Copenhagen
Business School
HANDELSHØJSKOLEN

Master's Thesis

Applying Machine Learning in Equity Trading

The Challenge of Beating a Self-Constructed Quantitative Benchmark Strategy Using Artificial Intelligence

Authors:

Christian S. Grønager - study number: 81226

Karl J. V. Vestergaard - study number: 43862

Supervisor:

Martin C. Richter

Study programme: MSc in Mathematical Business Economics

Date of submission: 1 August 2019

Number of pages: 84 **Number of characters**: 151,406

Acknowledgement

The authors will like to express their gratitude to those who have been involved in the process. Firstly, they will thank BankInvest to provide access to the data used in this thesis and to the colleagues at BankInvest who have provided insight to quantitative investing. Additional, they will acknowledge the supervisor, Martin C. Richter, to give exceptional supervision throughout the process, and thank Innolab Capital to provide insight within the field of machine learning. Lastly, they will like to thank their family and friends for being supportive throughout the process.

Resumé

I denne kandidatafhandling har vi konstrueret en simpel kvantitativ benchmark strategi, der er baseret på ni fundamentale nøgletal, som bekskriver en virksomheds værdi. Formålet med benchmark strategien er at købe underværdiansatte aktier og short-sælge overværdiansatte aktier. For at udvælge aktierne til porteføljen udregner vi en gennemsnitlig værdi-score på baggrund af de fundamentale nøgletal. For hver industrigruppe og for hver måned fra februar 1991 til januar 2019 udvælger vi på baggrund af den beregnede værdi-score ti procent af aktierne med den højeste værdi-score og de ti procent af aktierne med den laveste værdi-score i S&P 500 Indekset. De aktier med den højeste værdi-score bliver købt, mens de aktier med den laveste værdi-score bliver shortet. Alle investeringsstrategierne i denne afhandling er konstrueret til at være markeds- og dollarneutrale.

Udfordringen ved denne afhandling er, om kunstig intelligens kan udnyttes til at konstruere en portefølje, som kan slå den førnævnte benchmark strategi. For at løse denne udfordring har vi udvalgt tre forskellige supervised machine learning-algoritmer. Disse er naïve Bayes klassifikation, support vector machines og random forest. Problemet bliver opstillet som et klassifikationsproblem, hvor vi klassificerer de 20% højeste merafkast som 1, de 20% laveste som -1 og resten som 0. Det ekstreme merafkast bruges til at træne machine learning-algoritmerne på baggrund af 50 fundamentale nøgletal.

Dataet for machine learning-algoritmerne inddeles i tre dele: et trænings-, validerings- og et testdatasæt. I testperioden opnår naïve Bayes det højeste afkast, dog har support vector machines en lavere volatilitet og opnår dermed den højeste Sharpe ratio på 0.735. Random forest og benchmark strategien har et afkast på omkring nul. En stabilitetstest viser, at benchmark strategien og random forest opnår de højeste afkast tilbage i tid. I stabilitetstesten formår random forest at slå benchmark strategien i 12 ud af 21 perioder, hvorimod naïve Bayes og support vector machines kun formår at slå benchmark strategien i henholdsvis 5 og 6 perioder.

Konklusionen på afhandlingen er, at benchmark strategien har vist konsekvent at skabe positivt afkast, dog opnår strategien højere afkast i fortiden sammenlignet med det seneste årti. Machine learning strategierne viser hver i sær forskellige resultater, der er dog ingen af dem, der konsekvent har formået at slå benchmark strategien over tid.

Contents

1 Introduction

1.1 Motivation

The efficient market hypothesis is an investment theorem that says that it is impossible to continuously maintain exceptional high return. This is because the effectivity in the market always includes all relevant information regarding the stock price. The hypothesis says that all prices trades in equilibrium, which makes it difficult for investors to buy or sell respectively under- or overvalued stocks. Moreover, the theorem states that the only way to obtain higher returns is by taking higher risks on the investments. On the other hand, active portfolio managers believe that the fundamental price sometimes deviates from the market price. This is due to the fact that humans make mistakes and have biases towards the stocks which do not cancel out in aggregate. Therefore, active investors think of the market as being inefficient.

In 1970 Eugene Fama published his article "Efficient Capital Markets: A Review of Theory And Empirical Work". In his article, he states that security markets are extremely efficient and this theory was widely accepted among academic financial economists. However, during the last decades, many financial economists and statisticians started to have less confidence in the efficient market hypothesis. They believe that fundamental value metrics and the past stock price patterns can be used to predict the future stock prices. The article "The Efficient Market Hypothesis and its Critics" (Malkiel (2003)), finds evidence that predictive patterns in stock returns can appear over time, and concludes that the perfectly efficient market does not exist.

Since the beginning of the twenty-first century, we have seen a massive increase in computer power. Gordon Moore, the founder of Intel, predicted in 1965 that the computer power would double every year. Although, he revised his prediction in 1975 to be every second year, the prediction has been valid until today ("Moore's law" (2019)). As computer power has been improved, the possibility to built more advanced computer software has increased. Also, as we have seen a growth in the global data supply, the demand for artificial intelligence (AI), where computers are learning from experience to predict the future outcome, has increased as well. Machine learning

(ML) is one of the most exciting topics when we talk about AI and is widely incorporated in the financial sector. An article from Financials Times ("Make way for the robot stock pickers" (2016)), discuss the topic of whether AI can predict the stock market and potentially replace portfolio managers. Using advanced software and computer power, portfolio managers can analyse tons of data and apply their models in many different markets. If AI replaced just 15% of the employees in asset allocation, there would be about 1,000 fewer staff in the fund management roles in the UK. Analysts believes that this would lead to lower costs for investors and more substantial profits for portfolio managers.

In the research by Huerta et al. (2013), they managed to achieve substantial excess returns using machine learning. As Huerta (ibid) only used one ML algorithm, we will throughout this thesis, investigate the performance of three different algorithms, and compare those to a self-constructed simple benchmark. The purpose of this thesis is to examine whether the algorithms can add significant value when selecting stocks based on information from companies financial statements. Furthermore, this thesis will focus on the predictive power of the algorithms, in other words, how accurate the algorithms predict extreme movements of the excess return.

1.2 Thesis Statement

In this thesis, we will construct a long/short trading strategy based on the constituents in the S&P 500 Index, and the research question is whether machine learning algorithms can perform better than a self-constructed benchmark strategy, based on publicly available fundamental key-figures from financial statements. To answer the research question, the following sub questions will be investigated:

- To what extent is it possible, by the use of fundamental key-figures that describe the valuation of a company, to construct a simple long/short benchmark portfolio with a long-term positive return after transaction costs and short-selling fees?
- To what extent is machine learning algorithms able to select stocks and construct a portfolio

that performs better than a self-constructed benchmark strategy after transaction costs and short-selling fees?

- Which connections are there between predictive power and returns for the machine learning algorithms?

1.3 Limitations

1.3.1 Investment Universe

We have limited our investment universe to consist of companies which are part of the S&P 500 Index every month from February 1991 to January 2019. Furthermore, to construct an investment strategy, we are considering the public announced fundamental key-figures from companies quarterly or annually financial statements. The start date for the investment universe is set to 28 February 1991, due to poor data quality before that date.

1.3.2 Investment Strategy

Our investment strategy is limited to a long/short strategy. We are always having the same amount of capital in each of the positions, and thereby the portfolios are dollar-neutral. When selecting stocks, we are choosing the 10% highest and lowest ranked stocks according to a created score. Furthermore, we are rebalancing the portfolios each month and holds the stocks for one month. Additional, we are keeping the portfolios industry group neutral, and each month we are investing, there has to be a minimum of 15 stocks in the industry groups.

1.3.3 Machine Learning Algorithms

There is a wide range of different ML algorithms. The following three algorithms are used and analysed in this thesis:

- Naïve Bayes Classifier (NB)
- Random Forest (RF)
- Support Vector Machines (SVM)

These algorithms are chosen to get a various range of methods in order to predict the stock market. The Naïve Bayes Classifier is a probability model, Random Forest is a tree-based model, and SVM is a model that finds a hyperplane that distinctly classifies the data points. These algorithms are widely used in practice as they have shown excellent performance and is very different from each other in the way they are modelling the data. The algorithms will be described further in chapter 3.

1.3.4 Data Limitations

It is possible to get prices and fundamental key-figures from many data sources such as Bloomberg, Compustat, Datastream, FactSet, etcetera. For this thesis, we have got access to FactSet, which we are using as our primary data source throughout this thesis.

1.4 Related Work

The topic of using ML to predict stock prices has spread widely, and the number of research papers has grown during the last decades. When dealing with ML, two major prediction problems are considered, namely, regression and classification.

The research paper by Shen and Zhang (2012) applied ML algorithms as a regression problem to predict the next day stock trend. They used the correlation between the markets closing prices that stop trading right before or at the beginning of US markets. They reached high numerical results and accuracies around 75% on the NASDAQ-, S&P 500- and Dow Jones Industrial Average Index by the use of SVM.

Huang et al. (2002) used SVM as a classification problem. They compared SVM to several other classification methods, by testing the accuracy on the prediction of the financial movement direction on the NIKKEI 225 Index. They conclude that SVM outperforms the other ML algorithms such as Linear Discriminant Analysis and Neural Networks.

A study that is very similar to the problem of this thesis is the research by Huerta et al. (2013) that seeks to explore whether features such as financial statements and historical prices can predict stock returns. Huerta et al. are scoring each stock, and on behalf of that score, they train an ML model and construct a long/short portfolio. The classifier for the training data is constructed based on the highest and lowest volatility-adjusted price changes. To predict the stock movements, they use SVM. The algorithm was trained each month to adjust for shifting market conditions. Additionally, they separate the data into eight sectors. The best performing model was structured to hold the stocks for three months, and costs were not considered. This strategy reached an annual return of 15% and volatility less than 8% by investing in the 25% highest/lowest scored stocks in the long and short position.

1.5 Structure of this Thesis

The structure of this thesis is divided into six chapters with the purpose to end with a conclusion that answer the research questions stated in this chapter. A short description of the chapters is listed as follows:

Chapter 2 - Conceptual Framework: This chapter seeks to inform the reader about the essential topics of the financial market and methods used to construct a long/short investment strategy. Additional, we explain the use of artificial intelligence and why businesses are starting to have more focus on this topic.

Chapter 3 – Machine Learning: The third chapter will give a brief introduction into the different machine learning methods and the difficulty in finding a proper model. Furthermore, a more theoretical review of the algorithms that will be used throughout this thesis is presented. Lastly, performance evaluation techniques are introduced.

Chapter 4 - Methodology: The fourth chapter describes how we retrieve the data and the different data mining procedures we are using in order to form our strategies. Furthermore, we are describing the composition of the portfolios and how we are calculating the profits and losses. Some commonly used key statistics are introduced in order to compare each of the portfolios.

Chapter 5 - Results: In the fifth chapter, we will present the results we have obtained throughout our analysis.

Chapter 6 - Conclusion: Finally, we will answer the research question. This is considered based on our results obtained from our analysis. Furthermore, we will state some possible future adjustments that could improve the performance of the results we have obtained in this thesis.

2 Conceptual Framework

In this chapter, we will start with an introduction to three different ways of looking at the financial stock market. Secondly, we will take a closer look at some of the trading strategies that have worked historically and the difficulties of maintaining a low-risk portfolio. As this thesis is focussing on machine learning, we will also give a short introduction to artificial intelligence

2.1 The Efficient Inefficient Markets

A widely debated question in the financial markets is whether the markets are efficient, inefficient, or a combination of those. Pedersen (2015), shortly defines the three types as follows:

- Efficient Markets Hypothesis: The idea that all prices are adjusted for all relevant information at any given time. This hypothesis was developed by Fama in 1970.
- Inefficient Markets: The idea of the inefficient markets is that investors' irrationality and behavioral biases influence the prices.
- Efficiently Inefficient Markets: The idea of the efficient inefficient markets is that the markets are inefficient but with an extent of efficiency. The competition across the investors makes the markets almost efficient, but the markets are still so inefficient that the investors can be compensated for the cost and risk they have.

To sum this up, if the markets are efficient, the market prices would always be reflected by all relevant information as soon as it comes out. Therefore, there would be no point for investors to take more risk and pay billions of dollars in fees if the markets are fully efficient. It is more logical to believe that there is some inefficiency in the markets that make it possible for active investors to outperform the markets and gain additional profits. However, when Fama (1970) describes the efficient markets, he admits that some levels of markets information are not available for the public. As an example, insider information could indicate other movements of the stock

price than the publicly available information. Other studies have found evidence for inefficient markets. Frazzini and Pedersen (2013) discovered the betting against beta (BAB) factor. The factor is constructed by holding low-beta stocks, which are leveraged to a beta of one, and short selling high-beta stocks, which are de-leverage to a beta of one. They conclude that the BAB factor produces significant positive risk-adjusted returns. Asness et al. (2013) also found evidence of inefficient markets. They focused their study on a value- and momentum factor, in markets from different countries. Individually the value and the momentum factors achieved high Sharpe ratios, and in a combination, the Sharpe ratio was improved. Momentum stocks are stocks that have shown excellent performance, typically within a year, and the idea is that the current period's winners will continue to show excellent performance in the next period. Value stocks are often considered as stocks that deviate from their fundamental value, and value investing is a long-term mean reversion strategy. Stocks that are cheap relative to their fundamental value is often dropped in price, which makes value and momentum negative correlated factors.

As the studies suggest, the markets might not be perfectly efficient, however, as discussed earlier, the markets might not be extremely inefficient either. Pedersen (2015) defines the markets as "Efficiently Inefficient", and he describes it as follows:

"Prices are pushed away from their fundamental values because of a variety of demand pressure and institutional frictions, and, although prices are kept in check by intense competition among money managers, this process leads the markets to become inefficient to an efficient extent: just inefficient enough that money managers can be compensated for their costs and risks through superior performance and just efficient enough that the rewards to money management after all cost do not encourage entry of new managers or additional capital." (Pedersen (2015), p. 4)

By that, Pedersen indicates that it is possible for active portfolio managers to outperform the markets because patterns in prices and factors exist, which makes it possible to maintain additional profits. However, there is no guarantee that a strategy will generate positive profits, but there are some strategies that empirically have shown better profits than others over extensive periods. In the next section, we will further describe different investment strategies active portfolio managers are using based on fundamental analysis.

2.2 Investment Strategies

In this thesis, we will construct a simple benchmark strategy based on a range of fundamental key-figures which are considered as valuation parameters. Moreover, we will construct three machine learning strategies, which have access to additional fundamental key-figures such as profitability, liquidity, operation efficiency, etcetera. In this section, we will describe two investment strategies, which rely on two different factors. The first one is the valuation factor, and the second is the quality factor.

2.2.1 Value Investing

Value investing is the first strategy we will look into and can be defined as a strategy that seeks to buy stocks that appears to be cheap and short selling stocks that appear to be expensive. Often stocks are cheap because investors do not rely on the company, and on the other hand, stocks with relative high prices are companies which investors have an eye for. In other words, value investing is like betting against other investors. This strategy has been widely analysed for the last 50 years. Many studies have found evidence of different value measures to gain additional profits. In the book "What Works on Wall Street", by O'Shaughnessy (2005), he tests the Price-to-Earnings (PE) ratio for large-cap stocks, on a long-only portfolio, in the period from 1951 to 2003. Every year he ranks the companies from 1-10, where 1 is low PE ratio and 10 is high PE ratio. The test showed that the portfolio consisting of the companies with the lowest PE ratio had on average the best compounded return. Furthermore, two portfolios consisting of the 50 lowest and 50 highest PE ranked stocks, showed very opposite results. The portfolio consisting of the 50 lowest PE ranked stocks had an annually compounded return of 14.5%, and the other portfolio consisting of the 50 highest ranked PE stocks had an annually compounded return of 8.3%. Additionally, the standard deviation of the return for the 50 low-PE portfolio was 27.39% and for the 50 high-PE portfolio the standard deviation was 32.05%. O'Shaughnessy concludes the analysis by giving an advice to the readers: "Avoid stocks with the highest PE ratios if you want to do well".

Often value investing shows better performance for long time horizons. Tweedy (2009) tests different value measures in the research article "What Have Worked in Investing". The research found evidence of several parameters that showed high performance and consistent increasing performance for longer holding periods. This indicates that the value strategy is a game of patience, and a value investor will typically experience prolonged periods with low performance. However, it is important to remember that there is no guarantee that the price of a stock will increase even though the stock seems to be undervalued. Pedersen (2015) points out that an investor must ask the following question when seeking for the right value stocks to invest in: "Does the stock look cheap because it is cheap or because it deserves to be cheap?"

2.2.2 Quality Investing

The next strategy we will look into is the quality strategy. The essence of quality investing is to buy stocks from companies with good management and a strong balance sheet. A good management is able to see opportunities and capitalize on them. In the fast moving global economy, it is important to keep an eye on the development of the companies. Are they focusing on new products and reinvesting in new technology? This could be a strong indicator for a good management. Additionally, companies with a strong balance sheet can withstand adverse situations or unexpected challenges. In an article from Asness et al. (2018) "Quality Minus Junk", they define quality companies as a characteristic that investors are willing to pay more than the actual price for the stock. The value and quality investment strategies are often thought of as opposite strategies since value investors seek to buy cheap stocks and quality investors seek to buy "good" stocks that deserve a higher-than-normal price. However, both strategies have performed well historically, a concept of combining the two investment strategies has also shown great performance. Warren Buffett says in Berkshire Hathaway Inc. annual report of 2008: "Whether we're talking about socks or stocks, I like buying quality merchandise when it is marked down". This concept is often referred to as "quality at a reasonable price" by investing in stocks of high quality at a discounted price.

2.3 Quantitative Investing

Quantitative investing is where most of the human interactions are left out of the portfolio constructing. This investing method uses a computer-based model to screen and evaluate multiple factors. Often the model has access to a huge database with structured data. Such data could be fundamental data, historical prices, or news sentiments. Quantitative investing is typically divided the three categories: Fundamental quantitative analysis, Statistical arbitrage, and High-frequency trading. In this thesis, we focus on the fundamental quantitative investing method. This method seeks to find systematic trends by analysing the fundamental key-figures for each company. In other words, the fundamental quantitative investing method is built upon a combination of statistical data analysis, and economic and finance theory. Discretionary traders use similar information as the fundamental quantitative trader, but the quantitative trader models the strategies into a computer algorithm in order to learn the algorithm of how to select stocks. When a model is defined, the approach can then be applied to a wide range of stocks all over the world. There are both advantages and disadvantages for quantitative investing compared to discretionary trading. A disadvantage of quantitative investing is that the algorithms cannot be tailored for certain situations, and soft information, such as phone calls and human judgment. On the other hand, some of the advantages that quantitative investment contributes, are the ability to compare a large number and variety of stocks, eliminate human biases and gives the possibility to backtest on historical data.

2.4 Short Selling

Short-selling is basically opening a position by selling it first, assuming in the future one are able to buy it back at a cheaper price. In reality, one is borrowing the stock from the broker and are selling it in the market. Therefore short-selling is betting for the price to drop. The fees for borrowing a stock can vary from nearly 0% to 50% in extreme cases, but it depends on the overall markets conditions and the demand for the stocks. In periods with crises, some stocks are hardly

available for short-selling. Typically because the demand for short-selling stocks has driven up the fees, or because some countries are not allowing for short-selling in those periods.

When constructing a portfolio, a great way to limit the risk is to combine a long and short-selling strategy. A long/short portfolio is often considered as a market-neutral portfolio. Some advantages of a market-neutral portfolio is to be able to generate positive returns in a down market and to generate returns with a lower volatility profile.

2.5 Cost Measures

In a perfectly liquid world, investors would trade on any investment idea and frequently move in or out of positions. However, in the real world, investors have to take transaction costs into account. There are several ways to measure the transaction costs, and one of them is the "effective cost" measure. When buying stocks, this cost measure is defined as the difference between the execution price and the market price before the trade started (plus commissions):

$$TC^{\$} = P^{execution} - P^{before}. \tag{2.1}$$

The execution price is the average price for all shares bought, and the price before the trade started is the mid-quote price just before started trading. When selling stocks a similar approach is applied just with an opposite sign. For example, if an investor buys stocks for 100 dollars, she would end up having stocks for less than 100 dollars after the trade. This happens due to the effect of purchasing share forces the price away from the observed price, and of course, due to the commission fees to the broker. The transaction cost varies between markets, even between similar stocks or the size of the trades. Small trades tend to have low costs, while larger trades have higher costs. Engle et al. (2012) estimated that small orders have transaction costs on about 4 basis points, and for orders that constitute over 1% of the stocks typical trading volume has an average trading cost of 27 basis points. Therefore, if investors must trade a large position of one stock, the investor could split the trade over a couple of days and as a result of this lower the transaction costs.

2.6 Portfolio Construction and Risk Management

Active investors work hard to construct an optimal portfolio. However, there are some common principles most portfolio managers are using to obtain a robust portfolio:

- The positions of a portfolio must be diversified
- Reasonable position limits to eliminate cases where most of the portfolio value ends up in one position
- Consider the size of a trade and continuously resizing the position based on its potential and its risk
- Keep a reasonable low level of correlation between the positions

These statements are some of the most basic principles and are essential for a hedge fund to obtain a robust portfolio with limited risks. Pedersen (2015) states: "Hedge funds don't marry their positions and don't let their bets grow large inadvertently".

There are different ways to measure the risk of a security. The most common risk measure is the volatility, which is the standard deviation of the return. Volatility is an absolute risk measure that refers to the risk of withdrawing money at the wrong time. The portfolio manager is also interested in the portfolios' correlation with the market or another benchmark portfolio. The key principle of modern portfolio theory is the idea of diversification, i.e., to reduce the overall volatility through a combination of multiple stocks. An important component when constructing the portfolio is the covariance of all the securities. Beta measures the portfolios tendency to follow the market and is calculated by the covariance between a stock and a benchmark divided by the variance of the benchmark. If the overall portfolio is constructed to have a beta of one, it indicates, that if the return of the market or benchmark portfolio is increasing by one per cent, the return of the portfolio will everything being equal also increase by one per cent. A hedge fund often claims to be market-neutral, this means that the hedge fund does not depend whether the stock market is moving up or down. In order for this to work, it is important to have the same risk exposure in

the long and short positions. To secure the same risk exposure, one can match the beta for each position, or use beta to hedge out the market exposure. These methods will be further described in the methodology section 4.4.

