Decision Trees, Random Forests, and Gradient Boosting

Before starting this assignment, make sure to go through the R lab in chapter 8 of Introduction to Statistical Learning. Random forests and gradient boosting are often the default choice when building predictive models; it is important that you understand how they work and how to fit them.

ISLR 8.8

This problem will use the Carseats data in the ISLR package. We will predict sales using the other variables.

Problem 1

The code below splits the data into a training and test set

```
set.seed(1988)
N = nrow(Carseats)
train_prop = 0.8
train_index = sample(1:N, size = floor(N * train_prop), replace = FALSE)
train_dat = Carseats[train_index, ]
test_dat = Carseats[-train_index, ]
```

Problem 2

Fit a regression tree to the dataset. Report the MSE on your test set.

Problem 3

Prune the tree using cross-validation. Plot the deviance by tree size. How many folds does the CV function use by default?

Problem 4

The plot should show the deviance flattening out around a tree size of 8 and attaining its minimum at a size of 16. Prune the tree to a size of 8 and 16 and report test MSE.

Problem 5

Which pruning size would you prefer? Justify your answer.

Problem 6

Fit a random forest to the training data with the ntree range given below. Make a plot of training and test MSE by tree size. Report the minimum of test MSEs. Is the improvement meaningful over decision trees?

```
ntree = seq(1, 500, by = 25)
```

Problem 7

Perform gradient boosting for the range of learning rates given below. Fix the number of trees at 100 and interaction depth at 2. Report the minimum of test MSE.

```
lambda = seq(0.001, 0.5, length = 25)
```

Problem 8

Do the same analysis above, but with learning rate fixed at 0.01, ntree fixed at 500, and varying interaction depth in the range below.

```
interaction_depth = 1:10
```

Problem 9

Look at the gbm documentation. List three parameters in the model that can be tuned using cross-validation. Also state whether increasing them (holding all else constant) would make the model more or less flexible.

Problem 10

Re-run the analysis with a different seed (you don't have to present the results). Do you get different test MSEs? What can we change in our process above to stabilize the estimate of test error?