Practical Machine Learning Project - Writeup

Synopsis

In this report we aim to build a machine learning algorithm to predict activity quality from activity monitors. The data was pre divided into training and testing dataset. The traing dataset was first cleaned up in the following procedure: (1) get the response variable classe into a vector, named classe; (2) remove the non numeric variables; (3) remove the variables which are NA across the whole dataset; (4) remove the X variable; (5) combine the classe with the remaining dataset. After the cleaning, there are 56 variables left to use as predictors to predict the classe variable. Random forest algorith was used to build the classification trees, 10 cross validation was used for resampling and building the trees, 200 trees were evaluated. The final model was evaluated and it has a 0.04% out of sample errors. The model was used to predict the 20 samples in the testing dataset, and 100% accuracy was obtained.

Load the package needed for the tree classification

```
library(caret)
```

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

Load the data

```
training <- read.csv("pml-training.csv")
testing <- read.csv("pml-testing.csv")
```

Cleaning the data

The training dataset was first cleaned up using a series round of criteria. First, the classe was extracted and stored in a new variable; second, the non numeric variables were removed; third, the variables with all NA across the whole dataset were removed, these variables' names have some common characteristics, they are stared with max, min, avg, var, stddev, amp; forth, the X variable was removed; fifth, the classe variavle was combined with the remaining dataset.

```
classe <- training$classe
dim(training);dim(testing)
```

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

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

```
x <- sapply(training, is.numeric)
training <- training[, x]
dim(training)
```

```
## [1] 19622   123
```

```
variables <- names(training)
max <- grep("^max", variables, value = T)
min <- grep("^min", variables, value = T)
avg <- grep("^avg", variables, value = T)
var <- grep("^var", variables, value = T)
std <- grep("^stddev", variables, value = T)
amp <- grep("^amplitude", variables, value = T)
NAs <- c(max, min, avg, var, std, amp, amp)
val <- variables[! variables %in% NAs]
training <- training[, val]
training <- training[, -1]
training <- cbind(training, classe)
head(training)
```

```
##   raw_timestamp_part_1 raw_timestamp_part_2 num_window roll_belt
## 1           1323084231               788290         11      1.41
## 2           1323084231               808298         11      1.41
## 3           1323084231               820366         11      1.42
## 4           1323084232               120339         12      1.48
## 5           1323084232               196328         12      1.48
## 6           1323084232               304277         12      1.45
##   pitch_belt yaw_belt total_accel_belt gyros_belt_x gyros_belt_y
## 1       8.07    -94.4                3         0.00         0.00
## 2       8.07    -94.4                3         0.02         0.00
## 3       8.07    -94.4                3         0.00         0.00
## 4       8.05    -94.4                3         0.02         0.00
## 5       8.07    -94.4                3         0.02         0.02
## 6       8.06    -94.4                3         0.02         0.00
##   gyros_belt_z accel_belt_x accel_belt_y accel_belt_z magnet_belt_x
## 1        -0.02          -21            4           22            -3
## 2        -0.02          -22            4           22            -7
## 3        -0.02          -20            5           23            -2
## 4        -0.03          -22            3           21            -6
## 5        -0.02          -21            2           24            -6
## 6        -0.02          -21            4           21             0
##   magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm
## 1           599          -313     -128      22.5    -161              34
## 2           608          -311     -128      22.5    -161              34
## 3           600          -305     -128      22.5    -161              34
## 4           604          -310     -128      22.1    -161              34
## 5           600          -302     -128      22.1    -161              34
## 6           603          -312     -128      22.0    -161              34
##   gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z
## 1        0.00        0.00       -0.02        -288         109        -123
## 2        0.02       -0.02       -0.02        -290         110        -125
## 3        0.02       -0.02       -0.02        -289         110        -126
## 4        0.02       -0.03        0.02        -289         111        -123
## 5        0.00       -0.03        0.00        -289         111        -123
## 6        0.02       -0.03        0.00        -289         111        -122
##   magnet_arm_x magnet_arm_y magnet_arm_z roll_dumbbell pitch_dumbbell
## 1         -368          337          516      13.05217      -70.49400
## 2         -369          337          513      13.13074      -70.63751
## 3         -368          344          513      12.85075      -70.27812
## 4         -372          344          512      13.43120      -70.39379
## 5         -374          337          506      13.37872      -70.42856
## 6         -369          342          513      13.38246      -70.81759
##   yaw_dumbbell total_accel_dumbbell gyros_dumbbell_x gyros_dumbbell_y
## 1    -84.87394                   37                0            -0.02
## 2    -84.71065                   37                0            -0.02
## 3    -85.14078                   37                0            -0.02
## 4    -84.87363                   37                0            -0.02
## 5    -84.85306                   37                0            -0.02
## 6    -84.46500                   37                0            -0.02
##   gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z
## 1             0.00             -234               47             -271
## 2             0.00             -233               47             -269
## 3             0.00             -232               46             -270
## 4            -0.02             -232               48             -269
## 5             0.00             -233               48             -270
## 6             0.00             -234               48             -269
##   magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z roll_forearm
## 1              -559               293               -65         28.4
## 2              -555               296               -64         28.3
## 3              -561               298               -63         28.3
## 4              -552               303               -60         28.1
## 5              -554               292               -68         28.0
## 6              -558               294               -66         27.9
##   pitch_forearm yaw_forearm total_accel_forearm gyros_forearm_x
## 1         -63.9        -153                  36            0.03
## 2         -63.9        -153                  36            0.02
## 3         -63.9        -152                  36            0.03
## 4         -63.9        -152                  36            0.02
## 5         -63.9        -152                  36            0.02
## 6         -63.9        -152                  36            0.02
##   gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y
## 1            0.00           -0.02             192             203
## 2            0.00           -0.02             192             203
## 3           -0.02            0.00             196             204
## 4           -0.02            0.00             189             206
## 5            0.00           -0.02             189             206
## 6           -0.02           -0.03             193             203
##   accel_forearm_z magnet_forearm_x magnet_forearm_y magnet_forearm_z
## 1            -215              -17              654              476
## 2            -216              -18              661              473
## 3            -213              -18              658              469
## 4            -214              -16              658              469
## 5            -214              -17              655              473
## 6            -215               -9              660              478
##   classe
## 1      A
## 2      A
## 3      A
## 4      A
## 5      A
## 6      A
```

Building the prediction model using random forest algorithm. 10 fold cross validation was used to data split and 200 trees were evaluated.

```
Mod1 <-train(classe ~., method = "rf", trControl = trainControl(method = "cv", number = 10), ntree = 200, data = training)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

predict the testing dataset

```
testing <- testing[, val]
testing <- testing[, -1]
prediction <- predict(Mod1, testing)
```

Evaluate the out of sample error for the final model and print the predicted classe for the testing dataset

```
print(Mod1$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, ntree = 200, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 28
##
##         OOB estimate of  error rate: 0.06%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5580    0    0    0    0 0.0000000000
## B    2 3794    1    0    0 0.0007900974
## C    0    4 3418    0    0 0.0011689071
## D    0    0    3 3212    1 0.0012437811
## E    0    0    0    1 3606 0.0002772387
```

```
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
```