Practical Machine Learning - Peer Assessments_Prediction Assignment Writeup

Alexander Calzadilla Mendez

Thursday, February 19, 2015

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://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Set Directory and Source of Data for work

The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

```
setwd(dir="d:/Biblioteca/00 - COURSERA Johns Hopkins Specialization in Data Science/Modulo 08 - Practical Machine Learning/")
TrainingData <-read.csv("pml-training.csv")
TestingData  <-read.csv("pml-testing.csv" )
```

Loading all library used

```
library(ggplot2)
library(lattice)
library(caret)
library(rpart)
library(randomForest)
```

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

Exploratory Analysis and Visualisation of

the Data

```
# Row's Counting
nrow(TrainingData)
```

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

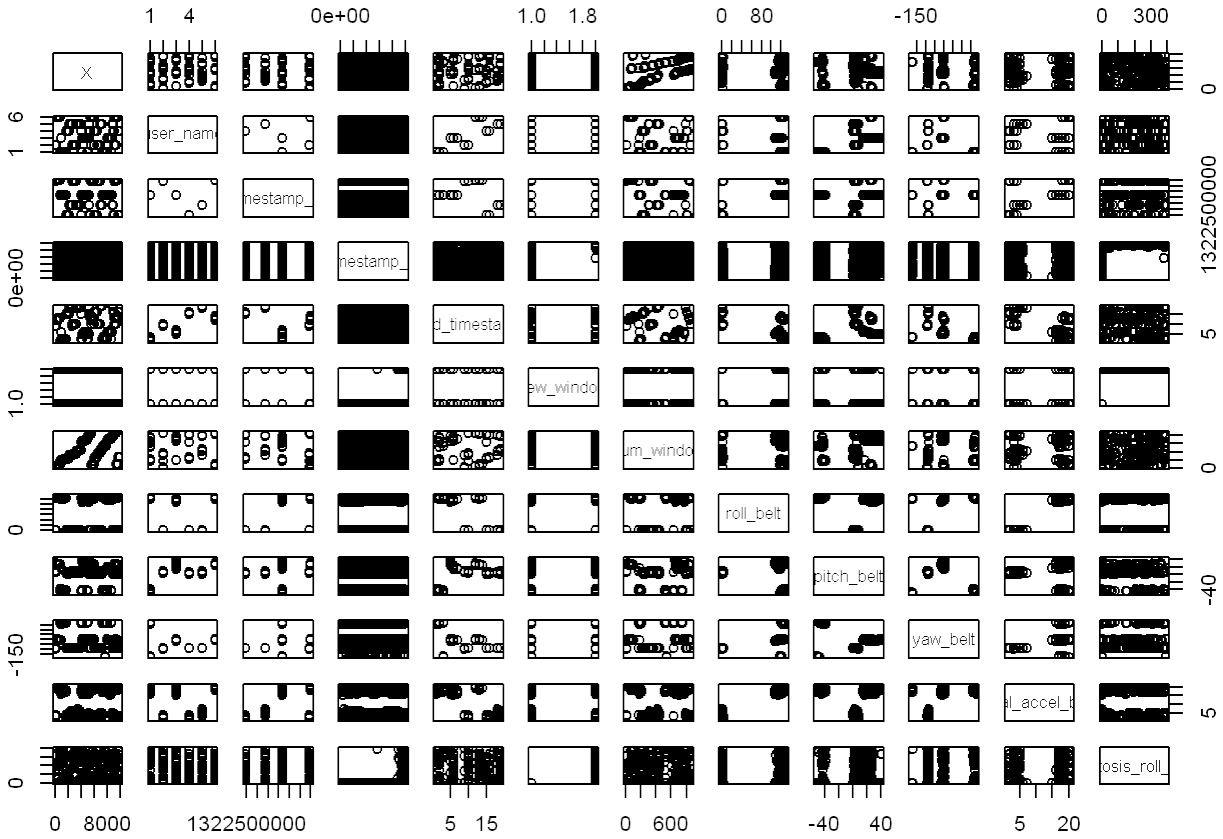
```
# Column's Counting
ncol(TrainingData)
```

