**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Course: Data Mining & Business Intelligence Lab (ITL601)**

**B.Tech. (Information Technology) – Semester VI**

**Academic Year: 2023-24 (Even Semester)**

**Practical 7**

**Aim:** Implement and evaluate any Clustering Algorithm using Python.

**Lab Objective:** To learn how to gather and analyse large sets of data to gain useful business understanding.

**Theory**: K means clustering, assigns data points to one of the K clusters depending on their distance from the center of the clusters. It starts by randomly assigning the clusters centroid in the space. Then each data point is assigned to one of the clusters based on its distance from the centroid of the cluster. After assigning each point to one of the clusters, new cluster centroids are assigned. This process runs iteratively until it finds a good cluster. In the analysis we assume that the number of clusters is given in advance and we have to put points in one of the groups.

In some cases, K is not clearly defined, and we have to think about the optimal number of K. K Means clustering performs best when data is well separated. When data points overlap this clustering is not suitable. K Means is faster as compared to other clustering techniques. It provides strong coupling between the data points. K Means clusters do not provide clear information regarding the quality of clusters. Different initial assignments of cluster centroid may lead to different clusters. Also, K Means algorithm is sensitive to noise. It may have stuck in local minima.

The goal of clustering is to divide the population or set of data points into a number of groups so that the data points within each group are more comparable to one another and different from the data points within the other groups. It is essentially a grouping of things based on how similar and different they are to one another.

We are given a data set of items, with certain features, and values for these features (like a vector). The task is to categorize those items into groups. To achieve this, we will use the K-means algorithm, an unsupervised learning algorithm. ‘K’ in the name of the algorithm represents the number of groups/clusters we want to classify our items into. The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity, we will use the Euclidean distance as a measurement.

The algorithm works as follows:

1. First, we randomly initialize k points, called means or cluster centroids.
2. We categorize each item to its closest mean, and we update the mean’s coordinates, which are the averages of the items categorized in that cluster so far.
3. We repeat the process for a given number of iterations and at the end, we have our clusters.

The “points” mentioned above are called means because they are the mean values of the items categorized in them. To initialize these means, we have a lot of options. An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set (if for a feature x, the items have values in [0,3], we will initialize the means with values for x at [0,3]).

**Data:**

|  |  |
| --- | --- |
| a | b |
| 45 | 31 |
| 12 | 67 |
| 20 | 21 |
| 56 | 43 |
| 85 | 74 |
| 46 | 85 |
| 23 | 41 |
| 19 | 52 |
| 78 | 13 |
| 12 | 25 |
| 44 | 41 |
| 32 | 62 |
| 36 | 78 |
| 35 | 52 |
| 53 | 49 |
| 23 | 36 |
| 44 | 22 |
| 55 | 89 |
| 88 | 15 |
| 99 | 35 |
| 74 | 88 |
| 21 | 20 |
| 16 | 17 |
| 89 | 64 |
| 76 | 22 |
| 75 | 72 |
| 47 | 12 |
| 16 | 81 |
| 21 | 96 |
| 10 | 63 |
| 77 | 66 |
| 72 | 69 |
| 33 | 7 |
| 3 | 21 |
| 36 | 72 |
| 27 | 5 |
| 13 | 19 |
| 38 | 16 |
| 73 | 8 |
| 2 | 39 |
| 95 | 7 |
| 47 | 64 |
| 9 | 44 |
| 82 | 46 |
| 6 | 16 |
| 55 | 37 |
| 32 | 38 |
| 17 | 98 |
| 50 | 98 |
| 84 | 99 |

**Source Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

data = pd.read\_csv("expt7.csv")

try:

    num\_clusters = int(input("Enter the number of clusters: "))

except ValueError:

    print("Invalid input. Please enter a valid integer.")

kmeans = KMeans(n\_clusters=num\_clusters)

kmeans.fit(data)

cluster\_centers = kmeans.cluster\_centers\_

labels = kmeans.labels\_

data["Cluster"] = labels

plt.scatter(data["a"], data["b"], c=labels, cmap="rainbow")

plt.scatter(cluster\_centers[:, 0], cluster\_centers[:, 1], c="black", marker="x", s=100, label="Centroids")

plt.xlabel("a")

