DATA SCIENCE SPECIALISM

MODULE 8 - PRACTICAL MACHINE LEARNING

Investigation into qualitative activity recognition information from a dumbbell lifting exercise and the predictive ability of a boosting machine learning algorithm

AUTHOR: FE YOUNG

DATE: 2015 Nov 21

ABSTRACT

Velloso (2013) investigated whether machine learning algorithms could accurately detect erroneous methods of lifting a dumbbell.

Following on from their research this analysis performs a predicton using the boosting machine learning algorithm rather than best fit Random Forest approach.

The predictive ability of Model One generated an overall accuracy was 0.96 and removing eight zero influence predictors reduced Model Two to an overall accuracy to 0.95. Model One was used to evaluate the prediction accuracy information located in the validation dataset. Model One correctly identified all 20 validation cases.

INTRODUCTION

The six male test subjects were of 20 - 28 years of age and inexperienced in dumbbell weight lifting exercises. The dumbbell weighed 1.25 kg.

Each subject performed a set of 10 repetitions of a unilateral dumbbell bicep curl in five different ways. Class A corresponded to the correct execution of the exercise while methods B through E corresponded to common dumbbell lifting mistakes namely (B) throwing the elbow to the front; (C) dumbbell lifted halfway; (D) dumbbell lowered halfway; (E) throwing hips to the front.

The question addressed in this report is can a machine learning model correctly identify 20 validation cases?

METHODOLOGY

Loading R packages:

~~~
```{R preprocessing, cache = TRUE}
 ##1 - loading libraries
 library(caret); library(ggplot2); library(data.table); library(plyr); library(dplyr); library(reshape2);
 library(ggplot2); library(knitr); library(rmarkdown); library(YaleToolkit)

 ##NOTE - for knitr/rmarkdown to work in RCONSOLE you are required to download the PANDOC package available online at: http://pandoc.org/installing.html
```
~~~

Loading required package: lattice
Loading required package: ggplot2
data.table 1.9.6 For help type ?data.table or https://github.com/Rdatatable/data.table/wiki
The fastest way to learn (by data.table authors): https://www.datacamp.com/courses/data-analysis-the-data-table-way

Attaching package: dplyr

The following objects are masked from package:plyr:

arrange, count, desc, failwith, id, mutate, rename, summarise, summarize

The following objects are masked from package:data.table:

between, last

The following objects are masked from package:stats:

filter, lag

The following objects are masked from package:base:

intersect, setdiff, setequal, union

Attaching package: reshape2

The following objects are masked from package:data.table:

dcast, melt

Loading required package: grid

What hardware/software combination did I use for this analysis?

~~~
```{r session info, cache = TRUE}

 ##2 - what hardware/software is this analysis using?

  sessionInfo()
```
~~~

R version 3.2.2 (2015-08-14)

```
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 8 x64 (build 9200)

locale:
[1] LC_COLLATE=English_United Kingdom.1252
[2] LC_CTYPE=English_United Kingdom.1252
[3] LC_MONETARY=English_United Kingdom.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.1252

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] YaleToolkit_4.2.2   rmarkdown_0.8    knitr_1.11
[4] reshape2_1.4.1      dplyr_0.4.3      plyr_1.8.3
[7] data.table_1.9.6    caret_6.0-58     ggplot2_1.0.1
[10] lattice_0.20-33

loaded via a namespace (and not attached):
[1] Rcpp_0.12.1       nloptr_1.0.4       iterators_1.0.8
[4] tools_3.2.2       digest_0.6.8       lme4_1.1-9
[7] nlme_3.1-122      gtable_0.1.2       mgcv_1.8-7
[10] Matrix_1.2-2     foreach_1.4.3      DBI_0.3.1
[13] parallel_3.2.2   SparseM_1.7        proto_0.3-10
[16] stringr_1.0.0    MatrixModels_0.4-1 stats4_3.2.2
[19] nnet_7.3-11      R6_2.1.1           minqa_1.2.4
[22] car_2.1-0        magrittr_1.5       htmltools_0.2.6
[25] scales_0.3.0     codetools_0.2-14   MASS_7.3-44
[28] splines_3.2.2    assertthat_0.1 pbkrtest_0.4-2
[31] colorspace_1.2-6 quantreg_5.19      stringi_0.5-5
[34] munsell_0.4.2    chron_2.3-47
```

Loading the training and validation datasets. Exploration of the training dataset.

~~~
```{loading datasets, cache = TRUE}
 ##3 - loading datasets

 #3.1 - trainingdataset
 if(!file.exists('pml-training.csv')) {
 download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", destfile = "pml-training.csv")
       }
 training <- read.table("pml-training.csv", sep = ",", header = T)

 ##3.2 - validation dataset
 if(!file.exists('pml-testing.csv')) {
 download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", destfile = "pml-testing.csv")
       }
 validation <- read.table("pml-testing.csv", sep =",", header = T)

