COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

2020

Programme: MSc in Business Administration and Information Systems
Master’s Thesis
Supervisor: Daniel Hardt
Student: Fredrik Ahnve, 115581
Student: Kasper Fantenberg, 124429
Student: Gustav Svensson, 124430
Date: 15th of May 2020

Algorithmic Trading – Predicting Stock Prices with Text Data

AN EVENT STUDY ON THE SWEDISH STOCK MARKET WITH CORPORATE AD-HOC DISCLOSURES USING SUPERVISED MACHINE LEARNING CLASSIFICATION

Number of pages: 149 / 160

Number of characters: 296 973 / 364 000

Abstract

This study mines circa 30 000 textual corporate ad-hoc disclosures issued during the last ten years by publicly traded companies on Nasdaq OMX Stockholm. Natural Language Processing methods are used together with supervised Machine Learning modeling to predict stock price movements after the disclosures are published. The study follows an event study structure and evaluates three labeling methods based in financial theory. Jensen's Alpha in the context of the Capital Asset Pricing Model, which is the most sophisticated model used, best assists the supervised labeling. This indicates that financial models can help isolate the effect of an informational event on stock prices. The results show that the best text data pre-processing method is TF-IDF using character grams that together with the best classifier Logistic Regression form the best Machine Learning model. The model produces a leverage of 6,3 percentage points above the ZeroR baseline. Finally, an algorithmic trading strategy is simulated using the model to evaluate whether it can create significant positive abnormal returns on the stock market. Several of the simulated trading strategies produce positive abnormal returns but none of them are statistically significant on a 0,05 level. Many improvement areas are identified for the machine learning model and the algorithmic trading strategy with potential to improve performance with relevance for future research on stock price prediction with textual data.

Keywords: Algorithmic Trading, Machine Learning, Stock Price Prediction, Ad-Hoc Disclosures, Natural Language Processing, Text Mining, Finance

Special thanks to,

Daniel Hardt, our supervisor, for his enthusiasm and helpful input troughout the process, and Klara Svensson, student at Stockholm School of Economics, for helping us with access to necessary data for the Trading Strategy part of the thesis.

1. Introduction

Predicting stock prices is a major branch in the field of financial research (Kim et al., 2018). The area is appealing both from an academic perspective and from the perspective of industry practitioners as this challenging task can create monetary value (dos Santos Pinheiro & Dras, 2017a). This branch has in spite of continuous efforts and different approaches not experienced significant improvements in the recent past (Kim et al., 2018). The slow progress can be explained by the challenge to discover new and meaningful factors that succeed in explaining stock price variance (Kim et al., 2018). Traditional measures generally use mathematical models with quantitative variables, such as historical time-series of stock prices and macro-/microeconomic indicators (Kim et al., 2018). By discovering new factors, financial research and stock price prediction can be taken to the next level.

Recent studies have increasingly focused on textual data for market analysis since words, phrases and documents in the financial sector convey relevant information that can be linked to investor sentiment and behavior (Kim et al., 2018). Analyzing such text documents and how markets react to them with machine learning methods is a novel approach for predicting stock prices (Kim et al., 2018). Analysis of financial text data is gaining growing attention as it provides an increasingly important approach to answering fundamental questions within the financial field (Kearney & Liu, 2014). The results of such studies have shown promising results in the latest years with implications for financial and economic theory, but there is clearly room for improvement, especially in regard to extracting relevant information from the documents (dos Santos Pinheiro & Dras, 2017a). With text mining as the overall research area of using text data as input to make inferences, finance constitutes just one of countless areas where text mining techniques can be deployed (Feinerer et al., 2008).

Mining the qualitative information in the text data, and subsequently using it in equity pricing models, may complement quantitative informational measures to enable better understanding of stock price variance (Kearney & Liu, 2014). Furthermore, text data can be a more independent way of testing market efficiency compared to quantitative, number-based measures. The underlying reason for this is that many of the quantitative measures are highly correlated, why anomalies may reflect the same regularity (Kearney & Liu, 2014). Studies have also confirmed that stock price prediction has improved when using text data rather than numerical data alone (Kim et al., 2018).

With recent advancements in computational power, algorithmic trading has become a trend in finance (dos Santos Pinheiro & Dras, 2017a). The concurrent advancements in machine learning and language processing have resulted in increased use of unstructured data as information source for investment strategies (Fisher et al., 2016). As traders make decisions every day whether to sell, buy or keep stocks, investment strategies with the help of clever algorithms and models can help investors make fast and possibly also more correct decisions. A quick reaction to new information is an important component in trading strategies (Leinweber & Sisk, 2011). According to the Efficient Market Hypothesis, it should however not be possible to outperform the market in the long term, where new public information should be incorporated in the market prices without delay (Fama 1991). Hence, researchers find it interesting to challenge the notion of market efficiency through predicting stock market behavior from newly disclosed information (dos Santos Pinheiro & Dras, 2017a), Muntermann & Guettler (2007) and Bank & Baumann (2015) focus on the perspective of information incorporation delay. Their results show that there seems to exist a time lag before stock prices fully reflect new, ad-hoc, information published by companies. As a computer might be faster at reading text than a human, the existence of information incorporation lag might allow a computer to make a decision and act on this decision before the market in general has reacted, thereby generating profits, given that the computer is successful at predicting stock prices.

Feuerriegel & Gordon (2018) build an algorithmic trading system based on corporate ad-hoc textual disclosures and show that, on the US market, it is capable of being profitable as a trading strategy. Feuerriegel & Gordon (2018) further state that even though their results of ad-hoc disclosures are clearly linked to economic outlook and imply predictive capabilities, the analysis needs to be repeated on several different markets. An ad-hoc disclosure is defined as a corporate disclosure that is made “ad-hoc”, i.e. unexpectedly, such as an unexpected press release which is the documents of. Non-ad-hoc disclosures are pre-scheduled disclosures, such as quarterly and annual reports (Feuerriegel & Gordon, 2018).

Based on previous research within this field, we create a machine learning model based on corporate ad-hoc disclosures. After a literature review on the subject, we find no studies that conduct stock price prediction with corporate ad-hoc disclosures on the Swedish stock market. As the Swedish stock market has relatively easily accessible data while also being a new market to conduct this kind of study on, it is deemed a reasonable delimitation for this study’s purpose and scope to focus on the Swedish market. To test the machine learning model on the stock market to evaluate the actual predictability of the model, a trading strategy is simulated on the Swedish market.

1.1. Research question

Can a machine learning model trained on textual corporate ad-hoc disclosures from the Swedish stock market exceed relevant baselines and generate positive abnormal returns used as a trading strategy?

Answering the research question contributes to the scientific areas of Machine Learning, Natural Language Processing in finance and financial research on stock markets. The study covers a research gap regarding similar studies on the Swedish stock market.

2. Theoretical background and literature review

2.1. Introduction

In this section we provide a review of information and previous studies relevant to the background of the research questions of this study, to have the best basis for constructing a machine learning model that predicts stock price movements from financial text data. The theory section is divided into five parts

1. **Financial Theory -** First, we introduce financial theory introducing traits, concepts and regulations relating to stock markets as a background for the context of the study.
2. **Automated trading strategies -** Event based trading and algorithmic trading are discussed to relate automated trading strategies to machine learning models from a practical perspective.
3. **Artificial Intelligence and Machine learning -** The technical concepts are introduced to provide context for this study in relation to learning and acting intelligently.
4. **Natural Language processing and literature review -** We link the financial and technical areas by introducing the concept of Natural Language Processing, and present previous Natural Language Processing studies in the financial sector, with an emphasis on the prediction of stock price movements with ad-hoc disclosures.
5. **Evaluation -** We discuss how a machine learning model can be evaluated from different perspectives.

2.2. Financial Theory

2.2.1. Stock markets

"The stock market" is a collective term for all public markets and exchanges where stocks are bought and sold. Stock markets are commonly referred to as secondary markets as stocks are traded among investors without directly affecting the company itself. This differs from the so-called primary markets, where stocks are bought directly from the company, thus adding more capital to the company. Stock market performance is generally quantified in indices that aggregates of the returns of companies listed on the market in order to reflect how the market in general and on average is developing. These indices can be comprised in various ways, e.g. by sector, size or to reflect all companies. (Young, 2019)

2.2.2. Information as a commodity on stock markets

Information comes in many forms and has many meanings. Depending on the context in which it is used, the concept can be defined differently (Floridi, 2010). A General Definition of Information (GDI) comprising "data + meaning" has become more prevalent over the past decades (Floridi, 2010). GDI has become the operational standard, especially in fields that treat data and information as reified entities, that is, as something that is concretized and can be manipulated, such as through data mining, text mining and information management (Floridi, 2010).

Information is the raw material of all financial decisions, why producing and processing information lies at the heart of the theory of the firm and of the study of financial markets and institutions (Liberti & Petersen, 2018). Seen as a commodity in an economical sense, information receives value because of its usefulness: it allows agents to take courses of actions (considering options, avoiding errors, making choices, etc.) with higher expected payoff or return (expected utility) than without the information (Floridi, 2010). Information has three properties that differentiate it from other "ordinary" goods. First, it is non-rivalrous. If I hold some information, another agent can hold the same piece of information simultaneously. This is not possible with goods in a traditional economic sense, such as a loaf of bread (Floridi, 2010). Second, information tends to be non-excludable, meaning it is easily disclosed and sharable. Some information (e.g. company insider information, intellectual property, military secrets, or other sensitive information) might however be protected by the holder, but exclusion is not a natural property of information, why protection demands a positive effort by the holder (Floridi, 2010). Third, the cost of reproducing information tends to be negligible (approximately zero marginal cost) once it is available (Floridi, 2010). Arguably, this has become even more prevalent in the information technology era, where information stored in text, sound and picture format has become decoupled from its physical containers such as books and video tapes, and instead easily shared over the Internet.

In summary, information is presented as a commodity in an economical sense. Information is valued from its usefulness, where it allows agents, human or non-human, to take courses of action, e.g. on a stock market. Information is further assumed to have a negligible marginal cost, especially since technology has enabled fast distribution.

2.2.3. Rules and regulations on stock markets

To reduce information asymmetry on the stock market and thus prohibit people from trading based on insider information, there are laws and regulations governing what information a company must disclose to the public and when. In the European Union (EU), all publicly traded companies must oblige by the Market Abuse Regulation (MAR) (European Parliament, 2014). Listed companies are required to publish quarterly and annual reports to the public with material information about the company's progress. These reports include qualitative information (e.g. management outlook, market developments, etc.) and a financial report in the quarterly and annual reports further detailing the quantitative aspects of the firm's performance (e.g. income statement, balance sheet, cash-flow statement). In general, annual reports are more comprehensive compared to quarterly reports. Furthermore, the annual report must be audited by an independent auditor. (Investopedia, 2019a).

Quarterly and annual reports are published at pre-announced dates. However, if the results in the report are significantly different from what the market expects, a profit warning must be issued ahead of the release of the report. Furthermore, the release of a quarterly or annual report is published with a press release containing a summary of the key points in the report. (Investopedia, 2019a, 2019c). In the US, companies also must submit a “10-k” report to the Securities and Exchange Committee with additional information (Kenton, 2019b).

In addition to these scheduled releases of corporate disclosures, companies in the EU must inform the public as soon as possible on any insider information which directly concerns the company. Insider information is information that is not publicly known and could affect investors’ valuation of the company. In some cases, when certain conditions are met, disclosing insider information can be delayed. However, at some point the insider information must be disclosed to the public. (European Parliament, 2014). These are often called ad-hoc press releases or unscheduled releases. In the US they are called 8-K reports and must be filed to the Securities and Exchange Committee (Investopedia, 2019b).

In summary, information is disclosed to the market according to market-specific rules and regulations to provide market actors with equal opportunities to receiving and acting upon novel information relevant for the valuation of the security. Some firm relevant information is released at pre-announced dates while other information is published ad-hoc, i.e. more unexpectedly.

2.2.4. The Efficient Market Hypothesis

The Efficient Market Hypothesis is a theory covering whether prices of securities at any point in time “fully reflect” certain information (Fama 1970). An early fundament for the hypothesis is said to have been laid already in the 16th century by the Italian mathematician Girolamo Cardano, who stated that the most fundamental principle in gambling is equal conditions, e.g. of opponents, money, situation, bystanders and the dice themselves (Sewell, 2011). Fama (1991) describes the hypothesis as a simple statement that helps studies in the area to progress. By using the theory’s assumptions about markets when conducting research on e.g. market inefficiencies, researchers are provided with a benchmark to sidestep the difficult problem of deciding what for example trading costs are, as they cannot be constant (Fama, 1991). Focus can instead be directed on the more interesting task of finding evidence of how stock prices are adjusted to different kinds of information. Implying that the Efficient Market Hypothesis strictly speaking is most certainly wrong may be worrying, but the hypothesis can be considered as profoundly true in spirit, why it is one of the strongest hypotheses in social sciences (Sewell, 2011). This is especially true from the viewpoint that science seeks the best hypothesis, why criticism is of limited value until the flawed hypothesis gets replaced by a better one (Sewell, 2011).

As a general review on market dynamics, Fama (1970) refers to the notion of capital markets as being “fair game” where returns above what could be expected from a “buy-and-hold” trade strategy is not possible on an efficient market. Fama (1970) describes that a market could be categorized as efficient if a select number of investors (market agents) are unable to consistently make better evaluations of available information than what is implicit in market prices.

In summary, the Efficient Market Hypothesis focuses on whether available information is fully reflected in security prices. It also theorizes how markets ‘should’ work in terms of ‘fair game’. The hypothesis is simple and useful in various financial studies, even though some preconditions are not entirely correct in practice.

2.2.5. Testing the Efficient Market Hypothesis

Ever since the introduction of the Efficient Market Hypothesis, researchers have tried to prove the hypothesis wrong in different ways by presenting evidence of various market inefficiencies. Finding evidence of market inefficiencies improves financial theory through an increased understanding of stock market behavior. Such evidence also provides potential strategies for generating abnormal returns, i.e. returns above what could be expected on an efficient market (dos Santos Pinheiro & Dras, 2017a). It is important to note that any evidence of a market inefficiency inherently is biased regarding how the financial model used for measuring the inefficiency defines "properly/normally" priced securities, i.e. how well the model performs (Fama, 1991). This bias results in that the evidence, i.e. exactly how inefficient a market is observed to be in the specific study, is be split between the chosen model's weakness and the actual market inefficiency (Fama, 1991). Fama (1991) however argues that studies on market inefficiencies nonetheless have improved our understanding of market behavior and that the research is among the most successful in empirical economics, with great prospects to remain so in the future.

Fama (1970) categorizes testing of market efficiency into three forms, "weak", "semi-strong" and "strong". The forms help illustrate assumptions of how efficient a stock market is. On a market that proves robust to the weak test, investors should not be able to systematically generate abnormal returns by exploiting *historical* information, i.e. by following "price patterns" (Fama, 1970). On a market that proves robust to the semi-strong test, investors should not be able to systematically generate abnormal returns by exploiting *public* information, thus comprising both historical and new information (Fama 1970). On a market that proves robust to the strong test, an investor should not be able to systematically generate abnormal returns by exploiting *any* information, i.e. both historic and new public- and insider information (Fama, 1970).

Available studies at the time of Fama (1970) indicate that the contemporary American financial market is in the semi-strong form, since there is proof that a select few investors have access to some monopolistic/insider information that allows them to consistently generate abnormal returns. Despite this, Fama (1970) concludes that the Efficient Market Hypothesis is a good approximation to reality for most investors.

The three test forms of market efficiency are briefly summarized in Table 1.

Table 1 - **Efficient Market Hypothesis**

Strong test form	**Semi-strong test form**	**Weak test form**
All new and historic public and insider information is reflected in the price.	All new and historic public information is reflected in the price.	Only historic information is reflected in the price.

Table 1 depicts the three test forms of the Efficient Market Hypothesis and their related assumptions.

In summary, testing the Efficient Market Hypothesis is something researchers have done for a long time and the area has good foresights to continue to be relevant. There are some obvious challenges when testing the Efficient Market Hypothesis since the financial model used for measuring the inefficiency inevitably affects the results. To generate abnormal returns, exploring market inefficiencies is central to be able to possibly take advantage of them. The semi-strong market form is the most tested form of market efficiency since it is a good approximation for most markets. As this study is focusing on ad-hoc disclosures that are public information, we test the semi-strong regarding how fast published disclosures are reflected in stock prices. The speed of which a market fully reflect new information is relevant for the simulation of a trading strategy in this study.

2.2.6. Evidence of market inefficiencies in specific informational events

All studies presented in this sub-section test the Efficient Market Hypothesis by investigating informational events and all test the semi-strong form of the Efficient Market Hypothesis. One of the most tested anomalies to the Efficient Market Hypothesis is the so-called Post Earnings Announcement Drift (PEAD), where abnormal returns have been observed up to twelve months after an earnings announcement highly above or below expectations. This anomaly was first discovered in 1968 (Ball & Brown, 2013) and the results has since then been replicated several times by studies on different markets (Dische, 2002; Hew et al., 1996; Liu et al., 2003). Another anomaly to the Efficient

Market Hypothesis that several studies have observed is that long-term abnormal returns can be observed for firms announcing stock repurchase programs. (Chan et al., 2004; Ikenberry et al., 1995; Peyer & Vermaelen, 2009).

Critics to studies illustrating such anomalies argue that the results are mainly due to flaws in the method of measuring abnormal returns, e.g. estimation of firm specific risk, issues with the data and/or the absence of a theory of market inefficiency as a null-hypothesis (Kothari, 2001). Additionally, research shows that discovered anomalies tend to disappear as investors use them as trading strategies, thus closing the gap to the anomaly (Chordia et al., 2009). The presented studies follow similar patterns where the different informational events initially, i.e. in the short-term, are followed by either an over- or underreaction, which in turn are followed by price reversals in the long term. Sewell (2011) mentions that short-term overreactions (Note; only overreactions) are common to many positively signaling news, why long-term reversal effects may be observed when markets realize its past errors.

In summary, this specific evidence of market inefficiencies shows what research has found both recently and historically, thereby providing a broader picture of studies testing the Efficient Market Hypothesis. In the short term, over- or underreactions can be relatively common in specific cases. In some cases, positive overreactions can be generalized to having long-term reversal effects.

2.2.7. Evidence of market inefficiencies - Time lag

The previously mentioned studies focus on anomalies from the perspective of correctitude, i.e., that a market seems to over- or underreact to a certain, defined, informational event with a specified time horizon. In such studies, the announcements often have a predictable component with an anticipated piece of information (Patel et al., 2003). Another type of market efficiency test is information immediacy, i.e. how soon the information of an event is reflected in the market. The Efficient Market

Hypothesis assumes that all new information is reflected in the market instantly, i.e. as soon as it is published. Hence, such research tests if a time-lag can be observed (Patel et al., 2003).

In this study, market reaction time from unexpected, ad-hoc, information is especially of interest. 8-K financial reports, i.e. ad-hoc disclosures from companies, are good examples of unexpected information. An ad-hoc (corporate) disclosure is in its ideal form information that is released unexpectedly to the market, why no abnormal price effects should be observed before the event day, as compared to pre-scheduled corporate disclosures such as quarterly or annual reports (Bank & Baumann, 2015). The content of an ad-hoc disclosure is information that is mandatory to release by law, such as events and/or changes in a company that could be of interest for shareholders and investors (Kim et al., 2018).

In this study, which aims at creating a machine learning model that can predict stock price movements based on the release of information from companies, it is relevant to understand how fast the market incorporates newly released information. This includes, but is not limited to, an abnormal returns perspective, i.e. how profitable such an algorithm could be when used as a trading tool for overperforming the stock market by being fast at processing new information.

2.2.8. Evidence of market time lags to ad-hoc disclosures

Groth & Muntermann (2011) study changes in intraday stock prices and trading volumes caused by ad-hoc disclosures on the German market. They find that after an ad-hoc disclosure, it takes on average 30 minutes for the price to fully reflect the new information. Muntermann & Guettler (2007) find no evidence for any abnormal activity regarding price or trading volume prior to ad-hoc disclosures. However, the larger the company that announces an ad-hoc disclosure, the less abnormal return and trading volume effect is observed immediately after the disclosure (Muntermann &

Guettler, 2007). Also, the higher the price and trading volumes are on the day before the disclosure, the higher they both are after the disclosure (Muntermann & Guettler, 2007).

Bank & Baumann (2015) also study ad-hoc disclosures on the German market. Instead of focusing only on intra-day returns and market behavior, Bank & Baumann (2015) conduct an event study that similarly to Muntermann & Guettler (2007) examines the day prior to and the day after the release of new, ad-hoc, information. One of the study's most important findings is that released information related to periodic reports seems to be incorporated into stock prices one day prior to being released to the market (Bank & Baumann, 2015). Such disclosures can be called ad-hoc because the exact date/time of the release is unexpected, but it still seems to have been priced into the security prior to the release as the disclosure relate to some pre announced report (Bank & Baumann, 2015). This is also in line with the findings of Baule & Tallau (2012).

The results of Bank & Baumann (2015) point in the same direction as those of Muntermann & Guettler (2007), since they find that stock prices often continue to adjust several days after an ad-hoc disclosure. Bank & Baumann (2015) conclude that the event day reactions can be explained by four different factors, which also nuance the results of Muntermann & Guettler (2007). These factors are index affiliation, market uncertainty, disclosure periodicity and the informativeness of the disclosure.

In summary, evidence for market time-lags in relation to ad-hoc disclosures seems to exist. Time-lags seem to vary from intra-day effects to price adjustments several days after the ad-hoc announcement. However, ad-hoc disclosures related to periodical reports seem to not have any time-lag in the incorporation of prices. This information gives this thesis reason to suspect that a successful algorithm could generate abnormal returns through being faster than the market when classifying ad-hoc disclosures not related to periodical reports.

2.2.9. Information processing costs as an explanation for time lag

Engelberg (2008) emphasizes individuals' (investors') information processing costs as a possible explanation to market anomalies. As information is not homogenous in type, he reasons that some news should be more easily deciphered and thereby quickly incorporated into market prices, whereas other information is more costly to process and therefore incorporated over time.

Engelberg (2008) uses the example of a financial statement versus a transcript of a conference call to illustrate information costs. The financial statement is largely made up of numbers organized in a standardized fashion so that an individual can process it effectively and efficiently. It is often easily created, summarized, stored, transmitted, and compared with financial statements of other firms. In contrast, the text of a conference call may be more difficult for an individual to process. It may take a sophisticated understanding of language, tone, or nuance. Also, a summary of its content may be more difficult to create, more subjective, less comparable across firms, and more difficult to transmit. (Engelberg, 2008)

To structure his reasoning, Engelberg (2008) uses the concepts of hard and soft information, where hard information is less costly and soft information is more costly to process. Liberti & Petersen (2018) recommend that, rather than classifying information as either hard or soft, one should think of the classification in terms of a continuum. Examples of information in the financial domain that could be classified as hard are financial statements, credit scores, stock returns and production output. With hard information, the context in which the information was collected is unimportant. If hard information is transmitted from the collector to e.g. a decision maker, no information is lost in the process; the meaning of the information only depends on the information itself. Due to these traits, hard information is almost always stored as numbers (Liberti & Petersen, 2018).

Examples of soft information in the financial domain are opinions, ideas, rumors, economic projections, statements of management's foresights, and market commentary (Liberti & Petersen, 2018). With soft information, context is important since the information might be a consequence of subjectivity or environmental factors when assembling the data. For example, one can always create a numerical score from soft information, e.g. an index reflecting how honest a potential borrower is. A numerical index in and of itself does however not make the information hard since the interpretation of honesty might be based on an individual's opinion rather than numerical data (Liberti & Petersen, 2018). Due to the traits of soft information, it is often stored in text format (Liberti & Petersen, 2018).

In summary, information processing costs deal with how market agents may have different costs when processing new information. The different processing costs can be put in a context of soft vs hard information. Soft information, which often comes in the form of text, inherits a higher processing cost compared to hard(er) information, which often comes in numerical form. Put in the context of this thesis, ad-hoc disclosures could take a longer time to (1) read and (2) correctly understand compared to harder information often found in e.g. financial statements. Also, some ad-hoc disclosures might be more difficult/take longer time for individuals (investors) to process than others, thus creating a time lag until the information is reflected on the market.

2.2.10. Summary of financial theory

This theory section focuses on defining and discussing financial markets and stock related concepts to provide a broad understanding of the financial domain with its rules, theories, and actors. The intention with this is to present the playing field for the area in which an algorithm that is built on market relevant information would play, since the business intention of the algorithm is to compete with other agents on the stock market.

Initially, information is presented as being the raw material of financial markets, where information can be considered as a commodity with low to no cost of reproduction. After describing how a stock market works with its rules and regulations, the description of the Efficient Market Hypothesis provides the reader with an understanding of how markets function in theory for reality approximation, i.e. how markets are assumed to reflect information in security prices. Subsequently, research depicting market inefficiencies is presented, thus helping the reader to understand market behavior in practice, which helps with revealing potential ways of generating abnormal returns. First, evidence for inefficient markets relating to specific informational events is presented. Second, market inefficiencies that relate to general time lags for ad-hoc information is presented.

For this study, the relevance of proven market inefficiencies lies in the indications that a successful algorithm could beat the market through correctly predicting the direction in which a stock price will move. This mainly regards to the studies proving that there is a time lag before ad-hoc disclosures are fully incorporated into the market. This indicates that, if an algorithm is successfully trained at classifying ad-hoc disclosures' impact on stock prices, the algorithm might be able to take advantage of the market's time lag, thus systematically beating the market. As a potential explanation for how time lags can exist on efficient markets, the concept information processing costs is explained. Information processing describes how soft information, e.g. text, is more costly to process for investors than hard(er) information, e.g. numbers.

With this financial introduction, the study continues to the more technical challenge of building an algorithm based on corporate textual ad-hoc disclosures to predict stock price movements to subsequently being part of a trading strategy.

2.3. Automated trading strategies

On the stock market, there could potentially exist as many trading strategies as there are investors. Trading can however be divided into different categories of methods. In this study, trading strategies that are automatized, i.e. based on one or more informational events, and involve algorithms, are of interest.

2.3.1. Event Based Trading

A method and system for trading based on news to the stock market is called Event Based Trading (O'Connor, 2009). Traders can pre-define their strategy of trading through including one or more trading rules based on comparisons of one or several event values that have been estimated (O'Connor, 2009). When novel event values, i.e. news, are released to the market, the values can be used through user input or directly as input from other outside sources to make it possible for the pre-defined strategy to be executed (O'Connor, 2009).

The popularity of Event Based Trading has grown as the trend of electronic trading has become well established and the matching of bids and offers, among other things, are today automatic (O'Connor, 2009). Traders are connected to electronic trading platform interfaces created by the stock exchanges and newly released information regarding securities is communicated in real time (O'Connor, 2009). To be able to profit in rapidly moving markets, it is important to be able to assimilate enormous quantities of data and react quickly. It is therefore beneficial if trading platforms offer tools that assist traders in executing their strategies fast and accurately (O'Connor, 2009). One particularly important feature of trading tools is that they are providing traders with the possibility to effectively use and monitor news announcements from companies and the media (O'Connor, 2009).

Figure 1 - **Example of a news trader application**

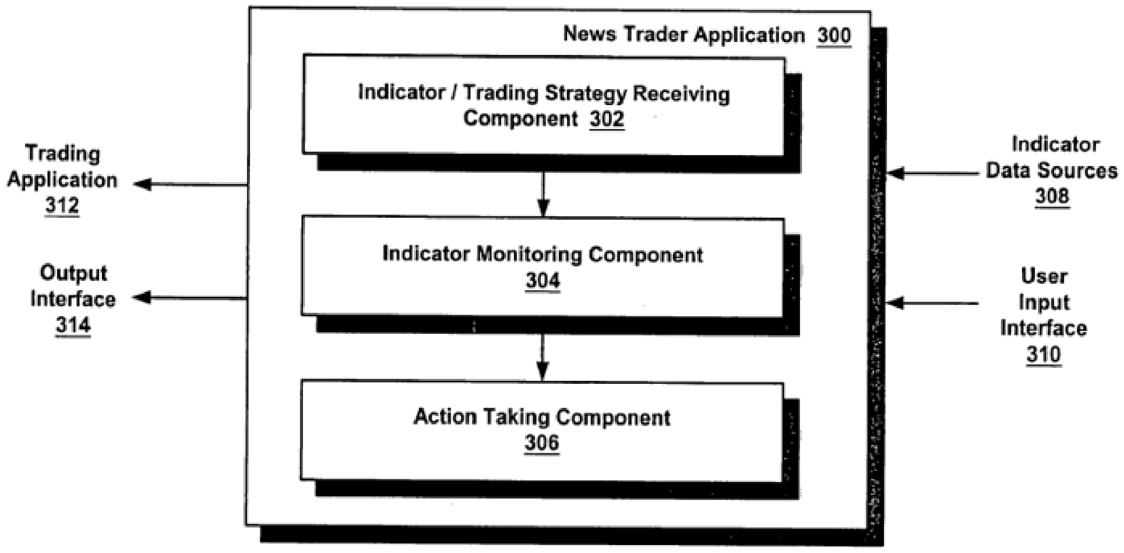

News Trader Application 300
Indicator / Trading Strategy Receiving Component 302
Indicator Monitoring Component 304
Action Taking Component 306
Trading Application 312
Output Interface 314
Indicator Data Sources 308
User Input Interface 310

Figure 1 depicts a conceptual overview of how an Event Based Trading application can be set up given inputs of news relevant for the valuation of a stock, and how those inputs can lead to action outputs with related interfaces.

Figure 2 - **Flow chart from input to output of an Event Based Trading strategy**

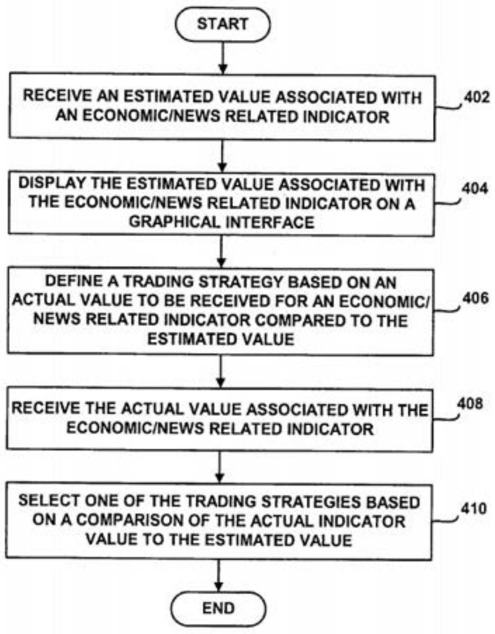

START
RECEIVE AN ESTIMATED VALUE ASSOCIATED WITH AN ECONOMIC/NEWS RELATED INDICATOR
402
DISPLAY THE ESTIMATED VALUE ASSOCIATED WITH THE ECONOMIC/NEWS RELATED INDICATOR ON A GRAPHICAL INTERFACE
404
DEFINE A TRADING STRATEGY BASED ON AN ACTUAL VALUE TO BE RECEIVED FOR AN ECONOMIC/ NEWS RELATED INDICATOR COMPARED TO THE ESTIMATED VALUE
406
RECEIVE THE ACTUAL VALUE ASSOCIATED WITH THE ECONOMIC/NEWS RELATED INDICATOR
408
SELECT ONE OF THE TRADING STRATEGIES BASED ON A COMPARISON OF THE ACTUAL INDICATOR VALUE TO THE ESTIMATED VALUE
410
END

Figure 2 depicts a flow chart that more specifically than Figure 1 goes through relevant practical steps that an Event Based Trading system in general follow before executing actions on the stock market.

Events are in Event Based Trading defined as any news-related indicator. As seen in Figure 1, an example method for implementing algorithmic trading based on news is presented. Its segments are related to the components in Figure 2. The Figure 2 flow chart shows a possible implementation of the operation and functionality of Figure 1 where the segments can be implemented in different ways to execute a certain strategy (O'Connor, 2009). The news that forms the basis for execution can come from any source that the trader selects, as for instance corporate ad-hoc disclosures that are released directly from companies, and in general also to trading platforms (O'Connor, 2009).

In summary, traders use certain informational events as inputs with pre-defined trading strategies to rapidly execute the strategy on the stock market. As stock markets today are highly digital and automated, it is important for traders to be able to assimilate large quantities of data, why the tools that stock exchanges offer are of importance for Event Based Trading. Among the most important tools are those that provide traders with the ability to monitor company news.

2.3.2. Algorithmic trading

Related to Event Based Trading is Algorithmic Trading, which more specifically describes the utilization of computers and algorithms that execute trading strategies. In recent years, with advances in computational power and in the ability of computers to process massive amounts of data, Algorithmic Trading has emerged as a strong trend in investment management (Ruta, 2014). Algorithmic Trading refers to a system that has automated one or more stages of the trading process in electronic financial markets with the help of computers (Nuti et al., 2011). The steps that are most automated are trade execution, recommendations on whether to buy or sell stocks, and pre-trade analysis. It can thus be made with more or less intervention of humans (Nuti et al., 2011). Algorithmic Trading usually involves dynamic planning, learning, reasoning, and decision making (Treleaven et al. 2013).

Algorithmic Trading comes with big research challenges, partly because missteps can result in grand economic consequences. One example of this is on May 6th, 2010 when erratically behaving trading algorithms resulted in the Dow Jones industrial average index plunging 9 % in one day, thus wiping out 600 million dollars of value (Treleaven et al. 2013). Algorithmic Trading has a key challenge regarding the quality and quantity of data it acts upon, especially as textual data becomes increasingly relevant in the field (Treleaven et al. 2013).

Algorithmic Trading has several system components that can be linked to some of the components in Event Based Trading. The first step is data access and cleaning, where collection of data and pre-processing is necessary for driving the Algorithmic Trading. The second step is to analyze the properties of the securities to find trading opportunities through using market data such as financial news. This is called pre-trade analysis. The third step is to generate the trading signal, which comprises “what” to trade and “when”. The fourth step is the execution of trading for the selected assets, i.e. “how”. The fifth and last step is the post-trade analysis, which looks at differences between prices on when the buy/sell orders were placed. Algorithmic Trading may be understood a system that is almost fully automated, but it is important to note that much of the effort that enables Algorithmic Trading is devoted to accessing data and cleaning it for the pre-trade stages. The latter stages are most often monitored closely by humans, even if actions are automated. (Nuti et al., 2011)

Algorithmic Trading can be divided into two basic strategy approaches: theory- and empirical-driven. The theory-driven approach assumes that the programmer or researcher chooses a hypothesis for what way securities are likely to behave, which is used for the modeling that tries to confirm the hypothesis (Nuti et al., 2011). The empirical-driven approach represents how the researcher with the help of an algorithm mines the data to identify patterns, without any hypotheses about the security’s volatility and behavior (Nuti et al., 2011). This study hypothesizes regarding assumptions of stock price behavior, why it rather belongs to the theory-driven approach.

In summary, Algorithmic Trading builds upon the framework of Event Based Trading through more specifically focusing on how one or more algorithms interact. Algorithmic Trading refers to any programmed trading system that automates one or more steps of the trading process, which is divided into several steps. Algorithmic Trading can have great impact on stock price movements, and text data is becoming increasingly relevant in the field. It is worth noting that even if some steps are automated, collecting and cleaning data takes a lot of effort. Algorithmic Trading can be divided into two different approaches, where the theory-driven approach mostly concurs with this study's approach of hypothesizing regarding security price behavior after ad-hoc disclosures are published.

2.4. Artificial Intelligence and Machine Learning

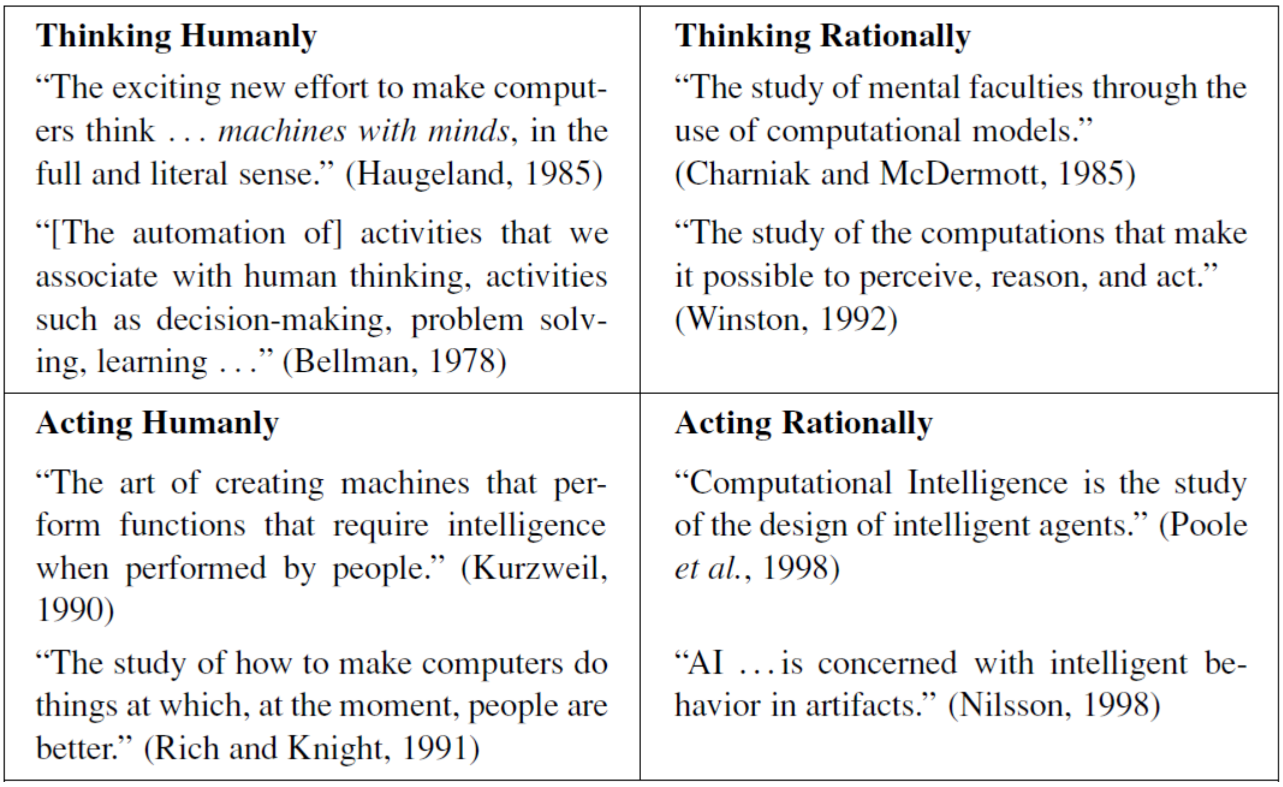
The concepts of Artificial Intelligence and Machine Learning are presented to provide background and context to this specific study and its relation to the two concepts. Since the concepts individually and together can be defined quite differently, some clarification helps putting this study in relation to both big domains.

