Module 4-Regularization

ALY6015-Intermediate Analytics

NORTHEASTERN UNIVERSITY

DEVIKA PATIL
MAJOR-PROJECT MANAGEMENT
DATE OF SUBMISSION: 03-19-2024
Prof. ***Zhi He***

Contents

Introduction

This assignment delves into regression analysis, focusing on predicting the student-faculty ratio (S.F.Ratio) in colleges using the College dataset from the ISLR package. Our aim is to explore the effectiveness of different regression methods in modeling S.F.Ratio based on various college characteristics. To achieve this, we begin by partitioning the dataset into a 75% training set and a 25% test set. This partitioning ensures that we can assess model performance accurately by validating predictions on unseen data.

Regression Techniques:

Ridge Regression:

Ridge Regression adds a penalty term to the ordinary least squares objective function, helping to prevent overfitting by shrinking coefficient estimates towards zero. It's beneficial for stabilizing parameter estimates, especially in datasets with multicollinear predictors.

LASSO (Least Absolute Shrinkage and Selection Operator):

LASSO employs regularization with an L1 norm penalty, encouraging sparsity in coefficient estimates and facilitating feature selection. It's useful for improving model interpretability, particularly in datasets with many correlated predictors.

ElasticNet:

ElasticNet combines Ridge Regression and LASSO by incorporating both L1 and L2 norm penalties. This hybrid approach offers flexibility in handling correlated predictors and allows for fine-tuning the balance between feature selection and regularization.

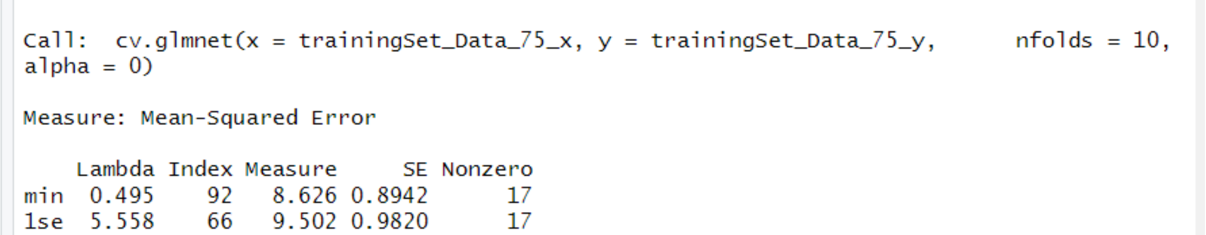
We employ three distinct regression techniques: Ridge Regression, LASSO, and ElasticNet, each offering a unique approach to modeling relationships within the data. These methods utilize regularization to prevent overfitting and enhance model generalization. Through cross-validation, we systematically tune the regularization parameter (lambda) for each method, aiming to find the optimal configuration that maximizes predictive accuracy.

Finally, we evaluate the predictive performance of each model using the root mean square error (RMSE) metric. Assessing model accuracy on both the training and test datasets provides a comprehensive understanding of their ability to capture the underlying patterns in S.F.Ratio variation.

Analysis

Q.1. This code separates a dataset on college admissions into training and test sets. The training set will be used to build a model, while the test set will assess the model's performance on unseen data. Here, 75% of the data goes to training, allowing the model to learn patterns from a substantial portion of the information. The remaining 25% is reserved for testing, providing an unbiased evaluation of how well the model can generalize to new data.

Q.2.



