Patrick Keener
SID: 1385832
DSC 423 – Assignment 4

Assignment 4

Honor Statement: I have completed this work independently. The solutions given are entirely my own work.

Question 1 – Regularization

Using the PISA dataset, build models using Ridge and LASSO regression then compare the two.

Ridge Regression

Discuss ridge trace plot & how it handles multicollinearity

The Ridge Regression model has the following form:

Equation 1 Ridge Regression

$$\boxed{\min_{\beta}\left[\sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ji}\right)^2 - \lambda\sum_{j=1}^{p}\left|\beta_j\right|^2\right]}$$

The term on the right, $\lambda\sum_{j=1}^{p}\left|\beta_j\right|^2$, serves as a counter balance to the term on the left. There are three important pieces of this equation:

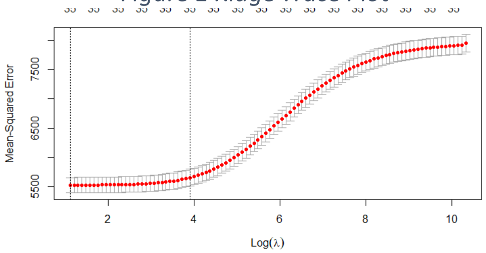
- The term on the right is *subtracted* from the term on the left,
- Both terms share betas,
- The Lambda function scales the effect of the term on the right.

The implication of the first two points is that the term on the right serves as a penalty for the term on the left. When betas increase in magnitude on the left, they also increase in magnitude on the right which increases the penalty; this mechanism serves to keep betas low.

Another implication of this function is how it impacts collinear terms. Collinear terms are variables that behave similarly to each other. In an OLS regression, collinear terms distort betas. For example: if the combined beta of two perfectly collinear terms is 10 then it doesn't matter if both betas are 5 or one beta is 1 and the other 9. Due to the penalty introduced by the term on the right, collinear variables will settle at the point where their weighted impact is maximized; in the earlier example it would force both betas to 5.

The impact of the penalty term λ is scalable. The term lambda (λ) controls how large of a penalty the term on the right gives. A lambda of .5 reduces the penalty by half, which a lambda of 2 will double the penalty.

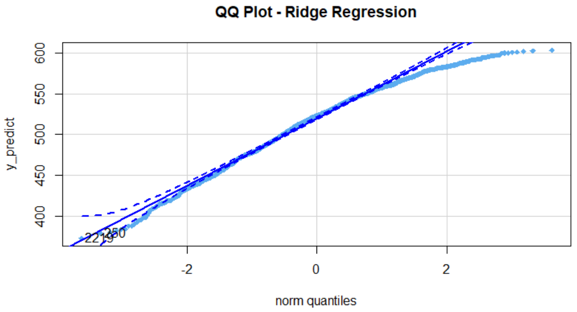
Figure 1 Ridge Trace Plot



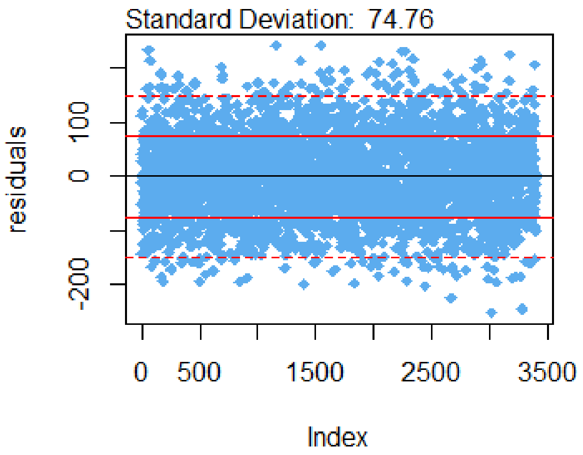
The trace plot's x-axis is the log of lambda and the x-axis is the mean-squared error. Along the top are the number of variables in the model (for Ridge Regression these will not change since variables are not removed).

