UPPSALA
UNIVERSITET

UPTEC F 16030
Examensarbete 30 hp
Juli 2016

Analysis of stock forum texts to examine correlation to stock prices

Olof Norlander

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Analysis of stock forum texts to examine correlation to stock prices

Olof Norlander

In this thesis, four methods of classification from statistical learning have been used to examine correlations between stock forum discussions and stock prices. The classifiers Naive Bayes, support vector machine, AdaBoost and random forest, were used on text data from two different stock forums to see if the text had any predictive power for the stock price of five different companies. The volatility and the direction of the price - whether it would go up or down - over a day was measured. The highest accuracy obtained for predicting high or low volatility came from random forest and was 85.2 %. For price difference the highest accuracy was 69.2 %, using the support vector machine. The average accuracy for predicting the price difference was 58.6 % and the average accuracy for predicting the volatility was 73.4 %. This thesis was made in collaboration with the company Scila which works with stock market security.

Handledare: Lars Gräns
Ämnesgranskare: Michael Ashcroft
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F 16030

Popular scientific summary in Swedish

I och med Internets framsteg finns mer och mer information tillgänglig på nätet. Där uttrycker folk sina åsikter och tankar om saker och ting på sociala medier och olika bloggar och forum. En av dessa saker är företag och dess aktier. På olika aktieforum skrivs det positiva och negativa inlägg om företag. Många personer som handlar med aktier läser dessa inlägg för att få information om företag och se vad andra personer har för åsikter. Detta kan i sin tur påverka hur de väljer att handla med aktierna. Om många skriver positiva saker om ett företag kan det resultera i att ännu fler vill handla med deras aktier vilket i sin tur leder till att aktiekursen stiger. Å andra sidan kanske det skrivs negativa saker om ett annat företag och folk börjar sälja av sina aktier i det företaget och aktiekursen sjunker. Kort sagt kan skriverier på olika bloggar och forum påverka ett företags aktiepris. Detta kan utnyttjas och en del försöker aktivt manipulera aktiekursen för att tjäna pengar. Manipulation av aktiepriser är olagligt och därför är det av intresse att upptäcka sådana försök.

Med hjälp av maskininlärning går det att bygga statistiska modeller för att försöka förutspå aktiepriset. Med dessa kan man se vilka ord som är hjälpsamma för dessa förutsägelser. Detta är ord som tycks korrelera väl med hur priset ändras och genom att undersöka dessa ord kan man se om det verkar vara något som kan tänkas påverka priset. Då kan man gå vidare och se om vissa användare använder dem frekvent. Detta skulle då kunna tyda på försök till prismanipulation och en mer grudlig undersökning kan inledas. Utöver detta kan det även vara intressant ur en investerares synpunkt att förutspå aktiepriest då man kan välja att sälja och köpa aktierna vid rätt tidpunkt.

De olika maskininlärnings-metoderna Naive Bayes, random forest, AdaBoost och suppot vector machine har använts för att förutspå om volatiliteten har varit hög eller låg samt om priset har gått upp eller ner under en dag med hjälp av inlägg folk har skrivit på två olika aktieforum. Undersökningen har gjorts för fem olika företag som haft volatila aktier. Detta då aktiepriset för företag med volatila aktier kan tänkas vara mer lättpåverkat av skriverier på forum. Träffsäkerheten för att förutspå volatiliteten har varit runt 60 till 85 % och för att avgöra om priset går upp eller ner under en dag har träffsäkerheten varit mellan 50 och 70 %.

Då det går att förutspå aktiepriserna med hjälp av inlägg på forumen finns det en viss korrelation. Det går dock inte att avgöra om det är ett kausalt samband eller ej. En möjlighet är att det snarare är aktiepriserna som påverkar vad folk skriver på forumen.

Contents

1 Introduction

Predicting stock prices is an interesting task. The stock market moves in many ways and many factors are affecting it. To start with, supply and demand affects the price, if many people are buying a stock, the price increases and if more people are selling it, the price decreases. [Investopedia] More specifically, the forces that move the prices can fall into three different categories: fundamental factors, technical factors and market sentiment. Fundamentals refer to a combination of an earning base and a valuation multiple. Technical factors comprise of external conditions that influence the supply and demand of a company's stock. There are many technical factors but examples are inflation and trends (for example, a stock that is moving up can gather momentum which influences more people to buy it). The last category – market sentiment, is the psychology of the participants of the market, both individually and collectively. [Harper]

People's sentiment towards a company can have a big impact on its stock price. If people have a positive sentiment towards a company, they want to buy their stocks and hence the price increases. On the other hand, if they have a negative sentiment (maybe they do not believe in the company or their products) they might sell off their stocks if they have any, which would cause the stock price to decrease. The problem is that it might be difficult to get hold of people's sentiment towards a company. One option is to make a sounding, simply ask people what they think about something, but that requires a lot of work and the data acquired might not be sufficient. With the emergence of big data and social media things have changed drastically however, and people express their opinion about a multitude of things online. One of these things is companies and their stock. On online forums devoted to this many people write a large amount of posts every day to discuss specific companies and how their stock is changing over time. People can express belief or disbelief about different companies and discuss with each other. On these forums one can obtain sentiment without having to do a sounding. People have already shared their thoughts without being asked. As earlier mentioned, the sentiment towards a company might affect the stock price. Why is this interesting? There are several answers to that question. One is from the company's point of view: if people are writing bad things about it they might want to know what they are doing wrong. In that way they can react to it and change their behaviour in a way that people find positive. Another answer is from the point of view of an investor: if the sentiment towards a company affects the price, it might be possible to predict how the price changes and then it is possible to make a profit out of it from

buying and selling at the right time respectively. This has other implications as well. One interesting such is that people try consciously to change the price by writing certain things. This kind of behaviour is illegal and hence it is of interest to find people who are doing it.

There has been a few occassions on which people have been charged with stock market manipulation, using social media. Some of those will be mentioned in section 2.

1.1 Statistical learning

What can be used for the task of analysing correlations between online forums and the stock market by predicting the stock prices is an area known as statistical learning. Statistical learning has a wide range of applications in several fields such as science, finance and industry. It can be used to predict how likely a person is to suffer from a second heart attack and identify factors behind it. It can also be used to identify handwritten text, or whether an email is spam or not. A spam detector, for instance, could work by checking the occurence of words in an email. Spam emails and proper emails are both characterised by different words, and if a word characteristic for spam has a high frequency in an email it could be classified as a spam mail. [Hastie, Tibshirani & Friedman, 2009]

A classic test case in statistical learning for classification techniques is the Iris data set shown in figure 1.1 on the next page. It was used by Ronald Fisher in 1936 when he with linear discriminant analysis tried to separate three related species of the iris flower based on their morphologic variation. Iris setosa, iris versicolor and iris virginica are similar in appearance but looking at their sepal and petal width and length one might differentiate them from each other. In figure 1.1 we can see that the setosa species (in red) is separable from the others by just looking at one of the features, such as the petal width or length. For versicolor and virginica the task is harder and the other features have to be considered too. [Fisher, 1936]

Another example of a machine learning task is that of a program learning to play a game like checkers or backgammon. Given a specific board state it can learn what would be the best possible move to make by playing a lot of practice games. [Mitchell, 1997]

Statistical learning can be split into two main categories, supervised and unsupervised learning. What differs unsupervised learning from supervised learning is that there is no response variable to supervise the analysis - we know X but not Y. The concept of supervised and unsupervised learning

will be covered more in the theory section.

1.2 Purpose

Scila AB is a company working with market surveillance. Hence it is in their interest to find out about attempts at stock market manipulation. An area that they do not cover yet, however, is online discussion forums. For this reason, analysing whether the writings on these forums actually have any influence on the stock market is of importance to them.

1.3 Goal

The goal is to use the data from the forums to see if it is possible to build a model which can predict how the stock price changes during a day. Furthermore it is of interest to see if certain words have more influence than others on the price change.

1.4 Limitations

The study is limited to data from two popular online forums, which are both in Swedish. Also the analysis will cover only five different companies. Furthermore it is limited to only predict the price given the texts from the forum rather than the sentiment of those texts, hence a sentiment analysis will not be performed.

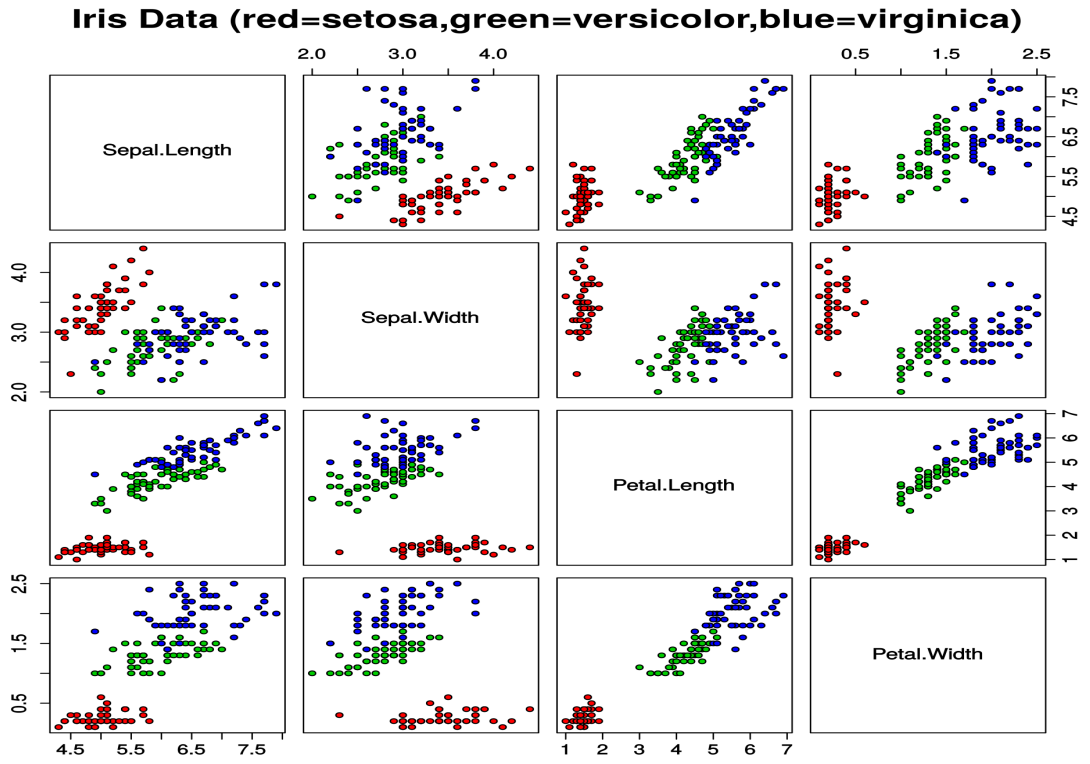

Figure 1.1 Scatterplot of the iris data set. [Wikipedia, 2015]

