Ecommerce_customer_churned-analysis

1. Project Overview

This project focuses on analyzing customer behavior and predicting churn using a large-scale e-commerce dataset containing 50,000 customer records. The objective is to understand customer engagement patterns and identify churn-driving factors.

2. Dataset Summary

Total Records: 50,000

Total Columns: 25

Key Features:

Demographics, Engagement Metrics, Purchase Behavior, Customer Value,

Churn Status.

3. Exploratory Data Analysis (Python)

Data cleaning, missing value handling using median imputation, feature engineering including churned_status and age_group creation were performed using Pandas and NumPy.

Data Loading: Imported the datasets using pandas

In [21]:
```python
import pandas as pd
import numpy as np
```

In [22]:
```python
df=pd.read_csv("ecommerce_customer_churn_dataset.csv")
```

Initial Exploration: Used df.info() to check structure and .describe() for summary statistics.

In [23]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 25 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Age                            47505 non-null  float64
 1   Gender                         50000 non-null  object
 2   Country                        50000 non-null  object
 3   City                           50000 non-null  object
 4   Membership_Years               50000 non-null  float64
 5   Login_Frequency                50000 non-null  int64
 6   Session_Duration_Avg           46601 non-null  float64
 7   Pages_Per_Session              47000 non-null  float64
 8   Cart_Abandonment_Rate          50000 non-null  float64
 9   Wishlist_Items                 46000 non-null  float64
 10  Total_Purchases                50000 non-null  float64
 11  Average_Order_Value            50000 non-null  float64
 12  Days_Since_Last_Purchase       47000 non-null  float64
 13  Discount_Usage_Rate            46500 non-null  float64
 14  Returns_Rate                   45509 non-null  float64
 15  Email_Open_Rate                47472 non-null  float64
 16  Customer_Service_Calls         49832 non-null  float64
 17  Product_Reviews_Written        46500 non-null  float64
 18  Social_Media_Engagement_Score  44000 non-null  float64
 19  Mobile_App_Usage               45000 non-null  float64
 20  Payment_Method_Diversity       47500 non-null  float64
 21  Lifetime_Value                 50000 non-null  float64
 22  Credit_Balance                 44500 non-null  float64
 23  Churned                        50000 non-null  int64
 24  Signup_Quarter                 50000 non-null  object
dtypes: float64(19), int64(2), object(4)
memory usage: 9.5+ MB
```

In [24]:
```
df.describe()
```

Out[24]:

	Age	Membership_Years	Login_Frequency	Session_Duration_Avg
count	47505.000000	50000.000000	50000.000000	46601.000000
mean	37.802968	2.984009	11.624660	27.660754
std	11.834668	2.059105	7.810657	10.871013
min	5.000000	0.100000	0.000000	1.000000
25%	29.000000	1.400000	6.000000	19.700000
50%	38.000000	2.500000	11.000000	26.800000
75%	46.000000	4.000000	17.000000	34.700000
max	200.000000	10.000000	46.000000	75.600000

8 rows × 21 columns

Missing Data Handling: Checked for null values and imputed missing values in the ecommerce_churned_dataset by the pandas and numpy

In [28]:
```
df.isnull().sum()
```

Out[28]:
```
Age                             2495
Gender                             0
Country                            0
City                               0
Membership_Years                   0
Login_Frequency                    0
Session_Duration_Avg            3399
Pages_Per_Session               3000
Cart_Abandonment_Rate              0
Wishlist_Items                  4000
Total_Purchases                    0
Average_Order_Value                0
Days_Since_Last_Purchase        3000
Discount_Usage_Rate             3500
Returns_Rate                    4491
Email_Open_Rate                 2528
Customer_Service_Calls           168
Product_Reviews_Written         3500
Social_Media_Engagement_Score   6000
Mobile_App_Usage                5000
Payment_Method_Diversity        2500
Lifetime_Value                     0
Credit_Balance                  5500
Churned                            0
Signup_Quarter                     0
dtype: int64
```

We have multiple missing values in the every column then we need to fill the missing values in the data set by using methods

In [29]:
```
numeric_cols = [
```

