Regularized Regressions: LASSO, Ridge, and Elastic Net

Ryan Cain, Amber Laster, and Kieran McCarthy

April 16, 2021

1 Introduction

Inevitably, as an introductory statistics student takes their first steps into the world of multiple linear regression they are often confronted with many decisions. Some of which are, should the model include all the variables recorded in the study? Is the model too complex; can it be made simpler without sacrificing accuracy? Is there a way to get better predictions? By its very nature, ordinary least squares regression, in its attempt to minimize the residual sum of squares, has a low amount of bias, or minimizes the bias to the best of the algorithm's ability. However, this low bias comes with the price of a possible increase in variance. This could lead to models that are overfit to the given data.

One common way to deal with overfitting the model is to use a subset of the model. The two main reasons one would be unhappy with their original model is the model's lack of prediction accuracy and the model's complexity in interpretation.[7, p. 57] The idea of trading in some bias for a reduction of overall variance when it comes to prediction is desirable. Not to mention that fewer variables would reduce the complexity of the model. Common algorithms such as forward stepwise and backward stepwise exist to address this concern and are often successful in identifying a better model with fewer predictors and possibly lower prediction error. However, because subset selection is a discrete process, it can still result in high variance, and does nothing to shrink the full model's prediction error.[7, p. 61]

An alternative to selecting a subset of the full model is to use Ridge regression. Ridge regression's objective is to reduce a predictor's coefficient to lessen the predictor's overall influence on the model, i.e. reduce the chance of overfitting the model due to predictors that are correlated. The term multicollinearity is often used to describe a model whose parameters are correlated. Ridge regression is one of the shrinkage or regularization methods to address this issue. Ridge regression was invented by Hoerl and Kennard in 1970.[4, p. 4] Ridge regression is a process that addresses the issues that arise from a model that suffers from multicollinearity. Multicollinearity can produce least squares estimates that are unbiased but can have variances that are quite large. The Ridge regression method imposes a penalty to each predictor's coefficient, introducing a small amount of bias with the hope of a large decrease in variance.[7, p. 63] What is meant by introducing bias, or applying a penalty, is the process of altering the predictors coefficients such that they typically move closer to zero. Therefore, these techniques are sometimes referred to as shrinkage or regularization methods. The result of introducing bias is that our generated model doesn't fit the training data as well. While this might initially seem to be undesirable, doing so could lead to better predictions with future testing data, and therefore these shrinkage methods are often used. One aspect of Ridge regression that is important to note is that the coefficients are shrunk towards zero but do not reach zero.[7, p. 63] This means that all predictors used will remain in the model when using Ridge regression.

Another common shrinkage method is LASSO regression. LASSO stands for 'least absolute shrinkage and selection operator'.[6, p .267] LASSO regression was first independently developed in 1986 then popularized in 1996 by Robert Tibshirani.[3] LASSO regression also shrinks coefficients but as opposed to Ridge regression, LASSO can shrink the unnecessary coefficients to zero.[7, p. 69] Like Ridge regression, LASSO applies a penalty to the coefficients. With the coefficients being shrunk to zero LASSO regression

also works as a variable selection algorithm as well as a regularization or shrinkage technique. LASSO essentially addresses the main issues of accuracy and interpret-ability in a single process. Some coefficients will be shrunk to zero and hence retains both aspects of subset selection and Ridge regression.

While LASSO regression appears to be superior to Ridge regression, LASSO regression does have some shortcomings. LASSO regression does not perform well when the number of parameters is larger than the sample size. [8, p. 301] Additionally, LASSO regression tends to only select one predictor from a group of predictors that are highly correlated.[8, p. 301] To address this issue for LASSO regression, a new regularization technique was proposed called Elastic Net by Hui Zou and Trevor Hastie. "[Elastic net] is like a stretchable fishing net that retains all the big fish".[8, p. 302] What is meant by this statement is that Elastic Net will select the significant predictor variables, even if they are correlated. The Elastic Net regularization process combines both penalties from Ridge regression and LASSO regression. The advantage of using both is that a subset of the model's parameters are selected making the model easier to understand, but if some of the remaining variables selected are correlated, the multicollinearity present could be taken care of by the Ridge regression penalty.

Ordinary least squares (OLS) regression generally does not always perform very well regarding parsimony and prediction when all variables are used. As such, the concept of introducing small amounts of bias into the parameters of the model with the hope of finding a decrease in variance is used and is often referred to as shrinkage methods. One shrinkage method, Ridge regression, helps deal with OLS models that suffer from multicollinearity with the intention of improving prediction but does not help with parsimony. This is because all variables will remain in the model. LASSO regression, on the other hand, will penalize variables such that their coefficients will drop to zero. This allows for a possible increase in prediction performance, as well as making the model simpler as it removes variables from the model. However, LASSO regression can struggle when the number of predictor variables is close to or exceeds the sample size. To deal with this issue another regularization technique was introduced called Elastic Net. Elastic Net essentially combines both types of penalties from Ridge and LASSO regression to come up with a regularization technique that will address prediction and parsimony without the shortcomings of LASSO regression, and generally outperforms both LASSO and Ridge regression when used appropriately.

2 Methods

As mentioned in the introduction, LASSO and Ridge regression introduce some bias with the hope of reducing variance. The degree of penalty that is applied is represented by λ which is estimated using cross-validation to try and identify the ideal balance between bias and variance to minimize the *MSE*. λ must be a value equal to zero or larger, up to positive infinity. All three methods use the Least Squares Estimator formula subject to some "shrinkage" penalty to either shrink or remove unnecessary estimators. The first part of each equation is the traditional residuals sum of squares (RSS) from the OLS method and the second part is the ℓ_1 or ℓ_2 penalization scaled by λ. It is helpful to note that if $\lambda = 0$ the OLS model is used. The penalty for each method is different, using the ℓ_1 norm for LASSO, ℓ_2 norm for Ridge regression, and a combination of both for the Elastic Net. The ℓ_1 norm uses the absolute value of the magnitude of the β_j's and creates a diamond shape for the constraint region. The ℓ_2 norm uses the square of the magnitude of the β_j's and creates a circle shape for the constraint region. The two different shapes that are created by the difference in the ℓ_1 and ℓ_2 constraint regions is displayed in Figure 1 below from [3] created by [1].

