Northeastern University

ALY 6015: Intermediate Analysis

Sanjana Mohile – 002123793

Assignment 4-Regularization

Northeastern University

ALY 6015, CRN 81048

Guided By: Yvonne Leung

Assignment Due Date: May 9th, 2022

Introduction

What is regularization and why is it needed?

While working with linear/multiple regression comprising many features suffers from some problems that include overfitting, multicollinearity, and intensive computational power. It is difficult to build a model with higher accuracy with these problems. To solve these problems, applying regularization becomes very important. We also apply regularization when the model lacks stability and generalization. Ridge and Lasso are two powerful regularization techniques. They work by penalizing the magnitude of coefficients of features along with minimizing the standard error between predicted and actual observations. The main difference between Ridge and Lasso is how they penalize the variables.

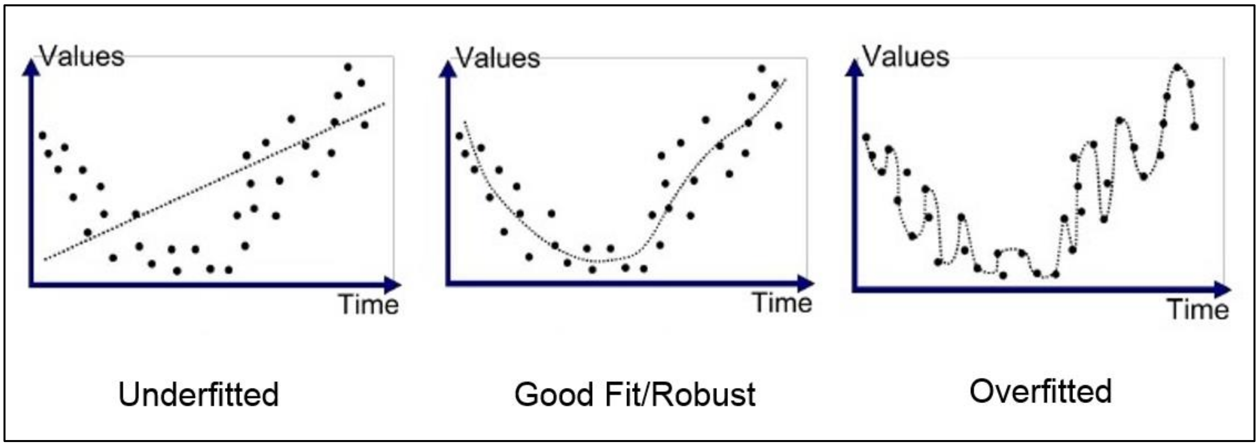
Ridge performs L2 regularization, i.e., it adds a penalty equivalent to the square of the magnitude of coefficients. It makes the model more flexible by increasing the value of Lambda in the formula. Hence, Lambda increases -> flexibility decreases -> bias error increases -> variance decreases. One disadvantage of Ridge is it includes all p predictors in the final model and the coefficients are shrunk towards zero but aren't zero. Below is the formula used to perform Ridge regression.

Objective = RSS + α * (sum of square of coefficients)

Lasso performs L1 regularization, i.e., it adds a penalty equivalent to an absolute value of the magnitude of coefficients. As compared to the Ridge regression, Lasso is more interpretable because it makes the coefficients zero and will make the variables disappear.

As a result, Lasso has fewer variables. Below is the formula used for performing Lasso regression.

Objective = RSS + α * (sum of absolute value of coefficients)



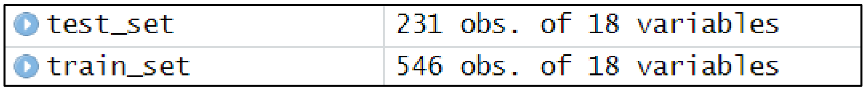
In the above figure, we understand the importance of penalizing the model. If we penalize it too much, there is a possibility that we might underfit the model (the regression line would be passing through only some points and has a high residual error). If we do not penalize the coefficients, there is a possibility that the model might be overfitted (the regression line would be passing through all the points and will have minimum residual error). Hence, finding an appropriate value of Lambda is important.

