CDW Practical Machine Learning Project

Charles Warden

11/28/2019

Summary

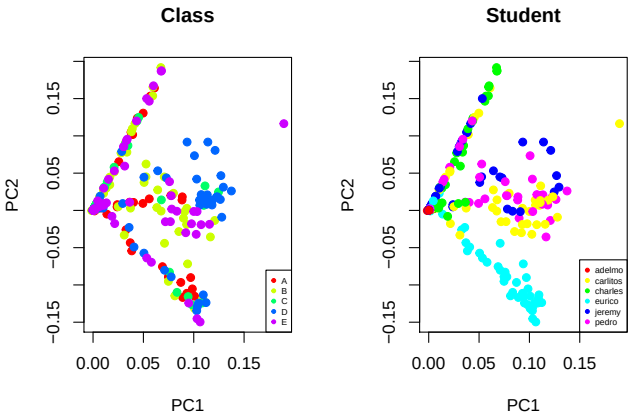
I believe these are the most important messages from this project:

- A split of training data into 60% training and 40% test data gave more realistic estimates of accuracy than cross-validation (and this may/probably be true for other datasets as well)
- For this project (and this analysis strategy), a "good" predictive method often has an accuracy closer to 70% (rather than 100%) in an independent test dataset. However, even 50% accuracy is more than you expect by chance, with 5 categories.

Exploratory Analysis

This was not specifically requested, but I wanted to get a feel for the data before starting analysis.

Figure 1: PCA Plots with Missing Values Omitted



There were 38 variables for each of the 'belt', 'forearm', 'arm', and 'dumbbell' measurements (for a total of 152 maximum possible predictor variables). The starting number of measurement was measurement count (each of which has more than one predictor to build a model from) was 406.

From this, I have the following observations:

1) There is noticable variation from the user (on PC2, we see this most clearly for Eurico)

2) I would guess that I need to *i)* filter features that better explain 'classe' class (over 'user_name' for student) and/or *ii)* include the individual in the prediction model.

Cross-Validation

In the `caret` package, I used the *trainControl()* function to perform cross-validation. This should be done before feature selection (and creating a new model), but I am not actually sure how this is implemented in this package.

I think this may not be the best solution, but I will describe estimates with a 60-40 train-test dataset later.

So, for all 3 models below, you should assume that I filled in the *trControl* parameter and performed 10-fold cross-validation. Random Forest predictions were made with "method ="rf", Boosting predictions were made with method ="gbm", and Linear Discriminant Analysis (LDA) predictions were made with method ="lda".

Model Testing

While there are statistical tests that can be used for feature selection, I will try to focus on what I have learned in this class (using the "Variable Importance" measures, calculated by the varImp() function in the `caret` package for Random Forest and Boosting with "gbm", as well as the maximum coefficient from `abs(modLDA$finalModel$scaling)` for Linear Discriminant Analysis).

In other words, I first tested a few of the methods that we discussed in this class. I report the ***training*** accuracy below, and I will then test re-creating the PCA plots with a filtered set of variables (as well as re-calculating accuracy).

Additionally, I got an error message related to low variance for some predictors. So, I reduced the starting set of predictors from **152** to **146** (by requiring variance > 1e-7). I also set the random seed to 0. However, I also then decided to skip regular linear regression and LASSO regression (since I am trying to predict a categorical variable).