   ##4 - exploring datasets
  dim(training); dim(validation); str(training, list.len = 160)
```
~~~

```
[1] 19622   160
[1] 20   160

'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 ...
$ 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 ...
$ amplitude_pitch_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_dumbbell   :   Factor w/ 3 levels "","#DIV/0!","0.00": 1 1 1 1 1 1 1 1 1 1 ...
$ total_accel_dumbbell   :   int 37 37 37 37 37 37 37 37 37 37 ...
$ var_accel_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_dumbbell   :   num NA NA NA NA NA NA NA NA NA NA ...
$ gyros_dumbbell_x   :   num 0 0 0 0 0 0 0 0 0 0 ...
$ gyros_dumbbell_y   :   num -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 ...
$ gyros_dumbbell_z   :   num 0 0 0 -0.02 0 0 0 0 0 0 ...
$ accel_dumbbell_x   :   int -234 -233 -232 -232 -233 -234 -232 -234 -232 -235 ...
$ accel_dumbbell_y   :   int 47 47 46 48 48 48 47 46 47 48 ...
$ accel_dumbbell_z   :   int -271 -269 -270 -269 -270 -269 -270 -272 -269 -270 ...
$ magnet_dumbbell_x   :   int -559 -555 -561 -552 -554 -558 -551 -555 -549 -558 ...
$ magnet_dumbbell_y   :   int 293 296 298 303 292 294 295 300 292 291 ...
$ magnet_dumbbell_z   :   num -65 -64 -63 -60 -68 -66 -70 -74 -65 -69 ...
$ roll_forearm   :   num 28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...
$ pitch_forearm   :   num -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63.8 -63.8 ...
$ yaw_forearm   :   num -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 ...
$ kurtosis_roll_forearm   :   Factor w/ 322 levels "","-0.0227","-0.0359",..: 1 1 1 1 1 1 1 1 1 1 ...
$ kurtosis_picth_forearm   :   Factor w/ 323 levels "","-0.0073","-0.0442",..: 1 1 1 1 1 1 1 1 1 1 ...
$ kurtosis_yaw_forearm   :   Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
$ skewness_roll_forearm   :   Factor w/ 323 levels "","-0.0004","-0.0013",..: 1 1 1 1 1 1 1 1 1 1 ...
$ skewness_pitch_forearm   :    Factor w/ 319 levels "","-0.0113","-0.0131",..: 1 1 1 1 1 1 1 1 1 1 ...
$ skewness_yaw_forearm   :    Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
$ max_roll_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ max_picth_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_forearm   :    Factor w/ 45 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...
$ min_roll_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_forearm   :    Factor w/ 45 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...
$ amplitude_roll_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_forearm  :    num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_forearm   :    Factor w/ 3 levels "","#DIV/0!","0.00": 1 1 1 1 1 1 1 1 1 1 ...
$ total_accel_forearm   :    int 36 36 36 36 36 36 36 36 36 36 ...
$ var_accel_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_forearm   :    num NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_forearm  :    num NA NA NA NA NA NA NA NA NA NA ...
$ gyros_forearm_x   :    num 0.03 0.02 0.03 0.02 0.02 0.02 0.02 0.02 0.03 0.02 ...
$ gyros_forearm_y   :    num 0 0 -0.02 -0.02 0 -0.02 0 -0.02 0 0 ...
$ gyros_forearm_z   :    num -0.02 -0.02 0 0 -0.02 -0.03 -0.02 0 -0.02 -0.02 ...
$ accel_forearm_x   :    int 192 192 196 189 189 193 195 193 193 190 ...
$ accel_forearm_y   :    int 203 203 204 206 206 203 205 205 204 205 ...
$ accel_forearm_z   :    int -215 -216 -213 -214 -214 -215 -215 -213 -214 -215 ...
$ magnet_forearm_x   :    int -17 -18 -18 -16 -17 -9 -18 -9 -16 -22 ...
$ magnet_forearm_y   :    num 654 661 658 658 655 660 659 660 653 656 ...
$ magnet_forearm_z   :    num 476 473 469 469 473 478 470 474 476 473 ...
$ classe   :    Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

The dataset consists of 19622 rows with 160 columns. Examination of the dataset concluded that 106 columns could be removed as they contained no valid information. These columns contained the words or abbreviations: kurtosis, mean, stddev, var, var_total, avg, skewness, max, min, new_window, num_window, amplitude. There are no columns with missing data or zero variance.

