Week 4 Assignment-Practical Machine Learning(Prediction)

Brenda M. Balala

November 1, 2018

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).

Data

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

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Initialization

```
> install.packages("caret")

package ‘caret’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\user\AppData\Local\Temp\RtmpS4sC0c\downloaded_packages

> library(caret)
Loading required package: lattice
Loading required package: ggplot2
```

```
Error: package or namespace load failed for 'ggplot2' in loadNamespace(j <- i
[[1L]], c(lib.loc, .libPaths()), versionCheck = vI[[j]]):
 there is no package called 'rlang'
Error: package 'ggplot2' could not be loaded
In addition: Warning messages:
1: package 'caret' was built under R version 3.5.1
2: package 'ggplot2' was built under R version 3.5.1
```

Loading Dataset

```
#Loading the dataset
training_data <- read.csv('pml_training.csv', na.strings = c("NA", "#DIV/0!",
""))
test_data <- read.csv('pml_testing.csv', na.strings = c("NA", "#DIV/0!", ""))
dim(training_data); dim(test_data)
[1] 19622   160
[1]  20 160
```

Data Preparation

Remove those data contains more than 95% of the observation to be NA, and filter out those records.

```
clnColumnIndex <- colSums(is.na(training_data))/nrow(training_data) < 0.95
clean_training_data <- training_data[,clnColumnIndex]
colSums(is.na(clean_training_data))/nrow(clean_training_data)
                   X            user_name raw_timestamp_part_1 raw_timestamp_part_2       cvtd_timestamp
                   0                    0                    0                    0                    0
          new_window           num_window            roll_belt           pitch_belt             yaw_belt
                   0                    0                    0                    0                    0
    total_accel_belt         gyros_belt_x         gyros_belt_y         gyros_belt_z         accel_belt_x
                   0                    0                    0                    0                    0
        accel_belt_y         accel_belt_z        magnet_belt_x        magnet_belt_y        magnet_belt_z
                   0                    0                    0                    0                    0
            roll_arm            pitch_arm              yaw_arm      total_accel_arm          gyros_arm_x
                   0                    0                    0                    0                    0
         gyros_arm_y          gyros_arm_z          accel_arm_x          accel_arm_y          accel_arm_z
                   0                    0                    0                    0                    0
        magnet_arm_x         magnet_arm_y         magnet_arm_z        roll_dumbbell       pitch_dumbbell
                   0                    0                    0                    0                    0
        yaw_dumbbell total_accel_dumbbell     gyros_dumbbell_x     gyros_dumbbell_y     gyros_dumbbell_z
                   0                    0                    0                    0                    0
    accel_dumbbell_x     accel_dumbbell_y     accel_dumbbell_z    magnet_dumbbell_x    magnet_dumbbell_y
                   0                    0                    0                    0                    0
   magnet_dumbbell_z         roll_forearm        pitch_forearm          yaw_forearm  total_accel_forearm
                   0                    0                    0                    0                    0
     gyros_forearm_x      gyros_forearm_y      gyros_forearm_z      accel_forearm_x      accel_forearm_y
                   0                    0                    0                    0                    0
     accel_forearm_z     magnet_forearm_x     magnet_forearm_y     magnet_forearm_z               classe
                   0                    0                    0                    0                    0
```