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

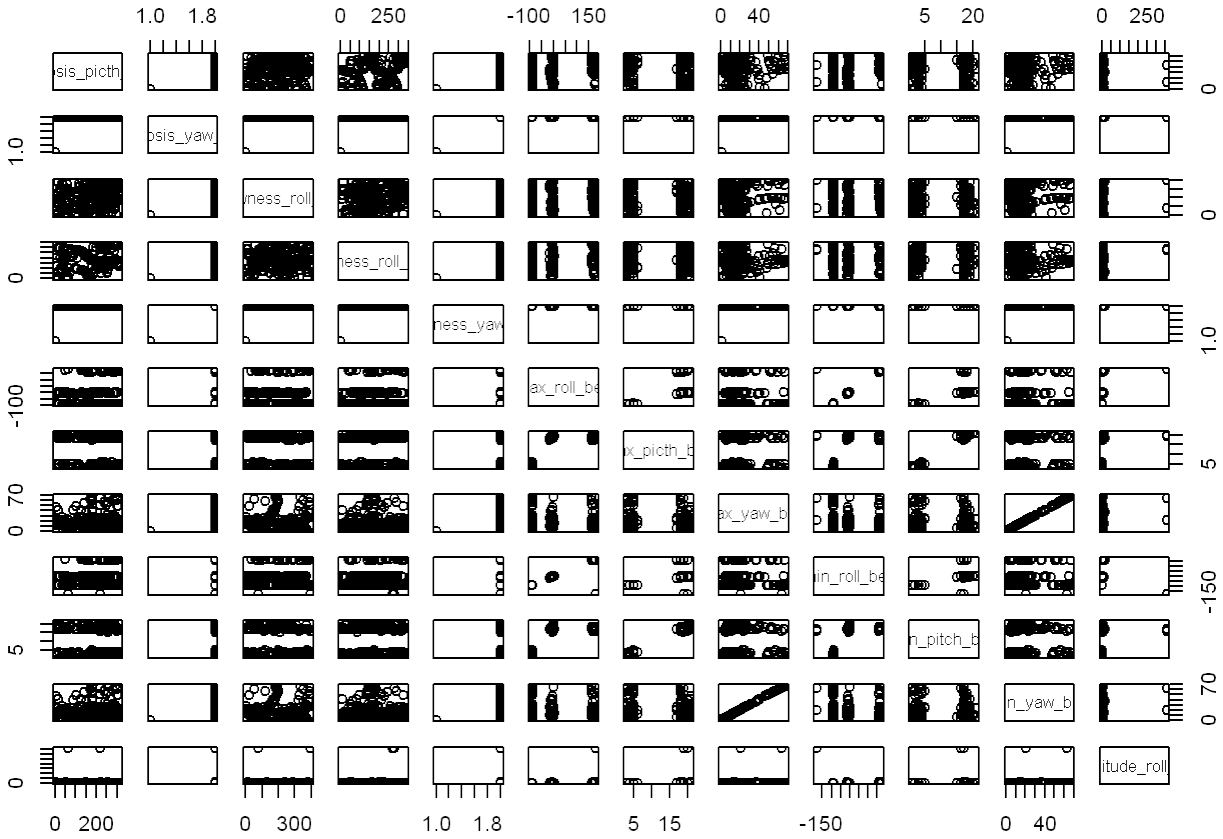
```
#Summary details:
summary(TrainingData[,c(1:2,159:160)])
```

```
##        X            user_name    magnet_forearm_z classe
##  Min.   :    1   adelmo  :3892   Min.   :-973.0   A:5580
##  1st Qu.: 4906   carlitos:3112   1st Qu.: 191.0   B:3797
##  Median : 9812   charles :3536   Median : 511.0   C:3422
##  Mean   : 9812   eurico  :3070   Mean   : 393.6   D:3216
##  3rd Qu.:14717   jeremy  :3402   3rd Qu.: 653.0   E:3607
##  Max.   :19622   pedro   :2610   Max.   :1090.0
```