~~~
```{R removing columns & splitting dataset, cache = TRUE}

 ##5 - removing unwanted columns and dealing with missing data

 ##5.1 - including the words: kurtosis, mean, stddev, var, var_total, avg, skewness, max, min, new_window, num_window, amplitude and the time variables
 trainingdata <- training[c(8:10, 37:48, 60:68, 84:86, 113:124, 151:160)]
 validationdataset <- validation[c(8:10, 37:48, 60:68, 84:86, 113:124, 151:160)]

 ##5.2 - where zeroVar = 0 AND nzv = TRUE remove columns? NOTHING TO DEAL WITH
 removezero1 <- nearZeroVar(trainingdata, saveMetrics = T)
  removezero1
```
~~~

```
              freqRatio    percentUnique    zeroVar    nzv
roll_belt      1.101904      6.7781062      FALSE      FALSE
pitch_belt      1.036082      9.3772296      FALSE      FALSE
yaw_belt      1.058480      9.9734991      FALSE      FALSE
gyros_belt_x      1.058651      0.7134849      FALSE      FALSE
gyros_belt_y      1.144000      0.3516461      FALSE      FALSE
gyros_belt_z      1.066214      0.8612782      FALSE      FALSE
accel_belt_x      1.055412      0.8357966      FALSE      FALSE
accel_belt_y      1.113725      0.7287738      FALSE      FALSE
accel_belt_z      1.078767      1.5237998      FALSE      FALSE
magnet_belt_x      1.090141      1.6664968      FALSE      FALSE
magnet_belt_y      1.099688      1.5187035      FALSE      FALSE
magnet_belt_z      1.006369      2.3290184      FALSE      FALSE
roll_arm      52.338462      13.5256345      FALSE      FALSE
pitch_arm      87.256410      15.7323412      FALSE      FALSE
yaw_arm      33.029126      14.6570176      FALSE      FALSE
gyros_arm_x      1.015504      3.2769341      FALSE      FALSE
gyros_arm_y      1.454369      1.9162165      FALSE      FALSE
gyros_arm_z      1.110687      1.2638875      FALSE      FALSE
accel_arm_x      1.017341      3.9598410      FALSE      FALSE
accel_arm_y      1.140187      2.7367241      FALSE      FALSE
accel_arm_z      1.128000      4.0362858      FALSE      FALSE
magnet_arm_x      1.000000      6.8239731      FALSE      FALSE
magnet_arm_y      1.056818      4.4439914      FALSE      FALSE
magnet_arm_z      1.036364      6.4468454      FALSE      FALSE
roll_dumbbell      1.022388      84.2065029      FALSE      FALSE
pitch_dumbbell      2.277372      81.7449801      FALSE      FALSE
yaw_dumbbell      1.132231      83.4828254      FALSE      FALSE
gyros_dumbbell_x      1.003268      1.2282132      FALSE      FALSE
gyros_dumbbell_y      1.264957      1.4167771      FALSE      FALSE
gyros_dumbbell_z      1.060100      1.0498420      FALSE      FALSE
accel_dumbbell_x      1.018018      2.1659362      FALSE      FALSE
accel_dumbbell_y      1.053061      2.3748853      FALSE      FALSE
accel_dumbbell_z      1.133333      2.0894914      FALSE      FALSE
magnet_dumbbell_x      1.098266      5.7486495      FALSE      FALSE
magnet_dumbbell_y      1.197740      4.3012945      FALSE      FALSE
magnet_dumbbell_z      1.020833      3.4451126      FALSE      FALSE
roll_forearm      11.589286      11.0895933      FALSE      FALSE
pitch_forearm      65.983051      14.8557741      FALSE      FALSE
yaw_forearm      15.322835      10.1467740      FALSE      FALSE
gyros_forearm_x      1.059273      1.5187035      FALSE      FALSE
gyros_forearm_y      1.036554      3.7763735      FALSE      FALSE
gyros_forearm_z      1.122917      1.5645704      FALSE      FALSE
accel_forearm_x      1.126437      4.0464784      FALSE      FALSE
```

```
accel_forearm_y     1.059406     5.1116094     FALSE     FALSE
accel_forearm_z     1.006250     2.9558659     FALSE     FALSE
magnet_forearm_x    1.012346     7.7667924     FALSE     FALSE
magnet_forearm_y     1.246914     9.5403119     FALSE     FALSE
magnet_forearm_z     1.000000     8.5771073     FALSE     FALSE
classe     1.469581     0.0254816     FALSE     FALSE
```

~~~
```{r whatis, cache = TRUE}

 ##5.3 - is there any missing data to impute?  NOTHING TO DEAL WITH
 whatis(trainingdata)
```
~~~

