Customer_shopping_behavior_analysis.

1. Project Overview

This project analyses customer shopping behaviour 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 behaviour to guide strategic business decisions.

---

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, Colour
- Shopping behaviour: Discount Applied, Promo Code Used, Previous Purchases, Frequency of Purchases, Review Rating, Shipping 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.

- First, import the pandas library to work with data. And first, import the pandas library to work with data.

```
import pandas as pd
df=pd.read_csv('E:/Desktop/DA DS 2025/Customer_Trends_Project/customer_shopping_behavior.csv')
```

- Verify data loaded successfully and check using head() function it's display first 5 rows.

```
df.head()
```

Output:-

	Customer ID	Age	Gender	Color	Season
0	1	55	Male	Gray	Winter
1	2	19	Male	Maroon	Winter
2	3	50	Male	Maroon	Spring

- Check dataset information to understand the structure of the dataset, we use:

```
df.info()
```

Output:-

```
<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
```

----------More_Records----------

```
 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)
```

- To understand the statistical distribution of the dataset, we use:

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

Output:-

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900
unique	NaN	NaN	2	25	4	NaN	50
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana
freq	NaN	NaN	2652	171	1737	NaN	96

- To identify missing values in the dataset, we use:

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

Output:-

```
Customer ID                0
Age                        0
Gender                     0
----------More_Records----------
Review Rating             37
Subscription Status        0
Payment Method             0
Frequency of Purchases     0
dtype: int64
```

- To fill missing values in the ***Review Rating*** column based on category-wise median, we use:

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

Output:-

```
Customer ID                0
Age                        0
Gender                     0
----------More_Records------------
Review Rating              0
Subscription Status        0
Payment Method             0
Frequency of Purchases     0
dtype: int64
```


- To make column names clean and SQL-friendly, we apply:

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

- To display all column names in the dataset, we use:

```
df.columns
```

Output:-

```
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')
```

- To segment customers into age groups, we use and confirm that the age segmentation was created correctly, display the age and age_group columns.

```
labels=['young_adult','adult','middle_aged','senior']
df['age_group']=pd.qcut(df['age'],q=4,labels=labels)
df[['age','age_group']].head(10)
```

Output:-

age	age_group
55	middle_aged
19	young_adult
----------*MORE*----------	
26	young_adult
57	middle_aged

- To convert categorical purchase frequency into numeric days, we use mapping.

```
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)
```

Output:-

purchase_frequency_days	frequency_of_purchases
14	Fortnightly
14	Fortnightly
----------*MORE*----------	
365	Annually
90	Quarterly

- To examine whether the discount_applied and promo_code_used columns contain similar information, display the first 10 records.

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

Output:-

discount_applied	promo_code_used
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes

- To check whether discount_applied and promo_code_used contain the same values, use:

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

Output:-

```
np.True_
```

- After confirming that promo_code_used contains the same information as discount_applied, we remove the redundant column.

```
df.drop("promo_code_used", axis=1, inplace=True)
```

- After completing data cleaning and feature engineering, export the final dataset as a CSV file.

```
df.to_csv("E:/Desktop/customer_behavior.csv", index=False)
```

- To change the working directory to Desktop before saving files, use:

```
import os
os.chdir("E:/Desktop")
os.getcwd()
```

```
'E:\\Desktop'
```

- To connect Python with PostgreSQL, install the required packages:

```
%pip install psycopg2-binary sqlalchemy
```

- After installing the required packages, verify that they are installed correctly by importing them.

```
import psycopg2
import sqlalchemy
print("Installed successfully")
```

```
Installed successfully
```

- This step connects Python to PostgreSQL and uploads the cleaned CSV file into a database table.

```
#for pgadmin
import pandas as pd
from sqlalchemy import create_engine
# Load data first
df = pd.read_csv("E:/Desktop/customer_behavior.csv")
username = "postgres"
password = "9027"
host = "localhost"
port = "5432"
database = "customer_behavior"
engine = create_engine(
    f"postgresql+psycopg2://{username}:{password}@{host}:{port}/{database}"
)
table_name = "customer"
df.to_sql(table_name, engine, if_exists="replace", index=False)
print("Data uploaded successfully!")
```

4.Data Analysis using SQL (Business Transactions)

We performed structured analysis in ***PostgreSQL & MySQL*** to answer key business ques.

