Predicting quality of exercise movements using data from accelerometers

Andrea Vallebueno

February 12, 2019

Executive summary

This report has the objective of fitting a model that predicts the quality of an activity performed at a specific point in time. We will fit a a classification tree and a random forest and explain the process of model selection.

It uses the Weight Lifting Exercises Dataset, which investigates how well an activity was performed by the wearer of accelerometers. For this dataset, six participants were 'asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five 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) and throwing the hips to the front (Class E)." While Class A identifies the correct, specified execution of the exercise, the other four classes capture common mistakes made. (Velloso, 2013)

We find that the random forest model presents a superior performance with an accuracy of 100% on the test set.

Setting up the environment

Let us begin by loading the necessary libraries.

```
library(caret)
library(ggplot2)
library(parallel)
library(doParallel)
library(dplyr)
library(ggraph)
library(igraph)
```

Exploratory Analysis

We begin by reading the training and testing data into R and observe that the training dataset consists of 19,622 observations of 160 variables. These belong to six subjects, as per the user_name variable, producing five different movements (A, B, C, D or E), as per the classe variable. This type of movement is the variable that we shall seek to predict with our model.

```
training<- read.csv("pml-training.csv")
testing<- read.csv("pml-testing.csv")
str(training, strict.width = 'wrap')
```

```
## 'data.frame':    19622 obs. of  160 variables:
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...
## $ user_name : Factor w/ 6 levels "adelmo","carlitos",..: 2 2 2 2 2 2 2 2 2 2
```

```
##    2 ...
## $ raw_timestamp_part_1 : int 1323084231 1323084231 1323084231 1323084232
##    1323084232 1323084232 1323084232 1323084232 1323084232 1323084232 ...
## $ raw_timestamp_part_2 : int 788290 808298 820366 120339 196328 304277
##    368296 440390 484323 484434 ...
## $ cvtd_timestamp : Factor w/ 20 levels "02/12/2011 13:32",..: 9 9 9 9 9 9
##    9 9 9 9 ...
## $ new_window : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ num_window : int 11 11 11 12 12 12 12 12 12 12 ...
## $ roll_belt : num 1.41 1.41 1.42 1.48 1.48 1.45 1.42 1.42 1.43 1.45 ...
## $ pitch_belt : num 8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.17 ...
## $ yaw_belt : num -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4
##    -94.4 ...
## $ total_accel_belt : int 3 3 3 3 3 3 3 3 3 3 ...
## $ kurtosis_roll_belt : Factor w/ 397 levels "","-0.016850",..: 1 1 1 1 1 1
##    1 1 1 1 ...
## $ kurtosis_picth_belt : Factor w/ 317 levels "","-0.021887",..: 1 1 1 1 1
##    1 1 1 1 1 ...
## $ kurtosis_yaw_belt : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1
##    ...
## $ skewness_roll_belt : Factor w/ 395 levels "","-0.003095",..: 1 1 1 1 1 1
##    1 1 1 1 ...
## $ skewness_roll_belt.1 : Factor w/ 338 levels "","-0.005928",..: 1 1 1 1 1
##    1 1 1 1 1 ...
## $ skewness_yaw_belt : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1
##    ...
## $ max_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ max_picth_belt : int NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_belt : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1
##    1 1 ...
## $ min_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_belt : int NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_belt : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1
##    1 1 ...
## $ amplitude_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_belt : int NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_belt : Factor w/ 4 levels "","#DIV/0!","0.00",..: 1 1 1 1
##    1 1 1 1 1 1 ...
## $ var_total_accel_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ gyros_belt_x : num 0 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.03 ...
## $ gyros_belt_y : num 0 0 0 0 0.02 0 0 0 0 0 ...
## $ gyros_belt_z : num -0.02 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02
##    0 ...
## $ accel_belt_x : int -21 -22 -20 -22 -21 -21 -22 -22 -20 -21 ...
## $ accel_belt_y : int 4 4 5 3 2 4 3 4 2 4 ...
```

