Customer Shopping Behavior Analysis

1. Project Overview

This project analyzes customer purchasing behavior using transactional sales data to generate actionable business insights.

The objective of this project is to build a complete end-to-end data analytics pipeline — starting from raw data processing in Python, performing structured business analysis in SQL Server, and finally developing an interactive dashboard in Power BI for decision-making.

The project demonstrates practical skills in ETL, SQL analytics, and data visualization.

2. Dataset Summary

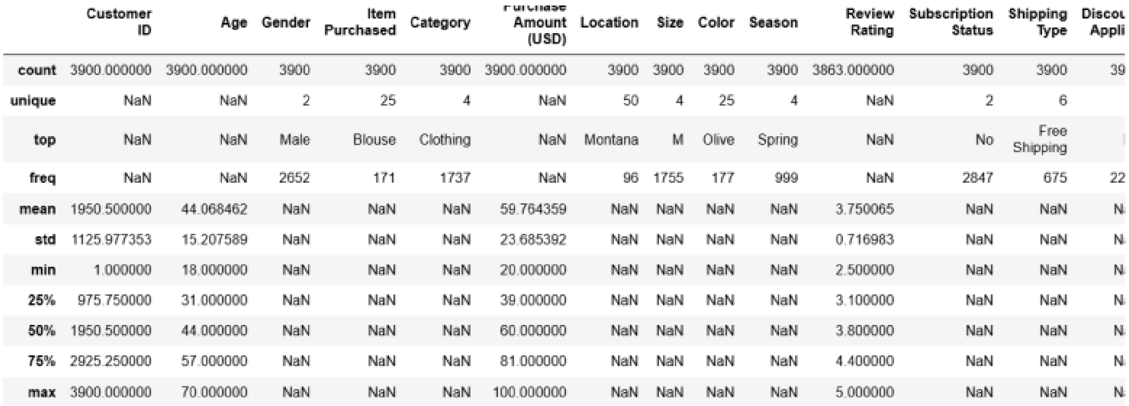
- Rows: 3,900
- Columns: 18
- Key Features:
  - Customer demographics (Age, Gender, Location, Subscription Status)
  - Purchase details (Item Purchased, Category, Purchase Amount, Season, Size, Color)
  - Shopping behavior (Discount Applied, Promo Code Used, Previous Purchases, Frequency of Purchases, Review Rating, Shipping Type)
- Missing Data: 37 values in Review Rating column

3. Exploratory Data Analysis using Python

We began data preparation and cleaning using Python in Jupyter Notebook.

● Data Loading

Imported dataset using pandas library.

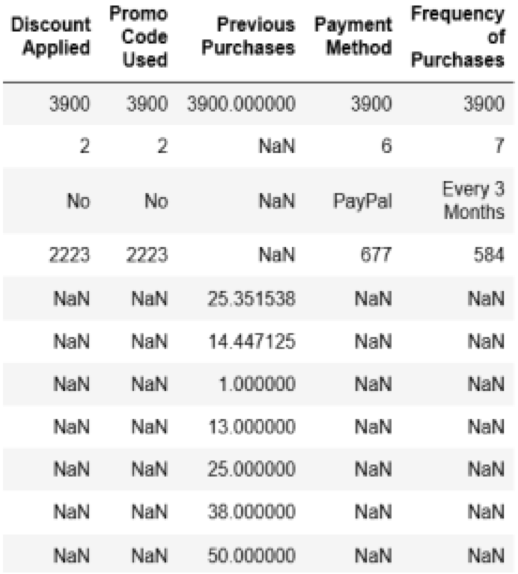


	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discou Appli
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	39
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2	6	
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No	Free Shipping	
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847	675	22
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN	NaN	N
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN	NaN	N
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN	NaN	N
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN	NaN	N
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN	NaN	N
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN	NaN	N
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN	NaN	N

- **Initial Exploration**

Used:

- df.info() to understand data structure
- df.describe() for statistical summary
- df.head() for preview



