Pre Live Session Unit 2 Assignment

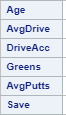
In golf there is an expression, drive for show put for dough ( dough =money). This data set contains average winnings for a given year of profession golfers along with additional performance statistics such as driving distance and accuracy, average number of puts per hole, as well as others including some with variable names that are only labled arbitrarily. The idea here is that although it is quite fun and impressive to watch some one hit a golf ball very hard and far, it actually matters more when they are closer to the whole and chipping and putting. Can we build a model to predict players average winnings given their various performance metrics?

For this assignment, we are going to play around a little bit with glmselect to get a better feel for issues when using metrics such as R-squared in terms of reporting predictive ability of a model and the idea of overfitting. Using the data set and sas code run the models and generate the output. There are 5 total models that are run in the code.

1. Examine the glmselect output from the first two proc glmselect (labeled M1,M2 in the code) calls and compare them in the following way.
   1. What is different between the two OLS models in terms of the predictors? (note we have tricked glmselct in doing OLS by specifying Forward feature selection with no stopping criterion)

One model has the following predictors only:

M1:



M2:

The other has the V12-V31 predictors included.

* 1. What are the two models R-square values and adjusted R-squared values?

M1:



M2:



* 1. Examine the Fit criteria and ASE plots. In terms of prediction do you think there is much harm in using all of the predictors versus using a feature selection approach to reduce the model down?

The Adjusted R-squared is significantly worse at .58 vs .61 when using all of the parameters vs using a reduced model. Also, the average squared error is lower with a reduced model for both train and test:

M2: Reduced M1:

Improvement to model can be gained by variable reduction given the results.

1. Compare the second and third proc glmselct calls (M2, M3). These both have the same predictors but one is OLS and the other is using LASSO feature selection using cross validation.
   1. Note the R-squared and Adjusted Rsquared and compare them.

M2:



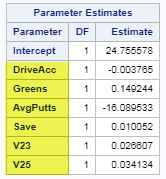
M3:



Interesting M2 explains more of the variability in the model than M3 does as both R-square and adj R-sq is better for M2.

* 1. What variables are included using the LASSO as a feature selection technique?

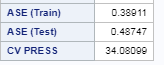
The highlighted variables are the variables included in the LASSO technique:



* 1. Suppose now that I told you that all of the predictors with generic names are just a bunch of random numbers, how does that piece of information potentially change your feeling on whether it matters or not to do feature selection.

LOL! Yes, there will be instances where we have a bunch of garbage data that are not good predictors of the target variable. I think it is really important to use some type of model reduction technique to get rid of the noise.

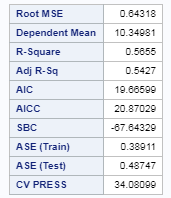
1. Compare the fourth and fifth glmselect calls. These models include interaction terms so the model is even more complex and the potential for overfitting becomes even greater.
   1. In model 5, examine the the CVpress fit criterion panel and compare it to the ASE plot for the test set. Does the CV fit panel mimick the ASE test performance pretty well?



No, not really. CVPress is a lot lower than ASE test. Suspect over fit here, but this was kind of suggested in the problem too.

* 1. In model 5 that uses the CV approach for feature selection, if we have used Adj-Rsquared rather than CV press, how good would you feel about the predictions you made with that particular model?

The adj R-sq is .5427 which isn’t that great. I wouldn’t put this as a predictive model in production. Even if it was, it would be difficult to trust a model on adj r square alone as there could be over fit in the model.



Bonus/Critical thinking: When comparing ASE plots of OLS and LASSO from our given code, you may have noticed that OLS seems to yield smaller test error values than LASSO. That may seem contradictory. Why do you think this is happening and why the actual values of the ASE for the OLS and LASSO models we ran are not directly comparable?