Prediction Assignment

Aaron Bench

May 14, 2019

```
knitr::opts_chunk$set(echo = TRUE)

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(corrplot)

## corrplot 0.84 loaded

library(caret)

## Loading required package: lattice

library(rpart)
library(rpart.plot)
```

Executive Summary

Wearable Fitness tracking devices have become increasingly more accurate and popular over the last decade. This study uses data from accelerometers on the belt, forearm, arm and dumbell of six participants who were asked to perform a series of tasks correctly and incorrectly in five different ways. This study looks at how accuratly we can predict the manner in which the participants performed the exercise.

Data Ingest

```
setwd("C:/Users/aaron/OneDrive/Documents/Data_Science")
#Ingest data

pml_training <- read.csv("pml-training.csv")

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

Data Cleansing

Remove NA values and vectors that are not in the testing sample.

```
V <- names(pml_testing[,colSums(is.na(pml_testing)) == 0])[8:59] #variable list to Keep  variable columns from pml_testing without NAs from columns 8:59

pml_training <- pml_training[,c(V, "classe")]

pml_testing <- pml_testing[, c(V, "problem_id")]

head(pml_testing)

##   roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x gyros_belt_y
## 1    123.00      27.00    -4.75               20        -0.50        -0.02
## 2      1.02       4.87   -88.90                4        -0.06        -0.02
## 3      0.87       1.82   -88.50                5         0.05         0.02
## 4    125.00     -41.60   162.00               17         0.11         0.11
## 5      1.35       3.33   -88.60                3         0.03         0.02
## 6     -5.92       1.59   -87.70                4         0.10         0.05
##   gyros_belt_z accel_belt_x accel_belt_y accel_belt_z magnet_belt_x
## 1        -0.46          -38           69         -179           -13
## 2        -0.07          -13           11           39            43
## 3         0.03            1           -1           49            29
## 4        -0.16           46           45         -156           169
## 5         0.00           -8            4           27            33
## 6        -0.13          -11          -16           38            31
##   magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm
## 1           581          -382     40.7    -27.80     178              10
## 2           636          -309      0.0      0.00       0              38
## 3           631          -312      0.0      0.00       0              44
## 4           608          -304   -109.0     55.00    -142              25
## 5           566          -418     76.1      2.76     102              29
## 6           638          -291      0.0      0.00       0              14
##   gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z
## 1       -1.65        0.48       -0.18          16          38          93
## 2       -1.17        0.85       -0.43        -290         215         -90
## 3        2.10       -1.36        1.13        -341         245         -87
## 4        0.22       -0.51        0.92        -238         -57           6
## 5       -1.96        0.79       -0.54        -197         200         -30
## 6        0.02        0.05       -0.07         -26         130         -19
##   magnet_arm_x magnet_arm_y magnet_arm_z roll_dumbbell pitch_dumbbell
## 1         -326          385          481     -17.73748       24.96085
## 2         -325          447          434      54.47761      -53.69758
## 3         -264          474          413      57.07031      -51.37303
## 4         -173          257          633      43.10927      -30.04885
## 5         -170          275          617    -101.38396      -53.43952
## 6          396          176          516      62.18750      -50.55595
##   yaw_dumbbell total_accel_dumbbell gyros_dumbbell_x gyros_dumbbell_y
## 1    126.23596                    9             0.64             0.06
```

```
## 2    -75.51480                    31              0.34             0.05
## 3    -75.20287                    29              0.39             0.14
## 4   -103.32003                    18              0.10            -0.02
## 5    -14.19542                     4              0.29            -0.47
## 6    -71.12063                    29             -0.59             0.80
##   gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z
## 1            -0.61               21              -15               81
## 2            -0.71             -153              155             -205
## 3            -0.34             -141              155             -196
## 4             0.05              -51               72             -148
## 5            -0.46              -18              -30               -5
## 6             1.10             -138              166             -186
##   magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z roll_forearm
## 1               523              -528               -56          141
## 2              -502               388               -36          109
## 3              -506               349                41          131
## 4              -576               238                53            0
## 5              -424               252               312         -176
## 6              -543               262                96          150
##   pitch_forearm yaw_forearm total_accel_forearm gyros_forearm_x
## 1         49.30       156.0                  33            0.74
## 2        -17.60       106.0                  39            1.12
## 3        -32.60        93.0                  34            0.18
## 4          0.00         0.0                  43            1.38
## 5         -2.16       -47.9                  24           -0.75
## 6          1.46        89.7                  43           -0.88
##   gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y
## 1           -3.34           -0.59            -110             267
## 2           -2.78           -0.18             212             297
## 3           -0.79            0.28             154             271
## 4            0.69            1.80             -92             406
## 5            3.10            0.80             131             -93
## 6            4.26            1.35             230             322
##   accel_forearm_z magnet_forearm_x magnet_forearm_y magnet_forearm_z
## 1            -149             -714              419              617
## 2            -118             -237              791              873
## 3            -129              -51              698              783
## 4             -39             -233              783              521
## 5             172              375             -787               91
## 6            -144             -300              800              884
##   problem_id
## 1          1
## 2          2
## 3          3
## 4          4
## 5          5
## 6          6
```

