Q6 Bank Customer Clustering

Aiming to enhance the operation efficiency, tailor services and manage risks effectively, it is critical for banks to cluster customers into several clusters. In this task, We first do the data manipulation and analysis to give a comprehensive insights into the data.

Data Manipulation and Analysis

I analyze the data mainly from the following aspects: Missing Value Detection, Anomaly Detection, Statistics properties

Missing Value Detection

It can be seen that there is no missing value in the dataet

```
Data columns (total 9 columns):
 #   Column                  Non-Null Count    Dtype
---  ------                  --------------    -----
 0   TransactionID           1041614 non-null  object
 1   CustomerID              1041614 non-null  object
 2   CustomerDateOfBirth     1041614 non-null  object
 3   CustGender              1041614 non-null  object
 4   CustLocation            1041614 non-null  object
 5   CustAccountBalance      1041614 non-null  float64
 6   TransactionDate         1041614 non-null  object
 7   TransactionTime         1041614 non-null  int64
 8   TransactionAmount (INR) 1041614 non-null  float64
dtypes: float64(2), int64(1), object(6)
memory usage: 79.5+ MB
```

Figure 1 Dataset Information

Outlier/Abnormal Detection

Although there is no missing value in the data, it does not mean all of the sample is normal. As for the gender column, there is an unknown gender label **T,** I simply **remove** the sample whose gender attribute is T.

```
M    760978
F    280635
T         1
Name: CustGender, dtype: int64
```

Figure 2 Gender Outlier

Statistics properties

In our dataset, three columns are numeric and the rest are str type

	CustAccountBalance	TransactionTime	TransactionAmount (INR)
count	1.041614e+06	1.041614e+06	1.041614e+06
mean	1.149986e+05	1.571221e+05	1.566096e+03
std	8.467609e+05	5.126352e+04	6.561464e+03
min	0.000000e+00	0.000000e+00	0.000000e+00
25%	4.728140e+03	1.240410e+05	1.600000e+02
50%	1.676852e+04	1.642470e+05	4.575000e+02
75%	5.742885e+04	2.000220e+05	1.200000e+03
max	1.150355e+08	2.359590e+05	1.560035e+06

Figure 3 Numerical Variable Statistics

Visualization

Variable Relationship

Numerical variable plays an important role in machine learning prediction or classification. Hence, I visualize the correlationship between the numerical variables through heat map and scatter plot.