2.7 Backtesting

When the strategy is defined, backtesting is a great tool to test the strategy. Backtesting is used to simulate the performance of the strategy on historical data. However, a backtest does not necessarily tell the truth about how the strategy will perform in the market today. Nevertheless, a backtest is never a bad idea, since it gives a great insight into how the strategy would have performed in the past. For example, if a backtest shows poor results, this could advise the investor to not implement the strategy and potentially spare the investor for losses. Lastly, a backtest can indicate how risky the strategy is, and potentially give ideas for improvements. To run a backtest, one must specify the following components:

- Universe: the investment universe of the stocks
- Signals: the input data, and how to analyse it
- Trading rule: a trading rule that tells when to buy and sell based on the signals, including how often the portfolio is rebalanced and the size of the positions
- Time lags: to make a reliable test, one needs to make sure that the data is used when the data was available –thereby eliminate look-ahead-bias

While performing a backtest, it is important to be aware of certain biases. The backtest results tend to look a lot better than in the real world, and several reasons could cause this to happen. First of all, the market as it was ten years ago is not the same as it is today. Second, certain data mining biases are unavoidable. When testing a strategy, the analyser always seeks to optimize the implementations towards a better result, but the changes in the implementations were not known back then. A third bias is a survivorship bias. Consider the Standard & Poor 500 Index.

If the backtest is based on the current stocks of the index, then the investment universe is biased. Companies today might not have been included in the index five or ten years ago. Creating a trading strategy without eliminating this bias, will typically generate high performance for the long position and poor performance for the short position, as the strategy only includes the surviving stocks that have performed well until today.

2.8 Artificial Intelligence

The interest of artificial intelligence (AI) has rapidly grown in the last couple of decades. The data information has continuously increased, and as everything is being digitalized, more and more data has been stored in databases. In a short explanation, AI makes it possible for machines to learn from experience to perform tasks based on those experiences. In 1997, the AI based Deep Blue chess machine managed to defeat the reigning world champion Garry Kasparov in a game of chess. Deep Blue was built as a brute-force searching machine. This means that it simulated a large range of chess games and was able to perform the best move based on all that information. Another way to use AI is in self-driven cars, which has reached significant improvement in the last couple of years. Waymo, a subsidiary of Alphabet Inc., has launched a limited trial of self-driven cars in Phoenix, Arizona ("Waymo Technology" (2019)). An example of how Waymo reacts to unforeseen events, like a jogger who passes the road without looking, is that Waymo is using its lasers to identify objects. Furthermore, Waymo is able to understand how the objects will interact in the near future and are able to make those predictions with a blink of an eye in order to avoid the object.

AI is often divided into seven categories:

- Knowledge reasoning
- Planning
- Machine Learning
- Natural language processing
- Computer vision
- Robotics
- Artificial general intelligence

The interest of this thesis is in the field of machine learning, that is the study of statistical models and computer systems to find patterns in data and predict future events. One way to apply machine learning in finance is to teach a model which stocks that have done well historically, based on a set of variables. If the model is well supervised, it is able to predict the future stock movements based on new observations. But a model will find correlations between everything, and often machine learning algorithms are referred to as a "black-box", which means that the model is so complex that humans cannot understand how the model has come to that conclusion. The next chapter will focus on the key concepts within machine learning and how to evaluate a model.

3 Machine Learning

3.1 The Essentials of Machine Learning

Artificial intelligence is nowadays almost part of our everyday life. For example, Siri and Alexa, the virtual voice assistants from Apple and Amazon, both rely on natural language generation and processing (NLP) and have the ability to have a short and understandable dialog with humans. Machine learning (ML) is also a significant part of these digital assistants, as they have access to a massive amount of data. Every time Siri or Alexa give one a wrong answer to your request they utilize the data and improves its response next time. As the available amount of data is increasing, ML has also become a central part of almost every business. ML is used to obtain as much information from the data as possible, trying to predict the future or maybe to work more efficiently. There is no reason to believe that this development will decrease, and as many people are struggling to understand what ML is, Daniel Faggella from Emerj has come up with his definition in the article "What is Machine Learning?" (2019):

"Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions."

As an example a few years ago, AlphaZero, an algorithm developed by Google, beat the world's best chess-playing computer program. The achievement was reached after the algorithm had taught itself how to play in under four hours. The difference between AlphaZero and its competitor is that AlphaZero has a machine learning approach with no human input apart from the basic rules of chess. In the beginning, AlphaZero took several random moves and lost the first many played games. Although AlphaZero learned from the previous games, and after four hours, it managed to beat the competitor algorithm. In the game, AlphaZero took an "arguably more human-like approach" in the search for moves than the competitor algorithm, that simulated many games in order to perform the best possible move ("AlphaZero AI beats champion chess program after teaching itself in four hours" (2017)).

The essentials of machine learning can be divided broadly into three parts of learning paradigms: Unsupervised, Supervised and Reinforcement learning. In the following sections, we will define the basics of the three learning paradigms.

3.1.1 Unsupervised Learning

In unsupervised learning, the algorithm tries to find a hidden structure in a complex and unlabeled dataset with multivariate relationships. Unlabeled data means that there is no "correct" way of seeing the data. In the search for a "common" structure, the algorithms use grouping as a technique to separate the data. Grouping or clustering is an excellent way to reduce the dimensionality, identify outliers, or find interesting relationships among the observations or variables. Clustering is based on similarities and distance, and the goal is to minimize the distance between the object within each cluster. Although clustering is a relatively primitive technique with no assumption of the data at the beginning, it has proven to be a helpful tool to understand the relationships in unstructured datasets. Another approach of unsupervised learning is concerned with the explanation of the variance-covariance structure among the variables. Through a few linear combinations of the p variables, much of variability can often be accounted for by a smaller number of k components. In the k components there is often as much information as in the p variables, and therefore the original dataset consisting of p variables can often be reduced to a smaller dataset consisting of k principal components. As the principal components are a linear combination of the p variables, the components are geometrically obtained from a rotation of the original variables with maximal variability and a simpler description of the covariance structure (Johnson and Wichern (2013)).

3.1.2 Supervised Learning

Supervised learning is often considered as the most common learning problem within the field of machine learning. The principle of supervised learning is that the algorithm both has information about the output variable, Y, and the input variables, X. The word "supervised" refers to a supervisor who has the correct answer, and the agent must learn from those answers. Using a

training set of observations $T = (x_i, y_i), i = 1, ...N$, the algorithm observes the values of the input and output variables to produce a function $\hat{f}$. As new inputs is observed, the algorithm utilize the function to estimate an output. The goal is to estimate a function $\hat{f}(x)$ that hold predictive information between the input and output variables (Hastie et al. (2009)).

An example of supervised learning is whether a bank should give a loan to a start-up company, in relation to the probability that the start-up will default in the near future. Using historical information about the financial condition of other start-up companies and their observed default rate, a supervised learning algorithm can predict the default rate based on current start-ups present financial conditions.

3.1.3 Reinforcement Learning

Reinforcement learning does not use historically labelled information to learn from but trains an agent on experienced information. The algorithm evaluates each step, and the goal is to maximize the reward in every situation. It is used by many software and machines to obtain the best path or the best behaviour. Reinforcement learning varies from supervised learning in a way that supervised learning utilizes labelled training data to predict the best answer, whereas the reinforcement algorithm has no data to learn from, but the agent evaluates what is best in any given task based on experience. Imagine a computer playing chess with no historical information besides the rules. Starting from scratch, the computer tries various moves and strategies to beat its opponent. After a lot of attempts, it finally improves the strategy and is able the predict the next best possible moves in each situation. As an example, AlphaZero, the previously mentioned chess-playing algorithm developed by Google, rely on reinforcement learning.

3.2 A Deeper Look into Supervised Machine Learning

In this thesis, we will focus on supervised machine learning algorithms. As discussed before, supervised learning algorithms learn from labelled data. After the algorithm has learned and

understood the data, the algorithm utilizes the patterns in the data in order to predict the output of new observations. Supervised learning can be divided into two categories, namely regression and classification. The difference between regression and classification is that regression problems predicts a numerical output based on observed inputs, whereas classification problems predicts the output label the data belongs to.

3.2.1 Regression and Classification

The primary difference in regression and classification problems is the output variable Y. In regression, the output variable $y_i \in \mathbb{R}$ is numeric, and one tries to estimate the relationship between the input, X, and the output, Y, to predict the value of new observations. An example of a regression model could be a model that describes the relationship between house prices and several variables such as the number of rooms, municipality, distance to the capital, and the distance to forest. Regression is used both in the classic statistical models but also in machine learning for algorithms such as support vector machine, classification and regression trees (CART), etcetera. To quantify how well the predictions actually match the observed data one usually uses the mean squared error (MSE), given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2, \tag{3.1}$$

where $\hat{f}(x_i)$ is the predicted observations. If the function fits the data well, the MSE will be small and high otherwise.

For classification, the output variable is categorical, $y_i \in Y = \{0, 1, 2, ..., g\}$, where g corresponds to the number of classes. Here one tries to classify the observations into predefined categories. The output can also be based on the likelihood that the observation belongs to the respective category. For example, a spam detector must estimate whether the email is spam or not. In this case, the output variable can be 1 (spam) or 0 (no-spam), but also a likelihood for the events. Instead of using the MSE to measure the accuracy of the estimated function, the training error rate (TER) is more appropriate for the classification problem:

$$\text{TER} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i). \tag{3.2}$$

The TER measures the proportion of misclassifications of the predicted $\hat{y}_i$. If the indicator function $I(y_i \neq \hat{y}_i) = 0$ then the observation is classified correctly, and otherwise it is a misclassification. Figure 3.1 shows an example of the regression and classification problem. In the right-hand side of

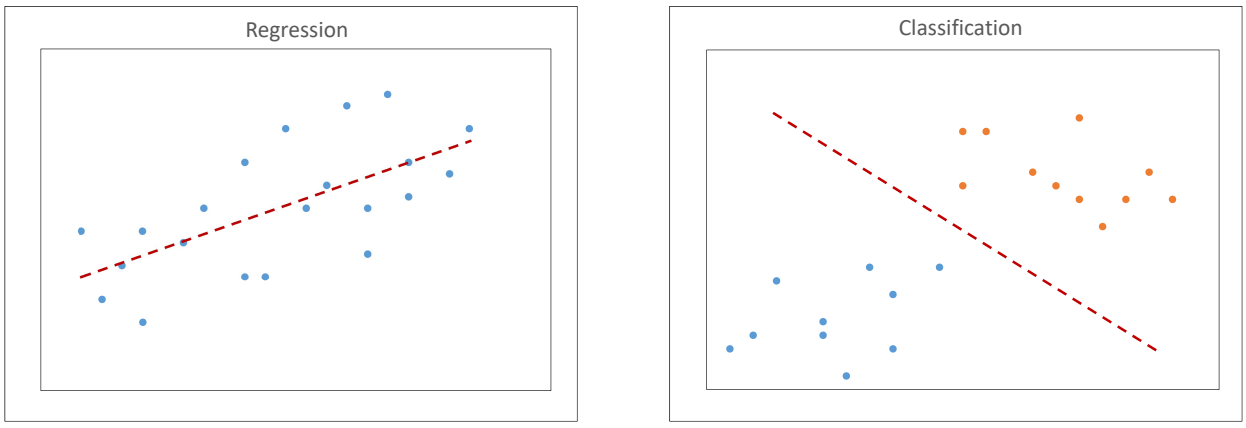


Figure 3.1: Regression vs. classification.

the figure, a perfect separating decision boundary is drawn. That is not always the case, as errors of prediction can occur since data often is noisy. Additional, the decision boundary does not have to be linear but can appear in many shapes.

3.2.2 Bias and Variance Trade-off

In general, we would like to have as little bias and variance as possible. However, those measures are opposing effects, and one cannot lower the bias without increasing the variance. In order to find the optimal balance between bias and variance, one evaluates several models in order to find the best parameters for the model. As an example, one sometimes splits the dataset into two parts: a training and test set. When evaluating how a model build on the training set performs both on the training and test set, one wants the prediction error to be as low as possible. If the model has a low prediction error on the training set, but a high prediction error on the test set, it is said that the model has high variance, and thus is overfitting the data. On the other hand, if the model has a high prediction error on both the training and test set, it is said that the model has high bias, and thus underfitting the data. In figure 3.2, the prediction error for a training and

test dataset is compared to the complexity of the model. The figure shows that if the model based on the training data is highly complex, the prediction error will tend to be low on the training data. In other words, the model will typically be overfitted and therefore not fit the test data very well, causing a higher prediction error for the test set. The goal is to find the optimal solution,

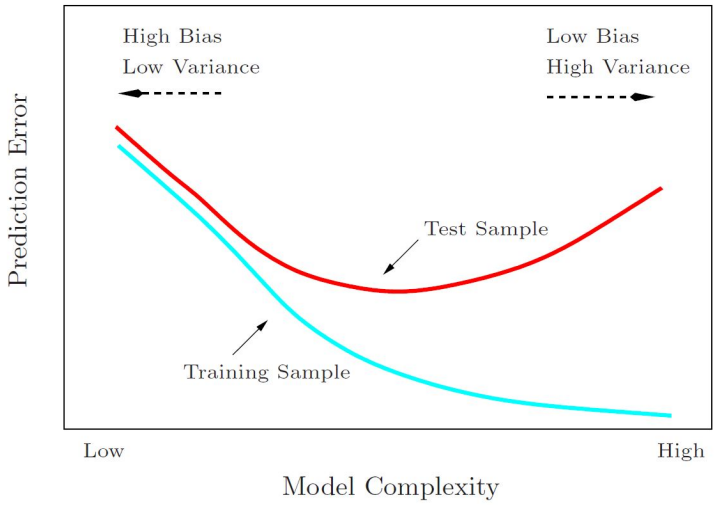


Figure 3.2: Prediction error of training and test data as a function of model complexity (Hastie et al. (2009), p. 38).

and that is a trade-off between the bias and variance. There are several ways to adjust the bias and variance. Most algorithms have parameters that regulate the complexity of the model. For example, in a simple CART model, the variance can be reduced by using fewer nodes in the tree or increased by adding more nodes. This process is often referred to as "hyperparameter tuning" in the literature, an is an essential part of the model evaluation phase.

3.2.3 Performance Measures for Classification

When modelling with a supervised machine learning algorithm, it is possible to obtain the precision of the predictions. This is very convenient since we want to find the best possible model based on a range of parameters. This thesis concerns a classification problem and there are several methods which can give an understanding of how well the model performs. Among those are the ROC curve, F-measure, G-mean, etcetera. However, in this section, we will introduce the confusion

matrix in a two-dimensional case, while give an example of a three-dimensional case, as we are using the elements of the confusion matrix to hyper-parameter tune the ML models.

Confusion Matrix

The confusion matrix shows how many of the observations which are correctly classified or misclassified. In the two-dimensional case the matrix is a 2×2 grid, and looks as follows:

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True Positive (TP)	False Positive (FP)
Predicted Negative Class	False Negative (FN)	True Negative (TN)

Table 3.1: Confusion matrix.

The first column of the confusion matrix represents the actual positive observations, while the second column represents the actual negative observations. To find the distribution of the actual positive and negative labelled observations, one can take the sum of each of the columns and compare it to the total number of observations. The most common measure from the confusion matrix is the accuracy or its reverse, the prediction error:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.3}$$

$$\text{Prediction error} = 1 - Accuracy \tag{3.4}$$

The accuracy gives the overall hit ratio of the model, and if the data is perfectly separable one wants to maximize the accuracy. However, most of the times, the data is not perfectly separable, and a higher overall accuracy could also lead to more false negative or false positive classifications. Therefore it is often convenient to know how many observations that are correctly classified or misclassified in the different states of the confusion matrix:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}, \tag{3.5}$$

$$\text{True Negative Rate (TNR)} = \frac{TN}{TN + FP}, \tag{3.6}$$

$$\text{False Negative Rate (FNR)} = \frac{FN}{TP + FN}, \tag{3.7}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{TN + FP}, \tag{3.8}$$

$$\text{Positive Predicted Value} = \frac{TP}{TP + FP}. \tag{3.9}$$

Equation (3.5) is also called sensitivity and measures how many of the actual positives that is classified correctly, where equation (3.6) is called specificity and measures the misclassification of the actual positives. Equation (3.9) measures the distribution of the actual positive class relative the predicted positive class and is also called the precision.

In the following example, the confusion matrix is extended to be a 3×3 grid. Two models that detect the accuracy for an investment strategy is evaluated. The classes are generated to be -1 for the 20% lowest returns, 1 for the 20% highest returns, and 0 otherwise. Among 100 stocks the models have predicted which of the stocks that belong to the different classes. Model 1 in table 3.2 has the best overall accuracy of 70%, while model 2 in table 3.3 has an accuracy of 64%.

	-1	**0**	**1**
-1	5	0	5
0	10	60	10
1	5	0	5

	class -1	**class 0**	**class 1**
sen	0.25	1	0.25
ppv	0.50	0.75	0.50

Table 3.2: Model 1 with 70% accuracy.

	-1	**0**	**1**
-1	12	10	0
0	8	40	8
1	0	10	12

	class -1	**class 0**	**class 1**
sen	0.60	0.67	0.60
ppv	0.55	0.71	0.55

Table 3.3: Model 2 with 64% accuracy.

By evaluating the models based on accuracy, model 1 seems to be a great model. However, when examining the sensitivity and the precision of the models, the conclusion is different. Model 1 has predicted a lot of the observations to be 0, which result in higher sensitivity and precision for that class. Although model 2 has lower accuracy, the sensitivity that measures how many of the actual

positives that are classified correct is much higher for the extreme classes. Moreover, the precision is also increased. Therefore, it indicates that model 2 is a better choice for predicting extreme returns.

3.3 Supervised Machine Learning Algorithms

In this section, we will describe the theory of the machine learning algorithm we are using to model different investment strategies. The algorithms we are focusing on are the Naïve Bayes Classifier, Random Forest and Support Vector Machine. These algorithms are supervised learning algorithms, and for this thesis, we are using them as a classification problem.

3.3.1 Naïve Bayes Classifier

The naïve Bayes (NB) classifier is a simple and fast algorithm based on Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{3.10}$$

Bayes' theorem says that the best way to find the probability of A given B is to find the probability of how many times A occurred with B out of all the times in which B occurred. Furthermore, Bayes' theorem applies the naïve independence assumption within the distributions of the variables (Lantz (2015)).

To implement NB, a training dataset is required. The training data must contain a matrix of m independent variables $\mathbf{X}^T = (\mathbf{x}_1^T, ..., \mathbf{x}_m^T)$ and a classification vector $\mathbf{y}$ with g possible classes. According to the assumption of independence between the variables $\mathbf{x}$, the possibility to classify y given $\mathbf{x}$ is:

$$P(y|\mathbf{x}_i) = P(y)\prod_{i=1}^{m} P(\mathbf{x}_i|y) \tag{3.11}$$

This tells us that NB considers each of the variables regardless of the class of y. An example could be to classify a person as male or female, based on the person's height, fat percentage, length of hair, and length of beard. One could imagine that there exists a negative correlation between the

length of the hair and the length of the beard. The NB will generate a probability for classifying the person as male, independent of the correlation between the variables. As the independence between the variables is ignored the class density estimates may be biased, but the bias does not have a huge impact on the posterior probabilities, especially not near the decision regions.

In a multivariate classification problem with k possible classes, we can obtain the maximum probability of the class $\hat{y}$ by the following equation:

$$\hat{y} = argmax_y P(y) \prod_{i=1}^{m} P(\mathbf{x}_i|y). \tag{3.12}$$

This function estimates the class $\hat{y}$ given the predictors. Although the rather naïve assumption, the NB often tends to outperform more advanced supervised machine learning algorithms (Hastie et al. (2009)).

3.3.2 Tree Based Models

The random forest (RF) algorithm was introduced by Breiman in 2001, as a further development of the decision tree. The algorithm produces a large number of de-correlated trees with the purpose of reducing the variance, which is represented in each of the individual decision trees (Hastie et al. (2009)). As each tree in the RF model contributes to the final model, we will start with an introduction to the original classification and regression tree (CART) model. The CART algorithm can be used as a regression and classification problem. However, as the focus of this thesis is within classification, we will limit the description of the CART to a classification problem.

The Underlying Decision Tree

The way to implement tree-based methods is to split the features by a threshold and fit the model based on these splits. When constructing a decision tree, three types of nodes are used: the root node, which is the top node of the tree, the internal nodes, which extent the branch, and the leaf node, which is the end of the branch.

Consider at dataset D with N observations, input variables $\mathbf{x}_i \in \mathbb{R}^p, \forall i \in N$ with p dimensions, and an output variable $y_i \in Y = \{0, 1, 2, ..., g\}$. The dataset D will continuously be dividend

into M nodes, $D^m \in D, m = \{1, ..., M\}$ with subsets of observations of the dataset. Each node is constructed by calculating the impurity of the node which can be done using various methods, such as misclassification error, cross-entropy, or the Gini index. The Gini index is used in the original definition of the CART model, and it indicates how many observations that are misclassified if it was classified random according to the distribution of classes in the respective node. The formula for the Gini index is:

$$\text{Gini index} = \sum_{g=1}^{G} \hat{p}_{mg}(1 - \hat{p}_{mg}), \tag{3.13}$$

where $\hat{p}_{mg}$ is the proportion of the g class in node m, and is calculated by:

$$\hat{p}_{mg} = \frac{1}{N_m} \sum_{\mathbf{x}_t \in R_m} I(y_i = g). \tag{3.14}$$

The variable x_i with the lowest impurity is selected in the root node and divides D into two internal nodes, D^2 and D^3, by a threshold t_m. Then the impurity subset D^2 is measured and compared to the remaining input variables. The variable with the lowest impurity is selected as a new internal node for the branch, and the same process is done for this node, etcetera. If an internal node has the lowest impurity compared to the remaining variables, it will be changed to a leaf node and ends the branch. The decision tree is complete, when all branches have reached an end node. The class of a new observation $\mathbf{x}_j$ is predicted by starting at the root node D^1, and evaluate each node to its threshold, by the following constraints:

$$L_{j,t} = \mathbf{x}_j \leq t_m$$

$$R_{j,t} = \mathbf{x}_j > t_m.$$

When a leaf node is reached, the prediction of the new observation $\hat{\mathbf{x}}_j$ is reached. However, it is easier said than done. When modelling decision trees, the size of the tree plays an important part of whether the model shows good or bad performance. A tree that contains many nodes tend to overfit the model, which could lead to a low accuracy on the test dataset. On the other hand, if the tree only consists of a few nodes, some structure of the data can potentially be left out and lead to underfitting of the model. To prevent too many or too few nodes in the model, a stopping parameter is often defined, which typically is a minimum number of observations in each leaf. One

of the biggest concerns with trees is the high variance. Small changes in the data can lead to significant changes in the way the tree is divided, which makes it hard to predict noisy data.

