#import packages

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

#create dataset using DataFrame

df=pd.DataFrame({'X':[0.1,0.15,0.08,0.16,0.2,0.25,0.24,0.3],

'y':[0.6,0.71,0.9,0.85,0.3,0.5,0.1,0.2]})

f1 = df['X'].values

f2 = df['y'].values

X = np.array(list(zip(f1, f2)))

print(X)

#centroid points

C\_x=np.array([0.1,0.3])

C\_y=np.array([0.6,0.2])

centroids=C\_x,C\_y

#plot the given points

colmap = {1: 'r', 2: 'b'}

plt.scatter(f1, f2, color='orange')

plt.show()

#for i in centroids():

plt.scatter(C\_x[0],C\_y[0], color=colmap[1])

plt.scatter(C\_x[1],C\_y[1], color=colmap[2])

plt.show()

C = np.array(list((C\_x, C\_y)), dtype=np.float32)

print (C)

#plot given elements with centroid elements

plt.scatter(f1, f2, c='#050505')

plt.scatter(C\_x[0], C\_y[0], marker='\*', s=200, c='r')

plt.scatter(C\_x[1], C\_y[1], marker='\*', s=200, c='b')

plt.show()

#import KMeans class and create object of it

from sklearn.cluster import KMeans

model=KMeans(n\_clusters=2,random\_state=0)

model.fit(X)

labels=model.labels\_

print(labels)

#using labels find population around centroid

count=0

for i in range(len(labels)):

if (labels[i]==1):

count=count+1

print('No of population around cluster 2:',count-1)

#Find new centroids

new\_centroids = model.cluster\_centers\_

print('Previous value of m1 and m2 is:')

print('M1==',centroids[0])

print('M1==',centroids[1])

print('updated value of m1 and m2 is:')

print('M1==',new\_centroids[0])

print('M1==',new\_centroids[1])