2.4.1. Artificial Intelligence (AI)

Artificial intelligence is a broad area that is related to this study in the sense that a smart algorithm that trades on the stock market somewhat mimics human behavior through aiming to make intelligent decisions. Russell & Norvig (2010) present several definitions of Artificial Intelligence, dividing them into four categories. Historically, different scientists have followed different definitions, and the definitions have disparaged and helped each other in the development of Artificial Intelligence as a scientific area (Russell & Norvig, 2010).

Figure 3 is a matrix depicting the four categories on two dimensions. The definitions on top concern *thought processes* and *reasoning* whereas the ones on the bottom address *behavior*. The definitions to the left comprise *human* performance, whereas those to the right regard *ideal* performance, i.e. rationality.

Figure 3 - **Eight definitions of Artificial Intelligence depicted in four categories (Russel & Norvig, 2010)**



Thinking Humanly	**Thinking Rationally**
"The exciting new effort to make computers think . . . *machines with minds*, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . ." (Bellman, 1978)	"The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly	**Acting Rationally**
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998) "AI . . . is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

Figure 3 depicts eight definitions of AI which can be divided into four categories. These represent different approaches and assumptions of AI relevant e.g. for which area research is conducted and what the goal of an AI application has.

Russell & Norvig (2010) argue that rationality is more amenable to scientific development of Artificial Intelligence than approaches based on human behavior or human thought, because rationality is mathematically well defined and can be "unpacked" to generate agents that can achieve it. In comparison, human behavior is well adapted to the environment humans live in and can be defined as the sum of all the things that humans do but is difficult to mimic. Furthermore, Russell & Norvig (2010) implicitly assume that action is central in the notion of Artificial Intelligence. Since rational action (behavior) is not always a consequence of rational thinking (inference), it is therefore irrelevant and often incorrect to define Artificial Intelligence in terms of a system that is "thinking rationally”. Russell & Norvig (2010) define a rational system as “a system that does the ‘right thing’, given what it knows”. Hence, Russell & Norvig (2010) support ideality over humanity, and behavior over thought processes, in terms of Artificial Intelligence.

In summary, Russell & Norvig (2010) argue that evaluating, thus defining, Artificial Intelligence in terms of something that is “acting rationally”, i.e. a rational agent, is more important than other definitions, for example those depicted in Figure 3. Connecting this discussion to an artificially intelligent agent trading on the stock market, the focus when building the agent is that it is *behaving ideally* regarding the goal of generating economic value.

2.4.2. Machine Learning (ML)

One trait often associated with “intelligence” is the ability of an agent to adapt to its environment, which in turn requires the capability to *learn* (Russell & Norvig, 2010). Russell & Norvig (2010) define a learning agent as an agent that “improves its performance on future tasks after making observations about the world”. *Machine Learning* can be viewed as the science addressing the learning capabilities of artificial systems. Even though Machine Learning is an important part of Artificial Intelligence, it is also a stand-alone science. It is a scientific field in the intersection of statistics, artificial intelligence and computer science, which can also be called “predictive analytics” or “statistical learning” (Müller & Guido 2016). In summary, “Machine learning is about extracting knowledge from data” (Müller & Guido, 2016).

Depending on the task at hand, different Machine Learning methods are suitable. These methods can be divided into three “types of feedback” categories (Russell & Norvig, 2010). *Unsupervised learning* methods are focused on extracting knowledge from data where there is no available feedback, i.e. no known output (Müller & Guido, 2016). In practice, these regards finding commonalities/patterns in the data set (Müller & Guido, 2016). Such commonalities can for example be used for finding preferences within a population, such as “25-year old men like chocolate flavored ice cream”, or for labeling unlabeled instances, such as generating a topic of a group of text document.

Reinforcement learning methods are concerned with extracting knowledge based on feedback from a series of reinforcements – rewards or punishments (Russell & Norvig, 2010). For example, an artificial agent responsible for recommending “movies you will enjoy” on a video streaming service receives feedback based on if the user chooses to watch a recommended movie or not. An agent might also learn which chess movements are successful based on how the opponent subsequently moves his pieces.

Supervised learning methods are concerned with extracting knowledge from data based on feedback from known output (Müller & Guido, 2016). For example, the system might be able to successfully classify cancer tumors as benign or malignant from learning about the traits of known output, i.e. tumors that have historically been benign or malignant. More relevant for this study, a system might also be able to predict the correct price of a stock after the disclosure of some information, given the knowledge about historical output, i.e. how similar information historically has affected the stock price.

In summary, Machine Learning considers artificial systems that learn from their environments in order to adapt to them. There are different Machine Learning methods suitable for different kinds of problems. Stock price prediction based on known historical price movements fits into the supervised learning methodology.

2.5. Natural Language Processing (NLP)

A concept related to Artificial Intelligence and Machine Learning is Natural Language Processing, which closely concerns this study. Natural Language Processing is the study and facilitation of communication between computers and people (Fisher et al., 2016). Natural Language Processing involves a variety of computational techniques with the goal of enabling computers to process human language from text (Fisher et al., 2016). More specifically, Natural Language Processing involves computational techniques that analyze and represent naturally occurring text at different levels of linguistical analysis for approaching various tasks and applications. It therefore includes both a set of theories and a set of technologies (Liddy, 2001). “Naturally occurring texts” can be in any area and language and it can be both written and oral if the language is inherently human.

2.5.1. Text mining

The process of facilitating communication between computers and people results in the inevitable challenge of dealing with text data. Text is a unique sort of data that has become common to analyze with the help of computers. This can be explained by how internet has become a very powerful source of textual information (Fawcett & Provost, 2013). Examining and analyzing, i.e. mining, text data can provide us with a better understanding text, which is a very important type of data that computers are not naturally proficient in seeing patterns in, compared to e.g. solely numerical problems that computers inherently are created to solve (Fawcett & Provost, 2013). Text is regularly referred to as “unstructured” and “dirty” data, which has implications on how it should be handled in order to make relevant inferences from mining it (Fawcett & Provost, 2013). Hence, mining text is a key aspect within Natural Language Processing. In the method section, different text pre-processing techniques for text data are discussed.

2.5.2. Natural Language Processing in the financial sector

Natural Language Processing in the financial sector has gained attention lately. Text documents within finance are of general interest since such documents are intended to communicate firm and market relevant information, including e.g. financial performance, management’s assessment of current and future performance and evidence of compliance with regulations (Fisher et al., 2016).

Businesses have through the growth in digital and social media increased the number of unstructured text documents, why Natural Language Processing further has potential to enhance its usefulness within finance (Fisher et al., 2016).

Natural Language Processing applications aim to mine such documents to reach a deeper understanding and make inferences, thus enhancing knowledge about text documents and their impact in respective field (Fisher et al., 2016). Within finance and stock market research, relevant text documents have a key advantage compared to number-based measurements since they can provide a potentially more independent way of challenging the Efficient Market Hypothesis (Kearney & Liu, 2014). Text data and Natural Language Processing can be used in various ways within the financial sector to be examined and learned from.

2.6. Natural Language Processing and stock price prediction

Several research studies have applied Natural Language Processing in the financial field to classify and learn from different kinds of text data for the purpose of predicting stock price movements. It is now well established that textual cues convey price-relevant information while it is a challenging research setup (Feuerriegel & Gordon, 2018). Older approaches use rather structured data while text is inherently, at least from a computer's perspective, unstructured. The research advancements in the field enable increasingly meaningful analyses of text information, which is a source of information that is very accessible (Groth & Muntermann, 2011).

In this section, studies using mainly ad-hoc disclosures are discussed together with a shorter section of what other types of text data that also can be used for the same purposes of predicting stock prices with machine learning. All studies within the area are summarized in Table 2.

2.6.1. Predicting stock prices with corporate disclosures

Corporate disclosures contain stock price relevant information like for instance quarterly earnings, management changes, risks and other important events (Feuerriegel & Gordon, 2018). Based on the premise of a semi-efficient market that reflects all publicly available information, whenever new information hits the market, one can expect stock prices to change (Feuerriegel & Gordon, 2018). Corporate disclosures are regulated in order to be released with equal access for all market participants, why they represent a potentially financially rewarding means of predicting and forecasting stock price movements (Feuerriegel & Gordon, 2018). The objectiveness of corporate disclosures as well as their market relevance, together with an inherently concise format and short publication times, add to the advantages of using textual disclosures for predicting stock prices (Feuerriegel & Gordon, 2018). Besides ad-hoc disclosures, other corporate disclosures that are interesting are annual, semi-annual and quarterly reports that have interesting sections that can be used in prediction tasks, similar to ad-hoc disclosures (Balakrishnan et al., 2010)

2.6.1.1. Kim et al. (2018) – ad-hoc disclosures

Kim et al. (2018) predict the direction of stock price movements based on 8-K filings, i.e. ad-hoc disclosures for four companies in the Unites States. The study predicts upward and downward movements by performing a classification task with text documents while separating firms based on sectors, since similar words across sectors can convey different sentiments (Kim et al., 2018). The study assumes that on the day after the announcement of the 8-K, the information has the greatest impact on the stock price (Kim et al., 2018). By using distributed representations, meaning that documents are embedded with class information in a multi-dimensional space, the identification of the sentiment class of a given document is possible by computing the relative distance of the words in the document to words already learned by the model (Kim et al., 2018).

Since documents in financial markets are not independent, meaning that they can impact stock prices for different amounts of time and relate to each other, Kim et al. (2018) visualize the reports over time to confirm the relationship between the documents and the stock price movements. This approach leads to a prediction performance of 25,4 % (Lift) over the baseline model. The evidence suggests that the direction of the stock price after a disclosure shifts accordingly with the polarity (positive, neutral or negative) of the sentiment of the reports (Kim et al., 2018). Kim et al. (2018) show that positive news cause stock prices to rise relatively fast while negative news affect stock prices with a somewhat delayed reaction. Furthermore, the study also provides a framework for traders through the visualization of sentiments in the ad-hoc disclosures to enable a method of making split-second data-informed decisions more easily (Kim et al., 2018).

2.6.1.2. Feuerriegel & Gordon (2018) - Regulatory disclosures (ad-hoc disclosures)

Feuerriegel & Gordon (2018) predict stock prices based on regulatory disclosures, i.e. ad-hoc disclosures that are regulated by law regarding how and when they must be released. Feuerriegel & Gordon (2018) focus on both short- and long-term stock price forecasts and use a dataset of about 75 000 disclosures on the German market. The study uses predictive models suitable for high-dimensional data and uses data- and knowledge-driven techniques to avoid overfitting (Feuerriegel & Gordon, 2018). This approach is especially suitable for when predicting stocks with the purpose of decreasing forecast errors for different business- and trading applications in financial markets (Feuerriegel & Gordon, 2018).

The information acquired from ad-hoc disclosures often convey broad spectrums of value-relevance, which is interesting from multiple perspectives, e.g. for analyzing past performance, current performance and outlook (Feuerriegel & Gordon, 2018). Researchers have demonstrated the predictive capabilities of disclosures regarding individual stock market returns in the short-term, why corporate ad-hoc disclosures have gained relevance. This has resulted in much new research within

Natural Language Processing and forecasting algorithms (Feuerriegel & Gordon, 2018). Feuerriegel & Gordon (2018) build upon multiple studies on the subject and add the long-term return layer to contemporary research, as the evidence of long-term predictive power of ad-hoc disclosures is scarce.

Based on the findings in the study, algorithmic trading systems concerning text-based stock price prediction, especially with the use of corporate ad-hoc disclosures, are evidently capable of executing profitable strategies of trading (Feuerriegel & Gordon, 2018). Feuerriegel & Gordon (2018) describe the main advantage of the forecasting of corporate disclosures to be that it enables detection of market movements that are too complex to detect for humans. Such forecast models manage to statistically significantly reduce prediction errors below baseline predictions based on historic lagged data, which confirms the relevance of the approach of text mining and price predicting with corporate ad-hoc disclosures (Feuerriegel & Gordon, 2018).

2.6.1.3. Rekabsaz et al. (2017) – Risk factor section in 10-K reports

Rekabsaz et al. (2017) predict stock price volatility with a sentiment analysis on corporate disclosures. The specific corporate disclosures of interest in the study are companies' annual disclosures, known as 10-K filings. They include comprehensive summaries of the company's business and risk factors (Rekabsaz et al., 2017). The section 1A in the 10-K report, called *Risk Factors*, is especially of interest since it covers the most important and significant risk for the firm and is also easier to process compared to processing the full document (Rekabsaz et al., 2017). Using this specific part of the 10-K disclosure is motivated further to mitigate that 10-K disclosures are getting increasingly complex and redundant. For example, a reader is required to have on average 21,6 years of formal education to be able to fully comprehend the document (Rekabsaz et al., 2017). Over the years, as there has been an increase of the length of the 10-K documents as well as an increase in the number of topics, Dyer et al. (2016) conclude that the risk factor topic seems to be the one topic with actual informativeness for investors, which makes it relevance for further text analysis for stock price prediction.

The 10-K reports average at about 5 000 words and seem to structurally change average content across firms in cycles of three to four years (Rekabsaz et al., 2017). To analyze the documents and the risk factor section, Rekabsaz et al. (2017) use state-of-the-art Information Retrieval term weighting models that utilize word embedding information and which have a significant impact on prediction accuracy. The weights of the words in the model are calculated by extending them to similar terms in the training documents as a basis for the sentiment analysis (Rekabsaz et al., 2017). Rekabsaz et al. (2017) further analyze sentiments of the documents sector-wise, because it is assumed that factors of uncertainty and instability are similar within sectors but different between sectors.

The results indicate that risk factors seem to be shared within sectors, even though models trained sector wise do not generate more accuracy compared to models run over multiple sectors (Rekabsaz et al., 2017). The volatility prediction as a continuous estimation reaches an r^2 of 0,527 which beats earlier similar studies (Rekabsaz et al., 2017).

2.6.1.4. Groth & Muntermann (2011), Muntermann & Guettler (2007) – ad-hoc disclosures

Groth & Muntermann, (2011) build upon the study of Muntermann & Guettler (2007) and use ad-hoc disclosures and Natural Language Processing as method to manage financial risk on the German stock market. Specifically, Groth & Muntermann (2011) explore the implications of risk that information newly available to market induces. New information can inherently be expected and has been proven to significantly drive stock price volatilities, why Groth & Muntermann (2011) try to identify among the text data which ad-hoc disclosures that have resulted in the most risk exposure.

Groth & Muntermann (2011) use Naïve Bayes, SVM, Neural Network and k-nearest neighbor to predict whether a specific ad-hoc disclosure changes the intraday risk profile, i.e. increases volatility, meaning that the stock price moves significantly due to the new information provided in the ad-hoc disclosure. Like Kim et al. (2018), 8-K reports are used for the training the algorithm. Groth & Muntermann (2011) evaluate their algorithm by using both a regular data mining baseline, and through a newly developed simulation-based evaluation which empirically tests the algorithm on the stock market to draw conclusions about the suitability of the text mining approach.

Groth & Muntermann (2011) find strong evidence of that unstructured textual data and machine learning techniques can be a valuable source of information for risk management in the financial sector. Furthermore, they find significant differences in performance between the techniques applied. Groth & Muntermann (2011) underline that supervised text data learning is suitable both from a "classic" learning evaluation perspective, and when using the algorithm as a simulation on the market which shows that intraday market risk exposures can be discovered trough using text mining techniques.

2.6.1.5. Balakrishnan et al. (2010) - Narrative Disclosure

Balakrishnan et al. (2010) investigate narrative disclosures in 10-K filings to see if they contain information value-relevant to predicting stock prices. Narratives are an important information source in the 10-K reports with management discussions often seen as a very important item regarding valuation (Balakrishnan et al., 2010). Since most earlier studies mostly analyze numerical data from the 10-K disclosures to predict stock prices because of the relatively higher cost of processing text data vs numerical, there is reason to believe that the information in the narrative disclosures is in general not fully reflected in stock prices at the time of publication (Balakrishnan et al., 2010).

To build a predictive classification model with the narrative disclosures, the disclosures are paired in training with the subsequent performance where outperforming, under-performing and average performance form the classes (Balakrishnan et al., 2010). After training, the models are tested as a trading strategy where they form a portfolio that is equally weighted to buy predicted over-performers and sell predicted under-performers (Balakrishnan et al., 2010). The returns for the portfolio test how prevalent value relevant information is in the narrative disclosures and the method's ability to extract it systematically (Balakrishnan et al., 2010). When analyzing the results with relevant indicators, the firms that are in a certain predicted class are also quantitatively measured with common financial risk factors/attributes in stock price prediction such as size, book-to-market and momentum (Balakrishnan et al., 2010).

The study finds correlation between the quantitative attributes and the text disclosures which is interesting, but it could also be possible that the text disclosure impacts the association between market performance and the quantitative attributes (Balakrishnan et al., 2010). This would align with the results from the text classification that seems to capture the same returns as the known risk factors that would predict similar stock price performance from the disclosure (Balakrishnan et al., 2010). The disclosure score hence does not seem to provide new information but as the interaction term between the text and the risk factors reliably differs from zero, it indicates that differences in disclosures affect confidence in the numerical estimates (Balakrishnan et al., 2010).

2.6.2. Predicting stock prices with other text data

In this study, corporate disclosures are the text data of interest that will be used when building machine learning models. It is although also worth mentioning the other types of text data that have been used for the same classification task of predicting stock price movements. This is since it provides an understanding of the different kinds of relevant textual documents in the financial field that can have predictive power when it comes to stock prices. These studies are further relevant to discuss since they provide an understanding of other relevant approaches which gives the study a better foundation for analysis of results and for pinpointing strengths/weaknesses using a certain type of text data.

The two other types of text data relevant within the field that will be discussed are financial news data and social media data.

2.6.2.1. Predicting stock prices with financial news

In addition to predicting stock price movements based on corporate disclosures, several research studies have attempted the same but with financial news as the data source. In a comprehensive study, (dos Santos Pinheiro & Dras, 2017a) use financial news headlines from US companies included in the S&P 500 stock index to predict stock price movements using a LSTM neural network with character level embeddings. The study evaluates the model on intraday (one hour after release) movements and intraday (t + 1 day) movements separately. For the intraday evaluation, all news released in one day are concatenated into one document. The results show that the model is competitive with a higher accuracy than most other models previously used on the same dataset, both on intraday and intraday. (dos Santos Pinheiro & Dras, 2017b).

(Hajek & Barushka, 2018) construct a deep neural network based on a combination of sentiment analysis and topic detection of financial news to predict stock price movements. The study collects financial news from 15 US companies listed on the New York Stock Exchange and constructs prediction models separately for each company to consider company specific characteristics. The results show a significant improvement when using a combination of sentiment analysis and topic detection, compared to only using either one of them. (Hajek & Barushka, 2018).

2.6.2.2. Predicting stock prices based on sentiment analysis of social media

In addition to predicting stock price movements through corporate disclosures or financial news, several studies use sentiment analysis of data from social media to predict stock price movements.

Tetlock (2007) used sentiment analysis to analyze the correlation between sentiment in news articles and market prices, concluding that media pessimism may affect both market prices and trading volume. Similarly, Bollen et al. (2011) used a system to measure collective mood through Twitter feeds and showed it to be highly predictive of the Dow Jones Industrial Average closing values. Following these results, other work has also social media information for stock market forecasting.

Xu & Cohen (2018) collect tweets from Twitter mentioning 88 US companies in order to assess the sentiment of investors and combine this data with historical stock price movements to predict future stock price movements. The study introduces StockNet, a deep generative model based on a neural network architecture. The study achieves slightly better results compared to the baseline models. (Xu & Cohen, 2018).

In another study attempting to predict stock price movements, Nguyen & Shirai (2015) gather one-year data from Yahoo Finance Message Board for 5 companies listed in the US. The message boards for each specific company is used by investors to discuss the company and their investments in it. The data is used to for topic detection and sentiment analysis in order to predict stock price movements using a proposed model called Topic Sentiment Latent Dirichlet Allocation (TSLDA). (Nguyen & Shirai, 2015).

2.6.3. Summary of NLP and its applications in the financial field

In summary, NLP involves a variety of computational techniques with the goal of enabling human-like processing of language from text. NLP has been applied to several different tasks and recently gained significant attention within the financial sector as a tool to predict stock price movements. Several studies have succeeded with various results in predicting these movements based on corporate disclosures, financial news or social media. Furthermore, some studies have combined the qualitative data of text with quantitative data such as accounting information and market-based data to further improve the accuracy

2.6.3.1. Overview of studies predicting stock prices with text data

Table 2 - **Summary of related work**

Author	**Text data**	**Scope**	**Feature selection**	**Method**	**Findings**
Kim et al. (2018)	8-K financial reports	4 listed US firms in the financial sector	BoW (TF, TFIDF) with unigram and bigram features	LR, RF, MNB, SVM, NBSVM, SPV	Prediction accuracy over the baseline model (LR/RF) of 24.5%.
Hajek & Barushka (2018)	Financial News	15 US firms listed on NYSE	BoW pre-defined word list (TF) and LDA	DNN	Integration of sentiment analysis and topic detection improves the performance. DNN outperforms the baseline (SVM).
Hájek, (2018)	Annual reports	1402 US firms listed on NYSE or Nasdaq	BoW (TF-IDF)	NN, NB, SVM, C4.5 DT, k-NN	NN performs similar to NB but outperforms the other methods.
Groth & Muntermann (2011)	Corporate disclosures that were published to fulfill regulatory legislation	Germany	BoW (TF-IDF)	NB, k-NN, NN, SVM	kNN, NNet, or SVM perform better than NB.
X. Li et al. (2014)		22 firms in Hong Kong	Pre-defined dictionary, BoW, SenticNet and Sentiment Polarity	SVM	Sentiment analysis slightly outperforms BoW and is major improvement to Sentiment Polarity
Dos Santos Pinheiro & Dras (2017)	Financial news	US firms from S&P 500 index	Character level embeddings	LSTM neural network	The proposed model is competitive with other state-of-the-art models.
Muntermann & Guettler (2007)	Corporate disclosures that were published to fulfill regulatory legislation	Germany		Hypothesis testing. I.e., not model training.	Significant abnormal intraday stock price and trading volume and 30-minute market time lag to ad-hoc disclosures
Quanzhi Li & Shah (2017)	30k messages from StockTwits	US	Lexicon / Sentiment-Oriented Word Embedding	SMO	The lexicon outperformed other lexicons built by the state-of-the-art methods, and the sentiment-oriented word vector performed better than the general word embeddings.
Mai et. al. (2019)	(1) Accounting data from Compustat (2) Equity trading data (3) textual disclosure data from 10-K reports	1,827 U.S. public companies	Word2vec	Deep learning Models	Textual disclosures can predict bankruptcies. Combining textual data with traditional accounting data and market-based data can further improve accuracy. and compares models
Qin & Yang (2019)	Conference calls following financial reports (transcribed and audio)	S&P 500	Pre-trained word embeddings	MDRM	Results show that the model that jointly considers verbal and vocal features achieves significant results
Xu & Cohen (2018)	Tweets from two years	88 companies in the US	BoW	StockNet (NN)	Performs better than strong baselines
Nguyen & Shirai (2015)	>35k documents from Yahoo Finance message board	5 companies in the US		Topic Sentiment Latent Dirichlet Allocation (TSLDA)	
Qing Li et al. (2015)	Financial news, message boards and stock data	Chineese companies	Tensors	Supervised tensor regression learning	The approach outperforms the state-of-the-art trading strategies
Schumaker et al. (2012)	Financial news (2800 articles)	Listed firms in the US	AZFinText	AZFinText	Outperforms the baseline
Balakrishnan et al. (2010)	Narrative disclosure in 10-K reports	4280 filings from 1236 US firms	BoW, TF	Disclosure score , Multiple regression	The disclosure score does not provide new information but the interaction term between the text and the risk factors reliably differs from zero which indicates that differences in disclosures affect confidence in the numerical estimates
Rekabsaz et al. (2017)	Risk factor section in 10-K-reports	Reports between 2006-2015 US stock market	BoW TC, TF, TF-IDF	Word embeddings-based IR Models	The volatility prediction on as a continuous estimatio reaches an r^2 of 0.527 which beats earlier similar studies
Feuerriegel & Gordon (2018)	Regulatory disclosures (Ad-hoc)	75000 documents between 1996-2016	BoW, TF, TF-IDF Stop word removal	Sentiment score, Multiple Regression	Algorithmic trading systems around text-based stock price prediction, especially with the use of corporate ad-hoc disclosures, are evidently capable of executin profitable strategies of trading

Table 1 depicts an overview of related studies predicting stock prices or relevant market indicators with text data. Most of the studies covered use corporate disclosures, both pre-announced and ad-hoc, as they most closely relate to this study's scope.

BoW = Bag of Words, C4.5 DT = C4.5 Decision tree, DNN = Deep neural network, k-NN = k – nearest neighbors, LDA = Latent Dirichlet Allocation, LR = Logistic regression, MDRM = Multimodal Deep Regression Model is Effec- tive, MNB = Multinominal Naïve Bayes, NB = Naïve Bayes, NBSVM = Naïve Bayes Support Vector Machine, NN = Neural network, RF = Random forest, SVM = Support vector machine.

2.7. Evaluation

In Machine Learning and Natural Language Processing, model performance must be evaluated. Thus, reference points are needed to determine how “good” or “bad” a certain model performs in a certain situation. Therefore, the question “did it perform substantially better than a baseline model?” is common in Machine Learning (Fawcett & Provost, 2013). A baseline needs to be chosen given the circumstances for the Machine Learning problem to be solved to be reasonable for model performance comparison. This is important to both demonstrate to stakeholders what value the data mining has led to, as well as for understanding if performance is improving.

For classification models it is often easy to simulate a baseline with the ZeroR model, which simply assigns all unseen instances of the test data to the largest label in the training dataset (Fawcett & Provost, 2013). If there is a 60/40 distribution in the dataset, the accuracy of the ZeroR model should be 60% as it classifies all instances to the label with the highest prevalence. If a model can classify the data better than the baseline, it has learned something. Sometimes it could be easy to beat such a model (or seem easy). Therefore, it is quite common to use one or more additional baselines created from the use of a simple but not simplistic model e.g. Logistic Regression with standard parameters. (Fawcett & Provost, 2013). In the research field of predicting stock prices with text data, it is also common to use several baselines as algorithms that are considered to be ‘basic’ as well as results from earlier studies for them to beat with proposedly more sophisticated models in line with the methodology of (dos Santos Pinheiro & Dras, 2017a).

In data science, it is important to remember the underlying task that is to be solved. Furthermore, one should evaluate the model performance from this perspective as well e.g. how well the model is performing in practice. This can be accomplished by looking at for example expected value where monetary results from different actions is applied to the confusion matrix. In general, and from a business perspective, it is common to use the simple method of evaluating the model based on the monetary value it creates. (Fawcett & Provost, 2013). A common way of measuring monetary value creation in the financial field is through abnormal returns after an ad-hoc disclosure (Muntermann & Guettler (2007).

In summary, evaluation and baselines are important concepts of a machine learning model's performance. The performance needs to be proven better than relevant baselines to claim any fame and make inferences about a certain model succeeding to learn from data and/or make a significant positive difference for any stakeholders. In classification, ZeroR models can easily be simulated by assigning all instances to the biggest class but in this classification task, baselines regarding expected value are also of interest to test if the model is useful from a monetary perspective.

2.8. Summary of theoretical background

To summarize the theoretical background which subsequently leads to the hypotheses below, we have discussed first of all what a stock market is and how information in relation to the market can be seen as a commodity that allows agents to take action. Furthermore, we discuss some of the specific rules that are relevant for the notion of stock markets, e.g. regulations for how information must be distributed in order to give agents equal opportunities to receive and act upon the information, which can be at both pre-announced dates as well as more unexpectedly.

After the summary of what a stock market is and how it is regulated, the Efficient Market Hypothesis is presented. The Efficient Market Hypothesis is one of the strongest hypotheses in social sciences. Despite having explicit weaknesses, the hypothesis is not hindered from being useful both from a theoretical and practical perspective. The Efficient Market Hypothesis views the stock market as a fair game where it should be impossible to continuously beat a representative buy-and-hold strategy if a market is efficient, where new and old information always is fully reflected in the stock price.

The hypothesis is simple as it provides ways of sidestepping difficult issues of defining e.g. trading costs while providing clear rules for testing it. Tests of the Efficient Market Hypothesis can be done in three forms where the semi-strong form is the most tested.

There are many studies that present evidence of inefficient markets in specific events which the stock market seems to either under- or overreact to why stock prices are adjusted “correctly” in the long term rather than in the short term. Some of the evidence for inefficiencies is relatively more robust as replications also find similar evidence, but critics of some of the results point out that any evidence of anomalies may be a result of flawed models that estimate normal returns and compute abnormal returns.

There is also evidence of inefficiencies that does not per se indicate that the market under- or overreact to a certain event but rather reflects new information in the stock price with an intra-day time lag on average. A time lag that on average is about 30 minutes is observed on the German stock market in two different studies when examining stock market reactions to ad-hoc disclosures that in their ideal form are released unexpectedly to the market. This is relevant for a trading strategy that in addition to attempting to correctly classify a certain stock price movement, which is difficult enough, also needs to beat the average market actor in order to capture value from the correctly classified ad-hoc disclosure when trying to take action before the new information is fully reflected in the stock price.

The prevalence of time lags to unexpected ad-hoc disclosures could be explained by the processing costs that new information implies where for instance different agents may have different costs to process the information. Furthermore, the cost of processing information seems to be possible to explain by how much soft vs hard information a certain piece of information has.

With the financial theory as a foundation that helps us approach the goals of the study, the theory section subsequently focusses on the concepts that more explicitly approaches the study’s research question. Firstly, Event Based Trading and Algorithmic trading is presented as the concept of automating strategies based on certain properties as for instance a machine learning model. In Event Based Trading, certain informational events are used as inputs with pre-defined trading strategies to rapidly execute actions which is enabled by the digitalization of stock markets and related services. Algorithmic Trading builds upon the Event Based Trading framework but with a narrower focus on how one or more algorithms interact in any programmed trading system which have in history proven also to have a great impact on stock markets, both good and bad. It is worth mentioning in relation to Event Based Trading and Algorithmic Trading that they may in their perfect form be very autonomous and automatic, but generally the steps of collecting and cleaning data requires a lot of energy. This study utilizes a theory-driven Algorithmic Trading approach as we theocratize about certain expected price movement behaviors after some specified events.

After Event Based Trading and Algorithmic Trading, the broad concepts of Artificial Intelligence and Machine Learning are presented to provide context to how this study relates to both concepts. For Artificial Intelligence, the study is relevant from the perspective of artificial behavior that behaves *ideally* in relation to a goal, which in this study is to among other things generate economic value. Machine Learning on the other hand more closely related to the study's practical steps, since it is more closely concerned with how systems can learn from environments to adapt to them. Stock price prediction based on known historical movements fits into the supervised learning methodology, which also relates to the theory-driven approach of Algorithmic Trading.

After discussing Machine Learning, Natural Language Processing is presented as the area within Machine Learning that this study covers. The attention for Natural Language Processing in the financial sector has in the latest years grown as many text documents with value relevant content are available for analysis. Three applications of Natural Language Processing based stock price prediction are common in the financial sector where corporate disclosure which are studied in this thesis is one, and financial news and social media constitute the other two.

Lastly, evaluation as a concept within machine learning and Natural Language Processing is presented which discusses that it is important to remember the underlying task to be solved. This has implication for how performance is evaluated in different ways regarding how the model performs e.g. in practice. For classification tasks, the ZeroR model is often used as a baseline to beat but also the expected value method focuses on monetary results which are of interest in this study as the trading algorithm is meant to be tested on the stock market. This leads the study to the two hypotheses that align with the research question of the study. The hypotheses specify what is tested based on the theoretical background and related assumptions, while separating the research question at the most natural point.

2.9. Hypotheses:

Research Question: *Can a machine learning model trained on textual corporate ad-hoc disclosures from the Swedish stock market exceed relevant baselines and generate positive abnormal returns used as a trading strategy?*

2.9.1. H1

A machine learning model built on ad-hoc disclosures from the Swedish market can learn from the data and classify unseen disclosures better than a relevant baseline.

2.9.2. H2

Since the stock market reflects new ad-hoc information in the stock price with a time lag, the model can be used as a part of a trading strategy on the Swedish stock market for generating abnormal returns.

2.9.3. Hypotheses description

H1 focuses on the machine learning domain. It is model building centered in order to lay weight on utilizing machine learning techniques and evaluation metrics in a desirable way. H2 is more practically concerned with testing the machine learning model on the market as a trading strategy. Consequently, H2 is testing whether the notion of a semi-strong market holds, since positive abnormal returns based on acting fast should not be possible to observe over time.

3. Methodology

3.1. Introduction

The methodology section provides the study with the methodological approaches with a combination of finance and machine learning perspectives. The subsequent sections “Experiment” describe more in depth the practical application of collecting, processing, and modeling the specific data in the study.

The methodology is divided into five parts:

1. **Event study** – Provides the study with a framework for conducting research on stock markets for measuring stock price movements in relation to a certain event
2. **Labeling** – With basis in measurements around an event, labeling concerns how documents are labeled to power the supervised learning of the machine learning models
3. **Pre-processing** – Different techniques and approaches for pre-processing text data are discussed
4. **Machine Learning models** – Relevant classifier models are discussed and the evaluation of them
5. **Trading strategy using algorithmic trading** – How to simulate the application of the machine learning model on a market for final evaluation is discussed.

These parts also represent the progression of answering the hypotheses that are contingent on each other. Some steps naturally must be taken to firstly address H1 before H2 has the prerequisites to be answered.

3.2. Event study

Analyzing stock price movements in relation to an informational event, as e.g. a profit warnings or ad-hoc disclosures, can with good reason follow an event study structure. An event study defines important methodological choices to be able to measure reactions of the market to any informational event (Kothari, 2001). Naturally, this includes questions about what time periods that are measured and what metrics that are used for measuring. Depending on the study, it can be relevant to measure different metrics, for example stock prices, trading volume and/or bid ask-spread (Kothari, 2001). Measuring stock price movements are however the most common (Kothari, 2001). MacKinlay (1997) state that there are four aspects important to define early on when conducting an event study:

1. Defining the event
2. Defining criterions for firms to be included
3. Choosing finance model
4. Defining the event window

Studies that similar to this analyze and construct models with ad-hoc disclosures, for example Kim et. al. (2018) and Feuerriegel & Gordon (2018), briefly mention the Efficient Market Hypothesis as a background for making assumptions about over-performing the market but focus mostly on Natural Language Processing methods and modeling. This indicates that that there might exist value in utilizing more financial theory to provide the best prerequisites for a machine learning model to succeed on a financial market. Further, Fama (1991) states that event studies are the best method to test whether the semi-strong form of the Efficient Market Hypothesis holds, which specifically H2 addresses. Consequently, to ensure that financial theory is utilized in a structured and normalized way to capture the potential value of financial theory, this study follows the framework of an event study.

The two hypotheses of the study have somewhat different properties which also means that the event study framework in some areas needs to be adjusted in accordance respective hypothesis. How these adjustments impact the method choices for each or both hypotheses within the event study framework is presented under the relevant definition section.

3.2.1. Defining the event

Since an event study is an analysis of the market's reaction to certain novel information, it is important to ensure that the information is indeed novel rather than old, otherwise the information should be presumed to already be reflected in the price (MacKinlay, 1997). This can of course not be entirely certain since marginal, although occasionally substantial, insider trading can be assumed to occur.

A defined event commonly refers to a type of event comprising several individual observations. The event definition should therefore clearly state what kind of new relevant information that meets certain criteria to be included as a part of the study (MacKinlay, 1997). The release of a firm's annual report could for instance be considered an event. Since all public firms are obliged to disclose this specific information, the definition of the event is extended to several observations of the same phenomenon (MacKinlay, 1997).

3.2.2. Event definition – Ad-hoc disclosures

The event in the study is defined as when firms on the Swedish stock market publish so-called ad-hoc disclosures. This means that information is published ad-hoc to inform the market about any new information that could be relevant for the valuation of the firm, which is regulated by law. Going forward in this study, the corporate disclosures published ad-hoc, which comprise text data, are referred to as "ad-hoc disclosures".

By clearly defining the event as above, this study is provided with rules about what kind of data that fulfils the prerequisites for being included, which in turn facilitates the data collection process. It also guides the calculation of price movements around the time of the publication of the ad-hoc disclosure, which has two aspects in this study. First, the measured effects of the ad-hoc disclosures are used for labeling documents, which form the basis for supervising the learning of the machine learning model. Second, the measured effects of the ad-hoc disclosures form the basis for how to measure the earnings generated by the constructed machine learning model when simulating it as a part of an algorithmic trading strategy on the market.

3.2.3. Define criterions for firms to be included

The criterions for firms to be included in an event study often deal with whether a certain stock exchange or country is of interest in the study, if there are any size restrictions for firms to meet or if there are any industry specific criterions used for selecting firms. Criterions that are linked to the event can also be relevant if the event only has occurred for a certain part of firms in the past. (Kothari, 2001)

3.2.4. Definition of criterions for firms – Swedish market

The criterions for a firm to be included in this study are that they are traded on the Swedish stock market Nasdaq on large, mid or small cap per 2020-03-11 and have released at least one ad-hoc disclosure between 2010-01-01 and 2020-03-11, where the ad-hoc disclosure(s) have been uploaded to and is searchable on Nasdaq OMX Group’s web page per 2020-03-11.