Random Forest

Random forest (RF) is an extension of the CART model, and benefits from the use of the bagging (bootstrap aggregation) technique. The idea of bagging is to reduce the variance within the trees, and the essentials of bagging is to find a suitable prediction from many noisy but approximately unbiased trees (Hastie et al. (2009)). For the classification problem, the RF built a committee of trees, which each have a vote for the outcome of the prediction.

The RF model consists of B identically distributed and de-correlated trees, where each tree $b \in B$ are built on bootstrapped samples Z_b of the training set. An RF tree T_b is constructed for each bootstrapped dataset Z_b. Each tree is constructed using the same technique as described in the previous section. When all trees T_b are constructed, a voting system to predict new observations $\mathbf{x}_j$ is created by the following formula:

$$\hat{C}_{rf}^{B}(\mathbf{x}) = majority\ vote\{\hat{C}_b(\mathbf{x})\}_1^B,$$

where $\hat{C}_b(\mathbf{x})$ is the class prediction of the bth RF tree. As the trees generated in the RF model is identically distributed, the expectations for each tree are the same, as is the bias for each tree. Hereby, the only improvement from the CART to the RF is through the reduction of the variance which is defined as $\frac{1}{B} \cdot \sigma^2$. To improve the variance reduction in RF, the tree-growing process is made through random selection of the input variables. Before each split, the algorithm selects $m \leq p$ random input variables, where m typically is equal to $\sqrt{p}$. Reducing m will typically reduce the correlation of each trees in the RF model, and hence reduce the variance (Hastie et al. (2009)).

The RF algorithm includes several parameters, among them are number of trees to grow, number of variables randomly sampled as candidates for each split, sample size to draw from the population, node size of each terminal node, maximum number of terminal nodes in the forest, etcetera. In practice, the default value for those parameters might not generate the best fit for the model. It is therefore necessary to tune these parameters, to find the best combination based on

different performance measures.

When the number of relevant variables is small relative for the total number of variables, RF tend to perform poorly for small m, since the change of getting a relevant variable in each split will be small. However, the hyper-geometric probability of getting at least one relevant variable in each split is calculated by the following formula:

$$1 - P(X = 0) = \frac{\binom{r}{y}\binom{p-r}{m-y}}{\binom{p}{m}}, \tag{3.15}$$

where r is the number of relevant variables, y is the number of observed successes, p is the total number of variables, and m is the number of variables to include in each split. If the dataset consists of $p = 45$ variables, where only 5 of those variables has a significant influence of describing the outputs and the rest of the variables are noisy, the probability of getting a relevant variables in each split is 59%, assuming $m = \sqrt{45} \approx 7$. This indicates that RF is relative robust and feature selection is only necessary in cases with hundreds of noisy variables and few relevant variables.

Another feature with RF is the variable importance plot. For every split in the tree, each variable is evaluated, and the sum of the Gini decrease for the chosen variable is accumulated across every tree in the forest. To give an average of the Gini decrease for every variable, the sum is divided by the number of trees in the forest. This is a great tool to analyse as it gives an idea of the variables impact on the outputs. In contrast, variable with low impact might be omitted from the model, to make it simpler and faster to fit and predict.

3.3.3 Support Vector Machines

The support vector machines (SVM) was introduced in 1995 by Cortes and Vapnik. It was introduced as a linear classification method with soft margin hyperplanes. The overall idea is to create a hyperplane, that can separate the observations into classes or at least try to separate them if possible. The optimal separating hyperplane is defined as a function that separates classes and maximizes the distance from each of the classes by its closest observations, also called support vectors. The boundaries that the hyperplane creates are used to classify new observations. There are two types of SVM problems, one separates the data perfectly, and the other is a non-separable

problem (Hastie et al. (2009)). These types are further described in the next sections.

The separable case

The definition of the separable two classifier problem, is to create a hyperplane function, such that all observations from one class is on one side of the hyperplane and all the observations associated with the second class is on the other side of the hyperplane.

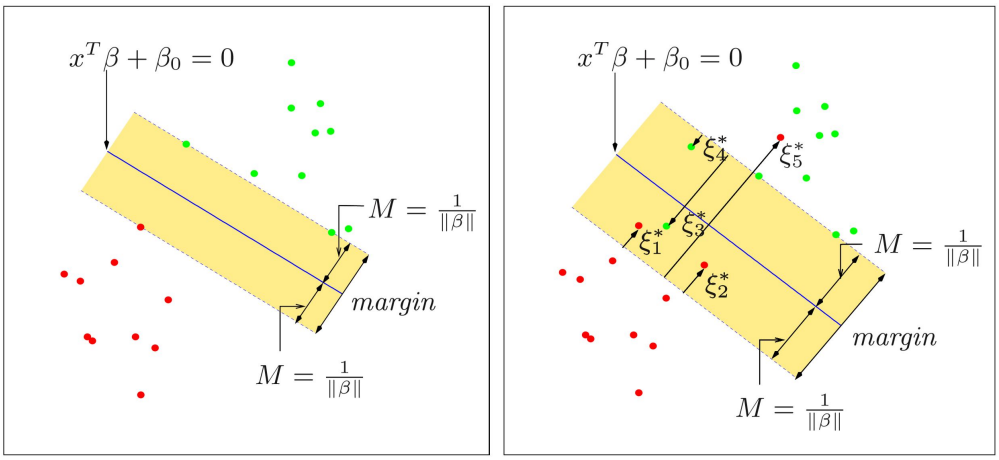


Figure 3.3: Separable vs. non-separable case, where ξ_i represents the misclassified observations (Hastie et al. (2009), p. 418).

Consider a dataset with N observations and two classes, where $\mathbf{x}_i \in \mathbb{R}^p$ is the input variables and $y_i \in Y = \{-1, 1\}$ is the response or output variable. The decision boundary function is defined as:

$$f(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\beta} + \beta_0 = 0 \tag{3.16}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is a unit vector: $||\boldsymbol{\beta}|| = 1$. The separating hyperplanes is defined as a margin around the function $f(\mathbf{x})$ with a minimum width of M on both sites, where $M = \frac{1}{||\boldsymbol{\beta}||}$. The observations on the edge of the margin are called "support vectors". The function $f(\mathbf{x})$ is seen as a boundary condition for all new observations $\mathbf{x}_j$, and on behalf of the boundary it classifies the observation

as -1/1 under the following condition:

$$\hat{y}_j = \begin{cases} -1 & \text{if } \hat{\mathbf{x}}^T\boldsymbol{\beta} + \beta_0 < 0 \\ 1 & \text{if } \hat{\mathbf{x}}^T\boldsymbol{\beta} + \beta_0 > 0 \end{cases}.$$

To ensure that all observations are at least M distance away from the decision boundary $f(\mathbf{x})$, the following condition have to be maximized:

$$\begin{aligned} &\max_{\boldsymbol{\beta},\beta_0,||\boldsymbol{\beta}||=1} M \\ &\text{subject to } y_i(\mathbf{x}^T\boldsymbol{\beta} + \beta_0) \geq M, i = 1, ..., N. \end{aligned} \tag{3.17}$$

This can be rewritten as a convex optimization problem, which is done by removing the constraint, $||\boldsymbol{\beta}|| = 1$ and replacing the condition in (3.17) with:

$$\frac{1}{||\boldsymbol{\beta}||} y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) \geq M \Rightarrow y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) \geq M||\boldsymbol{\beta}||,$$

and since all $\boldsymbol{\beta}$ and β_0 fulfill these inequalities at any positive scaled multiplier, it is possible to set $||\boldsymbol{\beta}||=\frac{1}{M}$. That gives the possibility to rewrite the condition from (3.17) as a minimization problem:

$$\begin{aligned} &\min_{\boldsymbol{\beta},\beta_0,} ||\boldsymbol{\beta}||^2 \\ &\text{subject to } y_i(\mathbf{x}^T\boldsymbol{\beta} + \beta_0) \geq 1, i = 1, ..., N. \end{aligned} \tag{3.18}$$

The constraint defines a margin around the linear decision boundary with a width of $\frac{1}{||\boldsymbol{\beta}||}$, thereby $\boldsymbol{\beta}$ and β_0 is chosen to maximize the width. These changes make it possible to set up the following Lagrange (primal) function and hereby minimize this problem with respect to $\boldsymbol{\beta}$ and β_0:

$$L_P = \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^{N} \alpha_i[y_i(\mathbf{x}^T\boldsymbol{\beta} + \beta_0) - 1]. \tag{3.19}$$

Moreover, by setting the derivatives of $\boldsymbol{\beta}$ and β_0 equal to zero, we obtain:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Leftrightarrow \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = \boldsymbol{\beta} \tag{3.20}$$

$$\frac{\partial L_P}{\partial \beta_0} = 0 \Leftrightarrow \sum_{i=1}^{N} \alpha_i y_i = 0, \tag{3.21}$$

and by substituting these result into (3.18) we get the following Lagrange (dual) function:

$$L_D = \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$
$$\text{subject to } \alpha_i \geq 0 \wedge \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{3.22}$$

The optimal solution is then obtained by maximizing L_D for $\alpha_i \geq 0$, by complying with its conditions that give a more simple convex optimization problem. Furthermore, this solution must satisfy Karush-Kuhn-Tucker (KKT) conditions, which include (3.20), (3.21), (3.22) and

$$\alpha_i[y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) - 1] = 0 \forall i. \tag{3.23}$$

From these conditions we observe that if $\alpha_i > 0$ then $y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) = 1$ which means that $\mathbf{x}_i$ is on the boundary of the hyperplane margin. But if $y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) > 1$ then $\mathbf{x}_i$ will not appear on the hyperplane margin, and $\alpha_i = 0$. From equation (3.20), the solutions for $\boldsymbol{\beta}$ are defined as a linear combination of the support vector points $\mathbf{x}_i$ and lies on the boundary of the hyperplane margin when $\alpha_i > 0$. β_0 is obtained with help of (3.23) for any of the support vector points $\mathbf{x}_i$. The optimal hyperplane is defined as the following decision boundary function, $\hat{f}(\mathbf{x}) = \mathbf{x}^T \hat{\boldsymbol{\beta}} + \hat{\beta}_0$ and this function is used to classify new observations $\mathbf{x}_j$.

The described case is for linear separable data, however, it is generally not possible to obtain perfectly separated data. To allow for misclassifications in the decision boundary, a new function is introduced. This leads to the next section, where the soft margin approach will be presented.

The non-separable case

As seen in the separable case, we were able to define a function $f(\mathbf{x})$ that satisfied the following constraint $y_i f(\mathbf{x}_i) > 0, \forall i$. However, this is not possible in the non-separable case. In this case, it is necessary to change the approach by allowing the observations to be on the wrong side of the decision boundary which is illustrated in the right-hand side of figure (3.3). The new approach is called soft margin support vector and introduces a slack variable ξ_i for each observation. This allows to change the constraint from (3.17) to:

$$y_i(\mathbf{x}_i^T \boldsymbol{\beta} + \beta_0) \geq M(1 - \xi_i), \tag{3.24}$$

where $\xi_i \geq 0, \forall i$ and $\sum_{i=1}^{N} \xi_i \leq C$. The error for misclassified observations, is measured as the distance towards the hyperplane margin, and thereby makes it possible to solve it as a convex optimization problem. With that said, a misclassfication will occur when $\xi_i > 1$. To obtain the optimal solution we use the same approach as for the separable case. At first, we rewrite the maximization problem into a minimization problem and the constrains for the mathematical optimization problem is written as:

$$\begin{gathered} \min_{\boldsymbol{\beta},\beta_0,} ||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{N} \xi_i \\ \text{subject to } \xi_i \geq y_i(\mathbf{x}^T\boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \forall i. \end{gathered} \tag{3.25}$$

The cost parameter C, defines the complexity of the model. It determines the amount of overlapping classes and tells how simple the model is. As in the separable case, the solutions to the non-separable problem is found through the Lagrange function. However, instead of just considering $\boldsymbol{\beta}$ and β_0, this case concerns three parameters $\boldsymbol{\beta}, \beta_0$ and ξ_i. The Lagrange function is defined as:

$$L_P = \frac{1}{2}||\boldsymbol{\beta}||^2 + \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i[y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{N} \mu_i\xi_i. \tag{3.26}$$

By setting the derivatives of $\boldsymbol{\beta}, \beta_0$ equal zero, we obtain the same results as in the separable case (3.20) and (3.21). Additionally, by setting the derivative of ξ_i equal zero we get:

$$\frac{\partial L_P}{\partial \xi_i} = 0 \Leftrightarrow \alpha_i = C - \mu_i, \forall i. \tag{3.27}$$

Furthermore, a constraint of $\alpha_i, \xi_i, \mu_i \geq 0, \forall i$ is added, which makes it possible to obtain the Lagrangian (dual) objective function by substituting (3.20),(3.21) and (3.27) into (3.26), and obtain the equation:

$$\begin{gathered} L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ \text{subject to } 0 \leq \alpha_i \leq C \wedge \sum_{i=1}^{N} \alpha_i y_i = 0. \end{gathered} \tag{3.28}$$

The optimal solution can be obtained by maximizing the Lagrangian (dual) function. As earlier the solution has to comply with equation (3.20),(3.21) and (3.27) and the KKT conditions, by

including the following constraints:

$$\alpha_i[y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1 - \xi_i)] = 0, \tag{3.29}$$

$$\mu_i\xi = 0, \tag{3.30}$$

$$y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1 - \xi_i) \geq 0, \tag{3.31}$$

for $i = 1, ..., N$. Hereby, an unique solution of the (primal) and (dual) problem can be constructed. From equation (3.20), the optimal solution for $\boldsymbol{\beta}$ is found as:

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N} \hat{\alpha}_i y_i \mathbf{x}_i \tag{3.32}$$

and with $\hat{\alpha}_i > 0$ for observations $\mathbf{x}_i$ that meets the constraint of (3.31) given the constraint of (3.29) is complied. These observations are the support vectors and have $\hat{\xi}_i = 0$ and are characterized by $0 < \hat{\alpha}_i < C$, the rest of the observations have $\hat{\xi}_i > 0$ and $\hat{\alpha}_i = C$. From constraint (3.29) any of the support vectors can be used to solve β_0, and the average of the solutions is used to obtain $\hat{\beta}_0$. Based on the results of $\hat{\boldsymbol{\beta}}$ and $\hat{\beta}_0$ the final decision boundary function is:

$$\hat{f}(\mathbf{x}) = \mathbf{x}^T\hat{\boldsymbol{\beta}} + \hat{\beta}_0 = \mathbf{x}^T \sum_{i=1}^{N} \hat{\alpha}_i y_i \mathbf{x}_i + \hat{\beta}_0. \tag{3.33}$$

Both the separable and the non-separable case find a linear decision boundary function, but as the complexity in data increases, a linear decision boundary might not be the right choice. In the next section we will address this problem.

Kernel functions

The SVM is a combination of the classifier described in the previous section and a kernel function. The kernel function is used when the input variables cannot be linear separable, and makes the decision boundary more flexible by using basis expansions, and to transform the data into higher dimensions which have a clearer insight of the data.

We start by defining the m'th input space as $h_m(\mathbf{x}_i), m = 1, ..., M$, which produces a nonlinear function. Since $h(\mathbf{x}_i)$ is only represented in the inner function of the Lagrange function, we do not

need to specify the transformation of $h(\mathbf{x}_i)$, but all we need to know os the kernel function:

$$K(\mathbf{x}, \mathbf{z}) = \langle h(\mathbf{x}), h(\mathbf{z}) \rangle. \tag{3.34}$$

The kernel computes the inner product of the transformed space. Thereby, the Lagrange (dual) function is stated as follows:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle h(\mathbf{x}_i), h(\mathbf{z}_j) \rangle. \tag{3.35}$$

The decision boundary can be found in the same way as earlier, and is defined by:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(\mathbf{x}, \mathbf{x}_i) + \hat{\beta}_0, \tag{3.36}$$

and at all times, the kernel function must be a symmetric positive (semi-) definite function.

There are several types of kernels and the three most frequently used kernels according to Hastie et al. (2009) are:

$$d\text{'th-Degree polynomial} : K(\mathbf{x}, \mathbf{z}) = (1 + \langle \mathbf{x}, \mathbf{z} \rangle)^d, \tag{3.37}$$

$$\text{Radial basis} : K(\mathbf{x}, \mathbf{z}) = exp(-\gamma ||\mathbf{x} - \mathbf{z}||^2), \tag{3.38}$$

$$\text{Neural network} : K(\mathbf{x}, \mathbf{z}) = tanh(\kappa_1 \langle \mathbf{x}, \mathbf{z} \rangle + \kappa_2). \tag{3.39}$$

When applying kernels to the SVM some additional parameters, other than C, must be defined. For example in the Radial basis kernel, an extra parameter, γ, is introduced and determines the strength of the support vectors. When creating a SVM model it is unknown which value of the parameters that gives the best results. Therefore, it is important to tune the model by trying multiple combination of the parameters to obtain the best classifier.

4 Methodology

In this chapter, we will present the data analysis from the start of how we are collecting the data to the final construction of the investment strategies. At the beginning of this chapter, a brief description of the dataset variables will be introduced. Additionally a data analysis process diagram will give an overview of how the data analysis is structured. At the end of this chapter, we will describe how the portfolio turnover i is calculated and introduce several basic financial key statistics we are using to evaluate the performance of the portfolios.

4.1 Dataset Description

The dataset we are using in this thesis consists of a time series for each company which are part of the S&P 500 Index from 28 February 1991 to 31 January 2019. Additionally, a range of variables that hold information about the company are included. A subset of the variables are listed below:

- **Date**: the last day in each month
- **CUSIP**: number that identifies most financial instruments
- **Ticker**: a symbol used to uniquely identify traded shares of a particular security
- **Company name**: the name of the company who issues the shares
- **Industry Group**: a class based on the GICS classification system
- **Industry Group name**: the name associated with the GICS classifier
- **Price_SSD**: the share price for the company corrected for splits, spinoffs and dividends
- **Price_SS**: the share price for the company corrected for splits and spinoffs

In addition to the above variables, the dataset consists of 46 fundamental key-figures. A lot of the fundamental key-figures are ratios of two or more fundamentals, such as “Enterprise value

divided by Sales". These key-figures hold information that contributes to the financial or economic well-being and the subsequent financial valuation of a company. In the following section, we will describe how we got access to the data, structured it in a database, divided it into industry groups, and prepared it for the different investment strategies.

4.1.1 Data Analysis Process Diagram

In figure 4.1, the data analysis process diagram is shown, which has the purpose of presenting the process of collecting, transforming, cleaning, and modelling the data we are using in our analysis. As shown, we have separated the process into five steps. In the first step, we are collecting the data and prepare it for the analysis. In step two, we are modelling and cleaning the data. Step three is divided into two parts. Part A concerns the constructing of the simple benchmark strategy, while in part B, we are using machine learning to model several investment strategies. In step four, we are identifying the stocks to invest in, and in step six, we are testing the performance of the investment strategies.

4.2 Data Preparation

Data collection and excellent data quality are essential parts of doing a practical project. We have retrieved the data used in the thesis from two separate sources. To construct the investment universe, we have used Compustat North America, and to download the fundamental key-figures, GICS codes, stock prices, and reporting dates we have used FactSet Research System Inc.

To ensure the best data quality as possible, we have spent an exceptional amount of time considering how we should collect the data in the best and smartest way. Much time has also been used to chat with the support from FactSet, in order to learn how to work with the system in the most efficient way. In the following sections, we will discuss how we retrieved the data, constructed the investment universe, and prepared the data for the modelling phase.

