FitBitProject

HAL

December 28, 2016

Executive Summary

I explore the performance of workouts in the HAR data set, and produce a model to predict the performance grade given a set of variables. Exploratory analysis does not show any obvious trends, making machine learning a good approach. I filter the number of variables based on the available data in the test set, and the definitions of the performance grades. With these variables, I create and test four machine learning models: rf, gbm, rpart, and lda. The rf and gbm models have the best accuracy with 97% and 98% respectively. These high scores show the value of machine learning algorithms for forcasting outcomes.

Setup

I downloaded the data from the project website, and read the training and test csv files.

Exploratory Analysis

I performed a quick exploratory analysis to understand the distribution of performance grades and the distribution of grades by person. The tables show a disproportionate number of A's and significant variety in performance by person. This suggests that a large, random set of training data should be used to avoid skew, and the user is an important variable to consider.

```
table(training$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

```
table(training$classe, training$user_name)
```

```
##
##     adelmo carlitos charles eurico jeremy pedro
##   A   1165      834     899    865   1177   640
##   B    776      690     745    592    489   505
##   C    750      493     539    489    652   499
##   D    515      486     642    582    522   469
##   E    686      609     711    542    562   497
```

Understanding the Data

There are too many variables to include in the machine learning algorithms, and by assessing the performance grade definitions, it is clear that some variables can be removed. First, the test set does not include any of the summary variables: average, standard deviation, skeweness, variance, minimum, maximum, amplitude, and kurtosis. These variables should be removed because they cannot be used to predict the outcome in the testing set. I also remove extra information about time: X, new-window, and cvtd-timestamp.

```
filter <- grep("kurtosis|avg|stddev|var|min|max|skewness|amplitude|X|new_window|cvtd_timestamp", names(
cleanDat <- training[,-filter]
```

It is also important to look at the definition of the performance grades because they highlight the kinds of variables that would be useful for a prediction. The definitions are as follows (http://groupware.les.inf.puc-rio.br/har).

Class A: exactly according to the specification Class B: throwing the elbows to the front Class C: lifting the dumbbell only halfway Class D: lowering the dumbbell only halfway Class E: throwing the hips to the front

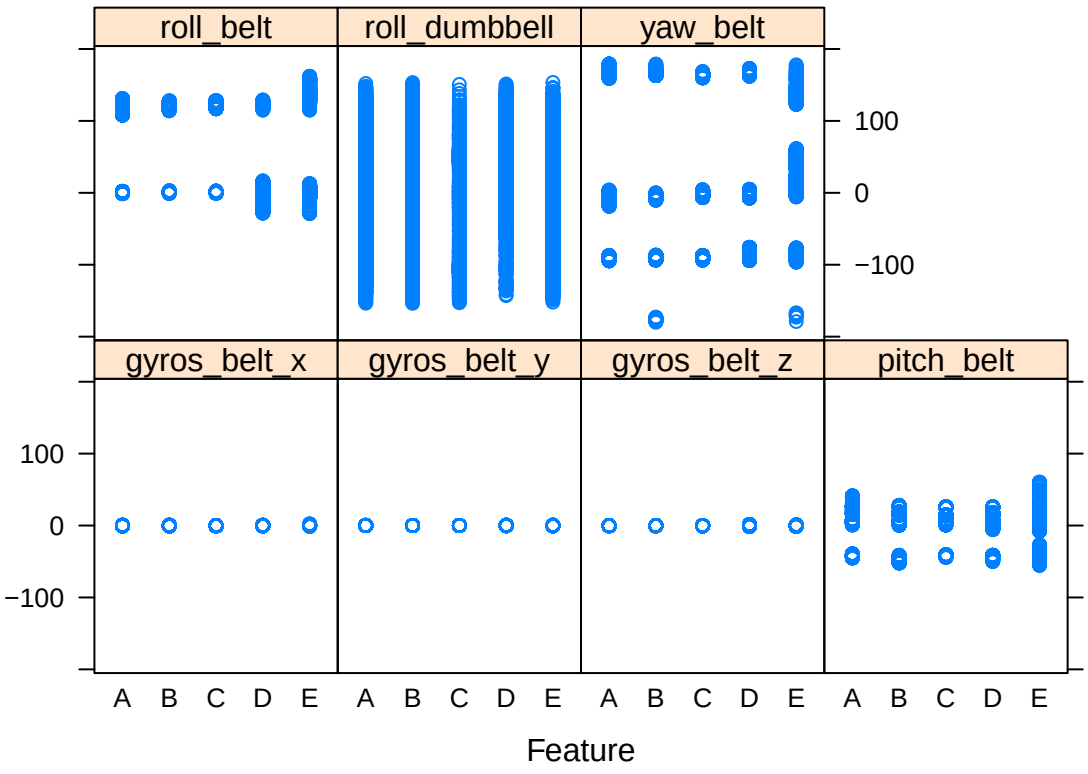
Class B highlights the use of forearms, Class C and D highlight the dumbell, and Class E highlights the use of belt. The definition of Class A does not highlight any particular feature, but perhaps, we can reach this classification by process of elimination. Notice, that classes B through E describe positions, not acceleration or magnet values. For this reason, I filter the variable set further to reflect the scope of the performance grade definitions.

