UiT The Arctic University of Norway

School of Business and Economics

Can machine learning beat the Norwegian stock market?

A comparison of popular machine learning models

Martin Aronsen and Christian Markussen

Master’s Thesis in Business Administration, BED-3901, June 2023

i. Acknowledgments

This thesis marks the end of our academic journey at the School of Business and Economics at UiT. As the last academic work for our master's degree, this thesis consists of 30-ECTS. We are writing this thesis for our minor subject, Finance. Dwelling into the field of machine-learning has been a pleasant experience, which has given us many challenges, leading to the lowest of lows and highest of highs. And we encourage anyone who is curious about machine-learning to write their thesis about it.

We would like to thank our supervisor, Thomas Leirvik. He has been a great sparring partner when it comes to developing the machine-learning models, as well as giving us ideas on how to improve upon our work.

Lastly, we would like to thank our fellow students, especially those who have joined us for excessive coffee breaks. If it weren't for those excessive breaks, we would have had much less stress at the end of our journey.

Tromsø, June 1st, 2023

Martin Aronsen & Christian Markussen

Keywords: Finance, Machine Learning, Market Efficiency, Oslo Stock Exchange, Portfolio

ii. Abstract

Multiple studies on the performance of machine-learning stock portfolios have shown the efficacy of machine-learning portfolios on large stock exchanges, especially the American- and Chinese market. Fewer studies have been conducted on smaller cap markets, which consists of smaller, less-liquid investment options. The purpose of this thesis is therefore to explore the possibilities to beat the Norwegian stock market using machine-learning modalities. Eight different machine-learning portfolios have been constructed based on probability outputs of support vector machines, random forests and logistic regression created using the R software and packages “e1071”, “randomForest”, “gbm” and “caret”.

Portfolios are tested from the end of 2013 to the end of 2022. Results of the thesis are in line with previous research that apply machine learning on the Oslo stock exchange for early periods in the sample, but find different results for the extended period. Machine-learning portfolios with monthly holding periods perform well before 2020, particularly the random forest portfolio. They do however lose their predictive power after this period and generate negative return beginning in 2021. Returns from daily portfolios are eaten up by transaction costs in multiple periods before 2020 and thus fail to consistently outperform the market. Some daily portfolios so show promise in the later period where the monthly portfolios underperform. The thesis therefore concludes that while machine-learning does show some promise on the Norwegian stock market, they cannot be relied upon to generate consistent outperformance over the benchmark index.

Table of Contents

List of Figures

List of Tables

1 Introduction

Financial literature has previously focused on market efficiency and whether it is possible to outperform the market. It is currently inconclusive if it is possible to consistently beat the market with trading strategies. However, new strategies and methods are being employed constantly. Investors consider technical and fundamental properties within a stock or stock market when creating strategies.

In recent years, machine-learning has been applied to the stock markets to predict future price movements. These models use powerful algorithms to capture relationships between technical and fundamental factors for an asset and the underlying returns. Amongst the popular models are Artificial Neural Networks, Support Vector Machines (SVM), Logistic Regression (LR), and Random Forest (RF).

Krauss et al. (2017) found that Deep Neural Networks, Gradient Boosted Trees, and RF models performed well on the S&P 500. Their feature space was the same developed by Takeuchi and Lee (2013), consisting of price momentums across different periods. They achieved a Sharpe ratio of 5,11 before transaction costs with the RF model and a Sharpe ratio of 1,9 after costs. In addition, the RF model performed well compared to the market index Sharpe ratio of 0,35. However, the period analyzed by Krauss et al. (2017) ranged from 1992 to 2015. They discovered that their machine-learning models performed well at the start and that returns declined in the later years, leading to negative returns after transaction costs.

Tan et al. (2019) used a RF model to predict stock excess returns on the Chinese stock exchange. They used a 5-class classification problem combined with technical and fundamental features for the RF model, and they achieved a Sharpe ratio of 2,75. However, Tan et al. (2019) compared their model with a model featuring the same feature space as the one used in Krauss et al. (2017) and found that the momentum features granted greater returns and Sharpe ratios.

Expanding upon Tan et al. (2019), Kilskar (2020) wrote her master thesis using the same model and features on the Norwegian stock exchange. She found similar results as the ones on the Chinese stock exchange, with a Sharpe ratio of 2,44. Similar to previous findings

(Krauss et al., 2017; Tan et al., 2019), Kilskar (2020) found that machine-learning models tend to perform well in earlier periods before their performance drops later.

1.1 Problem statement

Extensive research shows that machine-learning models have been able to perform well in large stock markets, like the USA and Chinese markets. However, few articles explore how the machine-learning models perform in smaller markets such as the Oslo Stock Exchange (OSE). Many machine-learning studies use large market indexes as their investment universe. In large stock markets, these consists of many large and highly liquid stocks. This is not the case for lower-cap markets such as the Oslo Stock Exchange (OSE). As a comparison, the S&P500 consists of 500 of the largest stocks on the American market, while the entire dataset used in this thesis has 424 unique stocks, and 356 after filtering out low-priced stocks (under 5NOK). Therefore, we will check how some of the popular machine-learning models perform on OSE compared to the OSEBX and against each other. To check if the machine-learning models can beat the OSEBX, we have chosen the following problem statement:

"Can machine-learning models beat the OSEBX?"

To answer this question, portfolios with daily and monthly holding periods are constructed using machine-learning models and compared with the performance of the OSEBX. Although the RF model has been applied to the OSE with monthly and daily holding periods previously (Kilskar, 2020), no study has applied the same methodology with other machine-learning models. Therefore, we seek to answer another research question as well:

"How do the different machine-learning models compare to each other?"

To answer these questions, a method like the ones of Kilskar (2020), Tan et al. (2019), and Krauss et al. (2017) will be used.

2 Risk and return in the financial market

2.1 Modern Portfolio Theory

Markowitz (1952) is one of the most pathbreaking scientific publications in financial economics and typically marks the start of what's referred to as *modern portfolio theory*. The paper presents an important contribution to the understanding of the relationship between the selection of specific financial assets, their returns, and the overall portfolio risk. Given the assumption that an investor is rational and that risk is something to be avoided, a theoretical framework for maximizing returns for a given level of risk or minimizing risk for a given level of return is provided.

For a portfolio consisting of different financial assets, the expected return of the portfolio can be given by the expression:

$$E(r_p) = \sum_{i=1}^{n} w_i E(r_i) \tag{1}$$

Where $E(r_p)$ is the expected value of the portfolio, w_i is the weight of asset *i* in the portfolio, and $E(r_i)$ is the expected return of asset i.

The variance, or risk, of this portfolio can then be calculated as:

$$\sigma_p^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i * \sigma_i) * (w_j * \sigma_j) * \rho_{i,j} \tag{2}$$

Where:

σ_p^2 is the variance of the portfolio,

w_i and w_j are the weights of assets i and j,

$\sigma_i \; and \; \sigma_j$ are the standard deviations, or volatilities, of the returns on asset *i* and *j*,

$\rho_{i,j}$ is the correlation between returns of assets i and j

Equation (2) shows that the portfolio risk, represented by the portfolio's variance, is dependent on the risk of the individual components of the portfolio, their weights, and the correlation of their returns. Because portfolio risk depends on the correlation of the individual assets in the portfolio, overall portfolio risk can be reduced by including assets with lower correlations. Arguing that an investor should be concerned with not only maximizing the returns of the portfolio but also minimizing its risk, Markowitz (1952) shows that an efficient frontier of portfolios that maximizes returns for every unit of risk (or alternately minimizes risk for every unit of return) can be made. The efficient fronter provides a framework for how a rational investor should allocate their wealth in accordance with their own risk tolerance. It also shows that in a market of rational investors, an investor would have to increase their risk exposure in order to increase returns.

2.2 Capital Asset Pricing Model

Building on the findings of Markowitz (1952) and Tobin (1958), the capital market pricing model (CAPM) was co-developed by Sharpe (1964), Lintner (1965), Mossin (1966), and Black (1972). It is a market-equilibrium model where efficient asset combinations can represent the market. Moreover, because an individual asset is part of the market portfolio, the relationship between the returns of the individual asset and the market portfolio is similar to that of a linear regression.

The CAPM can be given with the expression:

$$E(r_i) - r_f = \alpha + \beta(r_m - r_f) \tag{3}$$

Or:

$$E(r_i) = \alpha + \beta\left(r_m - r_f\right) + r_f \tag{4}$$

Where $E(r_i)$ is the expected return of asset i, r_f is the risk-free borrowing and lending rate, β is the sensitivity of returns of asset i with regards to market returns, and r_m is the market return.

The market factor coefficient β can be expressed as:

$$\beta = \frac{\sigma_{im}^2}{\sigma_m^2} \tag{5}$$

Where σ_{im}^2 is the covariance of returns for asset i and the market, and σ_m^2 is the variance of returns in the market portfolio.

Because the individual asset is correlated with the market, some of the returns of the individual asset can be explained by the returns for the market. Since the market "portfolio" in theory includes the individual asset, there are no more options to diversify the individual asset's risk away. The slope of the regression therefore signifies risk that cannot be diversified and should impact the expected return for the individual asset. An asset that has a higher (lower) beta is more (less) sensitive to general changes in the market and has higher (lower) undiversifiable risk. The asset should therefore also be expected to give a higher (lower) return.

2.3 Three – and four-factor model

While the CAPM assumes that all expected returns can be approximated by the general expectation of market returns and the beta, empirical studies have found that company-related variables such as size (Banz, 1981), price-to-earnings ratios (Basu, 1983), and book-to-market ratios (Rosenberg et al., 1998) can, in part, explain returns of a specific stock. Based on these findings, Fama and French (1993) expand the CAPM by including size and value factors.

Fama and French's three-factor model is expressed as:

$$R_{it} = \alpha + \beta_1 R_{mt} + \beta_2 SMB + \beta_3 HML + \Sigma\epsilon \tag{6}$$

Where R_{it} is the risk premium for asset *i* at time *t*, R_{mt} is the market risk premium at time *t*, SMB is a factor for size, HML is a factor for book-to-market ratio and β_1, β_2 and β_3 are the factor coefficients, $\Sigma\epsilon$ is the regression error which is assumed to have a mean of zero, and α are returns not explained by the factors in the model.

SMB stands for "small minus big" and is a factor consisting of the difference in returns for portfolios of small market value stocks and big market value stocks. In Fama and French's original model, for example, the data was divided into a total of six value-weighted portfolios, with three portfolios being in the lowest decile with regard to size and three varying degrees of book-to-market ratios. The other three portfolios consisted of the largest decile of market-value stocks with the same three levels of book-to-market ratios. SMB then became the monthly difference of the simple average of the three small market value portfolios and the three big market value portfolios (Fama & French, 1993). The SMB formula is given in (7):

$$SMB = \frac{1}{3}(S,H + S,M + S,L) - \frac{1}{3}(B,H + B,M + B,L) \tag{7}$$

Where:

S = The market value of the companies in the portfolio are in the small size category

B = The companies in the portfolio are in the large-size category

H = The companies in the portfolio have a high book-to-market equity ratio

M = The companies in the portfolio have an average book-to-market equity ratio

L = The companies in the portfolio have a low book-to-market equity ratio

HML stands for "high minus low" and is a factor for the difference in returns for portfolios of companies with high book-to-market equity and those with low book-to-market equity. The HML factor was constructed similarly to SMB. Two of the portfolios used in creating SMB were removed, as these had an average level of book-to-market ratios. HML then becomes the difference in simple average monthly returns for the two high book-to-market portfolios and the two low book-to-market portfolios. Following the same notation as (7), Equation (8) shows the HML function.

$$HML = \frac{1}{2}(S,H + B,H) - \frac{1}{2}(S,L + B,L) \tag{8}$$

Fama and French (1993) argue that while neither of their factors are based on theoretical concepts such as modern portfolio theory and equilibrium models, they can serve as proxies for common risk factors because they reflect economic fundamentals. Companies with a high book value of equity compared to their market value of equity tend to have low earnings on assets, while companies with a high market value of equity compared to their book value tend to have high earnings (Rosenberg et al., 1998). High book-to-market ratio companies are therefore riskier and should demand a higher return. Larger companies also tended to have higher earnings during the period Fama and French analyzed, as bigger companies were more likely to still perform well during an economic downturn (Fama & French, 1993).

Carhart (1997) adds to the model of Fama and French by including a factor that is commonly referred to as the momentum factor. The factor is based on an anomaly explored in Jegadeesh and Titman (1993), who found significantly higher returns than the market by utilizing a trading rule of buying stocks that had performed well in the recent past and selling those who had performed badly. Carhart's four-factor model can be expressed as:

$$R_{it} = \alpha + \beta_1 R_{mt} + \beta_2 SMB + \beta_3 HML + \beta_4 WML + \epsilon_{it} \tag{9}$$

The winners-minus-loser (WML) factor is created by utilizing the same method as in Jegadeesh and Titman (1993), constructing a portfolio that sells past losers and buys past winners. Losers are defined as the lowest 30% of the prior year return distribution, while winners are the top 30%.

As the effects of the different factors on stock returns have been empirically proven, it is normal to view them as common market effects. Because of this, it is possible to use the three- or four-factor model as an analytical tool when evaluating the performance of specific portfolios. Performing a linear regression of a portfolio's returns with the factors in the three- or four-factor model can reveal specific characteristics of the portfolio. Using the model's regression coefficients, it is possible to see if a portfolio's returns come from investing in, for example, smaller companies, value companies, or growth companies.

The regression intercept is also interesting, as it shows the returns in the portfolio not captured by the common market factors, and typically shows returns that are attributable to the investor's choices. A significant intercept, commonly referred to as alpha, therefore signifies that the investor's strategy has added (or subtracted) some value to (from) the portfolio that cannot be attributed to common factors.

2.4 Sharpe-ratio

Another comparative tool when analyzing portfolios is rates that display the relationship between a portfolio's return and risk. The most well-known of these rates is the Sharpe-rate developed in Sharpe (1966).

The Sharpe-ratio can be written as:

$$S_p = \frac{r_p - r_f}{\sigma_p} \tag{10}$$

Where:

S_p is the Sharpe-ratio of portfolio p.

r_p are the returns of portfolio p.

r_f is the risk-free rate.

σ_p is the standard deviation for portfolio p.

By dividing a portfolio's excess return by the portfolio's standard deviation, the Sharpe-ratio shows the excess return for each unit of standard deviation, which should be interpreted as the return for each unit of portfolio risk. The Sharp ratio thus captures the risk and return relationship first presented in Markowitz (1952).

The Fama-French-Carhart model is used to explain the returns of different machine-learning portfolios created in this thesis, while the Sharpe-ratio is used to compare the performance of the portfolios with the OSEBX index. It should be noted that we do not use an approximation of risk-free rate in this thesis. The Sharpe ratio will therefore not be calculated using returns in excess of the risk-free rate but simply the returns of the portfolios and the market index. This is not an issue, since the goal is to compare these portfolios to the market index. Still, it is worth noting that Sharpe ratios presented in this thesis might be higher for a given return than what is commonly reported elsewhere.

2.5 Beating the market

Traditional finance theories propose that financial markets consist of rational, risk-averse investors seeking to maximize returns for any given level of risk. The market thus reaches an equilibrium where no opportunities to generate excess returns without bearing larger risk exist. Despite this implication from the theoretical framework, many investors seek to create strategies that give a better risk-reward relationship than the general market can achieve. This is commonly referred to as trying to "beat the market".

There are many possible strategies to choose from when attempting to beat the market, but it is common to split these strategies into two categories; fundamental- and technical analysis. Fundamental analysis seeks to calculate the true intrinsic value of a stock by analyzing fundamental economic factors, often by estimating an expected net future cashflow. Typically, this includes studying the company's financials, industry, competitors, and the economy as a whole. A fundamental analyst looks for companies that are undervalued according to their analysis and buys them, expecting the price to increase to their "true" value in the future. As a result, fundamental analysts often have a long-term investment horizon and invest more in "value"-stocks (Jordanoski & Petrusheva, 2016).

Technical analysis, also called *chartin*g, only uses past trading data, such as trading volumes, bid-ask spreads, and prices to predict future price movements. The technical analyst is therefore not interested in the underlying company of the stock, but simply the stock itself. A few assumptions about the market are often attributed to the technical analyst. Firstly, all relevant company information is already priced in the stock, so fundamental analysis is unnecessary. Secondly, it is assumed that prices are not random, but move in trends. Lastly, history is repeated, or in other words, past prices can predict future prices (Bonga, 2015).

