FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Anastasia Serebryannikova

Predicting Stock Market Trends from News Articles

Institute of Formal and Applied Linguistics

Supervisors of the master thesis:	doc. RNDr. Vladislav Kuboň, PhD Dr. Malvina Nissim
Study programme:	Computer Science
Study branch:	Computational Linguistics

Prague 2018

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date .... signature of the author

Here I would like to thank all the people who were involved in conducting this work. Without your help, patience and support I would not have managed it.

First of all, I would like to thank my supervisors, doc. RNDr. Vladislav Kuboň, PhD and Dr. Malvina Nissim. Dear Dr. Kuboň, thank you for preparing me for all the difficulties that were awaiting for me, for your endless willingness to help and the great ideas that you were providing me with. Dear Dr. Nissim, thank you for your smile, cheerfulness, support and encouragement, our conversations were making me feel much better and much more confident about what I was doing. I am also greatly thankful for your flexibility and for always being available, especially during the final phase of my work.

I would also like to thank my family and my friends, who were always there to cheer me up and encourage me to continue working. I am very thankful to Alexander Manichev for the idea of the thesis topic as well as for his economical consultations throughout the year. I would also like to thank Xiaoyu Bai, Micha de Rijk and Rakel Brandt, who supported me a lot during my studies in Groningen. Besides, I am overly grateful to my beloved dorm team in Prague, especially to Maria Obedkova and Akshay Aggarwal, for always providing me with the reasons to not be working on my thesis as well as for their support, jokes and our nights out. Last but not the least, I am very thankful to Vladan Glončák, who was always there for me and without whom I would not have survived these two years of Master studies at all.

Title: Predicting Stock Market Trends from News Articles

Author: Anastasia Serebryannikova

Institute: Institute of Formal and Applied Linguistics

Supervisors: doc. RNDr. Vladislav Kuboň, PhD, Institute of Formal and Applied Linguistics, Dr. Malvina Nissim, Center for Language and Cognition Groningen

Abstract: In this work we made an attempt to predict the upwards/downwards movement of the S&P 500 index from the news articles published by Bloomberg and Reuters. We employed the SVM classifier and conducted multiple experiments aiming at understanding the shape of the data and the specifics of the task better. As a result, we established the common evaluation settings for all our subsequent experiments. After that we tried incorporating various features into the model and also replicated several approaches previously suggested in the literature. We were able to identify some non-trivial dependencies in the data which helped us achieve a high accuracy on the development set. However, none of the models that we built showed comparable performance on the test set. We have come to the conclusion that whereas some trends or patterns can be identified in a particular dataset, such findings are usually barely transferable to other data. The experiments that we conducted support the idea that the stock market is changing at random and a high quality of prediction may only be achieved on particular sets of data and under very special settings, but not for the task of stock market prediction in general.

Keywords: stock market prediction, machine learning, S&P 500 index, news articles

Contents

Introduction **3**

1 Theoretical Background **5**
1.1 Stock market indices 5
1.2 The functioning of the stock market 6
1.3 The stock market and financial news 6
1.4 Predictability of the stock market data 7

2 Previous work **9**
2.1 Classification vs. regression 9
2.2 Specific tasks and evaluation difficulties 10
2.3 Data 11
2.4 Machine learning methods and features 12
2.5 State-of-the-art results 15

3 Data **17**
3.1 Textual data 17
  3.1.1 Preprocessing of the dataset 18
  3.1.2 Train, development and test sets 18
  3.1.3 Basic statistics 19
3.2 Stock price information 21

4 Basic model and settings **23**
4.1 The baseline 24
4.2 The basic model 24
4.3 Data alignment 26
  4.3.1 Opening price as the starting point 28
  4.3.2 Closing price as the starting point 30
  4.3.3 Special treatment for Mondays 32
4.4 Conclusions and setup modifications 40

5 Model extensions **41**
5.1 Sentiment 41
  5.1.1 Sentiment dictionaries 41
  5.1.2 Calculating the sentiment of a text 43
  5.1.3 Sentiment as a text representation technique 43
  5.1.4 Appending sentiment information to the TF-IDF representation 45
  5.1.5 Normalizing sentiment scores 46
  5.1.6 Discussion 48
5.2 N-grams 49
5.3 BNS feature selection 50
5.4 Previous day information 52
5.5 News titles 54
5.6 Day of the week 55

6 Final models **56**
6.1 Combining all our findings 56
6.2 Final models and their performance on the test set 57
6.3 Discussion and error analysis 58
6.3.1 C value 58
6.3.2 Balancing out the training data 59
6.3.3 Extending the training data 59
6.3.4 Balancing out the testing data 60
6.3.5 Mixing development and test data 61
6.4 Final remarks 62

Conclusion **64**

Bibliography **66**

List of Figures **70**

List of Tables **71**

List of Abbreviations **73**

A Attachments **74**
A.1 DVD 74
A.2 Example of a news article from Bloomberg 74
A.3 Example of a news article from Reuters 75
A.4 Performance of the other models on the test set with adjusted C values 77
A.5 Excerpt from the financial dictionary by Loughran and McDonald [2011] 78
A.6 Correlation between the sentiment scores and the target labels . . 79

Introduction

In the recent years there is an ongoing trend to apply natural language processing (NLP) tools to the areas that at first appear to be completely unrelated to the area of linguistics. There are various examples of such applications in the fields of forensic investigation, medical research and financial forecasting. Whenever there is a huge amount of textual information available, NLP tools come very handy and allow for instant processing of large-scale data and automatized decision-making.

Stock market prediction is one of the areas where these virtues are extremely useful and can also allow for making profit. A lot of researchers are lately inclined towards the text mining approach to stock market prediction, which makes use of the unstructured data such as (financial) news articles, corporate disclosures or tweets to forecast the stock price movements. The amount of available textual information is constantly growing and the ability to process this data quickly and efficiently may substantially increase the chances of making an informed investment decision with highly profitable potential.

Generally speaking, predicting stock market trends is one of the possible applications of text mining techniques in financial domain. Other possible applications include FOREX rate prediction, Customer Relationship Management and financial cybersecurity (see Kumar and Ravi [2016]).

According to Nassirtoussi et al. [2014], at least three different fields need to be considered for a thorough research in the area of stock market prediction: linguistics, machine learning and behavioral economics (see Figure 1).

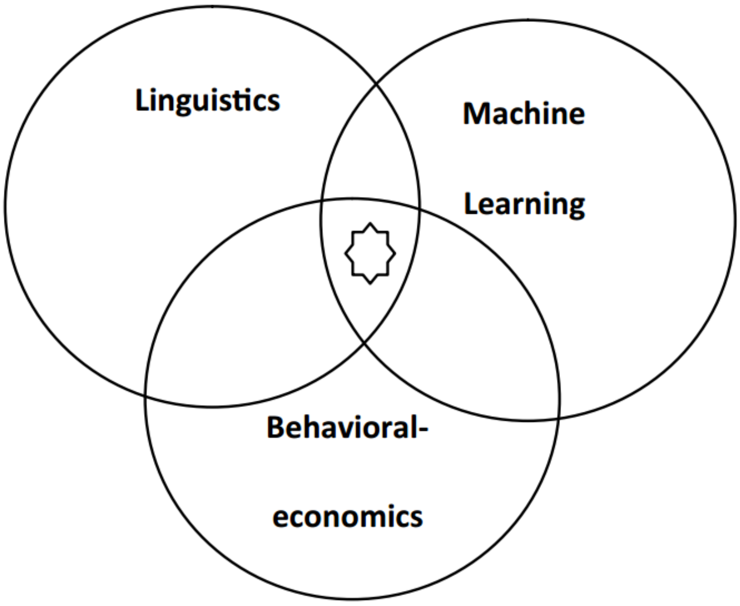

Figure 1: The interdisciplinary nature of the field (Nassirtoussi et al. [2014])

Linguistics is needed to enable the processing of the textual data and uncovering the means by which the text can convey sentiment or other meaning relevant to the task; machine learning serves for establishing the dependency between the textual data and the stock prices. Behavioral economics provides the theoretical explanation of why this kind of dependency actually exists and sets up the economical sense behind it. In our research we will mostly be concentrating on the first two of these fields.

The works conducted in the area of stock market prediction are diverse and have covered most of the possible settings for this task. However, the reported results are sparse and barely comparable to each other because of the different evaluation settings. We will follow one of the possible directions in this domain and make an attempt to forecast the daily direction of the S&P 500 index using financial news articles. Even though financial news is one of the main sources of information that are proven to affect the stock market, the dependency between the news and the stock market is non-straightforward and can be often interrupted by other factors, which makes the task of stock market prediction very challenging. This explains why most of the results achieved in this domain do not exceed 70% accuracy.

Considering the huge amount of work done in this area, the existing sparseness in the reported results and the incomparability of most of the implemented approaches, we have set the following goals for ourselves. First, we will try to set up the evaluation standards for this task and reveal the hidden variables that were ignored or overlooked by the previous researchers. As part of this effort, we will try to identify the observation window that serves best for the prediction of the S&P 500 index price movement and we will decide on the way to derive the target labels for the data samples. Second, as most of the research was conducted on different data, we will try to replicate some of the existing approaches on the same dataset in order to provide comparability between the achieved results. Apart from that, we will attempt to extend the model with some other features that we designed ourselves. Last, we will try to improve the results of the prediction by combining our findings with previous achievements in this domain.

The work will be structured as follows. In Chapter 1 we will provide the theoretical background behind the stock market predictability and give some general information about the functioning of the stock market that is relevant for our research. In Chapter 2 we will give an overview of the existing works conducted in the area of stock market prediction from news articles. Chapter 3 will be devoted to the description of the dataset that we will use for our experiments. In Chapter 4 we will describe the baseline for our experiments, the basic model and establish the common setup for all further experiments. In Chapter 5 we will explore various textual and non-textual features that could be used to enhance the model. In Chapter 6 we will evaluate our best models on the test set and comment on the results. In Conclusion we will give an overview of the work that has been done.

1. Theoretical Background

Is it generally possible to predict the stock market from news? What do we need to know about the functioning of the market to solve this task efficiently? These are the questions that we will try to answer in this chapter.

This chapter will be structured as follows. Section 1.1 will provide the description of the stock market indices in general and S&P 500 index (which we will be attempting to predict) in particular. In Section 1.2 we will explain the peculiarities of the stock exchange functioning that we need to consider in order to build a model that would have some economical sense behind it. In the Section 1.3 we will be talking about the connection between the stock market and the financial news. Section 1.4 will be devoted to the theory behind the predictability of the stock price movements.

1.1 Stock market indices

This work will focus on the prediction of the S&P 500 index, so we would like to provide some general information about the stock market indices in general and S&P 500 index in particular.

Stock market index is a measurement of (a section of) a stock market. It is usually calculated as a weighted average of the prices of selected stocks. A stock market index is usually regarded as an indicator of the current state of the market or market sector.

Standard and Poor's 500 index (most commonly referred to as S&P 500) is one of the most important and well-known stock market indices. S&P 500 index comprises the 500 most widely-traded U.S. companies. The value of the index is calculated upon the prices of the stocks of these companies weighted by capitalization. Obviously, the weighting strategy suggests that the rise in the price of a specific stock included in the index does not necessarily lead to the increase in the index price, and vice versa. For the index price to change significantly, some major events influencing a large spectrum of companies have to occur. Moreover, some of the events can cause a rise in one sector of the economy and a fall in another one, therefore the dependency between the the stock market index, the share prices of its constituents and the new piece of information introduced becomes even more difficult. S&P 500 index is usually used to describe the U.S. marketplace on the whole and serves as an indicator of the overall state of economy. It covers most sectors of the industry and describes approximately 80% of the total U.S. capitalization.

Despite the fact that the primary role of a stock index is providing an overview of (a segment of) the market, it is also possible to invest in stock indices. The most primitive strategy here is investing in all the stocks that are included in the index directly (a.k.a. index replication). However, some more advanced strategies also exist. For example, it is possible to invest in the mutual funds (such as Vanguard) that seek to replicate the performance of the index by investing in its constituents with similar weights. Another solution is investing in an exchange-traded fund (ETF), which is similar to a mutual fund except for the fact that it is traded on the market during the day (whereas a mutual fund is traded at the closing net asset value).

Therefore, the prediction of a stock market index does not only have a theoretical value, but can also be used for making profit.

1.2 The functioning of the stock market

The stock exchanges may function a bit differently from country to country. In this section we will give a brief overview of the New York Stock Exchange (NYSE) which is relevant to our research because we are considering the U.S. marketplace.

The trade at NYSE starts at 9:30 and finishes at 16:00 EST and usually happens from Monday to Friday. The pre-market trading activity takes place between 8:00 and 9:30, the after-hours trading happens from 16:00 till 20:00. The stock prices are changing constantly during the trading hours, therefore the main reference points to characterize the activity throughout the day are the opening and closing prices. The stock exchange is closed on weekends and public holidays.

The opening price of the following day is most commonly different from the closing price of the previous day due to the events occurring outside trading hours, e.g. the after-hours trading. Another numerical value characterizing the market that can be considered by the investors is the adjusted closing price. In contrast to the raw closing price, which reflects the last price at which the share was traded during the (trading) day, the adjusted closing price reflects not only the market events, but also the corporate actions, i.e. it is adjusted for dividends and splits.

1.3 The stock market and financial news

The stock market serves for issuing, buying and selling stocks that are regarded as a representation of a fractional ownership in a company. By selling the stocks, companies can get funding for their business. People who buy the stocks - the investors - can make profit from the dividends paid out by the company or resell their stocks when the price rises and gain profit based on the price change.

The dynamics of the stock prices is complex and is proven to be affected by various factors such as the overall state of economy, wars and political decisions made by the government, inflation and unemployment rates, operational activity of the company, demand for the company's products, the company's business strategy and so on.

The main mechanism behind the change of the stock price is the law of supply and demand. When there are more people willing to buy the shares of a company, the price for the share grows. When there are more shares offered than people are willing to buy, the price goes down. One of the main strategies when making the decision about buying or selling stocks is analyzing the available information about the company. Publicly owned companies have to disclose most of the information about their operational activity, financial situation, earning expectations and so on, so that the investors could make an informed decision.

The financial news articles is one of the most valuable sources of textual information among the ones that are proven to affect the stock market. They reflect the most recent changes in the company's strategies, the deals that the company has made and so on. Among all types of textual information that is available, financial news are considered by many (see Luss and d'Aspremont [2015], Schumaker et al. [2009]) to be the most relevant.

Let us consider the following example to demonstrate the dependency between the share price and the news articles. On 27^{th} of June 2012, Bloomberg published the news article about the patent dispute between Apple and Samsung and the court decision that was made in favour of Apple. An excerpt from this article is given below, the full version can be found in Appendix A.2.

Apple Wins Preliminary Injunction Against Samsung Tablet

Apple Inc. (AAPL) won a court order immediately blocking U.S. sales of Samsung Electronics Co. (005930)'s Galaxy Tab 10.1 tablet computer as the companies continue their global patent dispute. U.S. District Judge Lucy Koh in San Jose, California , issued the order yesterday after rejecting a similar request in December. Apple's request, part of a broader patent dispute over smartphones and tablets, was based on an appeals court finding that it will probably win its patent-infringement claim relating to the Tab 10.1 tablet. The world's two biggest makers of high-end phones have accused each other of copying designs and technology for mobile devices and are fighting patent battles on four continents to retain their dominance in the $219 billion global smartphone market. ...

This news article reflects the success of the Apple Inc. in dealing with patent affairs and conquering the corresponding market segment. In the days following the publication of this article, Apple stocks were continuously growing, which is a good example of the impact that positive news can have on the stock price.

However, even though sometimes it is possible to establish the connection between the information flow and the share price, this dependency is complex and non-straightforward. The market players sometimes act irrationally causing the effects of overreaction and underreaction; the market itself oscillates between the periods of higher and lower predictability. Furthermore, the connection between the stock price and the (financial) news may easily be interrupted by large unexpected trades. On top of all of that, there is always an element of randomness in the price change (see Section 1.4 for more details).

1.4 Predictability of the stock market data

It was believed for a long time that it is impossible to predict the stock price movements effectively. This assumption was based on the Efficient Market Hypothesis (EMH) and the Random Walk Theory (RWT). Efficient Market Hypothesis (Fama [1965]) claims that at a given point in time all the available information is already incorporated in the share price and therefore it is impossible to outwit the market and speculate on the stock prices making profit that is higher than average. In other words, it says that the stock price changes instantly after a new piece of information occurs, which means that the price of the stock changes faster than any action could possibly be taken. In this situation the current price of the stock reflects its intrinsic value in the best possible way, therefore it is impossible to buy an underrated stock and resell it with a profit later when its current value reaches its fair value.

The Random Walk Theory (Malkiel [1973]) states that the stock price movements resemble a random walk and therefore no reliable prediction can be made about the future changes, i.e. it says that neither the historical data nor the current economical situation or operational activity of the company actually influence the stock price directly. An efficient market is believed to be random because the occurrence of the events possibly influencing the stock price is also unexpected.

Despite the fact that both of these theories actually discouraged any type of attempts to predict the stock price movements, the recent advances in artificial intelligence and the growing availability of wide-scale data have made it possible to challenge the assumptions of the market unpredictability and to go beyond the predictions that are close to random. There are many possible approaches to stock market prediction, and most researchers usually stick to one of the following three:

1. Technical approach. Within this approach, the stock prices are predicted based on the historical information about the price changes. A set of technical indicators calculated upon this data is usually used to characterize the current state of affairs and make a prediction about the future dynamics. One of the most common techniques is finding repetitive trends in the dynamics of the technical indicators and/or historical prices (e.g. “head and shoulders” pattern).

2. Fundamental approach. This approach relies on the assumption that the stock price is independent on the past changes and is mostly affected by the events in the outer world as well as the overall economical or political situation. The information that is usually taken into account includes the inflation and unemployment rate, the organizational changes in the company and the recent deals that it was involved in, the overall political situation and the customers’ interest in the products of the company.

3. Combined approach. This approach tries to incorporate both types of information available (the historical data as well as the data characterizing the state of economy) in order to make a prediction.

As we are using textual data for the prediction, the most reasonable choice for our work would be fundamental approach. However, in some of our experiments we will also employ the combined approach by incorporating the historical data in the prediction model.

2. Previous work

This chapter will be devoted to describing the most recent and relevant research conducted in the area of predicting stock market trends *from news articles* or similar sources. An attempt to summarize the existing findings in this area has already been made in several works. A general overview of the applications of text mining in financial domain in general can be found in Kumar and Ravi [2016]. If we consider text mining applications to stock market prediction in particular, one of the most detailed overviews is provided in Nassirtoussi et al. [2014].

In the following sections we will try to point out the most recent advances in this domain and mention the works that we find the most relevant for our research. We will focus on the works that involve working with textual data, and namely news articles, as opposed to the ones focusing on time series information and technical indicators.

Most existing works differ with respect to the methodological decisions and design choices made by the authors. This chapter will be structured in accordance with the variability points in the existing research: the general approach, the subtasks and the evaluation setup, the used data and so on.

2.1 Classification vs. regression

One of the first and crucial decisions that one has to make when predicting stock market trends is to decide which type of machine learning task we are trying to solve, i.e. to get a very general understanding of how to tackle the problem. One of the possible aims is to predict the directionality of the movement (up/down), thus the stock market prediction turns into a binary **classification** task. In most cases the positive label (+1) denotes the upwards movement or no change in the price, whereas the negative label (denoted as -1 or 0) means the downwards movement. Such approach appears to be one of the most common ones and was used in Zhai et al. [2007], Hagenau et al. [2013], de Fortuny et al. [2014], Ding et al. [2015].

It is also possible to tackle the classification problem with more detail, introducing more than two possible outcomes (e.g. distinguishing between strong and weak changes and/or introducing a zero change as one of the possible outcomes). Then the problem of predicting stock market trends is also treated as a classification task, with all of the data points belonging to 3 to 5 different classes characterizing the directionality and/or intensity of the stock price movement. This approach was implemented in Gidofalvi and Elkan [2001], Mittermayer [2004], Rachlin et al. [2007], Gunduz and Cataltepe [2015].

Another task is predicting the exact value of the stock price. In this case the machine learning task appears to be a **regression** problem. The most successful attempts of predicting the stock price values via regression were made in Schumaker and Chen [2009a], Schumaker and Chen [2009b]. Generally speaking, this approach is selected rarely because in most cases the relative change in the price is more important than its exact value. As well as this, evaluating the performance of the system is less straightforward and requires using several evaluation metrics at once (e.g. closeness and directional accuracy). Alternatively, it is possible to use the coefficient of determination R^2. However, its main disadvantage is that it automatically grows when the number of explanatory variables increases, therefore it is impossible to directly compare models with different number of predictors.

