Name:	
Roll No:	
Class/Sem:	TE/V
Experiment No.:	8
Title:	Implementation of K-means Clustering algorithm.
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: To Implement K-Means clustering algorithm.

Objective:- Understand the working of K-Means algorithm and it's implemention using python/Java.

Theory:

K-Means Clustering is an unsupervised learning algorithm which groups the unlabeled dataset into different clusters. In this approach, the data objects ('n') are classified into 'k' number of clusters in which each observation belongs to the cluster with nearest mean.

Algorithm:

Input-

K:-number of clusters

D:- data set containing n objects

OutputA set of k clusters

Method-

Given k, the k-means algorithm is implemented in 5 steps:

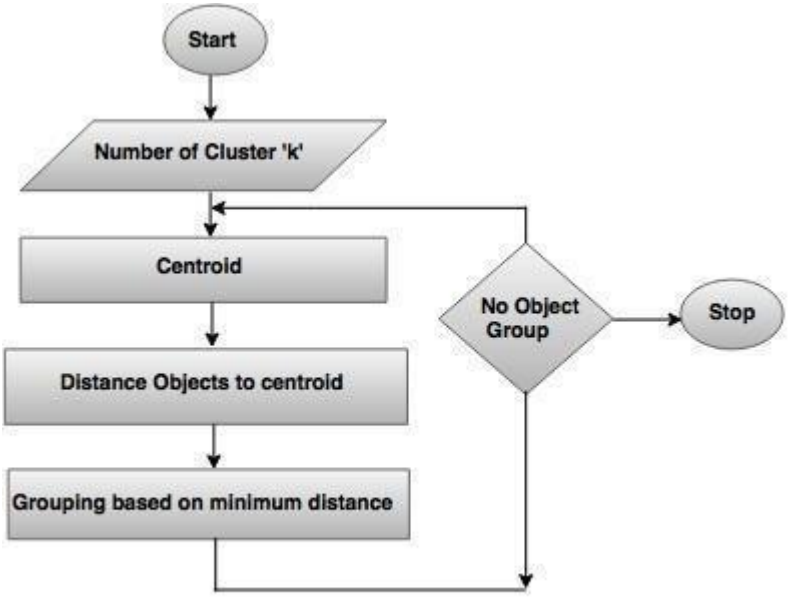
Step 1: Arbitrarily choose k objects from D as the initial cluster centers.

Step 2: Find the distance from each and every object in the dataset with respect to cluster centers.

Step 3: Assign each object to the cluster with the nearest seed point based on the mean value of the objects in the cluster.

Step 4: Update the cluster means i,e calculate the mean value of the objects for each cluster.

Step 5: Repeat the procedure, until there is no change.



K- mean Algorithm

Example: Cluster the following data into two clusters {2, 4, 10, 12, 3, 20, 30, 11, 25}

Step 1: Randomly assign the means: m1 = 3, m2 = 4

Step 2: Group the numbers close to mean m1 = 3 are grouped into cluster k1 and m2 = 4 are grouped into cluster k2

Step 3: k1 = {2, 3}, k2 = {4, 10, 12, 20, 30, 11, 25}, m1= 2.5, m2 = 16

Step 4: k2 = {2, 3, 4}, k2 = {10, 12, 20, 30, 11, 25}, m1= 3, m2 = 18

Step 5: k1 = {2, 3, 4, 10}, k2= {12, 20, 30, 11, 25}, m1= 4.75, m2 = 19.6

Step 6: k1 = {2, 3, 4, 10, 11, 12}, k2 = {20, 30, 25}, m1 = 7, m2 = 25

Step 7: k1 = {2, 3, 4, 10, 11, 12}, k2 = {20, 30, 25}, m1 = 7, m2 = 25 Step 8: Stop. The clusters in step 6 and 7 are same.

Final answer: k1 = {2, 3, 4, 10, 11, 12} and k2 = {20, 30, 25} **Program:**

```
x = int(input("Enter length: "))
dataset = [0] * x for i in range(x):
    dataset[i] = int(input("Enter dataset: "))

cluster1 = [] cluster2 = []


n = len(dataset) m1
= dataset[0] m2 =
dataset[n - 1]


print("DATASET:", dataset)
print("Initial Mean m1:", m1)
print("Initial Mean m2:", m2)


while True:
    p = [] q =
    []
     for i in range(n): g =
         abs(m1 - dataset[i]) h
```

```
= abs(m2 - dataset[i])

if g < h:

   p.append(dataset[i])

else:

   q.append(dataset[i]) new_m1 = sum(p) /

len(p) if len(p) > 0 else 0 new_m2 = sum(q)

/ len(q) if len(q) > 0 else 0


if m1 == new_m1 and m2 ==

new_m2: break m1 = new_m1 m2 =

new_m2


print("CLUSTER 1 p:", p) print("CLUSTER

2 q:", q)
```

Output:

```
= RESTART: C:/Users/student/Downloads/python8.py
Enter length: 9
Enter dataset: 2
Enter dataset: 3
Enter dataset: 4
Enter dataset: 10
Enter dataset: 11
Enter dataset: 12
Enter dataset: 20
Enter dataset: 25
Enter dataset: 30
DATASET: [2, 3, 4, 10, 11, 12, 20, 25, 30]
Initial Mean m1: 2
Initial Mean m2: 30
Final Clusters are
CLUSTER 1 p: [2, 3, 4, 10, 11, 12]
CLUSTER 2 q: [20, 25, 30]
```

Conclusion:

Implementing the K-means clustering algorithm is a powerful method for grouping data points into clusters, enabling data analysts and data scientists to uncover patterns, segment data, and make data-driven decisions. It has widespread applications and is essential for tasks such as customer segmentation, image processing, and anomaly detection. Careful consideration of the number of clusters and proper interpretation of results are crucial for the success of K-means clustering in practice.