Identification of Correct Barbell Exercise Execution Using Machine Learning

Sam Vacik

1/29/2022

Introduction

Human Activity Recognition (HAR) aims to identify movement type and performance based on physical data gathered by personal electronic devices such as Jawbone Up, Nike Fuelband, or Google Fitbit. Many users wear these devices during exercise, gain data with the desire to identify correct or poor execution, and then improve their technique In this data analysis, ten machine learning models will be built and tested to determine the best model capable of identifying the correct execution of a barbell lift independent of individual users. The best model will be used in the assessment to gauge effectiveness in apply practical machine learning to a real case to complete Practical Machine Learning through Coursera.

The dataset was available for download through the Practical Machine Learning Course Project web page on Coursera and information about the data was provided through the Human Activity Recognition data website (see the link in the following section). It was split into a different training and testing sets for this project. Only the training data is used in this project.

Setup, Exploration & Data Cleaning

The Human Activity Recognition dataset is utilized in this data analysis (see citation). The HAR dataset was collected from six participants who performed 10 repetitions of the dumbbell biceps curl in five ways, as documented in the "classe" variable in the dataset. Each participant worse a series of three-axial sensors positioned on the upper arm, wrist, waist, and one end of the barbell. Each variant of the biceps curl was assigned a classe letter from A to E. The class A variant represents correct execution of a biceps curl and the others correlate to incorrect execution. Each study participant possessed minimal experience of the dumbbell biceps curl and received supervision of their exercise execution by professional fitness experts. This study utilizes the training data set (file name "pml-training.csv") available through the HAR dataset.

This study utilizes the caret and ggplot2 packages in addition to custom functions written for conversion of alphabet letters to numbers, rounding numbers in a vector, and creation of a table containing the performance metrics against the test data.

Data exploration began by reading in the input training data set using the function "read.csv". The first seven columns contain data related to the user, date, time, and windows conducted. As the goal of this exercise is to build a prediction model of correct movements independent of user and temporal data, these columns will be removed from the dataset. The pml data set is 19622, 160 (rows by columns). Within the dataset, there are 1287472 NAs and 2589 indivisible numbers (i.e. divide by 0). Data cleaning removed the rows and columns associated with the NAs and indivisible numbers, reducing the dimensions of the dataset to 406 120 (rows by columns). In other words, the number of predictors reduced from 159 to 119 (a difference of 40). Lastly, the classe variable contains alphabetic letters (from "A" to "E" corresponding with various movements) that proved difficult to handle for the machine learning algorithms; the letters were converted to numbers of 1 to 5 (i.e. A replaced by 1, and so forth).

The reduced dataset contains 119 predictors and 406 rows of entries post data cleaning and remains excessive for potential feature selection, which could introduce bias into the results. This data analysis utilized a leave-one out (cross-validation) approach using random sampling with a chosen error rate of 0.8 to split the data set into a training and test set (using the function createDataPartition in the caret package). The resulting dimensions of the training set and the test set were 327 by 120 and 79 by 120 (rows by columns), respectively.

Initial data exploration occurred with a few variables such as total acceleration and roll per sensor. As can be seen in Figure 1, the data are fairly scattered in relation to classe, but there still appear possible linear and polynomial trends between some predictors and classe. The random sampling of the training data set into new training and test sets naturally resulted in different distributions as seen in Figure 2.