2.2 Specific tasks and evaluation difficulties

After a researcher has decided on a general approach, it is necessary to formulate the subtask(s) he/she is trying to solve. Stock market prediction by itself appears to be a very vaguely formulated goal because there are several things that we can actually try to predict depending on our final objective. Most researchers usually explore only one of the possible options for setting up the experiments in this domain without explaining most of their decisions in detail, which results in a very sparse set of settings for the models which are almost never the same across different studies, which makes the task of comparing different approaches and summarizing the existing experience even more difficult.

First, the discrepancies between different studies concern the time period for which the prediction of the stock price change is made. Second, the way to estimate the price change is also not uniform.

The time frame for the prediction (= the influence window) varies from very small time lags (e.g. 1, 2, 4 minutes in de Fortuny et al. [2014]) to much larger ones (1 hour in Mittermayer [2004], 1 day in Wuthrich et al. [1998]). Whereas most researchers usually define the influence window as something that follows the publication time (immediate following is not necessary), some researchers go further and define the influence window as something that starts even before a new piece of information is introduced (see Rachlin et al. [2007], Hagenau et al. [2013]).

On high-granularity level (per minute), one of the most common approaches is to use a time lag of 20 minutes (see Schumaker et al. [2012]). The popularity of this approach can be explained by the theoretical foundations behind it discovered by Gidofalvi and Elkan [2001], who showed that there is a 20-minute period of weak predictability following the appearance of a new piece of information.

Deriving the classification label from the observed market data is also not a task that is solved uniformly. The highest variance can be observed for the "next day" prediction, where every researcher is basically trying to predict his own indicator derived from the market data. Just to give a few examples:

- "The purpose of this study is to learn how news articles released within a day affect the **direction of [BIST 100 Index] open price** the next day", where direction of the open price is actually defined as "**direction of the price change from the closing price of the day before**" (Gunduz and Cataltepe [2015])
- "Our goal is to predict the **forthcoming trend which will last more than a predefined time length**" (within the 24-hours influence window) (Rachlin et al. [2007])
- "For the one-day-ahead setting, the labels are based on **opening and closing quotes**" (de Fortuny et al. [2014])

- “The target output consists in a binary variable where a value [1,0] represents that the **close price in the day** $t+1$ **will increase compared with the closing price in the day** t and a value [0,1] represents that the **close price in the day** $t+1$ **will decrease compared with the previous day**” (Vargas et al. [2017])
- “We estimate the binary movement where 1 denotes rise and 0 denotes fall, $y = \mathbb{1}(p_d^c > p_{d-1}^c)$, where p_d^c denotes the **adjusted closing price** adjusted for corporate actions affecting stock prices, e.g. dividends and splits” (Xu and Cohen [2018])
- “For events **during trading hours**, the stock price effect is calculated **between open and close** auctions. For events occurring **outside trading hours**, the effect calculation is based on **close prices of the previous day and open prices**” (Hagenau et al. [2013])

For the minute-level granularity the way to define the prediction target is usually more uniform and consists of calculating the difference between the price at the time of publication and the price at the end of the time lag and assigning a predefined label to it. However, alternative approaches to target label assignment are also possible, e.g. in Mittermayer [2004]: “we define as good news all press releases that lead the stock price concerned to peak, with an increase of at least +3%, at some point during the 60 minutes immediately after publication and have an average price level in this period that is at least 1% above the price at the time of the release”.

All together, we can see that there are various tasks that are gathered under the umbrella term of stock market prediction. In fact, most researchers have to start with formulating and specifying their own goals which are usually different from the ones that the other researchers have set. This is one of the main reasons why the evaluation standards for this task are weak. As part of our own work, we will address some of the issues connected to different experimental setups and evaluation settings in Section 4.3.

2.3 Data

A typical setup for the goal of stock market prediction using textual data implies using at least two different sources of information: the stock market information and the textual data. Some researchers also employ other types of data (e.g. the historical prices or technical indicators) to boost the accuracy of the classifier. The works that only use numerical data for the prediction stay behind the scope of this study.

With regard to the stock price information, several things can become prediction targets. It is possible to predict the price change for separate stocks or stock market indices. The researcher may choose to concentrate on the stocks of the companies from a specific branch of the industry (e.g. healthcare in Shynkevich et al. [2016]), conglomerates (Xu and Cohen [2018]) or companies with a high turnover (Mittermayer [2004]). On the other hand, it is possible to predict a somewhat more general state of the market through one of the indices such as S&P 500 (Vargas et al. [2017]), DJIA (Bollen et al. [2011]), BIST 100 (Gunduz and Cataltepe [2015]), BOVESPA (Nizer and Nievola [2012]) and so on. The indices are calculated as a weighted average of the prices of selected stocks from a specific market or market section. Thus, an index is usually regarded as a representation of the whole market and is used to track the changes in the market through time. Some researchers have also attempted to predict a different indicator, and namely the market volatility. For example, Lavrenko et al. [2000] tried to classify the news into the ones that are likely to affect the stock market behaviour and the ones that are not. Another possible prediction target is the value of stock returns (Seng and Yang [2017]).

The existing research varies also with respect to what kind of textual data is used for the prediction. There have been attempts to use corporate news articles (Hagenau et al. [2013]), financial news articles (Schumaker et al. [2012]), general worldwide news (Wuthrich et al. [1998]), press releases (Mittermayer and Knolmayer [2006], Luss and d'Aspremont [2015]) and corporate disclosures (Groth and Muntermann [2011]).

When using news articles, several researchers have tried incorporating news articles of different specificity level: e.g. company specific news and general market news in Zhai et al. [2007]. More elaborate division of news into categories of different relevance was used in Shynkevich et al. [2016] (stock-specific data, sub-industry-specific data, industry-specific data, group-industry-specific data, sector-specific data) and Schumaker and Chen [2009b] (same plus universal news). Sector-based training appeared to have the highest predictive power for forecasting the price movements of separate stocks in Schumaker and Chen [2009b]. For healthcare companies, group-industry-specific data turned out to be the most indicative (Shynkevich et al. [2016]). In both cases, however, the stock-specific news yielded the lowest performance.

Whereas the news articles and other official sources of business information are supposed to reflect the objective reality, there has also been some research on how the subjective information such as mood or anxiety level of the customers can affect the stock market. This type of research is usually conducted on the data gathered from social media, e.g. LiveJournal (Gilbert and Karahalios [2010]) or Twitter (Bollen et al. [2011], Mittal and Goel [2012], Xu and Cohen [2018]). Whereas the former approach (the one based on the news articles and such) attempts to represent the stock market as a system that reacts to the events rationally, the latter approach (focusing on social media) rather tries to model the dependency between the stock market and some subjective variable (e.g. public sentiment, Twitter mood or anxiety level), thus emphasizing the irrational nature of stock market and its actors.

Whereas the textual data appears to be one of the main sources of information for the fundamental approach to stock market trend prediction, some researchers may decide to incorporate some other market-related data, for example the technical indicators or the time series information (Vargas et al. [2017], de Fortuny et al. [2014], Shynkevich et al. [2014], Zhai et al. [2007]), thus implementing the combined approach mentioned in Section 1.4.

2.4 Machine learning methods and features

The most common approach to representing textual data in machine learning tasks is the *bag-of-words* (BOW) model, also called *vector space model.* Every document in the collection is represented as a vector in a multidimensional space. The features for the model are most commonly unigrams (separate tokens), sometimes bigrams or trigrams (two or three subsequent words). Every position in the vector corresponds to a specific feature (e.g. uni-/bi-/trigram). The easiest approach to represent a document as a vector is to calculate the term frequency (TF) for each of the features. However, this approach is not flexible enough as it gives wrong estimates for the documents of different length (it does not account for the fact that the longer the document, the more likely it is to encounter a particular word in it). For this reason, most researchers use TF-IDF instead (term frequency multiplied by inverse document frequency), which gives reliable estimates of how specific a particular word is in the context of all the documents in the collection.

The TF-IDF weight for a term t in the document d from the collection of documents D can be calculated as follows:

$$\text{TF-IDF}(t, d, D) = tf(t, d) \times idf(t, d, D), \tag{2.1}$$

where the equation for calculating the *idf* looks like this:

$$idf(t, d, D) = log\frac{|D| + 1}{|\{d \in D|t \in d\}| + 1} + 1 \tag{2.2}$$

Adding ones to the numerator and the denominator in the *idf* formula allows for smoothing and serves for preventing zero divisions. All together, it looks as if there was an extra document in the collection that contained all the words from the vocabulary exactly once. The extra one that is added to *idf* at the end is to make sure that the words having zero *idf* (i.e., the ones that occur in every document of the collection) are not completely ignored.

Given the BOW representation of all the data samples with TF-IDF weighting, it is possible to apply any machine learning technique to it. In the area of stock market prediction, the most commonly used method appears to be Support Vector Machines (SVM) (see Luss and d'Aspremont [2015], Schumaker and Chen [2009b]). Previous research also includes works conducted with Decision Trees (Rachlin et al. [2007]) and Naive Bayes (Gidofalvi and Elkan [2001]).

In the recent years there is an ongoing trend to apply artificial neural networks to solving problems in the areas connected to text processing. Unlike traditional machine learning approaches, neural networks tend to use other text representation techniques, namely word embeddings (see e.g. Ding et al. [2015]). More elaborate options employed for the task of stock market prediction are sentence embeddings (Vargas et al. [2017]) and event embeddings(Ding et al. [2015]).

Our work will concentrate on SVM, which was reported to be a very powerful method well-suited for the task of stock market prediction (see Mittermayer [2004], Mittermayer and Knolmayer [2006], Zhai et al. [2007], de Fortuny et al. [2014], Shynkevich et al. [2016]). Despite the fact that neural networks appear to be the state-of-the-art solution to most problems in the area of machine learning, we have several reasons to not chose neural networks in this research. First of all, there have already been several works employing neural networks of different architecture (CNN, RNN etc.) on exactly the same dataset (Ding et al. [2014, 2015], Vargas et al. [2017]), therefore coming up with something new would be problematic. Second, in our work we would like to devote quite a lot of time to exploring the shape of the data and understanding the nature of it better; SVM is better suited for that due to the running time and memory limitations. Moreover, analyzing the contribution of separate features to the predictive power of the model is also much simpler with SVM. The experiments we conducted are numerous (see Chapters 4, 5, 6) and performing all of them with NNs would be time-consuming and resource-inefficient, not to mention that we would not be guaranteed to achieve better results (see Groth and Muntermann [2011], where NNs are outperformed by SVM). We believe that SVM will allow us to dive deeper into the problem and to avoid too much hassle with defining the network architecture, which is not directly related to the task we are trying to solve. Using a simple method such as SVM that does not require too much tuning and is time-efficient will allow us to try more different setups for the experiments and to avoid a too shallow approach to the problem. Lastly, we would like to point out that the SVM approach is still popular in the recent works (Shynkevich et al. [2016], Rahman et al. [2017]) and appears to be the most common choice for the task of stock market prediction in general, which also makes it easier for us to replicate some of the existing solutions on the dataset we are using.

One of the additional steps that one may want to incorporate into the model is feature selection. Whereas the BOW model is already a valid representation of the textual data, one may want to modify it in some way, e.g. to select only the most important words for the document representation. Another option is adding additional sets of features (which do not necessarily have to be text-related) to the model.

One of the most common research directions is subsetting the most informative words from the vocabulary and only using these words for the data representation (via vector space model or similar). This method helps to decrease the sparseness of the feature space and has a positive impact on the training time.

The following methods have been proposed for feature selection:

- Mutual information (MI), Balanced mutual information (BMI) (see e.g. Gunduz and Cataltepe [2015])
- Chi-square (χ^2) (Groth and Muntermann [2011])
- Bi-normal separation (BNS) (Hagenau et al. [2013])
- Threshold of the term frequency (Schumaker and Chen [2009b])

Another direction of research is finding better representations for the textual data other than simple n-gram (most commonly, unigram) models. The existing research has shown that:

- Proper Noun representation of financial news articles performs better than BOW, Noun Phrases and Named Entities (Schumaker and Chen [2009a])

- 2-word combinations[1] (distance up to 5) perform better than unigrams, bigrams and noun phrases (Hagenau et al. [2013])

Another feature that is usually associated with the stock market prediction is the sentiment. It is more commonly used with Twitter data, which is believed to reflect the society mood pretty well (see Vu et al. [2012], Mittal and Goel [2012], Oliveira et al. [2017]). However, it is also not uncommon to apply sentiment analysis to the news articles in order to predict the stock price movements (see Schumaker et al. [2012], de Fortuny et al. [2014]).

Whereas the research conducted by Koppel and Shtrimberg [2006] shows that there is indeed some dependency between the sentiment of the news articles and the change of the stock price (the stock price movements can be used to assign the positive/negative sentiment labels to the news articles with an accuracy of 70.3%) and multiple works have shown that including the sentiment information does increase the accuracy of prediction (see de Fortuny et al. [2014]), some researchers still point out that "the sentiment results are inconsistent and often underperform the other models as compared to the bag-of-words approach or the technical indicators (often even underperforming a random classifier)" (de Fortuny et al. [2014]). Therefore, the sentiment information by itself does not seem to be a reliable indicator for the prediction of the stock price movement and should better be used in combination with the other features.

As already mentioned in Section 2.3, another common type of features are the technical indicators and/or historical prices. These types of features are completely independent on the textual data and are also taken from a separate source. Several works have shown that combining the textual data with technical indicators (Zhai et al. [2007]) or historical prices (Xu and Cohen [2018]) is a valid approach to solving the problem of stock market prediction.

2.5 State-of-the-art results

As mentioned before, most of the approaches we have seen are impossible to compare directly as most researchers are solving different tasks. Nevertheless, one thing that we can say for sure is that predicting stock market trends from news articles appears to be a very challenging task and most researchers are usually content with the predictions that are at least slightly better than random. Besides, since it is not really common to be reporting on the negative results, we do not know how many unsuccessful attempts to predict the stock market were made. Overall, however, we have to say that even the results slightly exceeding the baseline should not be considered unsatisfactory because even small improvements in the accuracy of the prediction lead to potentially high profits.

Most reported results are in the range of 60-70% accuracy, depending a lot on the task, the setup of the experiments and the data used as well as the methodological decisions made by the researchers.

For the dataset that we are using in our work (see more detailed description in Chapter 3), the highest results were achieved by Ding et al. [2015] and constitute

---

[1] These are very similar to skipgrams; the only difference is that the skipgrams preserve the word order and the 2-word combinations ignore it.

64.21% accuracy on the test set and **65.08%** on the development set when predicting the daily upwards/downwards movement of S&P 500 index. The authors employed a convolutional neural network that used event embeddings as input. This result will provide the main benchmark for our work.

Most of the other works in the domain of stock market prediction are not directly comparable to our research; however, we would like to mention them here to give a general overview of the field.

Short-term stock market prediction generally appears to be a bit easier task, which can be demonstrated by the works by Mittermayer and Knolmayer [2006] and Schumaker and Chen [2009b], who also concentrated on predicting the S&P 500 index. Schumaker and Chen [2009b] managed to achieve a directional accuracy of 71.18 % for the 20-minute time lag, whereas Mittermayer and Knolmayer [2006] reported an accuracy of 82% using noise-free data and a unique labeling approach for the 15-minutes-ahead prediction. We can see from these examples that the setup of the task (and, in particular, the choice of the time frame for the prediction) are the variables that influence the performance of the classifier a lot.

Many researchers also attempted to address the problem of specific stock prediction (as opposed to the prediction of the stock market indices). The reported results suggest that separate stocks can be predicted a bit more efficiently than the stock indices. However, the accuracy of the prediction still hardly ever exceeds 80%. For example, Rachlin et al. [2007] reports an accuracy of 80.6% when predicting the stock price change in a 5-way classification using an automatically created keyword dictionary. The results reported by Groth and Muntermann [2011] may also seem very good (78.96% accuracy) when mentioned without the context. However, the baseline for their system constituted 75%, which means that the model performs only slightly better than the baseline.

We would like to note that most of the results exceeding 70% accuracy actually look like outliers when we compare them to the majority of the works that strive to beat a random predictor. We are not attempting to diminish these achievements but we would like to point out that the results of transferring such findings to different data are likely to be unsatisfying. After conducting our own experiments (see Chapters 4, 5, 6) we are inclined to think that outstanding results can only be achieved when the model is tailored to a specific dataset and very special settings as well as evaluation standards are employed.

Considering multiple sources of information or adding more features to the model usually yields better results as compared to the simple BOW approach. For example, Zhai et al. [2007] reports 64.7% accuracy when using only textual data and 70.1% accuracy when combining the news with technical indicators. Employing sentiment data in the analysis also tends to help: the highest result in the experiments by de Fortuny et al. [2014] constitutes 69.05% and was achieved by employing the information about the sentiment of the news titles.

Occasionally the task of predicting the stock price movements for the stocks from a specific sector of the market can be solved pretty efficiently. For example, Shynkevich et al. [2014] achieved 81.31% accuracy when predicting the stock price changes in the healthcare sector by employing multiple kernel learning and using news articles of different specificity level. However, such results are rare and seem highly dependent on the data used.

Sometimes elaborate feature selection methods provide an unexpected boost in the quality of the prediction. This is the case with the work of Hagenau et al. [2013]: they reported 76.3% accuracy on a similar data and 65.4% on a different dataset when using 2-word combinations for the text representation and BNS-based feature selection utilizing feedback from the stock market. Despite the fact that this result is not directly comparable to our work (because it was achieved in short-term stock-specific prediction), we will try to reproduce this approach to find out whether it can be used effectively for the daily prediction of S&P 500 index as well.

Overall, we can say that whereas some researchers have managed to achieve relatively high results while solving some specific subtasks in this domain, we have not seen many confirmations of successful transfers of these approaches to a new data, which may give us an impression that the high results we have witnessed are sporadic and data-dependent.

All together, we can see that even though a lot of work has been conducted in this direction, the results are very sparse, which means that there are still a lot of weak points and much room for new experiments and possible improvements. We will try to fill in some of the research gaps in Chapters 4 and 5.

3. Data

The specifics of the task requires using at least two main sources of information: the textual data and the stock price data. Providing a reasonable mapping or alignment between the two (i.e. pairing the textual data with the corresponding stock price change) is a separate task that will be addressed in Section 4.3 in detail. In the current chapter we will try to give a general overview of the data that we used in the experiments.

Section 3.1 will be devoted to the description of the financial news dataset that we exploited in our experiments. In Section 3.2 we will provide an outline of the stock price data that we employed.

3.1 Textual data

In this work we are using the financial news dataset[1] that was collected and first used by Ding et al. [2014]. It was then exploited in the research conducted by Ding et al. [2015] and Vargas et al. [2017]. According to the authors, this dataset is one of the biggest ones ever used for the task of stock market prediction. Furthermore, it appears to be the only publicly available dataset comprising financial news articles.

The dataset consists of financial news articles published by Bloomberg and Reuters during the period of more than 7 years (2006 - 2013). More detailed information about the news articles that are included in the dataset can be found in Table 3.1. The dataset contains approximately 201 million tokens.

Source	Number of articles	Dates
Bloomberg	447768	20/10/2006 - 26/11/2013
Reuters	106494	20/10/2006 - 19/11/2013
Total	554262	20/10/2006 - 26/11/2013

Table 3.1: Basic dataset information

Every news article is annotated with a timestamp that enables further alignment with the stock price information. 391 articles from Bloomberg (less than 0.1% of the total number of articles) are only annotated with a date and do not have the information about the time of publication (we assume it to be 0:00 of the corresponding day).

Every news article also has a title associated with it; some of the data samples only contain the title, whereas the text of the article is missing. In such cases, we consider the title to be the whole text of the article.

The title, the timestamp and the text of the article itself constitute the data that we actually use. We ignore the URL of the article and the information about the author.

Sample articles from Bloomberg and Reuters are given in Appendices A.2 and A.3.

---

[1] Retrieved from `https://github.com/philipperemy/financial-news-dataset`

3.1.1 Preprocessing of the dataset

Preprocessing of the dataset as it can be downloaded from the link above involves deleting empty files (628 for Bloomberg, 27 for Reuters) and two more redundant files that do not contain news articles in them. As well as this, we had to replace new line characters in the news titles of three files with whitespaces to avoid parsing mistakes.

The timestamps from Bloomberg are given in UTC time already. However, the timestamps of the articles from Reuters are either in EST or EDT, so we transform them to the UTC format as well. However, the data alignment options that we explore in Chapter 4.3) always refer to EST/EDT, i.e. the local time.

As part of the cleaning procedure, we also remove the information about the editors and authors of the articles as well as the contact details (both of which are usually given at the end of the article) due to the fact that we consider this information to be unimportant for our research.

3.1.2 Train, development and test sets

For splitting the data into training, development and test sets we reproduce the approach that was used with this dataset before, which means that we are splitting the data temporally. We are using exactly the same settings as in the previous works conducted on this dataset in order to be able to compare our results to the previous findings.

Stock market prediction appears to be a time series problem, so the most common method employed for splitting the data is a temporal split. In such case the training data is usually guaranteed to precede the development and/or testing data in time. This approach models the situation when we train the model on the data that is available at the moment of the creation of the model and then test it on the data that appears in real time. The information about the temporal split that we performed is summarized in Table 3.2.