Remove unnecessary columns

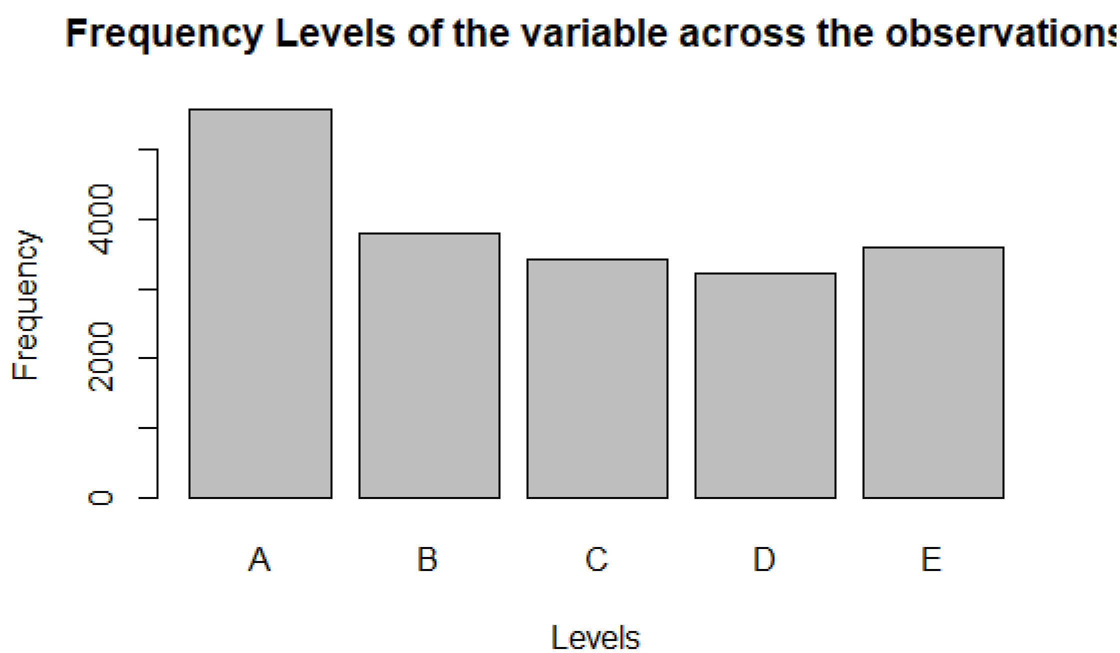
```
clean_training_data <- clean_training_data[,-c(1:7)]
clean_test_data <- test_data[,-c(1:7)]
```

Partition the training data into training set and cross validation set

```
inTrainIndex <- createDataPartition(clean_training_data$classe, p=0.75)[[1]]
training_training_data <- clean_training_data[inTrainIndex,]
training_crossval_data <- clean_training_data[-inTrainIndex,]
```

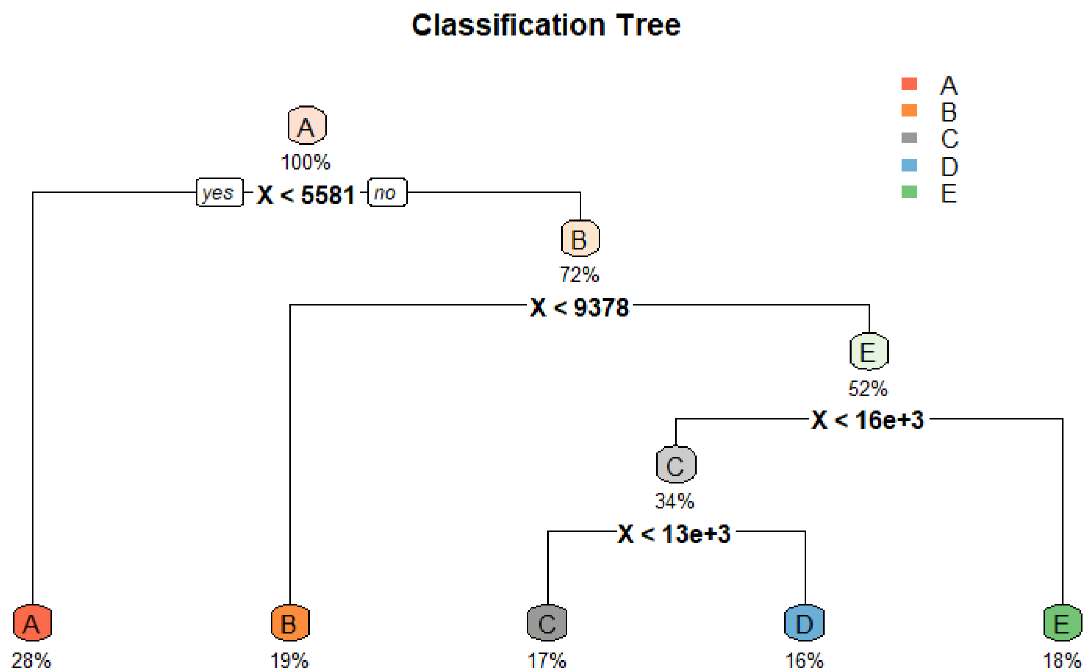
Plotting the frequency

```
plot(training_data$classe, main="Frequency Levels of the variable across the
observations", xlab="Levels", ylab="Frequency")
```



