Stocks vs. Bonds

A Data-Driven Approach to Asset Allocation Using Machine Learning

Master's thesis in computer science and engineering

Emil Hölvold
Nermin Skenderovic

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Stocks vs. Bonds

A Data-Driven Approach to Asset Allocation Using Machine Learning

Emil Hölvold
Nermin Skenderovic

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Stocks vs. Bonds
A Data-Driven Approach to Asset Allocation Using Machine Learning
Emil Hölvold
Nermin Skenderovic

Supervisor: Sebastianus Cornelis Jacobus Bruinsma, Department of Data Science and AI
Advisor: Karl Larsson, Nordea Bank Abp
Advisor: Fredrik Lundström, Nordea Bank Abp
Examiner: Stefan Lemurell, Department of Mathematical Sciences

Master's Thesis 2023
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The cumulative return of the stock index MSCI World and the bond index Bloomberg Global Aggregate between 2001 and 2023.

Typeset in LaTeX
Gothenburg, Sweden 2023

Stocks vs. Bonds
A Data-Driven Approach to Asset Allocation Using Machine Learning
Emil Hölvold
Nermin Skenderovic
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

This thesis explores the application of supervised machine learning algorithms to asset allocation strategies with the aim of enhancing investment decision-making processes. Collaborating with Nordea, one of the leading financial institutions in the Nordics, the study was conducted at their Asset & Wealth Management department to investigate the potential of developing a machine learning model, with the goal of improving portfolio performance and reducing risk in the context of a dynamic and uncertain financial market environment.

The research begins by analysing the current status of the field, examining the theoretical foundations of asset allocation, and identifying the shortcomings of traditional approaches. Additionally, the thesis raises a nuanced view of quantitative investing, with an in-depth exposition of the most common pitfalls and their consequences.

Building on this foundation and previous work, regression and classification algorithms are investigated together with premium financial data as potential solutions to overcome these limitations. Specifically, the Random Forest and XGBoost models are used to forecast movements for the upcoming month in a global stock and bond index. The signals generated by the models are then incorporated into a rule-based allocation model.

The findings of this research suggest that machine learning techniques can offer valuable insights and improved performance in asset allocation. The results highlight the potential of these models to identify leading indicators and exploit market inefficiencies, resulting in improved risk-adjusted returns. The best-performing model achieved an alpha of 2.05% during the backtest between 2020 and 2023, accompanied by an increase in Sharpe ratio and a decrease in volatility.

However, it is important to note that the effectiveness of machine learning algorithms is heavily dependent on the quality and availability of data, as well as the appropriate selection and calibration of model parameters. Financial markets are dynamic and subject to various factors, so ongoing adjustments are necessary to adapt to changing market conditions and mitigate risks.

Keywords: asset allocation, machine learning, quantitative investing, regression, classification, algorithms, Random Forest, XGBoost, leading indicators, risk-adjusted returns.

Acknowledgements

We would like to express our sincere gratitude and appreciation to the following individuals who have made invaluable contributions to the completion of this master's thesis:

First and foremost, we would like to extend our deepest thanks to our supervisors at Nordea, Karl Larsson and Fredrik Lundström. Their guidance, expertise, and continuous support throughout this project have been instrumental in its success. Their insightful feedback, constructive criticism, and relentless pursuit of improvement have profoundly influenced our understanding, guided us through the challenges we encountered, and propelled this project beyond what we could have achieved otherwise.

Furthermore, we acknowledge the guidance and assistance provided by Sebastianus Cornelis Jacobus Bruinsma, our supervisor at Chalmers. His tireless dedication to supporting us and willingness to allocate time for discussions when we encountered academic-related questions have been truly invaluable. His insightful feedback and suggestions have significantly enhanced the quality of this report.

We also express our gratitude to Antti Saari, the Nordea manager, for providing us with the opportunity to undertake this thesis within the organisation. The resources provided by Nordea have made all the difference between success and failure in this research.

We are also thankful for our examiners, thanking Johan Jonasson for believing in this project from the first day and for his work reviewing this report. The constructive feedback and valuable insights have helped refine our research. We are also thankful to Stefan Lemurell who could step in during the final part of the project and for being helpful in taking this project to completion.

Finally, we express our heartfelt appreciation to all those who have supported us during the course of this project, including our friends and family, whose unwavering encouragement and belief in our abilities have been a constant source of motivation.

Emil Hölvold, Nermin Skenderovic, Gothenburg, 2023-06-19

Glossary

Alpha: A term used to describe an investment strategy's ability to beat the market, or the relative performance to a benchmark.

Asset: An asset is anything of value that has the potential to generate future economic benefits, including stocks and bonds.

Asset allocation: "Asset allocation is an investment strategy that aims to balance risk and reward by apportioning a portfolio's assets according to an individual's goals, risk tolerance, and investment horizon." [1]

Basis point: Equivalent to 0.01%. The smallest measure used in quoting yields and interest rates.

Bias (prediction bias): An error that occurs due to erroneous assumptions in the learning algorithm.

Bond: A depth instrument issued by a government, municipality, or corporation to raise capital, where the issuer pays interest to the bondholder over a specified period.

Boosting: A technique in machine learning that iteratively combines multiple weak classifiers into a single strong classifier by applying weights to misclassified samples.

Classification: A supervised learning task where the goal is to assign input data to a specific category or class. It involves learning a mapping function from input features to discrete output labels.

Data leakage: when future information is used to predict past events.

Feature: A feature is an input variable or attribute that is used to make predictions or model a target variable. It is also sometimes called an independent variable.

Fixed-target portfolio: A portfolio that is rebalanced to keep the proportion between assets fixed.

Label: The correct answer or result for a given data point.

Overfitting: When a model is created that matches the training data too closely, resulting in a model that fails to make correct predictions on new data.

Portfolio: Refers to a collection of assets, such as stocks and bonds held by an individual or entity to generate income and/or achieve long-term financial goals.

Regression: A supervised learning task focusing on predicting continuous numerical values rather than discrete classes. It involves learning a mapping function from input features to a continuous output variable.

Stock: A unit of equity ownership in the capital stock of a corporation

Strong learner: A model capable of achieving arbitrarily good accuracy by learning complex patterns and relationships in the data.

Supervised learning: A type of training method in which the model is trained on predetermined labels.

Target variable: Also known as the dependent variable, is the variable that is being predicted or modelled by the machine learning algorithm. It is the output variable or the response variable.

Weak learner: A simple model that gives better results than a random prediction in a classification problem or the mean in a regression problem

Contents

Contents xi

List of Figures xv

List of Tables xvii

1 Introduction 1
- 1.1 Background . . . 1
  - 1.1.1 Asset Allocation . . . 3
  - 1.1.2 Nordea . . . 4
- 1.2 Aim . . . 4
- 1.3 Goals . . . 4
  - 1.3.1 Portfolio Allocation Targets . . . 5
- 1.4 Challenges . . . 6
- 1.5 Limitations . . . 6
- 1.6 Literature . . . 7
  - 1.6.1 Previous Work . . . 7
  - 1.6.2 Machine Learning in Finance . . . 7

2 Data 9
- 2.1 Description . . . 9
- 2.2 Preprocessing & Cleaning . . . 9
- 2.3 Feature Engineering . . . 10
  - 2.3.1 Value . . . 11
    - 2.3.1.1 Value Difference . . . 11
  - 2.3.2 Returns . . . 11
  - 2.3.3 Volatility . . . 11
  - 2.3.4 Exponential Moving Average (EMA) . . . 11
  - 2.3.5 Sharpe Ratio . . . 12
    - 2.3.5.1 Sharpe Ratio Difference . . . 12
  - 2.3.6 Sortino Ratio . . . 12
    - 2.3.6.1 Sortino Ratio Difference . . . 12
  - 2.3.7 Maximum Drawdown (MDD) . . . 13
  - 2.3.8 Correlation . . . 13
  - 2.3.9 First Derivative . . . 13
  - 2.3.10 Second Derivative . . . 14

2.3.11 Z-Score . . . 14
2.3.11.1 Z-Score Difference . . . 14
2.4 Analysis . . . 14

3 Machine Learning Fundamentals 17
3.1 Models . . . 17
3.1.1 Decision Trees . . . 18
3.1.2 Random Forest . . . 18
3.1.3 Gradient Boosting . . . 20
3.1.4 eXtreme Gradient Boosting (XGBoost) . . . 22
3.2 Model Evaluation Strategies . . . 22
3.2.1 Train-Validation-Test Split . . . 22
3.2.2 Cross-Validation . . . 23
3.3 Hyperparameter Tuning . . . 24
3.3.1 Model-Free Blackbox Tuning Methods . . . 25
3.4 Backward Feature Elimination . . . 26
3.5 Performance Metrics . . . 26
3.5.1 Coefficient of Determination (R^2) . . . 26
3.5.2 Accuracy . . . 27

4 Methodology 29
4.1 Development Approach . . . 29
4.2 Model Input Processing . . . 29
4.2.1 Merging . . . 29
4.2.2 Filtering . . . 30
4.2.3 Filling Missing Data . . . 30
4.2.4 Shifting independent variables . . . 30
4.2.5 Train-Validation-Test Split . . . 31
4.2.6 Numpy Transformation . . . 32
4.3 Hyperparameter Tuning . . . 32
4.3.1 Random Forest . . . 32
4.3.2 XGBoost . . . 32
4.4 Feature Elimination . . . 33
4.5 Model Evaluation . . . 33
4.5.1 Prediction Models . . . 35
4.5.2 Allocation Models . . . 35
4.6 Model Comparison . . . 36

5 Results 37
5.1 Prediction Models . . . 37
5.1.1 Prediction Regressors . . . 37
5.1.2 Prediction Classifiers . . . 38
5.2 Allocation Models . . . 40
5.2.1 Allocation Regressors . . . 41
5.2.2 Allocation Classifiers . . . 41
5.3 Rebalancing day . . . 42
5.4 Feature Importance . . . 42

6 Conclusion **45**
6.1 Discussion of Results 45
6.1.1 Prediction Models 45
6.1.1.1 Prediction Regressors 45
6.1.1.2 Prediction Classifiers 46
6.1.2 Allocation Models 46
6.1.2.1 Allocation Regressors 46
6.1.2.2 Allocation Classifiers 47
6.1.3 Feature Importance 47
6.1.4 Conclusions 48
6.2 Future Work 49
6.3 Social and Ethical Aspects 50

Bibliography **51**

A Appendix 1 **I**
A.1 Time Series I

List of Figures

1.1 The annual returns of the two indices studied in this project, with the four most exceptional years labelled in the graph. 2002 was a year when the economy was still struggling to recover from the recession that had started in 2001. The uncertainty in the economy led investors to seek the relative safety of bonds, driving up bond prices and depressing stock prices. The following year, 2003, the economy was recovering, contributing to the rise of stocks and bonds. In 2008, the world went into a severe worldwide crisis caused by the United States housing bubble bursting. As stocks plunged, investors shifted their money into lower-risk government bonds, increasing their price. In 2021 global bonds slumped into their first bear market in a generation, under pressure from central bankers determined to quash inflation caused by two years of expansionary fiscal policy during the COVID-19 pandemic. 2

2.1 MSCI World (Year-Over-Year) vs. US ISM PMI index, with a correlation of 0.739. 15

3.1 Example of how a regression tree can be used to fit a model to continuous non-linear data. Each leaf of the tree is labelled with a value, which is the output of the model. 18

3.2 The Random Forest will build **N** unique decision trees that will each make a prediction. Random Forest is a strong learner constructed of many smaller decision trees, known as weak learners. 19

3.3 Gradient Boosting learning curve. Figure illustraded by Aratrika Pal [39]. 21

3.4 When developing a model for time series forecasting, it is important to remove future data and gap samples from the training set to avoid data leakage. 23

3.5 When developing a model for time series forecasting, it is important to remove future data and gap samples from the training set to avoid data leakage. 24

3.6 Comparison of grid search and random search minimising a function with one important and one unimportant parameter. However, when both parameters have a large impact on the result, grid search usually performs better. This figure is based on the illustration by Bergsta and Bengio [44]. 25

4.1 Shifting the independent (X) features. 30
4.2 Train-Validation-Test Split. The models are trained on the training set, hyperparameter tuning is performed using the validation set, and the models' generalisation capabilities are tested on the test set. . . . 31
4.3 Feature importance from the initial Random Forest model trained on over 1200 features. However, due to the presence of noise, the number of features used in the final models were significantly reduced by up to 90%. 34
4.4 The backtested performance of MSCI Europe between 1995 and 2014 is almost twice as good when only considering the investment universe of 2014, a common error in quantitative investing. Graph produced by Yin Luo [20]. 34

5.1 Prediction results for the regressor models on the test set. 38
5.2 Prediction results for the classifier models on the test set. 39
5.3 Allocation results for the final models. 40
5.4 Predicted Risk-Adjusted Returns (PRED_RaR) from the XGBoost regressor and realised Risk-Adjusted Returns (RaR) from the Stock and Bond index. PRED_RaR has been a core part of the regressor version of the rule-based allocation model. 41
5.5 Rebalancing day matters. The total return of the Random Forest and XGBoost allocation models can vary by 8.15 and 11.51 percentage points, respectively, depending on when the portfolio is rebalanced. . 42
5.6 Both the XGBoost and Random Forest Stock classifiers were trained on the same 150 features. The Bond classifiers, however, reached optimal performance on slightly different features, hence Figure 5.6b contains an additional 40 features compared to Figure 5.6a. 43

6.1 Optimal portfolio (with actual instead of predicted returns) for the Risk-Adjusted Returns (RaR) based portfolio strategy. 47

List of Tables

3.1 Random Forest Parameters. These hyperparameters were optimised for both the regressor and the classifier. 20
3.2 XGBoost Parameters. These hyperparameters were optimised for both the regressor and the classifier. 22

4.1 Random Forest default vs. best parameters. See Table 3.1 for parameter definitions. 32
4.2 XGBoost default vs. best parameters. See Table 3.2 for parameter definitions. 33

5.1 Prediction performance (R^2-Score) of the regressor models for the Stock and Bond index (best values in bold). 38
5.2 Prediction performance (Accuracy) of the classifier models for the Stock and Bond index (best values in bold). 39
5.3 Allocation performance of the studied models (best values in bold). . 40
5.4 The Stock index's top ten features, combined from the two classifier models used to forecast the direction of the monthly returns. 43
5.5 The Bond index's top ten features, combined from the two classifier models used to forecast the direction of the monthly returns. 44

A.1 This table presents the time series data used, including Stock and Bond indices, risk-premium data, interest rates, sentiment surveys, and indicators on business, geopolitical and financial conditions. The data provides a comprehensive view of the financial spectrum covering various aspects such as market performance, risk assessment, and macroeconomic indicators. I

1

Introduction

Harry Markowitz published his landmark article Portfolio Selection 70 years ago, which ended up winning him the Nobel Memorial Prize in Economic Sciences in 1990 [2]. This groundbreaking theory empowered generations of academics and practitioners to pursue *asset allocation*, and it remains one of the most important strategies an investor can employ to increase returns and reduce the portfolio's overall volatility [3]. Since then, there have been many attempts to extend and enhance the work done by Markowitz, and yet despite all this progress, to this day, few would argue that asset allocation is an easy task. However, with recent success within the field of machine learning and the increased availability of data, some of the key challenges in portfolio theory might be close to a solution.

1.1 Background

The biggest challenge for every investor is how to increase returns while mitigating risk continuously. There is consensus within the industry about the importance of asset allocation, but looking at, e.g. the years 2002 and 2021 in Figure 1.1, fixed-target portfolios are not always optimal, as a dynamic portfolio overweight in bonds in 2002 and then overweight in stocks in 2021 would have yielded a higher return. In addition, the portfolio optimisation model (mean-variance analysis) suggested by Markowitz has limited impact in practice due to estimation issues when applied to real data.

Today, the financial market is one of the largest big data generators, with multiple terabytes being generated daily. Although this is being exploited to some extent by skilled financial institutions, there are still decisions based on purely classical financial theory, some of which have many drawbacks, argued by financial professor Marcelle Chauvet, among others, in [4], [5].

