DEGREE PROJECT IN MATHEMATICS,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Predicting Stock Price Direction for Asian Small Caps with Machine Learning Methods

TINA ABAZARI

SHERWIN BAGHCHESARA

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Abstract

Portfolio managers have a great interest in detecting high-performing stocks early on. Detecting outperforming stocks has for long been of interest from a research as well as financial point of view. Quantitative methods to predict stock movements have been widely studied in diverse contexts, where some present promising results. The quantitative algorithms for such prediction models can be, to name a few, support vector machines, tree-based methods, and regression models, where each one can carry different predictive power. Most previous research focuses on indices such as S&P 500 or large-cap stocks, while small- and micro-cap stocks have been examined to a lesser extent. These types of stocks also commonly share the characteristic of high volatility, with prospects that can be difficult to assess. This study examines to which extent widely studied quantitative methods such as random forest, support vector machine, and logistic regression can produce accurate predictions of stock price directions on a quarterly and yearly basis. The problem is modeled as a binary classification task, where the aim is to predict whether a stock achieves a return above or below a benchmark index. The focus lies on Asian small- and micro-cap stocks. The study concludes that the random forest method for a binary yearly prediction produces the highest accuracy of 69.64%, where all three models produced higher accuracy than a binary quarterly prediction. Although the statistical power of the models can be ruled adequate, more extensive studies are desirable to examine whether other models or variables can increase the prediction accuracy for small- and micro-cap stocks.

Keywords: Machine Learning, Classification, Classification Trees, Random Forest, Support Vector Machine, Logistic Regression, Stocks, Stock Market, Fund Management, Asset Management, Investments, Asia, Small Cap, Micro Cap.

Sammanfattning

Portföljförvaltare har ett stort intresse av att upptäcka högpresterande aktier tidigt. Detektering av högavkastande aktier har länge varit av stort intresse dels i forskningssyfte men också ur ett finansiellt perspektiv. Kvantitativa metoder för att förutsäga riktning av aktiepriset har studerats i stor utsträckning där vissa presenterar lovande resultat. De kvantitativa algoritmerna för sådana prediktionsmodeller kan vara, för att nämna ett fåtal, support vector machines, trädbaserade metoder och regressionsmodeller, där var och en kan bära olika prediktiv kraft. Majoriteten av tidigare studier fokuserar på index såsom S&P 500 eller storbolagsaktier, medan små- och mikrobolagsaktier har undersökts i mindre utsträckning. Dessa sistnämnda typer av aktier innehar ofta en hög volatilitet med framtidsutsikter som kan vara svåra att bedöma. Denna studie undersöker i vilken utsträckning väletablerade kvantitativa modeller såsom random forest, support vector machine och logistisk regression, kan ge korrekta förutsägelser av små- och mikrobolags aktiekursriktningar på kvartals- och årsbasis. I avhandlingen modelleras detta som ett binärt klassificeringsproblem, där avkastningen för varje aktie antingen är över eller under jämförelseindex. Fokuset ligger på asiatiska små- och mikrobolag. Studien drar slutsatsen att random forest för en binär årlig prediktion ger den högsta noggrannheten på 69,64 %, där samtliga tre modeller ger högre noggrannhet än en binär kvartalsprediktion. Även om modellerna bedöms vara statistiskt säkerställda, är det önskvärt med fler omfattande studier för att undersöka om andra modeller eller variabler kan öka noggrannheten i prediktionen för små- och mikrobolags aktiekursriktning.

Nyckelord: Maskininlärning, Klassificering, Klassificeringsträd, Random Forest, Support Vector Machine, Logistisk Regression, Aktier, Aktiemarknad, Fondförvaltning, Investeringar, Asien, Småbolag, Mikrobolag.

Svensk titel: Prediktering av aktiekursriktningen för asiatiska småbolagsaktier med maskininlärning

Preface

We would like to thank our supervisor Boualem Djehiche at KTH Royal Institute of Technology for valuable guidance and inputs throughout the course of this thesis. We would also like to express our gratitude to Mikael Sens at Handelsbanken Fonder AB for providing us with necessary means to carry out the research.

Stockholm, Sweden

Contents

1 Introduction 1

1.1 Background 1

1.2 Purpose and Problematization 2

1.3 Research Question 2

1.4 Scope and Delimitations 3

1.5 Limitations 3

1.6 Literature Review 3

1.6.1 Machine Learning in Stock Price Prediction 3

1.6.2 Financial Ratios in Stock Price Prediction 6

2 Financial Background 8

2.1 Stocks and the Stock Market 8

2.2 Stock Index 8

2.3 Fundamental Analysis and Technical Analysis 9

2.4 Other terminology 9

3 Mathematical Background 11

3.1 Machine Learning 11

3.2 Supervised Learning 12

3.2.1 Parametric vs Non-parametric Models 13

3.3 Support Vector Machines 13

3.3.1 Separating Hyperplanes 14

3.4 Decision Trees and Random Forest 15

3.4.1 Classification Trees 15

3.4.2 Bagging 17

3.4.3 Random Forest 18

3.4.4 Variable Importance in Random Forest 19

3.5 Logistic Regression 19

3.5.1 Fitting the Logistic Regression Model 20

3.6 Model Evaluation Metrics 21

3.6.1 Confusion Matrix 22

3.6.2 Kappa Statistic 23

3.6.3 Receiver Operator Characteristic Curve 24

3.6.4 Area Under Receiver Operator Characteristic Curve 25

3.7 Confidence Interval 25

4 Methodology 26

4.1 Research Outline 26

4.2 Data Collection 27

4.3 Selection of Variables 29

4.4 Preprocessing Data 30

4.4.1 Data Transformation 30

4.4.2 Feature Scaling 32

4.4.3 Data Cleaning 33

4.4.4 Data Cleaning for Quarterly Prediction 33
4.4.5 Data Cleaning for Yearly Prediction 34
4.5 Model Development and Evaluation 34
4.5.1 Model Selection 34
4.5.2 Visualization of Data Using Principal Components 35
4.5.3 Model Fitting and Computational Program 35
4.5.4 Cross Validation and Hyperparametertuning 35
4.5.5 Statistical Testing 36

5 Results **36**
5.1 Quarterly Prediction 36
5.1.1 Principal Components Analysis 36
5.1.2 Support Vector Machine 38
5.1.3 Random Forest 39
5.1.4 Logistic Regression 42
5.1.5 ROC Curve and AUC 44
5.2 Statistical Testing of Models 45
5.2.1 Standard Deviation and Confidence Interval of Models . . 45
5.2.2 Kappa Statistic 46
5.2.3 Hypothesis Test 46
5.3 Yearly Prediction 47
5.3.1 Principal Components Analysis 47
5.3.2 Support Vector Machine 48
5.3.3 Random Forest 49
5.3.4 Logistic Regression 52
5.3.5 ROC Curve and AUC 54
5.4 Statistical Testing of Models 55
5.4.1 Standard Deviation and Confidence Interval of Models . . 55
5.4.2 Kappa Statistic 55
5.4.3 Hypothesis Test 55

6 Discussion **56**

7 Conclusion **57**

8 Future Research **58**

1 Introduction

The Introduction section includes Background, Project Purpose and Problematization, Research Questions, Scope and Delimitations and Literature Review.

1.1 Background

The predictability of the stock market has been discussed for a long time. For many years, the *Efficient Market Hypothesis* (EMH) was the most established theory within finance. The EMH implies that abnormal returns cannot be gained by looking at historical price movements [1]. However, various studies have rejected the EMH and suggested that security return may actually be predictable [2][3]. Hence, different methods have been studied to predict future prices and detect high-performing stocks to one's advantage. However, stock price forecasting is still viewed as a challenging task given the non-stationary, volatile and noisy characteristics of the stock market. Unexpected events and news such as catastrophes, political tensions, or business conduct may all impact the predictability of financial time series. Also, over time the relationship between a financial time series and another linked variable could change [4][5]. Altogether this makes stock price prediction one of the most difficult tasks within time series forecasting.

Forecasting stock prices or stock price directions accurately has major importance for both financial market participants and researchers. Namely, stock market prediction provides an opportunity to gain high profits and minimize risks associated with investments. From the perspective of financial participants, finding accurate forecasting methods may improve investment performance significantly, and is therefore critical for developing a competitive edge. Consequently, there has been an upward trend in financial market participants investing in technology that can be integrated into investment processes [6]. This has particularly shed light on big data and statistical learning models as a complement to fundamental and technical analysis, and an increasing number of asset managers, such as Handelsbanken Fonder AB, have sought to incorporate these techniques in their investment processes [7].

Handelsbanken Fonder AB is a well renowned fund manager in Sweden and are currently investigating the business case for starting an Asian micro- and small-cap equity fund as a new offering to their private and institutional investors. To ensure that such a product is attractive, the portfolio managers want to support their investment decisions with a quantitative model which could help identify the Asian small- and micro-cap stocks which will potentially outperform their index.

1.2 Purpose and Problematization

Handelsbanken Fonder AB's value offerings comprise actively managed equity-fixed income- and multi-asset mutual funds. They are currently offering funds ranging from global equity funds to mutual funds with specific sector-, regional- or company-size focus. In order to meet the demand of various investors and further increase their competitive advantage, Handelsbanken seeks to complement its current offerings with new financial products. In particular, they are considering a new Asian equity fund, restricted to micro- and small-cap stocks.

The main responsibility of a portfolio manager is to invest clients' capital in securities with the aim to gain excess returns. In other words, they make decisions about acquiring and divesting assets based on analysis of companies' financial statements and their future outlook. Therefore, the daily tasks of Handelsbanken's equity portfolio managers involve screening, research, and evaluation of potential investments. Although the majority of their equity funds are limited to a certain region, the associated stock universe may still be broad and an extensive amount of work may be required to discover the stocks that may outperform their benchmark index. In order to make the investment process more efficient, Handelsbanken wishes for a quantitative model which can separate such stocks from other investments in the Asian small and micro cap universe. Additionally, to get a further insight into the market characteristics, Handelsbanken wants to investigate what factors may define outperforming Asian small- and micro-cap stocks, and whether these variables can be used as leading indicators to detect stocks that yield an excess return compared to a benchmark index early.

Most stock market predictive models as of now are not specifically aimed at micro- and small-caps, which in general have unique risk and volatility characteristics compared to companies with a large market capitalization, mainly related to default risk, according to Switzer [8]. Developing a predictive model for the micro- and small-cap stocks is thus both interesting and useful for investors. Further, most studies primarily focus on forecasting only one national index, rather than an entire region as Asia [9]. Thus, limiting the scope to the geographical area requested by Handelsbanken Fonder (Asia excluding Japan) could also contribute to further research topics as Asia is an emerging market, drawing attention from many equity investors. It is therefore deemed relevant for the field to investigate a predictive stock price direction model for micro- and small-cap stocks in Asia. In line with Handelsbanken Fonder's investment strategy, this thesis investigates quarterly and yearly predictions using quantitative and fundamental analysis.

1.3 Research Question

The purpose of this study is to develop a classification model that for the geographical region Asia early on can identify an outperforming stock when given financial metrics and macroeconomic variables, in the category micro- and small-cap. In this study, an outperforming stock is defined as one that yields an excess return compared to the benchmark index, described in the following section. In order to solve this task properly, the supporting questions (RQ) are:

- **RQ 1**: *Which model(s) can predict and identify outperforming Asian small- and micro-cap stocks on a quarterly- and one-year basis?*
- **RQ 2**: *Which financial metrics (company-specific and macroeconomic in variables) are identified by the models as having significant importance when detecting an outperforming stock for Asian small- and micro-cap dataset?*

1.4 Scope and Delimitations

The geographical region is exclusive to Asia excluding Japan due to the aforementioned request by Handelsbanken Fonder. In particular, the study focuses on countries in Asia in which Handelsbanken is interested to invest in, which are China, Hong Kong, India, Indonesia, Malaysia, Philippines, Singapore, South Korea, Taiwan, Thailand, and Vietnam. Lastly, the study is restricted to small- and micro-cap stocks. Since there is no strictly defined market capitalization range for small- and micro-cap stocks, the companies in the sample are chosen such that their market values range between the smallest and the largest benchmark index constituent's market capitalization. Since the market capitalization category of outperforming stocks may have had changed during the chosen time period in this study, the upper limit is set to somewhat larger than the one of the benchmark index. The resulting range is USD 20-9,500 million as of 2020-12-31.

1.5 Limitations

The data that this thesis aims to test is from the software that can be accessed from Handelsbanken Fonder's computers, which in this case is Bloomberg. Thus, the data is limited to what is offered by the data vendor. Nevertheless, Bloomberg is one of the largest financial data vendors, with fundamental data on approximately 85,000 companies. The amount of data collected is also limited by the monthly downloading limit of the Bloomberg license. Thus, the number of stocks and variables included are constrained to the data access.

1.6 Literature Review

1.6.1 Machine Learning in Stock Price Prediction

Asset price forecasting opposes the Efficient Market Hypothesis (EHM), one of the most well-known principles of finance. The theory suggests that all available information of an asset is incorporated in its market price, implying that one cannot gain abnormal returns by analyzing past asset prices and returns [10]. Since the release of the EMH, researchers have found EMH anomalies and have hence tried to find ways to early identify investments that may produce higher return than the market average. In recent years an increasing number of studies have tried to utilize statistical techniques and algorithms to model asset price movements and returns, as these can help to identify patterns in data [11][12].

Time-series forecasting and Machine Learning (ML) are two major methods that have been studied for predicting stock prices. Time series forecasting involves choosing a model for prediction based on past values [13]. Inter alia, Autoregressive integrated moving average (ARIMA) [14], Exponential Smoothing and Generalized autoregressive conditional heteroskedasticity model (GARCH) [15] are models that have been tested for stock price and stock volatility forecasting [16]. ARIMA has been one of the most widely used stock price forecasting models as it utilizes the underlying information in a variable's previous (lagged) values and past errors and it has especially shown to be robust and effective in short-term prediction of time series [17]. However, ARIMA has shown limitations when forecasting nonlinear data such as prices, since it is a linear model. To combat the problem, more recent studies have examined Machine Learning techniques such as decision trees (CART) [18], Support Vector Machines (SVM)[19] and Artificial Neural Networks (ANN) [20][9]. Hiransha, M et al evaluated various neural networks, including Long Short Term Memory (LTSM), Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN), for predicting stock prices of five selected large-cap stocks from NSE and New York Stock Exchange (NYSE). The 400 days prediction results were compared to an ARIMA forecast and it was shown that all neural networks outperformed ARIMA. It was suggested that univariate models such as the ARIMA failed to capture underlying market dynamics [21]. Qian, XY compared the precision of ARIMA to several ML algorithms such as Logistic Regression (LR), SVM, and MLP in predicting the price direction of three stock indices. All ML algorithms reached a higher prediction precision than ARIMA, and the SVM models outperformed all other models [22].

Using ML methods for price prediction has gained attention over the past 20 years due to their ability to manage large amounts of data. These methods have been especially useful for stock price and stock price direction prediction which are tasks requiring an extensive amount of data. The data sets are generally too complex for non-ML techniques to handle, meaning that useful information may not be fully utilized [23]. Various models have been examined to predict the stock price direction. Wang, H evaluated 10 different data mining techniques for predicting the price direction of the Hang Seng Index (HSI) using open price, low price, high price, the exchange rate between HKD and USD, and S&P 500 index closing price. The highest predictive performance was displayed by SVM and Least-squares SVM (LS-SVM), which are both algorithms without priori assumptions about the data. K-nearest neighbors (KNN) and the tree-based classification were ranked the lowest in terms of predictive ability, although all approaches yielded a hit ratio over 79% on the out-of-sample data [24]. Gupta, M et al attempted to predict the one-day ahead prices of companies within the

Banking and Financial sector in the National Stock exchange 50 index (NIFTY 50), using a Backpropagation Neural Network (BPNN) with both historical prices as well as macroeconomic factors as input. The suggested model was considered satisfactory for short-term stock index prediction [25]. Generally, ANN models have shown to have strong performance in one day ahead index prediction using closing, opening, lowest, and highest index value [26]. However, in another application of BPNN on financial time series prediction, the method was outperformed by SVM [27]. One major challenge of neural networks is that they are prone to overfitting and especially in tasks with high complexity [28].

More extensive comparisons between models have also been carried through. Ballings, M et al compared various ensemble methods including Random Forest (RF), AdaBoost, and Kernel Factory with single classifier methods such as Neural networks, LR, SVMs, and KNN for predicting the direction of one year ahead stock prices. Macroeconomic factors, as well as yearly fundamentals data for 5767 listed companies, were used as predictors. The results presented that RF had the best predictive performance in terms of median AUC, followed by SVM [29]. Similarly, single classifiers including LR, MLP, and CART were compared to ensemble classifiers such as bagging in quarterly stock return prediction. The experiment covered the Taiwanese stock market and used 19 financial ratios and 12 macroeconomic variables as independent variables, resulting in the ensemble methods producing the most accurate predictions [30]. In addition, other studies concluded that the ensemble classifier RF performed the best, with slightly higher accuracy than SVM for short- and long-term stock price direction prediction [31][32]. Ensemble methods have been notably popular since they have low variance. However in another study where Linear Discriminant Analysis (LDA), LR, ANN, RF and SVM were examined, it was found that SVM and RF outperformed the other models, although SVM was marginally better than RF [4]. Huang, W et al evaluated SVM by comparing its performance to LDA, Quadratic Discriminant Analysis (QDA), Elman BPNN, and a Randon Walk model (RW) for predicting the direction of the NIKKEI 225 index of Japan on a weekly basis. Also here, the highest hit ratio on the out-of-sample was produced by SVM [33]. Generally, SVM has shown to yield high classification accuracy in financial data applications [34][35]. This can be explained by the generalization property of SVM, meaning the SVM adapts well to unseen input data and avoids over-fitting since it uses the structural risk minimization principle. Furthermore, the SVM has desirable properties such as sparse representation and its ability to find global optima [36][37], in contrast to models such as the BPNN for which the gradient descent have led to a local optimum in several cases [38]. This is because training an SVM is done by solving a convex quadratic programming problem which always leads to a global optimum being found [39].

Put together, this literature review indicates that there is yet no established model for stock prediction, which can further serve as a basis for investment decisions. This study aims to test random forest, support vector machine, and logistic regression in order to provide some insight on which algorithms are adequate for Asian small cap stocks in terms of prediction accuracy, as different models have proven different predictive powers based on which context they are applied to. In particular, these three algorithms, especially random forest and SVM, have shown strong performance in previous research depending on the application.

1.6.2 Financial Ratios in Stock Price Prediction

Financial ratios are commonly analyzed to assess a company's performance and financial health. For example, financial ratios have been utilized in financial prediction tasks such as forecasting non-performing loans [40], financial distress and bankruptcies [41]. In the same way, statistical learning techniques can be used to model the relationship between financial ratios and future stock price direction. Prior to finding an appropriate model for stock price prediction, one must find a set of financial ratios that may affect future stock prices and thus can serve as explanatory variables for predicting future stock performance. Several previous studies have attempted to identify the relationship between different financial ratios and stock returns. Anwaar, M conducted a panel regression to see the effect of the five financial ratios: earnings per share (EPS), quick ratio (QR), return on assets (ROA), return on equity (ROE), and net profit margin (NPM) on stock returns. The study was performed on stocks listed on the London Stock Exchange, FTSE-100 Index, over the period 2005-2014 [42]. The results showed that ROA and NPM had a significant positive relationship to stock returns while QR and ROE had no significant impact. Moreover, EPS had a significant negative impact on stock returns. However, Emamgholipour, M et al found that high EPS affected stock returns of the current year positively. Additionally, they observed that price-to-earnings (P/E) and price-to-book (P/B) demonstrated significant negative effects on stock return of current and subsequent year [43]. Another study carried out a panel data analysis to examine the relationship between stock returns and current ratio (CR), earnings yield (EY) and NPM, on stocks listed on the Istanbul Stock Exchange for the period 2008 to 2016. Here, both EY and NPM had a significant positive relationship to stock returns. Two different methods, Parks Kmenta and Beck-Katz were used for this task in order to verify that the results are consistent [44]. Hobarth, L investigated the correlation between 17 financial variables and stocks' performance of US-listed firms during a period of 19 years. Summarized, the results suggested that companies with low book-to-market ratio (B/M), efficient working capital, high equity and low debt level, low total assets value, and high EBIT margin had higher stock returns [45].

Numerous studies have also covered this subject on Asian stock markets. For instance, it was found that there is a significant negative correlation between asset growth and future stock returns in several Asian markets including China, Hong Kong, Indonesia, Malaysia, Singapore, South Korea, and Thailand [46]. Moreover, by using regression it was seen for Indonesian manufacturing stocks that profitability ratios (NPM and ROE), asset turnover and P/B were strong determinants of the market-adjusted stock returns [47]. The predictability of stock returns was tested for Malaysian stocks over the period 2000 to 2009 using dividend yield (DY), EY and B/M as independent variables. The study applied the generalized least squares (GLS) method since the GLS manages heteroskedasticity and autocorrelation more efficiently compared to ordinary least squares (OLS). Ultimately, it turned out B/M had the highest predictive power, although at a 5% significance level all three financial ratios demonstrated predictive power [48]. Bayesian Model Averaging has also been used to study the future stock return predictability using financial information including DY, B/M, EY, default risk premium, monthly rate of three-month treasury bill, term premium, inflation rate, and tern spread. The data set consisted of historical data from 2001 to 2011 for 439 stocks listed on the Stock Exchange (SET). By determining posterior probabilities it was concluded that B/M, default risk premium and inflation rate were the most relevant predictors, although there were stronger predictors for large-cap stocks compared to small-cap stocks [49]. Experimental studies have also been made on the Hong Kong stock market using 17 HSI constituents. Multiple regression analysis was applied to find test the significance of 20 financial variables on stock returns, where the independent variables included ROA, ROE, DY, CR, QR, EPS, EPS ratio, P/E, P/B, price-to-sales (P/S), market capitalization, net profit growth, dividend per share (DPS), debt-to-equity (D/E) and return on capital employed (ROCE). The number of independent variables was reduced to five with factor analysis, and thereafter it could be seen that market capitalization and P/B were positively correlated to stock returns while EPS, DY, and P/S was negatively correlated [50].

Moreover, machine learning techniques have also been used for the same purpose. Delen, D et al tried a two-step approach, first applied explanatory factor analysis to identify the underlying dimension of a large set of financial measures, and thereafter used predictive modeling to find the relationship between the measures and firm performance. The experiment was performed on Turkish listed public companies with historical data covering 2005 to 2011, with four different decision tree algorithms. The earnings before tax-to-equity ratio appeared to be the leading indicator in every decision tree model, followed by the sales growth rate in the CART model and NPM [51]. Predictive models for micro- and small-cap stocks are as mentioned more complex in their nature, but models such as the Fama-French have been developed to explain the anomalies of these smaller stocks [52]. The model includes value, size, profitability, and market factors which in some studies have proved useful when combined with other machine learning methods [9].

Clearly, a wide range of financial variables have been explored in previous studies, but the results vary across the literature. Furthermore many studies have primarily focused on a national stock index, and have addressed only a few financial ratios. In contrast, this research paper aims to assess an extensive amount of variables, for stocks belonging to several countries. In addition, many different statistical methods have been used to examine the relationship between financial variables and future stock prices. As this study aims to predict the direction of stock prices using selected machine learning algorithms, these algorithms will also be applied when studying the association between financial metrics and stock returns. This background assures that previous research has indeed concluded the existence of predictive power between financial measures and price direction for stocks, which further states the relevance of them for this study.

Altogether, the novelties provided in this study are compared to the above presented literature review 1) research on prediction of stock price direction for small-cap stocks in contrast to previous studies which mostly conduct forecasting on large-cap stocks and indices 2) coverage of multiple regions rather than examining only one national index and its stock constituents.

2 Financial Background

This section aims to provide definitions of some central concepts and terminology that are used in this study and is related to the stock market.

2.1 Stocks and the Stock Market

A *stock* is a financial security that gives the owner (commonly called *shareholder*) of the security a share of ownership of the company. Usually, a stock gives the shareholder the right to vote on shareholders' meetings. Also, a company may choose to pay *dividends*, in other words, cash payments, to its shareholders. The stocks of publicly-held companies are traded on stock exchanges, and the prices are determined by supply and demand. Investors can gain profits from stocks either by receiving dividends or by buying stocks at a certain price and sell them at a higher price. However, stocks are seen as risky investments as can yield negative returns as well. Thus, it has been of interest for investors to analyze companies' financial performance and try to predict their future stock prices [53].

2.2 Stock Index

A stock index is a measure of a group of selected stocks or a specific segment of the stock market. Some of the most well known indices include Dow Jones Industrial Average (DJIA), S&P 500, and MSCI World. A stock index is determined by the prices of the stocks included in the index and the most common type of calculation is weighting each stock by its market capitalization. Thus an index corresponds to the performance of a group of stocks and its movements indicate changes in investors' sentiment, the prospects of the economy, and the constituent firms' financial health. Furthermore, there are various classifications for stock indices, for example global (S&P Global 1200 Index, MSCI World), regional (FTSE Asia Pacific Index, MSCI EM Latin America and national (the OMX Stockholm in Sweden, the British FTSE 100 Index, and the Chinese CSI 300 Index), industrialization level (developed, emerging and frontier market indices) or sectoral indices. Stock indices are commonly used by investors to follow the stock market movements. Particularly, in asset management businesses, index performances are regularly compared to the returns of actively managed mutual funds, as the latter is aimed to outperform a specific benchmark index [54].

2.3 Fundamental Analysis and Technical Analysis

Fundamental Analysis and *Technical Analysis* are two common approaches for assessing the future performance of stocks. Fundamental analysis involves estimating companies' intrinsic value and evaluating their long-term financial health by analyzing their financial statements, financial ratios, macroeconomic factors, and industry trends. When making an investment decision, the intrinsic value is compared to the market value.

In technical analysis, future stock price movements are forecasted by examining patterns in historical data such as past prices and trading volumes. In contrast to fundamental analysis, one does not take into account the underlying business of the company. Another major difference is that technical analysis is mostly used for short-term price movement prediction and trading while fundamental analysis is often applied for mid- and long-term investment horizons [7]. As this study aims to predict in mid to long term, the data set will mainly reflect variables typically used in fundamental analysis.

2.4 Other terminology

Multiple financial variables that are used in this study are derived from the financial measures presented below. The theory in this section is based on [53].

- **Market capitalization**: The value of a company's outstanding shares in total, also called equity value. Obtained by multiplying the number of shares outstanding with the price at which the company's share is traded. Stocks are often grouped by their market value, for example, small-cap, mid-cap and large-cap stocks.
- **Enterprise value (EV)**: The EV reflects the total value of a company, in the sense that it does not only take into account the market capitalization but also adds the *net debt* to it. The enterprise value is obtained by EV = Market capitalization + Net debt.
- **Net debt**: Indicates the financial liquidity of a company, by showing how much debt a company has left if its debt is paid with its liquid assets. The

net debt is defined as Short term debt + Long term debt - Cash and cash equivalents.

- **Valuation ratios**: show the relationship between a company's value (either equity value or enterprise value) to a fundamental measure from the income statement (e.g. revenue, EBIT, or net income). They reflect the price one must pay for the stock given a company's sales or profit, and thus give an indication of whether a company might be under or overvalued. Common valuation ratios include:

$$\begin{aligned}\text{Price to Earnings ratio (P/E)} &= \frac{\text{Share price}}{\text{Earnings per Share}},\\ \text{Price to Sales ratio (P/S)} &= \frac{\text{Share price}}{\text{Revenue per share}},\\ \text{EV/EBIT} &= \frac{\text{Enterprise value}}{\text{EBIT}}.\end{aligned}$$

- **Profitability ratios**: can be used to measure how well, during a given period, a company can generate profit compared to for example revenue, shareholders' equity, or assets. Thus, it reflects how efficient a company is in using its assets or revenues. Profitability ratios can be further divided into *margin ratios* and *return ratios*. Example of some margin ratios are:

$$\begin{aligned}\text{Operating margin (EBIT Margin)} &= \frac{\text{Operating income (EBIT)}}{\text{Revenue}} \times 100,\\ \text{Net income margin} &= \frac{\text{Net income}}{\text{Revenue}} \times 100,\end{aligned}$$

and return ratios:

$$\begin{aligned}\text{Return on equity (ROE)} &= \frac{\text{Net income}}{\text{Shareholders' equity}} \times 100,\\ \text{Return on invested capital (ROIC)} &= \frac{\text{Operating income} \times (1 - \text{Tax rate})}{\text{Invested capital}} \times 100,\\ \text{Return on assets (ROA)} &= \frac{\text{Net income}}{\text{Total Assets}} \times 100.\end{aligned}$$

- **Liquidity ratios**: are important ratios as they help to evaluate if a company will be able to pay its short-term debt obligations with its liquid or currents assets. Two of the most common liquidity ratios include:

$$\begin{aligned}\text{Current ratio} &= \frac{\text{Current assets}}{\text{Current liabilities}},\\ \text{Quick ratio} &= \frac{\text{Cash and cash eq.} + \text{Accounts receivable} + \text{Marketable securities}}{\text{Current liabilities}}.\end{aligned}$$

- **Leverage ratios**: show the relative amount of debt a company has taken. It reflects the company's ability to pay its long-term debt obligations, and thus also gives an indication of the financial health of a company. A widely used leverage ratio is:

$$\text{Debt to equity ratio} = \frac{\text{Total debt}}{\text{Total equity}}.$$

- **Dividends**: Are payments that a company gives to its shareholders. To compare dividends across different companies, there are ratios such as:

$$\begin{aligned}\text{Dividend yield} &= \frac{\text{Dividend per share}}{\text{Share price}},\\ \text{Dividend payout ratio} &= \frac{\text{Total dividends paid}}{\text{Net income}}.\end{aligned}$$

3 Mathematical Background

This chapter aims to establish terminology and describe the theoretical background and mathematical concepts for the methods applied in this research. Included are the mathematical models used as well as metrics to evaluate the final models. Vectors will be marked in bold and observed values are in lower case letters.

3.1 Machine Learning

Machine Learning (ML) is a field within Artificial Intelligence and Data Science, focusing on developing computer programs that can learn and improve through new experience by automation. One of the major benefits of ML models is that they have proven to efficiently identify patterns in large data sets and multi-dimensional data. The ability of ML algorithms to learn from their environment has made ML applicable in many various fields such as finance, marketing, medicine, engineering, and transportation. Also, as ML-based models are fed with new data they generally become more efficient and accurate. Yet, the use of ML models also comes with challenges. Mainly they have shown limited prediction accuracy in various applications due to their susceptibility to the input data. For example, erroneous results may be produced when models are trained with insufficient, incomplete, or biased data sets [55, p.1]. Some ML techniques are also prone to overfitting which makes the model less responsive to new input data, but various solutions have been introduced to deal with this issue [56].

3.2 Supervised Learning

The four approaches in ML are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised learning is a machine learning approach where the training is conducted on labeled data. Observations with the true responses are used for training a model, which aims to predict or classify the correct outcomes when inputted with new unseen data. Thus, the purpose of supervised learning is to model the relationship between explanatory variables and the response variable(s). A supervised learning algorithm intents to find the unknown function f that can map the predictor or feature space $\boldsymbol{X} = \left[X_1, X_2, \ldots, X_p\right]^T$ to the output space Y, such that

$$Y = f(\boldsymbol{X}), \quad f \in \mathcal{F}$$

where $\mathcal{F}$ denotes function space.

The data set on which the model is trained consists of a set of observations, where each observation $i, i = 1, \ldots, N$, is a tuple $(\boldsymbol{x_i}, y_i)$. Hence, a training set with N observations can be denoted as $\mathcal{T} = \left\{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_N}, y_N)\right\}$ where for the i:th observation, $\boldsymbol{x_i}$ is a vector of p features and y_i is the corresponding label [57, p.21]. The training data is generally assumed to be *independent and identically distributed* (*i.i.d*). Using the training set, the underlying function f can be estimated to $\hat{f}$ so that

$$Y \approx \hat{f}(\boldsymbol{X}).$$

Supervised learning problems can be divided into classification and regression. Therefore, in a supervised learning task, the choice of model highly depends on the type of problem, and there are algorithms designed for classification and regression respectively. In classification Y is categorical, meaning that an observation can be assigned to one specific class among a discrete number of K different classes, where $K \geq 2$. If the output Y is a quantitative variable and can take on numbers in the real value space $\mathbb{R}$, it is referred to as a regression problem [57, p.28].

This thesis focuses on binary classification, meaning it aims to predict which class an observation belongs to, among two classes. Here, the response variable is a binary random variable Y_i with the output space $Y \in \{\textit{Excess, Not Excess}\}$. The random explanatory variable $\boldsymbol{X_i} \in \mathbb{R}^{\mathrm{p}}$ represent fundamental information such as revenue, profit margin and other financial metrics about stock i.

3.2.1 Parametric vs Non-parametric Models

Furthermore, different learning algorithms are built on different assumptions. Generally, statistical learning models can be divided into parametric models and non-parametric models. Parametric models assumes the form of the function f, where f depends on a finite number of parameters $\boldsymbol{\beta}$, representing the information in the data. Consequently, rather than estimating the functional form of f, the problem is simplified to only estimating the coefficients $\boldsymbol{\beta} = \left[\beta_0, \beta_1, \ldots, \beta_p\right]$. The drawback of parametric models is that they are constrained to the predetermined function which may be far away from the correct f, leading to the risk of generating inaccurate predictions. In contrast, non-parametric models do not assume anything about the distribution and parameters of f, allowing the model to find an approximation of f which is more adjusted to the observations. Although non-parametric approaches are higher in flexibility, they can become very complex as the number of parameters is not definite. Also, since the entire functional form of f has to be approximated, non-parametric models usually require a larger training set [57, pp. 21-23].

3.3 Support Vector Machines

The support vector machine (SVM), also referred to as support vector machines (SVMs), is a classification model which extends the support vector classifier by enabling enlargement of the feature space using *kernels* to allow finding a linear boundary for classes in a higher dimension, if such cannot be found with the existing features. Thus when the relationship between the predictors and response variable is nonlinear, the SVM allows combating this nonlinearity [57]. SVM is developed for binary classification but has been developed to allow for more than two classes. For better intuitive understanding the mathematical theory for the SVM will be presented for a binary classification model. We denote two observations as $\boldsymbol{x_i}$ and $\boldsymbol{x_{i'}}$ in which the inner product of the two observations are

$$< \boldsymbol{x_i}, \boldsymbol{x_{i'}} >= \sum_{j=1}^{p} \boldsymbol{x_{ij} x_{i'j}}.$$

By this the linear SVM can be defined as per [57],

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i \boldsymbol{x}\boldsymbol{x_i}.$$

3.3.1 Separating Hyperplanes

The support vector machine is derived from the idea of separating linear hyperplane in the feature space. For two classes, y_1 and y_2, where each response y_i is assigned one of the classes based on the separating hyperplane

$$\{x : f(x) = \boldsymbol{x^T}\beta + \beta_0 = 0\}, \ \ ||\beta|| = 1,$$

which in turn classifies based on

$$y_i = \begin{cases} 1, \ f(\boldsymbol{x_i}) > 0 \\ -1, \ f(\boldsymbol{x_i}) < 0 \end{cases} , \ \ y_i f(\boldsymbol{x_i}) > 0 \quad \forall i.$$

The points which lie on $f(x) = 0$ thus lie on the hyperplane and the optimization problem will target finding the best separating hyperplane farthest from the training points, i.e. the largest margin M between the training points $T = \{y_i,\boldsymbol{x_i}\}_{i=1}^{n}$ in the two classes [58].

Therefore, the optimization problem to find a linear separable hyperplane with the training points on the correct side of the hyperplane can be formulated as

$$\begin{aligned} \max_{\beta_0,\boldsymbol{\beta},||\boldsymbol{\beta}||=1} \quad & M \\ \text{s.t.} \quad & y_i(\boldsymbol{\beta}^T\boldsymbol{x_i} + \beta_0) \geq M \quad \forall i. \end{aligned} \tag{1}$$

The above equation has no solution for $M > 0$ when the classes are not linearly separable. To allow for some miss classification in order to find a good enough margin, a *soft margin* is used by changing the optimization constraints.

$$\begin{aligned} \max_{\beta_0,\boldsymbol{\beta},||\boldsymbol{\beta}||=1,\boldsymbol{\epsilon}} \quad & M \\ \text{s.t.} \quad & y_i(\boldsymbol{\beta}^T\boldsymbol{x_i} + \beta_0) \geq M(1 - \epsilon_i), \\ & \epsilon_i \geq 0, \\ & \sum_{i=1}^{n} \boldsymbol{\epsilon_i} \leq C. \end{aligned} \tag{2}$$

Here, C is a non-negative tuning parameter seen as a penalty for the allowed misclassification and M the smallest distance from the hyperplane to the training observations [57]. If $C = 0$ the optimization problem does not allow for misclassification. ϵ_i is introduced as a slack variable that tells where the ith observation is located in relation to the hyperplane and margin, and is proportional to the amount of which the classification y_i is on the wrong side of the margin. If $\epsilon_i = 0 \quad \forall i$ then none of the predictors violate the margin.

The optimization problem above is better known as the *support vector classifier* and is appropriate when finding a good enough linear separator for the classes. Instead, the support vector machine allows an enlargement of the feature space for the predictors in order to find a separating hyperplane in the enlarged space. As an example, the basis function $h_m(\boldsymbol{x_i}) = (h_1(\boldsymbol{x_i}), \ldots, h_M(\boldsymbol{x_i}))$ can transform the feature space from $\mathbb{R}^{\mathrm{p}}$ to $\mathbb{R}^{2\mathrm{p}}$, meaning that translating the linear decision boundary found in the enlarged feature space would translate to a quadratic decision boundary with the original p predictors. The new non-linear function will instead decide the class of y_i according to $sign(\hat{f}(\boldsymbol{x_i}))$ where $\hat{f}(\boldsymbol{x_i}) = h^T(\boldsymbol{x_i})\boldsymbol{\beta} + \beta_0$.

3.4 Decision Trees and Random Forest

Decision trees are non-parametric tree-liked models for decisions, used in both regression and classification problems. Decision tree models use splitting rules to segment the predictor space into several distinct regions. By learning decision rules deducted from the training data, decision tree models aim to predict the response variable value. These models have been widely used in many ML applications due to their interpretability. Yet, they tend to be lack robustness, meaning they can be sensitive to variations in the data as small changes in the data can change the resulting tree significantly. To combat this problem, ensemble methods such as *Bagging*, *Boosting* and *Random Forest* have been introduced.

In the context of this study, the following sections will focus on *classification trees* rather than *regression trees.* Random forest will be the main focus of this study but in order to understand the random forest classifier, the concepts of classification trees and bagging and will be explained first.

3.4.1 Classification Trees

Classification trees are applied when predicting the category or class of an observation. Broadly, the classification process starts from the root node (top) of the tree where an observation is assigned to a sub-space according to a split rule. Thereafter, the observation reaches another decision node where the predictor space is again divided into sub-spaces and an observation is branched by another split rule, and so on. Each split rule consist of a threshold t_j for a predictor variable X_j, for example $X_j < t_j$ and $X_j \geq t_j$ for the left and right branch of a node respectively. In other words, the observation continues getting assigned to different sub-spaces of the predictor space at the internal nodes, which are the nodes in between the root node and the terminal nodes. This continues until reaching the leaves (terminal nodes) of the tree where the outcome is predicted.

The development of a classification tree can briefly be described as follows: First the p-dimensional predictor space is partitioned into J disjoint regions $R_1, R_2, \ldots, R_J$, through recursive binary splitting. The predictor space comprises the possible values of the predictors $X_1, X_2, \ldots, X_p$. When a new observation is assigned to a region R_j, the observation will be classified according to the be the *most commonly occurring class*, also called *majority class*, of the training data in that particular region R_j.

The recursive binary splitting starts from the top of the tree and then continuously stratifies the predictor space, where each split yields two new branches one level down on the tree. A predictor X_j and a cutpoint s is chosen by the algorithm so that the predictor space is divided into two regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ by using a *node impurity measure* $Q_m(T)$ as criterion, such as the misclassification error which the proportion of the training observations in region m which do not belong to the most commonly occurring class.

In mathematical terms, let $\hat{p}_{mk}$ denote the share of training observations belonging to class k in the m:th region. Then $\hat{p}_{mk}$ can be represented by:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I\left(y_i = k\right), \tag{3}$$

where $I\left(\cdot\right)$ is the indicator function, R_m represents region m and N_m is the number of observations in that region. Then observations $x \in R_m$ are classified according to:

$$k(m) = \arg\max_k \hat{p}_{mk}. \tag{4}$$

The are various measures that can be used by the algorithm to select the best split. Some node impurity measures $Q_m(T)$ that are commonly chosen for splitting the predictor space are *Misclassification error*, *Gini index* and *Cross-entropy* which are defined as follows:

$$\text{Misclassification error: } \frac{1}{N_m} \sum_{i \in R_m} I\big(y_i \neq k(m)\big) = 1 - \hat{p}_{mk(m)}, \tag{5}$$

$$\text{Gini index: } \sum_{k \neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}\left(1 - \hat{p}_{mk}\right), \tag{6}$$

$$\text{Cross-entropy: } -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}. \tag{7}$$

As mentioned the predictor space can be divided into the regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ at a given node such that the split minimizes the misclassification error. Yet, when growing a classification tree, Gini index and cross-entropy are often selected over misclassification error for evaluating the quality of a split. The Gini index and cross-entropy are differentiable and thus more suitable for numerical optimization. Additionally, compared to misclassification error, the cross-entropy and the Gini index have a higher sensitivity to node purity. If the values of the $\hat{p}_{mk}$:s are close to 0 or 1 the Gini index and cross-entropy will be near 0. This means that a small value of the Gini index or cross-entropy indicates that node m is pure. When dealing with a large data set, the Gini index may be more beneficial as the cross-entropy contains the more computationally expensive logarithm calculation. Note that the recursive binary splitting algorithm is greedy, as it at each step of the process makes the best split for that step rather than also considering the future steps when determining the best split [58, pp. 309-310].

3.4.2 Bagging

As mentioned before, one of the main drawbacks of decision trees is their high variance, meaning that they are sensitive to variations in the training data in the sense that a change in the data could yield an entirely different optimal decision tree. *Bagging*, short for *Bootstrap aggregation*, is used for decreasing the variance of a decision tree. *Bootstrapping* is equivalent to doing random sampling with replacement. The idea is to resample observations from a single data set to obtain several simulated data sets and make inferences about the corresponding population.

Suppose one has a a set of n independent random variables $Z_1, \dots, Z_n$ with mean $\bar{Z}$ and variance σ_Z^2. The variance of the mean $\bar{Z}$ is thus $\frac{\sigma_Z^2}{n}$, showing that taking the average of a set of random variables or observations leads to a smaller variance. In the same way one could develop several distinct decision trees using different training sets and take the average of the predictions of the models in order to lower the variance and consequently improve the prediction accuracy of a model. Now assume a training set $\mathcal{T} = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \dots, (\boldsymbol{x_N}, y_N)\}$ is used to fit a model and the prediction $\hat{f}(\boldsymbol{x})$ is obtained by an input $\boldsymbol{x}$. The bagging method first generates bootstrap samples from the original training set and for each bootstrapped training set $\mathcal{T}_b^*$, $b = 1, 2, \dots, B$, a model is fitted which provides a prediction $\hat{f}^{*b}(\boldsymbol{x})$. In a regression setting the average of the predictions is thereafter calculated, which is given by:

$$\hat{f}_{\text{bag}}(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(\boldsymbol{x}). \tag{8}$$

For classification with a classifier $\hat{G}(\boldsymbol{x})$, class k is assigned to $\boldsymbol{x}$ according to the *majority vote*:

$$\hat{G}_{bag}(\boldsymbol{x}) = \arg\max_k \hat{p}_k(\boldsymbol{x}), \quad k = 1, 2, \ldots K, \tag{9}$$

where $\hat{p}_k(\boldsymbol{x})$ is the proportion of trees assigning class k to the input $\boldsymbol{x}$, defined as:

$$\hat{p}_k(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^{B} I \left\{ \hat{f}^{*b}(\boldsymbol{x}) = k \right\}. \tag{10}$$

In cases when there are particular predictors which affect the response variable strongly, the trees often become similar and highly correlated since such predictor tends to placed at the root node for the majority of the trees. Taking the average of correlated trees results in a lower reduction in variance compared to averaging uncorrelated trees. That is, the impact of bagging will not be as significant as supposed to, when dealing with correlated trees.

3.4.3 Random Forest

Random Forest is an ensemble learning method that has been introduced as an enhancement of bagging. Random forest *decorrelates* the trees by allowing to choose one predictor out of a random subset of $m \leq p$ predictors when constructing a split, instead of choosing from all p features. At each split, a new random sample of m predictor is obtained. By only letting a subset of the predictors be candidates for each split, the problem of producing similar trees is avoided. On average $\frac{p-m}{p}$ splits will not use a certain predictor meaning that the strong predictor cannot be used as top split in all trees.

The test error of a bagged or random forest model can be estimated by the *out-of-bag* (OOB) error. Recall that in bagging (and hence also in random forest), bootstrapped subsets of the observations are used to fit multiple trees. In each bootstrap sample $\mathcal{T}_b^*$, $b = 1, 2, \ldots, B$, approximately one third of the observations are not used, which are called OOB samples. For each $\boldsymbol{x_i}$ in the training set, aggregate the votes solely over those trees that are fitted on bootstrap training samples $\mathcal{T}_b^*$ that do not contain x_i. This can be referred to as the OOB classifier. One can obtain an OOB prediction for each observation x_i, $i = 1, 2, \ldots, N$, by using the corresponding OOB classifier and taking the majority vote. Thereafter, by computing the misclassification error for these predictions, one obtains the OOB error.

When applying the random forest algorithm, there are several parameters for which a value must be set. To start with, the number of predictors that are considered at each split must be chosen. Setting a large predictor subset size m increases the chance of picking the predictors carrying the most information, but may also increase the correlation between the trees. Thus, when having many correlated predictors, a small m is more beneficial. Most commonly m is chosen such that $m \approx \sqrt{p}$. Furthermore, the number of trees B is relevant, particularly for the variance of the model, as a large number of trees will decrease the variance and improve the prediction. More precisely, the OOB error can be shown as a function of B, where the OOB error decreases as B grows. Nevertheless, using many trees comes with a higher computational time and as the number of trees increases the OOB error will decrease at a slower rate. Thus, the value of B should be chosen such that is sufficiently large for the OBB error to converge [57, pp. 317-321].

3.4.4 Variable Importance in Random Forest

In random forest, the are several approaches for obtaining variable importance measures, for instance, the *Gini importance* (also called the *Mean decrease in node impurity*). As mentioned before, at each split the optimal split is selected using a criterion such as the Gini index, or better known as Gini impurity. The Gini impurity measures the likelihood that an observation is mislabelled if it is randomly classified according to the distribution of class labels in the data set. It reflects how impure a node is, where a node is called pure if all observations belong to one specific class.

It is possible to see how much the impurity of a split decreases for each feature. To obtain a feature importance measure, one looks at how much the Gini impurity decreases for a feature X_j at a split, in each tree where the feature was used, and then takes the average over the trees. Repeating this procedure for each feature $j = 1, \ldots, p$, yields the relative feature importances [57, p.319].

3.5 Logistic Regression

Logistic Regression is a parametric method which classifies an observation by estimating the posterior probability of the observation belonging to a certain class, rather than predicting the response variable Y itself. The algorithm uses the logistic function $\pi(\boldsymbol{X})$ to map the outcome of the regression to the interval $[0, 1]$. Consider a binary classification problem with p features $\boldsymbol{X} = \left[X_1, X_2, \ldots, X_p\right]^T$ and a response variable $Y \in \{0, 1\}$. Then the conditional probability that an observation $\boldsymbol{x_i}$ belongs to class 1 is given by:

$$\pi(\boldsymbol{x_i}) = P(Y = 1 \mid \boldsymbol{X_i} = \boldsymbol{x_i}) = \frac{e^{\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1 + e^{\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_i}}. \tag{11}$$

Hence, for the probability of $Y = 0$:

$$P(Y = 0 \mid \boldsymbol{X_i} = \boldsymbol{x_i}) = 1 - P(Y = 1 \mid \boldsymbol{X_i} = \boldsymbol{x_i}) = \frac{1}{1 + e^{\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_i}}, \tag{12}$$

where β_0 is the intercept and $\boldsymbol{\beta} = \left[\beta_1, \ldots, \beta_p\right]^T$ are the regression coefficients.

The parameter are usually estimated by the maximum likelihood method, which is explained in section 3.5.1. If the obtained probability $P(Y_i = 1 \mid \boldsymbol{X_i} = \boldsymbol{x_i})$ is equal to or higher than a predetermined cutoff value c, observation $\boldsymbol{x}_i$ is assigned to class 1. Hence, the prediction $\hat{y}_i$ for the i:th observation can be expressed as:

$$\hat{y}_i = \begin{cases} 1, & \text{if } \pi(\boldsymbol{x_i}) \geq c, \\ 0, & \text{if } \pi(\boldsymbol{x_i}) < c. \end{cases} \tag{13}$$

The logistic function always takes the form of a an S-shaped distribution function. However, if a monotone transformation of $\pi(\boldsymbol{x_i})$ is linear in $\boldsymbol{x_i}$ the decision boundaries will be linear as well. In logistic regression, the *logit* or *log-odds* transformation is used, which here is a monotone transformation. By calculating the log-odds $log[p/(1-p)]$ one can derive:

$$\log\left(\frac{\pi(\boldsymbol{x_i})}{1-\pi(\boldsymbol{x}_i)}\right) = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_i, \tag{14}$$

which is linear in $\boldsymbol{x}_i$. In addition, equation 14 is also the inverse of the logistic function.

Logistic regression is frequently used in supervised learning tasks since it is easy to implement and interpret. In addition, the feature importance can be inferred from the estimated coefficients. As shown, logistic regression produces linear boundaries and thus it may produce inaccurate results for non-linearly separable data.

3.5.1 Fitting the Logistic Regression Model

The regression parameters $\beta_0, \beta_1, \ldots, \beta_p$ are unknown and can be estimated with the training set using *maximum likelihood*. Let $\theta = \{\beta_0, \boldsymbol{\beta}\}$ and assume the training set consists of N observations $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)$. Then one obtains the conditional probabilities:

$$\pi(\boldsymbol{x}_i; \boldsymbol{\theta}) := P\left(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}_i; \boldsymbol{\theta}\right) = \frac{e^{\beta_0 + \boldsymbol{\beta}^T x_i}}{1 + e^{\beta_0 + \boldsymbol{\beta} x_i}}, \quad i = 1, \ldots, N. \tag{15}$$

The maximum likelihood estimation (MLE) is performed by first constructing the likelihood function and then finding estimates $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ which maximize the likelihood function. In the case where the response variable is binary, the probability can be modeled by a Bernoulli distribution. The likelihood function can then be expressed as follows:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{N} \pi(\boldsymbol{x}_i; \boldsymbol{\theta})^{y_i} \left(1 - \pi(\boldsymbol{x}_i; \boldsymbol{\theta})\right)^{1-y_i}. \tag{16}$$

Since the logarithm is a monotonic function, the same value of θ is obtained by maximizing the likelihood function as by maximizing the *log-likelihood*. To simplify the maximization problem the log-likelihood is used, which changes products to sums:

$$\begin{aligned} l\left(\boldsymbol{\theta}\right) &= \sum_{i=1}^{N} \Big[y_i \log \pi\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right) + \left(1-y_i\right) \log\big(1-\pi\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right)\big)\Big] \\ &= \sum_{i=1}^{N} \left[y_i \left(\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_i\right) - \log\left(1 + e^{\left(\beta_0+\boldsymbol{\beta}^\top\boldsymbol{x}_i\right)}\right)\right]. \end{aligned} \tag{17}$$

To find the values $\theta = \{\beta_0, \boldsymbol{\beta}\}$ for which the log-likelihood function is maximized, one has to differentiate $l\left(\boldsymbol{\theta}\right)$ with respect to the $p+1$ parameters and set the derivatives to zero:

$$\frac{\partial l(\beta_0,\boldsymbol{\beta})}{\partial \beta_0} = \sum_{i=1}^{N} \big(y_i - \pi\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right)\big) = 0, \tag{18}$$

$$\frac{\partial l(\beta_0,\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{N} x_{ij}\big(y_i - \pi\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right)\big) = 0, \quad j = 1,\ldots,p. \tag{19}$$

This gives $p+1$ equations which are non-linear in $\boldsymbol{\theta}$. The equations are transcendental and do not have closed-form solutions. Therefore, it is usually solved numerically using the *Newton-Raphson* method. The first step then is to calculate the second derivatives:

$$\frac{\partial^2 l(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top} = -\sum_{i=1}^{N} \boldsymbol{x}_i\boldsymbol{x}_i^\top \pi\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right)\big(1-\pi\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right)\big). \tag{20}$$

Let $\boldsymbol{\theta}^{(n)}$ be the approximation of $\boldsymbol{\theta}$ obtained from a new Newton iteration, and $\boldsymbol{\theta}^{(n-1)}$ from the previous iteration,

$$\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)} - \left(\frac{\partial^2 l(\boldsymbol{\theta}^{(n-1)})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\right)^{-1} \frac{\partial l(\boldsymbol{\theta}^{(n-1)})}{\partial\boldsymbol{\theta}}. \tag{21}$$

The iteration continues until the algorithm converges to an estimate $\hat{\boldsymbol{\theta}}$ (although convergence is not guaranteed) [58, pp. 19-21].

3.6 Model Evaluation Metrics

When developing machine learning models, a central part of the process is to evaluate the performance of the models. There are various evaluation metrics that can be used to measure the accuracy, check the robustness, and compare the different models. This section presents the evaluation models that are used in this study.

3.6.1 Confusion Matrix

A confusion matrix is a $K \times K$ table (where K is the number of classes) that shows the number of correctly classified and misclassified observations. It is commonly used in classification tasks with data sets where the true values are known. Going forward, the focus of this section will be on the binary case, in other words, when the confusion matrix is 2×2.

The table displays the actual and predicted values and categorizes observations into *true positives* (TP), *true negatives* (TN), *false positives* (FP) and *false negatives* (FN). For a classification setting with classes 1 (outperforming stocks) and 0 (underperforming stocks) the above terms could be described as follows:

- **TP:** Observations the model predicts to class 1, and do actually belong to class 1.
- **TN:** Observations the model predicts to class 0, and do actually belong to class 0.
- **FP:** Observations the model predicts to class 1, but in reality belong to class 0.
- **FN:** Observations the model predicts to class 0, but in reality belong to class 1.

There are two types of misclassifications, FP and FN, which are typically referred to as *Type I* error and *Type II* error respectively. An illustrative confusion matrix is presented below.

		Predicted	
		1	0
Actual	1	TP	FN
	0	FP	TN

Depending on the task, the relevance of these two errors varies a lot, as one or both of these errors may lead to severe outcomes in some cases. For instance, in stock price direction prediction, classifying an actual underperforming stock as an outperformer could lead to a significant loss of money. Consequently, it is of necessity to obtain a low number of false positive observations in this experiment. On the other hand, classifying an outperforming stock as underperformer means missing out on a potentially successful investment, which is particularly negative for portfolio managers, whose main goal is to reach a higher return than an index. Hence, it is of interest to focus on both errors in this paper.

From the four items in the confusion matrix other measures can be calculated such as accuracy, sensitivity, precision and recall, which are defined as:

$$\text{Accuracy } = \frac{TP + TN}{\text{Total Population}} = \frac{\text{no. of correct classifications}}{\text{Total Population}}, \tag{22}$$

$$\text{Precision } = \frac{TP}{TP + FP}, \tag{23}$$

$$\text{Recall } = \frac{TP}{TP + FN}, \tag{24}$$

$$\text{F-measure } = \frac{(1+\beta)^2 \cdot \text{ Recall } \cdot \text{ Precision}}{\beta^2 \cdot \text{ Recall } + \text{ Precision}}. \tag{25}$$

The accuracy is simply the proportion of correct classifications. Precision is the percentage of correctly classified positive observations out of all observations that have been predicted positive. Recall measures the share of correctly predicted positive observations out of all positive observations in the dataset. Precision and recall are in particular beneficial when evaluating a classification model that has been trained on imbalanced data, but the different measures provide different information. For example, if the aim is to correctly classify as many outperforming stocks as possible then one wants the FN to be as low as possible, implying that the recall must be high while the precision could be lower. Conversely, if one seeks to minimize the number of underperforming stocks that are misclassified, the FP needs to be as low as possible, while one can allow for some FN. That is, a high precision is more important compared to recall. Lastly, the F-measure considers both precision and recall, but incorporates a factor β, such that the recall is as β times as important as precision [59].

3.6.2 Kappa Statistic

Kappa Statistic, known as Cohen's Kappa, is a statistic that takes into consideration that a correct classification occurring by chance. The calculation is comparing the observed agreement or accuracy, to the expected agreement or accuracy that is presented by pure guess. The Kappa Statistic is measured from 1 to -1, where 1 is a perfect agreement, 0 what can be expected from a pure guess, and -1 that it is worse than the pure guess. In general, a value above 0.2 shows fair agreement in prediction and actual value [60]. The calculation is presented below with a matrix to exemplify

Actual	1	0	*total*
Predicted			
1	A	C	m
0	B	D	k
total	n	p	q

where $q = A + B + C + D$ and m, k, n, p are the sums for their respective row or column. Then

$$
\begin{aligned}
p_e &= \tfrac{nm}{q} + \tfrac{pk}{q},\\
p_0 &= \tfrac{A+D}{q}.
\end{aligned}
$$

Then the Kappa Statistic can be calculated as

$$
Kappa = \frac{p_0 - p_e}{1 - p_e}. \tag{26}
$$

3.6.3 Receiver Operator Characteristic Curve

The receiver operator characteristic (ROC) curve is another evaluation tool for classification models. The ROC curve is presented in a two dimensional graph with the *True Positive Rate* (TPR) on the y-axis and the *False Positive Rate* (FPR) on the x-axis. The ROC curve shows the relative trade-offs between benefits and costs, represented by the true positives and false positives respectively. The coordinate (0,0) in a ROC graph implies that there are never positive classifications, meaning that there are no false positive errors nor true positives. Similarly, the coordinate (1,1) represents the scenario where only positive classifications are produced. Lastly, the best classification is reflected by the coordinate (0,1). The closer a classifier is to the northwest corner of the ROC space, the better it is considered. A ROC curve on the diagonal line $y = x$, also called *line of no discrimination*, corresponds to a classifier that randomly guesses.

The evaluation of a discrete classifier (classifier that directly predicts the class), is presented as a single point in the ROC graph. However, in many applications probabilistic classifiers such as logistic regression are used, which produce the class probability rather than directly the class decision itself. For these classifiers, a probability threshold value c is set, such that if a (binary) classifier predicts a higher value than the threshold, an observation is assigned to a certain class, and otherwise to the other class. The ROC curve plots the TPR and FPR at different probability thresholds, meaning that the TPR and FPR can be written as functions of c:

$$
\begin{aligned}
TPR(c) = \frac{TP(c)}{FN(c) + TP(c)} &= \frac{FP(c)}{\text{No. of positives}}\\
= 1 - \frac{FN(c)}{\text{No. of positives}} &= 1 - FNR(c),
\end{aligned} \tag{27}
$$

$$FPR(c) = \frac{FP(c)}{FP(c) + TN(c)} = \frac{FP(c)}{\text{No. of negatives}}, \tag{28}$$

where the TPR is equivalent to recall and the FPR is called *specificity*.

3.6.4 Area Under Receiver Operator Characteristic Curve

Models can also be evaluated using the *area under receiver operating curve* (AUC). It is a measure of classifier performance and it close to 1 if the performance is good, meaning that the model is classifying almost all observations correctly, and close to 0 if the performance is poor. The AUC can be determined by calculating the area under the ROC curve by trapezoidal integration, using the equations (28) and (27) :

$$AUC = \sum_i \{[1-(1-TPR(c)_i) \times \Delta FPR(c)] + \frac{1}{2}[\Delta TPR(c) \times \Delta FPR(c)]\}, \tag{29}$$

where

$$\Delta FPR(c) = FPR(c)_i - FPR(c)_{i-1}, \tag{30}$$

and

$$\Delta TPR(c) = TPR(c)_i - TPR(c)_{i-1}, \tag{31}$$

as provided by [61].

3.7 Confidence Interval

In classification, the classification error is derived from the sample data set used when fitting a model. However, the sample error $\widehat{\varepsilon_N}$ is not necessarily representative of the true error ε_N, meaning the error that would have been obtained using the entire population. To deal with this, one can construct a *confidence interval* (CI) for the classification error. A CI provides a range of values, in which the true value of an *population parameter* likely lies. A population parameter is a parameter value calculated on an entire population, rather than from a sample. Computing the CI for a statistic allows for generalization of the results to any other sample data set and the full population.

Let N be the number of observations and r be the number of incorrect classifications. Given a binary response variable Y, r can be seen as binomial random variable:

$$r \sim \text{Bin}(q, N). \tag{32}$$

For sufficiently large N, the binomial distribution can be approximated by a normal distribution:

$$r \sim \text{Bin}(q, N) \approx \mathcal{N}(Nq, Nq(1-q)), \tag{33}$$

where, $\hat{q}$ is estimated as:

$$\hat{q} = \frac{r}{N} \sim \mathcal{N}\left(q, \frac{q(1-q)}{N}\right). \tag{34}$$

The CI can then be calculated using a Z-distribution:

$$CI_{1-\alpha} = \hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{q}(1-\hat{q})}{N}} \tag{35}$$

where α is the chosen confidence level.

4 Methodology

The Data and Methodology section describes what data has been selected, how the data has been preprocessed, what mathematical models have been chosen and lastly how the models are trained and evaluated.

4.1 Research Outline

The methodology for this research paper can be summarized in figure 1.

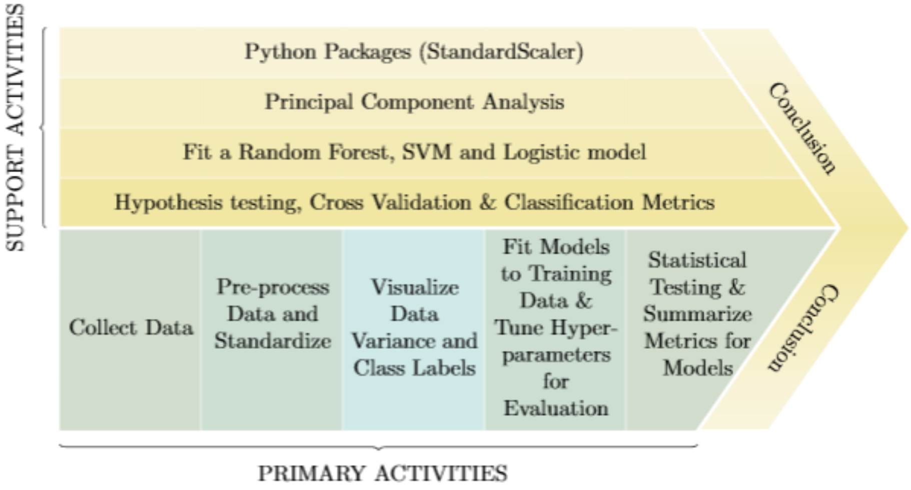

SUPPORT ACTIVITIES
Python Packages (StandardScaler)
Principal Component Analysis
Fit a Random Forest, SVM and Logistic model
Hypothesis testing, Cross Validation & Classification Metrics
Conclusion
Collect Data
Pre-process Data and Standardize
Visualize Data Variance and Class Labels
Fit Models to Training Data & Tune Hyper-parameters for Evaluation
Statistical Testing & Summarize Metrics for Models
Conclusion
PRIMARY ACTIVITIES

Figure 1: Methodology for Thesis

The goal of this study is to examine the statistical predictive power of three machine learning methods, random forest, SVM, and logistic regression, with the use of financial and macroeconomic variables. In order to do so, an initial dialogue with the Asset Management team at Handelsbanken was held to pin which variables are of interest, as well as a literature review. These variables were downloaded from a Bloomberg terminal where the variables could have unit USD (for example market capitalization), percentage (for example EPS growth or return on equity), or unitless ratios (for example P/E). Some variables were transformed to percentage change since this is assessed to yield more relevant information, which will be described in table 1. All variables were then standardized using built-in functions in Python. The same variables were used for the yearly and quarterly prediction models. To also see the variance of the data, the largest principal components were visualized, with the responses (1 for excess return, 0 for not excess return) labeled for the observations. Limited mathematical analysis was conducted with the principal component graphics as it is done solely to visualize part of the data. The models are then trained, tested, optimized, and tested for statistical power. Only then are final conclusions drawn.

4.2 Data Collection

The observations start from the period 12/31/2010 to 12/31/2020, with quarterly intervals with data from the last day of each quarter. The initial sample contains quarterly figures for 1110 stocks which correspond to a total of 45,797 observations. The data comprises closing prices, 35 financial variables for each company, and 6 macroeconomic variables. The data is collected from a Bloomberg terminal and as mentioned before, the collection is focused on countries in Asia. For an observation to qualify, it needed to fulfill both of the following criteria:

- Based in one of the following countries: China, Hong Kong, India, Indonesia, Malaysia, Philippines, Singapore, South Korea, Thailand, Taiwan and Vietnam.
- Have a market capitalization between 20 and 9,500 million USD.

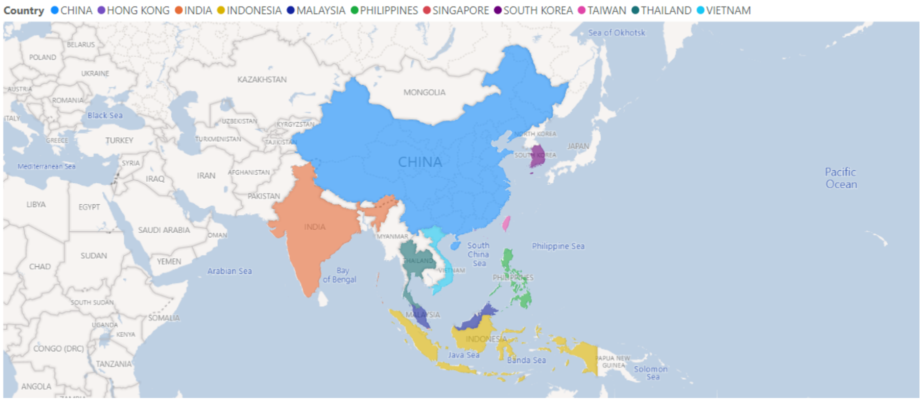


Figure 2: Geographical location of countries included in scope

The upper limit for market capitalization is set higher than the largest constituent's market value in the benchmark index, in order to capture those that previously have been small-cap stocks but grown to be mid- or large-cap.

All variables are described in table 1 below. The chosen range of the data set depends on several factors. Firstly, a ten-year period was used in order to decrease the risk of building a model on a too small data set. Secondly, to make the model adapted to the characteristics of the current stock market, 2020-12-31 was chosen as the end date, as it was the most recent quarter with companies' financial statements released. Furthermore, the frequency of the data was set to quarterly to avoid short-term fluctuations in stock prices, since the goal of this study is to predict on a quarterly and one year basis. Also, the majority of the companies included in the data set report their financial statements on a quarterly basis, hence for a fundamental analysis-based prediction task it comes naturally to use quarterly data sampling.

The initial data collection process was mainly focused on acquiring data for as many observations as possible. This is especially important as the financial data coverage was incomplete for small-cap companies in Asia. Thus, many observations may potentially be deleted in the data cleaning stage, leading to the risk of ultimately obtaining an insufficient amount of observations. Furthermore, it has been suggested that a balanced data set improves the classification accuracy of machine learning algorithms. More precisely, when there is a minority class in the training data, meaning there are only a few observations belonging to a specific class, a model tends to perform poorly in labeling the observations to this class [62]. Thus, the 45,797 observations are a results of a number of stocks being excluded in order to obtain a more balanced data set. Every 45,797 observations is then labeled, once for the quarterly dataset and one for the yearly dataset, based on the price direction of the following quarter and year. This means that the last observation for the quarterly dataset and the 4 last observations for the yearly dataset for one company was labeled NaN, as the label is given based on the return in the following quarter or year. Those having an excess return compared to the benchmark index are labeled as *Excess* (1) and those having a return under the benchmark index were labeled as *Not Excess* (0). To determine whether the stock yields an excess return it was compared to the MSCI AC Asia ex Japan Small Cap Index during the same period, also obtained from Bloomberg's database.

4.3 Selection of Variables

The initial selection of financial variables was based on the results of previous research on what financial factors affect stock price movements. This is complemented by interviews held with portfolio managers of Handelsbanken Fonder, to get further knowledge of what the main influences on stock returns are, as well as understanding which variables they wish to examine. Predominantly, the factors that are connected to stock prices can be divided into macroeconomic variables, company- and stock-specific variables, investors' sentiment, political events, and other news and events. This study will focus on quantitative factors, namely selected company-specific variables and a few macroeconomic variables. As opposed to other experiments on stock price prediction, this study will also include analysts' sales and stock price (target price) forecast and revisions. The stock market is said to be forward-looking and the expected future sales of companies are commonly used as a component when valuing stocks. Thus, stock price movements and returns may be influenced by analyst sales and target price revisions [63].

Company-specific factors are those related to individual stocks such as financial ratios, liquidity, debt management policies, momentum factors and valuation metrics. These can facilitate understanding the financial stability, the growth outlook, and the future value of a company, and consequently relevant to use when making predictions about whether certain investments will potentially yield high returns or not [26]. Furthermore, macroeconomic factors are included as there have been shown that changes in the overall economy of a country influence the market conditions and thus, also the growth and revenue outlook of companies [64]. The macroeconomic variables chosen in the study are limited to variables that are mainly global or cover the entire region of interest, that is Asia excluding Japan. For instance, 3M interest rate and GDP were found on an Asia ex. Japan basis and was thus included, in contrast to variables such as unemployment rate, which were only found on a national level. However, since money supply has shown to have a significant influence on the Asian stock market [65], the money supply in China was included to serve as an indicator for the entire region. The China money supply is represented by the variable names *Big4banks*, *M1 China* and *M2 China* in table 1. All other variables used in this study are also presented in table 1 and some of the key concepts and variables are further explained in the Financial Background in section 2.

4.4 Preprocessing Data

4.4.1 Data Transformation

Certain variables in this study are naturally expressed in absolute values but can be more appropriate to present in percentage change in order to make observations from a company with a market capitalization of 2 billion USD and another with 100 million USD more comparable. Two such examples of variables that are originally expressed in absolute values are net income and revenue, which are transformed to percentage changes from the last period. Meanwhile, financial ratios such as P/E, EBIT margin, and cash conversion are by definition expressed in relative terms and are comparable across different companies, and can thus be kept in their original form. However, there are some exceptions when transforming the data. For instance, market capitalization is in its original form expressed in absolute values but is both transformed into percentage change to make companies comparable and also kept in its original form (absolute value in USD) to capture company sizes. Moreover, since some data values can go from negative to positive, such as economic value added, percentage change cannot be calculated. For these types of observations the delta (absolute change) is taken instead. The percentage change (%Δ) and delta (Δ) are calculated as follows:

$$\%\Delta = \frac{x_t - x_{t-1}}{x_{t-1}} \times 100, \tag{36}$$

$$\Delta = x_t - x_{t-1}. \tag{37}$$

In table 1 below, a summary is shown. The 'format' column shows if the variables have been transformed into percent/delta change or been kept in their original form, or both. Original means in the format they were originally downloaded from Bloomberg, which can be in USD, percent change from last period, or ratios. When a % is seen in the 'format' column, it means it was transformed from its original form to percental change from last quarter or year, for the quarterly and yearly dataset respectively. Note that T12M is an abbreviation for trailing 12 months which refers to a company's financial data for the previous 12 consecutive months.

Features - Yearly and Quarterly Prediction		
Name	Format	Comment
Market capitalization	Original and %	
Revenue Growth	Original	
Revenue 5y Avg Growth	Original	
Net Income Margin T12M	Original and %	original USD

Net Income Margin (Q)	Original	original
Net Income Growth	Original	
EBIT Margin	Original	
EBIT Growth	Original and %	
Cash Conversion T12M	Original	
Cash Conversion (Q)	Original	
Dividend Payout Ratio	Original	If data not available, replace with 0
EV/EBIT	Original	
P/E	Original	
P/B	Original	
P/S	Original	
Free cash flow (FCF) YIELD	Original	
Return on Equity	Original	
Return on Invested Capital	Original	
Return on Assets	Original	
Debt/Equity	Original	
Current Ratio	Original	
Quick Ratio	Original	
Earnings per Share (EPS) T12M	Original	Sum of the most recent 12 months, four quarters, two semiannuals, or annual earnings per share (EPS)
EPS Growth	Original and %	Percentage increase or decrease of earning before extraordinary items by comparing current period with same period prior year. Calculated as: (EPS before XO Items - EPS before XO Items same period prior year) * 100 / EPS before XO Items from same period prior year/
Est. EPS 12M Forward	Original	
ISM PMI	Original	
M1 China (Money supply)	Original	
M2 China (Money supply)	Original	

3M interest rate Asia ex. Japan	Original	3 month interest rate
GDP Asia ex. Japan	Original and %	
Big4Banks (Money supply)	Original	Balance Sheet of China's 4 largest banks (Commercial Bank of China, China Construction Bank Corp, Agricultural Bank of China, Bank of China LTD)
Revenue T12M	Original and %	
Revenue (Q)	%	
Net Income T12M	%	
Net Income (Q)	%	
Cash from Operations T12M	%	
Cash from Operations (Q)	%	
Estimated sales	%	
Target Price	Delta and %	Delta : share price - target share price. Target price, (fair value) provided by the analyst covering the stock.
Economic Value Added	Original and Delta	Delta : EVA current period - EVA previous period
Revenue T12M	%	

Table 1: Features for Yearly and Quarterly Prediction. Original Format is when no alteration was made. % is when the variable is altered to percentage change from last year or quarter. Delta is when the variable was altered to difference between last year or quarter

4.4.2 Feature Scaling

As a next step in the data transformation process, all features are standardized using the Python package *StandardScaler* from `https://scikit-learn.org` which applies the following formula to each feature vector $j = 1, \dots, p$:

$$x'_{ij} = \frac{x'_{ij} - \bar{x}_j}{\sigma_j}, \;\; i = 1, \dots, n, \tag{38}$$

where x_{ij} is the original value for the i:th observation's j:th feature, $\bar{x}_j$ is the mean of the j:th feature vector and σ_j is the standard deviation of that feature vector.

Feature scaling is crucial in order to allow for comparability across the data set, since the features are often in different scales. In particular, without feature scaling, features with values that are of higher magnitude could dominate the predictions while the features with lower magnitude have less contribution to the response variable. Feature scaling is especially important for distance-based algorithms (such as SVM) and gradient descent-based algorithms (such as logistic regression) [57, p.165].

4.4.3 Data Cleaning

The total number of observations before any type of preprocessing is 45,797. A large number of data points were considered in the first step since many values were missing and had insufficient data, which would mean that a large proportion needed to be dropped. The final datasets used for the quarterly and yearly prediction are described more extensively below.

4.4.4 Data Cleaning for Quarterly Prediction

For the quarterly prediction, each observation has a label based on if the *following* quarter resulted in an excess return. For example, in table 2, 2021 Q1 has the label 1, meaning that in 2020 Q2, the stock beat the benchmark index. The variables that have been transformed into percentage change are however backward-looking and are the percentage change from the *previous* quarter. After this initial preprocessing, all NaN are dropped. This resulted in a total of 4,001 final observations with 46.91% of the labels having excess return, and 53.09% not excess return. The data set was deemed balanced and not in further need of sampling methods to reduce imbalance. As described in table 1, some variables are included in two different formats, for example, absolute value and percentage change. The resulting number of variables was then in total 47. An illustrative example for the quarterly dataset is shown in table 2. The same logic holds for the yearly dataset.

Date	Excess	Market Cap(USD)	Market Cap(%Δ)	P/E	M1 China
2020 Q1	1	100	NaN	3.2	21.2
2020 Q2	0	120	0.20	3.3	23.3
2020 Q3	1	100	-0.17	2.4	25.4
2020 Q4	NaN	90	-0.10	4.6	30.0

Table 2: Illustrative dataset for the quarterly prediction, for *one* company with data only for 2020, after data transformation. Note that the Excess label for the last quarter will be NaN, as well as the first entry for variables that are transformed to percental change from the previous quarter.

The training sample will contain 70% of the observations selected at random, and the fitted model will be tested on the remaining 30% of samples. The three models will be fitted on the training set and evaluated on the test set, and then be subject to hyperparametertuning. All models will after training, testing and hyperparametertuning be statistically evaluated.

4.4.5 Data Cleaning for Yearly Prediction

The proportion of returns in the data that are excess is 43.0%. The dataset is concluded to be balanced enough to rule out class balancing methods. After dropping the NaN values there are a total of 6,490 observations. The response variable for the prediction is binary: 1 for excess and 0 for not excess, and is forward-looking, meaning that for each observation, the algorithm assigns the label 1 or 0 depending on if there was an excess return 12 months *after* the current quarter. For the variables that have been transformed into percentage change, the algorithm compares percent change compared to 12 months *before* the quarter of the observation. There are in total 47 variables.

The training sample will contain 70% of the observations selected at random, and the fitted model will be tested on the remaining 30% of samples. The three models will be fitted on the training set and evaluated on the test set. The models will be fitted with no tuned parameters, and subsequently undergo hyperparametertuning in order to evaluate the improvement in performance.

4.5 Model Development and Evaluation

4.5.1 Model Selection

Since the purpose of this study is to predict future excess returns, the focus shall be on supervised learning techniques. Furthermore, as seen in the literature review, support vector machine is commonly used in stock price direction prediction due to its attractive properties, such as generating a sparse solution that is globally optimal. Additionally, the SVM generalizes well to unseen data, meaning that it avoids overfitting to the training data. The random forest has also proven to perform well and on occasions be the top performer, which makes of interest for this study. Moreover, the logistic regression model is common in most previous research and will be of interest since it is a pure linear discriminant, and hence its performance after hyperparameter tuning can indicate whether the data is linearly separable. Since this research paper will also examine feature importance, we exclude blackbox models such as the neural networks where the structure of the functions cannot be evaluated. In contrast to neural network models, tree-based models and logistic regression provide a visible structure and the feature importance can be obtained with several measures and statistics.

Furthermore, the purpose of this model is not to predict exact returns on a quarterly and yearly horizon and shall focus on stock price direction relative to the benchmark index. Therefore, regression models are excluded since the stock price direction prediction is modeled as a classification problem, where the aim is to predict whether a stock will achieve excess return or not. Excess return can be defined as the difference between the return of a stock and the index return for a specific period. The labels will then be based on if this difference is negative or positive. With the above reasoning, and with support in the Literature Review in section 1.6.1, the focus will be on three models:

- Support vector machine (SVM)
- Logistic regression
- Random forest classifier (RF)

4.5.2 Visualization of Data Using Principal Components

Before fitting the models a graphical representation of the data is conducted using principal components. *Principal Component Analysis* (PCA) is a renowned and widely applied method for reducing the dimensionality in a data set, as it summarizes the information in a large data to a new lower-dimensional data set consisting of the principal components. The principal components are linear combinations of the original variables and are uncorrelated. Furthermore, the principal components are constructed such that the first few ones represent most of the variation in the original data set. Hence, besides being a dimensionality reduction method, PCA can serve as a useful tool for visualizing high-dimensional data in a 2- or 3-dimensional space. This allows for analyzing the characteristics and properties of the data. In this experiment, PCA plots will be used as an aid to view and examine the distribution of the classes, rather than training the models on a PCA-transformed data set [66, pp. 78-79].

4.5.3 Model Fitting and Computational Program

The computational program used to fit the models was exclusively Python. For the support vector machine, the kernel applied yields different possibilities of expanding the feature space. A study on financial data found that the polynomial kernel resulted in slow training and also worse results than for the radial kernel [67]. When a polynomial kernel was applied to the dataset to be evaluated in this study, the Python program ran for over 40 minutes with no result, whereas the radial kernel would yield results in under a minute. Thus this study also excluded the polynomial kernel, and consider the radial basis only.

4.5.4 Cross Validation and Hyperparametertuning

After the models have been fitted, each model performance will be cross validated, with the number of folds being restricted to computational power. Hyperparametertuning will be conducted with built-in functions in Python, with supplementary written code to target specific parameters. Moreover, python has for each of the three models also built-in parameter values with aliases, which were used in this study. The number of features in the models will be denoted p. For logistic regression, there are different solvers, methods, to find the optimal regression parameters described in Section 3.5, for example 'netwton-cg will use the netwon method', lbgfs will use the Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm, and liblinear uses automatic parameter selection. For the SVM 'rbf' is the radial kernel. The gammas can be either 'auto' which will set the value to $\frac{1}{p}$ or 'scale' which sets value to $\frac{1}{p} \times \frac{1}{variance(p)}$. For the random forest, one can set the algorithm to perform bootstrapping by either 'True' or 'False'. The number of features considered per split is set to 'sqrt', meaning $\sqrt{p}$ features are considered per split. For full documentation one can refer to `https://scikit-learn.org`. The cross validation will be used again when finding the optimal hyperparameters in the hyperparameter space for each model, to increase statistical reliability.

4.5.5 Statistical Testing

As a final step, each model will be statistically evaluated in order to determine whether the models actually posses predictive power. Statistical tests are conducted to assure that the models are properly trained with adequate data preprocessing and that their performances statistically can be proven different and also show mathematical rigidity. This will be done by:

- Calculating the standard deviation of recall and precision of the models using cross validation.
- Conducting a hypothesis test to determine if the models are performing significantly differently.
- Constructing a confidence interval for the model errors using cross validation.

5 Results

As described in figure 1, The first step in the analysis is to visualize the variance of the data through PCA to examine the separability of the binary classes. After this step, the models are subject to model fitting and statistical evaluation.

5.1 Quarterly Prediction

5.1.1 Principal Components Analysis

The result of the first 4 PCA components in figure 3, labeled with their respective class, shows little sign of the two classes being clearly separable by the principal components, as bulk data for the classes mainly follow the same pattern. The argument holds when visualizing the first principal component with the second, and the third principal component with the fourth.

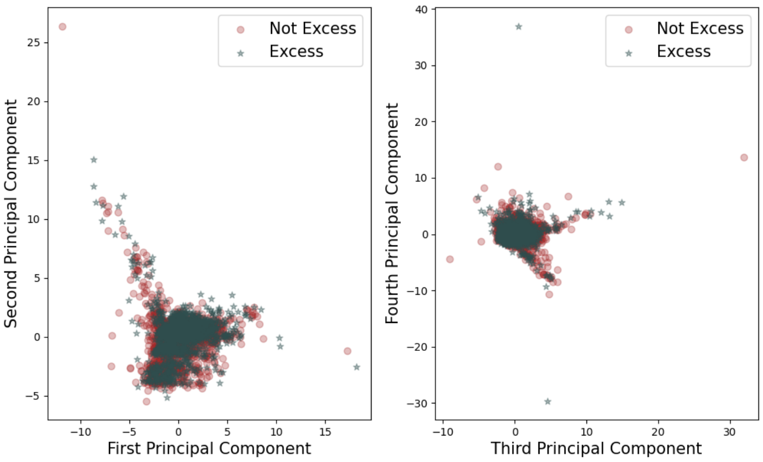
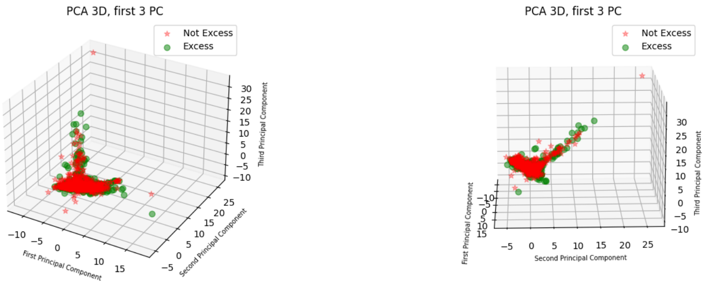


Figure 3: First four Principal components with labeled binary classification



(a) PCA 3D Visual

(b) PCA 3D Visual

Figure 4: 3D Visual of Classes with 3 Largest Principal Components

The 3D visual in figure 4 of the first three principal components further assures that there are no clear signs of the classes being separable in the third dimension when using the components with the largest variances. The findings suggest a linear decision boundary would perform poorly. This initial PCA analysis shows that classes 1 and 0 mostly follow the same distribution in their variance. Below the results from the three fitted models are presented and evaluated in how well they can distinguish between the binary classes.

5.1.2 Support Vector Machine

The SVM was fitted with a radial kernel. The default setting of the tuning parameter C is set to 1. The confusion matrix from this run is presented below in figure 5. The main diagonal shows the correct classifications, and shows that the model correctly classified 482 out of 644 observations as not excess return. The model correctly classified 244 out of 557 observations as having excess returns.

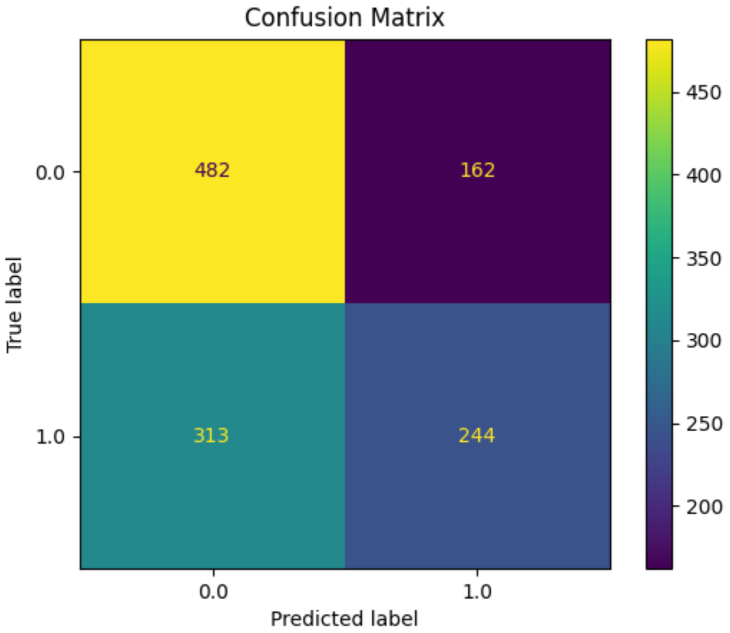


Figure 5: Confusion matrix quarterly prediction SVM

The metrics from this first run are presented in table 4. The initial model has poor recall performance, which means false negative rate is high. Recall ideally should be higher for the purpose of detecting stocks that yield excess return, meaning a high number of true positives and a low number of false negatives. The precision metric does perform better, which also is desirable in order to not label an actual underperforming stock as an excess yielding stock.

The SVM is tuned in order to optimize the performance, and the focus shall be to improve recall. This is done by tuning the hyperparameters in a 5 fold cross validation. Using the *GridSearchCV* package in Python, the hyperparameters are tuned by testing a total of 1×2×10 combinations. The results indicate an optimal performance of the model with parameters set to those seen in table 3.

Parameter	Parameter Space	Optimal Parameter
Kernel	rbf	rbf
gamma	auto, scale	auto
C	1,2,3,4,5,6,7,8,9,10	10

Table 3: Optimal Hyperparameters for SVM, Quarterly Prediction

The optimized model is cross validated 10 folds and presents an increase in recall and F1 score, for a small decrease in accuracy and precision. Since the large increase of recall, the metric of the correctly classified true positives in relation to false negatives, comes from a small decrease in overall performance, the model is deemed as improving its performance from the hyperparametertuning.

Metric	Initial SVM Model	Model with tuned hyperparameter
Accuracy	60.45 %	59.85 %
Precision	60.01 %	59.20 %
Recall	43.81 %	52.00 %
F1	50.57 %	54.30 %

Table 4: Results for SVM metrics before and after hyperparametertuning

Since the classification is occurring in the enlarged feature space with a radial kernel, the feature importance cannot be evaluated.

5.1.3 Random Forest

The next model evaluated was the random forest. Figure 6 is the confusion matrix for the initial run with no parameters tuned.

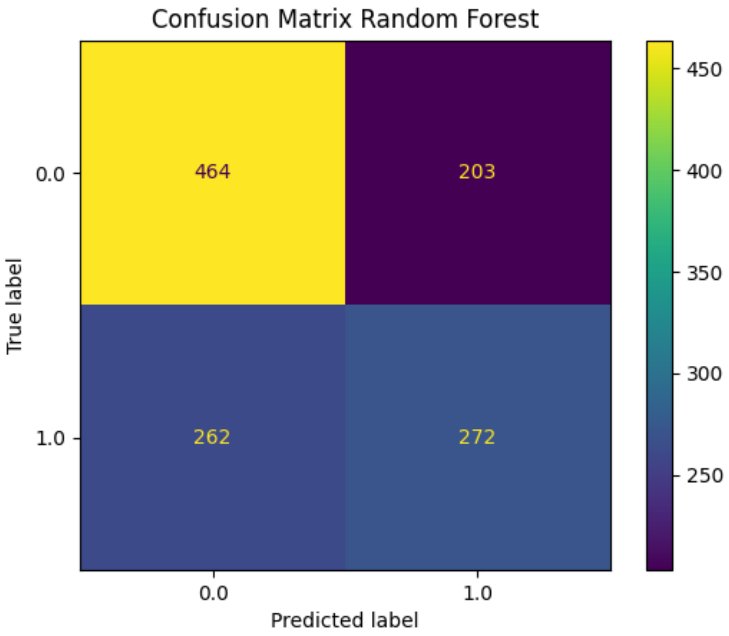


Figure 6: Confusion matrix quarterly prediction random forest

The results indicate an accuracy of 61.64%. The metrics for the classification are presented in table 6.

To optimize the results the hyperparameters are tuned a 3 fold cross validation is conducted 100 times, for 100×2×12×3×3×1 combinations of hyperparameters. The algorithm built runs an exhaustive search to find a model with low variance and improved performance. It shall also test to see if Bagging (bootstrap aggregation) will significantly improve the model. The results from searching the hyperparameter space are seen in table 5.

Parameter	Hyperparameter space	Best hyperparameters
Number of estimators	200 to 2000 with spacing 10	800
Max depth	10 to 120 with spacing 10	50
Minimum samples per split	1,2,10	2
Minimum samples per leaf	1,2,4	2
Bootstrap	True, False	False
Max Features	sqrt	sqrt

Table 5: Hyperparameters for the Random Forest model - Quarterly Prediction

The optimized model is run with a 10 fold CV and presents an accuracy of 62.61%. The classification report of the optimized model is presented below.

Metric	Initial Random Forest Model	Model with tuned hyperparameter
Accuracy	61.64 %	62.62 %
Precision	64.31 %	65.71 %
Recall	53.20 %	62.44 %
F1	55.04 %	61.00 %

Table 6: Results for quarterly prediction with Random Forest metrics before and after hyperparametertuning

The most influential features are extracted from the cross validated optimized model and are seen below in figure 7. The variables are scored through Gini impurity-based feature importance.

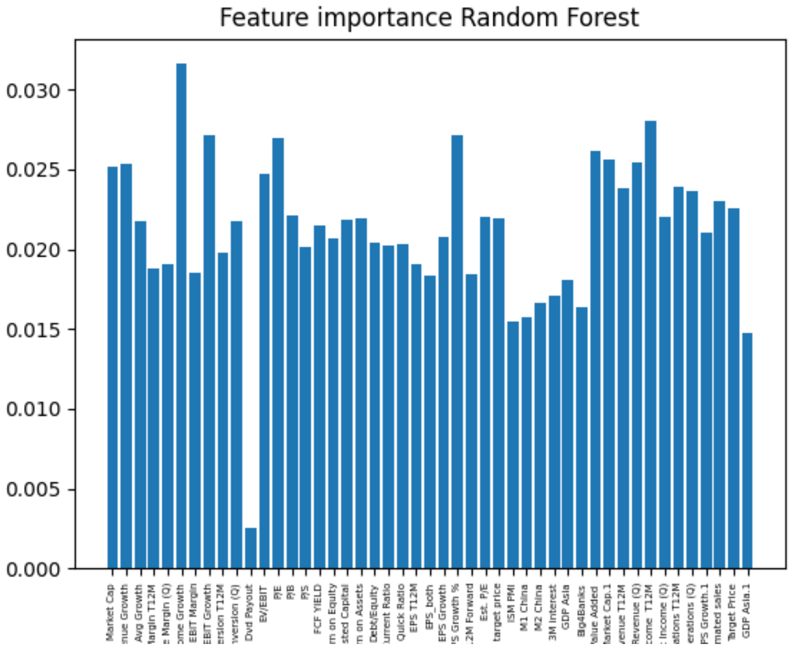


Figure 7: Random forest feature importance based on Gini's Index

Net income growth, net income trailing 12M, EPS growth (%) are found to be the three most influential variables while P/E and dividend payout ratio is presented as the least influential variable. In general, the fundamental variables are seen to be more important to the random forest model than the macro variables.

5.1.4 Logistic Regression

The next model to be presented is the logistic regression. The first run with no tuned hyperparameters presents an accuracy of 56.86%. The model was set to have maximum iterations 100,000 due to it not converging, with one reason possibly being seen from the initial PCA analysis since there seems to be no clear linear decision boundary from the first principal components. The confusion matrix is presented below in figure 8.

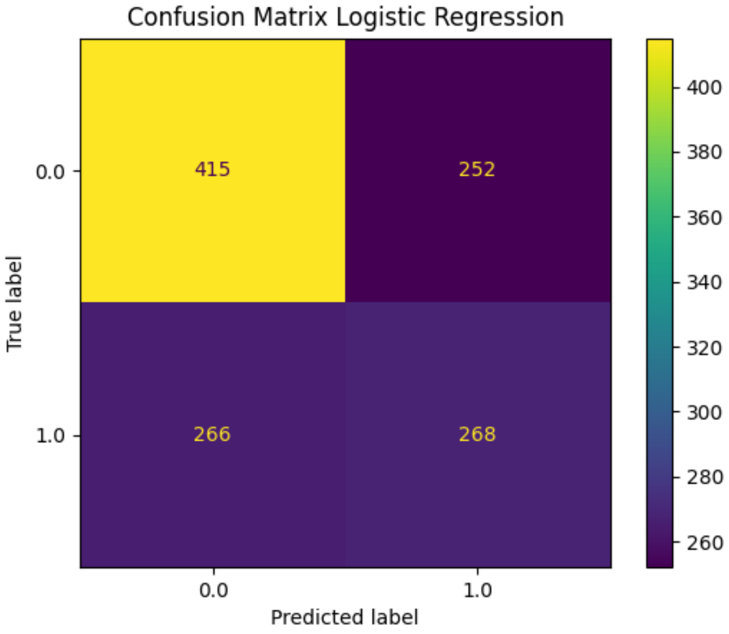


Figure 8: Confusion matrix quarterly prediction logisic regression

The confusion matrix show poor performance in detecting accurate classifications of excess yielding stocks, worse than the previous models tested above.

In order to see if the parameters can be tuned to improve the accuracy score, a 10 fold CV is repeated 3 times with 3×1×5 different combinations of parameters. These are shown in table 7.

Parameter	Hyperparameter Space	Best Hyperparameters
solver	newton-cg, lbfgs, liblinear	newton-cg
penalty	l2	l2
c value	100, 10, 1, 0.1, 0.01	0.1

Table 7: Hyperparameters for the Logistic Regression model - Quarterly Prediction

With these tuned hyperparameters, the results show an accuracy of 56.90%. The metrics scores from the optimized hyperparametertuning show an insignificant improvement from the initial model. Table 8 shows the results for the metrics before and after the tuned and cross validated hyperparameters.

Metric	Initial Logistic Regression Model	Model with Tuned Hyperparameters
Accuracy	56.86 %	56.90 %
Precision	56.40 %	56.70 %
Recall	52.20 %	52.30 %
F1	51.09 %	51.11 %

Table 8: Results for quarterly prediction with Logistic Regression metrics before and after hyperparametertuning

The feature importances are extracted from the optimized logistic regression model and seen below in figure 9.

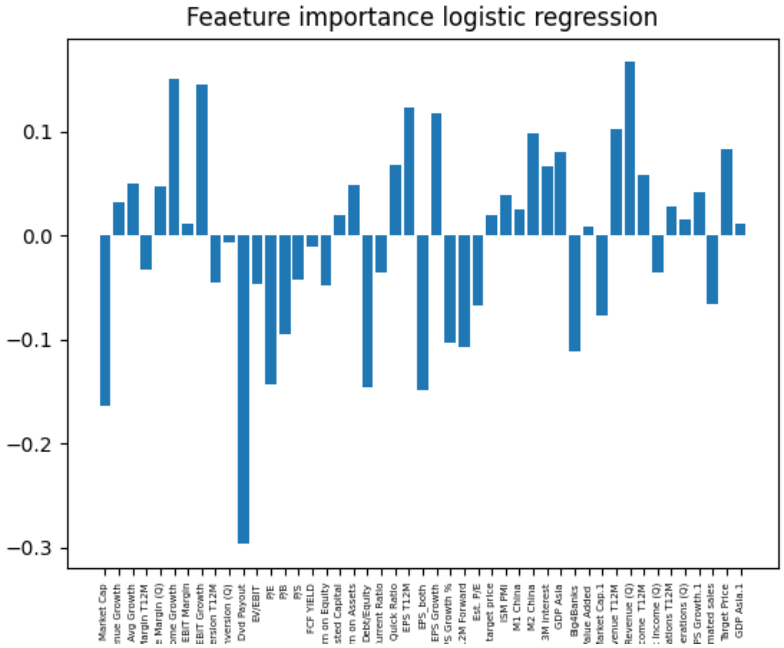


Figure 9: Feature importance for quarterly prediction - logistic regression

To understand if any variables can be concluded statistically significant, the $p - values$ are calculated. The results in table 9 show only two variables have a value below 0.05 a significance level $\alpha = 0.05$.

Feature	$p - value$
Market Cap Absolute	0.000
Revenue (Q)	0.015

Table 9: Statistical Significance of Variables, Significance Level $\alpha = 0.05$ - Logistic Regression Quarterly Prediction

5.1.5 ROC Curve and AUC

All three models are now compared using the receiver operating characteristic curve (ROC curve). The dashed red line indicates the threshold for a model is a random guess. One can see that the random forest model seems to perform best out of the three with an AUC of 0.66. The ROC curves for all classification models are plotted after a 10 fold cross validation on the test set.

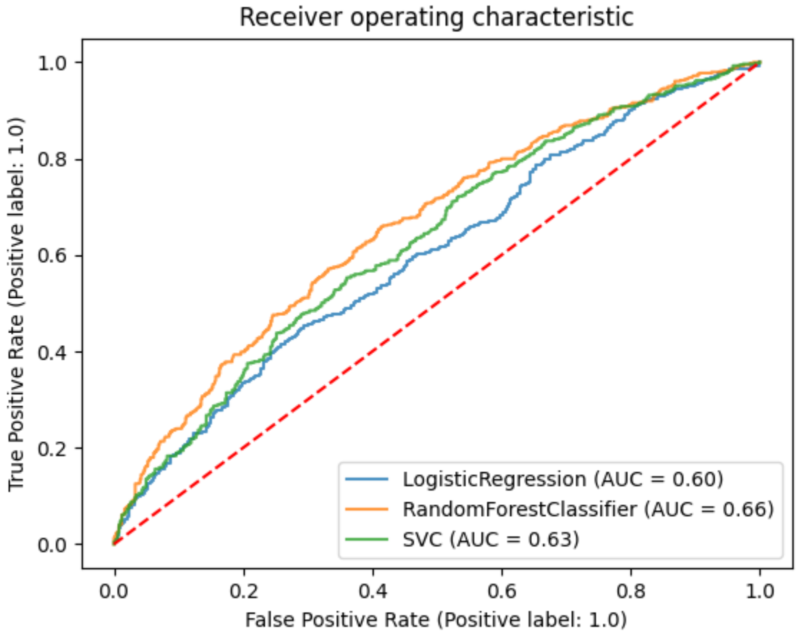


Figure 10: ROC curves for the three optimized models with AUC scores - quarterly prediction

From the ROC curve it is seen that all 3 models possess predictive power as the model's ROC curves lie above the dashed red line which indicates the random guess. The graph shows the tradeoff between the true positives and the false positives. The AUC represents how well the models distinguish between the two classes and is the probability that a randomly chosen positive observation is ranked higher than a randomly chosen negative observation. Since the ROC curves are above the red dashed line and AUC ≥ 0.5 the models are able to distinguish the excess and not excess return. It is however not a fair conclusion to say that the models are performing well, as the ROC curve would ideally be close to the northwest corner of the graph, which means a high rate of true positives for a small rate of false positives.

5.2 Statistical Testing of Models

5.2.1 Standard Deviation and Confidence Interval of Models

The models are run with a 10 fold cross validation where thereafter the standard deviations, σ, and error confidence intervals with $\alpha = 0.05$ are calculated.

Models	σ recall	σ precision	Error Confidence Interval
Random Forest	0.0640	0.0613	0.381 +/- 0.0022
Logistic Regression	0.0621	0.0437	0.425 +/- 0.0015
SVM	0.0579	0.0667	0.417 +/- 0.0020

Table 10: Standard Deviation and 95% Error Confidence Intervals - Quarterly Prediciton

Table 10 demonstrates that the standard deviations for recall and precision is low, indicating that the result is not at random for each iteration. The confidence intervals indicate that the error rates seem to be bound and not at random. This assures that the model has been properly trained.

5.2.2 Kappa Statistic

Model	Value
Random Forest	0.228
SVM	0.174
Logistic Regression	0.127

Table 11: Kappa Statistic for Quarterly Prediction.

The Kappa Statistic indicates that the values are not close to one for the models. However, they assure that the models are trained to predict better than completely at random as they are greater than 0. The metric also indicates that the random forest is the only algorithm with a kappa > 0.2 which indicates fair agreement of predictions. This is in line with what is observed from other statistical metrics, that random forest is performing better than the other models.

5.2.3 Hypothesis Test

To test whether the models are performing different, a hypothesis test is done at a level of $\alpha = 0.05$. The hypothesis are as follows

- H_0 : The performance of the two different models is not significantly different
- H_1 : The performance of the two models is significantly different

The tests are conducted pairwise with and the results are presented in table 12 with the $p-value$, $t-statistics$ for each test conducted.

Models	$p-value$	$t-statistic$	Reject H_0
Random Forest vs. SVM	0.012	4.777	YES
Random Forest vs Logistic Regression	0.009	12.785	YES
SVM vs Logistic Regression	0.267	-1.248	NO

Table 12: Hypothesis Test $\alpha = 0.05$

Concluding from the results above, it is shown that statistical significance between the model performances can be determined when comparing random forest to the other two. A statistically significant difference in performance between the SVM and logistic regression cannot be seen. It can therefore be said that the random forest model is performing best out of the three models in terms of accuracy, precision, recall, AUC, error rate and Kappa Statistic.

5.3 Yearly Prediction

Similar to the previous section, the results in the section begin with a principal component analysis to visualize variance of data. Thereafter the results for the binary models are shown after training, model fitting and analysis, followed by statistical testing.

5.3.1 Principal Components Analysis

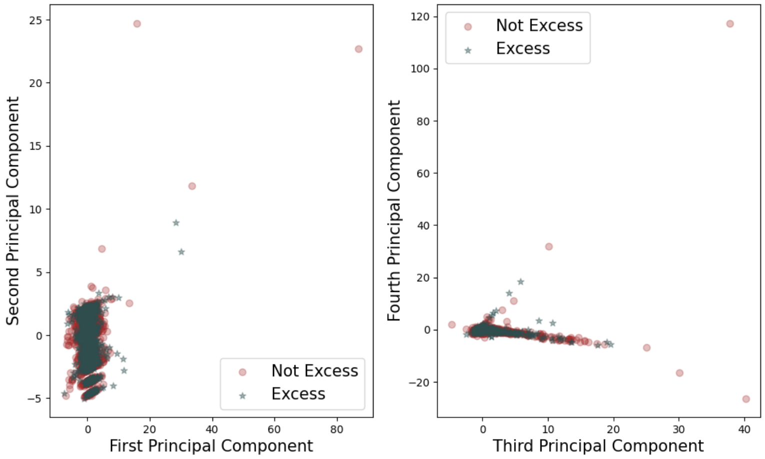


Figure 11: Principal component analysis with labeled binary classes

Initial inspection of a PCA analysis in figure 11 shows that ’not excess’ returns seem to have a higher variance along the axes with the principal component in both figures. The not excess return, in the figure where the first and second principal components are visualized, also shows slightly higher variance along the first principal component. Visualizing the principal components in 3D as seen in figure 12 also shows this pattern.

PCA 3D, first 3 PC
Not Excess
Excess
Third Principal Component
Second Principal Component
First Principal Component
PCA 3D, first 3 PC
Not Excess
Excess
First Principal Component
Second Principal Component
Third Principal Component

(a) PCA 3D Visual

(b) PCA 3D Visual

Figure 12: 3D Visual of Classes with 3 Largest Principal Components

5.3.2 Support Vector Machine

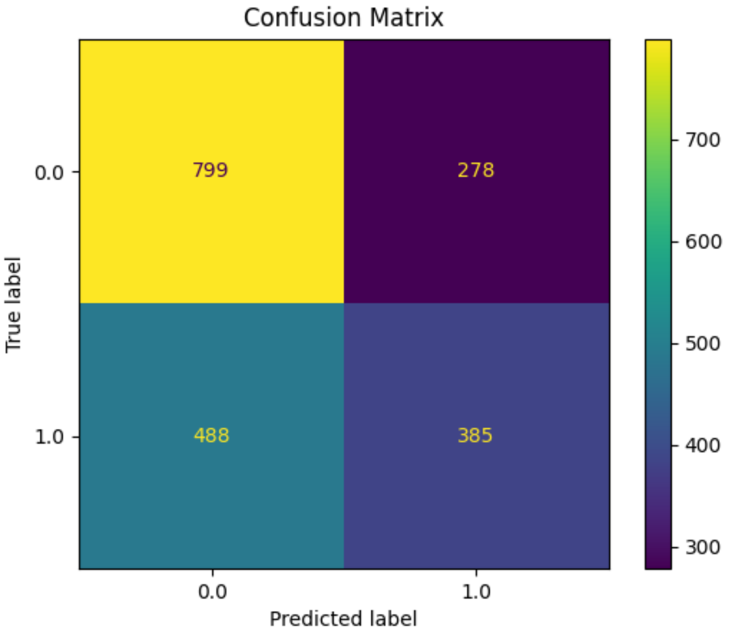


Figure 13: Confusion matrix yearly prediction with SVM

The confusion matrix shown in figure 13 for the initial model with a radial kernel presents an accuracy of 60.82%. The metrics for the run are presented in table 14.

In order to see if the model can be improved the parameters are tuned a 5 fold cross validation is conducted. The performance of the model is optimized for C= 8, gamma = scale. The optimal hyperparameters are stated below in table 13. The performance of the model is improved in regards to accuracy of almost 4%, and the results are seen in table 14.

Parameter	Parameter Space	Optimal Parameter
Kernel	rbf	rbf
gamma	auto, scale	scale
C	1,2,3,4,5,6,7,8,9,10	8

Table 13: Optimal Hyperparameters for SVM, Yearly Prediction

Metric	Initial SVM Model	Model with Tuned Hyperparameters
Accuracy	60.82 %	64.30 %
Precision	58.20 %	63.10 %
Recall	44.33 %	53.00 %
F1	50.33 %	64.91 %

Table 14: Results for yearly prediction with SVM before and after hyperparametertuning

5.3.3 Random Forest

The confusion matrix of the initial run with the random forest algorithm is presented in figure 14. The model performs well in classifying those observations that do not have excess return, but performs less accurately when predicting those that have excess return.

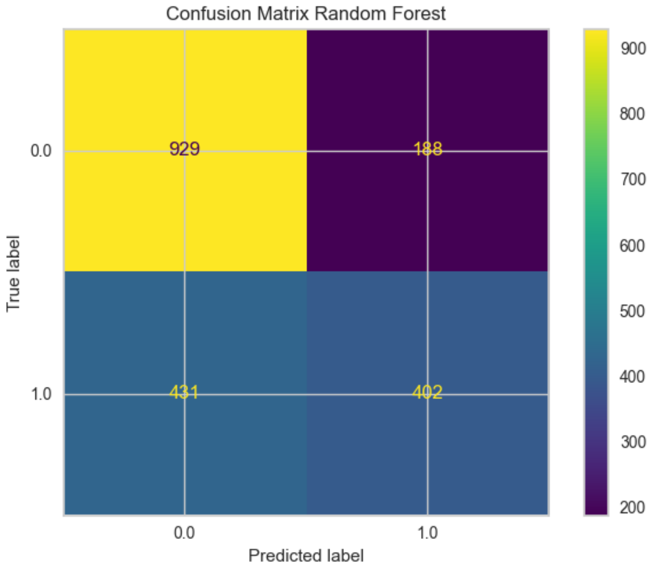


Figure 14: Confusion matrix yearly prediction with random forest

The initial model with no tuned parameters presents an accuracy of 68.26%. The classification report is presented below in table 16.

In order to see if the performance can be improved the hyperparameters are tuned with a 3 fold cross validation due to the large complexity of combinations. To optimize the results the hyperparameters are tuned with a 3 fold cross validation, conducted 100 times, for 100×2×10×3×3×1 combinations of hyperparameters. These combinations will make up the hyperparameter space in which the best combinations will be found. Table 15 presents the results.

Parameter	Hyperparameter Space	Best Hyperparameters
Number of estimators	200 to 2000 with spacing 10	1000
Max depth	10 to 120 with spacing 10	80
Minimum samples per split	1,2,10	10
Minimum samples per leaf	1,2,4	1
Bootstrap	True, False	False
Max Features	sqrt	sqrt

Table 15: Hyperparameters for the Random Forest model - Yearly Prediction

The accuracy of the optimized model is 69.64% with the metrics presented below.

Metric	Initial Random Forest Model	Model with Tuned Hyperparameters
Accuracy	68.30 %	69.64 %
Precision	68.10 %	69.20 %
Recall	66.20 %	68.10 %
F1	59.90 %	68.00 %

Table 16: Results for yearly prediction with Random Forest before and after hyperparametertuning

The optimized model's feature importance is examined and show in figure 15. The random forest model recognizes the market capitalization, Revenue 5y Avg Growth, and EPS trailing 12 months as the top 3 influential variables, closely followed by Debt/Equity ratio. Dividend payout is the least influential variable.

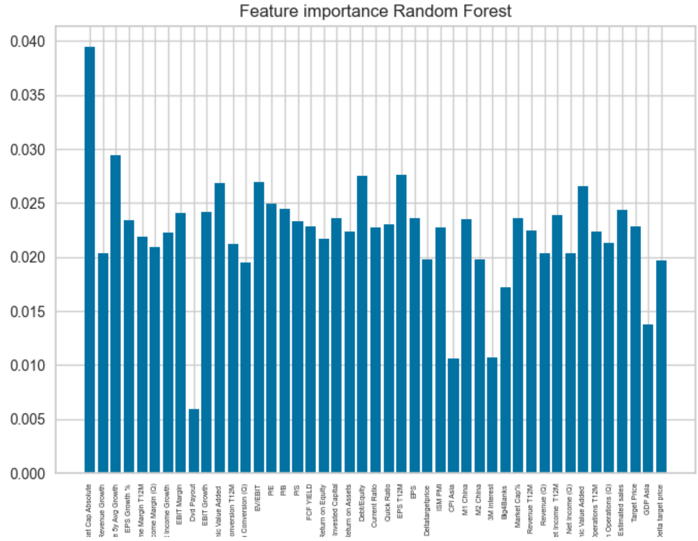


Figure 15: Random forest feature importance - optimized model

5.3.4 Logistic Regression

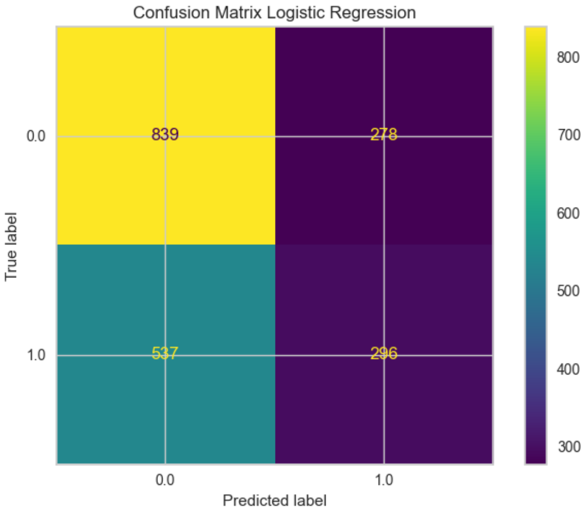


Figure 16: Confusion matrix logistic regression

Logistic regression presents an accuracy of 58.50%. The model performs well under the previously presented models. However, a hyperparameter tuning is conducted to aid in assessing the overall quality of data preprocessing. The hyperparameters tuned are the same as for the quarterly prediction and the optimized model parameters are presented in table 17.

Parameter	Hyperparameter Space	Best Hyperparameters
solver	newton-cg, lbfgs, liblinear	liblinear
penalty	l2	l2
c value	100, 10, 1, 0.1, 0.01	0.1

Table 17: Hyperparameters for the Logistic Regression model - Yearly Prediction

The optimized results indicate an accuracy of 58.51% when cross validated, and 60.93% as the highest score. The metrics for the tuned model are shown in table 18.

Metric	Initial Logistic Regression Model	Model with Tuned Hyperparameters
Accuracy	58.50 %	58.51 %
Precision	51.84 %	52.12 %
Recall	35.53 %	35.30 %
F1	42.17 %	42.10 %

Table 18: Results for yearly prediction with Logistic Regression before and after hyperparameter-tuning

The feature importances from the optimized model are found, after a 10 fold cross validation, as seen in figure 17.

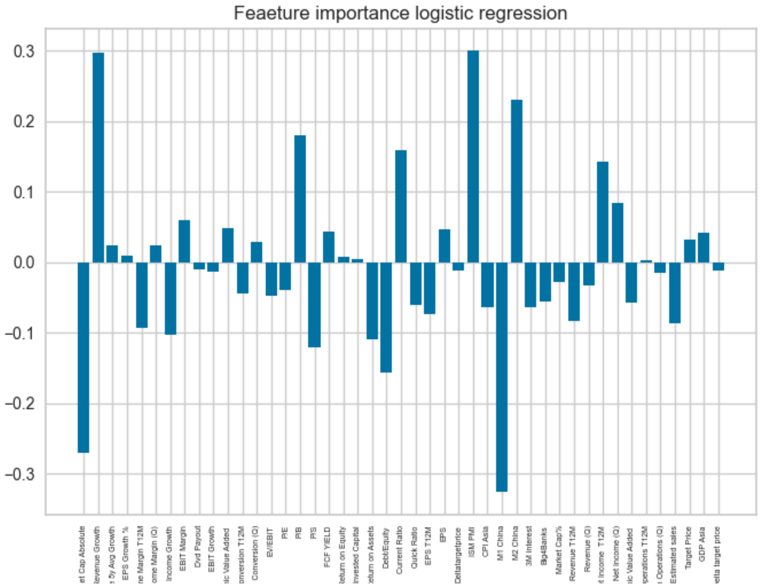


Figure 17: logistic regression feature importance

The performance of the logistic model has so far proven worse than the other two. A statistical test for the features shows which ones significantly contribute to the model. The significance test was done with significance level $\alpha = 0.05$. The significant variables are shown in table 19.

Feature	$p - value$
Market Cap Absolute	0.000
Revenue Growth	0.032
P/B	0.073
P/S	0.005
ISM PMI	0.000
M1 China	0.000
Revenue T12M	0.022
Net Income T12M	0.045

Table 19: Significalt variables - Logistic Regression Yearly Prediction with significance level $\alpha = 0.05$.

5.3.5 ROC Curve and AUC

The ROC curves of the three optimized models are all plotted after a 10 fold cross validation on the test set, and it is seen that the random forest model is outperforming the other two with an AUC score of 0.77. All of the curves are however above the dashed red line which indicates they perform better than a random guess. The ROC curve also shows that the random forest model yields a true positive rate (TPR) of around 0.6, for a false positive rate (FPR) of 0.2, while logistic regression yields around 0.3 TPR for a 0.2 FPR. This initial analysis shows that the random forest classifier is performing best.

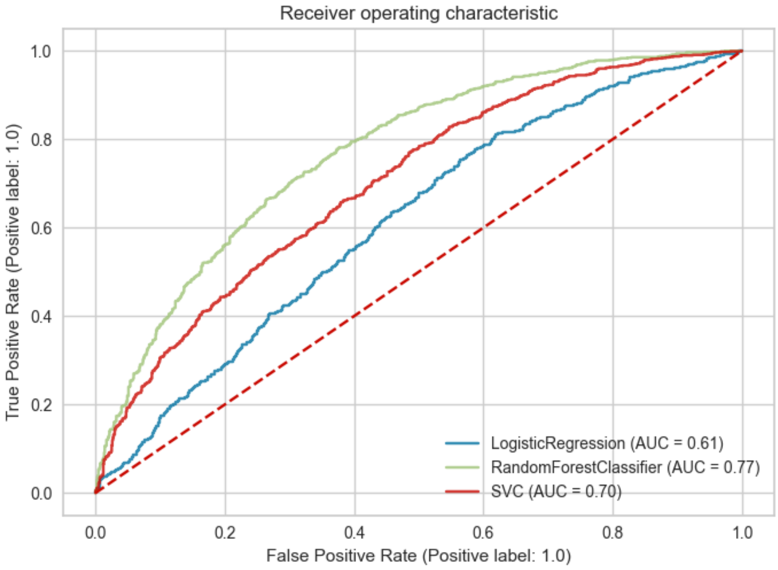


Figure 18: ROC curves for the optimized models wih AUC scores

5.4 Statistical Testing of Models

5.4.1 Standard Deviation and Confidence Interval of Models

The confidence intervals of the mean errors are constructed through a 10 fold cross validation, with $\alpha = 0.05$. The 95% confidence intervals and standard deviation for the models are presented in table 20.

Models	σ recall	σ precision	Error Confidence Interval
Random Forest	0.05950	0.03950	0.3359 +/- 0.0010
Logistic Regression	0.05126	0.05269	0.4123 +/- 0.0013
SVM	0.04510	0.05308	0.3713 +/- 0.0016

Table 20: Standard Deviation and Error Confidence Intervals - Yearly Prediciton

All three models present a bound error rate and also a low standard deviation, which assures that the models have been properly trained and do not produce metrics at random.

