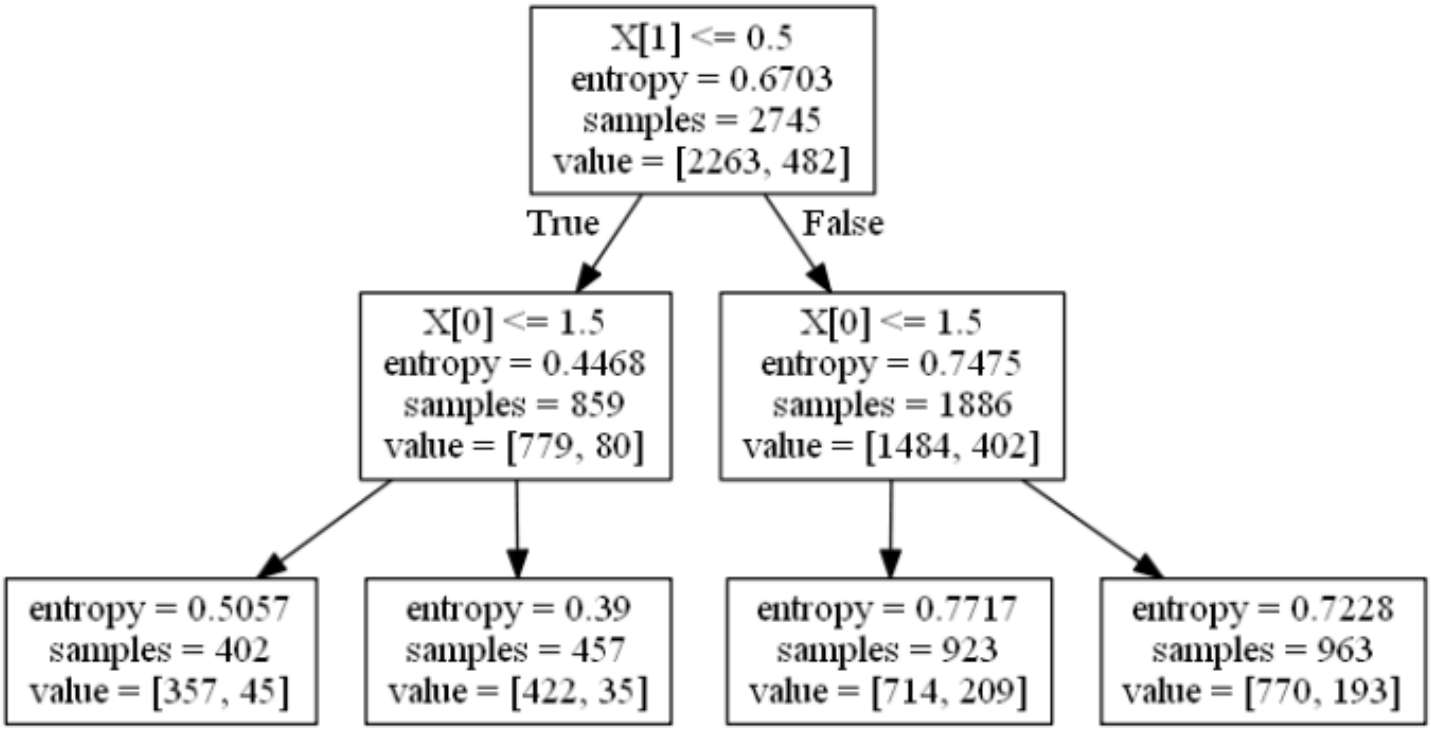
Olivier's blog

Decision trees

Publié le 2 février 20168 février 2016 par Olivier

This is an example of the result of a Decision tree generated with Scikit-Learn (http://scikit-learn.org) and Graphviz (http://www.graphviz.org/):



This Data set is issued from the Wesleyan University course 'Machine learning for data analysis' (https://www.coursera.org/learn/machine-learning-data-analysis).

In this Data set, a list of features (age, sex, race, depression, etc.) were used to explain regular smoking. In this case, only 2 features are used : « sex » and « white » (« white race »). These features are booleans. As a result, there are 4 groups at the bottom of the tree.

The whole Data set has been used. The chosen criteria to split the Data set is « entropy », more specifically « the entropy gain ». The objective is to maximize the entropy gain when chosing the feature used to split the Data set.

As we can see, the first feature used is the « sex » and the next is the « white race ».

Here is the confusion matrix and the accuracy score :

```
— Confusion matrix —
[[1505    0]
[ 325    0]]
— Accuracy score —
0.822404371585
```

The Python, adapted from Jen Rose and Lisa Dierker 's code :

```
import pandas as pd
import os
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
import sklearn.metrics
from sklearn import tree
from sklearn.externals.six import StringIO
import pydot

os.chdir(« C:\Users… »)

«  » »
Data Engineering and Analysis
«  » »
#Load the dataset
AH_data = pd.read_csv(« tree_addhealth.csv »)
data_clean = AH_data.dropna()

data_clean.dtypes
data_clean.describe()
«  » »
Modeling and Prediction
«  » »
#Split into training and testing sets
predictors = data_clean[['BIO_SEX','WHITE']]
targets = data_clean.TREG1

pred_train, pred_test, tar_train, tar_test  =   train_test_split(predictors, targets, test_size=.4)

pred_train.shape
pred_test.shape
tar_train.shape
tar_test.shape

#Build model on training data
classifier=DecisionTreeClassifier(criterion= »entropy »)
classifier=classifier.fit(pred_train,tar_train)

predictions=classifier.predict(pred_test)
```

```
print '— Confusion matrix —'
print sklearn.metrics.confusion_matrix(tar_test,predictions)
print '— Accuracy score —'
print sklearn.metrics.accuracy_score(tar_test, predictions)

#Displaying the decision tree

dot_data = StringIO()
tree.export_graphviz(classifier, out_file=dot_data)
graph = pydot.graph_from_dot_data(dot_data.getvalue())
graph.write_pdf(« graph.pdf »)

graph.write_png(« graph.png »)
```