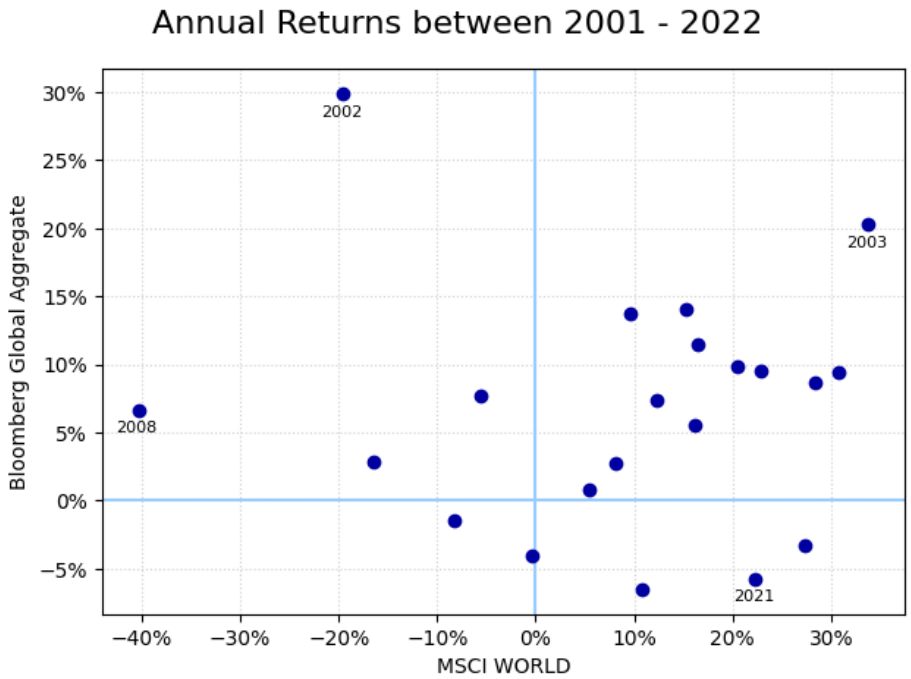


Figure 1.1: The annual returns of the two indices studied in this project, with the four most exceptional years labelled in the graph. 2002 was a year when the economy was still struggling to recover from the recession that had started in 2001. The uncertainty in the economy led investors to seek the relative safety of bonds, driving up bond prices and depressing stock prices. The following year, 2003, the economy was recovering, contributing to the rise of stocks and bonds. In 2008, the world went into a severe worldwide crisis caused by the United States housing bubble bursting. As stocks plunged, investors shifted their money into lower-risk government bonds, increasing their price. In 2021 global bonds slumped into their first bear market in a generation, under pressure from central bankers determined to quash inflation caused by two years of expansionary fiscal policy during the COVID-19 pandemic.

1.1.1 Asset Allocation

Asset allocation involves dividing your investments among different asset classes and is a critical consideration for constructing successful portfolios and assessing their risk exposure. Risk is defined in financial terms as the probability that an outcome or the actual gains of an investment will differ from an expected outcome, including the possibility of losing some or all of an original investment [6]. Risk is often measured by volatility and calculated according to Equation 1.1.

$$\sigma_N = \sqrt{\frac{\sum_{i=0}^{N}(x_i - \mu)^2}{N}} \cdot \sqrt{T}$$

$$\begin{aligned} \text{where: } & N = \text{the size of the population} \\ & T = \text{number of periods in the time horizon} \\ & x_i = \text{each return from the time horizon} \\ & \mu = \text{average return from the time horizon} \end{aligned} \tag{1.1}$$

A fundamental idea in finance is the relationship between risk and return. As risk increases, investors seek higher returns to compensate for taking such risks, called risk premium. Financial asset classes are a grouping of investments with similar characteristics, such as equities (e.g. stocks), fixed income (e.g. bonds), real estate, commodities, and currencies. The two most common financial asset classes are stocks and bonds, and they are therefore the focus of this thesis. Historical data shows that stocks are more volatile than bonds and are therefore considered the riskier asset among the two. Bonds are less volatile, known for their hedging characteristics and low correlation with the stock market, and due to these less risky attributes, it is well known that a diversified portfolio should include both [7]. Within these asset classes, there are further subcategories. However, these are beyond the scope of the thesis and are not explained in further detail.

When managing risk in a portfolio that includes these two asset classes, the most common approach is to adjust the allocation between stocks and bonds to balance risk exposure. For example, an investor with high risk tolerance might have a fixed-target portfolio with 80/20 stocks and bonds to gain greater exposure to the more volatile stock market, with the chance of higher returns and the risk of greater losses. Meanwhile, an investor with a low risk tolerance might instead have a fixed-target portfolio with 20/80 stocks and bonds to decrease potential risk, with the knowledge of maybe missing out on greater returns. However, as seen in Figure 1.1, bonds have experienced negative annual returns three times since 2001, while stocks have been rising, causing the most vulnerable investors to lose money on the investment that was considered safer.

As of early 2023, no better conventional strategy than these fixed-target portfolio allocations has been reported in the literature, and with 2022 being yet another year where bonds have failed to hedge the investors' savings, this topic is again trending.

1.1.2 Nordea

Nordea is the largest bank in the Nordics, with a market capitalisation of more than SEK 400bn and operations mainly in Sweden, Norway, Denmark, and Finland. The bank's core business areas include personal banking, business banking, large corporates & institutions and asset & wealth management. Asset & wealth management offers savings and investment products and manages the accumulated wealth of customers with a total asset under management (AuM) of EUR 359bn in Q4 2022 [8].

Within Nordea Asset & Wealth Management, the thesis was carried out in collaboration with the two teams, House View and Quantitative Solutions & Equity Research, in Investments. Alongside research papers, Investments provides the official strategic and tactical Nordea market views and model portfolios for retail clients. Currently, the tactical views published by House View are mainly based on economic models and the experienced opinion of investment strategists. However, developing deeper quantitative models could potentially enhance customer returns and the bank's competitive position.

1.2 Aim

The aim is to investigate how the predictive performance of gradient boosting algorithms translates into the field of asset allocation, as well as to examine whether an increased amount of premium data increases the predictive performance of machine learning models used in previous research for asset allocation. The hope is also that this report can increase transparency in the industry regarding investment strategies with comprehensive information on methods, models, and data.

1.3 Goals

The main goal of this thesis is to investigate how machine learning can be used to dynamically optimise a virtual portfolio of stocks and bonds for the upcoming month. As a proxy for the stock and bond market, two indices are used. The MSCI World Index (hereafter referred to as the Stock index) for stocks and the Bloomberg Global Aggregate Index (hereafter referred to as the Bond index) for bonds. The Stock index covers 85% of adjusted free-float market capitalisation in each included country, while the Bond index tracks investment-grade fixed-rate issuances from high-income corporations and countries spanning the globe. The coverage of these two indices replicates a well-diversified portfolio that spans most major sectors and developed countries in the global economy [9], [10].

The objective is to create a machine learning-based portfolio that will outperform a benchmark portfolio with a fixed-target allocation of 50% stocks and 50% bonds without increasing risk. The choice of a 50/50 fixed-target portfolio is significant as it represents a diversified portfolio, which is a fundamental concept in investment theory, dating back to Harry Markowitz and his Modern Portfolio Theory [2]. Using

this benchmark allows for meaningful comparisons and evaluation of the machine learning-based portfolio's effectiveness [11].

This problem statement will be approached by analysing the state of the economy with the help of financial data, see Section 2, and supervised learning algorithms, see Section 3.1, to find leading indicators for the stock and bond market, respectively. The portfolio allocation is re-evaluated monthly depending on future prospects for the various asset classes, and the proportion of stocks and bonds is set accordingly.

The problem is studied from a global perspective since the major financial markets historically have a strong correlation. The goal is divided into two smaller and more manageable sub-problems, according to the list below.

1. Find leading indicators for the Stock and Bond index using machine learning algorithms such as Random Forest and gradient boosting.

2. Predict the allocation of stocks and bonds that will outperform its benchmark with a lower or equal amount of risk for the upcoming month, see all targets in Section 1.3.1.

By taking on these challenges, the goal is to improve the process and quality of the *tactical level 1 bet*, i.e. weightings between stocks and bonds, at Nordea. This approach works as a bridge between classic portfolio theory and state-of-the-art machine learning. Increased precision in stock-and-bond-market forecasting helps investors outperform the market while reducing risk due to the diversification that decreases volatility in the portfolio.

1.3.1 Portfolio Allocation Targets

The performance of the models is evaluated monthly and measured in relative return compared to the targets listed in increasing difficulty below. The portfolio allocation is rebalanced monthly to ensure that decisions are based on the most recent available data. This provides a measurement that explicitly indicates how well the models perform relative to the current strategies used by Nordea.

1. The main target is to outperform an equal-weighted fixed-target portfolio on a monthly basis with less or equal risk.

2. The second target is to outperform the current Nordea allocation strategy.

Success is already achieved if a model reaches the first target since this proves that machine learning can be used to achieve a higher-return portfolio allocation of stocks and bonds compared to a fixed-target portfolio without increasing the risk. If a model also reaches the second target, it would perform better than Nordea's current strategy, which would be a great success.

The last target contains classified information and is only evaluated internally and therefore is not presented in this thesis.

1.4 Challenges

Since the thesis topic touches on financial and computer science questions, there are several challenges in both areas.

The modern portfolio theory by Harry Markowitz is still the most adopted strategy for asset allocation. Machine learning models have not yet become the industry standard. Therefore, there are no guarantees of successful results when evaluating the models.

The second challenge is from a data perspective. Good data is almost always more important than a perfect model when working with any analysis. Data must be chosen carefully to avoid situations where the choice of training data limits a good model. The approach to this challenge is explained in more detail in Section 2.

The third challenge is to avoid backtest overfitting, a common problem in mathematical finance. This occurs when historical market data is used to develop an investment strategy and numerous variations of the strategy are tested on the same data set. Backtest overfitting is now recognised as a primary reason why quantitative investment models and strategies that appear promising on paper (based on backtests) often fail to perform as expected in practice [12].

The fourth challenge is to divide the data used to train the models, as they need to generalise well across all market conditions. There have been seven major financial crises that the world has witnessed in the last 100 years, and they have all come in various forms, such as event-driven, cyclical, or structural.

1.5 Limitations

The machine learning models are only evaluated on the two asset classes, stocks and bonds. Therefore, predictions of other asset classes, specific regions, or individual financial instruments are beyond the scope of this thesis.

The models are developed to fit within the limitations of the Nordea allocation mandate. Hence, no asset class can allocate more than 60% and no less than 40% of the portfolio. The models can not allocate money in cash, meaning that the proportion of stocks and bonds must sum up to 100%. The models can not use short selling or leverage.

Only a frictionless market is considered. This means that trading occurs without transaction costs, taxes, restrictions, or impediments. Simulating a more realistic market with, e.g. transaction fees, can be interesting but is not relevant for this thesis, especially due to the low-frequency trading.

Additionally, it is important to note that certain time series, such as BNP numbers, are prone to revisions after their initial release. This introduces the potential for data leakage when using such data to train the models. However, considering the constraints of time and resources, a systematic and sustainable approach to accessing

primary data is not pursued, as the anticipated impact on the results is deemed relatively minor due to the few time series that are affected.

The final limitation is the lack of access to good, high-quality hardware. Only laptops with 11th Gen Intel i7 processors are available for this thesis. Not even a GPU, which is highly recommended when using more computationally heavy models, such as deep neural networks, due to the longer training times. Therefore, the models selected are partly chosen because they are lightweight and computationally efficient.

1.6 Literature

The financial field is highly researched due to its impact on society and the possible returns of successful methods and models. The subfield of asset allocation is no different, with Harry Markowitz and James Tobin winning the Nobel Memorial Prize in Economic Sciences for their work on portfolio theory. However, finance is also a secretive area where most research remains proprietary at the institutes that develop them, as any publication would result in a lost edge to the market.

1.6.1 Previous Work

Quantitative approaches built on machine learning models have previously been used to outperform the financial market. The predictive performance of Random Forest models has been consistent in recent years, achieving great results in financial areas such as algorithmic trading and stock analysis [13], [14]. A study conducted at Cornell University found that a Random Forest model could reduce risk while surpassing its benchmark by 3.4% annually [15]. In another study from Lund University, the author found a model that outperformed its benchmark by 3.7% annually with a lower risk [16]. These findings highlight the potential of the Random Forest algorithm in improving investment strategies, even if they used unconstrained allocation, which the rational long-term investor tends to avoid, and prediction horizons very different from what is studied in this thesis.

With gradient boosting outperforming Random Forest models in individual stock predictions in recent studies, the question remains as to whether performance translates into other financial domains, such as asset allocation [17].

1.6.2 Machine Learning in Finance

Over the years, numerous publications have explored machine learning in finance. However, given the failure to translate backtest profits into real-world success, the most intriguing publications instead emphasise the methodology of developing a financial machine learning model.

Marcos López de Prado, a renowned expert in quantitative finance and algorithmic trading, is the author of multiple well-cited books and papers on the subject. In the book *Advances in Financial Machine Learning*, he delves deep into the practical aspects of implementing machine learning models in finance, and he puts a significant

focus on the evaluation and validation of machine learning models in finance [18]. In the following book *Machine Learning for Asset Managers*, he provides a broad overview of machine learning techniques applied to the field of asset management [19]. While both books cover similar ground, Advances in Financial Machine Learning can be seen as a more specialised and advanced continuation of the concepts introduced in Machine Learning for Asset Managers.

Another influential publication on the subject is the Deutsche Bank Quant Handbook, whose second part has garnered widespread attention for its nuanced perspectives on quantitative investing [20]. Titled *Seven Sins of Quantitative Investing*, the paper, authored by Yin Luo and his team, offers a comprehensive and insightful analysis of why machine learning models in finance frequently fail to achieve real-world success. The seven sins are listed below:

1. **Survivorship bias:** Backtesting or evaluating strategies, excluding companies that have gone bankrupt, been delisted, or acquired.
2. **Look-ahead bias:** Unintentional incorporation of future information into the training and testing of predictive models
3. **Storytelling:** Making up a story ex-post to justify some random pattern
4. **Data mining and data snooping:** Repeatedly testing hypotheses or strategies on the same dataset, leading to potential overfitting and misleading results
5. **Transaction costs:** Ignoring transaction costs can lead to overestimated returns
6. **Outliers:** Ignoring or basing a strategy on a few extreme events can have an effect on how well the model generalises to future events.
7. **Shorting:** Taking a short position on cash products requires finding a lender, making it a non-implementable strategy for all stocks.

2

Data

Studying leading indicators is a long-lived tradition in economic research, dating back to at least 1946 with the book *Measuring Business Cycles* by Burns and Mitchell [21].

This chapter covers a description of the data, how it has been preprocessed, the features extracted from it, and the initial analyses done.

2.1 Description

The data used throughout the thesis were available through Refinitiv, one of the largest providers of premium financial market data and infrastructure. Refinitiv provides instant access to over 16 million economic indicators from 215 countries and regions dating back as far as 120 years [22].

In total, a set of 48 different time series were used and evaluated, see Appendix A.1. The time series covered various categories such as indices, risk-premium data, interest rates, sentiment surveys, and indicators on business, geopolitical, and financial conditions. Some of the time series used were composite indicators built with multiple underlying variables. The time series were chosen at the discretion of the advisors at Nordea, Karl Larsson and Fredrik Lundström.

Since most of the time series are provided by paid services, such as Refinitiv, they are rarely used in earlier work. Historical data and the work of Travis Berge have shown how some of these premium time series work as leading indicators of the economy and thus can provide an additional edge to earlier literature that only used stock market data such as price, volume, and volatility [17], [23]. Standard premium leading indicators are surveys measuring business climate, credit in the economy, and market sentiment. Local regions often have their own interpretation of these indicators. However, the American versions are commonly used due to their extended amount of historical data, the size of the surveys, and the proportion of American equity in global market cap.

2.2 Preprocessing & Cleaning

Data preprocessing-and-cleaning is crucial in enabling efficient analysis and obtaining satisfactory model results. The time series used were measured in different units,

used different granularities, or were issued in different regions as they measured different things. For example, some series used percentages (%), while others used basis points. Most series used the US dollar (USD) as currency, but some used Euro (EUR).

Once the preprocessed data was downloaded, missing and unwanted values such as NaN values and leading zeros (default value indicating missing value) were removed. In addition, if possible and needed, some data was also cut off to guarantee that the time series started on January 1st to make comparisons between time series easier in a later stage.

Another crucial filtering of the data was the cut-off point for the Stock and Bond index. Historical data for the Stock and Bond indices were available dating back to 1970 and 1991, respectively. However, the granularity of this data was limited to monthly intervals. It was not until 2001 for the Stock index and 1999 for the Bond index that the granularity increased to daily intervals. Therefore, to simplify the development process, only data starting from 2001 was used.

