Customer Shopping Behaviour Analysis project

Project Overview

This project analyses customer shopping behaviour to identify purchasing patterns , customer segments and revenue drivers. The workflow includes data cleaning in Python(Jupyter notebook), business analysis using SQL (Postgre SQL), and interactive Dashboard using Power BI . The goal is to simulate a real-world data analyst workflow: transforming raw customer data into actionable business insights.

DATASET

- **Rows – 4000**
- **Columns – 18**
- **Columns names – [Customer , Age , Gender , Item Purchased , Category , Purchase Amount , Location , Size , Color , Season , Review Rating , Subscription Status , Shipping Type , Discount Applied , Promo Code Used , Payment Method , Frequency of purchases]**

Performing EDA using Python

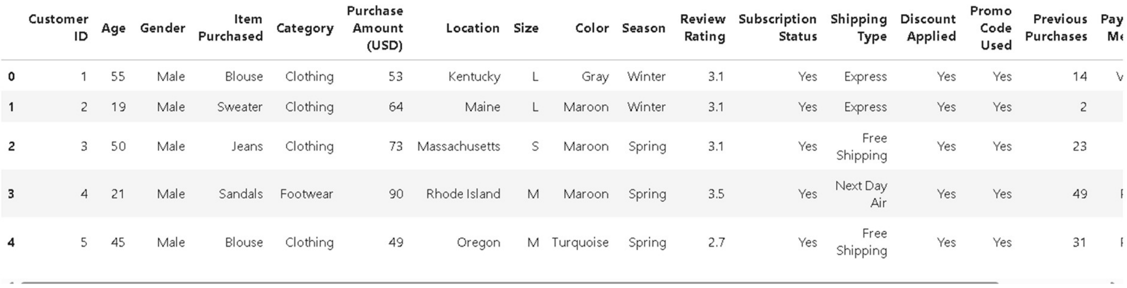
Here I used the python and their libraby pandas to preforming EDA

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

- **import dataset - df = pd.read_csv(r"C:\Users\devan\OneDrive\Attachments\Desktop\Analysis project\Customer data analysis projects\customer_shopping_behavior.csv")**

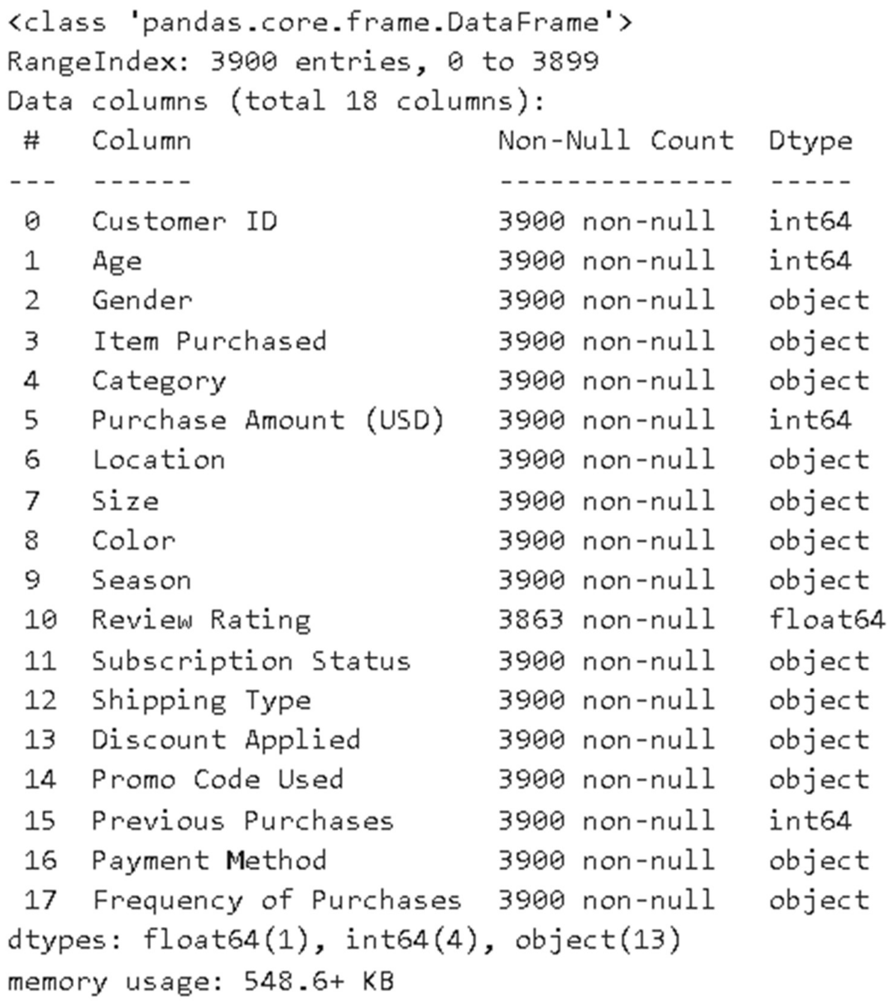
print(df)

