Peer Assessment: Prediction of Dumbbell Lifting with Data from Accelerometers

Date: August 21, 2014
Author: Michael W. Lee

Excutive Summary

This is the Peer Assessment assignment for the Practical Machine Learning Course in Datascience course series. The original training dataset was partitioned into `training` (60%) and `testing` (40%) datasets. After unneeded variables were removed, 53 variables were used to build a predictive model using boosting algorithm with `gbm`. The resulting model is highly accurate with in-sample accuracy of 98% and out of sample accuracy of 96%. The predictions were made on 20 cases in `pml-testing` dataset as part of the assignment, and all predictions were correct.

Introduction

There are inexpensive devices such as *Jawbone Up, Nike FuelBand,* and *Fitbit* available commercially which collect a large amount of data about personal activity and quantify how *much* of a particular activity. However, these devices rarely quantify *how well they do it.* In this project, data were collected from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. The goal of the project is to predict the manner in which they did the exercise.

Method

The dataset for this project was downloaded from course assignment website. The data came from the following publication:

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.

```
setwd("C:/Users/Michael/Desktop/machine")
pml <- read.csv("pml-training.csv", header=TRUE)
```

The six participants were healthy male subjects aged between 20-28 years, with litttle weight lifting experience. All participants could easily simulate the mistakes in a safe and controlled manner by using a relatively light dumbbell (1.25kg). They 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).

The accelerometers were attached to right arm, right forearm, belt, and dumbbell. Measurements of acceleration, gyros and magnet(compass) were taken in X, Y, Z coordinates. Roll, pitch and yaw were calculated from the above data. The `pml` dataset was partitioned into `pmltrain` (60%) and `pmltest` (40%). `pmltrain` was used for training and tuning, and `pmltest` was used only for cross validation purpose. The `caret` package was used to perform the training.

```r
library(caret);library(ggplot2)
```

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

```r
set.seed(35353)
inTrain <- createDataPartition(pml$classe, p = 0.6, list=FALSE)
pmltrain <- pml[inTrain,]
pmltest <- pml[-inTrain,]
```