Exploratory Data Analysis

In the previous assignment, we were provided with the same dataset and there we performed some EDA. On checking for null or missing values we understood that there were none. But we found some outliers in our dataset and in order to get an accurate analysis we remove those outliers. Then, we divided the dataset into public and private to gain insight into what kind of data are we working with.

Q1. Splitting the dataset into train and test subset to train subset.

For performing regression analysis on a dataset, we need to first divide the dataset into train and test subsets. We do that by deciding a ratio of separation. Here, the dataset is divided by a proportion of 70:30 where 70% of the data goes to the training subset and 30% of the data goes to the test subset. The proportion of training is always higher than testing because if there are more observations, the model gets trained more accurately.

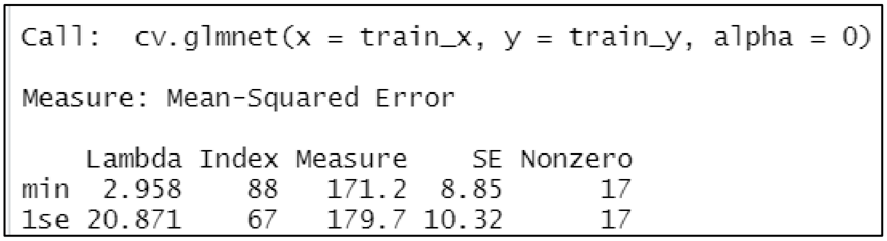


```
test_set      231 obs. of 18 variables
train_set     546 obs. of 18 variables
```

From the above figure, we see that 546 observations out of 777 are in the training subset and 231 observations are in the testing subset.

Q2. Estimating lambda.min and lambda.1se values using cv.glmnet function for ridge model.

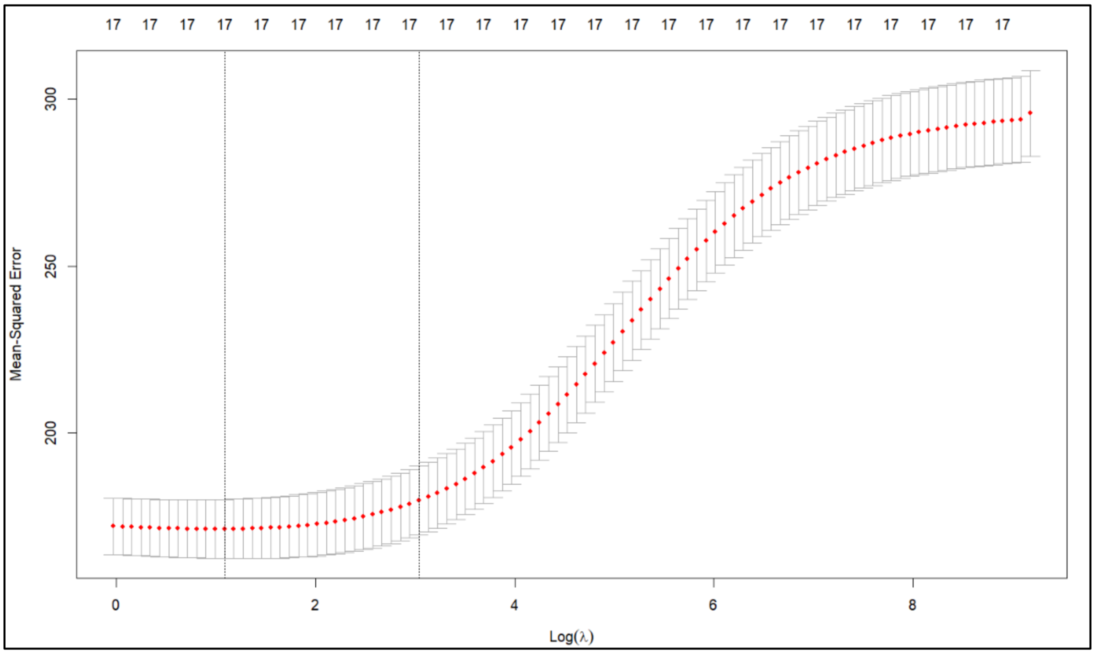
- Glmnet is an in-build function/package in R that allows us to fit generalized linear models by penalizing the alpha value.
- The penalty term lambda is used to regularize the coefficients. Here, ridge regression shrinks the coefficient and as a result, helps in reducing the model complexity and multi-collinearity.
- Lambda.min is the value of lambda that gives a minimum mean of the cross-validation error.
- Lambda. 1se is the maximum value of lambda that produces an error that is one standard error above the minimum.



```
Call:  cv.glmnet(x = train_x, y = train_y, alpha = 0)

Measure: Mean-Squared Error

    Lambda Index Measure    SE Nonzero
min  2.958    88   171.2  8.85      17
1se 20.871    67   179.7 10.32      17
```