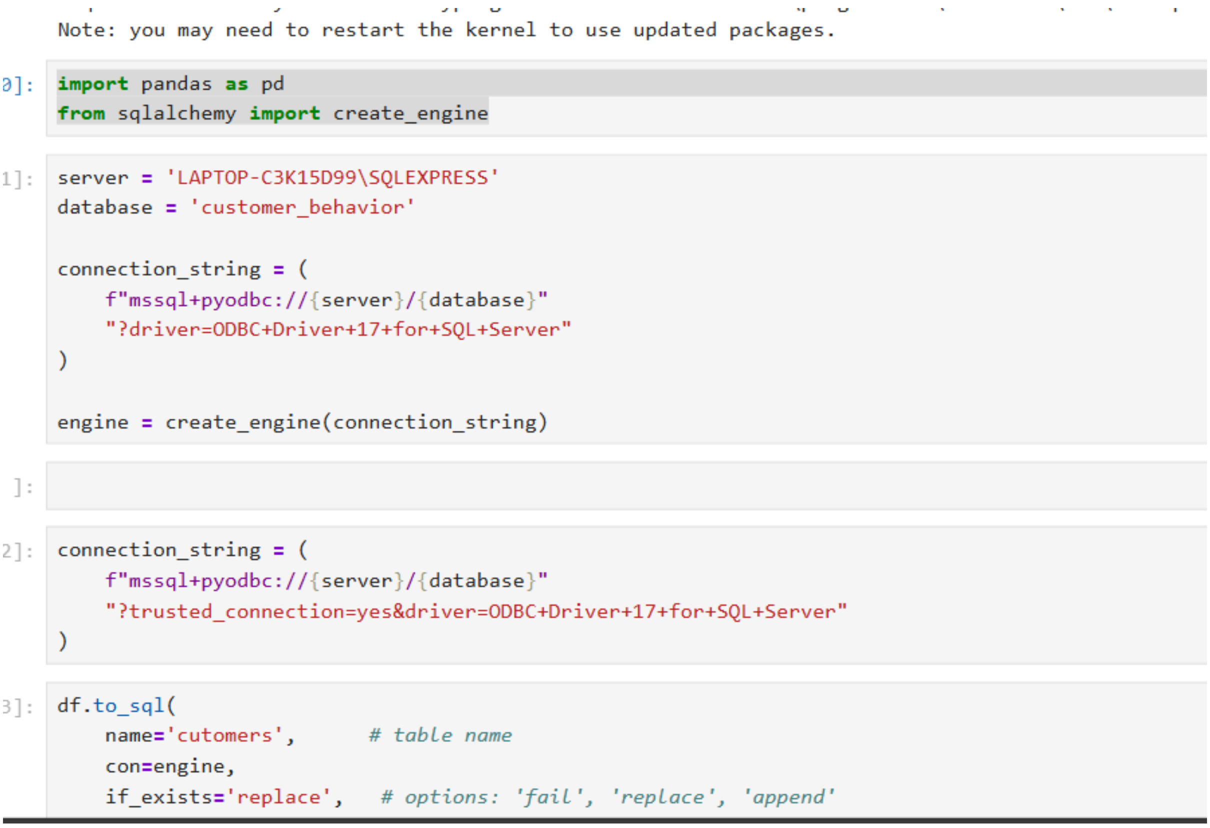
Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
3900	3900	3900.000000	3900	3900
2	2	NaN	6	7
No	No	NaN	PayPal	Every 3 Months
2223	2223	NaN	677	584
NaN	NaN	25.351538	NaN	NaN
NaN	NaN	14.447125	NaN	NaN
NaN	NaN	1.000000	NaN	NaN
NaN	NaN	13.000000	NaN	NaN
NaN	NaN	25.000000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

- **Missing Data Handling**
  - Identified null values
  - Imputed or cleaned missing values
  - Removed duplicates

- **Data Transformation**
  - Standardized column names
  - Converted date columns
  - Created new features (if any)
  - Performed type conversions

- **Database Integration**

Connected Python to **SQL Server** and exported the cleaned DataFrame into the database.



```
Note: you may need to restart the kernel to use updated packages.
```

```
import pandas as pd
from sqlalchemy import create_engine
```

```
server = 'LAPTOP-C3K15D99\SQLEXPRESS'
database = 'customer_behavior'

connection_string = (
    f"mssql+pyodbc://{server}/{database}"
    "?driver=ODBC+Driver+17+for+SQL+Server"
)

engine = create_engine(connection_string)
```

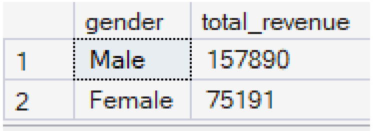
```
connection_string = (
    f"mssql+pyodbc://{server}/{database}"
    "?trusted_connection=yes&driver=ODBC+Driver+17+for+SQL+Server"
)
```

```
df.to_sql(
    name='cutomers',        # table name
    con=engine,
    if_exists='replace',    # options: 'fail', 'replace', 'append'
```