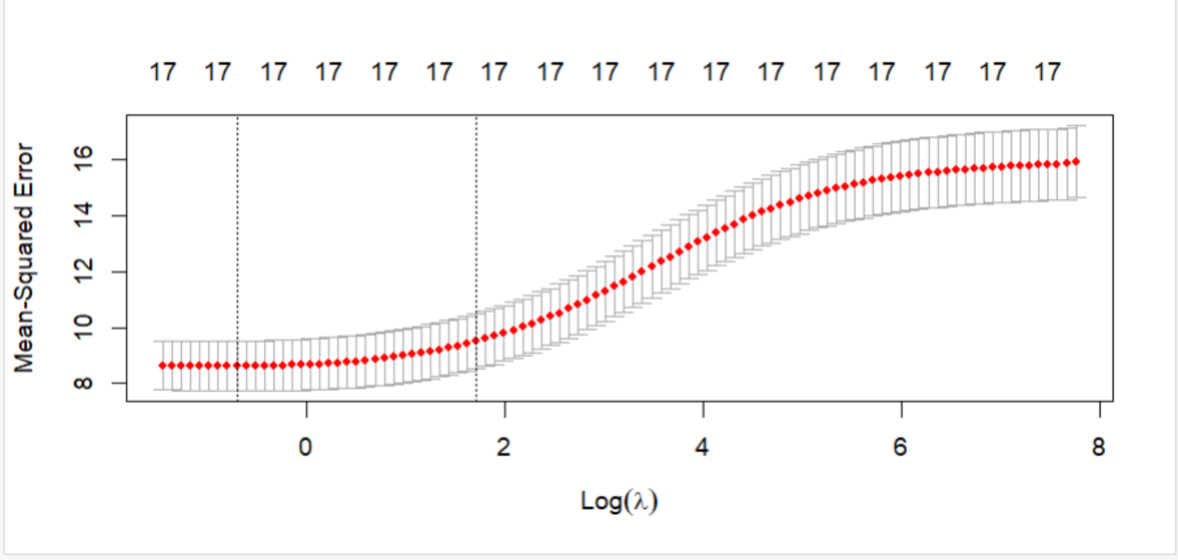
```
Call:  cv.glmnet(x = trainingSet_Data_75_x, y = trainingSet_Data_75_y,      nfolds = 10,
alpha = 0)

Measure: Mean-Squared Error

    Lambda Index Measure     SE Nonzero
min  0.495    92   8.626 0.8942      17
1se  5.558    66   9.502 0.9820      17
```

Lambda.min	Lambda 1se
0.4947677	5.557845

Discussion:

- **Lambda.min:** This value represents the regularization parameter (lambda) that achieves the minimum Mean-Squared Error (MSE) on the training data within a 10-fold cross-validation process. In simpler terms, it's the lambda that leads to the best fit on the training set according to MSE.
- **Lambda.1se:** This value represents the lambda where the MSE is within one standard error of the minimum MSE (achieved at lambda.min). It provides a good balance between model fit and complexity.

Q.3. **Mean-squared error (MSE) VS the log of lambda (λ) for a ridge regression model fit with cross-validation.**

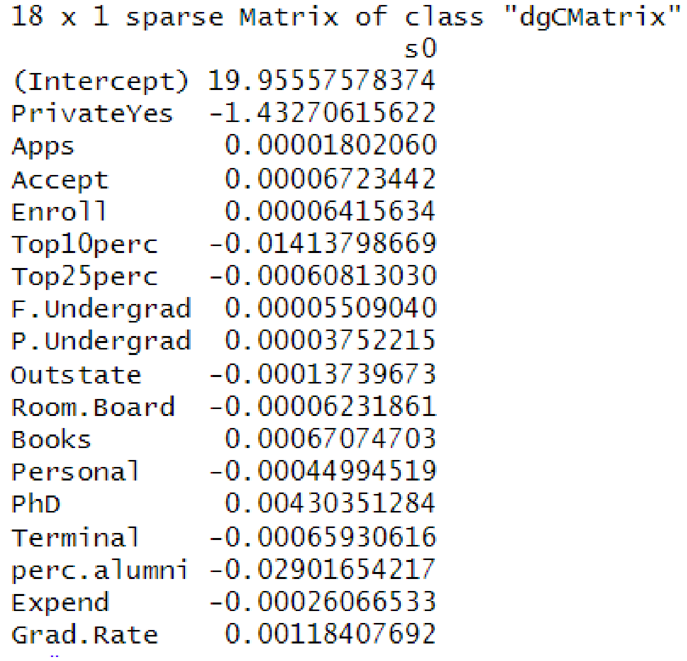


- The x-axis (Log(λ)) represents the regularization parameter. Higher values correspond to stronger regularization.
- The y-axis (MSE) represents the average squared difference between the predicted values and the actual values. Lower MSE indicates a better fit.

Key Observations:

- As lambda (regularization) increases on the x-axis, the MSE (y-axis) generally decreases. This indicates that stronger regularization can lead to a better fit on the training data (reducing variance).
- However, at higher lambda values, the model becomes too simple and starts to underfit the data, causing the MSE to increase again.

