UNIVERSITY OF TURKU

ABSTRACT

□ Bachelor´s thesis
☒ Master´s thesis
□ Licentiate´s thesis
□ Doctor´s thesis

Subject	Accounting and Finance	Date	9.5.2019
Author(s)	Tuomas Patrikainen	Student number	509957
		Number of pages	102
Title	Using machine learning to forecast long-term equity price movement		
Supervisor(s)	Ph.D. Jan Pfister		

Abstract

Predicting equity price movement is one of the fundamental challenges in finance, and even small improvements in prediction performance can be highly profitable for investors. Long-term investment is one of the popular investment strategies that investors follow. However, evaluating which companies are going to perform well in the future is difficult. This research presents machine learning aided approach to forecast long-term price movement of the stocks listed on the Helsinki Stock Exchange.

The purpose of the research is to find out which machine learning model performs the best in the Finnish financial markets and to understand what the key variables are, which have a major effect on the prediction accuracy of the models. The research is also testing whether the macroeconomic variables of Finland increase the accuracy of the machine learning models when forecasting long-term equity price movement. The following machine learning models are used in the research: logistic regression, support vector machine, decision tree, random forest, and k-nearest neighbors.

This research produced a number of key findings: the results from the models indicated that the best performance was achieved by the random forest model, which obtained classification accuracy of 65.3% and F1 score of 60.8%; the random forest model is able to give over 60% chance for an investor to pick a stock, which will have a 10% or higher return over the period of one year; the macroeconomic variables increased the prediction performance of every machine learning model used in the research.

The main conclusions drawn from this research are that the macroeconomic variables can provide new information, which is not explained by only using financial ratios in the models. Also, the equity prices in the Finnish financial markets are not equally random, meaning that they do not always follow a random walk process. Therefore, this research argues that the Finnish financial market is not highly efficient, thus stock prices are on some level predictable. These findings contribute to the financial theory of market efficiency.

Key words	Machine learning, supervised learning, equity price prediction, long-term investment
Further information	

UNIVERSITY
OF TURKU
Turku School of
Economics

USING MACHINE LEARNING TO FORECAST LONG-TERM EQUITY PRICE MOVEMENT

An empirical study of the Finnish financial markets

Master´s Thesis
in Accounting and Finance

Author:
Tuomas Patrikainen 509957

Supervisor:
Ph.D. Jan Pfister

9.5.2019
Turku

Table of Contents

List of figures

List of tables

LIST OF ABBREVIATIONS AND SYMBOLS

AMH	Adaptive market hypothesis
ANN	Artificial neural network
AUC	Area under the ROC curve
CPS	Close price of a stock
DPS	Dividend per share
DY	Dividend yield
EBIT	Earnings before interest and taxes
EMH	Efficient market hypothesis
EPS	Earnings per share
EXPINF6M	Expected inflation for the next 6 months
GDP	Gross domestic product
KNN	K-nearest neighbors
MC	Market cap
MLE	Maximum likelihood estimation
OMXH	OMX Helsinki Stock Exchange
PBR	Price-to-book ratio
PER	Price-earnings ratio
QP	Quadratic programming
QR	Quick ratio
ROC	Receiver operating characteristics
ROI	Return on invested capital
SVM	Support vector machine
TDTC	Total debt/total capital

1 INTRODUCTION

1.1 Background and motivation

Predicting stock market index or a stock price movement is a challenging problem in finance because a stock market is highly dynamic system. According to the efficient market hypothesis it is not possible to exactly predict the stock prices of companies because of random walk behavior in stock markets. The efficient market hypothesis argues that all the available information is continuously interpreted by the markets and instantly included in asset prices. If the markets are efficient and market prices only react quickly to new information, investors cannot make constantly risk-adjusted excess returns (Fama 1970).

Long-term investing is a popular investment strategy used by individual as well as institutional investors. However, evaluating which companies are going to perform well in the following months or years is challenging and predicting stock prices is certainly difficult. For any investor, especially for a value investor, it would be beneficial to know which way the stock prices will be developing in the near future. Investors are making investment decisions based on information from, for instance, financial statements, recent news about companies, industry trends, and future prospect of companies. They want to know if a company is worth to invest in and how valuable its stock will be in the near future. This research is tackling these problems by utilizing machine learning models, which will classify companies into "good" and "poor" investment categories based on information about financial ratios and macroeconomic variables.

Machine learning has gained a lot of attention in recent years and could also provide better methods and results for analyzing equity price movement. In machine learning, computers are programmed to optimize performance criterion of a model using experience. Experience refers to a past information, which is used for making predictions and decisions (Mohri, Rostamizadeh & Talwalkar 2012, 1). Shynkevich, McGinity, Coleman and Belatreche (2017) predicted short-term equity price movement using different technical indicators such as simple moving average, exponential moving average, relative strength index, the William's %R oscillator, and others as input variables in machine learning models. They used three different machine learning models support vector machine (SVM), k-nearest neighbors (KNN), and artificial neural networks (ANN) to forecast future directions of stock price movements. The results from Shynkevich et al. (2017) research indicate evidence that machine learning can be useful for predicting short-term equity price movement and give higher returns for investors.

Milosevic (2016) and Dutta, Bandopadhyay, and Segupta (2012) used machine learning models to forecast long-term equity price movement based on financial information of companies. Milosevic (2016) reported that the best machine learning model for predicting long-term stock price movement was a random forest model, which had a prediction accuracy of 75.1%. Also, Dutta et al. (2012) concluded that logistic regression achieved 74.6% level of accuracy when classifying stocks into "good" and "poor" investment categories based on their rate of return.

Earlier studies of forecasting stock price movement have pointed out that machine learning algorithms can enhance investment decisions and provide better returns with decent prediction accuracy. Therefore, this research will model and forecast long-term equity price movement in the Finnish financial markets using different machine learning models. The research will provide insight into how well machine learning models can predict price movements of the stocks listed on the OMXH Stock Exchange by testing different machine learning models and a wide variety of independent variables in the models. The results will provide more information about the efficiency of the Finnish financial markets as well.

1.2 Research objective and questions

The research objective is to study different machine learning models to find out which model can predict the best long-term price movement of the stocks listed on the Helsinki Stock Exchange. Also, the effects of macroeconomic variables of Finland on the equity price movement will be examined closely. For instance, could macroeconomic variables increase the prediction accuracy and provide more information for the models that is not explained by only using financial ratios of the companies. Therefore, the following research questions are:

- What is the best model for modeling and forecasting long-term equity price movement of the stocks listed on the Helsinki Stock Exchange?
- What are the core variables that explain the equity price movement and provide the best prediction results?
- Can macroeconomic variables enhance the prediction results of the machine learning models?

Following machine learning models; logistic regression, support vector machine (SVM), decision tree, random forest and k-nearest neighbors (KNN), will be used and compared to each other to analyze, which one of them provides the best results when using the same input variables in each model to forecast stock price movement. If the predicting performance of a model is between 60-70%, it could be a useful model for investing and considered as a good model for predicting the long-term equity price movement. Dutta et al. (2012) reported 74.6% and Milosevic (2016) 64.3% prediction accuracy using logistic regression when predicting equity price movement using only financial indicators as independent variables in the models. Milosevic (2016) obtained the best prediction result, 75.1% accuracy, with a random forest method.

The subject of the thesis is important since many investors are investing on long-term to save and gain capital for future plans, such as retirement, a college education or for a future house. Yet, predicting stock prices is a challenging problem in finance because they can be extremely volatile, especially on short-term. The focus of the research is on predicting the long-term movement of stock prices, which is a little bit different from the ordinary price prediction studies. Since there have been many price prediction studies before, this research is differentiated from the other similar studies in a couple of different ways.

The research will utilize financial ratios as well as macroeconomic variables to test if these variables could provide better results when predicting long-term equity price movement. Two kinds of models will be compared and examined closely; the models that contain only financial ratios and the other models that include also macroeconomic variables. This way the macroeconomic variables will be tested and analyzed if they could increase the performance of the models compared to the other models, which contain only financial ratios.

Earlier literature at the time this research was written, did not point out to have empirical studies that had examined on a deeper level the effects of macroeconomic variables in machine learning models when predicting long-term equity price movement. For instance, Ballings, Van den Poel, Hespeels and Gryp (2015) included macroeconomic variables in machine learning models to forecast equity price movement, but they did not examine how much the variables increased the accuracy of the models. Therefore, the effect of macroeconomic variables will be analyzed by comparing the two kinds of models.

1.3 Theoretical approach

According to the efficient market hypothesis (EMH) stock prices constantly reflect all available information on a financial market. In efficient markets, new information will spread instantaneously into the stock prices without delay. Also, successive price changes of individual stock are independent of the price movement of another stock. Therefore, each stock in the efficient market follows a process called random walk (Fama 1965).

If the stocks in the Finnish financial markets, follow a random walk process the machine learning models should not be able to obtain prediction accuracy over 50%. Meaning that the models will not provide excess returns for investors based on the information about financial ratios and macroeconomic variables. If the variables turn out not to be useful when predicting the long-term equity price movement, it means that the Finnish financial market fills semi-strong-form efficiency requirements (Pilbeam 2010, 240).

The efficiency of financial markets is one of the most studied areas in finance literature and earlier studies have reported results that have been aligned as well as conflicted with the EMH. For instance, Chitenderu, Mredza and Sibanda (2014) found evidence that stock prices were uncorrelated and followed a random walk process in the Johannesburg Stock Exchange during the years 2000–2011. On the other hand, Lo and McKinlay (1988) argued that stock prices did not follow random walk process in the US stock market from 1962 to 1985. They pointed out that stock returns can be predictable to some extent.

Dutta et al. (2012) study indicated that the Indian financial market does not fill semi-strong-form efficiency requirements and the stock prices are not always following a random walk process. This is because their logistic regression model was able to obtain 75.1% prediction accuracy when predicting the stock price movements. Also, Milosevic (2016) reported the same kind of results for the stocks listed on S&P 1000, FTSE 100, and S&P Europe 350 indexes. These results make it interesting to study the equity price movement also in the OMX Helsinki Stock Exchange.

1.4 Methodology, methods and data

This quantitative study includes machine learning models, which are logistic regression, SVM, decision tree, random forest, and KNN. Quantitative study is the most suited research method for analyzing the research problem because the data set is large and statistical methods will be used. These methods could provide reliable results and help to generalize them to the whole population, in this case, to the Finnish financial markets. The models are chosen because earlier literature has shown that they can achieve good results and be useful for predicting the equity price movement. All the machine learning models used in the study are based on supervised learning.

In supervised learning, a statistical model is built for predicting an output variable based on input variables that the model has not encountered before. In the data set that is used for learning, each independent or predictor variable (x_i, i = 1,2,…,n) has a corresponding response variable (y_i). This data set is first divided into training and testing sets, and then the constructed models will be fitted to the training set. The fitted models will infer a function that classifies the observations in the training set. After the training part, the inferred function can be used for new observations in the test set to map the response to the predictors. This is the testing part and the aim is to correctly predict the class of the response variable for each new observation in the test set (Mohri et al. 2012, 7; James, Witten, Hastie & Tibshirani 2013, 1, 26).

Data used in the research is collected from Thomson Reuters Datastream and it includes financial information of all non-financial companies, which have been listed on the OMX Helsinki Stock Exchange from 2000 to 2018. The data contains also different macroeconomic variables of Finland which are collected from the Datastream as well. In this research, a time-period is considered long-term when it is from quarter to another or longer. The whole sample time period is divided into training and testing sets in the following way. The time period of 2000 to 2015 is used for training the machine learning models and the time period of 2016 to 2018 is used for testing purposes where the models try to predict the equity price movement. The test set is used in order to gain understanding of how well the models will predict the long-term equity price movement and how their estimations differ from the other models.

In this research, the machine learning models will be trained in a way that they can predict which stocks will have 10% or higher return in a year and which ones will not have. Logistic regression model is the simplest model that is conducted in the research. The results from earlier literature have shown evidence that logistic regression can classify quite well stocks into two different investment categories: "good" and "poor" based on their returns (Dutta et al. 2012). After logistic regressions, SVM, decision tree, random forest, and KNN classification methods will be tested and analyzed. Ultimately, all the models will be compared with each other and their performances evaluated to find the best model for modeling and forecasting long-term equity price movement.

In the models, equity price movement is the dependent variable, which is a dummy variable and will have either value 1 or 0. If the value of the variable is 1, it indicates that the return of a stock has been 10% or more in a year. This stock will be considered then as a good investment. Otherwise, the equity price movement variable contains the value 0, which indicates that the return of a stock has been below 10% in a year. In the research, this stock will be categorized as a poor investment. A return of 10% has been selected as a benchmark because Milosevic (2016) used 10% return as a benchmark in his research as well. This helps to compare the results from this study to the results Milosevic (2016) obtained.

The models will use financial ratios of companies listed on the OMX Helsinki Stock Exchange and macroeconomic variables of Finland as independent variables. The same financial variables such as price-to-book ratio, price-earnings ratio, dividend per share, and quick ratio will be included in the models as Milosevic (2016) and Dutta et al. (2012) used in their research. The variables will be included because these studies have pointed out that they can explain well the performance of companies and their current financial state. These variables measure company performance from multiple aspects which helps to reduce overlapping information that could happen when too many financial ratios are included in the models. Overlapping information increases correlation between the independent variables and might ultimately provide biased or false results. Macroeconomic variables of Finland such as the unemployment rate, GDP, and expected inflation will be tested to analyze if they can increase the prediction accuracy of the models. Also, Ballings et al. (2015) included GDP and unemployment in their machine learning models when predicting equity price movement.

The forecasting will be done by using the same models constructed in the training part. The models try to predict the values of the equity price movement based on the information on testing data set that contains only observations from quarter one in 2016 to quarter two in 2018. Two types of models per each machine learning algorithm will be trained and tested. The first type of model contains only company specific information and the second type of model includes also macroeconomic variables. Ultimately, all models will be compared to each other to find the best model and examine whether the macroeconomic variables could provide better results for the prediction accuracy. The final results of the models will be evaluated by using different metrics, such as confusion matrix, precision, recall, F1 score, the receiver operating characteristics (ROC) curve, and the area under the ROC curve (AUC).

1.5 Anticipated contributions and limitations

The research provides information about which machine learning model could be useful when analyzing and predicting long-term equity price movement of the stocks listed on the Helsinki Stock Exchange. A successful model could provide useful information for choosing potentially good companies to make a long-term investment. The study will also test the effects of macroeconomic variables to the long-term equity price movement. It will be examined, if these variables could increase the prediction accuracy of the models used in this research. In addition, it will give an insight on how useful the financial information about companies and the macroeconomic variables of Finland are when analyzing long-term investment opportunities. Therefore, the study tests the market efficiency of the Finnish financial markets as well and will provide information on how efficient the market really is. The findings from this research contribute to the financial theory of market efficiency.

One limitation is the available data of the companies in the Finnish financial markets. The data used in the research contains quarterly financial information of the companies, but it is historical and static which might not give as accurate prediction results as preferred. Despite the fact that there are quite a few observations per year, the results can be reliable since there will be as many companies as possible. Therefore, there are more data points in the final data set. Also, the selected time period is wide and includes major market movements such as the financial crisis of 2007–2008.

The second limitation is that all possible machine learning models will not be tested in the research. Therefore, it could be possible that there exist even better machine learning models than the ones, which are tested in this research. Also, one other limitation is the quality of the data from Datastream. Many financial indicators were not available for the companies listed on the OMXH and some of the indicators did not contain values for each quarter. Instead for some financial indicators, the values stayed the same between each quarter and changed only on a yearly basis. The quality of the data affects greatly on the prediction results of the models.

1.6 Structure of the study

This thesis contains five main chapters and is divided in the following manner. The first chapter introduces to the topic, the second chapter covers the theory, the third chapter is about the data used in the research, the fourth chapter introduces the results, discusses and reflects them to the previous findings, and the fifth and final chapter presents the conclusions of the research, and suggestions for future studies.

The second chapter will begin by presenting the relevant finance theory for this study. Following topics will be covered: the efficient market hypothesis, the adaptive market hypothesis, and the main concepts of behavioral finance. After the finance theory is discussed, previous studies will be introduced. Main studies and findings that have been made in the field of predicting stock prices and their movements will be presented. Finally, at the end of chapter two, the theory of all the machine learning models that will be used in this study are presented. The theoretical framework of the different models; logistic regression, support vector machine, decision tree, random forest, k-nearest neighbors will be covered in this order.

In the third chapter, the data and the variables used in the machine learning models will be presented. Different variables, such as company specific variables and macroeconomic variables of Finland will be discussed at a more detailed level, for instance how they were calculated. Company specific variables are market capitalization (MC), P/B ratio (PBR), P/E ratio (PER), quick ratio (QR), earnings before interest and taxes (EBIT), earnings per share (EPS), dividend per share (DPS), dividend yield (DY), close price of a stock (CPS), return on invested capital (ROIC), and total debt divided by total capital (TDTC). The macroeconomic variables are unemployment rate (UR), gross domestic product (GDP), and expected inflation for the next 6 months (EXPINF6M). The descriptive statistics of the variables will be presented as well. Then, the chapter moves on to describing how the data was prepared for the analysis part of the research and what kind of steps and decisions were made along the way.

The fourth chapter introduces the results of all the machine learning models used in the study. In this part of the study, there are two types of models per each machine learning algorithm. The first model contains only company specific information, the chosen financial ratios. The second model includes also macroeconomic variables. These models will be compared to each other to examine what kind of effects the macroeconomic variables have on the equity price movement. This will also provide an answer to the research question whether the macroeconomic variables could provide better results for the prediction accuracy. The fourth chapter begins by presenting the results of logistic regression model and then moving on to presenting the results of other models in the same order as they were presented in the second chapter. All results of the models will be evaluated by using different performance metrics, which are confusion matrix, precision, recall, F1 score, the receiver operating characteristics (ROC) curve, and the area under the ROC curve (AUC). The results obtained from this study will be reflected to the results from the previous studies. At the end of the fourth chapter, all the results of the models will be summarized.

In the fifth and final chapter, the conclusions of the results will be presented and the main reasons why the results might differ from the previous studies in equity price movement will be discussed. At the end of the chapter, interesting topics and aspects will be provided to consider for continuing the research in predicting equity price movement.

2 THEORETICAL FRAMEWORK AND MODELS

In this chapter, the relevant finance literature and the essential previous studies that reflect on the topic of this research will be presented. Finance literature covers topics, such as the efficient market hypothesis, behavioral finance, and the adaptive market hypothesis. After the finance theory, the chapter moves on to describing the theoretical framework of the different models used in this study. The following models will be presented; logistic regression, SVM, decision tree, random forest and KNN.

The efficiency of financial markets is one of the most studied areas in finance literature and there is a great debate between supporters and opponents of the efficient market hypothesis. Therefore, there are also lots of empirical results for and against the efficient market hypothesis.

2.1 Efficient financial markets

According to Fama (1970) a market is efficient when prices of securities constantly reflect all available information. Rational investors are competing against each other to maximize their profits in an efficient financial market. They are constantly trying to predict the future values of securities. The competition among the investors in the marketplace is furious and will lead to a situation in which actual prices of securities already contain all available information (Fama 1965).

When new information appears, it will spread instantaneously and incorporated quickly into the prices of securities without delay. If financial markets are efficient, it means that technical analysis and fundamental analysis would not add value or be otherwise useful for investors. Technical analysis is an analysis method for forecasting future stock prices based on their past prices. Fundamental analysis is an analysis method where financial information of a company, such as assets, liabilities, and earnings will be evaluated to assess its intrinsic value. When financial markets are efficient, investors who have selected stocks for their portfolio using these analysis methods would not benefit by achieving greater returns than those who have randomly selected stocks in their portfolio (Malkiel 2003).

Welch and Goyal (2008) examined the equity premium prediction in the US markets. They used S&P 500 index returns from 1926 to 2005. Welch and Goyal (2008) found out that the models which contained performance variables, such as dividend-price ratio, earnings-price ratio, book value and others predicted poorly equity premium. This result supports that the US market is efficient since predicting equity premium is difficult and the models indicated to be unstable.

2.1.1 Random Walk Theory

The random walk theory or the random walk hypothesis states that successive price changes of individual security are independent of the price movement of another security. Therefore, each security in a stock market follows a process called random walk. In general, a statement that a financial market has no memory means that the information about past prices of individual securities are not useful for predicting the future price of the securities. Therefore, investors should not be able to predict the prices of securities with accuracy over 50 percent. The random walk theory is consistent with the efficient market hypothesis and assumes that markets are operating efficiently (Fama 1965).

There have been many empirical researches, which have examined the random walk theory and predictability of stock prices. For instance, Odean (1999) studied investors who had discount brokerage accounts and tested whether they had sufficient trading profits to cover their trading costs. He pointed out that investors with brokerage accounts did far worse than investors with a simple buy-and-hold strategy. The investors, who had discount brokerage accounts, bought securities that did not outperform the securities they sold. The securities they sold did not even cover trading costs and on average these investors bought securities that underperformed the ones they sold. Chitenderu, Maredza and Sibanda (2014) tested the presence of the random walk hypothesis in the Johannesburg Stock Exchange using monthly time series of the All Share Index from year 2000 to 2011. They concluded that in the Johannesburg Stock Exchange stock prices are uncorrelated and followed a random walk process.

On the other hand, Lo and MacKinlay (1988) found out results that contradict to the previous studies. They claimed that stock prices do not follow the random walk process. Lo and MacKinlay (1988) utilized sample period of 1962–1985 from US stock markets. They included different indexes and formed size-sorted portfolios to test random walk hypothesis for weekly US stock market returns. Lo and MacKinlay (1988) found out that the random walk model was strongly rejected for the entire sample period and for all tested subperiods. They pointed out that there exists evidence which supports the circumstance that stock returns can be predictable to some extent.

Fama and French (1988) had similar results as Lo and MacKinlay (1988). They reported that the autocorrelation; correlation between time series values with its lagged version values is weak for the daily and weekly holding periods, but stronger for long-horizon returns. Indicating that there exist periods when stock prices do not follow a random walk process. Especially, in long-term, stock prices could be predictable. Furthermore, Kwon, Choi and Moon (2002) indicated that the price movement of stocks is not purely random. They found out that there is a statistically significant correlation between prices of certain stocks. This means that in some cases, one stock can be used to forecast the price movement of another stock.

2.1.2 Efficient Market Hypothesis (EMH)

The efficient market hypothesis (EMH) is one of the major theories in finance. It states that the equity value of a listed stock reflects all available information and investors cannot make constantly risk-adjusted excessive returns since market prices should only react to new information. There are three different states of market efficiency; weak-form efficiency, semi-strong-form efficiency, and strong-form efficiency (Fama 1970).

In a market which fills weak-form efficiency requirements, the current prices of securities immediately and entirely reflect all past price information of the securities. Investors should not be able to gain constant excess returns by analyzing past price behaviors of securities to predict their future prices (Fama 1970). Technical analysis methods should not be useful for investors or provide consistent risk-adjusted excess returns for them in the long run. However, in a weak-form efficient market, fundamental analysis is useful and can provide consistent excess returns for investors who are using it (Pilbeam 2010, 240).

In a market which fills semi-strong-form efficiency requirements, the current prices of securities immediately and entirely reflect all information that is publicly available. Publicly available information can be, for instance, information related to an economy or company, such as interest rate changes, announcements of annual earnings, news about stock splits or changes in management (Fama 1970). In a semi-strong-form efficient market, investors should not be able to consistently earn risk-adjusted excess returns by utilizing information that is publicly available to predict future price movements. Neither technical nor fundamental analysis should be beneficial for the investors in a semi-strong efficient market. Although, the investors could use insider information to earn consistent risk-adjusted excess returns (Pilbeam 2010, 240).

In a market which fills strong-form efficiency, the current prices of securities immediately and entirely reflect all information that is public or private. Even investors who have access to insider information, for example, directors of the company or analysts, should not be able to gain constant risk-adjusted excess returns using this information when trading securities in the market (Fama 1970). Market efficiency can be tested through these three different stages of efficiency. There have been several empirical studies that have examined the weak-form market efficiency in different capital markets. Also, the semi-strong market efficiency has been examined in different academic studies, but the empirical results differ from each other considerably. The strong-form market efficiency has not been studied as broadly as the other market efficiencies. Also, the results from the strong-form market efficiency studies vary as well, but overall the results are still more towards to market inefficiencies and reject the strong-form market efficiency.

Borges (2010) examined the weak-form market efficiency in the European stock markets from January 1993 to December 2007. She found contradictory results for the weak-form efficiencies in UK, France, Germany, Spain, Greece, and Portugal stock markets. The empirical evidence indicates that strong positive first-order autocorrelation exists in daily returns of Greece and Portugal indexes. Positive first-order autocorrelation means that the observations, which are one apart from each other, are positively correlated. Therefore, according to Borges (2010) findings, Greece and Portugal financial markets did not fulfill the requirements of weak-form efficiency in this time period. Also, the EMH was rejected for UK and Greece financial markets, due to the presence of positive serial correlation (mean-aversion) in weekly returns. On the other hand, Borges (2010) reported that the EMH was not rejected for Germany and Spain, Germany being the most efficient market.

Basu (1977) tested the presence of semi-strong market efficiency by analyzing the relationship between price-earnings ratio and investment performance of equity securities. He used a large data sample of 1400 industrial firms listed on the NYSE in 1956–1971. He found out that the information of P/E ratio was not entirely incorporated in security prices, therefore this ratio could be used to predict future investment patterns. These results are inconsistent with the semi-strong-form of market efficiency. Basu (1977) pointed out that even after adjusting taxes and transaction costs, the stocks with a low P/E ratio outperformed the stocks with a high P/E ratio by earning higher risk-adjusted returns on average.

Chau and Vayanos (2008) examined the presence of strong market efficiency by studying the actions of monopolistic insider trader. They used an infinite-horizon and steady-state model, where new information was revealed to a monopolistic insider in every trading period. The information revealed contained expected dividend growth rates for different assets. The empirical results from Chau and Vayanos's (2008) study indicated opposite results to previous literature. They pointed out that a financial market can occasionally hold strong-form efficiency requirements, and still offer significant returns to investors who acquire information, despite of the presence of monopolistic insiders. When the financial market develops more towards continuous trading, the information that the insider trader has will be incorporated in the prices almost instantly. The information would be otherwise incorporated in the asset prices after a long series of dividend observations. Back and Pedersen (1998) examined strong-form market efficiency as well with a continuous-time and finite-horizon model. In this model a monopolistic insider is exposed to a flow of private information during the trading session. They pointed out that the insider trader reveals the obtained information slowly. This will cause a financial market not functioning efficiently.

