BI Norwegian Business School - campus Oslo

GRA 19703

Master Thesis

Thesis Master of Science

To what extent can machine learning algorithms predict long-term stock price directions on Oslo Børs and Nasdaq Stockholm?

Navn: Christian Eskil Myrseth, Martin Navestad

Start: 15.01.2021 09.00

Finish: 01.07.2021 12.00

BI Norwegian Business School

MSc in Finance

Title:

To what extent can machine learning algorithms predict long-term stock price directions on Oslo Børs and Nasdaq Stockholm?

Supervisor:

Kjell Jørgensen

Acknowledgment

This master thesis was written as a final part of our Master of Science degree in Finance at BI Norwegian Business School. Topics covered in the thesis are predictions, Machine Learning, and portfolio construction in the stock market. With no prior knowledge in the field of Machine Learning and time series forecasting, the work has proven to be both challenging and exciting.

We want to thank our supervisor Kjell Jørgensen for his valuable insights and excellent guidance. Additionally, we would like to express our gratitude towards friends and family who have contributed with assistance and motivation throughout the final semester of our master's degree.

BI Norwegian Business School Oslo

June 2021

Abstract

This study aims to examine what value Machine Learning algorithms give when trading from a long-term perspective. Historically, it has been hard to consistently gain an excess return from investing in the stock market. The high complexity and number of factors affecting the markets are complicating this task. In our research, the performance of Machine Learning algorithms such as Random Forest and Support Vector Machine were analyzed both with and without feature selection methods. The models' predictions and our constructed portfolios were compared to two benchmarks (Dummy Classifier and OSEBX/OMX30 index).

From our analyses, the Random Forest model with SVM-RFE feature selection was found to give the most promising prediction results, and the performance was analyzed both during the whole backtesting period and through times of crisis. When implementing a simple trading strategy utilizing the predictions, we found the same model with a portfolio construction of 30 companies to outperform both the benchmarks and other algorithms from late 2006 until the first quarter of 2021. During times of crisis, our reference Machine Learning model did not significantly outperform the benchmarks. However, it showed uplifting results in economic rebounds. Thus, the highest potential for the Machine Learning model might be its ability to identify the best-performing stocks in periods after financial recessions.

Table of Contents

1.0 INTRODUCTION

The stock market we know today has existed for several centuries, where the Amsterdam Stock Exchange, which was established in 1602, is considered to be the oldest (Braudel, 1982). Throughout time, the trading mechanisms in the stock markets have changed, in line with a growing industry. In 2018, the world stock exchanges reached a record-high market capitalization of $68.65 trillion (World Bank, n.d.), highlighting its strong impact on the global economy. To profit from the stock market, traders need to predict trends in stock market behavior. With the vast amount of capital being traded, stock market prediction has been of great interest for investors, and there have been several approaches to maximize returns (Khan et al., 2020). The most popular methods involve fundamental analysis, technical analysis, or a combination of these.

In the recent century, the quant revolution's upspring has forced a shift in trading behavior among investors. Access to data and increased computer power have been the most important factors leading to this revolution, which has also introduced a new approach to stock trading, technological analysis. *Machine Learning*, a branch of Artificial Intelligence (AI), is the most popular outbreak from the new quant revolution in predicting the stock markets (Rasekhschaffe & Jones, 2019).

Because of the great interest and potentially huge profits involved in making accurate stock market predictions, there have been several contributions to find the "perfect" model. Early research has relied heavily on purely traditional econometric techniques, such as regression, without successful results. It may be difficult to forecast any dynamic relationships between potential predictors and expected returns due to several reasons. For instance, financial data are considered noisy; factors may be subject to multicollinearity, and relationships between predictors and expected returns may be nonlinear, variable and/or contextual. Since the 2008 Financial Crisis, investors using quantitative factor models have struggled. As a result, many traders have focused their attention on developing models that use past data to dynamically "learn" from (Rasekhschaffe & Jones, 2019).

A study from Nti et al. (2019) undertakes a systematic and critical review of 122 past research works posted in academic journals within the period 2008-2018 in the area of stock market prediction using Machine Learning. Their findings revealed that approximately 66% of the studies reviewed were based on a technical analysis approach, 23% relied on fundamental analysis, and the remaining 11% used a combination of the two. In addition, Nti et al. (2019) argue that Support Vector Machine and Artificial Neural Networks were the most used Machine Learning algorithms for predicting the stock market.
However, with the increasing attention and importance of Machine Learning in quantitative finance, there have also been valid criticisms against such dynamic models. The debates are surrounding whether Machine Learning techniques can be considered as practical investment tools. Although there is no doubt that such algorithms can help detect contextual and nonlinear relationships within large datasets, the risk of *overfitting* poses a significant challenge when trying to extract signals from noisy historical data (Rasekhschaffe & Jones, 2019).

Our study is a contribution to a fast-growing literature in the use of Machine Learning for stock prediction. We have narrowed our research to companies listed on Oslo Børs and Nasdaq Stockholm for the period 2001-2021. Easy access to complete datasets has driven previous work to focus on large financial markets such as the US and the Asia Pacific. Additionally, it is challenging and time-consuming to collect complete financial data sets from Nordic companies given their relatively small size in the global economy. Thus, to the best of our knowledge, there does not exist any published research on the Scandinavian financial markets within the area of Machine Learning for long-term stock prediction.

This motivates us to expand existing knowledge of the underlying mechanisms for this local market. The main focus of our paper is to present an indication of what value Machine Learning can provide when making long-term stock predictions and trading based on fundamental and momentum analysis, as well as hand-picked macro variables, both in terms of average performance and during times of crisis. First, we present relevant theory and data collection before moving to our methodology, results, and conclusion. Lastly, we continue the research by presenting a variable importance table to identify the features that best predict the company's stock returns. Our analysis is based primarily on historical fundamentals and momentum in the stock prices for companies listed on Oslo Børs and Nasdaq Stockholm, in addition to macro variables for both the Scandinavian and the US markets. Thus, the main research question for our study is: *To what extent can machine learning algorithms predict long-term stock price directions on Oslo Børs and Nasdaq Stockholm?*

Methodologically, we have found supervised Machine Learning models such as Support Vector Machine (SVM) and ensemble learning models (Random Forest (RF)) to be best suitable for our research. The time series is divided into training samples and test samples to evaluate the models. We use the training samples of the data to fit the models with SVM and RF algorithms, whereas the test samples are applied as a baseline to evaluate the out-of-sample performance. Two feature selection algorithms will be used to remove factors considered low in importance, attempting to improve the model by reducing complexity and overfitting (Tatsat et al., 2020). The rolling window approach is utilized for estimation. In addition to comparing the models to each other, we also evaluate the performance against a standard benchmark model, Dummy Classifier, as well as a weighted average and a 50/50 split between the Oslo Børs Benchmark Index (OSEBX) and OMX Stockholm 30 Index (OMXS30).

There are several ways our work contributes to the existing literature within the field of stock prediction with Machine Learning.
First, the research paper by Ballings, Van den Poel, Hespeels & Gryp (2015) called "*Evaluating multiple classifiers for stock price direction prediction*" was the first of its kind to evaluate several Machine Learning models in stock prediction. Previous research has tended to narrow their attention to one single model. The study presents an excellent overview of the current standings of both theoretical and empirical aspects of Machine Learning models for stock prediction. Ballings et al. (2015) gathered yearly data from 5767 listed European companies and utilized 81 specific fundamentals and general economic features to predict stock returns. The study used ensemble methods such as Random Forest (RF), AdaBoost (AB) and Kernel Factory (KF), and benchmarked against single classifier models limited to Neural Networks (NN), Logistic Regression (LR), Support Vector Machines (SVM) and K-Nearest Neighbours (KNN) in predicting

the direction of stock prices one year ahead. Their main findings were that RF was the best performer, followed by SVM, KF and AB, emphasizing the importance of including ensembles in the sets of algorithms for stock prediction.

Further, the study by Yuan et al. (2020) showed that models such as Random Forest (RF), Support Vector Machine (SVM) and Artificial Neural Networks (ANN) have predictable power in the Shanghai Stock Exchange (SSE). The paper utilizes company fundamentals, momentum elements, along with volatility and other technical factors from nearly 3000 listed companies over an 8-year period. In the study, Yuan et al. (2020) found the RF model with feature selection through the same RF algorithm to be the best performer, outperforming the SHCI and HS300 benchmark indexes.

Our study utilizes company fundamentals, momentum factors, and macro variables in stock prediction by several Machine Learning models, similar to the work by Ballings et al. (2015) and Yuan et al. (2020). However, we focus on predicting quarterly stock price directions and narrow our geographical area to the Norwegian and Swedish financial markets. Further, to test our models through different economic periods (including structural breaks), we backtest our performance for over 14 years. Ballings et al. (2015) and Yuan et al. (2020) have chosen a smaller backtesting period equal to six years and eight years, respectively. Unlike the study by Ballings et al. (2015), which only emphasizes on evaluating the models against each other, we also present a more practical view on the performance of Machine Learning algorithms by benchmarking against both the indexes and constructed Dummy Classifier models.

Second, we refer to the research paper by Rasekhschaffe & Jones (2019): "*Machine Learning for Stock Selection*" as a great example of using Machine Learning techniques to predict stock returns while limiting the risk of overfitting. The study presents two primary ways to reduce the problem, namely feature engineering (which can increase the signal-to-noise ratio by transforming the data) and forecast combinations (reducing noise by focusing on robust relationships regardless of what forecasting technique and training window are being used). Rasekschaffe & Jones (2019) also suggest using company fundamentals in predicting the stock returns. However, the study does not focus on adding macroeconomic variables to enhance the stock predictions, which is also illustrated in other research papers such as Alberg & Lipton (2018). The close relationship between the financial markets and the global economy motivates us to narrow the absence of literature within this topic by investigating the influence of macroeconomic data on stock predictions. Besides, we refer to the study by S.-S. Chen (2009) on the influence of macroeconomic factors in predicting bear markets. Empirical evidence from the paper implies that macroeconomic variables, especially yield curve spreads and inflation rate, are valuable predictors of recessions in the US market. Hence, we want to examine the performance of our models, particularly during financial recessions, and test the same hypothesis of macroeconomic variables towards the Scandinavian market.

As illustrated by the systematic review from Nti et al. (2019), previous academic papers on the subject are primarily driven towards short-term stock prediction and portfolio construction. From the articles on technical analysis, the predictive timeframe of most papers was 1-day ahead, with the maximum being 1-month. Although some papers have been narrowed to 1-year ahead stock predictions, such as Ballings et al. (2015) and Scholz et al. (2015), we have not come across previous work on quarterly predictions and trading. Thus, we are motivated to fill the existing gap in the literature by focusing on the quarterly perspective both in terms of stock prediction and portfolio construction.

From our analyzes, we found both of our models to provide valuable insights in predicting stock price directions and earning excess returns. The study implied that the most effective forecasting algorithm was the Random Forest model with a Support Vector Machine Recursive Feature Elimination (SVM-RFE) feature selection, which gained a 15.17 percentage points higher annualized return than the weighted average benchmark index. Even though the model performed significantly better over the whole backtesting period, its performance during crisis periods was not better than the same benchmark in terms of returns. This contradicts the hypothesis by S.-S. Chen (2009) that the inclusion of macroeconomic variables makes it easier to predict bear markets. However, an interesting observation was found during the first periods after a financial recession, as our model produced significantly higher returns than the benchmarks. Hence, overall, our research suggests that Machine Learning models can produce excess value in long-term stock trading in the Scandinavian financial markets.

2.0 TRADING IN THE STOCK MARKET

There are several approaches to stock prediction and trading, whereas the most popular methods can be categorized into either fundamental analysis or technical analysis. Furthermore, with the recent development of data availability, social media and increased computer power, several other terms within the stock market have been introduced and gained more importance. For the subsequent sections, we present an overview of common characteristics involved in stock market prediction and trading, which we find relevant to our study.

Fundamental analysis

Fundamental analysis combines a company's financial statement along with information on its peers and the operating market to determine the intrinsic value of a stock (often denoted as the stock price). If the stock is believed to be incorrectly priced, the investor would either short or long the stock with an expectation that the mispricing will correct itself. The emergence of Machine Learning has enabled a more automatized procedure for stock market prediction when using unstructured data. Previous research argues that in some cases, a higher prediction accuracy for the long-term stock-price movement was achieved using fundamental analysis. Hence, it is not suitable for short-term stock-price changes (Nti et al., 2019).

Technical analysis

Technical analysis, also known as charting, involves using different charts of historical market prices and other technical indicators to predict mostly short-term changes in the stock market (Nti et al., 2019). It has proliferated for several decades, and traders have increasingly turned to the fundamentals of technical analysis in predicting stock market behavior (J. Chen, 2010). However, previous research presents different conclusions on whether a technical analysis approach adds value to stock selection, mainly because it is considered to be strongly related to the "random walk" (Roscoe & Howorth, 2009).

Technology methods

Technology methods involve a fundamental or technical approach combined with Machine Learning and computational techniques for analyzing stock market behavior. The most popular models used are the Hidden Markov Model, Neuro-Fuzzy Inference System, Time Series Analysis, Genetic Algorithm, Regression, Support Vector Machine, Mining Association Rules, and Principal Component Analysis. (Nti et al., 2019).

2.1 Why is the stock market hard to predict?

Producing accurate forecasts is subject to many obstacles. The following section presents theories on why predicting stock markets may be exceptionally hard. According to several finance theories, financial markets are in equilibrium, asset prices are efficient (implying that prices are in line with the theoretical and rational market price), and bubbles do not exist (potential large market swings result from exogenous events that are unpredictable). Further, the finance industry is affected by a large number of factors. Company-specific factors (e.g., liquidity, financial quality, and company news) and macroeconomic factors (e.g., interest rates, economic stability, political stability, and inflation) are of great importance. Hence, finding an optimal subsample of data variables to produce accurate forecasts may be challenging. Additionally, macroeconomic variables are subject to high uncertainty. Given that such factors are expected to be updated frequently, an inevitable trade-off among timeliness and reliability occurs when the raw data are only published in pieces and gradually through time (Bier & Ahnert, 2001). One of the most highlighted arguments against the stock markets' predictability is delivered by Fama and Malkiel, which argue that the markets follow a stochastic process and hence are not predictable. They present two famous hypotheses: the Efficient Market Hypothesis and the Random Walk Hypothesis (Nti et al., 2019).

2.1.1 The Efficient Market Hypothesis

The Efficient Market Hypothesis (EMH) is a theory by Fama (1970), which argues that the market is efficient and stock prices reflect all available information. Consequently, this theory makes it impossible to consistently outperform the market on a risk-adjusted basis since new information is the only factor that moves asset prices. The Efficient Market Hypothesis is often considered in three forms: weak, semi-strong, and strong. The weak-form hypothesis claims that all information derived from past prices is reflected in the current stock price. In the semi-strong form, the hypothesis states that the current stock price reflects all publicly available information. Lastly, the strong-form of the EMH argues that the stock price reflects all information, including insider information (Bodie et al., 2018). However, Grossman and Stiglitz (1980) argue that such an efficient market is impossible due to costly information. In later years, the hypothesis has been challenged by the upbringing of behavioral finance, which is the study of financial markets with the perspective of cognitive psychology and the limits to arbitrage. Unlike the Efficient Market Hypothesis, behavioral finance uses models that are not fully rational, either due to preferences or misbeliefs (Ritter, 2003).

2.1.2 The Random Walk Hypothesis

According to the Random Walk Hypothesis, changes in stock prices are random and cannot be predicted. Thus, price patterns and trends are not usable in predicting the future values of financial assets. In its simplest form, the random walk model is defined as:

$$X_t = X_{t-1} + \epsilon_t \qquad \textit{Equation 1}$$

where X_t is the process and ϵ_t is independently and identically distributed with zero mean and variance σ_t^2 (Nkemnole, 2016). The Random Walk Hypothesis casts doubt and presents challenges on both the technical and fundamental approaches for stock prediction. To this date, empirical evidence is in great support of the random walk model. For a technical analyst, there would be no real value in stock prediction if the Random Walk Hypothesis holds. Regarding the fundamental analyst, more challenges are related to showing that a simple random selection would be as good as a complicated fundamental procedure (Fama, 1995).

2.2 Simulated trading environment

There are several methods to test the performance of different trading models, with the best being the actual performance through real-life trading. However, this is a potentially expensive method, and most developers would like to have a clear assessment of their model before employing it in real trading. Therefore, it is essential to create other reliable tests to examine the performance. In our paper, we have chosen to design a simulated trading environment through portfolio backtesting.

