TILBURG UNIVERSITY

Understanding Society

“A Twitter sentiment analysis to predict the public’s opinion for an independent government institution”

A case study of an independent government institution

MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES TILBURG UNIVERSITY

Tilburg University

June 2021

Master Thesis Spring 2020-201

Robin Clemens Maria Boots

Under the supervision of:

University	dr. H. Weigand
Supply Value B.V.	MSc Thijs Latour

Preface

This thesis presents the result of the final graduation project in fulfillment of the MSc Data Science and Society at the University of Tilburg (UvT) and the Internship at the Digital department at Supply Value (SV).

I would like to express my gratitude to several people. First of all I would like to thank Thijs Latour and Hans Weigand for being my primary supervisors over the extent of this entire project. The help provided was always deeply appreciated and helped greatly in focusing towards more structure, insight and implementation of supervised classification methods. The guidance has been received as supportive and approachable. Secondly, I would like to express my gratitude towards the other team members from SV. The overall warm welcome to the team and the company as a whole has been satisfying above expectations. Moreover, the practical relevance and experience gathered over the course of my internship has been immense and extremely helpful in the determination of my future career path.

Abstract

The Raad voor Rechtsbijstand (RvR) is a Dutch independent government institution that focusses on (social) legal aid. Medio 2014, politicians in The Hague, the Netherlands, decided to alter the social legal aid system. This research centers on a specific process within the (social) legal aid system, called the picket process. The picket process ensures that citizens with a legal problem have access to affordable and good quality legal aid. The picket process is triggered by picket notifications. The picket process has also seen many alterations and new developments. An unfortunate development over the past years involves the reduced number of social lawyers entering the labor market. Additionally, it has always been an issue for the RvR to efficiently plan the picket process as it has been hard to the determine the root cause that triggers the picket notifications. This means that there is a clear need for more understanding towards this problem. The RvR believes that the social opinion may influence the number of picket notifications. The aim of this research is to identify and eventually predict the public's opinion in order to more efficiently plan the associated picket planning. In view of that, the following problem statement needed to be answered in this thesis: *"How can sentiment analysis be used to measure the influence of the public's opinion on the picket process?"*. The research question is answered using two datasets which are provided by the external case company and a dataset which is extracted using the Twitter API. The extracted Twitter data has been preprocessed in 12 steps. Subsequently lexicon based approaches have been applied before the Bag of Words (BoW) and TF-IDF features were selected. Lastly, six supervised machine learning techniques have been applied to demonstrate the most suitable classification techniques for region based Twitter Sentiment Analysis. Ultimately, the Extra Trees classifier turned out to be the best performing method for BoW and the linear classifiers for the TF-IDF feature.

Key words: Twitter sentiment analysis, Twitter API, lexicon based approaches, supervised machine learning algorithms, evaluation metrics

Table of Content

Chapter 1: Introduction

1.1 Background

One of the preconditions for a well-functioning rule of law is that citizens with a legal problem have access to affordable and good quality legal aid. In order to avoid that someone with a legal problem has to refrain from engaging legal aid because of the costs involved, The Legal Aid Act ensures a system of subsidies. On the basis of this law, those seeking legal aid with an income below a certain threshold and persons who receive an addition to their income automatically, such as suspects who have been detained, receive subsidized legal aid. In the Netherlands the Raad voor Rechtsbijstand (RvR), also referred to as the RvR, is responsible for the implementation of the Legal Aid Act. The RvR receives its budget from and is accountable to the Ministry of Justice and Security. In order to gain insight into the development of the system, the Ministry instructed the RvR in 2003 to set up a periodic registration system, hereafter referred to as 'the Monitor Subsidised Legal Aid Monitor' (MGR). This monitor pays attention to the development of the demand for and the supply of legal aid. Although no concrete recommendations for policy are made, the monitor aims to provide the basis for further discussion about the system of subsidized legal aid. To this end, it is an important source of information. In addition to the 'Monitor Subsidized Legal Aid 2017', 2018 saw the publication of the 'Zero Measurement' report with regards to subsidized legal aid in the Netherlands, which describes the state of affairs of the system in a more qualitative sense (Combrink-Kuiters et al., 2018). There are, after the release of the reports of three committees (Wolfsen, Barkhuysen and Van der Meer), changes in the system of subsidized legal aid to be expected. A reliable and thorough baseline measurement makes it possible to determine the consequences of the implementation of various measures in due course. The demand for subsidized legal aid is not a constant. There are three clusters of factors that cause the demand for subsidized legal aid to fluctuate, and those factors also influence the nature of this specific demand (Rijkschroeff et al., 2001; Ter Voert and Klein Haarhuis, 2015). Social developments such as a pandemic may lead to an increase or decrease in the number of persons applying for legal aid and to modifications in legal aid and to changes in the type of issue for which help is sought. In addition, the development of the Dutch legal culture is of influence. Alterations in jurisdiction or other processes such as Alternative Dispute Resolution (ADR) and an increase in the number of legal aid insurances have an effect on the demand. Finally, the legal structure influences the demand for legal aid. System changes, due to e.g. positive pilot results, and changes in legislation and policy that affect the target group of the Legal Aid Act, have a significant influence (van Gammeren-Zoeteweij et al., 2018).

Arrested suspects of certain criminal offences, foreign nationals whose freedom has been under the regulation of the ‘Vreemdelingenwet 2000’ or whose freedom has been deprived and psychiatric patients who are taken into custody, are entitled to consult a subsidized lawyer on duty. The allocation of lawyers is arranged through picket services. The RvR facilitates this provision. Attorneys are scheduled according to a rotation system in order to guarantee an available lawyer at all times.

As of the first of August 2014, a regulation for the provision of legal aid in picket cases became legally active (Raad voor Rechtsbijstand, 2014). The legal aid in picket cases came into effect. This regulation replaced all regional regulations concerning the picket organization. These regulations were renewed on May first, 2017, and the following elements of the picket regulations have been legally active and enforceable (van Gammeren-Zoeteweij et al, 2018):

- Lawyers can be registered for a maximum of three of the following picket types registered: criminal picket, juvenile picket, psychiatric patient picket and/or foreigners picket, or a maximum of two of the previous picket types in addition to participation in an availability roster for the registration center for asylum seekers.
- A preferred lawyer is only appointed if the litigant (applicant for justice) expressly requests it himself.
- In order to be able to participate in a picket scheme, an attorney at law must uphold to the following conditions:
    - meet the registration requirements for the type of picket that is involved (which includes the substantive requirements);
    - observe the availability times for picket reports (on working days, weekends and public holidays from 07.00-20.00 hours);
    - be in possession of a mobile telephone with internet access for the to receive and respond to picket reports;
    - to accept an automated (juvenile) picket call within 45 minutes (applies to preferred and/or scheduled lawyers) and, in the within 180 minutes in the case of a psychiatric patient pick-up;
    - to perform the picket shifts assigned to him personally; exchanging and deputizing a shift may be done by the lawyer, the participating attorney at law can do so himself via the exchange of a shift and its replacement can be arranged by the participating lawyer himself via the automated picket planning programme.

The RvR states how many picket reports were received digitally in 2017 and what proportion of these was done by preferred lawyers. The statistics in this paragraph relate only to the (juvenile) criminal picket notifications and psychiatric patient picket notifications. According to van Gammeren-Zoeteweij et al. (2018), a total of more than 142,000 picket reports were received digitally in 2017, of which more than 110,000 were adult criminal picket reports, more than 21,000 were juvenile criminal picket reports and juvenile punishment picket notifications, and over 10,000 notifications for psychiatric patient picket. In 37% of these picket reports (almost 53,000 times), the person seeking justice asked for a specific preferred lawyer. In 80% of the notifications the preferred lawyer accepted the notification. If the preferred lawyer does not accept a notification of preference, it is then reported to the lawyer who is scheduled to be on duty on that day. Approximately 37,000 notifications were handled by lawyers scheduled to be on duty.

Nevertheless, the RvR encounters serious problems estimating the inflow of picket notifications. In consequence, the matching of litigants and picket lawyers according to predictive planning is not efficient and the root cause of has not been detected yet. Thus, either litigants do not get the timely legal aid they need or there are too many lawyers on duty which results in more costs. The extent to which this issue occurs is not yet determined and this thesis project aims to contribute to more understanding regarding this issue.

1.2 Problem Statement

Within the Dutch legal aid sector there are many different opinions regarding the best course of action. The RvR, which is an independent Dutch government authority, is (partly) accused of not assigning social legal aid to litigants. There is little to none data driven research towards the matching of social picket lawyers and litigants and the relation between picket litigants and public opinion which makes it more difficult to detect the problem and efficiently plan the picket process. This thesis aims to increase the understanding of public opinion in relation to picket process by conducting a Twitter sentiment analysis (TSA). The TSA will be the fundament for prediction using supervised machine learning classifiers.

1.3 Research Questions

Primary research question: **How can sentiment analysis (SA) be used to measure the influence of the public's opinion on the picket process?**

Several sub-questions are necessary to constructive and argumentatively answer the main research questions:

- How can the public's opinion influencing the picket process be identified?
- What data and attributes are needed to analyse the influence of the public opinion on the picket process?

- How can a ground-truth be established in order to support the sentiment expressed by the public opinion?
- How can the sentiment expressed by the public opinion be measured?
- What is the influence of the public opinion on the picket process?

1.4 Scope

Within this project I will try to measure the influence of the public opinion on the picket process through a SA on Twitter data. The RvR encounters issues regarding the incoming picket notifications and its consequential actions. A case will be developed to indicate the scope of this problem. This case will be supported by a sentiment trend analysis on Twitter data to show the relation between the public opinion and the picket process.

1.5 Delimitations

Twitter by itself does not comprise the public opinion. Other social media platforms and microblogging forms and blogs need to be analyzed to gain more knowledge of the concepts a more enhanced grasp on the public opinion. Moreover, many people do not express their opinion online and there are many different media forms not taken into account. Hence, more data from different data sources need to be taken into account in order to achieve more generalizable results of the (online) public opinion. Due to time and resource limitations it is impossible to tackle and gather all this data. Gathering data from different sources would instigate different pre-processing and classification approaches. Ultimately, there has been a decision made to limit the data to one information source, Twitter. Twitter provides access to its Tweets by use of a Twitter Application Programming Interface (API).

1.6 Research Design

SA is the process of using natural language processing, statistics and text analysis to evaluate a certain emotion or opinion. Therefore, SA is also referred to as ‘opinion mining’ and considered a fundamental part of the text mining field within the data mining domain (Talib, Hanif, Ayesha & Fatima, 2016). SA and opinion mining have many similarities, but the terms are not interchangeable as there are distinct differences. These differences will be tackled in a subsequent part of this paper. Sentiment can be extracted from and derived to several topics and sources of data just as Tweets, reviews, comments, blogs or many other forms of microblogging. Research in the field of text mining utilizes classification, clustering and other data mining techniques to derive detailed information using systematic approaches of data sources which are too immense to be tackled by humans (Injadat, Salo & Nassif, 2016).

In this research supervised learning classification approaches, that classifiy Tweets into sentiment categories based on already sentimentally labeled training data, will be conducted. Therefore, the research project requires a data mining framework in order to adequately and efficiently perform the process. The CRISP-DM, which denotes Cross-Industry Process for Data Mining, will be the data mining framework used in this research. This describes the standard process for generating insights from raw data (Wirth & Hipp, 2000). The CRISP-DM model divides the process of data mining into six phases (see Figure 1): Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment (Wirth & Hipp, 2000). These phases do not necessarily take place sequentially: it should be possible to go back to a previous one. Also, the cycle is run over and over in order for new insights to be properly tracked. Hence, it is an iterative process that models a desirable sequence of events without excluding any possible alternatives. Further elaboration on the CRISP-DM process is provided in the Appendix.

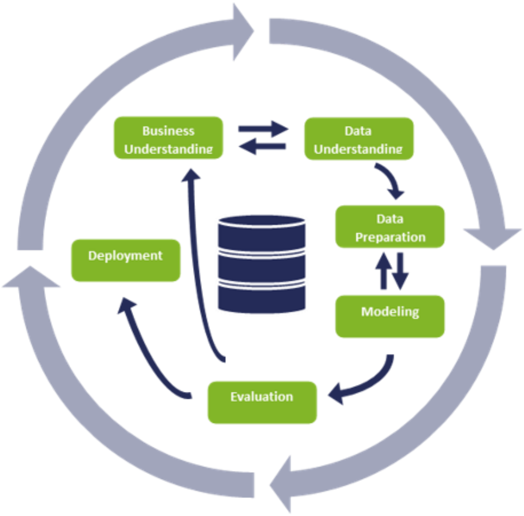


Figure 1 CRIPS-DM model

Chapter 2 Related work

To gain an improved understanding of the available research conducted with regards to SA, a literature study is conducted. The systematic literature review on SA has been performed in preparation of the thesis project to achieve a better understanding of application domains and techniques. Moreover a systematic literature towards the picket process of the RvR has been conducted to get a better knowledge of the concepts and antecedents of the picket process. The literature review encompasses the proposed answer to the sub-research questions: how can public events influencing the picket process be identified?; What data and attributes are needed to analyse the influence of the public opinion on the picket process?, and: how can a ground-truth be established in order to support the sentiment expressed by the public opinion?

