

Department of Mathematical Sciences Project IV - (MATH4072)

Equity Fund Profitability and Sustainability Modelling

*Using Multiple Response Regression*

**Author**: Raul Unnithan

**Supervisor**: Ric Crossman

April 24, 2025

#### Acknowledgement

Write an acknowledgement to Ric...

#### Declaration

This piece of work is a result of my own work and I have complied with the Department’s guidance on multiple submission and on the use of AI tools. Material from the work of others not involved in the project has been acknowledged, quotations and paraphrases suitably indicated, and all uses of AI tools have been declared.

Contents WHY THE RED BOXES?

**.** **3**

**[1 Introduction](#_bookmark0)**

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[1.1 Background](#_bookmark1)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[1.1.1 Multiple Response Regression](#_bookmark2)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[1.1.2 Equity Funds](#_bookmark3)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[1.2 Equity Fund Dataset](#_bookmark4)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[1.2.1 Overview](#_bookmark5)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[1.2.2 Responses Modelled](#_bookmark6)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

[1.2.3 Motivation](#_bookmark7)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

[1.3 Small-Scale Dataset](#_bookmark8)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

[1.4 Report Notation](#_bookmark9)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

[1.5 Report Outline](#_bookmark10)

#### [2 Exploratory Data Analysis](#_bookmark11) 6

. . . . . . . . . . . . . . . . . . . . . . . . . 6

[2.1 Initial Data Collection and Preprocessing](#_bookmark12)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

[2.2 Variable Explanation](#_bookmark13)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

[2.3 Exploratory Data Analysis](#_bookmark14)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

[2.3.1 Underlying Distributions](#_bookmark15)

. . . . . . . . . . . . . . . . . . . . . . . . . 8

[2.3.2 Correlation and Multi-Collinearity](#_bookmark17)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

[2.3.3 Imputing Missing Values](#_bookmark18)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

[2.3.4 Multicollinearity](#_bookmark19)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

[2.3.5 Multivariate Normality](#_bookmark20)

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

[2.4 Response Intercorrelation](#_bookmark21)

**.** **11**

[**3 Linear Regression**](#_bookmark23)

. 11

[3.1 Theory](#_bookmark24)

. 11

[3.1.1 Single-Response Linear Regression](#_bookmark25)

. 12

[3.1.2 Multiple Response Linear Regression](#_bookmark27)

. 14

[3.1.3 Stepwise Selection Methods](#_bookmark31)

. 15

[3.1.4 Analysis of Variance](#_bookmark32)

. 15

[3.1.5 Multiple Analysis of Variance](#_bookmark33)

. 17

[3.2 Application](#_bookmark34)

. 17

[3.2.1 Model Evaluation and Performance](#_bookmark35)

. 17

[3.2.2 Standard Multiple Linear Regression](#_bookmark38)

. 18

[3.2.3 Code Explanation](#_bookmark39)

. 18

[3.2.4 Small-Scale Example](#_bookmark40)

. 20

[3.2.5 Applying Stepwise Selection](#_bookmark41)

. 21

[3.2.6 Extending and Evaluating Stepwise Selection](#_bookmark42)

**.** **22**

[**4 Shrinkage Methods**](#_bookmark44)

. 22

[4.1 Theory](#_bookmark45)

. 22

[4.1.1 Ridge Regression](#_bookmark46)

. 23

[4.1.2 Lasso Regression](#_bookmark50)

. 24

[4.1.3 Shrinkage Methods for MRR](#_bookmark51)

. 24

[4.1.4 Reduced Rank Ridge Regression](#_bookmark53)

. 26

[4.1.5 Multivariate Regression with Covariance Estimation](#_bookmark56)

. 29

[4.2 Application](#_bookmark61)

. 29

[4.2.1 Small-Scale Example](#_bookmark62)

. 32

[4.2.2 Code Explanation](#_bookmark64)

. 33

[4.2.3 Results](#_bookmark65)

#### [5 Random Forests](#_bookmark66) 35

. 35

[5.1 Theory](#_bookmark67)

. 35

[5.1.1 Classification and Regression Trees](#_bookmark68)

. 36

[5.1.2 Bagging for CARTs](#_bookmark69)

. 36

[5.1.3 Random Forests](#_bookmark70)

. 37

[5.1.4 Multivariate Regression Trees](#_bookmark71)

. 38

[5.1.5 Multivariate Random Forests](#_bookmark73)

. 38

[5.1.6 Covariance Regression with Random Forests](#_bookmark74)

. 40

[5.2 Application](#_bookmark76)

. 40

[5.2.1 Small-Scale Example](#_bookmark77)

. 42

[5.2.2 Code Explanation](#_bookmark78)

. 42

[5.2.3 Results](#_bookmark79)

**.** **44**

[**6 XG Boost**](#_bookmark80)

. 44

[6.1 Theory](#_bookmark81)

. 44

[6.1.1 Introduction](#_bookmark82)

. 46

[6.1.2 Iterative Steps of XGBoost](#_bookmark87)

. 47

[6.1.3 Cholesky Decomposition](#_bookmark88)

. 48

[6.2 Application](#_bookmark89)

. 48

[6.2.1 Small-Scale Example](#_bookmark90)

. 52

[6.2.2 Code Explanation](#_bookmark94)

. 52

[6.2.3 Results](#_bookmark95)

**.** **54**

[**7 Conclusion**](#_bookmark96)

. 54

[7.1 Summary of Findings and Model Comparisons](#_bookmark97)

. 54

[7.1.1 Summary of Findings](#_bookmark98)

. 55

[7.1.2 Model Comparisons](#_bookmark99)

. 55

[7.2 Challenges and Limitations](#_bookmark100)

. 56

[7.3 Report Overview and Future Work](#_bookmark101)

**.** **61**

[**A Appendices**](#_bookmark155)

. 61

[A.1 Additional Data Visualisations](#_bookmark156)

. 62

[A.2 Extra Chapter Insights](#_bookmark159)

. 62

[A.2.1 Introduction](#_bookmark160)

. 63

[A.2.2 Exploratory Data Analysis](#_bookmark162)

. 63

[A.2.3 Multiple Response Linear Regression](#_bookmark164)

. 64

[A.2.4 Ridge Regression](#_bookmark166)

. 64

[A.2.5 Reduced Rank Ridge Regression](#_bookmark168)

. 67

[A.2.6 Multivariate Regression with Covariance Estimation](#_bookmark173)

. 68

[A.2.7 Cholesky-Gaussian XGBoost](#_bookmark175)

# Chapter 1 Introduction

## Background

* + 1. **Multiple Response Regression MAKE SURE YOU’RE CONSISTENT WITH USING THE DASH OR NO DASH (I’D SAY NO DASH BUT UNSURE) FOR ‘MULTIPLE RESPONSE/SINGLE RESPONSE’**

Single-response regression models one response against a set of predictor variables. This method can be extended to multiple predictors, but what if we need to model more than one response? One approach is to run several independent single-response models. However, this approach can lead to suboptimal predictions as it ignores the fact that when responses are correlated, the off-diagonal elements of the response-covariance matrix are non-zero. Multiple-response regression (MRR) addresses this issue, thereby improving predictions.

However, selecting the most relevant predictors remains a challenge, and each MRR model does this differently. This report has examined and applied different MRR models to an equity fund dataset to determine which of these fits it the best.

## Equity Funds

An equity fund is a pooled investment that puts money predominantly in stocks listed on major exchanges.[1](#_bookmark102) They provide investors with several benefits, such as diversification and potential improved returns, [1](#_bookmark102) but also come with risks associated with stock market volatility and losses. [1](#_bookmark102) Equity funds can be categorised according to factors such as their investment style and portfolio focus, which is how they are represented in this data. [1](#_bookmark102)

## Equity Fund Dataset

* + 1. **Overview**

The equity fund dataset analysed in this report contains information about a diverse range of funds, with varying equity fund characteristics,GET RID OF COMMA including, for example, the type of equity. Modelling equity fund data is important as it can indicate which funds are worth investing in and which are not. Traditional financial models prioritise profitability above all else.[2](#_bookmark103) However, in this report,GET RID OF COMMA there are two responses modelled – which are moderately correlated. These responses are return on equity (ROE) and sustainability score. A significant? response correlation is important here because, (KEEP COMMA) otherwise,GET RID OF COMMA one model for each response could be built.

## Responses Modelled

ROE is a metric of an equity fund’s profitability and how efficiently it generates those profits. [3](#_bookmark104) It is calculated by dividing net income by shareholders’ equity. [3](#_bookmark104) ROE is typically expressed as a percentage, so this value is multiplied by a hundred. In the context of an equity fund, net income is not derived from sales minus the cost of goods sold as it would be for a company. [4](#_bookmark105) Instead, an equity fund’s net income comes primarily from earnings such as interest, after subtracting operating expenses like

distribution costs. [4](#_bookmark105) The higher the ROE, the more efficiently an equity fund is generating income and growth from its portfolio of equity investments. [5](#_bookmark106)

The sustainability score measures an equity fund’s sustainability performance and how effectively it mitigates environmental, social, and governance (ESG) risks across its portfolio. Each equity fund’s sustainability score is its Morningstar Sustainability Rating, which directly measures these ESG factors. [6](#_bookmark107) The Morningstar Sustainability Rating is calculated through a multi-step process. First, a fund only qualifies for a rating if at least 67% of its assets have an assigned ESG score. [6](#_bookmark107) Each company within the fund’s portfolio is then scored, typically on a scale ranging from 1? to 100. [6](#_bookmark107) The fund’s portfolio score is adjusted by subtracting points for any ESG-related negative events, such as environmental incidents. [6](#_bookmark107) Finally, the equity fund’s overall sustainability score is then compiled as weighted averages over the past 12 months, placing more emphasis on recent performance. Lower sustainability scores indicate better ESG performance of an equity fund.

## Motivation

Modelling specifically ROE and sustainability scores together is important as it gives a more long-term view of each equity fund’s risk versus return potential. Sustainability has become more relevant as investors move to prioritise sustainable portfolios. A key driver of this trend is the growing concern over climate change and global warming, which have significant long-term social and economic implications. In a 2021 letter to other CEOs, Larry Fink, the CEO of BlackRock, wrote that 81% of a globally representative selection of BlackRock’s sustainable indexes outperformed their parent benchmarks. [7](#_bookmark108) This finding emphasises the importance of considering sustainability metrics alongside profitability measures when evaluating financial assets, including equity funds.

## Small-Scale Dataset

Every MRR model in this report generated predictions for the two responses. These predictions were evaluated using the average normalised root mean square error, which is the root mean square error

divided by the standard deviation averaged across each response (see subsection for more).

[3.2.1](#_bookmark35)

Each MRR model was also applied to a small-scale dataset with Exam Scores to demonstrate explicitly how these models make predictions:

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*1: Math Scores | *Y*2: Science Scores |
| 5 | 2 | 78 | 80 |
| 7 | 3 | 85 | 79 |
| 8 | 4 | 88 | 88 |
| 3 | 1 | 65 | 70 |
| 10 | 5 | 92 | 74 |

Table 1.1: Study Time vs. Exam Scores

The calculations involving this dataset used precise values throughout but,COMMA OTHER SIDE of ‘but’ where written, the intermediary and final results were rounded to 2 decimal places for brevity – ADD DASH unless explicitly stated otherwise. The Exam Scores dataset was used for this purpose because its values are straightforward, and the

[A.2.1](#_bookmark161)

responses have a moderate correlation of 0.526 (3dp), indicating that it requires WORD IS BUGGING A WEE BIT IT SEEMS MRR (see the full calculation of this).

for

## Report Notation

In this report, certain common terminology was used. First, the identity matrix was denoted throughout as **I** with dimensions subscripted where necessary. Second, there was the correlation and covariance between the responses, NO COMMAwhich, as stated, MRR considers while single-response regression does not. This consideration is essential for every MRR model as it is what qualifies them as multiple-response.

The correlation between the responses, referred to as the response intercorrelation, was defined as a matrix, **R**, using the formula: **R** = **D***−*1**ΣYD***−*1*,* where Cov(**Y**) = **ΣY** denotes the covariance matrix of the response variables, or the response covariance matrix, and **D** is a diagonal matrix of the standard deviations of the response variables. [8](#_bookmark109) Each element of **R** is computed as:

*Rij*

= (**ΣY**)*ij ,*

*σiσj*

…where (**ΣY**)*ij* is the covariance between response variables *i* and *j*, and *σi, σj* are their respective standard deviations. [8](#_bookmark109) This relation ultimately means that if an MRR model considers the off-diagonal elements in the response covariance matrix, it considers the response intercorrelation and vice versa.

## Report Outline

The rest of this report is broken down into chapters containing the exploratory data analysis, the MRR models and a conclusion comparing every models’ performance (SINGULAR if using my suggestion).

Chapter 2 I WOULD WRITE IN PRESENT TENSE WHEN OUTLINING YOUR ESSAY, NOT PAST AS YOU DO FOR THE REST OF THIS PARAcovered an Exploratory Data Analysis of the equity fund dataset. The master dataset included equity and bond funds with significant (IMPORTANT OR A LOT) missing data. This chapter focused on cleaning and refining the dataset, defining the predictor variables, applying imputation techniques to address missing data, ADD COMMA and identifying relationships between the two responses: ROE and sustainability score. Chapter 3 introduced Multiple Response Linear Regression. It delved into its underlying theory and explained how it can be combined with Multiple Analysis of Variance and stepwise selection to

select the most accurate models. These models were then evaluated on the equity fund dataset.

Chapter 4 progressed to shrinkage-based regression methods. It looked at such methods which are appropriate for MRR, including Multivariate Regression with Covariance Estimation and Reduced Rank Ridge Regression. These methods were then evaluated, and their performance was assessed to determine whether they offer improvements over the MRLR models.

Chapter 5 introduced non-linear modelling through Random Forests, including their adaptation to the multiple-response context with Covariance Regression with Random Forests. Covariance Regression with Random Forests was then applied to the dataset.

Chapter 6 continued the non-linear modelling approach by examining XGBoost. This chapter also outlined Cholesky Decomposition, which enabled XGBoost to be adapted for MRR. Cholesky- GET RID OF DASHDecomposition XGBoost was then evaluated on the equity fund dataset.

Finally, Chapter 7 gave an overall comparison of the performances of every modelling approach applied. It identified the most effective method for predicting ROE and sustainability scores,NO COMMA based on? the findings, COMMA HERE and discussed potential directions for future work.

# Chapter 2 Exploratory Data Analysis

Chapter 2 provides PRESENT TENSE FOR THE CHAPTER AS YOU ARE TALKING ABOUT IT WITHIN IT an outline of the process of cleaning the equity fund dataset. This includes its source,ADD COMMA explaining the predictors, to examining correlation and multi-collinearity, to imputing missing values,ADD COMMA and testing individual and joint underlying distributions.

## Initial Data Collection and Preprocessing

The dataset was obtained from Kaggle and simplified for this analysis. This report focuseS I’D USE PRESENT TENSE OR FUTURE TENSE (DEPENDING WHERE YOU ARE IN IT RELATIVE TO WHICH PART YOU ARE TALKING ABOUT) FOR YOUR ESSAY, BUT PAST TENSE FOR YOUR RESEARCH E.G. THE DATASETon analysing equity funds, but the master dataset contained a mixture of equity and bond funds. The first step was to filter out the bond funds and keep the equities. This filtering was done simply using Excel’s filter features. A text filter was used to select the relevant rows within the category column. Then, there was an issue with the columns because there were 130 dependent variables in this intermediary dataset. However, because Many variables were for bonds only, so these were removed. Some variables, such as a fund’s isin number, were irrelevant for analysis, further reducing the number of dependent variables to 29. Because it was critical to be able to look into how chosen independent variables would affect the dependent variables, so the aim was to have 10-20 variables in the model as this is a manageable amount. Also, a lot of the remaining variables served the same purpose, which would lead to multicollinearity. For example, management fees are a significant component of ongoing cost, so the management fees column was removed. The final step in preprocessing was to remove the rows corresponding to inverse equity funds as they perform the opposite of their underlying benchmark MAYBE SPECIFY WHAT THE OPPOSITE OF THEIR BENCHMARK IS?.

I AGREE WITH THE TENSING IN THE REST OF THE PARA ABOVE BECAUSE YOU USE PAST TENSE FOR ALL YOUR PROCESSING/FILTERING ETC. IT’S THE REFERENCES TO YOUR ESSAY WHICH SHOULD BE PRESENT/FUTURE.

## Variable Explanation

WORD BUGGING AGAIN HERE category gives the type of each equity fund. The cleaned equity fund dataset contained 91 unique equity types consisting of country equities, such as "Italy Equity", and sector equities, such as "Sector Equity Precious Metals". Sector equity funds invest in firms in a specific sector or industry, such as technology, regardless of the companies’ country of operation. The focus here is on sector-specific trends and prospects. In contrast, Country equity funds invest in firms incorporated in,ADD COMMA or with extensive operations in,ADD COMMA a specific country -CAN USE DASH regardless of the industry or sector. The focus of these equity funds is on the political and economic climate of the specific country.

A rating is given by analysts assigned to the equity fund, and this takes values *{*1, 2, 3, 4, 5*}*. In category 1, funds have excellent performance and high returns relative to their risk of investment. Category 2 includes funds with a solid performance and risk versus return performance, though not as good as category 1. Category 3 includes funds that are fairly valued and perform in line with market averages, and investors are advised to hold their positions. Category 4 includes funds that underperform their peers. Lastly, category 5 includes (I THINK ITS FINE TO REPEAT INCLUDES IF IT’S THE BEST WORD AS IT ADDS CONSISTENCY/CLARITY) funds that underperform over the long term and are higher risk, meaning they should, therefore, be withdrawn from.

An equity’s risk rating implies? its market volatility of the equity. [9](#_bookmark110) Market volatility? also uses the same values as a rating but measures the risk of investing in an equity fund. Category 1 funds are less volatile and more stable,ADD COMMA and thus are suitable for conservative investors. Category 2 funds are moderately more (MORE IS IMPLIED BY ‘MODERATELY’ COMPARED TO ‘LESS’ IN CATEGORY 1)volatile but still reasonably safe. Category 3 funds have medium volatility, balancing risk and potential returns.

Category 4 funds are more volatile, with a greater possibility of returns as well as a greater extent of exposure to market volatility. Lastly, Category 5 investments are subject to high risk and volatility,ADD COMMA and should only be entertained by investors with a high tolerance for risk (ITS IMPLIED YOU MEAN TOLERANCE FOR RISK/VOLATILITY I THINK).

WORD BUGGINGrating, and risk rating are ordinal variables as they exhibit an order by definition. category is a categorical variable because it represents different non-numeric groupings; USE SEMICOLON here,USE COMMA these are countries and sectors, NO COMMA that do not have a natural order.[10](#_bookmark111) The rest of the variables are continuous and numerical. equity style score is the difference between two investment strategies: growth and value stocks. [11](#_bookmark112) Growth stocks are expected to bring in a lot of capital due to strong growth in the underlying company. [12](#_bookmark113) Value stocks are either underrated or ignored by the market, and they could gain value eventually. [12](#_bookmark113) Therefore, a lower equity style score means the stock is more of a value stock, and a higher equity style score indicates more of a growth stock. equity size score measures the performance of said equity relative to the company’s value. [13](#_bookmark114) price prospective earnings gives the ratio between a company’s share price and earnings per share. price cash flow ratio is a multiple

that compares a company’s market value to the amount of capital it generates during a select period. [13](#_bookmark114) A dividend is the money companies might decide to pay you if you own their shares, and the dividend yield

is the dividend paid as a percentage of the share price. The dividend yield factor is the dividend yield, except it is used to compare the different equities, considering variations in dividend yield performance. historical earnings growth measures how a stock’s earnings per share has grown in the last 5 years. sales growth is the increase in sales of a company’s products/ services over time. asset cash is the proportion of a fund’s assets held in cash. holdings n stock is the number of different stock holdings each equity fund holds. A stock holding is the number of other companies in which the equity fund has invested in by holding their stocks. ongoing cost is the cost of the daily running of a business. [14](#_bookmark115) fund size is the sum of all the assets in the fund.

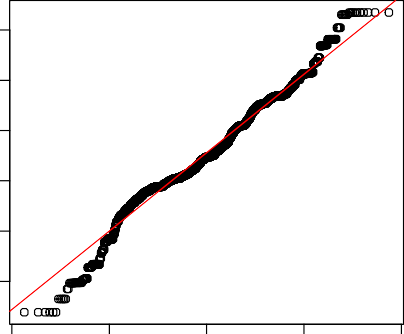
IN THIS PARA, I DON’T KNOW IF WORD HAS JUST LEFT OFF SOME CAPITAL LETTERS OR IF YOU ARE REFFERING TO VARIABLES SO UNCAPITALISING THEM ON PURPOSE. IF THE LATTER, I WOULD PUT YOUR VARIABLES IN QUOTATION MARKS TO MAKE IT CLEAR, E.G. ‘asset cash’ …

## Exploratory Data Analysis

After the dataset was cleaned and the variables specified, the next step was to perform data analysis in R and find underlying relationships between the variables. To fully represent the dataset, it is best to use actual values rather than simulate any missing values. So, this initial analysis was done with only rows with no missing values.

## Underlying Distributions

QQ plots indicate if a variable is normally distributed. The ROE and sustainability scores’ QQ plots are shown below:

**QQ Plot for roe**

Sample Quantiles

20

30

40

−4 −2 0 2 4

−10

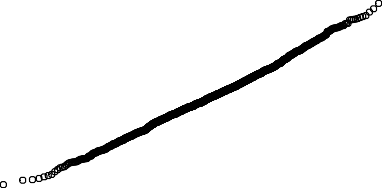
0

10

Theoretical Quantiles

**QQ Plot for Sustainability\_Score**

−3 −2 −1 0 1 2 3



Sample Quantiles

10

20

30

40

50

60

Theoretical Quantiles

Figure 2.1: QQ Plots for ROE and Sustainability Scores

From the QQ Plots above, there is strong evidence that ROE and sustainability score are normally distributed because the points on the QQ plot approximate a straight line.

## Correlation and Multi-Collinearity

It was also essential to examine the correlation within the dependent variables to determine whether any should be removed. High correlation implied multicollinearity, so this was checked before proceeding. After creating the correlation matrix for the dependent variables, there were two high correlation values. One was 0.8099, which was the correlation between price prospective earnings and price cash flow ratio. Based on the initial definitions of these variables, the latter is a more relevant factor in profitability and sustainability scores, so it was kept. The other high correlation was

-0.8029 between dividend yield factor and equity style score. Again, based on the definitions of these variables, the former is a more relevant factor in profitability and sustainability scores, so it was kept. This resulted in 12 predictor variables.

## Imputing Missing Values

Although high correlation implies multicollinearity, the opposite is not necessarily true because multicollinearity is not just about pairwise relationships. Multicollinearity arises from linear dependencies among all independent variables. For instance, if Variable *A* is a linear combination of Variables *B* and *C*, say *A* = 2*B* + *C*, multicollinearity can occur even if there is no high pairwise correlation 0.7.[15](#_bookmark116) Therefore, methods such as the Variance Inflation Factor (VIF) are used to detect multicollinearity beyond correlation. [16](#_bookmark117)

The dataset should also be complete before carrying out a VIF analysis. This involves analysing how the NA values are spread and how often they occur in each row. It was found that 90% of the rows had two NA entries or less, and because the dataset is large, only these rows were used in further analysis. There were three types of missing data here corresponding to the numerical, categorical and ordinal variables that needed to be imputed.

Random forest imputation was used to fill in the missing categorical and ordinal data for several reasons. Although it is computationally more expensive than using methods such as the mode to fill in missing data, it deals well with variables with no association. The Spearman’s Rank test confirmed this relationship between the ordinal variables, rating and risk rating. It concluded? an association of

-0.0728, indicating there is sufficient evidence to suggest a lack of association between them.

Random forest imputation also works for categorical and ordinal data through surrogate splits, and outliers influence them less as they use an ensemble of decision trees. [17](#_bookmark118) Random forest imputations also do not require prior assumptions of the underlying distribution, making them more flexible than methods such as linear model imputation, which requires underlying distribution normality. [17](#_bookmark118) The random forest model was trained on rows with no NA values, and was set to make sure it was predicting categorical and ordinal values rather than numerical ones. Ultimately, the model learned? PROBABLY MAKES SENSE BUT JUST CHECKING the relationships between the target variable and the other columns in the dataset.

Linear model imputation was used to fill in missing numerical features because it uses the relationships between variables rather than relying on simpler approaches such as mean imputation. This method assumes that each response variable is drawn from an underlying normal distribution, and this is supported by the Q-Q plots shown in Figure [2.1.](#_bookmark16) [18](#_bookmark119)

In the linear model, the ordinal predictors, which have integer values from 1 to 5, were used as numerical predictors as they have roughly equal spacing among levels.[19](#_bookmark120) The categorical predictor, category, was frequency encoded to prevent it from imposing a hierarchy in the linear model.

## Multicollinearity

VIF analysis was employed using a standard multiple response linear regression model of the responses,ADD COMMA ROE and sustainability score,ADD COMMA against the features, where the ordinal and categorical predictors were encoded as described previously. Following this, a dummy response variable was added for VIF calculation, and the VIF values were derived from this new model. If any predictors had VIF values greater than 10, they would have been removed due to multicollinearity. In this case, however, all VIF values were below 5. See Figure [A.1.](#_bookmark157) Therefore, none of the predictors exhibited multicollinearity.

Finally, this left a cleaned equity fund dataset with 1248 rows, each with individual equity fund information. In addition, there were 14 columns, with 2 response variables and 12 predictors.

## Multivariate Normality

Many of the MRR models in this report relied on the assumption of multivariate normality of the response variables. This assumption was evaluated using Mardia’s test, which assesses multivariate skewness and kurtosis. [8](#_bookmark109) The null and alternative hypotheses of Mardia’s test here were that the responses were and were not multivariate normally distributed, respectively.

In general, for Mardia’s test, if either the kurtosis or skewness statistic deviates significantly from their expected values under multivariate normality, then the null hypothesis is rejected. [8](#_bookmark109)

Mardia’s test returned p-values of 0.0732 for skewness and 0.0862 for kurtosis, meaning that in both cases, we failed to reject the null hypothesis of multivariate normality at a 5% significance level. Therefore, there was sufficient evidence to suggest a multivariate normal assumption for the responses

(see for more on how these p-values were derived).

