Experiment 8

Name: Hatim Sawai
Uid: 2021300108
Batch: C

Load the dataset

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

In [ ]:
```
df = pd.read_csv("CC GENERAL.csv")
df.head()
```

Out[ ]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTA
0	C10001	40.900749	0.818182	95.40	0.00	
1	C10002	3202.467416	0.909091	0.00	0.00	
2	C10003	2495.148862	1.000000	773.17	773.17	
3	C10004	1666.670542	0.636364	1499.00	1499.00	
4	C10005	817.714335	1.000000	16.00	16.00	

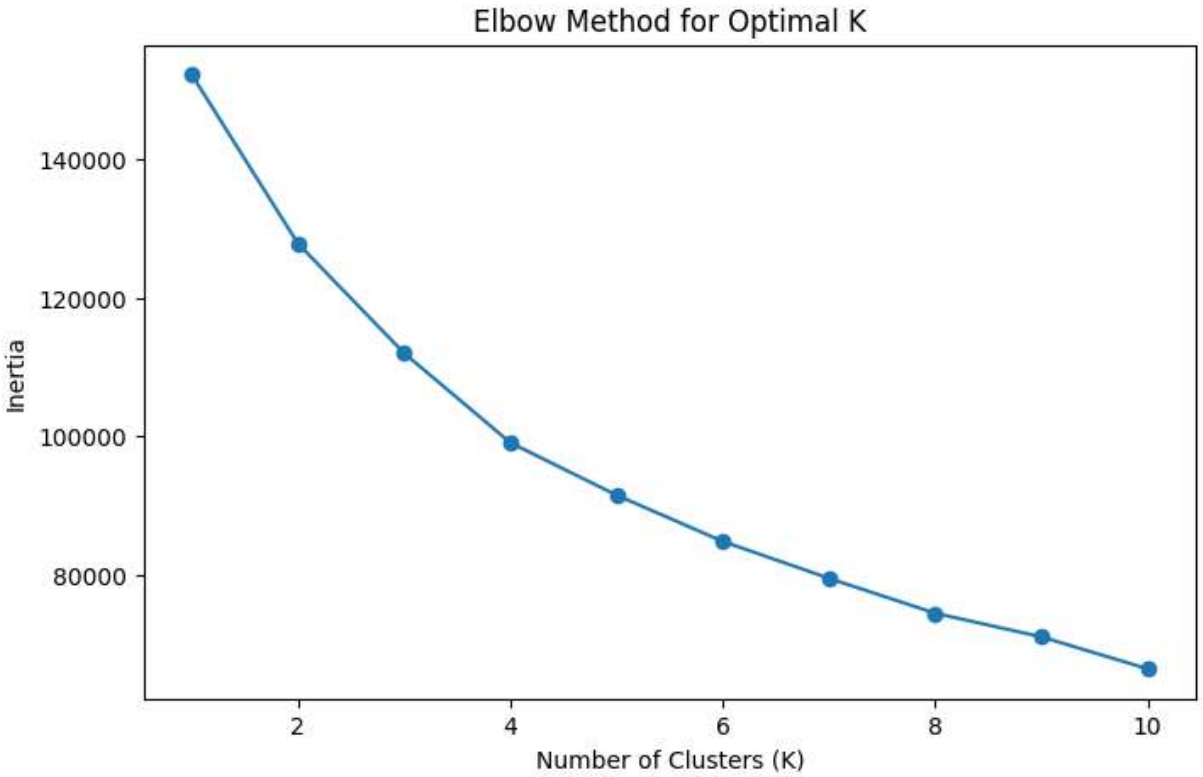
Data Preprocessing

In [ ]:
```
# Drop the 'CUST_ID' column as it's an identifier
df = df.drop("CUST_ID", axis=1)
# Handle missing values
df = df.fillna(0)  # You may choose another strategy based on your dataset
# Standardize the features
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df)
```

Determine no. of clusters using elbow method

In [ ]:
```
# Use the Elbow method to find the optimal number of clusters
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data_scaled)
    inertia.append(kmeans.inertia_)
```

```
# Plot the Elbow curve
plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), inertia, marker="o")
plt.title("Elbow Method for Optimal K")
plt.xlabel("Number of Clusters (K)")
plt.ylabel("Inertia")
plt.show()
```



