MASTER'S THESIS

Course code:
BE305E

Name:
Ole-Marius Hansen & Joachim Rustad

What impact do overnight returns have on the predictive success of different machine learning models for intraday stock market movements?

Date: 24.05.2024

Total number of pages: 53

NORD University

www.nord.no

Abstract

In this study, we predict the direction of intraday returns (IDR) for several stocks where the aim is to investigate if overnight returns (ONR) can contribute to better predictions. We apply two machine learning algorithms: Random Forest (RF) and Logistic Regression (GLM) in a classification setting. Each algorithm has two sub-models (1) with overnight returns and (2) without overnight return as features. In total, we compare 4 models on machine learning metrics and 4 portfolios on financial metrics to understand how overnight returns influence financial results and the quality of predictions for machine learning algorithms. We find evidence to suggest overnight returns contribute to better financial results when constructing portfolios that invest intraday as both portfolios with overnight returns as features provide higher returns than those without overnight returns. Additionally, both machine learning algorithms with overnight returns outperformed their relative counterparts when compared on measures such as accuracy, precision, recall and F1-score. Finally, variable importance and correlations between intraday returns and overnight returns suggest overnight returns significantly improve predictions of the direction of intraday returns.

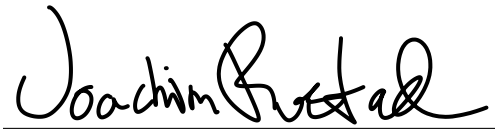
Preface

This master thesis is our final paper as part of our education in Master of Science in Business with a major in Finance and Investments at Nord University in Bodø. Our assignment is written in RMarkdown which provides a professional style and research paper quality layout. During our master studies, we were introduced to the R statistical software in combination with Finance and Machine Learning which we found very interesting. That is why we decided to incorporate Finance and Machine Learning in our final thesis first of all to challenge ourselves and secondly utilize what we have learnt over the past two years. Writing scripts, producing runnable code and finding data proved challenging as our thesis developed. We met several obstacles when gathering data, training machine learning models and performing our analysis.

We want to say a massive thank you to Thomas Leirvik our supervisor, who has provided us with the right guidance, support and valuable feedback throughout this semester. He provided us with valuable feedback several times when we thought we were stuck, and without his commitment, our thesis would not have been what it is today. Additionally, we want to say thank you to family, friends and employers who have supported and provided us with patience. We also want to use the opportunity to thank each other for the good cooperation during our Master's years, especially on this thesis and other previous assignments.

Finally, we want to say thank you to Nord University and Nord Business School (HHN) for the last two years.

Bodø, May 24th 2024



Joachim Rustad

Ole Marius Hansen

Contents

List of Figures

List of Tables

1.0 Introduction

In this thesis, we investigate the phenomenon of overnight return (ONR) using machine learning to analyse if ONR as a feature is a good indicator for predicting the direction of intraday return (IDR) for several stocks. We construct two portfolios for two different machine learning algorithms and a wide selection of features as predictors. Previous research suggests that ONR holds significant predictive power and can serve as a valuable feature in machine learning models. Machine learning techniques excel in handling complex datasets, making them particularly suited for analyzing the intricate dynamics of financial markets (Chen & Hao, 2017). We intend to check if ONR, when used as a feature, increases the prediction performance of the different methods we apply. The performance of the different machine learning models will be evaluated through different evaluation measures, such as accuracy, precision, recall and F1-score. The performance of the portfolios will be measured through different performance evaluations like Sharpe Ratio, Information Ratio, Sortino Ratio and Treynor Ratio. Finally, we regress each portfolio against the Fama French Five-factor model to investigate if we can generate alpha, and understand how the portfolios correlate against market, size, value, liquidity and momentum factors.

Overnight return is defined as the percentage change between the close-price at time t_{-1} and the open-price at time t_0 (Gao et al., 2022). Gao et al. (2022) explains returns by dividing it into two different categories, open-to-close and close-to-open. Close-to-close is better known as total returns and is the most common form of calculating returns. Open-to-close and close-to-open are a breakdown of total returns and are returns in trading hours and non-trading hours, respectively, better known as intraday return (IDR) and overnight return (ONR). To analyze if the prediction performance of IDR is affected by ONR, we run each of our machine learning models with and without ONR as a feature (Hansen & Rustad, 2023).

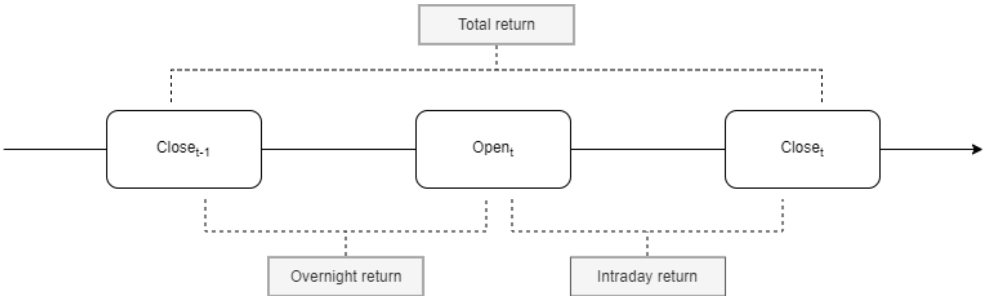


Figure 1: Illustration of IDR and ONR

A fundamental principle in finance is the risk-return trade-off. This principle suggests that the amount of return is reflected by its risk, meaning higher return contains higher risk and vice versa (Bodie et al., 2021). However, Cooper et al. (2008) and Haghani et al. (2022) find significant differences in return between ONR and IDR, where ONR are strongly positive, and IDR is hovering around zero and sometimes even negative. This difference in return is commonly referred to as the anomaly 'overnight effect' and challenges

the fundamental principle of the risk-return trade-off (Haghani et al., 2022). Similar findings by Kelly and Clark (2011) and Liu and Tse (2017) also highlight inconsistencies with this principle, showing that higher returns during overnight trading are accompanied by lower risk compared to intraday trading hours. Evidence by studies from Haghani et al. (2022); An et al. (2022) and Huang et al. (2021) indicates that this 'overnight effect' appears to be a global phenomenon. These are sensational findings, showing the relationship between ONR and IDR when it comes to risk and return, which is a central part of portfolio construction in general.

In the field of finance, risk plays a pivotal role, particularly when dealing with portfolios. In single-factor models like CAPM and multifactor models like Fama-French models, the risk is categorized into systematic and unsystematic factors. Systematic factors represent macroeconomic events, such as political risk, that cannot be diversified, while unsystematic factors encompass firm-specific risks, like debt levels, that can be diversified (Bodie et al., 2021). Diversification is well known investment strategy, and implies spreading risk across different assets, that operate in different regions, industries etc. This way the unsystematic risk can in the best case scenario be eliminated, resulting in a less risky portfolio (Bodie et al., 2021). We can evaluate the portfolio performance by using various risk measures such as the Sharpe Ratio, Information Ratio, Treynor Ratio, and Sortino Ratio (Bodie et al., 2021). These ratios provide different perspectives on the risk-adjusted returns of a portfolio. In our study, we focus on stocks from the Oslo Stock Exchange. The performance of these stocks is assessed using the previously mentioned performance evaluation ratios.

Portfolio construction is closely connected to risk factors and returns. In essence, a portfolio is a combination of different assets, but its construction is a complex process where various factors must be considered. These factors include preferences regarding return and liquidity, investment period, risk tolerance, and other limitations or preferences. The construction of portfolios often involves market analysis where these factors are considered. One of the key benefits of portfolios is the reduction of unsystematic risk through diversification, as previously explained. The construction of portfolios also involves deciding the optimal weight for the different assets based on the portfolio's objective. As well as monitoring the portfolio and making necessary adjustments to maintain its integrity (Kyurkchan, 2020). In our research, the portfolio solely consists of stocks from the Oslo Stock Exchange and are selected based on the prediction output.

Machine learning is a relatively new tool in several fields among these we find it in both research and finance. It involves the creation of models based on a provided dataset, which is typically split into training and test sets. There exists a variety of machine learning algorithms which vary in complexity and offer unique advantages. Machine learning techniques are broadly categorized into supervised and unsupervised learning. This thesis primarily focuses on supervised learning, where the model is trained using labelled data. Each data point in this case is associated with a known outcome or response variable. Conversely, unsupervised learning operates on unlabeled data, seeking to identify hidden patterns or structures within the dataset. The purpose of machine learning models is to analyse datasets, identify patterns, and learn relationships between variables.

Supervised learning can be further divided into classification and regression tasks. This thesis focuses on classification tasks, which deal with predicting discrete outcomes, often binary. On the other hand, regression involves predicting a continuous response variable. Machine learning models can be evaluated using various performance metrics (James et al., 2013). In our research, we employ supervised learning with a binary response variable to predict the direction of intraday return, where '1' indicates an up signal and '0' indicates a down signal. A more comprehensive discussion of machine learning and its components is discussed in subsection 3.0. Here, we delve into the theoretical underpinnings of these models and their relevance to our research objectives.

The use of machine learning in financial research has gained traction in recent years, owing to its effectiveness in navigating the complexities of financial systems (Karaca et al., 2020; Zhong and Enke, 2019). Machine learning algorithms offer advantages over traditional statistical techniques such as time series modelling in handling such complexities and are increasingly employed for accurate market predictions (Chen & Hao, 2017). Machine learning also has a wide range of applications, including various tasks in finance. In the context of this thesis, machine learning is utilized for conducting advanced quantitative market analysis. This analysis is based on historical data and typically makes use of a selection of machine learning algorithms. The machine learning algorithms learn underlying patterns of the historical data and create predictions for future values. In this thesis, we have selected two machine learning algorithms: Logistic regression (GLM) and Random Forest (RF). Logistic Regression, chosen for its simplicity, serves as a baseline model for comparison purposes. In contrast, Random Forest was chosen based on its widespread popularity, strong predictive performance, and versatility in handling complex datasets (Ballings et al., 2015). A more comprehensive discussion of these algorithms are provided in subsection 4.0.

The construction of portfolios using machine learning primarily revolves around applying the predictions from each machine learning model. These predictions can be for either individual stock returns, ETF's or index returns (Gao et al., 2022; Hansen and Rustad, 2023). Prediction output can be probabilities or the actual prediction value. In this thesis, each model generates predictions for either an increase or decrease. These predictions are provided as probabilities where we rank each stock from the highest probability of increase to the lowest. The top k stocks are then selected for the portfolio and are equally weighted similarly to Kilskar (2020). Since we are predicting the direction of intraday return our portfolios of stocks do not hold any positions overnight, and are rebalanced daily.

In this thesis we apply two machine learning supervised classification methods to predict the direction of stocks in the Norwegian market. We find significant differences between the portfolios with and without ONR as a feature in generating returns for both machine learning models. This indicates that ONR is a good indicator for predicting the direction of IDR, as we see a significant improvement in prediction power when comparing the results of the performance metrics applied. Additionally, we see in the variable importance plot (VIP) and correlation between ONR and IDR, that ONR as a variable plays a significant role and

improves the prediction power of IDR. Finally, all our portfolios except Random Forest without ONR can generate significant positive alpha when regressed against the Fama French Five Factor model.

In December of 2023, we conducted a smaller analysis in an assignment regarding the same topic. In that assignment, we focused on the SPY ETF, an exchange-traded fund tracking the S&P500 index. Here we did not find evidence of an 'overnight effect', indicating that ONR did not serve well as a predictor for IDR. The reason for this is twofold. First, SPY is one of the most traded assets in the world and contains many of the most analyzed firms such as Microsoft, Apple, Exxon, Facebook, and Coca-Cola. This indicates that the SPY ETF is more efficiently priced than other assets available for investors. Second, the significant autocorrelation relationship between ONR and IDR we found for single assets in this thesis is netted out in a portfolio of assets, such as the SPY ETF where we did not find any significant autocorrelation between IDR and ONR (Hansen & Rustad, 2023).

Part of the explanation for our findings regarding the Oslo stock exchange may be attributed to liquidity reasons. Branch and Ma (2012) emphasize that the effect of ONR on IDR is more pronounced in smaller and less liquid stocks, while Berkman et al. (2012) highlight the significance of stock valuation complexity in influencing this relationship. This underscores the impact of stock liquidity on the ONR-IDR dynamic. General studies on return and liquidity suggest that stocks with higher illiquidity tend to yield higher returns (Amihud, 2002). Moreover, a machine learning study on predicting returns in the Chinese stock market identifies liquidity as the most crucial predictor (Leippold et al., 2022). In our thesis we did not find any evidence to suggest illiquid assets contribute to higher returns than efficient ones when regressed against the Fama French Factor *liquidity*.

In studies utilizing machine learning for prediction in financial markets, classification models are often favoured over regression models. This approach involves predicting the direction of market movements rather than the exact values, which is considered more prevalent due to the dynamic, noisy, and chaotic nature of stock markets (Gao et al., 2022; Zhong and Enke, 2019). While predicting direction may appear simpler, it poses significant challenges for both investors and researchers. Despite these challenges, several research papers have successfully employed supervised learning in a classification setting to predict the direction of ETFs, yielding promising results (Malagrino et al., 2018; Hoseinzade and Haratizadeh, 2019; Long et al., 2019).

Malagrino et al. (2018) demonstrated the predictive power of their machine learning model by incorporating various global indices to forecast the direction of the Sao Paulo Stock Exchange, achieving a mean accuracy of 71% and a top accuracy of nearly 78%. Hoseinzade and Haratizadeh (2019) employed machine learning techniques to forecast the direction of US stock indices, including the S&P 500, NASDAQ, DJI, NYSE, and RUSSEL, surpassing existing algorithms in accuracy and performance. Similarly, Long et al. (2019) reported significant findings in their study on predicting the Chinese stock market index CSI 300, outperforming both machine learning and statistical models in terms of accuracy, profitability, and stability.

While there exists a wealth of academic research on the predictive capabilities of various machine learning models, relatively few studies have focused specifically on the predictive power of overnight return (ONR). However, Gao et al. (2022) demonstrated the effectiveness of machine learning in forecasting the direction of ONR across Asian, American, and European markets. Their proposed model outperformed competing methods in terms of accuracy, F-measure, and Sharpe Ratio, indicating promising potential for ONR prediction. In a study on the predictability of the Chinese crude oil market, Wen et al. (2021) identified a strong intraday reversal effect, with returns from the previous night exhibiting significant predictive power. Furthermore, they observed that predictability was enhanced during periods characterized by greater overnight information, higher trading volume, volatility, and lower liquidity. Similarly, Chu et al. (2019) investigated intraday momentum and reversal effects in the Chinese stock market, finding promising predictive results using the first half-hour of intraday returns as predictors. Their study revealed robust predictive power when incorporating factors such as yesterday's return, ONR, and day-of-week effects.

These findings collectively underscore the potential of machine learning techniques in uncovering predictive patterns within financial markets, particularly concerning overnight and intraday dynamics.