2.3 Feature Engineering

Feature engineering involves transforming raw data into a set of features that are utilised to train and enhance the performance of machine learning models. This enables extracting crucial information, identifying patterns, and computing domain-specific features. The feature engineering process is crucial to the success of machine learning models, as the quality of the features used to train the models can significantly impact performance.

In this case, the raw data was the time series value at a particular date. For each time series and feature category (described below), rolling windows of one month, one quarter, and one year were calculated. The window lengths were chosen partly on the recommendation of the Nordea advisors, but also because of the periodic nature of the financial market. Financial data is namely most often presented in a daily, monthly, quarterly, and yearly matter.

Because the different time series varied in granularity, the window frames were only computed if they were compatible. For example, it does not make sense to compute a monthly feature on a time series with only quarterly data.

Another issue with time-series feature engineering was the *initial* missing data. For example, a feature measured over one year generally needs one year of data before the first data point can be computed. The options were to either not use this initial period at all, backfill with the next occurring future value, or fill with some default value. To avoid losing more data than necessary and data leakage by backfilling, the elected method was to use a growing rolling window that starts at length one and grows until its intended length. Default values were not used as no appropriate value could be found.

In total, circa 1200 features were computed. The various feature categories are presented in further detail below.

2.3.1 Value

The value represents the observation recorded at a specific time for each individual time series. The granularity of these values can vary, ranging from daily, monthly, to quarterly, depending on the specific time series. It is important to note that the values are always organised in a temporal order.

2.3.1.1 Value Difference

The value difference is computed by subtracting the previous value from the current value over a rolling window, see Equation 2.1.

$$\text{Value Difference} = \text{Current Value} - \text{Previous Value}. \tag{2.1}$$

2.3.2 Returns

Return is the relative change between two values and is calculated according to Equation 2.2. The return feature was central in this thesis since it was the target variable (more on this in Section 4), and it was also used to calculate other features.

$$\text{Return} = \frac{\text{Current Value} - \text{Previous Value}}{|\text{Previous Value}|}. \tag{2.2}$$

2.3.3 Volatility

Volatility refers to the degree of variation or fluctuation in the price or value of a financial instrument (or market) over time. Volatility is often used as a measure of risk, since a highly volatile asset is considered riskier than a less volatile one. High volatility implies greater potential for both gains and losses, while low volatility implies relative stability with smaller potential gains or losses.

Volatility is typically measured using standard deviation, see Equation 1.1, which indicates the extent to which the values in a data set deviate from the mean.

2.3.4 Exponential Moving Average (EMA)

Exponential Moving Average (EMA) is a moving average that is often used to help identify trends and potential changes in the direction of an asset [24]. It is similar to the simple moving average (SMA), which is calculated by taking the mean of a specified number of data points for a given period of time. EMA differs by applying a smoothing factor, α, which will weigh recent data more heavily than older data and is calculated according to Equation 2.3.

$$\text{EMA} = \text{value} \cdot \frac{\alpha}{1 + \text{window\_size}} + \text{EMA}_{-1} \cdot \left(1 - \frac{\alpha}{1 + \text{window\_size}}\right) \tag{2.3}$$

where: $\alpha = 2$.

2.3.5 Sharpe Ratio

Sharpe ratio is a measure of risk-adjusted return used to evaluate the performance of an investment or portfolio. It was first proposed by Nobel laureate William F. Sharpe in 1966 under the name reward-to-variability ratio and is widely used by investors to compare the risk-adjusted returns of different investments [25].

The Sharpe ratio is calculated by subtracting the risk-free rate of return from the investment's return and dividing the result by the investment's standard deviation, see Equation 2.4. Values above 1 are generally considered good [26].

$$\begin{aligned} \text{Sharpe Ratio} &= \frac{R_p - R_f}{\sigma_p} \\ \text{where: } R_p &= \text{Expected return of the investment} \\ R_f &= \text{Risk-free rate of return} \\ \sigma_p &= \text{Standard deviation of the investment's returns.} \end{aligned} \tag{2.4}$$

For this thesis, *J.P. Morgan Global Aggregate Bond Index* was used as a risk-free asset. The reason is that it's backwards-looking instead of forward-looking, such as the *US 10-year treasury bond yield*.

2.3.5.1 Sharpe Ratio Difference

The Sharpe Ratio Difference is simply computed by subtracting the previous value from the current value.

$$\text{Sharpe Difference} = \text{Current Sharpe} - \text{Previous Sharpe}. \tag{2.5}$$

2.3.6 Sortino Ratio

Sortino ratio is very similar to the Sharpe ratio, but instead focuses only on the downside risk [27]. The only difference in the calculation is to divide by the standard deviation of the downside returns instead of the returns. See Equation 2.6.

$$\begin{aligned} \text{Sharpe Ratio} &= \frac{R_p - R_f}{\sigma_d} \\ \text{where: } R_p &= \text{Expected return of the investment} \\ R_f &= \text{Risk-free rate of return} \\ \sigma_d &= \text{Standard deviation of the downside.} \end{aligned} \tag{2.6}$$

2.3.6.1 Sortino Ratio Difference

The Sortino ratio difference is simply calculated by subtracting the previous value from the current value.

$$\text{Sortino Difference} = \text{Current Sortino} - \text{Previous Sortino}. \tag{2.7}$$

2.3.7 Maximum Drawdown (MDD)

Maximum drawdown (MDD) is a measure of the largest percentage drop in the value of an investment or portfolio from a previous peak to a subsequent trough. It is used to assess the risk of an investment and to evaluate the historical performance of an investment or portfolio [28].

The maximum drawdown is calculated by taking the difference between the highest value of the investment or portfolio and the subsequent lowest value, divided by the highest value. This provides a measure of the percentage decline in value from a peak to the subsequent lowest point, known as a trough, see Equation 2.8.

$$\text{MDD} = \frac{\text{Trough Value} - \text{Peak Value}}{\text{Peak Value}}. \tag{2.8}$$

2.3.8 Correlation

Correlation is a statistical technique that is used to measure the strength and direction of the linear relationship between two quantitative variables. In other words, correlation measures the extent to which the values of one variable are associated with the values of another variable. The correlation coefficient takes a value between -1 and +1.

- A correlation coefficient of -1 indicates a perfect negative correlation, where the two variables move in opposite directions (i.e. as one variable increases, the other decreases).
- A correlation coefficient of +1 indicates a perfect positive correlation, where the two variables move in the same direction (i.e. as one variable increases, the other also increases).
- A correlation coefficient of 0 indicates that there is no correlation between the variables.

The correlation is computed between the returns of the series and the returns of the target series (Stock or Bond), see Equation 2.9.

$$\begin{aligned} \text{Corr} &= \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \\ \text{where: } x_i &= \text{Values of the x-variable in a sample} \\ \bar{x} &= \text{Mean of the values of the x-variable} \\ y_i &= \text{Values of the y-variable in a sample} \\ \bar{x} &= \text{Mean of the values of the x-variable.} \end{aligned} \tag{2.9}$$

2.3.9 First Derivative

In order to quantify the trend direction and magnitude, a regression line is fitted to the series returns. From the fitted regression line, the slope coefficient is extracted,

see Equation 2.10.

$$y = kx + m \implies y' = k = \text{First Derivative.} \tag{2.10}$$

2.3.10 Second Derivative

In order to get a sense of how fast the rate of change itself is changing, a second derivative is calculated. The second derivative is computed by taking the relative change of the returns, i.e. "the returns of the returns", see Equation 2.11.

$$\text{Second Derivative} = \frac{\text{Current Returns} - \text{Previous Returns}}{\text{Previous Returns}}. \tag{2.11}$$

2.3.11 Z-Score

The Z-score, also referred to as the *standard score*, is a statistical measure that quantifies the number of standard deviations the observed value deviates from the mean [29]. The Z-score is the number of standard deviations by which the value of a raw score is above or below the mean value and is calculated using Equation 2.12. For example, if the Z-score equals 0, it indicates that the data point's score is identical to the mean, while a Z-score of 1.0 would indicate a value that is one standard deviation above the mean.

$$\begin{aligned} \text{Z-score} &= \frac{x - \mu}{\sigma} \\ \text{where: } x &= \text{Observed value} \\ \mu &= \text{Mean of the sample} \\ \sigma &= \text{Standard deviation of the sample.} \end{aligned} \tag{2.12}$$

2.3.11.1 Z-Score Difference

The Z-score difference is simply computed by subtracting the previous value from the current value, see Equation 2.13.

$$\text{Z-score Difference} = \text{Current Z-score} - \text{Previous Z-score.} \tag{2.13}$$

2.4 Analysis

A major part of developing machine learning models is understanding the data. Gaining insight into the fundamental information and relationships within the data can lead to identifying more effective features that can enhance the model's performance. Therefore, the initial part of the thesis was spent exploring and visualising the data, using different statistical functions to gain useful insights.

Correlation, mentioned in Section 2.3.8, was an initial metric used to rule out any obvious relationships between the Stock and Bond index to the rest of the time series.

Cross-correlation, similar to correlation, is a measure of similarity between two sets of time series as a function of the displacement of one relative to the other. Cross-correlation shows when the best match occurs as correlation is calculated over different time shifts. Cross-correlation is mainly used in portfolio management to measure the degree of diversification among the assets contained in the portfolio. This thesis used it in an attempt to find time shifts that would make a time series a potential leading indicator for the Stock or Bond index.

Windowed cross-correlation was also used, which is a variant of cross-correlation but with fixed time windows of correlation sliding across the time series.

Autocorrelation is equivalent to correlation when correlating a time series with itself. It was used in an attempt to find periodical patterns in the data.

The Granger causality test was also used, which is a statistical hypothesis test created by Nobel laureate Sir Clive Granger to determine whether a time series is useful for forecasting another.

When studying market data, one must consider absolute and relative changes in value, as momentum greatly impacts an asset's returns. There are many ways to find the current momentum of an asset, for example, Year-Over-Year (YoY) growth. YoY growth allows for gauging an asset's financial performance over time, whether it is improving, static, or worsening. For the analysis, YoY growth was used to visualise the time series from a new perspective as well as potentially finding new, lesser obvious correlations, see Figure 2.1.

Figure 2.1: MSCI World (Year-Over-Year) vs. US ISM PMI index, with a correlation of 0.739.

However, despite much time being invested into data exploration and analysis, little

was gained in terms of useful knowledge for the features or models moving forward, with the exception of the correlation feature.

3

Machine Learning Fundamentals

This chapter covers the fundamental theory and terminology as well as an introduction to the machine learning models and techniques used in the thesis. Furthermore, the performance metrics that are used to evaluate the quality of the models are introduced.

3.1 Models

The thesis used supervised learning algorithms, which is a type of machine learning in which an algorithm is trained to learn the relationship between input variables (also known as features) and output variables (also known as labels or targets) by using a set of labelled training data.

The models were constructed using regression (reg) and classification (clf) algorithms. Regression involves estimating the relationships between a dependent variable (target variable) and one or more independent variables (feature variables). On the other hand, classification focuses on identifying the category or sub-population to which an observation belongs.

The output of the models was predictions on how the indices would move, then used by a rule-based allocation model to create a portfolio allocation of Stocks and Bonds, with the aim of meeting all the targets stated in Section 1.3.1.

The machine learning models that were developed and evaluated are as follows (in order of implementation).

1. Random Forest Regressor (RandomForestRegressor from sklearn.ensemble [30])
2. Extreme Gradient Boosting Regressor (XGBRegressor from dmlc/xgboost [31])
3. Random Forest Classifier (RandomForestClassifier from sklearn.ensemble [32])
4. Extreme Gradient Boosting Classifier (XGBClassifier from dmlc/xgboost [31])

More complex and sophisticated models suited for time series data, such as Recurrent Neural Networks, were considered but not pursued due to their lack of interpretability and the limitations of available hardware (neural networks often require a GPU to complete training in a reasonable time).

3.1.1 Decision Trees

A decision tree is a decision support tool that uses a tree-like model of decisions with their possible consequences. Decision trees originated from the discipline of decision analysis but have since become a popular tool in machine learning and are integrated into several prominent machine learning algorithms, such as Random Forest (Section 3.1.2) and gradient boosting (Section 3.1.3).

One of the main advantages of decision trees is their interpretability, as they are easy to understand and can also be displayed graphically so that non-experts can interpret [33]. A disadvantage of decision trees when working with time series data is that they cannot capture temporal patterns such as trends, periodicities, and sequences.

Decision trees are a non-parametric supervised learning algorithm, which is used for both regression (continuous values) and classification (discrete values) tasks [34]. Decision trees where the target variable can take continuous values are also called regression trees. This can be useful when the relationship in the data is found to be non-linear, as seen in Figure 3.1. Decision trees where the target variable can take discrete values are also known as classification trees, or multi-class classification trees if it predicts outcomes for more than two classes.

Decision trees are weak learners. Weak learners are models that perform slightly better than random guessing or taking the mean of a sample and are often used together in ensemble models to form strong learners with higher accuracy [35].

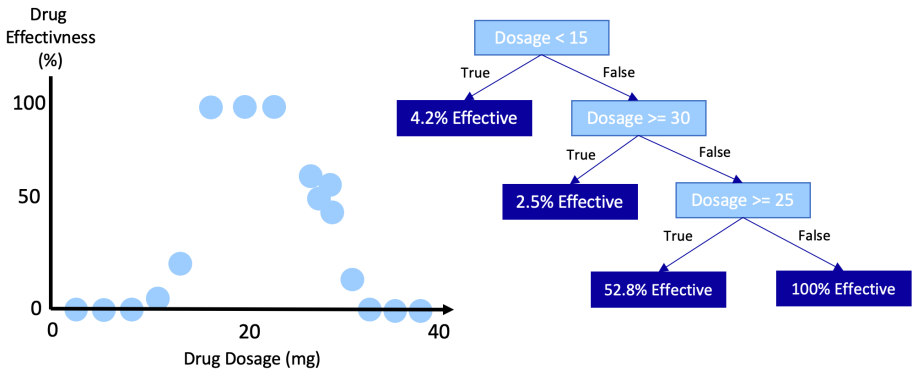


Figure 3.1: Example of how a regression tree can be used to fit a model to continuous non-linear data. Each leaf of the tree is labelled with a value, which is the output of the model.

3.1.2 Random Forest

Random Forest is an ensemble learning method first proposed in 1995 by Tin Kam Ho [35] and later expanded by Leo Breiman and Adele Cutler in 2001 [36] to the version that was used in this thesis; see Algorithm 1. Random Forest is a meta-estimator that constructs a multitude of decision trees at training time that either returns the average (regressor) or the majority class (classifier) of all individual trees; see Figure 3.2.

Algorithm 1: Random Forest Algorithm

$[T_b]^B \leftarrow$ The ensamble of trees;
for $b = 1$ *to* B **do**
1. Draw a bootstrap sample Z^* of size n from the training data.
2. Grow a random-forest tree T_b to the bootstrapped data by recursively repeating the following steps for each terminal node of the tree, until the minimum node size fraction s_{min} or the maximum number of terminal nodes k_{max} are reached.
   (a) Select m variables at random from the p variables.
   (b) Pick the best varaiable/split-point among the m.
   (c) Split the node into two child nodes.

end
Return $[T_b]^B$

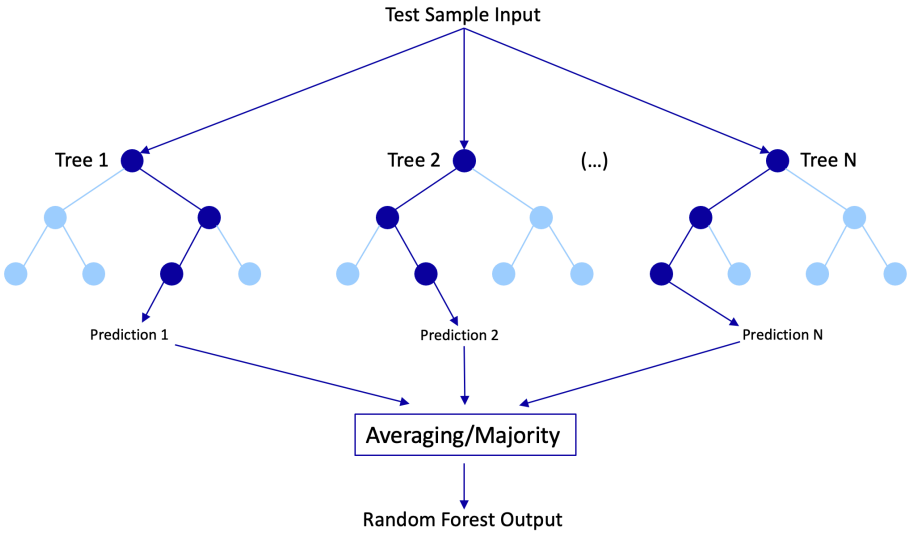


Figure 3.2: The Random Forest will build **N** unique decision trees that will each make a prediction. Random Forest is a strong learner constructed of many smaller decision trees, known as weak learners.