```
filter2 <- names(cleanDat)[grep("forearm|dumbbell|belt" , names(cleanDat))]
filter2 <- filter2[-grep("magnet|accel" , filter2)]
filtDat <- cleanDat[, c("user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", filter2, "classe")]
```

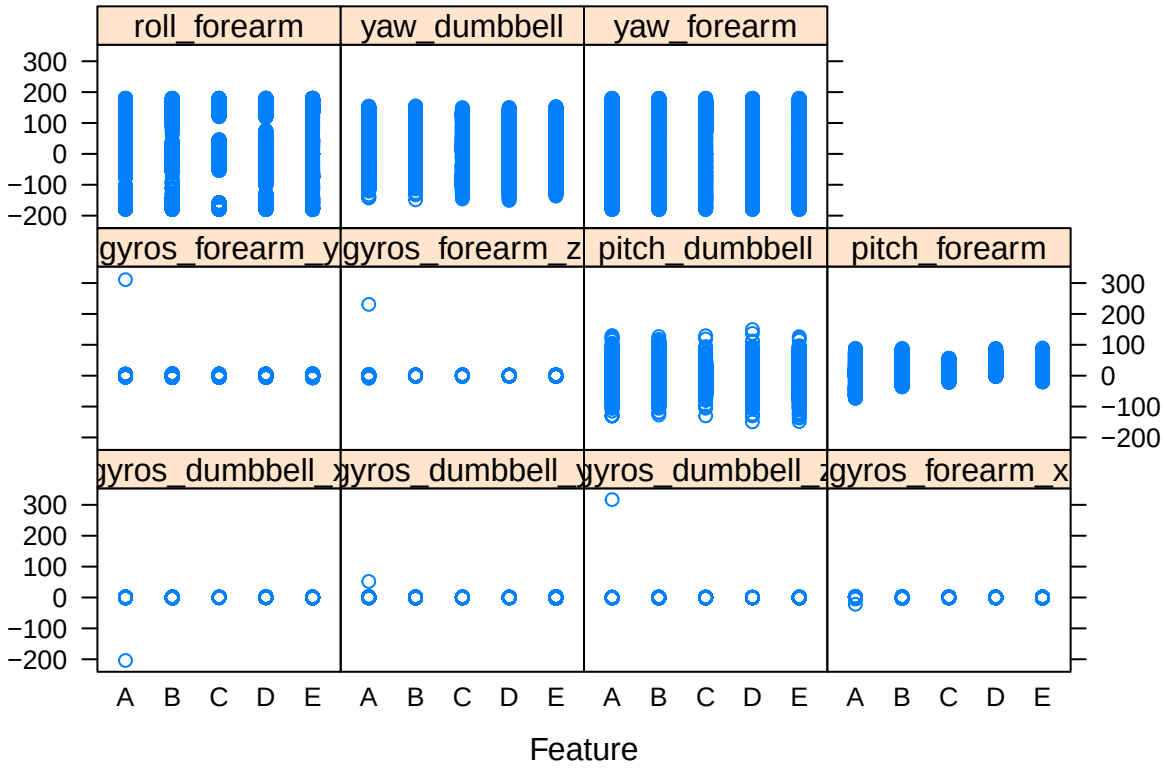
Machine Learning

I did a feature plot of each of the variables and did not spot any obvious trends. Machine learning offers a good approach for finding subtle trends in the data, so I look at four models described here.