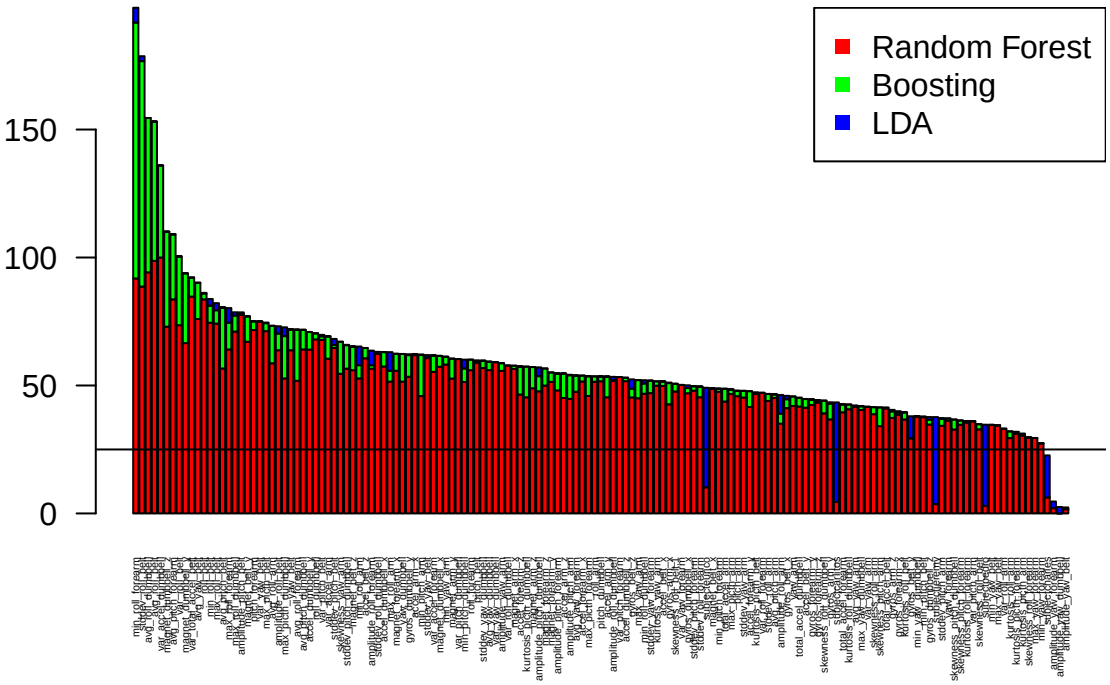
1) `caret` Random Forest (***Estimated* 100.0% Accuracy**, method = "rf")

2) `caret` Boosting (***Estimated* 100.0% Accuracy**, method = "gbm")

3) `caret` Linear Discriminant Analysis (***Estimated* 77.3% Accuracy**, method = "lda" –> previously estimated to have *96.1% Accuracy* with more features)

Figure 2: Variable Importance Measures

```
## [1] 406 146
```

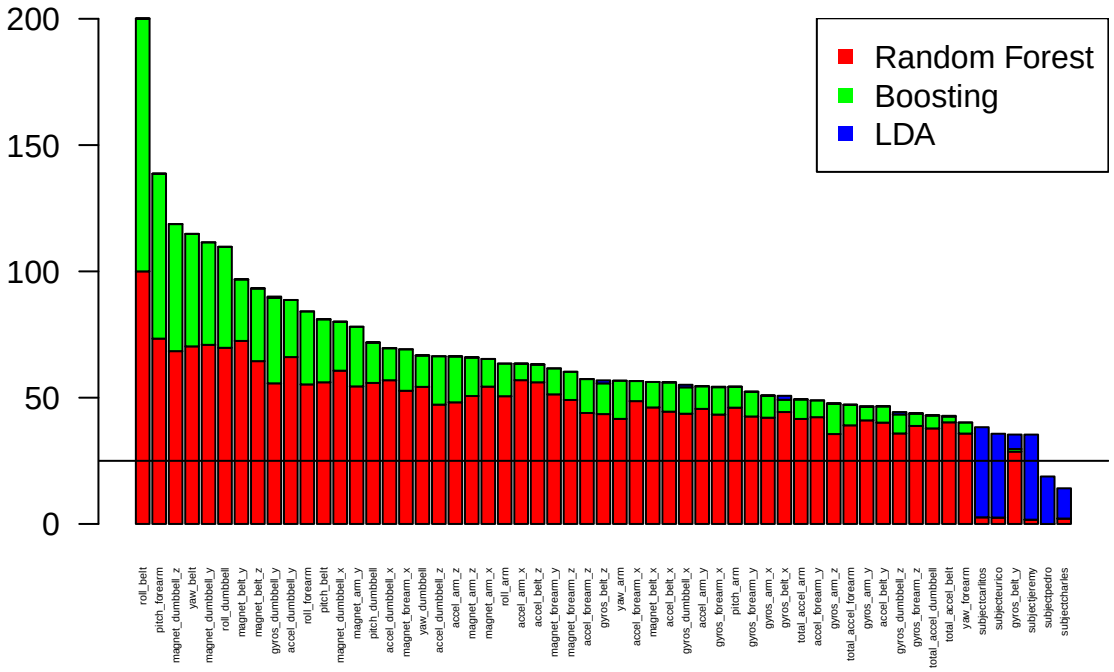


The plot above arguably doesn't show as much benefit to feature selection (but I believe there was an earlier plot with more features where the threshold of a sum of 25 was more meaningful, even though the units being summed are different for each method).

