**CODE:**

import random

from tabulate import tabulate

import math

import matplotlib.pyplot as plt

import numpy as np

try:

F = open("data.txt","r")

except:

print("Something went wrong while opening the file!")

try:

K = int(F.readline())

print("Number of classes: ",K)

except ValueError:

print("Can't convert to integer!")

def distance(p1,p2):

sum = 0

for m in range(len(p1)-1):

sum += (p1[m] - p2[m])\*\*2

return math.sqrt(sum)

def newCentroids(data,K):

c=[]

for i in range(len(data)):

d=[0 for x in range(len(data[i][0])-1)]

for j in range(len(data[i])):

for k in range(len(data[i][j])-1):

d[k] += data[i][j][k]

for m in range(len(d)):

d[m] /= len(data[i])

c.append(d)

return c

data = []

for line in F:

values = []

temp = list(line.split(' '))

faulty=False

for v in temp:

try:

values.append(float(v))

except ValueError:

print('Unknown character!')

faulty = True

if not faulty:

values.append(-1.0)

data.append(values)

#random.seed(2)

centroids = []

indices = random.sample(range(0,len(data)-2),K)

for i in indices:

if data[i] in centroids:

centroids.append(data[i+1])

else:

centroids.append(data[i])

print('Centroids: ',centroids)

noChange = True

changeOnce=False

itr = 0

classif=[]

for m in range(K):

classif.append([])

while(1):

print("itr: ",itr)

itr+=1

changeOnce = False

noChange = True

classif=[]

for m in range(K):

classif.append([])

for i in range(len(data)):

d=[]

for j in range(len(centroids)):

d.append(distance(data[i],centroids[j]))

ind = d.index(min(d))

if data[i][-1] == -1:

data[i][-1] = ind

if not changeOnce:

noChange = False

changeOnce = True

elif data[i][-1] != ind:

data[i][-1] = ind

if not changeOnce:

noChange = False

changeOnce = True

classif[ind].append(data[i])

for x in range(len(classif)):

print(tabulate(classif[x],headers=['x','y','class']))

if not noChange:

centroids = newCentroids(classif,K)

print("New Centroids: ")

print(centroids)

else:

break

print("Final Clusters: ")

for x in range(len(classif)):

print(tabulate(classif[x],headers=['x','y','class']))

classes = []

plt.figure(1)

for i in range(len(classif)):

classes.append(np.array(classif[i]))

x = classes[-1][:,0]

y = classes[-1][:,1]

plt.scatter(x,y)

cl = []

for i in range(K):

cl.append('Cluster'+str(i))

plt.xlabel('x1 Feature')

plt.ylabel('x2 Feature')

plt.title('K Means clustering')

plt.legend(cl)

plt.show()

**OUTPUTS:**