- **df.head()**



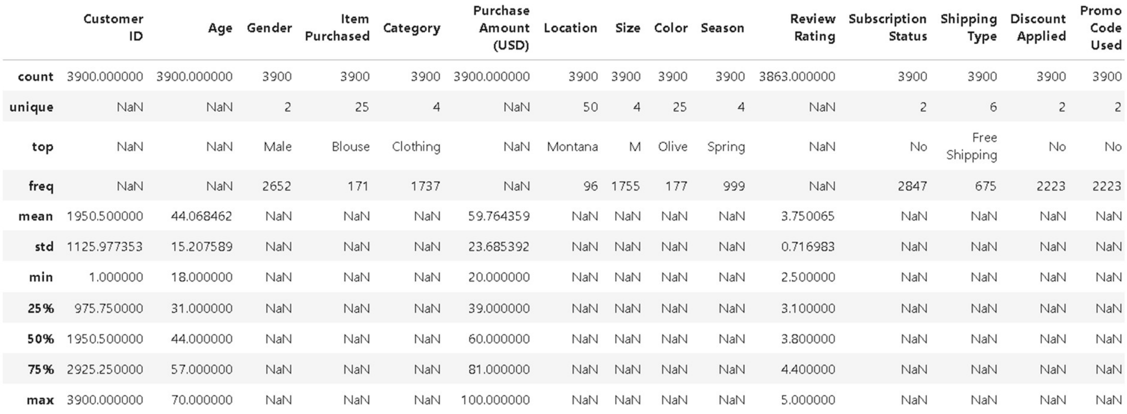
	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Pay Me
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	V
1	2	19	Male	Sweater	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

- **df.info()**



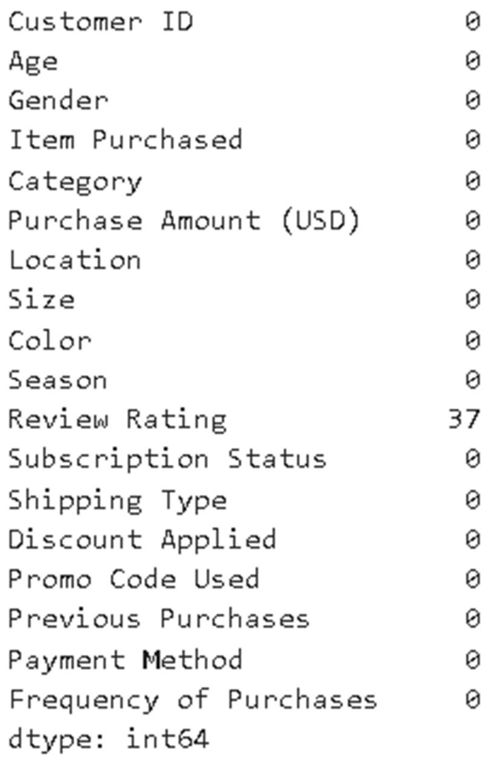
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Customer ID             3900 non-null   int64  
 1   Age                     3900 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(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

➢ df.isnull().sum()