With this definition follows that no discrimination is made on e.g. company size or industry. For example, Kim et al. (2018) reason that words convey different sentiments for different industries; they therefore separate firms and build industry specific models. Such an approach could have been relevant for this study, but given the time available, it has been deemed too complex to sufficiently tackle. Furthermore, a narrower selection of firms would reduce the number of instances in the dataset and thus impact the generalizability of the machine learning algorithm.

The above definition does however bring about a survivorship bias for companies that are listed for companies per 2020-03-11; companies that historically existed on OMX Stockholm, but were de-listed, are not included. De-listed companies might have been less successful than companies that have been listed for a long time, why there might exist a survivorship bias regarding firms included in the data set.

3.2.5. Choose model

The model choice regards how the price movement that occurs as an effect of the event are measured, which using financial terms is called measuring *returns*. (Kothari, 2001)

To isolate the effect of an event onto the price of a stock, one must more specifically measure *abnormal returns* (MacKinlay, 1997). The abnormal return is the difference between the actual return and the estimated return, as depicted in Equation 1 (MacKinlay, 1997).

Equation 1 - Calculation of abnormal returns

$$AR_{i\,T} = R_{i\,T} - E(R_{i\,T}|X_T)$$

$\boldsymbol{AR_{i\,T}}$ = *abnormal returns*
$\boldsymbol{R_{i\,T}}$ = *(actual) return*
$\boldsymbol{E(R_{i\,T}|X_T)}$ = *estimated normal returns*
$\boldsymbol{X_T}$ = *conditioning information for the normal return model*

To determine if the event affected the price of a stock, it is common to use a financial model for estimating normal returns. Since the calculation of normal returns is based on the risk factors adjusted for in the financial model, the choice of model used to estimate normal returns is not uncontroversial; any evidence of effect as occurring from a particular event (abnormal returns) becomes split between the weaknesses of the model and the effect of the actual event. When conducting an event study, one should therefore motivate why the chosen model is relevant for the task, and strengths and weaknesses of the model should be discussed (MacKinlay, 1997).

3.2.6. Model choices for measuring returns

The model of choice to estimate normal returns impacts how the machine learning model is trained in the supervised learning, because the model impacts the returns measured around the event. Within this study's scope, it is deemed possible to test three different models that have different levels of sophistication from a financial perspective. This is to see if a higher level of sophistication, i.e. taking a higher number of risk factors into account in the measurements of returns, leads to an improved supervised learning. Measuring returns with these three models is also relevant for H2 since it defines the abnormal returns associated with the simulation of the trading algorithm.

3.2.6.1. Returns

The first and most basic method for computing abnormal returns that is used in this study, takes the shortcut of not estimating any normal returns at all. This would imply that all observed stock price movements, i.e. returns, after an ad-hoc disclosure are assumed to be abnormal and simply calculated by dividing the stock price after the event with the price before the event. Going forward in this study, this method is called Returns.

Equation 2 - Returns

$$R_{i,n} = \frac{Stock\ price\ after\ event}{Stock\ price\ before\ price} - 1$$

$\boldsymbol{Stock\ price\ before\ event}$ = *The relevant price before the event defined in the event study to start measuring from*
$\boldsymbol{Stock\ price\ after\ event}$ = *The relevant price after the event defined in the event study to end measurements from*

3.2.6.2. Market Adjusted Returns

When evaluating whether a piece of firm specific information has changed the price of a stock, it is however important to consider that stock price movements might occur both because of company specific events and from more general macro events. Consequently, the fact that the stock price has moved in proximity to the observation of some event does not necessarily mean that the stock price has moved *because of* the event.

The second method of measuring returns partially mitigates risk regarding macroeconomic movements by defining abnormal returns as the difference between the stock price movement (return) and the market. This is called the Market Adjusted Returns model (Stephantt, 1984). For example, on the first trading day after the British referendum about Brexit, the Swedish stock market plummeted; the OMXS PI[1] traded nine percent down (-9%) compared to the previous day. Using the market-adjusted returns model, the expected return on this day for a hypothetical company is the same as the market return. However, if the company on this day had released a positive ad-hoc disclosure, the price might have recoiled with five percent (+5%), thus closing on minus four percent (-4%). Return (-4%) minus Market Return (-9%) equals Market-Adjusted Returns (+5%).

Using this example, if the actual return (-4%) would be used for supervising the learning of a machine learning model, the learning would be flawed because the model would associate the positive ad-hoc disclosure with a negative stock price movement. By instead using the Market Adjusted Returns (+5%), the machine learning model could potentially capture the signal of the ad-hoc disclosure more correctly.

Market Adjusted Returns calculation

The Market Return used in the Market Adjusted Returns calculation is in this study approximated with the broad OMX Stockholm General Index (OMXSGI). The OMXSGI includes the movements of all stocks listed on OMX Stockholm, adjusted for dividends. The adjustment for dividends contrasts it from the OMXS all-share index (OMXSPI), which does not adjust price movements. The OMXSGI is used instead of OMXSPI because price movements occurring from dividends are an example of a price movement that might obscure the price movement caused by an ad-hoc disclosure, i.e. the causality of the event that is being studied. The calculation is presented in Equation 3.

[1] Stockholm all-share index

Equation 3 - Market Adjusted Returns

$$R_{i,n,OMXGI} = \frac{Security\ price\ after\ event}{Security\ price\ before\ price} - \frac{OMXSGI\ price\ after\ event}{OMXSGI\ price\ before\ event}$$

$\boldsymbol{R_{i,n,OMXGI}}$ = *Market Adjusted Returns*
$\boldsymbol{Stock\ price\ before\ event}$ = *The relevant price before the event defined in the event study to start measuring from*
$\boldsymbol{Stock\ price\ after\ event}$ = *The relevant price after the event defined in the event study to end measurements from*
$\boldsymbol{OMXSGI\ price\ before\ event}$ = *The relevant index price measured at the same time as the stock price before the event*
$\boldsymbol{OMXSGI\ price\ after\ event}$ = *The relevant index price measured as the same time as the stock price after the event*

Jensen’s Alpha in the context of CAPM

More sophisticated financial models than the Market Adjusted Returns model estimate normal returns and adjust for additional risk factors associated with the stock. Risk is in this context used as a measure of uncertainty (Qian, 2019), where investors are assumed to be risk-averse (Jensen, 1967). One of the first, slightly more sophisticated methods of estimating normal returns is called the Capital Asset Pricing Model (CAPM) (Sharpe, 1964). CAPM estimates the future price of an asset as the risk-free rate plus a risk premium. The risk premium is estimated using the systematic risk associated with the security which is derived from its historical relationship with the returns of the market (Sharpe, 1964). With Jensen’s Alpha in the context of CAPM, abnormal returns (alpha) is the difference between the actual return and the normal return as estimated by the Capital Asset Pricing Model (CAPM). Even though Beta in this study represents the most financially sophisticated model, it has received critique for being too simplistic and that more risk factors should be included to better estimate normal returns. Such even more sophisticated models are too time consuming for this study to perform why the simplistic CAPM model is deemed better to use rather than using no risk-adjusting financial model at all.

Jensen's Alpha in the context of CAPM calculations

The third Price Movement calculation is defined as the abnormal returns commonly referred to as Jensen's Alpha (Jensen, 1967). Jensen's Alpha is depicted in Equation 4 and CAPM in Equation 5. As shown, Jensen's Alpha is the difference between the actual return and CAPM.

Equation 4 - Jensen's Alpha

$$Jensen's\ Alpha = R_{in} - [R_f + \beta_{im} * (R_{mn} - R_f)]$$

$\boldsymbol{Jensen's\ Alpha}$ = Abnormal return after an ad-hoc disclosure
$\boldsymbol{R_{in}}$ = Actual return for security i on day n after an ad-hoc disclosure
$\boldsymbol{R_f}$ = Risk-free rate
$\boldsymbol{\beta_{im}}$ = Beta of security i in relation to market m
$\boldsymbol{R_{mn}}$ = Actual returns for the market m on day n after the ad-hoc disclosure

Equation 5 - Capital Asset Pricing Model (CAPM)

$$ER_i = R_f + \beta_{im}\ (ER_m - R_f)$$

$\boldsymbol{ER_i}$ = Expected returns for security i
$\boldsymbol{R_f}$ = Risk-free rate
$\boldsymbol{\beta_{im}}$ = Beta of stock i in relation to market m
$\boldsymbol{ER_m}$ = Expected returns for the market m

CAPM was first formulated by Sharpe (1964) and estimates normal return of an asset as the return above the risk-free investment rate (R_f), where the price of any security equals that of the risk-free rate plus a risk premium (Jensen, 1967). The risk premium is estimated from the product of the systematic risk of the asset (β_{im}) and the risk premium of the market (R_{mn}).

The proxy for the risk-free rate used in this study is the rate of Swedish 3-months treasury bills. Treasury bills are commonly used as a proxy for the risk-free rate in financial research since they are among the most stable securities. The risk-free rate in Jensen's Alpha is expressed relative to the abnormal return of the period measured. Since the price movements (returns) as an effect of a released ad-hoc disclosure are measured daily, the risk-free rate of return is transformed to express that of the daily return. Equation 6 depicts the calculation of the risk-free rate.

Equation 6 - Calculation the risk-free rate of return

$$rf_{day} = (rf)^{1/365} - 1$$

$\boldsymbol{rf}$*= Risk free rate of return*

The market returns as expressed in CAPM is depicted in Equation 7. The market return is approximated with the return of OMXSGI during the time-period measured, i.e. one day.

Equation 7 - Calculation of market returns

$$R_{mn} = \frac{OMXGI\ after\ event}{OMXGI\ before\ event} - 1$$

$\boldsymbol{R_{mn}}$ *= Returns for the market* $\boldsymbol{m}$ *on day* $\boldsymbol{n}$.
$\boldsymbol{OMXSGI\ price\ before\ event} =$ *The relevant index price measured at the same time as the stock price before the event*
$\boldsymbol{OMXSGI\ price\ after\ event} =$ *The relevant index price measured as the same time as the stock price after the event*

Beta in CAPM works as a way of describing a security's systematic risk by relating the stock's historic price movements relative to those of the market. The risk is calculated using historic volatility. If Beta > 1, the stock is expected to move more volatile than the market, and vice versa (Sharpe, 1964). For example, if the market returns one percent on a given day, a stock with a historical Beta > 1 is expected to have a return >1 percent. In CAPM, a higher Beta constitutes a higher estimated normal return. Hence, the measured abnormal returns when using Jensen's Alpha could become smaller compared to the previously mentioned Price Movement calculations. This is the case because rational investors are assumed to be risk-averse, i.e. where return is desired and risk, *ceteris paribus*, is undesired (Jensen, 1967). Equation 8 depicts the calculation of Beta.

Equation 8 - Calculating Beta of a security

$$\beta_{im} = \frac{Cov(r_i, r_m)}{Var(\,r_m)}$$

$\boldsymbol{\beta}$ *= Beta of the security in relation to the market*
$\boldsymbol{Cov(r_i, r_m)}$ *= Covariance between security* i *and market m*

$\boldsymbol{Var(\ r_m)}$ = *Variance of market m*

Equation 9 depicts the calculation of the covariance as constituent in Beta. The index used for approximating the market return is OMXSGI. The estimation period used in this study is 30-60 days. This choice of estimation period is motivated in the next section.

Equation 9 - Calculation of $\boldsymbol{Cov(r_i, r_m)}$ = Covariance between stock $\boldsymbol{i}$ and market $\boldsymbol{m}$, $\boldsymbol{m}$ = OMXGI, adjusted for one degree of freedom.

$$Cov(r_i, r_m) = \frac{\sum_{n=1}^{N}(x_n - \bar{x})\ (\ y_n - \bar{y})}{N-1}$$

$\boldsymbol{xn}$ = *the price movement of stock i on day n.*
$\boldsymbol{\bar{x}}$ = *the mean average of all price movements of stock i during the estimation period.*
$\boldsymbol{yn}$ = *the price movement of OMXGI on day n.*
$\boldsymbol{\bar{y}}$ = *the mean average of all price movement of OMXSGI during the estimation period.*

Equation 10 depicts the calculation of variance as constituent in Beta. The index used in this study for approximating the market return is OMXSGI. The estimation period used in this study is 30-60 days. This choice of estimation period is motivated in the next section.

Equation 10 - Calculation of $\boldsymbol{Var(\ R_m)}$ = Variance of Returns on market $\boldsymbol{m}$

$$Var(\ R_m) = \frac{\sum_{n=1}^{N}(\ y_n - \bar{y})^2}{N}$$

y_n = *the price movement of m OMXGI on day n.*
$\boldsymbol{\bar{y}}$ = *the mean average of all price movement of OMXSGI during the estimation period.*

This summarized the calculations needed for computing Jensen’s Alpha in the context of CAPM.

3.2.6.3. More sophisticated models

The Fama & French (1993) three-factor model is an extended version of the CAPM that also includes the factors SMB[2] and HML[3]. Further developing the model, the five-factor model as developed by Fama & French (2015) also takes profitability and investment factors into account. These models are well known within finance and would be natural extensions of the Jensen's Alpha calculations, especially as Beta is included in both the three-factor and the five-factor model. With a bit more time it would have been preferable to also use one or both of the models, but it is simply not possible given the scope of the study as a lot of additional data is needed to compute the extra risk factors.

3.2.6.4. Summary of financial model choices

The choice of financial model for measuring stock price movements as an effect of ad-hoc disclosures regards both hypotheses of the study. To investigate if more sophisticated price movement measurements improve the learning abilities of the machine algorithm, this study uses and compares three methods for calculating price movements: Returns (simply using the returns as abnormal returns), Market Adjusted Returns (abnormal returns as calculated by adjusting returns with the market index), and Jensen's Alpha (abnormal returns as calculated by adjusting returns with Beta, the risk-free rate and the market index). The same three methods are also used when computing abnormal returns for the simulated trading strategy. The more sophisticated the model, the higher the demand for what is considered abnormal, which is relevant from both the supervised learning perspective and the trading algorithm evaluation perspective.

[2] Small minus big as small firms tend to have higher returns compared to large firms

[3] High minus low where firms with a high Book-to-market tend to have higher returns compared to firms with a low Book-to-market

The learning of the machine learning model could be improved when computing returns more sophistically, because financial theory suggests that it is better to adjust for risk factors, since it reduces noise in the data and therefore better captures the pure effect of an event. This can be tested in the context of this specific event study. To illustrate if this is the case, three different price movement calculations are used by applying a machine learning classifier onto each one. They are evaluated and compared, based on the respective accuracies of the classifier.

For the algorithmic trading strategy, higher demands for abnormal returns means that potential results would be more robust in financial research. Without a financial model to measure abnormal returns, it is difficult to know whether the algorithm succeeds in earning money, or if potential earnings would have occurred without the use of the algorithm. This means that the three measurement methods are evaluated from a slightly different perspective regarding the trading algorithm, compared to the construction of the machine learning model where the highest accuracy should indicate which method that learns the most.

3.2.7. Defining the event window

The event window specifies from which period that parameters regarding the defined event should be collected (MacKinlay, 1997). For example, the estimation of normal returns as defined by CAPM takes place before the event during the so-called estimation period, which is part of the event window. It is thus only for one of the three methods of measuring returns (Jensen's Alpha) relevant to have an estimation period as the two other methods only compute returns right before and after the event. Event studies using daily data most commonly use an estimation window of between 30 and 250 days before the event. The event window seems in many studies to be chosen arbitrarily (Aktas et al., 2007). If normal returns are estimated on data from 0 to 50 days before the event and abnormal returns are calculated on the event day (after the particular event observation occurs) and the day after, the event window would be defined as 50 days (estimation period) + event day +1 day (post-event measurements), thus equaling to 52 days.

3.2.7.1. Event windows for the supervised learning

In this study, there are multiple event window definitions, because three different price movement measurements are used when learning the model about what document represents what movement. For the Returns and Market Adjusted Returns methods, the event window is the time frame used when measuring the price movement, i.e. approximately one day when measuring the price movements used for supervising the machine learning algorithm.

For Jensen’s Alpha, the window is longer because it includes the estimation of Beta. The chosen estimation period for Beta is between 30-60 days before the event takes place. This somewhat shorter period compared to other financial studies is chosen for the purpose of maximizing the number of instances that can be labeled in the case of the particular dataset acquired for this study; for example, ad-hoc disclosures occurring in January 2010 do not have corresponding stock prices movements far enough back in time. We use a period of 60 days for the documents that have 60 days available in the dataset. For datasets shorter than 60 days, we accept Beta calculations based on as little as 30 days before the event takes place.

For events with less than 30 days, we replace the calculation of Jensen’s Alpha with the Market Adjusted Returns method. The choice of replacing instead of dropping rows is made because we want to better be able to illustrate the differences between the three compared types of price movement calculations used for supervising the learning of the algorithm. By dropping rows, the quality of the comparison is deemed to be reduced because differences in performance of the applied classifier could have been due to the difference in learned instances rather than due to differences between price movement calculations. A longer estimation period could help develop a more accurate beta which would lead to better estimations of normal returns, but as motivated above, more instances with an acceptable estimation period have been preferred.

3.2.7.2. Event window for simulating a trading strategy

As the trading strategy must be evaluated on more specific data than daily given that certain actions are simulated, the event window differs somewhat. With basis in theory about time lag which assumes that stock markets absorb new ad-hoc information with an average time lag of 30 minutes, the post event measurements are altered to test the algorithm properly as if it had been deployed in practice.

With access to tick data (each trade made on the market) for the simulation, the actions that are meant to generate abnormal returns need to be simulated as occurring right after the release of a new ad-hoc disclosure. The event window is hence directly after the event as if the document would trigger a trading action. When applying the trading algorithm in practice, the disclosure must be published before the beginning of the event window. The specific actions of the trading algorithm are discussed in later sections.

As the three price movement calculation methods are also used for the calculation of abnormal returns in the simulation, the same estimation period for Jensen’s Alpha is used. The difference is now that the event calculations start after the publication of the disclosure and ends one hour after. One hour holding period is motivated by the fact that we theorize about an average time lag of 30 minutes. With this relatively short measurement time, the effect of the ad-hoc disclosure is also more likely isolated as it is unlikely that other events that occur that change the value of a firm within one hour of publishing a new ad-hoc disclosure.

3.3. Labeling

3.3.1. Classification vs regression

After each instance in the dataset has been associated with a corresponding price movement through the utilization of the financial event study framework, it is possible to predict the returns occurring from new ad-hoc disclosures by applying a regression model. In this study however, we instead choose to classify the instances. As written by Fawcett & Provost (2013), "classification is about *whether* something will happen, whereas regression predicts *how much* something will happen". For the task of predicting the stock market and earning money, it is more difficult to predict a specific value of a certain performance measure because a real value prediction is more granular than a category prediction (Balakrishnan et al., 2010). In addition, classification can be clearly translated into rational action for an investor. If the labels are "Up" and "Down", one is easily translated into "Buy" and the other into "Sell" (instead of selling, one could also decide to short the stock depending on if the stock is already owned etc.); the degree of "Up" does not matter, only which action to take. Classification rather than regression is therefore used in this study.

3.3.2. Classes

With the choice of classification in a supervised setting as method, a specific number of classes must be defined. By looking at the problem as binary, the movements could be classified as "Up" or "Down". The natural cut-off threshold between the classes would then be 0; an ad-hoc disclosure resulting in a measured price movement above 0 would be labeled "Up", and an ad-hoc disclosure resulting in a measured price movement below 0 would be labeled "Down". Translated into action (Buy or Sell/Short for instance), all positive or negative price movements are however not desirable to act upon for a rational investor. First, each transaction does in practice come with an associated transaction cost (fee) to the stockbroker, which makes acting (Buying or Selling/Shorting) based on ad-hoc disclosures where the expected return is too low undesirable.

Second, if the expected return of a prediction is above but close to 0 (i.e. predicted as “Up”), there is a higher probability that this prediction could be a downward price movement, than if the expected return was positive and farther away from 0. More formally, the closer the prediction is to the cut-off threshold, the lower is the probability that the prediction belongs to the correct class, where the important part is that the actual price movement is above 0 for Buy actions and below 0 for Sell/Short actions. Since many documents also might convey little to no relevant information that investors act upon, there is reason to understand which documents that do not result in any significant reaction from the market. In these cases, the stock for example continues to be traded similarly after the disclosure as before publication which also indicates that there might be something to learn, e.g. what investors deem uninteresting, from those documents as well.

This also aligns with the notion that all observations of abnormal returns naturally are split between the model’s weaknesses and the actual effect of the event. Having in mind that the study uses three relatively simple measurement methods, with the most sophisticated method (Jensen’s Alpha) being simplistic compared to more complex models, there are plenty of weaknesses to be aware of that motivates that not all movements above/under 0 are actual negative or positive reactions to an ad-hoc disclosure.

Consequently, the three classes “Up”, “Down” and “Stable” are used in this study. This aligns with Balakrishnan et al. (2010), who also construct the stock price prediction into a three-class problem.

3.3.3. Class cut-off thresholds

With a binary problem such as “Up” or “Down”, the cut-off threshold would naturally be 0. The choice of the three classes “Up”, “Stable” and “Down” however results in a subsequent choice of which cut-off threshold to use when deciding on what underlying price movement that counts as "Stable" and what counts as "Up"/"Down". From a machine learning perspective, the best cut-off threshold should be where the model learns the most, which is reflected in the accuracy of the model; all classes are then valued equally. From a financial perspective, with regards to transaction costs, weaknesses of the financial models used for measuring price movements, and the risk for incorrect predictions by the machine learning model, it is desirable that this threshold is not too close to zero.

We therefore set the smallest acceptable classification threshold at 0,5 percent, and choose test 8 additional thresholds, to see at which threshold the model performs the best. The tested thresholds are thus 0,5 to 4,5 percent with a 0,5 increase each time. For example, when testing the 0.5 percent threshold, measured price movements (with the three different methods) that are higher than +0.5 percent are categorized as "Up", lower than -0.5 percent are categorized as "Down", and movements in between are categorized as "Stable".'

Table 3 - **Classes**

UP	**STABLE**	**DOWN**
The stock price is expected to increase significantly because of the ad-hoc disclosure.	The stock price is not expected to move significantly in any direction because of the published ad-hoc disclosure.	The stock price is expected to significantly decrease because of the ad-hoc disclosure.

Table 3 depicts and explain the classes used in the study both in the labeling process and the classification / prediction process.

3.4. Pre-processing

3.4.1. Introduction to pre-processing

To pre-process text data, we need to know what type of string data that we are dealing with. String data can be divided into four categories: (Müller & Guido, 2016).

1. Categorical data (e.g. specific pre-defined labels),

2. Free strings that can be semantically mapped to a category,
3. Structured string data
4. Free text data

In this study we focus on category (4), free text data.

Data in the form of text is commonly described as dirty and unstructured as it can come in many shapes and forms compared to for example numerical data that simply are numbers. Text does have structure, but the linguistic structure often requires more advanced pre-processing since text analysis is harder for computers. Hence, we need to transform the raw text data to something that the computer can handle and analyze. This is called pre-processing. (Fawcett & Provost, 2013). There are several methods to pre-process text data, each with their advantages and disadvantages. Different pre-processing methods are used and compared in this study to find the best way of preparing the data for the study's classification task. First, we introduce the Bag of Words approach that is often the first step used to quantify the qualitative nature of text. Second, we introduce two techniques that are commonly used to make sense of text data (1) One hot representation and (2) Embeddings based feature vector.

3.4.2. Bag of Words

As text data is hard for computers to analyze and make sense of, the Bag of Words technique is often used in Natural Language Processing to quantify the qualitative nature of text data. Bag of Words is considered as a simple but effective method of pre-processing text data. (Fawcett & Provost, 2013; Müller & Guido, 2016).

In a Bag of Words approach, there are three steps:

1. First, each document (instance) in the corpus (text dataset) is converted into a bag of words. This transformation is called tokenizing where each word in the document is treated as a

separate token. Thus, all formatting of the document, e.g. chapters, sentences, formatting and special characters, are removed.

2. Second, out of the corpus a vocabulary is built. The vocabulary consists of all tokens found in the corpus.
3. Third, each tokenized document is analyzed to search for the occurrence of words from the vocabulary. There are two main techniques used with a Bag of Words approach. (1) One hot representation, where the occurrence of words in the documents are counted, either as a binary count or a continuous count. (2) Word embeddings, where each word is mapped in a low-dimensional space using cosine similarity. (Goldberg, 2016).

Figure 4 - **A visualization of the Bag of Words technique**

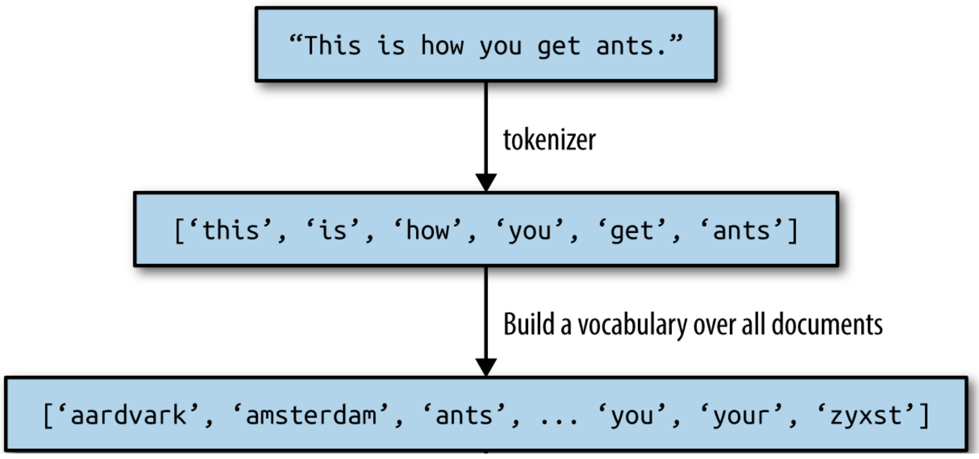

"This is how you get ants."
tokenizer
['this', 'is', 'how', 'you', 'get', 'ants']
Build a vocabulary over all documents
['aardvark', 'amsterdam', 'ants', ... 'you', 'your', 'zyxst']

Figure 4 is a visualization of the Bag of Words technique with a one hot representation commonly used in natural language processing to make sense of text data. In step one the document is tokenized. In step two a vocabulary of the corpus is constructed.

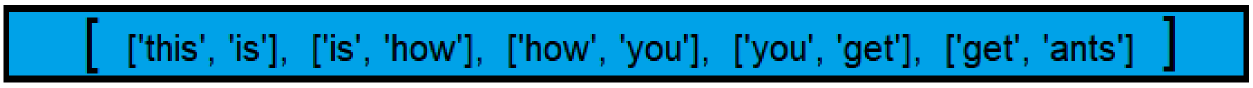
3.4.3. *N-grams*

One disadvantage of the bag of words technique is that e.g. the sentences "it's bad, not good at all" and "it's good, not bad at all" have the same representation even though the meanings of the sentences are opposite to each other. Hence, the N-gram technique is introduced where, in addition to using the words stand-alone, words are also used in conjunction with each other. In the abovementioned example, a bigram (two words together) results in “not good” and “not bad”. Using this technique can increase accuracy of a model as more meaning from a sentence can be included. The technique is called N-gram as the N can be replaced with any “number”. (Müller & Guido, 2016).

Only using one word is often called a unigram whereas using two words together is called a bigram, three words is called a trigram and so on. It is common to use all combinations below the selected N-gram; if for example a trigram is used, then bigrams and unigrams are also included. Obviously, this greatly increases the size of the vocabulary. Also, as the specificity of the model increases when using N-grams, the probability that e.g. a trigram occurs in a document is significantly lower than for a unigram or even a bigram, thus increasing model complexity without little or no increase in accuracy. (Müller & Guido, 2016).

Hence, it is common to use different techniques to remove N-grams that rarely occur or that occur very often and thus provide little information gain to the model. (Müller & Guido, 2016) One commonly used method used to remove N-grams (including unigrams / words) is called TF-IDF and is presented further below.

Figure 5 - **A visualization of the N-grams technique**



[['this', 'is'], ['is', 'how'], ['how', 'you'], ['you', 'get'], ['get', 'ants']]

Figure 5 is a visualization of the N-grams technique commonly used in conjunction with the Bag of Words approach in natural language processing to capture additional information in text data. Words a grouped together in pairs (bigrams), in pairs of three (trigrams) etcetera.

3.4.4. Character grams

Usually, words or unigrams are used as the smallest units in natural language processing. However, researchers have recently tried to further break down text data with the goal to improve accuracy. By looking at the character level of text data and creating character n-grams, sub-word information can be collected, thus increasing the probability to represent rare words in a better way as well as variations of words that otherwise would be treated as different words. (Wieting et al., 2016)

3.4.5. One-Hot Representation

A commonly used technique for representing the features extracted using the Bag of Words approach is called One-Hot Representation. In a One-Hot Representation, the dimensionality is the same as the number of features. This is often also called sparse feature vector since features is most often represented by a 0 as they only occur rarely in a document. Each document in the corpus is analyzed searching for the occurrence of each word. In a One-Hot Representation, each dimensionality can be represented in two ways: (1) By a binary count based on whether the specific word exists or not in each document. (2) Using a continuous count, e.g. based on the number of times each word occurs in each document is measured.

In Natural Language Processing, words vary in importance depending on the purpose of the model. As an example, words such as "he", "she", "this", "that" etc. occur in many documents in a corpus as they are needed to construct sentences. Such words are often called *stop words.* On the other hand, other words only occur in certain documents e.g. words as "lawsuit", "buyout", "bankruptcy" etc. These words most likely convey more information to the model compared to much more common words. Furthermore, some words only occur in very few documents, maybe one or two, and does thus not contribute particularly to the model. Furthermore, if a word occurs several times in a document it is likely that it is more significant compared to a word that only occurs once. As an example, if words such as "excellent", "great" or "strong" occur several times in a document, it is more likely to have a more significant meaning compared to if they only occur once. In the traditional bag of words approach, each feature is assigned a binary classification if it occurs in the document or not. (Fawcett & Provost, 2013).

Using *Term frequency* (TF) the number of times a feature occurs in a document is counted instead of using a binary count as in the basic bag of words approach. The notion of TF is that, for many text mining tasks, more frequent words should be more important than less frequent words. Figure 6 depicts a bag of words approach using TF and can be compared to Figure 6 that depicts the basic, binary, bag of words approach. (Fawcett & Provost, 2013).

Figure 6 - **A visualization of the Term frequency technique**

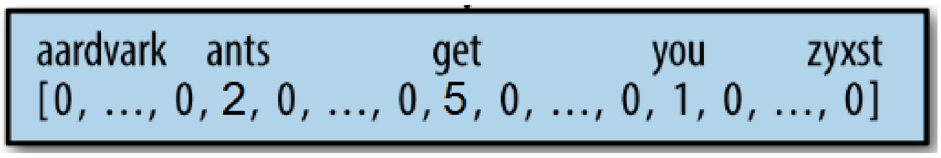


Figure 6 depicts how Term frequency is used in combination with the bag of words approach.

TF has a major drawback in that many words that occur frequently in one document also occurs frequently in other documents, for example stop words. One way of dealing with such words is to use the parameters included in TF. By setting a threshold for max_df (document frequency) in TF, words that occur in a higher number of documents than specified are not included in the vocabulary. Thus, common words that occur in most documents are removed from the vocabulary, leaving words that convey more information in the vocabulary. (Fawcett & Provost, 2013).

Another technique commonly used to overcome the problem with words that occur in many documents in the basic bag of words approach is called *Inversed Document Frequency* (IDF). The IDF technique provides words that occur in fewer documents with a higher weight, since they are assumed to be more significant. Conversely, words that occur in many documents are provided with a lower weight. (Fawcett & Provost, 2013) The IDF for a document is calculated as shown in Equation 11.

Equation 11 - Inverse Document Frequency (IDF)

$$Investe\ Document\ Frequency\ (IDF)_{(t)} = 1 + \log\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ t}\right)$$

Both TF and IDF can be used separately but it is common to use the techniques together by calculating the TF-IDF score for each feature as presented in Equation 12. The final score is increased if the word occurs frequently in a document, and decreased if the word occurs in many documents (Fawcett & Provost, 2013).

Equation 12 - TF-IDF

$$TFIDF_{(t,d)} = TF_{(t,d)} \times IDF_{(t)}$$

TF *= The frequency of a term in a specific document*
IDF *= The inverse document frequency of a word in the corpus*
TF IDF *= The combination between TF and IDF*

3.4.6. Embeddings based feature vector

One of the main drawbacks of using traditional bag of words approaches is that words are treated as independent features. Context is however often highly relevant, where words often are both dependent on and similar to other words. Embeddings based feature vector is a technique that attempts to solve these issues. Using unsupervised learning, tokens (e.g. words) are embedded on a pre-defined dimensional space by receiving a set of coordinates in the form of continuous vectors. This can be illustrated as constructing a map where tokens that occur in similar contexts (and thereby often have similar meaning) are mapped closely together. In contrast to a two-dimensional map, it is common to use somewhere between 50 and a few hundred dimensions when building a model. When each token has received its coordinates, it is possible to calculate the cosine similarity between the tokens, where similar tokens are geometrically closer to each other. For example, the tokens “yellow” and “orange” are often closer to each other than “yellow” and “horse” in a word embeddings model. (Goldberg, 2016).

Since the vectors reflect coordinates on the embeddings space, the vectors are dense, i.e. almost always non-zero. This contrasts one-hot representations where the vectors are sparse, i.e. mostly zero. The dense vectors of word embeddings often have the advantage of being able to highlight complex semantic relationships using simple mathematic operations. As an example, the relationship vector("King") – vector("Man") + vector("Woman") = vector("Queen") exists in the dataset trained by Le & Mikolov (2014). Another example is vector("Stockholm") – vector("Sweden") + vector("Germany") = vector("Berlin"). Another potential strength of embeddings is that there often are fewer dimensions, compared to one-hot representations where the number of dimensions equals the number of words. This reduction in the number of dimensions often result in reduced computation power. (Goldberg, 2016; Quanzhi Li & Shah, 2017).

When creating an embeddings-based model in practice, text data is fed into the model and assigned corresponding vector data from the embeddings, thus forming a dense vector. The sequence of vectors is subsequently fed into the deep network where they are converted to a compressed representation. The deep network is commonly a Recurrent neural network (RNN) or a Long Short Term Memory (LSTM) with dropout in order to reduce overfitting. The deep representations from the RNN / LSTM is transformed into the output classes in the fully connected layer. Softmax can be used in the output layer for both binary and multi classification. The full process of using dense vectors for text classification is visualized in Figure 7. (Nabi, 2018).

Figure 7 - **Visualization of a deep learning text classification process using embeddings**

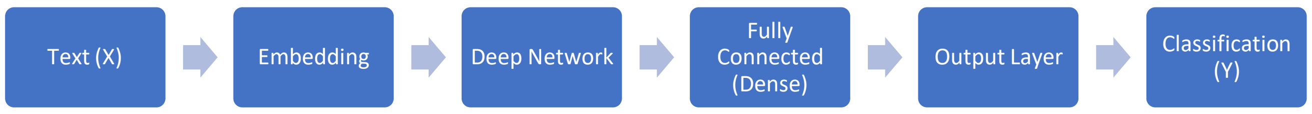


Figure 7 depicts the process of using a deep learning text classification model with embeddings.

When using embeddings for e.g. classification tasks, one can train embeddings on a corpus of choice, or alternatively use a vocabulary with pre-trained embeddings, such as those trained by Mikolov et al. (2013) or Pennington et al. (2014). Pre-trained models are often well-trained and include a broad range of words. Depending on the context of the task at hand, i.e. for example a prediction task where the documents contain “specialized” language, it could however be beneficial to fit a model to more specialized data. For example, in this thesis the documents most certainly include language semantics from financial/legal domains. A model trained on text data that includes such semantics could improve the quality of the embeddings.

In this study, it is not obvious which approach that is the better one. Training a model on the in-house corpus would probably create a more specialized model, whereas pre-trained models available on the Internet are probably very well constructed and include a larger vocabulary. Consequently, we choose to conduct this study on both pre-trained and in-house trained models, to identify which performs the best.

