STAT 675 – Homework 4
Solution compiled by Miranda Fix, Ben Goldman and Lei Yang

1. Let's look at the digits data with boosting. Download the digits data[1] and read the website for relevant information about the dataset. We are going to compare classifying the 4's and 9's, which tends to be difficult. Create a training and a test data set.

   Read the data by functions online [2]. And choose the cases of 4's and 9's.

   FOR each of the following base classifiers: logistic regression with 1 covariate, logistic regression with 10 covariates, trees with 1 split (stump), and trees with 10 splits; DO:

   (a) For AdaBoost, make a plot of the training error and test error as a function of the number of boosting iterations. Do you see evidence of overfitting?

   For logistic regression as base classifier, Ben and Lei have different results. Neither of them seem apparently wrong, so both are presented in Figure 1 and Figure 2 (and code attached). You can look into their code to see why there is difference.

   Trees with 1 split (stump), and trees with 10 splits as base classifiers are in Figure 3.

   Comparison between trees with 10 splits and logistic regression with 10 covariates in Figure 4.

   Though some of test errors fluctuate after some iterations, they don't grow apparently; thus we see no strong sign of overfitting in these figures.

   (b) Do the same for LogitBoost (You can look at `http://stat.ethz.ch/∼dettling/boosting.html` for an implementation for stump classifiers)

   Logitboost needs base function instead of base classifier. Because logistic regression is not efficient as a base function, we use weighted linear regression as the base function (Figure 5), as well as tree method (Figure 6). Again, since the test errors don't grow dramatically when iterations increase, there is no sign of overfitting in these figures.

   Comparison between 10 covariate linear regression and 10 split tree methods in Figure 7.
   Compared to AdaBoost results (Figure 8 and 9).

   Which procedure combination works best?

[1] `http://yann.lecun.com/exdb/mnist/`
[2] `https://gist.github.com/brendano/39760`