1.1 Problem statement

Building upon the foundation laid in the introduction, this thesis aims to rigorously investigate the relationship between Overnight Returns (ONR) and Intraday Returns (IDR). We accomplish this by employing established machine learning methodologies, which have gained widespread acceptance for their efficacy in addressing similar research questions.

In addition, this research will consider the selection of appropriate models and features, considering varying conditions as delineated in existing literature. The objective of the study is to ascertain whether the inclusion of ONR as a feature can enhance the predictive accuracy of the model and potentially yield excess returns.

The research question for the thesis is as follows:

What impact do overnight returns have on the predictive success of different machine learning models for intraday stock market movements?

This thesis is a continued study from an assignment we worked on in the previous semester and was meant to be a preparation study for the master thesis. The purpose of this thesis is to examine a less liquid market than the SPY ETF we investigated in our assignment to find out if ONR can predict the direction of IDR.

The layout of this thesis begins with an introduction that outlines the research problem and the significance of the study. This is followed by a literature review that delves into portfolio and machine learning theories. The subsequent section details the methodology and data used in the study, including data sources, collection methods, and the machine learning techniques employed. The analysis and results of the study are then presented, with a thorough discussion of the findings. The thesis concludes with a discussion that interprets the results in the context of the research problem, and a conclusion that summarizes the study and suggests areas for future research. References and appendices are included at the end to provide additional supporting information.

2.0 Portfolio theory

This section gives the basic theory to understand the research in this paper. It is split into two subsections: Factor Models and Model Evaluations. In the first subsection, Factor Models, we delve into the single-factor models, with a particular emphasis on the Capital Asset Pricing Model (CAPM), and multi-factor models, where our focus is primarily on the Fama-French multi-factor models. The second subsection, Model Evaluation, underscores the significance of robust evaluation methods in validating the efficacy of the models. Here, we explore commonly employed evaluation metrics such as the Sharpe Ratio, Information Ratio, Treynor Ratio and Sortino Ratio, providing a comprehensive understanding of their application and interpretation.

2.1 Factor models

As an integral part of portfolio theory, factor models provide a structured approach to understanding, managing and optimizing portfolios (Bodie et al., 2021).

In portfolio management, factor models are serving as a key instrument for identifying the determinants of asset returns and uncovering potential risks. These models break down the returns of an asset into different factors, which are variables that have a systematic influence on returns. By doing so, factor models help us understand the underlying drivers of asset performance (Bodie et al., 2021).

The utility of factor models extends to various aspects of portfolio management. In risk management, they enable the identification and quantification of different sources of risk in a portfolio. In performance attribution, factor models help in dissecting the portfolio's returns to understand which factors contributed to its performance. Lastly, in portfolio construction, factor models guide the allocation of assets by identifying those that offer the best return for a given level of risk based on their factor exposures (Bodie et al., 2021).

2.1.1 Single-factor models

Single-factor models are a type of financial model that associates the return of a security to a single risk factor in a linear model. These models are pivotal in understanding the interplay of factors affecting asset returns, enabling a deeper comprehension of investment dynamics (Bodie et al., 2021).

A single-factor model calculates the actual return based on the previously expected return $E(r_i)$ plus unanticipated surprises. These surprises, which represent the risk, are further divided into macroeconomic surprises or market risk m and firm-specific surprises e_i. These two components capture the systematic and unsystematic risk of return, respectively. It is assumed that these two risk measures are uncorrelated (Bodie et al., 2021).

To account for the sensitivity of different securities in various markets, the market risk, m, is multiplied by

the sensitivity coefficient β_i. This gives us Formula 1 for single-factor models:

$$r_i = E(r_i) + \beta_i \cdot m + e_i \tag{1}$$

Notice that if the market risk is 0, meaning no macro surprises, the excess return will equal the expected return of the previous period plus the firm-specific events only. This highlights the role of firm-specific events in influencing the returns when there are no macroeconomic surprises (Bodie et al., 2021).

2.1.2 Capital asset pricing model (CAPM)

The Capital Asset Pricing Model (CAPM) is a cornerstone of modern financial economics, providing a precise prediction of the relationship between the risk of an asset and its expected return. First published in the mid-1960s by William Sharpe, John Lintner and Jan Mossin, the model has since undergone numerous extensions. This subsection focuses on the simple CAPM model, but it's worth noting that more sophisticated variants may differ in their risk-return trade-off. However, the concept that risk premia are proportional to exposure to macroeconomic risk factors and independent to firm-specific risk factors generally holds true across all extensions (Bodie et al., 2021).

The CAPM calculates the actual return r_i based on the risk-free rate r_f plus the market sensitivity coefficient β_i known as the asset's beta. This beta is multiplied by the expected excess return, known as the equity risk premium ERP, which is the difference between market return r_m and risk-free rate r_f (Bodie et al., 2021).

The formula for the simple Capital Asset Pricing Model (CAPM) is shown in Formula 2:

$$E(r_i) = r_f + \beta_i \cdot (r_m - r_f) = r_f + \beta_i \cdot ERP \tag{2}$$

Where r_i is the expected return, r_f is the risk-free rate, β_i is the beta coefficient, and $r_m - r_f$ or ERP is the equity risk premium, where r_m is expected return on the market portfolio.

In the CAPM, the market portfolio r_m represents the expected return on a portfolio comprising all risky assets in the market, weighted by their market values. It embodies the market's inherent risk and return characteristics. The risk-free rate r_f serves as a benchmark for a risk-free position and is most commonly associated with government bonds and treasury bills. The difference between the expected market return and the r_f is known as the Equity Risk Premium ERP. The ERP is essentially the excess return that investing in the stock market provides over a risk-free rate. This premium motivates investors for taking relative higher risk when investing in the asset. The interplay of these three components forms the crux of the CAPM, helping investors understand the risk-return tradeoff in financial markets. The risk premium is the market return minus the risk-free rate $r_m - r_f$, multiplied by the beta β of the security or portfolio. The beta measures the sensitivity of the security's or portfolio's returns to the market return (Bodie et al., 2021).

Table 1: CAPM assumptions

Category	Assumption
1. Individual behavior	
	a) Investors are rational, mean-variance optimizers. b) Their common planning horizon is single period. c) Investors all use identical input lists, an assumption often termed homogeneous expectations. Homogeneous are consistent with the assumption that all relevant information is public available.
2. Market structure	
	a) All assets are public held and trade on public exchanges. b) Investors can borrow or lend at a common risk-free rate, and they can take short positions on trade securities. c) No taxes. d) No transaction costs.

As can be seen in Table 1 collected from Bodie et al. (2021, p. 276), there are several assumptions that underpin the CAPM model. These assumptions are organized into two categories: individual behaviour and market structure.

In terms of individual behaviour, The CAPM assumes rational behaviour among individual investors, who evaluate investment alternatives by evaluating risks against potential returns and select options that optimize this trade-off. Additionally, it posits that investors operate within a short planning horizon for their investment decisions. Moreover, the model assumes that all relevant information is readily accessible to individual investors, where all agents in the market form their expectations about the future on the same information (Bodie et al., 2021).

As for the market structure, the CAPM assumes that all assets in the market are equally available for all investors at any given time, meaning there are no restrictions or barriers to investing. Additionally, it assumes investors can freely borrow or lend at a universal risk-free rate, while also being able to engage in short selling of traded securities. Lastly, the model assumes that there are no taxes and costs when it comes to trading, thereby no influence on returns (Bodie et al., 2021).

These assumptions serve as simplifications within the CAPM framework and may not always align with real-world market conditions. Consequently, predictions made by the CAPM should be approached with caution. While the model offers valuable insights into risk-return dynamics, its practical application may demand adjustments to accommodate market imperfections and variations in investor behaviour that diverge from the model's assumptions (Bodie et al., 2021).

2.1.3 Multi-factor models

Multi-factor models are an extension of the single-factor model. As we mentioned regarding single-factor models, there are systematic and unsystematic risk factors occurring in the market. In single-factor models, these two risk types are symbolized with β_i and e_i, respectively. In a multi-factor model, there are multiple

macro events or systematic risk factors, where each risk has its own beta coefficient. The value of this $\beta_{i,k}$ represents how sensitive the specific company is to the specific risk. Beta values can differ based on various factors such as industry, company size, etc. Examples of risk factors could include interest risk or inflation risk, and risk related to changes in energy prices or GDP. Each risk factor has its own risk premium, meaning that the market return is reflected by multiple macroeconomic risk factors. It's reasonable to believe that this will provide a more accurate description of return compared to a single-factor model (Bodie et al., 2021).

Multi-factor models are useful in risk management applications as they provide a simple measure of risk exposure to different macroeconomic events. This allows investors and portfolio managers to identify risks and construct portfolios that hedge these risks (Bodie et al., 2021).

The formula for a multi-factor model is shown in Formula 3:

$$r_i = E(r_i) + \beta_i \cdot F_1 + \beta_2 \cdot F_2 + ...\beta_{i,k} \cdot F_k + e_i \tag{3}$$

Where $E(r_i)$ is the asset i's expected return, F_k is the k factor, $\beta_{i,k}$ is the sensitivity of the i asset to the k factor, and e_i is the firm-specific risk or unsystematic risk of asset (Bodie et al., 2021; Kyurkchan, 2020).

One of the key strengths of multi-factor models lies in their ability to offer a more nuanced and accurate estimation of returns. This is because they account for the varying sensitivities that different firms, operating in diverse markets, have towards distinct risks. Such varied responses to macroeconomic risks are beyond the scope of single-factor models, which makes multi-factor models a more comprehensive tool for risk assessment. Among the various multi-factor models available, the Fama-French multi-factor models stand out as widely recognized and commonly used examples (Bodie et al., 2021; Kyurkchan, 2020).

2.1.4 Fama-French factor models

Eugene F. Fama and Kenneth R. French developed a series of multi-factor models that identify the most significant sources of macroeconomic risk variables. These models serve as an extension of the Capital Asset Pricing Model (CAPM). In our study, we employ the five-factor model, which is a variation of the three-factor model with two additional variables. This subsection aims to provide a comprehensive overview of this multi-factor model, facilitating a general understanding of its structure and function (Bodie et al., 2021)).

The Fama-French three-factor model and its variants have been at the forefront of empirical research on security returns. These models are predicated on macroeconomic variables that have historically demonstrated a strong predictive capacity for past returns. As such, they are expected to capture systematic risk effectively and may also account for risk premiums (Bodie et al., 2021).

The Fama-French three- and five factor model shown in Formulas 4 and 5 respectively.

$$R_{i,t} = \alpha_i + \beta_{i,M} R_{M,t} + \beta_{i,SMB} SMB + \beta_{i,HML} HML + e_{i,t} \tag{4}$$

Where a_i is the intercept, R_{Mt} is the market return, SMB is the size premium, HML is the value premium and e_{it} is the error term (Bodie et al., 2021; Fama and French, 1993).

$$R_{i,t} = \alpha_i + \beta_{i,M} R_{M,t} + \beta_{i,SMB} SMB + \beta_{i,HML} HML + \beta_{i,MOM} MOM + \beta_{i,LIQ} LIQ + e_{i,t} \tag{5}$$

Where a_i is the intercept, R_{Mt} is the market return, SMB is the size premium, HML is the value premium, MOM is the momentum factor, LIQ is the return on liquidity factor, and e_it is the error term (Fama & French, 1993). The HML factor stands for High Menus Low and represents the spread in return between companies with high and low book-to-market ratios. The SMB factor stands for Small Minus Big and represents the spread in returns between small and large companies. The MOM factor stands for Momentum which is unique for the four-factor model and represents the tendency of stocks with high past returns to have high future returns, and vice versa. The final additional factor for the five-factor model is the liquidity factor, which measures the difference in returns between assets with high liquidity and those with low liquidity (Bodie et al., 2021; Fama and French, 1993).

As we can observe, the five-factor model introduces two additional variables, the momentum and liquidity factors, which enhance the model's explanatory power. However, one of the challenges with approaches like the Fama-French models is that the risk factors are not always clearly identified. Despite this, these models have demonstrated good predictive results across different periods and markets worldwide, making them a valuable tool in financial analysis (Bodie et al., 2021).

2.2 Performance evaluation

The objective of active portfolio management is to construct a better-performing portfolio than the benchmark. The excess return of a benchmark-beating portfolio is often known as alpha. By investing in a portfolio with positive alpha the investors will achieve portfolio returns above the benchmark. However, constructing portfolios with positive alpha is considered a challenging task (Kyurkchan, 2020). In portfolio management, performance evaluation plays a central role. It involves the measurement, assessment and analysis of investment results which serves as a tool for evaluating both the portfolio and portfolio manager. The main goal in performance evaluation is to consider the generated return up against its level of risk, and how this correspond with the portfolio's objectives. It offers valuable insights into the effectiveness of portfolio strategies and the competence of portfolio managers in realizing desired investment objectives (Kyurkchan, 2020; Bodie et al., 2021).

Performance evaluation is a useful tool for decision-making for both investors and portfolio/fund managers. By evaluating the performance of their portfolios, investors can compare the performance of their investments against the market benchmark or other comparable investments. This way allows investors to make informed decisions based on their risk tolerance and investment strategy. Furthermore, performance evaluation offers a transparent mechanism for holding managers accountable for delivering favourable results for investors. It serves as a tool that can provide valuable insights for strategic planning aimed at enhancing portfolio performance. This process not only assesses the current state of the portfolio but also aids in identifying potential areas of improvement. Thus, performance evaluation is integral to the strategic planning process, ultimately contributing to the overall success of the investment strategy (Kyurkchan, 2020; Bodie et al., 2021).

Performance evaluation in portfolio management carries significant implications for investment decisions. It equips investors with a quantitative understanding of their portfolio's return and risk, which is vital when deciding on investment positions and conducting necessary risk assessments. This information further provides insights into any adjustments required to align with the investment strategy and risk tolerance. Performance evaluation also offers a comprehensive view of the portfolio's performance in comparison to the benchmark and other similar investment alternatives. Moreover, it serves as a measure of the portfolio manager's performance, ensuring transparency for clients (Kyurkchan, 2020; Bodie et al., 2021).

2.2.1 Popular evaluation ratios

Common evaluation ratios for measuring portfolio performance are the Sharpe ratio, Information ratio, Sortino ratio and Treynor ratio. These provide quantitative measures that are a useful tool for decision-making for both investors and portfolio managers. Each of these evaluation ratios provides a unique perspective on investment performance, and therefore every measure has its unique approach and focus. Evaluation ratios serve as a critical tool in portfolio assessment and measure the performance of the portfolio relative to a risk-free position, considering the risk that has to be taken. These ratios measure not only the performance of the portfolio but also the performance of the portfolio manager (Kyurkchan, 2020; Bodie et al., 2021).

Sharpe ratio measures the trade-off between a portfolio's excess return and its risk in a specific period. The excess return is calculated by the difference between the return of the portfolio and the risk-free rate. Further, the excess return is divided by the portfolio's standard deviation which in this context often is called volatility or total risk. To compare portfolios, a higher Sharpe Ratio value is preferred. This indicates that the portfolio provides a better return for each unit of risk taken (Kyurkchan, 2020; Bodie et al., 2021).

