Practical Machine Learning Course Project

Juho Pesonen

31 December 2016

Background

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. In this project, your goal will be 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 (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here:

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

The test data are available here:

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

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. From there you will be able to find more information about the data.

Purpose

The main purpose of this exercise if to find how accelerometer data can be used to predict wheter a person correctly executes unilater dumbbell biceps curl. For this purpose there is a variable classe in the data: 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).

Read more: http://groupware.les.inf.puc-rio.br/har#ixzz4UP7Rgq00

Reading data

```
trainingData<-read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
testingData<-read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")
```

Installing packages

```
require(ggplot2)
```

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

```
require(caret)
```

```
## Loading required package: caret

## Loading required package: lattice
```

```
require(randomForest)
```

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

Probing the data

```
dim(trainingData)
```

```
## [1] 19622   160
```

```
dim(testingData)
```

```
## [1]  20 160
```

```
nsv<-nearZeroVar(trainingData, saveMetrics=T)
nsv
```

```
##                          freqRatio percentUnique zeroVar   nzv
## X                         1.000000  100.00000000   FALSE FALSE
## user_name                 1.100679    0.03057792   FALSE FALSE
## raw_timestamp_part_1      1.000000    4.26562022   FALSE FALSE
## raw_timestamp_part_2      1.000000   85.53154622   FALSE FALSE
## cvtd_timestamp            1.000668    0.10192641   FALSE FALSE
## new_window               47.330049    0.01019264   FALSE  TRUE
## num_window                1.000000    4.37264295   FALSE FALSE
## roll_belt                 1.101904    6.77810621   FALSE FALSE
## pitch_belt                1.036082    9.37722964   FALSE FALSE
## yaw_belt                  1.058480    9.97349913   FALSE FALSE
```

```
## total_accel_belt           1.063160    0.14779329   FALSE FALSE
## kurtosis_roll_belt      1921.600000    2.02323922   FALSE  TRUE
## kurtosis_picth_belt      600.500000    1.61553358   FALSE  TRUE
## kurtosis_yaw_belt         47.330049    0.01019264   FALSE  TRUE
## skewness_roll_belt      2135.111111    2.01304658   FALSE  TRUE
## skewness_roll_belt.1     600.500000    1.72255631   FALSE  TRUE
## skewness_yaw_belt         47.330049    0.01019264   FALSE  TRUE
## max_roll_belt              1.000000    0.99378249   FALSE FALSE
## max_picth_belt             1.538462    0.11211905   FALSE FALSE
## max_yaw_belt             640.533333    0.34654979   FALSE  TRUE
## min_roll_belt              1.000000    0.93772296   FALSE FALSE
## min_pitch_belt             2.192308    0.08154113   FALSE FALSE
## min_yaw_belt             640.533333    0.34654979   FALSE  TRUE
## amplitude_roll_belt        1.290323    0.75425543   FALSE FALSE
## amplitude_pitch_belt       3.042254    0.06625217   FALSE FALSE
## amplitude_yaw_belt        50.041667    0.02038528   FALSE  TRUE
## var_total_accel_belt       1.426829    0.33126083   FALSE FALSE
## avg_roll_belt              1.066667    0.97339721   FALSE FALSE
## stddev_roll_belt           1.039216    0.35164611   FALSE FALSE
## var_roll_belt              1.615385    0.48924676   FALSE FALSE
## avg_pitch_belt             1.375000    1.09061258   FALSE FALSE
## stddev_pitch_belt          1.161290    0.21914178   FALSE FALSE
## var_pitch_belt             1.307692    0.32106819   FALSE FALSE
## avg_yaw_belt               1.200000    1.22311691   FALSE FALSE
## stddev_yaw_belt            1.693878    0.29558659   FALSE FALSE
## var_yaw_belt               1.500000    0.73896647   FALSE FALSE
## gyros_belt_x               1.058651    0.71348486   FALSE FALSE
## gyros_belt_y               