Applying Artificial Intelligence on Momentum

An empirical study testing whether machine learning can exploit time-series patterns to generate abnormal profits

Copenhagen Business School

M.Sc. in Finance & Investments

May 2018

Master's Thesis

Author: Martin Brenøe
Student ID: 32173

Supervisor: Niklas Kohl

78 pages - characters: 179,130

Abstract

For decades scholars and practitioners have tried to create trading strategies that "beat the market". A more recent addition to the numerous trading strategies trying to create abnormal profits are the so-called quantitative trading strategies based on algorithmic trading.

Through an empirical study, this paper is investigating whether some of the most commonly used machine learning classification algorithms including k-NN, Random Forest and Naive Bayes Classifier are able to generate abnormal returns by exploiting time-series patterns.

The thesis is firstly presenting the results of 16 momentum portfolios, constructed from stocks listed on NYSE from January 1993 to December 2016, based on Jegadeesh and Titman (1993)'s framework. The results presented are in favor of the existence of momentum, as the momentum portfolios realize significant abnormal returns. Our findings therefore support that it is possible to generate abnormal returns based on time-series patterns.

Secondly, a training protocol is conducted where the stocks listed on NYSE from December 1925 to December 2016 are labelled with a class to separate the 25% best- and worst performing stocks each month. The algorithms are then trained on data from December 1925 to December 1992 to predict whether a stock is a "loser", "winner" or "neutral" stock based on its returns in the previous 12 months.

Finally, the algorithms' prediction abilities are tested on the same data-set applied for the momentum strategies. The thesis presents the results of 4 long-short equally weighted portfolios for each classification algorithm over 1-, 3-, 6- and 9-month holding periods. Of the investigated algorithms it is concluded that Random Forest is the best algorithm to predict winners and losers using time-series recognition. Its long-short portfolio with a holding period of 1-month yielded an average monthly return of 0.96% and a Sharpe ratio of 0.91. Moreover, our findings show that the portfolio generated a significant Jensen's alpha, which could not be explained by the additional three factors included in Carhart (1997)'s four factor model. The results from Carhart's four factor model, indicated that both the winner and loser portfolio mainly consisted of small stocks however. The results of a further investigation show, that the long-short portfolio with a holding period of 1 month still yielded abnormal returns when Random Forest was restricted to pick stocks with a market-cap above the 30%-percentile, which increases the robustness of our results.

The thesis concludes that it is possible to apply machine learning algorithms to generate abnormal returns based on historical return data only. However, due to the low prediction accuracy, combined with the fact that the advanced algorithms are 'black boxes', it is also concluded that it would require further research and tests of the algorithms before it would be realistic to apply them for portfolio management.

Acknowledgements

A sincere thank you to Niklas Kohl for providing proficient guidance on the thesis and for sharing both his industry and academic knowledge of trading. Also, a thank you to professor Lundgren and professor LeBaron from Brandeis International Business School for the discussions of respectively momentum trading and machine learning. Lastly, I would like to express my deepest gratitude to my better half for the endless support.

Contents

1 Background **1**

1.1 Motivation & Context . . . 1

1.2 Technical & Fundamental Analysis . . . 2

1.3 Research questions . . . 2

1.4 Scope & Limitations . . . 2

2 Related Research **4**

2.1 Momentum Strategy . . . 4

2.2 Trading with Machine Learning . . . 6

3 Momentum and the Efficient Market Hypothesis **8**

3.1 Efficient Market Hypothesis . . . 8

3.2 Stock Price Development models . . . 9

3.2.1 Martingale . . . 9

3.2.2 Random Walk Hypothesis . . . 9

3.3 Behavioral Finance . . . 10

3.4 Extensions of the momentum trading strategy . . . 11

3.4.1 The January effect . . . 11

3.4.2 Mean reversion . . . 11

4 Theoretical Framework of Machine-Learning **12**

4.1 The concepts of machine learning . . . 12

4.1.1 Training & testing data . . . 12

4.1.2 Supervised Learning . . . 12

4.1.3 Unsupervised Learning . . . 12

4.1.4 Classification . . . 13

4.1.5 Generalization, overfitting and underfitting . . . 13

4.1.6 Decision boundary . . . 14

4.2 Supervised Machine Learning models . . . 15

4.2.1 K-Nearest Neighbors Classifier . . . 15

4.2.2 Naive Bayes Classifier . . . 17

4.2.3 Decision tree . . . 19

4.2.4 Random Forest . . . 20

4.2.5 Kernelized Support Vector Machines (SVM) . . . 21

4.2.6 Neural Networks . . . 23

5 Applied Risk-adjustment Models **24**

5.1 Sharpe-Lintner CAPM . . . 24

5.2 Fama-French three-factor model . . . 24

5.3 Carhart's four-factor model 25
5.4 Underlying assumptions of the OLS estimator 25

6 Validity & Reliability of Data **26**

7 Data & Methodology of Momentum Strategies **27**
7.1 Data 27
7.1.1 Investment Universe - Momentum 27
7.1.2 Fama French Data 28
7.2 Methodology 30
7.3 Risk adjustment 31
7.3.1 No risk-adjustment & Sharpe Ratio 31
7.3.2 CAPM 32
7.3.3 Fama-French three-factor Model 33

8 Results - Momentum strategies **34**
8.1 Momentum results - No risk-adjustments 34
8.2 Momentum results - CAPM 35
8.3 Momentum results - Fama-French three-factor model 39
8.4 Data snooping 42
8.5 Momentum as a trading strategy 43

9 Data & Methodology of Machine Learning Strategies **45**
9.1 Data 45
9.1.1 Justification of the inputs 45
9.1.2 Training Data 45
9.1.3 Test Data 45
9.1.4 Construction of the momentum factor 46
9.2 Machine Learning - Methodology 46
9.2.1 Labelling 46
9.2.2 Setting up the data 47
9.2.3 Construction of portfolios 48
9.2.4 Model evaluation 48

10 Results - Machine Learning strategies **49**
10.1 K-Nearest Neighbor 49
10.2 Naive Bayes Classifier 54
10.3 Random Forest 59
10.3.1 Applied Parameters 59
10.3.2 Discussion of Random Forest Results 61
10.4 Applying machine learning for trading strategies 66

11 Conclusion and Further Implications **70**
11.1 Findings & Conclusion 70
11.2 Further implications 71
11.3 Final remarks 72

References **73**

Appendices **76**

A Our results vs. Jegadeesh & Titman's results **76**

B WML portfolios decomposed **77**

C Lagged Momentum **79**

D Hedged Momentum **79**

E Decision Tree **80**

F Random Forest - Large Cap **82**

G Codes **83**

H Data **83**

1 Background

The following section will first briefly present the background of machine learning and explain its relevance in trading financial assets. Then, we will go through the two most widely acknowledged approaches to asset management, technical- and fundamental analysis, and explain why this thesis is applying "technical"-data only. The objective of the paper will hereafter be formalized to research questions. Lastly, the scope & limitations of this paper will be discussed.

1.1 Motivation & Context

Machine learning is a research field that is combining artificial intelligence (AI) with statistics and computer science. The technology is widely applicable both in everyday life and for commercial use. Today machine learning can mostly be applied for very specific tasks, but it has a large potential. With a market projected to reach $70 billion by 2020 artificial intelligence is going to transform the world as we know it (PWC, 2017; Müller & Guido, 2017).

Machine learning based trading strategies is an area of extensively research and development for portfolio managers and trading companies. Internationally the use of machine learning has already been implemented in a variety of financial institutions, not only doing back-office tasks, but also for trading and portfolio management. Among asset managers, machine learning is mostly used by 'quant funds' - hedge funds that rely heavily on algorithmic trading. A unit applying machine learning and AI (usually referred to as a "AI unit") tends to be a part of a larger team to support with the portfolio construction under an asset managers supervision, and only a small fraction of the trades are estimated to be driven by machine learning, even in the 'quant funds' (Financial Stability Board, 2017).

According to the Financial Stability Board, machine learning and AI based portfolio management companies manage $10 billion in assets, but are growing rapidly. In Denmark the most notable implementation of machine learning was conducted by the asset management company Maj Invest back in 2012. The machine learning algorithm called SinAI was based on the Neural Networks algorithm, which should predict 30 stocks for a long position and 30 stocks to short with a holding period of 3 years. SinAI was shut down in 2014 however, as the long portfolio had gained 40% over 2 years, but the short position had gained 50% which meant a total loss of 10% for the investors (Finanswatch, 2014). Kurt Kara, Head of Global Value Equities at Maj Invest does not seem to have lost faith in artificial intelligence however. In March 2017 Kara stated that portfolio management in the future will be about who can understand and leverage artificial intelligence (Financial Stability Board, 2017; Kara, 2011).

This paper aims to investigate whether machine learning can generate abnormal returns by exploiting time series patterns, which we will show exists through the well-known 'momentum'-factor first shown by Jegadeesh and Titman (1993).

1.2 Technical & Fundamental Analysis

Trading strategies that "beat the market" can be traced back to the beginning of trading in financial assets. Asset managers may use several methods to obtain abnormal profits, but the approaches to systematic investing are usually divided into two categories: *Fundamental* analysis and *technical* analysis (Pedersen, 2015).

One of the most extensively researched topic in academic literature is whether time-series patterns can be exploited to create trading strategies with abnormal profits. The term for predicting stock performance based on historical return data is *technical analysis*. One of the most well-known trading strategies based on technical analysis are the momentum trading strategies, which are strategies taking a long position in the past winners and shorting the past losers. This paper will investigate the momentum trading strategies, and implement the framework in a machine learning setting. Hence, the inputs of the algorithms will be based on what would be characterized as 'technical' inputs. Technical analysis usually also involves studying volatility, high/lows, moving averages, spreads, etc., which we will not investigate (Pedersen, 2015).

Trading based on *fundamental analysis* is the term for the deciding to long or short stocks based on the profitability and growth prospects of each company to its value. For the fundamental analysis to be complete it requires analysis of quality of the company's management, assessment of the reliability of accounting numbers, analyzing the growth prospects of the value drivers, and an valuation by estimating the future cash-flows (Pedersen, 2015).

1.3 Research questions

The overall aim of this paper is to investigate the possibility of applying machine learning to exploit time-series patterns in order to generate abnormal profits.
More specifically the following research questions will be examined:

> Does the well-known momentum strategies provide evidence of the ability to generate abnormal returns through analysis of time series patterns?
> Can machine learning algorithms be trained to predict under- and overperforming stocks, only by the stocks' return data of the past 12 months?
> Can trading strategies constructed from machine learning models generate abnormal returns that cannot be explained by well-known factors?

1.4 Scope & Limitations

Since the above research questions could be answered in several ways, it has been necessary to limit the scope of the paper.

Data:

The investment universe is limited to stocks listed on NYSE in the time period of January 1993 to December 2016. Most scholars usually include NYSE together with AMEX and NASDAQ, but a combination of computational time of the algorithms and the fact that our primary econometric tool Stata has a limit of maximum 32762 variables, made us limit the investment universe to only include stocks listed on NYSE. Note that due to the fact that the momentum strategies with a 12 month formation- and holding period need 24 months lagged return data, we denote the sample period as (end of) December 1994 to (end of) December 2016 for both the momentum and the machine-learning based strategies.

Learning algorithms:

Numerous machine learning algorithms exist that could help answering the research questions. In this thesis we will only include the following:

k-Nearest Neighbors (k-NN)
Decision Tree
Naive Bayes Classifier
Random Forest

The reasoning for using the above algorithms is explained in section 4.

Algorithm inputs:

As discussed in section 1.2, there are several approaches to make effective use of available data to identify new signals on price movements. However, this thesis is concentrated around technical data, as the algorithms will be limited to make predictions of a stock's class only based on its return data the previous 12 months.

In this paper we have used unprocessed data, as the tested algorithms are not very sensitive to the data-properties, and because we only use one type of data. Pre-processing of data typically includes normalizing and re-sampling data. Normalization and re-sampling the data might have increased the performance of the algorithms as later discussed in section 11.2 (Müller & Guido, 2017).

Risk-adjustment & test specifications:

The machine learning based trading strategies are tested on slightly different market models compared to the momentum trading strategies as we add a momentum factor to the Fama and French (1992)'s three factor model, following Carhart (1997)'s four factor model. As the interpretation of the two factor models are almost the same, we omit reporting the results of regressing the machine learning based portfolios on Fama French's three-factor model.

Throughout section 10 we apply "prediction accuracy" as the measurement for sensitivity-testing the machine learning models' specifications. Better measurements could be applied, such as the 3x3 confusion matrix, or optimally by constructing portfolios for each test specification. Due to the time-constraints this was not feasible, but could be interesting for further research.

2 Related Research

This section will firstly review the most notable literature written about one of the most widely debated trading strategies since it was first shown by Jegadeesh and Titman (1993), namely the momentum strategy. Secondly, we present a review of related research of machine learning in trading financial assets and where this study fits in the academic discussion.

2.1 Momentum Strategy

Since the early times of trading, scholars have researched relative strength strategies buying past winners and selling past losers. Levy (1967) argued that a trading strategy that buys stocks with prices significantly higher than the average prices over the previous 27 weeks generated significant abnormal returns. However, Jensen and Bennington (1970) back-tested the strategy and found that it did not consistently outperform a buy and hold strategy outside of Levy's the sample period. Jensen and Bennington (1970) thus argued that Levy's findings were a result of sample selection bias (Levy, 1967; Jensen & Benington, 1970; Jegadeesh & Titman, 1993).

More recently, Jegadeesh and Titman (1993) presented trading strategies based on buying stocks, which have performed well in the past and shorting stocks that have performed poorly in the same preceding period. More specifically, they presented results showing that stocks listed on the New York Stock Exchange and American Stock Exchange in the sample period of 1965 to 1989, which had experienced high returns over the past one to four quarters on average performed better in the following 1 to 4 quarters than the stocks that had obtained low returns, which made it possible to create long-short trading strategies yielding abnormal returns. They applied overlapping holding periods and rebalanced the portfolios every month to maintain equally weighted portfolios. Jegadeesh and Titman (1993) presented results showing that 15 of the 16 long-short trading strategies yielded returns significantly different from zero. Moreover, they presented the results of regressing the momentum portfolio with a 6-month formation- and holding-period on CAPM. The portfolio generated a significant Jensen's alpha of 0.95% and a negative beta of -0.08. Jegadeesh & Titman concluded that the profitability of the trading strategies were related to market underreaction to firm-specific information. These results have been well accepted, but the interpretation of the evidence and the source of the profits have been widely debated (Jegadeesh & Titman, 1993).

Following the findings of Jegadeesh and Titman (1993), Conrad and Kaul (1998) analyzed 8 momentum trading strategies, longing past winners and shorting past losers, with equally long formation- and holding periods between 1 week and 36 months in different time periods within 1926 and 1989 using NYSE & AMEX stock data. In total 55 out of 120 of their long-short trading strategies yielded statistical significant profits. The results they presented supported the findings of Jegadeesh and Titman (1993), as they showed that the momentum strategy often yielded a significant return with a holding period between 3 to 12 month, except from 1926 to

1947. However, Conrad and Kaul (1998) argued that it is important to decompose the profits of the trading strategies, since trading strategies based on time-series patterns are based on the premise that stocks do not follow random walks. Conrad and Kaul (1998) claimed that trading based on past performance of the stock contains two components. The first part comes as a result of time series patterns in returns and the second part is from cross-sectional variation in the mean returns. They argued that if the source of the momentum strategies profit is in the cross-sectional component, the profit would arise even if stock prices were completely unpredictable and follow a random walk, because the momentum strategy on average is buying stocks with high-mean returns and selling stocks with low mean returns. Hence the cross-sectional spread would ensure the momentum strategy to be profitable on average even when the stocks follow a random walk. The results of decomposing the profits showed that cross-sectional variation in mean returns was an important source of the profitability and they could not reject that it explains the momentum strategy. Conrad and Kaul (1998) hence claimed that the findings suggested that the profitability of the momentum effect was not induced by market inefficiencies. However, their study is based on a strong assumption that means are stationary (Conrad & Kaul, 1998).

Jegadeesh and Titman (2001) re-investigated the momentum strategies. Firstly, they addressed the results of Conrad and Kaul (1998) together with alternative interpretations of the momentum strategies that had emerged after the publication. The most notable interpretation of Jegadeesh and Titman (1993)'s results was based on behavioral models, which were interpreting the abnormal returns as a product of inherent biases in the way investors interpret information. The behavioral models were implying that the momentum strategies works because of delayed overreactions, which eventually would be reversed. Jegadeesh and Titman (2001) thus argued that Conrad and Kaul (1998) and the behavioral models were making contradicting explanations of the momentum strategies: The behavioral models predicted that the stock price should revert back to its fundamental value implying that the returns should be negative after the holding period, while the hypothesis of Conrad and Kaul (1998) would imply that the momentum strategies yielded positive returns on average in any period after the formation. Jegadeesh and Titman (2001) found that the momentum strategies worked with a holding period up until 12 months, but from 13 to 60 months after the formation, the returns were negative on average. Jegadeesh and Titman (2001) thus argued that this result was consistent with the behavioral models but conflicting with Conrad and Kaul (1998)'s hypothesis.
Secondly, Jegadeesh and Titman (2001) addressed the critique from Fama and French (1996) among others, who stated that the results could be due to data snooping. Jegadeesh and Titman (2001) showed that the momentum strategies worked out-of-sample in the sample period between 1982 and 1998, which indicated that the initial results of Jegadeesh and Titman (1993) were not due to datasnooping.
Thirdly, Jegadeesh and Titman (2001) addressed the critics stating that the momentum portfolios were not properly risk-adjusted as the momentum strategies were merely a result of taking more risk. To address this issue they applied Capital Asset Pricing Model (CAPM) and Fama

and French (1992)'s three-factor model on the momentum strategy with a formation- and holding period of 6 months in order to risk-adjust the returns. The results indicated that both the winner- and loser portfolio had a market beta above 1, but that the loser had a higher market beta, which implied that the momentum portfolio had a negative market beta. The results of regressing the portfolios on Fama-French's three factor indicated that both the loser- and the winner portfolios consisted mainly of small stocks, but that the loser portfolio was loading more on the size factor. Moreover, the loser portfolio was also loading more on the High-Minus-Low factor, which implied that the momentum portfolio had a negative loading on all three factors. Most importantly, they showed that both the CAPM alpha and the Fama French alpha of the momentum portfolio were significantly different from zero. Hence, the results they presented indicated that the profit of the momentum strategy were robust to risk-adjustments (Jegadeesh & Titman, 2001).

One of the most recent publications investigating the momentum trading strategy is by Asness, Moskowitz, and Pedersen (2013), who examined both momentum and value strategies, and even more interestingly, investigated them jointly. Asness et al. (2013) claimed to find consistent evidence of abnormal returns on value and momentum strategies in 8 different geographies, not only in common stocks as is usually the subject of studies, but also in government bonds and commodities. An interesting finding, which also will be discussed in section 8.3, is that they consistently found a negative correlation between value- and momentum strategies. Moreover, they showed that longing both the momentum- and value strategies increased the abnormal returns, compared to investing in either strategy individually. They found however, that liquidity can partially explain the abnormal returns of the momentum and value trading strategies jointly (Asness et al., 2013).

Daniel and Moskowitz (2015) presented results which shows that momentum portfolios usually experience large drawdowns when the market rebounds after a crisis, and that these momentum "crashes" has happened almost consistently since 1927.

2.2 Trading with Machine Learning

Huerta, Elkan, and Corbacho (2011) presented pioneering results of machine learning algorithms' ability to successfully predict under- and overperforming stocks. They trained a Support Vector Machine-algorithm to identify the 25% highest and 25% lowest performing stocks in terms of volatility adjusted returns based on a set of technical and fundamental features. Next, they tested the algorithm once per month to try to predict the winners and losers of the next 91 days. They then created a long-short investment strategy, longing the predicted winners and shorting the predicted losers every month with a holding period of 91 days in the sample period from 1981 to 2010. The results they presented show that the trading strategy generated a significant yearly Jensen's alpha of 14.86% and a Sharpe ratio of 2.06 (Huerta et al., 2011).

An increasing trend within the research field is to try unconventional inputs for the machine learning algorithms. Recently machine learning algorithms' ability to predict under- and over-performing stocks from sentiment analysis has been an extensively researched topic, following the findings of Bollen and Mao (2011). They investigated whether Twitter sentiment analysis of a company could be applied to identify if its stock price would increase or decrease in value the following day. The sample period was February 28th 2008 to December 19th 2008 and included the constituents of the Dow Jones Industrial Average. They applied two sentiment algorithms, namely OpinionFinder and Google-Profile of Mood States (GPOMS) to label the sentiments of the companies investigated. If the company investigated was referred to in a Tweet, the sentiment would be categorized in one of six categories: "Happy", "Calm", "Alert", "Vital", "Sure" and "kind". The algorithms have restrictions, so only sentences formulated in certain ways would be included to avoid spam and misinterpretation. The algorithms would then identify buzz-words and label it in one of the six categories. So for instance if a Tweet stated "I'm loving the new Coca Cola Light", it would recognize the buzz-word "Loving", and classify the tweet as "Happy" for the Coca Cola stock. They would then count the frequency of the different "moods" and collect them over time. They then trained a Neural Networks algorithm using the six categories as inputs to predict whether the next day's stock price would go up or down in the period, and used the 19 days in December as the test period. The results Bollen and Mao (2011) presented indicated a high correlation between the sentiment on Twitter and the stock development, with a prediction accuracy of 86.7% (Bollen & Mao, 2011).

Imandoust and Bolandraftar (2014) showed that the Decision Tree algorithm could predict stock index directions with a high prediction accuracy based on technical and fundamental inputs jointly. In the paper they presented the results of the Decision Tree, Random Forest, and Naive Bayes Classifier algorithms' ability to predict the direction of the daily stock market index movements of Tehran Stock Exchange (TSE) based on 'technical'-, 'fundamental'-, and 'technical & fundamental jointly' inputs, in the sample period of April 2007 to March 2012. The results indicated that the best prediction accuracy is achieved by combining technical and fundamental data. Moreover, they showed that the Decision Tree model outperformed Random Forest and Naive Bayes Classifier in terms of prediction accuracy. The Decision Tree had an accuracy of 80.08%, whereas Random Forest and Naive Bayes Classifier obtained an accuracy of respectively 78.81% and 73.84%. It should be noted that a prediction accuracy of 73.84% is still considered to be high however (Imandoust & Bolandraftar, 2014; Müller & Guido, 2017).

The application of machine learning in stock prediction is still a limited field of research, but is vastly growing due to more powerful computers. This paper will, like Imandoust and Bolandraftar (2014), compare several machine learning algorithms. However, this paper will focus on machine learning algorithms' ability to predict under- and over-performing stocks and unlike other studies, we will test whether the abnormal returns can be captured by well-known factors (Brynjolfsson & Mcafee, 2017).

3 Momentum and the Efficient Market Hypothesis

The purpose of this paper is to find and exploit systematic time-series anomalies within the stock market, which means that we implicitly are looking for results that are contradicting the efficient market hypothesis (EMH), as the efficient market is defined as a market where security prices always 'fully reflect' all available information. The paper will refrain from answering whether the market is efficient or not however, as we do not know if the market models we apply capture all the risk-premiums in the market, which is known as the *joint hypothesis* problem (Fama, 1970; Campbell et al., 2011; Pedersen, 2015).
In this section we will firstly describe the efficient market hypothesis, followed by a discussion of the most notable stock price development models. Thereafter, we briefly describe some of the most acknowledged explanations of the existence of the momentum effect. Lastly, we will describe two extensions to the momentum strategy.

3.1 Efficient Market Hypothesis

Even though Jegadeesh and Titman (1993, 2001)'s findings were considered to be a strong challenge to the highly regarded efficient market hypothesis (EMH), many still believe that the hypothesis holds (Cuthbertson & Nitzsche, 2004).
The efficient market hypothesis is based on the idea that all traders are rational. The rational traders incorporate any information that is relevant to the price of the assets and adjust the prices accordingly. Hence, by this definition only new and unanticipated information can cause stock prices to change, which would make it impossible to create abnormal profits from trading on time-series patterns (Cuthbertson & Nitzsche, 2004). Thus, according to the efficient market hypothesis, trading strategies based on time-series patterns should not yield abnormal returns sustainably, as the efficient market hypothesis implies that the forecast errors of the stock prices should be zero on average (Cuthbertson & Nitzsche, 2004):

$$\epsilon_{t+1} = P_{t+1} \quad E_t(P_{t+1}) = 0 \tag{1}$$

Moreover, the forecast errors should be uncorrelated with the information available at time t or earlier, known as rational expectations (Cuthbertson & Nitzsche, 2004):

$$E(P_{t+1} \quad E_t(P_{t+1})) = E(\epsilon_{t+1}) = 0 \tag{2}$$

If the momentum trading strategies yield abnormal returns because of time-series patterns it implies that the error term ϵ_t of the forecasts are positively serially correlated, as it means that the expected returns are affected by the past returns. This is clearly a violation of the Efficient Market Hypothesis, as it violates both equation (1) and (2) (Cuthbertson & Nitzsche, 2004).

