behavior** to improve sales performance and long-term loyalty. Key challenges include:

- Variations in purchasing patterns across **demographics**, **product categories**, and **sales channels** (online vs. offline).
- Identifying factors influencing **repeat purchases**, such as:
  - Discounts and promotional offers
  - Product reviews and ratings
  - Seasonal trends
  - Payment and shipping preferences
  - Subscription status

Goal: To enable optimized pricing strategies, improved product offerings, and targeted marketing campaigns.

3. Dataset Summary

Characteristic	Details
Number of Rows	3,900

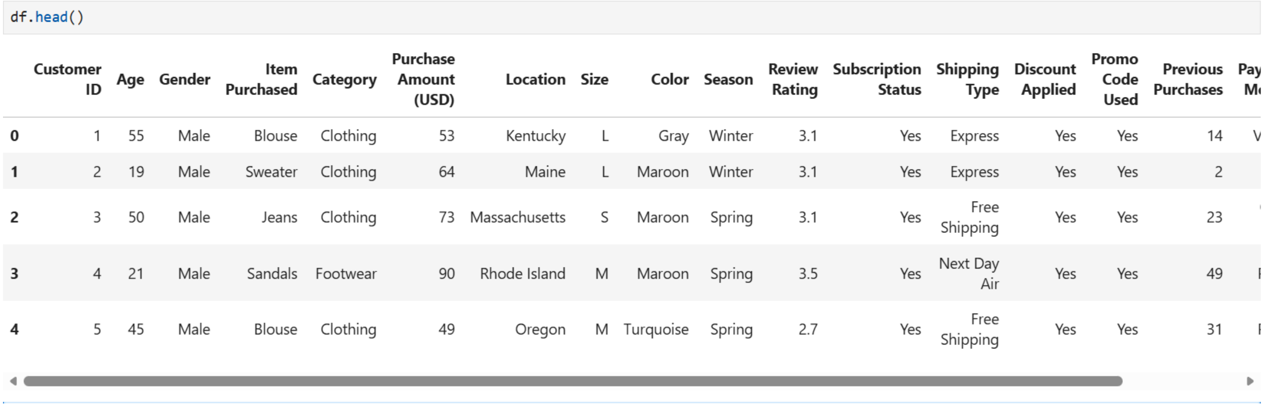
Characteristic	Details
Number of Columns	18
Key Features	Customer Demographics: Age, Gender, Location, Subscription Status; Purchase Details: Item Purchased, Product Category, Purchase Amount, Season, Size, Color; Shopping Behavior Indicators: Discount Applied, Promo Code Used, Previous Purchases, Purchase Frequency, Review Rating, Shipping Type
Missing Data	37 missing values in Review Rating (imputed using median per product category)

4. Exploratory Data Analysis (EDA) Using Python

EDA was conducted to understand the dataset, ensure quality, and prepare data for deeper analysis.

4.1 Data Loading

- Dataset imported using **Pandas** for data manipulation.

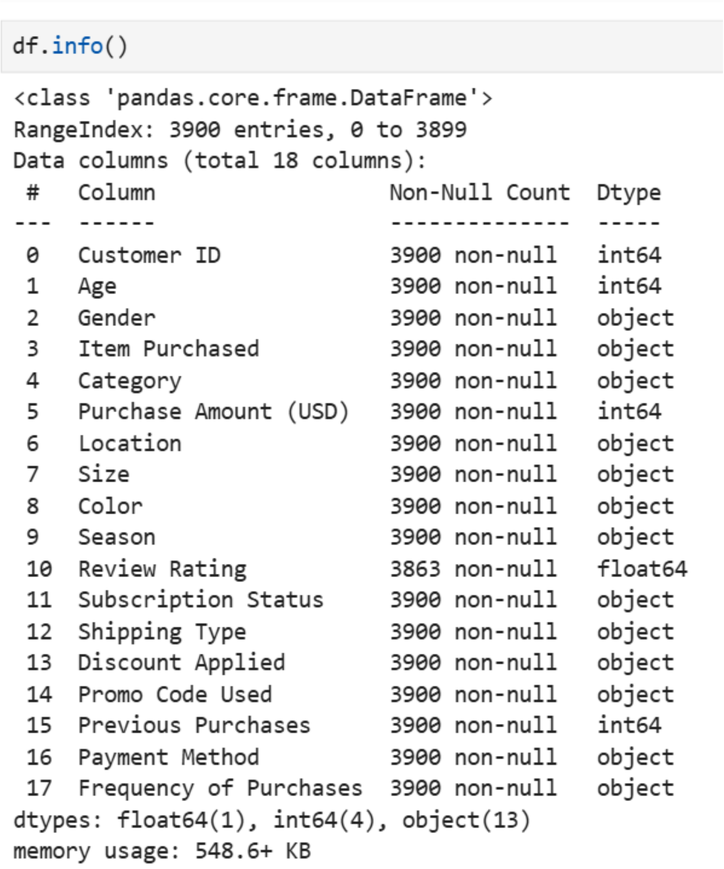


```
df.head()
```

	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	Pay M
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	V
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	F
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	F

4.2 Initial Data Exploration

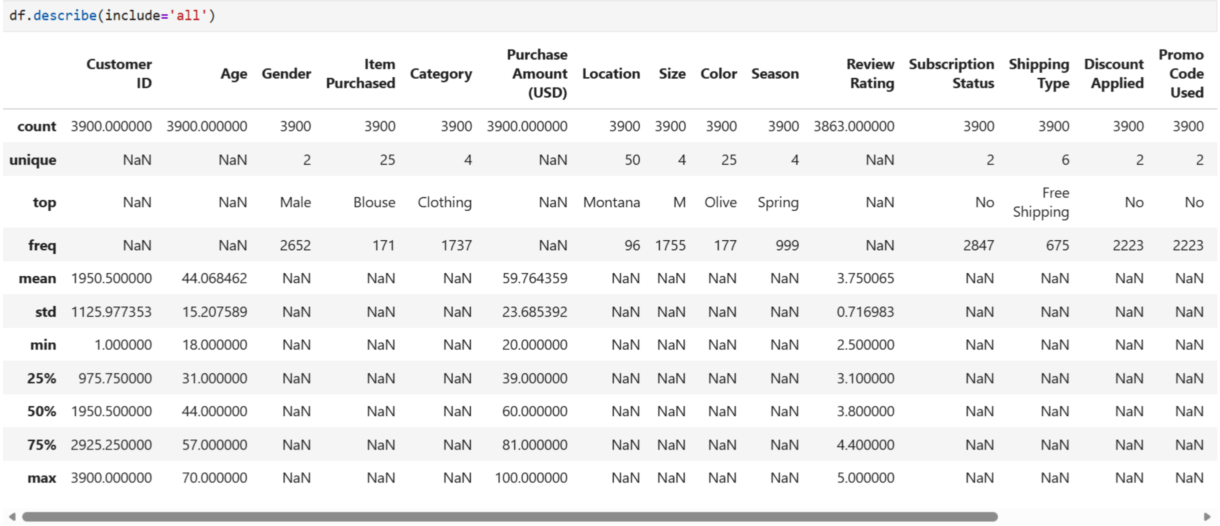
- df.info() examined dataset structure, data types, and missing values.



```
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
```

