Lab 4G - Growing trees

Directions: Follow along with the slides and answer the questions in **bold** font in your journal.

## Trees vs. Lines

* So far in the labs, we've learned how we can fit linear models to our data and use them to make predictions.
* In this lab, we'll learn how to make predictions by growing trees.
  + Instead of creating a line, we split our data into branches based on a series of *yes* or *no* questions.
  + The branches help sort our data into *leaves* which can then be used to make predictions.
* Start, by loading the titanic data.

## Our first tree

* Use the tree() function to create a *classification* tree that predicts whether a person survived the Titanic based on their gender.
  + A *classification* tree tries to predict which category a categorical variable would belong to based on other variables.
  + The syntax for tree is similar to that of the lm() function.
  + Assign this model the name tree1.
* **Why can't we just use a *linear model* to predict whether a passenger on the Titanic survived or not based on their gender?**

## Viewing trees

* To actually look at and interpret our tree1, place the model into the treeplot function.
  + **Write down the labels of the two *branches*.**
  + **Write down the labels of the two *leaves*.**
* Answer the following, based on the treeplot:
  + **Which gender does the model predict will survive?**
  + **Where does the plot tell you the number of people that get sorted into each leaf? How do you know?**
  + **Where does the plot tell you the number of people that have been sorted *incorrectly* in each leaf?**

## Leafier trees

* Similar to how you included multiple variables for a linear model, create a tree that predicts whether a person survived based on their gender, age, class, and where they embarked.
  + Call this model tree2.
* Create a treeplot for this model and answer the following question:
  + **Mrs. Cumings was a 38 year old female with a 1st class ticket from Cherbourg. Does the model predict that she survived?**
  + **Which variable ended up not being used by tree?**

## Tree complexity

* By default, the tree() function will fit a *tree model* that will make good predictions without needing lots of branches.
* We can increase the complexity of our trees by changing the complexity parameter, cp, which equals 0.01 by default.
* We can also change the minimum number of observations needed in a leaf before we split it into a new branch using minsplit, which equals 20 by default.
* Using the same variables that you used in tree2, create a model named tree3 but include cp = 0.005 and minsplit = 10 as arguments.
  + **How is tree3 different from tree2?**

## Misclassification rate

* Similar to how we use the *mean squared error* to describe how well our model predicts numerical variables, we use the *misclassification rate* to describe how our model predicts categorical variables.
  + The *misclassification rate* (MCR) is the number of people who were predicted to be in one category but were actually in another.
  + Fill in the blanks to create a function to calculate the MCR

calc\_mcr <- function(actual, predicted) {  
 sum(\_\_\_\_ != \_\_\_\_) / length(\_\_\_\_)  
}

## Predictions and Cross-validation

* Just like how with *linear models*, we can use cross-validation to measure our *classification trees* prediction accuracy.
  + Use the data function to load the titanic\_test data.
  + Fill in the blanks below to predict whether people in the titanic\_test data survived or not using tree1.

titanic\_test <- mutate(\_\_\_\_, prediction = predict(\_\_\_\_, newdata = \_\_\_\_, type = "class"))

* Then run the following to calculate the MCR

summarize(titanic\_test, mcr = calc\_mcr(survived, prediction))

## On your own

* **In your own words, explain what the *misclassification rate* is.**
* **Which model (tree1, tree2 or tree3) had the lowest misclassification rate for the titanic\_test data?**
* Create a 4th model using the same variables used in tree2. This time though, change the *complexity parameter* to 0.0001. Then answer the following
  + **Does creating a more complex *classification tree* always lead to better predictions? Why not?**
* A *regression tree* is a tree model that predicts a numerical variable. Create a *regression tree* model to predict the Titanic's passenger's ages and calculate the MSE.
  + Plots of regression trees are often too complex to plot.