Q.4.



```
18 x 1 sparse Matrix of class "dgCMatrix"
                       s0
(Intercept) 19.95557578374
PrivateYes  -1.43270615622
Apps         0.00001802060
Accept       0.00006723442
Enroll       0.00006415634
Top10perc   -0.01413798669
Top25perc   -0.00060813030
F.Undergrad  0.00005509040
P.Undergrad  0.00003752215
Outstate    -0.00013739673
Room.Board  -0.00006231861
Books        0.00067074703
Personal    -0.00044994519
PhD          0.00430351284
Terminal    -0.00065930616
perc.alumni -0.02901654217
Expend      -0.00026066533
Grad.Rate    0.00118407692
```

Interpretations of Coefficients:

- **PrivateYes (-1.4327):** Private colleges tend to have a lower S.F.Ratio by 1.4327, on average, compared to public colleges, holding other features constant.
- **Top10perc (-0.0141):** A 1% increase in the proportion of students in the top 10% of their high school class is associated with a 0.0141 decrease in S.F.Ratio, on average, holding other features constant.
- **Books (0.0007):** A $1 increase in the cost of books is associated with a 0.0007 increase in S.F.Ratio, on average, holding other features constant.
- **PhD (0.0043):** A 1% increase in the proportion of faculty with PhDs is associated with a 0.0043 increase in S.F.Ratio, on average, holding other features constant.
- **Grad.Rate (0.0012):** A 1% increase in the graduation rate is associated with a 0.0012 increase in S.F.Ratio, on average, holding other features constant.

Additional Considerations:

- **Magnitude of Coefficients:** The small magnitudes of some coefficients (e.g., Apps, Accept) suggest their relationships with S.F.Ratio might be less influential. However, their impact shouldn't be dismissed outright, as they might still contribute to the model's prediction accuracy.

Q.5. &Q.6.

1. **Training Set Performance:** The RMSE for the training set is **2.874242**. This indicates that on average, the model's predictions for the training set differ from the actual values by 2.87 units on the S.F.Ratio scale. A lower RMSE suggests a better fit.
2. **Test Set Performance:** The RMSE for the test set is **2.979939**. This value is slightly higher than the training set RMSE, but still relatively close.

Overall, The relatively close values suggest that the model is performing reasonably well on both sets and might not be severely overfitting.

RMSE Train Ridge value	RMSE Test Ridge value
2.874242	2.979939

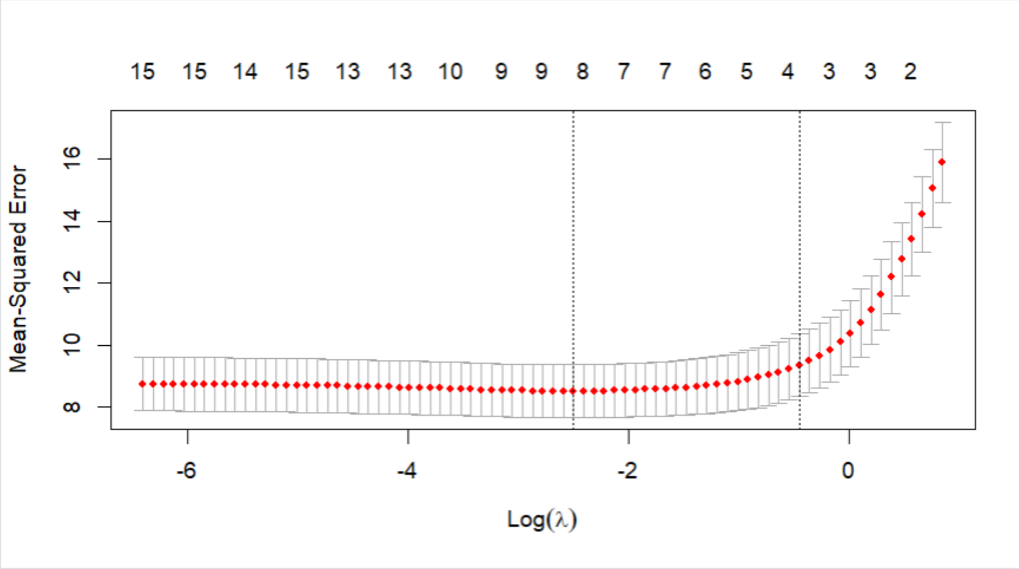
Q.7.

