Penalised Regression Methods and their Application in Financial Markets

Josh Kaura

April 2024

Declaration

This piece of work is a result of my own work except where it forms an assessment based on group project work. In the case of a group project, the work has been prepared in collaboration with other members of the group. Material from the work of others not involved in the project has been acknowledged and quotations and paraphrases suitably indicated.

Abstract

This report introduces the concept of penalised regression methods as a tool to overcome issues faced by ordinary least squares regression, including investigation into specific applications in real-world financial datasets.

Contents

1 Introduction **5**

2 Linear Regression **7**
2.1 Ordinary Least Squares Regression 7
2.2 Why Penalised Regression? 8
2.2.1 Dimensionality 9
2.2.2 Multicollinearity 9
2.2.3 Overfitting 12
2.3 What is Penalised Regression? 13

3 Penalised Regression Methods **15**
3.1 Ridge Regression 15
3.2 LASSO Regression 18
3.2.1 The LASSO 18
3.2.2 Ridge vs. The LASSO 19
3.2.3 Adaptive LASSO 22
3.2.4 Relaxed LASSO 22
3.3 Elastic Net Regression 23
3.4 Penalised Logistic Regression 24
3.4.1 Logistic Regression Model 24
3.4.2 Logistic Regression with LASSO and Ridge 25
3.5 The Bayesian Setting 25
3.6 Choosing λ 26
3.6.1 K-Fold Cross Validation 27
3.6.2 Nested Cross Validation 27

4 Simulations **31**
4.1 Motivation 31
4.2 Multicollinearity 31
4.2.1 Small Dimensions 31
4.2.2 Dense Data Generating Models 32
4.2.3 Sparse Data Generating Models 34
4.3 p close to n 36
4.3.1 n=3, p=2 36

4.3.2 p close to n in higher dimensions 37
4.4 Ridge, The Lasso, Elastic Net 38

5 Applications: Penalised Regression in Finance 41
5.1 Penalised Logistic Regression for Loan Status Classification . 41
5.1.1 The Dataset 41
5.1.2 The Model 42
5.1.3 Results 43
5.2 Stock Price Prediction 45
5.2.1 Stock Market Data 45
5.2.2 Results 47

6 Discussion 50

A Appendix A: R Code and Data 54

Introduction

Linear Regression is a statistical method used in a wide range of applications, such as in data analytics and machine learning. In the fields of statistical analysis, predictive modelling and machine learning, regression techniques serve as fundamental tools for understanding and modelling relationships between variables. Central to these methods lies Ordinary Least Squares (OLS) regression, renowned for its simplicity and efficacy in estimating the linear relationship between a dependent (response) variable and one or more independent variables. The primary motivation behind OLS regression is to minimise the sum of the squared differences between the observed and predicted values, which returns the line-of-best-fit through the data points in a linear sense.

However, despite its widespread application, OLS regression has limitations, particularly when dealing with high-dimensional data or when multicollinearity (a scenario where independent predictor variables are highly correlated) is present. These situations can lead to overfitting, where the model performs well on training data but poorly on unseen data, and can render the model estimates unstable and difficult to interpret.

Penalised regression methods (otherwise known as regularisation methods or shrinkage methods) include techniques such as Ridge Regression, the Lasso (Least Absolute Shrinkage and Selection Operator), and Elastic Net. These regression methods offer robust alternatives to OLS by introducing regularisation terms into the loss function that the methods seek to minimise. These regularisation terms penalise the size of the coefficients, which helps to prevent overfitting, improve model generalization to new data, and in some cases, aid in variable selection.

Ridge Regression introduces an l_2 penalty term, which is the square of the magnitude of coefficients. This method is particularly useful in mitigating the multicollinearity problem by shrinking the coefficients evenly, but it does not set any coefficients exactly to zero, which means it does not perform variable selection.

Lasso Regression, on the other hand, employs an l_1 penalty term, which is the absolute value of the magnitude of coefficients. This characteristic of the Lasso enables it to not only prevent overfitting but also to perform variable selection by setting some coefficients to zero, thus excluding some variables from the model entirely.

Elastic Net combines the penalties of Ridge and Lasso, making it particularly useful when dealing with highly correlated data. It blends the ability to perform variable selection with the ability to handle multicollinearity, offering a versatile and powerful modeling approach.

In the context of financial modelling, the predictive accuracy and interpretability of the model are paramount. Financial datasets are often complex, high-dimensional, and exhibit high correlations in the data, making penalised regression methods potentially viable models. These methods can enhance predictive performance and provide more reliable insights, which are crucial for credit risk assessment, price forecasting models, and many other financial applications.

The introduction of penalised regression methods represents a significant advancement in the field of statistical modelling, offering a more nuanced approach to dealing with the challenges posed by modern datasets. This report aims to delve into the intricacies of these methods, explore their theoretical underpinnings, demonstrate some of the predictive benefits and illustrate their practical applications in financial modeling, providing a comprehensive overview of how these techniques can be effectively employed to glean insights from complex financial data.

Linear Regression

2.1 Ordinary Least Squares Regression

The standard model of ordinary least squares (OLS) for multiple linear regression is perhaps the most widely understood, due to its simplicity and interpretability. Consider the standard model of multiple linear regression:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.1}$$

with $\boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^p, X \in \mathbb{R}^{nxp}$.

$\boldsymbol{y}$ is the response (dependent) variable; $\boldsymbol{X}$ is the predictor matrix, consisting of x_{ij} entries corresponding to the i^{th} of n total observations of each j^{th} of p total predictor variables (regressors); $\boldsymbol{\beta}$ is the coefficient vector, with each β_i controlling the influence of its corresponding predictor variable in the model; and ϵ is the error term.

The model terms expand to give:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \tag{2.2}$$

where the β_0 coefficient is commonly known as the intercept term.
The error term, ϵ, must conform to some key assumptions in order to ensure the accurate modelling power of OLS regression. These key assumptions are:

- $\mathbb{E}[\epsilon_i] = 0$: errors have mean 0 and do not depend on x
- $Var[\epsilon_i] = \sigma^2$: errors have a constant variance, are homoscedastic, and do not depend on x
- ϵ_i and ϵ_j are independent for all $i \neq j$
- $\epsilon_i \sim^{iid} N(0, \sigma^2)$: errors are independent and identically distributed as Normal with mean 0 and variance σ^2.

If these assumptions hold true, then the Gauss-Markov Theorem states states that OLS is the best unbiased linear estimator for the dataset.

However, though these assumptions may hold true in some scenarios, typically with small datasets, larger real-world datasets often do not conform, and hence OLS regression becomes a less powerful tool within data science and machine learning.
The β coefficients of the true, but unknown model are estimated by the OLS regression model, yielding $\hat{\beta}$ coefficients, by minimising the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n}(y_i - X_i\hat{\boldsymbol{\beta}})^2$$

A closed form expression for the sum of square errors estimate for least squares regression is therefore:

$$(Y - X\beta)^T(Y - X\beta) \tag{2.3}$$

Yielding a closed form expression for the ordinary least squares model coefficients:

$$\hat{\beta} = (X^TX)^{-1}X^TY \tag{2.4}$$

Theorem: The ordinary least squares regression coefficients are unbiased, that is $\mathbb{E}[\hat{\beta}] = \beta$
Proof: Given the ordinary leasy squares coefficient estimate in Equation 2.4,

$$\mathbb{E}[\hat{\beta}] = \mathbb{E}[(X^TX)^{-1}X^TY]$$

$$= (X^TX)^{-1}X^T\mathbb{E}[Y]$$

Noting that $\mathbb{E}[Y] = X\beta$:

$$\mathbb{E}[\hat{\beta}] = (X^TX)^{-1}(X^TX)\beta$$

$$= \beta$$

□

2.2 Why Penalised Regression?

To see why penalised regression methods are so widely used, understanding of the shortcomings of ordinary least squares regression is key. Some of the negative impacts of specific modelling issues that would ideally be eliminated, or at least reduced, are very common in real-world, often high-dimensional datasets.

When looking at the predictive ability of linear regression, one can decompose it's prediction error into square bias and variance. Least squares regression possesses the zero-bias property, but can display high variance in some scenarios. Reasons for high variance can come from various different modelling intricacies.

2.2.1 Dimensionality

As expressed before, real world-high dimensional datasets often violate the assumptions about ϵ that make ordinary least squares so powerful. The dimensionality of not only the true linear regression model, but also the dataset used to fit a linear regression model, is key to ensure statistical accuracy.

Ordinary least squares regression will include all p explanatory features (predictor variables). When there are large number of predictors, a smaller subset of $q < p$ particularly important predictors may be desirable. Not only does this aid in the interpretability of the linear regression model, but also this feature selection proposal could eliminate "noise" caused by less relevant predictors.

It is also important note that linear regression is not well defined when $p > n$, that is when the number of predictors is greater than the number of observed data points.
Equation 2.4 expresses the matrix equation for OLS estimates:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{2.5}$$

Noting that $\boldsymbol{Y} \in \mathbb{R}^n, \boldsymbol{\beta} \in \mathbb{R}^p, X \in \mathbb{R}^{nxp}$, it is clear that for $p > n$, $X^T X$ (the matrix to invert) is singular, so there exists no unique solution. This leads to problems fitting an accurate model to data consisting of lots of predictor variables and not many observations.

2.2.2 Multicollinearity

Definition. **Multicollinearity** is a statistical phenomenon that occurs when two or more independent variables in a regression model are highly correlated with each other.

In the context of Ordinary Least Squares (OLS) regression, multicollinearity can pose significant problems, affecting the precision of the estimates of the model's coefficients, which in turn can impact the interpretability and the reliability of the model.

One of the key assumptions made for our least squares regression model is that the errors, ϵ_i are independent of each other. Therefore, multicollinearity leads to a less accurate regression model, as highly correlated variables can cause to much "noise" in the model, which can lead to overfitting (see Section 2.2.3 .

Effects of multicollinearity on OLS regression:

- **Inflated Variance:** Multicollinearity increases the variance of the coefficient estimates, which means that the estimates of the coefficients become less precise. High variance can make the model coefficients unstable, where small changes in the data can lead to large changes in the model coefficients and hence extrapolated (predicted) values.
- **Unreliable Statistical Inferences:** Due to the inflated variances, the confidence intervals for the coefficient estimates can become very wide, and hypothesis tests (like t-tests for individual regression coefficients) may not be reliable. This can lead to difficulties in determining which independent variables are statistically significant predictors of the dependent variable.
- **Model Interpretability:** Multicollinearity can make it difficult to identify interpret the effects of individual predictor variables. When variables are highly correlated, it becomes challenging to distinguish their individual contributions to the response variable. Some or all predictors could become insignificant when they should be significant because of inflation in standard error for coefficients [2].

Testing for multicollinearity:
There are several ways to test for correlations between variables in a dataset:

- **Correlation Matrix:** One could examine the correlation matrix for the independent variables. A popular type of correlation calculation is the Pearson correlation coefficient, r:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{2.6}$$

  With x and y the two variables for which the correlation relationship is being analysed. High correlation coefficients (near -1 or 1) between lots of pairs of independent variables indicates potential multicollinearity in the dataset.
  Figure 2.1 (Left Panel) plots a heatmap of Pearson Correlation coefficients between independent numerical predictor variables in the **Credit** dataset within the **ISLR** package in R [7]. We see for this credit scoring dataset that there are clear correlations between **Rating, Income, Balance** and **Limit** explanatory variables.

