```python
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 15 15:12:15 2020

A comprehensively annotated example of XG-boost on the Iris dataset, just to
illustrate the use XG-boost as opposed to ordinary decision trees. In this case
the accuracy was amazing!

@author: Walter
"""
from sklearn.datasets import load_iris

iris = load_iris()

no_Samples, no_Features = iris.data.shape

#print(no_Samples)
#print(no_Features)
#print(list(iris.target_names))

#Spilting off 20% for the test data, leaving me with 80%
#for the training

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target,
                                    test_size=0.2, random_state=0)



#loading up XG-oost and converting the both groups of data into the DMatrix form that it supports
```

```python
import xgboost as xgb

train = xgb.DMatrix(X_train, label=y_train)
test = xgb.DMatrix(X_test, label=y_test)

#Defining the hyperparameters by defining the dictionary. Using softmax since this is a multiple
# classification problem. The other parameters should ideally be tuned
# through experimentation, much like the k count in k means

param = {
    'max_depth': 4,
    'eta': 0.3,
    'objective': 'multi:softmax',
    'num_class': 3}
epochs = 10

#an epoch is complete pass through the training data
#NB its not softmax that is minimized in XG-boost, but the crossentropy loss
#function, which is based on softmax. Crossentropy is calculated on a
#softmax output, that's why they are a standard couple in ML.
#Tree-based classifiers like XGB find "cuts", or portions of the variables'
#space in a way that minimizes the entropy of a dataset.

#Training the model

model = xgb.train(param, train, epochs)

#Using the trained model for the predictions

predictions = model.predict(test)
```

```
#print(predictions)

#Measuring the accuracy on the test data...

from sklearn.metrics import accuracy_score
Accuracy_Result = accuracy_score(y_test, predictions)

print("The accuracy of the XGBoost model was",Accuracy_Result)

#Returned result of 1 which means perfect accuracy
```