Despite the underlying assumptions about asset prices differing between fundamental- and technical analysts, there is an assumption both have in common: the market does, at least sometimes, price assets incorrectly. Whether fundamental or technical analysis can consistently beat the market is inconclusive, however. Due to the difficulty of predicting financial time series, there is no universal consensus among academics and practitioners regarding the plausibility of predicting them at all. An important point of contention in this debate is the efficiency of financial markets.

According to Fama (1965)'s *efficient market hypothesis*, markets will be efficient as long as there is a sufficient amount of intelligent fundamental- and technical analysts. If a correlation between the development of an asset price and either its price- and order history or the fundamental values of the underlying company exists, these investors will find the relationships and act on them in order to profit. When enough investors find these patterns, prices will be driven to a new equilibrium quickly after the relationship is discovered. Fama (1965) thus concludes that a market will be efficient if the prices fully reflect all current relevant information.

Fama (1970) further divides market efficiency into three degrees of efficiency; weak, semi-strong- and strong-form efficiency. The degree of market efficiency is based on the amount of information that is used to price an asset. In weak form efficiency, the information set "only" consists of all historical market data, such as previous and current asset prices and purchase volumes. This implies that if weak form efficiency holds, trading using technical indicators would be impossible. In semi-strong form efficiency, the information set consists of historical price data and all publicly relevant information for a company's securities. For semi-strong efficient markets, it is implied that neither technical- nor fundamental analysis can be used to beat the market. For strong form efficient markets, the information set contains absolutely all information that could be relevant to a company's securities, both private and public. In strong form efficient markets, not even company insiders should be able to use their private information to generate long-term returns in excess of the market.

Closely related to the efficient market hypothesis is the idea of prices in financial markets as a random walk. Empirically this was first observed in Kendall and Hill (1953), where Kendall remarked that time series from industrial indexes behaved "…*almost like a wandering series*" (Kendall & Hill, 1953). Malkiel (2003) provides a suggestion for the practical reasoning behind this phenomenon. If the efficient market hypothesis is correct, asset prices will reflect all current available information. When prices instantly or almost instantly reflect new information, tomorrow's price becomes a function of tomorrow's news. News is argued to be random and unpredictable and because tomorrow’s prices are a function of tomorrows’ news, they must also be random and unpredictable (Malkiel, 2003).

Many finance academics and practitioners challenge the assumption of the efficient market hypothesis and random walk theory, however. For instance, De Bondt and Thaler (1985) discovered a reversal effect in stock momentum from 1962-1982, where portfolios of previous losers outperformed the market by about 19.6%, while a market of past winners underperformed the market by about 5%. As mentioned in the previous section, Jegadeesh and Titman (1993) found that portfolios which bought winners and sold losers of past three- to twelve months generated significant returns above the market in in the following three- to twelve months. Brock et al. (1992) found that buy and sell signals from simple moving average strategies showed predictive capabilities in the American stock market. Similar results were found for multiple European stock markets in Metghalchi et al. (2012), where simple moving average strategies also showed significant excess returns compared to market indexes. A literature review on technical trading rules by Park and Irwin (2007) shows that 56 out 95 modern studies find positive results while utilizing technical trading rules. Coval et al. (2021) also find persistence in the performance of individual investors over a period of six years from 1991 to 1997, where top decile performers in the first three years, outperformed bottom decile performers in the following three years.

In defense of the efficient market hypothesis, Malkiel (2003) argues that while some strategies provide statistically significant excess returns in the periods they were tested, their economic relevance is not certain. As an example, Malkiel states that while momentum strategies showed abnormal returns in some periods of the 1990s, they subsequently underperformed in the year 2000. Schwert (2003) reviews studies on many of the common "anomalies" in returns over the '80s and '90s, such as the size effect, momentum, and the value effect. He notes that many of these effects seem to disappear in the years after they were initially introduced. The apparent reduction in significance is speculated to be due to inherent bias in academics and journals towards presenting findings that challenge existing theory, or that market participants quickly take advantage of new academic findings and cause the effects to disappear (Schwert, 2003). Both Malkiel and Schwert also stress the point of Jensen (1978) that an apparent market inefficiency is only valid if a market participant is able to realize returns from the strategy. In practice this means that the portfolio must survive the transaction costs incurred by employing the strategy.

3 Machine-learning

The early financial theory relies heavily on regression analysis, and most of the financial relationships presented in the previous sections are assumed to be linear. With continuing advances in computing power and the increasing availability of larger electronic datasets, econometricians, statisticians, and data scientists now easily utilize models that can discover nonlinear relationships between variables. Machine-learning models, specifically, have become more common in finance research, and strategies based on probability- or regression output from machine-learning models show promising results compared to older, more "economically intuitive" strategies.

The term "machine-learning" has grown in popularity in recent years. Machine-learning can be described as "A model that uses an algorithm to analyze input variables to produce an output" (Baloglu et al., 2022). Machine-learning algorithms can detect complex relationships between the input variables in large datasets. Typically, variables are called features when talking about machine-learning models.

The dataset must be divided into training and testing data for the machine-learning algorithm to detect relationships between the features. First, the algorithm will analyze training data to produce an output. Then the testing data will be tested against the output to see if the machine-learning model was able to predict the outcome (Baloglu et al., 2022). The following section presents the three machine-learning models that will be applied to our data: random forests, logistic regression, and support vector machines.

3.1 Random forest

As the name suggests, random forest is a form of tree-based models. Tree-based models are made by checking features, then moving on to the next feature in a binomial fashion. Since the tree moves in a binomial fashion, it means that each feature can only have two outcomes. Therefore, such tree-based models can be used to predict the variable being run through the model.

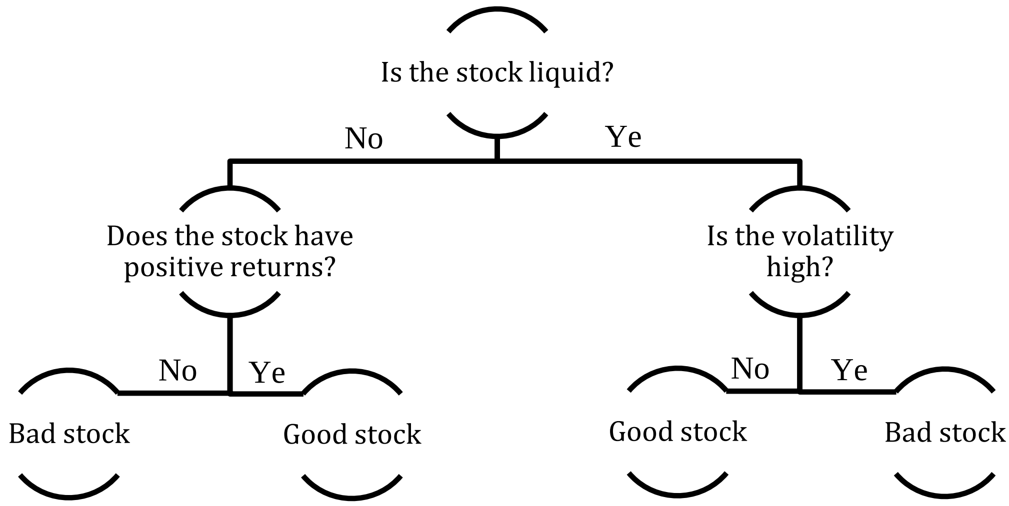

Figure 3.1: Illustration of a Decision Tree

Figure 3.1 shows how a possible decision tree could look like if it were to predict if a stock is "good" or "bad". It shows that the tree starts by checking the value of a feature, then moves down and checks the next feature. When the tree has reached its end, a prediction will be made.

The tree-based model is made by using training data, which will decide the importance of each feature. If a feature is important, it will be placed further up on the tree, since it will affect the outcome to a greater extent than a less important feature. Since the training data is used separately from the test data, errors can be expected regarding deviations from training data and testing data (James et al., 2021, s.327-345).

To prevent some of the errors in the tree-based model related to differences in training and test data, you can employ the use of bagging. Bagging is a form of bootstrapping, which means that you take the training data, split it up, and fit the tree. When this method is applied several times, it will reduce the variance of the prediction made by the model, since it simulates more training data. We denote the number of bags as n_{tree}. (James et al., 2021, s.340-343)

Another implication when making predictions based on a tree-model, is that some features will not contribute as much as they should since the important features are placed further up the tree. This can lead to some false predictions when using tree-based models. When bagging trees, this will lead to the trees becoming correlated with each other since all the trees will include the important features at the start of each tree. (James et al., 2021, s.340-345)

In order to reduce the error related to important features, we can use the random forest model. Random forest reduces the number of features the tree is allowed to choose at each split when making the tree. This will reduce the number of important features at the top of each tree, decorrelating the trees from each other. Features for each split can be denoted as m_{try}. (James et al., 2021, s.343-345)

Since the trees are created using bagging, the variance related to different training data and testing data will be reduced as well. The combination of bagging and reducing the impact of important features make the random forest quite good when making predictions.

3.2 Logistic regression

James et al. (2021, s.133-134) present the logistic regression as a probability function which can be written as:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \tag{11}$$

The logistic regression contains an intercept β_0 which will capture any bias or correct for large values of the X feature. When interpreting the β_1 coefficient, we have to consider if the coefficient is positive or negative, since it will explain if the X feature increases or decreases the probability of the outcome. A positive β_1 coefficient will increase the probability of the outcome, while a negative β_1 coefficient will decrease the probability of the outcome. (James et al., 2021, s.133-137)

When fitting the logistic regression, a method called maximum likelihood is used. Maximum likelihood is used to find the coefficients for the logistic regression that fits best with the model and the training data. (James et al., 2021, s.135-136)

The maximum likelihood function can be written as:

$$\ell(\beta_0, \beta_1) = \prod_{i:y=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})) \tag{12}$$

When the logistic regression model tests for probabilities, the outcome will be presented as a number between 0 and 1. This is due to the regression being placed in the numerator and denominator of the probability function. Because the probability function yields a number between 0 and 1, the probability can be interpreted as a percentage. (James et al., 2021, s.136-137)

As shown in the logistic regression (11), there is only one feature. However, the function can easily be expanded to include multiple features. We can rewrite the (11) function into:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X \ldots + \beta_p X_p}} \tag{13}$$

Function (13) shows the expanded logistic regression model. This extension to the logistic regression allows the use of more features. Same as in the regular logistic regression model, the interpretation of the model is the same. When fitting the coefficients, the maximum likelihood method is still employed. (James et al., 2021, s.137-139)

3.3 Support vector machine

The support vector machine (SVM) is an extension of the support vector classifier and utilizes kernels to enlarge the feature space. (James et al., 2021, s.380)

When creating a SVM, we have to consider a hyperplane, which can be expressed in p-dimensions and written mathematically as:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{14}$$

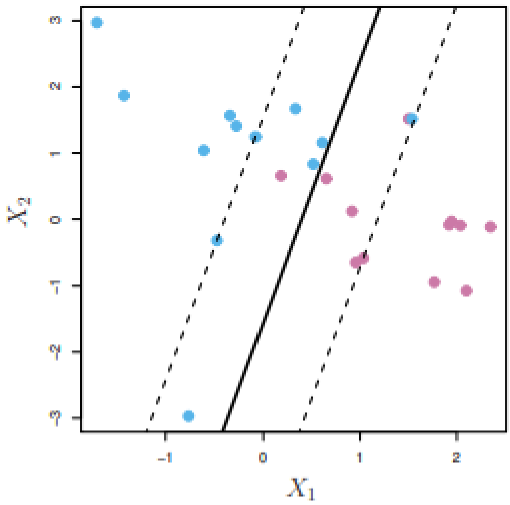

Figure 3.2: Visualization of hyperplane for two dimensions (James et al., 2021, s.378).

Equation (14) shows the hyperplane for p-dimensions and indicates the separation of the observations. Figure 3.2 illustrates the hyperplane for p = 2 dimensions. Separation is caused by (14) not being satisfied. If the equation is greater than 0, it shows that the observation is on one side of the hyperplane. In turn, if (14) is less than 0, the observation is on the other side of the hyperplane. (James et al., 2021, s.368)

In some cases, the hyperplane cannot separate two different observations with different classes. This means that some observations have been placed on the wrong side of the hyperplane. However, the SVM allows for some observations to be placed on the wrong side of the separating hyperplane as long as the majority of the observations are on the right side. The hyperplane will therefore include a margin for classifying the observations. (James et al., 2021, s.371-379)

The linear support vector classifier can be written as:

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle \tag{15}$$

Where $\langle x, x_i \rangle$ is:

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij}, x_{i'j} \tag{16}$$

When we apply the radial kernel method, we can write the support vector machine as:

$$K(x, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) \tag{17}$$

A radial kernel is chosen for the SVMs in this thesis as it is the most common kernel configuration in stock market prediction (e.g., Yu et al., 2014; Zhang et al., 2018; Ma et al., 2021). An illustration of the hyperplane in a radial kernel SVM with p = 2 dimensions is shown in Figure 3.3.

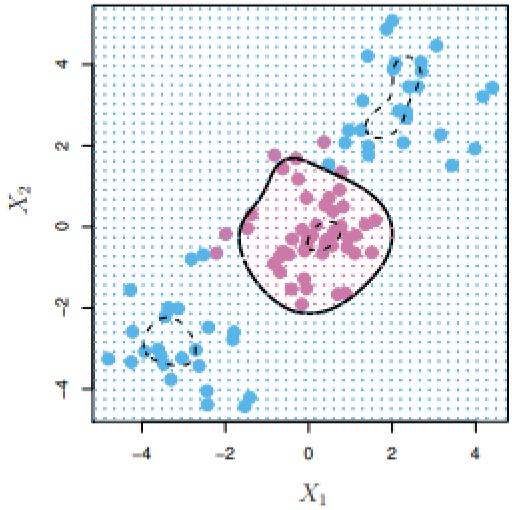

Figure 3.3: Visualization of a radial kernel with two dimensions (James et al., 2021, s.383).

4 Methodology

4.1 Data

The data consists of daily stock data from the Oslo stock exchange and several macro-economic factors. The time period for the dataset ranges from early 2010 to the start of 2023. However, the dataset is trimmed down while generating the model features. After including all macro-economic factors and features, the dataset spans 2548 days from 05/10/2011 to 15/11/2022. An additional 260 days are removed from the investment period to allow for training of the initial models (more on model training and "testing" is explained below). As such, the investment window consists of 2288 days from 29/06/2013 to 15/11/2022.

Stock data from the Oslo stock exchange were collected from the database TITLON. It is worth noting that after the Oslo stock exchange was merged with Euronext on the 30th of

November 2020 (Sirnes, n.d.), TITLON split the datasets into one prior to the merge and one following the merge. Both datasets contain daily stock prices as well as the main OSEBX index. The dataset prior to the merge has access to Fama-French-Carhart factors, yearly accounting data, and outstanding shares, while the dataset after the merge does not.

Index data that are collected from sources other than TITLON are the VIX index, Vanguard's European Stock Index Fund Admiral Shares, the S&P 500, and OMX Nordic 40. Other factors that have been collected are the European Brent Oil spot price, exchange rates between euros and NOK and between US dollars and NOK, as well as the LIBOR 3-month rate. The section "Features: macro-factors" gives a short introduction to these features.

VIX index daily data was downloaded from yahoo finance historical prices (Yahoo Finance, n.d.), OMX Nordic 40 data was downloaded from Nasdaq (Nasdaq Nordic, n.d.), the S&P500 and the Vanguard European index were retrieved from google finance in google spreadsheets using the GOOGLEFINANCE function with the ticker values "INDEXP:.INX" and "MUTF:VEUSX". Europe brent spot prices were downloaded from the U.S. Energy Information Administration (EIA, n.d.). Three month LIBOR rates, as well as EURO-NOK and USD-NOK exchange rates were downloaded from Marketwatch (Marketwatch, n.d-a; Marketwatch, n.d-b; Marketwatch, n.d-c).

4.2 Preprocessing

Some stocks have missing trading days in the dataset, which can be due to an error in the data collection process, instances where the stock was not traded for that respective day, or that the trading for of stock was suspended on the respective day. In cases of missing data, prices are set to the last registered closing price, and trading volume is set to zero.

