Course Project: Practical Machine Learning

Bryan Kong Chak Bun

September 2, 2019

Introduction

This project made use of the data of body movement for trying to have the Human Activity Recognition. Some basic predictive models, such as random forest, generalized boosted regression and trees, had been applied for prediction.

Getting and cleaning Data

Data was downloaded from the links of the course materials.

```
fileUrl1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv"
fileUrl2 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv"
download.file(fileUrl1, destfile = "./pmlTraining.csv")
download.file(fileUrl2, destfile = "./pmlTesting.csv")
pmlTraining <- read.csv("./pmlTraining.csv")
pmlTesting <- read.csv("./pmlTesting.csv")
summary(pmlTraining$"classe")
```

```
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

R packages were loaded for prediction.

```
library(caret)
```

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

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

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

##
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```
library(rpart)
```

We needed to check the number of varialbes in the training and testing files.

```
cbind(names(pmlTraining), names(pmlTesting))
```

```
##        [,1]                     [,2]
##   [1,] "X"                      "X"
##   [2,] "user_name"              "user_name"
##   [3,] "raw_timestamp_part_1"   "raw_timestamp_part_1"
##   [4,] "raw_timestamp_part_2"   "raw_timestamp_part_2"
##   [5,] "cvtd_timestamp"         "cvtd_timestamp"
##   [6,] "new_window"             "new_window"
##   [7,] "num_window"             "num_window"
##   [8,] "roll_belt"              "roll_belt"
##   [9,] "pitch_belt"             "pitch_belt"
##  [10,] "yaw_belt"               "yaw_belt"
##  [11,] "total_accel_belt"       "total_accel_belt"
##  [12,] "kurtosis_roll_belt"     "kurtosis_roll_belt"
##  [13,] "kurtosis_picth_belt"    "kurtosis_picth_belt"
##  [14,] "kurtosis_yaw_belt"      "kurtosis_yaw_belt"
##  [15,] "skewness_roll_belt"     "skewness_roll_belt"
##  [16,] "skewness_roll_belt.1"   "skewness_roll_belt.1"
##  [17,] "skewness_yaw_belt"      "skewness_yaw_belt"
##  [18,] "max_roll_belt"          "max_roll_belt"
##  [19,] "max_picth_belt"         "max_picth_belt"
##  [20,] "max_yaw_belt"           "max_yaw_belt"
##  [21,] "min_roll_belt"          "min_roll_belt"
##  [22,] "min_pitch_belt"         "min_pitch_belt"
##  [23,] "min_yaw_belt"           "min_yaw_belt"
##  [24,] "amplitude_roll_belt"    "amplitude_roll_belt"
##  [25,] "amplitude_pitch_belt"   "amplitude_pitch_belt"
##  [26,] "amplitude_yaw_belt"     "amplitude_yaw_belt"
##  [27,] "var_total_accel_belt"   "var_total_accel_belt"
##  [28,] "avg_roll_belt"          "avg_roll_belt"
```

```
##  [29,] "stddev_roll_belt"         "stddev_roll_belt"
##  [30,] "var_roll_belt"            "var_roll_belt"
##  [31,] "avg_pitch_belt"           "avg_pitch_belt"
##  [32,] "stddev_pitch_belt"        "stddev_pitch_belt"
##  [33,] "var_pitch_belt"           "var_pitch_belt"
##  [34,] "avg_yaw_belt"             "avg_yaw_belt"
##  [35,] "stddev_yaw_belt"          "stddev_yaw_belt"
##  [36,] "var_yaw_belt"             "var_yaw_belt"
##  [37,] "gyros_belt_x"             "gyros_belt_x"
##  [38,] "gyros_belt_y"             "gyros_belt_y"
##  [39,] "gyros_belt_z"             "gyros_belt_z"
