Investigating how well a unilateral dumbbell bicep curl was performed

Neil Caren

March 20, 2016

Summary

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it.

The goal of this reserach is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Environment Setup and data acquistion

The training data for this project are available here:

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

The test data are available here:

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

```
library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
setwd("~/Dropbox/scans/datascience/John Hopkins/Machine Learning")
training <- read.csv("pml-training.csv", stringsAsFactors = FALSE)
testing <- read.csv("pml-testing.csv", stringsAsFactors = FALSE)
```

Exploratory Data Analysis

The data has many variables with significant NA data poionts. After impact analysis it was decided to remove these columns along with other non-numewric variable that woudl get in the way of modeling

```
training2 <- training

for(x in ncol(training2):1) {
  if(anyNA(training2[,x]))
     training2[,x] <- NULL
}

training2ColNames <- colnames(training2)

for(x in ncol(training2):1) {
  #print(x)
  strTest <- substr(training2ColNames[x], 1,4)
  if(strTest == "kurt" | strTest == "skew" | strTest == "min_" | strTest == "max_" |
strTest == "ampl")
     training2[,x] <- NULL
}

training2$X<-NULL
training2$user_name<-NULL
training2$cvtd_timestamp<-NULL
training2$new_window<-NULL
```

Pre- processing

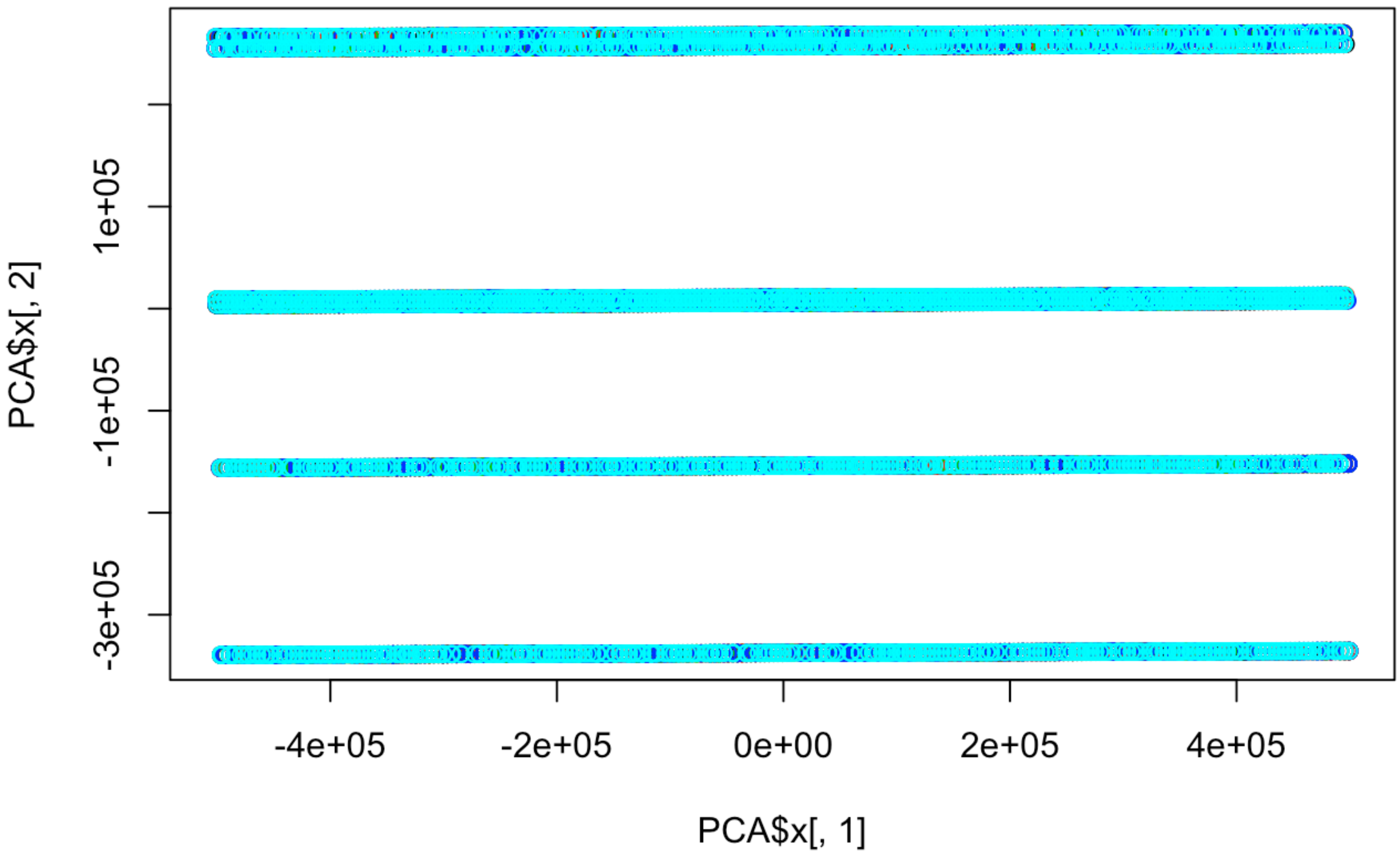
PCA analysis was performed and it was determind that while 2 PCA variables account for 99.9% of variance but there isn't a clear segregation of the data (see scatter Below) so I didn't continue with a PCA approach

```
PCA <- prcomp(training2[,-56])
summary(PCA)
```

```
## Importance of components:
##                              PC1       PC2   PC3   PC4   PC5   PC6   PC7
## Standard deviation     2.882e+05 2.049e+05 599.4 526.3 440.6 358.6 311.3
## Proportion of Variance 6.642e-01 3.358e-01   0.0   0.0   0.0   0.0   0.0
## Cumulative Proportion  6.642e-01 1.000e+00   1.0   1.0   1.0   1.0   1.0
##                          PC8   PC9  PC10  PC11  PC12  PC13  PC14  PC15
## Standard deviation     252.6 232.8 196.1 172.5 143.4 117.6 90.86 76.31
## Proportion of Variance   0.0   0.0   0.0   0.0   0.0   0.0  0.00  0.00
## Cumulative Proportion    1.0   1.0   1.0   1.0   1.0   1.0  1.00  1.00
##                         PC16 PC17  PC18  PC19  PC20  PC21 PC22  PC23  PC24
## Standard deviation     70.13 62.5 58.02 53.48 52.28 49.58 42.9 39.47 36.39
## Proportion of Variance  0.00  0.0  0.00  0.00  0.00  0.00  0.0  0.00  0.00
## Cumulative Proportion   1.00  1.0  1.00  1.00  1.00  1.00  1.0  1.00  1.00
##                         PC25  PC26  PC27  PC28  PC29  PC30  PC31  PC32
## Standard deviation     33.83 32.52 27.81 25.08 23.25 21.47 18.98 17.12
## Proportion of Variance  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## Cumulative Proportion   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
##                         PC33  PC34  PC35  PC36  PC37  PC38  PC39  PC40
## Standard deviation     14.99 13.98 9.752 7.431 6.957 6.674 6.124 4.118
## Proportion of Variance  0.00  0.00 0.000 0.000 0.000 0.000 0.000 0.000
## Cumulative Proportion   1.00  1.00 1.000 1.000 1.000 1.000 1.000 1.000
##                         PC41  PC42  PC43 PC44  PC45  PC46   PC47   PC48
## Standard deviation     3.644 3.471 3.314 1.95 1.487 1.078 0.4626 0.3933
## Proportion of Variance 0.000 0.000 0.000 0.00 0.000 0.000 0.0000 0.0000
## Cumulative Proportion  1.000 1.000 1.000 1.00 1.000 1.000 1.0000 1.0000
##                         PC49   PC50   PC51   PC52   PC53   PC54   PC55
## Standard deviation     0.359 0.3143 0.2409 0.2019 0.1859 0.1038 0.0368
## Proportion of Variance 0.000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
## Cumulative Proportion  1.000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
```

```
typeColor <- ifelse(training$classe == "A",1, ifelse(training$classe == "B", 2, ifels
e(training$classe == "C", 3,ifelse(training$classe == "D",4,5))))
```

