Project 2

Ryan Fischbach

3/30/2021

Abstract

The ability to predict the number of applications to a University is incredibly important to our client, who works with universities and tries to predict the number of applications received by each university every year. A model to help them to predict the number of applications would allow them to help staff the admissions office, estimate how much student housing is needed, and more. In this report, we discuss the process of estimating the number of applications a university will receive using data collected by the client. The steps in the process and comparison of results from different models are discussed.

Part 1: Data Cleaning

To transform this initial dataset into data that can be used to generate a model, the following steps were taken to clean the data.

The uncleaned dataset was loaded, containing 19 variables and 777 observations.

1. First, the private column, indicating if a school is private, was encoded from "Yes/No" to 1/0 to allow for it to be used in the modeling phase.
2. Next, the column including the name of the school was dropped because it provides no information in terms of our ability to predict the number of applications for a new college.
3. Additionally, at the request of our client, the Enroll column was dropped because oftentimes this data is not available from universities and thus can't benefit us in our analysis.
4. Lastly, at the request of our client, a new column was created to show the acceptance rate for universities (Acceptances/Applications) to attempt to normalize this information and the Acceptances column was removed.

Additionally, the dataset was scanned for missing values to uncover if any additional work needed to be done to deal with them. After the previous data cleaning steps, there are no missing values, requiring no action on our part.

After these efforts, the dataset has 17 features with 777 rows.

Part 2: Selection Only

For the first model, the client is requesting that we use a least-squares linear regression model (LSLR), where the response variable is the number of applications received by a university.

For this model, we are finding the coefficients $\hat{\beta}$ minimizing the Residual Sum of Squares:

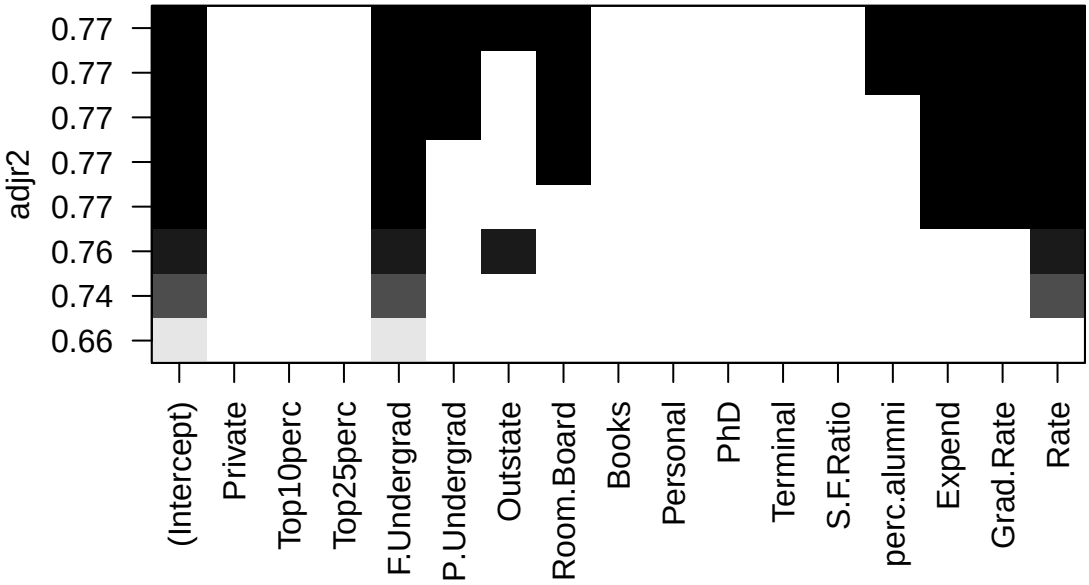
$RSS = (Y - \hat{Y})^T(Y - \hat{Y})$

To choose the best LSLR model for the task, Best Subset Selection (BSS) will be used. Best Subset Selection is a tool that allows the best features to be selected for an LSLR model.

BSS runs in two stages:

1. First, the "best" model according to a metric is determined for one explanatory variable, two explanatory variables, etc.
2. In Stage 2, we compare all of these best models and try to find the one that optimizes the adjusted R^2.

Figure 2.1: Performance of Models in BSS



From our BSS (results are shown in Figure 2.1), multiple different predictive models yield an adjusted R^2 metric of 0.77. However, to pick our final model, we will pick the simplest model that yields a 0.77 adjusted R^2 value. This model contains 4 explanatory variables.