##  [40,] "accel_belt_x"             "accel_belt_x"
##  [41,] "accel_belt_y"             "accel_belt_y"
##  [42,] "accel_belt_z"             "accel_belt_z"
##  [43,] "magnet_belt_x"            "magnet_belt_x"
##  [44,] "magnet_belt_y"            "magnet_belt_y"
##  [45,] "magnet_belt_z"            "magnet_belt_z"
##  [46,] "roll_arm"                 "roll_arm"
##  [47,] "pitch_arm"                "pitch_arm"
##  [48,] "yaw_arm"                  "yaw_arm"
##  [49,] "total_accel_arm"          "total_accel_arm"
##  [50,] "var_accel_arm"            "var_accel_arm"
##  [51,] "avg_roll_arm"             "avg_roll_arm"
##  [52,] "stddev_roll_arm"          "stddev_roll_arm"
##  [53,] "var_roll_arm"             "var_roll_arm"
##  [54,] "avg_pitch_arm"            "avg_pitch_arm"
##  [55,] "stddev_pitch_arm"         "stddev_pitch_arm"
##  [56,] "var_pitch_arm"            "var_pitch_arm"
##  [57,] "avg_yaw_arm"              "avg_yaw_arm"
##  [58,] "stddev_yaw_arm"           "stddev_yaw_arm"
##  [59,] "var_yaw_arm"              "var_yaw_arm"
##  [60,] "gyros_arm_x"              "gyros_arm_x"
##  [61,] "gyros_arm_y"              "gyros_arm_y"
##  [62,] "gyros_arm_z"              "gyros_arm_z"
##  [63,] "accel_arm_x"              "accel_arm_x"
##  [64,] "accel_arm_y"              "accel_arm_y"
##  [65,] "accel_arm_z"              "accel_arm_z"
##  [66,] "magnet_arm_x"             "magnet_arm_x"
##  [67,] "magnet_arm_y"             "magnet_arm_y"
##  [68,] "magnet_arm_z"             "magnet_arm_z"
##  [69,] "kurtosis_roll_arm"        "kurtosis_roll_arm"
##  [70,] "kurtosis_picth_arm"       "kurtosis_picth_arm"
##  [71,] "kurtosis_yaw_arm"         "kurtosis_yaw_arm"
##  [72,] "skewness_roll_arm"        "skewness_roll_arm"
##  [73,] "skewness_pitch_arm"       "skewness_pitch_arm"
##  [74,] "skewness_yaw_arm"         "skewness_yaw_arm"
##  [75,] "max_roll_arm"             "max_roll_arm"
##  [76,] "max_picth_arm"            "max_picth_arm"
##  [77,] "max_yaw_arm"              "max_yaw_arm"
##  [78,] "min_roll_arm"             "min_roll_arm"
```

```
##  [79,] "min_pitch_arm"            "min_pitch_arm"
##  [80,] "min_yaw_arm"              "min_yaw_arm"
##  [81,] "amplitude_roll_arm"       "amplitude_roll_arm"
##  [82,] "amplitude_pitch_arm"      "amplitude_pitch_arm"
##  [83,] "amplitude_yaw_arm"        "amplitude_yaw_arm"
##  [84,] "roll_dumbbell"            "roll_dumbbell"
##  [85,] "pitch_dumbbell"           "pitch_dumbbell"
##  [86,] "yaw_dumbbell"             "yaw_dumbbell"
##  [87,] "kurtosis_roll_dumbbell"   "kurtosis_roll_dumbbell"
##  [88,] "kurtosis_picth_dumbbell"  "kurtosis_picth_dumbbell"
##  [89,] "kurtosis_yaw_dumbbell"    "kurtosis_yaw_dumbbell"
##  [90,] "skewness_roll_dumbbell"   "skewness_roll_dumbbell"
##  [91,] "skewness_pitch_dumbbell"  "skewness_pitch_dumbbell"
##  [92,] "skewness_yaw_dumbbell"    "skewness_yaw_dumbbell"
##  [93,] "max_roll_dumbbell"        "max_roll_dumbbell"
##  [94,] "max_picth_dumbbell"       "max_picth_dumbbell"
##  [95,] "max_yaw_dumbbell"         "max_yaw_dumbbell"
##  [96,] "min_roll_dumbbell"        "min_roll_dumbbell"
##  [97,] "min_pitch_dumbbell"       "min_pitch_dumbbell"
##  [98,] "min_yaw_dumbbell"         "min_yaw_dumbbell"
##  [99,] "amplitude_roll_dumbbell"  "amplitude_roll_dumbbell"
## [100,] "amplitude_pitch_dumbbell" "amplitude_pitch_dumbbell"
## [101,] "amplitude_yaw_dumbbell"   "amplitude_yaw_dumbbell"
## [102,] "total_accel_dumbbell"     "total_accel_dumbbell"
## [103,] "var_accel_dumbbell"       "var_accel_dumbbell"
## [104,] "avg_roll_dumbbell"        "avg_roll_dumbbell"
## [105,] "stddev_roll_dumbbell"     "stddev_roll_dumbbell"
## [106,] "var_roll_dumbbell"        "var_roll_dumbbell"
## [107,] "avg_pitch_dumbbell"       "avg_pitch_dumbbell"
## [108,] "stddev_pitch_dumbbell"    "stddev_pitch_dumbbell"
## [109,] "var_pitch_dumbbell"       "var_pitch_dumbbell"
## [110,] "avg_yaw_dumbbell"         "avg_yaw_dumbbell"
## [111,] "stddev_yaw_dumbbell"      "stddev_yaw_dumbbell"
## [112,] "var_yaw_dumbbell"         "var_yaw_dumbbell"
## [113,] "gyros_dumbbell_x"         "gyros_dumbbell_x"
## [114,] "gyros_dumbbell_y"         "gyros_dumbbell_y"
## [115,] "gyros_dumbbell_z"         "gyros_dumbbell_z"
## [116,] "accel_dumbbell_x"         "accel_dumbbell_x"
## [117,] "accel_dumbbell_y"         "accel_dumbbell_y"
## [118,] "accel_dumbbell_z"         "accel_dumbbell_z"
## [119,] "magnet_dumbbell_x"        "magnet_dumbbell_x"
## [120,] "magnet_dumbbell_y"        "magnet_dumbbell_y"
## [121,] "magnet_dumbbell_z"        "magnet_dumbbell_z"
## [122,] "roll_forearm"             "roll_forearm"
## [123,] "pitch_forearm"            "pitch_forearm"
## [124,] "yaw_forearm"              "yaw_forearm"
## [125,] "kurtosis_roll_forearm"    "kurtosis_roll_forearm"
## [126,] "kurtosis_picth_forearm"   "kurtosis_picth_forearm"
## [127,] "kurtosis_yaw_forearm"     "kurtosis_yaw_forearm"
## [128,] "skewness_roll_forearm"    "skewness_roll_forearm"
```

```
## [129,] "skewness_pitch_forearm"   "skewness_pitch_forearm"
## [130,] "skewness_yaw_forearm"     "skewness_yaw_forearm"
## [131,] "max_roll_forearm"         "max_roll_forearm"
## [132,] "max_picth_forearm"        "max_picth_forearm"
## [133,] "max_yaw_forearm"          "max_yaw_forearm"
## [134,] "min_roll_forearm"         "min_roll_forearm"
## [135,] "min_pitch_forearm"        "min_pitch_forearm"
## [136,] "min_yaw_forearm"          "min_yaw_forearm"
## [137,] "amplitude_roll_forearm"   "amplitude_roll_forearm"
## [138,] "amplitude_pitch_forearm" "amplitude_pitch_forearm"
## [139,] "amplitude_yaw_forearm"    "amplitude_yaw_forearm"
## [140,] "total_accel_forearm"      "total_accel_forearm"
## [141,] "var_accel_forearm"        "var_accel_forearm"
## [142,] "avg_roll_forearm"         "avg_roll_forearm"
## [143,] "stddev_roll_forearm"      "stddev_roll_forearm"
## [144,] "var_roll_forearm"         "var_roll_forearm"
## [145,] "avg_pitch_forearm"        "avg_pitch_forearm"
## [146,] "stddev_pitch_forearm"     "stddev_pitch_forearm"
## [147,] "var_pitch_forearm"        "var_pitch_forearm"
## [148,] "avg_yaw_forearm"          "avg_yaw_forearm"
## [149,] "stddev_yaw_forearm"       "stddev_yaw_forearm"
## [150,] "var_yaw_forearm"          "var_yaw_forearm"
## [151,] "gyros_forearm_x"          "gyros_forearm_x"
## [152,] "gyros_forearm_y"          "gyros_forearm_y"
## [153,] "gyros_forearm_z"          "gyros_forearm_z"
## [154,] "accel_forearm_x"          "accel_forearm_x"
## [155,] "accel_forearm_y"          "accel_forearm_y"
## [156,] "accel_forearm_z"          "accel_forearm_z"
## [157,] "magnet_forearm_x"         "magnet_forearm_x"
## [158,] "magnet_forearm_y"         "magnet_forearm_y"
## [159,] "magnet_forearm_z"         "magnet_forearm_z"
## [160,] "classe"                   "problem_id"
```

