STUDENT DECLARATION

I hereby declare that the work being presented in this report entitled **"Customer Shopping Behaviour Analysis"** is an authentic record of my own work carried out under the supervision of **Prof. Neelam Yadav.**

Date:2 Dec,2025 **Signature Of Student**

Deepanshu Singh Chauhan

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of HOD

Prof.(Dr.) Devendra Kumar

HOD-MCA

Signature of Internal

Prof. Neelam Yadav

Program Leader (MCA)

ACKNOWLEDGEMENT

Any software project is not a work of an individual. It combines effort, ideas, suggestions, reviews, and hard work. I thank all who initiated and helped us to complete my project. One of the most pleasing aspects of collecting the necessary information and compiling it is the opportunity to thank those who have actively contributed to it.

I would like to thank **Prof. Neelam Yadav** guidance and cooperation render for following me to undergo training under his/her guidance; I am also thanks to all the other college staff who helped me despite their busy schedule.

My special thanks to **Prof. Neelam Yadav (Program Leader-MCA)** for his active support, affectionate, guidance and constant encouragement.
I am highly indebted to **Prof. (Dr.) Devendra Kumar (HOD-MCA)** for his continuous effort in building a good infrastructure and developing a professional attitude within ourselves during the academic period of MCA (Master of Computer Application).

Signature of Student

Deepanshu Singh Chauhan

Roll No-2400320140064

MCA-III Semester

ABSTRACT

Retail businesses generate large volumes of customer data, but most organizations struggle to convert this raw data into meaningful insights. This project, **Customer Shopping Behaviour Analysis**, aims to analyse the purchasing patterns of customers using a real dataset. The study covers customer demographics, preferences, spending patterns, frequency of purchase, and product category choices.

The analysis was performed using Python, Pandas, and Power BI for dashboard creation. The final outcome of this project is an interactive dashboard where decision-makers can visually explore insights such as top-selling product categories, high-value customers, gender-wise spending patterns, age-wise buying behaviour, seasonal trends, and correlations between income and spending.

This analysis helps businesses improve customer segmentation, plan targeted marketing campaigns, optimize product inventories, and enhance customer engagement strategies.

TABLE OF CONTENT

1. INTRODUCTION

Businesses collect massive amounts of customer data, but understanding patterns behind purchasing behaviour is essential to make informed decisions. This project studies transactional data to identify:

- What customers buy
- How much they spend
- Which groups generate the most revenue
- Whether discounts, reviews, or shipping choices affect spending
- Demographic trends

The dataset used in this project contains 3,900 customer purchase records with 18 columns covering demographic, behavioural, and transactional attributes. These include age, gender, location, subscription status, item purchased, product category, purchase amount, season, size, colour, review rating, shipping type, previous purchases, discount status, promo code usage, purchase frequency, payment method, and discount percentage. Together, these attributes provide a detailed view of customer shopping behaviour and enable a comprehensive analysis of spending trends, product preferences, loyalty patterns, and the impact of discounts and shipping choices on customer decisions.

1.1OBJECTIVES

- To analyse customer shopping behaviour using Python, SQL, and Power BI.
- To identify high-value customer segments based on spending and purchase frequency.
- To examine revenue distribution across demographics such as age, gender, and location.
- To study how discounts, review ratings, and shipping types influence purchase decisions.
- To understand the impact of subscription status on customer loyalty and total revenue.

- To visualize insights through an interactive Power BI dashboard for business decisionmakers.
- To uncover product and category trends that help improve inventory and marketing strategies.
- To provide clear, data-driven recommendations for improving customer engagement and retention.

1.2NEED OF PROJECT

Understanding customer shopping behaviour has become essential for businesses aiming to stay competitive in a data-driven market. This project helps organizations optimize their marketing budgets by identifying which customer groups and product categories contribute most to revenue. It also provides insights into customer loyalty, purchase frequency, and factors that influence repeat buying. By analysing spending patterns and preferences, the project highlights profitable items and categories that deserve strategic focus. Overall, this study enhances customer engagement, supports targeted decision-making, and enables companies to plan more effectively using clear, evidence-based insights.

To help businesses deeply understand customer buying patterns so they can improve marketing, product decisions, and overall sales performance.

2. FEASIBILITY STUDY

2.1 TECHNICAL FEASIBILITY

The technologies used in this project are modern, reliable, and fully compatible with industry analytics workflows. Python, along with libraries such as Pandas provides a powerful environment for data cleaning, transformation, and exploratory analysis. PostgreSQL enables structured data management and supports the execution of complex business queries. Power BI Desktop offers an intuitive interface to build interactive dashboards and visual reports. Jupyter Notebook ensures smooth execution of scripts and clear documentation. All tools are open-source or freely available, making them technically easy to deploy and maintain without specialized infrastructure.

2.2 OPERATIONAL FEASIBILITY

The complete analytics workflow from data preparation to dashboard visualization operates smoothly with minimal user intervention. The Power BI dashboard is designed to be intuitive, allowing managers and stakeholders to explore insights without technical expertise. Visualizations are clear and actionable, enabling teams to make informed decisions quickly. The system requires minimal training, integrates well with existing business processes, and can be used by non-technical personnel without difficulty.

2.3 ECONOMIC FEASIBILITY

This project is highly economical, as it relies primarily on open-source and freely available tools. There is no need for additional licensing, specialized hardware, or costly enterprise software. The entire system can be implemented on a standard computer, making it accessible even for small and medium-sized organizations. Overall, the project delivers significant analytical value at virtually zero monetary cost.

3. REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

- Processor: Intel Core i3 or higher
- RAM: Minimum 4 GB (8 GB recommended for smoother analysis)
- Storage: At least 10 GB free disk space
- Operating System: Windows 10/11 or equivalent

3.2 SOFTWARE REQUIREMENTS

- **Python 3.x** with required libraries (Pandas)
- **Jupyter Notebook** for interactive coding
- **PostgreSQL** for SQL-based analysis
- **Power BI Desktop** for dashboard creation
- **CSV Dataset** (Customer Shopping Behaviour Data)

3.3 FUNCTIONAL REQUIREMENTS

- **Data Loading and Cleaning-**The system should be able to load the dataset and clean it by removing errors, filling missing values, and preparing it for analysis.
- **Running Python Analysis-**Python scripts should run smoothly to perform data analysis, create charts, and generate important insights.
- **Executing SQL Queries-**The system should support SQL queries to answer business questions like revenue trends, customer segments, and product performance.
- **Creating Dashboards-**Power BI should be able to build interactive dashboards that show customer behaviour, sales patterns, and key statistics.
- **Data Filtering and Comparison-**Users should be able to filter and compare data based on age, gender, category, subscription status, and other fields.

