UiT The Arctic University of Norway

Faculty of bioscience, fisheries and economics

# Creating alpha with Machine Learning algorithms

Sondre Teigen

Master's thesis in Business administration, BED-3901, December 2024

# i. Acknowledgments

Writing this thesis has been an interesting and challenging journey, and it would not have been possible without the support and guidance of many individuals. I would like to take this opportunity to express my gratitude to those who have contributed to the successful completion of this work.

First and foremost, I am grateful to my supervisor, Thomas Leirvik for their guidance and constructive feedback. I am also thankful to my colleagues and peers at The University of Tromso, whose discussions and shared experiences have been most appreciated.

A special note of appreciation goes to Tore Teigen for his insightful participation in the more difficult aspects of Machine learning and economic theory in general. And at last I would thank Liv Mathisen for being able to listen to financial talk for half a year.

This thesis is the culmination of collective efforts, and I am deeply thankful to everyone who has played a part in this journey.

Tromsø, December $1^{st}$, 2024

Sondre Johann Teigen

# ii. Abstract

This thesis investigates the application of machine learning (ML) and traditional financial models in portfolio optimization, focusing on the OBX index. The research aims to determine whether ML algorithms can outperform traditional models in forecasting returns, estimating volatility, and optimizing portfolio weights.

The study employs advanced ML techniques such as Random Forest, Support Vector Machines, Gradient Boosting Machines, and k-Nearest Neighbors alongside traditional models, including ARIMA for return prediction and various GARCH frameworks for volatility modeling. Performance is evaluated using risk-adjusted metrics such as the Sharpe Ratio, Sortino Ratio, and Fama-French-Carhart regressions to assess the alpha generated by each model.

Results reveal that ML-based portfolios significantly outperform the benchmark OBX index in both risk and return. Notably, the Random Forest model with a nine-week rolling window achieved the highest annualized return of 17.83% and a cumulative total return of 97.71% over 200 weeks, while maintaining lower volatility than the benchmark. Traditional models also performed well, with the IGARCH-based portfolio showing strong results, although they fell short of ML-based approaches.

## Contents

## Table of figures



## Table of tables



## Table of equations

Keywords:

Portfolio allocation, optimal weights, machine learning, correlation matrix, volatility.

# 1 Introduction:

Financial markets have long presented investors with the challenge of balancing risk and reward to optimize portfolio performance. This thesis explores the application of machine learning (ML) models alongside traditional forecasting methods to determine optimal portfolio weights for the OBX index, which comprises the 25 most traded stocks on the Norwegian stock market. Specifically, the study aims to evaluate the predictive power of ML algorithms such as Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Gradient Boosting Machines (GBM) for both return and volatility forecasting. These methods will be compared to traditional models, including ARIMA for returns and various GARCH variants for volatility. By integrating these techniques, the study adopts a utility-maximizing perspective to analyse how risk reward tradeoffs can be optimized in a financial market.

The concept of "reward-risk timing," as described by Kirby and Ostdiek (2012), is the approach taken in this thesis. This dual focus on forecasting direction and volatility has been shown in prior research to generate alpha, making it a promising strategy for portfolio management. Pinelis and Ruppert (2022) demonstrated that combining return and volatility predictions using machine learning models like Random Forest resulted in better performance compared to traditional methods. Their findings suggest that machine learning algorithms not only enhance return forecasts but also improve portfolio construction by better capturing the complex relationships between financial variables.

Henrique et al. (2019) provide a comprehensive literature review on machine learning applications in financial market prediction, highlighting the substantial body of research dedicated to both direction and volatility forecasting. Studies like Gu et al. (2020) have found that Random Forest significantly improves the Sharpe ratio when compared to traditional buy-and-hold strategies. Additionally, Kim (2003) demonstrated that SVM models outperformed traditional approaches such as Backpropagation Neural Networks (BPN) and Case-Based Reasoning (CBR) in financial time series prediction. Similarly, Chen et al. (2003) identified the strong predictive capabilities of neural networks on the Taiwanese stock market, emphasizing the benefits of incorporating next-day probabilities for managing portfolio risk.

Collectively, these studies highlight the growing consensus around the utility of machine learning in enhancing financial forecasts.

To optimize portfolio weights, this thesis uses forecasted returns and volatilities generated from the ML algorithms and traditional models. Pinelis and Ruppert (2022) showed that machine learning models, especially Random Forest, outperform traditional linear models in return forecasting. Similarly, Kim and Han (2000) demonstrated the ability of neural networks to incorporate big data variables, significantly enhancing prediction accuracy. These findings underscore the importance of combining advanced machine learning techniques with traditional models to address the multifaceted challenges of financial forecasting.

Volatility prediction plays a critical role in portfolio risk management. Bollerslev (1986) introduced the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, building on Engle's (1982) earlier work. Variations such as GJR-GARCH and IGARCH (Nelson, 1990) have been developed to capture specific aspects of market volatility. This thesis incorporates GARCH (1,1), GJR-GARCH, and IGARCH models to forecast volatility and construct correlation matrices for the portfolio. These models are widely recognized for their ability to model and predict volatility, offering distinct advantages depending on the requirements of the analysis. Bollerslev et al. (2016) and Zhang et al. (2020) emphasized the accuracy of rolling window volatility calculations in financial forecasting. By adopting rolling windows of three, six, and nine weeks, this study examines how different time horizons influence the performance of both ML and traditional models in volatility predictions.

The integration of machine learning into financial forecasting has shown potential for enhancing portfolio management strategies. Neural networks, as demonstrated by Chen et al. (2003), provide valuable insights for risk-averse investors, particularly when probabilities of negative outcomes are forecasted. Such techniques allow for more informed investment decisions, such as avoiding investment on days, weeks, or months with unfavorable forecasts. This thesis investigates whether ML algorithms can replicate or exceed these benefits when applied to the OBX index, using financial and macroeconomic variables to predict excess returns and volatility.

The core of this thesis lies in its dual approach to return and volatility forecasting, utilizing both traditional and ML models. By comparing these methods across different rolling windows, the study seeks to answer critical questions about the relative strengths of machine learning and traditional models in financial forecasting. Additionally, this research explores whether machine learning algorithms respond differently to varying time horizons, providing new insights into their application in portfolio risk management. The findings of this study aim to contribute to the growing body of literature on machine learning in finance, offering practical implications for both academic researchers and investment practitioners.

*This study explores the potential for machine learning techniques to generate alpha by optimizing stock portfolio weights.*

*Can Machine Learning Algorithms Outperform Traditional Models in Portfolio Optimization?*

# 2 Theoretical framework:

## 2.1 Theoretical Framework – Introduction

In this section, the foundational theories and models that are used in creating optimal stock portfolios, using machine learning and traditional financial models are presented. Each model discussed here plays a role in achieving the objectives of this thesis: to determine the optimal weights for stocks in the OBX index and to enhance the portfolio's performance through prediction techniques. Theories such as the Minimum Variance Portfolio (MVP) (Markowitz, 1952), various machine learning methods, and volatility modeling techniques using GARCH collectively form the basis for the research.

The framework begins by exploring approaches to portfolio optimization. Markowitz's Minimum Variance Portfolio provides a framework for reducing risk in a diversified portfolio. The application of machine learning models such as Random Forest (Breiman, 2001), Support Vector Machines (SVM) (Vapnik, 1995), and others in predicting financial metrics like direction and volatility is examined. These machine learning models enable more data-driven decision-making compared to traditional models.

The use of different volatility prediction models, such as the GARCH models (Bollerslev, 1986), is also considered, allowing for capturing and modeling changing market conditions with greater accuracy. By discussing each model's strengths, weaknesses, and specific applications, this section aims to build a comprehensive framework that underpins the portfolio construction process.

To ensure clarity, each theoretical model and approach discussed in this section is directly linked to the research questions outlined in the introduction. Specifically, the Minimum Variance Portfolio is used to address the question of how traditional financial models can effectively balance risk and return in portfolio construction. The machine learning models—including Random Forest, SVM, GBM and KNN are integrated to investigate whether these advanced techniques can enhance the prediction of market movements and thereby improve the portfolio optimization process compared to traditional models. Further, the GARCH models are utilized to answer questions related to capturing volatility in financial markets and optimizing risk-adjusted returns through more accurate volatility forecasts. The integration of these models aims to test whether combining traditional and machine learning approaches can yield superior results in managing risk and enhancing returns. By explicitly linking each component of the theoretical framework to specific research questions, this section provides a structured basis for understanding the relevance of each model in addressing the overall objectives of the thesis.

The theories and models in this section are forming an approach to portfolio optimization. By combining insights from traditional finance, machine learning, and statistical techniques, this framework lays the foundation for an empirical exploration of the effectiveness of these models in creating optimized, risk-adjusted portfolios.

## 2.2 Risk / Return

A key component in a portfolio is the trade between risk and return. Markowitz (1952) explained the trade-off by introducing what is now known as *modern portfolio theory* (MPT) in his paper “portfolio selection”. The theory is built on the assumption that the rational investors task is to maximize utility. This can be done by either minimizing risk for a set level of return or maximizing returns for a given level of risk. For a diversified portfolio the expected return can be explained by

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

*Equation 1: expected return*

where $E(r_p)$ is the expected portfolio value, $w_i$ is the weight for the asset $I$ in the portfolio. The risk of the portfolio can be computed by

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

*Equation 2: portfolio risk*

where $\sigma_p^2$ is the portfolio risk, $w_i, w_j$ are the weights for assets $i, j$, $\sigma_i, \sigma_j$ are the portfolio volatility, $p_{i,j}$ are the correlation between the assets $i, j$.

The portfolio variance, also known as risk, equation has a dependency on the variance of the individual components. The weights and the correlations of the two assets $I, j$. Since the portfolio has a dependency on the correlations of assets $I, j$ the portfolio risk can be reduced by including assets with a lower correlation. Markowitz (1952) argued that the rational investor should therefore not only be concerned with maximizing returns, but also with minimizing risk.

A key component of Markowitz (1952) theory is the diversification of the portfolio. For instance, if the investor only invested in oil stocks, on the same market, in the same period, the assumption would be that the stocks probably have a higher correlation. And the portfolio is not actively trying to minimize the risk for every unit of return.

In this thesis, the risk/return framework serves as a foundation for evaluating the effectiveness of machine learning models in improving portfolio performance.

Following this, the application of machine learning models such as Random Forest (Breiman, 2001), Support Vector Machines (SVM) (Vapnik, 1995), and others in predicting financial metrics like direction and volatility is examined. These machine learning models enable more dynamic and data-driven decision-making compared to traditional models.

## 2.3 GARCH Models for volatility and correlations

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, introduced by Bollerslev (1986), is a key tool in financial econometrics for modelling and forecasting time series volatility. The GARCH model is particularly useful in capturing the time-varying nature of volatility, which is a common characteristic of financial returns. In this thesis, GARCH models are employed to estimate the volatility and correlations between assets in the OBX index, providing necessary insights for optimizing portfolio risk.

The GARCH model extends the original Autoregressive Conditional Heteroskedasticity (ARCH) model proposed by Engle (1982), allowing for a more flexible lag structure in modelling volatility. The GARCH (p, q) model is defined the parameters: $p$, the order of the GARCH terms (past conditional variances), and $q$, the order of the ARCH terms (past squared observations). The GARCH (1,1) model, commonly used in practice due to its balance of simplicity and effectiveness, expressed as:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

*Equation 3: GARCH (1,1)*

Where $\sigma_t^2$ is the conditional variance at time $t$, $\alpha_0$ is a constant, $\alpha_1$ represents the impact of past shocks (ARCH term) and $\beta_1$ represents the persistence of volatility, also called the GARCH term. This formulation allows the model to capture both short-term shocks and longer-term volatility persistence.

In addition to the standard GARCH model, other variations such as the GJR-GARCH and IGARCH models are also utilized in this thesis. The GJR-GARCH model, introduced by Glosten, Jagannathan, and Runkle (1993), is designed to account for the asymmetric effect of positive and negative shocks on volatility, often referred to as the leverage effect. This is particularly important when modeling financial markets, where negative returns tend to have a larger impact on future volatility than positive returns of the same magnitude. It can be denoted as

$$\sigma_{t+1}^2 = w + \alpha r_t^2 + \gamma I(r_t < 0) r_t^2 + \beta \sigma_t^2$$

*Equation 4: GJR-GARCH*

Where it is the same as the regular GARCH, but we add $\gamma I(r_t < 0)r_t^2$ where $\gamma$ is multiplied by the squared return $r_t^2$ when the return $r^t$ is negative, which is indicated by the indicator function $I(r_t < 0)$. The indicator function = 1 when $r^t$ is less than 0. This term allows the model to capture the asymmetric impact of negative returns on future volatility, which is the leverage effect.

