CST 383 - Intro to Data Science

Dr. Glenn Bruns

# Lab: Classification Trees

The purpose of this lab is to start getting some practice using Scikit-Learn to build classification trees.

**Instructions**:

1. Explain why decision trees are non-parametric models.
2. Create a Python file. Use this code to read and preprocess the data. (see number 7 for how to install graphviz)

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.tree import DecisionTreeClassifier, export\_graphviz

from sklearn.model\_selection import train\_test\_split

import graphviz

df = pd.read\_csv('https://raw.githubusercontent.com/grbruns/cst383/master/College.csv', index\_col=0)

1. Convert the 'Private' column to an numeric column with values 0 and 1 (1 for private colleges).
2. Do a little exploration of the data to remember what it’s like. E.g., use df.info(), df.describe().
3. We will try to predict whether a college is public or private. Select a few predictors, create NumPy arrays X and y, and then do a training/test split. Try hard to remember how to do this from memory. If you can't, refer to the hints.
4. Train a tree classifier using Scikit-Learn's DecisionTreeClassifier. Use the training data you created in the previous step.
5. Install graphviz by entering conda install python-graphviz at the Anaconda prompt. Then plot your tree using graphviz. Try playing with some of the options of export\_graphviz().
6. Use your classification tree to predict whether examples in your test data are public or private. Compute the confusion matrix and the accuracy of your predictions..
7. If you still have time, do the following:

* try building more classification trees, using different sets of input features
* look at, and play with, the hyperparameters available in DecisionTreeClassifier, especially max\_depth.
* see how much the classification tree that you produce depends on your particular training set

## Hints:

1. -
2. -
3. df['Private'] = (df['Private'] == 'Yes').astype(int)
4. -

predictors = ['Outstate', 'F.Undergrad']

X = df[predictors].values

y = df['Private'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30,

random\_state=0)

1. -

dot\_data = export\_graphviz(clf, precision=2,

feature\_names=predictors,

proportion=True,

class\_names=target\_names,

filled=True, rounded=True,

special\_characters=True)

# plot it

graph = graphviz.Source(dot\_data)

graph

1. -
2. -
3. -