[A.2.2](#_bookmark163)

However, the relatively low p-values suggested that the assumption may not hold perfectly, and this mild deviation from normality helped explain differences in performance between models that require distributional assumptions and those that do not.

## Response Intercorrelation

As stated in the introduction, considering response intercorrelation is what separate MRR from single-response regression. If the dataset consisted of response variables which shared no correlation, one model could be built for each response,NO COMMA and hence, MRR would not be needed.

Before proceeding with each MRR model, it was worth briefly examining the relationship between the two response variables: ROE and sustainability score. Figure [2.2](#_bookmark22) shows a scatter plot of these responses, with a fitted regression line included to highlight their correlation:

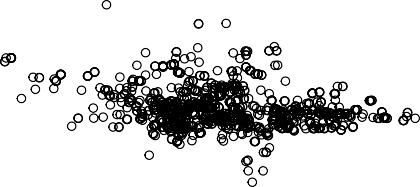
**ROE vs Sustainability Score**

Sustainability Score

40

50

−10 0 10 20 30 40



10

20

30

Return on Equity (ROE)

Figure 2.2: Scatter Plot of Return on Equity vs. Sustainability Score

The regression line above shows a slight negative slope, indicating a negative correlation between return on equity and sustainability score. Using the cor function in R gives a precise response correlation of

-0.313 (to 3sf). This means that, generally as ROE increases, the sustainability score decreases. Here were the full empirical covariance and correlation matrices between the responses (to 3sf):

**R** = 1*.*00 *−*0*.*313 *,* **Σ**

!

**Y**

*−*0*.*313 1*.*00

= 62*.*0 *−*8*.*79 *.*

*−*8*.*79 12*.*7

!

The negative correlation and covariance might be surprising given the current trends of moving toward a more sustainable climate. However, a lower sustainability score means an equity fund is a more sustainable option, so this does follow the expected trend.

# Chapter 3 Linear Regression

This chapter introduces the methodology of the first MRR model, multiple response linear regression, by starting with single-response linear regression. Stepwise selection methods are then introduced, followed by an explanation of how they can be extended to the multiple response case. Multiple analysis of variance is also covered, as it plays a key role in determining which predictors are included in the multiple response models. The accompanying code will then be explained, and the models in this chapter are evaluated using the average normalised root mean square error.

## Theory

* + 1. **Single-Response Linear Regression**

Before delving into multiple-response linear regression, let us start with single-response linear regression (SRLR). SRLR models the relationship between a single-response variable and a set of predictor variables. The general form of this model, with the dimensions subscripted, is:

**Y**(*n×*1) = **X**(*n×p*)***β***(*p×*1) + ***ϵ***(*n×*1)*,* (3.1)

where *n* is the number of observations, *p* the number of predictors (including the column of ones), **Y** the response variable vector, **X** the design matrix, ***β*** the coefficient vector and ***ϵ*** is the error variable vector.

Including a column of ones in **X** adds an intercept term to the model. This represents the expected response when all predictors are zero.

When modelling any data, the aim is to minimise the residuals. A residual is the difference between predicted and actual values. Reducing this ensures the model is as accurate as possible, which is done

by finding the most optimal coefficient vector, ***β***. This optimal coefficient vector, also known as the ordinary least squares estimator (OLS), ***β*ˆ**, is found by minimising the sum of squared residuals. It can also be defined using **X** and **Y**:

***β*ˆ**OLS **= (X*⊤*X)*−*1X*⊤*Y***.*

The model’s accuracy can then be evaluated using metrics such as the R-squared value, which measures the proportion of variance in the dependent variable explained by the independent variables.

SRLR comes with a few underlying assumptions, which relate to the properties of its error terms. They are linearity, homoscedasticity, normality, and independence of its variables. Linearity means there is a linear relationship between the dependent variables and the independent variables: *E*(*Yi|Xi*) = *β*0 + *β*1*xi*. The assumption of normality refers to the distribution of residuals, which are assumed to follow a normal distribution with mean zero and constant variance, *σ*. Homoscedasticity means we expect to see an even spread of residual values and that we can assume the model operates well for all values of the predictor variable. Finally, independence implies that the error covariance matrix is diagonal, and this is given by:

Cov(***ϵ***) = *σ*2**I***n,*

where *σ*2 is the true error variance.

Since the true error variance, *σ*2, is unknown, it is typically estimated from the residuals, ˆ***ϵ*** =

**Y** *−* **X*β***ˆ, where ***β***ˆ is the OLS estimator. An unbiased estimator of *σ*2 is given by:

*σ*ˆ2 = 1

*n − p*

ˆ***ϵ****⊤*ˆ***ϵ****,*

where *n* is the number of observations and *p* is the number of predictors. Therefore, the error covariance matrix can be estimated.

If we extended this method to model *m* responses using *m* independent single-response regression models, we would be ignoring the covariance between responses, leading to sub-optimal predictions when responses are correlated. See Equation [3.4](#_bookmark29) for a visual of this.

However, SRLR can be extended to the multiple-response setting while incorporating this structure.

This extended approach is known as multiple-response linear regression.

## Multiple Response Linear Regression

Multiple-response linear regression (MRLR) is the most basic model in MRR. Its general form is:

**Y**(*n×m*) = **X**(*n×p*) **B**(*p×m*) + **E**(*n×m*)*,* (3.2)

with,ADD COMMA

*E*(*ei*) = 0 and Cov(*ei, ek*) = *σik***I** *i, k* = 1*,* 2*, . . . , m,*

where *n* is the number of observations, *m* is the number of responses, and *p* is the number of predictors (including the column of ones), **Y** the response variable matrix, **X** the design matrix, **B** the coefficient matrix,ADD COMMA and **E** is the error variable matrix.

Including a column of ones in **X** allows the model to estimate an intercept for each response variable. These intercepts appear as the first row of the coefficient matrix,ADD COMMA **B**. Again, this represents the expected value of each response when all predictors are zero.

The first key difference between MRLR and SRLR lies in the structures of the responses, coefficients and errors. These are now matrices rather than vectors, reflecting that multiple response variables are modelled jointly.

As an illustrative example, suppose *n* = 4, *p* = 3, and *m* = 2. Then the regression equation becomes:

This expands to:



1 *x*

*x*

*e e*

**Y**(4*×*2) = **X**(4*×*3) **B**(3*×*2) + **E**(4*×*2)*.*

*y*11 *y*12 1 *x*11 *x*12



 

*b*01 *b*02

=

+







*e*11 *e*12

21 22

*y*





 

 

 

(3*×*2)

*e*31 *e*32

*e*41 *e*42

*y*



*y*31 *y*32

*y*41 *y*42

21 22

(4*×*2)

1 *x*31 *x*32

1 *x*41 *x*42

*·* *b*11 *b*12

*b*

21

*b*

22

(4*×*3)

21 22 *.*

(4*×*2)

Each row in **Y** corresponds to a single observation (i.e.,NO COMMA a data point), and each column represents one of the *m* response variables. The coefficient matrix **B** contains a separate column of regression coefficients for each response, and the error matrix **E** captures the errors associated with each response for every observation.

The second key difference between SRLR and MRLR lies in the structure of the error terms. MRLR allows for correlation between errors across different response variables within the same observation. This correlation is captured by the error covariance matrix, **ΣE** = *{σik}*, where *σik* denotes the covariance between the error terms *ei* and *ek ,I THINK ADD COMMA* associated with response variables *i* and *k,COMMA* within the

same observation. [20](#_bookmark121)

A sample version of the error covariance matrix can be estimated using a formula analogous to the unbiased estimator of *σ*2 in SRLR:

**ΣˆE**

1

=

*n − p*

**E***⊤***E***,*

where *n* is the number of observations and *p* is the number of predictors. For illustration, consider again the case where *n* = 4, *p* = 3 and *m* = 2:

**ˆ** 1 *⊤*

"*e*11 *e*21 *e*31 *e*41

*e*11 *e*12

 21 22

*e e*

#

4 2

*i*=1 *i*1

*e*

=

Σ

" Σ

4

*i*=1

Σ

Σ

*ei*1*ei*2#

**ΣE** = *n − p* **E E** =

*e*12

*e*22

*e*32

*e*42

*e*31

*e*32

4

*i*=1

*ei*1

*ei*2

4 2

*i*=1 *i*2

*e*

*.* (3.3)

*e*41 *e*42

If we used 2 independent single-response models, the off-diagonal elements of **ΣˆE** would be 0. In this case, **ΣˆE** instead becomes:

"Σ

0 #

4 2

*e*

**ˆ** *i*=1 *i*1

0 Σ

**ΣE** =

*e*

4 2

*i*=1 *i*2

*,* (3.4)

which may lead to suboptimal predictions in the presence of correlated responses.

This error covariance structure is important as it qualifies MRLR as an MRR model. It does this because it is used to compute the sample response covariance matrix, **ΣˆY**, as follows:

**ΣˆY** = Cov(**XB** + **E**)

= Cov(**XB**) + Cov(**E**)

= **B***⊤*Cov(**X**)**B** + Cov(**E**)

= **B***⊤***ΣˆXB** + **ΣˆE***,*

(3.5)

where the linearity property of covariance justifies the first expansion (see of Cov(**XB**) to **B***⊤*Cov(**X**)**B**).

[A.2.3](#_bookmark165)

for the simplification

Equation [3.5](#_bookmark30) shows that error interdependence remains, even after accounting for the predictors,

and is captured directly by the off-diagonal elements of

**Σ**ˆ **E**. Since

**Σ**ˆ **E** appears explicitly in the

expression for the response covariance matrix **Σ**ˆ **Y**, modelling this structure is essential to capturing

the covariance of the responses when modelling.

Therefore, a non-zero value of the off-diagonal element of **ΣE**, *σik*, implies that responses *i* and *k*

tend to vary together, suggesting interdependence between them.

MRLR comes with a few underlying assumptions, which can be extended from the single-response case. Linearity comes from the formula for MRLR,NO COMMA which, when expanded, is:

*Y*1 = *b*01 + *b*11*x*1 + *· · ·* + *br*1*xr* + *e*1*, Y*2 = *b*02 + *b*12*x*1 + *· · ·* + *br*2*xr* + *e*2*,*

.

.

*Ym* = *b*0*m* + *b*1*mx*1 + *· · ·* + *brmxr* + *em.*

These rows are also independent, another MRLR property from SRLR.

Normality also holds here, but it differs slightly. If we look at using several independent single-response models, each error term is independent and identically normally distributed, with *ϵi ∼ N* (0*,* **I**). However, in MRLR, the error terms, **E**, share the same distribution. They are assumed to have a multivariate normal distribution with constant variance: **E** *∼ Nm*(**0***,* **ΣE**), where **ΣE** is the covariance matrix of errors. The error terms associated with different responses may be correlated.[20](#_bookmark121) However, despite this consideration, we continue to assume that obser-

vations (i.e.,NO COMMA rows of **Y**) are uncorrelated. [20](#_bookmark121) To formalise this, let the vector of responses for the *i*th observation be denoted by:

**Y**(*i*) = **XB**(*i*) + **E**(*i*)*, i* = 1*,* 2*, . . . , m,*

with Cov(**E**(*i*)) = *σii***I** as expected, but the errors associated with different responses within the same trial may be correlated. [20](#_bookmark121) Each response variable *i* is modelled independently, but all responses share the same set of predictors.

The predictor matrix **X** must also have full column rank to ensure that the least squares estimator of **B** is well-defined. [20](#_bookmark121) Full rank ensures that the least squares estimator, **Bˆ**(*i*), exists and is unique for each response. [20](#_bookmark121) Under the assumptions of MRLR, where responses are estimated independently, OLS, which derives **B**ˆ (*i*), minimises the residual sum of squares (RSS) separately for each response in MRLR as shown here:

RSS(*i*) = **Y**(*i*) *−* **XB**(*i*)  2*.*

2

This logic for the OLS estimate is just like single-response but applied to each response variable separately:

**B**ˆ (*i*) = (**X***⊤***X**)*−*1**X***⊤***Y**(*i*)*.*

Collecting these univariate least squares estimates gives:

**B**ˆ = h**B**ˆ (1)

or equivalently:

**B**ˆ (2) *· · ·*

**B**ˆ (*m*)i = (**X***⊤***X**)*−*1**X***⊤* h**Y**(1) **Y**(2) *· · ·* **Y**(*m*)i *,*

**B**ˆ OLS = (**X***⊤***X**)*−*1**X***⊤***Y***.*

Note the similarity of this formula to the OLS estimator in SRLR.

Using the least squares estimate, **B**ˆ OLS, the matrices of the predicted values and residuals can be made. [20](#_bookmark121) The predicted values are calculated as:

**Y**ˆ = **XB**ˆ = **X**(**X***⊤***X**)*−*1**X***⊤***Y***.*

And,NO COMMA the residuals are given by:

**E**ˆ = **Y** *−* **Y**ˆ

= [**I** *−* **X**(**X***⊤***X**)*−*1**X***⊤*]**Y***.*

The predicted values and residuals can ultimately be used to evaluate MRLR’s performance, which we utilised for the equity fund dataset. See Section for more on how these criteria were used.

[3.2.1](#_bookmark35)

## Stepwise Selection Methods

Now that we have established MRLR and how it qualifies as an MRR model, we need to consider which predictors to use. Selecting the most significant predictors is crucial in evaluating model fit. There are always multiple ways to construct a model in a given situation, and we want to be able to judge which of those models is best. Here, we used selection methods.

These stepwise selection methods include forward, backward and bidirectional stepwise selection. These stepwise selection methods were used because they are more computationally efficient than best subset selection, as they consider a smaller set of models.

Forward stepwise selection starts with a model having no predictors and adds the predictor that produces the largest improvement in model fit. Model fit is usually determined using a selection criterion, for example, the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), or adjusted *R*2. At each step, the variable that contributes most significantly to this criterion is added. Backward stepwise selection begins with the full model as the initial model,ADD COMMA and successively removes

the predictor that makes the smallest contribution to the model according to the same criterion.

Bi-directional stepwise selection incorporates both forward and backward stepwise selection, with inclusion or exclusion of predictors at every step based on their impact on the model fit as a whole.

Another change from best subset selection is that all these stepwise selection methods consider the statistical impact of dropping variables that were considered previously.[21](#_bookmark122)

When using all of these selection methods in single-response regression, the “best” model is chosen for every number of predictors, and then the single best model is picked. This “best” criterion is given to the model with the smallest residual sum of squares, or largest *R*2. The single “best” model is then picked using a cross-validated prediction error, *Cp* (or the AIC), BIC, or the adjusted *R*2 value.

However, this changes for MRR as the response covariance matrix needs to be considered. Therefore, we needed to re-evaluate our selection criteria. This report used the multiple analysis of variance in conjunction with the average normalised root mean square error.

## Analysis of Variance

ANOVA primarily analyses the differences among group means and their associated variances. It helps determine whether there is a significant difference between the means of different groups by comparing the variance within groups to the variance between groups.

It can also be used to evaluate the predictor contribution to response variation. It does this by partitioning total response variance, or the total sum of squares (SST), into 2 components: the sum of squares of regression (SSR) and the sum of squares of error (SSE). The SSR is the variation in responses accounted for by the predictors. The SSE is the variation in responses not accounted for by the predictors. From their setup, These terms? are all related by the equation:

SST = SSR + SSE*.*

This decomposition is crucial in evaluating how much the predictors explain response variation. ANOVA results are displayed as a table which generates an F statistic, *F* = MSR , where MSR = SSR*/*df and MSE = SSE*/*df. This, in turn, generates a p-value and a low p-value (e.g. *p <* 0*.*05), which means that the predictors significantly impact response variation.

MSE

Sequential ANOVA is a type of ANOVA which decomposes the regression component of an ANOVA table by considering a sequence of nested models. This approach allows us to assess how the unexplained variation in the response changes as predictors are added to the model one at a time.

Here is how it works: first consider a sequence of *m* nested models *M*1 *⊂ M*2 *⊂ · · · ⊂ Mm*, where each model *Mj* contains all predictors from *Mj−*1 plus one additional predictor. Each model *Mj* is represented as:

*Mj*: E[*Y | X*] = *b*1 + *b*2*X*2 + *· · ·* + *bpj Xpj ,*

and the next model adds an extra predictor:

*Mj*+1: E[*Y | X*] = *b*1 + *b*2*X*2 + *· · ·* + *bpj Xpj* + *bpj*+1 *Xpj*+1 *.*

I PERSONALLY WOULDN’T BE PUTTING PUNCTUATION AT THE END OF EQUATIONS LIKE THE ONES ABOVE WHICH HAVE SUBSCRIPTS, BUT IF THAT’S NORMAL/YOU THINK ITS CLEAR, THEN IGNORE ME. Maybe a full stop at the end of all of them is a good compromise. Sequential ANOVA was useful in this report as it can be extended to the multivariate setting to evaluate which predictors are most appropriate for inclusion in MRLR.

## Multiple Analysis of Variance

Multivariate Analysis of Variance (MANOVA) extends ANOVA to multiple responses. It comes with the assumption that the responses need to have a multivariate normal distribution, like MRLR.[22](#_bookmark123)

MANOVA tests if the means of several dependent variables differ across independent variables. It also compares groups on a set of dependent variables simultaneously rather than conducting separate

ANOVAs for each dependent variable.[23](#_bookmark124) This report uses MANOVA to assess the impact of the predictors across the joint variation in responses. It assesses this using variance-covariance matrices instead of scalar variance values.

The total response variation, or the total sum of squares and cross products (SSCP) matrix, **T**, is partitioned into 2 matrices, like ANOVA. The first is the explained SSCP matrix, **H**, representing the response variation explained by the predictor variable(s). [24](#_bookmark125) The second is the unexplained SSCP matrix, **W**, representing the response variation not accounted for by the predictors. [24](#_bookmark125) By their set-up, **W** and **H** relate through the total SSCP matrix: **T** = **H** + **W**.

MANOVA is suitable for MRR because the total SSCP matrix, **T**, is an unscaled version of the response covariance matrix, i.e.:

**T** = (*n −* 1)**ΣY***,*

where *n* is the number of observations. This relation means MANOVA accounts for response intercorrelation by directly considering response covariances, which is ignored when using several independent univariate ANOVAs. Before proceeding with these values, let us look at how they are computed.

From single-response ANOVA, **H,COMMA** **W** and **T** are extensions of SSE, SSR and SST, respectively.

The unexplained SSCP matrix **W** is calculated as follows:

**W** = (**Y** *−* **Y**ˆ )*⊤*(**Y** *−* **Y**ˆ )*,*

where **Y** is the *n × m* matrix of observed response values, and **Y**ˆ is the *n × m* matrix of predicted

response values from the model. **Y**ˆ is computed using **Yˆ** = **XBˆ**, where **Bˆ** is the MRLR OLS estimator.

The explained SSCP matrix **H** is calculated as follows:

**H** = (**Y**ˆ *−* **Y**¯ )*⊤*(**Y**ˆ *−* **Y**¯ )*,*

where **Y**¯ is the overall mean response vector, *n ×* 1 given by **Y**¯

= 1 Σ*n*

**Y***i*.

The total SSCP matrix **T** is calculated as follows::

**T** = (**Y** *−* **Y**¯ )*⊤*(**Y** *−* **Y**¯ )*.*

*n i*=1

To test significance, MANOVA has multiple tests it can use. This report uses the Wilks’ Lambda (Λ) test, which measures the proportion of variance not explained by the independent variables:

Λ = *|***W***| ,*

*|***T***|*

where *|***W***|* and *|***T***|* are the determinants of the unexplained and total SSCP matrices, respectively.[24](#_bookmark125) A small Λ (close to 0) indicates the predictors explain a large proportion of response variation, and a large Λ (close to 1) indicates the predictors explain a small proportion of response variation.

The Wilks’ Lambda value is useful because it can be transformed into an F-statistic:

*F* = (1 *−* Λ)*/m*

(Λ)MAYBE BRACKET THE LAMBDA AS LOOKS SLIGHTLY WEIRD NEXT TO THE DIVIDE SIGN*/*(*N − m −* 1)

where: *m* is the number of dependent variables and *N* is the total sample size. The null hypothesis in MANOVA is that all group means for the dependent variables are equal. So, a significant F-test (*p <* 0*.*05) suggests at least one of the dependent variables significantly differs across groups.[23](#_bookmark124). However, this report uses the difference in Wilks’ Lambda values as specified by Sequential MANOVA. Sequential MANOVA extends sequential ANOVA by testing predictors one at a time in a nested sequence of models. It works the same as sequential ANOVA, but at each step, we compute the Wilks’

Lambda (Λ*j*) value for each model:

Λ1 *≥* Λ2 *≥ · · · ≥* Λ*m,*

where the difference: Λ*j −* Λ*j*+1 is the additional variance explained by the newly added predictor. A significant drop in Λ indicates that the new predictor significantly improves the model. [23](#_bookmark124)

Another advantage of using Wilks’ Lambda is that it prevents the inflation of Type I errors. If multiple dependent variables are analysed separately using ANOVA, the probability of finding at least one false positive increases. Since MANOVA jointly examines multiple dependent variables, it controls for the increased risk of false positives by adjusting for intercorrelations among the responses. [23](#_bookmark124) This provides a more accurate assessment of significant predictors.

## Application

* + 1. **Model Evaluation and Performance**

Multiple models were developed and evaluated throughout this report. Their performance was assessed using the average normalised root mean square error (ANRMSE).

While the root mean square error (RMSE) measures model error for each response variable separately, it retains the original units, making direct comparisons difficult. In contrast, the normalised RMSE (NRMSE) scales the error by the standard deviation of each response, returning a unitless and comparable measure of model performance. Since two response variables were considered, the NRMSE was computed for each and then averaged to obtain a single summary statistic, the ANRMSE.

Here is the formula for the ANRMSE:

 *,* (3.6)

Σ , q 1 Σ*n*

*m*

1

ANRMSE = 

*m*



*n*

*i*=1

*s*

*j*=1

(*yij − y*ˆ*ij*)2 

*j*

where *m* is the number of responses, *n* is the number of observations, *yij* is the observed value for response *j* under? IF OBSERVATION i IS LINKED TO RESPONSE j THEN I WOULDN’T USE A COMMA AS IT MAKES IT LOOK LIKE PART OF THE LIST. MAYBE USE A WORD LIKE UNDER OR A DASH WOULD BE FINE. observation *i*, *y*ˆ*ij* is the predicted value, ADD COMMA and *sj* is the sample standard deviation of the *j*th response given by:

,u 1 Σ

*n*

*s* = , *−*

*j n* 1

*i*=1

(*yij − y*¯*j*)2*.* (3.7)

The ANRMSE gives the proportion of natural variability in the responses not accounted for by the model. An excellent model has a 0-0.5 ANRMSE, a good model has a 0.5-0.6 ANRMSE, a satisfactory model has a 0.6-0.7 ANRMSE, and an unsatisfactory model has a I WOULD GET RID OF SPACE BETWEEN > AND 0.7. *>*0*.*7 ANRMSE.[25](#_bookmark126)

## Standard Multiple Linear Regression

Before proceeding with the analysis, we DO YOU MEAN I verified that the equity fund dataset followed the necessary assumptions from the Sequential MANOVA Stepwise Selection and MRLR. Most of the assumptions from MRLR and MANOVA were all catered for in the exploratory data analysis through methods such as Mardia’s Test for Skewness and Kurtosis. However, the assumption of a full rank predictor matrix, **X**, had yet to be checked. This was confirmed using QR-decomposition, which showed that the rank of **X** equalled the number of predictors.

The first model evaluated on the equity fund dataset was an MRLR using all of the available predictors. The residuals from the model were used to calculate the ANRMSE, which was 0.7606. This high value indicated that this model did not perform well on the equity fund dataset. This makes sense because the model was not built to understand the underlying complexity of the equity fund dataset, so it underfit PAST TENSE the data. Stepwise selection will consider this more, and it had been adapted

STEPWISE SELECTION CONSIDERS WHAT MORE, AND I THINK BOTH TENSES SHOULD BE CHANGED, TO PRESENT THEN NORMAL PAST. E.G. STEPWISE SELECTION CONSIDERS X MORE, AND IT WAS ADAPTED…

to the multiple response case using Sequential MANOVA.

## Code Explanation

All of these selection methods had to be manually coded because the stepAIC command, which performs stepwise selection in R, only applies to single-response regression models.

The selection process began? with different initial models depending on the chosen method. In forward selection, the model starts with no predictors. However, in backward and bidirectional selection, it starts with all available predictors included.

At each iteration, the algorithm evaluates the effect of adding or removing each candidate predictor using the functions ‘add predictors’ and ‘remove predictors’ THIS IS A GOOD EXAMPLE OF WHERE I THINK ADDING QUOTATIONS FOR YOUR VARIABLES/PREDICTORS MAKES YOUR WRITING A BIT CLEARER, respectively. The criterion for inclusion or exclusion is based on the change in Wilks’ Lambda (Λ), computed as:

∆Λ = Λprevious *−* Λnew*,*

where Λprevious is the Wilks’ Lambda from the current model, and Λnew is the value after including or excluding a given predictor.

The predictor associated with the largest reduction in Wilks’ Lambda is selected or removed, depending on the chosen selection method. If no predictor reduces Wilks’ Lambda, so ∆Λ *<* 0 for all candidates, the model remains unchanged for that step.

A key consideration in this procedure that has not been discussed thus far was the risk of overfitting. Just as in single-response regression, where the residual sum of squares always decreases with the inclusion of more variables, Wilks’ Lambda in MRR also tends to decrease monotonically as predictors are added. If used alone, this could result in models that overfit the data.

To address this, the algorithm includes an additional evaluation step using the ANRMSE, which is computed via *k*-fold cross-validation (CV) using the calculate nrmse function. A predictor is only accepted into,COMMA or removed from,COMMA the model if its inclusion or exclusion improves Wilks’ Lambda and reduces ANRMSE. This ensures that model selection favours variables that improve performance on unseen data. The new model formed is then stored in a list named ‘candidate models’ ILL STOP DOING QUOTATIONS IN GENERAL BUT I’D JUST DECIDE FOR YOURSELF WHETHER THEY ARE good THEN BE CONSISTENT.

