**Practical No. 04**

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pylab as plt

from sklearn.cluster import KMeans

X = np.random.uniform(0,1,50)

Y= np.random.uniform(0,1,50)

df\_xy = pd.DataFrame(columns=["X","Y"])

df\_xy.X = X

df\_xy.Y = Y

df\_xy.plot(x="X" , y="Y", kind = "scatter")

model1 = KMeans(n\_clusters = 3).fit(df\_xy)

df\_xy.plot(x = "X", y = "Y", c = model1.labels\_, kind="scatter", s = 10, cmap = plt.cm.coolwarm)

univ1 = pd.read\_excel("C:\\Users\\CSE-09\\Downloads\\University\_Clustering.xlsx")

univ1.describe()

univ = univ1.drop(["State"], axis=1)

#normalization function

def norm\_func(i):

    x = (i-i.min()) / (i.max()-i.min())

    return (x)

#normalization data frame (considering the numerical part of data)

df\_norm = norm\_func(univ.iloc[:, 1:])

#elbow curve

TWSS = []

k = list(range(2, 9))

for i in k:

    kmeans = KMeans(n\_clusters = i)

    kmeans.fit(df\_norm)

    TWSS.append(kmeans.inertia\_)

TWSS

#scree plot

plt.plot(k, TWSS, 'ro-');plt.xlabel("No\_of\_Clusters");plt.ylabel("total\_within\_SS")

model1.labels\_

mb = pd.Series(model1.labels\_)

univ['Clust'] = mb

univ.head()

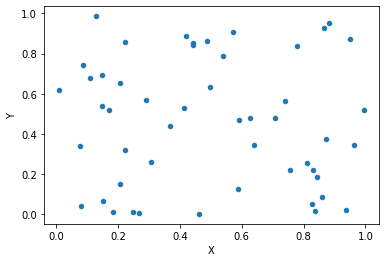
df\_norm.head()

univ = univ.iloc[:,[7,0,1,2,3,4,5,6]]

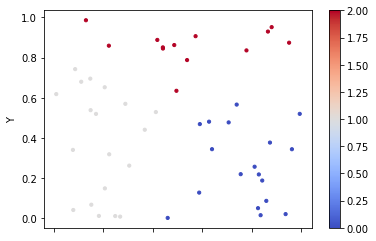
univ.head()

univ.iloc[:, 2:8].groupby(univ.clust).mean()

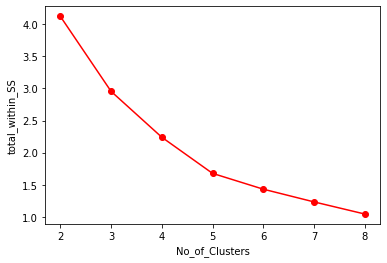
**Outputs:**



**Scatter plot show the slits of random data**



**Scatter plot show the slits of random data using Kmeans**



**Relation between TWSS (Total Within Sum of Square) and No. of Cluster**