Dataset	**Dates**	**Trading days**	**# of articles**
Train	20/10/2006 - 18/06/2012	1424	357516
Development	19/06/2012 - 21/02/2013	169	96327
Test	22/02/2013 - 21/11/2013	191	99183

Table 3.2: Train, development and test sets: the temporal split

We have to point out that the number of articles given in Table 3.2 is only indicative. In most of our further experiments the exact number of articles used for prediction will be changing in accordance with the observation window limitations.

For some of our experiments we will be using the Monday data separately from the data from the other days (see Section 4.3.3). This will influence the number of training and development data samples that we will employ; some more precise information is given in Table 3.3.

In all of our further experiments we will train the model on the **training data** (unless some other exceptional setting is specified explicitly) and test it on the **development data**. The development data serves for tuning the model and

Dataset	Number of days
Train total	1424
Train Monday	267
Train Tuesday - Friday	1157
Development total	169
Development Monday	31
Development Tuesday - Friday	138

Table 3.3: Number of training samples in different subsets of the data

allows for a better assessment of the features that are relevant for the prediction. This is the main difference between the development set and the **test set**, which is aimed at modeling the performance of the classifier on the previously unseen data.

Having conducted all the necessary experiments, we will develop the final models which will then be tested on the test set. The test set will not be employed in any other experiments.

3.1.3 Basic statistics

In this section we would like to gather some simple statistics about the dataset that we are using. We believe that having this type of information available helps us understand the shape of the data better and can sometimes be used efficiently to improve the quality of the prediction. Some of this information will indeed be used or commented on in the experiments that we conduct (see Chapters 4, 5). All the information reflected in this section was estimated from the training and development sets.

First thing that we would like to draw the attention to is the "news density" throughout the week. The number of news published on a certain day differs substantially depending on whether this day belongs to the workweek or to the weekend. This means that the number of news published on Saturdays and Sundays is much lower than the number of news published between Monday and Friday. The total number of news published on different days of the week as estimated from the training and development data can be seen in Figure 3.1. The observed difference is one of the reasons why we conduct the experiments described in Section 4.3.3 devoted to the Monday data alignment.

The dissimilarity between the workdays and the weekends is not the only peculiarity we can observe. The number of released news also depends a lot on the time of the day. Despite the fact that the news flow is more or less continuous and the news articles are published constantly, the publication time is not distributed uniformly, it has its ups and downs. We can observe the overall number of news published during different hours (from 0 to 23) in Figure 3.2.

This figure is interesting in many aspects. First of all, let us observe the that the greatest number of news is published between 4 and 5 p.m. This is not a coincidence as the stock market closes at 4 p.m. and most companies release the **material news announcements** 15 minutes after the trading stops. Another time when most of the important announcements are made is before the opening

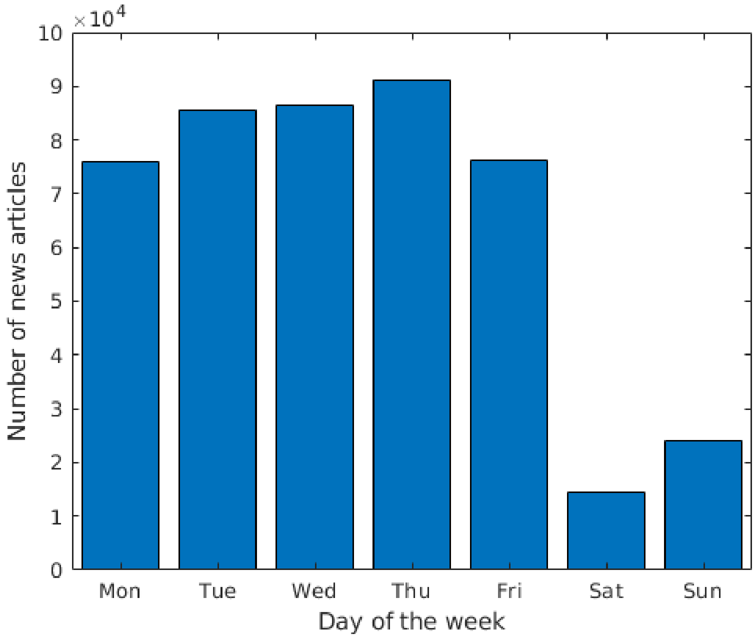

Figure 3.1: Total number of the news articles published per day of the week

of the trade (9:30 a.m.), which is reflected in the graph as another spike in the number of news published between 8 and 9 a.m.

Another interesting effect that we can observe here is the significant increase in the overall number of news between midnight and 1 a.m. Even though this sounds surprising at first, we can probably explain it by the effect of the **news embargo**, which does not allow the publication of some materials until a certain

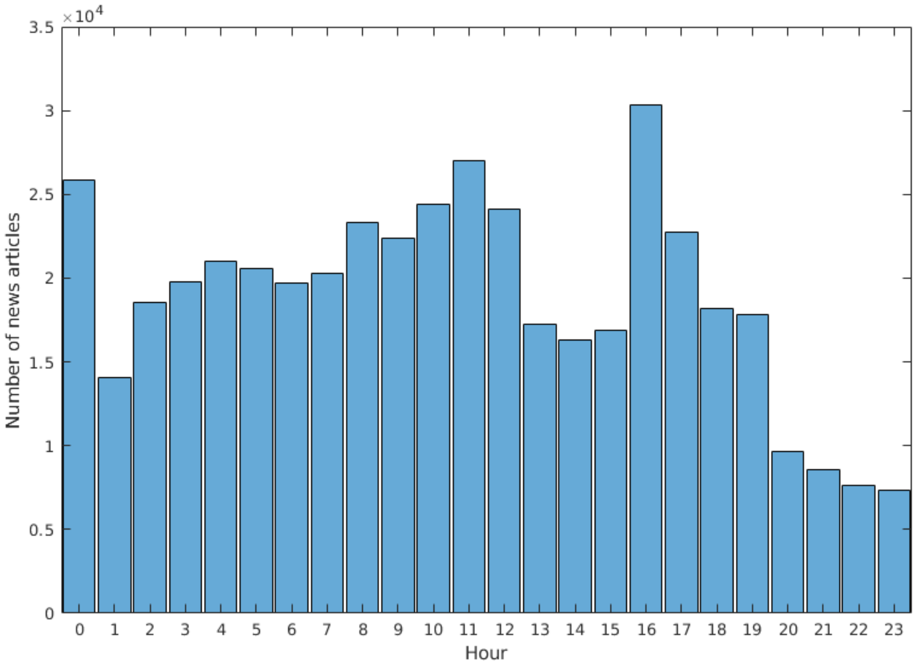

Figure 3.2: Distribution of the number of the news articles published during different hours

date. In case this is the only condition that needs to be met then those news are likely to be released right after midnight.

We can also see from this graph that most of the news is published during the day; the evening hours have the lowest amount of news overall.

All together, it seems that the distribution of news over time reflects the daily life cycle of the stock market pretty well, which makes us hope that we can use this information efficiently for the stock market prediction (see Section 4.3).

3.2 Stock price information

Another source of information that is not less important for the successful prediction is the stock market information. For this thesis work, we are using the stock data concerning the S&P 500 index.

We download the historical prices from the publicly available resource Yahoo Finance (`https://finance.yahoo.com/`). The numerical data that is available there includes the date, open, high, low, close, adjusted close prices and volume. The only data that we actually use is the date and the open and close prices.

As mentioned in Chapter 2, the methodological choices that various researchers make while trying to solve the task of stock market prediction vary a lot. This is also true with respect to deriving the labels for the data samples from historical prices.

Even for the binary classification task (upwards / downwards movement) the label for the same day may be different when calculated in a different way.

For most of our experiments, we will derive the labels from the opening and closing prices of the same day. The rule for the label assignment is as follows: if the closing price on day X is higher or equal to the opening price on the day X, we assign a positive label to that day. Vice versa, if the closing price is lower than the opening price, this day gets a negative label.

The formula for the label assignment would look like this:

$$L_X = \mathbb{1}\ (\frac{P_{close_X} - P_{open_X}}{P_{open_X}} \geq 0), \tag{3.1}$$

where L is the binary label for the day X, P_{open_X} is the opening price on the day X, P_{close_X} is the closing price on the day X.

However, as mentioned before, an alternative approach is also possible and has been used in several works. Within this approach, the starting point for the price change evaluation is not the opening price of the current day, but the closing price of the previous trading day, which we will denote as *X-1*. In particular, we would like to point out that for Monday the previous trading day would be Friday (because the stock market is closed during weekends and public holidays).

We will use closing prices in some of our experiments as well. The label derived from the closing price would be calculated as follows:

$$L_X = \mathbb{1}\ (\frac{P_{close_X} - P_{close_{X-1}}}{P_{close_{X-1}}} \geq 0), \tag{3.2}$$

Let us give an example to illustrate how the labels are assigned to certain trading days and why they can turn out to be different when applying varying strategies. In Table 3.4 we can see a piece of real stock market data that we used for building our model.

Date	Open	Close
2013-10-08	1676.219971	1655.449951
2013-10-09	1656.989990	1656.400024

Table 3.4: Extract from the stock price information

When deriving the label for the 9th of October 2013, we can either use the opening price from the same day or the closing price from the 8th of October as the starting point. The "ending" point in both cases will be the closing price on the 9th of October. In the former case the label for the 9th of October 2013 will be negative. In the latter case, this day will have a positive label.

Such discrepancies are the reason why the prediction of the labels derived from opening [and closing] prices [of the same day] is not exactly the same task as the prediction of the labels derived from closing prices [of two subsequent days]. Another consequence are the slightly different baselines for the two settings. We will further comment on the differences between these two tasks in Section 4.3.

4. Basic model and settings

Having mentioned the previous advances in this domain and having discussed the data that we are going to use, we can finally proceed with our own experiments. This chapter can be regarded as preliminary for our further research; however, the results that we are discussing here are exceptionally important for the future models that we build and constitute the basis for their performance.

As indicated before, one of main aims that we have defined for ourselves is exploring the methodological issues related to the task of stock market prediction. Every researcher is basically trying to solve his/her own task, which results in dozens of incomparable approaches to stock market prediction yielding completely different results. We see our mission in discussing and comparing some of them and revealing the hidden variables of this task that were never discussed previously. As an outcome, we will try to set up **evaluation standards** for this task (or at least a set of possible evaluation standards) that could be used by the researchers who attempt to solve this task in the future.

We have mentioned in Chapter 2 that each of the possible tasks in the domain of stock market prediction has its own peculiarities. For the task that we are trying to solve (the daily prediction of a stock index) one of the main variables that influences the obtained results a lot is the **data alignment**, i.e. the way of pairing the news belonging to the **observation window** with the stock market data from the **influence window**. There is no established way of defining any of them, which means that every time this basically happens in accordance with the will of the researcher (which does not add simplicity to the overall picture).

Consequently, one of the main goals that we will try to achieve in this direction is identifying the time frame which suits for the prediction of the stock market data the best (the observation window). With regard to the influence window, it will be fixed (next day prediction) to provide comparability with previous works conducted on this dataset. However, we will compare different approaches to deriving the labels for the data samples (i.e. using opening vs. closing prices as the starting point), which is a problem closely related to the definition of the influence window, as we will show in Section 4.3.2.

This chapter will embrace the first series of experiments that we conducted on our dataset. Despite the fact that most of them can be regarded as preparatory and do not involve much of a model building, the results reported in this chapter are very important for understanding the shape of the data and will directly influence further model extensions (see Chapter 5).

The chapter is structured as follows. In Section 4.1 we define the baseline that we will try to outperform in any case. In Section 4.2 we describe the basic model which is the starting point for our experiments and which we will tune and modify in order to achieve better results. Section 4.3 is devoted to the data alignment issues, which constitute the basis for the future performance of the model and are barely addressed in detail in previous works. In the same section we discuss the problem of data alignment for Mondays, which was also overlooked by most of the researchers. In Section 4.4 we summarize our findings related to the basic model building and the general setup of the experiments.

4.1 The baseline

In all further experiments the model that predicts the most frequent class for all the data samples is considered to be the baseline. The most frequent class as estimated from the training data is the upwards movement (which also includes the cases when the price stays the same). These samples are annotated as positive (i.e. 1) in our data. The baseline is therefore estimated as the percentage of positive labels in the dataset. For the labels derived from the opening [and closing] prices [of the same day], this information can be found in Table 4.1.

Dataset	Positive samples (%)	Negative samples (%)
Train	54.71	45.29
Development	53.85	46.15
Test	59.69	40.31

Table 4.1: The label distribution in training, development and testing data for the labels derived from opening prices

For some of our experiments we will derive the labels for the data samples from the closing prices of two subsequent trading days. As mentioned in Section 3.2, the results of such label assignment are not exactly the same as for the opening prices as the starting point. Therefore, the baseline is also a bit different.

The distribution of the labels derived from closing prices [of two subsequent days] is provided in Table 4.2. The baseline for the experiments conducted with this setting is, once again, estimated as the percentage of positive data samples.

Dataset	Positive samples (%)	Negative samples (%)
Train	54.78	45.22
Development	52.66	47.34
Test	59.69	40.31

Table 4.2: The label distribution in training, development and test data for the labels derived from closing prices

We can see that a difference in the label distribution exists, even though it is not very large. The label distribution in the test data even seems to be the same for both settings. Most importantly, the positive class stays prevailing in all the subsets of the data in both settings. This can be explained by the fact that the economy is generally growing and in long term, the stock prices are expected to go up (which is the theoretical basis for the passive investing strategy).

4.2 The basic model

The basic model that we start with can be described as follows. We consider the news that were published during the calendar day immediately preceding the day for which we make the prediction. For example, for predicting the stock price movement on the 20th of October 2012 we will consider the news that were published on the 19th of October 2012 between 0:00 and 23:59. It is important to note that we do not provide any special treatment for Mondays at this stage, i.e. the textual data for Monday stock price prediction are the news articles published *the calendar day before*, i.e. Sunday. We are mentioning this explicitly since we will actually be exploring other possible options for the Monday data alignment in Section 4.3.3.

All news articles published within the designated period of time are concatenated into one big piece of text that constitutes the main and only input to the model. The processing of the textual data includes cleaning, i.e. removing the information about the reporter and the editor of the article and their contact details as well as replacing the numbers with placeholders (“__NUMBER__”). The text is later tokenized and transformed to a TF-IDF representation. Cleaning also includes removing stop-words.

We use a Support Vector Machine (SVM) classifier with linear kernel and find the best C values using grid search. The basic model achieves an accuracy of **59.17** on the development set (with the baseline of 53.85). Some more specific information on the basic model performance can be found in Table 4.3.

	Negative	Positive
Precision	80.00	57.14
Recall	15.38	96.70
F-score	25.81	71.84
Accuracy	59.17	
F-score (macro)	48.82	
F-score (micro)	59.17	

Table 4.3: The performance of the basic model

We can see that the model already outperforms the baseline, which is good news. The bad news is that the overall accuracy still stays below 60% and the macro-averaged F-score is below 50%. Moreover, the model tends to classify most of the data samples into the majority class (“positive”), which yields high recall for the positive class and moderately high precision for the negative class, but very low recall for the negative class. More exact information about the actual output of the model can be found in the confusion matrix (see Table 4.4).

		Predicted label	
		0	1
True label	0	12	66
	1	3	88

Table 4.4: The confusion matrix for the basic model on the development set

4.3 Data alignment

Our first series of experiments are devoted to the *data alignment*. Under data alignment we understand the mapping of the textual information (i.e. news articles) to a certain stock market trend, i.e. the actual creating of the training and testing data for the machine learning algorithm. In most works this problem is usually addressed only in one way, which is normally presented as self-explanatory. However, we believe that this issue is not that straightforward and a careful choice of the observation window (= the time gap during which the news considered for the prediction are published) may have important consequences for the quality of the prediction.

Having the financial news articles on the one hand and the stock prices information on the other, it is necessary to align the data in such a way that the results provide the best performance of the classifier but stay meaningful from the economical point of view. The latter condition may be satisfied if the observation window (the time period when the news articles in question were published) is located before the prediction (influence) window (the time period for which we predict the stock price movement). It does not make much practical sense to predict the past changes based on the current data, even though such results could have a theoretical value. As we are concentrated on the real-life applications of the stock market prediction, we will attempt to predict the stock price movement in day X based on the information available *before* this movement occurs.

The observation window effects were studied in works conducted in the framework of technical analysis, e.g. in Shynkevich et al. [2014], where the authors investigated the influence of the window size effect for calculating the technical indicators and then used them for one-step-ahead forecasting. We will try to employ a similar approach on the textual data and exploit varying observation window lengths and cut-off times to define the time window that serves best for the prediction of the next day stock index movement.

For the short-term prediction, it seems rational to limit the textual data only to the news that appear during the trading time (see Mittermayer [2004], Schumaker et al. [2012]) – for the obvious reason that the high-granularity ticker data is only available during trading hours and therefore only the articles published during that time can be successfully mapped onto the stock market reaction without any further assumptions. However, such approach is not really common for the lower granularity level prediction (e.g. 1 day), where all the data that was published the day before is usually considered (Ding et al. [2015], Gunduz and Cataltepe [2015]). It would be naive to assume that the former approach can be transferred to the “next day” prediction in a straightforward manner; however, it provides us with the idea that multiple options are actually possible for one-day-ahead prediction and therefore we would like to devote some time to exploring them.

Our hypothesis that the publication time is important for the prediction is supported by the **time of day effect** discovered by Andersen and Bollerslev [1997] for S&P 500 index and further employed by Luss and d’Aspremont [2015]. According to this effect, the news published early during the day and late in the afternoon cause higher returns than the ones published in between. We can conclude from this observation that the news published during the day have different impact on the stock price depending on the time of publication. Therefore, we would like to attempt to find the time interval that would include the news that have a high impact on the price movements and exclude the ones that have a low impact on it.

The **influence window** as defined as above is fixed and constitutes one trading day, which is the setup that we use to provide comparability with the previous research conducted on this dataset. However, we will tweak this setup a bit by considering different possible definitions of trading day. As mentioned in Chapter 3, there are two possible ways of deriving the labels for the data samples based on the available stock market information. We can start counting from the closing price of the previous day or from the opening price of the current day, which corresponds to the two possible influence windows we can be considering for the "next day" prediction. We will explore both options for deriving the labels and comment on the results.

With regard to the **observation window**, we have decided to consider two possible variables for defining it. One of them is the observation window length H and the other is cut-off time T. For all the experiments in this section, we will be considering the news articles that were published during H hours before cut-off time T (i.e., the observation window is defined in the backwards direction from the cut-off time).

We have considered the following cut-off times:

- 9:30 of the current day (the time when the trade starts). This setting allows us to include the news that were published right before the opening of the stock market into consideration.
- 0:00 of the current day. This setting allows us to consider the news that were published during the calendar day before day X (this is the setting used in the basic model).
- 21:00 of the previous day. This and the following setting allow us to eliminate the news published late in the evening when no stock market activity is taking place.
- 18:00 of the previous day. This setting allows us to only consider the news published during the trading time and shortly after the trading ends.
- 15:00 of the previous day. This setting allows us to dismiss the news published during the closing of the trade from the observation window.

And the following lengths of the observation window:

- 9 hours: the shortest observation window. We mostly provide it to demonstrate that after a certain reduction of the duration of the observation window the accuracy starts degrading a lot.
- 12 hours: short-term observation window. Approximately half of the existent textual data is dismissed.
- 18 hours: short-term observation window with most part of the daytime included into consideration.

- 24 hours: mid-term observation window, the option used for the basic model.
- 30 hours: another mid-term observation window.
- 36 hours: extended mid-term observation window.
- 42 hours: long-term observation.
- 48 hours: long-term observation, 2 full days included.

The resulting observation windows will be produced as the intersection of all possible cut-off times with all possible durations of the windows (except for the cases when the resulting observation window is too short and hardly covers any trading time at all, i.e. 9 hours backwards from 9:30 and 0:00 of the current day).

We would also like to mention explicitly that the cut-off times only refer to the 24-hour period immediately preceding the prediction day and are not extended to the preceding days in case of long observation windows that cover more than one calendar day. Therefore, the observation windows that we define in this section are always continuous (in contrast to Section 4.3.3, where we also used some discontinuous observation windows).

We hope that the accuracies provided in different runs will model the answers to the following questions:

- When are the most important news articles published?
- How long does the news effect last?
- How long does it take for the investors to take the new information into consideration?

Another issue that is directly related to the problem of data alignment is selecting the right textual data for Mondays; we will address this question in detail in Section 4.3.3.

4.3.1 Opening price as the starting point

In this section we will provide the results of the experiments with different observation windows and labels derived from opening and closing prices of the same day.

The accuracies of the most successful runs in the setups with different alignments are shown in Table 4.5. For all of these experiments, the accuracy is measured on the development set; the displayed result is the highest one achieved after performing grid search.