2.1.1 Picket process and the public's opinion

An elaborate description of the Dutch picket process has been given in the aforementioned background of this paper. Therefore, the key aspects of the picket process with significant influence on this research will be provided in this section. Unfortunately, adequate academic research on the picket process relevant to TSA has not been performed. Hence, this research will focus on the aspect of picket notifications by utilizing a significant advantage of the Twitter API.

There are several different methods for a picket notification to go through the system. The general picket notification stream commences at a regional location which sends a notification to the central picket department. The picket notification is then dealt depending on the type of picket it involves. This type of picket notification is ignored in this study since this study focusses on streamlining the general picket process. Hence, in order to streamline the general picket process, the regional location is identified to be of fundamental importance. The dataset provided by the RvR involves the number of notifications send for a particular date. As such, the notifications have been visualized per for the year 2020. The results are provided per day in Figure 2.

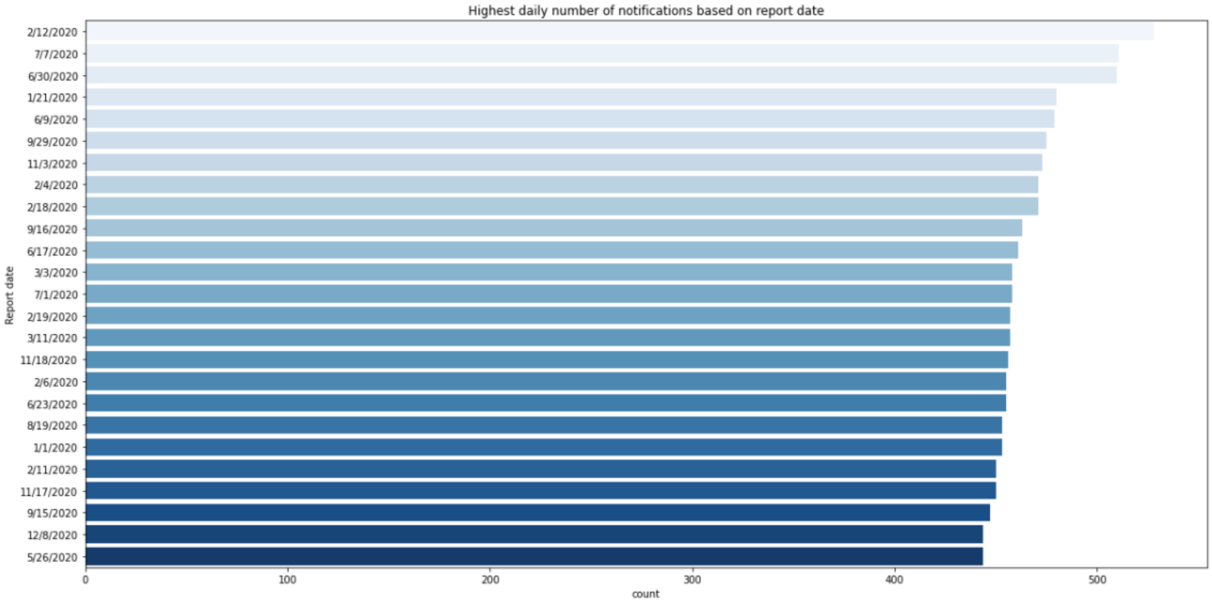


Figure 2 Notifications per day 2020

The root cause for the number of notifications per day has not been determined yet. However, it is noteworthy that the notifications are evenly spread while the corona measures have been applied. The measures were known a week in advance before implementation, the timeline of measures can be found in the Appendix. During the summer a turnaround was noticeable, the corona measures became less severe. As a consequence, bars opened again, minor events were allowed and more human traffic was noticeable. Hence, more picket notifications would be expected. Nonetheless, as the data was evenly spread, this was not the case. The public sentiment may have been positively surged and affected the number of picket notifications negatively. Nonetheless, many factors play a role affecting the picket process and the associated picket notifications, but in this research the assumption is being made that the number of picket notifications can be influenced by the social opinion.

The traditional way of understanding society its emotions involves market research in the form of interviews, focus groups, opinion polls and many other methods (Desai, 2002). Even though this has proven to be an adequate method, these procedures are limited by the duration of completion and cost. The data scientific approaches primarily focus on user generated, social media contents as it can almost immediately be processed (Schwartz and Ungar, 2015). Furthermore, an exponential growth of social media usage can be seen over the years, creating even more public content on all sorts of different matters available for analysis (Montangero and Furini, 2015). Next to Tweet content on social media platform Twitter, metadata is retrieved which includes information just as geographical location, timestamp and username (Stefanidis et al., 2017). Accordingly, social media platform generated contents offer a lot of opportunities to identify, analyse and ultimately understand people's sentiment.

2.1.2 Twitter sentiment analysis

A summary of the findings of the systematic literature is stated in this chapter, the entire systematic literature review can be seen in appendix X Systematic literature review. The databases used to search for the literature review are IEEE-Xplore, Springer, Worldcat, GoogleScholar and Elsevier. In total a vast amount papers are reviewed of which many have been thoroughly analyzed, these have been listed at the end of the Chapter. The review criteria for articles include published date, database, journal, topic and the number of citations. These criteria have helped to select the pertinent literature as the exponential development of research within this topic leaves room for inadequate and premature conclusions. The literature review identifies and analyzes shared components of the X articles that are most relevant for research in the field of TSA.

Microblogging platforms and in particular social media platforms just as Twitter form an active communication channel for daily situations and events. Research proposes that a fast examining of microblogging messages to identify relevant actionable information can result in tremendously useful information and help to obtain valuable insight in different fields. Twitter lends itself to be an exceptional supplier of real-time information. Nevertheless, Twitter is a vast pool of uncountable bytes of data. Text mining is required to specifically search and extract the information requested.

Text mining is seen as an emerging technology which entails the extraction of useful information from unstructured textual data (He et al., 2017; Martin and Rice, 2007). Social media is increasingly being used as a tool and source of text mining by data scientists with the aim to provide better service for their customers or contribute to their research field. The main applications of social media entail the identification of situational awareness and extraction of useful information. Situational awareness commences with the identification of the aspects regarding a certain event, followed by processing of the state, and finally comprehending the dynamics of relations among the causalities and location of the event (Stowe et al., 2016). Online distribution of useful and time-critical information during an event can be highly effective in revealing important insights into and subsequent to the event as the situation progresses (Stowe et al., 2016).

To illustrate, He et al. (2013) stated that many techniques can be used to apply business analytics on social media content. For example, a combination of text mining, content analysis, statistical analysis, and SA techniques can be utilized to identify Twitter social media content collected from organizations' accounts and webpages in order to generate meaningful insights and compare according sentiments.

Moreover, He et al. (2013) applied “a social media competitive analytics approach to analyze unstructured text content on the Facebook and Twitter sites of the three largest US pizza chains: Pizza Hut, Domino’s Pizza, and Papa John’s Pizza”. Their results revealed the value of social media competitive analysis and the power of text mining as an effective technique to extract business value from social media content. Other applications can even safe lives; citizen participation during specific events has been encouraged numerous times and the freely available data streams by responders whereby civilian posting something about the event or theme is evaluated as a sensor. This is exemplified by Sakaki et al. (2010) who demonstrated Twitter as an indicator for earthquakes with a detection probability of 96 percent. In the study by Sakaki et al. (2010) each Twitter user has been used as a sensor for location estimation in order to estimate the centre of the earthquake. This illustrates the potential of location based Twitter data.

SA and opinion mining are areas of text mining which assist in knowledge discovery and data mining. Both techniques focus on emotion recognition and polarity detection. The polarity score determines whether a tweet can be categorized positive or negative (or neutral). These techniques cover information retrieval, text analysis, machine learning and visualizations and therefore are labelled multi-disciplinary (Pawar et al., 2018). Even though SA and opinion mining seem similar, there are distinct differences. SA classifies in a binary polarity manner meaning the use of a relatively small number of classes, whereas opinion mining involves additional tasks just as summarization next to polarity detection. The use of Tweets, in this paper the use of textual content extracted from Tweets, presents certain challenges for classification and ultimately information extraction.

2.2 Sentiment analysis challenges

Detecting sentiment in Tweets differs substantially from sentiment detection in traditional text. Researchers are faced with several different challenges due to the informal characteristics associated with a social media platform like Twitter. The created content is ever evolving, extremely dynamic and Tweets have a specific length limitation (the length limitation has been adjusted from 140 characters to respectively 280 characters as of November 2017). According to Bermingham and Smeaton (2010), Twitter sentiment detection is a much easier task compared to longer text documents using Support Vector Machine (SVM) and Multinomial Naïve Bayes (MNB) classifiers. Even though topic relevance is not often considered in TSA, it can be identified more easily in Twitter posts as a result of hashtags or consideration of the presence of a specific word as an indicator.

Another aspect of TSA is the informal created content which is highly associated with the incorrect phrasing and use of language. Tweets consists of textual idiosyncrasies; the use of slangs, neologism (coining of new words) and emphatic language. Emphatic language has been proven to be present in one of every six posts (Brody and Diakopoulos, 2011). Moreover, as a consequence of misspellings and extensive use of incorrect grammar, a significant level of noise is present. This occurrence has an influence on the performance of the SA and is known as data sparsity (Saif et al, 2012). Saif et al. (2012) suggested semantic smoothing, discounting the effect of general words, to decrease the sparseness in the extracted data.

Next to the wide use of empathic language, Tweets are created in many different languages, sometimes even simultaneously. The length limitation assigned to Tweets increase the level of difficulty to detect the language of the text. Narr et al. (2012) performed a TSA on Tweets in four different languages. The results helped to form the conclusion that the suggested classifier, with similar applied pre-processing steps, performed successfully.

Negation words are present in Tweets and form the polarity. Therefore it is of utmost importance to deal with and detect negation appropriately. Researchers have widely adopted the approach of switching the polarity of negation words in the opposite direction. In addition, a pioneer in tackling negation words, is the study by Kiritchenko et al. (2014), who established two different lexicons, one with terms that often exist in context without the negation words and one with terms that exist in context with the negation words. The remarkable findings illustrated that negation with both positive and negative words suggest a negative sentiment.

Another challenge in TSA is the tackling of stop words. In every language there are words that are not very informative and occur too frequently. Luckily, in most languages, there is a list of stop words available and applicable in programming format. Nevertheless, pre-listed stop words lists need to be thoroughly scanned against the frequency of occurrence of specific words. Illustrated by the word ‘like’ which is a word that carries distinctive value in SA but it is also labelled a stop word in the English language. Saif et al. (2014) displayed fluctuations in classification performance, data scarcity and size of the classifiers feature space due to the application of six dissimilar stop word detection methods on six unrelated datasets.

An additional characteristic when assigning sentiment to text data is tokenization. Tokenization is the process of transforming sensitive data into non-sensitive data using tokens (Owoputi et al., 2013). In TSA, Tweets are split up into meaningful semantic units (tokens) which can take the form of emoticons, words, phrases or even symbols. In other words, normal text strings, sensitive data, is transformed into a list of desired words, non-sensitive data that can be used as input for the machine learning methods. Owoputi et al. (2013) is a forerunner in TSA-tokenization, and the study was the first to create an approach to tokenize Twitter specific data.

Lastly, in order to identify all the different aspects that need to be taken into account while conducting TSA, is multimodal content. Multimodal content refers to different forms of tweet content. Tweets can include videos or images next to text data. Images and videos can contain valuable information that may influence the SA and its performance. Nevertheless, image and video extraction is not used and will not be taken into account in this research since this is an under-investigated field of study.

2.3 Sentiment analysis levels

According to Pawar et al. (2015), SA consists of three levels. Document level which classifies the sentiment on a single topic. Ultimately, comparative learning, the discovery of commonalities and differences within the text, is not possible with regards to tasks performed on a document level. Another level distinguished by Pawar et al. (2015) is sentence level; it defines whether each sentence states a negative, positive or neutral opinion. Subjectivity classification is closely related to sentence level analysis except that it subjectivity level classification only determines whether a sentence or part of a sentence is subjective or objective.

A more detailed analysis is conducted on aspect level analysis, the third level of SA. In this form of analysis the aspects within a text are determined and subsequently classified in respective aspects. For instance in a review of a portable speaker: a customer says, “The handset is not convenient, but the sound is good.” In this example the aspects are handiness and sound. Aspect level is the preferred analysis method as comparative learning can be considered on this level.