- **Variance Inflation Factor (VIF):** VIFs quantify how much the variance of an estimated regression coefficient increases if your predictors are correlated, that is in comparison to if R_i^2 equaled zero: detailing the scenario in which the i^{th} independent variable is orthogonal to the other independent variables in the analysis. [15]. Variance Inflation Factor is determined with the following calculation:

$$VIF = \frac{1}{1 - R_i^2} = \frac{1}{Tolerance} \tag{2.7}$$

  Where R_i is the unadjusted coefficient of determination for regressing the i^{th} independent variable on the remaining ones, and the reciprocal of the VIF is known as tolerance.
  If no factors are correlated, the VIFs will all be equal to 1. In practice, $VIF >> 5$ suggests high multicollinearity in a dataset [2].
  The right panel of Figure 2.1 clearly shows that calculated VIFs of **Income, Balance** and **Limit** against the response **Rating** are > 5, and hence are likely contributing to multicollinearity within the credit scoring dataset.

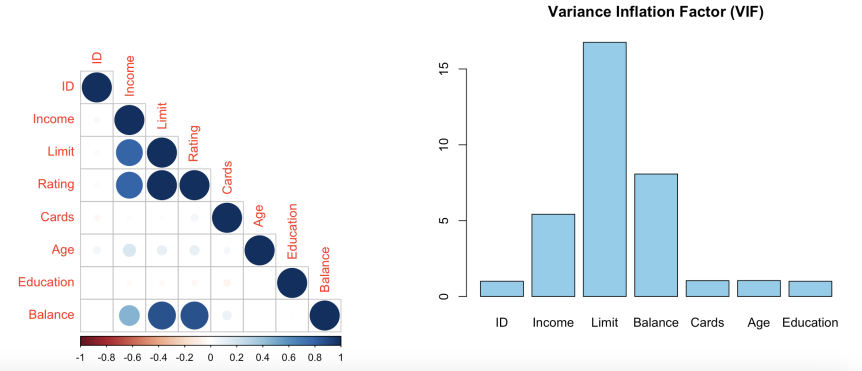

Figure 2.1: Left Panel: Pearson correlation coefficients plotted for each pair of numeric predictor variables in the **Credit** dataset. Right Panel: Bar chart showing the Variance Inflation Factor calcualated for each explanatory variable against the dependent **Rating** variable in the **Credit** dataset. R code found in A.

Addressing multicollinearity:
Several approaches can be taken to mitigate its effects:

- **Removing Variables:** Eliminate one or more of the highly correlated independent variables.
- **Combining Variables:** Combine highly correlated variables into a single predictor through techniques like Principal Component Analysis (PCA).

- **Regularisation:** Use penalised regression methods like Ridge regression or Lasso, which are designed to handle multicollinearity by shrinking or completely eliminating the coefficients of correlated predictors.

2.2.3 Overfitting

Definition. Overfitting occurs when the regression model fits too closely the training data, so performs poorly on unseen data [21]. This is often due to low bias, but high variance and an overly complex model, which leads to a model that is not well-generalised.

Effects of overfitting:

- **Poor generalisation:** Overfitted models reflect the noise and anomalies in the data (often caused by highly correlated variables), rather than the overall population. This means means that predictions made by the model on new data can be inaccurate and unreliable.
- **Complexity and Interpretability:** Overfitted models are often unnecessarily complex, including many predictors that may not be relevant to the underlying relationship being modeled. This complexity can make the model difficult to interpret and understand.

In a more mathematical context, we can view overfitting consequences by comparing how well a model performs on a training set in comparison to a testing set. One of the ways we do this is by calculating the mean squared error (MSE) of our regression model:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i}^2) \tag{2.8}$$

Where y_i is the true value of the response variable in the training/ testing set, and $\hat{y_i}$ is the estimated value predicted by the regression model, after fitting the corresponding x_i predictor variable values to their corresponding β_i coefficients.
When overfitting is present, the MSE calculated on the training set via Equation 2.8 tends to be low, but the MSE on the testing set is much higher, meaning ˆ predictions on unseen data are far less reliable.

Addressing Overfitting:

- **Simplifying the Model:** Reduce the complexity of the model by removing irrelevant features or using fewer parameters.

- **Cross-Validation:** Use cross-validation (see Section 3.5 to ensure that the model's ability to generalize is not due to the specific way the data was split.
- **Penalised regression:** Just like for multicollinearity, regularization techniques like Ridge and Lasso can reduce the risk of overfitting, as they add a penalty term to the loss function to constrain the size of the coefficients, effectively reducing model complexity.

2.3 What is Penalised Regression?

A penalised regression method is essentially a method of shrinking down a subsection of the $\hat{\beta}$ coefficients of the OLS regression model, in order to reduce the impact of features that are not as relevant to the model. Penalised regression methods are therefore sometimes known as 'shrinkage' methods, which force the regression model to shrink its coefficients towards 0 due the 'penalty' term imposed on its coefficients.

Recall that ordinary least squares (OLS) Regression selects predicted values $\hat{\beta}$ in order to minimize the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{n} (y_i - X_i\hat{\boldsymbol{\beta}})^2$$

Non-OLS regression selects coefficients in order to minimise a similar objective function.
Specifically, penalised regression adds a **penalty term** (also known as a regularisation term or shrinkage term),

$$\lambda ||\beta||_p \tag{2.9}$$

- $||\beta||_p$ is the p-norm of the coefficients: $\sum_i (|\beta_i|^p)^{\frac{1}{p}}$
- $\lambda > 0$ is a hyper-parameter, in this case known as the tuning parameter, defining how harshly the coefficients are penalised.

The aim is to now fit a penalised regression model to minimise the regularisation cost function:

$$\sum_{i=1}^{n} (y_i - X_i\hat{\beta})^2 + \lambda ||\beta||_p \tag{2.10}$$

Yielding penalised regression coefficients,

$$\hat{\beta}_p = \arg\min_{\beta} ||y - X\beta||_2^2 + \lambda ||\beta||_p^p \tag{2.11}$$

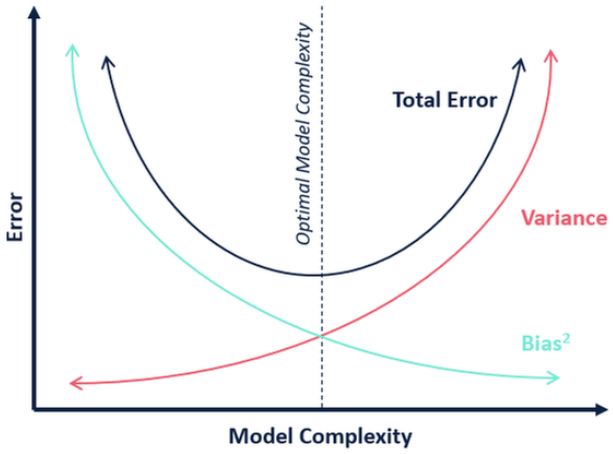

Figure 2.2: Figure showing the relationship between bias, variance, model complexity and mean squared error [3].

In the process of shrinking the coefficients of predictors deemed to be less relevant in the model, penalised regression methods introduce some **bias** into our regression model, in order to reducing **variance**.
The aim is tuning λ to find the optimum model by balancing the **trade-off** between bias and variance, as can be seen in Figure 2.2.

Penalised Regression Methods

This report will focus on 3 commonly used penalised regression methods:

- Ridge regression: $L_2 = (\sum_i \|\beta_i\|_2^2)^{\frac{1}{2}}$ penalty term
- LASSO regression: $L_1 = (\sum_i |\beta_i|)$ penalty term
- Elastic net regression: a weighted mix of LASSO and Ridge penalties

3.1 Ridge Regression

Ridge regression, first introduced by Hoerl and Kennard, 1970 [6] employs the L_2 regularisation term, in order to penalise the squares of the regression coefficients. Firstly, the setup of the environment within which ridge regression can be performed.

- Start with fixed independent covariates (predictor variables) $x_i \in \mathbb{R}^p, i = 1, ..., n$
- Observe $y_i = f(x_i) + \epsilon_i, i = 1, \ldots, n$
- $f : \mathbb{R}^p \to \mathbb{R}$ unknown
- $Var[\epsilon_i] = \sigma^2$

The Ridge regression is similar to Least Squares, but shrinks estimated coefficients towards 0 that the model deems to be less relevant to the response.
Given response vector $y \in \mathbb{R}^n$ and predictor matrix $X^{n \times p}$, the ridge coefficients are defined as:

$$\hat{\beta}_{ridge} = argmin||y - X\beta||_2^2 + \lambda||\beta||_2^2 \tag{3.1}$$

Where $\lambda \geq 0$ is the tuning parameter for the penalty, which is determined by K-Fold Cross Validation (see Section 3.6. λ determines the amount of shrinkage performed by the ridge regularisation term.
Usually when including the intercept term in the regression equation, its corresponding coefficient is not penalised, as centreing the columns of $\boldsymbol{X}$

solves to find the intercept $\hat{\beta}_0 = \bar{\boldsymbol{y}}$.
Hence, we can interpret the ridge regression as yielding ridge coefficients that minimise the corresponding cost function

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2 \tag{3.2}$$

Given the closed form expression for ordinary least squares in Equation 2.4, a closed form expression for the ridge coefficients can also be derived. Given the expression for ordinary least squares coefficients and introducing the ridge L_2 penalty term:

$$(Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta \tag{3.3}$$

the closed form expression for ridge coefficients can be derived.

$$(Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta$$

$$= (Y^T - \beta^T X^T)(Y - X\beta) + \lambda\beta^T\beta$$

Expanding out the brackets:

$$= Y^TY - Y^TX\beta - \beta^TX^TY + \beta^TX^TX\beta + \lambda\beta^T\beta$$

$$= Y^TY - 2Y^TX\beta + \beta^TX^TX\beta + \lambda\beta^T\beta$$

Taking the derivative and setting equal to zero to find the minimum:

$$\frac{d}{d\beta} = 0 - 2Y^TX + 2X^TX\beta + 2\lambda\beta = 0$$

And finally rearranging to find the ridge coefficient estimates:

$$\implies \hat{\beta}_{ridge} = (X^TX + \lambda I)^{-1}X^TY$$

Note: to impose the penalty terms with any shrinkage method, all predictor variables must be standardised, otherwise the magnitude of the coefficients will be skewed. For example, a feature in a smaller scale will be assigned a coefficient disproportionately large compared to features on a larger scale, and vice versa.
In practice, predictors in a dataset are all scaled to have variance 1, by dividing each predictor variable by its standard deviation:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n}\frac{1}{n}(x_{ij} - \bar{x}_{ij})^2}} \tag{3.4}$$

Theorem: (Existence theorem) For ridge regression, there always exists a

$\lambda > 0$ such that the MSE is less than that of the least squares estimate $\lambda = 0$.
Proof: proof of the theorem can be found in Hoerl (1970) [6]

This proof provides evidence that is the process of fitting a model to the training set, Ridge regression always can find a regularisation term that will be better than that of OLS regression.

Inspecting ridge coefficients in both panels of Figure 3.1 for the same credit scoring as used in , as λ increases, so does the penalty on the regression coefficients, as their magnitude decreases. Conversely, as the value of L_1 norm of the coefficients increases, the regularisation term decreases and hence less shrinkage occurs.
It is important to note that as the squares of the coefficients are penalised, the ridge coefficients, $\hat{\beta}_{ridge}$ may get close to 0, but can never equal 0. Therefore, ridge regression models contain all p predictors. True models that include all p predictors, or very close to all p predictors, are known as **dense** data generating models.

