Practical Machine Learing Final Prject

August 16, 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.

The aim of this report was to use data from accelerometers placed on the belt, forearm, arm, and dumbell of six participants to predict how well they were doing the exercise in terms of the classification in the data. More information is available from the website: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Loading Data and Processing

The following libraries are used for entire data processing and anlysis.

```r
library(caret)
library(corrplot)
library(rpart)
library(rpart.plot)
library(ggplot2)
library(knitr)
library(randomForest)
```

Setting the seed to reproduce the result.

```r
set.seed(1234)
```

Data

```r
# Keep in the working directory using these commands and download data.
if (!file.exists("data")) {dir.create("data")}

# file URL and destination file
fileUrl1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
destfile1 <- "./data/pml-training.csv"
fileUrl2 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
destfile2 <- "./data/pml-testing.csv"

# Downloading file including date and time.
download.file(fileUrl1, destfile = destfile1)
download.file(fileUrl2, destfile = destfile2)
dateDownloaded <- date()
```

Loading Train and Test Data Set.

```
# Train set
training_set <- read.csv("./data/pml-training.csv", na.strings= c("NA","","#DIV/0!"))
# Test set
testing_set <- read.csv("./data/pml-testing.csv", na.strings= c("NA","","#DIV/0!"))
```

```
# Deleting the column with all missing values.
training_set <- training_set[, colSums(is.na(training_set)) == 0]
testing_set <- testing_set[, colSums(is.na(testing_set)) == 0]
```

Removing all variables that are just identifier and are irrevelent for our project.

```
training <- training_set[, -c(1:7)]
testing <- testing_set[, -c(1:7)]
```

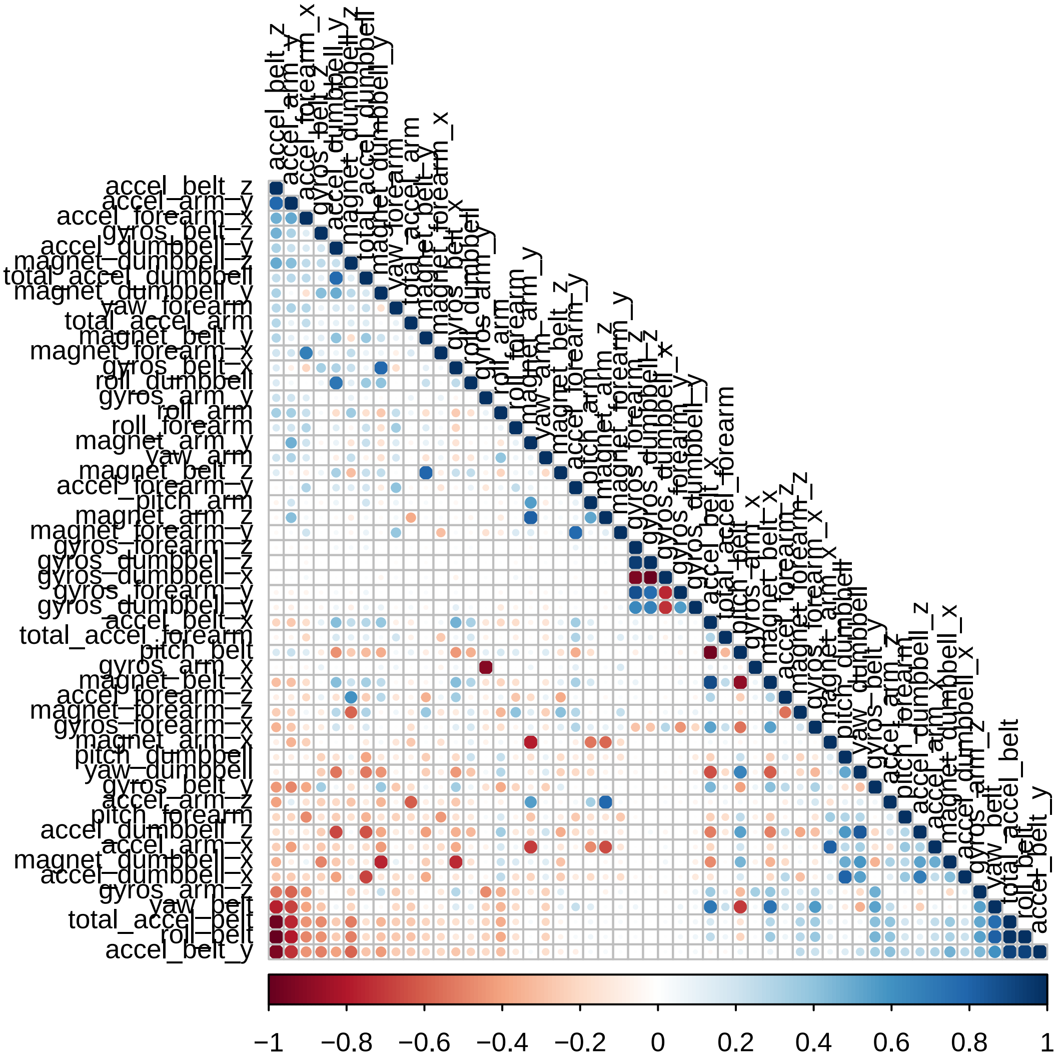
The training data set is divided into two subset, subtrain and subtest having 75 and 25 percent data respectively, to perform the cross validation.

```
sampling <- createDataPartition(y = training$classe, p = 0.75, list = FALSE)
subtrain <- training[sampling,]
subtest <- training[-sampling,]
```