Fitting the final LSLR model after performing Best Subset Selection, we get the regression line:

$\widehat{Applications} = 1,872 + 0.6314F.Undergrad + 0.094Expend + 23.84Grad.Rate - 4922AcceptanceRate$

After fitting our final model, we get a residual standard error of 1,875. This signifies that the residual sum of squares / the residual degrees of freedom is 1,875. The average of all applications across colleges is 3,001, signifying that the standard error of our model is very high. Additionally, looking at the weights of our model, we can see that the coefficients of F.Undergrad and Expend moved towards 0, while the coefficients of Grad.Rate and Acceptance Rate increased in size. This will impact our future predictions, any confidence intervals we generate, and more. Thus, LSLR does not look like the best option. An option that shrinks the estimates of the coefficients might perform better in this context.

To validate this model, 10-fold cross-validation was chosen. To perform 10-fold cross-validation, we create 10 sets of the data, all roughly equal size. Each small dataset is a fold, derived from the original dataset. One fold is used as a test set to validate the model, with the rest being used to train the model. Each combination of training and test sets is used, yielding 10 combinations of cross-validation testing. This method was chosen because of its balance of computational complexity compared to a more rigorous method (like Leave One Out Cross-Validation) and the ability to eliminate randomness (unlike the Validation Approach). With 777 observations, this method was appropriate for this problem context.

Using our final LSLR model, 10-fold cross-validation yielded a Root Mean-Squared Error of 1,900.10. In other words, the average residual for this LSLR model was roughly 1900. Considering that the mean of the number of applications to all universities is 3,001.64, this model should not be used to predict the number of applications a university receives because it will produce, on average, a prediction very far away from the true value.

Part 3: Shrinkage Only

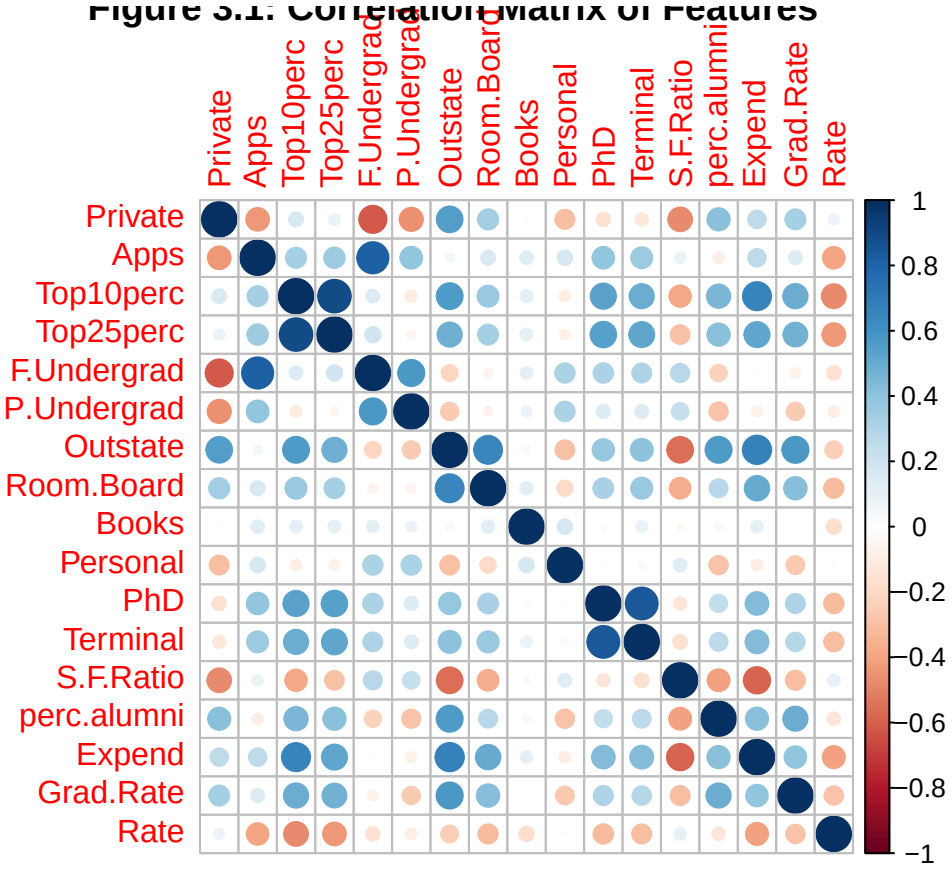
Ridge Regression is a method that adds a penalty term to the RSS, in essence penalizing the model for having large coefficients of explanatory variables. Based on the coefficients for Grad. Rate and Acceptance Rate being very large during LSLR, Ridge has the potential to lower these coefficients to help reduce model variance at the cost of bias.

