Course Project - Prediction Assignment Writeup

Minna Asplund

January 26, 2018

Overview

This document is the final report of the Coursera Practical Machine Learning course project. The document was written with R Studio using R Markdown language. Knitr was used to make the document into a HTML format.

The purpose of the writeup assignment is to predict how well 6 participants performed barbell lifts when asked to do those lifts correctly and incorrectly in 5 different ways.

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 (see the section on the Weight Lifting Exercise Dataset).

About 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 comes from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har. The source is: 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.

In the webpage above, there is a short description of the data:

"*Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).*

Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Participants were supervised by an experienced weight lifter to make sure the execution complied to the manner they were supposed to simulate. The exercises were performed by six male participants aged between 20-28 years, with little weight lifting experience. We made sure that all participants could easily simulate the mistakes in a safe and controlled manner by using a relatively light dumbbell (1.25kg)."

Libraries

```
library(ggplot2)
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)
library(gbm)
```

Data Preparation and Cleaning

Read the .csv files into dataset variables, and replace empty values with NA.

```
training_DS <- read.csv("pml-training.csv", sep=",", header=TRUE, na.strings = c("NA","",'#DIV/0!'))
testing_DS  <- read.csv("pml-testing.csv", sep=",", header=TRUE, na.strings = c("NA","",'#DIV/0!'))
dim(training_DS)
```

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

```
dim(testing_DS)
```

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

Next columns with missing values are removed.

```
training_DS <- training_DS[,(colSums(is.na(training_DS)) == 0)]
testing_DS <- testing_DS[,(colSums(is.na(testing_DS)) == 0)]
```

Additionally the first 7 columns are removed as they are not needed (x, user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp, new_window, num_window).

```
training_DS <- training_DS[, -c(1:7)]
testing_DS <- testing_DS[, -c(1:7)]
dim(training_DS)
```

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

```
dim(testing_DS)
```

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

There are 53 columns remaining in the datasets instead of the original 160 columns.

Dividing training dataset into training set and validation set

In order to ....

```
set.seed(4321)
inTraining <- createDataPartition(training_DS$classe, p = 0.7, list=FALSE)
training <- training_DS[inTraining, ]
validation <- training_DS[-inTraining, ]

dim(training)
```

```
## [1] 13737    53
```

```
dim(validation)
```

```
## [1] 5885   53
```