The best results by column are highlighted in bold. We can see that there is a certain trend that can be derived from this table. In most of the rows the accuracy is increasing when we shift the cut-off time to an earlier time point (except for the earliest option, 3 p.m.). This means that by eliminating the news published on the evening before the prediction day we actually increase the accuracy in most cases. The best performance for all durations of the observation window

Observation window	Cut-off time				
	9:30 day X	0:00 day X	21:00 day X-1	18:00 day X-1	15:00 day X-1
9 h	–	–	57.40	58.58	58.58
12 h	53.85	56.80	57.99	**62.72**	**60.95**
18 h	55.62	57.39	**60.35**	59.76	58.58
24 h	**56.21**	**59.17**	59.17	60.35	57.40
30 h	**56.21**	58.58	57.98	58.58	58.58
36 h	55.62	56.21	59.76	60.35	58.58
42 h	54.44	58.58	59.76	59.76	56.80
48 h	54.44	56.80	57.40	57.98	56.80

Table 4.5: Highest accuracy achieved on development set in settings with different alignments for the labels derived from opening [and closing] prices [of the same day]

was provided with the 18:00 cut-off time, which filters out exactly those news that were published the evening and the night before the prediction day.

Another thing that we can notice is that short-term and mid-term observation periods generally provide better performance than the long-term ones. All together, we can see that the shorter observation window we are using, the earlier cut-off time will be more effective. This is logical, because if we use a shorter observation window we would want it to contain the most relevant information. On the contrary, if we can allow a larger observation window, we would just want it to include as much information as possible. The most relevant news, as estimated from the previous paragraph, are the ones published between 3 and 6 p.m.

The cut-off time of 9:30 of the current day proved to be completely ineffective, most likely due to the fact that most part of the observation window actually covers the night, when there is not much going on. As well as this, the earliest cut-off time that we considered, 3 p.m. of the previous day, is not nearly as effective as 6 p.m. of the previous day, which serves as a good confirmation for the fact that the news published around the time when the trading stops are one of the most important ones during the day.

Long-term observation windows are not very efficient, which gives us the idea that the news effect wears off quite quickly and the news published two days before do not have a real impact on the stock market during the current day. Also if we compare Table 4.5 to the per-hour news distribution from Section 3.1.3 we can see that when the observation period includes one or more periods of high news frequency (e.g. at around 11 a.m. or 4 p.m.) the accuracy is higher than when it does not include any of them (e.g. 12 hours backwards from 9:30 of the prediction day). This means that the peak times in the news distribution usually provide more information for the prediction than the periods with low amount of news published.

The highest accuracy was achieved for the 12-hour long observation window that ends at 18:00 the day before the prediction day, i.e. 6 a.m. - 6 p.m. the day before. This corresponds well to the information that we have about the publication time of the most important news (before the trading starts and right after it finishes). The observation window in question includes the news articles published right before the beginning of the trade, during the trading time and right after the end of the trading day and yields the highest accuracy of **62.72%**.

Overall, we can see that the observation window is indeed important and tweaking the cut-off times and the length of the window can provide a significant improvement in the performance of the classifier. We can see that the best result that we achieved is 3.55% higher than the one that we got with our basic model (which is also reflected in the table, see 24-hour observation window and 0:00 cut-off time) and exceeds the baseline by more than 8.5%.

4.3.2 Closing price as the starting point

In this section we will provide the results for a similar setup but with a different strategy for deriving the labels. Now we will start counting from the closing price of the previous day, which will give us different labels for approximately 3.5% of all the data samples.

The highest accuracy achieved on the development set after using grid search is displayed in Table 4.6.

Observation window	Cut-off time				
	9:30 day X	0:00 day X	21:00 day X-1	18:00 day X-1	15:00 day X-1
9 h	–	–	56.80	57.40	**59.76**
12 h	52.66	55.62	56.80	**61.54**	59.17
18 h	53.25	56.21	**59.17**	57.40	56.21
24 h	**53.85**	56.80	57.40	59.76	56.21
30 h	52.66	56.80	58.58	58.58	56.80
36 h	53.25	56.21	**59.17**	58.58	56.80
42 h	**53.85**	55.62	58.58	57.40	55.62
48 h	**53.85**	**57.40**	57.40	57.99	54.55

Table 4.6: Highest accuracy achieved on development set in settings with different alignments for the labels derived from closing prices [of two subsequent days]

We can see from this table that the classification results are indeed different from the ones that we observed in the previous section. First of all, let us note that most of the trends that we described for the labels based on the opening prices have become blurred now. The highest achieved accuracy is also lower and constitutes **61.54%**.

However, some observations still hold. First of all, the most successful time frame for the prediction is still the same: 6 a.m. to 6 p.m. the day before. Second, 18:00 cut-off time still provides the best results for most possible durations of the observation window. Last, for most of the rows the accuracy is still increasing from left to right until it reaches the 18:00 cut-off time.

In general, it seems that the dependency between the time frame for the publication of the news articles and the labels derived from closing prices of two subsequent days is less straightforward than for the labels derived from opening and closing prices of the same day. The latter labels appear to be more predictable and form more reliable trends with respect to the different observation windows.

This is surprising at first glance because predicting the labels based on the opening prices involves operating with two unknown variables: the opening price and the closing price of the prediction day. However, when we predict the label which is based on the closing prices of two subsequent days, the starting point is already known, so, in fact, there is less uncertainty.

Our results show, however, that the labels derived from opening and closing prices of the same day are more predictable. The observed difference can probably be explained as follows. When we start counting from the closing price of the previous day, the influence window is actually broader and the price change is also reflecting the events occurring in the evening of the day preceding the prediction day after the end of the trade. This means that, in fact, the influence window starts at 16:00 the day before and most of the observation windows will be partially overlapping with the influence window. Therefore, there is not much time for the market to incorporate the new information into the stock price (in fact, it is actually impossible since the stock market is closed; these events will most likely influence the opening price of the next day but not the direction of the price change during the trading hours on the prediction day).

In fact, when predicting the price change in between two subsequent closing auctions, we need to predict the interaction between two subsequent changes: the change from the closing price of the previous day to the opening price of the prediction day and then the stock price movement during the prediction day (from the opening to the closing price). The former one also appears to be hidden in real time, with its results only reflected as the difference between the opening price of the prediction day and the closing price of the previous day with no intermediate results available to public in the meantime.

This means that when deriving the labels from closing prices we are actually trying to predict a more difficult pattern, because the result of such interaction is not trivial and may depend a lot on the time that it takes for different types of news articles to take effect (which can be varying a lot based on their importance). Some news articles may require immediate action, and in this case they will most likely affect the opening price of the prediction day; for some articles no immediate action needs to be taken and they are more likely to affect the overall price change the next day. Therefore, in this case we are actually trying to predict a two-phase stock price movement ($P_{close_{X-1}} \rightarrow P_{open_X}$ and $P_{open_X} \rightarrow P_{close_X}$), which is obviously a more difficult task since there are multiple possible options of how these two movements can interact with each other.

The existing peculiarities of this particular prediction goal can account for the relatively high results that we got for the early cut-off time of 3 p.m., when we stop observing even before the trading closes. This gives us a small time gap between the observation window and the influence window when some actions can possibly be taken in the stock market. Another unexpected result is the 48-hour observation window, which provided the highest accuracy for the cut-off time of 0:00 even though long observation periods generally proved to be less effective than the shorter ones.

Overall, however, this task appears to be more challenging since it involves the prediction of a two-step movement which is affected by various interacting factors. It also seems that the events that trigger the price change in between two subsequent closing auctions generally need a bit longer time to be incorporated into the price, which means that for the successful prediction of the labels derived from the closing prices we would need either larger observation windows or earlier cut-off times than for the prediction of the labels derived from opening prices.

4.3.3 Special treatment for Mondays

Throughout the previous experiments we treated Mondays just as any other day of the week. This means when we selected the textual data for Monday prediction we took the news published on the day immediately preceding Monday, i.e. Sunday (when using the observation window of 24 hours or less). Whereas this approach already proved to be quite efficient, we would like to investigate whether another type of data alignment could potentially be used for Mondays.

The necessity of special treatment for Mondays can be justified by several observations:

- There is more textual data available that we may consider for Mondays than for any other day that would be non-overlapping with the textual data from the other days. When using observation windows that are under 24 hours, most of the news published throughout the week are covered and used for the prediction. However, Friday and Saturday news usually stay out of sight if we only use the Sunday news for Monday prediction. Monday seems like the most reasonable day to experience the influence of the news published on Fridays and Saturdays, so it makes sense to use the news published on these days to extend the prediction data for Mondays.

- Mondays, unlike all the other days of the week, are not preceded by a working day. This means that the gap between the latest market events and the decisions made on Monday is much bigger for this day than for any other day of the week. Over the weekend the stock market is closed, which gives the investors and other market players more time to contemplate about the happenings of the previous week and to consider the news published throughout the weekend. This basically means that a longer observation period may be required for Mondays.

- The news published over the weekend is also different from the news that is published throughout the week because it reacts to a situation which is fixed and is not changing overtime. On the contrary, the news released on weekdays is published as a response to the stock market events that are occurring in real time. This may lead us to the conclusion that we need to reconsider the cutoff times for the Monday data (= for the weekend news).

Most researchers do not mention Mondays specifically. However, those who do sometimes choose completely opposite approaches with regard to the textual data used for Monday prediction. For example, Rahman et al. [2017] excluded the weekend news completely from the prediction motivating this by the fact that there were much fewer news articles published over the weekend as compared to the weekdays; as a consequence, they were using Friday news for the Monday stock market prediction. On the contrary, Söyland [2015] appended the news articles published on Saturdays and Sundays to the Friday news and used all of them to predict the stock returns on Mondays. Given this observation, it would be interesting to explore and compare several possible options.

We will proceed as follows. First of all, let us prove that Mondays are indeed special and different from the other days of the week and therefore need some special treatment. Let us see whether it is possible to predict the Monday data as effectively as the data from the other days of the week based on the same training data. The setup of the experiment is as follows. We train the classifier on different subsets of the data gathered from the other days of the week and test it on the Monday data and on the data from the day which was not used for the training. We repeat the experiment 4 times to compare Monday to each of the other days.

In order to use as much textual data as possible, we use both the training set and the development set. Whereas Mondays are put apart from the very beginning, the rest of the data (Tuesday - Friday) is split into 4 subsets (1 for each day of the week). The classifier is then trained on three days and tested on Mondays and the 4th day which was not used for the training. This experiment can be regarded as cross-validation among different days of the week with subsequent testing on Mondays.

As estimated from the previous experiments, the best data alignment so far is taking the news published between 6 a.m. and 6 p.m. the day before. We will exploit this data alignment for these experiments. The observed results are gathered in Table 4.7 for the labels derived from opening prices and in Table 4.8 for the labels derived from closing prices.

Train, **Test**				Test accuracy	Monday accuracy
Tue	Wed	Thu	Fri	53.85	50.67
Tue	**Wed**	Thu	Fri	55.62	50.00
Tue	Wed	**Thu**	Fri	54.04	47.99
Tue	Wed	Thu	**Fri**	54.86	49.66

Table 4.7: Cross-validation for the Monday data (labels derived from opening prices)

Train, **Test**				Test accuracy	Monday accuracy
Tue	Wed	Thu	Fri	52.92	50.34
Tue	**Wed**	Thu	Fri	54.10	49.33
Tue	Wed	**Thu**	Fri	52.17	46.64
Tue	Wed	Thu	**Fri**	54.86	51.01

Table 4.8: Cross-validation for the Monday data (labels derived from closing prices)

We can observe that there is a 3% to 6% difference between the accuracy of prediction for Mondays as opposed to the other days of the week. Whereas there is still some variance with respect to how easy it is to predict the data from certain days, the Monday data shows an obvious deviation from the other day-wise subsets. The prediction accuracy for Mondays never exceeds 51%, whereas for all the other days it never goes under 53%. We can conclude from these tables that Mondays are indeed different from the other days of the week and we may need to reconsider the training data we are using for Mondays as well as the data we are using for the other days of the week (e.g., it is possible to exclude the Monday samples from the training to reduce the amount of noise).

Another confirmation for the fact that Mondays are different lies in the fact that the baseline for Monday was actually lower for this set of experiments, which means that we could actually expect higher results for Mondays than for the other days (see Figure 4.1). If all the days of the week belonged to the same distribution, it would be more probable to get a higher accuracy for Mondays, which have a lower baseline. However, despite lower baseline, our experiments show much lower accuracy for Mondays, which is another confirmation for the fact that Mondays are different.

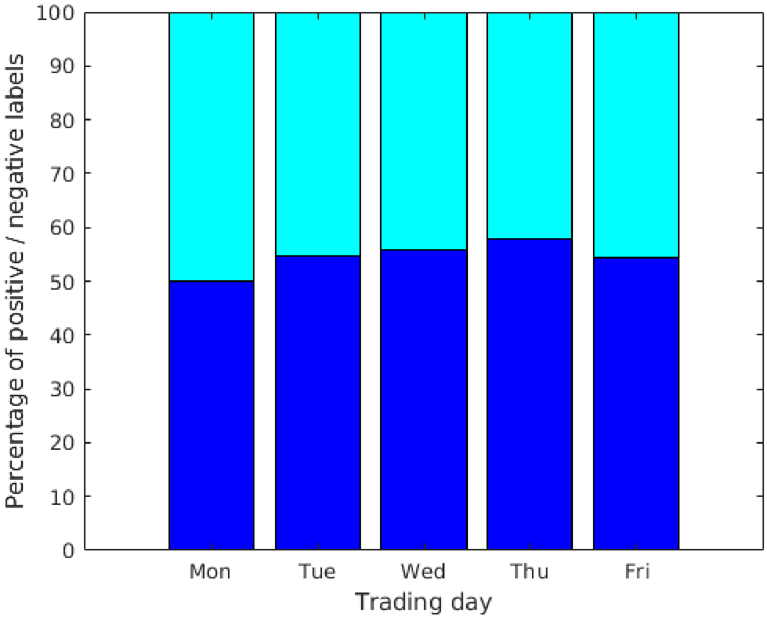

Figure 4.1: Label distribution on different trading days as estimated from training and development data

It is important to note that in all of these experiments we did not use any of the Monday data for the training. As this approach did not turn out to be successful, it seems that we need to incorporate the Monday data into the prediction model in order to be able to predict the labels for Mondays correctly. There are two possible solutions that we can implement.

On the one hand, we can try predicting the Monday data solely from Monday data. On the other hand, we can train on both the data from the other days as well as on the data from Mondays (which was the approach we used before; however, it may need some adjustments to compensate for the peculiarities of the Monday data). In parallel, knowing that the Monday data is special, we also need to consider the best training data for the other days. In particular, we need to find out whether it is effective (or not) to exploit the Monday data when making the predictions for the other days of the week. If so, which textual data exactly do we need to use?

Answering all these questions and implementing both of the solutions implies an explicit distinction between Mondays and the other days of the week, which may lead us to the idea of developing two separate classifiers: one for Mondays and another one for the rest of the week. It may well happen that different training data allows for the best prediction of the labels for Mondays and the rest of the week. We will test whether this approach is generally valid.

Before proceeding to testing the "two classifiers" approach we need to find out the best possible data alignment for Mondays, i.e. the type of textual data that allows to predict the Monday labels the best. We have already mentioned before that there are several possible options that we can consider for Mondays. Investigating these options will help us understand how to incorporate the Monday data in the prediction model in the best possible way.

Data alignment for Mondays

If we decide to train a separate classifier for predicting the Monday labels, we may need to reconsider the existing data alignment for Mondays. It may also turn out to be useful when using the Monday data alongside with the data from the other days: increased quality of the Monday data may affect the prediction accuracy in a positive way.

The data alignment that proved to be the most effective in the previous experiments is using the news published on the day before between 6 a.m. and 6 p.m. For Mondays, this means using Sunday news published between 6:00 and 17:59. However, as mentioned earlier in this chapter, it is quite probable that incorporating the news published on Fridays and Saturdays can be useful for Monday prediction. Only considering the news published on Sundays has some immediate consequences for the number of news used for the prediction (it is significantly lower than for any other day of the week) and, supposedly, for the importance of the news (they tend to sum up the outcomes of the week and are therefore more important). Another thing that we can consider is using different cutoff times for Mondays. Eliminating evening news has less sense for the weekend because for the other days of the week this is our way to avoid the news published outside trading hours; however, on Saturdays and Sundays the stock market is closed anyway.

Considering all the above mentioned, we would like to test the hypothesis whether the existing "default" alignment of the Monday stock price data with the Sunday news actually makes more sense than any other one and to test the performance of alternative alignments that may seem reasonable for this situation. Another possible approach that we have not mentioned yet is eliminating the weekend news completely and predicting the Monday labels from Friday news. We will pay some attention to this option as well.

In the following experiments we are planning to use only the Monday data for the comparison of existing data alignment options. If we use the existing split of the data into training and development sets and extract the Monday samples from it, the resulting data will be highly imbalanced (38% of positive samples in the development set as opposed to 51% positive samples in the training data), therefore we decided to use cross-validation for this subtask.

Despite the fact that predicting the stock prices is a time series problem, we are not actually using any time-dependent information so far, which means that we are not restricted to the splits of the data that use "past" information for training and "future" information for development / testing. This is why we consider using cross-validation for this subtask acceptable.

Let us extract the Monday data samples from the training and development datasets and apply 4-fold cross-validation to them. We have a total of 298 samples, 74 or 75 of which are used for testing throughout different splits. We are going to compare our usual alignment strategy (12 hours before 18:00 on the previous day, i.e. Sunday) with other possible alignment options.

Table 4.9 represents mean accuracies and mean F-scores calculated from the 4-fold cross-validation for the labels derived from opening prices. The baseline for these experiments is 50.01% (the number of positive and negative samples in the Monday subset of the data is approximately the same).

Time frame				Accuracy	F-score
6:00 - 17:59 (discont.)			Sun	**56.71**	**55.07**
		Sat	Sun	55.37	53.87
	Fri	Sat	Sun	53.06	49.27
	Fri			52.05	47.15
Full day (0:00-23:59)			Sun	51.02	49.83
		Sat	Sun	49.34	48.55
	Fri	Sat	Sun	54.73	50.12
	Fri			55.04	49.34
6:00 Fri	-	17:59	Sun	53.08	50.79
6:00 Sat	-	17:59	Sun	53.37	51.47

Table 4.9: Different data alignments for Mondays (for the labels derived from opening prices)

We can get a similar table for the labels derived from closing prices, see Table 4.10. The baseline for these experiments is a bit lower, and namely 49.68%: there are more Mondays that have a downwards trend in the dataset that we are using as estimated from the closing prices.

It can be seen from the tables that we tested several options for the Monday textual data alignment. We tried using existing cut-off times (6 p.m. with 12-hour observation window going backwards) and extending it to Saturdays and Fridays (with discontinuous observation window consisting of 2 12-hour observation periods for Saturday and Sunday or 3 12-hour observation windows for Friday, Saturday and Sunday). We also tried using only the Friday data for the prediction. Apart from that, we tried eliminating the cutoff times and using all the news published over the weekend by using the 24-hour observation window (the one that we used for our basic model, see Section 4.2). Finally, we tried using the continuous observation window starting at 6 a.m on Friday / Saturday and finishing at 6 p.m. on Sunday.

The results are non-trivial. For the labels based on the opening prices, it actually turns out that the "default" data alignment that we were using is the most effective one already. Using the data from Saturdays and Fridays has a negative effect on the quality of prediction. Moreover, we found out that aligning Monday

Time frame				Accuracy	F-score
6:00 - 17:59 (discont.)			Sun	54.40	52.54
		Sat	Sun	**57.41**	**56.17**
	Fri	Sat	Sun	49.68	48.61
	Fri			47.64	45.35
Full day (0:00-23:59)			Sun	47.33	46.34
		Sat	Sun	52.02	51.25
	Fri	Sat	Sun	53.72	52.41
	Fri			54.01	51.72
6:00 Fri	-	17:59	Sun	53.37	51.96
6:00 Sat	-	17:59	Sun	55.39	53.34

Table 4.10: Different data alignments for Mondays (for the labels derived from closing prices)

stock market information with news published on Fridays does not provide good performance either.

For the labels derived from the closing prices the results are a bit different, though consistent with what we know about the functioning of the stock market. When we predict the stock price change based on the opening price, we have one more unknown variable in the prediction model (as opposed to predicting the trend for the closing price of the previous day). This variable is the exact opening price, which is unknown before the opening of the trade when we make our prediction. On the contrary, the closing price of the previous day is known at the prediction moment. Knowing that, it does not seem surprising that for the prediction of the opening price direction we only need the most recent information, whereas for the prediction of the trend of the Friday closing price we need to consider all the events that are happening over the weekend after the stock market closes.