```r
head(pml_training)
```

```
##   roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x gyros_belt_y
## 1      1.41       8.07    -94.4                3         0.00         0.00
## 2      1.41       8.07    -94.4                3         0.02         0.00
## 3      1.42       8.07    -94.4                3         0.00         0.00
## 4      1.48       8.05    -94.4                3         0.02         0.00
## 5      1.48       8.07    -94.4                3         0.02         0.02
## 6      1.45       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

#Now we have the same number of vectors for each data frame
```

Split the data

Next, we need to split our training data set. We'll split the training data set by 60% for training and 40% for testing.

```
set.seed(1234)
part_train <- caret::createDataPartition(pml_training$classe, p=0.06,
list=FALSE) #Randomly part data

training_df <- pml_training[part_train,] #Assign the 60% parted data

training_test_df <- pml_training[-part_train,] #assign the rest (40%) for
testing.  Not to be confused with the separate model test data
```

Machine learning models

Random Forest

I understand from class that the random forest model will almost always be better than a decision tree because it essentially does a variety of decision trees with random subsets of the data. I will proceed with a Random Forest model.

```
RF <- caret::train(classe ~., method='rf', data=training_df, ntree=128) #A
google search told me 128 is the max optimal number of trees

RF_PREDICT <- stats::predict(RF, training_test_df) #STATS from base R

RF_Conf <- caret::confusionMatrix(training_test_df$classe, RF_PREDICT)

