from google.colab import files

import pandas as pd

import numpy as np, numpy.random

from scipy.spatial import distance

X = pd.read\_csv('/content/data.csv')

X=np.array(X)

FR = X[:,0]

YEAR = X[:,1]

CO = X[:,2]

FR=np.array(FR)

YEAR=np.array(YEAR)

CO=np.array(CO)

X=np.array(X)

k = 30

p = 5

# Print the number of data and dimension

n = len(X)

d = len(X[0])

addZeros = np.zeros((n, 1))

X = np.append(X, addZeros, axis=1)

print("The FCM algorithm: \n")

print("The training data: \n", X)

print("\nTotal number of data: ",n)

print("Total number of features: ",d)

print("Total number of Clusters: ",k)

# Create an empty array of centers

C = np.zeros((k,d+1))

#print(C)

# Randomly initialize the weight matrix

weight = np.random.dirichlet(np.ones(k),size=n)

print("\nThe initial weight: \n", np.round(weight,2))

for it in range(3): # Total number of iterations

    # Compute centroid

    for j in range(k):

        denoSum = sum(np.power(weight[:,j],2))

        sumMM =0

        for i in range(n):

            mm = np.multiply(np.power(weight[i,j],p),X[i,:])

            sumMM +=mm

        cc = sumMM/denoSum

        C[j] = np.reshape(cc,d+1)

    #print("\nUpdating the fuzzy pseudo partition")

    for i in range(n):

        denoSumNext = 0

        for j in range(k):

             denoSumNext += np.power(1/distance.euclidean(C[j,0:d], X[i,0:d]),1/(p-1))

        for j in range(k):

            w = np.power((1/distance.euclidean(C[j,0:d], X[i,0:d])),1/(p-1))/denoSumNext

            weight[i,j] = w

print("\nThe final weights: \n", np.round(weight,2))

for i in range(n):

    cNumber = np.where(weight[i] == np.amax(weight[i]))

    X[i,d] = cNumber[0]

print("\nThe data with cluster number: \n", X)

SSE = 0

for j in range(k):

    for i in range(n):

        SSE += np.power(weight[i,j],p)\*distance.euclidean(C[j,0:d], X[i,0:d])

print("\nSSE: ",np.round(SSE,4))