UNIVERSITÉ PARIS 13

Master Informatique EID2

Traitement numérique des données

TP1 Introduction au module scikit-learn

Mohamed Ben Saad
Elias ABDELLI

```python
# Mohamed BEN SAAD
# Elias ABDELLI

#####################################
### A. Importation des libraires ####
#####################################
from sklearn.datasets.samples_generator import make_blobs
from sklearn import *
import numpy as np
import matplotlib.pyplot as plt
```

```python
##########################################
### B. Manipulation d'un jeu de données ###
##########################################

#--------------------------------------#
##### 1- Chargez les données Iris : #####
#--------------------------------------#
iris = datasets.load_iris()     #Téléchargement des données
data = iris.data                #Données
var = iris.feature_names        #Noms des variables
classe = iris.target_names      #Noms des classes
```

```python
#-------------------------------------------------------------------------------------------#
#### 2- Affichez les données, les noms des variables et le nom des classes (utilisez print) ####
#-------------------------------------------------------------------------------------------#
print('Données:\n',data[:10]) #Affiche les 10 premières lignes des données
print('\n')
print('Variables:\n',var)     #Affiche les noms des variables
print('\n')
print('Classes:\n',classe)    #Affiche le nom des classes
```

```
Données:
 [[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]


Variables:
 ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']


Classes:
 ['setosa' 'versicolor' 'virginica']
```

```python
#-----------------------------------------------------#
## 3- Affichez le nom des classes pour chaque donnée ##
#-----------------------------------------------------#
data_classe = iris.target
size = iris.target.size

for i in range(1,size,10):
  print(data[i], classe[data_classe[i]])

# Ici on affiche 5 exemples par classe pour évité trop d'affichage
```

```
[4.9 3.  1.4 0.2] setosa
[4.8 3.4 1.6 0.2] setosa
[5.1 3.7 1.5 0.4] setosa
[5.4 3.4 1.5 0.4] setosa
[4.5 2.3 1.3 0.3] setosa
[6.4 3.2 4.5 1.5] versicolor
[5.9 3.  4.2 1.5] versicolor
[6.1 2.8 4.  1.3] versicolor
[5.5 2.4 3.7 1. ] versicolor
[6.1 3.  4.6 1.4] versicolor
[5.8 2.7 5.1 1.9] virginica
[6.4 2.7 5.3 1.9] virginica
[5.6 2.8 4.9 2. ] virginica
[7.9 3.8 6.4 2. ] virginica
[6.9 3.1 5.1 2.3] virginica
```

```python
#-----------------------------------------------------------------------------------------#
## 4- Affichez la moyenne (mean), l'ecart-type (std), le min et le max pour chaque variable ##
#-----------------------------------------------------------------------------------------#
print('Moyenne :\n', iris.data.mean(0))
print('Ecart-type :\n', iris.data.std(0))
print('Min :\n', iris.data.min(0))
print('Max :\n', iris.data.max(0))
```

```
Moyenne :
 [5.84333333 3.05733333 3.758      1.19933333]
Ecart-type :
 [0.82530129 0.43441097 1.75940407 0.75969263]
Min :
 [4.3 2.  1.  0.1]
Max :
 [7.9 4.4 6.9 2.5]
```

```python
#-----------------------------------------------------------------------------------------#
## 5- En utilisant les attributs size et shape, affichez le nombre de données, le nombre de ##
# ---------------------------- variables et le nombre de classes ----------------------------#
#-----------------------------------------------------------------------------------------#
print('NB Data :\n', iris.data.shape[0])
print('NB Variable :\n', iris.data.shape[1])
print('NB Classe :\n', np.unique(iris.target).size)
```

```
NB Data :
 150
NB Variable :
 4
NB Classe :
 3
```

```
##############################################
# C. Téléchargement et importation de données #
##############################################

#-----------------------------------------#
## 1- Importez les données 'MNIST original' ##
#-----------------------------------------#

# Le datasets ne se télécharge pas...
# mnist = datasets.fetch_mldata('MNIST original')
# print('Données:\n',mnist.data)
# print('Variables:\n',mnist.feature_names)
# print('Classes:\n',mnist.target_names)
# print('Moyenne :\n', mnist.data.mean(0))
# print('Ecart-type :\n', mnist.data.std(0))
# print('Min :\n', mnist.data.min(0))
# print('Max :\n', mnist.data.max(0))
# print('NB Data :\n', mnist.data.shape[0])
# print('NB Variable :\n', mnist.data.shape[1])
# print('NB Classe :\n', np.unique(mnist.target).size)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.py:85: DeprecationWarning: Function fetch_
  warnings.warn(msg, category=DeprecationWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.py:85: DeprecationWarning: Function mldata
  warnings.warn(msg, category=DeprecationWarning)
```

