Prediction Assignment Writeup 1

Oscar Chamberlain

24 de junho de 2017

```
knitr::opts_chunk$set(echo = FALSE)


## Executive Summary

#An evaluation was done of a set of data from accelerometers on the belt, fo
rearm, arm, and dumbell of 6 participants, where they performed barbell lift
s correctly and incorrectly in 5 different ways. Due to the high number of f
eatures or parameter, we decided to apply the Random Forrest method. It was
obtained an accurated model although it is difficult to say what are the mor
e relevant parameters that defined the prediction.

## 1. Background
# Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now poss
ible to collect a large amount of data about personal activity relatively in
expensively. These type of devices are part of the quantified self movement
- a group of enthusiasts who take measurements about themselves regularly t
o improve their health, to find patterns in their behavior, or because they
are tech geeks. The goal will be to use data from accelerometers on the bel
t, forearm, arm, and dumbell of 6 participants. They were asked to perform b
arbell lifts correctly and incorrectly in 5 different ways. More informatio
n is available from the website here: http://groupware.les.inf.puc-rio.br/ha
r (see the section on the Weight Lifting Exercise Dataset).

## 2. Objective
# The goal of the project is to predict the manner in which they did the exe
rcise. This is the "classe" variable in the training set. Create a report de
scribing how the model was built, how cross validation was used, what is th
e expected out of sample error, and why you made the choices you did. You wi
ll also use your prediction model to predict 20 different test cases.

## Data input

# The variables without specific feature were removed, although they were us
ed to analyse the characteristic of the data, they will not be useful for th
e analyses. So the following related variables were removed: skewness, minim
um (min), max (maximum), amplitude, standard deviation (stddev), average (av
g) and var

##Data Input
setwd("~/Data/Machine Learning/data")
if (!file.exists("data")){dir.create("data")}
fileURL1<- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-trainin
g.csv" # The training data
fileURL2<- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.
