Practical Machine Learning Course Project

Shujuan Huang

Sunday, December 27, 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, our 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.

Reproduceability

An overall pseudo-random number generator seed was set at 50 for all code. In order to reproduce the results below, the same seed should be used.Different packages were downloaded and installed, such as caret and randomForest.

Step 1: Load the data

```
library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-7
## Type rfNews() to see new features/changes/bug fixes.
```

```
setwd("C:/Users/shujuan/Desktop/coursera/practical machine learning/project")
train <- read.csv('pml-training.csv',na.strings=c("NA","","#DIV/0!"))
test <- read.csv('pml-testing.csv',na.strings=c("NA","","#DIV/0!"))
dim(train)
```

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

```r
dim(test)
```

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

```r
#str(train)
```

Step 2: Delete columns with all missing values

```r
train<-train[,colSums(is.na(train)) == 0]
test<-test[,colSums(is.na(test)) == 0]
```

Step 3: Remove variables that are irrelavant to our project, such as user name,raw__timestamp__part__1,raw__t

```r
#names(train2)
train   <-train[,-c(1:7)]
test <-test[,-c(1:7)]
names(train)
```

```
##  [1] "roll_belt"            "pitch_belt"           "yaw_belt"
##  [4] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
##  [7] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [10] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [13] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [16] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [19] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [22] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [25] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [28] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [31] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [34] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [37] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [40] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [43] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [46] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [49] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [52] "magnet_forearm_z"     "classe"
```

Step 4: Spliting the data into training and validation data sets

```r
set.seed(1111)
# Taking 70% for the training data and 30% for the validation data
inTrain <- createDataPartition(y = train$classe, list = FALSE, p=0.7)
train2 <- train[inTrain,]
validation <- train[-inTrain,]
```

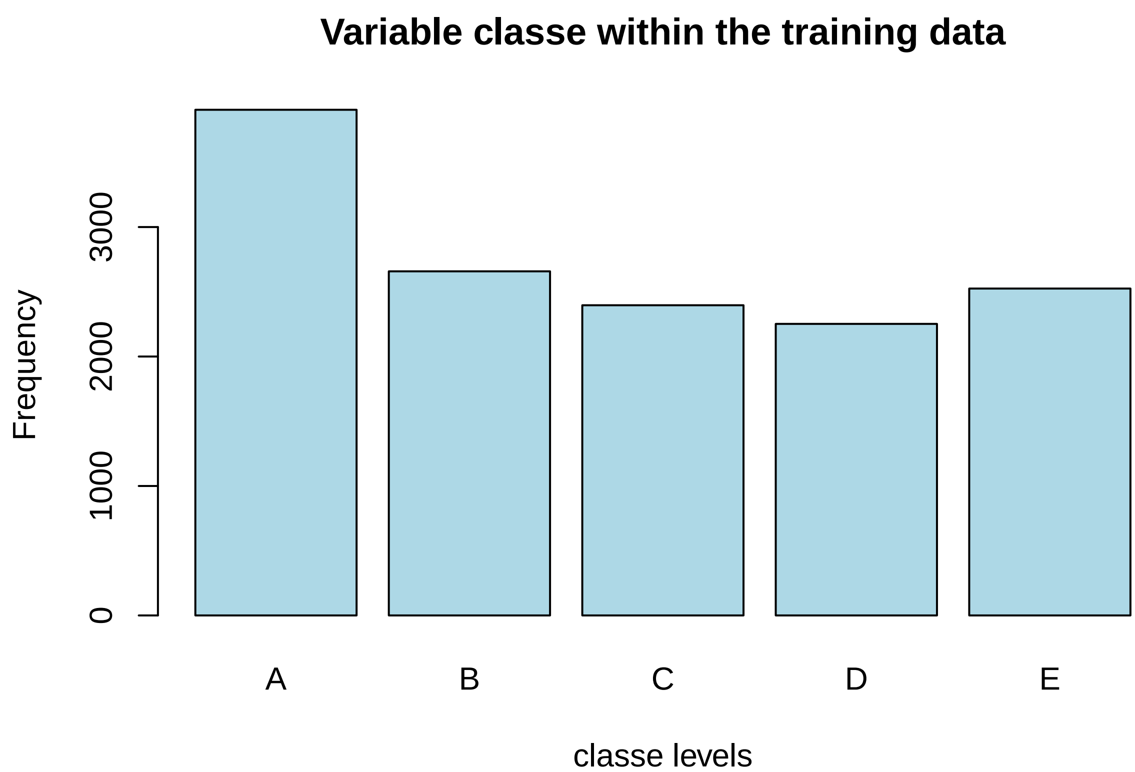
Model

Response Variable: classe

It is a factor variable with 5 levels. For this data set, "participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in 5 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)
- throwing the hips to the front (Class E)

```
par(mfrow=c(1,1))
plot(train2$classe, col="light blue", main="Variable classe within the training data", xlab="classe leve
```



