Week 4 Project Write Up

- Author: “Emilie Worsham”
- Date: “12/16/2017”

Project Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data

The Following Data Sources were used for this project:

- Training Data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)
- Test Data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)
- All of the data came from this source, and we thank them for allowing us to use this data for our projects:

http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har).

To save space in the code & final prinout I loaded the following libraries to by RStudio:

```
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(caret)
library(randomForest)
library(gbm)
library(plyr)
```

Download and Clean the Training Data

```
## download the training dataset
download.file(url = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",
              destfile = "C:/Users/erobe/Desktop/Saved Items/Coursera/Practical_Machine_learnin
g/Practical_Machine_Learning/pml-training.csv")

## Load training dataset
training <- read.csv("C:/Users/erobe/Desktop/Saved Items/Coursera/Practical_Machine_learning/Pra
ctical_Machine_Learning/pml-training.csv", na.strings=c("NA","#DIV/0!",""))

# Download the testing data
download.file(url = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",
              destfile = "C:/Users/erobe/Desktop/Saved Items/Coursera/Practical_Machine_learnin
g/Practical_Machine_Learning/pml-testing.csv")

# Load the testing dataset
testing <- read.csv("C:/Users/erobe/Desktop/Saved Items/Coursera/Practical_Machine_learning/Prac
tical_Machine_Learning/pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
```

Cleaning Data

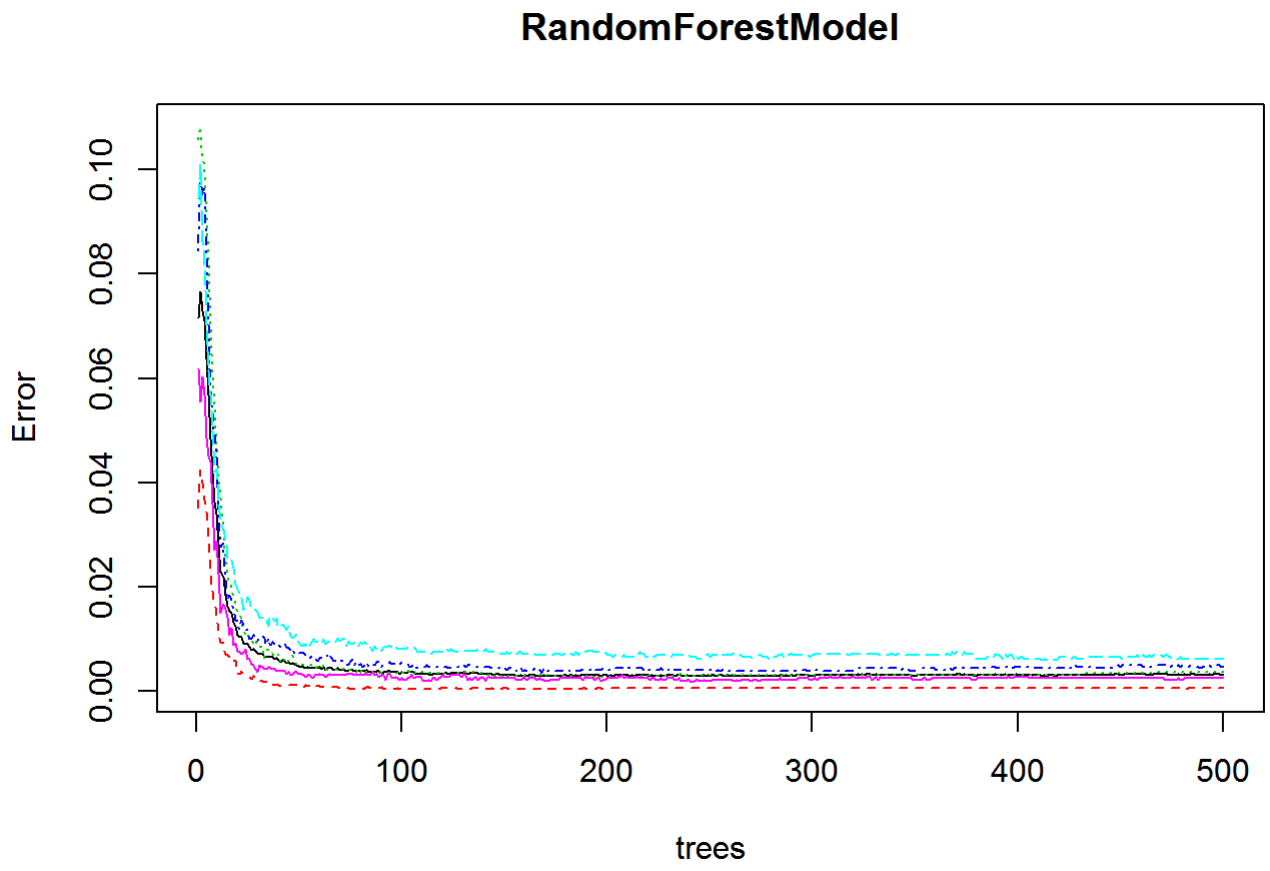
In this Section I will clean the data and explain the process as I do it. I am removing all columns that contain N/A (null) values and any columns that do not appear in the testing table.

```
features <- names(testing[,colSums(is.na(testing)) == 0])[8:59]

# Only use columns that appear in the testing cases.
training <- training[,c(features,"classe")]
testing <- testing[,c(features,"problem_id")]

dim(training); dim(testing);
```