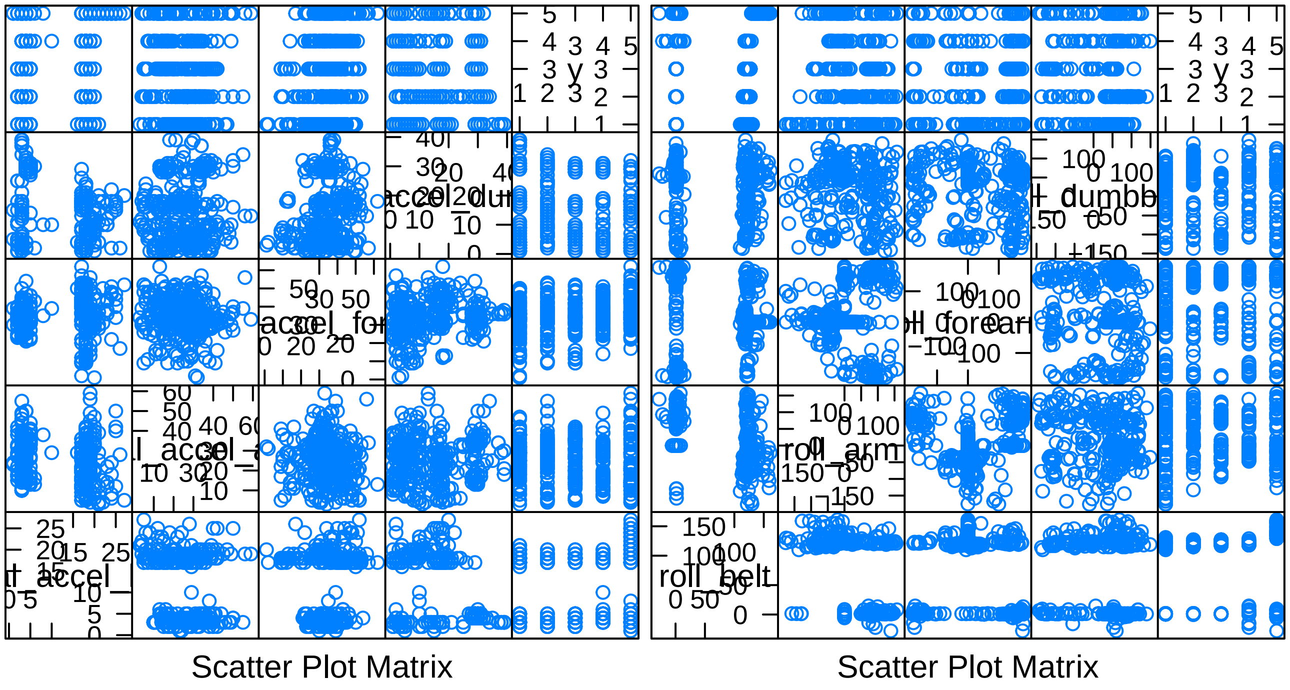

Figure 1 Comparison of Total Acceleration and Roll by Sensor against Classe (Y): (Left) Total Acceleration by Sensor VS Classe: Scatter plots showing various trends of total acceleration against the classe variable (represented in the chart as y) per sensor (belt, arm, forearm, and dumbbell). (Right) Roll Movement by Sensor VS Classe: Scatter plots showing various trends of roll against the classe variable (represented in the chart as y) per sensor (belt, arm, forearm, and dumbbell).

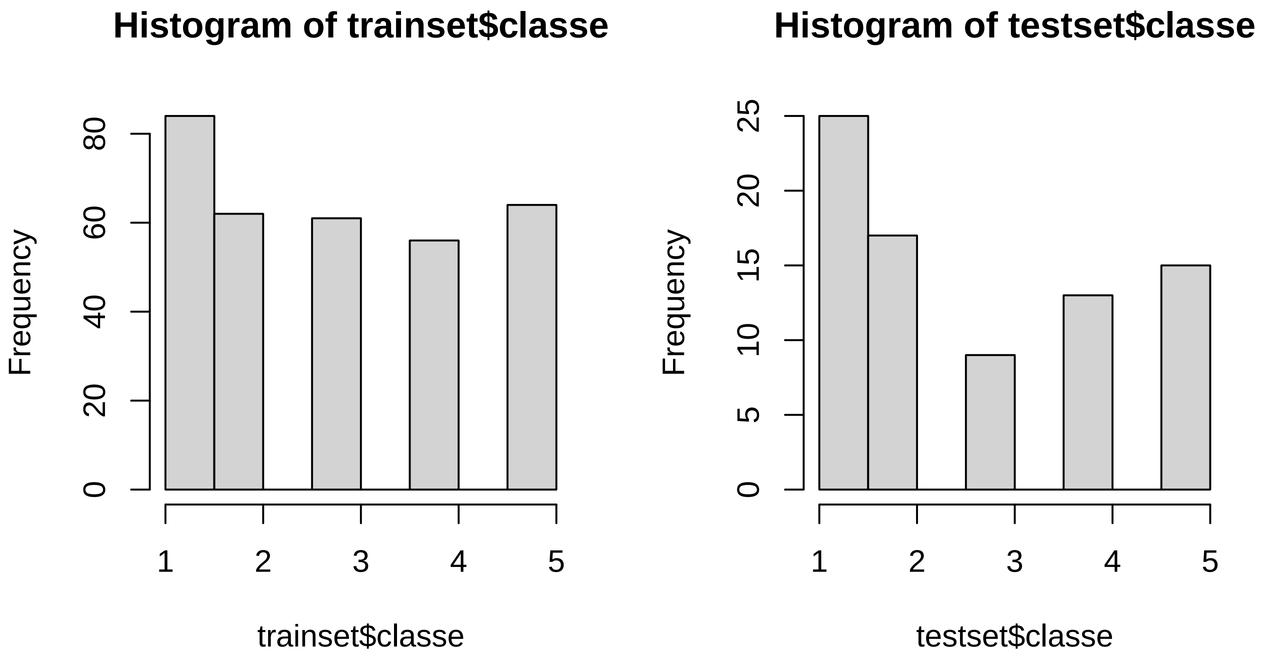

Figure 2 Histograms of the training and test sets as split from the original training set.