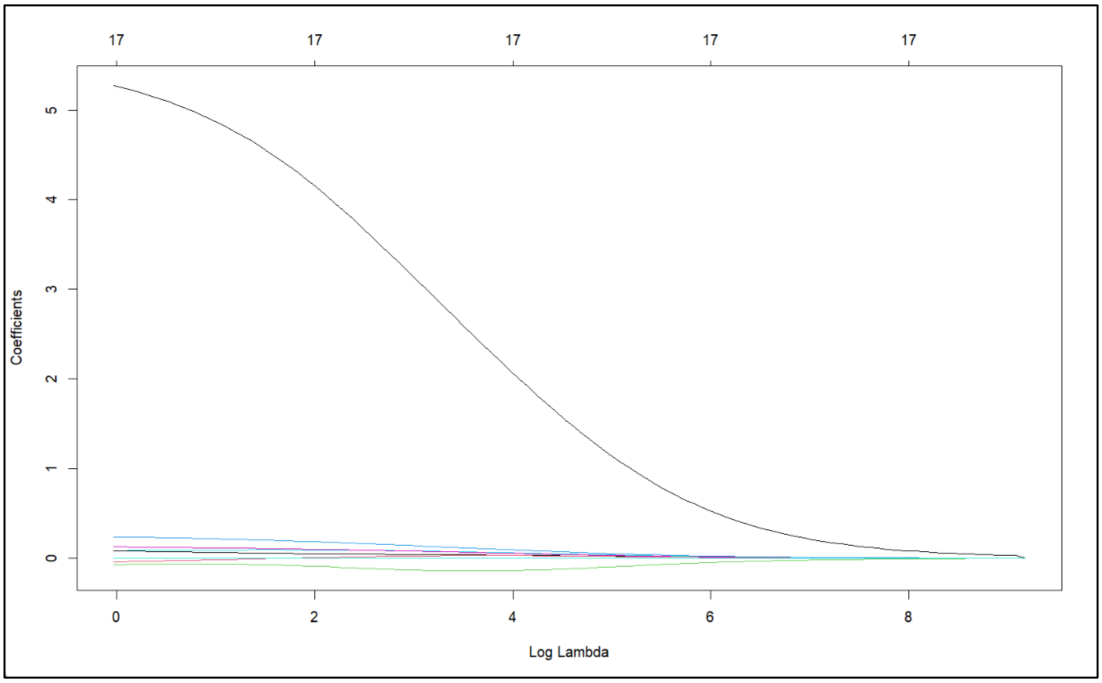
- ➔ In the above figure, we see the value of Lambda.min is 2.958 and that of lambda.1se is 20.871.
- ➔ When the value of alpha is set to be zero, R understands that the model wants to use ridge regularization.

Q3. Visualizing the above results to get a better insight.



- ➔ Above is the graph of the cross-validation curve i.e., red dotted lines along with the error bars that are the upper and lower standard deviation curves.
- ➔ The two dotted vertical lines are the values of lambda i.e., lambda.min and lambda.1se

- The graph is plotted by taking the log of values of lambda on the x-axis and MSE (Mean-squared error) on the y-axis.
- As we can see from the graph, along with the increasing value of lambda the value of MSE increases as well.
- The numbers on top of the graph represent the number of non-zero regression coefficients and in this case, all the numbers are 17 since Ridge regression does not make variables disappear but only tends to zero.