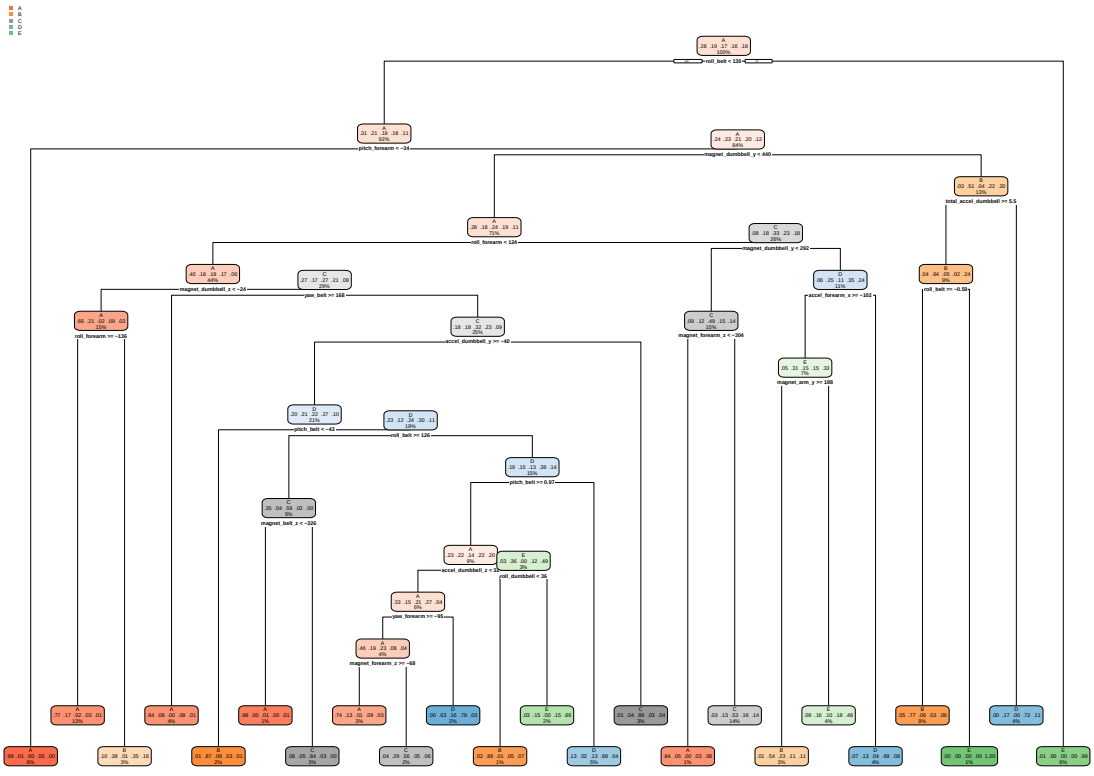
Building models

Three different models are built in order to select the best fitted model.The selection is done by comparing the accuracies of the models.

Decision Tree

At first the model is fitted with training data set.

```
model1_fitted <- rpart(classe ~ ., data = training, method = "class")
rpart.plot(model1_fitted)
```



Then the model is used in prediction with validation data set.

```
prediction_decision_tree <- predict(model1_fitted, validation, type = "class")
result_decision_tree <- confusionMatrix(prediction_decision_tree, validation$classe)
result_decision_tree
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
```

```
##          A 1478  184   24   50   13
##          B   46  653   91   69   93
##          C   69  183  818  144  158
##          D   58   74   60  630   54
##          E   23   45   33   71  764
##
## Overall Statistics
##
##                Accuracy : 0.738
##                  95% CI : (0.7265, 0.7492)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6683
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8829   0.5733   0.7973   0.6535   0.7061
## Specificity            0.9356   0.9370   0.8860   0.9500   0.9642
## Pos Pred Value         0.8451   0.6859   0.5962   0.7192   0.8162
## Neg Pred Value         0.9526   0.9015   0.9539   0.9333   0.9357
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2511   0.1110   0.1390   0.1071   0.1298
## Detection Prevalence   0.2972   0.1618   0.2331   0.1489   0.1590
## Balanced Accuracy      0.9093   0.7552   0.8416   0.8018   0.8351
```

The accuracy of **decision tree** is **0.738**.

Random Forest

At first the model is fitted with training data set.

```
model2_control <- trainControl(method = "cv", number = 3, verboseIter = FALSE)
model2_fitted <- train(classe ~ ., data = training, method = "rf", trControl = model2_control)
```