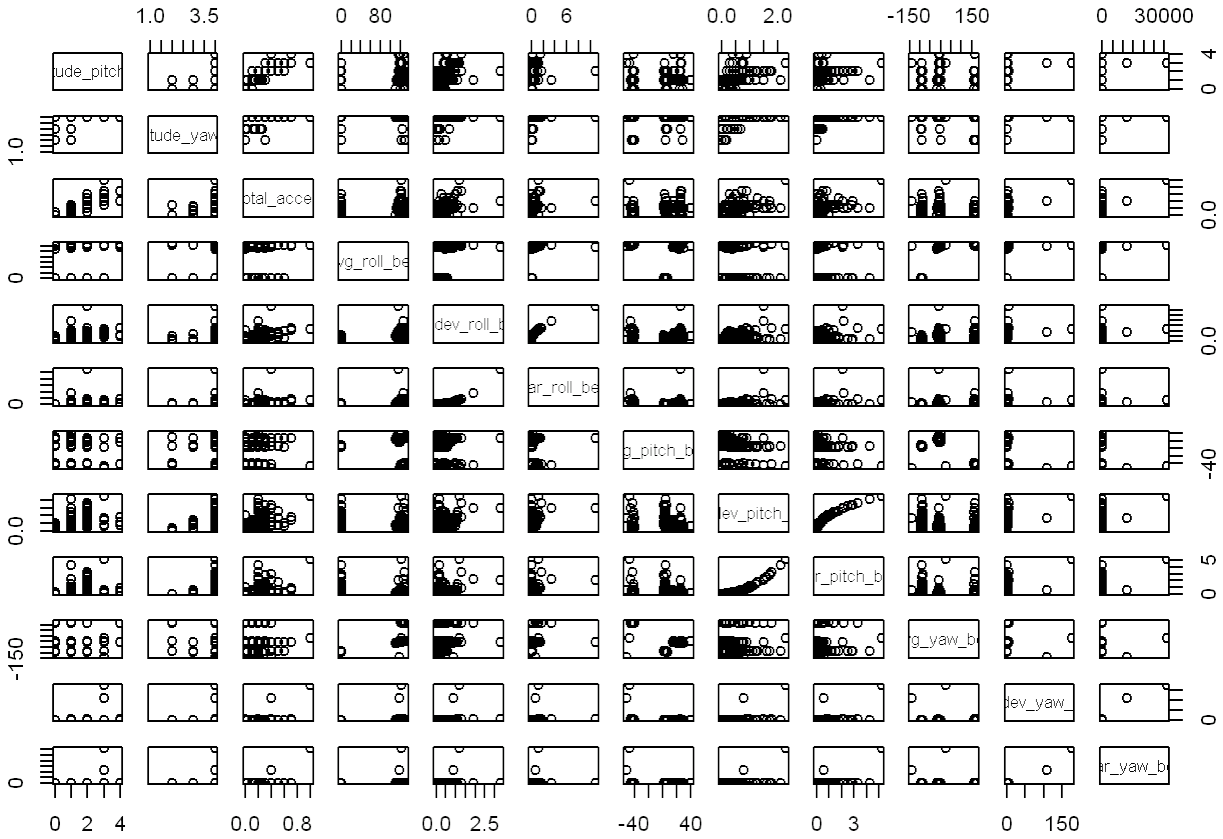
```
# Data Visualization using Scatterplot Matrices for visual reference
pairs(TrainingData[1:10000,1:12])
```




```
pairs(TrainingData[1:10000,13:24])
```




```
pairs(TrainingData[1:10000,25:36])
```