The process continues iteratively. In bidirectional selection, the algorithm stops if two consecutive iterations fail to improve the model. In forward or backward selection, it stops once all available predictors have been evaluated without improving Wilks’ Lambda and ANRMSE.

The final model is then selected as the one with the lowest ANRMSE from the candidate models list. If two models have a similar performance, the simpler model is preferred due to its interpretability. For brevity, this overall procedure will be referred to as Sequential MANOVA Stepwise Selection for the remainder of this paper. The method is first illustrated using a simpler dataset of Study Time

versus Exam Scores before being applied to the full equity fund dataset.

## Small-Scale Example

Let us take a small-scale dataset evaluating correlated Mathematics and Science Scores against factors that may influence them:

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*1: Math Scores | *Y*2: Science Scores |
| 5 | 2 | 78 | 80 |
| 7 | 3 | 85 | 79 |
| 8 | 4 | 88 | 88 |
| 3 | 1 | 65 | 70 |
| 10 | 5 | 92 | 74 |

Table 3.1: Study Time vs. Exam Scores

Let us now walk through Sequential MANOVA Forward Stepwise Selection. The first step is evaluating each predictor one at a time to see which predictor gives the largest Wilks’ lambda difference. Let us look at *X*1 as a predictor to demonstrate how the Wilks’ Lambda difference is computed.

The first step is computing the total, explained (and subsequently? unexplained) SSCP matrices.

This starts by first computing the means to get **Y**¯ = [*Y*¯1*, Y*¯2]*⊤*:

*Y*¯1

78 + 85 + 88 + 65 + 92

= = 81*.*6*,*

5

*Y*¯2

80 + 79 + 88 + 70 + 74

= = 78*.*2*.*

5

Next, compute the deviations from the mean as follows:

|  |  |
| --- | --- |
| *Y*1*,i − Y*¯1 | *Y*2*,i − Y*¯2 |
| *−*3*.*6  3*.*4 | 1*.*8  0*.*8 |
| 6*.*4 | 9*.*8 |
| *−*16*.*6 | *−*8*.*2 |
| 10*.*4 | *−*4*.*2 |

The total SSCP matrix, **T**, is then given by:

**T** = Σ (*Y*

" Σ*i*(*Y*1*,i − Y*¯1)2 Σ*i*(*Y*1*,i − Y*¯1)(*Y*2*,i − Y*¯2)#

*i*

1*,i*

*— Y*¯ )(*Y*

* *Y*¯ ) Σ (*Y*
* *Y*¯ )2

where *i* goes from 1 to 5, which is the number of observations. Now, compute each term:

1

2*,i*

2

*i*

2*,i*

2

*,*

Σ(*Y*1*,i − Y*¯1)2 = 449*.*2*,* Σ(*Y*2*,i − Y*¯2)2 = 184*.*8*,* Σ(*Y*1*,i − Y*¯1)(*Y*2*,i − Y*¯2) = 634*.*

*i*

*i*

*i*

Thus,

**T** = 449*.*2 634 *.*

" #

634 184*.*8

The next step is to compute the unexplained SSCP Matrix, **W**, which uses the prediction from

MRLR: **Y**ˆ = **XB**ˆ . Here, we want to compute the OLS estimator for MRR, **Bˆ**

, which is an identical process to that in single-response, i.e. by simply using **Bˆ** = (**X***⊤***X**)*−*1**X***⊤***Y**. In our case:

78 85 88 65 92 *⊤*

" #

**Y** =

80 79 88 70 74

and **X** =

1 1 1 1 1 *⊤*

*.*

" #

5 7 8 3 10

Remember, only *X*1 is being used here because a model with 1 predictor is being evaluated first. Plugging these values into the formula for the OLS estimator, **B**ˆ = (**X***⊤***X**)*−*1**X***⊤***Y**, gives:

" #

**Bˆ** = 56*.*47 72*.*23 *.*

3*.*81 0*.*90

Therefore, the predicted values are: *Y*ˆ1*,i* = 56*.*47 + 3*.*81*X*1*,i* and *Y*ˆ2*,i* = 72*.*23 + 0*.*90*X*1*,i.* Now, we can compute **W** = (**Y** *−* **Y**ˆ )*⊤*(**Y** *−* **Y**ˆ ) to 4sf:

" #

**W** = 25*.*73 50*.*86 *.*

50*.*86 160*.*9

Next, we compute the Wilks’ Lambda value using the determinants of these matrices:

Λ = *|***W***|* = 1553*.*08 = 0*.*0258*,*

*|***T***|* 60090*.*2

This value is very low, meaning the model including *X*1 explains a large proportion of the variation in the responses, which is ideal.

Next, the difference between this and the previous Wilks’ Lambda value is computed. The previous case was when we had the column of ones in our design matrix, **X**. This means the model only estimated the mean of each response, so no actual predictors were included to explain differences in **Y**. Hence, the predicted values were the column means. As a result, the explained SSCP matrix, **H**, is 0. Therefore, the unexplained SSCP matrix, **W ,COMMA**equals the total explained SSCP matrix, **T**, and Λ = 1. Computing the differences gives: 1-0.0258 = 0.9742. This is compared to the Wilks’ Lambda

Difference from only using *X*2, and the predictor with the largest difference is added to the model.

To calculate the ANRMSE, say for the model with only the *X*1 predictor, start by computing the

predictions. For MRLR, the predictions are computed using **Yˆ** = **XB**ˆ out previously for this model:

where **X** and **B**ˆ

have been set

1 5 

" #

1 3 

3*.*81 0*.*90

1 7

**Yˆ** = **XB**ˆ = 1 8  56*.*47 72*.*23

1 10

75*.*52 76*.*73

83*.*14 78*.*53

= 86*.*95 79*.*43 *.*

67*.*90 74*.*93

94*.*57 81*.*23

From here, the residuals can be computed as:

78 80

 

 

85 79

75*.*52 76*.*73

 

83*.*14 78*.*53

 

 2*.*49 3*.*25 

 

1*.*88 0*.*44

 

**Y** *−* **Y**ˆ

 

= 88 88 *−* 86*.*95 79*.*43 =  1*.*07 8*.*53 

65 70

92 74

67*.*90 74*.*93

94*.*57 81*.*23

*−*2*.*89 *−*4*.*95

*−*2*.*55 *−*7*.*27

This gives the standard deviation of each response as: (10*.*60*,* 6*.*80) after plugging this into Equation

[3.7](#_bookmark37) with *n* = 5 and *m* = 2. Finally, plugging all these values into Equation [3.6.](#_bookmark36) gives an NRMSE of each response variable as (to 3dp):

" #

0*.*214 *.*

0*.*835

This gives an overall ANRMSE of 0.524 (3dp) in this example.

## Applying Stepwise Selection

This method was then applied to the equity fund dataset with ‘roe’ and ‘sustainability score’ AGAIN QUOTATION MARKS ADD CLARITY (ESPECIALLY AS USUALLY WOULD BE ROE IN CAPS) as the responses, and MANOVA Stepwise Selection was applied, with category frequency encoded.

Both Forward and bi-directional Stepwise Selection led to the same model with all the predictors kept. This model was the sum of all the predictors in the dataset and NO COMMAtherefore, had the same ANRMSE. Backward elimination returned a different model to the other selection methods. The difference was that it did not include risk rating. This gave a higher ANRMSE than the previous model with 0.7924. It makes sense for the selection methods to potentially produce a different model because each selection method has different starting points and directions, which can lead to other paths through

the model and,NO COMMA ultimately, a different set of selected predictors.

## Extending and Evaluating Stepwise Selection

Although there were already a large number of predictors in this dataset, extending the model to consider non-linear and interaction terms improved its performance. To include these, all that had to be done was to make a new function which generated the new predictors. When generating the non-linear terms, the model improved as shown by the ANRMSE reducing to 0.6893. Using the interaction terms further improved the model, and the ANRMSE was reduced to 0.5613, a noteworthy improvement.

One counterargument to introducing these terms is that these models overfit the data, but the implementation of CV prevented this. Another potential issue is that collecting this extra information can be complex. However, all this data was already available.

The table below provides an overview of all the selection models evaluated in this chapter.

|  |  |
| --- | --- |
| **Model** | **ANRMSE** |
| Full Model - MRLR using all the Predictors | 0.7606 |
| Forward Stepwise Selection - Sum of all the Predictors | 0.7606 |
| Backward Stepwise Selection - One predictor removed | 0.7924 |
| Bidirectional Stepwise Selection - Sum of the Predictors | 0.7606 |
| Bidirectional Stepwise Selection with Interaction Terms | 0.5613 |
| Bidirectional Stepwise Selection with Non-Linear Terms | 0.6893 |

Table 3.2: Selection Methods and their ANRMSE Values

See

for a graphical illustration of Model 2: Forward Stepwise Selection in action.

Stepwise selection comes with multiple benefits. Firstly, it is easy to implement. Although manual

[A.2](#_bookmark158)

code had to be made in this case, it proved to be easy to test on the dataset. Stepwise selection reduces the work of model building to the click of a button and often yields simple results. [26](#_bookmark127) Also, stepwise selection can sift through lots of predictors to identify potential relationships, mainly when dealing with high-dimensional datasets.

However, they come with a few drawbacks. Firstly, although CV can be employed, they are prone to over-fitting because they can find idiosyncrasies in the population that do not exist. [26](#_bookmark127) Also, the stepwise selection method produces models that are highly sensitive to slight variations in the data, causing inconsistent results. Another issue of this method is inflated effect sizes, so even when the model identifies valid predictors, it overestimates their effect sizes.[26](#_bookmark127) This means that the impact of significant predictors on the response variables, ROE IF THIS ROE IS CAPITALISED, I THINK YOUR PREVIOUS ONE WAS NOT SUPPOSED TO BE LOWER CASE? and sustainability score, will be inflated.

Shrinkage Methods are the next type of regression model evaluated in this report. None of the stepwise selection drawbacks apply to shrinkage methods. They address them by penalising the model coefficients, which prevents over-fitting and reduces sensitivity to outliers.

# Chapter 4 Shrinkage Methods

This chapter delves into Shrinkage Methods, focusing on Multivariate Regression with Covariance Estimation and Reduced Rank Ridge Regression, which extend Lasso and Ridge Regression. First, it introduces the theory behind Lasso and Ridge Regression and then extends them to multiple responses. Next, it delves into the theory of Multivariate Regression with Covariance Estimation and Reduced Rank Ridge Regression, and finally,COMMA applies these concepts to the equity fund dataset.

## Theory

The methods given in Chapter [3](#_bookmark23) were discrete model search methods, where the best model is selected based on information criteria. The next type of model this report will consider is shrinkage methods, which are a continuous model search approach. In these methods, the entire model is trained, but the estimated regression coefficients are reduced in magnitude and, in some cases, even set to zero. This is achieved by minimising a loss function that combines model fit with a penalty on coefficient size.

## Ridge Regression

Ridge regression shrinks estimated regression coefficients towards zero by minimising the following penalised loss function:

2

2

*L*(***β***) = **Y** *−* **X*β***

 2 + *λ * ***β***

 2*,* (4.1)

where *|| · ||*2 is the *ℓ*2-norm, *λ ≥* 0 is a regularisation parameter, and all the variables follow on from SRLR; see Equation [3.1.](#_bookmark26) The tuning parameter is crucial as it controls the strength of the penalty, balancing the trade-off between model fit and coefficient magnitude. Therefore, as *λ* increases, the penalty term gets larger, so in order to minimise the entire loss function, the regression coefficients get smaller.

Ridge regression minimises the cost function, and this can be shown by differentiating the cost function with respect to ***β***. This gives the ridge regression closed-form coefficient estimator as:

***β*ˆ**Ridge = **X***⊤***X** + *λ***I** *−*1 **X***⊤***Y***.* (4.2)

*p*

See for the full derivation of the closed-form estimator.

[A.2.4](#_bookmark167)

Ridge regression is particularly useful when the predictors exhibit high multicollinearity or when

the number of predictors,COMMA *p ,COMMA* is close to or exceeds the number of observations,COMMA *n*. In such cases, the matrix **X***⊤***X** becomes singular, making the OLS estimator unstable or undefined. The ridge penalty term *λ * ***β *** 2 regularises the inversion by ensuring that **X***⊤***X** + *λ***I** is always invertible, leading to more stable coefficient estimates. This regularisation reduces the variance of the model at the expense of a small increase in bias, which generally improves model prediction accuracy when there is predictor multicollinearity or limited sample size.

2

Ridge regression is also faster than model-search based methods, such as stepwise selection in Subsection [3.1.3,](#_bookmark31) as it only trains one model.

Ridge regression estimates are obtained by solving the least squares problem,COMMA subject to a constraint

on the sum of squared coefficients:  ***β *** 2 = Σ*p β*2 *≤ c*, where *p* is the number of predictors and *c* is

2

*j*=1

*j*

a positive constant. This constraint defines an *ℓ*2-ball,NO COMMA which,COMMA in the two-dimensional case such that *p* = 2, corresponds to a filled circle and provides a geometric interpretation of the ridge penalty. See the

right-hand graphic in Figure for this interpretation:

[4.1](#_bookmark49)

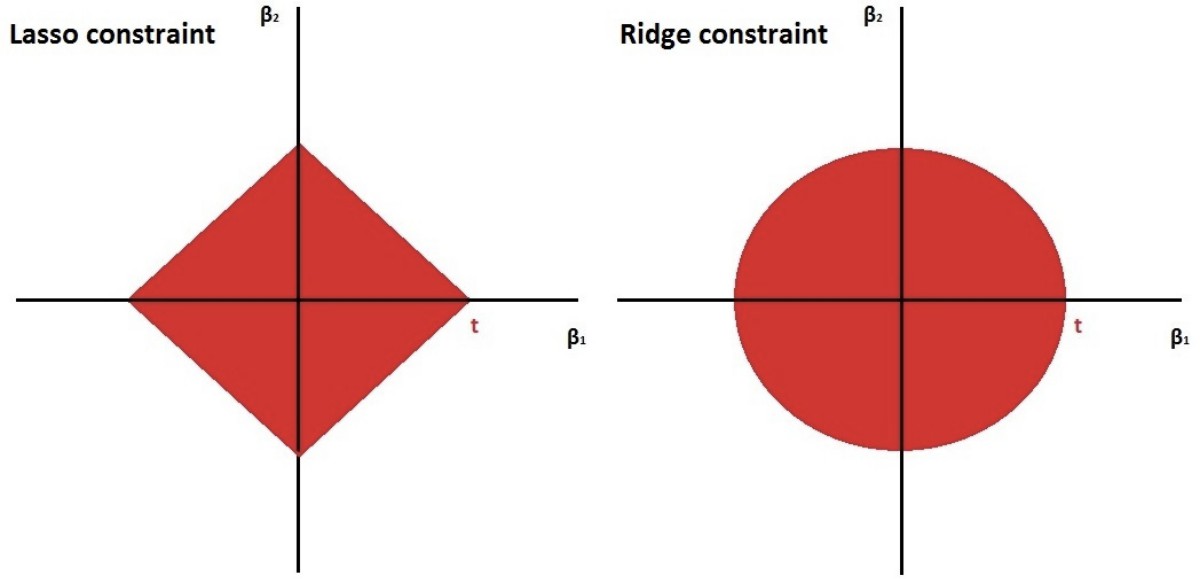


Figure 4.1: Left: the Lasso constraint *|β*1*|* + *|β*2*| ≤ c*, Right: the Ridge constraint *β*2 + *β*2 *≤ c*.

1

2

In higher dimensions, so as *p* increases, the *ℓ*2-norm forms a hypersphere, which is a higher-dimensional generalisation of the circle.

As the constraint region is circular, the ridge regression solution typically lies within the interior of the ball rather than on the axes. As a result, although ridge regression shrinks coefficients toward zero, it does not set any of them exactly to zero, even those corresponding to irrelevant predictors. This lack of sparsity is a key limitation, as ridge regression cannot perform feature selection. Ideally, a simpler model would exclude uninformative variables entirely. Lasso regression does exactly this.

## Lasso Regression

Lasso regression removes irrelevant features by shrinking their coefficients exactly to zero. Instead of penalising the squared magnitude of the coefficients like ridge, lasso penalises their absolute values. It shares the same aim of minimising a loss function as ridge but with a different penalty term:

*L*(***β***) = **Y** *−* **X*β***

2

 2 + *λ * ***β *** 1*,* (4.3)

where *|| · ||*1 is the *ℓ*1-norm and the other variables follow from Equation [4.1.](#_bookmark47)

Unlike ridge regression, Lasso has no closed-form solution because the *ℓ*1-norm penalty, ***β*** 1, is not differentiable at zero. This makes the optimisation problem non-smooth and requires iterative algorithms,COMMA such as coordinate descent,COMMA to solve it.

Like ridge regression, lasso introduces bias in exchange for reduced variance, which can improve prediction accuracy. Lasso is particularly useful in high dimensions, as it is scalable and can perform automatic feature selection by shrinking some coefficients exactly to zero. Both ridge and lasso can also mitigate multicollinearity. However, for the equity fund dataset used in this report, such issues were largely addressed during preprocessing. See Chapter [2.](#_bookmark11)

As with ridge regression, this optimisation problem can be viewed as least squares subject to a

different constraint, the *ℓ*1-norm:  ***β *** 1 = Σ*p |βj| ≤ c*. Geometrically, in the two-dimensional case,

*j*=1

when there are two predictors, this constraint defines a diamond-shaped region in the parameter space. The corners of this diamond lie on the coordinate axes, encouraging sparsity by increasing the chance that the solution lies exactly on one of them. This is why lasso can shrink some coefficients exactly to zero. See the left-hand graphic in Figure [4.1](#_bookmark49) for an illustration of this constraint.

In higher dimensions, the *ℓ*1-norm forms a cross-polytope. This is a higher-dimensional generalisation of the diamond, which continues to promote sparsity by encouraging axis-aligned solutions.

These regularisation methods, while developed for single-response models, can also be extended to MRR.

## Shrinkage Methods for MRR

Ridge regression extends to MRR by adding a Frobenius norm penalty to the coefficient matrix **B**:

1

min

**B**

2

 **Y** *−* **XB ** 2

+ *λ * **B ** 2 *,*

where *λ ≥* 0 is the regularisation parameter,  **B ** 2

*F*

*F*

*F*

*ij*

*q j*=1

*m i*=1

2 denotes the Frobenius norm

penalty, and the matrices **X**, **Y** and **B** are defined as in Equation [3.2.](#_bookmark28) [27](#_bookmark128)

= Σ

Σ

*b*

In single-response ridge regression, the penalty is given by the squared *ℓ*2-norm of the coefficient vector. See Equation [4.1.](#_bookmark47) However, since the standard *ℓ*2-norm applies only to vectors, it cannot directly penalise all the entries in the coefficient matrix. The Frobenius norm replaces it because it extends the *ℓ*2-norm to matrices, summing the squared entries across all rows and columns in **B**.

The closed-form solution for **Bˆ**Ridge in multiple response ridge regression is:

**Bˆ**Ridge = (**X***⊤***X** + *λ***I***m*)*−*1**X***⊤***Y***,* (4.4) which follows from single-response ridge regression. See Equation [4.2.](#_bookmark48)

Similarly, Lasso Regression can also be extended to MRR using the *ℓ*1-norm penalty:

1

min

**B**

2

 **Y** *−* **XB ** 2 + *λ * **B ** *.*

As in single-response lasso regression, the *ℓ*1-penalty shrinks some elements of **B** to exactly zero, enabling variable selection in the regression model.

2

1

Despite the flexibility of multivariate ridge and lasso, these methods do not account for the intercorrelation between response variables, which is critical to do IF IT’S THE ACCOUNTING FOR INTERCORRELATION THAT’S CRITICAL I WOULD ADD ‘TO DO’ in MRR. However, there are extensions of them that address this limitation. Two such approaches are reduced rank ridge regression and multivariate regression with covariance estimation.

## Reduced Rank Ridge Regression

In MRLR, predictions are generated using **Y** = **XB**ˆ where **Bˆ** = (**X***⊤***X**)*−*1**X***⊤***Y**.

OLS OLS

Reduced rank ridge regression (RRRR) modifies the estimation of the coefficient matrix using multiple response ridge regression (see equation [4.1.3)](#_bookmark51),COMMA and a rank constraint. It aims to solve the following optimisation problem:

**B**ˆ (*λ, r*) = arg min

**B**:rank(**B**)*≤r*

 **Y** *−* **XB ** 2

+ *λ * **B ** 2 *,* (4.5)

where *r ≤* min*{p*, *m}* controls the rank of **B** and *λ ≥* 0 is the ridge parameter. Note: *m* is the number of responses and *p* is the number of predictors.

*F*

*F*

Although the unconstrained ridge solution **B**ˆ Ridge is typically full rank, which is confirmed on the

equity fund dataset via QR-decomposition (see Subsection [3.2.2),](#_bookmark38) a rank constraint remains useful. It encourages dimensionality reduction and reflects the assumption that the responses are caused by a small number of latent variables. [27](#_bookmark128) This results in a model that not only captures the underlying low-dimensional structure of the data,COMMA but is also interpretable.

Before proceeding, assume that **X** and **Y** have been mean-centred, so each column has zero mean. As a result, the number of predictors, *p*, excludes the intercept term, and the column of ones is omitted from the design matrix. This assumption will be further justified later in this section.

The first step involves transforming this problem into a standard Reduced Rank Regression problem on an augmented dataset which incorporates the ridge penalty.[27](#_bookmark128) This augmentation involves adding artificial rows to the original predictor and response matrices.

The augmented least squares problem is equivalent to applying the closed-form ridge estimator from Equation [4.4,](#_bookmark52) when no rank constraint is applied. Augmentation is used instead in RRRR because it allows the RRRR problem to be solved more efficiently using Singular Value Decomposition (SVD), rather than solving the regularised optimisation problem directly.

Augmentation is also performed separately for each fixed *λ*, as the optimal value is not known in advance and is selected via CV in this report.

Here is augmentation written mathematically:

*∗*

**X**

(*n*+*p*)*×p*

### X

*√λ***I**

=

*∗*

(*n*+*p*)*×m*

! *,* **Y**

= **Y**! *.*

**0**

The augmented data leads to the following rank-constrained optimisation problem:

**B**ˆ (*λ, r*) = arg min

rank(**B**)*≤r*

 **Y***∗ −* **X***∗***B ** 2 *,*

where **B** can be estimated using OLS. See for the full derivation of this.

[A.2.5](#_bookmark169)

*F*

This problem can be simplified further due to the orthogonality of the OLS estimate. This orthogonal projection property means the squared error loss function can be decomposed into 2 parts:

 **Y***∗ −* **X***∗***B ** 2

*F*

= **Y***∗ −* **Y**ˆ *∗ *2

+ **Y**ˆ *∗ −* **X***∗***B ** 2 *,*

where **Y**ˆ *∗*

*R*

*R*

*F*

*R*

*F*

= **X***∗***B**ˆ *∗*

denotes the Ridge Regression estimate. [27](#_bookmark128) The first term of this equation does not

involve **B** so the objective function can be further simplified into:

*R*

**B**ˆ (*λ, r*) = arg min

*||***Y**ˆ *∗ −* **X***∗***B***||*2 *.* (4.6)

rank(**B**)*≤r*

The next step involves decomposing the matrix **Y**ˆ *∗*

*R*

*R*

*F*

This starts by taking its SVD:

*τ*

= Σ *di***u***i***v** *,*

ˆ *∗ ⊤*

**Y**

*R* *i*

*i*=1

where the *di*’s denote the singular values, AND **u***i ∈* R*n×*1 ,COMMA **v***i ∈* R*m×*1 denote the left and right singular vectors of **Y**ˆ *∗* ,NO COMMA respectively. [27](#_bookmark128)

*R*

This step is important because RRRR captures response intercorrelation through **v***i*. This can be proven by how the sample response covariance, **ΣY**, is defined:

**Σ** = 1 (**Y** *−* **Y**¯ )*⊤*(**Y** *−* **Y**¯ ) = 1 **Y***⊤***Y***.*

**Y**

*n −* 1 *n −* 1

The last equality holds because we have mean-centred **Y**. See relate without mean centring.

[A.2.5](#_bookmark170)

for how **ΣY** and **Y***⊤***Y** would

This is important because the SVD of **Y** in matrix form is: **Y** = **UDV***⊤*. Now, compute **Y***⊤***Y**

using its SVD form:

**Y***⊤***Y** = **VD***⊤***U***⊤***UDV***⊤*

= **VD**2**V***⊤,*

where the final equality is produced/derived? because **U** is orthonormal, meaning **U***⊤***U** = **I**, and **D** = **D***⊤*. [28](#_bookmark129)

This means **V** clearly diagonalises **Y***⊤***Y**. Therefore, it directly represents the principal directions of

response covariance. Thus, the columns of **V**, **v***i*, capture response intercorrelation.

Then, the Eckhart-Young Theorem is used to provide the best rank *r* approximation to **Y**ˆ *∗*

*R*

*r*

in the

Frobenius norm as follows:

*r*

*i*

**Y**ˆ *∗* = Σ *di***u***i***v***⊤.* [27](#_bookmark128)

*i*=1

The next step to RRRR is to define the projection matrix, **P***r*, which projects the predictions onto

an *r*-dimensional subspace: **P***r* = Σ*r* **v***i***v***⊤*.

*i*=1

*i*

Let **B**ˆ (*λ, r*) = **B**ˆ *∗* **P***r* and clearly, rank(**B**ˆ (*λ, r*)) *≤ r*, as rank(**P***r*) = *r*. [27](#_bookmark128) Substituting this into the fitted values, **X***∗***B**ˆ (*λ, r*), we obtain predictions corresponding to the rank-*r* solution in Equation [4.6,](#_bookmark55) and thus to the original problem in Equation [4.5:](#_bookmark54)

*R*

*∗* ˆ *∗* ˆ *∗*

Σ*τ*

! ,Σ*r*

*⊤* Σ

*⊤* ˆ *∗* [27](#_bookmark128)

**X B**(*λ, r*) = **X B***R***P***r* =

*i*=1

*⊤*

*di***u***i***v***i*

*j*=1

**v***j***v***j*  =

*i*=1

*r*

*di***u***i***v***i* = **Y***r.*

See

for the third simplification.