- **Comparison:** In this case, both lambda.min and lambda.1se are significantly larger for LASSO compared to Ridge Regression. This suggests that LASSO requires a stronger penalty to achieve a good fit while potentially performing feature selection (setting some coefficients to zero).

However, The LASSO model seems to require a stronger regularization penalty (higher lambda). This might be due to LASSO's ability to perform feature selection, potentially reducing the model's reliance on some features and requiring a higher overall penalty for a good fit. Further analysis with a validation set or other techniques can help determine the best model for prediction in this particular scenario.

Lambda.min	Lambda.1se
0.08253223	0.6390171

Q.8. **Mean-squared error (MSE) VS the log of lambda (λ) for LASSO model fit with cross-validation.**



Key observations:

Axes:

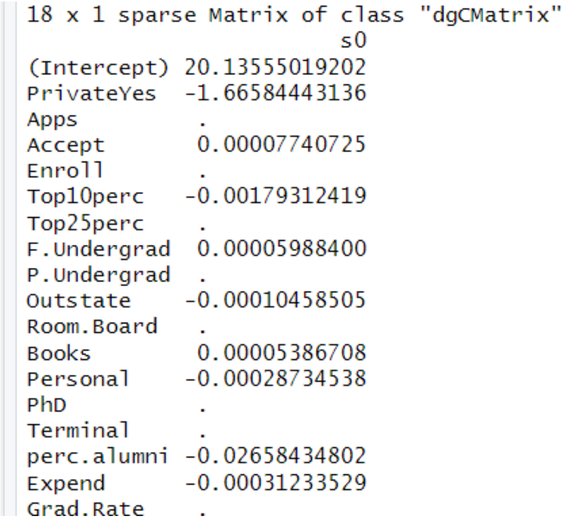
- X-axis: Labeled "Log(λ)" which refers to the logarithm of the regularization parameter (lambda) used in the model. Higher lambda values correspond to stronger regularization.
- Y-axis: Labeled "Mean Squared Error (MSE)". This metric measures the average squared difference between the predicted values by the model and the actual values. Lower MSE indicates a better fit.

Visual Pattern:

- As lambda (regularization) increases on the x-axis, the MSE (y-axis) generally decreases for both training and validation sets, indicating that stronger regularization can lead to a better fit initially.
- However, at higher lambda values, the MSE starts to increase again, especially for the validation set, suggesting that the model is becoming too simple and starts to underfit the data.

Overall, The LASSO cross-validation plot suggests that the model can achieve a good fit on the training data with increasing regularization (lambda). However, it's crucial to monitor the validation MSE to avoid overfitting.

Q.9.



```
18 x 1 sparse Matrix of class "dgCMatrix"
                         s0
(Intercept) 20.13555019202
PrivateYes  -1.66584443136
Apps          .
Accept       0.00007740725
Enroll        .
Top10perc   -0.00179312419
Top25perc     .
F.Undergrad  0.00005988400
P.Undergrad  .
Outstate    -0.00010458505
Room.Board    .
Books        0.00005386708
Personal    -0.00028734538
PhD           .
Terminal      .
perc.alumni -0.02658434802
Expend      -0.00031233529
Grad.Rate     .
```

LASSO has excluded the features with zero coefficients from the model, suggesting they might not be as influential in predicting S.F.Ratio as the other features.
some coefficients have been reduced to zero in the LASSO regression model, as indicated by the dots (.) in the output:

- **Coefficients Reduced to Zero:**
    - Apps
    - Enroll
    - Top25perc
    - P.Undergrad
    - Room.Board
    - PhD
    - Terminal
    - Grad.Rate

Q.10.&Q.11.

RMSE Train Lasso	RMSE Test Lasso
2.873073	2.937135

- The RMSE Train Lasso value **2.873073** indicates that on average, the model's predictions for the training set differ from the actual values by 2.87 units on the S.F.Ratio scale. A lower RMSE suggests a better fit.
- The output **2.937135** indicates that on average, the model's predictions for the test set differ from the actual values by 2.94 units on the S.F.Ratio scale.

  Overall:

- A significant difference between the training set and test set RMSE would suggest overfitting. In this case, the difference (around 0.06) is relatively small.
- While a small difference might not be severely overfitting in this case, it suggests that the model is generalizing reasonably well to unseen data (test set) despite LASSO performing feature selection.