Exploratory analysis and Model Development

We want to produce a correlation plot to see the variables relationship with each other.

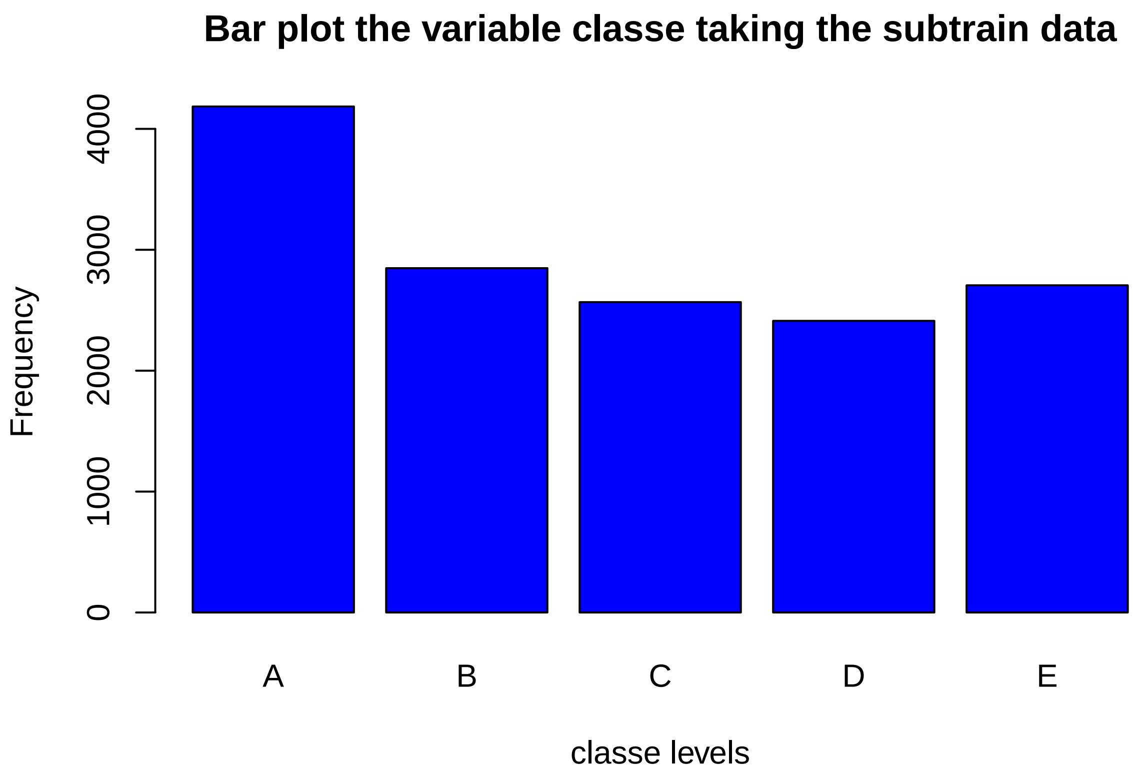
```
# Plotting a correlation matrix
correlMatrix <- cor(subtrain[, -length(subtrain)])
corrplot(correlMatrix, order = "FPC", method = "circle", type = "lower", tl.cex = 0.8,  tl.col = rgb(0,
```