The dataset contained 160 variables. Of these 160 variables, many variables had a large numbers of `NA`'s, and therefore, these variables were not used. User name and time stamps were irrelevant in generalizability of the prediction, so they were not used. Remaining 53 variables had complete data and were used for the project. The summary of these 53 variables is listed below.

```r
training <- pmltrain[, c(8,9,10,11,37:49,60:68,84:86,102,113:124,140,151:160)]
testing <- pmltest[, c(8,9,10,11,37:49,60:68,84:86,102,113:124,140,151:160)]
summary(training)
```

```
##    roll_belt       pitch_belt        yaw_belt      total_accel_belt
##  Min.   :-28.9   Min.   :-55.80   Min.   :-180.0   Min.   : 0.0    
##  1st Qu.:  1.1   1st Qu.:  1.72   1st Qu.: -88.3   1st Qu.: 3.0    
##  Median :112.0   Median :  5.26   Median : -14.6   Median :17.0    
##  Mean   : 63.8   Mean   :  0.18   Mean   : -11.5   Mean   :11.2    
##  3rd Qu.:123.0   3rd Qu.: 14.50   3rd Qu.:  12.2   3rd Qu.:18.0    
##  Max.   :162.0   Max.   : 60.30   Max.   : 179.0   Max.   :28.0    
##   gyros_belt_x      gyros_belt_y      gyros_belt_z     accel_belt_x 
##  Min.   :-1.0000   Min.   :-0.6400   Min.   :-1.460   Min.   :-83.0  
##  1st Qu.:-0.0300   1st Qu.: 0.0000   1st Qu.:-0.180   1st Qu.:-21.0  
##  Median : 0.0300   Median : 0.0200   Median :-0.100   Median :-14.0  
##  Mean   :-0.0025   Mean   : 0.0399   Mean   :-0.128   Mean   : -5.4  
##  3rd Qu.: 0.1100   3rd Qu.: 0.1100   3rd Qu.: 0.000   3rd Qu.: -5.0  
##  Max.   : 2.2200   Max.   : 0.6400   Max.   : 1.610   Max.   : 85.0  
##   accel_belt_y     accel_belt_z     magnet_belt_x    magnet_belt_y
##  Min.   :-69.0   Min.   :-269.0   Min.   :-52.0   Min.   :354  
##  1st Qu.:  3.0   1st Qu.:-162.0   1st Qu.:  9.0   1st Qu.:581  
##  Median : 31.0   Median :-150.0   Median : 35.0   Median :601  
##  Mean   : 29.8   Mean   : -71.5   Mean   : 55.8   Mean   :594  
##  3rd Qu.: 61.0   3rd Qu.:  28.0   3rd Qu.: 60.0   3rd Qu.:610  
##  Max.   :109.0   Max.   : 105.0   Max.   :485.0   Max.   :673  
##  magnet_belt_z     roll_arm        pitch_arm         yaw_arm      
##  Min.   :-623   Min.   :-180.0   Min.   :-88.80   Min.   :-180.00  
##  1st Qu.:-375   1st Qu.: -31.6   1st Qu.:-26.10   1st Qu.: -43.30  
##  Median :-319   Median :   0.0   Median :  0.00   Median :   0.00  
##  Mean   :-345   Mean   :  17.6   Mean   : -4.84   Mean   :  -1.02  
##  3rd Qu.:-306   3rd Qu.:  77.0   3rd Qu.: 11.10   3rd Qu.:  45.12  
##  Max.   : 293   Max.   : 179.0   Max.   : 88.50   Max.   : 180.00  
##  total_accel_arm  gyros_arm_x      gyros_arm_y      gyros_arm_z   
##  Min.   : 1.0    Min.   :-6.370   Min.   :-3.440   Min.   :-2.28  
##  1st Qu.:17.0    1st Qu.:-1.350   1st Qu.:-0.790   1st Qu.:-0.07  
##  Median :27.0    Median : 0.060   Median :-0.240   Median : 0.23  
##  Mean   :25.5    Mean   : 0.038   Mean   :-0.252   Mean   : 0.27  
```

```
##  3rd Qu.:33.0    3rd Qu.: 1.570   3rd Qu.: 0.140   3rd Qu.: 0.72
##  Max.   :65.0    Max.   : 4.870   Max.   : 2.810   Max.   : 2.95
##   accel_arm_x      accel_arm_y      accel_arm_z      magnet_arm_x
##  Min.   :-404.0   Min.   :-318.0   Min.   :-630.0   Min.   :-584
##  1st Qu.:-242.0   1st Qu.: -54.0   1st Qu.:-145.0   1st Qu.:-297
##  Median : -42.0   Median :  14.0   Median : -48.0   Median : 298
##  Mean   : -59.9   Mean   :  32.8   Mean   : -72.8   Mean   : 196
##  3rd Qu.:  84.0   3rd Qu.: 139.0   3rd Qu.:  22.0   3rd Qu.: 640
##  Max.   : 431.0   Max.   : 308.0   Max.   : 271.0   Max.   : 782
##   magnet_arm_y   magnet_arm_z  roll_dumbbell    pitch_dumbbell
##  Min.   :-386   Min.   :-597   Min.   :-153.5   Min.   :-149.6
##  1st Qu.: -14   1st Qu.: 124   1st Qu.: -16.1   1st Qu.: -41.0
##  Median : 197   Median : 442   Median :  48.7   Median : -21.1
##  Mean   : 154   Mean   : 304   Mean   :  24.8   Mean   : -10.9
##  3rd Qu.: 323   3rd Qu.: 543   3rd Qu.:  68.3   3rd Qu.:  17.1
##  Max.   : 583   Max.   : 694   Max.   : 153.6   Max.   : 149.4
##   yaw_dumbbell     total_accel_dumbbell gyros_dumbbell_x  gyros_dumbbell_y
##  Min.   :-142.89   Min.   : 0.0         Min.   :-204.00   Min.   :-2.10
##  1st Qu.: -77.64   1st Qu.: 4.0         1st Qu.:  -0.03   1st Qu.:-0.14
##  Median :  -5.01   Median :10.0         Median :   0.13   Median : 0.03
##  Mean   :   1.17   Mean   :13.8         Mean   :   0.15   Mean   : 0.05
##  3rd Qu.:  78.54   3rd Qu.:20.0         3rd Qu.:   0.35   3rd Qu.: 0.21
##  Max.   : 154.75   Max.   :58.0         Max.   :   2.22   Max.   :52.00
##  gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z
##  Min.   : -2.4    Min.   :-419.0   Min.   :-189.0   Min.   :-284.0
##  1st Qu.: -0.3    1st Qu.: -51.0   1st Qu.:  -7.0   1st Qu.:-142.0
##  Median : -0.1    Median :  -9.0   Median :  44.0   Median :  -2.0
##  Mean   : -0.1    Mean   : -29.1   Mean   :  53.8   Mean   : -39.1
##  3rd Qu.:  0.0    3rd Qu.:  10.0   3rd Qu.: 112.0   3rd Qu.:  37.0
##  Max.   :317.0    Max.   : 224.0   Max.   : 310.0   Max.   : 318.0
##  magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z  roll_forearm
##  Min.   :-639      Min.   :-3600     Min.   :-245.0    Min.   :-180.00
##  1st Qu.:-535      1st Qu.:  233     1st Qu.: -44.0    1st Qu.:  -0.55
##  Median :-480      Median :  312     Median :  15.0    Median :  21.70
##  Mean   :-333      Mean   :  226     Mean   :  46.8    Mean   :  33.88
##  3rd Qu.:-312      3rd Qu.:  391     3rd Qu.:  95.0    3rd Qu.: 140.00
##  Max.   : 583      Max.   :  633     Max.   : 451.0    Max.   : 180.00
##  pitch_forearm     yaw_forearm     total_accel_forearm gyros_forearm_x
##  Min.   :-72.40   Min.   :-180.0   Min.   :  0.0       Min.   :-22.000
##  1st Qu.:  0.00   1st Qu.: -68.4   1st Qu.: 29.0       1st Qu.: -0.210
##  Median :  8.89   Median :   0.0   Median : 36.0       Median :  0.050
##  Mean   : 10.58   Mean   :  19.2   Mean   : 34.8       Mean   :  0.155
##  3rd Qu.: 28.30   3rd Qu.: 110.0   3rd Qu.: 41.0       3rd Qu.:  0.560
##  Max.   : 89.80   Max.   : 180.0   Max.   :108.0       Max.   :  3.970
##  gyros_forearm_y  gyros_forearm_z  accel_forearm_x  accel_forearm_y
##  Min.   : -6.65   Min.   : -7.94   Min.   :-498.0   Min.   :-595
##  1st Qu.: -1.46   1st Qu.: -0.18   1st Qu.:-180.0   1st Qu.:  57
##  Median :  0.03   Median :  0.08   Median : -57.0   Median : 200
##  Mean   :  0.10   Mean   :  0.16   Mean   : -61.9   Mean   : 162
##  3rd Qu.:  1.65   3rd Qu.:  0.49   3rd Qu.:  76.0   3rd Qu.: 311
##  Max.   :311.00   Max.   :231.00   Max.   : 477.0   Max.   : 923
##  accel_forearm_z  magnet_forearm_x magnet_forearm_y magnet_forearm_z
##  Min.   :-446.0   Min.   :-1280    Min.   :-896     Min.   :-966
##  1st Qu.:-182.0   1st Qu.: -615    1st Qu.:  -7     1st Qu.: 190
```

```
##  Median : -40.0   Median : -378    Median : 591     Median : 509
##  Mean   : -55.6   Mean   : -312    Mean   : 378     Mean   : 392
##  3rd Qu.:  26.0   3rd Qu.:  -72    3rd Qu.: 737     3rd Qu.: 651
##  Max.   : 291.0   Max.   :  672    Max.   :1480     Max.   :1090
##  classe
##  A:3348
##  B:2279
##  C:2054
##  D:1930
##  E:2165
##
```

