Practical Machine Learning - Prediction Assignment

Jayendra Shinde

December 11, 2016

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://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Project Goal

The goal of our project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. We also created a report describing how we built our model, how we used cross validation, and what we think the expected out of sample error is, and why we made the choices we did. We also use our prediction model to predict 20 different test cases.

Loading the necessary packages

```
library(caret)
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
library(randomForest)
library(knitr)
library(data.table)
set.seed(12345)
```

Loading the necessary datasets

```
url_train <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
url_test <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
data <-read.csv(url(url_train), na.strings=c("NA","#DIV/0!",""))
new_testing <- read.csv(url(url_test), na.strings=c("NA","#DIV/0!",""))
```

Cleaning the data

```
missingcols <- sapply(data, function(x) { any(is.na(x)) })
# replace data by keeping only those variables that don't have missing data
data <- data[ , !missingcols]; dim(data)
```

```
## [1] 19622    60
```

```
new_testing <- new_testing[ , !missingcols]; dim(new_testing)
```

```
## [1] 20 60
```

Subsetting the data

Cleaning Variables

```
inTrain <- caret::createDataPartition(y = data$classe, p = 0.7, list = FALSE)

# subset
training <- data[inTrain, ]; dim(training)
```

```
## [1] 13737    60
```

```
testing <- data[-inTrain, ]; dim(testing)
```

```
## [1] 5885   60
```

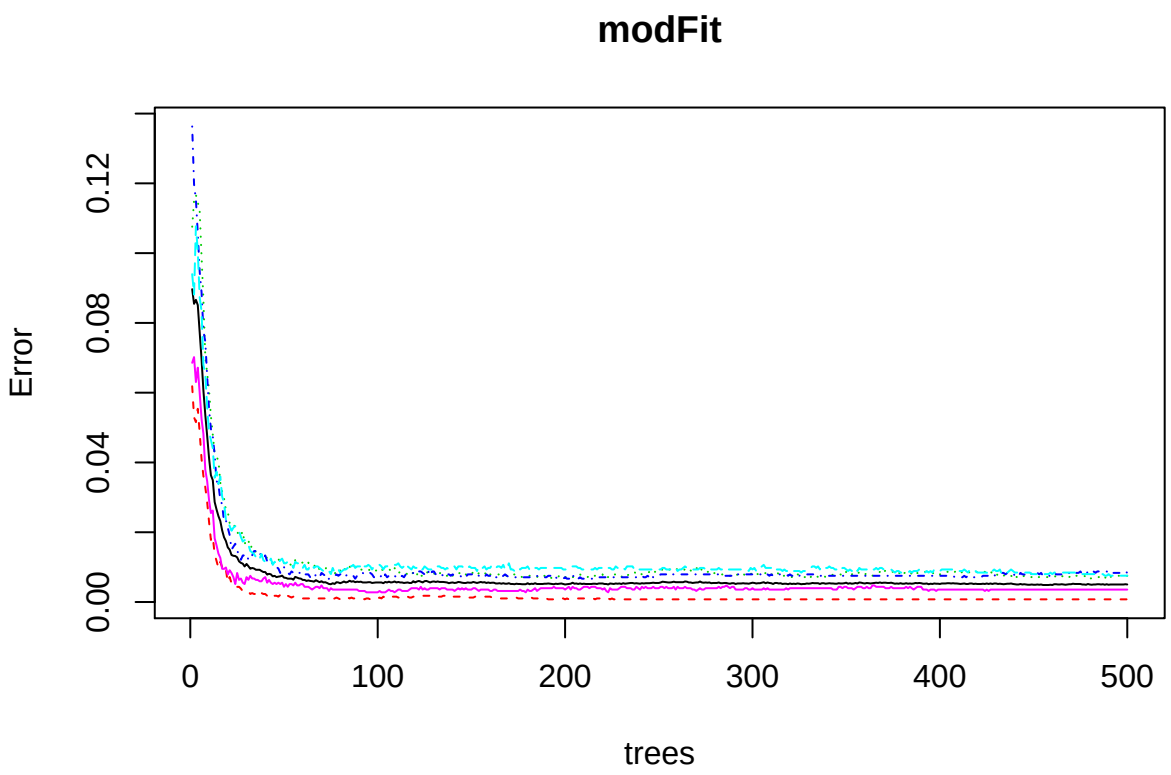
Using Random forest for prediciton

```
modFit <- randomForest(training$classe ~ ., data=training[,c(8:60)])
prediction <- predict(modFit, newdata = testing[,c(8:60)])
cmrf <- confusionMatrix(prediction, testing$classe)
cmrf
```