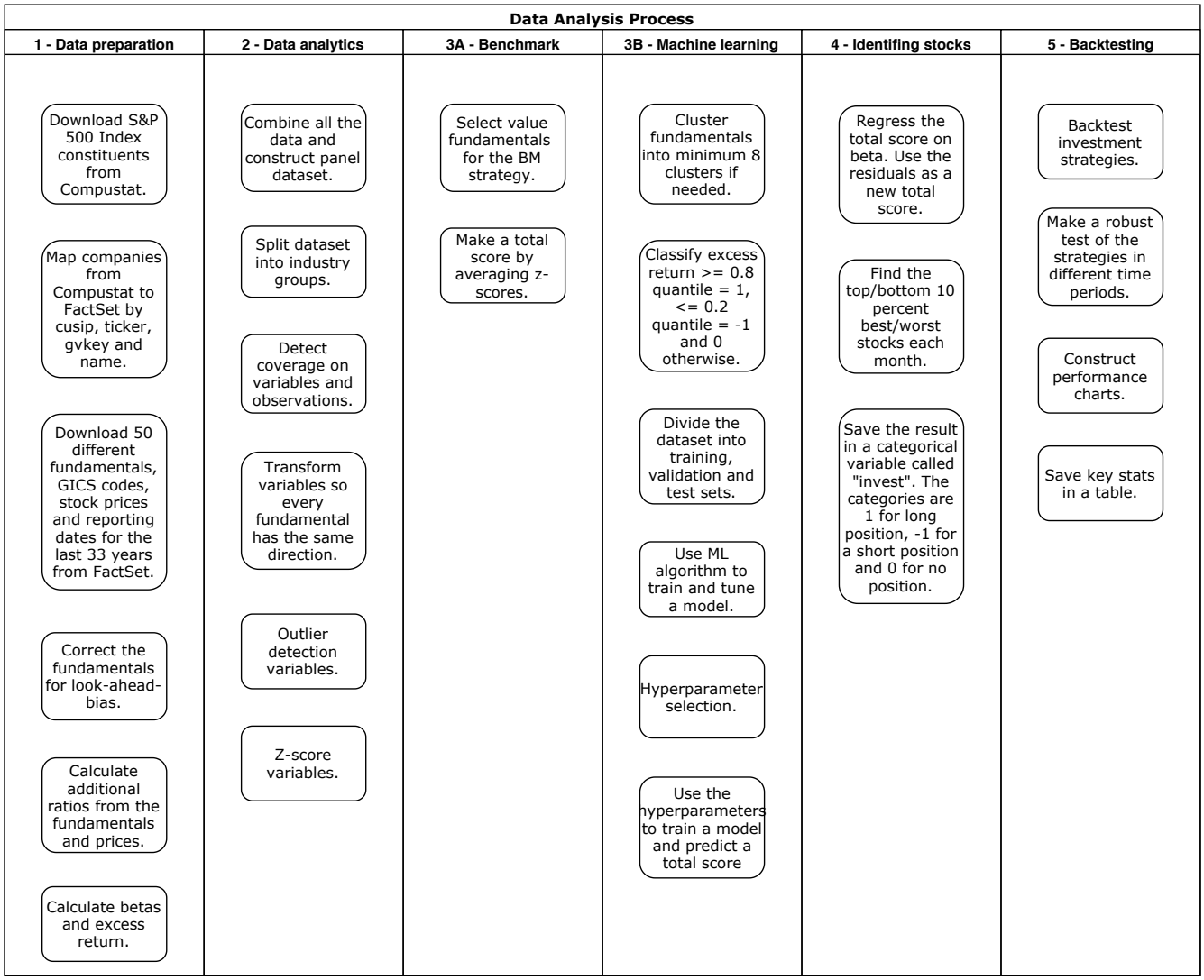


Figure 4.1: Data Analysis Process Diagram.

4.2.1 Data Providers

In this section, we will give a brief introduction to the data providers we have used for the analysis.

Compustat North America

To construct the investment universe, we have got access to Compustat North America. Compustat is a database of U.S. and Canadian fundamental and market information on active and inactive

publicly held companies. From Compustat we have retrieved the S&P 500 Index constituents. The access to Compustat is obtained via CBS since CBS has an agreement with Wharton Research Data Services[1] (WRDS). WRDS is a platform with multiple databases that provides the user access to data across multiple disciplines, including finance, marketing, and economics.

FactSet Research Systems Inc.

FactSet is a U.S. financial data and software company that offers access to a broad range of financial data and provides analytical applications. FactSet is used by a vast number of investment banks and hedge funds all over the world ("FactSet" (2019)). FactSet collects annual and quarterly financial statement data, per share data, derived ratios, stock prices, and business segments, which has made FactSet the most valued data source in our thesis. The Danish investment bank BankInvest have in agreement with FactSet provided us access to FactSet during this thesis.

4.2.2 Data Collection

The S&P 500 Index usually consists of 500 stocks. A list of the stocks which are part of the current index can be found on the webpage "https://www.slickcharts.com/sp500", but the constituents changes now and then because the companies market capitalization changes as well. If we are constructing a long-only portfolio that solely consists of the current stocks in the index, a backtest of that strategy would potentially overperform the market as we would only have included the most successful stocks until today. This bias is called a survivorship bias and is important to eliminate when comparing past performances against the index. It is also important when we are using machine learning to predict stock return based fundamental key-figures, that the machine learning algorithm has learned from all different types of companies financial situation. To eliminate this bias and construct the investment universe, we must find all the stocks which have been part of the index every month from the starting date to the ending date of the analysis.

[1]WRDS is the leading, comprehensive, internet-based data research service used by academic, government, non-profit institutions, and corporate firms "Welcome to WRDS!" (2019)

Investment Universe

In order to build the investment universe, we have retrieved index constituents from Compustat. In Compustat, we made a query regarding which stocks that have been part of the S&P 500 Index in the time frame of 31 December 1985 to 31 January 2019. In order to find information about the companies in FactSet, the query also contained company identification information such as the company name, ticker symbol, and CUSIP code.

iid	from	thru	co_conm	co_tic	co_cusip
1	1985-12-31	1994-07-12	UNITED CONTINENTAL HLDGS INC	UAL	902549500
4	2015-09-03		UNITED CONTINENTAL HLDGS INC	UAL	910047109
1	1985-12-31	1999-01-03	FOOT LOCKER INC	FL	344849104
1	2016-04-04		FOOT LOCKER INC	FL	344849104
1	2001-01-16	2009-03-26	NOBLE CORP PLC	NE	G65431101
1	2011-01-18	2015-07-19	NOBLE CORP PLC	NE	G65431101
4	2004-12-20		TWENTY-FIRST CENTURY FOX INC	FOXA	90130A101
1	2015-09-21		TWENTY-FIRST CENTURY FOX INC	FOXA	90130A200

Table 4.1: Raw constituents data from Compustat.

Seen in the table 4.1, the companies are part of the index in different periods. From 21 September 2015, Twenty-First Century Fox Inc. has two types of stocks which both are part of the index at the same time. The ticker symbols are the same, but the CUSIP codes are different. Further investigation shows that there is an error in the Compustat database. The ticker symbol of the stock with CUSIP code "90130A200" should have been "FOX" instead of "FOXA", hence that stock is the B stock-class of the company. A blank cell in the table indicates that the company was still part of the S&P 500 Index when data was retrieved.

As we only will be using monthly data, we have added two extra columns to the dataset. In the first column, we changed the dates of the column "from" to be the last day in the current month. A similar procedure was made for the column "thru", but in this case, we changed the date to be the last day in the previous month. The reason why we did this was due to the fact that we must

construct a dataset where we only have information about the companies ultimo each month. If a company is part of the index up to and including 26 March 2009, the last date it has been part of the index on a monthly basis is 28 February 2009.

The next part of the process was to construct a matrix, where the rows represent the number of stocks in the index and the columns represent the months in which the stocks were a part of the index. By doing so, we were able to count how many stocks that were part of the index each month. A slice of the matrix is shown in table 4.2.

	1985-12-31	**1986-01-31**	**1986-02-28**	**1986-03-31**	**1986-04-30**	**1986-05-31**	**...**
1	002824100	002824100	002824100	002824100	002824100	002824100	...
2	438516106	438516106	438516106	438516106	438516106	438516106	...
3	025537101	025537101	025537101	025537101	025537101	025537101	...
4	097023105	097023105	097023105	097023105	097023105	097023105	...
5	110122108	110122108	110122108	110122108	110122108	110122108	...
6	134429109	134429109	134429109	134429109	134429109	134429109	...
7	149123101	149123101	149123101	149123101	149123101	149123101	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.2: Matrix of constituents in the S&P 500 Index.

Technically, the S&P 500 Index is an index consisting of the 500 largest companies on the New York Stock Exchange (NYSE) and NASDAQ Stock Market, weighted by market capitalization. In figure 4.2 a chart shows that the typical amount of stocks included in the index is 500, but for two months there are 499 stocks in the index –the reason for this is unknown for us. In the period from 2014 to 2019, there is a small increase in stocks included in the index. The reason for this is that some companies listed in the index have issued multiple types of common stocks, and they are both traded enough to be part of the index. However, not all forms of common stock for each company are included. Instead, S&P picks and chooses which companies will have more than one class of common stock included, and the potential exists for the number of stocks in the S&P 500 Index to continue growing in the future ("How many stocks are in the S&P 500?" (2015)). The five companies who have two classes of common stocks in the index on 31 January

2019 are, Twenty-First Century Fox Inc., Under Armour Inc., Alphabet Inc., Discovery Inc., and News Corp.

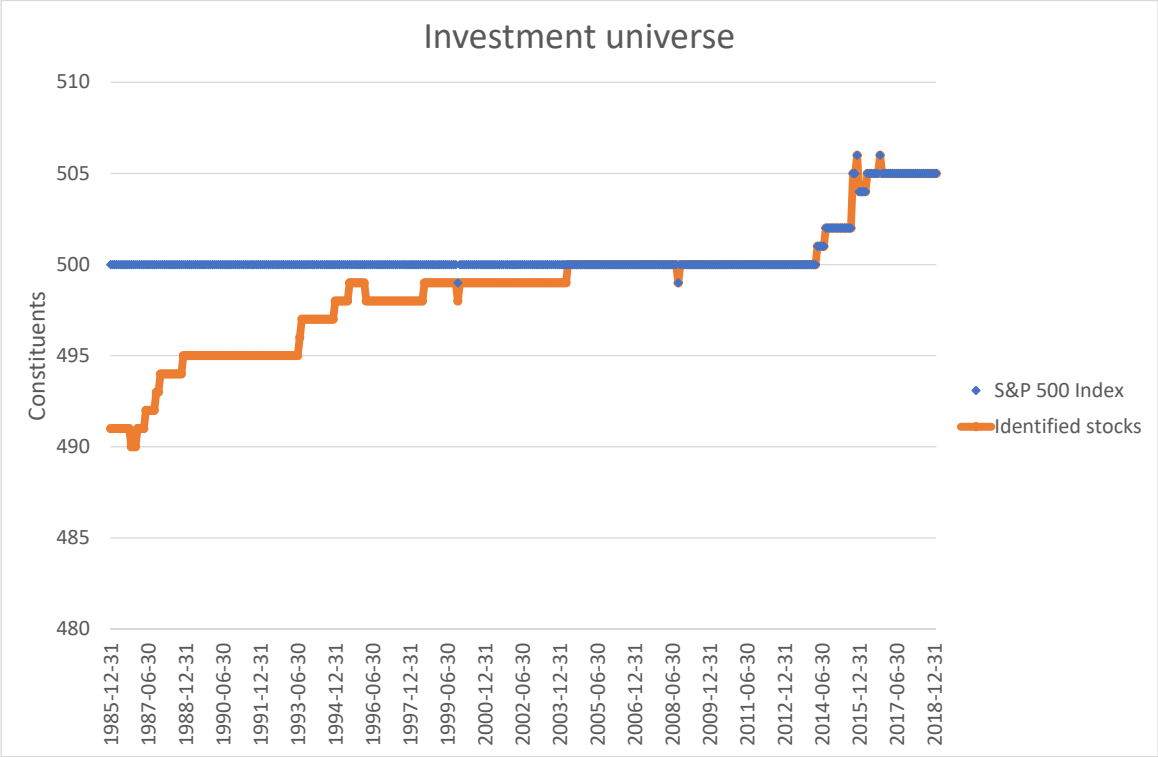


Figure 4.2: S&P 500 Index constituents vs. identified stocks.

From the matrix in table 4.2, a new dataset that includes the date, CUSIP code, ticker symbol, and company name was constructed. The dataset consists of 1278 different stocks and holds information about the constituents of the S&P 500 Index from 31 December 1985 to 31 January 2019. In order to retrieve the fundamental key-figures, we had to identify the stocks in FactSet. First, we manually looked up the CUSIP codes in FactSet, and by doing so, we were able to identify most of the stocks. However, the CUSIP code for 26 stocks was not recognized by FactSet. To identify these stocks, we looked up the ticker symbol and company name, and this lets us identify 17 more stocks. The last nine stocks were not to find in the FactSet database. Most of these stocks were part of the S&P 500 Index in the end of the 1980s and early 1990s. A search on the

web showed that most of these companies were merged, acquired, or defaulted, and therefore, we excluded these companies from the investment universe. A chart that shows the proportion of the identified stocks is showed in figure 4.2.

To retrieve the data from FactSet, we have used FactSet's Application Programming Interface (API), which makes it possible to retrieve data from FactSet's databases within Excel, using FactSet's self-programmed Excel formulas. In order to spend as less time as possible on the data retrieving, we have used Excel's programming language Visual Basic for Application (VBA), to write a macro which automatically retrieved the data and saved it in Excel files on a virtual storage cloud.

Fundamental Key-Figures

The variables we are using for the investment strategies are publicly available fundamental key-figures from financial statements. In a correspondence with FactSet's online support, there is a list with the 50 most commonly used fundamental key-figures in FactSet. In appendix A.1 there is a list of the fundamental key-figures. These fundamental key-figures are e.g., Current Assets, Earnings Per Share, Return on Equity, etcetera, and are grouped in the following eight different categories:

- Asset Turnover Analysis
- Coverage
- Leverage
- Liquidity
- Operating Efficiency
- Per Share
- Profitability
- Valuation

Using FactSet's API for Excel, we downloaded the fundamental key-figures for all the companies in our investment universe. We had to make six Excel-files for every key-figure because Excel could not handle all the data in one workbook. The key-figures was downloaded on a quarterly basis. However, if quarterly key-figures were not available, we downloaded annual key-figures. If no

key-figures were available on a specific day, it was left blank.

Look-Ahead-Bias

In FactSet, the fundamental key-figures are listed starting from the fiscal ending date and not on the actual reporting date, where the information is available for investors. That means as Apple Inc.'s fiscal ending date is 30 September, Apple Inc. will publish the financial report two or three months after this date. This bias is called a look-ahead-bias, and one has to correct for this when backtesting an investment strategy, to ensure one only uses information when it was available. If the bias is not taking into account, one will generate an investment strategy with information about the future, and potentially experience higher profits as the stock price does not reflect the publicly known fundamental value.

To correct for the look-ahead-bias, we had to download the reporting date of the financial statement. FactSet's support suggested using the formula "FF_EPS_RPT_DATE" to get the reporting date, and manually displace the key-figures. First, we matched the reporting dates quarterly. However, if the quarterly reporting dates were not available, we matched the reporting dates on an annual basis. If neither of the reporting dates were available, a conservative alternative of a three-month lag was used.

Prices

In order to calculate the total returns of the stocks, we are using historical prices, which are split, spinoff, and dividend adjusted. It is especially essential to correct for stock splits. If a company performs a 1:2 stock split, the price would be half the price from one day to another, and that would have a significant impact on the total return of the share.

As we discussed earlier, we must correct the fundamental key-figures for look-ahead-bias. However, as some of the key-figures are price ratios, we decided to calculate the ratios manually. The ratios are Earning per Share, Book Value per Share, Sales per Share, Cash Flow per Share, and Free Cash Flow per Share, each divided by the price. The price we are using to create the ratios

Date	Fiscal period	Date	Published
2018-02-28	9.7300	2018-02-28	9.7300
2018-03-31	10.3600	2018-03-31	9.7300
2018-04-30	10.3600	2018-04-30	9.7300
2018-05-31	10.3600	2018-05-31	10.3600
2018-06-30	11.0300	2018-06-30	10.3600
2018-07-31	11.0300	2018-07-31	11.0300
2018-08-31	11.0300	2018-08-31	11.0300
2018-09-30	11.9100	2018-09-30	11.0300
2018-10-31	11.9100	2018-10-31	11.0300
2018-11-30	11.9100	2018-11-30	11.9100
2018-12-31	12.1600	2018-12-31	11.9100
2019-01-31	12.1600	2019-01-31	12.1600

Table 4.3: Earnings per Share for Apple Inc. corrected for look-ahead-bias. As the fiscal year for Apple Inc. ends 30 September 2018, the Earnings per Share are reported from the 30 September 2018 and three months ahead in FactSet. However, the financial report was publicly announced in November 2018.

should not be dividend adjusted since the company only are paying dividends to the shareowners. Therefore, we also have to download price data, which only are splits and spinoffs adjusted.

Global Industry Classification Standard

In order to distinguish between the companies and make a balanced trading strategy, we must classify the companies relative to their peers. MSCI[2] and Standard & Poor's (S&P) developed the Global Industry Classification Standard (GICS) for use by the global financial community in 1999. The GICS structure consists of 11 sectors, 24 industry groups, 69 industries, and 158 sub-industries, in which S&P has classified all major companies ("Global Industry Classification Standard" (2019)). The GICS codes provide the classification point-in-time for every company in our investment universe. The point-in-time feature is very convenient because sometimes compa-

[2]MSCI Inc. is a global provider of equity, fixed income, hedge fund stock market indexes, and multi-asset portfolio analysis tools ("MSCI" (2019)).

nies can change GICS classifier. Companies changes GICS classifier if the company is merging with another company or if the company has been acquired. After such situations, the company focus area may be changed, and therefore the GICS classifier changes as well. In other situation, MSCI is reorganizing the classification structure as they did in September 2018 where MSCI, e.g., chose to move the sub-industry "Internet Software & Services" from the sector "Information Technology" to be part of the sector "Communication Services".

Index Prices

It is essential to construct a risk-adjusted portfolio, i.e., avoid to invest in extreme volatile stocks. Therefore, to reduce the overall portfolio volatility, we must diversify our investments to reduce the systematic risk. As our investment universe consists of companies in the S&P 500 Index, we have downloaded the historical prices of the index in order to calculate beta between the stocks and the index. However, instead of using the cap-weighted index, we are using the equally weighted index. The difference between the cap-weighted and the equally weighted index is that in the cap-weighted index, larger companies have a more significant position, whereas, in the equally weighted index, all companies have the same weight. As our portfolios will be equally weighted, we are using the S&P 500 Equally Weighted Index to calculate betas.

4.3 Data Analytics

In this section, we will describe the data mining process we have used to prepare the data for the modelling and analysis part. The overall process is divided into distinct parts, which include the grouping process, cleaning process, outlier detection and scaling of the features.

4.3.1 Industry Groups

As we are using fundamental key-figures to decide if a stock will have a positive or negative return in the next rebalancing period, it is essential to look at the fundamental key-figure of a

company relative to its peers. Therefore, we are grouping companies according to their GICS classification. In order to choose the classification standard, we have to make some considerations. Ideally, we would like to use the finest classification system possible, as that would ensure the best diversification among the companies. However, since we are investing in around 500 companies, the classification system called "sub-industries" that consists of 158 different groups is too specific. The second most diversified group consists of 69 industries. Monthly, there are on average less than ten stocks, and sometimes only one stock in this industry, and therefore, as we want to invest in the ten per cent best and worst stocks each month this industry is also too specific. On the other hand, the roughest classification system called "sector", includes on average 47 stocks every month which would be fine. The only problem with this classification system is that we do not think that the companies are well enough diversified. For example, "Consumer Services" like hotels and restaurants will be in the same classification group as "Automobiles & Components". Therefore, we are using the GICS classification system called "industry group" consisting of 24 categories to separate the stocks. On average, there are monthly 20 companies in every industry group, but in some cases, there are also just one or two companies in each group. Therefore, when we are rebalancing our portfolios, we are making a criterion that for each day there must be a minimum of 15 stocks in each industry group. If there are less than 15 stocks in a particular industry group, we will not invest in that industry group that day.

We want to stress that the further process of the data analysis and modelling is made individually for each industry group if not stated otherwise.

4.3.2 Coverage Detection

Most of the companies do in general report the fundamental key-figures we are using, but sometimes companies are not reporting every fundamental key-figure. However, this can be expected as for some industries and sectors these key-figures may be less relevant and therefore not included in the quarterly or annual report. For example, banks are not reporting Earnings Before Interest, Taxes, Depreciation, and Amortization (EBITDA), and one could expect a lot of missing values for that industry group. Furthermore, the accounting standards we know from today were not implemented

broadly before 2001, making it difficult to maintain full coverage on all the fundamentals before 2001 ("International Financial Reporting Standards" (2019)).

To ensure that we have the best data quality, but still have as much data as possible, we are doing some coverage detection of the variables and observations. Firstly, we have examined the overall coverage for the variables on a monthly basis. As shown in appendix figure A.2, the coverage is significantly lower before February 1991. Therefore, we have determined to start our investment universe at 28 February 1991.

Next, we will take a closer look at the coverage for each variable. As shown in table 4.4 the overall coverage for the industry groups is 88.6%. As mentioned earlier, some industry groups have in general low coverage because they do not report every fundamental key-figures. To improve the coverage ratio, we are removing variables with more than 10% of missing values within each industry group. Moreover, to ensure that two variables are not a linear combination of each other, we are creating a correlation matrix. If two variables have a correlation of 1, we are removing the variable with the lowest coverage.

Furthermore, we examine the coverage of the observations. If an observation are missing more than 10% of its variables, we will exclude it from the dataset. That leads to a deletion of approximately 3.5% of the observations.

The result of this process gives us an investment universe with almost complete coverage. As seen in table 4.4, the coverage for each industry group is now above 99%. For the ML portfolios, the number of variables in the industry groups ranges from 32 to 46 out of 50 variables in total. The number of variables in the industry groups for the BM portfolio ranges from 8 to 9. However, the BM portfolio only uses valuation key-figures, and the total number of valuation key-figures is 9. Although not every industry groups has the same number of variables, it is less important since we are creating a specific model for each industry group.

Industy Group	Industy Group name	Raw cov	ML cov	BM cov
1010	Energy	0.902	0.996	0.997
1510	Materials	0.923	0.994	0.99
2010	Capital Goods	0.927	0.992	0.997
2020	Commercial & Professional Services	0.899	0.993	0.996
2030	Transportation	0.91	0.991	0.991
2510	Automobiles & Components	0.92	0.994	0.993
2520	Consumer Durables & Apparel	0.927	0.994	0.993
2530	Consumer Services	0.914	0.993	0.998
2540	Media	0.883	0.992	0.989
2550	Retailing	0.924	0.992	0.997
3010	Food & Staples Retailing	0.917	0.996	0.999
3020	Food Beverage & Tobacco	0.917	0.996	0.999
3030	Household & Personal Products	0.916	0.995	1
3510	Health Care Equipment & Services	0.898	0.994	1
3520	Pharmaceuticals Biotechnology & Life Sciences	0.92	0.995	0.997
4010	Banks	0.652	0.995	0.997
4020	Diversified Financials	0.754	0.995	0.997
4030	Insurance	0.724	0.994	0.999
4040	Real Estate	0.825	0.997	0.996
4510	Semiconductors & Semiconductor Equipment	0.925	0.993	0.99
4520	Software & Services	0.9	0.992	0.992
4530	Technology Hardware & Equipment	0.921	0.996	0.998
5010	Media & Entertainment	0.919	0.992	1
5020	Telecommunication Services	0.898	0.994	0.995
5510	Utilities	0.912	0.995	0.999
6010	Real Estate	0.82	0.998	1
Grand Total		**0.886**	**0.994**	**0.996**

Table 4.4: Coverage table. The column "Raw cov" shows the coverage from 28 February 1991 to 31 January 2019 before any detection was made. The two other columns shows the coverage for the ML and BM strategies after coverage detection.