The Sharpe Ratio SR_p of a portfolio is calculated as follows:

$$SR_p = \frac{R_p - R_f}{\sigma_p} \tag{6}$$

Where R_p is the return of the portfolio, R_f is the risk-free rate, and σ_p is the standard deviation of the portfolio's return.

Information-ratio measures the consistency of active return or the abnormal return per unit of nonsystematic risk. Abnormal return, also known as alpha, is the excess return achieved by a portfolio over its benchmark. On the other hand, nonsystematic risk, often referred to as 'tracking error', 'active risk', or 'benchmark tracking error', represents the additional risk that the benchmark carries. The Information Ratio is often considered a crucial criterion for evaluating active investors, as it measures the ability of portfolio managers to generate excess returns relative to a benchmark, per unit of tracking error. A higher IR indicates a more skillful manager (Kyurkchan, 2020; Bodie et al., 2021).

The Information Ratio IR_p of a portfolio relative to a benchmark is calculated as follows:

$$IR_p = \frac{R_p - R_b}{\sigma_{(R_p - R_b)}} = \frac{R_a}{\sigma_a} \tag{7}$$

Where R_p is the return of the portfolio, R_b is the return of the benchmark, $\sigma_{(R_p - R_b)}$ is the standard deviation of the difference between the portfolio and benchmark return, and R_a and σ_a represent the active return and active risk, respectively.

Sortino ratio, a variant of the Sharpe ratio, uses the lower partial standard deviation (LPSD), or 'bad' standard deviation σ_d, instead of the total standard deviation σ_p. This provides a more nuanced view of risk, especially when the return distribution is non-normal, i.e., skewed or has extreme values. The use of standard deviation as a risk measure presents two problems in such scenarios: (1) it does not differentiate between upside and downside volatility, and (2) it does not consider the risk-free rate of return as a benchmark. The Sortino ratio addresses these issues by focusing on the standard deviation of negative excess returns or returns below the risk-free rate. In scenarios with the non-normal distribution of return, the Sortino rate is considered a more effective evaluation ratio compared to the Sharpe ratio. A higher Sortino ratio is preferred when comparing different portfolios and indicates more efficient performance adjusted for risk (Bodie et al., 2021).

The Sortino Ratio SR_p of a portfolio is calculated as follows:

$$SR_p = \frac{R_p - r_f}{\sigma_d} \tag{8}$$

Where R_p is the return of the portfolio, r_f is the risk-free rate, and σ_d represents the downside deviation, also known as the Lower Partial Standard Deviation (LPSD).

Treynor ratio, also known as the reward-to-risk ratio, measures the excess return per unit of risk like the Sharpe ratio, the difference is that it uses systematic risk instead of total risk. The systematic risk

of the portfolio is measured by its beta. The Treynor ratio is an appropriate measure for comparing the performance of different portfolio managers, as it considers systematic risk, which, unlike nonsystematic risk, cannot be significantly reduced through diversification. A higher Treynor rate indicates that the portfolio manager has been generating returns more efficiently and managing the exposure to market risk effectively. In other words, a larger Treynor ratio is generally preferable when comparing different portfolios (Bodie et al., 2021).

The Treynor Ratio TR_p of a portfolio is calculated as follows:

$$TR_p = \frac{R_p - r_f}{\beta_p} \tag{9}$$

Where R_p is the return of the portfolio, r_f is the risk-free rate, and β_p is the portfolio's beta, measuring its sensitivity to market movements.

3.0 Machine learning theory

This section clarifies the fundamental principles of machine learning that form the foundation of this thesis. It is organized into subsections, each dedicated to explaining key aspects of machine learning. Over recent years, machine learning has emerged as a prominent field within science and research. Consequently, this section focuses on providing a concise overview of the fundamentals. For a more comprehensive understanding of the topics discussed, we recommend referring to James et al. (2013).

3.1 Introduction

Financial markets are recognized as complex, chaotic, and dynamic systems, influenced by a multitude of highly interrelated factors, such as economic, industry-specific, company-specific, psychological, and political factors that drive market trends in various directions (Ballings et al., 2015; Zhong and Enke, 2017). The constant influx of information in the financial industry makes prediction a challenging task, even when the goal is merely to predict the direction of the market, which is the most common form of prediction (Karaca et al., 2020; Zhong and Enke, 2019; Ballings et al., 2015; Gao et al., 2022).

Machine learning has gained popularity among financial researchers and investors due to its ability to handle complex systems like financial markets (Chen & Hao, 2017). As a tool for understanding data, machine learning's importance has grown with the rise in computational power and information technology. Machine learning employs algorithms that are trained on historical datasets to create self-learning models. These models are capable of making predictions and classifications when presented with new data (James et al., 2013).

Within empirical studies in machine learning, there are two critical stages, which are fundamental processes known as the training- and test phases. In the training phase, the machine learning model learns to make predictions or decisions based on the dataset provided. This phase is also known as the training set and consists of input- and output data which the model uses to understand the underlying patterns and relationships. In the following phase, the test phase, the model is evaluated using a separate dataset consisting of unseen data. With the information from this phase, we can assess how well the model has learned from the training phase and how it performs on unseen data (James et al., 2013).

3.2 Supervised learning

Machine learning can be broadly divided into two fundamental categories: supervised and unsupervised learning. A supervised learning model is trained on the labeled dataset, which means the model is provided with a dataset where each observation has a corresponding output label. The model undergoes a learning process, also known as the training phase, where it learns from its inputs and the corresponding outputs. The goal of this training phase is to make accurate predictions for new, unseen data. In the subsequent phase, the

test phase, the model is provided with new unseen data and makes predictions based on what it has learned in the training phase. (James et al., 2013). It's important to emphasize the difference between splitting data within time series and cross-sectional analysis. Time series must consider temporal dependencies, whereas in cross-sectional analyzes the data is randomly split. To take into account the temporal dependence in time series, techniques such as rolling window and temporal cross-validation are used (James et al., 2013).

In contrast, unsupervised learning algorithms are not provided with any labelled data or correct results during the training phase. Instead, they must discover patterns and relationships within the data to conclude. Two common types of unsupervised learning algorithms are clustering and association. Clustering involves grouping similar instances, such as customer segmentation. The association involves finding associations between variables in the data, as seen in "customers who bought this also bought this item…" recommendations on websites (James et al., 2013).

The research in this thesis is based on supervised learning. This category can be further divided into classification and regression settings. In a classification setting, the output variable is categorical. For instance, in this study, we are performing a classification with two binary outcomes, where the model's prediction is either "1" for increase or "0" for decrease. In a regression setting, the output is a continuous value, such as predicting the exact dollar amount of return or loss (James et al., 2013).

In statistical learning, inputs are often referred to as predictors or independent variables, while in machine learning the term 'feature' is preferred. The output is often called the response- or dependent variable, while in machine learning it's commonly called the target variable. For clarity, we will refer to these variables as the target variables and features. Features is typically denoted as the symbol X consisting of p vectors $x_1, x_2, ..., x_p$ and the output variable as Y (James et al., 2013).

In supervised learning, the relationship between the input features and the target variable is often represented by equation 10

$$Y = f(X) + \varepsilon \tag{10}$$

Where Y is the target variable which we want to predict, X is the features we use to predict Y, ε is the error term, which is the difference between the predicted and actual value of Y, and f is the function that maps input X to output Y. This function is what we try to approximate in machine learning (James et al., 2013).

Two fundamental concepts in machine learning are prediction and inference. Prediction involves the use of a model's predictive power to make accurate forecasts on unseen data. In contrast, the inference is about interpreting the behaviour of the model and deriving insight from it. These two aspects of machine learning play distinct yet integral roles in understanding and applying machine learning techniques (Kilskar, 2020).

3.2.1 Evaluation and model assesment

A common method for dataset splitting involves allocating percentages, such as a 70-30 split, where 70% of the data is used for training and 30% for testing. While this approach allows for a quick assessment of the model's performance, more robust evaluation often relies on cross-validation techniques. Cross-validation is a key concept in model assessment, where the dataset is partitioned into multiple subsets for training and testing, ensuring the model's reliability across different data subsets (James et al., 2013).

Cross-validation can take various forms, including k-fold cross-validation, stratified cross-validation, and leave-one-out cross-validation. Each technique has its unique advantages and is used based on the specific scenario (James et al., 2013).

Another essential concept in machine learning evaluation is the bias-variance trade-off, which emphasizes the importance of balance between a model's simplicity and complexity. Simplistic models tend to make strong assumptions and therefore suffer from underfitting. This means that the model is not capturing underlying patterns in the data, which results in high bias and poor performance on both training and test data. Conversely, complex models, which make fewer assumptions about the data, suffer from overfitting. This means that the model captures noise with the underlying pattern, resulting in high variance. While this leads to good performance on the training data, it often results in poor performance on the test data (James et al., 2013).

The goal in machine learning is to create a model that generalizes well to new, unseen data. Achieving this entails finding the right balance between bias and variance. Techniques like cross-validation, as previously discussed, and regularization are commonly employed to achieve this balance (James et al., 2013).

Model assessment involves a variety of metrics. For classification problems, these include accuracy, specificity, precision, and the F1-score. These metrics are invaluable for comparing the performance of different models within a single study or across multiple studies. In classification problems, we are typically provided with a confusion matrix, which allows us to calculate these measurements.

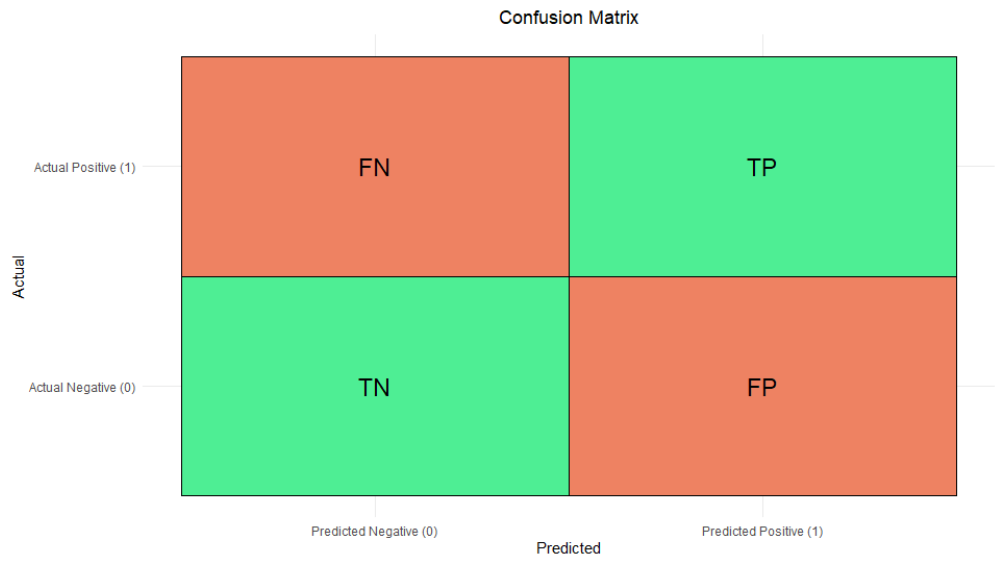


Figure 2: Illustration of Confusion Matrix

Note: The confusion matrix in Figure 2 illustrates the different outcomes resulting from a prediction process, where FN = False Negative, FP = False Positive, TN = True Negative, and TP = True Positive classifications. The color green represents the preferred outcomes where TP is the most favorable, and red represents the outcomes that are least preferred, where FP is the least favorable.

An confusion matrix often resembles Figure 2 where the abbreviations TP, FP, FN and TN represent True Positive, False Positive, False Negative, and True Negative, respectively. In scenarios where the prediction is either True Positive (TP) or True Negative (TN), the prediction is correct. Conversely, if the prediction is either False positive (FP) or False Negative (FN), the prediction is incorrect (James et al., 2013).

When predicting returns, our goal is to maximize the number of True Positive (TP), where we correctly predict a price increase, and True Negative, where we correctly predict a price decrease and thus avoid investing in that particular stock. This scenario is optimal for generating returns (James et al., 2013).

However, we aim to minimize the number of False Positive (FP) and False Negative (FN), where we incorrectly predict the price direction. A false negative scenario, where we fail to invest in a sock that increases in price, and a False Positive scenario, where we invest in a stock that decreases in price, are considered the most detrimental. These are scenarios we strive to avoid as they result in financial loss (James et al., 2013).

Accuracy is the ratio of the number of correct predictions to the total number of predictions. Mathematically it's defined in Formula 11:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{11}$$

It's favourable to have a high accuracy value, as it indicates that the model provides reliable predictions.

Anything above the probability of a coin toss (50%) is considered good.

Precision is the ratio that compares the true positive observations to the total number of positive observations. Mathematically it's defined in Formula 12:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}} = \frac{\text{TP}}{\text{TP + FP}} \tag{12}$$

It's favourable to have a high precision value, as it indicates that the model correctly predicts positive events and avoids false positive predictions. Everything above the probability of a coin toss is considered good.

Recall, also known as sensitivity, is the ratio that compares the true positive observations to all the observations in the predicted class (TP + FN). Mathematically it's defined in Formula 13:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} = \frac{\text{TP}}{\text{TP + FN}} \tag{13}$$

It's favourable to have a high recall value, as it indicates that the model correctly predicts positive events and avoids false negative predictions. Everything above the probability of a coin toss is considered good.

F1-score takes the harmonic mean of *precision* and *recall* as an attempt to find the balance between these two ratios. Mathematically it's defined in Fomula 14:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision}\ \ \text{Recall}}{\text{Precision + Recall}} \tag{14}$$

It's favourable to have a high F1-score, as it indicates a harmonic mean between precision and recall. Everything above the probability of a coin toss is considered good.

Formula 11, 12, 13 and 14 is collected from James et al. (2013); Bodie et al. (2021) and Kyurkchan (2020).

3.3 Random Forest

Random Forest (RF) is an ensemble machine-learning algorithm that leverages decision trees as its fundamental building blocks. Decision trees, known for their tree-like structure, are often visualized with nodes, making the model's predictions easy to interpret. This interpretability contributes to the popularity of this algorithm (James et al., 2013; Dash, 2022).

Decision trees can be employed in both classification and regression tasks, and they excel at capturing nonlinear and complex interactions between target variables and features, a task that linear models often struggle with. Figure 3 provides a visual representation of a simple decision tree, which comprises a root node (the starting point), a decision node (an internal node that split the data into different subsets based on specific feature conditions), and leaf nodes (the endpoint of the tree that represent the final prediction

and does not split any further) (Dash, 2022).