3.2 Stock Price Development models

3.2.1 Martingale

A Martingale is defined as a stochastic variable where the best forecast of all future values X_{t+j} are equal to the current value X_t, conditioned on the available information up to time t (Cuthbertson & Nitzsche, 2004):

$$E[X_t \mid \Omega_t] = X_t \tag{3}$$

This implies a “fair game”, which is also called as a martingale difference:

$$E[X_{t+1} - X_t \mid \Omega_t] = 0 \tag{4}$$

If expected stock returns are significantly different from zero, it conflicts with the condition of the martingale, that the expected price changes are zero based on the current information set up to time t as stated above. This can be shown mathematically:

$$E[E[P_{t+1}] - P_t \mid P_t, P_{t-1}, \ldots] = 0 \tag{5}$$

(Martingale)

$$E[E[P_{t+1}] - P_t \mid P_t, P_{t-1}, \ldots] \neq 0 \tag{6}$$

(Expected price changes are different from zero)

3.2.2 Random Walk Hypothesis

If the following equation holds true, stocks are said to follow a random walk with a drift μ: (Cuthbertson & Nitzsche, 2004):

$$X_{t+1} = \mu + X_t + \epsilon_{t+1} \tag{7}$$

X_{t+1} is a martingale and the $X_{t+1} - X_t$ is a fair game when $\mu = 0$. The ϵ_{t+1} is an identically and independently distributed random variable (iid) with the following properties (Cuthbertson & Nitzsche, 2004):

$$E_t[\epsilon_{t+1}] = 0\ ,\ \ E_t[\epsilon_m \epsilon_s \mid X_t] = \begin{bmatrix} \sigma^2 \\ 0 \end{bmatrix} \text{ for } \begin{matrix} m = s \\ m \neq s \end{matrix} \tag{8}$$

The random walk hypothesis is more restrictive than a martingale, as a martingale is not restricting higher conditional moments i.e. σ^2 to be statistically independent, and because the martingale is only restricting the ϵ_m and ϵ_s to be uncorrelated linearly, where the random walk is restricting ϵ_t and ϵ_s to be uncorrelated both linearly and non-linearly.

3.3 Behavioral Finance

The market can only be proclaimed to be inefficient relative to a specific risk-adjustment model, which aims to capture the movements of the stocks. However, as we will see later on, the models are not describing the real world perfectly and thus, there will always be residuals from the models. If these residuals have a systematic pattern it is typically classified as anomalies. Behavioral finance try to explain these anomalies by some variety of irrationality or non-standard preferences. It is important to emphasize that this paper is not trying to explain why the momentum effect exists, but in the following subsection we will briefly go through the most acknowledged explanations of the momentum effect (Cuthbertson & Nitzsche, 2004).

Numerous scholars have sought to explain the momentum anomaly. Most of them tend to be based on psychological factors, stating that investors make systematic mistakes. One of the most persistent explanations is that investors are underreacting to news such as earnings announcements, and overreacting when there is a series of good and bad news (Barberis et al., 1997).

Another interesting theory was developed by Hong and Stein (1999), who stated that the market contains two types of agents, "news-watchers" and "momentum traders". They claimed that news-watchers use fundamentals news about cash-flows and the momentum-traders base their decision on past returns. The underlying argument is that the news-watchers are processing the information with delays, which means that prices underreact to news, which momentum-traders can gain on by trend-chasing (Hong & Stein, 1999; Pedersen, 2015).

Fama and French (1996) argued that the overreaction hypothesis explained in 3.4 can be captured by their three-factor model, as they claimed that the abnormal returns of Bondt and Thaler (1985)'s contrarian strategies disappear when adding the two factors. But the two additional factors were not able to explain the abnormal returns found by Jegadeesh and Titman (1993), and thus not the underreaction explanation. Fama (1997) argued however, that the overreaction and underreaction explanations are not actually contradicting the efficient market hypothesis, based on two arguments: First, he argued that if underreaction is about as frequent as overreaction and the split is random, they are consistent with the efficient market hypothesis. Second, even if the anomalies are so large that it cannot be attributed to chance, the anomalies are still sensitive to methodology, as they tend to disappear when different models are applied to test for expected returns (Fama & French, 1996; Fama, 1997).

Lastly, some authors claim that other types of irrational investor behavior can explain the momentum effect, such as overconfidence and biased self-attribution. One of the most noteworthy is by Daniel, Hirshleifer, and Subrahmanyam (1998), who claimed to find evidence for overconfidence, which implies that traders attribute the ex-post short-term 'winners' that they have picked to be a result of superior trading skills and ex-post 'losers' to be a matter of bad luck. This overconfidence is pushing the prices above the fundamental value, as the investors will purchase more 'winners'. Eventually rational traders will see this diversion between the actual price and the fundamental price, and the gains of the winners will be reversed (Cuthbertson & Nitzsche, 2004; Daniel et al., 1998).

3.4 Extensions of the momentum trading strategy

The literature suggests several approaches to improve the momentum trading strategies. In this sub-section we will briefly explain the most noteworthy extensions to the momentum strategies.

3.4.1 The January effect

In the literature there exists several examples of calendar anomalies, such as the (in)famous Monday effect, day-of-the-week effect and January effect. The January effect is originally referring to the fact that returns in January historically have been unusually high (Fama, 1991). However, in this context the January effect is referring to the fact that winners historically have outperformed losers except for January where losers have outperformed winners on average, first observed by Jegadeesh and Titman (1993). This is a quite interesting observation, as it suggests that momentum strategies will perform better from February to December. The January effect will not be investigated in this paper (Jegadeesh & Titman, 1993; Cuthbertson & Nitzsche, 2004).

3.4.2 Mean reversion

Contrarian strategies, where you short past winners and long past losers, has been a very debated topic since Bondt and Thaler (1985) found that shorting the losers and longing the winners of the past 3- to 5- years, would generate abnormal returns in the subsequent 3-5 years. Moreover, Bondt and Thaler (1985) created an overreaction-hypothesis as an explanation for their findings. There has been extensive evidence against the long term contrarian strategies however. Zarowin (1990) found no evidence for the long-term contrarian strategy, but found that it was due to the size effect. Moreover, as noted earlier, Fama and French (1996) found that the three-factor model captured the long-term contrarian strategy.
Abnormal returns based on weekly- and monthly mean reversion was later found by Conrad and Kaul (1998) however. This result has caused scholars to skip a week or a month between formation- and holding period to increase the performance of the momentum effect e.g. French (2017)'s momentum portfolio, which we will discuss later is skipping the first month after formation. This paper tested whether skipping one month between the formation period and holding period made the momentum trading strategies perform better, reported in in appendix C. The results indicate that higher abnormal returns could be obtained by skipping a month, but will not be investigated further as we follow the framework of Jegadeesh and Titman (1993) (Bondt & Thaler, 1985; Zarowin, 1990; Jegadeesh & Titman, 1993; French, 2017).

4 Theoretical Framework of Machine-Learning

A common view in the portfolio management industry is that for machine learning to be an effective tool, the traders need to have an understanding of the algorithms. Moreover, many funds state that they are not yet comfortable fully automating trading management if they do not understand how a particular prediction is made (Financial Stability Board, 2017).
In this section we will introduce the concepts of machine learning followed by an overview of the supervised machine learning algorithms applied in this paper.

4.1 The concepts of machine learning

4.1.1 Training & testing data

Before implementing any trading strategy it is essential to test whether the trading strategy works. It is not any different for machine learning-based trading strategies. Unfortunately, it is not possible to use the same data applied to train the algorithm to evaluate the success of it, since the machine learning algorithms will always remember the whole data set of which it was trained and thus, for any point in time it will always be able to predict the correct result. Hence, to see if the algorithm will generalize well, we need a data set to train the algorithm, denoted as a training set, and a data set to assess the performance of the algorithm, denoted as a test set, which has data the algorithm has not seen before. The decision of where to split between the training set and the test set is somewhat arbitrary, but most scholars appear to use a test set, which constitutes 25% of the data (Müller & Guido, 2017). It is important that the data of the test set consists of the same type of input as the training set. In this thesis the inputs (features) x_j are continuous numbers: x_j 2 R, and the classes y_n are discrete numbers: y 2 $Y = 1, 2, \ldots, n$ where n denotes the number of classes.

4.1.2 Supervised Learning

There are overall two categories of machine learning, supervised- and unsupervised. Supervised learning is the most widely used and in general the most successful type of machine learning (Müller & Guido, 2017). It is also the type of machine learning that will be applied in this paper. *Supervised* learning refers to the fact that the person training the algorithm acts as a "teacher" for the algorithm by initially feeding the algorithm with both the inputs and correct output for each data point in the training set. The test set is unsupervised however, as the algorithm will receive input data only. Instead, the algorithm finds similarities between the inputs given and what it has learned in the training set to eventually make a prediction of the class label.

4.1.3 Unsupervised Learning

With unsupervised learning, the algorithm is not given any output to validate the prediction success in the training set. The algorithm is simply given the input data, and asked to find patterns in the data, which best separates the outputs (Hastie et al., 2017). Unsupervised learning is way beyond the scope of this paper, and will thus only be touched briefly. The difference between unsupervised learning and supervised learning is illustrated below.

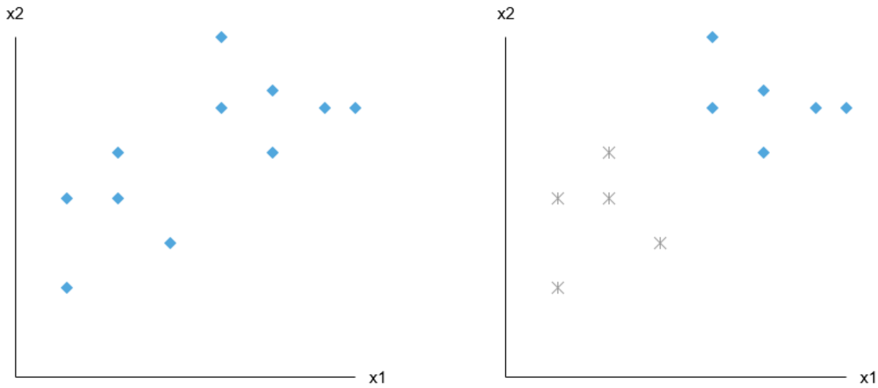

Figure 1: The figure displays the difference between unsupervised problems (left) and supervised problems (right). In the supervised problem, the algorithm will receive the information of the class labels initially in the training set, and can then decide which function separates them best. In the unsupervised problem, the algorithm first has to make a hypothesis of what class label the data points belong to only from the input data

4.1.4 Classification

There are overall two types of supervised machine learning problems. Classification- and regression problems. For regression problems, the goal is to predict a real number. Hence, if the aim of this paper were to predict the exact stock returns, it would be a regression problem. We are not interested in predicting the exact returns however, only whether the stocks are generating a high or low return in the respective period. More specifically the aim of this paper is to identify the underperforming and overperforming stocks every month, and separate them from the rest of the stocks. For a classification problem, the goal is to predict the *class label* from a choice of pre-defined classes. This is thus a classification problem. The possible outputs, which we have denoted winner-, loser- and neutral stocks are called *classes*. Every stock at a given time t belongs to one of these three classes, so this is a three-class classification problem. The winner stocks are labelled (y=1), neutral are labelled as (y=0), and loser stocks are labelled (y=2). The problem in this thesis is thus a supervised classification problem, as opposed to a regression problem (Müller & Guido, 2017).

4.1.5 Generalization, overfitting and underfitting

As discussed in section 4.1.1, the goal of supervised learning is to train a machine learning algorithm on training data, which enables the algorithm to predict on new, unseen data. If the algorithm is able to predict accurately on new unseen data, the model is said to generalize well. To optimize the ability to generalize, two pitfalls related to each other need to be avoided, namely *overfitting* and *underfitting* (Müller & Guido, 2017).
An algorithm can be trained based on many inputs and many sets of rules, which implies that

we build a very complex machine learning model. On the other hand we can also train the algorithm with very few inputs and restrict it to limit the predictions on fewer rules, which will make the model less complex. If we make the model too complex, it is called overfitting. It happens if a model is fitted too closely to the properties of the training set and creates a model that works perfectly for the training set, but is not able to generalize. On the other hand if the model is too simple the model will not capture the heterogeneity in the data, and the model will do badly not only on the new data but also on the training set. Choosing a model too simplistic is called underfitting. In conclusion the more complex the model is, the better it will be able to predict the training data, but if it gets too complex it will be focusing on the individual data points in the training set, and will not create rules that can be generalized to new data. The trade-off is illustrated below (Müller & Guido, 2017).

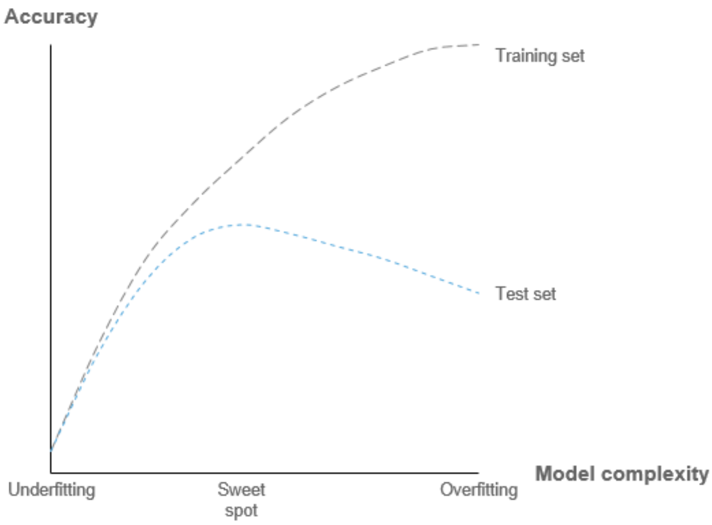

Figure 2: Illustrative example of the trade off between model complexity and accuracy. prediction accuracy is on the y-axis, and complexity is on the x-axis

4.1.6 Decision boundary

In supervised classification problems the algorithm is receiving both the inputs and outputs in the training set, from which the algorithm creates a function that best divides the classes. This function is called a *decision boundary.* The function can also be seen as the hypothesis that the algorithm is making. Intuitively the decision boundary is the "line" where the algorithm divides the inputs. The decision boundary can take many forms as the hypothesis can vary from a linear hypothesis to a non-linear hypothesis based on higher dimensions of the inputs, as seen in figure 3. For the algorithm to be able to make the best possible hypothesis, the correct inputs needs to be selected (Müller & Guido, 2017).

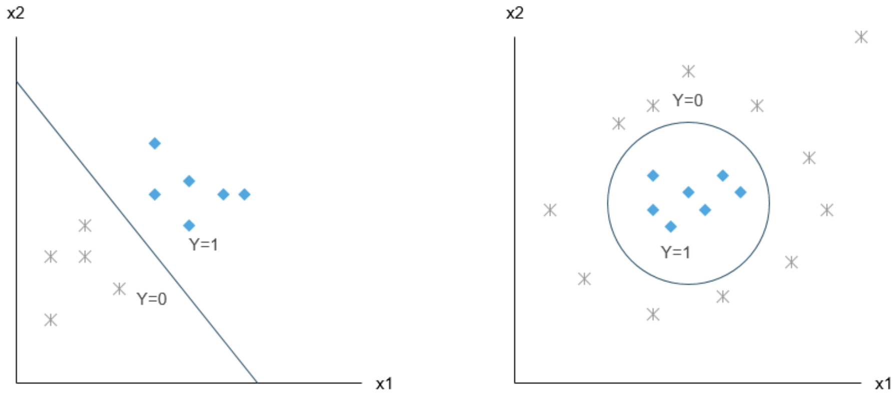

Figure 3: The figure displays a two-dimensional example of how decision boundaries can take different forms, depending on the classification problem. The dark blue line is the decision boundaries which is linear in the graph to the left and a circle in the graph to the right.

4.2 Supervised Machine Learning models

The purpose of this subsection is to provide an overview of how the algorithms included in this thesis work, and to examine the strengths and weaknesses of each algorithm. The following algorithms are investigated in the thesis:

K-Nearest Neighbors (k-NN)
Naive Bayes Classifiers
Decision tree
Random Forest

We will also briefly describe Support Vector Machines and Neural Networks, as these are state of the art machine learning algorithms, which were out of scope as described in section 1.4.

4.2.1 K-Nearest Neighbors Classifier

The k-Nearest Neighbors Classifier (k-NN) algorithm is one of the most simple machine learning algorithms. Storing the training set is all it requires to build the model. To make a prediction in the test set, the algorithm finds the closest data points in the training set, the *nearest neighbors*, and then label the class of the datapoint in the test set based on the majority of the nearest neighbors' class labels.

The k-NN algorithm's standard setting is to only consider one nearest neighbor from the training set, but the model can be extended to k number of neighbors however. When adding more neighbors to the model, the model is using a voting technique to assign the label of the unknown test point. This is illustrated in figure 4. In this paper, the k-NN algorithm counts the number of known loser-, neutral- and winner stocks data points closest to a data point in the test set, and then simply assigns the class that is most frequent (Scikit-Learn, 2017a).

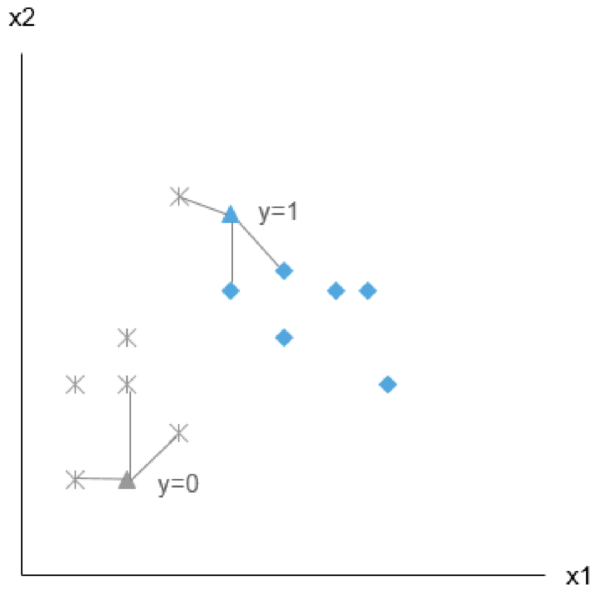

Figure 4: The figure displays a two-dimensional example of how k-Nearest Neighbor with 3 neighbors assigns the class label to the test data points. The triangles are the unknown data points in the test set, and the color represent the class they are labelled

The fewer the neighbors the more the k-NN model will follow the training data. Hence, adding more neighbors will lead to a more smooth decision boundary. A smoother model implies a less complex model. This can be illustrated by the two most extreme cases of the k-NN classifier: If the number of neighbors is set to the amount of data points in the training set, all predictions in the test set would be the same, because each test point would have all the training set's data points as neighbors, and thus it would simply predict the most frequent class in the training set. The other extreme case is using only a single neighbor where the prediction on the training set will be perfect. Going back to the under-fitting over-fitting discussion, it is important to emphasize that adding more neighbors makes the model simpler, which most likely will make the accuracy drop in the training set, however it does not necessarily mean that the prediction accuracy in the test set drops (Müller & Guido, 2017).

4.2.1.1 Parameters

The most important parameters to adjust for the k-Nearest Neighbors Classifier are quite obvious: How to measure the distance between the data points and the number of neighbors. In practice using a small number of neighbors works well, but it is necessary to do sensitive tests of the parameter. Choosing the right distance of the neighbors is beyond the scope of this paper. We apply the Euclidean distance, which is the standard setting. The distance function of Euclidean distance is $\sqrt{\Sigma(x \quad y)^2}$, which is an ordinary straight line distance between the data points. In a two-dimensional setting with a training-point (q1,q2) and a test point (p1,p2) the Euclidean distance would be $\sqrt{((q_1 \quad p_1)^2 + (q_2 \quad p_2)^2)}$ (Scikit-Learn, 2017a; Müller & Guido, 2017).

4.2.1.2 Strengths & weaknesses

The strengths of k-NN is that the model is easy to understand and despite its simplicity, k-NN has proven successful in a number of real-world application (Müller & Guido, 2017; Scikit-Learn, 2017a). As the k-NN model is a non-parametric model it is often successful where the decision boundary is very irregular. The drawbacks of the model is that it can be very slow if the training set is large either because there are a large number of features or because there are many data points. This is especially a big drawback when it comes to trading stocks where timing can be crucial. The model will be applied as the baseline, before getting involved with more advanced machine learning algorithms (Scikit-Learn, 2017a; Müller & Guido, 2017).

4.2.2 Naive Bayes Classifier

Naive Bayes Classifier is a well known algorithm in the machine learning community, and is recognized as a simple but effective algorithm. As discussed in section 4, Imandoust and Bolandraftar (2014) found that Naive Bayes Classifier were able to predict the stock movements with a high prediction accuracy (Imandoust & Bolandraftar, 2014).
Naive Bayes Classifier is based on Bayes' theorem with the "naive" assumption of independence between every pair of features. Bayes Theorem states the following (Scikit-Learn, 2017b):

$$P(y \; j \; x) = \frac{(P(x \; j \; y)P(y))}{P(x)} \tag{9}$$

The $P(y \; j \; x)$ is known as posterior probability. Posterior probability is the probability that it is the class y_n, after receiving the data of the dependent features x_j. In other words, the Naive Bayes Classifier algorithm, will pick the class which has obtained the highest posterior probability given the features it receives. This can be calculated using the naive assumption explained above, $P(x_j \; j \; y, x_1, \ldots, x_n) = P(x_j \; j \; y)$, which makes it possible to reduce the function to (Scikit-Learn, 2017b):

$$\hat{y} = \text{arg max} P(Y) \prod P(x_j \; j \; y) \tag{10}$$

Where P(Y) is the relative frequency of class y in the training set, and $P(x_j \; j \; y)$ is the probability of getting the value of the feature conditioned on it belonging to class y.
To get a better understanding of the algorithm an illustrative example is provided:
The goal is to predict whether a stock will generate a high return (=1) or not (=0) in time $t+1$. Suppose the following is the training set:

$r_{t \ 1}$	*High*	*Neutral*	*Low*	*High*	*High*	*Low*	*Neutral*	*Neutral*	*High*	*Low*
r_t	1	1	0	1	0	0	1	0	1	1

There are 10 stocks in total, $r_{t \ 1}$ is the input and represents the return of the previous month. r_t represents the output, which can be either 1 or 0. If r_t=1, the return in the current month was high, if r_t=0, the return was not high.

We then create statistics of each classes:

	High r_t	Not high r_t	*total*
High $r_{t\;\;1}$	3	1	4
Neutral $r_{t\;\;1}$	2	1	3
Low $r_{t\;\;1}$	1	2	3
total	6	4	10

which means

	High r_t	Not high r_t	$P(x)$
High $r_{t\;\;1}$	3	1	0.4
Neutral $r_{t\;\;1}$	2	1	0.3
Low $r_{t\;\;1}$	1	2	0.3
$P(Y)$	0.6	0.4	

Suppose the return in time t was low, and we now want to predict whether the return in $t+1$ is going to be high or not. Then it would be possible using equation (9) to calculate the probability of getting a high return applying Naive Bayes Theorem:

$$P(\text{high } r_{t+1}\, j \text{ low} r_t) = P(\text{low} r_{t\;\;1}\, j \text{ high } r_t) \quad \frac{\text{High } r_t}{\text{Low} r_{t\;\;1}}$$

P(Low$_{t\;\;1}$$j$ $High_t$)=(1/6)=16.67%, P(High$_t$)=60% and P(Low)=30%. Which then means that the probability is: 16.67% $\frac{60\%}{30\%}$ = 33.33%. With the same logic you can calculate the probability of the class being "not high" in $t+1$, which in this example would be 37.50%. Hence the Naive Bayes Classifier would predict the stock to generate a "not high" return in $t+1$.

In this paper the algorithm will be given several features, so the algorithm is picking the class with the highest product of the features' conditional probabilities $\prod P(x_j\ j\ y)$, multiplied with the relative frequency of the class $P(Y_n)$ as we can see from equation (10).

4.2.2.1 Parameters

Naive Bayes Classifier can be set to different distribution functions. In this paper, Gaussian distribution will be applied as our inputs will be continuous data (Müller & Guido, 2017). The likelihood of getting a feature value conditioned that it belong to a class is calculated as the following (Scikit-Learn, 2017b):

$$P(x_i\ j\ Y) = \frac{1}{\sqrt{2 \quad \pi \quad \sigma_y^2}} \quad e^{\frac{x_i-\mu_y}{2*\sigma_y^2}} \tag{11}$$

Where π is pi, and σ^2 is the variance of the features that belongs to the class. μ is the average value of the features of the class. All the statistics are collected in the training-set

4.2.2.2 Strengths & weaknesses

Naive Bayes Classifier is fast in training and relatively intuitive to understand. The model is very “naive” however, and the assumptions are usually violated. The model works very well in many settings, and as earlier discussed, Imandoust and Bolandraftar (2014) found that the algorithm could predict stock index movements with a high prediction accuracy (Scikit-Learn, 2017b; Müller & Guido, 2017).