```
    'Age',
    'Session_Duration_Avg',
    'Pages_Per_Session',
    'Wishlist_Items',
    'Days_Since_Last_Purchase',
    'Discount_Usage_Rate',
    'Returns_Rate',
    'Email_Open_Rate',
    'Customer_Service_Calls',
    'Product_Reviews_Written',
    'Social_Media_Engagement_Score',
    'Mobile_App_Usage',
    'Payment_Method_Diversity',
    'Credit_Balance'
]

df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].median())
```

This code is used to fill the all missing values in the median score

In [30]:
```
df.isnull().sum()
```

Out[30]:
```
Age                              0
Gender                           0
Country                          0
City                             0
Membership_Years                 0
Login_Frequency                  0
Session_Duration_Avg             0
Pages_Per_Session                0
Cart_Abandonment_Rate            0
Wishlist_Items                   0
Total_Purchases                  0
Average_Order_Value              0
Days_Since_Last_Purchase         0
Discount_Usage_Rate              0
Returns_Rate                     0
Email_Open_Rate                  0
Customer_Service_Calls           0
Product_Reviews_Written          0
Social_Media_Engagement_Score    0
Mobile_App_Usage                 0
Payment_Method_Diversity         0
Lifetime_Value                   0
Credit_Balance                   0
Churned                          0
Signup_Quarter                   0
dtype: int64
```

Null values are filled by the median score

In [31]:
```
df.shape
```

Out[31]:
```
(50000, 25)
```

Created age_group column by binning customer ages.

```python
import numpy as np

conditions = [
    df['AGE'] < 25,
    (df['AGE'] >= 25) & (df['AGE'] <= 35),
    df['AGE'] > 35
]

choices = [
    'low_age',
    'middle_age',
    'high_age'
]

df['age_group'] = np.select(conditions, choices, default='other_age')
```

4.Data Analysis using SQL

Overall churn rate: 28.9%

Country-wise churn analysis revealed higher churn in Australia, Canada, and USA.

Low login frequency and session duration strongly correlate with churn

In [58]:
```python
import pandas as pd
import sqlite3
```

In [59]:
```python
import sqlite3