2.1.3 The efficiency of the Finnish financial markets

The financial system in Finland has been strongly bank-centered and therefore small and medium-sized companies have been heavily dependent on loans from banks (Hyytinen & Väänänen 2002). The Helsinki Stock Exchange (HEX) was established on October 7, 1912 and is currently known as NASDAQ OMX Helsinki. At first, there was not a lot of trades made on the exchange, but the trading started to increase towards the end of the 1910s. In the twenty-first century, there were a couple of major mergers that concerned HEX. In 2003, HEX was merged with OM AB, which was the owner of the Stockholm Stock Exchange, and the new company was eventually renamed as OMX Ab. In 2008, OMX was also merged with NASDAQ and they formed the NASDAQ OMX group (Nyberg & Vaihekoski 2014).

As for any other stock market, also the Finnish stock market has had its ups and lows. One of the worst economic crises in Finland was the early 1990s depression. It had a deep impact on the economy of Finland; employment, stock prices, and trading volume dropped deeply. Overall, the stock market declined from the beginning of 1990 to mid-1993. In 1994, after the negative trend, the stock prices rose quickly to the level before the depression and even surpassed it. One of the major reasons for the quick recovery of the stock prices was the success of Nokia Corporation. In these times, the market capitalization of the company was more than 70% of the total market capitalization of the Helsinki Stock Exchange (Kiander & Vartia 2011; Nyberg & Vaihekoski 2014).

The efficiency of the Finnish financial markets has been examined from different perspectives, for instance, Hietala (1994) examined the efficiency of the Finnish market for right issues. Stock rights give existing shareholders right to buy a certain number of extra shares of a company at a particular price. Hietala (1994) concludes that the Finnish market for the rights has not been efficient between the years 1977–1981. The markets appeared to be inefficient for stockbrokers who could have used simple arbitrage rules to earn more profits, because they benefit from substantially lower transaction costs. Although, Hietala (1994) pointed out that it was not established in the research whether the stockbrokers could have earned substantial profits. This was due to infrequent profit opportunities and thin trading on the Helsinki Stock Exchange at that time.

Martikainen and Puttonen (1996) studied day-of-the-week effects in the Finnish stock market. They reported evidence which supported that the day-of-the-week effect exists in the Finnish cash and derivatives markets. Martikainen and Puttonen (1996) identified that in the cash market, there was evident negative returns on Tuesdays and Wednesdays. Also, the Monday effect was strong in both options and futures markets. In addition, Nyberg and Vaihekoski (2014) have studied equity premium in Finland. They used large sample data from 1912 to 2009 and proved that the Finnish market has offered lower real returns compared to the US market. However, the equity premium in Finland, 10.14% per annum, is higher compared to the equity premium in the US market, 9.35% per annum. Nyberg and Vaihekoski (2014) argued that the Finnish stock market has matured. Their empirical evidence suggests that the Finnish market efficiency has increased, and the market is becoming more connected with the international markets.

2.2 Behavioral finance

Behavioral finance is a sub-field of behavioral economics. It has gained popularity because the traditional finance theory about a rational investor who maximizes his expected utility in efficient markets is unable to explain many empirical findings that are contradictory to what the EMH states. Behavioral finance is receiving more attention and therefore the gap between these two finance theories is becoming closer. Behavioral finance helps to explain why and how inefficiencies in the financial markets might exist. The purpose of behavioral finance is to provide explanations why people make certain financial choices by combining behavioral and cognitive psychological theory with conventional economics and finance. It also studies how market prices and other market factors change when investors with different interests participate in a market and make trades with each other (Baker & Nofsinger 2010, 3; Shleifer 2003, 25).

Traditional finance assumes that markets are efficient and rational. Investors in these markets make unbiased decisions and maximize their expected value. An investor who makes suboptimal decisions and errors will be punished by poor outcomes. These suboptimal decisions and errors are not correlated with other market participants and therefore bad decisions would not have an impact on market prices. Over time, investors will learn to make better decisions or leave the marketplace, since unbiased rational investors will exploit the situation in their favor, as in the survival of the fittest (Baker & Nofsinger 2010, 333).

Behavioral finance, on the other hand, assumes that the thinking process of the human brain cannot be compared to a computer. Instead, human processes information through heuristics; mental shortcuts and emotional filters which able them to simplify complex problems and eliminate the need for extensive calculations. These heuristics affect financial decision making and can often lead to suboptimal decisions. Using heuristics and mental shortcuts humans can make irrational financial decisions by not following underlying concepts of risk aversion. Suboptimal financial decisions have a negative influence on the efficiency of capital markets and their personal wealth (Baker & Nofsinger 2010, 3).

2.2.1 Influence of psychological factors on financial markets

Investment decisions often involve uncertainty. Kahneman and Tversky (1979) have studied how humans make decisions in situations which involve risk and sometimes when the outcome is certain. They introduced a model called prospect theory. Investors make decisions according to the principles of prospect theory and they are expecting favorable returns. One of the major assumptions of prospect theory is that investors are more concerned with losses than gains. This leads to a behavior known as loss aversions where investors tend to assign more significance to avoiding loss than achieving gain (Kahneman & Tversky 1979).

Kahneman (2011, 20–21) describes two different ways of how the brain forms and process thoughts. These two modes of thought are called "System 1" and "System 2". System 1 operates fast and automatically with less effort and no sense of voluntary control. It is the intuitive way of thinking and making decisions. System 2 operates slower, requires more effort, and is more logical than System 1. System 2 is the analytical way of thinking and making decisions. People spend most of the time in System 1 and might over-rely on it. This can result in making wrong judgments and decisions due to biases and heuristics.

There are two main biases involved in investment decisions that most of the investors are exposed to. These biases are known as overconfidence and optimism. Overconfidence investors have tendency to overestimate or exaggerate their decisions (Ullah, Ullah & Rehman 2017). Overconfidence can be harmful for investors and deteriorate their ability to pick good investment options. Odean (1998) found out that overconfident traders conducted on average more trades and their expected utility was lower than the other traders which were less confident. Overconfident traders make biased judgments that could lead to lower returns and they also often hold undiversified portfolios. According to Odean (1998), overconfidence increases expected trading volume and market depth.

Optimism is another bias that might affect the decision making of investors. Optimism leads to increasing risk taking. Optimistic investors ignore risk and assume that the future is favorable for them. They also underestimate the likelihood of bad outcomes and are exposed to an illusion of control. Meaning that the optimistic investors believe that they have more control of their fate that what they truly have (Ullah et al. 2017; Kahneman & Riepe 1998). It can be concluded that the two psychological biases; overconfidence and optimism, can have a major impact on investors' decisions and therefore having also an influence on market efficiency.

2.2.2 Adaptive Market Hypothesis (AMH)

In finance, the EMH has been the most popular theory explaining financial markets. However, Andrew Lo (2004) has introduced the Adaptive Market Hypothesis (AMH) which reconciles principles of the EMH with the principles of behavioral finance. He argues that individuals are neither entirely rational, nor entirely irrational and therefore market inefficiencies do exist (Lo 2017, 186).

According to Lo (2017, 2), the AMH identifies that investors and financial markets behave more like biology than physics, implying that the principles of evolution, competition, innovation, reproduction, and adaptation are more useful for understanding the dynamics of the finance industry than the principles of rational economic analysis. Lo (2017, 2) argues that the AMH is a new version of the EMH which has been derived from evolutionary principles. The term "adaptive markets" means that human behavior and financial markets are shaped by evolutionary forces, and the word "hypothesis" is meant to combine and compare this framework to the EMH.

Lo (2017, 188) presents the following key principles of the AMH:

1. The actions of individuals are not always rational nor irrational. Forces of evolution have affected these actions over time
2. Suboptimal decisions and behavioral biases do exist, but individuals can learn and adjust their behavior and heuristics accordingly to negative feedback they receive from their environment
3. Evolution has enabled that individuals are capable of abstract thinking, such as making predictions about the future and adapt to changes in their environment
4. The interactions between individuals and how they behave affect financial market dynamics and environments where they interact
5. One of evolutional forces, survival, is the fundamental factor of competition, innovation, and adaptation

The AMH can explain economic behavior that is only approximately rational and economic behavior that looks completely irrational. According to Lo (2017, 189) individuals are not certainly sure how good their current heuristic is and whether it is "good enough" for their current environment. Individuals will make conclusions through trial and error. Based on their past experiences and best guesses individuals make choices that could be close to optimal. Individuals will learn and adapt their heuristics based on feedback, positive or negative, from the environment they are involved in. Individuals will also have to develop new heuristics and mental rules of thumb so that they are capable to solve new economic challenges. If the challenges of the environment do not change over time, their heuristics will reach close to optimal solutions of economic challenges (Lo 2017, 188).

Individuals will have to react to the changes happening in their environment by adjusting their heuristics, because already learned heuristics from the previous environment will not always be suitable for the new changes. If heuristics are not changed, the behavior of individuals will look irrational. Learning is a crucial factor for individuals to adapt to the new environment, and it happens only if they receive positive or negative feedback from the new environment. If the feedback is inappropriate individuals will learn suboptimal behavior and their behavior will look irrational again. It is also difficult for individuals to reach an optimal heuristic if the environment is constantly shifting. This will cause individuals to react irrationally, because their heuristic does not have enough time to adjust to the changes (Lo 2017, 189). For instance, a stock market is a highly complex system and many changes happen rapidly and some slower. Therefore, individuals have to constantly change their heuristics and behavior by learning; receiving positive and negative signal from the market, in order to make as optimal investment decisions as possible.

The AMH is not labeling any behavior as "irrational". However, it recognizes that suboptimal behavior often happens when individuals use their learnt heuristics in a new environment in which they have not been tested before (Lo 2017, 189). The EMH on the other hand, assumes only rational behavior of individuals, but as market inefficiencies do exists the EMH cannot provide answers to these inefficiencies. For instance, Gultekin and Gultekin (1983) have identified the January effect in most of the major industrial countries. The January effect refers to a phenomenon where returns are larger in January compared to other months. Also, Barone (1990), Agrawal and Tandon (1994) have reported similar results of the January effect. The EMH assumes that individuals have no memory of past conditions, they act rationally and accordingly to price changes. All market prices already contain all past information which means that historical data is not useful for predicting the future outcomes and therefore it should be irrelevant (Fama 1970).

Lo (2017, 208) explains that individuals do not always estimate the best use for their money and their buying decisions do not necessarily indicate their preferences. Instead, consumer behavior reflects their history of evolutionary and economic environments. Individuals use heuristics, behavioral biases, and rules of thumb they have learnt over time from their experiences when making decisions. The AMH assumes that the behavior of individuals is highly path dependent. According to Lo (2017, 208) the process of selection keeps the behavior of individuals not being chaotic. The selection process wipes out bad behavior and ensures that new behavior is good enough, but not necessarily optimal or rational.

2.3 Previous studies

Predicting stock prices or movement is certainly not an easy task. According to the EMH, stock prices in financial markets contain all available information and follow a random walk behavior (Fama 1970). This makes it almost impossible for investors to receive constant alpha, excessive returns, over time if financial markets are efficient. However, many empirical evidences have emerged, which are not aligned with the principles of the EMH and therefore behavioral finance has received popularity. In finance literature, there are many interesting studies, which have examined stock returns, for example, Fama and French are respected and important academic researchers in this area.

Fama and French (2012) have designed the Fama–French three factor model to describe stock returns. They argued that three factors; market risk, size, and value explain average stock returns in the USA better than the capital asset pricing model (Fama & French 2012). In 2015, Fama and French introduced a five-factor asset pricing model and showed evidence that the five-factor model performed even better in explaining average stock returns than the three-factor model. The new two factors in the five-factor model were profitability and investment (Fama & French 2015). These factors could be also used for predicting stock price movements. Therefore, in this research firm size and value ratios will be included in the models to examine the prediction capabilities of the variables.

Machine learning and artificial intelligence have received a lot of interest in recent years. In finance research, machine learning has been applied for predicting stock returns or stock and index price movements. Long- and short-term stock and index price movement have been studied by utilizing different machine learning algorithms and different input variables in the machine learning models. For instance, Shynkevich et al. (2017) compared different machine learning methods for forecasting short-term stock price movements. They used technical indicators as input variables for three different models, which were SVM, ANN, and KNN. Shynkevich et al. (2017) objective was to predict the future direction of stock price movements for different time periods. They used time horizons which were from 1 to 30 trading days. They classified the dependent variable, price movement, first by using two classes: up and down, and then three: up, down, and no movement.

Shynkevich et al. (2017) concluded that all the models outperformed a simple buy-and-hold strategy in all the forecasted time horizons. The highest prediction accuracy, 75.43%, was achieved by the SVM model with two different prediction classes for the dependent variable and forecasted horizon of 15 trading days. The highest prediction accuracy for the ANN model was 73.21% and for the KNN model it was 60.26%. The highest prediction results were achieved when just two different prediction classes were selected for the dependent variable. This research interestingly pointed out that SVM models outperformed even ANN models by achieving higher results from following performance indicators: prediction accuracy, return per trade, winning rate, and Sharpe ratio.

In addition, Huynh, Dang and Duong (2017) studied short-term equity and index price movement using deep neural network. They used online financial news and historical stock prices to make predictions for short-term time intervals (1 day, 2 days, 5 days, 7 days and 10 days). They made the short-term predictions for the S&P 500 index and three different companies: Google, Wal-Mart, and Boeing stocks to evaluate the effectiveness of their model. The best deep neural network model obtained 59.98% prediction accuracy for the S&P 500 index, around 62% accuracy for Google stock, around 66% accuracy for Wal-Mart stock, and around 60% accuracy for Boeing stock.

The long-term equity price movement has been studied as well. For instance, Milosevic (2016) used many machine learning models to predict the long-term price moment using financial ratios as input variables for the different models. Milosevic (2016) had wide data set, which contained a total of 1739 stocks from indexes such as S&P 1000, FTSE 100, and S&P Europe 350. He used quarterly data from 2012 to 2015. Milosevic (2016) used 10% benchmark when classifying the stocks into two different categories. The stocks which had 10% or higher returns in a year were classified as good stocks and the stocks which had lower returns than the benchmark returns were classified as poor stocks. He found out that the best model for predicting long-term equity price movement was a random forest with prediction accuracy of 75.1%. Milosevic (2016) used various other machine learning models, such as SVM, decision trees, logistic regression, and Bayesian network, but none of them performed as well as the random forest model. The most valuable financial ratios used in the models were book value, market cap, dividend yield, earnings per share, price-earnings ratio, price-to-book ratio, dividend per share ratio, current ratio, quick ratio, total debt to total equity ratio, and price history of the stocks.

Also, Dutta, Bandopadhyay and Sengupta (2012) studied long-term stock price movement in Indian stock market using logistic regression. They found out that logistic regression used with eight different financial ratios as explanatory variables can classify companies well into two categories – "good" or "poor", based on their rate of return. This logistic regression model achieved as high as 74.6% prediction accuracy. Stock returns were compared to given market return in each year. If a stock return was higher than the market return in a year, a stock would be classified as a good investment and otherwise as a poor investment. The following eight financial ratios used in the logistic regression model were percentage change in net sales, sales/net assets, price/cash earnings per share, price-to-book value, price-earnings per share, PBIDT/sales, cash price/earnings per share, and book value. Dutta et al. (2012) data sample included 30 largest companies in India measured by market capitalization from 2005 to 2008.

Furthermore, Ballings et al. (2015) evaluated multiple machine learning models when predicting stock price movements. They tested following machine learning models: logistic regression, ANN, KNN, SVM, random forest, adaptive boosting (AdaBoost), and kernel factory. Their data consisted 5767 publicly listed European companies between year 2009 and 2010. Ballings et al. (2015) included various financial ratios of the companies and macroeconomic variables, such as public debt, GDP, unemployment, and trade balance. They concluded that the best model was a random forest with AUC value of 90.37%. The second-best model was SVM with AUC value of 83.95%.

Other studies have used machine learning for forecasting index movements. For example, Phua, Zhu and Koh (2003) used neural networks to predict movement of five stock indexes: DAX, DJIA, FTSE-100 HIS, and NASDAQ. Their models were capable to forecast the direction of index movement with an average prediction accuracy above 60% for all five stock indexes, and their prediction results for FTSE-100 HIS and NASDAQ exceeded an average accuracy of 64%. Qian and Rasheed (2007) examined the performance of ANN, KNN, and decision tree models when forecasting the Dow Jones index. They used daily returns of the Dow Jones index from 4 June 1969 to 4 June 1973. Qian and Rasheed (2007) concluded that the best model was ANN with prediction accuracy of 60.09%. The KNN model achieved prediction accuracy of 56.64% and decision tree 56.38%.

Also, Leung, Daouk and Chen (2000) studied predictability of a stock market index movements. They utilized data from three major stock market indices – S&P 500 for the US, FTSE 100 for the UK, and Nikkei 225 for Japan from January 1967 to December 1995. Leung et al. (2000) compared different classification and level estimation models to examine which model is the best for predicting the sign of the index. They used classification models like linear discriminant analysis, binary choice, and neural networks. Leung et al. (2000) results indicate evidence that classification models perform better than their level estimation counterparts in terms of hit rate (number of times the predicted result is correct). A group of all four classification models achieved average hit ratio of 61.67%. Whereas the hit ratio for the group of all four level estimation models was 56.11%. They also pointed out that classification models can achieve higher trading profits that the level estimation models.

Machine learning models can be also used for predicting credit-risk or bankruptcy. Matoussi (2010) studied credit-risk prediction using logistic regression and ANN models. He found out that the ANN model performed better and provided more accurate predictions than logistic regression models, and that non-cash flow variables have a good prediction capacity. Bensic, Sarlija and Zekic-Susac (2005) studied small-business credit scoring using logistic regression, neural networks and decision trees. They argued that the ANN model was also the best model and it extracted a number of important features for small-business credit scoring. The important ones were credit programme characteristics and entrepreneur's personal and business characteristics.

The studies indicate that machine learning can be used for a variety tasks in finance. Also, the equity price prediction studies indicate that by only using financial ratios as inputs for the machine learning models the prediction results can be quite significant and useful for investors. The models can aid investors with different investment choices and assist them to select the good stocks from the bad ones. This makes it interesting to analyze furthermore if macroeconomic variables could enhance the prediction results. It is also intriguing to predict equity price movement of the companies listed on the Helsinki Stock Exchange since the market is not considered as efficient as the financial markets in the USA. Therefore, the prediction results could point out to be significant and provide more insight of the efficiency stage of the Finnish financial markets.

Table 1 Summary of equity price movement studies

Researchers	Name of study	Published	Market / Data	Time period	Models used	Results
Milosevic	Equity Forecast: Predicting Long Term Stock Price Movement using Machine Learning	2016	S&P 1000, FTSE 100 and S&P Europe 350	2012–2015	Decision trees, SVM, JRip, Random Tree, Random Forest, Logistic regression, Naïve Bayes, Bayesian Networks	Best model: random forest, accuracy of 75.1%

Dutta, Bandopadhyay and Sengupta	Prediction of Stock Performance in the Indian Stock Market Using Logistic Regression	2012	The Indian stock exchange (NIFTY index)	2005–2008	Logistic regression	Prediction accuracy of the model was 74.6%
Shynkevich, McGinity, Coleman and Belatreche	Forecasting price movements using technical indicators: Investing the impact of varying input window length	2017	S&P 500	2002–2012	SVM, ANN, KNN	The highest prediction accuracy, 75.43%, was achieved by SVM. The ANN model achieved 73.21% and the KNN model 60.26%. All the models performed better than a simple buy-and-hold strategy.
Huynh, Dang and Duong	A New Model for Stock Price Movements Prediction Using Deep Neural Network	2017	S&P 500, Google, Wal-Mart and Boeing stocks	2006–2013	Deep neural network model	The deep neural network model obtained following prediction accuracies 59.98% for the S&P 500 index, 62% for Google stock, 66% for Wal-Mart stock, and 60% for Boeing stock.

Ballings, Van den Poel, Hespeels and Gryp (2015)	Evaluating multiple classifiers for stock price direction prediction	2015	Publicly listed European companies	2009–2010	Logistic regression, ANN, KNN, SVM, random forest, AdaBoost, kernel factory	The best model was a random forest with AUC value of 90.37%. Other models obtained following results: SVM (80.95%), kernel factory (79.91%), AdaBoost (76%), ANN (72.79%), KNN (72.65), logistic regression (66.06%)

2.4 Assessing a machine learning model

Before presenting all the different machine learning models used in this study, it is important to understand how to evaluate the performance of the model and what are the most important underlying concepts when selecting and using different machine learning models. There is no one superior model that is always dominating all other models over every given data set (James et al. 2013, 29). Therefore, it is essential to be able to evaluate the different machine learning models in order to choose the best model for a given data set.

2.4.1 Prediction accuracy and model interpretability trade-off

Some machine learning methods are more complex and not as easily interpretable while other methods are simpler but more restrictive. For instance, linear regression is a fairly simple model and easily interpretable, but it is inflexible and restrictive because it models only linear relationship between the independent and dependent variables. There are still reasons why simpler but restrictive models are preferred. For example, if the inference of the model is the main goal then a simple linear regression might be a good choice. In the model, it is easy to understand the relationship between the independent and dependent variables (James et al. 2013, 25).

When choosing a more complicated model, it becomes more difficult to understand how an individual dependent variable is associated with the independent variable. These models are highly flexible, such as SVM with a non-linear kernel, which will be covered in chapter 2.5.2. In some cases, more flexible methods are preferred. For instance, when the prediction accuracy of the model is more important than the interpretability of the model. In stock price prediction studies more complicated and flexible methods are being used because it is often more important to achieve a higher accuracy for the models than what its interpretability might be (James et al. 2013, 26).

On the other hand, sometimes more accurate prediction results can be obtained by using less flexible and simpler methods. This peculiarity relates to a problem called overfitting, which might occur in highly flexible models (James et al. 2013, 26). Overfitting is a modeling error where a function is fitted too closely to a certain data set. Overfitted model fails to predict the future observations reliably (Alpaydin & Bach 2014, 39). In this study, the target is to achieve high accuracy for the models, but in a way that the results remain still quite easy to interpret, and not falling into overfitting the models.

2.4.2 Bias-variance trade-off

The bias-variance trade-off is a fundamental problem in machine learning and statistics. Bias is an error, which occurs when a too simple model is used for approximating a complex real-life problem. Real-life problems can be extremely complicated because the problems often do not follow a simple linear relationship. The model will have a high bias when it is too simple for the given problem. For instance, a simple linear regression will have a high bias when it is fitted for extremely complicated real-life problem that is not linear. A model with high bias misses the relevant relationships between dependent and independent variables. This means that the model is underfitting the data (Alpaydin et al. 2014, 32; James et al. 2013, 35).

Variance is an error, which measures the sensitivity of a model to small differences in a training data set. Simpler models usually have less variance compared to more complex ones. Complex models will model the random noise in the training data and therefore will contain higher variance. The random noise causes the models to give highly different results even when changes in the training data set are small. Therefore, the more complex models might result to overfitting (Alpaydin et al. 2014, 32; James et al. 2013, 34).

It is easy to either have a model with low variance or low bias. Although, an ideal situation for a machine learning model is to simultaneously achieve low bias and low variance. Achieving these both properties is a challenging task because when trying to achieve low bias for a model the variance will increase and vice versa. This refers to the bias-variance trade-off and it applies to all forms of supervised learning models used for classification as well as regression problems (James et al. 2013, 36). Therefore, it is present also in the models used in this study.

2.5 Machine learning models

In this chapter, statistical frameworks of the models will be introduced. The results of the models will be presented in the empirical analysis and results chapter. The response variable in the models is equity price movement, which is a binary variable. Therefore, following classification models: logistic regression, support vector machine, decision tree, random forest, and k-nearest neighbors are used in the research to model and predict the movement of the response variable. All the machine learning models used are based on supervised learning.

In supervised learning a statistical model is built for predicting an output variable based on input variables that the model has not encountered before. In the data set that is used for learning, each independent or predictor variable (x_i, i = 1,2,…,n) has a corresponding response variable (y_i). The data set is first divided into training and testing sets and then constructed models will first be fitted to the training set. The models will infer a function that classifies the observations in the training set. This function can be then used for the new observations in the test set to map the response variable to the predictors. The aim is to correctly predict the class of the response variable (y_i) for each new observation (x_i) in the test set. Supervised learning is most commonly used for classification, regression, and ranking problems (Mohri et al. 2012, 7; James et al. 2013, 1, 26).

2.5.1 Logistic regression

Logistic regression is one of the most widely used classifier when the dependent or response variable is discrete. Discrete variables can take either two or more possible values (Chikkodi & Satyaprasad 2010, 48). Generally, the dependent variable is dichotomous, meaning that the variable has only two categories or levels. The response variable in the model is equity price movement, which is a binary variable and can have either value 1 or 0. Logistic regression models the probability of the dependent variable, equity price movement, belonging to a particular category. In this case, if the return of stock is 10% or higher the variable belongs to category 1, and if the return is less than 10% the variable belongs to category 0.

In the model, logistic function is used to model the probability of the equity price movement for belonging either one of the two categories. Logistic function gives probabilities that are between 0 and 1 compared to a linear model, which can give probabilities that are not sensible. For instance, the values can be less than 0 or more than 1 (James et al. 2013, 130–131). In figure 1, the differences of the probability values given by linear and logistic regression are presented.

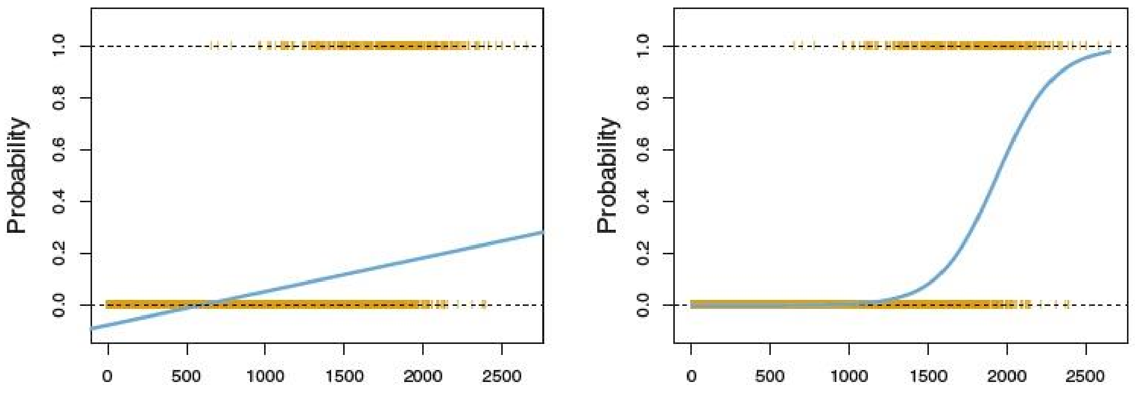


Figure 1 Predicting with linear vs logistic regression (James et al. 2013, 133)

In this research, multiple logistic regression model is used for predicting the outcome of the binary dependent variable, the equity price movement. Multiple logistic regression can be generalized as follows (James et al. 2013, 135; Hosmer, Lemeshow & Sturdivant 2013, 35–36):

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \qquad (1)$$

where
p_i is the probability the ith case encounters the event of the equity price movement
$p_i/1\text{-}p_i$ in the parenthesis is the odds and taking the logarithm of it, is called a log-odds or logit
$X = (X_1,\ldots,X_p)$ are predictor variables
$\beta = (\beta_0,\ldots,\beta_p)$ are coefficients

The coefficients are estimated using maximum likelihood estimation (MLE) because it has very desirable properties. When the sample size is large enough, MLE will be consistent; approaches to its true value, unbiased; expected value of the estimator equals the true parameter value, and efficient; has achieved the lowest possible variance among all other estimators. Therefore, it will be the most precise estimator among all (Eliason 1993, 17). Maximum likelihood method estimates the coefficients in a way that the model will yield a number close to 1 for all observations which have return at least 10% or higher and number close to 0 for all observations which have return below 10%. Likelihood function can be formalized by using the following mathematical equation (James et al. 2013, 133):

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \quad \prod_{i':y_{i'}=0}\left(1 - p(x_{i'})\right) \qquad (2)$$

The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to maximize the likelihood equation. The advantage of the logistic regression is that it does not require the independent variables to be normally distributed. Also, it does not make assumptions of the prior probabilities of the dependent variable (Ohlson 1980). In this case, any assumptions of the prior probabilities of a firm being successful are not made.

2.5.2 Support Vector Machine (SVM)

The original support vector machine (SVM) algorithm was first introduced in 1964 by Vapnik, Chervonenkis and co-workers, but their research paper about SVMs went first largely unnoticed until it was later modified by Cortes and Vapnik (1995). SVM was not popular at first because the statistical and machine learning community believed that SVMs were neither suitable nor relevant for practical applications, despite being theoretically appealing. Regardless of the pessimistic welcome at first, SVMs have gained popularity in recent years because they show better results than most other statistical models and even better or at least comparable results to neural network models (Wang 2005, 2).

SVM is a supervised machine learning model and can be used for either classification or regression problems. SVM can perform efficiently linear and non-linear classification problems and formulates a classification problem as a quadratic programming (QP) problem (Wang 2005, 1–2; Chen, Härdle & Moro 2011). SVM models have indicated great results in different academic studies. For instance, Fan and Palaniswami (2001) reported that SVM was able to identify stocks listed on the Australian Stock Exchange that outperformed the benchmark total return of 72%. The portfolio, which was formed by the SVM, had a total return of 208% over a five-year period. The SVM pointed out to be useful for selecting different stocks, which significantly outperform the benchmark index.

SVM can be also used for predicting stock prices, which is an example of a regression problem. Zhang, Teng and Chen (2018) pointed out that support vector regression can be combined with the firefly algorithm to obtain superior performance when forecasting stock prices. Also, Karazmodeh, Nasiri and Hashemi (2013) reported that SVM combined with another algorithm provided better results for forecasting stock prices.

In this research, SVM is used for classification problem, classifying companies into two classes based on information about financial and macroeconomic variables. The data points fall into the two classes that are represented in SVM as {-1, 1} where -1 represents the negative class and 1 the positive class. For instance, in this research, -1 in SVM will describe the class, which the stocks have not reached 10% return and 1 will describe the class, which the stocks have reached at least 10% return.

SVM classifies data points into the two categories by drawing first a (p-1) dimensional separating hyperplane in p-dimensional space. A hyperplane is a subspace of dimension one less than its ambient space. For instance, in 3-dimensional space a hyperplane is a flat 2-dimensional subspace, a plane, and in 2-dimensional space a hyperplane is a flat 1-dimensional line. In p-dimensional space, a hyperplane can be defined by the following mathematical equation (James et al. 2013, 338):

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p = 0 \qquad (3)$$

which in 2-dimensional space becomes simply the equation of a line (w * x + b = 0) because a hyperplane is a line in 2-dimensional space. Any $X = (X_1, X_2,..,X_p)^T$ in p-dimensional space, for which equation (3) holds, there is an observation on the hyperplane.

If separating hyperplane exists, it divides the p-dimensional space into two parts and can be used as a classifier, which has the following properties (James et al 2013, 340):

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \text{ if } y_i = 1 \qquad (4)$$

and

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \text{ if } y_i = -1 \qquad (5)$$

The test observation x_{i^*} is classified based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_p x_p^*$. If the test observation x^* is positive, it will be assigned to class 1 and if the observation is negative, it will be assigned to class -1. The magnitude of $f(x^*)$ is also important. The test observation x^* will be located far from the hyperplane when $f(x^*)$ is far from zero. The classifier can then be certain that the test observation belongs to a specific class. The test observation x^* will be located close to the hyperplane when $f(x^*)$ is close to zero. In this case, the classifier is less certain about which class the test observation x^* belongs to. This kind of classifier, which is based on a separating hyperplane, leads to a linear decision boundary. If separating hyperplane exists, then there exists also an infinite number of hyperplanes which can separate the observations into two classes. An infinite number of hyperplanes exists because the hyperplane can be shifted up or down just a small fraction from its original place and it will still separate the observations perfectly into two classes. This can be seen on the left-hand side of figure 2, where there are three separating hyperplanes drawn out of many other possible ones. The right-hand side of the figure illustrates the decision boundary made by the hyperplane (James et al 2013, 340–341; Huang, Nakamori, Wang 2005).

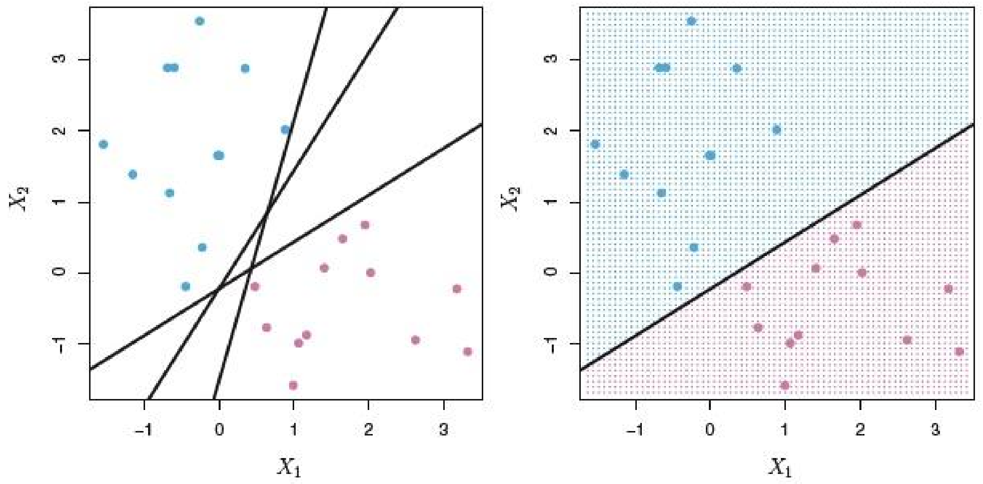


Figure 2 Multiple separating hyperplanes (James et al. 2013, 340)

The best possible hyperplane out of many is chosen by the maximal margin hyperplane, which is also known as the optimal separating hyperplane. Margin is the minimal perpendicular distance from a training observation to the hyperplane. First, the distance of each training observation to each hyperplane is calculated. Then the maximal margin hyperplane can be decided. It will be the farthest hyperplane from the training observations. The margin of the maximal margin hyperplane is the largest and therefore it has the longest minimum distance to the training observations. The maximal margin hyperplane is calculated by maximizing the margin, it is the solution for following maximization problem (James et al. 2013, 343; Huang et al. 2005):

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{maximize}\, M \tag{6}$$

$$subject\ to\ \sum_{j=1}^{p} \beta_j^2 = 1, \tag{7}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M\ \forall\ i = 1, \ldots, n. \tag{8}$$

where M is margin of the hyperplane and $\beta_0, \beta_1, \ldots, \beta_p$ are parameters which are chosen so that they will maximize M. When the maximal margin hyperplane is calculated from the training observations, the test observations can be classified using that same maximal margin hyperplane based on which side of the maximal margin hyperplane the test observations are located. The maximal margin hyperplane classifies the test observations perfectly if the defined margin is the largest for the test observations as well (James et al. 2013, 341). In figure 3, the hyperplane maximizes the margin and separates the training observations perfectly into two different classes.

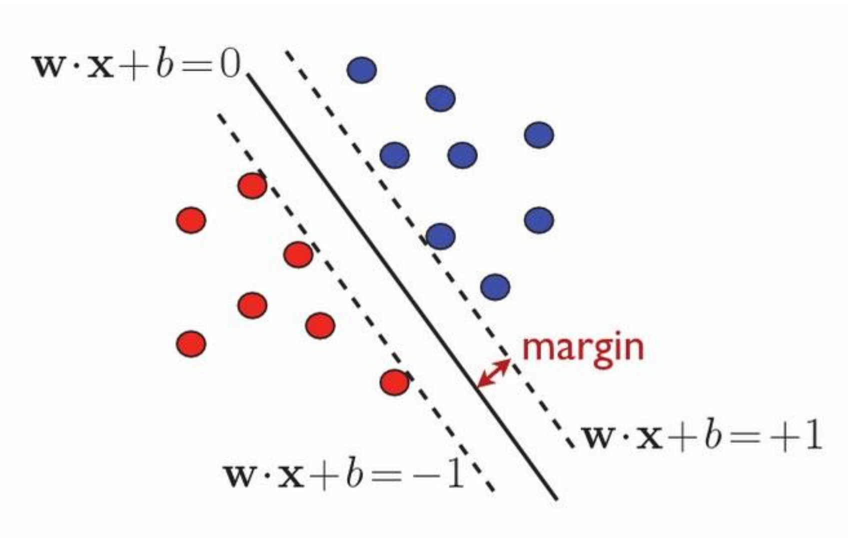

$\mathbf{w}\cdot\mathbf{x}+b=0$
margin
$\mathbf{w}\cdot\mathbf{x}+b=+1$
$\mathbf{w}\cdot\mathbf{x}+b=-1$

Figure 3 The maximal margin hyperplane (Mohri et al. 2012, 65)

In figure 3, four observations are equally distant from the maximal margin hyperplane and they are located on the dashed lines, which define how wide the margin will be. These four observations are the support vectors and they support the maximal margin hyperplane because if these points are moved to another location, the position of the maximal margin hyperplane will move as well. The maximal margin hyperplane is dependent only on these four support vectors and none of the other observations. This is because when the other observations are moved, it would not have an impact on the maximal margin hyperplane if they are not moved across the margins (James et al. 2013, 341–342; Huang et al. 2005).

In figure 3, the classifier separates perfectly the observations into two classes. Perfect linear classifiers cannot always be formed since there are cases were the observations of two classes are overlapping and therefore not perfectly linearly separable. In this case, the maximal margin hyperplane does not exist and the optimization problem (6-8) has no solutions with $M > 0$. The separating hyperplane can be extended to hyperplane that allows some observations to be separated into wrong classes using the soft margin technique, and therefore does not perfectly separate the observations into two classes (James et al. 2013, 343).

The hyperplane that uses soft margin has some benefits compared to the maximal margin hyperplane. It is more robust method, meaning that it is not as highly sensitive to changes in observations than the maximal margin hyperplane is. The maximal margin hyperplane can easily overfit the training data. Therefore, the soft margin classifier will usually achieve better classification of the training observations (Vapnik 2000, 137; James et al. 2013, 343–344). In figure 4, the soft margin hyperplane separates the observations into two different classes, but one red observation is on the other (wrong) side of the hyperplane and one blue observation is inside the margin, but it is correctly classified.

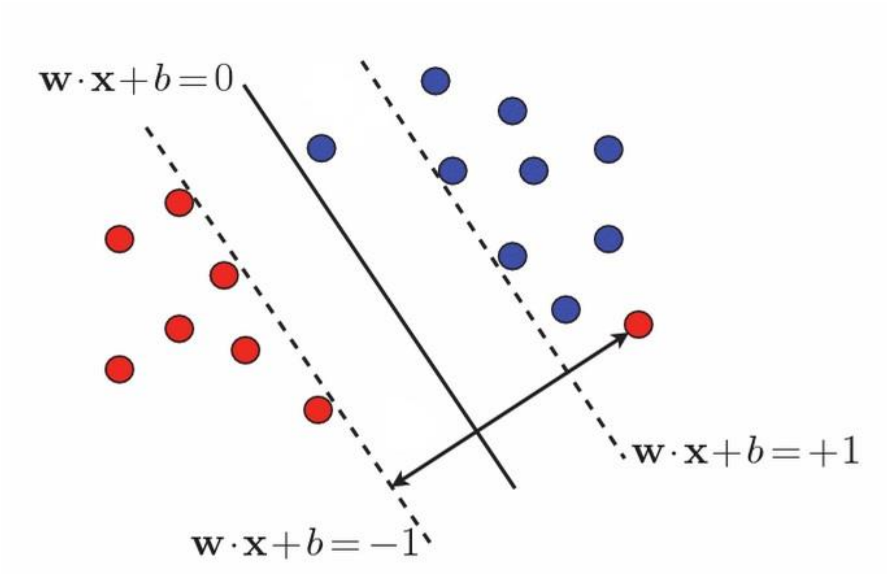


Figure 4 The soft margin hyperplane (Mohri et al. 2012, 71)

The support vector classifiers in figure 3 and 4 are linear, because the hyperplane is linearly separating the observations into two different classes. However, there are often cases where even the linear support vector classifier used with the soft margin technique performs poorly. This happens when the separating boundary is non-linear between the two classes. When classification with non-linear decision boundaries is done, the support vector machine should be used because it performs better. The support vector machine is an extension of the support vector classifier and it enlarges the feature space using kernels. Kernel methods are widely used in machine learning because they are flexible and efficient techniques to define non-linear decision boundaries. For the support vector classifier maximization problem (6–8) it turns out that the solution is just the inner products of the observation, which is for a and b vectors defined as $\langle a, b\rangle = \sum_{i=1}^{r} a_i b_i$ (James et al. 2013, 350; Mohri et al. 2012, 89).

In this research, the support vector machine with radial kernel function is used because the classification problem is non-linear. Support vector machine with radial kernel can be mathematically written by the following equation (Vapnik 2000, 145):

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) \qquad (9)$$

where γ (gamma) parameter is a positive constant that controls the bias-variance tradeoff. When the value of gamma is large, the support vector x_n does not have much impact on the classification of the training observation x_i. Therefore, the SVM model can capture more of the complexity of the data. But a too large gamma value causes the model to overfit the data and therefore have high variance and low bias. On the other hand, a low gamma parameter value allows the support vector to have a greater impact on the classification of x_i. This model will not overfit the data, but it might not learn a decision boundary that captures the shape and complexity of the data. The model will have a high bias and low variance (James et al. 2013, 353).

The term $\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2$ in the equation 9 will be large if the Euclidean distance of a test observation is large from a training observation. This means that $K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2)$ will be tiny. When training observations are far from test observations, they will not have any influence on the predicted class label of a test observation. This implies that the radial kernel behaves locally, resulting that only the close observations have an impact on the class label of a test observation (James et al. 2013, 353). Figure 5 presents how SVM with radial kernel can classify non-linear data and capture the decision boundary efficiently.

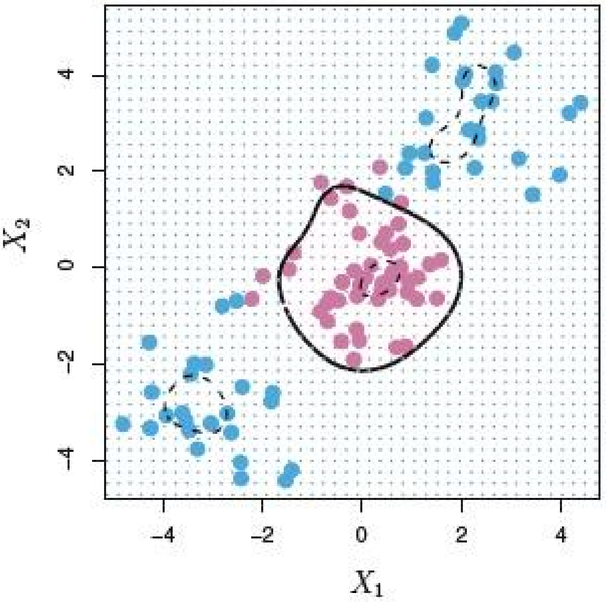


Figure 5 SVM with radial kernel fitted to non-linear data (James et al. 2013, 353)

2.5.3 Decision tree

A decision tree is a hierarchical tree-like model for supervised learning, it divides the predictor space into several simple local regions. This model can be used for both regression (regression tree model) and classification (classification tree model) problems. Classification trees will be covered in more detail in this chapter since the model is used in this study for a classification problem. Classification and regression trees are very similar, except that the dependent variable in classification model is categorical and in regression model it is continuous. The decision tree is a nonparametric model, which means that it does not require any parametric assumptions of the class densities and the tree structure is based on an empirical data it is used for. The structure consists decision nodes, branches, and leaf nodes (Alpaydin et al. 2014, 213; Tseng 2007).

When the model will be fitted to a given data set, the tree evolves, for example, more branches and leaves are added to it during the learning process based on the complexity of the data. Each decision node implements a function, which has discrete outcomes and it labels the different branches. Observations are tested by the functions at each node and they are divided into different branches based on the discrete outcomes of the functions. This process starts at the root of the tree, usually from the top, and repeats itself recursively until a leaf node is reached. Leaf nodes have an output label, which is the class code, and it will be assigned to each output. Leaf nodes define localized regions in the input space where the observations that are divided in the specific region have the same labels. Each decision node is a comparison test:

$$f_m(x) : x_j > w_{m0} \tag{10}$$

where x_j is numeric input and w_{m0} is a suitable threshold value. The decision node divides the input space into two regions: $L_m = \{x|x_j > w_{m0}\}$ and $R_m = \{x|x_j \leq w_{m0}\}$. The notion $\{x|x_j > w_{m0}\}$ means the region of the input space in which x_j takes on a value greater than the suitable threshold value, w_{m0}. This is called a binary split (Alpaydin et al. 2014, 213–215; James et al. 2013, 307).

Figure 6 presents how the decision tree splits observations into two different local regions based on the discrete outcomes of each leaf node of the tree. The leaf nodes are pictured as rectangles on the right-hand side of the figure and they define hyperrectangles in the input space shown in the left-hand side of the figure.

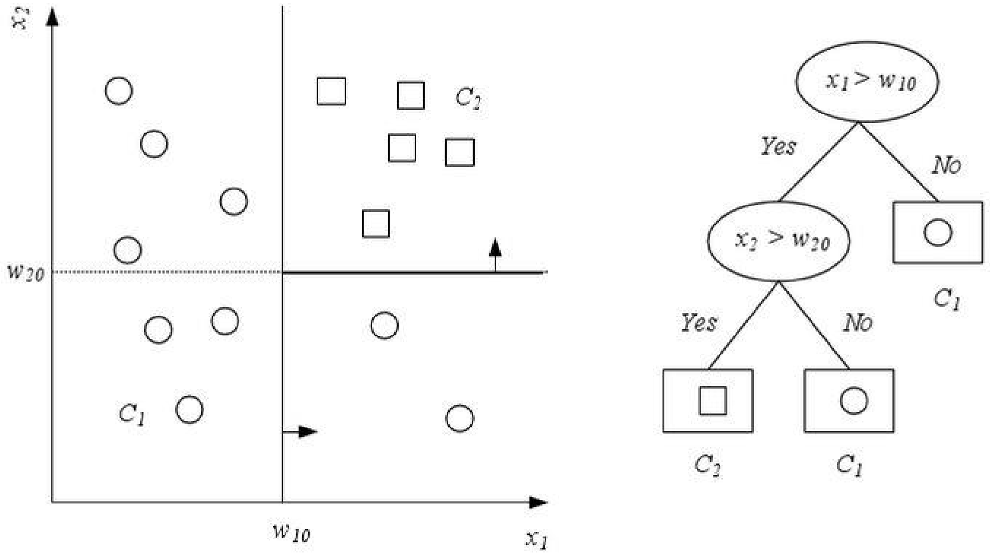


Figure 6 Decision tree model classifying data (Alpaydin et al. 2014, 214)

Binary split is used to grow a classification tree, and the goodness of a split is defined by an impurity measure, such as entropy or Gini index. A pure split is made if all the instances choosing a branch belong to the same class. In this case, no further splits have to be made and leaf node labeled with the class can be added at the end of each branch (Alpaydin et al. 2014, 216).

Entropy function measures the quality of a split in each node and it can be defined by

$$E = - \sum_{i=1}^{K} p_m^i log_2 p_m^i \qquad (11)$$

where p_m^i indicates the proportion of the population with a class label *i* in node *m*. It can therefore get values between 0 and 1. Node *m* is pure when p_m^i for all *i* are either 0 or 1. It is 0 when none of the instances belong to the specific class, and it is 1 when all instances belong to the class. When *m*th node is pure, the entropy function will get a small value, near zero. This means that a node contains mainly observations from a single class. Another function that measures the purity of a node is the Gini index and it can be defined by

$$G = \sum_{i=1}^{K} p_m^i (1 - p_m^i) \qquad (12)$$

which is a measure of total variance across all the K classes. The Gini index is numerically quite similar to entropy function. When all p_m^i are close to zero or one, the Gini index will have a small value as well, meaning that a node contains predominantly observations from a single class (Alpaydin et al. 2014, 216; Webb & Copsey 2011, 328–329; James et al. 2013, 312).

Either of the functions, entropy or Gini index, can be used when building a classification tree. Each node is split until pure nodes are achieved. Pure nodes contain just observations of a single class and therefore these nodes will not be split any further and they will form the leaf nodes of the tree. The tree will grow in a way that it maximizes the purity of the decision nodes relative to their respective parent nodes (Basak, Kar, Saha, Khaidem and Dey 2019).

Tree-based models are simple and easy to interpret. The hierarchical structure of the decisions allows a fast categorization of different inputs and does not require much computation. For instance, if the decisions are binary, having just two possible outcomes in each section of decision node, in the best situation, each decision node eliminates half of the cases. The decision tree is also easy to understand because it can be transformed to a set of if-then rules and therefore can be seen as closely mirroring human decision-making process. The model handles qualitative predictor variables well without the need to create separate dummy variables. Dummy variables can take just values 0 or 1, which indicate the absence or presence of some categorical effect that might occur and affect the outcome (Alpaydin et al. 2014, 215; James et al. 2013, 315; Sung, Chang and Lee 1999).

For these reasons, the decision tree is a popular choice when implementing machine learning and analyzing data. However, as it is easy to interpret and a fast method to implement, it is often not as accurate as the other machine learning models such as SVM. Still, it is a good choice for classification tasks or predicting outcomes (Tseng 2007; Sung et al. 1999). Decision tree models can also be enhanced by using several techniques like random forest, which will be covered next.

2.5.4 Random forest

Random forest is a popular machine learning algorithm because it is non-parametric classifier as well as the decision tree. The model does not require any learning parameters to be determined. Therefore, it does not require any assumptions of prior distributions. A random forest algorithm can solve a wide range of classification problems. The method is based on decision tree modeling where the algorithm uses an ensemble of many decision trees to reduce the effect of overfitting (Basak et al. 2019; Shalev-Shwartz & Ben-David 2014, 255). In other words, random forest grows multiple decision trees and combines them together to achieve more accurate and stable predictions. This process leads to significant improvement of the classification accuracy compared to just one decision tree (Breiman 2001).

In a random forest, each tree will be grown based on a random subset of the feature space. The feature space encompasses all the variables that are in a given data set. Random forest method divides the feature space M into small subsets of features $m = \sqrt{M}$, which will be selected randomly to grow each tree (Basak et al. 2019). This process reduces the effect of overfitting by decreasing the variance because the trees themselves are considering just a small subset of the features and each tree is a little bit different subset. Therefore, if one of the predictors has a strong effect on the dependent variable, it will not be included into all of the constructed trees. The constructed trees are more reliable because they are not as highly correlated between each other (James et al. 2013, 320; Hastie, Tibshirani & Friedman 2003, 588).

2.5.5 K-Nearest Neighbors (KNN)

The k-nearest neighbors (KNN) algorithm is also a non-parametric method, which can be used for classification and regression problems. Like decision tree or random forest, the

KNN algorithm does not make any assumptions about the underlying data. This is a useful feature because the real-world data does not follow any theoretical assumptions, for example, the data cannot be usually linearly separated nor it is uniformly distributed. The algorithm needs only the following information: the number of neighbors, the distance metric to use, and the training data set. (Batrinca, Hesse, Treleaven 2017; Webb et al. 2011, 152).

The KNN algorithm divides new observations into different groups based on their distance from the other observations. The distance can be calculated by using different functions, but the most common function used is the Euclidean distance function. The Euclidean distance can be defined by

$$D = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{13}$$

The Euclidean distance is a straight line between the two different observations in the feature space. If the observations have similar values, the Euclidean distance between the observations is a small value and vice versa (Shalev-Shwartz et al. 2014, 258; Batrinca et al. 2017). The KNN model attempts to estimate the conditional distribution of the dependent variable Y, and then classifies each observation to specific class with highest estimated probability. The model starts the classification by identifying K points (K is a positive integer, indicating the number of neighbors) in the training data set which are the closest to a test observation x_0, represented by $\mathcal{N}_0$. Then, the conditional probability for class *j* is estimated which is basically a fraction of the points in $\mathcal{N}_0$ where their values of the dependent variable equals *j*. Finally, the test observation x_0 is assigned to the class where it has the largest probability. The KNN method can be defined mathematically in following way (James et al. 2013, 39).

$$\Pr(Y = j | X = x_0) = \frac{1}{K}\sum_{i\in\mathcal{N}_0} I(y_i = j) \tag{14}$$

The KNN algorithm is often referred as “lazy learning” because it postpones its calculations until the part of classification. The algorithm does not use the training data observations to make any generalizations. Overall, the method is quite simple and one of the simplest machine learning algorithms, but despite of that, it can produce classifiers that have quite high accuracy. The accuracy of the KNN can also vary significantly if there are noisy or irrelevant features included in the model. Also, the chosen number of neighbors, K, affect highly on the classification performance of the model because it controls the bias-variance trade-off (Imandoust & Bolandraftar 2013; Martínez, Frías, Pérez & Rivera 2017).

The most basic version of KNN model is 1-nearest neighbor model where K = 1. This model has often low bias, but high variance. Meaning that the model is able to capture important relationships between features and the dependent variable, but when changes occur in the given data set the results vary a lot. The decision boundary of the model is non-linear and therefore overly flexible. In this case, the model is overfitting the data and producing results that will vary a lot between different data sets. That is why this model will not be reliable when predicting the future data points. As the value of K is increased, the models will become less flexible because its decision boundary will become closer to linear. This model will give results that are more stable as its variance is lower, but its bias is higher. This means that there will be some important relationships unrevealed (Batrinca et al. 2017; James et al 2013, 39–40; Imandoust et al. 2013).

The number of neighbors, K, is usually chosen empirically where different numbers of K values are tested, and the results are compared to each other. The K value, which gives the highest accuracy is chosen to define the classifier. There are also some advanced methods for choosing the right K value. For instance, Hassanat, Abbadi and Altarweneh (2014) have proposed a method in which a weak KNN classifier is used each time with different K value. They started with the K value of 1 and continued to value of the squared root of the size of the training set. These results from the weak classifiers are then combined using the weighted sum rule. Their results indicated that this method is competitive with other traditional KNN classifiers and can even outperform them in some cases.

In this study, a number of different neighbors are tested to examine which one of them provided the best accuracy. Figure 7 presents two different KNN models with different decision boundaries. The first model has K = 1 and the other one has K = 25. The decision boundary will resemble more linear as the number of neighbors in the model is increased.

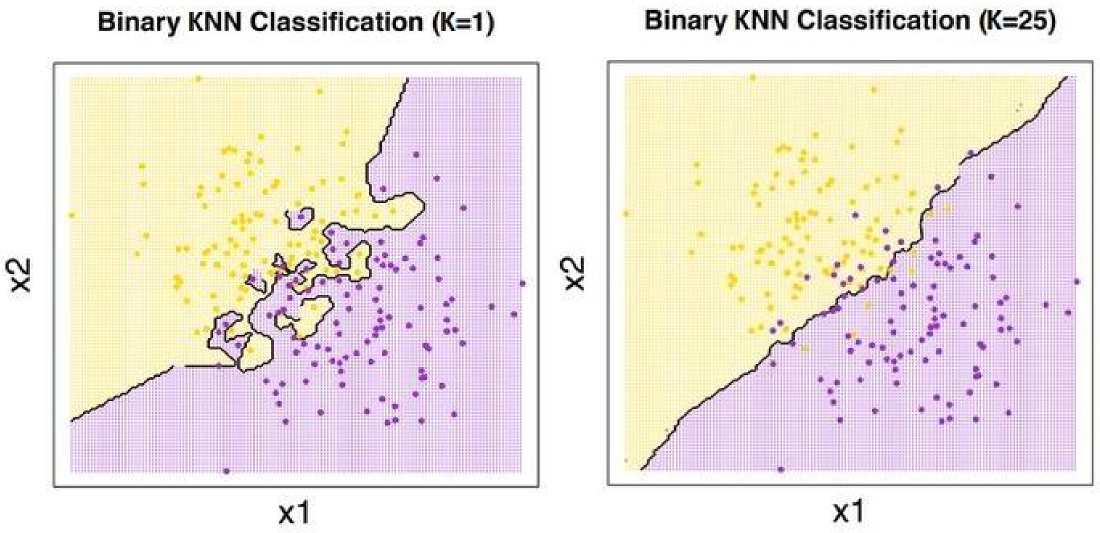


Figure 7 KNN models with different values of K (James et al. 2013, 41)

KNN method has many advantages and therefore it is also a popular choice for implementing machine learning. The method is simple, effective, and robust which can produce good results even when noisy training data is used. It is more effective when the training data is large. However, the method has also some disadvantages such as sensitivity and poor run-time. The KNN is sensitive to irrelevant and redundant features because the features do not provide any information which could be useful for classification. It has also long run-time when the training data is large because each distance between training samples have to be calculated (Imandoust & Bolandraftar 2013).

2.6 Building the hypotheses

This chapter presents the hypotheses of the study. In the study, there are two hypotheses that will be tested based on previous studies. Both hypotheses concern the machine learning models used in this study and the efficiency of the Finnish financial markets. Based on previous literature, the best model for predicting equity price movement has often been a random forest model and these results have pointed out that there exists positive evidence for predicting the equity price movement (Milosevic 2016; Ballings et al. 2015). The first hypothesis focuses on how well the machine learning models will perform on the data from the Finnish financial markets since there have not been earlier studies on this matter. The following main hypothesis is as follows:

- Hypothesis 1 (H1): By applying machine learning models and training them on the past data, it is possible to predict the equity price movement of the stocks listed on the OMXH

Similar hypothesis was tested by Milosevic (2016) for various stocks listed in different indices such as S&P 1000, FTSE 100 and S&P Europe 350. He concluded that it is on some level possible to predict what the performance of the stocks will be in the future based on only information about the financial ratios of companies. Milosevic (2016) reported the highest prediction accuracy of 75.1%. Also, Dutta et al. (2012) indicated similar results in their study.

In this study, the hypothesis will be tested for the Finnish financial markets for which it has not been tested before. It provides more information about the efficiency of the Finnish financial market. Especially, whether the market holds the weak-form or semi-strong form efficiency requirements. It will be tested whether past data about the financial ratios of the companies listed on the OMXH can be used for predicting their future performance. The second hypothesis of the study is the following one.

- Hypothesis 2 (H2): Macroeconomic variables will increase the prediction accuracy of the machine learning models compared to the same models, which do not contain the variables

The idea behind the second hypothesis is to examine on a deeper level, whether the macroeconomic variables of Finland could increase the prediction accuracy of the machine learning models. In the finance literature, companies face risks that can be divided into two major parts: market and company specific risks. These are also known as systematic or aggregate risk and unsystematic risk. Systematic risk refers to a market risk that affect all securities at the same time. This kind of risk cannot be diversified. Unsystematic risk refers to a risk, which is diversifiable because it is specific for the company which investors have invested. Investors can diversify the unsystematic risk by investing multiple companies instead of just one (Elosegui 2003).

In theory, the models, which contain only financial ratios should not be able to explain all the price movements of the stocks because of the systematic risk and other factors and risks which the stocks encounter. By including macroeconomic variables in the models, systematic risk can be examined. The macroeconomic variables should provide new information for the models, which cannot be explained by only using company specific information. This should enhance the estimates of the equity price movement and increase the prediction accuracy of the models which include also macroeconomic variables. The second hypothesis has not been tested before in the previous studies. But, for instance, Ballings et al. (2015) have also included macroeconomic variables in machine learning models to forecast equity price movement. However, Ballings et al. (2015) did not examine what kind of effects the macroeconomic variables had on the prediction accuracy of the models, for example, could they have increased the accuracy.

3 DATA, METHODOLOGY AND METHODS

In this chapter, the detailed description of the data and different variables used in the research will be presented. The chapter will begin by discussing the data and explaining the data preparation process, which is required in order to utilize the machine learning models correctly. Then, the chapter moves on to describing the different variables: how they are calculated and descriptive statistics of them. The variables used in the models can be divided into two main categories: financial ratios of companies and macroeconomic variables.

3.1 Data

The data for the research is gathered from Thomson Reuters Datastream. It contains quarterly closing prices of stocks listed on the OMX Helsinki Stock Exchange as well as all relevant quarterly financial ratios of the firms. The data contains also different macroeconomic variables of Finland, which are collected from the Datastream as well.

The data includes all non-financial companies which have been listed on the OMX Helsinki Stock Exchange from 2000 to 2018. All companies must have their headquarters in Finland or otherwise the companies are excluded from the data set. All financial sector companies have been excluded because they do not have all the same financial ratios available as the other companies have. For instance, current and quick ratios are not available for the financial companies. The whole dataset contains 73 different companies and their quarterly financial ratios from 2000 to 2018. The data is in panel format and contains also information of macroeconomic variables from the same time period.

The time period of 2000–2018 was selected because it includes a major bull and bear market movements, for example, the dot-com bubble in the early 2000s and the financial crisis of 2007–2008. Therefore, it is interesting to find out, how well the machine learning models are able to capture the major market movements. The data is also long enough, have enough data points, for the purposes of machine learning and making reliable predictions. The final dataset has 5475 data rows and over 70 variables, but only the most valuable variables, which explain the equity price movement best, were selected to the final models.

3.2 Data preparation

Before the different machine learning models can utilize the data, it has to be prepared and cleansed first. This process is important as well as the model construction part. Both of these processes will have an impact on the final results of the models and therefore it has to be done correctly in order to achieve valid results.

The data from Datastream contained some missing values. Which is why first, all these missing values were labeled as -99 999. The models will read these values as outliers and therefore these values would not have a major effect on the models and the predictions. This is necessary in order to keep the data in a panel data format and not having to narrow down the data set any further. For instance, if all missing values are dropped and just one company is missing the P/E value for the year 2000, all the other companies that have the P/E values available have to be dropped out as well from the data set in order to keep it in a balanced panel format. Therefore, missing values are labeled and not dropped. Also, Milosevic (2016) did the same process for the data used in his research. Python programming language was used to construct the whole data set and formatting it into the panel form. Python was also utilized in the data analysis and machine learning part of the study: constructing statistical models, making predictions, and receiving and evaluating the results.

The categorical feature, such as a company name in this case, was encoded. This means that all company names are converted to whole numbers starting from 0 up to 72. For example, in this study Afarak will be assigned 0 because it is the first company and Amer Sports is labeled as 1 and so on. After this part, all the numbers are shifted to separate columns, in this case, each company have its own column. Instead of one company name column, there are now 72 new columns for each company. Finally, all these columns are transformed into dummy variables were 1 means that the specific row information concerns the company and 0 that the information in the specific row does not concern the company. This must be done because the company names themselves do not have any hierarchical meaning, for instance, company name of Nokia is not bigger than company name of Elisa. If the encoded values are not transformed into dummy variables, the models will evaluate company name, which was assigned number 4 more relevant than company, which was assigned number 3 in the encoding part. The encoding was done in Python by utilizing an OneHotEncoder method from sklearn package. At the end of the encoding phase, one of the transformed columns has to be dropped out in order to avoid a dummy variable trap.

The dummy variable trap is also known as the perfect collinearity, where two or more independent variables are perfectly correlated. One variable can be predicted from the other variables. The dummy variable trap occurs when the same number of categories is transformed into the same number of dummy variables. It can be avoided by excluding one of the categorical variables in the model (Bech & Gyrd-Hansen 2005). For instance, in this study there are 73 different companies and therefore there are 72 different categorical variables. One of these variables has to be excluded to avoid the perfect collinearity and it does not matter which one of the variables is dropped.

Feature scaling is also needed to standardize the range of all the independent variables in the study. This process is also known as data normalization. Feature scaling is important because some machine learning algorithms will not work properly if the data is not standardized. For instance, many classifiers such as SVM or KNN calculate the distance of two different data points using the Euclidean distance. Therefore, if one of the features has a wider range of values, it will dominate the other feature, and this will lead to false results (Young & Jeong 2009; Bo, Wang & Jiao 2006). In the study, all variables were scaled by using a StandardScaler method in Python from sklearn package.

After the data preparation was done, the data was divided into training and testing data sets. First, the models are trained with the training data set and then the trained models try to predict the dependent variable in the test data set. The test data set contains new information that the models have not encountered before in the training phase. The training data set includes all the values from 1/2000 to 12/2015. The last two and half years, 1/2016–7/2018, of the whole-time period are selected in the test set for prediction purposes, instead of randomly picked values from the whole sample time frame. If the values are randomly picked from the whole-time frame, all the models will contain look-ahead biases.

Look-ahead bias occurs when the models have access to information, which would not be available and known during the period when it is analyzed (Daniel, Sornette & Woehrmann 2009). For instance, when predicting the values of the year 2002, the model already has information on some of the next and its following year values, which would not be normally known. Look-ahead bias increases accuracy of models and would lead to false and biased results. Due to this, the data is separated into training and testing sets based on the year and not a random selection.

3.3 Variables used in the models

In the research, all of the models contain the same independent variables. First, models with only financial ratios are tested and the predictions are made. Then, macroeconomic variables are included to compare the prediction results. Company specific variables are market capitalization (MC), Price-to-book ratio (PBR), Price-earnings ratio (PER), quick ratio (QR), earnings before interest and taxes (EBIT), earnings per share (EPS), dividend per share (DPS), dividend yield (DY), close price of a stock (CPS), return on invested capital (ROIC), total debt divided by total capital (TDTC). The macroeconomic variables of Finland are unemployment rate (UR), gross domestic product (GDP), and expected inflation for the next 6 months (EXPINF6M). In table 2, all the variables and their abbreviations are listed.

Table 2 Variables used in the models

Name of the variable	***Description of the variable***
MC	Market capitalization of a firm
PBR	Price-to-book ratio
PER	Price-earnings ratio
QR	Quick ratio
EBIT	Earnings before interest and taxes
EPS	Earnings per share
DPS	Dividend per share
DY	Dividend yield
CPS	Close price of a stock
ROIC	Return on invested capital
TDTC	Total debt to total capital
UR	Unemployment rate
GDP	Gross domestic product
EXPINF6M	Expected inflation for the next 6 months
EPM	Equity price movement (dependent variable)

Financial ratios are calculated from financial statements of companies by dividing two numerical values. These ratios describe company specific information (Goel 2016, 3). There are some statistical features that financial ratios point out to have. For instance, financial ratios are not normally distributed or their dispersion can be large. Occasionally, these both features might occur. Also, one problem relates to the collinearity of financial ratios. Since, many financial ratios are calculated by using the same factors in the equations, collinearity will occur on some level already. Some items in accounting statements also tend to move in the same direction which increases the correlation between the ratios (Horrigan 1965).

MC is the market value of a publicly traded company’s shares. It is calculated by multiplying the share price by the number of ordinary shares in issue. In Datastream, the issue amount is updated when new shares are issued or after a capital change. Jaffe and Westerfield (1989) reported that the size of a company has a significant effect on stock return. Smaller companies tend to be more profitable for investors. PBR measures a company’s current market price to its book value. It is calculated by taking the closing price of a share and dividing it by the book value per share. PER is calculated by dividing the price of a share by the earnings per share (Datastream 2018). The ratio gives a general idea of the quality of corporate earnings. If the value is lower than what the industry average value is, it indicates that the future earnings of the company are expected to be lower and vice versa (Bragg 2012, 145). Jaffe and Westerfield (1989) also reported that earnings-price ratio, which is the inverse of the price-earnings ratio, had also a significant effect on stock returns. The higher the earnings-price ratio was the higher the returns tend to be.

QR gives investors information about the short-term liquidity position of a company. It measures how well a company is able to meet its short-term obligations with its most liquid assets (Bragg 2012, 82). The ratio is calculated by adding cash and equivalents with receivables and then dividing the sum by current liabilities. EBIT represents the earnings of a company before interest expense and taxes. It is calculated by adding back interest expense on debt to the pre-tax income and subtracting interest capitalized from it. EPS is an indicator of the profitability of a company. The value is calculated by dividing the earnings of a company by the number of issued shares (Datastream 2018).

DPS is calculated by dividing the total amount of dividends paid over an entire year by the number of issued shares. In Datastream, DPS ratio is calculated on a rolling 12-month basis. It is intended to represent the anticipated payment over the following 12 months (Datastream 2018). DY measures how much a company distributes its profits through dividends each year relative to its share price (Bragg 2012, 133). In Datastream, dividend yield is calculated on gross dividends, which include tax credits. ROIC measures how well a company is using its money to generate returns. It is calculated using the following formula (Datastream 2018):

$$\frac{NI - BL + (IED - IC) * (1 - T)}{TC + SD\&LD} * 100 \qquad (15)$$

where;
NI = net income
BL = bottom line
IED = interest expense on debt
IC = interest capitalized
T = tax rate
TC = total capital (average of last year’s and current year’s)
SD&LD = short-term debt & current portion of long-term debt

TDTC measures the capital structure, financial solvency and the degree of leverage of a company. It is calculated using the following formula (Datastream 2018):

$$\frac{LD + SD\&LD}{TC + SD\&LD} * 100 \qquad (16)$$

where;
LD = long-term debt

TC = total capital
SD&LD = short term debt & current portion of long-term debt

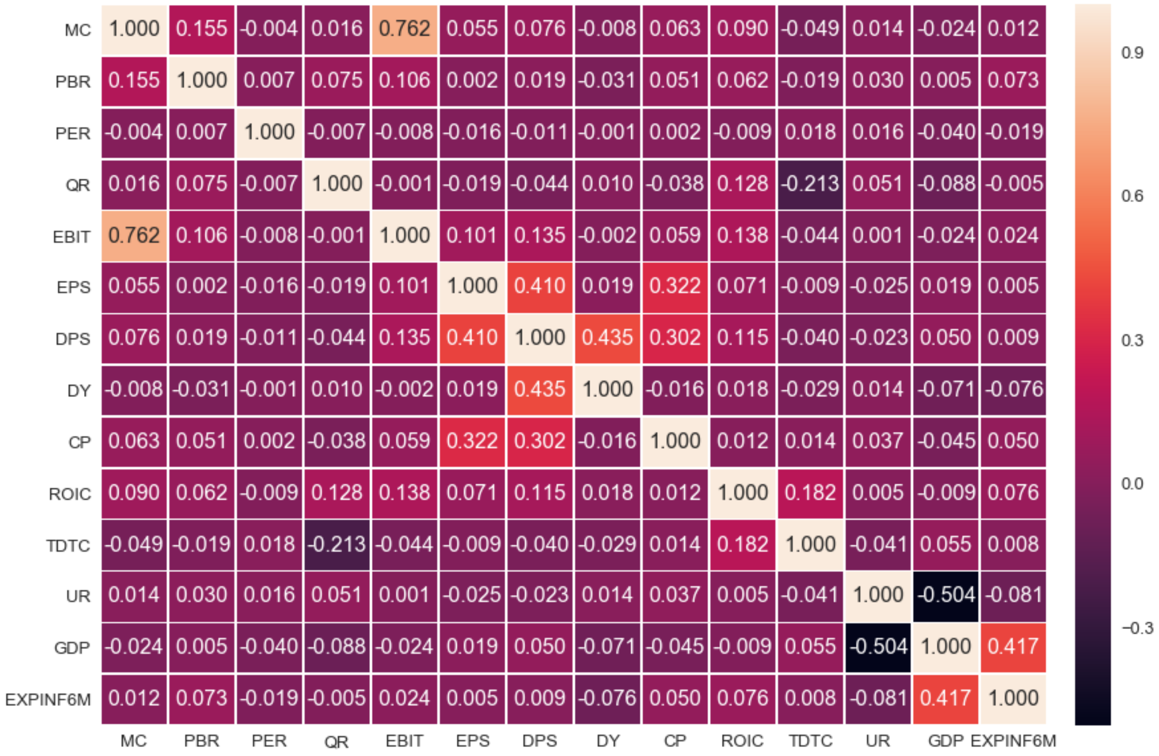
Macroeconomic variables evaluate the performance of the whole economy. In this case, the economy of Finland. UR is the number of unemployed people as a percentage of the labor force. The labor force consists of people whose age are between 15 to 74. In Finland, the unemployment rate sharply increased in the financial crisis of 2007–2008 (Official Statistics of Finland). In this study, GDP of Finland is measured quarterly. It is a monetary measure of the market value of all the final goods and services produced in Finland. EXPINF6M variable is measured quarterly. The information is gathered using a survey provided by CESifo Group Munich (Datastream 2018).

By including the macroeconomic variables of Finland in the models, they could indicate how well the economy of Finland is functioning and could also signal the future direction of the economy. If the economy is facing hard time, for instance, the unemployment rate is increasing, and the GDP of Finland is decreasing, it will also have an effect on the stock prices in the long run. Carcía-Ferrer and Bujosa-Brun (2000) forecasted the OECD industrial turning points. They pointed out that they were able to forecast the industrial turning points favorably with method that contained two-stage decision process. The method included anticipation of that a turning point is likely to occur and confirmation that the turning point will occur. Carcía-Ferrer and Bujosa-Brun (2000) included also survey data which improved the turning point forecasts. Ballings et al. (2015) included also GDP and unemployment variables in their machine learning models to predict equity price movement.

In this study, the predicted variable or dependent variable in the machine learning models is EPM, which contains either value 1 or 0. The variable is measured by calculating the yearly percentage change in each company's stock price. If the stock price has increased by 10% or more in a year, the EPM variable will be labeled as 1, otherwise as 0. The return of 10% has been selected as a benchmark, because Milosevic (2016) used 10% return as a benchmark in his research when predicting equity price movement. Therefore, it will be easier to compare the results to Milosevic (2016) research. If the individual stock has had return that is 10% or more, it is considered then as a "good" investment otherwise it is considered as a "poor" investment. Ferson and Harvey (1993) examined the risk and predictability of international equity returns. They reported that an asset pricing model was able to capture much of the predictability of the equity returns. Ferson and Harvey (1993) found out that time-varying risk premia was the largest factor which had an effect on the predictability of returns.

Variables were selected into the models first by including the same variables as Milosevic (2016) and Dutta et al. (2012) used in their research. Then, other financial ratios and macroeconomic variables were included and tested over multiple times to examine if they could increase the prediction accuracy of the models. For example, ROIC and TDTC were added, because they increased the prediction accuracy. Also, some of the variables were removed such as current ratio, because it did not increase the prediction accuracy. The independent variables affect greatly on the accuracy of the models and even small variable changes in the models can have a major impact on the accuracy level. The correlation table of the included variables is presented in table 3.

Table 3 Correlation matrix



	MC	PBR	PER	QR	EBIT	EPS	DPS	DY	CP	ROIC	TDTC	UR	GDP	EXPINF6M
MC	1.000	0.155	-0.004	0.016	0.762	0.055	0.076	-0.008	0.063	0.090	-0.049	0.014	-0.024	0.012
PBR	0.155	1.000	0.007	0.075	0.106	0.002	0.019	-0.031	0.051	0.062	-0.019	0.030	0.005	0.073
PER	-0.004	0.007	1.000	-0.007	-0.008	-0.016	-0.011	-0.001	0.002	-0.009	0.018	0.016	-0.040	-0.019
QR	0.016	0.075	-0.007	1.000	-0.001	-0.019	-0.044	0.010	-0.038	0.128	-0.213	0.051	-0.088	-0.005
EBIT	0.762	0.106	-0.008	-0.001	1.000	0.101	0.135	-0.002	0.059	0.138	-0.044	0.001	-0.024	0.024
EPS	0.055	0.002	-0.016	-0.019	0.101	1.000	0.410	0.019	0.322	0.071	-0.009	-0.025	0.019	0.005
DPS	0.076	0.019	-0.011	-0.044	0.135	0.410	1.000	0.435	0.302	0.115	-0.040	-0.023	0.050	0.009
DY	-0.008	-0.031	-0.001	0.010	-0.002	0.019	0.435	1.000	-0.016	0.018	-0.029	0.014	-0.071	-0.076
CP	0.063	0.051	0.002	-0.038	0.059	0.322	0.302	-0.016	1.000	0.012	0.014	0.037	-0.045	0.050
ROIC	0.090	0.062	-0.009	0.128	0.138	0.071	0.115	0.018	0.012	1.000	0.182	0.005	-0.009	0.076
TDTC	-0.049	-0.019	0.018	-0.213	-0.044	-0.009	-0.040	-0.029	0.014	0.182	1.000	-0.041	0.055	0.008
UR	0.014	0.030	0.016	0.051	0.001	-0.025	-0.023	0.014	0.037	0.005	-0.041	1.000	-0.504	-0.081
GDP	-0.024	0.005	-0.040	-0.088	-0.024	0.019	0.050	-0.071	-0.045	-0.009	0.055	-0.504	1.000	0.417
EXPINF6M	0.012	0.073	-0.019	-0.005	0.024	0.005	0.009	-0.076	0.050	0.076	0.008	-0.081	0.417	1.000

The correlation values are calculated before the missing values were converted into -99 999 values. The correlation table indicates that EBIT and MC have the highest correlation from of the other variables, positive correlation of 0.76. This makes sense because when the earnings of a company grow the market value of the company will be higher or vice versa. Also, EPS, DPS, and DY have a high correlation between each other. These ratios measure quite the same thing, but from a little bit different aspect. UR and GDP have the lowest negative correlation -0.5 among all the other variables. Indicating generally that when GDP of Finland grows the unemployment of Finland diminishes and vice versa. The descriptive statistics of the variables are presented in table 4.

Table 4 Descriptive statistics of the variables

	Count	***Mean***	***Std***	***Min***	***Median***	***Max***
MC	5461	1850.87	9731.62	1.27	169.26	256972.50
PBR	5148	2.32	3.69	-17.98	1.74	83.33
PER	4106	52.55	1151.75	0.70	16.10	59500.00
QR	5116	1.17	1.16	0.08	0.89	17.64
EBIT	5164	138393	550531	-2380000	11713	8311000
EPS	5448	0.59	1.40	0.00	0.33	52.08
DPS	5461	0.35	0.49	0.00	0.22	7.44
DY	5461	3.62	10.21	0.00	3.21	611.82
CP	5461	11.09	31.04	0.02	6.53	1331.92
ROIC	5044	6.97	24.00	-303.49	7.76	449.84
TDTC	5152	32.37	37.21	-920.54	34.84	303.72
UR	5402	8.38	1.25	5.56	8.33	11.13
GDP	5475	45856.71	3077.52	39100.00	46760.00	50441.00
EXPINF6M	5475	14.13	47.65	-91.70	25.00	91.70

Also, the descriptive statistics are calculated before the missing values were converted to -99 999 values. The count value of the variables varies because there is different amount of missing values present for each variable. The missing values are not included in the count variable. PER variable has the lowest count value indicating that this variable has the most missing values compared to the other variables. Next, it will be examined, how well the variables can predict the equity price movement and what kind of results they can provide.