- df.describe() generated summary statistics for numerical variables.

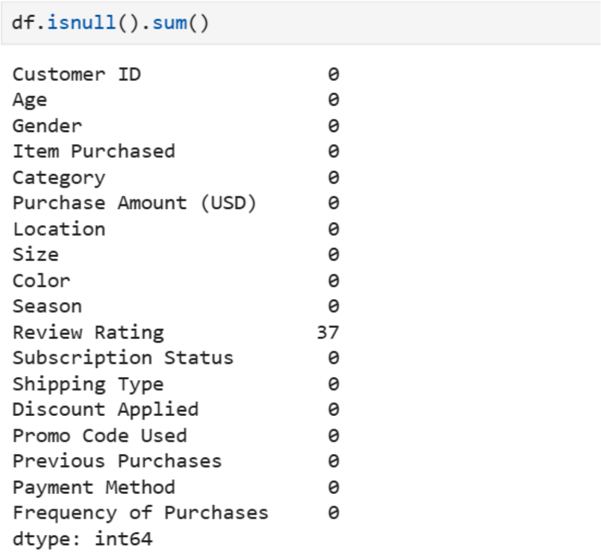


```
df.describe(include='all')
```

	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
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

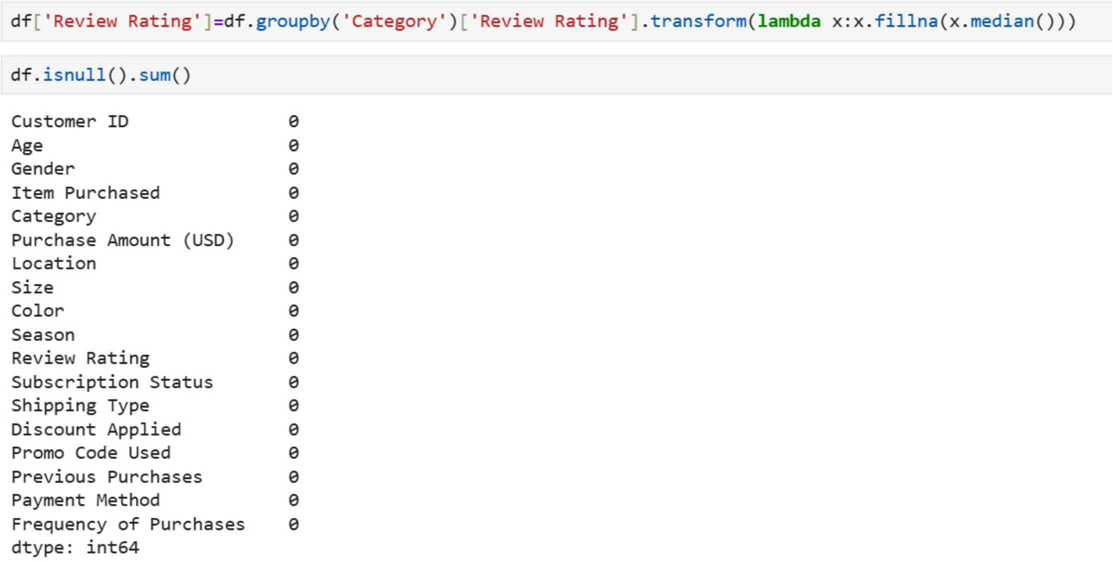
4.3 Missing Data Handling

- Missing values in **Review Rating** were imputed with the **median per product category**.



```
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
```



```
df['Review Rating']=df.groupby('Category')['Review Rating'].transform(lambda x:x.fillna(x.median()))
```

```
df.isnull().sum()
```

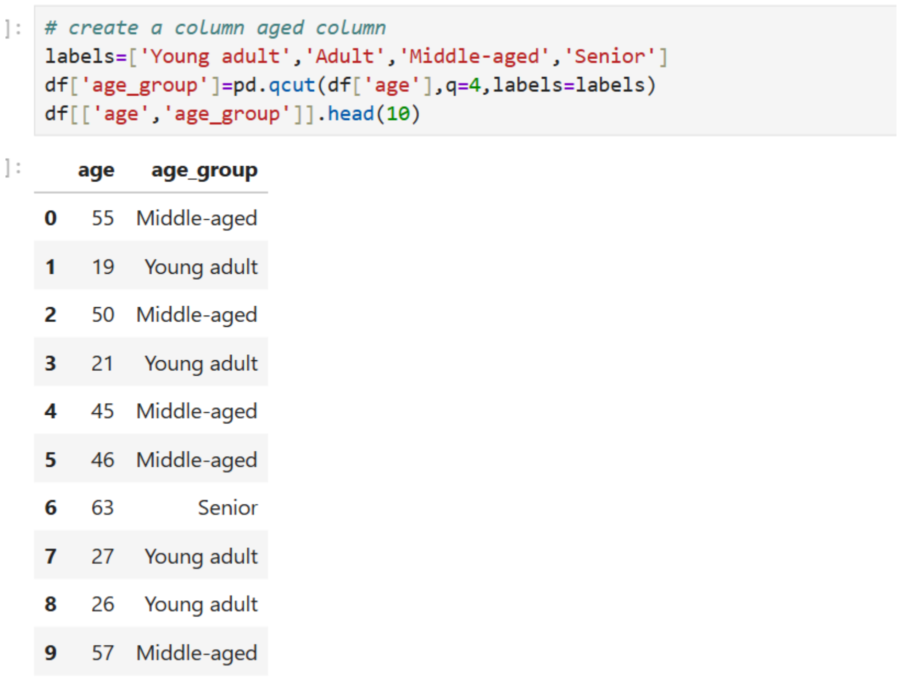
```
Customer ID               0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             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
```

4.4 Column Standardization

- Column names converted to **snake_case** for consistency and readability.