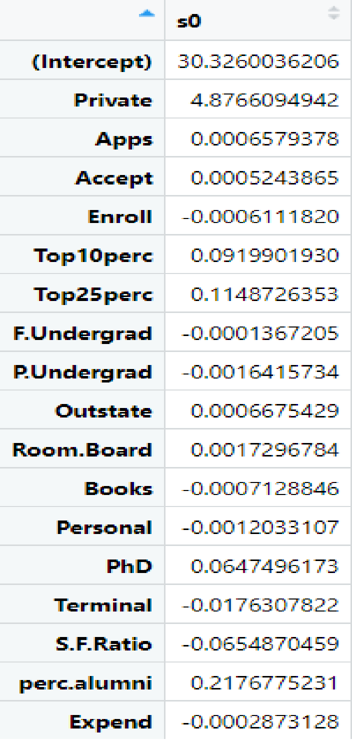
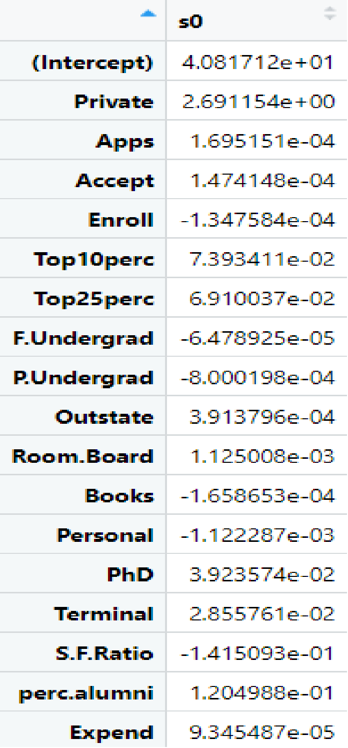


- The graph presented above shows the coefficient path.
- Here, the numbers on the top indicate the number of variables in the model.
- Y-axis represents regularized coefficients for each variable and the x-axis shows the logarithmic value of lambda.
- This graph basically represents the change in the predictor coefficient as the value of lambda increases.

- ➔ In the coefficient path plot, log lambda = 0 means no regularization is applied to the model.

Q4. Applying the Ridge regression model to the training dataset.

- ➔ Here, we make two ridge regression models. One with the minimum value of lambda and the second with the 1se value of lambda.

Minimum value of lambda	1se value of lambda
	

	s0
(Intercept)	30.3260036206
Private	4.8766094942
Apps	0.0006579378
Accept	0.0005243865
Enroll	-0.0006111820
Top10perc	0.0919901930
Top25perc	0.1148726353
F.Undergrad	-0.0001367205
P.Undergrad	-0.0016415734
Outstate	0.0006675429
Room.Board	0.0017296784
Books	-0.0007128846
Personal	-0.0012033107
PhD	0.0647496173
Terminal	-0.0176307822
S.F.Ratio	-0.0654870459
perc.alumni	0.2176775231
Expend	-0.0002873128

	s0
(Intercept)	4.081712e+01
Private	2.691154e+00
Apps	1.695151e-04
Accept	1.474148e-04
Enroll	-1.347584e-04
Top10perc	7.393411e-02
Top25perc	6.910037e-02
F.Undergrad	-6.478925e-05
P.Undergrad	-8.000198e-04
Outstate	3.913796e-04
Room.Board	1.125008e-03
Books	-1.658653e-04
Personal	-1.122287e-03
PhD	3.923574e-02
Terminal	2.855761e-02
S.F.Ratio	-1.415093e-01
perc.alumni	1.204988e-01
Expend	9.345487e-05

- ➔ From the above table, it is seen that model with a minimum value of lambda gives better performance and makes the values of coefficients tend to zero.
- ➔ On the other hand, the model with the 1se value of lambda gives higher values of coefficients.

