Machine Learning - Weight Exercise

Tim Kerins

September 4, 2018

Executive Summary

The purpose of this weightlifting exercise is to determine if we can accurately distinguish between 5 different classes of correctly and incorrectly performing a unilateral biceps curl by using accelerometers on the body and the dumbbell. Six subjects were used to perform ten repetitions each.

- Class A: Correctly
- Class B: Throwing Elbows in front
- Class C: Lifting only half way
- Class D: Lowering only half way
- Class E: Throwing Hips in front

The resulting data set was cleansed and divided into training and validation sets and applied to a random forest model. The accuracies were determined and variable importance's examined. Finally, a 20 observation test data set was then applied to the model and the resulting predictions were entered into machine learning quiz #4 to compare the results.

Summary of Conclusions

After data cleansing there were 53 variables including the "classe" variable that were put into the random forest model. The "classe" variable was the output variable with the other 52 being the predictor variables. Running the training set produced an error rate of 0.43% which implies an accuracy of 99.57%. Running the validating set to cross validate the model produced an accuracy of 99.47%. This out of sample accuracy was only slightly less than the training accuracy, which is to be expected. An analysis of variable importance showed the top 8 contributors to be magnet_dumbbell_x, roll_forearm, magnet_dumbbell_y, pitch_belt, magnet_dumbbell_z, pitch_forearm, yaw_belt, and roll_belt.

The test data was then loaded into the model. The resulting predictions were input into Quiz #4 and matched the answers expected.

Exploratory Data Analysis and Cleansing

Load Libraries

```
library(caret)
library(randomForest)
library(dplyr)
library(tidyr)
```

Download Data

```
url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
destfile <- "train.csv"
download.file(url, destfile)
train <- read.csv(destfile,na.strings=c("NA","#DIV/0!","","  "))
url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
```

```
destfile <- "test.csv"
download.file(url, destfile)
test <- read.csv(destfile,na.strings=c("NA","#DIV/0!","","  "))
```

Cleanse Data

Remove columns with NA values

```
train1 <- train %>% select_if(~ !any(is.na(.)))
test1 <- test %>% select_if(~ !any(is.na(.)))
```

Investigate the dimensions and data structure of the cleaned train and test data sets and then remove columns not needed for the model. (See the appendix for the details of the dimensions and structures and rationale for removing the columns below).

```
train2 <- select(train1,-X,-user_name,-raw_timestamp_part_1, -raw_timestamp_part_2,
                 -cvtd_timestamp,-new_window, -num_window)
test2 <- select(test1,-problem_id,-X,-user_name,-raw_timestamp_part_1,
                -raw_timestamp_part_2,
                -cvtd_timestamp,-new_window, -num_window)
```

Determine the dimensions of the data sets to be modeled

```
dim(train2);dim(test2)
```

```
## [1] 19622    53
```

```
## [1] 20 52
```

52 predictor variables will be modeled. The output of the model will be the "classe"" variable

Build Model

Split data into training and validating sets

```
set.seed(20)
inTrain = createDataPartition(y = train2$classe, p = .75)[[1]]
training = train2[inTrain,]
validating = train2[-inTrain,]
```

Build and train a random forest model

```
set.seed(20)
rfMod <- randomForest(classe~.,data = training)
rfMod
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.43%
```

```
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4184    1    0    0    0 0.0002389486
## B   15 2828    5    0    0 0.0070224719
## C    0   15 2552    0    0 0.0058433970
## D    0    0   18 2393    1 0.0078772803
## E    0    0    1    7 2698 0.0029563932
```

Running the training set produced an OOB error rate of 0.43% which implies an accuracy of 99.57%.

Cross Validate the model with the validating data set (See appendix for plot of Variable Importance)

```
#Test the model on the validation dataset.
pred <- predict(rfMod,newdata=validating)
confusionMatrix(pred,validating$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    3    0    0    0
##          B    2  945    4    0    0
##          C    0    1  848   11    0
##          D    0    0    3  792    1
##          E    0    0    0    1  900
##
## Overall Statistics
##
##                Accuracy : 0.9947
##                  95% CI : (0.9922, 0.9965)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9933
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9986   0.9958   0.9918   0.9851   0.9989
## Specificity            0.9991   0.9985   0.9970   0.9990   0.9998
## Pos Pred Value         0.9979   0.9937   0.9860   0.9950   0.9989
## Neg Pred Value         0.9994   0.9990   0.9983   0.9971   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2841   0.1927   0.1729   0.1615   0.1835
## Detection Prevalence   0.2847   0.1939   0.1754   0.1623   0.1837
## Balanced Accuracy      0.9989   0.9971   0.9944   0.9920   0.9993
```

Running the validating set to cross validate the model produced an accuracy of 99.47%. This out of sample accuracy was only slightly less than the training accuracy, which is to be expected.