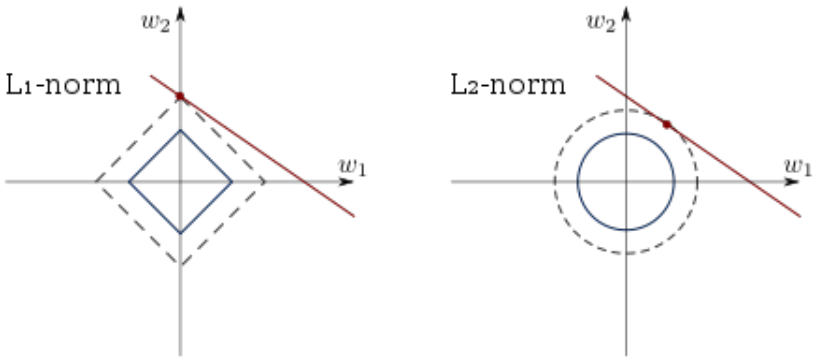


Figure 1: Forms of the constraint regions for LASSO and Ridge regression.

2.1 Method 1: Ridge

Ridge regression uses the ℓ_2 norm which squares the value of the parameter is it applied to. It is helpful to note that since these methods are applications to the OLS all assumptions to the OLS should be considered for the Regularization methods. Ridge regression is minimizing is the following equation:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=i}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

With the Lagrangian form that minimizes the β_j's as follows.[3]

$$\frac{1}{N}||y - X\beta||^2 + \lambda||\beta||^2$$

2.2 Method 2: LASSO

The LASSO method uses the ℓ_1 norm which applies the absolute value to the β_j along with a shrinkage parameter λ. This is also what allows LASSO to set some β_j's to 0 as it is possible for the regression line to fall directly on a corner, or 0, of the constraint region. Below is the equation that LASSO regression is minimizing:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=i}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

With the Lagrangian form that minimizes the β_j's as follows.[3]

$$\frac{1}{N}||y - X\beta||_2^2 + \lambda|\beta|$$

2.3 Method 3: Elastic Net

Elastic Net regression combines both penalties from LASSO and Ridge regression. Ridge and LASSO regression each have some short comings on their own combining them together for the Elastic Net method allows some of these issues to be overcome. Ridge regression will not remove any variables from the equation leaving an equation that might still be overly complicated or difficult to interpret. LASSO tends to select one out of a group of correlated variables eliminating the others and therefore possibly eliminating other useful variables. LASSO can also struggle when there are more predictors than data. The formula Elastic Net minimizes is as follows:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=i}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| + \lambda\sum_{j=1}^{p}\beta_j^2$$

With the Lagrangian form of the Elastic Net that minimizes the β_j's as follows.[2]

$$\frac{1}{N}||y - X\beta||^2 + \lambda_2||\beta||^2 + \lambda_1 * |\beta|$$

As before there is the *RSS* portion from OLS method. However, this time we have both the ℓ_1 and ℓ_2 penalties applied. The ℓ_1 and ℓ_2 are scaled by using two different λ's, $_1$ and λ_2 respectively. They are both found individually and are both estimated by using cross-validation. While all three methods theoretically have strengths and weaknesses some examples are provided below to illustrate each method.

3 Illustrative examples with R

The data used for the examples was based on the measurements of Haliotis abalones from Tasmania. The data was obtained from the UCI Machine Learning Repository.[5] We are interested in predicting the age of an abalone based on the number of shell layers the abalone has grown with the age of the abalone being 1.5 years more than the number of layers. This data set contains 4,177 observations of abalone with each observation containing eight measurements: the abalone's length (mm), diameter (mm), height (mm), whole weight (g), shucked weight (g), viscera weight (g), shell weight (g), and number of rings observed after cutting through the cone of the abalone.

The OLS model was created as a baseline for the LASSO, Ridge, and Elastic Net models. The residual plots were used to analyze the normality and homoskedastic assumptions and presence of outliers. Two outliers were removed from the data set as it appears they were data entry errors. Analysis of the diagnostic plots of the model predicting ring number from all variables did not satisfy the normality and homoskedastic assumptions of regression, so a transformation was attempted. The natural log of the response variable was used as a transformation to satisfy the assumptions.

The transformed OLS model showed all regressors were significant and the model was significant. The OLS model had an R^2 of 0.5934 or 59.34% of the total variability is explained by the OLS model. The Mean Square Error (MSE) for the OLS model is 0.155. Next, the Variance Inflation Factor (VIF) was found to determine if multicollinearity was present in the data set. We can see in figure 3 that based on the VIF there is multicollinearity in 5 out of the 7 predictors. Applying a max value of 20 for the VIF the Length, Diameter, Whole weight, Shucked weight, and Shell weight all have multicollinearity present.

