Logo

Description automatically generated

ALY6015 Module 4 Project – R Practice

**Authored by: Prateek Singh**

Course: ALY6015 – Intermediate Analytics

Submission Date: 02/14/2022

**Instructor’s name: Jiyoung Yun**

# Introduction

When a Machine Learning model is restricted to a training set, it is unable to perform effectively on unknown data. This is known as overfitting. Regularization is a method for reducing inefficiencies and avoiding overfitting by fitting the function suitably on the supplied training set. LASSO (Least Absolute Shrinkage and Selection Operator) regression is a regression model that employs the L1 Regularization approach. Ridge regression is a regression model that use the L2 regularization technique.

The "absolute value of magnitude" of the coefficient is added to the loss function as a penalty term in Lasso Regression (L).

Ridge regression adds a penalty component to the loss function called "squared magnitude of coefficient" (L).

We are considering a college dataset of 18 variables and 777 observations for building regularization models using Ridge and LASSO functions. For this assignment we are taking into consideration Graduation Rate variable as our response variable to build the linear models through Ridge and LASSO regression.

# Analysis

Part 1:

To evaluate how efficiently our model is predicting, we will go ahead and create a train and test set. A train data set has 70% of the observations from the original data set which is 545 observations in total, and the test data set has remaining 30% of the data set having 232 observations. As we are going to use glmnet function for fitting our model, we will be converting the train and test data frames to matrix as glmnet function required a matrix as input. We have used an advanced function, model.matrix, to convert our train and test data frame to a matrix because our dataset has a categorical variable and this function automatically assigns a dummy variable in this case.

Next step is to find the best value of lambda using the cross-validation technique and plot it. Y-axis represents the Mean Square Error value, x-axis is a log of lambda and the upper x-axis, which is the 3rd axis, is the number of non-zero coefficients in the model for that particular value of lambda. The plot is the error estimates or error metrics, and the red dot denoted the last metric which we have computed through cross validation process. From the graph we can see the lambda minimum value is 15 and lambda 1 SE is 7 which is maximum value within 1 standard error of the minimum. Hence, it can be safely stated that all the variables are not retained in the model. 7 non-zero coefficient variables represent the model contains only 7 variables and it has set at least eight variable coefficients to zero.

Text

Description automatically generated with medium confidence Histogram

Description automatically generated

Fig. 1: Finding Optimal Value of λ

We will use this lambda minimum and lambda 1 SE values to fit our model. We can see that the coefficients are different for both the models and is slightly lower for most variables in 1 SE model. Also, no coefficients were neutralized because of the Ridge Model.

Text

Description automatically generatedText

Description automatically generated

Fig. 2: Lambda Min and Lambda 1 SE Ridge Model fit

To compare, we ran ols model without regularization and the coefficients were much closer to lambda minimum.

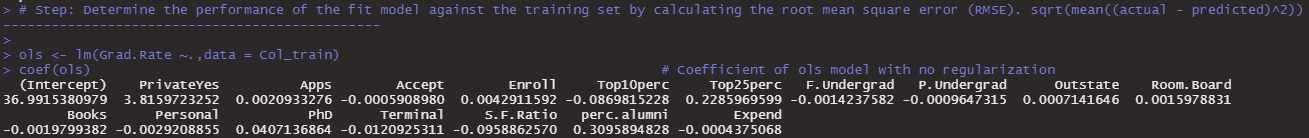


Fig. 3: Ols Model without Regularization

We now went ahead to predict new values for graduation rate against the test data set for ols linear regression model. Running Root Mean Squared Error gives how good is our model working. We created a predictive model for both of our train and test data set using Ridge Lambda 1 SE. It is evident that test RMSE value is significantly higher than train RMSE value which clarifies that the Ridge model is overfitted. Also, when compared with overall Test data set, the performance isn’t very bad nevertheless it increased trivially albeit no variables were eliminated.

Text

Description automatically generated

Fig. 4: Comparison between Train RMSE and Test RMSE

Part 2:

Now we are going to repeat the similar steps to fit a LASSO model. The plot representation will be same in both the model with exactly same values as we are using same train and test matrix.

Text

Description automatically generated Histogram

Description automatically generated

Fig. 5: Finding Optimal Value of λ for LASSO model

Now using the alpha value as 1 we can fit the LASSO model against train data set for both Lambda Minimum and Lambda 1 SE. For minimum lambda value the coefficients removed were 2 and hence the 15 degrees of freedom. When we increase the value of lambda to 1 standard error of the minimum value of lambda, now we see the degree of freedom go down to 7 neutralizing 10 coefficients. As we increase the value of lambda in LASSO model, and it approaches to infinity the number of coefficients left in the model is zero which suggests it to be a good fit. In below screenshots we can see the coefficients reduced to zero.