4.5 Feature Engineering

- **age_group**: Customer ages binned into predefined ranges.

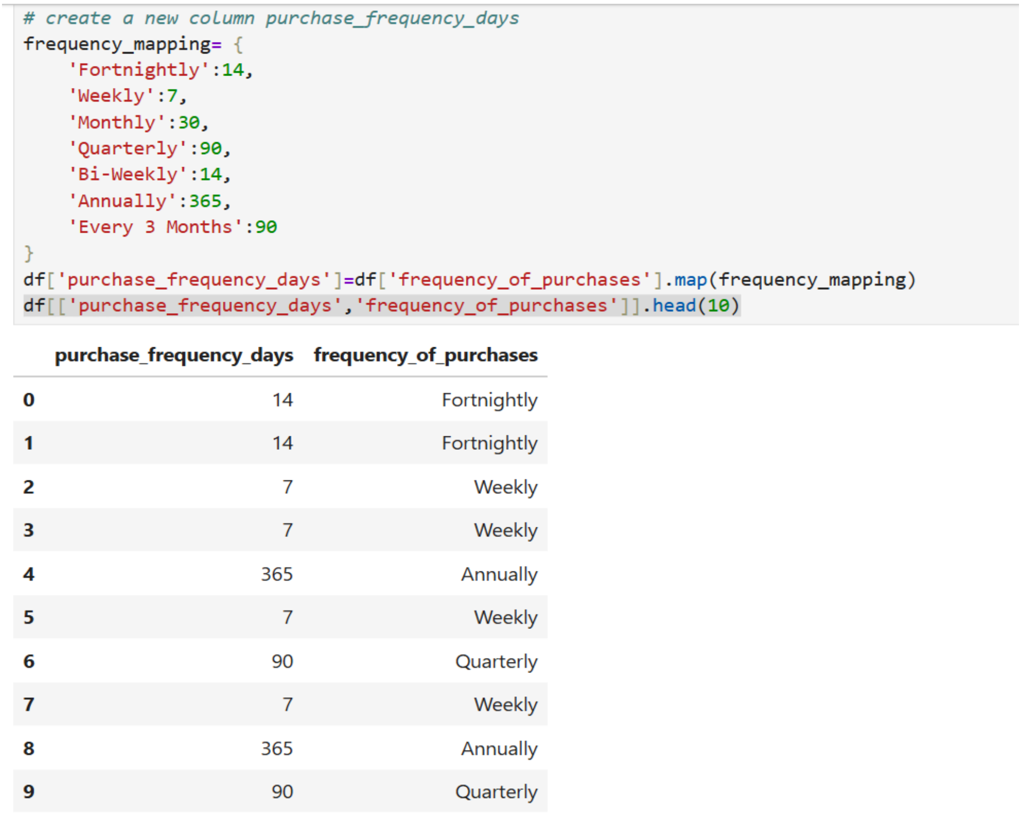


```
]: # create a column aged column
   labels=['Young adult','Adult','Middle-aged','Senior']
   df['age_group']=pd.qcut(df['age'],q=4,labels=labels)
   df[['age','age_group']].head(10)
```

]:

	age	age_group
0	55	Middle-aged
1	19	Young adult
2	50	Middle-aged
3	21	Young adult
4	45	Middle-aged
5	46	Middle-aged
6	63	Senior
7	27	Young adult
8	26	Young adult
9	57	Middle-aged

- **purchase_frequency_dates**: Derived to analyze buying patterns over time.



```
# create a new column purchase_frequency_days
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)
df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

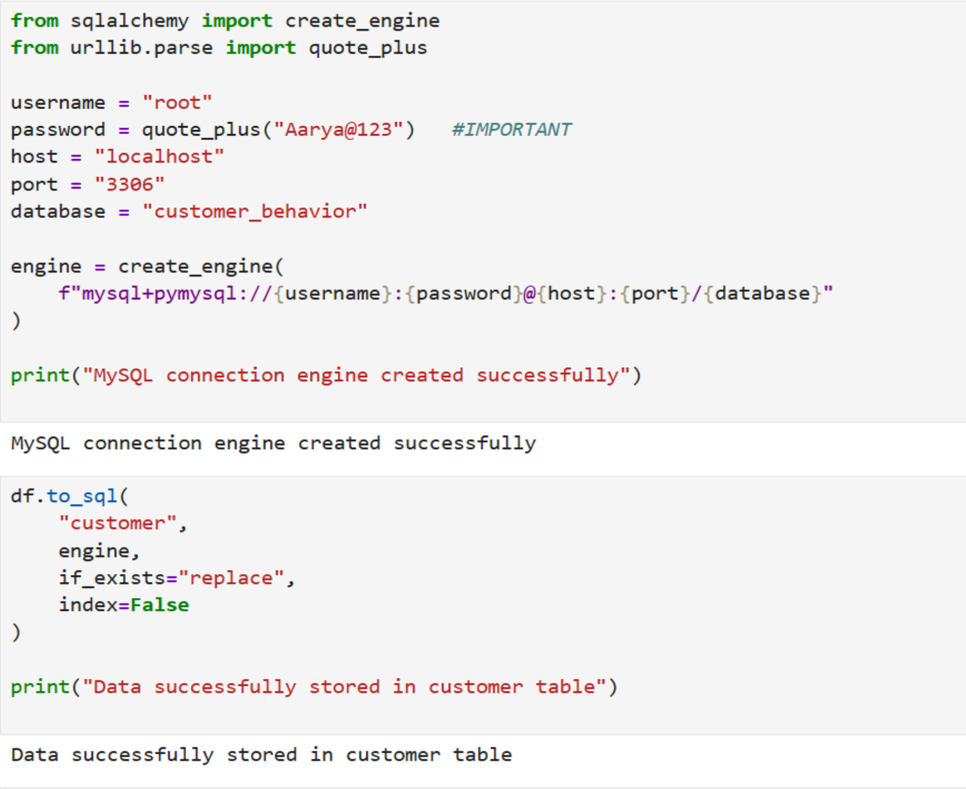
	purchase_frequency_days	frequency_of_purchases
0	14	Fortnightly
1	14	Fortnightly
2	7	Weekly
3	7	Weekly
4	365	Annually
5	7	Weekly
6	90	Quarterly
7	7	Weekly
8	365	Annually
9	90	Quarterly

4.6 Data Consistency Check

- Redundancy detected between discount_applied and promo_code_used.
- **Promo code column removed** to avoid duplication.

4.7 Database Integration

- Cleaned dataset connected to **MySQL Workbench** for SQL analysis and **Power BI** visualization.