[1] 19622 53 [1] 20 53

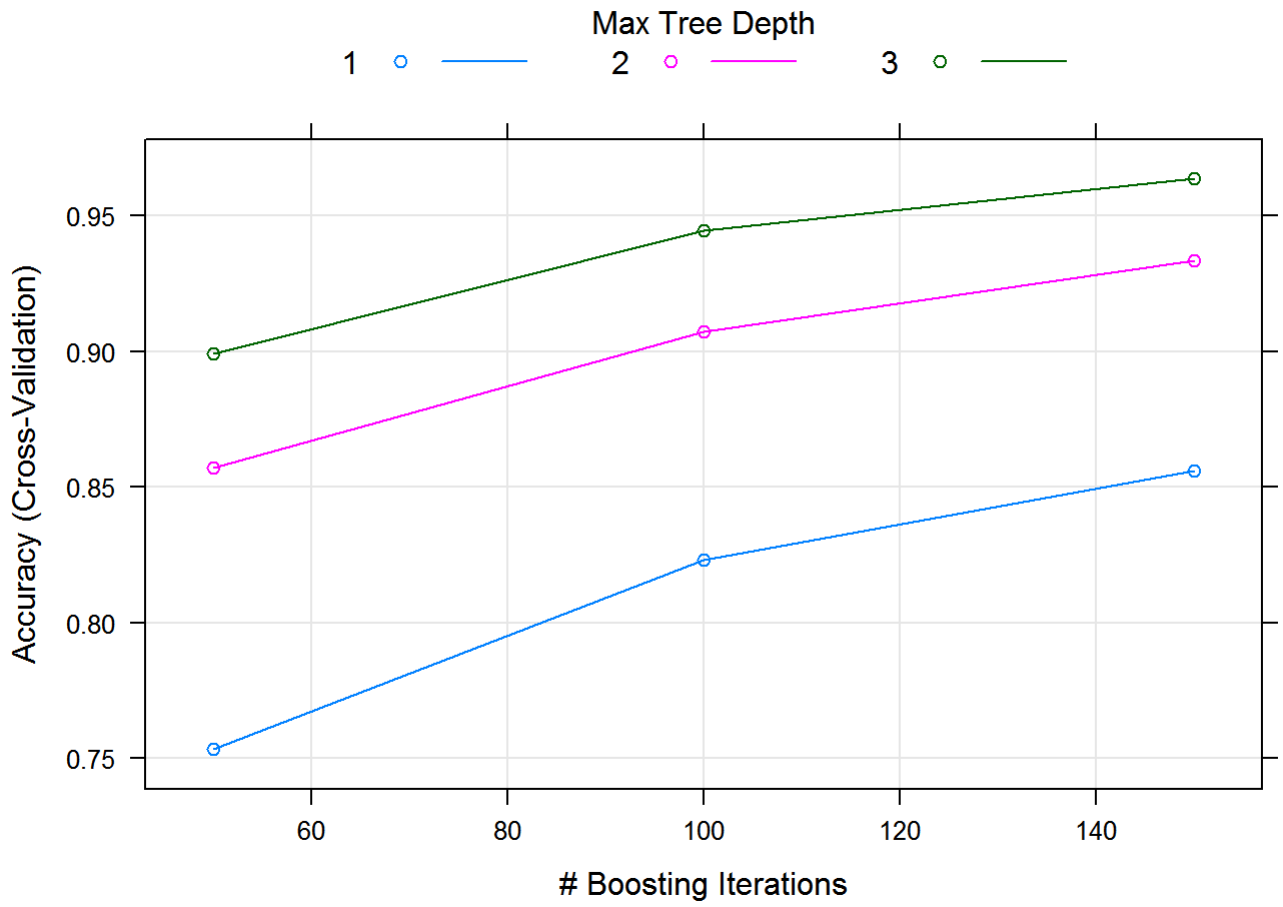
Setting up the Random Forest Model

Using the random forest model , the out of sample error should be small. I will be estimating the error using the 40% testing sample. We should expect an error estimate of < 3%.



Setting up the Boosting Model

```
## Stochastic Gradient Boosting
##
## 19622 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 17659, 17661, 17660, 17660, 17660, 17659, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7531848  0.6870539
##   1                  100      0.8230564  0.7760230
##   1                  150      0.8559786  0.8177106
##   2                   50      0.8572013  0.8190652
##   2                  100      0.9074003  0.8828267
##   2                  150      0.9334939  0.9158389
##   3                   50      0.8990931  0.8722651
##   3                  100      0.9445019  0.9297836
##   3                  150      0.9636637  0.9540274
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```



Now that the Models are set up now I will use them to predict the outcomes

Predicting with the Testing Data (pml-testing.csv)

Random Tree Prediction

```
predictRT <- predict(RandomForestModel, testing)

predictRT
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 B A B A A E D B A A B C B A E E A B B B Levels: A B C D E

** Boosting Model Prediction**

```
predictBoost <- predict(BoostingModel, testing)

predictBoost
```

[1] B A B A A E D B A A B C B A E E A B B B Levels: A B C D E

Submission File

As you can see from the Random Forest model outcome matrix it's about 99% accurate making it the more accurate of the two models.

```
pm_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pm_files(predictRT)
```