4. Data Analysis using SQL Server (Business Transactions)

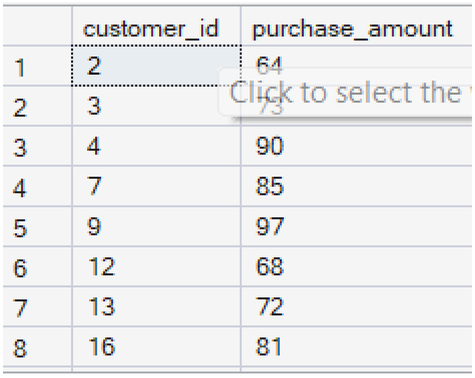
After loading the cleaned dataset into SQL Server, analytical queries were written to answer business questions.

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



	gender	total_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.

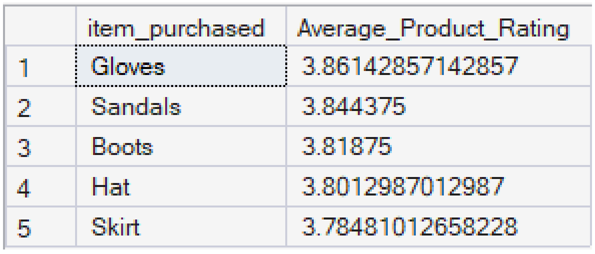


	customer_id	purchase_amount
1	2	64
2	3	73
3	4	90
4	7	85
5	9	97
6	12	68
7	13	72
8	16	81

Click to select the

---

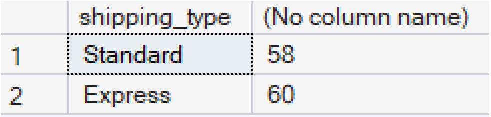
3. Top 5 Products by Rating – Found products with the highest average review ratings.



	item_purchased	Average_Product_Rating
1	Gloves	3.86142857142857
2	Sandals	3.844375
3	Boots	3.81875
4	Hat	3.8012987012987
5	Skirt	3.78481012658228

---

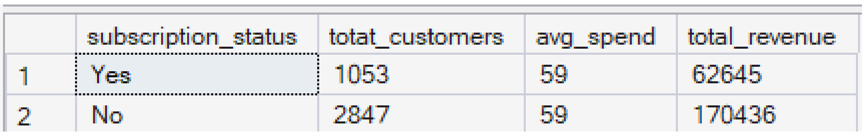
4. Shipping Type Comparison – Compared average purchase amounts between Standard and Express shipping.



	shipping_type	(No column name)
1	Standard	58
2	Express	60

---

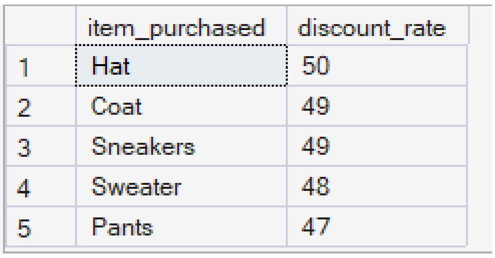
5. Subscribers vs. Non-Subscribers – Compared average spend and total revenue across subscription status



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

---

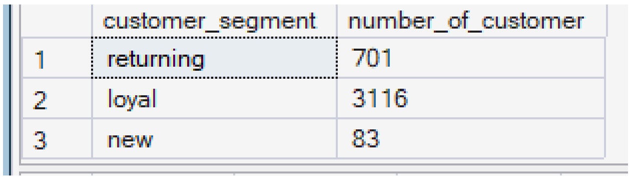
6. Discount-Dependent Products – Identified 5 products with the highest percentage of discounted purchases.



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

---

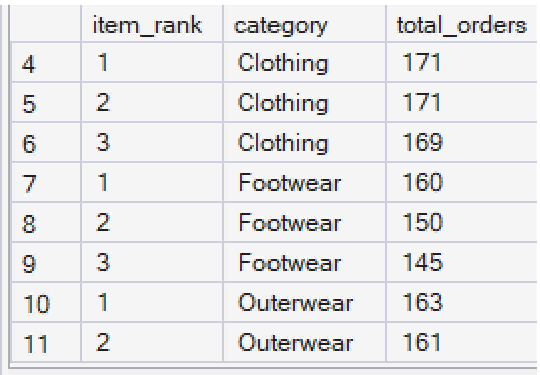
7. Customer Segmentation – Classified customers into New, Returning, and Loyal segments based on purchase history.