Q5. and Q6. Model Evaluation Metrics

RMSE is a way that helps to measure the error of a model when predicting the data.

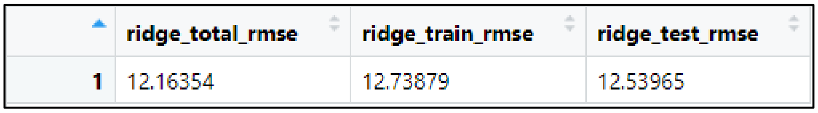
The formula for calculating the RMSE(Root Mean Squared Error) is :

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

where,

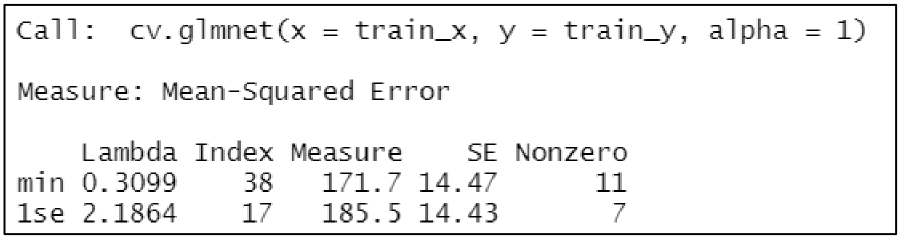
y_i = actual value
y_p = predicted value
n = number of observations/rows

While determining which model is a better fit, we always consider the RMSE value. Here, we can see that the ridge model with the complete dataset has the lowest value. Although, values for the training set and test RMSE do not have a larger difference. Hence, no overfitting or underfitting of data was observed.



	ridge_total_rmse	ridge_train_rmse	ridge_test_rmse
1	12.16354	12.73879	12.53965

Q7. Estimating lambda.min and lambda.1se values using cv.glmnet function for lasso model

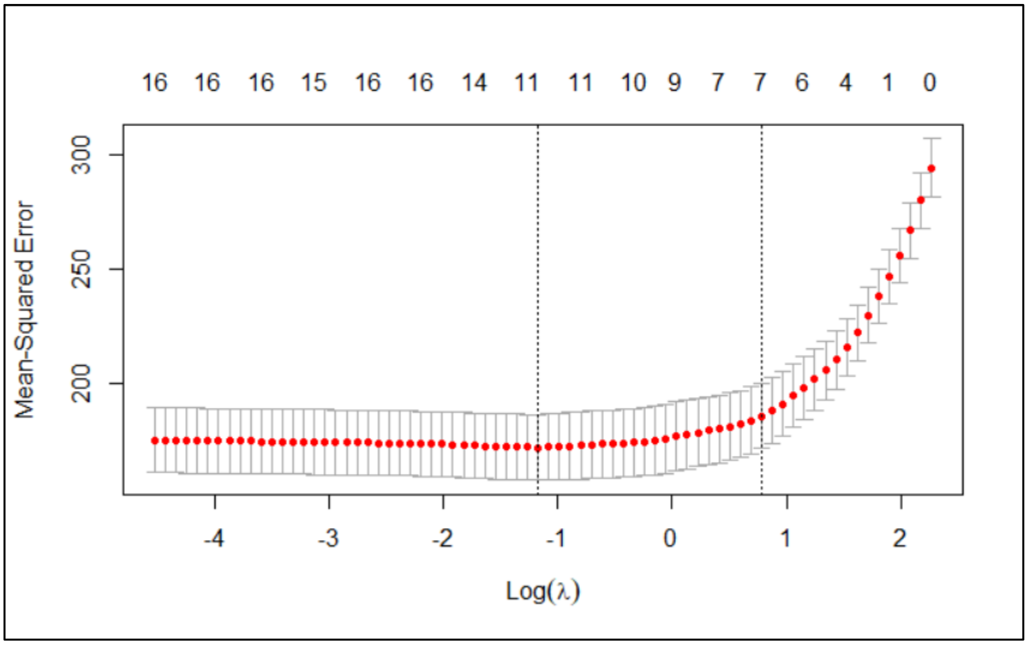


```
Call:  cv.glmnet(x = train_x, y = train_y, alpha = 1)

Measure: Mean-Squared Error

    Lambda Index Measure    SE Nonzero
min 0.3099    38   171.7 14.47      11
1se 2.1864    17   185.5 14.43       7
```

➔ We have already talked about lambda.min and lambda.1se value mean.