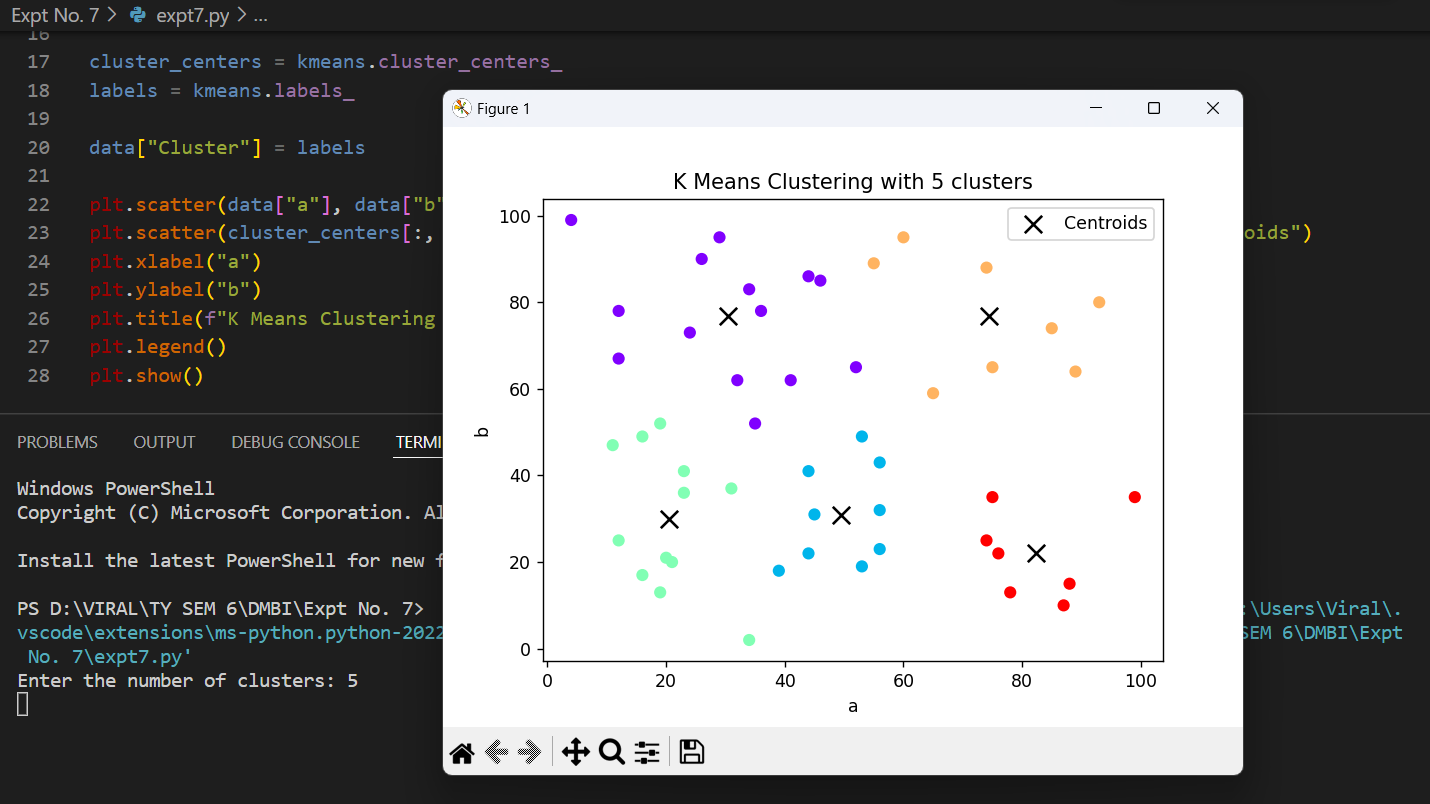
plt.ylabel("b")

plt.title(f"K Means Clustering with {num\_clusters} clusters")

plt.legend()

plt.show()

**Output:**

****

**Conclusion:** Learned about some of the key clustering algorithms used in data mining, each with its strengths and weaknesses depending on the data characteristics and clustering goals. Choosing the right algorithm often involves considering factors such as data distribution, cluster shapes, scalability, and interpretability.

**Lab Outcome:** Implement various data mining algorithms from scratch using languages like Python.

**Submitted Details -**

**Name of Student:** Parth Malviya

**Roll No.: 22**

**Date of Performance:** 05/03/2024

**Date of Submission:** 24/03/2024