In the literature, as aforementioned, SA has been conducted at the document, sentence and aspect level, where the sentence level analysis targets each sentence in order to assign it a certain polarity. Twitter has limited the length of each tweet resulting in single sentence content for the majority of Tweets. Hence, in the light of TSA there is no distinctive difference between aspect and sentence level analysis. As a consequence, TSA can be practiced on either sentence or aspect level.

2.4 Twitter sentiment analysis approaches

TSA methods use a classification mechanism, which produces a specific sentiment score as output, to classify linguistic structures in a given tweet. Subject to the type of sentiment classifier, the algorithm allocates a value from the classification model that is generated by training the classifier. This is known as a machine learning based method. Another approach in TSA is the allocation of a predefined value to words by an algorithm recognized as lexicon based methods. Machine learning and lexicon based methods use different calculations to generate the sentiment score. Generally, sentiment scores can be expressed in numerical output (e.g. polarity scores) and binary output (e.g. negative/positive). In the classification process, a classification model is used which runs a series of calculations to allocate polarity scores to the words in a tweet.

In practice both methodologies are used in TSA, see appendix 1. The numeric polarity score allows for a more detailed and thus clearer, more detailed representation of the sentiment in a text, whereas the binary sentiment score gives room for a faster understanding of the sentiment in a tweet.

In this paper, the theoretical framework and associated literature review is based on an adaptation of the study by Giachanou and Crestani (2016) and Ligthart et al. (2021) on the most important findings and key developments in the field of TSA relevant for this study. Other relevant literature has been added accordingly. According to a thorough literature review and for the purposes of this study two primary approaches have been acknowledged: machine learning and lexicon based approaches. Neethu and Rajasree (2013) described a machine learning approach as a machine learning method that also involves multiple diverse features to form a classifier that may identify Twitter posts (which express a sentiment or opinion). A lexicon based approach works differently than machine learning approach in the form that a lexicon based approach either utilizes a manually or automatically put together list of negative or positive words to extract the polarity of the researched text (Bonta and Janardhan, 2019). Hybrid approaches, which combine both machine learning and lexicon based methodologies to attain an improved performance and deep learning approaches, have not been taken into account for the purposes of this study (Kumar et al., 2012).

2.4.1 Lexicon based approaches

Lexicon based methods establish polarity (binary) or polarity score (numerical) to define the general opinion score of a tweet by leveraging lists of words. Contrary to machine learning methods, lexicon based approach do not have need of training data. Lexicon based methods can be categorized in two classes: dictionary-based and corpus-based. Dictionary-based approach encompasses the use of a created dictionary by initially taking a few words which can be extended through the use of e.g. WordNet, SentiWordNet and Multi-Perspective Question Answering (MPQA) until no more words can be tallied to the dictionary (Gupta et al., 2020). The corpus-based approach obtains sentiment orientation of context-specific words using a statistical, based on frequency of positive or negative identified context, or semantic approach, assigning sentiment value based on synonyms or antonyms of the particular word (Gupta et al., 2020).

In TSA, as a result of tweet idiosyncrasies and dynamic characteristics, lexicon based approaches are not as common as machine learning approaches. Nonetheless, there have been some widely adopted lexicon based algorithms created for SA in social media. WordNet dictionary is extensively utilized by many practitioners but it is detrimental to take into account what features are filtered out (El Hannach and Benkhalifa, 2018; Gupta et al. 2020; Musto et al., 2014; Wijayanti and Arisal, 2021). Thelwall et al. (2012) developed and improved SentiStrength; a human-coded lexicon containing a list of negations, emoticons and boosting words using empathic lengthening to detect sentiment strength of social media text. An extensively recognized lexicon based approach by Ortega et al. (2013) employs a three-step technique for TSA. The first step involves pre-processing which is followed by polarity detection and the final step employs rule-based classification. SentiWordNet and WordNet formed the foundation for polarity detection and rule-based classification in the three-step technique.

Hu et al. (2013) suggested an unsupervised SA method centered on emotional signals, in a particular emotion indication and emotion correlation. The noteworthy results of this study instigate the use of emotional signals in SA in order to achieve the best results. Another lexicon based approach by Saif et al. (2016), SentiCircles, takes into account patterns of words that co-appear in dissimilar context and this method outperformed MPQA and SentiWordNet based methods in terms of accuracy.

2.4.2 Machine learning approaches

A bigger part of the many proposed methods that tackle TSA are machine learning methods. These machine learning methods use classifiers that are trained on several features of Tweets. Classification approaches are known for prediction of qualitative responses. This particular type of data analysis has been an active part of scientific research and prominently in TSA. Even though all classification approaches are based on linear discriminants, a great quantity of methods and techniques are available (Apté & Weiss, 1997). It is difficult to establish the best approach for a specific situation, due to diverse characteristics of approaches and data it deals with. Ho and Pepyne (2012) demonstrated that classification tasks fall under the no-free-lunch theorem which states that there does not exist a one-size-fits-all classification approach. For that reason some widely adopted classifiers in this particular field are listed and elaborated.

The study by Go et al. (2009) has been a forerunner in the field of TSA, as they classified Tweets in a binary manner assigning respectively positive and negative polarity. They used a distant supervision machine learning classifier and based their data collection on the technique proposed by Read (2005). This technique used emoticons to reduce dependency in machine learning techniques and proved that dependency in sentiment classification can emerge in language, temporal, topic and domain type (Read, 2005). Repetitive Tweets (Retweets) and messages comprising of both negative and positive polarity were filtered out. The remaining training data was a large dataset of more than 1,500,000 million posts that were correspondingly distributed in the two categories. The machine learning classifiers were based on earlier work by Pang (2002), namely NB, SVM, and Maximum Entropy classifier (MaxEnt) were used in combination with unigrams, bigrams and part-of-speech (POS) features. The results instigated that the use of POS tags and also the addition of negation with unigrams are not beneficial for classification of polarity (Go et al., 2009). Ultimately, the NB classifier was identified as the most accurate machine learning method with an accuracy of approximately 83 percent.

Another outstanding study about TSA has been conducted by Pak and Paroubek (2010), which had similarities to the Go et al. (2009) since both used emoticons as labels to illustrate polarity of Twitter posts. Contradictory, Pak and Paroubek (2010) utilized a multiclass classification by adding a neutral polarity to the analyzed content and also used more features to establish the performance of their classifiers. The results displayed an increase in performance when more training data was used and the best performer was a MNB in conjunction with n-grams and, contradictory to Go et al. (2009), the use of POS tags.

Another great contribution to this specific field of study has been the findings by Barbosa and Feng (2010), who utilized three different sentiment detection tools to explain a collection of Tweets. Similar Tweets with different sentiment polarity scores, as a result of the alternative sentiment detection tools, were filtered out. The final training dataset consisted of approximately 200.000 Tweets. The classifiers were trained on different features in comparison to the aforementioned studies, Barbosa and Feng (2010) used syntax and meta-index features. The SVM classifier showed the best results: for subjectivity detection an accuracy score of circa 82 percent has been achieved, which was primarily due to the syntax features. Also an accuracy score of roughly 81 percent on polarity detection, which was mainly due to the meta-index features, was obtained. Syntax features involve for instance reTweets, hashtags, and URLs, and meta-index features include POS tags and polarity of words identification.

Studies by Wiebe et al. (1999), and Wiebe et al. (2005) have resulted in widely adopted use of lexicon resources for meta-index features such as the MPQA lexicon and SentiWordNet (Esuli and Sebastiani, 2006).

The SVM classifier is extremely common in tackling TSA. This has been illustrated by Bakliwal et al. (2012), Mohammed et al. (2013), Kiritchenko et al. (2014) and Ligthart et al. (2021) who addressed TSA with the SVM classier as common denominator. Bakliwal et al. (2012) applied the SVM classifier in combination with 11, one by one pre-processed, features in order to exemplify their effectiveness. Mohammed et al. (2013) also utilized many different features and Kiritchenko et al. (2014) used a linear-kernel SVM classifier as well as a MaxEnt classifier together with a selection of semantic, sentiment and surface form features. Ligthart et al. (2021) identified the SVM classifier as the most used supervised machine learning method in TSA. The SVM classifiers trained on multiple features outperformed other classifiers as well as SVM classifiers trained on unigrams. Accuracy scores were used to measure the performance.

Asiaee et al. (2012), offered a three-step ‘cascade classifier for TSA’ (Figure 3) with three sequential two-class classification steps. The first step aims at deriving Tweets based on the subject of interest, subsequently sentiment of Tweets is identified, followed by assigning sentiment polarity to the Tweets. Moreover, new classification methods, such as weighted SVM and k-Nearest Neighbors (KNN), were proposed in this study next to common classification approaches. The results have implied computational advantages if a lower-dimensional space is used, simultaneously the performance is not substantially affected.

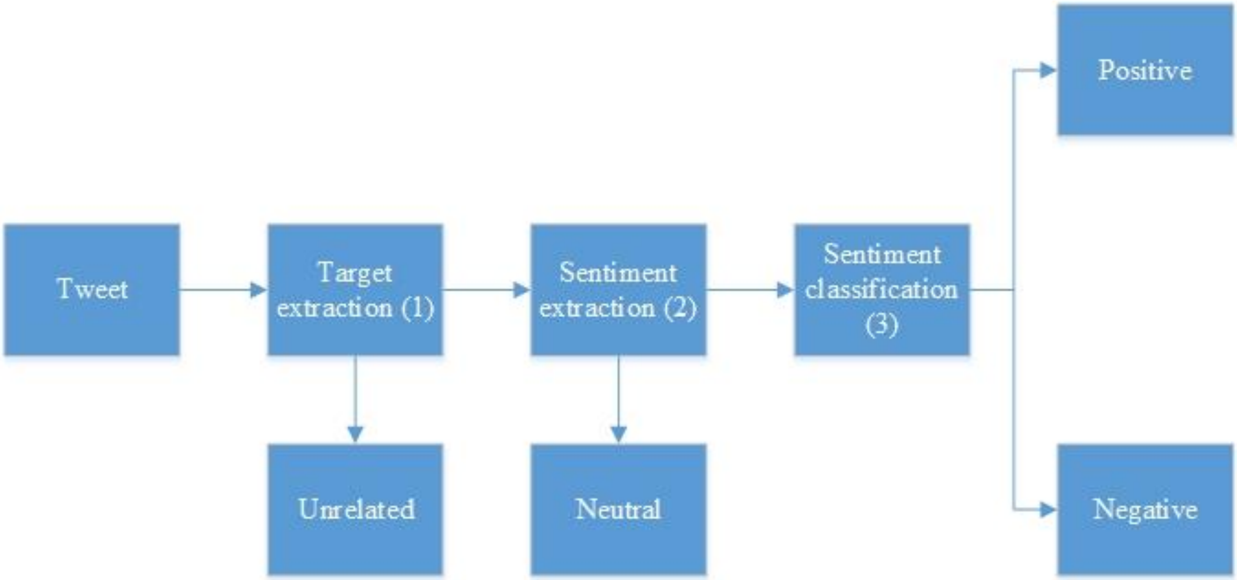


Figure 3 Cascade classifier for TSA extracted from Asiaee et al. (2012)

It is clarified that feature selection is crucial for the performance of supervised machine learning methods. Giachanou and Crestani (2016), identified several studies that focused on the impact of a variety of features in TSA (Appendix X). Three primary methods were identified and compared these; a baseline method trained on unigrams, (partial) tree kernels, and models based on features. The baseline method was least performing and the features that refer to the term sentiment polarity were most beneficial in the light of TSA. One of those studies was the study by Aisopos et al. (2011) which proposed the use of n-gram graphs to improve the accuracy in Multinomial MNB (MNB) and C4.5 classification methods. The C4.5 method, a DT classifier which can be applied to produce a decision based on a specified sample of data, was the best performing model (Siahaan et al., 2019). Kouloumpis et al. (2011), achieved the best performance through the use of microblogging, lexicon and n-grams features. POS was not identified as an adequate feature for sentiment classification. Using a NB classifier based on semantic features, Saif et al (2012) attained a higher F-harmonic accuracy score over POS and unigrams features for both positive and negative sentiment classification. Hamdan et al. (2013) managed to achieve a higher performance expressed in an increase of F-measure accuracy by correspondingly two and four percent utilizing senti-features from SentiWordNet, verb groups and adjectives from WordNet, and concepts from DBPedia. Additionally Aston et al. (2014) used different (a combination of) supervised algorithms with limitations on processing time and memory in order to increase performance using evaluation metrics to identify the top features.