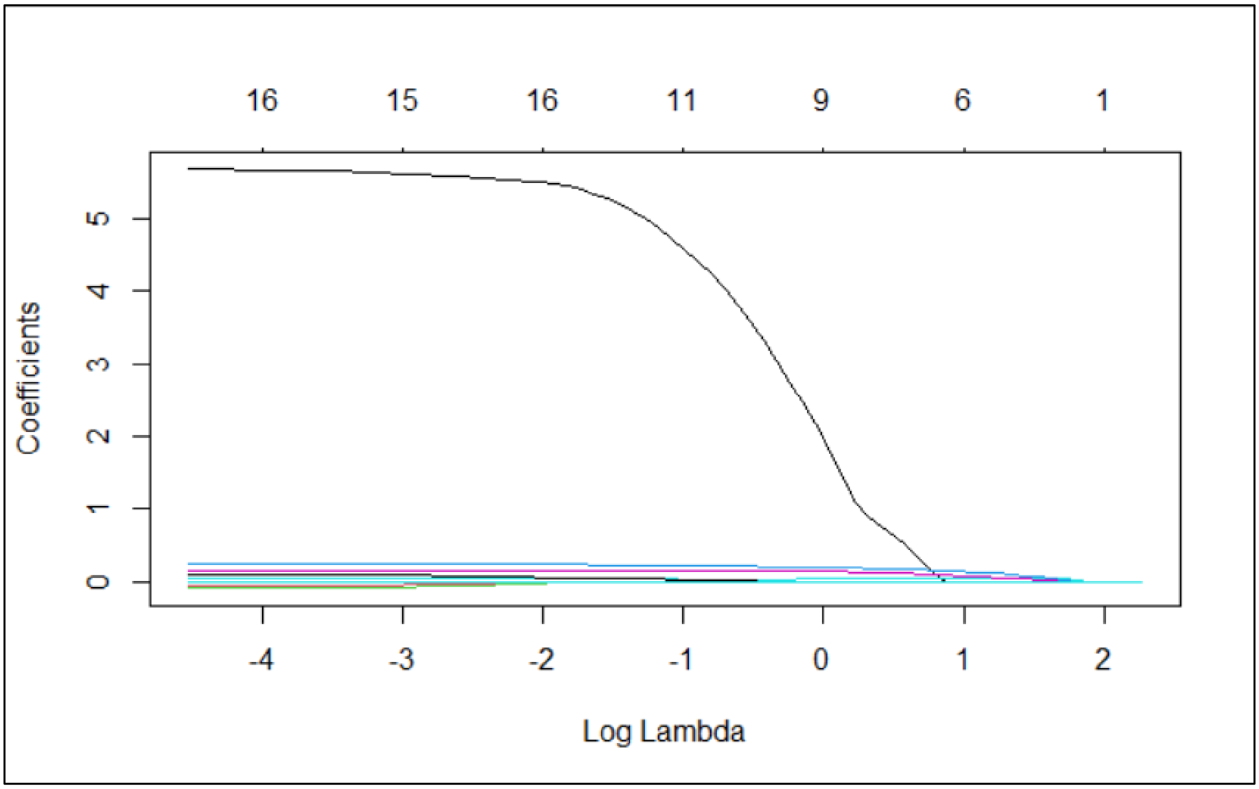
- We also know that when the value of alpha is 1, R understands that the model wants to use Lasso as the method of regularization.
- According to the figure, lambda.min value is 0.3099 and the value for lambda.1se is 2.186
- Since the value of lambda.min is closer to zero we use it for fitting our lasso model.