4.2.3 Decision tree

The decision tree algorithm was first introduced by Breiman, Friedman, Stone, and Olshen (1984), and has been used for numerous applications since. In section 4 we described that Imandoust and Bolandraftar (2014) found results, which indicated that the decision tree algorithm could predict stock index movements very accurately.

The decision tree algorithm can be boiled down to a hierarchy of if- & else statements, which means that understanding the decision tree algorithm's reasoning is easy. As illustrated in figure 5 below, the algorithm uses a set of binary rules to identify the class. Each node in the tree represents a threshold that eventually will lead to a terminal node called a leaf, which contains a class. In the machine learning context, the thresholds are called (split) tests (Müller & Guido, 2017; Scikit-Learn, 2017c).

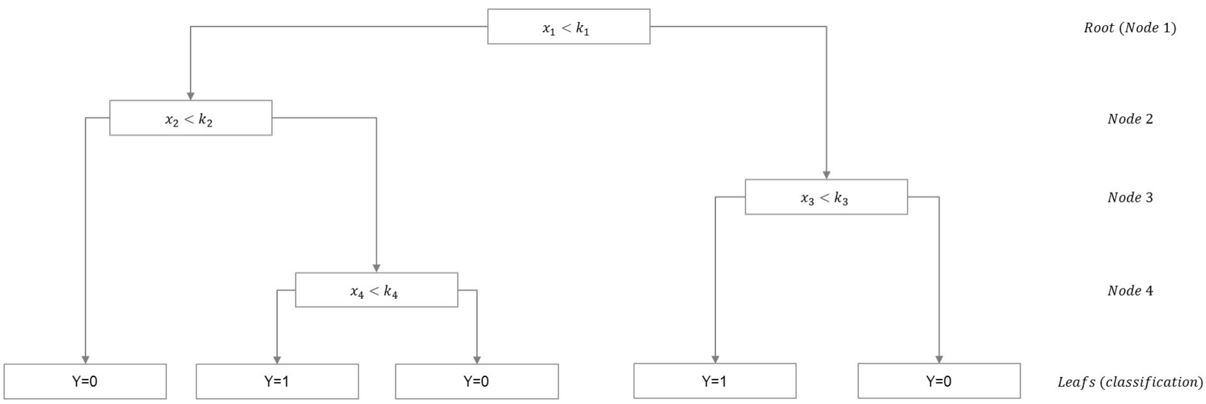

Figure 5: Illustration of how supervised problems can be solved with decision trees

The optimal tree would maximize the information at every single node simultaneously, but there is not a known method for a global optimal decision tree. Instead, the method is to start at the top node, also called the root, and optimize the information at every node individually. This recursive process is eventually creating a tree of binary tests. The process is repeated until all the features in the training set are classified, and each leaf only contains a single class. A leaf that only contains features of one class is called pure (Müller & Guido, 2017). Once the tree has been set up, a data point in the test set can be predicted by starting at the root of the decision tree and traverse the tree until a leaf is reached. The data point will then be labelled with the class of the leaf.

4.2.3.1 Parameters

Building a decision tree where all leafs are pure will most likely imply a too complex model, which will be 100% accurate when applied on the training set. There are several ways to prevent overfitting. The most common ones are called pre-pruning and post-pruning. Post-pruning is where you make the decision where to stop the tree after the tree has been fully grown. The decision is based on the amount of marginal information gained from the extra node. If the nodes contain little to no information you simply remove them. Pre-pruning is where you decide beforehand when to stop the creations of more nodes. Pre-pruning can be constructed in many different ways. In this paper we will only pre-prun by limiting the maximum depth of the tree, which means that only a limited number of consecutive tests can be conducted. If the decision tree is pruned, the leafs might not be pure, and instead the majority class in the leaf will be the determinant of the data point's class (Müller & Guido, 2017; Scikit-Learn, 2017c).

4.2.3.2 Strength & weaknesses of decision trees

The advantage of the decision tree algorithm is that the model is easy to visualize and can be interpreted by people with little statistical knowledge. Moreover, the algorithm is not affected by the type of data it is given, (binary, continuous, etc.), and no normalization or standardization of the data is necessary. The largest drawback is that the decision tree is usually overfitting, even with the pre-pruning tool. However, as already mentioned, Imandoust and Bolandraftar (2014) found that the decision tree showed promising results for predicting the stock index movements (Müller & Guido, 2017; Scikit-Learn, 2017c).
The results of the Decision Tree will only be documented in appendix E, as the trading strategies showed no sign of significance whatsoever, and as we will go through next, Random Forest provides a better alternative.

4.2.4 Random Forest

As discussed above, the decision tree algorithm has a tendency to overfit the training data. To address this problem the Random Forest algorithm was introduced by Breiman (2001). The main idea is to build several decision trees, which all perform very well but also overfit on part of the data. Random Forest is as the name implies, injecting randomness into the tree building process to ensure each tree is different. This is done in two ways; by bootstrapping the data, and by randomizing the available features in each split test. Thus, to build a trading strategy based on Random Forest, the first decision you take is the number of bootstrapped datasets you want create, which is the equivalent to deciding how many decision trees to build. By bootstrapping the training set Breiman (2001) showed that the decision trees were overfitting in different ways, which implied that the overfitting could be reduced by averaging their results while still retaining the performance of the decision trees (Breiman, 2001). Bootstrapping is a technique where new data-sets are created by randomly drawing datapoints with replacement. The advantage compared to e.g. Monte Carlo simulations is that it does not assume normal distribution. This is important because stock returns more often have 'fat tails', meaning that they usually have a kurtosis >3, moreover individual return data is often positively skewed (Munk, 2016). Results from bootstrapping are still limited to a specific model, null hypothesis and sample size however, and it can therefore be hard to generalize from (Cuthbertson & Nitzsche, 2004). Random Forest

has proven to be able to generalize well using the bootstrapping technique in this paper however, which we will see in section 10.
The second decision you need to take is the amount of features the algorithm will have available at every node. Random Forest will then choose the best possible test among the subsets of features, randomly given at every node.
When all the trees are created, a data point's class can be predicted by starting at the root of each tree in the Forest and traverse the trees until a leaf is reached in each tree. All trees are given an equal vote for the final classification of the data point. The class that has received the majority of the votes is then going to be the class that the Random Forest algorithm will predict the data point to be (Müller & Guido, 2017; Breiman, 2001).

4.2.4.1 Parameters

The two most important parameters to adjust are firstly the number of random features that the algorithm is given for each node (*maximum features*) and secondly the number of decision trees (*n_estimators*). The former is important to adjust because it affects the amount of randomness: If we let the number of features be equivalent to the total set of features, no randomness will be injected in the feature selecting and only the bootstrapping process will differentiate the decision trees. On the contrary, if the number of features to choose from are set to 1, the algorithm has no choice, and can only search for the highest information gained from the feature that was randomly selected. Hence, a higher number of features will imply higher similarities between the trees, and a lower number of feature will mean more different decision trees. With few number of features, the trees might grow very deep in order to be able to fit the data reasonable (Müller & Guido, 2017).

4.2.4.2 Strengths & Weaknesses

The Random Forest algorithm is among the most popular machine learning algorithms. It is powerful, and has the advantages of the Decision Tree, but is less biased towards overfitting. The downside is that it is much less intuitive. It is a lot harder to understand Random Forest compared to the decision tree algorithm, since there are several trees, which also grow much deeper. Moreover, with large data samples it takes a lot of computational power and can be time consuming (Müller & Guido, 2017).

4.2.5 Kernelized Support Vector Machines (SVM)

Kernelized Support Vector Machine algorithm (SVM) is out of scope for this paper, but we will briefly describe the algorithm, since Huerta et al. (2011) showed that attractive investment strategies could be created by using SVM to predict stocks as discussed in section 2.
SVM is an algorithm that is able to make non-linear decision boundaries, by using the higher dimensions of the features, where the decision boundary can be linear. This is called "kernelizing". There are several ways to create these high dimensional spaces; one of them is called the polynomial kernel, which computes a limited number of possible polynomials of the features: $x_j^2, x_j^3, \ldots, x_j^c$. When the SVM has found the dimensions that best divides the classes,

the algorithm is ranking the data points after how important they are to represent the decision boundary between the classes. The most important data points are the ones that are on the frontier between the two classes. These data points are called the support vectors. When the SVM is predicting the new data point's class, it measures the distance to each of the support vectors, and label the data point based on the length to the support vectors and the importance of them (Müller & Guido, 2017).

Looking at figure 3 again, the decision boundary to the right is possible to create with the SVM algorithm, which would look somewhat like the gray square in figure 6 below, using higher dimensions of the features (Müller & Guido, 2017).

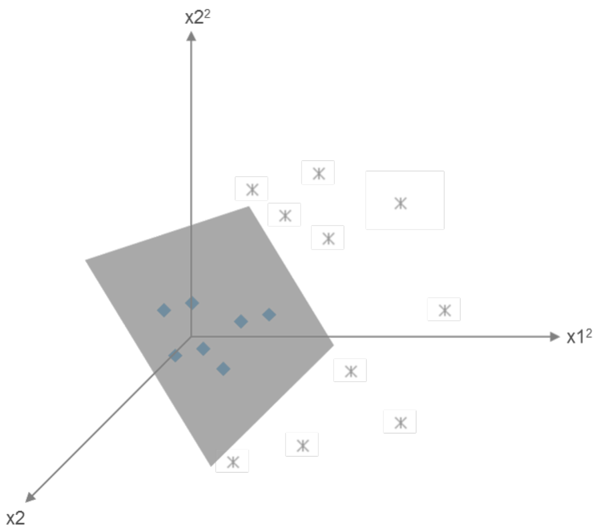

Figure 6: The illustration is displaying how the decision boundary of the figure to the right in figure 3 might look like in a higher dimension.

4.2.5.1 Strengths & Weaknesses

Compared to the models we test in this paper, the Kernelized Support Vector Machine is more powerful, and the decision boundaries are also much more complex. This is a strength, but also a weakness: The algorithm uses more time to calculate and obtain more memory usage than the models tested in the paper. SVM also requires pre-processing of the data and the parameters usually need excessive tuning. Lastly, the algorithm is a black-box, which is a significant drawback for portfolio management, which we mentioned in the beginning of the section. SVM was initially thought to be applied in this paper, as it is one of the most advanced machine learning models available. However, the computation time took more than 7 days, where the algorithm was stopped as it would compromise the rest of the paper (Financial Stability Board, 2017; Müller & Guido, 2017).

4.2.6 Neural Networks

Neural Networks is out of scope of this paper, as discussed in section 1.4, but will briefly be described since the algorithm is considered state-of-the-art, and is probably the most famous machine learning algorithm. Neural Networks has shown some very promising results of predicting stock index movements as discussed in section 2, and the machine learning algorithm that the asset management fund Maj Invest A/S implemented was also based on Neural Networks (Finanswatch, 2014).

The idea of Neural Networks is to mimic the learning patterns of a brain. The obvious advantage of Neural Networks is that it is capable of building extremely complex models. There are drawbacks of the algorithm however: A model build on Neural Networks is usually a black box. Hence, it is very hard to understand which assumptions the model bases its predictions on, as the algorithm uses the inputs in interaction with each other using many hidden layers. Moreover, it requires extensively pre-processing of the data, as the algorithm works best with standardized data, where all the data have the same properties (Müller & Guido, 2017).

5 Applied Risk-adjustment Models

To risk-adjust the trading-strategies several asset-pricing models are applied. In this section the models used to evaluate the performance of the trading strategies will be explained. Firstly, we review the standard Sharpe-Lintner CAPM, followed by the Fama-French three-factor model. Lastly we will go through Carhart's four-factor model, which the machine-learning based training strategies will be regressed on instead of the Fama-French three-factor model as discussed in section 1.4.

5.1 Sharpe-Lintner CAPM

The Sharpe-Lintner Capital Asset Pricing Model (CAPM) can be expressed as the following:

$$(R_i \quad r_f)_t = \alpha_i + \beta_{im}(R_i \quad r_f)_t + \epsilon_{it} \tag{12}$$

Where $\beta_{im} = \frac{cov(r_i;r_m)}{var(r_m)}$. The α_i represents Jensen's alpha, and is by definition assumed to be equal to zero. Hence, β_{im} is the only explanatory variable for the returns according to the CAPM. As beta represents the covariance between the asset returns and the market returns, stocks with high covariance with the market will be expected to generate higher returns compared to stocks with less covariance with the market (Campbell et al., 2011). If $\alpha_i > 0$ on a significant level, the asset i realizes an abnormal return relative to CAPM. Note that when we regress the long-short portfolios we do not subtract the risk-free rate as it is implied that it is done for both the long and the short strategy, which will equal each other out. This applies for all market models (Cuthbertson & Nitzsche, 2004).

5.2 Fama-French three-factor model

Fama and French (1992)'s three-factor model is in Fama and French (1996)'s paper used to test several anomalies found in the literature. As discussed in section 3.3 the three-factor model could not explain the findings of Jegadeesh and Titman (1993), but it is still a relevant model as Fama and French (1996) claimed it captures much of the cross-sectional variation in average stock returns (Fama & French, 1996). Moreover, applying it can help decomposing the trading strategies. The equation used for later regressions is the following:

$$(R_i \quad r_f)_t = \alpha_i + \beta_{i;m}(R_m \quad R_f)_t + s_i \quad SMB_t + h_i \quad HML + \epsilon_{it} \tag{13}$$

The three-factor model suggests that the return of any asset i can be explained by the market beta explained in section 5.1, and two other factors, namely the Small-Minus-Big (SMB) and High-Minus-Low (HML) factors. More specifically, SMB and HML are supposed to capture the abnormal profit that can be obtained by respectively buying small market cap stocks and selling big market cap stocks documented by Banz (1981), and buying stocks with a high book-to-market value while going short in stocks with a low book-to-market value documented by L. K. C. Chan, Hamao, and Lakonishok (1991) (Fama & French, 1992).

Moreover, Fama and French (1992) also claimed that the three-factor model is capturing other documented anomalies, including variables such as E/P and leverage. Thus, by adding these factors to the CAPM, Fama and French (1992) argued that a better measure of a portfolio's returns is obtained. Fama-French's three-factor model implies, as the Sharpe-Lintner CAPM, that $\alpha_i = 0$, and that $\alpha_i > 0$ represents an abnormal return of asset i. A further explanation of the construction of the factor-mimicking portfolios applied is explained in section 7.2.

5.3 Carhart's four-factor model

Carhart (1997) claimed to explain the abnormal returns of many mutual funds with a four-factor model. The model is simply an extension to the Fama French three-factor model that we described above, adding a factor mimicking portfolio based on Jegadeesh and Titman (1993)'s findings of momentum. The model will only be applied on the machine learning portfolios to investigate whether the models are picking stocks based on the momentum effect. Carhart (1997)'s four-factor model is expressed as the following:

$$(R_i \quad r_f)_t = \alpha_i + \beta_{im}(R_m \quad R_f)_t + s_i \quad SMB_t + h_i \quad HML_t + p_{i;t} \quad PR1YR_t + \epsilon_{it} \tag{14}$$

As with the other risk-adjustment models, alpha is assumed to be zero, $\alpha_i = 0$. The construction of the momentum factor will be explained further in the section 9.

5.4 Underlying assumptions of the OLS estimator

As described above, the performance of the trading strategies will be tested using linear regressions. In order for any ordinary least squares (OLS) regression to be unbiased, following assumptions need to hold (Brooks, 2011):

- The variance of the error terms are constant σ^2, meaning no heteroscedacity
- The error terms are independently distributed, meaning no autocorrelation in the error terms $cov(\epsilon_i, \epsilon_j) = 0$, or between the error term and the independent variable $cov(\epsilon_t, x_t)$
- The error terms are normally distributed with a mean equal zero: $\epsilon_t = N(0, \sigma^2)$

If the above assumptions hold true, it implies that $E[\hat{\alpha}] = \alpha$ and $E[\hat{\beta}] = \beta$ (Brooks, 2011).

6 Validity & Reliability of Data

Validity in a quantitative study can be defined as the collected data's degree of relevance for answering the research question (Andersen, 2013).
The collected data for all the trading strategies is assessed to be highly relevant, as it was collected solely for the purpose of answering the research question. The constituents of the data sample might influence the validity however. NYSE consists of a large fraction of small stocks, which implies that the machine learning algorithms will be trained mainly to predict time-series patterns of small stocks. This could potentially reduce the validity if it implies that the algorithms will be more likely to label the small stocks as winners and losers, and large cap stocks as neutral stocks. If this is true, the trading strategies might become unprofitable net of transaction costs. This will be discussed further in section 10.4.
It should also be noted that Python was applied for the machine learning part of the thesis, with the "Scikit-Learn" module. Scikit-Learn is a Python module that has a wide range of state-of-the art machine learning algorithms, but the algorithms are created with a general purpose, and is hence not created with the purpose of stock trading.
Lastly we note that we use overlap of our portfolios, which means that the error terms of the regressions will be serially correlated. This could be affecting the validity of the results. We will assume for the rest of this paper that the standard assumptions of the OLS estimator holds however (Cuthbertson & Nitzsche, 2004).

Reliability is referring to the completeness of data and how precise the concept is measured, including whether the results can been affected by coincidences (Andersen, 2013).
The reliability of the data is assessed to be high, as it was extracted from CRSP, a highly reliable source used by many acknowledged scholars. Moreover, several of the most significant biases has been sought to be avoided. Firstly, it is not possible to see a price, and then decide to trade on it, hence we used the "holding period return" provided by CRSP instead of calculating returns from price data. Secondly, we have included companies, which are not listed today to avoid survivorship bias. Moreover, if a stock was delisted in the month we constructed the portfolio or in the holding period, the stock was still included in the portfolio, as we would not know that the company would be delisted at the time it was constructed. This potential bias is referred to as a look-ahead-bias, which we also have tried to avoid, by only including stocks at the time they were listed and it was possible to trade them. Lastly, the issue of delisting was handled by using the "delisting return" if a stock was delisted. This return takes account of both delisting because of a merger, which would be known beforehand, and a bankruptcy, which might have stopped all trading of a stock immediately. However, two issues could potentially be addressed. Firstly, the constituents in the data slightly changed, depending on the data-collecting method. Secondly, some manual data-cleaning had to be done, which could cause some minor mistakes (Jegadeesh & Titman, 1993; Fama & French, 1996; Pedersen, 2015; Munk, 2016).

7 Data & Methodology of Momentum Strategies

The following section is reviewing the investment universe, followed by a description of how Fama-French's High-Minus-Low and Small-Minus-Big portfolios were constructed. After reviewing the data, we explain the methodology step-by-step to test the existence of the momentum effect first found by Jegadeesh and Titman (1993). Lastly we will discuss how to interpret the results.

7.1 Data

7.1.1 Investment Universe - Momentum

Firstly, monthly return data was gathered from Center for Research in Security Prices (CRSP) using Wharton Research Data Services. More precisely, we collected the 'adjusted monthly holding period return' data for all common stocks listed on New York Stock Exchange (NYSE) from January 1993 to end of December 2016, which included 6357 stocks in total. NYSE data was used due to several reasons: Firstly, the data could be extracted from a reliable source, namely CRSP, and required an insignificant amount of manual cleaning, which is especially important because we are testing the machine learning algorithms on the same data as the momentum portfolios, and clean data is essential in a machine learning setting, which we will explain in section 9.1. Secondly, a robust data-sample could be obtained only using NYSE. thirdly, CRSP has NYSE data available from 1925, which also is important for the tests of the machine learning algorithms. Fourthly, numerous scholars use NYSE data at least as a part of their data sample. Other stock exchanges could have been included as well, but as discussed in 1.4, it would also have had several drawbacks.

The actual test period of the momentum portfolios will be from end of December 1994, exactly 24 months after January 1993, because 12 months of formation returns are needed, and subsequently a 12 month holding period is required to test the 12 month formation- and holding period portfolios. Moreover, we want all the trading strategies in the paper to be tested on the exact same sample period to increase the comparability of the results. Hence, end of December 1994 to end of December 2016 will be denoted as the sample period for all trading strategies.

When regressing the portfolio on CAPM, the value-weighted monthly returns of S&P 500, extracted from CRSP, is applied as the market return. The S&P 500 is the leading stock index in the US, which captures the 500 largest stocks listed on NYSE or NASDAQ, and is often used as a proxy for the market return (Munk, 2016).

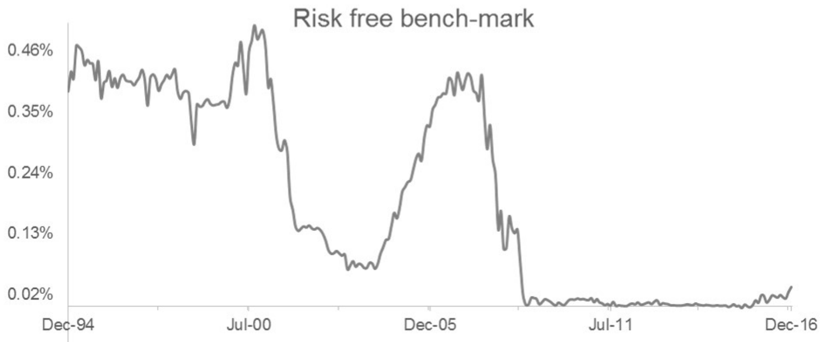

Figure 7: The return of the 1-month Treasury Bill over time

The risk-free return data is the interest rate on the 1-month Treasury Bill also extracted from CRSP, de-annualized using the formula $(1 + r_f)^{1=12}$ 1. This is the same proxy for the risk-free rate as in Jegadeesh and Titman (1993). French (2017) also uses the 1-month Treasury Bill as the proxy for the risk-free rate (French, 2017; Jegadeesh & Titman, 1993).

7.1.2 Fama French Data

Fama-French's three factors were collected on French (2017)'s webpage, instead of creating the portfolios ourselves in order to increase the validity. This subsection will describe how French (2017) constructed the three factors, which follows the methodology reported in Fama and French (1993).

French (2017) has included all common stocks listed on NYSE, AMEX and NASDAQ, however all financial companies are excluded since high leverage is not necessarily a sign of financial distress, which it typically indicates for non-financial companies (Fama & French, 1993).

The HML factor is relying on fundamental data of the companies. To ensure that the companies' fundamentals were publicly available for the respective returns, the SMB and HML factor-returns were calculated from July in time t to June in t+1. The following is a step-by-step description of how the HML and SMB factor-returns were created (Fama & French, 1993):

> The stocks were firstly divided into two groups in the end of June each year, based on their market value of equity (ME). As shown in figure 8, the median of the included stocks' market value of equity was set as the threshold, creating two segments, namely "Small" stocks and "Big" stocks.
>
> Simultaneously, the stocks were broken into three segments based on the their book-to-market value $\frac{BE}{ME}$. More specifically, the stocks were ranked in the end of June$_t$ each year, based on the book value of equity (BE) in the end of December in t 1 over the market value of equity in the end of June in t. The return data was excluded from the period if the book value of equity data was not available at December in t-1. The stocks were then divided into three segments based on their book-to-market value with the 30% percentile as the lower threshold, and 70% percentile as the upper threshold. The stocks with a book-to-market value below the 30% percentile were classified as "Growth" stocks, the stocks with a book-to-market value between the two thresholds were classified as "Neutral" stocks, and the

stocks with a book-to-market value above the top 30% percentile were classified as "Value" stocks. It is noteworthy that Fama and French (1993) mentions that the thresholds are arbitrary, and there has been no attempt to optimize them (Fama & French, 1993).
All the stocks were then divided into 6 portfolios since the stocks would either be "Big" or "Small" while simultaneously be "Growth", "Neutral" or "Value" stocks, as illustrated in figure 8 below. The value-weighted portfolios were created each year in the end of June: $w_{jt} = \frac{MV_{j,July_t}}{MV_{portfolio,July_t}}$, which implies that the six portfolios were rebalanced once per year.

	Median ME	
	Small Value	Big Value
70th percentile	Small Neutral	Big Neutral
30th percentile	Small Growth	Big Growth

Figure 8: The figure displays how the stocks are divided into six value-weighted portfolios each year in the end of June. The two horizontal lines are the 30% and 70% percentile of the BE/ME, and the center vertical line is the median ME of the included stocks.

From the six portfolios, the Small-Minus-Big (SMB) and High-Minus-Low (HML) portfolios were then calculated:

The SMB portfolio was calculated as the average return of the three "Small" portfolios minus the average return of the three "Big" portfolios each month.

$$\begin{aligned} SMB = 1/3(SmallValue + SmallNeutral + SmallGrowth) \\ 1/3(BigValue + BigNeutral + BigGrowth) \end{aligned} \tag{15}$$

The High-Minus-Low factor portfolio was calculated as the average return of the two "Value" portfolios minus the average return of the two "Growth" portfolios each month.

$$HML = 1/2 \quad (SmallValue + BigValue) \quad 1/2 \quad (SmallGrowth + BigGrowth) \tag{16}$$

Regressing the three factors, revealed that the average excess market return ($r_m \quad r_f$) had an average monthly return of 0.63% in the sample period, which is relative close to the excess return of the S&P 500 in the same period of 0.68%. The HML portfolio had an average monthly return of 0.26%. The SMB had an average monthly return of 0.15%. Both SMB and HML yielded insignificant abnormal returns when regressed on CAPM in the sample period.