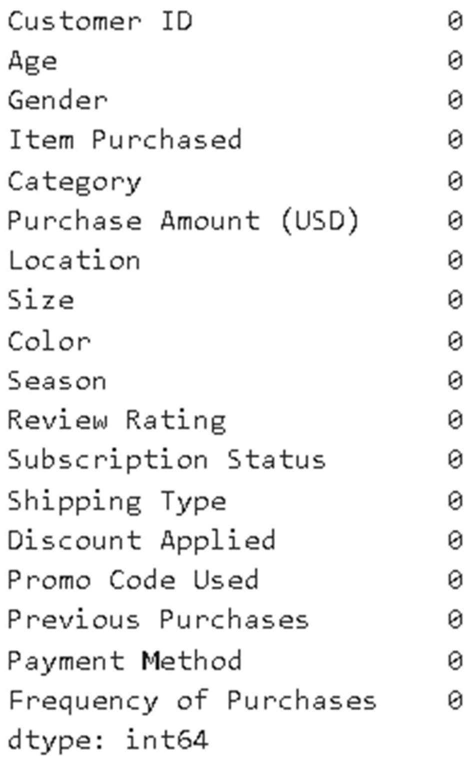


```
Customer ID               0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating            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['ReviewRating'].fillna(4.1)**

# fill the null value of review rating is 4.1

df.isnull().sum()



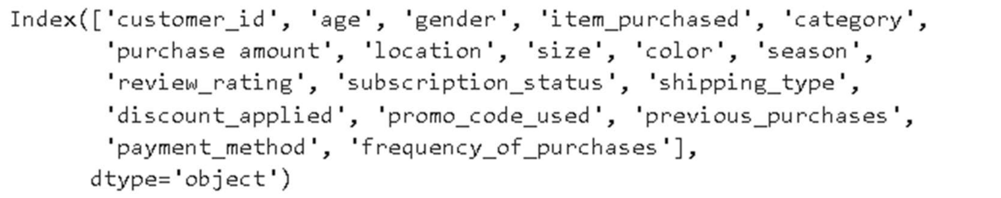
```
Customer ID                 0
Age                         0
Gender                      0
Item Purchased              0
Category                    0
Purchase Amount (USD)       0
Location                    0
Size                        0
Color                       0
Season                    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

- **df.columns = df.columns.str.lower()**

  df.columns = df.columns.str.replace(' ','_')

  df = df.rename(columns=({'purchase_amount_(usd)':'purchase amount'}))

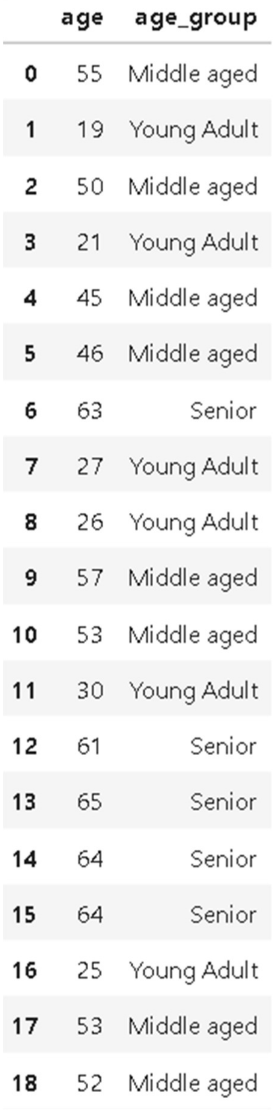
df.columns



```
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')
```

- **# create a new new columns age_group**

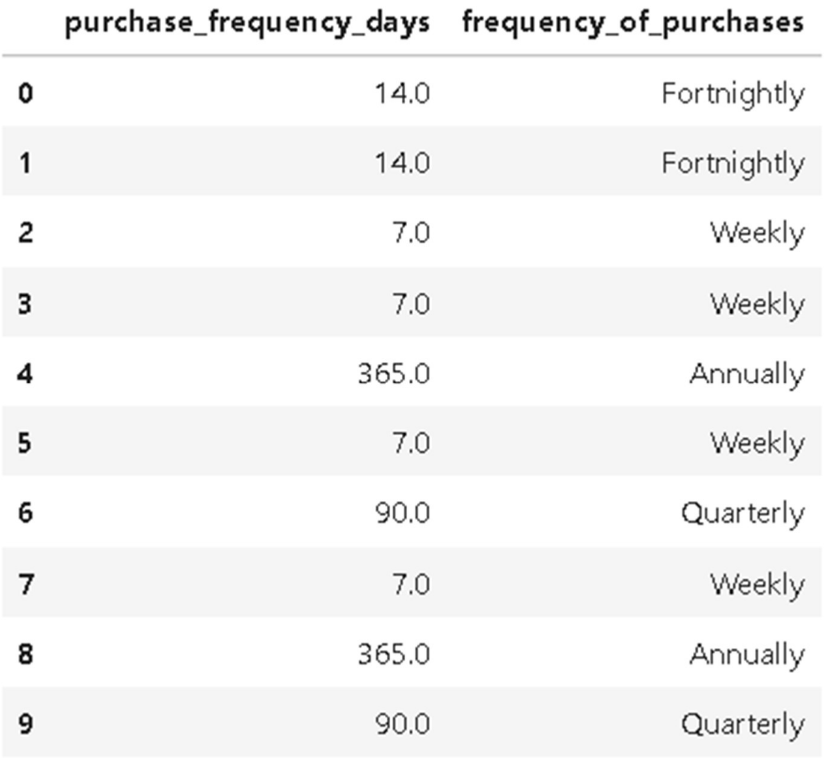
```
  labels = ['Young Adult','Adult','Middle aged',
'Senior']
df['age_group'] = pd.qcut(df['age'],q=4,labels=labels)
                                        # making a new column
                                          age_group
                                        # and devided age  group on
                                          the basis of age
```