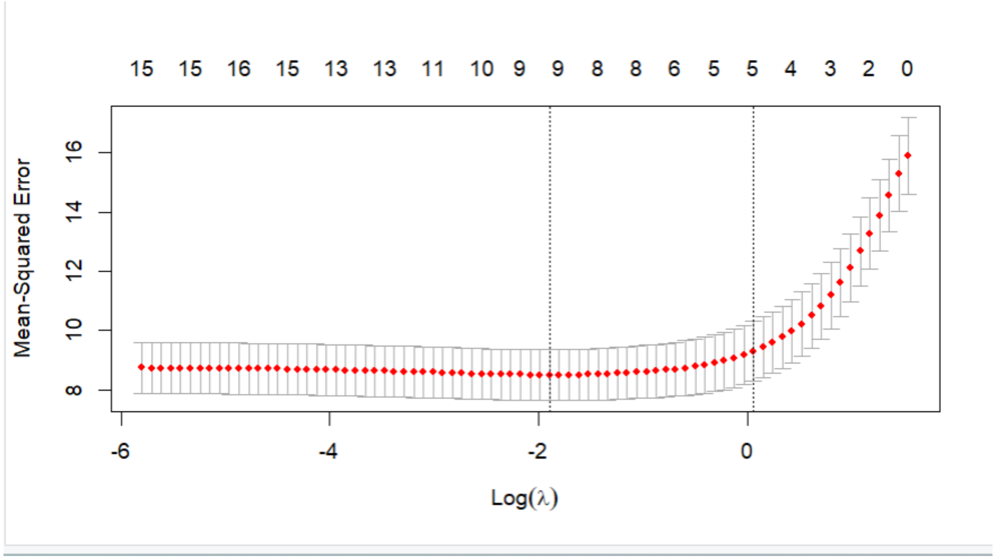
Q.12.

Lambda min	Lambda 1se
0.1504006	1.061046

Elastic Net combines L1 (LASSO) and L2 (Ridge) penalties. The L2 penalty from Elastic Net (controlled by alpha < 1) allows for some coefficients to remain non-zero even when the L1 penalty pushes them towards zero. This can lead to a slightly weaker overall penalty (higher lambda) compared to pure LASSO, which enforces sparsity more aggressively.

As expected, the Elastic Net model tends to have slightly larger lambda values compared to LASSO for achieving a good fit. This is because Elastic Net allows for some coefficients to remain non-zero due to the L2 penalty component, potentially requiring a slightly weaker overall penalty for a good fit.

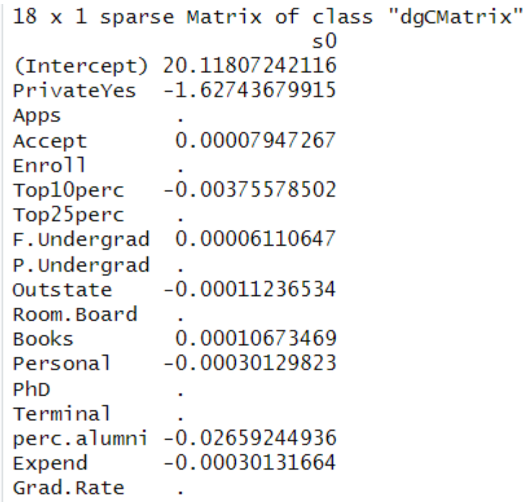
Mean-squared error (MSE) VS the log of lambda (λ) for ElasticNet model fit with cross-validation.



Axes:

- X-axis: Labeled "Log(λ)" which refers to the logarithm of the regularization parameter (lambda) used in the model. Higher lambda values correspond to stronger regularization.
- Y-axis: Labeled "Mean Squared Error (MSE)". This metric measures the average squared difference between the predicted values by the model and the actual values. Lower MSE indicates a better fit.