4 EMPIRICAL ANALYSIS AND RESULTS

In this chapter, the empirical results are presented and discussed. Also, the results will be compared to prior academic findings. There are two kinds of models per each machine learning algorithm. The first model contains only company specific information, the chosen financial ratios. The second model includes also macroeconomic variables. These two kinds of models will be compared to each other to examine the effects of macroeconomic variables on the equity price movement.

First, the results of the models that include only financial ratios are presented. Then, the results are compared and analyzed to the models that include all the variables. The results of the models will be evaluated by using different performance metrics, which are confusion matrix, precision, recall, F1 score, the receiver operating characteristics (ROC) curve, and the area under the ROC curve (AUC). Finally, all the results are summarized and analyzed.

4.1 Model diagnostics

The research problem is a binary classification problem where outcomes are classified into two groups, either positive or negative. There are four types of outcomes that could occur when performing classification predictions; true positive, true negative, false positive, and false negative. True positive occurs when a model predicted correctly an observation belonging to the positive class. Similarly, true negative occurs when a model predicted correctly an observation belonging to the negative class. A false positive occurs when a model predicts incorrectly an observation belonging to the positive class. A false negative, on the other hand, occurs when a model predicts incorrectly an observation belonging to the negative class (Alpaydin et al. 2014, 561–562; Powers 2011).

In this study, the positive class is determined by the equities that have had return 10% or higher in a year and the negative class the equities that have had a return lower than 10% in a year. For instance, if a model predicts correctly that the return of a stock will be 10% or higher, the outcome is labeled as true positive. When a model correctly predicts that the price movement is lower that 10%, this outcome will be classified as true negative. A false positive outcome occurs when a model predicts incorrectly that the equity price will move up 10% or more, but in reality, the equity price did not move up that much. A false negative outcome occurs when a model incorrectly predicts that the equity price will not move up 10% or more, but in reality, the equity price moved up by that much.

Classification models can be evaluated using different metrics and methods that are derived from these four outcomes. The main metrics and methods that are presented here are confusion matrix, precision, recall, F1 score, classification accuracy, the receiver operating characteristics (ROC) curve, and the area under the ROC curve (AUC). A confusion matrix is a K × K table, where K is the number of classes. In this case, a two-by-two matrix because the dependent variable in this study has two classes: good and pour. Confusion matrix describes the complete performance of a machine learning model. The diagonal of the matrix presents all the outcomes which the model has classified correctly. The top-right corner represents usually the outcomes of false positives and the bottom left corner usually the outcomes of false negatives. Confusion matrix is a convenient way to display the information about how the model classified all the outcomes. How many of the observations were correctly classified and how many of them were incorrectly classified. It forms the basis for the other metrics (Alpaydin et al. 2014, 564; Berthold, Borgelt, Höppner & Klawonn 2010, 99).

Precision is the number of correctly classified positive cases divided by the total number of cases, which the model has labeled to the positive class. In this case, precision states the following: all the stocks that have been classified into the positive class (return 10% or over), how many of them have actually had a return 10% or higher. Precision can have values between 0 and 1, where the best value is 1 and the worst value is 0. High precision relates to low false positive rate. The measure is calculated in the following way (Alpaydin et al. 2014, 562–564; Powers 2011).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{17}$$

Recall also known as sensitivity is the proportion of correctly classified positive cases divided by the number of all samples, which should have been identified as positive. Recall states the following: all the stocks that actually have 10% or higher return, how many of them are classified as 10% or higher. Recall can also have values between 0 and 1, where the value 1 is the best and the worst value is 0. The measure can be calculated using the following formula (Alpaydin et al. 2014, 562–564; Powers 2011).

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{18}$$

F1 score, also known as F-measure or F-score is a harmonic mean of precision and recall. It is used to measure a test's accuracy. F1 score explains how precise and robust the classifier is. It can have values between 0 and 1, where 1 is the best value and 0 is the worst value (Powers 2011).

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{19}$$

In the study, separate metric for the classification accuracy is also calculated to examine the differences between F1 score and the classification accuracy. Classification accuracy is the number of correctly predicted outcomes divided by the total number of predictions (Alpaydin et al. 2014, 562). Accuracy is a good measure when the target variable classes in the data are nearly balanced. But when the target variable classes in the data are unbalanced, where most of the target variable classes are in one class, the measure performs poorly and is misleading. In this study, the target classes in the data are quite balanced.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + False\ Negatives + True\ Positives} \tag{20}$$

Performance of a model can be also displayed by the receiver operating characteristics (ROC) curve. The ROC curve is a graphical plot that simultaneously presents the two types of errors of a classification model for all possible classification thresholds. Therefore, it is also useful for comparing different models to each other. The ROC is a curve of probability. The two types of errors are true positive rate, also known as sensitivity, and false positive rate also known as specificity. True positive rate is the proportion of positive data points that a model has correctly classified as positive divided by all positive data points.

$$True\ Positive\ Rate = \frac{True\ Positive}{False\ Negative + True\ Positive} \tag{21}$$

False positive rate is the proportion of negative points that a model has mistakenly labeled as positive divided by all negative data points. Both false positive and true positive rate can have values between 0 and 1. False positive rate can be calculated in the following way (Alpaydin et al. 2014, 561–562; Powers 2011).

$$False\ Positive\ Rate = \frac{False\ Positive}{False\ Positive + True\ Negative} \tag{22}$$

The area under the ROC curve (AUC) measures the entire area under the ROC curve. It is the overall performance of a classifier, an aggregate measure of performance across all possible classification thresholds. AUC of a classifier is the probability that the classifier will rank a randomly chosen positive observation higher than a randomly chosen negative observation. AUC can have values between 0 and 1, and the best classifier has the largest AUC value. The best ROC curve will reach to as close to the top left corner as possible, where the true positive rate is 1 and the false positive rate is 0 (Alpaydin et al. 2014, 562–563; Berthold et al. 2010, 98).

4.1.1 Logistic regression

The equation of the logistic regression model is estimated using the maximum likelihood estimation. It includes all the dependent variables and can be mathematically written as

$$
\begin{aligned}
Z = {} & -0.5605 + 0.0 \ldots * MC + 0.0 \ldots * PBR + 0.0 \ldots * PER + 0.0 \ldots * QR + 0.0 \ldots \\
& * EBIT + 0.0095 * EPS + 0.2478 * DPS - 0.1033 * DY - 0.0038 \\
& * CP + 0.0 \ldots * ROIC + 0.0 \ldots * TDTC + 0.0 \ldots * UR + 0.0 \ldots * GDP \\
& + 0.0136 * EXPINF6M
\end{aligned}
$$

where
Z = log(p1-p), p is the probability of the equity price movement being 10% or higher

In the equation some coefficients have values near zero. Meaning that their effect on the dependent variable is small. However, in the logistic regression model, 9 out of 14 independent variables are statistically significant. The p-values of the nine significant variables are lower than 0.05. Eight of these variables are financial ratios and one of the variables is macroeconomic variable. The significant variables are MC, PBR, PER, EBIT, DPS, DY, ROIC, TDTC and EXPINF6M. All the independent variables have statistically significant effect on the dependent variable, equity price movement. The summary results of the logistic regression model are presented in table 5. The results of the logistic regression, which includes only financial ratios can be found in the appendix 3.

Table 5 Summary of logistic regression with all independent variables

Summary of logistic regression						
Date: 2018-12-13			Pseudo R-squared: 0.105			
No. Observations: 5475			AIC: 6624.5752			
Dependent variable: EPM			BIC: 6723.6944			
Df Model: 14			Log-Likelihood: -3297.3			
Df Residuals: 5460			LL-Null: -3684.7			
Converged: 1.0000			LLR p-value: 2.5750e-156			
No. Iterations: 17.0000			Scale: 1.0000			
Variable	**Coef.**	**Std.Err.**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-0.5605	0.5175	-1.0831	0.2788	-1.5747	0.4538
MC	0.0000	0.0000	-3.4410	0.0006	-0.0001	0.0000
PBR	0.0000	0.0000	-3.0321	0.0024	0.0000	0.0000
PER	0.0000	0.0000	12.1384	0.0000	0.0000	0.0000
QR	0.0000	0.0000	0.4773	0.6331	0.0000	0.0000
EBIT	0.0000	0.0000	3.8330	0.0001	0.0000	0.0000
EPS	0.0095	0.0242	0.3913	0.6955	-0.0379	0.0568
DPS	0.2478	0.0993	2.4948	0.0126	0.0531	0.4425
DY	-0.1033	0.0153	-6.7522	0.0000	-0.1333	-0.0733
CP	-0.0038	0.0020	-1.8571	0.0633	-0.0077	0.0002
ROIC	0.0000	0.0000	2.7750	0.0055	0.0000	0.0000
TDTC	0.0000	0.0000	2.8468	0.0044	0.0000	0.0000
UR	0.0000	0.0000	0.9723	0.3309	0.0000	0.0000
GDP	0.0000	0.0000	1.4279	0.1533	0.0000	0.0000
EXPINF6M	0.0136	0.0008	17.5253	0.0000	0.0121	0.0151

All the variables, which are statistically significant, are highly statistically significant as well. Meaning that their p-values are lower than 0.001. The R^2 value is 0.105 which indicates that the logistic regression model explains 10.5% of the movement of the equity price variable. This result is higher compared to the logistic regression model, which contained only financial ratios as the independent variables in the model. The estimated coefficients of the independent variables are the log odds. They can be easily interpreted by converting them first to odds. In table 6, the log odds have been converted and the odds of the independent variables are presented.

Table 6 Odds ratios of the independent variables

Variables	***Odds***
Intercept	0.5709337
MC	0.9999673
PBR	0.9999880
PER	1.0000098
QR	1.0000015
EBIT	1.0000005
EPS	1.0095007
DPS	1.2812012
DY	0.9018173
CP	0.9962532
ROIC	1.0000057
TDTC	1.0000136
UR	1.0000110
GDP	1.0000046
EXPINF6M	1.0136755

Almost all of the independent variables have odds close to one except DPS variable, which has value significantly greater than one. This means that DPS variable is positively associated with the equity price movement. When dividend per share of a company is increased, it increases also the value of the company's stock. Surprisingly, DY variable has a value lower than one. It means that DY is negatively related to the equity price movement. For instance, when the dividend yield of a company increases, it has a negative impact on the value of the company stock on long run. This might be because the future cash flows of the company could be smaller in the future. However, the negative effect of the variable on the equity price movement is still small. The confusion matrices of the logistic regression models are presented in table 7. The first table presents the classification outcomes of the model that includes only financial ratios and the second table presents the outcomes of the model, which contains all the independent variables.

Table 7 Confusion matrices of logistic regression

Logistic regression with the financial ratios			
n=803	Predicted: No	Predicted: Yes	
Actual: No	386	55	441
Actual: Yes	301	61	362
	687	112	

Logistic regression with all the variables			
n=803	Predicted: No	Predicted: Yes	
Actual: No	191	250	441
Actual: Yes	100	262	362
	291	512	

The first logistic regression model classified a total of 447 (386 + 61) equities out of 803 correctly into the two classes. The first model had 301 false negative outcomes and 55 false positive outcomes. This indicates that the model predicted that these 301 stocks will not have a return of 10% or higher based on the information of the financial ratios. But in reality, the stocks achieved the benchmark return. For the 55 false positive cases, the model predicted that the stocks will have 10% or higher return based on the information about the financial ratios. But in reality, these stocks did not achieve the benchmark return.

The second logistic regression model classified 453 equities out of 803 correctly. For this model, the amount of the false negative outcomes was lower, but the amount of false positive outcomes was higher compared to the first model. There are 100 false negative outcomes and 250 false positive outcomes. The model predicted in 100 cases that the stocks will not achieve the benchmark return, when in reality the stocks reached the benchmark. In 250 cases, the model classified the stocks to the high return group, but in reality, these stocks did not have high returns.

The logistic regression with only financial ratios seems to predict most of the stocks into the negative class. Indicating that most of the stocks could not achieve 10% or higher return based on the information of all eleven financial ratios. When the macroeconomic variables are added into the logistic regression model, the outcomes of the predicted classes are more evenly distributed. The second model classified more outcomes of the equity price movement into the positive class than the first model. Overall, the second logistic regression model with all the independent variables points out to classify the equities better than the model, which contains only the financial ratios. The performance results of the two models are presented in table 8.

Table 8 Results of the logistic regression model

Independent variables in the model	***Precision***	***Recall***	***F1 score***	***Accuracy***
Only financial ratios	0.526	0.169	0.255	0.557
All variables	0.512	0.724	0.599	0.564

From the results in table 8, it can be concluded that the regression model, which has all the variables performed better and achieved better values for recall, F1 score and accuracy measures. Even though, the classification accuracy of the first model is 55.7%, the F1 score is extremely low, indicating that this model does not give reliable prediction results of the equity price movement. F1 score of the first model is low due to the low recall value. The accuracy and F1 score values differ highly because the number of false positives and false negatives are so different. The first model is picky and classifies many stocks into the negative class. Meaning that most of the stocks the models encounter will be classified as stocks, which will not reach return of 10% or higher. The second model performs much better than the first model. Even though, the classification accuracy value did not change much, the F1 score value is now much higher, reaching almost 60%.

Milosevic (2016) obtained F1 score of 63.3% for the logistic regression. This result is aligned with the result obtained in this study. The second model, which had the three macroeconomic variables, managed to come close to the same result that Milosevic (2016) achieved. In the logistic regression model, the three macroeconomic variables of Finland; UR, GDP and EXPINF6M increased significantly the F1 score of the model. Dutta et al. (2012) managed to get the accuracy of 74.6% for the logistic regression model. This is an especially high accuracy score and the logistic regression in this study did not manage to reach the same level. One reason is that the Finnish financial markets are more efficient than the emerging financial markets. Another reason concerns the quality of the data from the Datastream, which was not optimal. Many financial indicators were not available for the companies listed on the OMXH or some indicators did not contain values for each quarter. Instead for some financial indicators the values stayed at the same between each quarter and changed only on a yearly basis.

4.1.2 SVM

Radial kernel function was used for SVM models to fit them on the data. Other functions such as polynomial and sigmoid functions were also tested, but the radial kernel function performed the best for the given data set. There are two parameters that have an effect on the performance of the model. These two parameters are gamma and penalty parameter C. These parameters control the trade-off between bias and variance of the model (Scikit-learn). Different gamma and penalty parameter values were tested for both SVM models to analyze how much effect they had on the accuracy of the models.

Gamma values of 0.001, 0.005, 0.01, 0.03, 0.03, 0.04, 0.05 and 0.1 were tested for the SVM models. The first model contained only financial ratios as independent variables and the second model includes also macroeconomic variables. In figure 8, the effect of the gamma values to the prediction accuracy is visualized. All the tested gamma values for both SVM models are presented in the figure.

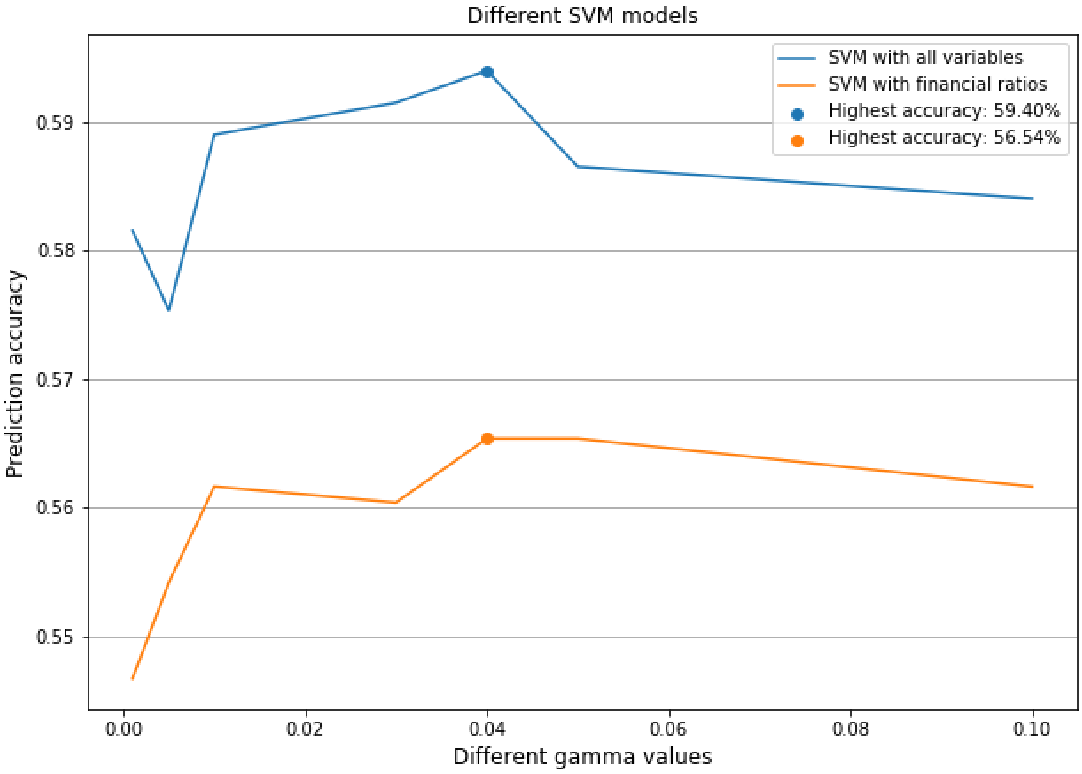


Figure 8 SVM models with different gamma values

Both SVM models achieved the highest accuracy with gamma value of 0.04. After the gamma value of 0.04, the prediction accuracy of the models started to decrease. The gamma value has the same kind of effect on both models. The effects of the penalty parameter values to the prediction accuracy are visualized in figure 9.

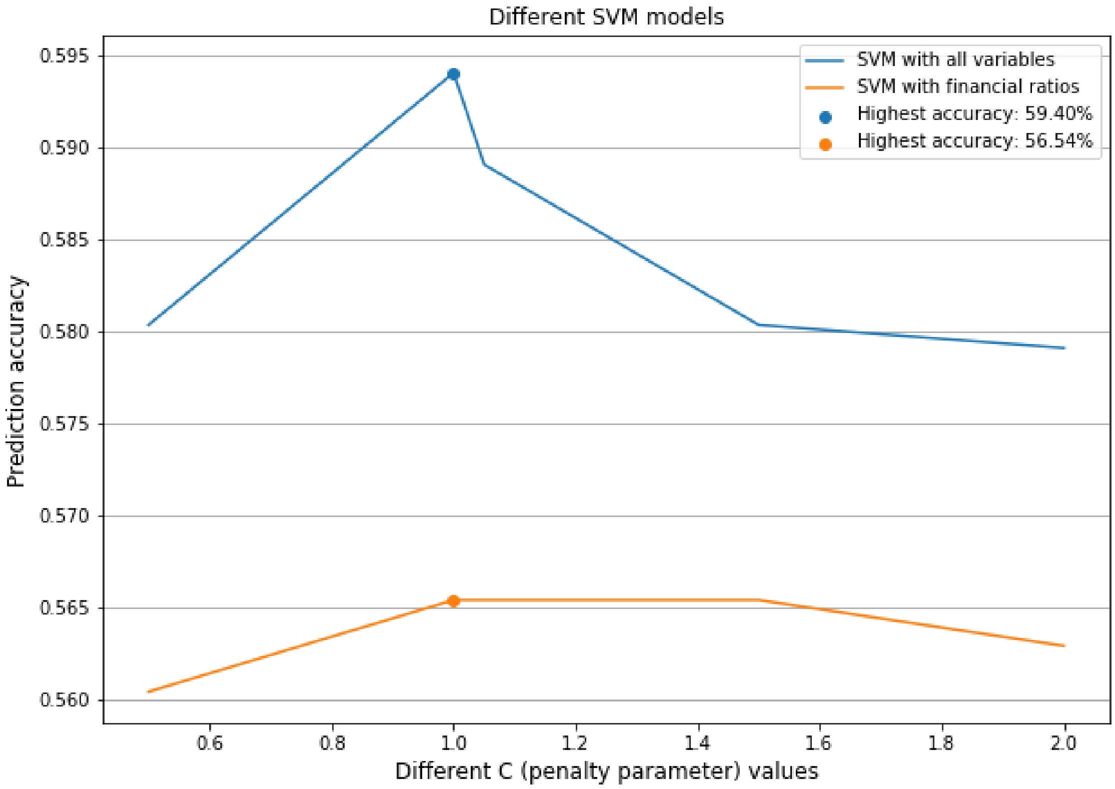


Figure 9 SVM models with different penalty parameter values

The penalty parameter value has much greater effect on the prediction accuracy than the gamma parameter. It had an even greater effect on the second model, which contained also the macroeconomic variables. First, the accuracy increases for both models and after the C value was 1 the best accuracy was reached. Larger C values seemed to decrease the accuracy of the models. The best SVM model, which included only financial ratios achieved classification accuracy of 56.54%, but the accuracy measured through F1 score was only 25.3%. The best SVM model which included also the macroeconomic variables had an accuracy of 59.40% and F1 score of 47.8%. In table 9, the confusion matrices of the both models are presented.

Table 9 Confusion matrices of SVM models

SVM with the financial ratios			
n=803	Predicted: No	Predicted: Yes	
Actual: No	395	46	
Actual: Yes	303	59	
	698	75	

SVM with all the variables			
n=803	Predicted: No	Predicted: Yes	
Actual: No	323	118	441
Actual: Yes	214	148	362
	537	266	

The first SVM model point out to predict most of the outcomes in the negative class and only some of the outcomes in the positive class. A total of 698 stocks were classified as low return stocks and only 75 of them were classified as high return stocks. The first model classified still around half of the outcomes correctly, 454 outcomes out of 803. There were many outcomes that point out to be false negative. The model predicted 303 stocks as not reaching to the benchmark return. But in reality, the stocks achieved the 10% benchmark return. The second SVM model with all the independent variables predicted outcomes little bit more also in the positive class. But most of the predictions are still made to the negative class. Both models are predicting that the stocks are not achieving the benchmark return based on the information of the financial ratios as well as the macroeconomic variables. Table 10 presents the results of both SVM models.

Table 10 Results of SVM models

Independent variables in the model	***Precision***	***Recall***	***F1 score***	***Accuracy***
Only financial variables	0.562	0.163	0.253	0.565
All variables	0.569	0.412	0.478	0.594

The first model obtained low accuracy measured by F1 score. This is because the recall value of the model is also extremely low. Most of the observations were classified into the negative class, lowering the recall score of the model. This model does not produce reliable prediction results for the equity price movement. The second model, which contained all the independent variables, performed better when measured with all the performance indicators. The F1 score value for the model was 47.8% and the classification accuracy was 59.4%. The macroeconomic variables point out to increase the accuracy of the model. Compared to the logistic regression model, the SVM model did not perform better. Milosevic (2016) reported F1 score of 62.4% for the SVM model which is much higher than what was achieved in this study. Shynkevich et al. (2017) achieved the highest prediction accuracy with the SVM model, which was 75.43%. This is also a much higher result than the prediction accuracy, which was reached in this research.

4.1.3 Decision tree

Decision tree models are the next models, which were tested. The first decision tree model contains only the financial ratios and the second model also includes the macroeconomic variables. Both models were constructed using the entropy function. Also, the Gini index function was tested, but it did not perform as well as the entropy function. Overall, the entropy function gave higher prediction results. In table 11, confusion matrices of the both models are presented.

Table 11 Confusion matrices of decision tree models

Decision tree with the financial ratios			
n=803	Predicted: No	Predicted: Yes	
Actual: No	267	174	441
Actual: Yes	180	182	362
	447	356	

Decision tree with all the variables			
n=803	Predicted: No	Predicted: Yes	
Actual: No	289	152	441
Actual: Yes	180	182	362
	469	334	

The outcomes of both models were quite evenly distributed between the two classes. The first model classifies 449 outcomes out of 803 correctly. The amounts of false positives and false negatives are quite same. The model had 174 false positive outcomes. These are the stocks, which the classifier predicts as high return stocks, but in reality, they provided low returns. The model had 180 false negative outcomes. These 180 stocks are classified as low return stocks, but they are actually high return stocks. The second model predicts 471 stocks correctly, which is higher than what the first model obtained. The amount of false positive outcomes is 152, which is lower compared to the first model. The amount of false negative outcomes is same as in the first model. The final results of the two decision tree models are presented in table 12.

Table 12 Results of the decision tree model

Independent variables in the models	***Precision***	***Recall***	***F1 score***	***Accuracy***
Only financial ratios	0.511	0.503	0.507	0.559
All variables	0.545	0.503	0.523	0.587

The first decision tree model, which included only financial ratios, reached the classification accuracy of 55.9% and the accuracy measured through F1 score was a little bit over 50%. The second model, which included also the macroeconomic variables achieved even better prediction accuracy. The classification accuracy was 58.7% and F1 score of the model was 52.3%. The macroeconomic variables point out to increase the prediction accuracy of the decision tree models as well. The decision tree models performed better than the SVM models, but so far, the best model is still the logistic regression, which contains the macroeconomic variables. Milosevic (2016) reported higher F1 score for the decision tree model, which was 66%. Also, the recall and precision values in Milosevic's (2016) research were higher than the values obtained in this study.

4.1.4 Random forest

Both random forest models were constructed using the entropy function, which was also used for the decision tree models. The Gini index function was tested as well to compare which function could give better performance results. But as for the single decision tree model also for the random forest models it turned out that the entropy function gave overall better results and higher accuracy score. In figure 10, the effects of different number of decision trees in the random forest models to the accuracy of the models are analyzed.

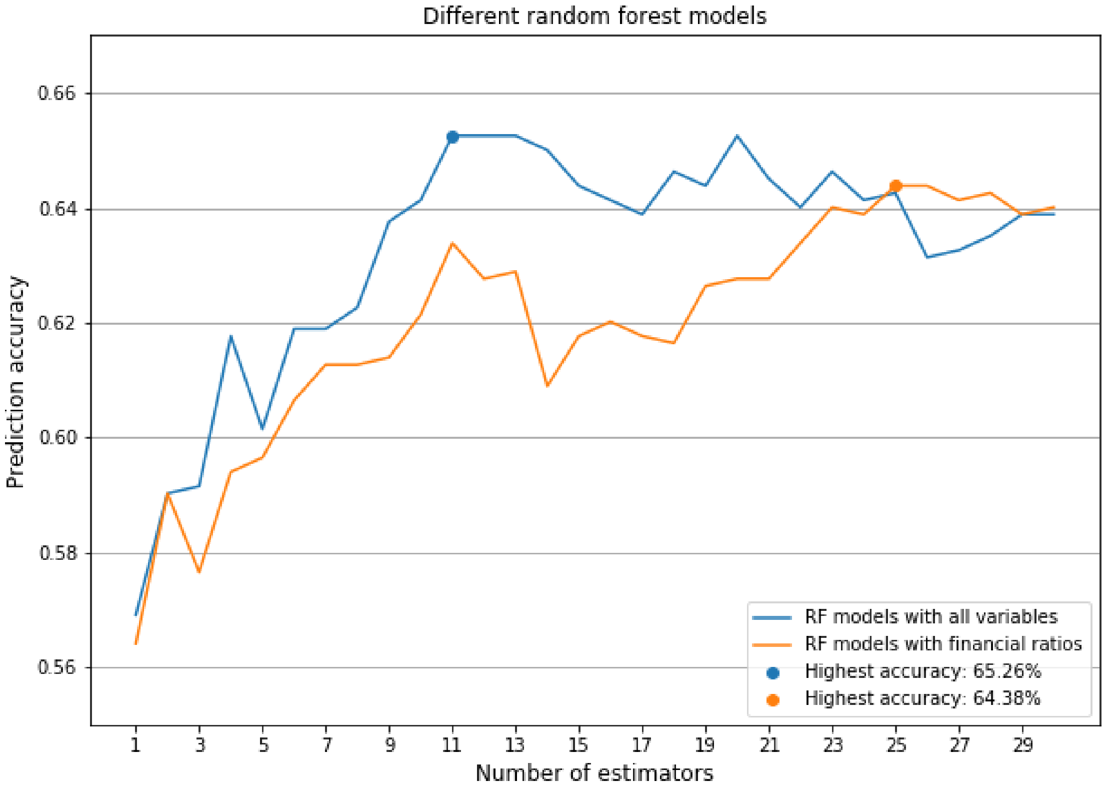


Figure 10 Random forest models with different number of decision trees

For both random forest models, 1 up to 30 different decision trees were included and tested to find out what is the best number of trees which gives the highest level of prediction accuracy. At the beginning, the accuracy of both models gets higher quite a lot, when the number of decision trees is increased. But after a certain point, the accuracy does not get much higher and it levels out around 65% even when the number of estimators is still increased in the models. The first random forest model with only financial ratios achieved the highest accuracy of 64.38% when it contained 25 decision trees. The second random forest model, which contained also the macroeconomic variables, achieved the highest accuracy of 65.26% when it had 11 decision trees. The confusion matrices for both models are presented in table 13.

Table 13 Confusion matrices of random forest models

Random forest with the financial ratios			
n=803	Predicted: No	Predicted: Yes	
Actual: No	318	123	441
Actual: Yes	163	199	362
	481	322	

Random forest with all the variables			
n=803	Predicted: No	Predicted: Yes	
Actual: No	308	133	441
Actual: Yes	146	216	362
	454	349	

The outcomes of the two models seem to spread quite evenly between the two classes. Both models predicted a few more stocks as low return stocks, but still the actual amounts and predicted amounts are in the same range. The first model, which contained only financial ratios, classified correctly 517 outcomes. This model had 123 stocks that were classified as low return stocks that were in reality high return stocks. Also, 163 stocks were predicted to be high return stocks, but in reality, these stocks provided lower returns.

The second random forest model, which included also the macroeconomic variables classified correctly 524 stocks. There were 133 false positive outcomes and 146 false negative outcomes. Final results of the models are shown in table 14.

Table 14 Results of the random forest model

Independent variables in the model	***Precision***	***Recall***	***F1 score***	***Accuracy***
Only financial ratios	0.618	0.549	0.582	0.644
All variables	0.619	0.597	0.608	0.653

The F1 score of the first model is 58.2% and the classification accuracy of the model is 64.4%. The second model had F1 score of 60.8% and the classification accuracy of 65.3%. The F1 score values are now higher because the recall values of the models are higher than what the earlier models obtained. The random forest models have the best accuracy score of the models so far. The second random forest model also performs a little bit better than the first one. Therefore, the macroeconomic variables add some information to the model, which is not only explained by financial ratios. Milosevic (2016) reported 76.5% F1 score for the random forest model, which is higher than what was achieved in this study. Overall, the results obtained in this research are aligned with the results from earlier academic studies. The best model in Milosevic (2016) and Ballings et al. (2015) studies were also a random forest model.

4.1.5 KNN

KNN models were constructed using the Euclidean distance function to calculate the distance of the test observations and assigning them to different groups. Also, different numbers of neighbors were tested to examine the bias-variance trade-off. In figure 11, the effect of different number of neighbors to the accuracy of the models have been visualized.

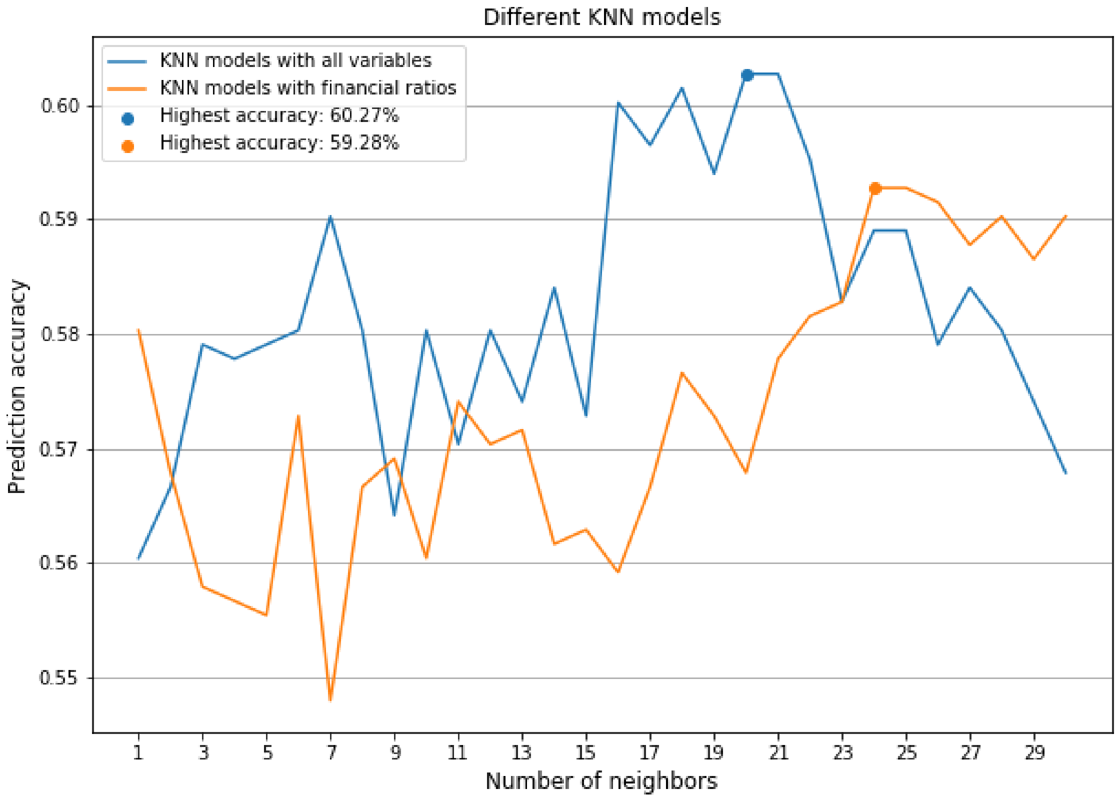


Figure 11 KNN models with different neighbors

Different number of neighbors have a strong effect on the prediction accuracy of the models. From 1 to 30 different neighbors were tested for the two models. The first model achieved the highest accuracy of 59.28% when it contained 24 neighbors. The second model reached the highest accuracy of 60.27% when it contained 20 neighbors. The confusion matrices of the two models are presented in table 15.

Table 15 Confusion matrices of the KNN models

KNN with the financial ratios			
n=803	Predicted: No	Predicted: Yes	
Actual: No	344	97	441
Actual: Yes	230	132	362
	574	229	

KNN with all the variables			
n=803	Predicted: No	Predicted: Yes	
Actual: No	302	139	441
Actual: Yes	180	182	362
	482	321	

The first model points out to predict over 70% of the outcomes to the negative class and the rest of the outcomes to the positive class, which means the recall value of the model is low. The second model predicted around 60% of the outcomes to the negative class and the rest of them to the positive class. The actual values are split around half between the positive and negative class; around 55% belong to the positive class and around 45% belong to the negative class. The first model classified correctly 476 stocks and the second model 484 stocks out of 803 stocks. The first model had 97 false positive outcomes and 230 false negative outcomes. The second model had 42 false positive outcomes more and 50 false negative outcomes less than the first model. The final results of the two models are shown in table 16.

Table 16 Results of the KNN models

Independent variables in the models	***Precision***	***Recall***	***F1 score***	***Accuracy***
Only financial ratios	0.576	0.365	0.447	0.593
All variables	0.567	0.503	0.533	0.603

The first model had a classification accuracy of 59.3%, but the F1 score pointed out to be rather low, only 44.7%. This is due to low recall value of the model. Meaning that only a few of the stocks that were high return stocks were classified as high return stocks. The second model had higher recall value compared to the first model. This means that it classified more stocks as high return stocks that truly were high return stocks. The second model also had a higher prediction accuracy when measured by classification accuracy metric and F1 score. The precision values are quite the same between the two models. All things considered, the second model performed better than the first model.

Overall, the macroeconomic variables seem to add information to the models which is not explained only by the financial ratios. The macroeconomic variables increased the prediction accuracy and the overall performance of all the models used in this research. In Milosevic's (2016) research, the accuracy of the KNN model was not tested. Shynkevich et al. (2017) reported the best prediction accuracy of 60.26% for the KNN model when predicting future directions of stock price movements. The results from Shynkevich et al. (2017) research are quite aligned with the results obtained in this study.

4.2 Summary of the results

In this chapter, all the results are summarized. First, the summary of all models that contained only financial variables are presented and analyzed. Then, the chapter will move on to presenting the summary of the models that have all the independent variables included. Also, the ROC curves of each model will be presented. Overall, the results of the machine learning models indicate evidence, how hard it actually is to predict which stocks are going to give high returns and which ones are not. In table 17, all the results of the models, which contained only financial ratios are presented. The best machine learning model, which contains the highest F1 score and classification accuracy is highlighted.

Table 17 Results of the machine learning models with only financial ratios

Algorithm	***Precision***	***Recall***	***F1 score***	***Accuracy***
Logistic regression	0.526	0.169	0.255	0.557
SVM	0.562	0.163	0.253	0.565
Decision tree	0.511	0.503	0.507	0.559
Random forest	**0.618**	**0.549**	**0.582**	**0.644**
KNN	0.576	0.365	0.447	0.593

The random forest model is the best model with highest precision, recall, F1 score, and classification accuracy values. The F1 score of the model is 58.2% and classification accuracy is 64.4%. These values are much higher compared to the other models. Therefore, the random forest model predicts the equity price movement the best. Only the decision tree model, which is the second best model measured by these metrics, comes quite close to the results of the random forest model. This is because these two models are quite similar. The random forest model is a boosted version of the decision tree model, including multiple decision trees. Each tree in random forest is only a small subset of the features and therefore the trees are a little bit different from each other. The random forest averages the results across the different trees which decreases the variance. This leads to more stable and accurate predictions because the algorithm reduces the effect of overfitting since the variance is lower.

The decision tree model contains higher variance and therefore the prediction results are lower compared to the random forest model. However, the decision tree model can predict the equity price movement correctly a little bit over 50% of the times. This is still a decent score for the model which makes it the second-best model among the models which contain only financial ratios as independent variables. The F1 score results of the SVM and logistic regression models are extremely low. This is due to the low recall values of the models. Meaning that the stocks that truly were high return stocks only a few of them were classified as high return stocks by both models. Overall, the models do not produce reliable prediction results for the equity price movement and perform quite badly. These results are quite opposite of the results Dutta et al. (2012) and Milosevic (2016) reported. They obtained much higher prediction results with these two machine learning models.

The receiver operating characteristic (ROC) curve is a common tool when dealing with binary classifiers. In figure 12, the ROC curves of all the models, which included only financial ratios as independent variables, are presented. The AUC values of the models are also displayed in the lower right corner of the figure.

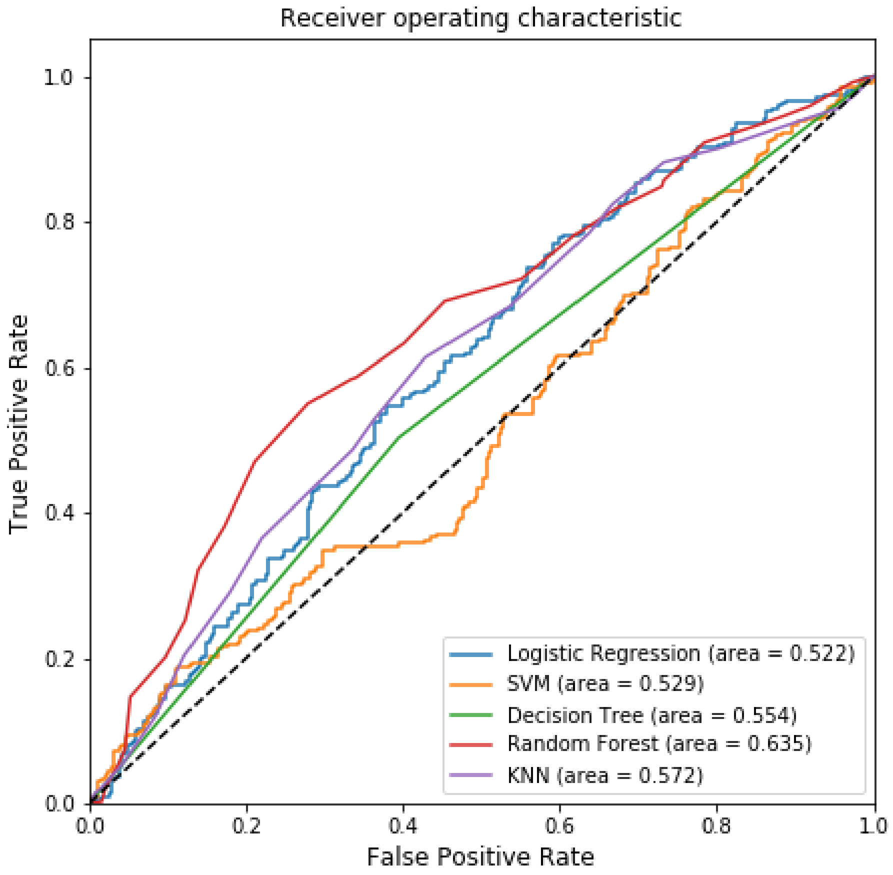


Figure 12 ROC curves of the models when financial variables are included

The dotted line represents the ROC curve of a purely random classifier. Also, the ROC curve analysis indicates the same kind of results as the previous performance metrics described. The random forest model is the best classifier of all the models. The ROC curve of the random forest model stays as far away from the dotted line as possible. In other words, the line of the random forest model is the closest one to the top-left corner. Meaning that the AUC value of the model is the greatest. This indicates that the model has the highest chance of being able to distinguish whether an observation belongs to the positive class or to the negative class. In this case, the random forest model has a 63.5% chance of predicting an observation correctly to the class where it truly belongs.

According to the AUC values, the second-best model points out to be KNN model with the AUC value of 57.2%. This result is a little bit different from what the previous metrics presented. Although, the KNN model had the second-best classification accuracy score, the recall and F1 score values were lower compared to the values of the decision tree model. The AUC values of the SVM and logistic regression models confirm that these models are still the worst models among all the models tested in this study. The AUC values of these two models are only a little bit above 50%. This means that the models have almost no discrimination capacity to distinguish between positive and negative class.

The AUC values of the KNN, SVM, and random forest models are much lower than what Ballings et al. (2015) reported in their research. Ballings et al. (2015) obtained AUC value of 90.37% for the random forest model, 80.95% for the SVM, and 72.65% for the KNN model. These results are extremely high compared to the results from this study and indicate that the price movements of European companies are much more predictable than the price movement of Finnish companies. However, the results cannot be compared straightforwardly because the time periods are different and also some of the independent variables used in both studies differ. In table 18, all the results of the models, which contained also macroeconomic variables are presented. The best machine learning model, which contains the highest F1 score and classification accuracy is highlighted.

Table 18 Results of the machine learning models with all the variables

Algorithm	***Precision***	***Recall***	***F1 score***	***Accuracy***
Logistic regression	0.512	0.724	0.599	0.564
SVM	0.569	0.412	0.478	0.594
Decision tree	0.545	0.503	0.523	0.587
Random forest	**0.619**	**0.597**	**0.608**	**0.653**
KNN	0.567	0.503	0.533	0.603

The random forest model is the best classifier among all other models when also macroeconomic variables are added in the models. The random forest model has the highest scores for every performance metric. F1 score of the model is 60.8% and the classification accuracy is 65.3%. These scores are even higher that what the previous random forest model achieved with only financial ratios as independent variables. The second-best model points out to be the logistic regression model when measured by F1 score. But KNN model had the second-best classification accuracy. The performance of the logistic regression model and the SVM model increased significantly when the macroeconomic variables were added into the models. The macroeconomic variables increased the performance of all the models. F1 score and classification accuracy of the models, which included also macroeconomic variables are higher compared to the models, which did not contain the macroeconomic variables. In figure 13, the ROC curves of the models which contained also the macroeconomic variables are presented. The AUC values of the models are also displayed in the lower right corner of the figure.

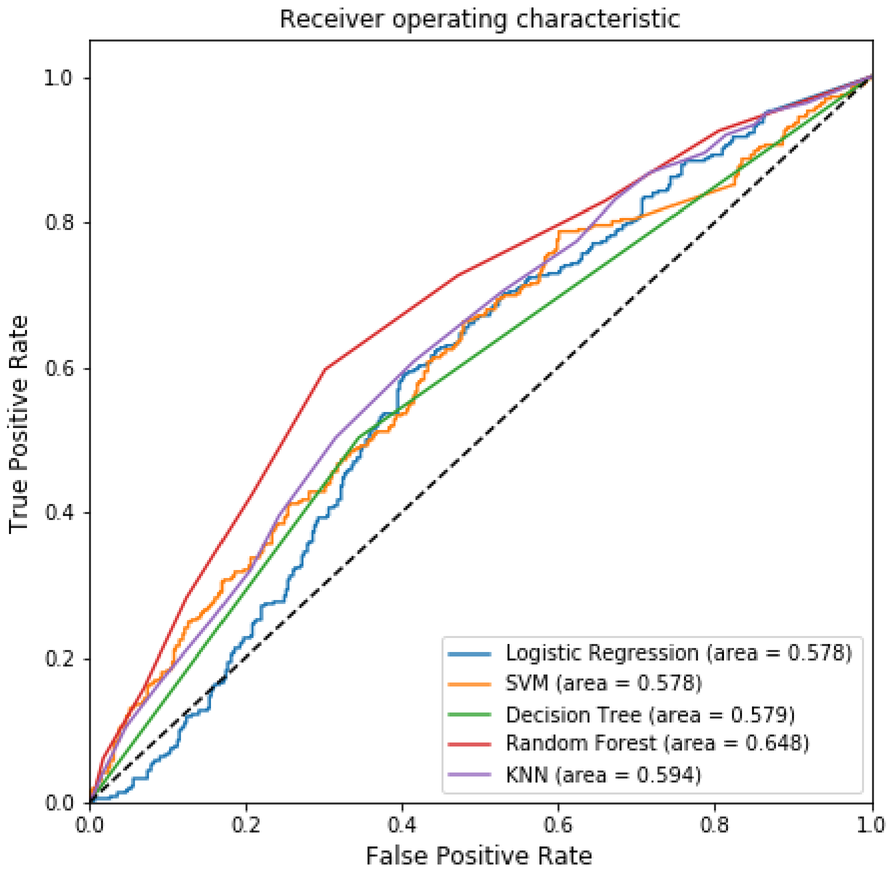


Figure 13 ROC curves of the models when all the variables are included

The ROC curve analysis presents the same kind of results than the previous performance metrics described. The ROC curve of the random forest model is farthest from the other ROC curves of the models. Therefore, it is the closest to the left corner indicating that the area under the curve is largest. The ROC curve also confirms that the best classifier is the random forest model. The AUC value of the random forest model is 64.8% indicating that the model has a 64.8% chance of predicting an observation correctly to its class where it truly belongs. The second-best model is the KNN model with AUC value of 59.4%. Also, now the AUC values of the logistic regression model and the SVM model are higher compared to the previous AUC values.

The AUC values of the models are again much lower than in Ballings et al. (2015) research. But overall, the AUC values of all the models are now higher than the values of the previous models, which included only financial ratios. The ROC curves of the models point out that it is challenging to predict whether a stock is going to perform well and give high returns or not.

5 CONCLUSIONS

5.1 The purpose of the study and conclusions

The purpose of this study was to find out which machine learning model could predict the best long-term equity price movement of the stocks listed on the Helsinki Stock Exchange. The purpose was also to examine the effects of macroeconomic variables to the equity price movement. For instance, could macroeconomic variables increase the prediction accuracy of the models. In the study, following machine learning models were tested: logistic regression, SVM, decision tree, random forest, and KNN. All the models were first tested with only financial ratios, and then, including also macroeconomic variables in the models to compare the results. Following 14 variables turn out to give the best results when forecasting the equity price movement:

- Market capitalization (MC)
- Price-to-book ratio (PBR)
- Price-earnings ratio (PER)
- Quick ratio (QR)
- Earnings before interest and taxes (EBIT)
- Earnings per share (EPS)
- Dividend per share (DPS)
- Dividend yield (DY)
- Close price of a stock (CPS)
- Return on invested capital (ROIC)
- Total debt divided by total capital (TDTC)
- Unemployment rate (UR)
- Gross domestic product (GDP)
- Expected inflation for the next 6 months (EXPINF6M)

The machine learning models were trained in a way that they can predict which stocks will have 10% or higher return in a year, and which ones will not have. The results of the models were evaluated by using the following performance metrics: confusion matrix, precision, recall, F1 score, ROC curve, and AUC. According to these performance metrics, the best model for predicting the long-term equity price movement was the random forest model. It was the best model when it was only used with financial ratios as well as when it also included the macroeconomic variables. The random forest model, which included all the independent variables, obtained the highest classification accuracy of 65.3% and the highest F1 score of 60.8%. Also, the AUC value of this model was the highest, 64.8%, compared to all other models used in the study. This result of the study is aligned with the results Milosevic (2016) and Ballings et al. (2015) reported in their studies. Although, the F1 score of the model was not as high as 76.5%, which Milosevic (2016) achieved. Also, the AUC value of the random forest model was not as high as in Ballings et al. (2015) research.

There are a couple of reasons why the prediction accuracy of the models did not reach as high as in the other studies. One is that this study was conducted from another perspective in which specific time period; quarter one in 2016 to quarter two in 2018 was selected to make the predictions. Instead of randomly dividing the whole data set into training and testing sets like in the study Dutta et al. (2012) or using other methods to divide the data set. For instance, in Milosevic (2016) research the data set was divided into training and testing sets using a 10-fold cross-validation.

In this research, the data was divided into training and testing sets based on the years. All observations before the year 2016 were selected in the training set and all observations from 2016 to 2018 were selected in the testing set. The whole data was not divided into training and testing sets randomly because in this case the models will contain a look-ahead bias. The look-ahead bias will occur in the model because it already knows some later data points when predicting the earlier ones. For instance, when predicting the second quarter value of the year 2012, the model already knows the values of the first quarter in the year 2012. Therefore, the look-ahead bias will increase the accuracy of the models. This is one of the reasons why the prediction accuracy of the logistic regression model was lower in this study compared to the results Dutta et al. (2012) reported in their research. Because of this fact, the results obtained from this study are not fully comparable with Dutta et al. (2012) research.

Another main reason for lower accuracy values and prediction results is the quality of the data. The whole data set was gathered from Thompson Reuters Datastream, which is a respected data source. However, the main concerns are related to the information about the companies listed on the Helsinki Stock Exchange. There were some financial ratios that were not available for some or all the Finnish companies. For instance, the beta values were not available for the companies. The missing values varied between companies as well, for instance, for some smaller companies there were more missing values than for the larger companies. Also, some financial ratios that were used in the study were not available for each quarter. For example, earnings per share, earnings before interest and taxes, price-to-book ratio, and a couple of other ratios. Therefore, their values were the same for quarter to another within a year and changing only on a yearly basis. The fact that the data is not as accurate as possible lowers also the prediction results of the models.

It must be pointed out that the results obtained from the models can be quite easily manipulated. For instance, by including the look-ahead bias in the model in order to get higher prediction results or selecting shorter time period for the testing sets which could increase the prediction results as well. The independent variables also affect greatly on the accuracy of the models. Interestingly, even small independent variable changes in the models can have a major impact on the final accuracy and the results of the models. One of the reasons for this is the nature of financial ratios. Horrigan (1965) has pointed out main problems related to financial ratios. The ratios often are not normally distributed, and their dispersion can be large. Also, financial ratios contain collinearity already, on some level since many ratios are calculated using the same factors in the equations. These factors can increase the variance in the models and therefore the variable changes can have great effect on the accuracy of machine learning models.

The first hypothesis of the study was the following one;

- Hypothesis 1 (H1): By applying machine learning models and training them on the past data, it is possible to predict the equity price movement of the stocks listed on the OMXH

Based on the results from this study, the hypothesis cannot be fully rejected. In order to reject the hypothesis, none of the models used in the research should not be able to predict the price movement with accuracy more than 50%. Meaning that the models will not provide excess returns for investors based on the information about financial ratios and macroeconomic variables. However, the random forest model is able to give over 60% chance for an investor to pick a stock, which will have a return of 10% or higher over the period of one year. Therefore, all the prices in the Finnish financial markets are not equally random. Indicating that the prices in the Finnish financial markets are not always following a random walk process. The stocks listed on the Helsinki Stock Exchange have been on some level predictable between the years 2016–2018 based on the information about financial ratios and macroeconomic variables. According to this finding the market does not fill semi-strong-from efficiency requirements in this time period.

There is still a need for studying the hypothesis more. For example, testing other financial ratios, technical indicators, and macroeconomic variables and possible including other data as well. One possibility could be to include qualitative data about the companies in the machine learning models as well and examine if the qualitative factors can increase the prediction accuracy of the models. Also, another time period and different benchmark return could be studied.

The second hypothesis of the study was the following one;

- Hypothesis 2 (H2): Macroeconomic variables will increase the prediction accuracy of the machine learning models compared to the same models, which do not contain the variables

The results in this study indicate that the macroeconomic variables can improve the prediction accuracy of the machine learning models. The macroeconomic variables increased the prediction accuracy of every model used in this research. For logistic regression and SVM models, the increase in F1 score was significant. The increase was not as significant for the other models, but their prediction performance did increase as well. Overall, the macroeconomic variables seem to provide new information for the model which is not explained only by the financial ratios. Especially, for a long-term investment, macroeconomic variables point out to be useful to include in the models. Therefore, this hypothesis cannot be fully rejected based on the findings obtained in this study.

Some of the conclusions drawn from this study are different compared to the prior studies that have predicted equity price movement using machine learning models. Prior studies have reported high prediction accuracies for the models. The accuracies reported have been well above 70 percent. However, it turned out that this was not the case in this study. There are many factors that have to be taken into account when utilizing machine learning models, such as what variables to include in the models and how the parameters of the models have to be adjusted. Although, this study confirms that the best model is a random forest model when predicting equity price movement. This finding has been confirmed also by Milosevic (2016) and Ballings et al. (2015). In addition to the previous studies, macroeconomic variables can also provide useful and valuable information for machine learning models to predict long-term equity price movement. The macroeconomic variables increased the prediction performance of every model used in this study.

5.2 Suggestions for future research

The following topics could be potentially interesting ways to continue the research in this area. Using machine learning models to forecast the direction of the Finnish stock market index. Comparing the predictability of large cap firms to small and medium-sized firms. For instance, examine if the price movements from large cap stocks are easier to predict compared to small cap stocks or is there any statistically significant differences. Different time periods could also be considered to find out if there exist periods when stock returns are easier to predict and periods when they are hard to predict.

It would also be interesting to examine if some other machine learning model could predict the equity price movement even better than the models used in this study. For instance, could an artificial neural network model achieve a higher prediction accuracy. Since the independent variables have a major effect on the prediction results of the models, it is important to also examine different independent variables in the models. For example, using technical indicators to predict short-term equity price movement.

REFERENCES

Agrawal, A. – Tandon, K. (1994) Anomalies or illusions? Evidence from stock markets in eighteen countries. *Journal of International Money and Finance,* Vol. 13 (1), 83–106.

Alpaydin, E. – Bach, F. (2014) *Introduction to machine learning.* 3rd ed. The MIT Press, London.

Back, K. – H. Pedersen (1998) Long-lived information and intraday patterns. *Journal of Financial Markets,* Vol. 1, 385–402.

Baker, K. – Nofsinger, J. (2010) *Behavioral finance: investors, corporations, and markets.* Wiley.

Ballings, M. – Van den Poel, D. – Hespeels, N. – Gryp, R. (2015) Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications,* Vol. 42 (20), 7046–7056.

Barone, E. (1990) The Italian stock market: efficiency and calendar anomalies. *Journal of Banking & Finance,* Vol. 14 (2–3), 483–510.

Basak, S. – Kar, S. – Saha, S. – Khaidem, L. – Dey, S. (2019) Predicting the direction of stock market prices using tree-based classifiers. *North American Journal of Economics and Finance*, Vol. 47, 552–567.

Basu, S. (1977) Investment performance of common stocks in relation to their price earnings ratios: a test of the efficient market hypothesis. *The Journal of Finance*, Vol. 32 (3), 663–682.

Batrinca, B. – Hesse, C. – Treleaven, P. (2017) Developing a volume forecasting model. *Journal of Applied Finance & Banking,* Vol. 7 (1), 1–40.

Bech, M. – Gyrd-Hansen, D. (2005) Effects coding in discrete choice experiments. *Health economics*, Vol. 14 (10), 1079–1083.

Bensic, M. – Sarlija, N. – Zekic-Susac, M. (2005) Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. *Intelligent systems in accounting, finance and management*, Vol. 13, 133–150.

Berthold, M. R. – Borgelt, C. – Höppner, F. – Klawonn, F. (2010) *Guide to Intelligent data analysis: How to intelligently make sense of real data.* Springer London.

Bo, L. – Wang, L. – Jiao, L. (2006) Feature scaling for kernel fisher discriminant analysis using leave-one-out cross validation. *Neural Computation*, Vol. 18 (4), 961–978.

Borges, M. (2010) Efficient market hypothesis in European stock markets. *The European Journal of Finance,* Vol. 16 (7), 711–726.

Bragg, Steven, M. (2012) *Business ratios and formulas: a comprehensive guide.* 3rd ed. Hoboken, N.J.

Breiman, L. (2001) Random forests. *Machine Learning,* Vol. 45, 5–32.

Carcía-Ferrer, A. – Bujosa-Brun, M. (2000) Forecasting OECD industrial turning points using unobserved components models with business survey data. *International Journal of Forecasting.* Vol. 16 (2), 207–227.

Chau, M. – Vayanos, D. (2008) Strong-form efficiency with monopolistic insiders. *The Review of Financial Studies,* Vol. 21 (5), 2275–2396.

Chen, S. – Härdle, W. – Moro, R. (2011) Modeling default risk with support vector machines. *Quantitative Finance,* Vol. 11 (1), 135–154.

Chikkodi, C. M. – Satyaprasad, B. G. (2010) *Business statistics.* Rev. ed. Himalaya Pub. House 2010.

Chitenderu, T. – Maredza, A. – Sibanda, K. (2014) The random walk theory and stock prices: evidence from Johannesburg stock exchange. *International Business & Economics Research Journal,* Vol. 13 (6), 1241–1249.

Corters, C. – Vapnik, V. (1995) Support-vector networks. *Machine Learning,* Vol. 20 (2), 273–297.

Goel, Sandeep (2016) *Financial ratios*. 1st ed. Business Expert Press, New York.

Daniel, G. – Sornette, D. – Woehrmann, P. (2009) Look-ahead benchmark bias in portfolio performance evaluation. *Journal of Portfolio Management*, Vol. 36 (1), 121–130.

Datastream (2018) Thomson Reuters Datastream. [Online]. Available at: Subscription Service (Accessed: October 2018)

Dutta, A. – Bandopadhyay, G. – Sengupta, S. (2012) Prediction of stock performance in the Indian stock market using logistic regression. *International Journal of Business and Information*, Vol. 7 (1), 105–136.

Eliason, Scott, R. (1993) *Introduction: the logic of maximum likelihood.* Little Green Book.

Elosegui, P. (2003) Aggregate risk, credit rationing and capital accumulation. *The Quarterly Review of Economics and Finance,* Vol. 43 (4), 668–696.

Fama, E. (1965) Random walks in stock market prices. *Financial Analyst Journal,* Vol. 21 (5), 55–59.

Fama, E. (1970) Efficient capital markets. *Journal of Finance*, Vol. 25 (2), 383–417.

Fama, E. – French, K. (1988) Permanent and temporary components of stock prices. *Journal of Political Economy*, Vol. 96 (2), 246–273.

Fama, E. – French, K. (2012) Size, value, and momentum in international stock returns. *Journal of Financial Economics*, Vol. 105 (3), 457–472.

Fama, E. – French, K. (2015) A five-factor asset pricing model. *Journal of Financial Economics*, Vol. 116, 1–22.

Fan, A. – Palaniswami, M. (2001) Stock selection using support vector machines. *International Joint Conference on Neural Networks,* Vol. 3, 1793–1798.

Ferson, W.E. – Harvey, C.R. (1993) The risk and predictability of international equity returns. *Review of Financial Studies*, Vol. 6 (3), 527–66.

Gultekin, M. N. – Gultekin, N. B. (1983). Stock market seasonality. International evidence. *Journal of Financial Economics,* Vol. 12 (4), 469–481.

Hassanat, A. – Abbadi, M. – Altarawneh, G (2014) Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. *International Journal of Computer Science and Information Security,* Vol. 12 (8), 33–39.

Hastie, T. – Tibshirani, R. – Friedman, J. (2003) *The elements of statistical learning: data mining, inference, and prediction*. 3rd ed. Springer, New York.

Hietala, P. (1994) The efficiency of the Finnish market for right issues. *Journal of Banking & Finance*, Vol. 18 (5), 895–920.

Hosmer, D. W. – Lemeshow, S. – Sturdivant, R. X. (2013) *Applied logistic regression.* 3rd ed. Wiley, Hoboken.

Horrigan, J. (1965) Some empirical bases of financial ratio analysis. *The Accounting Review*, Vol. 40 (3), 284–294.

Huynh, H. D. – Dang, L. M. – Duong, D. (2017) A New model for stock price movements prediction using deep neural network. *SoICT 2017,* 57–62.

Huang, W. – Nakamori, Y. – Wang, S. (2005) Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, Vol. 32 (10), 2513–2522.

Hyytinen, A. – Väänänen, L. (2002) Government funding of small and medium-sized enterprises. Discussion Paper, No 823. Helsinki, The Research Institute of the Finnish Economy (ETLA).

Imandoust, S. B. – Bolandraftar, M. (2013) Application of k-nearest neighbor (KNN) approach for predicting economic events: theoretical background. *International Journal of Engineering Research and Applications,* Vol. 3 (5), 605–610.

Jaffe, J. – Keim, D. – Westerfield, R. (1989) Earnings yields, market values, and stock returns. *Journal of Finance*, Vol. 44 (1), 135–48.

James, G. – Witten, D. – Hastie, T. – Tibshirani R. (2013) *An introduction to statistical learning: with applications in R*. Springer.

Kahneman, Daniel (2011) *Thinking fast and slow.* Farrar, Straus and Giroux, New York.

Kahneman, D. – Tversky, A. (1979) Prospect theory: an analysis of decision under risk *Econometrica*, Vol. 47 (2), 263–292.

Kahneman, D. – Riepe, M. (1998) Aspects of investor psychology. *The Journal of Portfolio Management*, Vol. 24 (4), 52–65.

Karazmodeh, M. – Nasiri, S. – Hashemi, S. M. (2013) Stock price forecasting using support vector machines and improved particle swarm optimization. *Journal of Automation and Control Engineering,* Vol. 1 (2), 173–176.

Kiander, J. – Vartia, P. (2011) Lessons from the crisis in Finland and Sweden in the 1990s. *Empirica*, Vol 38 (1), 53–69.

Kwon Y. – Choi, S. – Moon, B. (2002) Weighted scale-free network in financial correlation. *Journal of the Physical Society of Japan,* Vol. 71 (9), 2133–2136.

Leung, M. – Daouk, H. – Chen, A. (2000) Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting,* Vol. 16 (2), 173–190.

Lo, A. W. (2004) The adaptive market efficiency from an evolutionary perspective. *Journal of Portfolio Management,* Vol. 5 (30), 15–29.

Lo, Andrew W. (2017) *Adaptive markets: financial evolution at the speed of thought.* Princeton University Press.

Lo, A. W. – MacKinlay, A. (1988) Stock market prices do not follow random walks: evidence from a simple specification test. *Review of Financial Studies,* Vol. 1 (1), 41–66.

Malkiel, B. (2003) The efficient market hypothesis and its critics. *Journal of Economic Perspectives,* Vol. 17 (1), 59–82.

Martikainen, T. – Puttonen, V. (1996) Finnish day-of-the-week effects. *Journal of Business Finance & Accounting*, Vol. 23 (7), 1019–1032.

Martínez, F. – Frías, M. P. – Pérez, M. D. – Rivera, A. J. (2017) A methodology for applying k-nearest neighbor to time series forecasting. *The Artificial Intelligence Review,* 1–19.

Matoussi, H. (2010) Credit-risk evaluation of a Tunisian commercial bank: logistic regression vs neural network modelling. *Accounting and Management Information Systems*, Vol. 9 (1), 92–119.

Milosevic, N. (2016) Equity forecast: predicting long term stock price movement using machine learning. *Journal of Economics Library,* Vol. 3 (2), 288–294.

Mohri, M. – Rostamizadeh, A. – Talwalkar, A. (2012) *Foundations of machine learning.* MIT Press, Cambridge.

Nyberg, P. – Vaihekoski, M. (2014) Equity premium in Finland and long-term performance of the Finnish equity and money markets. *Cliometrica,* 8 (2), 241–269.

Odean, T. (1998) Volume, volatility, price and profit when all traders are above average. *The Journal of Finance,* Vol. 53 (6), 1887–1934.

Official Statistics of Finland (OSF), Labor force survey ISSN=1798-7857. <http://www.stat.fi/til/tyti/index_en.html>, retrieved 23.12.2018.

Ohlson, J. (1980) Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, Vol. 18 (1), 109–131.

Phua, P. – Zhu, X. – Koh, C. (2003) Forecasting stock index increments using neural networks with trust region methods. *Proceedings of the International Joint Conference on Neural Networks*, Vol. 1, 260–265.

Pilbeam, Keith (2010) *Finance & financial markets.* 3rd ed. Palgrave Macmillan, Basingstoke.

Powers, D. (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies,* Vol. 2 (1), 37–63.

Qian, B. – Rasheed, K. (2007) Stock market prediction with multiple classifiers. *Applied Intelligence*, Vol. 26 (1), 25–33.

Scikit-learn, documentation of SVM model. < https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>, retrieved 26.1.2019.

Shleifer, Andrei (2003) *Inefficient markets: an introduction to behavioral finance.* Oxford University press, Oxford.

Shalev-Shwartz, S. – Ben-David, S. (2014) *Understanding machine learning; from theory to algorithms.* Cambridge University Press, New York.

Shynkevich, Y. – McGinnity, T. M. – Coleman, S. – Belatreche, A. – Li, Y. (2017) Forecasting price movements using technical indicators: investigating the impact of varying input window length. *Neurocomputing*, Vol. 264, 71–88.

Sung, T. K. – Chang, N. – Lee, G. (1999) Dynamics of modeling in data mining: interpretive approach to bankryptcy prediction. *Journal of Management Information Systems,* Vol. 16 (1), 63–85.

Tseng, C. (2007) Data driven modeling of co-movement amoung international stock market. *Journal of Modelling in Management*, Vol. 2 (3), 195–207.

Ullah, I. – Ullah, A. – Rehman, N. (2017) Impact of overconfidence and optimism on investment decision. *International Journal of Information, Business and Management,* Vol. 9 (2), 231–243.

Vapnik, Vladimir, N. (2000) *The nature of statistical learning theory*. Springer-Verlag, New York.

Wang, Lipo (2005) *Support vector machines: theory and applications*. Springer-Verlag, Heidelberg.

Welch, I. – Goyal, A. (2008) A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies,* Vol. 21 (1), 1455–1508.

Webb, A. R. – Copsey, K. D. (2011) *Statistical pattern recognition*. 3rd ed. John Wiley & Sons.

Young, E. – Jeong, M. (2009) Class dependent feature scaling method using Bayes classifier for text datamining. *Pattern Recognition Letters,* Vol. 30 (5), 477–485.

Zhang, J. – Teng, Y. – Chen, W. (2018) Support vector regression with modified firefly algorithm for stock price forecasting. *Applied Intelligence,* 1–17.

APPENDIX 1 PYTHON CODE USED

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, roc_auc_score, roc_curve
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier

df = pd.read_excel("data.xlsx")

# Heatmap
sample = df.loc[:,['MC', 'PBR', 'PER', 'QR', 'EBIT', 'EPS', 'DPS', 'DY', 'CP',
                'ROIC', 'TDTC', 'UR', 'GDP', 'EXPINF6M']]

corr = sample.corr()

sns.set(rc={'figure.figsize':(16,10)}, font_scale=1.3)
heatmap_plot = sns.heatmap(corr, annot=True, linewidths=.5, fmt='.3f', annot_kws={"size":
16});
fig = heatmap_plot.get_figure()

# Filling missing values with -99 999
df = df.fillna(-99999)

# Summary of Logistic regression
import statsmodels.api as sm
X = df.loc[:,['MC', 'PBR', 'PER', 'QR', 'EBIT', 'EPS', 'DPS', 'DY', 'CP',
                'ROIC', 'TDTC', 'UR', 'GDP', 'EXPINF6M']].values

X_const = sm.add_constant(X)
y = df.loc[:,'EPM'].values
```

```
logit_model=sm.Logit(y,X_const)
result=logit_model.fit()
print(result.summary2())

# the odds:
np.exp(result.params)

# Data preprocessing:
X = df.loc[:,['Company', 'Date','MC', 'PBR', 'PER', 'QR', 'EBIT', 'EPS', 'DPS', 'DY', 'CP',
            'ROIC', 'TDTC', 'UR', 'GDP', 'EXPINF6M']]

y = df.loc[:,['Date','EPM']]

# Splitting the dataset into the Training and Test set
X_train = X[X['Date'] < '2016-01-01'].values
y_train = y[y['Date'] < '2016-01-01'].values

X_train = np.delete(X_train, 1, axis=1)
y_train = np.delete(y_train, 0, axis=1)

X_test = X[X['Date'] >= '2016-01-01'].values
y_test = y[y['Date'] >= '2016-01-01'].values

X_test = np.delete(X_test, 1, axis=1)
y_test = np.delete(y_test, 0, axis=1)

y = y_train.ravel()
y_train = np.array(y).astype(int)

y1 = y_test.ravel()
y_test = np.array(y1).astype(int)

# Encoding categorical data
# Training data:
labelencoder_X1_train = LabelEncoder()
X_train[:,0] = labelencoder_X1_train.fit_transform(X_train[:,0])

onehotencoder = OneHotEncoder(categorical_features = [0])
X_train = onehotencoder.fit_transform(X_train).toarray()
```

```
# Testing data:
labelencoder_X1_test = LabelEncoder()
X_test[:,0] = labelencoder_X1_test.fit_transform(X_test[:,0])

onehotencoder = OneHotEncoder(categorical_features = [0])
X_test = onehotencoder.fit_transform(X_test).toarray()

# Avoid dummy variable trap:
X_train = np.delete(X_train, 0, axis=1)
X_test = np.delete(X_test, 0, axis=1)

# Feature Scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)


# LOGISTIC REGRESSION:
# Fitting Logistic Regression to the Training set
logit = LogisticRegression(random_state=0)
logit.fit(X_train, y_train)

# Predicting the Test set results
y_pred = logit.predict(X_test)

# Making the Confusion Matrix
print(confusion_matrix(y_test, y_pred))

# Results
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("F1-score", metrics.f1_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))

# SVM:
acc = []
f1_scores=[]
pres=[]
```

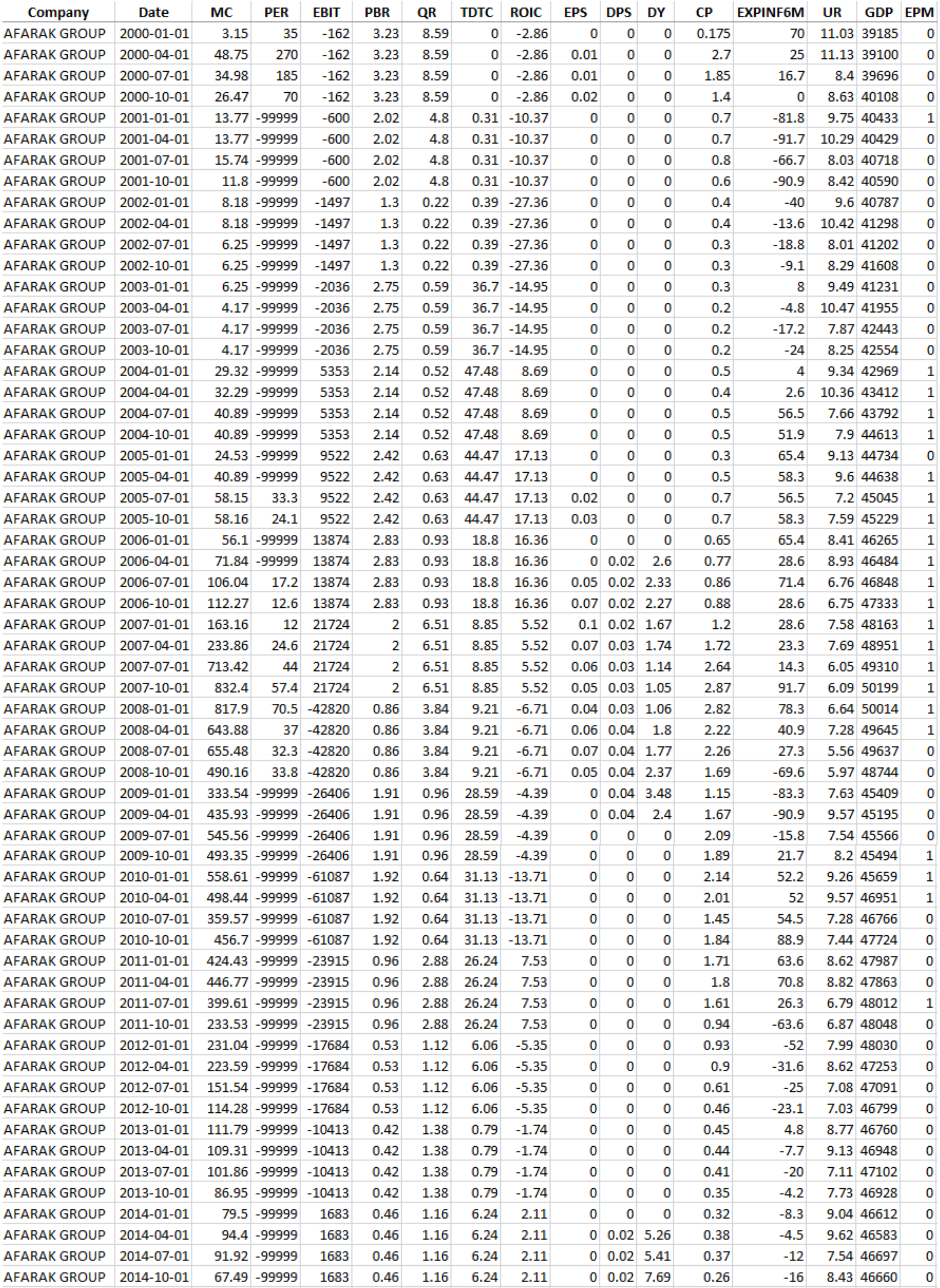
```
recall=[]
ypreds=[]
gammas=[0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1]
Cvalues=[0.5, 1, 1.05, 1.5, 2]

for val in gammas: # Cvalues list can be looped over here as well
    svm = SVC(C = 1, kernel = 'rbf', random_state = 0, probability=True, gamma=val)
    svm.fit(X_train, y_train)

    # Predicting the Test set results
    y_pred = svm.predict(X_test)
    ypreds.append(y_pred)
    acc.append(metrics.accuracy_score(y_test, y_pred))
    f1_scores.append(metrics.f1_score(y_test, y_pred))
    pres.append(metrics.precision_score(y_test, y_pred))
    recall.append(metrics.recall_score(y_test, y_pred))

# Results
print(f"Accuracy: {np.max(acc):.4f} with gamma={gammas[np.argmax(acc)]}")
print("F1-score", f1_scores[np.argmax(acc)])
print("Precision:", pres[np.argmax(acc)])
print("Recall:", recall[np.argmax(acc)])

# Making the Confusion Matrix
print(confusion_matrix(y_test, ypreds[np.argmax(acc)]))

# SVM with different gamma values
plt.figure(figsize=(10,7))
plt.grid(linestyle='-', linewidth=1, axis='y')
plt.plot(gammas, acc)
plt.scatter(gammas[np.argmax(acc)], max(acc), c='r', label=f"Highest accuracy: {max(acc)*100:.2f}%")
plt.xlabel('Different gamma values',  fontsize='large')
plt.ylabel('Prediction accuracy', fontsize='large')
plt.title('Different SVM models')
plt.legend()
plt.xlim(xmin=0, xmax=0.102)
plt.show()
```

```
# the best SVM model:
svm = SVC(kernel = 'rbf', random_state = 0, probability=True, gamma=gammas[np.argmax(acc)])
svm.fit(X_train, y_train)

# Decision Tree
# Fitting Decision Tree Classification to the Training set
dtree = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
dtree.fit(X_train, y_train)

# Predicting the Test set results
y_pred = dtree.predict(X_test)

# Making the Confusion Matrix
print(confusion_matrix(y_test, y_pred))

# Results
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("F1-score", metrics.f1_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))

# Random Forest
acc = []
f1_scores=[]
pres=[]
recall=[]
ypreds=[]
for n in range(1,31):
   classifier = RandomForestClassifier(n_estimators = n, criterion = 'entropy', random_state = 0)
   classifier.fit(X_train, y_train)