We have also investigated stocks that have had an extreme return and have found that there are some discrepancies in the dataset. Some stocks have had large price-changes in the dataset that did not coincide with their development on Euronext. These stocks are Seabird Exploration, Seadrill, Carbon Transition, TECO 2030, and Awilco Drilling.

In these specific instances, we have removed Seabird Exploration and Carbon Transition from the dataset outright, while TECO 2030 and Awilco Drilling have been modified. TECO 2030 was found to have their price multiplied by ten and trading volume divided by ten for every day prior to the 2nd of March 2021. Price and volume have therefore been divided and multiplied by ten, respectively. Awilco Drilling was discovered to have a wrong series of stock prices after the 21st of December 2022, and the dates preceding have been removed from the dataset. Seadrill was discovered to have gone bankrupt in late 2018 and re-established in the same year, and the re-established stock had therefore not been separated from the predecessor. This has been resolved by separating the two into two different entities. It is worth noting that Seadrill went bankrupt again in 2022, but due to feature generation, the stock does not meet the feature-generation requirement of having more than 240 trading days.

4.3 Training and testing periods

After edge-cases were removed and missing days were filled in, the dataset was split into training- and testing sections. Typically, the testing section is a hold-out-sample of the data that is used to test a model's performance on samples it has not encountered yet, hence the name "test data". In this thesis, the test portions of the data are used to simulate a trading strategy. For cross-sectional data, it is common to randomly allocate 70-80% of the cross-sectional data to training and 30-20% for testing. It makes little sense to randomly allocate training- and testing data in time series however, as the goal is to predict something that will happen in the future. When splitting time series', it becomes important to take the chronology of the data into account.

While it is possible to use any period before testing to train the data, in order to account for changes in market dynamics over time, an intuitive approach to splitting the data is using a rolling window. For this paper, a financial year is assumed to contain 240 days, split evenly among twelve 20-day months. Using these assumptions, a rolling window approach is illustrated in Figure 4.1.

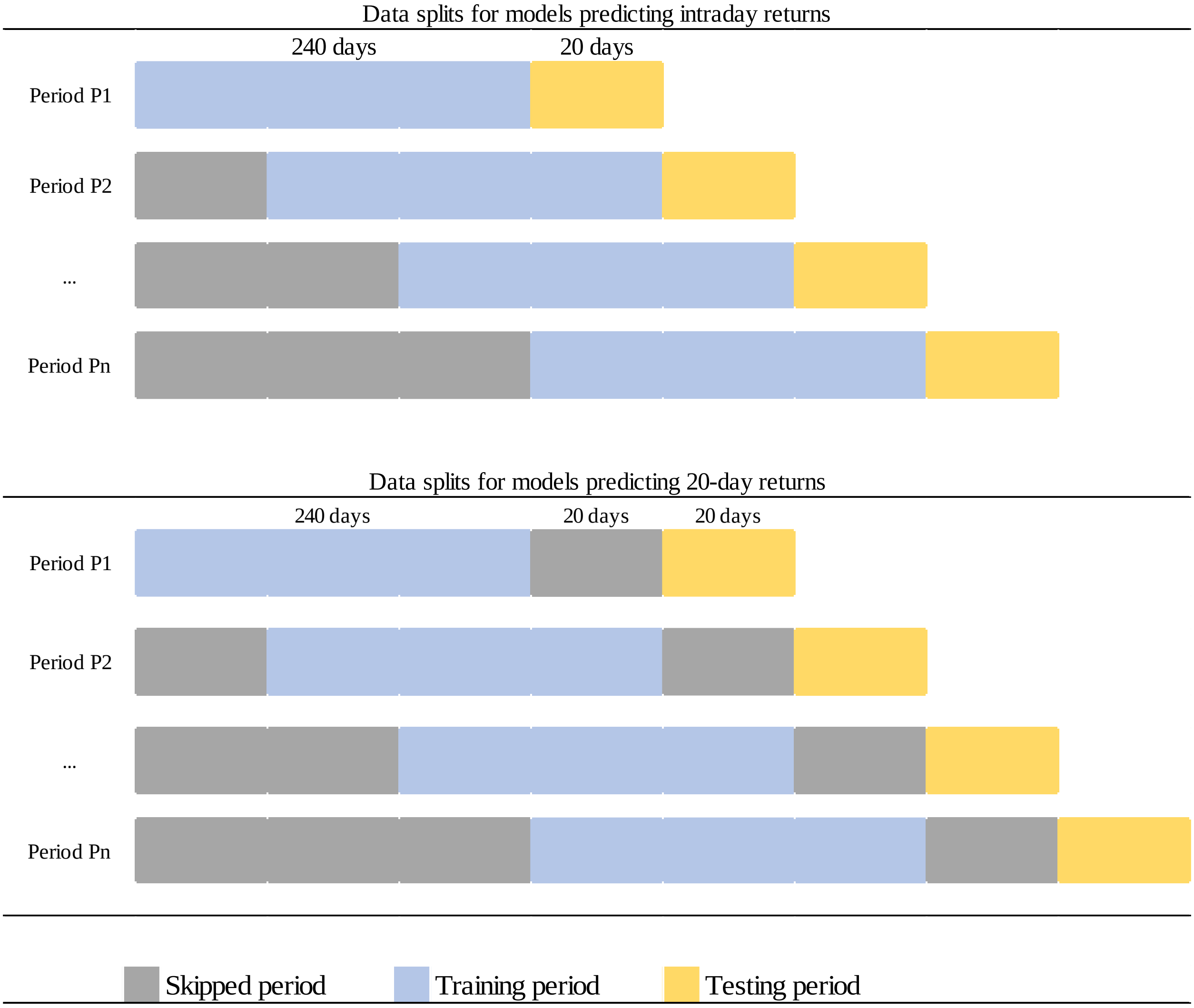

Figure 4.1: Rolling window approach for intraday and monthly models

In period P1, data for 240 days, or approximately a year, is used to train a model, while data from the following 20 days are used to test the model. After testing the model on the 20 days allocated for testing, the training and testing periods are shifted forward by 20 days to period P2. Data corresponding to the first 240 days in period P2 is then used to train a second model, which is used to predict observations within the last 20 days in period P2. This pattern continues until all testing periods have been predicted.

In this paper, two iterations of each machine-learning model are made. One where the model uses daily data to predict the relative position of a stock's next-day intraday returns. The other model uses daily data to predict the relative position of a stock's future monthly (20-day) returns. Daily data is used for the "monthly" models under the hypothesis that keeping the data frequency daily instead of monthly will lead to better predictions due to the model having more observations to use for training.

Since future monthly returns are calculated for each day as $r_{t+20} = \log\left(\frac{C_{t+20}}{O_{t+1}}\right)$, it is important to adjust the rolling window such that the training data does not include features that contain information not available at the time of its calculation. At the start of period t the most recent price information available would be the closing price at time t, meaning that the most recent date we could use to train the model would be at t_{-20}. The training- and testing period thus needs to be separated by a 20-day period to avoid the model using data that would not be available at the time of prediction.

4.4 Penny-stocks

It is common practice to remove "penny stocks", which is usually defined as a stock with a price lower than 5 USD. This is commonly done to reduce the large impact a small absolute change in price has on returns. Since the Oslo stock exchange is a smaller market, using 5 USD as a cut-off would remove most of the investment options. Therefore, stocks that close at a price under 5NOK are counted as penny stocks. For each training-testing period, stocks are removed if either the normal close price or adjusted close price is registered below 5NOK during any day within the testing period or the skippet period between the training and testing set in the case of the 20-day models. In other words, stocks are removed if they close at under 5NOK for any of the 240 days prior to the testing period in the case of the intraday models and removed if they close under 5NOK in any of the 260 days prior to the testing period for the 20-day models. Additionally, stocks with a previous close price or an adjusted close price below 5NOK at the time of investing will not be invested in.

4.5 Data scaling

For distance-based models, such as support vector machines and logistic regression, large differences in sizes and ranges of features can make some features disproportionately affect the predictions of the model, simply because the feature has larger numeric values and larger numeric ranges. To avoid features based on, for example, volume to largely impact the models, the features can be scaled. Each feature is rescaled using (18) to have a zero mean and unit variance.

$$z = \frac{(x - \bar{x})}{\sigma} \tag{18}$$

Features are not rescaled for the random forest as it only utilizes bagging and should not be affected by feature sizes. Feature scaling is done separately for each training-testing period. Because the ranges of the test period would not practically be known at the time of prediction, features are scaled based on means and variances from the corresponding training data. The "e1071" package for R has a scale argument in the svm() function that creates the SVM that internally scales the data using the same methodology as (18). For the logistic regression models, the preProcess() function from the "caret" package is used to scale the data.

4.6 Features

4.6.1 Classification feature

The models will use a classification problem to predict if a stock will become a winner or a loser, which can be interpreted as the probability of a stock being a winner. Classification problems have been shown to perform better than regression problems for predicting financial markets (Leung et al., 2000; Enke & Thawornwong, 2005). When creating a feature to define winners and losers, we follow the method of Krauss et al. (2017), using median returns as a baseline. The output of our models can then be interpreted as the probability of a stock being a winner for the given holding period. We will calculate the returns for each holding period by using the open price, this will allow us to calculate features using the close-price for the day before. Returns for stock s can therefore be written as (19), where t is the point in time from today, and m is the holding period:

$$R_{t,m}^{s} = \log\left(\frac{Close_{t+m}}{Open_{t+1}}\right) \tag{19}$$

With the defined returns, we can define the classification problem by:

$$y^{s} = Winner \mid R_{t.m}^{s} \geq \tilde{R}_{t,m} \tag{20}$$

$$y^{s} = Loser \mid R_{t.m}^{s} < \tilde{R}_{t,m} \tag{21}$$

Our classification feature seeks to capture which stocks will be considered winners by comparing the returns for stock s to the median returns for the given trading period.

Trading periods will be limited to intraday and monthly periods where t = 1 and m = [1, 20]. Both trading periods will apply the same features. Features that are stock specific will be constructed from the dataset as a whole. After the features are constructed, the dataset will look like a u X v matrix, where u are the stocks for each available trading day, and v is the features for each stock.

4.6.2 Predictor features

The predictor features can be broadly separated into four "feature spaces"; Krauss, TTR, Tan, and macro features. "Krauss" and "Tan" features are the momentum features used in Krauss et al. (2017) and the technical indicators of Tan et al. (2019) without turnover (as outstanding share data is not present in the dataset after the Oslo Stock Exchange-Euronext merge). These features were chosen as models created with these features significantly outperformed benchmark indexes on the American- and Chinese stock market in their respective studies, as well as the OSEBX in Kilskar (2020). TTR is a collection of technical indicators made using the TTR (technical trading rules) package in R, while macro factors are based on exchange rates, market indices, and LIBOR. The components of the TTR and macro factor groups are explained in more detail below.

Krauss

$$mom_{t,m}^{s} = \frac{P_t^s}{P_{t-m}^s} - 1 \quad m \in ((1, \dots, 20) \cup (40, \dots, 240)) \tag{22}$$

Where *m* is days.

This represents daily momentum factors for the most recent month and monthly momentum factors for the most recent year.

Tan

Table 4.1: Tan Feature Space

Features	Description	Formula
close_0/close_9, close_0/close_19, close_0/close_39, close_0/close_59, close_0/close_119	A momentum feature to capture the trends for stock prices.	$\frac{P_t^s}{P_{t-m}^s} - 1$ $m \in \{9,19,39,59,119\}$
close_19/close_0, close_39/close_0, close_59/close_0, close_119/close_0	A reversal of the momentum feature.	$\frac{P_{t-m}^s}{P_t^s} - 1$ $m \in \{19,39,59,119\}$
adjusted_close_19/close_59, adjusted_close_19/close_119	A momentum feature excluding the most recent month.	$\frac{P_{t-19}^s}{P_{t-m}^s} - 1$ $m \in \{59,119\}$
vol10/vol20, vol10/vol40, vol10/vol60, vol20/vol40, vol20/vol60, vol40/vol60	Volume acceleration.	$\frac{movavg(volume, m_1)}{movavg(volume, m_2)}$ $m_1 \in \{10,10,10,20,20,40\}$ $m_2 \in \{20,40,60,40,60,60\}$
volatility_10, volatility_20, volatility_40, volatility_60, volatility_120	Volatility for different time periods for the daily returns of the stocks.	$movstd(daily_R, m)$ $m \in \{10,20,40,60,120\}$
std(volume_10), std(volume_20), std(volume_40), std(volume_60), std(volume_120)	Standard deviation for the trading volume for the past m days for each stock.	$movstd(volume, m)$ $m \in \{10,20,40,60,120\}$

TTR

Due to the prevalence of moving average trading rules in academic papers on technical analysis, both simple- and exponential moving averages are included. The ratios of the close-price to moving averages, as well as ratios of short- and longer-term moving averages, are calculated to simulate the common strategy of providing sell or buy signals when one moving average overlaps the other. Oscillators like the relative strength index (RSI) and the stochastic oscillator are meant to show when a certain stock is overbought or oversold by comparing the previous n-period highs and lows with the current closing price. Some measurements of intraday volatility and overnight returns are also included specifically because half of the models predict intraday returns.

Table 4.2: TTR feature space

Feature name	Description	(R) Formula
SMA5, SMA10, SMA20, SMA50, SMA200	Simple moving averages over different time frames	$movavg(close, m, type = "s")$ $m \in \{5, 10, 20, 50, 200\}$
SMAdifflong, SMAdiffmid, SMAdiffshort	The ratio of close price to a 20- and 50-day simple moving average and the ratio of a 50-day and 200-day simple moving average	Long = $\frac{SMA20}{SMA200}$ Mid = $\frac{Close}{SMA20}$ Short = $\frac{Close}{SMA5}$
EMA5, EMA10, EMA20, EMA50, EMA200	Exponential moving averages over different time frames	$movavg(close, m, type = "e")$ $m \in \{5, 10, 20, 50, 200\}$
EMAdifflong, EMAdiffmid, EMAdiffshort	The difference in two exponential moving averages	EMAdifflong = $\frac{EMA20}{EMA200}$ EMAdiffmid = $\frac{Close}{EMA20}$ EMAdiffshort = $\frac{Close}{EMA5}$

RSI10, RSI20, RSI50, RSI100	Relative strength index over different time periods, calculated using simple moving averages	$RSI(close, m, maType = \text{"SMA"})$ $m \in \{5, 20, 50, 100\}$
BIAS5, BIAS10, BIAS20	The relative deviation of a stock's current price to that of a moving average	$\frac{close - SMA(m)}{SMA(m)} * 100$ $m \in \{5, 10, 20\}$
eTRIX15, eTRIX40	The daily change in a triple exponential average.	$TRIX(close, n = m, nsig = 9, maType = \text{"EMA"})$ $m \in \{15, 40\}$
williamR, williamR20	A variant of Williams %R that measures daily price in relation to its previous highest high (HH) and lowest low (LL)	$\frac{HH_{t:t-m} - Close_t}{HH_{t:t-m} - LL_{t:t-m}}$ $m \in \{5,20\}$
ROC1, ROC10, ROC20	The rate of change in price over certain time periods	$\frac{close_t}{close_{t-m}}\ m \in \{1,10,20\}$
Disparity5	Shows the relative position of the current closing price to a 5-day simple moving average, expressed in percentage	$\frac{close}{SMA5}$
fastK	The %K of a stochastic oscillator, where the %K is the relationship between the current close price, the highest high (HH), and lowest low (LL) over a 20day period	$\frac{Close - LL_{t:t-20}}{HH_{t:t-20} - LL_{t:t-20}}$
fastD	The three-day moving average of the fastK	$\frac{fastK_t + fastK_{t-1} + fastK_{t-2}}{3}$
slowD	The three-day moving average of the fastD	$\frac{fastD_t + fastD_{t-1} + fastD_{t-2}}{3}$
CLV	The close location value; is the position of the current close price relative to the same-day high and low price. Also known as the "money flow multiplier"	$\frac{(Close - Low) - (High - Close)}{High - Low}$

OBV1, OBV20, OBV40, OBV60, OBV120, OBV240	The relative change in the running total of a stock's trading volume (On-Balance Volume) over a period of m	$OBV(Close, Volume)$ $$OBV_m = \frac{OBV_t - OBV_{t-m}}{OBV_{t-m}}$$ $m \in \{1,20,40,60,120,240\}$
OvernightMove_t	The simple overnight returns for a stock at time t	$\frac{Open_t - Close_{t-1}}{Close_{t-1}}$
Intrday_Voll	The logarithmic "return" of the high and low within a day	$\ln\left(\frac{High}{Low}\right)$
DeltaIntraDay	The one-day change in intraday volatility	$Intrday\_Voll_t - Intrday\_Voll_{t-1}$

Macro-factors

The macro-factor feature space consists of the market indexes OSEBX, OMX Nordic 40, S&P500, the Vanguard European Stock Index Fund, and the Chicago Board Options Exchange (CBOE) 's VIX index, the currency exchange rates between EUR-NOK and USD-NOK, the daily three-month Libor rate and daily European brent oil spot prices. Because the dataset has a daily frequency, factors with daily changes were chosen, and both features of day-to-day, as well as month-to-month changes were generated.