The Integrated GARCH (IGARCH) model, is used to capture the persistence of volatility over time. The IGARCH model assumes that the impact of past shocks on current volatility does not decay completely, making it suitable for financial time series with highly persistent volatility patterns. The IGARCH model is denoted as

$$\sigma_{t+1}^2 = w + \alpha r_t^2 + (1 - \alpha)\sigma_t^2$$

*Equation 5: IGARCH*

Where everything is the same except for the $(1 - \alpha)$. This coefficient is applied to the previous period variance, $\sigma_t^2$. the coefficient adds up to 1, indicating that past variances are highly persistent and have a long-lasting impact.

These GARCH models are estimating the correlation matrix of the assets in the OBX index, which in turn is used to construct the covariance matrix for portfolio optimization. By accurately modelling volatility and correlations, the GARCH framework provides a more robust basis for managing risk and achieving optimal asset allocation in the presence of time-varying market conditions.

## 2.4 Minimum Variance Portfolio:

The MVP approach seeks to minimize risk by optimizing the weighting of assets, specifically by selecting assets with low or negative correlations to ensure effective diversification. This is important for the OBX index, where achieving optimal diversification can mitigate systematic risk common to this region.

Markowitz's Minimum Variance Portfolio (MVP) provides a benchmark for reducing risk in a diversified portfolio. It provides a clear reference point to evaluate more complex models by comparing how different approaches optimize the risk-return trade-off. One of the key reasons MVP is appropriate for this analysis is due to its simplicity and effectiveness in markets where data availability is limited.

The concept of the minimum variance portfolio (MVP) is that the portfolio should consist of a combination of stocks that have the lowest possible volatility for a given level of return. The portfolio consists of three main parts, the returns, the volatility, and the covariance matrix. In the case of this thesis, the parts will consist of predicted values. To find the MVP I will solve the optimization problem where the goal is to minimize the portfolio volatility.
Mathematically it can be illustrated as

$$\frac{min}{w}(w^T \sum_w)$$

*Equation 6: optimization problem*

which is subjected to $w^T 1 = 1$ where $w$ is the vector of the portfolio weight for each asset. $\sum$ is the covariance matrix of the asset returns. 1 is the vector of ones. The optimization will be solved using quadratic programming where we

$$min\ \frac{1}{2} x^T Q x \ + \ c^T x$$

*Equation 7: Quadratic programming*

which is subject to $Ax \ \leq \ b$, $Ex \ = \ d$ and $x \geq \ 0$. Where $Q \ = \ 2\sum$, $c$ is a zero vector. $A \ and \ b$ represents the inequality constraints and $E \ and \ d$ represent the equality constraints of the weights summing to one.

## 2.5 Machine learning in financial markets:

In 1970 Maikel and FAMA developed the efficient market hypothesis (EMH). According to the hypothesis financial markets follow random pathways and are therefore unpredictable (Henrique et al, 2019). The search for models that can predict the market is still highly researched, and in later years machine learning has become popular in predicting these pathways. Machine learning is defined as a subset of artificial intelligence and can be described in this context as a set of tools and techniques to analyze historical data, recognize patterns in a “training process” and use the process to make predictions about future market movements (Xiao et al, 2013).

Predicting time series data in financial markets with non-stationarity is a complex task, especially with traditional predictive financial models such as moving averages, autoregressive models, and discriminating analyses (Zhang et al, 2017). This data is often characterized as noisy and non-linear (Kumar and Thenmozhi, 2014). These could be variables that are influenced by untraditional but significant macro factors such as the political climate, is there political uncertainties, which could be measured by a sentiment variable. The investor's psychology could also be significant hence the CNN`s fear and greed index which is a psychological measurement of the marked temperament.

The use of machine learning in financial data analysis dates to the work by Hawley et al. (1990). Since then, there has been a significant increase in computing power, which has made it possible to use more complex algorithms. These algorithms often require a lot of computing resources and have only recently become accessible to the average investor due to advancements in technology. The rise in computing power has coincided with an increase in the amount and types of financial data available, which has helped to integrate machine learning more deeply into financial analysis.

These advanced algorithms, previously only available to large financial institutions, are now accessible to a wider range of investors. Research in this area, such as that by Henrique et al. (2019), shows that machine learning can often predict financial outcomes better than traditional methods. This suggests a strong potential for these tools to improve investment strategies. As machine learning becomes more integrated into financial analysis, it is increasingly seen as a valuable tool for making data-driven decisions that can lead to better financial outcomes.

Several different machine learning techniques are available for forecasting purposes, each with its unique strengths and weaknesses. These include methods such as random forest, support vector machines (SVM), neural networks, k-nearest neighbor (k-NN), and gradient boosting machines.

Each technique operates differently. For example, random forests build multiple decision trees and merge them to get a more accurate and stable prediction. Support vector machines are effective in high-dimensional spaces and are versatile as they can be configured with different kernel functions. Neural networks are particularly powerful for capturing nonlinear relationships in large datasets. K-nearest neighbor make predictions based on the closest data points in the feature space, providing intuitive, if computationally expensive, insights. Lastly, gradient-boosting machines sequentially build models and focus on correcting the errors in previous models to improve accuracy.

The rationale for employing multiple techniques in a forecasting model is to leverage their distinct approaches to better understand and predict the data. By comparing different models, analysts can identify the most effective technique for capturing the nuances of both direction and volatility in the dataset. This methodical comparison helps in selecting the machine learning model that best fits the data, optimizing forecasting accuracy.

### 2.5.1 Random Forrest Model:

The Random Forest model (RF) was developed by Breiman (2001). Random Forest is a supervised learning algorithm that can be used for both classification and regression tasks. This model operates as an ensemble of decision trees, formally expressed as

$$\{h(x, \theta_k), k = 1\},$$

*Equation 8: ensemble of decision trees*

where $\theta k$ are independent identically distributed random vectors. Each tree in the ensemble contributes a unit vote towards the most popular class for a given input $x$. Breiman (2001) describes this voting mechanism as follows:

$$\text{RF}(x) = \text{mode}\{h(x, \theta 1), h(x, \theta 2), \dots, h(x, \theta K)$$

*Equation 9:voting mechanism*

where $K$ represents the total number of trees in the forest, and $\theta k$ characterizes the random parameters that define each tree. The model utilizes an ensemble learning technique, which combines the predictions from multiple trees to produce a single prediction. This aggregation of diverse tree predictions enhances the model's ability to generalize, effectively reducing the overfitting problem often seen with individual decision trees. The ensemble approach ensures that the collective prediction is more stable and accurate than that of any individual tree within the forest. The model is illustrated below.

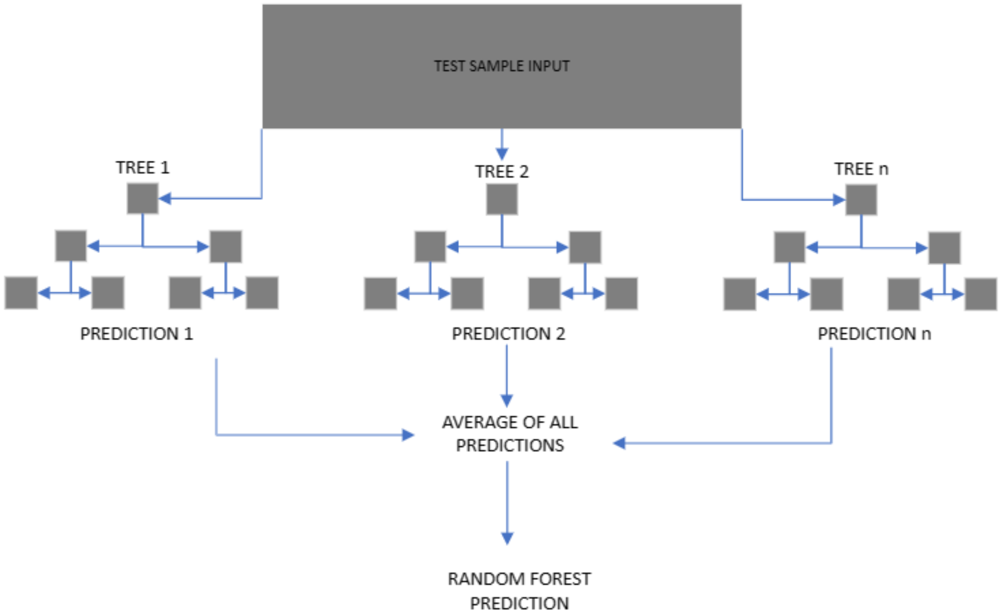


*Figure 1:Random Forest illustration*

The figure above provides a simplified overview of how the Random Forest (RF) model operates. In this model, input data is distributed among various decision trees, each of which assesses the importance of the data. Data deemed less critical by the trees are relegated to lower levels of the tree structure, thereby having a reduced impact on the final prediction outcome.

the Random Forest model also presents several challenges: Overfitting: Although Random Forest generally handles overfitting better than individual decision trees, particularly in cases with large amounts of data, it can still overfit if the data is noisy or if the trees are overly complex. Interpretability: Due to its complex ensemble structure involving numerous decision trees, Random Forest models are often considered as "black boxes." This means that it can be challenging to discern how specific features influence the overall predictions, making the model less interpretable compared to simpler, more transparent models. Number of Trees: The selection of an optimal number of trees in the forest is crucial for achieving the best performance. Too few trees might lead to underfitting, while too many can increase computational costs without corresponding gains in accuracy. This parameter must be carefully tuned, typically through cross-validation or similar techniques to balance between performance and computational efficiency.

### 2.5.2 Support Vector Machines:

Support vector machines (SVM) is a supervised learning algorithm. Created by Vapnik et al (1992). According to MathWorks (n.d.), SVM can be used for classification or regression and is most successful when using small and complex datasets. Suppose we have a training data set and want to find the linear function $f(x) = x'\beta + b$, it is necessary to ensure that the linear function is as flat as possible therefore we need to find $f(x)$ with the minimal normal value $J(\beta) = \frac{1}{2}\beta'\beta$ which is subject to the constraint that the residuals have a smaller or equal value than the noise term (MathWorks, n.d.), $\forall_n = |y_n - (x_n{'}\beta + b)| \leq \varepsilon$ in some cases no such function $f(x)$ exist and the implementation of slack variables $\xi_n \xi_n^*$ would be necessary (MathWorks, n.d.). The slack variables allow regression error and still satisfy the conditions. Including the slack variables to the objective function: $J(\beta) = \frac{1}{2}\beta'\beta + C\sum_{n=1}^{N}(\xi_n \xi_n^*)$. Which is subject to:

$\forall_n: y_n - (x_n'\beta + b) \leq \varepsilon + \xi_n$

$\forall_n: (x_n'\beta + b) - y_n \leq \varepsilon + \xi_n^*$

$\forall_n: \xi_n^* \geq 0$

$\forall_n: \xi_n \geq 0$

(MathWorks, n.d.),

### 2.5.3 K-Nearest-Neighbour:

KNN is a non-parametric machine learning method that can be used for both classification and regression. For a given input $x$ the algorithm identifies the $K$ closest points in the training data and makes predictions based on these points. The classification task can be described as follows, KNN assigns a class to $x$ by taking a majority vote among the classes of its $K$ nearest neighbours. The predicted class $\hat{y}$ is given by $\hat{y} = \text{mode}\{y_{i1}, y_{i2}, \dots, y_{iK}\}$ where $y_{ij}$ are the labels of the KNN to $x$. The classification is illustrated in the figure below where x is highlighted in the middle and every other point is K.

*Figure 2: KNN illustration*

For the regression KNN predicts the output by averaging the values of the KNN:

$$\hat{y} = \frac{1}{K}\sum_{k=1}^{K} y_{ik}$$

*Equation 10: KNN regression*

where $y_{ik}$ are the values of the KNN to $x$. When using the KNN it's important to compute the distance $d$ between the input $x$ and all points $x_i$ in the training set. The most common distance metrics are Euclidean, Manhattan, and Hamming. This thesis will use Euclidean which is explained by

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} \left(x_j - x_{ij}\right)^2}$$

*Equation 11:Euclidean*

The advantage of using KNN is that the model is simplistic and effective since it uses low-dimensional data. The disadvantages are that the performance decreases with an increase in the dimensionality of the data that it is sensitive to the scale of data features.