	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
10	53	Middle aged
11	30	Young Adult
12	61	Senior
13	65	Senior
14	64	Senior
15	64	Senior
16	25	Young Adult
17	53	Middle aged
18	52	Middle aged

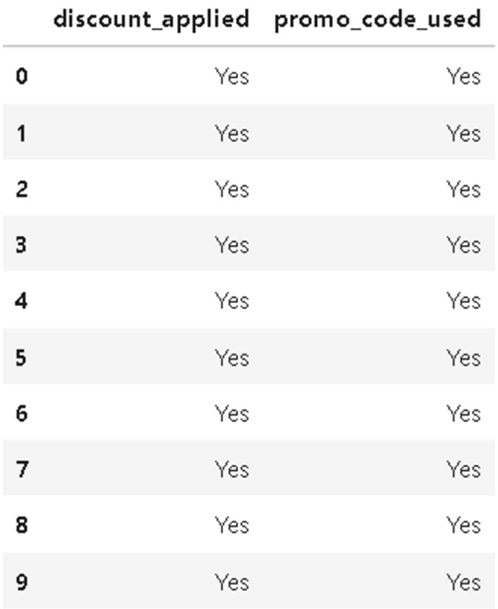
➢ # create columns purchase_frequency_days

```
frequency_mapping = {
    'Fortnightly': 14,            # replace frequency of
    'Weekly': 7,                    purchase columns
    'Monthly': 30,                  values  with
    'Quarterly' : 90,               numerical numbers.
    'Bi-Weekly' : 14,
    'Annually' : 365,
    'Even 3 Months' : 90
}

df['purchase_frequency_days'] = df['frequency_of_purchases'].map(frequency_mapping)

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



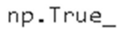
	purchase_frequency_days	frequency_of_purchases
0	14.0	Fortnightly
1	14.0	Fortnightly
2	7.0	Weekly
3	7.0	Weekly
4	365.0	Annually
5	7.0	Weekly
6	90.0	Quarterly
7	7.0	Weekly
8	365.0	Annually
9	90.0	Quarterly

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



	discount_applied	promo_code_used
0	Yes	Yes
1	Yes	Yes
2	Yes	Yes
3	Yes	Yes
4	Yes	Yes
5	Yes	Yes
6	Yes	Yes
7	Yes	Yes
8	Yes	Yes
9	Yes	Yes

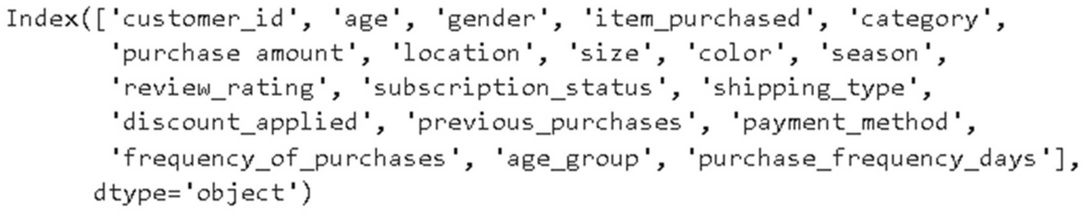
- (df['discount_applied'] == df['promo_code_used']).all()



