Assignment 4: Ridge & LASSO Regression for College dataset

Yash S

2024 – 2 – 9

Northeastern University: College of Professional Studies

Introduction

Introduction

The objective of this report is to perform regularization techniques on the College dataset from the ISLR package and build a linear regression models using Ridge & LASSO regression, to reduce overfitting while predicting the Graduation Rate (Grad.Rate) of an institute. This analysis follows a structed and methodical approach, such as data pre-processing, regularization parameter lambda evaluation, model training, performance evaluation, feature selection, diagnostics and model comparison. The goal of this report is to compare the performance metrics and meaningfully interpret the visualizations to provide with key insights, that will help us identify the significant predictors in different models.

Dataset Overview

The dataset comprises of 777 university with 18 variables which captures the various institutional characteristics of these universities. Our target variable here is the "Grad.Rate" which contains information of the Graduation Rate of that particular institution. We found that the data did not have missing values indicating that it is a complete dataset ready for analysis. The independent variables include numerical attributes that are related to enrolled students, qualifications of faculty, tuition fees and expenditure per student.

Key Variables:

1. **Nominal (Categorical variable):-**
   a) **Private:** This variable informs us whether the institute is public or private.
2. **Ordinal (Categorical variable):-**
   a) **Top10perc:** Percentage of students that are from the top 25% of their high school class.
   b) **Top25perc:** Percentage of students that are from the top 25% of their high school class.
   c) **PhD:** Percentage of Faculty with PhD in that particular institute.
   d) **Terminal:** Percentage of Faculty with Terminal Degree in that particular institute.
   e) **perc.alumni:** Percentage of alumni who donate.
   f) **Grad.Rate [Response Variable):** Graduation rate of that particular institute.
3. **Discrete:-**
   a) **Apps:** The total number of applications received by that particular institute.
   b) **Accept:** The total number of applications accepted by that particular institute.
   c) **Enroll:** The total number of students enrolled in that institute.
   d) **F.Undergrad:** The total number of full-time undergraduates in that particular institute.
   e) **P.Undergrad:** The total number of part-time undergraduates in that particular institute.
4. **Continuous:-**
   a) **Outstate:** Cost for out of the state tuition cost for that institute.
   b) **Expend:** Amount of expenditure per student for instructional purposes by that institute.
   c) **S.F.Ratio:** Student to faculty ratio of that institute.

d) **Room.Board:** Estimated cost of the room and the board.
e) **Books:** Estimated yearly cost of the books.
f) **Personal:** Estimated personal expenses per year.

Data Analysis

Explanatory Descriptive Analysis:

1. **Descriptive Statistics of Key Variables:**

```
Private        Apps            Accept          Enroll         Top10perc       Top25perc        F.Undergrad
No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139
Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992
          Median : 1558   Median : 1110   Median : 434   Median :23.00   Median : 54.0   Median : 1707
          Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700
          3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005
          Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643
 P.Undergrad          Outstate       Room.Board        Books           Personal         PhD
Min.   :    1.0   Min.   : 2340   Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
Median :  353.0   Median : 9990   Median :4200   Median : 500.0   Median :1200   Median : 75.00
Mean   :  855.3   Mean   :10441   Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
Max.   :21836.0   Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
   Terminal        S.F.Ratio       perc.alumni        Expend         Grad.Rate
Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186   Min.   : 10.00
1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00
Median : 82.0   Median :13.60   Median :21.00   Median : 8377   Median : 65.00
Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660   Mean   : 65.46
3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00
Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233   Max.   :118.00
```

```
Standard Deviation for Outstate: 4023.016
```

```
Standard Deviation for Expend: 5221.768
```

a. Private [Nominal]

There are around **212 public institutes** and **565 private institutes**.

b. Apps [Discrete (Count)]

The average number of applications received by the institutes are **3,002** with a median of **1,558** with lowest number of applications received of **81** and highest number of applications received is **48,094.**

c. Accept [Discrete (Count)]

The average number of accepted applications by the institutes are **2,019** with a median of **1,110** with lowest number of accepted applications by an institute at **72** and highest number of accepted applications by an institute is **26,330.**

d. Enroll [Discrete (count)]

The average number of students enrolled at an institute is **780** with lowest number of students enrolled at an institute at **35** and highest number students enrolled at an institute at **6,392**.

e. Top10perc [Ordinal (%)]

The average percentage of students in the top 10% of their high school is **27.56%** with a minimum of **1%** and a maximum of **96%**.

f. Top25perc [Ordinal (%)]

The average percentage of students in the top 25% of their high school is **55.8%** with a minimum of **9%** and a maximum of **100%** this ensure that the data is consistent.

g. F.Undergrad [Discrete (Count)]
The number of full-time undergraduates has a range from **139 to 31,643** students with a mean of **3,700** and median of **1707**.

h. P.Undergrad [Discrete (Count)]
The number of full-time undergraduates has a range from **1** to **21,836** students with a mean of **855.3** and median of **353**.

i. Outstate [Continuous ($)]
The average of out of state tuition is **$10,441** with a standard deviation of **$4,032**, with the cheapest tuition at **$2,340** and the most expensive tuition at **$21,700**.

j. Room.Board [Continuous ($)]
The average cost of room and board is **$4,358** with a minimum of **$1,780** and maximum is **$8,142**.

k. Books [Continuous ($)]
The average cost of books per year is **$549.4** with a minimum of **$96** and maximum is **$2,340**.

l. Personal [Continuous ($)]
The average cost of personal expenses is **$1,341** with a minimum of **$250** and maximum is **$6,800.**

m. PhD [Ordinal (%)]
The average of faculty percentage with PhD in an institute is **73%**, with a maximum value of **103%** indicating potential data inconsistency.

n. Terminal [Ordinal (%)]
The average of faculty percentage with a Terminal degree in an institute is **80%**, with a maximum value of **100%** indicating consistent data as compared to PhD faculty.

o. S.F.Ratio [Continuous (Ratio)]
The student-faculty ratio has a mean of **14.09** with a median of **13.60**.

p. perc.alumni [Ordinal (%)]
The average percentage of alumni who donate to their particular institutes are **22.74%** with a minimum of **0%** and maximum of **64%**.

q. Expend [Continuous ($)]
Every institute have an average expenditure of **$9,660** and a standard deviation of **$5,222**, with a lowest spend of **$3,186** and highest spend of **$56,233**.

r. Grad.Rate [Ordinal (%)] ***Response Variable***

The average graduation rate of an institute is **65.46%**, with a maximum value of **118%** indicating potential data inconsistency and with a minimum value of **10%**.

Regression Analysis:

a) Data Partitioning

The dataset was split into 70% of training data and 30% of testing data, this was done using the “caret” package ensuring a random split. A seed (123) was set to ensure data reproducibility. This also converted the Private column as a binary column named PrivateYes which has 0 for Public Institutes and 1 for Private Institutes.

