Qualifying Weight Lifting Workout

Luca Lo Iacono

5/1/2021

Executive Summary

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 the project is to find a classifier that predicts how well the exercise is performed, using data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants as predictors. We tried to model different classifiers. We found three models that are supposed to have excellent accuracy in predicting how well the exercise is performed.

Acknowledgements

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har.

Exploratory Data Analysis

Participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways, in particular 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.

The exercises were performed by six male participants aged between 20-28 years, with little weight lifting experience. It was made sure that all participants could easily simulate the mistakes in a safe and controlled manner by using a relatively light dumbbell (1.25kg).

Read more: http://groupware.les.inf.puc-rio.br/har#ixzz6tnHbqlcO

Our goal is to predict the manner in which participants did the exercise. This is the "classe" variable in the training set (pml-training.csv). A value for "classe" is not given in pml-testing.csv, for which we have to try a prediciton after having build an effective model.

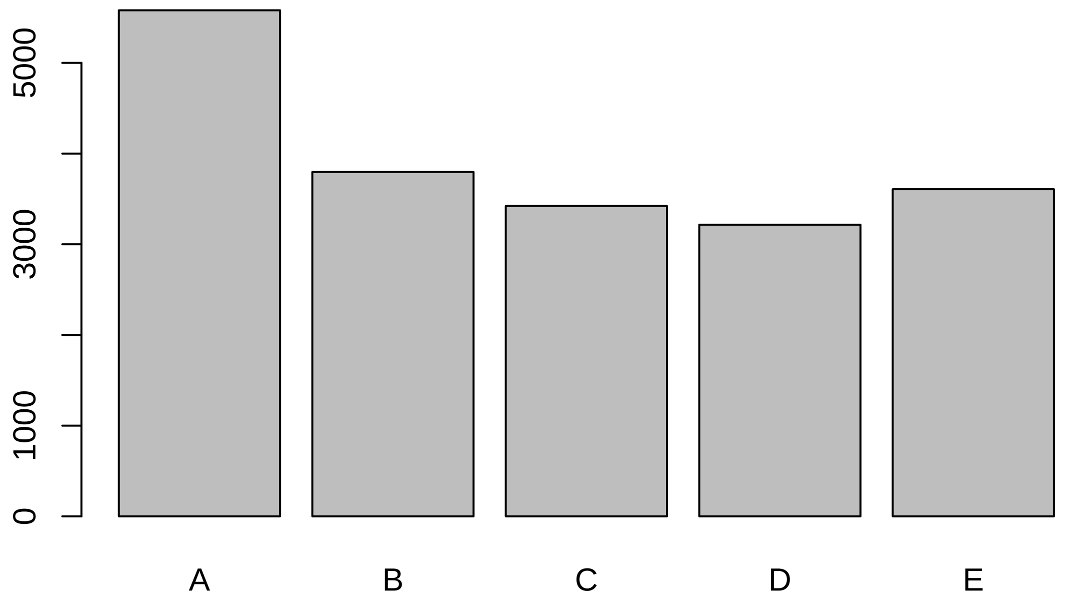
The *in sample* data set (pml-training.csv) includes *19622 observations* of **152 variables*.

After analyzing the structure of the data set we noticed that some features might be useless. Since we are trying to model a classifier that predicts the manner in which exercise are done, the model should be independent from the participants and from workout time. Therefore we will not include X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, new_window, num_window, cvtd_timestamp in our model.

Besides we are not going to include variables with too many NAs, such as max_roll_belt, max_picth_belt, min_roll_belt, etc., which have 19216 NAs out of 19622 observations.

And we are not going to include near zero variance predictors, such as X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, new_window, num_window, cvtd_timestamp.

The *in sample* data set looks to be quite balanced in terms of *classe* values:



Data Partition

We split the *in sample* data set in two data sets, which we are going to use to model (training data set) and to test (testing data set) each classifier. 70% of the *in sample* data set is dedicated to training data set, 30% to testing data set. In particular training data set includes 13737 observations and testing data set includes 5885 observations.

Classifier selection

Since we are trying to build a multiclass classifier using many different predictors we should focus our attention to Trees, which work well in non linear settings.

We should try three different models and check which performs better on the given data set:

- Bootstrap Aggregating Model
- Ramdom Forest Model
- Bosting with Tree Model

We use **10 fold cross validation** to tune model parameters.

Bootstrap Aggregating Model (TREEBAG)

```
## Bagged CART
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 12363, 12365, 12364, 12362, 12362, 12364, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9821652  0.9774378
```

Final Model

```
##
## Bagging classification trees with 25 bootstrap replications
```

Estimated *out of the bag* error for this model is **1.78%**.