```python
from sqlalchemy import create_engine
from urllib.parse import quote_plus

username = "root"
password = quote_plus("Aarya@123")   #IMPORTANT
host = "localhost"
port = "3306"
database = "customer_behavior"

engine = create_engine(
    f"mysql+pymysql://{username}:{password}@{host}:{port}/{database}"
)

print("MySQL connection engine created successfully")
```

```
MySQL connection engine created successfully
```

```python
df.to_sql(
    "customer",
    engine,
    if_exists="replace",
    index=False
)

print("Data successfully stored in customer table")
```

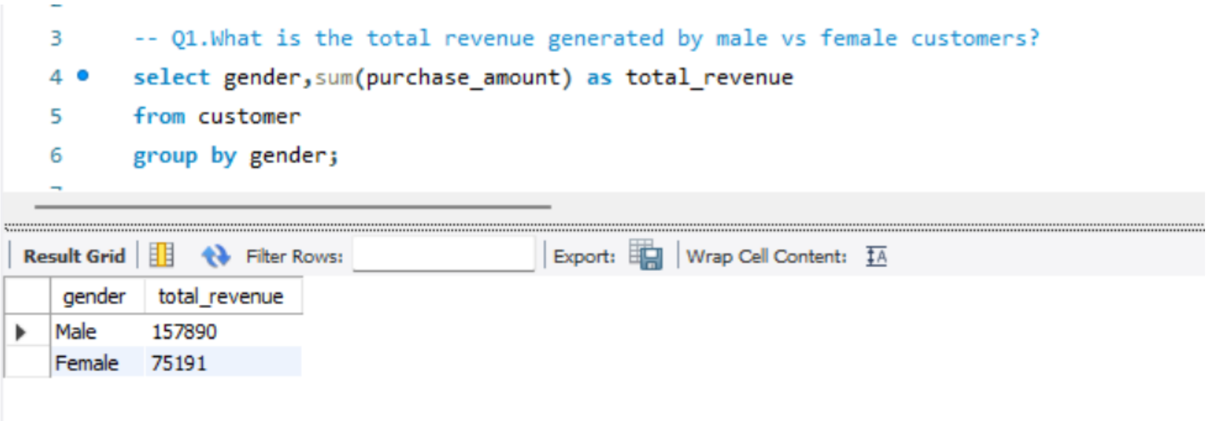
```
Data successfully stored in customer table
```

5. Data Analysis Using SQL

SQL queries were used to extract insights and validate patterns observed in Python EDA.

Key Analyses

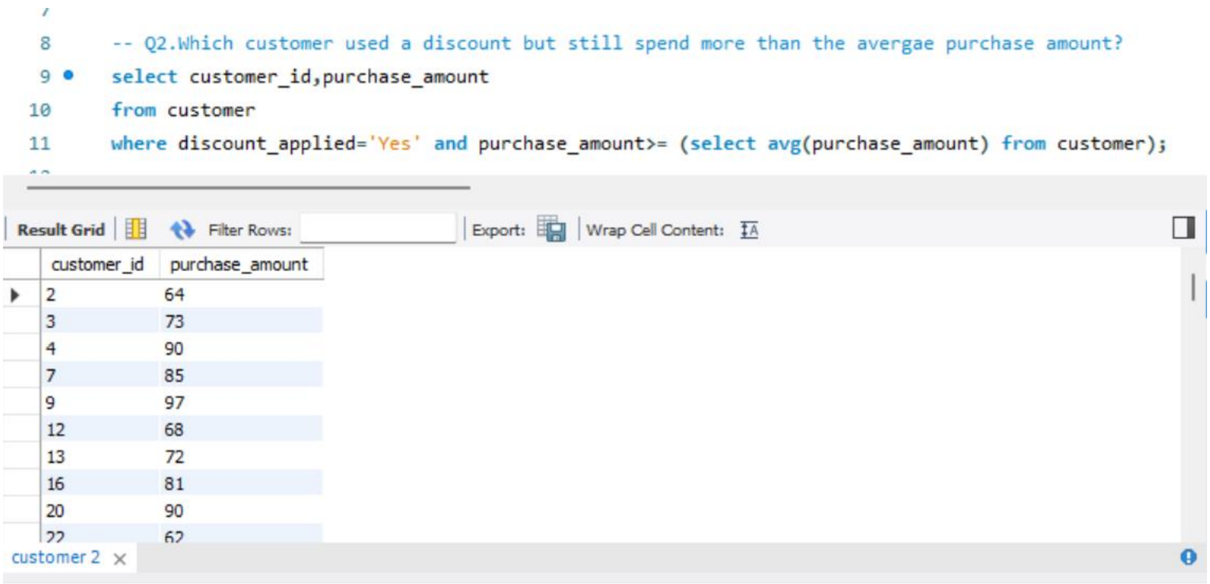
1. **Revenue by Gender:** Compared total revenue between male and female customers.