```
featurePlot(filtDat[4:10], y=filtDat$classe)
```



```
featurePlot(filtDat[11:21], y=filtDat$classe)
```



Machine Learning Setup

First, I determined the appropriate training set size. We cannot run the data on the entire training set because it is too large for some algorithms. I determined that a set with 1,500 rows would be an appropriate size for the rf, gbm, rpart, and lda algorithms since they completed in a reasonable amount of time. I chose these algorithms because they cover a variety of approaches.

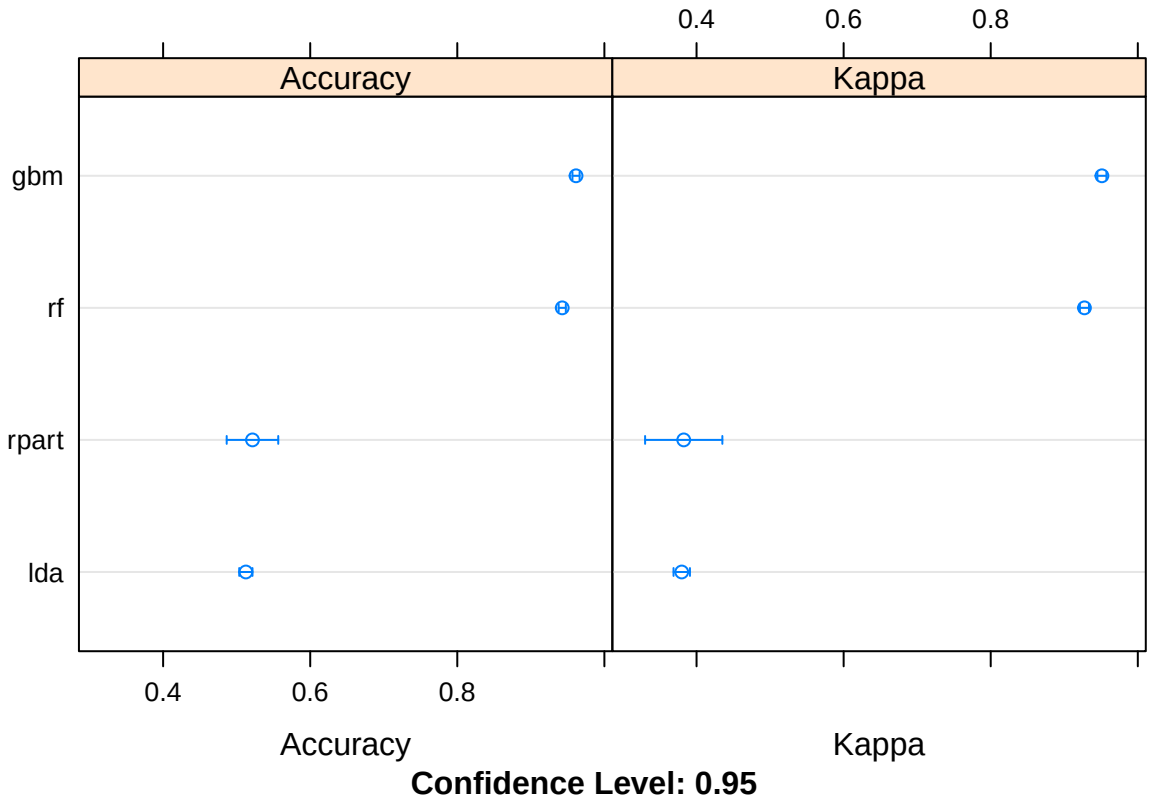
Cross Validation

I cannot simply take the first 1,500 rows of the filtered data because the data is sorted by timestamp, grouping performance measures in contiguous rows. This kind of sample would skew the training set, leading to poor model fits. Instead, I randomize the order of the filtered data, and take the first 1,500 rows of this randomized set.

Results

I resampled the data to test the accuracy of the predictions, and show that the gbm and rf models are the best models with 98% and 97% accuracy respectively. The data was resampled 25 times to look for any prediction bias. With these high accuracy values, these models should perform well in the test set.

```
results <- resamples(models)
dotplot(results)
```



I also include the corrolation values of the models because future work might explore stacking models for even higher accuracy. Given a 98% accuracy rate, I will not pursue that here. It is interesting that the gbm and rf models have high corrolation, but they both have high accuracy values, so their corrolation might be realted to accurate predictions.

```
modelCor(results)#most low except rf and gbm, but this makes sense with high accuracy
```

```
##               rf       gbm       rpart         lda
## rf    1.00000000 0.6090978  0.21966895  0.02530962
## gbm   0.60909779 1.0000000  0.16647742  0.05382350
## rpart 0.21966895 0.1664774  1.00000000 -0.07811271
## lda   0.02530962 0.0538235 -0.07811271  1.00000000
```

Error

The out of sample error for the gbm and rf models should be less than 3% given the validation results. It is important to test the models on data that is not part of the initial training data because it can illustrate a bias. By creating 25 different samples, independent of the initial training set, I mimic the test set, which allows for a good estimate of error.

Conclusion

The gbm model has the best accuracy in the validation set, so I will use this model for the 20 question quiz. By using cross-validation, I can be confident that the accuracy values in the validation sets will be similar to the test set. Future work might explore stacking models to improve the accuracy even more, but I will not

pursue that here. Overall, the high the accuracy scores for the gbm and rf models demonstrate the power of machine learning algorithms.