The outcome variable `classe` contains 5 factors - `A`,`B`,`C`,`D`, and `E`. Because the outcome is classification to one of these 5 factors, classification decision tree would be the best algorithm to use for this project.

When `rpart` algorithm was used with default settings, `D` could not be classified. The in-sample accuracy was 52%, and it was not good enough. When `rpart` algorithm with `control = rpart.control(minsplit=30, cp=0.001)` was used, the in-sample accuracy improved to 92%, but in-sample sensitivity of `B` and `C` were only 88%. The out of sample accuracy was only 33%. (data not shown) Therefore, recursive partition and regression algorithm is not adequate.

When `random forest` algorithm was used, there was not enough memory in my computer, and due to the large number of variables and observations, it crashed.

Finally, boosting algorithm with `gbm` was used with default settings.

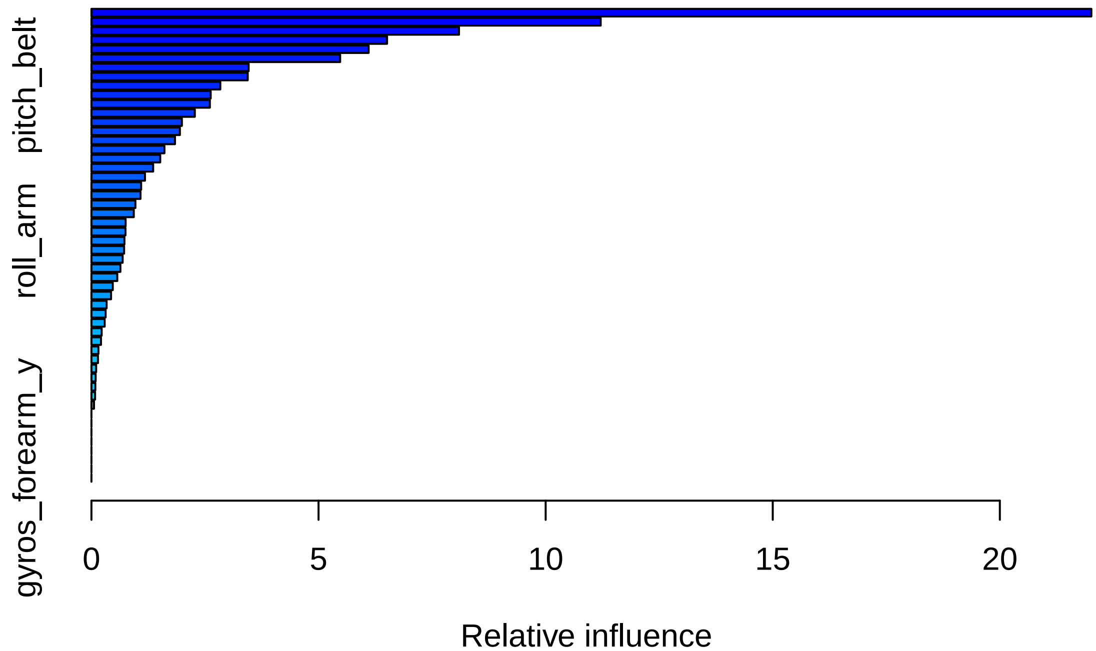
```
model1 <- train(classe ~., data=training, method="gbm", verbose = FALSE)
```

```
## Loading required package: gbm
## Loading required package: survival
## Loading required package: splines
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loading required package: parallel
## Loaded gbm 2.1
## Loading required package: plyr
```

Results

The relative influence of the variables are listed below. `roll_belt`, `pitch_forearm`, `yaw_belt`, `magnet_dumbbell_z`, `magnet_dumbbell_y`, and `roll_forearm` have the most influence in the prediction model.

```
summary(model1)
```




```
##                                    var  rel.inf
## roll_belt                    roll_belt 22.01468
## pitch_forearm            pitch_forearm 11.21205
## yaw_belt                      yaw_belt  8.09277
## magnet_dumbbell_z    magnet_dumbbell_z  6.50843
## magnet_dumbbell_y    magnet_dumbbell_y  6.10281
## roll_forearm              roll_forearm  5.47604
## gyros_belt_z              gyros_belt_z  3.46095
## magnet_belt_z            magnet_belt_z  3.44120
## pitch_belt                  pitch_belt  2.83745
## accel_forearm_x        accel_forearm_x  2.62575
## accel_dumbbell_y      accel_dumbbell_y  2.60743
## roll_dumbbell            roll_dumbbell  2.27593
## gyros_dumbbell_y      gyros_dumbbell_y  1.99107
## magnet_forearm_z      magnet_forearm_z  1.94771
## accel_dumbbell_x      accel_dumbbell_x  1.84011
## yaw_arm                        yaw_arm  1.60690
## magnet_dumbbell_x    magnet_dumbbell_x  1.51444
## accel_forearm_z        accel_forearm_z  1.35903
## magnet_belt_y            magnet_belt_y  1.17888
## magnet_arm_z              magnet_arm_z  1.09425
## accel_dumbbell_z      accel_dumbbell_z  1.08038
## magnet_forearm_x      magnet_forearm_x  0.96817
## accel_belt_z              accel_belt_z  0.93242
## gyros_belt_y              gyros_belt_y  0.75278
## magnet_arm_x              magnet_arm_x  0.74934
```

```
## roll_arm                           roll_arm  0.72636
## total_accel_dumbbell total_accel_dumbbell  0.71927
## gyros_arm_y                     gyros_arm_y  0.68668
## magnet_belt_x                 magnet_belt_x  0.63755
## magnet_arm_y                   magnet_arm_y  0.56885
## gyros_dumbbell_x           gyros_dumbbell_x  0.46956
## total_accel_forearm     total_accel_forearm  0.43239
## magnet_forearm_y           magnet_forearm_y  0.33428
## accel_arm_x                     accel_arm_x  0.31245
## accel_forearm_y             accel_forearm_y  0.29119
## pitch_dumbbell               pitch_dumbbell  0.22486
## accel_arm_y                     accel_arm_y  0.21001
## gyros_arm_x                     gyros_arm_x  0.15355
## gyros_forearm_z             gyros_forearm_z  0.14351
## yaw_dumbbell                   yaw_dumbbell  0.10381
## gyros_forearm_x             gyros_forearm_x  0.09226
## gyros_dumbbell_z           gyros_dumbbell_z  0.08567
## total_accel_arm             total_accel_arm  0.08025
## accel_arm_z                     accel_arm_z  0.05654
## total_accel_belt           total_accel_belt  0.00000
## gyros_belt_x                   gyros_belt_x  0.00000
## accel_belt_x                   accel_belt_x  0.00000
## accel_belt_y                   accel_belt_y  0.00000
## pitch_arm                         pitch_arm  0.00000
## gyros_arm_z                     gyros_arm_z  0.00000
## yaw_forearm                     yaw_forearm  0.00000
## gyros_forearm_y             gyros_forearm_y  0.00000
```