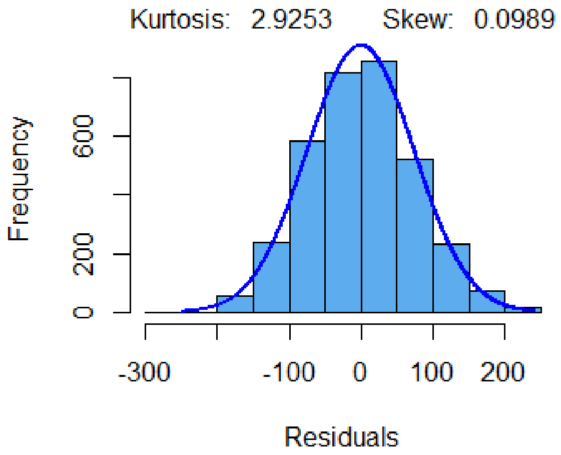
The dotted line represents the point with the smallest effective lambda. In Figure 1 this value is 3.1, however since it's a log the actual lambda is approximately 21.3.

The QQ plot (right) is seen to curve off the line at either end, implying a left skewness in the data. Possible transforms to correct this include square root and log transformations, which is consistent with the transformations done in assignment 3 for the OLS model.



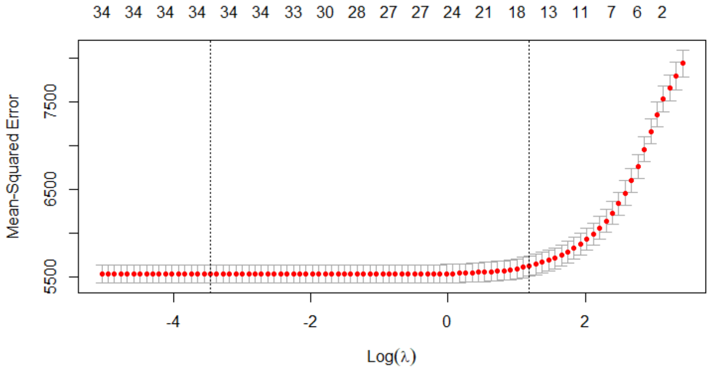
The residual plots from the ridge regression (below) appear to be normally distributed: the skew is .099 and kurtosis is 2.95, which is very close to a standard distribution. The residual chart on the left shows the 1 and 2 standard deviation lines (solid and dotted red lines, respectively). Additionally, there do not appear to be any patterns in the residual charts, so additional transformations do not appear necessary, though this is expected since the number of categorical variables are high and all numerical variables have already been transformed for normality.





LASSO Regression

Figure 2 LASSO Trace Plot



LASSO regression is similar to Ridge regression in that it biases betas towards 0, however it also has the feature of removing features entirely.

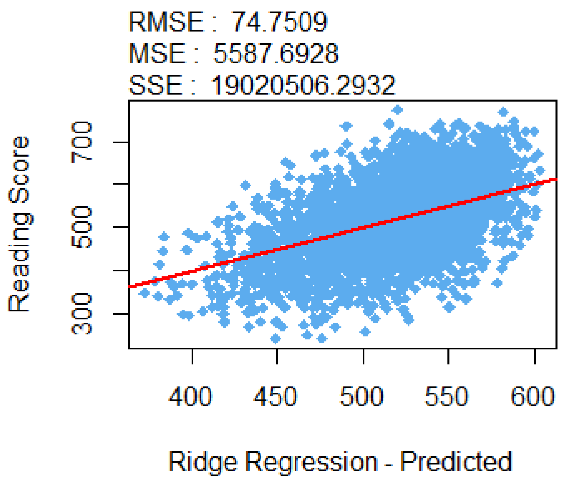
The feature space used consisted of 35 variables (starting set plus several engineered), and LASSO reduced the number of variables down to 17, as shown on the trace plot to the right.

LASSO reduced betas to 0 due to the nature of the feature space. The ridge regression penalty term result space is *squared*, and therefore it is curved; it is unlikely for a term to be reduced to *precisely* 0, even though they may get small. On the other hand, the LASSO penalty term space is *summed*, which leads to edges within the solution space, and those edges correspond to 0s at the inflection points. As the lambdas for LASSO become larger, more of these points are chosen and the betas for more variables are reduced to 0 and therefore ‘deselected’ from the model.

