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 pandas
trainingData = pandas.read_csv("O_S_I_train.csv")
```

In [14]:
```
# 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)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get
_loc(self, key, method, tolerance)
   2894             try:
-> 2895                 return self._engine.get_loc(casted_key)
   2896             except KeyError as err:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHas
hTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHas
hTable.get_item()

KeyError: 0

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
<ipython-input-14-5dc60f5dc5f1> in <module>
      4 
      5 clf=tree.DecisionTreeClassifier(max_leaf_nodes=20)
----> 6 X=np.array(trainingData[0])
      7 y=np.array(trainingData[1])
      8 clf=clf.fit(X,y)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem_
_(self, key)
   2900             if self.columns.nlevels > 1:
   2901                 return self._getitem_multilevel(key)
-> 2902             indexer = self.columns.get_loc(key)
   2903             if is_integer(indexer):
   2904                 indexer = [indexer]

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get
_loc(self, key, method, tolerance)
   2895                 return self._engine.get_loc(casted_key)
   2896             except KeyError as err:
-> 2897                 raise KeyError(key) from err
   2898 
   2899         if tolerance is not None:

KeyError: 0
```

In [ ]:
```
print(trainingData[0])
```

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

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

In [ ]:
```
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 [ ]:
```
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)
```