1.144000    0.35164611   FALSE FALSE
## gyros_belt_z               1.066214    0.86127816   FALSE FALSE
## accel_belt_x               1.055412    0.83579655   FALSE FALSE
## accel_belt_y               1.113725    0.72877383   FALSE FALSE
## accel_belt_z               1.078767    1.52379982   FALSE FALSE
## magnet_belt_x              1.090141    1.66649679   FALSE FALSE
## magnet_belt_y              1.099688    1.51870350   FALSE FALSE
## magnet_belt_z              1.006369    2.32901845   FALSE FALSE
## roll_arm                  52.338462   13.52563449   FALSE FALSE
## pitch_arm                 87.256410   15.73234125   FALSE FALSE
## yaw_arm                   33.029126   14.65701763   FALSE FALSE
## total_accel_arm            1.024526    0.33635715   FALSE FALSE
## var_accel_arm              5.500000    2.01304658   FALSE FALSE
## avg_roll_arm              77.000000    1.68178575   FALSE  TRUE
## stddev_roll_arm           77.000000    1.68178575   FALSE  TRUE
## var_roll_arm              77.000000    1.68178575   FALSE  TRUE
## avg_pitch_arm             77.000000    1.68178575   FALSE  TRUE
## stddev_pitch_arm          77.000000    1.68178575   FALSE  TRUE
## var_pitch_arm             77.000000    1.68178575   FALSE  TRUE
## avg_yaw_arm               77.000000    1.68178575   FALSE  TRUE
## stddev_yaw_arm            80.000000    1.66649679   FALSE  TRUE
## var_yaw_arm               80.000000    1.66649679   FALSE  TRUE
## gyros_arm_x                1.015504    3.27693405   FALSE FALSE
## gyros_arm_y                1.454369    1.91621649   FALSE FALSE
## gyros_arm_z                1.110687    1.26388747   FALSE FALSE
## accel_arm_x                1.017341    3.95984099   FALSE FALSE
## accel_arm_y                1.140187    2.73672409   FALSE FALSE
```

```
## accel_arm_z                1.128000    4.03628580   FALSE FALSE
## magnet_arm_x               1.000000    6.82397309   FALSE FALSE
## magnet_arm_y               1.056818    4.44399144   FALSE FALSE
## magnet_arm_z               1.036364    6.44684538   FALSE FALSE
## kurtosis_roll_arm        246.358974    1.68178575   FALSE  TRUE
## kurtosis_picth_arm       240.200000    1.67159311   FALSE  TRUE
## kurtosis_yaw_arm        1746.909091    2.01304658   FALSE  TRUE
## skewness_roll_arm        249.558442    1.68688207   FALSE  TRUE
## skewness_pitch_arm       240.200000    1.67159311   FALSE  TRUE
## skewness_yaw_arm        1746.909091    2.01304658   FALSE  TRUE
## max_roll_arm              25.666667    1.47793293   FALSE  TRUE
## max_picth_arm             12.833333    1.34033228   FALSE FALSE
## max_yaw_arm                1.227273    0.25991234   FALSE FALSE
## min_roll_arm              19.250000    1.41677709   FALSE  TRUE
## min_pitch_arm             19.250000    1.47793293   FALSE  TRUE
## min_yaw_arm                1.000000    0.19366018   FALSE FALSE
## amplitude_roll_arm        25.666667    1.55947406   FALSE  TRUE
## amplitude_pitch_arm       20.000000    1.49831821   FALSE  TRUE
## amplitude_yaw_arm          1.037037    0.25991234   FALSE FALSE
## roll_dumbbell              1.022388   84.20650290   FALSE FALSE
## pitch_dumbbell             2.277372   81.74498012   FALSE FALSE
## yaw_dumbbell               1.132231   83.48282540   FALSE FALSE
## kurtosis_roll_dumbbell  3843.200000    2.02833554   FALSE  TRUE
## kurtosis_picth_dumbbell 9608.000000    2.04362450   FALSE  TRUE
## kurtosis_yaw_dumbbell     47.330049    0.01019264   FALSE  TRUE
## skewness_roll_dumbbell  4804.000000    2.04362450   FALSE  TRUE
## skewness_pitch_dumbbell 9608.000000    2.04872082   FALSE  TRUE
## skewness_yaw_dumbbell     47.330049    0.01019264   FALSE  TRUE
## max_roll_dumbbell          1.000000    1.72255631   FALSE FALSE
## max_picth_dumbbell         1.333333    1.72765263   FALSE FALSE
## max_yaw_dumbbell         960.800000    0.37203139   FALSE  TRUE
## min_roll_dumbbell          1.000000    1.69197839   FALSE FALSE
## min_pitch_dumbbell         1.666667    1.81429008   FALSE FALSE
## min_yaw_dumbbell         960.800000    0.37203139   FALSE  TRUE
## amplitude_roll_dumbbell    8.000000    