Random Forest models inherit their great interpretability characteristics from decision trees and require fewer computations than neural networks, which is a great benefit.

The method has previously shown promising results within the financial field, unlike other decision tree algorithms that tend to overfit and struggle with noisy data [14]. Random Forest models overcome this problem by training multiple decision trees on subsets of available data to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone [37]. The current state of the art, presented by Pinelis and Ruppert 2022 [15], used this method and outperformed a buy and hold strategy by 3.4% while also gaining significant improvements in the Sharpe ratio.

The Random Forest model used in this thesis contains a total of 17 tunable hyperparameters. However, only a subset of these were used, which are defined in Table 3.1.

Parameter	**Definition**
max_depth	The maximum depth of the tree.
max_features	The number of features to consider when looking for the best split.
min_sample_leaf	The minimum number of samples required to be at a leaf node.
min_sample_split	The min number of samples required to split an internal node.
n_estimators	The number of trees in the forest.

Table 3.1: Random Forest Parameters. These hyperparameters were optimised for both the regressor and the classifier.

3.1.3 Gradient Boosting

The gradient boosting algorithm was the primary focus of the thesis. Gradient boosting is closely related to the simpler Random Forest algorithm. Both algorithms are ensemble learning methods that use weak learners in the form of decision trees to perform regression or classification. The key difference is that Random Forest is a bagging ensemble method, while gradient boosting is a boosting ensemble method. Bagging creates multiple diverse models by training them independently, while boosting creates models sequentially, with each new model learning from the errors made by the previous model, see Figure 3.3. Gradient boosting offers several advantages, such as great interpretability and fast training times, even on less powerful machines. Additionally, it tends to outperform Random Forest models in terms of accuracy and predictive power [38].

The algorithm, whose pseudocode can be seen in Algorithm 2, was originally developed by Jerome H. Friedman in 2001. The model creates a strong prediction by combining weak learners (denoted f_m) over a fixed number of iterations (denoted M). The number of boosting iterations M is chosen to be the one that minimises the

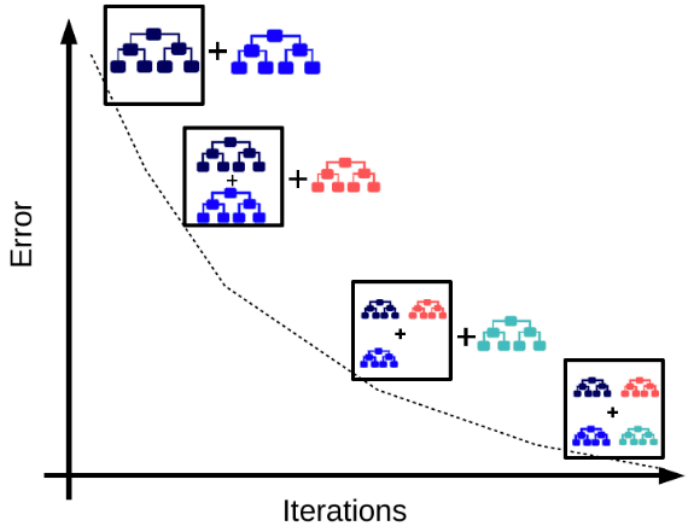


Figure 3.3: Gradient Boosting learning curve. Figure illustraded by Aratrika Pal [39].

Bayesian Information Criterion (BIC) of the final boosted model. The weak learners are simple binary decision trees that iteratively improve by learning from previous mistakes. At each iteration, a new residual value is predicted. The residual value is the difference between the estimated and true values. The weak learners then try to minimise the residual value until it reaches the fixed number of M iterations. At that point, it exits the outer loop and uses its current model to predict the results. The benefit of Gradient Boosting, which Friedman talks about in his paper, is how the algorithm has a low variance, resulting in good predictions on unseen data [40].

Algorithm 2: Gradient Boosting Algorithm

$f_0(x) \leftarrow argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$;
$M \leftarrow$ Number of iterations;
$N \leftarrow$ Numbers of datapoints;
for $m = 1$ *to* M **do**
  for $i = 1, ..., N$ **do**
    $r_{im} \leftarrow -[\frac{\delta L(y_i, f(x_i))}{\delta f(x_i)}]_{f=fm-1}$;
  end
  Fit regression tree to the targets r_{im} giving terminal regions
  $R_{jm}, j = 1, 2, ..., J_m$.
  for $j = 1, 2, ..., J_m$ **do**
    $\gamma_{jm} \leftarrow argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma$;
  end
  $f_m(x) \leftarrow f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
end
Return $\hat{f}(x) = f_M(x)$

The gradient boosting algorithm is also exciting from a financial perspective, as em-

pirical evidence shows that it can classify noisy data, a characteristic very common in finance [40]. Previous work with gradient boosting has also been done in other areas, such as recession and individual stock prediction, with great success [17], [23].

3.1.4 eXtreme Gradient Boosting (XGBoost)

XGBoost is a praised implementation of gradient boosting that specifically focuses on optimising the gradient boosting algorithm [31]. It offers several enhancements over traditional gradient boosting methods. These enhancements include a more regularised model to control overfitting, a customised loss function, and a highly efficient implementation that supports parallel processing and can handle large-scale data sets.

XGBoost has a total of 44 tunable hyperparameters. However, only a subset of these were used, which are defined in Table 3.2.

Parameter	**Definition**
colsample_bytree	Subsample ratio of columns when constructing each tree.
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree.
learning_rate	Boosting learning rate.
max_depth	Maximum tree depth for base learners.
n_estimators	Number of trees in Random Forest to fit.
subsample	Subsample ratio of the training instance.

Table 3.2: XGBoost Parameters. These hyperparameters were optimised for both the regressor and the classifier.

3.2 Model Evaluation Strategies

Model evaluation strategies play a vital role in developing and assessing machine learning models. They offer a systematic framework to determine the likely performance of a model on unseen data. In this context, two competing concerns arise: parameter estimates exhibit higher variance when there is less training data, and the performance statistic becomes more variable with limited testing data. Consequently, there are no universal solutions applicable to all data sets. Instead, the choice of strategy and data split should be tailored to the specific situation at hand. In this section, the two methods train-validation-test split and time-series cross-validation are explained, with their benefits and drawbacks.

3.2.1 Train-Validation-Test Split

Train-Validation-Test split is a model evaluation strategy where the available data is split into three unique subsets: one for training, one for validation, and one for testing. Models that do not perform hyperparameter tuning only use two subsets, train and test, as a validation set is unnecessary.

Normally, the original data set is shuffled before being split into subsets. Still, due to the temporal dependency of the data in financial time series and to prevent data leakage, the data is kept in its original order, as illustrated in Figure 3.4.

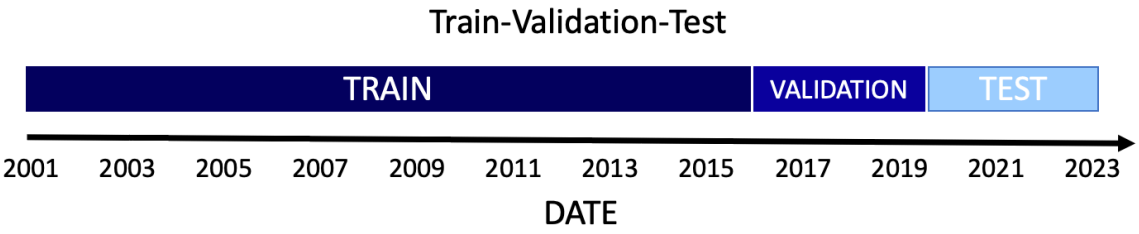


Figure 3.4: When developing a model for time series forecasting, it is important to remove future data and gap samples from the training set to avoid data leakage.

The training set is crucial for building a machine learning model. By analysing training data, the algorithm identifies relationships and patterns to determine optimal variable combinations to generate an effective predictive model [41]. The training set is usually the largest data set, often containing more than 70% of the total data [42].

The validation set is used to validate the model's generalisation performance during training and to tune the hyperparameters thereafter. It is necessary to have a validation data set in addition to the training set to avoid overfitting. However, the model can become indirectly biased towards the validation set because of the hyperparameter tuning.

The test set is a final out-of-sample (OOS) test of the model's generalisation ability. It is the best indication of the likely future performance of the model on unseen data. This sample is only used once the model development and hyperparameter tuning process is complete in order to detect and avoid backtest overfitting.

The size of the validation and test set can vary greatly and is highly dependent on the specific use case, but a good rule of thumb is to make them equal-sized.

Evaluating the models on the validation set prevents backtest overfitting by mitigating selection bias in multiple backtests. Refraining from conducting a backtest on the out-of-sample data until satisfactory results are obtained on the validation set reduces the risk of building a model based on a statistical fluke. Additionally, in the event of a negative outcome on the out-of-sample test, it is crucial to restart the process, as repeating tests on the same data is likely to lead to false discoveries. Typically, around 20 iterations are required to discover a false investment strategy within the standard significance level of 5%, something shown by López de Prado [18].

3.2.2 Cross-Validation

Cross-validation (CV) is a commonly used technique in machine learning to avoid overfitting and to evaluate a model's generalisation performance on an independent data set as an alternative to the basic train-validation-test split. It is often employed

in combination with hyperparameter tuning to determine the optimal hyperparameter values for a model. K-fold CV is a variant that divides the original data into K subsamples, see Figure 3.5, where each training sample is used to fit the model and tune the hyperparameters. CV is very useful when the dataset is small and when splitting the data into a typical train-validation-test set would significantly affect the model's accuracy.

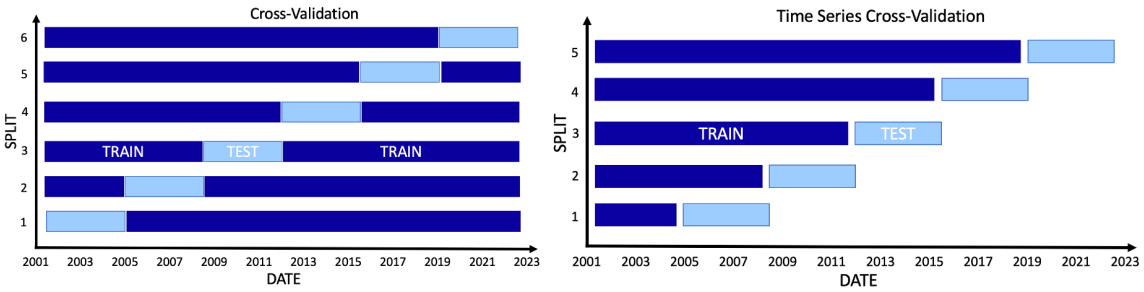


Figure 3.5: When developing a model for time series forecasting, it is important to remove future data and gap samples from the training set to avoid data leakage.

However, using a conventional CV approach may not be effective or valid when forecasting a time series due to the temporal dependency of the data, and access to future data during model training can result in data leakage. Ensuring that the test set has more recent data than the training set is crucial to address this challenge. This method is known as time-series cross-validation (TSCV). Another difference between CV and TSCV, also related to data leakage, is when each train/test set is adjacent to each other in sequence. There is a risk that variables at the end of the training set, known as gap samples, can incorporate some of the information from the test set, leading to look-ahead bias. Therefore, removing the gap samples when splitting the data is important.

3.3 Hyperparameter Tuning

Hyperparameter tuning, is an important part of the machine learning model development process, as the hyperparameter settings can significantly impact the model's performance. A hyperparameter is a control parameter that influences the learning process of a model, in contrast to other parameters, such as node weights, which are learnt by the model itself [43].

Hyperparameter tuning poses numerous challenges that make it a complex problem in practice. It can be computationally expensive when dealing with a large number of hyperparameters, and the configuration space is often intricate, involving a mix of continuous, categorical, and conditional hyperparameters. Additionally, optimising for generalisation is difficult, as training data sets are typically limited in size.

To address the challenge of generalisation performance, hyperparameter tuning is frequently employed in conjunction with model evaluation strategies, as outlined in Section 3.2. Generalisation performance is typically estimated through techniques

such as cross-validation on the training set or evaluation on a hold-out validation set.

3.3.1 Model-Free Blackbox Tuning Methods

There are many methods available to perform hyperparameter tuning, and due to the non-convex nature of the problem, global optimisation algorithms are usually preferred [43]. Model-free blackbox tuning methods are optimisation algorithms used to find the optimal configuration or parameters of a system without relying on an explicit mathematical model of the system. These methods are typically employed when the underlying system is complex, highly non-linear, or lacks a clear mathematical representation.

Grid search is the traditional and most basic method used to perform hyperparameter tuning. This method specifies a predefined set of hyperparameters that are then exhaustively searched. This ensures that no hyperparameter configuration is missed during the tuning process, in contrast to other methods, which may miss important configurations or spend more time exploring less promising ones. However, a major drawback of this method is that the number of evaluations grows exponentially as the set of hyperparameters grows, and thus the method is not always feasible.

A simple alternative to grid search is random search. As the name suggests, random search replaces the exhaustive enumeration of all combinations by randomly selecting the hyperparameters, where each setting is sampled from a distribution over possible parameter values. This method can outperform grid search, mainly when only a small number of hyperparameters affect the final performance of the machine learning algorithm, as illustrated in Figure 3.6 [44]. Random search can be a valuable method to initiate the search process, since it covers the entire configuration space, leading to the discovery of settings that often yield satisfactory performance. These settings can serve as a reference point when conducting more comprehensive guided search methods, such as grid search.

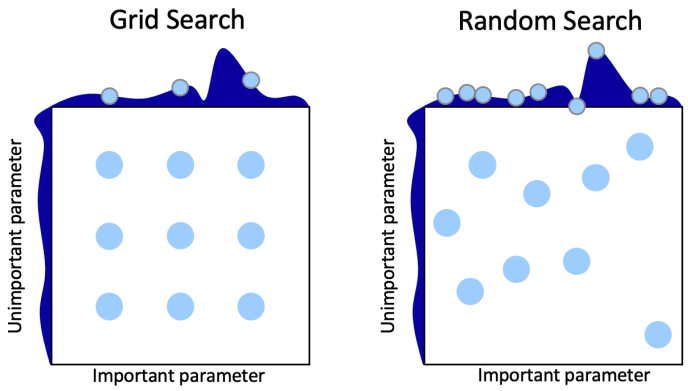


Figure 3.6: Comparison of grid search and random search minimising a function with one important and one unimportant parameter. However, when both parameters have a large impact on the result, grid search usually performs better. This figure is based on the illustration by Bergsta and Bengio [44].

3.4 Backward Feature Elimination

Backward feature elimination, also known as feature selection, is a technique which involves iteratively removing features from a model until an optimally performing subset of features is achieved. The process starts with a model that contains all the available features, and then one feature is removed at a time. The model is then evaluated with the reduced set of features, and the feature with the least impact on the model's performance is discarded. This process continues until the desired number of features or a predetermined threshold is reached.

One drawback with backward feature elimination is that features sometimes perform better when combined with other previously determined bad features, a result of non-linear relationships between the features. Such relationships can be hard to find when performing backward feature elimination, compromising the model's performance.

3.5 Performance Metrics

Performance metrics are crucial for model development and provide quantitative measurements of the quality of the model during all stages of its lifetime. They are used to evaluate trained models, for model selection and hyperparameter tuning during development and to monitor the performance of a deployed machine learning model in a production environment. Numerous performance metrics are available for evaluating models, each with unique characteristics. Thus, it is essential to understand when and how to utilise them appropriately and thoroughly. The metrics used in this thesis are explained below.