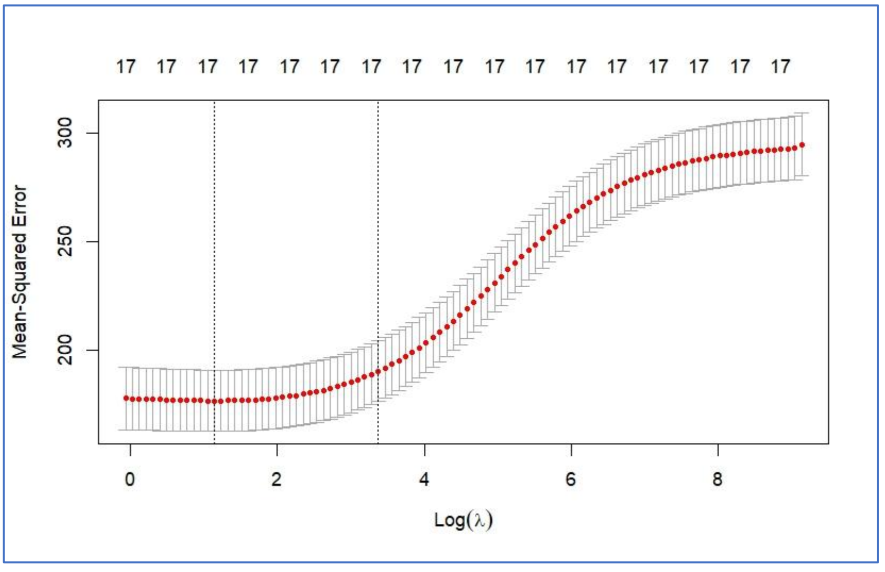
b) Ridge Regression

The Ridge Regression applies the L2 regularization, this shrinks the coefficients to a low value but does not remove them.

```
> lambda_min_ridge; lambda_1se_ridge
[1] 3.126268
[1] 29.15568
```

Lambda.min: This value represents that regularization parameter that minimizes the average of the cross-validation error. In this case at lambda.min is equal to **3.126**.
Lambda.1se: This value represents the largest regularization parameter within one standard error that minimizes the average of the cross-validation error. In this case lambda.1se is equal to **29.156**.
While lambda.min has lower regularization strength as compared to lambda.1se, the models trained with lambda.min will have the lowest possible error on the training data but the models trained with lambda.1se might generalise new data better as it employs a higher degree of regularization.

Visualization:
When we plot this function and interpret the graph, we can see a trade-off between the Mean-Square Error and the logarithm of the regularization parameter [Log(λ)], the two vertical dashes line are the lambda.min [approx. log(3.126) ≈ 1.5] and lambda.1se [approx. log(29.15568) ≈ 3.4], the error initially stays stable but begins to rise sharply after a point indicating over regularization.



Graph showing relationship between MSE and Log(λ)

Model Evaluation:
Since we aim for the lowest possible error on the training data, we select the lambda.min as the optimal value of lambda to fit a ridge regression model, following are the interpretation of the variables selected.

```
> coef(ridge_fit)
18 x 1 sparse Matrix of class "dgCMatrix"
                        s0
(Intercept) 33.01241331969
PrivateYes   4.96681936367
Apps         0.00050233995
Accept       0.00027582617
Enroll       0.00042144183
Top10perc    0.09909438566
Top25perc    0.10342512056
F.Undergrad -0.00008808234
P.Undergrad -0.00132793538
Outstate     0.00066804291
Room.Board   0.00180445179
Books       -0.00164519506
Personal    -0.00186418198
PhD          0.02710266570
Terminal     0.01068751335
S.F.Ratio    0.14568801562
perc.alumni  0.22846419273
Expend      -0.00021213114
```

The regression equation for the above is:

$$Grad.Rate=33.0124+4.9668\times PrivateYes+0.0005\times Apps+0.0003\times Accept+0.0002\times Enroll+0.0990\times Top10perc+0.1034\times Top25perc-0.0001\times F.Undergrad-0.0013\times P.Undergrad-0.0006\times Outstate+0.0018\times Room.Board-0.0016\times Books-0.0019\times Personal+0.0271\times PhD+0.0107\times Terminal+0.1457\times S.F.Ratio+0.2285\times perc.alumni-0.0002\times Expend$$

We can interpret this equation as the follows

i) <u>Intercept (33.0124):</u>

This value represents the baseline when all the other predictors are zero. This means the baseline graduation rate is **33.0124%.**

ii) <u>PrivateYes (4.9668)</u>

Private institutes have slightly higher graduation rate than public institutes, on an average student of private institutions are about **4.9668%** more likely to graduate as compared to their counterparts at public institutions.

iii) <u>Apps (0.0005)</u>

Each additional application received the graduation rate only slightly increases by **0.0005%.**

iv) <u>Accept (0.0003)</u>

Each additional acceptance received the graduation rate only slightly increases by **0.0003%.**

v) <u>Enroll (0.0004)</u>

Each additional enrolment received the graduation rate only slightly increases by **0.0004%.**

vi) <u>Top10perc (0.0990)</u>

Each percentage point increase in the students from the top 10% of their high school is associated with a **0.099%** of higher graduation rate.

vii) <u>Top25perc (0.1034)</u>

Each percentage point increase in the students from the top 25% of their high school is associated with a **0.1034%** of higher graduation rate.

viii) <u>F.Undergrad (-0.0001)</u>

Each additional full-time student slightly decreases the graduation rate by **-0.0001%.**

ix) <u>P.Undergrad (-0.0013)</u>

Each additional part-time student slightly decreases the graduation rate by **-0.0013%.**

x) Outstate (0.0006)

Each additional dollar of out-of-state tuition slightly increases the rate of graduation by **0.0006%.**

xi) Room.Board(0.0018)

Each additional dollar spent on rooms and boards increases the graduation rate by **0.018%.**

xii) Books (-0.0016)

Each additional dollar spent on books decreases the rate of graduation by **-0.016%.**

xiii) Personal (-0.0019)

Each additional dollar spent on personal expenses decreases the rate of graduation by **-0.019%.**

xiv) PhD (0.0271)

Each percentage point increase in the faculties with PhD is associated with a **0.0271%** increase in the graduation rates.

xv) Terminal (0.0107)

Each percentage point increase in the faculties with Terminal Degree is associated with a **0.0107%** increase in the graduation rates.

xvi) S.F.Ratio (0.1457)

Each unit increase in the student-faculty ratio increases the graduation rate by **0.1457%.**

xvii) perc.alumni (0.2285)

Each percentage point increase in the donating alumni is associated with an increase of **0.2285%** higher rate of graduation.

xviii) Expend (-0.0002)

Each additional dollar spent per student slightly decreases the graduation rate by **-0.0002%.**

Performance Evaluation:

```
> rmse_train_ridge; rmse_test_ridge
[1] 12.98732
[1] 12.04532
> r2_train_ridge; r2_test_ridge
[1] 0.4262697
[1] 0.5104268
```

i) Root Mean Squared Error (RMSE):

- Training: **12.987**
- Testing: **12.045**

ii) Coefficient of Determination (R^2):

- Training: **0.4263**
- Testing: **0.5104**

Based on the RMSE value and the R^2 value we can see that the model performs poorly on the training data but we can see that it performs well on the testing data which is unseen data. This means that the model generalizes well to unseen data and is not overfitting. Both RMSE and R^2 complement each other in this evaluation and we can notice a trend that while RMSE decreases in testing data the R^2 increases, this indicates similar level of explanatory power on both sets of data. The R^2 values explains around **42% variance** in training data and **51% variance** in testing data this suggests a decent fit.

c) LASSO Regression

The LASSO (Least Absolute Shrinkage and Selection Operator) Regression applies the L1 regularization, this shrinks the coefficients to zero, this effectively selects only the most important features and drops the rest.

