Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
NTNU Business School

Medby, Karl Olav
Nordgård, Aleksander

Machine learning for corporate announcement analysis and stock price prediction

Maskinlæring for analyse av børsmeldinger og aksjekursprediksjon

Master's thesis in Business Administration
Supervisor: Becker, Denis and Næss, Arild Brandrud
May 2019

NTNU
Kunnskap for en bedre verden

Medby, Karl Olav
Nordgård, Aleksander

Machine learning for corporate announcement analysis and stock price prediction

Maskinlæring for analyse av børsmeldinger og aksjekursprediksjon

Master's thesis in Business Administration
Supervisor: Becker, Denis and Næss, Arild Brandrud
May 2019

Norwegian University of Science and Technology
Faculty of Economics and Management
NTNU Business School

NTNU
Norwegian University of
Science and Technology

Sammendrag

I senere år har maskinlæring og tekstanalyse vist store fremskritt innenfor finansielle bruksområder. I denne oppgaven lager vi modeller ved hjelp av maskinlæring og språkteknologi for å komme med estimat på en aksjekursendring som følge av publikasjon av børsmeldinger på Oslo Børs, og bruker funnene til å argumentere mot den sterkeste formen for markedseffisiens. Vi sammenligner ni forskjellige modeller, før vi benytter de mest lovende i en long/short tradingstrategi med hedging mot Oslo Børs hovedindeks. Resultatene tyder på at det er mulig å oppnå meravkastning over indeksen, noe som viser at børsmeldinger bør være inkludert som beslutningsgrunnlag i en automatisert tradingstrategi.

Vår studie viste at beste resultatene ble oppnådd ved å representere tekstkorpuset som en TF-IDF-matrise, og deretter redusere dimensjonaliteten ved hjelp av latent semantisk analyse. En «naiv Bayes» klassifiseringsmodell ga best resultater ved kryssvalidering på treningsdataene, mens «gradient boosting» presterte best på testdataene.

Abstract

Machine learning and natural language processing have in recent years shown great promise in several financial applications. In this paper we create models using machine learning and natural language processing to estimate stock price changes related to the publication of corporate announcements on the Oslo Stock Exchange, and use the findings from our model to argue against the strongest form of market efficiency. We compare nine different models before the most promising are applied in a trading application with a long/short strategy hedged against the Oslo Stock Exchange benchmark index, which indicates that there is potential to achieve excess returns over the index, showing that corporate announcements should be included in an automated trading application.

Our study found that the best results came by representing the corpus in a TF-IDF matrix and reducing the dimensionality with latent semantic analysis before training the classifiers. A naive Bayes classifier gave the best cross-validation score on the training set, while a gradient boosting classifier performed best on the test set.

Table of contents

1 Introduction

The ability to predict future stock prices is the prime aspiration of anyone trying to make a profit in the stock market. In the quest to beat the market, investors are always searching for indicators that could help them tip the scale in their favor. Public sentiment and news articles have previously been shown to hold some predictive power towards stock prices (Gidófalvi, 2001; Seng and Yang, 2017), suggesting that text data can be used as sources of information to improve financial forecasts, which could provide investors with a competitive advantage in the market. Stock prices sometimes show considerable movement after corporate announcements are released, making the announcements interesting potential additions to a market prediction model.

Today, information is quickly spread through online sources such as news outlets, social media and a myriad of blogs, producing an overwhelming amount of data. Automatic information processing tools which can handle human written text, i.e., natural language, have therefore experienced a growing popularity in recent years and the application of these within finance has become a promising field of study. Since the biggest advantage in stock prediction goes to whomever can act fastest when new relevant information becomes available, computers' ability to perform quantitative analysis much faster than humans, along with recent years' improvements in natural language processing have made them popular for fast analysis of text data as well.

In this paper we use machine learning and natural language processing on a data set consisting of 107 232 corporate announcements published on the Oslo Stock Exchange, to analyze whether these corporate announcements contain information that makes machine learning models able to predict how the stock price might change. We show that we are able to estimate stock market movement based on this publicly available information, thus weakening the hypothesis of market efficiency, which states that it should be impossible to consistently achieve excess return over the market. We demonstrate a profitable, yet simple trading strategy based on our machine learning models which support our argument against the efficient market hypothesis. These findings are produced using logistic regression, a naive Bayes classifier, different ensembles of decision trees, and LSTM networks. We also use different ways of

vectorizing the text, i.e., word counts, term-frequency inverse-document-frequency with latent semantic analysis, pre-trained global vectors, and contextualized string embeddings.

Examination of our strongest performing model shows that the relationships it learned between financial concepts mentioned in corporate announcements and stock valuation are consistent with financial literature.

Our main research question is: *Can machine learning and natural language processing be used on corporate announcements published on the Oslo Stock Exchange to estimate stock price movement?* In addition, we attempt to answer the following related questions: *Should corporate announcements be used in an automated short-term trading strategy?* and*, is it possible to claim market efficiency on the Oslo Stock Exchange?*

The structure of this paper is as follows. First, we highlight some relevant previous research concerning market efficiency, sentiment analysis and stock price prediction. Then we present theoretical foundations in finance and machine learning, before describing the methodology used. Next, the results are presented, followed by a discussion. Lastly, we summarize the findings in the conclusion.

2 Previous work

In previous research, machine learning methods, such as natural language processing (NLP), have shown promising results in different financial applications, e.g., prediction of the stock market (Seng and Yang, 2017), prediction of bankruptcy (Næss *et al.*, 2017) and forecasting bitcoin price movement (Karalevicius, 2018). The results of previous work make NLP and machine learning in finance an interesting field of study.

This section presents previous research that is relevant to NLP and how it has been used to forecast changes in the stock market and applied to trading. In addition, some machine learning models are presented along with examples of how they have performed in related tasks to ours.

Gidófalvi (2001) used financial news articles in his prediction of short-term stock price movements. Similar to some models presented later in this paper, Gidófalvi trained a naive Bayes model to classify each news article as corresponding to either positive, negative or neutral changes in the associated stock price. The findings were statistically significant and

showed that financial news could indicate stock price changes on a time scale of about 40 minutes.

Public sentiment on Twitter has been shown to indicate how stock prices are going to change, and splitting the sentiment into several dimensions further improves the accuracy of prediction (Bollen, Mao and Zeng, 2011). These indicators were based on behavioral economics which states that both internal emotions and external factors play significant roles in human decision-making (Dolan, 2002). Bollen, Mao and Zeng showed that the hypothesis of not being able to predict correct more than 50% of the time is weakened. Their research resulted in a model with 86.7% accuracy in predicting the direction of the stock market's daily developments.

Experiments on the relationship between stock price movement and choice of words and tone used by authors in financial news articles showed that these variables could predict the direction of the price movement with an accuracy of 59% (Schumaker *et al.*, 2012). They were able to successfully predict the direction of price movement with a relatively high accuracy as well as use the model in a simple trading strategy to receive 3.3% return on investment.

Previous research on the effects of news and public sentiment on stock price movement showed that financial news articles could provide a competitive advantage if considered as a for decision making factor when trading, and that public sentiment is related to emotional fluctuations in investors which affect their decision making (Li *et al.*, 2014). They used weighted term vectors in their predictive model to analyze the effect news had on stock movements and used the sentiment these news articles evoked in the public to analyze whether sentiment impact stocks or not. Furthermore, the research found that the media's impact on the firm's stock prices varied based on firm characteristics and article content.

More research on sentiment in documents have strengthened the standing of sentiment analysis for making investment decisions (Seng and Yang, 2017). They used a bag-of-words method, i.e., counting word occurrence in a document, in combination with a financial lexicon to determine if documents were positively or negatively correlated with stock price movement. The degree of positive and negative sentiment was further and used as coefficients in a regression model which was used to predict price movements for the next period.

Deep learning, a cutting-edge field of machine learning, has been shown to improve sentiment analysis on news for stock prediction (Day and Lee, 2016). The choice of news provider significantly impacted performance when predicting stock prices, ranging from 1.5% return up to 22.4% return.

Using a type of neural network called LSTM, resulted in a model with an average accuracy of 70.4% in predicting the stock price movement on a specific company (Liu, 2018).

Following these previous research results, we implement similar approaches to our problem, as described in section 5.

3 Economic theory

Previous work regarding stock market prediction (Gidófalvi, 2001; Bollen, Mao and Zeng, 2011; Seng and Yang, 2017) often mention and argue against the efficient market hypothesis (EMH). In this paper we show strong indications that corporate announcements can be used to predict stock prices, which should not be possible according to the strongest form of EMH. Therefore, a discussion of the hypothesis is in order.

In this section Fama's hypothesis of market efficiency is introduced along with an explanation of the different degrees of market efficiency. Furthermore, some known criticism of the hypothesis and previous findings is addressed, and the section ends with relevant corporate finance theory and what signal effects corporate announcement might have.

3.1 Market efficiency

Market efficiency is the hypothesis of how stock prices reflect all relevant and available information in the market, meaning that corporate announcements should not contain information traders could use in their prediction of stock price movement. If such anomalies existed, they would be arbitraged away immediately in an efficient market (Fama, 1970).

There are three different degrees of market efficiency: weak, semi-strong and strong forms of efficiency. The weak form assumes that historical price movements do not affect future price movements, meaning momentum and methods like technical analysis should not be able to predict future development. Semi-strong form of market efficiency suggests that new public information in the market is quickly absorbed and reflected in stock prices, resulting in no room for traders to exploit new information for market prediction. The strong form of market

efficiency suggests that all public and private information is reflected in the stock prices. The strong form of market efficiency would imply that even an insider of the firm would not have any opportunity to use inside information to predict future price movement as this information should already be calculated in the stock price.