As the Norwegian stock exchange's benchmark, the OSEBX is included in the macro-factor feature space as a proxy for market trends in the Norwegian market.

OMX Nordic 40 is meant to capture the general market trend in Norway's neighboring markets. It is a market-weighted price index of the 40 largest and most actively traded stocks on the Nasdaq Nordic exchanges: Copenhagen, Stockholm, Helsinki, and Iceland. The index composition is revised twice a year, and the price is quoted in euros.

The S&P500 is the most recognized stock index on the planet and is a market-weighted index consisting of the 500 largest and most liquid stocks listed on the New York Stock Exchange (NYSE) and Nasdaq. As the S&P500 is followed by most investors worldwide and is made up of some of the largest companies in the world, it is intended to serve as a proxy for international market trends.

Vanguard's European Stock Index Fund (VUESX) aims to track the FTSE Developed Europe All Cap index, which is a market-cap weighted index representing stocks from 16 European markets (including the UK). The fund contains assets in excess of 1200 companies across Europe. VUESX is therefore chosen as a proxy for the market trends in European markets.

The volatility index (VIX) by the Chicago Board Options Exchange is an index that aims to reflect market expectations of future volatility. Using prices from options on the S&P500 index that expire in 23-37 days, it gives a quantitative measurement for market sentiment about the 30-day forward period.

LIBOR, or London interbank offered rate, is a daily benchmark for interest rates at which global banks are willing to lend to one another, given for five different currencies: US dollar, Swiss franc, euros, British pound sterling, and Japanese yen. It is administered and published by the Intercontinental Exchange (ICE), based on transaction data, transaction-based data or expert judgment based on availability. As a benchmark rate, LIBOR is assumed to influence other, more local, interest rates. In this thesis, the US dollar Libor is included as a feature to represent interest levels.

We use the prices for the indexes to calculate log returns for a given time period. Intraday investing will use the daily returns for index i, and monthly investing will use the monthly returns for index i. Index returns for both investing periods are given in (23) and (24):

$$index_t^i = log(\frac{P_t^i}{P_{t-1}^i}) \tag{23}$$

$$index20_t^i = log(\frac{P_t^i}{P_{t-20}^i}) \tag{24}$$

The other macro-economic features will use the same feature generation as the indexes, which will give the following equations:

$$lnLIBOR_{t,1} = log(\frac{L_t}{L_{t-1}}) \tag{25}$$

$$lnBrentOil_{t,1} = log(\frac{Oil_t}{Oil_{t-1}}) \tag{26}$$

$$lnUSD\_NOK_{t,1} = log(\frac{USDNOK_t}{USDNOK_{t-1}}) \tag{27}$$

$$lnEUR\_NOK_{t,1} = log(\frac{EURNOK_t}{EURNOK_{t-1}}) \tag{28}$$

Monthly macro-economic features will therefore be denoted as:

$$lnLIBOR20_t = log(\frac{L_t}{L_{t-20}}) \tag{29}$$

$$lnBrentOil20_t = log(\frac{Oil_t}{Oil_{t-20}}) \tag{30}$$

$$lnUSD\_NOK20_t = log(\frac{USDNOK_t}{USDNOK_{t-20}}) \tag{31}$$

$$lnEUR\_NOK20_t = log(\frac{EURNOK_t}{EURNOK_{t-20}}) \tag{32}$$

4.7 Transaction cost

In order to estimate realistic portfolio returns, transaction costs are subtracted from portfolio returns. The portfolio returns before transaction costs can be denoted as:

$$R_t^p = \sum_{n=0}^{n} w_t^s * r_t^s \tag{33}$$

Where:

R_t^p are the returns of portfolio p in period t.

n is the number of stocks chosen in each period.

w_t^s is the weight of stock s in period t.

r_t^s are the returns of stock s in period t.

When the transactional costs are added, the portfolio can be denoted as:

$$R_t^p = \sum_{n=0}^{n} w_t^s * (r_t^s - tc * 2) \tag{34}$$

Where tc is the transaction costs, which is multiplied by two to show that the transactional costs are added when buying and selling an asset.

Transaction costs are based on the costs of trading using Nordnet, as it is one of Scandinavia's most popular stock brokers. As all assets are “bought” at the start of the period and “sold” at the end of the period, a total of 20 transactions are assumed to happen every month.

Transactional costs for the monthly portfolios will be set to 0,04%, which is gathered from Nordnet (Nordnet, n.d.), given the number of trades being made each month. Since the intraday portfolios will make more trades than the monthly portfolios, the transaction costs for the intraday portfolios are set as 0,035%, given the price listing on Nordnet (Nordnet, n.d.).

4.8 Model specifications

The models will be used to create a monthly and an intraday portfolio. Since the feature space generated is so large, the features will be split across the monthly and intraday strategies. Monthly models will use the Tan-, 20-day macro features, and the TTR features. The intraday models will use the volume and volatility features of Tan, momentum features from Krauss, daily macro features, and all the TTR features. All models have been run using the program R.

4.8.1 Random Forest

As explained in section 2.1, a random forest must have a depth, be bagged, and decide how many features that are applied to each split. When modeling the random forest model, we can decide upon these variables. In order to run the random forest model, we have used the package "randomForest".

For j_{rf} we have left it to the default value for the model, which ranges from $1 \leftrightarrow \infty$.

We have decided to run $n_{tree} = 500$. Previously, Tan et al. (2019) tested the random forest model with 20-120 trees using 20 tree-increments. They found that 60 trees performed best in their case, but that more trees led to greater accuracy for the in-sample training. However, it has been shown that it is hard to overfit a random forest (James et al., 2021, s.340-341). Therefore, we have opted to use a greater number of trees since it would make the model more concise when making decisions throughout the period.

Features per split have been set to the default value $m_{try} = \sqrt{p}$, as well since there does not seem to be any increase in performance by changing $m_{try} \neq \sqrt{p}$. This would give our monthly model an $m_{try} \cong 10$, and our intraday model an $m_{try} \cong 9$.

4.8.2 Support Vector Machine

In a radial kernel SVM, the hyperparameters to be adjusted are the cost C and gamma γ. Additionally, the weighting of the classes can be adjusted in any SVM. The SVMs were constructed using the svm() function in the "e1071"- package for R. For the svm() function, the default value of γ is $\frac{1}{data\ dimension}$ and the default value of C is 1. Altering these parameters manually showed no significant improvement in prediction performance when tested on a smaller subset of stocks. Therefore, the values have been kept at their default settings. This means that, with 102 input features for the intraday model and 93 features for the monthly model, the specifications for the SVMs constructed in this paper are a C of 1 and γ's of $\frac{1}{102} \cong 0{,}0098$ and $\frac{1}{79} \cong 0{,}0127$ for monthly and intraday models, respectively. Using above and below median returns as the class separation criteria, the two classes should be symmetric. The class weights were therefore left equal.

Ideally, evaluation of performance in the training period using cross-validation and a grid search could be utilized to optimize the parameters for the SVM based on a target criterion, such as accuracy or AUC-score. The SVM is a computationally demanding model, however, and estimating a total of 218 SVMs without cross-validation and grid search already demands a lot of computation time. Given faster computation, both cross-validation and recursive feature selection could have been done for each period, and the results are hypothesized to have improved as a result. We regard this as a limitation in the study design that could significantly impact the conclusions of the paper and as a suggestion for future research opportunities.

4.8.3 Logistic Regression

The Logistic Regression model has been fitted with the calculated features as the x variables and the classification feature as the y variable. When a Logistic Regression is fitted, the results may vary each time the model is fitted. To reduce variation, a gradient-boosted logistic regression is chosen. This entails methods like the ones of the boosting in random forest models and will be denoted as n_{tree}.

The Logistic Regression model will have $n_{tree} = 500$. When performing the Logistic Regression, we have used the package "gbm" in R.

5 Results

5.1 Model performance

A classification model with a binary classification feature can have four different outcomes, often displayed in a confusion matrix. These outcomes are *true positive* (TP), *true negative* (TN), *false positive* (FP), and *false negative* (FN). True positives and -negatives are instances where a classification model is able to correctly predict a positive or negative outcome, whereas false positives and -negatives are instances where a classification model incorrectly predicts either a positive or negative. An illustration of a confusion matrix is given in Figure 5.1, and the confusion matrices for our models are provided in appendix A.

		True Class	
		Negative	Positive
Predicted class	Negative	TN	FN
	Positive	FP	TP

Figure 5.1: Illustration of a confusion matrix

Some important performance measurements can be calculated from the confusion matrix. For this thesis, the performance measurements accuracy, recall, precision, F1-score, specificity, and negative predicted value (NPV) are used for model comparison. Accuracy refers to the total amount of times a model is able to predict a true outcome and is the sum of true positives and true negatives divided by the total amount of observations the model is trying to classify. Recall is the true positives divided by all positives and shows the number of positive outcomes that the model is able to detect. Precision shows the percentage of instances when a positive prediction is correct and can be calculated by dividing the true positives by all positive predictions. F1-score is an alternative measurement to accuracy that provides a better estimation of model performance when class sizes are not equal. It is calculated as the geometric mean of precision and recall. Specificity can be seen as the recall equivalent for negative observations and shows how many of the total negative observations that the model is able to detect. NPV is the equivalent of precision for the negative observations and shows how many of the total negative prediction is actually correct.

Table 5.1: Model evaluation metrics

Evaluation metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Recall	$\frac{TP}{TP + FN}$
Precision	$\frac{TP}{TP + FP}$
F1-score	$2 * \frac{Recall\ *\ Precision}{Recall + Precision}$
Specificity	$\frac{TN}{TN + FP}$
Negative Prediction Value	$\frac{TN}{TN + FN}$

Another useful evaluation measurement is the area under the receiving operating characteristic (ROC) curve, called AUC. The ROC curve is a graph that shows the performance of a classification on different classification thresholds based on percentages. It is made by plotting the true positive rate (recall) against the false positive rate $\frac{FP}{FP+TN}$ for different prediction thresholds. The AUC can be interpreted as the probability that a model will rank a positive observation above a negative observation when the probability of a positive observation is higher. For example, a score of 1, or 100%, would mean a model that can perfectly separate all positives and negatives into their respective classes, a model with an AUC of 0.5, or 50%, would be a model where the classification of the observations is random, and a model with an AUC of 0 would be a model where classes are predicted completely opposite of their true classes.

Table 5.2: Performance of monthly- and intraday models

	SVM (monthly)	SVM (intraday)	RF (monthly)	RF (intraday)	LR (monthly)	LR (intraday)
Accuracy	0,5268	**0,5381**	0,5229	0,5342	0,5271	0,5356
Recall	0,5634	0,3082	**0,6160**	0,3964	0,6094	0,3721
Precision	0,5413	0,5203	0,5340	0,5102	0,5382	0,5130
F1-Score	0,5521	0,3871	**0,5721**	0,4462	0,5716	0,4313
Specificity	0,4874	**0,7447**	0,4230	0,6581	0,4387	0,6825
NPV	0,5098	0,5450	0,5064	**0,5482**	0,5113	0,5475
AUC	0,5350	0,5417	0,5274	0,5420	0,5320	**0,5438**

Table 5.2 shows the performance of the models using the selected evaluation criteria, where the highest value of each criterion is marked in bold. From recall and specificity numbers, it is apparent that the monthly models are more "positive" in their predictions than their intraday counterparts. There is no model that singles out as the best model, however. The intraday LR has the highest AUC value, while the intraday SVM has the highest accuracy, and the monthly RF has the best F1-score. Generally, intraday models have slightly higher accuracies and AUC-scores than monthly models, while the monthly models have significantly higher F1-scores due to the low recall values of the intraday models.

While a higher accuracy is intuitively desirable, other metrics can be equally or more important if the cost of a false negative and a false positive is different. For example, in a portfolio that only enters long positions, a false positive will likely be more detrimental than a false negative. The false positive will reduce the portfolio value while a false negative simply causes the portfolio to not increase in value. In "long-only" portfolios, model precision could be among the more important measurements. Despite having significantly higher recall, the monthly models also have higher precision than the intraday models. Typically, a higher recall means a softer predictor that predicts positives more often at the cost of precision. With both higher recall and precision, the monthly models are expected to perform better than their intraday counterparts.

5.2 Feature importance

Figures B.1-B.12 in appendix B show the mean importance of the different input features for each model. All three models calculate importance differently, so the actual values are not comparable across models. Feature importance is therefore shown as the relative importance of each feature to the most important feature in every model. Due to the large number of inputs, the figures are large and therefore reserved for the appendix.

The distribution of feature importance in RF and the LR models is larger for the monthly portfolios than for intraday, while the opposite is true for the SVM. Generally, the RF models have a smaller distribution of feature importance, with both models having more than half of the features above a relative importance of 0,5, and all features above a relative importance of 0,5 for intraday models. Meanwhile, LR has a larger distribution for monthly and intraday models, with only four features above a 0,5 relative importance for the monthly model and two features above a 0,5 relative importance intraday. The SVM functions as a sort of in-between with around the top quartile of feature importance above 0,5 and around the bottom quartile under 0,25 for both intraday and monthly models. Distribution of feature importance within the models indicate that some form of feature

For individual feature categories, mid to long-term on-balance volumes are important for all models. On-balance volume features are the five most important features for the intraday SVMs, while some on-balance volume features are among the eight most important features for all models. The intraday SVM also ranks all pure momentum features highly, as well as the ratios of closing price and short- and medium-term moving averages. Momentum is also important for the monthly SVM, both regarding pure momentum, such as *close_19* and *close_39*, the reversal of momentum, such as *close19_0* and *close39_0*, and rate-of-change features. Deviations from moving averages are also important for the monthly SVM, as *BIAS20*, *SMAdiffmid*, *EMAdiffmid*, and *BIAS10* are the four highest-ranked features.

Moving averages are also important for the monthly LR and RF models, though more the moving averages themselves than their relationship with closing price. Both models also find return volatility important. For example, the *vol120* feature is the most important feature for both models. Another shared feature type for the two models is the standard deviations of trading volumes. The feature *volumestd120* is the fourth highest ranked feature in the LR model and the second highest ranked feature in the RF model. Interestingly, none of the monthly models find any of the macro factors particularly important. The highest ranking of a macro feature in any of the three monthly models is *lnDeltaLibor20* at 31st in the RF model, and it is the only macro feature ranked in the top half of any model.

The macro features are significantly more important for the intraday RF, however. While the feature importance distribution is generally low for the intraday RF, all nine macro factors rank among the top 15 features, with *lnOil* ranking highest at the top three and *lnVanguard* ranking lowest at the 15th. Macro features also rank higher for the intraday LR model than its monthly counterpart, but are somewhat more dispersed among the top half, while the intraday SVM ranks all macro features in the bottom half. Similarly to the monthly models, the RF and LR models share their most important feature: *OvernightMove_t*. Unlike the other models, the intraday LR model has a mix of all the different features amongst the most important features.

5.3 Portfolio construction

Portfolios are constructed based on the likelihood that an observation will give higher returns than the median return over the next period. The ten most likely "winners", based on the model's prediction, are equally weighted in the portfolio. Equally weighted portfolios are chosen as they outperform mean-variance in many instances when a mean-variance approach is applied to out-of-sample data (DeMiguel et al., 2009). An additional inclusion criterion is that the probability of being a winner is above 50%. In other words, a stock must be more likely to become a winner than a loser to be included in the portfolio. Additionally, two ensemble portfolios are made using the simple averages of the probability output of the three different machine-learning types. Sometimes the models predict that none or fewer than ten stocks will be classed as winners. In such instances, portfolios with fewer than ten stocks are constructed. If two or more stocks have been assigned the same probability of becoming a winner and this causes more than ten stocks to be eligible for inclusion, the portfolio has been allowed to exceed ten stocks.

