EX.No: 9	Application of K-Means Clustering for Customer Segmentation using Scikit-learn in Python
DATE:	

AIM:

To implement and analyze K-Means clustering algorithm for segmenting customers based on their attributes (such as income and spending score) using **Scikit-learn** in Python.

K-MEANS CLUSTERING:

- **Clustering** is an unsupervised learning method used to group similar data points together without prior knowledge of labels.
- **K-Means** is one of the most popular clustering algorithms that:
  - Selects *K* cluster centers (centroids).
  - Assigns each data point to the nearest centroid.
  - Updates centroids based on cluster membership.
  - Repeats until centroids stabilize.
- It is widely used in **customer segmentation**, where businesses divide customers into groups for targeted marketing, recommendations, and personalized services.

MATHEMATICAL OBJECTIVE FUNCTION:

Minimize the *within-cluster sum of squares (WCSS)*:

Where K=number of clusters, = data point, = Centroid of cluster

ALGORITHM:

- **Step 1:** Choose the number of clusters K.
- **Step 2:** Initialize cluster centroids randomly
- **Step 3:** Assign each data point to the nearest centroid.
- **Step 4:** Update centroids by calculating the mean of points in each cluster.
- **Step 5:** Repeat steps 3–4 until centroids do not change significantly(convergence).

SOFTWARE & LIBRARIES REQUIRED:

- ✓ Python 3.x
- ✓ Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn

PROGRAM:1

Question

A retail store wants to segment its customers based on **Annual Income** and **Spending Score** using the **K-Means clustering algorithm**.

1. Manually create a dataset in Python (without using any CSV file). The dataset should contain the following 10 rows of data:

CustomerID	AnnualIncome	SpendingScore
1	15	39

2	16	81
3	17	6
4	18	77
5	19	40
6	20	76
7	21	6
8	22	94
9	23	3
10	24	72

2. Store the data in a **pandas DataFrame**.
3. Apply the **Elbow Method** to determine the optimal number of clusters.
4. Implement **K-Means clustering** with the chosen number of clusters.
5. Visualize the clusters using a **scatter plot**, labeling each cluster with different colors.

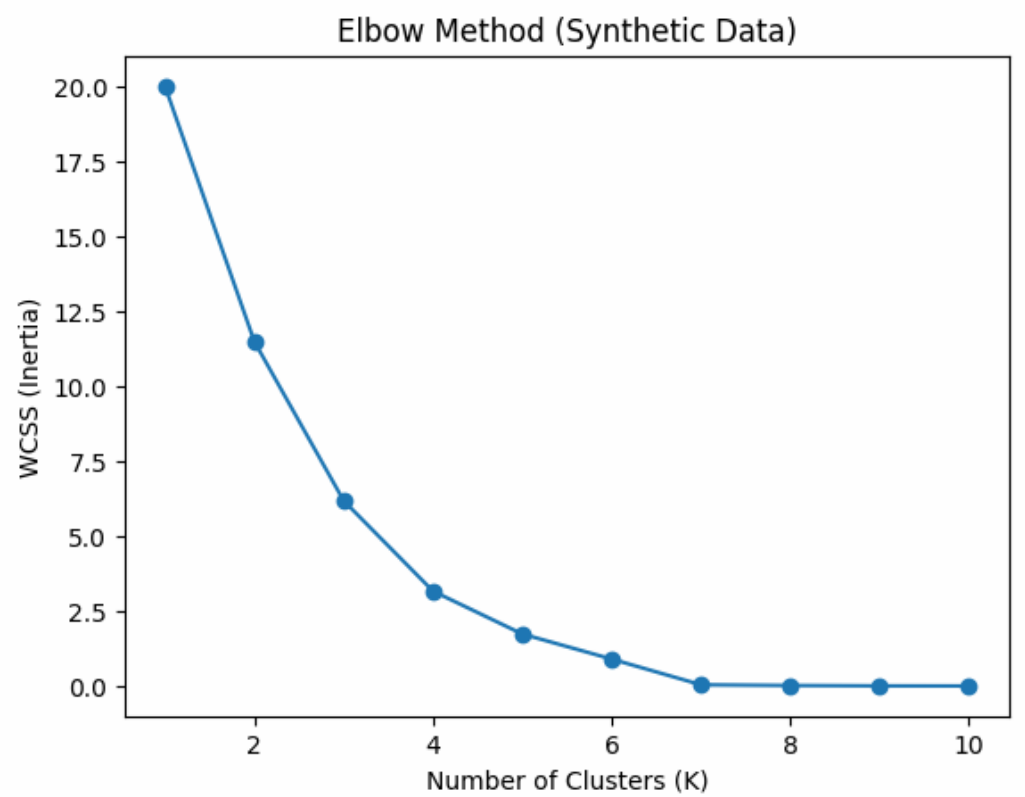
PYTHON CODING

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Step 1: Manually create dataset

data = {

   'CustomerID': [1,2,3,4,5,6,7,8,9,10],

   'Annual Income (k$)': [15, 16, 17, 18, 45, 46, 47, 80, 82, 85],

   'Spending Score (1-100)': [39, 81, 6, 77, 40, 42, 87, 20, 79, 17]