In technical terms, for Ridge Regression, we choose estimates of $\hat{\beta}$ that minimize:

$$RSS + \lambda \left||\hat{\beta}\right||_2 = (Y - X_D\hat{\beta})^T(Y - X_D\hat{\beta}) + \lambda_{lasso} + \lambda\hat{\beta}^T\hat{\beta}$$

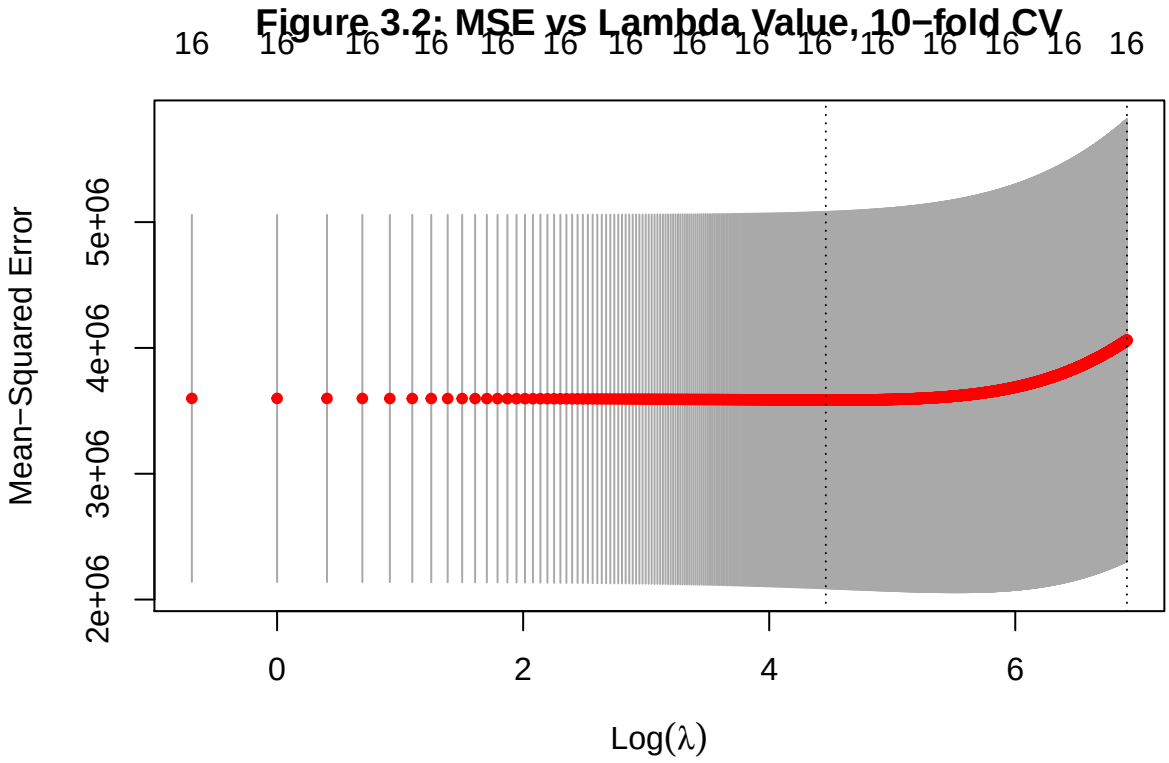
where $\lambda \geq 0$ is a scalar.

This approach comes with pros and cons. By changing the tuning parameters, λ, we can shrink the coefficients and variance of the coefficients. In other words, if we got another sample and used ridge regression again, the coefficients wouldn't change in size. However, this has the opportunity to shift these coefficients bias, or how far our estimates are from the true values.



Additionally, Figure 3.1 specifies that some features have high correlations with each other. Based on this, we should consider ridge regression and judge its performance compared to our first LSLR model.