Figure 8 - **Skip-gram and CBOW**

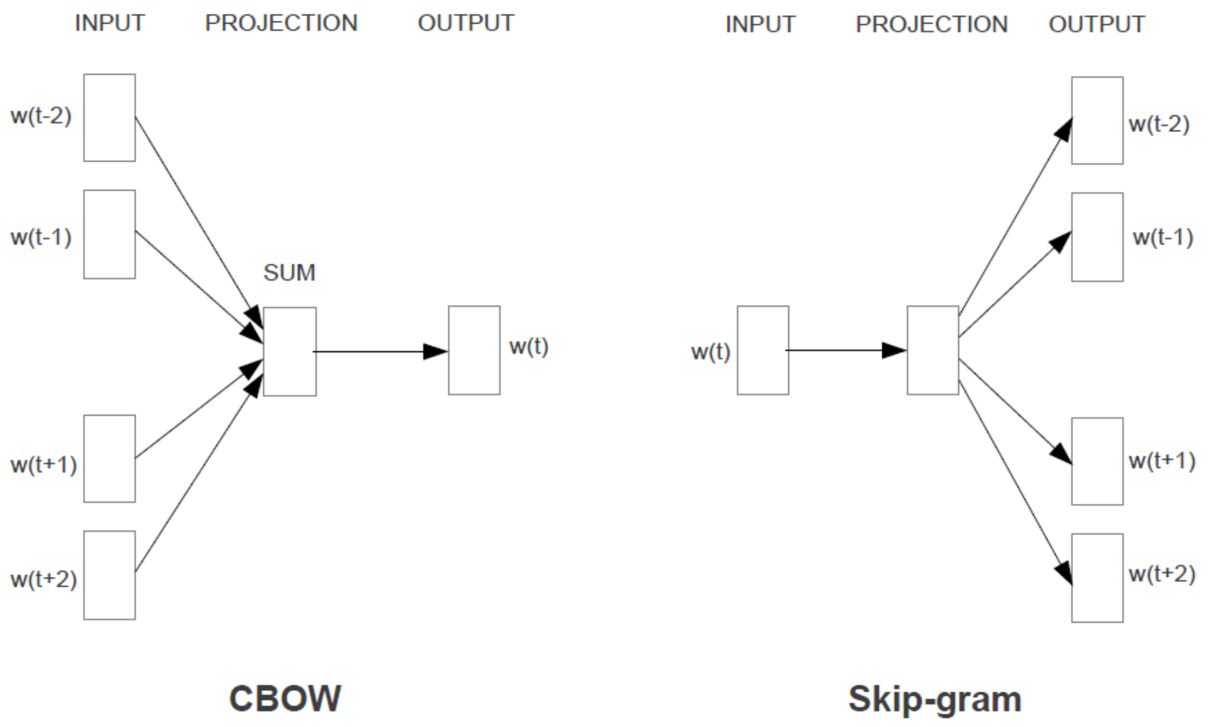
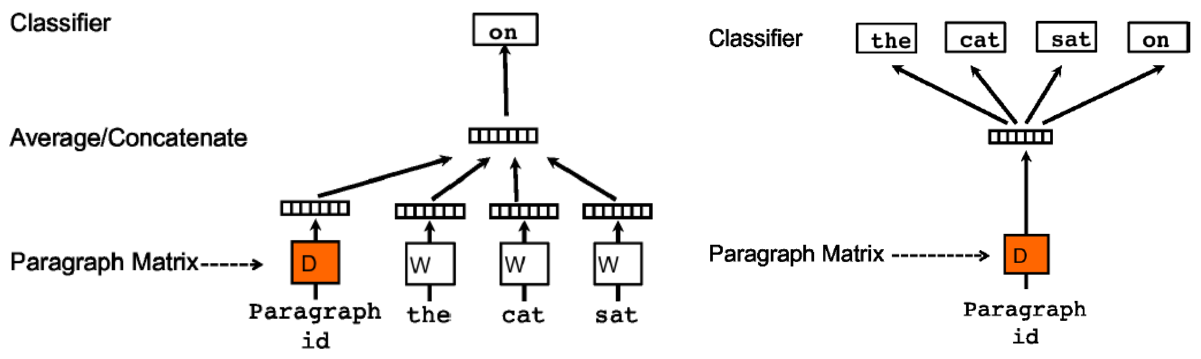


Figure 8 depicts the two word to vector (word2vec) neural networks architectures Continuous Bag of Words (CBOW) and Skip-Grams as described by Mikolov et al. (2013). When training a word2vec model using the CBOW architecture, the model tries to predict the current word based on the context (historic and future words). The Skip-Gram architecture instead tries to predict the surrounding words (historic and future words) from the current word.

Le & Mikolov (2014) improve these respective architectures by also adding paragraph vectors that should improve the model's ability to learn. The first is the so-called Distributed Memory model of Paragraph Vectors (PV-DM), which is similar to the Continuous Bag of Words (CBOW) architecture. In this architecture, the extra paragraph vector can be thought of as another word, which acts as a memory that remembers what is missing from the current context (e.g. sentence, paragraph or document) (Le & Mikolov, 2014). The second is the so-called Distributed Bag of Words model of Paragraph Vectors (PV-DBOW), which is similar to the Skip-Gram architecture. In this architecture, the extra paragraph vector acts as an independent predictor. These two architectures are also known as Document to Vector (Doc2vec).

Figure 9 - **Left: Distributed Memory model of Paragraph Vectors (PV-DM)**

Right: Distributed Bag of Words model of Paragraph Vectors (PV-DBOW)



Illustrations of how the two architectures PV-DM and PV-DBOW contribute to the predictive capabilities of an embeddings model.

3.4.6.1. Embeddings parameters

There are several parameters possible to tweak for the Embeddings, where the most important are mentioned here. The first is the type of architecture, either skip-gram (Word2Vec) and PV-DBOW (Doc2Vec), or CBOW (Word2Vec) and PV-DM (Doc2Vec). The former is often slower but better for infrequent words, whereas the latter often is faster (Google Code Archive, 2013). Second, different algorithms can be used for training the vectors; hierarchical softmax is often better for infrequent words, whereas negative sampling is better with low-dimensional vectors (Google Code Archive, 2013). Third is the dimensionality (size) of the vector space, where more is often better (Google Code Archive, 2013). Last is the context, i.e. the number of historic and future words used for predicting the current word (CBOW/PV-DM), or conversely, the number of historic and future words that the current word is trying to predict (skip-gram/PV-DBOW) (Mikolov, Chen, et al., 2013)

3.5. Machine Learning models

3.5.1. Introduction

After preprocessing the data, it is ready to be used as input for the models that try to learn from the data to classify new data. When constructing a classifier model, the goal is to predict a class out of one of the predefined classes. Classification can be separated into binary and multiclass classification. The Machine Learning methods are used in this context to classify instances (i.e. ad-hoc disclosures) correctly regarding how a stock price has historically reacted to a disclosure. As defined earlier, this is a multiclass classification problem with three classes for the machine learning model to predict. Below, the machine learning classifiers used in the study are presented.

3.5.2. Logistic regression

An area common to apply Machine Learning on is probability estimation, for example estimation of instances belonging to certain classes of interest. This estimate is often utilized in contexts where costs and benefits are relevant factors. In the case of the stock market and the prediction of price movements, a probability for a certain reaction from the stock market to a specified event is interesting as it can generate abnormal returns. By choosing a model with a certain objective function, it is possible to create a model that provides accurate estimates of probability. Logistic Regression is the most common model used for these kinds of problems. (Fawcett & Provost, 2013).

A Logistic Regression is perhaps not the most accurate name for the model as it not really performs what is commonly defined as a regression, which involves estimating a numerical target value. The important distinction between a regression and classification is that in classification the value of the target variable is categorical, compared to a regression where the target variable is continuous. A Logistic Regression produces a numeric estimate, but the values of the target variable are categorical. Hence, it can be viewed as a class probability estimation model and not a regression, even if this can be technically debated. Like Linear Regression and Support Vector Machines, Logistic Regression is also a linear model with class probability estimation, and it is useful in many applications. The basic version of what the model does is fitting a linear model to data to classify instances and what makes it different from e.g. linear regression and Support Vector Machines is the objective function it utilizes. (Fawcett & Provost, 2013).

Answering the question why not a simple basic linear model is enough to estimate class probability leads up to answering what exactly a Logistic Regression is good for. Instances that are to be classified that are farther from the separating boundary have intuitively a higher probability of belonging to one class or another and the function output states the distance that the instance has from the separating boundary. The output of the function states the distance from the separating boundary but the issue that arises is that the function ranges theoretically from ∞ to -∞ while probability estimates range from 0 to 1. So, with the variable of distance from the ratio, the probability, or odds, is calculated and the odds range from 0 to ∞. For instance, an event with the probability of 0.5 of happening has the odds of 50:50 or 1 while an event with the probability 0.9 has the odds 90:10 or 9 (see below table). If the modeling demand is to produce some notion of likelihood instead of class probability specifically, it can be done with log-odds. (Fawcett & Provost, 2013).

Table 4 - **Log-Odds**

Probability	**Odds**	**Log-odds**
0,5	50:50 or 1	0
0,9	90:10 or 9	2,19
0,999	999:1 or 999	6,9
0,01	1:99 or 0,0101	-4,6
0,001	1:999 or 0,001001	-6,9

Table 4 depicts how certain odds are translated into their Log-odds equivalent which is an integral part of the Logistic Regression classifier.

This is what a Logistic Regression does, using the same linear $f(\boldsymbol{x})$ to measure log-odds of a certain event to assign instances to classes. With some algebra the log-odds can be translated into probabilities, but the output of the model is still to be interpreted as the log-odds of an instance belonging to a certain class. However, as the log-odds can be translated into probabilities of class membership, the logistic regression model is simply thought of as a model that calculates specifically that. This can be translated in this study into the probability that a document, i.e. the disclosure, is belonging to one of the ‘Up’, ‘Down’ or ‘Stable’ classes. (Fawcett & Provost, 2013).

Equation 13 - Log odds linear function

$$\log\left(\frac{p_+(\boldsymbol{x})}{1-p_+(\boldsymbol{x})}\right) = f(\boldsymbol{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots$$

$\boldsymbol{f(x)}$ = *The Log odds for an instance to be assigned to a class*
$\boldsymbol{w_0 + w_1x_1 + w_2x_2} + \cdots$ = *The feature vectors in an instance that together provide the function with its Log-odds*
$\boldsymbol{p_+(x)}$ = *the model’s estimate of probability for an instance to belong to a class with data represented by feature vector* ***x.***
1 - $\boldsymbol{p_+(x)}$ = *The probability estimated for the event not occurring given the feature vector*

As aforementioned, the log-odds is often not per se interesting but given that they enable the solving of $p_+(\boldsymbol{x})$ we can with the help of Equation 14 solve it. (Fawcett & Provost, 2013).

Equation 14 - The Logistic function

$$p_+(\boldsymbol{x}) = \frac{1}{1 + e^{-f(\boldsymbol{x})}}$$

$\boldsymbol{p_+(x)}$ = the model's estimate of probability for an instance to belong to a class with data represented by feature vector ***x.***
$\boldsymbol{e^{-f(x)}}$ = The natural logarithm e to the power of the negative f(x)

If the logistic function is plotted, the predicted probability that an instance belongs to a certain class is represented by the S-shaped Sigmoid, where the probability estimation is squeezed into the important range of 0-1 based on the log-odds. (Fawcett & Provost, 2013).

Figure 10 - **Estimate of Class probability as a $\boldsymbol{f(x)}$**

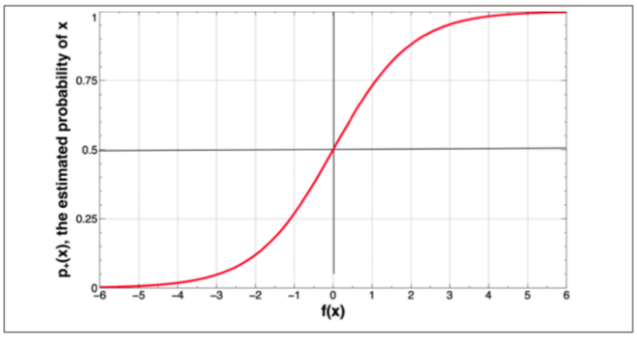


Figure 10 represents the estimate of class probability as a f (x) which in the so called "sigmoid" S-curve squeeze probabilities into the range of 0-1.

3.5.2.1. Multi-class Logistic Regression (MLR)

Since Logistic Regression in its original form performs binary classification tasks, some extensions of the model are needed when applying it to a multiclass problem (Van Leeuwen & Brummer, 2006). With this extension, the multi-class Logistic Regression, MLR, can be trained to discriminate between *N* number of classes (Van Leeuwen & Brummer, 2006), with *N*=3 in this study. The *N* number of classes still get their respective log-odds to discriminate class on which the classifier classifies the instance with the best log-odds (Van Leeuwen & Brummer, 2006). The relative contributions of the *N* classes in the training of the MLR is a question partially regarding weighting to optimize the conditions for the model to become generalizable (Van Leeuwen & Brummer, 2006). The mathematics that add up to the MLR is not the same as a one-vs.-rest approach, but it does result in one vector coefficient and intercepts per class from which the prediction is made (Müller & Guido, 2016).

3.5.2.2. Logistic Regression parameters

Different algorithms (solvers) for solving the optimization problem can be used in Sci-Kit Learn in Python. The default solver per Sci-Kit Learn version 0.21.2 is Liblinear. It treats the problem as binary with a one-vs.-rest approach to classifying instances. In contrast, the setting "SAGA" is an improved version of the SAG (Stochastic Average Gradient) descent for classifying (Defazio et al., 2014). SAGA can use either a one-vs-rest classification scheme, or calculate the multinomial loss when optimizing the cost function (Sci-Kit, 2019a)

The parameter in Logistic Regression that determines the strength of regularization tackling complexity is called C. A higher value of C results in less regularization and vice versa (Müller & Guido, 2016). In Sci-Kit, the penalty setting "L2" corresponds to Ridge Regression, i.e. where a lower C regulates coefficients closer to 0. The penalty "L1" corresponds to Lasso Regression, i.e. where a lower C can set some coefficients to *exactly* 0 (Müller & Guido 2017).

3.5.3. Naïve Bayes Classifier

Provost & Fawcett (2013) encourage Machine Learning strategies for solving a certain problem to use the simplest, i.e. least expensive, technique that works. Naïve Bayes is an especially simple, common and useful Bayesian method. Naïve Bayes is considered to be both efficient and effective why it is a powerful tool in Machine Learning (Fawcett & Provost, 2013). Naïve Bayes is specifically efficient as it looks at each feature individually and collects statistics per-class to learn parameters (Müller & Guido, 2016).

The Naïve Bayes classifier has its roots in Bayes' rule. Bayes' rule states that it is possible to "Compute the probability of our hypothesis *H* given some evidence *E* by instead looking at the probability of the evidence given the hypothesis, as well as the unconditional probabilities of the hypothesis and the evidence”. Bayes’ rule with the important notion of carefully interpreting conditional independencies are a foundation for a big part of advanced data science techniques. (Fawcett & Provost, 2013)

However, to apply Bayes’ rule in data science and Machine Learning, some rewriting of the original rule is needed, together with extended assumptions for a classification task. The assumption of probabilistic independence is therefore taken one step further. The notion of independence states that two events are independent if one does not give information about the other. Naive Bayes instead uses conditional independence, which means that even if we suspect that some evidence to some extent is not be completely independent (for example words in a document), the evidence is treated as independent anyway, since the classifier still works relatively effectively. Why it is called “Naïve” hence derives from the approach that the model uses, where each feature effect on the target variable is modeled independently, thus not considering feature interactions. (Provost & Fawcett, 2013)

Naive Bayes classifies instances by calculating the estimated probability that a given instance belongs to a certain class. The instance is assigned to the class with the highest estimated probability. The estimated probabilities are calculated using training data where the instances have been manually classified. Equation 3 illustrates the Naive Bayes equation. (Provost & Fawcett, 2013). For this study, the three-way classification has the classes of “Up”, “Stable” and “Down” that the NB classifier estimates probabilities for.

Equation 15 - Naive Bayes

$$p(c \mid E) = \frac{p(e_1 | \text{ c}) * p(e_2 | \text{ c}) \ldots \ldots p(e_k | \text{ c}) * p(\text{c})}{P(E)}$$

$\boldsymbol{p(c \mid E)}$ *= The probability of a certain class given the total presented evidence*

In our study, E (Evidence) = all the features (words/characters and N-grams etc depending on the pre-processing) in a specific ad-hoc disclosure that each have a certain probability to occur in a Up, Stable and Down document

$\boldsymbol{p(e_{1\ldots k} | \, c)}$ *= The probability that a certain feature (1st to the kth feature) belongs to one of the classes*

In our study, a feature's total occurrence in the class that is tested to give a probability figure. This is also be affected by e.g. TF-IDF pre-processing as it gives features different weights why it is not equal the exact occurrence of a certain feature in the class.

$\boldsymbol{p(c)}$ *= The probability of the class without any evidence*

i.e. in our study it depends on the weighting of the dataset which depends on if the natural division between classes are withheld or if altered.

$\boldsymbol{P(E)}$ *= The probability of the total evidence*

This probability is very difficult to calculate with text feature data but as the evidences for the three different classes can be compared with only the numerators and with the extended assumptions of feature probabilistic independence, we do not per se need to know this probability.

Calculating P(E) in Equation 15 is tricky and can therefore often, when using Naïve Bayes, be substituted with an easier method. This easier method however requires e_k to be mutually exclusive and exhaustive, i.e. a piece of evidence can only belong to one class. In the case of using word features as evidence, this is not possible since the feature can be present in e.g. both a Down- and Up-classified document. Fortunately, we are only interested in the ordinal scale, i.e. classifying ad-hoc disclosures as either Up, Down or Stable. Therefore, we do not actually need to calculate actual probability estimates, as the probability of the total evidence is the same for a given document no matter the class. Thus, we can classify an ad-hoc disclosure by only calculating the numerator of Equation 15 and then compare it between the three different classes. The class with the highest numerator is assigned to the instance. (Provost & Fawcett, 2013)

Naive Bayes takes all feature evidence into account while not requiring much computation time or storage, which adds to the method's advantages. As text data contains different dependencies among sentences and word combinations, it is a strength that Naive Bayes still can classify reviews effectively. However, everything is not ideal with the Naive Bayes model. A disadvantage with the model when used for classification is that the probability estimates themselves cannot be reliably used. This is because we only choose the class with the greatest probability estimate, which is why we do not need to calculate $P(E)$. In our case this does not create an issue, but in other business cases where probability estimates are used for cost/benefit analyses, this could impose a problem and Naïve Bayes should be used with caution. (Provost & Fawcett, 2016)

3.5.3.1. Naïve Bayes parameters

In text data classification, there are two Naïve Bayes classifiers that are the most used that are in the SciKit library: BernoulliNB and MultinomialNB. BernoulliNB assumes binary data and counts the number of times a specific feature in a class/document is not zero (Müller & Guido, 2016). MultinomialNB assumes count data and uses the average value of each feature per class (Müller & Guido, 2016). Count data thus works if each feature individually represents an integer count, e.g. how often a word appears in a document (e.g. term frequency) (Müller & Guido, 2016).

BernoulliNB and MultinomialNB have a parameter called alpha (a positive number) to control model complexity. The algorithm adds to a word *alpha* many virtual data counts. Since the relative distance between the probabilities of a word belonging to one class rather than the other decreases, it practically works to "smooth" the data statistics. The larger the alpha, the more data points are added, resulting in a smoother and more regularized model. In addition, it provides a probability to words that occur in the vocabulary but is absent in one (or more) classes, thus preventing the whole Naïve Bayes equation (probability of a document being labeled to that class) from becoming 0 (Medium, 2017). The Naive Bayes classifier is robust to the parameter setting of *alpha*, meaning it is often not too critical for model performance, but a slight modification in alpha often slightly improves model performance. (Müller & Guido, 2016)

3.5.4. Support Vector Machines

Fitting linear functions is an important feature of Machine Learning. Similar to Logistic Regression, Support Vector Machines (SVM) is also in its default version a linear model but when provided with an additional feature they both are given the freedom to create more complex, non-linear, functions. The most common techniques based on the fitting of parameters to non-linear, complex functions are SVM and Neural Networks.

A non-linear SVM works as a systematic way of adding complex terms to a linear function to fit it to certain data. SVM have a "kernel function" that maps original features to another, new, feature space which the model is fitted to. The method is often seen as difficult to understand because it "magically" achieves effectiveness which is hard to explain. Using different words for SVM, it is a linear discriminant which as indicated before means that SVM classifies instances based on linear functions. (Müller & Guido, 2016)

In practice, a traditional linear model draws a decision line that is used for classification, i.e. all the observations on one side of the line is assigned a certain class and all the observations on the other side is assigned the other class. In comparison, a linear support vector machines finds the widest bar that can be drawn between the different classes. Consequently, the decision line is in the center of the bar, where the area between the decision line and the edges of the bar is the margin of the model. Not all observations are used to determine the decision line. Used observations are called support vectors. By adding non-linear features to the data, the model can become powerful. Adding more features however increases the demand for computational power. By using the "kernel trick", such extensively high computational power can be avoided. The kernel trick involves computing the distance of the data points for the expanded feature representation without computing the actual expansion. (Müller & Guido, 2016).

3.5.4.1. Multiclass Support Vector Machines

The multiclass SVM has many similarities to multiclass Logistic Regression, but it generally in most implementations also makes use of a kernel matrix which can be advantageous (Van Leeuwen & Brummer, 2006).

3.5.4.2. Support Vector Machines parameters

This study tests three different types of Support Vector Machines. First is the "LinearSVM" class in Sci-Kit Learn, which uses the Liblinear solver to optimize the default squared hinge loss function (Sci-Kit, 2019d). Second is the "SGD Classifier", which instead uses Stochastic Gradient Descent for optimizing the default hinge loss function (Sci-Kit, 2019c). Third is the Support Vector Classifier (SVC), which as default uses the Radial Basis Function, also called the Gaussian Kernel, to optimize a loss functions as implemented by LibSVM (Chang & Lin, 2011; Sci-Kit, 2019e).

The most important parameter for Support Vector Machines is C, which is a regularization parameter that limits the importance of each point in the same way as the same parameter C does in a Logistic Regression. A low value of C creates a restricted model where specific points in the dataset barely have any impact on the model. However, with a larger value of C, the model can bend the decision line to better capture specific points that might have been misclassified. (Müller & Guido, 2016).

The second parameter is gamma, which controls the width of the Gaussian kernel, i.e. at which distance that points are considered close or not in the classification task. A small gamma equals a large radius for the Gaussian kernel, why more points are close to each other. On the other hand, a large gamma means that relatively fewer points are close to each other. The value of gamma impacts in practice the actual decision line, where a small value creates a smoother line and a larger value creates a line that focus more on single points. This parameter is only relevant for the third variant of Support Vector Machines presented above, SV (Müller & Guido, 2016).

3.5.5. BERT

BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) is a Natural Language Processing model developed by Google that in many cases achieves significantly higher results compared to other models. BERT can be used for many Natural Language Processing tasks including multi-classification. The BERT model is an unsupervised and deeply bidirectional system that uses contextual representations with a neural network. Pre-trained representations, as word embeddings, can either be context-free or contextual. A context-free model such as word2vec or GloVe creates a single word embedding representation for each word in the vocabulary. Hence, as some words can have different meanings in different contexts, the representation might not be completely correct. E.g. the word "bank" has the same representation in "bank deposit" and "river bank" even though the context is fundamentally different. However, contextual models instead create representations for each word based on the other words in the same sentence, thus allowing the context of the sentence to affect the embedding representation of the word. Previous models that use contextual representations such as ELMo, ULMFit, Generative Pre-Training and Semi-supervised Sequence Learning are all unidirectional or shallowly bidirectional. Thus, each word is contextualized using only the word to its right or left. However, some models use a combination of the representations from the left and the right but only shallowly. For example, in a bidirectional model, the sentence "I made a bank deposit" for the word "bank" either "I made" or "deposit" would be used. BERT however uses both the left and right context of a sentence and is thus bidirectional. A relatively simple approach is used to train BERT, 15% of the words in the input is masked and is subsequently run through a deep bidirectional Transformer encoder that predicts the masked word. (Devlin et al., 2019, 2018/2020).

Similar to Logistic Regression, BERT calculates the probability for an instance to belong to each class. Subsequently, the class with the highest probability is assigned to the instance. However, this also allows us to set different thresholds of probability depending on class as it might be more important to have a high precision among certain classes compared to others. For example, in our study classifying an instance as “Stable” does not incur any cost nor profit whereas if an instance is classified as “Up” but the stock goes down there are both a cost and a loss. Therefore, it might be valuable to set a higher probability threshold for the classes “Up” and “Down” compared to “Stable”.

Pre-training BERT is quite expensive as a lot of text data is used to create a comprehensive and accurate model. Therefore, there are several pre-trained models for BERT available. The researchers from Google that created BERT have released several pre-trained models that can be used. The English pre-trained version is based on Wikipedia and a dataset called BookCorpus. Additionally, there is a multilingual pre-trained model available from the same researchers at Google that has been trained on Wikipedia for each respective language. The multilingual BERT can handle text data in different languages in the same model. The top 100 languages (with the most articles) from Wikipedia are included in the multilingual version. The multilingual model includes Swedish. (Devlin et al., 2018/2020).

In addition to the pre-trained models published by the creators of BERT, several other pre-trained models are also available from different sources. The National Library of Sweden has published a freely available pre-trained Swedish model for BERT trained on books, news articles, government documents and the Swedish Wikipedia. (National Library of Sweden, 2020).

3.5.5.1. BERT Parameters

The parameter “max_seq_length” for BERT defines the number of tokens for each instance in the dataset. Hence, the instance is truncated if there are more tokens than the max_seq_length parameter. Furthermore, if there are fewer tokens than the max_seq_length in a document, it is padded in order for all instances to have the same number of tokens. (Foong, 2019)

3.5.6. Grid searching

Grid searching is the process of searching for optimal parameter settings in a certain model with a certain task to find the best version of the model possible. For example, with Naive Bayes, grid searching includes iterating over a number of different alpha values to find the best Alpha for model accuracy. (Müller & Guido, 2016)

In practice, grid searching is a concrete way of training and testing a certain model over and over until the best parameter value/parameter value combination are found. With more than one parameter, the grid searching process includes also trying x number of combinations of the parameters, which fast can turn the grid search to a process that demands extensive computer power and/or time. Since the test data is used to tune the parameters, it plays an important part in constructing the model. The model should thus be finally evaluated with unseen data, another set of test data why the former test data is commonly referred to as validation data. (Müller & Guido, 2016)

As such, a dataset should be divided into training, validation and testing data to be able to perform a grid search correctly. The training data trains the model, the validation data is used to tune the parameters through acting as an early stage test data, and finally the test data is the hold-out data that cannot impact the model's settings or other relevant choices, to be used as the best possible final test. (Müller & Guido, 2016)

3.5.7. K-fold Cross Validation

A common way of training and validating a model is to use K-fold Cross Validation, which can be made together with a grid search. The "k-fold" refers to the number of folds that the cross validation consists of for a certain training process. For instance, if a 5-fold cross validation is used, the dataset is randomly divided into five equal-sized parts. Each of the five parts is in the training process part of building the model four times (acting as training data) and one time for evaluating the model (acting as validation data), thus building five models with five different prediction scores. The subsequent "cross-validation accuracy" is the mean accuracy of the five models. (Müller & Guido, 2016)

Advantages of cross validation is that it might be more robust than using only 1-fold test data, as the risk of random bias is reduced. It can also provide more information, for example variance in output. Including more data into training the data can also be beneficial, as a higher number of validations can substitute a lower test allocation (i.e. 10-fold cross-validation means validating 10 times using 10 % of the data each time). According to Müller & Guido (2016), one issue with cross-validation is however that it roughly increases computing time with k times the original, thereby making it more costly to use. (Müller & Guido, 2016)

3.5.8. Evaluation of the machine learning models

3.5.8.1. Accuracy

Accuracy is naturally one of the most important metrics when evaluating a machine learning model. The accuracy is simply the ratio of correct classifications over all instances. The accuracy calculation when evaluating the machine learning model during H1 is depicted in Equation 16, using the terminology utilized in Figure 11.

Equation 16 - Accuracy calculation

$$Accuracy = \frac{TD + TS + TU}{D + S + U}$$

$\boldsymbol{TD}$ = *True Down*
$\boldsymbol{TS}$ = *True Stable*
$\boldsymbol{TU}$ = *True Up*
$\boldsymbol{D}$ = *Down*
$\boldsymbol{S}$ = *Stable*
$\boldsymbol{U}$ = *Up*

3.5.8.2. Baseline

The baseline chosen when evaluating the results and grid searches from the task of learning the machine learning model, is the ZeroR model. This means that the accuracies are compared to the most prevalent class in the dataset. This choice is made because it provides the study with a clean benchmark to maintain consistency between different steps of the experiment adhering to Hypothesis 1, such as (1) choosing labeling procedure, (2) choosing classifier and (3) choosing pre-processing techniques. The choice of the ZeroR model as baseline is the most relevant for the first step, i.e. when choosing a dataset, because the prevalences of the different classes vary between labeling procedures. When the study proceeds to evaluating and comparing classifiers and pre-processing techniques, the focus is naturally switched from the baseline itself, to the most simplistic model. For example, in step 2, when different classifiers are compared, the Logistic Regression with default settings is the model that other classifiers are compared to. Finally, in step 3, TF-IDF with default settings is the simple bag of words technique that embeddings and BERT are compared to. Still, the baseline to which leverage is calculated, is ZeroR.

3.5.8.3. Leverage and Lift

It is possible to compare the accuracy to the baseline using different types of metrics. Two common metrics are Leverage and Lift. Leverage is the difference between the algorithm's accuracy and the baseline. Lift is the ratio between the algorithm's accuracy and the baseline.

Equation 17 - Leverage

$$Leverage = Accuracy - Baseline$$

Equation 18 - Lift

$$Lift = \frac{Accuracy}{Baseline}$$

The different trait that comes with the two metrics is apparent using an example as depicted in Table 5. In the example, two machine learning models are compared, the first having an accuracy of 40 percent and the second having an accuracy of 50 percent. However, they have different baselines: 35 percent and 45 percent respectively. When evaluating the models using Leverage, their performances are judged to be equally good. When evaluating them using Lift, no. 1 is judged to be the better. It is apparent that Lift is "punishing" model no. 2 due to its higher baseline, whereas Leverage does not.

Table 5 - **Illustration of the difference between Leverage and Lift**

Model	**Accuracy**	**Baseline**	**Leverage**	**Lift**
1	40%	35%	5,0	14,3%
2	50%	45%	5,0	11,1%

Table 5 depicts the difference between using leverage and lift when evaluating a Machine Learning model against the baseline.

Which of these that is the better metric depends on the task at hand. In this study, the choice between the two matter when choosing labeling procedure, i.e. which classification threshold and price movement measurement (Returns, Market Adjusted Returns, or Jensen's Alpha) to use when constructing the final machine learning model. This is the case because the most prevalent class is used as baseline in this study, and the class prevalence changes both when different price movement calculations are used and when different thresholds are used.

For the different price movement measurements, a measurement which is prone to calculating smaller price movements should have higher prevalence of the "Stable" class. Since more complex financial models adjust for more risk measured as volatility, these are expected have a higher prevalence of "Stable" instances compared to less complex financial models. Moreover, a higher threshold means that classified instances need a larger price movement to be allocated into the "Up" and "Down" classes, resulting in higher prevalence of the "Stable" class and thereby a higher baseline.

Consequently, it is not obvious if a higher prevalence of "Stable" is better or worse than lower prevalence. Therefore, a labeling procedure with a higher prevalence of "Stable" should not be "punished" as is the case if Lift is used for evaluating. Hence, leverage rather than lift is used for evaluating the performance of the machine learning models in this study.

3.6. Trading strategy using algorithmic trading

To see if the model can generate any monetary value, we construct at trading strategy based on the theoretical parts of Algorithmic Trading previously presented. In the trading strategy our trained machine learning model is used on completely new documents to classify unseen data. For each new release of a corporate disclosure the text is tokenized and classified. A classification to the label "Up" or "Down" triggers an action to buy or short the stock while a classification of "Stable" results in a stay out of market strategy, i.e. not engaging in trading the stock.

For each trade, a holding period must be pre-defined i.e. the time period the trade should be active before realizing the returns. As previously presented, stock market reactions to new information have a time-lag of on average 30 minutes (Muntermann & Guettler, 2007). We base our holding period on the 30 minutes time lag but add another 30 minutes to make sure we capture the entire stock price movement. Thus, our holding period is 60 minutes from the minute the trade is performed. If a corporate disclosure is released when the stock market is closed, we use the first available price that follows the disclosure. If the corporate disclosure is released within 60 minutes of the stock market closing, we close the trade on the last available stock price even though the holding period is less than 60 minutes as we do not want to hold the position over night.

Furthermore, as we are primarily interested in capturing the abnormal returns and thus need to filter out any noise on the market, we apply a so-called market-neutral strategy. In a market neutral strategy, the effect of the market i.e. the systematic risk is cancelled out by also investing in the overall market

(Investopedia, 2020). If the stock is predicted to go up, we also short the index and if the stock is predicted to go down, we buy the index. The returns from each trade are accumulated to see the total returns of the trading strategy.

However, as it is uncertain how the trading strategy will perform and as the trading strategy must run for a longer period of time it is not feasible to apply it in practice in this study. Therefore, we simulate the trading strategy based on historical data as if it would be used as a trading algorithm. If the simulated trading strategy can generate significant abnormal returns, as defined previously, the model can be said to have created monetary value and subsequently also business value. This would encourage the utilization of the model in practice, i.e. actually deploying it on a market with relevant systems and interfaces to the trading platform. Hence, the simulation is meant to function as if it was a trading algorithm with relevant technical limitations a representative trading algorithm can be assumed to have when having specific text inputs while trying to make accurate split-second decisions.

Each trade is evaluated on tick data for stock prices using historical data (2020-01-01 to 2020-03-11). As this is a simulation some assumptions must be made. First, we assume a delay from the publication of an ad-hoc disclosure in order to process the data and initiate the trade. As many trading algorithms can act within milliseconds, but it is uncertain how long the process, as depicted earlier in Figure 2, of gathering the data, process it and perform the trade will take, we try several different time lags (0,1, 1, 5, 10 and 60 seconds) to see how this affects our results. Second, the market neutral strategy is simulated by applying Market Adjusted Returns model and Jensen's Alpha in the context of CAPM. We also calculate the Returns without using a market neutral strategy for comparison.

3.6.1. Confusion matrix

The output of a prediction can be visualized using a confusion matrix which can be interesting for the trading strategy as certain classes suggest different actions in practice. Each column comprises the number of *predicted* values of each class, and each row comprises the number of *true* values of each class. With a three-class problem as in this thesis, the confusion matrix is visualized with three columns and three rows, using the classes "Down", "Stable" and "Up". From the confusion matrix, several metrics can be calculated for evaluating the performance of the classification algorithm. With the results of a confusion matrix, it can also be discussed which types of false predictions might be more costly than other, which is highly relevant for an actual trading strategy.

Figure 11 - **Example of a confusion matrix using the labels 'Up', 'Down' and 'Stable'**

Actual / Predicted	Down (Predicted)	Stable (Predicted)	Up (Predicted
Down (Actual)	True Down (TD)	False Stable (FS)	False Up (FU)
Stable (Actual)	False Down (FD)	True Stable (TS)	False Up (FU)
Up (Actual)	False Down (FD)	False Stable (FS)	True Up (TU)

Figure 11 depicts a confusion matrix with the classes 'Up', 'Down' and 'Stable' to illustrate the different results from a classification. The confusion matrix can be used in a cost-benefit analysis and subsequently fine-tune a model.

3.6.2. Precision

Based on what can be observed in the confusion matrix, *Precision* is the accuracy of cases pertaining to a class. i.e., the number of correct classifications out of all of instances that are predicted to a class. Precision is the same calculation as accuracy, but over one (or multiple) classes; if the precision over all three classes is calculated, the calculation is identical to that of the accuracy.

When using the machine learning model as a trading strategy, no action is taken on instances predicted as "Stable". Such instances therefore come with no transaction cost, but also no expected return which is hence risk free for an investor. In contrast, “Up” and “Down” predictions come with certain transaction cost and expected returns. “False Up” and “False Down” predictions are therefore costly, but “False Stable” would not be. Consequently, a high precision of “Up” and “Down” is desirable. To measure this, one can use the confusion matrix to calculate the precision over these two classes, as depicted in Figure 11. Pushing errors towards false stables can therefore be motivated from an economical point of view and similar accuracies with more of those errors compared to costly errors would be preferable. This can for instance be achieved through altering parameters in some of the machine learning models that regard probability thresholds for classifying instances.

Equation 19 - Precision as calculated over the “Up” and “Down” classes

$$Precision_{Up+Down} = \frac{TU + TD}{U + D}$$

TU *= True Up*
TD *= True Down*
U *= Total Up*
D *= Total Down*

As explained in the previous section, a higher threshold should result in higher expected return per trade but fewer trades, resulting in a blurred optimum. Assuming that “True Up" and “True Down” predictions are true on the market, a precision of “Up” and “Down” above 50 percent should result in positive returns over time. This number may however be lower or higher depending on the variance and how many instances that should belong to the ‘Stable’ class with close to zero abnormal returns. Nevertheless, the algorithm’s inability to correctly infer the effect of an ad-hoc disclosure onto stock price returns might result in that the actual returns differ from what was expected from the model. Also, the weaknesses of the financial model used for measuring the returns generated by the algorithm (abnormal returns) might simulate an effect that is not true in practice. As this regard a fair share of uncertainty, it is time to move on to how the results of the trading strategy are evaluated.

3.6.3. Evaluation of the trading strategy

The domain-specific simulation based on performance through a trading strategy is evaluated somewhat differently compared to the machine learning model as the simulations results have a monetary impact among other relevant factors.

3.6.3.1. Abnormal returns

To measure and evaluate if the model succeeded in creating any monetary value, we use the same three methods for labeling to calculate returns. We apply the three methods (1) Returns (2) Market Adjusted Returns (3) Jensen's Alpha to the method Return Across Time and Security (RATS) (Ibbotson, 1975) RATS is a relatively simple method for measuring returns in the financial field without constructing reference portfolios and is basically a linear regression comprising of each trade where the alpha variable is the average abnormal return. We use the output from the regression to perform a one-sided T-test to see if the abnormal returns from the regression are statistically significantly greater than zero. We use a 5 percent significance threshold in the T-test which requires the T- value of alpha to be greater than 1,645 to be classified as statistically significantly larger than zero and thus conclude that the trading strategy did earn money.

Equation 20 - Return across time and security with Returns

$$R_t = \alpha$$

$\boldsymbol{R_t}$ = *Returns*

$\boldsymbol{\alpha}$ = *Abnormal returns*

Equation 21 - Return across time and security with Market Adjusted Returns

$$(R_t - R_{ft}) - (R_{mt} - R_{ft}) = \alpha$$

$(\boldsymbol{R_t - R_{ft}})$ = *Returns adjusted for the risk-free rate of returns*

$(\boldsymbol{R_{mt} - R_{ft}})$ = *Market returns adjusted for the risk-free rate of returns*

$\boldsymbol{\alpha}$ = *Abnormal returns*

Equation 22 - Return across time and security using Jensen's Alpha in the context of CAPM

$$\left(R_t - R_{ft}\right) = \alpha + \beta\left(R_{mt} - R_{ft}\right)$$

$\boldsymbol{\left(R_t - R_{ft}\right)}$ = *Returns adjusted for the risk-free rate of returns*

$\boldsymbol{\beta\left(R_{mt} - R_{ft}\right)}$ = *Capital Asset Pricing Model (CAPM)*

$\boldsymbol{\alpha}$ = *Abnormal returns*

Equation 23 - T statistic for all Abnormal Returns (alphas and Jensen's Alpha)

$$T = \frac{\bar{\alpha} * \sqrt{n}}{s}$$

$\boldsymbol{\bar{\alpha}}$ = *the mean of alpha for the three computation methods respectively*

$\boldsymbol{\sqrt{n}}$ = *Square root of the number of* n *observations*

$\boldsymbol{s}$ = *standard deviation of the Alphas*

4. Experiment H1 – Constructing the machine learning model

This part of the study concerns the practical measures based in the methodology sections taken to be able to answer Hypothesis 1.