All stocks are assumed to be bought at the open price at the start of the period and sold to close price at the end of the period. Returns for the portfolios are only based on the mean return of the n number of stocks held over the chosen period. Practically this means that all available funds would be allocated equally among the stocks in the portfolio, regardless of if the portfolio consists of 1 stock or ten stocks. We note that when the portfolio has a full investment in fewer than ten stocks, the diversification effect is reduced and removed in some cases. However, since this paper aims to see if machine-learning models can select winner stocks, we will allow these investments to be made without adjustments.

5.4 Returns Monthly

Following the method described in the section above, we will present the monthly portfolios. Figure 5.2 shows the cumulative returns before transaction costs for the monthly portfolios and the monthly returns for the OSEBX. The figure shows that the machine-learning portfolios perform well during the first four years of the sample period, before stagnating somewhat from the beginning of 2017. All portfolios are impacted by the financial downturn beginning in the last quarter of 2018 before rebounding a year later, meaning that all machine-learning portfolios are valued higher than an investment in OSEBX would be before the Covid-19 pandemic began in 2020. After 2020, the machine learning models seem unable to predict future monthly returns accurately. The RF portfolio, which is the highest performing monthly portfolio, rebounds quickly after the initial pandemic crash with three months of good predictions but fails to generate consistent positive returns for the following 26 months. All machine-learning portfolios end below the OSEBX as a result.

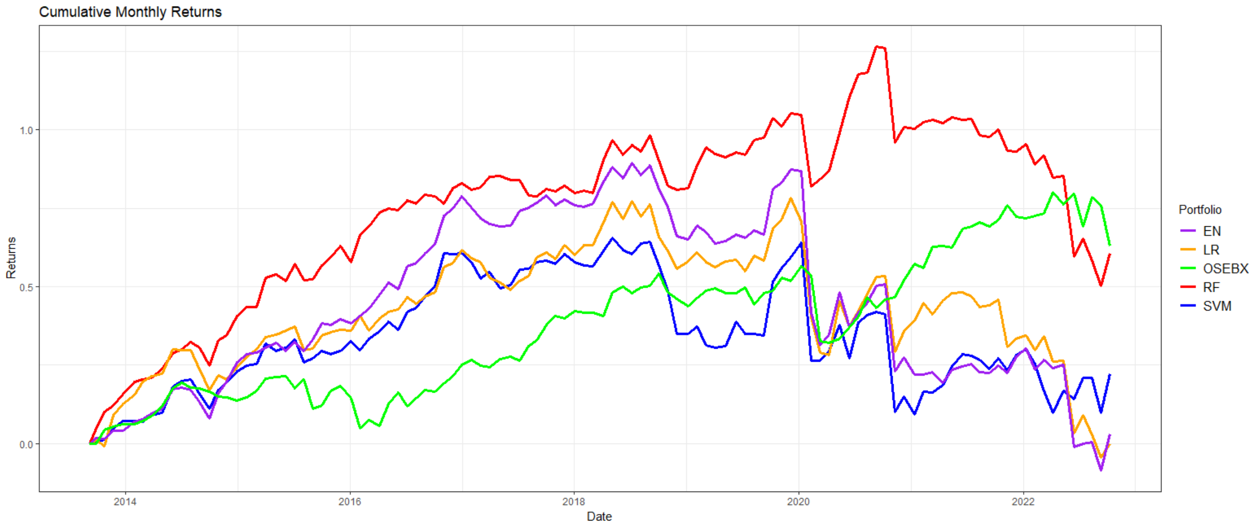

Figure 5.2: Monthly cumulative returns before transaction costs

Table 5.3 shows annual returns, volatility, and the Sharpe ratio for the full period without transaction costs. As illustrated in Figure 5.2, no machine-learning portfolio is able to outperform the OSEBX. Even before transaction costs are considered, the machine-learning portfolios generate a lower annualized return at a higher risk than the market. Among the machine-learning portfolios, the RF model portfolio performs the best, with both higher returns and lower volatility than its machine-learning counterparts. The logistic regression portfolio performs the worst, generating an annual return of just 0,0016% at an annual volatility of 22,63%.

Table 5.3: Monthly portfolio performance without transaction costs

	Full Period				
	SVM	RF	LR	EN	OSEBX
Annualized Returns	2,44 %	6,69 %	0,0016 %	0,33 %	7,49 %
Annualized Volatility	23,47 %	21,53 %	22,63 %	24,95 %	17,20 %
Sharpe	0,10	0,31	0,00	0,01	0,44

As is apparent from Figure 5.2, the reason machine-learning portfolios underperform the OSEBX is in large part due to the poor performance after 2020. Table 5.4 therefore reports the performance of the portfolios in subsets of the sample timeframe as well as the full period after subtracting the approximated transactional costs.

All machine-learning portfolios generate higher returns and lower volatility than the OSEBX for the 26.09.2013 - 31.12.2015 period. The RF portfolio performs particularly well in this period, with a Sharpe ratio of 2,25 and annual returns of 27,08%. Machine-learning portfolios are also less volatile than the market in the first four years of the sample period, and most portfolios outperform the market during these four years, with the exception being the RF portfolio which has a slightly lower Sharpe ratio over the 01.01.2016 - 31.12.2017 period.

Among the machine-learning portfolios the RF portfolio tends to perform better in periods of market decline. It had the lowest volatility and highest Sharpe ratio during both the market downturn in 2018 and after the pandemic hit at the start of 2020. As shown in Figure 5.2, it suffers the smallest initial losses at the start of these periods and quickly rebounds in the following months. In general, the results from the decomposed time periods are in accordance with the previous findings of Krauss et al. (2017) and Kilskar (2020), that machine-learning portfolios perform worse in the later time periods.

Table 5.4: Monthly portfolio performance after transaction costs

Monthly portfolios after transaction costs

Full period

	SVM	RF	LR	EN	OSEBX
Annualized Returns	1,49 %	5,73 %	-0,96 %	-0,63 %	7,49 %
Annualized Volatility	23,47 %	21,53 %	22,63 %	24,95 %	17,20 %
Sharpe	0,06	0,27	-0,04	-0,03	0,44

26.09.2013 - 31-12.2015

	SVM	RF	LR	EN	OSEBX
Annualized Returns	12,15 %	27,08 %	15,19 %	16,60 %	7,10 %
Annualized Volatility	11,34 %	12,03 %	13,47 %	10,55 %	15,55 %
Sharpe	1,07	2,25	1,13	1,57	0,46

01.01.2016 - 31.12.2017

	SVM	RF	LR	EN	OSEBX
Annualized Returns	14,45 %	8,63 %	12,50 %	18,16 %	13,04 %
Annualized Volatility	12,29 %	10,85 %	11,53 %	11,07 %	15,96 %
Sharpe	1,18	0,79	1,08	1,64	0,82

01.01.2018 - 31.12.2019

	SVM	RF	LR	EN	OSEBX
Annualized Returns	-1,46 %	10,52 %	6,50 %	3,87 %	6,85 %
Annualized Volatility	20,86 %	16,34 %	17,55 %	17,62 %	13,88 %
Sharpe	-0,07	0,64	0,37	0,22	0,49

01.01.2020 - 31.12.2020

	SVM	RF	LR	EN	OSEBX
Annualized Returns	-45,14 %	-5,23 %	-43,48 %	-60,93 %	-0,69 %
Annualized Volatility	53,18 %	45,03 %	46,89 %	57,38 %	27,16 %
Sharpe	-0,85	-0,12	-0,93	-1,06	-0,03

01.01.2021 - 15.11.2022

	SVM	RF	LR	EN	OSEBX
Annualized Returns	2,90 %	-22,88 %	-20,39 %	-14,32 %	7,08 %
Annualized Volatility	20,42 %	24,32 %	24,61 %	24,64 %	16,74 %
Sharpe	0,14	-0,94	-0,83	-0,58	0,42

5.5 Returns Intraday

Quantitative trading is often associated with high trading frequencies, and as such, a daily trading strategy is tested for machine-learning portfolios. Figure 5.3 shows the cumulative returns before transaction costs for the intraday machine-learning portfolios as well as the market index. A similar pattern to the monthly portfolios can be observed. The intraday portfolios differ in some respects, however. They are not as highly impacted by the market decline in late 2018 and contrary to the monthly portfolios, they show positive returns after the initial market drop in early 2020. A probable explanation for this is that while monthly and daily models are trained on and predict the same number of observations, only the observations in the beginning of the 20-day testing periods are used to create the monthly portfolios. Without any form of stop-loss as part of the portfolio strategy, monthly portfolios commit to a prediction longer than any of the intraday portfolios do. Developments past the initial day of a testing period will not have any effect on a monthly portfolio's composition.

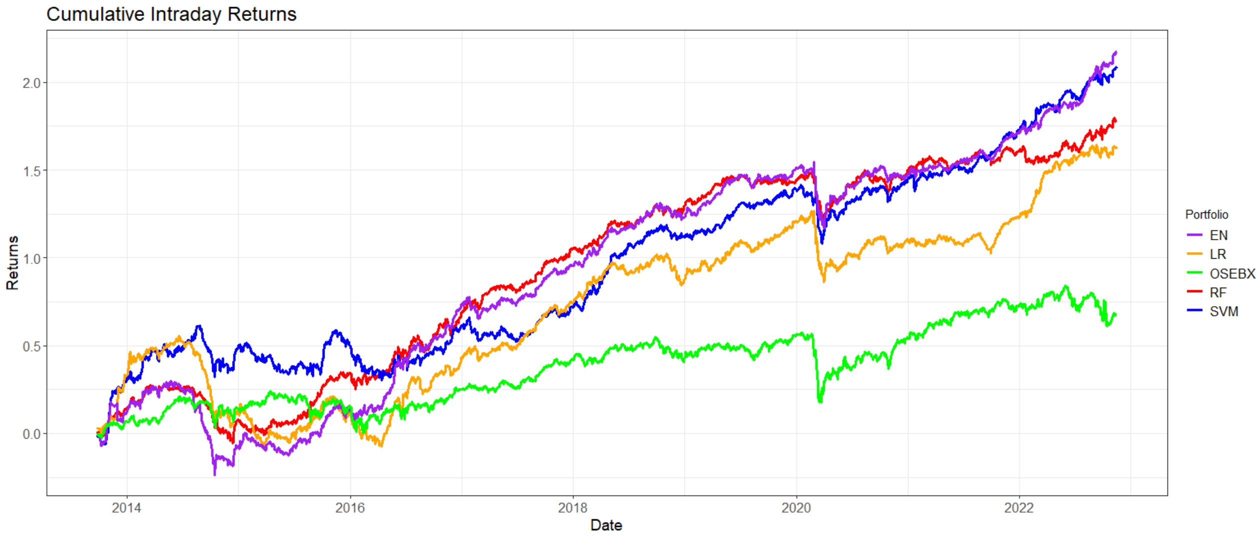

Figure 5.3: Intraday cumulative returns before transaction costs

Table 5.5 reports annual returns, annual volatility, and Sharpe ratio for all the intraday machine-learning portfolios without transaction costs compared with the market index. Without transaction cost, all intraday portfolios yield high annualized returns, and all portfolios outperform the OSEBX. The highest performer is the EN portfolio, with the highest return and Sharpe ratio. All portfolios except for the RF portfolio are more volatile than the market, however. The LR portfolio is again the worst performer among the machine-learning portfolios, with the lowest return and the second-highest annual volatility. Portfolios based on the SVM and RF perform similarly for the full period, but with different characteristics. Both have Sharpe ratios of 1,25, but the SVM yields higher returns and is more volatile than the random forest portfolio.

Table 5.5: Intraday portfolio performance before transaction costs

Full Period

	SVM	RF	LR	EN	OSEBX
Annualized Returns	22,87 %	19,51 %	17,85 %	23,92 %	7,49 %
Annualized Volatility	18,32 %	15,60 %	18,02 %	17,91 %	17,20 %
Sharpe	1,25	1,25	0,99	1,34	0,44

As with the monthly portfolios, intraday portfolio performance is separated into subsections and reported after transaction costs. Table 5.6 shows the portfolio performance for the intraday strategy after considering transaction costs. As expected, transaction costs greatly impact the portfolio return when trading daily compared to monthly portfolio rebalancing. Full-period returns show that the machine-learning portfolios made positive returns, but that all were outperformed by the OSEBX.

Decomposing the intraday portfolio returns after transaction costs into the same time periods as in Table 5.6 shows no systematic pattern like the monthly holding periods. No portfolio outperforms the market in the early 26.09.2013 - 31.12.2015 period or the pandemic year of 2020. However, at least two of the three machine-learning portfolios outperform the market in the remaining periods. The EN strategy does particularly well for the intraday portfolios. It is the highest performer for two of the periods, and the only portfolio that outperforms the market in all three positive periods. Interestingly, one portfolio based on an individual predictor generates negative returns for both periods where the EN performs best, while the SVM portfolio performs better during the 01.01.2018 - 31.12.2019 period, where all individual predictor portfolios generate positive returns. The SVM also performs better than both the EN portfolio and the RF portfolio during periods where no machine-learning portfolios were able to beat the market, despite the RF portfolio having the lowest volatility during these periods. Contrary to the monthly portfolios, the SVM, LR and EN intraday portfolios also generate high returns after transaction costs for the period after Covid-19.

Table 5.6: Intraday portfolio performance after transaction costs

Intraday portfolios after transaction costs

Full period

	SVM	RF	LR	EN	OSEBX
Annualized Returns	5,38 %	1,87 %	0,31 %	6,42 %	7,49 %
Annualized Volatility	18,32 %	15,60 %	18,02 %	17,91 %	17,20 %
Sharpe	0,29	0,12	0,02	0,36	0,44

26.09.2013 - 31-12.2015

	SVM	RF	LR	EN	OSEBX
Annualized Returns	5,06 %	-2,25 %	-11,23 %	-10,97 %	7,10 %
Annualized Volatility	22,95 %	15,82 %	21,95 %	22,60 %	15,55 %
Sharpe	0,22	-0,14	-0,51	-0,49	0,46

01.01.2016 - 31.12.2017

	SVM	RF	LR	EN	OSEBX
Annualized Returns	-6,78 %	17,62 %	12,91 %	23,23 %	13,04 %
Annualized Volatility	18,35 %	16,14 %	16,76 %	16,85 %	15,96 %
Sharpe	-0,37	1,09	0,77	1,38	0,82

01.01.2018 - 31.12.2019

	SVM	RF	LR	EN	OSEBX
Annualized Returns	16,40 %	2,11 %	6,03 %	10,35 %	6,85 %
Annualized Volatility	13,59 %	13,17 %	13,82 %	13,14 %	13,88 %
Sharpe	1,21	0,16	0,44	0,79	0,49

01.01.2020 - 31.12.2020

	SVM	RF	LR	EN	OSEBX
Annualized Returns	-12,02 %	-12,11 %	-29,02 %	-21,86 %	-0,69 %
Annualized Volatility	20,70 %	18,73 %	23,27 %	22,65 %	27,16 %
Sharpe	-0,58	-0,65	-1,25	-0,96	-0,03

01.01.2021 - 15.11.2022

	SVM	RF	LR	EN	OSEBX
Annualized Returns	16,46 %	-2,80 %	10,38 %	20,32 %	7,08 %
Annualized Volatility	14,64 %	15,26 %	14,37 %	13,51 %	16,74 %
Sharpe	1,12	-0,18	0,72	1,50	0,42

5.6 Stock selection

Table 5.7 shows the 15 most held stocks throughout the period for each of the monthly machine-learning portfolios. The table reports how many periods a stock was held and the average returns before transaction costs that stock had for the holding periods. Transaction costs are excluded to show how well the machine-learning portfolios pick winner stocks.

There are some overlaps for the most traded stocks among the four monthly portfolios. Since the EN portfolio is created using an average for the probability of the other portfolios, it would be expected that the EN portfolio shares the same stock selection as the other portfolios. Interestingly, the returns for overlapping stocks between the portfolios do not mean that the stock provides the same returns to the portfolio. For instance, Schibsted ser. A is traded amongst the top 15 stocks for all the portfolios. However, they provide negative returns for the EN and RF portfolios and positive returns for the LR and SVM portfolios. Although the machine-learning portfolios share many of the most traded stocks, it is evident that they do not share the same timing for when to add the stocks into the portfolio. Overlapping assets might therefore perform entirely differently between the four portfolios.