Theorem: Ridge coefficients are biased estimators.
Proof: This proof can be extended from the proof of least squares estimators being unbiased:

$$\mathbb{E}[\hat{\beta}_{ridge}] = \mathbb{E}[(X^T X + \lambda I)^{-1} X^T Y]$$
$$= \mathbb{E}[(X^T X + \lambda I)^{-1} (X^T X)(X^T X)^{-1} X^T Y]$$
$$= \mathbb{E}[(X^T X + \lambda I)^{-1} (X^T X)] \mathbb{E}[(X^T X)^{-1} X^T Y]$$

Noting that $\mathbb{E}[(X^T X)^{-1} X^T Y] = \mathbb{E}[\hat{\beta}_{ols}] = \beta$:

$$\mathbb{E}[\hat{\beta}_{ridge}] = \mathbb{E}[(X^T X + \lambda I)^{-1} (X^T X)]\beta$$

Hence bias is introduced, related to λ □
This demonstrates that λ controls the trade-off between bias and variance in the ridge regression model. Through ridge regression, users determine an acceptable loss in training accuracy (higher bias) in order to increase a given model's generalisation (lower variance). [8]

Ridge Advantages
Ridge regression's penalty term, whilst introducing some bias, also introduces major advantages over OLS regression for certain dataset features.

- Ridge regression can deal with $n < p$ problems. This is observed through analysis of the closed form matrix expression for Ridge coefficients:

$$\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T Y \tag{3.5}$$

Even if $\boldsymbol{X}^T\boldsymbol{X}$ is singular, the matrix term for which the inverse is required is non-singular for $\lambda > 0$, hence there exists a unique solution to the equation. The introduction of the penalty term

- Shrinks coefficients hence reduces the impact of predictor variables that are not relevant to the response variable. This is key in mitigating the impact of multicollinearity, as shrinkage is imposed to decrease the impact of noise caused by correlated variables.
- Decreases variance, leading to improved predictive performance on unseen data.

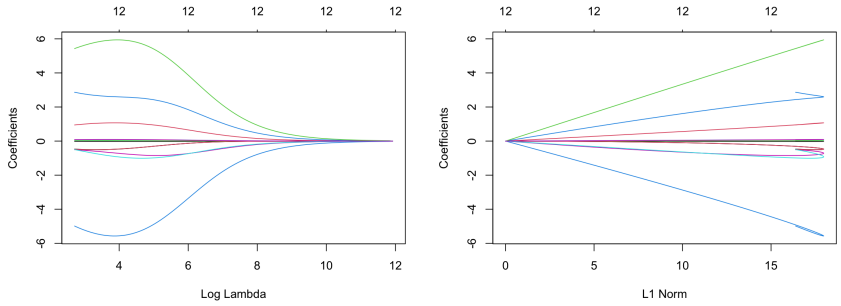

Figure 3.1: Trace plotd showing ridge coefficients against tuning parameter, λ, (left panel) and the L_1 norm (right panel) for the simulated credit scoring dataset [7]. Note that coefficients converge to 0 but never equal 0.

3.2 LASSO Regression

3.2.1 The LASSO

The LASSO (least absolute selection and shrinkage operator), first proposed by Robert Tibshirani [18] is the other main regularisation method. Where ridge doesn't set any coefficients exactly to 0, the L_1-penalty imposed by the lasso means that it can, in fact, perform **variable selection** in the linear model. This feature selection property is a key feature in correcting multicollinearity. [12]
Definition: The lasso estimates are defined as:

$$\hat{\beta}_{lasso} = argmin||y - X\beta||_2^2 + \lambda||\beta||_1 \tag{3.6}$$

which minimise the quantity

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum|\beta_j| \tag{3.7}$$

where, like ridge, $\lambda > 0$ is a tuning parameter, selected via Cross-Validation (Section 3.6).

Though there is no closed form expression available to calculate LASSO coefficients, The LASSO has similar advantages to ridge:

- Can deal with $n < p$ problems.
- Shrinks coefficients to reduces the impact of predictor variables that are not relevant to the response.
- Decreases variance by increasing bias, leading to improved predictive performance (on unseen data).

However, there are key differences between LASSO and Ridge regression that dictate the more desirable model depending on the dataset being learned from.

3.2.2 Ridge vs. The LASSO

Analysis of the variable selection feature of the LASSO leads into one of the key differences between LASSO and ridge regression:

- When actual data-generating mechanisms are dense, and the true data generating model includes lots of/ all p predictors, ridge regression is generally more accurate, as ridge models include all p predictors.
- When actual data-generating mechanisms are sparse, where the true data generating model includes a smaller subset of $q < p$ predictors, LASSO regression is generally more accurate, as the feature selection property yields sparse models so is more likely to be close to the true model.

Figure 3.2 (left panel) demonstrates the relationship between $\hat{\beta}_l asso$ coefficients and lambda. The variable selection feature is clear here: as λ increases, the number of coefficients shrunk to zero increases, decreasing the number of predictors included in the model and introducing a sparse, simpler model.

Ridge and lasso regression can also be formulated as constrained optimisation problems:

Ridge:

$$(Y - X\beta)^T(Y - X\beta) \text{ such that } \beta^T\beta \leq t$$

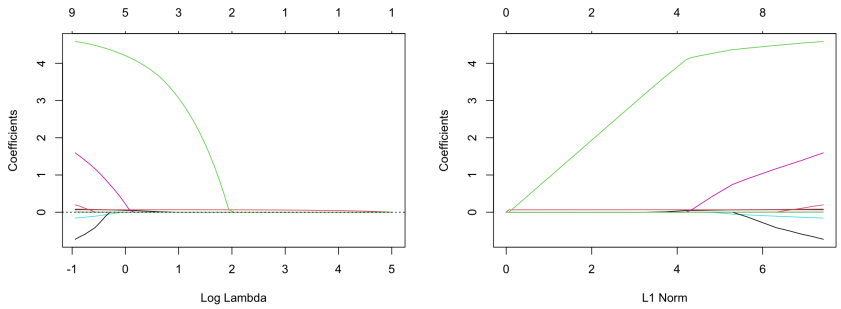

Figure 3.2: Trace plots showing lasso regression coefficients against tuning parameter, λ, (left panel) and L_1 norm (right panel) for the simulated **credit** dataset [7].

$$\implies \text{minimise} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}) \text{ subject to} \sum_{j=1}^{p} \beta_j^2 \leq t$$

LASSO

$$(Y - X\beta)^T (Y - X\beta) \text{ such that } |\beta| \leq t$$

$$\implies \text{minimise} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}) \text{ subject to} \sum_{j=1}^{p} |\beta_j| \leq t$$

For every $\lambda > 0$, there exists a t such that the unique solutions for Ridge and LASSO exist.
The constrained optimisation formulation demonstrates the differences in the L_2 and L_1 penalty terms. With 2 predictors, the $\hat{\beta}$ providing the minimum RSS, along with the penalty region defined by the conditions the optimisation problems are constrained by, can be plotted in order to observe how Ridge and LASSO regression select $\hat{\beta}$ differently. Figure 3.3 illustrates this, where the LASSO (left panel) has its penalty region constrained by $|\beta_1| + |\beta_2| \leq t$ and ridge (right panel) instead has its penalty region bound by the circle $\beta_1^2 + \beta_2^2 \leq t$.

In the special case of $n = p$, that is the number of observations is the same as the number of proposed predictors, $\boldsymbol{X}$ is the identity matrix (leading diagonal of 1s and 0s elsewhere, and the intercept $\beta_0 = 0$, the $\hat{\beta}$ coefficients for both ridge and LASSO regression can be computed explicitly. With this setup, OLS regression minimises $\sum_{j=1}^{p} (y_j - \beta_j)^2 \implies \hat{\beta}_j^{ols} = y_j$ Computing ridge coefficients:

$$\text{aim to minimise:} (y_j - \beta_j)^2 + \lambda \beta_j^2$$

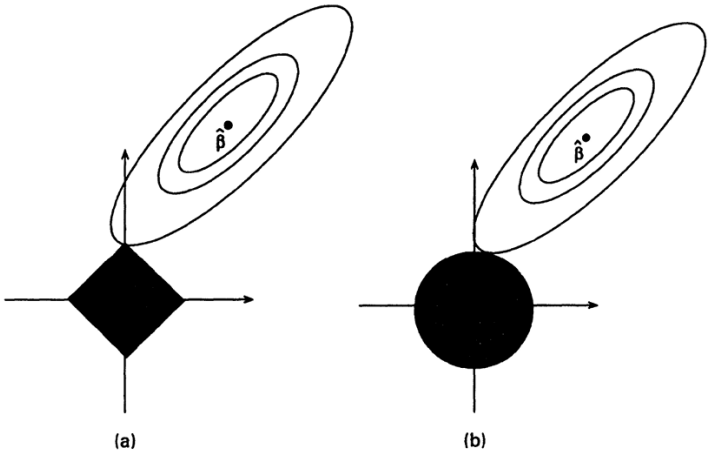

Figure 3.3: Figure taken from [18], showing graphically how LASSO (a) and ridge (b) coefficients are selected when balancing the RSS of the Least Squares coefficients (contours lines) with the respective penalty terms (black constraints). This illustrates why LASSO can select $\hat{\beta}_{lasso} = 0$, whilst $\hat{\beta}_{ridge}$ only get close to 0.

$$\begin{aligned} &= y_j^2 - 2y_j\beta_j + \beta_j^2 + \lambda\beta_j^2 \\ \frac{d}{d\beta} = 0 &= -2y_j + 2\hat{\beta}_j^{ridge} + 2\lambda\hat{\beta}_j^{ridge} \\ \implies \hat{\beta}_j^{ridge} &= \frac{y_j}{1+\lambda} \end{aligned} \tag{3.8}$$

To compute the LASSO coefficients:

$$\begin{aligned} \text{aim to minimise:} &(y_j - \beta_j)^2 + \lambda|\beta_j| \\ &= y_j^2 - 2y_j\beta_j + \beta_j^2 + \lambda|\beta_j| \\ \frac{d}{d\beta} = 0 &= -2y_j + 2\hat{\beta}_j^{lasso} + \frac{d}{d\beta}\hat{\beta}_j^{lasso} \end{aligned}$$

Differentiating piecewise:

$$\hat{\beta}_j^{lasso} = \begin{cases} y_j - \frac{\lambda}{2} & \text{if } y_j > \frac{\lambda}{2} \\ y_j + \frac{\lambda}{2} & \text{if } y_j < -\frac{\lambda}{2} \\ 0 & \text{if } |y_j| \leq \frac{\lambda}{2} \end{cases}$$

The coefficient estimates demonstrate the difference between the types of shrinkage. The Ridge regression shrinks the OLS estimate to close to 0 as λ increases, whereas the LASSO penalty shrinks the coefficients at a constant rate, unless

$$|y_j| = |\hat{\beta}_j^{ols}| \leq \frac{\lambda}{2}$$

in which case the L_1 penalty pushes the coefficient to exactly zero.

3.2.3 Adaptive LASSO

Since the introduction of the LASSO, multiple versions of the LASSO have been introduced in an attempt to further optimise shrinkage methods, and the bias-variance tradeoff. One of which is the adaptive LASSO.

Definition: The adaptive LASSO coefficients are defined to be [22]:

$$\hat{\beta} = \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda \sum_{j=1}^{p} w_j |\beta| \tag{3.9}$$

This yields estimates for coefficients using least squares, and then performs variable selection via the LASSO, by assigning weights, $w_j = \frac{1}{\hat{\beta}_j}$. As $\hat{\beta}_j$ increases, the weights w_j decrease, meaning the penalty imposed upon smaller coefficients is greater (so shrink to 0 more quickly), whereas larger coefficients have a smaller bias after shrinkage.

