**Assignment: Boosting**

**BUAD 5082 – Spring 2019**

**1.** **Objectives**

The purpose of this assignment is to provide you with an opportunity to investigate some of the ideas regarding boosting.

**2.** **What You Will Need**

Access to a Windows computer with R, and to the following files, which can be downloaded from the Class Schedule page of the course web site:

**3.** **What You Will Hand In**

Nothing, it is not expected for you to submit this assignment.

**4.** **Due Date**

March th, 2019: just before midnight.(Have a wonderful spring break but do not forget to work!) After this date we will post the solutions to the problem set.

**5.** **Note on Collaboration**

This is not an official assignment. You may work with others solve this problem set.

**6.** **Preliminaries:**

To get set up for the assignment, follow these steps:

1. As the first statement in your script file, enter rm(list=ls())

2. Each question and part in the assignment should begin with the following three comment lines where n is the question number and p is the part letter:

###########################

####QUESTION n PART p####

###########################

**7. Assignment Tasks:**

Theoretical Understanding. True/False statements.

|  |  |
| --- | --- |
| 1. Boosting can be applied to both regression and classification: | True False |
| 2. Like Bagging, Boosting uses the bootstrap to create independent trees: | True False |
| 3. Unlike Bagging and Random Forests, a large number of trees used in Boosting can cause a model to overfit: | True False |
| 4. The shrinking parameter λ controls the rate at which a Boosting model learns: | True False |
| 5. It is not often that only 1 split in each tree works well in Boosting: | True False |

Problems Using R on Classification Boosting.

1. Load the Boston and ada libraries. Set the seed to 5082
2. Create training and test indices using the sample function. Use an 80/20 split
3. Create a new dataframe from duplicating the Boston data set
4. Create a variable that stores the median of the medv variable.
5. Create a new column in your new data frame that assigns ‘Above’ or ‘Below’ to each value of medv, indicating whether it is above or below the median found in 4
   1. Tip: This can be done by replacing the medv column itself, or by creating a new vector and Column binding it onto dataframe. For the latter method, make sure to remove the previous medv values from model as they would be perfect predictors.
6. Print out the new column of the data frame, ensuring that values are only ‘Above’ and ‘Below’
7. Using your training set, create a boosting model using the ada function, 20 iterations, bagging fraction of .5, and stumps.

to classify the variable you created in 5.

Note: you may have to convert your median vector using as.factor() for this to work.

Tip: decision stumps can be created by using (control=rpart.control(maxdepth=1).

1. What is the training error and out of bag error of this model?
2. Use your model to make a set of predictions on your test set, and display the resulting confusion matrix.
3. Using a for loop to find the best number of iterations in your boosting model. Keep all other parameters the same but use the for loop to go through a sequence of iterations from 10 to 500 by 10 (may take a couple minutes to run). In the loop, apply each model to the test set and store the **test** **error rates** for each iteration. Then plot the relationship between test error rate and iteration number. Why does the overall trend of error rate go down then back up when the training model is applied to the test set?
4. From the perspective of a real estate agent, what are the strengths and limitations of predicting median house values in this way?