Random walk and *fair game* are terms often used to describe the different degrees of market efficiency. Random walk is defined as the behavior of stock prices where they seem to develop randomly and independent from historical prices and momentum. Fair game is a term based on the assumptions that prices in the stock market at all time reflects expectations and all available information in the market (Fama, 1970).

The theory of market efficiency seems ubiquitous in economic literature, yet some aspects of the theory remain controversial. One point of criticism is the existence of anomalies that arise from delay in information processing that traders can use to predict future price (Jensen, 1978; Schwert, 2003). The counterargument is that overreaction and underreaction from these anomalies equalize each other, which leaves no opportunities for traders to use the delay and anomalies to beat the market in the long run (Fama, 1997).

Tuyon and Ahmad (2016) did a literature review showing that the Malaysian stock market reflected the weak-form of efficiency market hypothesis while still acknowledging that the market sometimes had temporary inefficiencies. EMH was mostly invalidated, but several researchers agreed that the anomalies were impossible to exploit for trading purposes because of transaction costs, and therefore providing evidence that EMH in fact existed (Ţiţan, 2015). Additionally, Ţiţan mentioned several authors who found evidence that made falsifying random walk impossible, further strenghting the weak form of EMH.

3.2 Corporate finance and information content in corporate announcements

Investors and stockholders often use corporate announcements as signals on how the manager of the firm feel about the current financial situation. Announced stock repurchases have shown an average abnormal price rise of 2% after the announcements were made, a change that might happen because an announced repurchase potentially signals that managers are optimistic and confident about the company's future, or feel that the shares are underpriced. When managers announced repurchases of stocks, major shareholders were often found to hold on to their shares (Brealey, Myers and Allen, 2017).

Corporations which find themselves in situations where they need to raise capital for new investments have to make a decision with regards to capital structure. They often choose between issuing additional shares, borrowing from banks, or issuing new bonds (Brealey, Myers and Allen, 2017). Issuing new bonds does not affect ownership of the firm and will not affect how much earnings per share (EPS) investors get similarly to how issuing new stock would.

In an ideal world, according to the Modigliani and Miller theory, capital structure of firms should not affect their valuation (Modigliani and Miller, 1959). The way the market reacts to debt issuance does not seem to follow this theory, as announcements of bond issuance have shown different results in previous work; Mikkelson and Partch (1986) found that convertible bonds resulted in a significant decline in stock price, while straight bonds made no significant impact on the stock price. However, (Howton, Howton and Perfect, 1998) also found straight bonds to have a significant negative effect on the stock price.

4 Machine learning prerequisites

This section contains a brief introduction to relevant concepts in machine learning and natural language processing. First, a general introduction to the field of machine learning is presented, followed by different ways of representing text as numbers, and the classification models we use in this paper.

4.1 Machine learning and natural language processing

Machine learning has been defined as a "field of study where you give computers the ability to learn without being explicitly programmed" (Samuel, 1959). The field can be split into different approaches to learning: supervised learning, unsupervised learning, and semi-supervised learning. Supervised learning is when a model is trained on a data set containing both inputs and output labels, meaning that the model tries to predict known target values. An example of supervised learning is a classification task such as we consider in this paper, because the models learn to predict stock returns from text. Unsupervised learning is when the model is trained on a data set with only inputs and no output labels, meaning the model learns to recognize patterns in the data, e.g., clustering companies into groups based on accounting numbers. Semi-

supervised learning is a combination of supervised and unsupervised learning, where you have a data set consisting of both labeled and unlabeled data.

Natural language processing (NLP) is a subfield of machine learning which focuses on making algorithms that understand language and analyze it. The usage of NLP has grown in conjunction with an increased focus on big data and text analysis (Maynar and Bontcheva, 2014).

4.2 Word embeddings and vectorizing methods

In order to perform statistical computations on text, it has to be represented as numbers somehow. This can be done by either simple word-occurrence approaches like bag-of-words, or more by representing words using more sophisticated word embeddings, described below. An example of vectorizing is representing the word "hat" by a vector with 26 dimensions, where each element represents a letter in the English alphabet. The dimensions take the values 1 or 0 representing the presence or absence of the corresponding letter in the word. For the word "hat", each element would be 0, except the 1st, 8th and 20th, which would be 1 because they correspond to "a", "h", and "t" respectively (Zhang, Wang and Liu, 2018). Similarly, sentences or documents could be represented as vectors, where each element representing the presence or absence of the words in the corpus, this is known as bag-of-words, because you only count which words are present, but not in which order, as if each sentence or document was just a bag full of words (Jurafsky and Martin, 2014).

In this section each of the vectorizing methods and the word embedding methods used in this paper are introduced and the rationale behind each model is explained.

4.2.1 Term-frequency inverse-document-frequency

According to Aizawa (2003), one of the most used term weighting schemes in information retrieval systems of today is term-frequency inverse-document-frequency (TF-IDF). It is a word vectorizing method which weights vectors created with the bag-of-words method (Jurafsky and Martin, 2014). Instead of simply counting the number of times a word occurs in the document with no regards to context, TF-IDF also weights the words depending on how often they appear in the document relative to how often it appears in the complete corpus (Ramos, 2003). In other

words, if a word appears often in a document, but rarely in the corpus, it is considered important for the contents of the particular document. TF-IDF is calculated as follows (Ramos, 2003):

$$w_d = f_{w,d} \cdot \log\left(\frac{|D|}{f_{w,D}}\right)$$

w_d represents the word's weight, D is the collection of announcements, w is a word in the announcement, $d \in D$ represent an individual announcement, $f_{w,d}$ is the number of times w appears in d, $|D|$ is the size of the corpus, in terms of announcements and $f_{w,D}$ is the number of announcements in D which the word appears.

4.2.2 Latent semantic analysis

Latent semantic analysis (LSA) is a dimensionality reduction technique often used for text analysis (Landauer, Foltz and Laham, 1998). Dimensionality reduction means to represent a vector in fewer dimensions, with minimal loss of information. It is useful for identifying underlying concepts in a large amount of text, especially when the corpus contains a lot of synonyms and polysemy, as it represents the text by the concepts it contains and not by the individual words in it. The length of each document-vector is thus reduced from the number of words in the vocabulary to the number of concepts identified by LSA, which can be several orders of magnitude smaller.

LSA is done by performing global matrix factorization applied to a term-document frequency matrix (Pennington, Socher and Manning, 2014). The term-document frequency matrix M is of size $|D| \times n$ where $|D|$ is the number of corporate announcements in the corpus, and n is the number terms in the vocabulary, including bigrams. M is decomposed into three matrices: a document-to-concept matrix, U, of size $|D| \times r$, a diagonal concept-strength matrix, Σ, of size $r \times r$, and a term-to-concept matrix, V of size $n \times r$, where r is the number of underlying concepts identified by the LSA.

$$M = U\Sigma V^T$$

A low-rank approximation of M can be obtained by:

$$M \approx M_k = U\Sigma_k V^T$$

Where $k < r$ and k is the number of concepts used to represent the text. Σ_k is determined by replacing the $r - k$ smallest singular values in Σ with zeroes (Manning, Raghavan and Schütze, 2008).

In other words, LSA uses no humanly constructed dictionaries, knowledge bases, semantic networks, or similar, in order to extract meaning from the text (Laham, 1998). This results in concepts which includes words that are close in meaning and used in the same context.

4.2.3 Global vectors for word representation

Combining two models, global matrix factorization and local context window methods, results in the Global vectors for word representation (GloVe). GloVe is a global log-bilinear regression model that works by using statistical information gained by training the model on a nonzero element in a word-word co-occurrence matrix. The rationale behind GloVe vectors is to measure co-occurrence probabilities for the purpose of deciding which words are relevant to each other from which words are irrelevant (Pennington, Socher and Manning, 2014).

GloVe calculates these probabilities based on how often words occur together and which words are not likely to occur together. As an example, one could compare probability of the words ice and steam occurring with the words solid, gas, and water. GloVe would first calculate $P(\text{solid}|\text{ice})$ and $P(\text{solid}|\text{steam})$ and thereafter the ratio $\frac{P(\text{solid}|\text{ice})}{P(\text{solid}|\text{steam})}$. This will be done for each possible outcome, in order to determine the relative co-occurrence to other words in the corpus.

GloVe uses a specific weighted least squares model which is pre-trained on a large corpus which include Wikipedia 2014, twitter and data collected by crawling the internet, which results in vectors that have produced a model with 75% accuracy (Pennington, Socher and Manning, 2014).

4.2.4 Contextualized string embeddings

The concept of contextualized string embeddings, also known as FLAIR-embeddings, is based on the internal states of a character level language model. The language model is comprised of two recurrent neural networks, described in section 4.3.4. One was trained to predict the next character based on the beginning of a string, and the other to predict previous character based on the end of a string. This way the network learns features about the characters in a sentence based on their context, hence the name. This means that the same word will have a different embedding based on its context, and the model is thus able to differentiate between e.g., homonyms, which are words that have the same spelling, but different meanings based on the

context; the word “Washington” will have different meanings in the sentences “George Washington was” and “Washington is a place” (Akbik, Alan, Duncan Blythe, Roland, 2018).

These embeddings have been shown to produce great results in a wide range of NLP tasks, such as named entry recognition, part of speech tagging, and text classification.