2.2.1 Backtesting

Backtesting is a method used to evaluate the viability and performance of a trading strategy using historical data. According to Piard (2013), several potential pitfalls are involved in designing a reliable backtesting environment. First, one needs to consider the time factors involved and avoid *look-ahead bias* by not including information that is not available when decisions are made. Second, another factor to consider is the *survivorship bias*, meaning an investment universe where only the current stocks are included. Third, transaction costs are difficult to correctly introduce to the backtesting, as you would need to actually perform the trade to know them exactly. Fourth, including shorting can be challenging as it typically involves identifying a potential lender. Additionally, the cost of lending and the amount available is generally unknown. Lastly, one should include a long time period in the backtesting to avoid basing the trading strategy solely on outliers. Even though one manages to avoid such errors, many still argue that the flawless backtest is non-existing. The fact that backtesting is performed ex-post makes selection bias a dominant issue. That is, by testing several strategies on past data, some of them will likely yield satisfying results due to overfitting (Lopez de Prado, 2018). Thus, there is no guarantee that good performing backtests will yield similar results in the future.

2.2.2 Transaction costs

One of the most prominent issues in backtesting is the simulation of transaction costs. Total transaction costs can be split into direct and indirect costs. Direct transaction costs involve the commissions paid for each trade, whereas indirect transaction costs refer to both implementation shortfall and price impact. Implementation shortfall is the opportunity cost occurring from placing an order that is not immediately executed due to the market drifting. Thus, it is challenging to ensure whether a specific trade would have been possible to perform at a given price. Price impact is linked to the opportunity cost, as it is defined as the difference between the last traded price when the trade is executed and the actual price paid (Ødegaard, 2009). It is near impossible to calculate the exact transaction cost that would have occurred without actually performing the trades. Most academic papers on stock trading with Machine Learning models are trading more frequently, increasing the influence of transaction costs on the overall results. However, by rebalancing the portfolio quarterly, the transaction cost will not have the same effect on the performance.

3.0 DATA

In the subsequent sections, we will present the data used for our analysis and predictions; including sources, choice of features, and challenges related to our dataset. For the experiment, it is vital to have a sufficient number of observations and a complete dataset that covers the fields' important aspects. Thus, the databases from Refinitiv and Bloomberg were considered our best options for collecting company-specific data and macroeconomic variables.

3.1 Data collection

Our final dataset contains 15 011 quarterly observations with 109 different features collected from 115 companies listed on Oslo Børs and 129 listed on Nasdaq Stockholm. The data ranges from March 2000 (31/03-2000) to December 2020 (31/12-2020) and differs between the companies. The company fundamentals are obtained from the Refinitiv database, whereas macroeconomic variables have been collected from both Refinitiv and Bloomberg. The different input features are categorized into the three main groups; Fundamental Factors, Momentum Factors, and Macroeconomic Factors, with several connecting sub-categories. A complete overview of the input features is displayed in Table 3.1 below.

Category	Features
Fundamental Factors	
Liquidity factors	Current Ratio
	Quick Ratio
	Cash Ratio
	Cash Coverage Ratio
	Interest Coverage Ratio
Valuation factors	Price-to-Earnings Ratio
	Price/Book Ratio
	Price/Sales Ratio
	EV/EBITDA
	Market Cap / Free Cash Flow
	Market Cap / Tangible Book Value
	Market to Book Ratio
	EV/Operating Income
Leverage factors	Debt Ratio
	Debt-Equity Ratio
	Long-term Debt Ratio
Financial quality factors	Return on Equity (QTD)
	Return on Assets (QTD)
	Return on Assets (TTM)
	Profit Margin
	EBITDA-margin
	Operating Margin
	Gross Margin
	Cash from operating (TTM) / Net Profit (TTM)
	Cash from operating / Net Profit
	Revenue (TTM) / Market Capitalization
	Net Profit (TTM) / Market Capitalization
	Net Profit / Market Capitalization
	PEG Ratio
	Earnings yield
	Asset Turnover
	Inventory Turnover
	Receivables Turnover
	Payable Turnover
1-Year change in fundamentals	Change in Revenue (YoY)
	Change in EBIT (YoY)
	Change in EBITDA (YoY)
	Change in Net Income (YoY)
	Change in Cash from Operations (YoY)
	Change in Current Assets (YoY)
	Change in Current Liabilities (YoY)
	Change in Total Equity (YoY)
	Change in Total Liabilities (YoY)
	Change in Total Assets (YoY)
Quarterly change in fundamentals	Change in Revenue (QoQ)
	Change in Operating Expenses (QoQ)
	Change in Current Liabilities (QoQ)
	Change in Current Assets (QoQ)
	Change in Total Assets (QoQ)
	Change in Net Income (QoQ)
	Change in Total Liabilities (QoQ)
	Change in Long-Term Debt (QoQ)
	Change in Earnings per Share (QoQ)
	Change in Intangible Assets (QoQ)
	Change in Total Equity (QoQ)
	Change in Enterprise Value (QoQ)
	Change in Profit Margin (QoQ)
Momentum Factors	1-Month Return
	2-Month Return
	3-Month Return (Open Price)

	3-Month Return (Close Price)
	6-Month Return
	12-Month Return
	Last Quarter's Open Price
	Open Price 2-M
	Open Price 1-M
	Open Price
Macroeconomic Factors	Consumer Price Index (YoY)
	Price of Brent Oil
	3-Month Interbank Rate
	3-Month Government Bond
	10-Year Government Bond
	Unemployment Rate
	Gold Price
	Quarterly Change in GDP
	12-Month Change in US Interest Rate
	6-Month Change in US Interest Rate
	3-Month Change in US Interest Rate
	2-Month Change in US Interest Rate
	1-Month Change in US Interest Rate
	US Bid Rate
	12-Month Change in NOR/SWE Interest Rate
	6-Month Change in NOR/SWE Interest Rate
	3-Month Change in NOR/SWE Interest Rate
	2-Month Change in NOR/SWE Interest Rate
	1-Month Change in NOR/SWE Interest Rate
	NOR/SWE Bid Rate
	12-Month Change in Brent Price
	6-Month Change in Brent Price
	3-Month Change in Brent Price
	2-Month Change in Brent Price
	1-Month Change in Brent Price
	Last Quarter Brent Price
	12-Month Change in USD Exchange Rate
	6-Month Change in USD Exchange Rate
	3-Month Change in USD Exchange Rate
	2-Month Change in USD Exchange Rate
	1-Month Change in USD Exchange Rate
	12-Month Change in EUR Exchange Rate
	6-Month Change in EUR Exchange Rate
	3-Month Change in EUR Exchange Rate
	2-Month Change in EUR Exchange Rate
	1-Month Change in EUR Exchange Rate
	Last Quarter Index Return
	Change in Index Return
	Change in Price of Brent (QoQ)
	Change in Gold Price (QoQ)
	Change in 3-Month Government Bond Yield (QoQ)
	Change in 10-Year Government Bond Yield (QoQ)

Table 3.1: Complete overview of input features[1].

In the group Fundamental Factors, we have chosen to collect variables representing the different companies' valuation, growth, equity factor, size, financial quality, profitability, investment, and leverage. The sub-categories are

[1] The variable names are not necessarily the same as used in our programming, but rather a description to have a better understanding of each feature utilized.

carefully selected based on previous studies highlighting their importance in determining cross-sectional asset returns (Fama & French, 2018).

While the performance of individual companies on the stock market is strongly related to performance news (e.g., quarterly reports and other announcements), external factors are also influencing individual stocks and the market. There is contradicting evidence on the long-term relationship between different macroeconomic variables and the stock market, illustrated by the studies from Misra (2018) and Gopinathan & S Raja (2019) on the Indian stock market. The research from Misra (2018) shows that there exists a long-run causality between the macroeconomic variables of the Index of Industrial Production (IIP), inflation, interest rates, gold prices, exchange rate, foreign institutional investment, money supply, and BSE Sensex. A short-term relationship was also discovered between Inflation and BSE Sensex, as well as Money Supply and BSE Sensex.
On the contrary, the research from Gopinathan & S Raja (2019) argues that the conventional *Engle and Granger (1987)* and *Phillips and Ouliaris (1990)* tests show no relationship between stock prices and other macroeconomic variables. Despite the contradicting evidence, macroeconomic variables have proven to be leading indicators of predicting the bear stock market (S.-S. Chen, 2009). Thus, we have chosen to include *unemployment rates*, *interest rates*, and different *exchange rates* for Norway and Sweden to better understand the overall economic situation in the respective countries.
Brent Crude prices are also collected, given that it has a significant impact on many aspects of the world economy. Norway is the third-largest exporter of natural gas globally, and many companies listed on Oslo Børs are directly exposed to crude oil prices (Norsk Petroleum, 2021). The development of *Brent Crude* prices is therefore considered to have some predictable power in the Norwegian stock market.

Gold Price and the *US 10-year Treasury Yield* represent the development in the world economy. A study by El Hedi Arouri et al. (2015) on world gold prices and stock returns in China, supports the traditional view that the gold asset is considered a safe haven for investors. During recessions or periods with higher volatility, investors tend to move their capital towards gold and thereby push the asset's prices upwards. The *US 10-year Treasury Yield* is considered to be the world's most important interest rate, especially during the last decade. A shift upwards in the 10-year treasury yield can deteriorate stock returns, as investors may have less to profit from higher risk-taking. It also negatively affects the discounted values of company cash flows, especially for growth stocks, given that much of their value is calculated based on future expected income. We therefore often discern a shift from growth stocks towards value stocks when the 10-year yield appreciates (McCormick & Regan, 2021). On the opposite, a plummeting 10-year yield is often correlated with a bear stock market and global economic recession (e.g., Financial Crisis in 2008 and COVID-19 in 2020).

In the category Momentum Factors, we have gathered data related to long-term and shorter-term momentum in individual stock returns. According to behavioral finance, the best prediction of future market movements is that the trend will continue. It conflicts with the belief that all investors are rational, and highlights the role of psychology in the stock market (Chaffai & Medhioub, 2014). A study by Jegadeesh and Titman (1993) supports the view that individual stocks have momentum. The research showed that high-performing stocks over 3 to 12 months are more likely to pursue their positive momentum, whereas underperforming stocks have a higher probability of continuing their bad performances. Another theory within stock price prediction is that market prices follow a martingale, meaning that the best prediction for the next stock price is the current stock price. There is contradicting evidence on this matter, illustrated by a study from Kumar & Maheswaran (2012) on the Indian stock market, which found support for martingale in three out of six indexes.
Based on the previous research, we have chosen to include momentum features ranging from 12-month return until the most recent stock price.

3.2 Dependent variable

The dependent variable of our dataset is the *Stock excess return*, which forms the basis of our *Decision* variable. It represents the individual stock's return in excess of its respective index (OSEBX or OMXS30) from Quarter T to T+1. Since many stocks are upward sloping in a bull market and downward sloping in a bear market, we use the excess return to remove the market trend in the respective quarter. Stocks with a return above the median excess return for a specific quarter are categorized as *buy*, while the observations with a return below the median for the respective quarter are considered a *sell*. Thus, we are shifting the *Decision* variable by T-1 to know if the prices will increase/decrease in the next quarter (see Figure 3.1). Stock returns are usually highly volatile, which could complicate the process of detecting clear patterns in the dependent variable, thus affecting the overall accuracy of our final predictions.

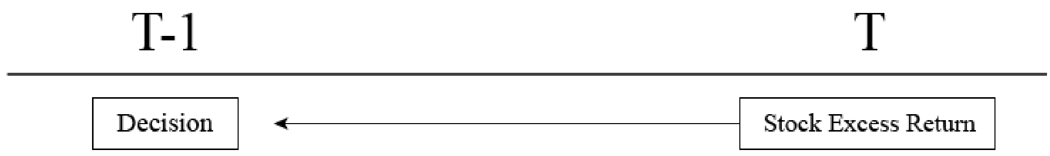


Figure 3.1: The stock excess return in time T is the basis for the decision of our models in T-1.

3.3 Challenges with the data

Ideally, we would have collected a larger time span for our data, as we initially looked at gathering financial reports from 1990 to 2021. An essential factor in building successful Machine Learning models involves having a sufficient number of high-quality data. However, the lack of availability for financial reports before the year 2000 made it difficult.

The final dataset was therefore selected based on whether the financial reports for the respective companies were complete without a significant number of missing values. Thus, even though a more extensive period of financial reports was available for some companies, we chose not to include those if there were insufficient data quality.

All companies were selected based on stock market listings per 30/09-20, which does not lead to a completely realistic backtesting scenario. Ideally, we would have collected data from all companies that have been listed on the stock exchanges for the chosen period, given that all other criteria were met. In this way, firms that have potentially gone bankrupt and/or been delisted throughout our backtesting period would have been included, thus removing survivorship bias. However, we found a large number of delisted companies to be the result of mergers or acquisitions. Additionally, lack of data and time dimensions made it difficult to distinguish whether a delisting resulted from bankruptcy or other factors. Since the difference between the two would have greatly affected the portfolio performance, it is crucial to have the correct data.

Companies on our initial ticker list that, for some reason, proved to lack available data during our backtesting period were included if all other requirements were met. Such data are still valuable when training our algorithms, regardless of whether they have available data until the end of the backtesting period. In section 5.2, we will further elaborate on how we solved this issue for our backtesting trading environment.

As our selected models only accepted complete time series, we also had to exclude variables without an adequate number of observations. Various factors that may have had a predictable power on the stock market returns, such as *dividends* and *R&D*, had to be removed to cope with these issues. Lastly, the desired number of variables versus the number of quality observations available had to be weighted. Choosing a data set with financial reports starting at the earliest available time for every company would produce more observations at the expense of a limited number of variables. Conversely, a data set starting at a later date where financial reports are of sufficient quality provides more variables but a reduced number of observations.

4.0 METHODOLOGY

This chapter centers around our selected Machine Learning models and forecasting techniques. The data preparations are first presented, followed by an introduction of the selected Machine Learning models. The main focus of this study is to identify to which degree Machine Learning algorithms can be utilized to predict stock returns, and not in making discoveries in the Machine Learning area. Thus, we limit this paper to only include a brief introduction to the elements of the Machine Learning methods. The preprocessing of the data was performed in Python[2], whereas *Scikit-learn* was used to conduct the analysis and deploy the Machine Learning algorithms. Scikit-learn is an open-source Machine Learning library that provides tools for preprocessing, model selection, built-in Machine Learning algorithms, models, and evaluation of the models (Pedregosa et al., 2011).

[2] Python is a general-purpose programming language widely used in Data Science and Machine Learning.

4.1 Data preprocessing

After acquiring quarterly fundamental data, macro variables, and stock prices, the data had to be preprocessed before it was employed in the models. The preprocessing consisted of several steps, where the first was to screen our entire dataset and remove insufficient data. The second step was to create new input features and concatenate the data files into one dataframe on a quarterly basis. Third, we applied the sliding window method to split our data into training sets and test sets. The fourth step was to standardize and scale all our features. Lastly, we performed feature selection on the entire feature set to select the most significant variables.

Step 1: Data Screening

After collecting financial reports from all available companies on the Oslo Stock Exchange and Nasdaq Stockholm for our selected period, we made some adjustments due to different challenges. The financial industry, with the sub-industries banking and insurance, are complicated and significantly affected by macro factors, making their stock prices far more uncertain and volatile compared to general industries (Y. Chen et al., 2020). In addition, we found the financial statements to be insufficient. Thus, difficulties in making accurate predictions and the lack of financial information led us to discard all firms connected to the finance industry.

Further, the requirements of having sufficient data to make accurate predictions created a trade-off between the desired number of companies and maintaining adequate observations for each company. After testing several options, we considered a minimum requirement of six years of available financial reports to be sufficient. Hence, all companies with less than six years of financial statements were removed from our original dataset.

Step 2: Feature Engineering and Concatenating

Our second step involved *feature engineering*, which is defined as using domain knowledge to create new features from the original dataset to increase the effectiveness of the Machine Learning model (Kuhn & Johnson, 2019). The features are created from raw data and then transformed into formats compatible with the Machine Learning process. Having correct features is crucial and often mentioned as the most important aspect of making successful predictions with Machine Learning algorithms. Following the creation of features, we additionally transformed most of the variables into stationary form. A stationary time series is defined as data without seasonal effects and trends, i.e., the properties are not dependent on time (see Figure 4.1). Hence, the features should have a constant mean, variance, and autocovariance in both their first and second momentum. It is vital to remove the trend and seasonality as it may affect the value of the time series at different periods of time (Hyndman & Athanasopoulos, 2018).