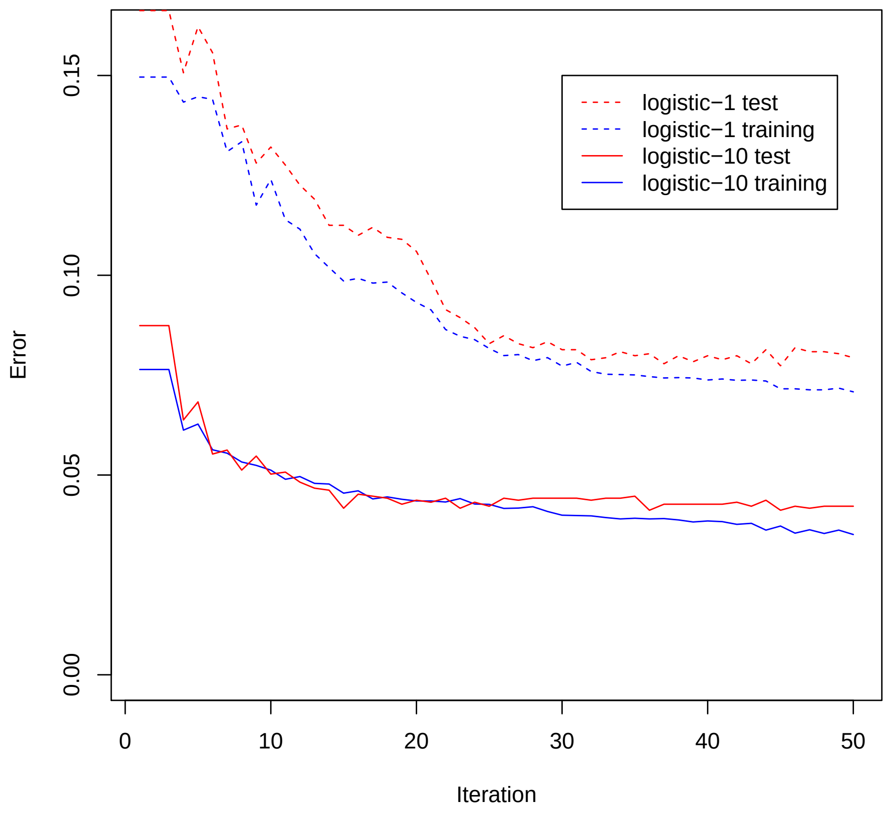

Figure 1: AdaBoost with logistic regression 1

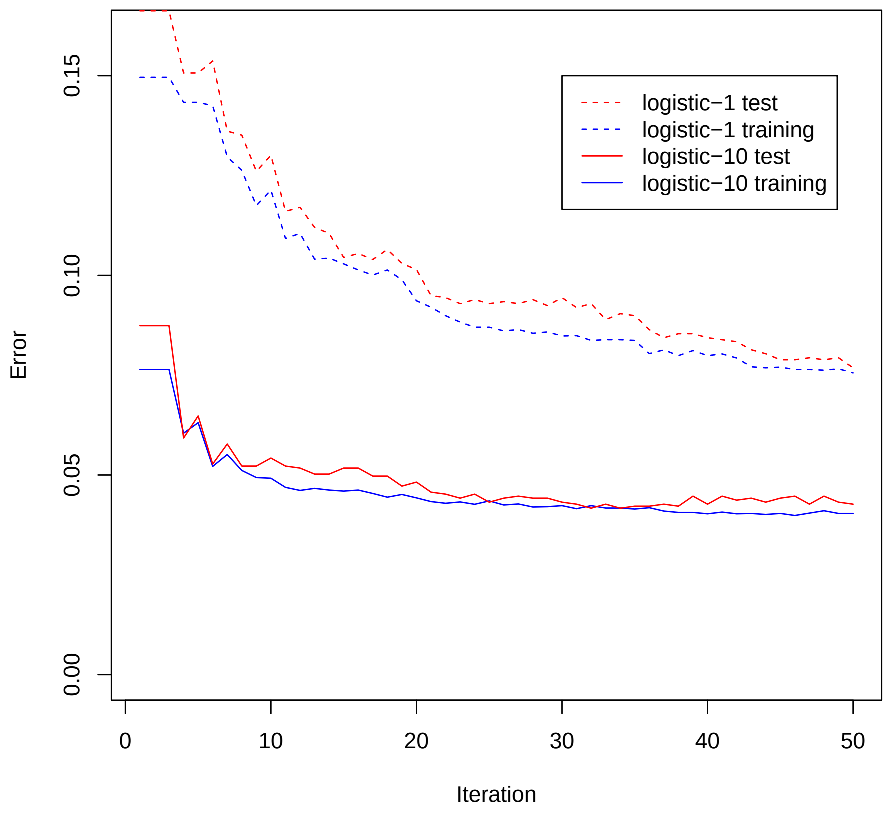

Figure 2: AdaBoost with logistic regression 2

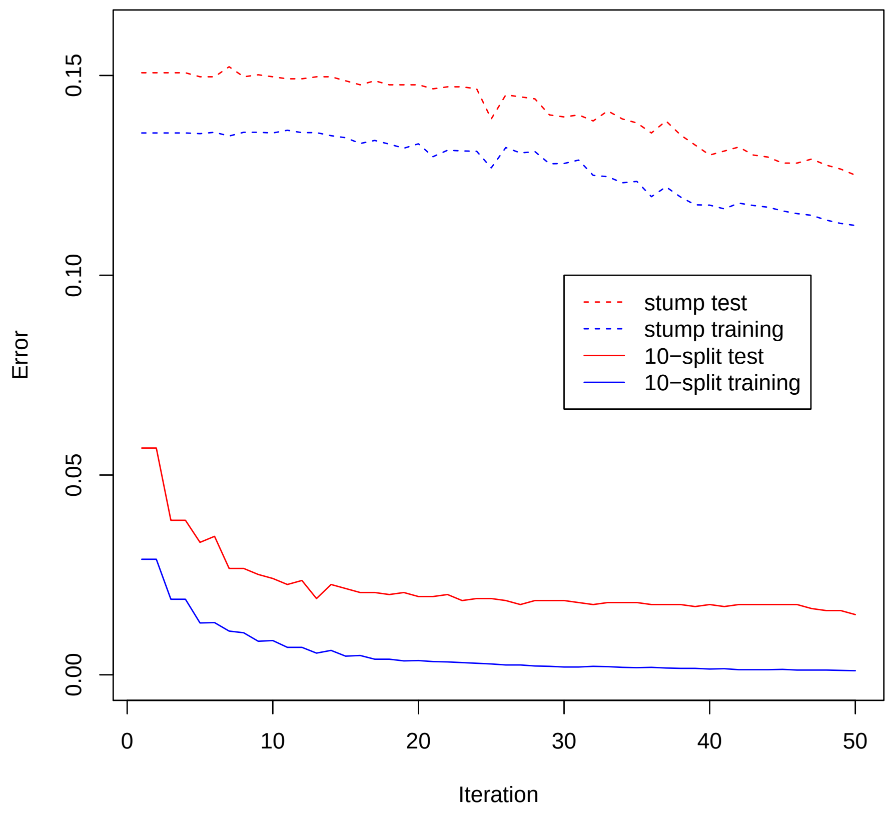

Figure 3: AdaBoost with trees

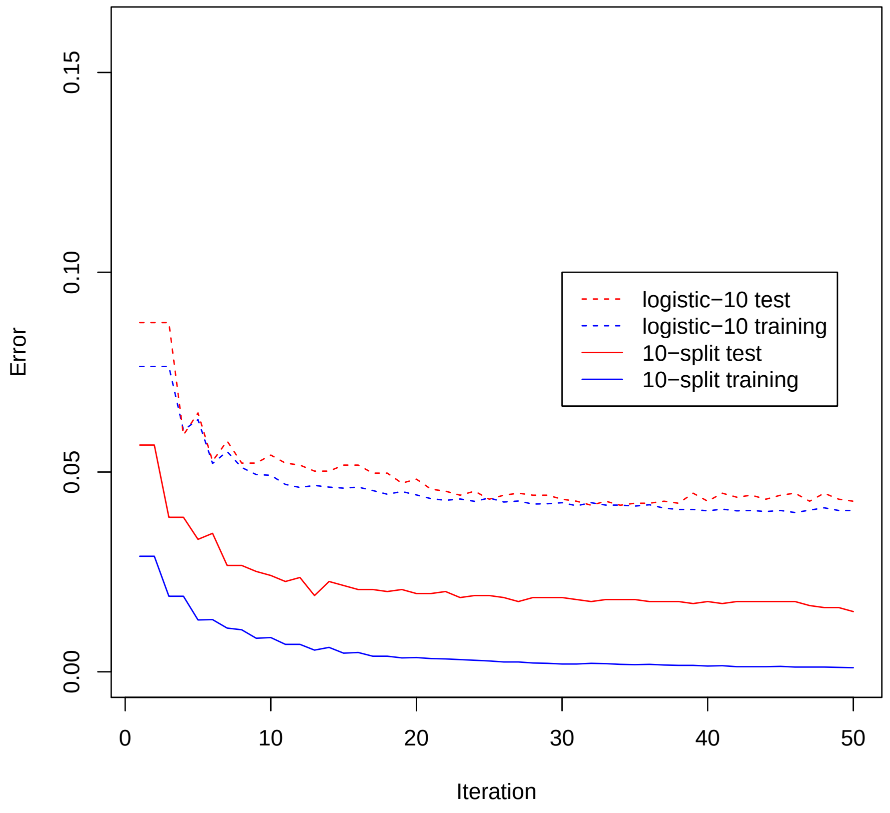

Figure 4: AdaBoost with 10 covariate logistic regression and 10 split trees

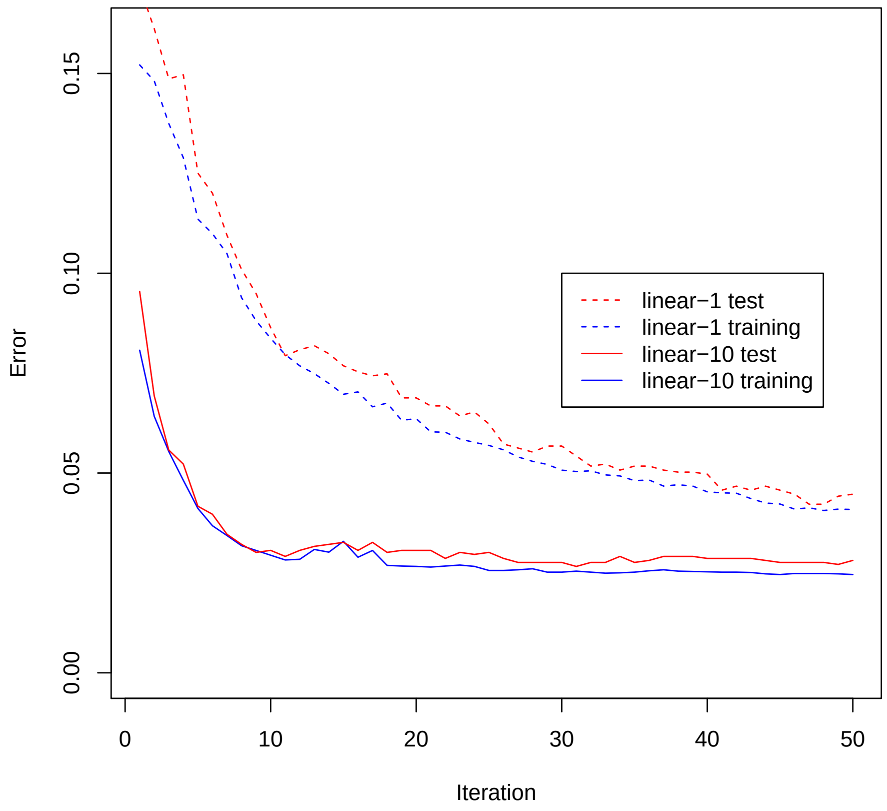