7.2 Methodology

The momentum strategies are strategies of buying past winners and short-selling past losers, which we in this paper denote as WML momentum portfolios.
We will follow the framework of Jegadeesh and Titman (1993), which means that sixteen WML momentum portfolios will be created, based on the returns in the past 1, 2, 3 and 4 quarters, and with holding periods of 1, 2, 3 and 4 quarters. The following subsection is a step-by-step description of the research design following the methodology of Jegadeesh and Titman (1993).

Firstly, the arithmetic returns were converted into log-returns to create the rolling formation period returns, since accumulating log-returns over a period can be done by summation (Munk, 2016):

$$r_{i;t}^{log} = ln(1 + r_{;ti}) \tag{17}$$

Secondly, we defined a formation period of J months. At the beginning of each month, the log-returns of the past J-month were summed together, creating rolling returns over the past J-periods:

$$r_{J}^{log} = \Sigma_{m=1}^{J} r_{i;t}^{log} \tag{18}$$

Thirdly, the monthly returns (equation 19) and the rolling returns of the formation-periods (equation 20) were converted back to arithmetic returns as the conversion into log-returns only was done to calculate the rolling returns of the formation periods:

$$r_{i;t}^{arithmetic} = e^{r_{i,t}^{log}} \quad 1 \tag{19}$$

$$r_{J;t}^{arithmetic} = e^{r_{J,t}^{log}} \quad 1 \tag{20}$$

In the rest of the paper arithmetic returns will be applied, as the results will be more comparable to the academic literature.

Fourthly, all available stocks were ranked each month, based on their cumulative return in the past J months. The stocks were then segmented into two categories, "past losers" and "past winners": Past losers were the 10% worst performing stocks in the past J months, and the past winners were the 10% best performing stocks in the past J months. Which again is a replication of the methodology of Jegadeesh and Titman (1993). Moreover, to be included in the holding period at time t, the stocks were required to have return data in the whole formation period.

Fifthly, an equally weighted portfolio of the "past winners" and an equally weighted portfolio consisting of the "past losers" were constructed each month, still following the framework of Jegadeesh and Titman (1993). It should be noted however, that we would typically obtain better diversification using different set of portfolio weights. In addition, everything else equal, an equally weighted portfolio consists of a larger fraction of small-cap stocks compared to a value weighted portfolio. This might influence an equally weighted portfolio to obtain higher returns than a value weighted portfolio would have, as small stocks are usually more volatile and will thus have higher expected returns (Munk, 2016). We will capture this effect by regressing on Fama-French's three-factor model however.

Lastly, the long-short WML momentum portfolios were created by buying the "past winners" portfolio, and short-selling the portfolio containing the "past losers" each month, with a holding period of K months. This means that we were using overlapping holding periods, and the WML momentum portfolios each month consisted of a series of K long and short positions. Hence, each portfolio cohort is assigned equal weights, 1/K. In example, it means that the WML momentum portfolios in time t, consist of the stocks selected in the current month, as well as the previous K-1 months. In the next month, *t+1*, a new long-short portfolio will be created and the stocks that were selected in time (*t+1* − K) are closed out. Still following the methodology of Jegadeesh and Titman (1993), we rebalanced every month to maintain equal weights.

7.3 Risk adjustment

A significant alpha when regressing a trading strategy on a model, can be a result of abnormal returns or due to an incorrect model, this is called the *joint hypothesis problem* as already discussed in section 3. To lower the risk of concluding based on an incorrect model, we will test the portfolios using several models. In the following subsection we will describe how to interpret the results reported in the next section. We focus on the 16 long-short trading strategies described above and a critical value of 5% (Cuthbertson & Nitzsche, 2004).

7.3.1 No risk-adjustment & Sharpe Ratio

Firstly, the returns of the WML momentum portfolios will be analyzed from a reward-to-variability, (μ_i vs. σ_i^2), perspective. The first results reported are the average monthly return of the WML momentum portfolios with the belonging t-statistics (Cuthbertson & Nitzsche, 2004):

$$SE_i = \frac{\sigma_i}{\sqrt{T_i}} \tag{21}$$

$$t_i = \frac{\hat{r}_i - 0}{SE(\hat{r}_i)} \tag{22}$$

(In other words, a two-sided t-test). The null hypothesis H_0, is that the returns are equal to zero, $H_0 : \hat{r}_i = 0$, and the alternative hypothesis is that the returns are significantly different from zero, $H_1 : \hat{r}_i \neq 0$. To be significantly different from zero, $\hat{r}_i \neq 0$ the t-statistics need to exceed the critical value. As we have 276 observations in the sample period, the critical value of a student's t-distribution is 1.97.

In addition, we will apply the popular Sharpe ratio, which is the ratio of the risk-premium measured as $E[r_i] - r_f$ divided with the standard deviation (Munk, 2016):

$$SR = \frac{E[r_i] - r_f}{\sigma_{ri}} \tag{23}$$

(For the long-short strategies we do not subtract the risk-free rate however). The ratio measures the reward per unit of risk, and one of the advantages of Sharpe ratio, is that it is unaffected of how much the trading strategy is leveraged, opposed to the alphas (Pedersen, 2015). On the other hand Munk (2016) claims that there are some issues with the underlying assumptions of Sharpe ratio: The first assumption is that we initially hold a riskless asset. The second underlying assumption is that we are considering mutually exclusive zero-investment strategies, which are financed by borrowing i.e. going short in the risk-free rate. The issue in this context is that our trading strategies are not mutually exclusive and that we are not going short in the risk-free rate. However, it is widely used to compare 'active' trading strategies, and the results will therefore be reported. For the r_f in equation (23) we use the average risk-free rate of 2016 (Munk, 2016).

7.3.2 CAPM

Originally, Jegadeesh and Titman (1993) claimed proof of the momentum effect on the basis of the Sharpe-Lintner CAPM, as they argued that the momentum trading strategies were yielding abnormal returns while at the same time having a negative post-ranking betas. They claimed that it implied that the abnormal profits were not due to higher expected returns of the stocks as result of taking more risk (Jegadeesh & Titman, 1993).
To test whether the WML momentum portfolios obtains abnormal returns in our sample period, we will perform the OLS-regression already introduced:

$$r_{A;t} = \alpha_A + \beta_A(r_{m;t} - r_{f;t}) + \epsilon_{A;t}$$

Where $r_{A;t}$ represents the returns of the WML momentum portfolios.
The momentum portfolios are zero-cost portfolios, which means that the long position in the past winner portfolio is financed by the short position in the past loser portfolio. It is important to note that the momentum portfolio is "free" to enter (excluding the transaction cost) but is not a riskless investment.
The null-hypothesis, is that the Jensen's alpha is 0, H_0: $\alpha_A = 0$ and the alternative hypothesis is that alpha is significantly different from zero, H_1: $\alpha_A \neq 0$. Under the null-hypothesis the ratio of α, to its standard error is distributed as a t-distribution (K. Chan, 1988):

$$t_i = \frac{(\alpha_A - 0)}{SE(\alpha_A)}$$

As for the basic returns, a t-statistic is significant if it exceeds the critical value of 1.97.

7.3.3 Fama-French three-factor Model

To help us overcome the joint hypothesis problem, we also regress the trading strategies on Fama-French's three-factor model already introduced (Cuthbertson & Nitzsche, 2004):

$$r_{A;t} = \alpha_A + \beta_A(r_{m;t} - r_{f;t}) + s_i \; SMB_t + h_i \; HML_t + \epsilon_{A;t}$$

As with the CAPM we use a two-sided t-test with the null-hypothesis $H_0 : \alpha_A = 0$ and similarly with the alternative hyptohesis $H1 : \alpha_A \neq 0$. As with the former risk-adjustment the significance level can be formulated with the two-sided t-test as:

$$t_i = \frac{(\alpha_A - 0)}{SE(\alpha_A)}$$

And again the t-statistic is significant if it exceeds the critical value of 1.97

8 Results - Momentum strategies

In the following section we will firstly present the results of the WML momentum portfolios and discuss the implications of them in order to find out if our results provides evidence that it is possible to generate abnormal profits based on time-series patterns. Next, we will discuss data snooping to address the critique from Fama and French (1996). Lastly, we will discuss the transaction costs and risks associated with following a momentum-based trading strategy.

8.1 Momentum results - No risk-adjustments

Table 1 reports the monthly returns of the 16 long-short WML momentum strategies, with the belonging t-statistics in the sample period of December 1994 to December 2016. J denotes the formation period, and K denotes the holding period.
The results reported in table 1 shows that all the momentum strategies yielded positive returns, in consistence with the findings of Jegadeesh and Titman (1993). It is interesting however, that none of the WML momentum returns had a t-statistic above the critical value of 1.97. The results in table 1 indicates that the WML momentum strategies are not very sensitive to changes in the specifications, but the best trading-strategy appears to be the strategy with a 6-month formation period and a holding period of 9 months (6/9).

J	K=	3	6	9	12
3		0.0039	0.0039	0.0038	0.0035
		(1.09)	(1.24)	(1.35)	(1.50)
6		0.0064	0.0060	0.0053	0.0036
		(1.48)	(1.53)	(1.58)	(1.23)
9		0.0069	0.0061	0.0043	0.0026
		(1.47)	(1.45)	(1.17)	(0.80)
12		0.0061	0.0038	0.0024	0.0011
		(1.3)	(0.91)	(0.64)	(0.33)

Table 1: The table reports the monthly returns for the 16 long-short "WML" momentum portfolios based on the past 3, 6, 9 or 12 months and held 3, 6, 9 or 12 months with the belonging t-statistics. The portfolios are equally weighted portfolios of the 10% best performing stocks over the formation period minus the 10% worst performing stocks in the same formation period. J denotes the formation period, and K represents the holding period

This is confirmed by the results reported in table 2, which is showing that the 6/9 trading strategy was also the one generating the highest Sharpe ratio, with an annualized Sharpe ratio of 0.35. This is not a very impressive Sharpe ratio compared to the S&P 500 index, which generated an annualized Sharpe ratio of 0.57 in the same period, hence from this perspective it is not a very attractive investment strategy. However, as Jegadeesh and Titman (1993, 2001) showed, the market beta of the momentum portfolios are usually negative, so it will be interesting to investigate whether the trading strategies yielded significant alphas when we regressed the trading strategies on CAPM.

J	K=	3	6	9	12
3		0.24	0.27	0.29	0.32
6		0.33	0.34	0.35	0.27
9		0.32	0.32	0.26	0.17
12		0.29	0.20	0.14	0.07

Table 2: The table reports the annualized Sharpe ratio of the 16 long-short WML momentum portfolios. Again J denotes the length of the formation period and K denotes the length of the holding period.

8.2 Momentum results - CAPM

Table 3 below, reports the results of regressing the 16 WML momentum portfolios on CAPM. The WML momentum strategy with a 6 month formation period and 3 month holding period (6/3) yielded the most significant alpha, therefore we will discuss on the basis of the 6/3 WML momentum strategy.

J	K=	3	6	9	12
3	$\alpha_{3;K}$	0.0078	0.0069	0.0062	0.0053
		(2.40)	(2.34)	(2.35)	(2.38)
	$\beta_{3;K}$	-0.5846	-0.4419	-0.3616	-0.2720
		(-7.73)	(-6.47)	(-5.88)	(-5.26)
6	$\alpha_{6;K}$	0.0106	0.0093	0.0078	0.0056
		(2.62)	(2.47)	(2.38)	(1.98)
	$\beta_{6;K}$	-0.6234	-0.4821	-0.3678	-0.2987
		(-6.62)	(-5.55)	(-4.84)	(-4.54)
9	$\alpha_{9;K}$	0.0112	0.0093	0.0069	0.0046
		(2.5)	(2.29)	(1.92)	(1.47)
	$\beta_{9;K}$	-0.6316	-0.4708	-0.3776	-0.3037
		(-6.09)	(-5.00)	(-4.53)	(-4.15)
12	$\alpha_{12;K}$	0.0010	0.0069	0.0049	0.0031
		(2.23)	(1.70)	(1.33)	(0.96)
	$\beta_{12;K}$	-0.5754	-0.4584	-0.3635	-0.2997
		(-5.55)	(-4.84)	(-4.30)	(-3.97)

Table 3: The table reports the 16 long-short Past Winner Minus Past Loser (WML) momentum portfolios' monthly Jensen's alpha and market beta with the belonging t-statistics. J represents the formation period, and K represents the holding period

Looking at table 3, the first thing that might jump to the eye is that the monthly alpha of all the WML momentum portfolios are positive and almost all of them are significant. This is a bit surprising keeping the results of the previous subsection in mind. The explanation is that the WML momentum portfolios had a very negative correlation with the market return. The market beta of the 16 (Past) Winner Minus (Past) Loser (WML) momentum portfolios varied from -0.27 to -0.63. These findings were first assumed to be an error in terms of methodology, as the betas of the WML momentum portfolios in the sample periods of Jegadeesh and Titman (1993, 2001) were negative in general, but close to zero. To test whether the results were an outcome of a mistake, we performed three tests:

Firstly, we repeated the methodology on another test-sample, namely the data sample used in Jegadeesh and Titman (1993), where we obtained the same results as Jegadeesh and Titman (1993), which are reported in appendix A. Secondly, we checked that we obtained the same results regressing the excess returns of the (past) winner and (past) loser portfolios on CAPM individually, and then subtract the loser from the winner. This yielded the same result in the end. Thirdly, a 'benchmark' was performed, where we regressed French (2017)'s momentum portfolio on CAPM in the same sample period. See section 9.2 for an explanation of how French (2017) constructed his WML momentum portfolio. French (2017)'s momentum portfolio had a negative beta of -0.32 (French, 2017). Hence, our findings indicate that the market beta of the WML momentum portfolios indeed are significantly negative in the sample period. This finding is also consistent with the findings of Daniel and Moskowitz (2015), who reported that the WML momentum portfolio, constructed almost the same way as French (2017)'s momentum portfolio, had an average market beta of -0.58 from 1927 to 2013 (Daniel & Moskowitz, 2015).

To find out why the momentum portfolios were yielding a negative beta, the momentum portfolios were decomposed in (past) winners and (past) losers - this is documented in appendix B. The findings were quite interesting: The market beta β_m of the winner portfolios were in general around 1, whereas the loser portfolios' market beta in general were in the space between 1.30 to 1.60, which implies that the loser portfolio is the predominant source of the negative market beta.

Daniel and Moskowitz (2015) reported results, which indicated that the momentum strategies experience significant declines after crises, and noted that the return of the loser portfolios had realized a gain of 163% in the three-month period of March-May 2009, in the aftermath of the financial crisis, where the winner portfolio only had a gain of 8% in the same period. In addition Daniel and Moskowitz (2015) found that the loser portfolio throughout history consistently obtains higher returns than the winner portfolio when the market rebounds. Daniel and Moskowitz (2015) claimed that these "momentum crashes", seemed to at least partially explained the high loser-beta (Daniel & Moskowitz, 2015).
Similar results could be obtained from our return data; the loser portfolio with 6 month formation and 3 month holding period (6/3) had a gain of 154.82% where the 6/3 winner portfolio only obtained a gain of 10.25% in the 3 month period of March-May 2009.
The momentum crash is also very easy to spot on figure 9, which shows the cumulative returns of

the 6/3 WML momentum portfolio in the sample period on a logarithmic scale. The WML momentum strategies generated negative returns in almost every month in 2009, and only started to consistently generate positive returns in February 2010. Looking at figure 9 again, it is clear that the predominant source of the crash comes from the loser portfolio. It can also be seen that the WML momentum portfolio was performing very well up until the financial crisis, but the momentum crash has erased much of the cumulative gain.

The crash following the dot-com bubble was also extreme, as the return of the 6/3 WML momentum portfolio in January 2001 alone was -40.84%. The difference between the momentum crash in 2001 and in 2009, besides the scale of the crash, is that the WML momentum strategies returned to performing well after the momentum crash in 2001, but the WML momentum strategies generated far smaller returns from 2010 to 2017, than they did before the financial crisis. There are different reasons for why this could be; one reason, could be that investors have become aware of the momentum effect, which might influence the profitability of the strategy as we will discuss further in section 8.5.

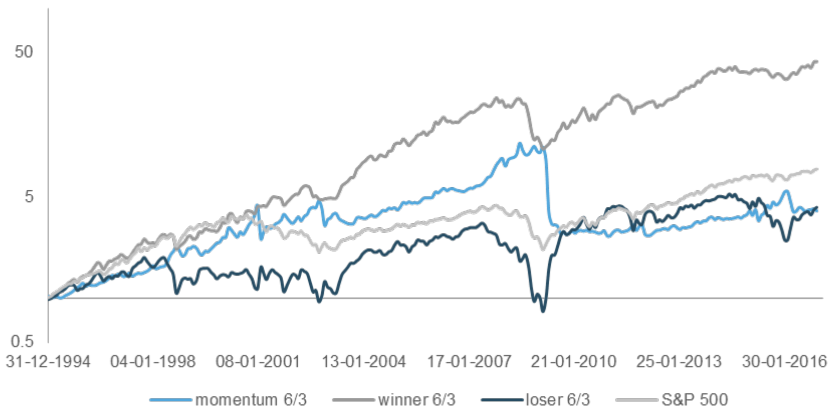

Figure 9: The figure displays the development of the WML momentum portfolio with a formation period of 6 months and a holding period of 3 months against S&P 500 in the sample period

The momentum portfolios' significant negative market betas could imply that the momentum portfolios typically have performed very well when the market has performed poor, which could be very attractive if the investor is preferring market neutral portfolios (Pedersen, 2015). Therefore we further investigated the market beta of the 6/3 WML momentum portfolio over time with a rolling window of 24 months, which can be seen in figure 10. The figure shows that the correlation between the momentum and the market is varying drastic over time. Moreover, it appears that the beta is positive in bull markets, but negative in bear markets, which is very attractive from a hedging perspective. However, it is also clear that the market betas remain negative in the post crisis period of time, which is consistent with the fact that momentum performs very badly when the market rebounds after a crisis (Daniel & Moskowitz, 2015).

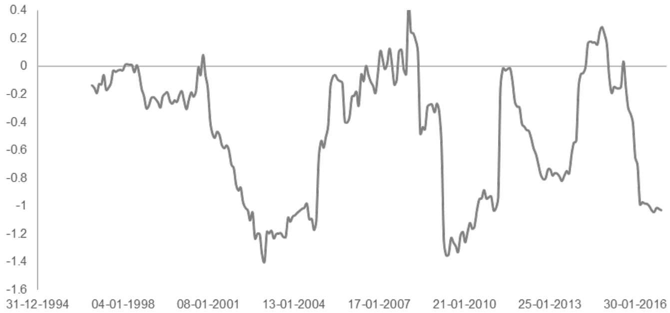

Figure 10: The figure displays the rolling market beta of the (6/3) momentum trading strategy with a 24 months rolling window

Daniel and Moskowitz (2015) also investigated whether the momentum portfolio could be hedged by creating a trading strategy where they longed both momentum and the market from 1927 to 2013. They concluded that this strategy would indeed hedge the underperformance of the market returns in the bear markets to some extend, because the market beta of momentum is negative in bear markets. However, they also found that the trading strategy underperforms on longer term compared to the market return (Daniel & Moskowitz, 2015).
The results reported in figure 11 and table 4 are the results of going long in the market portfolio and in the 6/3 WML momentum portfolio. The hedged portfolio performed better than the WML momentum portfolio and market portfolio did alone, which can be seen in figure 11. One could potentially jump to the conclusion that our results are contradicting the findings of Daniel and Moskowitz (2015), but as we have a relatively short sample period with two crises in it, our findings are not directly comparable. The hedged WML momentum portfolio yielded an average monthly return of 1.51%, and had a Sharpe ratio of 0.85. Moreover, the hedged momentum portfolios generated a significant monthly alpha of 1.26%, and only had a market beta of 0.38. However the hedged momentum portfolio still had a loss of 46.91% in April 2009, which indicates that the loss of the momentum "crash" was not hedged very well, which is also clear from figure 11.

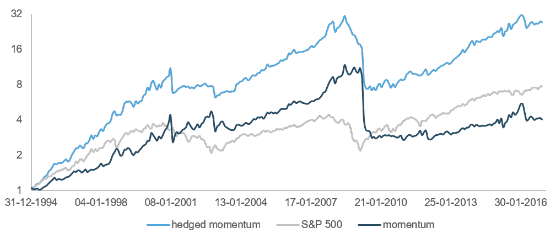

Figure 11: The figure displays the cumulative return of the hedged (6/3) momentum portfolios compared to the unhedged momentum portfolio and S&P 500

	WML hedged
r	0.0151
	(3.65)
SR	0.85
α	0.0126
	(3.08)
β	0.3769
	(3.98)

Table 4: This table presents the characteristics of the hedged (6/3) WML momentum's monthly return, Sharpe Ratio and CAPM results

These findings indicates that longing the WML momentum trading strategies together with a long position in the market could be an attractive trading strategy, however with some drawbacks. This will be discussed further in section 8.5 (Daniel & Moskowitz, 2015).

8.3 Momentum results - Fama-French three-factor model

The objective of regressing on Fama-French's three-factor model was to determine whether the significant alphas of the WML momentum portfolios could be explained by well-known factors. This subsection will first briefly discuss the characteristics of the Small-Minus-Big (SMB) and High-Minus-Low (HML) factors, followed by a discussion of the results of regressing the WML momentum portfolios on Fama-French's three-factor model, reported in table 5 and 6.

As mentioned in section 7.1.2 the HML and SMB factors did not yield significant alphas in the sample period. Especially the SMB factor seems to have performed bad in the sample period. This could potentially be explained by the fact that there has been two crises in the relatively short period of time, as small stocks usually have a higher beta and thus tend to perform worse when the market is stressed. This hypothesis was first confirmed by looking at the data, in which the SMB portfolio yielded negative returns around the dot-com bubble in 2000 with as extreme returns as three month in a row of -16.88%, -7.75% and -5.51% from March to May. However, the SMB portfolio actually performed well through-out the financial crisis in 2008. So a further investigation showed that the bad performance is simply due to overall low returns in the sample period.
An investigation of the HML portfolio revealed that the portfolio performed extremely well around the dot-com bubble, which is logical since that tech-companies typically would be growth companies with a low book-to-market value. The investigation also showed that HML has only generated an average monthly return of 0.06% since 2010. The fact that the HML portfolio has performed badly in recent years is not surprising considering that tech companies (which the US equity market has many of) have performed well, such as Netflix, Facebook, Amazon and Activision, moreover bio-tech companies have also performed well relative to the market the past decade, which are also most often characterized as growth companies. Nevertheless, the lack of abnormal returns might indicate that Fama & French would most likely not have found that they were anomalies if they had based their research on recent data.
The fact that both the SMB and HML have performed poorly in the sample period could be an expression of the fact that well known factors might stop working as trading strategies, because investors become aware of them and start to trade based on the factors, which will erase the abnormal profits that can be gained from the strategies. This will be discussed further in section 8.5.

Table 5 reports the monthly alpha of the 16 WML momentum portfolios when regressed on Fama-French's three factor model with the belonging t-statistics. As with the other tables, J represents the length of the formation period, and K represents the length of the holding period.

J	K=	3	6	9	12
3		0.0081	0.0074	0.0069	0.0062
		(2.45)	(2.49)	(2.61)	(2.81)
6		0.0114	0.0104	0.0091	0.0070
		(2.81)	(2.77)	(2.83)	(2.56)
9		0.0127	0.0110	0.0087	0.0065
		(2.87)	(2.77)	(2.52)	(2.17)
12		0.0120	0.0091	0.0071	0.0053
		(2.76)	(2.32)	(2.05)	(1.76)

Table 5: The table reports the 16 portfolios' monthly Fama French 3-factor alphas with belonging t-statistics in the sample period of December 1994 to December 2016. J represents the formation period, and K represents the holding period

The monthly alphas reported in table 5, indicate that the abnormal returns became slightly more significant when tested on the Fama & French three-factor model compared to CAPM, consistent with the findings of Jegadeesh and Titman (2001). It was now the 9/3 WML momentum portfolio that yielded the most significant alpha, which is due to a more negative factor loading on the HML & SMB factors than the 6/3 WML momentum portfolio, which can be seen in table 6.

Table 6 documents the 16 WML momentum trading strategies' market beta and their loading on the two additional factors, namely the High-Minus-Low (HML) and the Small-Minus-Big (SMB) factors.