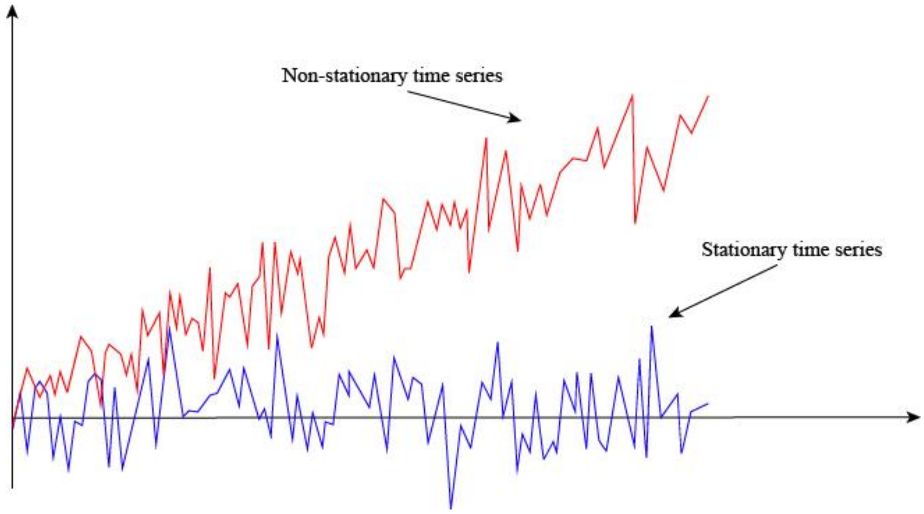


Figure 4.1: Non-stationary and stationary time series.

Although stationarity is not required for many Machine Learning algorithms, it provides a significantly stronger indication of the model's performance. The models are usually more capable of detecting underlying mechanisms rather than just identifying seasonality and trends in the presence of stationarity. The method we used to transform the variables into stationary form was *differencing* (i.e., calculating the differences between each period of observations) and the construction of ratios.

To create our final dataset, we merged all features into one dataframe on a quarterly basis. Further, missing data from our dataset had to be handled. The frequent occurrence of missing values in data, and the fact that most predictive models are unable to handle them, highlights the importance of addressing this prior to running the models (Kuhn & Johnson, 2019). Therefore, variables or financial reports without a sufficient number of observations were excluded from the final dataset. The remaining missing values were filled with the last available observation for that particular variable.

Step 3: Training and test set

The third step of our data preprocessing included splitting the data into training sets and test sets. Traditionally, this usually involves a 75-25 percent data split performed on the entire data sample, where 75% of the dataset is allocated to training and the remaining 25% to testing. However, when dealing with time-series data, the sliding window method is preferred, which corresponds to dividing the data into training and test sets for each period. We will further elaborate on this method in section 4.2.

Step 4: Feature scaling

After splitting the data into training and test sets, we performed *feature scaling*. The different variables vary largely in range and scale, which can be complicated since most classifiers calculate the difference between two variables by the distance. In addition, some features have broad ranges of values, which can be challenging because the distance governs those particular variables. Hence, by normalizing the range of all features, they will each contribute proportionately to the final distance.

The test set should be subject to new, unseen data, meaning that it should not be accessible at the training stage. We therefore transformed the data using a *StandardScaler*[3] after the sliding window split to avoid any bias during the evaluation of the models. For the same reason, scaling was performed on the training data, and then the testing data were normalized according to the training set. The *StandardScaler* function standardizes the variables such that the distribution is centered around zero (0) with a standard deviation of one (1) (Keen, 2017). Each feature is scaled based on the following formula:

$$z = \frac{(z - \sigma)}{\sigma^2} \qquad \textit{Equation 2}$$

[3] A function from the Scikit-learn library

where σ refers to the mean of the training samples and σ^2 is subject to the standard deviation of the training samples.

Step 5: Feature selection

Our final step of the data preprocessing involved reducing the number of input variables. *Feature selection* is a process that has proven to be effective and efficient in data preprocessing. It leads to simpler and more understandable models, data mining performance increases, and it improves the model's performance (Li et al., 2017). We used both the *Support Vector Machine Recursive Feature Elimination (SVM-RFE)* approach and the *Random Forest Feature Importance* method to perform our feature selection. Further details are provided in section 4.4.

4.2 Prediction with Machine Learning

There are mainly two elements involved in Machine Learning: a learning process to determine the most accurate fit for the independent variable and an algorithm that (based on the learning) models the relationship among independent and dependent variables (Jung et al., 2018). We will further elaborate on these two components in the subsequent section.

4.2.1 Splitting into training and test set

A clear distinction between the data used for training and data used to test the model's predictability is essential to ensure a trustworthy outcome. The training set is defined as the part of the original data that provides the baseline for further application and utilization. Hence, the model produces a result based on the features within the training set and compares it to the target variable. Depending on the comparison, the parameters of the model are subsequently adjusted. The test set corresponds to the holdout part of the original data, which predictions are evaluated against.

Without a clear distinction between the training set and test set, several issues may arise. First, the *bias-variance trade-off* needs to be considered when allocating data to training and test sets. In literature, this is referred to as a trade-off between a model's ability to minimize bias and variance. On the one hand, we have the concept of overfitting (high variance), which implies a model with very

complex hypotheses and a large number of features. Consequently, the model produces great prediction accuracy during training but makes frequent errors when used on data not seen before (Burkov, 2019).
On the other hand, underfitting (high bias) may be a potential issue when the model uses simple assumptions and few features, causing inflexible learning from the dataset (Briscoe & Feldman, 2011). Implementing complexity control can be helpful in the trade-off between minimizing bias and variance in the training and test data. That is, selecting subsets of the variables to be used.
Second, dissimilar characteristics between the training and test set may encounter another problem. Modeling patterns can be discovered in the training set that is not present in the test set, making even highly complex models unsuccessful in producing reliable predictions.

4.2.2 Sliding window

The *sliding window* approach is used extensively when working with time series data and stock price trends. In this approach, observations in time T-1, T, and T+1 are closely related to each other. Thus, it better reflects real-life scenarios where new information becomes available when moving to the next period. By using this method, we are continuously updating the information available when making predictions.
A sliding window approach involves splitting the data into a training set and test set containing *n* years of data, where the size of all training and test sets is kept constant over time. Thus, the first prediction period will only be based on the initial window, while the oldest observation will be excluded, and the newest will be added to the training set when moving to the next period. We will further elaborate on this subject in section 5.1.

4.3 Machine Learning

Unlike a standard linear regression, Machine Learning algorithms enable computers to discover patterns in cases where the task is not evident (Alpaydin, 2014). *Supervised learning* is used when the goal is task-driven, meaning that we already know the desired output (Tatsat et al., 2020). It can further be split into either classification or regression models. A classification model tries to predict a categorical output based on the training data, whereas a regression model predicts continuous outcomes (Tatsat et al., 2020). For stock prediction, a regression model attempts to estimate the stock price, whereas a classification model tries to predict whether the price will increase or decrease for a given period.

4.3.1 Classification models

In our thesis, classification methods involving Support Vector Machine (SVM) and Random Forest (RF) will be applied to the sliding window technique. The decision on which classification model to use was determined by several factors. First, the model's simplicity needs to be considered. A simpler model often has a shorter training time, is easier to understand, and is also more scalable. For stock selection, the model's ability to handle non-linearity between the different variables is especially important. Moreover, it is crucial to examine how the models handle larger datasets and a significant number of features without causing overfitting. Lastly, the interpretability of the model is of significance. Considering all these factors, we found SVM and RF to be the most suitable algorithms.

4.3.2 Support Vector Machine

The basic idea behind the SVM classifier, which is a generalization of a *maximum-margin classifier*, is to construct a line or a hyperplane of p-1 dimensions separating the observations into classes. A hyperplane divides the p-dimensional space into two halves and can be defined as:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \qquad \textit{Equation 3}$$

where the β_i*'s* are the coefficients, and X_i's are the points on the hyperplane. The maximum-margin classifier works by finding the hyperplane with the largest margin separating the observations in the training data (James et al., 2013). An example of a separating hyperplane is shown in figure 4.2.

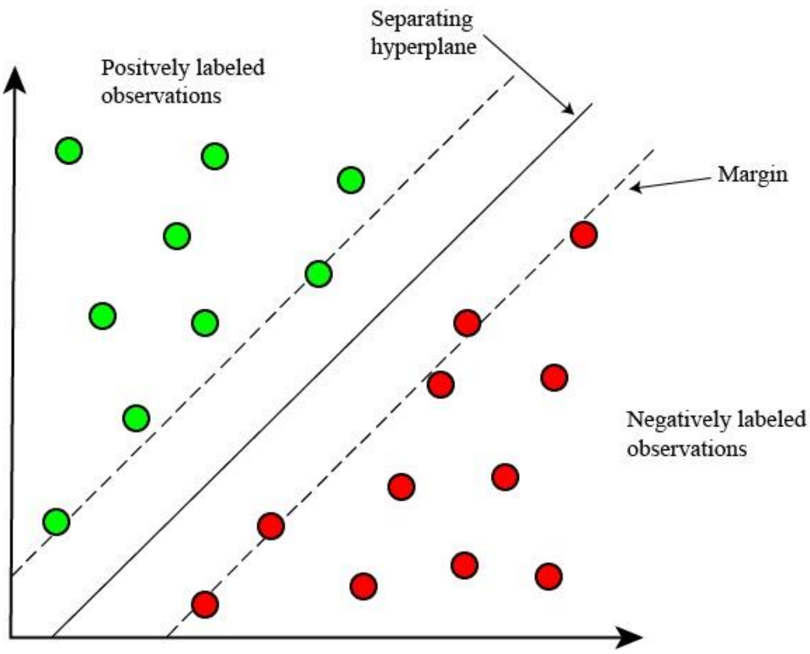


Figure 4.2: Example of the separating hyperplane in the Support Vector Machine.

However, a separating hyperplane does not always exist, causing the maximum-margin classifier to fail occasionally. To solve this problem, the Support Vector Classifier (SVC) introduces a soft margin that works similarly to the maximum-margin classifier. However, the SVC allows observations to be on the wrong side of the hyperplane as long as the majority of the observations are on the right side (James et al., 2013). It can thus be thought of as an optimization problem which is defined as:

$$maximize_{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n} M, subject\ to \sum_{j=1}^{p} \beta_j^2 = 1, \qquad \textit{Equation 4}$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M(1-\epsilon_i), \epsilon \geq 0, \sum_{i=1}^{n} \epsilon \leq C$$

where *M* is the width of the margin and *1,…,n* are slack variables informing about the location of the observation relative to the hyperplane. The parameter *C* refers to a budget of how much the margin can be violated. If *C = 0*, the model does not tolerate any margin violation, and the optimization problem will thus be equal to the maximal-margin classifier. Furthermore, as *C* increases, the model tolerates more violations to the margin (James et al., 2013). The solution to the optimization problem requires calculating the inner product of the observations. For two observations $x_i, x_{i'}$, the inner product can be computed as:

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j} \qquad \textit{Equation 5}$$

where *p* is the number of features (James et al., 2013). The SVM is an extension of the SVC and utilizes a kernel trick. This technique works by transforming classes that are not linearly separable into a higher dimensional feature space where linear separability is obtained. Several possible kernels can be applied with the SVM, where one example is the *radial kernel*, defined as:

$$K(x_i, x_{i'}) = e^{-\gamma (\sum_{j=i}^{p} \left(x_{ij} - x_{i'j}\right)^2} \qquad \textit{Equation 6}$$

The radial kernel uses the *Euclidean distance*[4] between the test observation and the training observation. From the Support Vector Machine classifier:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i) \qquad \textit{Equation 7}$$

we note that if the kernel ($K(x_i, x_{i'})$) is small, it will have little to no influence on the nonlinear function. More explicitly, training observations with a large Euclidean distance from the test observations will not influence the predictions. Thus, the radial kernel has a local behavior, where only the training observations close to the test observations affect the predictions (James et al., 2013).

4.3.3 Random Forest

A Random Forest (RF) is similar to a meta-algorithm that creates a large number of decision trees where each split in the tree considers a random sample. The random sample often includes only the square root of the total number of features in the feature set, implying that most of the variables are not even considered when splitting the tree. The reason for not considering all the features is to ensure that strong predictors are not used in the top split for all the trees and that the correlation between the trees created is low (James et al., 2013).

[4] The Euclidean distance between two points in Euclidean space is equal to the length of a line segment between the two points

The RF method utilizes *bootstrapping* and *bagging*. Bootstrapping is a technique to randomize the training data and create many sub-samples by randomly selecting observations from the training data. Further, it makes a prediction for each of the sub-samples allowing some of the observations to be repeated in several groups. After the sub-sample predictions are made, the technique called bagging is applied. The idea behind bagging is to average all of the predictions from the sub-samples (Suthaharan, 2016). Since the model utilizes several trees when making predictions, it tends to be more flexible and have less variance, reducing the probability of overfitting (Tatsat et al., 2020).

The process of the RF algorithm works by creating multiple sub-samples from the original dataset where the dimension *r* (number of features) of the sub-sample is $r \leq \sqrt{p}$, and *p* denotes the total number of features in the original dataset (Suthaharan, 2016). The sub-sample is then randomly altered using bootstrapping. Further, the decision-tree model is applied by randomly selecting *m* variables from the complete set of *p* variables and picking the best split-point among the *m* variables (Hastie et al., 2009). In our thesis, we are using the *Gini index* to evaluate the quality of the split-point. The Gini index is defined as:

$$G = \sum_{k=1}^{K} \hat{p}_{mk} (1 - \hat{p}_{mk}) \qquad \textit{Equation 8}$$

and measures the impurity of the node. A small *G* implies that a node contains mainly observations from one class (James et al., 2013). The node is then further split into two daughter nodes, and this process is repeated recursively for each terminal node until the minimum node size n_{min} is achieved (Hastie et al., 2009). A simple example of a Random Forest tree with a depth of three is shown in figure 4.3 below.

Because the Random Forest algorithm utilizes what is known as *recursive binary splitting*, it is often thought of as a top-down, greedy approach. Recursive binary splitting refers to the process of starting from the top of the decision tree and recursively searching for the best split at each step (James et al., 2013).

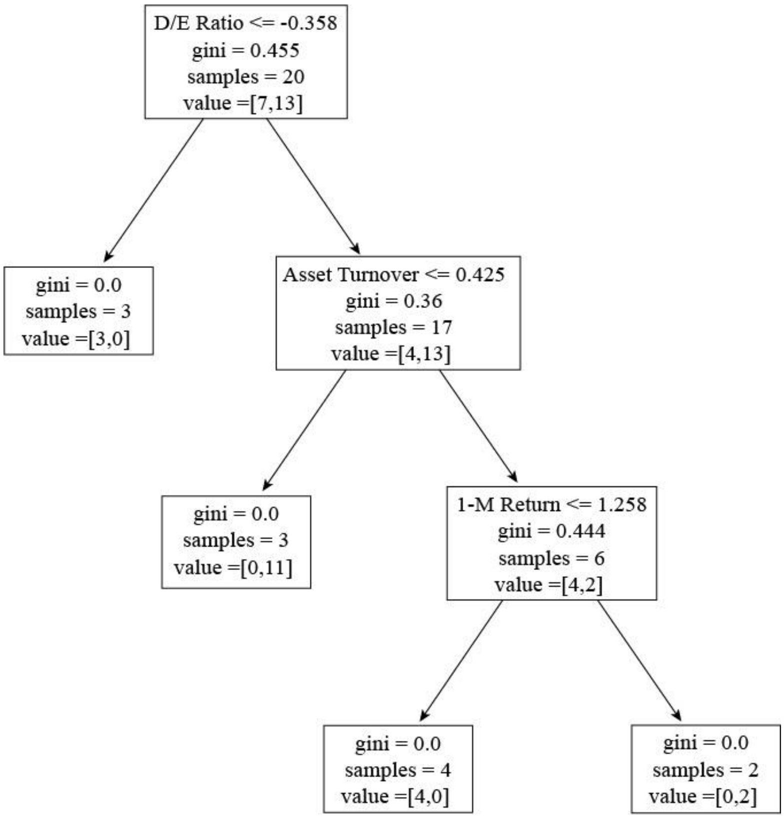


Figure 4.3: Example of a tree in a Random Forest model with a depth of three.

4.5 Feature selection

After the feature engineering, it is desirable to decrease the number of features in the model to only include informative variables. This is especially important if the variables added to the model are not tested for statistical significance. Removing non-informative features from the data set is called feature selection and is defined as the process of creating a subset of features from the original feature set. By removing redundant features, we may increase the model's accuracy and reduce learning time (Cai et al., 2018). Feature selection is also an important measure to deal with the dimensionality of the data. Essentially, we would want the number of observations to be as large as possible in order to reduce the effects of noise and outliers (Koutroumbas & Theodoridis, 2008). However, this is not always possible, making feature selection a vital tool to deal with high dimensionality.

There exist three classes of feature selection. Intrinsic methods refer to situations where the feature selection is performed within the model itself, whereas filter and wrapper methods use an external algorithm to select the features.

