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 (see the section on the Weight Lifting Exercise Dataset).

```
library(caret)
```

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

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

```
library(ggplot2)
library(rpart)
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(RColorBrewer)
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:rattle':
##
##     importance
```

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

```
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
library(plyr)
```

```
download.file(url = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",
              destfile = "./pml-training.csv", method = "curl")
training_data <- read.csv("./pml-training.csv", na.strings=c("NA","#DIV/0!",""))
download.file(url = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",
              destfile = "./pml-testing.csv", method = "curl")
testing_data <- read.csv("./pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
```

Cleaning the data

remove all columns contain NA and remove features that are not in the testing dataset.

```
 NA_trainingCount = sapply(1:dim(training_data)[2],function(x)sum(is.na(training_data[,x])))
    NA_traininglist = which(NA_trainingCount>0)
    training_data = training_data[,-NA_traininglist]
    training_data = training_data[,-c(1:7)]

    NA_testingCount = sapply(1:dim(testing_data)[2],function(x)sum(is.na(testing_data[,x])))
    NA_testinglist = which(NA_testingCount>0)
    testing_data = testing_data[,-NA_testinglist]
    testing_data = testing_data[,-c(1:7)]
    dim(training_data); dim(testing_data)
```

```
## [1] 19622    53
```

```
## [1] 20 53
```

Data partition

```
set.seed(12345)

inTrain <- createDataPartition(training_data$classe, p=0.6, list=FALSE)
training <- training_data[inTrain,]
testing <- training_data[-inTrain,]

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

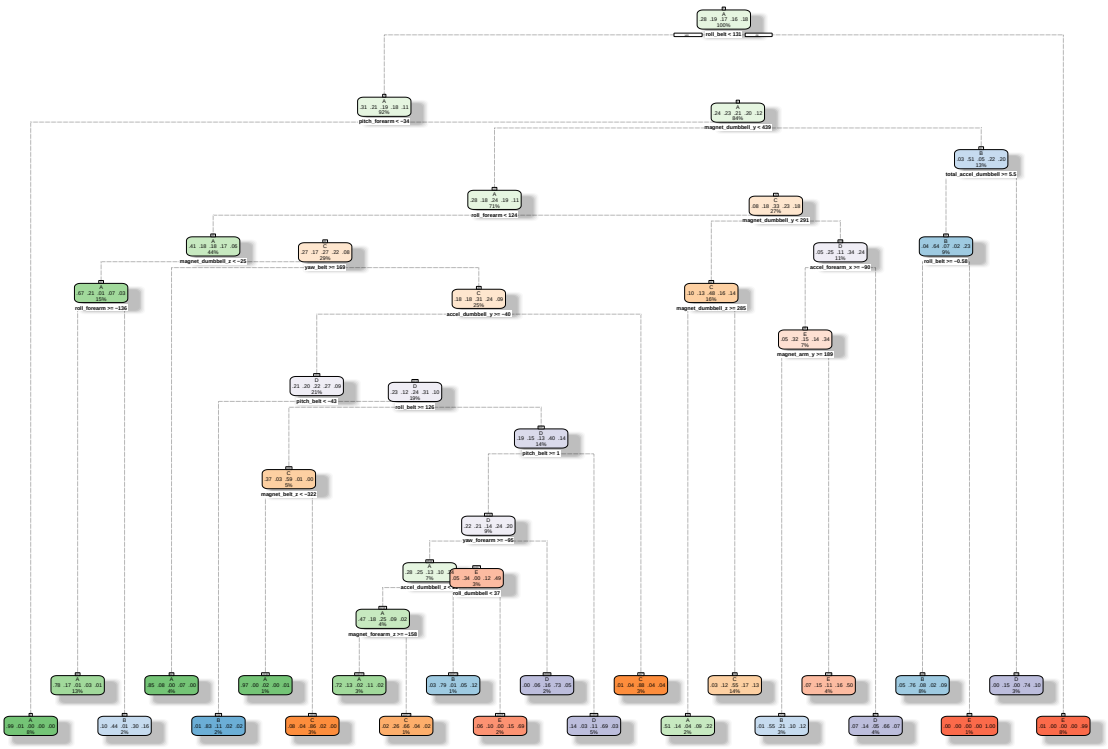
```
## [1] 11776    53
```

```
## [1] 7846   53
```

Decision Tree Model

```
modFit_DT <- rpart(classe ~ ., data = training, method="class")
fancyRpartPlot(modFit_DT)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2019−Oct−09 10:11:24 Connie

