Title: - K-Means Clustering for Point Dataset with Initial Centroids and Cluster Population

Aim/Objective: - To perform K-means clustering on a set of points using initial centroids, and to determine:

1. Which cluster a given point (P6) belongs to?
2. The population of the cluster around centroid m2 (C2).
3. The updated centroid values for m1 (C1) and m2 (C2).

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]

Software Required:

- Python programming environment (Jupyter Notebook, Google Colab, or any Python IDE)
- Libraries: numpy, matplotlib, math

Hardware Required:

- 4GB RAM
- Intel i3 or higher / AMD equivalent
- GPU for accelerated computations (if using deep learning frameworks)
- 120 GB SSD

Theory:

- **K-Means Clustering**: K-means is a popular unsupervised machine learning algorithm used for clustering datasets. It partitions the data into K clusters based on similarity. The algorithm iteratively refines the centroids of clusters and assigns each data point to the closest centroid until convergence.

- **Steps**:
    1. Choose K initial centroids.
    2. Assign each data point to the nearest centroid.
    3. Recalculate the centroids by averaging the points in each cluster.
    4. Repeat steps 2-3 until the centroids stabilize.

Procedure:

1. **Dataset Preparation**: Given a collection of 8 points, and initial centroids m1=P1 and m2=P8.
2. **Cluster Assignment**: Calculate the Euclidean distance from each point to both centroids (m1 and m2) and assign each point to the nearest centroid.
3. **Update Centroids**: After all points are assigned, calculate the new centroids by averaging the points in each cluster.
4. **Reiterate**: Repeat the process of reassigning points and updating centroids until the centroids converge.
5. **Find Results**:
    - Determine which cluster P6 belongs to.
    - Calculate the population of points around centroid m2.
    - Find the updated centroid values for m1 and m2.

Observation:

Which cluster does P6 belong to?

After calculating the Euclidean distance and assigning the points to the closest centroids, P6 is assigned to Cluster #2 (C2) as it is closer to m2.

What is the population of the cluster around m2?

Cluster #2 (C2) contains the following points: P5, P6, P7, and P8, giving it a population size of 4.

What is the updated value of m1 and m2?

After recalculating the centroids, the updated centroid for m1 (C1) is [0.13, 0.625], and the updated centroid for m2 (C2) is [0.24, 0.45].

Code:

```
#Import Dataset
import math
import numpy as np
#Read Dataset
#Points Dataset
xp=[0.1,0.15,0.08,0.16,0.2,0.25,0.24,0.3]
yp=[0.6,0.71,0.9,0.85,0.3,0.5,0.1,0.2]
l=len(xp)
#Centeroid Dataset C1
C1x=0.1
C1y=0.6
#Centeroid Dataset C1
C2x=0.3
C2y=0.2
#Create Cluster m1
m1x=[]
m1y=[]
m1x.append(0.1)
m1y.append(0.6)
#Create Cluster m2
m2x=[]
m2y=[]
m2x.append(0.3)
m2y.append(0.2)

#Find New Centeroid & Update Population/Insert Point into respective cluster
for i in range(1,l-1):
    dist1=math.sqrt((xp[i]-C1x)**2 + (yp[i]-C1y)**2)
    dist2=math.sqrt((xp[i]-C2x)**2 + (yp[i]-C2y)**2)
#Update Population
    if(dist1<dist2):
        m1x.append(xp[i])
        m1y.append(yp[i])
    else:
        m2x.append(xp[i])
        m2y.append(yp[i])

#Calulating lenth of new Population in m1 and m2
l1=len(m1x)
l2=len(m2x)

#Finding P6 belong to which cluster?
xfind=0.25
yfind=0.5
for i in range (0,l1):
    if(xfind==m1x[i]) and (yfind==m1y[i]):
        print("P6 Belong to m1")
```

```
for i in range (0,l2):
    if(xfind==m2x[i]) and (yfind==m2y[i]):
        print("P6 Belong to m2")

#Showing Population Size of M1 & M2
for i in range(0,l1):
    print("P",i+1," = [",m1x[i],",",m1y[i],"]")
print("M1 Population Size = ",l1)
for i in range(0,l2):
    print("P",i+1," = [",m2x[i],",",m2y[i],"]")
print("M2 Population Size = ",l2)

#Finding Updated Centeroid Value of C1 and C2
print("Centeroid Value of C1 = [",np.mean(m1x),",",np.mean(m1y),"]")
print("Centeroid Value of C2 = [",np.mean(m2x),",",np.mean(m2y),"]")
```

Output:

```
P6 Belong to m1
P 1  = [ 0.1 , 0.6 ]
P 2  = [ 0.15 , 0.71 ]
P 3  = [ 0.08 , 0.9 ]
P 4  = [ 0.16 , 0.85 ]
P 5  = [ 0.25 , 0.5 ]
M1 Population Size =  5
P 1  = [ 0.3 , 0.2 ]
P 2  = [ 0.2 , 0.3 ]
P 3  = [ 0.24 , 0.1 ]
M2 Population Size =  3
Centeroid Value of C1 = [ 0.148 , 0.712 ]
Centeroid Value of C2 = [ 0.24666666666666667 , 0.19999999999999998 ]
```

Conclusion:

The K-means clustering algorithm successfully categorized the given points into two clusters with their respective centroids. After updating the centroids based on the assigned points, we determined the population of the clusters and updated centroid values. This experiment helped in understanding how K-means clustering works and how centroids evolve based on the dataset.