One of the most fascinating aspects of the adaptive lasso is that it holds **oracle properties** [22].
The oracle property essentially states that as $n \to \infty$, the adaptive LASSO sets all the correct coefficients to zero of predictor variables in the model that are not relevant. That is, say we have some set, S, of predictor indices for the correct model of relevant predictors:

$$S = \{j \in 1, ..., p : \beta \neq 0\} \tag{3.10}$$

Then say we have some set S_λ of estimated coefficients deemed to not be equal to zero by some regularization method, that is:

$$S_\lambda = \{j \in 1, ..., p : \hat{\beta} \neq 0\} \tag{3.11}$$

If this regularization method were to have the oracle property, then

$$S_\lambda = S, n \to \infty \tag{3.12}$$

The adaptive lasso is said to have this oracle property asymptotically. Intuitively, it could be thought that for large enough λ would have this property, however upon studying various regularization methods and their variable selection properties, Fan and Li (2006) [4] suggested that the LASSO does not have this oracle property.

3.2.4 Relaxed LASSO

The relaxed LASSO works on the basis of performing variable shrinkage and variable selection separately.

Generally, LASSO regression is performed first to perform variable selection and obtain a more sparse model of relevant predictors, and then another regression method (e.g a combination of least squares and the lasso) is performed to fit the final model:

$$\hat{\beta}^{\lambda,\phi} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \sum_{j} \beta_j x_{ij})^2 + \lambda\phi \sum_{j=1}^{p} |\beta_j| \tag{3.13}$$

3.3 Elastic Net Regression

Elastic net regression provides a weighted balance between ridge and the LASSO, by adding the mixing parameter $\alpha \in (0, 1)$, to incorporate both l_1 and l_2 norm penalties into the RSS cost function. [23]. The elastic net regression coefficients are therefore obtained via the following formulation:

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||_2^2 + \alpha\lambda||\beta||_1 + \frac{1-\alpha}{2}\lambda||\beta||_2^2 \tag{3.14}$$

Note: both λ and α are selected by **cross validation**.

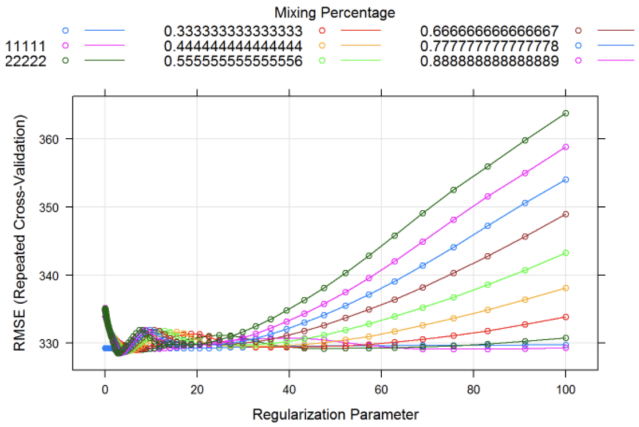

Figure 3.4: Figure showing RMSE calculated against tuning parameter, λ, with each coloured line representing a different mixing percentage, α, which controls the balance between the ridge and LASSO penalty within the model.

Advantages of Elastic Net Regression:

- Grouping Effect: Elastic Net can handle the 'grouping effect' more effectively than Lasso. The grouping effect corresponds to situations where several predictors are highly correlated, where consequently Lasso tends to select one variable from a group and ignore the others. Elastic Net, by linearly combining the l_1 and l_2 penalties, can select

groups of correlated variables, which is often more interpretable and desirable in practice.

- Stability in High Dimensions: In high-dimensional settings where the number of predictors greatly exceeds the number of observations, Lasso can behave erratically, whereas Ridge regression does not perform variable selection. Elastic Net provides a more stable and consistent approach by blending the features of both, leading to better performance in variable selection and coefficient shrinkage.

- Sparse Models: Like Lasso, Elastic Net encourages sparsity in the model, but with greater flexibility due to the additional l_2 penalty. This can lead to more accurate and interpretable models, especially in contexts where the true underlying model is believed to depend on only a subset of the available predictors.

3.4 Penalised Logistic Regression

Penalised logistic regression extends the traditional logistic regression model by introducing a penalty term to the loss function. Like the effect of the penalty terms on OLS regression, this adjustment helps to prevent overfitting, manage high-dimensional data, and improve model generalisation where binary outcomes are required.

3.4.1 Logistic Regression Model

Whereas OLS regression returns outcomes for a dependent variable that can be of any arbitrary magnitude, the basic logistic regression model predicts a binary outcome by modelling the log-odds of the dependent variable as a linear combination of the independent variables. The probability p_i that an event occurs is given by

$$p_i = \frac{1}{1 + e^{-X_i\beta}} \tag{3.15}$$

where X_i represents the predictor variables, and X_i represents the coefficients. The model is fitted by maximizing the likelihood function:

$$\mathbb{L}(\beta) = \prod_{i=1}^{N} p_i^{y_i}(1 - p_i)^{1-y_i} \tag{3.16}$$

where y_i are the binary outcomes, i indexes each observation and N is the total number of observations. Hence, the logistic regression model calculates the probability of the event $y_i = 1$ based on the predictors X_i.

3.4.2 Logistic Regression with LASSO and Ridge

In Ridge logistic regression, the l_2 penalty term is imposed, and the objective is the minimise the penalised cost function:

$$-[\sum_{i=1}^{N} y_i log(p_i) + (1-y_i) log(1-p_i)] + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3.17}$$

In LASSO logistic regression, the absolute values of β are penalised, and the objective is the minimise the cost function:

$$-[\sum_{i=1}^{N} y_i log(p_i) + (1-y_i) log(1-p_i)] + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.18}$$

3.5 The Bayesian Setting

: Ridge regression and the lasso can be viewed through a Bayesian lens. [7]. The Bayesian regression setting assumes a prior distribution, $p(\beta)$ for the coefficient vector, β. The likelihood of the data, expressed as $f(Y|X,\beta)$, details the likelihood function of the response vector Y, given the data X and coefficients β. Following Bayes' Theorem, and assuming X is fixed, we yield the form of posterior ditirbution up to a constant of proportionality

$$p(\beta|X,Y) \propto f(Y|X,\beta)p(\beta) \tag{3.19}$$

The usual linear model is assumed, with errors being independent and identically distributed from a normal distribution.
The assumed distribution of the prior, $p(\beta)$, dictates the specific penalised regression method corresponding to the posterior mode of the model. Specifically, assuming a density function $g(\bullet)$ such that $p(\beta) = \prod_{j=1}^{p} g(\beta_j)$, then

- If g corresponds to a Gaussian distribution with mean 0 and standard deviation a function of λ, that is

  $$\beta_j \sim N(0, \frac{\sigma^2}{\lambda}) \tag{3.20}$$

  Then it follows that the posterior mode for β is given by the ridge solution.
  This prior encodes the belief that, a priori, the coefficients are likely to be close to zero, with the strength of this belief controlled by the regularization parameter λ. The larger the value of λ, the stronger the belief that the coefficients are small, leading to more significant shrinkage.

- If g corresponds to a Laplace (double exponential) density, that is

$$p(\beta_j|\lambda = \frac{\lambda}{2}exp(-\lambda|\beta_j|) \tag{3.21}$$

  then it follows that the posterior mode of β is given by the lasso solution.
  The Laplace prior is sharply peaked at zero and has heavier tails than the Gaussian distribution. This characteristic encourages sparsity in the coefficients, β, with many coefficients pushed exactly to zero when λ is sufficiently large. The sparsity property makes Lasso particularly useful for variable selection in high-dimensional datasets where only a subset of predictors is believed to be associated with the response variable.

In summary, the Bayesian interpretation of Ridge and Lasso regression offers a probabilistic perspective on regularisation. Ridge regression assumes Gaussian priors on the coefficients, leading to shrinkage towards zero, while Lasso assumes Laplace priors, encouraging sparsity by pushing many coefficients exactly to zero. This interpretation connects the choice of penalty in penalized regression methods to prior beliefs about the nature of the regression coefficients and provides a principled framework for understanding and applying these methods when such prior distribution conditions may be assumed.

3.6 Choosing λ

The performance of penalized regression methods such as Ridge, Lasso, and Elastic Net heavily depends on the choice of the tuning parameter(s), which control the strength of the penalty applied to the model coefficients. Selecting the optimal value of these parameters is crucial for balancing the bias-variance trade-off and achieving the best predictive performance.
Firstly, noting the results of the two extremums of λ:

- $\lambda = 0 \implies \hat{\beta}_{ridge}, \hat{\beta}_{lasso} = \hat{\beta}_{ls}$, the least squares regression coefficients.
- $\lambda \to \infty \implies \hat{\beta}_{ridge} \to 0_+, \hat{\beta}_{lasso} = 0.$

Generally speaking, as the tuning parameter increases, the penalty imposed increases and hence both ridge and lasso estimate coefficients decrease.

For other regression methods, there are multiple different selection criterion that one can attempt to minimise, such as $C_p, AIC, BIC, AdjustedR^2 etc.$ However, these criterion depend on the dimensionality of the regression model, which is not pre-determined in all regularisation methods.
Therefore, to tune λ in practice, K-Fold Cross Validation is used.

3.6.1 K-Fold Cross Validation

K-Fold Cross Validation is executed on a data-set via the following algorithm, and as seen in Figure 3.5:

Algorithm : (K-Fold Cross Validation)

1. Split data into K number of folds (often K=10 is chosen).
2. Define a grid of possible λ values
3. For each λ, compute the MSE, as in Equation 2.8, on each i^{th} fold (validation set), and then compute the overall MSE for this λ value.
4. Average out the MSE for each fold
5. Select the value of λ for which the average MSE is the smallest

Figure 3.5: Figure showing graphically the K-Fold Cross Validation Algorithm.

Note: Generally speaking, λ is selected to minimise the MSE, however data scientists may select a slightly offset *lambda*. In the **glmnet** package in R, one of the hyperparameter outputs is the λ which outputs the simplest model possible within 1 standard error of the minimum MSE, in order to counteract overfitting ($\lambda > \lambda_{min.CV}$ selected) or underfitting ($\lambda < \lambda_{min.CV}$ selected) of the model. The affect of this on training MSE can be seen in Figure 3.7.

3.6.2 Nested Cross Validation

While standard cross-validation is effective for tuning parameter selection, it can introduce bias when used simultaneously for model selection and performance estimation. Nested cross-validation (NCV) addresses this issue by

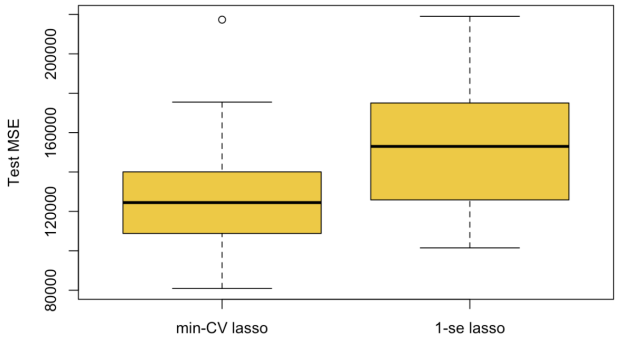

Figure 3.6: Figure showing how the selection of λ affects mean squared error of a simulated ridge regression [3].

providing a significantly less biased evaluation of the model's performance. [19]

Time Series λ Analysis:
In the specific setting of time-series data analysis (common in financial models such as stock market forecasting), standard k-fold cross validation should not be used. To simulate a real-world forecasting environment, that is from the viewpoint of the present and predicting the future, the forecaster must withhold all data about events that occur chronologically after the events used for fitting the model [17].
Since the test set data comes chronologically after the training set for k-fold cross validation, seen in Figure 3.5, a method is required that does not have chronological bias. One such method is utilising hold-out cross-validation where a subset of the data (split temporally) is reserved for validating the model performance.