H1: *A machine learning model built on ad-hoc disclosures from the Swedish market can learn from the data and classify unseen disclosures better than relevant baselines.*

4.1. Introduction

The steps taken to test H1 are presented in the following five sections.

1. **Creating the dataset -** Steps for collecting and matching relevant data are presented. This includes scraping and downloading data from different websites and databases, and subsequently matching relevant data points with each other to form a two-dimensional data frame.
2. **Labeling** - The steps for how the labeling ("Up", "Stable" or "Down") is conducted are explained. This includes comparing three methods for calculating price movements as an effect of an ad-hoc disclosure, and to compare different cut-off thresholds for the transformation of continuous price movements into classes. The subsequent analysis leads to the choice of labeling procedure for continuing the study. One labeling procedure is defined as one (1) price movement measurement approach and one (1) classification cut-off threshold.
3. **Comparing classifiers** - Different machine learning classifiers are compared through being applied onto the dataset with the chosen labeling procedure.
4. **Comparing pre-processing techniques -** Different pre-processing techniques are compared.
5. **Final model** - the best machine learning model is evaluated, thus answering Hypothesis 1. The model is trained using the labeling procedure (dataset) determined on in section two, the best-performing classifier from section three, and the best-performing pre-processing technique from section four.

Throughout the experiment, Python is used to gather and pre-process text data as well as to train and evaluate the Machine Learning models. The code used in this study is available on GitHub[4]

4.2. Creating the dataset

This section depicts how documents and stock data are downloaded and subsequently merged, thus creating a dataset in a two-dimensional format, where each row consists of one event/instance (ad-hoc disclosure) and each column represents a feature.

4.2.1. Collecting stock market data

Daily closing prices between 2010-01-01 and 2020-03-11 for each company currently listed on Nasdaq Stockholm is exported from Refinitiv Eikon (formerly Thomson Reuters Eikon). Due to limited time with access to Eikon, we only manage to export stock market data for currently (as of 2020-03-11) listed companies. The downloaded stock data comes in the form of stock time series. Metadata in the form of Reuters Instrument Code (RIC) is also available. RIC is a text ID of a stock type used in Eikon to categorize stocks and other financial instruments. It is closely related, but not identical, to stock tickers.

Additionally, we export daily closing prices for the market index OMXS GI between 2010-01-01 and 2020-03-11. This data is collected from Nasdaq OMX. OMXS GI is a value weighted price index, adjusted for dividends, that covers all stocks listed on Nasdaq OMX. This market index is used as the estimated normal return when calculating the Market Adjusted Returns. The market returns are also used when calculating Beta for measuring abnormal returns as Jensen's Alpha.

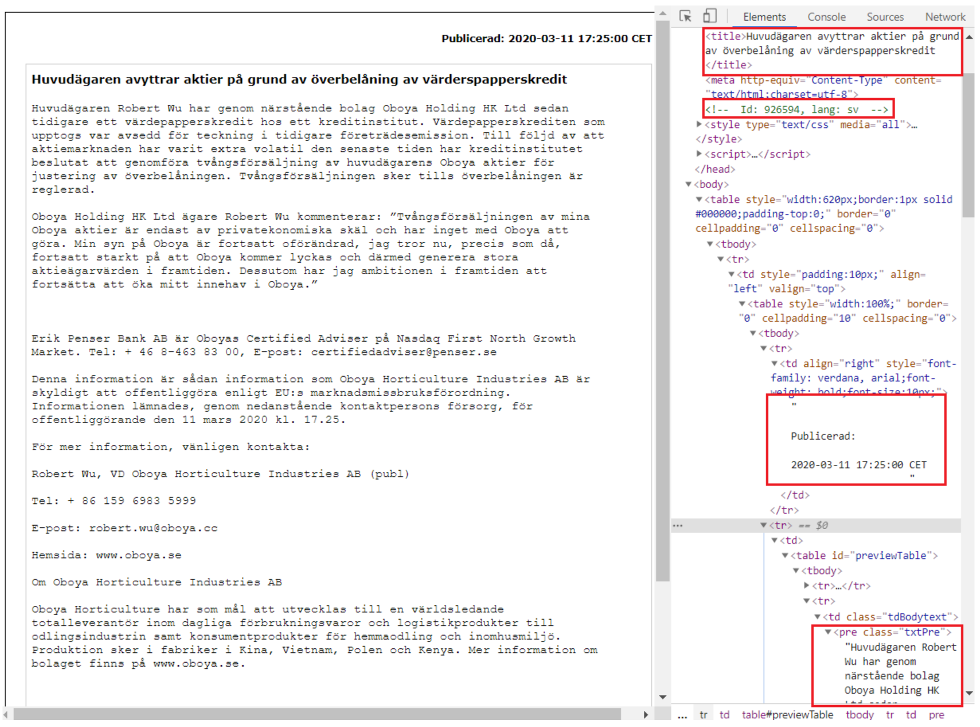
We also download data of the rate of 3-month Swedish Treasury bills from the Swedish Central Bank (Riksbanken). This is used as the risk-free rate when measuring abnormal returns using Jensen's Alpha.

[4] https://github.com/AlgoTrading2020/Thesis2020

4.2.2. Collecting Documents from Nasdaq OMX

The ad-hoc disclosures (documents) are scraped from the website of the owner of the OMX Stockholm stock exchange, Nasdaq Inc. Each document is published on the domain *newsclient.omxgroup.com*. Each document has a URL comprising the domain and a unique ID. The information needed for this study is embedded in the source code of the webpage. Scraped information includes (1) document body text, (2) document headline, (3) disclosure timestamp, (4) document language and (5) document ID. A screenshot of an example document is depicted in Figure 12. To the left, the document in its original format is visible. To the right, the content of the document is illustrated as embedded in the source code, including the title text, timestamp and parts of the body text. In addition, a commented section comprising the document ID and the language stamp is visible on the source code side.

Figure 12 - **Example of scraped ad-hoc disclosure**



Publicerad: 2020-03-11 17:25:00 CET

Huvudägaren avyttrar aktier på grund av överbelåning av värderspapperskredit

```
Huvudägaren Robert Wu har genom närstående bolag Oboya Holding HK Ltd sedan
tidigare ett värdepapperskredit hos ett kreditinstitut. Värdepapperskrediten som
upptogs var avsedd för teckning i tidigare företrädesemission. Till följd av att
aktiemarknaden har varit extra volatil den senaste tiden har kreditinstitutet
beslutat att genomföra tvångsförsäljning av huvudägarens Oboya aktier för
justering av överbelåningen. Tvångsförsäljningen sker tills överbelåningen är
reglerad.

Oboya Holding HK Ltd ägare Robert Wu kommenterar: "Tvångsförsäljningen av mina
Oboya aktier är endast av privatekonomiska skäl och har inget med Oboya att
göra. Min syn på Oboya är fortsatt oförändrad, jag tror nu, precis som då,
fortsatt starkt på att Oboya kommer lyckas och därmed generera stora
aktieägarvärden i framtiden. Dessutom har jag ambitionen i framtiden att
fortsätta att öka mitt innehav i Oboya."



Erik Penser Bank AB är Oboyas Certified Adviser på Nasdaq First North Growth
Market. Tel: + 46 8-463 83 00, E-post: certifiedadviser@penser.se

Denna information är sådan information som Oboya Horticulture Industries AB är
skyldigt att offentliggöra enligt EU:s marknadsmissbruksförordning.
Informationen lämnades, genom nedanstående kontaktpersons försorg, för
offentliggörande den 11 mars 2020 kl. 17.25.

För mer information, vänligen kontakta:

Robert Wu, VD Oboya Horticulture Industries AB (publ)

Tel: + 86 159 6983 5999

E-post: robert.wu@oboya.cc

Hemsida: www.oboya.se

Om Oboya Horticulture Industries AB

Oboya Horticulture har som mål att utvecklas till en världsledande
totalleverantör inom dagliga förbrukningsvaror och logistikprodukter till
odlingsindustrin samt konsumentprodukter för hemmaodling och inomhusmiljö.
Produktion sker i fabriker i Kina, Vietnam, Polen och Kenya. Mer information om
bolaget finns på www.oboya.se.
```

```
<title>Huvudägaren avyttrar aktier på grund
av överbelåning av värderspapperskredit
</title>
<meta http-equiv="Content-Type" content=
"text/html;charset=utf-8">
<!--  Id: 926594, lang: sv  -->
▶<style type="text/css" media="all">…
</style>
▶<script>…</script>
</head>
▼<body>
 ▼<table style="width:620px;border:1px solid
 #000000;padding-top:0;" border="0"
 cellpadding="0" cellspacing="0">
   ▼<tbody>
     ▼<tr>
       ▼<td style="padding:10px;" align=
       "left" valign="top">
         ▼<table style="width:100%;" border=
         "0" cellpadding="10" cellspacing="0">
           ▼<tbody>
             ▼<tr>
               ▼<td align="right" style="font-
               family: verdana, arial;font-
               weight: bold;font-size:10px;">
                 "

                 Publicerad:

                 2020-03-11 17:25:00 CET
                                       "
               </td>
             </tr>
···          ▼<tr> == $0
               ▼<td>
                 ▼<table id="previewTable">
                   ▼<tbody>
                     ▶<tr>…</tr>
                     ▼<tr>
                       ▼<td class="tdBodytext">
                         ▼<pre class="txtPre">
                           "Huvudägaren Robert
                           Wu har genom
                           närstående bolag
                           Oboya Holding HK
```

… tr td table#previewTable tbody tr td pre

Figure 12. Screenshot downloaded on 2020-03-30. Depicts the scraped data as it exists in the source code URL: [*https://newsclient.omxgroup.com/cdsPublic/viewDisclosure.action?disclosureId=926594*](https://newsclient.omxgroup.com/cdsPublic/viewDisclosure.action?disclosureId=926594)

4.2.2.1. Scraping script

The document scraping script is written in Python, using Anaconda and Jupiter Notebook. To simplify the code writing, several Python libraries are used. The most notable libraries are BeautifulSoup (reading, downloading, and cleaning relevant information embedded in the HTML code) and Pandas (processing and exporting the ready data). The script approaches the scraping through looping (Python "while" loop), where one cycle targets one URL and scrapes the content of the document on the resulting web page. Each loop ends by decreasing the ID with 1, thus moving "backwards" in ID numbers and thus also in time. The ID of the first document that is scraped is 926594, which is a document disclosed on 2020-03-11 at 17:25:00 CET. The last document included in the study has ID 379921, which is a document released on 2010-01-04 at 08:30:00 CET.

This study only includes Swedish companies. An additional delimitation has been made to only include companies listed on OMX Stockholm. Our approach to scraping documents by decrementing the document ID included in the URL does not differentiate between disclosures on different stock markets and in different languages; all documents uploaded to Nasdaq OMX Group are iterated over. To avoid having to scrape all disclosures published by Nasdaq (which would take an unnecessary long time, and require post-download filtering of relevant documents), we narrow the scraping down to documents that are written in Swedish, using the language stamp (e.g. "sv" for Swedish and "en" for English) embedded in the source code of each webpage. All disclosures made by companies on OMX Stockholm are released in Swedish and in English, where documents from foreign exchanges also are published in English. For the sake of this study, it does however not matter if we investigate Swedish or English documents if we stick to one of the two. When the loop encounters a document with another language stamp than Swedish, it decrements the ID and continues the loop, i.e. moves on to the next document. By scraping only documents written in Swedish, we also include documents from companies listed on the Finnish branch of Nasdaq and on other Swedish stock exchanges such as First North. These are removed in subsequent steps. The total number of documents scraped are 88 804. The exact procedure of the web scraping is visible in the code of the script.

4.2.2.2. Storage format

We start the study by saving the scraped document data to a flat-file format known as CSV (Comma Separated Values). After having downloaded and exported several thousands of rows, the CSV file could no longer be loaded correctly; the error message said that the rows had a varying number of columns. A possible cause of this is that CSV uses one separator of its columns (e.g. comma, semi-colon, etc). Such symbols often occur in the text information used in this study, which could be the reason for the erroneous loading of the file. From this reasoning, we instead try to save the data in JSON format. JSON uses braces, brackets, and quotations to *enclose* columns (Nugroho, 2019) in contrast to CSV that only *separates* columns. When testing, JSON indeed proves to ensure information consistency and correct loading of exported files. Hence, JSON is used as storage format for scraped documents.

4.2.3. Matching scraped documents with stock data

Each document needs to be matched with its corresponding stock price movement. The used time is the stamp visible on the document. In the scraped data, only the ID, language, timestamp, headline, and text are available; the name of the company is missing. The company name is however needed for matching the disclosure with the corresponding stock price movement. To solve this problem, we create a matching chain for the two datasets (documents and stock prices) that can be described in three distinct steps. To carry this out, we use Python and several Python libraries, most notably Pandas.

4.2.3.1. Matching documents with metadata

First, the scraped documents dataset is matched with additional metadata about each document. The sought information is the company name of each disclosure. Metadata is available on Nasdaq OMX website[5] which is the website where each document can be manually reached using a web browser. The metadata is scraped using the Google Chrome extension Web Scraper.

The metadata also includes timestamp and document headline, which are identical to the timestamp and headline for each document in the document dataset. Hence, timestamp and headline can be used for matching each document with its company name. Using only one of these two as matching key proves not to be strong enough: we find several examples of disclosures published on the same date and time by different companies (such as 2019-10-25 08:30:00). There are also disclosures with identical headlines that are published by different companies (such as "Quarterly Report"). Using a combination of the two however proves to be a strong matching key. We have not found any example where two different companies have published a disclosure using identical time stamps and headlines. Still, it is possible that there are a few such examples in the dataset, which is a potential weakness of this study. The fact that we have not found any examples of this, and due to the sheer number of documents used in this study, the effect of such incorrect matches onto the results of this study is deemed relatively small.

Due to the functionality of the Google Chrome Web Scraper tool, we (1) must start the tool so that it scrapes all documents present on the website (2) pause it and filter on relevant rows (such as disclosures by companies listed on OMX Stockholm), and (3) un-pause it. This results in a dataset which, in addition to OMX Stockholm documents, also contains some documents from other Nordic

[5] http://www.nasdaqomxnordic.com/nyheter/foretagsmeddelanden,

stock exchanges. Also, most companies in the dataset publish disclosures both in the local language and in English. The tool for some reason also creates duplicate rows of many disclosures. After dropping duplicate rows, the dataset consists of 67 521 rows. Since the metadata dataset contains multiple languages, it feels plausible that it would contain a higher number of rows than the document dataset. This is however not the case (67 521 < 88 804). When investigating, we find that all rows in the document dataset are not represented; metadata is lacking. After matching the documents dataset with the metadata dataset using timestamp + headline, the resulting rows are still 34 679. We deem this number to be large enough for fulfilling the purpose of this study, why the lack of metadata (which results in a large drop in the number of rows) is not investigated further.

4.2.3.2. Matching documents with RIC

Second, the extended document dataset is matched with a dataset containing RIC number, using the company name that was acquired in the previous step. This file is downloaded from Refinitiv Eikon and contains all stocks listed on OMX Stockholm per 2020-03-11, a total of 383. A screenshot of this dataset is available in the appendices.

If a company has more than one stock (e.g. Bonava A and Bonava B), rows pertaining to this company would occur twice in the matched dataset. This would increase the weight of disclosures made by these companies compared to disclosures made by other companies for learning the machine algorithm. Hence, we remove “duplicate” stocks; we choose to keep stocks with the highest trading volume. Trading volume is an indication of higher liquidity (easier to buy or sell a stock at any given time). Since the construction of the machine learning model in this study is based on the reactions to ad-hoc disclosures of market agents, using stocks with higher turnover could potentially provide the model with more immediate behaviors to disclosures. Immediate behaviors could be more relevant for the model to learn, since the strategy of the subsequent trading algorithm is to be faster than the average market agent, to enable generating profit. When removing “duplicate” stocks, the resulting number is 338 in the RIC number dataset.

Some company names are not written identically by Nasdaq and Eikon. For example, Nasdaq uses the full name of the Swedish company “Fastighets AB Balder”, whereas Eikon abbreviates the name to “Fast. Balder”. To match successfully, both company names are pre-processed/normalized and, especially those from Eikon, are truncated to enable matching on partial string with the company name present in the document dataset. In this example, Nasdaq’s “Fastighets AB Balder” is transformed into “fastighets ab balder”, and Eikon’s “Fast. Balder” is transformed into “balder”. The Eikon content *balder* is then matched with the Nasdaq content on partial string (fastighets ab *balder*).

The matched document contains 28 211 instances. The document previously contained 34 679 instances, which included disclosures made in Swedish from all Nordic OMX Group stock exchanges. The number of unique company names in the previous dataset was 496. Now, the number of companies are 323. Hence, the reduction in rows feels plausible. Since the number of companies in the resulting dataset are 323 out of a total of 338 that we would have expected, it seems that 15 OMX Stockholm companies have been dropped from the lack of meta data. Since the dropout is relatively small, we deem 323 companies enough for the purpose of this study.

4.2.3.3. Matching documents with stock prices

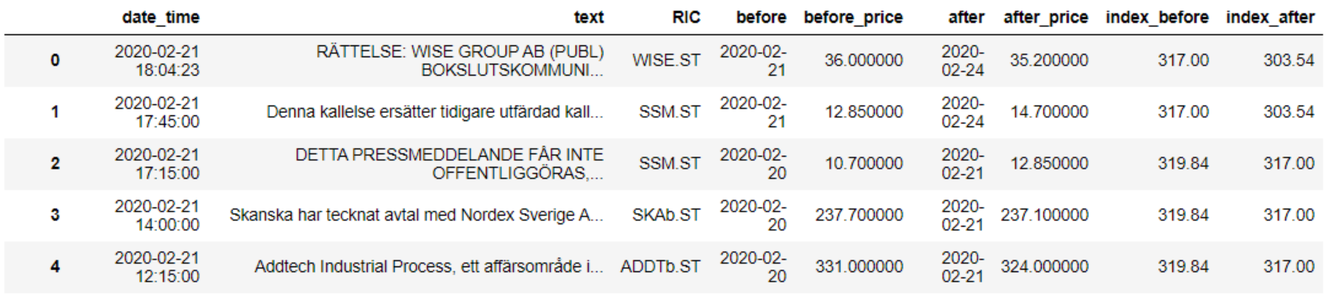
Third, we can combine the RIC number of each document with its timestamp to import the prices of a corresponding stock before and after the disclosure. Available data is (1) date and (2) closing price. Exact time is not available. In general, OMX Stockholm is only open on weekdays and closes at 17.30 CET. If a weekday occurs the day before a holiday, OMX Stockholm might close earlier. In Sweden, holidays are related to different types of events (e.g. cultural and religious) and are for example contingent on dates (such as Christmas day on December 25th) or weekdays (such as Midsummer

Day on the first Saturday after the summer solstice). To avoid having to spend time conforming to holidays and the corresponding closing time of OMX Stockholm for the 10-year period used in this study, we for simplicity reasons assume that the closing time of OMX Stockholm always is 17:30.

When calculating Returns, the first price that occurs after the ad-hoc disclosure is used as numerator, and the most recent price that occurred before the release is used as denominator. Since we use daily data in this study, these are always prices as they were at 17.30 each day. If a press release occurs at 17:30:00 exactly, the 17:30:00 closing price is regarded as "before price", and the closing price occurring 24 hours later is regarded as "after price".

The final number of rows in this dataset is 27 302. The loss of 909 rows is due to that we drop rows with null values. The occurrence of null values when matching with stock prices is due to two reasons. The first is that a disclosure made by a company on the first date of the available time series (i.e. on 2010-01-01, or on the date that a stock was listed onto OMX Stockholm) does not have a corresponding "before price" if it was made at or before 17:30:00. The second is that some companies included in this study have previously been listed on other stock exchanges than OMX Stockholm but were re-listed onto OMX Stockholm later. For example, there are examples in the dataset of disclosures that were made by a company when it was listed on Nasdaq First North. The stock data downloaded from Eikon only includes prices from the OMX Stockholm period of a stock. Hence, such disclosures lack corresponding stock prices on OMX Stockholm, but the disclosures themselves have "survived" the matching process so far due to having corresponding company names and RIC. Figure 13 depicts five example rows of the final, merged dataset.

Figure 13 - **Example of instances in the merged dataset**



	date_time	text	RIC	before	before_price	after	after_price	index_before	index_after
0	2020-02-21 18:04:23	RÄTTELSE: WISE GROUP AB (PUBL) BOKSLUTSKOMMUNI...	WISE.ST	2020-02-21	36.000000	2020-02-24	35.200000	317.00	303.54
1	2020-02-21 17:45:00	Denna kallelse ersätter tidigare utfärdad kall...	SSM.ST	2020-02-21	12.850000	2020-02-24	14.700000	317.00	303.54
2	2020-02-21 17:15:00	DETTA PRESSMEDDELANDE FÅR INTE OFFENTLIGGÖRAS,...	SSM.ST	2020-02-20	10.700000	2020-02-21	12.850000	319.84	317.00
3	2020-02-21 14:00:00	Skanska har tecknat avtal med Nordex Sverige A...	SKAb.ST	2020-02-20	237.700000	2020-02-21	237.100000	319.84	317.00
4	2020-02-21 12:15:00	Addtech Industrial Process, ett affärsområde i...	ADDTb.ST	2020-02-20	331.000000	2020-02-21	324.000000	319.84	317.00

Figure 13 depicts an example of instances from the merged dataset with all relevant data to calculate returns that subsequently is used in the labeling process. The dataset includes timestamp of the disclosure (date_time), disclosure (text), RIC, and stock prices of the respective "before" and "after" dates, assumed to be at 17.30 CET each day. The picture also includes index prices (OMXSGI) of the dates, also at 17.30 CET.

4.3. Labeling

This section describes how each row in the dataset is labeled either "Up", "Stable" or "Down". The label is a consequence of (1) how the price movement of a stock is calculated and (2) the cut-off threshold used for translating a continuous price movement into a categorical "Up", "Stable" or "Down".

4.3.1. Price Movement Measurements

The three methods for measuring price movements are defined and described in the method section. Returns (Equation 2), Market Adjusted Returns (Equation 3) and Jensen's Alpha in the context of CAPM (Equations 4 & 5) are the three methods used for measuring stock a price movement occurring as an effect of an ad-hoc disclosure.

4.3.2. Thresholds

When the price movement has been calculated, a cut-off threshold needs to be used for categorizing the price movement as either "Up", "Stable" or "Down". If the movement is above the positive value of the threshold, "Up" is assigned. If it is below the negative value of the threshold, "Down" is assigned. If it is between the positive and negative threshold values, "Stable" is assigned. The labeling of an instance is expressed in the Conditional Expressions X-Z, of which one is always True and two are always False for all values of Price Movement (PM), using a pre-defined Threshold value (T).

Expression X: $PM > T \rightarrow Up$

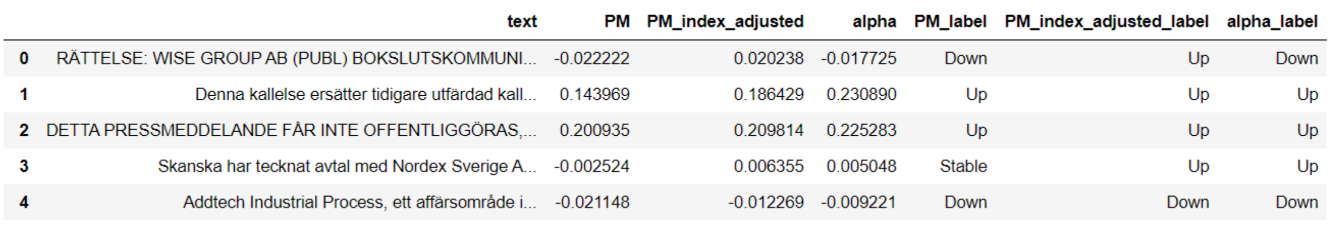
Expression Y: $T \geq \ PM \geq (-T) \rightarrow Stable$

Expression Z: $(-T) > PM \rightarrow Down$

For example, if a threshold (T) is set to 2%, and a disclosure resulted in a 1,85% stock price movement (PM), the movement is between +2% and -2%. It is thus True for Expression Y, and False for Expressions X and Z. It would thus be labeled “Stable”.

To test which threshold is optimal for the documents used in this study, we choose to compare different thresholds. What threshold that is optimal is evaluated together with the price movement measurement, presented in the next section. Figures 14 and 15 depict five sample rows using 0,5 percent and 4,5 percent cut-off thresholds respectively. “PM” is Returns, “PM_index_adjusted” is the Market Adjusted Returns, and “alpha” is Jensen’s Alpha. The “_label” abbreviation is the label corresponding to the particular measurement.

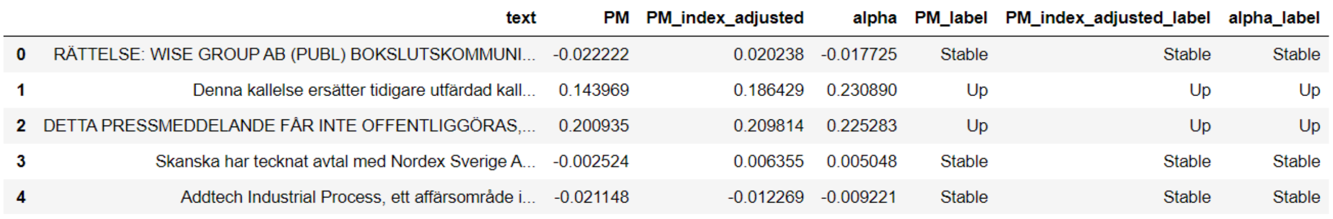
Figure 14 - **Example of labeled rows when the 0,5 percent cut-off threshold is used.**



	text	PM	PM_index_adjusted	alpha	PM_label	PM_index_adjusted_label	alpha_label
0	RÄTTELSE: WISE GROUP AB (PUBL) BOKSLUTSKOMMUNI...	-0.022222	0.020238	-0.017725	Down	Up	Down
1	Denna kallelse ersätter tidigare utfärdad kall...	0.143969	0.186429	0.230890	Up	Up	Up
2	DETTA PRESSMEDDELANDE FÅR INTE OFFENTLIGGÖRAS,...	0.200935	0.209814	0.225283	Up	Up	Up
3	Skanska har tecknat avtal med Nordex Sverige A...	-0.002524	0.006355	0.005048	Stable	Up	Up
4	Addtech Industrial Process, ett affärsområde i...	-0.021148	-0.012269	-0.009221	Down	Down	Down

Figure 14 shows an example of instances labeled with the different methods (Returns, Market Adjusted Returns and Jensen’s Alpha) with a stock price movement threshold of 0,5 percent.

Figure 15 - **Example of labeled rows using the 4,5 percent cut-off threshold is used.**



	text	PM	PM_index_adjusted	alpha	PM_label	PM_index_adjusted_label	alpha_label
0	RÄTTELSE: WISE GROUP AB (PUBL) BOKSLUTSKOMMUNI...	-0.022222	0.020238	-0.017725	Stable	Stable	Stable
1	Denna kallelse ersätter tidigare utfärdad kall...	0.143969	0.186429	0.230890	Up	Up	Up
2	DETTA PRESSMEDDELANDE FÅR INTE OFFENTLIGGÖRAS,...	0.200935	0.209814	0.225283	Up	Up	Up
3	Skanska har tecknat avtal med Nordex Sverige A...	-0.002524	0.006355	0.005048	Stable	Stable	Stable
4	Addtech Industrial Process, ett affärsområde i...	-0.021148	-0.012269	-0.009221	Stable	Stable	Stable

Figure 15 shows an example of instances labeled with the different methods (Returns, Market Adjusted Returns and Jensen's Alpha) with a stock price movement threshold of 4,5 percent.

4.3.3. Comparing Labeling Procedures

The construction of the final machine learning model is trained using only one "labeling procedure", one pre-processing approach and one classifier. Due to time constraints, there is not enough time to test all possible combinations. We therefore need to narrow possible combinations down as we go about the study.

One "labeling procedure" consists of one (1) type of Price Movement calculation and one (1) threshold. For example, Price Movement defined as Jensen's Alpha, using the 2 percent cut-off threshold, is regarded as one "labeling procedure". Currently, there are three different ways to calculating price movement and nine different thresholds to adhere categorizing price movements to, resulting in 27 different labeling procedures. Essentially, each labeling procedure is its own dataset. To be able to build machine learning models, it is therefore necessary to first determine which labeling procedure (dataset) to proceed the study with. The choice of dataset can in this context be considered a hyperparameter to decide on, similar to other hyperparameters tuned in subsequent sections.

We test all 27 combinations by applying one pre-processing type and one classifier. The chosen pre-processing type is TF-IDF and the chosen classifier is Logistic Regression, both with default settings as of the Python library Sci-Kit Learn version 0.21.2. TF-IDF and Logistic Regression are chosen because they are common and well-renowned techniques for text classification tasks. For the TF-IDF, default parameters include the word analyzer using unigrams without any restriction on maximum or minimum document frequency. For Logistic Regression, default parameters include using the Liblinear solver with a one-vs-rest classification scheme, L2 regularization and the regularization parameter C = 1.

Before the test is conducted, each dataset is stripped of instances occurring from 2020-01-01 onward. These instances are used in the last step of this study, which is to simulate if we can make money from using the final algorithm for trading on OMX Stockholm. The dataset used for this (and subsequent machine learning model training steps) thus consists of 26 841 instances. In addition, we perform an 80/20 train_test_split on each of the 27 labeling procedures. The 80% are used for grid searching hyperparameters using cross-validation, and the 20% are used as hold-out data for the final evaluation of the machine learning model and is therefore not used in this nor subsequent validation steps.

Tables 6, 7 and 8 contain the results from testing the labeling procedures. Table 6 consists of the results from testing Returns, Table 7 of Market Adjusted Returns and Table 8 of abnormal returns as Jensen's Alpha. The columns (X-axis) show the prevalence of the classes in each dataset ("Up, "Down" and "Stable"). The columns also show the accuracy and leverage of respective model. The rows (Y-axis) depict the thresholds. Thresholds higher than 2,0% convey relatively poor results, why they are not depicted in the table. Complete results are enclosed in the appendices.

For example, the first row in Table 6 depicts information from the prediction that was performed on the Price Movement, 0.5% threshold dataset. The prevalence of the "Up", "Down" and "Stable" classes are 39,265%, 37,323% and 23,412% respectively. The 5-fold cross validation accuracy using default TF-IDF and Logistic Regression is 41,938%. The leverage, i.e. the difference between the accuracy and the most prevalent class (which in this case is the "Up" class), is 41,938% – 39,265% = 2,674 percentage points.

Table 6 - **Results from trying different thresholds, using Returns**

Threshold	Up	Down	Stable	Accuracy	Leverage
0,5%	39,265%	37,323%	23,412%	41,938%	2,673
1,0%	30,631%	28,656%	40,714%	45,250%	4,536
1,5%	23,794%	21,908%	54,299%	56,110%	1,812
2,0%	18,499%	16,994%	64,507%	65,141%	0,633
2,5%	14,652%	13,310%	72,038%	72,280%	0,242

Table 6 presents the distribution of the dataset between the classes 'Up', 'Down' and 'Stable' using Returns to calculate stock price movements and different thresholds for stock price movements. Additionally, the accuracy result and leverage from baseline (ZeroR) using Logistic Regression with standard parameters is presented.

Table 7 - **Results from trying different thresholds, using Market Adjusted Returns**

Threshold	Up	Down	Stable	Accuracy	Leverage
0,5%	37,807%	38,324%	23,868%	42,898%	4,573
1,0%	28,623%	28,721%	42,656%	47,029%	4,373
1,5%	21,819%	21,475%	56,706%	58,136%	1,430
2,0%	17,115%	16,733%	66,151%	66,603%	0,452
2,5%	13,758%	13,147%	73,095%	73,174%	0,079

Table 7 presents the distribution of the dataset between the classes 'Up', 'Down' and 'Stable' using Market Adjusted Returns to calculate stock price movements and different thresholds for stock price movements. Additionally, the accuracy result and leverage from baseline (ZeroR) using Logistic Regression with standard parameters is presented.

Table 8 - **Results from trying different thresholds, using Jensen's Alpha**

Threshold	Up	Down	Stable	Accuracy	Leverage
0,5%	37,337%	37,025%	25,638%	42,404%	5,067
1,0%	28,414%	27,082%	44,505%	49,073%	4,569
1,5%	21,447%	20,222%	58,332%	59,999%	1,667
2,0%	16,910%	15,620%	67,469%	68,023%	0,554
2,5%	13,543%	12,318%	74,138%	74,381%	0,242

Table 8 presents the distribution of the dataset between the classes 'Up', 'Down' and 'Stable' using Jensen's Alpha to calculate stock price movements and different thresholds for stock price movements. Additionally, the accuracy result and leverage from baseline (ZeroR) using Logistic Regression with standard parameters is presented.

4.3.4. Labeling analysis

Looking at the results, the first observation is that the three different price movement measurements convey different class prevalence. Differences are expected, since the increased risk adjustment by the financial models should result in smaller price movement measurements. According to financial theory, price movements calculated using risk-adjusted models should isolate the effect of the event more correctly.

In addition, the machine learning model seems to learn more from being supervised by the more complex price movement calculations. This is especially the case for the two lowest thresholds: Jensen’s Alpha convey the highest accuracy for both the 0,5% and the 1,0% thresholds. Using Returns at the 1,0% threshold does however conceive a higher accuracy than the Market Adjusted Returns. In addition, using Returns does in general generate better accuracy when the thresholds are larger.

The best thresholds to use seem to be the two lowest thresholds; after 1,0%, a relatively larger drop in accuracy occurs for all models. Nevertheless, the machine learning model performs the best when being supervised by the Jensen’s Alpha price movement calculation, using the 0,5% classification threshold. Consequently, this is the labeling procedure that we choose to proceed the study with, which creates the dataset to continue the process with.

Table 9 - **Change of labels between Price Movements Measurements, using the 0,5% classification cut-off threshold**

	R to MAR	**R to JA**	**MAR to JA**
Down to Down	8 093	8 518	9 072
Down to Stable	1 516	1 237	1 070
Down to Up	409	263	144
Stable to Down	1 783	1 152	714
Stable to Stable	3 137	4 249	4 948
Stable to Up	1 364	883	745
Up to Down	410	268	152
Up to Stable	1 754	1 395	863
Up to Up	8 375	8 876	9 133
Number of changes	*7 236*	*5 198*	*3 688*
Change in percent	*27 %*	*19 %*	*14 %*
Total	*26 841*	*26 841*	*26 841*

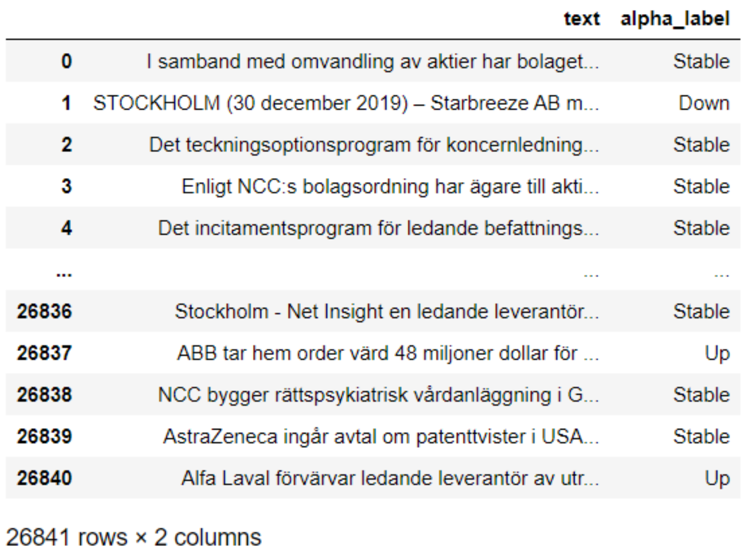
Table 9. Number of instances that change labels between different price movement measurements. E.g. Down to Stable show the number of instances classified as Down by one price movement measurement that changes to the label Stable using another price movement measurement. Furthermore, Down to Down is the number of instances that are labeled Down using both labeling techniques. R = Returns. MR = Market Adjusted Returns. JA = Jensen's Alpha.

Table 9 depicts the number of instances that switch labels when a different price movement measurement is used. A change of 14 to 27 percent is observed when using different labeling techniques. For example, when Jensen's Alpha is used, 8 518 of the Down instances are also labeled Down when Returns is used. However, 1 237 instances that were labeled Down when Returns was used, are instead labeled Stable when the Jensen's Alpha price movement calculation is used. Adhering to financial theory, labels based on Jensen's Alpha should be more correct than labels based on Returns.

4.3.5. Descriptive data

This sub-section includes some descriptive data of the chosen labeling procedure (Jensen's Alpha using 0,5% classification threshold) for the dataset. Figure 16 is a picture of the first five and last five rows in the dataset, depicting the text (ad-hoc disclosure/document) of each instance, and its corresponding label. The dataset consists of 26 841 instances of ad-hoc press releases, from 322 companies listed on OMX Stockholm per 2020-03-11.

Figure 16 - **Ten example instances of the dataset used for training and grid searching the Machine Learning model**



	text	alpha_label
0	I samband med omvandling av aktier har bolaget...	Stable
1	STOCKHOLM (30 december 2019) – Starbreeze AB m...	Down
2	Det teckningsoptionsprogram för koncernledning...	Stable
3	Enligt NCC:s bolagsordning har ägare till akti...	Stable
4	Det incitamentsprogram för ledande befattnings...	Stable
...	...	...
26836	Stockholm - Net Insight en ledande leverantör...	Stable
26837	ABB tar hem order värd 48 miljoner dollar för ...	Up
26838	NCC bygger rättspsykiatrisk vårdanläggning i G...	Stable
26839	AstraZeneca ingår avtal om patenttvister i USA...	Stable
26840	Alfa Laval förvärvar ledande leverantör av utr...	Up

26841 rows × 2 columns

Figure 16 depicts 10 example instances from the dataset that is used when gridsearching the best parameters for the Machine Learning model. The label is assigned by using Jensen's Alpha in the context of the Capital Asset Pricing Model (CAPM).

