|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Machine Learning  Practical # 10 | | |
|  |  |  |  |
|  |  |  |  |
| **Name** | Ninad Karlekar | **Roll Number** | 22306A1012 |
| **Subject/Course:** | Machine Learning | **Class** | M.Sc. IT – Sem III |
| **Topic** | K – Means Clustering | **Batch** | 1 |
|  |  |  |  |
| **Topic: 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() | | | |
|  | | | |
|  | | | |