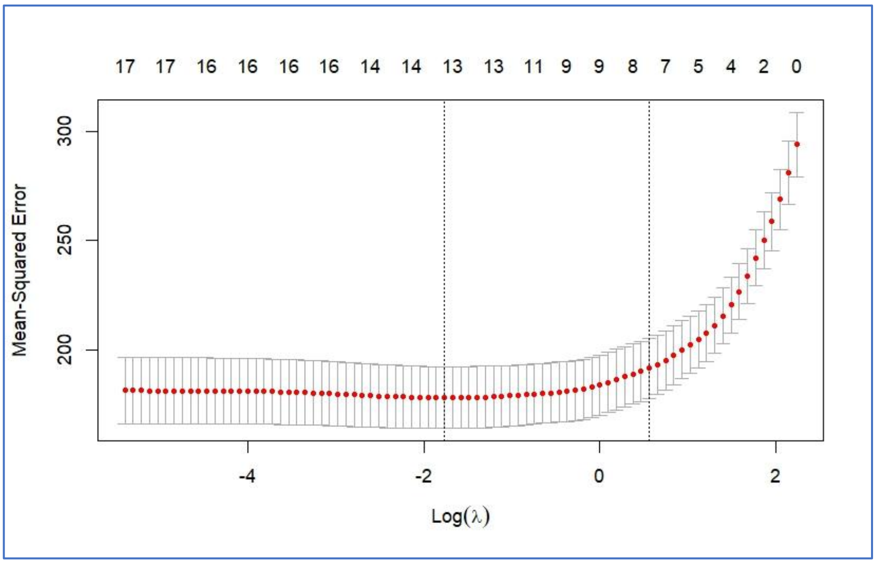
```
> lambda_min_lasso; lambda_1se_lasso
[1] 0.1707654
[1] 1.747837
```

Lambda.min (0.1708): This value of lambda minimizes the MSE on the cross-validated data. In this case at lambda.min is equal to **0.1708**. This value results in the most complex model with best performance on the training data

Lambda.1se (1.7478): This value of lambda is within one standard error that minimizes the average of the cross-validation error. In this case lambda.1se is equal to **1.7478**. This value results in a simpler model that balances the bias and the variance, reducing overfitting. While lambda.min has lower regularization strength as compared to lambda.1se, the models trained with lambda.min will have the lowest possible error on the training data but the models trained with lambda.1se might generalise new data better as it employs a higher degree of regularization, which will help prevent overfitting.

Visualization:

When we plot this function and interpret the graph, we can see a trade-off between the Mean-Square Error and the logarithm of the regularization parameter [Log(λ)], the two vertical dashes line are the lambda.min [approx. log(0.1707654) ≈-1.77] and lambda.1se [approx. log(1.747837) ≈ 0.56], as the log(λ) increases the mean-squared error stays relatively stable but begins to rise sharply after a point indicating that increasing regularization strength can cause the model to underfit. This means after log(1.747837) model becomes too simple to detect and capture the underlying patterns in the data effectively



Graph showing relationship between MSE and Log(λ)

Model Evaluation:

Since we aim for the lowest possible error on the training data, we select the lambda.min as the optimal value of lambda to fit a Least Absolute Shrinkage and Selection Operator (LASSO) regression model, following are the interpretation of the variables selected.

```
> coef(lasso_fit)
18 x 1 sparse Matrix of class "dgCMatrix"
                       s0
(Intercept) 31.9744173484
PrivateYes   5.1735026709
Apps         0.0008338388
Accept       .
Enroll       .
Top10perc    0.0770066238
Top25perc    0.1179043005
F.Undergrad  .
P.Undergrad -0.0014836187
Outstate     0.0007961264
Room.Board   0.0018215790
Books       -0.0006521375
Personal    -0.0017383176
PhD          0.0172825871
Terminal     .
S.F.Ratio    0.1539026536
perc.alumni  0.2633567143
Expend      -0.0003240021
```

The regression equation for the above is:

$$\text{Grad.Rate}=31.9744+5.1735\times\text{PrivateYes}+0.0008\times\text{Apps}+0.0770\times\text{Top10perc}+0.1179\times\text{Top25perc}$$
$$-0.0015\times\text{P.Undergrad}+0.0008\times\text{Outstate}+0.0018\times\text{Room.Board}-0.0007\times\text{Books}$$
$$-0.0017\times\text{Personal}+0.0173\times\text{PhD}+0.1539\times\text{S.F.Ratio}+0.2634\times\text{perc.alumni}-0.0003\times\text{Expend}$$

We can interpret this equation as the follows

i) <u>Intercept (31.9744)</u>

This value represents the baseline when all the other predictors are zero. This means the baseline graduation rate is **31.9744%.**

ii) <u>PrivateYes (5.1735)</u>

Private institutes have slightly higher graduation rate than public institutes, on an average student of private institutions are about **5.1735%** more likely to graduate as compared to their counterparts at public institutions.

iii) <u>Apps (0.0008)</u>

Each additional application received the graduation rate only slightly increases by **0.0008%.**

iv) <u>Top10perc (0.0770)</u>

Each percentage point increase in the students from the top 10% of their high school is associated with a **0.077%** of higher graduation rate.

v) <u>Top25perc (0.1179)</u>

Each percentage point increase in the students from the top 25% of their high school is associated with a **0.1179%** of higher graduation rate.

vi) <u>P.Undergrad (-0.0015)</u>

Each additional part-time student slightly decreases the graduation rate by **-0.0015%.**

vii) <u>Outstate (0.0008)</u>

Each additional dollar of out-of-state tuition slightly increases the rate of graduation by **0.0008%.**

viii) <u>Room.Board(0.0018)</u>

Each additional dollar spent on rooms and boards increases the graduation rate by **0.018%.**

ix) <u>Books (-0.0007)</u>

Each additional dollar spent on books decreases the rate of graduation by **-0.007%.**

x) <u>Personal (-0.0017)</u>

Each additional dollar spent on personal expenses decreases the rate of graduation by **-0.017%.**

xi) <u>PhD (0.0173)</u>

Each percentage point increase in the faculties with PhD is associated with a **0.0173%** increase in the graduation rates.

xii) <u>S.F.Ratio (0.1539)</u>

Each unit increase in the student-faculty ratio increases the graduation rate by **0.1539%.**

xiii) <u>perc.alumni (0.2634)</u>

Each percentage point increase in the donating alumni is associated with an increase of **0.2634%** higher rate of graduation.

xiv) <u>Expend (-0.0003)</u>

Each additional dollar spent per student slightly decreases the graduation rate by **-0.0003%.**

The following intercepts are set to zero as they do not significantly contribute in the model under the chosen regularization method.

i. <u>Accept</u>
ii. <u>Enroll</u>
iii. <u>F.Undergrad</u>
iv. <u>Terminal</u>

<u>Performance Evaluation:</u>

```
> rmse_train_lasso; rmse_test_lasso
[1] 12.93813
[1] 11.98879
> # Output R² values
> r2_train_lasso; r2_test_lasso
[1] 0.4306077
[1] 0.5150116
```

i) Root Mean Squared Error (RMSE):

- <u>Training:</u> **12.938**
- <u>Testing:</u> **11.989**

ii) Coefficient of Determination (R^2):