Figure 5: LogitBoost with linear regression

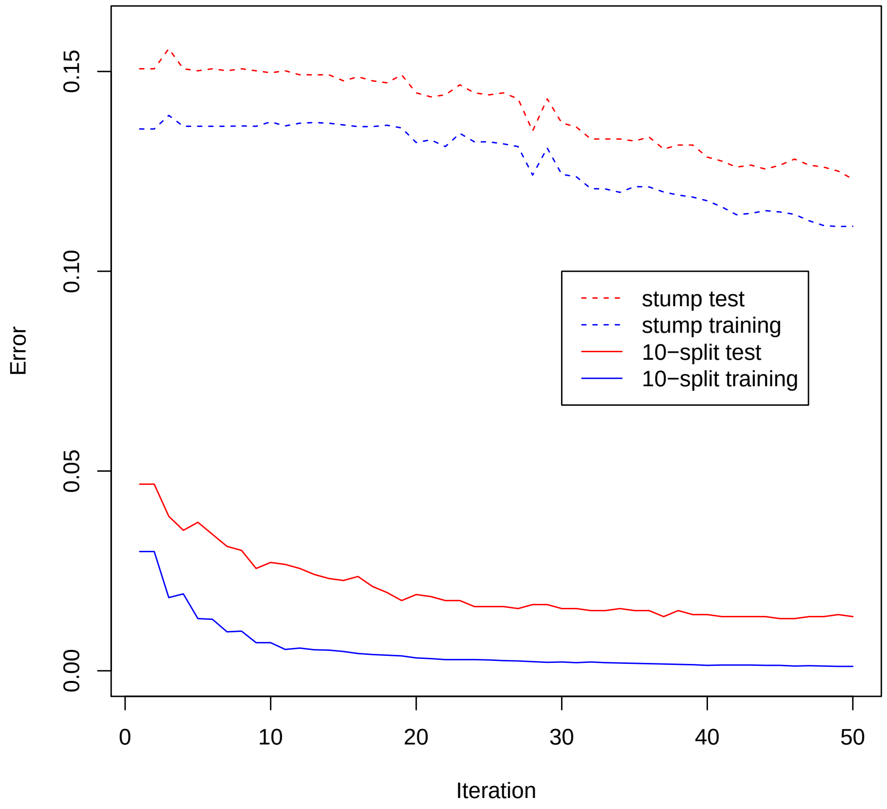

Figure 6: LogitBoost with trees

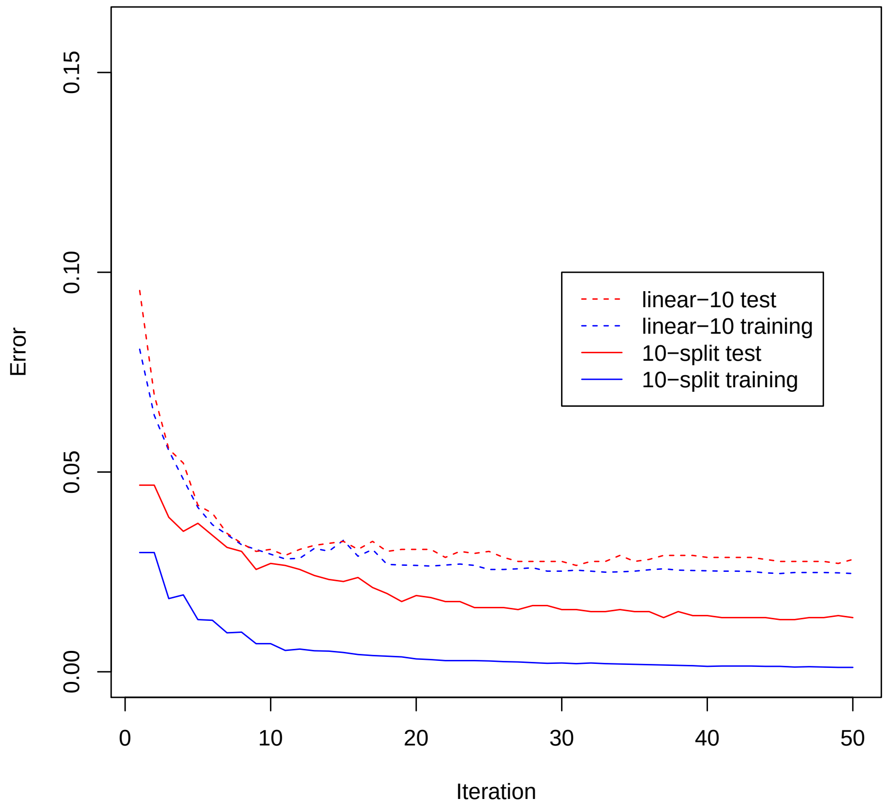

Figure 7: LogitBoost with 10-split trees and 10 linear regression

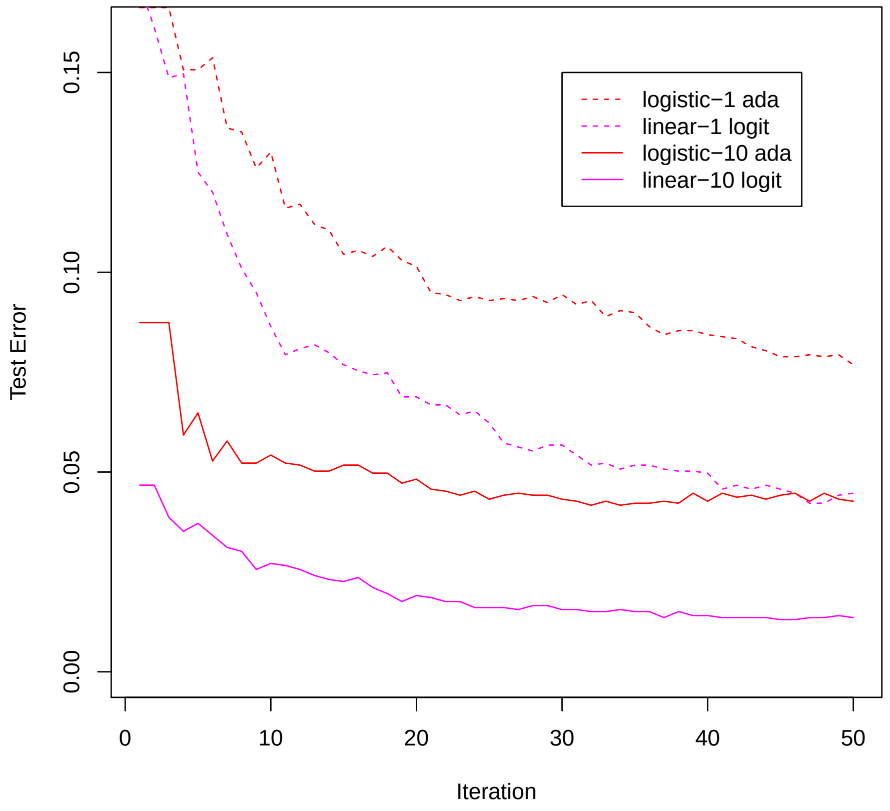

Figure 8: Test errors of AdaBoost and LogitBoost with regressions

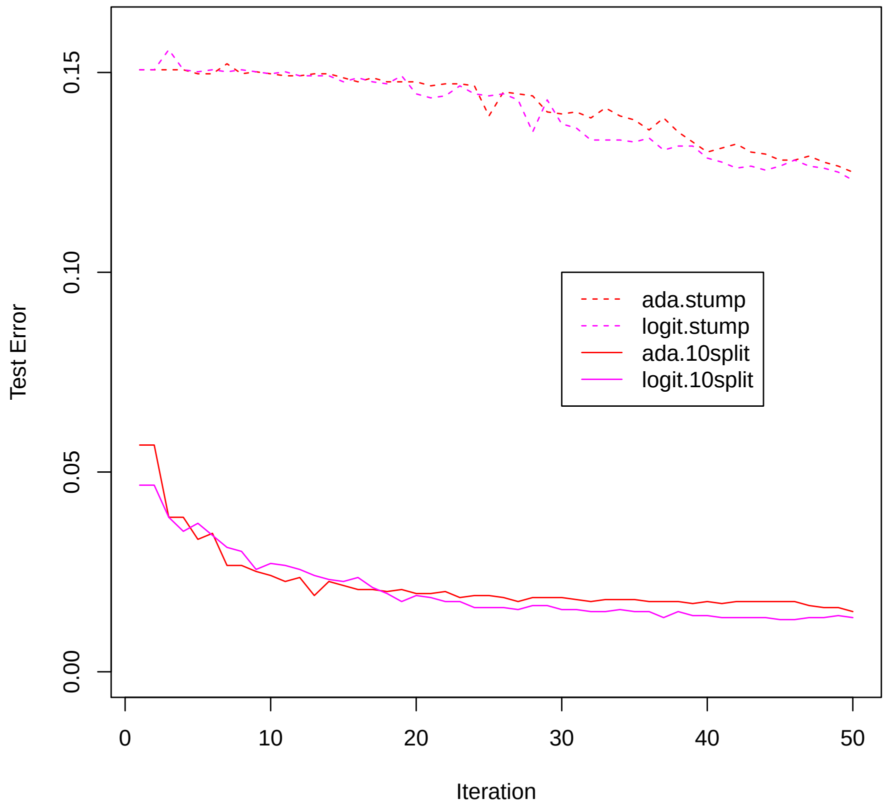

Figure 9: Test errors of AdaBoost and LogitBoost with trees

It seems tree with 10 splits with Adaboost or Logitboost don't differ much, and both are the best among all because of their least test errors.

2. Try the above, but with random forest instead, trying different combinations of `mtry` and number of bootstrap sample.

   Using the bigrf package in R, we fit random forests in parallel to the training data. The number of bootstrap samples ranged from 1 (a single unpruned tree) to 100, and we tried a few different values for the number of variables the random forest could split on: nsplitvar = 5, 10, 25, 50, 100, 200, 400, and 600. This was out of a total of 618 predictor variables, so 25 is close to the square root of that number and 600 is close to using all of them, a.k.a. bagging.

   Although the training error got slightly better (at least for smaller numbers of trees in Figure 10) as nsplitvar increased, what we really care about is the test error in Figure 11. As to be expected, the test error was the lowest when nsplitvar=25, so the recommendation to use the square root of the number of predictors seems warranted. The minimum test error (approx. 1.26 percent) was achieved using a bootstrap sample of around 60 trees, but remained relatively constant as the number of trees increased.