In-sample accuracy is 98% with sensitivity of the prediction > 95% and specificity > 99%. Out of sample accuracy is expected to be lower than 98%.

```r
inPred <- predict(model1)
confusionMatrix(training$classe, inPred)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 3317   22    5    2    2
##          B   45 2197   35    2    0
##          C    0   45 1991   15    3
##          D    0    1   41 1880    8
##          E    3   12   12   26 2112
## 
## Overall Statistics
## 
##                Accuracy : 0.976
##                  95% CI : (0.973, 0.979)
##     No Information Rate : 0.286
##     P-Value [Acc > NIR] : < 2e-16
## 
##                   Kappa : 0.97
##  Mcnemar's Test P-Value : 2.36e-08
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.986    0.965    0.955    0.977    0.994
## Specificity             0.996    0.991    0.993    0.995    0.995
## Pos Pred Value          0.991    0.964    0.969    0.974    0.976
## Neg Pred Value          0.994    0.992    0.990    0.995    0.999
## Prevalence              0.286    0.193    0.177    0.163    0.180
## Detection Rate          0.282    0.187    0.169    0.160    0.179
## Detection Prevalence    0.284    0.194    0.174    0.164    0.184
## Balanced Accuracy       0.991    0.978    0.974    0.986    0.994
```

After cross validation, out of sample accuracy is 96% with sensitivity of the prediction > 92% and specificity > 98%. Out of sample accuracy is only slightly lower than in-sample accuracy.

```r
outPred <- predict(model1, newdata=testing)
confusionMatrix(testing$classe, outPred)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2196   22    8    2    4
##          B   64 1416   35    3    0
##          C    0   29 1324   12    3
##          D    2    3   45 1224   12
##          E    1   14   19   22 1386
##
## Overall Statistics
##
##                Accuracy : 0.962
##                  95% CI : (0.957, 0.966)
##     No Information Rate : 0.288
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.952
##  Mcnemar's Test P-Value : 9.62e-13
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.970    0.954    0.925    0.969    0.986
## Specificity             0.994    0.984    0.993    0.991    0.991
## Pos Pred Value          0.984    0.933    0.968    0.952    0.961
## Neg Pred Value          0.988    0.989    0.983    0.994    0.997
## Prevalence              0.288    0.189    0.182    0.161    0.179
## Detection Rate          0.280    0.180    0.169    0.156    0.177
## Detection Prevalence    0.284    0.193    0.174    0.164    0.184
## Balanced Accuracy       0.982    0.969    0.959    0.980    0.989
```

Testing on the test set

The testing dataset consists of 20 observations. It is loaded and processed the same way as the training set.

```r
Testset <- read.csv("pml-testing.csv", header=TRUE)
Test <- Testset[, c(8,9,10,11,37:49,60:68,84:86,102,113:124,140,151:160)]
TestResult <- predict(model1, newdata=Test)
TestResult
```

```
##  [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
```

Based on the current model, prediction of all 20 cases were correct.