4.5.1 Support Vector Machine Recursive Feature Elimination

The Support Vector Machine Recursive Feature Elimination (SVM-RFE) is a wrapper method that works by backward selection. It first runs the SVM model with the entire feature set and then ranks the most important predictors by a measure of importance. After the first selection process, the SVM-RFE removes the least important variable and re-runs the SVM model with the smaller features set. This process is performed until the desired number of features is reached. The advantage of wrapper methods is their potential to search a wide range of feature subsets, providing a higher chance of finding the best subset of features. However, there are limitations related to this approach; the SVM-RFE does not consider different subsets, and it overlooks any potential meaningful interactions between features that are only significant in the presence of other features (Kuhn & Johnson, 2019).

4.5.2 Random Forest Feature Importance

A Random Forest Feature Importance is an intrinsic method of feature selection. In Scikit-learn, the RF model has a built-in function called *feature_importances_* , which presents the importance of each variable in the model. The function enables us to visualize the most influential features on the dependent variable. It utilizes *Gini importance*, which is a method that measures the impurity reduction introduced by each split in the tree. The features included in a split that leads to a significant decrease in impurity are considered important. For a given feature, the impurity importance is calculated as the sum of all impurity decrease measures where a split including the given feature is conducted. The total sum of decreases in impurity is then normalized by the number of trees in the forest (Nembrini et al., 2018).

5.0 EXPERIMENTAL DESIGN AND EVALUATION

In the subsequent sections, the experimental setup will be described, followed by a presentation of the selected performance measures and benchmark models used to evaluate our Machine Learning algorithms. The prediction results will be compared against two different benchmarks, whereas our constructed portfolios will additionally be measured against the indexes to better visualize potential excess returns. We will further explore the model's performance during periods of crisis. A *Variable Importance Plot* will also be created to visualize each feature's influence on the stock market prediction.

5.1 Experimental design

The initial step of our experiment was to avoid look-ahead bias. Thus, all the financial reports and *Quarterly Change in GDP* were accordingly lagged by one quarter. Look-ahead bias is a phenomenon that occurs when using data or information that is not available at the time of prediction because of differences in publication date. Most of the companies financial reports and information regarding GDP are published around two months after the end of the quarter they represent. Thus, when we are at, e.g., the end of the 1st quarter (Q1) 2020 and predicting for 2nd quarter (Q2) 2020, it is crucial to not use financial reports that represent Q1 2020 as they have not been published yet. Instead, financial reports from Q4 2019 are used to make predictions for Q2 2020. If we consider look-ahead bias in our predictions, the results can be unrealistically good and misleading as we are using information no other investor has available (Walimbe, 2017).

Data regarding stock prices and macroeconomic variables besides GDP were collected until the end of each quarter before we made our predictions. Since these are publicly available information accessible on the day they occur, we did not have to lag those variables. Figure 5.1 illustrates the time factor of our data used in predictions.

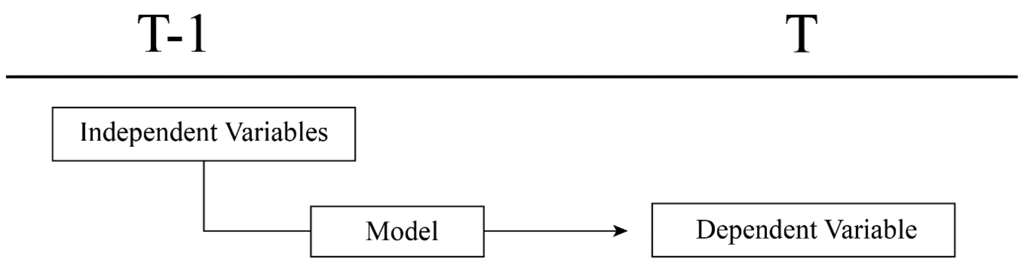


Figure 5.1: Illustration of how we use independent variables (Fundamental Factors and data regarding GDP) from time T-1 to predict the direction of the dependent variable in time T.

The consecutive step of our analysis involved selecting the number of features to include in the model before splitting the data into training and testing. From the original feature set, we had a total of 109 features. The number of features to include in a Machine Learning model is largely dependent on the problem. To find the reasonable number of variables, we tested several portions of the original data set and found the top 50% of the features to be the best performer. Hence, we ran the feature selection and picked the top 54 features from both the SVM-RFE and RF approaches.

Our third step of the experiment was to allocate data for training and testing through the sliding window approach and define our window size and prediction horizon. The initial training set consisted of quarterly data from Q2 2001 to Q2 2006, equivalent to 20 observations. Predictions were made for the subsequent quarter, meaning that the first testing period was set to Q3 2006 (see figure 5.2). In total, there were 10 111 training sets with corresponding test sets. To have the latest financial data available, we did not consider larger prediction horizons, as we might potentially miss important information. The window size was chosen based on a trade-off between having sufficient data for each prediction and the number of companies we could include. The size of the training window should be large enough to train complex models, yet not to the extent that we are unable to capture structural breaks and other changes in relations between variables. To illustrate, a window size of 20 observations requires a minimum of six years of financial reports for each company included in the model[5]. Thus, a larger window size would force us to exclude a great number of companies and further reduce

[5] To create our YoY variables, we need one additional year of data besides the window size.

our data set. Smaller training periods were also tested without improving the results compared to our selected window size.

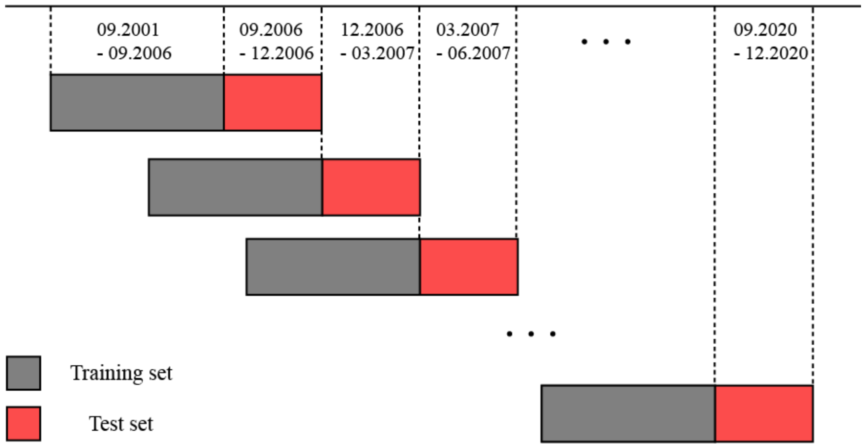


Figure 5.2: Quarterly sliding window, where each row represents a separate training- and test pair.

After the sliding window allocation and feature scaling, we calculated out-of-sample accuracy based on the comparison of predictions generated from the Support Vector Machine and the Random Forest model. The algorithms were run both with and without feature selection, resulting in a total of six models. Since Random Forest is a model based on randomization, we utilized a *random state parameter*[6] to ensure that our results are reproducible. The predictions obtained by the two models lay the basis for our portfolio construction.

5.2 Portfolio construction

To assess the performance of our models, we constructed several portfolios which differ in the maximum number of companies allowed in the portfolio. Each model is evaluated on its economic performance on the trading module, which refers to our simulated backtesting stock trading environment for a period of 14.5 years (Q3 2006 – Q1 2021). The time interval is divided into 58 trading sessions, and the portfolio is rebalanced at the end of every quarter.

At the beginning of each trading session, we use the model's predictions from the input data to allocate available money to the different assets. The predictions are associated with each stock's desired positions and can be either *buy* or *sell*. When

---

[6] A parameter in the Random Forest classifier from Scikit-learn which ensures that the random numbers are generated in the same order.

a *buy* recommendation is given, the model predicts that particular stock to outperform the next quarter's median return. On the contrary, a *sell* recommendation means that the model predicts a return below the next quarter's median return. For each prediction, the model presents a probability score illustrating its confidence level subject to the recommendation.
Further, companies that lack data during the backtesting period will not receive a recommendation from the model and are unavailable for trading from that particular quarter and onwards. If a recommendation is not available, the concerned stocks are sold off prior to the trading session to release potential funds for other available investment opportunities.

As illustrated by figure 5.3, input data for each trading session is the model's buy/sell recommendations, probability score, our selected trading strategy, the previous holdings in the portfolio, and the available money. The last rebalancing of the portfolio is performed at Q4 2020, and Q1 2021 is only utilized for generating portfolio returns.

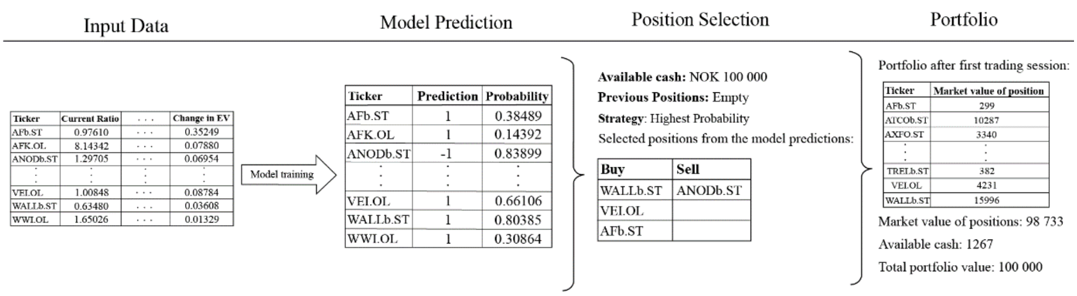


Figure 5.3: Example of the process for the first trading session.

5.3 Trading strategy

An initial cash holding of NOK 100 000 is similar to all the models, and the maximum limit of companies allowed in the portfolio range from 10 to 30 stocks. If a stock already in our portfolio is among the top *n* companies with the highest probability of being a *buy*, we increase our holding in that particular stock. To better compare the models, we also use the same weighted proportion of available cash to spend initially on each stock (more explanation in section 5.3.2). Further, any stocks in our portfolio receiving a *sell* recommendation from the model is always sold. This procedure is performed at the start of each trading session to update available money before buying stocks. If available money is not enough to open a new position on a specific stock, that recommendation is dropped. Throughout the whole backtesting period, we assume that all stocks included in the portfolio can be traded at their respective *Open Price* for the last available trading day of that particular quarter.

5.3.1 Highest probability

In our strategy, all stocks are ranked in a descending order based on their probability of beating the median quarterly return prior to each trading session. Depending on the maximum number of companies allowed in the portfolio, the top *n* stocks are selected to buy. If the models recommend less than *n buy* predictions, the remainder of the available money is saved for the next trading session. Additionally, if the maximum limit of companies allowed in the portfolio is reached, we only include a new stock if the *buy* probability is higher than one already in the portfolio. In this way, we always hold the *n* stocks with the highest probability of being a *buy*, and recommendations subject to a low probability are discarded.

5.3.2 Available money and weighted proportion of cash used

The portion of cash used on each stock (*p_cash*) dynamically changes with the probability the model provides on a given *buy* prediction. Thus, if a specific stock is subject to a high probability of being a *buy* in the following period, we increase our betting size. This procedure is chosen to utilize high conviction trades and avoid holding too much cash at later stages. Initially, we spend 20% of our available cash on each trade with a maximum of NOK 10 000 per trade. If the probability of a given *buy* prediction is higher than the 70% quantile in that trading period, we increase our *p_cash* to 50% with a maximum of NOK 50 000. Similarly, if the probability of a *buy* prediction is higher than the 90% quantile, *p_cash* is expanded to 80% with a maximum of NOK 100 000.

5.3.3 Transaction fees

The transaction fees included in this backtesting trading environment is similar to those provided by Nordnet, one of the most popular trading platforms in the Nordics. For each trade, the fees are 0.49%, with a minimum fee of NOK 79. Additionally, we only buy whole shares since Nordnet does not allow trading fractional shares. For simplicity and to illustrate the model's potential usage with today's trading costs, we have used the same equation throughout the backtesting period. The transaction fee for each trade is calculated as

$$max(79, 0.49\% * Order\ Size) \qquad \textit{Equation 9}$$

where the order size is the price of the stock multiplied by the number of shares.

5.4 Performance measures

To evaluate the performance of our Machine Learning algorithms, we have used several metrics derived from the *confusion matrix,* in addition to the *AUC score*. For the constructed portfolios, performance measures such as return, Sharpe ratio, standard deviation, max drawdown, and win rate are utilized to identify the best-performing model.

5.4.1 Classifiers

There are four possible outcomes for the confusion matrix (see figure 5.4). True positive (TP) refers to cases where the model predicts a positive value correctly, true negative (TN) is when the algorithm predicts a negative value that is actually negative, false positive (FP) is subject to predictions of a negative value that is actually positive, and false negative (FN) is the number of times the model incorrectly predicts the positive class as negative.

	Negative (Predicted)	Positive (Predicted)
Negative (Actual)	**TN**	**FP**
Positive (Actual)	**FN**	**TP**

Figure 5.4: Illustration of a Confusion Matrix.

In terms of evaluation metrics, we have included accuracy, misclassification rate, precision, recall, and F1-score. Accuracy refers to the overall accuracy, i.e., the correctly classified predictions, whereas misclassification rate is the percentage of predictions incorrectly classified. Precision is the number of times a prediction of a positive value was actually positive, whereas recall is defined as the fraction of all positive samples which were correctly predicted as positive by the model.

Lastly, F1-score is the harmonic mean between precision and recall. The formulas for each of the evaluation metrics are shown in table 5.1 below.

	Formula
Accuracy	$\frac{TN + TP}{TN + TP + FN + FP}$
Misclassification Rate	$\frac{FN + FP}{TN + TP + FN + FP}$
Precision	$\frac{TP}{TP + FN}$
Recall	$\frac{TP}{TP + FP}$
F1-Score	$2 * \frac{precision * recall}{precision + recall}$

Table 5.1: Evaluation metrics for our prediction results.

Classifiers are often measured based on their correct rate and accuracy. In stock classification, such performance measures are greatly affected by the chosen probability-threshold to separate the assets into the predetermined classes. Thus, to better visualize each model's accuracy, we also use the AUC score, which is the area under the ROC curve.

The ROC curve is defined as the *receiving operating characteristic* and is a common evaluation metric for classification algorithms. It is calculated from a combination of recall (true positive rate) and the false positive rate (FPR), whereas the recall is on the y-axis, and FPR constitutes the x-axis. The AUC score ranges from 0 to 1 and is calculated as the area under the drawn ROC curve. A score of 1 implies a perfect classifier, whereas a score above 0.5 suggests a model better than a random classifier. This evaluation metric can only be used on classifiers that provide some form of confidence or probability score, such as the SVM and RF (Burkov, 2019). However, this paper does not focus on achieving the highest possible accuracy, but rather on earning excess value from investing in the stock market based on the given predictions.

5.4.2 Portfolio

Since the goal is to earn an excess return above the OSEBX and OMXS30, the performance of our constructed portfolios is most important. We have evaluated each portfolio based on the following metrics:

	Formulas
Annualized Return	$r_p = \left(\left(\frac{End\ cash}{Initial\ cash}\right)^{\frac{1}{n}} - 1\right) * 100\%$
Standard Deviation	$SD = \sqrt{\frac{\sum \lvert x - \bar{x} \rvert^2}{n}}$
Sharpe Ratio	$SR = \frac{R_p - R_f}{\sigma_p}$
Win Rate	$WR = \frac{Profitable\ transactions}{Total\ transactions}$
Max Drawdown	$MDD = Max\left(\frac{(P_{peak} - P_{previous\ peak})}{P_{peak}}\right)$
Total Return	$TR = \frac{End\ cash}{Initial\ cash} - 1$

Table 5.2: Formulas for portfolio evaluation.

In table 5.2, the annualized return corresponds to the return each year over the period of our backtesting trading environment. The Sharpe ratio measures the return for each unit of risk of the portfolio and is a critical measure since we want to ensure that the potential excess return is not a result of excessive risk-taking. Furthermore, we evaluate the portfolio performance using win rate, which is defined as the portion of profitable transactions to the total number of transactions. To assess the risk characteristics, we utilize standard deviation and maximum drawdown. Maximum drawdown is the maximum decline after an observed peak in the portfolio. Thus, it measures the downside risk of the model over the specified period. The standard deviation informs us about the volatility of the portfolios. Lastly, the total return is measured in percentage for the whole trading period.

5.5 Benchmarks

In this paper, we also compare our constructed portfolios against two benchmarks – a *Dummy Classifier (DC)* and combinations of the OSEBX and OMXS30. Our models are compared against the DC in terms of classification accuracy and portfolio performance, while the benchmark indexes are only relevant for the portfolio evaluation.

5.5.1 Dummy Classifier

A Dummy Classifier is a benchmark model that utilizes simple rules rather than employing the training data to predict future directions. For the DC benchmark, we have used the function *DummyClassifier* from Scikit-learn and included both a *uniform* and *most frequent* strategy. The *uniform* strategy is similar to a Random Walk and involves making a random guess (uniformly) for the dependent variable in each prediction. In the financial world, this is often illustrated as a blindfolded monkey throwing darts at the financial pages in the newspaper (Malkiel, 2003). For the *most frequent* strategy, the predicted value is equal to the most common among the labels. Hence, the most frequent value within the training set will be selected for the corresponding test set.