Q8. Visualizing the graph for the above results.



- Above is the graph of the cross-validation curve for lasso regression. In question 3, we already discussed how to read the graph
- As we can see from the graph, along with the increasing value of lambda the value of MSE increases as well.
- Although, the value of MSE is steady till the log of lambda is 0. It then gradually increases after 0 and reaches a maximum of almost 300 values for MSE.

- ➔ The numbers on top of the graph represent the number of non-zero regression coefficients and in this case, the numbers vary from 0 to 16.

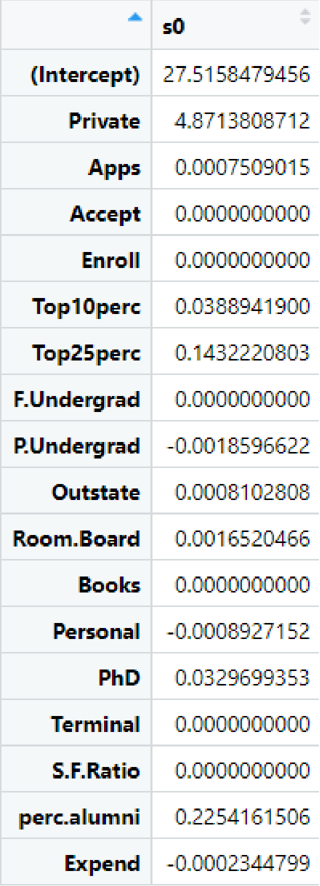


- ➔ The graph represented above is a coefficient path of the Lasso model. We already know how to read the graph.
- ➔ Here, the value of coefficients is steadily decreasing till the logarithmic value of lambda is -2. After that, there is a sudden decrease in the value of coefficients and the number of variables on the top axis also changed drastically.
- ➔ This is because lambda.1se lasso model eliminates more coefficients than lambda.min

Q9. Lasso regression for the training dataset.

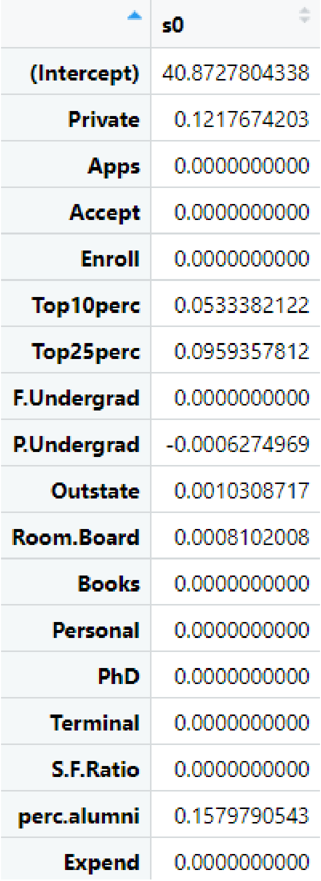
- As we know on applying Lasso regression, the coefficients having multi-collinearity disappear.

Lambda minimum



	s0
(Intercept)	27.5158479456
Private	4.8713808712
Apps	0.0007509015
Accept	0.0000000000
Enroll	0.0000000000
Top10perc	0.0388941900
Top25perc	0.1432220803
F.Undergrad	0.0000000000
P.Undergrad	-0.0018596622
Outstate	0.0008102808
Room.Board	0.0016520466
Books	0.0000000000
Personal	-0.0008927152
PhD	0.0329699353
Terminal	0.0000000000
S.F.Ratio	0.0000000000
perc.alumni	0.2254161506
Expend	-0.0002344799