1.97227602   FALSE FALSE
## amplitude_pitch_dumbbell   8.000000    1.95189073   FALSE FALSE
## amplitude_yaw_dumbbell    47.920200    0.01528896   FALSE  TRUE
## total_accel_dumbbell       1.072634    0.21914178   FALSE FALSE
## var_accel_dumbbell         6.000000    1.95698706   FALSE FALSE
## avg_roll_dumbbell          1.000000    2.02323922   FALSE FALSE
## stddev_roll_dumbbell      16.000000    1.99266130   FALSE FALSE
## var_roll_dumbbell         16.000000    1.99266130   FALSE FALSE
## avg_pitch_dumbbell         1.000000    2.02323922   FALSE FALSE
## stddev_pitch_dumbbell     16.000000    1.99266130   FALSE FALSE
## var_pitch_dumbbell        16.000000    1.99266130   FALSE FALSE
## avg_yaw_dumbbell           1.000000    2.02323922   FALSE FALSE
## stddev_yaw_dumbbell       16.000000    1.99266130   FALSE FALSE
## var_yaw_dumbbell          16.000000    1.99266130   FALSE FALSE
## gyros_dumbbell_x           1.003268    1.22821323   FALSE FALSE
## gyros_dumbbell_y           1.264957    1.41677709   FALSE FALSE
## gyros_dumbbell_z           1.060100    1.04984201   FALSE FALSE
## accel_dumbbell_x           1.018018    2.16593619   FALSE FALSE
## accel_dumbbell_y           1.053061    2.37488533   FALSE FALSE
## accel_dumbbell_z           1.133333    2.08949139   FALSE FALSE
```

```
## magnet_dumbbell_x          1.098266    5.74864948   FALSE FALSE
## magnet_dumbbell_y          1.197740    4.30129447   FALSE FALSE
## magnet_dumbbell_z          1.020833    3.44511263   FALSE FALSE
## roll_forearm              11.589286   11.08959331   FALSE FALSE
## pitch_forearm             65.983051   14.85577413   FALSE FALSE
## yaw_forearm               15.322835   10.14677403   FALSE FALSE
## kurtosis_roll_forearm    228.761905    1.64101519   FALSE  TRUE
## kurtosis_picth_forearm   226.070588    1.64611151   FALSE  TRUE
## kurtosis_yaw_forearm      47.330049    0.01019264   FALSE  TRUE
## skewness_roll_forearm    231.518072    1.64611151   FALSE  TRUE
## skewness_pitch_forearm   226.070588    1.62572623   FALSE  TRUE
## skewness_yaw_forearm      47.330049    0.01019264   FALSE  TRUE
## max_roll_forearm          27.666667    1.38110284   FALSE  TRUE
## max_picth_forearm          2.964286    0.78992967   FALSE FALSE
## max_yaw_forearm          228.761905    0.22933442   FALSE  TRUE
## min_roll_forearm          27.666667    1.37091020   FALSE  TRUE
## min_pitch_forearm          2.862069    0.87147080   FALSE FALSE
## min_yaw_forearm          228.761905    0.22933442   FALSE  TRUE
## amplitude_roll_forearm    20.750000    1.49322189   FALSE  TRUE
## amplitude_pitch_forearm    3.269231    0.93262664   FALSE FALSE
## amplitude_yaw_forearm     59.677019    0.01528896   FALSE  TRUE
## total_accel_forearm        1.128928    0.35674243   FALSE FALSE
## var_accel_forearm          3.500000    2.03343186   FALSE FALSE
## avg_roll_forearm          27.666667    1.64101519   FALSE  TRUE
## stddev_roll_forearm       87.000000    1.63082255   FALSE  TRUE
## var_roll_forearm          87.000000    1.63082255   FALSE  TRUE
## avg_pitch_forearm         83.000000    1.65120783   FALSE  TRUE
## stddev_pitch_forearm      41.500000    1.64611151   FALSE  TRUE
## var_pitch_forearm         83.000000    1.65120783   FALSE  TRUE
## avg_yaw_forearm           83.000000    1.65120783   FALSE  TRUE
## stddev_yaw_forearm        85.000000    1.64101519   FALSE  TRUE
## var_yaw_forearm           85.000000    1.64101519   FALSE  TRUE
## gyros_forearm_x            1.059273    1.51870350   FALSE FALSE
## gyros_forearm_y            1.036554    3.77637346   FALSE FALSE
## gyros_forearm_z            1.122917    1.56457038   FALSE FALSE
## accel_forearm_x            1.126437    4.04647844   FALSE FALSE
## accel_forearm_y            1.059406    5.11160942   FALSE FALSE
## accel_forearm_z            1.006250    2.95586586   FALSE FALSE
## magnet_forearm_x           1.012346    7.76679238   FALSE FALSE
## magnet_forearm_y           1.246914    9.54031189   FALSE FALSE
## magnet_forearm_z           1.000000    8.57710733   FALSE FALSE
## classe                     1.469581    0.02548160   FALSE FALSE
```