Table 5.7: The 15 most held stocks for monthly portfolios

Support Vector Machine			Random Forest		
Name	Frequency	Returns	Name	Frequency	Returns
ORKLA	23	0,0104	VEIDEKKE	27	0,0150
TELENOR	21	0,0060	MOWI	25	0,0092
MOWI	19	0,0136	SPAREBANK 1 SR-BK	25	0,0141
TOMRA SYSTEMS	19	0,0182	ATEA	23	-0,0016
GJENSIDIGE FORSIKR	18	0,0106	BOUVET	20	-0,0692
AF GRUPPEN	17	0,0143	GJENSIDIGE FORSIKR	20	0,0050
HAFSLUND SER. B	17	0,0359	ORKLA	20	0,0127
SCHIBSTED SER. A	15	0,0085	LERØY SEAFOOD GP	19	-0,0125
ARENDALS FOSSEKOMP	14	0,0300	NORWAY ROYALSALMON	19	0,0135
BOUVET	14	-0,0817	TOMRA SYSTEMS	19	0,0279
EQUINOR	14	0,0135	AUSTEVOLL SEAFOOD	17	0,0102
OLAV THON EIENDOMS	14	0,0172	ENTRA	17	0,0106
YARA INTERNATIONAL	14	0,0001	SCHIBSTED SER. A	17	-0,0053
BAKKAFROST	13	0,0225	SCHIBSTED SER. B	17	0,0063
FLEX LNG	13	0,0190	SALMAR	15	0,0247

Logistic Regression			Ensemble		
Name	Frequency	Returns	Name	Frequency	Returns
EQUINOR	28	0,0028	VEIDEKKE	25	0,0182
PIONEER PROPERTY	22	0,0065	ORKLA	24	0,0129
MOWI	21	0,0138	TELENOR	20	0,0000
ORKLA	21	0,0062	TOMRA SYSTEMS	20	0,0239
YARA INTERNATIONAL	20	-0,0043	BOUVET	19	-0,0831
BOUVET	19	-0,0918	EQUINOR	19	0,0122
SCHIBSTED SER. A	19	0,0105	MOWI	19	0,0119
VEIDEKKE	18	0,0203	AF GRUPPEN	17	0,0241
AKER	17	-0,0133	EUROPRIS	17	0,0014
HAFSLUND SER. A	17	0,0352	SCHIBSTED SER. A	17	-0,0045
SPAREBANK 1 SR-BK	16	-0,0046	ATEA	16	-0,0090
AF GRUPPEN	15	0,0123	HAFSLUND SER. B	16	0,0308
ARENDALS FOSSEKOMP	15	0,0196	LERØY SEAFOOD GP	16	-0,0041
NORWEGIAN PROPERTY	15	0,0178	ENTRA	15	0,0159
GOODTECH	14	-0,0216	GJENSIDIGE FORSIKR	15	0,0101

Table 5.8 show the 15 most frequently held stocks for the intraday portfolios. In addition, the table shows the periods each stock is held and the returns they provide to the portfolio before transaction costs.

Like with the monthly portfolios, there is an overlap of stocks across the portfolios. Returns for a stock are not the same across the portfolios, meaning that the intraday portfolios also do not share the same timing.

Both the monthly and intraday portfolios share the same stocks among the 15 most traded stocks. Indicating that the different machine-learning models create portfolios which tend to pick the same stocks, albeit at different points in time. Seeing as the models use the same feature space, it could explain why there are similar trades. However, the difference in the models makes it so that the portfolios are not 1:1 in terms of trades.

Table 5.8: The 15 most held stocks for intraday portfolios

Support Vector Machine		
Name	Frequency	Returns
TELENOR	789	-0,0007
YARA INTERNATIONAL	774	-0,0002
EQUINOR	770	0,0007
MOWI	768	0,0015
DNB	750	-0,0004
NORSK HYDRO	485	0,0000
GJENSIDIGE FORSIKR	429	0,0004
SCHIBSTED SER. A	382	0,0002
SUBSEA 7	382	0,0008
ORKLA	377	0,0008
SALMAR	358	0,0003
AKER BP	327	0,0010
BAKKAFROST	310	0,0009
NORWEGIAN ENERGY	277	0,0049
STOREBRAND	276	0,0006

Random Forest		
Name	Frequency	Returns
GJENSIDIGE FORSIKR	648	0,0002
MOWI	633	0,0003
ORKLA	608	0,0004
YARA INTERNATIONAL	595	-0,0001
TELENOR	517	-0,0010
EQUINOR	499	0,0017
DNB	483	0,0002
MAGNORA	450	0,0132
SALMAR	445	0,0006
SCHIBSTED SER. A	386	0,0004
STOREBRAND	359	0,0010
BAKKAFROST	354	0,0004
NORSK HYDRO	345	0,0004
TGS	344	0,0015
SPAREBANK 1 SR-BK	333	0,0001

Logistic Regression		
Name	Frequency	Returns
YARA INTERNATIONAL	767	-0,0003
EQUINOR	620	0,0016
MOWI	551	0,0013
DNB	527	-0,0002
TELENOR	516	-0,0002
SALMAR	482	0,0010
ORKLA	459	0,0005
MAGNORA	448	0,0101
GJENSIDIGE FORSIKR	438	0,0005
NORSK HYDRO	418	0,0002
SCHIBSTED SER. A	405	0,0006
TOMRA SYSTEMS	338	0,0009
TGS	320	-0,0005
PIONEER PROPERTY	294	0,0011
AKER BP	293	0,0007

Ensemble		
Name	Frequency	Returns
YARA INTERNATIONAL	840	-0,0003
MOWI	782	0,0006
EQUINOR	694	0,0014
TELENOR	681	-0,0004
DNB	653	-0,0009
GJENSIDIGE FORSIKR	620	0,0003
ORKLA	549	0,0010
SALMAR	495	0,0012
NORSK HYDRO	452	0,0004
SCHIBSTED SER. A	439	0,0011
MAGNORA	329	0,0113
BAKKAFROST	328	0,0008
TGS	322	0,0002
SUBSEA 7	321	0,0008
AKER BP	305	0,0020

5.7 Fama-French-Carhart Regression

As mentioned in section 2,4, the Fama-French-Carhart model is used to explain the return characteristics of the machine-learning portfolios. The factors have been collected from TITLON. Since access to the Fama-French-Carhart factors is only available for the OSE part of TITLON, we have limited the regression analysis to the period of 26th of September 2013 to 27th of November 2020.

Table 5.9 shows the results for the monthly and intraday portfolios before transaction costs. Interestingly, the monthly portfolios do not have any significant coefficients for the Fama-French-Carhart factors. In addition, the R^2 is low, indicating that the factors poorly explain the returns of the portfolios. Since there is a low R^2 as well as non-significant coefficients, the returns do not seem to follow any patterns in terms of investments. Neither portfolio report a significant α, despite the fact that transaction costs are excluded, and the analysis does not include dates after 27.11.2020.

The intraday portfolios however, do have some significant factors. Every intraday portfolio has a significant α, indicating that the portfolios produce returns unexplained by the other factors. While not presented in Table 5.9, none of the intraday portfolios have significant positive alphas after transaction costs are subtracted from returns. Interestingly, the SVM portfolio does not bias any of the factors, while the other three portfolios do bias some factors. Amongst the other three portfolios, all of them biases stocks which are underpriced. Additionally, the RF and LR portfolios biases small company stocks, while the EN portfolio shows some bias towards the market. Similarly to the monthly portfolios, the intraday portfolios have a low R^2, indicating that the factor model does not explain the portfolios' returns well.

Table 5.9: Fama-French-Carhart regression for monthly- and intraday portfolios

Fama-French-Carhart regression with monthly portfolios				
	SVM	RF	LR	EN
Intercept (α)	0,0047	0,0110	0,0068	0,0066
MKT	-0,1707	-0,0981	-0,1568	-0,0934
SMB	-0,0719	-0,2646	-0,2748	-0,2289
HML	0,1102	0,0460	-0,0297	0,0422
MOM	-0,1535	0,1223	-0,1729	-01689
R^2	0,0271	0,0507	0,0503	0,0389
Fama-French-Carhart regression with intraday portfolios				
	SVM	RF	LR	EN
Intercept (α)	0,00079**	0,00081***	0,00059*	0,00083**
MKT	0,03562	0,01170	0,03790	0,04774(.)
SMB	-0,04683	-0,01439*	-0,06580*	-0,02730
HML	-0,02718	-0,01931(.)	-0,03859(.)	-0,04355*
MOM	-0,02354	0,00101	-0,03473	-0,02731
R^2	0,00383	0,00085	0,00705	0,00510

***p < 0,001, **p < 0,01, *p < 0,05, (.) p < 0,1

6 Conclusion

The aim of this thesis is to examine if utilizing machine-learning for stock portfolio construction on the Oslo stock exchange can create portfolios that generate higher returns than the OSEBX. Three of the most common machine-learning models are used: Support Vector Machines, Random Forest, and Logistic Regression. Because multiple models are applied to financial data, the thesis has a secondary goal of comparing the machine-learning models.

Common classification model criteria suggest that the models are not especially good at classifying future stock placements, with general accuracies just above 50% for all models. Monthly model prediction seems to fit the construction of long-only portfolios better than intraday models, as they report both higher recall and precision than intraday models, and in turn achieve a significantly higher F1-score. Feature importance distributions for all but the intraday RF model suggest that some features have little explanation power and could potentially be removed to reduce noise.

Two portfolio strategies are created based on probability outputs from each machine-learning model. One portfolio strategy involves predicting 20-day future returns and has a holding period of 20 days, while the other strategy involves predicting next-day intraday returns and has a holding period from exchange open to exchange close on the same day. An ensemble portfolio for both strategies is also created based on a simple average of the three model probabilities. The portfolios consist of the ten stocks most likely to outperform the median future return based on the machine-learning model predictions, given a constraint that stocks included must be more likely to have above median returns than below (i.e.. the probability of being classified as above median is 50% or above).

There is an overlap among the 15 most traded stocks in the portfolios for the monthly and intraday strategies. Where they seem to pick the same stocks, however, they do not provide the same returns for each portfolio. The overlap and difference in returns can be explained by timing, where the different portfolios do not trade the stocks at the same time. It is however not surprising that the portfolios hold the same stocks, since the models are using the same feature space.

After transaction costs of 3,5- and 4 basis points are subtracted from the intraday and 20-day portfolios, respectively, all monthly portfolios outperform the OSEBX up until the start of 2020, while neither intraday portfolio outperforms the OSEBX over the same period. This aligns with previous research on applying monthly- and daily trading strategies created by machine-learning on the Norwegian stock market (Kilskar, 2020).

None of the models avoided the large market dip caused by the start of the Covid-19 pandemic. After 2020, monthly portfolios either stagnate or decline in value. This drastic performance dip during- and post Covid- 19 caused monthly portfolios to underperform compared to the index over the study period. Contrary to the monthly portfolios, SVM, LR and EN intraday portfolios showed promise in generating returns post 2020, as all portfolios were able to beat the market in this period.

Comparing our findings to the ones of Tan et al. (2019) and Krauss et al. (2017), there is some discrepancy between the returns from our portfolios and their portfolios. However, this might be due to their portfolios being run for a longer period, granting the portfolios a longer time to accumulate returns when compared to the benchmark index. Our paper also looks at an extended period for a smaller market than the US and Chinese markets. The machine-learning portfolios' performance decline over time could be explained by their more frequent use within the stock markets.

A Fama-French-Carhart four-factor regression on returns before transaction costs for the 29.06.2013 - 27.11.22 period shows no significant alpha for any of the monthly machine-learning portfolios, but did show significant alphas for all the intraday portfolios. Additionally, the intraday portfolios did show some bias towards underpriced stocks as well as smaller company stocks. In addition, the regressions are generally unable to explain any significant portion of the portfolio variances, with R2- values below 12% for all portfolios.

In summary, this thesis concludes that machine-learning techniques cannot be utilized to achieve consistently higher risk-adjusted returns than the general market on the Norwegian stock exchange. Despite the promising results in the period before 2020, monthly rebalanced machine-learning portfolios are not able to keep outperforming the OSEBX in later periods. Intraday portfolios do however show promise for later periods, even after transactional costs.

Due to inconsistent performance in earlier periods however, we are not able to conclude that these are a reliable source of outperformance in the Oslo stock exchange.

6.1 Limitations

When writing this thesis, we have had to consider limitations to our work. These limitations can be caused by the dataset, or simplifications of reality in order to maintain focus on the objective of the thesis.

Limitations that occur from the dataset is that accounting data is not available for periods after the end of 2020, as well as the lack of Fama-French-Carhart factors. Access to accounting data could improve the features used within the machine-learning models. The lack of Fama-French-Carhart factors after the end of 2020 means that testing which kinds of stocks are biased within the machine-learning portfolios will not be done after the end of 2020.

Further, the transactional costs are only regarded as direct costs, meaning that transaction costs related to bid-ask spread will not be considered. There is also an assumption that the portfolio investments are large enough to meet the minimum transaction costs.

Stocks are assumed to be purchased and sold to the open and close price. Hence there are assumed to be no liquidity issues.

This thesis only uses data with daily fluctuations. Accounting data and macro factors that are not updated daily are therefore not considered as features for our models. Many of the models in previous research make use of such features, especially accounting data to achieve good results. Other potentially relevant features such as NIBOR, inflation rate and unemployment rate are also excluded.

Computing power has also been a particularly significant limitation for this study. More powerful computing power would allow easier parameter tuning and more model variations.

6.2 Future work

When working on this paper, we have gotten many ideas on how to build upon our paper. In this section, we will give some possible extensions to our paper.

Since our paper has looked at different machine-learning models, one could pick one and try to optimize the feature space or do more thorough testing of hyperparameters. Suggestions would be recursive feature elimination and principal component analysis.

We have also discovered, through early testing, that the classification feature is important, and different classification methods yield different results when applied to the models. Therefore, we suggest comparing results from different classification features, such as the multi-class predictions of Tan et al. (2019), the median binary approach utilized in this thesis, an up-or-down classification variable, or a binary classification using cross-sectional quantiles of return, such as Zhang et al. (2018) or Yu et al. (2014). Pinelis and Ruppert (2022) also test an interesting regression approach where machine-learning is utilized to predict the future returns and volatility of a market index, and a mean-variance model is used to allocate resources between the market index and a risk-free asset.

We only applied equally weighted portfolios when creating our portfolios with the machine-learning models. Another possibility is the mean-variance portfolios or the minimum-variance portfolio, and the weights could even be adjusted by the probability given by the models. Combining modern portfolio theory with machine-learning models could improve their performance in later periods.