Recently, Carvalho and Plastino (2021) have published a subsequent paper dealing with research on feature and meta-level features in TSA. In the article of Carvalho and Plastino (2021), a systematic analysis of employed types of features, n-gram language model, word embeddings and meta-index features in TSA is exhibited. Although, the results indicated that these types of features can attain higher performance scores if used in combination with SVM, Logistic Regression (LR), and Random Forest (RF), word embedding and n-gram features were the lesser performers in comparison to meta-features. Carvalho and Plastino (2021) also concluded that the combination of the meta-features and n-grams perform statistically better in sentiment classification of Tweets.

From the relevant literature it can be stated that social media platforms as Twitter can be used as a reliable source of information to help researchers detect and extract people's sentiment. As a result, using Twitter to analyze the (underlying) sentiment in event, business or location-related posts could assist significantly to the detection and prediction of location related sentiment. Accordingly, this could lead to an improved planning process which is more aligned to customer needs. The review of the literature indicated a wide variety of research analyzing sentimental features in the detection of location related sentiment in order to detect and predict potential minor incidents that can be categorized under the picket regulation. Hence, this study performs the most used unsupervised and supervised machine learning techniques to identify the role of sentimental features and the subsequent SA with the aim to detect and predict potential events in a certain area to streamline the associated planning process in an independent government authority. A framework for SA techniques is indicated in Figure 4 and will be used as a benchmark towards TSA in this study.

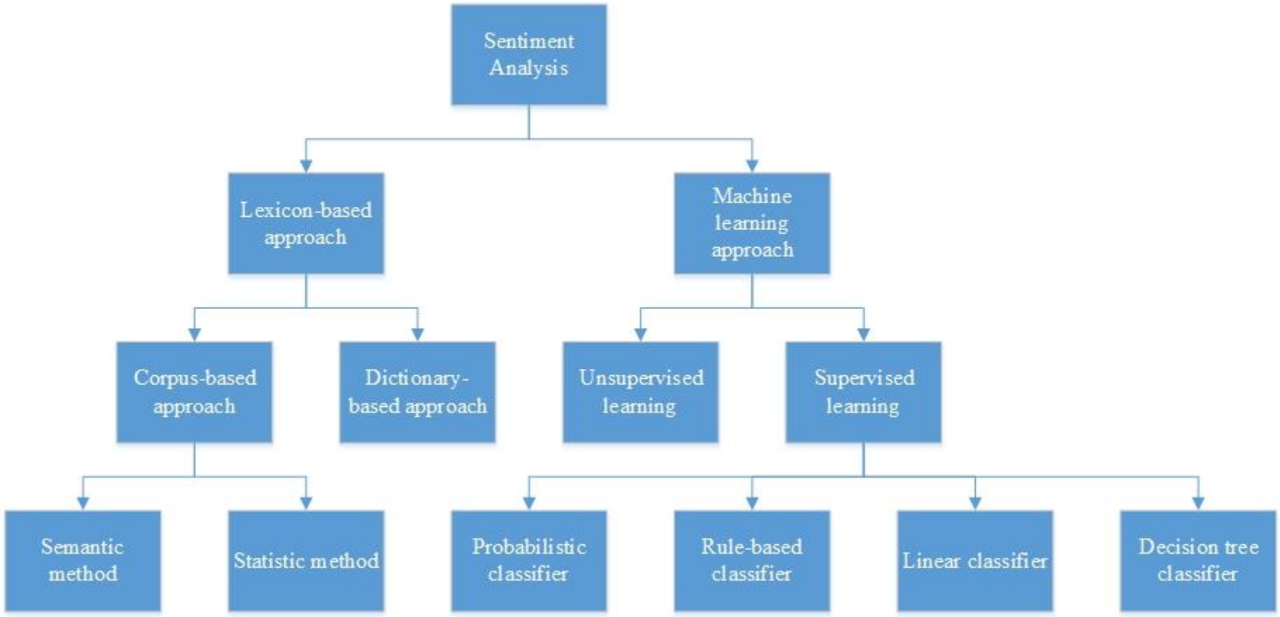


Figure 4 Sentiment analysis techniques extracted from Kaur and Sharma (2019)

According to Giachanou and Crestani (2016), Kumar and Jaiswal (2020), and Ligthart et al. (2021) are SVM, NB, LR, DT and EM the most frequently used supervised machine learning techniques with the highest associated overall performance in TSA (see Table 1). In consequence, based on the relevant literature, the most popular supervised machine learning classifiers will be described in the methodology section and eventually applied in this research to establish the best approach for predicting the public opinion, which potentially affects the picket process.

Author	Classification approaches								
	SVM	NB	NN	LR	ME	KNN	RF	DT	EM
Agarwal et al. (2011)	X								
Ahmed and Salim (2013)	X	X	X						X
Ahamad et al. (2018)	X	X		X			X		
Aisopos et al. (2011)		X						X	
Al-Moslmi et al. (2017)	X	X	X						X
Asiaee et al. (2012)	X	X				X		X	
Aston et al. (2014)				X				X	X
Bakliwal et al. (2012)	X	X							
Barbosa and Feng (2010)	X								
Bermingham and Smeaton (2010)	X	X							
Davidov et al. (2010)						X			
De Oliveira et al. (2018)	X	X		X				X	X
Genc-Nayebi and Abran (2017)	X	X		X				X	
Go et al. (2009)	X	X							
Hamdan et al. (2013)	X	X							
Jiang et al. (2011)	X				X				
Kasmuri and Basiron (2017)	X	X		X				X	
Kiritchenko et al. (2014)	X								
Kouloumpis et al. (2011)									X
Kumar and Garg (2019)	X	X	X	X	X		X	X	
Kumar and Jaiswal (2020)	X	X	X	X		X		X	X
Kumar and Sharma (2017)	X	X	X		X			X	
Madhala et al. (2018)	X	X			X				
Mite-Baidal et al. (2018)	X	X			X				X
Mohammed et al. (2013)	X								
Pak and Paroubek (2010)	X								
Qazi et al. (2017)	X	X			X	X			
Salah et al. (2019)	X	X	X		X			X	
Saif et al. (2012)		X							
Shayaa et al. (2018)	X	X						X	X

Table 1 Most common machine learning approaches in TSA

Chapter 3 Methodology

This section explains the methodology that will be used to answer the research questions. A thorough explanation of the required models will be presented. The goal is to build a supervised-learning classifier that can, to a certain degree, accurately classify Tweets into sentiment categories. After the preparations, an explorative data analysis will be provided.

3.1 Classification methods

Jurafsky and Martin (2017) recognized classification as a popular method for SA in order to establish a sentiment trend. The aim of the project is to identify the most suitable classification approach for the case company. At first a simple model will be applied, followed by more complex supervised machine learning approaches. According to Michie et al. (1994), classification adopts observations, obtains features from these observations and subsequently assigns classes to new observations. In this project the most common approach to classification; supervised learning methods, and traditional methods; lexicon based approaches, have been used. Supervised learning methods involve training sets. Training sets contain data with provided class labels, and the training data helps the classifier to predict labels for new observations (Jurafsky and Martin, 2017). Lexicon based approaches do not require training data, machine learning techniques or model training.

3.2 Lexicon based approaches

Lexicon based classification approaches are exceedingly simple and intuitive. Despite the fact that there exist various dissimilar varieties of the lexicon based classifiers, the approaches are all identical (Ligthart et al., 2021). The method pairs each document, Tweets in the case of TSA, with the sentiment word lists, which usually consist of a negative sentiment list of words and a positive list of words. Consequently, the classifier amounts the number of pairs for the negative and positive sentiment, and after this the sentiment scores are added up. Ultimately, the lexicon based classifier evaluates the sentiment scores for each class and classifies the Tweet based on the highest scored sentiment. The class is labelled neutral when there is not a word in the Tweet with sentimental value or the final compounded sentiment score is equal to zero. Note that words may have different weights and therefore can have a different impact on the overall sentiment of the document. For instance, ‘John is a good person, but Joey is better’ highlights the importance of the weight factor in SA as ‘good’ and ‘better’ both involve positive benefit but the latter is clearly meant to be more impactful.

In this research two common lexicon based approaches in TSA have been applied. VADER is one of the most generalized rule-based model for SA of social media text. VADER, for Valence Aware Dictionary for sEntiment Reasoning, is a well-documented open source model specialized in social media SA (Hutto and Gilbert, 2014). An alternative open source tool to VADER and often used in TSA in combination with Python is the Natural Language Toolkit (NLTK) created by Loper and Bird (2002). NLTK allows for computational linguistics and it is at the core of many Natural Language Programming (NLP) systems (Loper and Bird, 2002). Moreover, it is increasingly used in TSA as it allows to be built upon by other libraries just as TextBlob (Loria et al., 2014, Sazzed and Jayarathna, 2021). TextBlob is a Python library that helps to perform many NLP tasks, such as SA (Loria, 2018).

All in all, it can be concluded that there is an abundance of unsupervised approaches to TSA and there exist a great number of contradictory findings towards the use of lexicon and rule-based methods. These approaches, categorized as unsupervised learning methods, are highly dependent on the domain of interest and the form of language used (Ligthart et al., 2021; Zahoor and Rohilla, 2020). Moreover, Ribeiro et al. (2016) identified high fluctuations in performance per SA-tool used in relation to the data enquired. A primary reason for the fluctuated results was assigned to the static characteristics of lexicon based approaches. Additionally, language is ever evolving and dynamic, in particular in the social media domain, hence the development of lexicon based approaches takes the form of a similar exponential trend as there are publications in the TSA domain, where also existing lexicon and rule-based approaches have proven to iterate over time to expand existing dictionaries (Ligthart et al, 2021). TextBlob and VADER are such existing dictionaries. Both classifiers operate in a similar manner and both offer a list of features, therefore both have been applied in this research in order to demonstrate the most suitable lexicon based approach for the case company. A general lexicon approach is indicated in Figure 5.

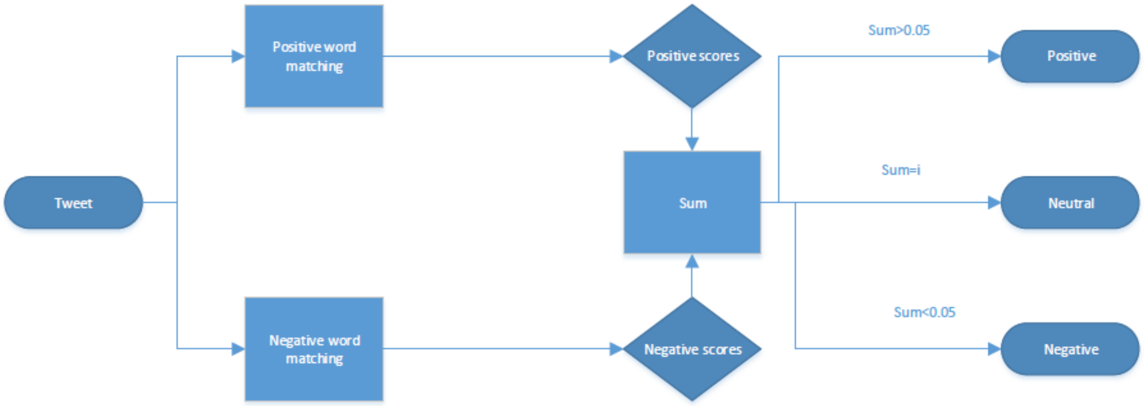


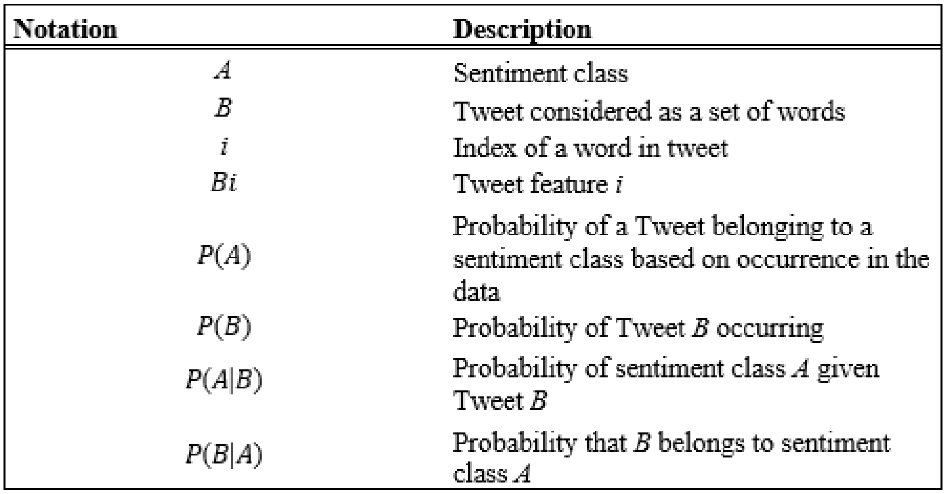
Figure 5 A general lexicon approach

3.3 Probabilistic classification method

The starting point for the supervised machine learning algorithms is the NB model. The NB method, also referred to as independence Bayes or simple Bayes, is a probabilistic classification algorithm. It is deemed as a relatively easy to build, fast, simple and versatile classifier since it does not require any complicated iterative parameter estimation schemes (Kabakchieva, 2013). The classifier is simple but also renowned to outperform more complex and sophisticated algorithm (Zhang, 2004). Moreover, it is valuable for application to both large and small datasets and it is a common classifier in the field of text mining (Chen et al., 2009). Bakliwal et al. (2012), Go et al. (2009) and Neethu and Rajasree (2013) achieved relatively high accuracy results of approximately 85 percent using the NB classifier, instigating an expectation of successful implementation and high results. Thus it can be stated that the NB classifier may be a suitable classification approach to determine the public sentiment.