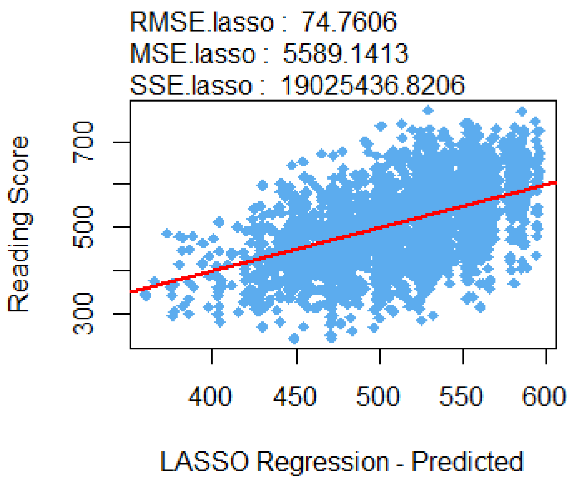
Conclusions

The LASSO and Ridge models are quite different in construction yet perform remarkably similar. The trace plots show very different patterns- the ridge trace plot is sigmoidal in nature, whereas the LASSO trace plot is an exponential curve. Further, the LASSO model reduces dimensions from 35 down to 17, whereas the ridge model does not reduce dimensionality at all, which leads to a relatively parsimonious model in the case of LASSO.

Interestingly, the errors are very similar for both models despite the fact that LASSO has fewer variables, although it should be noted that the OLS model built in assignment 3 has slightly superior outputs to both of them (RMSE of 73.1211, MSE of 5346.7019, SSE of 18200173.2318).

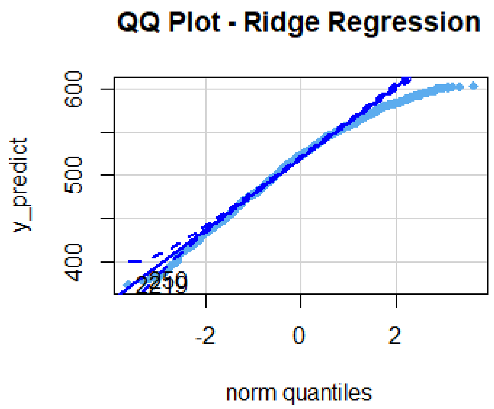


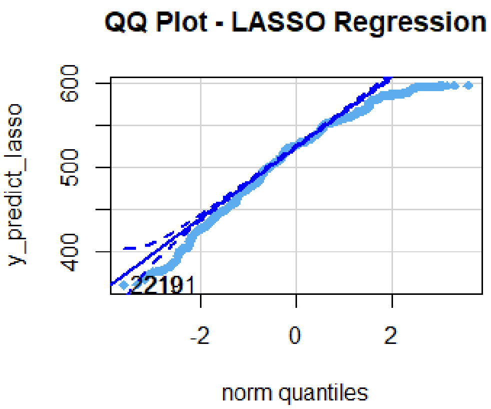
```
RMSE :  74.7509
MSE :  5587.6928
SSE :  19020506.2932
```



```
RMSE.lasso :  74.7606
MSE.lasso :  5589.1413
SSE.lasso :  19025436.8206
```

The QQ plots for both models (below) have some interesting characteristics. Both curve off the 45-degree line at both ends downward, which indicates the data is left skewed and therefore a sqrt or ln transformation may improve accuracy. Given that the OLS model uses a sqrt transformation implies that it may be beneficial to do that here as well. Overall, even though the models perform similarly the LASSO model would be a better choice in practice. This model has around half of the variables so it should be significantly more efficient to run, especially on large datasets.



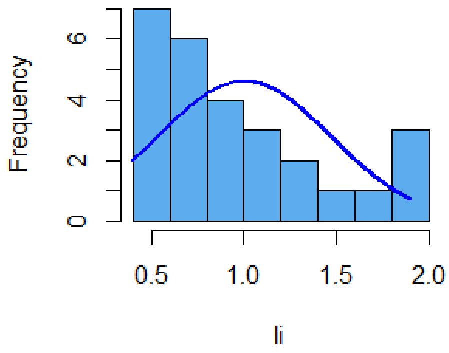


Question 2 – Remission

The Remission dataset contains one binary variable, remiss, and 6 numerical/continuous variables: cell, smear, infil, li, blast, and temp; the goal is to create a model to predict the probability of remission. The comparison between GLM and LM is conducted at the end instead of in between model building and analysis.

Model Building

Feature Selection



Starting with univariate analysis, normalcy of each variable was examined (graph at right). None of the variables could be coerced into a normal distribution, and it did not seem to increase p-values in testing, so no transformations were applied.

Models and Variables were evaluated first on their t-tests and AIC, then on their Areas Under the Curve ("AUC"): AUSEC (Sensitivity), AUACC (Accuracy), AUSPC (Specificity), and AUROC (Receiver Operator Characteristics).