4.3 Classifiers

Text classification means to place different samples of text into categories based on their content. In a financial context this could mean to determine if the stock price will go up, down or remain unchanged as a result of the publication of a corporate announcement. In NLP, classification models are trained with text as input and one or more class labels as output. The model then learns what properties of the input are associated with the different classes (Drummond, 2018).

This section explains the rationale behind each classifier used in this paper and present some of the mathematics used in the different calculations.

4.3.1 Logistic regression

Logistic regression is a form of standard linear regression which can be applied to classification problems (Jukes, 2018). The model estimates the log-odds, z, for each class by way of linear regression:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n,$$

where $x_1$to x_n represents the values of the concepts from the transformed term-frequency matrix, and $\beta_0$to β_n represents the learned weights for each attribute (Jukes, 2018). In order to get a conditional probability estimate for each class given the occurrence of concepts, x_1 to x_n in the sample, with a value between 0 and 1, the log-odds are then converted to:

$$P(c_0|x_1 \dots x_n) = \frac{1}{1 + e^{-z}}$$

where c_0 represents class 0.

4.3.2 Naive Bayes

Naive Bayes is a simple classification model that utilizes Bayes' rule along with an naive assumption that input attributes are independent given the class (Webb, 2018). The assumption that the input attributes are independent might seem difficult to justify when the input is text since words in human language must be related in order for the text to have meaning. However, it has been shown that the naive Bayes classifier can still perform quite well even if the assumption of independent attributes is broken (Zhang, 2004).

Naive Bayes classifiers take advantage of Bayes' theorem:

$$P(y|\boldsymbol{x}) = \frac{P(y)P(\boldsymbol{x}|y)}{P(\mathbf{x})}$$

where y is the class, and $\boldsymbol{x}$ is the input vector. The probability for each class is estimated from the conditional probability of said class given the presence or absence of each word or feature in the input vector.

The probabilities $P(\boldsymbol{x}|y)$ and $P(\boldsymbol{x})$, are given by

$$P(\mathbf{x}|y) = \prod_{i=1}^{n} P(x_i|y)$$

$$P(\mathbf{x}) = \prod_{i=1}^{k} P(c_i)P(\mathbf{x}|c_i)$$

respectively, where $\boldsymbol{x}$ is the feature vector, and x_i is the ith attribute, n is the number of attributes, c_i is the ith class, and k is the number of classes. This classifier uses a similar linear model as the logistic regression, the difference being how the parameters are chosen (Webb, 2018).

The Bernoulli variant of naive Bayes takes in binary feature vectors, simply indicating the presence or absence of each word or feature, instead of the number of times each feature appears (Mccallum and Nigam, 1998).

4.3.3 Decision tree ensemble methods

A decision tree is a classification method where one attribute of the sample is the root with branches representing the presence or absence of said attribute. From each branch follows another node which represents a different attribute which then branches off again until it reaches a leaf which represents the prediction of the model (Quinlan, 1986). The complexity of the trees

is determined by the number of nodes, i.e., the depth. A decision tree with only the root node and two branches is called a decision *stub.*

Although decision trees are simple and sometimes work well, they tend to suffer from high variance, to which two methods have been developed to counter: *bootstrap aggregation* (bagging) and *boosting* (Hastie, Tibshirani and Friedman, 2017). In the following three sections we present two boosting techniques and one bagging technique.

4.3.3.1 Adaptive boosting

AdaBoost, shortened for adaptive boosting, is a method that is trying to weight data points in a way that incentivizes sequentially trained weak learners to focus on observations that are difficult to correctly classify, and leave the observation that is already handled well (Schapire, 2013). The weighting and classifying are done sequentially as the algorithm proceed, meaning that the weighting gets adjusted at each step of the algorithm. Just like other boosting methods AdaBoost combine many relatively weak and inaccurate rules to create a highly accurate prediction rule (Schapire, 2013).

The boosting works by sequentially fitting decision stubs to the data (Zhu *et al.*, 2009). After the first stub has been trained, the samples are weighted based on the performance of the current stub – correctly classified samples get a reduced weight and misclassified samples get an increased weight. The factor of which the weights are adjusted depends on the performance of the new stub; if it is weak the samples are barely adjusted, but if it is strong, they are significantly adjusted. The next stub will then be trained on the weighted data set which means it is more likely to fit to the heavier weighted samples, after which the weights are again re-adjusted based on the performance of the last tree, and another tree is trained based on these weights. This process continues for the number of estimators specified by the user. The final model takes the weighted sum of the predictions from the weak estimators and predicts the class with the most votes.

4.3.3.2 Stochastic gradient boosting

Gradient boosting is another boosting method that, just like AdaBoost, is trying to combine many weak learners into one strong classifier. The major differences between AdaBoost and gradient boosting is that gradient boosting trains many models in a gradual, additive and

sequential manner. In other words, the two algorithms address the shortcomings of weak learners differently (Friedman, 2001).

The main idea behind gradient boosting is to minimize a loss function by sequentially adding weak learners to make better and better predictions (Mason *et al.*, 2000). What kind of loss function the model uses depends on what type of problem that is being solved.

First the algorithm makes an initial prediction, often the mean of the output values. It then calculates the error with respect to the loss function, what Friedman call "pseudo"-residuals (Friedman, 1999). Next, it trains a shallow decision tree to estimate the pseudo residuals. The model output is then the initial guess plus the estimated residual scaled down by a regularization constant called "learning rate". Thereafter the pseudo residuals from the combined model are calculated, before another shallow tree is trained to predict these. This process continues for the specified number of estimators.

Subsampling is a proposed improvement to this method and means to train each step on a randomly selected subsample of the data. This improves both performance and computational speed (Friedman, 1999).

4.3.3.3 Random forest

Random forest is a classification technique which, instead of boosting, uses bootstrap aggregation (bagging) method when combining an ensemble of decision trees in order to reduce variance and the effect of noise (Breiman, 2001). Random forest works by building multiple decision trees from random subsets of the training data, which then vote on the predicted class (Hastie, Tibshirani and Friedman, 2017). This gives more accurate predictions. This results in a classifier that is more robust to overfitting, yet still produces more accurate predictions than a single decision tree (Breiman, 2001).

4.3.4 Long short-term memory networks

Long short-term memory (LSTM) networks are a special architecture of recurrent artificial neural networks. An artificial neural network is a network of information processing nodes and non-linear activation functions organized into an input layer, one or more hidden layers, and an output layer. The output from the input layer is the input to the first hidden layer whose output

is the input to the next layer and so forth. The information is fed forward through the network until the final prediction comes out from the output layer (Zhang, Wang and Liu, 2018).

A recurrent neural network is a type of artificial neutral network that predicts the next element in the output by taking as input its own state from the previous step as well as the current input element. This means that each step is dependent on all the previous steps in the sequence. The weights in the network are estimated based on the gradient of the loss function, and one problem that arises in recurrent neural networks is the *vanishing gradient.* The problem is that because of the chain rule, the gradient (i.e., the derivative of the loss function in multi dimensions) is the product of the gradients of all previous steps in the sequence, which are often less than one, and thus the gradient for information in the sequence far away from the current step, approaches zero. The result of this is that the network essentially only focuses on a few recent elements in the sequence when predicting the next. One proposed solution to this is the LSTM network.

An LSTM network is a type recurrent neural network which in addition to the standard hidden layer has a cell state comprising of 'remember' and 'forget' gates. This means that it can learn to add and remove information from its own state, giving it the ability to keep track of long-term dependencies in the input (Zhang, Wang and Liu, 2018). LSTM approaches the problems regarding context management by splitting it in two sub-problems: how to remove information that is no longer needed and adding information that the model probably need for later decision making (Jurafsky and Martin, 2014).

5 Methodology

Before we train and compare the classification models, the data gets preprocessed, i.e., we clean up the text to reduce noise, and vectorized, meaning we convert the cleaned text into a numerical representation. The classifiers are then used in a trading application to show how these models perform compared to the benchmark index. These steps are presented in this section after a description of the data set. Lastly, we present evaluation metrics used to compare the models.

5.1 Data set

The data set used in this experiment consists of 107 232 corporate announcements published by companies listed on the Oslo stock exchange (including Merkur Market and Oslo Axess)

between January 1st, 2000 and December 31st, 2018, accompanied with the 1-day percentage change in corresponding stock prices after the news story was published. The return was calculated based on the closing price on the last trading day before publication, and the close price on the first trading day after the stock news item was published. If the message was published before or during stock market opening hours, the return was calculated from the day before to the present day, and if the message was published after opening hours, the return was calculated from the day of publishing until the next day. Only daily price history was available, so in order to keep the data consistent and comparable across messages published during different times of day, closing prices were used for all samples.

No assumptions were made regarding each company's correlation with the market. Therefore, the absolute return was used as target variable in the models. However, in the results section it is shown that the models still perform well on excess return during out-of-sample testing.

Table 1 shows descriptive statistics for the data set. For comparison the corresponding 1-day returns on the Oslo Stock exchange index, OSEBX, ("Hovedindeksen") are included.