Data Analysis

The data analysis used a seed of 2134 to ensure reproducibility of the results. Each of the models were established using the "train" function from the R caret package.

This experiment tested ten machine learning methods available through the train function of the R package caret to determine which model will perform best in identifying correct barbell exercise execution (independent of user). The data analysis ran the train function with the corresponding model (parameter "method" in the function call) and then predicted movement type of the test data.

In this experiment, ten machine learning methods available through the R package caret are tested to determine which may perform best in identifying correct barbell exercise execution. The methods applied include: the general linear model (GLM); k-nearest neighbors (KKNN); bayesian ridge regression (BRIDGE); random forest (RF); treebagging (TBAG); model tree (M5); quantile random forest (QRF); multivariate adaptive regression spline (earth); cubist (cubist); and bayesian random neural networks (BRNN).

The best performing algorithm will be chosen based on which achieves the best performance and error metrics against the training and testing sets. The performance metrics include the r-squared (R^2), root mean square error (RMSE), and the mean absolute error (MAE). The error metrics include the counts of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) in addition to sensitivity (Sens), specificity (Specs), positive predictive value (PosPredVal), negative predictive value (NegPredVal), and the accuracy (acc). The best algorithm will have the highest R^2, lowest RMSE, and lowest MAE while maximizing the TP/TN values and minimizing the FP/FN values.

The performance and error metrics are captured in Tables 1 and 2, respectively. As seen in Table 1, the performance ranges by method and by the randomly sampled leave-one-out method, but the performance varies across the data split. Overall, the top three performers across the training data set in descending order of r-squared value are the quantile random forest (QRF), random forest (RF), and treebagging (TBAG). The worst three performers in order of increasing r-squared value for the training set are the general linear model (GLM), the quantile regression neural network (QRNN), and the quantile regression (QRNC). These results vary when the models were used to predict values based on the test data. The best perdictors on the test data in order of descending r-squared value are quantile random forest (QRF), random forest (RF), and cubist (cubist) and the worst performers in order of increasing r-squared are quantile regression neural network (QRNN), quantile regression (RQNC), and the multivariate adaptive regression (MARS-earth).

```
##        Method  Train-R2 Train-RMSE Train-MAE   Test-R2 Test-RMSE  Test-MAE
## 1         GLM 0.1848690 55.0850970 6.1268527 0.6187176 0.9480159 0.6455696
## 2        KKNN 0.4169321  1.1830059 0.6858383 0.6522942 0.9000703 0.6075949
## 3      Bridge 0.3306241  6.2467992 1.3183967 0.6522772 0.9000703 0.6075949
## 4          RF 0.6807919  0.9553373 0.8159487 0.7279482 0.7955573 0.4303797
## 5        TBAG 0.6116398  0.9154187 0.6620226 0.6701998 0.8787217 0.5189873
## 6        RQNC 0.3928540  1.6217876 0.8515746 0.5414896 1.0311601 0.6329114
## 7         QRF 0.7032213  0.8009105 0.4689357 0.7899015 0.7026172 0.3670886
## 8  MARS-Earth 0.3530723  1.1806507 0.9888217 0.5542678 1.0188104 0.6835443
## 9      Cubist 0.4603202  1.1706292 0.7528915 0.7178870 0.8113124 0.4050633
## 10       BRNN 0.4310083  1.1950587 0.8817220 0.6510782 0.9000703 0.5822785
```

Table 1 R2, RMSE, & MAE by Method: Comparison table showing the R2, RMSE, and MAE values by method against the training and testing datasets.

The trends found in the performance metrics reflect in the error metrics. The top performers according to the error metrics include random forest (RF), quantile random forest (QRF), and cubist, which produced the fewest FP and FN values of the ten algorithms tested. The differences between the random forest and quantile random forest models is very small (a difference of 0.0126582 in terms of accuracy). When comparing the ML algorithms by their performance and error metrics, it becomes apparent that the best performing algoirhtms are quantile random forest (QRF), random forest (RF), and cubist.

```
##        Method TP FP FN TN Sens     Specs PosPredVal NegPredVal       Acc
## 1         GLM 14  5 11 49 0.56 0.9074074  0.7368421  0.8166667 0.7974684
## 2        KKNN 11  3 14 51 0.44 0.9444444  0.7857143  0.7846154 0.7848101
## 3     Brridge 11  2 14 52 0.44 0.9629630  0.8461538  0.7878788 0.7974684
## 4          RF 17  0  8 54 0.68 1.0000000  1.0000000  0.8709677 0.8987342
## 5        TBAG 13  0 12 54 0.52 1.0000000  1.0000000  0.8181818 0.8481013
## 6        RQNC 12  0 13 54 0.48 1.0000000  1.0000000  0.8059701 0.8354430
## 7         QRF 16  0  9 54 0.64 1.0000000  1.0000000  0.8571429 0.8860759
## 8  MARS-Earth  9  0 16 54 0.36 1.0000000  1.0000000  0.7714286 0.7974684
## 9      Cubist 17  3  8 51 0.68 0.9444444  0.8500000  0.8644068 0.8607595
## 10       BRNN 12  2 13 52 0.48 0.9629630  0.8571429  0.8000000 0.8101266
```