As can be seen from Table 4.10, the highest prediction accuracy can be achieved when using the news published on Saturdays and Sundays during the daytime (6 a.m. - 6 p.m.). Whereas the usage of both of these days can be justified by the fact that we are actually counting the change in the price from Friday evening, the fact that the same cutoff time and observation window length that we were using for the other days of the week appeared to be the most effective one is surprising. In fact, the only difference between the winning approach and the usage of continuous observation window from 6 a.m. on Saturday till 6 p.m. on Sunday is avoiding the Saturday evening news and the news published during the night, which are, apparently, introducing some noise. We can deduct from these observations that the late evening news on the weekends are as unimportant as they are on any other day of the week.

Developing two separate classifiers

Having figured out which textual data is the best fit for Monday stock price prediction, let us proceed with developing two separate classifiers, one of which will predict the labels for Mondays and the other one - the labels for the other days of the week. So far we have mainly discussed the training data that should be used for the Monday classifier training. However, a similar question may be asked with respect to the training data for the other days. Given the assumption that the weekend news are different from the news published over the week and therefore the dependency between the textual data used for the prediction and the prediction target is not the same for Mondays as for the other days, we can come to the conclusion that eliminating Monday data from the training data for the other days may increase the accuracy because Monday data is a potential noise source.

In the following experiments we will test the usage of different training data for the Monday classifier and the Tuesday-Friday classifier. We will first validate this approach on the labels derived from the opening prices. The results of our experiments are shown in Table 4.11. Once again, we used the data alignment that proved to be the most effective in the previous experiments, i.e. we are using the news published the day before within a 12-hour observation window that ends at 18:00. Luckily enough, in the previous section we have proved that the existing data alignment is already optimal for Mondays (news published between 6 a.m. and 6 p.m. on Sundays predict the Monday labels the best). Therefore, our task comes down to comparing the performance of the classifier when testing on the whole data or on its subset. To be on the safe side, however, we also test the performance of the "extended" Monday data ("Mon ext." in Table 4.11), which showed the second best performance in predicting the Monday data in our previous experiments (see Table 4.9). The extended Monday data comprises the news published on Saturdays and Sundays between 6 a.m. and 6 p.m.

Following the concept of developing a separate classifier for Mondays and using only the Monday data for training it, we can also test more elaborate models on the Monday data (the ones featuring bigrams and trigrams). As the Monday data dimensionality is originally much lower than the dimensionality of the data for the whole week (after TF-IDF transformation), we are more likely to experience an improvement when using more complex language models. In Section 5.2 we will show, however, that this approach does not work particularly well if the dictionary is too big and the resulting bigram or trigram feature vectors are very sparse.

Training Monday	Correct labels	Training Tue - Fri	Correct labels
Mon - Fri	**22 / 31**	**Mon - Fri**	**84 / 138**
Mon ext.[1], Tue - Fri	20 / 31	**Mon ext., Tue-Fri**	**84 / 138**
Mon, unigrams	19 / 31	Tue - Fri	81 / 138
Mon, bigrams	19 / 31		
Mon, uni-,bigrams	18 / 31		
Mon, trigrams	20 / 31		
Mon, uni-,bi-,trigrams	18 / 31		

Table 4.11: Separate models for Monday and the rest of the week (labels derived from opening prices)

---

[1] Monday extended data: news published between 6:00 and 17:59 on Saturdays and Sundays

We can observe from Table 4.11 that training the Monday classifier only on the Monday data does not outperform the approach that implies using the data from all the days of the week. The same can be said about the classifier for Tuesday - Friday: excluding the Monday data from the training set decreases the accuracy of the prediction. As well as this, we have shown that the extended Monday data does not perform better than the "default" Monday data (which also follows from our previous experiments). The bigram and trigram models for the Monday data also do not outperform the usage of the data from the whole week. To sum up, the data alignment that we were using before appeared to be the most successful one. When predicting the Monday labels, it turns out to be the most effective to use the data from the whole week. The same holds for the Tuesday to Friday prediction. All together, it is possible to predict 22 + 84 = 106 out of 169 labels correctly, which gives us the accuracy of 62.72 which we reported before. In a nutshell, we have not managed to increase the accuracy by employing different alignment options for Mondays. Moreover, it follows from our results that it does not really make sense for the opening-price-based labels prediction to distinguish between Mondays and other days.

Now that we have tested this approach on the labels derived from the opening prices we can also employ this method for the prediction of the labels derived from the closing prices. The situation with the closing prices appears to be trickier because our previous experiments have shown that we actually need to extend the textual data that we were using for Monday prediction and to append the Saturday data to it.

Just as in the experiments with the opening prices labels, we will try both the "default" data alignment for Mondays that we were using before (6:00 - 17:59 on Sundays) and the "extended" one that we are expecting to perform better based on our results from Table 4.10. Whereas our previous experiments have shown that using the extended version of the textual data (6:00 - 17:59 on Saturdays and Sundays) is more promising for the Monday prediction based solely on Monday data, we also need to know whether it performs better in combination with the data from the other days. As we have already demonstrated that employing more elaborate language models (using n-grams) for the Monday data still does not outperform the approach that implies combining the Monday data with the data from the other days, we are not testing the performance of the n-gram models this time. The results of our experiments can be observed in Table 4.12.

Training Monday	Correct labels	Training Tue - Fri	Correct labels
Mon ext., Tue - Fri	21 / 31	Mon ext., Tue - Fri	81 / 138
Mon, Tue - Fri	**23 / 31**	**Mon, Tue - Fri**	**83 / 138**
Mon ext.	19 / 31	Tue - Fri	81 / 138
Mon	15 / 31		

Table 4.12: Separate models for Monday and the rest of the week (labels derived from closing prices)

Whereas the extended Monday data actually performs much better than the "default" Monday data when used separately, it does not have such good predictive power when combined with the data from the other days of the week. In fact, it turns out once again that the existing data alignment is the best possible one. Using the data from all of the days without any special extensions for Monday gives the best results. However, in this particular case having two separate classifiers may yield a small increase in the overall accuracy. In our previous experiments, when using the same classifier for Mondays and the other days, we achieved the highest accuracy of 61.54, which corresponds to 21 + 83 = 104 correctly classified samples out of 169. We can see from Table 4.12, however, that the highest number of correctly classified samples that one can achieve is 23 + 83 = 106 out of 169, which corresponds to the accuracy of 62.72. This accounts for the fact that the C value that yields the highest performance is not the same for the Monday data and for the rest of the week. This is something we cannot really consider when predicting all the labels at once.

What we have shown by these series of experiments is that despite the fact that Mondays do indeed have some peculiarities, it is still better to use the data from the whole week to predict the Monday labels as well as the labels for the other days of the week. One of the possible explanations for that is the size of the training data: the more data we have, the better. We have shown that the potential noise that it introduces can still be disregarded and incorporating all the available data in the prediction model actually outperforms the approach of using smaller but supposedly "cleaner" data.

We also need to account for the fact that by dismantling the Monday data from the training set for the other days of the week, we actually exclude a lot of negative samples (the ratio between the positive and the negative samples is the lowest for Mondays). This could be one of the reasons why the accuracy drops when we exclude Mondays from the training data for the other days.

4.4 Conclusions and setup modifications

In this chapter we compared different strategies for deriving the labels for the data samples and the performance of the model on different observation windows. Apart from that, we investigated the possible data alignment options for Monday.

The conclusions that we can draw from these experiments are as follows. For both strategies of label assignment, the most successful observation window turned out to be **6 a.m. - 6 p.m.** the day before. The labels derived from closing prices provided a lower performance overall; apart from that, we found out that predicting such labels is a more complex problem since it involves forecasting the interaction between two subsequent price movements.

With regard to the Monday alignment, we have proven that Mondays do **not** need special treatment and the textual data that was used for Monday prediction by default (i.e. the Sunday news articles) already provides the best performance.

Based on these observations, we would like to establish the common setup for all subsequent models that we build in this work. With regard to the textual data, we will only consider the news published between **6 a.m.** and **6 p.m.** on the **calendar day** preceding the prediction day with **no exceptions** (e.g. Mondays). The labels for the prediction will be derived from **opening and closing prices** of the **same** trading day.

5. Model extensions

Now that we have established the setup for the experiments and figured out the best possible data alignment we can proceed with incorporating further features into the model. For all of the models in this section, unless specified explicitly, we will use the data alignment option that proved to be the most effective in the previous chapter, i.e. we will use the observation window of 6 a.m. - 6 p.m. the day before and predict the labels derived from opening and closing prices (see Section 4.4).

One of the most common features associated with the task of stock market prediction is the sentiment. The polarity of the news articles is believed to be helpful when predicting the behaviour of the stock market. It seems reasonable to assume that in most cases positive news would cause an upwards movement of the stock prices, whereas the negative ones would lead to a fall. Various researchers have tried using the sentiment information either as the main representation of the textual data (Mittal and Goel [2012], Vu et al. [2012]) or as an additional feature derived from the text (Schumaker et al. [2012], de Fortuny et al. [2014]). We will explore both of these options in Section 5.1.

In the other sections we will discuss further textual and non-textual features that could be used in the area of stock market prediction. Section 5.2 will be devoted to the extension of the text representation to n-grams. In Section 5.3 we will be talking about Bi-normal separation, an elaborate feature selection technique that could be used for effective dimensionality reduction. In Section 5.4 we will try to incorporate the information about the stock market behaviour into the prediction model. In Section 5.5 we will make an attempt to use only the news titles as an input for our model. In Section 5.6 we will explore the influence of the day-of-the-week feature on the prediction accuracy.

Every feature will be discussed separately and in Chapter 6 we will define several models (some of them including a combination of various features) that will be evaluated on the test data.

5.1 Sentiment

In this section we will describe our attempts at incorporating the sentiment information into the model and comment on the results.

5.1.1 Sentiment dictionaries

One of the most common ways of acquiring sentiment information from the text is using a sentiment dictionary which assigns a positive / negative or some other score to every word in the vocabulary. In this work we will try using two existing sentiment dictionaries, and namely the **financial dictionary** by Loughran and McDonald [2011] and a general purpose sentiment dictionary **SentiWordNet** (Esuli and Sebastiani [2007]).

Financial dictionary

The financial dictionary by Loughran and McDonald [2011][1] was developed for the goal of opinion mining in the financial domain and is therefore well-suited for our task.

We used the edition of the dictionary from the year 2014 which contains 85131 vocabulary items. There are separate entries for different word forms (e.g. there are three separate entries for *worsen*, *worsened* and *worsening*), therefore no additional preprocessing of the tokens is required to apply the dictionary on our dataset.

Every vocabulary item is annotated with the following sentiments: Negative, Positive, Uncertainty, Litigious, Constraining, Superfluous, Interesting. Those scores are basically binary: the cell at the intersection of the word row and the sentiment column contains either a zero or a year number. The year number corresponds to the year when the word in question was added to the dictionary and / or annotated with this sentiment.

Apart from the sentiment information itself, the dictionary contains the information about the modality of the word, where "1" corresponds to a strong modal, "2" indicates a moderate modal and "3" denotes a weak modal. Another sentiment-related information that is available is whether the word is contained in the Harward IV negative word list[2], which is a collection of words with negative sentiment from the areas of psychology and sociology.

Apart from that, every word is annotated with a word count, word proportion, average proportion, standard deviation and document count which are estimated from a large collection of documents. Other annotations include the number of syllables, source and a sequence number, as well as irregularity (for verbs).

This dictionary will be the main one used in most of our experiments since it is domain-specific and also includes diverse sentiment annotation. We will be using all the provided sentiment dimensions (7) as well as the modality, the Harward IV information and the irregular verbs, unless stated otherwise.

To give a better idea of how the financial dictionary is structured, we provide an excerpt from it in Appendix A.5.

SentiWordNet

SentiWordNet is another source for sentiment information that we use in our experiments. Developed as a multi-purpose sentiment dictionary, it is based on the WordNet lexical database. It assigns a positivity score, negativity score and objectivity score to every synset from WordNet.

Each of the scores is a real value between 0.0 and 1.0. and the sum of all the scores is 1. For example, the synset "asset.n.1" (asset, noun, most common meaning) has a positivity score of 0.625, a negativity score of 0.0 and an objectivity score of 0.375.

The entries in this dictionary are synsets, therefore every time we want to calculate the sentiment score for a particular word we need to lemmatize it, assign a POS tag to it and to disambiguate the word sense. We perform the

---

[1] The current version of the dictionary can be found here: `https://sraf.nd.edu/textual-analysis/resources/#Master%20Dictionary`

[2] Currently available here: `http://www.wjh.harvard.edu/~inquirer/Negativ.html`

lemmatization and the POS tagging, however, we believe that introducing word sense disambiguation as an additional step will only increase the noise, therefore the sentiment that we extract always refers to the most common sense of the word (which has number 1).

5.1.2 Calculating the sentiment of a text

Both of these resources allow for estimating the sentiment associated with a separate token but not with the whole text. Having diverse sentiment annotation available, we would like to try to make use of all of it and to not restrict ourselves to the positive / negative scores or their combination. Therefore, in most of our experiments we will try representing the text as a point in a multidimensional space formed by various sentiments. In this case the sentiment score for every dimension can be calculated as the sum of all the respective sentiment scores of all the words contained in this text or as an average over all the sentiment scores. The former option does not sound very reliable since the texts are of different length and the sentiment scores for longer texts will be too much different from the ones for shorter ones. Averaging sounds like a better option, however, different possible strategies can be employed.

For the first series of experiments we will stick to the easiest option of normalizing the sentiment scores by summing up the scores for all the words (sentiment-wise, of course) and then dividing each of the resulting sentiment scores by the total number of words in the respective text. In Section 5.1.5 we will try different normalization options and we will show that they are, in fact, interchangeable.

We would like to mention explicitly that by saying “text”, we actually mean the textual data used for the prediction of the next day stock market movement, i.e. the concatenation of multiple news articles belonging to the time interval we defined in the previous chapter. That is, in most of the experiments we will be calculating the sentiment of the *day* and not the sentiments of separate news articles. In Section 5.1.5 we will show that calculating the sentiment scores for every news article from the observation window separately and then averaging over them gives essentially the same results.

5.1.3 Sentiment as a text representation technique

Our first intention was to append the sentiment information to the TF-IDF representation of the text. However, it may well happen that the sentiment scores by themselves are already a valid representation of the text. In this Section we will test this hypothesis by applying several possible settings:

- The sentiment scores are calculated upon the text and serve as the only representation of the textual data, the TF-IDF information is ignored.

- TF-IDF weighting is only performed for the words conveying sentiment and the rest of the words are omitted. This setting can be regarded as sentiment-based dimensionality reduction.

- TF-IDF weighting is only performed for the words conveying sentiment and the sentiment scores are further appended to the TF-IDF vector.

We can see that all of these settings do not employ the full TF-IDF representation that we used in our previous experiments. This will allow us to better estimate the contribution of the sentiment information to the classifier performance. Moreover, we will be able to understand whether the sentiment scores / the sentiment words or the combination of both can serve as a valid representation of the available textual data. Obviously, the sentiment words are only a subset of all the words used in the text and since there is so much interest towards sentiment in the domain of stock market prediction, it would be interesting to find out whether there is actually any relevant information contained in non-sentiment words.

We will test these settings on the 12-hour observation window which we defined as the most informative in the previous chapter (6 a.m. - 6 p.m. the day before). However, as the results that we are getting are not satisfying, we will also try larger observation windows (24 hours and 36 hours) with the same cut-off time to validate our results. Those two observation windows provided comparable performance, as can be seen in Chapter 4.

The results that we obtained with the abovementioned three settings on different observation windows are listed in Table 5.1. For comparability, we provide the results achieved with a simple TF-IDF representation which does not involve any sentiment at all (these are the models that we tested in the previous chapter). For this table the sentiment scores were estimated from the financial dictionary; the reported results were achieved on the development set.

Observation window	12 h	24 h	36 h
Only sentiment scores[3]	53.85	53.85	53.85
TF-IDF only for words conveying sentiment	56.80	59.76	53.25
TF-IDF for sentiment words + sentiment scores	58.58	59.76	53.25
Basic TF-IDF without sentiment	62.72	60.35	60.35

Table 5.1: Highest accuracy achieved on development set using sentiment-based representations of the text

We can see from this table that this attempt was not successful and the sentiment data does not provide enough information for the classifier to perform well. The model featuring only the sentiment scores performs at the baseline (53.85%), predicting the most frequent class (positive) for most of the data samples.

TF-IDF weighting for sentiment words performs slightly better for the two smaller observation windows but still does not outperform the results achieved without considering any sentiment information and by employing all the words from the text.

Finally, the TF-IDF weighting of the sentiment words together with the sentiment scores allows for a small improvement for the 12-hour observation window but the overall result still stays below the benchmark (62.72% accuracy).

All together, we can see that the sentiment scores / sentiment words cannot be a valid representation of the textual data. However, we can still try using

---

[3] Normalization is performed by dividing the sum of the sentiment scores by the total number of words in the document

the sentiment information together with the TF-IDF representation that we used before and hope for an improvement.

5.1.4 Appending sentiment information to the TF-IDF representation

In this section we will try appending the sentiment scores to the TF-IDF representation of the text that was used in the previous chapter. We will compare the performance of the financial dictionary with the performance of SentiWordNet. Also we will try to decrease the sparsity in the sentiment scores acquired from the financial dictionary by eliminating some of the dimensions or combining several sentiments into one.

The overall setting of the experiments in this section is as follows. We calculate the TF-IDF representation of the textual data from the observation window and calculate the sentiment scores for this text based on the respective sentiment dictionary. The sentiment dictionaries are:

- Complete findict: the full financial dictionary with 12 dimensions: 7 sentiment scores (Negative, Positive, Uncertainty, Litigious, Constraining, Superfluous, Interesting), 3 dimensions for the modality (one-hot encoding for each of the three possible modalities), Harward IV negative word list, irregular verbs. Therefore the feature vector for every data sample will be extended by 12 points after the TF-IDF transformation.

- Findict no harward: the financial dictionary that does not include the Harward IV information, since the latter was reported to be misleading in the financial domain (Loughran and McDonald [2011]). 11 dimensions in total are added to the feature vector representing every data sample.

- Findict extended neg and pos: the financial dictionary with reduced sparsity which contains only negative and positive scores for all the words. The word is assigned a negative score if it is annotated with at least one of the following sentiments: Negative, Uncertainty, Litigious, Constraining, Harward IV. The positive scores stay unchanged. Every feature vector is extended by two dimensions.

- Findict extended neg and pos, no harward: the financial dictionary with reduced sparsity which contains only negative and positive scores for all the words. The word is assigned a negative score if it is annotated with at least one of the following scores: Negative, Uncertainty, Litigious or Constraining, i.e. the Harward IV scores are excluded again. Two dimensions are added to the feature vector.

- SentiWordNet. We are adding three features to the prediction model (positivity, negativity and objectivity scores).

In all of these settings, we sum up the scores for all the words in the text sentiment-wise (positive with positive, negative with negative and so on) and then divide each of the resulting sentiment values by the total number of words in the text. Once again, here “text” refers to the concatenation of the news articles belonging to the observation window. In Section 5.1.5 we will also present some other strategies for normalization and we will show that the normalization strategy is not really important as long as it is present.

Just as before, we test these approaches on the observation window that we established in the previous chapter (6 a.m. - 6 p.m.) as well as on two larger windows (24 and 36 hours) with the same cut-off time (6 p.m.). The accuracies achieved on the development set are reflected in Table 5.2.

Observation window	12 h	24 h	36 h
TF-IDF + complete findict	62.72	60.35	60.35
TF-IDF + findict no harward	62.72	60.35	60.35
TF-IDF + findict extended neg and pos	62.72	60.35	60.35
TF-IDF + findict extended neg and pos, no harward	62.72	60.35	60.35
TF-IDF + SentiWordNet	62.72	60.35	60.35
No sentiment involved	62.72	60.35	60.35

Table 5.2: Appending the sentiment information from different sentiment dictionaries to the TF-IDF representation

We can see that the results reflected in this table are uniform and therefore disappointing. Despite the fact that we have achieved some improvement as opposed to the models reflected in Table 5.1, none of the approaches actually beats the plain TF-IDF representation with no sentiment involved. Moreover, all the sentiment dictionaries behave exactly the same and do not influence the performance of the model in any way. The main "achievement" is that none of the approaches actually lowers the prediction accuracy, which can be the case sometimes, as we will see in Section 5.1.5. All the accuracies that we got stay on the level of the benchmarks achieved in the previous chapter (62.72% for the 12-hour observation window, 60.35% for the 24 and 36-hour observation windows).

We can see that the sparseness of the scores or the type of the sentiment annotation involved does not have any impact on the model. We also cannot say that the sentiment scores "get lost" among the other features because of their low number (they are appended to the feature vectors that are already 200+K positions long) – this is not true since, as we will see in Section 5.4, even a low number of features can influence the performance of the model a lot if they are significant and introduce new information.

