Prediction Assignment Writeup

Takayuki Sato

Oct.4th.2017

Executive Summary

In this study, we conducted a machine learning(Random Forest) by using 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.

We predict the manner in which they did the exercise. This is the "classe" variable in the training set.

Load Library

```
library(ggplot2)
library(dplyr)
library(caret)
library(randomForest)
```

Load Training / Test data and Data manipulation

```
temp <- tempfile()
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",temp)
train <- read.csv(temp, header=T, stringsAsFactors = FALSE, na.strings=(c("NA", "")))
train <- tbl_df(train)
unlink(temp)

temp <- tempfile()
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",temp)
test <- read.csv(temp, header=T, stringsAsFactors = FALSE, na.strings=(c("NA", "")))
test <- tbl_df(test)
unlink(temp)

## Combine training & test data
full  <- bind_rows(train, test) # bind training & test data
## Checking data size
dim(full)
```

```
## [1] 19642   161
```

```
## Checking NA data
na_count_full <- sapply(full, function(y) sum(is.na(y)))
head(na_count_full[na_count_full>0],15)
```

```
##   kurtosis_roll_belt  kurtosis_picth_belt    kurtosis_yaw_belt
##                19236                19236                19236
##   skewness_roll_belt skewness_roll_belt.1    skewness_yaw_belt
##                19236                19236                19236
##        max_roll_belt       max_picth_belt         max_yaw_belt
```

```
##                19236                19236                19236
##        min_roll_belt       min_pitch_belt         min_yaw_belt
##                19236                19236                19236
##  amplitude_roll_belt amplitude_pitch_belt   amplitude_yaw_belt
##                19236                19236                19236
```

There are many variables containing too many NA data.

Since it is hard to impute those NA data, We decide to eliminate the variables.

(Also, eliminate "X"(it's just a serial number) and "cvtd_timestamp"(dupulicates with "raw_stamp"") for machine learning.)

```
full <- full %>% select(-contains("_pitch_"),-contains("_roll_"),-contains("_yaw_"),-contains("_picth_"
                        -var_total_accel_belt,- var_accel_arm,-var_accel_dumbbell ,-var_accel_forearm
                        -X, -cvtd_timestamp)
## Checking NA data
na_count_full <- sapply(full, function(y) sum(is.na(y)))
na_count_full
```

```
##            user_name raw_timestamp_part_1 raw_timestamp_part_2
##                    0                    0                    0
##           new_window           num_window            roll_belt
##                    0                    0                    0
##           pitch_belt             yaw_belt     total_accel_belt
##                    0                    0                    0
##         gyros_belt_x         gyros_belt_y         gyros_belt_z
##                    0                    0                    0
##         accel_belt_x         accel_belt_y         accel_belt_z
##                    0                    0                    0
##        magnet_belt_x        magnet_belt_y        magnet_belt_z
##                    0                    0                    0
##             roll_arm            pitch_arm              yaw_arm
##                    0                    0                    0
##      total_accel_arm          gyros_arm_x          gyros_arm_y
##                    0                    0                    0
##          gyros_arm_z          accel_arm_x          accel_arm_y
##                    0                    0                    0
##          accel_arm_z         magnet_arm_x         magnet_arm_y
##                    0                    0                    0
##         magnet_arm_z        roll_dumbbell       pitch_dumbbell
##                    0                    0                    0
##         yaw_dumbbell total_accel_dumbbell     gyros_dumbbell_x
##                    0                    0                    0
##     gyros_dumbbell_y     gyros_dumbbell_z     accel_dumbbell_x
##                    0                    0                    0
##     accel_dumbbell_y     accel_dumbbell_z    magnet_dumbbell_x
##                    0                    0                    0
##    magnet_dumbbell_y    magnet_dumbbell_z         roll_forearm
##                    0                    0                    0
##        pitch_forearm          yaw_forearm  total_accel_forearm
##                    0                    0                    0
##      gyros_forearm_x      gyros_forearm_y      gyros_forearm_z
##                    0                    0                    0
##      accel_forearm_x      accel_forearm_y      accel_forearm_z
##                    0                    0                    0
```

```
##     magnet_forearm_x     magnet_forearm_y     magnet_forearm_z
##                    0                    0                    0
##               classe           problem_id
##                   20                19622
```

It seems much better.