J K=	3			6			9			12		
	Rm-rf	HML	SMB	Rm-rf	HML	SMB	Rm-rf	HML	SMB	Rm-rf	HML	SMB
3	-0.5656	-0.1662	0.0167	-0.4331	-0.2227	-0.0006	-0.3551	-0.2614	-0.0524	-0.2695	-0.2888	-0.0896
	(-7.45)	(-1.53)	(0.16)	(-6.37)	(-2.28)	(-0.01)	(-5.84)	(-3.00)	(-0.64)	(-5.35)	(-4.00)	(-1.32)
6	-0.6117	-0.3383	-0.0291	-0.4762	-0.389	-0.0887	-0.3662	-0.4282	-0.1404	-0.2994	-0.4458	-0.1680
	(-6.53)	(-2.52)	(-0.23)	(-5.56)	(-3.17)	(-0.77)	(-4.95)	(-4.03)	(-1.41)	(-4.75)	(-4.93)	(-1.9)
9	-0.6249	-0.5305	-0.1425	-0.4706	-0.5620	-0.1877	-0.3805	-0.5827	-0.2167	-0.3080	-0.5812	-0.2368
	(-6.15)	(-3.64)	(-1.04)	(-5.16)	(-4.30)	(-1.82)	(-4.78)	(-5.10)	(-2.02)	(-4.48)	(-5.89)	(-2.56)
12	-0.576	-0.6622	-0.2244	-0.4633	-0.6875	-0.2505	-0.3694	-0.6842	-0.2787	-0.3058	-0.6678	-0.2926
	(-5.76)	(-4.61)	(-1.67)	(-5.14)	(-5.32)	(-2.07)	(-4.66)	(-6.01)	(-2.61)	(-4.39)	(-6.68)	(-3.12)

Table 6: the table reports the result of regressing the 16 long-short WML momentum portfolio's on Fama-French's three factors. J represents the formation period, and K represents the holding period

Again, and not surprisingly, the WML momentum portfolios had significantly negative market betas. For a further discussion of the portfolios' market beta see section 8.2.

As stated above, the abnormal returns of the WML momentum portfolios became more significant when regressed on the three-factor model, which seems to be at least partially explained by the fact that the WML momentum trading strategies consistently had a negative loading on the HML factor - significantly negative in 15 out of the 16 trading strategies. This indicates that the WML momentum portfolios on average are long in growth stocks and short in value stocks. The fact that the correlation with the HML factor was negative is not very surprising as value strategies typically buy long term past losers and sell long term past winners (Fama & French, 1996; Bondt & Thaler, 1987). Moreover, as mentioned in section 2, Asness et al. (2013) found that the momentum strategy consistently had a negative correlation with HML throughout time. Asness et al. (2013) therefore tested the HML and momentum strategies together on several markets and found that the HML and momentum portfolios yielded better returns jointly compared to applying them as two separate strategies. As we will discuss in section 11.2, the HML factor could therefore be useful for improving the machine learning algorithms, but joint portfolios of HML and momentum will not be investigated in this paper (Asness et al., 2013).

The WML momentum portfolios' slope on the SMB factor was most frequently negative, consistent with Jegadeesh and Titman (2001)'s findings, and significantly negative for 6 of the strategies. This result was a bit surprising as it was expected that the correlation would be positive, since smaller stocks in general tend to perform better (Fama & French, 1996). Moreover, the portfolios were created with equal weights, which means that the WML momentum portfolios has a larger fraction of small cap stocks compared to applying value weighted portfolios, as explained in section 7.2. But as it turned out, the WML portfolios had a negative correlation with the SMB factor in general, which implies that the momentum strategies on average is long in larger stocks and short in smaller stocks in terms of market cap.
Looking at the winner and loser portfolios individually, reported in appendix B, both of them loaded significantly on the SMB factor however. The loser portfolios' slope on the SMB factor was around 0.6 to 0.7 where the the winner portfolios had a slope on SMB approximately between 0.4 to 0.5. The findings indicate that size in interaction with the momentum strategies could increase the prediction abilities of the machine learning algorithms, but will not be investigated further in this paper.

8.4 Data snooping

Fama and French (1996) stated that they were reluctant to add the momentum factor to their three-factor model even though the three factor model could not capture the momentum effect, and even stated that the reason that the momentum strategies yielded abnormal returns might be "a spurious result of data snooping" (Fama & French, 1996).

Data snooping is the term for trying different model specifications out, and apply the information obtained, to guide towards a certain result that is desired (Cuthbertson & Nitzsche, 2004). As Munk (2016) states it: "If you look long enough for a pattern, you will see one". Campbell et al. (2011) however argues that datasnooping biases are almost impossible to avoid since economics is non-experimental, and it therefore is not possible to run a new experiment to create new original data (Campbell et al., 2011). In the following we will discuss the sensitivity of the momentum-based strategies and the opportunity of data-snooping.

The best way to see if a trading strategy generating abnormal returns is due to data snooping is to apply it on a different data set e.g on a different market or in the same market but at a different time period. The trading strategies need to work out-of-sample after the publication of the trading strategies as well (Munk, 2016).

In section 8.2 we found that the abnormal returns of the momentum strategies were significant, which is consistent with previous findings in the literature (Jegadeesh & Titman, 1993, 2001; Daniel & Moskowitz, 2015; Asness et al., 2013). Hence, the results indicate that the abnormal returns found by Jegadeesh and Titman (1993, 2001) are not a matter of data snooping. Moreover, the momentum strategies' test-statistics were not very sensitive to changes in the length of formation and holding period as 11 of the 16 reported portfolios yielded significant Jensen's alpha, which could not be explained by adding the HML and SMB factor. Thus, the results documented in this paper indicates that the abnormal returns of the momentum strategies are most likely not a matter of data snooping.

8.5 Momentum as a trading strategy

The results reported in the previous subsections indicated that momentum-based strategies can generate significant abnormal returns. This subsection will discuss the feasibility of implementing a momentum based trading strategy from the perspective of a hedge fund. Firstly, we will discuss the associated transaction costs, followed by a discussion of the implied risks, and lastly we will discuss the hedged momentum strategy.

If a trading strategy appears to be profitable, the next thing a hedge-fund would do, would be to consider whether the strategy would survive transaction costs (Pedersen, 2015).
The momentum-based trading strategies reported in this paper require a lot of trading activity compared to a buy-and-hold strategy, as the strategies imply both longing and shorting between 200 and 300 stocks, which can be seen in figure 12. Hence, the associated trading costs would most likely be high, which means that the net profit of the momentum-based strategies would be lower. It is not the large portfolio sizes in itself that would be critical for a large hedge fund, it might even be an advantage that the portfolios are large, as it would imply that the market impact, everything else equal, would be smaller. However the amount of stocks, which would be needed to be replaced each month is important in terms of transaction cost. If we follow the trading strategies exactly as Jegadeesh and Titman (1993) suggests, we would have to adjust between 400 and 600 positions each months because the portfolios are rebalanced monthly to remain equally weighted portfolios. This would not be implemented in practice, as it would lead to a large turnover. Instead a hedge fund would most likely try to construct portfolios and rebalance them after what would maximize the performance after transactions costs. This could be an interesting topic for further research (Pedersen, 2015).
Another potential issue is related to the fact that the strategies might involve trading illiquid assets, since both the loser and winner loaded significantly on the SMB factor. This is a potential issue for two closely related reasons: Firstly, it means that it might be hard to sell the assets exactly when the trading strategy is requiring you to do it, and secondly the market impact costs will be larger when trading illiquid stocks - this is especially a problem for large investors, which are usually taking relatively large positions. Thus, it could be interesting to see whether the trading strategies would work on a data-sample excluding small-cap stocks (Pedersen, 2015).

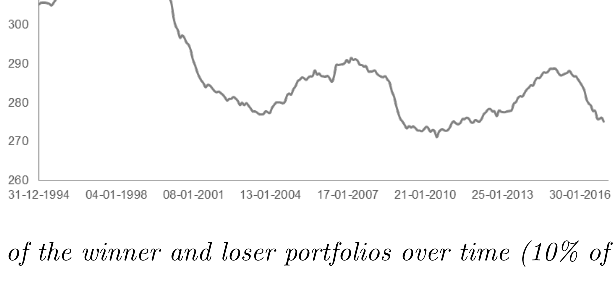

Figure 12: portfolio size of the winner and loser portfolios over time (10% of the stocks listed on NYSE)

As we saw in section 8.2, there can be large risks involved with following a momentum-based strategy blindly. If a hedge fund experienced a similar decline to the momentum crash in 2009, it would be devastating, as investors of hedge funds require that they make money in any environment (hence the "hedge"). Daniel and Moskowitz (2015) argued however, that the momentum crashes are cyclical and therefore predictable, which means that a hedge fund could stop following the strategy when a bubble has burst. This could be feasible in practice to some extend, since the momentum crashes happen when the markets are recovering, and according to Pedersen (2015), some hedge funds actually follow this strategy to some extend. It is important to note however, that as there only has been a limited number of crisis throughout the years, there is only limited data to support this (Pedersen, 2015; Daniel & Moskowitz, 2015).
Another risk of following a momentum-based trading strategy is related to the fact that momentum does not seem to have the same strong positive trend it had before the financial crisis. This could just be because the post-crisis period is a short period of time, which can potentially be regarded as an "unusual" time period, but there is the risk that the low returns of momentum is because the public has become aware of the momentum effect. When the public become aware of a systematic mispricing of assets, it is likely that rational investors will try to take advantage of it by longing (shorting) the undervalued (overvalued) assets. If enough investors do so, the mispricing disappears (Munk, 2016).
Lastly there is the risk of taking short positions. When shorting assets there is a risk of "buy-ins", which is when the lender wants the assets back early, which typically happens in times where it is not very attractive to close the short position (Pedersen, 2015).

In section 8.2 we argued that momentum strategies could be profitably hedged with a long position in the market, since the momentum-based trading strategies had a significantly negative market beta, which indicated that they generate positive returns in bear markets and negative returns in bull markets. The hedged 6/3 WML momentum portfolio seemed as an attractive investment strategy, as it obtained a high Sharpe-ratio of 0.85 and a significant monthly abnormal return of 1.26% together with a beta of only 0.38, which is documented in table 4 and figure 11. However, the trading costs and risks associated with momentum-based trading strategies might cause the hedged portfolio to be less profitable net of transaction costs. Moreover, investors would still have punished a hedge fund hard in the aftermath of the crisis, if it had followed the hedged momentum strategy, since the hedged portfolio still had a severe loss in 2009 as a consequence of the momentum crash.

Overall the results were in favor of the hypothesis that abnormal returns can be generated from time-series data. Now it is interesting to see whether the machine-learning algorithms are able to exploit time-series patterns to generate abnormal returns as well. The machine learning algorithms will only be feeded with a stock's returns of the past 12 months, and will be tested in the same sample period as the momentum strategies, which will make the trading strategies comparable.

9 Data & Methodology of Machine Learning Strategies

In the previous section we found that the momentum strategies could yield significant alphas when regressed on CAPM and Fama French's three-factor model. In continuation of these findings we expect the machine learning models to identify overperforming and underperforming stocks based only on time series patterns. This section will first justify and describe the data applied for the machine learning based trading strategies, followed by an explanation of how the momentum factor was constructed, which we use in Carhart's four-factor model. When we have described the data we will explain how we constructed the trading strategies. Lastly we will describe how we will evaluate the algorithms' performance.

9.1 Data

9.1.1 Justification of the inputs

There are many examples of literature that claim that stocks are not statistically independent. The most well-known theories proclaim that auto-correlation of stocks is negative on very short term (1 week to 1 month), and is positive on medium run (3 months to 12 months), and negative on long run (12 month to 36 months) (Munk, 2016; Jegadeesh & Titman, 1993, 2001; Bondt & Thaler, 1985, 1987; Campbell et al., 2011). In the previous section we presented results, which indicated that the momentum based strategies generate abnormal returns on medium run, which implies that stock returns have predictable time-series patterns to some extend. In this section we will try to exploit this predictability, by using machine learning algorithms. Therefore, the classification algorithms will receive a stock's past returns lagged 1 to 12 months as the features each month, which means that the algorithms will have the same underlying data to pick stocks from as the momentum-based trading strategies.

9.1.2 Training Data

The data in the training set was return data of common stocks listed on New York Stock Exchange (NYSE) from December 1925 to December 1992, and includes 4325 stocks. The data-collection procedure was the same as described in section 7.1, where we also argued that NYSE data was collected because it includes a lot of stocks and the data was of high quality, which is very important in this context, as it can have enormous effect on the performance of the machine learning models (Müller & Guido, 2017).

9.1.3 Test Data

The investment universe was exactly the same as we used for the momentum strategies to make all the trading strategies comparable. This means that the sample period was from December 1994 to December 2016 based on stocks listed on NYSE. See section 7.1 for a further explanation of the test set. The decision of this exact sample period was not completely arbitrary, as it means that the test set will make up approximately (25%) of the available time period on CRSP, which is the most common split between training- and test-sets, as discussed in section 4.

9.1.4 Construction of the momentum factor

French (2017)'s momentum-portfolio was applied as the momentum factor, which is constructed using almost the same methodology as Carhart (1997). The methodology deviates slightly from Carhart (1997) however, as French (2017) creates six value-weighted portfolios:

> The stocks were firstly divided into two categories each month, based on the median market value of equity, categorizing the stocks as either ”Big” or ”Small”.
>
> Simultaneously, three segments were created each month, based on the aggregated returns in the formation period (t-12 to t-2): The 30% best stocks were categorized as ”High”, the 30% worst stocks as ”Low”, and the 40% stocks in-between as ”Neutral”.
>
> In total 6 value weighted portfolios could then be constructed, as a stock could either be ”Big” or ”Small”, while simultaneously ”Low”, ”Neutral” or ”High” each month.
>
> The momentum portfolio was then constructed by estimating the average of the two ”High” portfolios minus the average of the two ”Low” portfolios each month (French, 2017):

$$\text{MOM} = 1/2(\text{Small High} + \text{Big High}) \quad 1/2(\text{Small Low} + \text{Big Low}) \tag{24}$$

9.2 Machine Learning - Methodology

9.2.1 Labelling

For a supervised machine learning algorithm to learn, the algorithm needs to receive the output data initially, as discussed in section 4. Hence, we labelled the stocks in the following way:

Firstly all available stocks were ranked each month in both ascending and descending order. First in descending order to create winners such that:

$$r_{t;1} < r_{t;2} < r_{t;3} \text{ would imply ranks of } c_{r_{t,1}} = 3, c_{r_{t,2}} = 2, c_{r_{t,3}} = 1 \text{ in time t} \tag{25}$$

Where $c_{r_{t,i}}$ represents the rank of a stock i in descending order in time t.
Afterwards the losers were created by ranking the stocks each month in ascending order:

$$r_{t;1} < r_{t;2} < r_{t;3} \text{ would imply ranks of } z_{r_{t,1}} = 1, z_{r_{t,2}} = 2, z_{r_{t,3}} = 3 \text{ in time t} \tag{26}$$

Where $z_{r_{t,i}}$ represents the rank of a stock i in ascending order in time t.

Each stock was then labelled:

$$y_{t;i} = \begin{cases} 1 \quad \text{if} \quad 1 \qquad c_{r_{i,t}} \qquad q_t \\ 2 \quad \text{if} \quad 1 \qquad z_{r_{i,t}} \qquad q_t \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{27}$$

Where q_t represents 25% of the N available stocks at time t . "Winners" were labelled $y_{t;i}$=1, "Losers" were labelled $y_{t;i}$=2 and "Neutral" were labelled $y_{t;i}$=0. Hence, we trained the machine learning algorithms to predict the 25% worst and best stocks. This process was done for both the training set and the test set. The reason for using 25% as the threshold was to some extend arbitrary. The reason for not setting it to 10% was because it would increase the risk of making false conclusions due to an imbalanced dataset: The two most common mistakes are called a false positive (type I error) and a false negative (type II error). An example of the former is if a stock is labelled by the algorithm as a "winner", but is actually a "neutral" stock. The latter is if the stock actually is a "winner" but is labelled as a "neutral" stock. We want to avoid both types of errors, but especially type I errors.
If the threshold was set to 10% of N, it would imply a high probability of type II errors, as an algorithm could predict with 80% accuracy on average only by randomly guessing on "Neutral". This error was lowered by using 25% of N instead, which is the same methodology as Huerta et al. (2011) (Huerta et al., 2011).

9.2.2 Setting up the data

After having labelled the stocks in both the training and test set, the data needed to be set-up to be applicable in a machine-learning context:

Firstly, all the returns were lagged 1 to 12 months to create the input data. See section 9.1 for a further discussion of this.
Secondly, the data was reshaped. The most intuitive way of understanding the data, is to think of it as a table. Each row is a data point, which has a label (the output, denoted as "Y"), while the columns to the right of the label are the features (the input).
An illustrative snapshot of the data representation could look like the following:

Y	R_{L1}	R_{L2}	...	R_{L12}
0	0.007	0.006	...	0.003
1	0.021	0.018	...	0.012
0	0.047	0.086	...	0.001
2	0.042	0.001	...	0.003

This was done for both the training set and the test set. However, the machine learning model was 'told' which is which, so it only received the output Y in the training set. For a stock to be included in time t, we required the stock to have return data in the previous 12 months. This was required both in the training and the test set.
After the data had been set-up, the machine learning models were ready to be tested.

9.2.3 Construction of portfolios

When a machine learning algorithm had made its predictions of the winners and losers in the sample period, long-short trading strategies were created, buying the machine learning algorithm's predicted winners and short-selling the predicted losers each month. For the machine-learning based trading strategies we used holding periods of 1, 3, 6 and 9 month, which means that in total 4 long-short trading strategies were created for each algorithm. The trading strategies with 12 month holding periods were omitted, because the algorithms were only trained to predict the monthly winners. As with the momentum-based portfolios we used overlapping holding periods. Moreover, we rebalanced the portfolios monthly to maintain equally weighted portfolios.

9.2.4 Model evaluation

Each machine learning model was evaluated in two overall categories: How accurately it predicted, and more importantly how its trading strategies performed using the same risk-adjustments as discussed in section 7.2, except that Fama-French's three factor model was replaced with Carhart's four factor model.

9.2.4.1 Accuracy

In section 10 we will refer to "accuracy" or "prediction accuracy" quite often. This is the most widely applied measurement within the machine-learning universe to find out if an algorithm is performing well. Accuracy is just the term for the relative frequency of the correct predictions in the test set. In this context accuracy is not a good measurement, and will only be used as an indicative tool, as it does not unveil how often we predict loser stocks to be winner stocks and vice versa. To get a more granular view of the prediction accuracy, we also report a 3x3 confusion matrix, which can give us a better understanding of this. The confusion matrix does still not tell us whether the trading strategies are yielding abnormal returns however, which is the main focus of the paper. Hence, we will focus more on the performance of the trading strategies (Hastie et al., 2017).

9.2.4.2 Risk-adjustment of the trading strategies

We will only explain how to interpret the results of Carhart's four factor model, as we have already explained the other models in section 7.3.

To test whether the abnormal returns were explained by well-known factors including the momentum factor, we regressed the trading strategies on the following (Carhart, 1997):

$$r_{A;t} = \alpha_A + \beta_{A;m} \quad (r_{mt} \quad r_{ft}) + s_i \quad SMB_t + h_i \quad HML_t + o \quad MOM + \epsilon_{A;t}$$

The null-hypothesis of the model is that alpha is zero, $\alpha_A = 0$. If $\alpha_A > 0$ on a significant level, the trading strategy is earning abnormal returns. The critical value of the t-statistic is still 1.97.

10 Results - Machine Learning strategies

In the previous section we described the methodology to create the machine-learning based trading strategies. This section will report the results and evaluate each machine learning algorithm's ability to predict under- and over-performing stock. In the end of the section we will discuss the feasibility of implementing the algorithm-based trading strategies in a real-world setting.

10.1 K-Nearest Neighbor

The first machine learning algorithm tested was the k-Nearest Neighbor algorithm (k-NN). Before we could create the trading strategies based on k-NN, we needed to decide how many "Neighbors" were optimal to apply. Looking at figure 13 below, it is clear to see that the prediction accuracy of k-NN increased when we applied more neighbors in the model specifications. But as noted earlier, accuracy is not a very good measurement of how well the model is stock-picking. The fact that the prediction accuracy is below 50% means that we cannot know whether the model is actually predicting the "losers" and "winners" better when adding more neighbors, which is what we are interested in, or whether it is just classifying more and more of the stocks to be "neutral stocks". The latter would imply that at a certain level the accuracy would be 50%, but the model would just predict that we should never invest as all stocks will be classified as "neutral". Hence, choosing the number of neighbors in the model could not only be based on the highest possible prediction accuracy. Instead it was chosen more or less arbitrarily based on the guidelines from the literature (Müller & Guido, 2017).

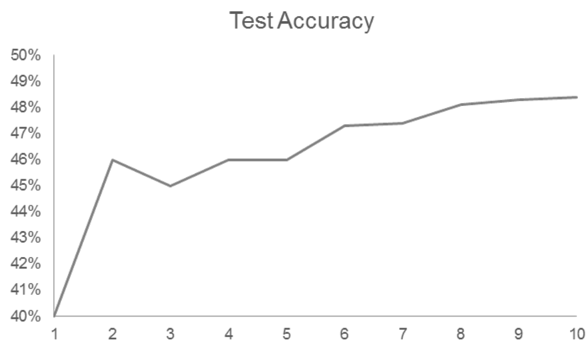

Figure 13: The figure displays the prediction accuracy of the k-Nearest-Neighbor on the y-axis and number of neighbors on the x-axis.

According to Müller and Guido (2017) the best model specification of k-NN in general is between 1 and 10 neighbors. It is clear to see from figure 13 that the incremental prediction power for each neighbors added is declining. Because the incremental prediction power when adding more than 6 neighbors were below 1%, we chose to apply 6 neighbours in the finale model to create the trading strategies from. Further tests could be conducted to find the best test-specification, which could be interesting for further research. See a further discussion of this in section 1.4.

The prediction accuracy of the 6-NN was only 47.3%. To get a more granular understanding of the accuracy, we created an extended 3x3 confusion matrix:

	Predicted Winner	Predicted Neutral	Predicted Loser
Winner	3.23%	18.27%	3.28%
Neutral	5.15%	40.69%	4.72%
Loser	3.29%	17.98%	3.38%

Table 7: The table reports k-NN's 3x3 confusion matrix, which is the relative frequency of each outcome

From table 7 it is clear that the winner portfolio consists of many loser stocks, and vice versa. Moreover, both portfolios consist of a large fraction of "neutral" stocks. This is the type I error, that we wanted to avoid. Hence, the accuracy results did not lead to high expectations of abnormal returns.

The discussion of k-NN's trading strategies will be based on the Winner-Minus-Loser (WML) portfolio with a three-month holding period, firstly because it is the trading strategy yielding the most interesting results, and secondly because the results of the four holding periods are very similar.

Table 8 reports the average monthly returns of the trading strategies created from k-NN's predictions with the belonging t-statistics.

Holding period=	1	3	6	9
winner	0.0117	0.0118	0.0115	0.0114
	(3.49)	(3.59)	(3.55)	(3.50)
loser	0.0110	0.0109	0.0110	0.0109
	(3.02)	(3.01)	(3.09)	(3.10)
WML	0.0007	0.0009	0.0004	0.0005
	(0.96)	(1.64)	(0.97)	(1.45)

Table 8: The table reports the monthly returns of the portfolios created by 6-NN's stock predictions with the belonging t-statistics. The sample period is December 1994 to December 2016. WML is the long-short strategy and stands for Winner-Minus-Loser

The results show that the long-short WML portfolio with a three-month holding period obtained the highest average monthly returns of (only) 0.09%. The results reported in table 8 indicate that the 3-month winner portfolio was generating very profitable returns, with an average monthly return of 1.18%. Hence, the source of the low performance of the WML portfolio was the high returns of the loser portfolio, which generated a monthly return of 1.09%.

Holding period=	1	3	6	9
WML Sharpe ratio	0.20	0.35	0.21	0.31

Table 9: The table reports the annualized Sharpe Ratios of the 4 Winner-Minus-Loser (WML) trading strategies created from the 6-NN algorithm in the sample period of December 1994 to December 2016

Interestingly and a bit surprisingly, table 9 shows that the 3-month WML portfolio had a Sharpe-ratio of 0.35, which is as high as the best momentum WML portfolio's Sharpe ratio. This indicates that the portfolio had a very low standard deviation, considering the fact that the monthly returns were very low. It is important to note that it is not possible that the winner and loser portfolios consist of the same stocks at the same time, but looking at figure 14 it seems as if the portfolios have the same characteristics. To investigate this further we regressed the winner portfolio on the loser portfolio. This resulted in a r^2 of 0.985, which indicates that the portfolios are highly correlated.