- <u>Training:</u> **0.4306**
- <u>Testing:</u> **0.5150**

Based on the <u>RMSE value and the R^2 value</u> we can see that the model performs <u>slightly poor on the training data</u> but we can see that it performs <u>very well on the testing data</u> which is unseen data. This means that the model generalizes well to unseen data and is not overfitting. Both RMSE and R^2 complement each other in this evaluation and we can notice a trend that while RMSE decreases in testing data the R^2 increases, this indicates similar level of explanatory power on both sets of data. The R^2 values explains around **43% variance** in training data and **52% variance** in testing data this suggests a decent fit.

d) Step-wise Feature Selection

The method for feature selection was step-wise selection which uses AIC-based forward and backward selection to select the most influential predictors.

```
Call:
lm(formula = Grad.Rate ~ Private + Apps + Top25perc + P.Undergrad +
    Outstate + Room.Board + Personal + perc.alumni + Expend,
    data = train_set)

Residuals:
    Min      1Q  Median      3Q     Max
-51.958  -7.430  -0.454   6.877  51.155

Coefficients:
              Estimate Std. Error t value             Pr(>|t|)
(Intercept) 33.4595503  3.1611031  10.585 < 0.0000000000000002 ***
PrivateYes   5.2645566  1.9033280   2.766              0.00587 **
Apps         0.0010436  0.0002266   4.605           0.00000517 ***
Top25perc    0.1755962  0.0378963   4.634           0.00000452 ***
P.Undergrad -0.0016522  0.0004142  -3.988           0.00007574 ***
Outstate     0.0008413  0.0002711   3.104              0.00201 **
Room.Board   0.0019041  0.0006840   2.784              0.00556 **
Personal    -0.0018738  0.0009133  -2.052              0.04070 *
perc.alumni  0.2832253  0.0589896   4.801           0.00000205 ***
Expend      -0.0004470  0.0001586  -2.819              0.00500 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.08 on 536 degrees of freedom
Multiple R-squared:  0.4287,    Adjusted R-squared:  0.4191
F-statistic: 44.68 on 9 and 536 DF,  p-value: < 0.00000000000000022
```

The regression equation for the above model is as follows:

Graduation Rate=33.46+5.26·PrivateYes+0.0104·Apps+0.176·Top25perc−0.0016·P.Undergrad +0.000841·Outstate+0.00402·Room.Board−0.00187·Personal+0.282·perc.alumni −0.000447·Expend

We can interpret the logistic regression coefficients as follows:

We can interpret this equation as the follows

i) <u>Intercept (33.46):</u>

This value represents the baseline when all the other predictors are zero. This means the baseline graduation rate is **33.46%.**

<u>PrivateYes (5.26)</u>

Private institutes have slightly higher graduation rate than public institutes, on an average student of private institutions are about **5.26%** more likely to graduate as compared to their counterparts at public institutions.

ii) <u>Apps (0.0104)</u>

Each additional application received the graduation rate only slightly increases by **0.0104%.**

iii) <u>Top25perc (0.176)</u>

Each percentage point increase in the students from the top 25% of their high school is associated with a **0.176%** of higher graduation rate.

iv) <u>P.Undergrad (-0.0016)</u>

Each additional part-time student slightly decreases the graduation rate by **-0.0016%.**

v) <u>Outstate (0.0008413)</u>

Each additional dollar of out-of-state tuition slightly increases the rate of graduation by **0.0008413%.**

vi) <u>Room.Board(0.00402)</u>

Each additional dollar spent on rooms and boards increases the graduation rate by **0.00402%.**

vii) <u>Personal (-0.00187)</u>

Each additional dollar spent on personal expenses decreases the rate of graduation by **-0.00187%.**

viii) <u>perc.alumni (0.282)</u>

Each percentage point increase in the donating alumni is associated with an increase of **0.282%** higher rate of graduation.

ix) <u>Expend (-0.000447)</u>

Each additional dollar spent per student slightly decreases the graduation rate by **-0.000447%.**

```
> step_summary$r.squared; AIC(model_step); BIC(model_step)
[1] 0.4286648
[1] 4369.056
[1] 4416.385
```

The AIC value for the step-wise model is **4369.056**. The BIC value for the step-wise model is **4416.385.** R^2 value is lower **0.42** than the previous models this warrants further investigation of the model and the influence of unusual observations.

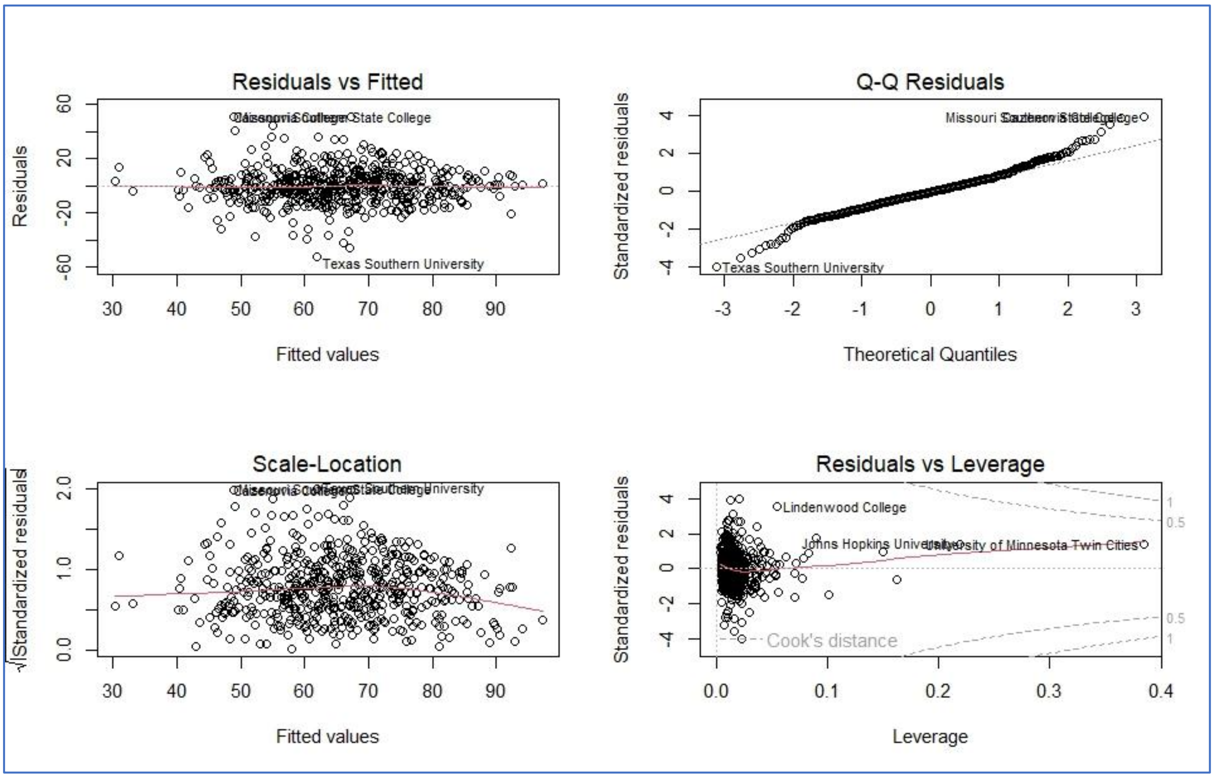
e) Diagnostic Plots and Refinement

i. Residual vs Fitted: There is a visible trend of randomly scattered points and outliers suggesting potential issues with related to homoscedasticity and some deviation from linearity.

ii. Q-Q Residuals: The Q-Q plot compares the standard residuals; the graph shows deviation from the line at the tails indicate that the residuals are not entirely perfectly normally distributed.

iii. Scale Location: We can see the square root of the standardized residuals of the points are equally spread and not funnel shaped, but the pattern suggests outliers. The pattern suggests homoscedasticity and that the outliers could significantly impact the model.

iv. Residuals vs Leverage: The points with high leverage are potential outliers as shown by Cook's Distance; these impact the regression model significantly.