References

Baloglu, O., Latifi, S. Q., & Nazha, A. (2022). What is machine learning? *Archives of disease in childhood - Education & practice edition*, *107*(5), 386-388. https://doi.org/10.1136/archdischild-2020-319415

Banz, R. W. (1981). The relationship between return and market value of common stocks. *Journal of Financial Economics*, *9*(1), 3-18. https://doi.org/https://doi.org/10.1016/0304-405X(81)90018-0

Basu, S. (1983). The relationship between earnings' yield, market value and return for NYSE common stocks: Further evidence. *Journal of Financial Economics*, *12*(1), 129-156. https://doi.org/https://doi.org/10.1016/0304-405X(83)90031-4

Black, F. (1972). Capital market equilibrium with restricted borrowing. *The Journal of Business*, *45*(3), 444-455. http://www.jstor.org/stable/2351499

Bonga, W. G. (2015). The need for efficient investment: Fundamental analysis and technical analysis. *Social Science Research Network (SSRN)*(April 11, 2015). http://ssrn.com/abstract=2593315

Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance*, *47*(5), 1731-1764. https://doi.org/10.2307/2328994

Carhart, M. M. (1997). On persistence in mutual fund performance. *The Journal of Finance*, *52*(1), 57-82. https://doi.org/https://doi.org/10.1111/j.1540-6261.1997.tb03808.x

Coval, J. D., Hirshleifer, D., & Shumway, T. (2021). Can individual investors beat the market? *The Review of Asset Pricing Studies*, *11*(3), 552-579. https://doi.org/10.1093/rapstu/raab017

De Bondt, W., & Thaler, R. (1985). Does the stock market overreact? *The Journal of Finance*, *40*(3), 793-805. https://doi.org/10.1111/j.1540-6261.1985.tb05004.x

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, *22*(5), 1915-1953. https://doi.org/https://doi.org/10.1093/rfs/hhm075

EIA, U. S. E. I. A. (n.d.). *Europe brent spot price fob*. https://www.eia.gov/dnav/pet/hist/rbrteD.htm

Enke., D., & Thawornwong., S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems With Applications, 29,* 927-940. https://doi.org/10.1016/j.eswa.2005.06.024

Fama, E. F. (1965). The behavior of stock-market prices. *The Journal of Business, 38*(1), 34-105. http://www.jstor.org/stable/2350752

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383-417. https://doi.org/https://doi.org/10.2307/2325486

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3-56. https://doi.org/https://doi.org/10.1016/0304-405X(93)90023-5

Greenwell, B., Boehmke, B., & Cunningham, J. (2022). gmb: Generalized boosted regression models. https://CRAN.R-project.org/package=gbm

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning : with applications in R* (2nd ed.). 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. https://doi.org/10.2307/2328882

Jensen, M. C. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*, *6*(2), 95-101. https://doi.org/https://doi.org/10.1016/0304-405X(78)90025-9

Jordanoski, I., & Petrusheva, N. (2016). Comparative analysis between the fundamental and technical analysis of stocks. *Journal of Process Management and New Technologies*, *4*(2), 26-31. https://japmnt.com/archives/volume-4-issue-2-2016/

Kendall, M. G., & Hill, A. B. (1953). The analysis of economic time-series-part I: Prices. *Journal of the Royal Statistical Society. Series A (General)*, *116*(1), 11-34. https://doi.org/10.2307/2980947

Kilskar, S. (2020). *Aksjeutvelgelse ved bruk av Random Forest: en maskinlæringstilnærming til meravkastning* Nord Universitet]. https://hdl.handle.net/11250/2682978

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689-702. https://doi.org/https://doi.org/10.1016/j.ejor.2016.10.031

Kuhn, M. (2022). caret: Classification and regression training. https://CRAN.R-project.org/package=caret

Leung, M. T., Daouk, H., & Chen, A.-S. (2000). Forecasting stock indices: A comparison of classification and level

estimation models. *International Journal of Forecasting*, *16*, 173-190. https://doi.org/10.1016/S0169-2070(99)00048-5

Liew, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22. https://CRAN.R-project.org/doc/Rnews/

Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, *47*(1), 13-37. https://doi.org/10.2307/1924119

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems With Applications*, *165*, 113973. https://doi.org/https://doi.org/10.1016/j.eswa.2020.113973

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of Economics Perspectives*, *17*(1), 59-82. https://doi.org/10.1257/089533003321164958

Marketwatch. (n.d-a). *3 Month London Interbank Offered Rate in USD (LIBOR)*. https://www.marketwatch.com/investing/interestrate/liborusd3m?countrycode=mr&mod=search_symbol

Marketwatch. (n.d-b). *Euro/Norwegian Krone*. https://www.marketwatch.com/investing/currency/eurnok?mod=search_symbol

Marketwatch. (n.d-c). *USD/Norwgian Krone*. https://www.marketwatch.com/investing/currency/usdnok?mod=search_symbol

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, *7*(1), 77-91. https://doi.org/10.2307/2975974

Metghalchi, M., Marcucci, J., & Chang, Y.-H. (2012). Are moving average trading rules profitable? Evidence from the European stock markets. *Applied Economics*, *44*(12), 1539-1559. https://doi.org/10.1080/00036846.2010.543084

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2022). Misc functions of the Department of Statistics, Probability Theory Group. https://CRAN.R-project.org/package=e1071

Mossin, J. (1966). Equilibrium in a capital asset market. *Econometrica*, *34*(4), 768-783. https://doi.org/10.2307/1910098

Nasdaq Nordic. (n.d.). *OMXN40, OMX Nordic 40, (SE0001809476)*. https://www.nasdaqomxnordic.com/indexes/historical_prices?Instrument=SE0001809476

Nordnet. (n.d.). *Prisliste*. https://www.nordnet.no/no/kundeservice/prisliste

Park, C.-H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis? [https://doi.org/10.1111/j.1467-6419.2007.00519.x]. *Journal of Economic Surveys*, *21*(4), 786-826. https://doi.org/https://doi.org/10.1111/j.1467-6419.2007.00519.x

Pinelis, M., & Ruppert, D. (2022). Machine learning portfolio allocation. *The Journal of Finance and Data Science, 8*, 35-54. https://doi.org/https://doi.org/10.1016/j.jfds.2021.12.001

Rosenberg, B., Reid, K., & Lanstein, R. (1998). Streetwise

Persuasive Evidence of Market Inefficiency (Spring 1985). In P. L. Bernstein & F. J. Fabozzi (Eds.), *The Best of The Journal of Portfolio Management* (pp. 48-55). Princeton University Press. https://doi.org/doi:10.1515/9781400829408-007

Schwert, G. W. (2003). Chapter 15: Anomalies and market efficiency. In *Handbook of the Economics of Finance* (Vol. 1, pp. 939-974). Elsevier. https://doi.org/https://doi.org/10.1016/S1574-0102(03)01024-0

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk [https://doi.org/10.1111/j.1540-6261.1964.tb02865.x]. *The Journal of Finance*, *19*(3), 425-442. https://doi.org/https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Sharpe, W. F. (1966). Mutual Fund Performance. *The Journal of Business*, *39*(1), 119-138. http://www.jstor.org.mime.uit.no/stable/2351741

Sirnes, E. (n.d.). *TITLON*. https://titlon.uit.no/

Takeuchi, L., & Lee, Y.-Y. (2013). *Applying deep learning to enchance momentum trading strategies in stocks* [Working paper]. Standford University. https://cs229.stanford.edu/proj2013/TakeuchiLee-ApplyingDeepLearningToEnhanceMomentumTradingStrategiesInStocks.pdf

Tan, Z., Yan, Z., & Zhu, G. (2019). Stock selection with random forest: An exploitation of excess return in the Chinese stock market. *Heliyon, 5*(8), e02310. https://doi.org/https://doi.org/10.1016/j.heliyon.2019.e02310

Tobin, J. (1958). Liquidity preference as behavior towards risk. *The Review of Economic Studies*, *25*(2), 65-86. https://doi.org/10.2307/2296205

Yahoo Finance. (n.d.). *CBOE Volatility Index (^VIX)*. https://finance.yahoo.com/quote/%5EVIX/history/

Yu, H., Chen, R., & Zhang, G. (2014). A SVM Stock Selection Model within PCA. *Procedia Computer Science*, *31*, 406-412. https://doi.org/https://doi.org/10.1016/j.procs.2014.05.284

Zhang, R., Lin, Z.-a., Chen, S., Lin, Z., & Liang, X. (2018). Multi-factor stock selection model based on kernel support vector machine. *Journal of Mathematics Research*, *10*(5), 9-18. https://doi.org/10.5539/jmr.v10n5p9

iii. Appendix

Appendix A: Confusion matrices for the different models

Logistic Regression Monthly

		True Class: Negative	True Class: Positive
Predicted class	Negative	59179	55401
	Positive	77439	91248

Logistic Regression Intraday

		True class: Negative	True class: Positive
Predicted class	Negative	94404	80358
	Positive	53587	56421

Figure A.0.1: Confusion Matrices for LR Models

Random forest Monthly

		True Class: Negative	True Class: Positive
Predicted class	Negative	56540	55156
	Positive	80194	91377

Random forest intraday

		True Class: Negative	True Class: Positive
Predicted class	Negative	91905	78398
	Positive	56086	58381

Figure A.0.2: Confusion Matrices for RF Models

Support Vector Machine Monthly

		True class: Negative	True class: Positive
Predicted class	Negative	65203	62057
	Positive	71415	84592

Support Vector Machine Intraday

		True class: Negative	True class: Positive
Predicted class	Negative	103774	89250
	Positive	44217	47529

Figure A.0.3: Confusion Matrices for SVM Models

Appendix B: Relative feature importance for each model

Logistic Regression for intraday prediction

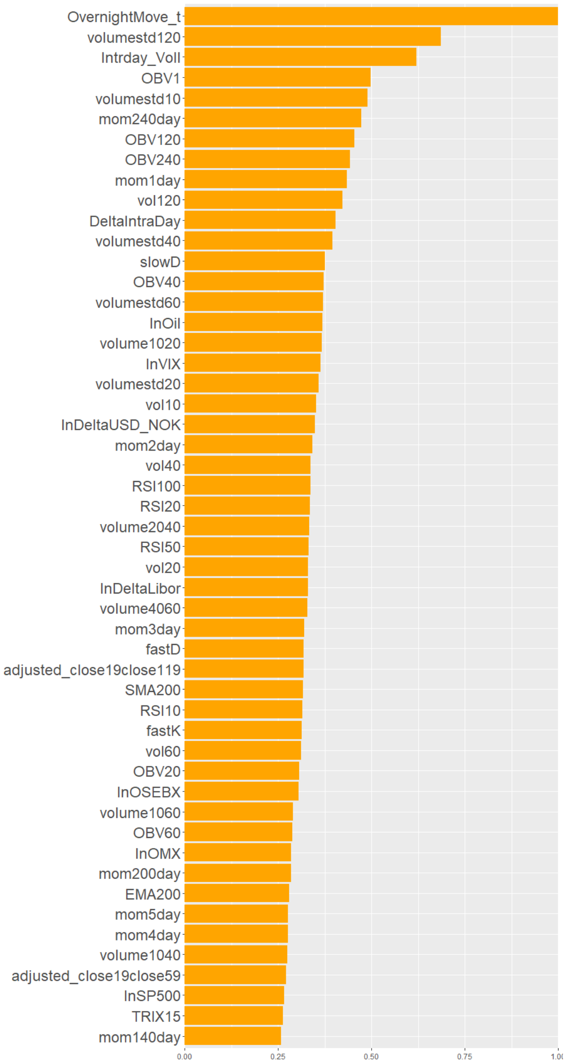

Figure B.1: Features in the Top Half of Relative Feature Importance for the Intraday LR Model

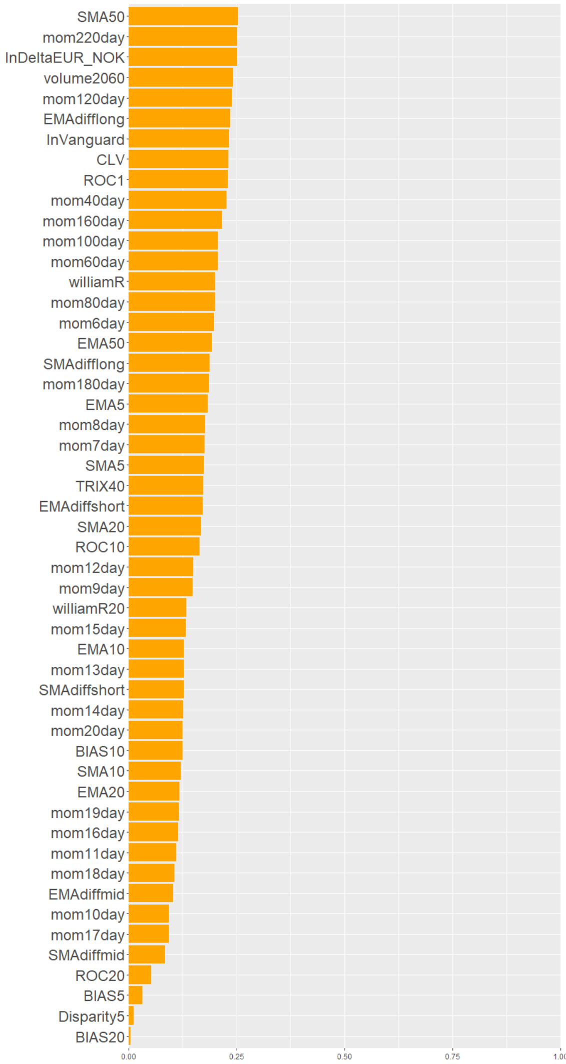

Figure B.2: Feature in the Bottom Half of Relative Feature Importance for the Intraday LR Model

Logistic Regression for monthly prediction

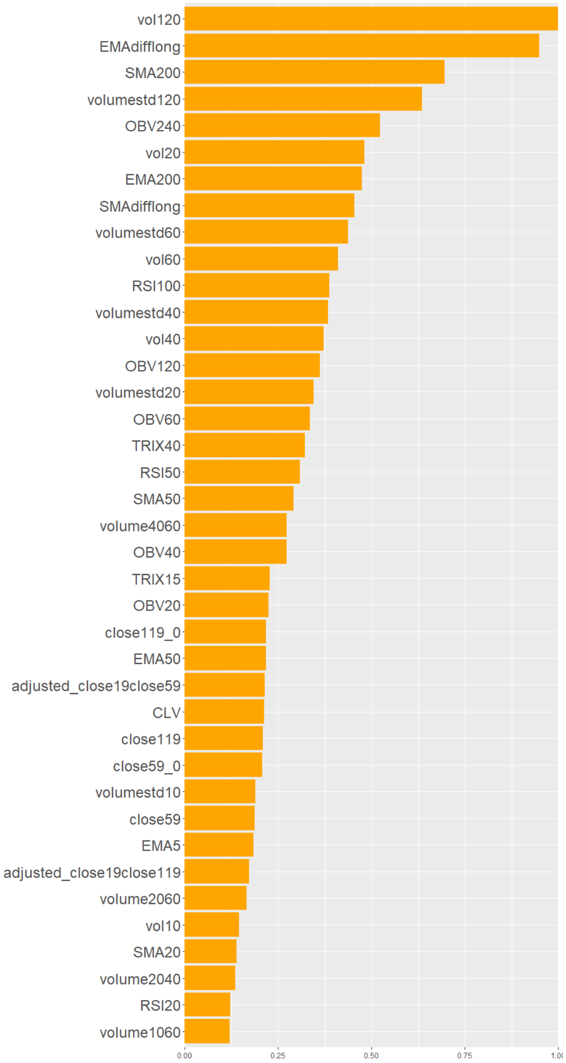

Figure B.3: Features in the Top Half of Relative Feature Importance for the Monthly LR Model

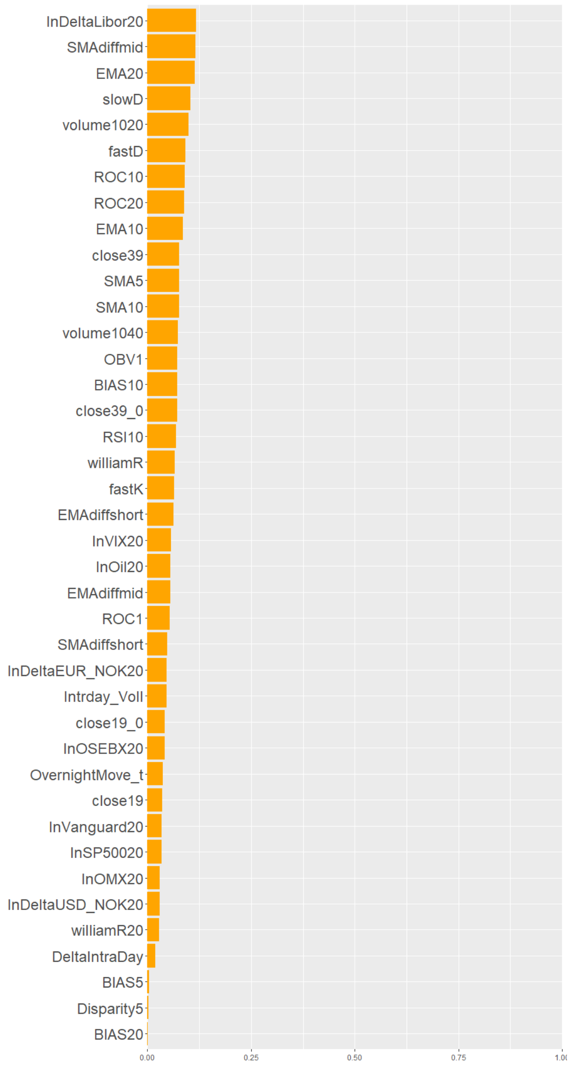

Figure B.4: Features in the Bottom Half of Relative Feature Importance for the Monthly LR Model