Text

Description automatically generated Text

Description automatically generated

Fig. 6: Lambda Min and Lambda 1 SE LASSO Model fit

Now to compare and calculate the root mean squared error, we run the ols model without regularization. Coefficients for ols model seems to be lot closer to LASSO lambda minimum model than LASSO lambda 1SE model. Coefficients in 1SE model tends to be little smaller than minimum model.

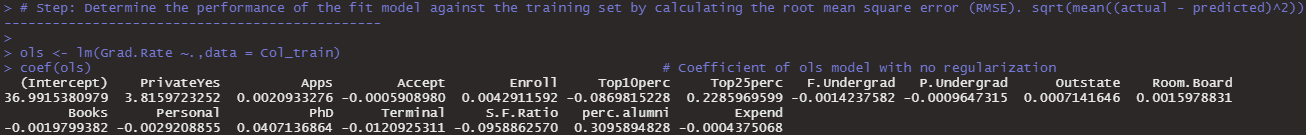


Fig. 7: Ols Model without Regularization

To predict new values for graduation rate against the test data set for this LASSO linear regression model we computed Root Mean Squared Error which gives how good is our model working. We created a predictive model for both of our train and test data set using LASSO Lambda 1 SE. It is evident that Test RMSE value is greater than Train RMSE value which affirms that the LASSO model is overfitting but lesser than Ridge model. Also, when compared with overall Test data set, the test model performance isn’t very bad nevertheless it went up trivially albeit variables were eliminated.

Text

Description automatically generated

Fig. 8: Comparison between Train RMSE and Test RMSE for LASSO Model

# Conclusion

LASSO model is slightly better than Ridge model with the sample train and test data set created as the test RMSE values for LASSO is marginally lesser than the Ridge model. Despite the fact, the coefficients in the LASSO models were reduced to zero, both the models were overfitting. LASSO model was expected to perform better than Ridge, however the overfitting of the model was unanticipated. This means the model works best when predicting on the sample but fails while predicting out of sample.

Residual standard error value for stepwise selection fitness model is lesser than that of Ridge and LASSO both when test data set is considered. Hence, it can be considered as the best fit model. Stepwise selection process also selects the best fitting predictor variables for the model hence can avoid the situation of overfitting the model.

# Reference

1. Ridge regression | code and analysis - youtube. (n.d.). Retrieved February 7, 2022, from <https://www.youtube.com/watch?v=fWg1_G7EkVI>
2. Gupta, A. (2021, June 29). Regularization in machine learning. GeeksforGeeks. Retrieved February 16, 2022, from <https://www.geeksforgeeks.org/regularization-in-machine-learning/>
3. Harris, M. (2016, February 16). Model overfitting – some basics. Quantitative Archaeology. Retrieved February 16, 2022, from <https://matthewdharris.com/2016/02/16/model-overfitting-some-basics/>
4. Ku, D. (2018, July 16). Why lasso yield a higher MSE then ridge? Cross Validated. Retrieved February 16, 2022, from <https://stats.stackexchange.com/questions/357413/why-lasso-yield-a-higher-mse-then-ridge>
5. 2.Starmer. J. (2018, September 17). Machine Learning Fundamentals: Bias and Variance. Retrieved on 5th February, 2022 from <https://www.youtube.com/watch?v=EuBBz3bI-aA&ab_channel=StatQuestwithJoshStarmer>

# Appendix

# ALY6015 Module 4 R Practice: Singh Prateek ------------------------------------------------

#----------------- Author: Prateek Singh

#----------------- Submission Date: 14th Feb, 2022

#----------------- Tutor: Jiyoung Yun

# Step: Installing New Libraries ------------------------------------------------

install.packages("Metrics")

install.packages("glmnet")

# Step: Importing Libraries ------------------------------------------------

library(ISLR)

library(dplyr)

library(glmnet)

library(caret)

library(Metrics)

library(psych)

# Step: Load the inbuilt Dataset and do the EDA ------------------------------------------------

data("College")

str(College)

describe(College)

summary(College)

# Step: Split the data into a train and test set ------------------------------------------------

set.seed(3456)

Index\_Train <- createDataPartition(College$Grad.Rate, p = 0.7, list = FALSE, times = 1)

Col\_train <- College[Index\_Train,]

Col\_test <- College[-Index\_Train,]

# Create a model matrix for both train and test data

Col\_train\_x <- model.matrix(Grad.Rate ~.,Col\_train)[,-1]

Col\_test\_x <- model.matrix(Grad.Rate ~.,Col\_test)[,-1]

