Below code help you to read your data from your directory [your_TP_data] and extract feature based on [your_feature_extractor]

At the end your data will be available in x_data (features) and y_data (labels)

Entrée [235]:

```
import csv
import numpy as np
#set data_dir to the directory of your data files
data_dir= "H:\Home\Documents\ProjetIA\Dataset\Dataset/"

# Read file info file to get the list of audio files and their labels
file_list=[]
label_list=[]
with open(data_dir+"info.txt", 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        # The first column contains the file name
        file_list.append(row[0])
        # The last column contains the lable (language)
        label_list.append(row[-1])


# create a dictionary for labels
lang_dic={'EN':0,'FR':1,'AR':2,'JP':3}

# create a list of extracted feature (MFCC) for files
x_data=[]

for audio_file in file_list:
    file_feature = feature_extractor_1(data_dir+audio_file)
    #file_feature = feature_extractor_2(data_dir+audio_file)
    #add extracted feature to dataset
    x_data.append(file_feature)

# create a list of labels for files
y_data=[]
for lang_label in label_list:
    #convert the label to a value in {0,1,2,3} as the class label
    y_data.append(lang_dic[lang_label])
```

Entrée [ ]:

```
#random forest prend une matrice de taille inférieure ou égale a 2, donc je
#il a une dimension de taille 3
```

## 3. Shuffle your data

Using below code your data (features and corresponding labels) will be shuffled

Entrée [236]:

```
import random

# shuffle two lists
temp_list = list(zip(x_data, y_data))
random.shuffle(temp_list)
x_data, y_data = zip(*temp_list)
```

Entrée [252]:

```
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x_data,
                                                    y_data,
                                                    test_size=0.20,
                                                    shuffle=True)


# Train model
#clf.fit(X_train, y_train)

# Predict the test data
#y_pred = clf.predict(X_test)
```

## 4. Build your classifier

Now everything (almost) ready to build your classifier.

Below code is an example for creating an Random Forest classifier, training , and calculating its accuracy

Entrée [ ]:

```

```

Entrée [253]:

```
#RANDOM FOREST CLASSIFIER

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

clf = RandomForestClassifier(max_depth=70)
#en mettant max_depth a 9 on obtient 90%
clf.fit(x_data, y_data)
# Train model
#clf.fit(X_train, y_train)
# Predict the test data
#y_pred = clf.predict(X_test)
# the resulted accuracy is on a small set which is same for train and test
#print("Accuracy",clf.score(x_data, y_data))
#print("Accuracy : ",accuracy_score(y_test,y_pred))

print("Accuracy with all data : ",clf.score(x_data, y_data))
```

```
Accuracy with all data :  0.9949066213921901
```

Entrée [254]:

```
#GAUSSIAN NAIVES BAYES CLASSIFIER

from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()
clf.fit(x_data, y_data)
# Train model
#clf.fit(X_train, y_train)
# Predict the test data
#y_pred = clf.predict(X_test)
# the resulted accuracy is on a small set which is same for train and test
#print("Accuracy with all data : ",clf.score(x_data, y_data))
#print("Accuracy : ",accuracy_score(y_test,y_pred))

print("Accuracy with all data : ",clf.score(x_data, y_data))
```

```
Accuracy with all data :  0.4601018675721562
```

Entrée [255]:

```
#C-SUPPORT VECTOR CLASSIFIER
```