Machine Learning - Block02 Assignment 01

Agustín Valencia

1. Ensemble Methods

The file spambase.csv contains information about the frequency of various words, characters, etc. for a total of 4601 e-mails. Furthermore, these e-mails have been classified as spams (spam = 1) or regular e-mails (spam = 0). You can find more information about these data at https://archive.ics.uci.edu/ml/datasets/Spambase.

Your task is to evaluate the performance of Adaboost classification trees and random forests on the spam data. Specifically, provide a plot showing the error rates when the number of trees considered are 10, 20, . . . , 100. To estimate the error rates, use 2/3 of the data for training and 1/3 as hold-out test data.

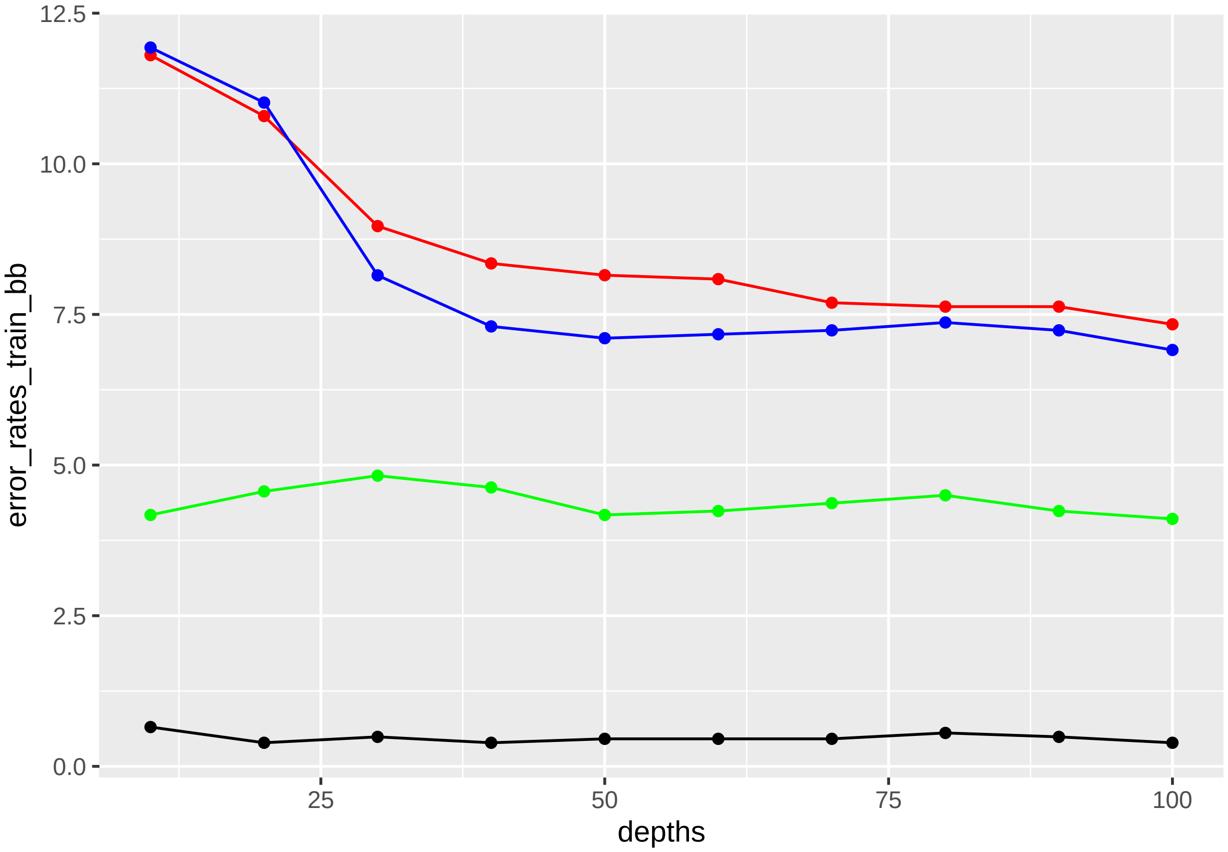
To learn Adaboost classification trees, use the function blackboost() of the R package mboost. Specify the loss function corresponding to Adaboost with the parameter family. To learn random forests, use the function randomForest of the R package randomForest. To load the data, you may want to use the following code:

```r
sp <- read.csv2("data/spambase.csv")
sp$Spam <- as.factor(sp$Spam)
```

Solution

For trees trained using adaboost we have:

```r
## Splitting data
set.seed(42)
n <- dim(sp)[1]
idxs <- sample(1:n, floor(2*n/3))
train <- sp[idxs,]
test <- sp[-idxs,]

get_missed <- function (true, predicted) {
    confussion <- table(true, predicted)
    #tn <- confussion[1,1]
    #tp <- confussion[2,2]
    fn <- confussion[1,2]
    fp <- confussion[2,1]
    total <- sum(confussion)
    #success <- (tp + tn) / total * 100
    miss <- (fp + fn) / total * 100
    return(miss)
}

# Training
nums <- seq(10,100,10)
formula <- Spam ~ .
error_rates_train_bb <- c()
error_rates_test_bb <- c()
error_rates_train_rf <- c()
error_rates_test_rf <- c()
depths <- c()
for (i in nums) {
    bb <- blackboost (
            Spam ~ .,
            data = train,
            family = AdaExp(),
            control = boost_control(mstop = i)
         )

    rf <- randomForest(
            Spam ~ .,
            data = train,
            ntree = i
    )

    predicted <- predict(bb, train, type = "class")
    miss <- get_missed(train$Spam, predicted)
    error_rates_train_bb <- append(error_rates_train_bb, miss)

    predicted <- predict(bb, test, type = "class")
    miss <- get_missed(test$Spam, predicted)
    error_rates_test_bb <- append(error_rates_test_bb, miss)

    predicted <- predict(rf, train, type = "class")
    miss <- get_missed(train$Spam, predicted)
    error_rates_train_rf <- append(error_rates_train_rf, miss)

    predicted <- predict(rf, test, type = "class")
    miss <- get_missed(test$Spam, predicted)
    error_rates_test_rf <- append(error_rates_test_rf, miss)

    depths <- append(depths, i)
}

p <- ggplot()
p <- p + geom_line(aes(x = depths, y = error_rates_train_bb), color = "red")
p <- p + geom_line(aes(x = depths, y = error_rates_test_bb), color = "blue")
p <- p + geom_line(aes(x = depths, y = error_rates_train_rf), color = "black")
p <- p + geom_line(aes(x = depths, y = error_rates_test_rf), color = "green")
p <- p + geom_point(aes(x = depths, y = error_rates_train_bb), color = "red")
p <- p + geom_point(aes(x = depths, y = error_rates_test_bb), color = "blue")
p <- p + geom_point(aes(x = depths, y = error_rates_train_rf), color = "black")
p <- p + geom_point(aes(x = depths, y = error_rates_test_rf), color = "green")

p
```


12.5
10.0
7.5
5.0
2.5
0.0
error_rates_train_bb
25
50
75
100
depths