4.3.5.1. Document text statistics

Figure 17 depicts two boxplots, where the Y-axis reflects the document lengths (number of words in each document). Documents outside the 10th percentile (i.e. the shortest 5 percent and the longest 5 percent) are considered outliers, meaning that the whiskers reach to the lower 5 percent and the higher 95 percent of the number of words in a document. Both visualizations are of the same boxplot; the difference is that the visualization to the left includes outliers, whereas outliers are removed from the right-side visualization. The box represents documents with the number of words amounting to the second and third quartiles (25 - 75 percent of all documents). The orange line reflects the median number of words.

Figure 17 - **Boxplots depicting the number of words in each document**

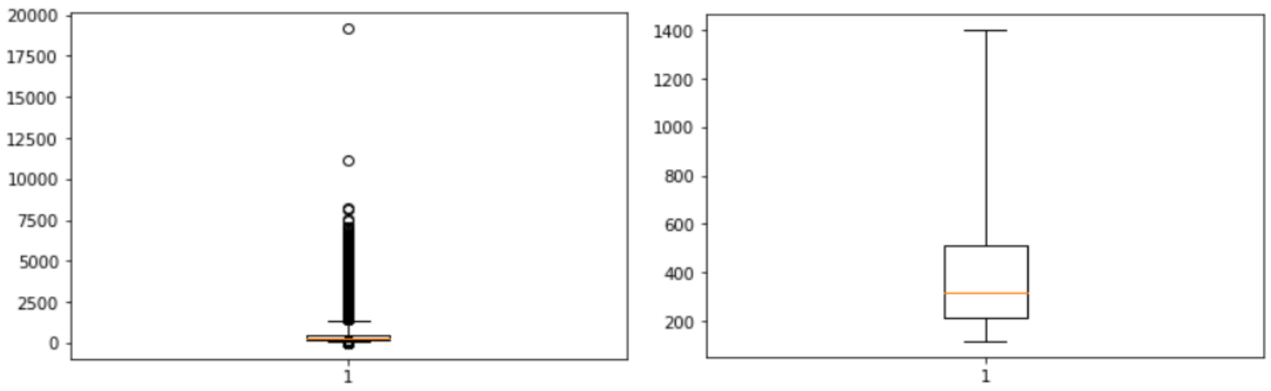


Figure 17 show two boxplots of the number of words in the documents of the dataset used to train and evaluate the Machine Learning models. The left boxplot shows the entire dataset including outliers whereas the boxplot to the right removes the outliers (i.e. data above 95 percent and below 95 percent).

Table 10 - **Descriptive statistics of the text data in the corpus**

5%	25% (1Q)	Median (2Q)	75% (3Q)	95%	Min	Max	Mean
117	215	316	511	1 398	1	19 165	486

Table 10 depicts the descriptive statistics of the boxplots (Figure 17) about the structure of the distribution of the data set.

Table 10 depicts some descriptive statistics of the boxplots. All numbers reflect the number of words in a document. As observed, the median number of words in each document are 316 (mean 486), where 90 % of all documents contain between 117 and 1398 words. The largest document is a document containing 19 165 words. The shortest documents are five documents containing a single word/token. The total number of words (tokens split on space) over the whole corpus are 13 043 595, of which 329 914 are unique. Approximately 244 000 out of these 329 914 are "real" words; the rest are other types of tokens, such as names of persons, URL's, numbers, or equivalent.

Figure 18 - **Visualization of the publication distribution, no. of documents per year**

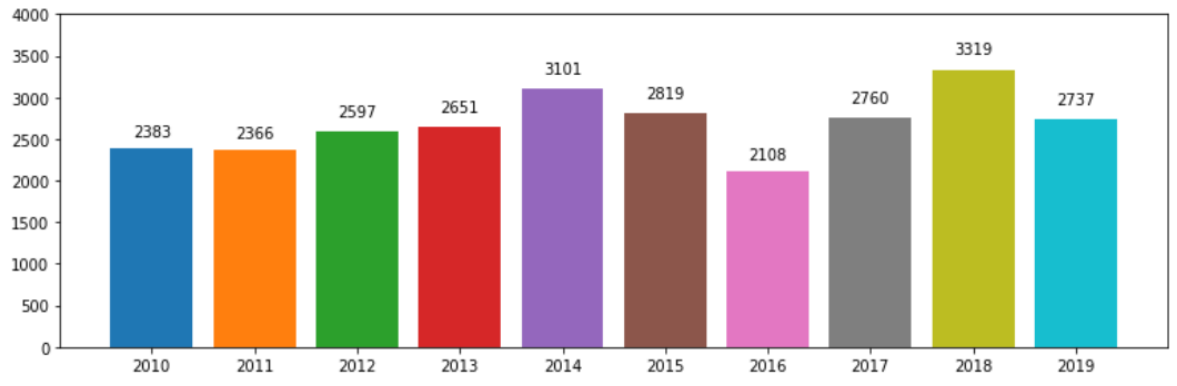


Figure 18 depicts the publication distribution over the ten years included in the study. The X-axis represents the year, the Y-axis represents the number of published disclosures. It seems to be a relatively even distribution between the years.

Figure 19 - **Visualization of the publication distribution, no. of documents per month**

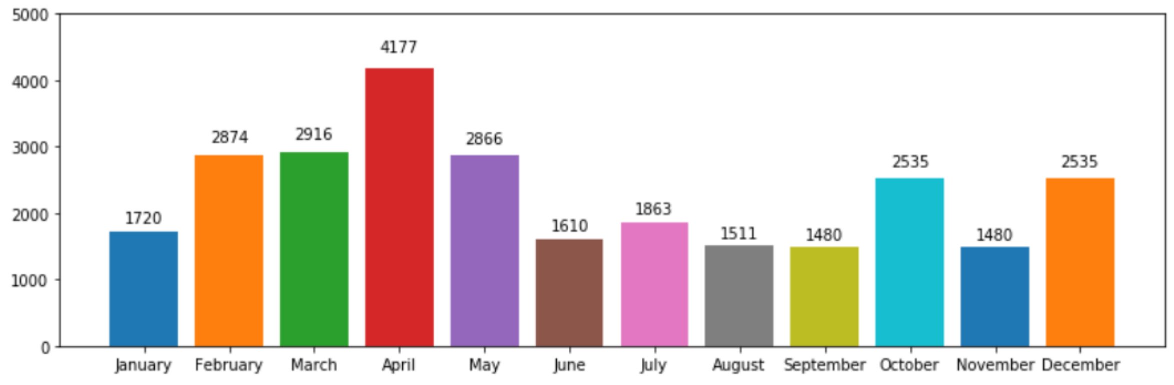


Figure 19 depicts the publication distribution on a monthly basis. All numbers are the total number of disclosures for the month over the nine years included in this dataset. As visualized, April has the highest share of publications.

4.4. Comparing classifiers

Now that a labeling procedure is decided on, we proceed with comparing and choosing a classifier. The types compared are (1) Logistic Regression, (2) Naïve Bayes, and (3) Support Vector Machines. Neural Networks is not included since it is used by BERT, which is applied onto the dataset in parallel to one of these models. The experiment for BERT is presented in another section.

The choosing is conducted by applying each of the models onto the dataset and evaluating the Leverage from the cross-validation accuracy against the baseline. The model with the highest Leverage is deemed the best, and thus the one that we continue with. The cross-validation is performed using the same dataset division as for the labeling approach selection, i.e. on the Jensen's Alpha 0,5 percent threshold labeling procedure, performing an identical 80/20 split on the 26 841 rows from 2010-01-01 to 2019-12-31. The 80 percent training data is cross validated over in this step. The cross-validation accuracy is the reported accuracy; the 20 percent hold-out data is saved for evaluating the final model.

The classification approach is kept simple to avoid time consuming calculations, thereby leaving more time for focusing on pre-processing procedures such as TF-IDF and embeddings. Hence, the settings used for each of the models are the default per Sci-Kit Learn version 0.21.2.

4.4.1. Results Logistic Regression

The settings used for the Logistic Regression are the same as presented when choosing the labeling procedure. The used (default) settings are, among others, C = 1, solver = Liblinear, multi_class = ovr (one-vs-rest classification scheme). The results are presented in Table 11.

Table 11 - **Results from the Logistic Regression**

Threshold	Labeling	Up	Down	Stable	Accuracy	Leverage	Time
0,5%	Jensen's Alpha	37,337%	37,025%	25,638%	42,404%	5,067	2m 33s

Table 11 presents the test results from the Logistic Regression model applied on the dataset created using Jensen's Alpha for labeling and a threshold of stock price movement of 0,5 percent.

4.4.2. Results Naïve Bayes

The used Naïve Bayes algorithm is Multinomial. The used (default) settings are, among others, the smoothing parameter Alpha = 1. The results are presented below in Table 12.

Table 12 - **Results from the multinomial Naïve Bayes classifier**

Threshold	Labeling	Up	Down	Stable	Accuracy	Leverage	Time
0,5%	Jensen's Alpha	37,337%	37,025%	25,638%	40,425%	3,088	1m 29s

Table 12 presents the test results from the Naïve Bayes model applied on the dataset created using Jensen's Alpha for labeling and a threshold of stock price movement of 0,5 percent.

4.4.3. Results SVM (SGD) Classifier

The Stochastic Gradient Descent (SGD) Classifier has default loss function = hinge, thus making it into a Linear Support Vector Machines, although with a Stochastic Gradient Descent algorithm (instead of Liblinear for the "Linear SVM") for solving the loss function. The penalty (regularization) is L2. The constant multiplied with the regularization term, which is also used for computing the learning rate, is alpha. Default alpha = 0,0001. (Sci-Kit, 2019c). The results are presented below in Table 13.

Table 13 - **Results from SVM (SGD)**

Threshold	Labeling	Up	Down	Stable	Accuracy	Leverage	Time
0,5%	Jensen's Alpha	37,337%	37,025%	25,638%	41,733%	4,396	1m 34s

Table 13 presents the test results from the SVM(SGD) model applied on the dataset created using Jensen's Alpha for labeling and a threshold of stock price movement of 0,5 percent.

4.4.4. Results Linear SVC

The Linear Support Vector Classifier is a Support Vector Machines using the Liblinear algorithm for solving the loss function, similar to default Logistic Regression as per Sci-Kit learn 0.21.2. The regularization parameter C defaults to 1. The (default) loss function is squared hinge. The regularization penalty is L2. (Sci-Kit, 2019d). The results are presented in Table 14.

Table 14 - **Results from Linear SVC**

Threshold	**Labeling**	**Up**	**Down**	**Stable**	**Accuracy**	**Leverage**	**Time**
0,5%	Jensen’s Alpha	37,337%	37,025%	25,638%	41,161%	3,824	1m 52s

Table 14 presents the test results from the Linear SVC model applied on the dataset created using Jensen’s Alpha for labeling and a threshold of stock price movement of 0,5 percent.

4.4.5. Results Kernelized SVC

The Support Vector Classifier (SVC) is similar to the “Linear SVC”, but is built on LIBSVM instead of Liblinear. The default kernel is the Radial Basis Functon (Gaussian), default regularization parameter C = 1, and default kernel coefficient “gamma” = 1/n, where n is the number of features (Sci-Kit, 2019e). The results are presented in Table 15.

Table 15 - **Results from Non-Linear SVC**

Threshold	**Labeling**	**Up**	**Down**	**Stable**	**Accuracy**	**Leverage**	**Time**
0,5%	Jensen’s Alpha	37,337%	37,025%	25,638%	37,337%	0,000	1h 52m 44s

Table 15 presents the test results from the Non-Linear SVC model applied on the dataset created using Jensen’s Alpha for labeling and a threshold of stock price movement of 0,5 percent.

4.4.6. Classifier selection analysis

After testing a variety of classifiers which utilize different loss functions and algorithms for trying to predict price movements occurring from ad-hoc disclosures, none of the alternatives beats Logistic Regression using default settings. The kernelized Support Vector Machines scores the lowest and sticks out regarding computation time compared to the other classifiers; all classifiers take less than three minutes using default parameters to perform a 5-fold cross-validation (1 core), whereas the kernelized Support Vector Machines takes almost two hours.

Logistic Regression might have been boosted from being the model used when choosing the labeling procedure. For example, it is possible that the second-best model, SGD Support Vector Machines, would perform better than Logistic Regression when another labeling procedure is used, such as using the 1,0% threshold or another price measurement type. Nevertheless, Logistic Regression performed the best on this simple test and is therefore the model that we proceed with, instead of comparing the models after grid searching the hyperparameters. This allows us to put more emphasis on comparing different pre-processing techniques.

4.5. Comparing pre-processing techniques

Next, the Jensen's Alpha label 0,5% threshold dataset, together with Logistic Regression, is used for testing different kinds of pre-processing techniques. The compared pre-processing techniques are TF-IDF, Word Embeddings and Document Embeddings.

4.5.1. TF-IDF

First, we delve deeper into TF-IDF by grid searching over different hyperparameters. The same train_test_split is performed on the dataset for this step, meaning that the same 80% of the 26 841 row dataset is used to grid-search over, using 5-fold cross-validation. The reported accuracy is the cross-validation accuracy; the hold-out data is saved for evaluating the final model, and is therefore not used in this step.

We separate the grid search into two batches (scripts) based on the TF-IDF analyzer parameter (word-/character gram), since it is not necessary for these two to search over the same n_gram space. We also try different restrictions on document frequency. In addition, several parameters for the Logistic Regression are tested. We choose to test the Liblinear and Saga solvers. The Liblinear solver can only handle the one-vs-rest scheme (OVR), whereas the Saga (Stochastic Average Gradient) solver can handle both OVR and multinomial loss (Sci-Kit, 2019b). We deem it too time consuming to test both solvers on OVR. Hence, Liblinear is tested with OVR, and Saga with multinomial. For computation speed purposes, we divide the grid search into two additional batches to enable grid searching different parameter settings simultaneously. We also run the scripts using multiple cores. The scripts are run on Google Cloud servers (16 cores, 104GB RAM).

Table 16 depicts the model parameters and the best results from respective script. When multiple settings are tried, those producing the best results are underlined. The prefix "TFIDF" adheres to parameters relating to the TF-IDF pre-processing, and the "CLF" adheres to parameters relating to Logistic Regression.

Table 16 - **TF-IDF grid search**

	Batch 1	**Batch 2**	**Batch 3**	**Batch 4**
TFIDF__analyzer	Word	Word	Character	Character
TFIDF__ngram_range	(1,1), (1,2), **(1,3)**	(1,1), (1,2), **(1,3)**	(2,7), (3,8), **(4,9)**	(2,7), (3,8), **(4,9)**
TFIDF__max_df	**1.0**, 0.8	1.0, **0.8**	**1.0**, 0.8	1.0, **0.8**
TFIDF__min_df	**1**, 50	**1**, 50	**1**, 50	**1,** 50
CLF__C	0.001, **1**, 4	0.001, **1**, 4	0.001, 1, **4**	0.001, 1, **4**
CLF__solver	Liblinear	Saga	Liblinear	Saga
CLF__multi_class	OVR	Multinomial	OVR	Multinomial
Time	35m (8/8 cores)	58m (4/4 cores)	12h 15m (8/16 cores)	10h 27m (12/16 cores)
Baseline	37,337%	37,337%	37,337%	37,337%
Accuracy	42,665%	42,663%	42,777%	42,870%
Leverage	**5,328**	**5,326**	**5,440**	**5,533**

Table 16 depicts all data set batches and the different pre-processing parameters they have been grid searched on. The best parameter combination per batch is showed per parameter in bold and the Leverage score is what is compared.

The character grams perform better than word grams, where the best performing model (Batch 4) uses 4 to 9 characters in its vocabulary, where max document frequency is set to 80% of all documents, and minimum document frequency is set to 1 document. The best settings for the Logistic Regression are the Saga solver with multinomial loss. Batch 1 and Batch 2 were run on two different CPU's, which constitutes understandable differences in computing time. Batch 3 and Batch 4 were however run simultaneously on two identical machines, but Batch 3 (using the Liblinear solver) needed to be reduced to 8 cores (out of maximum 16) to avoid memory error on the 104GB memory machine. In contrast, Batch 4 (using the Saga solver) managed to stay below maximum memory using 12 cores.

4.5.2. Embeddings

The next pre-processing approach is Embeddings. To implement Doc2Vec, Gensim is utilized. Gensim ("Generate Similar") is a Python machine learning library with various tools for realizing unsupervised semantic modeling from plain text (Řehůřek, 2019a). When constructing embeddings, it uses the Word2Vec as presented by Mikolov et al. (2013) and Doc2Vec as presented by Le & Mikolov (2014). Gensim enables compatibility with Sci-Kit Learn through the Doc2Vec and Word2Vec Sci-Kit Learn wrappers. When implementing the Word2Vec, it is however necessary that all words (tokens) in a predicted document exists in the vocabulary. When using data that was not used for training the model, previously unseen tokens results in that the script crashes. To solve this problem, we write code manually by adhering to the Word2Vec with Logistic Regression guide constructed by Susan Li (S. Li, 2018b). The approach used by S. Li (2018b) to create document vectors from word embeddings is to average each dimension of the word vectors in each document (also mentioned as a feasible approach by Řehůřek (2019)) which is therefore also the approach used in this study.

Due to limited time, we choose to test the Word2Vec approach using a pre-trained Word2Vec model only. In contrast, the Doc2Vec model is trained on the in-house dataset only. The pre-trained model is downloaded from the Nordic Language Processing Laboratory (NLPL), which is a shared repository of large-text resources, an initiative by the University of Oslo to enable fast experimentation and replication of Natural Language Processing tasks (Fares et al., 2017). The model was used in the 2017 CoNLL (conference on Computational Natural Language Learning) shared task. It was trained using the UDPipe (Straka & Straková, 2016) on Swedish Wikipedia and "common crawl" material, i.e. data scraped from various pages all over the web. The model is trained using the Skip-Gram architecture and negative sampling algorithm (as opposed to hierarchical softmax), with a context window of 10 and minimum word frequency of 10. Each word is embedded on a 100-dimensional space. It has been trained with 2 iterations (epochs).

Since there are 5 cross-validation folds, the vocabulary of the Doc2Vec is at any point in time built on 4 folds of the 80% training data, i.e. 0,8*0,8 = 0,64% of the 26 841 rows. Since the folds used for training are changing, the vocabulary size is also changing. As a point of reference, a random cross-validation model trained on 4 folds of the training data consists of 69 720 words. In contrast, the vocabulary of the imported Word2Vec model consists of 3 010 472 words.

To evaluate the embeddings techniques on the task at hand, the same 80/20 train_test_split that was performed for previous grid searches is conducted on the 26 841 rows dataset, where the 80% share is grid-searched over using 5-fold cross-validation. The cross-validation is the reported accuracy; the 20% hold-out data is not used. Since the Word2Vec model is imported, the grid search for this machine learning model only regards to the parameters pertaining Logistic Regression. For Doc2Vec, the grid search also pertains the embeddings model.

The parameter settings that are grid searched over are visualized in Table 17, where the best settings are in bold when multiple settings are tried. For computation time purposes, the Doc2Vec model is split into four batches to be grid-searched simultaneously. One uses the Distributed Memory model of Paragraph Vectors (PV-DM) architecture, and the other the Distributed Bag of Words model of Paragraph Vectors (PV-DBOW) architecture. The latter is similar to the Word2Vec Skip-Gram architecture (Le & Mikolov, 2014), used by the pre-trained model. Also, for time purposes, only the default Negative sampling algorithm is tested, using the default setting 5 for the number of drawn noise words (Gensim, 2019). The Doc2Vec scripts are run on Google Cloud servers (16 cores, 104GB memory). Multiple cores are used both when building the Doc2Vec model, and for cross-validating. The Word2Vec is run on a 4-core PC, with 8GB memory.

Table 17 - **Embeddings parameter grid search**

	Word2Vec	**Doc2Vec 1**	**Doc2Vec 2**	**Doc2Vec 3**	**Doc2Vec 4**
Architecture	Skip-Gram	PV-DBOW	PV-DBOW	PV-DM	PV-DM
Algorithm	Negative sampling	Negative sampling	Negative sampling	Negative sampling	Negative sampling
Size	100	**50**, 100, 300	**50**, 100, 300	50, 100, **300**	50, 100, **300**
window	10	2, 5, **8**	2, 5, **8**	**2**, 5, 8	**2**, 5, 8
min_count	10	5	5	5	5
epochs	2	1, 5, 15, **30**	1, 5, 15, **30**	1, 5, 15, **30**	1, 5, 15, **30**
Solver	Liblinear, Saga	Saga	Liblinear	Saga	Liblinear
Multi_class	Auto[6]	Multinomial	OVR	Multinomial	OVR
C	0.001, 1, **4**	0.001, 1, **4**	0.001, 1, **4**	**0.001**, 1, 4	**0.001**, 1, 4
Baseline	37,337%	37,337%	37,337%	37,337%	37,337%
Accuracy	40,448%	41,938%	41,519%	40,406%	39,843%
Leverage	3,111	4,601	4,182	3,069	2,501

Table 17 depicts the grid searched parameters for the different embedding methods. The best parameters for each data batch are in bold.

The best leverage is received by the Doc2Vec 2 model that uses the PV-DBOW setting and the Saga Logistic Regression solver. For the four Doc2Vec grid searches, the choice of architecture seems to make the largest difference to the scores. This is further motivated from the higher performance by the Word2Vec model, which is trained on the Skip-Gram architecture, like PV-DBOW; despite

[6] Setting ”Auto” means that one-vs-rest (OVR) is chosen when solver is Liblinear, and Multinomial is chosen when solver is Saga.

lacking a paragraph vector, it still performs better than Doc2Vec PV-DM. In addition, the Saga solver performs better for Doc2Vec, but worse for Word2Vec. Naturally however, the Word2Vec model also differs from the Doc2Vec because it is pre-trained. The pre-trained model's vocabulary is much larger, but perhaps lacks some specialized words that occur in the in-house dataset. Comparing these state-of-the-art embeddings models to the more basic TF-IDF bag of words-approach, none of the embeddings models perform better than TF-IDF. However, the fact that TF-IDF was used for choosing the labeling procedure (0,5% Jensen's Alpha dataset) might affect the results slightly in favor of TF-IDF.

4.5.3. BERT

We use the original BERT code published by Google to fine-tune the model based on two pre-trained models (Devlin et al., 2018/2020). In order to allow multi-classification, the code in "run_classifier.py" under "get_label" for the relevant processor used is slightly altered to allow classification based on our three labels "Up", "Down" and "Stable" instead of the original labels "1" and "0". The dataset is split and transformed to fit the BERT model. For the training set 80% of the dataset is used and the remaining 20% is split equally between the dev set and test set.

We first try the multilingual pre-trained model provided by Google on our dataset and achieve a test score only slightly above the baseline. As the multilingual model is pre-trained on 100 different languages and thus only have a limited vocabulary for each language, we subsequently try the pre-trained Swedish model from The National Library of Sweden with text data from several sources as our primary pre-trained model.

Fine tuning the BERT model with a pre-trained model require a lot of computational power why we use Google Cloud Platform to create a Linux Virtual Machine (n1-standard-8 with 8 vCPUs and 30 GB memory) to run the code on. Furthermore, we use Google Cloud Platform Storage to store the data and output from the model. This significantly increase our ability to run different BERT models. To set up the virtual environment and run the code, we use a guide published by Google on how to fine tune BERT with Google Cloud Platform (Google, 2020). The BERT code published by Google[7] is uploaded to the Virtual Machine together with the respective pre-trained model. The command in Appendices, which is slightly altered for each pre-trained model, is used to initiate and run BERT through the Virtual Machine. Based on the distribution of the dataset presented above, we use the the third quartile (500) of words for the parameter max_seq_length.

Table 18 - **Results from BERT**

Model	Dataset	Max_Seq	Baseline	Eval Accuracy	Leverage	Eval Loss
KB	Jensen's Alpha 0,5%	500	37,337%	41,628%	4,291	1,059

Table 18 shows the result from applying the pre-trained Swedish model from The Swedish Royal Library (KB) on the dataset labeled using Jensen's Alpha and a threshold of 0,5 percent.

BERT with a pre-trained Swedish model produces an accuracy of 41,6 percent with a leverage of 4,3 above the baseline using the same dataset as Logistic Regression. Thus, the BERT model does not perform better than the best Logistic Regression model.

4.6. Summary of results of constructing the machine learning modeling – H1

Out of the 27 labeling procedures (i.e. datasets) compared, the best labeling approach is concluded to be the 0,5% cut-off threshold, using the Jensen's Alpha measurement. The number of instances in this dataset is 26 841, since the instances from 2020-01-01 onward are removed, to be used for simulating the use of the best algorithm onto the stock market.

[7] https://github.com/google-research/bert#pre-trained-models

The best model when comparing Logistic Regression, Multinomial Naïve Bayes, SGD Support Vector Machines, Linear Support Vector Machines and Kernelized Support Vector Machines (all using default settings) with TF-IDF pre-processing (also default settings) and the 0,5% threshold Jensen's Alpha dataset, is Logistic Regression.

When lastly comparing TF-IDF, Word2Vec and Doc2Vec with Logistic Regression as classifier, and BERT which uses a Neural Network, the best-performing pre-processing technique is TF-IDF.

4.6.1. Answering H1

H1: *A machine learning model built on ad-hoc disclosures from the Swedish market can learn from the data and classify unseen disclosures better than a relevant baseline.*

To answer the first hypothesis of this thesis, we choose to evaluate the best model on the hold-out dataset that has not yet been used. Hence, the model is trained on the same 80% dataset that has been used for grid searching and cross-validating. The used TF-IDF settings are the best-performing settings, i.e. character grams (4,9), max_df = 0,8 (80%) and min_df = 1. The used Logistic Regression settings are the saga solver with multinomial loss, with the regularization parameter C = 1. The resulting baseline (the most prevalent class), accuracy and leverage are presented in Table 19. The accuracy is the test accuracy, contrasting the cross-validation accuracies presented in previous sections.

Table 19 - **Final Evaluation Results**

Threshold	**Labeling**	**Up**	**Down**	**Stable**	**Test Accuracy**	**Leverage**
0,5%	Jensen's Alpha	37,337%	37,025%	25,638%	43,658%	6,321

Table 19 depicts the results for the best machine learning model Logistic Regression with parameters: character grams (4,9), max_df = 0,8 (80%) and min_df = 1.

The leverage toward the baseline is 6,321 (43,658% accuracy), as compared to the cross-validation leverage of 5,553 (42,870% accuracy) that was achieved when grid-searching. This higher accuracy is somewhat surprising, since we assume that the best settings, if anything, could have included a marginal overfitting on the cross-validation accuracy and thereby resulted in a slightly lower accuracy on the hold-out data. Instead, the accuracy improves by 0,8 percentage points on the hold-out data. One possible explanation for this higher accuracy is that the model producing the final results is trained on 80% of the dataset, whereas the model cross-validated over at any point in time only utilized 4/5 of the 80% for training, i.e. 0,8*0,8 = 0,67% of the dataset. The extra data might improve the quality of the model and thereby its predictive ability. Another possible explanation is that these hold-out results score better from pure chance. The Confusion Matrix of this final evaluation is depicted in Table 21. The Classification Report, including Precision, Recall, F1-score and Support of each class, is included in the appendices.

Nevertheless, the accuracy achieved on the hold-out (test) data does in fact score 6,321 percentage points above the ZeroR baseline, meaning that the machine learning model built on ad-hoc disclosures from the Swedish market has learned from the data and could classify unseen disclosures better than a relevant baseline. Consequently, **Hypothesis 1 is confirmed.**

5. Experiment H2 – Simulating an algorithmic Trading Strategy

This section is dedicated to the practical steps taken for answering Hypothesis 2.

H2: *Since the stock market reflects new ad-hoc information in the stock price with a time lag, the model can be used as a part of a trading strategy on the Swedish stock market for generating abnormal returns.*

5.1.1. Introduction

The steps to test H2 include:

1. **Constructing a data set to simulate the trading algorithm** – Documents and metadata is collected using the same approach as in chapter four in order to simulate a trading strategy.
2. **Setting the parameter for the trading algorithm –** Using the previously trained model, a probability threshold is set with the goal to increase precision in the classes 'Up' or 'Down'.
3. **Applying the algorithm on the stock market -** Finally, the simulated algorithm is applied on the trading dataset and using different simulations in delay between publication and initiation of trade as well as different probability thresholds.

As previously presented, the best model is Logistic Regression with character grams (4,9) with an accuracy of 43,658 percent against the baseline of 37,337 percent, i.e. a leverage of 6,321. The trading strategy is firstly evaluated on tick data. However, as tick stock data for 40 percent of the trading dataset is not available, hourly stock data is also used separately.

Throughout the experiment, Python is used to gather and pre-process text data as well as to apply the Machine Learning model to predict stock price movements on previously unseen data. Finally, Python is also used to simulate the trading strategy and evaluate it. The code used in the following section is available on GitHub.[8]

5.2. Constructing the trading dataset

Text data and metadata for corporate disclosures released on Nasdaq OMX website between 2020-01-01 and 2020-03-11 is used to simulate the trading strategy. These are separated from the full dataset collected in chapter four before training the model, and instead used to simulate the trading strategy. A total of 461 documents are collected during the time frame. To calculate the price movements, we use tick data downloaded from Swedish House of Finance which is a part of Stockholm School of Economics. However, due to some missing data on their servers, we are only able to extract tick data for roughly 60 percent of the trading dataset. Thus, we also collect hourly data from Refinitiv Eikon to test the full dataset as well.

5.2.1. Descriptive data of the dataset

The trading dataset contains 461 instances from 233 companies listed on Nasdaq OMX 2020-03-11. Figure 20 depicts two boxplots of the number of words per instance. The left visualization includes outliers. The right-hand side visualization is the same boxplot, but where outliers have been excluded. The median as well as the mean value is higher than for the entire dataset indicating that on average the length of the documents are longer in the trading dataset compared to the training dataset. Table 20 contains some descriptive data used to visualize the boxplots. Each number is the number of words (split on space) per document. Figures 21 and 22 depict the distribution of disclosure publications

[8] https://github.com/AlgoTrading2020/Thesis2020

based on time of day and day of week. As depicted, most documents in the trading dataset are published before the stock market opens which is of interest when evaluating the trading strategy.

Figure 20 - **Boxplots of text data**

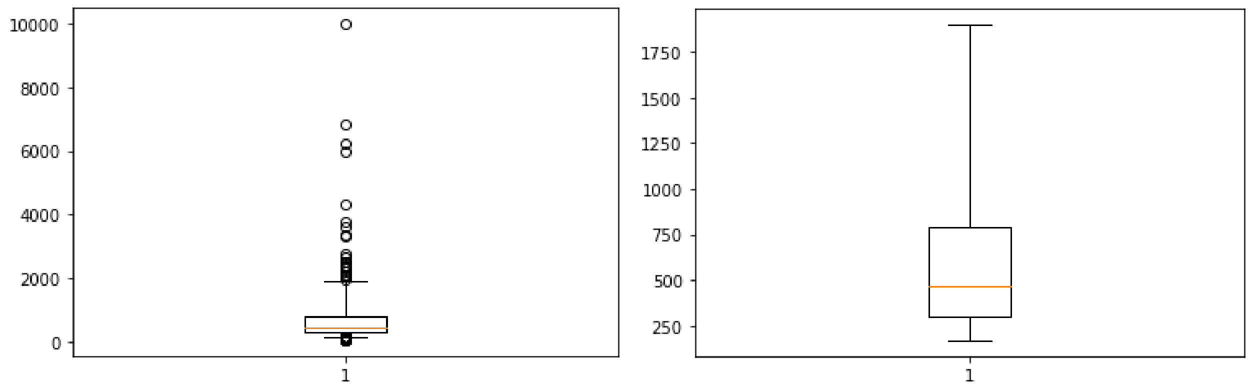


Figure 20 show two boxplots of the number of words in the documents of the trading data set. The left boxplot shows the entire dataset including outliers whereas the boxplot to the right removes the outliers (I.e. data above 95 percent and below 95 percent).

Table 20 - **Descriptive data of the documents in the trading data set**

5%	25% (1Q)	Median (2Q)	75% (3Q)	95%	Min	Max	Mean
169	298	463	794	1 899	49	9986	694

Table 20 presents some descriptive data of number of words in the documents from the trading dataset.

Figure 21 - **Visualization of the publication distribution, time of day**

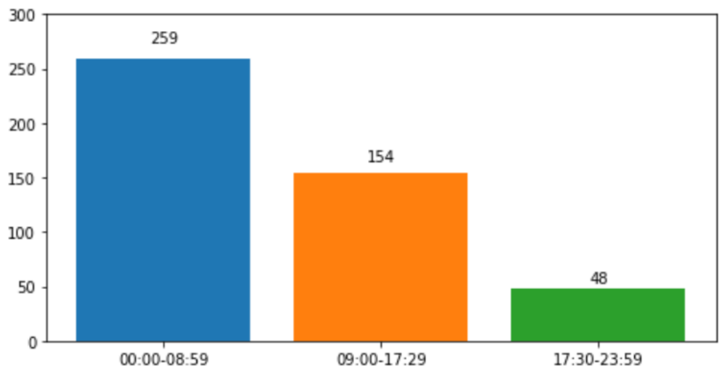


Figure 21 shows the distribution publication time of the documents in the trading dataset. As shown, 66 percent of the disclosures are published when the markets are closed either before or after.

Figure 22 - **Visualization of the publication distribution, day of week**

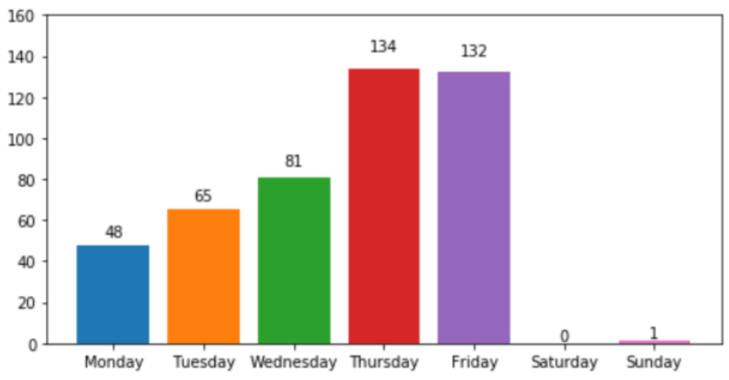


Figure 22 depicts the distribution of publication of ad-hoc disclosures in the trading dataset across day of week.

5.3. Logistic Regression probability threshold

From a business perspective we are primarily interested in classifying ‘Up’ and ‘Down’ with a high precision since these classes result in an action to buy or short a stock. Hence, the cost of incorrectly classifying an instance as Stable is zero, whereas incorrectly classifying an instance as ‘Up’ or ‘Down’ has a cost in trading fees as well as a potential negative return. Since the final model in H1 is evaluated by predicting the classes of the test dataset, we can produce a Confusion Matrix from that step, as depicted in Table 21. For the model with the highest accuracy, the mean precision of ‘Up’ is about 43%, and the mean precisions of ‘Down’ and ‘Stable” are 44 percent with the default settings.

Table 21 - **Confusion Matrix from predicting unseen data**

	Down (Predicted)	*Stable (Predicted)*	*Up (Predicted)*	**Total**
Down (Actual)	927 (44 %)	242	819	1988
Stable (Actual)	474	418 (44 %)	484	1376
Up (Actual)	715	291	999 (43 %)	2005
Total	2116	951	2302	5 369

The confusion matrix of the test set used for evaluating the machine learning model in Experiment H1. This is thus not the same data set as the trading strategy data set. It works as an approximation to see which errors that the best machine learning model makes between the classes. The parentheses depict the Precision of each predicted class (Down, Stable, Up).

Since Logistic Regression can provide the probability that an instance belongs to each class, we can set a higher probability threshold for the classes ‘Up’ and ‘Down’ to increase the precision of these classes. The change in precision between different thresholds on this test dataset can guide the choice of threshold to use when predicting instances in the trading dataset. However, increasing the probability threshold also decreases the number of actions that the trading algorithm would perform. Figure 24 depicts how the mean precision of ‘Up’ and ‘Down’ changes with a higher threshold, on the test dataset from H1. Furthermore, Figure 24 also depicts how the number of actions that the trading algorithm would take changes with a higher threshold on the same dataset. The large decrease in precision when reaching a high threshold is due to the model classifying very few instances as ‘Up’ or ‘Down’ and thus each classification has a major effect on the mean precision.

From a business perspective it is reasonable to assume that a higher precision on the actions should have a higher expected return per action. However, the highest precision might not be where the algorithm makes the highest monetary value in total, since the number of actions rapidly decrease with a higher threshold. Since our trading dataset only spans over 2,5 months and contains 461 instances due to limitations in data retrieval, we must choose a threshold that generates a large enough number of actions to test the trading algorithm on. We choose to test the threshold that results in the trading algorithm acting (Buying or Shorting) on approximately 50 percent of the ad-hoc disclosures in the test dataset, which is at the 44 percent classification threshold. Hence, we expect the trading algorithm to take about 230 actions on our trading dataset which should be enough to generalize the results. Additionally, we also run the trading strategy with default probability threshold (33%) for comparison to see if the raised threshold results in any improvement or not.

Figure 23 - **Mean Precision versus number of actions**

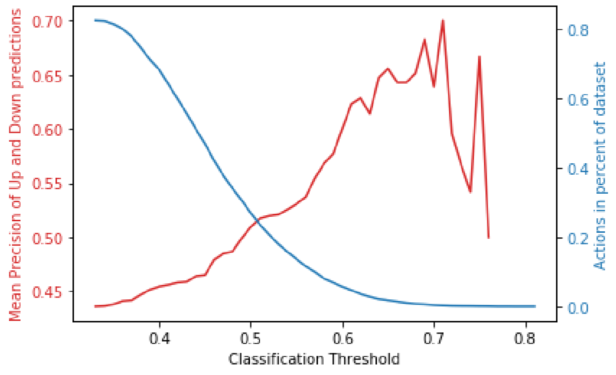


Figure 23 depicts how the mean precision of classes 'Up' and 'Down' i.e. the actions change when the classification threshold for the same classes are increased. Furthermore, the decrease in number instances classified as 'Up' or 'Down' in relation to the dataset as the threshold is increased is also depicted. This illustrates the trade-off between a high precision and the amount of actions taken. The graph is calculated on the test dataset from Experiment H1.