In terms of portfolio construction for the two Dummy Classifier benchmark models, the initial available money is NOK 100 000, similar to our trading strategy. However, *p_cash* is always 20% with a maximum of NOK 20 000 since we cannot distinguish between the probabilities of each stock. Additionally, a new stock is only included in the portfolios if the maximum number of companies is not reached, and all *sell* predictions are performed at the start of each trading session. For the *uniform* strategy, we ran 1000 simulations and took the average of all outcomes to cope with potential outliers.

5.5.2 OSEBX and OMXS30

Since our portfolios are constructed with stocks listed on either Oslo Børs or Nasdaq Stockholm, we have benchmarked against both a weighted average between their respective indexes and a 50/50 split. The weighted average is calculated based on the number of companies in our portfolio coming from either Oslo Børs or Nasdaq Stockholm, and it is dynamically changing as we reconstruct our portfolio. Hence, this metric varies depending on the portfolio we benchmark against. It is calculated as:

$$N_{OB,t} * I_{OB,t} + N_{NS,t} * I_{NS,t} = WBI_t \qquad \textit{Equation 10}$$

where $N_{OB,t}$ is the portion of companies in our portfolio listed on Oslo Børs, with their corresponding OSEBX index $I_{OB,t}$, and $N_{NS,t}$ refers to the portion of companies listed on Nasdaq Stockholm with $I_{NS,t}$ as the OMXS30 values. This gives us the Weighted Benchmark Index, WBI_t, where all variables are dependent on time. The 50/50 split refers to a 50% weight for the two indexes throughout the whole backtesting period.

5.6 Evaluating performance during financial recessions

Historically, there have been several financial recessions affecting the stock market, such as Black Monday (1987), the dot-com bubble of 2001-2002, and the Financial Crisis in 2008-2009. In statistics, this is usually referred to as a *structural break*, meaning a time series that abruptly changes at a point in time (Stata, n.d.). Structural breaks can potentially lead to high forecasting errors and general unreliability of the model, especially if the break materializes in the middle of a time series sample. A usual method of managing this issue is to split the dataset into several pieces and only consider the periods before and after the break. However, the stock market is particularly vulnerable to structural breaks. To illustrate, the US stock market has experienced eight bear markets[7] from 1926-2019 (*How Long Do Downturns Last?*, 2020). Thus, how investors manage their portfolios during downturns greatly influences their overall performance. For our thesis, we will examine the performance of our models during both the Financial Crisis of 2008 and the still ongoing COVID-19 Pandemic.

5.7 Variable importance

The same method used for our Random Forest feature selection process, *feature_importances_*, is also selected to visualize the degree of importance our most influential features carry on the dependent variable.

The SVM-RFE feature selection method does not assign individual importance scores in the same matter as the RF method but rather ranks the variables according to their relevance. Within each rank, one cannot distinguish between the degree of feature importance. Hence, we refrain from displaying a variable importance plot for this method. A complete overview of the 54 features selected from both methods is given in table 10.2 in the appendix.

[7] Corresponding to a drop of more than 20 percent of major indexes.

6.0 RESULTS

In the next part, we present the empirical results of our Machine Learning models and portfolios against the selected benchmarks. In table 6.1, the evaluation metrics are summarized for the respective models and benchmarks. Tables 6.2 and 6.3 presents the results and performance of our constructed portfolios and the benchmark indexes, respectively. Lastly, we present a variable importance plot and analyze the most influential features from our best-performing model.
The results may differ depending on the performance metrics. Since accuracy in classification models is usually determined by the selected threshold, we mainly focus on the AUC score in our analysis. Further, because portfolio performance has a higher importance in our paper, it is given more attention.

6.1 Prediction results

In our analysis, both the Support Vector Machine and Random Forest outperformed the Dummy Classifier benchmark models in an overall assessment. However, in terms of accuracy, precision, and misclassification rate, the benchmark model with the *most frequent* strategy was similar to our Machine Learning algorithms.

	SVM (None)	SVM (SVM-RFE)	SVM (RF)	RF (None)	RF (SVM-RFE)	RF (RF)
Accuracy	53.605%	**54.050%**	53.921%	52.656%	52.863%	53.150%
Misclassification Rate	46.395%	**45.950%**	46.079%	47.344%	47.137%	46.850%
Precision	54.121%	**54.535%**	54.466%	53.319%	53.494%	53.835%
Recall	55.020%	**55.664%**	54.902%	52.246%	52.930%	52.500%
F1-Score	54.567%	**55.094%**	54.683%	52.777%	53.210%	53.159%
AUC score	53.518%	52.759%	53.996%	53.718%	**54.181%**	54.004%

	Dummy Classifier (Most Frequent)	Dummy Classifier (Uniform)
Accuracy	53.506%	49.968%
Misclassification Rate	46.494%	50.032%
Precision	54.591%	50.606%
Recall	48.652%	49.958%
F1-Score	51.451%	50.280%

Table 6.1: Performance of our prediction models and benchmarks.

Table 6.1 illustrates that all our models achieved above 50% accuracy and outperformed the Dummy Classifier (DC) benchmark model utilizing a *uniform* strategy, indicating that the predictive performance is better than a random guess. However, the DC with the *most frequent* strategy showed better results than our RF models regarding the accuracy, misclassification rate, and precision.

With SVM-RFE feature selection, the SVM model delivered the highest accuracy, precision, recall, and F1-score among our Machine Learning algorithms. In terms of AUC score, the RF model with SVM-RFE feature selection proved to be the best performer. A higher AUC score indicates that the model has a greater chance of separating the predetermined class labels as the given probability increases. That is, if a model provides a high probability for a prediction, it has a greater chance of correctly separating the classes when the AUC score is higher. The logic behind our trading strategy is to only include the stocks with the largest probability of being a buy. Thus, the implicit threshold for probability will be high, suggesting that a model with a greater AUC score will be better at selecting the correct stocks. The RF model with SVM-RFE feature selection is therefore considered to have the best potential for gaining excess profits in our trading strategy.

To further examine the robustness of our two best prediction models and to ensure that they are statistically better than a random guess, we performed a binomial test. Since our dependent variable is calculated based on the median quarterly return, it follows a uniform distribution. There is approximately a 50% probability for each outcome (buy/sell), and they are all independent of one another. Hence, we can use the *binomial distribution model* to statistically test our models, which is given by:

$$X \sim B(n = 10111, p = 0.5, q = 0.5) \qquad \textit{Equation 11}$$

where *n* is the number of trials, *p* is the probability of success for each outcome, and *q* is the probability of failure.
For the SVM model with SVM-RFE feature selection, there is a probability of 3.02e-17 that the accuracy is actually 50%. For the second model with the highest AUC score, RF with SVM-RFE feature selection, the same probability is 4.97e-10. Thus, in both instances, we can conclude that the models' accuracy is statistically better than a random guess.

6.2 Portfolio results

In the next section of our paper, we first evaluate our constructed portfolios against each other to identify the best performer. We further assess our selected reference model against the weighted average and the 50/50 split between OSEBX and OMXS30, in addition to similar portfolios based on predictions from the Dummy Classifier.

		SVM (None)	SVM (SVM-RFE)	SVM (RF)	RF (None)	RF (SVM-RFE)	RF (RF)
N = 30	Win Rate	**65.789%**	58.284%	58.644%	54.615%	53.518%	51.958%
	Max Drawdown	-57.281%	-58.487%	-58.061%	-56.268%	-55.499%	-68.637%
	Standard Deviation	12.094%	11.904%	**11.781%**	13.556%	12.165%	13.550%
	Sharpe Ratio	1.148	0.931	1.095	1.154	**1.606**	0.861
	Annualized Return	15.840%	13.045%	14.845%	17.663%	**21.555%**	13.685%
	Total Return	743.263%	491.721%	644.120%	957.458%	**1594.879%**	542.208%
N = 20	Win Rate	60.479%	58.289%	60.952%	56.944%	56.264%	55.281%
	Max Drawdown	-53.979%	-58.647%	**-52.391%**	-52.645%	-60.114%	-67.425%
	Standard Deviation	12.346%	12.725%	12.598%	12.763%	13.783%	12.629%
	Sharpe Ratio	1.428	1.094	1.361	1.260	1.038	0.676
	Annualized Return	19.579%	15.904%	19.106%	18.066%	16.305%	10.511%
	Total Return	1236.512%	749.985%	1161.967%	1011.260%	793.705%	325.945%
N = 10	Win Rate	58.407%	58.750%	65.137%	60.079%	61.111%	59.608%
	Max Drawdown	-55.620%	-56.661%	-60.159%	-53.099%	-55.341%	-60.521%
	Standard Deviation	12.381%	13.395%	13.912%	13.018%	12.553%	12.860%
	Sharpe Ratio	1.089	1.128	1.109	1.283	1.277	1.101
	Annualized Return	15.406%	17.095%	17.380%	18.668%	18.007%	16.135%
	Total Return	698.610%	885.803%	921.202%	1096.241%	1003.201%	774.881%

1) *The risk-free rate is calculated as the average weighted Norwegian and Swedish 10-year government bond yields, where the weights depend on the origin of the companies included in the models.*
2) *Where N is the maximum number of stocks in the portfolio*

Table 6.2: Portfolio performance by prediction model and portfolio size.

From table 6.2, we observe that the RF model with SVM-RFE feature selection and portfolio size of 30 (N = 30) has the highest annualized return with 21.555%. It has a maximum drawdown of -55.499% and the highest overall Sharpe ratio of 1.606. Further examination of the risk characteristics reveals that the SVM model with RF feature selection has the lowest standard deviation (11.781%). In terms of win rate, the SVM model without feature selection significantly outperforms the other models.

While we find the highest annualized return among the models with a portfolio size of 30, it is on average higher when reducing the number of companies. This is in line with our trading strategy since the implicit probability-threshold for including stocks in our portfolio will be higher with a lower N. However, the enhanced return comes at the expense of higher risk as the average standard deviation increases. With fewer companies in the portfolio, we get a lower diversification, and losing trades have a higher impact on the overall result. The same logic does not apply for the maximum drawdown, as one would expect. On the contrary, we observe that the maximum drawdown on average decreases with a reduced portfolio size.

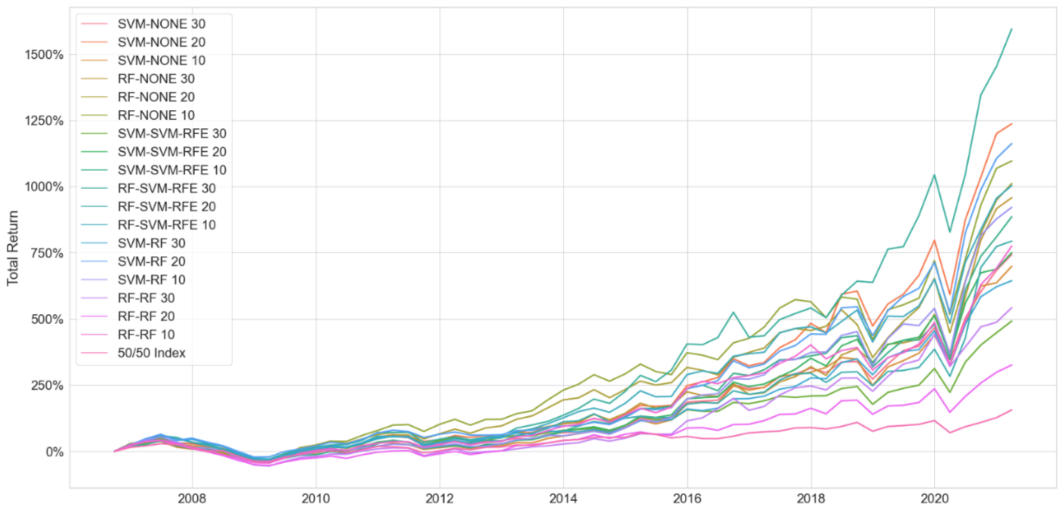


Figure 6.1: Comparison of the different portfolios and the 50/50 benchmark index.

As illustrated by figure 6.1, all of our models outperformed the 50/50 benchmark index for the backtesting period. Additionally, we can observe that all models and the benchmark index follow the same trends and are highly correlated. Since they are trading with the same selection of stocks, this is expected. From 2012-2018 the RF model with no feature selection and 30 companies outperformed the others, whereas from 2018 until the end, the RF model with SVM-RFE feature selection overtook as the best performer. An interesting finding is that the model showing the best predictive performance, but the lowest AUC score, was not among the top performers in our trading strategy. On the contrary, the RF model with SVM-RFE feature selection and portfolio size 30 is the best model in terms of annualized and total returns. This is in line with our theory suggesting that the highest AUC score has the best potential in our trading strategy. Consequently, we use the RF (SVM-RFE) model as the representative of Machine Learning in the following analyses. The portfolio size is also set to 30 for the Dummy Classifier benchmarks to have a better comparison.

6.2.1 RF model with SVM-RFE feature selection

Table 6.3 compares our reference Machine Learning model against two benchmark indexes and two portfolios constructed from Dummy Classifier predictions. The weighted average benchmark index is calculated based on the companies included in our RF (SVM-RFE) portfolio. To calculate the Sharpe ratio in table 6.3, 6.4, and figure 6.3, we have used the average between the Norwegian and Swedish 10-year government bond yields for each quarter.

	RF (SVM-RFE)	DC (Most frequent)	DC (Uniform)	Index (Weighted)	Index (50/50)
Annualized Return	21.555%	14.715%	13.425%	6.380%	6.701%
Standard Deviation	12.165%	12.251%	11.081%	9.150%	9.484%
Max Drawdown	-55.499%	-59.229%	-53.352%	-53.884%	-51.623%
Sharpe Ratio	1.591	1.021	1.013	0.457	0.474

Table 6.3: Summary statistics for our best-performing model and the respective benchmarks in the backtesting trading environment.

We observe that our RF model delivered approximately 6.84 percentage points higher annualized return than the best-performing benchmark. Despite the excess return, it does not seem to come at the expense of higher risk, as the standard deviation is slightly lower. Additionally, the Sharpe ratio is significantly higher compared to the same benchmark.

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
RF (SVM-RFE)														
Return	6.51%	-44.44%	55.24%	48.07%	-15.03%	8.53%	49.39%	38.00%	55.08%	5.08%	20.71%	15.10%	55.07%	35.74%
Standard Deviation	12.27%	5.89%	14.44%	8.56%	13.22%	6.60%	5.81%	9.76%	13.20%	13.73%	4.51%	8.71%	7.24%	20.66%
Max Drawdown	-15.66%	-44.44%	0.00%	0.00%	-22.10%	-7.36%	0.00%	-5.56%	-6.10%	-15.05%	0.00%	-5.64%	0.00%	-18.86%
Sharpe Ratio	0.130	-8.315	3.611	5.296	-1.321	1.148	8.291	3.754	4.151	0.355	4.530	1.700	7.603	1.730
DC (Most_frequent)														
Return	-1.87%	-50.85%	68.57%	38.69%	-18.19%	12.15%	17.06%	12.79%	44.26%	6.40%	22.82%	4.12%	61.38%	33.80%
Standard Deviation	10.36%	8.67%	12.22%	11.26%	9.56%	11.73%	7.23%	5.88%	12.79%	9.72%	4.18%	9.59%	8.52%	23.00%
Max Drawdown	-17.05%	-50.85%	0.00%	-6.76%	-20.83%	-12.64%	-4.46%	-4.37%	-1.82%	-4.32%	2.70%	-9.86%	0.00%	-22.10%
Sharpe Ratio	-0.655	-6.383	5.357	3.194	-2.158	0.954	2.190	1.947	3.436	0.637	5.397	0.400	7.200	1.470
DC (Uniform)														
Return	-2.06%	-45.73%	69.06%	24.19%	-17.86%	17.30%	47.72%	10.96%	24.05%	22.22%	8.30%	-3.31%	31.61%	31.33%
Standard Deviation	8.62%	7.89%	8.89%	10.08%	10.01%	10.32%	2.41%	4.68%	7.70%	6.39%	2.13%	10.53%	6.22%	25.00%
Max Drawdown	-14.04%	-45.73%	0.00%	-9.14%	-23.30%	-6.47%	0.00%	-3.87%	-1.62%	-0.76%	-0.72%	-15.48%	0.00%	-27.13%
Sharpe Ratio	-0.808	-6.364	7.423	2.128	-2.027	1.584	19.253	2.052	3.085	3.446	3.759	-0.342	5.078	1.253
Index (Weighted)														
Return	-2.50%	-45.74%	52.23%	19.48%	-14.67%	13.99%	21.17%	9.77%	0.28%	5.96%	8.05%	-11.15%	25.51%	5.84%
Standard Deviation	7.48%	6.55%	11.91%	6.85%	9.67%	6.89%	4.90%	1.44%	9.72%	6.76%	2.18%	10.79%	4.45%	15.29%
Max Drawdown	-11.58%	-41.84%	0.00%	-4.99%	-20.26%	-5.63%	-2.35%	0.00%	-14.03%	-8.15%	-0.37%	-16.68%	0.00%	-20.89%
Sharpe Ratio	-0.991	-7.674	4.126	2.446	-1.770	1.893	4.071	5.836	-0.002	0.852	3.575	-1.060	5.722	0.382
Index (50/50)														
Return	2.22%	-45.94%	55.75%	19.84%	-14.05%	14.30%	21.87%	7.96%	2.32%	8.39%	11.86%	-7.31%	22.93%	5.08%
Standard Deviation	6.37%	10.87%	11.26%	9.17%	10.34%	7.10%	4.51%	2.86%	8.33%	6.11%	2.45%	10.75%	4.05%	15.72%
Max Drawdown	-9.06%	-45.94%	0.00%	-7.53%	-20.59%	-5.24%	-1.60%	-0.20%	-11.09%	-5.55%	0.88%	-15.98%	0.00%	-21.09%
Sharpe Ratio	-0.422	-4.642	4.679	1.865	-1.594	1.879	4.572	2.307	0.242	1.338	4.724	-0.707	5.646	0.323

Table 6.4: Performance of our reference Machine Learning models and the respective benchmarks.

