In [ ]:
```
# Author : Amir Shokri
# github link : https://github.com/amirshnll/Online-Shoppers-Purchasing-Intent
ion/
# dataset link : http://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purcha
sing+Intention+Dataset
# email : amirsh.nll@gmail.com
```

In [1]:
```
import csv

def transformDataMTT(trainingFile, features):

    transformData=[]

    labels = []

    blank=""

    # Now we are finally ready to read the csv file
    with open(trainingFile,'r') as csvfile:
        lineReader = csv.reader(csvfile,delimiter=',')
        lineNum=1
        # lineNum will help us keep track of which row we are in
        for row in lineReader:
            if lineNum==1:
                header = row
            else:
                allFeatures=list(row)
                featureVector = [allFeatures[header.index(feature)] for featur
e in features]
                if blank not in featureVector:
                    transformData.append(featureVector)
                    labels.append(int(row[1]))
            lineNum=lineNum+1
        return transformData,labels
    # return both our list of feature vectors and the list of labels
```

In [2]:
```
# Let's take this for a spin now
trainingFile="O_S_I_train.csv"
features=["Administrative","Informational","ProductRelated","ProductRelated_Du
ration","BounceRates","ExitRates","PageValues","SpecialDay","Month","Operating
Systems","Browser","Region","TrafficType","VisitorType","Weekend","Revenue"]
trainingData=transformDataMTT(trainingFile,features)
```

In [3]:
```
# We are now ready to train our Decision Tree classifier
from sklearn import tree
import numpy as np
clf=tree.DecisionTreeClassifier(max_leaf_nodes=20)
X=np.array(trainingData[0])
y=np.array(trainingData[1])
clf=clf.fit(X,y)
```

In [4]:
```
import graphviz
with open("MTTTEST.dot","w") as f:
    f = tree.export_graphviz(clf,
                             feature_names=features,out_file=f)
```

In [5]:
```
clf.feature_importances_
```

Out[5]:
```
array([0.00000000e+00, 9.98560734e-01, 0.00000000e+00, 3.69042494e-04,
       0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
       2.76781870e-04, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
       7.93441362e-04, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00])
```

In [6]:
```
def transformTestDataMTT(testFile,features):

    transformData=[]
    ids=[]
    blank=""
    with open(testFile,"r") as csvfile:
        lineReader = csv.reader(csvfile,delimiter=',')
        lineNum=1
        for row in lineReader:
            if lineNum==1:
                header=row
            else:
                allFeatures=list(row)
                featureVector = [allFeatures[header.index(feature)] for featur
e in features]
                #featureVector=list(map(lambda x:0 if x=="" else x, featureVec
tor))
                transformData.append(featureVector)
                ids.append(row[0])
            lineNum=lineNum+1
    return transformData,ids
```

In [21]:
```
def MTTTest(classifier,resultFile,transformDataFunction=transformTestDataMTT):
    testFile="O_S_I_test.csv"
    testData=transformDataFunction(testFile,features)
    result=classifier.predict(testData[0])
    with open(resultFile,"w") as mf:
        ids=testData[1]
        lineWriter=csv.writer(mf,delimiter=',')
        lineWriter.writerow(["ShopperId","Revenue"])
        for rowNum in range(len(ids)):
            try:
                lineWriter.writerow([ids[rowNum],result[rowNum]])
            except Exception as e:
                print (e)
# Let's take this for a spin!
resultFile="result.csv"
MTTTest(clf,resultFile)
```