Task: Creating a classifier using RandomForests

Description:

1. Data processing has been done using **pandas** package.
2. The given input data is split into two components:
   1. Training Data: Data used to train the RandomForest classifier
   2. Test Data: Data used to test the efficiency of the classifier
3. Before the data is fed into the classifier, a sanity check is done to ensure equal no. of columns in training and test data.
4. A **RandomForest** classifier is created using **scikit\_learn** package and training data is fed as input to model the data.
5. The classifier is then tested against test data and error is calculated. The error is 0.155306678407. This can be further minimized by increasing the no. of decision trees of the classifier.
6. Finally, a sample data is taken from test input and the output is predicted using the classifier. The error between predicted and actual output is zero.

Source Code:

#Sai Saketh Nandagiri

#Creating a Random Classifier for given TestData

import numpy as np

import pandas as pd

#Application of RandomForestClassifier for modeling the data

from sklearn.ensemble import RandomForestClassifier

#Reading the csv document

InputData=pd.read\_csv("C:\Users\saketh sai\Desktop\DataScience\DataScience\DataScience\Vish\InterviewTest.csv")

#Preparing Training and Testing Data

TrainData=InputData.ix[range(50000)]

TestData=InputData.ix[range(50000,len(InputData))]

print "Length of Training Data: "+str(len(TrainData))

print "Length of Testing Data: "+str(len(TestData))

#Unloading target column for both training and test data

TrainOutput=TrainData.pop("Output")

TestOutput=TestData.pop("Output")

#Sanity check

print "Q: Cols in Train and Test Data:"+str(TrainData.columns.equals(TestData.columns))

#Creating a RandomForest Classifier

#There are 1000 decision trees created to make the decision

rf=RandomForestClassifier(n\_estimators=1000, criterion="entropy")

#Training the classifier

rf.fit(TrainData, TrainOutput)

#Testing the performance with TestData

Error=np.average(np.abs(TestOutput^rf.predict(TestData)))

print "Error in classification is: "+str(Error)

#Sample Test Data for representation

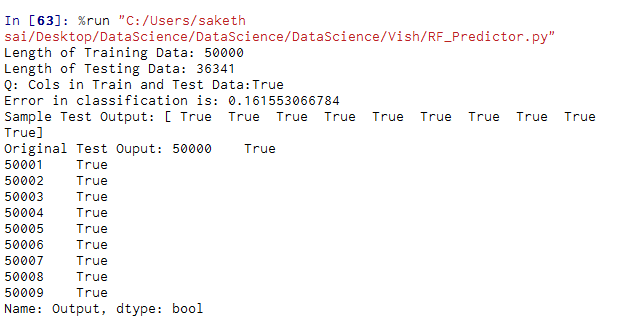
SampleTestData=TestData.ix[range(50000,50010)]

SampleTestOutput=rf.predict(SampleTestData)

print "Sample Test Output: "+str(SampleTestOutput)

print "Original Test Ouput: "+str(TestOutput.head(10))

Output:



References:

1. RandomForestClassifier documentation, <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
2. Tutorial on classification using Pandas and Scikit-Learn, <http://pyvideo.org/depy-2015/classification-using-pandas-and-scikit-learn.html>
3. Understanding RandomForest classifier, <https://en.wikipedia.org/wiki/Random_forest>