4.3.3 Outlier Detection

Sometimes companies report fundamental key-figures which can be considered as an extreme value relative to the previously reported key-figure or its peers. An extreme value can occur in different ways. For example, it can happen that companies or the data provider accidentally report the fundamental key-figures for one period in billions instead of millions. Alternatively, some key-figures such as Dividend Payout Ratio has Net Income in the denominator, and since Net Income can go towards zero, the Dividend Payout Ratio can explode.

To avoid extreme values among the fundamental key-figures, we are truncating them down to a minimum, which can be done in different ways. In descriptive statistics, the interquartile range (IQR) measures the spread from the 75th quantile (Q_3) to the 25th quantile (Q_1). Often the IQR is used to detect outliers. In the literature, an outlier is often defined as an observation that falls below $Q_1 - 1.5 \cdot \text{IQR}$ or above $Q_3 + 1.5 \cdot \text{IQR}$. If the data is normally distributed $Q_1 - 1.5 \cdot \text{IQR}$ corresponds to -2.698 standard deviations from the mean, which corresponds to less than 0.5% of the data. Since the fundamental key-figures are far from normally distributed (most of them are

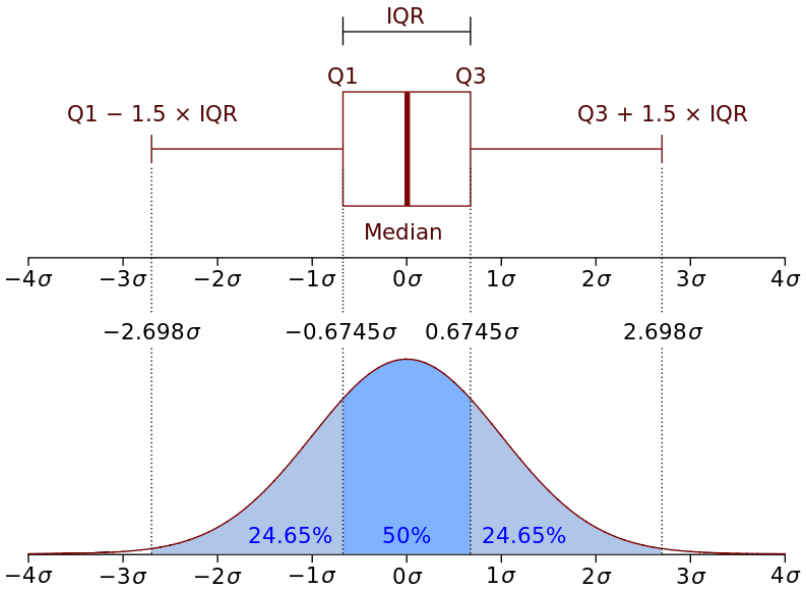


Figure 4.3: Boxplot with IQR and a probability density function of a normal distributed population ("Interquantile range" (2019)).

very right skew distributed), we do not find the IQR method suitable. Instead, we are using a conservative 1% quantile truncating in both tails. That means if an observation is lower than the 1% quantile or higher than the 99% quantile, we are replacing those values with the respectively 1% and 99% quantile value.

4.3.4 Standardized Scores

Standardized scores, or sometimes called z-scores are frequently used in statistics to compare an observation to a theoretical deviate, such as standard normal deviate. Z-scores are calculated by subtracting the population mean from each observation value and then dividing by the population standard deviation:

$$z = \frac{x - \mu}{\sigma}. \tag{4.1}$$

Other standardization method is to sort the values by lowest to highest and score each observation relative to its position in the dataset. For example, if one has 10.000 observations, the observations will be scored $1, 2, 3, \ldots, 10.000$. Afterwards, it would be appropriate to rescale the dataset so the values lies in between, e.g., [-1:1]. Using this standardization method, outliers or very high values in the dataset, would not have a significant impact anymore. On the other hand, if lots of observations have the exact same value, this method randomly ranks these observations. Therefore, if one has many observations with the same value, it would be appropriate to give these observations the same score.

Since we are interested in keeping the distribution for the variables, we find z-scoring suitable for our case. Z-scores are especially useful for a multidimensional dataset when variables on different scales need to be compared. For example, for our BM portfolio, we are creating a total valuation-score by averaging different fundamental key-figures. Since the key-figures are not following the same data range, we are standardizing all the key-figures. As an approximator for μ we are using the arithmetic mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, and for σ we are using the sample standard deviation $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2}$. By z-scoring the variables we avoid that some key-figures will have a

major impact on the total valuation-score.

4.4 Beta Stabilized Portfolio

In statistics, beta corresponds to the slope of a linear regression, in which the regression equation for a simple linear model is:

$$y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta \cdot x_i}_{\text{linear relationship}} + \underbrace{\epsilon_i}_{\text{error term}} \tag{4.2}$$

where $i = 1, 2, 3, ..., N$ represents the number of observations.

The error term:

$$\epsilon_i = y_i - (\alpha + \beta \cdot x_i) \tag{4.3}$$

accounts for the deviation from the model due to other factors, which cannot be explained by x. What we are interested in is to fit a regression line between y and x, which on average keeps the errors from every point to the line as small as possible. Mathematically this is solved by estimate α and β such that the sum of squared errors

$$\sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - (\alpha + \beta \cdot x_i))^2 \tag{4.4}$$

is minimized. This is also called ordinary least squared (OLS) method.

If we let

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad \text{and} \quad \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{4.5}$$

the solution for the parameters α and β are:

$$\hat{\beta} = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{4.6}$$

and

$$\hat{\alpha} = \bar{y} - \hat{\beta} \cdot \bar{x}. \tag{4.7}$$

As discussed in the Conceptual Framework, it is important to risk-adjust the portfolio, and a very useful risk adjustment tool is the stock's beta. Beta measures the stock's volatility towards a

benchmark and is estimated on a rolling basis. Mathematically, we split the time interval $[0, T]$ into N equidistant subintervals, which is months in our case. Further we calculate the subperiod returns of the share $r_s^1, r_s^2, r_s^3, \ldots, r_s^N$ and the subperiod returns of the benchmark $r_{bm}^1, r_{bm}^2, r_{bm}^3, \ldots, r_{bm}^N$. Depending on the chosen subperiod we can calculate the associated beta as:

$$\beta_{s,bm}^{subperiod} = \frac{Cov(r_s^{subperiod}, r_{bm}^{subperiod})}{Var(r_{bm}^{subperiod})}. \tag{4.8}$$

For example, if the return of the benchmark increases (decreases) by 1%, the return of the share price tends to increase (decrease) by β times 1%. If β is estimated to be 0.5, the return of the share tends to have half of the return as the benchmark for the subperiod, everything else being equal. In the long run, the β of a portfolio or a share versus the benchmark index tend to be positive.

Many hedge funds claim to be market-neutral. This important feature means that the hedge fund's performance does not depend on the market movements and the hedge fund will perform equally well both in bull and bear markets. Mathematically, to be market-neutral means that the portfolio has a $\beta = 0$. In practice, it can be challenging to select stocks to ensure that the hedge fund has the same β exposure in their long and short positions. However, one can also use β to hedge out the market exposure of one position. In other words, for every dollar of exposure to the hedge fund strategy, one needs to short β dollars of the market (Pedersen (2015), p. 28).

Even though it can be challenging for a hedge fund to maintain a market-neutral portfolio, there exist some practical methods the hedge fund is using when selecting stocks. A straightforward and widely used method is to rank all the stocks from bad to good performing according to the hedge fund investment strategy. Afterwards, the hedge fund is grouping the stocks in intervals based on the beta value. In each interval, the hedge fund both selects low and high ranked stocks, to ensure that they include high and low beta stocks in the portfolio. This method works well for larger investment universes. However, in our case, we are selecting stocks within every industry group. Monthly there are often 20 or fewer stocks in the industry groups, and therefore this method is not suitable in our situation. Instead we are making a regression of the total score against the S&P 500 Equally Weighted Index beta. From this regression, we are using the residuals as a new ranking score. If the original total score and beta have a linear relationship, this method will,

on average, contribute to include more stabilized beta stocks in both the long and short position. For example, if the slope is positive, we are choosing high ranking stocks with a lower beta and low-ranking stocks with a higher beta. When this process is repeated for every rebalance day, we will end up having a more stabilized beta portfolio in the long run.

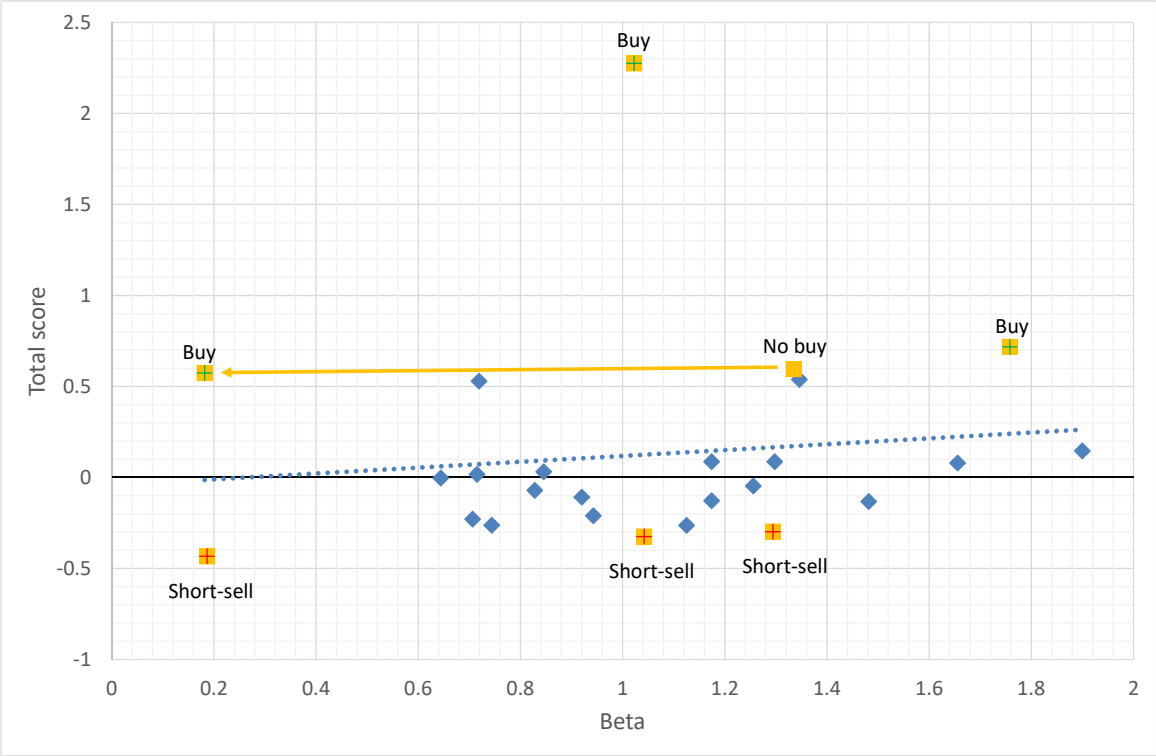


Figure 4.4: Scatterplot of the total score against beta including a regression line.

Figure 4.4 illustrates a linear regression of the total score for industry group 2520 on 30 September 1999 against the beta. On this date, the industry group consists of 25 different stocks. Since we are investing in the 10% lowest and highest scores, we will invest in three stocks in the long and short position this date. When selecting stocks without the beta adjustment, the long position have an average beta of 1.37, and the short position have an average beta of 0.84. As we are interested in being market-neutral, this is considered as a significant difference in the average beta. However, when using the residuals from the regression line as a new total score, we obtain a

new average beta of 0.99 in the long position. As a result, we have reduced the market-exposure for the long position, while the short position is unchanged.

4.5 Quantitative Benchmark Strategy

It is said that 90% of the data in the world has been created in the past few years ("Big Data: Are you ready for blast-off?" (2014)). As computers are being more powerful, it is natural to use the data to predict the future. Within finance, data from companies financial statements are being collected and stored in expensive databases. Quantitative investing is a combination of using statistics, computer science, and finance to develop a trading strategy. Based on advanced data analytics, the portfolio manager codify her trading strategy into an algorithm. Based on some criteria, the algorithm analyses an enormous amount of data and selects potential stocks to invest in. In this section, we are constructing a so-called simple quantitative benchmark strategy. The strategy is "simple" in the sense of that we are selecting stocks based on an average of some fundamental key-figures from FactSet. From that score, we are selecting the 10 per cent highest and lowest scores. Hence, we do not evaluate whether it makes sense to invest in the selected stocks, based on human judgement.

4.5.1 Construction of Simple Benchmark Strategy

As discussed in the Conceptual Framework, a successful strategy that has performed well historically is the so-called value strategy, and to construct the quantitative benchmark strategy, we are using the fundamental key-figures considered as "valuation" parameters in FactSet. The reason why we only are using the valuation key-figures is that we believe in the hypothesis, that stocks sometimes are under- or overvalued relative to their "equilibrium" price. Put differently, we believe that it is possible to identify stocks that are priced lower or higher than indicated by the fundamental value of the company. This strategy is said to be invented by Benjamin Graham as he started teaching the philosophy of patience and focus on undervalued stocks in the early 1950s. It has proved to be a successful strategy since many of Graham's students have earned a fortune

using this strategy, among them Warren Buffett (Christensen (2015)).

The nine key-figures that are classified as valuation parameters in FactSet are listed as follows:

- Price-to-Earnings
- Price-to-Book value
- Price-to-Sales
- Price-to-Cash Flow
- Price-to-Free Cash Flow
- Dividend Yield
- Enterprise Value-to-EBIT
- Enterprise Value-to-Sales
- Total Debt-to-Enterprise Value

However, in order to calculate a weighted average of the fundamental key-figures, they have to be ordered. That means the key-figures have to indicate the same thing. We are constructing the key-figures to be the higher value the higher is the likelihood for the company to be undervalued. Therefore, we are dividing the Earnings-, Book Value-, Sales-, Cash Flow-, and Free Cash Flow-per Share with the price. Additional, we are taking the inverse of Total Debt-to-Enterprise Value.

From the selection of the variables, we are separating the companies in categories based on their industry group. If an industry group contains less than 15 stocks on a given rebalancing date, we are not investing in that industry that date. To determine which stocks to invest in on a monthly basis, we are calculating a weighted average of z-scores. This score is our total valuation score and reflects the over- and undervalued stocks. In order to maintain a beta neutral portfolio on average, we create a regression of the total valuation score against the beta for the S&P 500 Equally Weighted Index. The residuals of the regression are determined as the new total valuation score, as described in section 4.4. Lastly, we identify the 10% highest and lowest valuation scores. We classify the highest 10% as 1 and the lowest 10% as -1, to indicate that we respectively are going long and short in those stocks.

4.6 Machine Learning Strategy

Most financial studies are modelling with returns, instead of the prices of the stocks. There are two main reasons for using returns. First, the return of a stock is a scale-free summary of an investment. Second, return series have more attractive statistical properties than price series (Tsay (2002)).

There are several definitions of returns, and one of the most traditional assumptions is that the simple return, are normal independently and identically distributed with fixed mean and variance. The one-period simple net return of a stock from $t-1$ to t is calculated by:

$$r_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}}. \tag{4.9}$$

Although the simple return has tractable statistical properties, it encounters several difficulties. When buying stocks, the returns tend to have positive excess kurtosis. This indicates that the probability of extreme returns is higher compared to a normal distribution.

4.6.1 Labelling Returns for Classification

Machine learning models can either be used for regression or classification problems. As we are trying to predict whether a stock has an extreme excess return in the future, we find it suitable to divide the excess returns into groups and use the machine learning algorithms as a classification problem.

As proposed by Huerta et al. (2013), modelling of returns are often divided into three categories:

- Real returns
- Excess return relative to the sector
- Returns divided by volatility estimate

The first option that uses real returns, make the model focus on stocks with higher volatility as those also typically have the highest return from one period to another. This strategy tends to

form portfolios with larger drawdowns and volatility. Option two regresses returns on the sector index makes this focus on modelling excess return within the sector. Option three creates a list of ordered stocks with volatility-adjusted returns.

Although Huerta et al. are using option three, option two is our preferred choice. Specifically, we are estimating the β between the stock returns and the average returns for the industry group on a three-year rolling basis. Hereafter, we are calculating the monthly excess return as:

$$r_s^{excess} = r_s - \beta_{s,ig} \cdot r_{ig}. \tag{4.10}$$

To label the excess returns, we experimented with different options. The first option was to label the excess return in three classes. These classes were 1 for the highest excess return, -1 for the lowest, and 0 otherwise. To avoid imbalanced data, we started to split the classes into three equal parts. One of the difficulties in using this labelling technique was that the model was not able to distinguish the classes, and predicted almost 90% of the excess returns as 0. However, Huerta et al. suggests to remove the neutral ones. The reason behind this is that the model now is a two-class classification problem, instead of a multi-class, which is easier to handle. Additional, Huerta et al. explains in their study that the mid-ranking volatility-adjusted returns tend to follow the trend of the market. Furthermore, the mid-ranking volatility-adjusted returns also tend to follow the unsystematic structure in the data with no strong correlations to the explanatory variables. By removing the neutral stocks from the training dataset, we experienced that the overall performance increased. Therefore we chose to disregard the neutral class from the training model. As we now have a two-class classification problem, we can model more extreme excess returns without an imbalanced dataset. Although, the training model has to contain a reasonable number of observations to avoid underfitting. We find it suitable to do an 20/80 quantile classification split of the excess returns. In other words, we are labelling the 20% lowest excess returns as -1 and the 20% highest excess returns as 1, and hereby using 40% of the original training datasets to train our models.

4.6.2 Data Separation

First of all, we are splitting the datasets into three parts: training, validation, and test dataset. We are splitting the dataset into three parts, as we want to optimise the models for the validation set. Only when we have a proper model that is tested on the validation set, we will include the test set, to see how the model fits on the test data. The three datasets represent the following periods:

- Training dataset from 28 February 1991 to 31 January 2011
- Validation dataset from 28 February 2011 to 31 January 2015
- Test dataset from 28 February 2015 to 31 January 2019

This corresponds to about 70%/15%/15% of the whole time period. However, this separation does not exactly match the distribution of the observations in the datasets, as the number of stocks in the industry groups varies over time. Not all industry groups contain observations in each period, and some industry groups do not fulfil the constraint to a minimum of 15 stocks each rebalancing date. Therefore we are excluding industry groups in the machine learning part if they do not meet these criteria in just one of the datasets. As a result of this, the machine learning datasets includes 15 of the 26 industry groups. Table 4.5 shows the coverage of the different datasets.

4.6.3 Variable Reduction

Discussed in section 4.3.2, the number of variables for each industry group differs. However, machine learning algorithms like naïve Bayes classifier and support vector machines shows poor accuracy with many irrelevant features ("Feature Selection Techniques in Machine Learning with Python" (2018)). As our dataset contains many similar variables, we are clustering those variables that are similar to each other.

From a correlation matrix, we are using K-means with the Silhouette method to find the optimal number of clusters. K-means is a nonhierarchical clustering method designed to group items in

Period	1010	1510	2010	2020	2030	2510	2520	2530	2540
Train	0.62	0.78	0.72				0.82		
Vali	0.20	0.12	0.13				0.05		
Test	0.18	0.10	0.14				0.13		

Period	2550	3010	3020	3030	3510	3520	4010	4020	4030
Train	0.67		0.68		0.62	0.53	0.91	0.53	0.66
Vali	0.16		0.17		0.17	0.22	0.04	0.23	0.17
Test	0.16		0.15		0.21	0.25	0.05	0.24	0.17

Period	4040	4510	4520	4530	5010	5020	5510	6010
Train		0.52	0.88	0.63			0.74	
Vali		0.22	0.11	0.23			0.13	
Test		0.25	0.01	0.15			0.13	

Table 4.5: The coverage of the datasets for each industry group. Industry groups with low coverage are excluded from the datasets.

a collection of K clusters. The number of clusters is either specified in advance or determined as part of a procedure. In this case, the procedure is the Silhouette method. The way K-means works is that it starts to partition the items into K initial clusters. For each cluster, the items are reassigned to the cluster whose centroid (mean) is nearest. The distance is computed using the Euclidean distance of the correlation matrix of all the variables. As the items change clusters, new centroids are recalculated. This procedure is repeated until no more reassignments take place (Johnson and Wichern (2013)). The Silhouette method measures the quality of a cluster. That is, this method determines how similar each item is within its cluster. A good cluster combination has a high average silhouette width. The average width is calculated for different values of K clusters, and the cluster with the highest width is considered as the best cluster. The formula for the silhouette width is defined as:

$$S_i = \frac{(b_i - a_i)}{max(a_i, b_i)}, \tag{4.11}$$

where $b_i = min_K d(i, K)$ is the smallest distance to the cluster where item i does not belong, and

a_i is the average dissimilarity between i and all other items of the cluster to which i belongs.

As we have eight different categories of fundamental key-figures, we want at least eight clusters, and the silhouette width defines the optimal number of clusters from eight and beyond. From the optimal number of clusters, new features are created based on an average of the identified correlated variables.

Random forest works well with lots of features since it does not consider all variables, but only a sample of them. Therefore we do not find it necessary to lower the dimension of the datasets for RF.