In addition, the fairly arbitrary choice of test set for k-fold cross validation means that the calculated test set error could be a poor estimate of error on an independent test set.
Nested cross-validation utilises an inner loop for parameter tuning and an outer loop for error analysis, as outlined in the Algorithm below and illustrated in Figure 3.8 , providing a possible method for a more appropriate choice of λ constant for fitting penalised regression models to time series data.

Algorithm: (Nested Cross Validation)

1. Outer Loop: The dataset is split into k_{outer} folds. Each fold in turn

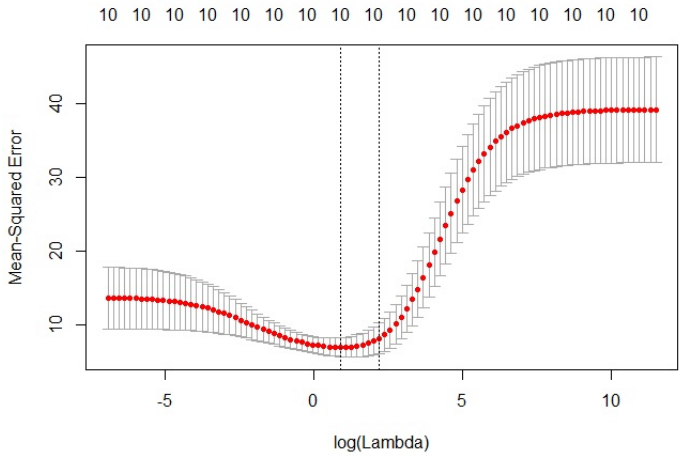

Figure 3.7: Figure showing how the mean squared error changes as λ increases for a simulated ridge regression.

is used as a test set, with the remaining $k_{outer} - 1$ folds used for the inner loop.

2. Inner Loop: Within each outer fold, the data is further divided into k_{inner} folds for the purpose of tuning parameter selection, following the cross-validation process described in 3.6.1 .

3. Model Training and Testing: For each outer fold, the model is trained on the inner loop's data with the optimal tuning parameters selected from the inner CV and then tested on the outer test fold. This provides an unbiased estimate of the model performance

One of the main advantages of nested CV is unbiased performance estimation: NCV allows for an unbiased estimate of the model's predictive performance on new data by separating the data used for tuning parameter selection from the data used for performance evaluation. Another advantage is robust model selection: NCV is particularly useful when comparing multiple models or sets of tuning parameters, as it ensures that the performance estimation is not overly optimistic.

In conclusion, selecting the tuning parameter through cross-validation is essential for optimizing the performance of penalized regression models. Nested cross-validation further enhances this process by providing a more reliable estimate of the model's ability to generalize, making it a valuable tool in scenarios involving multiple models or when an unbiased performance estimate is crucial.

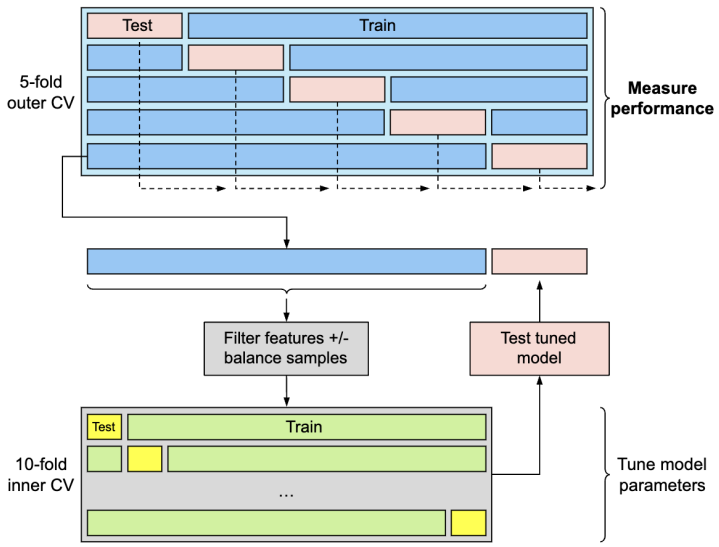

Figure 3.8: Figure illustrating the nested cross validation process, with the outer and inner folds, taken from [10].

Simulations

4.1 Motivation

In this section, simulated data will be used to compare penalised regression methods in different scenarios, focusing on how they differ and address features in the model compared to ordinary least squares. All R code for these simulations can be found in A.

4.2 Multicollinearity

4.2.1 Small Dimensions

This simulation details a model consisting of 2 highly correlated predictors. This simulation is based on an example in [3]. Assuming a correct linear model:

$$y = x_1 + x_2 + \epsilon \tag{4.1}$$

Defining the intercept, $\beta_0 = 0$; $\beta_1 = 1$; $\beta_2 = 1$ and the "noise"/ error term, $\epsilon \sim N(0, 1)$.
Simulating highly correlated variables introduces multicollinearity into the model:

$$x_1 \sim N(0, 1)$$

$$x_2 \sim N(0.95x_1, 0.1^2)$$

Note x_2 is simply a scaling of x_1, with some "noise" added.

Calculated least squares estimated coefficients and ridge estimated coefficients; iterating this process 30 times; and plotting the estimated $\hat{\beta}_1^{ls}, \hat{\beta}_2^{ls}$ in comparison to $\hat{\beta}_1^{ridge}, \hat{\beta}_2^{ridge}$, we see the spread between estimates in Figure 4.1.

In conclusion, the $\hat{\beta}$'s show far more spread for ordinary least squares regression than ridge regression when multicollinearity in present. In this case, the ridge penalty has led to muchmore stability is coefficient estimates, as the introduction in bias has led to a vast decrease in variance, yielding much lower model prediction error.

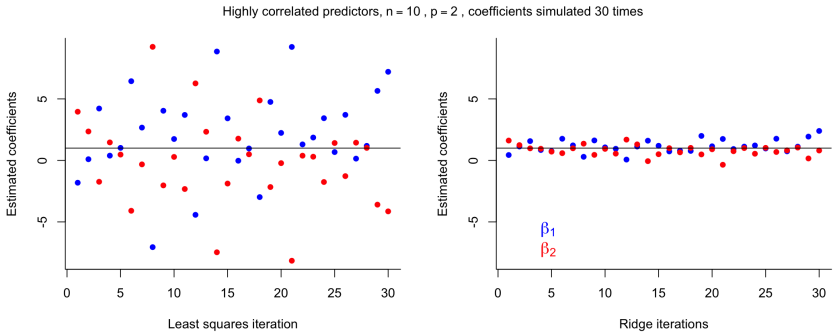

Figure 4.1: Figure showing plotted $\hat{\beta}_1$ (blue) and $\hat{\beta}_2$ (red) for least squares (left) and ridge (right) against each iteration, i, in comparison to the assumed correct coefficients (dotted lines).

4.2.2 Dense Data Generating Models

To somewhat accurately investigate multicollinearity within a high-dimensional dataset, Monte Carlo simulations can be utilised to assess the behaviour of the lasso, ridge and elastic net regularisation terms and resulting predictive abilities of their respective models by the empirical process of actually drawing lots of random samples and observing this behaviour. [13]. In these high-dimensional simulations, the regression models willbe trained on half of the simulated data, and will be analysed by averaging the MSE on the testing set (the other half of the data) across the iterations of Monte Carlo simulation. To fairly compare the regression models on these simulated datasets, the true data generating models used will separate cases of dense and sparse data generating mechanisms.

The Simulated Dataset.

Firstly, specifying the magnitude of collinearity by $\boldsymbol{\rho} = (0.0, 0.3, 0.6, 0.9)$ to yield the variance-covariance matrix, $\sum$, a positive-definite symmetric matrix specifying the covariance matrix of the variables, such that:

$$\sum = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{bmatrix}$$

Then, monte carlo simulations will be performed using the **MASS** package in R [20] to simulate the matrix of predictors:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}$$

with $x_{ij} \sim N(0,1)$, with a total of n observations and p covariates. The response is simulated from the true data generating model:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon \tag{4.2}$$

with $\epsilon \sim N(0, 1^2)$

The models will be trained on a training set of $n = 250$ observation of $p = 100$ predictors. Firstly, setting $\beta_j = 1 for j = 1, ..., p and \beta_0 = 0$, the proposed true data generating model is such that the intercept is 0 and all other coefficients are 1:

$$y = 0 + x_1 + x_2 + \cdots + x_100 + \epsilon \tag{4.3}$$

Penalised regression methods can be analysed on a true dense data generating mechanism for simulated datasets of varied multicollinearity. The predictive performance of different models is observed by training the models on the observations, and calculating the mean squared error after prediction on the testing set of remaining data points, that is:

$$MSE = \sum_{i=1}^{n} (y_i - \hat{y_i}^2) \tag{4.4}$$

for $i = 1, ..., n$

Table 4.1 shows the results yielded by the Monte Carlo simulation, by averaging testing MSE values over 100 iterations. As can be seen, the OLS regression model performs well with zero correlation introduced into the training set, hence OLS assumptions are satisfied and the Gauss-Markov Theorem holds. However, OLS regression performs increasingly more poorly on the test set as the correlation between predictor variables increases, with a significantly greater average test MSE. Contrastingly, LASSO and Ridge penalised models perform better as multicollinearity in the dataset is introduced. This speaks to the respective L_1 and L_2 penalty terms abilities to overcome correlations in the data.

Table 4.1: Test MSE

	$\rho = 0.0$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.9$
OLS	0.80	24.39	48.22	71.36
LASSO	1.74	1.74	1.69	1.96
Ridge	1.69	1.44	1.25	1.07

Table 4.2 shows the optimal λ value for each ρ, that is λ such that the test MSE is minimised, hence showing the amount of shrinkage utilised by each model. Clearly, the ridge model was more comfortable shrinking multiple coefficients. This speaks to ridge being the more appropriate model for true dense data generating models (lots of predictors relevant to the response), whereas lasso's tendency to shrink covariates to 0 forces the tuning parameter to be much smaller in this scenario.

Table 4.2: Optimal λ

	$\rho = 0.0$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.9$
LASSO	0.037	0.010	0.010	0.010
Ridge	0.10	8.17	20.15	20.55

Table 4.3 below shows the $\hat{\beta}_j$ estimates for $j = 0, 1, ..., 6$, where the simulation yields the highest amount of multicollinearity in the dataset, $\rho = 0.9$. Clearly, the ridge regression model has selected coefficients that diverge the least from the true $\beta_j = 1.0$ for $j = 1, \ldots, 6$ for this simulation, and also selects the intercept closest to the true $\beta_0 = 0$. The OLS regression model has obtained values generally with the most deviation from the true data generating model, which is a key indicator of overfitting, helping to explain the high test MSE (prediction error).