### 2.5.4 Gradient Boosting Machines:

Gradient boosting machines (GBM) is an ensemble learning method, where the algorithm combines the predictions of weak learners to create a stronger more robust model (Masui, T, n.d.). GBM creates trees sequentially, where each tree corrects the errors of the previous tree. GBM focuses on the weakness in the model, focusing on the data points that were poorly created. The objective is to minimize the mean squared error. The objective function for GBM is:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - F(x_i)\right)^2$$

*Figure 3: GBM objective function*

The objective function (Masui, T, n.d.) is given by $J(\theta)$. N is the number of data points. $y_i$ is the target for the $i-th$ data point. $F(x_i)$ is the current prediction. The boosting process contains of sequentially adding weak learner to the ensemble, where each is correcting the errors of the prior one (Masui, T, n.d.). At each iteration, a new weak learner is added to the ensemble to minimize the gradient of the objective function with respect to the current prediction. For the $m-th$ iteration the model is made by fitting a weak learner to the negative gradient in the loss function. Weak learner: $h_m(x) = \arg\min_h \sum_{i=1}^{N}\left[-\frac{\partial J(\theta)}{\partial F(x_i)}\right]^2$. The negative gradient term $-\frac{\partial J(\theta)}{\partial F(\boldsymbol{x_i})}$ is the residual of the loss function with respect to the current prediction (Masui, T, n.d.). To control the contribution of the weak learner to the overall model, the learning rate $(\eta)$ is introduced. The prediction at each iteration is scaled by the learning rate (Masui, T, n.d.):$F(x) \leftarrow F(x) + \eta \cdot h_m(x)$. A small learning rate prevents overfitting and improves the generalization of the model. In some cases, it is necessary to implement regularization to prevent overfitting. Often if the learning rate is too high. Adding the regularization term to the objective function:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - F(x_i)\right)^2 + \lambda \cdot \Omega(h_m)$$

*Equation 12: GBM objective function with normalization term* where $\Omega(h_m)$ is the regularization term. $\lambda$ controls the strength of the regularization (Masui, T, n.d.).

## 2.6 ARIMA

The Autoregressive integrated moving average (ARIMA) model were introduced by Box and Jenkins (1970). The model has become one of the most popular models used in stock prediction. The ARIMA model assumes that the future value of stock is a linear combination of past values and errors. The model is given by

$$y_t = \theta_0 + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q}$$

*Equation 13: ARIMA model*

$y_t$ is the actual value, $\varphi_i, \theta_j$ are the coefficients, $p, q$ are the integers, respectively the autoregressive and the moving average polynomials. The weakness of the ARIMA model, compared to ML models is the inability to use nonlinear data, but in some cases are a more simplistic model beneficial. It’s also beneficial to use more traditional models as a performance comparison.

## 2.7 Rolling window Volatility

Rolling window volatility is a method used to estimate the variability of returns over a moving time window, providing insights into how market volatility evolves over time. In this approach, a fixed-length time period, or "window," (e.g., 3, 6, or 9 weeks) is moved incrementally across the dataset, and volatility is calculated within each window. This dynamic technique allows for real-time tracking of volatility patterns, which is crucial for risk management and portfolio optimization (Bollerslev, 1986; Zhang et al., 2020).

By continuously updating volatility estimates, rolling windows capture short-term market fluctuations and help identify periods of heightened or reduced risk. This is particularly useful in financial markets where volatility is not constant but clusters in time, as noted in the GARCH model framework by Bollerslev (1986). For instance, during periods of economic uncertainty or market stress, rolling window volatility can reveal sudden spikes in risk, enabling investors to adjust their portfolios accordingly.

In portfolio management, rolling window volatility is often paired with predictive models to refine risk assessments and optimize asset allocation. For example, Zhang et al. (2020) emphasized that rolling windows enhance the performance of volatility models by adapting to recent market changes, improving their predictive accuracy.

## 2.8 Portfolio construction

The machine learning and ARIMA-GARCH portfolio construction requires multiple steps. First, the stock direction is predicted using the classification specification in the different algorithms (Random Forest, SVM, KNN, GBM, ARIMA). This is done individually for each stock. The second step involves predicting volatility using the regression specification in the different algorithms. For the ARIMA model, the volatility is predicted using three different GARCH models. Third, the correlation matrices are calculated using these GARCH models. Fourth, the covariance matrix is calculated using the predicted volatilities and the correlation matrices.

The machine learning portfolio construction similarly requires multiple steps. First, the stock direction is predicted using the classification specification in the different algorithms. Second, the volatility is predicted using the regression specification. Third, the correlations are calculated from the different GARCH models. Fourth, the covariance matrix is calculated from the predicted volatilities and correlations. Finally, the optimal weights are calculated using the predicted direction and the predicted covariance matrix. The weights are solved using quadratic programming, and these weights are then multiplied by the returns to get the weight-adjusted returns of the portfolio.

For the ARIMA-GARCH models, the return direction is predicted using the ARIMA model, and the volatilities are predicted using the different GARCH models. From this point onward, every step is the same as in the machine learning portfolios, including correlation matrix calculation, covariance matrix construction, and optimization of weights to achieve an optimal risk-return balance.

## 2.9 Model evaluation

### 2.9.1 Sharp Ratio

The Sharpe Ratio, introduced by Sharpe (1966), is a key metric for assessing the performance of an investment by adjusting for its risk. It is calculated as the difference between the portfolio return and the risk-free rate, divided by the portfolio's standard deviation (a measure of risk). The formula for the Sharpe Ratio is given by:

$$S = \frac{E(R_p) - R_f}{\sigma_p}$$

*Equation 14: Sharp Ratio*

Where S is the Sharpe Ratio $E(R_p)$ represents the expected return of the portfolio. $R_f$ is the risk-free rate, and $\sigma_p$ is the standard deviation of portfolio returns. The Sharpe Ratio provides a standardized way to measure risk-adjusted returns, allowing investors to compare the performance of different portfolios or investment strategies.

The Sharpe Ratio is used to evaluate the portfolios constructed using both machine learning models and traditional approaches. By comparing the Sharpe Ratios, it is possible to determine which methodology provides a superior balance of return relative to the risk undertaken. A higher Sharpe Ratio indicates that a portfolio has achieved higher returns per unit of risk, making it a valuable tool for evaluating whether machine learning-based portfolios outperform those built using conventional techniques.

### 2.9.2 Information ratio

The Information Ratio (IR) is another key performance metric used to evaluate the efficiency of an investment strategy relative to a benchmark. Unlike the Sharpe Ratio, which compares portfolio returns to the risk-free rate, the Information Ratio evaluates the excess return of a portfolio over a benchmark, adjusted for the volatility of that excess return. The formula for the Information Ratio is given by:

$$IR = \frac{R_p - R_b}{\sigma(R_p - R_b)}$$

*Equation 15: Information Ratio*

where $R_p$ is the portfolio return, $R_b$ is the benchmarke return, and $\sigma(R_p - R_b)$ is the tracking error, which is the standard deviation of the difference between portfolio returns and benchmark returns. The Information Ratio is particularly useful for comparing the performance of active investment strategies, as it measures both the magnitude of outperformance and the consistency of that outperformance.

In this thesis, the Information Ratio is used to assess the added value of the machine learning-based portfolios compared to the traditional portfolios and the benchmark index (the OBX index). A higher Information Ratio indicates that the portfolio not only achieves higher excess returns but does so with consistent risk management. By using the Information Ratio, it is possible to determine whether machine learning models provide a sustainable advantage over traditional approaches in terms of both performance and risk control.

## 2.10 Sortino Ratio

The Sortino Ratio is a modified version of the Sharpe Ratio, used to evaluate the risk-adjusted performance of an investment while only considering downside risk. Unlike the Sharpe Ratio, which penalizes both upside and downside volatility, the Sortino Ratio focuses solely on the negative deviations from a defined acceptable return (often zero or the risk-free rate). This makes it particularly suitable for investors who are more concerned with downside risk than overall volatility. The formula for the Sortino Ratio is given by:

$$S = \frac{E(R_p) - R_f}{\sigma_d}$$

*Equation 16: Sortino Ratio*

Where $S$ represents the Sortino Ratio $E(R_p)$ is the expected return of the portfolio, $R_f$ is the risk-free rate, and $\sigma_d$ is the standard deviation of the negative asset returns. By focusing on downside risk, the Sortino Ratio provides a clearer picture of how effectively each portfolio limits losses, making it a valuable measure for risk-averse investors. A higher Sortino Ratio indicates that the portfolio delivers returns while minimizing the risk of significant losses.

## 2.11 Fama French Carhart

Fama and French (1993) expanded on the CAPM model by including size and value factors which could explain portfolio returns. Based on the empirical findings of Banz (1981), Basu (1983) and Rosenberg et al (1998), Fama and French created the three-factor model, which could be explained by:

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

*Equation 17: three-factor model*

$R_{it}$ is the risk premium at time $t$. $\alpha$ are the returns not explained by the model, $R_{mt}$ is the risk premium of the market at time $t$. $\beta_1 . \beta_2, \beta_3$ are the factor coefficients. SMB (small minus big) is the size factor, HML (high minus low) is the book to market factor, and $\sum\epsilon$ is the regression error.

The small minus big factor captures the additional return investors have historically received from investing in stocks of smaller companies compared to larger companies. It is calculated as the difference in returns between a portfolio of small-cap stocks and a portfolio of large cap stocks. SMB is given by.

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

*Equation 18: SMB*

where $S$ is the market value of the small cap stocks in the portfolio, $B$ are the companies is the portfolio with a large market cap. $H$ are the stocks in the portfolio with a high book-to-market equity ratio, $M$ is the stocks with an average book-to-market equity ratio. $L$ are the socks with low book-to-market equity ratio. The high minus low factor captures the higher returns that investors have historically received from investing in stocks with high book-to-market values, also called value stocks, compared to stocks with low book-to-market values, also called growth stocks. HML is given by.

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

*Equation 19: HML*

Fama and French (1993) argue that although their factors don't derive from theoretical frameworks such as modern portfolio theory or equilibrium models, they effectively proxy common risk factors due to their reflection of economic fundamentals. They observed that companies with high book-to-market ratios typically have lower earnings on assets and are riskier, thus demanding higher returns—a finding also supported by Rosenberg et al. (1998). Conversely, companies with high market-to-book ratios tend to exhibit higher earnings. Additionally, larger companies, which are more resilient during economic downturns, showed higher earnings in the period analysed by Fama and French.

Carhart (1997) expands on the Fama and French model by incorporating a factor commonly known as the momentum factor. This addition draws on the anomaly identified by Jegadeesh and Titman (1993), who demonstrated significantly higher market returns through a trading strategy that involved buying stocks with strong recent performance and selling those with poor performance. This led to the development of Carhart's four-factor model.

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

*Equation 20: Carhart's four-factor model*

The winners-minus-losers (WML) factor employs the method used by Jegadeesh and Titman (1993), constructing a portfolio that sells stocks with the lowest returns and buys those with the highest over the previous year. Specifically, "losers" are defined as the bottom 30% of the return distribution from the prior year, and "winners" as the top 30%.

The efficacy of different factors on stock returns has been empirically validated, positioning them as common market effects. This validation allows the three- or four-factor model to be used as analytical tools in evaluating the performance of specific portfolios. By conducting a linear regression of a portfolio's returns against the factors included in these models, one can discern characteristics of the portfolio. The model's regression coefficients can indicate whether a portfolio's returns stem from investments in, for instance, smaller companies, value companies, or growth companies.

Additionally, the regression intercept, also known as alpha, is significant as it represents the portion of returns not explained by common market factors, often reflecting the impact of the investor's choices. A notable intercept suggests that the investor's strategy has either positively or negatively influenced the portfolio beyond what can be attributed to common factors.

## 2.12 Potential Pitfalls or Risks of the Theories/Models

While the models and approaches discussed in this thesis offer powerful tools for portfolio optimization, they also come with inherent risks and limitations that must be considered.

**Minimum Variance Portfolio (MVP)**: The MVP relies on the assumption of normality in asset returns, which may not hold in real financial markets where returns often exhibit skewness and kurtosis (Cont, 2001). Furthermore, the MVP's effectiveness is contingent on accurate estimation of the covariance matrix, which can be challenging in practice due to limited data availability or changing market conditions (Ledoit & Wolf, 2004).