csv" # the test data"
download.file(fileURL1, destfile="./data/training.csv")
download.file(fileURL2, destfile= "./data/test.csv")
train<-read.csv("training.csv")
test<-read.csv("test.csv")

# Although we will not include the "str" in the paper is very important the
```

```r
observation of the variables for the decision of what needed to be removed.
#str(train, vec.len=1,list.len=160, give.length=3)

library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.3
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 3.3.3
```

```
## Loading required package: ggplot2
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library('randomForest') # classification algorithm
```

```
## Warning: package 'randomForest' was built under R version 3.3.3
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

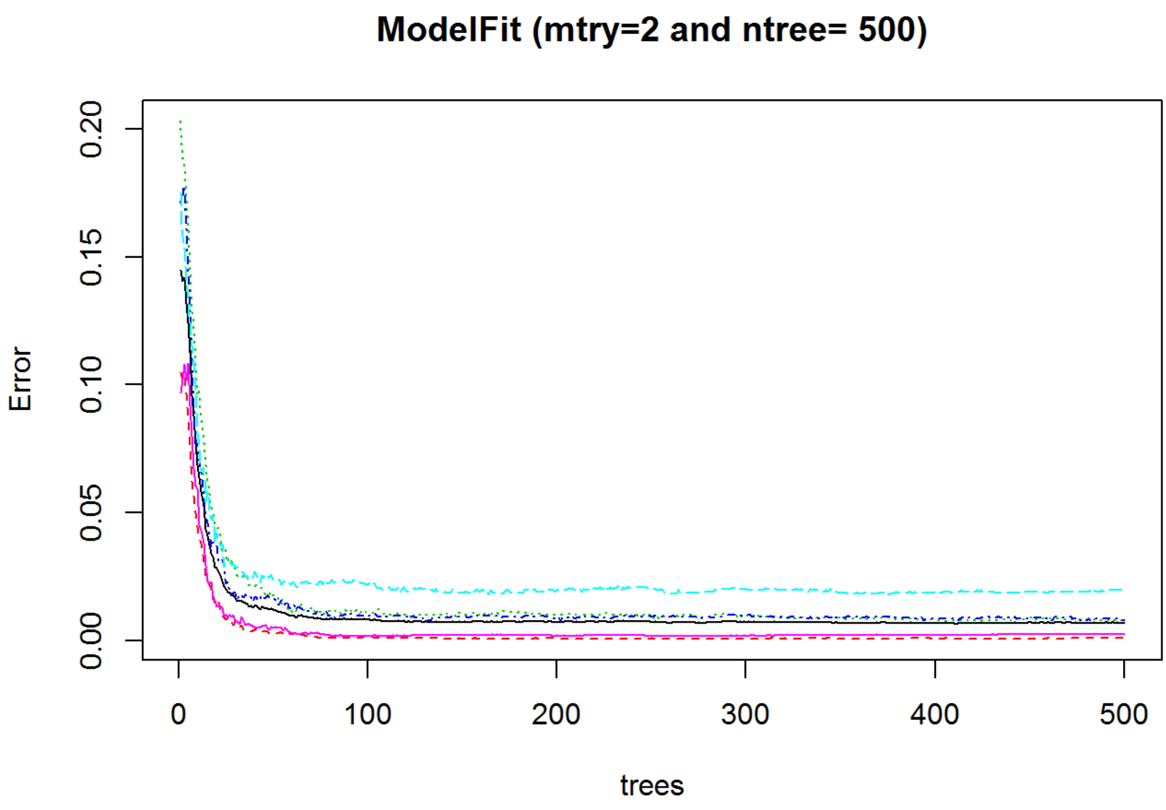
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## Variables removed

testA<-select(select(select(select(select(select(select(select(test,-contain
s("kurtosis")),-contains("skewness")),-contains("min")),-contains("max")),-c
ontains("amplitude")),-contains("stddev")),-contains("var")),-contains("av
g"))

trainA<-select(select(select(select(select(select(select(select(train,-conta
ins("kurtosis")),-contains("skewness")),-contains("min")),-contains("max")),
-contains("amplitude")),-contains("stddev")),-contains("var")),-contains("av
g"))

## Removing the variables related to general information like user_name or t
ime. As the general information about the people and date were not relevant
for the analyses, they wer also removed.
testA<-select(testA,-(X:num_window))
trainA<-select(trainA,-(X:num_window))


## 2. Cross Validation Approach:
#We followed the recomended steps:
#2.1 Use the training set
#2.2 Split it into training/test sets

set.seed(1234)
inTrain <- createDataPartition(y=trainA$classe,p=0.7, list=FALSE)
training <- trainA[inTrain,]
testing <- trainA[-inTrain,]


#2.3 Build a model on the training set
# It was choosen the Random Forest model using a mtry of 2 (Number of variab
les randomly sampled as candidates at each split.)

modelFit <- randomForest(classe ~., data = training, mtry=2)
modelFit