```
## $ accel_belt_z : int 22 22 23 21 24 21 21 21 24 22 ...
## $ magnet_belt_x : int -3 -7 -2 -6 -6 0 -4 -2 1 -3 ...
## $ magnet_belt_y : int 599 608 600 604 600 603 599 603 602 609 ...
## $ magnet_belt_z : int -313 -311 -305 -310 -302 -312 -311 -313 -312 -308
##    ...
## $ roll_arm : num -128 -128 -128 -128 -128 -128 -128 -128 -128 -128 ...
## $ pitch_arm : num 22.5 22.5 22.5 22.1 22.1 22 21.9 21.8 21.7 21.6 ...
## $ yaw_arm : num -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...
## $ total_accel_arm : int 34 34 34 34 34 34 34 34 34 34 ...
## $ var_accel_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ gyros_arm_x : num 0 0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02 ...
## $ gyros_arm_y : num 0 -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.02 -0.03
##    -0.03 ...
## $ gyros_arm_z : num -0.02 -0.02 -0.02 0.02 0 0 0 0 -0.02 -0.02 ...
## $ accel_arm_x : int -288 -290 -289 -289 -289 -289 -289 -289 -288 -288 ...
## $ accel_arm_y : int 109 110 110 111 111 111 111 111 109 110 ...
## $ accel_arm_z : int -123 -125 -126 -123 -123 -122 -125 -124 -122 -124 ...
## $ magnet_arm_x : int -368 -369 -368 -372 -374 -369 -373 -372 -369 -376 ...
## $ magnet_arm_y : int 337 337 344 344 337 342 336 338 341 334 ...
## $ magnet_arm_z : int 516 513 513 512 506 513 509 510 518 516 ...
## $ kurtosis_roll_arm : Factor w/ 330 levels "","-0.02438",..: 1 1 1 1 1 1 1
##    1 1 1 ...
## $ kurtosis_picth_arm : Factor w/ 328 levels "","-0.00484",..: 1 1 1 1 1 1
##    1 1 1 1 ...
## $ kurtosis_yaw_arm : Factor w/ 395 levels "","-0.01548",..: 1 1 1 1 1 1 1
##    1 1 1 ...
## $ skewness_roll_arm : Factor w/ 331 levels "","-0.00051",..: 1 1 1 1 1 1 1
##    1 1 1 ...
## $ skewness_pitch_arm : Factor w/ 328 levels "","-0.00184",..: 1 1 1 1 1 1
##    1 1 1 1 ...
## $ skewness_yaw_arm : Factor w/ 395 levels "","-0.00311",..: 1 1 1 1 1 1 1
##    1 1 1 ...
## $ max_roll_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ max_picth_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_arm : int NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_arm : int NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_arm : num NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_arm : int NA NA NA NA NA NA NA NA NA NA ...
## $ roll_dumbbell : num 13.1 13.1 12.9 13.4 13.4 ...
## $ pitch_dumbbell : num -70.5 -70.6 -70.3 -70.4 -70.4 ...
## $ yaw_dumbbell : num -84.9 -84.7 -85.1 -84.9 -84.9 ...
## $ kurtosis_roll_dumbbell : Factor w/ 398 levels "","-0.0035","-0.0073",..:
```

```
##    1 1 1 1 1 1 1 1 1 1 ...
## $ kurtosis_picth_dumbbell : Factor w/ 401 levels
##    "","-0.0163","-0.0233",..: 1 1 1 1 1 1 1 1 1 1 ...
## $ kurtosis_yaw_dumbbell : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1
##    1 1 ...
## $ skewness_roll_dumbbell : Factor w/ 401 levels "","-0.0082","-0.0096",..:
##    1 1 1 1 1 1 1 1 1 1 ...
## $ skewness_pitch_dumbbell : Factor w/ 402 levels
##    "","-0.0053","-0.0084",..: 1 1 1 1 1 1 1 1 1 1 ...
## $ skewness_yaw_dumbbell : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1
##    1 1 ...
## $ max_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
## $ max_picth_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_dumbbell : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1
##    1 1 1 1 ...
## $ min_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_dumbbell : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1
##    1 1 1 1 ...
## $ amplitude_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
## [list output truncated]
```

Given the large size of the training set, we will create a validation set and then proceed to the exploratory analysis on our resulting training set.

```
set.seed(411)
inTrain <- createDataPartition(y = training$classe, p = 0.8, list = FALSE)
trainingset <- training[inTrain,]
validationset<- training[-inTrain,]
```