```sql
-- Q1.What is the total revenue generated by male vs female customers?
select gender,sum(purchase_amount) as total_revenue
from customer
group by gender;
```

gender	total_revenue
Male	157890
Female	75191

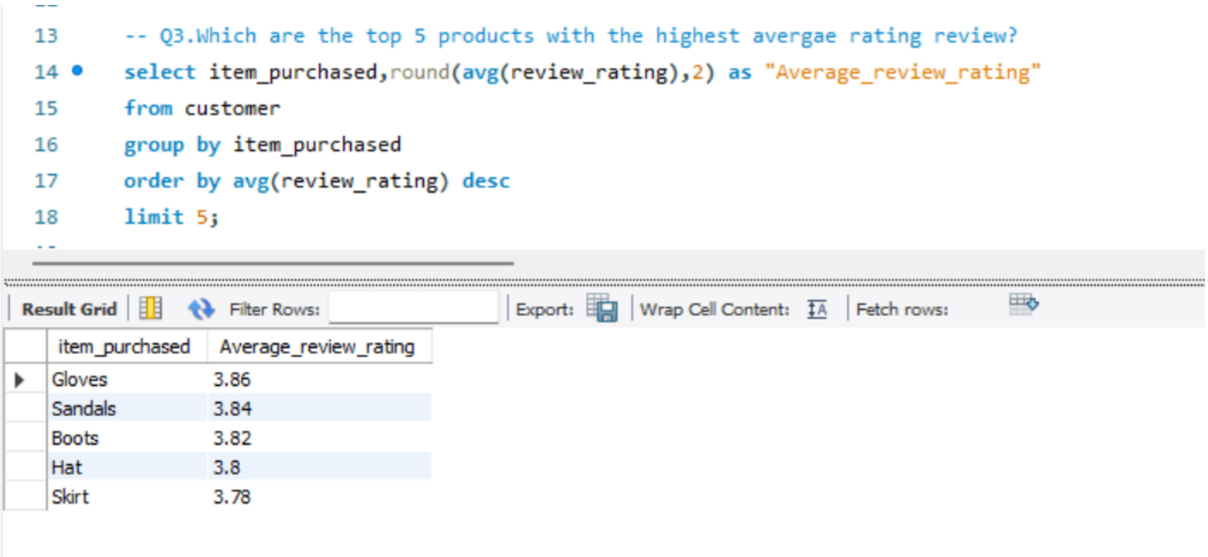
2. **High-Spending Discount Users:** Identified customers availing discounts but spending above average.



```
-- Q2.Which customer used a discount but still spend more than the avergae purchase amount?
select customer_id,purchase_amount
from customer
where discount_applied='Yes' and purchase_amount>= (select avg(purchase_amount) from customer);
```

customer_id	purchase_amount
2	64
3	73
4	90
7	85
9	97
12	68
13	72
16	81
20	90
22	62

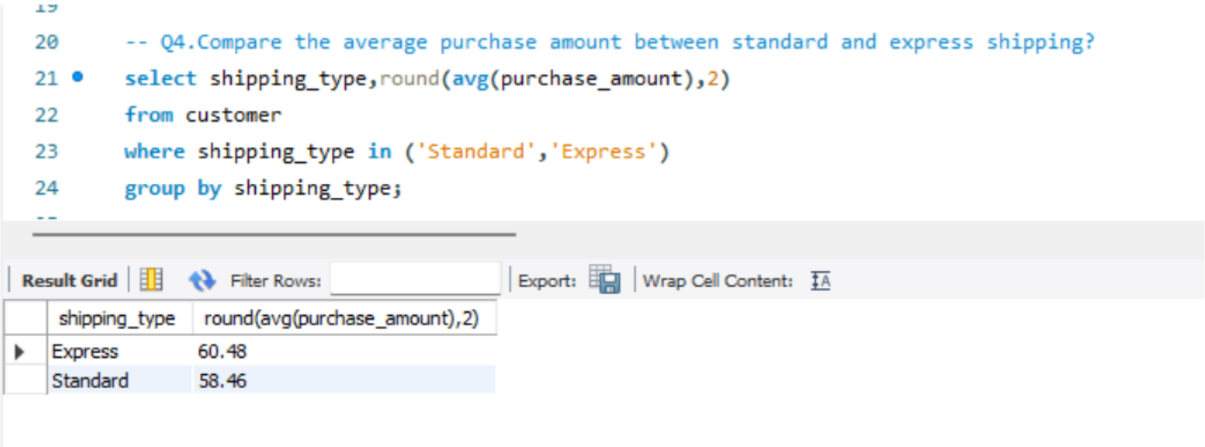
3. **Top 5 Products by Rating:** Ranked products using average customer review ratings.



```
-- Q3.Which are the top 5 products with the highest avergae rating review?
select item_purchased,round(avg(review_rating),2) as "Average_review_rating"
from customer
group by item_purchased
order by avg(review_rating) desc
limit 5;
```

item_purchased	Average_review_rating
Gloves	3.86
Sandals	3.84
Boots	3.82
Hat	3.8
Skirt	3.78

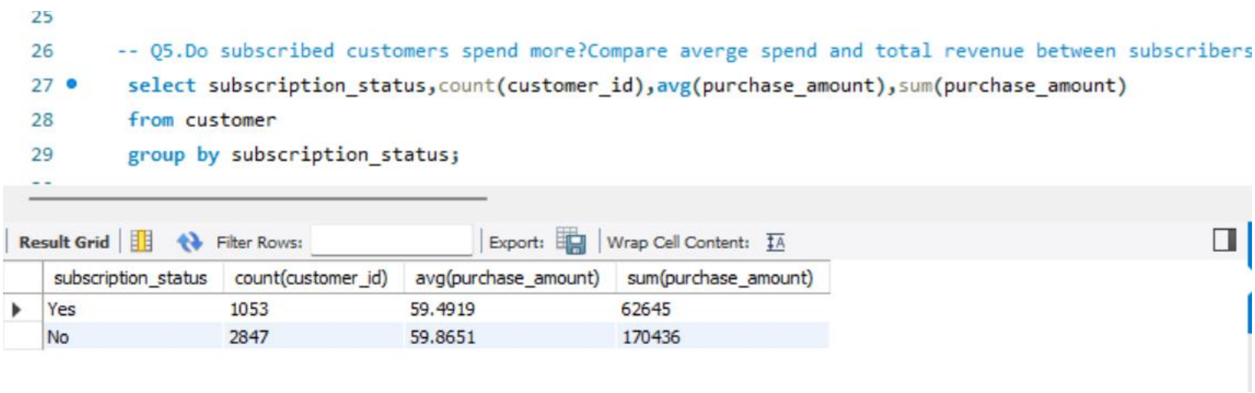
4. **Impact of Shipping Type:** Compared purchase amounts between standard vs express shipping.



