Practical Machine Learning - course project

Peter Kuzma

28. december 2015

Synopsis

Six young health participants 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).

Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Participants were supervised by an experienced weight lifter to make sure the execution complied to the manner they were supposed to simulate.

Executive Summary

For this course project our task was to build a prediction model how participants did the exercise. Training data was downloaded fromhttps://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv . Test data was downloaded fromhttps://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv .

In our model we used 57 out of 160 variables. The rest were removed because of missing values or value in variable could affect our prediction model. Our final model was built using LDA (Linear Discriminant Analysis), resulting in nearly 86% accuracy. Meaning our *out of sample error* was approximately 14%.

Exploratory Data Analysis

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

```
## Loading required package: lattice
## Loading required package: ggplot2
## Note: the specification for S3 class "family" in package 'MatrixModels' seems equival
ent to one from package 'lme4': not turning on duplicate class definitions for this clas
s.
```

```
# Load the data
```

```
pml_train <- read.table("../../Coursera/machine_learning/pml-training.csv",sep = ",", header=TRUE, na.strings=c("NA", "-", "?","","#DIV/0!"), stringsAsFactors=F)
pml_test <- read.table("../../Coursera/machine_learning/pml-testing.csv",sep = ",", header=TRUE, na.strings=c("NA", "-", "?","","#DIV/0!"), stringsAsFactors=F)
```

First we inspect the data and clean it up.

```
# check the data
summary(pml_train)
```

```
# let's do the list of columns with missing values
na_count <- sapply(pml_train, function(y) sum(length(which(is.na(y)))))
na_count <- data.frame(na_count)
na_count
```

```
# only keep columns with at least 50% non-blanks
# http://stackoverflow.com/questions/15968494/how-to-delete-columns-with-na-in-r?answertab=votes#tab-top
pml_train_clean <- pml_train[, colSums(is.na(pml_train)) < nrow(pml_train) * 0.5]
pml_test_clean <- pml_test[, colSums(is.na(pml_test)) < nrow(pml_test) * 0.5]

# remove columns that could affect our prediction model
pml_train_clean <- subset(pml_train_clean, select=-c(new_window, num_window))
pml_test_clean <- subset(pml_test_clean, select=-c(new_window, num_window))
pml_train_clean <- pml_train_clean[,-1]; pml_test_clean <- pml_test_clean[,-1]

# check we got in both the same fields - except for classe everything is in order
dim(pml_train_clean); dim(pml_test_clean)
```

```
## [1] 19622    57
```

```
## [1] 20 57
```

Building our prediction model

```
# for reproducibility
set.seed(8232)
```

We have a medium/large training set and validation set. Therefor we can easily split training set into training and testing set. Taking into account we have validation set we choose to split it 80:20.