Exploring this further, we note from table 6.4 that the first few years do not reveal any outperformance. However, in 2010 and between 2013-2015, our RF model delivered significantly higher returns than the comparable benchmarks, which laid the basis for its superior result for the overall period. Thus, in the absence of bull markets and when the indexes experienced modest returns, the model still produced satisfying results. Another interesting observation is that even though our model yielded the overall highest annualized return, it is not close to the largest return for a given year. This is held by the Dummy Classifier models, which gained around 69% return in 2009. On the other hand, when examining the risk characteristics, we observe that our reference model delivered five years with no drawdown, whereas the closest benchmarks had at best three years.

Random Forest (SVM-RF) statistics:	
Average position value	NOK 14 627.56
Number of unique stocks in the portfolio	173
Total number of trades	1244
Longest held position in a stock	SAGAa.ST for 45 quarters (30/09/09 - 30/09/20)
The largest position in a single stock	NOK 646 704 (SAGAa.ST)
Origin of the stocks in the portfolio	7 Norwegian and 20 Swedish stocks in the portfolio on average

1) *Further portfolio statistics are found in appendix A.*

Table 6.5: Summary statistics for our selected portfolio in the backtesting trading environment.

Further examination of our reference model (see table 6.5) reveals that it held 173 out of 245 unique stocks available over the 14.5-year period. The model performed on average 21.4 trades per quarter, where the maximum would have been 30 trades (selling 15 stocks and buying 15 stocks at every trading session). Moreover, the average position value equaled NOK 14 627.56, whereas the largest position was in SAGAa.ST, which at one point resembled 44.83% of the portfolio and a market value of NOK 646 704. SAGAa.ST is also subject to the longest-held position and was included in the portfolio for 11 years and one quarter. On average, the portfolio held 7 Norwegian stocks and 20 Swedish stocks.

RF (SVM-RFE) after transaction costs:	
Win Rate	50.909%
Maximum Drawdown	-60.159%
Standard Deviation	12.396%
Sharpe Ratio	1.2563
Annualized Return	17.563%
Total Return	944.518%
Total number of transactions	980

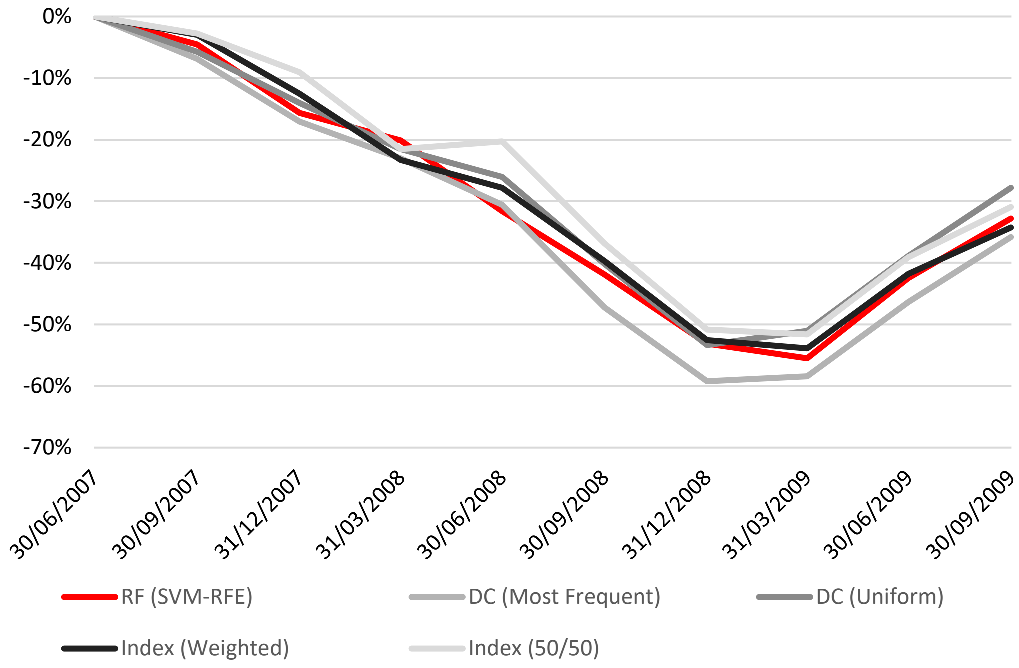
Table 6.6: Trading performance for our reference model after including transaction costs in the backtesting trading environment.

After including transaction costs in our backtesting trading environment, we observe from table 6.6 that our reference model still outperformed the other benchmarks. The fees do not significantly affect the results, as our quarterly rebalancing of the portfolio limits the number of transactions. Annualized return is reduced by approximately 4 percentage points compared to the backtesting environment before fees. Further, the reduced win rate indicates that we miss out on potential *buy* opportunities due to the added transaction cost, which is also illustrated by fewer trades. On average, the model performed 16.9 trades per quarter. Thus, transaction costs can be viewed as a constraint to our portfolio as the model is unable to take all of its preferred positions, compared to the scenario without fees.

6.3 Portfolio performance during crisis periods

The following section of our analysis investigates the portfolio return during periods of crisis, and evaluates whether the models can predict structural breaks more accurately than the benchmarks. We decided to center around both the Financial Crisis of 2008 and the COVID-19 Pandemic. The official date of the Financial Crisis is between mid 2007 and early 2009. Hence, we evaluated our portfolios from Q2 2007 to Q3 2009. For the COVID-19 Pandemic, we assessed the portfolio performance from Q4 2019 to Q1 2021. Since macroeconomic factors are leading predictors of bear markets, a natural hypothesis would be that the Machine Learning model delivers better than the benchmark indexes during structural breaks.

6.3.1 Financial Crisis of 2008



	RF (SVM-RFE)	DC (Most Frequent)	DC (Uniform)	Index (Weighted)	Index (50/50)
Annualized Return	-14.704%	-16.246%	-12.223%	-12.205%	-13.770%
Standard Deviation	16.114%	17.153%	14.967%	13.739%	14.985%
Maximum Drawdown	-55.499%	-58.409%	-51.033%	-49.517%	-51.623%

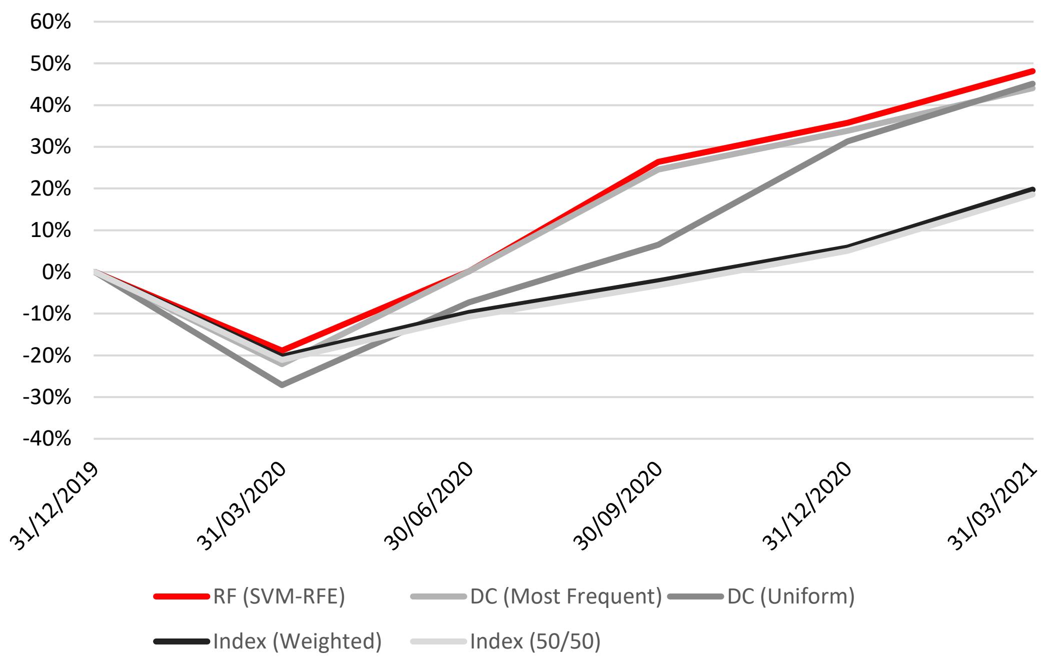
Figure 6.2: Trading performance in the period Q2 2007 to Q3 2009 (Financial Crisis) [8].

However, from figure 6.2, we see that all the models follow the same bear trend during the Financial Crisis, with only minor differences. Our reference Machine Learning model delivered 2.49 percentage points worse in annualized return than the top-performing benchmark. The volatility is also higher than all comparable benchmarks except the *most frequent* Dummy Classifier model. Furthermore, the maximum drawdown of 55.5% for the RF (SVM-RFE) model over the whole period occurs during the Financial Crisis. Compared to the benchmark models, the Financial Crisis is subject to the worst-performing period for our reference model when considering the entire backtesting session. Hence, indicating that our long-term view and quarterly data provide some limitations, as the model lacks the ability to capture short-term indicators prior to a structural break.

[8] We have not included Sharpe ratio for the Financial Crisis sample since a negative Sharpe ratio is unuseful.

6.3.2 COVID-19

For the still ongoing COVID-19 pandemic, figure 6.3 shows that our model and the benchmarks followed the same trend initially. We observe that our reference model had a maximum drawdown of 18.86% in the worst quarter, whereas the weighted average and 50/50 split benchmark indexes experienced a drawdown of 20.29% and 21.09%, respectively. Considering the entire period, our model delivered 21.37 and 22.34 percentage points higher annualized return compared to the same benchmark indexes. The Dummy Classifier (DC) benchmark models yielded significantly high returns in the same period, modulating the performance from our model. However, we observe a significantly lower Sharpe ratio compared to our reference model. Thus, the high return comes at the cost of increased risk.



	RF (SVM-RFE)	DC (Most Frequent)	DC (Uniform)	Index (Weighted)	Index (50/50)
Annualized Return	36.943%	33.910%	34.711%	15.578%	14.606%
Standard Deviation	16.199%	18.391%	19.366%	12.978%	12.938%
Maximum Drawdown	-18.856%	-22.100%	-27.129%	-20.294%	-21.092%
Sharpe Ratio	2.252	1.819	1.768	1.165	1.093

Figure 6.3: Trading performance in the period Q4 2019 to Q1 2021 (COVID-19 pandemic).

A more interesting observation was found when examining the periods after a financial shock. Our model yielded almost 48 percentage points higher return than the weighted benchmark index for the first two years following the Financial Crisis. The same trend is observed for the one-year period after March 2020, where the reference Machine Learning model gained approximately 32.36 percentage points higher return than the same benchmark. Hence, it might be that the highest potential for our Machine Learning model lies in the ability to identify the best *buy* possibilities following a financial recession or structural break.

6.4 Variable importance plot

In this section, we present the most influential variables from the Random Forest Feature Importance approach based on all test periods. As soon to be observed, variables concerning macro factors are considered most important according to this method. An illustration of the 20 most influential features is provided in descending order of importance for the Random Forest in figure 6.4.

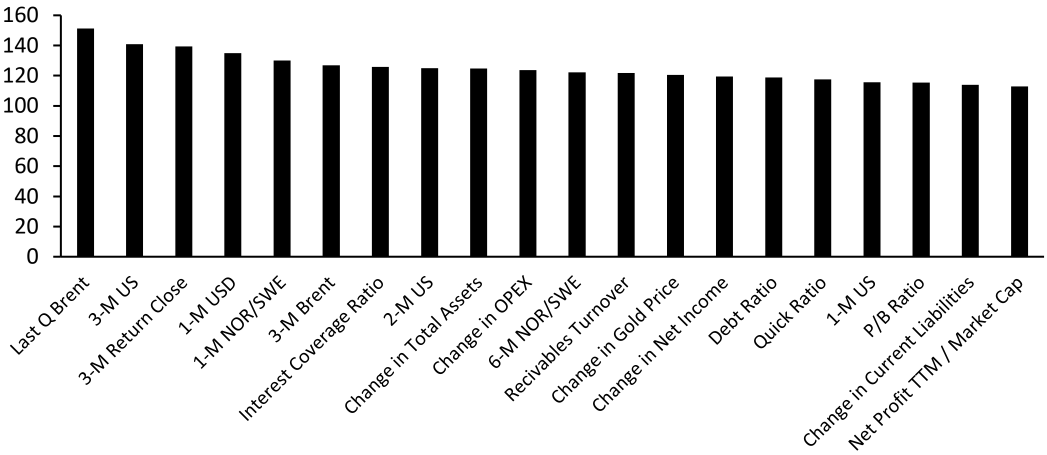


Figure 6.4: Feature importance for the Random Forest method.

From the variable importance plot, we observe that among the top 5 variables, only macroeconomic- and momentum factors are selected. *Last Q Brent* seems like a valid choice as the most influential feature, given the great impact the oil industry is considered to have on the Norwegian stock market. *3-M US* represents the development of the US 10-year Treasury Yield, by many referred to as the world's most important interest rate, whereas the *1-M USD* constitutes the development of the world's reserve currency. Hence, they are believed to have a significant influence on the global financial markets. The *3-M Return Close* is also a valid choice given that stocks frequently follows a momentum. Lastly, the

1-M NOR/SWE is essential as it represents the development of the respective countries' economic conditions.

From the original dataset, the three categories, Macroeconomic Factors, Momentum Factors, and Fundamental Factors, each holds 37%, 11%, and 52% of all variables, respectively. From table 10.2 in the appendix, we observe that macroeconomic variables stand for 50% of the features selected, whereas 39% are fundamental factors. Hence, implying that macroeconomic features are most important in determining the stocks' future directions and that we could have included a higher number of such factors.

7.0 DISCUSSION

During our work with the dissertation, we have encountered several challenges regarding the data and backtesting, which may have impacted our results. In this section, some of those challenges and weaknesses will be addressed. Further, we will elaborate on the robustness of our model and potential practical usage, as well as discuss our findings and their implications to the Efficient Market Hypothesis. Lastly, we will identify relevant areas of further research within the utilization of Machine Learning in stock selection.

7.1 Weaknesses with our data

For the final dataframe, many variables had to be registered with 0, especially for the early years 2000-2005. Adding zeros and filling variables with the latest available value may have affected the overall structure of our data and made it more linear than it otherwise would have been. Consequently, to avoid excessive data manipulation, we had to discard many years of company fundamentals. A number of the features we set out to use before retrieving the data were also useless due to low quality. Generally, more data are assumed to provide better quality since it allows for a stronger foundation to train the algorithms. Overall, this can partially explain why our constructed fundamental features were not as informative to the prediction as the macroeconomic- and momentum factors, which had higher data quality.

7.2 Challenges with backtesting

Backtesting is a challenging process, given that a strong performance in the trading simulation does not guarantee future returns. Further, by collecting companies based on stock listings as per 30/09-20 due to data challenges, we did not encounter potential bankruptcy and/or delisting of firms in our backtesting environment. Survivorship bias is therefore present in our analyses since the model will only trade stocks that are still present today. Thus, we have to be careful when evaluating our strategies and models. Additionally, by testing several approaches and trading strategies, you can most likely find one that yields good results. This makes it difficult to determine whether backtesting results are actually promising or a product of pure luck. The challenges related to backtesting are considered one of the main fundamental questions in quantitative finance. A foolproof solution to this problem would most likely provide investors with solid and risk-free returns (Lopez de Prado, 2018).