3.5.1 Coefficient of Determination (R^2)

The coefficient of determination, more commonly known as r-squared (or R^2), is a statistical measure that represents the proportion of the variance in the dependent (target) variable that can be explained by the independent variables (features) in a regression model. In other words, it indicates how well the regression model fits the observed data. The R^2-score is calculated according to Equation 3.1.

$$
\begin{aligned}
R^2 &= 1 - \frac{\sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \\
\text{where: } & y_i = \text{Observed value} \\
& \bar{y} = \text{Mean value of a sample} \\
& n = \text{Number of observations} \\
& f(x_i) = \text{Predicted value of } y_i \\
& x_i = \text{Set of input features}
\end{aligned}
\tag{3.1}
$$

The *r2_score* function from sklearn.metrics is used for practical implementation. The best possible score is 1.0, indicating a perfect fit. On the contrary, a score of 0.0

corresponds to a constant model that always predicts the average of the dependent variable, disregarding the input features. The score can also be negative because the model can be arbitrarily worse [45]. In finance, this metric is often used to determine the percentage of a price movement in a stock that is attributed to the price movement of the corresponding index [46].

3.5.2 Accuracy

Accuracy is a commonly used metric to evaluate the performance of a classification model. Accuracy measures how well the model correctly predicts the class labels of the input data, calculated according to Equation 3.2. The accuracy is the proportion of correct predictions (both true positives and negatives) over the total number of predictions made.

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\ \text{where: } TP &= \text{True Positive} \\ TN &= \text{True Negative} \\ FP &= \text{False Positive} \\ FN &= \text{False Negative} \end{aligned} \tag{3.2}$$

4

Methodology

Ensuring the prevention of the *Seven Sins of Quantitative Investing*, mentioned in Section 1.6.2, has been a central focus of the thesis, influencing every stage of the development process to uphold the integrity of the results. This chapter provides further elaboration on how the errors have been addressed. In addition, this chapter outlines the methods used during the course of the thesis, as well as the underlying choices behind them. The tools used to develop the models and the system as a whole are presented. The main parts of a machine learning project are also presented and discussed in the context of this specific thesis.

4.1 Development Approach

The approach used during this project can be described as a cycle with three steps: data analysis, model development, and evaluation of results. This cycle was used in a systematic way to thoroughly test the concepts and gain an understanding of how the different models interpret the data.

The implementation relied on Python and its renowned libraries for machine learning and numerical computation, including NumPy, SciPy, scikit-learn, pandas, and Matplotlib, among others. This decision was made due to the abundant availability of comprehensive online resources, the prevalence of open-source code, and the authors' extensive prior experience with these tools.

4.2 Model Input Processing

Once the feature engineering was complete, the data was almost ready to be fed into the models, but first, it needed to be processed and transformed in a suitable way for the task at hand.

4.2.1 Merging

The first part of the input processing consisted of merging all feature-engineered time series into a single pandas DataFrame table. This was done using the target variable as the basis and left-joining the remaining series onto the target variable, using forward-fill if needed. In this way, the series with more infrequent granularity

could still be used as input, and a lagged version of the target variable could also be used as input. Forward-fill was used instead of backfill to avoid data leakage.

4.2.2 Filtering

The next step was to filter the data in rows (date) and columns (features). The data was filtered to span the range from 2003 to 2023. As mentioned previously, the lower limit of the year 2001 was initially chosen because that was the year the Stock index started with daily granularity. However, as some time series did not begin until later than 2001, 2003 was chosen as a balanced lower limit between missing out on real data vs. including too much artificial data, as is explained in the next Section.

The columns that were Bond-specific were dropped when predicting the Stock index and vice versa. For example, dropping the Bond-correlation feature when predicting the Stock index.

4.2.3 Filling Missing Data

A problem with left-merging all time series onto the target index was that most, but crucially not all, time series went as far back as the target index. Therefore the resulting merge would have ended up containing some initial missing values, which, e.g. the Random Forest model does not support. Instead, the initial missing data for each feature was filled with the feature's average value.

4.2.4 Shifting independent variables

Because a subgoal of the project was to predict the returns of the Stock and Bond index one month ahead, the models also had to be trained with this time shift in mind. This was achieved by shifting the input features forward one month, including a shifted variant of the target variable, see Figure 4.1.

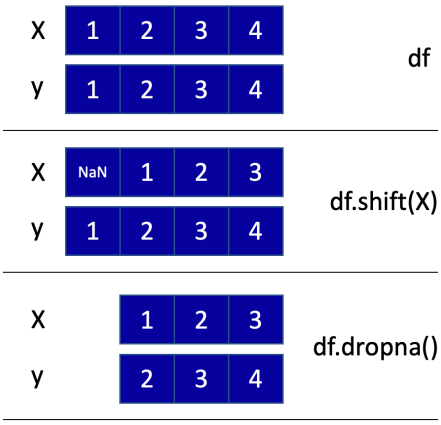


Figure 4.1: Shifting the independent (X) features.

4.2.5 Train-Validation-Test Split

The data remaining after the initial pre-processing was split into three sets in consultation with the Nordea advisors; Train (2003-2016), Validation (2016-2020), and Test (2020-2023), see Figure 4.2.

The entire data span included a total of 5197 days, with 3371 belonging to the train set, 1043 to the validation set, and 783 to the test set.

The train set contained 13 years of data corresponding to 65% of the total dataset. The period includes remarkable events such as the aftermath of the dot-com crash, including the following bull market that lasted until the global financial crisis in 2008. During this period, the correlation between the Stock and Bond index was low, as the Bond index increased when Stocks declined.

The validation set contained four years of data corresponding to 20% of the total dataset. The period was overshadowed by geopolitical tensions, concerns about global economic slowdown, and policy uncertainties that contributed to market swings.

The test set contained three years of data corresponding to 15% of the total dataset. The period includes remarkable events such as the COVID-19 crash in 2020, the bull market in 2021 as a result of the extreme expansionary fiscal policy and 2022 with historically low returns for a 60/40 portfolio. As the correlation between the stock and bond market was high in 2022, failing to hedge the investors, this test set also provided a measure of how good the models were at predicting relative *future* performance between the two assets.

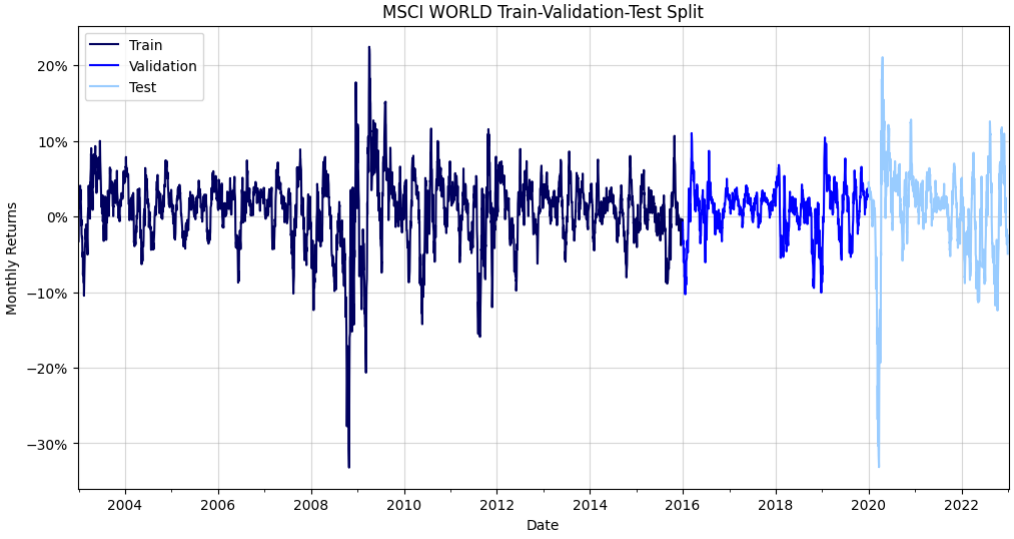


Figure 4.2: Train-Validation-Test Split. The models are trained on the training set, hyperparameter tuning is performed using the validation set, and the models' generalisation capabilities are tested on the test set.

Cross-validation (see Section 3.2.2) was also explored, but due to the limited amount of data, Time Series Cross-Validation was not able to perform well because of the

first few folds, which got little data to train on in correlation to the test data, which in turn brought the whole average down.

4.2.6 Numpy Transformation

In order to make the data compatible with the models from the scikit-learn and dmlc libraries, the pandas DataFrames were transformed into NumPy ndarrays as the final step before training the models.

4.3 Hyperparameter Tuning

Hyperparameter tuning was performed using a custom implementation of sklearn's *GridSearchCV*, as CV (Cross-Validation) was replaced by evaluation on a hold-out validation set instead. GridSearch can be computationally expensive, as mentioned in Section 3.3, but a parallel version was implemented, reducing the search time.

Random search (sklearn: *RandomizedSearchCV*) was tested as well but was outperformed by grid search due to the complex relationship between the parameters.

4.3.1 Random Forest

For the Random Forest models, there are a total of 12 tunable hyperparameters. However, only the five parameters in Table 4.1 were tuned. These were found to have the greatest impact on performance, and tuning all 12 parameters would be too time-consuming.

Parameters	**Default**	**Best REGR**	**Best CLF**
max_depth	None	20	10
max_features	1.0	10	sqrt
min_sample_leaf	1	2	1
min_sample_split	2	2	3
n_estimators	100	1000	200

Table 4.1: Random Forest default vs. best parameters. See Table 3.1 for parameter definitions.

4.3.2 XGBoost

For the XGBoost models, there are a total of 44 tunable hyperparameters. However, only the six parameters in Table 4.2 were tuned. These were found to have the greatest impact on the performance, and tuning all 44 parameters would be too time-consuming.

Parameters	Default	Best REGR	Best CLF
colsample_bytree	1.0	0.1	0.1
gamma	0	0.02	0.02
learning_rate	0.3	0.4	0.2
max_depth	6	3	15
n_estimators	100	25	25
subsample	1.0	0.7	0.6

Table 4.2: XGBoost default vs. best parameters. See Table 3.2 for parameter definitions.

4.4 Feature Elimination

All 1200+ features were initially fed into the Random Forest and XGBoost models. However, early findings suggested that this method was not feasible, since the models kept overfitting to the training data, even though parameters like `max_features` and `colsample_bytree` were thoroughly used. Therefore, feature elimination was required.

The development stage for the models consisted of a two-step process. First, an initial hyperparameter-optimised model using all features was developed. Then, a second and final hyperparameter-optimised model was created, using only the top 150 performing features of the previous model, which consistently gave better results. The number 150 was chosen by trial and error; using less than 150 gave worse results due to *underfitting*, and using more than 150 gave worse results due to *overfitting*. See Figure 4.3 for the feature importance ranking of the initial Random Forest model.

For the Random Forest model, the top features of only an initial Random Forest model were used. But for the XGBoost model, the top features of both an initial XGBoost model and an initial Random Forest model were tested. It turned out that the XGBoost model, which used the top features of a Random Forest model, was actually the most performant.

4.5 Model Evaluation

A backtest from 2020 to 2023 on the Stock and Bond index was performed when evaluating the prediction and allocation models.

Using Stock and Bond market indices can be helpful in mitigating survivorship bias when backtesting an allocation strategy. Indices maintain a history of their constituent stocks, and by accessing the historical composition of an index, the performance is based on companies that were part of the index at specific points in time, even if they are no longer actively traded. Figure 4.4 shows how only considering the current investment universe can lead to overly optimistic performance on historical backtests.

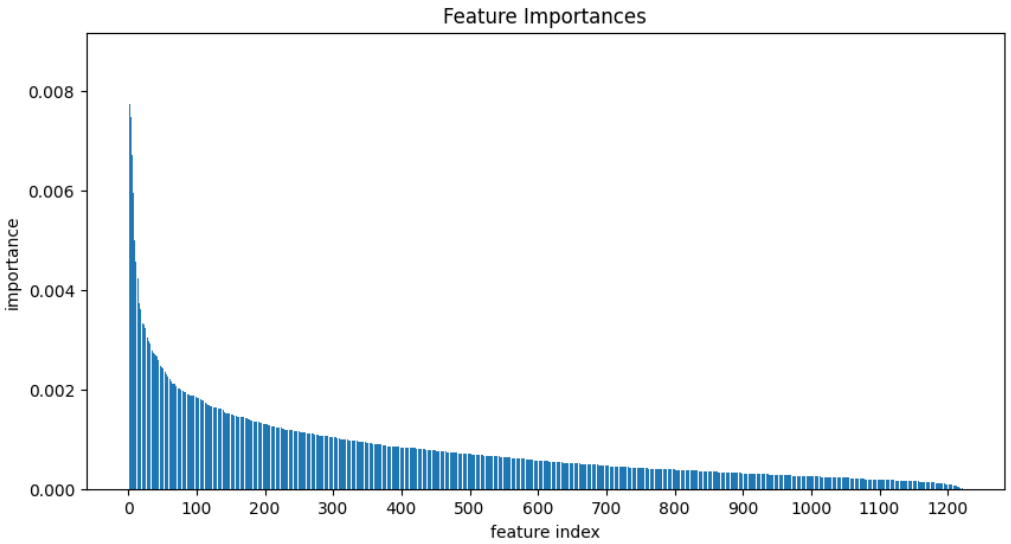


Figure 4.3: Feature importance from the initial Random Forest model trained on over 1200 features. However, due to the presence of noise, the number of features used in the final models were significantly reduced by up to 90%.

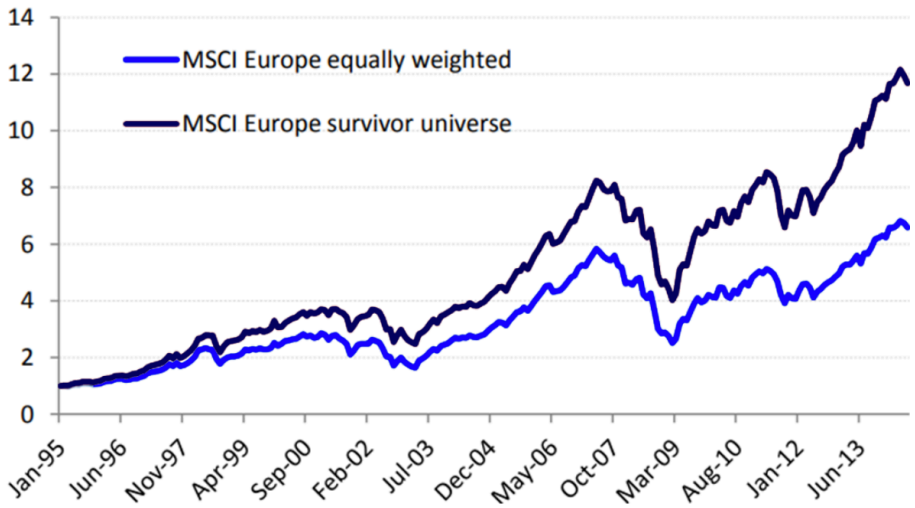


Figure 4.4: The backtested performance of MSCI Europe between 1995 and 2014 is almost twice as good when only considering the investment universe of 2014, a common error in quantitative investing. Graph produced by Yin Luo [20].

4.5.1 Prediction Models

The Random Forest and XGBoost machine learning models were used to predict the movements of the Stock and Bond index for the upcoming month. The regressors predicted the returns of the respective index for the upcoming month, while the classifiers only predicted the *direction* of the returns.

The models were evaluated on the validation set for model selection and ultimately evaluated on the test set for general predictive performance. The regressor models were evaluated using the R^2 score, while the classifier models were evaluated using accuracy.

4.5.2 Allocation Models

The portfolio's performance was assessed by rule-based allocation models through backtesting, using signals from the prediction models. There were two types of allocation models, one for the regressors and one for the classifiers.

The regressors used annual risk-adjusted returns as the allocation strategy, which is calculated according to equation 4.1. The asset that exhibited the highest predicted risk-adjusted return was selected for overweighting in the portfolio.

$$\text{Risk-Adjusted Returns} = \frac{\text{Yearly Returns}}{\text{Yearly Volatility(of daily returns)}} \tag{4.1}$$

The classifiers used the predicted direction of the Stock index as the allocation strategy. If the Stock index returns were predicted to be positive, the Stock index was overweight and the Bond index was underweight, otherwise, the Bond index was overweight and the stock index was underweight.

As mentioned in the limitations in Section 1.5, the allocation models were restricted to a 60/40 overweight/underweight principle.