2 Background

2.1 Stock price manipulation incidents

In recent years there has been a number of instances where people have tried to manipulate the stock price through online communication. In October 2013 a false press release was made through the service company Cision. According to this press release, Samsung Electronics had bought the Swedish company Fingerprint Cards (a company working with fingerprint biometrics) and the result was that the stock price for Fingerprint Cards skyrocketed within minutes. The problem is that the press release was fake. Samsung had never actually bought Fingerprint Cards, the whole thing was a scam. Someone had just forged the press release to make a profit from the impact on the stock market. All trades with the stock (FING B) were cancelled and further trading stopped but the impact of the release was not limited to the Fingerprint stock alone. Another company – Precise Biometrics – working in the same field as Fingerprint Cards was affected, with increasing stock prices as well and other competitors were also affected. [Stengård, 2013]

Another example of such incident is from 2015 when two medical students were risking prosecution for trying to manipulate the stock price of pharmaceutical companies. The two students bought stocks in a company and used their knowledge in medicine to write positive reviews about the companies on stock forums in order to boost the price. After the price had increased they would sell their stocks and move on to a new company. This is an interesting question about the grey zone between freedom of speech and inappropriate market manipulation. The crime classification was grave inappropriate market manipulation (in Swedish: grov otillbörlig marknadspåverkan). [VA, 2015]

Furthermore there is also an example of how a Scottish man used Twitter to manipulate the stocks of an American company. In January 2013, he created Twitter accounts with names similar to those of famous research firms and then tweeted that two companies were facing federal investigations when, in fact they were not. This caused those companies' stock prices to plummet and he used his girlfriend's account to buy stocks at the depressed price. [The Guardian, 2015]

2.2 Related work

In 2014, results by Wu et al. suggested that there were strong correlations between the sentiment on

online forums and the volatility of stocks. They had used a support vector machine model to classify the sentiment of posts on the website Sina Finance and then used it on a GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model for the financial time series. [Wu, Zheng & Olson, 2014]

In a work by Ranco et al., they look at the twitter volume and sentiment over a period of 15 months for 30 stock companies. They find a low correlation between the time series over the whole time period but also a significant dependence between the sentiment on Twitter during the peak of Twitter volume, and abnormal returns from the stock. [Ranco et al, 2015]

Nguyen et al use the support vector machine to predict whether the price would go up or down on a day. This was done using the sentiment from the Yahoo Finance message board and they achieved an accuracy ranging from 54 % to 71 %. [Nguyen, Shirai & Velcin, 2015]

Schumaker and Chen use financial news to predict if the stock price will go up or down 20 minutes after a news article is released in another work. The average directional accuracy they achieved was 58.17 %, using the support vector machine. In their model, only proper nouns were chosen from news articles to serve as text features. [Schumaker & Chen, 2009]

In a study by Si et al., the Vector Autoregressive Model is used with a sentiment time series (where they have analysed the sentiment from Twitter) to predict whether the price will go up or down. With their models they managed to achieve accuracies around 60 %. [Si et al, 2014]

3 Data

The data that has been used can be divided into two main categories, those are text data and stock data. The text data consists of all the posts people have written on the online forums. That is the data that will be used to predict the changes in the stock price. The stock data is the simply the data that tells us how the stock price has changed.

3.1 Text data

The text data comes from the discussion forums on Avanza, a Swedish bank, and the economy section of Flashback which is a forum for discussing many different topics. From these two sources, all posts from threads concerning the companies being analysed were used as the input data to try predicting the stock data.

A problem with the text data was that the posts were not annotated with sentiment so it was not possible to use the sentiment directly to make the analysis. Upon investigation the posts were typically noisy and many posts contained no information that seemed relevant to the company for which the thread was about. Also many posts were more like a discussion between users on the forum, for instance someone could write something about the company (or something completely irrelevant), another user comments on that post but points out that the author of the original post was writing poorly which would obviously be irrelevant for the change in the stock price. This differed a bit between the two sites, however, and on Flashback, people tended to stick more to the subject.

3.2 Stock price data

For the stock data there are different interesting aspects to look at. One is the price difference over a time interval, for example the difference between the closing price of a day and the opening price that day. It tells us if the stock has increased or decreased in value. Another one is the swing which is a measure of the volatility of the stock. The swing is defined as the difference between the highest and the lowest traded price of a stock for a particular time interval, in example for a day. A highly volatile stock has many and high changes in its price while the price of a not so volatile stock is more stable. Basically the price of a volatile stock fluctuates more over time than the price of a

stable stock. Figure 3.2.1 shows a volatile stock in green (Fingerprint Cards) versus a less volatile stock in blue (PepsiCo). It depicts the closing price for each stock and each trading day over a one year period, from May 2015 to May 2016 and was generated in R using price data from Nasdaq OMX Nordic for Fingerprint Cards and Google Finance for PepsiCo.

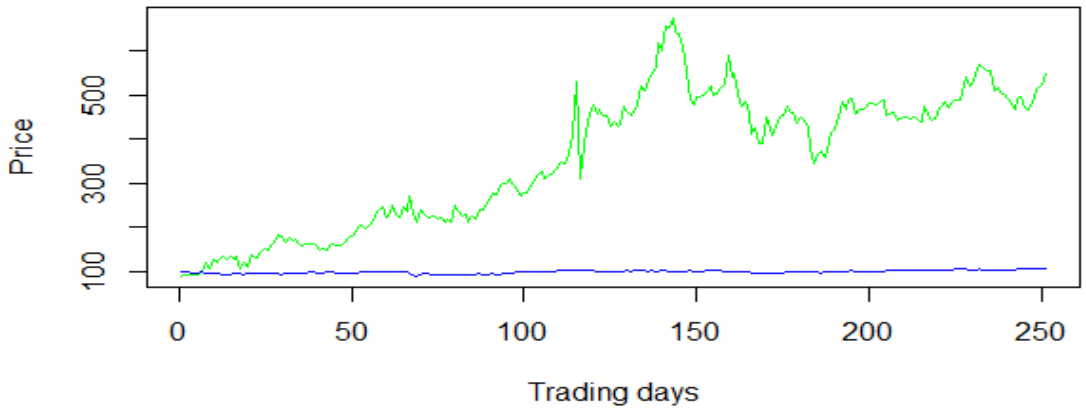

Figure 3.2.1
Volatile stock in green represented by Fing B and
non volatile stock in blue by PepsiCo.

4 Theory

In this part, the theory behind the problem will be discussed. The topics of text mining, features, statistical models and model evaluation will be covered. This will be followed by a brief discussion about causality.

4.1 Text mining

Text mining is the process of obtaining text and extracting useful information from it. Text mining is a wide area and includes many different categories such as information retrieval, text classification and clustering, entity, event and relation extraction. [Kao & Poteet, 2007] Hence, many of the aspects of this thesis falls within the realm of text mining. From the starting point of getting the text to the processing of it and to the methods of obtaining something useful from it.

4.2 Features

It is hard to make statistical models using the words in a text directly. Rather some sort of mathematical representation of the text has to be used instead. With the vector space model, documents can be represented as a vector of the terms, where the terms are represented by real numbers, reflecting their importance in the text. [Raghavan & Wong, 1986]

According to Murty and Devi (2015), “A feature is a property or characteristic of a pattern.”. They use the word pattern since their book covers pattern recognition but other words could be point, vector, sample or instance. Other words for feature could be variable or attribute. As an example, if we would like to classify a car and a truck the features length, height or weight could be used. When working with texts, for classifying a document regarding football and a document about religion, features such as the words goalkeeper, ball, church and pray could be considered instead. There are different ways to use these features; we could look at whether a word is present or not but we could also use the count of that word in the text or some other weighting method.

Choosing the features is an important part of the problem and will have a large impact on how well the model performs. All parts of the text will not be useful if we want to classify a text by its sentiment or category. Due to its importance, feature selection and some methods for it will be covered more thoroughly in section 4.2.2.

4.2.1 Feature engineering

There are different methods to process the text to get a useful feature representation. Some of these are n-gram representation, tokenisation, stemming and lemmatisation.

Tokenisation is the process of breaking a text down to smaller parts or tokens. Examples of these tokens could be sentences, paragraphs, words or symbols.

A similar approach to tokenization is n-grams. An n-gram is a sequence of n tokens. When n is one it is called a unigram, when it is two it is called a bigram, three yields a trigram and so on. Table 4.2.1 shows the n-gram representation of the sentence "The weather is great today" with n as one, two and three.

Table 4.2.1: n-gram representations of the sentence "The weather is great today"

n-gram	Sentence	Size
1-gram	The, weather, is, great, today	5
2-gram	The weather, weather is, is great, great today	4
3-gram	The weather is, weather is great, is great today	3

Using n-grams allows the capture of relations between adjacent words. An example of this is how the meaning of the sequence "not good" differs when using a unigram and a bigram respectively. Using a unigram representation will treat the words separately while a bigram representation will take their relation into consideration so that "not good" can be represented as bad.

Stemming deals with the inflection of words and results in a reduced form of the word, the word stem. An example of how this works is how the inflected forms of the word argue are reduced to a base. Consider argue, argues, argued and arguing. Stemming these words will result in the stem argu which is not a word itself but can be used to represent the mentioned forms of the word argue. The goal of this is to have a single formulation of the word that can be used as a feature rather than using all the different forms as separate features. Several stemming algorithms exist, examples of those are the Snowball stemmer and the Porter stemmer.