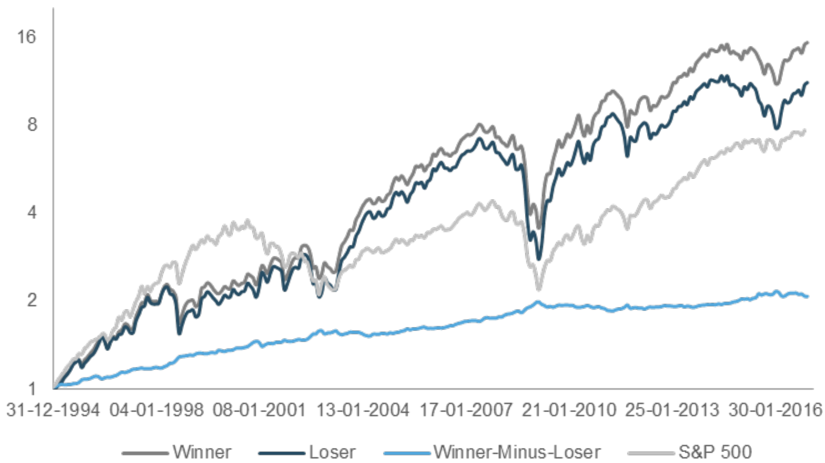

Figure 14: The figure shows 6-Nearest Neighbors's 3-month trading strategy's cumulative returns in the test period. Note that it is log-scale and the risk-free rate is added to the Winner-Minus-Loser portfolio

Looking at the initial results presented above one might jump to the conclusion that it would be optimal to just create a long-only trading strategy, or short the market return. Shorting the market return is usually easier and cheaper than shorting a loser portfolio of 300 stocks. Shorting the market is also a way to avoid the risk of "buy-ins" as discussed in section 8.2. However, when we regressed the portfolios using CAPM reported in table 10 below, it becomes clear that the winner portfolio had a high covariance with the market, which implied that the alpha of the winner portfolio became insignificant when regressed on CAPM.

Table 10 reports the results of regressing the portfolios on CAPM, with the belonging t-statistics.

Holding Period		Winner	Loser	WML
1	$\alpha_{1;p}$	0.0024	0.0013	0.0012
		(1.33)	(0.56)	(1.60)
	$\beta_{1;p}$	1.0802	1.1479	-0.0677
		(25.67)	(24.09)	(-4.00)
3	$\alpha_{3;p}$	0.0027	0.0011	0.0015
		(1.50)	(0.56)	(3.16)
	$\beta_{3;p}$	1.0615	1.1573	-0.0958
		(25.73)	(24.84)	(-8.49)
6	$\alpha_{6;p}$	0.0023	0.0014	0.0009
		(1.32)	(0.69)	(2.42)
	$\beta_{6;p}$	1.0582	1.1355	-0.0773
		(25.96)	(24.7)	(-8.53)
6	$\alpha_{9;p}$	0.0023	0.0014	0.0010
		(1.32)	(0.61)	(2.92)
	$\beta_{9;p}$	1.0540	1.1173	-0.0634
		(25.99)	(24.73)	(-8.37)

Table 10: The table reports Jensen's alpha and CAPM beta of the portfolios based on 6-Nearest Neighbors' stock predictions with the belonging t-statistics in the sample period of December 1994 to December 2016. WML (Winners-Minus-Loser) is the long-short portfolio.

Interestingly, the abnormal returns of the WML portfolios were significant for the holding periods of 3 to 9 months. Especially the trading strategy with a holding period of 3 months, which generated a monthly alpha of 0.15% with a belonging t-statistic of 3.16, which is a more significant Jensen's alpha than the most significant momentum portfolio's Jensen's alpha. The reason for this is firstly that the returns of the WML portfolio were positive almost independently of the state of the market, which is documented in figure 14, and secondly because the loser portfolio was yielding lower returns than the winner portfolio but had a higher market beta, resulting in a negative market beta for the WML portfolio. The fact that the WML portfolio had positive returns almost independently of the market conditions, makes it appear to be a low-risk investment. If we suppose that the risk really is low, an investor can apply leverage to the investment to make the portfolio attractive, assuming that the investor can do it at no cost. This will be discussed further in section 10.4.

Table 11 below reports the results of regressing the portfolios on Carhart's four factor model, with the belonging t-statistics.

Holding Period		$\alpha_{K;p}$	Market	HML	SMB	MOM
	winner	0.0023	0.9756	0.4234	0.4469	-0.2134
		(2.2)	(38.18)	(11.93)	(13.73)	(-9.91)
1	loser	0.0014	1.0200	0.4376	0.5164	-0.2584
		(1.12)	(34.49)	(10.65)	(13.71)	(-10.37)
	WML	0.0010	-0.0443	-0.0142	-0.0695	0.0450
		(1.33)	(-2.56)	(-0.59)	(-3.15)	(3.08)
	winner	0.0026	0.9584	0.4220	0.4381	-0.2130
		(2.53)	(38.97)	(12.35)	(13.98)	(-10.28)
3	loser	0.0012	1.0304	0.4523	0.5225	-0.2594
		(1.09)	(38.84)	(12.27)	(15.46)	(-11.6)
	WML	0.0014	-0.0720	-0.0303	-0.0844	0.0464
		(3.16)	(-6.85)	(-2.07)	(-9.01)	(5.23)
	winner	0.0022	0.9567	0.4260	0.4385	-0.2087
		(2.23)	(40.22)	(12.89)	(14.47)	(-10.41)
6	loser	0.0015	1.0063	0.4376	0.5158	-0.2651
		(1.42)	(38.7)	(12.12)	(15.57)	(-12.1)
	WML	0.0007	-0.0495	-0.0116	-0.0773	0.0565
		(2.06)	(-6.41)	(-1.08)	(-7.86)	(8.67)
	winner	0.0022	0.9526	0.4269	0.4370	-0.2086
		(2.24)	(40.51)	(13.07)	(14.59)	(-10.52)
9	loser	0.0014	0.9937	0.4390	0.5106	-0.2536
		(1.36)	(39.16)	(12.45)	(15.79)	(-11.86)
	WML	0.0008	-0.0411	-0.0121	-0.0736	0.0450
		(2.9)	(-6.57)	(-1.39)	(-9.24)	(8.54)

Table 11: The table reports the results of regressing the '6-Nearest Neighbors' trading strategies on Carhart's four factor model, with belonging t-statistics. The sample period is December 1994 to December 2016. WML stands for 'Winner-Minus-Loser' and are the long-short portfolios.

From the results of regressing the portfolios on Carhart (1997)'s four factors reported in table 11, it is clear that both the winner and loser portfolios were positively correlated with both the HML and SMB factors. The trading strategy yielding the most significant abnormal return was still the WML portfolio with a holding period of three months. The portfolio's loser and winner portfolio loaded significantly on both the HML- and SMB factors. The loser portfolio seems to be loading slightly more on both factors however, which means that the WML portfolio had a negative factor loading on both HML and SMB.

More surprisingly, both the winner and loser portfolio were negative correlated with the momentum factor. Considering the fact that the algorithm only received historical return data as the input, the expectation was that the algorithm would pick winner stocks that would correlate positively with the momentum portfolio. However, having the loading on the other three factors reported in table 11 in mind, the negative correlation makes sense: Both the winner and loser portfolio had a high market beta, whereas the momentum strategies had a significant negative market beta. Moreover, both the winner and loser had a positive correlation with the HML factor, which momentum had a negative correlation with. The winner portfolio had a slightly less negative loading on the momentum factor, which influenced the WML to have positive factor loading on momentum. The fact that all of the WML portfolios still generated a significant alpha except the 1-month holding period, implies that the well-known factors cannot explain the abnormal returns of the WML portfolios created by the k-NN algorithm.

In conclusion the WML trading strategy with a three-month holding period, was clearly the best trading strategy of the four strategies. It was the one yielding the highest Sharpe-ratio, and most significant abnormal returns when regressed on both CAPM and Carhart (1997)'s four factor model. Hence, the results indicates that it is possible to generate a positive significant abnormal returns from machine learning algorithms, so now it will be interesting to see whether the more complicated algorithms will perform better. The usage of the algorithm in a real investment framework will be discussed in section 10.4.

10.2 Naive Bayes Classifier

Naive Bayes Classifier was the best performing algorithm in terms of accuracy, with an accuracy of 51.6%, which is still not very impressive. However, as argued in section 9.2.4 the accuracy alone is not a very good measurement of an algorithm's ability to predict winner and losers. We therefore created a 3x3 confusion matrix to further investigate Naive Bayes Classifier's predictions:

	Predicted Winner	Predicted Neutral	Predicted Loser
Winner	0.59%	18.80%	5.39%
Neutral	0.63%	45.12%	4.83%
Loser	0.57%	18.21%	5.88%

Table 12: The table reports Naive Bayes 3x3 confusion matrix. The matrix indicates the relative frequency of the outcomes

From the confusion matrix it is clear to see that the Naive Bayes Classifier is having a large fraction of loser stocks in the winner portfolios and vice versa. This implies that the algorithm has predicted many false positives, which is related to the type I error. The confusion matrix does not tell whether trading strategies from the model obtain abnormal returns, but it is indicating that the Naive Bayes Classifier has not been able to distinguish successfully between the classes.

Table 13 displays the monthly returns of the trading strategies based on the Naive Bayes Classifier's predictions, together with the t-statistics.

Holding period=	1	3	6	9
winner	0.0138	0.0089	0.0087	0.0088
	(3.35)	(2.29)	(2.28)	(2.33)
loser	0.0115	0.0116	0.0116	0.0116
	(2.36)	(2.41)	(2.46)	(2.52)
WML	0.0023	-0.0027	-0.0029	-0.0028
	(0.94)	(-1.31)	(-1.60)	(-1.70)

Table 13: The table report Naive Bayes Classifier's portfolios' monthly returns with the belonging t-statistics in the sample period of December 1994 to December 2016

The results indicate that the winner portfolio generated high returns, but unfortunately so did the loser portfolio, which was the same problem as with the k-NN algorithm, as discussed in the previous subsection. This is confirmed by looking at figure 15, which shows the cumulative returns of the 1-month WML trading strategy: The winner portfolio was performing very well on average, but so was the loser portfolio, which means that the WML portfolio was only generating modest returns. The only WML trading strategy with a positive return was the 1 month holding period WML portfolio, which generated a monthly return of 0.23%, hence this is the only trading strategy that will be discussed.

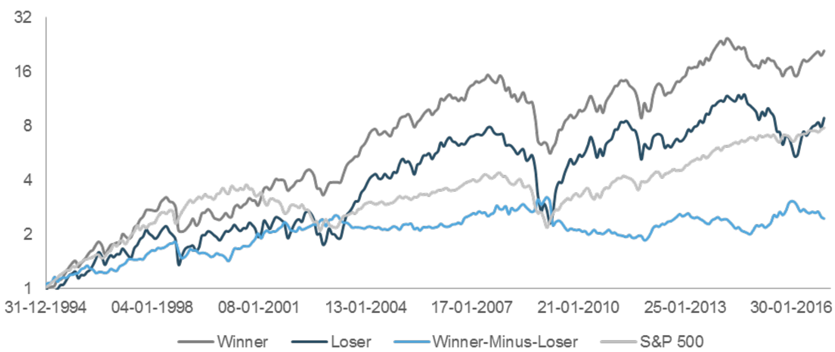

Figure 15: The figure displays the cumulative return of Naive Bayes Classifier's 1 month trading strategy against S&P 500 on a logarithmic scale. The WML is added with the risk-free rate

Table 14 reports the Sharpe-ratio of the 4 long-short trading strategies.

Holding period=	1	3	6	9
WML Sharpe ratio	0.20	-0.28	-0.33	-0.36

Table 14: The table report Naive Bayes Classifier's annualized Sharpe Ratio of the Winners-Minus-Loser portfolios in the sample period of December 1994 to December 2016

The WML trading strategy with a holding period of 1 month had a Sharpe ratio of 0.20, which is worse than the Sharpe ratio of the k-NN WML portfolio with a three-month holding period and only a little higher than the 9/12 WML momentum trading strategy, which was the fourth worst WML momentum trading strategy out of the 16 WML momentum trading strategies. From this perspective, the trading strategy created by the Naive Bayes Classifier does not appear very attractive. However, as it has been seen before, the picture might change when we risk-adjust with CAPM.

Table 15 reports the results of regressing the trading strategies on CAPM with the belong t-statistics.

Holding Period		Winner	Loser	WML
1	$\alpha_{1;p}$	0.0039	-0.0003	0.0042
		(1.41)	(-0.11)	(1.8)
	$\beta_{1;p}$	1.1830	1.4673	-0.2843
		(18.53)	(20.6)	(-5.25)
3	$\alpha_{3;p}$	-0.0010	-0.0001	-0.0009
		(-0.44)	(-0.04)	(-0.46)
	$\beta_{3;p}$	1.1852	1.4465	-0.2613
		(21.44)	(20.59)	(-5.75)
6	$\alpha_{6;p}$	-0.0012	0.0001	-0.0012
		(-0.5)	(0.03)	(-0.71)
	$\beta_{6;p}$	1.1634	1.4133	-0.2499
		(21.73)	(20.59)	(-6.17)
6	$\alpha_{9;p}$	-0.0010	0.0003	-0.0013
		(-0.46)	(0.1)	(-0.84)
	$\beta_{9;p}$	1.1661	1.3850	-0.2189
		(22.29)	(20.68)	(-6.00)

Table 15: The table report the Jensen's alpha and the CAPM beta of the trading strategies based on Naive Bayes Classifier's stock predictions with the belonging t-statistics. WML stands for Winners Minus Loser, and is the long-short portfolios.

The loser portfolio with a holding period of 1 month had a beta of 1.47 and the winner portfolio only had a beta of 1.18, which implied that the market beta of the WML portfolio had a significant negative beta of -0.28. This was not enough to yield significant abnormal returns, as the monthly Jensen's alpha of the 1-month WML only was 0.42% with a t-statistic of 1.8, which was not very surprising considering the prior findings.

Table 16 reports the results of regressing the trading strategies on Carhart (1997)'s four factor model.

Holding Period		$\alpha_{K;p}$	Market	HML	SMB	MOM
	winner	0.0024	1.1484	0.4607	0.4456	-0.0234
		(0.99)	(19.63)	(5.67)	(5.98)	(-0.47)
1	loser	-0.0000	1.2765	0.5321	0.8422	-0.3767
		(-0.00)	(28.35)	(8.51)	(14.68)	(-9.93)
	WML	0.0024	-0.1282	-0.0714	-0.3966	0.3533
		(1.23)	(-2.7)	(-1.08)	(-6.57)	(8.84)
	winner	-0.0025	1.1419	0.4649	0.5045	-0.0353
		(-1.32)	(24.98)	(7.32)	(8.66)	(-0.92)
3	loser	0.0003	1.2543	0.5317	0.8266	-0.3840
		(0.14)	(28.55)	(8.71)	(14.77)	(-10.37)
	WML	-0.0028	-0.1124	-0.0668	-0.3221	0.3487
		(-1.78)	(-3.00)	(-1.28)	(-6.75)	(11.04)
	winner	-0.0022	1.094	0.4398	0.5380	-0.0927
		(-1.24)	(25.8)	(7.47)	(9.97)	(-2.59)
6	loser	0.0005	1.2258	0.5267	0.8107	-0.3768
		(0.26)	(28.85)	(8.92)	(14.98)	(-10.52)
	WML	-0.0026	-0.1323	-0.0869	-0.2727	0.2841
		(-1.83)	(-3.82)	(-1.81)	(-6.18)	(9.73)
	winner	-0.0016	1.0741	0.3937	0.5430	-0.1459
		(-0.94)	(26.81)	(7.07)	(10.64)	(-4.32)
9	loser	0.0007	1.2010	0.5121	0.7929	-0.3689
		(0.38)	(29.09)	(8.93)	(15.08)	(-10.6)
	WML	-0.0022	-0.1269	-0.1184	-0.2499	0.2229
		(-1.66)	(-3.94)	(-2.65)	(-6.09)	(8.21)

Table 16: The table reports the results of regressing Naive Bayes Classifier's portfolios on Carhart's four factor model in the sample period of December 1994 to December 2016. WML are the long-short Winners-Minus-Losers trading strategies.

The results reported in table 16, shows that the abnormal returns of the 1-month WML portfolio was still insignificant when regressed on Carhart's four factor model. It is noteworthy, even though it is not reported in the paper that the alpha actually was significant when applying Fama-French's 3-factors, but adding the momentum factor made the alpha insignificant again.

From table 16 we can also see that both the winner and loser portfolios had a positive factor loading on the HML and SMB factor, but the winner had a lower factor loading than the loser portfolio, which means that the WML portfolio had a negative correlation with both HML and SMB. The alpha of the 1-month WML portfolio became less significant when regressed on the Carhart's four factor model however, as the portfolio had a significant positive loading on the momentum factor. The fact that the trading strategy had a positive correlation with momentum was not very surprising in itself, as the expectation was that the model should learn the time-series patterns that we found in section 8 to some extend. It was surprising however, to see that the WML portfolio only had a positive slope on the momentum factor because the loser portfolio had a more negative slope than the winner, and not because the winner had a positive momentum slope. The winner's negative factor loading was not significant however.

Looking at figure 16 it seems as if the Naive Bayes Classifier assigned a higher probability of being a loser stock if the past return has had a high volatility, as the portfolio size of the loser increased rapidly around the two large crisis. This might also explain why the loser portfolio had such a positive correlation with the SMB factor, as smaller stocks tend to be more volatile in general. Moreover, the loser portfolio was significantly larger through-out the whole sample period. This could indicate that Naive Bayes Classifier predicted that a stock was likely to be a loser stock if the returns had been negative in the previous 12 month, but that positive returns in the previous months not necessarily made a stock likely to be a winner. This is consistent with the results of table 16, where we saw that the loser portfolio was loading significantly negative on the momentum factor.

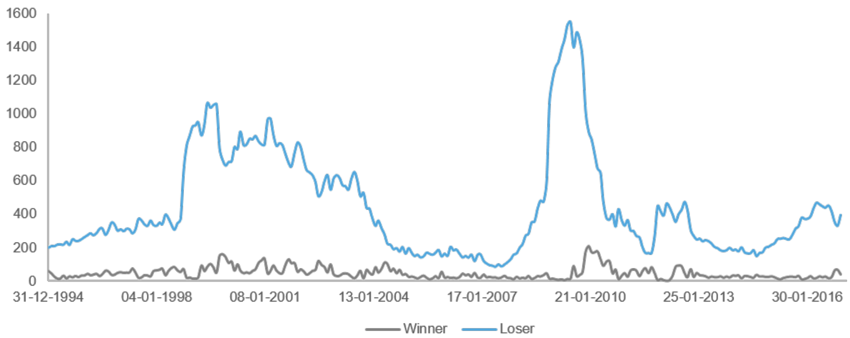

Figure 16: Number of stocks included in Naive Bayes Classifier's winner and loser portfolio in the test period

Overall the trading strategies created by the Naive Bayes Classifier were not very attractive, as none of them turned out to yield significant returns. Looking at figure 15 again, the problem did not seem to be due to any particular event, it just seems as if the loser portfolio is overall performing too well. As it was already disclosed, the decision trees did not yield any significance either, but luckily the Random Forest algorithm created some very interesting results which we will go through now.

10.3 Random Forest

Random Forest predicted the classes of the stocks with an accuracy of 51.2%, slightly worse than Naive Bayes Classifier. It is important to emphasize again however, that we are interested in the accuracy in terms of picking the correct winners and the correct losers only, meaning the accuracy by itself is only indicative. Therefore we created the below 3x3 confusion matrix, which shows the relative frequency of the 9 possible outcomes.

	Predicted Winner	Predicted Neutral	Predicted Loser
Winner	1.08%	2.15%	2.20%
Neutral	0.90%	47.59%	2.08%
Loser	1.09%	21.06%	2.50%

Table 17: The table reports the 3x3 confusion matrix of Random Forest's stock predictions. It shows the relative frequency of each possible outcome.

The matrix indicates that the Random Forest model is not predicting the winner and loser stocks very well. The algorithm seems to have predicted a large fraction of losers to be winners and vice versa. It is important to note however, that the confusion matrix does not take two very important factors into account when trading stocks:
Firstly, the matrix does not say anything about the timing of picking the winners and losers. When creating long-short strategies, it is not critical that the algorithm sometimes predicts that a loser stock is as winner, as long as the loser portfolio consists of a larger fraction of losers in the same month and vice versa.
Secondly, the confusion matrix does not tell us how much of a "loser" the predicted losers are, and how much of a "winner" the predicted winners are.

Hence, the most important results will still be the performance of the portfolios when regressed on the market models. The results of these tests are displayed in table 20-23, which will be discussed after the discussion of the applied specifications below.

10.3.1 Applied Parameters

n_estimators was set to 200, which means that 200 bootstrapped decision trees were applied. The general opinion about the number of decision trees is that the models only improve when adding more trees. The reason for 200 decision trees was based on diminishing returns of adding more trees as we see in figure 17, combined with the fact that computation time was increasing exponentially. In comparison Imandoust and Bolandraftar (2014) used 100 decision trees (Imandoust & Bolandraftar, 2014; Müller & Guido, 2017).

max features was set to the auto specification, which is the square-root of the m number of feature $\sqrt{m_{features}}$. This implies that the model was randomly given 3 features per node, which implies that the decision trees will be highly randomized as we will get further into below.

Maximum depth was not applied, which means that the model is highly overfitting, since it will not stop the decision tree until all the leafs are pure. Imandoust and Bolandraftar (2014) did not prune their trees either. With 200 decision trees the accuracy on the training set was 100%. Below we will get into the effect of setting a maximum depth of the decision trees (Imandoust & Bolandraftar, 2014).

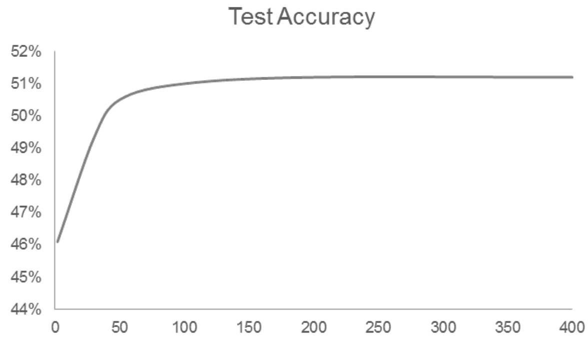

Figure 17: The figure shows the relation between number of decision trees (x-axis) used in the Random Forest algorithm and the accuracy of the stock prediction (y-axis). The figure clearly shows that the incremental accuracy is declining.

In section 4.2.4 it was described that the *maximum feature* specification can be important to adjust to increase the Random Forest model's prediction accuracy. The lower the maximum features per note, the more random the decision trees will be, but on the hand it also increases the complexity. As an example, if it only has 1 feature to choose from, it implies that the model cannot choose which feature has the highest explanatory power, but can only choose the best possible threshold for the randomly selected feature. Hence, a low number of features can imply very deep decision trees to fit the data. On the other hand, if we set the maximum feature to 12, the decision trees will always have all the features to choose from, and thus no randomness will be injected in the feature selection, which means that the only randomness in the model will be the bootstrapping of the return data. It is not given that a lower number of features is better than a higher number of features or vice versa.
To see if the model could be improved compared to our initial results reported, a manual search changing the *maximum feature* parameter was conducted, reported in table 18.

Maximum Features	1	2	3	4	6	9	12
Accuracy	51.13%	51.16%	51.17%	51.12%	51.10%	51.09%	51.07%

Table 18: Random Forest's prediction accuracy based on the maximum amount of features it was given at each node with 200 decision trees, and no maximum depth

The results of the manual search documented above shows that the optimal number of features per node was 3 (in terms of accuracy), which was also the standard specification.
A manual search of the optimal *maximum depth* parameter was conducted as well, holding the other parameters constant. The results of the search is reported below in table 19. From

the table it can be seen that the accuracy was highest with a maximum depth of 20. 4 WML trading strategies were therefore created based on a *maximum depth*=20, using the methodology described in 9.2. The results of the pre-pruned portfolios were very comparable to the ones reported, but were actually slightly less significant, and are thus not reported in the paper.

Maximum Depth	10	15	20	25	30	40	50
Accuracy	50.92%	51.10%	51.23%	51.22%	51.19%	51.19%	51.13%

Table 19: Random Forest's prediction accuracy changing the maximum depth of the decision trees, with 200 decision trees, and 3 maximum features

Lastly, it should be re-emphasized that accuracy is not a very good measurement in this context, and hence a further investigation would be necessary to conclude which specifications are optimal, as discussed in section 1.4. However, the results reported in table 18 and 19 indicates that the algorithm is relatively insensitive to changes in the specifications.

10.3.2 Discussion of Random Forest Results

Table 20 below reports the monthly returns of the trading strategies with the belonging t-statistics.

Holding period=	1	3	6	9
Winner	0.0175	0.0144	0.0124	0.0116
	(3.09)	(2.7)	(2.46)	(2.39)
Loser	0.00781	0.0099	0.0108	0.0110
	(1.65)	(2.13)	(2.33)	(2.43)
WML	0.0096	0.0045	0.0017	0.0006
	(4.07)	(2.62)	(1.26)	(0.53)

Table 20: The table reports Random Forest's monthly returns and belonging t-statistics in the sample period Dec. 1994 to Dec. 2016. WML stands for Winners-Minus-Loser and is the long-short portfolios

The results displayed in table 20 shows that the Winner-Minus-Loser (WML) portfolio with a 1 month holding period was generating the highest return with an average monthly return of 0.96% in the sample period of 22 years, which is impressive. In addition, the annualized Sharpe ratio was 0.91, reported in table 21, indicating that a high standard deviation could not explain the high returns. From figure 18 below, it is also clear to see that the portfolio had a low volatility, and the volatility even seems to be low through-out both of the crises.