The evaluation of the allocation models encompassed six widely used performance metrics in finance, listed below:

- **Total Return** often aligns with investors' objectives, allowing for meaningful comparisons between investment options or asset allocation strategies.
- **Alpha** helps assess the investment's performance relative to a benchmark and can explain whether a strategy has added value beyond what can be explained by market movements.
- **Volatility** provides insights into an investment or portfolio's risk and potential fluctuations. Volatility is a crucial component of risk management.
- **Sharpe Ratio** provides a measure of risk-adjusted performance, how a strategy has historically managed risk, and whether risk levels are acceptable within their risk tolerance.

- **Sortino Ratio** is similar to the Sharpe ratio, but focuses specifically on the downside risk.
- **Maximum Drawdown (MDD)** provides insights into the worst peak-to-trough decline experienced by an investment or portfolio. MDD is particularly relevant for investors who prioritise capital preservation.

Because of the uncertain nature of the stock market, the models were backtested for all rebalancing days in the month, and an average was computed as the final result.

4.6 Model Comparison

Once all the models were fully developed, they were compared against the target benchmarks defined in Section 1.3.1, as well as against each other, the results of which are presented in the next chapter.

5

Results

This chapter highlights the results and findings of the project, demonstrating how they align with the project's goals. The first and second section gives an in-depth analysis of the prediction and allocation model results, focused towards the first goal of the thesis, to build an outperforming portfolio. The last section is focused on the data and how feature importance has been utilised to achieve the thesis' second goal, to find leading indicators for the Stock and Bond index.

5.1 Prediction Models

The prediction models were assessed through a backtest, a historical simulation that gauges the performance of a strategy if executed in the past. While the results of the backtest do not guarantee future performance and only serve as a hypothetical sanity check, it is widely employed in the industry, despite being criticised for its susceptibility to errors [18]. One of the reasons for the popularity of backtesting is because it allows for result comparisons with other studies, providing insights into the performance of developed models in relation to existing literature.

5.1.1 Prediction Regressors

The regressor prediction models were the main focus of this thesis, as the hypothesis was that the increased detail that a regressor could provide over a classifier would be beneficial for how the rule-based allocation models would perform.

The results of the hyperparameter-optimised regressor models on the test set can be seen in Figure 5.1, together with the performance metrics in the various data sets in Table 5.1. The table includes a separate test set that excludes the year 2020, which marked the beginning of the COVID-19 pandemic. This separation allows for testing the model's resilience to large outliers, as numerous time series experienced significant deviations during and after the market crash in 2020.

When predicting the monthly returns for the Stock index, XGBoost achieved an R^2-score of 0.233 on the test set that excluded 2020, with the magnitude of the predictions posing as a primary drawback.

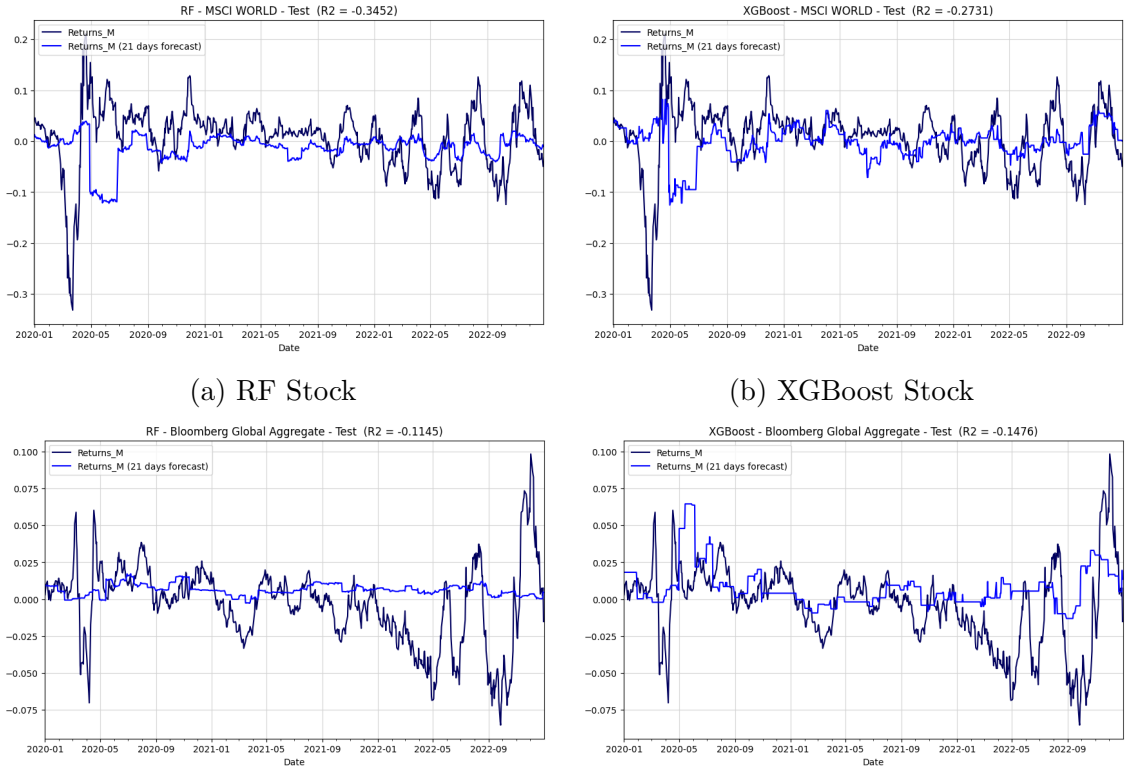
However, predicting the Bond index proved to be more challenging, with difficulties in identifying patterns and relationships in the data evident from the low R^2-score

REGRESSOR

Model	Train	Validation	Test	Test (2021-23)
RF Stock	**0.997**	0.159	-0.345	0.188
XGBoost Stock	0.773	**0.254**	**-0.273**	**0.233**
RF Bond	0.540	-0.024	**-0.115**	-0.302
XGBoost Bond	**0.615**	**0.091**	-0.148	**-0.186**

Table 5.1: Prediction performance (R^2-Score) of the regressor models for the Stock and Bond index (best values in bold).

on all three data sets. Regardless of the inclusion or exclusion of 2020, the test set did not yield a positive R^2-score.



(a) RF Stock

(b) XGBoost Stock

(c) RF Bond

(d) XGBoost Bond

Figure 5.1: Prediction results for the regressor models on the test set.

5.1.2 Prediction Classifiers

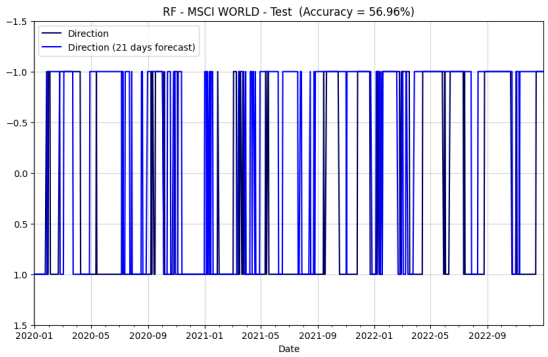
Studying classifier models was not considered until the latter parts of the project, when the initial regressor prediction models did not perform as well as initially hoped, but the classifiers quickly showed promising results. The classifier prediction models were used to predict asset trend for the upcoming month, and the result from backtesting on the test set can be seen in Figure 5.2, together with performance metrics on the various data sets in Table 5.2.

CLASSIFIER

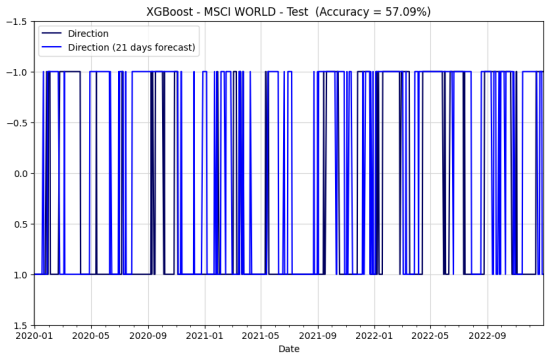
Model	Train	Validation	Test	Test (2021-23)
RF Stock	98.84%	73.06%	56.96%	58.93%
XGBoost Stock	**99.05%**	**74.69%**	**57.09%**	**62.96%**
RF Bond	96.11%	51.58%	**45.59%**	**47.79%**
XGBoost Bond	**98.04%**	**62.13%**	42.15%	40.50%

Table 5.2: Prediction performance (Accuracy) of the classifier models for the Stock and Bond index (best values in bold).

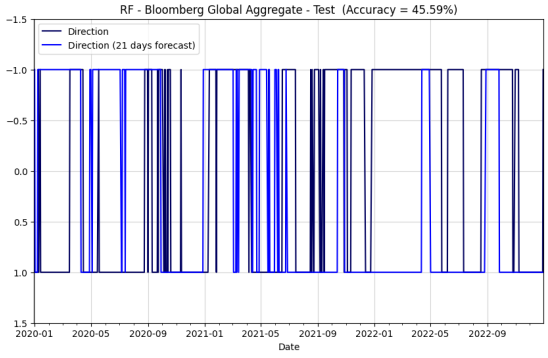
The same patterns that occurred for the regressors were seen here as well. The XGBoost model was still the most accurate when predicting stocks, achieving an accuracy of 62.96% on the test set, discarding 2020. Bonds proved difficult to forecast again, with both XGBoost and Random Forest scoring an accuracy below 50%.



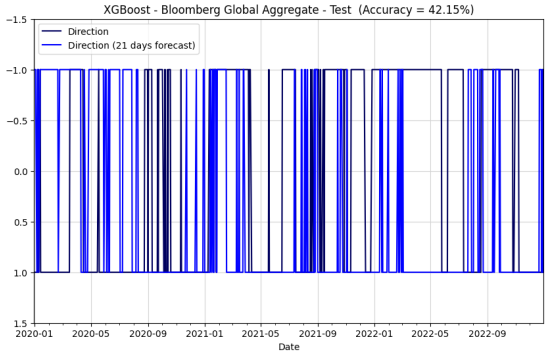
(a) RF Stock



(b) XGBoost Stock



(c) RF Bond

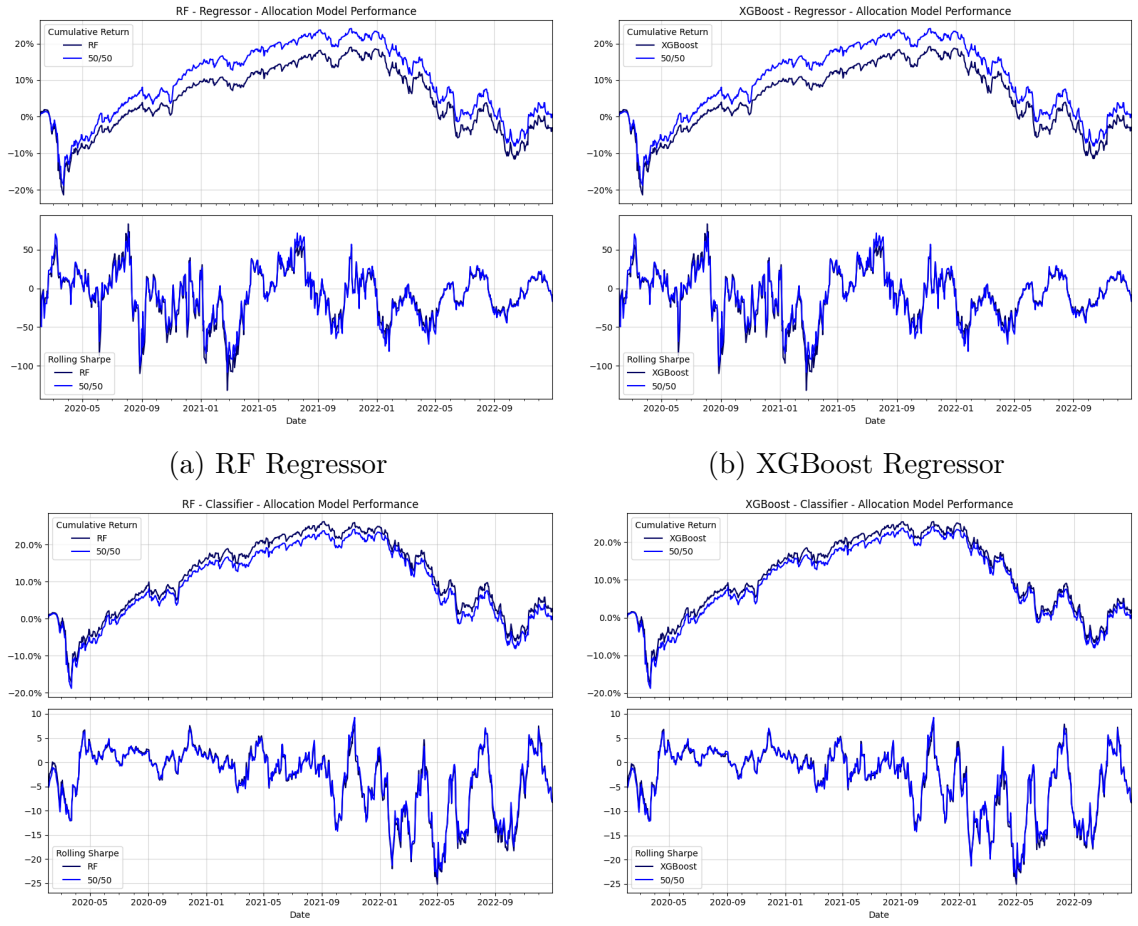


(d) XGBoost Bond

Figure 5.2: Prediction results for the classifier models on the test set.

5.2 Allocation Models

Two types of rule-based allocation models were created. The resulting allocation models can be seen in Figure 5.3, along with the various portfolio metrics in Table 5.3.



(a) RF Regressor

(b) XGBoost Regressor

(c) RF Classifier

(d) XGBoost Classifier

Figure 5.3: Allocation results for the final models.

Model	Total Return	Alpha	Volatility	Sharpe	Sortino	MDD
50/50	0.49%	-	0.77%	-1.65	-2.51	-32.16%
RF (reg)	-3.24%	-3.72%	0.84%	-5.97	-8.91	**-30.62%**
XGBoost (reg)	-3.10%	-3.59%	0.84%	-5.80	-8.66	-30.66%
RF (clf)	**2.53%**	**2.05%**	**0.73%**	**1.06**	**1.62**	-32.21%
XGBoost (clf)	1.84%	1.36%	0.76%	0.11	0.17	-32.08%

Table 5.3: Allocation performance of the studied models (best values in bold).

5.2.1 Allocation Regressors

The total return for the 50/50 benchmark model was 0.49%, with a Sharpe and Sortino ratio of -1.65 and -2.51, respectively. But the regressor allocation models only achieved a total return of -3.24% and -3.10% for the Random Forest and XGBoost models, respectively. In addition, the Sharpe and Sortino ratios of the models were significantly lower compared to the benchmark as well.

Although the continuous nature of risk-adjusted returns would have made it possible to work with thresholds to decide the portfolio weights between neutral (±0%), single (±5%) or double overweight (±10%), the difference between the Stock and Bond index, as illustrated in Figure 5.4 made it difficult to find a threshold that would generalise well, especially in less volatile periods outside of the test set. Instead, the rule-based allocation regressor model only used double-overweight depending on which asset had the highest predicted risk-adjusted return.

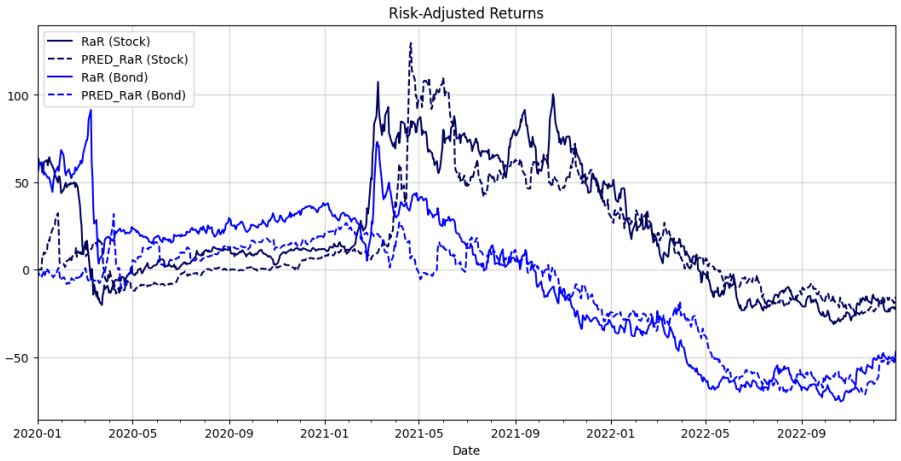


Figure 5.4: Predicted Risk-Adjusted Returns (PRED_RaR) from the XGBoost regressor and realised Risk-Adjusted Returns (RaR) from the Stock and Bond index. PRED_RaR has been a core part of the regressor version of the rule-based allocation model.