Table 4.3: $\hat{\beta}_j$ estimates:

	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$
OLS	0.123	1.05	0.57	1.26	1.04	0.93	1.16
LASSO	0.18	1.27	0.56	1.64	1.46	0.56	1.63
Ridge	0.087	1.03	0.95	1.01	1.04	0.98	1.02

In conclusion, it is clear that for true data generating models of lots of relevant predictors to the response, ridge regression is the most powerful predictive model. As multicollinearity increases in the data, OLS is proven in this simulation to be a poor predictive model in comparison. This simulation study has shown that in examples of a similar nature, the introduction of bias in the $\hat{\beta}$ coefficient estimates could be crucial in order to yield accurate predictions on unseen data.

4.2.3 Sparse Data Generating Models

A sparse data generating model is a model of predictor covariates relevant to the response being significantly less than that of the total number of covariates included in the training dataset. With the feature selection property of LASSO regression, the resulting models are sparse and hence would be epected to perform well in this setting. In this simulation, ridge, LASSO and OLS will be compared as before, but adaptive LASSO and relaxed LASSO are included to investigate these models as well.

For this simulation, the dataset is set up as before, however the response is simulated from a sparse data generating model, that is:

$$\boldsymbol{\beta} = (1, 0, 0, 1, 0, 0, 1, 0, 0, ...) \tag{4.5}$$

proposes the true data generating model with $\beta_j = 1 \forall i = 3k+1, k = 1, ..., \frac{p}{3}$. That is:

$$y = \beta_1 x_1 + \beta_4 x_4 + \beta_7 x_7 + ... + \beta_1 00 x_1 00 + \epsilon \tag{4.6}$$

As LASSO is theoretically the regression model of choice for sparse data generating models, comparison with the adaptive lasso and relaxed lasso variations could be an area of interest for certain datasets. Fitting of the adaptive lasso model requires introduction of the weight function, w_j, in the regularisation term. This requires estimated coefficients of the ridge regression model, $\hat{\beta}_j$ to create the absolute weights vector:

$$w_j = \frac{1}{|\hat{\beta}_j|} \tag{4.7}$$

Implementing this penalty factor into the LASSO regression model yields the adaptive LASSO coefficients:

$$\hat{\beta}_{ada} = \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda \sum_{j=1}^{p} w_j |\beta| \tag{4.8}$$

Imposing the same monte carlo simulated dataset conditions as before and training the regression models on 1 half of the dataset of $n = 500$ observations, the average test MSE over 10 monte carlo simulations on the remaining half of the dataset (the unseen data points) is recorded. Table 4.4 below provides these results.

Table 4.4: Test MSE

	$\rho = 0.0$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.9$
OLS	0.28	2.85	5.43	8.40
LASSO	1.39	1.28	1.29	1.27
Ridge	1.62	1.60	1.58	1.45
Adaptive LASSO	1.24	1.23	1.22	1.29
Relaxed LASSO	1.51	1.32	1.42	1.27

As seen, OLS still fails to adapt its regression model accurately as correlations between independent covariates increases, however performs extremely well given uncorrelated predictors. In addition, the LASSO models becomes the more effective of the penalised regression models, due to the tendency to force coefficients to be exactly equal to 0 in order to yield sparse regression models.

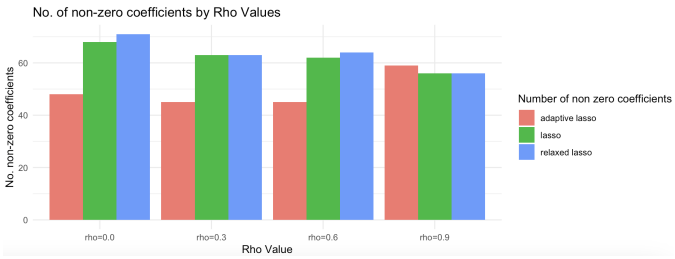

Figure 4.2: Figure showing the number of non-zero coefficients

Plotting the number of non-zero coefficients selected by lasso regression models for their optimal λ, as seen in Figure 4.2, illustrates each version of he LASSO's variable selection properties:

Noting that the true number of non-zero coefficients in this model is 34, Figure 4.2 shows that generally the adaptive lasso appears to be closest to this number. The oracle property of the adaptive lasso states that as $n \to \infty$, the number of non-zero $\hat{\beta}$ converges to the true number. Scaling the number of training data points, N, from 100 to 100, 000 in 1 iteration of the simulation, Figure 4.3 plots the number of non-zero coefficients selected by the adaptive lasso. The graph would suggest this asymptotic oracle property, perhaps verifying the oracle property proposal [22].

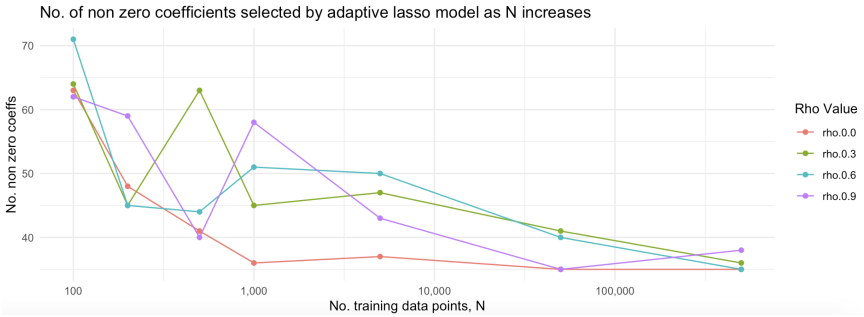

Figure 4.3: Trace plot of the number of non zero coefficients selected by the adaptive lasso as n increases.

R Code for this simulation is found in A.

4.3 p close to n

4.3.1 n=3, p=2

This simulation is based on a similar study in [3], and aims to understand the robustness of OLS and Ridge regression in the case of few data observations. Focusing on a model of 2 uncorrelated predictors, with $n = 3$ predictors, x_1

and x_2 are simulated:

$$x_1 \sim N(0,1)$$
$$x_2 \sim N(0,1)$$

The response, y is simulated from a true data generating model of

$$y = x_1 + x_2 + \epsilon \tag{4.9}$$

with $\epsilon \sim N(0,1)$. Hence, the true coefficients in this simulation are $\beta_0 = 0$, $\beta_1 = 1$, $\beta_2 = 1$.

Upon iterating estimated coefficients, $\hat{\beta}$, 30 times for both least squares and ridge regression, and plotting the spread of these coefficients in Figure 4.4, it is clear the ridge regression model is more robust in this situation, and yields more consistent estimates, closer to the true β_j values.

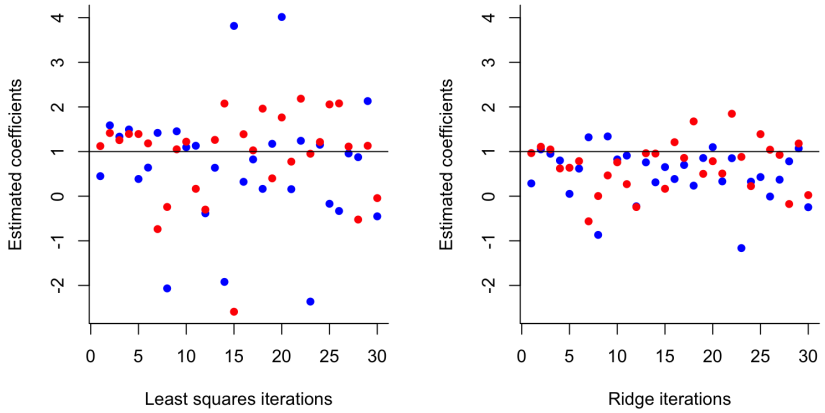

Figure 4.4: Figure showing plotted least squares coefficients (left panel) and ridge coefficients (right panel) for $\hat{\beta}_1$ (blue) and $\hat{\beta}_2$ (red) against each iteration, i. The black horizontal line in each plot is the true coefficient value.

Conclusion: As seen in Figure 4.4, for each iteration, i, ridge estimated coefficients are far more consistent than those for least squares, proving that the introduction of some bias has again decreased variance for this model of few observation in comparison to number of predictor variables.

4.3.2 p close to n in higher dimensions

Here, properties of ridge regression are generalised in higher dimensions, where the number of predictor variables is still close to the number of observations.

For this simulation, $n = 100$ and $p = 99$ are used. Like before:

$$x_i \sim N(0, 1), i = 1, \ldots, p \tag{4.10}$$

observed 100 times, with

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_9 9 x_9 9 + \epsilon \tag{4.11}$$

the proposed model for the linear regression, presenting y as a linear combination of the predictor variables.

The results from the simulation are plotted in Figure 4.5, which provides similar evidence that ridge is far more robust in this example, and the regression coefficients are again more consistent and close to the true value.

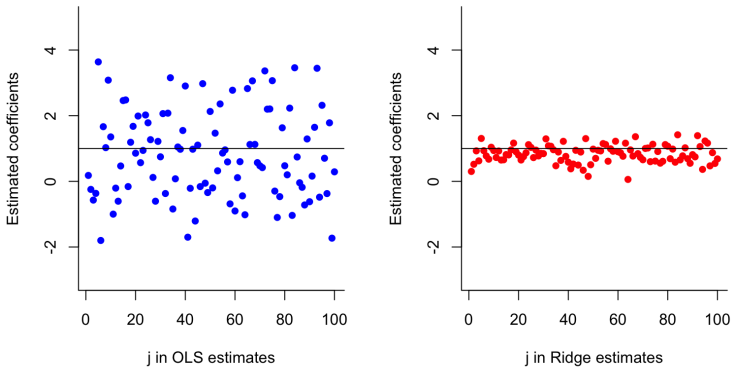

Figure 4.5: Figure showing coefficient estimates for a least squares (left panel) and ridge (right panel) regression model across a n=100, p=99 simulated dataset. The black horizontal line represents the true $\hat{\beta}_j$ value.

In Figure 4.5 suggests that the variance (spread) of ridge coefficients is far smaller than that of the least squares estimates. This means that the ridge coefficients are far more consistent for this p=99, n=100 simulation, so it would be assumed that the model is more accurate, and will perform better on unseen data. R Code for this simulation is found in A.

4.4 Ridge, The Lasso, Elastic Net

In this section, results are explored of applying ridge, the lasso and elastic net regression models to a simulated dataset, *seatpos* in the R package *faraway* [5]. The purpose of this simulation is to investigate the effect of the

tuning parameter, $\lambda > 0$, on the training MSE and coefficent shrinkage of each penalised regression model, and not to perform any predictive application.

The simulated dataset of n=38 observations and p=9 independent variables (8 predictors) is based on modelling the driving seat position of an individual against features such as foot size, leg length, height etc. The first few observation of this dataset is included in the table in Appendix A.
Intuitively, this dataset includes multicollinearity. We can verify this using a pairs plot to analyse the possible correlations within the dataset. As seen in Figure 4.6, features such as *Age* and *Height* are fairly uncorrelated, whereas features such as *Height* and *HtShoes* present high correlation.

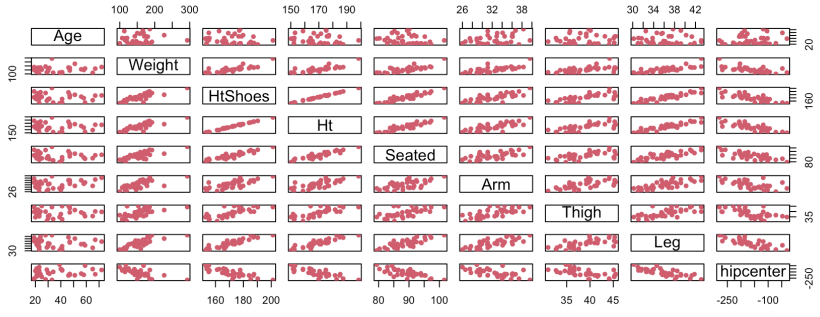

Figure 4.6: Figure showing a pairs plot of the simulated dataset, used to inspect possible correlated relationships between variables.