   # Predicting the Test set results
   y_pred = classifier.predict(X_test)
   ypreds.append(y_pred)
   acc.append(metrics.accuracy_score(y_test, y_pred))
   f1_scores.append(metrics.f1_score(y_test, y_pred))
   pres.append(metrics.precision_score(y_test, y_pred))
   recall.append(metrics.recall_score(y_test, y_pred))
```

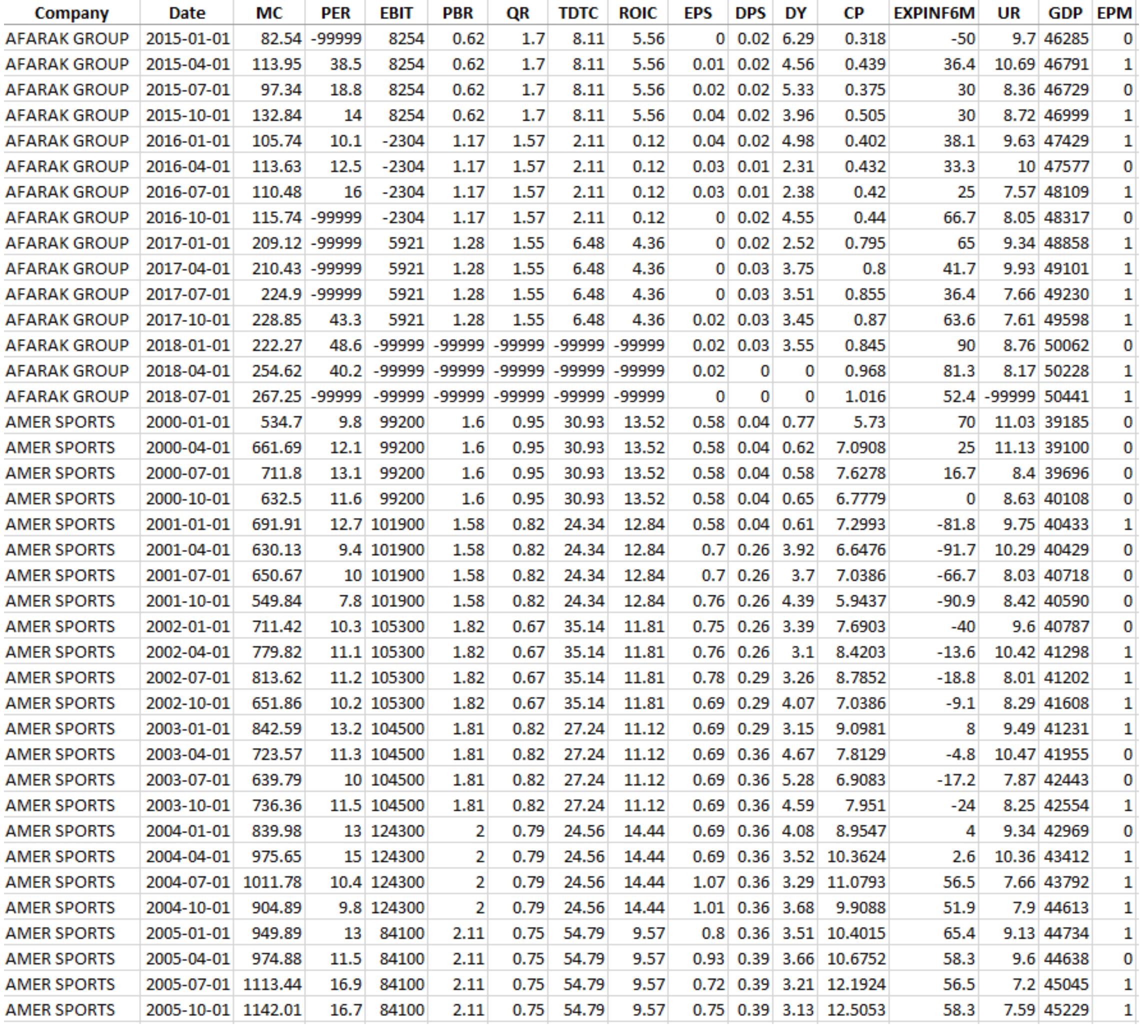
```
# Predicting the Test set results
print(f"Accuracy: {np.max(acc):.4f} with n_estimators={np.argmax(acc)+1}")
print("F1-score", f1_scores[np.argmax(acc)])
print("Precision:", pres[np.argmax(acc)])
print("Recall:", recall[np.argmax(acc)])

# Making the Confusion Matrix
print(confusion_matrix(y_test, ypreds[np.argmax(acc)]))

# Best random forest Classifier
rfor = RandomForestClassifier(n_estimators = np.argmax(acc)+1, criterion = 'entropy', random_state = 0)
rfor.fit(X_train, y_train)

nestimators = list(range(1,31))
plt.figure(figsize=(10,7))
plt.grid(linestyle='-', linewidth=1, axis='y')
plt.plot(nestimators, acc)
plt.scatter(np.argmax(acc)+1, max(acc), c='r', label=f"Highest accuracy: {max(acc)*100:.2f}%")
plt.xticks(np.arange(min(nestimators), max(nestimators)+1, 2.0))
plt.xlabel('Number of estimators', fontsize='large')
plt.ylabel('Prediction accuracy', fontsize='large')
plt.ylim(ymin=0.55, ymax=0.67)
plt.title('Different random forest models')
plt.legend()
plt.show()

# KNN
# Fitting different KNN models to the Training set & making predictions
acc = []
f1_scores=[]
pres=[]
recall=[]
ypreds=[]
for k in range(1,31):
    classifier = KNeighborsClassifier(n_neighbors = k, metric = 'minkowski', p = 2)
    classifier.fit(X_train, y_train)
    # Predicting the Test set results
    y_pred = classifier.predict(X_test)
    ypreds.append(y_pred)
    acc.append(metrics.accuracy_score(y_test, y_pred))
    f1_scores.append(metrics.f1_score(y_test, y_pred))
    pres.append(metrics.precision_score(y_test, y_pred))
    recall.append(metrics.recall_score(y_test, y_pred))

# Predicting the Test set results
print(f"Accuracy: {np.max(acc):.4f} with k={np.argmax(acc)+1}")
print("F1-score", f1_scores[np.argmax(acc)])
print("Precision:", pres[np.argmax(acc)])
print("Recall:", recall[np.argmax(acc)])

# Making the Confusion Matrix
print(confusion_matrix(y_test, ypreds[np.argmax(acc)]))

# best KNN classifier
knn = KNeighborsClassifier(n_neighbors = np.argmax(acc)+1, metric = 'minkowski', p = 2)
knn.fit(X_train, y_train)

# Accuracy with different k values
kvalues = list(range(1,31))
plt.figure(figsize=(10,7))
plt.grid(linestyle='-', linewidth=1, axis='y')
plt.plot(kvalues, acc)
plt.scatter(np.argmax(acc)+1, max(acc), c='r', label=f"Highest accuracy: {max(acc)*100:.2f}%")
plt.xticks(np.arange(min(kvalues), max(kvalues)+1, 2.0))
plt.xlabel('Number of neighbors',  fontsize='large')
plt.ylabel('Prediction accuracy', fontsize='large')
plt.title('Different KNN models')
plt.legend()
plt.show()

# ROC curve
roc_auc1 = roc_auc_score(y_test, logit.predict(X_test))
fpr1, tpr1, thresholds1 = roc_curve(y_test, logit.predict_proba(X_test)[:,1])

roc_auc2 = roc_auc_score(y_test, svm.predict(X_test))
```

```
fpr2, tpr2, thresholds2 = roc_curve(y_test, svm.predict_proba(X_test)[:,1])

roc_auc3 = roc_auc_score(y_test, dtree.predict(X_test))
fpr3, tpr3, thresholds3 = roc_curve(y_test, dtree.predict_proba(X_test)[:,1])

roc_auc4 = roc_auc_score(y_test, rfor.predict(X_test))
fpr4, tpr4, thresholds4 = roc_curve(y_test, rfor.predict_proba(X_test)[:,1])

roc_auc5 = roc_auc_score(y_test, knn.predict(X_test))
fpr5, tpr5, thresholds5 = roc_curve(y_test, knn.predict_proba(X_test)[:,1])

fig = plt.figure(figsize=(7,7))
plt.plot(fpr1, tpr1, label= f'Logistic Regression (area = {roc_auc1:.3f})')
plt.plot(fpr2, tpr2, label= f'SVM (area = {roc_auc2:.3f})')
plt.plot(fpr3, tpr3, label= f'Decision Tree (area = {roc_auc3:.3f})')
plt.plot(fpr4, tpr4, label= f'Random Forest (area = {roc_auc4:.3f})')
plt.plot(fpr5, tpr5, label= f'KNN (area = {roc_auc5:.3f})')
plt.plot([0, 1], [0, 1],'r--', color='black')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate', fontsize='large')
plt.ylabel('True Positive Rate', fontsize='large')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.show()
```

APPENDIX 2 FIRST 100 ROWS OF THE DATA SET



Company	Date	MC	PER	EBIT	PBR	QR	TDTC	ROIC	EPS	DPS	DY	CP	EXPINF6M	UR	GDP	EPM
AFARAK GROUP	2000-01-01	3.15	35	-162	3.23	8.59	0	-2.86	0	0	0	0.175	70	11.03	39185	0
AFARAK GROUP	2000-04-01	48.75	270	-162	3.23	8.59	0	-2.86	0.01	0	0	2.7	25	11.13	39100	0
AFARAK GROUP	2000-07-01	34.98	185	-162	3.23	8.59	0	-2.86	0.01	0	0	1.85	16.7	8.4	39696	0
AFARAK GROUP	2000-10-01	26.47	70	-162	3.23	8.59	0	-2.86	0.02	0	0	1.4	0	8.63	40108	0
AFARAK GROUP	2001-01-01	13.77	-99999	-600	2.02	4.8	0.31	-10.37	0	0	0	0.7	-81.8	9.75	40433	1
AFARAK GROUP	2001-04-01	13.77	-99999	-600	2.02	4.8	0.31	-10.37	0	0	0	0.7	-91.7	10.29	40429	0
AFARAK GROUP	2001-07-01	15.74	-99999	-600	2.02	4.8	0.31	-10.37	0	0	0	0.8	-66.7	8.03	40718	0
AFARAK GROUP	2001-10-01	11.8	-99999	-600	2.02	4.8	0.31	-10.37	0	0	0	0.6	-90.9	8.42	40590	0
AFARAK GROUP	2002-01-01	8.18	-99999	-1497	1.3	0.22	0.39	-27.36	0	0	0	0.4	-40	9.6	40787	0
AFARAK GROUP	2002-04-01	8.18	-99999	-1497	1.3	0.22	0.39	-27.36	0	0	0	0.4	-13.6	10.42	41298	0
AFARAK GROUP	2002-07-01	6.25	-99999	-1497	1.3	0.22	0.39	-27.36	0	0	0	0.3	-18.8	8.01	41202	0
AFARAK GROUP	2002-10-01	6.25	-99999	-1497	1.3	0.22	0.39	-27.36	0	0	0	0.3	-9.1	8.29	41608	0
AFARAK GROUP	2003-01-01	6.25	-99999	-2036	2.75	0.59	36.7	-14.95	0	0	0	0.3	8	9.49	41231	0
AFARAK GROUP	2003-04-01	4.17	-99999	-2036	2.75	0.59	36.7	-14.95	0	0	0	0.2	-4.8	10.47	41955	0
AFARAK GROUP	2003-07-01	4.17	-99999	-2036	2.75	0.59	36.7	-14.95	0	0	0	0.2	-17.2	7.87	42443	0
AFARAK GROUP	2003-10-01	4.17	-99999	-2036	2.75	0.59	36.7	-14.95	0	0	0	0.2	-24	8.25	42554	0
AFARAK GROUP	2004-01-01	29.32	-99999	5353	2.14	0.52	47.48	8.69	0	0	0	0.5	4	9.34	42969	1
AFARAK GROUP	2004-04-01	32.29	-99999	5353	2.14	0.52	47.48	8.69	0	0	0	0.4	2.6	10.36	43412	1
AFARAK GROUP	2004-07-01	40.89	-99999	5353	2.14	0.52	47.48	8.69	0	0	0	0.5	56.5	7.66	43792	1
AFARAK GROUP	2004-10-01	40.89	-99999	5353	2.14	0.52	47.48	8.69	0	0	0	0.5	51.9	7.9	44613	1
AFARAK GROUP	2005-01-01	24.53	-99999	9522	2.42	0.63	44.47	17.13	0	0	0	0.3	65.4	9.13	44734	0
AFARAK GROUP	2005-04-01	40.89	-99999	9522	2.42	0.63	44.47	17.13	0	0	0	0.5	58.3	9.6	44638	1
AFARAK GROUP	2005-07-01	58.15	33.3	9522	2.42	0.63	44.47	17.13	0.02	0	0	0.7	56.5	7.2	45045	1
AFARAK GROUP	2005-10-01	58.16	24.1	9522	2.42	0.63	44.47	17.13	0.03	0	0	0.7	58.3	7.59	45229	1
AFARAK GROUP	2006-01-01	56.1	-99999	13874	2.83	0.93	18.8	16.36	0	0	0	0.65	65.4	8.41	46265	1
AFARAK GROUP	2006-04-01	71.84	-99999	13874	2.83	0.93	18.8	16.36	0	0.02	2.6	0.77	28.6	8.93	46484	1
AFARAK GROUP	2006-07-01	106.04	17.2	13874	2.83	0.93	18.8	16.36	0.05	0.02	2.33	0.86	71.4	6.76	46848	1
AFARAK GROUP	2006-10-01	112.27	12.6	13874	2.83	0.93	18.8	16.36	0.07	0.02	2.27	0.88	28.6	6.75	47333	1
AFARAK GROUP	2007-01-01	163.16	12	21724	2	6.51	8.85	5.52	0.1	0.02	1.67	1.2	28.6	7.58	48163	1
AFARAK GROUP	2007-04-01	233.86	24.6	21724	2	6.51	8.85	5.52	0.07	0.03	1.74	1.72	23.3	7.69	48951	1
AFARAK GROUP	2007-07-01	713.42	44	21724	2	6.51	8.85	5.52	0.06	0.03	1.14	2.64	14.3	6.05	49310	1
AFARAK GROUP	2007-10-01	832.4	57.4	21724	2	6.51	8.85	5.52	0.05	0.03	1.05	2.87	91.7	6.09	50199	1
AFARAK GROUP	2008-01-01	817.9	70.5	-42820	0.86	3.84	9.21	-6.71	0.04	0.03	1.06	2.82	78.3	6.64	50014	1
AFARAK GROUP	2008-04-01	643.88	37	-42820	0.86	3.84	9.21	-6.71	0.06	0.04	1.8	2.22	40.9	7.28	49645	1
AFARAK GROUP	2008-07-01	655.48	32.3	-42820	0.86	3.84	9.21	-6.71	0.07	0.04	1.77	2.26	27.3	5.56	49637	0
AFARAK GROUP	2008-10-01	490.16	33.8	-42820	0.86	3.84	9.21	-6.71	0.05	0.04	2.37	1.69	-69.6	5.97	48744	0
AFARAK GROUP	2009-01-01	333.54	-99999	-26406	1.91	0.96	28.59	-4.39	0	0.04	3.48	1.15	-83.3	7.63	45409	0
AFARAK GROUP	2009-04-01	435.93	-99999	-26406	1.91	0.96	28.59	-4.39	0	0.04	2.4	1.67	-90.9	9.57	45195	0
AFARAK GROUP	2009-07-01	545.56	-99999	-26406	1.91	0.96	28.59	-4.39	0	0	0	2.09	-15.8	7.54	45566	0
AFARAK GROUP	2009-10-01	493.35	-99999	-26406	1.91	0.96	28.59	-4.39	0	0	0	1.89	21.7	8.2	45494	1
AFARAK GROUP	2010-01-01	558.61	-99999	-61087	1.92	0.64	31.13	-13.71	0	0	0	2.14	52.2	9.26	45659	1
AFARAK GROUP	2010-04-01	498.44	-99999	-61087	1.92	0.64	31.13	-13.71	0	0	0	2.01	52	9.57	46951	1
AFARAK GROUP	2010-07-01	359.57	-99999	-61087	1.92	0.64	31.13	-13.71	0	0	0	1.45	54.5	7.28	46766	0
AFARAK GROUP	2010-10-01	456.7	-99999	-61087	1.92	0.64	31.13	-13.71	0	0	0	1.84	88.9	7.44	47724	0
AFARAK GROUP	2011-01-01	424.43	-99999	-23915	0.96	2.88	26.24	7.53	0	0	0	1.71	63.6	8.62	47987	0
AFARAK GROUP	2011-04-01	446.77	-99999	-23915	0.96	2.88	26.24	7.53	0	0	0	1.8	70.8	8.82	47863	0
AFARAK GROUP	2011-07-01	399.61	-99999	-23915	0.96	2.88	26.24	7.53	0	0	0	1.61	26.3	6.79	48012	1
AFARAK GROUP	2011-10-01	233.53	-99999	-23915	0.96	2.88	26.24	7.53	0	0	0	0.94	-63.6	6.87	48048	0
AFARAK GROUP	2012-01-01	231.04	-99999	-17684	0.53	1.12	6.06	-5.35	0	0	0	0.93	-52	7.99	48030	0
AFARAK GROUP	2012-04-01	223.59	-99999	-17684	0.53	1.12	6.06	-5.35	0	0	0	0.9	-31.6	8.62	47253	0
AFARAK GROUP	2012-07-01	151.54	-99999	-17684	0.53	1.12	6.06	-5.35	0	0	0	0.61	-25	7.08	47091	0
AFARAK GROUP	2012-10-01	114.28	-99999	-17684	0.53	1.12	6.06	-5.35	0	0	0	0.46	-23.1	7.03	46799	0
AFARAK GROUP	2013-01-01	111.79	-99999	-10413	0.42	1.38	0.79	-1.74	0	0	0	0.45	4.8	8.77	46760	0
AFARAK GROUP	2013-04-01	109.31	-99999	-10413	0.42	1.38	0.79	-1.74	0	0	0	0.44	-7.7	9.13	46948	0
AFARAK GROUP	2013-07-01	101.86	-99999	-10413	0.42	1.38	0.79	-1.74	0	0	0	0.41	-20	7.11	47102	0
AFARAK GROUP	2013-10-01	86.95	-99999	-10413	0.42	1.38	0.79	-1.74	0	0	0	0.35	-4.2	7.73	46928	0
AFARAK GROUP	2014-01-01	79.5	-99999	1683	0.46	1.16	6.24	2.11	0	0	0	0.32	-8.3	9.04	46612	0
AFARAK GROUP	2014-04-01	94.4	-99999	1683	0.46	1.16	6.24	2.11	0	0.02	5.26	0.38	-4.5	9.62	46583	0
AFARAK GROUP	2014-07-01	91.92	-99999	1683	0.46	1.16	6.24	2.11	0	0.02	5.41	0.37	-12	7.54	46697	0
AFARAK GROUP	2014-10-01	67.49	-99999	1683	0.46	1.16	6.24	2.11	0	0.02	7.69	0.26	-16	8.43	46660	0



Company	Date	MC	PER	EBIT	PBR	QR	TDTC	ROIC	EPS	DPS	DY	CP	EXPINF6M	UR	GDP	EPM
AFARAK GROUP	2015-01-01	82.54	-99999	8254	0.62	1.7	8.11	5.56	0	0.02	6.29	0.318	-50	9.7	46285	0
AFARAK GROUP	2015-04-01	113.95	38.5	8254	0.62	1.7	8.11	5.56	0.01	0.02	4.56	0.439	36.4	10.69	46791	1
AFARAK GROUP	2015-07-01	97.34	18.8	8254	0.62	1.7	8.11	5.56	0.02	0.02	5.33	0.375	30	8.36	46729	0
AFARAK GROUP	2015-10-01	132.84	14	8254	0.62	1.7	8.11	5.56	0.04	0.02	3.96	0.505	30	8.72	46999	1
AFARAK GROUP	2016-01-01	105.74	10.1	-2304	1.17	1.57	2.11	0.12	0.04	0.02	4.98	0.402	38.1	9.63	47429	1
AFARAK GROUP	2016-04-01	113.63	12.5	-2304	1.17	1.57	2.11	0.12	0.03	0.01	2.31	0.432	33.3	10	47577	0
AFARAK GROUP	2016-07-01	110.48	16	-2304	1.17	1.57	2.11	0.12	0.03	0.01	2.38	0.42	25	7.57	48109	1
AFARAK GROUP	2016-10-01	115.74	-99999	-2304	1.17	1.57	2.11	0.12	0	0.02	4.55	0.44	66.7	8.05	48317	0
AFARAK GROUP	2017-01-01	209.12	-99999	5921	1.28	1.55	6.48	4.36	0	0.02	2.52	0.795	65	9.34	48858	1
AFARAK GROUP	2017-04-01	210.43	-99999	5921	1.28	1.55	6.48	4.36	0	0.03	3.75	0.8	41.7	9.93	49101	1
AFARAK GROUP	2017-07-01	224.9	-99999	5921	1.28	1.55	6.48	4.36	0	0.03	3.51	0.855	36.4	7.66	49230	1
AFARAK GROUP	2017-10-01	228.85	43.3	5921	1.28	1.55	6.48	4.36	0.02	0.03	3.45	0.87	63.6	7.61	49598	1
AFARAK GROUP	2018-01-01	222.27	48.6	-99999	-99999	-99999	-99999	-99999	0.02	0.03	3.55	0.845	90	8.76	50062	0
AFARAK GROUP	2018-04-01	254.62	40.2	-99999	-99999	-99999	-99999	-99999	0.02	0	0	0.968	81.3	8.17	50228	1
AFARAK GROUP	2018-07-01	267.25	-99999	-99999	-99999	-99999	-99999	-99999	0	0	0	1.016	52.4	-99999	50441	1
AMER SPORTS	2000-01-01	534.7	9.8	99200	1.6	0.95	30.93	13.52	0.58	0.04	0.77	5.73	70	11.03	39185	0
AMER SPORTS	2000-04-01	661.69	12.1	99200	1.6	0.95	30.93	13.52	0.58	0.04	0.62	7.0908	25	11.13	39100	0
AMER SPORTS	2000-07-01	711.8	13.1	99200	1.6	0.95	30.93	13.52	0.58	0.04	0.58	7.6278	16.7	8.4	39696	0
AMER SPORTS	2000-10-01	632.5	11.6	99200	1.6	0.95	30.93	13.52	0.58	0.04	0.65	6.7779	0	8.63	40108	0
AMER SPORTS	2001-01-01	691.91	12.7	101900	1.58	0.82	24.34	12.84	0.58	0.04	0.61	7.2993	-81.8	9.75	40433	1
AMER SPORTS	2001-04-01	630.13	9.4	101900	1.58	0.82	24.34	12.84	0.7	0.26	3.92	6.6476	-91.7	10.29	40429	0
AMER SPORTS	2001-07-01	650.67	10	101900	1.58	0.82	24.34	12.84	0.7	0.26	3.7	7.0386	-66.7	8.03	40718	0
AMER SPORTS	2001-10-01	549.84	7.8	101900	1.58	0.82	24.34	12.84	0.76	0.26	4.39	5.9437	-90.9	8.42	40590	0
AMER SPORTS	2002-01-01	711.42	10.3	105300	1.82	0.67	35.14	11.81	0.75	0.26	3.39	7.6903	-40	9.6	40787	0
AMER SPORTS	2002-04-01	779.82	11.1	105300	1.82	0.67	35.14	11.81	0.76	0.26	3.1	8.4203	-13.6	10.42	41298	1
AMER SPORTS	2002-07-01	813.62	11.2	105300	1.82	0.67	35.14	11.81	0.78	0.29	3.26	8.7852	-18.8	8.01	41202	1
AMER SPORTS	2002-10-01	651.86	10.2	105300	1.82	0.67	35.14	11.81	0.69	0.29	4.07	7.0386	-9.1	8.29	41608	1
AMER SPORTS	2003-01-01	842.59	13.2	104500	1.81	0.82	27.24	11.12	0.69	0.29	3.15	9.0981	8	9.49	41231	1
AMER SPORTS	2003-04-01	723.57	11.3	104500	1.81	0.82	27.24	11.12	0.69	0.36	4.67	7.8129	-4.8	10.47	41955	0
AMER SPORTS	2003-07-01	639.79	10	104500	1.81	0.82	27.24	11.12	0.69	0.36	5.28	6.9083	-17.2	7.87	42443	0
AMER SPORTS	2003-10-01	736.36	11.5	104500	1.81	0.82	27.24	11.12	0.69	0.36	4.59	7.951	-24	8.25	42554	1
AMER SPORTS	2004-01-01	839.98	13	124300	2	0.79	24.56	14.44	0.69	0.36	4.08	8.9547	4	9.34	42969	0
AMER SPORTS	2004-04-01	975.65	15	124300	2	0.79	24.56	14.44	0.69	0.36	3.52	10.3624	2.6	10.36	43412	1
AMER SPORTS	2004-07-01	1011.78	10.4	124300	2	0.79	24.56	14.44	1.07	0.36	3.29	11.0793	56.5	7.66	43792	1
AMER SPORTS	2004-10-01	904.89	9.8	124300	2	0.79	24.56	14.44	1.01	0.36	3.68	9.9088	51.9	7.9	44613	1
AMER SPORTS	2005-01-01	949.89	13	84100	2.11	0.75	54.79	9.57	0.8	0.36	3.51	10.4015	65.4	9.13	44734	1
AMER SPORTS	2005-04-01	974.88	11.5	84100	2.11	0.75	54.79	9.57	0.93	0.39	3.66	10.6752	58.3	9.6	44638	0
AMER SPORTS	2005-07-01	1113.44	16.9	84100	2.11	0.75	54.79	9.57	0.72	0.39	3.21	12.1924	56.5	7.2	45045	1
AMER SPORTS	2005-10-01	1142.01	16.7	84100	2.11	0.75	54.79	9.57	0.75	0.39	3.13	12.5053	58.3	7.59	45229	1

APPENDIX 3 SUMMARY OF LOGISTIC REGRESSION WITH FINANCIAL RATIOS

Summary of logistic regression with financial ratios						
Date: 2018-12-13			Pseudo R-squared: 0.048			
No. Observations: 5475			AIC: 7040.9418			
Dependent variable: EPM			BIC: 7120.2371			
Df Model: 11			Log-Likelihood: -3508.5			
Df Residuals: 5463			LL-Null: -3684.7			
Converged: 1.0000			LLR p-value: 7.0448e-69			
No. Iterations: 16.0000			Scale: 1.0000			
Variable	**Coef.**	**Std.Err.**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	0.2776	0.0616	4.5055	0.0000	0.1568	0.3983
MC	0.0000	0.0000	-3.3783	0.0007	0.0000	0.0000
PBR	0.0000	0.0000	-3.2854	0.0010	0.0000	0.0000
PER	0.0000	0.0000	12.9092	0.0000	0.0000	0.0000
QR	0.0000	0.0000	-0.0413	0.9671	0.0000	0.0000
EBIT	0.0000	0.0000	3.6825	0.0002	0.0000	0.0000
EPS	0.0001	0.0007	0.1421	0.8870	-0.0013	0.0015
DPS	0.4394	0.0926	4.7474	0.0000	0.2580	0.6209
DY	-0.1607	0.0147	-10.9555	0.0000	-0.1895	-0.1320
CP	-0.0037	0.0018	-2.0773	0.0378	-0.0072	-0.0002
ROI	0.0000	0.0000	2.1710	0.0299	0.0000	0.0000
TDTC	0.0000	0.0000	2.7606	0.0058	0.0000	0.0000