- **Revenue Analysis**

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;
```

→ Output:-

	gender text	total_revenue numeric
1	Female	75191
2	Male	157890

Q2.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 age_group DESC;
```

→ Output:-

	age_group text	total_revenue numeric
1	young_adult	62143
2	senior	55763
3	middle_aged	59197
4	adult	55978

- **Discount & Pricing Insights**

Q3.Which top 20 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)
      ORDER BY purchase_amount DESC
      LIMIT 20;
```

→ Output:-

	customer_id bigint	purchase_amount bigint
1	1301	100
2	1480	100
3	862	100
----------More----------		
18	43	100
19	456	100
20	1592	100

Q4.Which 5 products have the highest percentage of purchases with discounts applied?

```
SELECT * FROM customer;
SELECT item_purchased,
       COUNT(*) AS total_purchases,
       ROUND(SUM(CASE WHEN discount_applied = 'Yes' THEN 1 ELSE 0 END) * 100.0
       / COUNT(*),2) AS discount_percentage
FROM customer
GROUP BY item_purchased
ORDER BY discount_percentage DESC
LIMIT 5;
```

→ Output:-

	item_purchased text	total_purchases bigint	discount_percentage numeric
1	Hat	154	50.00
2	Sneakers	145	49.66
3	Coat	161	49.07
4	Sweater	164	48.17
5	Pants	171	47.37

- **Product Performance**

Q5.Which are the top 5 products with the highest average review rating?

```
SELECT item_purchased,
       AVG(review_rating) AS avg_rating
FROM customer
GROUP BY item_purchased
ORDER BY avg_rating DESC
LIMIT 5;
```

→ Output:-

	item_purchased text	avg_rating double precision
1	Gloves	3.8614285714285725
2	Sandals	3.8443750000000003
3	Boots	3.8187500000000005
4	Hat	3.8012987012987005
5	Skirt	3.784810126582278

Q6.What are the top 3 most purchased products within each category?

```
SELECT category, item_purchased, purchase_count
FROM (
    SELECT category,
           item_purchased,
           COUNT(*) AS purchase_count,
           ROW_NUMBER() OVER (PARTITION BY category
                              ORDER BY COUNT(*) DESC) AS rank_num
    FROM customer
    GROUP BY category, item_purchased
) ranked
WHERE rank_num <= 3;
```

→ Output:-

	category text	item_purchased text	purchase_count bigint
1	Accessories	Jewelry	171
2	Accessories	Sunglasses	161
	----------More----------		
9	Footwear	Sneakers	145
10	Outerwear	Jacket	163
11	Outerwear	Coat	161

- **Shipping & Purchase Behavior**

Q7.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;
```

→ Output:-

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

- **Subscription & Customer Value**

Q8.Do subscribed customers spend more? Compare average spend and total revenue between subscribers and non-subscribers.

```
SELECT subscription_status,
      ROUND(AVG(purchase_amount),2) AS avg_spend,
      SUM(purchase_amount) AS total_revenue
FROM customer
GROUP BY subscription_status;
```

→ Output:-

	subscription_status text	avg_spend numeric	total_revenue numeric
1	No	59.87	170436
2	Yes	59.49	62645

Q9.Are repeat buyers (more than 5 previous purchases) more likely to subscribe?

```
SELECT
CASE
    WHEN previous_purchases > 5 THEN 'Repeat Buyer'
    ELSE 'Others'
END AS buyer_type,
subscription_status,
COUNT(*) AS total_customers
FROM customer
GROUP BY buyer_type, subscription_status;
```

→ Output:-

	buyer_type text	subscription_status text	total_customers bigint
1	Repeat Buyer	No	2518
2	Repeat Buyer	Yes	958
3	Others	Yes	95
4	Others	No	329