RF_Conf

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 5119   46   35   40    5
##          B  223 3151  155   20   20
##          C   22  100 3054   38    2
##          D   19   13  162 2806   23
##          E    5   32  131   92 3130
##
## Overall Statistics
##
##                Accuracy : 0.9359
##                  95% CI : (0.9322, 0.9394)
##     No Information Rate : 0.2921
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9188
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9501   0.9428   0.8634   0.9366   0.9843
## Specificity            0.9903   0.9723   0.9891   0.9860   0.9830
## Pos Pred Value         0.9760   0.8829   0.9496   0.9282   0.9233
## Neg Pred Value         0.9796   0.9872   0.9683   0.9877   0.9967
## Prevalence             0.2921   0.1812   0.1918   0.1624   0.1724
## Detection Rate         0.2776   0.1709   0.1656   0.1521   0.1697
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9702   0.9576   0.9263   0.9613   0.9836
```

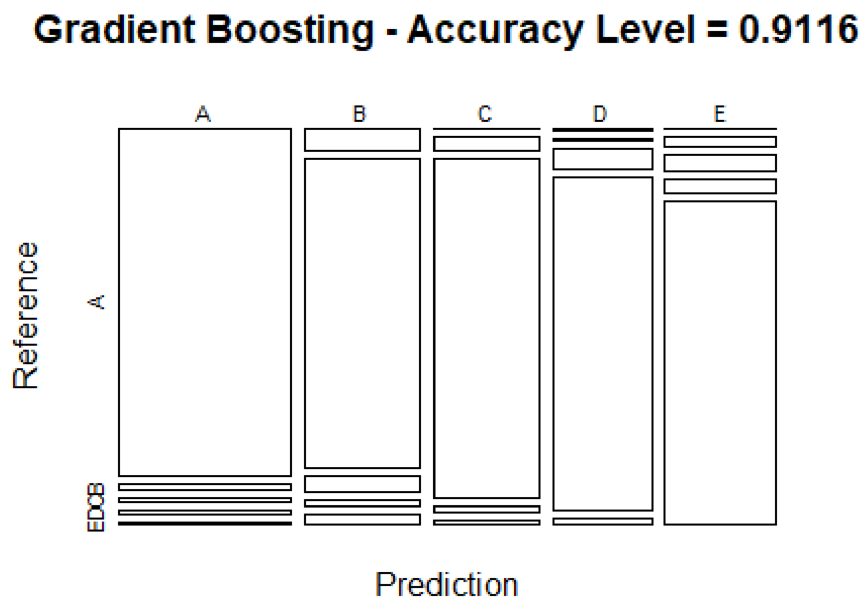
We get 94% accuracy with the Random Forest model. Next, I'll compare it to the Gradient Boosting Model.

Gradient Boosting Model (GBM)

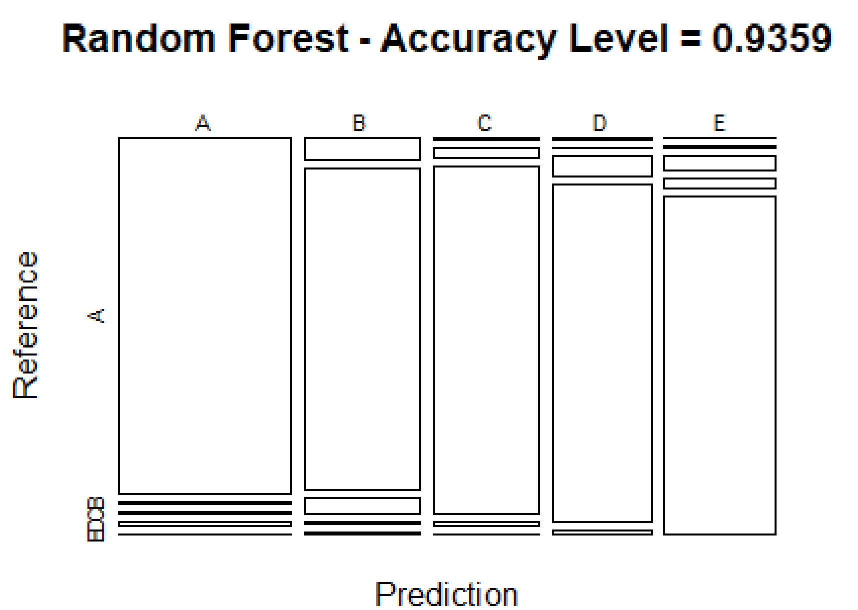
```
GBM <- caret::train(classe ~., method='gbm', data=training_df, verbose=FALSE)

GBM_PREDICT <- stats::predict(GBM, training_test_df)

GBM_Conf <- caret::confusionMatrix(training_test_df$classe, GBM_PREDICT)
GBM_Conf

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 5023   91   59   61   11
##          B  219 3026  177   46  101
##          C    1  129 3007   51   28
##          D   22   20  173 2754   54
##          E   15   91  142  139 3003
##
## Overall Statistics
##
##                Accuracy : 0.9116
##                  95% CI : (0.9074, 0.9157)
##     No Information Rate : 0.2863
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8882
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9513   0.9014   0.8451   0.9027   0.9393
## Specificity            0.9831   0.9640   0.9860   0.9825   0.9746
## Pos Pred Value         0.9577   0.8479   0.9350   0.9110   0.8858
## Neg Pred Value         0.9805   0.9777   0.9638   0.9807   0.9871
## Prevalence             0.2863   0.1820   0.1929   0.1654   0.1733
## Detection Rate         0.2724   0.1641   0.1630   0.1493   0.1628
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9672   0.9327   0.9155   0.9426   0.9570
```

Comparison Summary of Random Forest Vs. Gradient Boosting

```
plot(GBM_Conf$table, col = GBM_Conf$byClass,
     main = paste("Gradient Boosting - Accuracy Level =",
                  round(GBM_Conf$overall['Accuracy'], 4)))
```




```
plot(RF_Conf$table, col = RF_Conf$byClass,
     main = paste("Random Forest - Accuracy Level =",
                  round(RF_Conf$overall['Accuracy'], 4)))
```



Conclusion

The Random Forest model appears to be slightly stronger than the Gradient Boosting Model. The Random Forest model predicts Classe with nearly 94% accuracy in my training data set when splitting the training set data 60% train and 40% test.

Prediction on Testing data set

```
RF_predictOnTest <- stats::predict(RF, pml_testing)

RF_predictOnTest

##  [1] B A B A A E D D A A C C B A E E A B B B
## Levels: A B C D E
```