7.2.1 Challenges with transaction costs

For our backtesting trading environment, a simplistic measure of transaction costs has been used. First, we assume that the direct transaction costs are calculated based on the same factor throughout the whole period. However, this factor was higher in earlier time periods than today due to the new electronic platforms. Thus, a slight overestimation of returns in the early years is inevitable with this solution. Second, we do not account for the bid-ask spreads and liquidity issues, and rather assume that each stock can be bought at its respective *open price* for every trading session. Lastly, our data set does not provide the volume available at different bid-ask prices, making us unable to examine how the price impact related to portfolio size could potentially affect returns.

7.3 Robustness of our models

To cope with the challenges related to backtesting and provide more reliable results, we decided to test our designed trading strategy on several models and different company allocations. Interestingly, all our models outperformed the benchmark indexes in terms of annualized and risk-adjusted returns. Our constructed portfolios gained on average an excess annualized return above our 50/50 split and weighted average benchmark indexes with 9.90 and 10.22 percentage points, respectively. We also observe an average Sharpe ratio of 1.147

for our models. In comparison, the 50/50 split and weighted benchmarks had a Sharpe ratio of 0.494 and 0.477 for the same period. Additionally, our models produced significant prediction properties with an average AUC score of 53.696% and accuracy of 53.374%. This is considered quite respectable within the finance industry since, in theory, you would only need to get over half of your trades correct to make money. Thus, our findings both before and after transaction costs challenge the semi-strong form Efficient Market Hypothesis, which claims that our models should not be able to produce excess risk-adjusted returns. A possible explanation for our portfolios' outperformance may be due to the long-term perspective and quarterly rebalancing, which reduces the risk, transaction costs and minimizes the opportunities for losing trades. Further, we do not consider potential liquidity issues surrounding the stocks traded. Our assumption that all stocks can be traded at their respective open prices and not accounting for potential price impact may have affected the overall results. The survivorship bias also prevents the model from potentially investing in companies that go bankrupt.

However, given that all our constructed portfolios were able to significantly outperform the benchmark indexes, there is a high probability that it can yield excess returns also in the future. Hence, taking all the limitations into account, it still seems profitable to trade on the predictions presented by the model. More sophisticated strategies can also be implemented to potentially increase profitability. Alternatively, the complex nature of the finance industry and the lack of future guarantees from backtesting can facilitate another use of our model. Portfolio managers and other professionals often have a large group of stocks to analyze and include in their strategies. An alternative use case for our prediction model could therefore be as a stock screener to identify potential companies before implementing further analyzes and their trading strategies.

7.4 Fundamental analysis with Machine Learning

One of the fundamental ideas in finance is that all agents are rational. However, this idea has been challenged repeatedly, with the latest being the surge in behavioral finance which focuses on investors' behavior and its implications on the market. Returns are often affected by the irrationality of the investor, which can be overconfidence, limited attention, or other cognitive biases. In large, we find the challenge of removing the irrationality of the investors from the equation interesting. Thus, Machine Learning is a compelling subject, and while it is not entirely without bias, it is a significant step in the right direction.

7.5 Further research within the field

We have identified several areas where the research of Machine Learning in finance could expand. First, more complex algorithms and models, especially *Deep Learning* techniques, which have gained more attention in recent years, could be implemented. Ideally, applying more sophisticated models, e.g., ANN, would be preferable in terms of accuracy. However, such techniques usually require more computing power and a massive amount of high-quality data and are more time-consuming. Additionally, even the most straightforward neural net could be challenging to interpret since ANN is characterized as a “Black Box”-algorithm[9]. Further research within the field could also perform more thorough hyperparameter optimization to enhance the model’s predictability, which had to be left out for our experiment due to computer limitations.

Second, it is still essential to explore new features that may have higher predictability, such as volatility and technical factors. Additionally, the feature selection process needs to be optimized, i.e., by creating a more sophisticated procedure to determine the accurate number of features.

Third, the backtesting trading environment could be more realistically constructed in terms of including accurate transaction costs. For smaller stock exchanges (i.e., Oslo Børs and Nasdaq Stockholm), liquidity issues are also more severe. Hence, there is a need to continually improve how to account for such factors.

8.0 CONCLUSION

Throughout history, investors have been searching for better methods to predict future stock price directions. The most popular involve fundamental analysis, technical analysis, or a combination. However, with the increased availability of data and computational power, more sophisticated methods such as Machine Learning have been developed. The main objective of our dissertation was to

[9] Common definition of complex algorithms where it is challenging to interpret how the system reaches its conclusion.

examine whether Machine Learning algorithms can add value when constructing portfolios based on predictions from long-term stock price directions.
As presented in this study, the finance industry is highly complex, making it challenging to predict stock prices with high accuracy. Additionally, weaknesses in our dataset, such as low-quality input and lack of data, further complicated the analysis. After reviewing different models, we found the Support Vector Machine (SVM) and the Random Forest (RF) classifier to best fit our needs in terms of simplicity, as well as its ability to handle non-linear relationships and a large number of features. For each classifier, we ran three different configurations in terms of feature selection; “None”, where no selection was performed, and Support Vector Machine Recursive Feature Elimination (SVM-RFE) and Random Forest (RF) feature selection, where the top 50% variables were selected.

From our research, the Random Forest model with SVM-RFE feature selection was found to yield the highest AUC score, annualized- and risk-adjusted return. Considering our entire backtesting period, the model significantly outperformed both the benchmark indexes and the two Dummy Classifier models. Overall, it gained 6.84 percentage points higher annualized return than the best-performing benchmark model. Compared to the 50/50 split and weighted average benchmark indexes, our model yielded 14.85 and 15.17 percentage points in excess annualized return, respectively. However, when examining crisis periods, the performance was not equally impressive. For both structural breaks considered, our reference Machine Learning model did not outperform the benchmarks. On the other hand, further examination revealed that the highest outperformance from our model was found in periods following a financial recession.

We believe the results show the value Machine Learning can bring to investors in the stock market. It may not completely substitute existing models and human interaction due to the highly complex nature of the industry. Nevertheless, it can serve as a supporting and supplementary tool in making objective and rational decisions. The new digital era and increased data availability highlight the importance of continuously seeking growth opportunities and innovations using big data analysis in the finance industry to enhance competitive advantage.

9.0 BIBLIOGRAPHY

Alberg, J., & Lipton, Z. C. (2017). Improving factor-based quantitative investing by forecasting company fundamentals. *ArXiv Preprint ArXiv:1711.04837*.

Alpaydin, E. (2014). *Introduction to machine learning* (Third edition). The MIT Press.

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*(20), 7046–7056. https://doi.org/10.1016/j.eswa.2015.05.013

Bier, W., & Ahnert, H. (2001). Trade-off between timeliness and accuracy. *Document de Travail*, *4299*.

Bodie, Z., Kane, A., & Marcus, A. J. (2018). *Investments*. /z-wcorg/.

Braudel, F. (1982). *The wheels of commerce* (1. [print.]).

Briscoe, E., & Feldman, J. (2011). Conceptual complexity and the bias/variance tradeoff. *Cognition*, *118*(1), 2–16. https://doi.org/10.1016/j.cognition.2010.10.004

Burkov, A. (2019). *The Hundred-Page Machine Learning Book*.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Chaffai, M., & Medhioub, I. (2014). Behavioral Finance: An Empirical Study of the Tunisian Stock Market. *International Journal of Economics and Financial Issues*, *4*(3), 527–538. ABI/INFORM Collection; Publicly Available Content Database.

Chen, J. (2010). *Essentials of technical analysis for financial markets*. Wiley. http://rbdigital.oneclickdigital.com

Chen, S.-S. (2009). Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance*, *33*(2), 211–223. https://doi.org/10.1016/j.jbankfin.2008.07.013

Chen, Y., Liu, K., Xie, Y., & Hu, M. (2020). Financial Trading Strategy System Based on Machine Learning. *Mathematical Problems in Engineering*, *2020*, 3589198. https://doi.org/10.1155/2020/3589198

El Hedi Arouri, M., Lahiani, A., & Nguyen, D. K. (2015). World gold prices and stock returns in China: Insights for hedging and diversification strategies.

Economic Modelling, *44*, 273–282. https://doi.org/10.1016/j.econmod.2014.10.030

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, *25*(2), 383–417. JSTOR. https://doi.org/10.2307/2325486

Fama, E. F. (1995). Random walks in stock market prices. *Financial Analysts Journal*, *51*(1), 75. ABI/INFORM Collection.

Fama, E. F., & French, K. R. (2018). Choosing factors. *Journal of Financial Economics*, *128*(2), 234–252. https://doi.org/10.1016/j.jfineco.2018.02.012

Gopinathan, R. & S Raja Sethu Durai. (2019). Stock market and macroeconomic variables: New evidence from India. *Financial Innovation*, *5*(1), 1–17. ABI/INFORM Collection; Publicly Available Content Database. https://doi.org/10.1186/s40854-019-0145-1

Grossman, S. J., & Stiglitz, J. E. (1980). On the Impossibility of Informationally Efficient Markets. *The American Economic Review*, *70*(3), 393–408. JSTOR.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed). Springer.

How Long Do Downturns Last? (2020, August 3). http://blogs.harvard.edu/proreview/how-long-do-downturns-last/#:~:text=Generally%2C%20economic%20recessions%20don%27t,decline%20since%20World%20War%20II

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.).

James, G., Witten, D., Hastie, T., & Tibshirani, R. (Eds.). (2013). *An introduction to statistical learning: With applications in R*. Springer.

Jegadeesh, N., & Titman, S. (1993). Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. *The Journal of Finance*, *48*(1), 65–91. JSTOR. https://doi.org/10.2307/2328882

Jung, J.-K., Patnam, M., & Ter-Martirosyan, A. (2018). An Algorithmic Crystal Ball: Forecasts-based on Machine Learning. *IMF Working Papers*, *18*, 1. https://doi.org/10.5089/9781484380635.001

Keen, B. A. (2017, May 10). Feature Scaling with scikit-learn. *Benalexkeen.Com*. https://benalexkeen.com/feature-scaling-with-scikit-learn/

Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-01839-w

Koutroumbas, K., & Theodoridis, S. (2008). *Pattern Recognition*. Elsevier Science. https://books.google.no/books?id=QgD-3Tcj8DkC

Kuhn, M., & Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*.

Kumar, D., & Maheswaran, S. (2012). Testing the Martingale Hypothesis in the Indian stock market: Evidence from multiple variance ratio tests. *Decision*, *39*(2), 62–85. ABI/INFORM Collection.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, *50*(6), 1–45.

Lopez de Prado, M. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/bilibrary/detail.action?docID=5240570

Malkiel, B. G. (2003). *A random walk down Wall Street: The time-tested strategy for successful investing*. Completely revised and updated edition. New York : W.W. Norton, [2003] ©2003. https://search.library.wisc.edu/catalog/999956933302121

McCormick, L., & Regan, M. (2021, March 16). Why 10-Year Treasury Yields Get All the Attention. *Bloomberg*. https://www.bloomberg.com/news/articles/2021-03-16/why-10-year-treasury-yields-get-all-the-attention-quicktake

Misra, P. (2018). An investigation of the macroeconomic factors affecting the Indian stock market. *Australasian Accounting, Business and Finance Journal*, *12*(2), 71–86.

Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance? *Bioinformatics*, *34*(21), 3711–3718. https://doi.org/10.1093/bioinformatics/bty373

Nkemnole, E. (2016). A Hidden Markov Model inference approach to testing the Random Walk Hypothesis: Empirical evidence from the Nigerian Stock Market. *Journal of Economic and Financial Sciences*, *9*(3), 696–713.

ABI/INFORM Collection; Publicly Available Content Database. https://doi.org/10.4102/jef.v9i3.66

Norsk Petroleum. (2021, March 25). *EXPORTS OF OIL AND GAS*. EXPORTS OF OIL AND GAS. https://www.norskpetroleum.no/en/production-and-exports/exports-of-oil-and-gas/

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2019). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 1–51.

Ødegaard, B. A. (2009). Hva koster det å handle aksjer på Oslo Børs? *Praktisk Økonomi & Finans*, *25*(1), 93–99. idunn.no. https://doi.org/10.18261/ISSN1504-2871-2009-01-1

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

Piard, F. (2013). *Quantitative Investing: Strategies to exploit stock market anomalies for all investors*. Harriman House Limited.

Rasekhschaffe, K. C., & Jones, R. C. (2019). Machine Learning for Stock Selection. *Financial Analysts Journal*, *75*(3), 70–88. https://doi.org/10.1080/0015198X.2019.1596678

Ritter, J. R. (2003). Behavioral finance. *Selected Papers from the Fourteenth Annual APFA/PACAP/FMA Finance Conference*, *11*(4), 429–437. https://doi.org/10.1016/S0927-538X(03)00048-9

Roscoe, P., & Howorth, C. (2009). Identification through technical analysis: A study of charting and UK non-professional investors. *Accounting, Organizations and Society*, *34*(2), 206–221. https://doi.org/10.1016/j.aos.2008.05.003

Scholz, M., Nielsen, J. P., & Sperlich, S. (2015). Nonparametric prediction of stock returns based on yearly data: The long-term view. *Insurance: Mathematics and Economics*, *65*, 143–155. https://doi.org/10.1016/j.insmatheco.2015.09.011

Stata. (n.d.). *Tests for structural breaks in time-series data*. Retrieved May 10, 2021, from https://www.stata.com/features/overview/structural-breaks/

Suthaharan, S. (2016). *Machine learning models and algorithms for big data classification: Thinking with examples for effective learning*. Springer.

Tatsat, H., Puri, S., & Lookabaugh, B. (2020). *Machine learning and data science blueprints for finance: From building trading strategies to robo-advisors using Python*. http://proquest.safaribooksonline.com/?fpi=9781492073048

Walimbe, R. (2017, April 21). Avoiding Look Ahead Bias in Time Series Modelling. *Data Science Central*. https://www.datasciencecentral.com/profiles/blogs/avoiding-look-ahead-bias-in-time-series-modelling-1

World Bank. (n.d.). *Market capitalization of listed domestic companies (current US$)*.

X. Yuan, J. Yuan, T. Jiang, & Q. U. Ain. (2020). Integrated Long-Term Stock Selection Models Based on Feature Selection and Machine Learning Algorithms for China Stock Market. *IEEE Access*, *8*, 22672–22685. https://doi.org/10.1109/ACCESS.2020.2969293

List of Tables

List of Figures

10.0 APPENDIX

10.1 Appendix A: Portfolio results and analysis for our reference Machine Learning model

We have analyzed the top 10 traded stocks from our RF model with SVM-RFE feature selection to better understand the dynamics behind the stocks chosen by the model. Median market capitalization for the stocks included in our backtesting trading environment is NOK 5093 (millions) per 31.03.2021, whereas, for the top 10 most traded stocks, it is NOK 20 605 (millions). All of the top 10 most traded stocks are still listed as of 31.03.2021.

	Ticker	Number of trades	Market Capitalization (millions)	Annualized return[10]
Sagax AB	SAGAa.ST	54	58754	32.99%
AF Poyry AB	AFb.ST	42	27314	17.48%
Fastighets AB Balder	BALDb.ST	42	75174	27.06%
Atlas Copco AB	ATCOb.ST	38	497187	16.14%
Biotage AB	BIOT.ST	33	9066	20.47%
XANO Industri AB	XANOb.ST	33	2706	18.17%
Betsson AB	BETSb.ST	31	9029	25.08%
Vitrolife AB	VITR.ST	31	23433	34.20%
DNO ASA	DNO.OL	30	7846	0.50%
FastPartner AB	FPARa.ST	30	17774	15.83%

1) Market Capitalization as of 31.03.2021 (Refinitiv)

Table 10.1: Portfolio summary statistics.

[10] Annualized return as per our backtesting period (Q4 2006 – Q1 2021)

10.2 Appendix B: Feature selection

A complete overview of the 54 features selected by the SVM-RFE feature selection and the RF feature selection. The variable names are the actual shorter versions that we used in our programming for simplicity.