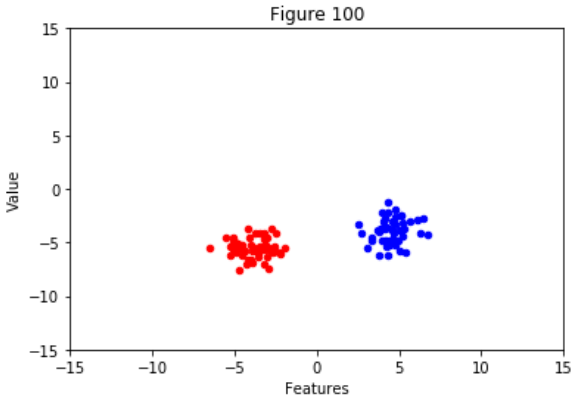
```
########################################
# D. Génération de données et affichage #
########################################

#-----------------------------------------------------------------------------------------#
## 1. Utiliser l'aide (help) pour voir comment utiliser la fonction datasets.make_blobs ##
#-----------------------------------------------------------------------------------------#
help(make_blobs)
```

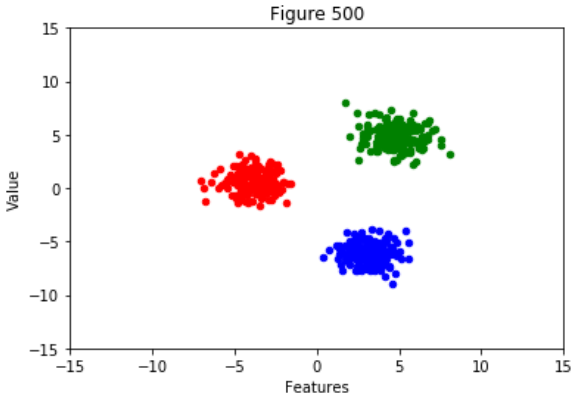
```
# ================================== #
# Fonction d'affichage de Scatter plot #
# ================================== #
def affichescatter(x,y,n,t):
  color = ['red','blue','green','yellow']
  plt.figure()
  plt.title(t)
  for i in range(n):
    plt.scatter(x[i,0], x[i,1], c = color[y[i]], s = 20)
  plt.xlim(-15,15)
  plt.ylim(-15,15)
  plt.xlabel('Features')
  plt.ylabel('Value')
  plt.show()
```

```
#-------------------------------------------------------------------#
## 2. Générez 1000 données de deux variables réparties en 4 groupes ##
#-------------------------------------------------------------------#
x1,y1 = make_blobs(n_samples= 1000, n_features = 2, centers =4, cluster_std=0.60, random_state=0)
t1 = 'Figure 1000'
affichescatter(x1,y1,y1.size,t1)
```


```
#-------------------------------------------------------------------#
## Générez 100 données de deux variables réparties en 2 groupes ##
#-------------------------------------------------------------------#
x2,y2 = make_blobs(n_samples= 100, n_features = 2, centers =2)
t2='Figure 100'
affichescatter(x2,y2,y2.size,t2)
```




```
#-------------------------------------------------------------------#
## Générez 500 données de deux variables réparties en 2 groupes ##
#-------------------------------------------------------------------#
x3,y3 = make_blobs(n_samples= 500, n_features = 2, centers =3)
t3='Figure 500'
affichescatter(x3,y3,y3.size,t3)
```



```
#-------------------------------------------------------------#
## Concaténez (vstack et hstack) les deux jeux de données ##
#-------------------------------------------------------------#
x4,y4 = np.vstack((x2,x3)), np.hstack((y2,y3))
t4='Figure 100 + Figure 500'
affichescatter(x4,y4,y4.size,t4)
```