5.4.2 Kappa Statistic

Model	Value
Random Forest	0.363
Support Vector Machine	0.277
Logistic Regression	0.127

Table 21: Kappa Statistic for the 3 models - Yearly Prediction

The Kappa Statistic shows improvement compared to the quarterly prediction for random forest and support vector machine as indicated by the metrics, and an insignificant change for logistic regression. The Kappa Statistic reassures that the yearly prediction algorithms do seem to exhibit more predictive power compared to the quarterly prediction and that the SVM and random forest present fair agreement in predictions.

5.4.3 Hypothesis Test

To assess the statistical quality of the models, a hypothesis test at level $\alpha = 0.05$ is done to examine of the models are performing differently with statistical significance. The tested hypothesis are

- H_0 : The performance of the two different models is not significantly different
- H_1 : The performance of the two models is significantly different

Models	$p - value$	$t - statistic$	Reject H_0
Random Forest vs. SVM	0.003	8.786	YES
Random Forest vs Logistic Regression	0.001	7.025	YES
SVM vs Logistic Regression	0.004	-5.046	YES

Table 22: Hypothesis Test $\alpha = 0.05$

It can be shown that all three models are performing statistically different at significance level $\alpha = 0.05$. This implies that the random forest model is the best performing model.

6 Discussion

This study was novel in two ways, that a majority of research papers focus on mid to large-cap stocks, and that the geographical area was covering for the prediction was multiple countries in the Asian region rather than one. The results have been presented for yearly and quarterly prediction, for the binary case of the stock yielding an excess return or not compared to the index MSCI AC Asia EX Japan. The models have been tested in multiple different aspects in order to determine their statistical significance. None of the models show high variance or a tendency to perform well at random, which is further assured through cross validation. The models have also been evaluated with multiple metrics, some being accuracy, precision, recall, and F1. None of the metrics show high variance and a 95% confidence interval could be constructed.

The best model performing model among the quarterly and yearly prediction is in both the yearly and quarterly case the random forest model, followed by the support vector machine. The random forest prediction on a yearly basis shows an accuracy of 69.64% after hyperparametertuning and cross validation. With the dataset for this study and the subsequent findings, it can be concluded that the models perform better when predicting the small- and micro-cap stocks on a yearly time horizon compared than a quarterly, possibly because the bulk of the data for the quarterly dataset shows more variance in the PCA analysis compared to the yearly dataset, which can increase the difficulty in predicting the outcome. These results might also indicate that the chosen variables are more suited for a yearly prediction horizon. Similar conclusions are drawn when predicting one-year stock movement (up, down) in the research article by Ballings et. al (2015), where the random forest model was the top performer, followed by the support vector machine [29]. In line with their conclusion, the findings from here would also support that the SVM and random forest model are adequate candidates when predicting stock price direction.

From this study, it can also be concluded that the models perform differently in all cases except in the quarterly case between SVM and logistic regression, and that the models show room for improvement when tuning the hyperparameters. For the quarterly prediction, it could be stated that the random forest model performed better, but that the ROC curve still shows the tendency of a close relationship between TPR and FPR tradeoff. The collected data for this research purpose indeed had a high variance, also in line with a general characteristic for micro- and small-cap stocks. The maximum excess return of all observations in the quarterly observations was 346.9%, and the minimum excess return was -53.9% and for the yearly observations the numbers were 954.5% and -976.5% respectively. The abnormal behaviour of these stocks can have contributed to the algorithms having difficulty in finding patterns for classifying the data.

The random forest and logistic regression model also present which features are the most influential and significant. However, the poor performance from the logistic regression gives little incentive to draw general conclusions. For quarterly prediction, the random forest places the top three influential variables as net income growth, net income trailing 12M, and EPS growth in percent, whereas the logistic regression model shows no statistical significance for these variables at a significance level of $\alpha = 0.05$. For the yearly prediction models, the random forest and logistic regression model align in placing market capitalization in terms of USD value as one of the most significant variables. The random forest model further identifies EPS trailing 12M as an influential variable, which aligns with previous studies by Anwaar, M and Emamgholipour [43][42]. However, these studies were seen to present conflicting results in whether it has a positive or negative impact. Debt/Equity is also found by the random forest to be an influential variable, which Hobarth, L also concluded in his research [45]. The feature importances highlighted in the results show some alignment with previous research, and limited similarities between the logistic and random forest model, which makes it desirable to investigate the feature importance further in order to draw a conclusion. Suggestions will be discussed in the final section of this study.

7 Conclusion

Three different models were tested on a binary classification problem for stock price direction of Asian small- and micro-cap stocks. The random forest model performs superior in both the quarterly and yearly prediction and it is seen that there is a statistical significance in the performance difference compared to the logistic model and support vector machine. The statistical testing of the models in all cases assures that the data has been trained in a manner to reduce model variance. This is promising as it indicates that machine learning models can be appropriate for this type of dataset. We have also seen that all three models show room for improvement in their statistical performance, which suggests that the performance possibly further can be enhanced with the use of different variables or preprocessing techniques that were out of the scope for this research. This study also concludes that the models in general perform better on a yearly prediction horizon compared to a quarterly, which is also suggested by the visual results of the PCA plots where the quarterly dataset shows a greater variance in the bulk of the data. The conclusions for the feature importance are deemed inconclusive but show some similarities with previous research done. However, since the performance of the yearly random forest model is found the best among the three considered, the results from the three most influential variables can indicate that market capitalization, revenue 5 year average growth and EPS trailing 12M are interesting to investigate further. With the results presented above, the answers to the research questions can be presented as follows:

- **RQ 1** The best model among the three for the dataset in this study, after hyperparametertuning and cross validation, was found to be the random forest model for a yearly price direction prediction, with an accuracy of 69.64%. Random forest was found best also for quarterly prediction, with an accuracy of 62.62%.
- **RQ 2** The results were inconclusive, as the logistic regression and random forest model do not align in identifying the influential variables. However, as the random forest model presents the best performance for yearly prediction, its results after cross validation and hyperparametertuning indicate that market size (USD), revenue 5 year average growth (%) and EPS trailing 12M can be candidates for the three most influential variables, but are subject to further testing and investigation.

8 Future Research

When calculating if the observation yields an excess return or not, this study compares for the quarterly prediction the next quarter on the last day, and for the yearly prediction the last day of the fourth following quarter. For future purposes, to reduce the impact of day-specific events, it can be of interest to instead calculate the average return for the last week of the quarter.

Another topic neglected in this study is look-ahead bias. This study assumes that all information is available on the last day of the quarter, when there actually could have been a delay in when this information was available. This could potentially lead to overconfidence in the prediction results. Some studies have been carried out to evaluate the effect look ahead bias has on performance. Jenke, R et. al use a Monte Carlo study to eliminate look-ahead bias [68]. Baquero, G et al. also suggest weighing procedures in order to account for the look-ahead bias, where they found around 3.8% overestimation when evaluating hedge fund performance with time series data [69].

As the purpose of this study furthermore was not to assess predictability for different or specific company sectors, it can for the future also be relevant if the model is improved when excluding certain sectors or limiting the observations to only one sector. For example, some studies have considered only predicting banking and financial sectors such as one by Vikalp, R et al. [25].

Another suggestion for future research on small- and micro-cap stocks can be to remove stocks that have abnormally high or low excess returns. Removing extreme values or outliers has been seen to improve results in classification and regression techniques, although it cannot be generalized as common practice for all datasets [57].

References

[1] Burton G Malkiel and Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

[2] Stephen F LeRoy and Richard D Porter. The present-value relation: Tests based on implied variance bounds. *Econometrica: Journal of the Econometric Society*, pages 555–574, 1981.

[3] Burton G Malkiel. The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1):59–82, 2003.

[4] Manish Kumar and M Thenmozhi. Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*, 2006.

[5] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. Stock price prediction using the arima model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pages 106–112. IEEE, 2014.

[6] Richard Mathieson Jeff Shen, Raffaele Savi. New technologies changing asset management. *BlackRock*, 2020.

[7] JG Agrawal, V Chourasia, and A Mittra. State-of-the-art in stock prediction techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(4):1360–1366, 2013.

[8] Lorne N Switzer. The behaviour of small cap vs. large cap stocks in recessions and recoveries: Empirical evidence for the united states and canada. *The North American Journal of Economics and Finance*, 21(3):332–346, 2010.

[9] Penglei Gao, Rui Zhang, and Xi Yang. The application of stock index price prediction with neural network. *Mathematical and Computational Applications*, 25(3):53, 2020.

[10] Tim Bollerslev, Ray Y Chou, and Kenneth F Kroner. Arch modeling in finance: A review of the theory and empirical evidence. *Journal of econometrics*, 52(1-2):5–59, 1992.

[11] M Hiransha, E Ab Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Nse stock market prediction using deep-learning models. *Procedia computer science*, 132:1351–1362, 2018.

[12] Vatsal H Shah. Machine learning techniques for stock prediction. *Foundations of Machine Learning— Spring*, 1(1):6–12, 2007.

[13] Bruce L Bowerman and Richard T O'Connell. *Time series and forecasting.* Duxbury Press North Scituate, MA, 1979.

[14] Prapanna Mondal, Labani Shit, and Saptarsi Goswami. Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2):13, 2014.

[15] Basel MA Awartani and Valentina Corradi. Predicting the volatility of the s&p-500 stock index via garch models: the role of asymmetries. *International Journal of Forecasting*, 21(1):167–183, 2005.

[16] Him Tang, Kai-Chun Chiu, and Lei Xu. Finite mixture of arma-garch model for stock price prediction. In *Proceedings of the Third International Workshop on Computational Intelligence in Economics and Finance (CIEF'2003), North Carolina, USA*, pages 1112–1119, 2003.

[17] Aidan Meyler, Geoff Kenny, and Terry Quinn. Forecasting irish inflation using arima models. 1998.

[18] Chih-Fong Tsai and Yu-Chieh Hsiao. Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, 50(1):258–269, 2010.

[19] Ruo-Nan Duan, Jia-Yi Zhu, and Bao-Liang Lu. Differential entropy feature for eeg-based emotion classification. In *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 81–84. IEEE, 2013.

[20] Erkam Guresen, Gulgun Kayakutlu, and Tugrul U Daim. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, 38(8):10389–10397, 2011.

[21] M Hiransha, E Ab Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Nse stock market prediction using deep-learning models. *Procedia computer science*, 132:1351–1362, 2018.

[22] Xin-Yao Qian and Shan Gao. Financial series prediction: Comparison between precision of time series models and machine learning methods. *arXiv preprint arXiv:1706.00948*, pages 1–9, 2017.

[23] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[24] Phichhang Ou and Hengshan Wang. Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, 3(12):28–42, 2009.

[25] Vikalp Ravi Jain, Manisha Gupta, and Raj Mohan Singh. Analysis and prediction of individual stock prices of financial sector companies in nifty50. *International Journal of Information Engineering and Electronic Business*, 11(2):33, 2018.

[26] Mingyue Qiu, Yu Song, and Fumio Akagi. Application of artificial neural network for the prediction of stock market returns: The case of the japanese stock market. *Chaos, Solitons & Fractals*, 85:1–7, 2016.

[27] Francis EH Tay and Lijuan Cao. Application of support vector machines in financial time series forecasting. *omega*, 29(4):309–317, 2001.

[28] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

[29] Michel Ballings, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. Evaluating multiple classifiers for stock price direction prediction. *Expert systems with Applications*, 42(20):7046–7056, 2015.

[30] Chih-Fong Tsai, Yuah-Chiao Lin, David C Yen, and Yan-Min Chen. Predicting stock returns by classifier ensembles. *Applied Soft Computing*, 11(2):2452–2459, 2011.

[31] Jigar Patel, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, 42(1):259–268, 2015.

[32] Nikola Milosevic. Equity forecast: Predicting long term stock price movement using machine learning. *arXiv preprint arXiv:1603.00751*, 2016.

[33] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers & operations research*, 32(10):2513–2522, 2005.

[34] Yanshan Wang. Stock price direction prediction by directly using prices data: an empirical study on the kospi and hsi. *International Journal of Business Intelligence and Data Mining*, 9(2):145–160, 2014.

[35] Zuherman Rustam, DF Vibranti, and Dhian Widya. Predicting the direction of indonesian stock price movement using support vector machines and fuzzy kernel c-means. In *AIP Conference Proceedings*, volume 2023, page 020208. AIP Publishing LLC, 2018.

[36] Mohsen Behzad, Keyvan Asghari, Morteza Eazi, and Maziar Palhang. Generalization performance of support vector machines and neural networks in runoff modeling. *Expert Systems with applications*, 36(4):7624–7629, 2009.

[37] Li-Juan Cao and Francis Eng Hock Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks*, 14(6):1506–1518, 2003.

[38] Marco Gori and Alberto Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.

[39] Christopher JC Burges, David J Crisp, et al. Uniqueness of the svm solution. In *NIPS*, volume 99, pages 223–229. Citeseer, 1999.

[40] Kevin Greenidge and Tiffany Grosvenor. Forecasting non-performing loans in barbados. *Journal of Business, Finance & Economics in Emerging Economies*, 5(1), 2010.

[41] Jie Sun and Hui Li. Data mining method for listed companies' financial distress prediction. *Knowledge-Based Systems*, 21(1):1–5, 2008.

[42] Maryyam Anwaar. Impact of firms performance on stock returns (evidence from listed companies of ftse-100 index london, uk). *Global Journal of Management and Business Research*, 2016.

[43] Milad Emamgholipour, Abbasali Pouraghajan, Naser Ail Yadollahzadeh Tabari, Milad Haghparast, and Ali Akbar Alizadeh Shirsavar. The effects of performance evaluation market ratios on the stock return: Evidence from the tehran stock exchange. *International Research Journal of Applied and Basic Sciences*, 4(3):696–703, 2013.

[44] Hakkı Öztürk and Tolun A Karabulut. The relationship between earnings-to-price, current ratio, profit margin and return: an empirical analysis on istanbul stock exchange. *Accounting and Finance Research*, 7(1):109–115, 2018.

[45] Lukas Lorenz Höbarth. *Modeling the relationship between financial indicators and company performance. An empirical study for US-listed companies.* PhD thesis, WU Vienna University of Economics and Business, 2006.

[46] Tong Yao, Tong Yu, Ting Zhang, and Shaw Chen. Asset growth and stock returns: Evidence from asian financial markets. *Pacific-Basin Finance Journal*, 19(1):115–139, 2011.

[47] Dwi Martani and Rahfiani Khairurizka. The effect of financial ratios, firm size, and cash flow from operating activities in the interim report to the stock return. *Chinese Business Review*, 8(6):44, 2009.

[48] Sina Kheradyar, Izani Ibrahim, and F Mat Nor. Stock return predictability with financial ratios. *International Journal of Trade, Economics and Finance*, 2(5):391, 2011.

[49] Kmonwan Chairakwattana, Sarayut Nathaphan, et al. *Stock Return Predictability by Bayesian Model Averaging: Evidence from Stock Exchange of Thailand.* Faculty of Commerce and Accountancy, Thammasat University, 2013.

[50] LAI Ping-fu Brian-CHO Kwai-yee. Relationships between stock returns and corporate financial ratios based on a statistical analysis of corporate data from the hong kong stock market. *Public Finance Quarterly*, 1:111, 2016.

[51] Dursun Delen, Cemil Kuzey, and Ali Uyar. Measuring firm performance using financial ratios: A decision tree approach. *Expert systems with applications*, 40(10):3970–3983, 2013.

[52] Luis Amézola Berenguer. A 5-factor risk model for european stocks. Master's thesis, Universitat Politècnica de Catalunya, 2017.

[53] Jonathon Berk, Peter DeMarzo, Jarrod Harford, Guy Ford, Vito Mollica, and Nigel Finch. *Fundamentals of corporate finance*. Pearson Higher Education AU, 2013.

[54] Mikael Sens. Interview, Handelsbanken Asset Management. Stockholm, Sweden, 2021.

[55] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[56] H Jabbar and Rafiqul Zaman Khan. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). *Computer Science, Communication and Instrumentation Devices*, pages 163–172, 2015.

[57] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[58] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[59] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[60] Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363, 2005.

[61] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[62] Vaishali Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47, 2012.

[63] Wayne A. Thorp. How to Profit From Revisions in Analysts' Earnings Estimates. `https://www.aaii.com/journal/article/how-to-profit-from-revisions-in-analysts-earnings-estimates?`, 2005. [Online; accessed 12-02-2021].

[64] Raghutla CHANDRASHEKAR, P Sakthivel, T Sampath, and Krishna Reddy CHITTEDI. Macroeconomic variables and stock prices in emerging economies: A panel analysis. *Theoretical & Applied Economics*, 25(3), 2018.

[65] Praphan Wongbangpo and Subhash C Sharma. Stock market and macroeconomic fundamental dynamic interactions: Asean-5 countries. *Journal of asian Economics*, 13(1):27–51, 2002.

[66] Ian T Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

[67] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

[68] Jenke R Ter Horst, Theo E Nijman, and Marno Verbeek. Eliminating look-ahead bias in evaluating persistence in mutual fund performance. *Journal of Empirical Finance*, 8(4):345–373, 2001.

[69] Guillermo Baquero, Jenke ter Horst, and Marno Verbeek. Survival, look-ahead bias, and persistence in hedge fund performance. *Journal of Financial and Quantitative analysis*, pages 493–517, 2005.

TRITA-SCI-GRU 2021:9623

www.kth.se