Upon fitting ridge, lasso and elastic net regression methods are fit to the dataset. It is discovered that for elastic net regression, the hyperparameter α returns the smallest MSE value at $\alpha = 0.01$. Comparison of mean squared error results, as well as the magnitude of regression coefficients, establishes how these models differ. Figure 4.7 and figure 4.8 show these differences. In Figure 4.7, the left-hand vertical dotted line of each panel represents the value of λ for which the MSE is minimised, whereas the right-hand vertical dotted line of each panel represents the value of λ that returns the simplest model given its MSE is within 1 standard error of the minimum MSE. R Code for this simulation is found in A.

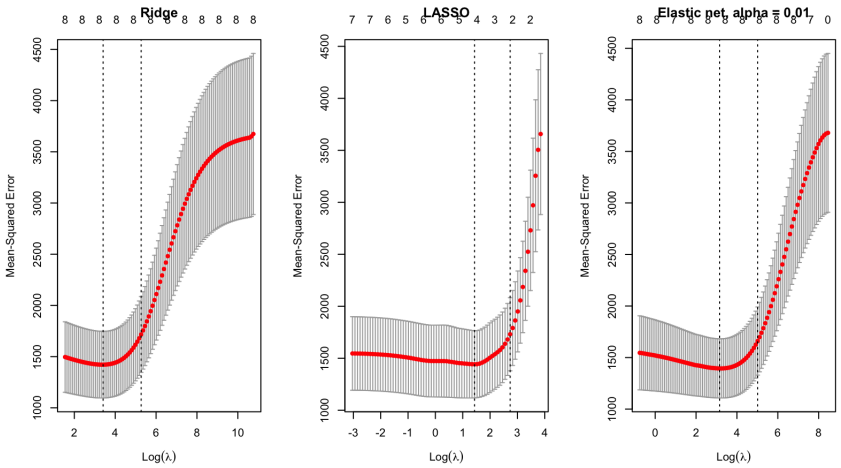

Figure 4.7: Figure showing the training MSE of each of Ridge (left panel), LASSO (centre panel) and Elastic Net (right panel)

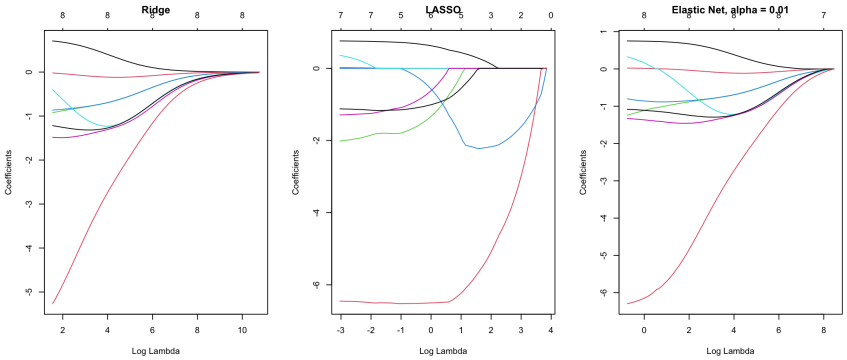

Figure 4.8: Figure showing the header of a selected subsection of a credit scoring dataset, used in our investigation of relevant predictor variables.

Applications: Penalised Regression in Finance

5.1 Penalised Logistic Regression for Loan Status Classification

Logistic regression, which is useful for predicting the occurrence or non occurrence of a quality or outcome based on values of a set of forecaster variables, is a multivariate analysis model [9]. In the area of banking, corporate finance and investments, logistic regression applications have frequently been used, specifically for the default-prediction model.
Accurate prediction of creditworthiness is crucial for financial institutions to make informed decisions about issuing loans and lines of credit. Traditional logistic regression often faces challenges like multicollinearity and model overfitting in credit scoring datasets. This section explores the application of penalised regression methods—ridge and lasso logistic regression to credit score classification. This investigation includes a case study on a real-world dataset, analysing whether penalised logistic regression models could be a more appropriate approach than that of standard logistic regression.

5.1.1 The Dataset

For this investigation, a loan status classification dataset is used (Kaggle.com: A) to analyse how statistically viable penalised logistic regression methods can be in comparison to standard logisitic regression, and whether or not a financial institution in this field might choose to employ these techniques. The dataset includes 28638 observations of 10 predictor variables, with 1 binary response variable **loan status**. The first few observations can be seen in Table 5.1.

Table 5.1: Loan Status Classification Dataset

age	income	home ownership	loan length	loan intent
22	59000	rent	123	personal
21	9600	own	5	education
25	9600	mortgage	1	medical

loan grade	loan amount	loan status	default before?	cred hist len
D	35000	1	Y	3
B	1000	0	N	2
C	5500	1	N	3

5.1.2 The Model

The response variable **loan status** is a binary response variable, taking values 0 or 1, for a "good" or "bad" loan respectively. A loan is said to be bad if a customer is in default, which means that person is unable to service their financial debt/ obligation [1]. The response, denoted Y, follows the Bernoulli distribution (Binomial distribution with $n = 1$) with unknown probability, p, of success:

$$Y = \begin{cases} 1 & \text{success (bad loan - customer in default), with probability } p, \\ 0 & \text{failure (good loan), with probability } 1 - p. \end{cases}$$

Like seen in section 3.4, the logistic regression model works to calculate the probability of class membership, so in this example provides the probability that a loan is in default (bad):

$$p(Y = 1) = \frac{1}{1 + e^{-y}}$$
$$= \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ...}}$$

This is achieved by maximising the likelihood function:

$$\mathbb{L}(\beta) = \prod_{i=1}^{N} p_i^{y_i} (1 - p_i)^{1 - y_i} \tag{5.1}$$

The logistic regression model classifies a response from an unseen set of data by calculating the probability of success:

$$P(Y = 1) > 0.5 \implies Y = 1 \tag{5.2}$$

To assess the assumed effectiveness of including the L_1 and L_2 penalties, multicollinearity within the dataset can be investigated. High correlation between lots of variables within the dataset should lead to inflated coefficients and overfitting for the standard logistic regression model, whilst the penalised logistic regression models should retain a higher predictive ability, as seen in simulations in Section (reference section).

To test for multicollinearity within the dataset, variance inflation factors (VIFs) can be calculated, as specified in Section 2.2.2. Table 5.2 shows the highest 4 VIFs calculated for this dataset, for numerical variables.

Table 5.2: Variance Inflation Factors

Predictor	income	loan_length	loan_amnt	cred_hist_length
VIF	4.4	2.1	2.1	4.2

Noting that all $VIF < 5$ for numerical values, and a dataset containing lots of observations for only 11 explanatory variables, it could be concluded that the bias-variance tradeoff imposed by penalised logistic regression penalties may not benefit this particular financial model. However, penalised logsitic regression model showing powerful predictive abilities of at least similar to that of the standard logistic regression would suggest a powerful model in the case of a much less simple, high-dimensional dataset.

To test the hypothesis that penalised logistic regression is a powerful tool for credit-deafult classification applications, ridge, lasso and standard logistic regression models is fit to a random 70% of the data to train the models. The model's predictive accuracy is tested on the remaining 30% of the data (the testing set).

5.1.3 Results

To analyse the predictive abilities of the model, the classifications of unseen data of each model are compared against the true classification of the response in the testing set. For each model, a confusion matrix is produced that outputs the counts of each models predicted classifications on the test set against the true value, as can be seen in the Tables 5.3 below.

5.3.1:Standard

	true	
pred	0	1
0	6655	817
1	275	845

5.3.2:LASSO

	true	
pred	0	1
0	6657	784
1	273	878

5.3.3:Ridge

	true	
pred	0	1
0	6707	878
1	223	784

Calculating the accuracy (proportion of correct classifications), sensitivity (proportion of correct "bad loan"/ customer in default classifications) and specificity (proportion of correct "good loan" classifications) [11] provides the necessary data to conclude whether the penalised logistic regression models may be more desirable in the credit classification field than that of standard logistic regression. These values can be found in Table 5.4, all rounded to 3 significant figures.

Table 5.4: Accuracy, Sensitivity and Specificity

	Accuracy	Sensitivity	Specificity
Standard	0.873	0.508	0.960
LASSO	0.877	0.528	0.961
Ridge	0.872	0.472	0.968

Though the ridge penalty yields the most desirable specificity, meaning the model is most likely to correctly classify a loan which will not end up defaulting, the lasso logistic regression model has the highest accuracy and sensitivity. Given this application is regressing the response on just 11 predictor variables, with not a significantly high amount of multicollinearity within the dataset, one can conclude that for a financial institution, the lasso logistic regression model is a more appropriate model to fit than the standard logistic regression model.

The penalised logistic model should, in theory confirmed by simulation in section 2.2.2, continue to outperform the standard logistic regression model further as an increased number of explanatory variables are introduced, perhaps with an inherit correlation between other features as well. In fact, one would assume that banks have extremely large datasets for loan application classification.

Inspection of the estimated coefficients of the most accurate model signal which variables are most likely to have the greatest impact on the determination of loan status, which can not only be useful to financial institutions, but also to prospective customers who may wish to judge their chances before applying for a loan. For example, a customer who was previously defaulted on a loan, may wish to understand if this is likely to impact their ability to receive a loan even if they now own a home outright. Inspection of the relevant coefficients in the lasso model provides this information:

$$\hat{y} = 0.0324(cb\_person\_default\_on\_fileY) - 1.18(person\_home\_ownershipOWN) + 0.09165(person\_home\_ownershipRENT)$$

Interpretation of these coefficients show that home ownership ($\hat{\beta} = -1.18$) decreases the chances of a classification of 1 (customer defaulting) more than a previous default record ($\hat{\beta} = 0.0324$) increases the chances of a "bad loan" classification. This would suggest that a customer who has previously defaulted on a loan, but now bought a home, could reapply for a loan with a bank using the same logistic regression model.

In conclusion, for a dataset of this nature, penalised logistic regression is a very powerful tool. The ability of the L_1 and L_2 penalties to limit model overfitting and mitigate against "noise" in high-dimensional datasets could provide financial institutions, depending on their standards for accuracy, sensitivity and specificity, with accurate enough models to ensure profitability as datasets continue to increase in dimensionality. R Code for this study is found in A.

5.2 Stock Price Prediction

Stock price prediction, commonly referred to as stock price forecasting, involves estimation of a future stock price traded on an exchange. Accurate stock price forecasting can provide highly desirable opportunities for profit, and provides a base for companies or individuals looking for the potential to make a "bet" on the market, which may involve buying or a stock that may be predicted to trend up or down respectively.

Correlation between stock prices is extremely potent in financial markets. Though news events, economic recessions, potential conflict and many more factors contribute to continuous random fluctuations in stock prices, competing companies and relevant index funds often see extremely similar trends in stock price. This study will investigate the accuracy of penalised regression models in counteracting overfitting that is commonly seen when ordinary least squares regression is applied to stocks exhibiting high correlations.