```
-- Q4.Compare the average purchase amount between standard and express shipping?
select shipping_type,round(avg(purchase_amount),2)
from customer
where shipping_type in ('Standard','Express')
group by shipping_type;
```

shipping_type	round(avg(purchase_amount),2)
Express	60.48
Standard	58.46

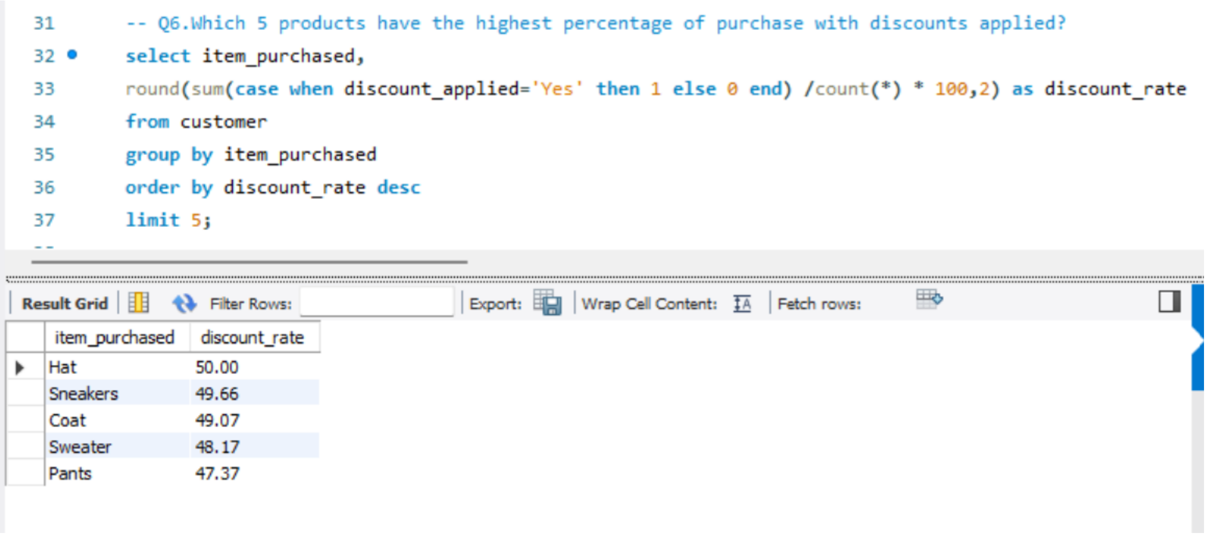
5. **Subscribers vs Non-Subscribers:** Compared average spending and total revenue contribution.



```sql
-- Q5.Do subscribed customers spend more?Compare averge spend and total revenue between subscribers
select subscription_status,count(customer_id),avg(purchase_amount),sum(purchase_amount)
from customer
group by subscription_status;
```

subscription_status	count(customer_id)	avg(purchase_amount)	sum(purchase_amount)
Yes	1053	59.4919	62645
No	2847	59.8651	170436

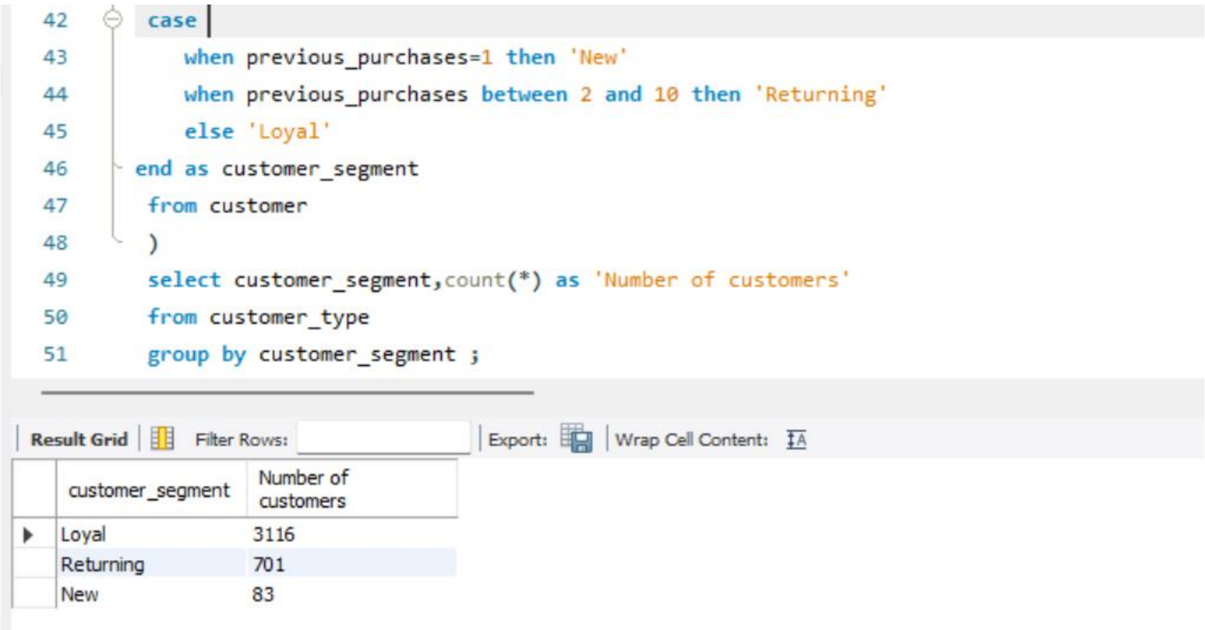
6. **Discount-Dependent Products:** Identified products most influenced by discounts.