5.1.5 Normalizing sentiment scores

In this section we will study the effect of normalization on the performance of the model. We will try different normalization techniques on one of the settings used in the previous sections, namely we will try using the complete financial dictionary and append the sentiment scores to the TF-IDF representation of the textual data. We will test our approach on two observation windows: 12 h backwards from 18:00 the day before and 24 h backwards from 18:00 the day before. As we are using the full version of the financial dictionary, the TF-IDF vectors will be extended by 12 positions in all the cases.

We will try the following types of normalization:

- No normalization. The sentiment scores are summed up and the resulting vector is appended to the TF-IDF vector.
- Length normalization. The sentiment scores for the whole text[4] are divided by the length of the text, i.e. by the number of words in the text. This is the option we used in the previous experiments.
- Sum normalization. The sentiment scores for the whole text are divided by the sum of all the sentiment scores. This is similar to the normalization used in SentiWordNet; all the sentiment scores will sum up to 1 in that case.
- TF-IDF normalization. We extract the sentiment scores from all the documents (for the financial dictionary, the resulting scores will be non-negative integers, i.e. they will basically represent the counts of the words annotated with a certain sentiment) and perform TF-IDF transformation on them. This will allow to increase the importance of less frequent sentiments and to decrease the impact of more widespread ones on the prediction.
- The sentiment scores are calculated for every news article within the observation window separately (with length normalization) and the sentiment score of the day is then calculated as an average over the sentiment scores of all the articles in question.
- There is a single sentiment score calculated for each news article according to the formula $\frac{P-N}{P+N+1}$, where P is the number of positive words and N is the number of negative words in this article. The sentiment of the day is then calculated as an average over the sentiments of separate news articles belonging to the observation window. The formula is taken from Boudoukh et al. [2013] and is similar to the ones used in Twedt and Rees [2012] and Sherif and Leitch [2017].

The effects of different types of normalization are summarized in Table 5.3.

Observation window	12 h	24 h
TF-IDF + sentiment scores (complete findict)		
no normalization	55.03	55.62
length normalization	62.72	60.35
sum normalization	62.72	60.35
TF-IDF normalization	62.72	60.35
sentiment scores for separate articles + averaging	62.72	**60.95**
single sentiment score for every article + averaging	62.72	60.35
No sentiment involved	62.72	60.35

Table 5.3: Different types of normalization when incorporating the sentiment scores into the prediction model

---

[4] The sentiment scores for all the words from the financial dictionary are binary, therefore the sentiment scores for the text are basically counts of words with a particular sentiment.

We can see from this table that whenever some type of normalization is present, the performance of the classifier is not really affected by the sentiment scores and stays at the same level as it was without the sentiment. There is a small exception with regard to the 24-hour observation window, where a small improvement was achieved, however, taking into account the rest of the results in this table, this improvement looks more like a coincidence. Moreover, this is not our main observation window, the improvement is really small and the achieved result is still below the main benchmark of 62.72%.

One of the most important observations that we can derive from this table is that if we do not perform any normalization, the performance degrades a lot and the classifier performs only slightly above the baseline.

5.1.6 Discussion

What we have witnessed in the previous sections is that the sentiment information is not generally capable of improving the classifier performance. Of all the experiments that we have run we have only managed to achieve an improvement in *one* case. Moreover, we have seen that when applied incorrectly, the sentiment scores cause the performance of the classifier to degrade.

The most obvious explanation that we can come up with is that the sentiment scores that we are getting are basically averaged over a large amount of news articles (up to 300 per day) and therefore the outcoming sentiment is very likely to be close to the mean values. The news articles we are using are diverse and, obviously, some of them may contain negative, positive or some other sentiments. However, the news flow on the whole is likely to be close to neutral on every separate day. What is more, financial news articles are definitely not a source of information that contains a lot of subjectivity (or sentiment, which is closely related to it) at all, so in general it would not be reasonable to expect many polar sentiment reactions.

Moreover, we have seen that the sentiment information by itself does not serve as a valid representation of the text. Therefore, it is logical that the sentiment information does not introduce any new separation hyperplanes for the classifier. In fact, if we think about it, the sentiment scores are just another way of weighting the terms in the text, where most of the tokens are actually ignored and the rest are assigned some predefined weights. Given that we already have a pretty elaborate weighting mechanism (TF-IDF), we can argue that adding some more features delivered by a more simple weighting algorithm (sentiment analysis) will not improve the performance of the classifier.

Our results seem to be well in line with some of the discussions in the previous works. For example, de Fortuny et al. [2014] argues that the sentiment results are inconsistent and often underperform even a random classifier. Apart from that, we have to say that all the successful applications of sentiment information that we have seen usually involved using the sentiment of separate news articles and not the overall sentiment of the news flow (Schumaker et al. [2012], Vu et al. [2012], de Fortuny et al. [2014]).

Whenever the researchers tried to calculate the sentiment of the day or to average the sentiment score over a large number of articles/tweets, the process usually involved some non-obvious averaging strategies (see Ding et al. [2013], Sherif and Leitch [2017]). We do not claim that none of them could be used for our task, but we believe we have enough reasons to claim that the ones that we tried do **not** work.

We will take one more additional step to prove that there is indeed not much connection between the sentiment scores and the classification target. To show that, let us calculate the Pearson correlation coefficient between each of the sentiment dimensions and the labels of the data samples. We will perform this operation both for the financial dictionary data (i.e. 12 sentiment indicators) and the SentiWordNet data (3 sentiment scores).

The correlation between the sentiment scores and the target labels for the full version of the financial dictionary is given in Appendix A.6. The correlation coefficients for the SentiWordNet features are given in Table 5.4. In both cases, the coefficients were calculated based on the training dataset with the observation window of 12 hours (6 a.m. - 6 p.m. the day before).

	Negativity	Objectivity	Positivity	**Target**
Negativity	1	0.29453	0.80611	**-0.0019525**
Objectivity		1	0.26191	**-0.02939**
Positivity			1	**0.0111**
Target				**1**

Table 5.4: Correlation between SentiWordNet features and target labels

We will not discuss the correlation coefficients for different sentiments acquired from the financial dictionary in much detail; the only thing that we will mention is that for all of the features the correlation coefficients are extremely low (see Appendix A.6 for more details).

The same holds for the SentiWordNet features: all the correlation coefficients are close to zero. Nevertheless, there is a high correlation between the positivity and negativity score, which can be explained by the limitations on the weights imposed by SentiWordNet (the scores should sum up to 1, which means that all the points that are not attributed to objectivity are distributed between positivity and negativity scores). This is the possible explanation for why the highest correlation between the sentiment scores and the target labels is actually achieved for the objectivity score (nevertheless, its absolute value is still very low).

5.2 N-grams

In this section we will discuss the most simple way of expanding the textual data representation, and namely we will try using the n-grams as the features for the BOW model. In all the previous experiments we were using unigrams as features for the BOW approach; however, using bigrams or trigrams is also a common technique in text processing. Whereas several researchers working in the domain of stock market prediction have tried using bigrams (Hagenau et al. [2013], Rahman et al. [2017], Oliveira et al. [2017]), most of the works employing BOW actually feature unigrams (Mittermayer [2004], Zhai et al. [2007], Groth and Muntermann [2011]). Furthermore, none of the n-gram approaches was previously tested on our dataset, so filling in this gap sounds like a good start.

We will try combining various n-gram features as suggested in Table 5.5. For each of the rows of the table, we used the n-gram features marked as positive as the input to the model and measured the accuracy on the development set. The observation window is 6 a.m. - 6 p.m. the day before (as usual).

Unigrams	**Bigrams**	**Trigrams**	**Accuracy**
+	−	−	**62.72**
−	+	−	56.80
−	−	+	56.21
+	+	−	60.36
−	+	+	56.80
+	+	+	59.76

Table 5.5: The accuracy of the model on the development set when employing n-grams

We can see from this table that, contrary to what we thought, adding n-gram features to the model does not influence its performance in a positive way. The highest performance is still achieved with unigrams; the second and third places are taken by the combinations of unigrams with the n-grams of higher orders. We can see from these results that the unigrams give the highest contribution to the prediction power of the model for our data. We assume that introducing n-gram features of higher order can be helpful when the initial dimensionality of the feature space is low; however, for our data the initial dimensionality of the unigram feature space is already 200+K features, which means that for bigrams or trigrams (or their combination) it will be even higher. Of course, another problem is the sparseness of the bigram / trigram feature space, where most of the n-grams are usually present in a very low number of documents.

We have seen in this section that n-grams by themselves do not really improve the performance of the model; however, a combination of a high-dimensionality feature space and a clever feature selection algorithm can help fight the sparsity of the features and increase the accuracy of the prediction.

5.3 BNS feature selection

One of the methods proposed for the feature selection in the domain of stock market prediction is Bi-normal separation (BNS). This method was reported to outperform Chi-square and Mutual Information and provided unexpectedly high results in conjunction with 2-word combinations (which can be approximated by skipgrams) in Hagenau et al. [2013].

This method was first proposed in Forman [2003], where it was contrasted with other feature selection techniques and provided a significantly better performance in multiple text classification tasks. BNS score for a feature is calculated as follows:

$$\mathrm{BNS} = |\mathrm{F}^{-1}(\mathrm{tpr}) - \mathrm{F}^{-1}(\mathrm{fpr})| \tag{5.1}$$

where tpr is the true positive rate, fpr is the false positive rate and F^{-1} is standard normal distribution's inverse cumulative probability function (a.k.a. z-score). One of the possible views on BNS is that it measures the separation between two standard normal curves which model the prevalence of a feature in the documents of positive and negative classes. The relative position of these two curves is then uniquely prescribed by the true positive rate and the false positive rate, which correspond to the areas under the tail of each curve (see Figure 5.1 and Forman [2003] for more details).

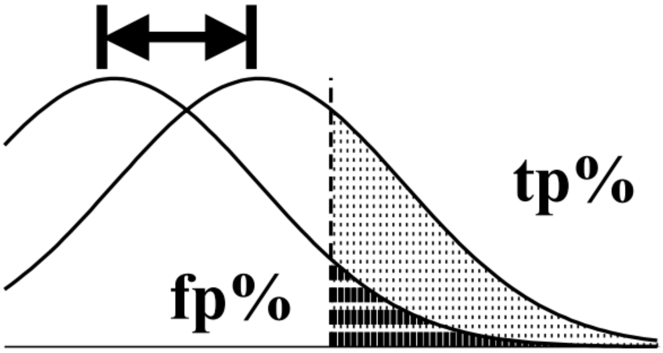

Figure 5.1: One of the possible views on BNS (Forman [2003])

Following the ideas of Hagenau et al. [2013], we tested the BNS feature selection method on the feature spaces formed by different text representation techniques, namely n-grams and skipgrams. Whereas n-grams are combinations of tokens immediately following each other, skipgrams are combinations of tokens that allow for a distance greater than 0 between them. For example, 3-skip-2-gram is a combination of two words that have a distance of up to 3 between them (for the usual bigrams, the distance is 0).

For the n-grams, we tried unigrams, bigrams and trigrams. For the skipgrams, we tried combinations of two words that allow for a maximum distance of 3, 4 and 5 between them (3-skip-2-grams, 4-skip-2-grams and 5-skip-2-grams respectively). For each of the feature types we selected 1K, 10K, 100K or 1M most informative features (the ones with the highest BNS score). The results of our experiments on the development set using the usual observation window of 6 a.m. - 6 p.m. are summarized in Table 5.6.

N best features	1K	10K	100K	1M
unigrams	55.62	59.76	62.13	62.72
bigrams	57.99	60.95	60.95	60.95
trigrams	60.35	57.99	60.95	59.76
3-skip-2-grams	52.07	60.35	63.91	61.54
4-skip-2-grams	55.03	62.72	62.72	63.91
5-skip-2-grams	55.03	61.54	65.09	62.13

Table 5.6: BNS feature selection accuracy on the development set

The results we observe in this table are quite interesting. First of all, let us note that this feature selection method does not give us any improvement for the unigram model. In fact, the dimensionality of the unigram feature space is only 262K, which means that if we select 1M most informative features we are basically selecting all of them. Consequently, the best result for the unigrams is achieved when the maximum possible number of features is selected and it does not outperform the previous results.

The situation is different for the bigrams, for which we can observe an improvement even with a low number of features (1K). If we recall our results from the previous section we will see that the bigram-only model provided an accuracy of 56.80%, whereas here we were able to achieve 57.99% just by selecting the 1K most informative features. By increasing the number of features we can also raise the accuracy till 60.95%. The dimensionality of the bigram feature space for our data is 10M, so reducing it to a lower number seems to be a very reasonable strategy.

The behaviour of trigrams is not stable and depends a lot on the number of selected features. This may account for the sparseness of the trigram feature space.

In accordance with the observations of Hagenau et al. [2013], the highest results are achieved for the skipgrams. The performance of the 4-skip-2-grams is the most predictable, with the accuracy steadily increasing with the growth of the number of selected features. For the 3-skip-2-grams and the 5-skip-2-grams the accuracy is increasing till the number of features reaches 100K and then it goes down. The highest result achieved overall is 65.09% for the 5-skip-2-grams and 100K best features. Here, for the first time, we are able to achieve the same results that were reported by Ding et al. [2015] on the development set.

All together, we see that this feature selection technique does not perform well for the low dimensionality spaces (e.g. the ones formed by unigrams). However, when the complexity and the sparseness of the features increases, this method allows for achieving significantly better results. Taking into account the improvement that we managed to gain, we are planning to include BNS feature selection combined with 5-skip-2-grams in our final model (see Chapter 6).

5.4 Previous day information

Following the ideas of Zhai et al. [2007], Bollen et al. [2011] and Luss and d'Aspremont [2015], we will try to incorporate the historical data into the model. One of the most obvious types of information that we may want to consider is the information about the stock price movements in the days preceding the prediction day.

First, let us decide on the type of information that we will use. We will consider three following types of indicators:

- Target labels for the previous days, i.e. the binary indicators of whether the stock price went up or down on a particular day. These are basically the labels that we derive from opening and closing prices for the samples in our dataset.

- More specific labels for the previous trading days which reflect not only the direction of the movement, but also its intensity. We will employ the strategy of assigning one of the 5 labels (-2, -1, 0, 1, 2) to each trading day where the "intense" (-2, +2) labels serve for the annotation of the days for which the change in the price exceeded 1%, which is a type of change that the S&P 500 index experiences quite rarely.
- Exact percentage of the price change calculated as $\frac{P_{close_X} - P_{open_X}}{P_{open_X}} \cdot 100\%$

We will test these three different types of information on a sample observation period of three days. We will be appending a feature vector containing the data about the stock price change during the three trading days preceding the prediction day to the usual TF-IDF representation. That is, if we denote the prediction day as X then we will be including the data about the stock price movement on the **trading** days $X-1$, $X-2$ and $X-3$. Just as before, we will be using the observation period of 12 hours for the textual data, i.e. 6 a.m. to 6 p.m. on **calendar** day $X-1$. Table 5.7 contains the highest accuracy achieved on the development set when incorporating different types of information about the price change in the previous days.

Previous day information	Accuracy
Binary	**63.91**
Specific	**63.91**
Exact	62.72
None	62.72

Table 5.7: Highest accuracy achieved on development set when incorporating different types of information about stock price movements from 3 days preceding the prediction day

We can see from this table that the exact price movements (in percentages) do not improve the accuracy of the prediction, however, the less sparse representations of the same information (i.e., the binary and the specific labels for the previous trading days) perform much better and give us the highest accuracy of **63.91%**. As the results are the same for these two types of information, we decided to continue with the more precise one (5 possible labels) in our subsequent experiments. Thus, we will be able to employ more specific information but the ability of the classifier to generalize will not be lost.

Having decided on the type of the information that we want to incorporate, we can experiment with the number of days from which we gather this information. In the previous experiment we used three days as the observation window, however, it may well turn out that some other number of days works better.

Once again, we would like to mention explicitly that the number of days that we are talking about refers only to the stock market information that we use; the textual data that we employ for the prediction does not change for these experiments and always contains the news published on calendar day $X-1$, which does not even have to be a trading day (in case we are predicting the stock price movement for Monday). Similarly to our previous experiments, we will be appending the information about the stock price movements during the previous trading days to the TF-IDF representation of the textual data from our usual observation window. For the previous day stock market information, we will try 1, 2, 3, 4, 5, 7 and 10 trading days as observation windows. The results of our experiments are reflected in Table 5.8.

Days	Accuracy
1 day	62.72
2 days	63.31
3 days	63.91
4 days	**65.68**
5 days	62.13
7 days	62.13
10 days	61.54
None	62.72

Table 5.8: Highest accuracy achieved on development set when incorporating the stock price movement (specific) information from varying number of days

We can see from this table that the accuracy is growing when we increase the number of days from 1 to 4 and decreases afterwards. The highest accuracy we managed to achieve is **65.68%**, which is higher than any other result reported for the development set on this data.

The observed effects can be explained as follows. If the number of days is too high, there are too many possible combinations of the labels for the previous days and it is not possible to identify any patterns in the data. However, if the number of days is lower, the delivered information may be significant. Our hypothesis is that 4 days is exactly the time period that may contain some trends that may help predict the stock market. For example, together with the prediction day they cover one trading week, which may probably help predict the target label on Fridays (if the labels for Monday-Thursday are known). Moreover, some patterns such as “head and shoulders” are also likely to be identified within a similar number of days.

Seeing that the information about the previous days provides such a great improvement in terms of the accuracy of the prediction, it would be reasonable to ask ourselves how well will the classifier perform given only the information about the stock price changes in the previous days and not given the textual data from the day preceding the prediction day. We tried answering this question for the observation periods of 4 and 10 days and we found out that the classifier performs at the baseline (53.85%) in both cases. Therefore, the historical information of this type only performs well together with the textual data.

Together with BNS feature selection, the previous day information also sounds like a good candidate to be included in our final model (see Chapter 6).

5.5 News titles

As suggested in de Fortuny et al. [2014], Ding et al. [2014] and Vargas et al. [2017], we tried using only the titles of the news for the prediction. The approach is exactly the same as before: we concatenate the titles of the news belonging to the observation window with each other and use them as the input to the model.

Using the same observation window (6 a.m. - 6 p.m. the day before) did not give satisfactory results; the highest accuracy achieved constituted 56.62%, which is much lower than the respective result for the full data.

The mentioned works used different data representation techniques (structured events for Ding et al. [2014], sentence embeddings for Vargas et al. [2017]) or were trying to solve a different task on a different dataset (short-term prediction based on single news articles and their sentiment in de Fortuny et al. [2014]), which may explain the low performance of our system. Overall, it seems that news titles do not provide enough information unless some elaborate data transformations are performed.

For the approach that we are using, full text of the article definitely provides more available information and allows for a better prediction quality.

5.6 Day of the week

Having conducted all the Monday-related experiments in Section 4.3.3, we did not want to give up the idea of including the information about the day of the week in the prediction model. This sounded logical due to the fact that several stock market effects are known to take place during specific days of the week. For example, the “Monday effect” states that the stock returns on Monday will normally follow the prevailing trend that took place on Friday.

We tried incorporating the information about the day of the week (Monday - Friday) for which we are making the prediction into the model. We used one-hot encoding to represent the categorical day-of-the-week variable and appended the resulting vectors to to the TF-IDF representation of the textual data. The results were not satisfying because the accuracy went down till 61.54%, which is below the benchmark of 62.72%. We can conclude from these results that this information is, apparently, not very significant and introduces noise for the classifier.

If we use binary data representation (Monday vs. non-Monday), the performance of the classifier stays on the same level as without the day-of-the-week information (62.72% accuracy), so in this case this variable does not seem to affect the prediction quality either.

Day of the week was one of the “outer world” types of information that could have had a possible effect on the stock market; however, it seems from our experiments that even if some of these factors can affect the stock market, day of the week is probably not one of them. However, discovering such influence in a long-term perspective would be surprising, so knowing that the stock price movements do not really depend on the day of the week would probably be relieving for most of the investors.

6. Final models

In the previous chapters we discussed the settings for the evaluation and some of the features that can be used to extend the model. In this chapter we will test the models that provided the best performance on the test set and comment on the results.

The chapter will be structured as follows. In Section 6.1 we will describe two new models that would include different combinations of the features that performed well on the development set. In Section 6.2 we will list the models that we are going to evaluate on the test set and provide the performance that they demonstrated. In Section 6.3 we will discuss the results that we got and suggest some modifications to the models to understand the low scores that we obtained. In Section 6.4 we will summarize the outcomes of our experiments on the test set.

6.1 Combining all our findings

From Chapter 5 we know that there were two types of features that showed high performance on the development set, and namely BNS feature selection with skipgrams (see section 5.3) and previous day stock market information (see section 5.4). In this section we will try combining these two types of features and testing the resulting models first on the development set and then on the test set.

We will combine the specific labels for the 4 previous trading days with the two best skipgram-based models (with BNS feature selection) from section 5.3:

- 5-skip-2-grams, 100K best features (Combined model 1)
- 4-skip-2-grams, 1M best features (Combined model 2).

The full specifications for the two combined models can be found in Table 6.1.