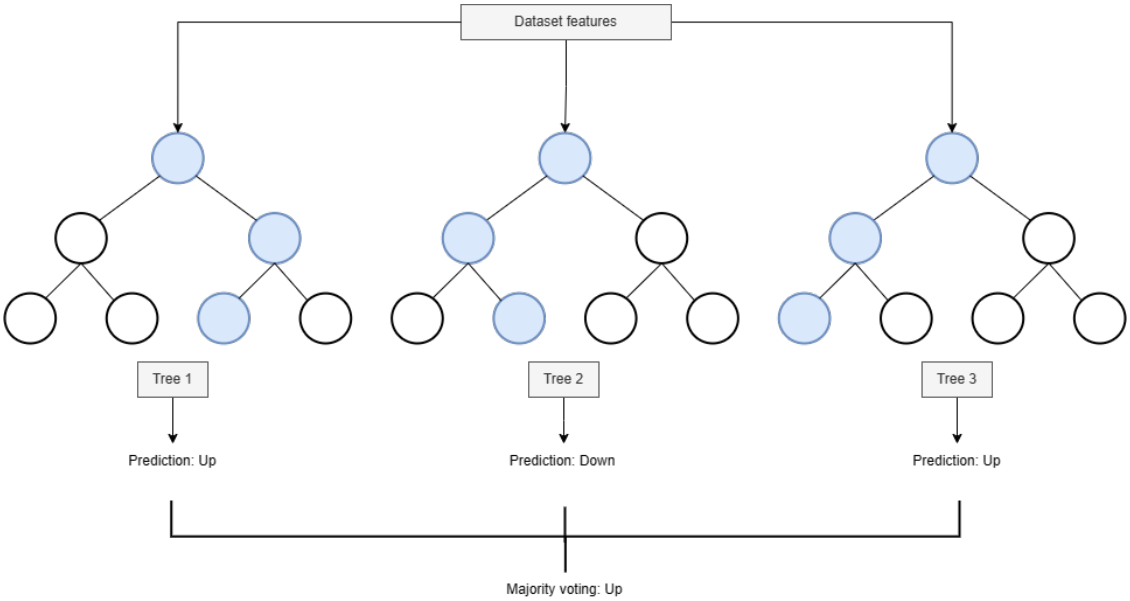


Figure 3: Illustration of Decision Tree

In Figure 3 we see three simple decision trees. Each tree begins at a root node and moves to a decision node based on a predetermined threshold. This threshold is used to decide which path to follow, leading us to the leaf node where the final decision for the individual tree is made. As depicted in the figure, the final prediction is determined by the majority of votes from the individual trees. In this case, the majority of votes from these decision trees is 'Up', therefore the final prediction is 'Up'. In the context of regression tasks, the final prediction is the average of all the individual predictions. (James et al., 2013; Dash, 2022).

Both regression trees and classification trees operate similarly. The key difference lies in the predetermined threshold used in classification to decide the split between the decision nodes and the root of the leaf node (James et al., 2013).

When working with decision trees, it's crucial to determine the optimal tree size to avoid common issues such as overfitting and underfitting. Overfitting occurs when the decision tree is overly complex, causing the model to capture noise in the data rather than the underlying pattern. While overfit models perform well in training data, they tend to perform poorly on new, unseen data (James et al., 2013).

Conversely, underfitting occurs when the model is too simplistic to capture the complexity of the data, resulting in poor performance on both training and test data. Increasing the complexity of the model can help address underfitting (James et al., 2013).

Decision trees, as initially mentioned, offer several advantages. They are known for their interpretability, simplicity, and flexibility. They can handle both numerical and categorical data and are often used for

illustrative purposes due to their tree-like structure. However, they may not match the prediction accuracy of more complex models. Ensemble methods, such as random forest, trade some interpretability for improved accuracy (James et al., 2013).

Like most machine learning models, decision trees also have their disadvantages. We have already discussed the issues of overfitting and underfitting, which are essential considerations in machine learning. These issues are often discussed in the context of the bias-variance trade-off theory (James et al., 2013).

Another significant concern is the consequence of mispredictions, particularly in the context of classification trees. This is often represented by the confusion matrix. The most dramatic misprediction, known as a False Positive (FP), occurs when we predict a stock increase, but the actual outcome is a decrease. A substantial number of FP predictions can result in a poorly performing portfolio and, in the worst case, substantial losses. A slightly less dramatic but still significant misprediction is a False Negative (FN), where we predict the stock to decrease, so we do not invest in the stock, but the stock increases, causing the portfolio to miss out on potential returns (James et al., 2013).

Decision trees can struggle with multiway splits, leading to overfitting. They have high variance, meaning small input changes can cause large output changes. Their prediction surface isn't smooth, and they find it challenging to model additive structures where the target variable is a combination of several inputs (James et al., 2013).

Traditional simple decision trees have been superseded by more efficient algorithms, one of which is the ensemble algorithm known as Random Forest (RF). This algorithm bears a strong resemblance to the decision tree algorithm previously discussed, with the key distinction being that Random Forest is constructed using a random subset of features (Ballings et al., 2015).

To comprehend how this algorithm functions, it is essential to define two central concepts: *bootstrapping* and *bagging*. Bootstrapping refers to the model's generation of a random subset of variables from the original dataset. This process involves sampling data points with replacement, which allows for some observations to be repeated while others may not appear at all. The random selection of features for each tree aids in preventing overfitting and encourages diversity among the trees, marking the training phase of the model (James et al., 2013).

Subsequently, the algorithm amalgamates the trees into an ensemble using an approach known as bagging or bootstrap aggregation. Bagging enhances the model's performance by reducing variance by averaging the predictions derived from multiple trees (James et al., 2013).

During the test phase, the model evaluates all the trees it has constructed, runs unseen data or observations through each tree, and gathers their predictions. In a regression context, the model averages all the trees' predictions. In a classification context, the final class label is determined by the majority vote among the tree's predictions (James et al., 2013).

In the field of machine learning, Random Forest (RF) has emerged as one of the most effective algorithms. According to a study by Ballings et al. (2015), it was identified as the leading algorithm for predicting stock prices, with the Support Vector Machine (SVM) algorithm following as a primary contender among common algorithms. Its robust performance makes it a promising candidate for analyzing the relationship between ONR and market dynamics. Additionally, it offers the advantages of handling nonlinear relationships and not requiring variable transformation before applying RF (James et al., 2013). Random Forest is renowned for its ability to produce accurate predictions and is considered one of the top-performing algorithms in various domains, including finance (Ballings et al., 2015). Despite the lack of consensus on the superior algorithm, Ballings et al. (2015) concluded that Random Forest outperforms most common algorithms, surpassing the second-best algorithm, SVM, by a significant margin. This conclusion is supported by studies on machine learning algorithms by Patel et al. (2015). Conversely, Kumar (2006) found that SVM slightly outperforms RF. Nevertheless, RF is deemed particularly effective for tasks of this nature.

3.4 Logistic Regression (GLM)

Logistic regression, a key component of supervised learning, is typically represented by an S-shaped curve. This model is predominantly employed in binary classification problems, leveraging a logistic function to establish a threshold. This threshold subsequently classifies the outcome into binary digits, either 0 or 1 (James et al., 2013).

The logistic regression model finds extensive application in various fields where binary outcomes are common. For instance, it can be used to predict the presence or absence of a particular health condition, the likelihood of a customer making a purchase, or the probability of failure of a given process or product (James et al., 2013).

The formula for logistic regression is as follows:

$$\Pr(G = K \mid X = x) = \frac{1}{1 + \sum_{y=1}^{K-1} \exp(\beta_{y_1} + \beta_y^T x)} \tag{15}$$

Where $\Pr(G = K \mid X = x)$ is the probability that the outcome variable G is of class K given the features $X = x$, β_{y_1} and β_y^T are the parameters of the model which are learned from the training data.

Logistic regression is frequently employed in academic research and is often used for its simplicity, interpretability, and efficiency. The model's simplicity is evident in the straightforward interpretation of predictor coefficients as odds ratios, a feature that is particularly advantageous when the primary objective is data comprehension. Logistic regression outputs probabilities, thereby providing a nuanced perspective on predictions. Instead of merely predicting the occurrence of an event, it estimates the probability of the event's occurrence. The model's computational efficiency makes it an attractive choice in scenarios where computational resources are constrained. Logistic regression exhibits robustness to noise and can mitigate

overfitting through appropriate model parameter selection. Its versatility is demonstrated by its ability to handle both continuous and categorical variables, including ordinal and nominal data. Due to its simplicity and performance, logistic regression often serves as a baseline model, offering a useful benchmark for comparing the performance of more complex models in various machine learning tasks (James et al., 2013).

However, despite its numerous advantages, logistic regression is not without limitations. The model presupposes a linear relationship between the logit of the response and the predictor, an assumption that may not be valid for all datasets. The model's performance may be adversely affected if the predictors exhibit high levels of multicollinearity. Additionally, logistic regression is sensitive to outliers, which, if not properly addressed, can result in biased estimates. These limitations necessitate careful consideration when selecting logistic regression as the analytical method for a given research problem (James et al., 2013).

4.0 Methodology and data

This chapter focuses on the data and methodology applied in this study. Our study predicts the direction of intraday return for several stocks. Where the aim is to investigate if overnight return can contribute to better predictions of intraday return. We apply two machine learning methods: Random Forest (RF) and Logistic Regression (GLM) in a classification setting. Each method has two models (1) with overnight return as features and (2) a model without overnight return included. In total, we will be comparing 4 models on machine learning metrics and 4 portfolios on financial metrics to understand how overnight returns influence financial results and the quality of predictions for machine learning models. The 4 portfolios and models will be referred to as Random Forest with overnight return: RF_{ONR}, Random Forest without overnight return: RF, GLM with overnight return: GLM_{ONR} and GLM without overnight return: GLM in the upcoming sections.

Our study follows some of the methodology applied by Kilskar (2020), Gao et al. (2022) and Hansen and Rustad (2023). Similarly to Kilskar (2020) we apply a rolling window approach to train and predict intraday return direction. From Hansen and Rustad (2023) and Gao et al. (2022), we train and predict using two machine learning models; Random Forest and GLM to compare performance and include overnight returns as features to capture if overnight returns can contribute to better financial results. Overall, our study follows these steps:

1. Collect and structure data
2. Create and define formulas for each model
3. Create a rolling window for training and test sections.
4. Model specifications
5. Construct portfolios
6. Evaluate models
7. Limitations

4.1 Data

Our study covers daily stock data of all stocks listed on the Oslo Stock Exchange (OSE) in the period from 01.01.2010 to 27.11.2023 downloaded from the Titlon database. Daily stock data consists of: open, close, high, low, volume and the number of trades. For the total period, we have 315 individual stocks. Additionally, we include global variables that apply to all stocks which consist of: SMB, LIQ, MOM, OSEBX log returns, OBX log returns, Norwegian Overnight Weighted Average (NOWA) rate log returns and Norway 3 months Bond log returns. We add macro variables to each stock similarly to the global variables that consist of: SPY ETF, GLD ETF and USO ETF. From the daily stock data, we filter out stocks that have prices below 5 NOK/share and above 1000 NOK/share to exclude potential noise from penny stocks and high-price stocks from our portfolio construction. Kilskar (2020), argues that penny stocks can contribute to extraordinary

returns as they can be influenced by smaller events. Summary statistics for global and macro variables are provided in Table 10 presented in Appendix A.

The inclusion of SPY which tracks the S&P 500 index, allows us to gather movements in one of the largest and most influential markets globally. GLD which tracks gold prices has proven to move inversely when markets experience drawdowns and is often considered a safe haven (Ryan et al., 2024). Norway and OSE are often considered to be an energy-rich Stock Exchange with several companies that either extract crude oil or offer services to the domestic oil industry. We include the USO ETF which covers crude oil prices primarily for the US crude oil type WTI. Although Norwegian companies usually extract and trade Brent rather than WTI studies have proven that there exists a tight spread between the two (Ruble & Powell, 2021). Overall, we include daily stock data along with attributes such as the number of trades and the daily volume. The global variables allow us to capture changing market conditions in Norway, while the macro variables allow us to capture changing market conditions globally.

4.1.1 Programming and missing values

In this study, we use the R software by R Core Team (2023) to train our machine learning models and perform the analysis. Specifically, we use the caret package from Kuhn and Max (2008) to compute the machine learning metrics accuracy, precision, recall and F1-score. We use the ranger package from Wright and Ziegler (2017) to train and predict the Random Forest models. We apply the stats package provided within the R software by R Core Team (2023) to train and predict the Logistic Regression (GLM) models. GGplot2 by Wickham (2016) to visualize our results as plots, stargazer by Hlavac (2022) to display regression results in formatted tables and the PerformanceAnalytics package by Peterson and Carl (2020) to calculate portfolio metrics such as Sharpe Ratio, Sortino Ratio, Treynor Ratio and Information Ratio.

To account for missing values in our dataset we incorporate a forward-fill method where the last known observation is used for the missing value. Additionally, if the last known observation is also a missing value, we replace the missing value with zero until a new observation is known. Finally, we compute the rolling volatility for 10 days, 20 days and 40 days meaning the first 40 observations in the dataset are excluded altogether.

4.2 Feature construction and selection

As our aim is to understand the influence of overnight returns by predicting the direction of intraday returns, we will compute the target variable and several features for each stock, such as intraday return (IDR), overnight return (ONR), total return (TR), intraday volatility (LnHiLo) and a rolling k day volatility (vol_{kd}).

To account for stock splits, dividends and other adjustments to stock prices over time we need to use adjusted prices. If we do not account for adjusted prices we would typically experience large drawdowns if a company distributes dividends. In our initial dataset from Titlon we were not supplied with adjusted prices for High,

Low or Open only for the Close price for each stock. For this reason, we compute the adjusted prices manually for High, Low and Open represented by $Price_{adj,t}$ by dividing the adjusted close price supplied by Titlon $Close_{adj,t}$ on the actual close price $Close_t$ and multiply with the base price, represented by $Price_t$ in Formula 16.

$$Price_{adj,t} = \frac{Close_{adj,t}}{Close_t} * Price_t \tag{16}$$

With the adjusted prices we can compute the adjusted returns for each stock. Overnight Returns (ONR) and Intraday Returns (IDR) are the two elements that make up the traditional Total Return (TR) as shown in Figure 1 introduced in section 1.

Adjusted total returns (TR) are defined as the relative change between $Close_{adj,t}$ and $Close_{adj,t-1}$ as shown in Formula 17:

$$TR_{adj,t} = \frac{Close_{adj,t} - Close_{adj,t-1}}{Close_{adj,t-1}} \tag{17}$$

Adjusted intraday return (IDR) are defined as the relative change between $Close_{adj,t}$ and $Open_{adj,t}$ as shown in Formula 18:

$$IDR_{adj,t} = \frac{Close_{adj,t} - Open_{adj,t}}{Open_{adj,t}} \tag{18}$$

Adjusted overnight returns (ONR) are defined as the relative change between $Open_{adj,t}$ and $Close_{adj,t-1}$ as shown in Formula 19:

$$ONR_{adj,t} = \frac{Open_{adj,t} - Close_{adj,t-1}}{Close_{adj,t-1}} \tag{19}$$

Intraday volatility (LnHiLo) are defined as the logarithmic difference between the adjusted $High_{adj,t}$ price and the adjusted $Low_{adj,t}$ price as shown in Formula 20:

$$LnHiLo_{adj,t} = log(\frac{High_{adj,t}}{Low_{adj,t}}) \tag{20}$$

Rolling 10, 20 and 40 day volatility from time T_1 to time T_2 is calculated using Formula 21. Where R_i is the return on day i. $\overline{R}_{T,k}$ is the average return for the k days ending at time T (the rolling mean return). k is the rolling window size (10, 20 or 40 days). While T_1 is the starting time for the overall period and T_2 is the ending time for the overall period as shown in Formula 21:

$$\text{Rolling k day volatility}_T = \sqrt{\frac{1}{k-1}\sum_{i=T-k+1}^{T}(R_i - \overline{R}_{T,k})^2} \tag{21}$$

As we predict daily intraday return directions for each stock, we risk including features that may present look-ahead-bias in our models. To account for this, we ensure features are lagged from 1 to 3. Specifically, high, low, close, volume, trades, IDR, TR, LnHiLo, vol_10d, vol_20d, vol 40d, all global features and all

macro features are lagged three times.