Dark red and blue colors indicate a highly correlated variables in this plot and there is not much concern on those, so include all the variables into the model.

Now, let's see the frequency of output variable "classe" in the data by plotting bar plot.

```
plot(subtrain$classe, col = "blue", main = "Bar plot the variable classe taking the subtrain data", xlab
```

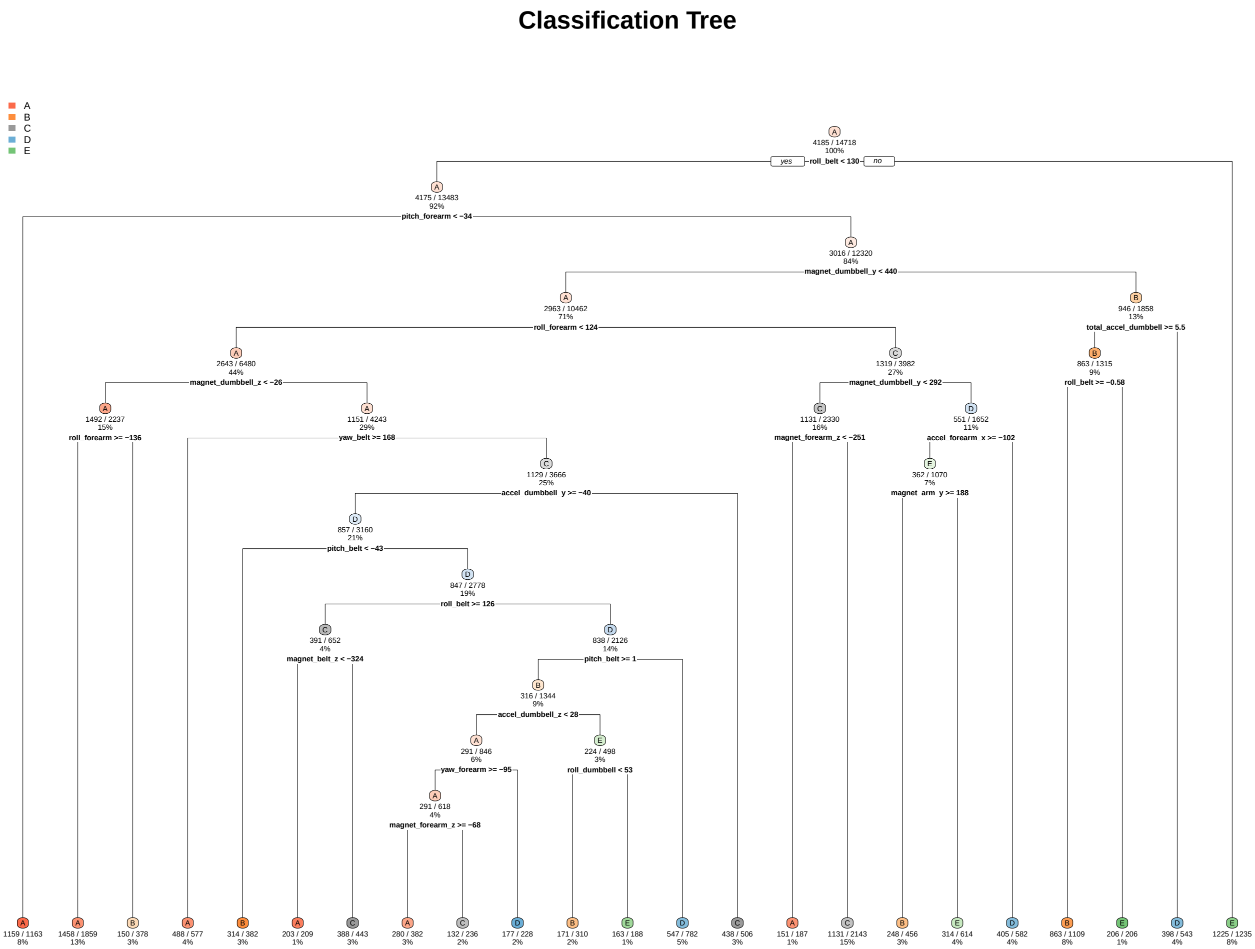


The graph shows that the frequency for each level is in the same order beside for level A. Level A has higher frequency.

Prediction Model:Decision tree

Let us develop a model using decision tree, make prediction and plot the decision tree.

```
model1 <- rpart(classe ~. , data = subtrain, method = "class")
prediction1 <- predict(model1, subtest, type = "class")
rpart.plot(model1, main = "Classification Tree", extra = 102, under = TRUE, faclen = 0)
```



Confusion matrix using the subtest data result.

```
confusionMatrix(prediction1, subtest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1235  157   16   50   20
##          B   55  568   73   80  102
##          C   44  125  690  118  116
##          D   41   64   50  508   38
##          E   20   35   26   48  625
##
## Overall Statistics
##
##                Accuracy : 0.7394
##                  95% CI : (0.7269, 0.7516)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6697
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8853   0.5985   0.8070   0.6318   0.6937
## Specificity            0.9307   0.9216   0.9005   0.9529   0.9678
## Pos Pred Value         0.8356   0.6469   0.6313   0.7247   0.8289
## Neg Pred Value         0.9533   0.9054   0.9567   0.9296   0.9335
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2518   0.1158   0.1407   0.1036   0.1274
## Detection Prevalence   0.3014   0.1790   0.2229   0.1429   0.1538
## Balanced Accuracy      0.9080   0.7601   0.8537   0.7924   0.8307
```

Prediction Model: Random forest

```r