<table>
<tr><td>Observation window</td><td colspan="2">6 a.m. - 6 p.m. day X-1</td></tr>
<tr><td>Influence window</td><td colspan="2">Trading day X</td></tr>
<tr><td>Prediction target</td><td colspan="2">$\mathbb{1}\,(\frac{P_{close_X}-P_{open_X}}{P_{open_X}} \geq 0)$</td></tr>
<tr><td>Special treatment for Mondays</td><td colspan="2">None</td></tr>
<tr><td>Text representation</td><td>5-skip-2-grams</td><td>4-skip-2-grams</td></tr>
<tr><td>Feature selection</td><td>BNS, 100K features</td><td>BNS, 1M features</td></tr>
<tr><td>Sentiment</td><td colspan="2">None</td></tr>
<tr><td>Previous day information</td><td colspan="2">4 days, specific labels</td></tr>
<tr><td>Day of the week information</td><td colspan="2">None</td></tr>
<tr><td>Titles vs. full articles</td><td colspan="2">Full articles</td></tr>
</table>

Table 6.1: Specifications for the combined models

We will first try running these models on the development set because we have never actually tested the performance of the multi-feature models on this data. The highest accuracy that we managed to achieve on the development set with

Combined model 1 is **62.13%**, which is significantly lower than the results that we obtained when adding only single features to the model (BNS feature selection with skipgrams OR previous day information). Combined model 2 showed a slightly better performance on the development set, and namely **63.31%**. However, this is still lower than the results that each of the employed features demonstrated separately. Therefore we decided that it would be reasonable to test the single-feature models on the test set as well since the results that they provided on the development set were much higher.

6.2 Final models and their performance on the test set

Based on our experiments in Chapter 5, we can say that our best models were:

- 5-skip-2-grams with BNS feature selection (100K best features)
- TF-IDF representation combined with the stock market information from 4 previous days (specific labels)

Together with the two combined models from section 6.1, that gives us 4 models that we can evaluate on the test set.

The specifications for these models as well as the C values that proved to be the best on the development set are given in Table 6.2. To see whether there is any improvement as opposed to the less elaborate models, we have also decided to test the performance of the basic model (see section 4.2) and the enhanced basic model that we developed at the end of chapter 4 (see section 4.4).

Model	**C$_{best}$(dev)**	**Dev. set accuracy**
Basic model (Section 4.2)	14	59.17
Enhanced basic model (Section 4.4)	24	62.72
5-skip-2-grams + BNS (100K best features)	26	65.09
TF-IDF + 4 prev. days info	20	65.68
Combined model 1	18	62.13
Combined model 2	12	63.31
Baseline		53.85

Table 6.2: Models that will be tested on the test set

The results of the experiments on the test set are reflected in Table 6.3. To get these results, we used the C values that provided the best performance on the development set and trained the classifier on the training set (i.e., we did not include the development set into the training data).

Model	**C_{best}(dev)**	**Test set accuracy**
Basic model (Section 4.2)	14	59.16
Enhanced basic model (Section 4.4)	24	57.59
5-skip-2-grams + BNS (100K best features)	26	58.64
TF-IDF + 4 prev. days info	20	57.59
Combined model 1	18	59.16
Combined model 2	12	53.40
Baseline		59.69

Table 6.3: Performance of the models on the test set

We can see from this table that all of the models performed *below the baseline*. We have to note that the baseline for the test set is much higher than for the development set (59.69 vs. 53.85), however, this is not the main reason why we are observing such results. In the next section we will try to figure out what happened.

6.3 Discussion and error analysis

We have witnessed in the previous section that all of our models actually failed to show any good results on the test set. In this section we will test several hypotheses why this could have happened.

6.3.1 C value

We can see from Tables 6.2 and 6.3 that the C values that we employed for the classifier are rather high, which could have led to overfitting. Let us try setting the C parameter to a lower value and see what happens. We will try running the same models but with different C values: we will try C = 1, 10, 20 and $C_{best}(dev)$. Table 6.4 shows the best accuracy achieved on the test set and the C value that led to this performance.

Model	**C**	**Test set accuracy**
Basic model (Section 4.2)	1	59.69
Enhanced basic model (Section 4.4)	1	59.69
5-skip-2-grams + BNS (100K best features)	10	**60.21**
TF-IDF + 4 prev. days info	1	59.69
Combined model 1	1	59.69
Combined model 2	1	59.69
Baseline		59.69

Table 6.4: Performance of the models on the test set with adjusted C values

We can see that setting the C value to 1 leads to the baseline-level performance of all the models. However, if we have a look at the confusion matrix for all these cases, we will see that the classifier is basically predicting the most common class for all the samples (see Table 6.5).

		Predicted label	
		0	1
True label	0	0	77
	1	0	114

Table 6.5: The confusion matrix for the models performing at the baseline on the test set

We can see from these experiments that even if the high C value leads to overfitting, we cannot really fix this by setting it to some default values because the resulting model fails to generalize anyway and is performing as a majority class classifier. One of the exceptions is the skipgram-based model, which performs above the baseline with C = 10. However, this model is still very biased towards the positive class and the improvement in its performance as compared to the baseline is almost negligible.

6.3.2 Balancing out the training data

We have seen that all the models that we developed are very biased towards the positive class. We can try reducing the influence of this factor by balancing out the training data, i.e. to provide the classifier with the same amount of positive and negative data samples. As there are more positive samples in the training data, we will randomly remove some of the positive samples from the training data so that the ratio between the positive and negative classes would be exactly 50:50.

We will test the performance of our models on the test data with C = 1 and C = $C_{best}(dev)$. Table 6.6 reflects the highest accuracy that we achieved and the C values that we used for the classifier.

We have to note, however, that using the $C_{best}(dev)$ is not fully legit here as the model is a bit different due to the changes in the training data. However, it will give us an idea of how the model performs with a non-default C value.

We can see that now all the models actually perform way below the baseline with one unexpected exception of the basic model, which shows above-baseline performance for the default value of the C parameter. Overall, the results in this table are mostly below 50% accuracy, which tells us that being deprived of the information about the majority class, the classifier fails to find any connection between the data samples and the target labels.

6.3.3 Extending the training data

We can make one more attempt of modifying the training data, and namely extend it with the development set. More training data is always better, so we

Model	**C$_{best}$(dev)**	**Test Acc.**	**C**	**Test Acc.**
Basic model (Section 4.2)	14	48.17	1	**60.21**
Enhanced basic model (Section 4.4)	24	47.64	1	40.31
5-skip-2-grams + BNS (100K best feats.)	26	55.50	1	40.84
TF-IDF + 4 prev. days info	20	42.93	1	45.03
Combined model 1	18	51.83	1	40.84
Combined model 2	12	52.36	1	40.31
Baseline				**59.69**

Table 6.6: Performance of the models on the test set after training on the balanced training data

may hope that it gives us some improvement. We tried running all our models on the test set with C = 1, 10, 20 and $C_{best}(dev)$. Table 6.7 gives us the accuracy of the models and the C values that led to this performance.

Model	**C**	**Test set accuracy**
Basic model (Section 4.2)	1	59.69
Enhanced basic model (Section 4.4)	1	59.69
5-skip-2-grams + BNS (100K best features)	1	59.69
TF-IDF + 4 prev. days info	1	59.69
Combined model 1	1	59.69
Combined model 2	1	59.69
Baseline	–	59.69

Table 6.7: Performance of the models on the test set after extending the training data

We can see that the best performance all the models can give us is the baseline-level performance which is reached with the default value of the C parameter. Just as in Section 6.3.1, this performance is achieved when all the samples are classified into the majority (positive) class, which is, of course, not the behaviour that we wanted to get.

6.3.4 Balancing out the testing data

Having conducted all these experiments with changing the C value and modifying the training data, we are likely to come to the idea that the test data that we are using is somewhat different from what we observed in the training set and from what we were expecting to see based on the development set.

The easiest modification that we can apply to the testing data is to bring the label distribution to the 50:50 ratio. One of the possible reasons why all our models perform so badly is because the baseline is too high and we cannot beat it. Let us lower the baseline by eliminating some of the positive samples from the test set and making the number of positive and negative samples in the test data equal.

The performance of the models on the modified test set is given in Table 6.8.

Model	**C$_{best}$(dev)**[1]	**Test set accuracy**
Basic model (Section 4.2)	14	51.30
Enhanced basic model (Section 4.4)	24	51.95
5-skip-2-grams + BNS (100K best features)	26	51.30
TF-IDF + 4 prev. days info	20	51.30
Combined model 1	18	51.95
Combined model 2	12	50.65[2]
Baseline	–	**50.00**

Table 6.8: Performance of the models on the balanced test set

The good news is, all our models now perform above the baseline. The bad news is, there is no actual difference between any of the models and the simple ones perform just at the same level as the more elaborate ones. Moreover, there is still a strong connection between the specific values of the C parameter and the overall performance of the model (see the footnote for Combined model 2: the performance of the model increased by almost 4% when we employed a different C value). Also we can see that the exceedance of the baseline level for the test set (around 2%) is not at all comparable to the one that we got on the development set (up to 12% improvement), which shows that the successful results are possible, but almost random.

6.3.5 Mixing development and test data

We have seen from the previous sections that all the models that performed well on the development data failed on the test set or showed a much lower performance. Now we have to ask ourselves whether there is anything special about the test set or the development set that could lead to such results.

We have clearly seen in Chapters 4 and 5 that there were some patterns that we could recognize in the development data that led to an improvement in the performance of the classifier. However, when we try transferring these findings to a different dataset (test data), it turns out that our model fails to generalize at all.

We will perform one more series of experiments to see to which extent the performance of the models depends on the test data that we use. We will merge the development and the test set together and then randomly split them into two independent subsets. We will then test the performance of the models on each

---

[1] C = 1 always gives 50% accuracy.

[2] With C = 10, Acc = 54.55

of these newly generated test sets. Each of the resulting sets should contain approximately half of the samples from the development set and half of the samples from the test set.

The total number of the data samples in development and test sets is 360, so each of our new test sets will consist of 180 samples. We will repeat the splitting procedure twice. Sets 1.1. and 1.2 represent the first split; sets 2.1 and 2.2 correspond to the second split. The results of our experiments are reflected in Table 6.9.

Model	**C**	**Set 1.1 Acc.**	**Set 1.2 Acc.**	**Set 2.1 Acc.**	**Set 2.2 Acc.**
Basic model (Section 4.2)	14	55.55	**63.33**	**60.56**	57.78
Enhanced BM (Section 4.4)	24	56.67	**63.33**	**62.22**	57.78
5-skip-2-grams + BNS (100K)	26	58.33	**65.00**	**65.00**	58.33
TF-IDF + 4 prev. days info	20	58.33	**64.44**	**62.22**	60.56
Combined model 1	18	57.22	**64.44**	**62.22**	59.44
Combined model 2	12	57.78	**58.33**	**59.44**	56.67
Baseline		**57.22**[3]	**56.67**[4]	**57.22**	**56.67**

Table 6.9: Performance of the models on the test set with extended training data

The results we can observe are interesting. First of all, let us note that now all non-basic models perform at the baseline or higher for all the splits. Second, all the models except for the combined model 2 show an improvement compared to the basic model and the enhanced basic model (or at least perform at the same level). Another trend that can clearly be seen from this table is that there are "good" (1.2 and 2.1) and "bad" (1.1 and 2.2) splits. For the "good" splits, the performance is almost as high as the one that we reported on the development data. For the "bad" ones, however, the performance is significantly lower. The difference in the performance on the different splits of the data reaches 7.5%, which is an unreasonably high value for our task. Such variance between the performance of the models on different data is barely acceptable if we want to build a model that would be transferable to new data.

6.4 Final remarks

We have seen in this chapter and the previous ones that the models that we built performed really well on the development set (recall that for the model featuring the stock market information we actually achieved an accuracy that was higher than the ones previously reported for this dataset). However, transferring these models to the test set was not successful.

We have conducted several experiments to figure out what could have caused such results. It seems that, generally speaking, it is possible to develop a model

---

[3] 103 out of 180 samples are positive

[4] 102 out of 180 samples are positive

that would perform well on a particular dataset; however, the generalization abilities of such model are questionable.

The relatively high results that we were getting on the development set actually show that there are some patterns in the data that could be used to predict the stock market trends; however, these patterns appear to be non-universal and are barely transferable to other datasets.

The overall conclusion that we can make is that all the methods that we have tried can work on certain data if we tailor the parameters of the model (e.g. the C value, the number of best features that we select with BNS or the number of previous days that we consider when gathering the stock market information). However, the optimal values for these parameters appear to be strongly correlated with the actual data used, therefore it is almost impossible to build a system that would perform equally well on all types of data.

For comparison, you can check the performance of some further models on the test set with adjusted C values in Appendix A.4. We can see that it is generally possible to beat the baseline on the test set and to achieve the results better than the ones we have demonstrated in the previous sections; however, none of the models presented in Appendix A.4 showed exceptional results on the development set, which means that this success is sporadic and no reliable assumptions about the performance of particular models on the given data can be made.

Conclusion

This work was devoted to predicting stock market trends from news articles. We attempted to understand the shape of the data better and conducted several experiments to establish the best settings for solving the task of S&P 500 index forecasting. Our experiments on the development set have shown that the observation window of 6 a.m. - 6 p.m. on the day $X - 1$ serves best for the prediction of the upwards / downwards movement of the index on the day X. We also compared different strategies for deriving the labels for the data samples and we established the common evaluation settings for all our further experiments. These settings can also be used in the future works conducted in the domain of stock market prediction.

After developing a basic model and modifying it according to our findings connected to the data alignment, Monday specifics and label assignment, we experimented with adding additional features to the model. It turned out that the sentiment of the news flow as well as the day of the week do not help predicting the stock market. On the other hand, including the information about the stock price movements in the previous days does help. More elaborate text representation techniques (i.e. skipgrams) combined with BNS feature selection also brought some good results on the development set.

However, it turned out that our findings are barely transferable to the test set. All the models that we developed had baseline performance, or lower. We have conducted some more experiments to understand the reasons for this failure and it basically turned out that all the settings that provided good performance on the development set failed on the test set. Each of the features that we considered does yield some improvement but the extent of this improvement depends largely on the given data. Whereas we were able to identify some trends and some patterns in the development data, these assumptions proved unreliable on a previously unseen data. The variance in the performance of the models on different data appears to be too high.

Coming back to what we started from, we can say that the task of stock market prediction is indeed very challenging. We have shown that for some data it is possible to make predictions that are better than random; however, such findings are not likely to work well on the different data.

The overall explanation for this can lie in the complexity of the task and the unpredictability of the stock market functioning in general. There is no economical theory that could fully explain all the effects of a particular event on the stock market; modeling such influences is the task that the current economical theory has not yet managed to solve efficiently. Also, as we remember, the Random Walk Theory states that the stock prices are changing at random; we have witnessed some of the confirmations for this statement since we were not able to identify any trends that would work on any given data.

Moreover, in our case the events potentially influencing the stock market were displayed through news articles, which adds another source of distortion to the overall picture. All together, the connection between the news articles and the stock market behaviour appears to be very vague, if present in general. Modeling such dependency from the theoretical perspective is also a task that was not fully solved yet; establishing the theoretical foundations for it could make a work of thousands of pages. However, after conducting this research we believe that it is probably not worth the effort because the amount of labour required for that will probably not be outweighed by the potential profits.

Despite the fact that some works conducted in the domain of stock market prediction did show some good results, we are not aware whether the developed models can successfully be transferred to the other data. In fact, the existing research shows quite the opposite, because whenever some researchers attempt to replicate somebody else's approach on their newly gathered dataset, their own model always performs significantly better than the model of their antecedents (see Ding et al. [2015] and their attempt to replicate the work by Luss and d'Aspremont [2015]). An even better example can be provided by Mittal and Goel [2012], who developed a model based on the ideas of Bollen et al. [2011] but could only achieve an accuracy that was 11.5% lower than the one that was reported by Bollen et al. [2011] on their data. Also if we have a look at the experiments of Hagenau et al. [2013] we will see that transferring a model to a new dataset inevitably leads to a huge decrease in terms of accuracy; however, we have to admit that in the abovementioned work they nevertheless managed to stay above the baseline.

In a way, we can say that our approach was too simplistic for the given problem; we definitely were not considering many of the possible influences on the stock market and were mostly concentrating on the textual data with some addition of technical information (the stock market data from the previous days). However, as already mentioned, these processes are not yet fully modeled in theory, so we did our best trying to use the sources of information that were available and exploring the dependencies that we could identify in the given data.

Bibliography

Torben G Andersen and Tim Bollerslev. Intraday periodicity and volatility persistence in financial markets. *Journal of empirical finance*, 4(2-3):115–158, 1997.

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

Jacob Boudoukh, Ronen Feldman, Shimon Kogan, and Matthew Richardson. Which news moves stock prices? A textual analysis. Technical report, National Bureau of Economic Research, 2013.

Enric Junqué de Fortuny, Tom de Smedt, David Martens, and Walter Daelemans. Evaluating and understanding text-based stock price prediction models. *Information Processing & Management*, 50(2):426–441, 2014.

Tina Ding, Vanessa Fang, and Daniel Zuo. Stock market prediction based on time series data and market sentiment. *URL http://murphy.wot.eecs.northwestern.edu/~pzu918/EECS349/final_dZuo_tDing_vFang.pdf*, 2013.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1415–1425, 2014.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Ijcai*, pages 2327–2333, 2015.

Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: a high-coverage lexical resource for opinion mining. *Evaluation*, 17:1–26, 2007.

Eugene F Fama. The behavior of stock-market prices. *The journal of Business*, 38(1):34–105, 1965.

George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

Gyozo Gidofalvi and Charles Elkan. Using news articles to predict stock price movements. *Department of Computer Science and Engineering, University of California, San Diego*, 2001.

Eric Gilbert and Karrie Karahalios. Widespread worry and the stock market. In *ICWSM*, pages 59–65, 2010.

Sven S Groth and Jan Muntermann. An intraday market risk management approach based on textual analysis. *Decision Support Systems*, 50(4):680–691, 2011.

Hakan Gunduz and Zehra Cataltepe. Borsa istanbul (BIST) daily prediction using financial news and balanced feature selection. *Expert Systems with Applications*, 42(22):9001–9011, 2015.

Michael Hagenau, Michael Liebmann, and Dirk Neumann. Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems*, 55(3):685–697, 2013.

Moshe Koppel and Itai Shtrimberg. Good news or bad news? Let the market decide. In *Computing attitude and affect in text: Theory and applications*, pages 297–301. Springer, 2006.

B Shravan Kumar and Vadlamani Ravi. A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, 114:128–147, 2016.

Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Language models for financial news recommendation. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 389–396. ACM, 2000.

Tim Loughran and Bill McDonald. When is a liability not a liability? Textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

Ronny Luss and Alexandre d'Aspremont. Predicting abnormal returns from news using text classification. *Quantitative Finance*, 15(6):999–1012, 2015.

Burton Gordon Malkiel. *A random walk down Wall Street.* W. W. Norton, 1973.

Anshul Mittal and Arpit Goel. Stock prediction using twitter sentiment analysis. *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)*, 15, 2012.

M-A Mittermayer. Forecasting intraday stock price trends with text mining techniques. In *Proceedings of the 37th annual Hawaii international conference on system sciences, 2004*, pages 10–pp. IEEE, 2004.

Marc-Andre Mittermayer and Gerhard F Knolmayer. Newscats: A news categorization and trading system. In *ICDM'06. Sixth International Conference on Data Mining, 2006.*, pages 1002–1007. IEEE, 2006.

Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. Text mining for market prediction: A systematic review. *Expert Systems with Applications*, 41(16):7653–7670, 2014.

P S M Nizer and Julio C Nievola. Predicting published news effect in the Brazilian stock market. *Expert Systems with Applications*, 39(12):10674–10680, 2012.

Nuno Oliveira, Paulo Cortez, and Nelson Areal. The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73:125–144, 2017.

Gil Rachlin, Mark Last, Dima Alberg, and Abraham Kandel. Admiral: A data mining based financial trading system. In *IEEE Symposium on Computational Intelligence and Data Mining, 2007. (CIDM 2007.)*, pages 720–725. IEEE, 2007.

Asyraf Safwan Ab Rahman, Shuzlina Abdul-Rahman, and Sofianita Mutalib. Mining textual terms for stock market prediction analysis using financial news. In *International Conference on Soft Computing in Data Science*, pages 293–305. Springer, 2017.

Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009a.

Robert P Schumaker and Hsinchun Chen. A quantitative stock prediction system based on financial news. *Information Processing & Management*, 45(5):571–583, 2009b.

Robert P Schumaker, Yulei Zhang, and Chun-Neng Huang. Sentiment analysis of financial news articles. In *20th Annual Conference of International Information Management Association*, 2009.

Robert P Schumaker, Yulei Zhang, Chun-Neng Huang, and Hsinchun Chen. Evaluating sentiment in financial news articles. *Decision Support Systems*, 53(3): 458–464, 2012.