A ridge regression model was fit and the performance of said model was analyzed using 10 fold cross-validation. We will use 10 fold cross-validation again for the balance of accuracy and speed, in addition to keeping the technique used for model validation the same across all models. To determine the best value of the tuning parameter λ, a sequence was created between 0 and 1,000 by increments of 0.5. At each step in the sequence, 10 fold cross-validation was performed to determine the test MSE associated with that value of $/lambda$. The λ value contributing to the smallest MSE was chosen as the final parameter for the ridge regression model because it achieved the best performance.



As seen above in Figure 3.2, using a variety of λ values and 10 fold cross-validation, we arrived at an optimal λ of 86.5. In other words, this was the value of λ that minimized the Mean Squared error. The Root Mean Squared Error at this optimal value was 1,893.40. This serves as an improvement of 0.3% over the LSLR model previously chosen. Thus, if deciding between LSLR and this Ridge Regression model, we would choose the Ridge Regression model because of its lower RMSE. However, once again given the context of the number of applications to universities having an average of 3001.64, this model does not make sense to use given its high average error.

Our final model takes the form:

$$\widehat{Applications} = 2006.83 - 466.42Private + 8.609Top10perc - 2.134Top25perc + 0.623F.Undergrad - 0.104P.Undergrad + 0.0722Outstate + 0.243Room.Board - 0.163Books - 0.107Personal - 0.781PhD - 7.772Terminal + 6.467S.F.Ratio - 21.573perc.alumni + 0.067Expend + 18.680Grad.Rate - 4532.623AcceptanceRate$$

Table 1: Comparing the Full Model and Shrinkage

	Full.Model	Shrinkage
(Intercept)	2348.2598541	2006.8274265
Private	-347.5626759	-466.4233407
Top10perc	7.8541101	8.6096348
Top25perc	-4.0717941	-2.1345399
F.Undergrad	0.6573792	0.6238441
P.Undergrad	-0.1459922	-0.1043386
Outstate	0.0785571	0.0722781
Room.Board	0.2395974	0.2431598
Books	-0.2109560	-0.1631667
Personal	-0.1280700	-0.1071706
PhD	-0.9071427	-0.7818893

	Full.Model	Shrinkage
Terminal	-9.2067192	-7.7728607
S.F.Ratio	2.6489940	6.4674838
perc.alumni	-21.5665924	-21.5732254
Expend	0.0661854	0.0677508
Grad.Rate	18.1385951	18.6808938
Rate	-4740.8151250	-4532.6234032

Using Table 1 as a reference, it appears that Ridge Regression shrank the coefficients of several explanatory variables, while the coefficients of others increased. Two coefficients that shrank are Terminal and Top25perc. Their coefficients moved from -9.206 to -7.772 and -4.07 to -2.134 respectively. This suggests that these original weights' size did not contribute towards predictive accuracy via a test metric, and thus Ridge Regression shrunk them to minimize its RSS and penalty term.

