15.095 Homework 3
Due: Oct 29, 2018

In this assignment, you will run Optimal Classification Trees (OCTs) and Optimal Imputations (`opt.impute`) on the Iris data set from the UCI Machine Learning Repository. This is a classification data set with has $n = 150$ observations, $p = 4$ features, and $k = 3$ class labels. A csv file with the data is on Canvas: *iris.csv*. In this file, the last column is the class label which will be predicted, and rest of the columns are numeric features. For parts 4 and 5, we have provided the csv file *iris-missing.csv* which is the same data set, but with 50% of the entries missing completely at random from each of the 4 feature columns.

Note: If do you not have OptimalTrees and OptImpute installed, you can use the following instructions for installation with the zip file uploaded on Canvas (`http://jack.dunn.nz/OptimalTrees.jl/latest/installation.html#Installing-a-precompiled-system-image-1`).

Begin by reading in the data and splitting into 50%/25%/25% training, validation, and testing sets.

1. **Optimal Classification Trees (OCT)**
   Fit an Optimal Classification Tree with parallel splits model on the data set. Tune the minbucket, and max_depth parameters for the model. Report the best combination of parameters along with the training, validation, and testing accuracy values.

2. **Variable Importance**
   When training a machine learning model, it is also important for us to understand which variables are most important or influential in the final model. For example, in a linear regression model $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$, we can sort the absolute values of the coefficients $(|\beta_1|, |\beta_2|, \ldots, |\beta_p|)$ to come with a ranked list of the variables in order of their relative influence on the final prediction.

   (a) Define your own metric / method to determine variable importance in an OCT model. You can just describe the algorithm; you do not need to implement this.

   (b) Run `OptimalTrees.variable_importance(lnr)` to find the most important variables in this dataset, using the OCT model that you trained in part 1.

3. **OCT and ANN:** Is it possible to construct an artificial neural network that gives the exact same predictions as the OCT model that you found in part 1? If so, how many nodes are in each layer of this neural network?

4. **Missing Data Imputation**
   For this question, we will consider the same data set, but with 50% of the values missing completely at random (MCAR). Read in the data from *iris-missing.csv*, and impute the missing values using mean and `opt.knn` imputation, leaving out the outcome variable.(You just need to impute missing data on whole data set, you don't need to impute missing data on training and test data separately.) Report the MAE for each method; which is more accurate? Re-train the OCT models from 1, and report the best combinations of parameters and test accuracy scores. Give some intuition for why one method outperforms the other. *Hint:* Look at the correlation matrix for the data.

5. **Imputation including the Y-variable**
   Rerun the `opt.knn` imputation, except this time include the outcome variable as a column in the data frame to be imputed. Re-train the OCT models from parts 1, and report the test accuracy scores. What do you observe? Do you think that it is a good idea to include the Y-variable in our imputation? Why or why not?