Lambda 1se



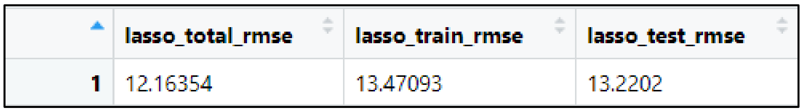
	s0
(Intercept)	40.8727804338
Private	0.1217674203
Apps	0.0000000000
Accept	0.0000000000
Enroll	0.0000000000
Top10perc	0.0533382122
Top25perc	0.0959357812
F.Undergrad	0.0000000000
P.Undergrad	-0.0006274969
Outstate	0.0010308717
Room.Board	0.0008102008
Books	0.0000000000
Personal	0.0000000000
PhD	0.0000000000
Terminal	0.0000000000
S.F.Ratio	0.0000000000
perc.alumni	0.1579790543
Expend	0.0000000000

- The above table shows the coefficient value of the predictor variables after we fit the lasso regression model to the training data set.
- The left side of the table shows the value of coefficients by lambda.min model and the right side of the table shows coefficients by lambda.1se model.
- The model with lambda.1se has a greater number of variables that are eliminated from the model as compared to the model with lambda.min

➔ Accept, Enroll, F.Undergrad, Books, Terminal, and S.F.Ratio are the variables whose coefficients are zero and eliminated from both the models.

➔ Apps, Personal, PhD, and Expend are the extra variables whose variables are eliminated from the lambda.1se model.

Q10. And Q11. Model Evaluation Metrics



	lasso_total_rmse	lasso_train_rmse	lasso_test_rmse
1	12.16354	13.47093	13.2202

➔ The above table shows us the value of RMSE of the lasso model with lambda.1se model for the total dataset, train dataset, and test dataset.

➔ As we can see, the model having a complete dataset has less RMSE value. This might be because of the increase in the number of observations.

➔ But there is no underfitting or overfitting of the data observed with these RMSE values.

Q12. Concluding which model had better performance and the reason behind it.

➔ After performing ridge and lasso on our training and test dataset, we understand that there is no overfitting or underfitting by either of the models.

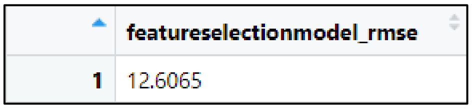
➔ However, ridge regression was able to give an RMSE value lesser than what lasso regression gave.

➔ Although the value was increased only by 1, it can potentially change the accuracy of the model.

➔ Usually, lasso regression is better than ridge because it acts as a feature selection model itself.

➔ But sometimes ridge regression acts better in the case of correlated features as compared to Lasso because it uses all the features, but the coefficients will be distributed according to the correlation.

Q13. Comparing the models with the feature selection model to see if the performance is changed.



	featureselectionmodel_rmse
1	12.6065

➔ I tried performing stepwise selection techniques here. But the RMSE value for both, forward and backward selection techniques was the same i.e., 12.6065

➔ The value for the stepwise selection technique model and ridge regression model is almost the same with some negligible difference.

➔ Hence, we can either use ridge regression or feature selection.

➔ I would prefer using ridge regression as it is time-saving and needs less computation power.

Conclusion :

By completing this assignment, I understood the concept of regularisation. How to check if the model is under fitted or overfitting and what can be done in either of the cases. The fundamental difference between Ridge and Lasso was also understood. First, we divided the data into train and test splits and compared which model gave the minimal RMSE value. We also did some cross-validation methods on our models.

References

1. Ajitesh Kumar(10th July 2019) The what, when, and why of regularization in Machine Learning. Last retrieved on 05th May 2022 https://dzone.com/articles/what-when-amp-why-of-regularization-in-machine-learning
2. Aarshay (28th January 2016) A complete tutorial on Ridge and Lasso Regression. Last retrieved on 05th May,2022 https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-complete-tutorial
3. Utkarsh Kumar (1st January 2020) Create Matrix and Data frame from lists in R Programming. Last Retrieved on 05th May 2022 https://www.geeksforgeeks.org/create-matrix-and-data-frame-from-lists-in-r-programming
4. Trevor Hastie, Junyang Qian, Kenneth Tay (1st November 2021) An introduction to glmnet https://glmnet.stanford.edu/articles/glmnet.html
5. Zach (10th May 2021) What is considered a good RMSE value? Last retrieved on 5th May 2022 https://www.statology.org/what-is-a-good-rmse