**Machine Learning Models**: Machine learning models such as Random Forests, SVMs, KNN, and Gradient Boosting Machines are prone to overfitting, especially when applied to noisy and non-stationary financial data (Zhang et al., 2020). Overfitting can lead to models that perform well on historical data but poorly on unseen future data. This risk necessitates careful use of regularization techniques, cross-validation, and feature selection to ensure that models generalize well (Hastie et al., 2009).

**GARCH Models**: GARCH models often used for modeling volatility, but they also have limitations. They assume that volatility follows a specific autoregressive pattern, which may not always capture sudden market shifts (Poon & Granger, 2003). Parameter estimation for GARCH models can be sensitive to the chosen sample period, leading to instability in predictions (Bollerslev et al., 1992).

**ARIMA-GARCH Models**: The ARIMA component used to predict return direction assumes linear relationships in time series data, which may not fully capture the complexities of financial markets (Tsay, 2010). Additionally, the reliance on past data for both ARIMA and GARCH models means that they can be slow to respond to sudden changes or new information, making them less effective during periods of market turbulence.

By acknowledging these risks and limitations, this thesis aims to provide a balanced view of the models used, emphasizing both their potential advantages and the challenges that come with their application in real-world financial contexts.

# 3 Methodology:

## 3.1 Stock information:

The stocks selected for this study are those included in the OBX index, which comprises the 25 most traded securities on the Oslo Stock Exchange. The stock datasets is collected from the TITLON database and range from 01.05.2016 – 09.24.2019. The OBX index is a benchmark for Norwegian equity markets, representing a diverse set of industries that reflect the overall performance of the Norwegian economy.

The OBX index includes companies across various sectors, such as energy, finance, consumer goods, and telecommunications, providing a diversified selection of stocks. Given the high representation of energy companies the index also has an inherent sectoral bias. This bias towards the energy sector makes risk management a crucial part of the portfolio construction process, particularly in relation to market fluctuations driven by energy prices. The composition of the OBX is periodically revised to ensure that it reflects the most traded and liquid stocks, which adds a dynamic aspect to the index and impacts portfolio strategy adjustments over time.

## 3.2 Commodities:

**Brent Crude Oil (BZ=F)**: Brent is one of the major global oil benchmarks, representing the pricing of oil extracted from the North Sea. Its price movements are highly correlated with global energy market dynamics and have a significant impact on the overall Norwegian economy, given the country's strong connection to oil production.

**Crude Oil (CL=F)**: Crude oil is a critical component of the global energy supply. The inclusion of Crude Oil prices allows for a more comprehensive assessment of the impact of energy commodities on portfolio risk and return.

**Natural Gas (NG=F)**: Natural Gas is another important energy commodity.. Natural Gas is included to diversify the energy component of the portfolio.

**Silver (SLV)**: Silver, often used as both an industrial and precious metal, provides diversification beyond energy commodities. It has historically been considered a hedge against inflation and a store of value, making it valuable for portfolio risk management.

**Energy ETF (CHIE)**: The China Energy ETF provides exposure to the energy sector in one of the world's largest and fastest-growing markets. This ETF is used to add an international diversification component to the portfolio's energy exposure.

Additionally, exchange rate data is collected for the **EUR/USD (EURUSD=X)**, as exchange rate movements can significantly impact commodity prices and, subsequently, portfolio performance. By including this data, the analysis can account for currency risk, particularly important for investors exposed to multiple currencies.

The inclusion of these commodities and ETFs aims to enhance the predictability of the algorithms. The dataset is constructed using historical data sourced from Yahoo Finance, spanning several years, which ensures that the models have sufficient data to train effectively across different market environments and cycles.

## 3.3 Analysis:

The analysis conducted in are evaluating the effectiveness of different portfolio construction methodologies, including traditional financial and modern approaches using machine learning and advanced statistical techniques. The core objective of this analysis is to determine whether the use of machine learning models provides significant advantages in enhancing the risk-adjusted returns of a portfolio consisting of OBX index stocks.

The analysis is divided into several phases:

**Data Preprocessing**: The first phase involves preprocessing the data collected from various sources, including TITLON for stocks, Yahoo Finance for commodities, and other relevant indices. This step ensures that the data is cleaned, free of missing values, and normalized where necessary to facilitate model training and analysis.

**Model Training and Validation**: The second phase is focused on training different machine learning models like Random Forest, SVM, GBM and KNN. These models are trained using historical data, and hyperparameters are tuned using cross-validation to optimize model performance. Each model is evaluated based on its ability to predict key metrics such as return direction, volatility, and correlation among assets. Performance metrics like Root Mean Squared Error (RMSE) and accuracy are used to assess the quality of predictions.

**Portfolio Optimization**: In the third phase, the predictions from each model are used as inputs for portfolio optimization. The optimization process involves calculating the covariance matrix of predicted returns and using quadratic programming to determine the optimal asset weights that achieve the desired risk-return balance.

**Performance Evaluation**: The constructed portfolios are then evaluated using the performance metrics, Sharpe Ratio, Information Ratio, and Sortino Ratio. These metrics provide insight into the risk-adjusted performance of each portfolio, helping to compare the benefits of using machine learning models versus traditional optimization techniques. Additionally, the performance of the portfolios is analysed across different market conditions to assess robustness.

**Comparative Analysis**: This multi-step analysis provides a thorough examination of the capabilities and limitations of the various approaches under consideration. By comparing traditional and machine learning-based models, this thesis aims to provide a deeper understanding of how modern computational techniques can be leveraged to enhance financial decision-making and portfolio management.

## 3.4 ARCH and GARCH model to create a correlation matrix.

The GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model plays an essential role in this thesis by providing a robust method for modeling the volatility and correlation structure between the assets included in the portfolio. A well-defined correlation matrix is critical in constructing an optimal portfolio, as it directly affects the calculation of the covariance matrix.

The GARCH model (Bollerslev, 1986) builds on the concept of modeling volatility as an autoregressive process, allowing for the persistence of volatility over time. The GARCH (1,1) model, which is widely used in practice, incorporates both past residuals and past variances to predict future volatility. This ability to account for longer-term trends in volatility makes GARCH a preferred model for financial time series, especially when attempting to forecast volatility and correlation among multiple assets.

In this thesis, GARCH models are employed to estimate both individual asset volatilities and the correlations between them, providing a comprehensive view of the dynamics at play within the portfolio. By applying GARCH to each asset, a conditional variance series is generated, which is subsequently used to construct the correlation matrix. This correlation matrix is essential for creating the covariance matrix that forms the basis of the MVP.

The correlation matrix derived from GARCH is particularly useful in capturing time-varying relationships between assets. Unlike static correlations, which assume constant relationships over time, GARCH-derived correlations can adapt to changing market conditions, such as periods of heightened economic uncertainty or market crashes. This flexibility is crucial for managing risk in a diversified portfolio.

The correlation matrix is constructed using the rolling estimates of variances and covariances obtained from the GARCH models. This approach helps in adapting the portfolio to recent market conditions, providing a more realistic representation of the current risk structure. By accounting for the dynamic nature of asset correlations, the resulting covariance matrix ensures that the portfolio optimization process is grounded in a more accurate reflection of market risks, thereby contributing to a more resilient and well-diversified portfolio.

## 3.5 Model specifications

The models used in this thesis, including Random Forest, Gradient Boosting Machines, Support Vector Machines, and k-Nearest Neighbours, are carefully configured based on academic literature and iterative experimentation. Each model's hyperparameters are selected to achieve a balance between predictive accuracy and computational efficiency, while also avoiding overfitting and underfitting.

### 3.5.1 Random Forest Specifications

The Random Forest model for volatility prediction (regression) was specified as follows: *Number of Trees (ntree):* 500. The number of trees is critical as too few can reduce model accuracy, while too many can lead to high computational costs without significant improvement (Liaw & Wiener, 2002). *Number of Predictors (mtry):*4. This value represents the number of predictors randomly selected at each split. This helps control model complexity and ensures that splits are diverse enough to prevent overfitting (Breiman, 2001). *Variable Importance (importance):* True. This setting specifies that the importance of variables should be assessed during training to help understand which features contribute most to the model's predictions (Strobl et al., 2007). *Additional Specifications:* The model also employed bootstrap sampling, out-of-bag (OOB) error estimation for validation, and a fixed random seed for reproducibility (Breiman, 2001).

For classification tasks, the same features were used, with an additional *parameter type = class* to ensure the model managed the task appropriately as a classification problem (Liaw & Wiener, 2002).

### 3.5.2 Gradient Boosting Machines (GBM) Specifications

The Gradient Boosting Machines (GBM) were specified as follows: *Distribution*: Gaussian for regression tasks and Bernoulli for classification tasks, to appropriately model the distribution of the target variable (Friedman, 2001). *Interaction Depth*: Set to 2, controlling the complexity of the individual trees. This is a conservative value intended to prevent overfitting (Hastie, Tibshirani, & Friedman, 2009). *Learning Rate (Shrinkage)*: 0.003. A small learning rate helps stabilize the boosting process, gradually improving the model's fit to avoid overfitting (Chen & Guestrin, 2016). *Bag Fraction: 0.5,* indicating that 50% of the data is used for each iteration, which aids in reducing overfitting (Friedman, 2002). Training Fraction: Set to 1, ensuring that all data is used for training. *Minimum Observations per Node (n.minobsinnode):* 20, providing a balance between capturing trends and avoiding noise (Friedman, 2001). *Cross-Validation (CV Folds):* Set to 1, used mainly for baseline estimation (Hastie et al., 2009).

### 3.5.3 Support Vector Machine (SVM) Specifications

The Support Vector Machines (SVM) were trained using different specifications for regression and classification: *Regression:* The epsilon-regression method was used, which helps control the sensitivity of the model to small errors (Smola & Schölkopf, 2004). *Classification*: The C-classification method was used Cortes & Vapnik, (1995). *Kernel:* Radial Basis Function (RBF) was chosen for both models, with a cost parameter of 10 and a gamma value of 0.1. These hyperparameters were selected to balance between model flexibility and avoidance of overfitting (Hsu, Chang, & Lin, 2010).

### 3.5.4 k-Nearest Neighbors (kNN) Specifications

The k-Nearest Neighbors (kNN) model's hyperparameters were tuned to optimize model performance: **Number of Neighbours (k)**: Tuned over a set {5, 7, 9} to find the optimal value. Different values were tested to determine which provided the best trade-off between bias and variance (Peterson, 2009). **Grid Search for Regression**: A grid search was performed to identify the best hyperparameters for the regression model (Hastie, Tibshirani, & Friedman, 2009), while classification tasks used direct tuning (Bishop, 2006).

### 3.5.5 ARIMA-GARCH Model Specifications

The ARIMA component was used to predict return direction, while the GARCH component estimated volatility. These models were specified based on best practices in financial econometrics (Engle, 1982) and were tuned iteratively to identify the optimal parameters (Bollerslev, 1986; Tsay, 2010).

### 3.5.6 General Model Specification Approach

All model specifications were developed based on a combination of academic literature and trial and error. The final settings were selected to ensure model performance while avoiding both overfitting and underfitting. Cross-validation was utilized extensively during hyperparameter tuning to ensure that each model generalizes well to new data. By balancing model complexity with robustness, this thesis aims to produce a well-validated framework for predicting stock returns and constructing optimal portfolios.

## 3.6 Optimal weights and covariance matrix

The approach for constructing optimal weights is based on Markowitz's Mean Variance Portfolio. The objective of the portfolio weights is to find the combination that maximizes returns while minimizing portfolio risk, measured by volatility. These weights were calculated using quadratic programming in R with the following constraints:

$$\omega_i \geq 0, \forall_i = 1, \ldots, n$$

*Equation 21: quadratic programming*

This constraint prevents short selling, ensuring that no position has a negative weight, which would imply borrowing stocks to sell, an approach not considered suitable for this portfolio.

**Sum-to-One Constraint**: The weights of all assets must sum to 1, ensuring that the entire portfolio is fully invested. Mathematically, this is expressed as:

$$\sum_{i=1}^{n} \omega_i = 1$$

*Equation 22: sum to one constraint*

**Conditional Weight Constraint**: Based on the probability of a positive return. If the machine learning algorithms predict the probability of a positive return for the asset to be 50% or less, the weight for that asset is set to zero. This can be represented as:

$$\omega_i = 0 \; if \; ProbPosRet_i \leq 0.5$$

*Equation 23: Conditional Weight Constraint*

This constraint ensures that only assets with a positive outlook are included in the portfolio, contributing to a more conservative strategy that focuses on minimizing downside risks.