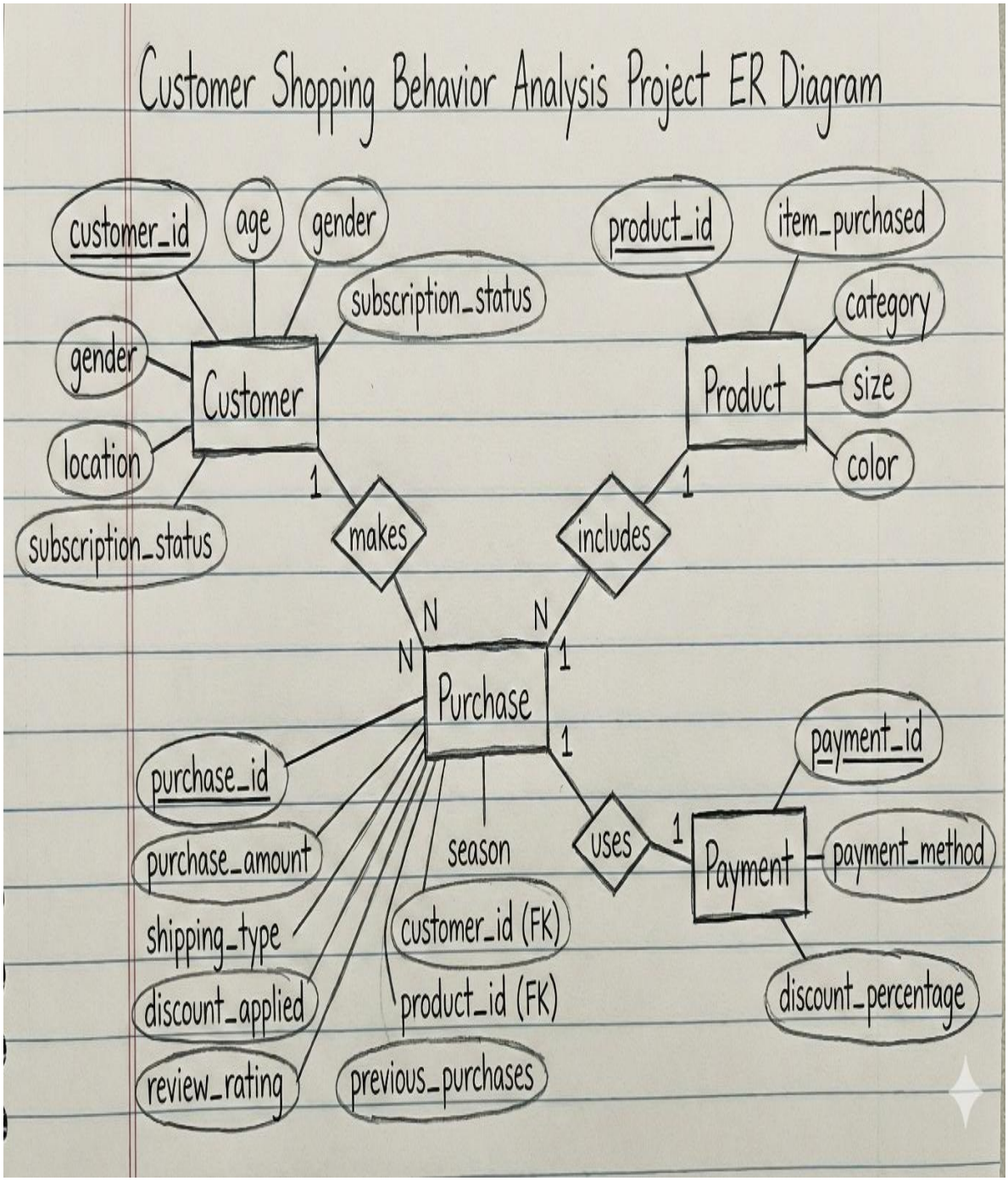
- **Exporting Reports-**The system should allow exporting charts, insights, and dashboards as reports for presentations or business use.

3.4 NON-FUNCTIONAL REQUIREMENTS

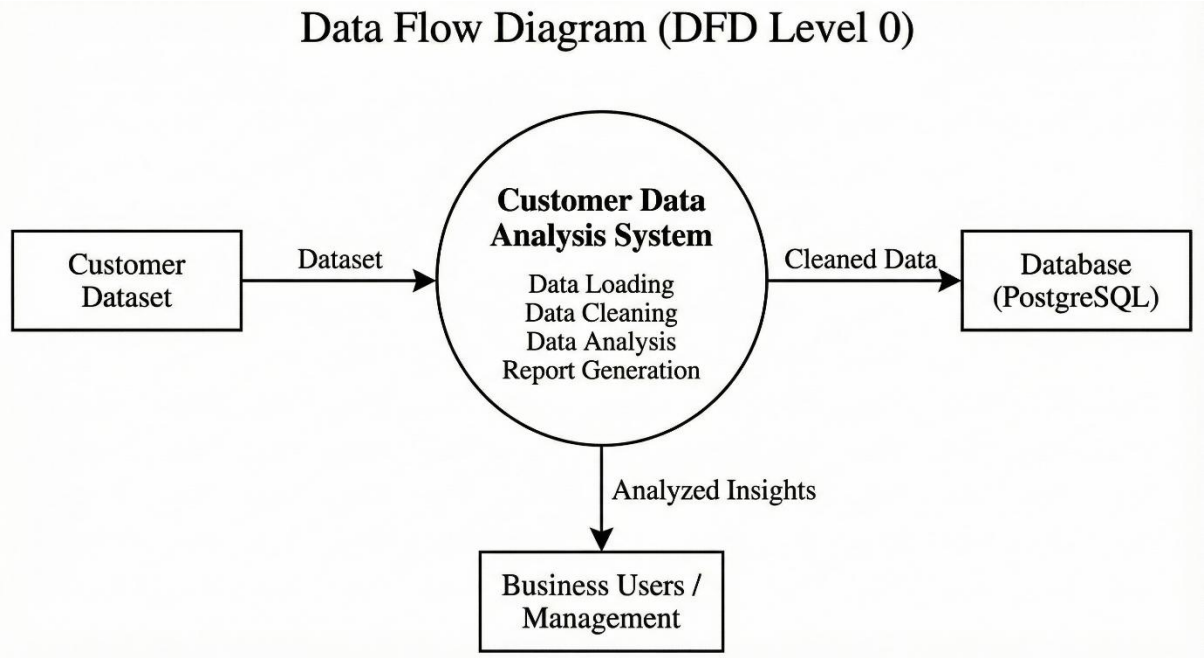
- **Usability-**Dashboards and reports should be simple to understand and easy to navigate.
- **Performance-**All analysis and dashboards should load quickly and run smoothly.
- **Accuracy-**Results must be correct and based on clean, reliable data.
- **Scalability-**The system should handle larger datasets or more features in the future.
- **Maintainability-**Code and dashboards should be easy to update, modify, or improve when needed.

