**Ex 4: K Means algorithm**

**Apply k-Means algorithm to cluster a set of data stored in a CSV file and comment on the quality of clustering.**

LOKESH.K

240711008

**Aim:**

To build anApply k-Means algorithm to cluster a set of data stored in a CSV file and comment on the quality of clustering.

.

**Algorithm:**

**Step 1**: Import necessary libraries (pandas, numpy, matplotlib, sklearn) and load the dataset.

**Step 2**: Extract the relevant features — Annual Income and Spending Score — for clustering.

**Step 3**: Apply the Elbow Method: run K-Means for different values of k (1 to 10) and plot the WCSS values to find the optimal number of clusters.

**Step 4**: Perform K-Means clustering with the selected number of clusters (optimal k) and assign a cluster label to each customer.

**Step 5**: Visualize the clusters using a scatter plot and evaluate the clustering performance using the Silhouette Score.

**Program:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score

# Load the dataset

file\_path = "Mall\_Customers.csv"

df = pd.read\_csv(file\_path)

# Selecting relevant features

X = df[['Annual Income (k$)', 'Spending Score (1-100)']].values

# Finding the optimal number of clusters using the Elbow Method

wcss = []  # Within-cluster sum of squares

for k in range(1, 11):

    kmeans = KMeans(n\_clusters=k, random\_state=42, n\_init=10)

    kmeans.fit(X)

    wcss.append(kmeans.inertia\_)

# Plot the Elbow Method graph

plt.figure(figsize=(8, 5))

plt.plot(range(1, 11), wcss, marker='o', linestyle='--')

plt.xlabel('Number of Clusters')

plt.ylabel('WCSS (Within-Cluster Sum of Squares)')

plt.title('Elbow Method for Optimal k')

plt.show()

# Based on the Elbow Method, selecting the optimal k

optimal\_k = 5

# Applying k-Means clustering

kmeans = KMeans(n\_clusters=optimal\_k, random\_state=42, n\_init=10)

df['Cluster'] = kmeans.fit\_predict(X)

# Visualizing the clusters

plt.figure(figsize=(8, 6))

for cluster in range(optimal\_k):

    plt.scatter(X[df['Cluster'] == cluster, 0], X[df['Cluster'] == cluster, 1], label=f'Cluster {cluster}')

# Plot centroids

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s=200, c='black', marker='X', label='Centroids')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.title('Customer Clusters based on Annual Income and Spending Score')

plt.legend()

plt.show()

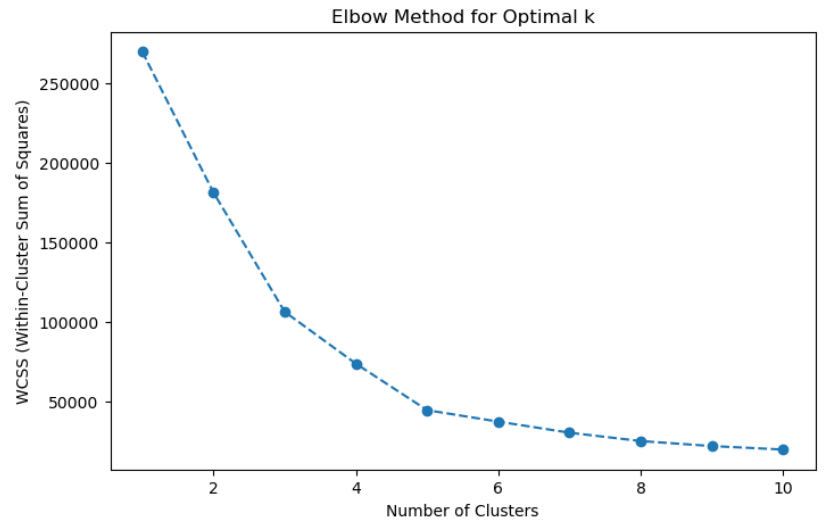
# Compute silhouette score

silhouette\_avg = silhouette\_score(X, df['Cluster'])

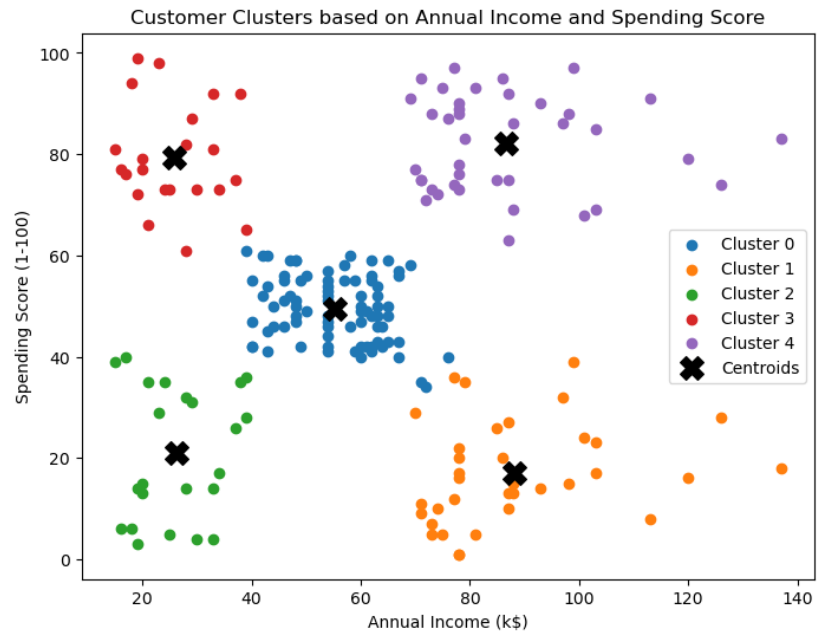
print(f'Silhouette Score: {silhouette\_avg:.3f}')

**OUTPUT**

**Elbow Method Graph:**



**Clustered Data Visualization:**



**Silhouette Score:**

Silhouette Score**: 0.554**