}

df = pd.DataFrame(data)

print(df)


# Step 2: Select features
```

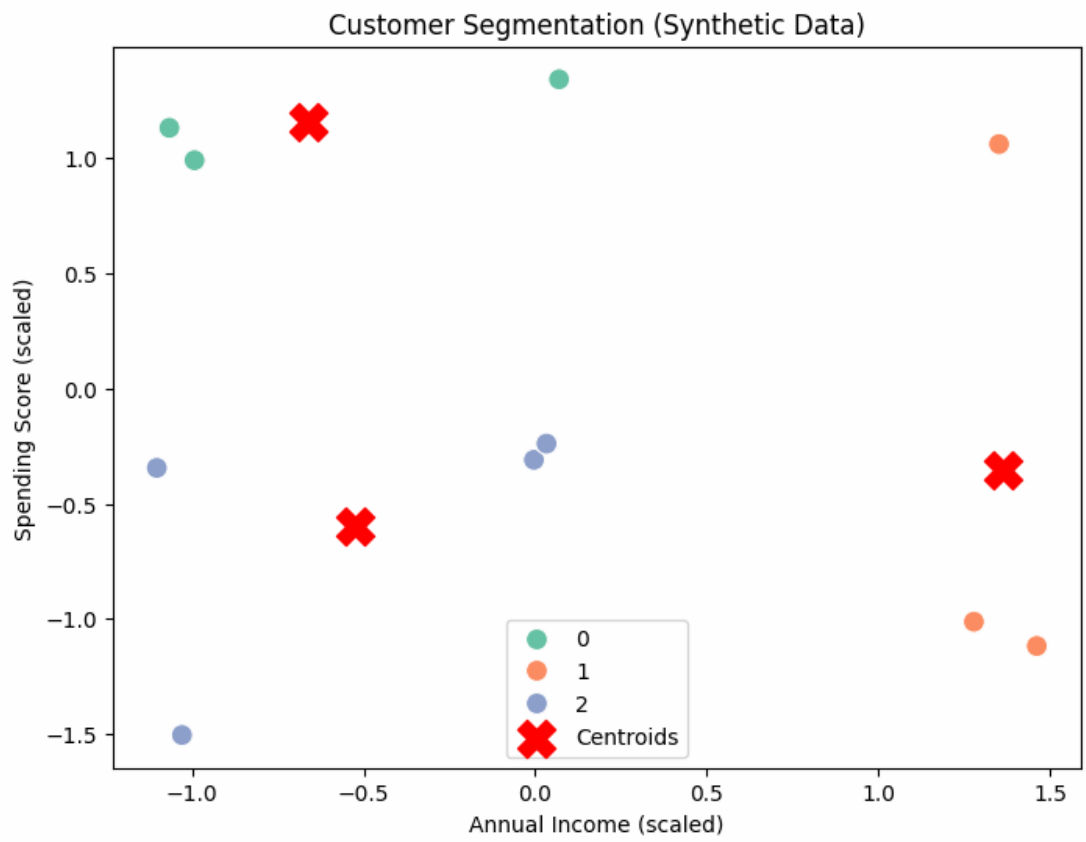
```
X = df[['Annual Income (k$)', 'Spending Score (1-100)']].values
# Step 3: Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Step 4: Elbow method
inertia = []
for k in range(1, 11):
    model = KMeans(n_clusters=k, random_state=42)
    model.fit(X_scaled)
    inertia.append(model.inertia_)

plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel("Number of Clusters (K)")
plt.ylabel("WCSS (Inertia)")
plt.title("Elbow Method (Synthetic Data)")
plt.show()
# Step 5: Apply KMeans (e.g., k=3)
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X_scaled)
df['Cluster'] = clusters
print(df)
# Step 6: Visualize clusters
plt.figure(figsize=(8,6))
sns.scatterplot(x=X_scaled[:,0], y=X_scaled[:,1], hue=clusters, palette='Set2', s=100)
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
            s=300, c='red', marker='X', label='Centroids')
plt.xlabel("Annual Income (scaled)")
plt.ylabel("Spending Score (scaled)")
plt.title("Customer Segmentation (Synthetic Data)")
plt.legend()
plt.show()
```

OUT PUT

	CustomerID	Annual Income (k$)	Spending Score (1-100)
0	1	15	39
1	2	16	81
2	3	17	6
3	4	18	77
4	5	45	40
5	6	46	42
6	7	47	87
7	8	80	20
8	9	82	79
9	10	85	17



	CustomerID	Annual Income (k$)	Spending Score (1-100)	Cluster
0	1	15	39	2
1	2	16	81	0
2	3	17	6	2
3	4	18	77	0
4	5	45	40	2
5	6	46	42	2
6	7	47	87	0
7	8	80	20	1
8	9	82	79	1
9	10	85	17	1



PROGRAM:2

Note:

1. Place the **customers.csv** file in the same folder as your Python code.
2. Run the code in **Spyder / Jupyter Notebook / VS Code**.
3. It will show:
    - Dataset preview
    - Elbow curve (to choose clusters)
    - Segmentation scatter plot

QUESTION

A retail store wants to segment its customers based on **Annual Income** and **Spending Score** using the **K-Means clustering algorithm**.

1. Create a CSV file named customers.csv with the following data:

CustomerID	AnnualIncome	SpendingScore
1	15	39
2	16	81
3	17	6
4	18	77
5	19	40

6	20	76
7	21	6
8	22	94
9	23	3
10	24	72

2. Load the dataset from the CSV file into Python using **pandas**.
3. Apply the **Elbow Method** to determine the optimal number of clusters.
4. Implement **K-Means clustering** with the chosen number of clusters.
5. Visualize the clusters using a **scatter plot**, labeling each cluster with different colors.

PYTHON CODING

```
# K-Means Customer Segmentation using CSV file

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

# Step 1: Load the CSV file

df = pd.read_csv("customers.csv")

print("Dataset Preview:")

print(df.head())

# Step 2: Select features (Age, Annual Income, Spending Score)

X = df[['Age', 'AnnualIncome', 'SpendingScore']]

# Step 3: Find the optimal number of clusters using Elbow Method

wcss = []

for i in range(1, 11):

    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)

    kmeans.fit(X)

    wcss.append(kmeans.inertia_)