Apply K-Means Clustering

In [ ]:
```
# Choose the optimal K value and fit the K-Means model
optimal_k = 3  # Replace with the chosen K value
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
df["Cluster"] = kmeans.fit_predict(data_scaled)
```

In [ ]:
```
# Analyze the characteristics of each cluster
cluster_means = df.groupby("Cluster").mean()
cluster_means
```

Out[ ]:

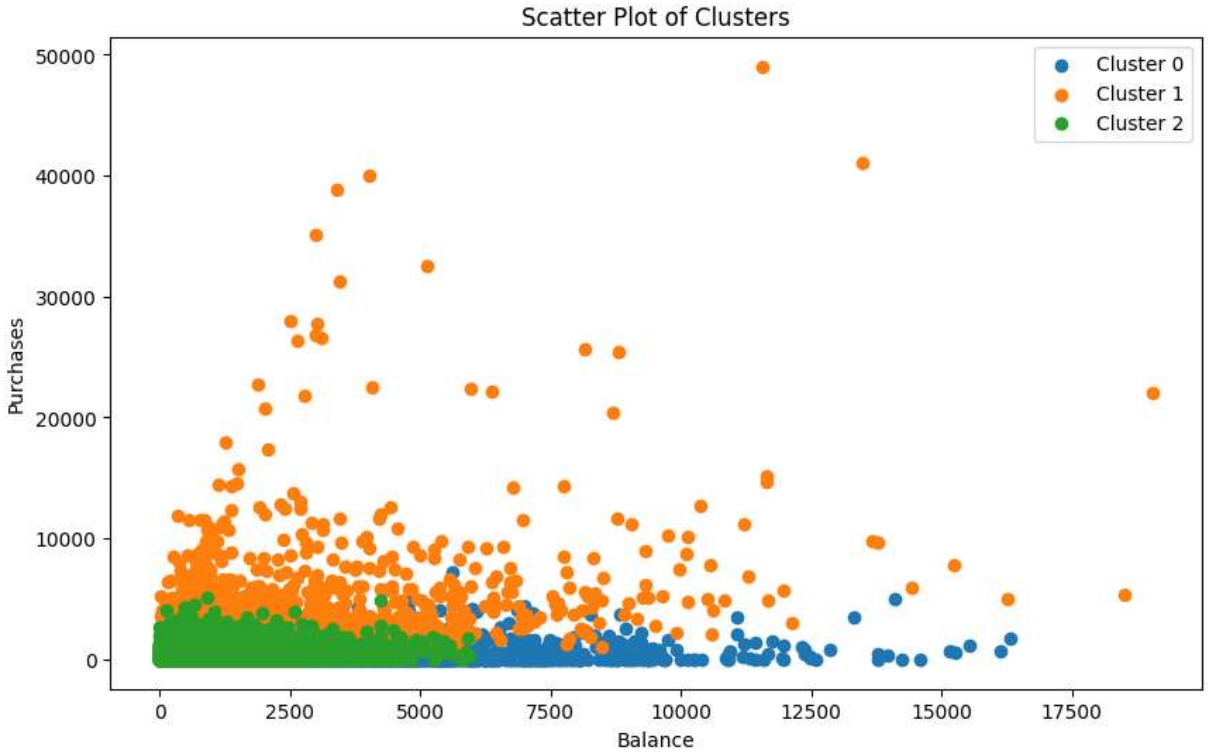
Cluster	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLME
0	3985.674014	0.958839	382.738556	248.550496	
1	2230.302719	0.981507	4264.856928	2714.253395	
2	799.671003	0.834970	505.449791	253.078771	

Visualize the clusters

In [ ]:
```
selected_clusters = [0, 1, 2]  # Replace with the clusters you want to visualize

plt.figure(figsize=(10, 6))
for cluster in selected_clusters:
    cluster_data = df[df["Cluster"] == cluster]
    plt.scatter(
        cluster_data["BALANCE"], cluster_data["PURCHASES"], label=f"Cluster {cluste
    )

plt.title("Scatter Plot of Clusters")
plt.xlabel("Balance")
plt.ylabel("Purchases")
plt.legend()
plt.show()
```



Interpret the clusters

The above plot shows the clusters of the data points. The data points are clustered into 3 clusters. The data points in the cluster 0 are the ones which have low income and low spending score. The data points in the cluster 1 are the ones which have high income and high spending score. The data points in the cluster 2 are the ones which have medium income and medium spending score.