Table 1: Data set descriptive statistics

	Return	Index return	Excess return
Count	107232	107232	107232
Mean	0.014295	0.000742	0.013552
Standard Deviation	1.432150	0.015083	1.432080
Min	-0.980272	-0.099476	-0.985156
25% percentile	0.017621	-0.006144	-0.018246
50% percentile	0.00000	0.001285	-0.000192
75% percentile	0.019417	0.008206	0.018570
Max	432.333333	0.106706	432.323418

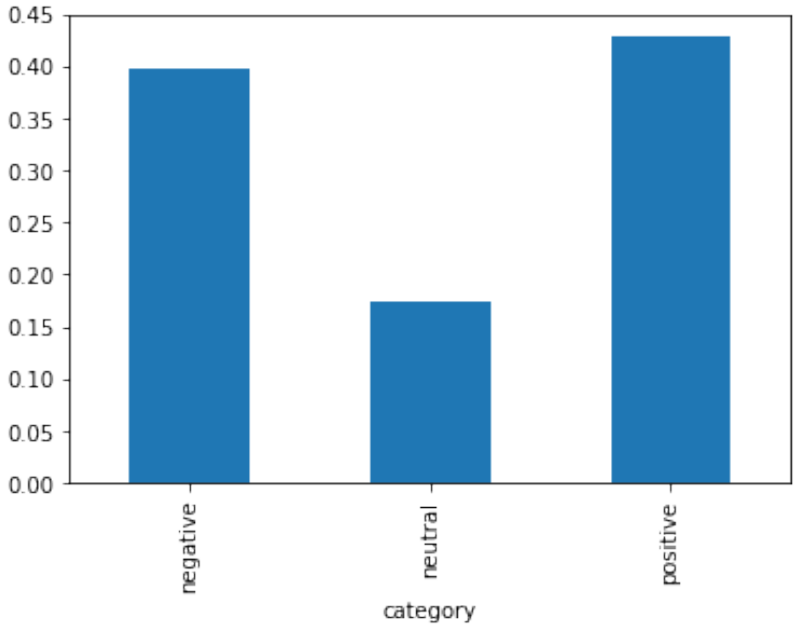


Figure 1: Category frequency distribution

The returns have a median of 0.0% and mean of +1.5%. This means that on average there is a positive expected return associated with publication of corporate news messages on the Oslo Stock exchange. The largest negative return one day after a corporate announcement is a 98% drop in share price, and the highest positive return is over 43232%.

The prices in this data set are not adjusted for splits and dividends. Hence, announcements containing both "ex dividend" and "today" (in total 138 samples) as well as those containing the phrase "stock split" and "today" (5 samples) are removed to avoid misclassifications. In case an unrelated corporate announcement has been published by a company on the same day as ex dividend or stock split the return might be misleading to the model, but this is unlikely to occur often enough to impact the results significantly.

Because of the added complexity of analyzing more than one language, only English samples were included in the data set.

The last 5000 samples are set aside as the test set to be used for out of sample testing. That includes announcements published after April 27. 2018 until December 31. 2018. The test set is chosen to be the tail of the data set because we want the models to generalize into the future, and make sure they did not merely learn temporal or other between sample dependencies in the data set.

5.2 Text pre-processing

Before vectorizing the input, the data set has to be preprocessed to reduce the risk of fitting to noise, e.g., making sure the words “Flower”, and “flower”, are identified as the same instead of treated as completely independent.

For the models to learn generalizable information instead of focusing on specific companies and other entities, company names, phone numbers, e-mail addresses and website addresses were replaced with the tokens, “company”, “phone_number”, “email_address” and “website”, respectively.

Some text processing was specific to the bag-of-words-based approaches. Namely stemming the words using the Porter stemming algorithm (Porter, 1980), meaning reducing the words to their root form by removing inflection (e.g., the words “company” and “companies” both become “company”), and the deleting of stop words, which are words that are extremely common and contribute little to none to the meaning of the sentence, such as “the”, “is”, and “are”. With the goal of further reducing the input space, numbers, punctuation and special characters were removed. Additionally, all text was converted to lower case.

5.3 Categorization

We defined the problem as a classification problem because it simplifies the task to estimating only the direction of change, instead of direction and magnitude. Another advantage of converting the output to categorical data, is that the predictions serve as easily operationalizable trading signals, as we show below with a trading application. We categorize the returns into three classes: “negative”, “neutral”, or “positive”, referring to the percentage change in stock price. A return above 0.4% is considered positive, and below –0.4% is considered negative, while the rest are considered neutral. The neutral class is included because not all corporate announcements contain information that is associated with a significant change in valuation. The choice of ±0.4% as the threshold for significant return is an empirical decision based on the distribution of returns. It is observed that 9% of the training data has returns of 0.00, and closer inspection of the distribution reveals a peak at around ±0.4%.

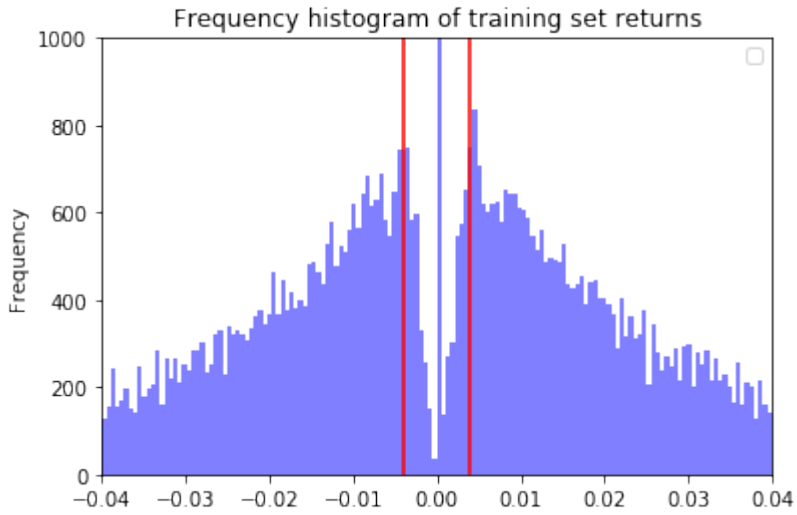


Figure 2: Frequency histogram for returns on the training set, between –4% and 4%. The red lines separate the categories for negative, neutral, and positive returns

5.4 Models compared

The following machine learning approaches were compared. For the non-deep learning-based methods (models 1–6), implementations from the Scikit-learn library for Python 3 were used (Pedregosa *et al.*, 2011). Unless otherwise specified, default parameters were used.

5.4.1 Baselines: Random classifier and greedy classifier

If EMH and the random walk hypothesis holds true, stock prices move at random and should not be predictable. Therefore, to test whether the models are able to extract information about future price movements from corporate announcements, they should at the bare minimum be able to outperform a random classifier which guesses the class by random selection.

The second baseline model is a classifier which simply predicts any sample to be the most frequent category from the training set. In this data set, the most common class is "positive" return. Thus, the greedy classifier predicts that all announcements are in the category "positive".

If the corporate announcements contain information which the classification models learn to associate with certain changes in stock price, the models must outperform these baseline classifiers.

5.4.2 Model 1: Logistic regression using Word count vectorization

Perhaps the simplest way to construct a text classification model is the basic bag-of-words method. Here, each document is represented as a vector containing the counts of each word in the corpus, which we then used as input in a logistic regression classifier. This results in a sparse input vector and considers no relationship between different words.

5.4.3 Model 2: Logistic regression using bigrams, TF-IDF vectorization and LSA

Two weaknesses with Model 1 are that every word is weighted equally, and that absolutely no context is considered. The first problem can be resolved by using TF-IDF vectorization, meaning that some information about the language is incorporated in the vectorization stage. The second problem can be remedied by including bigrams in the model, which means that the model can distinguish between for example the bigrams "not profitable" and "very profitable".

Furthermore, testing shows that reducing the vectors using latent semantic analysis with 85 components improves performance and reduces overfitting. Implementing these three refinements leads to some performance improvements over Model 1, as shown in the results section.

5.4.4 Model 3: Bernoulli naive Bayes classifier with bigrams, TF-IDF vectorization and LSA

Further improvements to the model can be made by applying a more suitable classification algorithm. The Bernoulli naive Bayes classifier takes in binary values, so each of the features from the LSA output are set to 0 if they are not relevant to the sample, and 1 if they are relevant to the sample. This subtle regularization combined with the powerful naive Bayes model leads to the model with the highest cross-validation F1-score.

5.4.5 Models 4–6: Ensemble decision trees with bigrams, TF-IDF vectorization and LSA

Three different ensemble decision tree methods were used, adaptive boosting classification with a max depth of 1 (i.e., it used decision stubs), gradient boosting classification, and random forest classification, both with a max depth of 4. All three ensembles had 64 estimators each. For gradient boosting the subsampling parameter was set to 0.5.

5.4.6 Models 7–8: LSTM neural networks with word embeddings

Preliminary experiments were conducted with two artificial neural network models. One using pre-trained GloVe embeddings available from the Flair NLP toolkit (Akbik, Alan, Duncan Blythe, Roland, 2018), and one using contextualized string embeddings trained from scratch on this corpus. The neural networks were single layer LSTM networks with a hidden size of 512 nodes.

The models were trained for 150 epochs with an initial learning rate of 0.1, which means that the optimizing algorithm performs 150 passes through the training data and updates the model parameters by the negative of the gradient multiplied by the learning rate, which gets gradually reduced, each time.

5.4.7 Model 9: Soft Vote Ensemble with bigrams, TF-IDF and LSA

A soft voting ensemble classifier was trained with a clone of the Bernoulli naive Bayes and Stochastic gradient boosting classifiers. These two were chosen because they had the highest average F1-score during cross-validation, and their trading returns on the training set were fairly uncorrelated. The soft voting ensemble classifier determines the class probabilities by averaging the predicted class probabilities from the underlying models and predicting the class with the highest average predicted probability.