```sql
-- Q6.Which 5 products have the highest percentage of purchase with discounts applied?
select item_purchased,
round(sum(case when discount_applied='Yes' then 1 else 0 end) /count(*) * 100,2) as discount_rate
from customer
group by item_purchased
order by discount_rate desc
limit 5;
```

item_purchased	discount_rate
Hat	50.00
Sneakers	49.66
Coat	49.07
Sweater	48.17
Pants	47.37

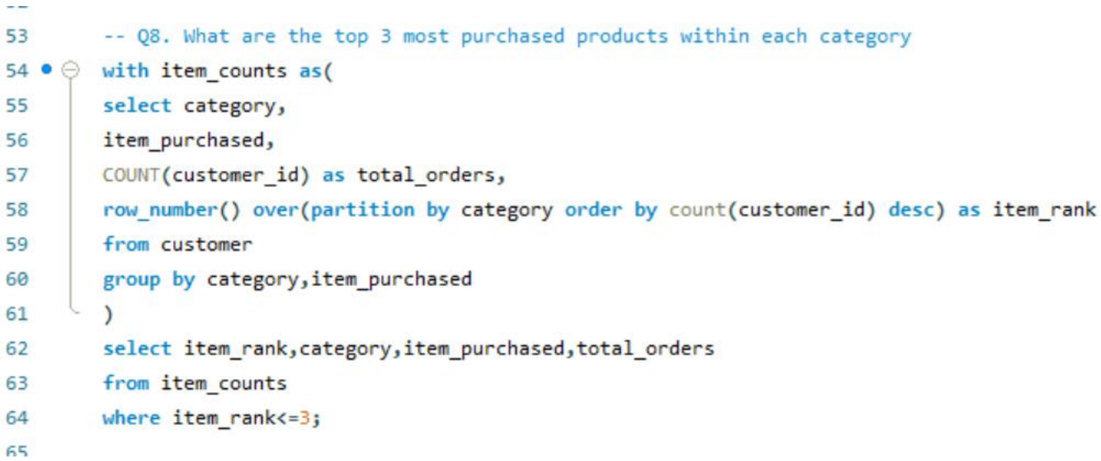
7. **Customer Segmentation:** Classified customers as New, Returning, or Loyal based on purchase frequency.



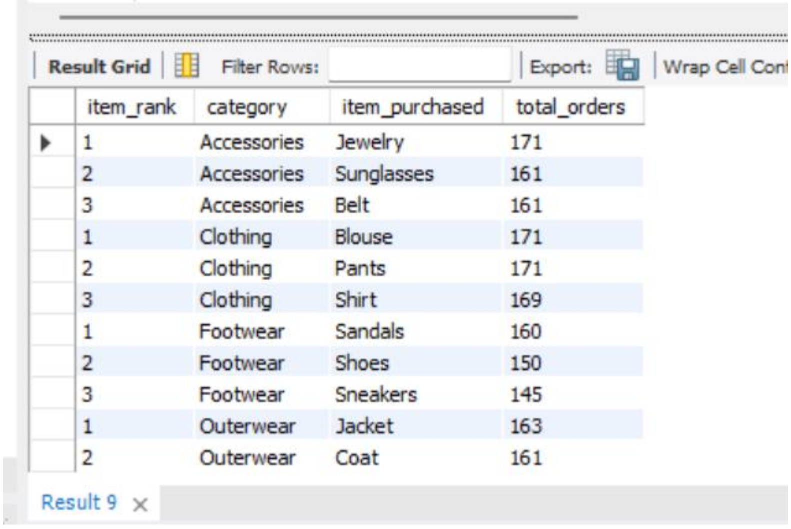
```
  case
     when previous_purchases=1 then 'New'
     when previous_purchases between 2 and 10 then 'Returning'
     else 'Loyal'
end as customer_segment
  from customer
  )
  select customer_segment,count(*) as 'Number of customers'
  from customer_type
  group by customer_segment ;
```

customer_segment	Number of customers
Loyal	3116
Returning	701
New	83

8. **Top 3 Products per Category:** Highlighted category-wise bestsellers.

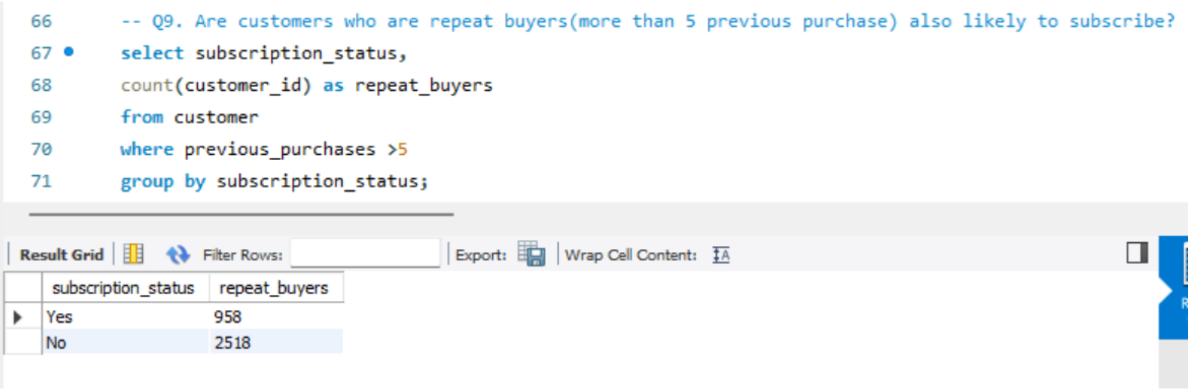


```
-- Q8. What are the top 3 most purchased products within each category
with item_counts as(
select category,
item_purchased,
COUNT(customer_id) as total_orders,
row_number() over(partition by category order by count(customer_id) desc) as item_rank
from customer
group by category,item_purchased
)
select item_rank,category,item_purchased,total_orders
from item_counts
where item_rank<=3;
```



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

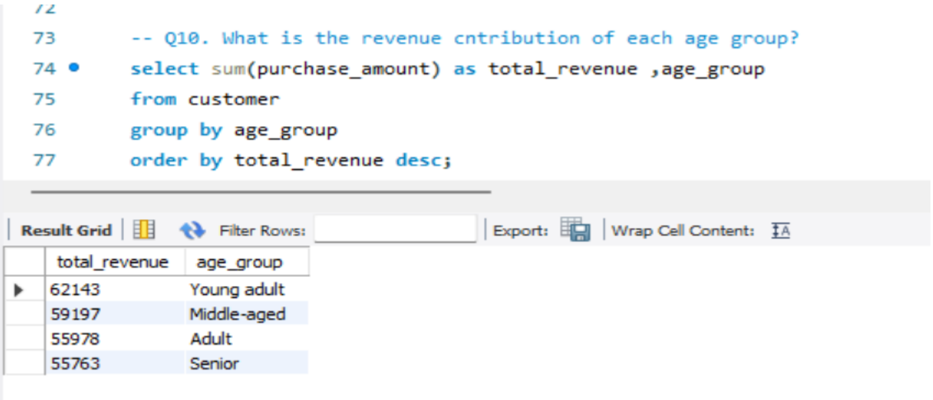
9. **Repeat Buyers & Subscription Behavior:** Evaluated subscription likelihood among frequent buyers.