Random forest for intraday prediction

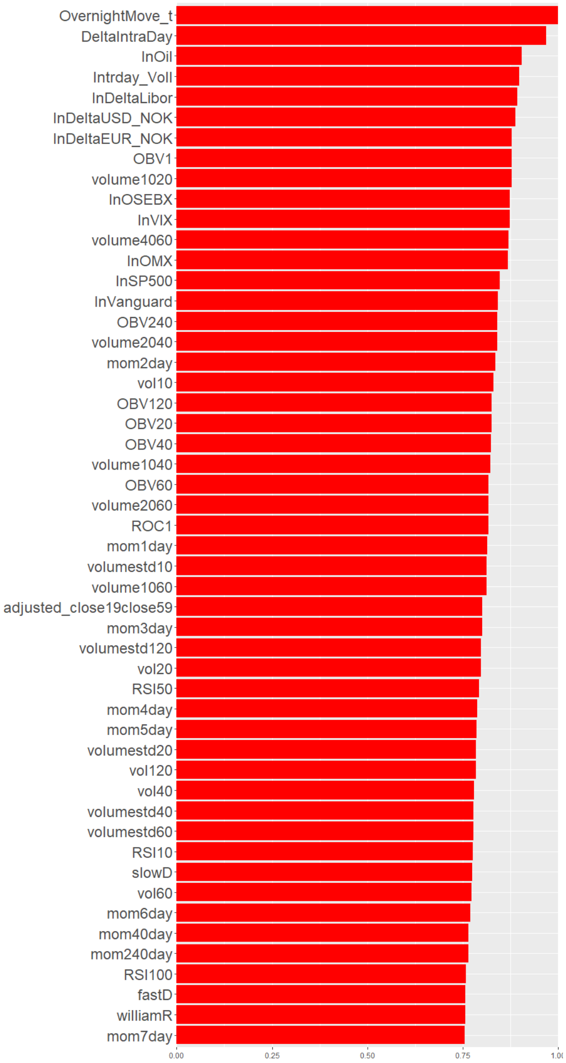

Figure B.5: Features in the Top Half of Relative Feature Importance for the Intraday RF Model

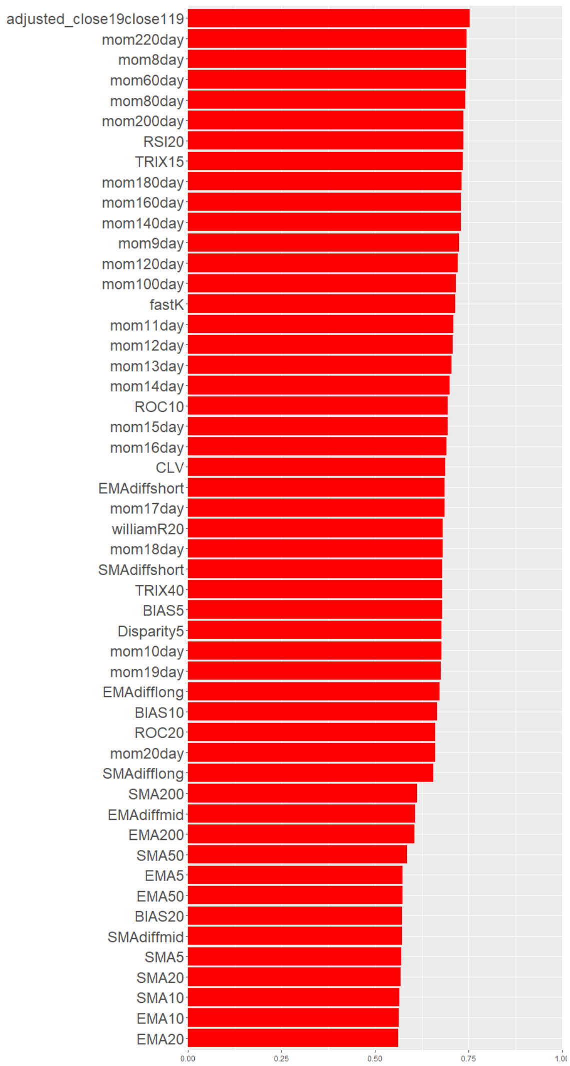

Figure B.6: Features in the Bottom Half of Relative Feature Importance for the Intraday RF Model

Random forest for monthly prediction

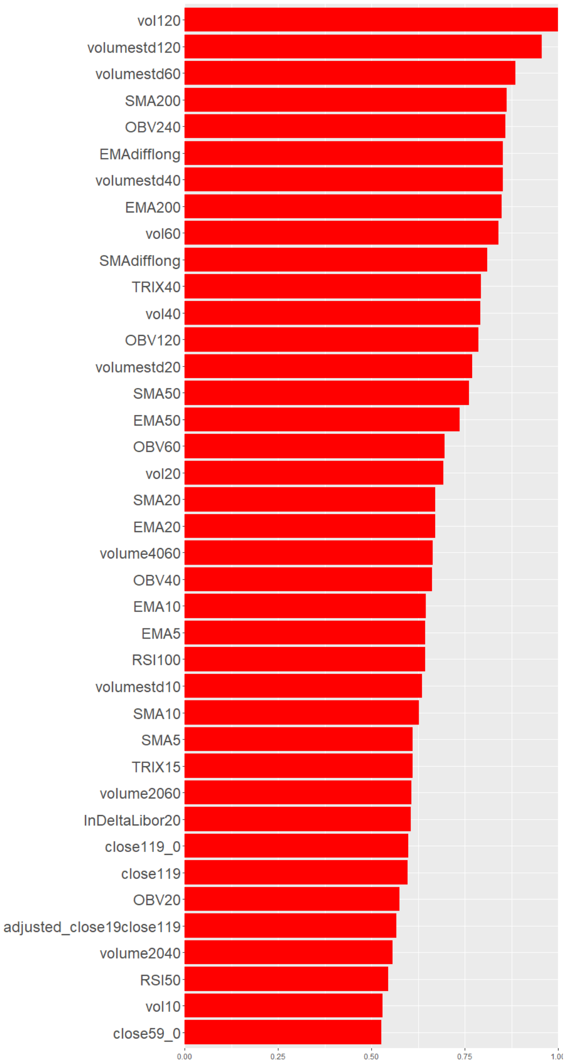

Figure B.7: Features in the Top Half of Relative Feature Importance for the Monthly RF Model

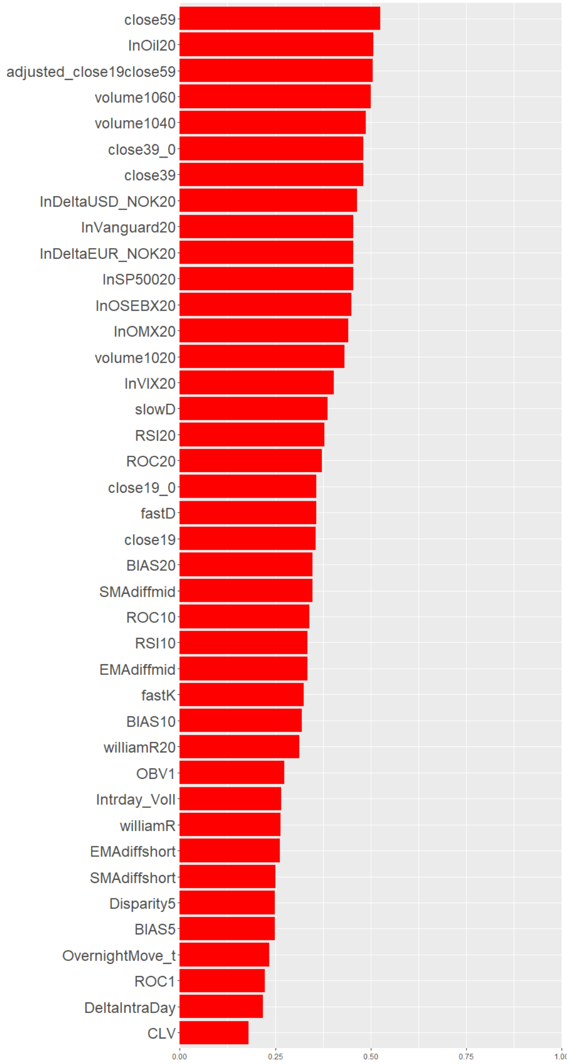

Figure B.8.4: Features in the Bottom Half of Relative Feature Importance for the Monthly RF Model

Support Vector Machine for intraday predictions

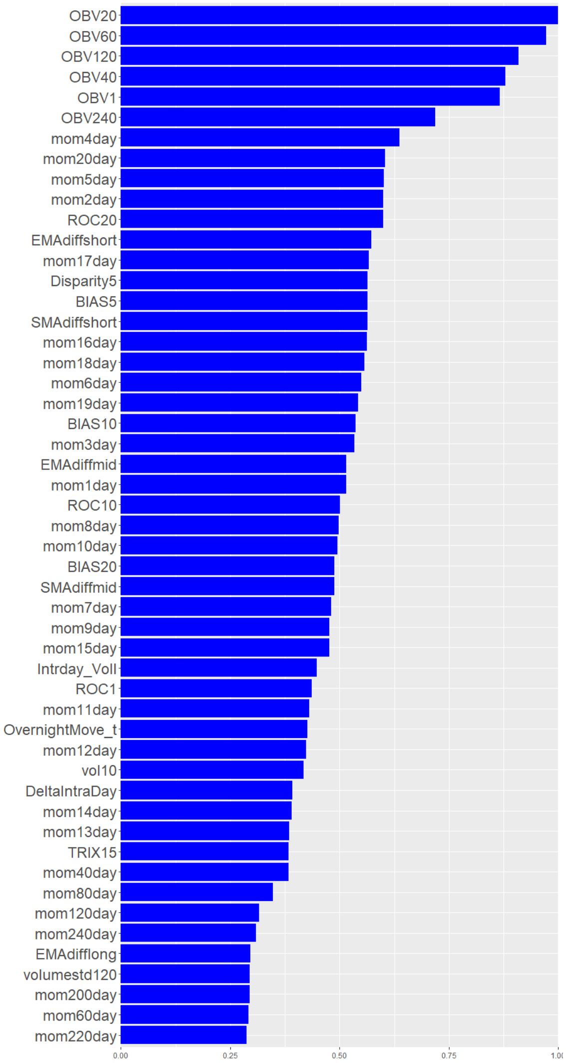

Figure B.9: Features in the Top Half of Relative Feature Importance for the Intraday SVM Model

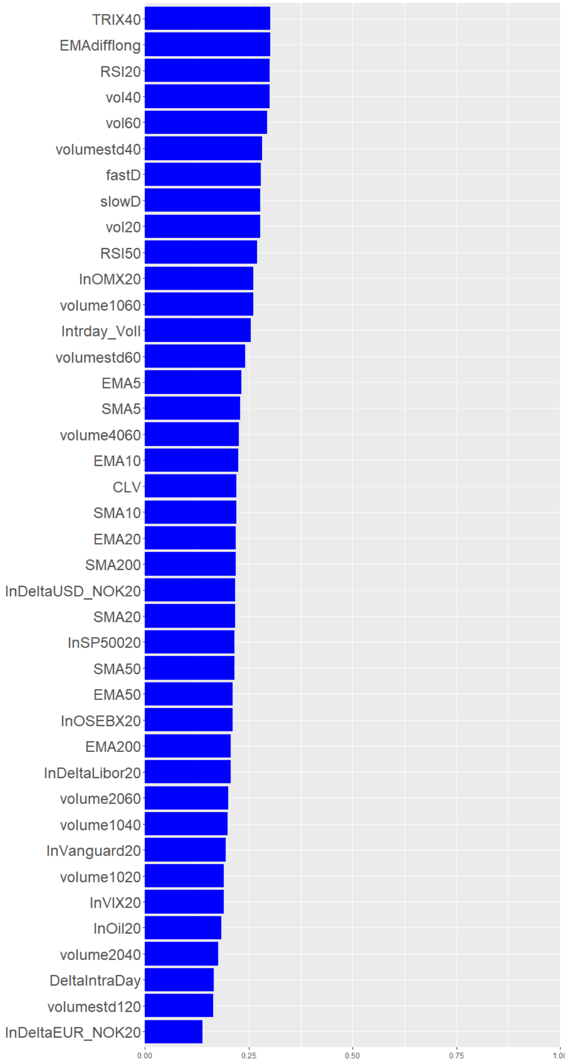

Figure B.5: Features in the Bottom Half of Relative Feature Importance for the Intraday SVM Model

Support Vector Machine for monthly prediction

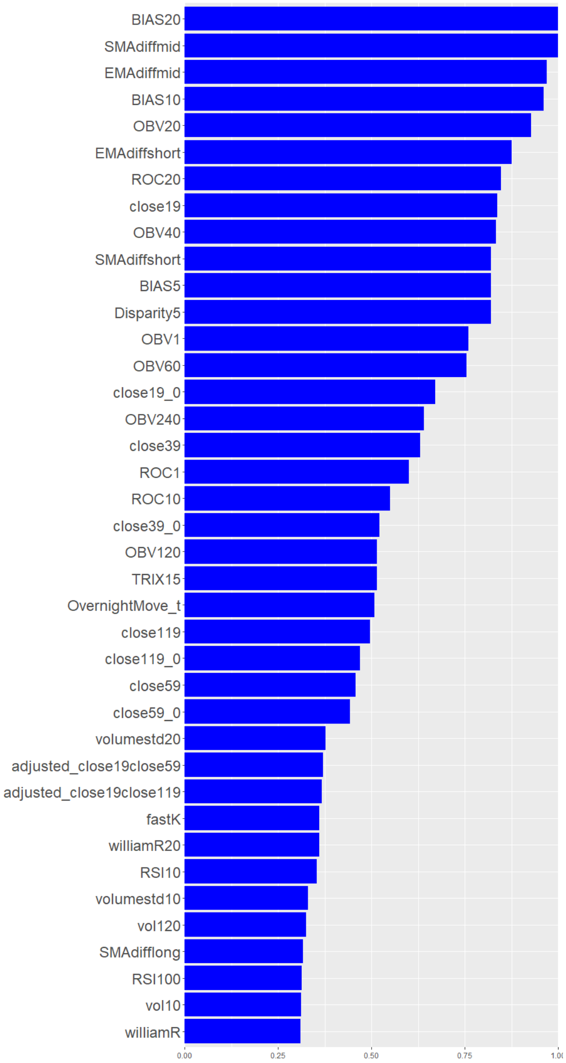

Figure B.7: Features in the Top Half of Relative Feature Importance for the Monthly SVM Model

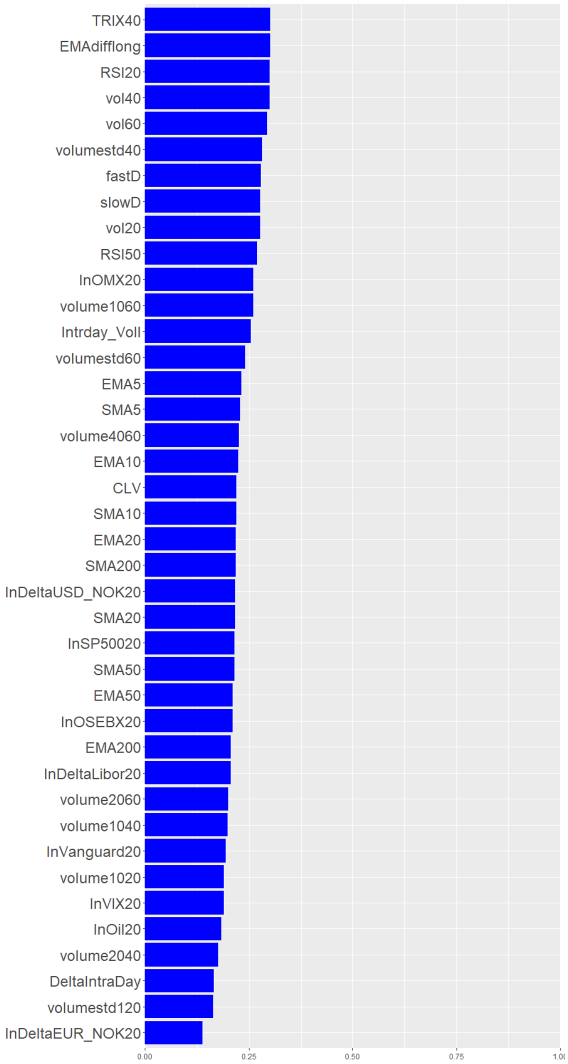

Figure B.6: Features in the Bottom Half of Relative Feature Importance for the Monthly SVM Model