```
variable.name     type     missing     distinct.values     precision     min     max
  1 roll_belt     numeric     0     1330     1e-02     -28.9     162
  2 pitch_belt     numeric     0     1840     1e-02     -55.8     60.3
  3 yaw_belt     numeric     0     1957     1e-02     -180     179
  4 gyros_belt_x     numeric     0     140     1e-02     -1.04     2.22
  5 gyros_belt_y     numeric     0     69     1e-02     -0.64     0.64
  6 gyros_belt_z     numeric     0     169     1e-02     -1.46     1.62
  7 accel_belt_x     numeric     0     164     1e+00     -120     85
  8 accel_belt_y     numeric     0     143     1e+00     -69     164
  9 accel_belt_z     numeric     0     299     1e+00     -275     105
10 magnet_belt_x     numeric     0     327     1e+00     -52     485
11 magnet_belt_y     numeric     0     298     1e+00     354     673
12 magnet_belt_z     numeric     0     457     1e+00     -623     293
13 roll_arm     numeric     0     2654     1e-02     -180     180
14 pitch_arm     numeric     0     3087     1e-02     -88.8     88.5
15 yaw_arm     numeric     0     2876     1e-02     -180     180
16 gyros_arm_x     numeric     0     643     1e-02     -6.37     4.87
17 gyros_arm_y     numeric     0     376     1e-02     -3.44     2.84
18 gyros_arm_z     numeric     0     248     1e-02     -2.33     3.02
19 accel_arm_x     numeric     0     777     1e+00     -404     437
20 accel_arm_y     numeric     0     537     1e+00     -318     308
21 accel_arm_z     numeric     0     792     1e+00     -636     292
22 magnet_arm_x     numeric     0     1339     1e+00     -584     782
23 magnet_arm_y     numeric     0     872     1e+00     -392     583
24 magnet_arm_z     numeric     0     1265     1e+00     -597     694
25 roll_dumbbell     numeric     0     16523     1e-09     -153.7137292     153.5455708
26 pitch_dumbbell     numeric     0     16040     1e-09     -149.5936479     149.4024436
27 yaw_dumbbell     numeric     0     16381     1e-09     -150.8711542     154.9522941
28 gyros_dumbbell_x     numeric     0     241     1e-02     -204     2.22
29 gyros_dumbbell_y     numeric     0     278     1e-02     -2.1     52
30 gyros_dumbbell_z     numeric     0     206     1e-02     -2.38     317
31 accel_dumbbell_x     numeric     0     425     1e+00     -419     235
32 accel_dumbbell_y     numeric     0     466     1e+00     -189     315
33 accel_dumbbell_z     numeric     0     410     1e+00     -334     318
34 magnet_dumbbell_x     numeric     0     1128     1e+00     -643     592
35 magnet_dumbbell_y     numeric     0     844     1e+00     -3600     633
36 magnet_dumbbell_z     numeric     0     676     1e-01     -262     452
37 roll_forearm     numeric     0     2176     1e-02     -180     180
38 pitch_forearm     numeric     0     2915     1e-02     -72.5     89.8
39 yaw_forearm     numeric     0     1991     1e-02     -180     180
40 gyros_forearm_x     numeric     0     298     1e-02     -22     3.97
41 gyros_forearm_y     numeric     0     741     1e-02     -7.02     311
42 gyros_forearm_z     numeric     0     307     1e-02     -8.09     231
43 accel_forearm_x     numeric     0     794     1e+00     -498     477
44 accel_forearm_y     numeric     0     1003     1e+00     -632     923
45 accel_forearm_z     numeric     0     580     1e+00     -446     291
46 magnet_forearm_x     numeric     0     1524     1e+00     -1280     672
47 magnet_forearm_y     numeric     0     1872     1e-03     -896     1480
48 magnet_forearm_z     numeric     0     1683     1e-04     -973     1090
49 classe     pure factor     0     5     NA     A     E
```

It was decided to split the dataset into two randomly selected pieces using the createDataPartition command because of the large dataset size. The two pieces: 60% (11767 rows) for model training and 40% (7846 rows) for model testing were chosen by trial and error. The training model provided evidence that model accuracy increased as the size of training dataset was increased, but was constrained by computing power.

A validation dataset has been supplied containing 20 rows in order to fulfil the project requirement for this Data Science Specialism module. One point per row will be awarded for each correctly predicted answer by the generated model.

~~~
```{R splitting, cache = TRUE}

 ##6 - splitting the dataset 70:30 training:testing
 split1 <- createDataPartition(y = trainingdata$classe, p = 0.6, list = FALSE)
 trainingdataset <- trainingdata[split1, ]
 testingdataset <- trainingdata[-split1, ]
 dim(trainingdataset); dim(testingdataset)
```
~~~

```
[1] 11776 49
[1] 7846 49
```

The training model instructions required that the classe (A - E) variable was to be predicted by the model. To train the model the classe variable had to be removed so not to predict itself.

~~~
```{R premodelling, cache = TRUE}
 ##7 - classe ~ user_name + all variables INCLUDING PREPROCESSING
 namestraining <- names(trainingdataset[c(-49)])
 form <- as.formula(paste("classe~", paste(namestraining, collapse = "+"), sep = ""))
```
~~~

The choice of training model was related to its computational RAM (random access memory) expense, the time available to complete the module project analysis submission and the defined

accuracy of the model compared to others as described by Jeff Leek in the video lecture on Boosting (see References). This led to the selection of the boosting model - command "gbm".

Preprocessing of the training dataset was performed at the same time as model training and it centred and scaled the all variables. If any other preprocessing commands were added to the model the computer produced a BSoD (blue screen of death).

The boosting model on the training dataset was run several times and each time there was a slightly different accuracy output so for reproducibility a seed was set, number 1258 was used.