Our next step will be to check for missing values and re-shape our dataset accordingly. We notice that there are several NA values and proceed to count them in order to identify which variables are useful for prediction.

```
na_values <- is.na(trainingset)
col_nas <- apply(na_values, sum, MARGIN = 2)
print('Number of missing values per column')
```

```
## [1] "Number of missing values per column"
```

```
print(col_nas)
```

```
##                         X                 user_name      raw_timestamp_part_1
##                         0                         0                         0
##      raw_timestamp_part_2            cvtd_timestamp                new_window
##                         0                         0                         0
##                num_window                roll_belt               pitch_belt
##                         0                         0                         0
##                 yaw_belt         total_accel_belt       kurtosis_roll_belt
##                         0                         0                         0
##       kurtosis_picth_belt        kurtosis_yaw_belt       skewness_roll_belt
##                         0                         0                         0
```

```
##     skewness_roll_belt.1        skewness_yaw_belt            max_roll_belt
##                        0                        0                    15374
##           max_picth_belt             max_yaw_belt            min_roll_belt
##                    15374                        0                    15374
##           min_pitch_belt             min_yaw_belt      amplitude_roll_belt
##                    15374                        0                    15374
##     amplitude_pitch_belt       amplitude_yaw_belt     var_total_accel_belt
##                    15374                        0                    15374
##            avg_roll_belt         stddev_roll_belt            var_roll_belt
##                    15374                    15374                    15374
##            avg_pitch_belt         stddev_pitch_belt            var_pitch_belt
##                    15374                    15374                    15374
##              avg_yaw_belt           stddev_yaw_belt              var_yaw_belt
##                    15374                    15374                    15374
##              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            var_accel_arm            avg_roll_arm
##                        0                    15374                    15374
##           stddev_roll_arm             var_roll_arm            avg_pitch_arm
##                    15374                    15374                    15374
##          stddev_pitch_arm            var_pitch_arm              avg_yaw_arm
##                    15374                    15374                    15374
##           stddev_yaw_arm             var_yaw_arm              gyros_arm_x
##                    15374                    15374                        0
##              gyros_arm_y              gyros_arm_z              accel_arm_x
##                        0                        0                        0
##              accel_arm_y              accel_arm_z             magnet_arm_x
##                        0                        0                        0
##             magnet_arm_y             magnet_arm_z        kurtosis_roll_arm
##                        0                        0                        0
##        kurtosis_picth_arm         kurtosis_yaw_arm        skewness_roll_arm
##                        0                        0                        0
##        skewness_pitch_arm         skewness_yaw_arm             max_roll_arm
##                        0                        0                    15374
##             max_picth_arm              max_yaw_arm             min_roll_arm
##                    15374                    15374                    15374
##             min_pitch_arm              min_yaw_arm       amplitude_roll_arm
##                    15374                    15374                    15374
##       amplitude_pitch_arm        amplitude_yaw_arm            roll_dumbbell
##                    15374                    15374                        0
##            pitch_dumbbell             yaw_dumbbell   kurtosis_roll_dumbbell
##                        0                        0                        0
##  kurtosis_picth_dumbbell    kurtosis_yaw_dumbbell   skewness_roll_dumbbell
##                        0                        0                        0
##  skewness_pitch_dumbbell    skewness_yaw_dumbbell        max_roll_dumbbell
##                        0                        0                    15374
##        max_picth_dumbbell         max_yaw_dumbbell        min_roll_dumbbell
##                    15374                        0                    15374
```

```