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

plt.title('Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')
```

```
plt.show()
# Step 4: Train K-Means with optimal number of clusters (say 4 or 5)
kmeans = KMeans(n_clusters=4, init='k-means++', random_state=42)
df['Cluster'] = kmeans.fit_predict(X)
print("\nClustered Data:")
print(df.head())
# Step 5: Visualize the clusters
sns.scatterplot(data=df, x="AnnualIncome", y="SpendingScore",
                hue="Cluster", palette="deep", s=100)
plt.title("Customer Segmentation using K-Means")
plt.show()
```

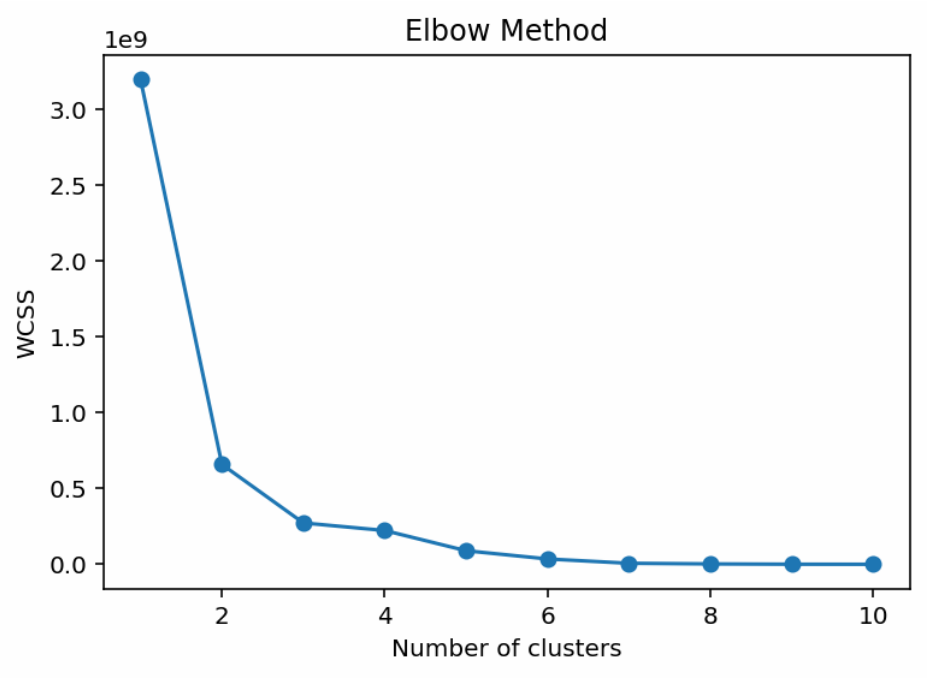
OUT PUT

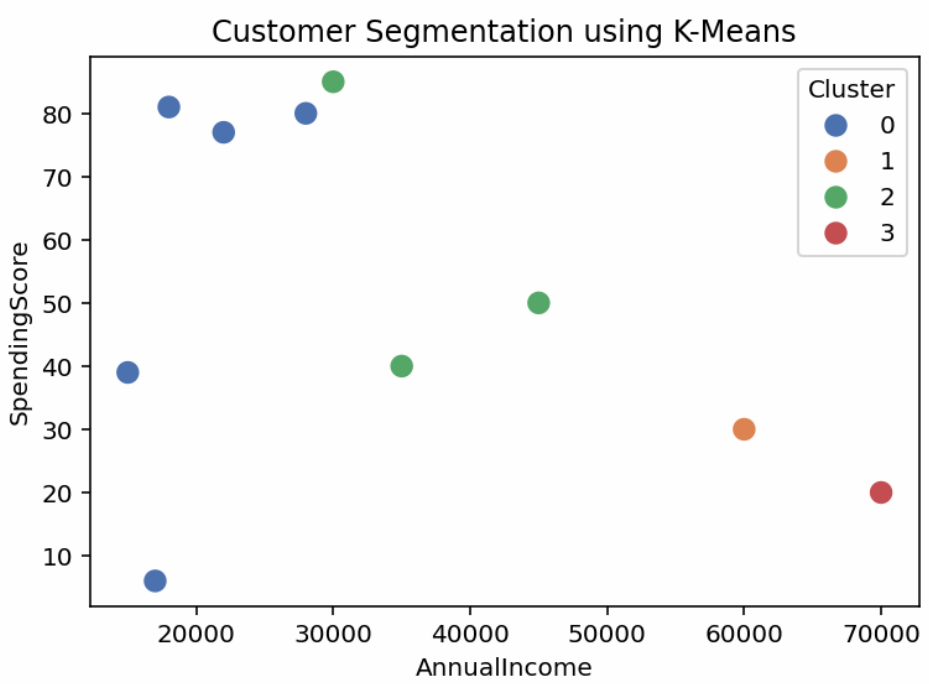
Dataset Preview:

	CustomerID	Age	AnnualIncome	SpendingScore
0	1	19	15000	39
1	2	21	18000	81
2	3	20	17000	6
3	4	23	22000	77
4	5	31	35000	40

Clustered Data:

	CustomerID	Age	AnnualIncome	SpendingScore	Cluster
0	1	19	15000	39	0
1	2	21	18000	81	0
2	3	20	17000	6	0
3	4	23	22000	77	0
4	5	31	35000	40	2





Result:

The experiment successfully applied **K-Means clustering** to segment customers into groups based on income and spending behavior. Visualization confirmed the effectiveness of clustering, and the elbow method helped determine the optimal number of clusters.

EXERCISE:

1. A shopping mall collects customer information containing **Age** and **Annual Income**. The management wants to identify different customer groups for targeted marketing campaigns.

The following dataset contains information of 10 customers:

CustomerID	Age	AnnualIncome
1	19	15000
2	21	18000
3	20	17000
4	23	22000
5	31	35000
6	35	40000
7	40	45000
8	52	60000
9	58	65000
10	63	70000

Question:

1. Load the above dataset into a pandas DataFrame (you can enter manually or save as CSV and load).
2. Apply the **Elbow Method** to determine the optimal number of clusters.
3. Perform **K-Means clustering** to group customers.
4. Visualize the clusters in a scatter plot (Age vs Annual Income).
5. Interpret the results (e.g., young-low income, middle-aged-high income, etc.).

2. A retail chain wants to classify customers into lifestyle groups using three features: **Age**, **Annual Income**, and **Spending Score**.

The following dataset contains information of 10 customers:

CustomerID	Age	AnnualIncome	SpendingScore
1	19	15000	39
2	21	18000	81
3	20	17000	6
4	23	22000	77
5	31	35000	40

6	35	40000	50
7	40	45000	60
8	52	60000	30
9	58	65000	20
10	63	70000	70

Question:

1. Load the above dataset into a pandas DataFrame (either manually or using CSV).
2. Standardize the features since they are in different ranges.
3. Use the **Elbow Method** to find the best number of clusters.
4. Perform **K-Means clustering**.
5. Visualize the clusters in a **3D scatter plot** (Age vs Income vs Spending Score).
6. Explain the characteristics of the clusters (e.g., young-high income-high spending, old-low income-low spending, etc.).