The objective function being minimized is expressed as:

$$\frac{min}{w} w^{\mathrm{T}} \Sigma_w$$

*Equation 24: objective function*

where $w$ is the vector of weights, and $\Sigma$ is the covariance matrix for that week's returns. The covariance matrix is derived from the GARCH models, capturing the time-varying correlations and volatilities among the different assets.

Quadratic programming was used to solve this optimization problem, ensuring that the derived weights provide the best possible balance between risk and return according to Markowitz’s framework. The implementation in R includes tools such as quadprog to handle the optimization problem with multiple constraints effectively. By including specific rules against short selling and requiring sum-to-one constraints, the constructed portfolio aims to maintain a high level of stability while being fully invested.

Additionally, by incorporating machine learning predictions into the weight optimization process, this approach attempts to blend traditional financial theory with modern computational techniques to enhance decision-making. The combination of these methodologies ensures that the resulting portfolio not only considers historical risk patterns but also adapts dynamically based on forward-looking predictions provided by machine learning models.

# 4 Results

## 4.1 Result introduction

In this chapter, the results of the portfolio construction and optimization processes are presented in detail. This chapter will address how well the models performed in constructing a portfolio for the OBX index, aiming to enhance risk-adjusted returns.

The evaluation includes metrics for both regression and classification models, such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and various classification metrics like accuracy, precision, recall, and F1 score. Additionally, the cumulative returns of portfolios over a 200-week period are presented, along with key risk-adjusted performance metrics such as Sharpe Ratio, Information Ratio, Tracking Error, and Beta. The chapter concludes with a comparative analysis of the 8 best-performing models out of an initial set of 50, including insights from a Fama-French Carhart factor.

## 4.2 Regression model evaluation.

The volatility forecasting and stock direction forecasting is solved using two different machine learning specifications, respectively regression and classification. Hence it is necessary to use different methods of model evaluation. For the regression models they will be evaluated by retrieving the RMSE and MAE. Root mean square error or RMSE is a metric which gives the average distance between the predicted values and the actual values. The result should be between 1 and 0 where closes to 0 is the preferred result. RMSE is given by:

$$\text{RMSE} = \sqrt{\Sigma(X_i - Y_i)^2 / \text{n}}$$

*Equation 25: RMSE*

Where $X_i$ is the predicted value and $Y_i$ is the observed value. Mean absolute error or MAE measures the average absolute difference between the predicted values and the actual values.

Unlike RMSE, MAE does not square the errors, which allows for equal weights to every error. MAE is useful when the goal is to understand the error without the considerations of over- or underestimations. MAE is given by:

$$MAE = \frac{1}{n}\sum_{i=1}^{N} |y_i - \hat{y}\,|$$

*Equation 26: MAE*

Where $y_i$ is the actual value and $\hat{y}$ is the predicted value. Below are the results of the RMSE and the MAE from the machine learning portfolios.

*Table 1: MAE and RMSE results*

| Volatility ML/ARIMA model evaluation | | |
|---|---|---|
| | RMSE | MAE |
| RegularGARCH | 0.0124 | 0.0111 |
| GJRGARCH | 0.0051 | 0.0049 |
| IGARCH | 0.0144 | 0.0110 |
| RandomForest_RW3 | 0.0048 | 0.0039 |
| RandomForest_RW6 | 0.0046 | 0.0035 |
| RandomForest_RW9 | 0.0034 | 0.0029 |
| SVM_RW3 | 0.0098 | 0.0091 |
| SVM_RW6 | 0.0048 | 0.0043 |
| SVM_RW9 | 0.0049 | 0.0042 |
| GBM_RW3 | 0.0099 | 0.0099 |
| GBM_RW6 | 0.0099 | 0.0091 |
| GBM_RW9 | 0.0101 | 0.0092 |
| KNN_RW3 | 0.0091 | 0.0078 |
| KNN_RW6 | 0.0126 | 0.0119 |
| KNN_RW9 | 0.0103 | 0.0101 |
| Naive_Baseline_RW3 | 0.0138 | 0.0128 |
| Naive_Baseline_RW9 | 0.0069 | 0.0062 |
| Mean_Baseline_RW3 | 0.0075 | 0.0067 |
| Mean_Baseline_RW9 | 0.0067 | 0.0061 |

To evaluate the portfolios, it is useful to create baseline variables for comparison. Since the chosen portfolios have different rolling window volatility estimations, two baseline variables of each RW volatility was estimated. Respectively, Naïve and Mean. Naïve uses the last observed value at the prediction for all instances in the dataset. Mean baseline calculates the mean volatility from the data and use this as the constant prediction value for every instance in the dataset. If the ML models significantly outperform the baselines, the predictions are capturing useful patterns in the data. From the table we can see that there are varying performances from the ML portfolios, with RandomForest_RW9 being the one with lowest, which is the best, RMSE and MAE score. Furthermore, is apparent that the RF and SVM portfolios performed more accurately overall than the KNN and GBM portfolios. Although it's worth mentioning that most of the ML portfolios perform well compared to the baselines. The GARCH models, which account for the volatility element of the ARIMA portfolios perform varying, where the Integrated GARCH performed insufficiently and the GJRGARCH performed good compared to the baselines and the ML portfolios.

## 4.3 Classification model evaluation

The classification models forecast on a binary variable; hence the results will be binary where 1 = Positive return forecast and 0 = Negative return forecast. These results are printed in a confusion matrix where each prediction can have one of four outcomes, true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Where true positive and true negative is the preferred results. From the confusion matrix different measurements of model performance can be calculated. This thesis will use accuracy, precision, recall and F1-score for evaluation.

Accuracy is measured by $\frac{TP + TN}{TP + TN + FP + FN}$ where it's the correct prediction divided by ever prediction. Accuracy gives a percentage return which indicates how often the model predicts the correct result. Precision is measured by $\frac{TP}{TP + FP}$ where it's the correct positive prediction divided by every positive prediction. Precision shows how often the model can predict a positive return correct. Recall is measured by $\frac{TP}{TP + FN}$ where the correct positive prediction is divided by every positive day. Recall shows how many positive predictions the model can predict. F1-score is measured by $2 * \frac{Recall * Precision}{Recall + Precision}$ and calculates the geometric returns of recall and precision. F1-score is an alternative to accuracy.

| Model evaluation: classification models | | | | |
|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1-Score |
| RF | 0.57 | 0.66 | 0.67 | 0.67 |
| SVM | 0.53 | 0.58 | 0.57 | 0.58 |
| KNN | 0.46 | 0.51 | 0.52 | 0.50 |
| GBM | 0.52 | 0.57 | 0.56 | 0.57 |
| ARIMA | 0.29 | 0.35 | 0.32 | 0.33 |

*Table 2: Classification evaluation*

The table above shows that the RF portfolios preform best, with an average accuracy of 57%. We can also see that the RF model performs best on precision, recall and F1-score. SVM is the next best with and average accuracy of 53%, GBM has an average accuracy of 52%, KNN has an average accuracy of 46% and ARIMA has an average accuracy of 29%. The ARIMA accuracy is low, but that can be explained by the fact that it predicted negative returns for every week of the test set for 9 stocks.

## 4.4 Cumulative returns analysis

This section presents the cumulative returns for the portfolios constructed using the MVP portfolios over a 200-week period. The cumulative returns analysis serves as a good metric for evaluating the long-term growth of an investment and provides insight into how effectively each model captures market trends and adjusts to changing conditions.

### 4.4.1 Cumulative Return Performance for Selected Models

To illustrate the effectiveness of each MVP portfolio constructed using different machine learning techniques, cumulative return plots are provided for each of the four main machine learning models Random Forest, Gradient Boosting Machine, Support Vector Machine, and k-Nearest Neighbors individually. These individual plots highlight the performance of the MVP portfolios when enhanced by each respective machine learning approach, allowing for a more detailed analysis of how each model influenced overall returns.

In addition to the individual plots, a combined cumulative return plot is provided that includes all four models, benchmarked against the OBX index.

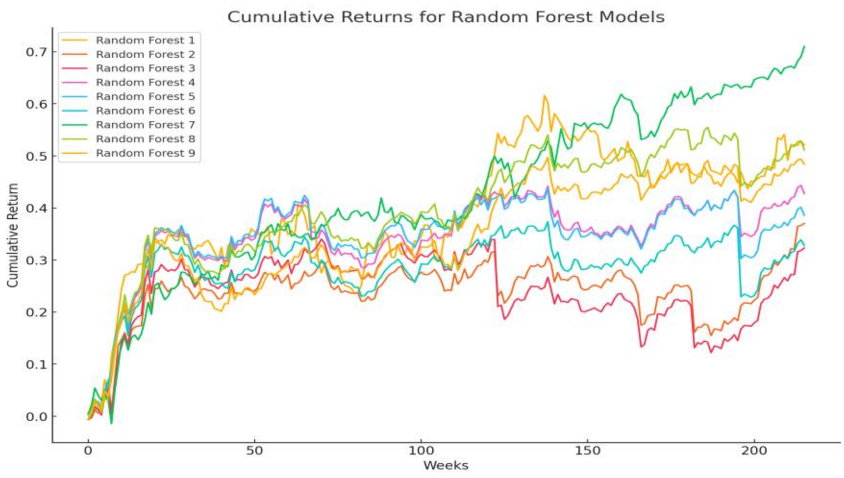


*Figure 4: Cumulative returns for RF*

The graph above illustrates the random forest MVP, as seen in the graph there are a lot of different results. The best performing portfolio were using a rolling window of 9 weeks and a regular GARCH (1,1) model. The worst performing one were using a rolling window of 9 weeks and an IGARCH model. As seen in the graph all the portfolios performed good within the first 0-50 weeks. The models that performed best were able to avoid significant negative shifts. However, they also tended to miss larger positive shifts.

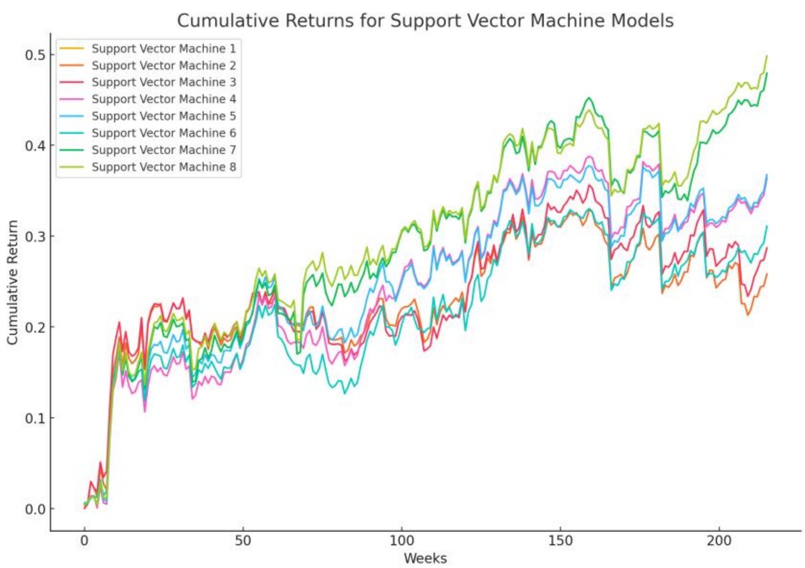


*Figure 5: Cumulative returns for SVM*

The graph above illustrates the SVM minimum variance portfolios. The best performing portfolio were a rolling window with 9 weeks using a regular GARCH (1,1) model. Its apparent that all off the SVM portfolios couldn’t avoid shocks to the negative side.

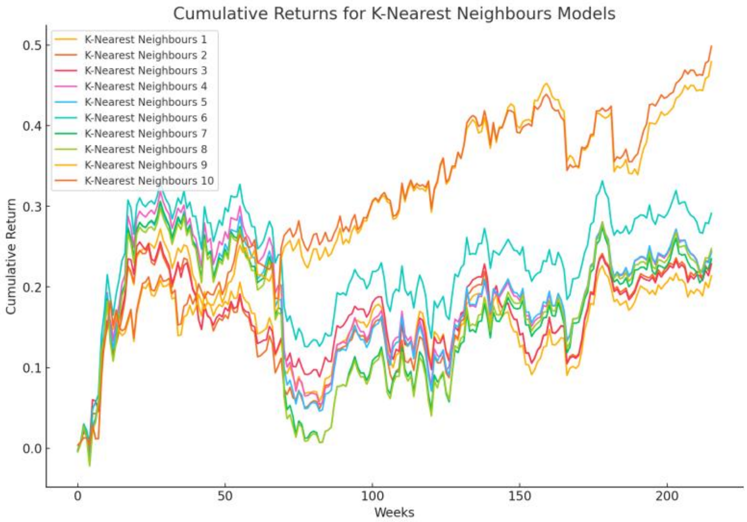


*Figure 6: Cumulative return for KNN*

The graph above illustrates the K-NN minimum variance portfolios. The portfolios that performed best were using a 3-week rolling window volatility and a GARCH (1,1) and IGARCH. Although the returns a relatively low compared to the other models. Seemingly the model did well in avoiding negative shocks.

