CST 383 - Intro to Data Science

Dr. Glenn Bruns

# Lab: Decision trees, part 2

In this lab we will look at how classification trees are built. Look at hints only if you get stuck.

1. We defined the Gini index value for a node in a classification tree as

Where p is the estimated probability of either of the two classes. For example, if a node has 30 training instances of class A, and 50 training instances of class B, then the estimated probability of class A is 30/(30 + 50). What is the Gini index value for this node?

1. Create a new Python file. Fill in the code below to create a function that gives the Gini value for a node in a binary classification tree given values for the number of instances of each class. class\_counts is a list of length two.

def gini(class\_counts):

""" return the Gini value for a node in a binary classif. tree """

# your code here (don't forget the return statement)

1. Test your function. What is gini([30, 50])? gini([10, 10])? What is gini([20, 0])? What is gini([100, 0])?
2. Add the following code at the top of your file to read and preprocess the data.

df = pd.read\_csv("https://raw.githubusercontent.com/grbruns/cst383/master/heart.csv")

df['output'] = df['output'] - 1

df = df[['age', 'maxhr', 'restbp', 'output']]

sns.scatterplot(x='age', y='maxhr', hue='output', data=df)

1. Run the code and look at the plot. If we were going to build a classification tree, and split first on 'age', what do you think a good age value to split on would be?
2. Compute the Gini index for df as a whole. For this you just need the number of rows with output = 0 and the number of rows with output = 1.
3. Now consider a split on age < 50. Write code to compute the Gini index for the case of of age < 50 and the Gini index for the case of age >= 50. For the case of age < 50, get the rows of df where age < 50, then count the number of rows with output = 0 and output = 1.
4. Now compute the overall Gini index value for the split on age < 50. First you need to compute the fraction of nodes associated with age < 50 (call it fraction\_lo) and the fraction of nodes associated with age >= 50 (call it fraction\_hi). Then get the Gini value for the split like this (in pseudo code):

gini\_split = gini\_lo \* fraction\_lo + gini\_hi \* fraction\_hi

The split is useful if the Gini value for the split is lower than the GIni value for the root.

1. Is a split on age < 40 better than a split on age < 50?
2. Compute the Gini value for all age splits where age ranges from 20 to 80. Then plot the Gini split value for all the ages (age on x axis, Gini value on y axis). What is the best age value for a split on age?
3. If you still have time, do the same thing for features 'maxhr' and 'restbp'. What is the best feature and best split?

## 

## Hints

1. About 0.47.

def gini(class\_counts):

""" return the Gini value for a node in a binary classif. tree """

if sum(class\_counts) == 0:

return 0

p = class\_counts[0]/sum(class\_counts)

return 2 \* p \* (1 - p)

1. -
2. -
3. -

gini\_root = gini([(df['output'] == i).sum() for i in [0,1]])

split\_val = 50

df\_lo = df[df['age'] < split\_val]

df\_hi = df[df['age'] >= split\_val]

counts\_lo = [(df\_lo['output'] == i).sum() for i in [0,1]]

counts\_hi = [(df\_hi['output'] == i).sum() for i in [0,1]]

gini\_lo = gini(counts\_lo)

gini\_hi = gini(counts\_hi)

fraction\_lo = df\_lo.shape[0]/df.shape[0]

fraction\_hi = df\_hi.shape[0]/df.shape[0]

gini\_split = fraction\_lo \* gini\_lo + fraction\_hi \* gini\_hi

1. -
2. My plot looks like this:

