SCHOOL OF
ECONOMICS AND
MANAGEMENT

Predicting the Movement of the S&P 500 Index using Machine Learning

by

Bakary Bah

August 2023

MSc in Data Analytics and Business Economics

Supervisor: Anders Vilhelmsson

Abstract

Predicting the stock market has been a longstanding topic of interest in financial research. It is regarded as a highly challenging but important task given the vital role the financial markets play in shaping the global economies. In this thesis, the goal is to predict the movement of the S&P 500 Index using machine learning methods. To this end, we apply two machine learning algorithms, random forest, and logistic regression, to financial data in a quest to try and predict if the S&P 500 Index will move in a positive or negative direction the following day. To further test the validity of the best performing machine learning model in our study, we develop a dynamic trading strategy where the predictions of the model act as an investment signal. If the model predicts that the S&P 500 Index will move in a positive direction the following day, we invest in equities (SPDR S&P 500 ETF Trust). Conversely, if the model predicts a negative movement, we instead invest in fixed income (Vanguard Total Bond Market ETF). We assess the performance of the trading strategy by comparing its Sharpe ratio to a second strategy, a traditional portfolio that holds 60% equities and 40% fixed income.

Keywords: Machine Learning, S&P 500 Index, Random Forest, Logistic Regression

Acknowledgements

I would like to express my deepest appreciation to my family and friends for your unwavering support during the process of writing this thesis. I would also like to thank my supervisor, Anders Vilhelmsson. Your feedback and guidance have been highly appreciated.

Lastly, I would like to extend my appreciation to Thorbjörn Wallentin and OQAM Asset Management for the opportunity to write my thesis with OQAM.

Table of Contents

1. Introduction

Predicting the directional movement of the stock market has been a longstanding topic of interest in financial research. It is regarded as a highly difficult but important task given the vital role the financial markets play in shaping the global economies. Portfolio managers are interested in the ongoing research in this field due to the practical implications that would follow (Huenermund, Kaminski & Schmitt, 2021). Investors and asset managers who are able to accurately predict the financial markets are positioned to gain an invaluable advantage in regards to developing investment strategies capable of generating high returns while maintaining low risk, ultimately providing value to their stakeholders.

However, according to the most stringent form of the efficient market hypothesis (Fama, 1970) it is not possible to predict the stock market. The efficient market hypothesis (EMH) suggests that financial markets are efficient, and that all available information already is reflected in the price of a security. This suggests that any attempt to predict market movements based on historical data or other available information would be unsuccessful. Another important theory in financial economics pertaining to the predictability of the stock market is the random walk hypothesis. It extends the notion of the EMH by suggesting that changes in asset prices are completely random and unpredictable. Although these theories provide a foundational framework, they have also influenced a body of literature seeking to test and challenge their stance. Lo and Mackinlay (1988) found evidence supporting the notion that weekly returns do not follow random walks. Drawing from the field of behavioural finance, Lo (2004) introduced the adaptive market hypothesis which suggests that markets can switch between periods of efficiency and inefficiency, ultimately leaving room for market prediction.

Traditional statistical methods such as autoregressive integrated moving average (ARIMA) has been the dominant approach for predicting stock prices and returns (Efendi, Arbaiy & Deris, 2018). Although such time series methods have a solid theoretical foundation, they have a limited capacity to capture patterns in complex non-linear data (Zhong & Enke, 2017).

The emergence of machine learning techniques and the increase in computational power have paved the way for a paradigm shift in regards to stock market prediction. Unlike the traditional statistical methods, machine learning algorithms are able to capture non-linear patterns in complex dataset which makes them an interesting alternative (Chen & Hao, 2017).

In this thesis, the goal is to predict the movement of the S&P 500 Index with the highest accuracy possible to investigate the feasibility of using machine learning in achieving this task. To this end, we apply two machine learning algorithms, as random forest, and logistic regression, to financial data in a quest to try and predict if the S&P 500 Index will move in a positive or negative direction the following day. Thus, we frame our research question as a binary classification problem. Evaluation metrics are then used to assess model performance. Achieving an accuracy above 50% signifies that a model is performing better than random chance. In order to further test the validity of the best performing machine learning model in our study, we develop a dynamic trading strategy where the predictions of the model act as an investment signal. If the model predicts that the S&P 500 Index will move in a positive direction the following day, we invest in equities (SPDR S&P 500 ETF Trust). Conversely, if the model predicts a negative movement, we instead invest in fixed income (Vanguard Total Bond Market ETF). We assess the performance of the trading strategy by comparing its Sharpe ratio to a second strategy, a traditional portfolio that holds 60% equities and 40% fixed income (Rekenthaler & J, 2022). While there are numerous studies that apply machine learning algorithms to predict the movement of the stock market, we contribute to the existing literature by expanding our research beyond prediction accuracy. By developing a simple trading strategy based on the predictions of the best performing model, we include a practical way of analyzing how well the predictions translate into profitable investment decisions.

The remainder of this thesis is structured as follows: In section 2, we provide an in-depth literature review concerning stock market prediction using machine learning. In section 3, we describe the data collection, preprocessing, and transformation steps taken to prepare the dataset. Transitioning to section 4, we provide a detailed description of the machine learning algorithms utilized and the related theory. In section 5, we describe the construction of the investment strategies. In, section 6, we present and discuss the empirical results of our study, before concluding the thesis in section 7.

2. Literature Review

Predicting the stock market is a difficult and challenging task. Nevertheless, there are numerous studies trying to forecast the directional movement of the financial markets. Therniaki and Hoseinzade (2013) deployed an ANN model in an attempt to predict the S&P 500 Index using 27 different economic features. The outcome of the study was positive and showed that the ANN model can outperform traditional econometric methods. In the study of Shen et al. (2012), the importance of the feature selection process is highlighted. The authors include global stock indices, commodities, and exchange rates as features for predicting the directional movement of the S&P 500 Index, Hang Seng Index (HSI), and Deutscher Aktienindex (DAX).

Kumar and Thenmozhi (2011) predicted the daily directional movement of the CNX Nifty Index using random forest and SVM. The feature space in the study consists of 12 various technical indicators such as relative strength index (RSI) and momentum. According to the empirical results, the SVM outperforms the random forest model. In a later study, Ballings et al. (2015) employed several machine learning algorithms such as LR, RF, AdaBoost and SVM, to predict the directional movement of various European equities one year into the future. The results of the study show that the random forest was the best performing model. In the study by Patel et al. (2015), the authors develop four machine learning models: ANN, SVM, random forest, and naïve bays. The partial objective of the study is to predict the directional movement of the CNX Nifty Index using technical indicators as features. In the first approach, the technical indicators are continuous values. In the subsequent approach, they are converted into discreate values. The authors find that the random forest model has the highest accuracy for the first approach. However, the accuracy of all prediction models improves when the features are discreate. Huang (2019) uses the same technical indicators as Patel et al. as features and the same machine learning algorithms to predict the direction of the Taiwan Stock Exchange. Interestingly, the author finds that ANN outperforms the other models, including the random forest.

Fisher and Kraus (2018) deployed an LSTM model using stock price data on equities that are constituents of the S&P Index to predict their directional movement. The authors of the study showed that the LSTM model could derive meaningful insights from financial data. Based on the performance metrics used in the study, the LSTM outperformed the random forest and the logistic regression. Huang et al. (2005) investigated the feasibility of using SVM to predict the weekly directional movement of the NIKKEI 225 Index. The authors compared the SVM

with linear discriminant analysis and other alternatives. However, the outcome of the study showed that the SVM was the best performing model.

Basak et al. (2019) explored the feasibility of predicting the movements of ten individual stocks by using technical indicators together with two machine learning models, random forest, and XGB. Both models performed well, however, the RF outperformed the XGB model. Liu et al. (2016) predicted the daily movement of the S&P 500 Index by using an SVM model with a radial basis function. In their research, the authors utilized exchange rates and commodities as input features. The best performing model achieved an accuracy of approximately 62.51%. Di (2014) investigated the possibility of using machine learning to predict the trend of the S&P 500 Index. As in the previous mentioned study, Di utilized an SVM classification model and 12 technical indicators which provided an accuracy of 56%. In (Wang & Choi, 2013), a hybrid approach that combines principal components (PCA) with SVM in an attempt to predict the Hang Seng Index is investigated. The study uses the power of PCA to conduct feature selection before fitting the SVM model to the selected features. The study provided a prediction accuracy of 62.80%

In light of the reviewed literature, it is evident that this is a challenging but highly relevant field of research. However, most of the literature in this section is focused on the predictive accuracy the models in the particular studies are able to obtain. With this thesis, we hope to contribute to the existing literature by expanding our research slightly beyond only focusing on prediction accuracy. By incorporating a simple trading strategy based on the predictions of the best performing model, we introduce a practical and tangible way of assessing how well the predictions translate into informed investment decisions.

3. Data

This section provides a detailed overview of the data used in the predictive modelling. It details the data collection, preprocessing, and transformation procedures conducted in order to prepare the data for the machine learning models. Furthermore, the rationale behind the feature selection is presented in light of the objective of this study to predict the daily movement of the S&P 500 index.

3.1 Data Collection

The raw data in this research consists of historical prices and levels on financial securities and indices, including the S&P 500 Index. All data is in a daily frequency that ranges from 03.01.2008 to 31.12.2022, resulting in 3783 observations. Furthermore, the data is retrieved from Yahoo Finance using the yfinance API available in Python.

In the following section, we detail the data collected with the intention to serve as predictors or features in this thesis. Shen et al. (Shen, Jiang & Zhang, 2012) incorporate global indices, commodities, and exchange rates as predictors in their research. In this thesis, we follow this approach with the aim of capture market dynamics. The Dow Jones Industrial Average (DJIA) is a benchmark index that reflects the performance of 30 large U.S. companies across different sectors. The inclusion of DJIA is motivated by the expectation that it might capture the broader market sentiment. Large technology stocks have the potential to shape market trends. By including large individual technology companies like Google, Apple, and Amazon, we hope that they can serve as leading indicators for broader market movements, given their substantial weight within the S&P 500. The U.S. Dollar Index (DXY) serves as a proxy for the performance of the U.S. dollar against other major currencies. As currency fluctuations can impact global trade and overall investment sentiment, we include the index as it might be a good indicator for the S&P 500's trajectory. We also include the CBOE Volatility Index (VIX). The VIX measures market volatility and investor sentiment. It is derived from option prices on the S&P 500 Index and signals the overall market's expectations for volatility over the next 30 days. As high VIX values often correlate with market downturns, we hope to capture how changes in the VIX might indicate the directional movement of the S&P 500 Index. The 5-year and 10-year CBOE Treasury Note Yield indices are included to capture changes in interest rate expectations. Lastly, commodities futures like oil, copper, and gold, are included with the motivation being that commodity prices have an

impact on the overall economy and therefore, could be good indicators for predicting the movement of the S&P 500 Index.

3.2 Data Preprocessing

3.2.1 Data Cleaning

Data cleaning is an important step when preparing data intended to be used for predictive modelling. A common issue that researchers face is the challenge of missing values in the dataset. This can bias the results of machine learning models or negatively impact the accuracy of their predictions. In this thesis, all variables that were collected exhibited a minimal number of missing values, ranging between 1 and 4 instances. The proportion of missing values for each feature was less than 0.01% of the total number of observations. Because of the time series nature of the data, we solved this problem by employing the forward-fill technique. This technique uses the most recent observed value for a feature to fill missing values. Thus, it maintains the temporal sequence in our time series data.

3.2.2 Feature Engineering

Previous literature has shown that technical indicators can be useful when predicting stock prices or market movements (Kara, Acar Boyacioglu & Baykan, 2011). They can capture characteristics such as momentum and market trend that potentially could have predictive power. Therefore, we utilize the OHLCV data retrieved for the S&P 500 Index to calculate 6 different technical indicators which we include as features in this study. All indicators are calculated using the Pandas TA library. Furthermore, the selection of technical indicators and their window lengths is based on previous literature (Patel et al., 2015).

The simple moving average (SMA) is a technical indicator that smooths out price fluctuations and captures underlying trends. In this study, we use the 10-day SMA which is a rolling average of the closing prices over the last 10 days. The formula for the 10-day SMA can be written as:

$$SMA_{10} = \frac{C_t + C_{t-1} \cdots + C_{t-9}}{10} \quad (1)$$

Where C_t is the closing price of an asset at time *t.* By including the SMA we introduce a lagging indicator that has the potential to capture the short-term trend and shifts in momentum.

The weighted moving average (WMA) is an indicator that can be viewed as a compliment to the SMA. The WMA assigns weights to the price observations. It assigns higher weights to more recent prices, making it more responsive to recent price movements. We use the 10-day WMA which can be expressed by the following formula:

$$WMA_{10} = \frac{nC_t + (n-1)C_{t-1} + \cdots + C_{t-9}}{n + (n-1) + \cdots + 1} \quad (2)$$

Where C_t is the closing price at time *t* and *n* represents the window length which in this case is 10 days.

The relative strength index (RSI) is a widely recognized momentum indicator that quantifies the magnitude of recent price changes. It helps in recognizing the points where a security is overbought or oversold. The formula for RSI can be written as:

$$RSI = 100 - \frac{100}{1 + (\sum_{i=0}^{n-1} Up_{t-1}/n)/(\sum_{i=0}^{n-1} Dw_{t-1}/n)} \quad (3)$$

Where Up_t represents upward price change and Dw_t stands for downward price change at time *t*. Furthermore, *n* represents the number of periods which typically is 14 days. This is also the window length utilized in this thesis.

The William's %R is a versatile indicator that similar to the RSI also measures overbought and oversold points for a security over a specified period. In this study we use the default period of 14 days. The indicator is calculated as the ratio of the difference between the highest high and the current closing price to difference between the highest high and the lowest low over the selected period. The formula for Williams %R can be expressed as:

$$William's\ \%R = \frac{H_n - C_t}{H_n - L_n} \times 100 \quad (4)$$

Where H_n represents the highest high at time *t* and L_n represents the lowest low during the same time period. Williams %R yields values between -100 and 0.
Momentum is a technical indicator that measures the rate of change in a security's price. It is calculated as the difference between the closing price at time *t* and the closing price at *t – k* where *k* represents the number of time periods. In this study we use a window length or time period of 10 days which also is the default value in the Pandas TA library. Momentum provides insights into the strength and direction of price movements. The formula for momentum can be expressed as:

$$Momentum = C_t - C_{t-9} \tag{5}$$

Where C_t represents the closing price of the asset or security at the current time period *t* and C_{t-9} represents the closing price of the same asset 9 trading days earlier.

Stochastic %K is a momentum indicator that compares the current closing price of an asset relative to its price range over a selected period. The default value of 14 days is used in this thesis. Stochastic %K provides insights into momentum and potential reversal points which could help our models capture short-term market trends. The formula for the indicator can be written as:

$$Stochastic\ \%K = \frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \tag{6}$$

Where HH_t and LL_t represent the highest high and lowest low prices in the past *t* days respectively.

3.2.3 Target Variable

In this thesis, we frame the movement of the S&P 500 Index as a binary classification problem. To derive the binary classes, we calculate the daily logarithmic returns for the S&P 500 Index using the following formula:

$$Log\ Return_t = ln\left(\frac{P_t}{P_{t-1}}\right) \tag{7}$$

Where P_t is the adjusted closing price of the S&P 500 at time *t*, and P_{t-1} is the adjusted closing price of the previous day.

With the log returns calculated, we categorize each trading day or observation as either "Up" or "Down". An observation is categorized as "Up" if the log return of the S&P 500 Index is positive. Conversely, if the log return is negative or equal to zero, we label that observation as "Down". The final step in constructing the target variable involves encoding our two classes into a numerical format. We assign the label 1 to the "Up" class, and the label 0 to the "Down" class.

3.2.4 Data Transformation

An important aspect of working with time series data is to test for stationarity. Non-stationary data often show trends, cycles, and other patterns that aren't stable over time. This can lead to the identification of spurious relationships between variables (Brooks, 2008). Furthermore, models built on such data might not generalize well as the patterns identified are not consistent.

In order to check the features for stationarity, we utilized the augmented-Dickey Fuller test with a significance level of 0.05. All variables apart from the technical indicators and the logarithmic return of the S&P 500 Index exhibited non-stationarity. A common approach used to detrend non-stationary variables is to take their first difference and, in some cases, their second difference (Wooldridge, 2013). In this study, we transform all variables that exhibited non-stationarity by taking the logarithmic difference, using the same formula as in equation (7). Following the transformations, we completed another round of augmented-Dickey Fuller tests and observed that all features now appeared to be stationary.

We standardize all features to ensure that they are on the same scale. This is important as differences in scale can lead to biases during the modelling process where variables with larger values are favoured. We also lagged all features by one period or day to align them with the target variable. This allows us to utilize information available today, to predict the directional movement of the S&P 500 Index the following day. After all transformations, we have a final dataset that ranges from 07.01.2008 to 31.12.2022, resulting in 3777 observations, and 18 features.

4. Methodology

4.1 Logistic Regression

Logistic regression is a common machine learning method which often is applied to solve binary classification tasks (Sperandei, 2014). As a linear classifier, it assumes that the decision boundaries separating the data points into distinct classes are linear. The model predicts the probability of an observation belonging to one of two classes by transforming a linear combination of input features using the sigmoid function. The output variable of the logistic regression model is restricted to an interval between 0 and 1, allowing it to be interpreted as a class probability. The sigmoid function is defined as:

$$g(z) = \frac{1}{(1 + e^{-z})} \tag{8}$$

In the formula for logistic regression, the y represents the binary target variable (1 or 0), while x is a representation of the input features. The goal of logistic regression is to find the best fitting parameters θ that minimize the difference between the predicted probabilities and the actual class labels in the training data. The formula for the logistic regression model can be written as:

$$p(y = 1|x; \theta) = h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{\theta^T X}} \tag{9}$$

In order to map the output of a logistic regression model to a class label, a decision threshold must be defined (Zweig & Campbell, 1993). In this thesis, the default threshold of 0.5 is used as it maximizes the accuracy for balanced datasets. If the output of the logistic regression model is greater than 0.5, the model assigns the observation to class 1 and predicts that the S&P 500 will move in a positive direction the next day. If the probability is less than 0.5, the model will assign the observation to class 0, which predicts that the S&P 500 will move in a negative direction.

4.2 Random Forest

Random forest is an ensemble learning method that is widely used for both classification and regression tasks (Breiman, 2001). It belongs to the family of decision-tree based algorithms and operates by constructing and training multiple decision trees in parallel. Decision trees are hierarchical structures that recursively partition the feature space based on the value of the input features (Quinlan, 1986). Each internal node in a tree represents a decision based on a specific feature, and each leaf node represents the predicted output or class label. Decisions trees are easy to interpret and can capture complex non-linear relationships with low bias. However, they tend to have high variance as they are prone to overfitting when generalizing to new data.

The random forest algorithm combines several decision trees in an ensemble. To construct the ensemble, a user-defined number of trees are built in parallel. During the construction of each tree, a random subset of the training data is drawn with replacement. This technique, known as bagging, helps reduce variance and improves the model's stability by introducing diversity in the training data.

Additionally, at each split point of a tree, only a random subset of the total features is considered for determining the best split. By doing so, the random forest algorithm decorrelates the individual trees in the ensemble which leads to a further reduction in variance (Belgiu & Drăgu, 2016).

Once all decision trees in the ensemble are constructed, the final prediction of the random forest algorithm is obtained by aggregating the predictions of the individual trees. For classification tasks, a majority vote system is utilized. If the majority of the trees in the ensemble predicts that the S&P 500 will move in a positive direction the next day, that will be the final prediction of the model for that specific data point. The prediction of the random forest model can be written as:

$$\hat{y} = Majority\ vote\left(\sum_{b=1}^{B}\frac{\hat{C}_b(x)}{b}\right) \tag{10}$$

The $\hat{y}$ is the final aggregated predicted class label generated by the model, while $\hat{C}_b(x)$ represents the prediction of an individual decision tree. Furthermore, the *B* represents the number of actual decision trees in the ensemble.

4.2.1 Hyperparameter Tuning

Hyperparameter tuning plays a critical role in optimizing the performance of many machine learning models (Hoque & Aljamaan, 2021). In the case of random forest, selecting appropriate hyperparameters can significantly impact the model's ability to generalize and make accurate predictions.

The number of estimators ("n_estimators") is an important hyperparameter. It determines the number of decision trees to construct and include in the random forest model. A larger value for the number of estimators can lead to improved model performance. However, it also increases the computational cost and training time. Conversely, a small value could result in underfitting, where the model fails to capture the pattern in the data.

The maximum depth of the decision trees ("max_depth") is another important hyperparameter. It controls the depth to which each decision tree in the ensemble is allowed to grow. A deeper tree can capture more complex patterns in the data but may also lead to overfitting, reducing the model's ability to generalize to new data. A decision tree that is too shallow might however fail to capture complex relationships or patterns.

Another crucial hyperparameter is the maximum features ("max_features"). It is the parameter that determines how many features to consider when searching for the optimal split at each node. A lower value will reduce the randomness in the random forest model and could therefore prevent overfitting. However, a lower value comes with a trade-off as it potentially could lead to overfitting if it's adjusted too low. Setting "max_features" to "auto" allows the algorithm to consider the square root of the total features or predictors, while "log2" considers the log base 2 of the total features.

We designed a search space with candidate values for a selection of the hyperparameters in the random forest model. The grid search algorithm in Python was then used to find the optimal combination of hyperparameter values. The objective was set to maximize model accuracy. In table 2, we summarize the search space of evaluated candidates and the optimal hyperparameter values that were selected.

Table 2. Random forest hyperparameters.

Hyperparameter	Search space	Default value	Selected value
n_estimators	{200, 300, 400}	100	300
max_depth	{4, 5, 6, 7}	none	4
max_features	{sqrt, log2}	sqrt	sqrt

There are several other hyperparameters that could be tuned in order to potentially improve the performance of the random forest model. However, tuning hyperparameters can be very computationally expensive and time consuming. Furthermore, a larger search space could potentially find more optimal values for the selected hyperparameters. This would however lead to an additional increase in computational cost that could be considerably high. In this thesis, we only consider a small number of hyperparameters and a limited search space of candidate values due to limitations in computational power and time.

4.3 Cross-Validation

In order to utilize machine learning algorithms, training and test data are required. A common approach is to partition the data where 80% is used to train the algorithm and the remaining 20% is used to test the model's performance on previously unseen data (Lindholm et al., 2022). If the data is large enough, this approach might be sufficient. However, if data is limited or scarce, this method might fail as machine learning algorithms require a lot of training data in order to ensure optimal performance (Hastie et al., 2021).

Cross-validation is a widely used technique that addresses this issue. In cross-validation, the training data is randomly partitioned into "k-folds" of equal size. The model is then trained on k-1 folds and tested or validated on the fold that was left out. This is an iterative process that is repeated until all folds have served as both training and testing data.

As the data in this research is time series data, regular cross-validation where the data is randomly partitioned and shuffled is inappropriate. In this thesis, we therefore utilize rolling window cross-validation, where the temporal structure of the data is maintained (Bergmeir & Benítez, 2012).

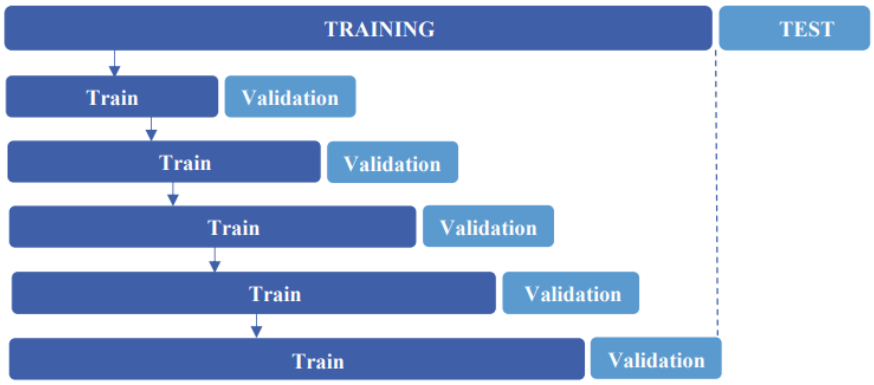

Figure 1. Illustration of a 5-fold rolling window cross-validation.

As shown in Figure 1, we use a 5-fold rolling window cross-validation technique. In this method, the dataset is split into training and testing sets by using a sliding window over time. The training set consists of data up to a certain point in time and the testing data contains data in the future. The window is rolled forward one step at a time, and the model is retrained and validated at each step. The machine learning models in this thesis are trained on the period 07.01.2008 to 05.07.2020 and then tested between 06.07.2020 and 31.12.2022.

4.4 Feature Importance

Feature importance is a useful method in the field of machine learning that provides insight into which features or variables that have the most impact on a model's predictions. Analyzing the features and their contribution to the model's performance can help identify irrelevant or redundant features that potentially can be removed in order to simply the model and reduce the computational cost. Decision-tree based models, such as the random forest algorithm, have built-in methods to calculate feature importance. Random forest assigns an importance score to each feature based on its contribution to the reduction in impurity across all trees in the ensemble. In order to allow us to compare the relative significance of each feature, the importance scores are normalized to sum up to 1. A higher feature score indicates that a variable has a greater relative impact on the model's predictions. There are several other ways to assess features importance. However, in this thesis we focus on the built-in method within the random forest algorithm due to its simplicity.

4.5 Evaluation Metrics

In this section we will describe the key evaluation metrics used to assess the performance of our machine learning models in regards to predicting the movement of the S&P 500 Index. Evaluation metrics play an important role in understanding how well a model generalizes to new data and the quality of its predictions. There are several different metrics that can be used to gauge the performance of a model. However, in this thesis we focus on accuracy and precision as we're dealing with a classification task.

4.5.1 Accuracy

Accuracy is a performance metric that quantifies the effectiveness of a model by calculating the proportion of correct predictions. It is computed by dividing the number of correct predictions by the total number of predictions. The formula for accuracy can be written as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

In this formula, *TP* represents the number of true positive predictions, *TN* signifies the number of true negative predictions, *FP* denotes the number of false positive predictions, and *FN* indicates the number of false negative predictions.

4.5.2 Precision

Precision is a metric that measures a model's ability to correctly predict positive instances. In the context of this thesis, precision provides information on how often a model is correct when it predicts that the S&P 500 will move in a positive direction the following day. The formula for precision can be written as:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

5. Trading Strategies

In order to further test the validity of our prediction models, we develop two distinct trading strategies and compare their performances. The first strategy is a traditional portfolio that allocates 60% to equities and 40% to fixed income (Rekenthaler, 2022). This buy and hold strategy is well-known in the financial industry for its ability to achieve a balance between

risk and return. Therefore, it will serve as a benchmark in this thesis. The second portfolio will follow a dynamic allocation strategy that leverages the predictions of our best-performing machine learning model. The predictions of the model will act as a signal for portfolio allocation decisions. If the model predicts that the S&P 500 Index will move in a positive direction the following day, we allocate 100% to equities. Conversely, if the model predicts a negative movement, we instead allocate 100% to fixed income.

To compare the performances of the two portfolios, we will use the Sharpe ratio. The Sharpe ratio is a measure that evaluates a portfolio's return relative to its volatility or risk (Sharpe, 1966). A higher Sharpe ratio indicates a better risk-adjusted performance, as the portfolio generates more excess return for each unit of risk. Conversely, a lower Sharpe ratio suggests that the portfolio is not adequately compensating for the level of risk it entails. The formula for the Sharpe ratio can be written as:

$$S_p = \frac{r_p - r_f}{\sigma_p} \tag{13}$$

Where r_p is the portfolio return, r_f the risk-free rate, and σ_p the volatility or standard deviation of the portfolio. The Sharpe ratio will be reported on a yearly basis as this is common practice in the financial industry. The investment horizon for the two portfolios is equivalent to the period in the testing dataset as it is for this period the predictions are made. Thus, the investment horizon is between 06.07.2020 and 31.12.2022.

5.1 Trading Strategy Data

Because it is not possible to invest directly in the S&P 500 Index, we use the SPDR S&P 500 ETF Trust (SPY) to represent equities. SPY is an exchange-traded fund that is structured to closely replicate the performance of the S&P 500 Index. The Vanguard Total Bond Market ETF (BND) is selected to represent fixed income. BND is an exchange-traded fund that tracks a diverse range of fixed income securities. Furthermore, the three-month U.S. Treasury bill serves as a proxy for the risk-free rate in our Sharpe ratio calculations, a common approach in the financial industry. All data pertaining to the investment strategies in this thesis was retrieved from Yahoo Finance.

6. Results and Discussion

In this part of the thesis, the empirical results of the prediction models and the trading strategies are presented and discussed. The results of the prediction models are analyzed using the performance metrics mentioned earlier in this thesis, accuracy, and precision. The confusion matrices of the models are also presented. Furthermore, the results of the investment strategies are analyzed and compared using the Sharpe ratio. This section is then concluded with a discussion regarding the limitations pertaining to this study and reflections regarding potential future research.

6.1 Prediction Results

The empirical results show that the random forest model outperforms the logistic regression model. The RF model achieved an accuracy of approximately 54.5%, indicating that it correctly predicts the movement of the S&P 500 Index 54.5% of the time. The precision score of around 53.9% signifies that when the RF model predicts that the S&P 500 Index will move in a positive direction the following day, it is correct 53.9% of the time. The relatively balanced relationship between accuracy and precision suggests that the model is achieving a good compromise between correctly predicting positive movements and minimizing false positives.

Table 2. Performance of machine learning models

Model	**Accuracy**	**Precision**
Random Forest	54.5%	53.5%
Logistic Regression	50.2%	51.6%

In contrast, the logistic regression model has an accuracy of approximately 50.2%, suggesting that its predictions are slightly better than random chance. The model has a precision score of around 51.6% which implies that when it predicts a positive movement the following day, it is accurate 51.6% of the time.

To further asses the performance of our machine learning models, we compare the prediction results to previous studies. Di (2014) predicted the next day price trend of the S&P 500 Index using a SVM model. The accuracy achieved in the study was 56%. Likewise, Liu, Wang, Xiao, and Liang (2016) predicted the daily movement of the S&P 500 Index and achieved an accuracy of 62.51%. Our RF model has a slightly lower accuracy than Liu et al. (2016) but

our results appear to be empirically reasonable. The comparison with previous studies highlights the complexity and difficulty of predicting the movement of the S&P 500 Index.

We continue our evaluation of the prediction models by analyzing their confusion matrices. The matrices, displayed in figure 3, tabulates the performance of the models in terms of the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) made for each class.

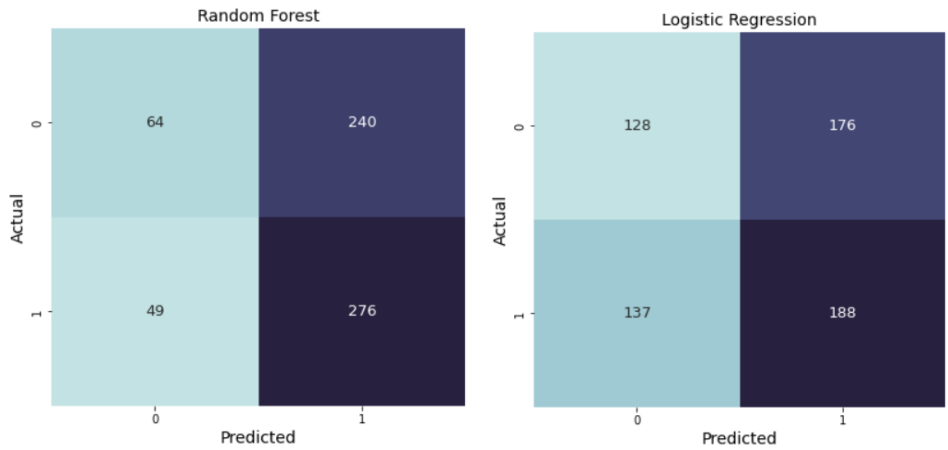

Figure 2. Confusion matrix of the machine learning models.

By analyzing the confusion matrix of each machine learning model, we can conclude that both models exhibit a tendency to predict upward movements in the S&P 500 Index more frequently than they do downward movements. The RF model misclassifies approximately 15.2% of actual up movements as down movements and around 78.3% of actual down movements as up movements. In contrast, the LR model misclassifies around 42.1% of actual upward movements as down movements and approximately 57.9% of actual downward movements as upward movements. Both models tend to perform better when predicting upward movements rather than downward movements. An imbalanced dataset can many times be the reason for this. However, the data used in this research is rather balanced. Approximately 54% of the observations in the target variable belong to class 1 (upward movements) while 45% belong to class 0 (downward movements). Thus, we disregard this as a possible explanation.

Although the superior performance of the RF model is subtle, it is meaningful. The modest differences in accuracy and precision can significantly impact trading decisions and strategies, highlighting the importance of even slight improvements in model performance.

As the RF model was the best performing model, it will be our focus in the following section, with an emphasize on feature importance. According to the feature importance analysis, the U.S. dollar index (DXY) was the most important variable for predicting the movement of the S&P 500 Index. A possible explanation for this finding lies in the fact that changes in the relative value of the U.S. dollar can influence international trade dynamics, potentially indirectly affecting investor sentiment and market movements.

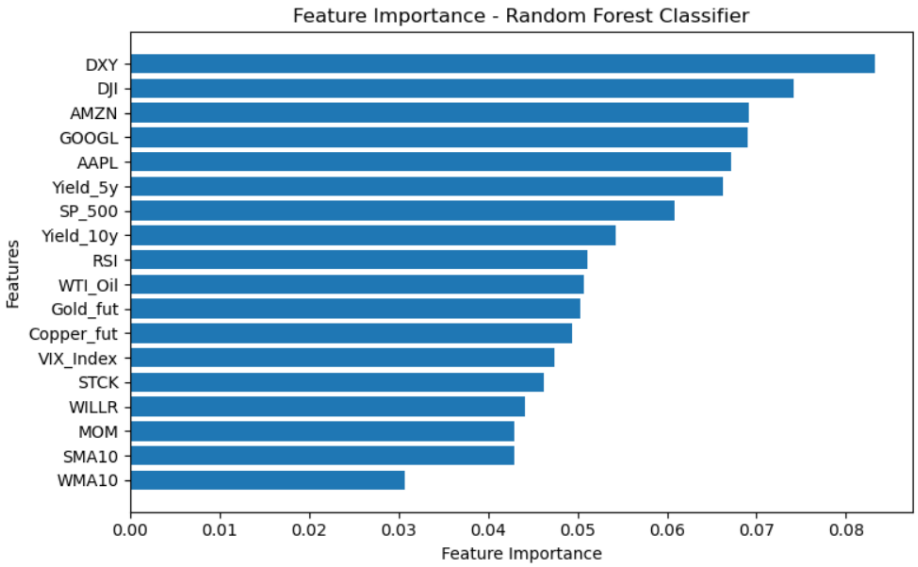

Figure 3. Feature importance analysis.

Following the DXY, the Down Jones Industrial Average (DJIA), Amazon, Google, and Apple have notable importance when it comes to contributing to the RF model's predictive accuracy. The DJIA is a market index that represents many different industries which potentially allows it to capture the overall market sentiment and trend. Amazon, Google, and Apple are all large technology companies that represent a considerable proportion of the S&P 500 Index. Thus, we expected them to be amongst the variables with the highest impact on model performance. The lagged return of the S&P 500 Index is only the 7th most important feature according to the analysis. This is quite surprising as we would expect it to be one of the features with the highest impact, given that it is directly derived from the S&P 500 Index itself. Equally surprising is the fact that all technical indicators apart from RSI are amongst the features with the least importance when it comes to the model's predictive performance since they also are directly derived from OHLCV data on the S&P 500. Furthermore, the VIX has a relatively low impact on the model's predictive performance. Again, this is surprising as we expected the market's overall expectation of volatility to be one of the features with the

highest impact when it comes to predicting the movement of the S&P 500 Index. Parsimonious models are often favored when it comes to machine learning. We therefore used the outcome of the feature importance analysis to build a simpler model by only using the 10 most important variables. However, the accuracy fell from 54.5% to approximately 51% as a result of the experiment. Thus, we reinstated all features as the objective was to predict the movement of the S&P 500 Index with the highest accuracy possible.

6.2 Trading Strategy Results

The results of the two strategies show that the portfolio that followed the predictions of the random forest model achieved an annualized return of 41.7% with a standard deviation of 16.7%. In contrast, the benchmark portfolio that followed the 60/40 strategy generated a return of 33.8% with a portfolio volatility of 12.1%. According to these results, the machine learning-based strategy is capable of generating positive returns for the period tested. However, although the dynamic allocation strategy provided a higher return compared to the benchmark portfolio, we are primarily interested in the risk-adjusted performance captured by the Sharpe ratio.

Table 3. Performance of investment strategies.

Portfolio	**Annualized Return**	**Annualized Volatility**	**Sharpe Ratio**
60/40 Strategy	33.8%	12.1%	2.440
RF Strategy	41.7%	16.7%	2.231

A higher Sharpe Ratio indicates a better risk-adjusted performance, highlighting a strategy's ability to generate returns relative to the risk undertaken. By analyzing the Sharpe ratios, we observe that the traditional 60/40 strategy outperforms our random forest strategy. The benchmark portfolio generated a Sharpe ratio of 2.440 while the random forest-based portfolio accumulates a Sharpe ratio of 2.231. This indicates that the benchmark portfolio achieves a more optimal balance between risk and return. An investor holding the benchmark portfolio would receive a higher compensation for each unit of risk undertaken. However, the Sharpe ratios of the two investment strategies are not that significantly different. This suggests that if our random forest model achieved a slightly higher accuracy, the dynamic allocation strategy potentially would be able to beat the traditional 60/40 strategy.

6.3 Limitations and Future Research

Predicting the movement of the S&P 500 Index using machine learning is a challenging and complex task and as such, it is important to acknowledge the limitations pertaining to this study. The quality and amount of data utilized is a determining factor for the performance of any machine learning model. Although there is a rationale behind the features selected in this study, a limitation is that we only use financial data. Due to the inherent complexity of the financial markets, it is possible that the data used in this research isn't diverse enough to fully capture the underlying factors that drive market movements. Future research could augment the dataset used by including macroeconomic indicators, news data, and sentiment analysis from social media platforms to capture a more holistic view of market dynamics.

In this thesis, we frame the movement of the S&P 500 Index as a binary classification problem. This might overlook potential nuances within the financial markets. By adding a third category to the binary classes, future research could potentially capture other dynamics such as sideway movements which might generate additional insights.