This has shown that the proposed solution **B**ˆ (*λ, r*) = **B**ˆ *∗* **P***r* is the minimiser of WORD REALLY FUCKING WITH SOME OF THESE LINES BUT IM GOING THROUGH WHAT IS STILL INTACT.

[A.2.5](#_bookmark171)

*R*

and hence the

[4.6](#_bookmark55)

RRRR optimisation problem. [27](#_bookmark128) Writing it explicitly in terms of **X***,* **Y***, λ*,NO COMMA and *r* gives the following:

**B**ˆ (*λ, r*) = **B**ˆ *∗* **P***r* = (**X***∗⊤***X***∗*)*−*1**X***∗⊤***YP***r* = (**X***⊤***X** + *λ***I**)*−*1**X***⊤***YP***r.* [27](#_bookmark128)

*R*

Therefore, **Y**ˆ (*λ, r*) = **XB**ˆ (*λ, r*) = **X** **X***⊤***X** + *λ***I** *−*1 **X***⊤***YP** = **Y**ˆ **P** .

*r*

*λ r*

From this estimate, the role of the ridge closed-form solution becomes clear: the full-rank ridge

estimator **B**ˆ *∗* is first computed, and then its fitted values **Y**ˆ *λ* = **X***∗***B**ˆ *∗* are projected onto a lower-

*R*

*R*

dimensional subspace. The rank-*r* constraint then arises from this projection, which maps the fitted values from the original *m*-dimensional response space to an *r*-dimensional subspace spanned by the top *r* right singular vectors.

While RRRR accounts for response covariance by applying SVD to a low-rank approximation of the fitted response matrix, an alternative approach is to model the inverse error covariance matrix, which captures conditional dependencies between responses,NO COMMA directly. This idea underlies Multivariate Regression with Covariance Estimation, which jointly estimates both the regression coefficients and the inverse error covariance structure.

## Multivariate Regression with Covariance Estimation

Multivariate Regression with Covariance Estimation (MRCE) builds upon MRLR by jointly estimating the regression coefficient matrix **B** and the inverse error covariance matrix **Ω** = **ΣE***−*1.

Before solving this problem, we again assume that **X** and **Y** have been mean-centred. Unlike in RRRR, this is for ease of notation rather than a necessity.

Starting with the familiar MRLR model:

**Y**(*n×m*) = **X**(*n×p*)**B**(*p×m*) + **E**(*n×m*)*,*

where *p* no longer includes the column of ones.

Each row **y***i ∈* R1*×m* of **Y** follows the conditional distribution: **y***i|***x***i ∼ Nm*(**x***i***B***,* **ΣE**)*,* from the multivariate normality of the errors. Therefore, the probability density function for the *ith* row, i.e. a single observation, is:

*⊤*

(2*π*)*m/*2*|***Σ**

*|*1*/*2

2

*i*

*i*

**E**

*p*(**y**

*i*

*i*

*|* **x** ) = 1 exp *−* 1 (**y**

**E**

*—* **x B**) **Σ**

*−*1(**y**

*—* **x B**) *.* (4.7)

*i*

*i*

Equation [4.7](#_bookmark57) implies that, in MRLR, the negative log-likelihood function of (**B***,* **Ω**) can be expressed up to a constant as:

*g*(**B***,* **Ω**) = tr 1 (**Y** *−* **XB**)*⊤*(**Y** *−* **XB**)**Ω** *−* log *|***Ω***|.* [29](#_bookmark130)

*n*

See Section for the full derivation.

[A.2.6](#_bookmark174)

The MRCE estimation for **B** extends this by adding 2 penalties to the negative log-likelihood function,COMMA *g,COMMA* to construct a sparse estimator of **B** depending on **Ω**:

(**B**ˆ *,* **Ω**ˆ ) = arg min

,*g*(**B***,* **Ω**) + *λ*1

Σ *|ωj′j|* + *λ*2 Σ

*p*

*q*

Σ *|bjk|*,

*,* (4.8)

**B***,***Ω** ,

*j′̸*=*j*

*j*=1 *k*=1 ,

where *ωj′j* are the elements of **Ω**, *bjk* are the elements of **B**, and *λ*1 controls sparsity in **Ω** and *λ*2 for **B**. [29](#_bookmark130)

A sparse estimator of **B** forces many coefficients to zero, improving feature selection and efficiency.

Equation [4.8](#_bookmark58) can be solved in 2 ways: by solving it for **Ωˆ** with **Ω** fixed at a chosen **Ω**0:

DO YOU MEAN HERE …: BY SOLVING FOR BHAT WITH B FIXED AT A CHOSEN B0, AND BY SOLVING FOR OMEGAHAT WITH OMEGA FIXED AT A CHOSEN OMEGA0.. OR ACTUALLY IS IT OMEGAS THAT ARE FIXED FOR BHAT, I DON’T GET THE SENTENCE BASICALLY.

with **B** fixed at a chosen **B**0 and for **Bˆ**

* + - 1. This is the formula for **Bˆ**, fixing **Ωˆ** as **Ω**0, therefore the **Ω** terms in Equation from the objective function:

[4.8](#_bookmark58)

can be removed

**B**ˆ (**Ω** ) = arg min ,tr 1 (**Y** *−* **XB**)*⊤*(**Y** *−* **XB**)**Ω** + *λ*

0

Σ *|b*

*|*, *.* (4.9)

**B** , *n*

0 2 *jk*

*j,k*

* + - 1. This is the formula for **Ω**, fixing **B** as **B**0, therefore the **B** terms in Equation [4.8](#_bookmark58) can be removed from the objective function:

**Ω**ˆ (**B** ) = arg min ,tr 1 (**Y** *−* **XB** )*⊤*(**Y** *−* **XB** )**Ω** *−* log *|***Ω***|* + *λ* Σ *|ω ′ |*, *.* (4.10)

,

0

0

0

,

**Ω** , *n*

1 *j j*

*j′̸*=*j*

Before delving further into this problem, let us discuss how this model qualifies as MRR and,NO COMMA consequently, how it accounts for response intercorrelation.

Since MRCE estimates both **B** and **Ω** simultaneously, the coefficient matrix is optimised considering response covariance. This reasoning follows from the explanation for why MRLR qualifies as an MRR. See Equation [3.5](#_bookmark30) for how **ΣY**, **B** and **ΣE** relate.

The error covariance matrix,COMMA **ΣE ,COMMA**describes marginal dependencies between response variables, whereas the inverse error covariance matrix,COMMA **Ω** = **ΣE***−*1 ,COMMAdescribes conditional dependencies.[30](#_bookmark131) This means that if Ω*ij* = 0, then responses *Yi* and *Yj* are conditionally independent given all other responses and, if Ω*ij ̸*= 0, then responses *Yi* and *Yj* have a dependency not accounted for by the predictors. [30](#_bookmark131) Thus, MRCE does not assume independent residuals,COMMA but instead models how responses are related even after accounting for predictors.

In contrast, MRLR assumes that **ΣE** is unstructured but fixed during estimation, and does not account for conditional residual dependencies when estimating **B**.

Equations [4.9](#_bookmark59) and are each solved using a distinct approach. Equation [4.9](#_bookmark59) is solved by the

[4.10](#_bookmark60)

following algorithm. Let us call it Algorithm 1. This algorithm comes from Rothman et al. (2010) and needs some set-up: given **Ω** and an initial value **B**ˆ (0), let **S** = **X***⊤***X** and **H** = **X***⊤***YΩ**.

1. For the *m*-th iteration, set **B**ˆ (*m*) **B**ˆ (*m−*1). Visit all entries of **B**ˆ (*m*) in some sequence, and for each entry (*r, c*), update ˆ*b*(*m*) by minimising the objective function with respect to that single

*←*

*rc*

entry, keeping all other coefficients fixed:

ˆ*b*(*m*) *←* sign ˆ*b*(*m*) + *hrc − urc* ˆ*b*(*m*) + *hrc − urc* *− nλ*2 *,*

+

*rc*

*rc*

*srrωcc*

*rc*

*srrωcc*

*srrωcc*

where *urc* = Σ*p*

*j*=1

Σ

*jk*

*q k*=1

ˆ*b*(*m*)*srjωkc* and (*a*)+ means taking max(*a,* 0).

1. If Σ

is the ridge closed-form estimator of the coefficient matrix. See Equation

ˆ*b*(*m*) *−* ˆ*b*(*m−*1) *< ϵ* Σ

ˆ*b*Ridge , then stop, otherwise go to the prior step. Here:

[4.4](#_bookmark52)

**B**ˆ Ridge

Note: *ϵ* is a convergence tolerance, specifying how small the change in **B**ˆ for the algorithm to be considered converged.

[4.10](#_bookmark60)

*j,k*

*jk*

*jk*

*j,k*

*jk*

for this estimator.

must be between iterations

Equation

is solved using the graphical lasso (glasso) algorithm due to its speed. [29](#_bookmark130) Glasso is,

in fact, widely used for estimating sparse inverse covariance matrices in high-dimensional cases.

The glasso algorithm for MRCE, from Friedman et al. (2008), is as follows:

1. Initialise **Ω** with: **Ω**(0) = (**Σ**(*m*) + *λ*1**I**)*−*1*,* where

**E**

**Σ**(*m*) = 1 (**Y** *−* **XBˆ**(*m*))*⊤*(**Y** *−* **XBˆ**(*m*))*,*

**E**

*n*

is the initial error covariance matrix.

1. For the *k*-th iteration and for each *j* = 1*,* 2*, . . . , q* in **Ω**(*k*), solve the graphical lasso regression problem using the previous iteration **Ω**(*k−*1):

***β***ˆ = arg min 1

***β***

2

 **Ω**(*k−*1) *−* **Ω**(*k−*1) ***β *** 2 + *λ * ***β *** *,*

*−j,j*

*−j,−j*

2

1

1

where **Ω**(*k−*1) is the *q −* 1 vector containing all elements in the *j*-th column except for the

*−j,−j*

diagonal element Ω(*k−*1) and ***β*** is an estimate of **Ω**(*k*)

by solving the lasso regression problem

*jj*

using **Ω**(*k−*1).

*−j,j*

1. Update **Ω** by filling in the corresponding row and column using:

(*k*)

**Ω**

*−j,j*

(*k*)

*−j,−j*

= *−***Ω**

***β***ˆ*.*

1. Repeat steps 2 and 3 until convergence, so when *||***Ω**(*k*) *−* **Ω**(*k−*1)*||∞ < ϵ*, where *ϵ* is chosen.

Now, we can explicitly outline the full MRCE algorithm,COMMA from Rothman et al. (2010),COMMA as follows: For fixed values of *λ*1 and *λ*2, initialise **B**ˆ (0) = **0** and **Ω**ˆ (0) = **Ω**ˆ (**B**ˆ (0)).

1. Compute **B**ˆ (*m*+1) = **B**ˆ (**Ω**ˆ (*m*)) by solving Equation

[4.9](#_bookmark59)

1. Compute **Ω**ˆ (*m*+1) = **Ω**ˆ (**B**ˆ (*m*+1)) by solving Equation

using Algorithm 1.

using the glasso algorithm.

[4.10](#_bookmark60)

1. If Σ

ˆ*b*(*m*+1) *−* ˆ*b*(*m*) *< ϵ* Σ

ˆ*b*Ridge then stop; otherwise, go back to Step 1.

This process uses blockwise descent to compute a local solution for the objective function value. [29](#_bookmark130)

[4.8](#_bookmark58)

*j,k*

*jk*

*jk*

*j,k*

*jk*

with steps 1 and 2 decreasing

MRCE performance also depends on the regularisation parameters, which are selected via CV:

(*λ∗, λ∗*) = arg min Σ

*K*

1

2

*λ*1*,λ*2

*k*=1

 **Y**(*k*) *−* **X**(*k*)**B**(*−k*)  2 *,*

*λ*1*,λ*2

*F*

where **Y**(*k*) is the matrix of responses with observations in the *k*th fold, **X**(*k*) is the matrix of predictors

of observations in the *k*th fold, and **B**(*−k*)

*λ ,λ*

1 2

is the estimated regression coefficient matrix computed with

observations outside the *k*th fold, with tuning parameters *λ*1 and *λ*2. [29](#_bookmark130)

The next section will talk about how these 2 methods were evaluated on the equity fund dataset.

## Application

* + 1. **Small-Scale Example**

Let us now apply both of these shrinkage methods to a small dataset before proceeding to the equity fund dataset. We will use the familiar Mathematics and Science scores data:

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*1: Math Scores | *Y*2: Science Scores |
| 5 | 2 | 78 | 80 |
| 7 | 3 | 85 | 79 |
| 8 | 4 | 88 | 88 |
| 3 | 1 | 65 | 70 |
| 10 | 5 | 92 | 74 |

Table 4.1: Study Time vs. Exam Scores

#### Reduced Rank Ridge Regression

This example will show a rank-1 reduction of RRRR using *λ* = 0*.*1 as the shrinkage parameter.

Based on our prior justification of RRRR as an MRR model, mean-centring DASH?? is essential to prevent biasing from the mean structure. The initial predictor matrix,NO COMMA **X**, NO COMMA – BE CONSISTENT IN YOUR SENTENCE FOR X AND Yand initial response matrix **Y** are:

5 7 8 3 10 *⊤*

" #

**X** =

2 3 4 1 5

*,* **Y** =

78 85 88 65 92 *⊤*

*.*

" #

80 79 88 70 74

After mean-centering DASH??, i.e. **X˜** = **X** *−* **X¯** and **Y˜** = **Y** *−* **Y¯** , the predictor and response matrices become:

**X**˜ =

*−*1*.*6 0*.*4 1*.*4 *−*3*.*6 3*.*4 *⊤*

*−*1 0 1 *−*2 2

" #

*,*

**Y**˜ =

*−*3*.*6 3*.*4 6*.*4 *−*16*.*6 10*.*4 *⊤*

1*.*8 0*.*8 9*.*8 *−*8*.*2 *−*4*.*2

" #

*.*

We apply reduced rank ridge regression with *λ* = 0*.*01 by first incorporating ridge regularisation, which we define as:

**X***∗* = "

**X˜**

*√λ***I**

# *,* **Y***∗* = "**Y˜** # *.*

**0**

Since *λ* = 0*.*01, we compute *√λ* = *√*0*.*01 = 0*.*1, so the augmented matrices are:

**X***∗* =

*−*1*.*6 0*.*4 1*.*4 *−*3*.*6 3*.*4 0*.*1 0 *⊤*

*−*1 0 1 *−*2 2 0 0*.*1

" #

*,* **Y***∗* =

*−*3*.*6 3*.*4 6*.*4 *−*16*.*6 10*.*4 0 0 *⊤*

1*.*8 0*.*8 9*.*8 *−*8*.*2 *−*4*.*2 0 0

" #

*.*

The Ridge penalty is incorporated into **X***∗* through *√λ***I**. The augmented system behaves like a standard regression problem, meaning we solve for **B** using the standard OLS formula. Let us denote

this as **B**ˆ *∗*

*R*

as a reminder that we are incorporating ridge regression.

This formula is: ˆ *∗* = (**X***∗⊤***X***∗*)*−*1**X***∗⊤***Y***∗*. After subbing in **X***∗* and **Y***∗*, this gives us the aug-

**B**

*R*

mented coefficient matrix, **B**ˆ *∗*

*R*

(to 2dp) as:

ˆ *∗*

**B**

*R*

= 7*.*40 *−*2*.*28

*−*6*.*18 5*.*47

" #

Thus, the ridge predictions are (to 2dp):

ˆ *∗* = **X***∗***B**ˆ *∗* =

**Y**

*R*

*R*

*−*1*.*82 *−*0*.*91 2*.*28 *−*2*.*73 3*.*19 *−*0*.*23 0*.*55

"*−*5*.*67 2*.*96 4*.*19 *−*14*.*29 12*.*81 0*.*74 *−*0*.*62#*⊤*

The next step is to perform singular value decomposition on this prediction, i.e. **Y**ˆ *∗* = **UDV***⊤,* with

*R*

singular values (to 2dp) of 21.21 and 2.29, giving:

**D** = "21*.*21 0 # *.*

0 2*.*29

For full derivation of **D***,* **U** and **V**, see [A.2.5.](#_bookmark172)

Since *D*1 = 21*.*21 *≫ D*2 = 2*.*29, the best rank (*r*=1) approximation is taken on the first column of

**D,COMMA** and so,NO COMMA I DON’T REALLY GET THE WORDING HEREtake the first columns of **U** and **V**. Therefore, **Y**ˆ *∗ ≈ d*1**u**1**v***⊤,* where (to 2dp):

*d*1 = 21*.*21*,* **u**1

= h*−*

0*.*28 0*.*13 0*.*22

1

*−*0*.*69 0*.*62 0*.*03

1

*⊤*

i *,* **v**

*−*0*.*02

1 = h

*⊤*

i *.*

0*.*97 0*.*22

However, this is just an intermediary step as we still need to rescale the coefficient matrix. This is done through the projection matrix, **P***r* (to 2dp):

**P** = **v v***⊤* = "0*.*95 0*.*22#

*r*

1

1

0*.*22 0*.*05

Finally, the reduced rank ridge regression estimator is:

**B**ˆ *r*

= **B**ˆ *∗* **P***r*

= 6*.*54 1*.*49

*−*4*.*68 *−*1*.*07

" #

Now, we can do the final predicted values, which are **Yˆ** = **X˜***∗***Bˆ***r*, giving (to 2dp):

*R*

**Yˆ** =

*−*1*.*32 0*.*60 1*.*02 *−*3*.*24 2*.*94

"*−*5*.*78 2*.*62 4*.*47 *−*14*.*18 12*.*87#*⊤*

Remember, these predictions are on the **mean centred Y**. We can remove this mean centring, but that does not affect the model fit evaluation which uses?? the ANRMSE.

#### Multivariate Regression with Covariance Estimation

Now, we perform an MRCE iteration on Table [4.1](#_bookmark63) with *λ*1 = *λ*2 = 0*.*1. Remember we want to mean

centre **X** and **Y**, so let us use **X˜**

and **Y˜**

from the RRRR calculation:

**X**˜ =

*−*1*.*6 0*.*4 1*.*4 *−*3*.*6 3*.*4 *⊤*

*−*1 0 1 *−*2 2

" #

*,*

**Y**˜ =

*−*3*.*6 3*.*4 6*.*4 *−*16*.*6 10*.*4 *⊤*

1*.*8 0*.*8 9*.*8 *−*8*.*2 *−*4*.*2

" #

*.*

From here on, **X** = **X**˜ and **Y** = **Y**˜ . This is to make notation easier. Let us first compute **Bˆ**Ridge

because that is used in determining when Algorithm 1 stops:

**Bˆ**Ridge = (**X***⊤***X** + *λ* **I**)*−*1**X***⊤***Y** = " 5*.*07 *−*0*.*77#

2

*−*2*.*19 2*.*89

For Algorithm 1, initialise **B**ˆ (0) = **0** and **Ω**ˆ (0) = **Ω**ˆ (**B**ˆ (0)). To compute **Ω**ˆ (0), first do the error, **E**(0):

**E**(0) = **Y** *−* **XBˆ**(0) =

*.*

1*.*8 0*.*8 9*.*8 *−*8*.*2 *−*4*.*2

"*−*3*.*6 3*.*4 6*.*4 *−*16*.*6 10*.*4 #*⊤*

Now, compute the residual covariance matrix **Σ**(**0**):

**E**

**Σ**(**0**) = 1 **E**(0)*⊤* **E**(0) = "89*.*84 30*.*28#

**E**

*n*

30*.*28 36*.*96

Next, add this to a regularisation parameter,COMMA *λ*2 = 0*.*1. The regularisation parameter avoids dividing by zero when inverting **ΣE**(0) for the precision matrix (to 2sf):

" #

**Ω**(0) = 0*.*015 *−*0*.*013

*−*0*.*013 0*.*037

Now compute **B**ˆ (1) = **B**ˆ (**Ω**ˆ (0)) by solving Equation [4.9](#_bookmark59) using Algorithm 1. The manual calculation of this will be shown on 1 element of **B**(1): ˆ*b*(1). First compute **H** and **S**:

11

**S** = **X***⊤***X** = "29*.*2 17# *,* **H** = **X***⊤***YΩ**(0) = "1*.*37 *−*0*.*41# *.*

17 10

0*.*78 *−*0*.*21

Now set **Bˆ**(1) = **Bˆ**(0) and update ˆ*b*(1) with the minimiser of the objective function along its coordinate

11

Direction,COMMA given by:

ˆ*b*(1) *←* sign ˆ*b*(1) + *h*11 *− u*11 ˆ*b*(1) + *h*11 *− u*11 *− nλ*2 *,*

11 11

*s*11*ω*11

11

*s*11*ω*11

*s*11*ω*11 +

Σ

where *u*11 = Σ2

2

*k*=1

*j*=1

*jk*

ˆ*b*(1)*s*1*jωk*1*.* After subbing in and computation, this gives: ˆ*b*11 = 1*.*95*.* Doing

this on every element, we get:

**Bˆ**(1) = 1*.*95 0 *.*

" #

0 0

The next step of Algorithm 1 involves checking for convergence of **Bˆ** relative to **Bˆ**Ridge. Here, if:

Σ ˆ*b*(1) *−* ˆ*b*(0) *< ϵ* Σ ˆ*b*Ridge *,*

*jk*

*jk*

*jk*

*j,k*

*j,k*

then stop; otherwise, go to the prior step. The left-hand side simplifies TO OR TOO??:

Σ ˆ*b*(1) *−* ˆ*b*(0) = Σ *|*ˆ*b*(1)*|* = 1*.*95*.*

*jk*

*jk*

*jk*

*j,k*

*j,k*

And the right-hand side, TO OR TOO??:

*ϵ* Σ ˆ*b*Ridge = 10*.*92*ϵ.*

*j,k*

*jk*

Therefore CHANGE ONE OF THE THEREFORES TO SOMETHING ELSE E.G. Therefore, we want …, which means that …, we want 1*.*95 *<* 10*.*92*ϵ*. Therefore *ϵ* ≳ 0*.*18. Let us choose *ϵ* = 0*.*2 so we can move on to the next phase of MRCE, estimating **Ω** using the graphical lasso algorithm. Firstly, let us set a new initial **Ω**, using this updated **Bˆ**:

**Σ**(1) = 1 (**Y** *−* **XBˆ**(1))*⊤*(**Y** *−* **XBˆ**(1)) = "25*.*24 19*.*97# *.*

**E**

*n*

19*.*97 36*.*96

When inverting, again use a regularisation parameter,COMMA *λ*1 = 0*.*1,COMMA to prevent a zero determinant. This gives:

" #

**Ω**(0) = 0*.*069 *−*0*.*037 *.*

*−*0*.*037 0*.*047

Like for Algorithm 1, we will only apply this manually to one element. Here, it will be Ω(1). Now, let us solve the graphical lasso regression problem using **Bˆ**(1). For *j* = 1, which updates the first column and row of **Ω**, we define:

21

(**Σ**(1))*−*1*,−*1 = 36*.*96*,* (**Σ**(1))*−*1*,*1 = 19*.*97*.*

**E E**

We compute the inverse submatrix: Ω(1) = (**Σ**(1))*−*1 = 0*.*027 and Ω(1)

= (**Σ**(1))*−*1

= 0*.*050 to

contribute to ***β*ˆ**:

*−*1*,−*1

**E** *−*1*,−*1

*−*1*,*1

**E** *−*1*,*1

**ˆ** (1)

***β*** = *−***Ω**

*−*1*,−*1

(**Σ**(1))*−*1*,*1 = *−*0*.*00135*.*

Next, ***β*ˆ** undergoes soft-thresholding:

**E**

***β*ˆ** = sign(***β*ˆ**) max(*|****β*ˆ***| − λ*1*,* 0) = 0 The updated element is computed as:

= *−***Ω**

(1)

**Ω**

*−*1*,*1

(1)

*−*1*,−*1

***β***ˆ = 0*.*

This occurs for each element in **Ω**(1) and these steps continue until convergence, so when *||***Ω**(*k*) *−*

**Ω**(*k−*1)*||∞ < ϵ*, where *ϵ* is chosen for controlled estimation. WORD MIGHT JUST BE FUCKING WITH IT BUT IT’S A BIT UNCLEAR IF THIS IS HOW UR SENTENCE IS WRITTEN.

This completes one full MRCE iteration: initialising **B**(0) and **Ω**ˆ (0) and then updating to **B**(1) and **Ω**ˆ (1) after solving the penalised optimisation problem. This process continues iteratively until convergence. The final **Bˆ**,NO COMMA is then used to compute the predictions: **Yˆ** = **XBˆ**(*n*). This can again have its centring removed, but it is not needed when evaluating the ANRMSE.

## Code Explanation

The MRCE model was applied using the MRCE package, which jointly estimates the regression coefficient matrix and the inverse error covariance matrix of the response variables. The optimal values of the regularisation parameters *λ*1 and *λ*2 were selected using a 5-fold CV procedure,COMMA implemented in the fit mrce function. A grid of candidate values was evaluated by training and validating the model across CV folds, NO COMMAand the parameter pair, (*λ*1*, λ*2), which minimised the cross-validated ANRMSE that? was used for final model fitting. Test-set predictions were then computed using the fitted coefficient matrix. Next, RRRR was then evaluated using the fit rrridge function, which applies a rank constraint to the regression coefficient matrix. A ridge-penalised estimate is first computed to stabilise the solution, followed by SVD to retain only the top singular components, enforcing a reduced rank. The resulting

coefficient matrix is then used to generate test-set predictions, which are evaluated via ANRMSE.

IF THESE LAST 2 SENTENCES IN PRESENT TENSE WERE PART OF YOUR PROCESS, THEY SHOULD ALSO BE IN PAST TENSE LIKE THE REST OF THE PARA.

Next, additional feature transformations were applied to try to improve the performance of these methods. A degree-two polynomial expansion was applied using ‘poly’, and second-order interaction terms were generated using ‘model.matrix’ MAYBE ADD QUOTATIONS IN THIS SENTENCE. Finally, all models are compared based on their test set ANRMSE scores as specified previously. AGAIN, IS LAST SENTENCE PAST TENSE?

## Results

These regression models were applied to the dataset to evaluate how well the predictors model the response variables: ROE and sustainability score. The performance of each model was assessed using the ANRMSE, as was the case with the other models in this report.

The baseline models show that RRRR had the smallest ANRMSE of 0.751, followed by MRCE at 0.761. This means that the reduced-rank constraint in RRRR enabled it to better capture common structures PLURAL I THINK among responses, particularly where there were weak intercorrelations between? them. MRCE, by contrast, involves an additional step of covariance estimation. This introduces bias where response intercorrelations are weak, which they were here,NO COMMA with a correlation of -0.3171 between the ROE and sustainability score. MRCE will performs better under stronger correlations among response variables. The inclusion of polynomial and interaction feature transformations improved performance in both models. MRCE benefited most from polynomial features, whose ANRMSE fell from 0.761 to 0.678 — the lowest ANRMSE for any of the polynomial forms. RRRR also benefited to a lesser extent,NO COMMA from 0.751 to 0.727, suggesting that the inclusion of polynomial expansion introduced unnecessary complexity

not well used under a low-rank constraint. DUNNO WHATS HAPPENING WITH WORD HERE.

Adding interaction terms improved performance further. MRCE had the lowest overall ANRMSE of 0.678, and RRRR achieved 0.723. This suggests that MRCE’s covariance estimation benefits more from structured interactions between predictors, whereas RRRR is impacted more by the added multicollinearity from such terms.

These results indicate that the performance of each model depends on the type of introduced features. Subsequently, the regression models were fitted to the data to examine how well the predictors model the response variables: ROE and sustainability score. Each model’s performance was measured using ANRMSE, with the smaller value reflecting the better performance.

Generally, While MRCE had the smallest ANRMSE in the most complex feature setting, RRRR showed a more stable performance across conditions. Thus, RRRR could be considered the more stable shrinkage method for this dataset.

|  |  |
| --- | --- |
| **Model** | **ANRMSE** |
| Model 1: MRCE | 0.7612 |
| Model 2: RRRR | 0.7510 |
| Model 3: MRCE including Polynomial Terms | 0.6781 |
| Model 4: RRRR including Polynomial Terms | 0.7276 |
| Model 5: MRCE including Interaction Terms | 0.6777 |
| Model 6: RRRR including Interaction Terms | 0.7231 |

Table 4.2: Comparison of ANRMSE values across shrinkage methods

The models in this chapter were also surprising because they performed as well as, or sometimes

even worse than, MRLR in Chapter on the equity fund dataset (see Table [3.2](#_bookmark43) for the evaluation

[3](#_bookmark23)

of the Chapter model). This was unexpected given that MRCE and RRRR are designed to handle

[3](#_bookmark23)

multiple responses with more considerations, particularly in the presence of multicollinearity or high-dimensional predictors, meaning they should generally be more accurate in modelling.

However, the predictors used here were not highly correlated after the pre-processing steps described in Chapter [2,](#_bookmark11) and the dataset is not high-dimensional (*p* = 14, *n* = 1269). As such, the benefits of shrinkage and dimensionality reduction were less pronounced. In fact, standard OLS outperformed both MRCE and RRRR, likely because it preserved the relationship between the predictors and responses,COMMA without the added bias introduced by regularisation.

Another aspect that was examined was how each feature contributes to each response.

Feature Importance (MRCE vs. RRRidge)

roe sustainability\_score

sales\_growth sales\_growth

risk\_rating\_encoded risk\_rating\_encoded

rating\_encoded rating\_encoded

price\_cash\_flow\_ratio price\_cash\_flow\_ratio

ongoing\_cost holdings\_n\_stock

Feature

historical\_earnings\_growth

ongoing\_cost holdings\_n\_stock

historical\_earnings\_growth

Model

 MRCE

RRRidge

fund\_size fund\_size

equity\_size\_score equity\_size\_score

dividend\_yield\_factor dividend\_yield\_factor

category\_encoded category\_encoded

asset\_cash asset\_cash

−0.1 0.0 0.1 0.2 −0.2 0.0 0.2

Coefficient Value

Figure 4.2: MRCE vs RRRR Feature Contributions

For ROE CAPS?, both models identify price cash flow ratio, category encoded, and equity size score as important predictors. However, MRCE applies stronger shrinkage to these variables because it estimates both the regression coefficients and the precision matrix simultaneously. As a result, price cash flow ratio has a much smaller coefficient in MRCE, suggesting that RRRR depends on it more heavily. sales growth has little influence in either model, while asset cash is slightly negative in MRCE but negligible in RRRR. MAYBE QUOTATION THESE VARIABLES, LIKE sales growth ETC.

For the sustainability score, the differences between models are more noticeable. sales growth is the strongest positive predictor in both cases, with MRCE assigning it a slightly larger coefficient. On the other hand, historical earnings growth has a strong positive effect in RRRR but is heavily shrunk in MRCE, likely due to concerns about multicollinearity, despite being addressed earlier in Chapter [2.](#_bookmark11) Interestingly, category receives a higher coefficient in MRCE, suggesting it places more emphasis on the type of equity fund compared to RRRR. equity size score is close to zero in MRCE but still contributes in RRRR, and price cash flow ratio again has a larger absolute coefficient in RRRR. Although shrinkage methods like MRCE and RRRR are helpful in reducing overfitting and handling high-dimensional data, their linearity-dependent nature can limit their expressiveness. Here, as seen, there will be DOES PRESENT OR FUTURE MAKE MORE SENSE? a loss of some interaction and patterns. Random Forests mitigate THESE? flaws by employing an ensemble of decision trees for learning non-linear interactions and patterns, eliminating the need for feature engineering. The application of an ensemble also reduces overfitting, which further supports

the notion that Random Forests are particularly suitable for handling complex datasets.

# Chapter 5 Random Forests

This chapter delves into Random Forests. It starts with classification and regression trees, from which we can create an ensemble of decision trees. This gives rise to random forests, which improve predictive performance. The forests are then extended to MRR, and the loss function is adjusted to form Covariance Regression Random Forests, which are tested on the equity fund dataset.

## Theory

The previous modelling approaches used in this project, MRLR (see Chapter [3)](#_bookmark23) and MRCE and RRRR (see Chapter [4),](#_bookmark44) rely on global parametric assumptions. These include properties such as the multivariate normality of the response,COMMA and linear relationships between predictors and responses. Tree-based models, however, are non-parametric, meaning they require no prior assumptions about the data. These algorithms work by using a set of splitting rules to partition? the feature space into smaller regions with similar response values.

## Classification and Regression Trees

Random forests are tree-based models built upon classification and regression trees (CARTs) as a building block. A CART partitions the training data into groups with similar response values,COMMA and then predicts the same category for all data within a given subgroup.

The CART starts with a root node containing all the objects,COMMA and is then divided into “leaf” nodes by recursive binary splitting. [31](#_bookmark132) Each leaf node relates to a hyper-rectangle in the feature space, *Rj*, *j* = *{*1*, . . . , J}*. In other words, the feature space defined by *X*1*, X*2*, . . . , Xp* is split into *J* distinct regions which do not overlap, as shown below:

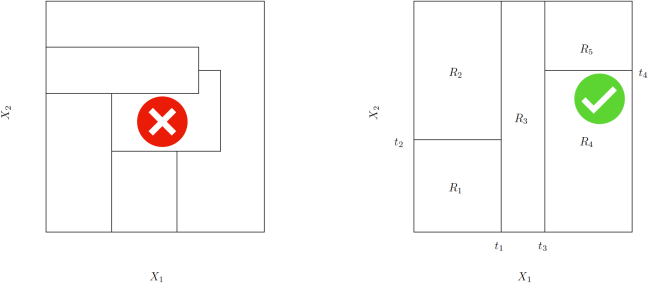


Figure 5.1: Incorrect vs. Correct Partitioning

The Greedy-Fitting Algorithm ensures each split is chosen to minimise the RSS:

Σ Σ *yi − y*ˆ*R* 2 *,*

*J*

*j*

*j*=1 *i*:*xi∈Rj*

where *y*ˆ*Rj* is the mean response for the training observations within the *j*-th rectangle.

A regression tree is used because the responses here, ROE and sustainability score, are continuous. A regression tree assigns a value to each predictor space, *Rj*. This means that the model predicts the output (i.e. which feature space) based on the average response values for all observations in that subgroup.

The Greedy fitting algorithm provides good predictions for the training data but may overfit said data?, resulting in poor performance on the test dataset. However, there exists a method called pruning that can mitigate this issue. Pruning involves removing branches that contribute little to the tree’s prediction accuracy. This is typically based on a validation set or a complexity penalty.

A common approach to pruning is the weakest link pruning algorithm, which iteratively prunes the least important branches to control the bias–variance trade-off. This algorithm introduces a non-negative tuning parameter for regression trees, *α*, which minimises the following cost-complexity objective:

*|T |*

Σ

Σ (*yi − y*ˆ*R* )2 + *α|T |,*

*j*=1 *i*:*xi∈Rj*

*j*

where *|T |* is the number of terminal nodes (size) of the tree *T* .

## Bagging for CARTs

Although CARTs are easy to interpret and are similar to standard decision-making processes, the trees generally have high variance. This means that small sample changes lead to significant changes in the fit, and they tend to have poor predictive accuracy. Bootstrap aggregating, also known as bagging, improves the stability and accuracy of classification and regression algorithms by averaging models. Bagging helps reduce variance and avoid over-fitting, but the model becomes more challenging to interpret.

For regression trees, all the predictions are averaged to obtain:

*f*ˆbag

*B*

(*x*) = 1 *f*ˆ*b*(*x*)*, B*

Σ

*b*=1

where *B* is the total number of bootstrap samples (i.e.,NO COMMA the number of trees trained), and *f*ˆ*b*(*x*) is the prediction from the *b*-th tree.

Bagging for CART addresses the overfitting issue in two ways: it can grow large trees with minimal (or no) pruning and rely on the averaging effect of bagging to reduce variance, or alternatively, it can prune each tree individually.

Bagging is advantageous when there is a large amount of data, as is the case here for the equity fund dataset. This is because the empirical distribution will be closer to the actual underlying population’s distribution. Additionally, it is the best option when the aim is to minimise the variance of a predictor. However, bagging also brings some disadvantages. Firstly, interpretability is lost as the final estimate is not a tree. Secondly, it is an ensemble prediction, i.e. multiple trees are combined to make the final prediction. [32](#_bookmark133)THERE WAS A SECOND FULL STOP HERE: DO JUST DO A BRIEF SCAN OF PUNCTUATION, ITS DIFFICULT FOR ME TO TELL WHEN ITS WORD BUGGING OR IF ITS ACTUALLY A MISTAKE. Also, bagging trees are inherently correlated. This is problematic because the more correlated the random variables are, the smaller the variance reduction of their average, which can

undermine one of its key advantages.

## Random Forests

Random Forests make bagged CART models more independent, improving variance reduction in their ensemble predictions. They do this by resampling observations and restricting the model to random subspaces, *X′ ⊂ X* . This ensures the tree explores different parts of the data, meaning the ensembles are more diverse and hence influential. [33,3](#_bookmark134)[4](#_bookmark135)

The Random Forests algorithm is used in the following way: a bootstrap resample (*y∗, x∗*)*n*

is taken

*i i i*=1

of the training data, like for bagging. Then, the tree is built and each time a split in a tree is considered, *m* predictors are randomly selected out of the full set of *p* predictors as split candidates,COMMA and the best split within those *m* predictors is found. Finally, the first two steps are repeated, averaging the prediction of all the regression forest trees.

Random forests are a great option, but they are not interpretable when bagging and using random subspaces. Thus, quantifying the importance of each variable can be difficult. There are two popular approaches which quantify this importance.

One approach is to run a loop over each tree in the forest to determine the significance of each feature variable *xj*, where *j* = 1*, . . . , p*. Every node that splits on *xj* for each tree is identified, and the improvement in the chosen loss criterion, such as accuracy or the Gini index, is calculated for each split. Each improvement is then summed for every tree in the forest, indicating the significance of *xj*. Another approach is to use out-of-bag (OOB) error estimates, which can be computed via bagging,NO COMMA

to determine the significance of each feature *xj*, where *j* = 1*, . . . , p* [33](#_bookmark134). For each tree, the OOB samples (i.e.,NO COMMA the observations that were not included in the bootstrap sample used to train that tree) are used to evaluate predictive accuracy. To assess the importance of *xj*, the entries in the *j*-th column of the OOB feature matrix, denoted *x*oob, are randomly permuted, breaking the association between *xj* and the rest of the variables. The modified matrix, *x*oob*∗*, is then passed through the tree,NO COMMA and the resulting change in predictive accuracy is computed. This decrease in accuracy caused by the permutation is averaged across all trees, providing a measure of the importance of feature *xj*.

Random forests inherit the advantages of bagging and trees, thus?? tuning is rarely needed. However, they are challenging to implement and have issues with extrapolation.

## Multivariate Regression Trees

CARTs are limited to single-response variables, making them unsuitable for modelling relationships between multiple correlated responses. Multivariate Regression Trees (MRTs) generalise CARTs by modelling all responses simultaneously. [35](#_bookmark136)

MRTs build decision trees by recursively splitting the data along predictor variable thresholds,COMMA in order to minimise variability across the response matrix. Each split aims to group observations with similar values NOT SURE ABOUT THE WORDING HEREacross all response variables. [35](#_bookmark136) This recursive splitting means that MRTs are able to detect patterns that occur only within specific regions of the predictor space, including non-linear relationships and interactions that other models may miss.

Although MRTs can be grown until each leaf contains a single observation, this leads to overfitting. Therefore, in practice, trees are pruned using CV, typically selecting the smallest tree within one standard error of the minimum cross-validated error. [36](#_bookmark137) This is known as the 1-standard error rule.

Unlike single-response regression trees, MRTs reveal how relationships between predictors and responses vary across different regions of the predictor space.[35](#_bookmark136)

MRTs adapt the impurity criterion from CARTs to account for all response variables simultaneously, evaluating splits by minimising the total variation across the multivariate response. [36](#_bookmark137) The impurity measure in MRTs minimises this total variance, and it is the total sum of squares of the responses around the multivariate mean at each node:

*n p*

Σ Σ

impurity = (*yij − y*¯*j*)2 *,*

*i*=1 *j*=1

where *yij* represents the *j*-th response for the *i*-th sample, and *y*¯*j* is the mean response at the node. [31](#_bookmark132) Because impurity minimisation is performed over the joint space, MRTs naturally cluster correlated responses, thereby preserving their multivariate structure. However, MRTs do not explicitly model

changes in the covariance structure across predictor values.

The resulting tree structure is highly interpretable, with each node representing a split on an explanatory variable,COMMA and each leaf corresponding to a distinct cluster. This makes it straightforward to visualise local interactions and the influence of predictors.

However, different visualisation tools are needed for MRTs to interpret their results. For instance, at each node of the tree, a histogram shows the distribution of each response within that node, helping to visualise the multivariate outcomes and understand the impact of splits on all responses jointly. [31](#_bookmark132) In Figure [5.2,](#_bookmark72) the left panel visualises the MRR setup: a response matrix, with two continuous response variables *Y*1 and *Y*2, is regressed on a set of predictors *X*1*, X*2*, X*3, shown as distinct columns in the predictor matrix. The right-hand panel displays the fitted MRT, where each internal node represents a binary split on one of the predictors. Each terminal node contains a pair of bar plots showing the mean values of *Y*1 and *Y*2 for the observations within that group. This helps assess how

each split affects the joint distribution of responses.

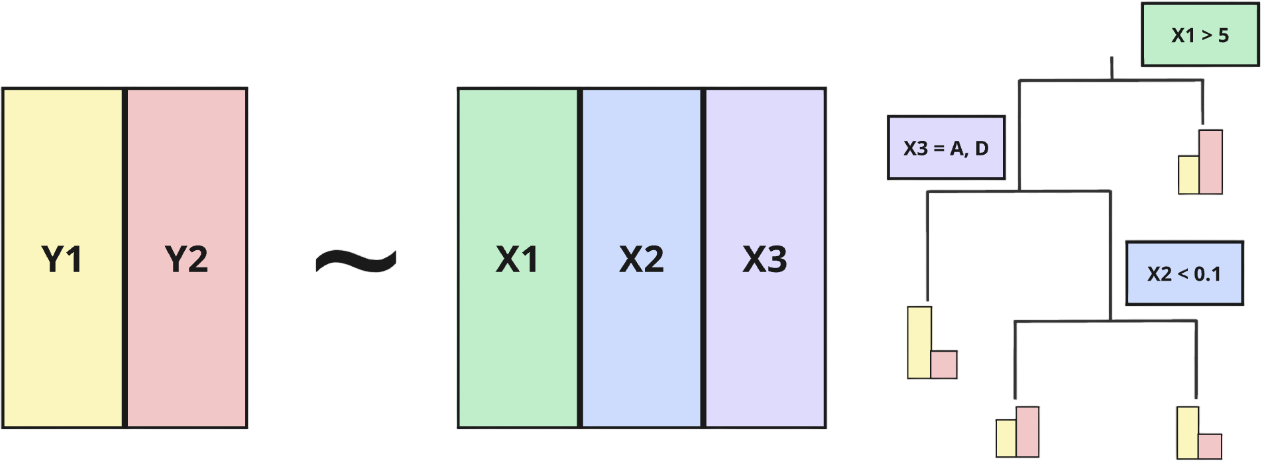


Figure 5.2: MRT Splitting. [35](#_bookmark136)

Note: if a terminal node contains fewer bars than the number of responses, this indicates that one or more response variables have a mean of zero within that group. In other words, bars are only displayed for response variables whose node-wise means are non-zero.

Although MRTs improve on CARTs in MRR, they come with some drawbacks. One such drawback is the inherent instability of tree-structured predictors,COMMA caused partly by the split function’s greedy optimisation. [31](#_bookmark132) Multivariate random forests can be used instead,COMMA to resolve this issue.

## Multivariate Random Forests

Multivariate Random Forests (MRFs),COMMA similar to or as opposed to?? univariate random forests, COMMA extend MRTs by constructing an ensemble of them through bootstrap resampling and random predictor subsampling. [37](#_bookmark138) This ensemble process helps mitigate the instability of single-tree models, making MRFs less sensitive to small changes in the data.

However, while MRFs are a great extension of MRTs, they exhibit an important limitation in MRR. Although their loss function can be adjusted to implicitly consider response correlations, MRFs do not explicitly force I THINK FORCE IS BETTER GRAMMATICALLY, BUT IF IT DOESN’T MAKE SENSE THEN ENFORCE IS FINE that each split to lead to a meaningful change in the response covariance structure. As a result, when responses are highly correlated, MRFs fail to consider the correlations optimally when building trees, potentially leading to suboptimal splits. [37](#_bookmark138)

## Covariance Regression with Random Forests

Covariance Regression with Random Forests (CRRFs) extend MRFs by incorporating a splitting criterion,COMMA which maximises differences in sample covariance estimates between child nodes. Unlike MRFs,

which focus solely on predicting the conditional mean, CRRFs explicitly account for response intercorrelation. They extend tree-based methods to estimate the conditional covariance matrix of a multivariate response,COMMA given a set of covariates. [38](#_bookmark139) While CRRFs are designed to estimate covariance structures, they inherently compute the sample response mean at each terminal node. These mean values can be direct response estimates, allowing CRRFs to extend to MRR.

Let Σ**y***i* be the true conditional covariance matrix of **y***i* (the responses), based on covariates **x***i*, and Σ**X** bethe collection of all conditional covariance matrices for *n* observations: THE FIRST PART OF THIS SENTENCE WAS A BIT UNCLEAR TO ME SO I TRIED TO WRITE IT HOW I INTERPRETED IT, WHICH EVEN IF WRONG ID STILL LOOK AT MAKING IT CLEARER.

Σ**X** = *{*Σ**y** : *i* = 1*, . . . , n}.* [38](#_bookmark139)

*i*

Also, let Σˆ**y***i* be the estimated conditional covariance matrix of **y***i* based on covariates **x***i*, and Σˆ**X** be the collection of estimated conditional covariance matrices for *n* observations. [38](#_bookmark139) IN THE SENTENCE BELOW YOU USE THE COMMAND ‘TRAIN A RANDOM FOREST’ INSTEAD OF ‘A RANDOM FOREST IS TRAINED’. A FEW PAGES BEFORE I CHANGED ALL OF YOUR ACTIVE COMMANDS TO PASSIVE ONES, FOR EXAMPLE I WOULD HAVE CHANGED ‘TRAIN’ TO ‘IS TRAINED’. ITS FINE TO DO EITHER, I FEEL THE LATTER IS MORE SUITABLE (PASSIVE), BUT IF YOU PREFER THE FORMER THAT’S FINE, JUST MAKE SURE THE ENTIRE PARAGRAPH IS CONSISTENT.

First, train a random forest with the set of covariates **X** to find subgroups of observations with similar covariance matrices of **Y,COMMA** using decision trees to uncover the data structures. These trees are built with a splitting criterion that will be defined later. THIS SENTENCE IS A GOOD EXAMPLE OF WHY I THINK PASSIVE IS BETTER, BECAUSE IT WOULD READ WORSE IF YOU SAID, ‘BUILD THESE TRESS WITH A SPLITTING CRITERION THAT WILL BE DEFINED LATER’. SO FOR CONSISTENCY I WOULD ALWAYS STICK TO PASSIVE PERSONALLY.[38](#_bookmark139) The tree-growing process follows the CART algorithm [39](#_bookmark140) which aims to obtain subgroups of observations with distinct covariance matrices. Hence, a customised splitting rule at each node will be used to increase the difference in covariance matrices between two child nodes in the tree.

|  |  |  |  |
| --- | --- | --- | --- |
| [40,4](#_bookmark141) | 1,4 | 2,4 | [3](#_bookmark144) |

Before defining this splitting rule?,COMMA let us first set up more notation. **Σ***L* is the sample response covariance matrix estimate, so Σˆ**y***i* of the left node, which is written as follows:

**Σ***L* = 1 (**y**

Σ

*i*

*nL −* 1 *i∈tL*

*—* **Y**¯

*L*)(**y***i*

*—* **Y**¯

*L*)*⊤,*

where *tL* is the set of indices (or the positions within the original data),NO COMMA of the observations in the left node, *nL* is the left node size, and:

**Y**¯ = 1 Σ **y** *.* [38](#_bookmark139)

*L*

*nL*

*i*

*i∈tL*

The estimate of the sample response covariance matrix for the right node, **Σ***R*, is calculated similarly, where *tR* is the set of indices of observations in the right node and *nR* is its size.

The splitting criterion of CRRF is:

*√nLnR × d*(**Σ***L,* **Σ***R*)*,* (5.1)

where *d*(**Σ***L,* **Σ***R*) is the Euclidean distance between the upper triangular part of the two matrices,COMMA and is computed as follows:

*d*(**D***,* **E**) = , Σ(*Dij − Eij*)2*,*

*i*=1 *j*=*i*

,uΣ

*q*

*q*

where **D***q×q* and **E***q×q* are symmetric matrices. [38](#_bookmark139) The best split among those possible is the one that maximises [5.1.](#_bookmark75) This split is what facilitates CRRF as an MRR model as it explicitly uses a response covariance-based splitting criteria. CRITERIA OR CRITERION?

The final covariance matrices are estimated using random forests. For a new observation, nearest neighbour observations are used to estimate the response mean and final covariance matrix, ensuring both the predicted value and its uncertainty are captured. [38](#_bookmark139) This set of observations is called the Bag of Observations for Prediction (BOP).

For a new observation **x***∗*, the set of nearest neighbour observations are formed with the oob

observations. [44,4](#_bookmark145)[5](#_bookmark146) The *BOPoob* for a new observation is:

*B*

[

*BOPoob*(**x***∗*) = *Ob*(**x***∗*)*,*

*b*=1

where *B* is the tree numbers and *Ob*(**x***∗*) is the set of oob observations in the same terminal node as

**x***∗* in the *b*th tree, and each tree is built with a selected random sub-sample. [38](#_bookmark139)

After training the random forest with the new split for a new observation **x***∗*, we form *BOPoob*(**x***∗*), which is key for response prediction. [38](#_bookmark139) The response mean and covariance matrix are then estimated using the sample mean and covariance matrix of the observations in *BOPoob*(**x***∗*), respectively:

**Y**ˆ *∗* = 1

*|BOPoob*(**x***∗*)*|*

*i∈BO*Σ*Poob*(**x***∗*)

*yi,*

ˆ *∗*  1

**Σ**

**Y** = *|BOP* (**x***∗*)*| −* 1

*oob*

(*yi*

*i∈BOPoob*(**x***∗*)

Σ

*—* **Y**ˆ *∗*)(*yi*

*—* **Y**ˆ *∗*)*⊤,*

where *yi* is an observation from *BOPoob*(**x***∗*), **Y**ˆ *∗* is the predicted response mean ,COMMAand **Σ**ˆ *∗*

**Y**

represents

the estimated conditional covariance matrix of the responses given **x***∗*.

## Application

* + 1. **Small-Scale Example**

We will work with this familiar dataset again:

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*1: Math Scores | *Y*2: Science Scores |
| 5 | 2 | 78 | 80 |
| 7 | 3 | 85 | 79 |
| 8 | 4 | 88 | 88 |
| 3 | 1 | 65 | 70 |
| 10 | 5 | 92 | 74 |

Table 5.1: Study Time vs. Exam Scores

Each tree is trained on a bootstrap sample, which is obtained by randomly drawing observations with replacements from the original dataset. For the provided dataset, a possible bootstrap sample could be, for example, rows 1,1,2,4 and 5. Here, you can see that row 1 has been repeated, and row 3 has been left out. Another possible bootstrap sample is just getting each row once, so rows 1,2,3,4 and 5. This corresponds to:

(5*,* 2*,* 78*,* 80)

 

 

(7*,* 3*,* 85*,* 79)

**X** =  (8*,* 4*,* 88*,* 88)  *.*

 

(3*,* 1*,* 65*,* 70)

 

(10*,* 5*,* 92*,* 74)

After constructing a bootstrap sample, recursive partitioning is performed by selecting the split that maximises the scaled Euclidean distance between the covariance matrices of the left and right child

nodes. The Euclidean distance is computed as:

,uΣ

*d*(**Σ***L,* **Σ***R*) = , Σ

*q*

*q*

(**Σ***L −* **Σ***R*)2*,* (5.2)

*ij*

*ij*

*i*=1 *j*=*i*

where **Σ***L* and **Σ***R* are the empirical covariance matrices of the left and right nodes, respectively.

In reality, the CRRF will evaluate all possible splits and compute [5.1,](#_bookmark75)NO COMMA but,COMMA for simplicity, let us assume the split that maximises [5.1](#_bookmark75) is *X*1 *<* 7. It is important to understand how [5.1](#_bookmark75) is computed for any split. First, let us show how this is done for *X*1 *<* 7. This starts by splitting the data into 2 child nodes: left and right. The left child node contains the observations and responses of:

**X** = 5 2 78 80 *,*

" #

*L*

3 1 65 70

and the right child node contains:

7 3 85 79



 

**X***R* = 8 4 88 88 *.*

 

10 5 92 74

For each node, the mean response is computed as:

**Y¯** *L*

= 1 **Y**

2 *L*

Σ

= h71*.*5 75i *,*

**Y¯** *R*

= 1 **Y**

3 *R*

Σ

= h88*.*33 80*.*33i *.*

The covariance matrices are then estimated using:

**Σ***L* = 1 (**Y**

Σ

*i*

*nL −* 1 *i∈L*

*—* **Y¯**

*L*)(**Y***i*

*—* **Y¯**

)*⊤,* **Σ***R* = 1 (**Y**

*nR −* 1 *i∈R*

Σ

*L*

*i*

*—* **Y¯**

*R*)(**Y***i*

*—* **Y¯**

*R*)*⊤.*

The computed covariance matrices for the left and right nodes are:

**Σ***L* = "142*.*25 105*.*25# *,* **Σ***R* = " 41*.*67 *−*10*.*67# *.*

105*.*25 97*.*25

*−*10*.*67 21

Applying the Euclidean distance formula to these matrices gives (to 2dp):

*d*(**Σ***L,* **Σ***R*) = √(142*.*25 *−* 41*.*67)2 + 2(105*.*25 *−* (*−*10*.*67))2 + (97*.*25 *−* 21)2 = 206*.*89*.*

Finally, the scaled Euclidean distance is:

*√nLnR × d*(**Σ***L,* **Σ***R*) = 206*.*89*√*2 *×* 3 = 506*.*77

Since CRRF selects the split that maximises this distance, we can see the split at *X*1 = 7 does lead to a large Euclidean distance between the response sample covariance matrices of each child node.

When predicting, each new observation is assigned to a terminal node based on its feature values. The prediction is computed using the BOP within the same node. For a new test point **x***∗* = (6*,* 3), which falls into the left node, the relevant BOP observations here are:

*BOPoob*(**x***∗*) = *{*(78*,* 80)*,* (65*,* 70)*}.*

The predicted mean response is then given by:

**Yˆ** *∗* = 1

*|BOPoob*(*x∗*)*|*

**Y***i*

*i∈BOPoob*(*x∗*)

Σ

= 71*.*5 *.*

75

" #

The covariance structure of the predictions is estimated using:

**Σˆ***∗* = 1

*|BOPoob*(*x∗*)*| −* 1

(**Y***i*

*i∈BOPoob*(*x∗*)

Σ

*—* **Yˆ** *∗*)(**Y***i*

*—* **Yˆ** *∗*)*⊤.*

This results in

" #

**Σˆ***∗* = 142*.*25 105*.*25 *.*

105*.*25 97*.*25

In this case, the prediction,NO COMMA **Yˆ** *∗*,NO COMMA and its covariance matrixNO COMMA **Σˆ***∗*NO COMMA correspond to **Y¯** *L* and **Σ***L* respectively,COMMA because there is only 1 tree that has been made. For CRRF, many trees are formed from multiple bootstraps because each bootstrap leads to different splits being determined to maximise the Euclidean distance. As a result, the output **Yˆ** *∗* will be adjusted when more and more trees are added.

## Code Explanation

CRRF came with a pre-set package in R called CovReg. This made its implementation simpler than manual coding in R. However, there were still some necessary considerations.

To prevent over-fitting, explicit stopping criteria were implemented in the tree growth process. A node was set as terminal if either the depth limit of the tree was reached, depth = 0, or the number of samples in the node fell below the threshold, *n <* 5. In these cases, the node retains the mean and covariance of the responses, rather than splitting further. This procedure stops the tree from becoming too large and avoids overfitting.IF ‘THESE CASES’ ARE HYPOTHETICAL, LEAVE IT PRESENT TENSE, BUT IF THEY ACTUALLY HAPPENED IN YOUR PROCESS, THEN PUT IT IN PAST TENSE LIKE THE REST OF YOUR PROCESS IN THE PARA.

Furthermore, to evaluate the performance of CRRF, a 5-fold CV was implemented using the ANRMSE. The data was randomly split into 5 folds of equal size, where each fold was used as a validation set once, and the remaining 4 folds formed the training set. This ensured that every observation was used in training and validation. At the end of the CV, the final ANRMSE of all the test folds was used to measure CRRF fit. I THINK PAST IS BETTER HERE, BUT HONESTLY I FEEL LIKE MAYBE BOTH WORK SO JUST GO WITH YOUR GUT.

This code also uses parallelisation, which speeds up training by running multiple trees concurrently rather than sequentially. Parallel computing makes programs and processes run faster as more CPUs are used. [46](#_bookmark147) They are implemented using multi-threading, allowing trees to be trained simultaneously, thereby significantly reducing computation time. AND I THINK THIS ONE WORKS IN PRESENT AS ITS NOT EXPLICTLY YOUR PROCESS YOU’RE DESCRIBING.

## Results

The CRRF model achieved an ANRMSE of 0.5238, the lowest value thus far. This model did not even consider interaction and polynomial terms, which were not explicitly tested as RFs already capture non-linear relationships and interactions through their splits.

The CRRF model was also analysed in relation to how it determined predictor feature importance in modelling the 2 responses, ROE and sustainability score.

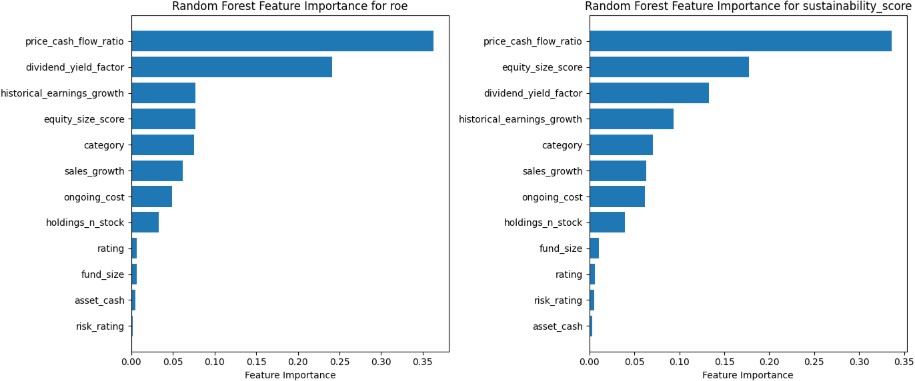


Figure 5.3: CRRF Feature Importance

For ROE, CRRF assigns the highest importance to price cash flow ratio, dividend yield factor,NO COMMA and equity size score. Therefore, these features lead to the most significant changes in the covari- ance structure of responses. In contrast, sales growth, risk rating,NO COMMA and asset cash have minimal influence, implying they do not meaningfully alter response covariances.

For sustainability score, price cash flow ratio again is the strongest predictor, reinforcing its role in separating subgroups with differing response covariances. equity size score becomes more relevant here, which implies that it determines sustainability-linked covariance changes. historical earnings growth and dividend yield factor remain relevant but with a (ID ADD ‘a’) smaller effect compared to price cash flow ratio. sales growth remains as important for sustainability score as it is for ROE,COMMA and risk rating still has minimal importance.

Overall, the consistent importance of price cash flow ratio highlights its strong role in dictating which equity funds should be invested in long-term,COMMA based on the CRRF. Meanwhile, features with minimal covariance impact, like risk rating and asset cash, are not prioritised.

Although CRRFs have well-modelled IS ‘WELL-MODELLED’ COMMONLY USED? this dataset, they have some drawbacks. First, they are computationally intensive because they need to calculate covariance matrices at each split. Second, although CV mitigates this somewhat, they are prone to over-fitting with irrelevant features. XGBoost can deal with these issues because it uses approximate tree learning and implements regularisation.

# Chapter 6 XG Boost

This chapter investigates Extreme Gradient Boosting by first explaining its principles in the single-response case, including how it improves upon traditional gradient boosting. This method is then extended to the multivariate case using the Cholesky Decomposition to decorrelate the responses prior to fitting. The resulting model is then evaluated on the equity fund dataset to assess its performance. PRESENT TENSE FOR THE CHAPTER YOU ARE IN

## Theory

Now, we move on to Extreme Gradient Boosting, which differs from Random Forests as it builds trees sequentially rather than independently. While Random Forests rely on bagging to reduce variance, XGBoost fits each new tree to the negative gradients of the loss function – DASH (BUT NO DASH IS ALSO FINE IF YOU PREFER) with respect to the current model’s predictions. This iterative process forms an ensemble, refining predictions at every step.

## Introduction

Gradient tree boosting builds a model by sequentially adding regression trees to minimise a predefined loss function. Each new tree is trained on the negative gradient of the current model, enabling the ensemble to iteratively refine its predictions. [47](#_bookmark148)

Formally, for a dataset with *n* observations and *m* features, we write:

*D* = *{*(**x***i, yi*)*},* **x***i ∈* R *, yi ∈* R*, i* = 1*, . . . , n.*

*m*

The prediction for each input, **x***i*, is the sum of *K* regression trees:

*K*

Σ

*y*ˆ*i* = *ϕ*(**x***i*) = *fk*(**x***i*)*, fk ∈ F,* (6.1)

*k*=1

where each *fk* is an individual regression tree in the ensemble,COMMA and *F* is the space of all possible regression trees. [47](#_bookmark148)

One immediate point of concern is that the estimate will keep increasing as the number of regression trees does. However, each function *fk* is not an independent prediction of *yi ,COMMA*but rather a correction term trained to minimise the loss function with respect to the ensemble’s current prediction.

For a general loss function *ℓ*(*yi, y*ˆ*i*), this correction corresponds to the negative gradient, *gi*, of the loss with respect to the current prediction:

*f* (**x** ) *≈ −∂ℓ*(*yi, y*ˆ*i*)

*k*

*i*

*∂y*ˆ

*y*ˆ =*y*ˆ(*t−*1)

*i*

= *−g .*

*i*

*i*

In the case of the squared error loss function:

*i*

1 2

*ℓ*(*yi, y*ˆ*i*) = 2 (*yi − y*ˆ*i*) *,* (6.2)

the negative gradient simplifies to the residual *yi − y*ˆ(*t−*1).

*i*

The space of all possible regression trees can be formally written as:

*F* = *f* (**x**) = *wq*(**x**) *q* : R*m → {*1*, . . . , T}, w ∈* R*T* } *,* (6.3)

where *T* is the number of leaves in the tree, *q* is a function that maps an input **x** to its corresponding leaf index, and *w ∈* R*T* is a vector of leaf weights. [47](#_bookmark148) Each weight *wj* corresponds to a real-valued score assigned to leaf *j*. Therefore, the leaf weights are the prediction scores returned by the tree. When an input **x***i* is passed through the tree, the structure *q* routes it to a specific leaf *q*(**x***i*), and the tree returns the corresponding score *wq*(**x***i*) as its prediction.

Unlike classification trees,NO COMMA which output discrete class labels at each leaf, regression trees assign

continuous numerical values, also referred to as prediction scores, to their leaves. These values are learned during training and represent the model’s output for inputs that fall into their?? respective leaf.

Building on Equations [6.1](#_bookmark83) and [6.3,](#_bookmark85) each regression tree has its own structure, *q*(*k*), and set of leaf scores, *w*(*k*). [47](#_bookmark148) Hence, the final model prediction is the sum of these scores across all trees:

*K*

*i* Σ

*y*ˆ = *w*(*k*) *.*

*q*(*k*)(**x***i*)

*k*=1

Extreme Gradient Boosting, otherwise known as XGBoost, improves gradient boosting by including regularisation explicitly in the loss function to address over-fitting and improve model predictions. Its objective function is:

*n T*

*L*(*ϕ*) = Σ *l*(*yi, y*ˆ*i*) + Σ Ω(*ft*)*,* (6.4)

*i*=1

*t*=1

where *l* represents the loss function, and Ω(*ft*) is a regularisation term. The loss function in XGBoost regression can take various forms, such as the log loss. For this report, however, the unscaled squared error loss was used due to its simplicity in gradient derivation. See Equation [6.2.](#_bookmark84) The omission of the

1 has no impact on model performance, as the constant does not affect the direction of the gradient

2

or the speed of convergence.

XGBoost uses regularisation to help control model complexity and mitigate over-fitting as follows:

1

Ω(*ft*) = *γT* + 2 *λ*

Σ*j*=1

*w*2*,*

*T*

*j*

where *T* is the number of leaves in the tree, *wj* are the leaf weights, *γ* is the pruning parameter which penalises additional leaves, and *λ* controls the regularisation of leaf weights. [47](#_bookmark148)

The pruning parameter can be selected automatically via CV or hyper-parameter tuning. However, it is ultimately set by the user. See Equation [6.5](#_bookmark86) for more on this parameter.

These weights are what XGBoost aims to determine, as they comprise the prediction values as- signed to each leaf. XGBoost computes them using the Approximate Greedy Split-Finding Algorithm, which is detailed in Subsection [6.1.2.](#_bookmark87)

XGBoost also incorporates shrinkage and column subsampling to reduce overfitting further. Shrink- age scales the leaf weights of newly added trees by a factor of (I THINK ADD ‘OF’) *η ∈* (0*,* 1], known as the learning rate. This reduces the influence of each individual tree, giving subsequent trees more room to improve the model incrementally. [47](#_bookmark148) Column subsampling, in which only a random subset of features is considered when constructing each tree, not only acts as a form of regularisation but also improves computations of the parallel algorithm. [47](#_bookmark148)

XGBoost optimises the objective function using a second-order Taylor expansion. With constant

terms omitted, the loss function at iteration *t* is approximated as:

(*t*)

*n*

*L*

*≈* Σ*i*=1

1 2

*gift*(*xi*) + 2 *hift*(*xi*)

+ Ω(*ft*)*,*

(*t−*1)

where *gi* = *∂* (*t−*1) *l*(*yi, y*ˆ(*t−*1)) and *hi* = *∂*2 *l*(*yi, y*ˆ ) are the first and second-order gradients of

*i*

*y*ˆ*i* *i*

*y*ˆ(*t−*1) *i*

the loss function, respectively. [47](#_bookmark148)

To construct decision trees, XGBoost evaluates potential binary splits using the Greedy Algorithm for Split Finding. This algorithm involves maximising a gain function:

Gain =

[47](#_bookmark148)

1. *G*2

+

*L*

2 (*GL* + *GR*)2

*—*

*G*

*R*

*— γ.*

(6.5)

1. *HL* + *λ*

*HR* + *λ HL* + *HR* + *λ*

Each candidate split divides the training data into two subsets based on a feature threshold. The instances satisfying the condition (e.g.,NO COMMA *xj ≤ s*) are assigned to the left node, *L*, while those that do not are assigned to the right node, *R*.

Here, *GL* and *HL* are the sums of first and second-order gradients for the left child (L), *GR* and *HR* are the corresponding sums for the right child (R), *λ* is a regularisation parameter, and *γ* is the pruning parameter. [47](#_bookmark148) A “child” refers to the rows selected when a decision tree splits at a node.

Equation [6.5](#_bookmark86) means that a node will only be split if the gain in the objective function exceeds the pruning parameter, *γ*. Higher values of *γ* encourage shallower trees by penalising complexity.

Each component within the Gain’s bracket is called the (similarity) score. So, the Gain can be

written as:

where:

1

Gain = 2 [*SL* + *SR − SP* ] *− γ,*

*G*2 *G*2

(*GL* + *GR*)2

*SL* = *L , SR* = *R , SP* = *.*

*HL* + *λ*

*HR* + *λ*

*HL* + *HR* + *λ*

Note: *SP* is the score for (ID USE ‘FOR’) the parent node of the new node being derived.

## Iterative Steps of XGBoost

As outlined in Chen and Guestrin (2016), the exact Greedy Algorithm occurs by iterating over each feature dimension,COMMA *(USE COMMA IF k REFERS TO ‘FEATURE DIMENSIONS’) k*, ranging from 1 to *m*, where *m* is the total number of predictors.

For each feature, potential split points are evaluated by sorting the training instances based on their corresponding feature values. During this process, the cumulative sums of gradients and Hessians for the left child node, denoted *GL* and *HL*, are initialised to zero and updated at each step as:

*GL ← GL* + *gj, HL ← HL* + *hj.*

The corresponding values for the right child are computed as *GR* = *G − GL* and *HR* = *H − HL*, where

*G* and *H* are the total sums over the parent node.

The quality of each split is evaluated using the gain, which measures the reduction in loss. At each step, the algorithm compares the gain of the current split to the best one found so far:

Gainnew = *SL* + *SR − SP ,*

and retains the maximum.

Note: the pruning parameter *γ* is applied after split selection, not during the gain evaluation.

After evaluating all candidate splits, the algorithm selects the one with the highest gain. If no split yields a positive gain, the node remains a leaf. This approach ensures that each decision tree

grows in a way that minimises residual error at each step, leading to more accurate predictions.

The exact Greedy Algorithm is ideal for XGBoost, but due to its computational cost, an ap- proximate greedy algorithm is used in practice. This method is similar to the prior one, COMMA but with some adjustments,NO COMMA which are explained below, again from Chen and Guestrin (2016).

The approximate greedy algorithm begins by iterating over each feature dimension *k* = 1*, . . . , m*, where *m* is the total number of features.

For each feature *k*, a set of candidate splits *Sk* = *{sk*1*, sk*2*, . . . , skl}* is proposed by dividing the feature values into percentiles. This ensures the split proposals are distributed evenly across the feature range, capturing key points that are likely to lead to huge loss reduction. Split proposals can be generated globally or locally: in the global setting, split points are selected once per tree and reused at each node, while in the local setting, new split points are computed independently for each node.

After generating the candidate splits, the algorithm evaluates their quality. For each proposed split *sk,v* in feature *k*, the gradients and Hessians of instances falling between two consecutive split points are accumulated. The gradients and Hessians are then computed as:



where *Gkv* denotes the sum of gradients,COMMA and *Hkv* the sum of Hessians over the interval between *sk,v−*1 and *sk,v*. This process approximates the exact split-finding procedure by evaluating only a limited subset of potential split points.

Once the gradients and Hessians have been computed for all proposed splits, the algorithm selects the split that maximises the gain. This follows the same logic as the exact greedy method but limits evaluation to the proposed split points, significantly reducing computational cost.

The same procedure described earlier is then applied,COMMA but restricted to the proposed splits. While this approximation sacrifices some precision, it greatly improves efficiency, allowing XGBoost to scale to large datasets.

After identifying the optimal split, the corresponding leaf weight is computed as:

*Gj*

*wj* = *−H* + *λ,*

*j*

where *Gj* is the sum of gradients for all instances in the leaf, *Hj* is the sum of Hessians for all instances in the leaf, and *λ* is the regularisation parameter. [47](#_bookmark148)

Each instance’s prediction is updated as follows:

*y*ˆ(*t*) = *y*ˆ(*t−*1) + *ηwj,*

where *y*ˆ(*t−*1) represents the previous prediction, *wj* is the weight of the leaf the instance falls into, and

*η* is the learning rate – DASH which controls the step size of updates. [47](#_bookmark148)

## Cholesky Decomposition

Although a prior distribution is not required to apply Cholesky decomposition, assuming a multivariate normal distribution (MVN) helps satisfy its requirements, which are a positive definite symmetric matrix. The probability density function of an MVN distribution is given by:

(2*π*)*D|*Σ

*|*

2

**Y**

**Y**

*f* (**Y***|****µ***

1

*,* Σ ) = √

**Y**

**Y**

exp *−* 1 (**Y** *−* ***µ***

**Y**

)*⊤*Σ*−*1(**Y** *−* ***µ***

) *,*

**Y**

where ***µ*Y** *∈* R*D* represents a vector of conditional means, Σ**Y** is the positive definite symmetric *D×D*

response covariance matrix, *D* is the number of responses, and *| · |* denotes the determinant. [48](#_bookmark149)

In the bivariate case (*D* = 2), the local response covariance matrix for instance *i*, denoted **Σ***i***y**, is:

**Σ***i***y** =

2

*i,*1

" *σ*

1. *ρi*(**x**)*σi,*1(**x**)*σi,*2(**x**)# *,*

*ρi*(**x**)*σi,*1(**x**)*σi,*2(**x**) *σ*2

*i,*2

(**x**)

where,COMMA for each *i* = 1*, . . . , N* .*ρi,*COMMA (**x**) is the conditional correlation between the two responses, the diagonal entries represent the variances, and the off-diagonal entries represent the covariances. [48](#_bookmark149)

To ensure positive definiteness of the response covariance matrix, **ΣY**, a common and efficient approach is to use the Cholesky decomposition, which factorises the matrix as follows:

**ΣY** = **LL***⊤,*

where **L** *∈* R*D×D* is a lower triangular matrix. [48](#_bookmark149) This decomposition guarantees positive definiteness as long as all diagonal elements,COMMA *ℓii* of **L,COMMA** are strictly positive. The *D*(*D −* 1)*/*2 off-diagonal elements,COMMA *ℓij* (for *j < i*),COMMA may take any value in R.

The Cholesky decomposition uses the following algorithm to derive **L**:

* 1. Initialise **L** as an *n × n* zero matrix.
  2. For each row *i* (from 1 to *n*):
     + Compute the diagonal element *Lii*:

*Lii* =

,u,

*Aii −*

*i−*1

2

Σ

*L*

*ik*

*k*=1

* + - For each column *j* (below diagonal, *j > i*):

*i−*1

*Aji −* Σ *Ljk Lik*

*Lji* = *k*=1

*Lii*

* 1. Return **L**.

Cholesky decomposition can be used to remove intercorrelation between response variables by apply- ing the transformation **L***−*1**Y***⊤*, which produces decorrelated responses. [49](#_bookmark150) This decorrelation allows XGBoost to be applied independently to each transformed response. After prediction, the original correlated response space is recovered by reapplying **L**, which reintroduces the covariance structure.

There does exist a version of XGBoost, Multi-Output XGBoost, which supports MRR with an in-built mechanism. However, its methodology still needs to be fleshed out.

## Application

* + 1. **Small-Scale Example**

We will work with the familiar dataset evaluating Mathematics and Science Scores:

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*1: Math Scores | *Y*2: Science Scores |
| 5 | 2 | 78 | 80 |
| 7 | 3 | 85 | 79 |
| 8 | 4 | 88 | 88 |
| 3 | 1 | 65 | 70 |
| 10 | 5 | 92 | 74 |

Table 6.1: Study Time vs. Exam Scores

We have:

**X** =

5 7 8 3 10 *⊤*

2 3 4 1 5

" #

*,* **Y** =

78 85 88 65 92 *⊤*

*,*

" #

80 79 88 70 74

**Y¯** =

h81*.*6 78*.*2i *.*

First, we need the response covariance matrix, **ΣY**:

*⊤*

**Σ** = 1 (**Y** *−* **Y**¯ )*⊤*(**Y** *−* **Y**¯ ) = "112*.*30 37*.*85# *,*

**Y**

*n −* 1

37*.*85 46*.*20

So,COMMA each response value has its column mean subtracted from it. The matrix is symmetric and positive definite because its (1,2) and (2,1) entries are the same, and it has positive eigenvalues.

Next comes the Cholesky Decomposition of the response covariance matrix. This subsection is for clarity, so for ease of decomposition,NO COMMA let us show Cholesky Decomposition on a “nicer” matrix:

" #

**A** = 4 2 *.*

2 3

We aim to find a lower triangular matrix **L** such that:

"

#

**A** = **LL***⊤,* **L** = *L*11 0

*L*21 *L*22

The first step is to compute *L*11 using the formula for diagonal elements:

*Lii* =

,u,

*Aii −*

*i−*1 *k*=1

2 *⇒ L*11 =

*L*

*ik*

√*A*11 =

*√*4 = 2 *⇒* **L** =

2 0

"

#

*L*21 *L*22

Next, compute *L*21. For off-diagonal elements:

Σ

*i−*1

*Aji −* Σ *Ljk Lik*

*Lji* = *k*=1 *⇒ L*21 =

*Lii*

*A*21 *L*11

2

= 2 = 1 *⇒* **L** =

2 0

1 *L*22

"

#

Finally, compute *L*22, going back to our diagonal element formula:

*Lii* =

,u,

*Aii −*

*i−*1

*L*2

Σ

*⇒ L*22 =

q*A*22 *− L*2

= *√*3 *−* 1 =

*√*2 *⇒* **L** =

"2 0 #

*ik* 21

*k*=1 1 2

*√*

See for verification that this decomposition is correct.

[A.2.7](#_bookmark177)

Applying this algorithm to **ΣY,** when passing it through R’s inbuilt chol function, gives??:

**Σ** = **LL***⊤* = "10*.*60 0 # "10*.*60 3*.*57#

**Y**

3*.*57 5*.*78 0 5*.*78

Now,COMMA we can decorrelate the responses using **L***−*1:

" #

**Y**˜ *⊤* = **L***−*1**Y***⊤* = 7*.*36 8*.*02 8*.*30 6*.*13 8*.*68

9*.*29 8*.*71 10*.*09 8*.*32 7*.*43

Here,COMMA the off-diagonal elements of the covariance matrix of the transformed responses are zero, and it leaves the identity matrix, **I**2:

Cov(**Y**˜ ) *≈* "1 0#

0 1

Now that we have decorrelated the responses, we can apply an XGBoost model to each response without needing to consider the response covariance matrix.

Let us apply XGBoost, with a squared error loss function, to response *Y*˜1, Mathematics Scores

against the predictors *X*1 and *X*2. XGBoost starts by initialising predictions for all samples. The initial prediction is typically 0.5, but let us start here instead by taking the mean of *Y*˜1:

*Y*ˆ (0) = 7*.*36 + 8*.*02 + 8*.*30 + 6*.*13 + 8*.*68 = 38*.*49 = 7*.*70*.* (6.6)

1 5

Thus, the initial prediction for all samples is:

*Y*ˆ (0) = h

1

7*.*70 7*.*70 7*.*70 7*.*70 7*.*70

5

i*⊤ .*

Next, the residuals are calculated as the difference between the actual values of?? *Y*˜1 and the initial pre- dictions (to 2dp):

*r*1 = *Y*˜1

*— Y*ˆ (0) = h

*⊤*

i *−* h

7*.*36 8*.*02 8*.*30 6*.*13 8*.*68

*⊤*

7*.*70 7*.*70 7*.*70 7*.*70 7*.*70

i

= h*−*

1

0*.*34 0*.*32 0*.*60

*⊤*

*−*1*.*57 0*.*98

i *.*

The residuals are the errors in the initial predictions, which will be used to fit the first XGBoost tree. They are taken because these THE WORD THESE DOESN’T REALLY WORK HERE I DON’T THINK are half the gradients when using the squared error loss function. The Gradients, *gi*, and Hessians, *hi*, are needed because they are critical in evaluating the Score and computing the Gain. The gradients are calculated as:

*g* = *∂L* = *∂* Σ(*Y*ˆ

*n*

*i*

*∂Y*ˆ*i*

*∂Y*ˆ*i i*=1

*i*

*— Y* ) = 2(*Y*

2

*i*

*— Y*ˆ ) = 2*r ,*

where the summation vanishes because we are looking at each *i*th element when differentiating. From the residuals, the gradients are (to 2dp):

*i*

*i*

*i*

*g* = h*−*0*.*68 0*.*64 1*.*21 *−*3*.*13 1*.*96i*⊤ .*

For squared error loss, the second-order gradient (Hessian) is:

*∂*2*L*

*hi* = *∂Y*ˆ 2 = 2*.*

*i*

Since we are using the squared error loss function, the Hessian for all points is:

*h* = h *⊤*

i

2 2 2 2 2

Now, we train a small decision tree to predict the gradients. This starts with the parent node, *P* , which contains all the data. Each tree starts with a single leaf, and all residuals go to that leaf. [50](#_bookmark151) All that needs to be done here is to compute its ID SPECIFY WHAT ‘IT’ IS REFERRING TO HERE OUT OF LEAF OR TREE score using the exact residuals:

(Σ *gP* )2

*P*

*SP* = Σ *h*

= 0*.*

+ *λ*

where *gp, hp* are the Gradients and Hessians of the parent node, respectively. This makes sense because, as a result of this loss function, the gradients are proportional to the residuals, which always sum to zero. When the gradients in a node are very different, they cancel each other out,COMMA and the Score is relatively small,COMMA and you expect large gradients when initialising.THIS PREVIOUS SENTENCE NEEDS BREAKING UP SOMEHOW, I’VE DONE ONE SUGGESTION[50](#_bookmark151) In contrast, when the gradients are similar, or there is just one of them, they do not cancel out,NO COMMA and the Score is relatively large.[50](#_bookmark151)

XGBoost will then evaluate splits after this AFTER WHAT EXACTLY? by comparing the Gain for each split in *X*1 and *X*2 in this dataset,COMMA and finding the split which maximises said Gain. Maximising the Gain means there is a better splitting of the gradients into clusters of similar values. [50](#_bookmark151) Iterating through all potential split points can be computationally expensive, so this is where a weighted quantile sketch approximates the

[A.2.7](#_bookmark176)

best-split points using weighted quantiles. See the number of candidate split points.

for more. This speeds up training by reducing

Now, let us show how the gain is computed for an example split: *X*1 *<* 7. IS THIS SUPPOSED TO BE FULL STOP OR COMMA*X*1 *<* 7 splits the data into 2 sets of predictors and corresponding transformed responses:

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*˜1: Mathematics Scores | *r*0: Residuals |
| 5 | 2 | 7.36 | -0.34 |
| 3 | 1 | 6.13 | -1.57 |

Table 6.2: Study Time vs. Mathematics Scores for *X*1 *<* 7

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*˜1: Mathematics Scores | *r*0: Residuals |
| 7 | 3 | 8.02 | 0.32 |
| 8 | 4 | 8.3 | 0.60 |
| 10 | 5 | 8.68 | 0.98 |

Table 6.3: Study Time vs. Mathematics Scores for *X*1 *≥* 7

Science Scores, *Y*ˆ2, have been removed as we are just modelling *Y*˜1 here, and the initial residuals are placed for clarity when calculating the gradients and Hessians.

[6.3](#_bookmark93)

Let us call [6.2](#_bookmark92) the “left” (*L*) node and cumulative gradients and Hessians as:

the “right” (*R*) node. From each node, calculate AGAIN, ACTIVE COMMAND IS FINE, BUT JUST DECIDE IF YOU PREFER IT TO PASSIVE, SO ‘IS CALCULATED’… the

2 2 2 3

*GL* = Σ 2(*ri*)*L* = *−*3*.*81*, HL* = Σ 2 = 4*, GR* = Σ 2(*ri*)*R* = 3*.*81*, HR* = Σ 2 = 6*.*

*i*=1

*i*=1

*i*=1

*i*=1

This is used to calculate the similarity scores for the left and right nodes with *λ* = 0*.*1:

*G*2 *G*2

*SL* = *L* = 3*.*62*, SR* = *R* = 2*.*42*.*

*HL* + *λ HR* + *λ*

Now,COMMA compute the Gain for *X*1 *<* 7 in comparison with the parent node:

Gain = *SL* + *SR − SP* = 6*.*04*.*

For simplicity, assume the tree split of *X*1 *<* 7 gives the optimal gain so we can adjust the predictions. The leaf weight differs by the child node that? each row falls in,COMMA and is calculated as follows (to 2dp):

*w* = *GL* = 0*.*95*, w*

*—*

*L HL* + *λ R*

= *GR* = 0*.*63

*HR* + *λ*

*− −*

The predictions are then adjusted using the leaf weight with the learning rate *η*, which we set as 0.3. Here, the initial prediction was the mean, from [6.6.](#_bookmark91) So, the final transformed predictions are:

*Y*ˆ (1) = *Y*ˆ (0) + *η × wL* = 7*.*98*, Y*ˆ (1) = *Y*ˆ (0) + *η × wR* = 7*.*51*.*

1*,L*

1

1*,R*

1

These need to be converted back to the original response space with the original correlation structure, using: **L***Y*ˆ (1) or **L***Y*ˆ (1). This gives the following original predictions (to 2dp):

1*,L*

1*,R*

Left Node : *Y*ˆ (1) = 84*.*55*,* Right Node : *Y*ˆ (1) = 79*.*61*.*

1 1

## Code Explanation

This section describes how Cholesky-Gaussian XGBoost was applied to the equity fund dataset. The primary implementation detail (which has not been covered thus far) involved incorporating *k*-fold CV.

The dataset was partitioned into five folds using KFold(), with shuffling enabled to ensure random- ness in CV. For each fold, the empirical response covariance matrix of the target variables was com- puted using numpy.cov() to capture interdependencies between responses. Cholesky decomposition was then applied via numpy.linalg.cholesky(), producing a lower triangular matrix **L**. Its inverse, **L***−*1, was computed using numpy.linalg.inv() to decorrelate the response variables prior to model training. Each decorrelated response variable was then modelled independently using an XGBoost regression model that was? implemented with (OR COULD YOU SAY using?) XGBRegressor(). The models were trained on the transformed responses using model.fit(X train, Y train), and predictions were generated on the validation set using model.predict(X val). As the predictions were produced in the transformed space, they were mapped back to the original scale by multiplying by (I KNOW IT REPEATS ‘by’ BUT WORTH IT FOR CLARITY I THINK) the Cholesky factor via numpy.dot(). This step ensured that the predicted responses retained their original correlation structure. The adjusted predictions were subsequently used to compute the ANRMSE across all folds.

## Results

XGBoost achieved an ANRMSE of 0.426, indicating an excellent fit. This was the best performance among all the models tested and highlights XGBoost’s effectiveness in modelling complex datasets.

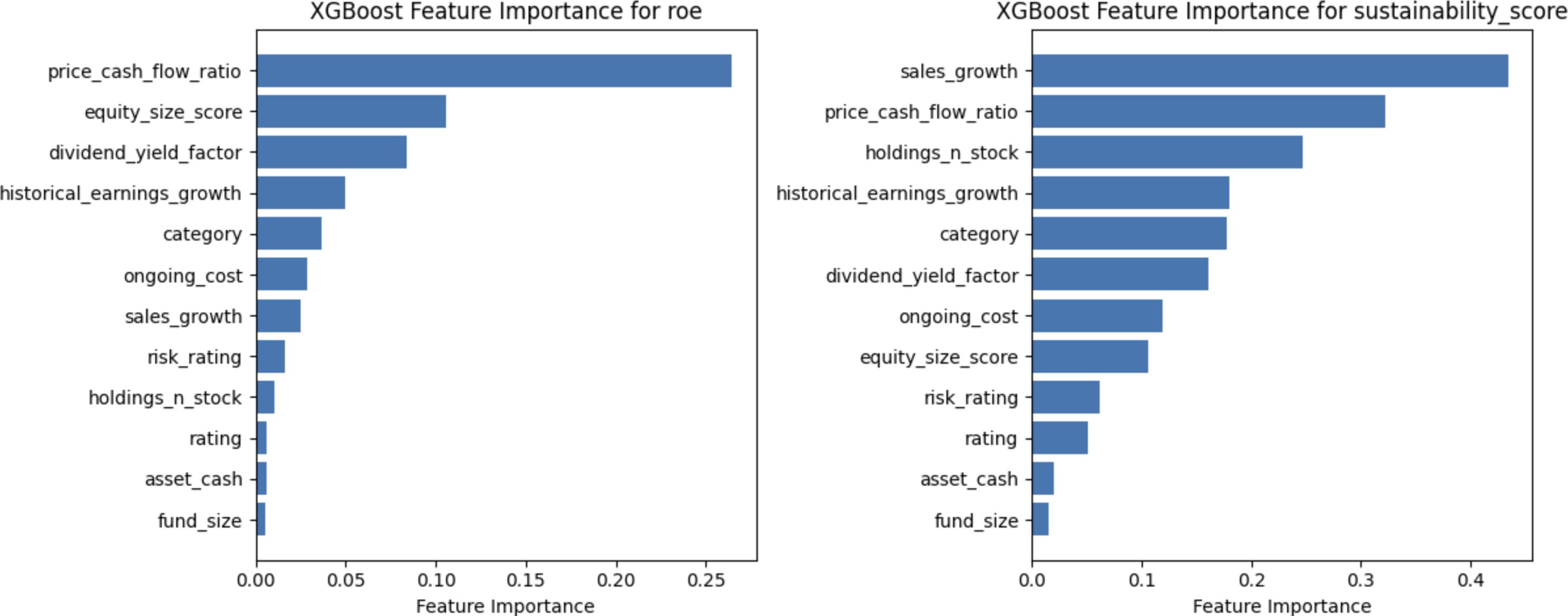


Figure 6.1: XGBoost Feature Importance

The most important feature in explaining the ROE was the price cash flow ratio. This means that, following decorrelation, this ratio had the greatest impact in terms of accounting for variation in the response. However, due to the Cholesky transformation, some of its influence may have been partially transferred to the sustainability score when mapping back to the original response space.

Other notable contributors included the equity size score and the dividend yield factor, re- flecting their substantial role in predicting the decorrelated target. On the other hand, variables such as asset cash and fund size had relatively low feature importance, indicating that these features pro- vided little additional information once dependencies between responses had been taken into account. The most impactful feature for the sustainability score was sales growth. Unlike ROE, where a dominant predictor emerged, sustainability appeared to be influenced by a few predictors, as indicated by the more even distribution of feature importance. The price cash flow ratio also ranked highly, similar to its importance in predicting ROE. This supports the fact/idea/hypothesis/notion etc. (CHOOSE WHICHEVER WORKS BETTER) that prior to decorrelation, ROE and the

sustainability score shared some correlation, which was separated using Cholesky decomposition.

Other significant features for the sustainability score included holdings n stock and historical earnings growth, reinforcing the fact/idea etc. (AGAIN CHOOSE WHICH WORKS BETTER) that sustainability was influenced by multiple predictors rather than a single dominant one.

XGBoost proved highly effective on the equity fund dataset, but it has limitations in certain cases. One notable drawback is that its interpretation can be challenging due to its inherent complexity. [51](#_bookmark152)

# Chapter 7 Conclusion

## Summary of Findings and Model Comparisons

* + 1. **Summary of Findings**

Here is a table summarising the key figures generated from this report from the model methods:ITS NOT DEEP BUT IF YOU HAVE TWO COLONS TOGETHER THIS CLOSE IN YOUR PDF FORMAT, THEN COULD CHANGE THE SECOND ONE TO A DASH, SO ‘TABLE 7.1 - …’ Table 7.1: Model Fit Comparison on Equity Fund Dataset

|  |  |  |
| --- | --- | --- |
| **Model** | **ANRMSE** | **Model Fit** |
| **Chapter 3: Linear Regression** | | |
| Model 1: MRLR with all predictors | 0.7606 | Unsatisfactory |
| Model 2: Forward Selection | 0.7606 | Unsatisfactory |
| Model 3: Backward Selection | 0.7924 | Unsatisfactory |
| Model 4: Bidirectional Selection | 0.7606 | Unsatisfactory |
| Model 5: Bidirectional Selection with Interaction Terms | 0.5613 | Good |
| Model 6: Bidirectional Selection with Non-Linear Terms | 0.6893 | Satisfactory |
| **Chapter 4: Shrinkage Methods** | | |
| Model 7: MRCE | 0.7612 | Unsatisfactory |
| Model 8: RRRR | 0.7510 | Unsatisfactory |
| Model 9: MRCE including Polynomial Terms | 0.6781 | Satisfactory |
| Model 10: RRRR including Polynomial Terms | 0.7276 | Unsatisfactory |
| Model 11: MRCE including Interaction Terms | 0.6777 | Satisfactory |
| Model 12: RRRR including Interaction Terms | 0.7231 | Unsatisfactory |
| **Chapter 5: Random Forests** | | |
| Model 13: Covariance Regression with Random Forests | 0.5238 | Good |
| **Chapter 6: XGBoost** | | |
| Model 14: Cholesky-Decomposition XGBoost | 0.4267 | Excellent |

The model fit was defined based on criteria set out in Chapter 3 (see subsection [3.2.1).](#_bookmark35)

## Model Comparisons

The MRLR model using all predictors performed poorly, and all the stepwise selection methods (for- wards, backwards and bi-directional) failed to improve performance significantly. Backward selection performed the worst, suggesting that eliminating variables here led to worse model performance.

However, the introduction of interaction terms, alongside bi-directional stepwise selection, led to a huge improvement which resulted in a “Good” model fit. Similarly, the addition of non-linear terms, also selected through bi-directional stepwise selection, improved upon the MRLR model (with just the standard predictors), and achieved a “Satisfactory” fit. I ADDED THE BRACKETS AS THOSE TWO SENTENCES BACK-TO-BACK ARE QUITE LONG SO HAVE TRIED TO REMOVE A CLAUSE HERE AND THERE TO MAKE THE PARA MORE READABLE

The shrinkage methods aimed to improve regression stability by using penalisation techniques. However, the MRCE and RRRR models, with the regular features, performed similarly to their coun- terparts in standard regression, again returning an “Unsatisfactory” fit.

Introducing polynomial terms in MRCE improved performance, but the same approach for RRRR was less effective. Adding interaction terms improved MRCE but did not improve RRRR as much.

The performance of shrinkage methods was also noteworthy. This is because, despite being specifically de- signed to handle multiple responses, particularly under multicollinearity or high-dimensional settings, both MRCE and RRRR performed as poorly as, or even worse than, the MRLR models. AGAIN THE PREVIOUS SENTENCE WAS VERY LONG, SO THIS IS ONE WAY OF SHORTENING IT.This can be attributed to the structure of the dataset: after pre-processing (see Chapter [2),](#_bookmark11) the predictors were not highly correlated, and the data was not high-dimensional (*p* = 14*, n* = 1269). In this case, the advantages of regularisation and rank reduction are less impactful.

Tree-based models significantly improved upon linear and shrinkage-based methods. The Random Forest model was significantly better than previous models and was evaluated as a “Good” fit model. Progress was also found with XGBoost, which was the first “Excellent” model.

The strong performance of XGBoost and CRRF, particularly THIS SENTENCE IS TOO LONG AS WELL I THINK, THIS IS ONE WAY TO SHORTEN IT BUT IF IT MAKES LESS SENSE ID STILL FIND ANOTHER WAY, AS IN THE OTHER ONESin comparison to the limited improvement achieved by including interaction terms in regression models, indicates that the equity fund re- sponses, sustainability score and ROE, are driven by subtle, conditional interactions among predictors. The only marginal multivariate normality, evidenced by Mardia’s test (see Subsection [2.3.5),](#_bookmark20) further corroborates the existence of non-linear effects and differential impacts. Overall, the findings suggest that the most effective modelling approaches are those that accommodate flexible, local structure and interaction effects, as opposed to approaches based on rigid parametric or distributional assumptions. Ultimately, the results suggest that the predictors influence fund outcomes in ways that are not purely additive or globally consistent. Instead, relationships appear to be context-dependent. This means that the effect of one variable likely changes based on the values of others. This aligns with the success of models like XGBoost and CRRF, which are designed to capture these conditional structures.

Conversely, the poor performance of linear and shrinkage-based models reflects their inability to adapt to such nuanced, data-specific patterns.

## Challenges and Limitations

This report faced different methodological and practical challenges. The first of which was finding an appropriate dataset to evaluate these models. This was a challenge because it involved finding a dataset that satisfied all the assumptions underpinning each MRR model. In particular, priority was given to datasets with independent observations, multivariate normality between responses, full rank in the predictor matrix, and a moderate degree of correlation between response variables. These criteria ensured that the models could be appropriately applied and compared.

In retrospect, several limitations of the equity fund dataset itself became clearer through the modelling process. Although the dataset had a reasonably large number of observations, the number of predictors, *p* = 14, limited the potential benefits of the shrinkage methods used (MRCE and RRRR). Missing data presented another challenge. In the report, I used? linear model imputation for numerical

variables and random forests for the categorical ones. While these approaches are more sophisticated than simple imputation, they assume that data are missing at random, that is, the missingness depends only on observed values. This assumption may not hold, particularly if some variables are missing non-randomly and influenced by unobserved or sensitive information.

In retrospect, a formal assessment of the missingness mechanism should? have been carried out. For example, Hotelling’s multivariate *t*-test or absolute standardised mean differences could have been used to test if the distributions of observed covariates differed between missing and non-missing groups.

Ultimately, more advanced imputation techniques, such as multiple imputation via chained equa- tions, provide an improved method of dealing with missing data.

Another limitation in this report was the relatively low degree of feature engineering. For example, frequency encoding was used for the equity category feature, which treated each fund category as a separate level. However, many of these categories could have been effectively grouped — for instance, by region (e.g., Sweden, Switzerland), sector (e.g., Consumer Goods), or market capitalisation (e.g., Large-Cap). Grouping them based on this basis might have improved performance in every model. While some models incorporated interaction terms and polynomial features, a more systematic feature engineering process, such as using principal component analysis, could have improved model performance or revealed complex patterns. With more time and hence familiarity with the financial

context of the dataset, this could have been explored further.

Moving beyond the dataset, another challenge was determining what constitutes a valid MRR model. Initially, models were included that, while multivariate in their structure, did not strictly qualify as multiple-response. For example, multivariate ridge regression (see subsection [4.1.3)](#_bookmark51) applies a penalty across the full coefficient matrix, regularising all responses jointly, but it does not explicitly incorporate the response covariance matrix. Therefore, it is not a proper MRR model.

Finally, one of the most significant limitations was the interpretability of more complex models, particularly those in later chapters. While highly accurate, models such as XGBoost are often regarded as “black boxes”,COMMA OUTSIDE QUOTES as their internal decision structures are not easily interpretable. [52](#_bookmark153) However, this lack of interpretability does not diminish their utility: the ability to predict accurately can still inform decision-making,COMMA even if the model’s internal mechanisms are not clear.

## Report Overview and Future Work

This report began by extending SRLR to MRLR. Upon examining MRLR,? it was then combined with stepwise selection and sequential MANOVA to identify relevant predictors. The analysis then progressed to shrinkage-based methods, namely MRCE and RRRR, both of which introduced regularisation to improve model stability. Following this, tree-based approaches were explored through CRRFs, which aimed to maximise Euclidean distance between nodes during splits. Finally, XGBoost was adapted to multiple responses via the Cholesky-Gaussian decomposition, allowing for the decorrelation of response variables and the use of independent models on each response.

While this report has extensively covered several MRR models on a single dataset, there remain multiple avenues for future work. One direction would be to apply these models across a broad set of datasets to understand how performance varies under different conditions. For example, this might include varying levels of multicollinearity, response correlation,NO COMMA and sample size. This could help to further clarify which types of data are best suited for each model,COMMA and why certain models perform better in specific scenarios.

Alternatively, future work could involve expanding the set of models applied to this dataset using Gaussian Processes. For example, neural networks can be extended to MRR, and hence evaluated on this dataset. Ultimately, this could improve both predictions and insight into the underlying structure of the equity fund data.

# Bibliography

1. James Chen. Investing in equity funds: A beginner’s guide. *Investopedia*, April 2024. Reviewed by Gordon Scott, Fact checked by Ariel Courage.
2. B.R. Kumar. Sustainable finance and investment: The intersection of profitability and environ- mental impact. *Library Progress (International)*, 44:16477–16485, 10 2024.
3. Jason Fernando. Return on equity (roe) calculation and what it means. *Investopedia. URL:* [*https://www.*](http://www/) *investopedia. com/terms/r/returnonequity. asp (date of access: 14.12. 2023)*, 2023.
4. Will Kenton. Net income (ni): Definition, uses, and formula, 2024. Investopedia. Updated June 25, 2024. Reviewed by Khadija Khartit and fact-checked by Kirsten Rohrs Schmitt.
5. Mitchell Franklin, Patty Graybeal, and Dixon Cooper. *Principles of accounting. Volume 1: Fi- nancial accounting*. OpenStax, 2019.
6. James Chen. Morningstar sustainability rating: Definition and how it works, 2023. Investopedia. Updated November 30, 2023. Reviewed by Gordon Scott.
7. Larry Fink. Larry fink’s 2021 letter to ceos.

[https://www.blackrock.com/corporate/about-us/](https://www.blackrock.com/corporate/about-us/sustainability-resilience-research)

[sustainability-resilience-research](https://www.blackrock.com/corporate/about-us/sustainability-resilience-research), January 2021. Chairman and CEO of BlackRock.

1. Kanti V Mardia, John T Kent, and Charles C Taylor. *Multivariate analysis*. John Wiley & Sons, 2024.
2. Stockopedia. Risk rating, 2024. Accessed: 27 November 2024.
3. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statis- tical Learning with Applications in R*. Springer, 2 edition, 2021.
4. AMG Funds LLC. Equity style analysis: Growth vs. value, 2023. Accessed: 27 November 2024.
5. Adam Hayes. Growth stock: What it is, examples, vs. value stock, December 2023. Updated 11 December 2023. Reviewed by Chip Stapleton. Fact checked by Kirsten Rohrs Schmitt. Accessed: 27 November 2024.
6. Hitul Adatiya. Eda on european mf dataset, 2022. Version 2 of 4. Accessed: 27 November 2024. Licensed under Apache 2.0 open source.
7. Chris B. Murphy. Operating costs definition: Formula, types, and real-world examples, 2024. Updated 28 June 2024. Reviewed by David Kindness. Fact checked by Amanda Jackson. Accessed: 27 November 2024. Part of the series: The Evolution of Accounting and its Terminology.
8. Michael H. Kutner, Christopher J. Nachtsheim, John Neter, and William Li. *Applied Linear Statistical Models*. McGraw-Hill/Irwin, 5 edition, 2005.
9. David A. Belsley, Edwin Kuh, and Roy E. Welsch. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley, 1980.
10. Shangzhi Hong and Henry S Lynn. Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction. *BMC medical research methodology*, 20:1–12, 2020.
11. C´atia M. Salgado, Carlos Azevedo, Hugo Proen¸ca, and Susana M. Vieira. Missing data. In Ross C. Mitchell, editor, *Secondary Analysis of Electronic Health Records*, pages 143–162. Springer, Cham, 2016. Open Access chapter distributed under the terms of the Creative Commons Attribution- NonCommercial 4.0 International License.
12. openfunds. Fund ratios and exposures, 2020. Accessed: 2025-04-03.
13. Richard Johnson and Dean Wichern. Multivariate linear regression models: Section 7.7. In *Applied Multivariate Statistical Analysis: Pearson New International Edition*, pages 360–429. Pearson Education, Limited, 6th edition, 2013. Accessed: 27 November 2024.
14. Smith Gary. Step from stepwise. *Journal of Big Data*, 5(1):1–12, 2018.
15. Gudmund R. Iversen. Multivariate analysis of variance and covariance (manova and mancova). In Michael S. Lewis-Beck, Alan Bryman, and Tim Futing Liao, editors, *The SAGE Encyclopedia of Social Science Research Methods*, volume 2, pages 702–703. SAGE Publications, Inc., Thousand Oaks, CA, 2004.
16. Newsom. Multivariate analysis of variance. *Psy 522/622 Multiple Regression and Multivariate Quantitative Methods*, 2024. Winter 2024 Lecture Notes.
17. STAT 505 Pennsylvania State University. Lesson 8: Multivariate analysis of variance (manova), 2024.
18. Daniel N Moriasi, Jeffrey G Arnold, Michael W Van Liew, Ronald L Bingner, R Daren Harmel, and Tamie L Veith. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3):885–900, 2007.
19. Kristin L. Sainani. Multivariate regression: The pitfalls of automated variable selection. *PM&R*, 5(9):791–794, 2013.
20. Ashin Mukherjee and Ji Zhu. Reduced rank ridge regression and its kernel extensions. *Statistical analysis and data mining: the ASA data science journal*, 4(6):612–622, 2011.
21. Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.
22. Adam J Rothman, Elizaveta Levina, and Ji Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962, 2010.
23. Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 02 2011.
24. Frederik Questier, Raf Put, Danny Coomans, Beata Walczak, and Yvan Vander Heyden. The use of cart and multivariate regression trees for supervised and unsupervised feature selection. *Chemometrics and Intelligent Laboratory Systems*, 76(1):45–54, 2005.
25. Elsevier Inc. *Data Science Process*. Elsevier, Amsterdam, Netherlands, 2019. DOI: https://doi.org/10.1016/B978-0-12-814761-0.00002-2.
26. Leo Breiman. Statistics department university of california berkeley, ca 94720. 2001. *Random Forests*, 2001.
27. Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.
28. Quebec Centre for Biodiversity Science. Qcbs r workshop series: Workshop 10: Advanced multi- variate analyses in r, 2023. Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Inter- national Licenses all content.
29. G. De’ath. Multivariate regression trees: A new technique for modeling species-environment relationships. *Ecological Society of America*, 88:2783–2792, 2002.
30. Mark Segal and Yuanyuan Xiao. Multivariate random forests. *WIREs Data Mining and Knowledge Discovery*, 1(1):80–87, 2011.
31. Cansu Alakus, Denis Larocque, and Aur´elie Labbe. Covariance regression with random forests.