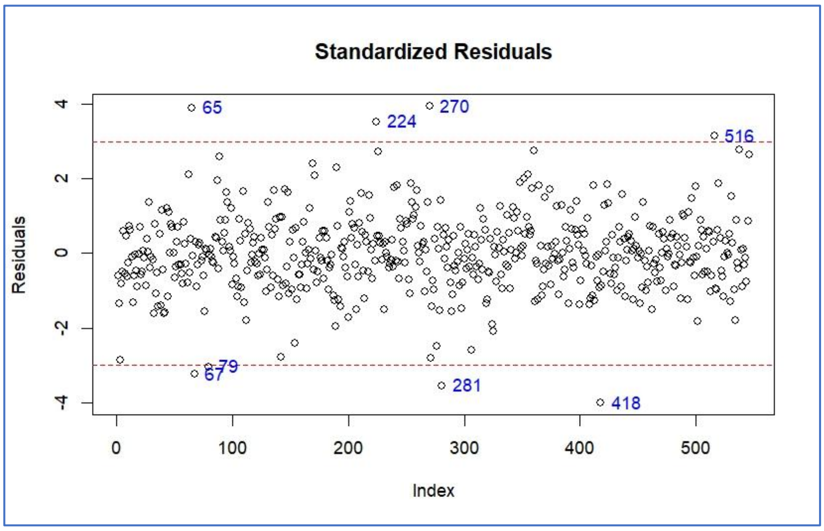


VIF for all predictors that was **under 5**, indicating there is some correlation between the predictors but they are not severe enough to cause any issues in the regression model.

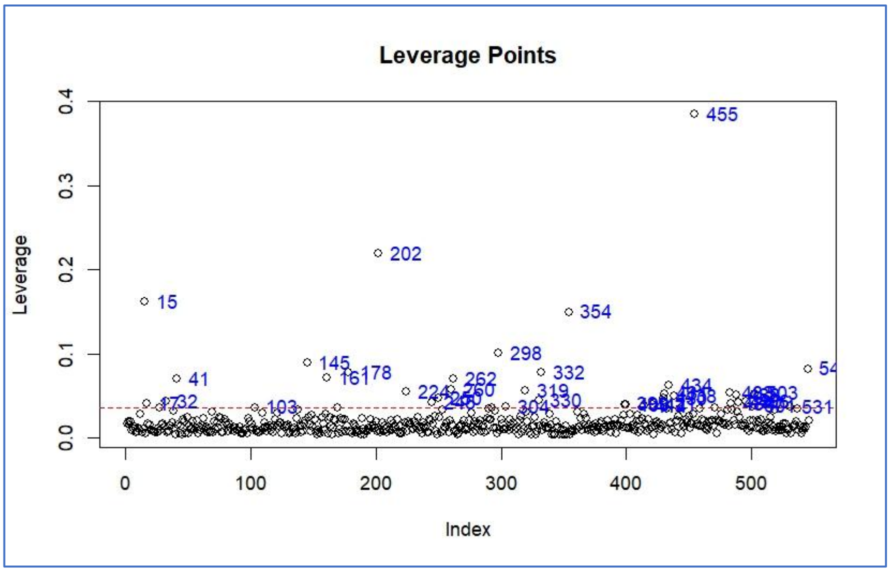
```
> vif(model_step)
    Private        Apps   Top25perc P.Undergrad    Outstate  Room.Board    Personal perc.alumni      Expend
   2.274442    2.108313    1.760413    1.462850    3.664375    1.786960    1.229228    1.725266    2.092018
```

Handling the Unusual Observations for improved linear regression modelling:

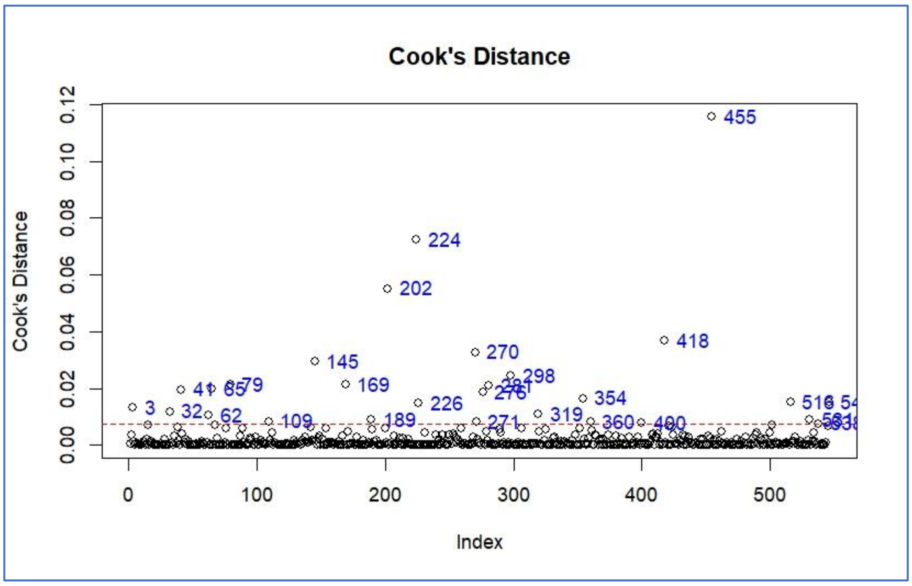
1. Outliers (8): **Eight values** with high standardized residuals (z-score) were calculated from the Step-Wise Regression Model, these values were recognized and removed as they were above the threshold value of 3. These observations do not fit the genral pattern and prove to be a poor fit for these points, such points are represented by unusual financial structure, tuition, enrollment characteristics that are different from majority of the colleges.



2. <u>High-Leverage Points(38):</u> The Hat values were calculated and **38 observations** were recognized to be significantly higher than the average leverage values, this is two times the average leverage values. This is calculated by measuring the influence of the **38 individual data-points** on the fitted values. These include Harvard University, MIT, Boston University and many other large institutes which have extreme predictor values.



3. <u>Influencial Points (28):</u> The Cooks Distance showed **28 data-points** to be influencial points, this is calculated my measuring the influence of deleteing the given observations. These are identified by checking the values greater than 4 divided by the number of observations. These points are important to remove as they might be both outliers and high-leverage points.



4. **56 points unusual observations** warrants the removal of all the unsual observations and re-running the model once again.