```
np.True_
```

- df.drop('promo_code_used',axis=1,inplace=True)

```
print(df)              # deleted the
                         promo_code_used columns
                         bcz they and discount applied
                         have same valued we check
                         above
df.columns
```

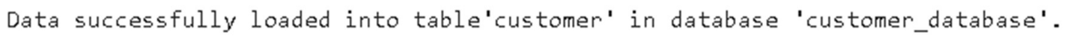


```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'previous_purchases', 'payment_method',
       'frequency_of_purchases',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

- pip install psycopg2_binary sqlalchemy
- from sqlalchemy import create_engine

```
username = "postgres"
password = "MANDIRps%406"
host = 'localhost'
port = "5433"
database = "customer_database"
```

```
engine =
create_engine('postgresql://postgres:MANDIRp%
406@localhost:5433/customer_database')
# step 2 Load dataframe into Postgresql
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}'.")
```

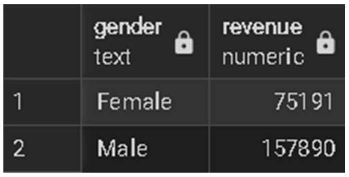


```
Data successfully loaded into table'customer' in database 'customer_database'.
```

Data Analysis using SQL (Business Transactions)

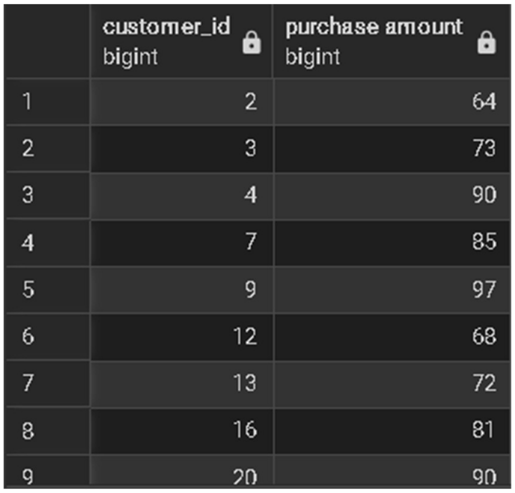
We performed structured analysis in PostgreSQL to answer key business questions:

1. What is the total revenue generated by male vs female customer?



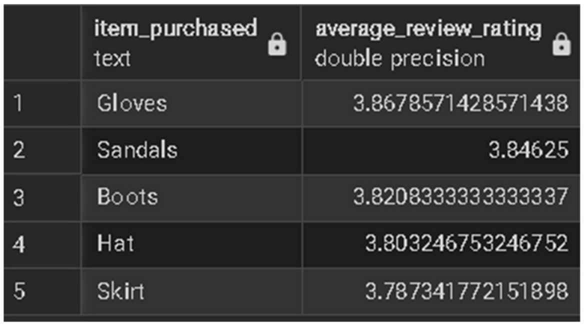
	gender text	revenue numeric
1	Female	75191
2	Male	157890

2. .Which customers used a discount but still spent more than the average purchase amount?



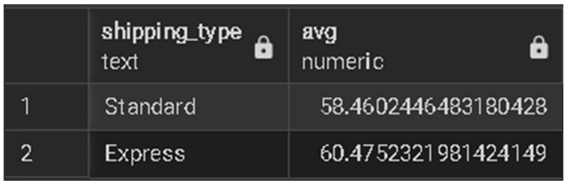
	customer_id bigint	purchase amount bigint
1	2	64
2	3	73
3	4	90
4	7	85
5	9	97
6	12	68
7	13	72
8	16	81
9	20	90

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



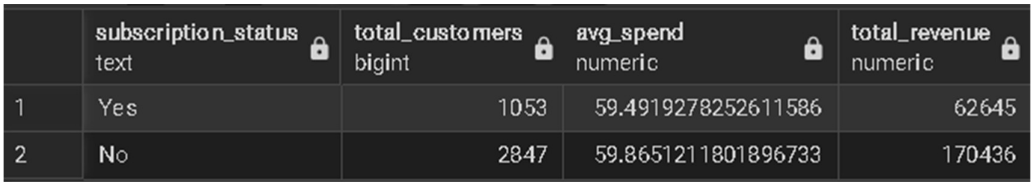
	item_purchased text	average_review_rating double precision
1	Gloves	3.8678571428571438
2	Sandals	3.84625
3	Boots	3.8208333333333337
4	Hat	3.803246753246752
5	Skirt	3.787341772151898

