Practical Machine Learning

Anteneh

July 14, 2015

First we load the necessary packages and set the seed in order to get reproducible results.

```
library(AppliedPredictiveModeling)
library(caret)
```

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

```
library(rattle)
```

```
## Loading required package: RGtk2
## Rattle: A free graphical interface for data mining with R.
## Version 3.5.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart.plot)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
#Downloading the dataset
#fileURL1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
#download.file(fileURL1, destfile = "pml-training.csv", method="curl")
#fileURL2 <- " https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
#download.file(fileURL2, destfile = "pml-testing.csv", method="curl")
```

Our data may have NA, blank and #DIV/0!. To get rid of these we defined vector of na.strings and replace by NA. Because both data sets contain columns with all missing values, we will delete these inoredr to get clean data.

```
## [1] 19622   160
```

```
## [1]  20 160
```

We have checked the dimension of the new data set and then delete columns with missing values.

The first 7 columns such as user_name, raw_timestamp_part_1, raw_timestamp_part_2 cvtd_timestamp, new_window, and num_window are unnecessary for predicting our project, we delete all these variables.

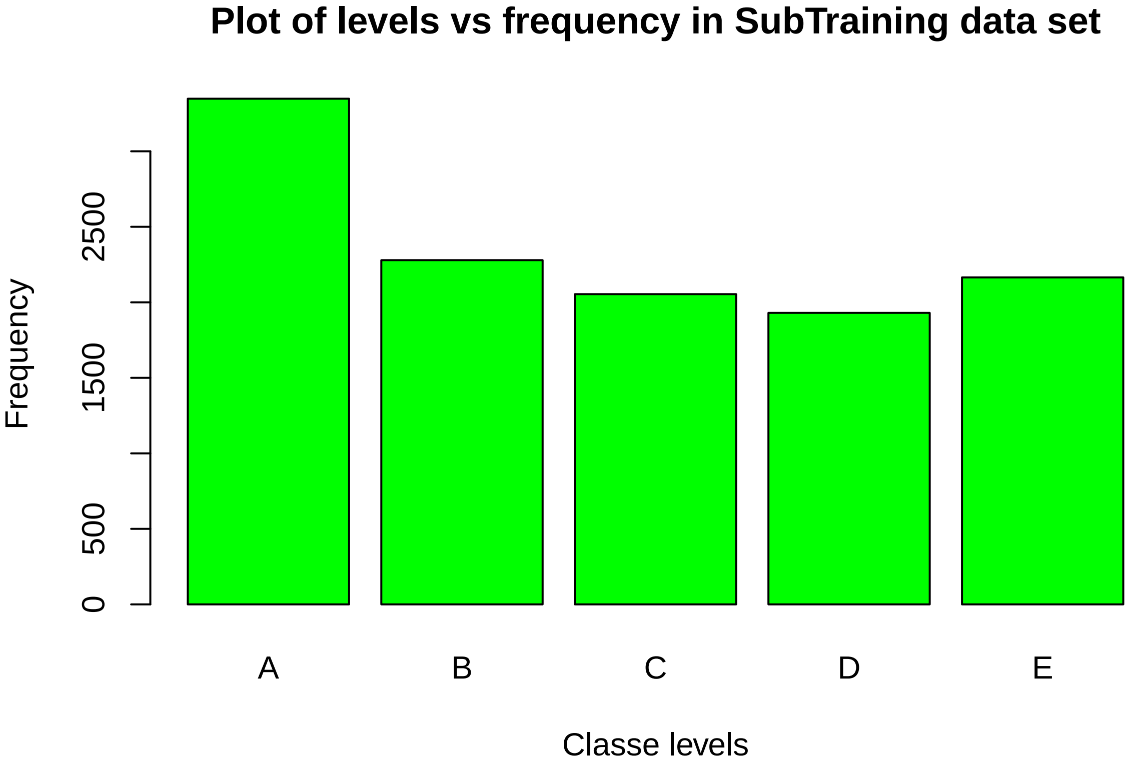
Our new training data set contains 53 variables and 19622 observations where as the testing data set contains 53 variables and 20 observations.

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

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

Since the data we are working is too large to perform an algorithm, the given training data set partitioned into two: Training data set into two data sets, 60% for myTraining, and subTest 40%. This can be performed with random sampling without replacement.