After removing the outliers and re-running the model resulted in increased **Adjusted R^2 value** from <u>0.4191</u> to <u>0.4942</u> as we can see below.

```
> step_summary$adj.r.squared; cleaned_summary$adj.r.squared
[1] 0.4190715
[1] 0.4942892
```

```
Call:
lm(formula = Grad.Rate ~ Private + Apps + Top25perc + P.Undergrad +
    Outstate + Room.Board + Personal + perc.alumni + Expend,
    data = train_set_cleaned)

Residuals:
    Min      1Q  Median      3Q     Max
-36.032  -7.214  -0.320   6.477  32.970

Coefficients:
              Estimate Std. Error t value             Pr(>|t|)
(Intercept) 34.2437836  3.0131775  11.365 < 0.0000000000000002 ***
PrivateYes   5.0266999  1.8053398   2.784              0.00558 **
Apps         0.0013595  0.0002753   4.938        0.00000109306 ***
Top25perc    0.1495960  0.0351798   4.252        0.00002543696 ***
P.Undergrad -0.0016624  0.0007164  -2.320              0.02073 *
Outstate     0.0013557  0.0002641   5.134        0.00000041285 ***
Room.Board   0.0018730  0.0006374   2.939              0.00345 **
Personal    -0.0022726  0.0009649  -2.355              0.01892 *
perc.alumni  0.3144223  0.0530118   5.931        0.00000000576 ***
Expend      -0.0010721  0.0002310  -4.641        0.00000447659 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.91 on 480 degrees of freedom
Multiple R-squared:  0.5036,    Adjusted R-squared:  0.4943
F-statistic: 54.11 on 9 and 480 DF,  p-value: < 0.00000000000000022
```

Regression equation for the above model is

Grad.Rate=31.9744+5.1735×PrivateYes+0.0008×Apps+0.0770×Top10perc+0.1179×Top25perc −0.0015×P.Undergrad+0.0008×Outstate+0.0018×Room.Board−0.0007×Books−0.0017×Personal+0.0173×PhD+0.1539×S.F.Ratio+0.2634×perc.alumni−0.0003×Expend

f) All-subset Regression

After dealing with the outliers and refining the predictor values, an all-subset regression was used to confirm that only variables that make sense to the model are chosen.

```
Selection Algorithm: exhaustive
         PrivateYes Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books
1  ( 1 ) " "        " "  " "    " "    " "       " "       " "         " "         "*"      " "        " "
2  ( 1 ) " "        " "  " "    " "    " "       " "       " "         " "         "*"      " "        " "
3  ( 1 ) " "        " "  " "    " "    " "       "*"       " "         " "         "*"      " "        " "
4  ( 1 ) " "        " "  " "    " "    " "       "*"       " "         " "         "*"      " "        " "
5  ( 1 ) " "        " "  " "    " "    " "       "*"       " "         " "         "*"      "*"        " "
6  ( 1 ) "*"        "*"  " "    " "    " "       "*"       " "         " "         "*"      " "        " "
7  ( 1 ) "*"        "*"  " "    " "    " "       "*"       " "         " "         "*"      "*"        " "
8  ( 1 ) "*"        "*"  " "    " "    " "       "*"       " "         " "         "*"      "*"        " "
9  ( 1 ) "*"        "*"  " "    " "    " "       "*"       " "         "*"         "*"      "*"        " "
         Personal PhD Terminal S.F.Ratio perc.alumni Expend
1  ( 1 ) " "      " " " "      " "       " "         " "
2  ( 1 ) " "      " " " "      " "       "*"         " "
3  ( 1 ) " "      " " " "      " "       "*"         " "
4  ( 1 ) " "      " " " "      " "       "*"         "*"
5  ( 1 ) " "      " " " "      " "       "*"         "*"
6  ( 1 ) " "      " " " "      " "       "*"         "*"
7  ( 1 ) " "      " " " "      " "       "*"         "*"
8  ( 1 ) "*"      " " " "      " "       "*"         "*"
9  ( 1 ) "*"      " " " "      " "       "*"         "*"
```

```
> which.min(reg_summary$cp)
[1] 9
> which.max(reg_summary$adjr2)
[1] 9
```

As we can see from the above all-subset regression that all the variables chosen in the step-wise feature selection significantly contribute to the best model with minimum Mallow's Cp and maximum Adjusted R^2 is the 9^{th} model which is one and the same as the step-wise regression model.

g) Model Evaluation and Performance

1. Ridge Regression Model VS LASSO Regression Model

a) Ridge Regression Model

```
> rmse_train_ridge; rmse_test_ridge
[1] 12.98732
[1] 12.04532
> r2_train_ridge; r2_test_ridge
[1] 0.4262697
[1] 0.5104268
```

i. Root Mean Squared Error (RMSE):

- Training: **12.987**
- Testing: **12.045**

ii. Coefficient of Determination (R^2):

- Training: **0.4263**
- Testing: **0.5104**

b) LASSO Regression Model

```
> rmse_train_lasso; rmse_test_lasso
[1] 12.93813
[1] 11.98879
> # Output R² values
> r2_train_lasso; r2_test_lasso
[1] 0.4306077
[1] 0.5150116
```

i. Root Mean Squared Error (RMSE):

- Training: **12.938**
- Testing: **11.989**

ii. Coefficient of Determination (R^2):

- Training: **0.4306**
- Testing: **0.5150**

CONCLUSION

In terms of the RMSE the Ridge Regression Model **(12.045)** has lower value than the LASSO Regression Model **(12.989).** This indicates that the ridge regression model has better predictive accuracy. However, the LASSO regression model **(0.515)** has slightly higher R^2 value on the testing data than the ridge regression model **(0.5104)** suggesting the LASSO explains the variance slightly better.

This outcome is as expected because the Ridge Regression handles multicollinearity better by shrinking the coefficients leading to improved predictions and the LASSO Regression performs feature selection by setting some coefficients to zero, resulting in simpler model.

2. **Step-Wise Regression Model VS Cleaned Step-Wise Regression Model**

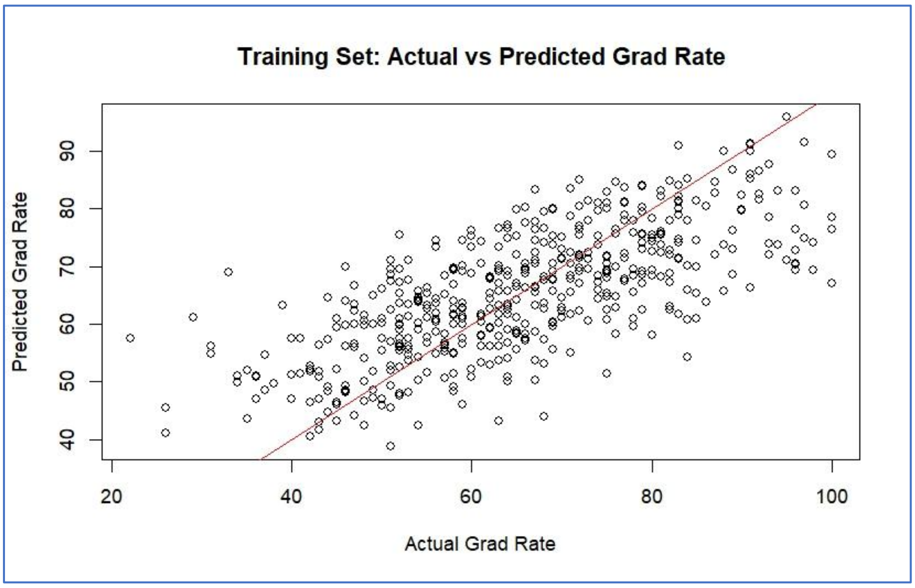
```
> step_summary$adj.r.squared; cleaned_summary$adj.r.squared
[1] 0.4190715
[1] 0.4942892
> AIC(model_step); AIC(cleaned_model_step)
[1] 4369.056
[1] 3744.67
> BIC(model_step); BIC(cleaned_model_step)
[1] 4416.385
[1] 3790.809
```

As we can see from the above the cleaned model has the best adjusted R^2 value and lowest **AIC** (3744.67) and **BIC** (3790.81) values. Hence, we proceed with the cleaned model.