We notice that there are a lot of variables that have near zero variance. Even though some of them might be useful for this assignment I am still removing all TRUE variables from the further analysis to keep this straightforward.

```
trainingData<-trainingData[,-nearZeroVar(trainingData)]
```

We can still see that there are a lot of variables with missing values. I will now delete all variables with missing values so that they do not interfere with analysis. We also see that X, user_name, timestaps and num_window have nothing to do with accelometers so we remove them too.

```
trainingData<-trainingData[,!sapply(trainingData,function(x) any(is.na(x)))]
trainingData$X<- NULL; trainingData$user_name<-trainingData$raw_timestamp_part_1<-trainingData$raw_times
```

Fitting the model

Now the data is ready for analysis. I have chosen to use Random Forest approach as it is typically accurate method.

```
set.seed(1111)
inTrain<-createDataPartition(y=trainingData$classe, p=0.7, list=FALSE)
training<-trainingData[inTrain,]
testing<-trainingData[-inTrain,]
modelFit<-randomForest(classe~., data=training, type="class")
```

Data is cross-validated by taking 70 % of the data for training set and 30 % for testing set.

```
modelFit
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training, type = "class")
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.5%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3902    4    0    0    0 0.001024066
## B   12 2643    3    0    0 0.005643341
## C    0   13 2381    2    0 0.006260434
## D    0    0   25 2225    2 0.011989343
## E    0    0    3    4 2518 0.002772277
```

```
pred<-predict(modelFit,testing)
confusionMatrix(pred, testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    4    0    0    0
##          B    0 1134    5    0    0
##          C    0    1 1021   11    0
##          D    0    0    0  953    2
##          E    0    0    0    0 1080
##
## Overall Statistics
##
##                Accuracy : 0.9961
##                  95% CI : (0.9941, 0.9975)
```

```
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9951
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9956   0.9951   0.9886   0.9982
## Specificity            0.9991   0.9989   0.9975   0.9996   1.0000
## Pos Pred Value         0.9976   0.9956   0.9884   0.9979   1.0000
## Neg Pred Value         1.0000   0.9989   0.9990   0.9978   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1927   0.1735   0.1619   0.1835
## Detection Prevalence   0.2851   0.1935   0.1755   0.1623   0.1835
## Balanced Accuracy      0.9995   0.9973   0.9963   0.9941   0.9991
```

The results show that the model is 99.54 % accurate, making the expected out-of-sample error estimate 0.46%.We can now finally predict what the class would be for the testing data subjects:

```
predictfinal<-predict(modelFit, testingData, type="class")
predictfinal
```

```
##  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
```