Predicting the Quaity of a bicep Curl Execution

M. E. Crotzer

February 1, 2018

Summary

Qualitative Activity Recognition (QAR) has become an active area of research with broad applicability in sports medicine and, in the health area, motor skills recovery. The focus on QAR is to pridict how well the activity was performed and not which activity. In this study, 6 male subjects performed a Unilateral Dumbell Biceps Curl correctly (classe A), and purposefully, 4 incorrect ways (Classes B to E). Data was recorded on sensing devices on the glove, armband, belt, and dumbbell. A random forest analysis was used to "train" the recognition process. The model so trained was applied to a test set with a 0.994 recognition accuracy. The model was then run on a validation data set.

Data Retrieval and Preparation

The data was contained in 2 set: Training and testing. The training set had provisions for all of the sensing data on 19622 observations, but only a fraction had data. The testing set had 20 observations with identical provisions. both data sets were subset to inclde only the recorded data. The training set was divided into a training and test set to to test the model. The original training set of 20 was used as a validation data set.

```
ptrn<-read.csv('./pml-training.csv', header=TRUE)
pval<-read.csv('./pml-testing.csv', header=TRUE)

library("dplyr", lib.loc="~/R/win-library/3.4")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("caret", lib.loc="~/R/win-library/3.4")
```

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

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

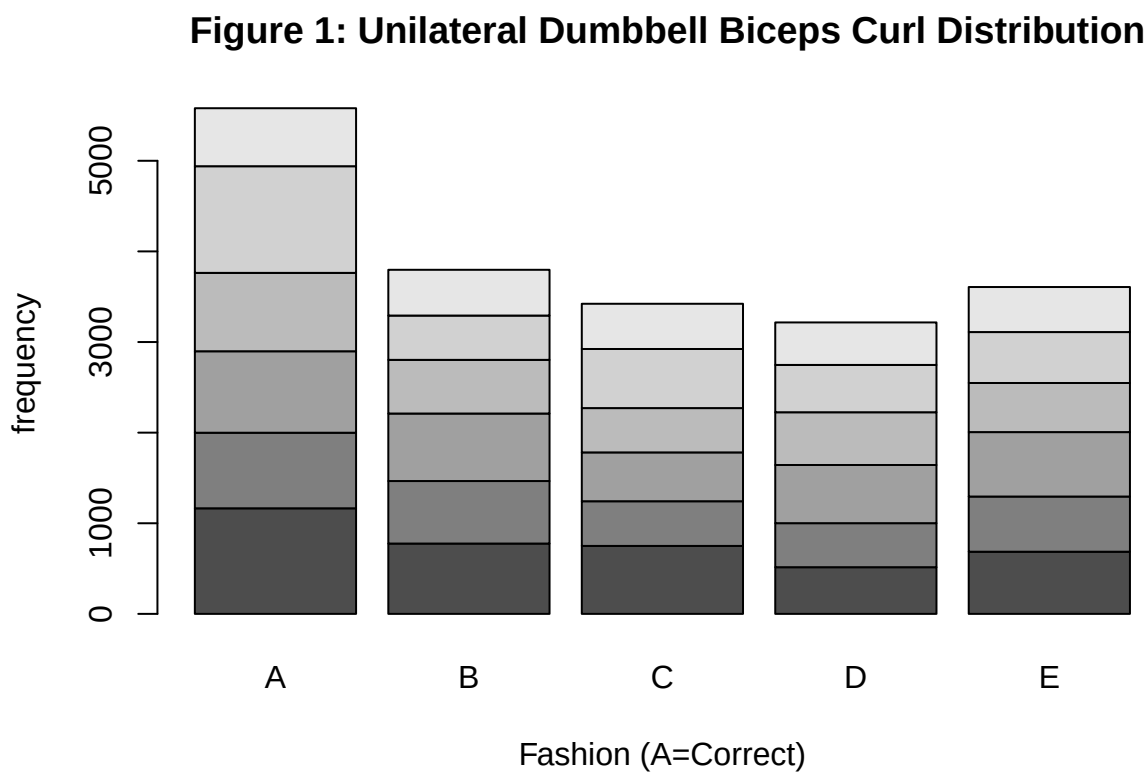
```
library("knitr", lib.loc="~/R/win-library/3.4")

ptrn1 <- select(ptrn, user_name,roll_belt:total_accel_belt, gyros_belt_x:total_accel_arm,
                gyros_arm_x:magnet_arm_z, roll_dumbbell:yaw_dumbbell,
                total_accel_dumbbell, gyros_dumbbell_x:yaw_forearm, total_accel_forearm,
                gyros_forearm_x:classe)

ptrn2 <- select(ptrn1, -user_name)

pval1 <- select(pval, roll_belt:total_accel_belt, gyros_belt_x:total_accel_arm,
```

```
                gyros_arm_x:magnet_arm_z, roll_dumbbell:yaw_dumbbell,
                total_accel_dumbbell, gyros_dumbbell_x:yaw_forearm, total_accel_forearm,
                gyros_forearm_x:magnet_forearm_z)
```

