Practical No: 10

K – Means Clustering.

AIM: Implement the classification model using K-means clustering with Prediction, Test score and Confusion Matrix.

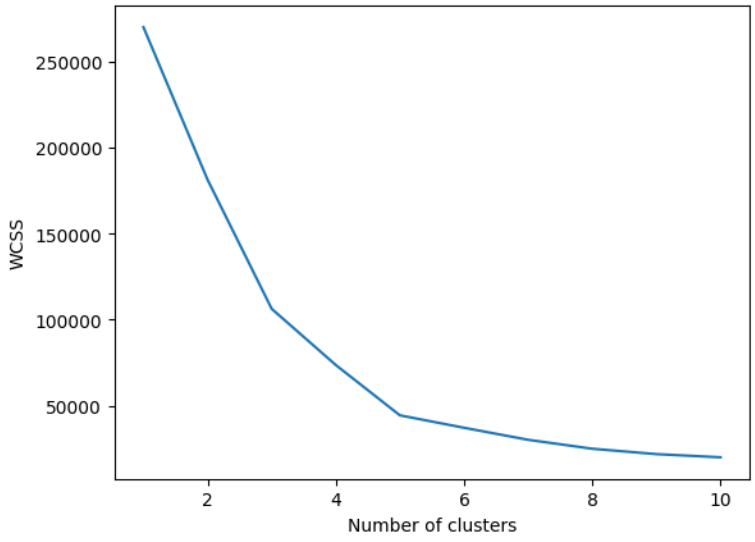
Description:

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

Code and output:

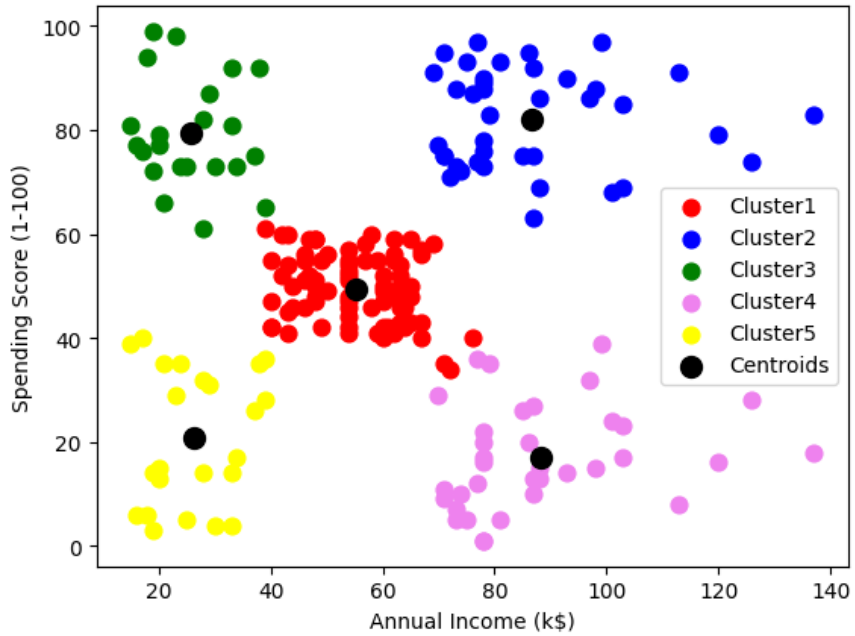
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
#Import the dataset and slice the important features
dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3,4]].values
#Find the optimal k value for clustering the data.
from sklearn.cluster import KMeans
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)
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



```
#The point at which the elbow shape is created is 5.
kmeans = KMeans(n_clusters=5,init="k-means++",random_state=42)
y_kmeans = kmeans.fit_predict(X)

plt.scatter(X[y_kmeans == 0,0], X[y_kmeans == 0,1], s = 60, c = 'red', label = 'Cluster1')
plt.scatter(X[y_kmeans == 1,0], X[y_kmeans == 1,1], s = 60, c = 'blue', label = 'Cluster2')
plt.scatter(X[y_kmeans == 2,0], X[y_kmeans == 2,1], s = 60, c = 'green', label = 'Cluster3')
plt.scatter(X[y_kmeans == 3,0], X[y_kmeans == 3,1], s = 60, c = 'violet', label = 'Cluster4')
plt.scatter(X[y_kmeans == 4,0], X[y_kmeans == 4,1], s = 60, c = 'yellow', label = 'Cluster5')
plt.scatter(kmeans.cluster_centers_[:,0],
kmeans.cluster_centers_[:,1],s=100,c='black',label='Centroids')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100')
plt.legend()
plt.show()
```



Learning:

This code snippet demonstrates the implementation of K-Means clustering on a Mall Customers dataset using Python's scikit-learn library. It first imports necessary modules and reads the dataset, selecting two key features – Annual Income and Spending Score. The optimal number of clusters (k) is determined by plotting the Within-Cluster-Sum-of-Squares (WCSS) against different k values. In this case, the elbow method suggests k=5. The K-Means algorithm is then applied, and the clusters are visualized with a scatter plot, showcasing distinct clusters based on customers' Annual Income and Spending Score. The black points represent cluster centroids, providing insights into customer segmentation for targeted business strategies.