4. Compare the average purchase amounts between standard and express shipping?



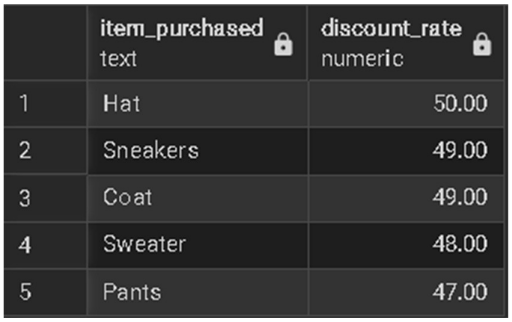
	shipping_type text	avg numeric
1	Standard	58.4602446483180428
2	Express	60.4752321981424149

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



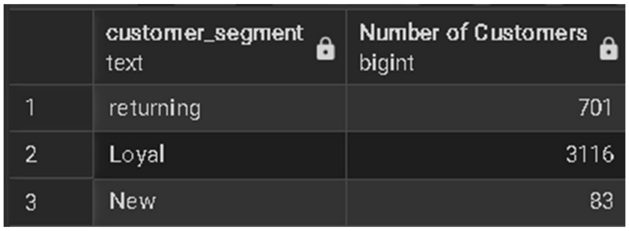
	subscription_status text	total_customers bigint	avg_spend numeric	total_revenue numeric
1	Yes	1053	59.4919278252611586	62645
2	No	2847	59.8651211801896733	170436

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



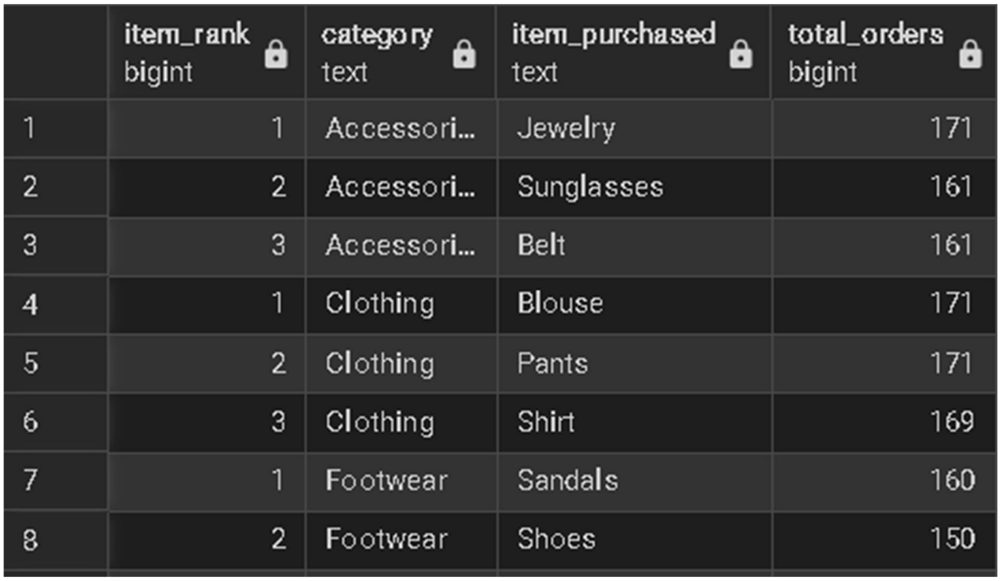
	item_purchased text	discount_rate numeric
1	Hat	50.00
2	Sneakers	49.00
3	Coat	49.00
4	Sweater	48.00
5	Pants	47.00

7. .Segment customers into New, Returning and Loyal based on their total number of previous purchases, and show the count of each segment.