~~~
```{R training model, cache = TRUE}
 ##8 - Model 1
 set.seed(1258)
 model1 <- train(form, data = trainingdataset, preProcess = c("scale", "center"), method = "gbm", verbose = F)
```
~~~

On completion of the model it was noted that eight predictors had no model influence and were removed from the training dataset. The predictors are accel_belt_x, accel_belt_y, pitch_arm, gyros_arm_z, accel_arm_y, yaw_dumbbell, yaw_forearm and gyros_forearm_y.

~~~
```{r removal}
 ##9 - removing additional variables
 trainingdataset2 <- trainingdataset[c(1:6, 9:13, 15:17, 19, 21:26, 28:38, 40, 42:49)]
 dim(trainingdataset2)
```
~~~

```
[1] 11776   41
```

The model was run again without these eight predictors for the purpose of cross validation with 50% of the original training dataset rows randomly chosen. Would the model accuracy improved without these eight variables?

~~~
```{r splitting 2}
 ##10 - splitting the training dataset into 2 pieces 50:50
 split2 <- createDataPartition(y = trainingdataset2$classe, p = 0.5, list = FALSE)
 trainingdataset3 <- trainingdataset2[split2, ]

 ##11 - removing the classe variable
 namestraining <- names(trainingdataset3[c(-41)])
 form2 <- as.formula(paste("classe~", paste(namestraining, collapse = "+"), sep = ""))

 ##12 - MODEL 2
 model2 <- train(form2, data = trainingdataset3, preProcess = c("scale", "center"), method = "gbm", verbose = F)
```
~~~

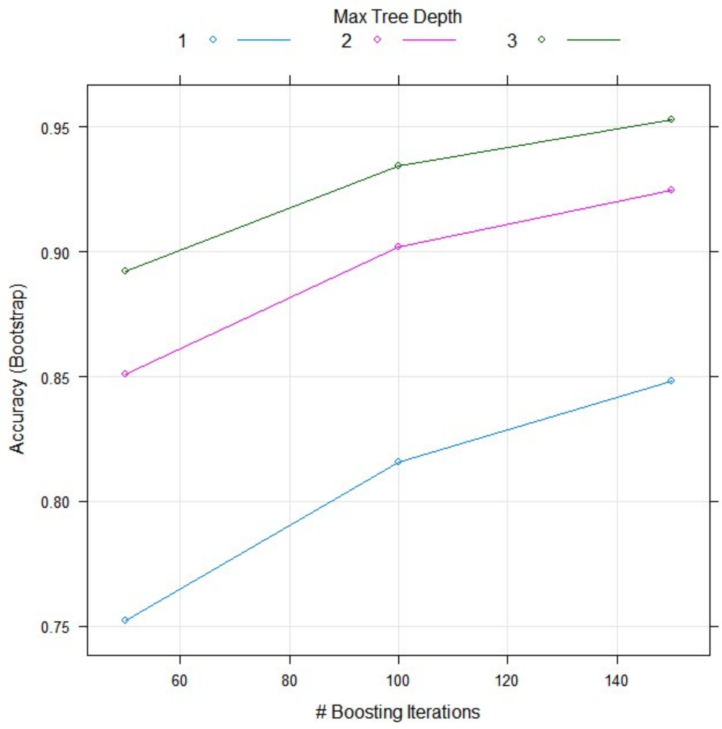
On satisfactory training of the final model it was used to predict the validation dataset.

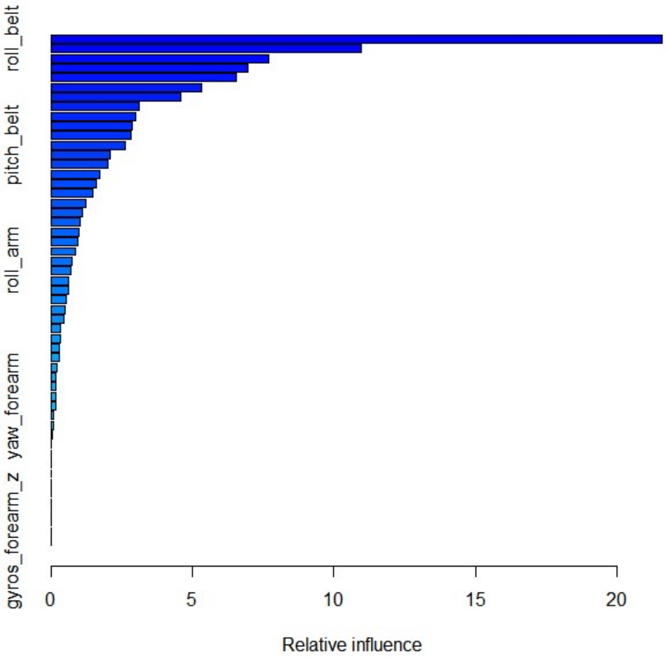
RESULTS

The results from the Model One and its predictive accuracy on the testing dataset:

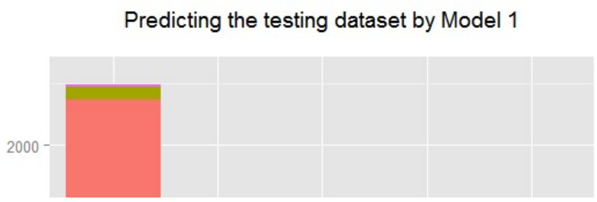
~~~
```{r boosting model 1, cache = TRUE}

 ##13 - Model 1 results
 print(model1$finalModel)
 plot(model1)
 summary(model1)
 prediction1 <- predict(model1, testingdataset)
 qplot(prediction1, colour = classe, fill = classe, data = testingdataset, main = "Predicting the testing dataset by Model 1\n", ylab = "Count\n")
 confusionMatrix(testingdataset$classe, predict(model1, testingdataset))
```
~~~

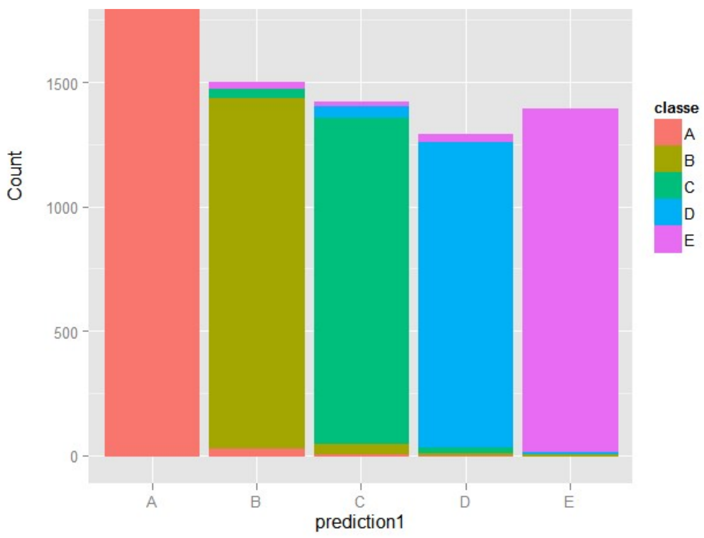
```
A gradient boosted model with multinomial loss function.
150 iterations were performed.
There were 48 predictors of which 39 had non-zero influence.
```





