import math

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

import numpy as numpy

Data= pd.read\_table("bank full.csv",sep = None,engine="python")

cols= ["age","balance","day","duration","campaign","pdays","previous"]

Data.head()

data\_encode=Data.drop(cols, axis= 1)

data\_encode= data\_encode.apply(LabelEncoder().fit\_transform)

data\_rest= Data[cols]

Data= pd.concat([data\_rest,data\_encode], axis= 1)

data\_train, data\_test= train\_test\_split(Data, test\_size= 0.33, random\_state= 4)

X\_train= data\_train.drop("y", axis= 1)

Y\_train= data\_train["y"]

X\_test= data\_test.drop("y", axis=1)

Y\_test= data\_test["y"]

scaler= StandardScaler()

scaler.fit(X\_train)

X\_train= scaler.transform(X\_train)

X\_test= scaler.transform(X\_test)

K\_cent= 8

km= KMeans(n\_clusters= K\_cent, max\_iter= 100)

km.fit(X\_train)

cent= km.cluster\_centers\_

max=0

for i in range(K\_cent):

  for j in range(K\_cent):

    d= numpy.linalg.norm(cent[i]-cent[j])

    if(d> max):

      max= d

d= max

sigma= d/math.sqrt(2\*K\_cent)

GTG= numpy.dot(G.T,G)

GTG\_inv= numpy.linalg.inv(GTG)

fac= numpy.dot(GTG\_inv,G.T)

W= numpy.dot(fac,Y\_train)

row= X\_test.shape[0]

column= K\_cent

G\_test= numpy.empty((row,column), dtype= float)

for i in range(row):

  for j in range(column):

    dist= numpy.linalg.norm(X\_test[i]-cent[j])

    G\_test[i][j]= math.exp(-math.pow(dist,2)/math.pow(2\*sigma,2))

prediction= numpy.dot(G\_test,W)

prediction= 0.5\*(numpy.sign(prediction-0.5)+1)

score= accuracy\_score(prediction,Y\_test)

print(score.mean())

#Output

0.8876675603217158