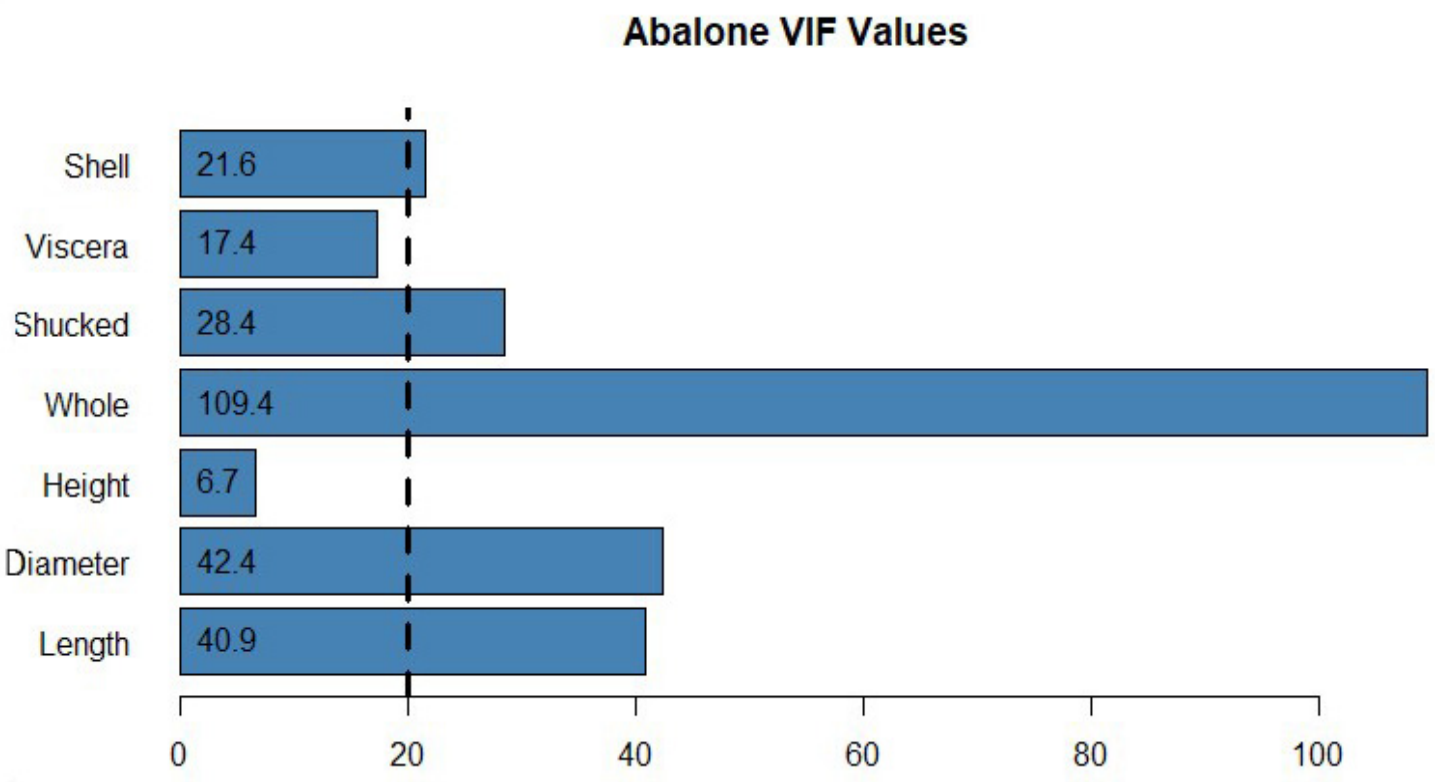
```
## Multicollinearity

#Find the VIF and create vector of the values
vif(ringmod2)
vif_values <- vif(ringmod2)

#Create horizontal bar chart to display each VIF value
bp <- barplot(vif_values, main = "Abalone VIF Values", horiz = TRUE, col = "steelblue")


#Add values
text( 0,bp,round(vif_values,1), cex=1,pos=4)

#Add a vertical line at 20 to illustrated the cutoff value
abline(v = 20, lwd = 3, lty = 2)
```



The multicollinearity present can cause issues with the estimators being too large or possibly having the wrong signs. The application of the LASSO, Ridge, and Elastic Net methods may help to reduce or remove the multicollinearity and produce a model better suited for interpretation or prediction. In this section we will compare the coefficients of the different methods, the R^2, the MSE, and the Mean Absolute Percentage Error (MAPE) to determine the effectiveness of the regularized regression methods.

3.1 Example 1: Ridge and LASSO Regression

3.1.1 Ridge Regression

For the glmnet function the data first needs to be split into X and Y variables. Next the data was split into training and testing data to first be used for all 3 methods.

```
#Setting up Model variables for glmnet, Training and testing data
x_vars <- model.matrix(log(Rings)~Length + Diameter + Height + Whole + Shucked + Viscera
    + Shell , abalone2)[,-1]
y_var <- log(abalone2$Rings)

set.seed(86)
train = sample(1:nrow(x_vars), nrow(x_vars)/2)
x_test = (-train)
y_test = y_var[x_test]
```

The function cv.glmnet is used with the lambda_seq to apply 10 fold cross validation and find the optimal λ. Then the function glm.net with alpha=0 is used to find the Ridge regression model. Note: The way glmnet works is by setting an alpha value, when alpha = 0, Ridge regression is done, when alpha = 1, LASSO regression is done. For alpha values between 0 and 1 elastic net is run where smaller values are more weighted towards ridge regression and larger values are more weighted towards lasso regression. An alpha of 0.5 would mean an equal weight of penalty for both ridge and lasso regression penalties that make up the elastic net overall penalty.

```
############## Ridge
library(glmnet)
lambda_seq <- 10^seq(2, -2, by = -.1)

#Finding Best Lambda
R_cv_output <- cv.glmnet(x_vars[train,], y_var[train], alpha = 0, lambda = lambda_seq)
R_best_lam <- R_cv_output$lambda.min

#Plots
plot(R_cv_output)
plot(R_cv_output$glmnet.fit, "lambda", label = TRUE)

#Ridge Model
R_best <- glmnet(x_vars[train,], y_var[train], alpha = 0, lambda = R_best_lam)
coef(R_best)

#Ridge Predictions
R_pred <- predict(R_best, s = R_best_lam, newx = x_vars[x_test,])

#Calculating R Squared
R_actual <- y_test
R_preds <- R_pred
R_rss <- sum((R_preds - R_actual) ^ 2)
R_tss <- sum((R_actual - mean(R_actual)) ^ 2)
R_rsq <- 1 - R_rss/R_tss
R_rsq
[1] 0.583282

#MSE Ridge
R_mse <- mean((R_actual - R_preds)^2)
R_mse
[1] 0.04352106
```

The Ridge regression reduced the MSE to 0.0435 and the R^2 slightly reduced to 0.583. It severely penalized many of the coefficients. It reduced all the coefficients other than the Height and Diameter to less than 1. The plot 2 below shows the coefficients as the lambda moves from -4 to 4. As you can see they move towards 0 the larger λ becomes.