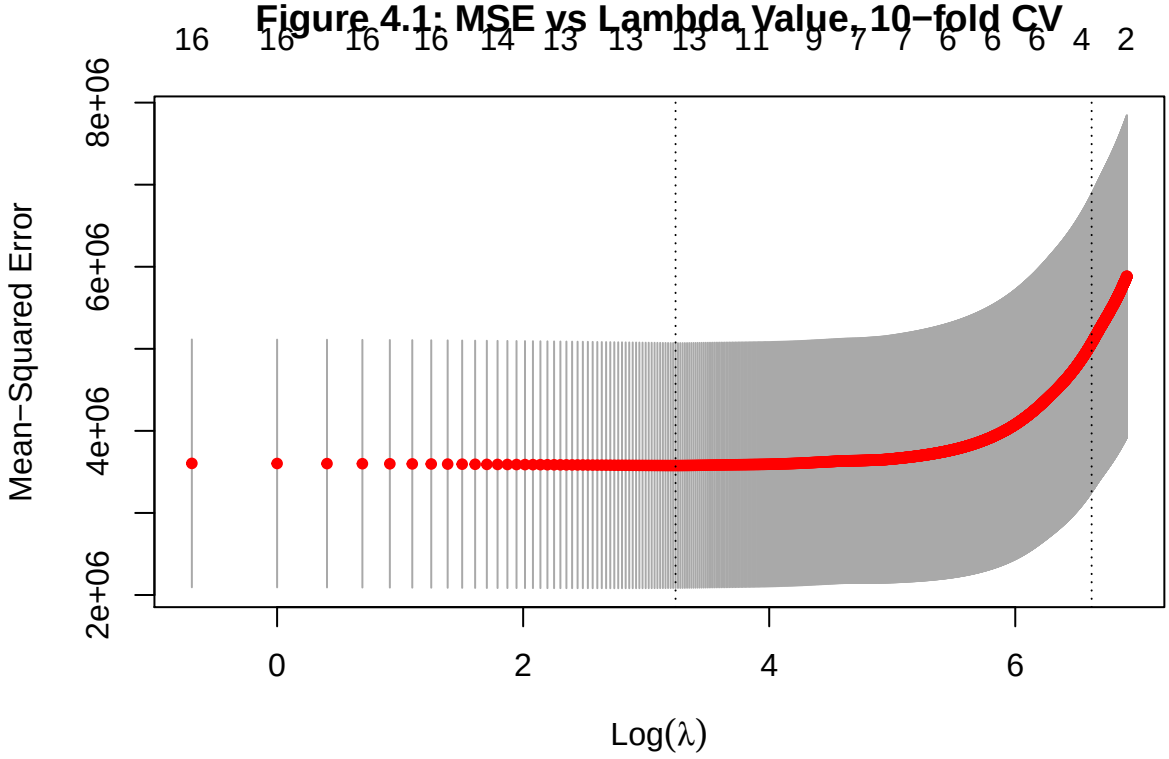
Part 4: Selection and Shrinkage

Lasso Regression is a technique that allows for a combination of selection and shrinkage. In other words, it offers the benefit of being able to choose explanatory variables in the model and reduce their weights to mitigate instability and overfitting. However, often times lasso prefers models with fewer explanatory predictors. In this context, with a high number of explanatory predictors, a selection technique could help eliminate correlated and or useless features for this prediction task.

For Lasso regression, we choose the estimates of $\hat{\beta}$ that minimize the term:

$RSS + \lambda_{lasso} \left||\hat{\beta}_j\right||_1 = (Y - X_D\hat{\beta})^T(Y - X_D\hat{\beta}) + \lambda_{lasso} \sum_{j=1}^{k} \left|\hat{\beta}_j\right|$

where $\lambda_{lasso} \geq 0$ is a scalar.



To determine the optimal value of the tuning parameter λ for our Lasso model, we fit a large amount of models using different *lambda* values and pick the model that minimizes a test metric (see Figure 4.1). We will use 10 fold cross-validation again for the balance of accuracy and speed, in addition to keeping the technique used for model validation the same across all models. After performing 10 fold cross-validation using the tuning parameter λ in the range 0 to 1,000 by increments of 0.5, we find that λ equal to 24.5 minimizes the Root Mean-Squared Error at 1,890 (as shown in Figure 4.1). Based on this performance, out of the LSLR, Ridge, and Lasso models fit so far, we would choose this Lasso Model because it yields the smallest test RMSE.

The final regression line using $\lambda = 24.5$ is: $\widehat{Applications} = 1840 - 164.506Private + 2.412Top10perc + 0.647F.Undergrad - 0.111P.Undergrad + 0.056Outstate + 0.218Room.Board - 0.034Books - 0.091Personal - 4.261Terminal - 17.123perc.alumni + 0.064Expend + 17.047Grad.Rate - 4687.960AcceptanceRate$

Table 2: Comparing the Full Model, Shrinkage, and Lasso

	Full.Model	Shrinkage	Lasso
(Intercept)	2348.2598541	2006.8274265	1840.3807367
Private	-347.5626759	-466.4233407	-164.5061325
Top10perc	7.8541101	8.6096348	2.4126668
Top25perc	-4.0717941	-2.1345399	0.0000000
F.Undergrad	0.6573792	0.6238441	0.6470725
P.Undergrad	-0.1459922	-0.1043386	-0.1116821
Outstate	0.0785571	0.0722781	0.0569717
Room.Board	0.2395974	0.2431598	0.2186829
Books	-0.2109560	-0.1631667	-0.0349367
Personal	-0.1280700	-0.1071706	-0.0911838
PhD	-0.9071427	-0.7818893	0.0000000

	Full.Model	Shrinkage	Lasso
Terminal	-9.2067192	-7.7728607	-4.2610610
S.F.Ratio	2.6489940	6.4674838	0.0000000
perc.alumni	-21.5665924	-21.5732254	-17.1239147
Expend	0.0661854	0.0677508	0.0646154
Grad.Rate	18.1385951	18.6808938	17.0470519
Rate	-4740.8151250	-4532.6234032	-4687.9600306

Part 5: Elastic Net

ElasticNet is a model that combines both Ridge and Lasso regression, resulting in both shrinkage and selection! To do so, we choose the estimates of $\hat{\beta}$ that minimize:

$RSS + \lambda \sum_{j=1}^{k}((1-\alpha)\hat{\beta_j^2} + \alpha \left|\hat{\beta}_j\right|)$

where $\lambda \geq 0$ and $\alpha \geq 0$ are scalars.