The NB classifier algorithm is an algorithm that assumes a simplistic state of the document it aims to classify (Liu et al, 2013). The simplistic view relies on the text within the subjected document to be in an arbitrary form without any particular word order or grammar (Jurafsky & Martin, 2017). The NB algorithm name stems from the assumption that all the input, in this research the extracted Tweets, is independent of one another.

The foundation of this algorithm is the Bayes theorem. The probability is represented by $P()$, the class referred to with A and the text, words, of a particular Tweet regarded as B. As such, $P(A|B)$ denotes the probability of class A given Tweet B. The algorithm is fed with the probability of the class to occur in the extracted dataset, is stated as $P(A)$. Furthermore, the probability of a Tweet belonging to a class, while the Tweet itself is given, is denoted as $P(B|A)$. Subsequently the total probability of the Tweet being present, is $P(B)$. Table 2 shows the notation of the NB algorithm.



Notation	Description
A	Sentiment class
B	Tweet considered as a set of words
i	Index of a word in tweet
Bi	Tweet feature i
$P(A)$	Probability of a Tweet belonging to a sentiment class based on occurrence in the data
$P(B)$	Probability of Tweet B occurring
$P(A\|B)$	Probability of sentiment class A given Tweet B
$P(B\|A)$	Probability that B belongs to sentiment class A

Table 2 Notation of the NB algorithm

$$P(A|B) = \frac{P(B|A)\ x\ P(A)}{P(B)}$$

In Bayesian theorem, *P(A)* is referred to the prior probability of proposition, *P(B)* is the prior probability of the event A given the evidence of B, and therefore referred to as the prior probability of evidence. $P(B|A)$ is referred to as the likelihood. Ultimately, the following formula can be created:

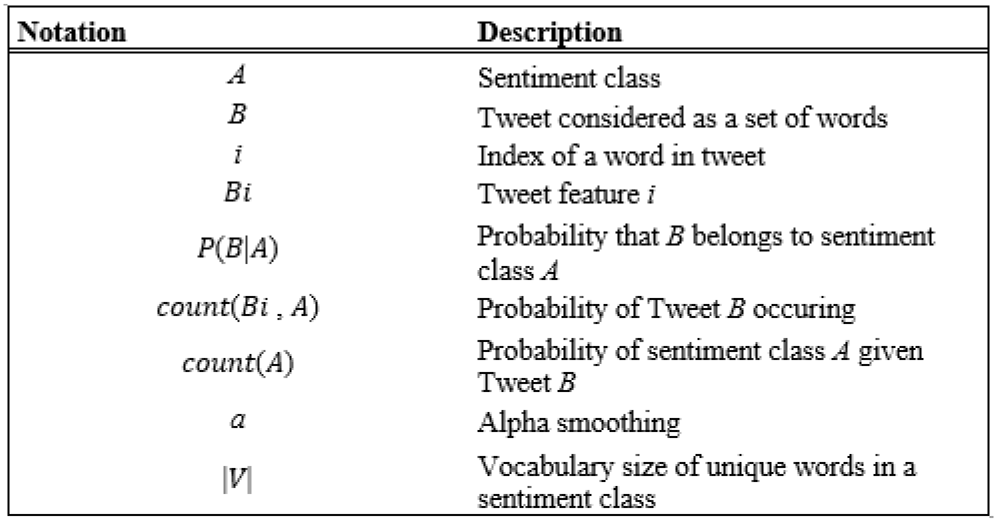
$$Posterior = \frac{Likelihood\ x\ Proposition}{Evidence}$$

As aforementioned, the word ‘Naïve’ in the NB algorithm implies the independence of the words in the Tweets. However, in practice, this does not sway. For instance, “tennis” and “Roland Garros” could appear in a Tweet, but the independence assumption claims every word in the Tweet (represented by B_i) to be impartial with regards to any other word in the Tweet (symbolized by B_{i+n}). Hence the following formula can be generated, as solely the words present within the pre-processed Tweets are utilized to determine the probabilities.

$$P(B_i\,|B_{i+1}\,,\,\dots\,,\,B_{i+n},\,A) = P(B_i\,|A)$$

In light of the case, the NB algorithm classifies the Tweets in either “Negative”, “Neutral”, or “Positive” classes. It does so by detecting the frequency of specific words in a Tweet with its equivalent label being “Negative”, “Neutral”, or “Positive”, denoted as $P(B|A)$, and as well by calculating the overall probability of a Tweet being “Positive” or “Negative” without looking at the content of a Tweet $P(A)$. This value is divided by a constant $P(B)$ which is identical for all “Negative”, “Neutral”, or “Positive” Tweets. As a consequence, for this research, the data must take the form of a vector of distinctive words in order to allow for a feature set to be retrieved, which also contains a label of either “Negative”, “Neutral”, or “Positive”. Ultimately, based on these feature sets and according probability, the NB algorithm decides how to classify a newly supplied Tweet.

In SA, the MNB is often preferred over the NB algorithm (Jurafsky & Martin, 2017). The MNB is more extensive as the algorithm does the exact same as the NB, except it takes not only the words into account but also the word frequencies. This difference is translated to the calculation of the likelihood, denoted by $P(B|A)$. The notation of the MNB algorithm is provided below:



Notation	Description
A	Sentiment class
B	Tweet considered as a set of words
i	Index of a word in tweet
Bi	Tweet feature i
$P(B\|A)$	Probability that B belongs to sentiment class A
$count(Bi\ , A)$	Probability of Tweet B occuring
$count(A)$	Probability of sentiment class A given Tweet B
a	Alpha smoothing
$\|V\|$	Vocabulary size of unique words in a sentiment class

Table 3 MNB algorithm

The mathematical equation for the MNB algorithm is denoted by:

$$P(Bi\ |A) = \frac{count(Bi\ , A) + \text{a}}{count(A) + |V|}$$

Count(B_i,*A*) where B_i is a Tweet feature i, representing a word feature of class *A*. Another difference compared to the NB algorithm is the use of the smoothing parameter, labelled as a. The alpha ensures the prevention of zero probabilities when a word is not present in the training data. *Count(A)* denotes the total quantity of words in a class, and $|V|$ represents the quantity of unique words in a class.

The MNB algorithm determines the probability of a Tweet to a particular class by calculating the probability of the presence of that class in the provided data. The word frequency and meaning of words help to calculate the probability of a specific class. The highest prior probability is leading if it is unclear to which class a Tweet needs to be assigned.

3.4 Support Vector Machine classification

SVM classification algorithm is a classification approach for both nonlinear and linear data. The SVM classification technique is characterized by hyper plane in high-dimensional space to separate classes (Meyer and Wien, 2015). SVM's have been a recognized algorithm known for accurate and robust performance methods among all well-known supervised machine learning algorithms. The aim of SVM is to distinguish members of the two classes, or more in case of a multi-class classification, in the training data in order to identify the best classification. Furthermore, SVM maximizes the margin between the classes to ensure the identification of the best function. Hence, it achieves not only high classification performances, but also gives opportunity for the correct classification of the future data.

To clarify how the SVM algorithm operates, assume to work with a two dimensional dataset, denoted by x_1 and x_2. The SVM method searches for the best hyperplane that can attain the highest separated margins (Kumari and Chitra, 2013), where the separation line with the highest margin is denoted by:

$$w \cdot \mathrm{x} - b = 0$$

The w represent the weight vector to the hyperplane while b denotes the offset known as bias (Kumar and Chitra, 2013). Moreover, if the data is linearly dividable, two hard margins can be drawn parallel to the hyperplane, of which the support vectors of each class fall on the line of the hard margins (Kumar and Chitra, 2013). This is depicted in Figure 6:

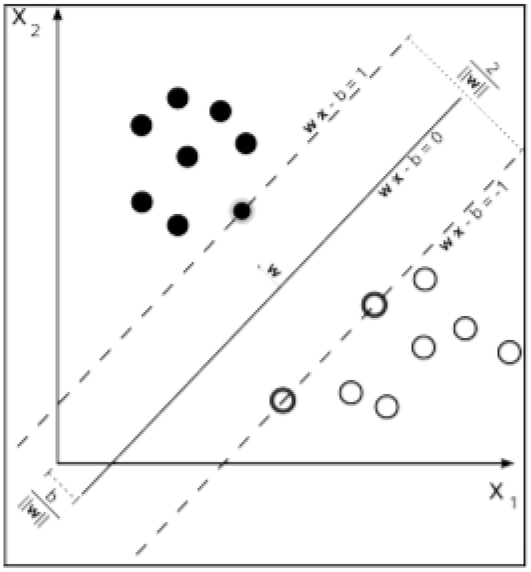


Figure 6 Extracted from Kumar and Chitra (2013)

It is common in SVM classification algorithms to include a hinge loss function, denoted by *C*, to form a maximum margin regularization parameter (Barlett et al., 2008). The notation of the SVM parameters are depicted in Table 4.

Notation	Description
w	Weight vector to the hyperplane
B	The bias term
C	Regulization parameter
x	Input value

Table 4 Notation of the SVM algorithm

In the model scenario the classes are linearly dividable. In practice, and for the case in this research, the classes are more frequently not linearly dividable. In such context the side on the plane determines the class to which the observation is assigned. Nevertheless, it is not a single plane that separates the datasets, an entire group of planes do. The group of planes is determined by a higher order function to split the dataset. A kernel trick is applied to attain a non-linear SVM classifier. The function maps the points of the non-linear dataset, using for instance the square (root), to data points in a linear dataset. Theoretically, a non-linear SVM classifier would look like Figure 7. Nonetheless, datasets consist of noise and random errors which need to be taken into account to prevent overfitting of the model in the training set. An adequate balance between errors and overfitting needs to be made in order for the model to still produce generic results.

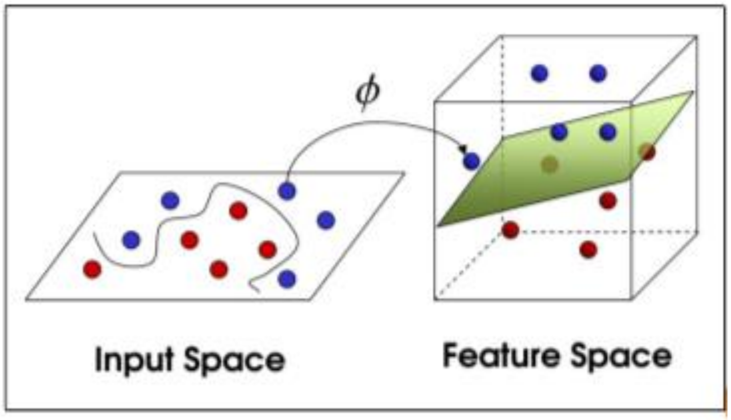


Figure 7 Theoretical non-linear SVM classifier

In this research there are three features (positive, negative and neural), and as a result the SVM is involved with a three dimensional space with a hyperplane. In this three dimensional case the hyperplane transforms data into another dimension, also exemplified by Figure 7, so that it can be classified (Meyer and Wien, 2015).

3.5 Logistic regression classification

The logistic regression classification algorithm, similar to linear regression, regresses on probabilities of labels. LR used logistic function to predict the probability of a class, because it is the easiest learning construct to use when the target variable is prescriptive. The linear regression fits a single straight line to the data, where the LR algorithm fits a S-shaped curve to the data, listed as the sigmoid or the logistic function (Daumé III, 2012). In a linear model (e.g. linear regression), a gradient descent can be applied to minimize the errors. The gradient descent calculates the difference between actual values and predicted values, of which all terms are squared and divided by total number of term (Daumé III, 2012). Thus the distance is either zero (correct prediction) or one (incorrect prediction). This is called the zero-one loss, indicated in Figure 8.

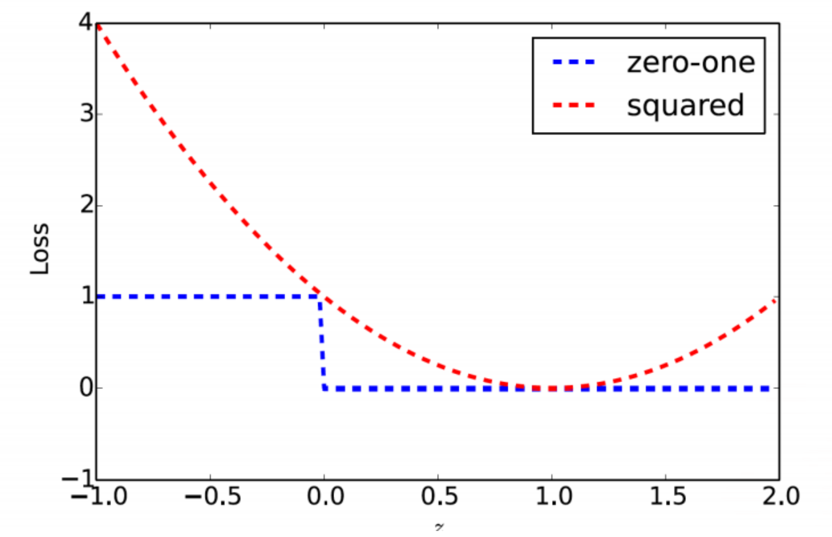


Figure 8 Extracted from Daumé III (2012)

The zero-one loss function involves the gradient being zero at all places except if the loss is one. However, for the LR algorithm, a better suitable loss function is required as the linear regression is not suitable for classification, since outliers can distort the decision boundary. The better suitable loss function for the LR algorithm is the sigmoid (also referred to as the logistic) function.

The LR algorithm is used to estimate binary values based on a provided set of independent variables. In other words, it predicts the probability of the occurrence of an event by fitting the data to a logistic function (Kleinbaum et al., 2002). For instance, when the dependent variable is binary (a class/ stochastic event), the probability can denoted by $(P / (1-P)) = ax + b$, where P is the probability of the label being positive with a value between [0-1]. The logit function helps to map P to $[-\infty, \infty]$. The notation of the LR algorithm parameters is provided in Table 5:

Notation	Description
P	Probability of class being positive
$1-P$	Probability of the class being negative
w	The weight vector
x	The input value
b	The bias term
σ	Sigmoid function
$\sigma(z)$	The inverse-logit function

Table 5 Notation LR algorithm

In order to attain the probability of a positive class, the following equation is applied:

$$f\,(x,\ w,\ b) = \sigma(wx + b)$$

Where *x* denotes the feature vector, *w* represents the weight and *b* signifies the bias respectively (Daumé III, 2012). The parameter σ is incredibly important and denotes the special function called sigmoid (or logistic) function, $\sigma(z)$ is labelled the inverse-logit function, signified by logit−1 (z) and $\sigma(z)$ is characterized as the following formula:

$$\sigma(z) = \frac{1}{1 + exp(-z)} = logit^{-1}\,(z)$$

Ultimately, the LR classification algorithm takes a linear classifier and applies an extra sigma function to predict the probability of a class by:

$$Ppred = \sigma(wx + b)$$

3.6.1 Ensemble classifier algorithm

Ensemble algorithms learn based on several other models. The ensemble model is employed to counter data sensitivity and inflexibility (Ligthart et al., 2021). The most common approaches are boosting and bagging, which train models in a sequential and the parallel manner. When boosting is applied, models take into account the mistakes made by the preceding model. In terms of bagging, the models are trained using a random subset. In this research, the extra trees classifier algorithm, adaptive boosting and gradient boosting classifier have been identified as ensemble methods.

3.6.2 Extra tree classifier

DTs are supervised machine learning methods to tackle classification problems. These algorithms work on both categorical and numeric data. The aim of the algorithm is to split the sample into two or more homogenous subpopulations based on the most significant splitter in the input variables. DTs consists of nodes, decision points within trees that contain data, edges, the connection in between nodes, and leaves, terminal nodes which represent the predicted outcome (Daumé III, 2012). DTs work particularly well with non-metric data. Similar to other pattern recognition methods, it works well with real value data. They can be regarded as a list of tests, organized in hierarchical structures, which can be used to classify objects. Machine learning in decision trees is about constructing that tree.

Simplified, it is a tree structure with decision rules. DTs are made based on training data in a top-down, broad-to-specific direction (Apté & Weiss, 1997). DTs are regularly employed to detect the most ideal strategies to attain a specific target in real-world settings, since the output is relatively straightforward transformable in a step-by-step manner. DTs have been discovered as proper predictors for determining sentiment of new Tweets and as a result will be used in this research (Wakade et al, 2012; Zuo 2018).

RF is an algorithm in the domain of both regression and classification. The RF is a tree-based model that merges DT to provide overall accurate predictions. RF utilizes bagging techniques (which uses samples of bootstraps) in order to produce the number of DTs. Subsequently RF conglomerates the decision trees predictions based on majority voting criteria. Ultimately, the final predictions will encompass the class with the most votes. The notations for RF are shown in the similarly labelled Table and the according mathematical equations are the following:

$$P = mode\ \{T_1(y), T_2(y)\ , \ldots\ , T_m(y)\}$$

$$P = mode\ \{\sum\ T_m(y)\}$$

P is the probability, which is the result produced by the classification algorithm for the data passing through the trees *Tn*.The difference of the RF algorithm compared to a simple DT algorithm is the use of randomness towards the training data (and a subset of features). The degree of accuracy and overfitting are reduced due to the decreased chance of high variance by the randomness of the RF algorithm.

Notation	**Description**
P	Probability of a specific class
Tm	The number of decision trees

Table 6 Notation RF algorithm

Comparable to a RF classifier is the Extra Tree (ET) classifier, which is also referred as the Extremely Randomized Trees due to random selection of splits and features (Pal, 2005). The ET algorithm implies that random splits are selected from randomly selected characteristics of which the best split is selected in order to separate a node in groups. Hence, ET classifiers test random over fraction of features and are therefore easier to train. Furthermore, ET classifiers help to increase predictive performance by mitigating overfitting of a single node.

The primary difference exists within the decision to split nodes and the usage of input data. The ET classifier splits randomly and uses the entire dataset, while the RF classifier determines the optimum split and utilizes bootstrapping (Oshiro et al., 2012). The ET classifier reduces bias whereas the RF algorithm reduces variance. For this work, the ET algorithm has been chosen over the RF algorithm due to the computational advantages.

3.6.3 Boosting classifiers

Gradient Boosting (GB) classifier aims to convert weak learnings into strong learners by fitting a novel tree on an altered version of the original dataset (Athanasiou and Maragoudakis, 2017). The GB algorithm can be best understood if the AdaBoost (AB) algorithm is explained first. The AB algorithm involves the training of a DT while assigning equal weight to the observations. Weights are altered based on classification performance, where easy to classify observations receive lower weights and observations that are hard to classify experience an increase in weight (Rathi et al., 2018). The subsequent tree is fitted on the data with modified weight and this is a recurrent process for a definite number of repetitions. The final predictions entail the weighted sum of the previous trees.

The difference between the AB classifier and GB classifier is the manner in which the models identify the limitations of weak decision trees. GB algorithm uses gradients, the alteration in weight in relation to the alteration in error, in the according function of weak decision trees. The function looks like:

$$y = ax+b+e.$$

Here *a* denotes the alpha, *x* symbolizes the input value, *b* is again the bais and *e* represents the error term. Contrary to general error terms, the GB error term allows one to have more control in optimizing the function and subsequently correspond to practical applications. The notation for the GB algorithm is denoted in the Table 7:

Notation	Description
y	Output value
a	Alpha smoothing
x	The input value
b	The bias term
e	The error term

Table 7 Notation GB algorithm

At first, in AB classification, weight is assigned to the trained classifiers based on the accuracy of the previous training, and subsequently the better performing classifiers receive higher weight in order to impact the result. The updated weight is assigned using the following equation:

$$D_{t+1}(i) = \frac{D_t(i)\ exp\ (-a_t y_i h_t(x_i))}{Z_t}$$

$D_t(i)$ denotes the prior weight, a_t is the weight assigned to the classier based on the error rate, h_t represents the output of the classier for the input value of x_i and Z_t is the sum of weights and normalizes the total number of weights. The notation for the AB classifier can be seen in Table 8:

Notation	**Description**
$D_t(i)$	Prior weight
a_t	The assigned weight for e
y_i	Input parameter
h_i	The output value for xi
x_i	Input value
Z_t	The sum of weights

Table 8 Notation AB algorithm

3.7 General framework for a comparative TSA

All in all, based on all aspects of TSA in the previous sections and the identified gap in research of TSA with a focus on the external company's its domain, a framework has been designed to schematize comparative TSA with a focus on the case company its needs. First a simple representation of this framework is given (Figure X). The requirements for TSA need to be identified in order to determine the most suitable TSA framework. If the most suitable lexicon based approach to SA is identified within this case, the performance of the machine learning models will improve as the scale of the data increases.

Key steps in data understanding, preparation and modelling are displayed in the framework (Figure X). The proposed comparative TSA framework is evaluated by the extent to which its fits the case company its needs. In order to serve that purpose, this thesis characterizes TSA as the mining of Twitter data for information in relation to what people feel and think with a regional focus (Kouloumpis et al., 2011). Moreover, metrics, to be discussed in the next chapter, will be used to evaluate the performance of the supervised machine learning approaches, listed in previous chapter, based on different features and lexicon approaches, in order to ultimately distinguish the most suitable model for mining Twitter data in the light of the case company its requirements. These requirements may be fulfilled by location based sentiment analysis. Ultimately, this leads to the following framework:

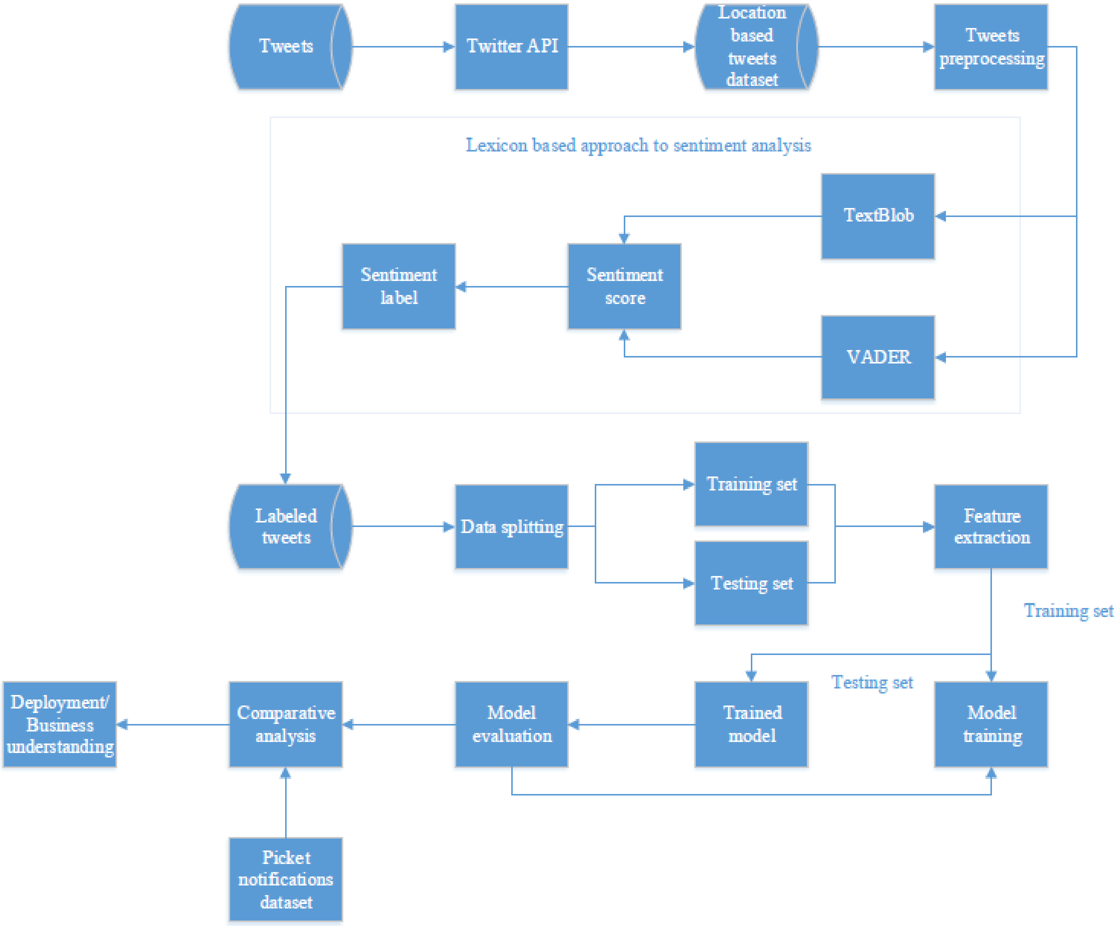


Figure 9 Framework for TSA in this thesis

Chapter 4 Experimental Setup

This chapter will start with an introduction to the datasets and how the data has been gathered. Followed by the data collection and consequently the preprocessing steps are discussed. Moreover, lexicon based approaches, feature selection and the classification algorithms are discussed. Finally the metrics to evaluate the algorithms are presented.