4.3 Rolling window

Common split ratios between training and test datasets in machine learning are 60:40, 70:30 and 80:20 (James et al., 2013). In this study, we apply a rolling window approach similar to Kilskar (2020) to ensure cross-validation across different time periods. This method also allows us to capture changing markets as new technologies and financial theories are discovered. Additionally, incorporating a rolling window approach allows us to include all listed and delisted stocks on the Oslo Stock Exchange in the period. This means we mitigate a possible survivorship-bias in our stock selection. The training period in this study is over two years while the test or trading period is over one year. The preceding training period in the next fold incorporates the test set from the previous fold as visualized in Figure 4.

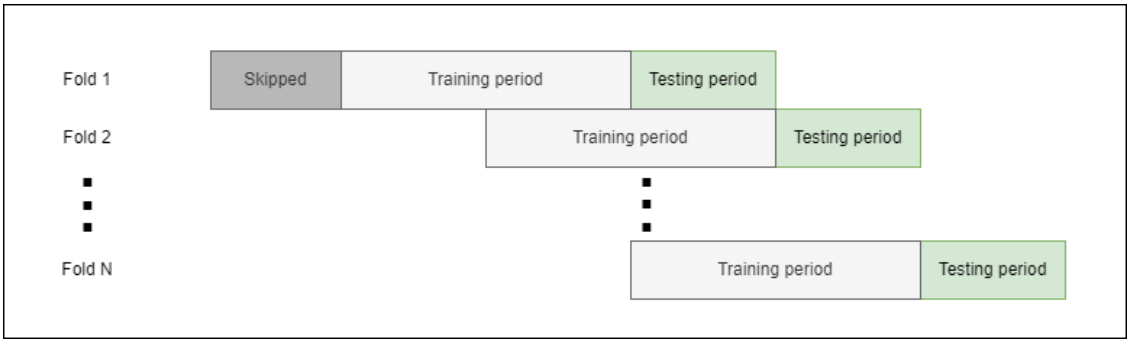


Figure 4: Illustration of Rolling Window

4.4 Model specifications

Both Random Forest (RF) and Logistic Regression (GLM) are trained on the same dataset using the exact same stocks over the same period. Each stock is trained separately for the training period using Formula 22 and Formula 23.

$$\text{Formula with ONR: Target} \sim f(Stock, Global, Macro, ONR) \tag{22}$$

$$\text{Formula with ONR: Target} \sim f(Stock, Global, Macro) \tag{23}$$

Random Forest has shown a flattening in performance as the number of trees increase where the ideal number of trees is considered to be around 500 (Kilskar, 2020; James et al., 2013). Additionally, in this study we make predictions for 630 stocks across two models ($630 = 315 * 2$) which are computationally expensive. With this in mind, we applied the ranger package in R that works well with high dimensional data (Wright & Ziegler, 2017).

Each model generates predictions for all available stocks each day in our test set. The prediction outputs for each model are probabilities of either increase (1) or decrease (0) in the intraday return for any given

stock. Probabilities range between 0-1 or 0%-100%. When each model predicts either an increase or decrease in the direction of intraday returns, we extract the top k stocks each day with the highest probability for an increase. The identified stocks with the highest probability are selected and passed on to the portfolio construction.

4.5 Portfolio construction

For each model, a portfolio is constructed. Random Forest (RF) has two models: one with overnight return (ONR) and one without ONR. Logistic Regression (GLM) has the same specification. Meaning we have four portfolios: RF_{ONR}, RF, GLM_{ONR} and GLM. All four portfolios are constructed using the same requirements/rules: (1) long-only, (2) intraday holding period, and (3) top k stocks chosen. We chose to use long-only for a less complex implementation given the size and complexity of our study. As we are predicting the direction of intraday returns, we chose an intraday holding period. Even though a model will recommend the same stock two days in a row we will buy and sell the stock on day 1 and repeat for day 2.

The number of stocks that we buy and sell each day in each portfolio is set at k = 10. Additionally, we incorporated an evenly weighted portfolio, meaning the 10 stocks with the highest probability of an increase get assigned a 10% weight in the portfolio. Initially, when starting our analysis our intention was to use the probabilities of increase to create weights for each stock in our portfolios. As we have 315 individual stocks the result from probability weighting was minor. We got a list of stocks where each stock had a maximum weight of 10.5% in the portfolio. This means the top 10 best probabilities were almost evenly weighted anyway. Our choice of settling on k = 10 as the number of stocks in the portfolio and the evenly weight of stocks in our portfolios can be split into three parts. (1) Kilskar (2020) generates higher returns when applying a set of 10 stocks in her portfolio compared to 5 and 15. (2) This thesis aims to understand the role of overnight returns in predicting the direction of intraday returns and has a strategic focus on the role of overnight returns rather than portfolio optimization or diversification. (3) Given our strategic focus we applied a practical approach to portfolio management settling on evenly weighted portfolios with empirically proven k = 10. Overall, this approach allowed us to scale our methodology across several machine learning models with a focus on the differences between the models merely the inclusion of overnight returns.

4.6 Model evaluation

To evaluate how our models perform in a machine learning setting we measure their accuracy, precision, recall and F1-Score. Comparing each model against each other these metrics allow us to understand how well they predict. Since the models are trained on the same dataset using the same two formulas (1) with overnight return and (2) without overnight return we can compare them directly to understand if the models with overnight return perform better than those without overnight return. Additionally, we will compute the variable importance through the Gini impurity measure for the stocks that contribute to the highest and

lowest returns in the portfolio with the highest returns. This allows us to understand if overnight returns as a feature have an impact on generating portfolio returns.

4.7 Limitations

Our study does not account for some factors associated with machine learning. Specifically, (1) feature engineering which would allow us to test several variables before including them in the final models (James et al., 2013). (2) Hyperparameter tuning of Random Forest and Logistic Regression (GLM) (James et al., 2013). (3) Evaluation of several other machine learning models such as Neural Networks see Long et al. (2019), Deep Learning see Gao et al. (2022) or other ensemble methods see James et al. (2013).

As for portfolio construction and optimization, we do not investigate if the number of stocks in each portfolio would alter the performance, or if combinations of the four portfolios could contribute to even better returns or create maximum return and minimum risk portfolios. Our financial results do not account for trading costs, or bid-ask spreads and assume it is possible to purchase at the opening price and sell at the closing price. Liquidity and the time from prediction result to order execution would introduce bid-ask spreads, slippage and some time lag from when the stock exchange opens and the trades are executed.

Additionally, since we use the open price both as a feature itself and within overnight return (ONR) in all models we would need to collect the opening price for each stock to make predictions. Realistically, the time from collecting the opening prices to the execution of trades would not allow us to trade at the open price. As stocks can move either sideways, up or down we found it difficult to account for all the scenarios. A potential solution would be to set a percentage based on historical price movements in the first minutes of trading that either decreases or increases the opening price. The problem with this approach is that altering historical stock prices would greatly impact the predictions and robustness of our model which can lead to overfitting (James et al., 2013).

The same problem arises for the closing price as for the opening price. This study takes each stock's intraday returns in the portfolio to calculate the daily return of the portfolio. As the intraday returns are calculated using the opening and closing prices this study assumes investors can purchase at the opening price and sell at the closing price. In reality, this is unrealistic. Investors would experience higher or lower returns based on the exact time they would purchase and sell each stock. High-frequency trading or algorithmic trading has become more popular as technology allows investors to execute transactions in milliseconds. A full technical implementation of this trading strategy can account for the lag between collecting the opening price and the execution of trades but will not account for slippage. Therefore, we accept that the theoretical returns the portfolios display in this study are unrealistic and hard to replicate in practice. An adaptation of this trading strategy would most likely yield less returns as the investor would experience bid-ask spreads, trading costs, slippage and different returns as a result of the limitations in this study.

5.0 Analysis and results

In this section, we analyze our findings which we have split into two main parts (1) Portfolio results and (2) Machine learning results. We look at individual differences between each portfolio and model to understand the differences between the inclusion and exclusion of overnight returns.

First, we present the cumulative and rolling annualized returns for each portfolio compared against the benchmark $OSEBX$. Cumulative returns allow us to understand how each portfolio develops over time and visualize cyclical phases. Rolling annualized returns allow us to understand the volatility of returns split into trading years of 252 days. Continuing the analysis on returns we compute correlations between the portfolios and the benchmark. This allows us to understand if the portfolios tend to behave in the same way and if they follow the market. We then compute common annualized portfolio metrics such as: Return, Volatility, Sharpe Ratio, Sortino Ratio, Information Ratio and Treynor Ratio. Finally, we regress each portfolio returns r_P against the Fama French Five-factor model to understand if the portfolios generate significant positive alpha and how each portfolio correlates against market, size, value, momentum and liquidity factors.

For the Machine Learning section, we compare each model against each other based on: accuracy, precision, recall and F1-score. We investigate the most selected stocks for the model with the highest overall score and the stocks that contribute to the highest returns in the model with the highest returns overall. Additionally, we investigate variable importance for the stock that contributes with the highest and lowest returns in the portfolio with the highest overall return to understand if overnight returns have an impact on returns.

Finally, to end our analysis we investigate correlations between intraday return and overnight returns in combination with the portfolio and machine learning results to understand if there exists an overnight return effect.

5.1 Portfolio results

Our initial dataset spans from 08.01.2010 to 24.11.2023 where the first two years are used for training each model which gives us the total trading period from 08.01.2012 to 24.11.2023. In that time frame the model with the highest cumulative return is RF_{ONR} as seen in Figure 5. The model with the second highest cumulative return is GLM_{ONR} also with overnight returns included as features. We notice a large discrepancy between the Random Forest models compared to the GLM models. The Random Forest model is more affected by the exclusion of overnight returns than the GLM model. Overall, all models beat the benchmark OSEBX and both portfolios containing overnight returns are superior to their counterparts without overnight returns.

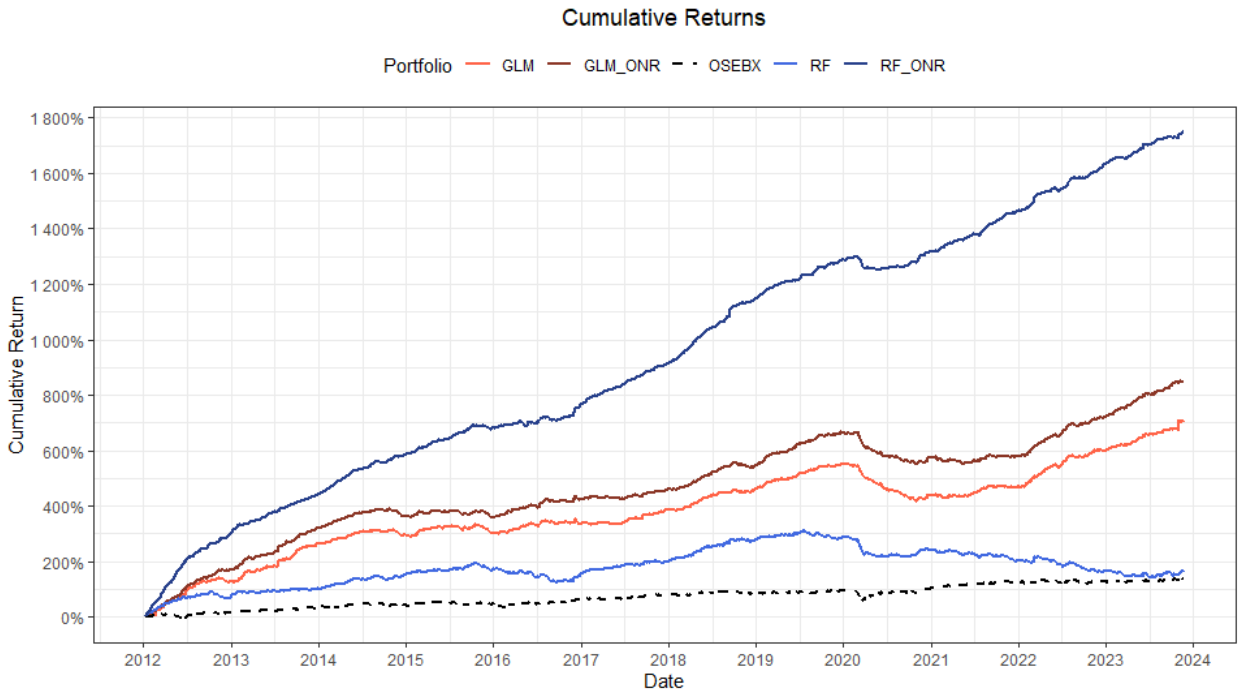


Figure 5: Cumulative Returns

Note: Cumulative returns calculated for each portfolio from 08.01.2012 to 24.11.2023

All portfolios quickly start to generate positive returns as the period starts where RF_{ONR} is the most notable followed by GLM_{ONR}. Worse off is RF, after an impressive start closely following the other portfolios, RF continues to increase at a slower rate midway through 2012-2013. All portfolios continue to increase at varying rates until 2016 when the rate changes. RF starts to take on losses while the other 3 experience a lower return. From 2017 the portfolios continued to accumulate returns until March 2020 when COVID-19 happened, and stock markets took a hit (Taera et al., 2023). RF fails to recover the losses it experienced in the Covid event while both GLM portfolios GLM_{ONR} and GLM are only back to pre-COVID levels midway through 2022. However, Random Forest with overnight returns RF_ONR has recovered a year later and continues to accumulate returns almost linearly until the end of our trading period.