5.2.2 Allocation Classifiers

Due to the poor accuracy of the Bond classifier prediction models, the rule-based allocation strategy only utilised the signals from the Stock prediction model.

As a consequence of using only one signal, the allocation model was limited to only using double overweights (±10%), as opposed to single (±5%) or neutral overweights (±0%), which is otherwise a possibility within the Nordea allocation mandate.

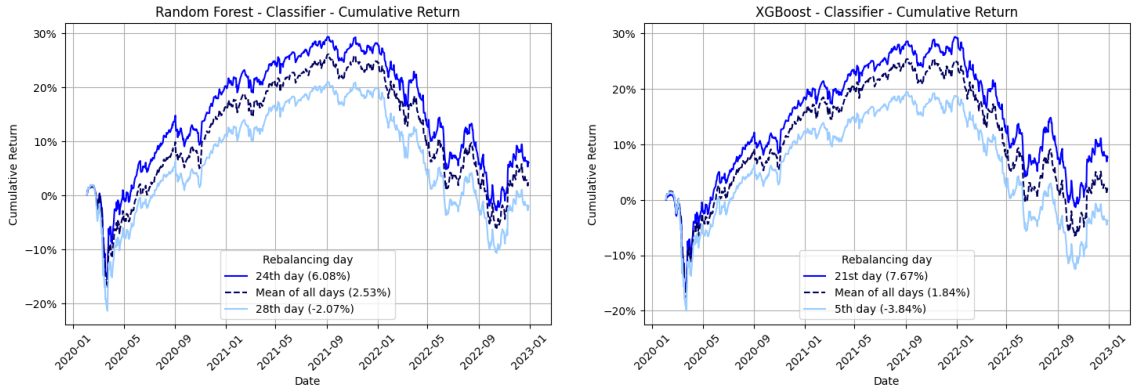
Despite these limitations, the Random Forest classifier allocation model outperformed the benchmark by more than 2% while simultaneously decreasing risk, resulting in a higher risk-adjusted return overall.

Although the XGBoost Stock prediction classifier had a better accuracy on the test set compared to the Random Forest Stock prediction classifier (Figure 5.2), the total

return of the XGBoost allocation model was only 1.84% on the backtest, compared to 2.53% for the Random Forest model. Still, the result was an improvement over the 50/50 benchmark portfolio across all performance metrics.

5.3 Rebalancing day

The choice of which day in the month to rebalance the portfolio is a crucial factor that significantly impacts the total return and is relevant to asset allocation practices in general, extending beyond the scope of financial machine learning. Figure 5.5 illustrates how the performance of the allocation classifier models is affected by the choice of rebalancing day. The figure reveals that the Random Forest model exhibits more consistent returns, while the XGBoost model shows greater variability with a higher peak (7.67% compared to 6.08%) and a lower bottom (-3.84% compared to -2.07%).



(a) Random Forest Classifier

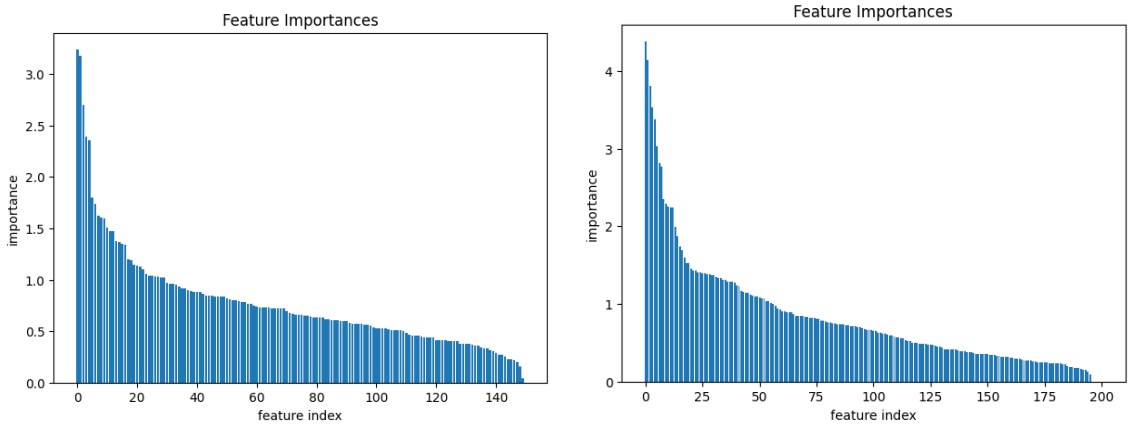
(b) XGBoost Classifier

Figure 5.5: Rebalancing day matters. The total return of the Random Forest and XGBoost allocation models can vary by 8.15 and 11.51 percentage points, respectively, depending on when the portfolio is rebalanced.

5.4 Feature Importance

The use of backtests as an industry-standard performance metric in financial machine learning has faced criticism, and the results have been labelled as pseudo-discoveries by prominent figures in quantitative forums, such as Marcos López de Prado [18]. Feature importance provides an alternative evaluation approach to a machine learning model in conjunction with historical backtests. By analysing feature importance, insights can be gained into when and which features influenced the model's performance, thereby uncovering the inner workings of the often-discussed "black box" in artificial intelligence. This analysis enables the elimination of noisy features and facilitates the assessment of performance for new combinations of time series. Consequently, feature importance was conducted prior to the backtests to inform the subsequent analysis.

The extracted features and their importance can be seen in Figure 5.6. The figure highlights the models' dependence on multiple features rather than relying heavily on a single one. When performing feature elimination, only 78 features significantly impacted the XGBoost model when forecasting the Bond index, compared to 150 features for the Random Forest model. But when predicting the Stock index, both models used the same 150 features. Furthermore, a comparison of the most important Stock features, seen in Table 5.4, and the most important Bond features, seen in Table 5.5, reveals that there was no overlap in the top features between the two models.



(a) Stock Features (b) Bond Features

Figure 5.6: Both the XGBoost and Random Forest Stock classifiers were trained on the same 150 features. The Bond classifiers, however, reached optimal performance on slightly different features, hence Figure 5.6b contains an additional 40 features compared to Figure 5.6a.

Feature & Time Series	**Importance**
Monthly change of OECD OC Composite Leading indicator	3.40%
Quarterly change of OECD US Composite Leading indicator	3.18%
1st derivative of US CLI monthly returns	2.70%
Z-Score of US Industrial Production	2.39%
2nd derivative of OECD US Composite Leading indicator monthly change	2.36%
Z-Score of OECD OC Composite Leading indicator	1.80%
1st derivative of OECD US Composite Leading indicator quarterly returns	1.74%
Z-score of quarterly changes in the AAII Investor Sentiment Survey % BULLISH	1.62%
Quarterly volatility of the Stock Index	1.61%
Value of the Stock Index	1.60%

Table 5.4: The Stock index's top ten features, combined from the two classifier models used to forecast the direction of the monthly returns.

Feature & Time Series	**Importance**
1st derivative of monthly returns on UMICH Consumer Sentiment Survey	4.38%
Z-Score difference of yearly returns on ISM non-manufacturing survey	4.14%
Z-Score difference of quarterly returns on ISM purchasing managers index	3.81%
Z-Score difference of yearly returns on Duncan Leading Indicator	3.53%
Yearly volatility on ETH Zürich Global Barometer Leading	3.38%
2nd derivative of monthly change on ETH Zürich Global Barometer Leading	3.03%
Z-Score difference on quarterly returns on ISM non-manufacturing survey	2.81%
1st derivative of monthly returns on ISM purchasing managers index	2.77%
Yearly volatility on the Bond index	2.35%
Monthly volatility on ETH Zürich Global Barometer Coincident	2.29%

Table 5.5: The Bond index's top ten features, combined from the two classifier models used to forecast the direction of the monthly returns.

6

Conclusion

This chapter discusses the results, as well as conclusions from the results. Possible future work is presented, and social and ethical aspects are touched upon.

6.1 Discussion of Results

The results can be divided into two parts, the prediction model results and the allocation model results.

6.1.1 Prediction Models

The prediction performance result metrics can be seen in Tables 5.1 and 5.2 for the regressor and the classifier, respectively.

6.1.1.1 Prediction Regressors

Amongst the Stock regressors, the XGBoost Stock model performs the best on the validation set, outperforming the Random Forest model. Outperformance continues into the out-of-sample (OOS) test set and the special COVID-excluded variant of the test set. This is in line with the initial hypothesis that XGBoost would be able to make more accurate predictions than Random Forest. However, the results differ slightly when looking at the Bond prediction models. XGBoost still manages to outperform Random Forest on the validation set but fails to generalise as well into the OOS test set, where Random Forest outperforms the XGBoost model. Although, XGBoost is once again the better predictor in the COVID-excluded test set.

Two crucial conclusions from these results should be noted. First, not a single model achieved a positive R^2-score on the test set, indicating that they are not suitable as predictors because the predictions are worse than if you were only to predict the mean. This was the main reason why classifier models were also considered. However, two of the four models, the Stock predictors, did manage to achieve a positive R^2-score on the COVID-excluded test set, perhaps indicating that in a more stable market, without once-in-a-lifetime market crashes, the models could, after all, be suitable for prediction. The second conclusion to note is that the Bond predictor models perform very poorly overall. The only positive R^2-score was by the XGBoost model on the validation set, and its R^2-score of 0.091 is still only

marginally better than the mean of zero. Most crucial is that the Bond predictor models do not generalise well at all.

6.1.1.2 Prediction Classifiers

Amongst the classifiers, the XGBoost Stock model outperforms all other models across all data sets. This includes the Random Forest Stock model, which, despite slightly underperforming the XGBoost Stock model, exhibits a negligible difference in performance. Consequently, both models can be considered suitable for prediction purposes.

Perhaps what is most impressive about the XGBoost stock model is its generalisability, maintaining an accuracy of around 60% on both test sets. While a 60% success rate may not seem significant at first glance, in the context of the model's ability to outperform the market, it translates into a high "batting average". Batting average is a term originally from baseball and refers to a statistical technique used to measure an investment manager's ability to meet or beat an index [47]. This indicates that the XGBoost model consistently predicts profitable opportunities, surpassing the market's performance more often than not. Renowned fund managers such as Peter Lynch have frequently emphasised how a 60% batting average has been instrumental in his success.

However, the bond models are not suitable for prediction, as they fail to generalise and are no better than random guessing, achieving test accuracies below 50%. This was the primary reason why the signals from these models were not used as input to the rule-based allocation model.

Unfortunately, the classifiers still show signs of overfitting, as can be seen by the dramatic decrease in accuracy from the train set to the validation and test sets.

6.1.2 Allocation Models

The allocation models result including all performance metrics can be seen in Table 5.3 and may be the most interesting results for this thesis.

6.1.2.1 Allocation Regressors

Firstly, the regressor prediction models' predictive ability (or lack thereof) is revealed, achieving a negative alpha with Sharpe and Sortino ratios lower than the benchmark. The end product is a portfolio with lower returns and higher risk compared to the 50/50 benchmark, opposite to the desired outcome. However, XGBoost seems to outperform Random Forest in both returns and risk, but the difference is small.

The choice of allocation strategy, Predicted Risk-Adjusted Returns (RaR), should also be mentioned, which was chosen as the allocation strategy together with the Nordea supervisors. The regressor allocation results suggested perhaps RaR wasn't the best-suited strategy. Still, when simulating the portfolio using actual returns with actual RaR instead of predicted, the model had an outperformance of 1.31%,

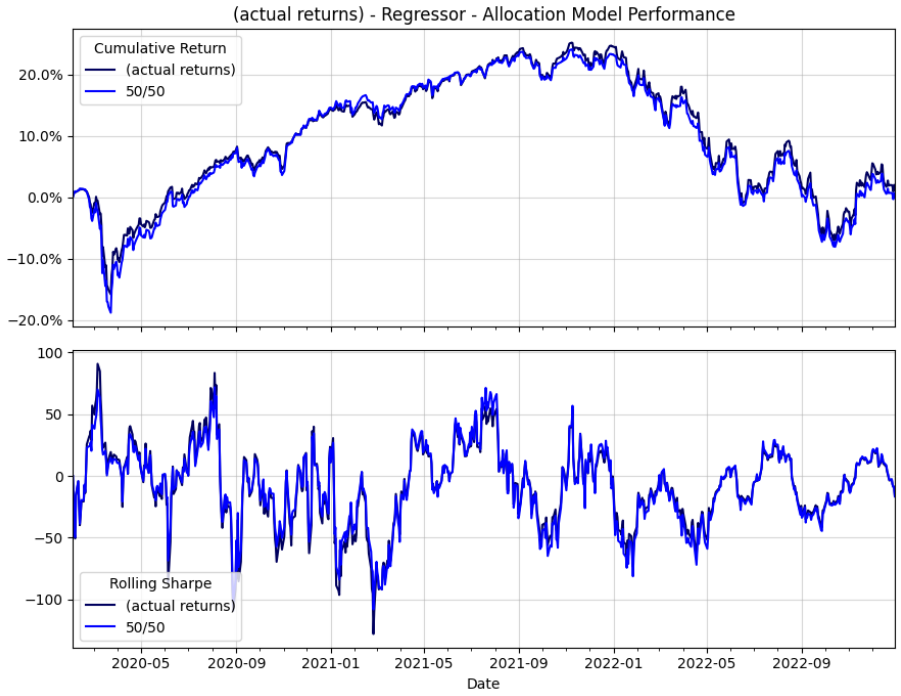


Figure 6.1: Optimal portfolio (with actual instead of predicted returns) for the Risk-Adjusted Returns (RaR) based portfolio strategy.

as shown in Figure 6.1. This suggests RaR is a valid allocation strategy and implies that if the regressor predictions were more accurate, model outperformance over the benchmark would be possible.

6.1.2.2 Allocation Classifiers

The classifiers show that it is possible to beat the benchmark of a 50/50 portfolio. Both classifiers managed to produce a portfolio with higher returns and lower risk compared to the benchmark, exactly as intended. The XGBoost classifier managed to outperform the benchmark across all metrics, while the Random Forest classifier outperformed the benchmark on all metrics except one; Maximum Drawdown.

6.1.3 Feature Importance

The feature importance analysis conducted in this thesis yields crucial insights into the leading indicators for the Stock and Bond index, providing immediate implications as decision-making tools alongside the signals generated by the developed allocation models. By uncovering the complex patterns and relationships within the data, the machine learning models provide valuable information that complements traditional economic analysis

The time series and features are presented in Table 5.4 and Table 5.5 and includes the most important indicators found in this project. One interesting result from studying these tables is how the two Stock models, which reached good predictability on the out-of-sample dataset, could do so on the same set of 150 features. The

underperforming Bond models, however, could not agree on the top features, where the XGBoost model only found significant importance in 78 features, almost half as many as in the Random Forest model. This can explain parts of the lesser results, as the patterns found by the models are rather a statistical fluke that do not generalise to unseen data.

The results from the feature importance also confirmed the hypothesis that the premium data provided by Refinitiv would give a further advantage compared to the publicly available data used in other projects.

This finding also shows the importance of not only studying the time series themselves, but also studying them from all the additional angles feature engineering provides to the data. However, to gain full advantage of the results from the feature importance, more studies are needed, which is discussed in Section 6.2

6.1.4 Conclusions

Maximum Drawdown (MDD) can be a deceiving metric without context. For the allocation models studied in this project, there seems to be a close correlation between the total return of the model and its MDD, which can be explained by the fact that the model performs better during the risk-on period compared to the other models and therefore has a bigger drawdown during the risk-off period. MDD is perhaps more insightful when two models have similar returns, but one has a significantly higher MDD than the other.

Interestingly, the final allocation results seem to contradict the initial hypothesis that XGBoost performs better than Random Forest. In the end, the Random Forest classifier achieved the largest alpha with the lowest risk (with the exception of MDD) among the two. This may be an unexpected result, since XGBoost was the better predictor. However, this can be explained by the fact that even though the Random Forest classifier guessed the wrong direction of returns more often than XGBoost, the magnitude of the predicted directions play a crucial part in the models' allocation performance.