4.Design

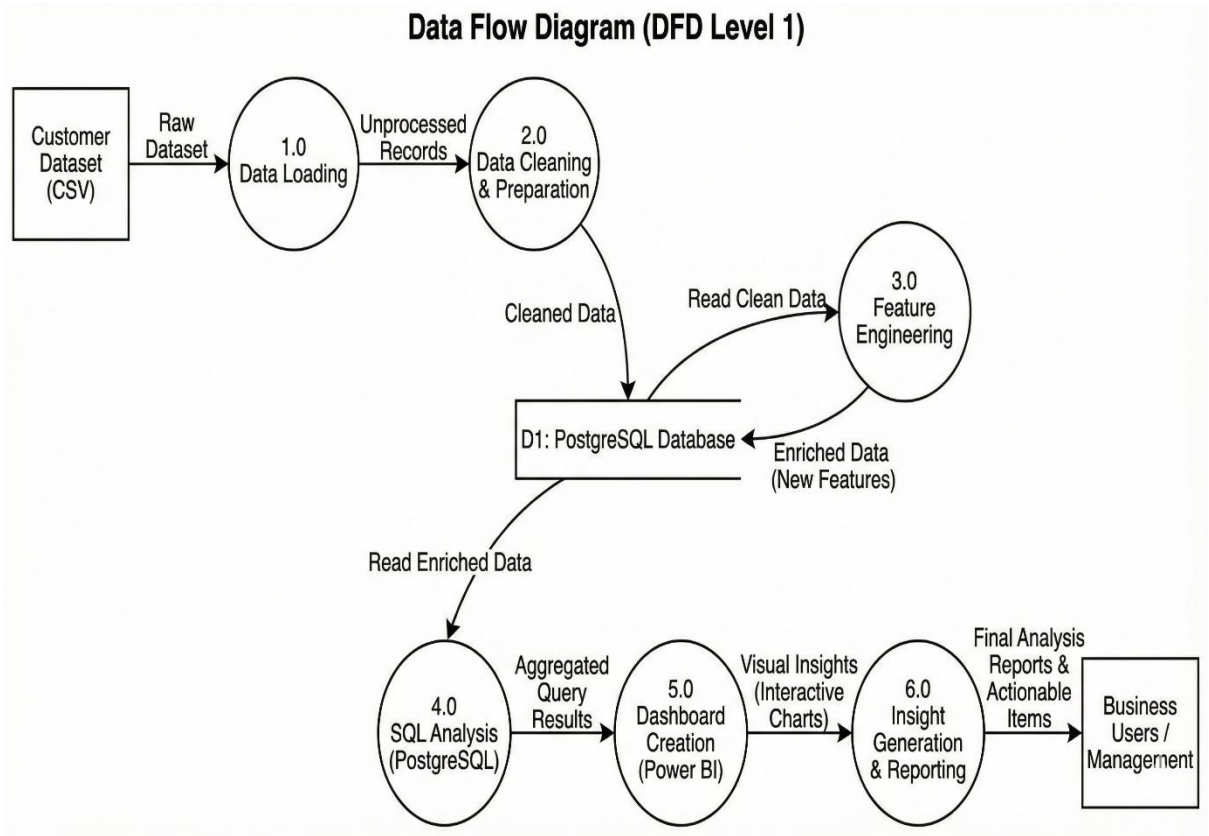
4.1 ER Diagram



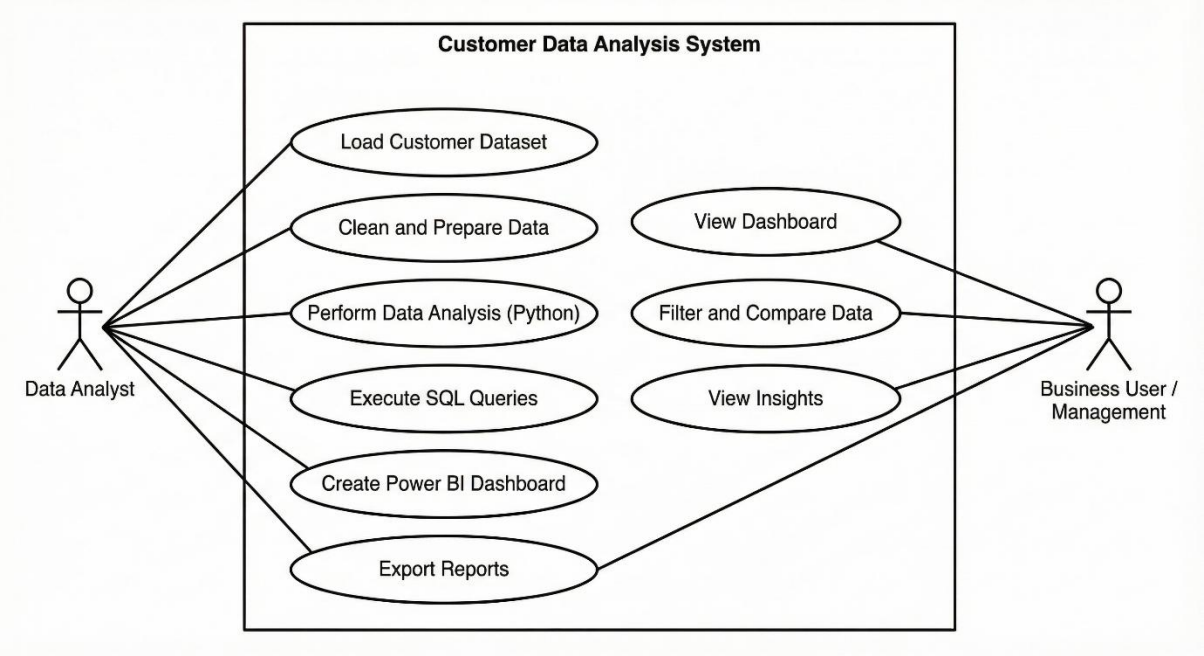
4.2 DFD LEVEL 0



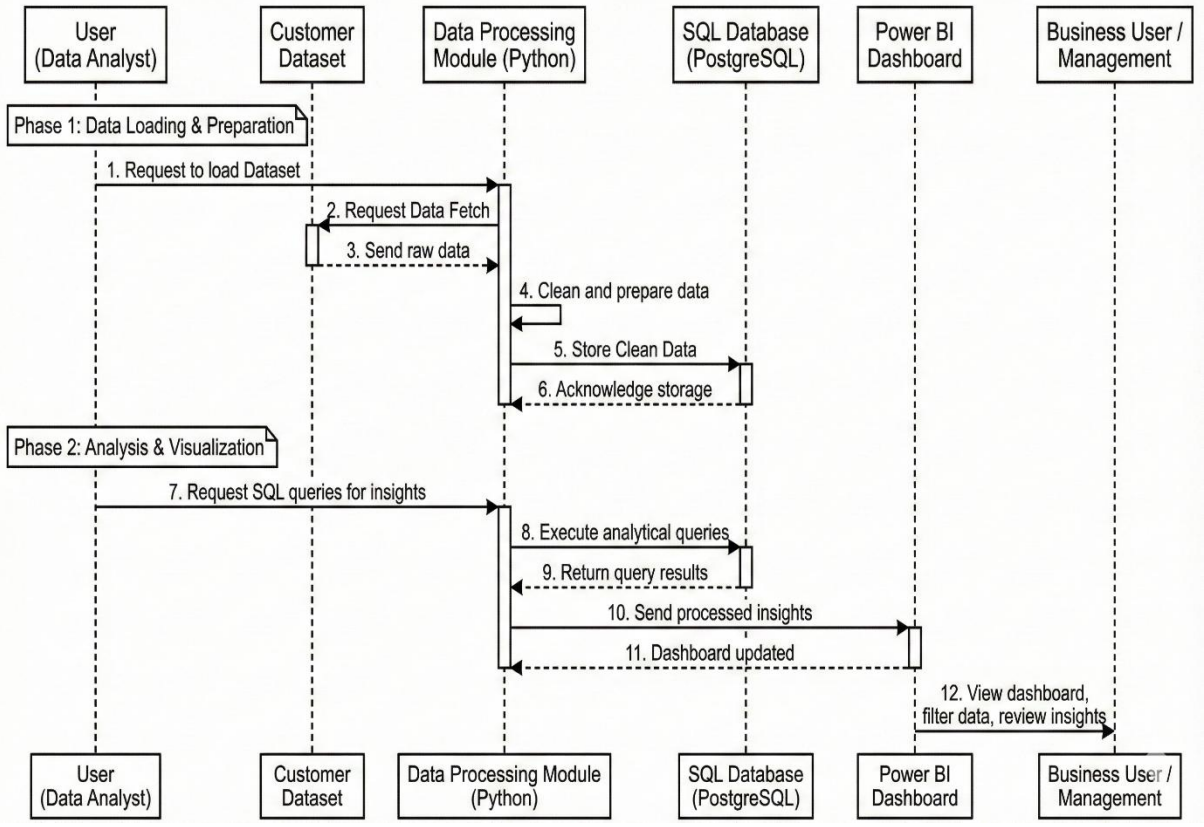
4.3 DFD LEVEL 1



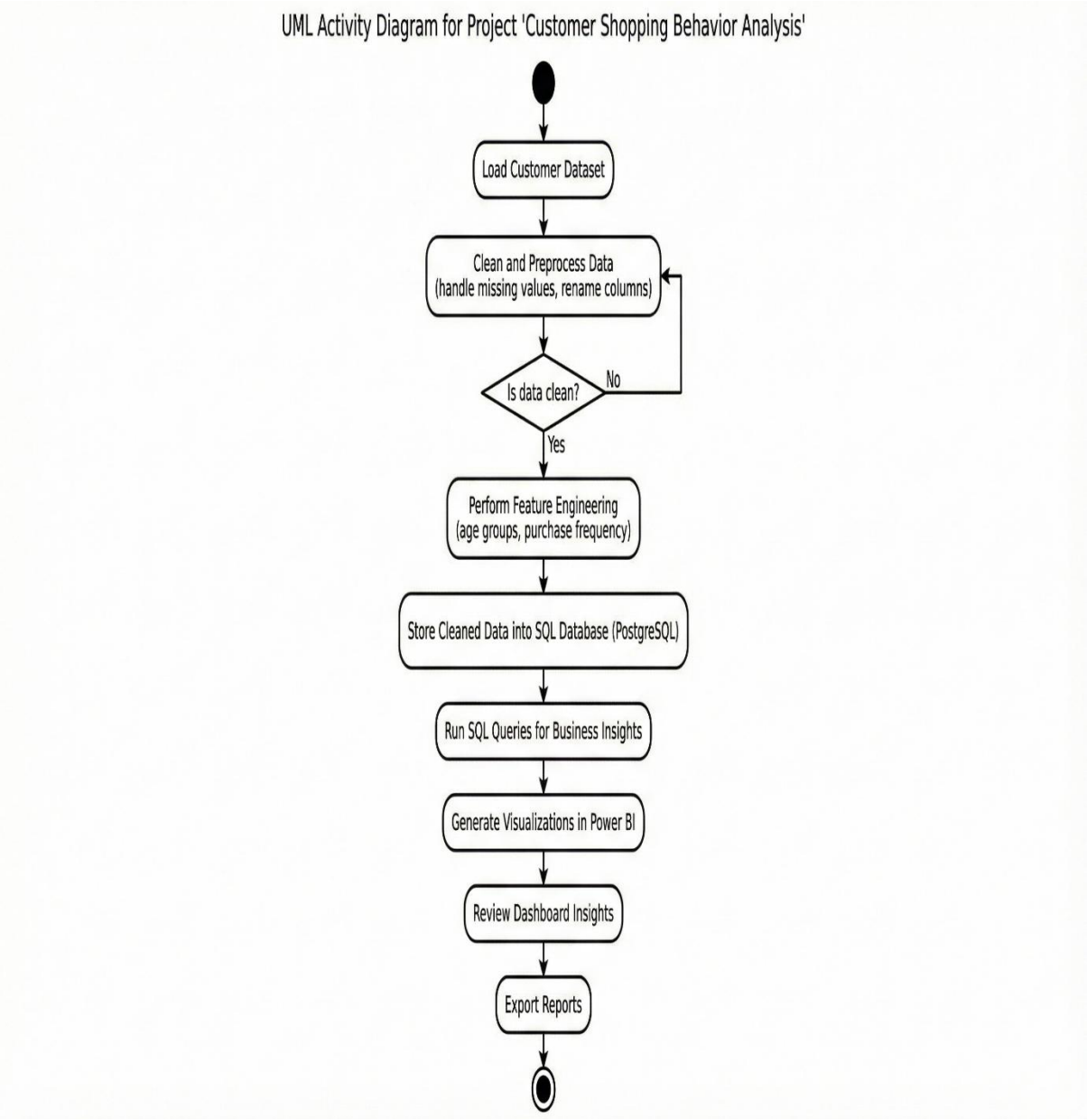
4.4 Use Case Diagram



4.5 Sequence Diagram



4.6Activity Diagram Prompt



5. GUI

5.1 POWER BI DASHBOARD EXPLAINATION WITH SCREENSHOTS

The Customer Behaviour Dashboard provides an interactive and visually clear interface that allows users to explore customer purchase patterns and business insights. The layout is divided into slicers, KPIs, and graphical charts to help management understand customer behaviour quickly and efficiently.