Algorithm: Random Forest

```
#Random Forest Model
model_RF <- randomForest(classe ~. , data=train2, method="class")

# Test results on validation dataset: In sample
prediction2 <- predict(model_RF, train2, type = "class")
confusionMatrix(prediction2, train2$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 3906    0    0    0    0
##          B    0 2658    0    0    0
##          C    0    0 2396    0    0
##          D    0    0    0 2252    0
##          E    0    0    0    0 2525
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9997, 1)
##     No Information Rate : 0.2843
##     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             0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2843   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

```r
# Predicting on validation dataset
prediction1 <- predict(model_RF, validation, type = "class")

# Test results on validation dataset: Out of sample
confusionMatrix(prediction1, validation$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    4    0    0    0
##          B    0 1134    5    0    0
##          C    0    1 1021   11    0
##          D    0    0    0  953    2
##          E    0    0    0    0 1080
##
## Overall Statistics
##
##                Accuracy : 0.9961
##                  95% CI : (0.9941, 0.9975)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.9951
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9956   0.9951   0.9886   0.9982
## Specificity            0.9991   0.9989   0.9975   0.9996   1.0000
## Pos Pred Value         0.9976   0.9956   0.9884   0.9979   1.0000
## Neg Pred Value         1.0000   0.9989   0.9990   0.9978   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1927   0.1735   0.1619   0.1835
## Detection Prevalence   0.2851   0.1935   0.1755   0.1623   0.1835
## Balanced Accuracy      0.9995   0.9973   0.9963   0.9941   0.9991
```

```
# Display the final model: Variable Importance
varImp(model_RF)
```

```
##                        Overall
## roll_belt            870.96813
## pitch_belt           468.85599
## yaw_belt             613.47475
## total_accel_belt     148.50967
## gyros_belt_x          69.79212
## gyros_belt_y          78.08862
## gyros_belt_z         214.82065
## accel_belt_x          85.54908
## accel_belt_y          91.26645
## accel_belt_z         289.93082
## magnet_belt_x        175.37797
## magnet_belt_y        272.30288
## magnet_belt_z        282.14406
## roll_arm             228.71433
## pitch_arm            122.10326
## yaw_arm              167.01253
## total_accel_arm       71.75931
## gyros_arm_x           95.65742
## gyros_arm_y           97.04614
## gyros_arm_z           40.68738
## accel_arm_x          168.78915
## accel_arm_y          107.97035
## accel_arm_z           88.76261
## magnet_arm_x         175.55525
## magnet_arm_y         150.23714
## magnet_arm_z         128.16092
## roll_dumbbell        308.59228
## pitch_dumbbell       125.30317
## yaw_dumbbell         174.90957
## total_accel_dumbbell 173.85508
## gyros_dumbbell_x      91.45836
## gyros_dumbbell_y     160.58089
## gyros_dumbbell_z      60.88134
## accel_dumbbell_x     167.02654
```

```
## accel_dumbbell_y     304.39678
## accel_dumbbell_z     236.76261
## magnet_dumbbell_x    339.96919
## magnet_dumbbell_y    487.50523
## magnet_dumbbell_z    520.96300
## roll_forearm         420.64505
## pitch_forearm        564.68137
## yaw_forearm          122.64917
## total_accel_forearm   79.61545
## gyros_forearm_x       55.31541
## gyros_forearm_y       87.68944
## gyros_forearm_z       58.23671
## accel_forearm_x      230.92168
## accel_forearm_y      100.87137
## accel_forearm_z      169.33553
## magnet_forearm_x     148.02267
## magnet_forearm_y     161.11790
## magnet_forearm_z     204.34862
```

Conclusion

Based on the results, the in sample accuracy of the model is 100%, which is excellent. The out-of-sample accuracy is 99.61%, which is lower than the in sample accuracy as we expected.

Prediction on Testing Dataset

```r
# names(test)
# head(test)
# predict outcome levels on Testing data set using Random Forest algorithm
final <- predict(model_RF, test, type="class")
final
```

```
##  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
```