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training, mtry = 2)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.68%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3903    2    0    0    1 0.0007680492
## B   19 2636    3    0    0 0.0082768999
## C    0   17 2377    2    0 0.0079298831
## D    0    0   43 2208    1 0.0195381883
## E    0    0    1    5 2519 0.0023762376
```

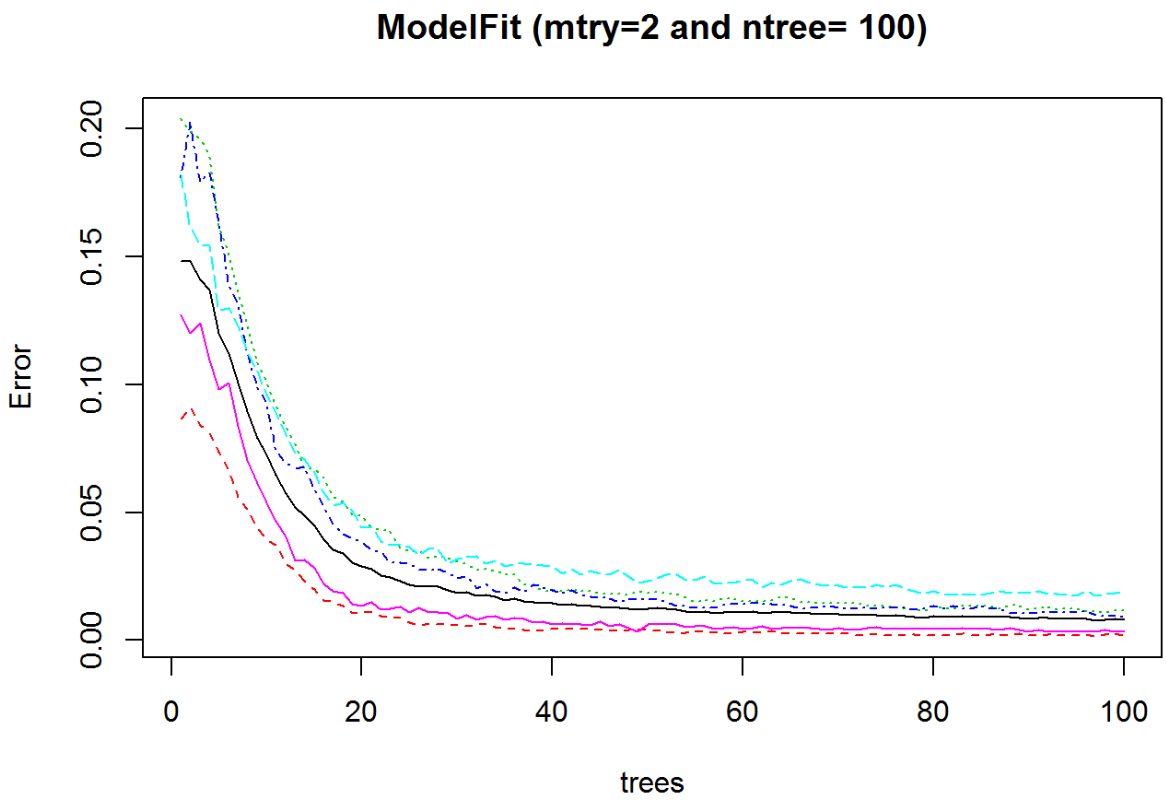
```
plot(modelFit,main="ModelFit (mtry=2 and ntree= 500)")
```




```
# We can observe that either with a number of tree of 100 we still obtained
a reasonably adjustment.
modelFit1 <- randomForest(classe ~., data = training, mtry=2, ntree=100)
modelFit1
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training, mtry = 2,      ntre
e = 100)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.82%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3898    4    2    1    1 0.002048131
## B   22 2627    9    0    0 0.011662904
## C    0   20 2374    2    0 0.009181970
## D    0    0   40 2210    2 0.018650089
## E    0    0    2    7 2516 0.003564356
```

```
plot(modelFit1,main="ModelFit (mtry=2 and ntree= 100)")
```




```
#2.4 Evaluate on the test set

pred <- predict(modelFit,testing)
tr<-testing$predRight <-pred==testing$classe
table(pred,testing$classe)
```

```
##
## pred    A    B    C    D    E
##    A 1672    9    0    0    0
##    B    2 1129    7    0    0
##    C    0    1 1018   11    1
##    D    0    0    1  952    1
##    E    0    0    0    1 1080
```

```
pred1 <- predict(modelFit1,testing)
tr1<-testing$predRight <-pred1==testing$classe
table(pred1,testing$classe)
```

```
##
## pred1    A    B    C    D    E
##     A 1673    8    0    0    0
##     B    1 1129    7    0    0
##     C    0    2 1018    9    0
##     D    0    0    1  955    1
##     E    0    0    0    0 1081
```

```
#2.5 Repeat and average the estimated errors

Accuracy<-sum(tr)/(sum(tr)+sum(tr==0))
Accuracy
```

```
## [1] 0.9942226
```

```
Accuracy1<-sum(tr1)/(sum(tr1)+sum(tr1==0))
Accuracy1
```

```
## [1] 0.9950722
```

```
# We can conclude that although the estimate error rate increased a little b
it from 0,63% to 0,87% the accuracy for the second model with ntree=100 wer
e similar 0.995

## 3. Course Project Prediction Quiz Portion
testC <- predict(modelFit,testA)
testC
```

```
##  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
```