Identification of Correct Barbell Exercise Execution Using Machine Learning

Samantha Vacik

2/3/2022

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.

Exploratory Analysis & 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.

Data exploration and cleaning began by examination of the training set. The testing dataset will be similarly be examined and cleaned up before application of the chosen ML algorithm. The first seven columns of each data set 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 were removed from the dataset. The training set was (19622, 160) (rows by columns). The data set contained NA and "#DIV/0!" values and were removed. This reduced the training and testing sets to (19622, 53) rows by columns. The number of predictors was 52 variables. Lastly, the classe variable contains alphabetic letters (from "A" to "E" corresponding with various barbell executions) that proved difficult to handle for the machine learning algorithms (even with setting the classe variable to a factor); the letters were converted to numbers of 1 to 5 (i.e. "A" replaced by 1, "B" replaced by 2, and so forth).

This data analysis utilized a leave-one out cross-validation (LOOCV) approach using random sampling with a chosen error rate of 0.75 to split the original training data set into a training and testing subsets.

Initial data exploration occurred with a few variables. The data were 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 testing subsets resulted in similar distribution as seen in Figure 1. There is a higher frequency of the classe variable equating to values of 1, 2, and 3 in the training subset than in the testing subset. Additionally, the frequencies of classe equating to 3 and 4 in the testing subset are nearly equal. These slight differences between the two distributions may lead to bias in some of the models.

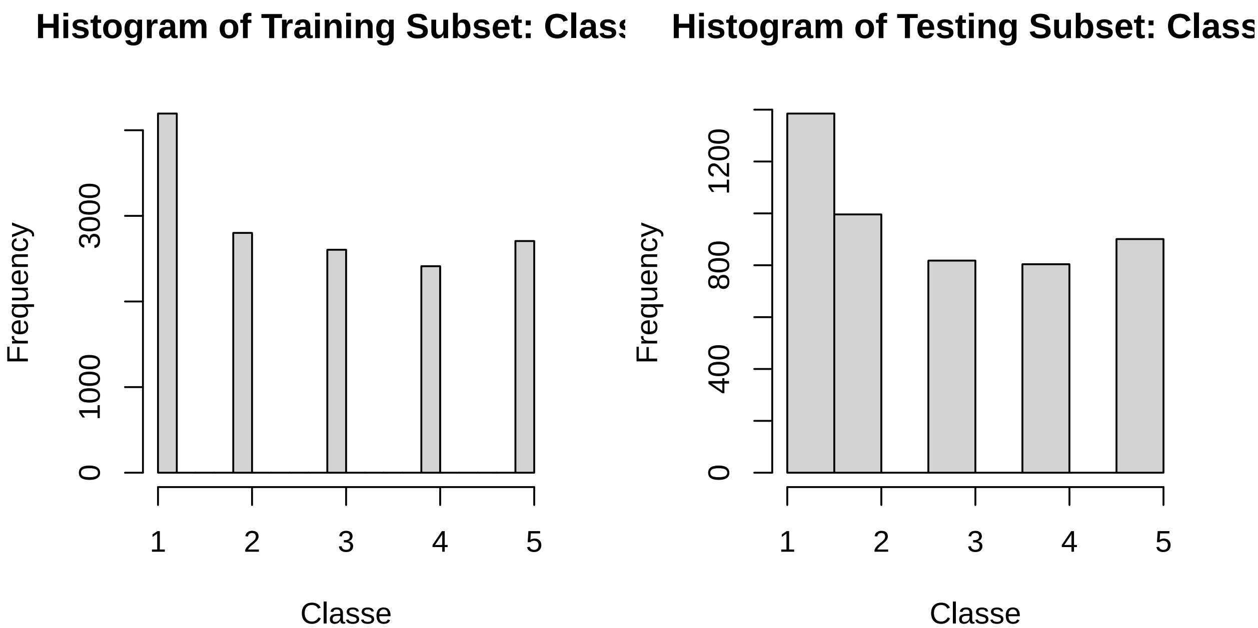

Figure 1 Training/Testing Subset Histograms: Histograms of the training and testing subsets as split from the original training set (pml-training.csv from the HAR dataset) by random sampling in the LOOCV method.