5.5 Model selection

The classification models are evaluated by their micro average F1-score. Micro average means that the total numbers of true and false positives and false negatives across all categories are used when calculating the F1-score, which is the harmonic mean of precision and recall:

$$\mathrm{F1} = 2\frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Where precision and recall are defined as follows:

$$\mathrm{recall} = \frac{\mathrm{true\ positives}}{\mathrm{true\ positives} + \mathrm{false\ negatives}}$$

$$\mathrm{precision} = \frac{\mathrm{true\ positives}}{\mathrm{true\ positives} + \mathrm{false\ positives}}$$

In this context recall can be interpreted as what fraction of opportunities for significant return the model recognizes, and precision can be interpreted as the fraction of opportunities the model predicts which preforms as expected. The trade-off between these two metrics comes down to the model users' individual risk preference and use case; maximizing the F1-score represents a balanced approach.

The F1-score was measured for the different approaches, excluding neural network-based approaches, by performing a k-fold cross-validation across the training set, with $k = 50$. Cross-validation works by splitting the training set randomly into k parts, then holding one of those out for validation and fitting the estimator on the $k - 1$ others. This gets done k times, such that each part has been used for validation once. This returns the average F1-score for all folds. The purpose of cross-validation is to see how well a model is able to generalize to out-of-sample data without involving the test set. Compared with using a single validation set, this is a more robust way of comparing models and tuning parameters without fitting to the test set. However, due to the high computational cost of cross-validation and training neural networks, Models 7 and 8 were not included, but in fact validated on a single validation set sampled from the training set. The best candidates from cross-validation were then compared in a simplified trading simulator.

5.6 Trading application

The most obvious application of these techniques is to create trading signals for automated trading. We propose a simple strategy where one enters a long (short) position when the model predicts a classifies a message as positive (negative) and do nothing when a message is classified as neutral. Each position is held for one trading day. In order to isolate the models' specific return, the return is calculated as excess return compared with the Oslo Stock Exchange index, OSEBX. If the model enters a long position in a stock, it enters a short position on the index, and vice versa.

The models predict probabilities for the three classes, and how certain the model needs to be in order for the prediction to be considered a trading signal depends on the user's utility of wealth function, i.e., their risk preference. The most risk averse trader would look for an estimated class probability of close to 1, whereas the risk neutral investor would only need the probability to be higher than 1/3, meaning anything that is better than random.

A risk-agnostic approach can be to compare the per-sample Sharpe ratio, here calculated as the geometric average percentage return per sample divided by the standard deviation of returns. This provides a more conservative estimate of the Sharpe ratio than if one were to use the arithmetic average.

$$\text{Sharpe} = \frac{\sqrt[N]{\frac{w_N}{w_0}} - 1}{\sqrt{\text{Var}(r)}}$$

Where N is the number of samples, w_i is the value of the portfolio after i samples (trading steps), and $\text{Var}(r)$ is the variance of the percentage return per sample. We select the probability decision threshold which maximizes the Sharpe ratio on the training set for each model, before comparing the models' performance on the test set.

The results of the trading strategy are based on the data in our data set, which means that the return from a trade is calculated as if the stock was bought (sold) at the previous close price before the publication of the corporate announcement, and then sold (bought) at the closing price after the publication of the announcement.

5.7 Polarity detection

While there is certainly loss of profit associated with every form of misclassification, the most damaging errors occur when the model returns a trading signal with the wrong polarity, i.e., the trading algorithm goes long (short) on a message when the associated return in reality turns out to be significantly negative (positive).

We measure the polarity detection performance by the win-loss ratio. It is calculated by dividing the total number of profitable trades by the total number of losing trades. In order to have comparable ratios between model development and trading application, class labels are used to compare wins and losses. This means that returns of less than 0.4% up or down are not included.

$$\frac{W}{L} = \frac{\text{correct non-neutral predictions}}{\text{wrong non-neutral predictions where the true value is also non-neutral}}$$

6 Results

In this section we present the F1-scores from the cross-validation, along with the models' performance on the entire training and test set. One commonality between the models is that they have low recall for neutral samples. This is remedied in the trading application below by requiring a certain minimum estimated probability for the non-neutral class in order to act on it.

Table 2: Models' performance from cross-validation, training set, and test set without any thresholding

Model	CV average F1-score (k=50)	Train F1-score	Train Win/Loss	Test F1-score	Test Win/Loss
Random classifier	0.3325	0.3334	1.0023	0.3356	1.0645
Greedy classifier	0.4236	0.4236	1.0509	0.4076	1.0335
M1: Logistic regression (Word count)	0.4126	**0.5359**	**1.7018**	0.4206	1.1365
M2: Logistic regression (TF-IDF + LSA)	0.4369	0.4523	1.2057	0.4304	1.1602
M3: B. naive Bayes (TF-IDF + LSA)	**0.4422**	0.4472	1.1783	0.4154	1.0755
M4: Adaptive boosting (TF-IDF + LSA)	0.4358	0.4550	1.2180	0.4302	1.1614
M5: Gradient boosting (TF-IDF + LSA)	0.4364	0.4895	1.4307	**0.4338**	**1.1702**
M6: Random forest (TF-IDF + LSA)	0.4362	0.4472	1.1819	0.4224	1.1198
M7: LSTM (GloVe)	N/A	0.5149	1.6380	0.3906	1.0054
M8: LSTM (flair)	N/A	0.5345	1.6826	0.4056	1.0616
M9: Soft vote ensemble	0.4418	0.4710	1.3124	0.4310	1.1434

These results show that models 1–6 and 9 all outperform the baselines on the test set. The first baseline, the random classifier, has an F1-score of $\frac{1}{3}$, recall of $\frac{1}{3}$ and precision of the classes equal to their frequencies. The greedy classifier has a precision for the most common class equal to its frequency, and undefined precision for the two other classes. It has a recall of 100% on the most common class and 0% on the two others. This means that on the "positive" class, it has a precision of 0.43 and a recall of 1.00, which gives an F1-score of 0.60. The two other classes get F1-scores of 0.00. The micro average F1-score turns out to be 0.43. Empirical tests confirmed these estimates for both baselines.

The results from this experiment show that Model 1, logistic regression with basic word count vectorization, does not outperform the baseline in cross-validation, and the high

performance on the training set is likely to be overfitting. Several classification algorithms were tested with this vectorization approach, however, they performed consistently worse than those where TF-IDF and LSA were used. As shown in table 2, Model 1 significantly underperforms the baseline on the cross-validation and has a large gap between training and test-set performance. It is therefore rejected as a candidate for trading purposes. Applying TF-IDF vectorization and including n-grams with $n = [1, 2]$, as well as latent semantic analysis in Model 2, improves the cross-validation and test set results significantly. In Model 3, we replace the logistic regression classifier with a naive Bayes classifier with binarized LSA vectors as input to estimate the probability of each class. This model gives the highest cross-validation F1-score but has weaker performance on the test set than some of the other models.

The three different ensembles of decision trees, models 4, 5, and 6, all outperform the baseline in cross-validation but does not seem to significantly outperform the logistic regression or naive Bayes classifier. Stochastic gradient boosting scores the second highest in cross-validation and shows the best performance on the test set.

Models 7 and 8, the LSTM neural networks perform exceptionally well on the training data, but fails on the test set, which is a clear sign of overfitting.

Model 9, the soft voting ensemble of models 3 and 5, show good performance during both cross-validation, and on the test set, which indicates that it might be one of the most robust models.

6.1 Trading application

This section contains a proof of concept in the form of a trading application using the above results. Figure 3 shows a comparison of per-sample Sharpe ratios against the minimum probability decision threshold. If at any point the portfolio value reaches zero or below, or if the model doesn't make any trades, the Sharpe ratio is considered not-a-number and is excluded from the plot.

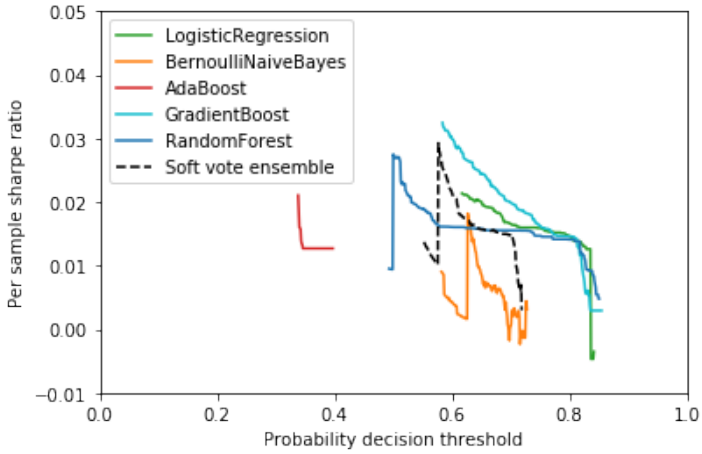


Figure 3: Per sample Sharpe ratios on the training set, for different probability decision thresholds

Table 3: Trading performance on the training and test set, from the most promising models

Model	Probability threshold	Train Sharpe	Train W/L	Train annualized return	Testing set per-sample-Sharpe	Test W/L	Test annualized return
M2: Logistic regression	0.617	0.0216	3.4406	10287.88%	0.0128	1.2500	93.03%
M3: B. naive Bayes	0.627	0.0196	1.7843	200.30%	0.0230	**2.5000**	36.30%
M4: adaptive boosting	0.338	0.0212	**4.6017**	9506.45%	0.0157	1.500	132.55%
M5: gradient boosting	0.584	**0.0329**	4.2129	334633.46%	0.0201	1.2500	394.45
M6: random forest	0.500	0.0282	2.2332	163655.48%	0.0110	1.0000	72.60%
M9: Soft vote ensemble	0.577	0.0132	2.2655	**4600158.39%**	**0.0234**	2.3889	**10411.73%**

The greedy classifier will stay completely passive for a decision boundary above the frequency of the most common class, and the random classifier will stay passive for every decision boundary above $\frac{1}{3}$ because all class probabilities are always exactly equal to $\frac{1}{3}$. They are not included in this section because the portfolio value reached zero for every probability threshold tested using them.