From above exercise, we learnt that variable “classes” is available in the training data but not in the testing data.

To clean the data, we needed to eliminate those columns with near zero variance and columns with NA in both the training and testing data.

```
#removing the nzv in training data
nzvTraining <- nearZeroVar(pmlTraining)
training <- pmlTraining[ , -nzvTraining]

#removing the nzv in the testing data
nzvTesting <- nearZeroVar(pmlTesting)
testing <- pmlTesting[ , - nzvTesting]

#removing the NA in the training data
idenNATraining <- sapply(training, function(x) mean(is.na(x))>0.95)
training <- training[, idenNATraining == FALSE]
```

```r
#removing the NA in the testing data
idenNATesting <- sapply(testing, function(x) mean(is.na(x))>0.95)
pmlTraining <- testing[, idenNATesting == FALSE]

cbind(names(training), names(testing))
```

```
##       [,1]                    [,2]
##  [1,] "X"                     "X"
##  [2,] "user_name"             "user_name"
##  [3,] "raw_timestamp_part_1"  "raw_timestamp_part_1"
##  [4,] "raw_timestamp_part_2"  "raw_timestamp_part_2"
##  [5,] "cvtd_timestamp"        "cvtd_timestamp"
##  [6,] "num_window"            "num_window"
##  [7,] "roll_belt"             "roll_belt"
##  [8,] "pitch_belt"            "pitch_belt"
##  [9,] "yaw_belt"              "yaw_belt"
## [10,] "total_accel_belt"      "total_accel_belt"
## [11,] "gyros_belt_x"          "gyros_belt_x"
## [12,] "gyros_belt_y"          "gyros_belt_y"
## [13,] "gyros_belt_z"          "gyros_belt_z"
## [14,] "accel_belt_x"          "accel_belt_x"
## [15,] "accel_belt_y"          "accel_belt_y"
## [16,] "accel_belt_z"          "accel_belt_z"
## [17,] "magnet_belt_x"         "magnet_belt_x"
## [18,] "magnet_belt_y"         "magnet_belt_y"
## [19,] "magnet_belt_z"         "magnet_belt_z"
## [20,] "roll_arm"              "roll_arm"
## [21,] "pitch_arm"             "pitch_arm"
## [22,] "yaw_arm"               "yaw_arm"
## [23,] "total_accel_arm"       "total_accel_arm"
## [24,] "gyros_arm_x"           "gyros_arm_x"
## [25,] "gyros_arm_y"           "gyros_arm_y"
## [26,] "gyros_arm_z"           "gyros_arm_z"
## [27,] "accel_arm_x"           "accel_arm_x"
## [28,] "accel_arm_y"           "accel_arm_y"
## [29,] "accel_arm_z"           "accel_arm_z"
## [30,] "magnet_arm_x"          "magnet_arm_x"
## [31,] "magnet_arm_y"          "magnet_arm_y"
## [32,] "magnet_arm_z"          "magnet_arm_z"
## [33,] "roll_dumbbell"         "roll_dumbbell"
## [34,] "pitch_dumbbell"        "pitch_dumbbell"
## [35,] "yaw_dumbbell"          "yaw_dumbbell"
## [36,] "total_accel_dumbbell" "total_accel_dumbbell"
## [37,] "gyros_dumbbell_x"      "gyros_dumbbell_x"
## [38,] "gyros_dumbbell_y"      "gyros_dumbbell_y"
## [39,] "gyros_dumbbell_z"      "gyros_dumbbell_z"
## [40,] "accel_dumbbell_x"      "accel_dumbbell_x"
## [41,] "accel_dumbbell_y"      "accel_dumbbell_y"
## [42,] "accel_dumbbell_z"      "accel_dumbbell_z"
```

```
## [43,] "magnet_dumbbell_x"    "magnet_dumbbell_x"
## [44,] "magnet_dumbbell_y"    "magnet_dumbbell_y"
## [45,] "magnet_dumbbell_z"    "magnet_dumbbell_z"
## [46,] "roll_forearm"         "roll_forearm"
## [47,] "pitch_forearm"        "pitch_forearm"
## [48,] "yaw_forearm"          "yaw_forearm"
## [49,] "total_accel_forearm"  "total_accel_forearm"
## [50,] "gyros_forearm_x"      "gyros_forearm_x"
## [51,] "gyros_forearm_y"      "gyros_forearm_y"
## [52,] "gyros_forearm_z"      "gyros_forearm_z"
## [53,] "accel_forearm_x"      "accel_forearm_x"
## [54,] "accel_forearm_y"      "accel_forearm_y"
## [55,] "accel_forearm_z"      "accel_forearm_z"
## [56,] "magnet_forearm_x"     "magnet_forearm_x"
## [57,] "magnet_forearm_y"     "magnet_forearm_y"
## [58,] "magnet_forearm_z"     "magnet_forearm_z"
## [59,] "classe"               "problem_id"
```

