Using Credit Card Dataset develop a customer segmentation using KMeans to define marketing strategy.

The Dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months.

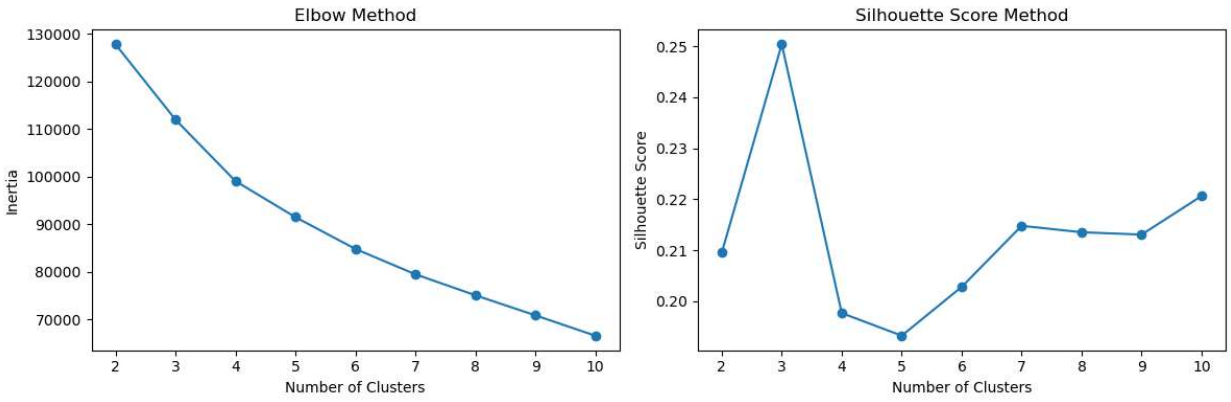
The file is at a customer level with 18 behavioral variables.

Following is the Data Dictionary for Credit Card dataset :-

CUST_ID : Identification of Credit Card holder (Categorical) BALANCE : Balance amount left in their account to make purchases (BALANCE_FREQUENCY : How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated) PURCHASES : Amount of purchases made from account ONEOFF_PURCHASES : Maximum purchase amount done in one-go INSTALLMENTS_PURCHASES : Amount of purchase done in installment CASH_ADVANCE : Cash in advance given by the user PURCHASES_FREQUENCY : How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased) ONEOFFPURCHASESFREQUENCY : How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased) PURCHASESINSTALLMENTSFREQUENCY : How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done) CASHADVANCEFREQUENCY : How frequently the cash in advance being paid CASHADVANCETRX : Number of Transactions made with "Cash in Advanced" PURCHASES_TRX : Numbe of purchase transactions made CREDIT_LIMIT : Limit of Credit Card for user PAYMENTS : Amount of Payment done by user MINIMUM_PAYMENTS : Minimum amount of payments made by user PRCFULLPAYMENT : Percent of full payment paid by user TENURE : Tenure of credit card service for user

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import warnings
warnings.filterwarnings("ignore")

# Load the dataset
data = pd.read_csv("desktop/python/CC_GENERAL.csv")

# Drop categorical columns (CUST_ID) as they won't be used in clustering
data = data.drop("CUST_ID", axis=1)

# Handling missing values
data = data.fillna(0)  # Replace missing values with 0

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

# Finding the optimal number of clusters using the Elbow Method
```

```
inertia_values = []
silhouette_scores = []

for n_clusters in range(2, 11):
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(scaled_data)
    inertia_values.append(kmeans.inertia_)
    if n_clusters > 1:
        silhouette_scores.append(silhouette_score(scaled_data, kmeans.labels_))

# Plotting the Elbow Method and Silhouette Score
plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)
plt.plot(range(2, 11), inertia_values, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method')

plt.subplot(1, 2, 2)
plt.plot(range(2, 11), silhouette_scores, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score Method')

plt.tight_layout()
plt.show()

# Based on the plots, let's choose the number of clusters (e.g., 4)
num_clusters = 4

# Perform KMeans clustering
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(scaled_data)

# Add cluster labels to the original data
data['Cluster'] = kmeans.labels_

# Analyzing the clusters
cluster_summary = data.groupby('Cluster').mean()

# Print cluster summary
print(cluster_summary)
```



Cluster	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES
0	1012.658327	0.789924	270.041785	209.937299
1	894.907458	0.934734	1236.178934	593.974874
2	3551.153761	0.986879	7681.620098	5095.878826
3	4602.449658	0.968389	501.862982	320.188797

Cluster	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY
0	60.371441	596.509903	0.170145
1	642.478274	210.570626	0.885165
2	2587.208264	653.638891	0.946418
3	181.759123	4521.509581	0.287832

Cluster	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY
0	0.086301	0.080558
1	0.297070	0.711842
2	0.739031	0.788060
3	0.138911	0.185671

Cluster	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX
0	0.114846	2.125471	2.903193
1	0.042573	0.790021	22.091773
2	0.071290	2.085575	89.359413
3	0.484792	14.294904	7.665831

Cluster	CREDIT_LIMIT	PAYMENTS	MINIMUM_PAYMENTS	PRC_FULL_PAYMENT
0	3277.886179	974.260054	535.522656	0.077981
1	4213.207678	1332.194205	633.740218	0.269258
2	9696.943765	7288.739497	1970.476256	0.286707
3	7546.160857	3484.054216	2000.543192	0.034888

Cluster	TENURE
0	11.446568
1	11.594595
2	11.951100
3	11.386800