Confusion Matrix

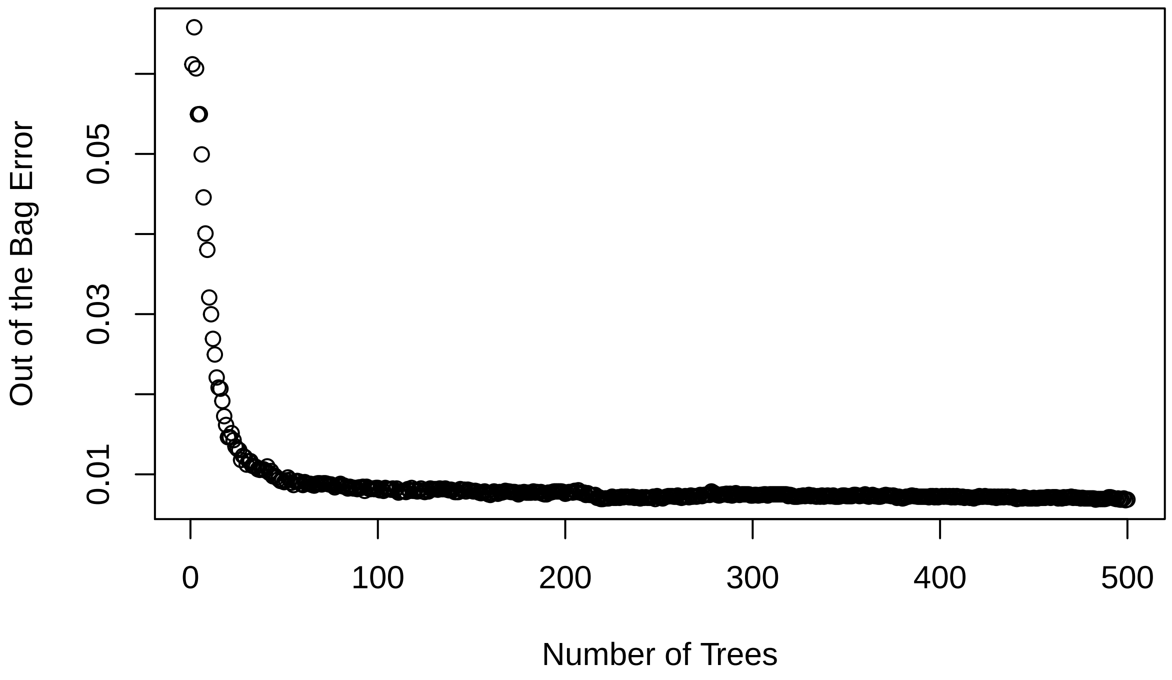
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1662   11    2    1    0
##          B    5 1122    8    7    1
##          C    3    2 1009    9    1
##          D    2    3    6  945    2
##          E    2    1    1    2 1078
##
## Overall Statistics
##
##                Accuracy : 0.9883
##                  95% CI : (0.9852, 0.9909)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9852
##
##  Mcnemar's Test P-Value : 0.3909
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9928   0.9851   0.9834   0.9803   0.9963
## Specificity            0.9967   0.9956   0.9969   0.9974   0.9988
## Pos Pred Value         0.9916   0.9816   0.9854   0.9864   0.9945
## Neg Pred Value         0.9971   0.9964   0.9965   0.9961   0.9992
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2824   0.1907   0.1715   0.1606   0.1832
## Detection Prevalence   0.2848   0.1942   0.1740   0.1628   0.1842
## Balanced Accuracy      0.9948   0.9903   0.9902   0.9888   0.9975
```

Ramdom Forest Model (RF)

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 12364, 12363, 12363, 12364, 12363, 12363, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9918467  0.9896851
##   27    0.9922837  0.9902378
##   52    0.9844216  0.9802932
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

Final Model

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, verbose = FALSE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.68%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3903    1    2    0    0 0.0007680492
## B   24 2623   11    0    0 0.0131677953
## C    0    9 2376   11    0 0.0083472454
## D    0    2   24 2224    2 0.0124333925
## E    0    1    2    5 2517 0.0031683168
```



Estimated *out of the bag* error for this model is **0.68%**.

