Prediction Assignment Writeup

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.The goal of this project is to predict the manner in which they did the exercise.

Data Exploration

```
#loading required libraries
library(caret)
```

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

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

```
library(rattle)
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(randomForest)
```

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

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

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

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

```
#I have downloaded the file into my laptop and will read the files from there
training=read.csv('c:/Users/weey/Desktop/pml-training.csv',na.strings = c("N
A", ""))
testing=read.csv('c:/Users/weey/Desktop/pml-testing.csv',na.strings = c("NA",
""))
```

Having downloaded the file, I did a bit exploration in excel and determined that the first 7 columns were not reuqired and so I have removed it.

Furthermore, there were a lot of columns with NA results and with little information. These will hardly value add to our prediciton model. Therefore, I have also removed it.

```
Newtraining=training[,colSums(is.na(training))==0]
Newtraining=Newtraining[,c(8:ncol(Newtraining))]

Newtesting=testing[,colSums(is.na(testing))==0]
Newtesting=Newtesting[,c(8:ncol(Newtesting))]
```

I will find the best model through the use of our training data. I will further split the training data into train set and test set.

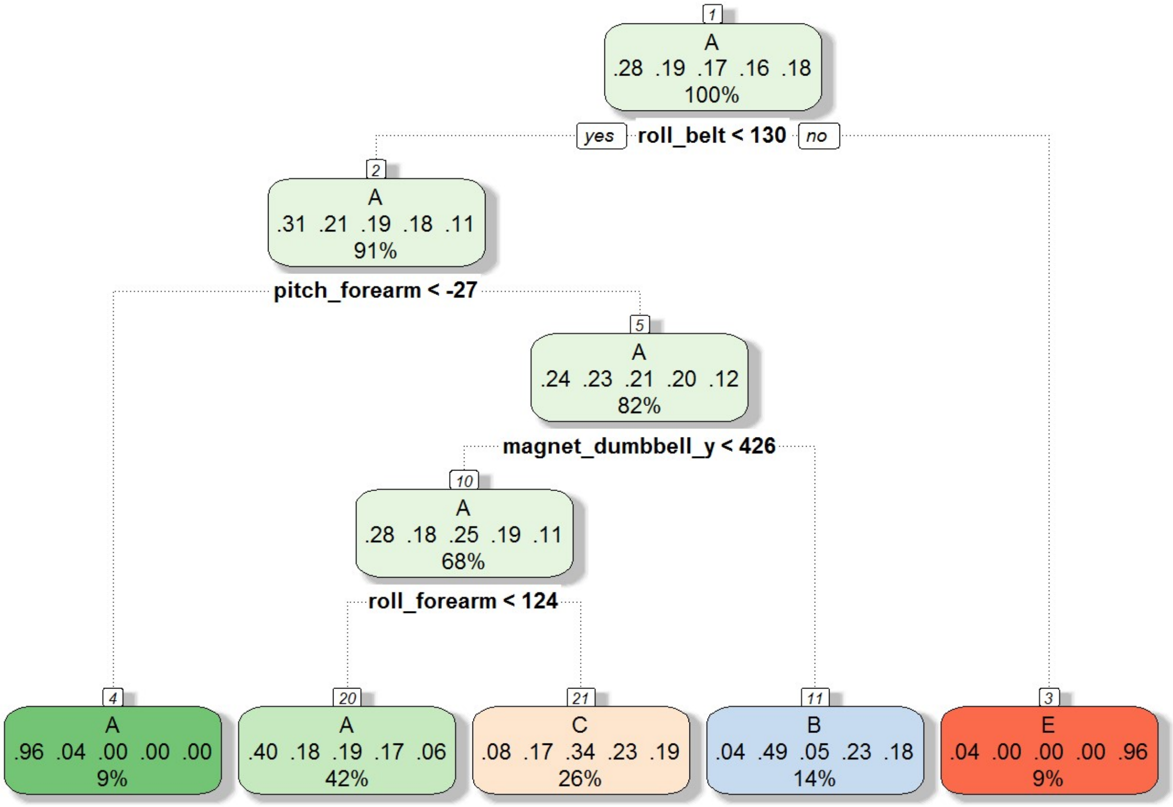
```
set.seed(345)
intrain=createDataPartition(Newtraining$classe,p=0.75,list=FALSE)
Ntrain=Newtraining[intrain,]
Ntest=Newtraining[-intrain,]
```

We will first try using the Decision Tree method

```
#train decisiontree
dttrain=train(classe~.,method='rpart',data=Ntrain)
```

```
## Loading required package: rpart
```

```
fancyRpartPlot(dttrain$finalModel)
```



Rattle 2017-Apr-01 21:38:28 weey

```
preddt=predict(dttrain,Ntest)
confusionMatrix(preddt,Ntest$classe)$overall[1]
```

```
##  Accuracy
## 0.4949021
```

With an accuracy of only 49%, it is definitely not satisfactory

Now let us try the Random Forest method instead

```
#train randomforest
set.seed(456)
rftrain=randomForest(classe~.,data=Ntrain)
predrf=predict(rftrain,Ntest)
confusionMatrix(predrf,Ntest$classe)$overall[1]
```

```
##  Accuracy
## 0.9955139
```

This gives us an accuracy of 99%! With such a high accuracy, we shall use this model on the actual test data.

```
predict(rftrain,Newtesting)
```

```
##  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
```

Below is the result

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