Scatter



Model Selection

Prior to model selection I chose to center and scale the data. I first attempted a glm model but it failed to converge. Next I used a RF and was able to complete and predict 100% on the training set I partioned as a cross reference check. Code and results are detailed below

Step 1 Fit RF

```
pproc <- preProcess(training2[,-56], method=c("center", "scale"))
training3 <- predict(pproc, training2)
modFit <- train(classe~., method="rf", data=training3)
```

Partion / subset training data with by creating a index at random to set aside for a cross reference approach

```
randomSampleIndex <- sample (1:nrow(training3))
training4 <- training3[randomSampleIndex,]
pred4 <- predict(modFit, training4)

crossRefComp <- data.frame(pred4, training4$classe)
numErrors <- ifelse(pred4 == training4$classe,0,1)
```

Number of Errors after predicting the cross reference training data is equal to 0

Model evaluation

Looking at the confusion matrix shows that we have a very small error rat of just .06%

```
modFit$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: 28
##
##         OOB estimate of  error rate: 0.06%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5580    0    0    0    0 0.0000000000
## B    2 3794    1    0    0 0.0007900974
## C    0    4 3418    0    0 0.0011689071
## D    0    0    3 3212    1 0.0012437811
## E    0    0    0    1 3606 0.0002772387
```

With a strong model we move pre-process and the test data to evaluate the model results

Prepare Test data and run a prediction

```
testing2 <- testing

for(x in ncol(testing2):1) {
  if(anyNA(testing2[,x]))
     testing2[,x] <- NULL
}

testing2ColNames <- colnames(testing2)

for(x in ncol(testing2):1) {

  strTest <- substr(testing2ColNames[x], 1,4)
  if(strTest == "kurt" | strTest == "skew" | strTest == "min_" | strTest == "max_" |
strTest == "ampl")
     testing2[,x] <- NULL
}

testing2$X<-NULL
testing2$user_name<-NULL
testing2$cvtd_timestamp<-NULL
testing2$new_window<-NULL

testing3 <- predict(pproc, testing2)
predTest <- predict(modFit, testing3)
```

Result / Conclusion

The prediction came in at 100% accurate based on seperate quiz answers

Predictions for the test set is as follows

```
##  [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
```