Confusion Matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1671    7    0    0    0
##          B    0 1132    2    1    0
##          C    2    0 1022    8    1
##          D    0    0    2  954    3
##          E    1    0    0    1 1078
##
## Overall Statistics
##
##                Accuracy : 0.9952
##                  95% CI : (0.9931, 0.9968)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.994
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9939   0.9961   0.9896   0.9963
## Specificity            0.9983   0.9994   0.9977   0.9990   0.9996
## Pos Pred Value         0.9958   0.9974   0.9894   0.9948   0.9981
## Neg Pred Value         0.9993   0.9985   0.9992   0.9980   0.9992
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2839   0.1924   0.1737   0.1621   0.1832
## Detection Prevalence   0.2851   0.1929   0.1755   0.1630   0.1835
## Balanced Accuracy      0.9983   0.9966   0.9969   0.9943   0.9979
```

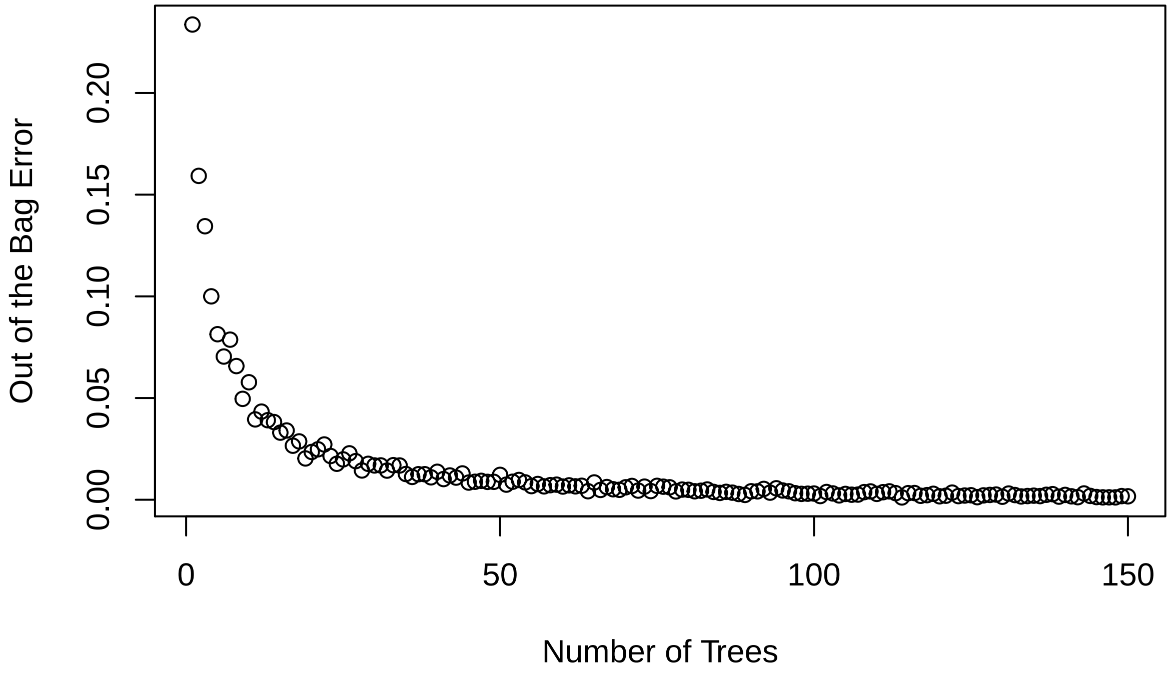
Boosting with Tree Model (GBM)

```
## Stochastic Gradient Boosting
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 12362, 12363, 12365, 12364, 12364, 12362, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7521331  0.6857057
##   1                  100      0.8241248  0.7774119
##   1                  150      0.8540456  0.8153238
##   2                   50      0.8540439  0.8151062
##   2                  100      0.9051455  0.8799516
##   2                  150      0.9298969  0.9112867
##   3                   50      0.8956094  0.8678241
##   3                  100      0.9399443  0.9240082
##   3                  150      0.9606906  0.9502621
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
```

```
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150, interaction.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

Final Model

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 52 predictors of which 52 had non-zero influence.
```



Estimated *out of the bag* error for this model is **0.17%**.

Confusion Matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1646   39    0    0    4
##          B   17 1071   23    5    9
##          C    8   26  990   34    8
##          D    2    2   11  917   15
##          E    1    1    2    8 1046
##
## Overall Statistics
##
##                Accuracy : 0.9635
##                  95% CI : (0.9584, 0.9681)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9538
```

```
##
##  Mcnemar's Test P-Value : 1.56e-06
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9833   0.9403   0.9649   0.9512   0.9667
## Specificity            0.9898   0.9886   0.9844   0.9939   0.9975
## Pos Pred Value         0.9745   0.9520   0.9287   0.9683   0.9887
## Neg Pred Value         0.9933   0.9857   0.9925   0.9905   0.9925
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2797   0.1820   0.1682   0.1558   0.1777
## Detection Prevalence   0.2870   0.1912   0.1811   0.1609   0.1798
## Balanced Accuracy      0.9865   0.9645   0.9746   0.9726   0.9821
```

Classifiers Comparison

We just fitted three different models:

- Bootstrap Aggregating Model with an accuracy of 98%
- Ramdom Forest Model with an accuracy of 99%
- Bosting with Tree Model with an accuracy of 96%

All the models have an *excellent* accuracy which is greater than 96%.

All the models have a *very similar* accuracy, which means that the three models have similar performances.

Here we can see that the prediction of the three models on the "out of sample" data. We added a column with a majority vote result to highlight the differences.

```
##     result.treebag result.rf result.gbm majority.vote
## V1               B         B          B             B
## V2               A         A          A             A
## V3               B         B          B             B
## V4               A         A          A             A
## V5               A         A          A             A
## V6               E         E          E             E
## V7               D         D          D             D
## V8               B         B          B             B
## V9               A         A          A             A
## V10              A         A          A             A
## V11              B         B          B             B
## V12              C         C          C             C
## V13              B         B          B             B
## V14              A         A          A             A
## V15              E         E          E             E
## V16              E         E          E             E
## V17              A         A          A             A
## V18              B         B          B             B
## V19              B         B          B             B
## V20              B         B          B             B
```

Conclusion

We found three different classifier with an accuracy greater than 96.

All the models have a *very similar* accuracy, which means that the three models have similar performances.

These models proved to predict correctly all the 20 *classe* values from *Out of Sample* data set.