Lemmatisation is similar to stemming but has some important differences. Like stemming the goal is to represent a group of words as a single word to reduce the amount of features. Lemmatisation

works by determining the lemma for a word. A lemma is the canonical, or dictionary form of a word. To illustrate this we can consider different forms of the word “be", such as "is", "are" and "were". The lemma is "be" and a lemmatiser would count "is", "are" and "were" as "be" as well. A stemmer would not since the stem is different. Meanwhile a lemmatiser would also count the different forms of "argue" as "argue", like a stemmer. Lemmatisation could also takes synonyms into consideration, as an example both "car" and "automobile" could be considered as the same word.

4.2.2 Feature selection

Feature selection is made to create a subset of the features to be used in the model. When working with text data, many of the features will be irrelevant for the model. There are many words or features that have little impact on the meaning or sentiment of a text. Examples of such features could be words like with, and, the, of, for, or symbols like question marks, asterisks, brackets and so on. Selecting a subset of features reduces the dimensionality of the problem by decreasing the amount of data. This will lead to shorter training times for the models, reduced risk for overfitting and also a simpler model which might be easier to interpret.

There exists a number of methods for feature selection. A few of these are the chi-squared method, information gain and TF-IDF (Term Frequency – Inverse Document Frequency). In this paper the TF-IDF method and information gain will be discussed as those are the ones that have been used.

Term frequency – inverse document frequency

Term frequency – inverse document frequency or TF-IDF is a measurement of a term's importance in a collection of documents. The idea is that words that occur often in a document are important but at the same time words that appear in all documents carry little information. That could be words such as "the" or "and" which will not tell us a lot about the document. The term frequency is the number of times a term appears in a document, $tf(t,D) = f_{t,D}$ where t is the term, D is the document and f the frequency. The inverse document frequency reflects how important a word is and takes into account if it is common or rare in all the documents. The term frequency treats the words as equally important but the inverse document frequency changes this by reducing the weight of words that appear in many different documents. [Leskovec, Rajaraman & Ullman, 2014] It can be defined as

$$\log\left(1+\frac{N}{n_t}\right) \quad (4.2.1)$$

where N is the number of documents and n_t the number of documents containing the term. With this given, TF-IDF can be defined as follows:

$$TF-IDF=\frac{f_{t,d}}{|d|}\cdot\log\left(1+\frac{N}{n_t}\right). \quad (4.2.2)$$

The term frequency has here been normalised by the number of terms in document d. [Metzler, 2008]

Information gain

Information gain uses the measure entropy to find the best features. Entropy characterises the purity or impurity of an arbitrary collection of examples. Entropy, which can be considered as a measure of randomness can be defined as

$$H(Y)=-\sum p(y)\log_2(p(y)) \quad (4.2.3)$$

where Y is a class feature and p(y) the marginal probability density of Y in a training set S. If the observed values of Y are partitioned according to another feature X and the entropy of Y due to the partitions is smaller than the entropy before the partition, there is a relationship between X and Y. The entropy of Y after observing X would then be

$$H(Y|X)=\sum p(x)\sum p(y|x)\log_2(p(y|x)). \quad (4.2.4)$$

With entropy as a criterion of impurity in the set S, a measure that reflects additional information about Y that X gives, can be defined. This measure is the decrease of entropy in Y and is called information gain and can be written as following

$$IG=H(Y)-H(Y|X)=H(X)-H(X|Y). \quad (4.2.5)$$

Information gain works as an indicator of the dependency between X and Y. For the purpose of feature selection, the features are ranked according to their entropy and the information gain filter evaluates the features based on their information gain, considering one feature at a time. [Bolón-

Canedo et al, 2015]

4.3 Statistical models

"Statistical learning refers to a set of tools for modeling and understanding complex datasets." It connects with computer science and machine learning especially and is a novel field in statistics [James et al, 2013]. From the data available for the task it can be split into two main fields – supervised learning and unsupervised learning. There is also a mix of both – semi supervised learning. Furthermore supervised learning can be subdivided even further. In supervised learning we can deal with either a regression problem or a classification problem while in unsupervised learning it is harder to make such a split. Instead of this, rather a set of different methods for the problem can be considered.

4.3.1 Supervised and unsupervised learning

The difference between supervised and unsupervised learning is the presence or absence of a response variable. In the supervised learning problem there is a response for each observation of the predictors. In example, if we have the predictor variables x_i where i = 1, ..., n and measurements for them there is also an associated response y_i. The goal is to create a model that uses the predictor variables to make an as accurate as possible prediction for the response variable. This can be made for the sake of the prediction itself or to analyse the relationship between the predictor variables and the response variable.

The unsupervised learning problem is more challenging as the response variables are missing. Basically we are dealing with the problem of having x_i but not y_i. For instance, regression cannot be performed as there is no response to predict. The name unsupervised learning stems from the fact that there is no supervision from the response variable. One might ask the question of what can be gained from unsupervised learning as there is no response to predict. One answer would be analysis of how the variables or observations are related. As for supervised learning there exists a set of methods to take on the problem: clustering is one and principal components analysis is another but they will not be discussed as the problem of this report is one of supervised learning nature. [James et al, 2013]

Semi supervised learning is a blend of supervised and unsupervised learning. In this setting the

response measurements are only available for some of the observations. That is, we have x_i and y_i but for some values of i, y is unknown. This situation can occur if it is easy to get hold of the predictor variables but acquiring the response variable is more challenging. [James et al, 2013]

4.3.2 Regression and classification

Depending on the type of the output or response variable, the problem is either a regression problem or classification problem. Regression models predict a quantitative response while classification models predict a qualitative response. Examples of quantitative responses could be temperature, blood pressure or stock prices while examples of qualitative responses could be sentiment (negative, neutral, positive), a weather condition (sunny or not sunny, windy or not windy) or a nominal representation of the stock price (increase, decrease). [Hastie, Tibshirani & Friedman, 2009]

The goal of this thesis is to analyse whether the writings on forums influences the stock prices which can be done in different ways. One approach is the following: given predictor variables x_i, predict the y_i, where x_i is some representation of the forum texts and y_i is some measurement of the stock price, in example the volatility or the closing price for a day. Note that this could also be considered as a classification problem, depending on how the response variable is defined. Rather than looking at an exact increase in stock price, we could just say that the stock price has increased. In this case the response variable can take the qualitative forms increase, decrease and no change. The same goes for the volatility, either the value could be analysed, for regression, or whether the volatility is high or low, for classification.

There exists a plethora of models for classification and regression and a few of them will be discussed here.

4.3.3 Linear regression

Linear regression is a statistical method that has been around for a long time. Despite a simple nature they can provide good results and it is in general easy to understand the results. A linear regression model assumes that there is a linear relation between the predictor variables X and the response variable Y. Mathematically the linear regression model can be formulated as

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \qquad (4.3.1)$$

where β_0 is the bias (intercept) and β_j are weights to be calculated. The linear model assumes that the regression function is linear or that it can at least be well approximated by a linear function. To estimate the coefficients β_j a set of training data can be used, (x_1, y_1) ... (x_N, y_N). Every $x_i = (x_{i1}, x_{i2}, ..., x_{ip})^T$ is a vector containing the measurements for the features for the ith case. A common method for estimation is the least squares method which picks β_j to minimize the residual sum of squares (RSS - which will be discussed more in section 4.4.1)

$$RSS(\beta) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2. \quad (4.3.2)$$

Using the ordinary least squares method might yield very large negative or positive values for the coefficients as they may cancel each other out. A way to deal with this is using ridge regression which puts a penalty on the size of the coefficients. The ridge coefficients minimize a penalised sum of squares according to

$$\hat{\beta}^R = \underset{\beta}{argmin} \{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \} \qquad (4.3.3)$$

where λ is the shrinkage parameter. A high value of λ results in large shrinkage, the coefficients approach zero while $\lambda = 0$ is just the ordinary least squares method. Lambda can be equal to or greater than zero but not negative. [James et al, 2013]

When the regression coefficients have been calculated, the model can be used to predict new values for the response variable given the predictor variables.

4.3.4 Decision trees

Tree-based methods split the feature space into a number of regions, using a set of rules and then fit a model in each region, making a prediction using the mean or the mode of the training observations in the region to which it belongs. Decision trees can be used for both regression and classification and the method is simple and easy to interpret. [James et al, 2013] Figure 4.3.2 demonstrates a decision tree and its resulting feature space.

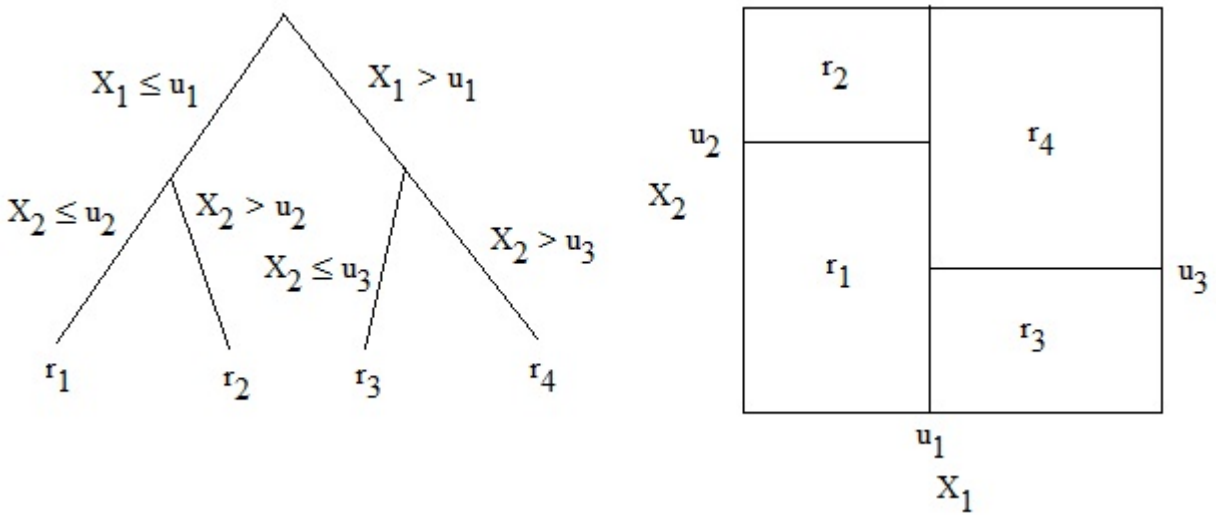

Figure 4.3.2

A decision tree and the feature space associated with it.