conn = sqlite3.connect("churn.db")
df.to_sql("customers", conn, if_exists="replace", index=False)
```

Out[59]: 50000

Select the data with pandas and sql

In [60]:
```python
pd.read_sql("SELECT * FROM customers;", conn)
```

Out[60]:

	AGE	GENDER	COUNTRY	CITY	MEMBERSHIP_YEARS	LOGIN_FREQU
0	43.0	Male	France	Marseille	2.9	
1	36.0	Male	UK	Manchester	1.6	
2	45.0	Female	Canada	Vancouver	2.9	
3	56.0	Female	USA	New York	2.6	
4	35.0	Male	India	Delhi	3.1	
...	...	...	...	...	...	
49995	38.0	Female	USA	Los Angeles	10.0	
49996	37.0	Male	USA	Chicago	1.4	
49997	44.0	Female	USA	Phoenix	2.8	
49998	41.0	Female	USA	Chicago	2.9	
49999	56.0	Male	UK	Leeds	2.2	

50000 rows × 27 columns

Overall churn rate: 28.9%

In [61]:
```
query = """
SELECT
    ROUND(AVG(CHURNED) * 100, 2) AS churn_rate
FROM customers;
"""
pd.read_sql(query, conn)
```

Out[61]:

	churn_rate
0	28.9

Calculate the churn_rate in the country-wise

In [62]:
```
#country-wise
query= """
select COUNTRY,
   ROUND(AVG(CHURNED)*100,2) AS churn_rate
from customers
GROUP BY COUNTRY
ORDER BY churn_rate DESC;
"""
```

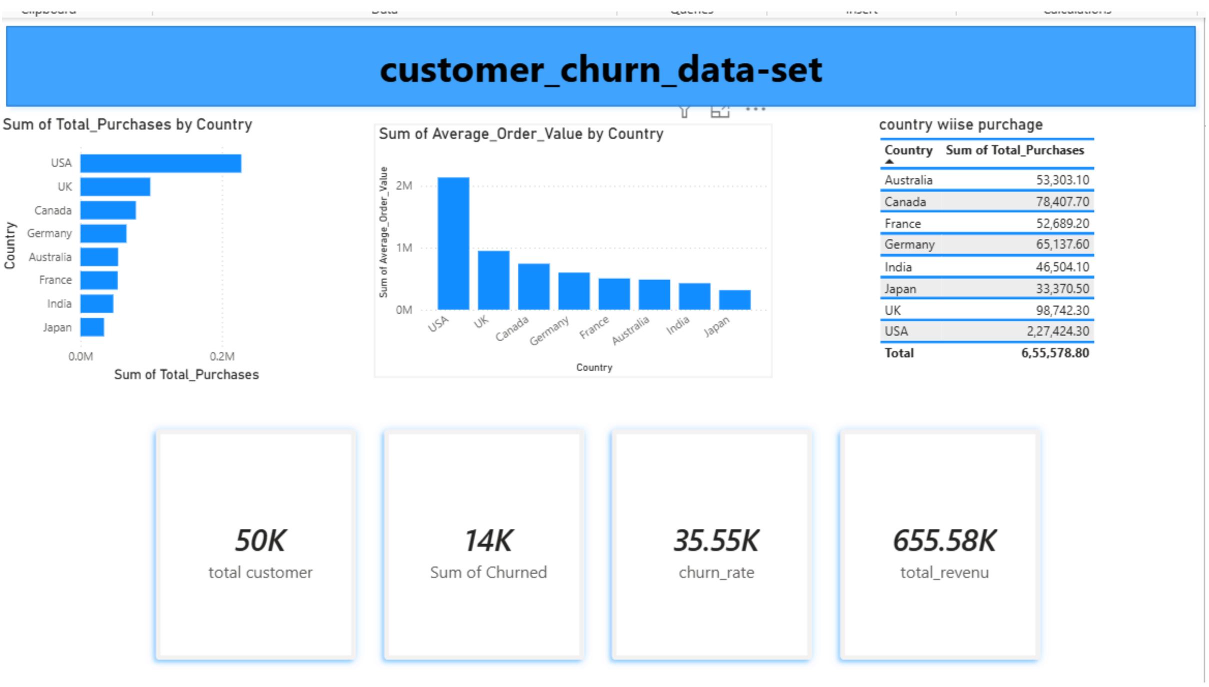
```
pd.read_sql(query,conn)
```

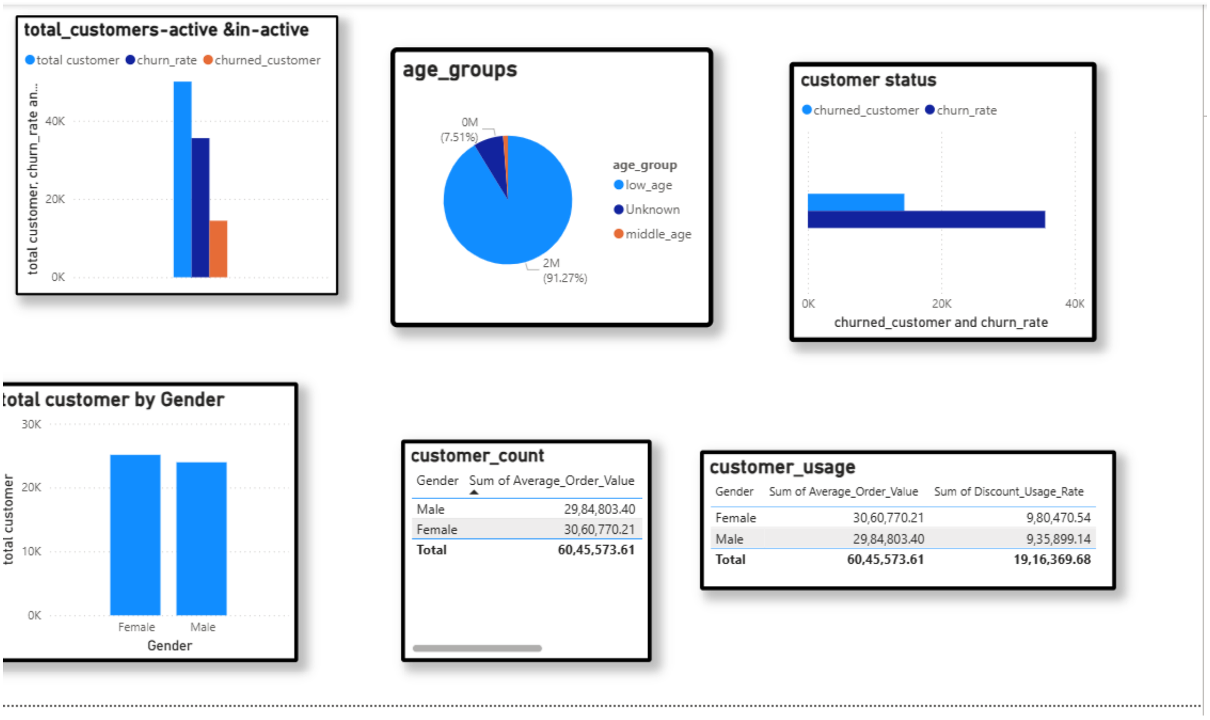
Out[62]:

	COUNTRY	churn_rate
0	Australia	29.89
1	Canada	29.35
2	USA	29.08
3	India	29.01
4	Germany	28.83
5	UK	28.79
6	Japan	27.83
7	France	27.29

5.Dashboard in Power BI

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





country wise customers active

total_Purchases by Country

Australia 53,303.10	Germany 65,137.60	UK 98,742.30
Canada 78,407.70	India 46,504.10	USA 2,27,424.30
France 52,689.20	Japan 33,370.50	