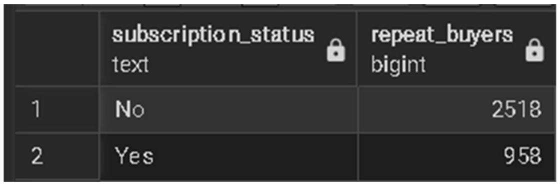
	customer_segment text	Number of Customers bigint
1	returning	701
2	Loyal	3116
3	New	83

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



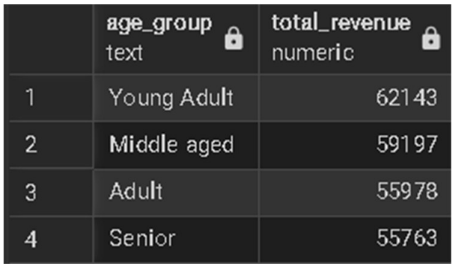
	item_rank bigint	category text	item_purchased text	total_orders bigint
1	1	Accessori...	Jewelry	171
2	2	Accessori...	Sunglasses	161
3	3	Accessori...	Belt	161
4	1	Clothing	Blouse	171
5	2	Clothing	Pants	171
6	3	Clothing	Shirt	169
7	1	Footwear	Sandals	160
8	2	Footwear	Shoes	150

9. Are customers who are repeat buyers (more than 5 previous purchase) also likely to subscribe?



	subscription_status text	repeat_buyers bigint
1	No	2518
2	Yes	958

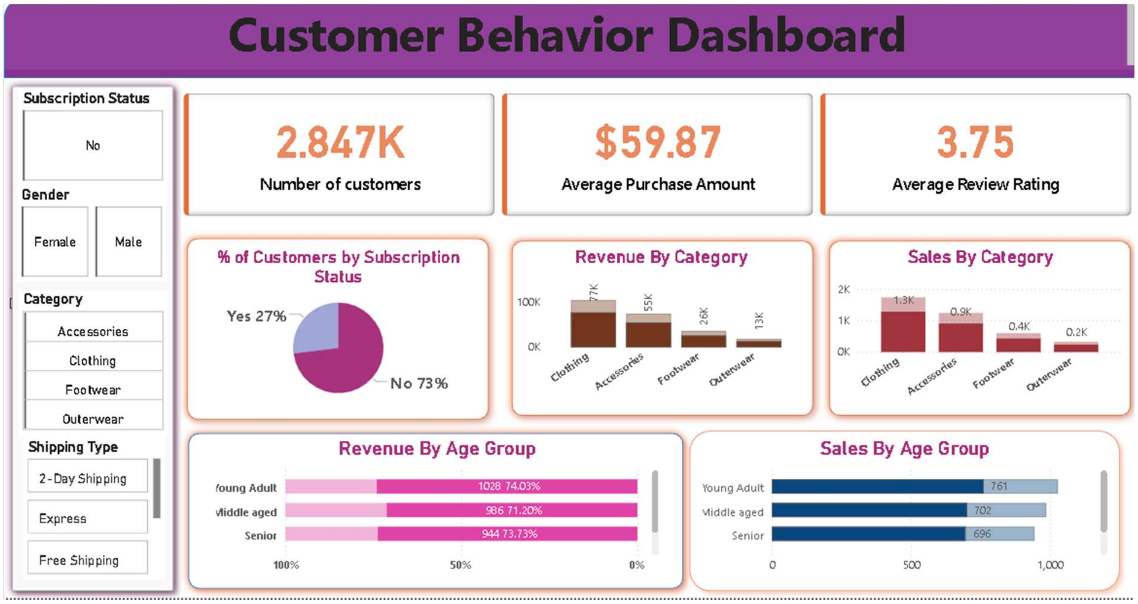
10. What is the revenue contribution of each age group?



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

Dashboard in Power BI

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



The Power BI dashboard provides an interactive overview of customer metrics. KPI cards summarize customer count, purchase value, and review ratings. Category charts highlight revenue distribution, while age-group visuals reveal generational spending patterns. Filters allow

dynamic exploration by gender, shipping type, and subscription behavior.

Key Insights

- **Non-subscribers represent a large revenue opportunity**
- **Young adults drive the highest purchase frequency**
- **Clothing category dominates sales volume**
- **Discount usage increases conversion but impacts margins**
- **Express shipping customers tend to spend more**

Business Recommendations

- **Launch targeted subscription campaigns**
- **Strengthen loyalty rewards for repeat buyers**
- **Optimize discount strategies for profitability**
- **Promote high-rated products aggressively**
- **Focus marketing on high-value age segments**