4.6.4 Accuracy and Hyperpameter Tuning

As mentioned in section 3.3, the RF and SVM algorithms have several parameters to tune to obtain the model with the highest accuracy. To find the most stable and suitable combination of parameters, we are implementing a grid search. Based on several combinations of parameters, models are produced from the training dataset with a prediction on the validation dataset.

For the RF model, we are tuning over the parameters mtry and nodesize. Mtry is the number of variables randomly sampled as candidates at each split, and nodesize is the minimum size of terminal nodes. We are using these parameters as we find it essential to know how many variables that have to be included to obtain the best model. If many variables give the best accuracy, it indicates that the data constitute of a few relevant variables, as the probability of getting a variable with low impurity is high. Moreover, we are using the nodesize in the tuning process, as the nodesize indicates how far a tree can grow. A good accuracy for a deep tree indicates that the data from the training to the validation period is similar, and the model can be more detailed without increasing the bias.

For the SVM model, we are using the radial kernel, which introduces a parameter other than cost. The parameter for the radial kernel is gamma and indicates the smoothness of the curvature of the decision boundary. A high value of gamma takes fewer support vectors into account, while a small value of gamma considers a larger amount of support vectors near the decision boundary.

If the value of gamma is minimal, the decision boundary acts similar to a linear model. The cost parameter indicates the cost of misclassifying an observation. A small cost focus on a wide hyperplane and vice versa.

The NB model does not include any parameter to tune as it only considers the conditional probability to classify observations.

The Tuning Process

The hyperparameters for RF and SVM are listed in the following table:

RF										
mtry	3	5	7	9	11	13	15	17	19	21
nodesize	7	14	21	28	35	42	49	56	63	70
SVM										
gamma	0.015625	0.03125	0.0625	0.125	0.25	0.5	1	2	4	8
cost	1	2.5	5	10	25	50	100	250	500	1000

Table 4.6: The tuning parameters for RF and SVM.

For each combination of the parameters, a model based on the training dataset is created with a prediction on the validation dataset. For every prediction, several accuracy measures are stored in a data frame. The accuracy measures are listed as follows:

- The overall hit-ratio
- The sensitivity for -1
- The sensitivity for 1
- The sensitivity for 0
- The positive predicted value for -1
- The positive predicted value for 1
- The negative predicted value for -1
- The negative predicted value for 1

To select the best combination of the parameters, several heatmaps are constructed. The heatmaps are a 10×10 matrix that spans all the different combination of the hyperparameters. We are using a heatmap to select the best combination of hyperparameters, as we want a stable model,

and not necessarily the overall best combination, as that combination easily could be achieved due to luck. Moreover, we are interested in finding the model with the best combination of the positive predicted value, negative predicted value, sensitivity and hit-ratio. As described in section 3.2.3, it is not sufficient to evaluate a model only based on the overall hit-ratio. A model with a high hit-ratio can also be a model with a low precision rate for negative excess returns, and the contribution to the high hit-ratio comes from a high precision rate for the positive excess return. Therefore it is essential when we want to find a stable model, that we take all the different accuracy measures into account.

mtry/ nodesize	7	14	21	28	35	42	49	56	63	70	Average
3	0,225	0,244	0,256	0,300	0,244	0,272	0,275	0,275	0,259	0,289	0,264
5	0,259	0,284	0,284	0,256	0,275	0,250	0,259	0,284	0,259	0,272	0,268
7	0,247	0,247	0,217	0,247	0,250	0,256	0,284	0,263	0,256	0,259	0,253
9	0,238	0,238	0,244	0,250	0,253	0,250	0,259	0,272	0,250	0,263	0,252
11	0,259	0,265	0,235	0,235	0,272	0,205	0,263	0,250	0,244	0,247	0,247
13	0,213	0,272	0,263	0,238	0,247	0,250	0,259	0,250	0,256	0,275	0,252
15	0,247	0,275	0,247	0,232	0,259	0,272	0,247	0,238	0,272	0,247	0,253
17	0,259	0,188	0,232	0,244	0,309	0,250	0,253	0,250	0,238	0,225	0,245
19	0,225	0,213	0,263	0,250	0,198	0,238	0,247	0,200	0,247	0,250	0,233
21	0,275	0,235	0,300	0,222	0,238	0,220	0,238	0,250	0,210	0,275	0,246
Average	**0,245**	**0,246**	**0,254**	**0,247**	**0,254**	**0,246**	**0,258**	**0,253**	**0,249**	**0,260**	**0,251**

0,251	0,259	0,259	0,261	0,263	0,269	0,268	0,268
0,251	0,252	0,253	0,254	0,260	0,264	0,265	0,264
0,243	0,242	0,245	0,246	0,255	0,256	0,260	0,256
0,247	0,249	0,248	0,244	0,251	0,251	0,256	0,256
0,253	0,251	0,247	0,245	0,253	0,248	0,253	0,253
0,244	0,243	0,252	0,256	0,261	0,252	0,251	0,250
0,239	0,238	0,248	0,250	0,252	0,244	0,243	0,241
0,243	0,238	0,250	0,241	0,243	0,238	0,237	0,238

Figure 4.5: Heatmap for industry group 4010 that shows the positive predicted value (PPV) for negative excess returns. The best PPV is obtained where mtry = 17 and nodesize = 35. However, this combination of parameters is not the most stable one as the PPV for the parameters around varies a lot. A more stable combination of parameters is found for mtry = 5 and nodesize = 49. The dotted lines indicates an average of the parameters in the bottom grid.

4.6.5 Final Machine Learning Strategies

The final machine learning models consists of the following combination of parameters:

	SVM			RF			NB
industry group	**clusters**	**gamma**	**cost**	**variables**	**mtry**	**nodesize**	**clusters**
1010	9	8	5	43	19	56	9
1510	9	0.015625	1	46	13	49	9
2010	13	8	25	46	21	56	13
2520	16	0.015625	10	45	19	49	16
2550	13	8	1	45	15	49	13
3020	11	0.125	1000	44	17	63	11
3510	8	0.015625	1	35	17	56	8
3520	9	0.5	50	45	15	63	9
4010	15	0.015625	1000	33	5	63	15
4020	8	0.5	10	32	21	21	8
4030	10	0.015625	100	33	17	63	10
4510	9	8	1	43	13	63	9
4520	12	2	25	43	17	49	12
4530	13	0.015625	500	43	15	63	13
5510	8	0.0625	50	46	17	49	8

Table 4.7: The final combination of machine learning parameters.

To select the stocks to invest in, we are training the machine learning models on the training and validation datasets. Each model is trained specifically for each industry group. The models are trained to predict extreme excess returns. When predicting on the test dataset, each stock will get a likelihood to have either an extreme positive or extreme negative excess return. This likelihood corresponds to a total score, and as we did for the BM strategy, we are making a regression of the total likelihood score against the beta for the S&P 500 Equally Weighted Index. From the regression, we are using the residuals as the new total score, in order to make the score more stabilised compared to the beta. Lastly, we identify the 10% highest and lowest scores. We classify the highest 10% as 1 and the lowest 10% as -1, to indicate that we respectively are going long and

short in those stocks.

4.7 Portfolio Turnover, Transaction Costs, and Short Fees

In this section, we will outline how we are calculating the profit and loss for our portfolios. We are using the same procedure for all our investment strategies, whether it is a backtest of the benchmark or a machine learning strategy. To make the trading algorithm more realistic, we are adjusting for transaction costs and short-selling fees. Adjusting for transaction costs makes the performance of the portfolio more sensitive to high frequently rebalancing. To simplify, we are not reinvesting profits, but keeps profits on a bank account with no interest rates. Furthermore, we assume that we have money to cover losses.

4.7.1 Portfolio Turnover

The portfolios we are constructing includes a long and short position, and therefore, we are calculating the turnover for each position separately. At each rebalancing date, the same amount of capital is invested in both positions and divided equally between the identified stocks. The turnover for each position at time t is calculated as the difference in the invested capital from $t-1$ to t. The turnover for each rebalancing period is withdrawn and saved on a bank account with no interest rates. If the turnover is negative, we will add more capital to the position in order to keep the invested capital fixed. The total turnover for the whole period is the total value on the bank account at time T:

$$\text{TT} = \sum_{t=1}^{T} Turnover_t^{long} + \sum_{t=1}^{T} Turnover_t^{short}, \tag{4.12}$$

where T is the total number of rebalancing dates.

4.7.2 Transaction Costs and Short Fees

The costs for our trading strategies includes both transaction costs and short-selling fees. The short-selling fee is fixed at 30 basis points per year, which corresponds to 30/12 basis points per

month for each dollar shorted. The transaction costs are fixed at 5 basis points for each bought or sold dollar. The total cost is defined as:

$$\text{TC} = \sum_{t=1}^{T} transactioncost_t^{long} + \sum_{t=1}^{T} transactioncost_t^{short} + \sum_{t=1}^{T} shortfee_t, \tag{4.13}$$

and the realised turnover is obtained by subtracting (4.13) from (4.12):

$$\text{RT} = TT - TC. \tag{4.14}$$

4.8 Portfolio Performance Measures

To evaluate the performance of our portfolios, we are calculating the annualised return, annualised standard deviation, annualised Sharpe ratio, and maximum drawdown.

To calculate the annualised return, we are estimating the expected returns as the arithmetic average:

$$\text{E[R]} = \frac{(R_1 + R_2 + ... + R_T)}{T}, \tag{4.15}$$

where T is the number of time periods. To annualise the expected return, we multiply $E[R]$ by n which is the number of periods per year:

$$\text{E[R]}^{annual} = E[R] \cdot n. \tag{4.16}$$

The standard deviation is calculated as the square root of the variance, which is estimated as the squared deviations around the arithmetic average:

$$\sigma = \sqrt{\frac{(R_1 - \bar{R})^2 + (R_2 - \bar{R})^2 + ... + (R_T - \bar{R})^2}{T - 1}} \tag{4.17}$$

To annualise the standard deviation, we multiply σ with the square root of n:

$$\sigma^{annual} = \sigma \cdot \sqrt{n}. \tag{4.18}$$

The annualised Sharpe ratio is calculated as the annual return divided by the annually standard deviation. In general Sharpe ratio is a risk-reward ratio and is found by the expected excess return,

$E(R - R^f)$, divided with the standard deviation of $(R - R^f)$, where R^f is the risk-free rate. The formula for the Sharpe ratio is as follows:

$$\text{SR} = \frac{E[R - R^f]}{\sigma[R - R^f]}. \tag{4.19}$$

As the capital in our strategies is financed by short-selling stocks, we are already in excess. Therefore, we can rewrite the annualised Sharpe ratio as:

$$\text{SR}^{annual} = \frac{E[R]^{annual}}{\sigma[R]^{annual}}. \tag{4.20}$$

Another essential performance measure is the portfolio's drawdown (DD). The DD is the decline of the portfolio value during a specific period. Often the drawdown is quoted as the percentage change of the portfolio value. Usually, one is interested in finding the maximum drawdown (MDD). To calculate the MDD we will introduce the high water mark (HWM). The HWM represents the highest achieved price in the past:

$$\text{HWM}_t = max_{s \leq t} P_s. \tag{4.21}$$

Using the HWM we can calculate the DD as follows:

$$\text{DD}_t = \frac{HWM_t - P_t}{HWM_t}. \tag{4.22}$$

The MDD is found by taking the maximum DD over the time period:

$$\text{MDD}_T = \max_{t \leq T} DD_t. \tag{4.23}$$

If the MDD is significant large, it is costly for the hedge fund, and the portfolio is considered as a risky portfolio.

5 Results

In this chapter, we will present the results we have obtained during our analysis. The first part will focus on a representation of the results for the benchmark strategy. The representation consists of the performance for the whole time period, and how stabilised the strategy is. Furthermore, we will present the performance of the test period. The second part will focus on a representation of the machine learning strategies. This consist of the performance of the test period for the three ML strategies. The third part will compare the BM to the ML strategies. Lastly, we will examine whether there is a relationship between the overall hit ratio and return for all the investment strategies.

5.1 Results for the Benchmark Strategy

The benchmark strategy is based on nine different valuation fundamental key-figures. Using a simple quantitative model, we have identified the 10 % most under- and overvalued stocks each month. A portfolio consisting of the identified stocks from 28 February 1991 to 31 January 2019 is constructed, and the performance presented in the following table:

R	**Beta lng**	**Beta sht**	**AnnR**	**AnnV**	**AnnSR**	**MDD**	**Hit**	**Hld lng**	**Hld sht**
1.817	1.094	1.053	0.065	0.099	0.658	-0.404	0.507	6.563	5.017

Table 5.1: Performance and evaluation parameters for the benchmark strategy from 28 February 1991 to 31 January 2019.

The benchmark strategy has generated a positive return on 1.817, including transaction costs and short-selling fees. The portfolio is market-neutral on average as the beta for the long and short positions is close to each other at an acceptable level. The maximum drawdown is -0.404, which means that the bank account has at some point in time lost 0.404 of the realised turnover. The overall hit ratio indicates that the strategy has managed to correctly identify 50.7% of the stocks excess return relative to the industry group. The 95% confidence interval for the hit ratio is

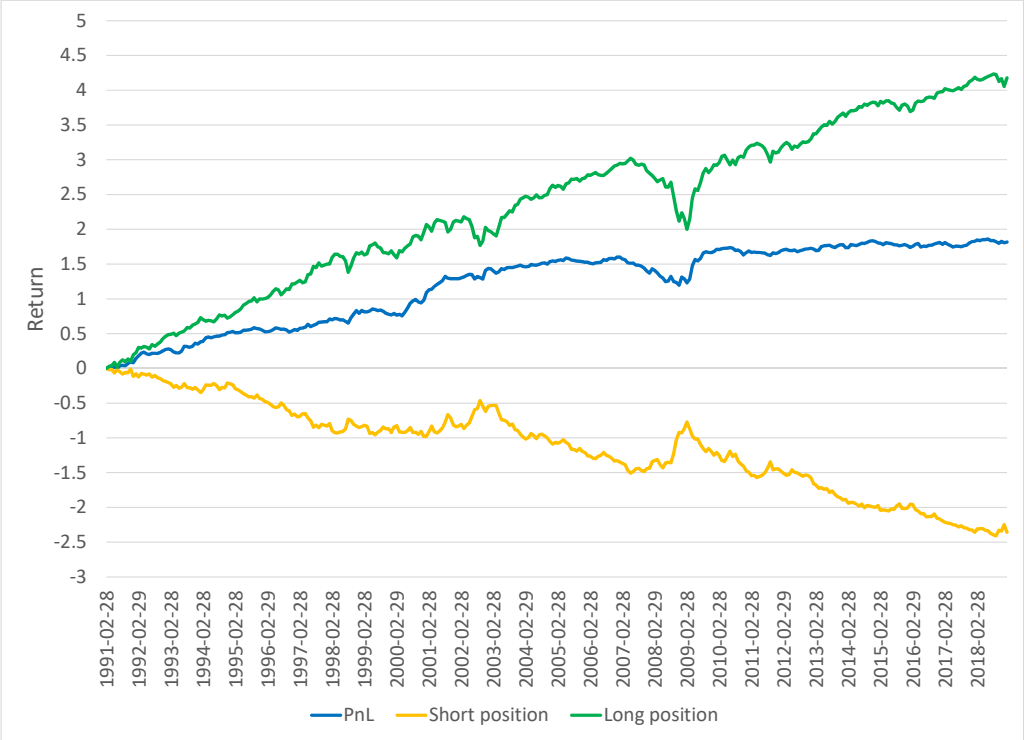


Figure 5.1: Profits and losses for the benchmark strategy.

[50.1%; 51.3%]. The "no information rate" (NIR), which is the largest class percentage in the data, is 51.1%, and the p-value for whether the hit ratio is greater than the NIR is 0.922. This indicates that there is no statistical evidence for that the hit ratio is greater than the NIR. The average holding period for the long position is 1.5 months longer than for the average holding period for the short position. The average holding period indicates that the fundamental value and the price of the stock have come into equilibrium after 6.5 months for an undervalued stock, and 5 months for an overvalued stock. A chart of the return is shown in figure 5.1. The chart shows that in general, the long position generates positive turnover while the short position generates negative turnover. However, in times with crises, there is an opposite trend. From 28 February 1991 to 30 June 2005 (approximately half of the time period), the strategy has generated a return of 1.579. In contrast, from 31 July 2005 to 31 January 2019, the strategy has generated a return of 0.238.

It indicates that the benchmark strategy has performed better in the past compared to the last decade.

5.1.1 Benchmark Stability Test

To test whether the benchmark strategy is stable we have performed a "leave one out" test for the variables. The test will show if any of the variables have a significant impact on the overall result. The annual return, standard deviation and Sharpe ratio for each test is represented in table 5.2. By comparing the mean of the tests to the BM strategy, we find that the mean Sharpe ratio for the tests is lower than the Sharpe ratio for the BM strategy. This indicates that the BM strategy is better when combining all the variables. The mean volatility of the tests is almost equal to the volatility of the BM strategy, which means that on average, the strategies have the same volatility. Although the strategies show different results, it seems like a reasonable assumption that the BM strategy is stable.

Missing variable	**AnnR**	**AnnVol**	**AnnSR**
Dept/Enterprise Value	0.064	0.104	0.620
Dividend Yield	0.065	0.099	0.651
Enterprise Value/EBIT	0.061	0.101	0.601
Enterprise Value/Sales	0.063	0.107	0.586
Earnings/Price	0.056	0.101	0.552
Book Value/Price	0.060	0.088	0.680
Sales/Price	0.066	0.091	0.730
Cash Flow/Price	0.044	0.092	0.480
Free Cash Flow/Price	0.055	0.102	0.537
Mean	0.059	0.098	0.604
Standard deviation	0.007	0.006	0.077
Benchmark strategy	0.065	0.099	0.659

Table 5.2: Performance for the stability test.

5.1.2 Benchmark Results for the Test Period

To compare the BM- and ML strategies, we will evaluate the performance from 28 February 2015 to 31 January 2019 that corresponds to our test period for the ML strategies. As shown in table 5.3, the BM strategy shows poor performance with an annual return on 0.4%. In the next section, we will present the results for the ML strategies, and compare these results to the BM strategy.

5.2 Results for the Machine Learning Strategies

The three selected ML models have the possibility to fit a model based on all the 50 fundamental key-figures from FactSet. However, as the NB and SVM are sensitive to highly correlated variables, we have lowered the dimension by clustering the variables for those models. Additional, the RF and SVM are tuned using a grid search, and the most stable hyperparameters are selected. The number of variables and hyperparameters for the ML models are listed in table 4.7 in the previous chapter. The models are trained to predict extreme excess return within each industry group. From the prediction, we identify the 10% highest and lowest likelihoods for an extreme excess return. The training period for the final models goes from on 28 February 1991 to 31 January 2015, and the test period goes from on 28 February 2015 to 31 January 2019. The portfolio performance for the ML- and BM strategies are presented in table 5.3, and a chart of the return is shown in appendix figure A.3.

	R	**Beta lng**	**Beta sht**	**AnnR**	**AnnV**	**AnnSR**	**MDD**	**Hit**	**Hld lng**	**Hld sht**
SVM	0.152	1.075	1.033	0.038	0.052	0.735	-0.057	0.516	4.784	4.706
NB	0.159	1.010	1.029	0.040	0.061	0.650	-0.108	0.522	7.610	8.278
RF	-0.014	0.984	1.017	-0.003	0.056	-0.062	-0.078	0.499	5.346	4.703
BM	0.017	1.086	1.078	0.004	0.062	0.068	-0.068	0.496	6.071	5.041

Table 5.3: Performance and evaluation parameters for all the strategies from 28 February 2015 to 31 January 2019.

The NB and SVM strategies show great performance in the test period. The annualised return

is around 4%, but the volatility for the SVM is lower than the volatility for the NB, which causes the highest Sharpe ratio for the SVM. What is remarkable when comparing the strategies is the difference between the MDD and the average holding period. The NB has almost double as high MDD than the SVM, which indicates that the NB is a more risky model. Additional, the average holding period for NB is over three months longer compared to the SVM. The results for the RF has shown poor performance in the test period. However, the interesting thing is that the RF and the BM strategy have shown similar results. Even though the return for the RF is negative, the overall key statistics are not far from the key statistics for the BM strategy. It indicates that the RF might not be a bad model compared to the BM strategy. When evaluating the hit ratio for the strategies, it indicates that there is a link between a high hit ratio and a high return. However, only the NB has a statistically significant p-value at a 0.05 level, for the hit ratio is greater than the NIR. The confidence interval and p-values for the strategies can be found in appendix A.5.

5.2.1 Prediction Power

In this section, we will investigate the relationship between the likelihood for an extreme excess return and the return. As the portfolio consists of a long and short position, the machine learning models predicts the likelihood of extremely positive and negative excess returns. To investigate the relationship, we have divided the likelihoods into ten intervals consisting of 10% of the stocks. The likelihood intervals are shown in table 5.4. The intervals from 0.0 to 0.5 represent stocks with a higher likelihood for extreme negative excess returns, and the intervals from 0.5 to 1.0 represents stocks with a higher likelihood for extreme positive excess returns.

Short	[0.0 ; 0.1[	[0.1 ; 0.2[	[0.2 ; 0.3[	[0.3 ; 0.4[	[0.4 ; 0.5[
Long	[1.0 ; 0.9[	[0.9 ; 0.8[	[0.8 ; 0.7[	[0.7 ; 0.6[	[0.6 ; 0.5[

Table 5.4: Likelihood intervals.

Figure 5.2 illustrates the return for each likelihood interval for the NB. The charts for the SVM and RF are shown in appendix A.6 and A.7. The interesting thing is whether there is a separation between the realised turnover for the different likelihoods. It seems like there is the

correct tendency for the NB, but the order of the return for SVM and RF is not as good as for the NB. The SVM and RF have obtained the best results in the intervals $]0.8; 0.9]$ and $]0.7; 0.8]$ for the long position. Furthermore, the interval with the highest likelihood, $]0.9; 1.0]$, has shown the worst performance for RF. This indicates that RF has found it challenging to find the highest extreme excess returns.