```
summary(model1)

       var          rel.inf
roll_belt   21.83897809
pitch_forearm   11.36695394
yaw_belt     8.97865050
magnet_dumbbell_z     6.57919727
magnet_dumbbell_y     5.74142037
roll_forearm     5.14858798
magnet_belt_z     3.99357158
pitch_belt     3.37862292
accel_forearm_x     3.31485298
gyros_belt_z     2.93819732
accel_dumbbell_y     2.81185277
gyros_dumbbell_y     2.21240996
roll_dumbbell     2.18541918
magnet_forearm_z     2.08175431
accel_forearm_z     1.94206210
magnet_dumbbell_x     1.76878150
yaw_arm     1.61450841
magnet_belt_y     1.52090293
accel_dumbbell_x     1.46213992
magnet_forearm_x     1.00908163
magnet_arm_z     0.99856300
magnet_arm_x     0.85182530
roll_arm     0.78633180
magnet_belt_x     0.71144488
magnet_arm_y     0.69694596
gyros_arm_y     0.62057215
accel_dumbbell_z     0.53867952
gyros_dumbbell_x     0.50729571
gyros_belt_y     0.46878891
accel_belt_z     0.40128026
accel_arm_x     0.32579427
gyros_dumbbell_z     0.28890462
accel_arm_y     0.24173411
pitch_dumbbell     0.19890487
accel_forearm_y     0.14343623
accel_arm_z     0.12345823
magnet_forearm_y     0.09371805
gyros_forearm_z     0.06176832
gyros_belt_x     0.05260815
accel_belt_x     0.00000000
accel_belt_y     0.00000000
pitch_arm     0.00000000
gyros_arm_x     0.00000000
gyros_arm_z     0.00000000
yaw_dumbbell     0.00000000
yaw_forearm     0.00000000
gyros_forearm_x     0.00000000
gyros_forearm_y     0.00000000
```





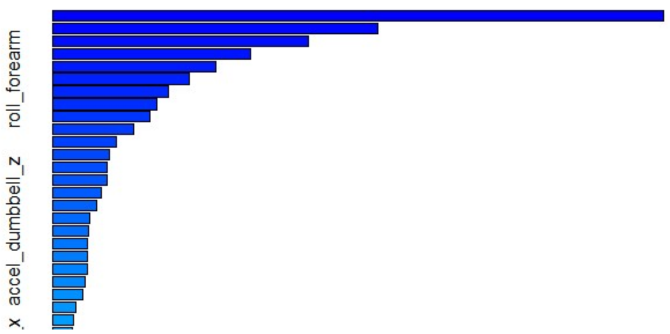
```
Confusion Matrix and Statistics