4.1 Programming software and environment

The algorithms used in this work have been developed with the help of the Python 3.8 open source software. The simplicity regarding documentation, replicability and usability convenience, and recognition within the scientific community were decisive in the choice to use this software. Moreover, due to the wide adoption of the language in the scientific community, there exist countless available libraries that support functionalities within this language. In this research, for instance, common libraries have been called upon, such as: Sklearn, VADER, NLTK, TextBlob, Pandas, and Numpy (Buitinck et al., 2013; Hutto and Gilbert, 2014; Loper and Bird, 2002; Loria, 2018; McKinney, 2010; Oliphant, 2006). The full list of packages and libraries can be found in the Appendix.

While Python code can be written within many different environment and platforms, the code for this research has been written using the open source platform Anaconda (Yan and Yan, 2018). Anaconda allows for simple package and library deployment and management, collection of data from several sources, instant running of code (which makes it easier to track bugs), and to top it off, it is free.

4.2 Data collection

In this paper, two datasets have been used. One dataset has been provided by the external company SV. The second dataset has been extracted directly from Twitter using the Twitter API.

The first dataset has been provided by the case company and it consists of the relevant picket data. It consists of 16 columns and 11846 rows, which has been reduced to three columns (date, region and notifications) in order to establish a causal relation between the location based sentiment and the number of notifications send from that region. The number of notifications per region have been visualized in Figure 10, in order to demonstrate the origin of most notifications. The cities that send the most notifications to the central picket department are used as a source, with the help of longitude and latitude coordinates, for the extraction of Twitter data.

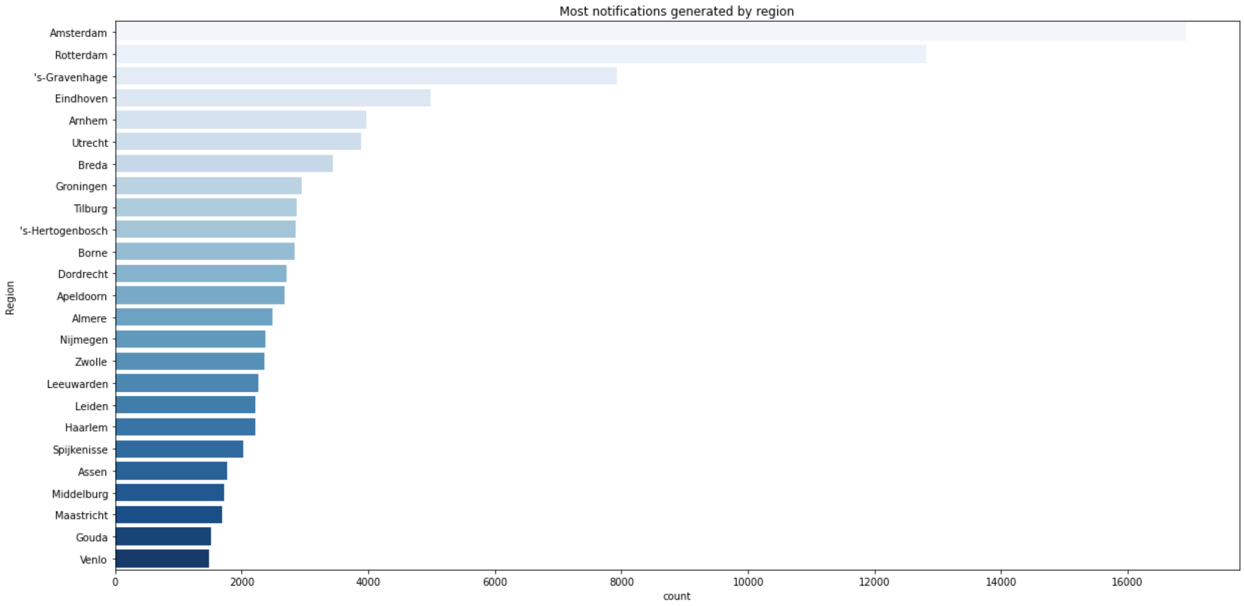


Figure 10 Most notifications per region

The second dataset consists of these extracted Tweets. The second dataset needs to be labelled to ensure training data for the classifiers. However, if the results diverge significantly, an error has occurred and the assumption can be made that a human error is the cause. After labelling, the data is pre-processed.

Twitter gives access to its platform to all. There are several APIs; the Twitter Search API, Twitter Streaming API and the newest, launched in January 2021, the Twitter API for academic research (developer.twitter.com, 2021). The Twitter API its primary goal is to allow the use of Twitter data in order to simplify the programmatical application and ultimately impact of Twitter data in the world. The latest API permits the extraction of 10 million tweets per month and is entirely free, if one is provided with licensed access. Overall, the Twitter API is quite successful and it is exponentially used in academic research, specifically in TSA. Unfortunately, the academic research product track only allows the use of this API in Postman, a collaboration platform for API development for now (Postman.com, 2021), therefore it is not used in this research.

The Twitter Stream API, which is part of the REST API, provides people with low-threshold access to query tweets against the indices of popular or recent Tweets. As a consequence, one can extract historical data with keywords and other features. However, for this particular research, the Twitter API has been accessed using a different approach than key word search. Considering the fact that a vast amount of data was needed to produce appropriate results, the Twitter Stream API was used to retrieve tweets by geolocation. The key word search could have caused ambiguity. For example, searching Tweets based on the key word ‘corona’ or ‘covid19’, might have resulted in collecting tweets that convey negative laden sentiment. Moreover, searching tweets based on terms used in the picket process would have resulted in a biased dataset due to the lack of diverse input. Hence, in this work, the public opinion is evaluated using data extracted from Twitter based on geographical location. The regions that send the most notifications to the central picket department have been identified as the geographical locations of interest in this research. Those regions are the five biggest cities in the Netherlands (when Arnhem is not taken into account). Given the fact that the case company its needs encompass the evaluation of the public's opinion and its relation to the number of picket notifications, and even though there exist several options to extract Twitter data, this was evaluated the best option to serve the case company its needs, as picket notifications can be traced back to a regional area. The code for extracting Twitter data using the Twitter API can be found in the appendix.

Despite the fact that there are many programmatical possibilities with Twitter data, only the text of tweets have been studied for sentiment analysis. The retrieved language of the text within Tweets and the actual extraction of Tweets formed considerable constraints. The text needed to be translated to English without the meaning of the text to alter. The Google Translate Python package has been proven to be a valuable tool for researchers in order to conduct a comparative analysis (De Vries et al., 2018). However, this tool does not account of typos and sarcastic or cynical phrases. Therefore, the language constraint is dealt with by only extracting English written Tweets. Second, the Twitter Search API only returns a maximum of 900 Tweets per 15 minutes for a single query. Therefore, a loop has been built to repetitively keep extracting Tweets for the city based on the provided coordinates. It took several days to extract the vast amount of Tweets since only 1-2 percent of the Tweets are geotagged (Schlosser et al., 2021). The total number of Tweets for all five cities in the second dataset add up to 100103 and have been stored in .csv file. The .csv file contains both Tweets and Retweets. The relevant Tweets have been labelled positive, neutral or negative, whereas the irrelevant Tweets have been excluded.

4.3 Data pre-processing

For data analysis, data pre-processing is absolutely essential as the raw, unstructured data is not ready for a sophisticated analysis. The unstructured data includes duplicates, noise, inconsistencies and useless information (Khan et al., 2016). It is especially important in text classification as text data is associated with distinctively different challenges opposed to numerical data. Hence, text data needs to be converted into an analysable format. The data needs to be cleaned in order to be subject to supervised classification methods and model evaluation. Furthermore, Twitter text data regularly contains use of slang, typos, unorthodox symbols and abbreviations, which complicate and deteriorate the performance of the classification approaches. For instance, in the social media domain, users write “Bruv”, “Brat”, “Bro” or “Brother” to say brother, of which the first citations instigate the use of slang, followed by an abbreviation. Although it is relatively easy for humans to understand these citations, it is extremely difficult for machine learning algorithms to comprehend and ultimate deal with. Hence, pre-processing is a key aspect of TSA to convert unstructured data in a different improved format for analysis (Vijayarani et al., 2015).

4.3.1 Tweet duplicates removal

It may occur that similar Tweets are present in the retrieved dataset. The duplicates must be removed to guarantee the uniqueness and objectivity of each Tweet (Go et al., 2009). The datasets are worked with in a Python data frame, which represents the extracted information in the form of columns: username, date, sentiment and text of the Tweet. Furthermore, the structure in a Python data frame permits the search of duplicates using the .duplicates() function. Consequently, the data frame examines for the same rows and drops these, except for the first cases as these are maintained.

4.3.2 Retweet removal

Retweets, referred to as ‘RT’, are reproductions of an already existing Tweet, either from oneself or from a different user. Twitter users frequently add ‘RT’ at the beginning of each Tweet to indicate that one is re-posting an existing message. Retweets influence the TSA negatively by replicating an existing sentiment and therefore increasing its objective weight towards the public sentiment. Thus, it has to be removed from the dataset (Go et al., 2009). The Python data frame searches for ‘RT’ strings at the beginning of Tweets and filter these Tweets out of the dataset. The ‘RT’ search specifically focusses on the capitalized ‘RT’, hence it is not likely that other Tweets will be removed as a consequential side effect of removing Retweets.

4.3.3 Tweet hashtag removal

Twitter is known for its widely adopted use of hashtags. Hashtags instigate a certain theme, topic or feeling, thus it is identified as relevant for the sentiment. Hashtags are removed from the Tweets in order for the subsequent analysis to take the theme, topic or feeling into account. For the data frame to identify these hashtags, it searches for a specific string through the use of regular expressions. Thompson (1968) referred to regular expressions as “a method to locate specific character strings embedded in text”. The data frame searches for the hashtag and removes it from the text, while the theme, topic or feeling remains present.

4.3.4 Tweet username removal

Usernames link the Tweets to specific accounts and hold no additional value. The usernames need to be removed to avoid any processing delays or unnecessary inconsistencies. Usernames are depicted by the

‘@’-symbol and can therefore be easily removed following the similar regular expression method as described in the ‘Tweet hashtag removal’ section.

4.3.5 Tweet conversion to lowercase

All Tweets are transformed to lower case letters to allow the dictionaries to search terms and assign sentiment accordingly. The dictionaries operate best if all tweets are placed in a consistent form (Gull et al., 2016). The dictionaries applied in later stages are case sensitive and look up string of text in lower case letters, subsequently Tweets needed to be transformed to lowercase letters.

4.3.6 Tweet HTML removal

Users often refer to HTML links/tags to redirect readers to other websites or articles. These links do not hold any value towards the sentiment and thus need to be removed. The HTML links all commence with the same letter and can therefore be searched and completely removed using the regular expression method.

4.3.7 Conversion of contracted words

Contracted words entail the merging of two words into one, e.g. ‘I’ll’, which is a combination of I and will. Such contracted words form significant obstacles since the punctuation within the text of Tweet have to be removed and the remaining of the contracted words develop to have a different meaning than intended. Hence, the overall sentiment of the particular Tweet is negatively impacted. Also, the contracted words frequently involve stop words. The stop words are removed from the data using an NLTK called stop words, which is a library full of such semantically meaningless words (Fiori, 2014). Furthermore, the most common contracted words are also listed in a NLTK tokenized toolkit, which has been applied to replace an recognized contracted word (or words) by the nested word in the toolkit.

4.3.8 Punctuation removal

The assumption is made that punctuation, which also involves emoticons, carries no value towards sentiment and it therefore infects the data for the machine learning approaches. Python allows the use of an inbuilt function to identify, with the help of a loop, and remove any punctuation that is located within the Tweet. The inbuilt function replaces the punctuation with a blank space in order to account for the function of the punctuation, which may connect two words, to prevent negatively impacting the data for analysis.

4.3.9 Slang removal

Slang is more recurrent in informal text than formal text. Tweets only contain a maximum of 280 characters and there are not strict guidelines on how to use the Twitter platform. Thus it can be stated that

Twitter is an extremely good source of informal text and therefore also of slang. Wu et al. (2018) demonstrated the use of a Slang Sentiment Dictionary (SlangSD) in combination with several lexicon based approaches. Unfortunately, the SlangSD is not publicly available anymore and is fairly expensive. Therefore alternatives were sought after to tackle the slang in the dataset for this particular research. Nevertheless, some alternatives did not prove to add significant value as internet slang dictionaries and text slang translators such as 'www.noslang.com' only produced accuracy scores of less than 12 percent. The proposed method to remove slang would involve the evaluation of all words in the dataset against the slang dictionary by using an iterative loop over all text, and the matching words would be replaced by the nested words within the dictionary its list. As a consequence, all slang would be moved.

