

REGULARIZATION



SHAWN NJOROGE

# **INTRODUCTION**

The goal of this report is to compare and evaluate various regression methods—specifically Ridge regression, LASSO (Least Absolute Shrinkage and Selection Operator), and stepwise selection—for predicting graduation rates using the College dataset. The College dataset contains various predictors related to colleges in the United States, and our goal is to develop predictive models that can accurately estimate graduation rates using these predictors.   
Ridge regression and LASSO are regularisation techniques for predictive modelling that reduce overfitting and deal with multicollinearity. Both methods modify the traditional least squares objective function by adding a penalty term that controls the size of predictor coefficients. Ridge regression uses an L2 penalty, whereas LASSO employs an L1 penalty, which has the added advantage of performing feature selection.

Stepwise selection, on the other hand, is a traditional statistical method that systematically selects the most significant predictors using statistical criteria such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). It iteratively adds and removes predictors from the model to improve model fit.   
Throughout this report, we will:

1. Ridge regression and LASSO are implemented using the glmnet package in R, with cross-validation used to select optimal regularisation parameters.
2. To select features and build a regression model, use stepwise selection with R's MASS package.
3. Evaluate and compare the performance of these models on both training and testing datasets, with root mean square error (RMSE) serving as the primary evaluation metric.

# **ANALYSIS**

**Ridge Regression**

The Ridge regression model aims to predict college graduation rates (Grad.Rate) using various predictors from the College dataset. Ridge regression uses a regularisation parameter (lambda) to penalize large coefficients, which reduces overfitting and improves model robustness. The regularisation technique reduces model complexity by shrinking less important predictors to zero while retaining the influence of significant predictors.   
  
The summary of the Ridge regression model (ridge\_model) sheds light on its key components. The model's intercept (a0) is approximately 35.34, indicating the baseline graduation rate when all predictors are zero. The model coefficients (beta) are represented by a sparse matrix (dgCMatrix), which shows the impact of each predictor on Grad.Rate. Notably, the Private predictor has been effectively penalized to zero (.), indicating that it has no significant influence on graduation rates in this model. The model's degrees of freedom (df), lambda value (lambda) for regularisation, deviance ratio (dev.ratio), null deviance (nulldev), and number of passes (npasses) over the data during optimization all contribute to a better understanding of the model's performance and complexity.

In interpreting the coefficients (beta), we look at how predictors affect graduation rate. Positive coefficients (Top10percent, Top25percent, PhD, S.F. Ratio, perc.alumni) indicate a positive relationship with graduation rate, while negative coefficients (F.Undergrad, P.Undergrad, Room.Board, Books, Personal, Terminal, Expend) indicate a negative relationship. The effective elimination of the Private predictor (.) demonstrates the regularisation effect, which prioritises relevant predictors while reducing noise.

The Ridge regression model has a training root mean square error (RMSE) of around 12.38, which represents the average prediction error on the training dataset. The slightly higher test RMSE of approximately 14.17 indicates some degree of model overfitting, with the model performing slightly worse on unseen data than on training data. Despite this, the Ridge regression model demonstrates how regularisation can help manage model complexity and improve generalisation.   
  
Further evaluation and comparison with LASSO and stepwise selection will provide a comprehensive understanding of the performance and suitability of various regression methods for predicting graduation rates using the College dataset.

**LASSO**

The LASSO (Least Absolute Shrinkage and Selection Operator) regression model is used to forecast college graduation rates (Grad.Rate) using predictors from the College dataset. L1 regularisation is used to induce sparsity and perform feature selection. This technique penalises the absolute values of coefficients, effectively reducing less important predictors to zero while retaining influential ones.   
  
The LASSO model (lasso\_model) produces a sparse matrix (dgCMatrix) of coefficients (beta), which depicts the impact of each predictor on Grad.Rate. Notably, the Private predictor is effectively penalised (.), indicating its insignificance in predicting graduation rates. Positive coefficients (Top10perc, Top25perc, PhD, perc.alumni) indicate a positive relationship with graduation rate, whereas negative coefficients (P.Undergrad, Personal, Expend) indicate a negative relationship. The selective inclusion of predictors demonstrates LASSO's feature selection capability.