Inspecting the correlation matrix of the models' training set returns (table 4) shows a low correlation between the naive Bayes and gradient boosting classifiers, which are the two models with the highest cross-validation F1-score. Thus, it should be possible to construct a potentially more robust trading algorithm by training a soft vote ensemble classifier based on

those two, which means that it predicts the class based on the average predicted probabilities from these two models.

Table 4: Correlation matrix of training set returns

	AdaBoost	BernoulliNaiveBayes	GradientBoost	LogisticRegression	RandomForest
AdaBoost	1.000000	0.042040	0.867867	0.963418	0.786345
BernoulliNaiveBayes	0.042040	1.000000	0.041219	0.048558	0.038077
GradientBoost	0.867867	0.041219	1.000000	0.862088	0.860812
LogisticRegression	0.963418	0.048558	0.862088	1.000000	0.795496
RandomForest	0.786345	0.038077	0.860812	0.795496	1.000000

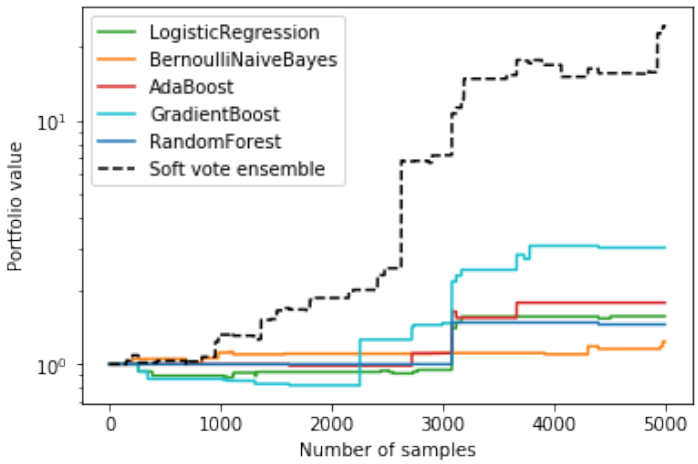


Figure 4: Out of sample returns (logarithmic scale)

Figure 4 shows the return for each model on the out-of-sample data, on a logarithmic scale. The graph makes it clear that the soft voting ensemble performs exceptionally well on the test set, with a large return, and a high win-loss ratio. We can also see from the figure that the gradient boosting classifier has the highest return, and the naive Bayes classifier achieves relatively low return, but also low variance.

All the trading models show strong performance on both the training set and on the test set, all with annualized excess returns above 30%. The returns are much higher on the training set, which can be a sign of some overfitting, however, the cross-validation F1-scores and out-of-sample returns show that the models are likely to have learned some generalizable patterns.

The win/loss ratios of the trading models differ from the measure presented for the individual models above because the certainty threshold is more conservative. The results match the expectation that a higher certainty requirement leads to a higher win-loss-ratio.

One caveat of the simulation is that in reality it might not possible to buy/sell the securities at the previous closing price at any given time, especially when the announcement is published outside of the stock exchange's opening hours, however, the test is a useful demonstration of the strong potential for excess returns, hedged against systemic risk.

6.2 Feature importance

In this sub-section we present the feature importances from the naive Bayes classifier. Feature importance is a measure of how influential each attribute of the input vector is for prediction. This gives an idea of which concepts the model has identified as having a significant effect on the expected change in stock price. We chose Model 3, the Bernoulli naive Bayes classifier, because it had the best performance in cross-validation and has conveniently interpretable coefficients.

The feature importances are represented by the estimated conditional log probability of each class with respect to each attribute. An attribute is an underlying concept identified by LSA, and they are here represented by the 4 most relevant terms for each concept. Because the terms have been stemmed, nouns are shown in singular form, and verbs in present tense when feasible. Table 5 shows the 15 most positive and negative attributes in descending order. We observe that the model emphasizes several terms that are consistent with financial theory presented in section 3.2, and a closer discussion follows below.

Table 5: The 15 most positive and negative features, according to the Bernoulli naive Bayes classifier

Most Positive tokens	Most Negative tokens
purchase, email, share purchase, company purchase	share buyback, bond, bond issue, new bond
company, disclosure, large shareholding, disclosure large	trade hug, company notification email, company
day, capital, today, today day	email, address, shareholder, email email
notification trade, notification, mandatory, trade	ose, investigation, investigate movement, movement share
present, contract, attach, vessel	announce, hugin issuer, origin distribute, announce origin
share buyback, buyback company, number phone, phone	report, annual, calendar, financial calendar
placement, private placement, purchase share, private	special observation, special, rs, oslo rs
annual report, annual, attach, report company	buy, phone number, number, statement
special observation, special, attach, company	share, share buy, purchase share, offer
extraordinary, contract, annual overview, overview	share buy, compon, company share, overview
th, repurch, st, rd	extraordinary, report, average price, average
contract, annual report, quarter result, award	purchase share, purchase, finance, offer
compon, det norsk, det, norsk	buy, quarter, informtion please, conduct oper
offer, state, order, new face	right, subscript, list, quarter
matching halt, match, oslo rs, rs	hold, company notify offic, company share

7 Discussion

Our results have in most cases been consistent with our expectations, and in this section, we discuss the implications of our findings. Overall, most models performed close to our expectations, but the surprising result was that the neural network approaches were the worst performing. One possible reason for this is that corporate announcements by themselves might not actually contain more information than the other models were able to fit to. Corporate announcements are just one of many factors which might affect stock prices, meaning that there is a lot of noise in the data which the neural networks can potentially overfit to. They are also more prone to overfitting, because the text has undergone less preprocessing and might contain noise which has been removed from the input into models 1–6 and 9.

7.1 Market efficiency implications

Our results show that the majority of machine learning methods were able to use information presented in corporate announcements to improve prediction of stock prices and outperform the baselines. This goes directly against the hypothesis of strong form of market efficiency (Fama,

1970), which says that the models should not be able to exploit public information for stock price prediction as both private and public information should already be priced into the stock market. If the definition above was to hold, new public information could not affect the stock price at all, as strong form of efficiency means that even insider trading should be impossible due to information already being priced into the stock market before it is being publicly announced. This is inconsistent with our findings, which thus falsify the assumption of the strong form of market efficiency.

Overall, our results are in line with previous work on stock market prediction using public available information like public sentiment (Bollen, Mao and Zeng, 2011; Li *et al.*, 2014) and news articles (Day and Lee, 2016). The argument in this paper is supported by the mentioned previous research, which found that they could use public available information as an advantage when predicting stock price, further weakening the hypothesis of strong form of market efficiency.

As we see, there are strong indications that information in corporate announcements has an effect on stock prices. If corporate announcements affect the price, there might be windows and anomalies investors could utilize in order to beat the market. However, because of the limitations in the data set, we are unable to identify the exact moment of the stock price movement, meaning we cannot know if the price movement happened at the exact moment, or slightly before or after the announcement was made. Because of this we cannot directly falsify the semi-strong form of market efficiency.

One possible reason that our models are able to achieve the excess return listed in the result section, might be that the market participants are not able to react at the exact moment of the corporate announcement, as machine potentially could have. Human investors might hold inefficient portfolios due to their inability to absorb and react to new and relevant information (Brealy, Myers and Allen, 2017) while machines can analyze new data much faster.

While some previous work were able to identify a window of opportunity investors could exploit (Gidófalvi, 2001; Tuyon and Ahmad, 2016), supporting our results, others found that these windows were unexploitable for trading purposes due to transaction costs (Ţiţan, 2015). The latter finding is somewhat consistent with the perception that anomalies do exists, but where Ţiţan's findings indicated anomalies were unexploitable due to transaction costs, Fama argued that anomalies are neglectable because overreactions and underreactions in the

market happened just as often, making it impossible to earn abnormal return in the long run (Fama, 1997). These findings might explain how some of the models are able to gain excess return over the market. If the models can identify when the anomalies will cause an underreaction in the market and when it will cause an overreaction, they might be able to use this knowledge as an advantage to beat the market.

We have not done any experiments concerning historical prices, so with regards to market efficiency this means we can neither confirm nor falsify the hypothesis of weak form of market efficiency.

Overall, our research coincides with previous work and may be seen to falsify the hypothesis of strong market efficiency on Oslo Stock Exchange. We cannot, however, make decisive conclusions with regards to semi-strong or weak form of market efficiency on Oslo Stock Exchange.

7.2 Trading implications

The results from our trading application show that there is an opportunity for excess return by trading based on information in corporate announcements. We also show that this is a task that can be performed by an automated trading system using machine learning and natural language processing.

From figure 3, we observe that the Sharpe ratio is generally highest when the probability threshold is just above the limit where the models turn a profit on the training set. It is unsurprising, because a lower threshold means that the model makes more trades, and in general, the model makes profitable trades. This means that there is still a risk that the model might provide negative return for the Sharpe ratio maximizing probability threshold, and a more risk averse trader might want to choose a threshold slightly above the optimal. As the threshold is increased, the Sharpe ratio falls, because the model makes fewer trades, and loses out on some profit in exchange for reduced risk. When the threshold reaches a certain point, the models make no trades, and thus zero return.