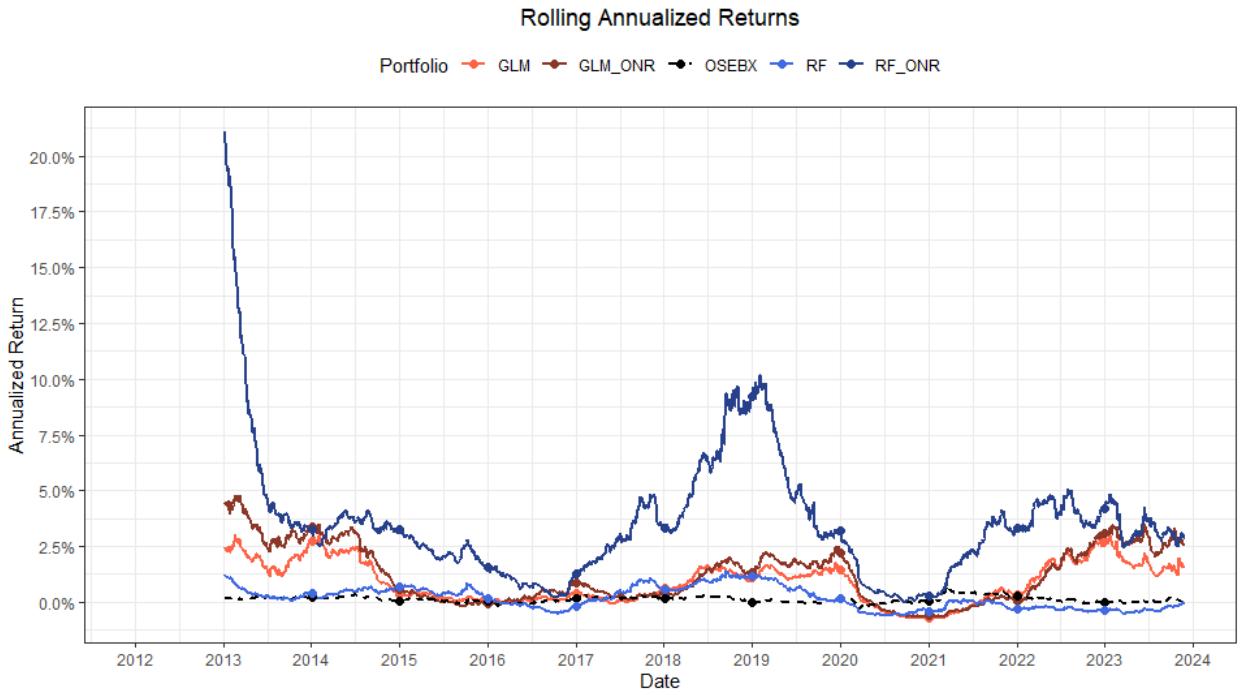


Figure 6: Rolling Annualized Returns

Note: Rolling Annualized Returns calculated for each portfolio are based on 252 trading days in a year.

Rolling annualized returns allow us to capture the relative daily change in annualized return. In Figure 4 we visually show how the portfolios rolling annualized return develop over time. All start off well above the benchmark while Random Forest without overnight return RF rather quickly starts to yield annualized returns close to the benchmark index $OSEBX$. From 2016 to March 2020 Random Forest with overnight return RF_{ONR} has a period with high returns while the 3 others generate returns but at a slower rate. March 2020 to 2022 show how each portfolio was affected by the COVID-19 event and subsequently how Random Forest without overnight returns RF fails to recover.

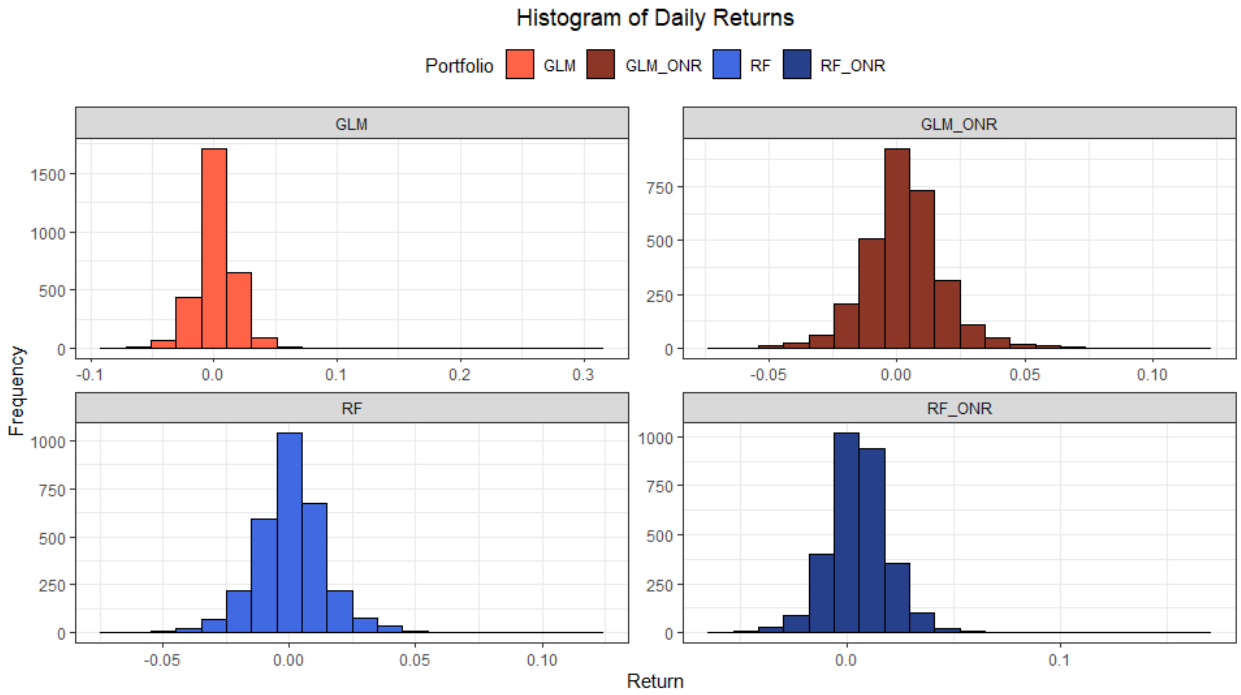


Figure 7: Histograms of daily returns for portfolios

As for the distribution of daily returns we notice a convergence around the mean for all portfolios as seen in Figure 7. Additionally, we notice a positive skew for all portfolio returns where GLM without overnight return GLM has the largest skew as shown in Table 2. A higher positive skew indicates fatter right tails in our portfolios which are considered more uncommon than fat left tails in finance (Bodie et al., 2021).

Table 2: Summary statistics for portfolio daily returns

	RF_ONR	RF	GLM_ONR	GLM
Mean	0.01	0.00	0.00	0.00
SD	0.02	0.01	0.02	0.02
Min	-0.06	-0.07	-0.07	-0.07
Max	0.16	0.12	0.12	0.31
Median	0.01	0.00	0.00	0.00
Skewness	1.01	0.48	0.59	2.37
Kurtosis	8.46	5.27	4.43	39.32

Both GLM models GLM_{ONR} and GLM do correlate 0.29 with the market benchmark $OSEBX$, while Random Forest without overnight returns RF, which is the worst performing model in terms of cumulative returns has the highest correlation against the market. This confirms our visual interpretation of the cumulative and annualized returns during the period, that the worst performing portfolio Random Forest without overnight return RF cannot recover after setbacks and reverts towards the market while the others continue to generate returns.

Table 3: Correlation matrix between portfolios and the market

	RF_ONR	RF	GLM_ONR	GLM	OSEBX
RF_ONR	1				
RF	0.72	1			
GLM_ONR	0.41	0.36	1		
GLM	0.38	0.33	0.77	1	
OSEBX	0.33	0.38	0.29	0.29	1

5.1.1 Portfolio metrics

By looking at the cumulative return for each portfolio from section 5.1 we understand how well each portfolio performed in the 12-year period. Cumulative returns, however, do not give us an indication of how each portfolio handles risk. To capture how each portfolio handles risk we measure annualized Sharpe Ratio and Sortino Ratio. Additionally, we compute the Information Ratio and Treynor Ratio to look at active return and excess return. Investors naturally seek high returns and minimal risk. Table 4 shows that the Random Forest portfolio with overnight return RF_{ONR} is the best portfolio when comparing Sharpe Ratio, Sortino Ratio, Information Ratio and Treynor Ratio. Additionally, we see that both portfolios with overnight return RF_{ONR} and GLM_{ONR} outperform the portfolios without overnight return RF and GLM respectively. All portfolios beat the market index $OSEBX$ in terms of annualized return. However, risk-adjusted return measured by the Sharpe and Sortino ratios shows that the market index outperforms Random Forest without overnight return RF.

Table 4: Portfolio metrics annualized

Metrics	RF_ONR	RF	GLM_ONR	GLM	Market
Annualized Return	323.37	11.74	97.90	74.52	10.75
Annualized Volatility	24.32	23.36	25.24	27.26	16.85
Annualized Sharpe Ratio	12.79	0.36	3.65	2.55	0.44
Annualized Sortino Ratio	12.33	0.69	4.62	3.55	0.71
Annualized Information Ratio	201.86	0.68	53.38	36.78	NaN
Annualized Treynor Ratio	103.53	2.57	33.25	23.22	1.19

Note:
Table shows the annualized portfolio metrics for the 4 portfolios and the market benchmark in percent. A risk-free-rate of 3 percent and 252 trading days within a year is used.

Sharpe Ratio typically measures the return for every unit of risk the investor has taken. According to Bodie et al. (2021) a Sharpe Ratio greater than 1 is considered acceptable while 2 is very good and 3 is excellent. Random Forest with overnight returns RF_{ONR} and both GLM models GLM_{ONR} and GLM achieve an excellent result. The Sortino Ratio similar to the Sharpe Ratio emphasizes downside risk and has the same grading scale. Similarly, all three portfolios that performed excellently measured by the Sharpe Ratio performed excellently also when measuring their Sortino Ratio. However, Random Forest without overnight return RF are outperformed by the market benchmark $OSEBX$ on both metrics.

Information Ratio provides insight into the risk-adjusted return relative to its benchmark (Bodie et al., 2021). An Information ratio above 1 is considered exceptional by investors (Bodie et al., 2021). Our results show that the same portfolios with excellent Sharpe and Sortino ratios are also exceptional in terms of the Information Ratio. Again, we see Random Forest without overnight returns RF underperforming relative to its peers. The Treynor ratio tells us how much excess returns were generated per unit of portfolio risk (Bodie et al., 2021). Interpretation usually suggests a higher value is better. Again, we see the three portfolios outperforming Random Forest without overnight returns RF meaning we can subsequently rank each portfolio from best (1) to worst (4) based on the portfolio performance metrics:

1. Random Forest with overnight returns
2. GLM with overnight returns
3. GLM without overnight returns
4. Random Forest without overnight returns

5.1.2 Fama French Factors

Fama French factors can contribute to understanding how portfolios correlate against the market, size, value, momentum and liquidity factors as they aim to explain the average returns for stocks (Fama & French, 1993). Since our portfolios rebalance daily and do not hold any positions overnight, we regress our portfolios against the Fama French factors using daily returns in Table 5.

Table 5: Regression Results: Fama French Models

	Dependent variable:			
	RF_ONR	RF	GLM_ONR	GLM
	Random Forest		GLM	
	(1)	(2)	(3)	(4)
Market Premium	0.447***	0.500***	0.424***	0.455***
	(0.025)	(0.023)	(0.026)	(0.029)
Size Premium	−0.298***	−0.268***	−0.198***	−0.251***
	(0.029)	(0.027)	(0.031)	(0.033)
Value Premium	0.029	0.046***	0.038*	0.041*
	(0.019)	(0.018)	(0.020)	(0.021)
Momentum	−0.099***	−0.087***	−0.077***	−0.087***
	(0.020)	(0.019)	(0.021)	(0.023)
Liquidity	−0.031	−0.008	0.022	0.015
	(0.028)	(0.026)	(0.029)	(0.032)
Constant	0.006***	0.0004	0.003***	0.002***
	(0.0003)	(0.0002)	(0.0003)	(0.0003)
Observations	2,968	2,968	2,968	2,968
R^2	0.145	0.173	0.102	0.106
Adjusted R^2	0.143	0.172	0.100	0.105

Note: *p<0.1; **p<0.05; ***p<0.01

Table shows regression results for each portfolio against Fama French Factors. Each portfolio and the Market Premium (Equity risk premium) has been adjusted with a risk-free rate of 3 percent.

Our results show a positive significant alpha for all models except the Random Forest model without overnight returns RF. The Logistic Regression (GLM) portfolios GLM_{ONR} and GLM perform quite similarly as seen in previous sections where GLM_{ONR} perform better than GLM. The same can be said for the regression results where GLM_{ONR} provides a higher significant positive alpha than GLM.

As we have seen in previous sections the overall best model is RF_{ONR} and the trend continues when we regress against each factor. All portfolios have positive significant coefficients against the market premium, meaning higher market premiums provide better portfolio performance. Larger firms perform better than smaller firms in all our portfolios as the size premium is negative and significant. Interestingly, the portfolios are not affected largely by value as we observe significance levels only at 10 percent for GLM_{ONR} and GLM. However, the worst performing model Random Forest without overnight returns RF has a significant coefficient at the 1 percent level. The best model RF_{ONR} are not affected by high-value firms as the coefficient are not significant at 10, 5 or 1 percent levels. Negative and significant momentum factors for all models show that portfolios leaning towards previous losers outperformed those leaning towards winners. Liquidity is not a significant factor for our portfolios as neither coefficient is significant. Finally, we notice each model has an adjusted r-squared below 20 percent indicating there are other factors that would explain the variance in the portfolio returns that we have not captured in these regressions.

5.2 Model evaluation

In section 5.1 we analyzed how each portfolio performed financially and in this section, we focus on how each portfolio or model performs on common machine learning metrics. From each model's confusion matrix, we compute their Accuracy, Precision, Recall and F1-score as presented in Table 6. While subsections 5.2.1 and 5.2.2 cover stock selections, contributors to return and variable importance for the best model from Table 6. Performance metrics for each model allow us to understand the robustness of the predictions. Variable importance in combination with identifying contributors to return and how each stock is selected allows us to understand how overnight returns influence a machine learning model such as Random Forest and GLM. As the total number of stocks in the dataset is comprehensive, we will focus on the top 10 most selected stocks and the stocks with the highest contribution to returns across each fold for subsections 5.2.1 and 5.2.2.

Table 6: Mean performance metrics for each model

Model	Accuracy	Precision	Recall	F1
GLM	57.32	64.96	65.22	62.33
GLM_ONR	57.53	65.04	65.15	62.55
RF	57.58	63.52	70.87	64.04
RF_ONR	60.23	65.66	73.41	66.70

Note:
Table shows the mean performance metrics for each machine learning model. The mean is taken across each stock (315) and each fold (11).

Table 6 shows the Accuracy, Precision, Recall and F1-Score for GLM, GLM_{ONR}, RF and RF_{ONR}. As each stock in our dataset has 4 machine learning models in each fold, we calculated the performance metrics across all 11 folds and have taken the average of those metrics for each model. The Random Forest model containing ONR as features RF_{ONR} are superior to the Random Forest without ONR RF on Accuracy, Precision, Recall and F1-Score. The Logistic Regression (GLM) models GLM_{ONR} and GLM_{ONR} give the same impression, but the difference between the two is minor compared to the Random Forest models. Continuing the trend from previous sections we observe from Table 6 that the best overall model is RF_{ONR}.

