**Assignment – 4**

**Problem statement:** Write a program to do following:

We have given a collection of 8 points. P1=[0.1,0.6] P2=[0.15,0.71] P3=[0.08,0.9] P4=[0.16,0.85] P5=[0.2,0.3] P6=[0.25,0.5] P7=[0.24,0.1] P8=[0.3,0.2]. Perform the k-mean clustering with initial centroids as m1=P1=Cluster#1=C1 and m2=P8=cluster#2=C2.Answer the following:

a) Which cluster does P6 belong to?

b) What is the population of a cluster around m2**?**

c) What is the updated value of m1 and m2?

**Library:**

Software used:

1. Python 3.x
2. Google Colab

Libraries and packages used:NumPy, Matplotlib, scikit-learn

**Theory:**

**Methodology:**

* K-means Clustering is a popular unsupervised machine learning algorithm used for partitioning data into distinct clusters. It groups the unlabeled dataset into different clusters. Here K defines the number of predefined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
* The algorithm aims to minimize the variance within each cluster while maximizing the variance between clusters. The process involves iteratively assigning data points to the nearest cluster centroid and updating the centroids based on the mean of the points assigned to each cluster.
* The k-means clustering algorithm mainly performs two tasks:

1. Determines the best value for K center points or centroids by an iterative process.
2. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

**Initialization:**

The algorithm starts with an initial guess for the cluster centroids. These centroids can be randomly chosen from the dataset or predefined.

**Assignment Step:**

Each data point is assigned to the nearest cluster centroid based on a distance metric, typically Euclidean distance.The distance between a data point *x* and a centroid *mi*

**Update Step:**

After all data points have been assigned to clusters, the centroids are updated by computing the mean of all data points assigned to each cluster.

**Convergence:**

Steps 2 and 3 are repeated iteratively until the cluster assignments and centroids no longer change significantly or a predefined number of iterations is reached.

**Advantages:**

1. Simplicity: K-means is straightforward to implement and easy to understand.
2. Efficiency: It is computationally efficient and scales well to large datasets.
3. Versatility: Suitable for a wide range of applications and data types.
4. Scalability: Performs well even with a large number of dimensions.
5. Interpretability: Results are easily interpretable, especially with low-dimensional data.

**Disadvantages:**

1. The choice of initial centroids can impact the final clustering results.
2. The algorithm requires specifying the number of clusters beforehand.
3. K-means assumes that clusters are spherical and of similar size.
4. Outliers can significantly affect the cluster centroids and the overall clustering outcome.
5. The algorithm's convergence to a local minimum is not guaranteed to be the global minimum.

**Applications with example:**

1. Customer Segmentation: In marketing, K-means clustering can be used to segment customers based on their purchasing behavior. For example, a retail company can cluster customers into groups such as high-value customers, frequent buyers, and occasional shoppers.
2. Anomaly Detection: In cybersecurity, K-means clustering can be utilized to detect anomalies or unusual patterns in network traffic. For example, network administrators can cluster network traffic data and identify clusters with significantly different characteristics, indicating potential security threats or anomalies.
3. Document Clustering: In natural language processing, K-means clustering can be employed to cluster similar documents together. For instance, news articles can be clustered into groups based on their topics, allowing users to explore related articles more efficiently.

**Working / Algorithm:**

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

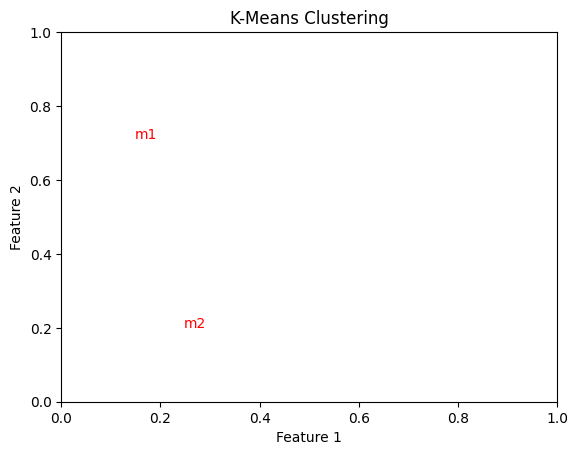
Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassigning each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready

**Diagram:**



**Conclusion:**

In conclusion, this assignment demonstrates the effectiveness of K-means clustering in partitioning data into distinct clusters based on similarity. We have explored its simplicity, efficiency, and versatility, showcasing its applicability across various domains such as customer segmentation, anomaly detection, and document clustering.