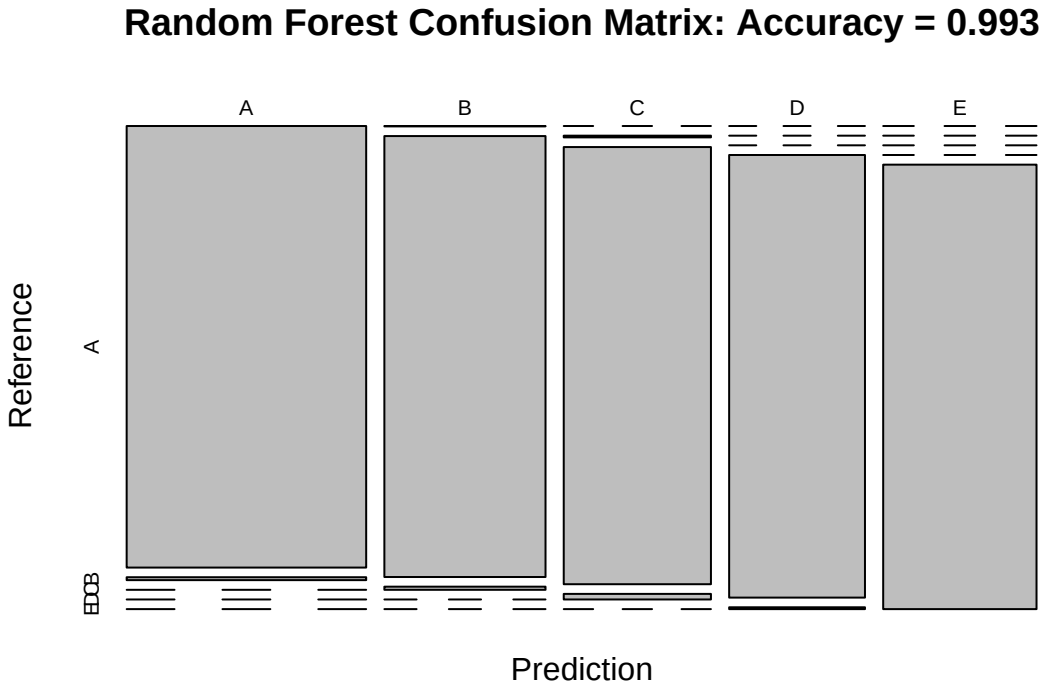
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673   11    0    0    0
##          B    1 1124    8    0    0
##          C    0    4 1018   13    0
##          D    0    0    0  951    4
##          E    0    0    0    0 1078
##
## Overall Statistics
##
```

```
##                Accuracy : 0.993
##                  95% CI : (0.9906, 0.995)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9912
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9868   0.9922   0.9865   0.9963
## Specificity            0.9974   0.9981   0.9965   0.9992   1.0000
## Pos Pred Value         0.9935   0.9921   0.9836   0.9958   1.0000
## Neg Pred Value         0.9998   0.9968   0.9984   0.9974   0.9992
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1910   0.1730   0.1616   0.1832
## Detection Prevalence   0.2862   0.1925   0.1759   0.1623   0.1832
## Balanced Accuracy      0.9984   0.9925   0.9944   0.9929   0.9982
```

```
plot(modFit)
```




```
plot(cmrf$table, main = paste("Random Forest Confusion Matrix: Accuracy =", round(cmrf$overall['Accuracy
```



Predicting the results for the test dataset

```
prediction_test <-predict(modFit, newdata = new_testing)
prediction_test
```

```
##  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
```