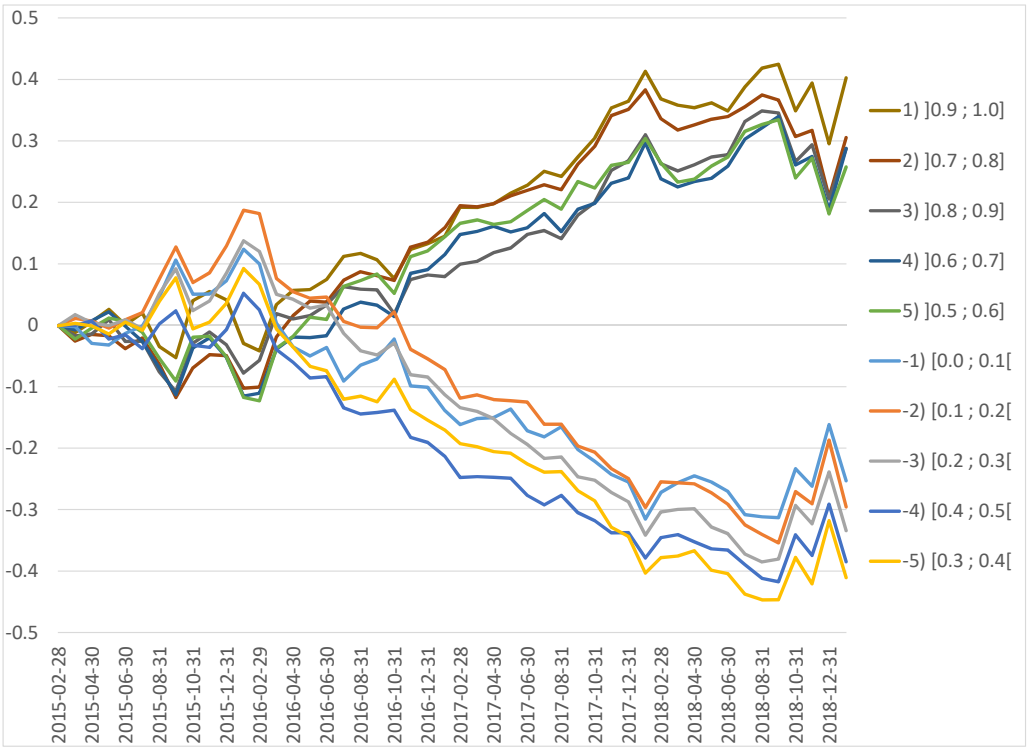


Figure 5.2: Profits and losses for different likelihood prediction intervals for the NB.

5.2.2 Industry Group Contribution

The contribution of each industry group is presented in figure 5.3. From the figure, we see that all the ML strategies achieve a positive turnover in 6 out of the 15 industry groups, but RF has struggled to select the best stocks in several industry groups. The Health Care sector consisting of the industry groups Health Care Equipment & Services and Pharmaceuticals, Biotechnology & Life Sciences, have in general shown poor performance. However, this sector is a notoriously

fickle industry. Often the fundamental key-figures does not represent the value of the stocks, but instead, the prices are influenced by expectations to the growth of the companies.

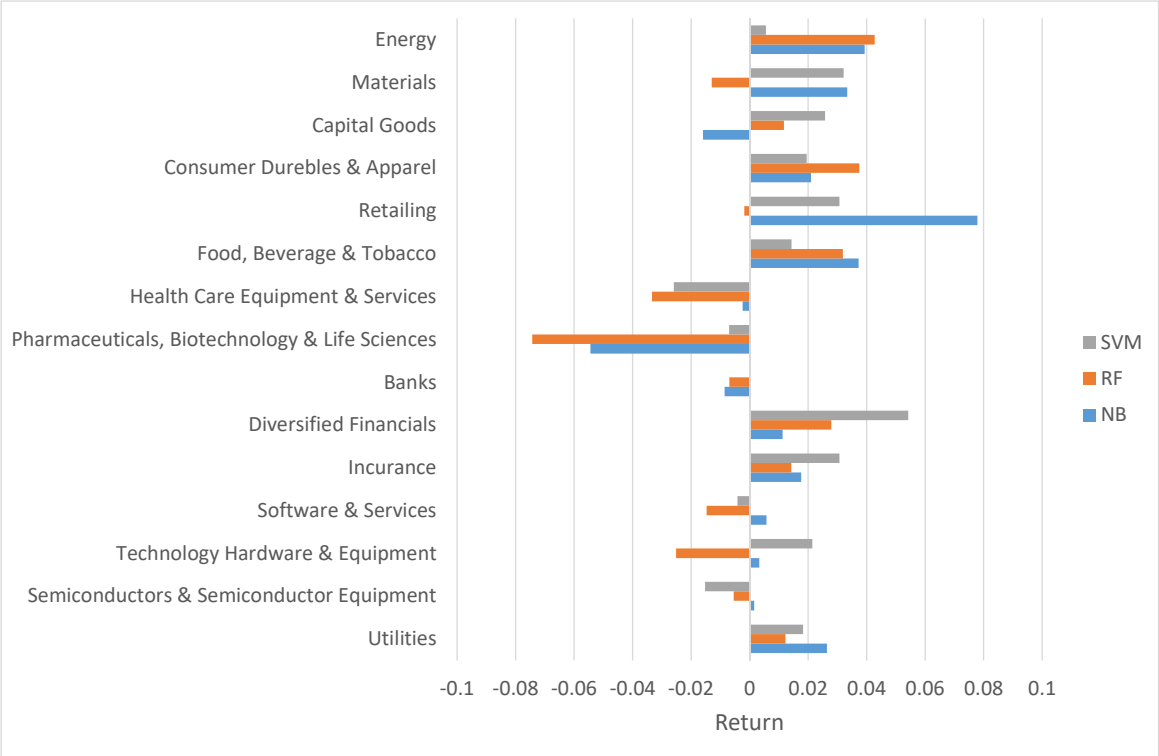


Figure 5.3: Contribution chart for the industry groups in the test period.

5.3 Machine Learning Stability Test

To test how the ML models have performed in the past, we have evaluated several models at different time periods. The models are based on data from the training and validation datasets. For example, if we predict from 28 February 2001 to 31 January 2005, the model is trained on the observations from 28 February 1991 to 31 January 2001 and from 31 March 2005 to 31 January 2015. We are offsetting the training period two months after the test period to eliminate quarterly tendencies directly related to the fundamental key-figures. The ML models are created to predict the four years tendency on a one-year rolling basis. In figure 5.4, the annual Sharpe ratios for all

the strategies are shown.

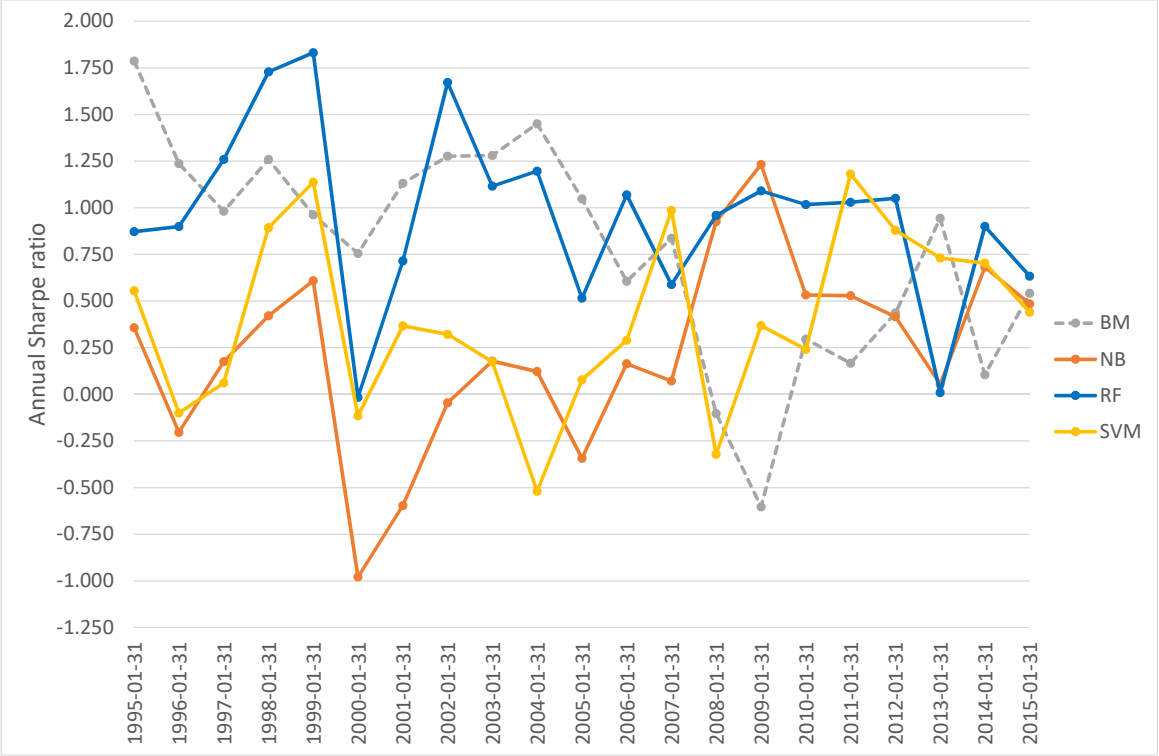


Figure 5.4: Chart with the annual SR for the strategies.

The figure shows that the RF, in general, achieves the highest Sharpe ratios relative to the other ML strategies. Furthermore, the RF seems to be the most stable model with an average annual SR at 0.959. The average annual SR for SVM is 0.398, and for the NB it is 0.228. However, if we only include the past ten periods, the NB and the SVM have obtained a higher average annual SR, but RF still has the highest annual SR. Furthermore, table 5.5 shows that the average annual SR for the BM is lower in the past ten periods compared to the whole period.

Ending dates for the interval	**BM**	**NB**	**RF**	**SVM**
28 February 1995 to 31 January 2015	0.781	0.228	0.959	0.398
28 February 2006 to 31 January 2015	0.323	0.509	0.835	0.550

Table 5.5: Average annual SR for the stability test.

When looking at how many times the models have obtained a higher annual SR than the BM

strategy, we see the same tendency. RF has achieved a higher annual SR than the BM in 12 out of 21 periods, which corresponds to 57.1%. The NB and SVM have only obtained a higher annual SR than the BM in around 25% of the times. However, in the past ten periods, all of the ML models have obtained higher annual SR than the BM at least 50% of the times.

Ending dates for the interval	**NB**	**RF**	**SVM**
28 February 1995 to 31 January 2015	5	12	6
28 February 2006 to 31 January 2015	5	8	5

Table 5.6: Number of times the models has obtained a higher average annual SR than the BM.

5.4 Return versus Hit Ratio

To investigate the relationship between the return and the hit ratio, we are using the information from the stability test performed in the previous section. Figure A.8 in the appendix, shows a scatterplot of the return against the hit ratio. The trend indicates that a better hit ratio leads to a higher return, and the correlation between the features is 0.693. A linear regression between the return and the hit ratio has a beta on 11.065 with a p-value near zero and an r-squared on 0.481. This indicates that there is a significant positive relationship between the return and the hit ratio.

6 Conclusion

In this final chapter, we will answer and discuss the thesis statement stated in section 1.2, based on our analysis and results. Furthermore, we will introduce some interesting future changes of this thesis, which could lead to an improvement in the results.

6.1 The Findings of This Thesis

In this thesis, we have constructed a simple investment strategy using nine fundamental key-figures that describes the valuation of a company. Every month the strategy identifies the most over- and undervalued stocks within each industry group. From the identified stocks, we construct a long/short portfolio which we are using as our benchmark (BM). The annual Sharpe ratio (SR) in the period from 28 February 1991 to 31 January 2019 is 0.658. Most of the turnover is realised in the first half of the period, but the turnover is diminishing in the latter half of the period. A stability test constructed using the "leave one out" technique shows that portfolios consisting of eight out of the nine fundamental key-figures also obtained a positive return. The minimum annual SR for the portfolios is 0.480. In the past four years, the strategy has achieved an annual SR at 0.068. Even though the return is positive, the annual SR is ten times lower compared to the whole time period. It indicates that it is possible to construct a simple benchmark strategy that generates a positive return. However, the benchmark strategy tends to perform better in the past compared to recent times. This is also what other investors have experienced. According to the US fund management company Wisdom Tree, the valuation key-figure Price-to-Book value has in the last decade been the worst in history to measure the value of a stock. It is often suggested that value investing have been matured in popularity, and lost the effectivity as the strategy has become widely known. Another factor that has performed well since the financial crisis is the growth factor. Growth companies are companies showing great potential in future growth. It could be stocks from companies with strong historically and forward earnings growth, strong return on equity and strong stock performance. As an example, the FAANG stocks consisting of

Facebook Inc., Amazon, Apple Inc., Netflix and Google, is considered as growth stocks and almost all the companies have achieved at least a ten times increase of the stock price since the end of the financial crisis. However, Cliff Asness from AQR points out that even though value investing have underperformed the last decade, it is too early to say that value investing is dead and a turnaround in the future is not impossible "Value investing: is the age-old strategy dead?" (2019).

In this thesis, we are using three different supervised machine learning (ML) algorithms to identify stocks with either extremely high or low excess returns. In the test period from 28 February 2015 to 31 January 2019, the ML models are predicting the likelihood for an extremely positive or negative excess return based on fundamental key-figures. The results show that the naïve Bayes (NB) classifier and the support vector machines (SVM) achieve a higher SR than the BM portfolio. The random forest (RF) model achieved a small positive return, but after transaction costs and short-selling fees, the turnover became negative and ended having a lower SR than the BM portfolio. To test the stability of the ML models, we trained and tested the models in several time periods. The stability test shows that the ML models, in general, have achieved a positive annual SR, and the hit ratios range from 48.7% to 54%. This indicates that it is possible for the supervised machine learning algorithms to predict stocks with an extreme excess return. By comparing the ML models to the BM strategy, the RF model has achieved the highest annual SR. In contrast, the NB and SVM achieve, on average, an annual SR that is significantly lower than the annual SR for the BM. However, when investigating the performance in the past ten years of the stability test, all the ML models achieve an average annual SR that is higher than the BM strategy. One interesting thing about the RF model is the variable importance feature, that calculates the impurity of the variables at each split. It turns out that RF primarily has used five fundamental valuation key-figures to build the model in the test period. As shown in appendix figure A.9 the key-figures are Book-to-Price, Sales-to-Price, Cash Flow-to-Price, Earnings-to-Price and Free Cash Flow-to-Price. This may be the reason why the RF and the BM strategy shows similar results in the test period, but also for the stability test.

A linear regression of the returns against the hit ratios shows a statistically significant relationship, and the correlation between the features is 0.693. It proofs that there is a positive relationship

between the return and the hit ratio for the ML models.

In general, the ML models have shown positive performance. However, none of the ML models has proven to beat the BM strategy consistently. In recent times the NB and SVM achieve the highest annual SR, but they have not performed well in the past. Conversely, RF has achieved good performance in the past but has struggled in recent times. Furthermore, the RF model follows the same trend as the BM strategy. The conclusion of this thesis is that it has not been consistently possible to beat a self-constructed quantitative benchmark strategy using artificial intelligence.

6.2 Future Work

As this thesis only uses fundamental key-figures, it could be interesting to include technical key-figures such as trends in stock prices and moving average. Technical key-figures do not attempt to evaluate a stocks intrinsic value, but instead, use price movements to recognise trends that suggest what the stock price will be in the future.

Another approach to building a more robust ML model is by combining multiple of ML algorithms and identify the stocks based on an average of the likelihood predictions of the models. By combining various models, it is possible to identify stocks with an overall high likelihood for an extreme excess return. Moreover, instead of predicting an extremely high or low excess return in one model, one can create two models, each of which predicts either the extreme high and extreme low excess return. This can potentially improve the performance of the models as they only have to focus on predicting one specific class. However, one difficulty using this method is that the models potentially can predict the same stocks as being good and bad. Furthermore, one will experience an imbalanced data problem.

In this thesis, we have focused on creating an overall model, that works in every time period. However, as times changes, fundamental key-figures obtained in the past may not be relevant in newer times. Therefore, it can be interesting to construct a model on a rolling basis. However, it is necessary to have a reasonable size of observations in the training datasets to avoid high bias in the models.

Bibliography

"AlphaZero AI beats champion chess program after teaching itself in four hours"
2017. https://www.theguardian.com/technology/2017/dec/07/alphazero-google-deepmind-ai-beats-champion-program-teaching-itself-to-play-four-hours. Dec-2017.

Asness, C. S., A. Frazzini, and L. H. Pedersen
2018. Quality minus junk. *Springer*.

Asness, C. S., T. J. Moskowitz, and L. H. Pedersen
2013. Value and momentum everywhere. *The Journal of Finance*.

"Big Data: Are you ready for blast-off?"
2014. https://www.bbc.com/news/business-26383058. Mar-2014.

Christensen, J. I.
2015. *Value and Momentum - a winning combination*. Jyske Capital.

Engle, R., R. Ferstenberg, and J. Russell
2012. Measuring and modeling execution cost and risk. *The Journal of Portfolio Management*.

"FactSet"
2019. https://en.wikipedia.org/wiki/FactSet. Jun-2019.

Fama, E. F.
1970. Efficient capital markets: A review of theory and empirical work. *American Finance Association*.

"Feature Selection Techniques in Machine Learning with Python"
2018. https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-wit
Oct-2018.

Frazzini, A. and L. H. Pedersen

2013. Betting against beta. *Elsevier*.

"Global Industry Classification Standard"

2019. `https://en.wikipedia.org/wiki/Global_Industry_Classification_Standard`. Feb-2019.

Hastie, T., R. Tibshirani, and J. Freidman

2009. *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer.

"How many stocks are in the S&P 500?"

2015. `https://www.nbc-2.com/story/29616116/how-many-stocks-are-in-the-sp-500`. Jan-2015.

Huang, W., Y. Nakamori, and S.-Y. Wang

2002. Forecasting stock market movement direction with support vector machine. *Elsevier*.

Huerta, R., F. Corbacho, and C. Elkan

2013. Nonlinear support vector machine can systematically identify stocks with high and low future returns. *IOS Press*.

"International Financial Reporting Standards"

2019. `https://en.wikipedia.org/wiki/International_Financial_Reporting_Standards`. Apr-2019.

"Interquantile range"

2019. `https://en.wikipedia.org/wiki/Interquartile_range`. Jun-2019.

Johnson, R. and D. Wichern

2013. *Applied Multivariate Statistical Analysis*. Pearson Education UK.

Lantz, B.

2015. *Machine Learning with R*. PACKT.

"Make way for the robot stock pickers"

2016. https://www.ft.com/content/84bb5c72-37a9-11e6-9a05-82a9b15a8ee7. Jun-2016.

Malkiel, B. G.

2003. The efficient market hypothesis and its critics. *Journal of Economic Perspectives*.

"Moore's law"

2019. https://en.wikipedia.org/wiki/Moore%27s_law. Jun-2019.

"MSCI"

2019. https://en.wikipedia.org/wiki/MSCI. Jul-2019.

O'Shaughnessy, J. P.

2005. *What Works on Wall Street*. McGraw-Hill.

Pedersen, L. H.

2015. *Efficiently Inefficient How Smart Money Invests and Market Prices Are Determined*. Princeton University Press.

Shen, H. J. S. and T. Zhang

2012. Stock market forecasting using machine learning algorithms. *Standford University*.

Tsay, R. S.

2002. *Analysis of Financial Time Series*. John Wiley & Sons, Inc.

Tweedy, B. C. L.

2009. *What Has Worked in Investing*. Tweedy, Browne Company LCC.

"Value investing: is the age-old strategy dead?"

2019. https://www.irishtimes.com/business/personal-finance/value-investing-is-the-age-old-strategy-dead-1.3842954. Apr-2019.

"Waymo Technology"

2019. https://waymo.com/tech/. Jan-2019.

"Welcome to WRDS!"

2019. `https://wrds-web.wharton.upenn.edu/wrds/`. Jan-2019.

"What is Machine Learning?"

2019. `https://emerj.com/ai-glossary-terms/what-is-machine-learning/`. Feb-2019.