On the left side, the dashboard includes **interactive slicers** for Subscription Status, Gender, Category, and Shipping Type. These allow users to filter the data and immediately view how different customer groups behave. This makes the dashboard highly dynamic and useful for comparative analysis.

At the top, three main **Key Performance Indicators (KPIs)** summarize the most important metrics:

- **Number of Customers**
- **Average Purchase Amount**
- **Average Review Rating**

These KPIs provide a quick overview of customer engagement, spending levels, and satisfaction.

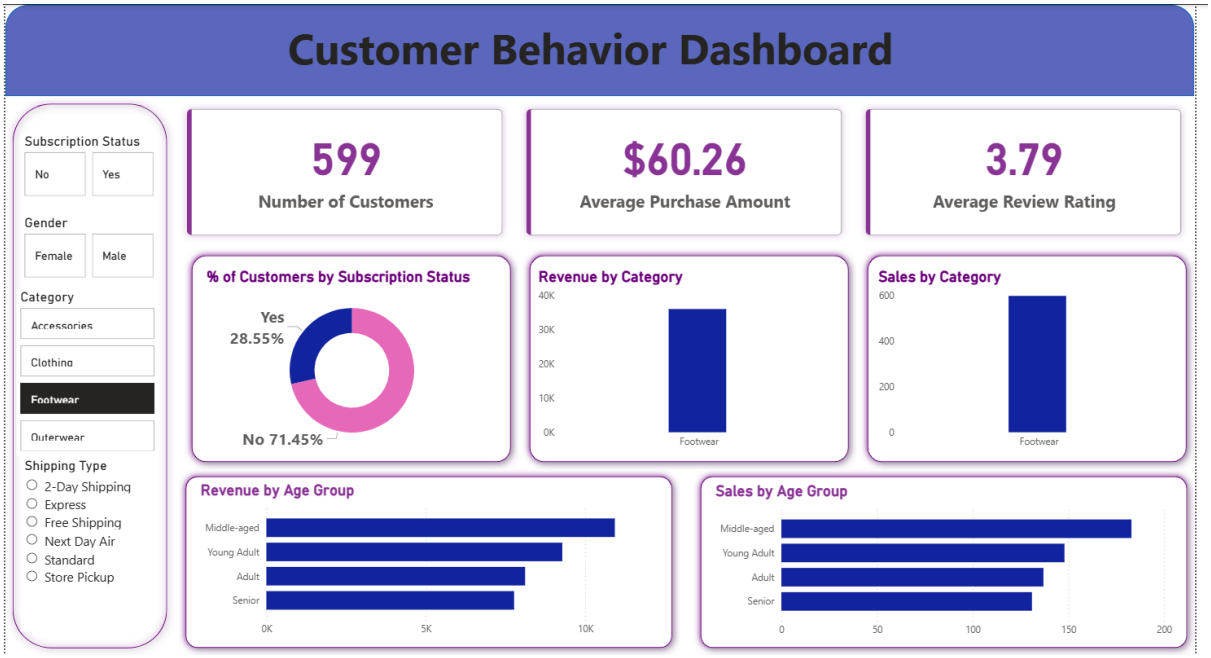
The centre section visualizes the **percentage of customers by subscription status**, helping identify the proportion of paying or loyal customers. The **Revenue by Category** and **Sales by Category** charts show which product categories are performing well, allowing businesses to focus on profitable segments.

The bottom section includes **Revenue by Age Group** and **Sales by Age Group** charts. These visuals help identify which age groups contribute the most to revenue and which groups have the highest purchasing frequency.

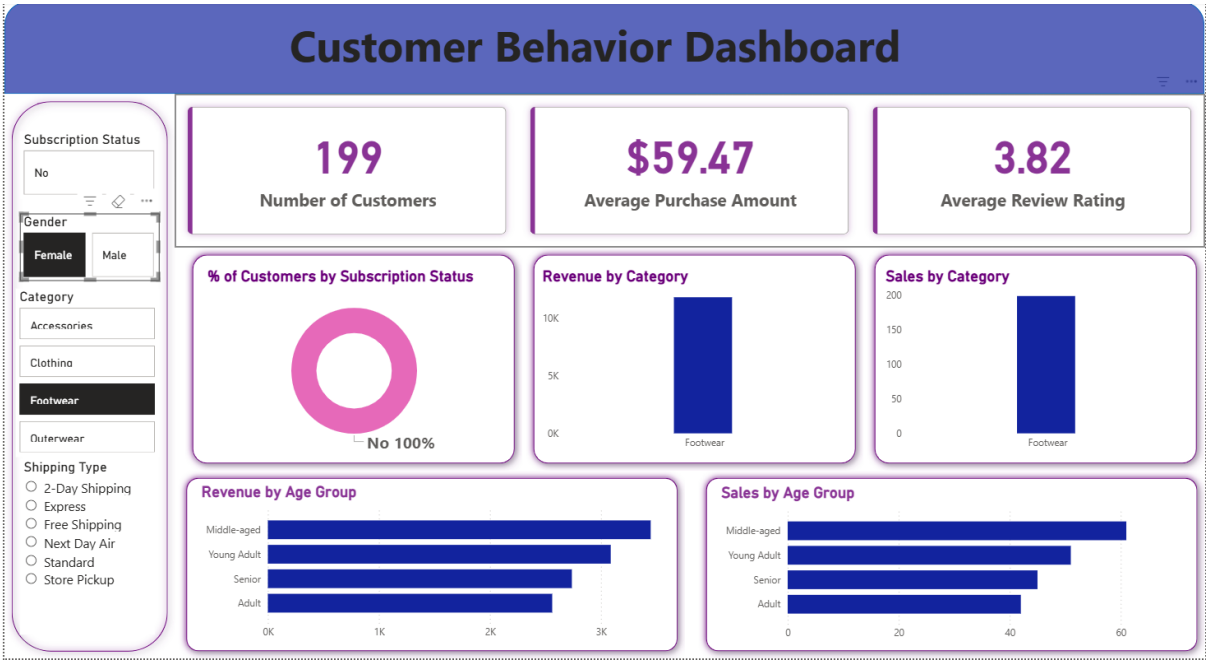
Overall, the dashboard is intuitive, interactive, and provides a complete view of customer behaviour through well-structured charts and KPIs. It allows decision-makers to filter data, compare customer segments, and quickly identify trends that support data-driven business strategies.