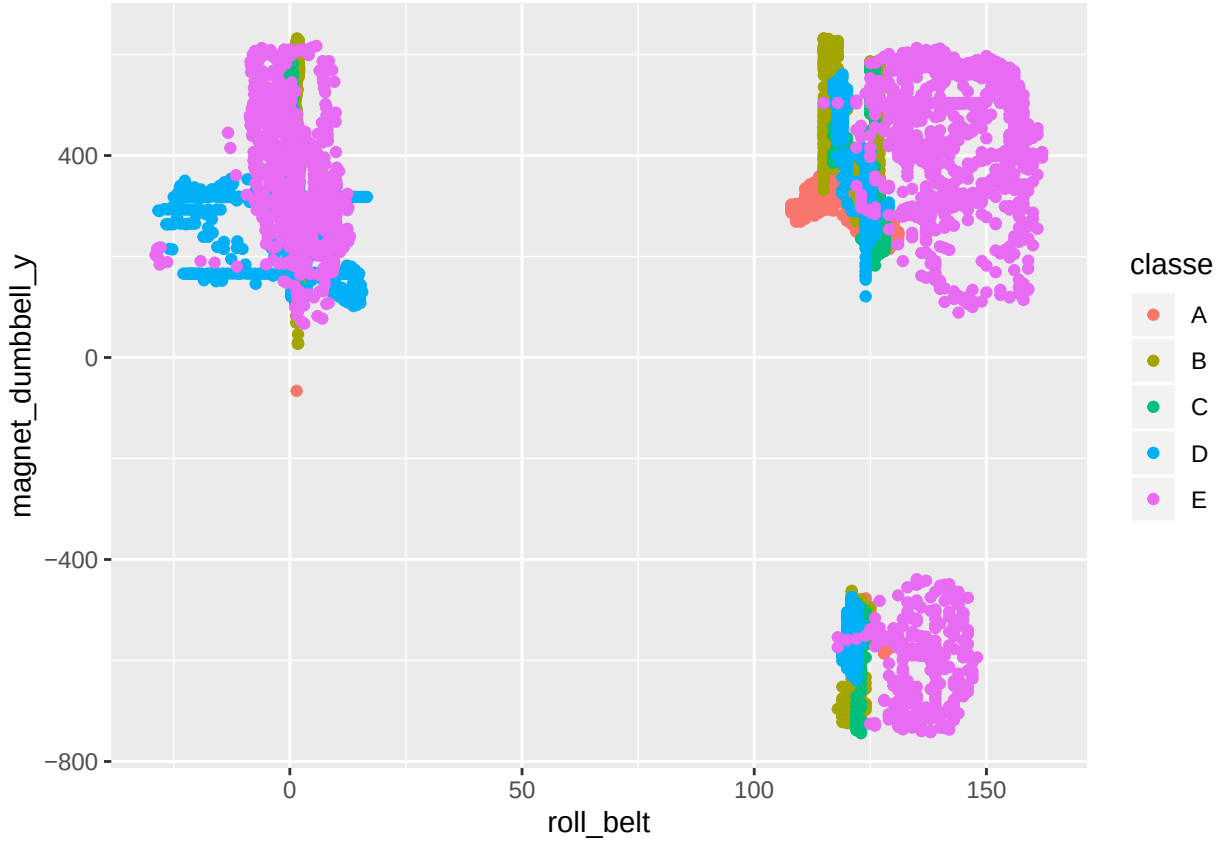
```
set.seed(12345)

prediction1 <- predict(modFit_DT, newdata=testing, type = "class")
confusionMatrix(prediction1, testing$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1995  246   49   83   51
##          B   75  890  111  119  115
##          C   44  198 1094  153  143
##          D   74  112   79  840   94
##          E   44   72   35   91 1039
##
## Overall Statistics
##
##                Accuracy : 0.7466
##                  95% CI : (0.7368, 0.7562)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6787
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
```

```
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8938   0.5863   0.7997   0.6532   0.7205
## Specificity            0.9236   0.9336   0.9169   0.9453   0.9622
## Pos Pred Value         0.8230   0.6794   0.6703   0.7006   0.8111
## Neg Pred Value         0.9563   0.9039   0.9559   0.9329   0.9386
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2543   0.1134   0.1394   0.1071   0.1324
## Detection Prevalence   0.3089   0.1670   0.2080   0.1528   0.1633
## Balanced Accuracy      0.9087   0.7600   0.8583   0.7992   0.8414
```

Random Forest Model

```
modfit_RF=randomForest(classe~., data=training, method='class')
prediction2 = predict(modfit_RF,testing,type='class')
qplot(roll_belt, magnet_dumbbell_y, colour=classe, data=training)
```




```
confusionMatrix(prediction2, testing$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2229    5    0    0    0
##          B    3 1513    6    0    0
##          C    0    0 1357    5    2
##          D    0    0    5 1281    6
```

```
##          E    0    0    0    0 1434
##
## Overall Statistics
##
##                Accuracy : 0.9959
##                  95% CI : (0.9942, 0.9972)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9948
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9987   0.9967   0.9920   0.9961   0.9945
## Specificity            0.9991   0.9986   0.9989   0.9983   1.0000
## Pos Pred Value         0.9978   0.9941   0.9949   0.9915   1.0000
## Neg Pred Value         0.9995   0.9992   0.9983   0.9992   0.9988
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2841   0.1928   0.1730   0.1633   0.1828
## Detection Prevalence   0.2847   0.1940   0.1738   0.1647   0.1828
## Balanced Accuracy      0.9989   0.9976   0.9954   0.9972   0.9972
```

Conclusion

Decision tree model is about 75% accuracy while random forest model has 99% accuracy.