Setup & Data Analysis

As seen in Table 1, this experiment tested ten machine learning methods available through the train function of the R package caret to determine which model performed best in identifying correct barbell exercise execution (independent of user and denoted by classe variable equal to "A" or 1) and the predict function was used to predict correct movement from each model on the testing subset.

```
##                                               Method Acronyms
## 1                               General Linear Model      GLM
## 2                                K-Nearest Neighbors     KKNN
## 3                          Bayesian Ridge Regression   BRIDGE
## 4                                      Random Forest       RF
## 5                                        Treebagging     TBAG
## 6                                       Model Tree 5       M5
## 7                            Quantile Random Forest       QRF
## 8  Multivariate Adaptive, Regression (MARS) - earth    Earth
## 9                                             Cubist   Cubist
## 10                     Bayesian Random Neural Networks     BRNN
```

Table 1 Machine Learning (ML) Methods: Machine Learning methods used in this experiment and acronym definitions.

The best performing algorithm was chosen based on which achieved the best performance and error metrics against the training and testing sets. The performance metrics included the r-squared (R^2), root mean square error (RMSE), and the mean absolute error (MAE). The error metrics included 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). For example, the best algorithm would achieve the highest R^2, lowest RMSE, and lowest MAE while maximizing the TP/TN values and minimizing the FP/FN values. For additional consideration between models with similar performance and errors statistics based on the correct barbell execution (classe equaled “A” or 1), the final model error rates were plotted along with the confusion matrices for just the top performing models.

Results

As seen in Table 2, the performance ranges by method and by the randomly sampled LOOCV method used to split the training data set. Overall, the top performers across the training subset included the quantile random forest (QRF), k-nearest neighbors (KKNN), random forest (RF), cubist (cubist), and the model tree (M5), which all achieved an R^2 value of greater than 0.9 for both the training and testing subsets. The worst performers included the multivariate adaptive regression (earth), general linear model (GLM), and the bayesian ridge regression (bridge).

These results vary when the models were used to predict values based on the testing subset. The same best and worse performers occur, but with some slight changes. The best performers improved in terms of R^2 value overall with use of the testing subset. For example, the model tree (M5) R^2 value changed from ~0.914 on the training subset to ~0.941 on the testing subset. Similarly, two of the worst performing ML algorithms – namely the Bridge and GLM models – R^2 value decreased from the training subset to the testing subset. Interestingly, the worst performing algorithm (MARS-earth) R^2 value increased significantly across the data split. The BRNN model R^2 also increased. These trends seem to reflect the differences noted earlier between the two distributions and the LOOCV approach by random sampling contributed to this occurrence in addition to the method inherent in each model.

```
##        Method   Train-R2 Train-RMSE  Train-MAE   Test-R2  Test-RMSE    Test-MAE
## 1         GLM 0.51866484  1.0267881 0.81233547 0.5009558 1.04223611 0.746533442
## 2        KKNN 0.98150127  0.2015078 0.02620893 0.9928901 0.12448917 0.011419250
## 3      Bridge 0.51855468  1.0270046 0.81315647 0.4969160 1.04643417 0.749592170
## 4          RF 0.96497819  0.3345649 0.23934346 0.9886123 0.15772651 0.022838499
## 5        TBAG 0.69436695  0.8205958 0.61610857 0.6911030 0.82068984 0.529159869
## 6          M5 0.91456590  0.4327824 0.17765990 0.9415010 0.35927469 0.060154976
## 7         QRF 0.99015815  0.1471082 0.01984127 0.9958864 0.09472205 0.006525285
## 8  MARS-Earth 0.06634506  1.4303608 1.24462268 0.6047794 0.92742342 0.624388254
## 9      Cubist 0.94477782  0.3490195 0.14952203 0.9956092 0.09789797 0.008768352
## 10       BRNN 0.54389800  0.9993224 0.76685919 0.7563905 0.72813422 0.436378467
```