I also suppressed warnings when training the LDA model, using the strategy described here.

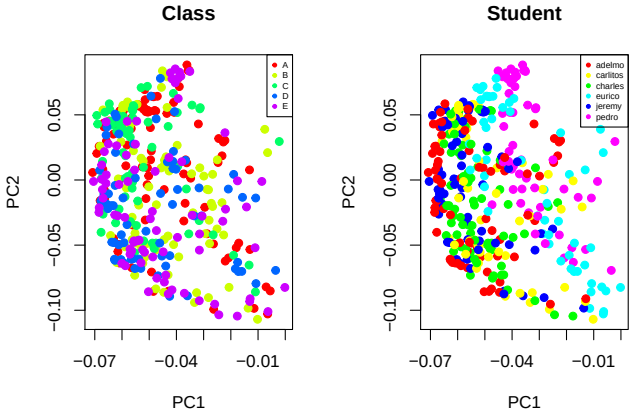
However, if you filter more variables, this appears to make the feature selection even less relevant (and I have noted the considerable loss of LDA accuracy above):

Figure 3: Variable Importance Measures (without missing values in the TEST data set)



If I re-create the PCA plots with the filtered set of features, it now looks like this:

Figure 4: PCA with Filtered Features



The student effect still seems larger than the class effect, but I think it was worth seeing if this might have provided a visualiziation to show benefit to feature selection. If the drop in LDA accuracy was accurate, **maybe this is actually shows *little or no benefit* to the additional variable filtering** (although you have to have features present in the test data, or impute missing values).

Additionally, there were some predictors with missing values in the test dataset. So, if I filter the variables for those present in the test dataset, then the starting number of predictors becomes **53** (including the categorical "user_name" for the student name). Interestingly, I have to be careful about the order - if I filter the missing test samples first, then I keep a lot more training samples. If I don't do that, then the run-time considerably increases. Plus, this means we are theoretically expecting a model with high estimated accuracy with ~2% of the measurements (406/19622), **with a considerably shorter run-time**.

If I then use the filtered set of 53 variables (condensing all "subject" variables into 1 for "user_name"), these are the estimated training accuracies:

1) `caret` Random Forest (***Estimated* 100% Accuracy**, method = "rf")

2) `caret` Boosting (***Estimated* 100% Accuracy**, method = "gbm")

3) `caret` Linear Discriminant Analysis (***Estimated* 75.1% Accuracy**, method = "lda")

Since one model had an estimated accuracy of 100%, I decided to skip creating a combined predictor.

Strictly speaking, I am assuming estimated accuracy will decrease on the training set (when I reduce the number of features). However, if I simplify the model, I hope the accuracy in independent validation data sets can increase. However, both estimated accuracies with cross-validation in the training set were similar.

Course Project Prediction Quiz (Preliminary Result)

I noticed that the final quiz is a grade for accuracy of the model. I understand that this is supposed to be a dataset that you cannot test prior to locking-down a model (to avoid over-fitting, and reduced reproducilbity in truely independent datasets). However, *I would also usually consider a model with 100% accuracy to be suspicious* (and I was therefore expecting to lose points in that section).

If I don't pick a single model, I can use the output from 3 models to compare predictions (and assess the accuracy in the test dataset), then the predictions are accurate for 7 / 8 measurements (out-of-sample error at **87.5%**, *but* 12 samples would have unknown samples and *60%* of samples with uncertain assignments may or may not be acceptable)

If I fill in all the results for a given method I have the following results:

1) `caret` Random Forest (**75% Accuracy (15/20)**, method = "rf")

2) `caret` Boosting (**80% Accuracy (16/20)**, method = "gbm")

3) `caret` Linear Discriminant Analysis (**50% Accuracy (10/20)**, method = "lda")