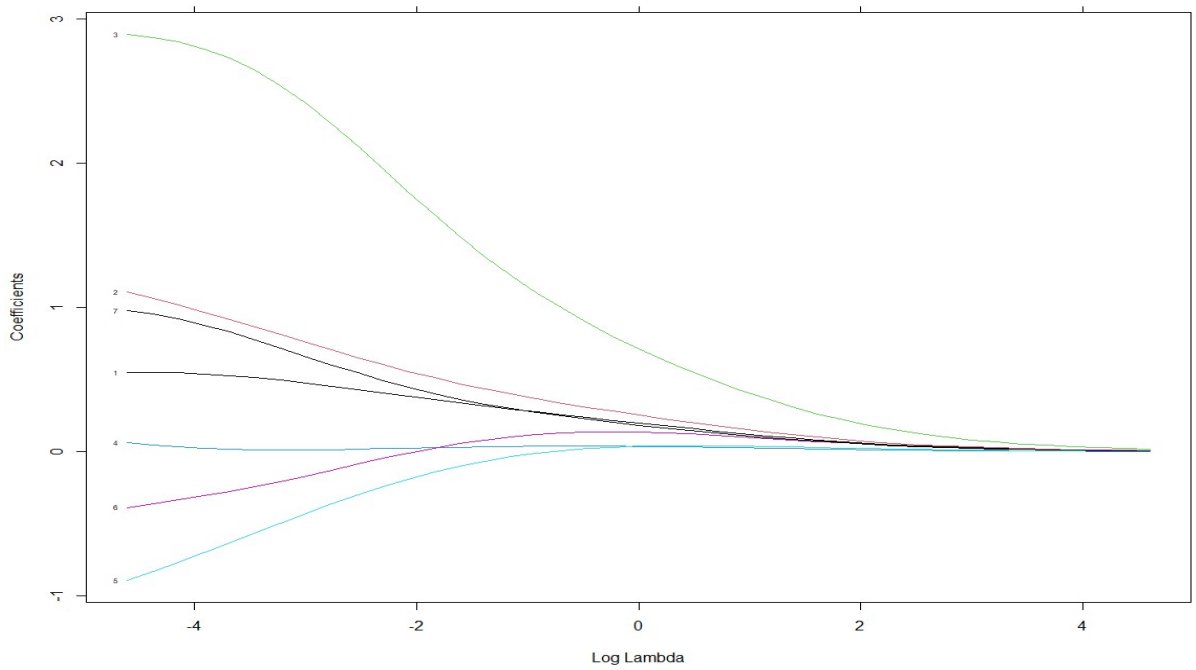


Figure 2: Abalone Coefficients for Ridge as Lambda Increases

Figure 3 shows the movement $\log(\lambda)$ as λ increases from -4 to 4 for selection purposes. It is ideal to chose a λ that will minimize the MSE for the model. For the Ridge model as the λ increases the MSE increases quickly, therefore the lower values for λ appear to be the best choice.

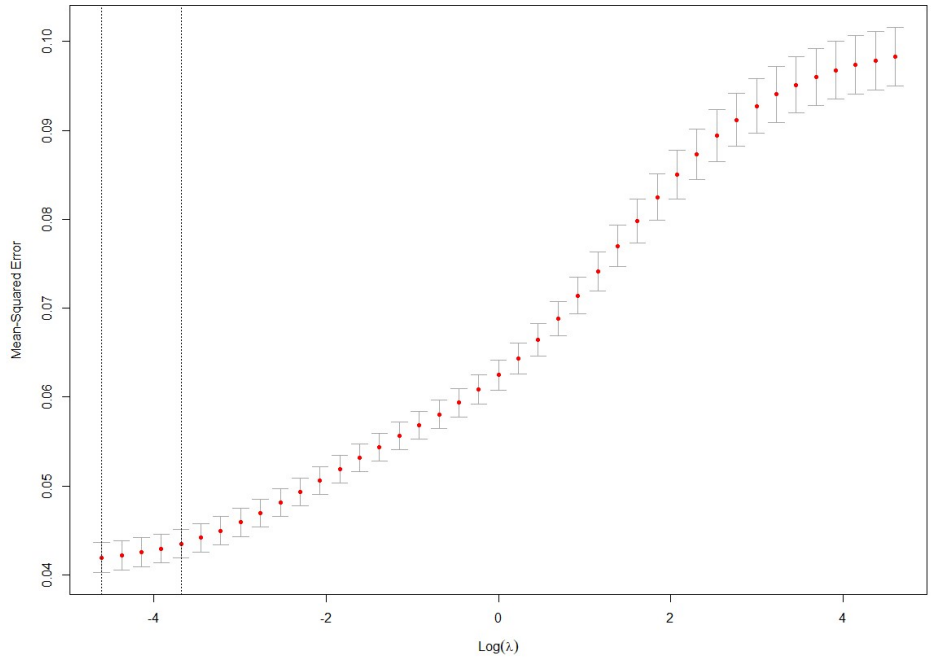


Figure 3: Ridge Lambda Selection

3.1.2 LASSO

The same steps are used in R for the LASSO method with the exception that the alpha in glmnet function must be changed to "1" to perform LASSO instead of Ridge done previously.

```
############### LASSO
library(glmnet)
lambda_seq <- 10^seq(2, -2, by = -.1)

#Finding Best Lambda
L_cv_output <- cv.glmnet(x_vars[train,], y_var[train], alpha = 1, lambda=lambda_seq)
L_best_lam <- L_cv_output$lambda.min

#Plots
plot(L_cv_output)
plot(L_cv_output$glmnet.fit, "lambda", label = TRUE)

#LASSO Model
L_best <- glmnet(x_vars[train,], y_var[train], alpha = 1, lambda = L_best_lam)
coef(L_best)

#LASSO Predictions
L_pred <- predict(L_best, s = L_best_lam, newx = x_vars[x_test,])

#Calculating R Squared
L_actual <- y_test
L_preds <- L_pred
L_rss <- sum((L_preds - L_actual) ^ 2)
L_tss <- sum((L_actual - mean(L_actual)) ^ 2)
L_rsq <- 1 - L_rss/L_tss
L_rsq
[1] 0.5707695
```

```
#MSE
L_mse <- mean((L_actual - L_preds)^2)
L_mse
[1] 0.04482784
```

The result of LASSO is that the final model will exclude the Length, Whole, and Viscera variables. It also reduces the MSE to 0.0448 and the R^2 is only reduced to 0.5707. However this model does add some benefit in that there are less predictors so the interpretation could be easier while still retaining most of the benefits of the OLS. Figure 4 below shows how the coefficients change as different λ's are chosen. Unlike Ridge regression, the coefficients here move much more erratically with some reducing to zero quickly, and others more slowly.