Table 2 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 reflected in the error metrics, as seen in Table 3. The top performers according to the error metrics included quantile random forest (QRF), cubist (Cubist), K-nearest neighbors (KKNN), and random forest (RF), which all produced the fewest FP and FN values of the ten algorithms tested. The differences in terms of the performance and error metrics between these four algorithms is small. Additional looks at the error versus model parameters and their confusion matrices helped to narrow the decision of which model to choose between the top 4: k-nearest neighbors (KKNN), random forest (RF), quantile random forest (QRF), and Cubist (cubist).

```
##        Method   TP  FP  FN   TN      Sens     Specs PosPredVal NegPredVal
## 1         GLM  537  61 848 3458 0.3877256 0.9826655  0.8979933  0.8030655
## 2        KKNN 1382   7   3 3512 0.9978339 0.9980108  0.9949604  0.9991465
## 3     Brridge  534  56 851 3463 0.3855596 0.9840864  0.9050847  0.8027353
## 4          RF 1364   5  21 3514 0.9848375 0.9985791  0.9963477  0.9940594
## 5        TBAG  778 119 607 3400 0.5617329 0.9661836  0.8673356  0.8485151
## 6          M5 1364  31  21 3488 0.9848375 0.9911907  0.9777778  0.9940154
## 7         QRF 1384   5   1 3514 0.9992780 0.9985791  0.9964003  0.9997155
## 8  MARS-Earth  750  39 635 3480 0.5415162 0.9889173  0.9505703  0.8456865
```

```
## 9      Cubist 1384   0   1 3519 0.9992780 1.0000000  1.0000000  0.9997159
## 10       BRNN 1004  79 381 3440 0.7249097 0.9775504  0.9270545  0.9002879
##          Acc
## 1  0.8146411
## 2  0.9979608
## 3  0.8150489
## 4  0.9946982
## 5  0.8519576
## 6  0.9893964
## 7  0.9987765
## 8  0.8625612
## 9  0.9997961
## 10 0.9061990
```

Table 3 Error Statistics by Method for the Correct Barbell Execution (classe = A or 1): 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 (RF), quantile random forest (QRF), and K-nearest neighbors models are shown in Figure 2. The k-nearest neighbors model with fewer number of clusters as shown by the lower mean squared error for k = 4 or less. The quantile random forest and random forest model achieved almost identical plots of error versus number of trees. Both were able to achieve an error rate of 0.05 with less than 100 trees. The k-nearest neighbors model performed better than the random forest model overall in terms of its performance and error metrics, but both perform worse than the cubist and quantile random forest models in terms of their error metrics. The quantile random forest model and the cubist model also possess very similar performance and error metrics. An additional look at the confusion matrices for these four models was performed.