- **Customer Segmentation**

Q10.Segment customers INTO New, Returning, and Loyal based on previous purchases and show the count of each segment.

```
SELECT
CASE
    WHEN previous_purchases = 0 THEN 'New'
    WHEN previous_purchases BETWEEN 1 AND 5 THEN 'Returning'
    ELSE 'Loyal'
END AS customer_segment,
COUNT(*) AS total_customers
FROM customer
GROUP BY customer_segment;
```

→ Output:-

	customer_segment text	total_customers bigint
1	Loyal	3476
2	Returning	424

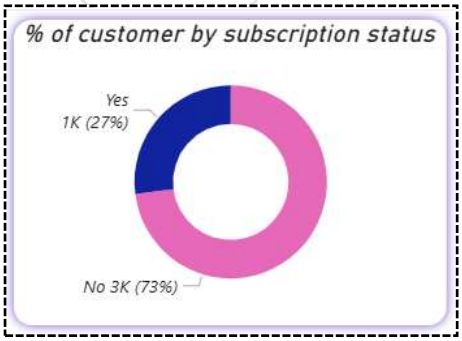
5.Create Dashboard Using Power BI

*We built an interactive dashboard in **Power BI** to present insight visually.*

- ***Overall Business Overview (KPI Cards)***
  - **Total Customers:** 3.9K
  - **Average Purchase Amount:** $59.76
  - **Average Review Rating:** 3.75

Insight:

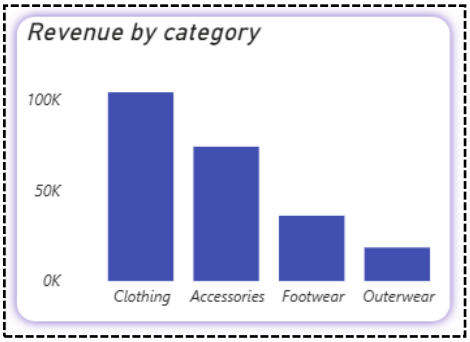
  - The company has a strong customer base of nearly 4,000 transactions.
  - Average spending is around $60 per purchase.
  - Customer satisfaction is moderate (3.75/5), indicating room for service improvement.



- ***Subscription Analysis***
  - 73% customers are Non-Subscribers (3K)
  - 27% customers are Subscribers (1K)

Insight:

  - Majority of customers are not subscribed.
  - High opportunity to increase subscription adoption through loyalty programs and exclusive benefits.



- ***Revenue* by Category**

Revenue ranking:

- Clothing – Highest revenue (~100K+)
- Accessories – Second highest (~70K+)
- Footwear – Moderate (~35K+)
- Outerwear – Lowest (~15K+)

Insight:

- Clothing is the main revenue driver.
- Outerwear has low revenue and may require promotional strategies.

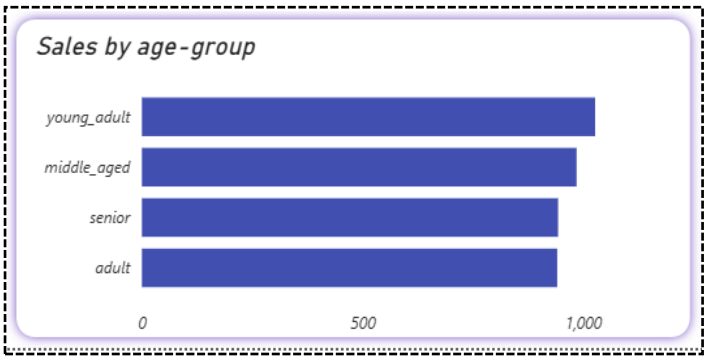
- ***Customers* by Category**

Customer distribution:

- Clothing – Highest customer count
- Accessories – Second highest
- Footwear – Moderate
- Outerwear – Lowest

Insight:

- Category popularity directly impacts revenue.
- High customer volume in Clothing leads to highest sales.