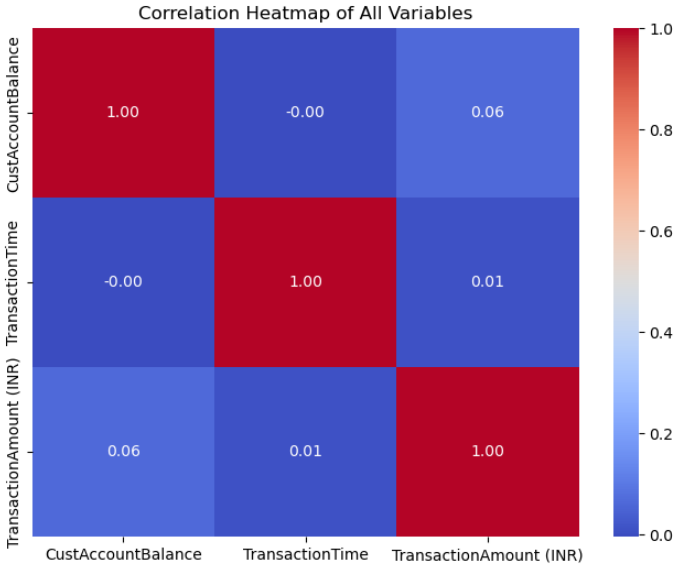

Figure 4 Correlationship Heatmap of numerical variables

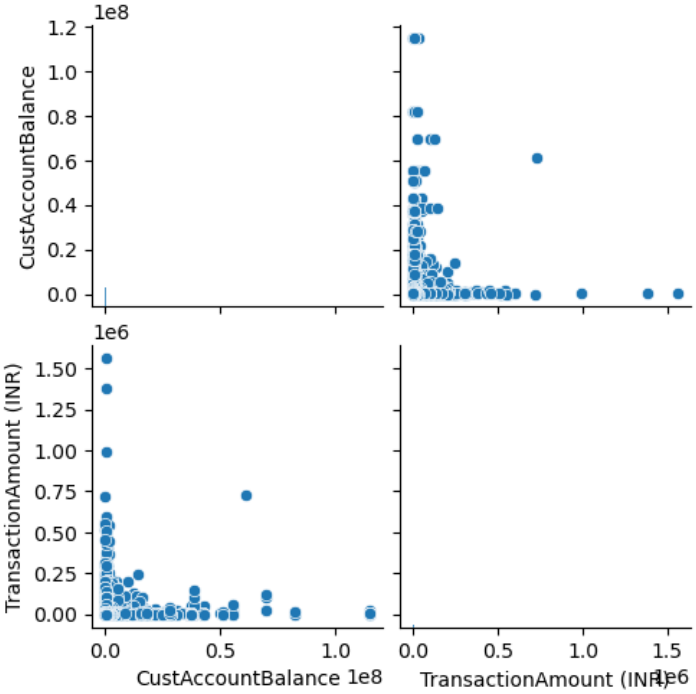

Figure 5 Scatter Plot between CustAccountBalance and Transaction Amount

From the figures above, we can find that the correlationship between Transaction amount and customer account balance is stronger than the correlationship between Transaction and transaction time.

In addition, gender also plays an important role in daily transactions activities. Therefore, I plot the gender distribution in our data with bar plot.

Figure 6 Gender Distribution

Transaction Distribution

To better understand the transaction distribution and the frequency per customer, I visualize the customer transaction frequency with bar plot, horizontal axis refers to the number of transactions, and the vertical axis refers to the number of customers.

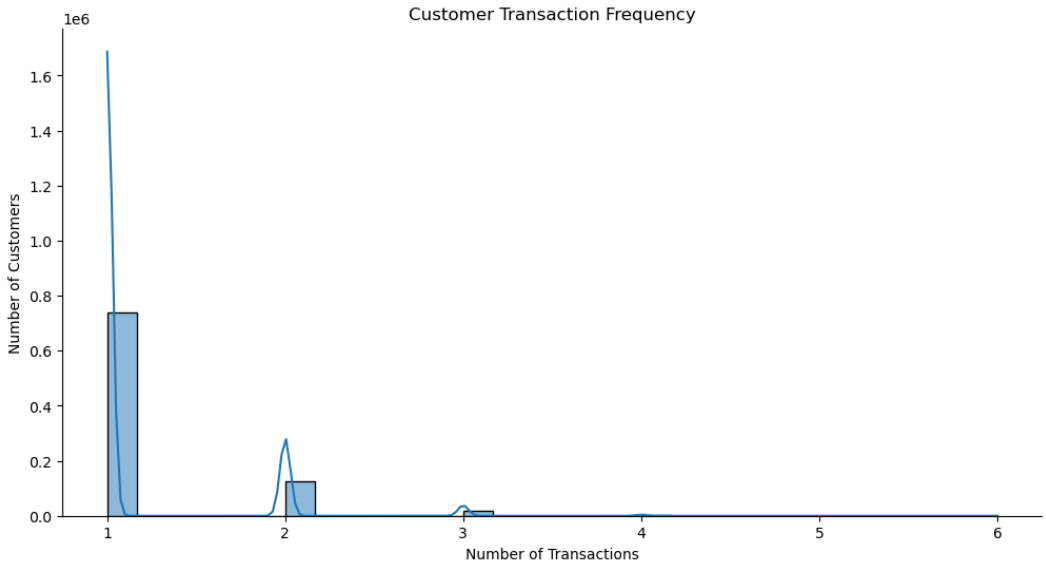

Figure 7 Customer Transaction Frequency

It is seen that most customers have 1 to 2 transactions, it follows a long tail distribution. This can be also proved by the average transaction amount per customer distribution as below.