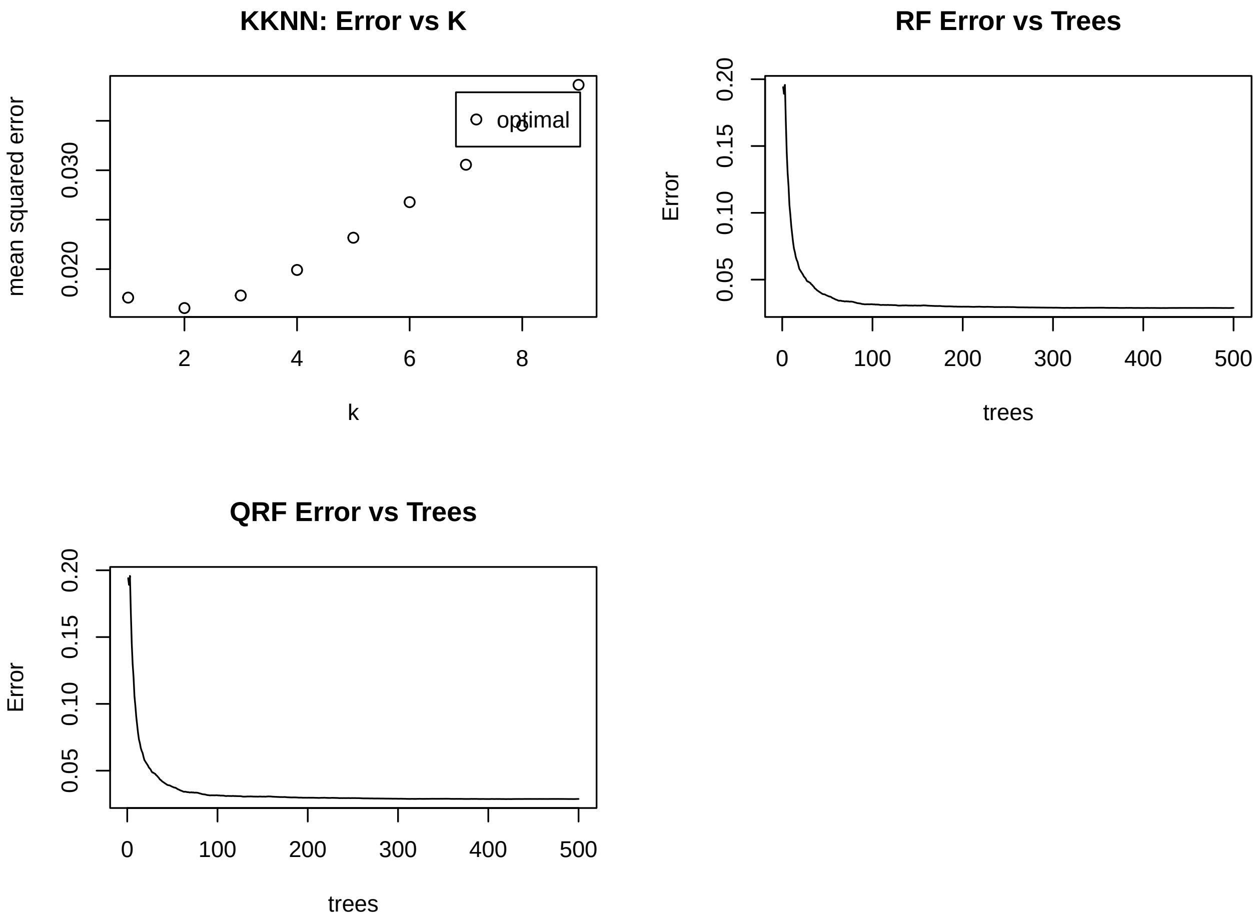

Figure 2 Model Error Plots: Plots of error rates for the KKNN, RF, and QRF final models showing changes in error rate with respect to k number of clusters, error versus RF number of trees trees, and error versus QRF number of trees, respectively.