exploratory Data Analysis

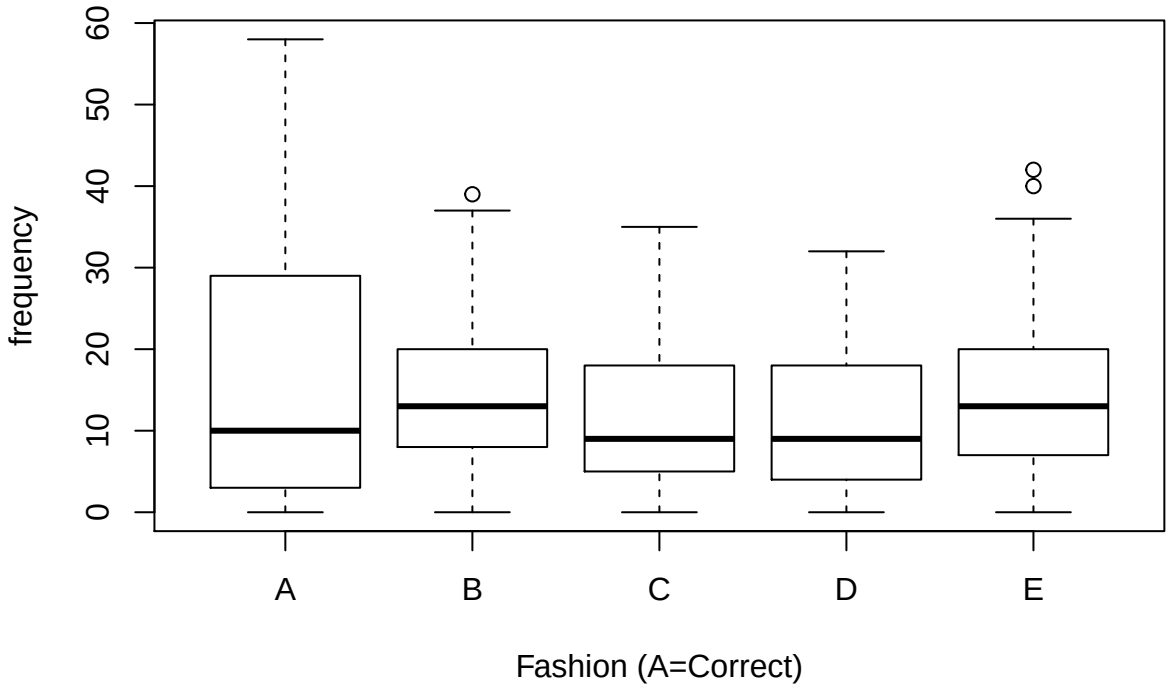
Summaries of some of the data is presented in Figures 1 to 3. figure 1 shows there were more correct xecutions relative to incorrect. The incorrect fashions were relatively uniform. With 52 variables, one can explore all aspects of the bicep curl. Figures 2 and 3 focus on Total dumbbell acceleration by fashion and participant.

```
tblFashions <- with(ptrn1, table(user_name,classe))
barplot(tblFashions, main = " Figure 1: Unilateral Dumbbell Biceps Curl Distribution",
        xlab="Fashion (A=Correct)", ylab="frequency")
```



