```
#SDL Assignment 4
#Decision Tree Algorithm
import pandas as pd
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.externals.six import StringIO
from IPython.display import Image
import numpy as np
import pydotplus


data = pd.read_csv('zbills.csv')    #Data-Set
x = data.iloc[:,:4]                 #Predicted Value
y = data.iloc[:,4]                  #Response Value

#Train Algorithm & Return PREDICTION set
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size = 0.2)
#Split Data-Set into xtrain,ytrain,xtest,ytest
clf = DecisionTreeClassifier()
clf.fit(xtrain,ytrain)
ypredict = clf.predict(xtest)
Accuracy = clf.score(xtest,ytest) * 100
tries = 1
'''
while Accuracy != 100.0000:
    tries += 1
    xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size = 0.2)
    clf.fit(xtrain,ytrain)
    ypredict = clf.predict(xtest)
    Accuracy = clf.score(xtest,ytest) * 100

print 'No of TRIES: ', tries'''
print confusion_matrix(ytest,ypredict)          #confusion_matrix(
Actual-Class , Predicted-Class )
print Accuracy                                  #PRINT --> Accuracy
print classification_report(ytest, ypredict)    #Classification Report

new = np.array([[0.74521,3.6357,-4.4044,-4.1414]])
ypredict = clf.predict(new)
print "Predicted Value: ", ypredict

dot_data = StringIO()
export_graphviz(clf,
out_file=dot_data,filled=True,rounded=True,special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('tree.png')
```