3. **Cleaned Step-wise Regression Model VS Ridge Regression Model VS LASSO Regression Model**

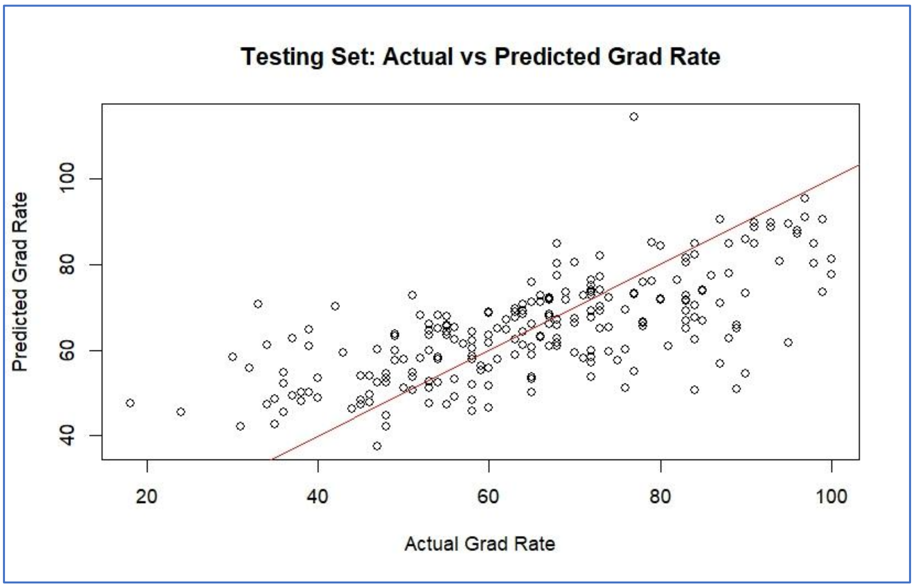
Cleaned Step-Wise Regression Model

The scatter plot below shows a positive correlation between the actual and the predicted graduation rates of the training set.



This model predicts the training set well.

The scatter plot below shows a positive correlation between the actual and the predicted graduation rates of the testing set.



This model does not predict the test data so well.

```
> # Training Set & Testing Set performance metrics
> train_mse; test_mse
[1] 116.677
[1] 154.9391
> train_rmse; test_rmse
[1] 10.80171
[1] 12.44745
> train_r2; test_r2
[1] 0.5035967
[1] 0.4771922
```

i. Mean Squared Error (RMSE):

- Training: 116.677
- Testing: 154.9391

ii. Root Mean Squared Error (RMSE):

- Training: 10.80171
- Testing: 12.44745

iii. Coefficient of Determination (R^2):

- Training: 0.5035967
- Testing: 0.477192

Interpretation of the above values:

As we can see MSE and RMSE is lesser for the testing set but the R^2 value is higher for the training set. In summary, the model performs better on training dataset suggesting overfitting, meaning the it does not generalize well with the test (new) data.

CONCLUSION

- LASSO Regression produces a simple model, this is done by eliminating the irrelevant predictors (Accept, Enroll, F.Undergrad, Terminal).
- Ridge Regression retained all features while effectively reducing overfitting.

- **Step-Wise Regression** yielded similar results to the LASSO regression, but is prone to multicollinearity. Removing the unusual observations improved the stability of the model.

Conclusion

Conclusion:

In this analysis, we successfully compared the LASSO regression, Ridge Regression, and Step-Wise Regression for predicting the Graduation Rate. We found that the LASSO Regression outperformed the Ridge Regression by reducing the model complexity all while maintaining the predictive power. The step-wise regression model provided with similar results but it lacked the robustness of the regularization techniques. This analysis highlights the importance of dealing with unusual observations and selecting the optimal predictors.

Works Cited

- Bluman, A. (2018). *Elementary statistics: A step-by-step approach* (10th ed.). McGraw Hill.
- Kabacoff, R. I. (2022). *R in action: Data analysis and graphics with R and tidyverse* (3rd ed.). Manning Publications.
- RDocumentation. (n.d.). ISLR::College dataset. Retrieved from *https://rdrr.io/cran/ISLR/man/College.html*
- *R-bloggers. (2021, October 6). Lambda.min, lambda.1se, and cross-validation in LASSO (binomial response). R-bloggers. https://www.r-bloggers.com/2021/10/lambda-min-lambda-1se-and-cross-validation-in-lasso-binomial-response/*