We conducted Some data manipulations for Machine Learning.

```
## Factorize some variables
full$classe  <- factor(full$classe)
full$user_name <- factor(full$user_name)
full$new_window <- factor(full$new_window)
full$user_name <- factor(full$user_name)

## Split the data back into a train set and a test set
train3 <- full[1:19622,]
test3 <- full[19623:19642,]
train3 <- train3 %>% select(-problem_id)
test3 <- test3 %>% select(-classe, -problem_id)
```

Machine Learning (Random Forest)

At first, we split training data into training_train/training_test for CV.

```
## Spllit training data
set.seed(62433)
inTrain <- createDataPartition(train3$classe, p=0.7, list=FALSE)
training_train <- train3[inTrain,]
training_test <- train3[-inTrain,]
```

We built Rondom Forest model by using training data set. Also, applied the model to training_test data set.

```
set.seed(62433)
mod_rf <- randomForest(classe ~., data=training_train, n.tree = 1000)
prediction_rf <- predict(mod_rf, training_test)
table(training_test$classe, prediction_rf)
```

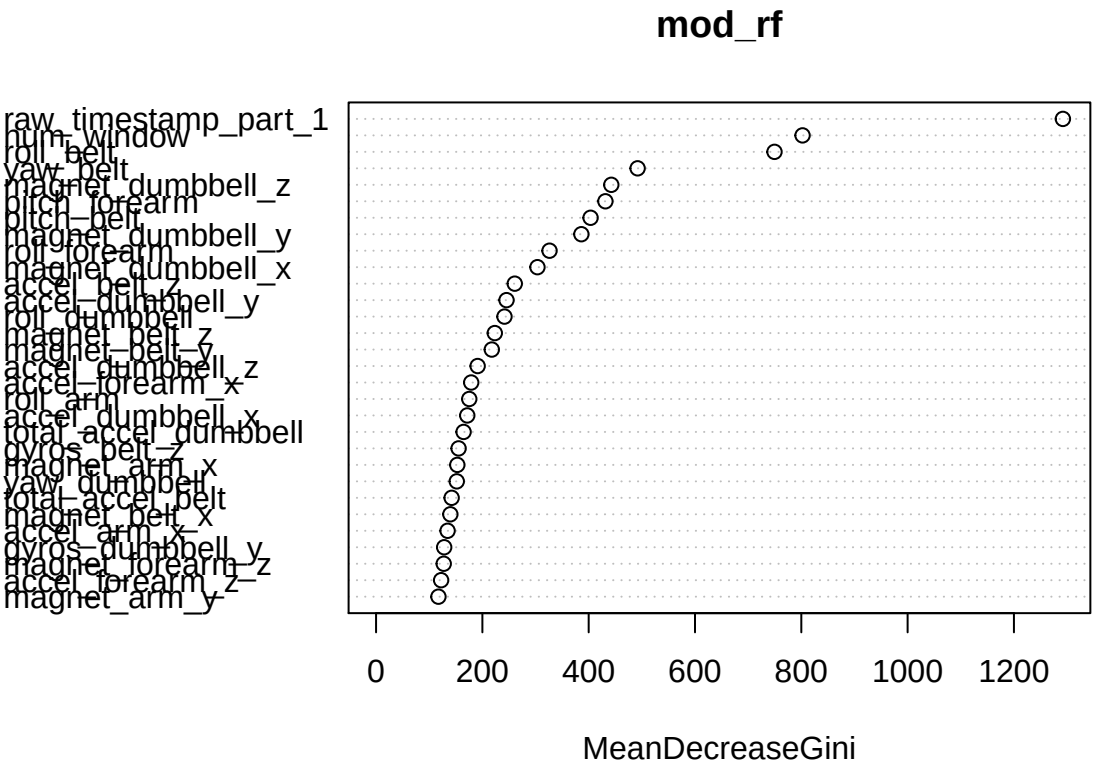
```
##    prediction_rf
##        A    B    C    D    E
##   A 1674    0    0    0    0
##   B    1 1138    0    0    0
##   C    0    2 1024    0    0
##   D    0    0    1  963    0
##   E    0    0    0    0 1082
```

```
sum(diag(table(training_test$classe, prediction_rf)))/nrow(training_test)
```

```
## [1] 0.9993203
```

The accuracy is over 99%.

```
varImpPlot(mod_rf)
```



Looking the relative importance of variables, "raw_timestamp_part1", "num_window" and "roll_belt" are so high scores.

Predict by using test data.

We finally predict 20 different test cases by using the Random Forest model.

```
prediction_rft <- predict(mod_rf, test3)
solution <- data.frame(Problem_ID = 1:20, classe = prediction_rft)
solution
```

```
##    Problem_ID classe
## 1           1      B
## 2           2      A
## 3           3      B
## 4           4      A
## 5           5      A
## 6           6      E
## 7           7      D
## 8           8      B
## 9           9      A
## 10         10      A
## 11         11      B
## 12         12      C
## 13         13      B
## 14         14      A
```

```
## 15          15      E
## 16          16      E
## 17          17      A
## 18          18      B
## 19          19      B
## 20          20      B
```