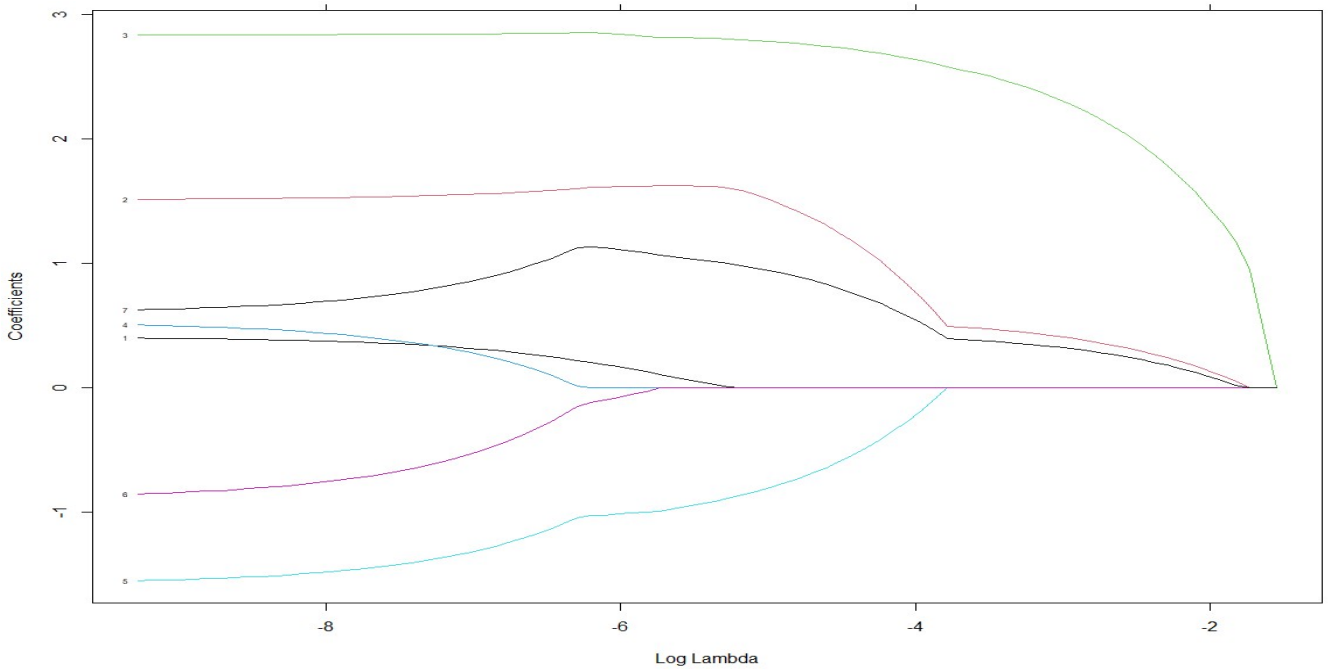


Figure 4: Abalone Coefficients for LASSO as Lambda Increases

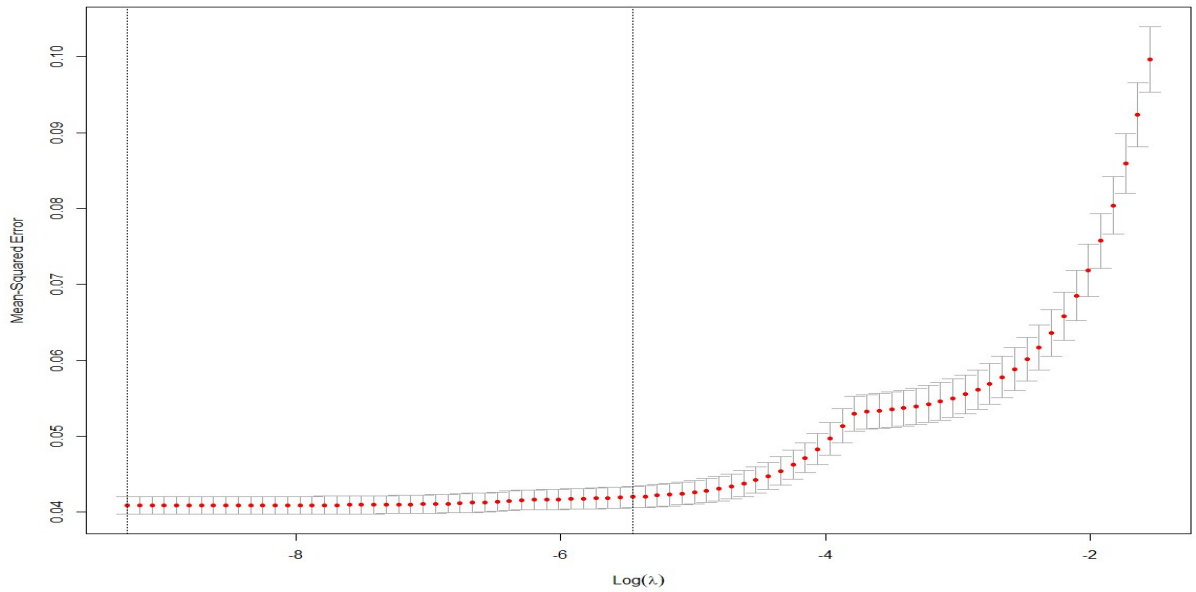


Figure 5: LASSO Lambda Selection

Figure 5 shows $\log(\lambda)$ as λ is increasing. As λ increases the MSE stays relatively stable for much longer in comparison to the Ridge regression λ. Therefore LASSO is able to use a higher λ value which may help in reducing some of the coefficients to 0.

Both the Ridge and LASSO methods appear to provide benefits to the model compared to the OLS model. The Elastic Net method will be applied to see if the combination of the LASSO and Ridge methods further improves the model.