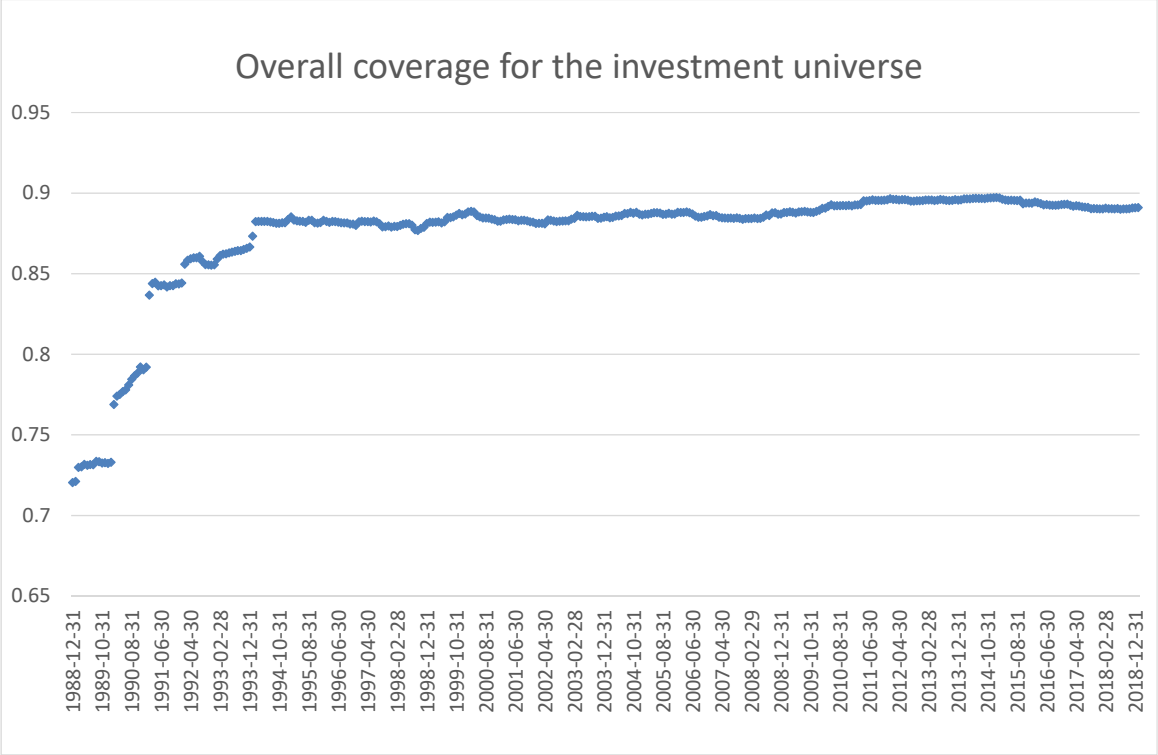
A Appendix - Charts and Tables

A.1 List of Fundamental Key-Figures From FactSet

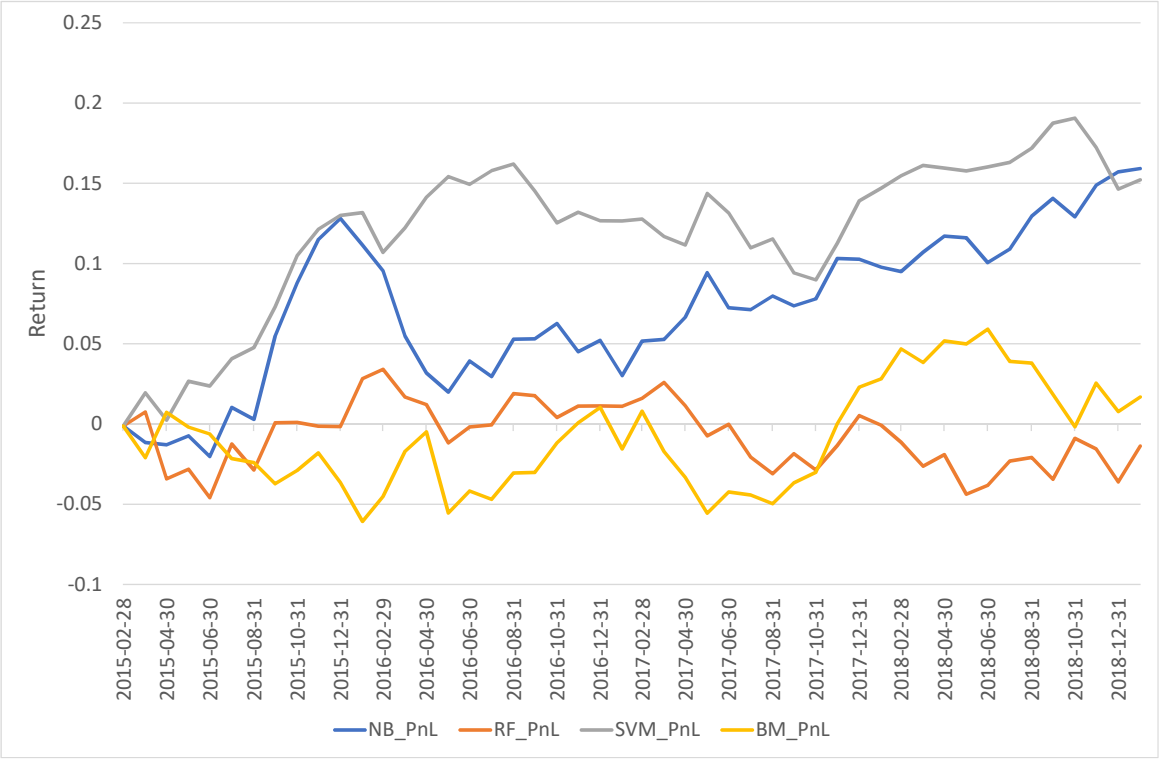
Fundamental key-figure	**FactSet formula**	**Category**
Current Assets	FF_ASSETS_CURR	Asset Turnover Analysis
Cash & ST Investments	FF_CASH_ST	Asset Turnover Analysis
Inventories	FF_INVEN_TURN	Asset Turnover Analysis
Fixed Assets	FF_PPE_NET	Asset Turnover Analysis
Receivables	FF_RECEIV_TURN	Asset Turnover Analysis
Cash Dividend Coverage Ratio	FF_CASH_DIV_COVG_RATIO	Coverage
Total Debt_EBITDA	FF_DEBT_EBITDA_OPER	Coverage
Fixed-charge Coverage Ratio	FF_EBIT_OPER_FIX_CHRG_COVG	Coverage
EBIT_Interest Expense (Int. Coverage)	FF_EBIT_OPER_INT_COVG	Coverage
Total Debt_Total Assets	FF_DEBT_ASSETS	Leverage
Total Debt_Equity	FF_DEBT_EQ	Leverage
LT Debt_Total Capital	FF_LTD_TCAP	Leverage
Net Debt_Total Capital	FF_NET_DEBT_TCAP	Leverage
Total Debt_Total Capital	FF_TOT_DEBT_TCAP_STD	Leverage
Cash Ratio	FF_CASH_RATIO	Liquidity
Current Ratio	FF_CURR_RATIO	Liquidity
Asset Turnover (x)	FF_ASSET_TURN	Operating Efficency
Assets_Employee (actual)	FF_ASSETS_PER_EMP	Operating Efficency
Payables Turnover (x)	FF_PAY_TURN	Operating Efficency
Revenue_Employee (actual)	FF_SALES_PER_EMP	Operating Efficency
Working Capital Turnover	FF_SALES_WKCAP	Operating Efficency
Book Value per Share	FF_BPS	Per Share
Dividends per Share	FF_DPS	Per Share
EBIT (Operating Income) per Share	FF_EBIT_OPER_PS	Per Share
EPS	FF_EPS	Per Share

Fundamental key-figure	**FactSet formula**	**Category**
Free Cash Flow per Share	FF_FREE_PS_CF	Per Share
Cash Flow per Share	FF_OPER_PS_NET_CF	Per Share
Dividend Payout Ratio	FF_PAY_OUT_RATIO	Per Share
Sales per Share	FF_SALES_PS	Per Share
Cash Flow Return on Invested Capital	FF_CF_ROIC	Profitability
Free Cash Flow Margin	FF_FREE_CF	Profitability
Gross Margin	FF_GROSS_MGN	Profitability
Net Margin	FF_NET_MGN	Profitability
Net Operationg Cash Flow	FF_OPER_CF	Profitability
Operating Margin	FF_OPER_MGN	Profitability
Pretax Margin	FF_PTX_MGN	Profitability
Return on Assets (pct)	FF_ROA	Profitability
Return on Common Equity	FF_ROCE	Profitability
Return on Equity (pct)	FF_ROE	Profitability
Return on Invested Capital	FF_ROIC	Profitability
Return on Total Capital	FF_ROTC	Profitability
Price_Earnings	FF_PE	Valuation
Price_Book Value	FF_PB	Valuation
Price_Sales	FF_PS	Valuation
Price_Cash Flow	FF_PCF	Valuation
Price_Free Cash Flow	FF_PFCF	Valuation
Total Debt_Enterprise Value	FF_DEBT_ENTRPR_VAL	Valuation
Dividend Yield	FF_DIV_YLD	Valuation
Enterprise Value_EBIT	FF_ENTRPR_VAL_EBIT_OPER	Valuation
Enterprise Value_Sales	FF_ENTRPR_VAL_SALES	Valuation

A.2 Monthly Overall Coverage of Variables for the Investment Universe



A.3 Realised Turnover for all the Strategies in the Test period



A.4 Confusion Matrix for the BM Strategy for the Whole Period

```
Confusion Matrix and Statistics

          Reference
Prediction   -1    1
        -1 7714 7172
        1  7504 7382

Accuracy : 0.5071
95% CI : (0.5014, 0.5127)
No Information Rate : 0.5112
P-Value [Acc > NIR] : 0.92223

Kappa : 0.0141
Mcnemars Test P-Value : 0.00629

Sensitivity : 0.5069
Specificity : 0.5072
Pos Pred Value : 0.5182
Neg Pred Value : 0.4959
Prevalence : 0.5112
Detection Rate : 0.2591
Detection Prevalence : 0.5000
Balanced Accuracy : 0.5071

Positive Class : -1
```

A.5 Confusion Matrix for the Strategies in the Test Period

A.5.1 Confusion Matrix SVM

```
Confusion Matrix and Statistics

          Reference
Prediction   -1    1
        -1 1096  985
        1  1029 1052

Accuracy : 0.5161
95% CI : (0.5008, 0.5314)
No Information Rate : 0.5106
P-Value [Acc > NIR] : 0.2427

Kappa : 0.0322

Mcnemars Test P-Value : 0.3380

Sensitivity : 0.5158
Specificity : 0.5164
Pos Pred Value : 0.5267
Neg Pred Value : 0.5055
Prevalence : 0.5106
Detection Rate : 0.2633
Detection Prevalence : 0.5000
Balanced Accuracy : 0.5161

Positive Class : -1
```

A.5.2 Confusion Matrix NB

```
1   Confusion Matrix and Statistics
2
3   Reference
4   Prediction   -1    1
5              -1 1079 1002
6              1   988 1093
7
8   Accuracy : 0.5219
9   95% CI : (0.5066, 0.5371)
10  No Information Rate : 0.5034
11  P-Value [Acc > NIR] : 0.008845
12
13  Kappa : 0.0437
14
15  Mcnemars Test P-Value : 0.770731
16
17  Sensitivity : 0.5220
18  Specificity : 0.5217
19  Pos Pred Value : 0.5185
20  Neg Pred Value : 0.5252
21  Prevalence : 0.4966
22  Detection Rate : 0.2593
23  Detection Prevalence : 0.5000
24  Balanced Accuracy : 0.5219
25
26  Positive Class : -1
```

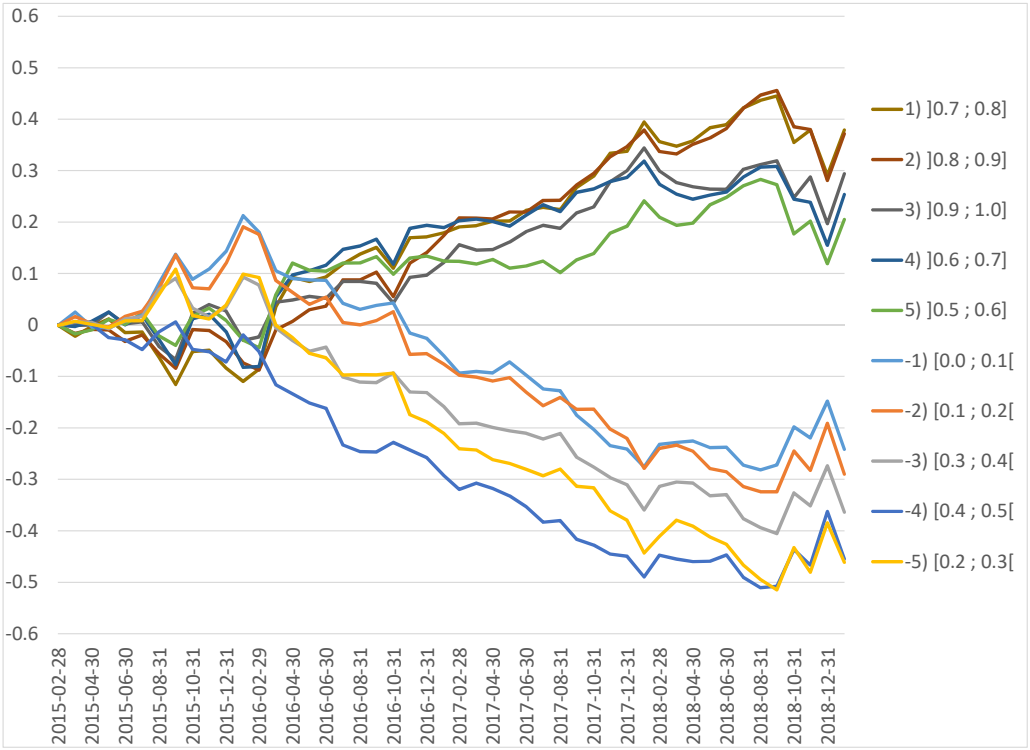
A.5.3 Confusion Matrix RF

```
1   Confusion Matrix and Statistics
2
3   Reference
4   Prediction   -1    1
5              -1 1084  997
6              1  1089  992
7
8   Accuracy : 0.4988
9   95% CI : (0.4835, 0.5141)
10  No Information Rate : 0.5221
11  P-Value [Acc > NIR] : 0.99875
12
13  Kappa : -0.0024
14  Mcnemars Test P-Value : 0.04632
15
16  Sensitivity : 0.4988
17  Specificity : 0.4987
18  Pos Pred Value : 0.5209
19  Neg Pred Value : 0.4767
20  Prevalence : 0.5221
21  Detection Rate : 0.2605
22  Detection Prevalence : 0.5000
23  Balanced Accuracy : 0.4988
24
25  Positive Class : -1
```

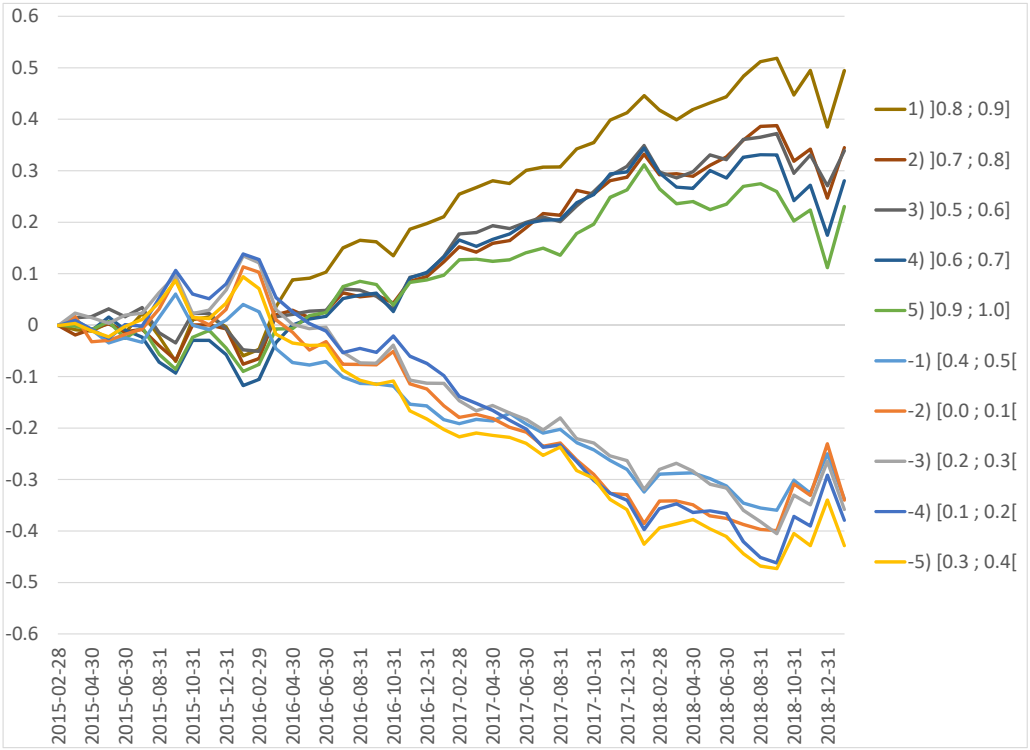
A.5.4 Confusion Matrix BM

```
Confusion Matrix and Statistics

          Reference
Prediction   -1    1
        -1 1140 1159
         1  1159 1140

Accuracy : 0.4959
95% CI : (0.4813, 0.5104)
No Information Rate : 0.5
P-Value [Acc > NIR] : 0.7174

Kappa : -0.0083
Mcnemars Test P-Value : 1.0000

Sensitivity : 0.4959
Specificity : 0.4959
Pos Pred Value : 0.4959
Neg Pred Value : 0.4959
Prevalence : 0.5000
Detection Rate : 0.2479
Detection Prevalence : 0.5000
Balanced Accuracy : 0.4959

Positive Class : -1
```

A.6 Realised Turnover for Different Likelihood Prediction Intervals for the SVM



A.7 Realised Turnover for Different Likelihood Prediction Intervals for the RF



A.8 Scatterplot with a Linear Regression of the Return against the Hit Ratio

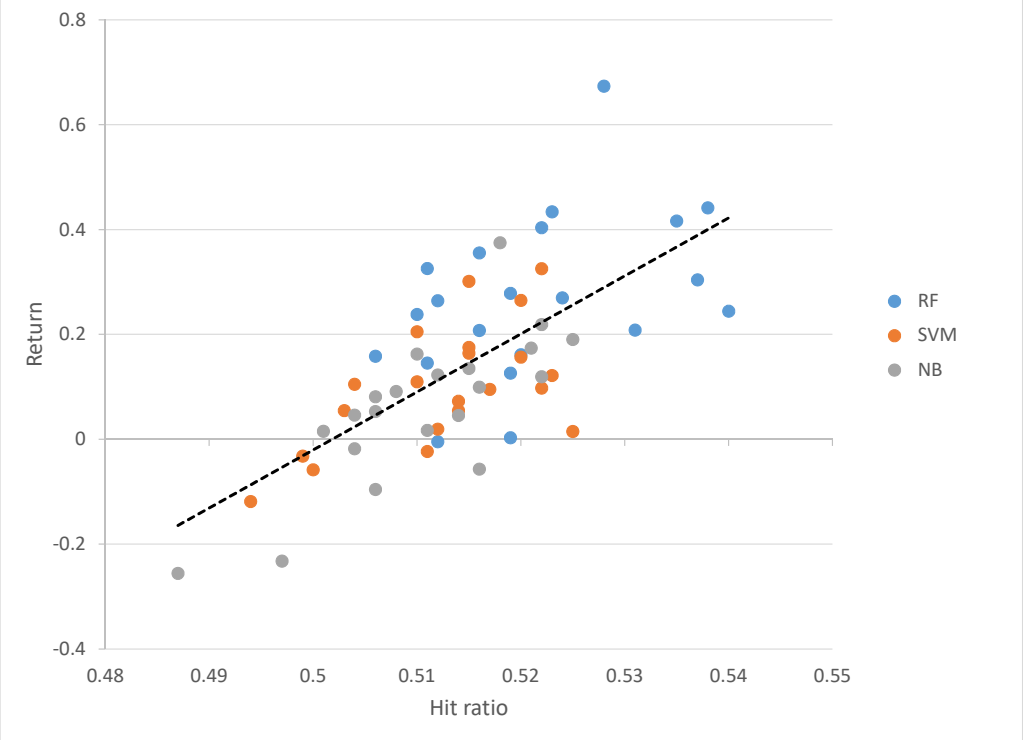
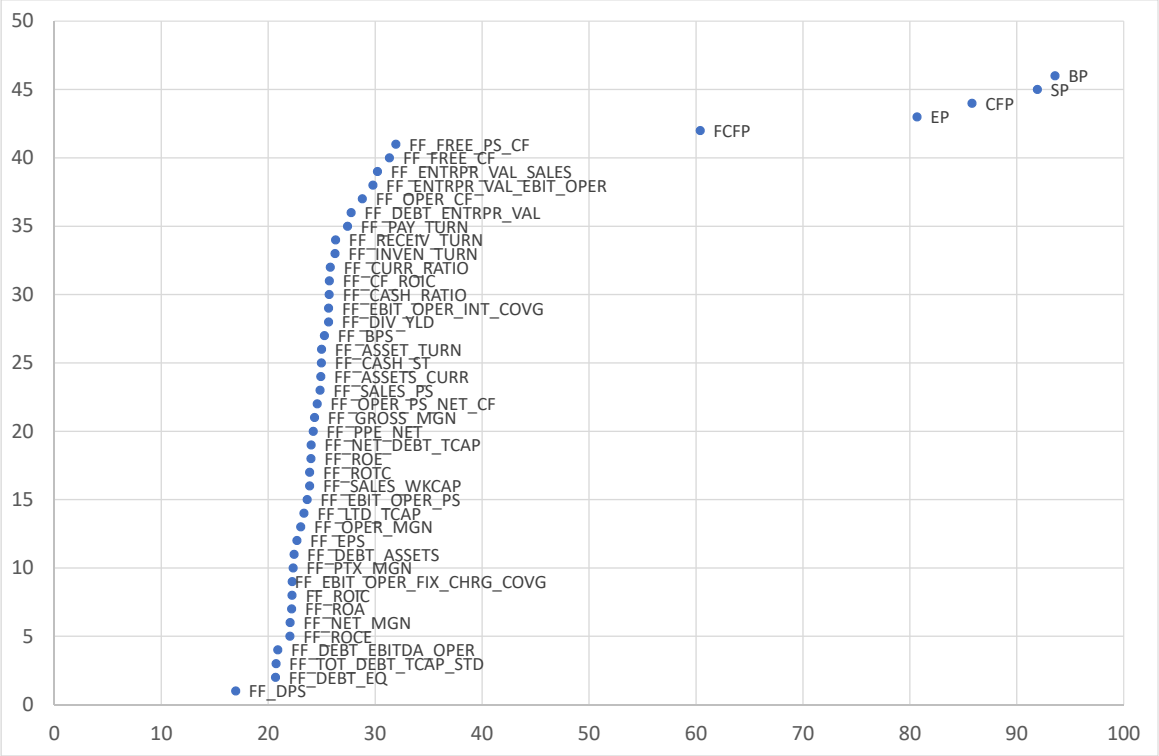


Figure A.1: Predicted return and hit ratio for a four year period on a one-year rolling basis from 28 February 1991 to 31 January 2015.

A.9 Average Variable Importance for Random Forest



B Appendix - USB Drive

B.1 Data

In the Data folder, one can find the data from Compustat used to identify the S&P 500 Index constituents. Moreover, the folder includes the fundamental key-figures, prices, industries and reporting dates from FactSet. Furthermore, the data folder also contains the processed data, VBA codes and charts used in the analysis and results. The data from Compustat and FactSet is confidential material.

B.2 R Scripts

In the R folder, one can find the R scripts used in the analysis. The insourcing of packages, self-programmed functions and data are combined in the file called ratioanalysis.R.