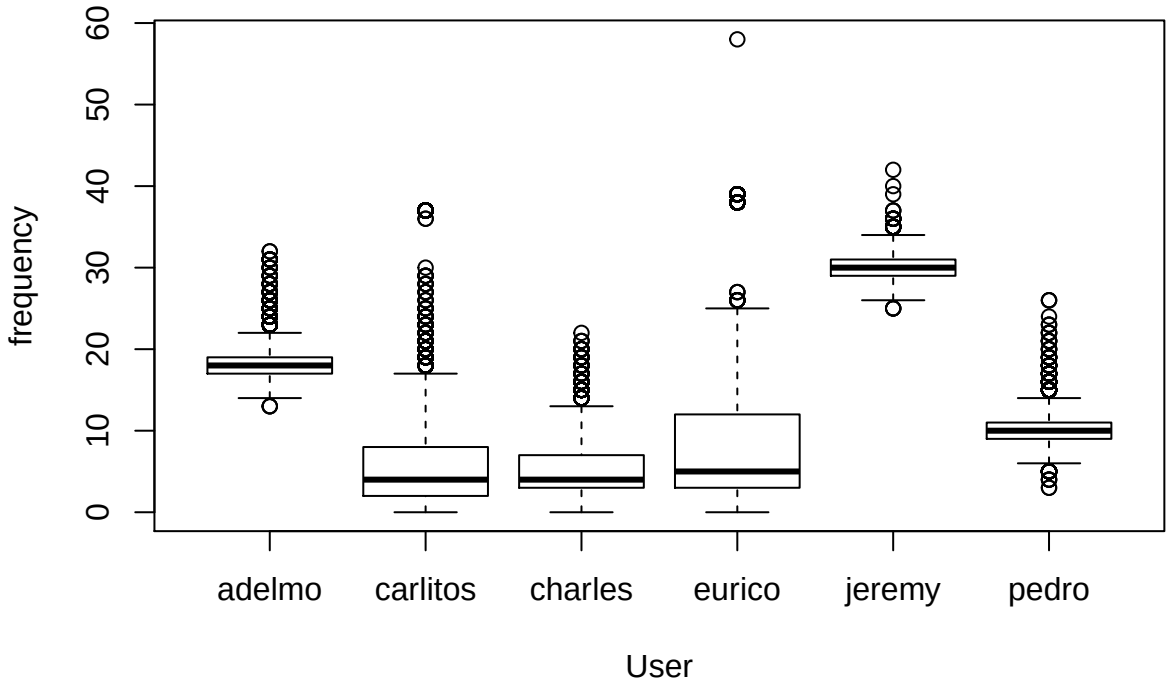

```
boxplot(ptrn1$total_accel_dumbbell ~ ptrn1$classe,
        main = "Figure 2:Total Dumbbell Acceleration by Fashion",
        xlab="Fashion (A=Correct)", ylab="frequency")
```

Figure 2:Total Dumbbell Acceleration by Fashion




```
boxplot(ptrn1$total_accel_dumbbell ~ ptrn1$user_name,
        main = "Figure 3: Total Dumbbell Acceleration by User",
        xlab="User", ylab="frequency")
```

Figure 3: Total Dumbbell Acceleration by User



Modelling

A random forest decision process was set up to model the dumbell execution. 5-fold cross validation was used and parallel processing was needed to converge to a model in reasonable time. Execution of the process took 10 minutes on a relatively old Lenovo computer. The in-model accuracy on 13737 samples was 0.9912.

```
## Training and Test set
set.seed(13118)
intrn <- createDataPartition(y=ptrn2$classe, p=0.7, list=FALSE)
trn <- ptrn2[intrn,]; tst <- ptrn2[-intrn,]

##  rf model with parallel processing
library(ParallelForest)
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
x <- trn[,-53]
y <- trn[,53]

cluster <- makeCluster(3)
registerDoParallel(cluster)
fitControl <- trainControl(method = "cv",
```

```
                           number = 5,
                           allowParallel = TRUE)

modFit <- train(x, y, method="rf", data=trn, trControl = fitControl)

stopCluster(cluster)
registerDoSEQ()
```

Prediction

The resulting model was applied to the test set carved out of the original training set. The predicted results were compared to the actual values via a confusion matrix. The accuracy on this set was 0.9944. The model was then run on the validation set. Presumably, there should be high confidence that this prediction is correct.

```
## Predict on Test set
print(modFit)
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10991, 10989, 10989, 10989, 10990
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9911920  0.9888575
##   27    0.9906094  0.9881208
##   52    0.9838399  0.9795566
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
predtrn <- predict(modFit, newdata=tst)
confusionMatrix(predtrn, tst$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1670    3    0    0    0
##          B    4 1134   11    0    0
##          C    0    2 1015   11    0
##          D    0    0    0  952    1
##          E    0    0    0    1 1081
##
## Overall Statistics
##
##                Accuracy : 0.9944
##                  95% CI : (0.9921, 0.9961)
```

```
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9929
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9976   0.9956   0.9893   0.9876   0.9991
## Specificity            0.9993   0.9968   0.9973   0.9998   0.9998
## Pos Pred Value         0.9982   0.9869   0.9874   0.9990   0.9991
## Neg Pred Value         0.9991   0.9989   0.9977   0.9976   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2838   0.1927   0.1725   0.1618   0.1837
## Detection Prevalence   0.2843   0.1952   0.1747   0.1619   0.1839
## Balanced Accuracy      0.9984   0.9962   0.9933   0.9937   0.9994
```

```
##Predict on validation set
predval <- predict(modFit, newdata=pval1)
table(predval)
```

```
## predval
## A B C D E
## 7 8 1 1 3
```