##       min_pitch_dumbbell         min_yaw_dumbbell  amplitude_roll_dumbbell
##                    15374                        0                    15374
## amplitude_pitch_dumbbell   amplitude_yaw_dumbbell     total_accel_dumbbell
##                    15374                        0                        0
##       var_accel_dumbbell        avg_roll_dumbbell     stddev_roll_dumbbell
##                    15374                    15374                    15374
##        var_roll_dumbbell       avg_pitch_dumbbell    stddev_pitch_dumbbell
##                    15374                    15374                    15374
##       var_pitch_dumbbell         avg_yaw_dumbbell      stddev_yaw_dumbbell
##                    15374                    15374                    15374
##         var_yaw_dumbbell         gyros_dumbbell_x         gyros_dumbbell_y
##                    15374                        0                        0
##         gyros_dumbbell_z         accel_dumbbell_x         accel_dumbbell_y
##                        0                        0                        0
##         accel_dumbbell_z        magnet_dumbbell_x        magnet_dumbbell_y
##                        0                        0                        0
##        magnet_dumbbell_z             roll_forearm            pitch_forearm
##                        0                        0                        0
##              yaw_forearm    kurtosis_roll_forearm   kurtosis_picth_forearm
##                        0                        0                        0
##     kurtosis_yaw_forearm    skewness_roll_forearm   skewness_pitch_forearm
##                        0                        0                        0
##     skewness_yaw_forearm        max_roll_forearm        max_picth_forearm
##                        0                    15374                    15374
##          max_yaw_forearm        min_roll_forearm        min_pitch_forearm
##                        0                    15374                    15374
##          min_yaw_forearm   amplitude_roll_forearm  amplitude_pitch_forearm
##                        0                    15374                    15374
##    amplitude_yaw_forearm      total_accel_forearm        var_accel_forearm
##                        0                        0                    15374
##         avg_roll_forearm      stddev_roll_forearm         var_roll_forearm
##                    15374                    15374                    15374
##        avg_pitch_forearm     stddev_pitch_forearm        var_pitch_forearm
##                    15374                    15374                    15374
##          avg_yaw_forearm       stddev_yaw_forearm          var_yaw_forearm
##                    15374                    15374                    15374
##          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
##                        0
```

```
col_nas_mean <- apply(na_values, mean, MARGIN = 2)
print('Percentage of missing values per column')
```

```
## [1] "Percentage of missing values per column"
```

```
print(col_nas_mean)
```

```
##                        X                user_name     raw_timestamp_part_1
```

```
##                0.000000                0.000000                0.000000
##     raw_timestamp_part_2          cvtd_timestamp              new_window
##                0.000000                0.000000                0.000000
##              num_window             roll_belt            pitch_belt
##                0.000000                0.000000                0.000000
##              yaw_belt        total_accel_belt      kurtosis_roll_belt
##                0.000000                0.000000                0.000000
##     kurtosis_picth_belt       kurtosis_yaw_belt      skewness_roll_belt
##                0.000000                0.000000                0.000000
##    skewness_roll_belt.1       skewness_yaw_belt           max_roll_belt
##                0.000000                0.000000                0.979298
##           max_picth_belt           max_yaw_belt           min_roll_belt
##                0.979298                0.000000                0.979298
##           min_pitch_belt           min_yaw_belt     amplitude_roll_belt
##                0.979298                0.000000                0.979298
##    amplitude_pitch_belt      amplitude_yaw_belt     var_total_accel_belt
##                0.979298                0.000000                0.979298
##            avg_roll_belt        stddev_roll_belt           var_roll_belt
##                0.979298                0.979298                0.979298
##           avg_pitch_belt       stddev_pitch_belt          var_pitch_belt
##                0.979298                0.979298                0.979298
##             avg_yaw_belt         stddev_yaw_belt            var_yaw_belt
##                0.979298                0.979298                0.979298
##            gyros_belt_x            gyros_belt_y            gyros_belt_z
##                0.000000                0.000000                0.000000
##            accel_belt_x            accel_belt_y            accel_belt_z
##                0.000000                0.000000                0.000000
##           magnet_belt_x           magnet_belt_y           magnet_belt_z
##                0.000000                0.000000                0.000000
##                roll_arm               pitch_arm                 yaw_arm
##                0.000000                0.000000                0.000000
##          total_accel_arm           var_accel_arm            avg_roll_arm
##                0.000000                0.979298                0.979298
##          stddev_roll_arm            var_roll_arm           avg_pitch_arm
##                0.979298                0.979298                0.979298
##         stddev_pitch_arm           var_pitch_arm             avg_yaw_arm
##                0.979298                0.979298                0.979298
##           stddev_yaw_arm             var_yaw_arm            