Exercise Analysis

Rene Keller

5/21/2020

Introduction

This paper describes the analysis of the Exercise data. Two datasets were downloaded. First, the trainng set: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

and then the test Set

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

Loading Data

First we load the data and the different librarires (including caret).

```
trainDS<-read.csv("pml-training.csv")
testDS<-read.csv("pml-testing.csv")
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(randomForest)

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

Data Cleaning

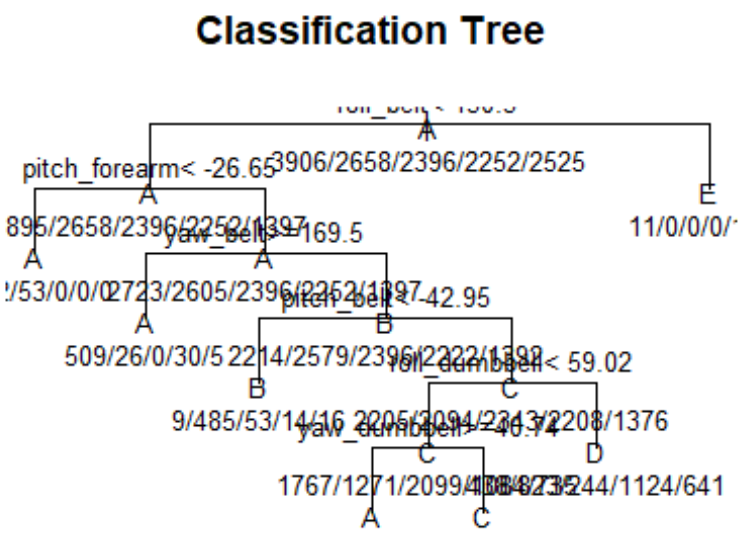
First we notice that there are a lot of columns that have all NAs as well as some columns that only show data for the beginning of the frame. Examining the test dataset, these are all NAs in the test dataset, hence they are removed. Finally, we create a Training and Test dataset (with a 70:30 split) from the original Test Dataset

```
testclean_final<-testDS[,c(8:11,46:49,84:86,102,122:124,140,160)]

traincleanint<-trainDS[,c(8:11,46:49,84:86,102,122:124,140,160)]
inTrain<-createDataPartition(y=traincleanint$classe,p=0.7,list=FALSE)

trainClean<-traincleanint[inTrain,]
testClean<-traincleanint[-inTrain,]
```

Random Tree

The first model that i fit will be a random tree. See below the resulting tree and its accuracy.

```
## create the model
modelTree<-train(classe~.,method="rpart", data=trainClean)
## Plot the Tree
plot(modelTree$finalModel,uniform=TRUE,main="Classification Tree")
text(modelTree$finalModel,use.n=TRUE,all=TRUE,cex=0.8)
```




```
##Create the prediction
predictTree<-predict(modelTree,testClean)
## Sumamry table
table(predictTree,testClean$classe)

##
## predictTree    A    B    C    D    E
##           A 1220  402  226  298  157
```

```
##            B    3  195   22    7    4
##            C  266  172  640  185  143
##            D  182  370  138  474  275
##            E    3    0    0    0  503

##Calculate accuracy
accuracyTree<-sum(predictTree==testClean$classe)/length(testClean$classe)
```

One can see that the accuracy of 0.5152082 is pretty low and the model does not do a great job predicting the test dataset.

Random Forests

Based on the disappointing results from the Random tree model, we now fit a random foorest model to the data.

```
## create the model
modelForest<-randomForest(classe~., data=trainClean)
## Plot the Tree
## plot(modelTree$finalModel,uniform=TRUE,main="Classification Tree")
##Create the prediction
predictForest<-predict(modelForest,testClean)
## Sumamry table
table(predictForest,testClean$classe)

##
## predictForest    A    B    C    D    E
##             A 1669   11    0    0    0
##             B    4 1111    3    0    2
##             C    0   16 1005    6    2
##             D    0    1   18  957    1
##             E    1    0    0    1 1077

##Calculate accuracy
accuracyForest<-sum(predictForest==testClean$classe)/length(testClean$classe)
```

One can see that the accuracy of 0.988785 is much better in predicting the testing Dataset.

Predicting the Final Test

We use the forest model from the previous sesction to now predict the test dataset. We also print out the results

```
##Create the prediction
predictForestFINAL<-predict(modelForest,testclean_final)

predictForestFINAL

##  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
```