5.2.1 Stock Market Data

The data these models are trained on is compiled from the start of 2021 until the start of 2022, where the closing price of Google stock will be modelled as the response variable to predict, with proposed explanatory features including opening price, highest price, lowest price and volume of stock traded for each day of Apple stock and Nasdaq 100 index fund. Apple is included in the investigation to analyse the impact of one of Google's largest competitors [14], and the Nasdaq 100 is included as it is an index fund of 100 of the United States' most profitable big tech companies, and it is said to have influence over tech company stock [16]. The first 3 days of the training data, of which there are 14 variables spanning 250 days total, can be viewed in Table 5.5.

Table 5.5: Stock Market Data for Stock Price Prediction

Date	Google Close	Google Open	Google High	Google Low
2020-01-02	68.3685	67.0775	68.407	67.0775
2020-01-03	68.0330	67.3930	68.625	67.2772
2020-01-06	69.7105	67.5000	69.825	67.5000

Goog Volume	Appl Open	Appl High	Appl Low	Appl Volume
28132000	74.0600	75.150	73.7975	135480400
23728000	74.2875	75.145	74.1250	146322800
34646000	73.4475	74.990	73.1875	118387200

NSDQ Open	NSDQ High	NSDQ Low	NSDQ Volume
214.4	216.16	213.98	30969400
213.3	215.47	213.28	27518900
212.5	215.59	212.24	21655300

Investigation of multicollinearity, undertaken by the plotting of the Pearson correlation matrices as seen in Figure 5.1 reveals a potential linear regression modelling flaw, pointing towards inflated coefficients and overfitting.

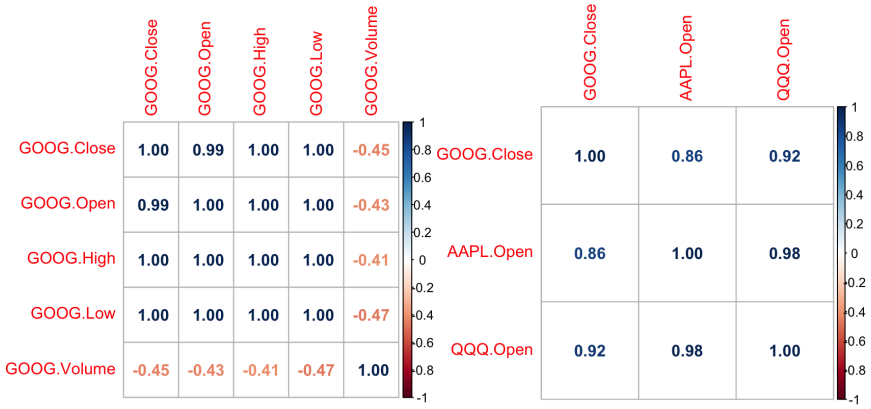

Figure 5.1: Figure showing Pearson correlation matrices within the stock price prediction data.

As can be seen in Figure 5.1, the Pearson correlation coefficients calculated between Google stock price indicators ≈ 1(opening price, closing price, highest price, lowest price). There is a similar magnitude of correlation between the prices of the 3 different stocks, and given Pearson correlation coefficient, $r = 1 \implies$ perfectly positive correlation, multicollinearity can be deduced in the dataset. As seen in Section 2.2.2, this indicates that the introduction of a penalty term may be an appropriate regression method to employ for stock price forecasting.

For this application, ridge, lasso and elastic net regression models are all fit to the data, to explore the different results yielded between the 3 different models and deduce the most appropriate for forecasting in stock price datasets which exhibit similar relationships between exploratory predictors.

5.2.2 Results

For the elastic net regression model, the optimum mixing parameter, α^*, was found to be $\alpha^* = 0.13$ for this study. To find the optimum α^*, which controls the balance between l_1 and l_2 penalties within the elastic net regression model and hence the bias-variance tradeoff, the model was fit iteratively for α increasing from 0 to 1 in 0.01 increments. The optimum α was recorded for which the mean squared error was at its minimum when calculated on the training set:

$$\alpha^* = \arg\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{X_i}\hat{\boldsymbol{\beta}}) \tag{5.3}$$

Where in this case for elastic net regression,

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||_2^2 + \alpha\lambda||\beta||_1 + \frac{1-\alpha}{2}\lambda||\beta||_2^2 \tag{5.4}$$

Table 5.6 below provides mean squared error results obtained after fitting the penalised regression models on the training set (2020-2021), along with the optimum λ, which in this case was taken to be, similarly to in the selection of α, the tuning parameter that also minimised the mean squared error.

Table 5.6: Training MSE and λ Minimising MSE

Model	Optimum λ	Training MSE
Ridge	0.8749442	0.8971229
LASSO	0.05245153	0.4398327
Elastic Net	0.00673034	0.316318

To draw conclusions on the effectiveness of each model for this stock price forecasting application, unseen data, namely the price of Google stock on daily close for the year of 2021 and the year of 2022, is estimated after the relevant predictors are fit to each of the 3 sets of $\hat{\beta}$ estimated coefficients, and the MSE is calculated on each year, acting as 2 testing sets. Both years are tested, as the trading year of 2021 showed similar trends to the 2020 training set (both fairly steady up-trends), whilst 2022 was generally a more volatile year for tech stocks (unlike the testing set). Table 5.7 below provides the stock price forecast mean squared error for each of the trading years 2021 and 2022.

Table 5.7: Test MSE for Forecasted Stock Price, 2021 and 2022

Model	2021 Forecasting MSE	2022 Forecasting MSE
Ridge	42.75756	33.53733
LASSO	1.473568	2.169171
Elastic Net	0.7492672	1.103368

Interpreting these results, the elastic net and lasso regression models can both be seen to be fairly accurate in this setting, and the forecasted MSEs being close to the training MSE would suggest both models are well enough defined to perform well on the testing set as well as the training set. However, the results obtained from the ridge model juxtapose this, as the performance on the testing sets is significantly worse than on the training set. This is symptomatic of overfitting and would suggest that the noise caused by extremely highly correlated predictor variables has diverted $\hat{\beta}_{ridge}$ coefficients too far from the true model. One possible cause of this could be Ridge regression inability to perform feature selection, and would suggest the true model, $y = X\beta$ does not include all p predictors, but a smaller subset of them. Figures 5.2 and 5.3 plot the forecasted closing price of Google stock obtained by each of the models within the same time series as the true values.

Figure 5.2: 2021 time series data for predicted Google stock closing price against the actual price (blue).

Figure 5.3: 2022 time series data for predicted Google stock closing price against the actual price (blue).

In conclusion, the elastic net and lasso models clearly perform well under highly correlated stock market data. The ridge model's inability to force coefficients to 0 leaves it vulnerable to inflated coefficients due to noise between highly correlated variables. The elastic net, performing best, perhaps does

this through its ability to mitigate from the grouping effect, in which lasso perhaps removes relevant variables from the model due to high correlation with another variable, instead of assigning both a similar coefficient [23]. R Code for this study is found in A.

Discussion

This report looked to explore penalised regression methods as alternative linear regression models to the standard ordinary least squares, as well as touching on areas of study such as Bayesian Shrinkage and Logistic Regression, whereby penalised methods could exhibit similar benefits. This has been achieved by exploring the theory behind regularisation terms, and then investigating the impact on simulated datasets and real-world datasets exhibiting features that do not adhere to data assumptions required by other regression methods.

Chapter 2 examined ordinary least squares regression, and explored some of the drawbacks under certain conditions. This led to the motivation behind penalised regression methods and the issues with OLS regression that data scientists may wish to negate, and introduction of L_2 (ridge) and L_1 (lasso) penalty terms provides a method that allows for balancing of the bias-variance trade off.

For ridge and lasso regression, it is clear the circumstances in which each method is more desirable. When a true data generating model is dense, and lots of predictors are relevant to the response, ridge regression exhibits powerful robustness, consistency and predictive accuracy. Conversely, when a true data generating model is sparse, and a smaller subset of predictors are relevant to the response, the feature selection property of the LASSO allows it to counteract overfitting and yield a simpler model.

For logistic regression, explored in the loan status classification example of Section 5.1, penalised methods have benefits over standard logistic regression, which similarly to OLS can also yield inaccurate results in high-dimensional datasets with high levels of correlation. Similarly, in stock price prediction, the extremely high correlation between variables allows elastic net regression, with the α parameter to balance the L_1 and L_2 penalties, to not only shrink variables to yield a simpler linear regression model, but to also counteract the grouping effect.

Further investigation into regularisation, in a similar manner to this re-

port, could involve research into neural networks with L_1 and L_2 penalties, which have applications in deep learning and prediction.

Bibliography

[1] Peter Crosbie and Jeffrey Bohn. Modeling default risk. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Volume 2: Corporate Debt Valuation with CCA*, pages 471–506. World Scientific, 2019.

[2] Jamal I Daoud. Multicollinearity and regression analysis. In *Journal of Physics: Conference Series*, volume 949, page 012009. IOP Publishing, 2017.

[3] Hailang Du. Machine learning and neural networks, 2023.

[4] Fan and Li. *Variable Selection via Non concave Penalized Likelihood and Its Oracle Properties.* Journal of the American Statistical Association, 96, 1348–1360, 2001.

[5] Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models.* CRC press, 2016.

[6] Arthur E. Hoerl and Robert W. Kennard. *Ridge Regression: Biased Estimation for Nonorthogonal Problems.* University of Delaware and E. 1. du Pont de Nemours Co., 1970.

[7] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. *An introduction to statistical learning*, volume 112. Springer, 2013.

[8] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

[9] Saro Lee. Application of likelihood ratio and logistic regression models to landslide susceptibility mapping using gis. *Environmental Management*, 34:223–232, 2004.

[10] Goldmann K Pitzalis C McKeigue P Barnes MR Lewis MJ, Spiliopoulou A. nestedcv: an r package for fast implementation of nested cross-validation with embedded feature selection designed for transcriptomics and high dimensional data, 2023.

[11] Laurence S Magder and James P Hughes. Logistic regression when the outcome is measured with uncertainty. *American journal of epidemiology*, 146(2):195–203, 1997.

[12] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[13] Christopher Z Mooney. *Monte carlo simulation*. Number 116. Sage, 1997.

[14] Martin Moore and Damian Tambini. *Digital dominance: the power of Google, Amazon, Facebook, and Apple*. Oxford University Press, 2018.

[15] Robert M O'brien. A caution regarding rules of thumb for variance inflation factors. *Quality & quantity*, 41:673–690, 2007.

[16] Gary C Sanger and John J McConnell. Stock exchange listings, firm value, and security market efficiency: The impact of nasdaq. *Journal of Financial and Quantitative Analysis*, 21(1):1–25, 1986.

[17] Leonard J Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, 16(4):437–450, 2000.

[18] Robert Tibshirani. *Regression Shrinkage and Selection via the Lasso*. Journal of the Royal Statistical Society. Series B, 1996.

[19] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7:1–8, 2006.

[20] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[21] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

[22] Hui Zou. *The Adaptive Lasso and Its Oracle Properties*. Journal of the American Statistical Association, 2006.

[23] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

Appendix A: R Code and Data

All R code used in this can be found at:
https://github.com/joshkaura/Penalised-Regression-Methods/tree/main

Seatpos data in *faraway*: Table: Seatpos Dataset - Faraway Package in R [5]

Age	Weight	HtShoes	Ht	Seated	Arm	Thigh	Leg	hipcenter
46	180	187.2	184.9	95.2	36.1	45.3	41.3	-206.3
31	175	167.5	165.5	83.8	32.9	36.5	35.9	-178.
23	100	153.6	152.2	82.9	26.0	36.6	31.0	-71.673

Loan Status classification dataset: https://www.kaggle.com/datasets/laotse/credit-risk-dataset