gyros_arm_x
##                0.979298                0.979298                0.000000
##              gyros_arm_y             gyros_arm_z            accel_arm_x
##                0.000000                0.000000                0.000000
##              accel_arm_y             accel_arm_z            magnet_arm_x
##                0.000000                0.000000                0.000000
##             magnet_arm_y            magnet_arm_z       kurtosis_roll_arm
##                0.000000                0.000000                0.000000
##       kurtosis_picth_arm        kurtosis_yaw_arm       skewness_roll_arm
##                0.000000                0.000000                0.000000
##       skewness_pitch_arm        skewness_yaw_arm            max_roll_arm
##                0.000000                0.000000                0.979298
##            max_picth_arm             max_yaw_arm            min_roll_arm
##                0.979298                0.979298                0.979298
##            min_pitch_arm             min_yaw_arm      amplitude_roll_arm
##                0.979298                0.979298                0.979298
##      amplitude_pitch_arm       amplitude_yaw_arm           roll_dumbbell
```

```
##                0.979298                0.979298                0.000000
##           pitch_dumbbell            yaw_dumbbell   kurtosis_roll_dumbbell
##                0.000000                0.000000                0.000000
##  kurtosis_picth_dumbbell   kurtosis_yaw_dumbbell   skewness_roll_dumbbell
##                0.000000                0.000000                0.000000
##  skewness_pitch_dumbbell   skewness_yaw_dumbbell        max_roll_dumbbell
##                0.000000                0.000000                0.979298
##       max_picth_dumbbell         max_yaw_dumbbell        min_roll_dumbbell
##                0.979298                0.000000                0.979298
##       min_pitch_dumbbell         min_yaw_dumbbell  amplitude_roll_dumbbell
##                0.979298                0.000000                0.979298
## amplitude_pitch_dumbbell   amplitude_yaw_dumbbell     total_accel_dumbbell
##                0.979298                0.000000                0.000000
##       var_accel_dumbbell        avg_roll_dumbbell     stddev_roll_dumbbell
##                0.979298                0.979298                0.979298
##        var_roll_dumbbell       avg_pitch_dumbbell    stddev_pitch_dumbbell
##                0.979298                0.979298                0.979298
##       var_pitch_dumbbell         avg_yaw_dumbbell      stddev_yaw_dumbbell
##                0.979298                0.979298                0.979298
##         var_yaw_dumbbell         gyros_dumbbell_x         gyros_dumbbell_y
##                0.979298                0.000000                0.000000
##         gyros_dumbbell_z         accel_dumbbell_x         accel_dumbbell_y
##                0.000000                0.000000                0.000000
##         accel_dumbbell_z        magnet_dumbbell_x        magnet_dumbbell_y
##                0.000000                0.000000                0.000000
##        magnet_dumbbell_z             roll_forearm            pitch_forearm
##                0.000000                0.000000                0.000000
##              yaw_forearm    kurtosis_roll_forearm   kurtosis_picth_forearm
##                0.000000                0.000000                0.000000
##     kurtosis_yaw_forearm    skewness_roll_forearm   skewness_pitch_forearm
##                0.000000                0.000000                0.000000
##     skewness_yaw_forearm         max_roll_forearm        max_picth_forearm
##                0.000000                0.979298                0.979298
##          max_yaw_forearm         min_roll_forearm        min_pitch_forearm
##                0.000000                0.979298                0.979298
##          min_yaw_forearm   amplitude_roll_forearm  amplitude_pitch_forearm
##                0.000000                0.979298                0.979298
##    amplitude_yaw_forearm      total_accel_forearm        var_accel_forearm
##                0.000000                0.000000                0.979298
##         avg_roll_forearm      stddev_roll_forearm         var_roll_forearm
##                0.979298                0.979298                0.979298
##        avg_pitch_forearm     stddev_pitch_forearm        var_pitch_forearm
##                0.979298                0.979298                0.979298
##          avg_yaw_forearm       stddev_yaw_forearm          var_yaw_forearm
##                0.979298                0.979298                0.979298
##          gyros_forearm_x         gyros_forearm_y         gyros_forearm_z
##                0.000000                0.000000                0.000000
##          accel_forearm_x         accel_forearm_y         accel_forearm_z
##                0.000000                0.000000                0.000000
##         magnet_forearm_x        magnet_forearm_y        magnet_forearm_z
##                0.000000                0.000000                0.000000
##                   classe
##                0.000000
```

```
high_missing_values <- col_nas_mean[col_nas_mean > 0.10]
print('Number of variables with percentage of missing values higher than 10%')
```

```
## [1] "Number of variables with percentage of missing values higher than 10%"
```

```
length(high_missing_values)
```

```
## [1] 67
```

