Weight Lifting Exercises Quality

Executive Summary

The aim of this exercise is to estimate a prediction algorithm to forecast the manner in which a number of individuals do weight lifting. The data used in this exercise is from following paper:

- Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013. Available here

We use a Random Forest model in all numeric variables with no missing values in the data set. The pros of this approach is the high accuracy regularly achieved through this model while, in our particular case, the over-fitting not being a priori a big concern since all data is obtained in the same way and therefore error should be equally distributed in all cases (compared to the case where the data would have been obtained in different sessions and measurement equipment that would very probably bias the outcome of this model) Moreover, the number of cases is well balanced for all possible responses, so there risk of the random forest over-fitting the accuracy for one of the responses type but leaving the rest inaccurate is not big. On the other hand, random forest model is generally described as a "black box" model in which is difficult to understand what is the mechanism behind the model. Besides, this process requires a relative big amount of computation which may lead to performance problems in its estimation if the amount of data become very large or if the capacity of the computer is limited.

Selection of variables

The raw data set can be downloaded in this link
The complete data set includes one variable describing the possible outcome (5 factors from a to E) and 159 variables. There are 19,622 observations.
Many of the variables are not particularly useful (i.e.: names of the individual performing the exercise, time at which the exercise was performed) or have a large number of missing values. Thus, as a first step we pick only the variables with numeric values and no missing values in any case.

In this same step we split the complete data set into two separate sets:

1. Train data set containing 60% of the observations to train the data.
2. Test data containing 40% of observations to apply the model and get a reliable measure of the accuracy of the models.

The table in *appendix 1* shows the variables selected for the different trains sets in which train our model.

Training the model

We start by calculating parameters using a random forest model. We use for this the train set. This is, including the 27 variables in our original data that are numeric and have no missing values.

Below are displayed the results of this model in the training set. Package randomForest has been used using the complete set of 27 variables selected in previous section (see appendix 1).

```
##
## Call:
##  randomForest(formula = classe ~ ., data = train.data)
##                Type of random forest: classification
```

```
##                       Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 0.86%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3340    6    0    0    2    0.002389
## B   15 2251   12    0    1    0.012286
## C    0   14 2024   14    2    0.014606
## D    0    0   24 1904    2    0.013472
## E    0    3    1    5 2156    0.004157
```

Estimated Out of Bag error is 0.86 (or accuracy being 1-0.86=99.14%). This is measure is the equivalent of accuracy using cross validation (see link)
In *appendix 2* you can see a table with measured importance of the variables included in the model.

Testing the model

We will then apply the random forest estimated in previous step to test data.

Below the results of the confusion matrix and statistics:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2224    6    0    0    0
##          B    3 1504    8    0    1
##          C    2    7 1353   15    3
##          D    0    1    7 1269    5
##          E    3    0    0    2 1433
##
## Overall Statistics
##
##                Accuracy : 0.992
##                  95% CI : (0.99, 0.994)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.996    0.991    0.989    0.987    0.994
## Specificity             0.999    0.998    0.996    0.998    0.999
## Pos Pred Value          0.997    0.992    0.980    0.990    0.997
## Neg Pred Value          0.999    0.998    0.998    0.997    0.999
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.283    0.192    0.172    0.162    0.183
## Detection Prevalence    0.284    0.193    0.176    0.163    0.183
## Balanced Accuracy       0.998    0.994    0.992    0.992    0.996
```

The accuracy in the test data is above 99% (in line with oob error estimated in training section), so this seems to be a pretty good model for our purposes.

Graphs

Appendix 1: Variables selection

```
## 'data.frame':    11776 obs. of  28 variables:
##  $ classe           : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ roll_belt        : num  1.41 1.42 1.48 1.48 1.42 1.42 1.43 1.45 1.45 1.42 ...
##  $ pitch_belt       : num  8.07 8.07 8.05 8.07 8.09 8.13 8.16 8.17 8.18 8.21 ...
##  $ yaw_belt         : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 ...
##  $ gyros_belt_x     : num  0.02 0 0.02 0.02 0.02 0.02 0.02 0.03 0.03 0.02 ...
##  $ gyros_belt_y     : num  0 0 0 0.02 0 0 0 0 0 0 ...
##  $ gyros_belt_z     : num  -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 0 -0.02 -0.02 ...
##  $ roll_arm         : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -128 ...
##  $ pitch_arm        : num  22.5 22.5 22.1 22.1 21.9 21.8 21.7 21.6 21.5 21.4 ...
##  $ yaw_arm          : num  -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...
##  $ gyros_arm_x      : num  0.02 0.02 0.02 0 0 0.02 0.02 0.02 0.02 0.02 ...
##  $ gyros_arm_y      : num  -0.02 -0.02 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 -0.03 0 ...
##  $ gyros_arm_z      : num  -0.02 -0.02 0.02 0 0 0 -0.02 -0.02 0 -0.03 ...
##  $ roll_dumbbell    : num  13.1 12.9 13.4 13.4 13.1 ...
##  $ pitch_dumbbell   : num  -70.6 -70.3 -70.4 -70.4 -70.2 ...
##  $ yaw_dumbbell     : num  -84.7 -85.1 -84.9 -84.9 -85.1 ...
##  $ gyros_dumbbell_x : num  0 0 0 0 0 0 0 0 0 0.02 ...
##  $ gyros_dumbbell_y : num  -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 ...
##  $ gyros_dumbbell_z : num  0 0 -0.02 0 0 0 0 0 0 -0.02 ...
##  $ magnet_dumbbell_z: num  -64 -63 -60 -68 -70 -74 -65 -69 -64 -68 ...
##  $ roll_forearm     : num  28.3 28.3 28.1 28 27.9 27.8 27.7 27.7 27.6 27.2 ...
##  $ pitch_forearm    : num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63.8 -63.8 -63.8 -63.9 ...
##  $ yaw_forearm      : num  -153 -152 -152 -152 -152 -152 -152 -152 -152 -151 ...
##  $ gyros_forearm_x  : num  0.02 0.03 0.02 0.02 0.02 0.02 0.03 0.02 0.02 0 ...
##  $ gyros_forearm_y  : num  0 -0.02 -0.02 0 0 -0.02 0 0 -0.02 -0.02 ...
##  $ gyros_forearm_z  : num  -0.02 0 0 -0.02 -0.02 0 -0.02 -0.02 -0.02 -0.03 ...
##  $ magnet_forearm_y : num  661 658 658 655 659 660 653 656 657 659 ...
##  $ magnet_forearm_z : num  473 469 469 473 470 474 476 473 465 478 ...
```

Appendix 2: Random Forest variable importance

	MeanDecreaseGini
roll_belt	1189.20
pitch_belt	651.88
yaw_belt	860.99
gyros_belt_x	127.75
gyros_belt_y	146.66
gyros_belt_z	355.35
roll_arm	318.79
pitch_arm	175.04

	MeanDecreaseGini
yaw_arm	325.62
gyros_arm_x	138.40
gyros_arm_y	154.64
gyros_arm_z	78.89
roll_dumbbell	505.04
pitch_dumbbell	267.11
yaw_dumbbell	327.07
gyros_dumbbell_x	152.75
gyros_dumbbell_y	330.85
gyros_dumbbell_z	96.39
magnet_dumbbell_z	691.58
roll_forearm	545.34
pitch_forearm	764.04
yaw_forearm	214.39
gyros_forearm_x	90.02
gyros_forearm_y	133.49
gyros_forearm_z	90.48
magnet_forearm_y	247.01
magnet_forearm_z	331.24