Appendix

R Code:

```
#Authors: Yash S
#Created: 2025-02-01
#Edited: 2025-02-09
#Course: ALY6015
#Assignment 4

cat("\014") # clears console
rm(list = ls()) # clears global environment
try(dev.off(dev.list()["RStudioGD"]), silent = TRUE) # clears plots
try(p_unload(p_loaded(), character.only = TRUE), silent = TRUE) # clears packages
options(scipen = 100) # disables scientific notion for entire R session

library(pacman)
p_load(tidyverse, caret, glmnet, ISLR, car, leaps)

# Loading the college dataset and saving it as dataframe
data("College")
college <- as.data.frame(College)
# To build regularization models by using Ridge and Lasso (least absolute shrinkage and
selection operator).
# Predict Grad.Rate for all models.

# EDA on the college dataset
summary(college) # 0 NA

# 1. Splitting the data into train and test set
# Maintaining a % of event rate 70/30 split
```

```
set.seed(123)

trainIndex <- createDataPartition(college$Grad.Rate, p = 0.7, list = FALSE, times = 1)

train_set <- college[trainIndex, ]

test_set <- college[-trainIndex, ]


x_train <- model.matrix(Grad.Rate ~ ., data = train_set)[,-1]

y_train <- train_set$Grad.Rate

x_test <- model.matrix(Grad.Rate ~ ., data = test_set)[,-1]

y_test <- test_set$Grad.Rate


##############################################################################
#####

# Ridge Regression

# 2. Finding best values for lambda

set.seed(123)

ridge_model <- cv.glmnet(x_train, y_train, alpha = 0) # Alpha = 0 for Ridge

# Comparing Lambda values

lambda_min_ridge <- ridge_model$lambda.min

lambda_1se_ridge <- ridge_model$lambda.1se

lambda_min_ridge; lambda_1se_ridge


# 3. Plotting the Ridge regression model

plot(ridge_model)


# 4. Fitting a Ridge regression model with minimum lambda value

ridge_fit <- glmnet(x_train, y_train,alpha = 0, lambda = lambda_min_ridge)

# Checking Coefficients

coef(ridge_fit)


# 5. Determining RMSE for training set
```

```
pred_train_ridge <- predict(ridge_fit, s = lambda_min_ridge, newx = x_train)
rmse_train_ridge <- sqrt(mean((y_train- pred_train_ridge)^2))

# 6. Determining RMSE for testing set
pred_test_ridge <- predict(ridge_fit, s = lambda_min_ridge, newx = x_test)
rmse_test_ridge <- sqrt(mean((y_test- pred_test_ridge)^2))

rmse_train_ridge; rmse_test_ridge

# Determining R-squared for training and testing data
# Computing R-squared for Ridge Regression
# Training Data
ss_res_ridge_train <- sum((y_train- pred_train_ridge)^2)
ss_tot_ridge_train <- sum((y_train- mean(y_train))^2)
r2_train_ridge <- 1- (ss_res_ridge_train / ss_tot_ridge_train)
# Testing Data
ss_res_ridge_test <- sum((y_test- pred_test_ridge)^2)
ss_tot_ridge_test <- sum((y_test- mean(y_test))^2)
r2_test_ridge <- 1- (ss_res_ridge_test / ss_tot_ridge_test)

r2_train_ridge; r2_test_ridge
###############################################################################
#####
# LASSO Regression
# 7. Finding best values for lambda
set.seed(123)
lasso_model <- cv.glmnet(x_train, y_train, alpha = 1)  # Alpha = 1 for LASSO
# Comparing Lambda values
lambda_min_lasso <- lasso_model$lambda.min
lambda_1se_lasso <- lasso_model$lambda.1se
```

```
lambda_min_lasso; lambda_1se_lasso

# 8. Plotting the LASSO regression model
plot(lasso_model)

# 9. Fitting a LASSO regression model with minimum lambda value
lasso_fit <- glmnet(x_train, y_train, alpha = 1, lambda = lambda_min_lasso)
# Checking Coefficients
coef(lasso_fit)

# 10. Determining RMSE for training set
pred_train_lasso <- predict(lasso_fit, s = lambda_min_lasso, newx = x_train)
rmse_train_lasso <- sqrt(mean((y_train- pred_train_lasso)^2))

# 11. Determining RMSE for testing set
pred_test_lasso <- predict(lasso_fit, s = lambda_min_lasso, newx = x_test)
rmse_test_lasso <- sqrt(mean((y_test- pred_test_lasso)^2))

rmse_train_lasso; rmse_test_lasso

# Determining R-squared for training and testing data
# Computing R-squared for LASSO Regression
# Compute R² for Lasso Regression
ss_res_lasso_train <- sum((y_train- pred_train_lasso)^2)
ss_tot_lasso_train <- sum((y_train- mean(y_train))^2)
r2_train_lasso <- 1- (ss_res_lasso_train / ss_tot_lasso_train)

ss_res_lasso_test <- sum((y_test- pred_test_lasso)^2)
ss_tot_lasso_test <- sum((y_test- mean(y_test))^2)
```

```
r2_test_lasso <- 1- (ss_res_lasso_test / ss_tot_lasso_test)


# Output R² values
r2_train_lasso; r2_test_lasso
##########################################################################################
# Comparison
# 13. Step-wise Feature Selection
model_step <- step(lm(Grad.Rate ~ ., data = train_set), direction = "both")
step_summary <- summary(model_step)
step_summary
step_summary$r.squared


# Plot diagnostic graphs for the regression model
par(mfrow = c(2, 2))
plot(model_step)
dev.off()


# Check for multicollinearity using VIF
# Now, running VIF function
vif(model_step)


# Handling unusual observations
# Identifying Outliers
standardized_residuals <- rstandard(model_step)
outlier_threshold <- 3
outliers <- which(abs(standardized_residuals) > outlier_threshold)
print(outliers)
# Visualizing outliers
```

```
plot(standardized_residuals, main = "Standardized Residuals", ylab = "Residuals", xlab =
"Index")

abline(h = c(-outlier_threshold, outlier_threshold), col = "red", lty = 2)

text(outliers, standardized_residuals[outliers], labels = outliers, col = "blue", pos = 4)


# Identifying high-Leverage points

leverage <- hatvalues(model_step)

leverage_threshold <- 2 * mean(leverage)

high_leverage <- which(leverage > leverage_threshold)

print(high_leverage)

# Visualizing high leverage points

plot(leverage, main = "Leverage Points", ylab = "Leverage", xlab = "Index")

abline(h = leverage_threshold, col = "red", lty = 2)

text(high_leverage, leverage[high_leverage], labels = high_leverage, col = "blue", pos = 4)


# Identifying influential observations

cooks <- cooks.distance(model_step)

influential_threshold <- 4 / nrow(train_set)

influential_points <- which(cooks > influential_threshold)

print(influential_points)

# Visualizing influential points

plot(cooks, main = "Cook's Distance", ylab = "Cook's Distance", xlab = "Index")

abline(h = influential_threshold, col = "red", lty = 2)

text(influential_points, cooks[influential_points], labels = influential_points, col = "blue", pos
= 4)


# Combining all unusual observations into a single vector

unusual_points <- sort(unique(c(high_leverage, outliers, influential_points)))

print(unusual_points)
```

```
# Removing the unusual observations
train_set_cleaned <- train_set[-unusual_points, ]

cleaned_model_step <- lm(Grad.Rate ~  Private + Apps + Top25perc + P.Undergrad +
               Outstate + Room.Board + Personal + perc.alumni + Expend,
               data = train_set_cleaned)
cleaned_summary <- summary(cleaned_model_step)
cleaned_summary

step_summary$adj.r.squared; cleaned_summary$adj.r.squared



# All subset regression
best_subset <- regsubsets(Grad.Rate ~., data = train_set_cleaned, nvmax = 9)
reg_summary <- summary(best_subset)
reg_summary
# Best model by Mallow's Cp and BIC
which.min(reg_summary$cp)
which.max(reg_summary$adjr2)

# Since all subset regression confirms that step-wise feature selected model is the best model
# We evaluate the cleaned best model
# Predictions on Training Set
train_pred <- predict(cleaned_model_step, newdata = train_set_cleaned)

# Computing performance metrics for Training Set
train_mse <- mean((train_set_cleaned$Grad.Rate- train_pred)^2)
train_rmse <- sqrt(train_mse)
train_r2 <- 1- (sum((train_set_cleaned$Grad.Rate- train_pred)^2) /
sum((train_set_cleaned$Grad.Rate- mean(train_set_cleaned$Grad.Rate))^2))
```

```
# Predictions on Testing Set
test_pred <- predict(cleaned_model_step, newdata = test_set)

# Computing performance metrics for Testing Set
test_mse <- mean((test_set$Grad.Rate- test_pred)^2)
test_rmse <- sqrt(test_mse)
test_r2 <- 1- (sum((test_set$Grad.Rate- test_pred)^2) / sum((test_set$Grad.Rate-
mean(test_set$Grad.Rate))^2))

# Training Set & Testing Set performance metrics
train_mse; test_mse
train_rmse; test_rmse
train_r2; test_r2

# Plotting Residuals for Training Set
plot(train_set_cleaned$Grad.Rate, train_pred, xlab = "Actual Grad Rate", ylab = "Predicted
Grad Rate", main = "Training Set: Actual vs Predicted Grad Rate")
abline(0, 1, col = "red")

# Plotting Residuals for Testing Set
plot(test_set$Grad.Rate, test_pred, xlab = "Actual Grad Rate", ylab = "Predicted Grad Rate",
main = "Testing Set: Actual vs Predicted Grad Rate")
abline(0, 1, col = "red")

###############################################################################
#####
AIC(model_step); AIC(cleaned_model_step)
BIC(model_step); BIC(cleaned_model_step)
```