*BMC bioinformatics*, 24(1):258, 2023.

1. L Breiman, JH Friedman, RA Olshen, and CJ Stone. Classification and regression trees. boca raton, florida: Chapman hall/crc, 1984.
2. Hoora Moradian, Denis Larocque, and Fran¸cois Bellavance. L 1 l 1 splitting rules in survival forests. *Lifetime data analysis*, 23:671–691, 2017.
3. Sami Tabib and Denis Larocque. Non-parametric individual treatment effect estimation for sur- vival data with random forests. *Bioinformatics*, 36(2):629–636, 2020.
4. Cansu Alaku¸s, Denis Larocque, S´ebastien Jacquemont, Fanny Barlaam, Charles-Olivier Martin, Kristian Agbogba, Sarah Lipp´e, and Aur´elie Labbe. Conditional canonical correlation estimation based on covariates with random forests. *Bioinformatics*, 37(17):2714–2721, 2021.
5. Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *Project Euclid*, 2019.
6. Benjamin Lu and Johanna Hardin. A unified framework for random forest prediction error esti- mation. *Journal of Machine Learning Research*, 22(8):1–41, 2021.
7. Cansu Alakus, Denis Larocque, and Aurelie Labbe. Rfpredinterval: An r package for prediction intervals with random forests and boosted forests. *arXiv preprint arXiv:2106.08217*, 2021.
8. N Azizah, LS Riza, and Y Wihardi. Implementation of random forest algorithm with parallel com- puting in r. In *Journal of Physics: Conference Series*, volume 1280, page 022028. IOP Publishing, 2019.
9. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
10. Alexander M¨arz. Multi-target xgboostlss regression. *arXiv preprint arXiv:2210.06831*, 2022.
11. DSG Pollock and Emi Mise. The cholesky decomposition of a toeplitz matrix and a wiener- kolmogorov filter for seasonal adjustment. Technical report, 2020.
12. StatQuest with Josh Starmer. XGBoost Part 1 (of 4): Regression, 2019. [Online; accessed 6- March-2025].
13. XGBoosting. Xgboost advantages and disadvantages (pros vs cons), 2024. Accessed: 2025-04-11.
14. Jo Jackson. Multivariate techniques: Advantages and disadvantages.

[https://classroom.](https://classroom.synonym.com/multivariate-techniques-advantages-and-disadvantages-123456.html)

,

[synonym.com/multivariate-techniques-advantages-and-disadvantages-123456.html](https://classroom.synonym.com/multivariate-techniques-advantages-and-disadvantages-123456.html)

2018. Updated September 05, 2018. Accessed December 06, 2024.

1. Alexander Von Eye and G Anne Bogat. Testing the assumption of multivariate normality. *Psy- chology Science*, 46:243–258, 2004.

# Chapter A Appendices

## Additional Data Visualisations

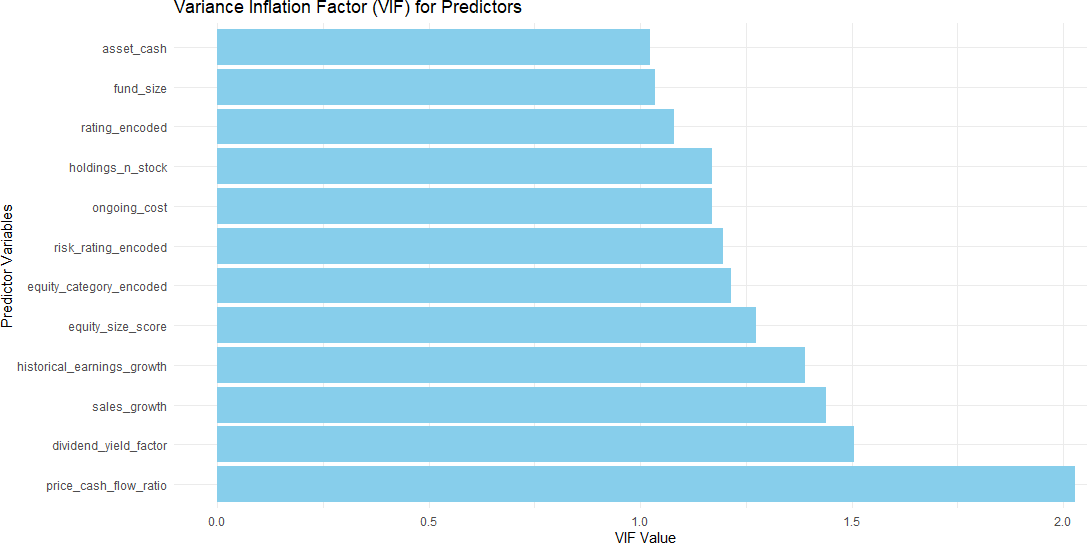


Figure A.1: VIF Values post Dataset Cleaning

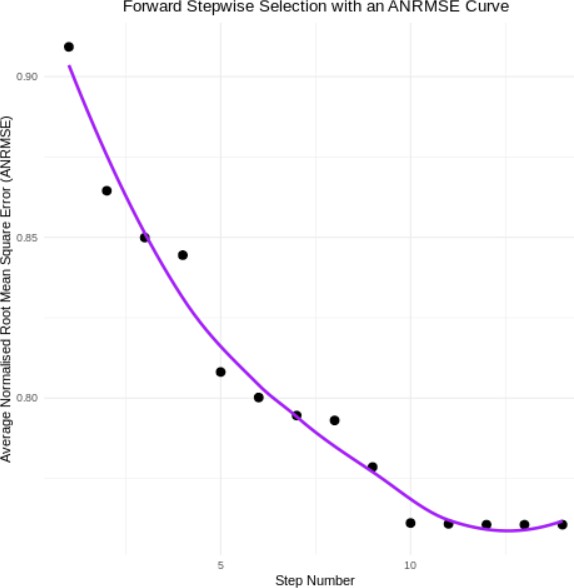


Figure A.2: Forward Stepwise Selection with an ANRMSE Curve

## Extra Chapter Insights

* + 1. **Introduction**

#### Correlation Between Math Scores and Science Scores

|  |  |  |  |
| --- | --- | --- | --- |
| *X*1: Hours Studied | *X*2: Time Spent on Papers | *Y*1: Math Scores | *Y*2: Science Scores |
| 5 | 2 | 78 | 80 |
| 7 | 3 | 85 | 79 |
| 8 | 4 | 88 | 88 |
| 3 | 1 | 65 | 70 |
| 10 | 5 | 92 | 74 |

Table A.1: Study Time vs. Exam Scores Let each pair (*xi, yi*) represent Math (*x*) and Science (*y*) scores:

(*x*1*, y*1) = (78*,* 80)*,* (*x*2*, y*2) = (85*,* 79)*,* (*x*3*, y*3) = (88*,* 88)*,* (*x*4*, y*4) = (65*,* 70)*,* (*x*5*, y*5) = (92*,* 74)*.*

Next, compute the means. Let *x*¯ be the mean of the Math scores, and *y*¯ be the mean of the Science scores:

*x*¯ =

78 + 85 + 88 + 65 + 92

5

= 81*.*6*,*

*y*¯ =

80 + 79 + 88 + 70 + 74

5

= 78*.*2*.*

For each pair (*xi, yi*), compute *xi − x*¯ and *yi − y*¯ . For clarity, the table below shows these values:

*xi yi xi − x*¯ *yi − y*¯

|  |  |  |  |
| --- | --- | --- | --- |
| 78 | 80 | 78 *−* 81*.*6 = *−*3*.*6 | 80 *−* 78*.*2 = 1*.*8 |
| 85 | 79 | 85 *−* 81*.*6 = 3*.*4 | 79 *−* 78*.*2 = 0*.*8 |
| 88 | 88 | 88 *−* 81*.*6 = 6*.*4 | 88 *−* 78*.*2 = 9*.*8 |
| 65 | 70 | 65 *−* 81*.*6 = *−*16*.*6 | 70 *−* 78*.*2 = *−*8*.*2 |

92 74 92 *−* 81*.*6 = 10*.*4 74 *−* 78*.*2 = *−*4*.*2

Next, multiply each pair of deviations and sum them:

5

Σ

(*xi − x*¯)(*yi − y*¯) = (*−*3*.*6)(1*.*8) + (3*.*4)(0*.*8) + (6*.*4)(9*.*8) + (*−*16*.*6)(*−*8*.*2) + (10*.*4)(*−*4*.*2) = 151*.*4*.*

*i*=1

Next, compute the Sum of Squares for Each Variable:

5

Σ

(*xi − x*¯)2 = (*−*3*.*6)2 + (3*.*4)2 + (6*.*4)2 + (*−*16*.*6)2 + (10*.*4)2 = 449*.*2*,*

*i*=1

5

Σ

(*yi − y*¯)2 = (1*.*8)2 + (0*.*8)2 + (9*.*8)2 + (*−*8*.*2)2 + (*−*4*.*2)2 = 184*.*8*.*

*i*=1

Finally, apply the Pearson Correlation Formula. The Pearson correlation coefficient *r* is given by

(*xi x*¯)(*yi y*¯)

*r* = √ΣΣ *− −*

(*xi − x*¯)2 Σ(*yi − y*¯)2

151*.*4

=

*√*

449*.*2 *×* 184*.*8

*≈* 0*.*526*.*

Thus, the correlation between the Math and Science scores in this dataset is approximately 0*.*526, indicating a moderate positive relationship.

## Exploratory Data Analysis

#### Mardia’s Test for Multivariate Normality

Mardia’s test evaluates whether a set of multivariate data follows a multivariate normal distribution by examining two key properties: multivariate skewness and multivariate kurtosis. [53](#_bookmark154)

The multivariate skewness statistic is defined as:

1 *N*

Σ

*N*

Σ

*β*1*,d* = *N* 2

h(*xi − x*¯)*⊤S−*1(*xj − x*¯)i *,*

*i*=1 *j*=1

3

where *xi* is the observation vector for the *i*-th observation, *x*¯ is the sample mean vector, *S* is the sample covariance matrix, *N* is the number of observations and *d* is the number of response variables. [53](#_bookmark154)

Under the null hypothesis of multivariate normality, the skewness statistic *β*1*,d* is asymptotically distributed as chi-squared with *d*(*d*+1)(*d*+2) degrees of freedom. [53](#_bookmark154) The p-value is calculated as:

6

*p* = *P* *χ*2 *> β*1*,d* *,*

df

which corresponds to the upper-tail probability of the chi-squared distribution.

The multivariate kurtosis statistic is defined as:

2

*β*2*,d* =

1

*N*

*N*

Σ

*i*=1

h(*xi − x*¯)*⊤S−*1(*xi − x*¯)i *,*

For a multivariate normal distribution, the expected value of *β*2*,d* is *d*(*d* + 2), and the variance is approximately:

Var(*β*

2*,d*

) = 8*d*(*d* + 2) *.* [53](#_bookmark154)

*N*

To assess deviation from normality, the kurtosis statistic is standardised:

*β*2*,d − d*(*d* + 2)

*Z* = √Var(*β*

*.*

2*,d*)

The p-value is then computed from the standard normal distribution as: *p* = 2*P* (*Z > |z|*)*.* This two-tailed test detects whether the kurtosis significantly deviates from the expected value under mul- tivariate normality.

For the equity fund dataset, the sample covariance matrix is the response covariance matrix be- tween ROE and sustainability score. *d* = 2 because there are 2 responses and *N* = 1248, which is the number of equity funds in the dataset.

## Multiple Response Linear Regression

#### Covariance Expansion

Cov(**XB**) = E[(**XB** *−* E[**XB**])(**XB** *−* E[**XB**])*⊤*]*.*

Since expectation is linear: E[**XB**] = E[**X**]**B**, the centered term becomes:

(**XB** *−* E[**X**]**B**) = (**X** *−* E[**X**])**B***.*

Substitute this into the covariance definition:

Cov(**XB**) = E h(**X** *−* E[**X**])**B** *·* (**X** *−* E[**X**])*⊤***B***⊤*i *.*

Factor out **B** because it is constant:

Cov(**XB**) = **B***⊤*E h(**X** *−* E[**X**])(**X** *−* E[**X**])*⊤*i **B***.*

The middle term is just Cov(**X**), therefore: Cov(**XB**) = **B***⊤*Cov(**X**)**B***.*

## Ridge Regression

#### Closed-Form Coefficient Estimator Derivation

We begin with the ridge regression loss function:

2 2

*,*

*L*(***β***) = **Y** *−* **X*β*** + *λ* ***β***

2

2

where **X** *∈* R*n×p*, **Y** *∈* R*n×*1, and ***β*** *∈* R*p×*1.

To find the closed-form solution, we differentiate *L*(***β***) with respect to ***β*** and set the gradient equal to zero giving:

*∇****β****L*(***β***) = *−*2**X**

*⊤*

(**Y** *−* **X*β***) + 2*λ****β*** = 0*,*

**X***⊤***Y** = **X***⊤***X*β*** + *λ****β****,*

(**X***⊤***X** + *λ***I**)***β*** = **X***⊤***Y***.*

Hence, the closed-form solution for ridge regression is:

***β***ˆRidge = (**X***⊤***X** + *λ***I**)*−*1**X***⊤***Y***.*

* + 1. **Reduced Rank Ridge Regression** **Minimisation Problem Simplification** Starting with the penalised objective:

*.*

*F*

 **Y** *−* **XB **

*F*

2 + *λ * **B ** 2

Because, the above is the sum of two Frobenius norms, this can be written as a single Frobenius norm of a block matrix as follows:

¨ **Y** *−* **XB**!¨2

Now note that:

= ¨ *−√λ***B** ¨ *.*

**Y** *−√* **XB**! = **Y**! *−* **X**

*F*

! **B***.*

*— λ***B 0** *√λ***I**

Therefore, the penalised problem can be written as an equivalent least squares problem on an aug- mented dataset:

 **Y** *−* **XB ** 2 + *λ * **B ** 2 = ¨ **Y**! *−* **X B**¨ = **Y***∗ −* **X***∗***B ** 2 *.*

*F*

*F*

2

*F*

!

*√*

¨ **0** *λ***I** ¨*F*

Hence, the original penalised objective can be rewritten as the following unconstrained least squares objective on an augmented dataset:

*F*

**B**ˆ (*λ, r*) = arg min

rank(**B**)*≤r*

 **Y***∗ −* **X***∗***B ** 2 *.*

#### Mean Centring and the Response-Covariance Matrix

The response matrix **Y** can be written as the sum of its mean-centred component and the mean

matrix: **Y** = (**Y** *−* **Y**¯ ) + **Y**¯ *.* In this decomposition, **Y** *−* **Y**¯ is a matrix where each row is the mean response vector **y**¯.

is the mean-centred version of **Y**, while **Y**¯

Substituting the decomposition **Y** = (**Y** *−* **Y**¯ ) + **Y**¯ Expanding this product results in

into **Y***⊤***Y**, gives [(**Y** *−* **Y**¯ ) + **Y**¯ ]*⊤*[(**Y** *−* **Y**¯ ) + **Y**¯ ]*.*

**Y***⊤***Y** = (**Y** *−* **Y**¯ )*⊤*(**Y** *−* **Y**¯ ) + **Y**¯ *⊤*(**Y** *−* **Y**¯ ) + (**Y** *−* **Y**¯ )*⊤***Y**¯ + **Y**¯ *⊤***Y**¯ *.*

The two cross terms **Y**¯ *⊤*(**Y** *−* **Y**¯ ) and (**Y** *−* **Y**¯ )*⊤***Y**¯ are equal to zero. This vanishing comes from

the fact that the mean response matrix **Y**¯ is constant across all rows, meaning that the sum of all

deviations from the mean is equal to zero. Since:

*n*

Σ

(**y***i −* **y**¯) = 0*,*

*i*=1

it follows that when multiplied by **Y**¯ , the result also vanishes:

*n n*

Σ **y**¯*⊤*(**y***i −* **y**¯) = **y**¯*⊤* Σ(**y***i −* **y**¯) = 0*.* [28](#_bookmark129)

*i*=1

*i*=1

Since this holds for all response variables, it means that **Y**¯ *⊤*(**Y** *−* **Y**¯ ) = 0 and (**Y** *−* **Y**¯ )*⊤***Y**¯ Substituting these results back into the expansion simplifies the equation to:

**Y***⊤***Y** = (**Y** *−* **Y**¯ )*⊤*(**Y** *−* **Y**¯ ) + **Y**¯ *⊤***Y**¯ *.*

= 0*.*

The term **Y**¯ *⊤***Y**¯

can be further simplified. Since each row of **Y**¯

is equal to the mean response vector

**y**¯, and there are *n* such rows, the sum simplifies as follows:

**Y**¯ *⊤***Y**¯

= *n***y**¯*⊤***y**¯*.*

This scaling by *n* occurs because the same mean vector is repeated for all observations. Substituting this result into the equation gives:

**Y***⊤***Y** = (**Y** *−* **Y**¯ )*⊤*(**Y** *−* **Y**¯ ) + *n***Y**¯ *⊤***Y**¯ *.* [28](#_bookmark129)

This equation shows that **Y***⊤***Y** consists of the variance of the mean-centred data plus the contribution

of the mean structure. In its final form, this is:

**Y***⊤***Y** = (*n −* 1)**ΣY** + *n***Y¯** *⊤***Y¯** *,*

where **Y¯** *⊤***Y¯**

captures the mean effects of **Y**.

#### Projection Simplification

Starting with:

*τ i*=1

*di***u***i***v***⊤*

*i*

*r*

*j*=1

! ,Σ

**v***j***v***⊤* =

Σ*i*=1

*τ*

*di***u***i* ,**v***⊤*

Σ*j*=1

*r*

**v***j***v***⊤* *.*

The inner expression simplifies using orthonormality:

Σ

*j*

*i*

*j*

* If *i ≤ r*:

**v***⊤* Σ **v***j***v***⊤* = **v***⊤.*

*r*

* If *i > r*:

*i j* *i*

*j*=1

*r*

**v***⊤* Σ **v***j***v***⊤* = **0***⊤.*

*i*

*j*

*j*=1

So the entire sum becomes:

*r*

Σ *di***u***i***v** *.*

*⊤*

*i*

*i*=1

#### Singular Value Decomposition

Given the matrix **Yˆ** *∗* :

**R**

*−*5*.*67 2*.*96 4*.*19 *−*14*.*29 12*.*81 0*.*74 *−*0*.*62 *⊤*

**Y**

" #

=

**ˆ** *∗*

**R**

*.*

*−*1*.*82 *−*0*.*91 2*.*28 *−*2*.*73 3*.*19 *−*0*.*23 0*.*55

We aim to compute its Singular Value Decomposition: matrix:

**Y**

**R**

**ˆ** *∗* = **UDV***⊤.* First, compute the Gram

**Y**ˆ *∗ ⊤***Y**ˆ *∗*

*R*

*R*

= "427*.*7629 96*.*57640# *.*

96*.*57640 27*.*33693

This symmetric matrix is used to compute the eigenvalues and eigenvectors, which in turn provide the singular values. These singular values are the square roots of the eigenvalues of **Y**ˆ *∗ ⊤***Y**ˆ *∗* . The

eigenvalues of **Y**ˆ *∗ ⊤***Y**ˆ *∗*

*R R*

are: *λ*1 = 449*.*838519 and *λ*2 = 5*.*261272.

Hence:

*R R*

**D** = "*√*449*.*838519 *√* 0

0

5*.*261272

# = "21*.*209397 0 # *.*

0 2*.*293746

The eigenvectors of **Y**ˆ *∗ ⊤***Y**ˆ *∗* are the matrix **V**:

*R R*

" #

**V** = 0*.*9748562 *−*0*.*2228349 *.*

0*.*2228349 0*.*9748562

Now, we compute **U** using the formula: **U** = **Yˆ** *∗* **VS***−***1**. Since **V** is an orthonormal matrix, we have

**R**

**V***⊤* = **V***−*1, leading to (2dp):

0*.*28 0*.*13 0*.*22 0*.*69 0*.*62 0*.*03 0*.*02 *⊤*

"*− − −* #

**U** = *.*

*−*0*.*22 *−*0*.*68 0*.*56 0*.*23 0*.*11 *−*0*.*17 0*.*29

## Multivariate Regression with Covariance Estimation

#### Negative Log-Likelihood Derivation

Starting with the familiar MRLR model:

**Y**(*n×m*) = **X**(*n×p*)**B**(*p×m*) + **E**(*n×m*)*,*

where *p* no longer includes the column of ones.

Each row **y***i ∈* R1*×m* of **Y** follows:

**y***i|***x***i ∼ Nm*(**x***i***B***,* **ΣE**)*,*

The probability density function for a single observation is:

*⊤*

*p*(**y**

*i*

*|* **x** ) = 1 exp *−* 1 (**y**

**E**

*—* **x B**) **Σ**

*−*1(**y**

*—* **x B**) *.*

*i*

*i*

Assuming independence across observations, the total log-likelihood becomes:

*i*

(2*π*)*m/*2*|***Σ**

*|*1*/*2

2

*i*

*i*

**E**

*n*

Σ

log *L*(**B***,* **ΣE**) = log *p*(**y***i |* **x***i*)

*i*=1

*nm n*

= *−* log(2*π*) *−* log *|***Σ**

2

2

*| −* 1 Σ(**y**

*i*=1

*—* **x B**) **Σ**

*⊤*

*i*

*i*

*−*1(**y**

*—* **x B**) (\*)

*i*

*i*

Subbing the residual matrix, **E** = **Y** *−* **XB**, into the quadratic form gives:

*n*

**E**

2

**E**

Σ*i*=1

*n*

(**y***i −* **x***i***B**)*⊤***ΣE***−*1(**y***i −* **x***i***B**) = tr

**E***⊤***EΣE***−*1

where **Ω** = **ΣE***−*1.

= tr (**Y** *−* **XB**)*⊤*(**Y** *−* **XB**)**Ω** *,*

Substituting back into (\*), the log-likelihood becomes:

log *L*(**B***,* **Ω**) = *−nm* log(2*π*) + *n* log *|***Ω***| −* 1 tr (**Y** *−* **XB**)*⊤*(**Y** *−* **XB**)**Ω** *.*

2

2

2

Ignoring the constant term, *nm* log(2*π*), the negative log-likelihood (NLL) becomes: NLL(**B***,* **Ω**) = 1 tr (**Y** *−* **XB**)*⊤*(**Y** *−* **XB**)**Ω** *− n* log *|***Ω***|.*

2

2

2

Finally, multiplying and dividing the trace term by *n*, and omitting the constant 1 , we obtain the

2

simplified negative log-likelihood:

*g*(**B***,* **Ω**) = tr 1 (**Y** *−* **XB**)*⊤*(**Y** *−* **XB**)**Ω** *−* log *|***Ω***|.*

*n*

This is the negative log-likelihood function (up to a constant) for the MRLR model.

## Cholesky-Gaussian XGBoost

#### Weighted Quantile Sketches

In the approximate split-finding algorithm, as already stated, candidate split points are chosen based on percentiles of the feature values. The rank function *rk*(*z*) calculates the proportion of instances where feature *k* is less than *z*, weighted by second-order gradient statistics *h*. [47](#_bookmark148) The algorithm aims to find split points *sk*1*, sk*2*, . . . , skl* such that the difference between the ranks of consecutive points is small, controlled by *ϵ*, which is an approximation factor. [47](#_bookmark148) This process ensures that the split points are well-distributed across the range of the feature values.

The Hessian, *hi*, is used as a weight for each instance or row in the dataset, meaning that data points with larger Hessians carry more influence in determining the split. This weighting reflects the importance of the instance in minimising loss. Higher-weighted instances show where the model struggles, guiding the algorithm to focus on these critical areas during split finding.

Traditional quantile sketch algorithms work for datasets where all instances have equal weights.[47](#_bookmark148) However, no existing algorithm efficiently handles weighted data. This creates challenges in approxi- mating split points for datasets with imbalanced or sparse data distributions. Randomised sorting or heuristic approaches have been used to address this issue, but these methods lack theoretical guar- antees. [47](#_bookmark148) As a result, they may fail to identify the optimal split points, leading to suboptimal model performance. To overcome this limitation, XGBoost introduces a novel weighted quantile sketch algo- rithm that can handle weighted data with provable theoretical guarantees. [47](#_bookmark148) This algorithm ensures accurate split findings by accounting for the distribution of weights across the dataset.

#### Cholesky-Decomposition Verification

We verify by computing:

**LL***⊤* = "2 0# "2 1#

1 *√*2

0 *√*2

As desired.

(2 2) + (0 0) (2 1) + (0 *√*2)

(1 *×* 2) + (*√*2 *×* 0) (1 *×* 1) + (*√*2 *× √*2)

" #

= " *× × × × * #

= 4 2 = **A**

2 3