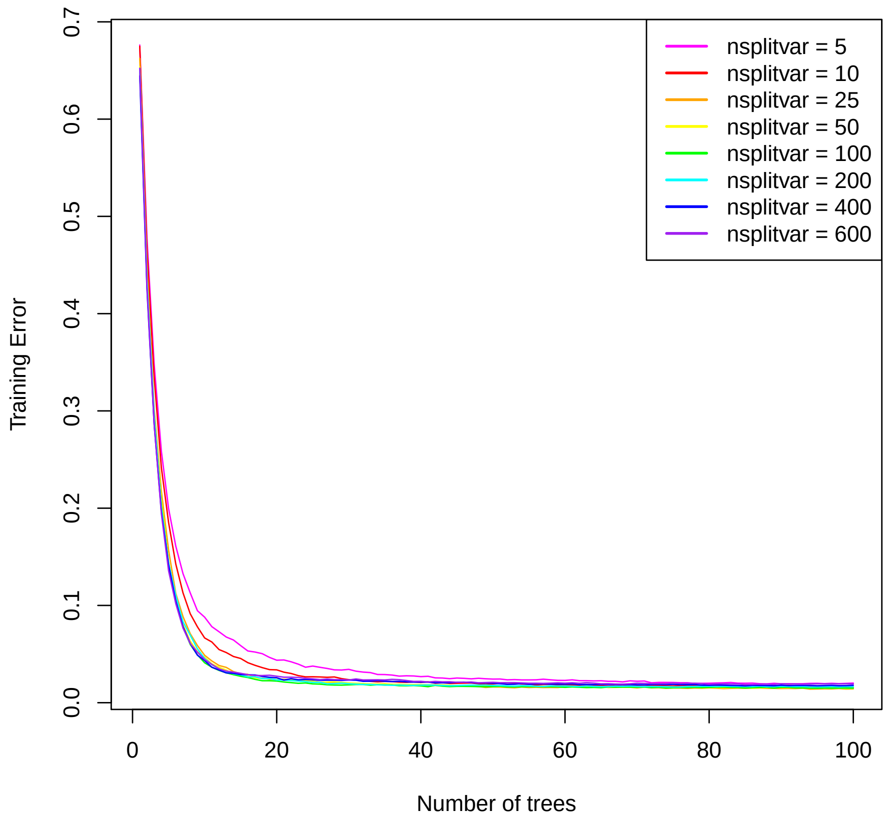

Figure 10: Random forest training

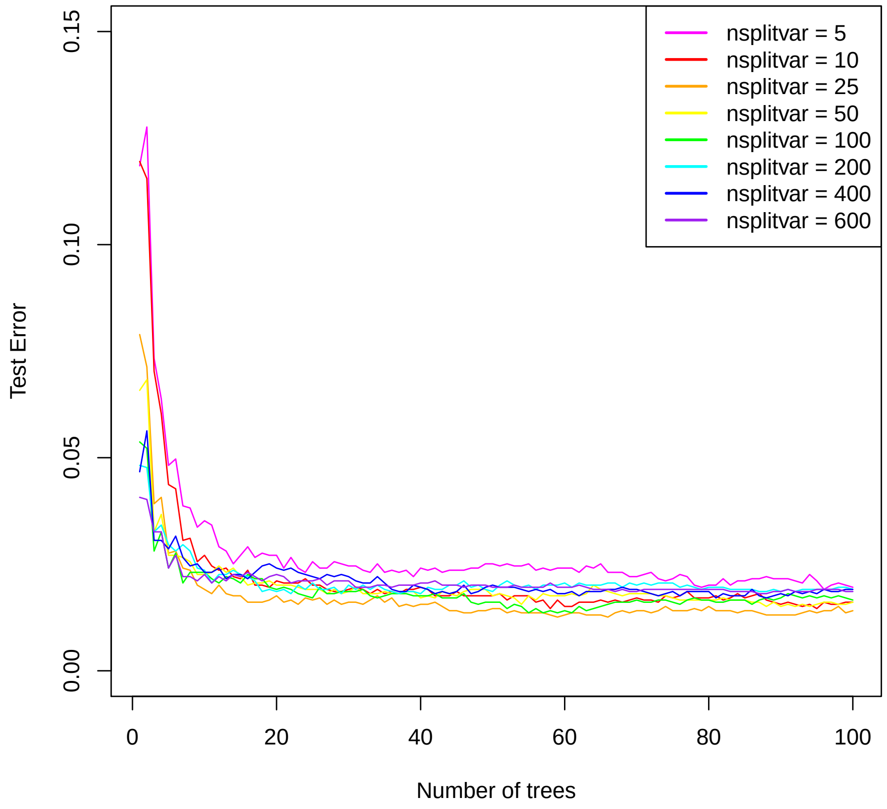

Figure 11: Random forest Test