What happens is that first the feature space is split into two regions and the response is modelled by the mean of Y in each region. The variable and split point are chosen to get the best fit. After this one or both of the regions are split again and again until some criterion is reached. In figure 4.3.2 the first split has occurred at $X_1 = u_1$. After this the region $X_1 \le u_1$ is split at $X_2 = u_2$ and the region $X_1 > u_1$ is split at $X_1 = u_3$. The result is the right hand panel of figure 4.3.2. The feature space has been divided into four regions r_1, ..., r_4 and the objective is to find regions that minimize the RSS which is given by

$$\sum_{j=1}^{5} \sum_{j \in R_j} (y_i - \hat{y}_{R_j})^2. \quad (4.3.4)$$

Every observation that falls into a specific region is predicted the same which is the mean response for the training instances in that region. The regions are called terminal nodes or leaves and the points where the split occurs are called internal nodes.

If the problem regards classification the method differs a little but is in general similar. Rather than predicting each observation in the region from the mean response, each observation is predicted to belong to the majority class in that region. [James et al, 2013]

4.3.5 Random forests and bagging

One of the problems with decision trees is that they suffer from high variance. If a random split of the training data into two parts is made and a decision tree is fitted to them it could give very different results. Bagging (bootstrap aggregation) is a way to deal with the high variance. The idea behind bagging is to use bootstrapping: it takes repeated samples from the training data set, builds a prediction model for each set and then averages the predictions. Bagging can be used on many different methods for regression or classification. It can be used on regression trees by building n trees, using n bootstrapped training sets. Each individual tree will have high variance but low bias. If the n trees are averaged the variance will be reduced. [James et al, 2013]

Random forests is essentially the same method as bagging but there is a slight change to it. At each split in the tree, a random set of predictors are considered and only one of those predictors from the random set is allowed to be used in the split. The reasoning behind this is that there might be one very strong predictor. If that is the case, most of the generated trees will use this predictor for the top split and hence those trees will look very similar and the predictions from the trees will be strongly correlated. With random forests, this strong predictor will not be allowed in the sets of predictors where it is absent and hence the trees grown will look more different. This in turn results in a reduction of the variance. If the size for the random set of predictors is equal to number of predictors, it is the same thing as using bagging. [James et al, 2013]

4.3.6 Boosting trees

Boosting is another way to improve the performance of decision trees and like bagging it is not restricted to regression trees, it can be used for classification as well. Boosting works in a way similar to bagging which builds several trees from bootstrapped data. Boosting, however, does not rely on bootstrapping. Instead it grows the trees using information from previous trees and each tree is fit on a modified version of the original data.

Boosting is a slow learning method. With the present model, a decision tree is fit to the residuals of

the model rather than the response Y. The new decision tree is added to the fitted function and the residuals are updated. The trees can be quite small, only a few terminal nodes as determined by the parameter d. This fitting of small trees to the residuals improves the model in parts where it does not perform well. There is also a shrinkage parameter lambda which slows down the process further which lets even more trees with different shapes take on the residuals. There are three different parameters that can be tuned for the boosting; they are the number of trees B, the shrinkage parameter lambda and the number of splits in each tree, d, the interaction depth.

An example of a popular boosting algorithm is AdaBoost which was developed by Freund and Schapire in 1997. For classification we can consider a problem with two classes $Y \in \{A, B\}$. With a vector of predictor variables X, a classifier H(X) will make a prediction giving either A or B. If the classifier was weak its error rate would be only a little better than simply guessing. Weak classifiers are desired and normally the trees are designed to be weak. AdaBoost uses the weak algorithm in steps on modified versions of the data, yielding a sequence of weak classifiers, $H_m(X)$ where m = 1, 2, ..., M. Then all the predictions are combined and a final prediction is obtained from a majority vote from the predictions. At each step of the boosting there are some weights calculated by the boosting algorithm. Their purpose is to give more influence to the better classifiers. Also other weights $w_1, w_2, \ldots w_N$ are applied to the training observations (x_i, y_i) where i = 1, 2, ..., N. The weights w_i will be modified according to the success of the classifier. If observation i is misclassified the weight will be increased while it will be decreased if it was classified correctly. The result of this is that the observations that were hard to classify will get more influence during each iteration so that each successive classifier can focus on them. [Hastie, Tibshirani & Friedman, 2009] In figure 4.3.6 we can see how AdaBoost turns weak learners into strong ones.

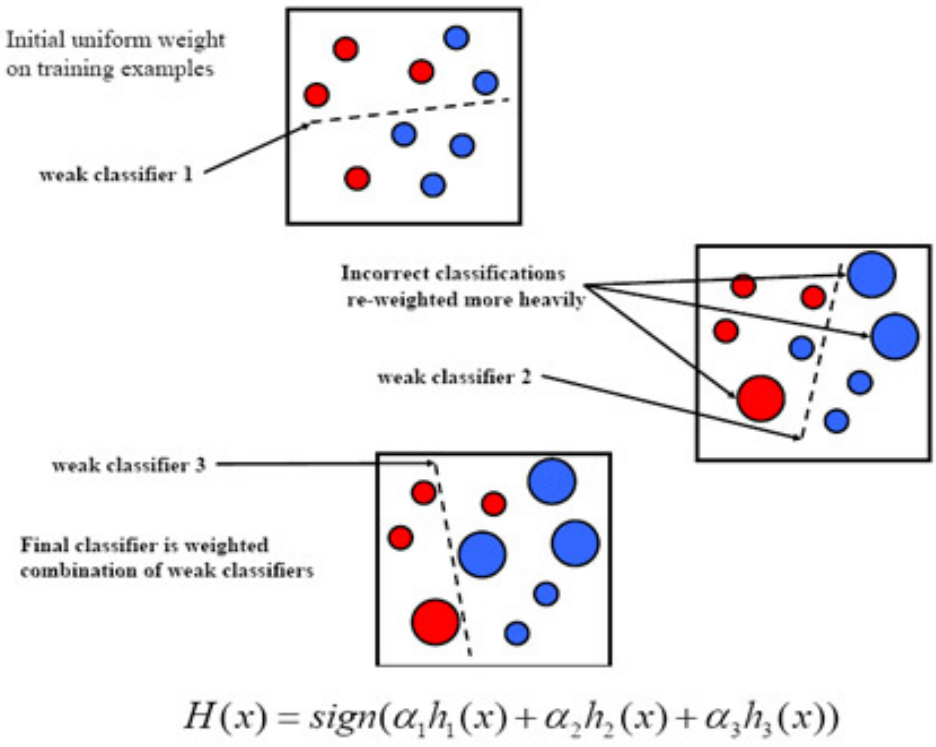

Figure 4.3.6. How AdaBoost works by applying weights on the weak learners. [Kim]

4.3.7 Naive Bayes

Naive Bayes is a simple, yet popular, model for classification. It is good when the feature space is high-dimensional. Naive Bayes has its basis in Bayes' Theorem which calculates a conditional probability in the following way:

$$p(C_k|x) = \frac{p(C_k) \cdot p(x|C_k)}{p(x)}. \quad (4.3.5)$$

The naiveness in the name comes from the fact that it makes the assumption that the features are conditionally independent. The Naive Bayes classifier calculates the probability of a feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ being in class C_k in the following way:

$$p(C_k|x) = p(C_k) \prod_{i=1}^{n} p(x_i|C_k). \quad (4.3.6)$$

The assumption of independent features is rather optimistic but still Naive Bayes tends to perform

well in some situations. For instance, popular application domains include spam filtering and document classification. [Hastie, Tibshirani & Friedman, 2009]

4.3.8 Support vector machine

Support vector machines can be used for both classification and regression but the discussion here will be limited to the classification setting. For the purpose of classification, support vector machine uses a hyperplane to separate the classes. In geometry a hyperplane is a subspace of one dimension less than the space that surrounds it – if the surrounding space is two dimensional, the hyperplane will be one dimensional for example. The hyperplane is chosen to maximise the margin between the different classes as much as possible. Many hyperplanes can perform well on training data but the generalization performance on test data can deteriorate and choosing the hyperplane which maximises the margin improves the training error. Figure 4.3.8.1 shows the hyperplane which has the maximum margin between the classes so that the decision boundary separates the classes as much as possible.

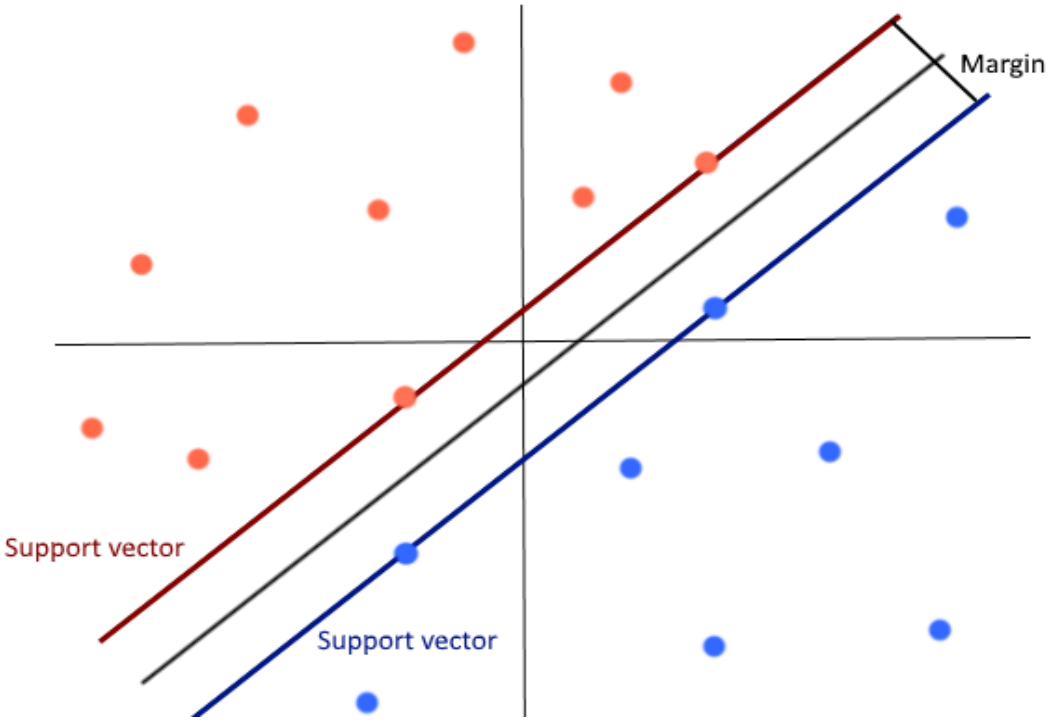

Figure 4.3.8.1. Hyperplane (the middle line) with the maximum margin.