After first having established the probability thresholds based on the test dataset, the model is subsequently applied to the trading dataset to classify the instances. Using the probability threshold of 44 percent for the classes 'Up' and 'Down', the model classifies, out of the 461 instances, 164 instances as 'Up', 131 instances as 'Down' and 166 instances as 'Stable' For comparison, data for both default (33%) and 44 percent threshold as well as data for the total dataset and the dataset using minute data (instances with missing data removed) is presented below in Table 22. The predictions yield a higher action rate for the 44 percent probability threshold than expected, 64 percent instead of 50 percent.

Table 22 - **Classification results using different thresholds and data**

Probability threshold	Data	Up	Down	Stable	Instances	Action rate
Default	Total	215	193	53	461	88,5 %
44 %	Total	164	131	166	461	64 %
Default	Tick	125	123	27	275	90 %
44 %	Tick	94	82	99	275	64 %

Table 22 presents the results from applying the Machine Learning model on our trading dataset to predict the instances using the full dataset as well as the dataset where tick data is available. Furthermore, data using the default probability threshold and the choosen threshold of 44 percent is presented for comparison.

5.4. Simulating the trading algorithm

5.4.1. Locating stock prices

Using minute data for stock prices, we calculate the returns generated by the simulated trading strategy using Returns, Market Adjusted Returns and Jensen’s alpha. I.e. the same methods used to label the instances.

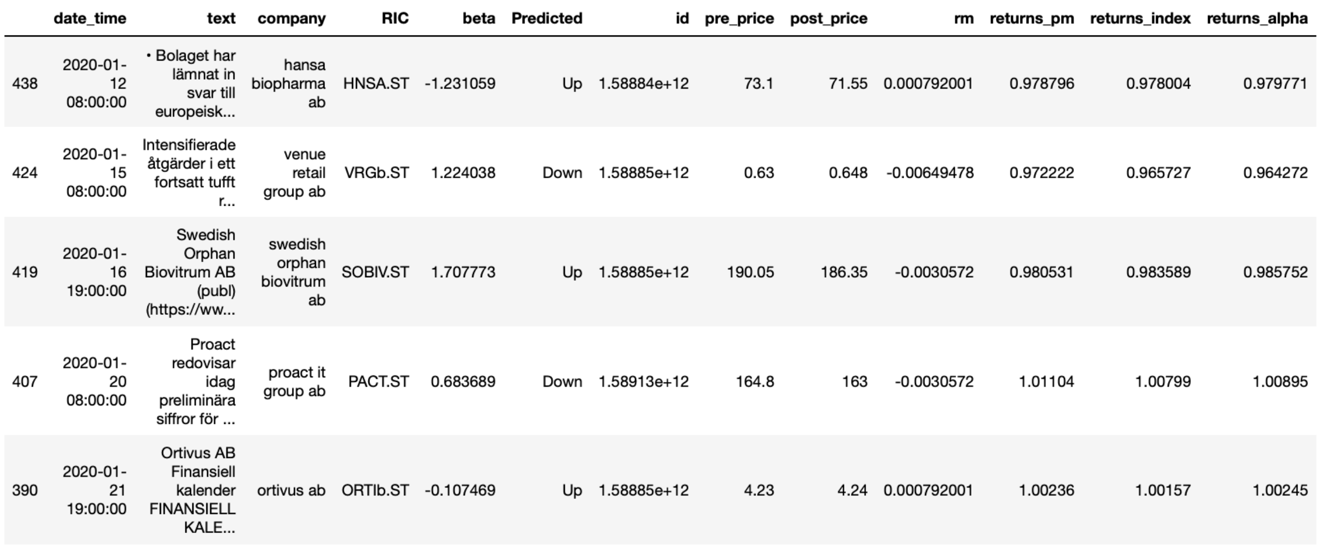
For each instance predicted as ‘Up’ or ‘Down’, i.e. the actions of the simulated trading strategy, the timestamp of the ad-hoc disclosure is extracted. The timestamp is subsequently used to calculate the trading time, i.e. when the trade is initiated. For corporate disclosures released when the market is closed (between 17:30 and 9:00, weekends, etc.), the first price available after the release is used, i.e. in most cases the opening price. For ad-hoc disclosures released during the day using tick data, we use as previously presented a simulated delay and the first price available after is used. For hourly data we use the first available stock price after the corporate disclosure is published, thus the time lag varies between 0 and 60 minutes. From the trading time, one hour is added to calculate when the trade should be terminated.

To locate stock prices, each instance is assigned a unique ID to match with tick stock data from the Swedish House of Finance. To match with hourly stock data, the RIC code is used. Finally, market returns for the corresponding time frame are matched with each trade. As the holding period is only one hour, and the fact that the risk-free rate is close to zero, we use zero as a proxy for the risk-free rate when evaluating the trading strategy.

5.4.2. Calculating returns

Using the same technique to create the labels, returns are calculated for simple Returns, Market Adjusted Returns and Jensen's Alpha with CAPM. An example of the final dataset is presented in Figure 24.

Figure 24 - **Examples from trading dataset after calculating returns**



	date_time	text	company	RIC	beta	Predicted	id	pre_price	post_price	rm	returns_pm	returns_index	returns_alpha
438	2020-01-12 08:00:00	• Bolaget har lämnat in svar till europeisk...	hansa biopharma ab	HNSA.ST	-1.231059	Up	1.58884e+12	73.1	71.55	0.000792001	0.978796	0.978004	0.979771
424	2020-01-15 08:00:00	Intensifierade åtgärder i ett fortsatt tufft r...	venue retail group ab	VRGb.ST	1.224038	Down	1.58885e+12	0.63	0.648	-0.00649478	0.972222	0.965727	0.964272
419	2020-01-16 19:00:00	Swedish Orphan Biovitrum AB (publ) (https://ww...	swedish orphan biovitrum ab	SOBIV.ST	1.707773	Up	1.58885e+12	190.05	186.35	-0.0030572	0.980531	0.983589	0.985752
407	2020-01-20 08:00:00	Proact redovisar idag preliminära siffror för ...	proact it group ab	PACT.ST	0.683689	Down	1.58913e+12	164.8	163	-0.0030572	1.01104	1.00799	1.00895
390	2020-01-21 19:00:00	Ortivus AB Finansiell kalender FINANSIELL KALE...	ortivus ab	ORTIb.ST	-0.107469	Up	1.58885e+12	4.23	4.24	0.000792001	1.00236	1.00157	1.00245

Figure 24 depicts an example of the trading dataset with stock movement predictions form the Machine Learning model and returns calculated using the three different methods; Returns, Market Adjusted Returns and Jensen's Alpha.

5.4.2.1. Results using tick data

The returns, using minute data and 60 second delay with the probability threshold of 44 percent, are cumulated and the results are presented in Figure 25. Additionally, the same data and delay but with default (33%) probability threshold is presented in Figure 26 for comparison. The results show that the three methods for calculating returns perform similarly and we thus only present Jensen's Alpha going forward when comparing different delays and probability thresholds.

Figure 25 - **Cumulative returns using tick data and 44 percent probability threshold**

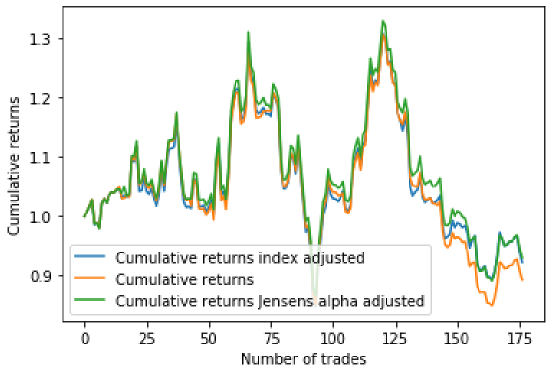


Figure 25 depicts the cumulated returns of the simulated trading strategy using 44 percent probability threshold and a 60 second delay from publication to trade with tick data. Results using Returns, Market Adjusted Returns and Jensen's Alpha is presented.

Figure 26 - **Cumulative returns using tick data and default probability threshold**

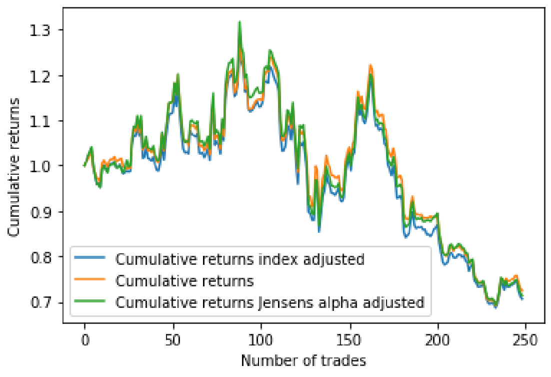


Figure 26 depicts the cumulated returns of the simulated trading strategy using default probability threshold and a 60 second delay from publication to trade with tick data. Results using Returns, Market Adjusted Returns and Jensen's Alpha is presented.

Additionally, the trading strategy is simulated using different delays between the time of publication of an ad-hoc disclosure and the time the trade is initiated. The results are presented in Figure 27 below and show promising results and indicate that a shorter delay yields higher returns. However, as presented in Table 23 we cannot say that the simulations achieve abnormal returns that are significantly greater than zero as the T value is below 1,645.

Figure 27 - **Cumulative returns from the trading strategy with tick data and different delays using Jensen's Alpha**

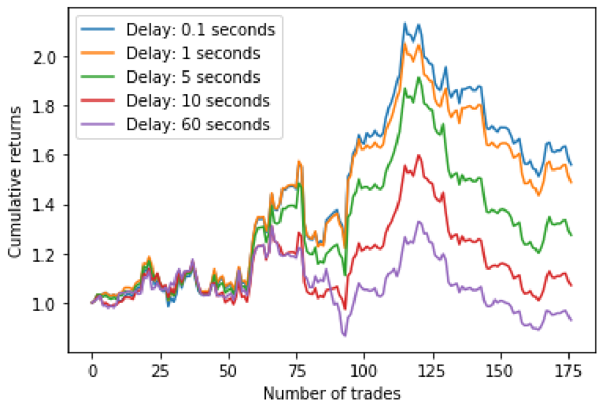


Figure 27 shows cumulated returns from the trading strategy using 44 percent probability threshold and various delays between publication of an ad-hoc disclosure and initiation of the trade.

Table 23 - **Results from the simulated trading strategy with tick data using different delays**

Probability threshold	**Standard deviation**	**Mean abnormal returns**	**T value**	**Number of trades**	**Delay**
44 percent	2,9505	0,002	0,008791	176	60 sec
44 percent	3,0797	0,0857	0,369337	176	10 sec
44 percent	3,3785	0,1935	0,759678	176	5 sec
44 percent	3,4347	0,2837	1,095832	176	1 sec
44 percent	3,4621	0,3112	1,192329	176	0,1 sec

Table 23 shows the results from simulating a trading strategy using the Machine Learning model with tick data and different delays between publication of an ad-hoc disclosure and initiation of the trade. The standard deviation, mean value and T-value of the abnormal returns generated by the trading strategy is presented as well as the number of trades performed. None of the results are significant on a 5 percent level.

5.4.2.2. Results using hourly data

As 40% of the instances are removed in the trading dataset using minute data due to missing data in the database as previously mentioned we also use hourly data to test the trading strategy. The cumulated returns using the 44 percent probability threshold are presented in Figure 28 and results using the default (33%) threshold are presented in Figure 29.

Figure 28 - **Cumulative returns using hourly data and 44 percent probability threshold**

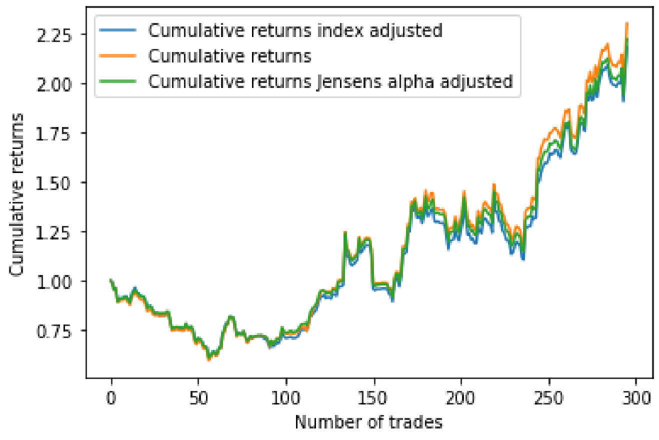


Figure 28 depicts the cumulated returns of the simulated trading strategy using 44 percent probability threshold using hourly data. Results using Returns, Market Adjusted Returns and Jensen's Alpha is presented.

Figure 29 - **Cumulative returns using hourly data and default probability**

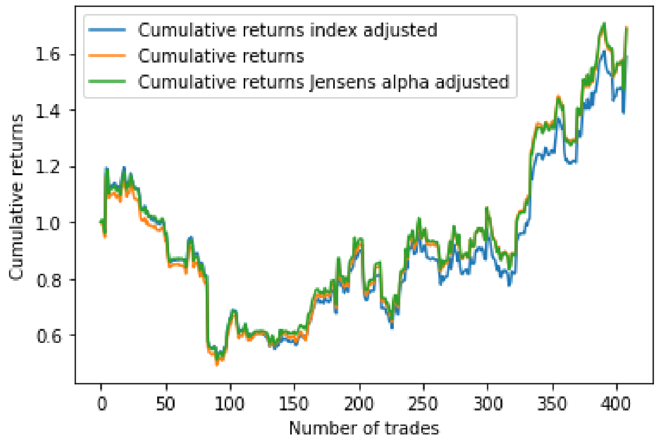


Figure 29 depicts the cumulated returns of the simulated trading strategy using default probability threshold using hourly data. Results using Returns, Market Adjusted Returns and Jensen's Alpha are presented.

The cumulated returns show promising results, however as presented in Table 24 we cannot say that these results are statistically significantly larger than zero as the T value is below 1,645 for the 5 percent significance level.

Table 24 - **Results from the trading strategy with hourly data**

Probability threshold	**Standard deviation**	**Mean abnormal returns**	**T value**	**Number of trades**
33 %	3,7719	0,1995	1,068166	408
44 %	3,5370	0,3315	1,609662	295

Table 24 shows the results from simulating a trading strategy using the Machine Learning model with hourly data. The standard deviation, mean value and T-value of the abnormal returns generated by the trading strategy is presented as well as the number of trades performed. None of the results are significant on a 5 percent level.

5.5. Summary of results of simulating an algorithmic Trading Strategy – H2

Applying the machine learning model from chapter four on our trading dataset using tick stock data show promising results where several of the tested trading algorithms (with different delays and thresholds) have positive cumulated returns over the trading period. However, these results are not statistically significantly greater than zero on a 5 percent significance level and we can thus not assume that these results are not due to the variance in stock movements. Additionally, we apply the same trading strategy but with hourly data as 40 percent of tick stock data is not available. Using the trading strategy with hourly data yield similar results to using minute data with positive cumulated returns but these results are not statistically significantly greater than zero either.

5.5.1. Answering H2

H2: *Since the stock market reflects new ad-hoc information in the stock price with a time lag, the model can be used as a part of a trading strategy on the Swedish stock market for generating abnormal returns.*

Based on the results from the trading strategy we do generate abnormal returns for many of the simulated trading strategies, but we cannot accept H2 as none of the results are statistically significantly greater than zero on a 5 percent significance level.

6. Discussion

6.1. Introduction

The discussion is first divided for discussing the two hypotheses separately. The discussions are subsequently merged to summarize the section and discuss the hypotheses in relation to each other and the implications of the combined results.

Research Question:

Will a machine learning model built on textual corporate ad-hoc disclosures from the Swedish market exceed relevant baselines and generate positive abnormal returns used as a trading strategy?

The two hypotheses embody the research question through separating it at the most natural distinction between machine learning modeling and trading strategy application while being naturally contingent on each other. The research question is answered in the end of the discussion section.

6.2. H1 discussion

Utilizing textual data in the financial sector is evidently an area that is gaining attention. Multiple studies show that there is a considerable number of different types of relevant text documents in finance that can be mined and carry relevant information for different purposes, such as stock valuation. This study investigates corporate ad-hoc disclosures, which represent documents that are inherently written to convey new, value relevant information for investors to act upon. The purpose of the documents, along with that they follow strict laws regarding equal access of market participants, gives the documents clear and undoubted theoretical and practical connection to stock price valuation.

Studies on other markets show promising results when studying and building machine learning models based on corporate disclosures (Balakrishnan et al., 2010; Feuerriegel & Gordon, 2018; Groth

& Muntermann, 2011; Kim et al., 2018; Muntermann & Guettler, 2007; Rekabsaz et al., 2017). Since no other study seems to have analyzed and constructed a supervised machine learning model from corporate ad-hoc disclosures on the Swedish market, while those studies recommend replication on new markets, it gives the study a feasible delimitation that is adopted. As established from building a machine learning model, and expected, the studied documents carry value relevant information and demonstrate predictive power to their related stock prices. This shows that the Swedish market works similarly to other markets where research has been made on ad-hoc disclosures which is the basis for confirming Hypothesis 1.

The machine learning model is however not easily compared with those of other studies. The specific research area of predicting stock prices with ad-hoc disclosures with text mining techniques is not yet thoroughly investigated, why a standardized way of comparing results with other studies is lacking. For example, Kim et al., (2018) utilize a two-way classification task with only 'Up' and 'Down' predictions whereas this study uses three classes. Balakrishnan et al., (2010) use three classes similar to this study, but evaluate their results from the perspective of reducing forecast errors, whereas this study evaluates the machine learning model from a prediction accuracy perspective.

Also, different studies use different baselines, and the datasets used by the different studies differ vastly. For example, Kim et al., (2018) exceed their baseline of unigram Logistic Regression with 13,89 percentage points (Leverage), compared to this study's 6,321 leverage over the ZeroR baseline. But since Kim et al. (2018) only have observations from four companies and discriminate the models per sector while counting sentiment scores, it is difficult to compare with this study's several hundred companies and almost 30 000 observations. Still, and as aforementioned, the results presented in this study point in the same direction as those of other studies by showing another proof of concept when it comes to building a machine learning model based on ad-hoc disclosures with predictive power.

State-of-the-art text analysis techniques such as word- and document embeddings (dense vectors) are in this study compared with the traditional bag of words method Term Frequency – Inversed Document Frequency (TF-IDF, sparse vectors). Like the classifier comparison with Logistic Regression, the more simplistic pre-processing method performs surprisingly well with an accuracy above the best-performing state-of the art embeddings techniques, including BERT with a pre-trained Swedish vocabulary. One potential answer to why this might be is that the pre-trained embeddings are trained on different types of text data that might significantly differ from ad-hoc disclosures. Words in a financial setting might thus convey a different message that in normal text document. Kim et al. (2018) theorizes that financial text data conveys different meanings across industries. It is thus reasonable to assume that this applies for financial text data in general compared to e.g. Wikipedia or general news which most embeddings are trained on. However, the Doc2Vec pre-processing technique, trained on the in-house dataset, still does not exceed the performance of TF-IDF. In conclusion, the more advanced semantic information that is acquired from the dense embeddings vectors when predicting does not help the model. It should however be mentioned again that parts of the results also can be assigned to the slight bias after the hyperparameter search for the best dataset that was made with TF-IDF, standard settings, pre-processing.

In the learning, the main difference from other studies is in the supervised labeling of instances, where we use financial theory to reduce noise when trying to compute the actual stock price effect that a certain ad-hoc disclosure can be said to explain. More specifically, financial theory is utilized for motivating a comparison between measuring price movements as returns or as abnormal returns, where abnormal returns are risk adjusted returns both in terms of the market-adjusted returns model, and as Jensen’s Alpha. The idea is that finance theory-based calculations should result in improved learning through adjusting for risk factors, thereby reflecting the effect of an ad-hoc disclosure onto the price movement more correctly. As the results show, especially Jensen’s Alpha abnormal returns seem to help the machine learning model in the learning process. In addition, the largest learning gains from the financial models are observed for the smallest classification cut-off threshold, 0,5%, where the learning gains by the financial models is to a high degree diminished for higher thresholds. This could be a result of that the machine learning model benefits from the improved isolation of the stock price effect on ad-hoc disclosures provided by the financial models, but that the effect is diminished when the Stable class becomes more prevalent. This is likely the biggest research contribution in the study, which also implies that more can be done to further develop the labeling process in the supervised learning for similar studies.

In summary, the use of textual ad-hoc disclosures when building a machine learning model for stock price prediction has been proven a useful source and this study confirms what several other studies conclude. It is interesting that the relatively simple pre-processing techniques and machine learning models perform better than the more advanced, state-of-the-art models and techniques do. Financial theory seems to improve the labeling process which is an interesting insight.

6.2.1. Limitations in the machine learning modeling

As one of this study’s contribution to the area is through using relatively sophisticated labeling, it is worth mentioning that from a financial theory perspective, Jensen’s Alpha in the context of CAPM is still a rather simplistic model with well-known successors model wise. The labeling process could be improved by extending the estimation period of normal returns which was relatively short as we wanted to keep as many instances as possible. Furthermore, the labeling process could also benefit from using more factors that adjust risk in stock returns. The three- and five-factor models would for instance be a natural improvement to add for studies with similar purposes. With Jensen’s Alpha that adjust for fewer risk factors than the more sophisticated models, it is still interesting to see that the best result for that labeling model was at the lowest threshold of 0,5 %. As all models results of abnormal returns are split between the model's weaknesses and actual abnormal returns, there is good reason to assume model weakness in Jensen's Alpha which motivates a threshold > 0. Even lower thresholds than 0,5 % could however have been tested as the lowest tested threshold returned the highest leverage against the baseline.

As there naturally is a focus on collecting many instances of data, it came with the implication of only accessing daily data 10 years back. This means that some assumptions about how to measure the effects around an event had to be made which with more specific data could have been further improved to isolate the event and stock price movements. This gives the relatively sophisticated labeling method of Jensen's Alpha suboptimal settings to perform which further likely impacted results negatively in the supervised learning.

Since all corporate disclosures have been scraped from the Nasdaq webpage, not all documents might fit the perfect definition of an ad-hoc disclosure; some of the corporate disclosures used in the study might have been pre-announced, while other documents contain little to no information. However, since pre-announced documents should have already been somewhat adjusted for by the market, as theorized, and since the supervised learning is set up as a multiclass problem, such documents should not convey a large price movement, thus being classified as Stable. However, speculating about the amount of "bad" documents is not very fruitful, why it is a weakness that we deemed it too time consuming to look more closely at the data and perhaps find some rule to exclude documents that do not meet the criteria for being an ad-hoc disclosure relevant for the study.

In this study, several steps for creating the dataset and building the models are conducted. We prioritized proceeding the study rather than spending a lot of time on each step. The focus was on a more holistic level, such as building the dataset and calculating different price movement measurements, to eventually be able to compare labeling procedures, classifiers, and embeddings to TF-IDF, and finally simulate a test of the model on the market. Looking back, some steps that are conducted in the early stages can be improved for future research. More precisely, more time for cleaning the documents could potentially improve the models. Utilizing techniques such as stemming, and lemmatization could also have been tried but was excluded for time and scope reasons and time was instead focused on more state-of-the art pre-processing. The largest difference from additional pre-processing might be to reduce the time it takes for building the models, since the models will not have to spend time processing tokens that carry very little to no relevant information.

Another limitation is that this study does not take into account that financial text documents and the content of them might have significantly different meanings depending on industry and sector. Thus, an improvement could be to construct models per industry especially when using dense embeddings.

When matching documents downloaded from Nasdaq with stock data downloaded from Refinitiv Eikon, it is evident that the two services do not always write companies' names identically. To match the non-identical strings with each other, we pre-process the strings and subsequently match on partial string. Because of this, when using the finalized model for trading, 316 instances out of the original 27 211 instance dataset appear to be matched with incorrect stock price data. These are approximately 1,6 percent of the total number of instances, why the effect of these incorrect instances on the machine learning model's ability to learn seems small. Still, the model could have improved its learning if the matching had been even more correct or the wrongly matched instances would have been eliminated.

6.3. H2 discussion

With the assumption of a semi-strong market, new information should be reflected in the stock price immediately after being published, but some studies suggest that at least in relation to new, ad-hoc, information there could exist a structural time-lag. The advantage of using ad-hoc disclosures with a supervised learning approach is that the forecasts potentially can detect market movements that are too complex for humans to grasp and also, sometimes, detect market movements humans will grasp, just faster. It should however be very clear that any trading algorithm expecting, or at least hypothesizing, to beat the market timewise does nothing short of making a bold claim. However, these kind of bold, theory-backed claims is what pushes the boundaries of stock market research and has the possibility to detect market inefficiencies.

The prerequisites for the trading algorithm was however never to always beat the market, rather that it on average potentially could capture parts of a stock price movement going to its new equilibrium. This implies that sometimes the timing would be good, and sometimes it could be worse, but given a correct classification of the movement, the algorithm should not lose a significant amount on any "correct" action. If the price in a certain trade is adjusted instantly (thus not being able to capture the price movement), the strategy would not lose more money than trading costs. If some classifications render in false prediction of the opposite price movement, we will get negative abnormal returns.

The results show that the algorithm at least does not lose money with the cumulated returns of the simulation, using tick data, 44 percent probability threshold and a delay of 5 seconds, of roughly 50 % and an average return per action of 0,28 percent of the ca 176 actions. The results are, as aforementioned, although not significant on a 5 percent significance level as there is a high variance in the returns. If the results indicate that the algorithm could earn money is hence a difficult question. Beforehand, a very high variance is expected as the algorithm's purpose in a sense is to time the market as it starts to move, to soon after fully reflect new information. Also, as the mean precision of 'Up' and 'Down' is 46,39 percent using the 44 percent probability threshold on our test dataset we expect the trading algorithm to make a significant amount of false classifications. However, we only need to make positive abnormal returns on average to earn money in the long run. Thus, a mean precision below 50 percent does not necessarily mean that we cannot make any money, because some of the false 'Up' or 'Down' will actually be 'Stable' and thus only incur trading cost and a very small stock return (positive or negative). The high variance although makes the abnormal returns alpha not significantly larger than zero. To draw any conclusions from the abnormal returns are hence hard. It is perhaps not very controversial to claim that any investor would be very pleased to have a return on investment of 50 percent in a three month-period, but as there is a lack of statistical significance, the results are too likely to be a consequence of sheer luck, and no definitive conclusion can therefore be drawn.

What the results partially indicate, especially when looking at returns when using tick stock data and different delays, is that the Swedish stock market seems to be more efficient than for instance the German market where the average time lag of 30 minutes was observed. This implies that a trading algorithm using ad-hoc disclosures will have a tough time generating abnormal returns systematically when a new document is released. The Swedish stock market seems to approximately be efficient according to the semi-strong notion of the Efficient Market Hypothesis. Returns however seem to increase with a shorter time delay on the same data but as the results are not significantly greater than zero, we cannot conclude any minimal time-lag either. We do not perform any significance tests between the results using different time lags either.

This research setting is highly challenging as anyone learning how to beat the stock market almost impossibly could continue beating the market in perpetuity as new techniques and knowledge about markets are quickly absorbed by trading agents. If the results of this study to someone would seem interesting enough to try on the market and it delivered some kind of abnormal return on average when tested, it would likely not take long before the time lag gets diminished by agents trading with the same insight/algorithm and only the traders with the fastest algorithm would earn abnormal returns. For research purposes however, learning about the stock market provides interesting insights, and perhaps what is learnt with H1 provides interesting insight about the disclosures being published every day that carry value relevant information for a machine learning model to learn.

As the portfolio generated a rather large return on investment given the period of time, whether being lucky or not, it would be interesting to test it on the market with actual trading system integration and implementation as it at least does not seem to lose money. That the algorithm is relatively risk averse comes from the fact that a number of false ‘Up’ and ‘Down’ belong to the Stable class as well as the fact that the threshold for classifying an instance as ‘Up’ or ‘Down is higher than for ‘Stable. This implies no action and a stay out of market strategy. As the algorithm can develop and learn more from new documents while improving some aspects, it might as a trading strategy be able to form a basis for a future model that could generate significant abnormal returns.

In summary, the trading algorithm tried its best in the simulation and did generate abnormal returns, that although were not significant enough to claim their existence as an effect of the machine learning model empowered trading strategy. This shows no proof that the Swedish market has structural time lag for reflecting new ad-hoc disclosures in stock prices. The algorithm does however not lose any money and as it is relatively risk-averse, it could be further improved as accuracy clearly has room for improvement to rerun simulations and see if it then more conclusively can be said to have generated positive abnormal returns.

6.3.1. Limitations in the trading strategy simulation

The trading dataset works in practice as a “final test set” of hold-out data with 461 instances, that the classifier predicts as either ‘Up’ or ‘Down’ which implies action, or ‘Stable’ which does not trigger any action. Something that could have been improved in the simulation is a larger data set with tick stock data. A larger data set would have given the algorithm a greater opportunity to be tested and a better foundation to test whether the algorithm can succeed in generating significant positive abnormal returns. If an extended test with 1 000 observation/actions would show a similar average return of the actions and a similar standard deviation for Alpha, the results could have been significant on a 5 percent level. Of course, it is difficult to know if the same pattern would occur and is mostly speculation at this stage, but nonetheless more data to validate the trading strategy would help the evaluation to be more conclusive. This especially as the algorithm reached results close to being significant.

It is interesting that the higher probability threshold that pushes the number predictions towards the Stable class as expected generates a higher T-statistic value and abnormal returns even though it still is not significant. If an additional data validation set with tick data could have been created for the trading strategy, it would be interesting to grid search different probability thresholds and not just look at the Up and Down Precision vs number of trades on the test dataset from H1. This could help find and validate the best probability threshold that would decrease financial risk by pushing even more risky predictions towards Stable, where predictions are risk-free, while maximizing expected returns by acting on more certain predictions of Up and Down movements.

A concrete limitation for the trading strategy is that trading costs are not assumed or calculated. As the event study methodology sidestep trading and information costs to isolate and calculate abnormal returns with risk adjusting factors, this study follows best practice in finance. It should however be noted that the algorithmic trading strategy in this study is subject to potentially more transaction costs than other more long-term strategies as it is focused on making many actions even intraday. Many actions intraday have an impact on the trading costs that normally are a function of the number (and size) of transactions. Assuming that this trading strategy might be more costly than other more long-term strategies hence mean that any real implementation should take into consideration the exact trading cost that the trading platform has. With this computation it can be contrasted against the expected number of transactions during the period the strategy is supposed to be active and subtracted from any expected returns.

One of the simplifications of the trading strategy simulation is that an equivalent investment is assumed for all actions. This is not possible in practice as one must purchase at least one stock and the price for different stocks naturally differ significantly. An implementation in practice hence must consider the potential firms that the algorithm could invoke action for and proposedly develop a rule for a size-span of every investment decision based on the algorithm. Another important consideration for the trading strategy is that if for instance a small firm has a relatively low trading volume, an action could make the stock price change significantly. In an extreme case this could mean that it is the action of the trading strategy itself that forces a stock price to its new equilibrium which eliminates the return that it is supposed to capture. This would however not be an issue in mid- to large-cap companies at least on the Swedish market as they have high trading volumes on average.

6.4. Summarizing discussion of H1 and H2

In summary, the study has answered and discussed the two hypotheses that are naturally related in the sense that H2 is mainly contingent on the foundation from H1 while however having more business-related challenges to approach. Building the study from the perspective that H1 is prioritized to focus on the machine learning domain to create an optimal model before simulating it as part of a trading strategy helps put focus on questions in the right order.

Finance theory and specifically theory of abnormal returns have impacted the supervised labeling process why finance and machine learning are prevalent throughout both hypotheses and impact different methodological choices. The purpose of H2 is to apply the business value aspect more concretely which for instance impacted how different errors that the algorithm can make are valued which in H1 did not impact modeling, because accuracy and leverage in the iteration processes were of highest importance. These perspectives complement each other as the trading strategy simulation in a sense becomes the final test for the model built upon theoretical frameworks of machine learning, text mining and finance.

6.5. Answering the research question

The research question was formulated as the following:

Can a machine learning model trained on textual corporate ad-hoc disclosures from the Swedish stock market exceed relevant baselines and generate positive abnormal returns used as a trading strategy?

The machine learning built on corporate ad-hoc disclosures in this study exceeds relevant baselines and generates positive abnormal returns, although not significantly larger than zero on a 5 percent level. We can thus only answer yes to the first part of the research question, and no on the second part.

6.6. Recommendations for future research

The research area of utilizing text mining techniques for ad-hoc disclosures for stock price prediction is an exciting area that with good reason should be further researched. There are many approaches and methodological choices possible and then different markets and alterations in the exact document are not even mentioned. Here, we list some interesting identified areas within the research topic that we recommend for future research.

One concrete way to build upon the labeling procedure is to utilize known and more sophisticated risk adjusting models than those used in this study, such as the Fama & French (1993, 2015) three-factor and five-factor models. This might result in an even better supervisory signal, further improving the machine learning model's ability to learn.

This thesis follows a three-class problem. The results of the comparison of different price movement measurements and classification cut-off thresholds however indicate that the machine learning model learns the most when using smaller classification thresholds. Therefore, testing the model's learning ability from even smaller thresholds, where one might also try a threshold of 0 (thus eliminating the Stable class), would be interesting.

With a bit more time, it would have been interesting to closer analyze the words and documents from a linguistic perspective, to see both which types of documents and which words that best predict stock prices. This analysis could be done in many layers and be very specific, or just through looking at some patterns in the data, perhaps using an unsupervised machine learning modeling method.

The Swedish stock market, at least from a published study perspective, seems rather unanalyzed regarding any text data as input to a supervised machine learning approach. Therefore, besides further improving analysis of ad-hoc disclosures on the Swedish market, text sources such as social media and financial news could also be analyzed and used for building stock price prediction classifiers.

In order to provide the model with better prerequisites to learn from the text data, building one model per some definition of sector can further increase the value of specific words in different sectors as they may across sectors convey different sentiments but be more similar within the same sector. This is the approach that (Kim et al., 2018) adopt but as they only study four firms, it would be interesting to on a bigger scale tag all firms on a certain market to belong to x number of sectors and then build one model per sector.

One interesting approach to improving the Doc2Vec would be to try a semi-supervised approach, similar to the approach used in the guide by S. Li (2018a). In the original Doc2Vec approach as presented by Le & Mikolov (2014), each document is tagged with a unique document vector, which in practice is a unique ID number per document. S. Li (2018a) instead tags each document with its respective class. Using the problem of this study as an example, each document would be tagged with either Up, Stable or Down, instead of a number. This approach is also used by Kim et al. (2018), in what they call the Supervised Paragraph Vector (S-PV) approach. This approach would have been preferable to test in the thesis, but was discovered when the modeling was almost finished, why re-running all models was not feasible.

A final area that could be interesting for future research is to differentiate between ad-hoc disclosures, since the content in them can differ significantly. Some ad-hoc disclosures that could be considered similar given some definition might carry relatively more predictive components to build a model from. Many ad-hoc disclosures carry little to no value, which in that case would allow the exclusion of them.

7. Conclusion

The study has built a machine learning classifier with corporate ad-hoc disclosures from the Swedish market through testing state-of the art methods in Natural Language Processing. As the classifier exceeds relevant baselines, it can be concluded that there is value relevant information that a machine learning model can learn from the ad-hoc disclosures published by Swedish firms. As no previous study have conducted a similar study on the Swedish market, this study has introduced these techniques on a new market and confirmed the relevance of text mining in finance and the value of ad-hoc disclosures as stock price predictors.

The assumption of a time-lag for the market to fully reflect the ad-hoc disclosures in the stock price on the Swedish market was not conclusively proven, why the semi-efficient market form still is a good approximation of the Swedish market efficiency. As a result of that, the simulated trading strategy did not earn significant abnormal returns. Still, the returns were high from a return on investment perspective in relation to the time-period that the simulation was active. This, which encourages further improvements of the model though correcting some limiting aspects that were discovered late.

This study's strongest contribution is the utilization of relatively sophisticated labeling methods based in financial theory which helps isolate the effect of a certain text document onto the price movement through reducing known risk factors. Even more sophisticated ways of measuring abnormal returns could further improve the supervised labeling to help create a model that could empower a successful algorithmic trading strategy.