Depuration and Cleaning of the Data set

According with the Exploratory Analysis, there are 160 columns in the Training Set, and must be excluded columns with NA values.

```
TrainingFinalData<-TrainingData
TrainingFinalData[ TrainingFinalData == '' | TrainingFinalData == 'NA'] <- NA
indx <-which(colSums(is.na(TrainingFinalData))!=0)
TrainingFinalData<-TrainingFinalData[, -indx]
TrainingFinalData<-TrainingFinalData[,-(1:7)]
```

Creating a Data set that are Valid

This will be useful for Cross validation with the Training Set.

```
InTraining  <- createDataPartition(y=TrainingFinalData$classe,p=0.70,list=FALSE)
TrainingFinalData <- TrainingFinalData[InTraining,]
ValidateSet <- TrainingFinalData[-InTraining,]
```

Elaboration of Prediction model

Here will be used Random Forest library to train the Prediction Model set to predict the weight lifting quality in the

Training Set.

```
Pmodel <- train(classe~., data=TrainingFinalData, method = "rf", tuneLength = 1, ntree = 25)
print(Pmodel)
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 13737, 13737, 13737, 13737, 13737, 13737, ...
##
## Resampling results
##
##   Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.9884408  0.9853745  0.002321714  0.002937213
##
## Tuning parameter 'mtry' was held constant at a value of 7
##
```

Testing the Prediction Model

For the test we use the Confussion Matrix to evaluate the Prediction Model set versus the Validate Data set.

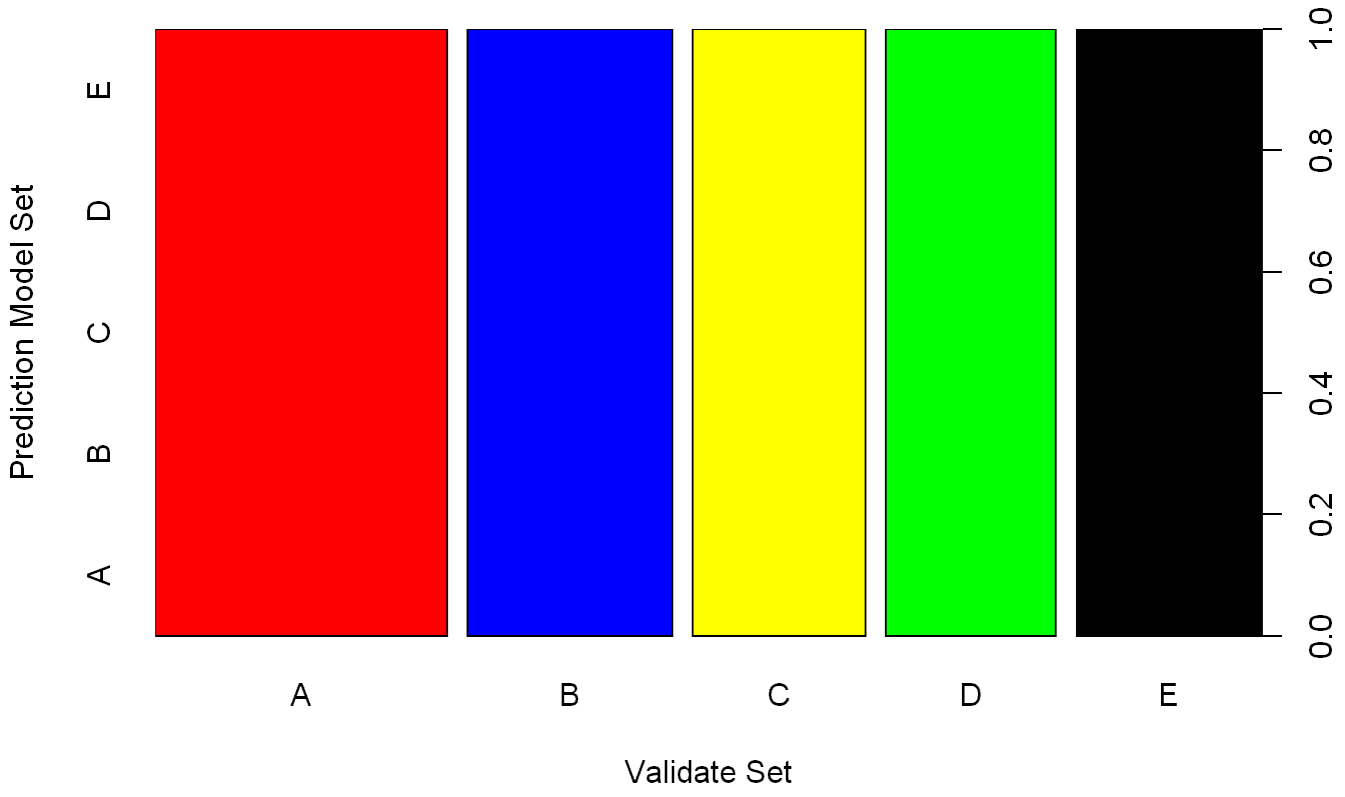
```
confusionMatrix(predict(Pmodel, ValidateSet), ValidateSet$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1169    0    0    0    0
##          B    0  819    0    0    0
##          C    0    0  693    0    0
##          D    0    0    0  680    0
##          E    0    0    0    0  742
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9991, 1)
##     No Information Rate : 0.2849
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2849   0.1996   0.1689   0.1657   0.1808
## Detection Rate         0.2849   0.1996   0.1689   0.1657   0.1808
## Detection Prevalence   0.2849   0.1996   0.1689   0.1657   0.1808
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Graphical diagram of the Prediction Model's accuracy