	customer_segment	number_of_customer
1	returning	701
2	loyal	3116
3	new	83

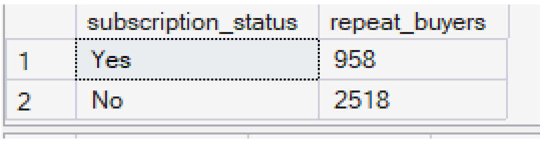
---

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



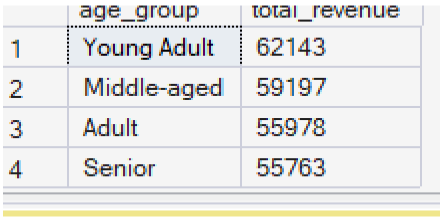
	item_rank	category	total_orders
4	1	Clothing	171
5	2	Clothing	171
6	3	Clothing	169
7	1	Footwear	160
8	2	Footwear	150
9	3	Footwear	145
10	1	Outerwear	163
11	2	Outerwear	161

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



	subscription_status	repeat_buyers
1	Yes	958
2	No	2518

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



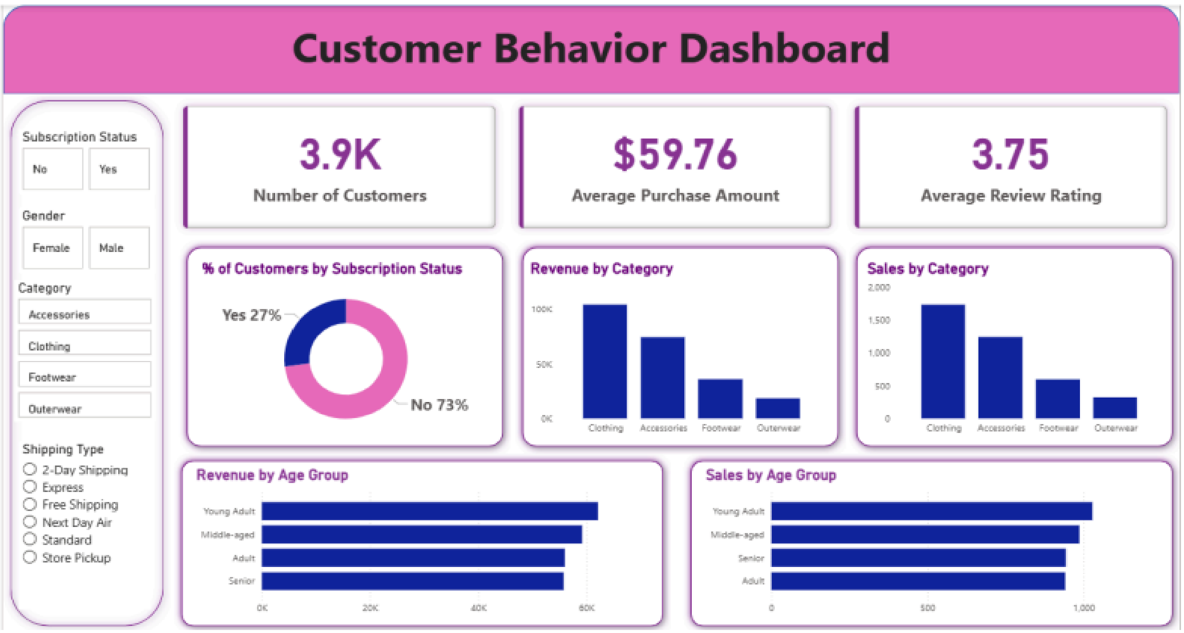
	age_group	total_revenue
1	Young Adult	62143
2	Middle-aged	59197
3	Adult	55978
4	Senior	55763

SQL Concepts Used

- Joins
- CTE (Common Table Expressions)
- Aggregate Functions
- Window Functions
- GROUP BY & HAVING
- Subqueries

5. Dashboard in Power BI

Finally, we built an interactive dashboard in Power BI to present insights visually.



---

6. Key Business Insights

- Identified high-value customers
- Discovered peak sales periods
- Analyzed category performance trends
- Observed customer purchasing patterns
- Evaluated discount impact on revenue

---

7. Business Recommendations

● Improve customer retention through loyalty programs

● Optimize discount strategy for higher profitability

● Focus marketing on high-performing categories

● Target high-spending customer segments

● Use seasonal trends for campaign planning