We need to separate the training data for cross validation.

```
inTrain <- createDataPartition(training$classe, p = 0.7, list = FALSE)
training <- training[inTrain, ]
validation <- training[-inTrain,]
```

We noticed that the variable of "problem_id" in the test data was useless for prediction and should be removed. Moreover, we wanted to add a colume of "classe" in the test data for the prediction purpose.

```
testing <- testing[, 1:58]
classe <- rep(NA, nrow(testing))
testing <- cbind(testing, classe)
```

The first five variables in the data files are were not useful for prediction. We removed these columns from the files.

```
testing <- testing[, -(1:5)]
training <- training[, -(1:5)]
validation <- validation[, -(1:5)]
```

Having cleaned the data, we could have the prediction processes.

Prediction Processes

Prediction with Generalized Boosting Regression

```
trControl <- trainControl(method = "cv", number = 2)

modFit1 <- train(classe~., data = training, method = "gbm", trControl =
trControl, verbose = FALSE)
pred1 <- predict(modFit1, training)
```

```
result1 <- confusionMatrix(pred1, training$classe)
result1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 3899    8    0    1    0
##          B    7 2635   14    6    3
##          C    0   15 2378   16    2
##          D    0    0    2 2228   14
##          E    0    0    2    1 2506
##
## Overall Statistics
##
##                Accuracy : 0.9934
##                  95% CI : (0.9919, 0.9947)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9916
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9913   0.9925   0.9893   0.9925
## Specificity            0.9991   0.9973   0.9971   0.9986   0.9997
## Pos Pred Value         0.9977   0.9887   0.9863   0.9929   0.9988
## Neg Pred Value         0.9993   0.9979   0.9984   0.9979   0.9983
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2838   0.1918   0.1731   0.1622   0.1824
## Detection Prevalence   0.2845   0.1940   0.1755   0.1634   0.1826
## Balanced Accuracy      0.9986   0.9943   0.9948   0.9940   0.9961
```

Prediction with Random Forest

```
modFit2 <- train(classe~., data = training, method = "rf", trControl =
trControl)
pred2 <- predict(modFit2, training)
result2 <- confusionMatrix(pred2, training$classe)
result2
```

```
## 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
```