Table 2 Error Statistics by Method: Comparison showing values of true positives (TP), false positives (FP), false negatives (FN), true negatives (TN), sensitivity (Sens), specificity (Specs), positive predictive value (PosPredVal), negative predictive value (NegPredVal), and accuracy (Acc) by ML algorithm.

The final models of the random forest and quantile random forest models are shown in Figure 4. The two models possess very similar performance and error metrics. The quantile random forest model performs better in terms of its performance metrics with a higher r-squared value and lower RMSE/MAE values than the random forest model and the random forest model finds one additional true positive compared to the quantile random forest. When examining their error versus number of trees plots, the differences are again slight, but the RF model achieves a smaller error rate than the QRF model.

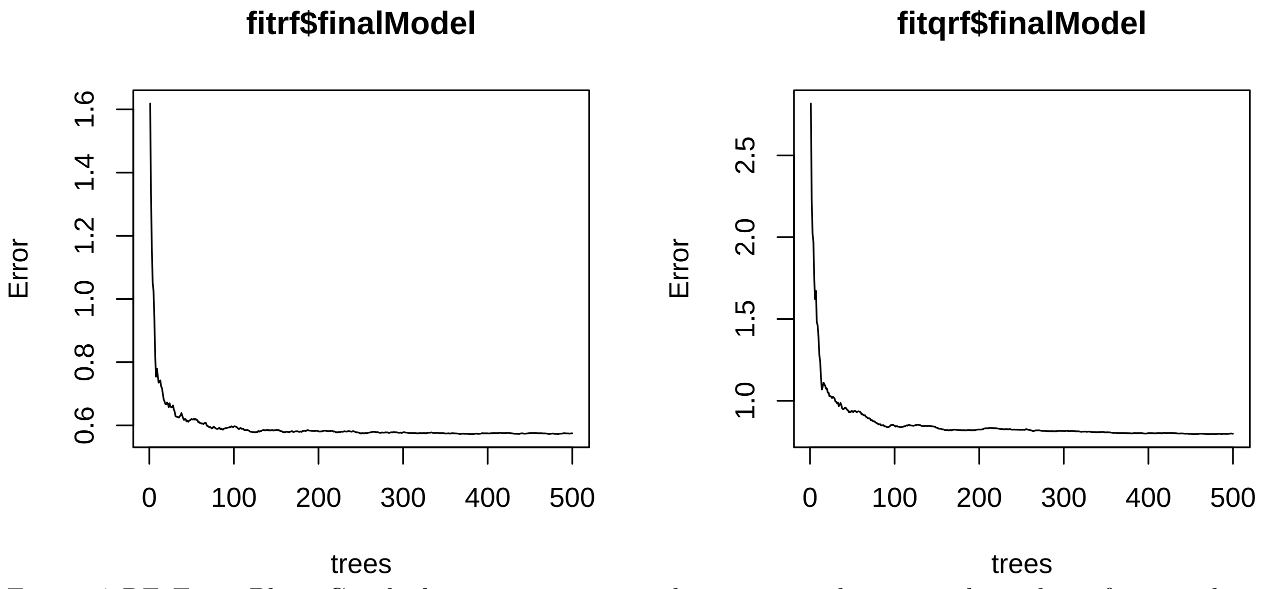

Figure 4 RF Error Plot: Graph depicting error rate decreasing with increased number of trees when implementing the quantile random forest machine learning algorithm.

Conclusions

Ten different ML algorithms were tested and compared to find the best predictor of activity recognition of a correct barbell curl. When conducting the data analysis, the quantile random forest (QRF) and random forest (RF) models produced the greatest number of true positives/false negatives and the lowest number of false positives/false negatives while achieving the highest r-squared values of the other methods tested. Between these two models, there are slight differences in their error statistics but greater difference in their performance metrics. The differences in r-squared, RMSE, and MAE values between these two algorithms are significant and the quantile random forest (QRF) model performed better than the random forest model by these metrics. The random forest (RF) model achieved one greater true positive than the quantile random forest (QRF) model and achieves a smaller error rate as the number of trees increases. In conclusion, the random forest (RF) model performs best and will be used to test against the quiz data set (aka the test data set included in the file “pml-testing.csv”) for the assessment in this final project of the Practical Machine Learning course on Coursera.

Citation

Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6_6.