Sometimes the classes are not lineary separable in the original space. The support vector machine deals with this by mapping the feature vector into a high dimensional space where the classes might be linearly separable so it can use linear classifiers. [Harman & Kulkarni, 2011] T.M. Cover stated famously in 1965:

A complex pattern-classification problem, cast in a high-dimensional space nonlinearly, is more likely to be linearly separable than in a low-dimensional space, provided that the space is not densely populated.

[Cover, 1965] This statement is now known as Cover's theorem. In figure 4.3.8.2 we can see how the features which are not originally linearly separated can be so in a higher dimension.

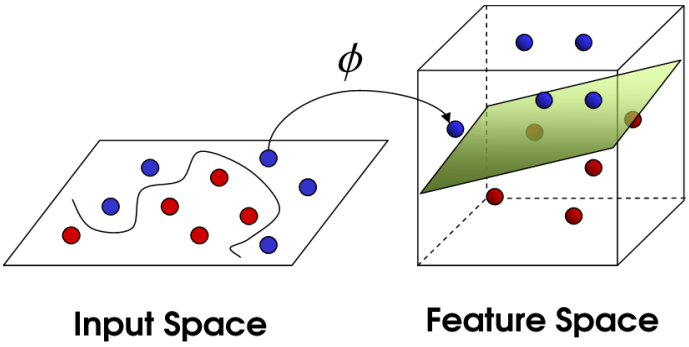

Figure 4.3.8.2 Mapping of the feature space to higher dimension.
[Spencer, 2015]

Projection into a higher space is often computationally demanding, however, and fortunately there is a way around this. Classifying a feature vector involves the use of dot products and by replacing the dot products by a kernel function, which is easier to compute, the projection into higher space can be avoided. [Harman & Kulkarni, 2011] This procedure is known as the kernel trick.

The kernel function is one of the parameters in the support vector machine. Another parameter that can be tuned is the complexity parameter – c. The c parameter relates to the margin of the hyperplane, a larger value of c means that a hyperplane with a smaller margin will be chosen and conversely, a small value of c results in a hyperplane with larger margin.

4.4 Model evaluation

Model evaluation is an important part of the problem. We want to assess how well a model performs in order to choose the best model for the given problem. Here sometimes a trade-off has to be made though since there are different aspects to consider. It is possible that one model performs better than another model but has much lower interpretability. In that case we might want to choose the model that performs worse (if it is within a tolerable rate) if we in addition to the prediction accuracy also are interested in how the response is connected to the predictors. If the performance

of the better model is statistically significantly better than that of the worse model, the better model should be chosen.

4.4.1 Error estimation

When it comes to regression there are several error figures to look at. Some of these are the residuals, residual sum of squares (RSS), the mean squared error (MSE), the root mean squared error (RMSE) and the absolute error. Furthermore we can also talk about the training error, the validation error and the test error. A residual is the difference between the i:th observed response value and the corresponding predicted value,

$$e_i = y_i - \hat{y}_i \quad (4.4.1)$$

and the residual sum of squares is the sum of the squared residuals

$$RSS = e_1^2 + e_2^2 + ... + e_n^2. \quad (4.4.2)$$

The mean squared error is defined by the mean of the square of the residuals, expressed as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (4.4.3)$$

and is one of the most commonly used measurements of the model performance when it comes to regression. The absolute error, ϵ, is given by the absolute value of the difference between the observation and prediction such as

$$\epsilon = |y_i - \hat{y}_i|. \quad (4.4.4)$$

For classification there are other error figures to look at. Among these are accuracy, precision, recall and F1-score. Accuracy is the percentage of correct classifications from the total number of classifications. A common way to visualize the results in classification is through a confusion matrix which is a table that shows the performance of the classifier. Table 4.4.1 underneath is an example of a confusion matrix.

Table 4.4.1: Confusion matrix

Actual class	Predicted class	
	Up	Down
Up	30	20
Down	10	40

In the example confusion matrix there are 50 instances of up and 50 instances of down. 30 times up was correctly classified as up and 20 times it was incorrectly classified as down. For down, it was correctly classified 40 times and incorrectly classified 10 times. Going back to the error measurements, precision is defined as

$$precision = \frac{t_p}{t_p + f_p} \quad (4.4.5)$$

where t_p is true positive and f_p false positive. In this case the precision for up would be 30/(30 + 10) which is 0.75. The true positives in this case would be the ups that are classified as up and the false positives are downs that have been classifed as up. Recall is given by

$$recall = \frac{t_p}{t_p + f_n} \quad (4.4.6)$$

with f_n being the rate of false negatives and t_p true positives again. The recall for up would then be 30/(30+20) which is 0.6. The false negatives in this case are the ups that have been classified as down. The difference from precision is the inclusion of false negatives rather than false positives. Thus precision tells us about the performance with respect to false positives while recall tells us about it with respect to false negatives. Finally there is also the F1-score which is a weighted mean of precision and recall. It is defined as

$$F1 = \frac{2t_p}{2t_p + f_p + f_n}. \quad (4.4.7)$$

4.4.2 Cross-validation

When evaluating the model, typically a large set of data is required. First the model needs a set of data to train on, the training set. Then the model needs a set of data for validation, the validation set, used to find the best model fitted to the training set. Lastly it also needs a test set, a set of data that the model has not seen before so we can get an estimate of how well the model performs on unbiased data. In the validation set, the best model is chosen and then this model is used on the test set, to see how well it performs on previously unseen data.

If the available data set is big, simply splitting it into three equally sized parts is good but commonly the available data is not enough so it is split into around 50 % training data and then 25 % each for validation and testing [Hastie, Tibshirani & Friedman, 2009]. A way to choose the size of the training set is by looking at the learning curve which shows the performance of the model as a function of the training sample size. The learning curve can depend on which classifier is being used and how much the classes are separated.

The training error is the error from the model on the training set, validation error is the error on the validation set and the test error is the error on the test set. Typically it is interesting to look at the testing error as it tells us how the model performs on data it has not seen while being built.

If we do not have that much data to use, we might end up with too few training examples which can affect the model negatively. One common way to get around this is by using cross-validation (CV). Cross validation effectively removes the need for a validation set by parting the data into k different folds (if k-fold CV is used) and then training and testing on these folds. The way this works is by splitting the data into k different folds, training it on k-1 folds and then testing on the remaining fold. The results are combined and used to estimate the prediction error. [Hastie, Tibshirani & Friedman, 2009] Figure 4.4.2 shows the data set being divided into five parts (or folds) for 5-fold cross-validation.

1	2	3	4	5
Train	Validation	Train	Train	Train

Figure 4.4.2. 5-fold cross-validation

4.4.3 Overfitting

A potential problem in machine learning and statistical learning is overfitting. Overfitting occurs when the model adapts too much to the training data. If the training data contains a lot of noise, the model will adapt to this noise which causes it to perform poorly on new data. The model can perform perfectly on the training data because it has learned from that data but new data is likely to be different and so the model will not be able to deal with it as efficiently.

4.5 Causality

The statistical methods presented can find correlations between the data and the stock prices which can be useful for predictions. They can not, however, tell us if there is any causal relationship between the forum texts and the price. It might be possile to predict how the price changes very well, using the text data if it correlates well but this is no guarantee that the writings have actually affected the price. There is a famous phrase “Correlation does not imply causation.” which reflects this.

An example of how things could work is if a user writes a text: A and the stock price B goes up. If this happens a few times A and B can correlate well and it seems like A is the reason B increases but there might have been another cause for B going up. The cause for B going up could be a press statement released by the company or a big order they made. If that is the case, A and B correlates but the relationship is not causal. It might in fact be the other way around, if B has been increasing, the writer is more inclined to write A.

5 Method

This part describes the method used for the project. First an introduction to the software will be given, mentioning a bit about Java, R and Weka and then the processing of the text will be covered. This is followed by a bit about the features and lastly the computations.

Making the analysis, it is possible to use either the text as it is or making a sentiment analysis for each post, aggregating the sentiment for one day and use it as a feature rather than the text itself. To make a sentiment analysis one requires a lot of labeled data and labeling the data oneself is a lot of work and it might be hard to catch the sentiment of the individual posts. For this reason, using the whole text for each day to see if it has any predictive power is more convenient. In this way, using the feature selection methods, one can also find out what words are useful for predicting the stock price, that is, words that could be influential for people's buy and sell behaviour.

5.1 Technology/software

First Java was used to organise and process the text so it could be used by the programs for machine learning. Java is a general purpose, concurrent, class based, object-oriented programming language. [Gosling et al, 2015] It might not be the best choice when it comes to machine learning tasks but it was chosen because it is the language that is used at Scila. So first the data was organised in Java so it could be processed by Weka, which is a collection of machine learning algorithms for data mining tasks [Weka, 2016]. When the text had been processed in Java and the words had been ranked by their TF-IDF score, the data was imported to Weka. In Weka further feature selection was performed using information gain and then the statistical models were applied to the data. Besides Java and Weka, also R was used. R is a programming language and environment for statistical computing and graphics. [The R Foundation]

5.2 Processing the data

The raw text data itself is hard to use for the statistical models. To improve the performance of the models it is a good idea to process the text before extracting the features. The data was processed by removing stop words, URLs and certain non-word characters. After this the texts were stemmed by the Snowball stemmer and then the text was ready to use. The TF-IDF-score was calculated for all words and then used as weighting for the computations. Furthermore, since the stock market closes

at 17.30, forum posts that were posted 17.30 or later were sorted out since they can not possibly influence the price for that day. They might influence the next day but since the posts are more likely to influence the price in a short time frame it was decided to not include them at all and use midnight as a delimiter for when the posts can affect the price.

5.3 Features

A set of nonword characters such as !, ?, ':)', + and – were used in addition to the from occurring in the texts as well as usernames. Besides this also the number of posts during a day related to the stock being analysed was used as a feature.

Using special characters such as exclamation marks and emoticons like ':)' might better capture the sentiment of a post. For instance, if a person uses a lot of exclamation marks in a post it might reflect a stronger sentiment which could have a bigger impact on people's buy or sell behaviour in addition to more informative words in a post.

For the text features the TFIDF score was calculated for all words, nonword characters and combinations of nonword characters, like ':)', in all posts. Then the highest ranked ones were chosen as a subset of text features. After this information gain was used as an additional feature selection method to find the features among the chosen ones that had the largest predictive power.

All the text features were represented as unigrams.

5.4 Computations

To test for the forum-stock-price relation the computations were done in some different ways. Classifiers were used to see if they could predict whether the price would have increased or decreased over a day as well as if the volatility was high or low. For the price difference it was easy to get the response variable, simply subtract the opening price from the closing price to see whether the result was positive or negative. When it came to measuring whether the volatility was high or low it was less straightforward. To do this, the swing was calculated for all days for which there was text data available. Then either the median or the average swing was used, depending on which gave a more even distribution of high's and low's in order to reduce the class imbalance. Then the volatility was set to either high or low for each day.

Originally the plan was to use regression to determine the end price of the day or the swing for the day but it was decided that predicting an exact number would probably be too hard. For the regression, R was used but after it was decided to cut that out, R was only used to generate the figure with the volatile and non-volatile stock.

Furthermore the accuracy for only guessing up or down for each company over the time period analysed was evaluated. This was to test whether it would be better to just guess that the stock price would go only up or only down than using the models. Also a test with mixed up stock and text data was performed but this will be covered more in section 6.4.

6 Results

In this section the results of the classification methods will be presented. They were all obtained using 10-fold cross-validation. First is however some of the results from the information gain feature selection. After this follows the performance of the best classifiers for the text data from Avanza, for both price difference and volatility. This is followed by the best classifiers for the Flashback text data, again for both price difference and volatility.

The performance of the different classifiers differed a bit depending on which stock was analysed and from which forum the data was taken as well as whether the volatility or price difference was examined.

6.1 Information gain features

Using the information gain feature selection method, the results varied a lot. For the price difference estimations it generally ended up selecting only around 10-40 features. When it came to the volatility, however, it could end up selecting around 10-1400 features so there was a large discrepancy. For the Flashback forum data, which was more sparse than the Avanza forum data, the number of features selected tended to be lower. Moreover, some of the features selected by information gain were just combinations of special characters, symbolising more advanced emoticons or some kind of figure (or the part of a figure).

List of some features that are likely to have an impact. (Translated from Swedish)

Increase
Decrease
Down
Up
Buy
Sell
Result
Development
Nervous
Panic

Worry

Hausse

Long-term

List of some features that are not likely to have an impact.

Writing skills (is one word in Swedish)

'(('

Student cruise (one word in Swedish)

'))'

'('

6.2 Avanza text data

These are the results obtained by using the data from Avanza. They will be presented for each company, both the performance of the volatility and price difference computations.

6.2.1 Price difference

Fingerprint Cards

The best classifier for Fingerprint cards when it came to the price difference was AdaBoost with 1000 iterations. This gave an accuracy of 54.5 %.

```
Incorrect: 360 | Correct: 431
Total accuracy: 0.5449
up  down
210 192  |  up
168 221  |  down
```

Class	Recall	Precision	F1
Up	0.52	0.56	0.54
Down	0.57	0.54	0.55

Precise Biometrics

For Precise Biometrics the support vector machine had the best result when it came to the price difference. The accuracy was 57.1 % with a linear kernel and a complexity parameter of 300.

```
Incorrect: 358 | Correct: 478
Accuracy: 0.5717703349282297
Up  Down
112 266  |  Up
 92  366  |  Down
```

Class	Recall	Precision	F1
Up	0.3	0.55	0.38
Down	0.8	0.58	0.67

Anoto Group

For Anoto Group the best classifier was AdaBoost when it came to price difference. The accuracy was 59.4 % with 1000 iterations.

```
Incorrect: 271 |  Correct: 396
Accuracy: 0.5937
Up  Down
277  93   |  Up
178 119  |  Down
```

Class	Recall	Precision	F1
Up	0.75	0.61	0.67
Down	0.4	0.56	0.47

Karo Pharma

For Karo Pharma, AdaBoost gave the best results with an accuracy of 53.0 % when using 10 iterations.

```
Incorrect: 381 | Correct: 430
Accuracy: 0.5302
Up  Down
262 174  |  Up
207 168  |  Down
```

Class	Recall	Precision	F1
Up	0.6	0.56	0.58
Down	0.45	0.49	0.47

Sensys Gatso Group

For Sensys Gatso Group the best classifier was AdaBoost with 100 iterations which yielded an accuracy of 58.3 %.

```
Incorrect: 273 | Correct: 382
Accuracy: 0.5832
Up  Down
252 125  |  Up
148 130  |  Down
```

Class	Recall	Precision	F1
Up	0.67	0.63	0.65
Down	0.47	0.51	0.49

6.2.2 Volatility

Fingerprint Cards

The accuracy for Fingerprint Cards was 85.2 % with random forest when classifying the volatility with Avanza text data. This was achieved with 100 trees and depth 10.

```
Incorrect: 117 | Correct: 674
Accuracy: 0.8520
High  Low
160  23  |  High
94  514  |  Low
```

Class	Recall	Precision	F1
High	0.87	0.63	0.73
Low	0.85	0.96	0.9

Precise Biometrics

For Precise Biometrics and volatility, random forest had the best accuracy. 80.5 % was obtained with 250 trees and depth 10.

```
Incorrect: 163 | Correct: 673
Accuracy: 0.8050
High  Low
93 147  |  High
16 580  |  Low
```

Class	Recall	Precision	F1
High	0.39	0.85	0.53
Low	0.97	0.8	0.88

Anoto Group

For Anoto Group the best result came from random forest. The accuracy was then 74.8 % with a depth of 10 and 100 grown trees.

Incorrect: 168 | Correct: 499
Accuracy: 0.7481
High Low
89 135 | High
33 410 | Low

Class	Recall	Precision	F1
High	0.39	0.85	0.53
Low	0.93	0.75	0.88

Karo Pharma

When measuring the volatility, again random forest gave the best accuracy, 79.0 % with 250 trees and depth 5.

Incorrect: 170 | Correct: 641
Accuracy: 0.7904
High Low
145 130 | High
40 496 | Low

Class	Recall	Precision	F1
High	0.53	0.78	0.63
Low	0.94	0.79	0.85

Sensys Gatso Group

Best performing on the volatility for Sensys Gatso Group was random forest with an accuracy of 84.7 %. This result was achieved with 100 trees and depth 10.

Incorrect: 100 | Correct: 555
Accuracy: 0.8473
High Low
85 94 | High
6 470 | Low

Class	Recall	Precision	F1
High	0.47	0.93	0.63
Low	0.99	0.83	0.9

6.3 Flashback text data

6.3.1 Price difference

Fingerprint Cards

On Flashback AdaBoost was the best classifier for Fingerprint Cards when it came to price difference. It had 59.5 % accuracy with 10 iterations.

```
Incorrect: 137 | Correct:  201
Accuracy: 0.5947
Up  Down
167  23  |  Up
114  34  |  Down
```

Class	Recall	Precision	F1
Up	0.88	0.59	0.71
Down	0.23	0.6	0.33

Precise Biometrics

For Precise Biometrics random forest had the best prediction accuracy, 64.7 % with 100 trees and depth 5.

```
Incorrect: 41 | Correct:   75
Accuracy: 0.6466
Up  Down
45 18  |  Up
23 30  |  Down
```

Class	Recall	Precision	F1
Up	0.71	0.66	0.69
Down	0.57	0.63	0.59

Anoto Group

The best classifier for Anoto Group on the Flashback data was AdaBoost which had an accuracy of 51.8 % obtained with 10 iterations.

```
Incorrect: 93 | Correct: 100
Accuracy: 0.5181
Up  Down
53 44  |  Up
49 47  |  Down
```

Class	Recall	Precision	F1
Up	0.55	0.52	0.53
Down	0.49	0.52	0.5

Karo Pharma

For Karo Pharma AdaBoost had the best accuracy. It was 58.7 % with 10 iterations.

```
Incorrect: 31 | Correct: 44
Accuracy: 0.5867
Up  Down
27   12  |  Up
19   17  |  Down
```

Class	Recall	Precision	F1
Up	0.69	0.59	0.64
Down	0.47	0.59	0.52

Sensys Gatso Group

For Sensys Gatso Group support vector machine was the best classifier when using a linear kernel with c = 100. It then achieved an accuracy of 69.2 %.

```
Incorrect: 16 | Correct: 36
Accuracy: 0.6923
Up  Down
28    2 |  Up
14    8 |  Down
```

Class	Recall	Precision	F1
Up	0.93	0.67	0.78
Down	0.36	0.8	0.5

6.3.2 Volatility

Fingerprint Cards

For Fingerprint on Flashback, the best classifier was random forest with 72.8 % accuracy. The number of trees grown was 100 and the depth 10.

```
Incorrect: 92 | Correct: 246
Accuracy: 0.7278
High  Low
24 88  |  High
4 222  |  Low
```

Class	Recall	Precision	F1
High	0.21	0.86	0.34
Low	0.98	0.72	0.83

Precise Biometrics

For the Flashback data AdaBoost had the best performance. The accuracy was 69.8 % with 10 iterations.