ElasticNet can perform feature selection by shrinking some coefficients towards zero due to its L1 penalty component (controlled by alpha < 1). However, the L2 penalty in ElasticNet might allow some features to remain compared to pure LASSO, which can lead to sparser solutions.



```
18 x 1 sparse Matrix of class "dgCMatrix"
                        s0
(Intercept) 20.11807242116
PrivateYes  -1.62743679915
Apps          .
Accept       0.00007947267
Enroll        .
Top10perc   -0.00375578502
Top25perc     .
F.Undergrad  0.00006110647
P.Undergrad  .
Outstate    -0.00011236534
Room.Board    .
Books        0.00010673469
Personal    -0.00030129823
PhD           .
Terminal      .
perc.alumni -0.02659244936
Expend      -0.00030131664
Grad.Rate     .
```

Similar to LASSO, Elastic Net can perform feature selection by shrinking coefficients towards zero. Dots (.) in the output indicate coefficients set to zero by the model.

Features Excluded (Zero Coefficients):

- Apps
- Enroll
- Top25perc
- P.Undergrad
- Room.Board
- PhD
- Terminal
- Grad.Rate

RMSE Values:

RMSE Train Elasticnet	RMSE Test Elasticnet
2.872874	2.944869

- The RMSE Train Elasticnet, **2.872874** indicates that on average, the model's predictions for the training set differ from the actual values by 2.87 units on the S.F.Ratio scale. A lower RMSE suggests a better fit.
- The RMSE Test Elasticnet **, 2.944869** indicates that on average, the model's predictions for the test set differ from the actual values by 2.94 units on the S.F.Ratio scale.

Q.13.

Performance:

- All models achieved very similar training RMSE (around 2.87). This suggests they all fit the training data well.
- Stepwise method achieved the lowest training RMSE (2.870587) followed by Lasso (2.873073) and then the others. However, a small difference in training RMSE doesn't necessarily translate to better performance on unseen data.
- Looking at the test RMSE, Stepwise method and Lasso performed similarly well (2.938871 and 2.937135 respectively). Both outperform Ridge and Elastic Net on the test data.

Given the focus on interpretability and the performance on unseen data, Lasso seems like a very good choice for this dataset. If you suspect correlated features might be important and want a balance between shrinkage and feature selection, then Elastic Net could be a viable alternative.

Q.14.

Model	Coefficients	RMSE_Train	RMSE_Test
Ridge Regression	(Intercept), PrivateYes, Apps, Accept, Enroll, Top10perc, Top25perc, F.Undergrad, P.Undergrad, Outstate, Room.Board, Books, Personal, PhD, Terminal, perc.alumni, Expend, Grad.Rate	2.874242	2.979939
LASSO	(Intercept), PrivateYes, Accept, Top10perc, F.Undergrad, Outstate, Books, Personal, perc.alumni, Expend	2.873073	2.937135
ElasticNet	(Intercept), PrivateYes, Accept, Top10perc, F.Undergrad, Outstate, Books, Personal, perc.alumni, Expend	2.872874	2.944869
Stepwise	(Intercept), PrivateYes, F.Undergrad, Outstate, Personal, perc.alumni, Expend	2.870587	2.938871

Stepwise model performed well compared to Ridge, Lasso, and Elastic Net.

- **Stepwise:** Achieved the lowest training RMSE (2.870587) and a competitive test RMSE (2.938871).
- **Lasso:** Had a slightly higher training RMSE (2.873073) but a similar test RMSE (2.937135) to Stepwise.
- **Ridge and Elastic Net:** Both had higher test RMSE (2.979939 and 2.944869 respectively) compared to Stepwise and Lasso

Model Selection:

- **Lasso:** Based on the results, Lasso seems like a good choice. It achieved a good balance between training fit and test performance. Additionally, it performs feature selection by setting coefficients of unimportant features to zero. This can improve model interpretability and potentially reduce overfitting.
- **Elastic Net:** Elastic Net offers a compromise between Ridge and Lasso. It can be useful if you suspect correlated features might be important but want to avoid setting some coefficients to zero entirely.

Conclusion

In conclusion, among the various regression methods explored in this assignment, LASSO and ElasticNet stand out as the chosen models for predicting student-faculty ratios in colleges based on the College dataset from the ISLR package.

Both LASSO and ElasticNet offer distinct advantages:

Feature Selection: LASSO, in particular, is well-suited for feature selection by driving certain coefficients to zero, effectively performing variable selection. This ability is crucial for identifying the most relevant predictors among a large set of potential variables, enhancing model interpretability and reducing overfitting.