Reference
Prediction A     B     C     D     E
    A    2204    19     6     3     0
    B      55  1428    31     2     2
    C       0    44  1300    24     0
    D       1     4    42  1228    11
    E       2    27    18    21  1374

Overall Statistics

Accuracy : 0.9602
95% CI : (0.9557, 0.9645)
No Information Rate : 0.2883
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9497
Mcnemar's Test P-Value : 1.922e-12

Statistics by Class:

                     Class: A  Class: B  Class: C  Class: D  Class: E
Sensitivity            0.9744   0.9382   0.9306   0.9609   0.9906
Specificity            0.9950   0.9858   0.9895   0.9912   0.9895
Pos Pred Value         0.9875   0.9407   0.9503   0.9549   0.9528
Neg Pred Value         0.9897   0.9851   0.9850   0.9924   0.9980
Prevalence             0.2883   0.1940   0.1781   0.1629   0.1768
Detection Rate         0.2809   0.1820   0.1657   0.1565   0.1751
Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
Balanced Accuracy      0.9847   0.9620   0.9600   0.9760   0.9900
```

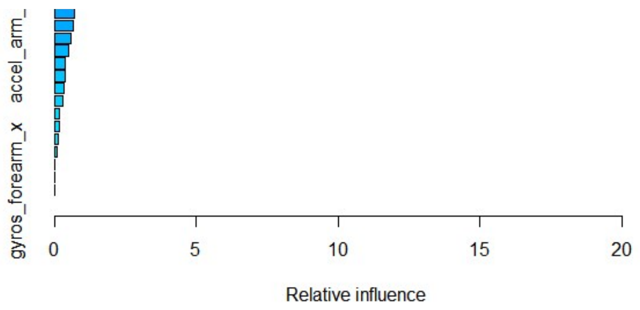
The resultant statistical output was examined. The overall accuracy of the model was 0.96. The positive predictive value (PPV) was over 0.95 for classes A, C to E and class B above 0.94 whereas the negative predictive value (NPV) was above 0.98 for all classes.

The results for Model Two (below) demonstrated that removing the eight predictors reduced the accuracy of the boosting model from 0.96 to 0.95. The PPV was reduced for four of the classes to 0.94 but class B reduced to 0.91. For NPV all class values were above 0.98. As the accuracy of the model dropped without these eight predictors the first model was chosen to make predictions for the validation dataset.

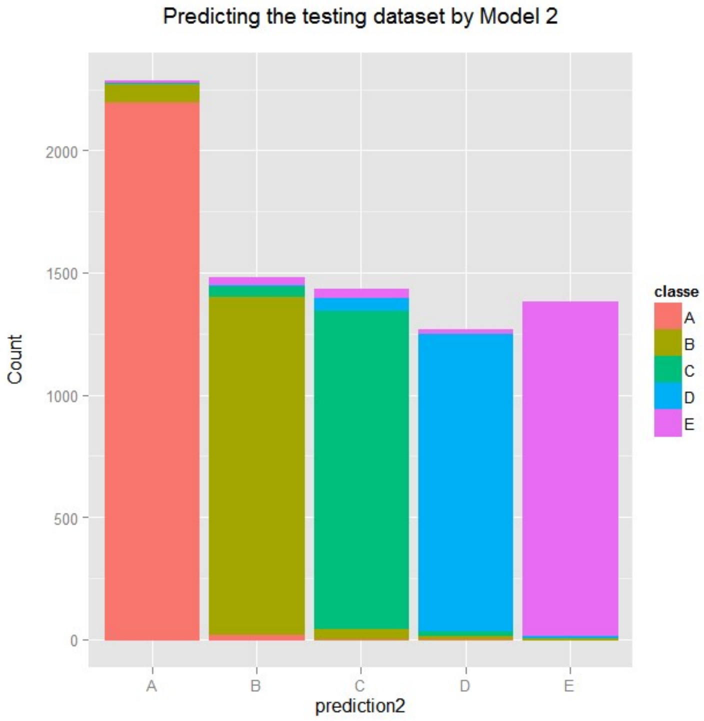
~~~
```{r boosting model 2, cache = TRUE}
 ##14 - Model 2 results
 print(model2$finalModel)
 summary(model2)
 prediction2 <- predict(model2, testingdataset)
 qplot(prediction2, colour = classe, fill = classe, data = testingdataset, main = "Predicting the testing dataset by Model 2\n", ylab = "Count\n")
 confusionMatrix(testingdataset$classe, predict(model2, testingdataset))
```
~~~