3.2 Example 2: Elastic Net

The process in R for the Elastic Net is relatively the same the main difference is that the alpha now must be set to 0.5. The different levels of alpha in the glmnet function change which method will be used where 0 is Ridge and 1 is LASSO.

```
############### Elastic Net
library(glmnet)
lambda_seq <- 10^seq(2, -2, by = -.1)

#Finding Best Lambda
N_cv_output <- cv.glmnet(x_vars[train,], y_var[train], alpha = 0.5, lambda = lambda_seq)
N_best_lam <- N_cv_output$lambda.min

#Plots
plot(N_cv_output)
plot(N_cv_output$glmnet.fit, "lambda", label = TRUE)

#Elastic Net Model
N_best <- glmnet(x_vars[train,], y_var[train], alpha = 0.5, lambda = N_best_lam)
coef(N_best)

#Elastic Net Predictions
N_pred <- predict(N_best, s = N_best_lam, newx = x_vars[x_test,])

#Calculating R Squared
N_actual <- y_test
N_preds <- N_pred
N_rss <- sum((N_preds - N_actual) ^ 2)
N_tss <- sum((N_actual - mean(N_actual)) ^ 2)
N_rsq <- 1 - N_rss/N_tss
N_rsq
[1] 0.5790217

#MSE
N_mse <- mean((N_actual - N_preds)^2)
N_mse
[1] 0.043966
```

For Elastic Net the coefficients are again affected where some are removed and some are reduced. The R^2 is again slightly lower at 0.579, but with a simpler model. The MSE for the Elastic Net is also similar to the Ridge and LASSO models at 0.0439. The coefficient and λ plots are also provided below. The Elastic Net graphs look similar to the LASSO plots.