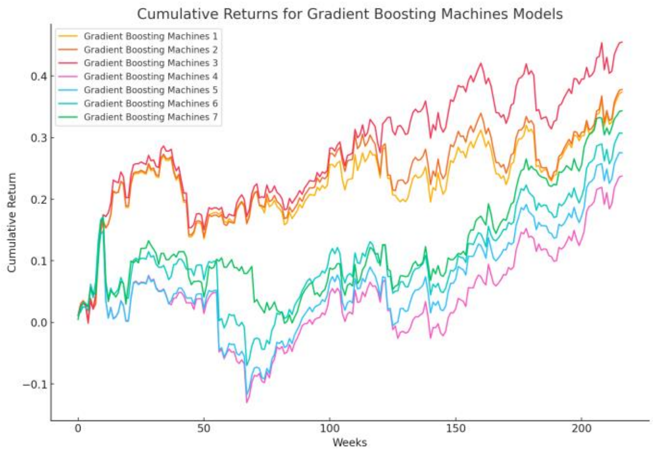


*Figure 7: Cumulative returns for gbm*

The GBM MVP portfolios with a rolling window off three weeks had a strong start compared to the rest. Seemingly the models did not avoid negative shocks well and had rather large periods off losses compared to other models, especially the RF models.

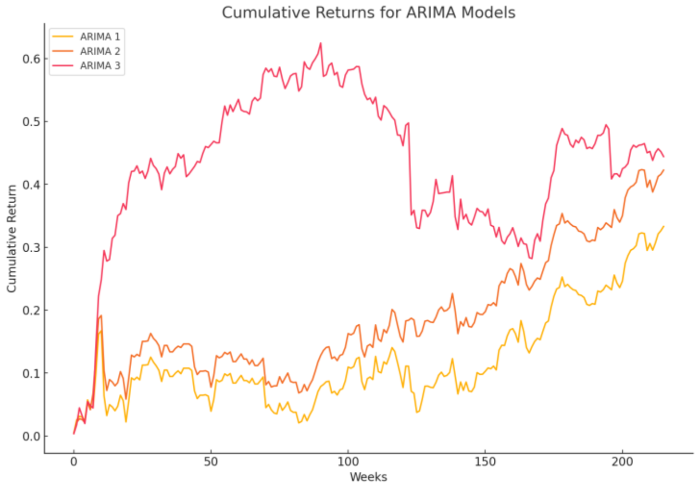


*Figure 8: Cumulative returns for ARIMA*

The ARIMA portfolios above illustrates a different picture than the ML portfolios. One of the ARIMA model using the integrated GARCH model to forecast volatility performed surprisingly well in the first 100 weeks before it dropped by almost 50% in the next 75 weeks. This could be explained by that the predictions from the ARIMA model were similar for every week and therefor it was more exposed to a downshift in one industry, rather than diversifying the risk across industries.

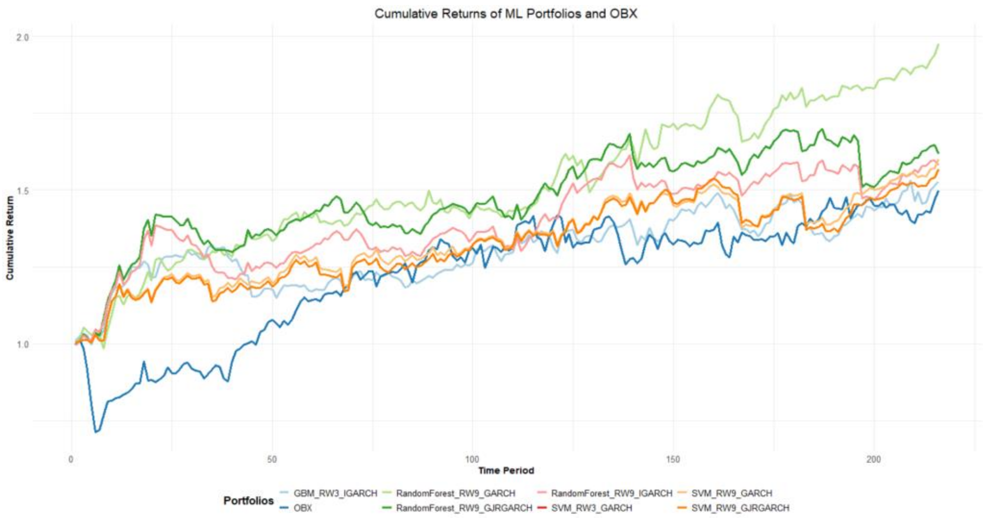


*Figure 9: best performing portfolios compared to benchmark*

The graph above illustrates the best performing portfolios and compares it to the benchmark OBX. The analysis shows that the machine learning-based portfolios—particularly those constructed using Random Forest, Gradient Boosting Machines and Support Vector Machines—demonstrated more substantial cumulative returns compared to ARIMA and the benchmark. This suggests that the predictive power of machine learning models helped in selecting asset weights that adjusted better to market dynamics, thus capturing upward trends more effectively.

### 5.4.2 Comparative Insights

*Machine Learning vs. Traditional Models*: The machine learning models exhibited higher cumulative returns, indicating their ability to dynamically adjust portfolio allocations based on predicted returns and volatility. These models managed to outperform the benchmark during periods of market uncertainty, demonstrating their effectiveness. They also handled negative shocks significantly better than the traditional models.

*Risk-Adjusted Growth:* While the cumulative returns were generally higher for the machine learning models, they also experienced periods of increased volatility. The Gradient Boosting Machine model showed both high cumulative returns and high fluctuations, indicating a potentially higher risk-reward trade-off compared to other models. KNN was the model with the lowest average annual volatility. But again, it also misses the good volatility (positive).

## 4.5 Risk adjusted performance metrics.

This section presents the risk-adjusted performance metrics for the ten MVP portfolios by different machine learning techniques, compared against the OBX benchmark. The evaluation includes Sharpe Ratio, Information Ratio, Tracking Error, Beta, and Sortino Ratio, which provide an overview of how well each portfolio managed risk in relation to their returns over the 200-week evaluation period.

**ML Portfolios Additional Metrics Summary**

| Machine Learning Portfolios | Total Return | Risk Free Rate | Annual Return | Annual STD | Beta | Sharpe Ratio | Tracking Error | Information Ratio | Sortino |
|---|---|---|---|---|---|---|---|---|---|
| RF_RW9_GARCH | 97.71% | 3% | 17.83% | 11.56% | 0.0086 | 1.2834 | 0.2141 | 0.3557 | 0.4132 |
| ARIMA_IGARCH | 61.74% | 3% | 12.27% | 12.08% | 0.0581 | 0.7676 | 0.2093 | 0.0981 | 0.0728 |
| RF_RW9_IGARCH | 58.07% | 3% | 11.65% | 11.22% | 0.0148 | 0.7713 | 0.2113 | 0.0679 | 0.0213 |
| SVM_RW9_GJRGARCH | 56.79% | 3% | 11.43% | 11.84% | 0.0130 | 0.7123 | 0.2150 | 0.0566 | 0.0091 |
| SVM_RW3_GARCH | 52.77% | 3% | 10.74% | 11.95% | 0.0600 | 0.6476 | 0.2082 | 0.0251 | -0.0404 |
| GBM_RW3_GARCH | 69.02% | 3% | 13.47% | 13.31% | -0.0229 | 0.7863 | 0.2287 | 0.1421 | 0.1456 |
| SVM_RW9_GARCH | 56.79% | 3% | 11.43% | 11.84% | 0.0130 | 0.7123 | 0.2150 | 0.0566 | 0.0091 |
| RF_RW9_GJRGARCH | 60.00% | 3% | 11.98% | 11.60% | 0.0086 | 0.7745 | 0.2143 | 0.0822 | 0.0426 |
| SVM_RW9_GJRGARCH_2 | 53.84% | 3% | 10.93% | 11.65% | 0.0025 | 0.6806 | 0.2155 | 0.0329 | -0.0253 |
| Benchmark | 49.80% | 3% | 10.22% | 18.18% | 1.0000 | 0.3971 | 0.0000 | 0.0000 | 0.0000 |

*Table 3: portfolio evaluation*

### 4.5.1 Overview of Metrics and Results

*Total Return:* The total returns for the portfolios range from 97.71% (Portfolio 1) to 52.77% (Portfolio 5), compared to 49.80% for the OBX benchmark. Portfolio 1 achieved the highest total return, significantly outperforming the OBX.

*Annual Return:* The annualized return ranged from 17.83% for Portfolio 1 to 10.22% for the OBX benchmark. Portfolios 1, 2, and 6 had the highest annual returns, showcasing their effectiveness in generating consistent growth over the 200-week period.

*Annual Standard Deviation:* The annual standard deviation, indicating portfolio volatility, ranged from 11.22% to 13,31% The OBX benchmark had the highest annual standard deviation (18.18%), suggesting that the portfolios enhanced by machine learning and ARIMA had lower volatility.

*Sharpe Ratio:* The Sharpe Ratio for each portfolio was calculated to assess risk-adjusted returns. The highest Sharpe Ratio observed among the selected models was 1.2834 (Portfolio 1), significantly higher than the OBX's Sharpe Ratio of 0.3971. Higher Sharpe Ratios indicate better risk-adjusted performance.

*Tracking Error*: which measures the deviation of portfolio returns from the benchmark, ranged from 0.2082 to 0.2287. Portfolios with higher tracking error, such as Portfolio 6, typically deviated more aggressively from the benchmark, which resulted in higher potential returns but also increased the risk of underperformance during adverse conditions.

*Information Ratio:* which measures the active return relative to the benchmark adjusted for tracking error, was highest for Portfolio 1 (0.3557). This indicates that Portfolio 1 not only outperformed the benchmark but did so with a relatively consistent approach, thereby reflecting successful active management.

**Beta**: Beta values for the portfolios ranged from 0.0086 to 1.0000. Portfolio 6 had a slightly negative beta (-0.0229), suggesting a minor inverse relationship with the market. This low beta behavior suggests that most machine learning-enhanced portfolios had less exposure to overall market risk compared to the benchmark, which helped cushion volatility during market fluctuations.

**Sortino Ratio**: The Sortino Ratio, which focuses specifically on downside risk, ranged from 0.4132 to -0.0404. Portfolio 1 again led with a Sortino Ratio of 0.4132, showing its ability to generate returns while managing downside risk effectively. On the other hand, Portfolio 5 had a negative Sortino Ratio (-0.0404), indicating poor risk-adjusted performance when considering only downside volatility.

### 4.5.2 Comparative Analysis of Risk-Adjusted Metrics

A detailed comparison of the risk-adjusted performance metrics reveals several insights:

*Performance Variability:* Portfolio 1 stands out as the best performer in terms of both total return and risk-adjusted performance (Sharpe and Sortino Ratios). The enhanced ability to mitigate downside risk while achieving substantial growth demonstrates the effectiveness of machine learning models in adapting to changing market conditions.

*Tracking Error and Active Management:* Portfolios with higher tracking errors, such as Portfolio 6 (0.2287), displayed more aggressive active management. This often paid off in terms of achieving high returns but came at the cost of increased volatility. Portfolio 1, with a slightly lower tracking error but higher Sharpe and Information Ratios, achieved a better balance between active management and consistent returns.

*Market Sensitivity:* The beta values for most portfolios were significantly lower than 1, indicating limited sensitivity to market-wide movements. Portfolio 6's negative beta suggests a decoupled or inverse relationship with market trends, which might have contributed to stability during market downturns but also reduced its ability to benefit from bull markets.

*Downside Risk Management:* The Sortino Ratios varied significantly across the portfolios, highlighting differences in downside risk management. Portfolio 1 performed the best in terms of controlling downside risk while still generating positive returns. Portfolios with negative Sortino Ratios (such as Portfolio 5) underperformed when considering downside volatility specifically, indicating areas where model performance could be improved.