Flexibility: ElasticNet combines the strengths of both LASSO and Ridge Regression by incorporating both L1 and L2 penalties, offering greater flexibility in handling different types of datasets. This hybrid approach enables ElasticNet to address multicollinearity while still performing feature selection.

Regularization: Both LASSO and ElasticNet provide regularization, which helps prevent overfitting and improves the generalization ability of the models. By penalizing the size of the coefficients, these methods promote simpler models that are less prone to noise in the data.

Model Performance: Despite their simplicity, LASSO and ElasticNet often achieve competitive performance compared to more complex regression techniques. In this assignment, they demonstrated reasonable predictive accuracy, as indicated by the root mean square error (RMSE) values on both the training and test datasets.

Therefore, for predicting student-faculty ratios in colleges, LASSO and ElasticNet emerge as preferred choices due to their ability to handle high-dimensional data, perform feature selection, and mitigate overfitting. These models offer a balance between simplicity, interpretability, and predictive accuracy, making them valuable tools for practical applications in regression analysis.

Appendix

```
library(ISLR)


college_dataset_ISLR<- College

head(college_dataset_ISLR)

str(college_dataset_ISLR)

#------------------

##Question 1

#------------------

set.seed(20353)

sample_size_75 <- floor(0.75 * nrow(college_dataset_ISLR))


trainingSet_Indices <- sample(seq_len(nrow(college_dataset_ISLR)), size = sample_size_75)

trainingSet_Data_75 <- college_dataset_ISLR[trainingSet_Indices, ]

testSet_Data_25 <- college_dataset_ISLR[-trainingSet_Indices, ]


trainingSet_Data_75_x <-model.matrix(S.F.Ratio ~ .,trainingSet_Data_75)[,-1]

testSet_Data_25_x <-model.matrix(S.F.Ratio ~ .,testSet_Data_25)


trainingSet_Data_75_y <- trainingSet_Data_75$S.F.Ratio

testSet_Data_25_y <- testSet_Data_25$S.F.Ratio


#------------------

## Ridge Regression

#------------------

#------------------

##Question 2

#------------------
```

```
library(glmnet)
set.seed(20353)
crossValidation_fit_glmnet_ridge <- cv.glmnet(trainingSet_Data_75_x, trainingSet_Data_75_y,
                    alpha = 0, nfolds = 10)
crossValidation_fit_glmnet_ridge
cat("Lambda.min:", crossValidation_fit_glmnet_ridge$lambda.min, "\n")
cat("Lambda.1se:", crossValidation_fit_glmnet_ridge$lambda.1se, "\n")

log(crossValidation_fit_glmnet_ridge$lambda.min)
log(crossValidation_fit_glmnet_ridge$lambda.1se)

#------------------
##Question 3
#------------------
plot(crossValidation_fit_glmnet_ridge)

#------------------
##Question 4
#------------------
ridgeModel_q4 <- glmnet(trainingSet_Data_75_x, trainingSet_Data_75_y,
           alpha = 0, lambda = crossValidation_fit_glmnet_ridge$lambda.min)
options(scipen = 999)
coefficients(ridgeModel_q4)

#------------------
##Question 5
#------------------
predicted_trainingSet_Data_75_y_ridge <- predict(ridgeModel_q4, newx = trainingSet_Data_75_x)
```

```
rmse_train_ridge <- sqrt(mean((trainingSet_Data_75_y -
predicted_trainingSet_Data_75_y_ridge)^2))

rmse_train_ridge

#library(Metrics)

#rmse(trainingSet_Data_75_y,predicted_trainingSet_Data_75_y_ridge)


#------------------

##Question 6

#------------------

predicted_testSet_Data_25_y_ridge <- predict(ridgeModel_q4, newx = testSet_Data_25_x)

rmse_test_ridge <- sqrt(mean((testSet_Data_25_y - predicted_testSet_Data_25_y_ridge)^2))

rmse_test_ridge




#------------------

## LASSO

#------------------

#------------------

##Question 7

#------------------

set.seed(20353)