SVM - RFE	Random Forest
Quick Ratio	Current Ratio
Return on Assets	Quick Ratio
Debt Ratio	Return on Equity
Long-Term Debt Ratio	P/B Ratio
Asset Turnover	P/S Ratio
EBITDA-margin	Debt Ratio
Inventory turnover	Asset Turnover
Operating margin	Interest Coverage Ratio
ROA TTM	EV / EBITDA
YoY growth Current Liabilities	MCAP / FCF
1-M Return	Receivables Turnover
2-M Return	Revenue TTM / Market Cap
3-M Return Close	Net Profit TTM / Market Cap
3-M Return Open	Net Profit / Market Cap
12-M Return	PEG Q
NIBOR/STIBOR 3M	M/B
GNGT3M	YoY growth EBIT
Unemployment Rate	YoY growth Cash from Operation
GDP Q Change	YoY growth Current Liabilities
12-M US	YoY growth Total Equity
6-M US	1-M Return
3-M US	2-M Return
2-M US	3-M Return Close
1-M US	Open Price 1-M
US Bid	Open Price
12-M NOR/SWE	CPI YoY
6-M NOR/SWE	Brent
3-M NOR/SWE	Gold Price
2-M NOR/SWE	GDP Q Change
1-M NOR/SWE	6-M US
Bid NOR/SWE	3-M US
12-M Brent	2-M US
6-M Brent	1-M US
2-M Brent	US Bid
1-M Brent	6-M NOR/SWE

12-M USD	2-M NOR/SWE
6-M USD	1-M NOR/SWE
3-M USD	3-M Brent
2-M USD	2-M Brent
12-M EUR	Last Q Brent
6-M EUR	12-M USD
3-M EUR	1-M USD
2-M EUR	12-M EUR
1-M EUR	3-M EUR
Return Index	1-M EUR
Change in Index	Change in Revenue
Change in Current Liabilities	Change in Operating Expenses
Change in Current Assets	Change in Current Liabilities
Change in Total Assets	Change in Total Assets
Change in Total Liabilities	Change in Net Income
Change in EV	Change in EV
Change in Brent	Change in Profit Margin
Change in 3-Month Government Yield	Change in Gold Price
Change in 10-Year Government Yield	Change in 3-Month Government Yield

Table 10.2: Overview of all 54 features selected by the SVM-RFE and Random Forest method.

10.3 Appendix C: Feature computation

Illustration of how we calculate each input feature used in our models. A total of 109 features were used in our analyses.

Category	Feature Computation
Fundamental Factors	
Liquidity factors	$Current\ Ratio = \frac{Current\ Assets_t}{Current\ Liabilities_t}$
	$Quick\ Ratio = \frac{Current\ Assets_t - Inventory_t}{Current\ Liabilities_t}$
	$Cash\ Ratio = \frac{Cash_t}{Current\ Liabilities_t}$
	$Cash\ Coverage\ Ratio = \frac{EBIT_t + D\&A_t}{Net\ Interest_t}$
	$Interest\ Coverage\ Ratio = \frac{EBIT_t}{Net\ Interest_t}$
Valuation factors	$Price - to - Earning = \frac{Open\ Price_t}{Earning\ per\ share_t}$
	$Price\ to\ book\ Ratio = \frac{Open\ Price_t}{Book\ Value_t}$
	$Price\ to\ sales\ Ratio = \frac{Open\ Price_t}{Revenue_t}$
	$\frac{EV_t}{EBITDA_t}$
	$\frac{Market\ Capitalization}{Free\ Cash\ Flow} = \frac{Market\ Capitalization_t}{Cash\ from\ operation_t - Capex_t}$
	$\frac{Market\ Capitalization}{Tangible\ Book\ Value} = \frac{Market\ Capitalization_t}{Total\ Fixed\ Assets_t}$
	$\frac{M}{B} = \frac{Market\ Capitalization_t}{(Total\ Assets_t - Total\ Liabilities_t)}$
	$\frac{EV}{OI} = \frac{EV_t}{EBIT_t}$
Leverage factors	$Debt\ Ratio = \frac{Total\ Liabilities_t}{Total\ Assets_t}$
	$Debt - Equity\ Ratio = \frac{Total\ Liabilites_t}{Total\ Equity_t}$
	$Long - term\ Debt\ Ratio = \frac{Long - term\ Debt_t}{Total\ Assets_t}$
Financial quality factors	$Return\ on\ Equity\ (QTD) = \frac{Net\ Income_t}{Total\ Equity_t}$
	$Return\ on\ Assets\ (QTD) = \frac{Net\ Income_t}{Total\ Assets_t}$
	$Return\ on\ Assets\ (TTM) = \frac{Net\ profit\ (TTM)_t}{Total\ Assets\ (TTM)_t}$
	$Profit\ Margin = \frac{Net\ Income_t}{Revenue_t}$
	$EBITDA - margin = \frac{EBITDA_t}{Revenue_t}$
	$Operating\ Margin = \frac{EBIT_t}{Revenue_t}$
	$Gross\ Margin = \frac{Revenue_t - Operating\ Expenses_t}{Revenue_t}$
	Cash from operating (TTM) / Net Profit (TTM) $= \frac{Cash\ from\ operations\ (TTM)_t}{Net\ Profit\ (TTM)_t}$
	Cash from operating / Net Profit $= \frac{Cash\ from\ Operations_t}{Net\ Profit_t}$

	$\text{Revenue (TTM) / Market Capitalization} = \frac{Revenue\ (TTM)_t}{\text{Market Capitalization}_t}$
	$\text{Net Profit (TTM) / Market Capitalization} = \frac{Net\ Profit\ (TTM)_t}{\text{Market Capitalization}_t}$
	$\text{Net Profit / Market Capitalization} = \frac{Net\ Profit_t}{\text{Market Capitalization}_t}$
	$PEG\ Q = \frac{P/E\ Ratio_t}{Quarterly\ change\ in\ EPS_t}$
	$Earnings\ yield = \frac{EBIT_t}{EV_t}$
	$Asset\ Turnover = \frac{Revenue_t}{Total\ Assets_t}$
	$Inventory\ Turnover = \frac{Revenue_t}{Inventory_t}$
	$Recivables\ Turnover = \frac{Revenue_t}{Recievables_t}$
	$Payable\ Turnover = \frac{Revenue_t}{Account\ Payable_t}$
1-year growth in fundamental	$Change\ in\ Revenue(YoY) = \frac{Revenue_t}{Revenue_{t-4Q}}$
	$Change\ in\ \ EBIT\ (YoY) = \frac{EBIT_t}{EBIT_{t-4Q}}$
	$Change\ in\ \ EBITDA\ (YoY) = \frac{EBITDA_t}{EBITDA_{t-4Q}}$
	$Change\ in\ \ Net\ Income\ (YoY) = \frac{Net\ Income_t}{Net\ Income_{t-4Q}}$
	$Change\ in\ \ Cash\ from\ Operations\ (YoY) = \frac{Cash\ from\ Operations_t}{Cash\ from\ Operations_{t-4Q}}$
	$Change\ in\ \ Current\ Assets\ (YoY) = \frac{Current\ Assets_t}{Current\ Assets_{t-4Q}}$
	$Change\ in\ \ Current\ Liabilities\ (YoY) = \frac{Current\ Liabilities_t}{Current\ Liabilities_{t-4Q}}$
	$Change\ in\ Total\ Equity\ (YoY) = \frac{Total\ Equity_t}{Total\ Equity_{t-4Q}}$
	$Change\ in\ Total\ Liabilities\ (YoY) = \frac{Total\ Liabilities_t}{Total\ Liabilities_{t-4Q}}$
	$Change\ in\ Total\ Assets\ (YoY) = \frac{Total\ Assets_t}{Total\ Assets_{t-4Q}}$
Quarterly change in fundamentals	$Change\ in\ Revenue\ (QoQ) = \frac{Revenue_t}{Revenue_{t-1Q}}$
	$Change\ in\ Operating\ Expenses\ (QoQ) = \frac{OPEX_t}{OPEX_{t-1Q}}$
	$Change\ in\ Current\ Liabilities\ (QoQ) = \frac{Current\ Liabilities_t}{Current\ Liabilities_{t-1Q}}$
	$Change\ in\ Current\ Assets\ (QoQ) = \frac{Current\ Assets_t}{Current\ Assets_{t-1Q}}$
	$Change\ in\ Total\ Assets\ (QoQ) = \frac{Total\ Assets_t}{Total\ Assets_{t-1Q}}$
	$Change\ in\ Net\ Income\ (QoQ) = \frac{Net\ Income_t}{Net\ Income_{t-1Q}}$
	$Change\ in\ Total\ Liabilities\ (QoQ) = \frac{Total\ Liabilities_t}{Total\ Liabilities_{t-1Q}}$
	$Change\ in\ Long-Term\ Debt\ (QoQ) = \frac{Long-Term\ Debt_t}{Long-Term\ Debt_{t-1Q}}$
	$Change\ in\ Earnings\ per\ Share\ (QoQ) = \frac{Earnings\ per\ Share_t}{Earnings\ per\ Share_{t-1Q}}$
	$Change\ in\ Intangible\ Assets\ (QoQ) = \frac{Intangible\ Assets_t}{Intangible\ Assets_{t-1Q}}$
	$Change\ in\ Total\ Equity\ (QoQ) = \frac{Total\ Equity_t}{Total\ Equity_{t-1Q}}$
	$Change\ in\ Enterprise\ Value\ (QoQ) = \frac{Enterprise\ value_t}{Enterprise\ value_{t-1Q}}$

	$Change\ in\ Profit\ Margin\ (QoQ) = \frac{Profti\ Margin_t}{Profit\ Margin_{t-1Q}}$
Momentum Factors	$1 - Month\ Return = \frac{Price_t}{Price_{t-1M}}$
	$2 - Month\ Return = \frac{Price_t}{Price_{t-2M}}$
	$3 - Month\ Return\ Close = \frac{Close\ Price_t}{Close\ Price_{t-3M}}$
	$3 - Month\ Return\ Open = \frac{Open\ Price_t}{Open\ Price_{t-3M}}$
	$6 - Month\ Return = \frac{Price_t}{Price_{t-6M}}$
	$12 - Month\ Return = \frac{Price_t}{Price_{t-12M}}$
	$Last\ Quarter's\ Open\ Price = Open\ Price_{t-1Q}$
	$Open\ Price\ 2 - M = Open\ Price_{t-2M}$
	$Open\ Price\ 1 - M = Open\ Price_{t-1M}$
	$Open\ Price = Open\ Price_t$
Macroeconomic Factors	$Consumer\ price\ index_t\ (YoY)$
	$Price\ of\ Brent\ Oil = Price\ of\ Brent\ Oil_t$
	$3 - Month\ Interbank\ Rate = 3 - Month\ Interbank\ Rate_t$
	$3 - Month\ Government\ Bond = 3 - Month\ Government\ Bond_t$
	$10 - Year\ Government\ Bond = 10 - Year\ Government\ Bond_t$
	$Unemployment\ Rate = Unemployment\ Rate_t$
	$Gold\ Price = Gold\ Price_t$
	$Quarterly\ change\ in\ GDP = \frac{GDP_t}{GDP_{t-1Q}}$
	$12 - Month\ change\ in\ US\ Interest\ rate = \frac{US\ Interest\ Rate_t}{US\ Interest\ Rate_{t-12M}}$
	$6 - Month\ change\ in\ US\ Interest\ rate = \frac{US\ Interest\ Rate_t}{US\ Interest\ Rate_{t-6M}}$
	$3 - Month\ change\ in\ US\ Interest\ rate = \frac{US\ Interest\ Rate_t}{US\ Interest\ Rate_{t-3M}}$
	$2 - Month\ change\ in\ US\ Interest\ rate = \frac{US\ Interest\ Rate_t}{US\ Interest\ Rate_{t-2M}}$
	$1 - Month\ change\ in\ US\ Interest\ rate = \frac{US\ Interest\ Rate_t}{US\ Interest\ Rate_{t-1M}}$
	$US\ Bid\ Rate = US\ Bid\ Rate_t$
	$12 - Month\ change\ in\ NOR/SWE\ Interest\ rate = \frac{NOR/SWE\ Interest\ Rate_t}{NOR/SWE\ Interest\ Rate_{t-12M}}$
	$6 - Month\ change\ in\ NOR/SWE\ Interest\ rate = \frac{NOR/SWE\ Interest\ Rate_t}{NOR/SWE\ Interest\ Rate_{t-6M}}$
	$3 - Month\ change\ in\ NOR/SWE\ Interest\ rate = \frac{NOR/SWE\ Interest\ Rate_t}{NOR/SWE\ Interest\ Rate_{t-3M}}$
	$2 - Month\ change\ in\ NOR/SWE\ Interest\ rate = \frac{NOR/SWE\ Interest\ Rate_t}{NOR/SWE\ Interest\ Rate_{t-2M}}$
	$1 - Month\ change\ in\ NOR/SWE\ Interest\ rate = \frac{NOR/SWE\ Interest\ Rate_t}{NOR/SWE\ Interest\ Rate_{t-1M}}$

	$NOR/SWE\ Bid\ Rate = NOR/SWE\ Bid\ Rate_t$
	$12 - Month\ change\ in\ Brent\ Price = \frac{Brent\ price_t}{Brent\ price_{t-12M}}$
	$6 - Month\ change\ in\ Brent\ Price = \frac{Brent\ price_t}{Brent\ price_{t-6M}}$
	$3 - Month\ change\ in\ Brent\ Price = \frac{Brent\ price_t}{Brent\ price_{t-3M}}$
	$2 - Month\ change\ in\ Brent\ Price = \frac{Brent\ price_t}{Brent\ price_{t-2M}}$
	$1 - Month\ change\ in\ Brent\ Price = \frac{Brent\ price_t}{Brent\ price_{t-1M}}$
	$Last\ Quarter\ Brent\ Price = Price\ of\ Brent_{t-1}$
	$12 - Month\ change\ in\ USD\ Exchange\ rate = \frac{USD\ to\ NOR/SWE_t}{USD\ to\ NOR/SWE_{t-12M}}$
	$6 - Month\ change\ in\ USD\ Exchange\ rate = \frac{USD\ to\ NOR/SWE_t}{USD\ to\ NOR/SWE_{t-6M}}$
	$3 - Month\ change\ in\ USD\ Exchange\ rate = \frac{USD\ to\ NOR/SWE_t}{USD\ to\ NOR/SWE_{t-3M}}$
	$2 - Month\ change\ in\ USD\ Exchange\ rate = \frac{USD\ to\ NOR/SWE_t}{USD\ to\ NOR/SWE_{t-2M}}$
	$1 - Month\ change\ in\ USD\ Exchange\ rate = \frac{USD\ to\ NOR/SWE_t}{USD\ to\ NOR/SWE_{t-1M}}$
	$12 - Month\ change\ in\ EUR\ Exchange\ rate = \frac{EUR\ to\ NOR/SWE_t}{EUR\ to\ NOR/SWE_{t-12M}}$
	$6 - Month\ change\ in\ EUR\ Exchange\ rate = \frac{EUR\ to\ NOR/SWE_t}{EUR\ to\ NOR/SWE_{t-6M}}$
	$3 - Month\ change\ in\ EUR\ Exchange\ rate = \frac{EUR\ to\ NOR/SWE_t}{EUR\ to\ NOR/SWE_{t-3M}}$
	$2 - Month\ change\ in\ EUR\ Exchange\ rate = \frac{EUR\ to\ NOR/SWE_t}{EUR\ to\ NOR/SWE_{t-2M}}$
	$1 - Month\ change\ in\ EUR\ Exchange\ rate = \frac{EUR\ to\ NOR/SWE_t}{EUR\ to\ NOR/SWE_{t-1M}}$
	$Last\ Quarter\ Index\ Return = Index\ Return_{t-1Q}$
	$Change\ in\ Index\ Return = \frac{Index\ Return_{t-1Q}}{Index\ Return_{t-2Q}}$
	$Change\ in\ Price\ of\ Brent\ (QoQ) = \frac{Price\ of\ Brent_t}{Price\ of\ Brent_{t-1Q}}$
	$Change\ in\ Gold\ Price\ (QoQ) = \frac{Gold\ Price_t}{Gold\ Price_{t-1Q}}$
	$Change\ in\ 3 - Month\ Government\ Bond\ Yield = \frac{3 - Month\ Government\ Bond\ Yield_t}{3 - Month\ Government\ Bond\ Yield_{t-1Q}}$
	$Change\ in\ 10 - Year\ Government\ Bond\ Yield = \frac{10 - Year\ Government\ Bond\ Yield_t}{10 - Year\ Government\ Bond\ Yield_{t-1Q}}$

Table 10.3: Feature computation for all variables included in the final dataset

10.4 Appendix D: Hyper-parameters

Below we outline the hyper-parameter values utilized for all the predictive models presented in the thesis. All Machine Learning models have the same hyper-parameters regardless of portfolio construction and/or feature selection. Hence, the only differences are the input features and type of algorithms. The hyper-parameters are set to the default values provided by sklearn, for simplicity.

	Random Forest	Support Vector Machine
C	-	1
Kernel	-	RBF
Number of estimators	100	-
Max features	$\sqrt{n}$	-
Criterion	Gini	-
Max depth	None	-
Gamma	-	$\frac{1}{n * var(X)}$

1) *n is the number of features*

2) *X is the array of independent variables*

Table 10.4: Hyperparameters for the Random Forest model and Support Vector Machine model.