Another limitation lies in the assumption that the hyperparameter tuning process of the random forest model yielded the optimal parameter values. However, due to computational constraints, we only considered a small subset of all the possible hyperparameters that potentially could be tuned. Furthermore, we employed a rather narrow grid space with candidate values for the hyperparameters. As a result, the selected values for the hyperparameters might not represent the optimal combination for maximizing the performance of the random forest model. Additionally, we focus our study around only two algorithms, the random forest, and the logistic regression model. It is possible that other machine learning algorithms would yield better performances in terms of capturing market dynamics and predictive accuracy. Future research could focus on optimizing a greater number of hyperparameters over a larger grid space of candidate values, and employing alternative machine learning algorithms such as SVM, ANN and XGB. This could perhaps further improve the predictive accuracy.

In terms of the investment strategies deployed in this study, we only examined two approaches. Although the results were insightful, future research could potentially delve deeper by implementing several investment strategies focusing on other financial securities such as E-mini S&P 500 Index Futures. Furthermore, future research could consider the effect of transactions costs as these are disregarded in this study in favour of simplicity.

7. Conclusion

In this thesis, the goal was to predict the movement of the S&P 500 Index with the highest accuracy possible to investigate the feasibility of using machine learning in achieving this task. Out of the two models we deployed in this study, the random forest turned out to be the best performing model with an accuracy of approximately 54.5%. Although the model only achieves a slightly higher accuracy than what would be considered random guessing (50%), the outcome should not be considered as unsatisfactory as even small differences in accuracy can have huge implications for an investment strategy. Furthermore, the trading strategy developed based on the predictions on the random forest model fails to outperform the benchmark portfolio. This highlights the difficulties of predicting the stock market.

8. References

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

Basak, S., Kar, S., Saha, S., Khaidem, L. & Dey, S.R. (2019) Predicting the direction of stock market prices using tree-based classifiers. *North American Journal of Economics and Finance*. 47. doi:10.1016/j.najef.2018.06.013.

Belgiu, M. & Drăgu, L. (2016) Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*.114. doi:10.1016/j.isprsjprs.2016.01.011.

Bergmeir, C. & Benítez, J.M. (2012) On the use of cross-validation for time series predictor evaluation. *Information Sciences*. 191. doi:10.1016/j.ins.2011.12.028.

Breiman, L. (2001) Random forests. *Machine Learning*. 45 (1). doi:10.1023/A:1010933404324.

Brooks, C. (2008) *Introductory Econometrics for Finance*. vol. 2,. Cambridge University Press. . doi:10.1017/cbo9780511841644.

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

Di, X. (2014) Stock Trend Prediction with Technical Indicators using SVM. *Stanford University*.

Efendi, R., Arbaiy, N. & Deris, M.M. (2018) A new procedure in stock market forecasting based on fuzzy random auto-regression time series model. *Information Sciences*. 441. doi:10.1016/j.ins.2018.02.016.

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

Fischer, T. & Krauss, C. (2018) Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. 270 (2). doi:10.1016/j.ejor.2017.11.054.

Hoque, K.E. & Aljamaan, H. (2021) Impact of hyperparameter tuning on machine learning models in stock price forecasting. *IEEE Access*. 9. doi:10.1109/ACCESS.2021.3134138.

Huang, C.-S. & Liu, Y.-S. (2019) Machine Learning on Stock Price Movement Forecast: The Sample of the Taiwan Stock Exchange. *International Journal of Economics and Financial Issues* |. 9 (2).

Huang, W., Nakamori, Y. & Wang, S.Y. (2005) Forecasting stock market movement direction with support vector machine. *Computers and Operations Research*. 32 (10). doi:10.1016/j.cor.2004.03.016.

Huenermund, P., Kaminski, J.C. & Schmitt, C. (2021) Causal Machine Learning and Business Decision Making. *Academy of Management Proceedings*. 2021 (1). doi:10.5465/ambpp.2021.12517abstract.

Kara, Y., Acar Boyacioglu, M. & Baykan, Ö.K. (2011) Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications*. 38 (5). doi:10.1016/j.eswa.2010.10.027.

Kumar, M. & M., T. (2011) Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest. *SSRN Electronic Journal*. doi:10.2139/ssrn.876544.

Liu, C., Wang, J., Xiao, D. & Liang, Q. (2016) Forecasting S&P 500 Stock Index Using Statistical Learning Models. *Open Journal of Statistics*. 06 (06). doi:10.4236/ojs.2016.66086.

Lo, A.W. (2004) The adaptive markets hypothesis. *Journal of Portfolio Management*.30 (SUPPL.). doi:10.3905/jpm.2004.442611.

Lo, A.W. & MacKinlay, A.C. (1988) Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test. *The Review of Financial Studies*. 1 (1), 41–66. doi:10.1093/rfs/1.1.41.

Niaki, S.T.A. & Hoseinzade, S. (2013) Forecasting S&P 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*. 9 (1). doi:10.1186/2251-712X-9-1.

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

Quinlan, J.R. (1986) Induction of Decision Trees. *Machine Learning*. 1 (1). doi:10.1023/A:1022643204877.

Rekenthaler & J (2022) *Why the 60/40 Portfolio Continues to Outlast Its Critics,*. https://www.morningstar.com/portfolios/why-6040-portfolio-continues-outlast-its-critics.

Shen, S., Jiang, H. & Zhang, T. (2012) Stock market forecasting using machine learning algorithms. *Department of Electrical Engineering, Stanford University*.

Sperandei, S. (2014) Understanding logistic regression analysis. *Biochemia Medica*. 24 (1). doi:10.11613/BM.2014.003.

Wang, Y. & Choi, I. (2013) Market Index and Stock Price Direction Prediction using Machine Learning Techniques: An empirical study on the KOSPI and HSI. *arXiv preprint arXiv:1309.7119*. 00.

Wooldridge, J.M. (2013) Introductory Econometrics A Modern Approach, Fifth Edition. *Cengage Learning, South-Western*.

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

Zweig, M.H. & Campbell, G. (1993) Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine. *Clinical Chemistry*.39 (4). doi:10.1093/clinchem/39.4.561.