5.2.1 Stock selection and contribution to returns

Section 5.1 showed us that overnight return contributes to better financial results and Section 5.2 highlights the impact of overnight returns when evaluating machine learning models on common performance metrics. In this section, we analyze the stock selections and the contributors to return for RF_{ONR}. From Table 7 we see the 10 most selected stocks across each fold and in total across all folds. Since we have a trading window from 08.01.2012 to 24.11.2023 as described in Section 4.3 it means we have 11 folds or trading periods each containing approximately 252 trading days within a year. As our trading period stops in November of 2023 and is a month shy of 12 years the rolling window logic creates an exception for Fold 11 which has 335 trading days as the period starts on 08.01.2022 and ends on 24.11.2023.

Table 7: Top 10 stocks selected by the Random Forest model with ONR

Stock	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Fold 11	Total
OBXEXDBEAR	86	247	2	27	42	4	41	122	32	0	0	603
NANO	0	0	0	0	0	9	8	218	0	0	335	570
MGN	1	14	61	142	114	109	80	20	2	0	11	554
SSO	0	0	0	9	1	0	4	10	0	182	332	538
EMGS	1	35	50	48	164	83	51	2	11	24	12	481
GOD	57	33	35	4	26	136	50	0	2	0	75	418
PGS	23	25	14	24	226	88	1	2	4	8	0	415
FRO	1	131	10	2	5	216	26	2	1	14	6	414
ODF	118	0	26	51	15	73	0	74	41	0	9	407
PDR	17	13	2	132	0	12	6	58	20	67	78	405

Note:
Table shows the top 10 stocks selected by the Random Forest model with overnight returns for each fold and in total over the trading period.

OBXEXDBEAR is an ETF that provides inverse returns to the development in the OBX index. Meaning as the index returns decrease the ETF returns increase. The ETF was included in the portfolio 247 times in fold 2 which ranges from 08.01.2013 to 07.01.2014 and 603 times in total over the entire period. Table 7 also shows how some stocks like NANO and SSO are either selected many times within the same fold or not at all, suggesting that the models have picked up a signal for these stocks resulting in a high probability of increase in intraday return. Stocks such as MGN, ODF and GOD are selected more evenly across the folds. Gyldendal (GYL) is the least selected stock across all folds with only one inclusion in Fold 11.

Continuing the analysis of the stocks contained in portfolio RF_{ONR} we present the 10 stocks that contribute with the highest return for RF_{ONR} in Table 8. Interestingly we observe only two stocks from the table containing the most selected stocks in the table with return contributions. Meaning 8 of the top 10 stocks with the highest contribution to the total portfolio return are not among the top 10 most selected stocks. Magnora (MGN) is the stock that has contributed with the highest return in the portfolio followed by OTS and PDR. Seadrill (SDRL) is the stock that has provided the lowest returns (-30%) in the portfolio.

Table 8: Top 10 stocks that contributes the most to the portfolio return for the Random Forest model with ONR in percent

Stock	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Fold 11	Total
MGN	-0.17	0.97	13.14	31.24	34.13	33.75	28.98	4.85	1.39	0.00	3.32	151.60
OTS	0.00	0.00	0.00	22.44	0.00	0.00	46.02	0.00	0.00	30.91	11.34	110.70
PDR	3.18	0.31	0.46	13.76	0.00	4.75	2.17	33.60	10.72	11.34	15.14	95.42
IMSK	15.68	0.00	0.00	0.00	13.82	0.00	33.75	0.00	0.00	0.00	0.00	63.25
NEXT	0.00	0.00	0.00	4.62	0.00	-1.48	0.82	0.00	-0.25	0.00	58.57	62.29
DAT	27.64	19.93	9.73	0.00	0.00	0.00	0.00	0.00	1.46	0.00	0.00	58.75
BIRD	33.26	6.89	0.00	0.18	0.34	2.70	2.19	4.04	0.00	0.00	7.93	57.54
GOD	20.98	2.49	5.54	-0.48	1.37	8.53	5.65	0.00	0.00	0.00	11.57	55.66
SINO	43.97	7.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	51.09
NORTH	1.41	3.68	25.78	-0.52	0.00	1.67	3.25	1.37	0.27	1.75	10.37	49.01

Note:
Table shows the top 10 stocks that contributes with the highest returns for the Random Forest model with overnight returns for each fold and in total over the trading period.

Summarizing Tables 7 and 8 we observe that the number of times a stock is selected does not translate to a high contributor of returns. In fact, only Magnora (MGN) and Petrolia (PDR) are the only stocks from the high return contributors that appear in the most selected table.

5.2.2 Variable importance

To understand why a stock is either the most selected or contributes to the highest returns from subsection 5.2.1 we compute the Gini Impurity measure which allows us to evaluate which variables are the most important features for RF_{ONR}. Figure 8 shows the top 20 most important variables for the stock that contribute to the highest and lowest return within the RF_{ONR} portfolio. For Magnora (MGN) we clearly see that ONR is the most important variable, for Seadrill (SDRL) on the other hand overnight return is only the 13th most important variable. We also see that there are small differences in the Gini Impurity value between each variable for Seadrill compared to the large difference between overnight return (ONR) and the second most important variable for Magnora.

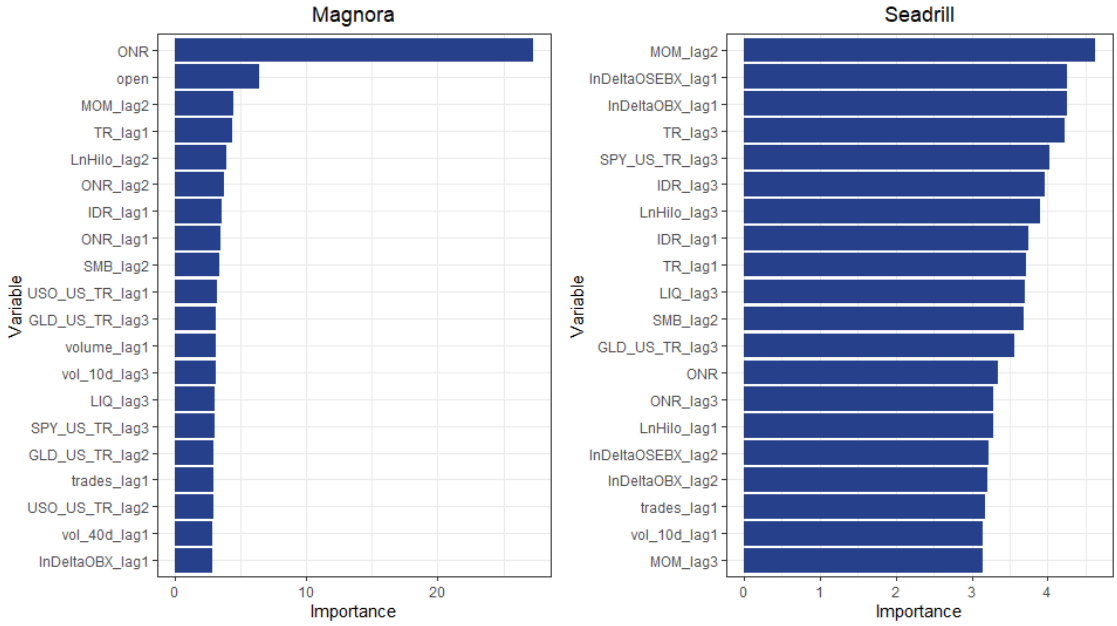


Figure 8: Variable importance for the stocks with highest and lowest returns

From Figure 8 we discovered that overnight returns (ONR) are the most important variable for the stock Magnora (MGN) which is the stock that contributed with the highest return in the RF_{ONR} portfolio. While the stock with the lowest contribution to returns Seadrill, overnight return does not appear to be an important variable. To understand if this phenomenon exists for the other stocks we compute the variable importance for the top and bottom 10 percent selected and contributors in Figure 9.

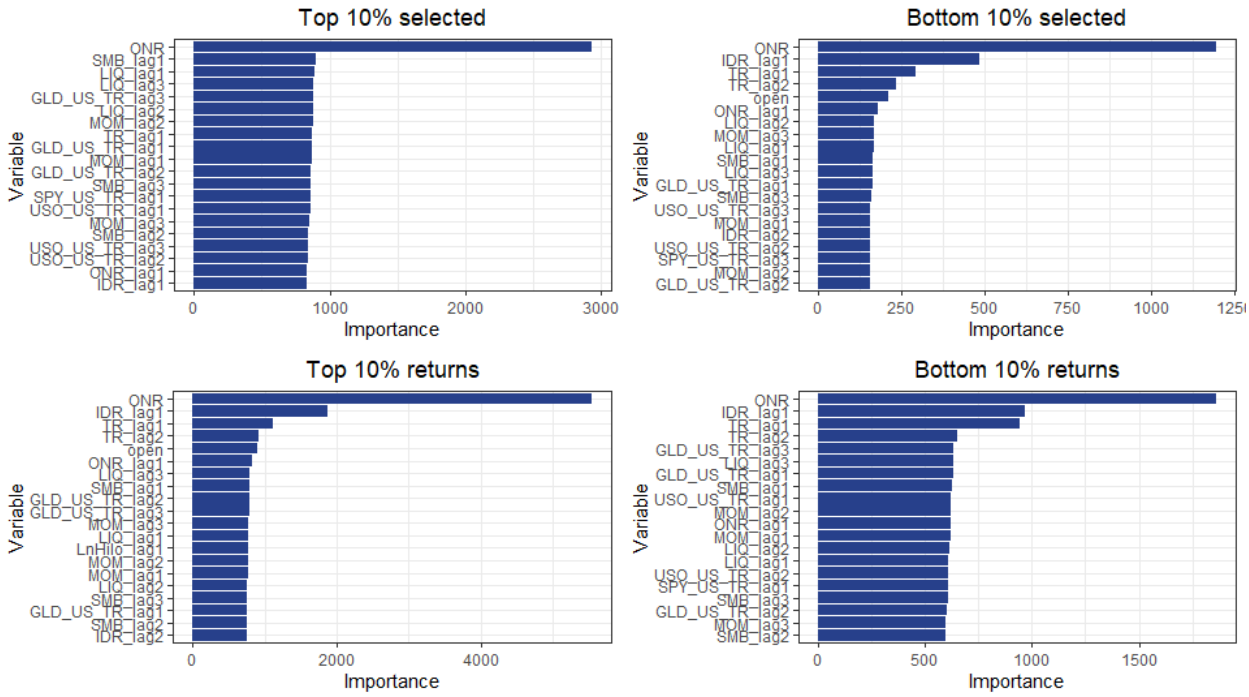


Figure 9: Variable importance for the top and bottom 10 percent stocks

Interestingly, we observe that overnight return (ONR) is the most important variable in the top- and bottom-selected stocks and for the top- and bottom contributors to RF_{ONR} cumulative return. This tells us that overnight return (ONR) is an important feature in a model predicting the direction of intraday return (IDR) regardless of how many times the stock is included in the RF_{ONR} portfolio and how much return it contributes with. We do however notice that for both the top 10 percent most selected and top 10% contributors with the highest return, overnight return is substantially more important than the second most important feature captured by the Gini Impurity value. While the bottom plots show a smaller difference between the best and second-best features.

Finally, we compute the overall feature importance for all stocks included in RF_{ONR} in Figure 10 which again highlights overnight returns as the most important feature. Overall, our analysis shows that overnight returns are an important feature when predicting the direction of intraday returns regardless of how many times it is selected or how much return the stock contributes. An explanation for this phenomenon might be due to the number of stocks to choose from. As described in section 4.5 we found that the number of stocks with close to or similar probabilities was plenty, meaning the competition to be included in the portfolio RF_{ONR} is quite high. A model then would not inherently be considered a bad model even though it is not included in the portfolio, but the size and fierce competition do not allow for more than 10 stocks to be included in the portfolio. This conclusion is also supported by the performance metrics from section 5.2 which shows the overall performance metrics for all portfolios, where RF_{ONR} and GLM_{ONR} beat their counterparts without overnight returns included.

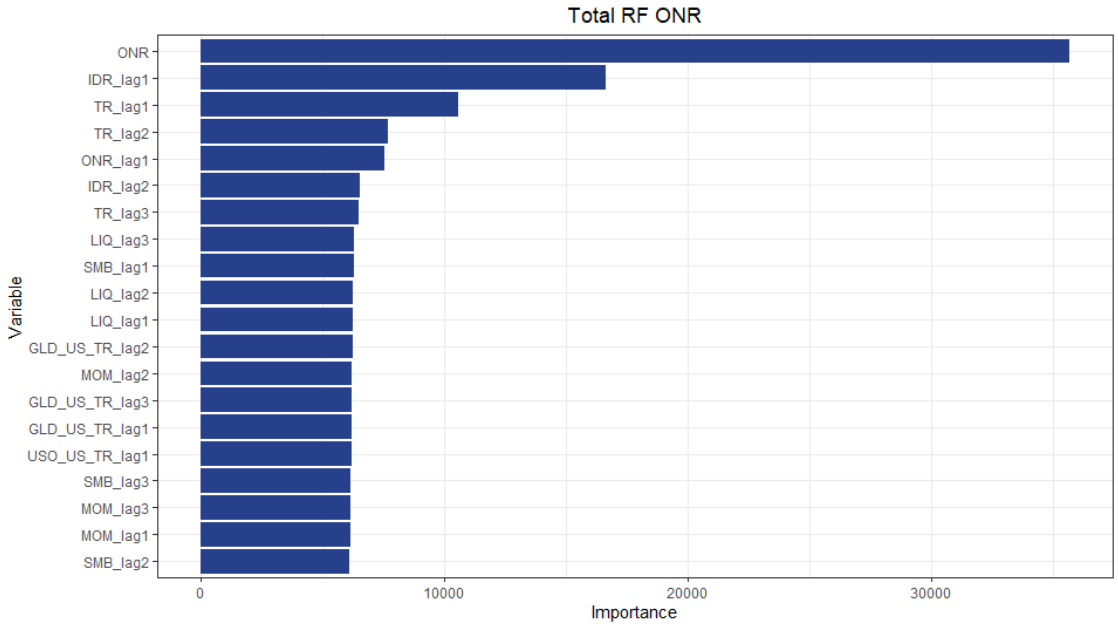


Figure 10: Variable importance for all stocks in Random Forest with ONR

5.3 Overnight return effect

In our analysis of Machine Learning metrics and financial performance of each portfolio, we have discovered how overnight returns when included as a feature contribute to better financial and machine learning results. Both models containing overnight returns as a feature GLM_{ONR} and RF_{ONR} beat their comparable kind without overnight returns and were able to generate positive alpha. The portfolios with overnight returns provide the highest returns measured by the cumulative return as well as risk-adjusted return measured by Sharpe and Sortino compared to their counterparts. Additionally, we find active returns for both overnight return portfolios as well as excess returns.