Using 10 fold cross-validation, we choose the combination of the tuning parameter λ and α that minimize the RMSE. We used 10 fold cross-validation again for the balance of accuracy and speed, in addition to keeping the technique used for model validation the same across all models. We choose $\alpha = 0.55$ and $\lambda = 63.00$ to yield a minimum RMSE value of 1708.4. 10-fold cross-validation was used in this context to keep the techniques used standard across all models. By keeping the technique standard, this allows us to better judge performance on the same playing field and choose the best model. This RMSE value suggests that on average, our predictions using this ElasticNet model are off by 1708.4 applications. This is a very high average error relative to the mean of all applications 3001.64.

Based on our optimal α value, we can see that this model is closer to Lasso regression. Based on the value we are minimizing above, an $\alpha = 0.55$ signifies we are weighting the penalty term of Lasso 55% of the total and the Ridge penalty term 45%.

Table 3: Comparing the Full Model, Shrinkage, Lasso, and ElasticNet

	Full.Model	Shrinkage	Lasso	Elastic.Net
(Intercept)	2348.2598541	2006.8274265	1840.3807367	2013.5261405
Private	-347.5626759	-466.4233407	-164.5061325	-388.7969602
Top10perc	7.8541101	8.6096348	2.4126668	5.4418794
Top25perc	-4.0717941	-2.1345399	0.0000000	-0.1766801
F.Undergrad	0.6573792	0.6238441	0.6470725	0.6321079
P.Undergrad	-0.1459922	-0.1043386	-0.1116821	-0.1098016
Outstate	0.0785571	0.0722781	0.0569717	0.0682758
Room.Board	0.2395974	0.2431598	0.2186829	0.2376679
Books	-0.2109560	-0.1631667	-0.0349367	-0.1329041
Personal	-0.1280700	-0.1071706	-0.0911838	-0.1059773
PhD	-0.9071427	-0.7818893	0.0000000	0.0000000
Terminal	-9.2067192	-7.7728607	-4.2610610	-7.8097296
S.F.Ratio	2.6489940	6.4674838	0.0000000	2.1292440
perc.alumni	-21.5665924	-21.5732254	-17.1239147	-20.6237824
Expend	0.0661854	0.0677508	0.0646154	0.0670268
Grad.Rate	18.1385951	18.6808938	17.0470519	18.1887265
Rate	-4740.8151250	-4532.6234032	-4687.9600306	-4584.3421323

Looking at the coefficients of the ElasticNet model in Table 3, we can see that both shrinkage and selection

occured. This highlights the benefits of ElasticNet over LSLR, Ridge, and Lasso regression.

Part 6: Conclusion

After assembling a LSLR model using BSS, a Ridge Regression model, a Lasso Regression model, and ElasticNet model, we can compare their performance using the resulting Residual Mean Square Error metric achieved from 10 fold cross-validation. RMSE outlines how far, on average, the model's prediction was from the true value, so we are trying to minimize this metric. Thus, we want to choose the model with the smallest RMSE. Comparing the 4, we achieved RMSE values of 1900.1, 1893.4, 1890, and 1708.4 for LSLR & BSS, Ridge, Lasso, and ElasticNet respectively. Thus, we choose the ElasticNet Model because it minimized the RMSE with respect to our data.

The ElasticNet regression line we chose for the model is as follows: $\widehat{Applications} = 2013.526 - 388.79Private + 5.44Top10perc - 0.176Top25perc + 0.632F.Undergrad - 0.109P.Undergrad + 0.068Outstate + 0.237Room.Board - 0.132Books - 0.105Personal - 7.809Terminal - 20.623perc.alumni + 0.067Expend + 18.188Grad.Rate - 4584.342AcceptanceRate$

Despite choosing the best performing model, the best RMSE was 1708.4 relative to the mean of our target value (the number of applications we are trying to predict) being 3,001.64. This suggests that on average, our model will be ~56.9% off. I would caution use of this model because of the high average error using these regression techniques and this data.