## 4.6 FAMA FRENCH CARHART four factor model

This section presents the evaluation of the selected portfolios using the Fama-French Carhart factor model. The Fama-French Carhart model extends the traditional Fama-French three-factor model by incorporating a momentum factor, providing a comprehensive way to assess the sources of portfolio returns. The goal of this evaluation is to determine the extent to which systematic factors (market, size, value, and momentum) explain the performance of each of the selected portfolios.

The results are summarized in the following table, which includes the factor loadings for market (MKT), size (SMB), value (HML), liquidity (LIQ) and momentum (MOM), along with the corresponding alpha values for each portfolio. Alpha represents the portion of portfolio return that is not explained by these factors, offering an indication of the skill or value added by the specific portfolio construction methodology. A significant positive alpha suggests that the portfolio generated excess returns beyond those attributable to standard risk factors, which could imply the success of the machine learning-enhanced approach in identifying opportunities that traditional factor models may not capture.

Below, the table provides an overview of the estimated coefficients and their statistical significance for the eight best-performing portfolios. Each coefficient provides insights into the sensitivity of the portfolio to different systematic risk factors, helping to understand which models were most effective at capturing aspects of market performance.

Table 1: Fama-French-Carhart Four-Factor Model Results

| Portfolio | Alpha | Mkt | SMB | HML | MOM | LIQ | $R^2$ |
|---|---|---|---|---|---|---|---|
| rf_rw9_11garch | 0.002636* | 0.006875 | -0.244757. | 0.093451 | -0.129272 | -0.128995 | 0.05743 |
| rf_rw9_gjrgarch | 0.001450 | 0.056540 | -0.217917. | 0.159698 | 0.006308 | -0.359559*** | 0.08823 |
| rf_rw9_igarch | 0.001421 | 0.014179 | -0.296293* | 0.142551 | 0.053782 | 0.331279*** | 0.08904 |
| svm_rw3_11garch | 0.001672 | 0.008312 | -0.500499*** | 0.077836 | 0.089322 | 0.226405* | 0.1292 |
| svm_rw9_11garch | 0.001839. | 0.002078 | 0.458811*** | 0.048113 | 0.095764 | 0.223223* | 0.12 |
| gbm_rw3_igarch | 0.001458 | 0.055453 | 0.161595 | 0.051079 | 0.118162 | 0.152350 | 0.04624 |
| arimagjrgarch | 0.002593* | 0.036811 | 0.255619. | 0.135835 | 0.107315 | 0.112433 | 0.04374 |

*Table 4: Fama French Carhart results*

The Fama-French-Carhart analysis provided valuable insights into the sources of portfolio returns for the selected portfolios. Three portfolios demonstrated statistically significant alpha, indicating that their excess returns could not be fully explained by the market, size

(SMB), value (HML), or momentum (MOM) factors. Among these, the portfolio with the highest alpha achieved a value of 2.6%, highlighting its capacity to generate returns beyond the scope of traditional risk factors.

The portfolios showed varying sensitivities to the overall market, as indicated by their beta coefficients relative to the market factor. While most portfolios had a positive relationship with the market, the magnitude of the factor loadings suggests differing levels of market exposure. Portfolios with higher market beta tended to perform better during bullish phases but were also more vulnerable to market downturns. This highlights the importance of balancing market exposure in constructing robust portfolios.

A significant relationship with the SMB factor was observed in seven out of the eight portfolios, indicating a preference for large-cap stocks. This tilt towards larger companies suggests that the machine learning models, and the optimization process favoured the stability and liquidity associated with large-cap stocks. However, this preference may have limited the potential upside from smaller, higher-growth firms.

All portfolios exhibited a positive but statistically insignificant relationship with the HML factor, suggesting a slight preference for value stocks over growth stocks. The lack of statistical significance implies that the value-growth dynamics did not strongly influence the portfolio returns, potentially because the models prioritized other characteristics, such as momentum or volatility.

The momentum factor varied significantly across portfolios, with some showing strong positive loadings and others demonstrating weak or even negative relationships.

Portfolios with positive momentum relationships likely capitalized on trends in asset prices, aligning with the predictive capabilities of machine learning models. However, those with weaker momentum relationships may have been constrained by volatility forecasts.

The significant alpha values in three portfolios underscore the effectiveness of the portfolio construction methodology in capturing excess returns not explained by systematic factors. This suggests that the models incorporated unique signals or relationships that traditional factor models failed to account for. The machine learning portfolios' ability to generate alpha highlights their potential for identifying complex patterns in financial data, which can lead to improved investment strategies.

Portfolios without significant alpha values likely relied more heavily on systematic factors such as market trends or size preferences. These results indicate that while machine learning models added value, their contributions were sometimes constrained by the overarching market environment or data limitations.

The low $R^2$values across the board indicate that a large portion of the returns remained unexplained by the Fama-French-Carhart model. This limitation suggests that additional factors or alternative modelling frameworks might better capture the nuances of portfolio returns, such as industry-specific variables or macroeconomic indicators.

# 5 Discussion:

The objective for this thesis was to explore the potential for machine learning techniques to generate alpha by optimizing portfolio weights. And to see whether machine learning algorithms generate better portfolios than traditional models. The results section highlights that the eight selected portfolios outperformed the OBX benchmark in both risk and return metrics. The top-performing portfolio utilized the Random Forest model with a rolling window of nine weeks, combined with a regular GARCH model. This portfolio achieved an annualized return of 17.83% and a risk level of 11.56%, significantly outperforming the benchmark, which had an annualized return of 10.22% and a risk level of 18.18%. Over the 200-week period, the Random Forest portfolio delivered a cumulative return of 97.71%, nearly doubling the benchmark's cumulative return of 49.80%. the highest portfolio accumulated a sharp ratio off 1,28 and a Sortino ratio off 0,41, which is broadly considered a good result. The fama French Carhart model showed that the to portfolio generated a statistically significant alpha off 2,6%. The ARIMA portfolios were varying, but the best performing portfolio had a portfolio risk of 12,08%, and an annualized return off 12,27%, which also significantly beats the OBX on both risk and return. The Portfolio also had a Sharp ratio of 0,76 and a Sortino ratio of 0,07. The sharp ratio is within a decent range but the sortino ratio was very low indicating poor risk-adjusted performance when considering only downside volatility. Both the best-performing machine learning (ML) portfolio and the top traditional portfolio demonstrated statistically significant alpha in the Fama-French-Carhart regression, indicating the presence of excess returns not accounted for by the four-factor model. The ML portfolio achieved an alpha of 2.63%, while the traditional portfolio closely followed with an alpha of 2.59%. This suggests that both portfolios were able to generate returns beyond those explained by market risk, size, value, liquidity and momentum factors.

These results underscore the strength of the portfolio construction methodologies employed, highlighting their ability to exploit inefficiencies and opportunities in the market. The slightly higher alpha of the ML portfolio may point to its enhanced capacity to capture complex, non-linear relationships in the data, providing a marginal edge over the traditional approach. Additionally, the statistically significant alpha suggests that both models could offer value to investors seeking strategies that outperform standard benchmarks, even when accounting for widely recognized risk factors.

The GARCH models played a important role in constructing the portfolios by accurately capturing and forecasting volatility, which is crucial for optimizing portfolio weights. However, their performance varied across different configurations. Overall, the regular GARCH (1,1) model, combined with a rolling window of nine weeks, delivered the best results, particularly for the machine learning-based portfolio. For the traditional portfolio, the IGARCH model outperformed the other variants, indicating its strength in modeling persistent volatility over time.

The portfolio with the lowest volatility consistently achieved the best performance, emphasizing the importance of precise volatility modeling in portfolio optimization. This finding reinforces the effectiveness of the GARCH (1,1) model in balancing risk and return, as it provided the most accurate volatility estimates, enabling superior portfolio construction. These results highlight how selecting the appropriate volatility modeling framework can significantly impact overall portfolio performance.

The machine learning portfolios exhibited generally low accuracy, precision, and F1 scores, which aligns with expectations due to the inherent challenge of fitting a single model to stocks from diverse industries. The varying characteristics and dynamics of these industries likely contributed to the models' limited predictive precision across the board. Among the machine learning models, Random Forest (RF) demonstrated the best overall performance, followed by Support Vector Machines (SVM), Gradient Boosting Machines (GBM), and finally K-Nearest Neighbours (KNN).

While RF and SVM models had comparable annualized standard deviations, GBM exhibited slightly higher STD, indicating greater variability in its predictions. KNN, although achieving the lowest STD, struggled to deliver high returns, making it less effective for portfolio optimization. As a result, no KNN-based portfolios ranked among the top eight performing portfolios. This highlights that while KNN was able to minimize risk to some extent, its inability to capture sufficient returns rendered it unsuitable for outperforming the benchmark or other models.

These findings underscore the importance of balancing risk and return when evaluating machine learning models for financial applications. While RF and SVM emerged as robust options, the performance variations among models suggest that their effectiveness may depend on specific portfolio objectives, or the characteristics of the assets being analysed. Future research could explore industry-specific model tuning or hybrid approaches to improve overall predictive performance and portfolio outcomes.

## 5.1 Limitations and Pitfalls

While this study provides valuable insights into portfolio optimization using machine learning and traditional models, several limitations and potential pitfalls must be acknowledged. These challenges highlight areas where further refinement or alternative approaches may be needed to enhance the reliability and applicability of the findings.

Models such as GARCH and ARIMA rely on assumptions of stationarity, linearity, and normality in financial time series data. However, financial markets often exhibit non-linear and non-stationary behaviour, which can limit the predictive accuracy and robustness of these models. GARCH models, for instance, may struggle to fully capture volatility in rapidly changing market conditions or during periods of extreme turbulence.

Machine learning models, especially Random Forest and Gradient Boosting Machines, risk overfitting to historical data, which can reduce their ability to generalize to new, unseen market conditions.

Machine learning models often function as "black boxes," making it difficult to interpret the underlying drivers of their predictions.

The study's analysis was conducted over a limited 200-week period, which may not fully capture long-term market trends or account for extreme events such as financial crises.

The choice of input features, including financial and macroeconomic variables, plays a critical role in model performance. The omission of potentially influential factors due to data unavailability may limit the accuracy and robustness of predictions.

Feature engineering for machine learning models introduces subjectivity, which could inadvertently bias the results.

The use of rolling windows (three, six, and nine weeks) allows the models to adapt to changing market conditions, but it also introduces potential limitations. A short rolling window may lead to overly reactive models that fail to capture longer-term trends. Longer windows may smooth out short-term volatility, reducing the model's ability to respond to sudden market shifts.

While the study employs advanced methods like GARCH for covariance estimation, the results may still be influenced by outliers or extreme correlations during volatile periods. Portfolio optimization assumes that the historical relationships between assets will persist, which may not always hold true, particularly in dynamic and unpredictable markets.

Financial markets are influenced by a wide array of factors, many of which are unpredictable. Machine learning models may fail to adapt to sudden market changes, such as geopolitical events or rapid technological advancements.

Machine learning models, especially ensemble methods like Gradient Boosting Machines, are computationally intensive and require significant resources for training and optimization. This can be a constraint in real-time applications where rapid decision-making is critical. Scaling these models to larger datasets or portfolios with more assets may introduce additional computational and storage demands.

# 6 Conclusion

This thesis aimed to explore the potential of machine learning (ML) techniques in generating alpha by optimizing portfolio weights and to determine whether these approaches can outperform traditional financial models. By integrating advanced ML algorithms such as Random Forest (RF), Support Vector Machines (SVM), Gradient Boosting Machines (GBM), and k-Nearest Neighbours (KNN) with traditional models like ARIMA and GARCH, this research provided insights into the effectiveness of these methods in predicting returns, estimating volatility, and constructing risk-adjusted portfolios for the OBX index.

The machine learning-based portfolios consistently outperformed the benchmark OBX index in terms of both risk and return. The Random Forest portfolio, in particular, demonstrated exceptional performance, achieving an annualized return of 17.83% with a risk level of 11.56%, significantly better than the benchmark's return of 10.22% and risk of 18.18%.

Over the 200-week evaluation period, the Random Forest portfolio achieved a cumulative return of 97.71%, nearly doubling the benchmark's cumulative return of 49.80%.

Portfolios enhanced by ML models demonstrated superior Sharpe and Sortino Ratios compared to traditional models, reflecting their ability to achieve higher returns per unit of risk and better downside risk management.

The highest-performing portfolio achieved a Sharpe Ratio of 1.28 and a Sortino Ratio of 0.41, showcasing the effectiveness of ML in optimizing the risk-return trade-off.