We also discovered the role and importance of overnight returns as a feature for a machine-learning model for a single stock. The feature importance plots also showed the importance of overnight returns regardless of their return and inclusion in the portfolio RF_{ONR}. However, there was a small difference between the top- and bottom-selected and top- and bottom-contributors where the top stocks had a significant difference between the first and second most important feature. While the bottom stocks had smaller differences between the first and second most important variables. Even though the top stocks showed a larger difference in Gini Impurity compared to the bottom stocks we find this phenomenon interesting. A possible explanation could be due to the fierce competitiveness of being included in the portfolio consisting of 10 stocks out of 315 individual stocks. This means stocks ranked below 11 cannot necessarily be considered bad when the mean Accuracy is 60.23 percent for RF_{ONR}. Finally, stocks move either sideways, up or down due to other several factors which we have not been able to capture in our model specification. This means stocks with low

returns or low selection might still indicate that overnight return is the most important feature in our model specification, but their intraday return is influenced by factors we have not captured.

Table 9: Average correlation between IDR and ONR

Metrics	Negative significant	Not significant	Positive significant
Min	-0.79	-0.15	0.04
Mean	-0.32	-0.01	0.08
Max	-0.04	0.17	0.23
N	269.00	39.00	7.00

Note:
Table shows the correlation between intraday returns and overnight returns. Stocks are placed in each significance category based on the Pearsons t-test results

To conclude our analysis of the role of overnight returns when predicting the direction of intraday returns, we compute the correlation between intraday returns and overnight returns for all stocks in our dataset. Table 9 shows the average correlation between IDR and ONR grouped into three categories: (1) negative significant, (2) not significant and (3) positive significant determined by the Pearsons t-test. Correlation results show that out of 315 individual stocks 269 of them have a significant negative correlation between intraday return and overnight return. This means as the overnight return is negative $ONR < 0$, intraday return tends to be positive $IDR > 0$ and vice versa. Only 7 of the 315 stocks have a positive significant correlation while 39 have a non-significant correlation. Figure 11 show the histogram of t-values from the Pearsons t-test.

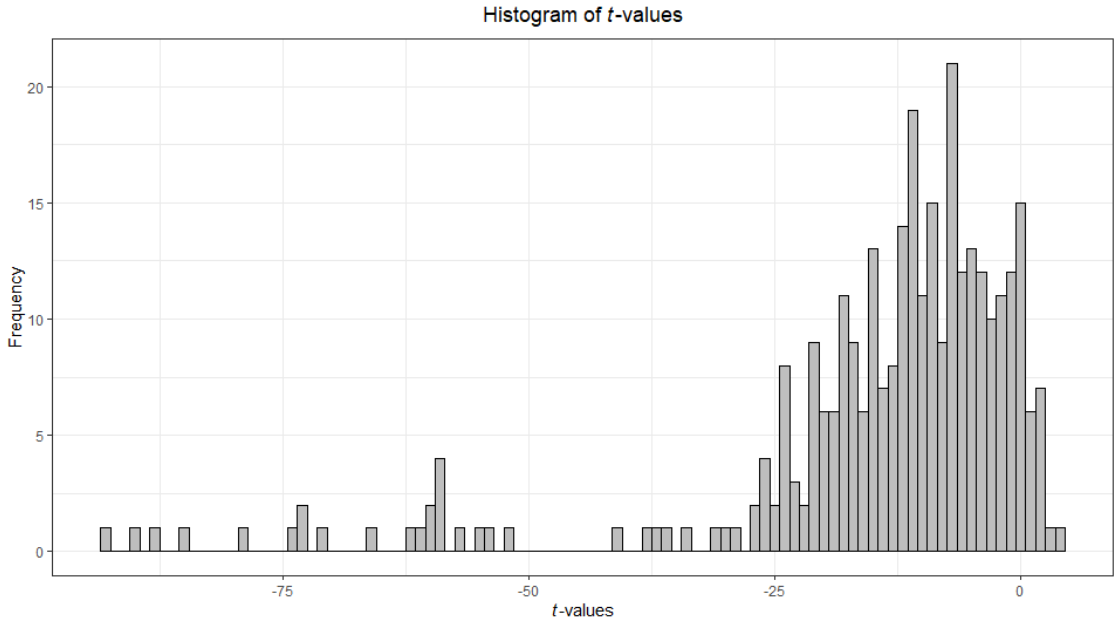


Figure 11: Histograms of t-values between IDR and ONR

6.0 Conclusion and discussion

This study aimed to understand the role of overnight returns as predictors for the direction of intraday returns for stocks on the Oslo Stock Exchange. We combine machine learning and finance to capture the effect of overnight returns. Using 4 portfolios where 2 included overnight returns and the other 2 did not include overnight returns across 2 machine learning models we computed common financial performance metrics as well as machine learning metrics.

Overall, we find that overnight returns have a significant contribution in predicting the direction of intraday returns. Annualized returns for RF_{ONR} and GLM_{ONR} are 323.37 and 97.9 percent respectively compared to their counterparts RF and GLM which had 11.74 and 74.52 percent. The total cumulative return for the portfolios was 1747.58, 163.58, 846.50 and 702.80 percent for RF_{ONR}, RF, GLM_{ONR} and GLM respectively over the close to twelve-year period.

We show that the inclusion of overnight returns in predictions can lead to significantly higher risk-adjusted returns. Annualized Sharpe Ratio for the RF_{ONR}, RF, GLM_{ONR} and GLM portfolios are 12.79, 0.36, 3.65 and 2.55 percent respectively. The Sortino ratio which emphasizes downside risk comes in at 12.33, 0.69, 4.62 and 3.55 percent for RF_{ONR}, RF, GLM_{ONR} and GLM. Comparing our portfolio results against the market captured by active return in the Information Ratio we have 201.86, 0.68, 53.38 and 36.78 percent for RF_{ONR}, RF, GLM_{ONR} and GLM. Indicating the active decision taken in each day's stock selection provides higher returns than the market index OSEBX in the same period. Excess returns over systematic risk captured by the Treynor Ratio 103.53, 2.57, 33.25 and 23.22 percent for RF_{ONR}, RF, GLM_{ONR} and GLM confirms overnight return as features can provide higher returns at approximately the same risk as portfolios without overnight returns.

We also show how overnight returns contribute to better results for machine learning models as the accuracy for each model is 60.23, 57.53, 57.58 and 57.32 for RF_{ONR}, RF, GLM_{ONR} and GLM while the F1-score is 66.70, 62.55, 64.04 and 62.33 respectively. Additionally, we analyze how overnight return as a feature plays a crucial role in predicting the direction of intraday returns, captured by the variable importance plots. Finally, we statistically test for a correlation between overnight returns and intraday returns and find a significant negative correlation for the majority of stocks in our dataset.

Summarized we prove that overnight returns can contribute to better financial results when included as features for stocks on Oslo Stock Exchange. It can provide better machine learning model predictions and has a negative correlation with intraday returns. Overall, this study provides insight into how overnight returns contribute to returns for investors, traders and portfolio managers. We suggest a deeper study of different trading strategies that evaluate other machine learning techniques, investigate the effect in other markets, or optimize the portfolios with different weights, number of stocks or selection criteria.

References

Amihud, Y. (2002). Illiquidity and stock returns: Cross-section and time-series effects. *Journal of Financial Markets*, *5*. https://doi.org/10.1016/S1386-4181(01)00024-6

An, Y., Huang, L., & Li, Y. (2022). The Asymmetric Overnight Return Anomaly in the Chinese Stock Market. *Journal of Risk and Financial Management*, *15*(11). https://doi.org/10.3390/jrfm15110534

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

Berkman, H., Koch, P. D., Tuttle, L., & Zhang, Y. J. (2012). Paying Attention: Overnight Returns and the Hidden Cost of Buying at the Open. *Source: The Journal of Financial and Quantitative Analysis*, *47*(4), 715–741.

Bodie, Z., Kane, A., & Marcus, A. (2021). *Investments, 12th Edition*. McGraw-Hill Education.

Branch, B., & Ma, A. ((2012). Overnight Return, the Invisible Hand Behind Intraday Returns? *Journal of Applied Finance (Formerly Financial Practice and Education)*, *22*(2), 11. https://ssrn.com/abstract=2689719

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, *80*. https://doi.org/10.1016/j.eswa.2017.02.044

Chu, X., Gu, Z., & Zhou, H. (2019). Intraday momentum and reversal in Chinese stock market. *Finance Research Letters*, *30*, 83–88. https://doi.org/10.1016/j.frl.2019.04.002

Cooper, M. J., Cliff, M. T., & Gulen, H. (2008). Return Differences between Trading and Non-trading Hours: Like Night and Day. https://ssrn.com/abstract=1004081

Dash, S. (2022, November 2). *Decision Trees Explained - Entropy, Information Gain, Gini Index, CCP Pruning*. Towards Data Science. https://towardsdatascience.com/decision-trees-explained-entropy-information-gain-gini-index-ccp-pruning-4d78070db36c

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/10.1016/0304-405X(93)90023-5

Gao, R., Zhang, X., Zhang, H., Zhao, Q., & Wang, Y. (2022). Forecasting the overnight return direction of stock market index combining global market indices: A multiple-branch deep learning approach. *Expert Systems with Applications*, *194*. https://doi.org/10.1016/j.eswa.2022.116506

Haghani, V., Ragulin, V. V., & Dewey, R. (2022). Night Moves: Is the Overnight Drift the Grandmother of All Market Anomalies. http://dx.doi.org/10.2139/ssrn.4139328

Haghani, Victor and Ragulin, Vladimir V and Dewey, Richard, Night Moves: Is the Overnight Drift the Grandmother of All Market Anomalies (June 17, 2022). Available at SSRN: https://ssrn.com/abstract=4139328 or http://dx.doi.org/10.2139/ssrn.4139328.

Hansen, O.-M., & Rustad, J. (2023). Machine Learning Models for Predicting ETF Returns: The Role of Overnight Returns. *Unbuplished report*.

Hlavac, M. (2022). *Stargazer: Well-Formatted Regression and Summary Statistics Tables*. Bratislava, Slovakia. https://CRAN.R-project.org/package=stargazer
Note: R package version 5.2.3.

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, *129*, 273–285. https://doi.org/10.1016/j.eswa.2019.03.029

Huang, A. Y. H., Hu, M. C., & Truong, Q. T. (2021). Asymmetrical impacts from overnight returns on stock returns. *Review of Quantitative Finance and Accounting*, *56*(3), 849–889. https://doi.org/10.1007/s11156-020-00911-y

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning* (Vol. 103). Springer New York. https://doi.org/10.1007/978-1-4614-7138-7

Karaca, Y., Zhang, Y. D., & Muhammad, K. (2020). Characterizing Complexity and Self-Similarity Based on Fractal and Entropy Analyses for Stock Market Forecast Modelling. *Expert Systems with Applications*, *144*. https://doi.org/10.1016/j.eswa.2019.113098

Kelly, M. A., & Clark, S. P. (2011). Returns in trading versus non-trading hours: The difference is day and night. *Journal of Asset Management*, *12*(2), 132–145. https://doi.org/10.1057/jam.2011.2

Kilskar, S. (2020). *Aksjeutvelgelse ved bruk av Random Forest: en maskinlæringstilnærmelse til meravkastning* [Master's thesis, Nord University]. https://hdl.handle.net/11250/2682978

Kuhn & Max. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, *28*(5), 1–26. https://doi.org/10.18637/jss.v028.i05

Kumar, M. (2006). FORECASTING STOCK INDEX MOVEMENT: A COMPARISION OF SUPPORT VECTOR MACHINES AND RANDOM FOREST. *Indian Institute of Capital Markets 9th Capital Markets Conference Pape*. Retrieved April 30, 2024, from http://dx.doi.org/10.2139/ssrn.876544

Kyurkchan, A. G. (2020). *Quantitative investment analysis (Fourth edition.)* (Fourth edition). Wiley.

Leippold, M., Wang, Q., & Zhou, W. (2022). Machine learning in the Chinese stock market. *Journal of Financial Economics*, *145*(2), 64–82. https://doi.org/10.1016/j.jfineco.2021.08.017

Liu, Q., & Tse, Y. (2017). Overnight returns of stock indexes: Evidence from ETFs and futures. *International Review of Economics and Finance*, *48*, 440–451. https://doi.org/10.1016/j.iref.2017.01.005

Long, W., Lu, Z., & Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, *164*, 163–173. https://doi.org/10.1016/j.knosys.2018.10.034

Malagrino, L. S., Roman, N. T., & Monteiro, A. M. (2018). Forecasting stock market index daily direction: A Bayesian Network approach. *Expert Systems with Applications*, *105*, 11–22. https://doi.org/10.1016/j.eswa.2018.03.039

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

Peterson, B. G., & Carl, P. (2020). PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis. https://CRAN.R-project.org/package=PerformanceAnalytics

R Core Team. (2023). R: A Language and Environment for Statistical Computing. https://www.R-project.org/

Ruble, I., & Powell, J. (2021). The Brent-WTI spread revisited: A novel approach. *The Journal of Economic Asymmetries*, *23*, e00196. https://doi.org/10.1016/j.jeca.2021.e00196

Ryan, M., Corbet, S., & Oxley, L. (2024). Is gold always a safe haven? *Finance Research Letters*, *64*, 105438. https://doi.org/10.1016/j.frl.2024.105438

Taera, E. G., Setiawan, B., Saleem, A., Wahyuni, A. S., Chang, D. K. S., Nathan, R. J., & Lakner, Z. (2023). The impact of Covid-19 and Russia–Ukraine war on the financial asset volatility: Evidence from equity, cryptocurrency and alternative assets. *Journal of Open Innovation: Technology, Market, and Complexity*, *9*(3). https://doi.org/10.1016/j.joitmc.2023.100116

Wen, D., Wang, Y., & Zhang, Y. (2021). Intraday return predictability in China's crude oil futures market: New evidence from a unique trading mechanism. *Economic Modelling*, *96*, 209–219. https://doi.org/10.1016/j.econmod.2021.01.005

Wickham, H. (2016). *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wright, M. N., & Ziegler, A. (2017). Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software*, *77*(1), 1–17. https://doi.org/10.18637/jss.v077.i01

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, *67*, 126–139. https://doi.org/10.1016/j.eswa.2016.09.027

Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, *5*(1). https://doi.org/10.1186/s40854-019-0138-0

Appendix A

Table 10: Summary statistics for global and macro variables

Variables	Min	Max	Mean	Median	Std..Dev.
SMB	-0.0431	0.6000	3e-04	-1e-04	0.0131
LIQ	-0.1763	0.0503	0e+00	0e+00	0.0096
MOM	-0.0619	0.0721	3e-04	4e-04	0.0117
lnDeltaOSEBX	-0.0918	0.0618	4e-04	8e-04	0.0115
lnDeltaOBX	-0.0895	0.0629	4e-04	6e-04	0.0120
NOWA_DayLnrate	0.0000	0.0002	1e-04	1e-04	0.0000
bills_3month_Lnrate	0.0000	0.0002	1e-04	1e-04	0.0000
GLD-US_TR	-0.0919	0.0480	1e-04	2e-04	0.0097
SPY-US_TR	-0.1159	0.0867	4e-04	4e-04	0.0109
USO-US_TR	-0.2919	2.1342	1e-04	2e-04	0.0427