SCREENSHOTS

ALL



ONLY FEMALE



5.2 DATASET TABLE STRUCTURE

Columns include:

- age, gender, location
- item_purchased, category
- purchase_amount, season
- shipping_type
- discount_applied
- subscription_status
- review_rating
- previous_purchases etc

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Customer I	Age	Gender	Item Purcha	Category	Purchase Ar	Location	Size	Color	Season	Review Rati	Subscriptior	Shipping Ty	Discount Ap	Promo Cod	Previous Pu	Payment M	Frequency of Purcha	
2	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo	Fortnightly	
3	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	Cash	Fortnightly	
4	3	50	Male	Jeans	Clothing	73	Massachuse	S	Maroon	Spring	3.1	Yes	Free Shippir	Yes	Yes	23	Credit Card	Weekly	
5	4	21	Male	Sandals	Footwear	90	Rhode Islan	M	Maroon	Spring	3.5	Yes	Next Day Ai	Yes	Yes	49	PayPal	Weekly	
6	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shippir	Yes	Yes	31	PayPal	Annually	
7	6	46	Male	Sneakers	Footwear	20	Wyoming	M	White	Summer	2.9	Yes	Standard	Yes	Yes	14	Venmo	Weekly	
8	7	63	Male	Shirt	Clothing	85	Montana	M	Gray	Fall	3.2	Yes	Free Shippir	Yes	Yes	49	Cash	Quarterly	
9	8	27	Male	Shorts	Clothing	34	Louisiana	L	Charcoal	Winter	3.2	Yes	Free Shippir	Yes	Yes	19	Credit Card	Weekly	
10	9	26	Male	Coat	Outerwear	97	West Virgini	L	Silver	Summer	2.6	Yes	Express	Yes	Yes	8	Venmo	Annually	
11	10	57	Male	Handbag	Accessories	31	Missouri	M	Pink	Spring	4.8	Yes	2-Day Shipp	Yes	Yes	4	Cash	Quarterly	
12	11	53	Male	Shoes	Footwear	34	Arkansas	L	Purple	Fall	4.1	Yes	Store Picku	Yes	Yes	26	Bank Transf	Bi-Weekly	
13	12	30	Male	Shorts	Clothing	68	Hawaii	S	Olive	Winter	4.9	Yes	Store Picku	Yes	Yes	10	Bank Transf	Fortnightly	
14	13	61	Male	Coat	Outerwear	72	Delaware	M	Gold	Winter	4.5	Yes	Express	Yes	Yes	37	Venmo	Fortnightly	
15	14	65	Male	Dress	Clothing	51	New Hamps	M	Violet	Spring	4.7	Yes	Express	Yes	Yes	31	PayPal	Weekly	
16	15	64	Male	Coat	Outerwear	53	New York	L	Teal	Winter	4.7	Yes	Free Shippir	Yes	Yes	34	Debit Card	Weekly	
17	16	64	Male	Skirt	Clothing	81	Rhode Islan	M	Teal	Winter	2.8	Yes	Store Picku	Yes	Yes	8	PayPal	Monthly	
18	17	25	Male	Sunglasses	Accessories	36	Alabama	S	Gray	Spring	4.1	Yes	Next Day Ai	Yes	Yes	44	Debit Card	Bi-Weekly	
19	18	53	Male	Dress	Clothing	38	Mississippi	XL	Lavender	Winter	4.7	Yes	2-Day Shipp	Yes	Yes	36	Venmo	Quarterly	
20	19	52	Male	Sweater	Clothing	48	Montana	S	Black	Summer	4.6	Yes	Free Shippir	Yes	Yes	17	Cash	Weekly	
21	20	66	Male	Pants	Clothing	90	Rhode Islan	M	Green	Summer	3.3	Yes	Standard	Yes	Yes	46	Debit Card	Bi-Weekly	
22	21	21	Male	Pants	Clothing	51	Louisiana	M	Black	Winter	2.8	Yes	Express	Yes	Yes	50	Cash	Every 3 Months	
23	22	31	Male	Pants	Clothing	62	North Carol	M	Charcoal	Winter	4.1	Yes	Store Picku	Yes	Yes	22	Debit Card	Quarterly	
24	23	56	Male	Pants	Clothing	37	California	M	Peach	Summer	3.2	Yes	Store Picku	Yes	Yes	32	Debit Card	Annually	
25	24	31	Male	Pants	Clothing	88	Oklahoma	XL	White	Winter	4.4	Yes	Express	Yes	Yes	40	Credit Card	Weekly	
26	25	18	Male	Jacket	Outerwear	22	Florida	M	Green	Fall	2.9	Yes	Store Picku	Yes	Yes	16	Debit Card	Weekly	
27	26	18	Male	Hoodie	Clothing	25	Texas	M	Silver	Summer	3.6	Yes	Express	Yes	Yes	14	PayPal	Annually	
28	27	38	Male	Jewelry	Accessories	20	Nevada	M	Red	Spring	3.6	Yes	Next Day Ai	Yes	Yes	13	Credit Card	Annually	
29	28	56	Male	Shorts	Clothing	56	Kentucky	L	Cyan	Summer	5	Yes	Next Day Ai	Yes	Yes	7	Bank Transf	Every 3 Months	

6. CODING

6.1 Python scripts for cleaning, missing value handling

[6]:
```
import pandas as pd

df = pd.read_csv(r"C:\Users\satya\Downloads\customer_shopping_behavior.csv")
df.head()
```

[6]:

	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	I
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	I

[2]:
```
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
```

[3]:
```
df.describe()
```

[3]:

	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

[4]:
```
df.describe(include='all')
```

[4]:

	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

[5]:
```
# Checking if missing data or null values are present in the dataset

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

[5]:
```
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
```

[9]:
```
# Renaming columns according to snake casing for better readability and documentation

df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ','_')
df = df.rename(columns={'purchase_amount_(usd)':'purchase_amount'})
```

[10]:
```
df.columns
```

[10]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'promo_code_used', 'previous_purchases',
       'payment_method', 'frequency_of_purchases'],
      dtype='object')
```

[11]:
```
# create a new column age_group
labels = ['Young Adult', 'Adult', 'Middle-aged', 'Senior']
df['age_group'] = pd.qcut(df['age'], q=4, labels = labels)
```

