**Aim:** To Study and Implement K‐Means algorithm

**Objective:-** Understand the working of K‐Means algorithm and it’s implemention using python.

**Theory:**

In statistics and machine learning, k‐means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

Input

K:-number of clusters

D:- data set containing n objects

Output

A set of k clusters

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

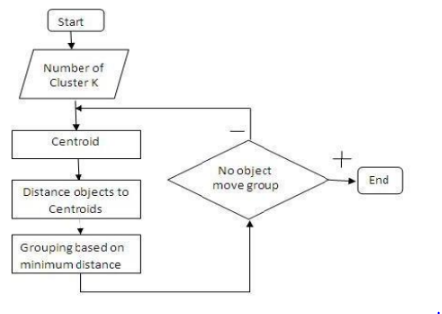
Step 1: Arbitrarily choose k objects from D as the initial cluster centers.

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 in meaning.



Example: d= {2,4,10,12,3,20,30,11,25} k =2

1. Randomly assign mean m1=3 and m2 = 4

Therefore, k1 = {2,3} Therefore, k1 = {4,10,12,20,30,11,25}

2. Randomly assign mean m1=2.5 and m2 = 16

Therefore, k1 = {2,3,4} Therefore, k1 =

{4,10,12,20,30,11,25}

3. Randomly assign mean m1=3 and m2 = 18

Therefore, k1 = {2,3,4,10} Therefore, k1 = {12,20,30,11,25}

4. Randomly assign mean m1=7 and m2 = 25

Therefore, k1 = {2,3,4,10,11,12} Therefore, k1 =

{20,30,25}

5. Randomly assign mean m1=7 and m2 = 25

Therefore, we stop as we are getting same mean

values.

6. Therefore, Final clusters are : k1 = {2,3,4,10,11,12} Therefore, k1 = {20,30,25}

**CODE:**

x = int(input("enter length : "))

dataset = [0] \* x

for i in range(x):

    dataset[i] = int(input("enter dataset"))

list1 = dataset

m = list1

print("DATASET: ", m)

n = int(len(m))

# randomly selecting mean

m1 = list1[0]

m2 = list1[n-1]

print("mean m1 :", m1)

print("mean m2 :", m2)

# first iteration

iteration = 1

p = [0]\*x  # declaring array

q = [0]\*x

for i in range(n):

    g = abs(m1-m[i])

    h = abs(m2-m[i])

    if g < h:

        p[i] = m[i]

    else:

        q[i] = m[i]

print("CLUSTER 1 p: ", p)

print("CLUSTER 2 q: ", q)

print("ITERATION NO : ", iteration)

# removing zero from clusters

q = list(filter(lambda num: num != 0, q))

p = list(filter(lambda num: num != 0, p))

print(p, q)

**OUTPUT:**

PS D:\Vartak college\SEM 5\DWM\code> py .\kmean.py

enter length : 10

enter dataset11

enter dataset12

enter dataset13

enter dataset14

enter dataset15

enter dataset12

enter dataset15

enter dataset156

enter dataset14

enter dataset13

DATASET: [11, 12, 13, 14, 15, 12, 15, 156, 14, 13]

mean m1 : 11

mean m2 : 13

CLUSTER 1 p: [11, 0, 0, 0, 0, 0, 0, 0, 0, 0]

CLUSTER 2 q: [0, 12, 13, 14, 15, 12, 15, 156, 14, 13]

ITERATION NO : 1

[11] [12, 13, 14, 15, 12, 15, 156, 14, 13]

**CONCLUSION:**

Thus, we have studied to implement and understand the working of K‐Means algorithm using python. k‐means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.