As we see from table 3, the best models have a win-loss ratio on the test set above 2, which means that they make a profit on more than 66% of the trades they make. The win-loss ratio is not a perfect metric by itself, because it disregards the value of the wins and losses, but in conjunction with the other metrics this is a strong indicator that the models have the ability

to produce positive excess return over time. A high win-loss ratio seems to go together with a high Sharpe ratio; however, they do not seem to correlate with a high annualized rate of return. We consider the annualized rate of return to be of less importance when evaluating the models, because it can be compensated for by leveraging or deleveraging the investment in a model with a higher Sharpe ratio, and in doing so achieving lower risk with the same expected return.

All the models perform better on the training set than on the test set, which can be a sign of slightly overfitting. However, they consistently score higher in cross-validation than on the test set, indicating that the test set has a higher difficulty than the training set. This strengthens the argument that the models are able to make a positive excess return on out-of-sample data, as they still perform well on the test set.

There is always uncertainty associated with making statements about the future based on historical data, and some there are limitations when back testing a trading strategy, such as liquidity concerns and transaction cost. Further, the model does not consider possible differences between publication times (companies might favor publishing certain announcements outside of the stock exchange opening hours). Thus, the exact rates of return from the trading simulation should not be taken at face value; rather, they should be interpreted as a strong indicator that corporate announcements are likely to be a useful addition as a basis for decision making in an automated trading application.

7.3 Feature importance

Since the models have shown an ability to learn patters in the announcements which predict the change in stock valuation, it is interesting to look at what those patterns are. At first glance, it is not obvious how to interpret the feature importances from the naive Bayes classifier because they are a reduced representation of the underlying concepts and their interactions. However, some of the terms stand out.

First of all, it seems like the model places a high value on announcements which talk about contracts. This means that companies which publish announcements about a contract are expected to increase in market valuation. This makes intuitive sense, because this might represent an unexpected increase in revenue, or at that the company has secure revenue for the duration of the contract, which could be several years. The result is either a likely boosting to profitability or a reduction in risk (or both), which are two factors most investors value.

Secondly, terms alluding to annual reports and share repurchase appear to be both highly positive *and* negative. This could mean that the model simply expects these announcements to be very non-neutral, without being able to predict the direction of change. With regards to the publication of annual reports, this comes as no surprise; the financial reports can either disappoint or positively surprise investors – either way they tend to cooccur with large price movements. With regards to the terms "share buyback" and "share repurchase" it is slightly less intuitive, as previous research has found that the price tends to increase after such an announcement has been published (Brealy, Myers and Allen, 2017). One would therefore expect the terms to be explicitly positive. One possible explanation that share buyback is not strictly interpreted as positive might be that even though it indicates an optimism in the firm and reduces the number of stakeholders, the firm has to spend cash on buying back shares, reducing short term liquidity. Some might interpret this as a signal of lacking investment opportunities, and thus reducing expectations of growth. Another possible explanation might be that the share repurchase was expected and already priced in but leads to volatility if it differs from expectations.

Furthermore, the tokens "bond issue" and "new bond" are categorized with a negative effect on stock price movement. This is contradictory to the Modigliani and Miller theory, and consistent with research claiming that issuing new bonds are likely to affect stock price negatively. There seems to be consensus in literature that issuance of convertible bonds have a negative impact on stock prices (Mikkelson and Partch, 1986). The term "convertible" does not appear as an important term, but it is possible that the model has simply learned to associate the issuance of straight and convertible bonds as a single concept, and thus mistakenly classify all bond issuance as a negative indicator.

The Oslo Stock Exchange has multiple tools available to regulate the trading process (OsloBørs, 2019). Our model considers two of those tools to be amongst the most important features: *matching halt* and *special observation*. When the exchange imposes a matching halt, it means that they stop automatically matching buyers and sellers, effectively preventing trades which are not done in person or over telephone. This can be done because of the publication of especially price sensitive announcements, if there is suspicion that some market participants hold private information regarding the company, or if there has been suspicious trading activity. This, according to the model, is associated with a price increase. Special observation has no

direct effect on the trading process on the stock but serves as a signal from the exchange that there currently is particular uncertainty regarding the valuation of a stock. The model expects large price movements when a stock is placed under special observation, but does not know the direction, as it appears as both one of the most positive and one of the most negative attributes.

In the cases when suspicious movements in a company's valuation lead to an investigation into the respective trades, to determine if an error in the system or some illegal activity has occurred, the model unsurprisingly considers this to be a strong negative indicator.

8 Conclusion

We have shown that natural language processing and machine learning can extract information about the market's reaction to publication of certain corporate announcements on the Oslo Stock Exchange and make predictions about resulting stock price movements. Following these results, we argue against strong market efficiency on the Oslo Stock Exchange.

Examining the feature importances of Model 3 shows that it has identified terms and concepts associated with an increase or decrease in valuation which are consistent with financial literature and economic intuition.

Traditional machine learning approaches outperformed the more cutting-edge deep learning-based models, but as our experiments with the latter were limited, we highlight it as a promising area of research. The Bernoulli naive Bayes classifier, with TF-IDF and LSA, was the model that showed the best ability to generalize to our data set, and had the second highest per sample Sharpe ratio and the highest win-loss ratio on the out-of-sample trading test. The best Sharpe ratio was achieved by combining the naive Bayes and gradient boosting classifier in an ensemble.

From an instrumental perspective, the results in this paper are promising. The goal of most computational trading models is to predict which stocks will provide excess return over the market index, and we have in this paper shown several models that can do so. We presented several models that outperformed the baseline classifier and combined the two most promising in an ensemble. We then presented a proof-of-concept long/short hedging strategy which showed that our models have the potential to produce strong excess return over the Oslo Stock Exchange benchmark index, and that corporate announcements should be included in an automated trading strategy.

8.1 Future work

Although the results in this paper are promising, there are still many opportunities for further experimentation. Firstly, it would be interesting to study price history on a more granular level, for example using down to minute even millisecond resolution to get a better understanding of the level of market efficiency on Oslo Stock Exchange. Additionally, more time could be invested in optimizing the hyperparameters of the methods used and testing more combinations of vectorization and classification techniques. Deep learning methods, for example Google's BERT (Devlin *et al.*, 2018) and OpenAI's GPT-2 (Radford *et al.*, 2019), have shown exceptional results in many NLP tasks, and we therefore believe they have more potential for this research problem than our approaches achieved and that they are a promising direction of future research. Furthermore, since corporate announcements do not exist in a vacuum, one could expect multimodal models, i.e., combining the data set with more information, such as earlier corporate announcements, financial reports or even stock price history, might improve the results as it provides some context to the announcements. Lastly, it could be useful to develop a multi-lingual model. In this case, if one were to include announcements published in Norwegian it would more or less double the number of labeled announcements, which could lead to the models being more robust to overfitting.

As we can see, there are many opportunities for future research in the field of natural processing language for finance, and we believe that text analytics shows great promise for improving stock price forecasts and decision making for traders.

Acknowledgements

We are grateful for the support from our supervisors, associate professor Arild Brandrud Næss and associate professor Denis Mike Becker, for lending us their expertise and advice. We also thank the rest of the academic staff at NTNU Business School for providing feedback on our presentations during seminars. Any mistakes in this paper are our own and should not in any way reflect poorly on the reputations of our esteemed colleagues.

Finally, we are grateful to the NTNU High-Performance Computing Group for generously granting us access to their data cluster for running our computations.

References

Aizawa, A. (2003) 'An information-theoretic perspective of tf-idf measures', *Information Processing and Management*, 39(1), pp. 45–65. doi: 10.1016/S0306-4573(02)00021-3.

Akbik, Alan, Duncan Blythe, Roland, V. (2018) 'Contextual string embeddings for sequence labeling', *In Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649. Available at: http://aclweb.org/anthology/C18-1139.

Bollen, J., Mao, H. and Zeng, X. (2011) 'Twitter mood predicts the stock market', *Journal of Computational Science*. Elsevier B.V., 2(1), pp. 1–8. doi: 10.1016/j.jocs.2010.12.007.

Brealey, R. A., Myers, S. C. and Allen, F. (2017) 'Valuating bonds', in *Corporate Finance*. 12. editio. McGraw-Hill Education, pp. 46–75.

Brealy, R. A., Myers, S. C. and Allen, F. (2017) 'Payout policy', in *Principles of Corporate Finance*. 12. editio. McGraw-Hill Education, pp. 410–135.

Breiman, L. (2001) 'Random Forests - Original Paper', *Vasa*, pp. 1–33. Available at: http://oz.berkeley.edu/~breiman/randomforest2001.pdf%0Ahttp://medcontent.metapress.com/index/A65RM03P4874243N.pdf.

Day, M. Y. and Lee, C. C. (2016) 'Deep learning for financial sentiment analysis on finance news providers', *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*. IEEE, (1), pp. 1127–1134. doi: 10.1109/ASONAM.2016.7752381.

Dolan, R. J. (2002) 'Emotion, cognition, and behavior.', *Science (New York, N.Y.)*. American Association for the Advancement of Science, 298(5596), pp. 1191–4. doi: 10.1126/science.1076358.

Fama, E. F. (1970) 'Efficient Capital Markets : A Review of Theory and Empirical Work', *The Journal of Finance*, 25(2), pp. 383–417.

Fama, E. F. (1997) 'Market Efficiency, Long-Term Returns, and Behavioral Finance', *Journal of Financial Economics*, 49, pp. 283–306. doi: 10.2139/ssrn.15108.

Friedman, J. H. (1999) 'Stochastic gradient boosting', *Technical Discussion of TreeNet*, pp. 1–10.

Friedman, J. H. (2001) 'Greedy Function Approximation: A gradient boosting

Machine', *Annals of statistics*. doi: 10.1214/aos/1013203451.