```
plot(predict(Pmodel,newdata=ValidateSet[,-ncol(ValidateSet)]),ValidateSet$classe, xlab="Validate Set"
, ylab="Prediction Model Set",col = c("red", "blue","yellow","green", "black"))
```




```
#  Brief description about each class:

#  A: Exactly according to the specification
#  B: Throwing the elbows to the front
#  C: Lifting the dumbbell only halfway
#  D: Lowering the dumbbell only halfway
#  E: Throwing the hips to the front
```

Estimation of the Accuracy of the Prediction Model

```
accurate <- c(as.numeric(predict(Pmodel,newdata=ValidateSet[,-ncol(ValidateSet)])==ValidateSet$classe
))
MAccuracy <- sum(accurate)*100/nrow(ValidateSet)
message("Accuracy of Prediction Model set VS Validate Data set = ", format(round(MAccuracy, 2), nsmal
l=2),"%")
```

```
## Accuracy of Prediction Model set VS Validate Data set = 100.00%
```

Forecast on the testing set:

```
# Number of rows:
nrow(TestingData)
```

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

```
# Number of columns:
ncol(TestingData)
```

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

```
# Summary details:
summary(TestingData[,c(1:2,159:160)])
```

```
##        X            user_name magnet_forearm_z   problem_id
##  Min.   : 1.00   adelmo  :1   Min.   :-32.0    Min.   : 1.00
##  1st Qu.: 5.75   carlitos:3   1st Qu.:275.2    1st Qu.: 5.75
##  Median :10.50   charles :1   Median :491.5    Median :10.50
##  Mean   :10.50   eurico  :4   Mean   :460.2    Mean   :10.50
##  3rd Qu.:15.25   jeremy  :8   3rd Qu.:661.5    3rd Qu.:15.25
##  Max.   :20.00   pedro   :3   Max.   :884.0    Max.   :20.00
```

```
Ptest<-predict(Pmodel, TestingData)
print(Ptest)
```

```
##  [1] 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
```

Conclusion

A 100% accuracy was computed here, but must be taken some caution due to the use of Random forest, tends to overfitting the results.