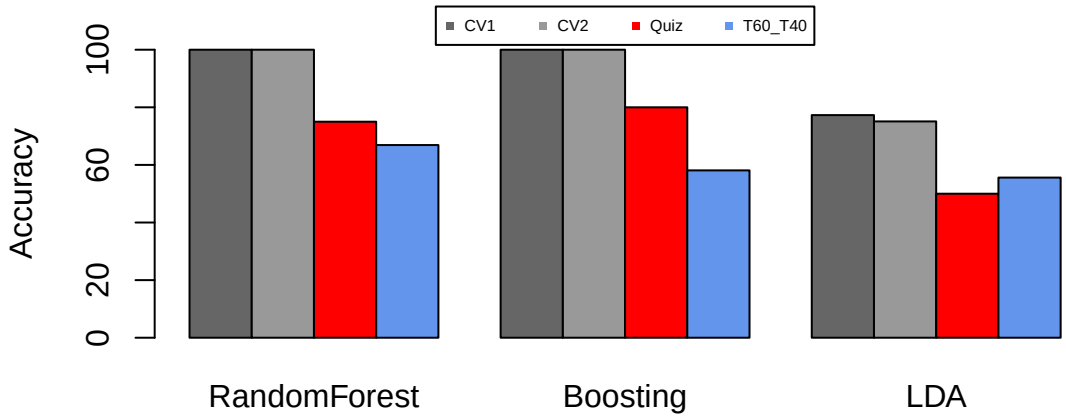
Even though I was able to figure out the status for **18/20** measurements (being right for at least one model), I was really supposed to just pick one method. So, I uploaded the set of answers with a **majority vote** (or leaving the prediction blank), which resulted in a grade of 80% (for **16/20** measurements). So, in the interests of honesty, I did not continue re-taking the quiz to get a higher percentage in the course.

To be fair, it is a bit of a stretch to connect the stategies. However, this is *might* be similar to what I believe needs to be done for RNA-Seq gene expression (based upon personal experience, as well as benchmarks calculated here), if you could view all models and use personal judgement from experience when something was ambiguous (or didn't seem right).

Either way, the estimated accuracy was considerably higher than the true accuracy. So, my point about the considerably reduced run-time may not be valid. **However, I was right to question an estimated accuracy of 100%**.

Additional Estimation with 60-40 Training-Test Dataset

Figure 5: Summary of Accuracy



CV1 is cross validation with all features (among samples without any missing values). **CV2** is cross-validation with a filtered set of features. The "**Quiz**" accuracy is the separate set of 20 measurements used for the quiz grade, and the "**T60_T40**" is the accuracy estimated on the 40% test samples from the 60% training samples (all in the "training" dataset, added for this section).

The order is perhaps a little confusing in that a training sample estimation comes before the separate "quiz" test set (since I want to re-emphasize the predicted accuracy from cross validation was **too high**). So, the red bar is the more realistic accuracy, and I was testing is the 60-40 split could yield a more accurate estimation. **This matches what is described in the lecture, in terms of a 60-40 split being a preferable option to cross-validation**. I believe this also matches my experience in terms of preferring validation in large independent cohorts (*rather than with cross-validation*).

Discussion of Limitations / Errors

From this project, I would expect that individual variability can introduce a considerable challenge. Also, I think having a test set with 20 measurements may have also not been ideal (since I would expect more variability in accuracy estimates with smaller sample sizes).

I have a Fitbit, and this matches my own experience with limitations in the predictive power. For example, it was able to tell 1 time that I was on the elliptical, but it never recognized my spin class as "exercise" (although I could see noticable increases in my heart rate). The accuracy of the model for sleep also seemed to considerably vary over time (which I guessed was due to defining different models, but I can't really say that for certain).

If it is not always safe to assume newer models will work better (which I would say roughly matches my experience), then this could be very important in terms of saving previous models and providing semi-automated results (where the users can choose earlier models, if they think something looks very wrong with the most recent model provided by a company).

Formatting requirements

As requested in the "Reproducibility" section of the assignment, I am not posting the R markdown code. This is somewhat contradictory to the requirement of "Github repo with your R markdown and compiled HTML file describing your analysis", but I hope the HTML alone is OK (otherwise, the R markdown includes the code). I used this discussion to learn more about creating a link to view a formated webpage from GitHub (rather than the GitHub HTML source code).

Based upon an compiled Word document, this report has 1,780 words.