The LASSO model's coefficients show how predictors affect graduation rate (Grad.Rate). Key predictors (Top10percent, Top25percent, PhD, perc.alumni) have a positive impact on graduation rates, whereas others (P.Undergrad, Personal, Expend) have a negative impact. The regularisation effect effectively eliminates the Private predictor (.), highlighting the significance of relevant predictors.

The LASSO model has a training root mean square error (RMSE) of around 12.38, which represents the average prediction error on the training dataset. However, the test RMSE of approximately 14.22 indicates some degree of model overfitting, as the model performs slightly worse on unseen data than on training data. Despite this, the LASSO regression model exhibits effective feature selection and predictive performance, emphasizing its usefulness in managing model complexity and improving generalization.

**Stepwise Selection**

The stepwise regression procedure seeks to identify a subset of predictors that best predict college graduation rates (Grad.Rate) using the College dataset, with an iterative selection process guided by the Akaike Information Criterion (AIC) to balance model complexity and fit.   
  
Starting with a full model that includes all predictors (Private, Apps, Accept, Enrol, Top10percent, Top25percent, F.Undergrad, P.Undergrad, Outstate, Room.Board, Books, Personal, PhD, Terminal, S.F. Ratio, perc.alumni, Expend), the stepwise regression procedure iteratively adds or removes predictors to improve model fit while minimizing AIC.   
  
The stepwise regression output shows how the model evolved, with predictors being added or removed in order of their contribution to model fit. Notably, several predictors (Books, Top10perc, S.F.Ratio, Terminal, F.Undergrad) are gradually removed from the model, indicating their limited impact on predicting.

The final stepwise regression model includes a subset of predictors (Private, Apps, Accept, Enrol, Top25perc, P.Undergrad, Outstate, Room.Board, Personal, PhD, perc.alumni, Expend) that collectively achieve a training root mean square error (RMSE) of around 12.24, which represents the average prediction error on the training dataset. However, the test RMSE of approximately 14.47 indicates some degree of model overfitting, as the model performs slightly worse on unseen data than on training data.

When comparing stepwise regression to Ridge and LASSO regression models for predicting college graduation rates (Grad.Rate), all three methods perform similarly. The stepwise regression model has a training RMSE of about 12.24 and a test RMSE of about 14.47, which is similar to the Ridge (training RMSE ≈ 12.38, test RMSE ≈ 14.17) and LASSO (training RMSE ≈ 12.38, test RMSE ≈ 14.22) regression models. The regression method used is determined by the objectives: Ridge regression is useful for dealing with multicollinearity and retaining all predictors, whereas LASSO regression excels at feature selection and model interpretability by reducing less important predictors to zero.

Stepwise regression strikes a balance between model complexity and performance by iteratively selecting predictors using model fit criteria. Finally, the preferred method is determined by the analysis goals, dataset characteristics, and desired balance of model interpretability and predictive accuracy. Further exploration and evaluation of various regression techniques can help guide the choice of the best approach for predicting graduation rates in the College dataset.

# **CONCLUSION/INTERPRETATIONS**

In conclusion, our analysis of Ridge, LASSO, and stepwise regression models for predicting college graduation rates (Grad.Rate) shows that all methods perform similarly in terms of predictive accuracy. The stepwise regression model performs competitively, with a training RMSE of about 12.24 and a test RMSE of about 14.47, comparable to the Ridge and LASSO regression models. Each regression technique provides distinct advantages: Ridge regression is effective for managing multicollinearity and preserving all predictors, while LASSO regression excels at feature selection and producing sparse, interpretable models. Stepwise regression strikes a pragmatic balance between complexity and performance by iteratively selecting predictors based on model fit criteria. Overall, the optimal regression method is determined by the desired balance of model complexity, interpretability, and predictive accuracy.