Prediction with Trees

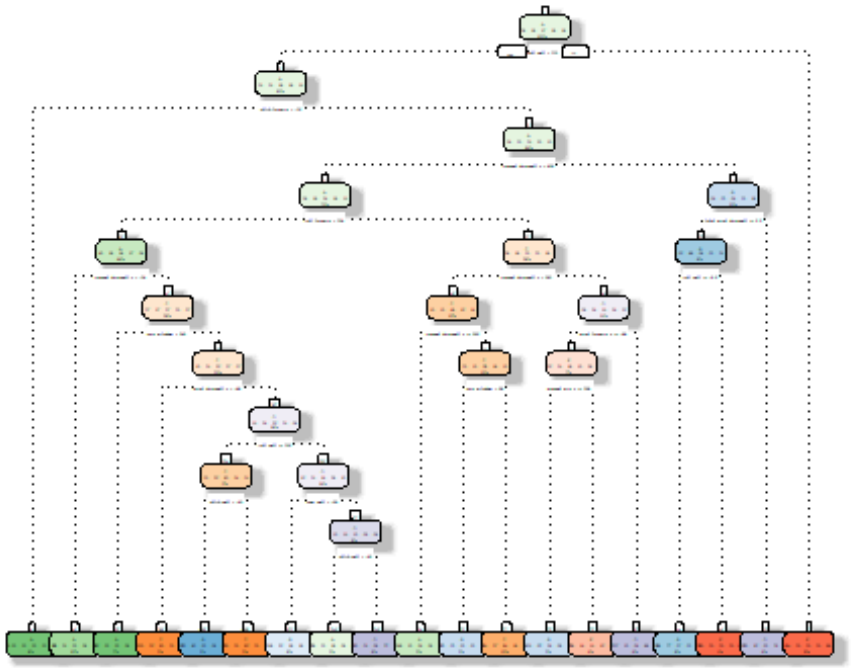
```
modFit3 <- rpart(classe~., data = training, method = "class")
pred3 <- predict(modFit3, newdata = training, type = "class")
result3 <- confusionMatrix(pred3, training$classe)
result3
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 3541  604  115  285  237
##          B  234 1659  305  199  317
##          C   23  150 1815  309  143
##          D   55  165  101 1375  208
##          E   53   80   60   84 1620
##
## Overall Statistics
##
##                Accuracy : 0.7287
##                  95% CI : (0.7212, 0.7361)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6538
```

```
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9066   0.6242   0.7575   0.6106   0.6416
## Specificity            0.8738   0.9048   0.9449   0.9539   0.9753
## Pos Pred Value         0.7405   0.6113   0.7439   0.7222   0.8540
## Neg Pred Value         0.9592   0.9094   0.9486   0.9259   0.9236
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2578   0.1208   0.1321   0.1001   0.1179
## Detection Prevalence   0.3481   0.1976   0.1776   0.1386   0.1381
## Balanced Accuracy      0.8902   0.7645   0.8512   0.7823   0.8084
```

```
#plot the tree
fancyRpartPlot(modFit3, caption = "Prediction with Tree")
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Prediction with Tree

Cross Validation.

From above process, we learnt that the random forest had the highest level of accuracy. The tree prediction had the fast speed. We decided use the random forest model to have cross validation.

```
predValid4 <- predict(modFit2, newdata = validation)
resultValid4 <- confusionMatrix(predValid4, validation$classe)
resultValid4
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1152    0    0    0    0
##          B    0  838    0    0    0
##          C    0    0  708    0    0
##          D    0    0    0  677    0
##          E    0    0    0    0  748
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9991, 1)
##     No Information Rate : 0.2794
##     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             0.2794   0.2033   0.1717   0.1642   0.1814
## Detection Rate         0.2794   0.2033   0.1717   0.1642   0.1814
## Detection Prevalence   0.2794   0.2033   0.1717   0.1642   0.1814
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

The results showed the random forest model prediction had a very high level of accuracy that we could apply it in the test data.

Prediction with Testing Data

We only have 20 observations in the testing data with the random forest.

```
predTest <- predict(modFit2, newdata = testing)
```

The prediction result was shown as follows.

```
predTest
```

```
##  [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

summary(predTest)

## A B C D E
## 7 8 1 1 3
```