Col\_train\_y <- Col\_train$Grad.Rate

Col\_test\_y <- Col\_test$Grad.Rate

# Step: Ridge Regression ------------------------------------------------

# Step: Use the cv.glmnet function to estimate the lambda.min and lambda.1se values ------------------------------------------------

set.seed(3456)

cv\_ridge <- cv.glmnet(Col\_train\_x, Col\_train\_y, nfolds = 10)

log(cv\_ridge$lambda.min)

log(cv\_ridge$lambda.1se)

# Step: Plot the results from the cv.glmnet function ------------------------------------------------

plot(cv\_ridge)

# Step: Fit a Ridge regression model against the training set and report on the coefficients ------------------------------------------------

Ridge\_model\_min <- glmnet(Col\_train\_x, Col\_train\_y, alpha = 0, lambda = cv\_ridge$lambda.min)

Ridge\_model\_min

coef(Ridge\_model\_min)

Ridge\_model\_1se <- glmnet(Col\_train\_x, Col\_train\_y, alpha = 0, lambda = cv\_ridge$lambda.1se)

Ridge\_model\_1se

coef(Ridge\_model\_1se)

# Step: Determine the performance of the fit model against the training set by calculating the root mean square error (RMSE). sqrt(mean((actual - predicted)^2)) ------------------------------------------------

ols <- lm(Grad.Rate ~.,data = Col\_train)

coef(ols) # Coefficient of ols model with no regularization

pred.ols <- predict(ols, new = Col\_test)

rmse(Col\_test$Grad.Rate, pred.ols) # Root Mean Square Error

pred.train <- predict(Ridge\_model\_1se, newx = Col\_train\_x) # College Train Set Predictions

Col\_train\_rmse\_Ridge <- rmse(Col\_train\_y, pred.train)

Col\_train\_rmse\_Ridge

# Step: Determine the performance of the fit model against the test set by calculating the root mean square error (RMSE). Is your model overfit? ------------------------------------------------

pred.test <- predict(Ridge\_model\_1se, newx = Col\_test\_x) # College Test Set Predictions

Col\_test\_rmse\_Ridge <- rmse(Col\_test\_y, pred.test)

Col\_test\_rmse\_Ridge

# Step: LASSO Regression ------------------------------------------------

# Step: Use the cv.glmnet function to estimate the lambda.min and lambda.1se values ------------------------------------------------

set.seed(3456)

cv\_lasso <- cv.glmnet(Col\_train\_x, Col\_train\_y, nfolds = 10)

log(cv\_lasso$lambda.min)

log(cv\_lasso$lambda.1se)

# Step: Plot the results from the cv.glmnet function ------------------------------------------------

plot(cv\_lasso)

# Step: Fit a Lasso regression model against the training set and report on the coefficients. Do any coefficients reduce to zero? If so, which ones?------------------------------------------------

Lasso\_model\_min <- glmnet(Col\_train\_x, Col\_train\_y, alpha = 1, lambda = cv\_lasso$lambda.min)

Lasso\_model\_min

coef(Lasso\_model\_min)

Lasso\_model\_1se <- glmnet(Col\_train\_x, Col\_train\_y, alpha = 1, lambda = cv\_lasso$lambda.1se)

Lasso\_model\_1se

coef(Lasso\_model\_1se)

# Step: Determine the performance of the fit model against the training set by calculating the root mean square error (RMSE). sqrt(mean((actual - predicted)^2)) ------------------------------------------------

ols <- lm(Grad.Rate ~.,data = Col\_train)

coef(ols) # Coefficient of ols model with no regularization

pred\_ols <- predict(ols, new = Col\_test)

rmse(Col\_test$Grad.Rate, pred\_ols) # Root Mean Square Error

pred\_train <- predict(Lasso\_model\_1se, newx = Col\_train\_x) # College Train Set Predictions

Col\_train\_rmse\_Lasso <- rmse(Col\_train\_y, pred\_train)

Col\_train\_rmse\_Lasso

# Step: Determine the performance of the fit model against the test set by calculating the root mean square error (RMSE). Is your model overfit? ------------------------------------------------

pred\_test <- predict(Lasso\_model\_1se, newx = Col\_test\_x) # College Test Set Predictions

Col\_test\_rmse\_Lasso <- rmse(Col\_test\_y, pred\_test)

Col\_test\_rmse\_Lasso

# Step: Fit a model and compare ------------------------------------------------

set.seed(123)

Train\_Index <- createDataPartition(College$Grad.Rate, p=0.7, list = FALSE, times = 1)

train\_model <- College[Train\_Index,]

test\_model <- College[-Train\_Index,]

Model <- step(lm(Grad.Rate ~., data = test\_model), direction = 'both')

coef(Model)

summary(Model)