# **REFERENCES**

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1), 1-22.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301-320.

# **APPENDIX**

library(glmnet)

library(caret)

library(ISLR)

library(MASS)

data("College")

# Check for missing values in the input matrix

sum(is.na(x))

#RIDGE REGRESSION

#Split the data

set.seed(123)

trainIndex <- createDataPartition(College$Grad.Rate, p = .8, list = FALSE)

trainData <- College[trainIndex, ]

testData <- College[-trainIndex, ]

#Cross-validation to Select Lambda

x <- as.matrix(trainData[, -ncol(trainData)])

y <- trainData$Grad.Rate

cv\_model <- cv.glmnet(x, y, alpha = 0, nfolds = 10)

lambda\_min <- cv\_model$lambda.min

lambda\_1se <- cv\_model$lambda.1se

# Plot Cross-Validation Results

plot(cv\_model)

#Fit Ridge Regression Model

ridge\_model <- glmnet(x, y, alpha = 0, lambda = lambda\_min)

summary(ridge\_model)

#Examine Coefficients

coef(ridge\_model)

#Evaluate Model Performance

# Predict on training set

train\_pred <- predict(ridge\_model, newx = x)

train\_rmse <- sqrt(mean((y - train\_pred)^2))

# Predict on test set

test\_x <- as.matrix(testData[, -ncol(testData)])

test\_y <- testData$Grad.Rate

test\_pred <- predict(ridge\_model, newx = test\_x)

test\_rmse <- sqrt(mean((test\_y - test\_pred)^2))

train\_rmse

test\_rmse

#LASSO

# Split data into training and testing sets

set.seed(123)

trainIndex <- createDataPartition(College$Grad.Rate, p = .8, list = FALSE)

trainData <- College[trainIndex, ]

testData <- College[-trainIndex, ]

# Fit LASSO regression model and evaluate

# Prepare training data

x\_train <- as.matrix(trainData[, -ncol(trainData)])

y\_train <- trainData$Grad.Rate

# Estimate lambda values using cross-validation

lasso\_cv <- cv.glmnet(x\_train, y\_train, alpha = 1, nfolds = 10)

lambda\_min <- lasso\_cv$lambda.min

# Plot cross-validation results

plot(lasso\_cv)

# Fit LASSO regression model with optimal lambda

lasso\_model <- glmnet(x\_train, y\_train, alpha = 1, lambda = lambda\_min)

# View coefficients

coefficients <- coef(lasso\_model)

print(coefficients)

# Evaluate model performance on training set

train\_pred <- predict(lasso\_model, newx = x\_train)

train\_rmse <- sqrt(mean((y\_train - train\_pred)^2))

print(paste("Training RMSE:", train\_rmse))

# Prepare test data

x\_test <- as.matrix(testData[, -ncol(testData)])

y\_test <- testData$Grad.Rate

# Evaluate model performance on test set

test\_pred <- predict(lasso\_model, newx = x\_test)

test\_rmse <- sqrt(mean((y\_test - test\_pred)^2))

print(paste("Test RMSE:", test\_rmse))

# Fit stepwise selection model using MASS package

stepwise\_model <- stepAIC(lm(Grad.Rate ~ ., data = trainData))

# Predict on training and testing sets

stepwise\_train\_pred <- predict(stepwise\_model, newdata = trainData)

stepwise\_test\_pred <- predict(stepwise\_model, newdata = testData)

# Calculate RMSE

stepwise\_train\_rmse <- sqrt(mean((trainData$Grad.Rate - stepwise\_train\_pred)^2))

stepwise\_test\_rmse <- sqrt(mean((testData$Grad.Rate - stepwise\_test\_pred)^2))

print(paste("Stepwise Selection Training RMSE:", stepwise\_train\_rmse))

print(paste("Stepwise Selection Test RMSE:", stepwise\_test\_rmse))