Incorrect: 35 | Correct: 81
Accuracy: 0.6983
High Low
3 33 | High
2 78 | Low

Class	Recall	Precision	F1
High	0.08	0.6	0.15
Low	0.98	0.7	0.82

Anoto Group

For the Anoto Flashback data, the best classifier for volatility was the support vector machine. It had an accuracy of 67.3 %. This was achieved with a Gaussian kernel and complexity parameter 50.

Incorrect: 63 | Correct: 130
Accuracy: 0.6736
High Low
15 41 | High
22 115 | Low

Class	Recall	Precision	F1
High	0.27	0.41	0.32
Low	0.16	0.74	0.78

Karo Pharma

Incorrect: 30 | Correct: 45
Accuracy: 0.6
High Low
6 22 | High
8 39 | Low

Class	Recall	Precision	F1
High	0.21	0.43	0.29
Low	0.83	0.64	0.72

The best performing classifiers was support vector machine. It scored 60 % accuracy with c as 50 and a linear kernel.

Sensys Gatso Group

The best accuracy came from support vector machine with 59.6 % accuracy with complexity parameter 50.

Incorrect: 21 | Correct: 31
Accuracy: 0.5961
High Low
23 4 | High
17 8 | Low

Class	Recall	Precision	F1
High	0.85	0.58	0.69
Low	0.32	0.67	0.43

6.4 Testing

The models have performed well, with a high accuracy in general (at least in the area of predicting volatility). This tells us that there is a good correlation between the stock prices and the texts but it cannot tell us if there is a relationship between the texts and the price. A test was performed to see whether the accuracy would decrease when predicting the prices for one company, using text data from another company. The models were trained, validated and tested with textdata $\mathbf{X_1}$ belonging to company 1 and $\mathbf{Y_2}$ as stock data belonging to company 2. If the accuracy would drop to around 50 % which would be equivalent to random guessing, it could serve as an implication that the text data was actually helpful for predicting the price of the company it is associated with only. This could also imply some causal relationship.

Using the text data from one company to predict the volatility for another company actually yielded a similar result. When using random forest to predict volatility for Anoto Group by using text data for Karo Pharma from Avanza, an accuracy of 67.8 % was achieved. Performing the same kind of test on Anoto Group price data with Fingerprint Cards text data from Flashback yielded an accuracy of 65.0 % with random forest. With Karo Pharma price data and Sensys Gatso Group text data from Flashback, 71.0 % accuracy was obtained with AdaBoost. 73.7 % accuracy was achieved when predicting Precise Biometrics volatility with the help of Fingerprint Cards text data from Avanza and AdaBoost classifier.

Doing the same thing for the price difference, predicting Fingerprint Cards price with Karo text data from Avanza yielded an accuracy of 53.7 %. With price data from Precise Biometrics and text data for Anoto Group from Flashback the accuray was 58.3 %.

6.5 Consistently guessing up or down for a whole period

6.5.1 Avanza

For Fingerprint Cards, the distribution was 383 times up and 367 times down. Up was the majority and and just guessing up over the whole series would yield an accuracy of 51.1 %. For Precise Biometrics the numbers were 378 up counts and 458 down counts. Sticking to down only would result in 54.8 % accuracy. Anoto Group had 370 ups and 297 downs so only guessing up would give 55.4 % accuracy. For Karo Pharma there were 436 up counts and 375 down counts and consistently guessing up would yield 53.8 % accuracy. Finally, Sensys Gatso Group saw 377 ups and 278 downs over their time series so sticking to up would give an accuracy of 57.6 %.

6.5.2 Flashback

the distribution was 190 times up and 148 times down. Up was the majority and and just guessing up over the whole series would yield an accuracy of 56 %. For Precise Biometrics the numbers were 63 up counts and 53 down counts. Sticking to up only would result in 54.3 % accuracy. Anoto Group had 97 ups and 96 downs so only guessing up would give 50.2 % accuracy. For Karo Pharma there were 39 up counts and 36 down counts and consistently guessing up would yield 52 % accuracy. Finally, Sensys Gatso Group saw 30 ups and 22 downs over their time series so sticking to up would give an accuracy of 57.7 %.

6.6 Summary of results

6.6.1 Price difference

When measuring the price difference, AdaBoost had the best overall accuracy. Seven times it was the best classifier. Second came support vector machine which was the best classifier twice. Random forest was third, being the best classifier once and Naive Bayes never had the highest accuracy. The results are summarised in table 6.6.1

When using the Flashback data the highest accuracy achieved was 69.2 % when using the support vector machine classifier. This was for Sensys Gatso Group. The lowest accuracy was 51.8 %, for Anoto Group, using AdaBoost as classifier. For the Avanza data the best accuracy was 59.4 % for Anoto Group, using AdaBoost. The lowest accuracy for text data from Avanza was 53 % and came from Karo Pharma when using AdaBoost. The average accuracy for all companies and both forums

when it comes to the price difference was 58.6 %.

Table 6.6.1: Ranking of best classifiers for price difference

Classifier	# of best performances	Avanza data	Flashback data
AdaBoost	7	4	3
Support vector machine	2	1	1
Random forest	1	-	1
Naive Bayes	-	-	-

6.6.2 Volatility

When measuring the volatility random forest was in general the best classifier. It came out best six of the times. On the second place was support vector machine which was best three of the times, followed by AdaBoost which had the highest performance once. Also this time Naive Bayes never had the best performance. These results are presented in table 6.6.2

Table 6.6.2: Ranking of best classifiers for volatility

Classifier	# of best performances	Avanza data	Flashback data
Random forest	6	5	1
Support vector machine	3	0	3
AdaBoost	1	-	1
Naive Bayes	-	-	-

The highest accuracy for the volatility came from random forest on Fingerprint Cards with the Avanza forum data. The accuracy in this case was 85.2 %. The best performance on the Flashback data came from random forest again. Also this time on Karo Pharma with 72.8 % accuracy. The lowest accuracy on the Flashback data was 59.6 %, scored with support vector machine on Sensys Gatso Group. For Avanza it was 74.8 % by random forest on Anoto Group. The average accuracy for all companies on both forums was 73.4 % when it came to volatility.

6.6.3 Forums

Following is also two tables showing the results for the different forums. In the first table, table 6.6.3, the results for the Avanza text data is presented.

Table 6.6.3: Results for Avanza data

Company	Accuracy [%] (direction)	Accuracy [%] (volatility)	Classifier (direction)	Classifier (volatility)
Anoto	59.4	74.8	AdaBoost	Random Forest
Fingerprint	54.5	85.2	AdaBoost	Random Forest
Karo	53	79	AdaBoost	Random Forest
Precise	57.1	80.5	SVM	Random Forest
Sensys	58.4	84.7	AdaBoost	Random Forest

In the next table, table 6.6.4, the results for the text data from the Flashback forums is shown.

Table 6.6.4: Results for Flashback data

Company	Accuracy [%] (direction)	Accuracy [%] (volatility)	Classifier (direction)	Classifier (volatility)
Anoto	51.8	67.3	AdaBoost	SVM
Fingerprint	59.5	72.8	AdaBoost	Random Forest
Karo	58.7	60	AdaBoost	SVM
Precise	64.7	69.8	Random Forest	AdaBoost
Sensys	69.2	59.6	SVM	SVM

7 Discussions and conclusions

7.1 Discussions

For predicting the price difference with the models, the accuracy ranged from around 50 % to almost 70 %. On average the accuracy was 58.6 % which is comparable to the results by Schumaker and Chen who achieved 58.17 % although they used financial news rather than text from forums for predictions. 58.6 % is also better than just guessing up or down. However, the only two times it made it above 60 % was for the Flashback data, which had very few instances.

When it came to predicting the volatility, the accuracy was better. The range was around 60 % to 85 % and the average accuracy was 73.4 %. For the volatility however, the data suffered from class imbalance due to the troubles with deciding when the volatility was high and when it was low. This was due to the fact that most of the time it was neither high nor low, rather somewhere in between. This imbalance is likely to improve the accuracy.

Due to the very low number of features selected with information gain and the dubious nature of some of those features, it is likely that they do not have anything to do with how the stock price changes, they simply correlate well. With such small feature sets, it is likely not enough information to have an impact on the stock price. Some of the words might be meaningful for real, but if only around ten words are enough to make an accurate prediction, it is likely that it is just a good correlation since those ten words alone probably would not have enough influence on the price. Some of the features such as '((' and similar have been observed to be used for drawing figures, such as a rocket, often accompanied by texts about the stock price sky rocketing. In that case those figures could have been made a lot during days when the price has increased and hence it has been useful for predicting the price difference but those figures themselves will not have so much to do with the increase of the price, again it is just a good correlation. Although it is possible that more sentimental buyers could feel more assured to buy the stocks.

Another possible explanation for those features is that a specific user uses them a lot and that user might write things that could be influential on the price. In that case, whenever that user writes something that causes a change in the price, he or she also uses those figures and hence it will be considered influential.

The statistical models do not tell us anything about causality so the test with mixed up data was performed to see if it would have any impact on the accuracy. Due to the fact that it was possible to predict the volatility and price difference for the stock of a company, using text data from another company, which yielded similar accuracy, it is most likely that the correlation the models find does not signify any causal relationship. The models can predict the volatility or price difference of stocks using text data from the forums, but it is possible that there is no causal relationship. It might as well be possible to achieve the same accuracy using data from a completely different forum concerning totally different topics, such as a gardening forum in example. If using text data from a gardening forum would give similar results, we could probably reject the idea that the stock forum text data gives us anything meaningful for the sake of price prediction, good correlations can be found regardless of the source and it is not possible to assess the impact of the writings on the stock data using this method.