[12]:
```
df[['age','age_group']].head(10)
```

[12]:

	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

[13]:
```python
# create 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)
```

[14]:
```python

df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

[14]:

	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

[15]:
```python
df[['discount_applied','promo_code_used']].head(10)
```

[16]:
```python
(df['discount_applied'] == df['promo_code_used']).all()
```

[16]:
```
np.True_
```

[17]:
```python
# Dropping promo code used column

df = df.drop('promo_code_used', axis=1)
```

[18]:
```python

df.columns
```

[18]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'previous_purchases', 'payment_method',
       'frequency_of_purchases', 'age_group', 'purchase_frequency_days'],
      dtype='object')
```

[19]:
```
!pip install psycopg2-binary sqlalchemy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: psycopg2-binary in c:\users\satya\appdata\roaming\python\python313\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\satya\appdata\roaming\python\python313\site-packages (2.0.44)
Requirement already satisfied: greenlet>=1 in c:\users\satya\appdata\roaming\python\python313\site-packages (from sqlalchemy) (3.2.4)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\satya\appdata\roaming\python\python313\site-packages (from sqlalchemy) (4.15.0)
```

[20]:
```python
from sqlalchemy import create_engine

username = "postgres"        # pgAdmin me login wala username
password = "1234"         # tumhara PostgreSQL password
host = "localhost"
port = "5432"
database = "customer_behavior"

engine = create_engine(f"postgresql+psycopg2://{username}:{password}@{host}:{port}/{database}")
```

[21]:
```python
table_name = "customer"

df.to_sql(table_name, engine, if_exists="replace", index=False)

print(f"Data successfully loaded into table '{table_name}' in database '{database}'.")
```

```
Data successfully loaded into table 'customer' in database 'customer_behavior'.
```

6.2 SQL queries

customer_behavior/postgres@PostgreSQL 17

No limit

Query | Query History

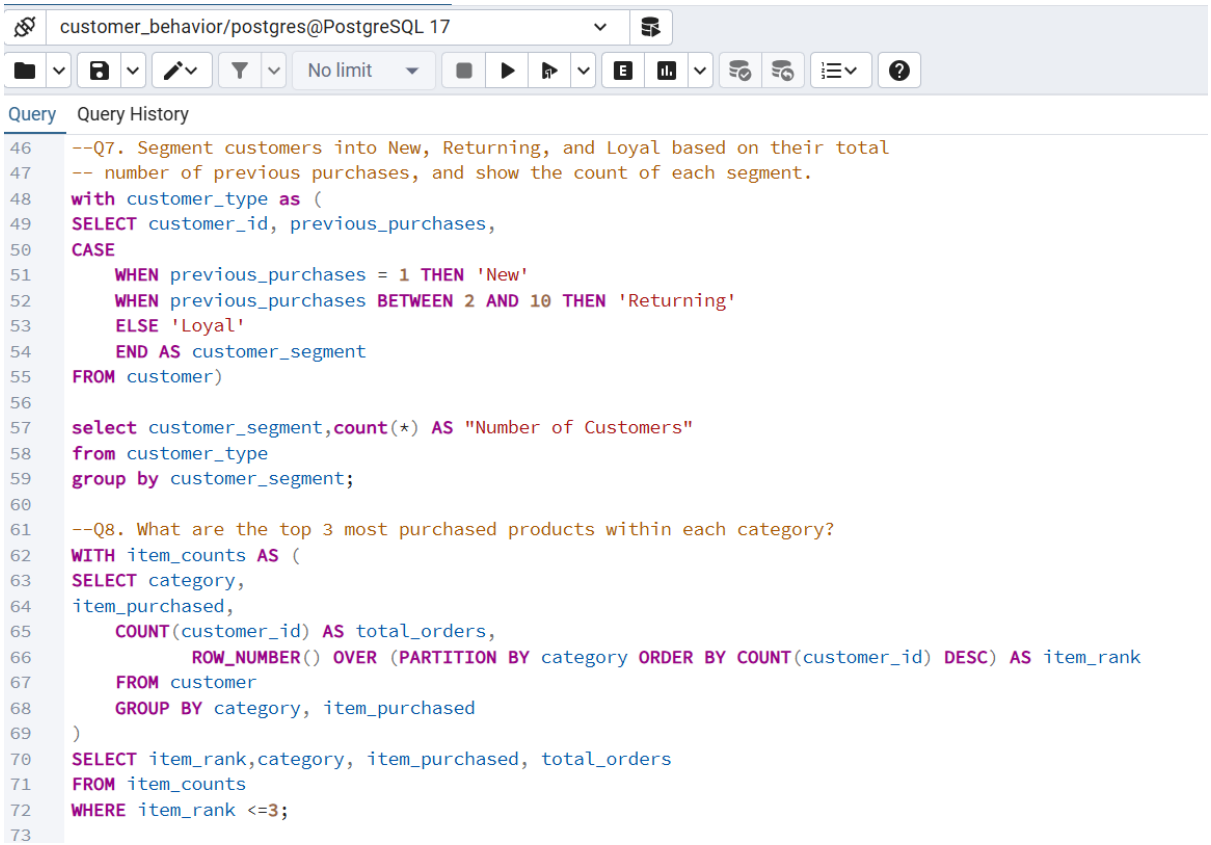
```sql
--Q1. What is the total revenue generated by male vs. female customers?
select gender, SUM(purchase_amount) as revenue
from customer
group by gender


--Q2. Which customers used a discount but still spent more than the average purchase amount?
select customer_id, purchase_amount
from customer
where discount_applied = 'Yes' and purchase_amount >= (select AVG(purchase_amount) from customer)


-- Q3. Which are the top 5 products with the highest average review rating?
select item_purchased, round(avg(review_rating::numeric),2) as "Average Product Rating"
from customer
group by item_purchased
order by avg(review_rating) desc
limit 5

--Q4. Compare the average Purchase Amounts 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;
```