We observe that 67 variables have a missing value rate of 97.9% or higher, indicating that these may not be very useful for prediction. Moreover, we observe that in the cases where these variables have a value of "NA", 34 other variables have a blank value. From here on, we will only focus on the remaining variables as there is not enough information on the previously mentioned variables to use these for prediction.

```
NAvalues <- NULL
for (i in 1:160) {
  if (mean(is.na(trainingset[,i])) != 0) {
    NAvalues[i]<-i
  } else {NAvalues[i]<-0}

}

Blankvalues<-NULL
for (i in 1:160) {
  if (class(trainingset[1, i]) == "factor" & trainingset[1, i] == "") {
    Blankvalues[i]<-i
  } else {
    Blankvalues[i]<-0
    }
}
combined<-c(NAvalues, Blankvalues)
combined<-combined[combined != 0]
newtrainingset<- trainingset[, -combined]
```

We check that we have maintained variables with enough variability with the following zero covariate analysis.

```
nsv<-nearZeroVar(newtrainingset, saveMetrics = TRUE)
nsv
```

```
##                      freqRatio percentUnique zeroVar   nzv
## X                     1.000000  100.00000000   FALSE FALSE
## user_name             1.105226    0.03821899   FALSE FALSE
## raw_timestamp_part_1  1.032258    5.33154978   FALSE FALSE
## raw_timestamp_part_2  1.000000   88.17122110   FALSE FALSE
## cvtd_timestamp        1.012510    0.12739665   FALSE FALSE
## new_window           47.304615    0.01273966   FALSE  TRUE
## num_window            1.032258    5.46531626   FALSE FALSE
## roll_belt             1.093185    7.60557997   FALSE FALSE
## pitch_belt            1.039216   11.10261800   FALSE FALSE
## yaw_belt              1.114796   11.83514874   FALSE FALSE
## total_accel_belt      1.065337    0.18472514   FALSE FALSE
```

```
## gyros_belt_x          1.068493    0.84718772   FALSE FALSE
## gyros_belt_y          1.141084    0.42677878   FALSE FALSE
## gyros_belt_z          1.045614    1.05739219   FALSE FALSE
## accel_belt_x          1.038400    1.00643353   FALSE FALSE
## accel_belt_y          1.094249    0.89814638   FALSE FALSE
## accel_belt_z          1.094964    1.87273075   FALSE FALSE
## magnet_belt_x         1.065517    1.96827823   FALSE FALSE
## magnet_belt_y         1.123552    1.83451175   FALSE FALSE
## magnet_belt_z         1.015707    2.80272629   FALSE FALSE
## roll_arm             50.111111   15.90547169   FALSE FALSE
## pitch_arm            90.200000   18.52347283   FALSE FALSE
## yaw_arm              29.736264   17.23039684   FALSE FALSE
## total_accel_arm       1.048951    0.42040894   FALSE FALSE
## gyros_arm_x           1.034568    4.00025479   FALSE FALSE
## gyros_arm_y           1.459854    2.35046818   FALSE FALSE
## gyros_arm_z           1.099057    1.49691063   FALSE FALSE
## accel_arm_x           1.028777    4.89203134   FALSE FALSE
## accel_arm_y           1.044444    3.38875088   FALSE FALSE
## accel_arm_z           1.028846    4.91114084   FALSE FALSE
## magnet_arm_x          1.014493    8.47824702   FALSE FALSE
## magnet_arm_y          1.040541    5.49716542   FALSE FALSE
## magnet_arm_z          1.044944    8.00687942   FALSE FALSE
## roll_dumbbell         1.067308   85.85260208   FALSE FALSE
## pitch_dumbbell        2.375000   83.54035289   FALSE FALSE
## yaw_dumbbell          1.106383   85.39397414   FALSE FALSE
## total_accel_dumbbell  1.089483    0.26753296   FALSE FALSE
## gyros_dumbbell_x      1.042194    1.51602013   FALSE FALSE
## gyros_dumbbell_y      1.263948    1.74533410   FALSE FALSE
## gyros_dumbbell_z      1.096774    1.27396649   FALSE FALSE
## accel_dumbbell_x      1.018797    2.64348048   FALSE FALSE
## accel_dumbbell_y      1.052083    2.93012294   FALSE FALSE
## accel_dumbbell_z      1.174603    2.56067265   FALSE FALSE
## magnet_dumbbell_x     1.042553    6.95585706   FALSE FALSE
## magnet_dumbbell_y     1.183824    5.33154978   FALSE FALSE
## magnet_dumbbell_z     1.019355    4.26778776   FALSE FALSE
## roll_forearm         11.227437   12.65048729   FALSE FALSE
## pitch_forearm        64.770833   17.37053315   FALSE FALSE
## yaw_forearm          14.804762   11.75234091   FALSE FALSE
## total_accel_forearm   1.126646    0.44588827   FALSE FALSE
## gyros_forearm_x       1.043880    1.84725142   FALSE FALSE
## gyros_forearm_y       1.034700    4.64997771   FALSE FALSE
## gyros_forearm_z       1.163636    1.87273075   FALSE FALSE
## accel_forearm_x       1.092105    4.98757883   FALSE FALSE
## accel_forearm_y       1.024691    6.28702465   FALSE FALSE
## accel_forearm_z       1.016807    3.62443468   FALSE FALSE
## magnet_forearm_x      1.045455    9.49105039   FALSE FALSE
## magnet_forearm_y      1.130435   11.74597108   FALSE FALSE
## magnet_forearm_z      1.041667   10.42104593   FALSE FALSE
## classe                1.469388    0.03184916   FALSE FALSE
```