The Fama-French-Carhart analysis revealed statistically significant alpha values for the top-performing portfolios, with the highest alpha reaching 2.6%. This indicates that these portfolios generated returns beyond those explained by traditional market factors, emphasizing the unique value of ML-enhanced strategies.

Traditional models such as ARIMA and GARCH played a critical role in constructing covariance matrices and estimating volatility. While the best-performing ARIMA portfolio achieved decent results (12.27% annual return with a risk level of 12.08%), it fell short of ML-based portfolios in both returns and risk-adjusted metrics.

ML models demonstrated varying levels of accuracy, precision, and F1 scores, reflecting challenges in fitting a single model to stocks from diverse industries. KNN, for example, struggled to balance low risk with high returns, resulting in its exclusion from the top-performing portfolios.

Traditional models like ARIMA were less adaptive to dynamic market conditions, while ML methods occasionally exhibited overfitting and sensitivity to parameter tuning.

The results of this thesis highlight the transformative potential of machine learning in portfolio optimization. By leveraging advanced algorithms, investors can achieve superior returns and manage risk more effectively, even in volatile and uncertain market environments. The ability of ML models to identify complex, non-linear patterns in financial data offers a significant edge over traditional methods, especially in capturing alpha and adapting to evolving market conditions.

Despite its contributions, this study faced limitations, including the relatively short 200-week evaluation period and the sectoral bias inherent in the OBX index. Additionally, while ML models provided robust results, their interpretability remains a challenge, and their reliance on historical data may limit their adaptability during unprecedented market disruptions.

Future studies could address these limitations by:

- Expanding the dataset to include a broader range of indices, sectors, and economic conditions.
- Exploring the integration of alternative ML techniques, such as deep learning models, to enhance predictive accuracy and portfolio performance.
- Investigating hybrid approaches that combine ML and traditional models to balance adaptability, interpretability, and robustness.
- Applying these methodologies to real-world scenarios, assessing their practical feasibility and scalability.

This research demonstrates that machine learning techniques, when combined with traditional financial models, have the potential to be a valuable tool in portfolio management by delivering better performance and uncovering new opportunities for alpha generation. By building on these findings, future advancements in financial technology and data analytics could further refine and enhance the strategies available to investors, paving the way for more efficient and effective financial markets.

# 7 References:

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

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

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer. https://doi.org/10.1007/978-0-387-45528-0

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Bollerslev, T., Chou, R. Y., & Kroner, K. F. (1992). ARCH modeling in finance: A review of the theory and empirical evidence. Journal of Econometrics, 52(1–2), 5–59. https://doi.org/10.1016/0304-4076(92)90064-X

Bollerslev, T., Tauchen, G., & Zhou, H. (2016). Expected stock returns and variance risk premia. The Review of Financial Studies, 22(11), 4463–4492. https://doi.org/10.1093/rfs/hhn114

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

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

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Chen, N. F., Roll, R., & Ross, S. A. (2003). Economic forces and the stock market. The Journal of Business, 59(3), 383–403. https://doi.org/10.1086/296317

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. Quantitative Finance, 1(2), 223–236. https://doi.org/10.1088/1469-7688/1/2/304

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/BF00994018

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987–1007. https://doi.org/10.2307/1912773

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

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. The Journal of Finance, 48(5), 1779–1801. https://doi.org/10.1111/j.1540-6261.1993.tb05128.x

Gu, S., Kelly, B. T., & Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5), 2223–2273. https://doi.org/10.1093/rfs/hhz068

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

Hawley, D. D., Johnson, J. D., & Raina, D. (1990). Artificial neural systems: A new tool for financial decision-making. Financial Analysts Journal, 46(6), 63–72.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications, 124, 226–251. https://doi.org/10.1016/j.eswa.2019.01.012

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. The Journal of Finance, 48(1), 65–91. https://doi.org/10.1111/j.1540-6261.1993.tb04702.x

Kirby, C., & Ostdiek, B. (2012). It's all in the timing: Simple active portfolio strategies that outperform naïve diversification. Journal of Financial and Quantitative Analysis, 47(2), 437–467. https://doi.org/10.1017/S0022109012000109

Kim, K. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1–2), 307–319. https://doi.org/10.1016/S0925-2312(03)00372-2

Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Systems with Applications, 19(2), 125–132. https://doi.org/10.1016/S0957-4174(00)00027-0

Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. Journal of Multivariate Analysis, 88(2), 365–411. https://doi.org/10.1016/S0047-259X(03)00096-4

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R News, 2(3), 18–22.

MathWorks. (n.d.). Understanding support vector machine regression. Retrieved on 03.02.2024, from https://se.mathworks.com/help/stats/understanding-support-vector-machine-regression.html#buyrzdg-1

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

Nelson, D. B. (1990). ARCH models as diffusion approximations. Journal of Econometrics, 45(1–2), 7–38. https://doi.org/10.1016/0304-4076(90)90107-A

OpenAI. (2023). *ChatGPT (March 2023 version)* [Large language model]. Retrieved from https://openai.com/chatgpt

Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883. https://doi.org/10.4249/scholarpedia.1883

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

Poon, S.-H., & Granger, C. W. J. (2003). Forecasting volatility in financial markets: A review. Journal of Economic Literature, 41(2), 478–539. https://doi.org/10.1257/002205103765762743

Rosenberg, B., Reid, K., & Lanstein, R. (1985). Persuasive evidence of market inefficiency. Journal of Portfolio Management, 11(3), 9–16.

RStudio Team. (2022). *RStudio: Integrated development environment for R* (Version 2022.07.1). Boston, MA: RStudio, PBC. Retrieved from https://www.rstudio.com/

Sharpe, W. F. (1966). Mutual fund performance. The Journal of Business, 39(1), 119–138. https://doi.org/10.1086/294846

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

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and Computing, 14(3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics, 8, 25. https://doi.org/10.1186/1471-2105-8-25

Titman, S. (1993). The impact of capital structure on the strategic behavior of firms. Journal of Financial Economics, 5(1), 1–24.

Tsay, R. S. (2010). Analysis of financial time series (3rd ed.). Wiley. https://doi.org/10.1002/9780470644560

Vapnik, V. (1995). The nature of statistical learning theory. Springer. https://doi.org/10.1007/978-1-4757-2440-0

Vapnik, V. N., & Chapelle, O. (2000). Bounds on error expectation for support vector machines. Neurocomputing, 10(3), 183–197.

Yahoo Finance. (n.d.). Financial market data. Retrieved 05.02.2024, from https://finance.yahoo.com

Zhang, D., Jiang, C., Li, X., & Tang, T. (2020). Time-series momentum and volatility scaling in asset allocation. Journal of Financial Markets, 52, 100569. https://doi.org/10.1016/j.finmar.2020.100569

Zhang, T., & Xiao, K. (2013). Improved gradient boosting machines for financial forecasting. Journal of Financial Data Science, 8, 223–245.

## 7.1 References for R Packages

**caret**

Kuhn, M. (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

**randomForest**

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News, 2*(3), 18–22.

**e1071**

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2021). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-9. Retrieved from https://CRAN.R-project.org/package=e1071

**forecast**

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, *27*(3), 1–22. https://doi.org/10.18637/jss.v027.i03

**rugarch**

Ghalanos, A. (2022). *rugarch: Univariate GARCH Models*. R package version 1.4-2. Retrieved from https://CRAN.R-project.org/package=rugarch

**PerformanceAnalytics**

Peterson, B. G., & Carl, P. (2022). *PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis*. R package version 2.0.4. Retrieved from https://CRAN.R-project.org/package=PerformanceAnalytics

**quantmod**

Ryan, J. A., & Ulrich, J. M. (2022). *quantmod: Quantitative Financial Modelling Framework*. R package version 0.4-20. Retrieved from https://CRAN.R-project.org/package=quantmod

**xts**

Ryan, J. A., & Ulrich, J. M. (2022). *xts: eXtensible Time Series*. R package version 0.13. Retrieved from https://CRAN.R-project.org/package=xts

**glmnet**

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22. https://doi.org/10.18637/jss.v033.i01

**GBM**

Ridgeway, G. (2006). Generalized boosted models: A guide to the gbm package. *R vignette*.

**dplyr**

Wickham, H., François, R., Henry, L., & Müller, K. (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9. Retrieved from https://CRAN.R-project.org/package=dplyr

**ggplot2**

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag. https://doi.org/10.1007/978-3-319-24277-4

# iii. Appendix

*Appendix 1: Summary data for every portfolio*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **randomforrest portfolios** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **total return** | 60,65 % | 40,19 % | 33,08 % | 49,88 % | 42,90 % | 35,20 % | 98,81 % | 62,35 % | 58,66 % |
| **rf** | 3,00 % | 3,00 % | 3,00 % | 3,00 % | 3,00 % | 3,00 % | 3,00 % | 3,00 % | 3,00 % |
| **annual ret** | **12,09 %** | **8,47 %** | **7,12 %** | 10,23 % | 8,97 % | 7,53 % | 17,99 % | 12,37 % | 11,75 % |
| **annual std** | 14,98 % | 12,49 % | 13,35 % | 11,61 % | 12,68 % | 12,16 % | 11,53 % | 0,12051 | 0,11193 |
| **sharp ratio** | **0,6069** | **0,43832** | **0,30884** | **0,6230266** | **0,47109** | **0,37247** | **1,300003** | **0,77781** | **0,7821** |
| | **11** | **gjr i** | | **11 gjr** | **i** | | **11 gjr** | **i** | |
| **svm portfolios** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **rf** | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 |
| **TOTAL RETURN** | 56,90 % | 26,16 % | 29,49 % | 40,08 % | 40,68 % | 32,65 % | 56,90 % | 0,60095 | 53,86 % |
| **annual ret** | 11,45 % | 5,75 % | 6,42 % | 8,45 % | 8,56 % | 7,04 % | 0,114536 | 0,11996 | 0,1093 |
| **annual std** | 11,81 % | 0,11288 | 0,11552 | 0,1111429 | 0,11035 | 0,11113 | 0,118145 | 0,11569 | 0,1162 |
| **sharp r** | **0,71553** | **0,24399** | **0,29593** | **0,4905806** | **0,50415** | **0,36336** | **0,715526** | **0,7776** | **0,68248** |
| | **11 gjr** | **i** | | **11 gjr** | **i** | | **11 gjr** | **i** | |
| **knn portfolios** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **rf** | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 |
| **TOTAL RETURN** | 21,69 % | 24,03 % | 23,56 % | 23,08 % | 23,89 % | 30,55 % | 22,55 % | 24,07 % | 30,01 % |
| **annual ret** | 0,04839 | 0,05322 | 0,05224 | 0,0512563 | 0,05292 | 0,06627 | 0,050178 | 0,05329 | 0,06522 |
| **annual std** | 0,10818 | 0,11151 | 0,10738 | 0,1335846 | 0,13527 | 0,12802 | 0,140059 | 0,14068 | 0,13214 |
| **sharp r** | **0,16997** | **0,20823** | **0,20714** | **0,1591221** | **0,16943** | **0,28334** | **0,144065** | **0,16555** | **0,26654** |
| | **11 gjr** | **i** | | **11 gjr** | **i** | | **11 gjr** | **i** | |
| **gbm portfolios** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **rf** | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 |
| **TOTALRETURN** | 40,97 % | 41,46 % | 52,97 % | 22,35 % | 27,03 % | 31,30 % | 36,74 % | 35,45 % | 0,48542 |
| **annual ret** | 0,08617 | 0,08708 | 0,10776 | 0,049761 | 0,05929 | 0,06775 | 0,078238 | 0,07577 | 0,09994 |
| **annual std** | 0,12123 | 0,12231 | 0,11924 | 0,1293213 | 0,12952 | 0,12759 | 0,119998 | 0,12018 | 0,11471 |
| **sharp r** | **0,46335** | **0,4667** | **0,65212** | **0,1528056** | **0,22612** | **0,29588** | **0,401992** | **0,38088** | **0,60974** |
| | 11 gjr | i | | | | | | | |
| **arima** | 1 | 2 | 3 | | | | | | |
| rf | 0,03 | 0,03 | 0,03 | | | | | | |
| totalret | 48,00 % | 69,55 % | -20,39 % | | | | | | |
| annualret | 0,09897 | 0,13553 | -5,34 % | | | | | | |
| annualstd | 0,15668 | 0,13281 | 26,27 % | | | | | | |
| sharp r | **0,4402** | **0,79459** | **-0,31757** | | | | | | |