Correction Activité 5 : K-means

```
from google.colab import drive

drive.mount('/content/gdrive')

import pandas

import imageio

from sklearn.preprocessing import StandardScaler

from matplotlib import pyplot as plt

from sklearn.preprocessing import MinMaxScaler

#from scipy.cluster.hierarchy import dendrogram, linkage

from sklearn.cluster import KMeans

import numpy as np

Columns = ['NumTimesPrg', 'PlGlcConc', 'BloodP', 'SkinThick', 'TwoHourSerIns', 'BMI', 'DiPedFunc',
'age', 'HasDiabetes']

dataframe = pandas.read_csv('/content/gdrive/MyDrive/ML/pratique/pima-indians-diabetes.csv')

dataframe.columns=Columns

#####################  Partie 1

#construction de l'ensemble de données formé par 100 individus et 2 variables

array = dataframe.values

data = array[:100,1:3]

#normalisation

scaler = MinMaxScaler(feature_range=(0, 1))

data = scaler.fit_transform(data)

scaler.transform(data)

# Affichage des points initiaux

plt.scatter(data[:,0], data[:,1], c='r')

plt.show()
```

```
###################### Partie 2
#effectuer la catégorisation en 2 classes avec k-means
kmeans = KMeans(n_clusters=2, n_init='auto')
kmeans.fit(data)
y_km = kmeans.fit_predict(data)
#Afficher les poitns aprés la la catégorisation pour 2 classes
plt.scatter(data[y_km ==0,0], data[y_km == 0,1], s=20, c='r')
plt.scatter(data[y_km ==1,0], data[y_km == 1,1], s=20, c='m')
#Conclusion : points abérants ont été inclus dans l'une des deux classes

#effectuer la catégorisation en 3 classes avec k-means
kmeans = KMeans(n_clusters=3, n_init='auto')
kmeans.fit(data)
y_km = kmeans.fit_predict(data)
#Afficher les poitns aprés la la catégorisation pour 3 classes
plt.scatter(data[y_km ==0,0], data[y_km == 0,1], s=20, c='r')
plt.scatter(data[y_km ==1,0], data[y_km == 1,1], s=20, c='m')
plt.scatter(data[y_km ==2,0], data[y_km == 2,1], s=20, c='y')
#Conclusion : la troisième classe ne contient que les points abérants

#effectuer la catégorisation en 4 classes avec k-means
kmeans = KMeans(n_clusters=4, n_init='auto')
kmeans.fit(data)
y_km = kmeans.fit_predict(data)
#Afficher les poitns aprés la la catégorisation pour 4 classes
plt.scatter(data[y_km ==0,0], data[y_km == 0,1], s=20, c='r')
plt.scatter(data[y_km ==1,0], data[y_km == 1,1], s=20, c='m')
plt.scatter(data[y_km ==2,0], data[y_km == 2,1], s=20, c='y')
plt.scatter(data[y_km ==3,0], data[y_km == 3,1], s=20, c='b')
#plt.scatter(data[y_km ==4,0], data[y_km == 4,1], s=10, c='r')
#Conclusion : sensibilité de k-means au bruit: un groupe  qui a été divisé en deux
```