Temp was tested for usefulness in prediction: first, as it was given, and second, after the smallest value was subtracted. Subtracting the smallest value had no impact, therefore this line of experimenting was abandoned.

Evaluating each of the variable's predictive abilities on their own showed that only li had a p-value < .05, and while blast had a p-value of .09, its AUCs were very poor relative to li. A combination of the two led to an AUC somewhere in between li and blast on their own, and the t-tests for both jumped into the .4-.5 range; the blast + li model was discarded.

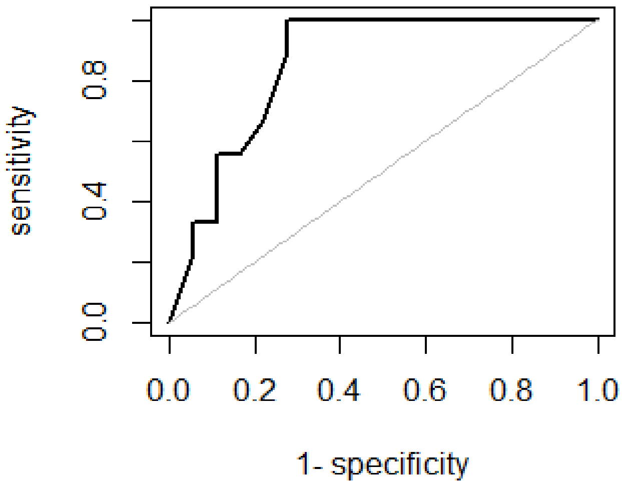
Other variables were tested in varying configurations together and apart, and several second order terms (interaction and polynomial) were tested as well; none had significant values and most caused li to become insignificant.

Finally, a LASSO regression was applied which suggested li to be the only useful variable. The second to last remaining variable was cell, so this was tested in conjunction with li. The p-value was higher than expected (.30), and while it did improve the AIC and other evaluative criteria slightly, it was too far outside of the .05 threshold to be considered for the final model.

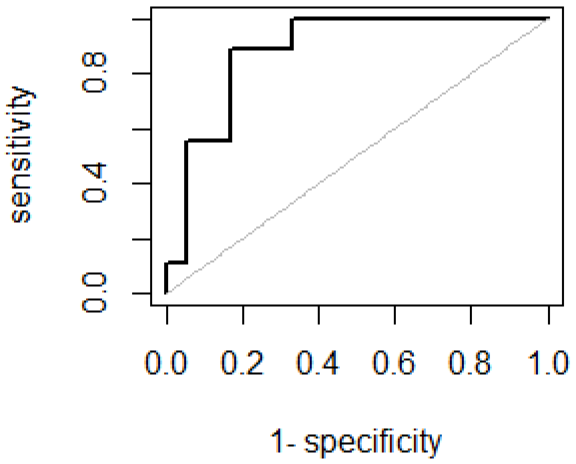
Model Analysis

The final model is very simple: li by itself. It was evaluated on AIC, AUSEC, AUROC, AUACC, AUSPC, and was tested for significance and determined to be the best model using the available predictors.

Measure	Value
AIC	30.073
AUROC	0.8549
AUSEC	0.7695
AUSPC	0.535
AUACC	0.6235



For comparison, the graph and stats for all predictors had slightly higher AUCs, however the AIC was 35.7, vs 30.0 for the model with just li. The minor differences in measurements are likely not worth it considering the added cost of using 6 variables over 1 for very little gain in accuracy.



Measure	Value
AIC	35.751
AUROC	0.8827
AUSEC	0.7901
AUSPC	0.5905
AUACC	0.6571

Patrick Keener
SID: 1385832
DSC 423 – Assignment 4

Comparison of GLM and LM

The command lm stands for Linear Model, and is used to conduct linear regression – simple and multiple regression with the option to use WLS, and is constrained to dependent variables that follow a normal distribution. The glm command stands for Generalized Linear Model and it allows specification of the family of distributions that the variance of the data follows: binomial, gaussian (same as lm()), gamma, inverse gaussian, poisson, quasibinomial (logit), quasipoisson, and quasi (corresponds to several other variance functions), which allows for a much broader range of analysis.

Concretely, the glm function allows use of the glm model which allows us to do logistic regression (on a binomial distribution), whereas the lm function allows use of only the OLS and WLS models which only allow simple and multiple regression (on a normal distribution). Both models output an lm object.