Bear also in mind that the result is the average result of simulating the portfolio on each rebalancing day of the month. As seen in Figure 5.5, XGBoost has a higher upside than Random Forest but also a lower downside.

A possible explanation for why the Stock index was easier to predict than the Bond index could be because, during the train and validation period (2003-2020), the Stock and Bond index had a tendency for negative correlation, i.e. as stocks went down, bonds went up, but during the test period (2020-23), this tendency leaned more towards a positive correlation. The lower performance across all three data sets may indicate the need for additional time series specifically targeted towards the Bond index.

A possible explanation for why classifier models achieved a better risk-adjusted return compared to regressors could be due to the additional prediction dimension expected of regressors. It is one thing to predict the direction of the return, but

another to predict both the direction *and* magnitude of the return.

In conclusion, due to the equal overall performance between XGBoost and Random Forest, arguments for using any of the two algorithms can be found in this report. However, a more eligible way forward would be to study how the two algorithms can complement each other, for example, in a more complex ensemble model, something discussed in Section 6.2.

6.2 Future Work

After evaluating the project, several improvement opportunities have been identified.

Firstly, the Bonds prediction model could be further studied and greatly improved. Possible approaches include researching and computing even more features, performing an even more exhausting optimal hyperparameter search, and perhaps most importantly, researching new time series with a higher chance of serving as leading indicators for the Bond index. Increasing the predictive performance of the Bond prediction model would most likely improve allocation performance, especially during risk-off markets, where bonds typically serve as a safer asset class.

Secondly, more sophisticated and powerful models like Recurrent Neural Networks (RNNs) and Generative Adversarial Networks (GANs) could be investigated as potentially better prediction models. For example, RNNs can be used for stock return prediction by leveraging their ability to capture sequential dependencies in time series data. RNNs can process historical stock price data as sequential input and learn to model temporal patterns and relationships in the data. By training an RNN on a large dataset of historical stock prices, it can learn to make predictions on future stock returns based on the patterns it has learned. And GANs can be used for stock return prediction by training a generator network to generate synthetic stock price or market data. The generator could be trained to produce data that resembles real stock market dynamics, such as price fluctuations, volume patterns, and correlations. The generated data can then be used to simulate various scenarios, evaluate investment strategies, or analyse market behaviour.

Furthermore, the feature importance obtained from the machine learning models provides new topics for future studies, which were beyond the scope and timeframe of this thesis. Exploring these features can shed light on their contextual relevance, such as whether they are significant only in specific environments or extend to other asset classes. It is important to note that the investigations conducted in this thesis are by no means exhaustive on this topic, and further research is required on non-machine learning-related aspects concerning the identified leading indicators from the model.

Lastly, a third prediction model, e.g. LightGBM (a faster and more memory-efficient alternative to XGBoost), could be added to build an even more general and robust ensemble model with three strong learners, which could perform even better than the current models individually.

A common denominator for most of the previously mentioned suggestions is the need

for more powerful hardware. As mentioned in the limitations in Section 1.5, this project was severely limited in terms of the computational capacity of the machines on which the models were being trained. This was partly why the models used in this project were chosen to begin with. For future work, better hardware, using a GPU at the very least, is therefore strongly recommended.

6.3 Social and Ethical Aspects

In asset allocation, it is important to consider social and ethical aspects, as trust from customers is a critical component of business. At Nordea, this is taken one step further, as sustainability is the core of the company. Several key issues must be taken into account when considering the social and ethical aspects of asset allocation, including, but not limited to fairness, transparency, and accountability. However, this becomes even more complicated with the introduction of machine learning, with issues such as bias, privacy, and equality.

Nordea has established itself as one of the world's leading banks in terms of its commitment to sustainability, something it is transparent about in the quarterly reports [8]. This includes asset and wealth management operations that work with environmental, social, and governance (ESG) considerations. There is growing recognition of the importance of ESG considerations in asset allocation, including the need to consider the impact of investments on climate change, human rights, and other social and environmental factors. Since such considerations are not implemented in the model, it is important that strategists working with the results are aware of these issues.

Asset allocation can exacerbate existing economic inequalities if it is not done fairly and equitably. At Nordea, the model will be used to benefit all asset and wealth management customers, such as charities, unions, and private customers. Due to the size of Nordea and the organisations with which they work, this impacts society as a whole.

When developing the models, personal data was avoided to protect individuals from harm.

Bibliography

[1] J. Chen. "What is asset allocation?" (2022), [Online]. Available: https://www.investopedia.com/terms/a/assetallocation.asp (visited on 06/04/2023).

[2] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952. DOI: https://doi.org/10.1111/j.1540-6261.1952.tb01525.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1952.tb01525.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1952.tb01525.x.

[3] T. D. Kinlaw. Willliam Kritzman. Mark, *Asset Allocation, From theory to practice and beyond*. Wiley, 2021.

[4] M. Chauvet and S. Potter, "Predicting a recession: Evidence from the yield curve in the presence of structural breaks," *Economics Letters*, vol. 77, no. 2, pp. 245–253, 2002, ISSN: 0165-1765. DOI: https://doi.org/10.1016/S0165-1765(02)00128-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165176502001283.

[5] M. Chauvet and S. Potter, "Forecasting recessions using the yield curve," *Journal of Forecasting*, vol. 24, pp. 77–103, Feb. 2005. DOI: 10.2139/ssrn.274202.

[6] F. I. R. Authority. "Investing basics - risk." (2023), [Online]. Available: https://www.finra.org/investors/investing/investing-basics/risk (visited on 02/13/2023).

[7] A. Ilmanen, "Stock-bond correlations," *The Journal of Fixed Income*, vol. 13, no. 2, pp. 55–66, 2003, ISSN: 1059-8596. DOI: 10.3905/jfi.2003.319353. eprint: https://jfi.pm-research.com/content/13/2/55.full.pdf. [Online]. Available: https://jfi.pm-research.com/content/13/2/55.

[8] Nordea, *Fourth-quarter and full-year financial report 2022*, https://www.nordea.com/en/doc/interim-report-fourth-quarter-2022.pdf, Accessed: 2023-04-14, Dec. 2022.

[9] SPDR, *Spdr bloomberg global aggregate bond ucits etf (dist)*, https://www.bloomberg.com/quote/GLBL:LN, Accessed: 2023-04-14, Mar. 2023.

[10] MSCI, *Msci world index*, https://www.msci.com/World, 2022. (visited on 12/12/2022).

[11] D. K. Horter. "A brief history and outlook for traditional 60/40 investment portfolios." (2022), [Online]. Available: https://www.tfafunds.com/uploads/documents/2022-02-28-A-Brief-History-and-Outlook-for-Traditional-60-40-Investment-Portfolios.pdf (visited on 12/01/2022).

[12] D. H. Bailey, J. Borwein, M. López de Prado, A. Salehipour, and Q. J. Zhu, "Backtest overfitting in financial markets," *Automated Trader*, Feb. 2016, Forthcoming. [Online]. Available: https://ssrn.com/abstract=2731886.

[13] H. Li, Z. Yang, and T. Li, "Algorithmic trading strategy based on massive data mining," *Stanford University Stanford*, 2014.

[14] L. Khaidem, S. Saha, and S. R. Dey, *Predicting the direction of stock market prices using random forest*, 2016. DOI: 10.48550/ARXIV.1605.00003. [Online]. Available: https://arxiv.org/abs/1605.00003.

[15] M. Pinelis and D. Ruppert, "Machine learning portfolio allocation," *The Journal of Finance and Data Science*, vol. 8, pp. 35–54, 2022, ISSN: 2405-9188. DOI: https://doi.org/10.1016/j.jfds.2021.12.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405918821000155.

[16] N. Wölner-Hanssen, "Asset allocation: A machine learning strategy," *Lund University Finance Society*, 2021. [Online]. Available: https://linclund.com/wp-content/uploads/2021/05/TQR-Asset-Allocation-Random-Forest-1.pdf.

[17] K. Hongjoong, "Mean-variance portfolio optimization with stock return prediction using xgboost.," *Economic Computation & Economic Cybernetics Studies & Research*, vol. 55, no. 4, 2021.

[18] M. Lopez de Prado, *Advances in Financial Machine Learning*. Hoboken, NJ: Wiley, 2018, ISBN: 978-1-119-48208-6.

[19] M. Lopez De Prado, *Machine Learning for Asset Managers*. One Liberty Plaza, 20th Floor, New York, NY 10006, USA: Cambridge University Press, 2020, ISBN: 978-1-108-79289-9. DOI: 10.1017/9781108883658.

[20] Y. Luo, S. Wang, M.-A. Alvarez, J. Jussa, A. Wang, and G. Rohal, "Seven sins of quantitative investing," *DB Quant Handbook, Part II*, Sep. 2014.

[21] W. C. M. Arthur F. Burns, *Measuring Business Cycles*. NBER (National Bereau of Economic Research), 1946.

[22] Refinitiv, *Economic data*. [Online]. Available: https://www.refinitiv.com/en/financial-data/economic-data.

[23] T. J. Berge, "Predicting recessions with leading indicators: Model averaging and selection over the business cycle," in D. W. Bunn, Ed., Journal of Forecasting, 2014, pp. 455–471. [Online]. Available: https://www.kansascityfed.org/documents/7708/rwp13-05.pdf.

[24] J. Chen. "What is ema? how to use exponential moving average with formula." (2022), [Online]. Available: https://www.investopedia.com/terms/e/ema.asp (visited on 03/29/2023).

[25] W. F. Sharpe, "Mutual fund performance," *The Journal of Business*, vol. 39, no. 1, pp. 119–138, 1966, ISSN: 00219398, 15375374. [Online]. Available: http://www.jstor.org/stable/2351741 (visited on 03/29/2023).

[26] J. Fernando. "Sharpe ratio formula and definition with examples." (2023), [Online]. Available: https://www.investopedia.com/terms/s/sharperatio.asp (visited on 03/29/2023).

[27] W. Kenton. "Sortino ratio: Definition, formula, calculation, and example." (2023), [Online]. Available: https://www.investopedia.com/terms/s/sortinoratio.asp (visited on 05/30/2023).

[28] A. Hayes. "Maximum drawdown (mdd) defined, with formula for calculation." (2022), [Online]. Available: https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp (visited on 03/29/2023).

[29] S. Nevil. "How to calculate z-score and its meaning." (2023), [Online]. Available: https://www.investopedia.com/terms/z/zscore.asp (visited on 04/10/2023).

[30] scikit-learn, *Sklearn.ensemble.randomforestregressor*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html, 2023. (visited on 03/20/2023).

[31] D. |. D. (M. L. Common, *Dmlc xgboost extreme gradient boosting*, https://github.com/dmlc/xgboost, 2016.

[32] scikit-learn, *Sklearn.ensemble.randomforestclassifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, 2023. (visited on 05/27/2023).

[33] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. New York: Springer, 2013, ISBN: 978-1-4614-7137-0.

[34] IBM. "Decision trees." (2023), [Online]. Available: https://www.ibm.com/seen/topics/decision-trees (visited on 03/20/2023).

[35] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[36] L. Breiman, "Random forests," English, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. [Online]. Available: http://dx.doi.org/10.1023/A%3A1010933404324.

[37] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006. DOI: 10.1109/MCAS.2006.1688199.

[38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd. New York: Springer, 2009, ch. 10. Boosting and Additive Trees, pp. 337–384, ISBN: 978-0-387-84857-0.

[39] A. Pal. "Gradient boosting trees for classification: A beginners guide." (Oct. 2020), [Online]. Available: https://medium.com/swlh/gradient-boosting-trees-for-classification-a-beginners-guide-596b594a14ea (visited on 06/01/2023).

[40] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," in E. Mammen, Ed., Institute of Mathematical Statistics., 2001, ch. The Annals of Statistics, Ann. Statist. 29(5), pp. 1189–1232, ISBN: 0-201-17236-4. [Online]. Available: https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boostingmachine/10.1214/aos/1013203451.full.

[41] C. D. L. Daniel T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley, 2014, ISBN: 9781118874059. DOI: 10.1002/9781118874059.

[42] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Datamining: The ASA Data Science Journal*, vol. 15, no. 4, pp. 531–538, Aug. 2022.

[43] F. Hutter, L. Kotthoff, and J. Vanschoren, "Chapter 1: Hyperparameter optimization," in *Automated Machine Learning: Methods, Systems, Challenges*, One New York Plaza, Suite 4600: Springer Publishing Co Inc, 2019, pp. 3–38, ISBN: 978-3-030-05317-8.

[44] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012. [Online]. Available: `https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf`.

[45] scikit-learn, *Sklearn.metrics.r2_score*, `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html`, 2023. (visited on 04/04/2023).

[46] S. Nevil. "Coefficient of determination: How to calculate it and interpret the result." (2023), [Online]. Available: `https://www.investopedia.com/terms/c/coefficient-of-determination.asp` (visited on 04/04/2023).

[47] W. Kenton. "Batting average." (2022), [Online]. Available: `https://www.investopedia.com/terms/b/batting-average.asp` (visited on 06/04/2023).

A

Appendix 1

A.1 Time Series

Table A.1: This table presents the time series data used, including Stock and Bond indices, risk-premium data, interest rates, sentiment surveys, and indicators on business, geopolitical and financial conditions. The data provides a comprehensive view of the financial spectrum covering various aspects such as market performance, risk assessment, and macroeconomic indicators.

MSCI WORLD Global Stock index
Bloomberg Global Aggregate Global Bond index
MSCI WORLD U$ - PER P/E ratio for global stocks
MSCI WORLD U$ - PRCE TO 12 MTHS FE P/E ratio for global stocks based on 12m forward earnings estimates
MSCI WORLD U$ - RETURN ON EQUITY Return on equity for global stocks
MSCI WORLD U$ - Revision Breadth Earnings estimate revision momentum for global stocks
IBES MSCI WORLD Earnings Yield Global stocks earnings yield (earnings/price)
ICE BofA Global High Yield Constrained Index Global high yield bonds yield spread
ICE BofA Euro Corporate Index Euro corporate bond yield spread
ICE BofA US Corporate Index US corporate bond yield spread
JPM EMBI Global Diversified Emerging market bond index yield spread
JPM GBI Global All Traded Government bond index
JPM GBI Global All Traded - Yield Government bond index yield

US FEDERAL FUNDS TARGET RATE Reserve policy rate
US BREAK-EVEN INFLATION 10Y Market-implied inflation expectations
TRUS10T US 10 year government bond
US MONEY SUPPLY M2 CURA Growth in money supply (exkl. Central bank reserves)
US COMMERCIAL BANK ASSETS Credit growth
US AHE: ALL EMPS - TOTAL PRIVATE Number of jobs in US economy
US ALL EMPS - NONFARM INDUSTRIES TOTAL US Wage growth
US ATLANTA FED WAGE GROWTH TRACKER US Wage growth
US National Association of Home Builders Housing Market Housing market sentiment
US New Private Housing Units Started New housing construction
Duncan Leading Indicator Indicator of fixed investments
GDP NOWCAST US GDP estimator
OECD OC CLI Global leading economic indicator
OECD US CLI US leading economic indicator
US The Conference Board Leading Economic Indicators US leading business indicator
Nordea Financial Conditions Market implied financial conditions
US SENTIMENT SURVEY Sentiment survey. Degree of optimism and pessimism
BofAML Net OW Global Equities Institutional investor survey
CESI - G10 Citigroup economic surprise index
US INDUSTRIAL PRODUCTION US industrial production
Geopolitical Risk Index Geopolitical risk index
Global Barometer Coincident Global economic indicator
Global Barometer Leading

Global leading economic indicator
US EMPIRE STATE MFG SURVEY US regional business indicator
US NFIB SMALL BUSINESS OPTIMISM US business indicator
US FRB RICHMOND MFG US regional business indicator
US PHILADELPHIA FED MBOS US regional business indicator
US TOTAL MEASURE OF CEO CONFIDENCE US business indicator
US CONSUMER CONFIDENCE US consumer confidence
US Chicago Purchasing Manager Business Barometer US regional business indicator
US ISM PURCHASING MANAGERS INDEX US business indicator - manufacturing
US ISM NONMANUFACTURERS SURVEY US business indicator - non-manufacturing
BD IFO BUSINESS CLIMATE GERMANY German business indicator
US C&I LOAN SVY US bank lending indicator
NAAIM EXPOSURE INDEX Sentiment survey
US UMICH CSS US Consumer expectations