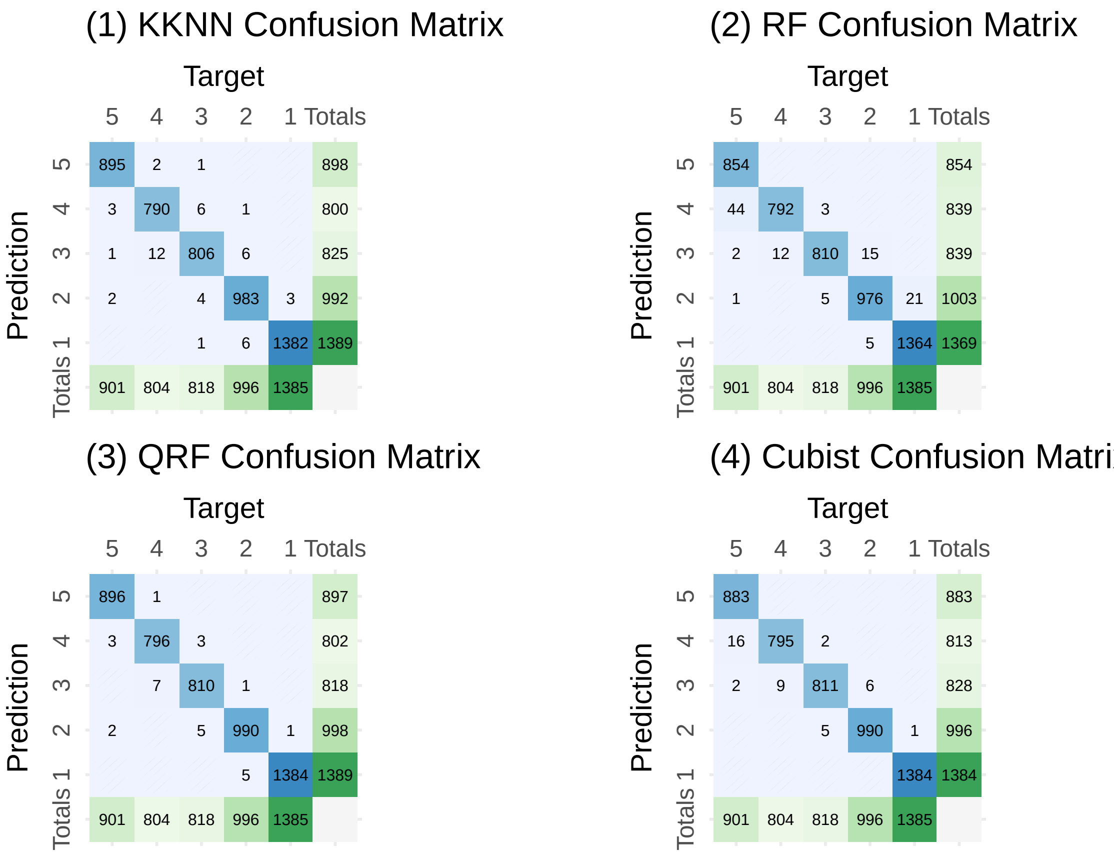

Figure 3 Confusion Matrices for Top 4 Performing ML Algorithms: (1) K-Nearest Neighbors (KKNN), (2) Random Forest (RF), (3) Quantile Random Forest (QRF), and (4) Cubist (cubist). Each graph captures the total counts in the column and row labeled with symbol for summation.

As seen in Figure 3, the FN/FP counts reveal more information about the performance of each algorithm with respect to predicting each barbell execution type (classe equals "A" or 1 for correct execution and "B" or 2 and so forth for incorrect execution). The k-nearest neighbors model generated fewer FN/FP than the random forest model (the worst performer of the top four), but overall the k-nearest neighbors model performed worse than the quantile random forest and the cubist models. Between these latter two models, the quantile random forest generated fewer FP and FN than the cubist model but it does tend to both over and under classify more than the cubist model. The cubist model, however, consistently produced more FN and FP for each classe compared to the quantile random forest model.

Conclusions

Ten different ML algorithms were tested and compared to find the best predictor of activity recognition of a correct barbell curl (represented in the HAR dataset by classe equal to "A" or 1). When conducting the data analysis, the quantile random forest (QRF), cubist (cubist), k-nearest neighbors (KKNN), 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 four models, there are slight differences in their performance metrics but greater difference in their error metrics. The differences in TP/TN and FN/FP values between these two algorithms are significant and the quantile random forest (QRF) model performed better than other top performers. In conclusion, the quantile random forest (QRF) model performs best and was chosen 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 from John Hopkins University 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.