Given the number of variables, we calculate a correlation matrix on the numeric variables. We observe that several variables have a correlation higher than 0.80.

```
correlations<- abs(cor(newtrainingset[, -c(1, 2 ,3 , 4 ,5 , 6, 60)]))
diag(correlations)<-0
correlations<- as.data.frame(correlations)
subcor<-correlations[correlations > 0.8]
subcor
```

```
##  [1] 0.8156068 0.9811437 0.9256751 0.9920028 0.9659760 0.8867352 0.8156068
##  [8] 0.9811437 0.9280589 0.9750715 0.9659760 0.8929742 0.9256751 0.9280589
## [15] 0.9340529 0.9920028 0.9750715 0.9340529 0.8867352 0.8929742 0.9182593
## [22] 0.9182593 0.8132600 0.8132600 0.8136752 0.8136752 0.8065444 0.8490474
## [29] 0.9828042 0.9301233 0.9828042 0.9454978 0.8065444 0.8490474 0.8647177
## [36] 0.9301233 0.9454978 0.8647177
```

Model selection

Since our objective is to predict a factor variable with 5 levels, we will focus on non-linear models. We will start with a classification tree to get an idea of accuracy metrics.

```
set.seed(411)
model1<- train(classe ~ . , data = newtrainingset, method = "rpart")
print(model1$finalModel)
```

```
## n= 15699
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 15699 11235 A (0.28 0.19 0.17 0.16 0.18)
##   2) X< 5580.5 4464     0 A (1 0 0 0 0) *
##   3) X>=5580.5 11235  8197 B (0 0.27 0.24 0.23 0.26)
##     6) X< 9377.5 3038     0 B (0 1 0 0 0) *
##     7) X>=9377.5 8197  5311 E (0 0 0.33 0.31 0.35) *
```

```
model1predictions<- predict(model1, newdata = validationset)
confusionMatrix(validationset$classe, model1predictions)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    0    0    0    0
##          B    0  759    0    0    0
##          C    0    0    0    0  684
##          D    0    0    0    0  643
##          E    0    0    0    0  721
##
## Overall Statistics
##
##                Accuracy : 0.6617
##                  95% CI : (0.6467, 0.6765)
##     No Information Rate : 0.522
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5695
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000       NA       NA   0.3521
## Specificity            1.0000   1.0000   0.8256   0.8361   1.0000
## Pos Pred Value         1.0000   1.0000       NA       NA   1.0000
## Neg Pred Value         1.0000   1.0000       NA       NA   0.5856
## Prevalence             0.2845   0.1935   0.0000   0.0000   0.5220
## Detection Rate         0.2845   0.1935   0.0000   0.0000   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000       NA       NA   0.6760
```

We note that this first model (Model 1) has 15,699 nodes and a relatively low accuracy of 66% on the validation set. As detailed in the confusion matrix, the model correctly classifies A, B and E classes, but incorrectly classifies as E all those observations from the C and D classes.

We now fit a second model, a random forest model, to improve our accuracy.

```
cluster <- makeCluster(detectCores() - 1)
registerDoParallel(cluster)
fitControl <- trainControl(method = "cv", number = 5, allowParallel = TRUE)
model2 <- train(classe ~ . , data = newtrainingset[, 2:60], method = "rf",
                trControl = fitControl)
stopCluster(cluster)
registerDoSEQ()

print(model2$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 41
##
##         OOB estimate of  error rate: 0.06%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4464    0    0    0    0 0.0000000000
## B    3 3035    0    0    0 0.0009874918
## C    0    3 2735    0    0 0.0010956903
## D    0    0    3 2570    0 0.0011659541
## E    0    0    0    1 2885 0.0003465003
```