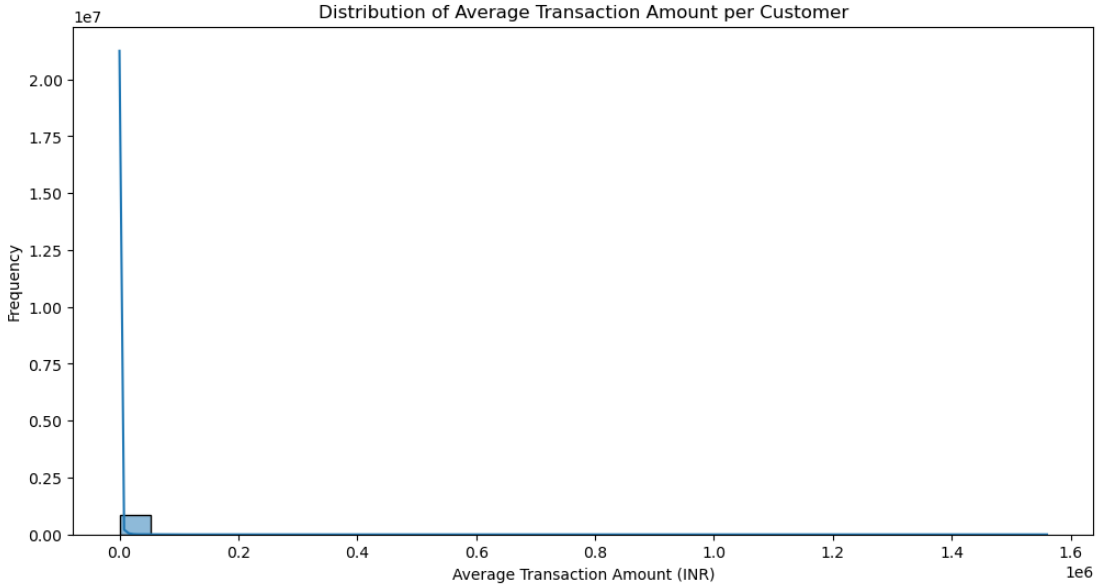

Figure 8 Average transaction amount per customer

Given the transaction date and the transaction amount, we can plot the transaction trends over time as below:

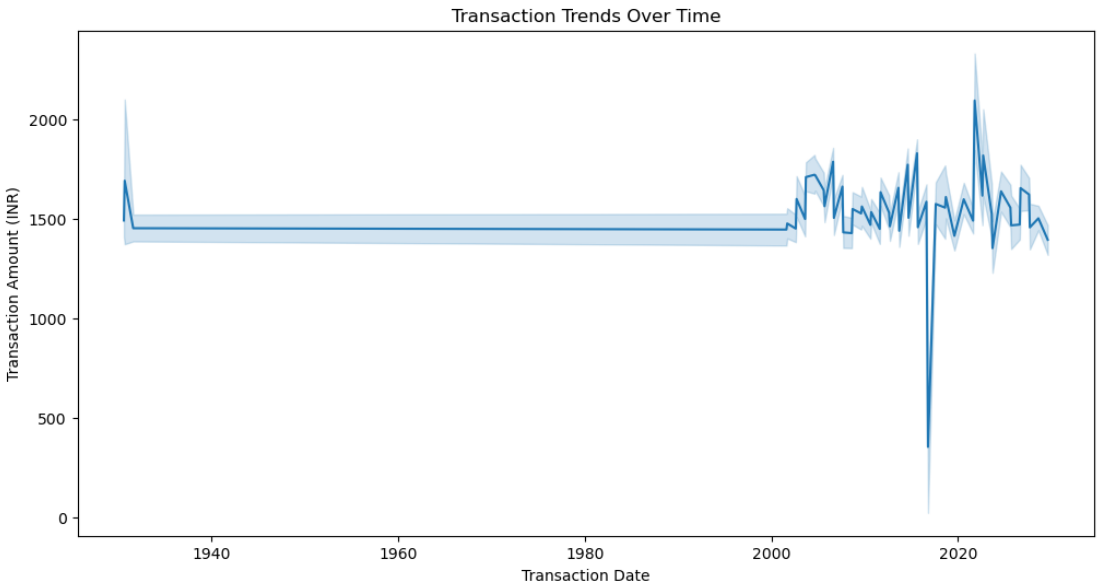

Figure 9 Transaction Trends Over time

Transactions occur mainly after 2000 and have a tendency to fluctuate more around 2020, counting all the way to 2029, which represents data that is actually a synthetic data.

Clustering

I use three cluster algorithms to cluster the customers according to their transactions amount. These three clustering algorithms are Kmeans, DBSCAN, HDBSCAN.

```
def kmeans_fit(self, features: Optional[list] = None):
    data = self.data.copy()
    if features:
        data = data.loc[:, features]
    # data = pd.get_dummies(data=data)
    kmeans_model =KMeans(n_clusters=8, n_init="auto", random_state=42)
    labels = kmeans_model.fit_predict(data)

    # Visualize the clusters
    plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis', edgecolor='k', s=50)
    plt.title('KMeans Clustering Result')
    plt.show()
```

```
def dbscan_fit(self, features: Optional[list] = None):
    data = self.data.copy()
    if features:
        data = data.loc[:, features]
    # data = pd.get_dummies(data=data)
    dbscan_model = DBSCAN()
    labels = dbscan_model.fit_predict(data)

    # Visualize the clusters
    plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis', edgecolor='k', s=50)
    plt.title('DBSCAN Clustering Result')
    plt.show()
```

```python
def hdbscan_fit(self, features: Optional[list] = None):
    data = self.data.copy()
    if features:
        data = data.loc[:, features]
    # data = pd.get_dummies(data=data)
    hdbscan_model = HDBSCAN(min_cluster_size=8)
    labels = hdbscan_model.fit_predict(data)

    # Visualize the clusters
    plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis', edgecolor='k', s=50)
    plt.title('HDBSCAN Clustering Result')
    plt.show()
```

And I find that the customers in the same cluster shares the similar account balance for all this three clustering algorithms.