Determine Variable Importance. (see appendix for the Variable Importance Plot)

From the chart in the appendix we can see that the top 8 contributors to importance are magnet_dumbbell_x, roll_forearm, magnet_dumbbell_y, pitch_belt, magnet_dumbbell_z, pitch_forearm, yaw_belt, and roll_belt.

Run the model using the test dataset

```
# Predict Classes using test data
test_pred <- predict(rfMod, newdata = test2)
test_pred
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

These resulting predictions were input into Quiz #4 and were validated to be correct.

Appendix

Evaluate the dimensions and structure of the train dataset after NAs are removed

```
str(train1)
```

```
## 'data.frame':    19622 obs. of  60 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 1323
##  $ 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 ...
##  $ 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 ...
##  $ 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 ...
##  $ 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 ...
##  $ total_accel_dumbbell: int  37 37 37 37 37 37 37 37 37 37 ...
##  $ 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 ...
##  $ total_accel_forearm : int  36 36 36 36 36 36 36 36 36 36 ...
##  $ 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 ...
```

There are 19,622 observations of 60 variables. "class"" is the output variable. The first 7 variables are indexes or names or summary variables that are not needed for the model so these were removed as shown in the data cleansing section of this document.

Evaluate the dimensions and structure of the test data set after NAs are removed.

```
dim(test1)
```

```
## [1] 20 60
```

There are 20 observations of 60 variables. An analysis of the structure (not shown here for brevity) shows the variables are identical to the train set except that "classe"" is not present and "problem_id"" is an extra variable that is not in the train data set. The first seven columns are identical to the train set and we'll remove these as we did for the training set and also remove "problem_id"

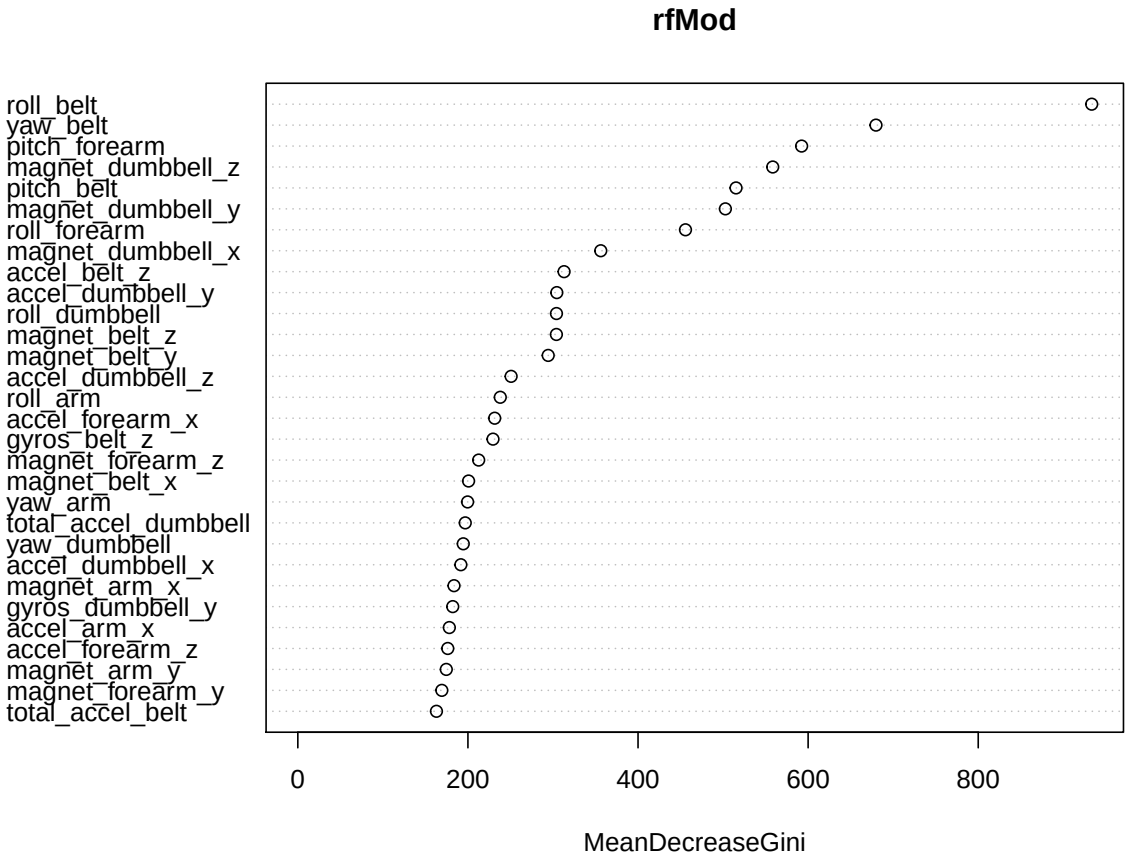
The resulting dimensions that will be input to the model are as follows. Note that train2 includes the the output variable "classe"):

```
## [1] 19622    53
```

```
## [1] 20 52
```

Plot Variable Importance

```
varImpPlot(rfMod)
```



Citations

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.