Top of Form

Bottom of Form

import math

print("\n")

print("let first column be its name of your data.(include in number of column)")

num\_column\_data=input("enter number of columns in your dataset:")

num\_rows\_data=input("enter number of rows(tuples) in your dataset:")

data=[]

for i in range(0,num\_rows\_data):

data.append([])

for j in range(0,num\_column\_data):

data[i].append([])

if j==0:

data[i][j]=raw\_input("Enter name of your data of "+str(i+1)+"tuples:")

else:

data[i][j]=input("Enter data at "+str(j)+":")

k=input("enter k (number of custers) :")

if num\_column\_data==0 or num\_rows\_data==0:

data=[["d1",185,72],["d2",170,56],["d3",168,60],["d4",179,68],["d5",182,72],["d6",188,77],["d7",180,71],["d8",180,70],["d9",183,84],["d10",180,88],["d11",180,88],["d12",180,67],["d13",177,76]]

def euclideanDistance(array,mean):

sosquare=0

for i in range(1,len(array)):

sosquare=sosquare+((array[i]-mean[1][i])\*\*2)

return [mean[0],math.sqrt(sosquare)]

def minimum(array):

if len(array)>1:

pos=array[0][0]

min=array[0][1]

for i in range(1,len(array)):

if array[i][1]<min:

pos=array[i][0]

min=array[i][1]

return [pos,min]

else:

return array[0]

def initmean(data,k):

marray=[]

for i in range(0,k):

marray.append([i,data[i]])

return marray

def updateCentoid(cen,pos,data):

newcen=[float(0) for i in range(len(data))]

for i in range(1,len(data)):

newcen[i]=(cen[pos][1][i]+float(data[i]))/2

return newcen

def kmean(data,k):

array=[]

centroid=initmean(data,k)

oldcentoid=[]

j=0

while oldcentoid!=centroid:

j=j+1

oldcentoid=centroid[:]

cluster=[]

for l in range(0,k):

cluster.append([])

for i in range(len(data)):

eu\_dis=[]

for l in range(0,k):

eu\_dis.append(euclideanDistance(data[i],centroid[l]))

centroid[minimum(eu\_dis)[0]]=[minimum(eu\_dis)[0],updateCentoid(centroid,minimum(eu\_dis)[0],data[i])]

cluster[minimum(eu\_dis)[0]].append(data[i])

for i in range(0,k):

print(str(i+1)+" cluster is "+str(cluster[i]))

kmean(data,k)

**OUTPUT**:

d50112@d50112-ThinkCentre-M720t:~/Desktop$ python kk.py

let first column be its name of your data.(include in number of column)

enter number of columns in your dataset:3

enter number of rows(tuples) in your dataset:4

Enter name of your data of 1tuples:a1

Enter data at 1:1

Enter data at 2:1

Enter name of your data of 2tuples:a2

Enter data at 1:2

Enter data at 2:1

Enter name of your data of 3tuples:a3

Enter data at 1:2

Enter data at 2:3

Enter name of your data of 4tuples:a4

Enter data at 1:4

Enter data at 2:2

enter k (number of custers) :2

1 cluster is [['a1', 1, 1], ['a2', 2, 1]]

2 cluster is [['a3', 2, 3], ['a4', 4, 2]]