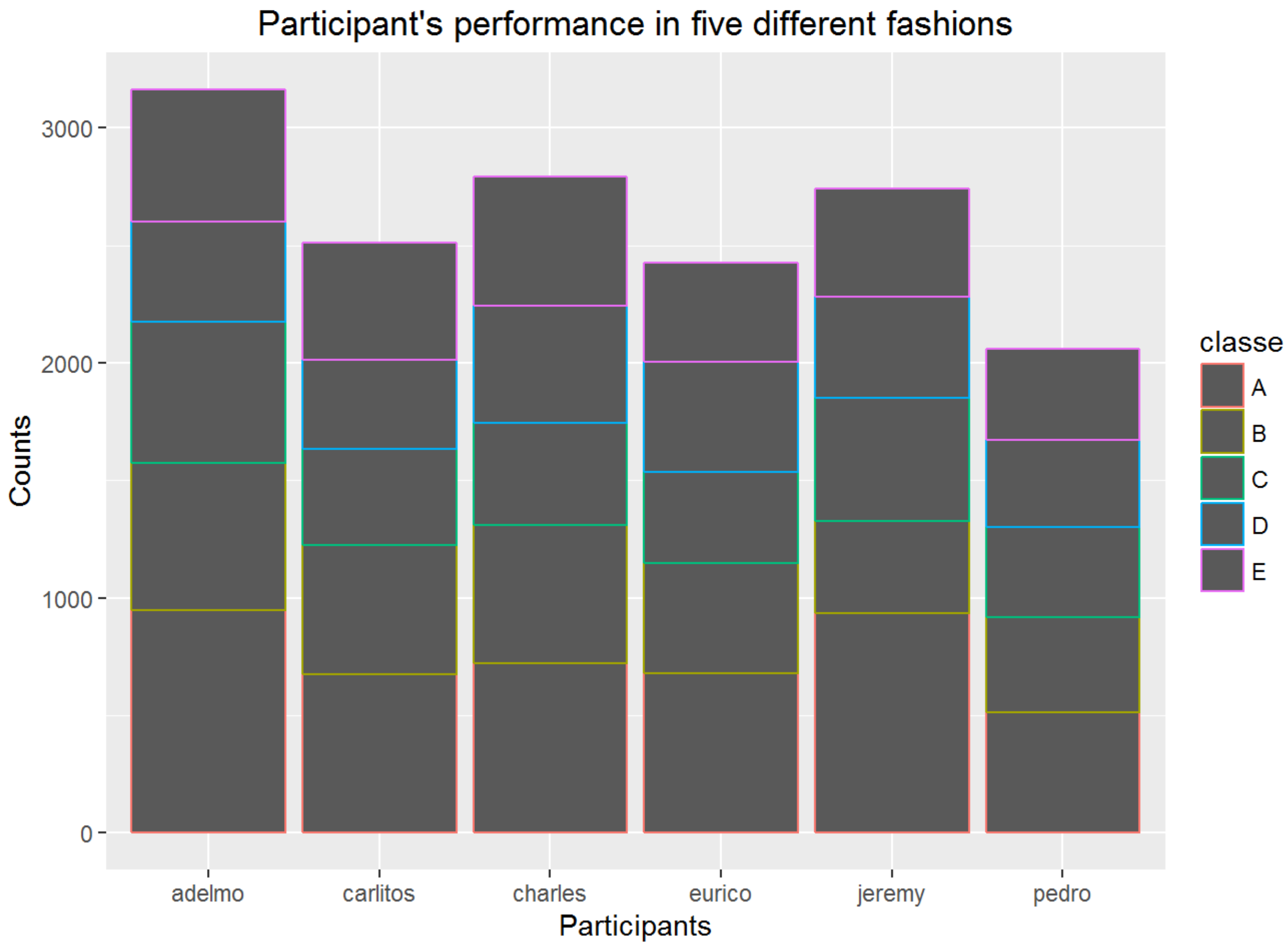
```
inTrain <- createDataPartition(y=pml_train_clean$classe,p=0.80, list=FALSE)
training <- pml_train_clean[inTrain,]
testing <- pml_train_clean[-inTrain,]

# let's check how well did our participants do the practice according to our trainning data
table(training$user_name,training$classe)
```

```
##
##              A   B   C   D   E
##   adelmo   945 630 599 429 562
##   carlitos 672 553 409 378 498
##   charles  721 588 437 496 552
##   eurico   678 470 388 467 423
##   jeremy   936 391 524 432 461
##   pedro    512 406 381 371 390
```

```
table(training$classe)
```

```
##
##    A    B    C    D    E
## 4464 3038 2738 2573 2886
```

```
qplot(user_name,colour=classe,data=training, main = "Participant's performance in five different fashions", xlab = "Participants", ylab = "Counts" )
```



Now it is time to build our prediction model. We chose to do it by LDA method.

```
#LDA
modlda <- train(classe ~ .,data=training,method="lda")
```

```
## Loading required package: MASS
```

```
plda <- predict(modlda,testing)
table(plda)
```

```
## plda
##    A    B    C    D    E 
## 1097  715  780  640  691
```

```
confusionMatrix(testing$classe,plda)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1012   90   13    0    1
##          B   84  553  115    7    0
##          C    1   71  593   17    2
##          D    0    1   54  552   36
##          E    0    0    5   64  652
## 
## Overall Statistics
##                                           
##                Accuracy : 0.857           
##                  95% CI : (0.8456, 0.8678)
##     No Information Rate : 0.2796          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.8193          
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9225   0.7734   0.7603   0.8625   0.9436
## Specificity            0.9632   0.9358   0.9710   0.9723   0.9787
## Pos Pred Value         0.9068   0.7286   0.8670   0.8585   0.9043
## Neg Pred Value         0.9697   0.9488   0.9423   0.9732   0.9878
## Prevalence             0.2796   0.1823   0.1988   0.1631   0.1761
## Detection Rate         0.2580   0.1410   0.1512   0.1407   0.1662
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9429   0.8546   0.8657   0.9174   0.9611
```

Our model has 85.7% accuracy. This means our out of sample error is 14.3%.

Now we use our prediction model on validation set.

```
pred <- predict(modlda,pml_test_clean)
table(pred)
```

```
## pred
## A B C D E
## 6 8 2 1 3
```

```
pred
```

```
##  [1] B B B A A E D C A A B C B A E E A B B B
## Levels: A B C D E
```

The results are used for the second part of the assignment - the programming portion of the Course Project.