```sql
-- Q9. Are customers who are repeat buyers(more than 5 previous purchase) also likely to subscribe?
select subscription_status,
count(customer_id) as repeat_buyers
from customer
where previous_purchases >5
group by subscription_status;
```

subscription_status	repeat_buyers
Yes	958
No	2518

10. **Revenue Contribution by Age Group:** Identified high-revenue demographic segments.



```sql
-- Q10. What is the revenue cntribution of each age group?
select sum(purchase_amount) as total_revenue ,age_group
from customer
group by age_group
order by total_revenue desc;
```

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

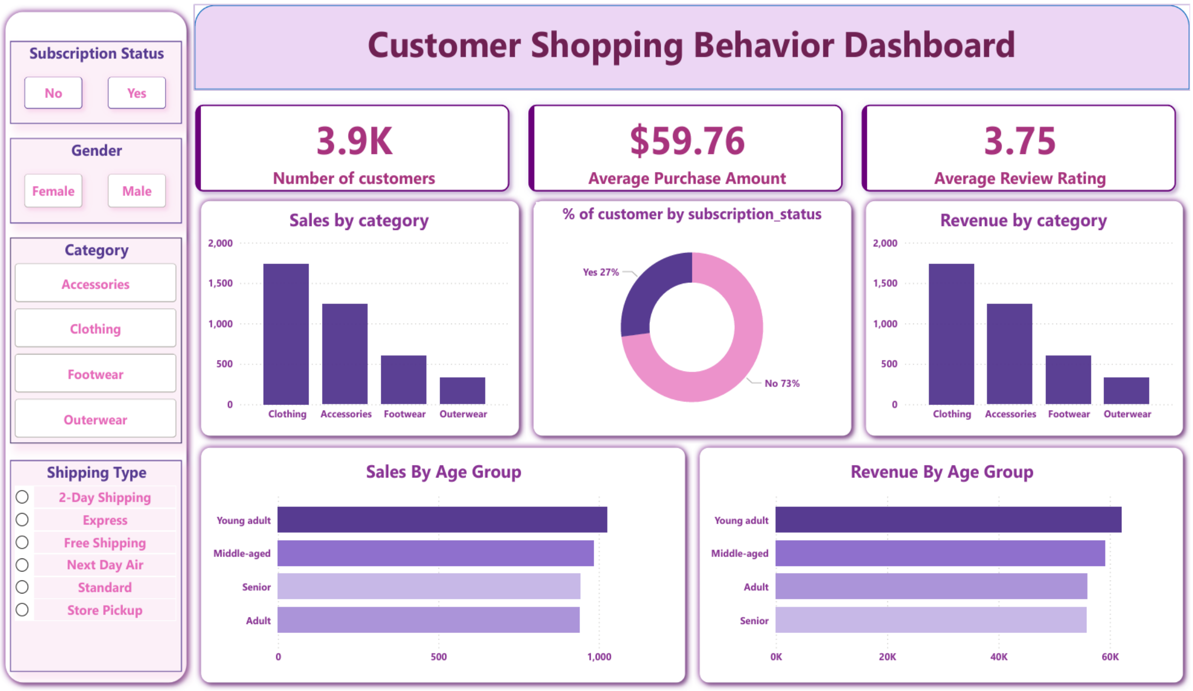
6. Power BI Dashboard

An interactive dashboard was developed to visualize insights for decision-makers.

Dashboard Features:

- **Customer Overview:** Displays total customers, average purchase amount and average review rating.
- **Revenue Analysis:** Shows revenue by category, and can include breakdowns by gender, age, and subscription status.
- **Spending Behavior:** Highlights the relationship between discounts, shipping type, and purchase amounts.
- **Product Performance:** Lists top-rated and bestselling products to identify key performers.
- **Customer Segmentation:** Categorizes customers as New, Returning, and Loyal customers for targeted marketing.

Interactive filters and drill-down features enable stakeholders to explore data dynamically.



Customer Shopping Behavior Dashboard

3.9K Number of customers

$59.76 Average Purchase Amount

3.75 Average Review Rating

7. Business Recommendations

1. **Boost Subscription Adoption:** Offer exclusive benefits (discounts, faster shipping, early access) to encourage subscriptions.
2. **Strengthen Loyalty Programs:** Reward repeat buyers to enhance lifetime value.
3. **Review Discount Strategy:** Balance sales volume growth with profit margin control.
4. **Strategic Product Positioning:** Promote top-rated and bestselling products in campaigns.
5. **Targeted Marketing:** Focus on high-revenue age groups and express-shipping users to improve ROI.

8. Executive Summary

The analysis of 3,900 customer transactions provided deep insights into:

- Demographic-based spending behavior
- Subscription patterns and repeat purchase tendencies
- Product performance and discount dependency

Using **Python** for EDA, **SQL** for structured analysis, and **Power BI** for visualization, the project successfully generated actionable insights that inform marketing strategy, loyalty programs, and sales optimization.

9. Conclusion

- Customer spending behavior varies across **age groups** and **subscription status**.
- **Subscribers and repeat customers** contribute higher revenue.
- Discounts influence purchases but require careful management to maintain profit margins.
- Certain products are highly **discount-dependent**.
- Targeted marketing and loyalty programs can improve **customer retention** and revenue.

10. Future Scope

- **Predictive Analysis:** Implement ML models to forecast purchasing behavior and spending trends.
- **Customer Churn Analysis:** Identify customers at risk and design retention strategies.
- **Advanced Customer Segmentation:** Apply clustering for detailed behavioral segments.
- **Real-Time Data Integration:** Continuous monitoring of transactions for timely insights.
- **Personalized Recommendations:** Suggest products based on past behavior.
- **Marketing Campaign Evaluation:** Analyze promotional effectiveness for future optimization.
- **Integration with Social Media:** Incorporate social sentiment data to understand brand perception and buying triggers.

11. References / Acknowledgements

1. **Dataset Source:** Transactional dataset obtained from publicly available sources for educational purposes.
2. **Tools and Libraries:** Analysis and visualization implemented independently using **Python (Pandas)**, **SQL (MySQL Workbench)**, and **Power BI**.
3. **Learning Resources:** General guidance and learning references were consulted from publicly available tutorials and documentation to understand tool functionalities and best practices.