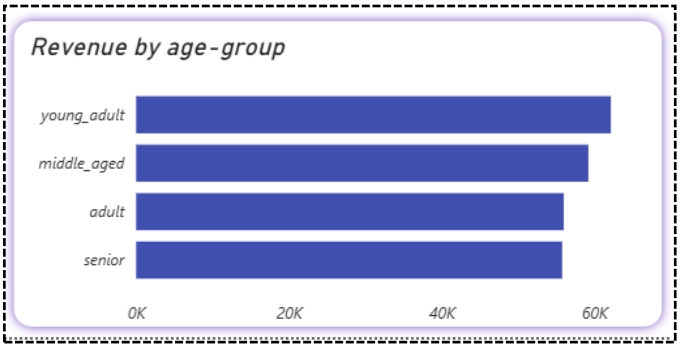
- ***Sales by Age Group***

Sales trend:

  - Young Adults – Highest
  - Middle Aged – Second
  - Senior – Third
  - Adult – Slightly lower

Insight:

  - Young adults are the most active buyers.
  - Middle-aged customers also show strong engagement.
  - Marketing campaigns should focus on these two segments.



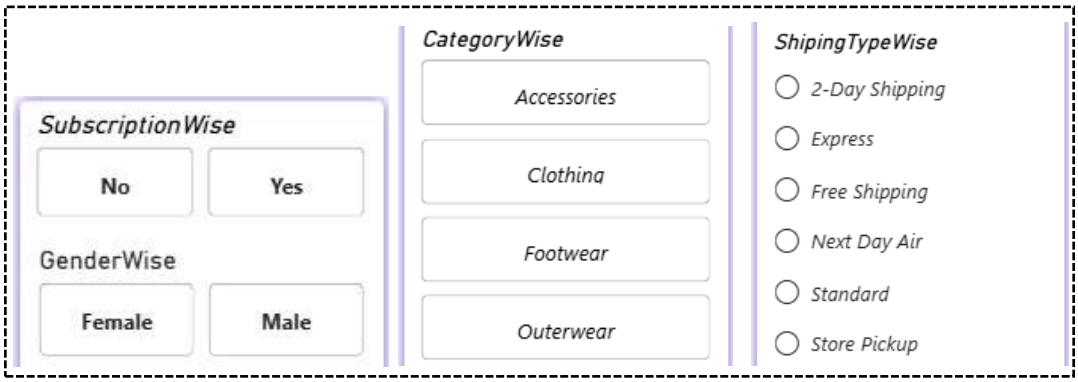
- ***Revenue by Age Group***

Revenue contribution:

  - **Young Adults – Highest (~60K+)**
  - **Middle Aged – Second (~55K+)**
  - **Adults – Third (~50K+)**
  - **Seniors – Slightly lower (~48K+)**

Insight:

  - Young adults contribute the most revenue.
  - Middle-aged segment also has high purchasing power.
  - Revenue distribution is fairly balanced across age groups.



- ***Dashboard Filters (Interactive Analysis)***

The dashboard allows filtering by:

  - Subscription Status
  - Gender (Male / Female)
  - Category
  - Shipping Type (2-Day, Express, Free, Next Day Air, Standard, Store Pickup)

Insight:

  - Helps analyze customer behavior dynamically.
  - Useful for decision-making and segmentation analysis**.**

- ***Key Business Insights***
  - Clothing dominates both revenue and customer base.
  - Young adults are the most valuable segment.
  - Subscription penetration is relatively low (27%).
  - Average rating suggests scope for improving customer satisfaction.
  - Outerwear category requires performance improvement strategies.

- ***Business Recommendations***
  - Promote subscription plans to increase recurring revenue.
  - Offer targeted discounts for Outerwear category.
  - Focus marketing campaigns on Young Adult and Middle-Aged segments.
  - Improve product quality to increase review ratings.
  - Implement loyalty programs for repeat customers.

- ***Conclusion***

This dashboard provides a comprehensive overview of:

  - Revenue performance
  - Customer segmentation
  - Product category trends
  - Subscription behavior
  - Age-based sales patterns

These insights help in making **data-driven business decisions** to increase revenue and customer engagement.

Final_Dashboard :-