Holding period=	1	3	6	9
WML Sharpe ratio	0.91	0.57	0.27	0.11

Table 21: The table reports the four long-short Winners-Minus-Loser (WML) portfolios' Sharpe ratio in the sample period December 1994 to December 2016.

It is also clear from both table 20 and 21 that the length of the holding period and the performance of Random Forest's trading strategies has an inverse relationship. The WML portfolio with a 3-month holding period yielded a monthly return of 0.45% with a t-statistic of 2.62, and a Sharpe ratio of 0.57. The WML portfolios with a 6- and 9- month holding period yielded returns insignificantly different from zero and low Sharpe-ratios. Hence, the discussion will mostly be based on the results of the WML trading strategy with a 1-month holding period.

It is noteworthy that Random Forest's long-short strategies are the only long-short trading strategies generating positive returns with a significant t-statistic of all of the long-short machine learning based trading strategies tested. The source of the high returns of the WML portfolio with a 1 month holding period appears to be partly because the winner portfolio was doing extremely well, but also because the loser portfolio was generating far smaller returns in comparison. However, we are more interested in the results of regressing the trading strategies on CAPM to see if the WML portfolios actually yielded abnormal returns.

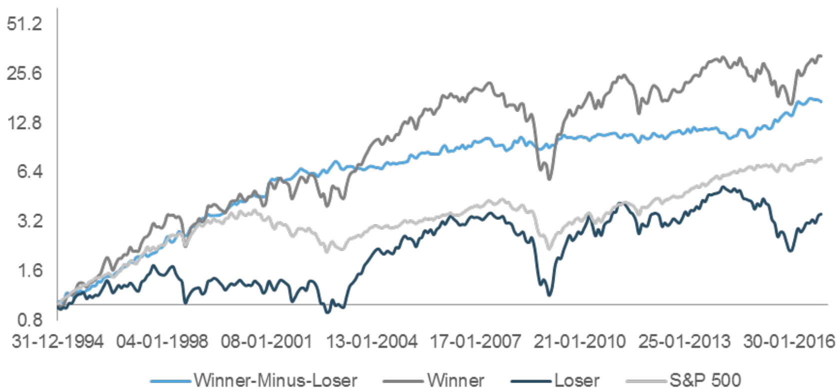

Figure 18: The figure displays the development of Random Forest's 1 month portfolios against S&P 500 on a logarithmic scale. Note that the risk-free rate has been added to the WML portfolio

The results of regressing the portfolios on CAPM is reported in table 22 with the belonging t-statistics. The table shows that Jensen's alpha of the 1-month holding period trading strategy was 0.80%, with a t-statistic of 3.46. This makes sense since the winner portfolio had a positive Jensen's alpha (although insignificant), and the loser portfolio had a negative Jensen's alpha (also insignificant). It is also clear from figure 18, that the winner portfolio on average was overperforming compared to the market, while the loser was underperforming on average. Moreover, the market beta of the WML portfolio was only 0.24, since the winner portfolio had a market beta of 1.62 and the loser had a market beta of 1.39.

Holding Period		Winner	Loser	WML
1	$\alpha_{1;p}$	0.0045	-0.0035	0.0080
		(1.2)	(-1.14)	(3.46)
	$\beta_{1;p}$	1.6233	1.3870	0.2364
		(18.61)	(19.26)	(4.39)
3	$\alpha_{3;p}$	0.0019	-0.0014	0.0033
		(0.55)	(-0.47)	(1.98)
	$\beta_{3;p}$	1.5545	1.3835	0.1710
		(19.11)	(20.12)	(4.4)
6	$\alpha_{6;p}$	0.0005	-0.0005	0.0011
		(0.15)	(-0.19)	(0.8)
	$\beta_{6;p}$	1.4715	1.3812	0.0903
		(19.23)	(20.54)	(2.96)
9	$\alpha_{9;p}$	-0.0000	-0.0001	0.0001
		(-0.01)	(-0.05)	(0.1)
	$\beta_{9;p}$	1.4217	1.3554	0.0664
		(19.64)	(20.54)	(2.71)

Table 22: The table report the Jensen's alpha and the CAPM beta of the portfolios based on Random Forest's stock predictions with the belonging t-statistics in the sample period of December 1994 to December 2016. WML stands for Winners Minus Loser and are the "zero-cost" portfolios.

It is noteworthy that the WML portfolios created by the predictions of Random Forest are the first of the long-short portfolios covered in this paper which does not have a negative market beta. The fact that the winner portfolio had such a high beta is not very surprising however, considering the fact that the monthly return was 1.75%. Figure 18 confirms the findings, as it shows that both the winner and the loser portfolio in general follows the market development but are much more volatile than the market. Moreover, it is clear that the winner portfolio consistently outperforms the loser portfolio even through times of the crises. This makes the portfolio very attractive, as it seems to be close to market neutral.

The 3-month WML portfolio also yielded a positive significant Jensen's alpha, but only barely significant with a t-statistic of 1.98, and the abnormal returns were insignificant for the longer holding periods.

Finally, the results of regressing the portfolios on Carhart (1997)'s four factor model reported in table 23 below were very interesting.

Holding Period		$\alpha_{K;p}$	Market	HML	SMB	MOM
	winner	0.0060	1.3785	0.4474	0.8044	-0.5045
		(2.18)	(20.58)	(4.81)	(9.43)	(-8.94)
1	loser	-0.0030	1.1927	0.5215	0.8256	-0.3956
		(-1.56)	(25.32)	(7.97)	(13.76)	(-9.97)
	WML	0.0091	0.1858	-0.0741	-0.0212	-0.1089
		(3.86)	(3.27)	(-0.94)	(-0.29)	(-2.28)
	winner	0.0037	1.2912	0.4319	0.8533	-0.5456
		(1.63)	(23.59)	(5.68)	(12.24)	(-11.83)
3	loser	-0.0013	1.2075	0.5327	0.8216	-0.3470
		(-0.7)	(27.69)	(8.79)	(14.79)	(-9.44)
	WML	0.0050	0.0837	-0.1008	0.03177	-0.1985
		(3.11)	(2.17)	(-1.88)	(0.65)	(-6.1)
	winner	0.0021	1.2217	0.4293	0.8308	-0.5186
		(1.04)	(24.68)	(6.24)	(13.17)	(-12.43)
6	loser	-0.0005	1.2139	0.5341	0.8036	-0.3275
		(-0.30)	(28.52)	(9.03)	(14.82)	(-9.13)
	WML	0.0027	0.0077	-0.1048	0.0272	-0.1912
		(2.2)	(0.26)	(-2.58)	(0.73)	(-7.76)
	winner	0.0013	1.1909	0.4340	0.8018	-0.4772
		(0.69)	(25.81)	(6.77)	(13.64)	(-12.27)
9	loser	-0.0002	1.1914	0.5327	0.7951	-0.3209
		(-0.09)	(28.83)	(9.28)	(15.1)	(-9.21)
	WML	0.0015	-0.0005	-0.0988	0.0066	-0.1563
		(1.54)	(-0.02)	(-3.05)	(0.22)	(-7.97)

Table 23: The table reports the results of regressing Random Forest's portfolios on Carhart (1997)'s four factor model in the sample period of December 1994 to December 2016, with belonging t-statistics. Random Forest was trained to predict monthly winners on historical return data only.

The abnormal returns of the 1- and 3-month WML portfolios could not be explained by adding the three factors. The 1-month WML portfolio still yielded the most significant abnormal return and will therefore still be the basis of the discussion going forward.

Both the 1-month winner- and the loser portfolio had a positive correlation with the HML and SMB factor, but as the loser portfolio was loading more on both factors it implied that the WML portfolio had a negative loading on both the SMB and HML factor. A particularly interesting observation is that both the winner and the loser portfolio had a very high factor loading on the SMB factor. Hence, it seems as if the algorithm has mostly identified small stocks to be both in the winner and in the loser portfolio. It is important to note that the two portfolios' constituents are mutually exclusive, so it has to be different small stocks in the portfolios. The reason for the high factor loading on the SMB factor could be because the algorithm has 'identified' the most volatile stocks to be the winners and loser, and as we have already mentioned, smaller stocks are usually more volatile (Munk, 2016). Hence, the algorithm's decision boundary might have been based on extremely high (low) returns for the winner (loser) stocks. This will be discussed further in section 10.4.

Both the winner and the loser portfolios had a significantly negative slope on the momentum factor. At first sight it was a bit surprising that the winner's correlation with the momentum portfolio was very negative, however holding it together with previous findings it makes sense. The winner portfolio had a high covariance with the market and a positive correlation with the HML factor. Both of which the momentum portfolio had a negative correlation with.

The winner portfolio had a more negative factor loading than the loser portfolio on the momentum factor, which implies that the WML portfolio also had a negative factor loading on the momentum factor.

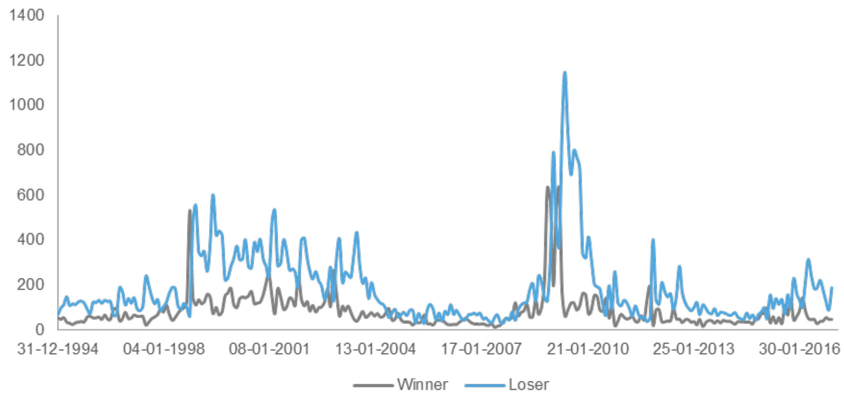

Figure 19: Number of stocks included in Random Forest's winner and loser portfolio in the test period

Figure 19 shows the number of predicted winners and losers throughout the sample period. The winner portfolio is most frequently consisting of around 60 stocks (median), which is very few stocks compared to the momentum trading strategies. The loser portfolio is requiring more trading: The portfolio size is most frequently around 121 (median), and is very dynamic, which is clear from figure 19. The winner portfolio seem to be very large around the times of crises whereas the loser portfolio seems to become extraordinary large in the aftermath of crises.

In conclusion the WML portfolios with a 1-month and 3-month holding period implemented on the basis of the Random Forest algorithm were generating abnormal returns, which could not be explained by well-known factors. In the following subsection it will be discussed whether it is feasible to follow the trading strategies in practice.

10.4 Applying machine learning for trading strategies

From the findings reported in the previous subsections it became clear that the results of applying machine learning algorithms to predict stocks based solely on time-series data turned out to be very varying, with the Random Forest algorithm as the most successful algorithm by far. This subsection will address the primary challenges of implementing the machine-learning based trading strategies from the perspective of a hedge fund. More specifically, we will discuss the transaction costs and capacity problems that might occur when following each algorithm's trading strategy.

The 6-Nearest Neighbor performed worst in terms of accuracy with an average prediction accuracy of 47.3%. Moreover, the confusion matrix indicated that the algorithm was unsuccessful, as the matrix revealed that the algorithm had picked many "false positive" stocks. However, the long-short 3-month WML portfolio generated positive returns, and a statistical significant Jensen's alpha. The investment strategy does not seem very profitable to follow however, as it generated very small monthly returns of 0.09% in the sample period. Especially when looking at figure 20 below, it seems to be a complicated trading strategy to implement as it involves a lot of trading each month, which implies that the turnover of assets are high. Hence, there will be some trading costs from both commission and bid-ask spreads. However, for a large hedge fund these costs would most likely be small. A large hedge fund would on the other hand take large positions, which means it would trade for more than what is available at the bid or ask price, as explained in section 8.5. Hence, taking transaction cost into account, the trading strategy would most likely not generate positive returns, let alone abnormal returns. However, we also argued that the WML portfolio appeared to be close to market neutral, which could make it an attractive investment if it is possible for the investor to leverage the portfolio costless. Leveraging the investment is not without risks however, and with the confusion matrix in mind, it would be hard to justify that following k-NN's predictions indeed is a low-risk trading strategy, hence this does not appear very appealing without further tests of the algorithm (Pedersen, 2015).

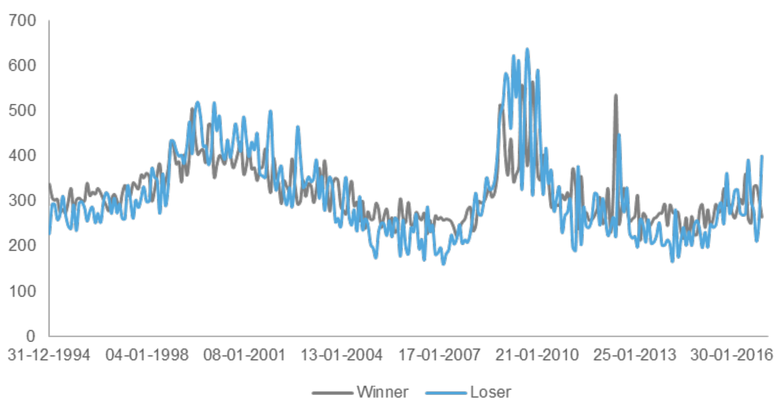

Figure 20: the figure displays the number of stocks included in 6-Nearest Neighbors's winner and loser portfolio in the test period

Naive Bayes Classifier was the best performing algorithm in terms of accuracy with a prediction accuracy of 51.6%, but as it turned out, it was the worst performing algorithm in terms of generating abnormal returns. The only portfolio based on Naive Bayes Classifier yielding a positive return was the portfolio with a one-month holding period, but the Sharpe ratio was only 0.20, and the Jensen's alpha was not significant. Hence, it was far from a successful trading strategy, and should not be implemented.

Random Forest was predicting the stock classes with a prediction accuracy of 51.2%, which meant it on average was predicting slightly worse than Naive Bayes Classifier. Moreover, the confusion matrix indicated that the algorithm had predicted many "false positive" stocks, which we wanted to avoid. Nevertheless, the long-short WML trading strategies with a holding period of 1- and 3-months were generating high monthly returns statistically different from zero, and significant Jensen's alphas. Moreover, regressing the portfolios on Carhart (1997)'s four factor model could not explain the abnormal returns. Thus, the conclusion from section 10.3.2 is that Random Forest algorithm seems to be able to create very attractive investment strategies. The best trading strategy was the WML portfolio with a 1 month holding period, which yielded the most significant abnormal returns and had a Sharpe ratio of 0.91. Thus, the following discussion will address the issues related to following Random Forest's stock predictions, based on the WML portfolio with a 1-month holding period:

Firstly, figure 19 in section 10.3.2 indicates that there is a large turnover of stocks, which will imply transaction costs in terms of both commission and bid-ask spreads, but also market impact costs, which we will discuss further below. In addition the loser portfolio is shorting over a thousand stocks around May 2009, which is documented in figure 19. This might not be feasible to its full extent. Firstly, because the lender of the stocks might want to get their assets back sooner than warranted - which is the "buy-in" risk, also discussed in section 8.5. Secondly, it might not be possible to short-sell all the stocks to the available bid prices at the exact time the algorithm is dictate. On the contrary, shorting a thousand stocks is better than shorting ten stocks in terms of market impact, as it implies smaller positions in each stocks which, everything else equal, makes the market impact smaller.

Secondly, the winner and loser portfolios' high factor loading on SMB makes it questionable whether the algorithm would be able to pick stocks well based on data with larger stocks, or if it only works because the algorithm has identified the small illiquid stocks as winners and losers. If the latter is true, it might not be feasible to implement the strategy for a hedge fund because the amount of stocks that can be traded at the available bid or ask price is too small compared to what a hedge fund would need to invest (Pedersen, 2015). In other words, market impact might make the trading strategy less profitable, and maybe even unprofitable. To address this issue, we firstly extracted the end-of-the month price and "shares outstanding" from CRSP of all the stocks in the test-set, and calculated the market value of equity (ME) for each month by saying $Price \quad \# \; shares \; outstanding$ (divided by 1,000,000). Secondly we extracted the 30% ME-fractile each month from French (2017)'s web-page, which is the 30% percentile of the market value of equity of stocks listed on New York Stock Exchange, calculated as the end-of-the-month *Price* \# *shares outstanding* (divided by 1,000,000). The 30%-percentile was chosen instead of the median to retain some robustness of the test, as Random Forest only tend to pick between 1% and 2% of the available stocks as winners and losers, which means the portfolios would become too small if we had used the median. The table below shows the 30% percentile and median market value of equity in the first- and last month of the sample period in USDm (French, 2017).

	30%-percentile	Median
December 1994	244.89	616.59
December 2016	1245.41	2757.86

Table 24: New York Stock Exchange's 30% percentile and median market value of equity in USDm

The stocks that had a market cap below the 30%-percentile in time t were removed only in time t, but still around 1500 stocks were completely removed from the data-set. The remaining 4800 stocks were then tested with Random Forest's algorithm. The results of the 1-month WML portfolio are reported in table 25, and all the results are fully disclosed in appendix F.
It turned out that the winner portfolio now consisted of 45 stocks on average instead of 60, and the loser portfolio had decreased to 91 stocks from 126 on average, which shows that the portfolio consisted of a high fraction of small cap stocks. The average monthly return of the 1-month WML portfolio on large cap stocks was less significant, but the portfolio was still yielding abnormal returns that could not be explained by the well-known factors, which increases the robustness of the results and the feasibility of applying Random Forest's long-short portfolio with a 1-month holding period as a trading strategy.

Monthly return	SR	CAPM α	CAPM β	Carhart α	Carhart β	HML	SMB	MOM
0.0091	0.78	0.0075	0.2320	0.0084	0.1901	-0.0779	-0.0285	-0.0860
(3.48)	-	(2.91)	(3.88)	(3.21)	(3.01)	(-0.89)	(-0.35)	(-1.61)

Table 25: The table report the results of Random Forest's 1-month WML portfolio restricted to only pick stocks larger than the 30% percentile of NYSE-listed stocks

Lastly, and perhaps the most significant issue to address of using Random Forest is the "black box issue", which is the lack of understanding the underlying assumptions that the model has based its predictions on. As we described in section 4, many funds state that they are not yet comfortable fully automating trading management if they do not understand how a particular prediction is made. With the Random Forest algorithm we can obtain little information about why the model has picked the stocks it did. It is only possible to extract what the algorithm finds the most important features which can be seen in table 26 below, which is not telling us a lot.

Feature	$r_{t\ 1}$	$r_{t\ 2}$	$r_{t\ 3}$	$r_{t\ 4}$	$r_{t\ 5}$	$r_{t\ 6}$	$r_{t\ 7}$	$r_{t\ 8}$	$r_{t\ 9}$	$r_{t\ 10}$	$r_{t\ 11}$	$r_{t\ 12}$
Feature Importance	8.6%	8.4%	8.4%	8.3%	8.3%	8.3%	8.3%	8.3%	8.3%	8.3%	8.3%	8.3%

Table 26: The table report the results of Random Forest's feature importance. It can be seen it assigns approximately the same importance to all the lagged returns.

To really interpret the model's decision boundaries it would require analysis of all the 200 individual decision trees, and as there was no maximum depth set, the length of the trees can be very deep. This would make the task infeasible. Hence it is a black box.

Despite this significant drawback, Random Forest clearly has some ability to predict winner and loser stocks as it has been able to make a long-short trading strategy yielding abnormal returns for 22 years, which cannot be explained by the most well-known factors. The combination of the black box issue and the results of the confusion matrix might make it hard to sell to investors however.
Instead of implementing Random Forest as a fully automated operator, which substitutes the traders it could be a complimentary tool. It could be a powerful complimentary tool for a trader because Random Forest is not merely trying to copy the way traders think and reason, but the algorithm is instead ”hunting” for similarities in the past 70 years, creating a set of rules to trade upon, which are too complex for a trader to find with manual search. Hence, Random Forest could be implemented as a tool working on the same principles as existing techniques used in systematic investing, to identify new signals or price movements. However, it would still require more research and tests to find the best possible indicators, as the model is not better than the data it receives.

11 Conclusion and Further Implications

This section consists of three subsections. Firstly, the main findings answering the research questions stated in section 1.3 will be concluded upon. Secondly, future work that could improve the results will be discussed along with other relevant areas of interest, which the scope of the thesis did not allow us to cover. Lastly, the thesis will end with final remarks of the future application of machine learning in trading of financial assets.

11.1 Findings & Conclusion

Overall the results of the thesis indicate that the use of machine learning algorithms can generate abnormal returns based on time-series data, which cannot be explained by well-known factors. However, there are some major drawbacks, which means that further research and tests are necessary.

The results presented in section 8 were in favor of the existence of momentum, as the WML momentum portfolios generated significant abnormal returns from December 1994 to December 2016. Hence, the results imply that abnormal profits can be realized from analyzing time-series patterns of stock returns. However, it was also discovered that applying momentum as a trading strategy involves risks of drawdowns in the aftermath of crises, consistent with the findings of Daniel and Moskowitz (2015).

The results presented in section 8.2 showed that a trading strategy based on a long position in both the long-short 6/3 WML momentum portfolio and the market portfolio performed better than the 6/3 WML momentum and market portfolio did alone. However, we discussed in section 8.5 that the issues of following the momentum-based trading strategies might make it less attractive net of transaction costs.

Based on the methodology presented in 9.2, two algorithms were capable of constructing long-short trading strategies yielding abnormal returns by only receiving a stock's returns of the past year, namely k-NN and Random Forest. Among the machine learning algorithms investigated, Random Forest created the most significant abnormal returns. The long-short WML portfolio with a 1-month holding period yielded a promising average monthly return of 0.96% and a Sharpe ratio of 0.91. Moreover, the portfolio yielded a significant Jensen's alpha, which could not be explained by the three additional factors of Carhart (1997)'s four factor model.

In section 10.4 we addressed the issue that both the long and short positions of Random Forest's WML portfolio with a holding period of 1 month appeared to consist mainly of small stocks. In order to help determining whether the strategy would be profitable to implement after transactions costs, Random Forest was tested again only on stocks with a market cap larger than the 30%-percentile. This caused the number of constituents in both the long- and short portfolios to decrease significantly, which underlined that Random Forest indeed had been identifying a lot of small stocks as winner and losers. The returns of the WML portfolio were still high with an average monthly return of 0.91% and a Sharpe ratio of 0.78 however. Moreover, the alphas were still significant when risk-adjusting with CAPM and Carhart's four factor model.

Lastly, it was concluded that the biggest short-coming of applying Random Forest in portfolio management is the the lack of understanding how a particular prediction is made. The black-box issue combined with the confusion matrix's unfavourable results reported in table 17, makes it seem unfeasible to fully automate the model presented in this thesis. For the algorithm to be fully automated, it would require more research and tests. The results reported are promising for applying machine learning in portfolio management however.

11.2 Further implications

While the thesis was limited to including time series data as the only input to the algorithms, it could be interesting to include other types of features. There are potentially many other inputs that could increase the performance, such as other "technical" inputs including volatility, the current month (due to the January effect described in section 3.4), moving averages, relative performance tools, size (to capture the SMB factor), and a HML input. Moreover, it could be very interesting to see the implications of incorporating fundamentals including profitability, coverage and leverage. As an example, Huerta et al. (2011) showed that applying fundamental and technical inputs jointly improved the results significantly on the SVM's ability to predict the winner and loser stocks (Huerta et al., 2011; Pedersen, 2015).

The machine learning models' performance could also be enhanced using the same inputs, but with a different methodology. One of the most significant issues of our methodology is the fact that we use an imbalanced training set. A well-known method called re-sampling, could help overcoming this problem by creating a more balanced (synthetic) training set. The most applied method is called over-sampling where you simply create synthetic data-points by interpolating data between several of the minority classes.
In addition it would be interesting to see if applying rolling windows of the training- and test-windows would improve the results of Random Forest (García, 2007).

Lastly it would be interesting to look at more advanced machine learning algorithms. Support Vector Machine (SVM) has in the literature proven to be a superior model for predicting stock index directions and future "winners" and "losers" (Huerta et al., 2011; Kumar & M., 2007).
Hence, further studies could include a comparison of the SVM and Random Forest. SVM requires a lot of pre-processing and tuning in order to perform however, and training it with a regular computer can take several weeks in terms of computational time.