Gidófalvi, G. (2001) 'Using New s Articles to Predict Stock Price Movements', (December 2004).

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. second edi. Springer. doi: 10.1007/978-0-387-84858-7.

Jensen, M. C. (1978) 'Some anomalous evidence regarding market efficiency', *Journal of Financial Economics*, 6(2–3), pp. 95–101. doi: 10.1016/0304-405X(78)90025-9.

Jukes, E. (2018) *Encyclopedia of Machine Learning and Data Mining (2nd edition)*, *Reference Reviews*. doi: 10.1108/rr-05-2018-0084.

Jurafsky, D. and Martin, J. H. (2014) *Speech and Language Processing (2008)*. doi: 10.1162/089120100750105975.

Karalevicius, V. (2018) 'Using sentiment analysis to predict interday Bitcoin price movements', *Journal of Risk Finance*, 19(1), pp. 56–75. doi: 10.1108/JRF-06-2017-0092.

Laham, P. W. (1998) *Introduction to Latent Semantic Analysis*. Available at: http://lsa.colorado.edu/papers/dp1.LSAintro.pdf (Accessed: 16 May 2019).

Landauer, T. K., Foltz, P. W. and Laham, D. (1998) 'An Introduction to Latent Semantic Analysis 1995 cited 2k.pdf', *Discourse processes*, pp. 259–284.

Li, Q. *et al.* (2014) 'The effect of news and public mood on stock movements', *Information Sciences*. Elsevier Inc., 278, pp. 826–840. doi: 10.1016/j.ins.2014.03.096.

Li, S. (2018) *Latent Semantic Analysis & Sentiment Classification with Python*. Available at: https://towardsdatascience.com/latent-semantic-analysis-sentiment-classification-with-python-5f657346f6a3 (Accessed: 1 May 2019).

Liu, H. (2018) 'Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network', *Department of Electrical and Computer Engineering*. Available at: http://arxiv.org/abs/1811.06173.

Manning, C. D., Raghavan, P. and Schütze, H. (2008) 'Ch 18 - Matrix decompositions and latent semantic indexing', in *Introduction to Information Retrieva*. Cambridge University Press, pp. 403–419.

Mason, L. *et al.* (2000) 'Boosting Algorithms as gradient Descent', *Advances in neural information processing systems*, pp. 512–518. doi: 10.1103/PhysRevD.91.072004.

Maynar, D. and Bontcheva, K. (2014) 'Natural Language Processing', in *Perspectives on ontology learning*. Sheffield: Dept. of computer Science, University of Sheffield, UK, pp. 51–67.

Mikkelson, W. H. and Partch, M. M. (1986) 'Valuation effects of security offerings and the issuance process', *Journal of Financial Economics*. North-Holland, 15(1–2), pp. 31–60. doi: 10.1016/0304-405X(86)90049-8.

Myers, S. C. and Majluf, N. S. (1984) 'Corporate financing and investment decisions when firms have information that investors do not have', *Journal of Financial Economics*. North-Holland, 13(2), pp. 187–221. doi: 10.1016/0304-405X(84)90023-0.

Næss, A. B. *et al.* (2017) 'Konkursprediksjon for norske selskaper - En sammenligning av regresjonsmodeller og maskinlæringsteknikker', in *Bred og spiss! : NTNU Handelshøyskolen 50 år : en vitenskapelig jubileumsantologi*. fagbokforlaget.

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

Pennington, J., Socher, R. and Manning, C. D. (2014) *GloVe: Global Vectors for Word Representation*. Available at: https://nlp.stanford.edu/projects/glove/ (Accessed: 1 May 2019).

Ramos, J. (2003) 'Using TF-IDF to Determine Word Relevance in Document Queries', *Proceedings of the first instructional conference on machine learning*, 242, pp. 133–142. doi: 10.15804/tner.2015.42.4.03.

Schapire, R. E. (2013) 'Explaining AdaBoost', in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pp. 37–52.

Schumaker, R. P. *et al.* (2012) 'Evaluating sentiment in financial news articles', *Decision Support Systems*. North-Holland, 53(3), pp. 458–464. doi: 10.1016/J.DSS.2012.03.001.

Schwert, G. W. (2003) 'Chapter 15 Anomalies and market efficiency', in *Handbook of the Economics of Finance*. Elsevier, pp. 939–974. doi: 10.1016/S1574-0102(03)01024-0.

Seng, J. L. and Yang, H. F. (2017) 'The association between stock price volatility and financial news – a sentiment analysis approach', *Kybernetes*, 46(8), pp. 1341–1365. doi: 10.1108/K-11-2016-0307.

Ţiţan, A. G. (2015) 'The Efficient Market Hypothesis: Review of Specialized Literature

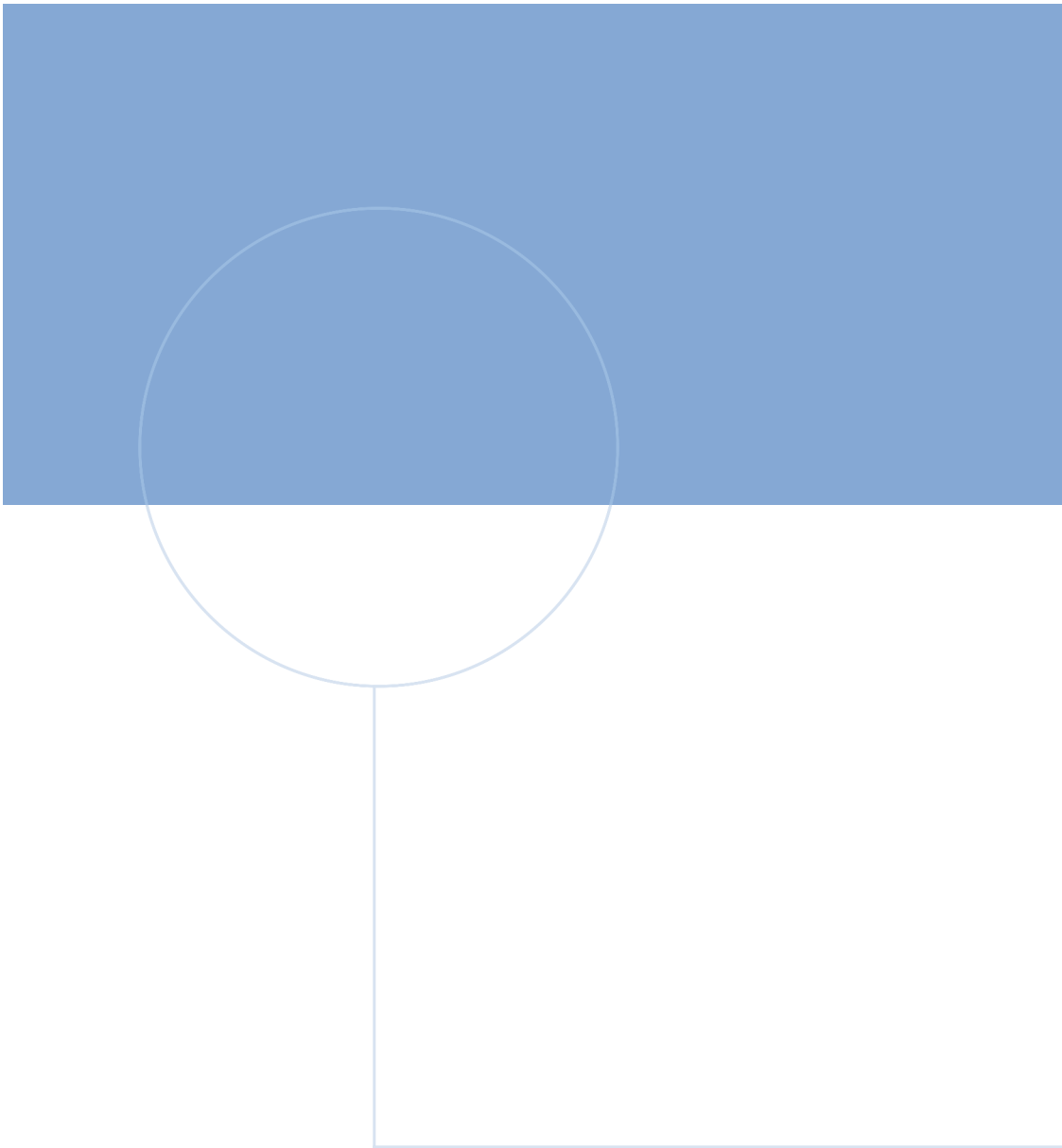
and Empirical Research', *Procedia Economics and Finance*, 32(15), pp. 442–449. doi: 10.1016/S2212-5671(15)01416-1.

Tuyon, J. and Ahmad, Z. (2016) 'Behavioural finance perspectives on Malaysian stock market efficiency', *Borsa Istanbul Review*. Elsevier Ltd, 16(1), pp. 43–61. doi: 10.1016/j.bir.2016.01.001.

Webb, G. I. (2018) 'Naïve Bayes', in *Encyclopedia of Machine Learning and Data Mining*, pp. 879–880.

Whitaker, B. (2018) *[EMNLP] What is GloVe? Part I – Towards Data Science*, *Towards Data Science*. Available at: https://towardsdatascience.com/emnlp-what-is-glove-part-i-3b6ce6a7f970 (Accessed: 1 May 2019).

Zhang, L., Wang, S. and Liu, B. (2018) 'Deep Learning for Sentiment Analysis : A Survey'. Available at: http://arxiv.org/abs/1801.07883.



NTNU

Kunnskap for en bedre verden