PRATICAL MACHINE LEARNING COURSE PROJECT

Gongyao Wang

February 10, 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://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/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 (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/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.

Project goal and submission

The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Peer Review Portion Your submission for the Peer Review portion should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders.

Data processing and cleaning

An overall pseudo-random number generator seed was set at 1000 for all code. In order to reproduce the results below, the same seed should be used. If different packages versions are downloaded and installed, such as caret and randomForest, these may produce different outcome.

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

set.seed(1000)
```

read data from the files that were downloaded from website

```
trainingData <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
testingData <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))
```

Delete columns with all missing values

```
trainingData <-trainingData[,colSums(is.na(trainingData)) == 0]
testingData <-testingData[,colSums(is.na(testingData)) == 0]
```

Delete variables that are not for our current project: user_name, raw_timestamp_part_1, raw_timestamp_part_,2 cvtd_timestamp, new_window, and num_window (columns 1 to 7).

```
trainingData <- trainingData[,-c(1:7)]
testingData <- testingData[,-c(1:7)]
```

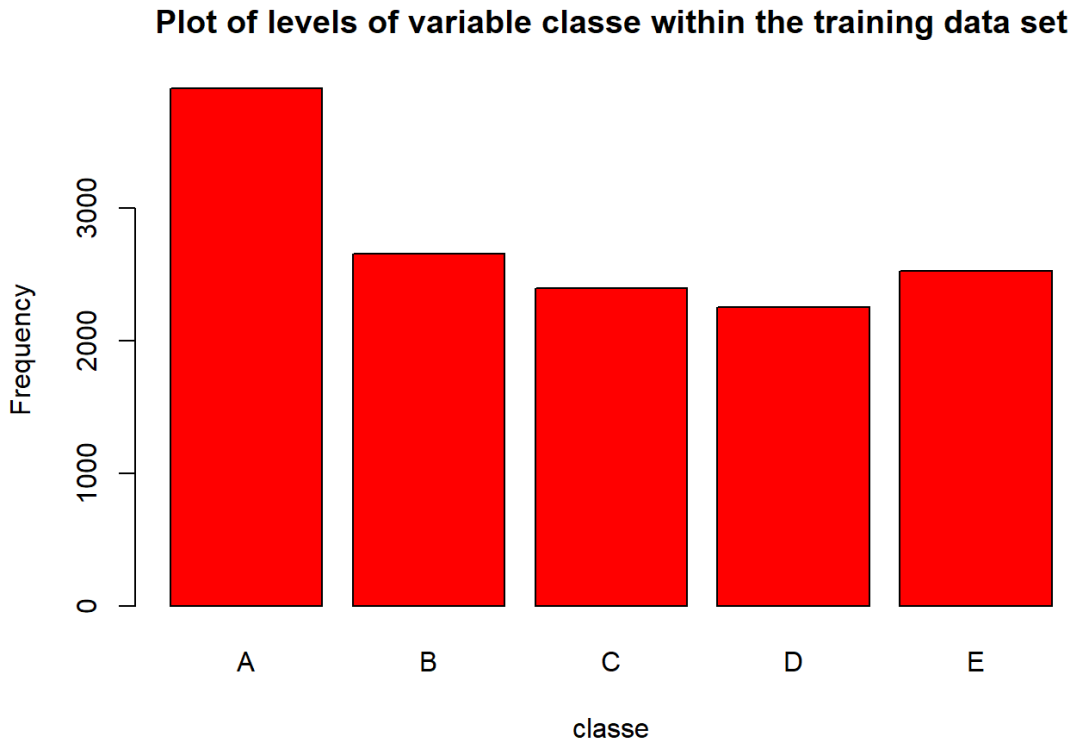
Partition Training data

Here I will split the training data with 70/30 into train and test sets.

```
subdata <- createDataPartition(y = trainingData$classe, p=0.70, list=FALSE)
TraintrainingData <- trainingData[subdata, ]
TesttrainingData <- trainingData[-subdata, ]
```

The variable “classe” 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 training data set.

```
plot(TraintrainingData$classe, col="red", main="Plot of levels of variable classe within the training data set",
xlab="classe", ylab="Frequency")
```



Based on the graph above, we can see that each level frequency is within the same order of magnitude of each other. Level A is the most frequent while level D is the least frequent.

Model 1: Using Decision Tree to predict

```
model_1 <- rpart(classe ~ ., data=TraintrainingData, method="class")

# Prediction:
prediction_1 <- predict(model_1, TesttrainingData, type = "class")
```

Test results on the TesttrainingData data set:

```
confusionMatrix(prediction_1, TesttrainingData$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1461  150   10   57   50
##          B   88  814  138   68   71
##          C   14  111  790  161   83
##          D   44   56   47  643   82
##          E   67    8   41   35  796
##
## Overall Statistics
##
##                Accuracy : 0.7653
##                  95% CI : (0.7543, 0.7761)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7028
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8728   0.7147   0.7700   0.6670   0.7357
## Specificity            0.9366   0.9231   0.9241   0.9535   0.9686
## Pos Pred Value         0.8455   0.6904   0.6816   0.7374   0.8405
## Neg Pred Value         0.9488   0.9309   0.9501   0.9360   0.9421
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2483   0.1383   0.1342   0.1093   0.1353
## Detection Prevalence   0.2936   0.2003   0.1969   0.1482   0.1609
## Balanced Accuracy      0.9047   0.8189   0.8470   0.8102   0.8521
```

Model 2: Using Random Forest to predict

```
model_2 <- randomForest(classe ~. , data=TraintrainingData, method="class")

# Prediction:
prediction_2 <- predict(model_2, TesttrainingData, type = "class")
```

Test results on TesttrainingData data set:

```
confusionMatrix(prediction_2, TesttrainingData$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1671    4    0    0    0
##          B    3 1134    7    0    0
##          C    0    1 1017   10    0
##          D    0    0    2  953    3
##          E    0    0    0    1 1079
##
## Overall Statistics
##
##                Accuracy : 0.9947
##                  95% CI : (0.9925, 0.9964)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9933
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9956   0.9912   0.9886   0.9972
## Specificity            0.9991   0.9979   0.9977   0.9990   0.9998
## Pos Pred Value         0.9976   0.9913   0.9893   0.9948   0.9991
## Neg Pred Value         0.9993   0.9989   0.9981   0.9978   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2839   0.1927   0.1728   0.1619   0.1833
## Detection Prevalence   0.2846   0.1944   0.1747   0.1628   0.1835
## Balanced Accuracy      0.9986   0.9968   0.9945   0.9938   0.9985
```

The choice was made

Based on the above two models, Random Forest algorithm is better than Decision Trees since accuracy for Random Forest model was 0.9947 (95% CI: (0.9925, 0.9964)) compared to 0.7653 (95% CI: (0.7543, 0.7761)) for Decision Tree model. Therefore, the random Forest model is choosen. The expected out-of-sample error is estimated at 0.0053, or 0.53%. Our Test data set comprises 20 cases. With an accuracy above 99% on our cross-validation data, we can expect that very few, or none, of the test samples will be missclassified.

PREDICT THE 20 CASES FOR QUIZ

The final outcome using the Prediction Model 2 (Random Forest) applied against the Testing dataset is as follows:

```
finalPrediction <- predict(model_2, testingData, type="class")
finalPrediction
```

```
##  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
```