Jia-Lang Seng and Hsiao-Fang Yang. The association between stock price volatility and financial news – a sentiment analysis approach. *Kybernetes*, 46(8): 1341–1365, 2017.

Mohamed Sherif and Darren Leitch. Twitter mood, CEO succession announcements and stock returns. *Journal of Computational Science*, 2017.

Yauheniya Shynkevich, T Martin McGinnity, Sonya Coleman, Yuhua Li, and Ammar Belatreche. Forecasting stock price directional movements using technical indicators: investigating window size effects on one-step-ahead forecasting. In *2104 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 341–348. IEEE, 2014.

Yauheniya Shynkevich, T Martin McGinnity, Sonya A Coleman, and Ammar Belatreche. Forecasting movements of health-care stock prices based on different categories of news articles using multiple kernel learning. *Decision Support Systems*, 85:74–83, 2016.

Christian Söyland. Interday news-based prediction of stock prices and trading volume. Master's thesis, Chalmers University of Technology, 2015.

Brady Twedt and Lynn Rees. Reading between the lines: An empirical examination of qualitative attributes of financial analysts' reports. *Journal of Accounting and Public Policy*, 31(1):1–21, 2012.

Manuel R Vargas, Beatriz S L P de Lima, and Alexandre G Evsukoff. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 60–65. IEEE, 2017.

Tien-Thanh Vu, Shu Chang, Quang Thuy Ha, and Nigel Collier. An experiment in integrating sentiment features for tech stock prediction in Twitter. 2012.

Beat Wuthrich, Vincent Cho, Steven Leung, D Permunetilleke, K Sankaran, and J Zhang. Daily stock market forecast from textual web data. In *1998 IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, pages 2720–2725. IEEE, 1998.

Yumo Xu and Shay B. Cohen. Stock movement prediction from tweets and historical prices. 2018.

Yuzheng Zhai, Arthur Hsu, and Saman K Halgamuge. Combining news and technical indicators in daily stock price trends prediction. In *International symposium on neural networks*, pages 1087–1096. Springer, 2007.

List of Figures

1 The interdisciplinary nature of the field (Nassirtoussi et al. [2014]) 3

3.1 Total number of the news articles published per day of the week . 20
3.2 Distribution of the number of the news articles published during different hours 20

4.1 Label distribution on different trading days as estimated from training and development data 34

5.1 One of the possible views on BNS (Forman [2003]) 51

List of Tables

3.1 Basic dataset information 17
3.2 Train, development and test sets: the temporal split 18
3.3 Number of training samples in different subsets of the data 19
3.4 Extract from the stock price information 22

4.1 The label distribution in training, development and testing data for the labels derived from opening prices 24
4.2 The label distribution in training, development and test data for the labels derived from closing prices 24
4.3 The performance of the basic model 25
4.4 The confusion matrix for the basic model on the development set 25
4.5 Highest accuracy achieved on development set in settings with different alignments for the labels derived from opening [and closing] prices [of the same day] 29
4.6 Highest accuracy achieved on development set in settings with different alignments for the labels derived from closing prices [of two subsequent days] 30
4.7 Cross-validation for the Monday data (labels derived from opening prices) 33
4.8 Cross-validation for the Monday data (labels derived from closing prices) 33
4.9 Different data alignments for Mondays (for the labels derived from opening prices) 36
4.10 Different data alignments for Mondays (for the labels derived from closing prices) 36
4.11 Separate models for Monday and the rest of the week (labels derived from opening prices) 38
4.12 Separate models for Monday and the rest of the week (labels derived from closing prices) 39

5.1 Highest accuracy achieved on development set using sentiment-based representations of the text 44
5.2 Appending the sentiment information from different sentiment dictionaries to the TF-IDF representation 46
5.3 Different types of normalization when incorporating the sentiment scores into the prediction model 47
5.4 Correlation between SentiWordNet features and target labels . . . 49
5.5 The accuracy of the model on the development set when employing n-grams 50
5.6 BNS feature selection accuracy on the development set 51
5.7 Highest accuracy achieved on development set when incorporating different types of information about stock price movements from 3 days preceding the prediction day 53

5.8 Highest accuracy achieved on development set when incorporating the stock price movement (specific) information from varying number of days 54

6.1 Specifications for the combined models 56
6.2 Models that will be tested on the test set 57
6.3 Performance of the models on the test set 58
6.4 Performance of the models on the test set with adjusted C values 58
6.5 The confusion matrix for the models performing at the baseline on the test set 59
6.6 Performance of the models on the test set after training on the balanced training data 60
6.7 Performance of the models on the test set after extending the training data 60
6.8 Performance of the models on the balanced test set 61
6.9 Performance of the models on the test set with extended training data 62

A.1 Performance of other models on the test set 77
A.2 Excerpt from the financial dictionary 78
A.3 Correlation between the financial dictionary features and the target labels 79

List of Abbreviations

BMI	Balanced Mutual Information
BNS	Bi-normal separation
BOW	Bag of words
CNN	Convolutional Neural Network
EMH	Efficient Market Hypothesis
ETF	Exchange-Traded Fund
IDF	Inverse Document Frequency
MI	Mutual Information
NN	Neural Network
POS	Part of speech
RNN	Recurrent Neural Network
RWT	Random Walk Theory
S&P 500	Standard and Poor's 500 index
SVM	Support Vector Machine
TF	Term frequency
TF-IDF	Term Frequency - Inverse Document Frequency

A. Attachments

A.1 DVD

The attached DVD contains the scripts, the full results of all the experiments and the processed version of the dataset.

A.2 Example of a news article from Bloomberg

– Apple Wins Preliminary Injunction Against Samsung Tablet
– By Joel Rosenblatt, Edvard Pettersson and Jun Yang
– 2012-06-27T08:08:32Z
– `http://www.bloomberg.com/news/2012-06-27/apple-wins-preliminary-injunction-against-samsung-galaxy-tab-1-.html`

Apple Inc. (AAPL) won a court order immediately blocking U.S. sales of Samsung Electronics Co. (005930)'s Galaxy Tab 10.1 tablet computer as the companies continue their global patent dispute. U.S. District Judge Lucy Koh in San Jose, California, issued the order yesterday after rejecting a similar request in December. Apple's request, part of a broader patent dispute over smartphones and tablets, was based on an appeals court finding that it will probably win its patent-infringement claim relating to the Tab 10.1 tablet. The world's two biggest makers of high-end phones have accused each other of copying designs and technology for mobile devices and are fighting patent battles on four continents to retain their dominance in the $219 billion global smartphone market.

Suwon, South Korea-based Samsung will take necessary legal steps in response to yesterday's ruling on the year-old model, the company said in a statement today. "This is an extension of their fight in the smartphone space," Kim Young Chan, a Seoul-based analyst at Shinhan Investment Corp., said by phone. "If you look at precedents, different cases yielded different rulings. As long as smartphones aren't blocked, Samsung's fundamentals will stay intact." Quarterly sales of the Tab 10.1 in the U.S. may total about 300,000 units on average, Kim said. Samsung, which doesn't disclose shipment figures for smartphones and tablets, sold 44.5 million smartphones globally in the first quarter, according to market researcher Strategy Analytics . Apple is also the biggest buyer of Samsung chips and displays. Hague Ruling Samsung shares gained 2.5 percent to 1,167,000 won at the close of trading in Seoul , while the benchmark Kospi index was little changed.

The U.S. sales ban follows last week's ruling in The Hague that Apple has to compensate Samsung for damages caused by a breach of patents since August 2010. Apple also sought a U.S. ban on the latest model of the Galaxy S smartphone, which went on sale in the market this month. "In this case, although Samsung will necessarily be harmed by being forced to withdraw its product from the market before the merits can be determined after a full trial, the harm faced by Apple absent an injunction on the Galaxy Tab 10.1 is greater," Koh said in yesterday's ruling. She said a June 29 hearing to address Apple's third request to block Samsung's tablet computer wasn't needed. A trial is set for July

30. The public interest “favors the enforcement of patent rights ,” Koh wrote. “Although Samsung has a right to compete, it does not have a right to compete unfairly, by flooding the market with infringing products.” Samsung Products Samsung said in a statement that it was disappointed in the ruling, while it won’t have significant impact on its business. Other Tab products will continue to be available to U.S. consumers, the South Korean company said. The ruling “will ultimately reduce the availability of superior technological features to consumers in the U.S.,” Samsung said in the statement. If Apple continues to sue based on “generic design” patent claims “innovation and progress in the industry could be restricted.” Koh yesterday rejected Samsung’s arguments that the injunction was overbroad because the infringement claim was based on one aspect of the overall product, and that it would hurt Samsung’s relationships with wireless carriers that provide the Galaxy Tab to their customers. ‘Important Driver’ Koh wrote in December that “design is an important driver in the demand for tablet sales.” “Samsung cannot be heard to complain about broken business relationships that it has established on infringing products.”

On June 4, Koh rejected Apple’s second request to ban the tablet sales while the U.S. Court of Appeals for the Federal Circuit in Washington was considering her first such denial in December. Koh said then she didn’t have jurisdiction to issue a preliminary injunction because the appeals court hadn’t issued a mandate. On June 19, the appeals court reaffirmed its May determination that Apple’s patent on a design of the tablet is likely to withstand challenges to its validity. That decision allowed the Cupertino, California-based company to renew its request to block sales of Samsung’s tablet in the U.S. Kristin Huget, an Apple spokeswoman, reiterated an earlier company statement saying it must “protect Apple’s intellectual property when companies steal our ideas.” Koh required Apple to post a $2.6 million bond to cover a potential damages payment to Samsung if Apple loses the case. The case is Apple Inc. v. Samsung Electronics Co. Ltd., 11- cv-01846, U.S. District Court, Northern District of California (San Jose).

To contact the reporters on this story: Joel Rosenblatt in San Francisco at jrosenblatt@bloomberg.net ; Edvard Pettersson in Los Angeles at epettersson@bloomberg.net ; Jun Yang in Seoul at jyang180@bloomberg.net To contact the editor responsible for this story: Michael Hytha at mhytha@bloomberg.net

A.3 Example of a news article from Reuters

– P&G keeps profit view despite softer sales
– By Jessica Wohl
– Thu Dec 11, 2008 1:04pm EST
– `http://www.reuters.com/article/2008/12/11/us-procter-idUSTRE4BA2SJ20081211`

CHICAGO (Reuters) - Procter & Gamble Co (PG.N) said on Thursday it was on track to meet its earnings forecasts in the current quarter and fiscal year, although sales will be weaker than expected this quarter because of the recession.

The world’s largest consumer products maker also said it may get out of the pharmaceutical business as it focuses on products such as toothpaste and feminine care items. P&G shares rebounded after falling 2.7 percent. While other industries such as auto and apparel retailers felt a big hit from consumers cutting back during the recession, “the picture isn’t nearly as bleak in fast-moving consumer goods,” Chairman and Chief Executive A.G. Lafley told analysts at a meeting in New York. P&G is recession-resistant, but not recession-proof, he said. Lafley expects to see early signs of economic recovery in the next year. P&G said commodity costs would still rise this year, but not as much as it thought a few weeks ago. However, it expects the stronger dollar to hurt sales more than previously expected. A stronger dollar lessens the dollar value of sales outside the United States. Lafley also said P&G would stop making new investments in pharmaceuticals, consider divesting its healthcare brands and focus its health business on over-the-counter healthcare products such as Pepto Bismol and Prilosec. P&G’s prescription drugs include Actonel and Enablex. Jon Moeller, who is set to become chief financial officer on January 1, said private label products are growing “modestly” in the United States and P&G has seen consumers trade down to less expensive branded goods. For example, a shopper who used to buy P&G’s high-end Pampers diapers may now buy the company’s less-expensive Luvs brand, or someone who bought pricey Tide laundry detergent may switch to P&G’s cheaper Gain brand. To keep consumers interested, P&G has been promoting the value of its products, including an ad campaign that tells men that its Gillette Fusion razor costs as little as $1 per week to use. Consumer value is “always important and absolutely critical in times like these,” Lafley told the crowd of about 250 people at the meeting. While P&G does not anticipate making significant price moves now, Lafley said it would continue to raise prices when it needs to recover higher costs. SALES PRESSURE P&G said retailers, distributors and consumers have lowered inventories in developed and developing markets. Consumers have begun to use up what they have in their pantries rather than stock up on items such as shampoo and detergent as they try to curb spending. P&G did not expect inventory levels to be adjusted so quickly or dramatically, Moeller said. The company said second-quarter organic sales growth, which excludes the impact of acquisitions, divestitures and foreign exchange, will fall short of its forecast range of 4 percent to 6 percent. Full-year organic sales growth is still expected to be in the range of 4 percent to 6 percent. Morningstar analyst Lauren DeSantowas not “terribly surprised” by the lowered revenue outlook for the second quarter. “P&G is a cost-cutting story right now, but I’m encouraged that they expect to meet their top-line target for the year” she said. Other plans discussed on Thursday included a focus on sustainable products, such as larger rolls of toilet paper, and efforts to cut transportation and media costs. P&G still plans to earn $1.58 to $1.63 per share for the fiscal second quarter that ends this month and $4.28 to $4.38 per share in fiscal 2009, which ends in June. Analysts, on average, expect P&G to earn $1.56 this quarter and $4.28 this year, according to Reuters Estimates. The company now expects an additional $2 billion in commodity costs in fiscal 2009, down from a forecast of $2.7 billion it gave at the end of October. It originally expected $3 billion in such costs this year. While the forecast has come down, it will still be the highest cost increases P&G has faced, up from a $1.5 billion rise in fiscal 2008. P&G now expects foreign exchange to cut sales by about 5 percent, both in the current quarter and the fiscal year, a deeper cut than the 1 percent to 2 percent hit it expected. Changes in areas such as production helped P&G cut over $600 million in costs over the last five years. The company could save another $600 million over the next five years by making moves such as using trains more often to transport goods in Western Europe, said Chief Operating Officer Bob McDonald. P&G shares were up 40 cents at $59.52 in midday trading. P&G shares, a component of the Dow Jones industrial average, fell nearly 19.5 percent from the beginning of the year through Wednesday. The shares rose about 14.2 percent in 2007. (Additional reporting by Christopher Kaufman and Aarthi Sivaraman in New York; Editing by Derek Caney and Andre Grenon)

A.4 Performance of the other models on the test set with adjusted C values

The results of testing further models on the test set and adjusting the C values.

Model	**C**	**Test set accuracy**
Enhanced basic + lemmatization	30	60.73
TF-IDF + 3 prev. days specific info	26	60.73
TF-IDF + 3 prev. days binary info	26	60.21
bigrams	1	59.69
bigrams + lemmatization	1	59.69
3-skip-2-grams + BNS (100K best features)	6	60.21
4-skip-2-grams + BNS (1M best features)	12	61.26
lem. + 4-skip-2-grams + BNS (1M best features)	16	60.21
TF-IDF (12 h before 18:00) + av. sent[1]	2	60.21
TF-IDF (24 h before 18:00) + av. sent[2]	20	60.21
Baseline	–	59.69

Table A.1: Performance of other models on the test set

---

[1] 5th (penultimate) model from Table 5.3, 12-hour observation window

[2] 5th (penultimate) model from Table 5.3, 24-hour observation window

A.5 Excerpt from the financial dictionary by Loughran and McDonald [2011]

Word	Sequence Number	Word Count	Word Proportion	Average Proportion	Std Dev	Doc Count	Negative	Positive	Uncertainty	Litigious	Constraining	Superfluous	Interesting	Modal	Irr_Verb	Harvard_IV	Syllables	Source
AARDVARK	1	81	5.69E-09	3.07E-09	5.78E-07	45	0	0	0	0	0	0	0	0	0	0	2	12of12inf
AARDVARKS	2	2	1.40E-10	8.22E-12	7.84E-09	1	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABACI	3	8	5.62E-10	1.69E-10	7.10E-08	7	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABACK	4	5	3.51E-10	1.73E-10	7.53E-08	5	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABACUS	5	1752	1.23E-07	1.20E-07	1.11E-05	465	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABACUSES	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	12of12inf
ABAFT	7	4	2.81E-10	3.25E-11	3.10E-08	1	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABALONE	8	80	5.62E-09	3.88E-09	1.04E-06	30	0	0	0	0	0	0	0	0	0	0	4	12of12inf
ABALONES	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	12of12inf
ABANDON	10	80492	5.65E-06	5.25E-06	4.12E-05	45941	2009	0	0	0	0	0	0	0	0	1	3	12of12inf
ABANDONED	11	174298	1.22E-05	1.22E-05	8.81E-05	83234	2009	0	0	0	0	0	0	0	0	2	3	12of12inf
ABANDONING	12	15926	1.12E-06	9.95E-07	1.45E-05	10125	2009	0	0	0	0	0	0	0	0	2	4	12of12inf
ABANDONMENT	13	177889	1.25E-05	1.19E-05	8.19E-05	65686	2009	0	0	0	0	0	0	0	0	1	4	12of12inf
ABANDONMENTS	14	7091	4.98E-07	6.85E-07	1.75E-05	3891	2009	0	0	0	0	0	0	0	0	2	4	12of12inf
ABANDONS	15	6430	4.52E-07	2.13E-07	4.48E-06	4625	2009	0	0	0	0	0	0	0	0	2	3	12of12inf
ABASE	16	46	3.23E-09	3.19E-09	7.35E-07	39	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABASED	17	72	5.06E-09	1.10E-08	2.33E-06	59	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABASEMENT	18	6	4.21E-10	9.29E-11	4.02E-08	6	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABASEMENTS	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABASES	20	2	1.40E-10	1.64E-10	1.28E-07	2	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABASH	21	3	2.11E-10	8.76E-11	4.94E-08	3	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABASHED	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABASHEDLY	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	12of12inf
ABASHES	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABASHING	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABASHMENT	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABASHMENTS	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABASING	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABATE	29	22938	1.61E-06	9.55E-07	1.32E-05	12824	0	0	0	0	0	0	0	0	0	1	2	12of12inf
ABATED	30	26393	1.85E-06	9.47E-07	1.39E-05	12126	0	0	0	0	0	0	0	0	0	2	3	12of12inf
ABATEMENT	31	99853	7.01E-06	3.69E-06	4.10E-05	31115	0	0	0	0	0	0	0	0	0	2	3	12of12inf
ABATEMENTS	32	10159	7.14E-07	4.77E-07	1.50E-05	5543	0	0	0	0	0	0	0	0	0	2	3	12of12inf
ABATES	33	331	2.33E-08	1.47E-08	1.30E-06	263	0	0	0	0	0	0	0	0	0	2	2	12of12inf
ABATING	34	1821	1.28E-07	7.47E-08	3.16E-06	1256	0	0	0	0	0	0	0	0	0	2	3	12of12inf
ABATTOIR	35	125	8.78E-09	5.79E-09	8.25E-07	56	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABATTOIRS	36	136	9.55E-09	5.80E-09	1.24E-06	41	0	0	0	0	0	0	0	0	0	0	3	12of12inf
ABBE	37	180	1.26E-08	1.33E-08	1.92E-06	106	0	0	0	0	0	0	0	0	0	0	1	12of12inf
ABBES	38	4	2.81E-10	8.95E-11	6.04E-08	2	0	0	0	0	0	0	0	0	0	0	2	12of12inf
ABBESS	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	12of12inf

Table A.2: Excerpt from the financial dictionary

A.6 Correlation between the sentiment scores and the target labels

	Constraining	Harward IV	Interesting	Irr Verb	Litigious	Modal 1	Modal 2	Modal 3	Negative	Positive	Superfluous	Uncertainty	**Target**
Constraining	1	0.17656	-0.087516	-0.14806	0.24369	0.21318	0.15856	0.062384	0.12579	-0.11614	0.023465	0.055141	**-0.0253**
Harward IV		1	-0.017432	0.30535	0.019112	-0.033433	0.43114	0.27757	0.79582	-0.037906	0.051926	0.44448	**-0.0366**
Interesting			1	-0.10816	0.05302	0.0058721	-0.15524	-0.036546	0.037772	0.021606	-0.00022845	-0.03662	**0.0029**
Irr Verb				1	-0.21304	-0.24079	0.42153	0.21836	0.25066	-0.031072	0.072623	0.301	**-0.0159**
Litigious					1	-0.03007	-0.17438	-0.11944	0.1852	-0.21974	-0.018559	-0.17057	**0.0220**
Modal 1						1	0.027121	0.035601	-0.10513	0.013863	-0.059582	0.00098495	**-0.0114**
Modal 2							1	0.40793	0.3074	0.17065	0.063571	0.48414	**-0.0053**
Modal 3								1	0.24525	0.082288	0.06551	0.84182	**0.0010**
Negative									1	-0.12265	0.013063	0.39724	**-0.0175**
Positive										1	0.001243	0.13527	**0.0542**
Superfluous											1	0.078744	**0.0180**
Uncertainty												1	**-0.0166**
Target													**1**

Table A.3: Correlation between the financial dictionary features and the target labels