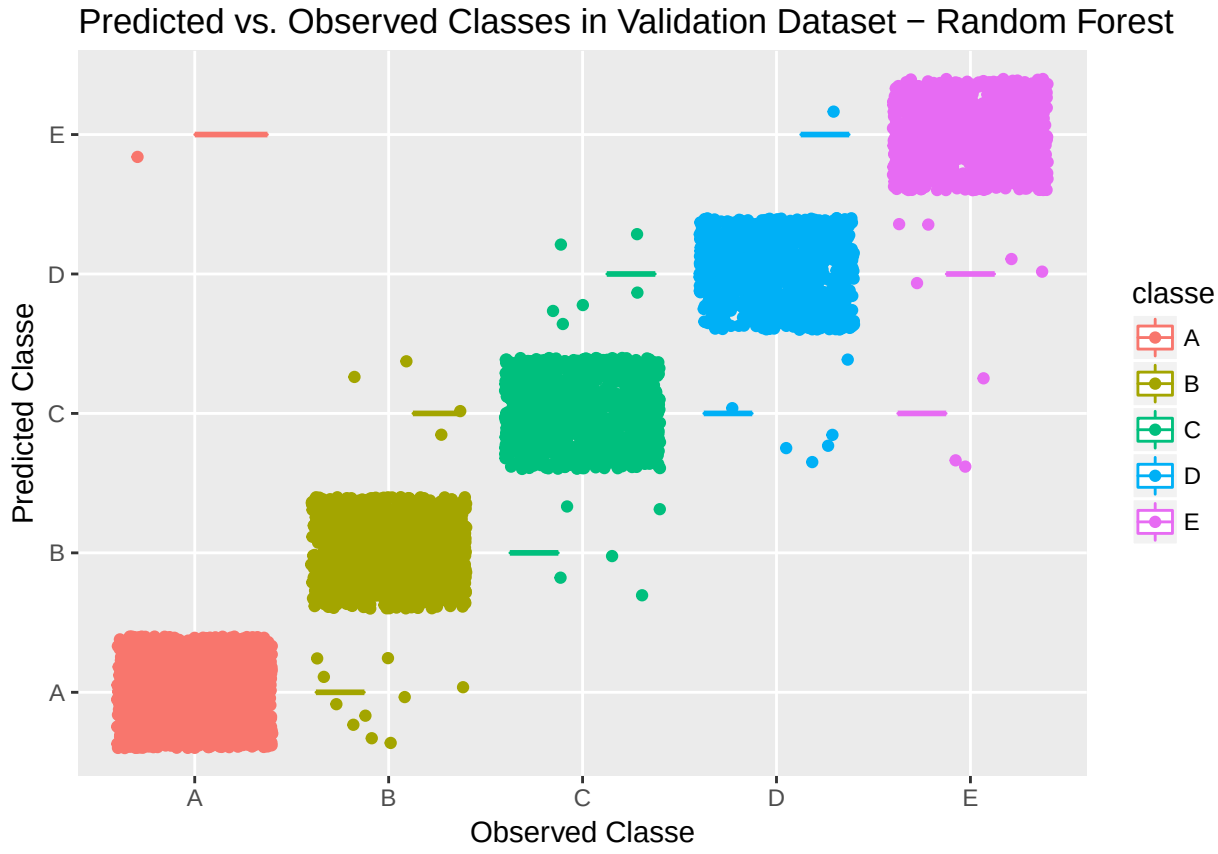
Then the model is used in prediction with validation data set.

```
prediction_random_forest  <- predict(model2_fitted, newdata = validation)
result_random_forest <- confusionMatrix(prediction_random_forest, validation$classe)
result_random_forest
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673   10    0    0    0
##          B    0 1125    5    0    0
##          C    0    4 1015    6    3
##          D    0    0    6  957    5
##          E    1    0    0    1 1074
##
## Overall Statistics
##
```

```
##                Accuracy : 0.993
##                  95% CI : (0.9906, 0.995)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9912
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9877   0.9893   0.9927   0.9926
## Specificity            0.9976   0.9989   0.9973   0.9978   0.9996
## Pos Pred Value         0.9941   0.9956   0.9874   0.9886   0.9981
## Neg Pred Value         0.9998   0.9971   0.9977   0.9986   0.9983
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1912   0.1725   0.1626   0.1825
## Detection Prevalence   0.2860   0.1920   0.1747   0.1645   0.1828
## Balanced Accuracy      0.9985   0.9933   0.9933   0.9953   0.9961
```

```
qplot(classe, prediction_random_forest, data = validation,  colour = classe, geom = c("boxplot", "jitte
```



The accuracy of **random forest** is **0.992**.

General Boosted Model

At first the model is fitted with training data set.