Since the test was performed with stock forum texts, it might be, that the market movements for the companies tested are correlated. The stocks of those companies might have had similar movements which could explain why it was possible to predict the volatility and price difference with such accuracy. Performing more tests with more companies stock data and especially companies from very different fields could clarify this. It could also be that the language is similar in the different threads on the forums so it is possible to find the words that correlate well when using different companies' text data.

There is also the possibility that the the posts on the forums do not affect the stock price but rather the other way around, the price has a larger impact on people's writings. When the price goes up, people tend to write positive things and when the price goes down, people write less positive thing. This brings us back to the rocket example: the price is increasing and someone makes a rocket figure in the text so it seems like the rocket figure is good at predicting that the price will go up when it is actually the opposite, the price went up so someone made a rocket figure.

If there would be a causal relationship between the forum text data and the stock price, it seems that the volatility is more influenced by the discussion boards since the accuracy for predicting it is higher.

An important thing to consider is the small amount of response variables. As the price was only

examined on a daily basis, there were not so many observations available. Typically for these statistical models, many more observations are required. If the price could be evaluated several times during one day, the models would likely be more trustworthy. Also, when an impactful post is written on a popular forum, it is likely that people will act on the information as soon as possible. Hence it would be easier to relate a specific post to the change in price if the price was measured soon after the post was written rather than at the end of the day. Once again back to the rocket example: the price increases a lot in the first half of the day so someone makes a rocket figure on the forum. At the end of the day the price has increased and it seems like the rocket figure has influenced the price but in reality the price begun increasing before the rocket figure was posted.

Another way to make the models could be to manually select a number of words to be used for the analysis. Using financial experts to create a list of words that are likely to be influential on the stock prices it would be possible to simply extract those words from the texts and count their occurence using their counts as a feature. A similar possibility would be to expand the stopword list to remove other words that are likely to not play too big of a role.

Using the predictions for the sake of investment would be hard. Since the price will probably change within a short time frame after something been written, getting hold of the post and making the prediction must be done soon in order to act on the information. The work here is limited to making only one prediction per day, so one would have to wait until the end of the day and gather all the data before predicting whether the price will go up or down. Then the methods are still limited to comparing the closing price with the opening price, and the price at the moment before the market closes will most likely differ from the price as the market opens, rendering the prediction of the price change with respect to the opening price useless.

7.2 Conclusions

Several different models have been used to predict the the price difference over a day and the volatility for a day using text data from two different discussion forums. For predicting the price difference, the highest accuracy was 59.4 % for the Avanza text data and 69.2 % was achieved for the Flashback text data. For predicting the volatility, the highest accuracy was 85.2 % for the Avanza text data and 72.8 % for the Flashback text data. The average accuracy for price difference was 58.6 % and 73.4 % for volatility. AdaBoost was the best classifier most of the times for predicting the price difference and random forest for predicting the volatility.

The statistical methods were better than the method of just guessing that the price would go up and sticking to that, or down and sticking to that over a time period. That method yielded at best an accuracy of 57.7 % whereas the statistical model for that company gave 69.2 % accuracy. Furthermore it is also lower than the average accuracy for predicting the price difference with the statistical methods.

Since the classifiers perform well even when using the wrong text data, one has to look at what features are selected with information gain in order to get an idea of whether there might be a causal relationship or not. If the features seem reasonable for influencing the price, there might be a causal relationship. The classifiers can probably perform well even with the wrong data since the there are similarities between the texts among the different companies. It should be noted though, that mostly the performance was better when using the right text data for the right company.

The statistical models can adapt to the stock forum text data that is used as input and use it for prediction on the price data with good results. They will find some correlation between the response variables and the features regardless of whether the text data is associated with the company for which the price is predicted or not. Hence, using the texts only is too crude of a method to analyse if there is a relationship or not. Since the stock forum text data is quite similar, the models can generalise well so using a different kind of text data could be a better way of testing the models on random data.

These methods, using the forum texts as predictors have been useful for predicting the stock price but the models cannot tell us whether the posts did influence the price or not.

7.3 Future work

For future works a sentiment analysis on all the posts could be performed as well to test whether the sentiment as a feature would have any effect on the results. It could also be possible to make analyses on other companies, also international companies, using other message boards in other languages. A possible good source would be Twitter, as there is a lot of data available there.

Furthermore, if more fine grained stock price data would be available, it would be helpful to split each trading day into several smaller intervals as that would increase the number of training

examples and since the changes in stock price due to discussions on forums are more likely to happen in shorter time frames. With more fine grained price data it might also be possible to continuously build models and predict stock prices (if both text data and stock data is gathered continuously), possibly making the methods useful for investment.

Another interesting thing to do would be using text data from a completely different forum that has nothing to do with stocks or companies. Some candidates for this could be gardening forums, movie forums or family forums (although there are of course many other possibilities). For the stock prices, this would basically be random data to be used as predictor variables. The tests could also be performed with stock data from other companies doing business in more unrelated fields.

Lastly, defining the volatility as high or low in a more sophisticated way that balances the classes might give more trustworthy results for the volatility predictions. Furthermore it might also be possible to separate it into three classes, high, low and medium.

References

[Bolón-Canedo et al, 2015] Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A. (2015). Feature selection for high-dimensional data. New York: Springer

[Cover, 1965] Cover, T.M. (1965). Geometrical and Statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers,* vol. EC-14, p. 326-344

[Fisher, 1936] Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, vol. 7, p. 179-188

[Google Finance, 2016] PepsiCo https://www.google.com/finance?q=pepsico&ei=EQpaV9H8L9jrsgHGkYvYCw , [2016-05-15]

[Gosling et al, 2015] Gosling, J., Joy, B., Steele, G., Bracha, G., Buckley, A. (2015). The Java Language Specification. Redwood City: Oracle America

[The Guardian, 2015] Scottish man used Twitter to launch $1.6m stock market scam, says US jury. *The Guardian*, 2015-11-06 https://www.theguardian.com/technology/2015/nov/06/scottish-man-used-twitter-as-part-of-failed-stock-market-scam-says-us-jury [2016-05-02]

[Harman & Kulkarni, 2011] Kulkarni, S., Harman, G. (2011). An Elementary Introduction to Statistical Learning Theory. Hoboken: Wiley

[Harper] Harper, D.. Forces That Move Stock Prices http://www.investopedia.com/articles/basics/04/100804.asp , [2016-05-17]

[Hastie, Tibshirani & Friedman, 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning. 2nd edition. New York: Springer

[Investopedia] Stocks Basics: What Causes Stock Prices To Change? http://www.investopedia.com/university/stocks/stocks4.asp , [2016-05-17]

[James et al, 2013] James, G., Witten, D., Hastie, T., Tibshirani, R., (2013). An introduction to statistical learning. New York: Springer

[Kao & Poteet, 2007] Kao, A., Poteet, S. R. (2007). Natural Language Processing and Text Mining. London: Springer

[Kim] Kim, K.. Viola-Jones and Morphology-based Face Detector. http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject_KH.htm, [2016-05-20]

[Leskovec, Rajaraman & Ullman, 2014] Leskovec, J., Rajaraman, A., Ullman, J. D. (2014). Mining of massive datasets. 2nd edition. Cambridge: Cambridge University Press

[Metzler, 2008] Metzler, D., (2008). Generalized inverse document frequency. *ACM 17th Conference on Information and Knowledge Management* (CIKM). p. 399-408. Napa Valley, California, USA October 26–30, 2008

[Mitchell, 1997] Mitchell, T. M. (1997). Machine learning. New York: McGraw-Hill

[Murty & Devi, 2015] Murty, M. N., Devi V. S. (2006). Introduction to Pattern Recognition and Machine Learning. Singapore: World Scientific Publishing Co

[Nasdaq Nordic, 2016] Historiska Kurser – Aktier. http://www.nasdaqomxnordic.com/shares/historicalprices , [2016-05-15]

[Nguyen, Shirai & Velcin, 2015] Nguyen, T. H., Shirai, K., Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems With Applications*, vol. 42, p. 9603–9611

[The R Foundation] What is R? https://www.r-project.org/about.html [2016-05-02]

[Raghavan &Wong, 1986] Raghavan, V. V., Wong, S. K. M. (1986). A critical analysis on vector space model for information retrieval. *Journal of the American Society for Information Science*, vol 37, p. 279-287

[Ranco et al, 2015] Ranco, G., Darko, A., Caldarelli, G., Grcar, M., Mozetic, I. (2015). The Effects of Twitter Sentiment on Stock Price Returns. *PLOS ONE*, 10(9): e0138441. doi:10.1371/journal.pone.0138441

[Schumaker & Chen, 2009] Schumaker, R. P., Chen, H. (2009). A Quantitative Stock Prediction System Based on Financial News. *Information Processing & Management*, vol 45, p. 571-583

[Si et al, 2014] Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., Deng, X. (2014). Exploiting Topic based Twitter Sentiment for Stock Prediction. *Conference on Empirical Methods in Natural Language Processing* (EMNLP), p. 1139-1145. Doha, Qatar October 25-29, 2014

[Spencer, 2015] Spencer, N. (2015). Machine Learning: Supervised Learning pt. 2 http://www.nelsonspencer.com/blog/2015/2/15/machine-learning-supervised-learning-pt-2, [2016-05-21]

[Stengård, 2013] Stengård, M. (2013). Så lurade bedragarna hela börsen. *Aftonbladet*, 2013-10-11 http://www.aftonbladet.se/minekonomi/article17641651.ab [2016-05-02]

[VA, 2015] Läkarstudenterna tjänade miljoner på att manipulera aktiekurser. *Veckans Affärer*, 2015-08-06 http://www.va.se/nyheter/2015/08/06/lakarstudenterna-tjanade-miljoner-pa-att-manipulera-aktiekurser/ [2016-05-02]

[Weka, 2016] Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/ , [2016-05-02]

[Wikipedia, 2016] Wikipedia, Iris Flower Dataset https://en.wikipedia.org/wiki/Iris_flower_data_set [2016-05-04]

[Wu, Zheng & Olson, 2014] Wu, D. D., Zheng, L., Olson, D. L. (2014). A Decision Support Approach for Online Stock Forum Sentiment Analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, p. 1077-1087