Decision Tree Model

```
# let's set the seed first
> set.seed(2007)
> modFit1<-rpart(classe~., data=training_data, method="class")
> rpart.plot(modFit1, main="Classification Tree", extra=100, under=TRUE, facl
en=0)
```




```
> predict1<-predict(modFit1, subTraining, type="class")
> confusionMatrix(predict1, subTraining$classe)
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 2523  340   20  143   90
         B  155 1287  183  143  121
         C  142  181 1342   89   79
         D   45   53  224 1237  235
         E   65  133   28   77 1370

Overall Statistics

               Accuracy : 0.7529
                 95% CI : (0.7445, 0.7612)
    No Information Rate : 0.2843
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6869
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: A Class: B Class: C Class: D
Sensitivity            0.8611   0.6454   0.7468   0.7324
Specificity            0.9196   0.9276   0.9423   0.9354
Pos Pred Value         0.8097   0.6813   0.7321   0.6895
Neg Pred Value         0.9434   0.9160   0.9463   0.9469
Prevalence             0.2843   0.1935   0.1744   0.1639
Detection Rate         0.2448   0.1249   0.1302   0.1200
```

```
Detection Prevalence   0.3024   0.1833   0.1779   0.1741
Balanced Accuracy      0.8903   0.7865   0.8445   0.8339
                     Class: E
Sensitivity            0.7230
Specificity            0.9640
Pos Pred Value         0.8189
Neg Pred Value         0.9392
Prevalence             0.1839
Detection Rate         0.1329
Detection Prevalence   0.1623
Balanced Accuracy      0.8435
```

Random Forest Model

```
> inTrain<-createDataPartition(y=training_data$classe, p=0.7, list=FALSE)
> subTraining<-training_data[inTrain,]
> subTesting<-training_data[-inTrain,]
> dim(subTraining); dim(subTesting)
[1] 13737   160
[1] 5885  160
> modFit2<-randomForest(classe~.,subTraining,method="class")
> predict2<-predict(modFit2, subTesting, type="class")
> confusionMatrix(predict2, subTesting$classe)
> # let's set the seed first
> set.seed(2007)
> modFit2<-randomForest(classe~.,subTraining,method="class")
> # let's use it for prediction on subTesting
> predict2<-predict(modFit2, subTesting, type="class")
> #show the results
> confusionMatrix(predict2, subTesting$classe)
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1255    9    0    0    0
         B    0  842    6    0    0
         C    0    3  761   13    0
         D    0    0    3  709    3
         E    0    0    0    1  808

Overall Statistics

               Accuracy : 0.9914
                 95% CI : (0.9882, 0.9939)
    No Information Rate : 0.2844
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9891
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            1.0000   0.9859   0.9883   0.9806   0.9963
```

```
Specificity            0.9972   0.9983   0.9956   0.9984   0.9997
Pos Pred Value         0.9929   0.9929   0.9794   0.9916   0.9988
Neg Pred Value         1.0000   0.9966   0.9975   0.9962   0.9992
Prevalence             0.2844   0.1935   0.1745   0.1638   0.1838
Detection Rate         0.2844   0.1908   0.1724   0.1607   0.1831
Detection Prevalence   0.2864   0.1922   0.1761   0.1620   0.1833
Balanced Accuracy      0.9986   0.9921   0.9920   0.9895   0.9980
```

Conclusion

We are going to select Random Forest model due to better accuracy results which is 99% or (0.9939) compared to Decision Tree method (0.7529). The expected out-of-sample error is calculated as 1 - accuracy for predictions made against the cross-validation set, thus our expected out-of-sample error is 0.005 or 0.5%.