model2 <- randomForest(classe ~. , data = subtrain , method = "class")
prediction2 <- predict(model2, subtest, type = "class")
confusionMatrix(prediction2, subtest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    3    0    0    0
##          B    1  944   10    0    0
##          C    0    2  843    6    0
##          D    0    0    2  798    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 0.9951
##                  95% CI : (0.9927, 0.9969)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9938
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9993   0.9947   0.9860   0.9925   1.0000
## Specificity            0.9991   0.9972   0.9980   0.9995   1.0000
## Pos Pred Value         0.9979   0.9885   0.9906   0.9975   1.0000
## Neg Pred Value         0.9997   0.9987   0.9970   0.9985   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2843   0.1925   0.1719   0.1627   0.1837
## Detection Prevalence   0.2849   0.1947   0.1735   0.1631   0.1837
## Balanced Accuracy      0.9992   0.9960   0.9920   0.9960   1.0000
```

Conclusion:

Model based on Random Forest perform better than Decision Tree. Accuracy for RF model has 0.9992 (95% CI:(0.9927,0.9969)), compared to 0.739 (95% CL:(0.7269,0.7516)) from DT model. Thus model based on Random Forest is more better, and is choosen. The accuracy of the model is 0.9992. The expected out-of-sample error is 0.005. With above accuracy that we get from cross-validation data, we can now confidently apply onto the real test data.

```r
predictfinal <- predict(model2, testing, type = "class")
predictfinal
```

```
##  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
```

Submission

```r
# Write file for submission
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i, ".txt")
    write.table(x[i], file = filename, quote = FALSE, row.names = FALSE, col.names = FALSE)
  }
}
pml_write_files(predictfinal)
```

Reference

[1] Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.