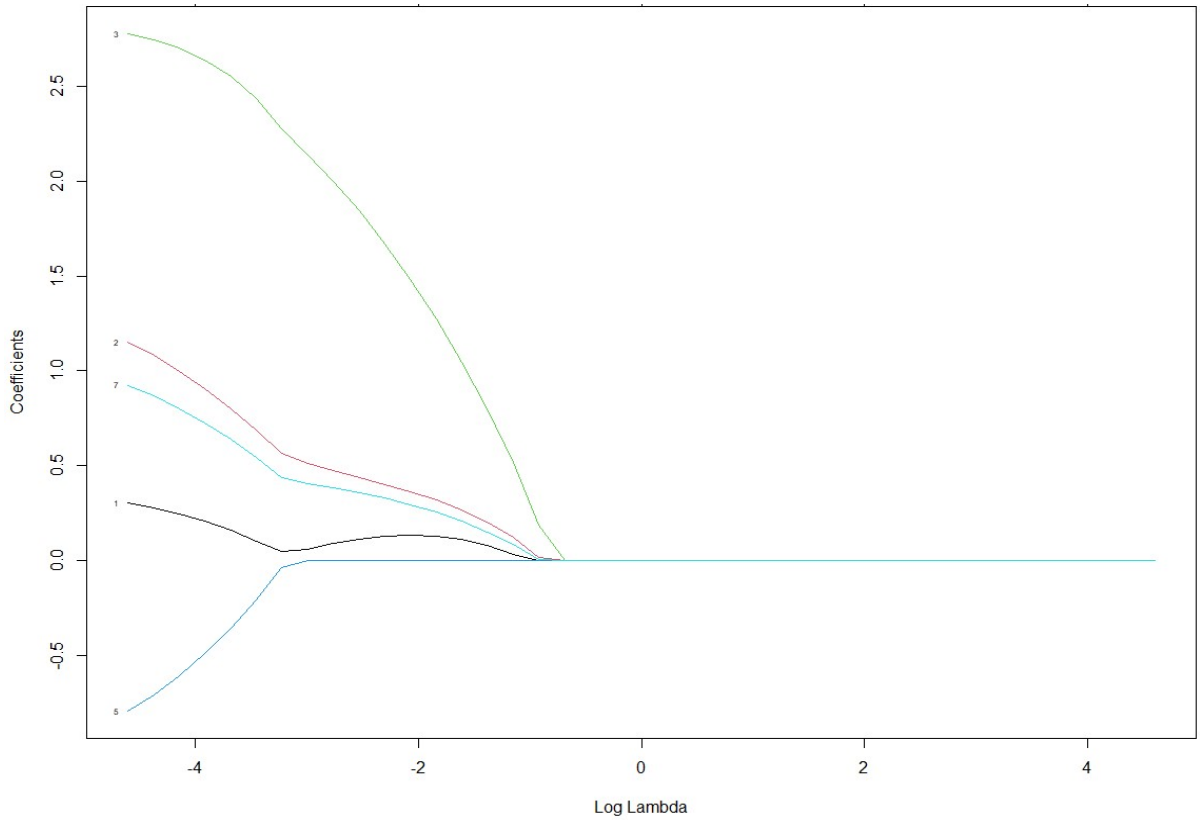


Figure 6: Abalone Coefficients for Elastic Net as Lambda Increases

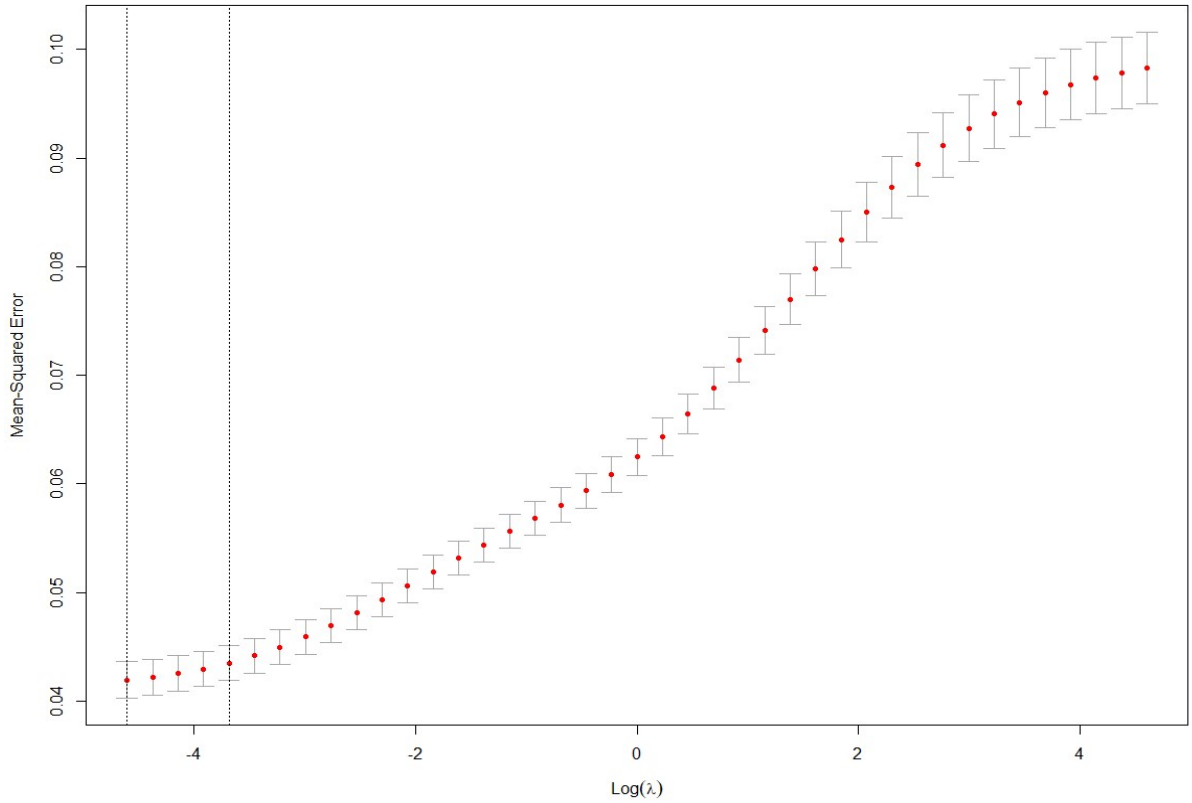


Figure 7: Elastic Net Lambda Selection

3.2.1 Comparison of Models

The table 1 gives us a side by side comparison of all of the coefficients for the OLS, Ridge, LASSO, and Elastic Net models.

Variable	OLS	Ridge	LASSO	Elastic Net
(Intercept)	1.20531	1.21651538	1.3636187	1.2954318
Length	0.32933	0.54644458	.	0.3055971
Diameter	1.45387	1.10359785	1.3103523	1.1522497
Height	2.80209	2.88625880	2.7427179	2.7756384
Whole	0.62326	0.05978089	.	.
Shucked	-1.65490	-0.89221363	-0.6413836	-0.7915657
Viscera	-0.81916	-0.38566870	.	.
Shell	0.46406	0.97889634	0.8318673	0.9235059

Table 1: Coefficients for all 4 Models

The coefficients fluctuate quite a bit as you move from model to model. First, Ridge reduces most coefficients with the exception of the Length, Height, and Shell weight. Those coefficients are increased relative to the OLS model. For LASSO it sets the Length, Whole, and Viscera to zero and reduces all others except Shell, which is increased. Lastly, the Elastic Net removes only the Whole and Viscera coefficients. The Elastic Net also decreased all coefficients except for Shell weight. The Ridge regression kept all the variables in the model compared to the the LASSO which removed 3 variables and the Elastic Net which only removed 2 variables. The Elastic Net kept the Length where the LASSO removed the variable. A comparison of this model that should be noted is that the Shell weight may be a significant predictor that is not fully captured by the OLS model as all three of the regularization models increase the coefficient. In addition the Whole weight in grams and the Viscera weight in grams may not be necessary. It may be that these two weights are in fact captured by the Shell weight which was increased in all the models.

A box-plot was created to compare the mean MAPE for each of the 4 models calculated 30 different times. With only 5% of the data used for the training data and the rest used for testing data.

```
################ MAPE Simulations
MAPE.simulation <- function(numloops = 50, train.p = 0.5){

  ols.mape = numeric()
  rid.mape = numeric()
  las.mape = numeric()
  ela5.mape = numeric()

  for( i in (1:numloops)) {
    index.train = sample((1: nrow(abalone2)),train.p*nrow(abalone2))
    train.data = abalone2[index.train,]
    test.data = abalone2[-index.train,]
    X <- model.matrix(log(Rings)~Length + Diameter + Height + Whole + Shucked + Viscera
        + Shell , train.data)[,-1]
    Y <- log(train.data$Rings)
    lambda_seq <- 10^seq(2, -2, by = -.1)

    ols.model = lm(log(Rings) ~ Length + Diameter + Height + Whole + Shucked + Viscera +
         Shell, data = train.data)

    rid.lam = cv.glmnet(x=X, y=Y, alpha = 0, lambda = lambda_seq)
    rlam = rid.lam$lambda.min
    rid.model = glmnet(x=X, y=Y, alpha = 0, lambda = rlam)

    las.lam = cv.glmnet(x=X, y=Y, alpha = 1,lambda = lambda_seq)
    llam = las.lam$lambda.min
    las.model = glmnet(x=X, y=Y, alpha = 1, lambda = llam)
```

```
26
27      ela.lam = cv.glmnet(x=X, y=Y, alpha = 0.5, lambda = lambda_seq)
28      elam = ela.lam$lambda.min
29      ela.model = glmnet(x=X, y=Y, alpha = 0.5, lambda = elam)
30
31      ols.pred = predict(ols.model, test.data)
32      rid.pred = predict(rid.model, s=rlam, newx=data.matrix(test.data[,-c(1,9)]))
33      las.pred = predict(las.model, s=llam, newx=data.matrix(test.data[,-c(1,9)]))
34      ela.pred = predict(ela.model, s=elam, newx=data.matrix(test.data[,-c(1,9)]))
35
36      ols.mape[i] = mape(actual = log(test.data$Rings), predicted = exp(ols.pred))*100
37      rid.mape[i] = mape(actual = log(test.data$Rings), predicted = exp(rid.pred))*100
38      las.mape[i] = mape(actual = log(test.data$Rings), predicted = exp(las.pred))*100
39      ela5.mape[i] = mape(actual = log(test.data$Rings), predicted = exp(ela.pred))*100
40      print(x=(i/numloops)*100, digits = 4)
41    }
42
43    boxplot(ols.mape, rid.mape, las.mape, ela5.mape,
44            names = c("OLS", "Ridge", "LASSO", "ElaNet"),
45            ylab = "MAPE")
46
47    summary(ols.mape)
48    print("COMPLETE!")
49  }
50
51  MAPE.simulation(250, 0.05)
```