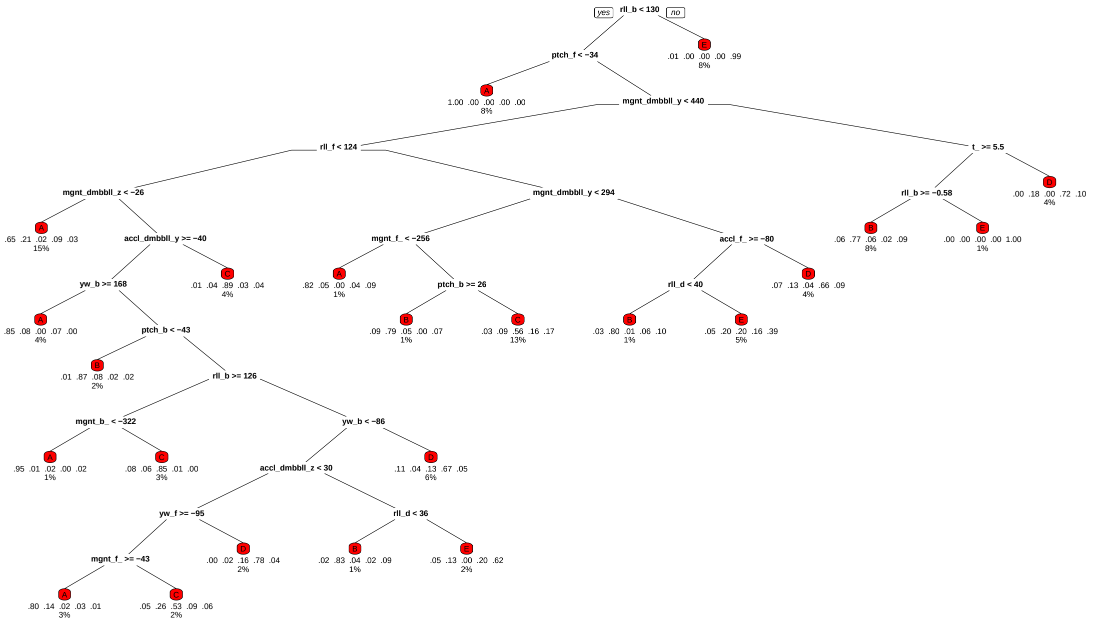
When we look at variable "classe", it contains 5 levels: A, B, C, D and E. A plot of the outcome variable will allow us to see the frequency of each levels in the SubTraining data set. As we can see in the figure below level A has more than 4000 occurrences than other levels.



For prediction we used Decision Tree and Random Forest prediction models.

1. Predicting using Decision Tree and Testing the results on SubTestingset data set

Plot of the Decision Tree



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2015  316   30  119   48
##          B   46  809   63   29   74
##          C   68  165 1120  220  190
##          D   77  121   78  817   81
##          E   26  107   77  101 1049
##
## Overall Statistics
##
##                Accuracy : 0.7405
##                  95% CI : (0.7307, 0.7502)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6705
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9028   0.5329   0.8187   0.6353   0.7275
## Specificity            0.9086   0.9665   0.9007   0.9456   0.9514
## Pos Pred Value         0.7971   0.7924   0.6353   0.6959   0.7713
## Neg Pred Value         0.9592   0.8961   0.9592   0.9297   0.9394
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2568   0.1031   0.1427   0.1041   0.1337
## Detection Prevalence   0.3222   0.1301   0.2247   0.1496   0.1733
## Balanced Accuracy      0.9057   0.7497   0.8597   0.7904   0.8394
```

2. Predicting using Random Forest and Test the results on SubTestingset data set.

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2231    6    0    0    0
##          B    1 1504   14    0    0
##          C    0    8 1354   22    0
##          D    0    0    0 1261    3
##          E    0    0    0    3 1439
## 
## Overall Statistics
## 
##                Accuracy : 0.9927
##                  95% CI : (0.9906, 0.9945)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9908
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9996   0.9908   0.9898   0.9806   0.9979
## Specificity            0.9989   0.9976   0.9954   0.9995   0.9995
## Pos Pred Value         0.9973   0.9901   0.9783   0.9976   0.9979
## Neg Pred Value         0.9998   0.9978   0.9978   0.9962   0.9995
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1917   0.1726   0.1607   0.1834
## Detection Prevalence   0.2851   0.1936   0.1764   0.1611   0.1838
## Balanced Accuracy      0.9992   0.9942   0.9926   0.9901   0.9987
```

As shown above the accuracy for Random Forest model is 0.9927 where as for for Decision Tree model is 0.7405. Therefore Random Forest algorithm is chosen because it performed better than Decision Trees. From our cross-validation data none of the test samples will be miss classified.

```
##  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
```

```r
# Write files 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(predictedresult)
```

References

1. http://www.jstatsoft.org/v28/i05/paper

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.