References

Aktas, N., Bodt, E., & Cousin, J.-G. (2007). Event studies with a contaminated estimation period. *Journal of Corporate Finance*, *13*, 129–145. https://doi.org/10.1016/j.jcorpfin.2006.09.001

Balakrishnan, R., Qiu, X. Y., & Srinivasan, P. (2010). On the predictive ability of narrative disclosures in annual reports. *European Journal of Operational Research*, *202*(3), 789–801. https://doi.org/10.1016/j.ejor.2009.06.023

Ball, R., & Brown, P. R. (2013). Ball and Brown (1968): A Retrospective. *The Accounting Review*, *89*(1), 1–26. https://doi.org/10.2308/accr-50604

Bank, M., & Baumann, R. H. (2015). Market efficiency under ad hoc information: Evidence from Germany. *Financial Markets and Portfolio Management*, *29*(3), 173–206. https://doi.org/10.1007/s11408-015-0250-8

Baule, R., & Tallau, C. (2012). *Market Response to Mandatory Pre-Earnings-Announcements—Evidence from Ad-Hoc Disclosures in Germany* (SSRN Scholarly Paper ID 1660679). Social Science Research Network. https://doi.org/10.2139/ssrn.1660679

BERT FineTuning with Cloud TPU: Sentence and Sentence-Pair Classification Tasks. (n.d.). Google Cloud. Retrieved April 17, 2020, from https://cloud.google.com/tpu/docs/tutorials/bert?hl=sv

Chan, K., Ikenberry, D., & Lee, I. (2004). Economic Sources of Gain in Stock Repurchases. *Journal of Financial and Quantitative Analysis*, *39*(3), 461–479. https://doi.org/10.1017/S0022109000003987

Chen, J. (n.d.). *Market Neutral*. Investopedia. Retrieved May 14, 2020, from https://www.investopedia.com/terms/m/marketneutral.asp

Chordia, T., Goyal, A., Sadka, G., Sadka, R., & Shivakumar, L. (2009). Liquidity and the Post-Earnings-Announcement Drift. *Financial Analysts Journal*, *65*(4), 18–32. https://doi.org/10.2469/faj.v65.n4.3

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. *ArXiv:1407.0202 [Cs, Math, Stat]*. http://arxiv.org/abs/1407.0202

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2020). *TensorFlow code and pre-trained models for BERT* [Python]. Google Research. https://github.com/google-research/bert (Original work published 2018)

Dische, A. (2002). Dispersion in Analyst Forecasts and the Profitability of Earnings Momentum Strategies. *European Financial Management*, *8*(2), 211–228. https://doi.org/10.1111/1468-036X.00185

dos Santos Pinheiro, L., & Dras, M. (2017a). Stock Market Prediction with Deep Learning: A Character-based Neural Language Model for Event-based Trading. *Proceedings of the Australasian*

Language Technology Association Workshop 2017, 6–15. https://www.aclweb.org/anthology/U17-1001

dos Santos Pinheiro, L., & Dras, M. (2017b). Stock Market Prediction with Deep Learning: A Character-based Neural Language Model for Event-based Trading. *Proceedings of the Australasian Language Technology Association Workshop 2017*, 6–15. https://www.aclweb.org/anthology/U17-1001

Dyer, T., Lang, M., & Stice-Lawrence, L. (2016). Do managers really guide through the fog? On the challenges in assessing the causes of voluntary disclosure. *Journal of Accounting and Economics*, *62*(2), 270–276. https://doi.org/10.1016/j.jacceco.2016.08.001

Engelberg, J. (2008). Costly Information Processing: Evidence from Earnings Announcements. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1107998

European Parliament. (2014, April 16). *REGULATION (EU) No 596/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL*. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32014R0596&from=SV#d1e2510-1-1

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work*. *The Journal of Finance*, *25*(2), 383–417. https://doi.org/10.1111/j.1540-6261.1970.tb00518.x

Fama, E. F. (1991). *Efficient Capital Markets: II - FAMA - 1991—The Journal of Finance—Wiley Online Library*. https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1540-6261.1991.tb04636.x

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

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, *116*(1), 1–22. https://doi.org/10.1016/j.jfineco.2014.10.010

Fares, M., Kutuzov, A., Oepen, S., & Velldal, E. (2017). *Word vectors, reuse, and replicability: Towards a community repository of large-text resources*. 6.

Fawcett, T., & Provost, F. (2013). *Data Science for Business*.

Feinerer, I., Hornik, K., & Meyer, D. (2008). Text Mining Infrastructure in *R*. *Journal of Statistical Software*, *25*(5). https://doi.org/10.18637/jss.v025.i05

Feuerriegel, S., & Gordon, J. (2018). Long-term stock index forecasting based on text mining of regulatory disclosures. *Decision Support Systems*, *112*, 88–97. https://doi.org/10.1016/j.dss.2018.06.008

Fisher, I. E., Garnsey, M. R., & Hughes, M. E. (2016). Natural Language Processing in Accounting, Auditing and Finance: A Synthesis of the Literature with a Roadmap for Future Research. *Intelligent Systems in Accounting, Finance and Management*, *23*(3), 157–214. https://doi.org/10.1002/isaf.1386

Floridi, L. (2010). *Information: A Very Short Introduction*. Oxford University Press.

Foong, N. W. (2019, December 2). *Beginner's Guide to BERT for Multi-classification Task*. Medium. https://towardsdatascience.com/beginners-guide-to-bert-for-multi-classification-task-92f5445c2d7c

Gensim. (2019). *Gensim: Doc2Vec Wrapper for Sci-Kit Learn*. https://radimrehurek.com/gensim/sklearn_api/d2vmodel.html

Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research, 57*, 345–420.

Google Code Archive. (2013). *Word2Vec*. https://code.google.com/archive/p/word2vec/

Groth, S. S., & Muntermann, J. (2011). An intraday market risk management approach based on textual analysis. *Decision Support Systems*, *50*(4), 680–691. https://doi.org/10.1016/j.dss.2010.08.019

Hájek, P. (2018). Combining bag-of-words and sentiment features of annual reports to predict abnormal stock returns. *Neural Computing and Applications*, *29*(7), 343–358. https://doi.org/10.1007/s00521-017-3194-2

Hajek, P., & Barushka, A. (2018). Integrating Sentiment Analysis and Topic Detection in Financial News for Stock Movement Prediction. *Proceedings of the 2nd International Conference on Business and Information Management*, 158–162. https://doi.org/10.1145/3278252.3278267

Hew, D., Skerratt, L., Strong, N., & Walker, M. (1996). Post-earnings-announcement Drift: Some Preliminary Evidence for the UK. *Accounting and Business Research*, *26*(4), 283–293. https://doi.org/10.1080/00014788.1996.9729519

Ibbotson, R. G. (1975). Price performance of common stock new issues. *Journal of Financial Economics*, *2*(3), 235–272. https://doi.org/10.1016/0304-405X(75)90015-X

Ikenberry, D., Lakonishok, J., & Vermaelen, T. (1995). Market underreaction to open market share repurchases. *Journal of Financial Economics*, *39*(2), 181–208. https://doi.org/10.1016/0304-405X(95)00826-Z

Investopedia. (2019, September 19). *Quarterly Report Definition*. https://www.investopedia.com/ask/answers/122214/what-quarterly-report.asp

Jensen, M. C. (1967). *The Performance of Mutual Funds in the Period 1945-1964* (SSRN Scholarly Paper ID 244153). Social Science Research Network. https://doi.org/10.2139/ssrn.244153

Kearney, C., & Liu, S. (2014). Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, *33*, 171–185. https://doi.org/10.1016/j.irfa.2014.02.006

Kenton, W. (2019a, May 24). *Annual Report Definition*. https://www.investopedia.com/terms/a/annualreport.asp

Kenton, W. (2019b, June 1). *10-K Definition*. https://www.investopedia.com/terms/1/10-k.asp

Kim, M., Park, E. L., & Cho, S. (2018). Stock price prediction through sentiment analysis of corporate disclosures using distributed representation. *Intelligent Data Analysis*, *22*(6), 1395–1413. https://doi.org/10.3233/IDA-173670

Kothari, S. P. (2001). Capital markets research in accounting. *Journal of Accounting and Economics*, *31*(1), 105–231. https://doi.org/10.1016/S0165-4101(01)00030-1

Le, Q., & Mikolov, T. (2014). *Distributed Representations of Sentences and Documents*. 9.

Li, Qing, Jiang, L., Li, P., & Chen, H. (2015, February 18). Tensor-Based Learning for Predicting Stock Movements. *Twenty-Ninth AAAI Conference on Artificial Intelligence*. Twenty-Ninth AAAI Conference on Artificial Intelligence. https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9409

Li, Quanzhi, & Shah, S. (2017). Learning Stock Market Sentiment Lexicon and Sentiment-Oriented Word Vector from StockTwits. *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 301–310. https://doi.org/10.18653/v1/K17-1031

Li, S. (2018a, December 4). *Multi-Class Text Classification with Doc2Vec & Logistic Regression*. Medium. https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4

Li, S. (2018b, December 6). *Multi-Class Text Classification Model Comparison and Selection*. Medium. https://towardsdatascience.com/multi-class-text-classification-model-comparison-and-selection-5eb066197568

Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, *69*, 14–23. https://doi.org/10.1016/j.knosys.2014.04.022

Liberti, J. M., & Petersen, M. A. (2018). *Information: Hard and Soft*. 57.

Liddy, E. D. (2001). *Natural Language Processing*. *2001*, 15.

Liu, W., Strong, N., & Xu, X. (2003). Post–earnings–announcement Drift in the UK. *European Financial Management*, *9*(1), 89–116. https://doi.org/10.1111/1468-036X.00209

MacKinlay, A. C. (1997). *Event studies in economics and finance*. 27.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ArXiv:1301.3781 [Cs]*. http://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *ArXiv:1310.4546 [Cs, Stat]*. http://arxiv.org/abs/1310.4546

Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists* (1 edition). O'Reilly Media.

Muntermann, J., & Guettler, A. (2007). Intraday stock price effects of ad hoc disclosures: The German case. Journal of International Financial Markets, Institutions and Money, 17(1), 1–24. https://doi.org/10.1016/j.intfin.2005.08.003

Nabi, J. (2018, October 5). Machine Learning—Word Embedding & Sentiment Classification using Keras. Medium. https://towardsdatascience.com/machine-learning-word-embedding-sentiment-classification-using-keras-b83c28087456

National Library of Sweden. (2020, February 4). KB tillgängliggör kraftfulla modeller för språkförståelse [Text]. KB. https://www.kb.se/samverkan-och-utveckling/nytt-fran-kb/nyheter-samverkan-och-utveckling/2020-02-04-kb-tillgangliggor-kraftfulla-modeller-for-sprakforstaelse.html

Nguyen, T. H., & Shirai, K. (2015). Topic Modeling based Sentiment Analysis on Social Media for Stock Market Prediction. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1354–1364. https://doi.org/10.3115/v1/P15-1131

Nugroho, P. (2019, June 29). *Saving Data as JSON in Unity*. Medium. https://medium.com/@prasetion/saving-data-as-json-in-unity-4419042d1334

Nuti, G., Mirghaemi, M., Treleaven, P., & Yingsaeree, C. (2011). Algorithmic Trading. *Computer*, *44*(11), 61–69. https://doi.org/10.1109/MC.2011.31

O'Connor, G. J. (2009). (54) SYSTEMAND METHOD FOR EVENT-BASED. *2009*, *2009*, 18.

Patel, A., Brooks, R. M., & Patel, A. (2003). *Journal of Business, 2003, vol. 76, no. 1)* (Vol. 2003).

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Peyer, U., & Vermaelen, T. (2009). The Nature and Persistence of Buyback Anomalies. *The Review of Financial Studies*, *22*(4), 1693–1745. https://doi.org/10.1093/rfs/hhn024

Qian, J. (2019). *An Introduction to Asset Pricing Theory*. 104.

Qin, Y., & Yang, Y. (2019). What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 390–401. https://doi.org/10.18653/v1/P19-1038

Řehůřek, R. (2019a). *Gensim*. https://radimrehurek.com/gensim/about.html

Řehůřek, R. (2019b). *Gensim: Doc2Vec*. https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html#sphx-glr-auto-examples-tutorials-run-doc2vec-lee-py

Rekabsaz, N., Lupu, M., Baklanov, A., Hanbury, A., Duer, A., & Anderson, L. (2017). Volatility Prediction using Financial Disclosures Sentiments with Word Embedding-based IR Models. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1712–1721. https://doi.org/10.18653/v1/P17-1157

Russell, S. J., Norvig, P., & Davis, E. (2010). *Artificial intelligence: A modern approach* (3rd ed). Prentice Hall.

Schumaker, R. P., Zhang, Y., Huang, C.-N., & Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems*, *53*(3), 458–464. https://doi.org/10.1016/j.dss.2012.03.001

Sci-Kit. (2019a). *1.1. Linear Models—Scikit-learn 0.22.2 documentation*. https://scikit-learn.org/stable/modules/linear_model.html#id27

Sci-Kit. (2019b). *sklearn.linear_model.LogisticRegression—Scikit-learn 0.22.2 documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Sci-Kit. (2019c). *sklearn.linear_model.SGDClassifier—Scikit-learn 0.22.2 documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

Sci-Kit. (2019d). *Sklearn.svm.LinearSVC — scikit-learn 0.22.2 documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

Sci-Kit. (2019e). *Sklearn.svm.SVC — scikit-learn 0.22.2 documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Sci-Kit. (2020). *Non-linear SVM — scikit-learn 0.22.2 documentation*. https://scikit-learn.org/stable/auto_examples/svm/plot_svm_nonlinear.html

Sewell, M. (2011). History of the Efficient Market Hypothesis. *2011*, *2011*, 14.

Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk*. *The Journal of Finance*, *19*(3), 425–442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Stephantt, J. (1984). *A Comparison of Event Study Methodologies Using Daily Stock Returns: A Simulation Approach THOMAS DYCKMAN, * DONNA PHILBRICK,f AND*.

Straka, M., & Straková, J. (2016). UDPipe. *Http://Ufal.Mff.Cuni.Cz/Udpipe*. https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1702

Van Leeuwen, D., & Brummer, N. (2006). Channel-dependent GMM and Multi-class Logistic Regression models for language recognition. *2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*, 1–8. https://doi.org/10.1109/ODYSSEY.2006.248094

Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2016). Charagram: Embedding Words and Sentences via Character n-grams. *ArXiv:1607.02789 [Cs]*. http://arxiv.org/abs/1607.02789

Xu, Y., & Cohen, S. B. (2018). Stock Movement Prediction from Tweets and Historical Prices. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1970–1979. https://doi.org/10.18653/v1/P18-1183

Young, J. (2019, May 2). *Understanding Market Indexes and Their Uses Helps Investors*. Investopedia. https://www.investopedia.com/terms/m/marketindex.asp

Appendices

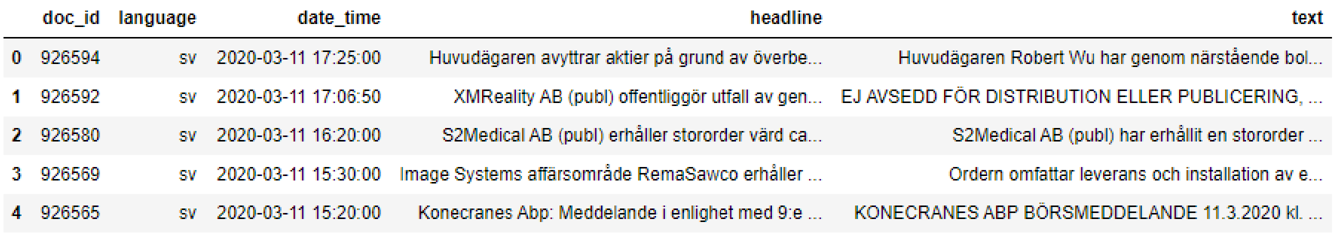
Appendix 1 - Code

All code is available on Github, accessible through this link:

https://github.com/AlgoTrading2020/Thesis2020

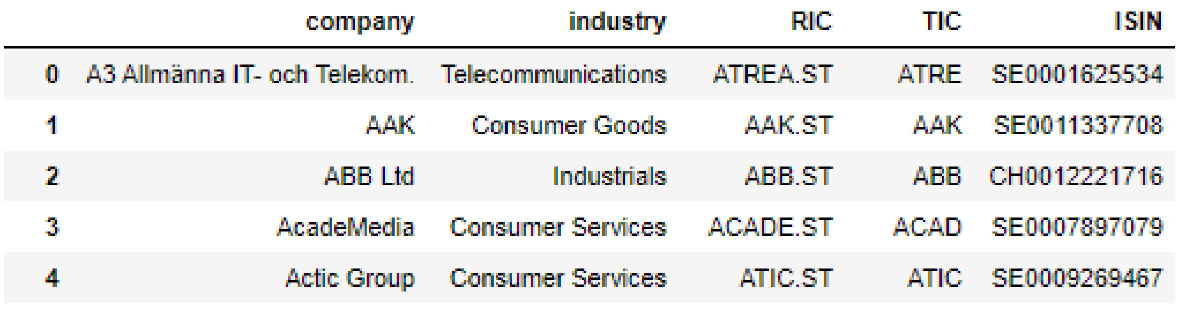
Appendix 2 - Example Figures from Experiment H1

Figure 30 - **Example of five scraped ad-hoc disclosures. The total number of scraped documents are 88 804.**



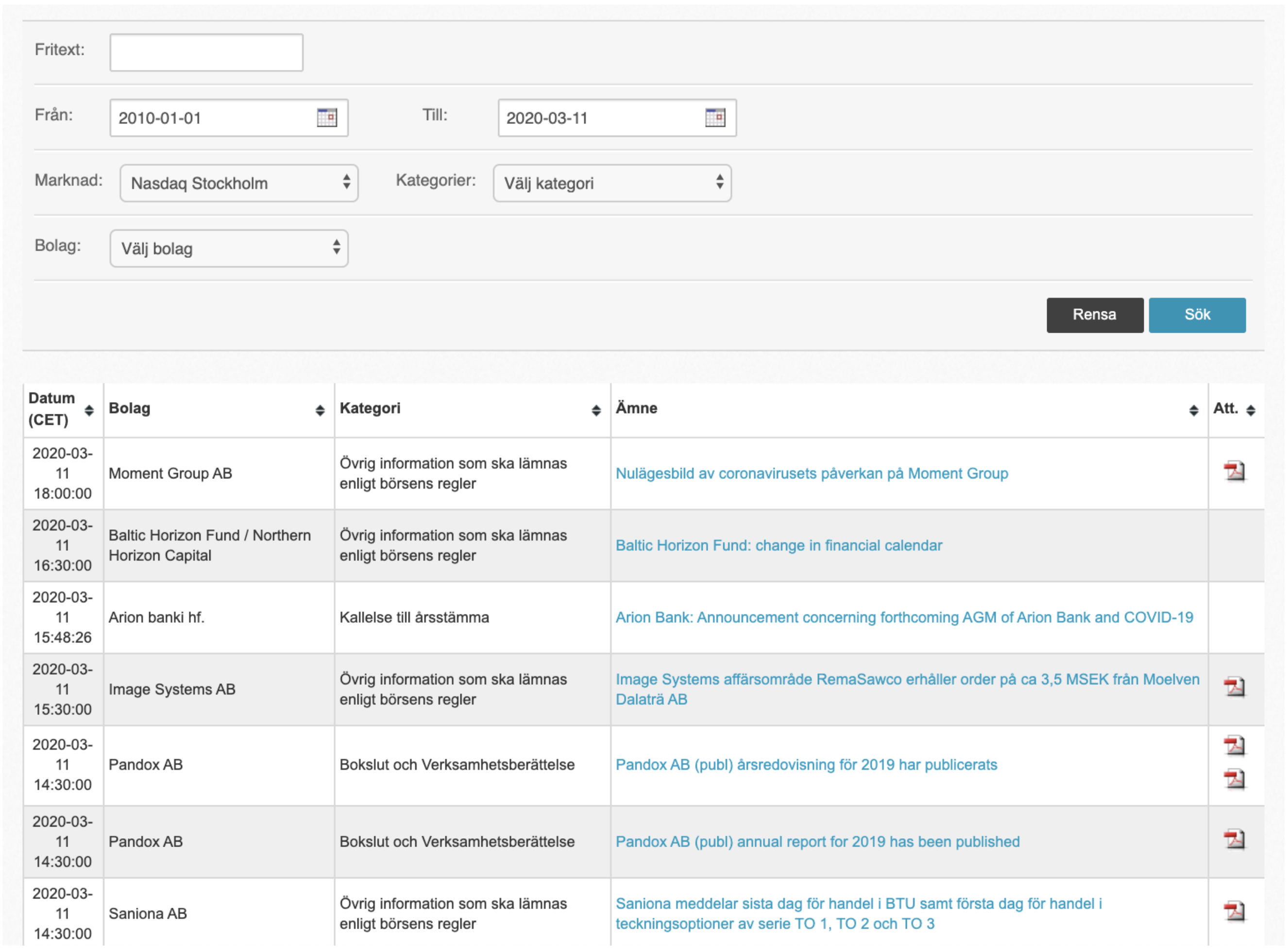
	doc_id	language	date_time	headline	text
0	926594	sv	2020-03-11 17:25:00	Huvudägaren avyttrar aktier på grund av överbe...	Huvudägaren Robert Wu har genom närstående bol...
1	926592	sv	2020-03-11 17:06:50	XMReality AB (publ) offentliggör utfall av gen...	EJ AVSEDD FÖR DISTRIBUTION ELLER PUBLICERING, ...
2	926580	sv	2020-03-11 16:20:00	S2Medical AB (publ) erhåller stororder värd ca...	S2Medical AB (publ) har erhållit en stororder ...
3	926569	sv	2020-03-11 15:30:00	Image Systems affärsområde RemaSawco erhåller ...	Ordern omfattar leverans och installation av e...
4	926565	sv	2020-03-11 15:20:00	Konecranes Abp: Meddelande i enlighet med 9:e ...	KONECRANES ABP BÖRSMEDDELANDE 11.3.2020 kl. ...

Figure 31 - **File containing company names and their corresponding RIC number**



	company	industry	RIC	TIC	ISIN
0	A3 Allmänna IT- och Telekom.	Telecommunications	ATREA.ST	ATRE	SE0001625534
1	AAK	Consumer Goods	AAK.ST	AAK	SE0011337708
2	ABB Ltd	Industrials	ABB.ST	ABB	CH0012221716
3	AcadeMedia	Consumer Services	ACADE.ST	ACAD	SE0007897079
4	Actic Group	Consumer Services	ATIC.ST	ATIC	SE0009269467

Figure 32 - **Screenshot of scraped metadata.**

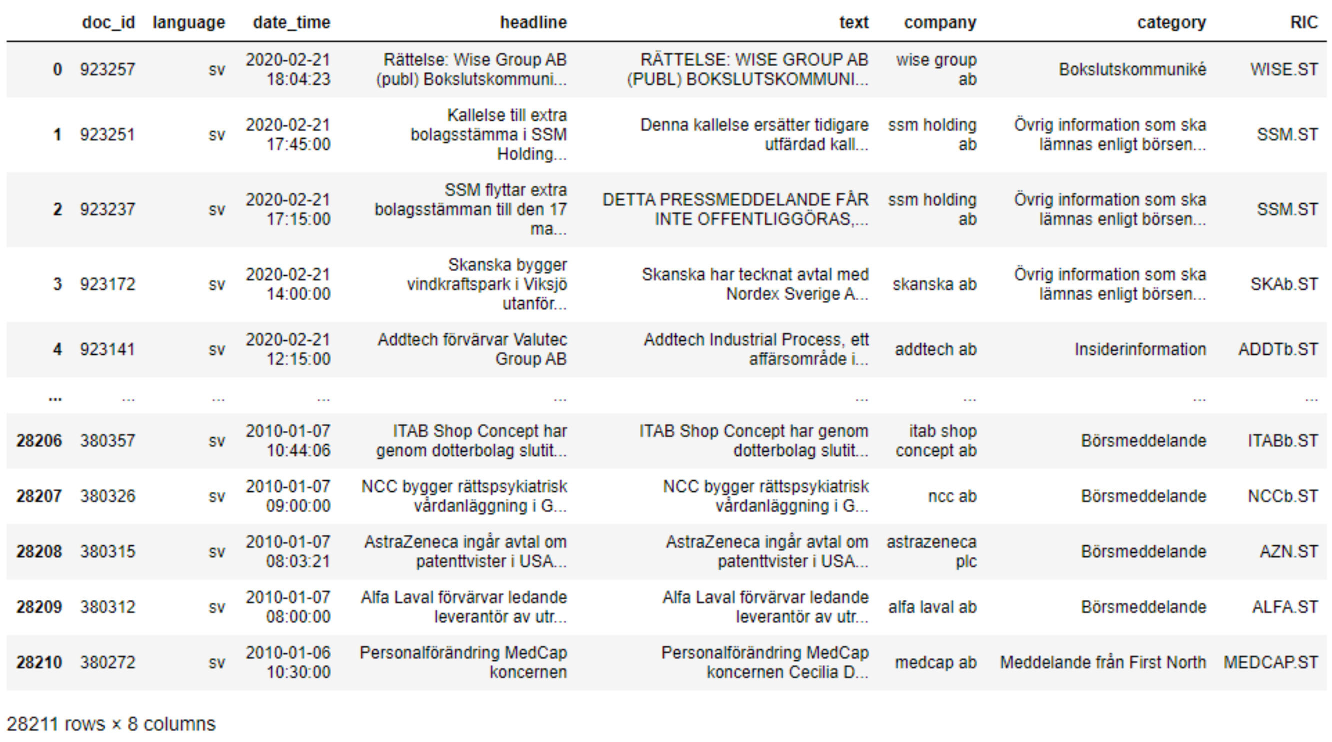


Fritext:			
Från:	2010-01-01	Till:	2020-03-11
Marknad:	Nasdaq Stockholm	Kategorier:	Välj kategori
Bolag:	Välj bolag		

Datum (CET)	Bolag	Kategori	Ämne	Att.
2020-03-11 18:00:00	Moment Group AB	Övrig information som ska lämnas enligt börsens regler	Nulägesbild av coronavirusets påverkan på Moment Group	📄
2020-03-11 16:30:00	Baltic Horizon Fund / Northern Horizon Capital	Övrig information som ska lämnas enligt börsens regler	Baltic Horizon Fund: change in financial calendar	
2020-03-11 15:48:26	Arion banki hf.	Kallelse till årsstämma	Arion Bank: Announcement concerning forthcoming AGM of Arion Bank and COVID-19	
2020-03-11 15:30:00	Image Systems AB	Övrig information som ska lämnas enligt börsens regler	Image Systems affärsområde RemaSawco erhåller order på ca 3,5 MSEK från Moelven Dalaträ AB	📄
2020-03-11 14:30:00	Pandox AB	Bokslut och Verksamhetsberättelse	Pandox AB (publ) årsredovisning för 2019 har publicerats	📄 📄
2020-03-11 14:30:00	Pandox AB	Bokslut och Verksamhetsberättelse	Pandox AB (publ) annual report for 2019 has been published	📄
2020-03-11 14:30:00	Saniona AB	Övrig information som ska lämnas enligt börsens regler	Saniona meddelar sista dag för handel i BTU samt första dag för handel i teckningsoptioner av serie TO 1, TO 2 och TO 3	📄

Source: http://www.nasdaqomxnordic.com/-nyheter/foretagsmeddelanden. *The picture depicts the columns Datum (timestamp), Bolag (company), Kategori (category) and Ämne (headline).*

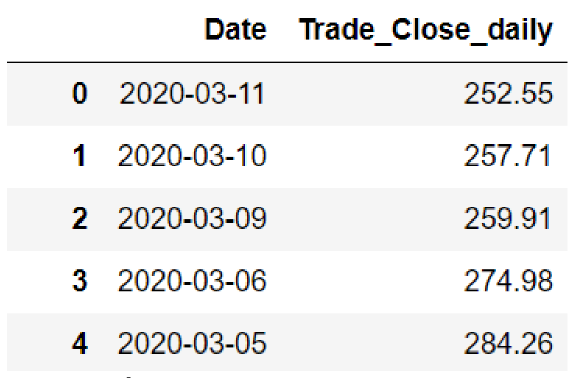
Figure 33 - **Screenshot of the dataset where each document has been matched with its company name, and subsequently with its RIC number.**



	doc_id	language	date_time	headline	text	company	category	RIC
0	923257	sv	2020-02-21 18:04:23	Rättelse: Wise Group AB (publ) Bokslutskommuni...	RÄTTELSE: WISE GROUP AB (PUBL) BOKSLUTSKOMMUNI...	wise group ab	Bokslutskommuniké	WISE.ST
1	923251	sv	2020-02-21 17:45:00	Kallelse till extra bolagsstämma i SSM Holding...	Denna kallelse ersätter tidigare utfärdad kall...	ssm holding ab	Övrig information som ska lämnas enligt börsen...	SSM.ST
2	923237	sv	2020-02-21 17:15:00	SSM flyttar extra bolagsstämman till den 17 ma...	DETTA PRESSMEDDELANDE FÅR INTE OFFENTLIGGÖRAS,...	ssm holding ab	Övrig information som ska lämnas enligt börsen...	SSM.ST
3	923172	sv	2020-02-21 14:00:00	Skanska bygger vindkraftspark i Viksjö utanför...	Skanska har tecknat avtal med Nordex Sverige A...	skanska ab	Övrig information som ska lämnas enligt börsen...	SKAb.ST
4	923141	sv	2020-02-21 12:15:00	Addtech förvärvar Valutec Group AB	Addtech Industrial Process, ett affärsområde i...	addtech ab	Insiderinformation	ADDTb.ST
...	...	...	...	...	...	...	...	...
28206	380357	sv	2010-01-07 10:44:06	ITAB Shop Concept har genom dotterbolag slutit...	ITAB Shop Concept har genom dotterbolag slutit...	itab shop concept ab	Börsmeddelande	ITABb.ST
28207	380326	sv	2010-01-07 09:00:00	NCC bygger rättspsykiatrisk vårdanläggning i G...	NCC bygger rättspsykiatrisk vårdanläggning i G...	ncc ab	Börsmeddelande	NCCb.ST
28208	380315	sv	2010-01-07 08:03:21	AstraZeneca ingår avtal om patenttvister i USA...	AstraZeneca ingår avtal om patenttvister i USA...	astrazeneca plc	Börsmeddelande	AZN.ST
28209	380312	sv	2010-01-07 08:00:00	Alfa Laval förvärvar ledande leverantör av utr...	Alfa Laval förvärvar ledande leverantör av utr...	alfa laval ab	Börsmeddelande	ALFA.ST
28210	380272	sv	2010-01-06 10:30:00	Personalförändring MedCap koncernen	Personalförändring MedCap koncernen Cecilia D...	medcap ab	Meddelande från First North	MEDCAP.ST

28211 rows × 8 columns

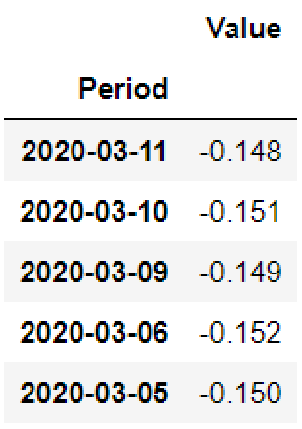
Figure 34 - **Five example rows of OMXSGI**



	Date	Trade_Close_daily
0	2020-03-11	252.55
1	2020-03-10	257.71
2	2020-03-09	259.91
3	2020-03-06	274.98
4	2020-03-05	284.26

Source: Yahoo Finance

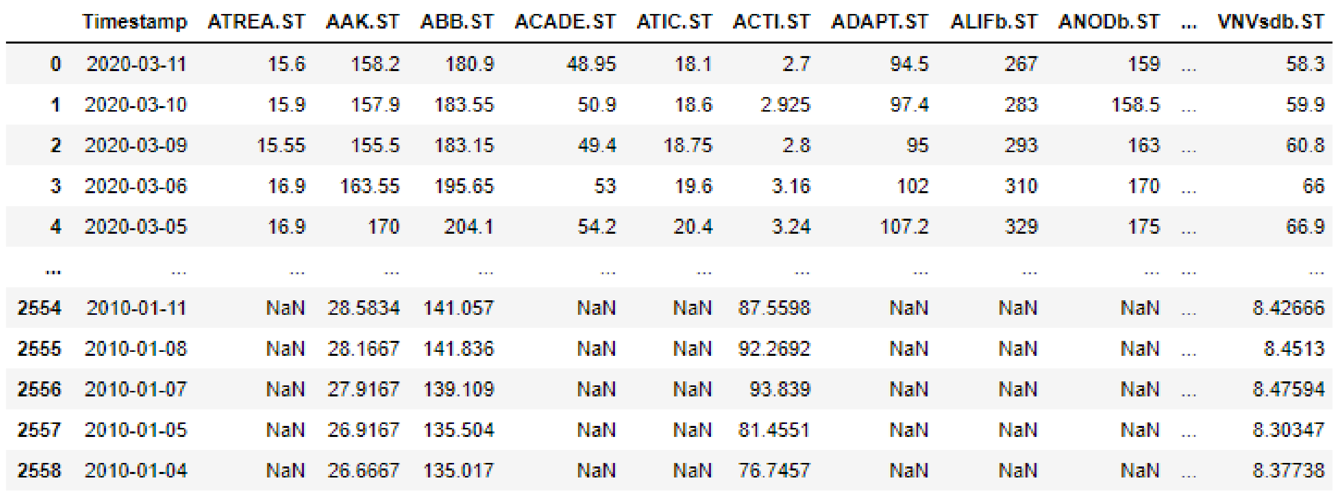
Figure 35 - **Five example rows of Swedish three month Treasury bills. The value is expressed as the annual rate of return in percent.**



	Value
Period	
2020-03-11	-0.148
2020-03-10	-0.151
2020-03-09	-0.149
2020-03-06	-0.152
2020-03-05	-0.150

Source: Riksbanken (Swerdish Central Bank)

Figure 36 - **Sample of rows from the file containing all stocks listed on OMX Stockholm per 2020-03-11, during the period 2010-01-01 to 2020-03-11. Each stock is a time series, represented by its RIC code.**



	Timestamp	ATREA.ST	AAK.ST	ABB.ST	ACADE.ST	ATIC.ST	ACTI.ST	ADAPT.ST	ALIFb.ST	ANODb.ST	...	VNVsdb.ST
0	2020-03-11	15.6	158.2	180.9	48.95	18.1	2.7	94.5	267	159	...	58.3
1	2020-03-10	15.9	157.9	183.55	50.9	18.6	2.925	97.4	283	158.5	...	59.9
2	2020-03-09	15.55	155.5	183.15	49.4	18.75	2.8	95	293	163	...	60.8
3	2020-03-06	16.9	163.55	195.65	53	19.6	3.16	102	310	170	...	66
4	2020-03-05	16.9	170	204.1	54.2	20.4	3.24	107.2	329	175	...	66.9
...	...	...	...	...	...	...	...	...	...	...	...	...
2554	2010-01-11	NaN	28.5834	141.057	NaN	NaN	87.5598	NaN	NaN	NaN	...	8.42666
2555	2010-01-08	NaN	28.1667	141.836	NaN	NaN	92.2692	NaN	NaN	NaN	...	8.4513
2556	2010-01-07	NaN	27.9167	139.109	NaN	NaN	93.839	NaN	NaN	NaN	...	8.47594
2557	2010-01-05	NaN	26.9167	135.504	NaN	NaN	81.4551	NaN	NaN	NaN	...	8.30347
2558	2010-01-04	NaN	26.6667	135.017	NaN	NaN	76.7457	NaN	NaN	NaN	...	8.37738

Source: Refinitiv Eikon

Appendix 3 - Labeling results

Tables depicting complete results from the labeling procedures. The Y-axis depicts the results from using different thresholds to determine what constitutes a Stable price movement, and what constitutes Up/Down. All results are acquired from running Logistic Regression (default settings) using TF-IDF (default settings).

Table 25 - **Returns**

Threshold	Up	Down	Stable	Accuracy	Leverage
0,5%	39,265%	37,323%	23,412%	41,938%	2,673
1,0%	30,631%	28,656%	40,714%	45,250%	4,536
1,5%	23,794%	21,908%	54,299%	56,110%	1,812
2,0%	18,499%	16,994%	64,507%	65,141%	0,633
2,5%	14,652%	13,310%	72,038%	72,280%	0,242
3,0%	12,048%	10,474%	77,478%	77,552%	0,075
3,5%	9,994%	8,504%	81,502%	81,515%	0,014
4,0%	8,299%	6,977%	84,724%	84,724%	0,000
4,5%	6,949%	5,682%	87,370%	87,365%	-0,005

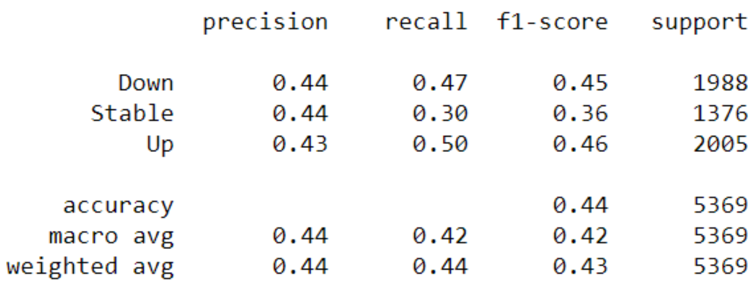
Table 26 - **Market Adjusted Returns**

Threshold	Up	Down	Stable	Accuracy	Leverage
0,5%	37,807%	38,324%	23,868%	42,898%	4,573
1,0%	28,623%	28,721%	42,656%	47,029%	4,373
1,5%	21,819%	21,475%	56,706%	58,136%	1,430
2,0%	17,115%	16,733%	66,151%	66,603%	0,452
2,5%	13,758%	13,147%	73,095%	73,174%	0,079
3,0%	11,354%	10,521%	78,125%	78,274%	0,149
3,5%	9,482%	8,555%	81,963%	81,888%	-0,075
4,0%	7,852%	7,140%	85,008%	84,976%	-0,033
4,5%	6,665%	5,947%	87,388%	87,374%	-0,014

Table 27 - **Jensen's Alpha**

Threshold	Up	Down	Stable	Accuracy	Leverage
0,5%	37,337%	37,025%	25,638%	42,404%	5,067
1,0%	28,414%	27,082%	44,505%	49,073%	4,569
1,5%	21,447%	20,222%	58,332%	59,999%	1,667
2,0%	16,910%	15,620%	67,469%	68,023%	0,554
2,5%	13,543%	12,318%	74,138%	74,381%	0,242
3,0%	11,075%	9,920%	79,005%	78,973%	-0,033
3,5%	9,273%	8,094%	82,633%	82,517%	-0,116
4,0%	7,847%	6,711%	85,442%	85,465%	0,023%
4,5%	6,623%	5,570%	87,807%	87,831%	0,023%

Appendix 4 - Classification Report, H1 Evaluation



```
              precision    recall  f1-score   support

        Down       0.44      0.47      0.45      1988
      Stable       0.44      0.30      0.36      1376
          Up       0.43      0.50      0.46      2005

    accuracy                           0.44      5369
   macro avg       0.44      0.42      0.42      5369
weighted avg       0.44      0.44      0.43      5369
```

Appendix 5 - Code used to initiate BERT on Linux servers from Google Gloud Platform

```
export STORAGE_BUCKET=gs://"BUCKET-NAME"

export BERT_BASE_DIR="PATH TO PRE-TRAINED MODEL"

export TASK_NAME="UNIQUE IDENTIFIER"



python3 ./run_classifier.py \

--task_name=cola \

--do_train=true \

--do_eval=true \

--do_predict=true \

--data_dir=${STORAGE_BUCKET}/data1 \

--vocab_file=$BERT_BASE_DIR/vocab.txt \

--bert_config_file=$BERT_BASE_DIR/config.json \

--init_checkpoint=$BERT_BASE_DIR/bert_swedish_model.ckpt \

--max_seq_length=300 \

--train_batch_size=32 \

--learning_rate=5e-5 \

--num_train_epochs=3.0 \

--output_dir=${STORAGE_BUCKET}/${TASK_NAME}-output/ \

--do_lower_case=false \

--keep_accents=true \
```