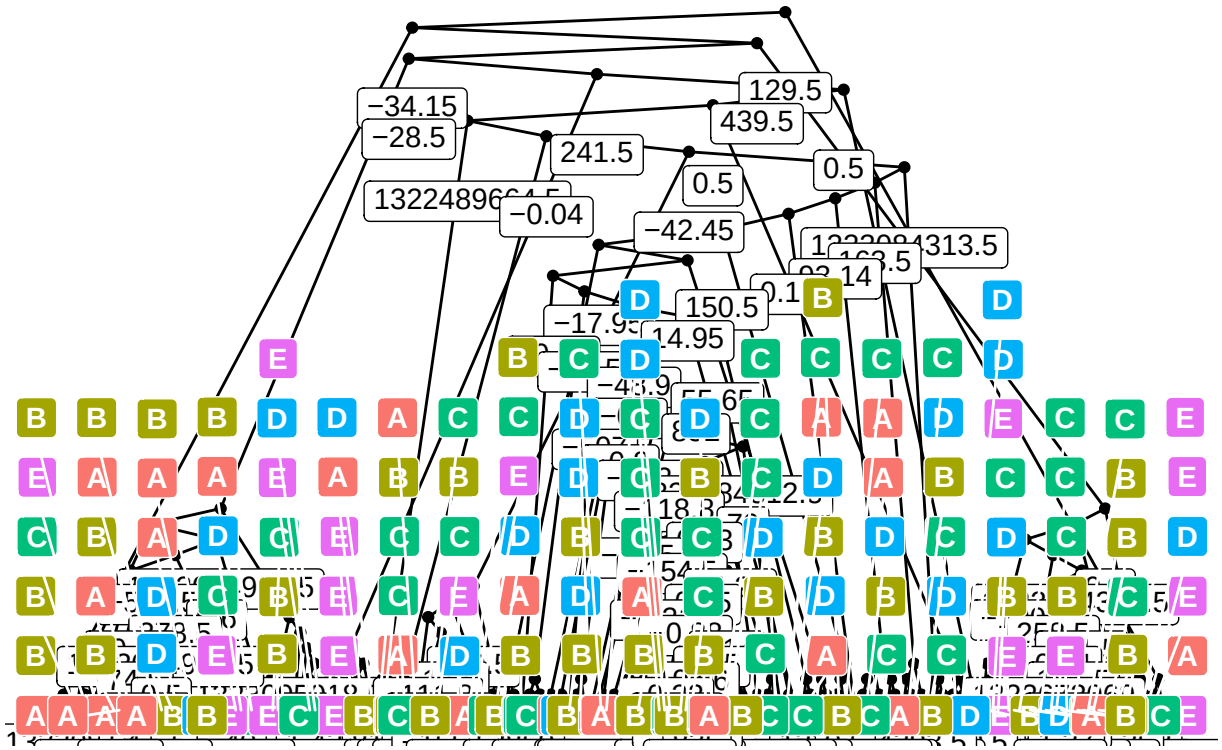
```
model2predictions <- predict(model2, newdata = validationset)
confusionMatrix(validationset$classe, model2predictions)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    0    0    0    0
##          B    2  757    0    0    0
##          C    0    0  684    0    0
##          D    0    0    1  641    1
##          E    0    0    0    0  721
##
## Overall Statistics
##
##                Accuracy : 0.999
##                  95% CI : (0.9974, 0.9997)
##     No Information Rate : 0.285
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9987
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   1.0000   0.9985   1.0000   0.9986
## Specificity            1.0000   0.9994   1.0000   0.9994   1.0000
## Pos Pred Value         1.0000   0.9974   1.0000   0.9969   1.0000
## Neg Pred Value         0.9993   1.0000   0.9997   1.0000   0.9997
## Prevalence             0.2850   0.1930   0.1746   0.1634   0.1840
## Detection Rate         0.2845   0.1930   0.1744   0.1634   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9991   0.9997   0.9993   0.9997   0.9993
```

We note that there is an important trade-off here in terms of accuracy and speed compared to our first model. Accuracy for Model 2 increases to 100% on the validation set, although it is much more computationally demanding and subject to overfitting. We note that in this section we are using cross validation as the resampling method in the trainControl function, and changing to 5 the number that specifies the quantity of folds for k-fold cross validation.

For purely visual purposes, we now graph one of the trees from our model to get an idea of how the variables are interacting. We have chosen to plot tree k= 1. Note that the code for this graph, which is purely to illustrate our model, has been sourced from Shirin's playgRound and can be found at this webpage. Please find full reference in the *Sources* section.

```
tree_func(final_model = model2$finalModel, 2)
```



Checking accuracy on our test set

As a final step, and as part of the Data Science in R Specialization, we will check the accuracy of this model on a test set to predict the class or quality of movement of 20 different observations. We expected the out of sample error or generalization error to be higher than the in-sample error, particularly due to overfitting in random forest models. However, when predicting on the testing set we found that it maintained solid accuracy at 100%. Note that the labels for the test set were not provided, and as such the predictions were checked by the auto-grader of the course.

Conclusions

Following an exploratory analysis and model selection process, we have fitted a random forest model with strong accuracy metrics on our validation data set and test set which classifies the quality of a particular activity using 60 variables.

Sources

Glander, Shirin. Plotting trees from Random Forest models with ggraph. Shirin's playgRound. 2019. URL: https://shiring.github.io/machine_learning/2017/03/16/rf_plot_ggraph

Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer

Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6_6.