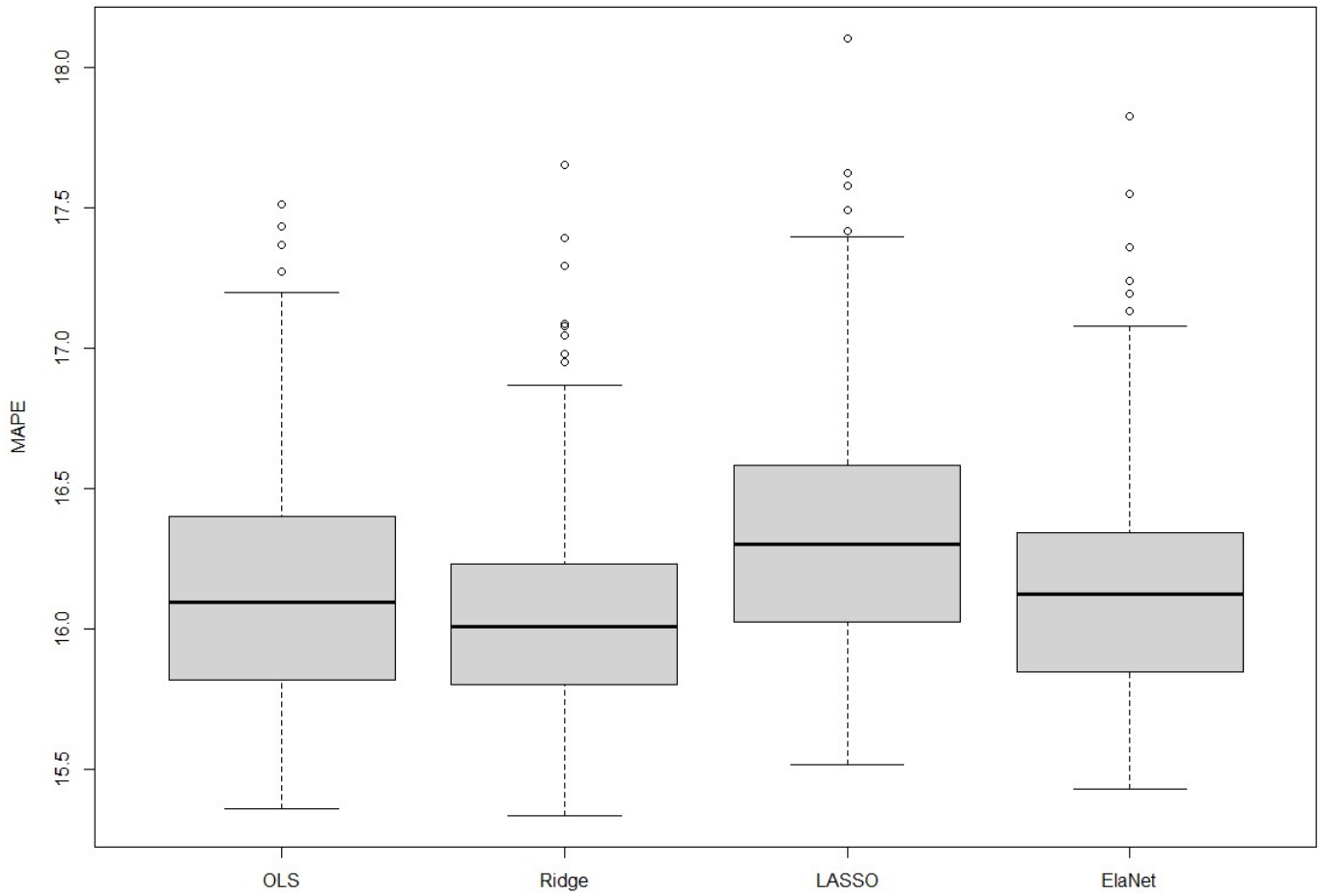


Figure 8: MAPE Boxplot for all 4 Models

Figure 8 is a box plot of the mean MAPE for each of the 4 models which shows us that again all of the Regularization models have a lower MAPE and thus better predictions relative to the OLS model. The LASSO appears to have the lowest mean MAPE with the Elastic Net being in between the Ridge and LASSO means.

The second comparison we would like to make is the following table, 2, with the R^2, MSE, and the mean MAPE for each of the models.

Model	R^2	MSE	Mean MAPE
OLS	0.5934	0.1550	16.13
Ridge	0.5833	0.0435	16.05
LASSO	0.5708	0.0448	16.35
Elastic Net	0.5790	0.0440	16.15

Table 2: R^2, MSE, and MAPE Comparison

This table shows us that the R^2 appears to remain relatively consistent across all 4 models (it does not fluctuate more than 5% for any of the models). The OLS does retain the highest R^2 which shows that it would fit the data the best of the 4 models. However, this is to be expected as one of the main features of regularized regression is to trade bias for prediction, i.e. we had a small decrease in R^2, but a noticeable decrease in MSE. The MSE shows a relatively significant decrease by decreasing over 0.10 from the OLS model to the regularized models. While the mean MAPE is the lowest when using Ridge regression, overall there is minimal improvement for the 3 regularization methods relative to the OLS.

It is reasonable to believe that either the LASSO or Elastic Net model may be a better model for the data set. The LASSO and Elastic Net give a simpler model without loss of necessary predictors. While at the same time preserving the R^2 value, MAPE, and minimizing the MSE.

Based on the models and comparisons it appears for this data the LASSO would be the best model. Therefore using LASSO the model for the Abalone data is:

$$log(Rings) \sim 1.3636 + 1.3104(Diameter) + 2.7427(Height) - 0.6414(Shucked) + 0.8319(Shell)$$

The LASSO has an R^2 0.5708, which suggests only a minimal decrease when compared to the OLS model.The LASSO model also had the second lowest MSE of 0.0448. While the LASSO model technically had the largest MAPE of 16.35, the difference between the worst and best MAPE was less than half a percent. The LASSO has the fewest predictors which allows it to be a simpler and more parsimonious model while still retaining the ability to accurately model and predict the data.

References

[1] Nicoguaro - Own work CC BY 4.0. *Forms of the constraint regions for lasso and ridge regression.* URL: https://commons.wikimedia.org/w/index.php?curid=58258966.

[2] *Elastic net regularization.* URL: https://en.wikipedia.org/wiki/Elastic_net_regularization.

[3] *Lasso (statistics).* URL: https://en.wikipedia.org/wiki/Lasso_(statistics).

[4] Aditya Singh. *Ridge Regression.* URL: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf.

[5] *The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait.* URL: https://archive.ics.uci.edu/ml/datasets/abalone.

[6] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. DOI: https://www.jstor.org/stable/2346178.

[7] Jerome Friedman Trevor Hastie Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2017.

[8] Hui Zou and Trevor Hastie. "Regularization and Variable Selection via the Elastic Net". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 67.2 (2005), pp. 301–320. DOI: https://www.jstor.org/stable/3647580.