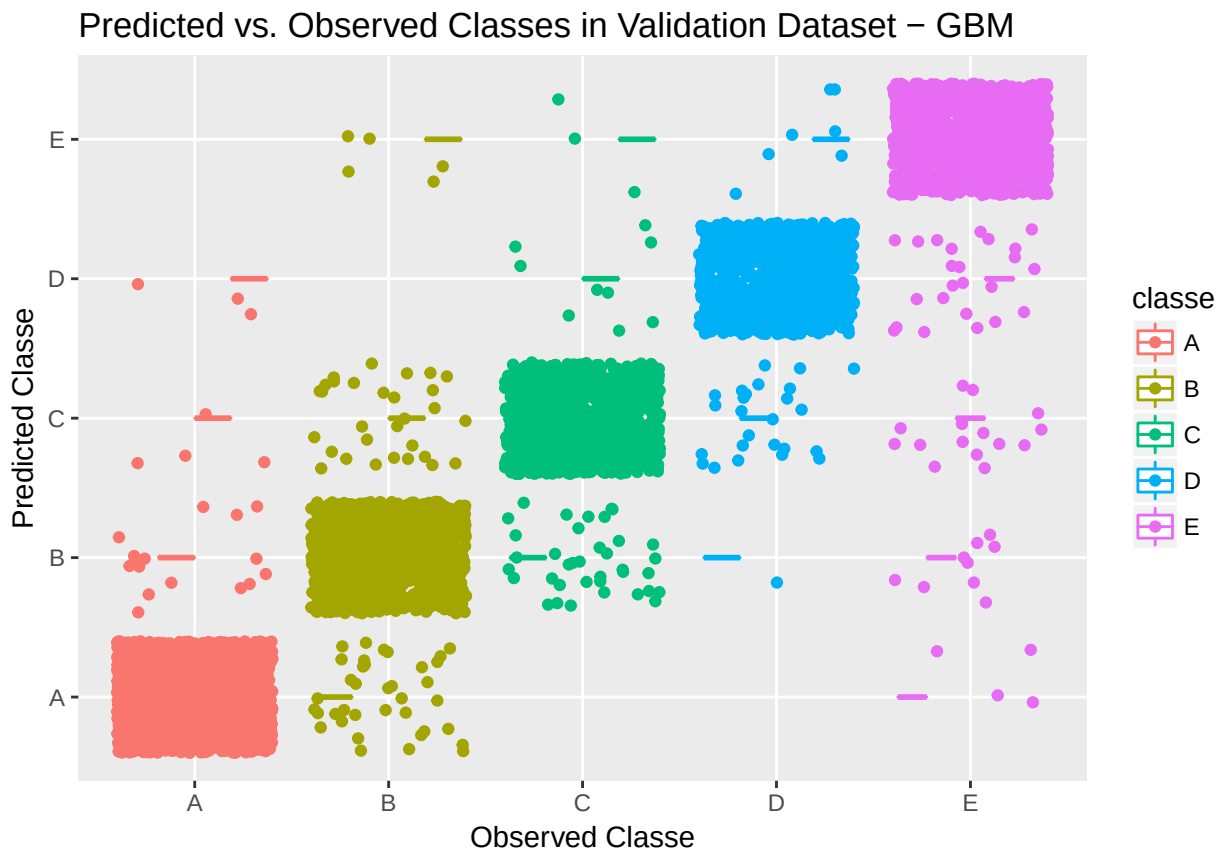
```
model3_control <- trainControl(method = "repeatedcv", number = 3, repeats = 1)
model3_fitted <- train(classe ~ ., data = training, method = "gbm", trControl = model3_control, verbose
```

Then the model is used in prediction with validation data set.

```
prediction_gbm  <- predict(model3_fitted, newdata = validation)
result_gbm <- confusionMatrix(prediction_gbm, validation$classe)
result_gbm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1652   37    0    0    4
##          B   15 1067   37    1    9
##          C    4   30  977   25   15
##          D    3    0    9  931   24
##          E    0    5    3    7 1030
##
## Overall Statistics
##
##                Accuracy : 0.9613
##                  95% CI : (0.956, 0.966)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.951
##  Mcnemar's Test P-Value : 6.119e-07
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9869   0.9368   0.9522   0.9658   0.9519
## Specificity            0.9903   0.9869   0.9848   0.9927   0.9969
## Pos Pred Value         0.9758   0.9451   0.9296   0.9628   0.9856
## Neg Pred Value         0.9948   0.9849   0.9899   0.9933   0.9893
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2807   0.1813   0.1660   0.1582   0.1750
## Detection Prevalence   0.2877   0.1918   0.1786   0.1643   0.1776
## Balanced Accuracy      0.9886   0.9619   0.9685   0.9792   0.9744
```

```
qplot(classe, prediction_gbm, data = validation,  colour = classe, geom = c("boxplot", "jitter"), main =
```



The accuracy of **general boosted model** is **0.960**.

Model Selection

The Random Forest model had the best accuracy rate, so it is selected to be run with actual testing data set (testing_DS), which has not been used so far.

```
final_prediction  <- predict(model2_fitted, newdata = testing_DS)
final_prediction
```

```
##  [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
```