11.3 Final remarks

The application of machine learning is growing vastly these years. More importantly, the applications of machine learning are no longer bounded to be merely repetitive tasks. Machine learning is now capable of identifying scin cancer as well as leading dermatologists, and automated cars are already implemented around the world. Tasks that usually would be defined as non-routine, non-repetitive tasks (Kubota, 2017). It is therefore not a question of "if", but "when" and "to which extend" fully automated machine learning trading-algorithms will be normalized. It will require a strong track record of "beating the market", and as it has been seen before in history, there can be pitfalls using algorithmic trading due to data or coding errors. As an example quant funds lost $100 billion dollars in a week in 2007 under what seemed to be normal market conditions, which is often referred to as a "black swan" incident. These black swan incidents are happening more frequently on a small scale however (Cohan, 2011). Clifford Asness, a widely acknowledged scholar and hedge fund manager, seems to believe it is not because of the algorithms but merely a result of...

> "... a strategy getting too crowded ... and then suffering when too many try to get out the same door"

(Cohan, 2011)

In the end, machine learning algorithms will be a part of the future in portfolio management, but it will require as much work as ever to gain profits from algorithmic strategies. It takes creativity to find new methods, inputs and applications, and most importantly risk management to avoid "black swan" incidents.

References

Andersen, I. (2013). *Den skinbarlige virkelighed - vidensproduktion i samfundsvidenskaberne.* Samfundslitteratur.

Asness, C. S., Moskowitz, T. J., & Pedersen, L. H. (2013). Value and momentum everywhere. *The Journal of Finance*, *68*(3), 929–985. doi: 10.1111/jofi.12021

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

Barberis, N., Shleifer, A., & Vishny, R. (1997). A model of investor sentiment. doi: 10.3386/w5926

Bollen, J., & Mao, H. (2011). Twitter mood as a stock market predictor. *Computer*, *44*(10), 91–94. doi: 10.1109/mc.2011.323

Bondt, W. F. M. D., & Thaler, R. (1985). Does the stock market overreact? *The Journal of Finance*, *40*(3), 793–805. doi: 10.1111/j.1540-6261.1985.tb05004.x

Bondt, W. F. M. D., & Thaler, R. H. (1987). Further evidence on investor overreaction and stock market seasonality. *The Journal of Finance*, *42*(3), 557. doi: 10.2307/2328371

Breiman, L. (2001). *Machine learning (2001).* Kluwer Academic Publishers.

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). *Classification and regression trees.* Taylor & Francis. Retrieved from `https://books.google.dk/books?id=JwQx-WOmSyQC`

Brooks, C. (2011). *Introductory econometrics for finance.* Cambridge University Press.

Brynjolfsson, E., & Mcafee, A. (2017). *What's driving the machine learning explosion?* Retrieved 2018-05-11, from `https://hbr.org/2017/07/whats-driving-the-machine-learning-explosion`

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (2011). *The econometrics of financial markets.* Princeton University Press.

Carhart, M. M. (1997). On persistence in mutual fund performance. *the Journal of Finance*, *LII*(1).

Chan, K. (1988). On the contrarian investment strategy. *journal of Business*, *61*(2), 147–163.

Chan, L. K. C., Hamao, Y., & Lakonishok, J. (1991). Fundamentals and stock returns in japan. *the Journal of Finance*, *46*(5), 1739-64.

Cohan, W. D. (2011). *Man vs. machine on wall street: How computers beat the market.* Retrieved 2018-04-18, from `https://www.theatlantic.com/business/archive/2011/03/man-vs-machine-on-wall-street-how-computers-beat-the-market/73120/`

Conrad, J., & Kaul, G. (1998). An anatomy of trading strategies. *Journal of Finance*, *11*.

Cuthbertson, K., & Nitzsche, D. (2004). *Quantitative financial economics: stocks, bonds and foreign exchange.* Wiley.

Daniel, K., Hirshleifer, D., & Subrahmanyam, A. (1998). Investor psychology and security market under- and overreactions. *The Journal of Finance*, *53*(6), 1839–1885. doi: 10.1111/0022-1082.00077

Daniel, K., & Moskowitz, T. (2015). Momentum crashes. doi: 10.3386/w20439

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383. doi: 10.2307/2325486

Fama, E. F. (1991). Efficient capital markets: Ii. *THE JOURNAL OF FINANCE*, *XLVI*(5).

Fama, E. F. (1997). Market efficiency, long-term returns, and behavioral finance. *SSRN Electronic Journal*. doi: 10.2139/ssrn.15108

Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, *47*(2), 427-465.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3-56. doi: 10.1016/0304-405x(93)90023-5

Fama, E. F., & French, K. R. (1996). Multifactor explanations of asset pricing anomalies. *The Journal of Finance*, *51*(1), 55–84. doi: 10.1111/j.1540-6261.1996.tb05202.x

Financial Stability Board. (2017). *Artificial intelligence and machine learning in financial services.* Retrieved 2018-03-22, from `http://www.fsb.org/2017/11/artificial-intelligence-and-machine-learning-in-financial-service/`

Finanswatch. (2014). *Maj invest siger nej til sinai.* Retrieved 2018-03-22, from `https://finanswatch.dk/Finansnyt/article6751217.ece`

French, K. R. (2017). Retrieved 2018-03-22, from `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`

García, J. M. R. A. R. S. J., V. Sánchez. (2007). The class imbalance problem in pattern classification and learning. *II Congreso Espanolde Informática (CEDI 2007)*(3), 978-84.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2017). *The elements of statistical learning: data mining, inference, and prediction.* Springer.

Hong, H., & Stein, J. (1999). Differences of opinion, rational arbitrage and market crashes. doi: 10.3386/w7376

Huerta, R., Elkan, C., & Corbacho, F. (2011). Nonlinear support vector machines can systematically identify stocks with high and low future returns. *SSRN Electronic Journal*. doi: 10.2139/ssrn.1930709

Imandoust, S. B., & Bolandraftar, M. (2014). Forecasting the direction of stock market index movement using three data mining techniques: the case of tehran stock exchange. *International Journal of Engineering Research and Applications*, *4*(6), 106-117.

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of finance*, *48*(1).

Jegadeesh, N., & Titman, S. (2001). Profitability of momentum strategies: An evaluation of alternative explanations. *Journal of finance*, *56*(1).

Jensen, M. C., & Benington, G. A. (1970). Random walks and technical theories: Some additional evidence. *Journal of finance*, *25*(2).

Kara, K. (2011). *Kurt kara / maj invest – artificial intelligence investor conference.* Retrieved 2018-04-20, from `http://www.machinelearning.ai/artificial-intelligence/kurt-kara-maj-invest-artificial-intelligence-investor-conference/`

Kubota, T. (2017). *Deep learning algorithm does as well as dermatologists in identifying skin cancer.* Retrieved 2018-04-18, from `https://news.stanford.edu/2017/01/25/`

artificial-intelligence-used-identify-skin-cancer/

Kumar, M., & M., T. (2007). Support vector machines approach to predict the sp cnx nifty index returns. *SSRN Electronic Journal*. doi: 10.2139/ssrn.962833

Levy, R. A. (1967). Relative strength as a criterion for investment selection. *Journal of finance*, *48*(1).

Munk, C. (2016). *Financial markets and investments.*

Müller, A. C., & Guido, S. (2017). *Introduction to machine learning with python - a guide for data scientists.* O'Reilly.

Pedersen, L. H. (2015). *Efficiently inefficient: how smart money invests and market prices are determined.* Princeton University Press.

PWC. (2017). *The future of ai is here.* Retrieved 2018-03-22, from http://pwcartificialintelligence.com/

Scikit-Learn. (2017a). *1.6. nearest neighbors.* Retrieved 2018-03-23, from http://scikit-learn.org/stable/modules/neighbors.html

Scikit-Learn. (2017b). *1.9. naive bayes.* Retrieved 2018-03-23, from http://scikit-learn.org/stable/modules/naive_bayes.html

Scikit-Learn. (2017c). *Decision trees.* Retrieved 2018-04-28, from http://scikit-learn.org/stable/modules/tree.html

Zarowin, P. (1990). Size, seasonality, and stock market overreaction. *Journal of Financial and Quantitative analysis*, *25*(01), 113–125.

Appendices

A Our results vs. Jegadeesh & Titman's results

The data was in the sample period of January 1965 to December 1989, using stocks listed on NYSE, AMEX and NASDAQ from July 1962 to December 1989.

J	K=	3	6	9	12	J	K=	3	6	9	12
3		0.0032	0.0058	0.0061	0.0069	3		0.0033	0.0064	0.0063	0.0067
		(1.10)	(2.29)	(2.69)	(3.53)			(1.11)	(2.32)	(2.9)	(3.50)
6		0.0084	0.0095	0.0102	0.0086	6		0.0083	0.0095	0.009	0.0089
		(2.44)	(3.07)	(3.76)	(3.36)			(2.40)	(3.01)	(3.67)	(3.47)
9		0.0109	0.0121	0.0105	0.0082	9		0.114	0.0122	0.0109	0.0084
		(3.03)	(3.78)	(3.47)	(2.89)			(3.10)	(3.81)	(3.37)	(2.98)
12		0.0131	0.0114	0.0093	0.0068	12		0.0128	0.0114	0.0085	0.0071
		(3.74)	(3.40)	(2.95)	(2.25)			(3.74)	(3.40)	(2.95)	(2.25)

Table 27: The results of testing our methodology (right) on the same data a Jegadeesh & Titman 1993 (left).

B WML portfolios decomposed

J	K=	3	6	9	12
3	winner	0.0059	0.00528	0.0049	0.0043
		(2.69)	(2.52)	(2.52)	(2.3)
	loser	-0.0019	-0.0017	-0.0013	-0.0010
		(-0.56)	(-0.51)	(-0.42)	(-0.34)
6	winner	0.0075	0.0064	0.0055	0.0043
		(3.26)	(3.02)	(2.8)	(2.31)
	loser	-0.0032	-0.0029	-0.0023	-0.0013
		(-0.84)	(-0.79)	(-0.68)	(-0.4)
9	winner	0.0074	0.0061	0.0047	0.0036
		(3.3)	(2.87)	(2.39)	(1.9)
	loser	-0.0038	-0.0032	-0.0022	-0.0011
		(-0.93)	(-0.85)	(-0.6)	(-0.31)
12	winner	0.0061	0.0044	0.0034	0.0026
		(2.8)	(2.12)	(1.73)	(1.39)
	loser	-0.0038	-0.0025	-0.0014	-0.0005
		(-0.94)	(-0.65)	(-0.40)	(-0.15)

Table 28: CAPM alpha

J	K=	3	6	9	12
3	winner	0.9471	1.0055	1.0269	1.057
		(18.5)	(20.89)	(22.76)	(24.33)
	loser	1.5317	1.4474	1.3885	1.3290
		(19.46)	(19.29)	(19.27)	(19.59)
6	winner	0.9481	0.9997	1.0356	1.0555
		(17.9)	(20.33)	(22.75)	(24.35)
	loser	1.5715	1.4817	1.4034	1.3542
		(17.71)	(17.6)	(17.75)	(18.18)
9	winner	0.9617	1.0200	1.0477	1.0663
		(18.48)	(20.87)	(22.81)	(24.46)
	loser	1.5933	1.4908	1.4253	1.3700
		(16.86)	(16.88)	(17.11)	(17.44)
12	winner	0.9891	1.0311	1.0575	1.0686
		(19.48)	(21.35)	(23.23)	(24.54)
	loser	1.5645	1.490	1.4211	1.3683
		(16.49)	(16.67)	(16.86)	(17.17)

Table 29: CAPM beta

J	K=	3	6	9	12
3	winner	0.0041	0.0034	0.0032	0.0027
		(2.79)	(2.66)	(2.75)	(2.41)
	loser	-0.0040	-0.0039	-0.0037	-0.0034
		(-1.39)	(-1.46)	(-1.46)	(-1.49)
6	winner	0.0059	0.00498	0.0042	0.0031
		(3.62)	(3.33)	(3.13)	(2.48)
	loser	-0.0056	-0.0054	-0.0049	-0.0039
		(-1.68)	(-1.74)	(-1.74)	(-1.51)
9	winner	0.0062	0.0049	0.0037	0.0026
		(3.63)	(3.11)	(2.56)	(1.95)
	loser	-0.0065	-0.0061	-0.0050	-0.0039
		(-1.85)	(-1.88)	(-1.69)	(-1.42)
12	winner	0.0052	0.0036	0.0026	0.0019
		(3.01)	(2.23)	(1.78)	(1.33)
	loser	-0.0068	-0.0055	-0.0044	-0.0035
		(-1.95)	(-1.79)	(-1.5)	(-1.26)

Table 30: FF3 alpha

J	K=	3			6			9			12		
		Market	HML	SMB	Market	HML	SMB	Market	HML	SMB	Market	HML	SMB
3	winner	0.9225	0.4445	0.6244	0.9777	0.4336	0.6041	0.9995	0.4080	0.6231	1.0285	0.3881	0.5277
		(27.6)	(9.26)	(13.88)	(33.12)	(10.23)	(15.2)	(36.93)	(10.5)	(15.36)	(39.24)	(10.32)	(14.96)
	loser	1.4881	0.6108	0.6076	1.4107	0.6563	0.6047	1.3546	0.6695	0.6123	1.2980	0.6769	0.6174
		(22.37)	(6.4)	(6.79)	(22.79)	(7.38)	(7.25)	(23.35)	(8.04)	(7.84)	(24.7)	(8.97)	(8.72)
6	winner	0.9183	0.3486	0.6153	0.9694	0.3291	0.5579	1.0047	1.0047	0.5126	1.0240	0.2828	0.4845
		(24.71)	(6.53)	(12.3)	(28.41)	(6.72)	(12.14)	(32.6)	(6.89)	(12.35)	(35.36)	(6.8)	(12.43)
	loser	1.5300	0.6869	0.6444	1.4455	0.7184	0.6466	1.3709	0.7329	0.6529	1.3234	0.7286	0.6525
		(20.01)	(6.26)	(6.26)	(20.37)	(7.05)	(6.77)	(21.21)	(7.9)	(7.5)	(22.38)	(8.58)	(8.19)
9	winner	0.9310	0.2609	0.5419	0.9882	0.2444	0.4922	1.0148	0.2195	0.4661	1.0331	0.2090	0.4447
		(23.86)	(4.66)	(10.32)	(27.05)	(4.66)	(10.01)	(30.33)	(4.57)	(10.35)	(33.27)	(4.69)	(10.64)
	loser	1.5559	0.7914	0.6844	1.4588	0.8064	0.6800	1.3953	0.8023	0.6828	1.3412	0.7902	0.6814
		(19.21)	(6.8)	(6.8)	(19.74)	(-1.85)	(6.83)	(20.53)	(8.22)	(7.46)	(21.48)	(8.82)	(8.11)
12	winner	0.9565	0.1919	0.4826	0.9967	0.1611	0.4541	1.0233	0.1544	0.4281	1.0350	0.1564	0.4079
		(24.24)	(3.39)	(9.08)	(26.93)	(3.03)	(9.12)	(29.93)	(3.15)	(9.3)	(32)	(3.37)	(9.37)
	loser	1.5325	0.8541	0.7070	1.4600	0.8486	0.7046	1.3927	0.8386	0.7068	1.3408	0.8242	0.7005
		(19.12)	(7.42)	(6.55)	(19.77)	(8.00)	(7.09)	(20.49)	(8.6)	(7.73)	(21.34)	(9.14)	(8.28)

Table 31: FF3 factors

C Lagged Momentum

Skipping 1 month between the formation period and holding period.

J	K=	3	6	9	12
3		0.40%	0.34%	0.30%	0.26%
		(1.49)	(1.50)	(1.48)	(1.57)
6		0.61%	0.54%	0.46	0.27%
		(1.69)	(1.68)	(1.66)	(1.13)
9		0.67%	0.57%	0.38%	0.19%
		(1.67)	(1.62)	(1.24)	(0.71)
12		0.63%	0.39%	0.22%	0.07%
		(1.43)	(0.99)	(0.64)	(0.24)

Table 32: Returns of the 16 momentum portfolios lagged 1 month

D Hedged Momentum

Full results of the long position in the (6/3) trading strategy and the market

Monthly return	Sharpe ratio	CAPM alpha	CAPM beta	FF alpha	FF beta	HML	SMB
0.0151	0.8450	0.0126	0.3769	0.0137	0.3759	-0.3099	-0.2100
(3.65)	-	(3.08)	(3.98)	(3.38)	(4.03)	(-2.31)	(-1.67)

Table 33: Returns of the 16 momentum portfolios lagged 1 month

E Decision Tree

E.0.0.1 No Risk-adjustment

Holding period=	1	3	6	9
winner	0.0153	0.0133	0.0106	0.0096
	(2.39)	(2.25)	(1.93)	(1.82)
loser	0.0096	0.0113	0.0116	0.0116
	(1.96)	(2.42)	(2.56)	(2.63)
WML	0.0057	0.0020	-0.0010	-0.0021
	(1.64)	(0.82)	(-0.53)	(-1.30)

Table 34: The table report Decision trees monthly returns and t-statistics in the sample period Dec. 1994 to Dec. 2016. WML stands for Winners-Minus-Loser and is the long-short portfolios

Holding period=	1	3	6	9
WML Sharpe ratio	0.36	0.18	-0.11	-0.27

Table 35: The table reports the four Decision Tree long-short (WML) portfolios' Sharpe ratio in the sample period December 1994 to December 2016.

E.0.0.2 CAPM

Holding Period		Winner	Loser	WML
1	$\alpha_{1;p}$	0.0017	-0.0018	0.0036
		(0.38)	(-0.55)	(1.03)
	$\beta 1, p$	1.7087	1.3926	0.3161
		(15.89)	(18.24)	(3.96)
3	$\alpha_{3;p}$	0.0003	0.0001	0.0002
		(0.07)	(-0.55)	(0.08)
	$\beta 3, p$	1.6332	1.3630	0.2701
		(16.94)	(19.29)	(4.85)
6	$\alpha_{6;p}$	-0.0017	0.0006	-0.0023
		(-0.44)	(0.2)	(-1.2)
	$\beta 6, p$	1.5224	1.3382	0.1842
		(17.09)	(19.87)	(4.21)
9	$\alpha_{9;p}$	-0.0023	0.0008	-0.0031
		(-0.65)	(0.27)	(-1.99)
	$\beta 9, p$	1.4710	1.3180	0.1530
		(17.55)	(20.31)	(4.23)

Table 36: The table report the Jensen's alpha and the CAPM beta of the portfolios based on Decision tree with the belonging t-statistics in the sample period of December 1994 to December 2016.

E.0.0.3 Carhart's 4-factor

Holding Period		$\alpha_{k;p}$	Market	HML	SMB	MOM
	winner	0.0048	1.3656	0.3850	0.9255	-0.7130
		(1.37)	(16.12)	(3.27)	(8.58)	(-9.99)
1	loser	-0.0018	1.2079	0.5791	0.8727	-0.3616
		(-0.8)	(22.77)	(7.86)	(12.91)	(-8.09)
	WML	0.0066	0.1577	-0.1940	0.0528	-0.3514
		(1.96)	(1.95)	(-1.73)	(0.51)	(-5.16)
	winner	0.0032	1.2943	0.4133	0.9492	-0.7114
		(1.15)	(19.28)	(4.43)	(11.1)	(-12.58)
3	loser	0.0000	1.1905	0.5736	0.8534	-0.3401
		(0.15)	(26.4)	(9.16)	(14.85)	(-8.95)
	WML	0.0032	0.1039	-0.1603	0.0958	-0.3713
		(1.47)	(1.98)	(-2.2)	(1.43)	(-8.39)
	winner	0.0006	1.2164	0.4863	0.9418	-0.6486
		(0.24)	(20.85)	(6.00)	(18.10)	(-13.19)
6	loser	0.0003	1.1783	0.5688	0.8327	-0.3092
		(0.19)	(28.16)	(9.78)	(15.62)	(-8.77)
	WML	0.0002	0.0381	-0.0825	0.1091	-0.3394
		(0.15)	(0.97)	(-1.52)	(2.19)	(-10.29)
	winner	-0.0006	1.1954	0.4997	0.9306	-0.5756
		(-0.29)	(22.15)	(6.66)	(13.53)	(-12.65)
9	loser	0.0005	1.1644	0.5533	0.8023	-0.2956
		(0.29)	(29.1)	(9.95)	(15.74)	(-8.76)
	WML	-0.0011	0.03104	-0.0535	0.1283	-0.2801
		(-0.84)	(0.97)	(-1.2)	(3.14)	(-10.36)

Table 37: The table reports the results of regressing Decision Trees's portfolios on Carhart (1997)'s four factor model in the sample period of December 1994 to December 2016, with belonging t-statistics.

F Random Forest - Large Cap

F.0.0.1 No Risk-adjustment

Holding period=	1	3	6	9
winner	0.0171	0.0142	0.0126	0.0120
	(2.93)	(2.59)	(2.43)	(2.41)
loser	0.0080	0.0097	0.0105	0.0109
	(1.63)	(2.03)	(2.21)	(2.35)
WML	0.0091	0.0045	0.0022	0.0011
	(3.48)	(2.44)	(1.48)	(0.94)

Table 38: The table report Large Cap Random Forest's monthly returns and t-statistics in the sample period Dec. 1994 to Dec. 2016. WML stands for Winners-Minus-Loser and is the long-short portfolios

Holding period=	1	3	6	9
Winner-Minus-Loser	0.7781	0.5317	0.3181	0.2006

Table 39: The table reports the four long-short Winners-Minus-Loser (WML) portfolios' Sharpe ratio in the sample period December 1994 to December 2016.

F.0.0.2 CAPM

Holding Period		Winner	Loser	WML
1	$\alpha_{1;p}$	0.0038	-0.0037	0.0075
		(0.96)	(-1.18)	(2.91)
	$\beta 1, p$	1.6774	1.4454	0.2320
		(18.6)	(19.69)	(3.88)
3	$\alpha_{3;p}$	0.0015	-0.0020	0.0034
		(0.41)	(-0.65)	(1.88)
	$\beta 3, p$	1.5859	1.4290	0.1569
		(18.82)	(20.45)	(3.7)
6	$\alpha_{6;p}$	0.0004	-0.0011	0.0016
		(0.13)	(-0.38)	(1.08)
	$\beta 6, p$	1.5103	1.4245	0.0857
		(19.11)	(20.85)	(2.52)
9	$\alpha_{9;p}$	0.0001	-0.0006	0.0007
		(0.03)	(-0.19)	(0.56)
	$\beta 9, p$	1.4643	1.3988	0.0655
		(19.74)	(21.08)	(2.38)

Table 40: The table report the Jensen's alpha and the CAPM beta of the portfolios based on Large Cap Random Forest's stock predictions with the belonging t-statistics in the sample period of December 1994 to December 2016. WML stands for Winners Minus Loser and are the "zero-cost" portfolios.

F.0.0.3 Carhart's 4-factor

Holding Period		$\alpha_{K;p}$	Market	HML	SMB	MOM
	winner	0.0054	1.4337	0.4413	0.7618	-0.5048
		(1.79)	(19.82)	(4.39)	(8.27)	(-8.28)
1	loser	-0.0030	1.2436	0.5192	0.7904	-0.4188
		(-1.48)	(25.09)	(7.5)4	(12.52)	(-10.03)
	WML	0.0084	0.1901	-0.0779	-0.0285	-0.0860
		(3.21)	(3.01)	(-0.89)	(-0.35)	(-1.61)
	winner	0.0034	1.3152	0.4524	0.8478	-0.5654
		(1.39)	(22.49)	(5.57)	(11.38)	(-11.48)
3	loser	-0.0016	1.2460	0.5297	0.7978	-0.3663
		(-0.87)	(27.29)	(8.35)	(13.71)	(-9.52)
	WML	0.0050	0.0692	-0.0773	0.0500	-0.1992
		(2.84)	(1.62)	(-1.3)	(0.92)	(-5.54)
	winner	0.0022	1.2521	0.4509	0.8266	-0.5402
		(1.39)	(23.89)	(6.19)	(12.38)	(-12.23)
6	loser	-0.0009	1.2500	0.5282	0.7821	-0.3467
		(-0.51)	(28.06)	(8.53)	(13.78)	(-9.23)
	WML	0.0031	0.0020	-0.0773	0.0445	-0.1935
		(2.26)	(0.06)	(-1.68)	(1.05)	(-6.91)
	winner	0.0016	1.2232	0.4392	0.7819	-0.5022
		(0.82)	(25.14)	(6.5)	(12.62)	(-12.25)
9	loser	-0.0004	1.2278	0.5206	0.7625	-0.3406
		(-0.21)	(28.75)	(8.78)	(14.02)	(-9.47)
	WML	0.0020	-0.0046	-0.0814	0.0194	-0.1615
		(1.82)	(-0.17)	(-2.19)	(0.57)	(-7.17)

Table 41: The table reports the results of regressing Large Cap Random Forest's portfolios on Carhart (1997)'s four factor model in the sample period of December 1994 to December 2016, with belonging t-statistics. Random Forest was trained to predict monthly winners on historical return data only.

G Codes

See USB stick both Stata and Python files

H Data

See USB stick