## Première partie : Afficher les données :

### Code réaliser :

from scipy.io import arff

import matplotlib.pyplot as plt

data=arff.loadarff(open('./artificial/2d-10c.arff','r'))

listX = []

listY=[]

for point in data[0]:

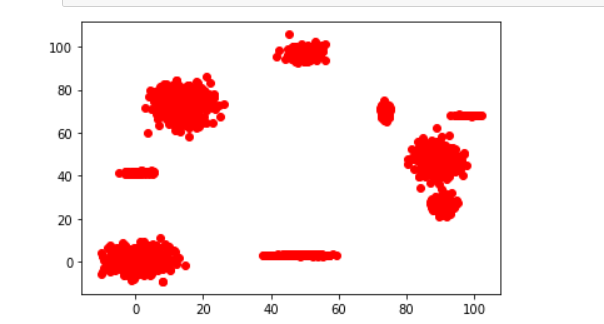
listX.append(point[0])

listY.append(point[1])

plt.plot(listX,listY, 'ro')

plt.show()

### Affichage :



from sklearn.cluster import KMeans

from sklearn.datasets import make\_blobs

import numpy as np

X = []

n\_samples=1500

for point in data[0]:

X.append([point[0],point[1]])

X=np.array(X)

kmeans = KMeans(n\_clusters=3, random\_state=0).fit\_predict(X)

plt.subplot(221)

plt.scatter(X[:, 0], X[:, 1], c=kmeans)

### Metrique

from sklearn import metrics

find = False

nb\_cluster = 2

metric\_best = 0

limit=20

iteration = 0

while(not find & (iteration < limit)):

kmeans = KMeans(n\_clusters=nb\_cluster, random\_state=0).fit\_predict(X)

find\_metric = metrics.silhouette\_score(X, kmeans, metric='euclidean')

if find\_metric > metric\_best:

metric\_best=find\_metric

nb\_cluster+=1

else:

find=True

iteration+=1

print(nb\_cluster)