crossValidation_fit_glmnet_lasso <- cv.glmnet(trainingSet_Data_75_x, trainingSet_Data_75_y,

                              alpha = 1, nfolds = 10)


cat("Lambda.min:", crossValidation_fit_glmnet_lasso$lambda.min, "\n")

cat("Lambda.1se:", crossValidation_fit_glmnet_lasso$lambda.1se, "\n")


log(crossValidation_fit_glmnet_lasso$lambda.min)

log(crossValidation_fit_glmnet_lasso$lambda.1se)
```

```
#------------------
##Question 8
#------------------
plot(crossValidation_fit_glmnet_lasso)


#------------------
##Question 9
#------------------
lassoModel_q9 <- glmnet(trainingSet_Data_75_x, trainingSet_Data_75_y,
              alpha = 1, lambda = crossValidation_fit_glmnet_lasso$lambda.min)
options(scipen = 999)
coefficients(lassoModel_q9)


#------------------
##Question 10
#------------------
predicted_trainingSet_Data_75_y_lasso <- predict(lassoModel_q9, newx = trainingSet_Data_75_x)
rmse_train_lasso <- sqrt(mean((trainingSet_Data_75_y -
predicted_trainingSet_Data_75_y_lasso)^2))
rmse_train_lasso
#library(Metrics)
#rmse(trainingSet_Data_75_y,predicted_trainingSet_Data_75_y_lasso)


#------------------
##Question 11
#------------------
predicted_testSet_Data_25_y_lasso <- predict(lassoModel_q9, newx = testSet_Data_25_x)
rmse_test_lasso <- sqrt(mean((testSet_Data_25_y - predicted_testSet_Data_25_y_lasso)^2))
```

```
rmse_test_lasso


#------------------

## ElasticNet Let Alpha = 0.5

#------------------

#------------------

##Question 12

#------------------

set.seed(20353)

crossValidation_fit_glmnet_elasticNet <- cv.glmnet(trainingSet_Data_75_x, trainingSet_Data_75_y,

                              alpha = 0.5, nfolds = 10)

cat("Lambda.min:", crossValidation_fit_glmnet_elasticNet$lambda.min, "\n")

cat("Lambda.1se:", crossValidation_fit_glmnet_elasticNet$lambda.1se, "\n")

log(crossValidation_fit_glmnet_elasticNet$lambda.min)

log(crossValidation_fit_glmnet_elasticNet$lambda.1se)


plot(crossValidation_fit_glmnet_elasticNet)


elasticNetModel_q12 <- glmnet(trainingSet_Data_75_x, trainingSet_Data_75_y,

               alpha = 0.5, lambda = crossValidation_fit_glmnet_elasticNet$lambda.min)

options(scipen = 999)

coefficients(elasticNetModel_q12)


predicted_trainingSet_Data_75_y_elasticNet <- predict(elasticNetModel_q12, newx = trainingSet_Data_75_x)

rmse_train_elasticNet <- sqrt(mean((trainingSet_Data_75_y - predicted_trainingSet_Data_75_y_elasticNet)^2))

rmse_train_elasticNet
```

```
predicted_testSet_Data_25_y_elasticNet <- predict(elasticNetModel_q12, newx = testSet_Data_25_x)

rmse_test_elasticNet <- sqrt(mean((testSet_Data_25_y - predicted_testSet_Data_25_y_elasticNet)^2))

rmse_test_elasticNet


#------------------

##Question 14

#------------------

set.seed(20353)

stepwise_model_q14 <- step(lm(S.F.Ratio ~ ., data = trainingSet_Data_75), direction = 'both')

summary(stepwise_model_q14)


stepwise_model_q14_prediction_train <- predict(stepwise_model_q14, newdata = trainingSet_Data_75)

rmse_stepwise_model_q14_train <- sqrt(mean((trainingSet_Data_75$S.F.Ratio - stepwise_model_q14_prediction_train)^2))

rmse_stepwise_model_q14_train


stepwise_model_q14_prediction_test <- predict(stepwise_model_q14, newdata = testSet_Data_25)

rmse_stepwise_model_q14_test <- sqrt(mean((testSet_Data_25$S.F.Ratio - stepwise_model_q14_prediction_test)^2))

rmse_stepwise_model_q14_test
```

References

- Dev, S. (2023b, January 20). *Ridge regression and lasso regression: A beginner’s guide*. Medium. https://medium.com/@devsachin0879/ridge-regression-and-lasso-regression-a-beginners-guide-b3e33c77678

- Taylor, C. (2024, March 14). *How to use elastic net regression*. Medium. https://towardsdatascience.com/how-to-use-elastic-net-regression-85a6a393222b