customer_behavior/postgres@PostgreSQL 17

No limit

Query | Query History

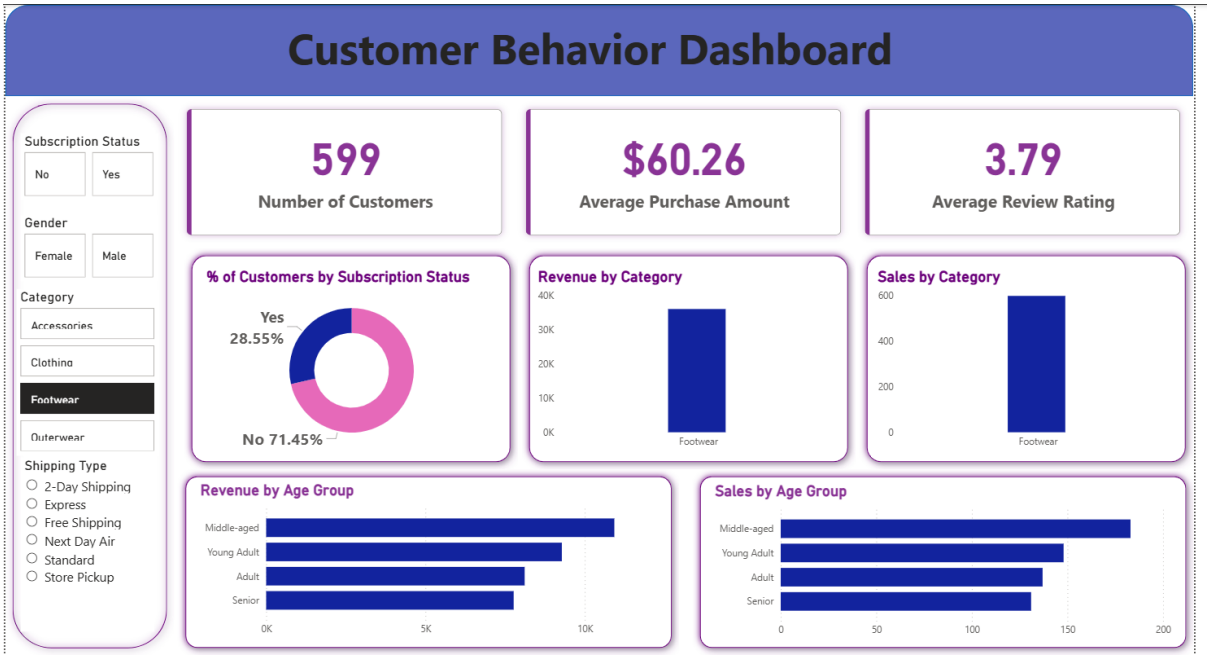
```sql
--Q5. Do subscribed customers spend more? Compare average spend and total revenue
--between subscribers and non-subscribers.
SELECT subscription_status,
       COUNT(customer_id) AS total_customers,
       ROUND(AVG(purchase_amount),2) AS avg_spend,
       ROUND(SUM(purchase_amount),2) AS total_revenue
FROM customer
GROUP BY subscription_status
ORDER BY total_revenue,avg_spend DESC;

--Q6. Which 5 products have the highest percentage of purchases with discounts applied?
SELECT item_purchased,
       ROUND(100.0 * SUM(CASE WHEN discount_applied = 'Yes' THEN 1 ELSE 0 END)/COUNT(*),2) AS discount_rate
FROM customer
GROUP BY item_purchased
ORDER BY discount_rate DESC
LIMIT 5;
```




```sql
--Q7. Segment customers into New, Returning, and Loyal based on their total
-- number of previous purchases, and show the count of each segment.
with customer_type as (
SELECT customer_id, previous_purchases,
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;

--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;

```

```sql
--Q9. Are customers who are repeat buyers (more than 5 previous purchases) also likely to subscribe?
SELECT subscription_status,
COUNT(customer_id) AS repeat_buyers
FROM customer
WHERE previous_purchases > 5
GROUP BY subscription_status;

--Q10. What is the revenue contribution of each age group?
SELECT age_group,
SUM(purchase_amount) AS total_revenue
FROM customer
GROUP BY age_group
ORDER BY total_revenue desc;
```

6.3 PowerBI Dashboard



7. FUTURE SCOPE

1. Machine Learning–Based Customer Segmentation

The project can be extended by applying ML clustering algorithms (such as K-Means) to group customers based on age, spending habits, subscription status, and purchase frequency. This will help identify loyal, returning, and high-value customers more accurately.

2. Customer Churn Prediction

Predictive models can be developed to identify customers who are likely to stop purchasing or unsubscribe. This helps businesses take timely actions to retain valuable customers.

3. Automated Dashboard Refresh

Power BI can be connected to auto-refresh pipelines so the dashboard updates automatically whenever new purchase data arrives. This ensures real-time decision-making.

4. Real-Time Data Streaming Integration

The system can be enhanced by integrating real-time data sources, allowing organizations to monitor customer activity, sales trends, and revenue changes instantly.

5. Product Recommendation System

A recommendation engine can be added to suggest relevant products to customers based on their past purchases, category preferences, and top-rated items from the dataset.

6. Advanced BI Features (Drill-through, What-If Analysis)

Future dashboards can include drill-through reports, scenario analysis, and AI-generated insights to provide deeper exploration and better business understanding.

7. Enhanced SQL Analytics

More complex SQL queries can be designed to analyze cross-category behavior, seasonal patterns, and multi-purchase trends for richer business insights.

8. Integration with Marketing Automation

Insights from the analysis can be pushed into marketing tools to automate personalized offers, subscription reminders, and discount campaigns for targeted customer groups.

8. CONCLUSION

This project successfully examined customer shopping behaviour using Python, SQL, and Power BI. The analysis helped convert raw data into meaningful insights that can support better business decisions. The key conclusions are:

• **Better Customer Understanding-**The project identifies how different customers behave based on age, gender, category preference, subscription status, and shipping type.

• **Improved Customer Targeting-**Businesses can focus on profitable customer groups and design marketing campaigns based on actual spending patterns.

• **Stronger Customer Retention-**The analysis shows how previous purchases, review ratings, and subscriptions affect customer loyalty, helping companies create better retention strategies.

• **Clear Product Insights-**Top-performing categories and high-rated products can be identified easily, helping businesses plan inventory and product focus.

• **Data-Driven Decisions-**The interactive Power BI dashboard makes it easy for managers to explore data, compare groups, and understand trends without technical skills.

• **Answers the Business Problem-**The project directly supports the business goal by using data to identify trends, understand customer needs, and improve overall marketing and product strategy.

9.REFRENCES

1. **Python Software Foundation, *Python Documentation*, 2024. Available: https://www.python.org/doc/**
2. ***Pandas Documentation*, pandas.pydata.org, 2024. Available: https://pandas.pydata.org/docs/**
3. **PostgreSQL Global Development Group, *PostgreSQL 17 Documentation*, 2024. Available: https://www.postgresql.org/docs/**
4. Microsoft Corporation, *Power BI Desktop User Guide*, 2024. Available**: https://learn.microsoft.com/power-bi/**
5. Customer Shopping Behaviour Dataset
6. Business Problem Statement of Customer Shopping Behaviour