```
A gradient boosted model with multinomial loss function.
150 iterations were performed.
There were 40 predictors of which 38 had non-zero influence.
```





```
     var      rel.inf roll_belt      20.41139035
pitch_forearm      11.13230517
yaw_belt      8.57040521
magnet_dumbbell_z      7.38331461
magnet_dumbbell_y      5.62204502
roll_forearm      4.81180232
pitch_belt      4.22906941
magnet_belt_z      4.10488396
gyros_belt_z      3.48826302
accel_forearm_x      2.93847954
roll_dumbbell      2.76104853
accel_dumbbell_y      2.63277996
gyros_dumbbell_y      2.10455965
accel_dumbbell_x      1.85007941
accel_forearm_z      1.51293178
magnet_forearm_z      1.47116829
yaw_arm      1.35515145
magnet_dumbbell_x      1.14543364
roll_arm      1.13477891
magnet_belt_y      1.09946230
magnet_arm_x      1.07115271
magnet_arm_z      1.04113224
gyros_dumbbell_x      1.02664711
magnet_forearm_x      0.94197974
accel_dumbbell_z      0.92651947
accel_belt_z      0.76461104
magnet_arm_y      0.71078786
gyros_arm_y      0.67536804
magnet_belt_x      0.67337698
magnet_forearm_y      0.59799559
accel_arm_x      0.49878590
gyros_belt_y      0.39914959
accel_forearm_y      0.29285900
gyros_dumbbell_z      0.24848172
pitch_dumbbell      0.19907974
gyros_forearm_z      0.07547864
accel_arm_z      0.05144473
gyros_arm_x      0.04579738
gyros_belt_x      0.00000000
gyros_forearm_x      0.00000000
```



```
Confusion Matrix and Statistics

          Reference
```

```
Prediction A     B     C     D     E
     A   2202    17    10     3     0
     B     79  1385    48     3     3
     C      0    49  1297    21     1
     D      1     3    56  1213    13
     E      5    26    32    20  1359

Overall Statistics

Accuracy : 0.9503
95% CI : (0.9453, 0.955)
No Information Rate : 0.2915
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9371
Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: A   Class: B   Class: C   Class: D   Class: E
Sensitivity            0.9628     0.9358     0.8988     0.9627     0.9876
Specificity            0.9946     0.9791     0.9889     0.9889     0.9872
Pos Pred Value         0.9866     0.9124     0.9481     0.9432     0.9424
Neg Pred Value         0.9849     0.9850     0.9775     0.9928     0.9973
Prevalence             0.2915     0.1886     0.1839     0.1606     0.1754
Detection Rate         0.2807     0.1765     0.1653     0.1546     0.1732
Detection Prevalence   0.2845     0.1935     0.1744     0.1639     0.1838
Balanced Accuracy      0.9787     0.9575     0.9439     0.9758     0.9874
```

The prediction results of the 20 validation cases using Model One:

~~~
```{r validation predictions, cache = TRUE}

 ##15 - Predictions with Model 1
 predict(model1, validationdataset)
```
~~~

```
[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
```

Of the 20 cases all 20 have been correctly predicted.

REFERENCES

Velloso E, Bulling A, Gellersen H, Ugulino W, Fuks H (2013) Qualitative Activity Recognition of Weight Lifting Exercises, Proceedings of the 4th International Conference in Cooperation with SIGCHI (Augmented Human 2013), Stuttgart, Germany

Guillaume Bourgault & Chris W (2015) Distribution of each variable for each test subject and each class (A - E) online at https://class.coursera.org/predmachlearn-034/forum/thread?thread_id=20

Leek J (2015) Boosting video lecture available online at https://class.coursera.org/predmachlearn-034/lecture/49

APPENDICES

APPENDIX 1: Codebook

Abbreviations for parts of the column names:-

gyros <- gyroscope
x <- x axis
y <- y axis
z <- z axis
accel <- accelerometer
magnet <- magnetometer

Meaning of classe headings:-

(A) correct execution of the exercise
(B) throwing the elbow to the front
(C) dumbbell lifted halfway
(D) dumbbell lowered halfway
(E) throwing hips to the front

Position of sensors:-

belt around the waist
arm around the upper arm
forearm around the lower arm
dumbbell on the end of the dumbbell

APPENDIX 2

The information for this project comes from these sources:

http://groupware.les.inf.puc-rio.br/har

The links to the datasets are:

Entire dataset:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

Validation dataset:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

APPENDIX 3

With only 2 GB of hard disk space the html document could not be constructed within the R package. Therefore I had to write the whole html file myself using Notepad++. So if the result tables look a little strange and columns not correctly aligned this is the reason why.

~~~
```{r html}
render("project.Rmd", html_document(), quiet = T)
```
~~~