4.3.10 Standardization of words

Due to all the previous text, solely text remains. The subsequent step is the standardization of words. This step in the pre-processing encompasses the search of words with several identical letters in a row, e.g. 'meeee'. The standardization process minimizes all the letters in the sequence to only two of the specific letter, in the example this would result in 'mee'. Although, this is still not the correct spelling, this issue will be solved in the next step. Standardization of words help to accurately classify the text of Tweets by iteratively evaluating the tweets for words with multiple of the same letters in a sequence and reduces them to two.

4.3.11 Standardization of words

Bakliwal et al. (2012), highlighted the importance of correct spelling in order to accurately classify text data. Misspellings can result in wrongful interpretation of words, instigating different semantics which would ultimately affect the accuracy of the machine learning classification approaches. Every word in each Tweet need to be checked by the spelling checker package supported by Python. The spelling checker is a trained corpus, with an accuracy of approximately 80 percent, which replaces misspelled words with words nested in the corpus (Norvig, 2016).

4.3.12 Filter of stop words

Filtering out of stop words has been simplified by the NLTK stop words for Python. The toolkit contains 153 English stop words such as "the" and "a". Stop words are regularly used words that do not hold any value towards the sentiment. The words in the dataset are scanned over an iterative loop and compared to the NLTK stop words list, the words are removed if a match is detected.

4.3.12 Lemmatization

Another common approach used in pre-processing is lemmatization. Again a NLTK can be used to perform this step. Plisson et al. (2004) described lemmatization as the process of obtaining the normalized word. The NLTK wordnet dictionary is used to analyse the words based on its general, normal form. The lemma, the basic form of the word, replaces the several different spellings of a single word which may have slightly different meaning. The variance of the Tweets is reduced by reducing the words with similar spelling to the same lemma, improving the performance of the classification.

4.4 Sentiment analysis using lexicon based approaches

A baseline needs to be established in order to demonstrate and evaluate the most applicable machine learning algorithm for regional based TSA. As aforementioned, two popular and high performing TSA methods are TextBlob and VADER (Hutto and Gilbert, 2014; Sazzed and Jayarathna, 2021). Both libraries are well document, open source and specifically applicable to informal documents. Hence, the decision had been made to establish a baseline with the most suitable lexicon approach (and the MNB classifier). After application of the lexicon based classifiers to the preprocessed dataset, the baseline is determined by evaluation of the class distribution over the entire dataset and distribution per city. The code used for the lexicon based SA approaches can be found in the Appendix.

As indicated by Figure X, all words within each Tweet within the dataset is evaluated and assigned a score. This score is based on earlier determined weight and value which is saved in either the VADER or TextBlob library. The words are assigned a score between [-1 and 1]. This is known as the polarity score. The score is multiplied by its assigned weight and eventually, for all the words in the document, the scores are added up to generate a single output. Consequently, the proportion per sentiment present within the document, in this case each particular Tweet, is computed and represented as output. At first, both TextBlob and VADER do not represent the output in a binary manner, but generate the output in numeric scores between [-1 and 1]. This is the compound score which is converted into one of the three sentiment categories; positive, neutral or negative. Generally, Tweets that exhibit a compound score of lower or equal to -0.05 are assumed to imply a negative sentiment. Thus, such Tweets come to be labelled as negative. Similarly, texts that exhibit a compound score equal or higher than 0.05 get labelled as positive. As such, for every polarity score amid these values, the document is deemed neutral. For every column definition, an if-else statement, which permits to perform a specified function on each cell, has been utilized. In the case of this research, if-else statements loops are used as inputs to define single-lined functions.

4.5 Feature selection

In earlier sections several features have been identified and discussed. Features aid in structuring of the original data (Carvalho and Plastino, 2021). In TSA features are especially important due to the lack of consistency in the format and shape of Tweets. This section highlights the features used in this research.

This research is new with regards to the picket process, therefore only two features have been used in order to set a foundation for future research. One of the explicit features that have been used is the Bag-of-Words (BOW) approach. BOW is seen as a traditional approach and it describes the occurrence of words within a text. It builds a vocabulary of the words and it measures their presence. BOW counts the words or set of words that occur next to each other, also known as n-grams, and creates a numerical vector which form the input feature for machine learning models (Ligthart, 2021). In essence, BOW reviews the frequency of each word in the total number of Tweets. This total number is also referred to as weight.

Another way to transform text, introduced by Salton (1986) is term frequency inverse document frequency (TF-IDF). TF-IDF identifies how often a provided word exists within a text in the corpus and establishes a log-ratio between the number of documents that contain a particular word and the total number of documents (Giachanou and Crestani, 2016; Ligthart, 2021; Salton, 1986). Although, TFI-dF does not capture words across documents, it accounts for common words within a document and the length of the document where BOW does not account for the length of document (Ghag and Shah, 2014). Moreover, TFI-dF prompts less urgency to deal with stop words explicitly since it penalizes frequent words. Therefore, it is useful in search queries and information retrieval as it can rank the relevance of returned results.

In case of TSA, distinctive features can be isolated from the text with the help of BOW and TFI-dF methods. The distinctive features, words, are assigned to specific classification categories using frequency count of words and weight of words. The impact on a particular classification category is valued in a quantitative manner in vector space (Ligthart, 2021). The feature space can be used as input for the machine learning algorithms in order to correctly classify and predict. In this research BOW and TFI-dF will be applied and compared to distinguish which feature helps to attain the best classification performance.

4.6 Classification

Classification will focus on the predictive part of the research. This experiment involves the task of classification based on the identified features in previous section. The task includes 6 classifiers; MNB, SVM, LR, and the ensemble classifiers ET, GB and AB.

The dataset consisting of Tweets has been split up, using train_test_split function from Sklearn package, with a test size of 0.25. As a result, since 69013 Tweets remained after data cleaning and preprocessing, the dataset is split up in a test set of 17253 Tweets and training set of 51760.

Furthermore, the parameters per classifier need to be determined. First, the parameters of the SVM have been determined as the MNB is a simple classifier has not needed any parameter specification, as such the optimization parameter for the SVM is labelled ‘C’ and has been set to 1. Moreover, parameter kernel has been set to ‘linear’ due to associated computation training time. Second, there are parameters that needed to be set for the LR. The parameter labelled ‘C’, which regularizes the parameters value in order to reduce overfitting, has been set to 3, the random state has been set to 1000, to ensure that the splits are reproducible, and the class has been set to ‘multinomial’ as there are three sentiment classes. For the ET classifier and the boosting classifiers the optimum values for the ‘splitter’ and ‘criterion’ parameters have been explored and the assumption is made that there is no general guideline for these parameters. Hence, experimentation is accepted to be suitable. An overview of the machine learning algorithms and their hyperparameters are provided in Table 9:

Models	**Hyperparameters**
SVM	Kernel='linear, C=1
LR	Multi_class=multinomial, C=3
ET	N_estimator=300, random_state=5, max_depth=300
GB	Learning_rate=0.3, random_state=52, max_depth=10
AB	N_estimator=10, random_state=5

Table 9 Machine learning hyperparameters

Hereafter, it has been determined which classification algorithm fits best to establish the public sentiment. It is essential to be aware of the state of the dataset used. The dataset used in this research is imbalanced. This entails that accuracy scores may easily reach high scores, due to the fact that minority classes are scarcely represented (Ganganwar, 2012). Ultimately, to generate more reliable results, multiple evaluation metrics has been applied.

4.7 Evaluation Metrics

The aim in this paper is to classify the public opinion expressed in a tweet, hence the problem in this research, and in general TSA, can be deemed a classification problem. According to Giachanou and Crestani (2016), Kumar and Jaiswal (2020), and Ligthart et al. (2021) are similar metrics used to traditional classification problems: precision, recall, F-score and accuracy. Consequently, these metrics will be used to evaluate the classification methods used in this research. It depicts the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), which help to identify the evaluation metrics by comparing the predictions of the classification method with the ground truth (Giachanou and Crestani, 2016). Table 10 shows the notation of the evaluation metrics.

Notation	Description
TP(class)	True positives of a sentiment class
TN(class)	True negatives of a sentment class
FP(class)	False positives of a sentiment class
FN(class)	False negatives of a sentiment class

Table 10 Notation evaluation metrics

The precision value, introduced by Sokolova and Lapalme (2009), is the number of correctly classified sentiments and it can therefore be annotated as precision. Precision is calculated by taking the TP (of a class) and dividing this number by the number of TP and FP (of the same class), measuring the percentage of the class that is indeed positive (equation below).

Recall is also introduced by Sokolave and Lapalme (2009), and it accounts for the number of sentiments correctly assigned to be positive. Hence, recall measures the completeness of the results and it is calculated by the number of TP divided by the total number of TP and FN, indicated by equation below.

The most commonly used evaluation metric is accuracy and it measures the correctness of the prediction. It takes the total number of true predictions (TP and TN) divided by the total number of predictions (TP, TN, FP and FN). The accuracy equation is shown in equation below and extracted from Sokolova and Lapalme (2009).

The F1-score, also referred to as (harmonic) F-score or F-measure accuracy, uses the precision and recall scores to demonstrate the relationship between the data's provided labels and those classified by the chosen classifier (Sokolova and Lapalme, 2009). The F1-score is calculated by the multiple of precision and recall scores divided by the addition of precision and recall score, multiplied by two (see equation below). Although it is not detrimental to predict the neutral class, in case of measuring the F1-score in a multi-class problem where sentiment is classified in positive, neutral or negative, it is vital to take into account the neutral F1-score in order to establish the correct ground truth for correct prediction and calculation of the F1-score of the other classes (Giachanou and Crestani, 2016).

$$precision(class) = \frac{TP(class)}{TP(class) + FP(class)}$$

$$recall(class) = \frac{TP(class)}{TP(class) + FN(class)}$$

$$accuracy(class) = \frac{TP(class) + TN(class)}{TP(class) + TN(class) + FP(class) + FN(class)}$$

$$F1\text{-}score = 2 * \frac{precision * recall}{precision + recall}$$

Furthermore, the F1-score has two variations: F1-macro, which considers all classes as equal, and F1-micro, which counts all data points independent of its class (Benevenuto et al., 2010). As a result, the F1-macro score has been key in generating reliable conclusions as it has helped to create a better manifestation of the performance of the classifiers.

In order to extract the precision, recall, accuracy and ultimately the F1-score a multiclass confusion matrix, based on the findings by Sokolave and Lapalme (2009), has been utilized and can be seen In Table 3.

		Predicted classes		
		"positive"	**"neutral"**	**"negative"**
Actual classes	**"positive"**	TP(pos)	A(pos|neu)	A(pos|neg)
	"neutral"	A(neu|pos)	TP(neu)	A(neu|neg)
	"negative"	A(neg|pos)	A(neg|neu)	TP(neg)

Table 11 Multiclass confusion matrix (Sokolova and Lapalme, 2009)

TP(pos) = True positives for class positive

A(pos|neu) = Actual class positive, predicted class neutral

In this research, diverse supervised machine learning algorithms for classification of sentiments on Dutch-based Tweets have been performed. The proposed methodology for this work is depicted in Figure X.

Chapter 5 Results

In this chapter, the results will be offered. The results will be introduced in a similar fashion as the methodology is set up. Thus, this section will start with the results after data preprocessing and lexicon based approaches, followed by results regarding features selection, and it will continue with predictions of the classifiers.

5.1 Data preprocessing

At first, the data, consisting of 100103 have been cleaned and preprocessed. The relevant Tweets have been labelled positive, neutral or negative, whereas the irrelevant Tweets have been excluded. 69013 Tweets remained after cleaning and preprocessing, see Table 12.

City	Count
Amsterdam	17038
Eindhoven	13947
Rotterdam	13224
The Hague	13103
Utrecht	11701

Table 12 Tweets count after preprocessing

Tweets posted to the Twiter platform from the region Amsterdam add up to 17038, whereas the total number of Tweets stemming from Utrecht add up to only 11701. This can be explained by the number of Tweets that were extracted. The run time to extract 25000 Tweets for the city of Utrecht exceeded the maximum run time, which resulted in an error. Hence, a limit was set to the number of items extracted in order to deal with this issue.

5.2 Lexicon based approaches

After applying the lexicon based approach it can be stated that the impact of the differential amount of Tweets is minimized as the class distribution is almost similar. This can be seen in Figure 11 and Figure 12, which depicts the class distribution per city for respectively both the associated TextBlob score and the VADER score.

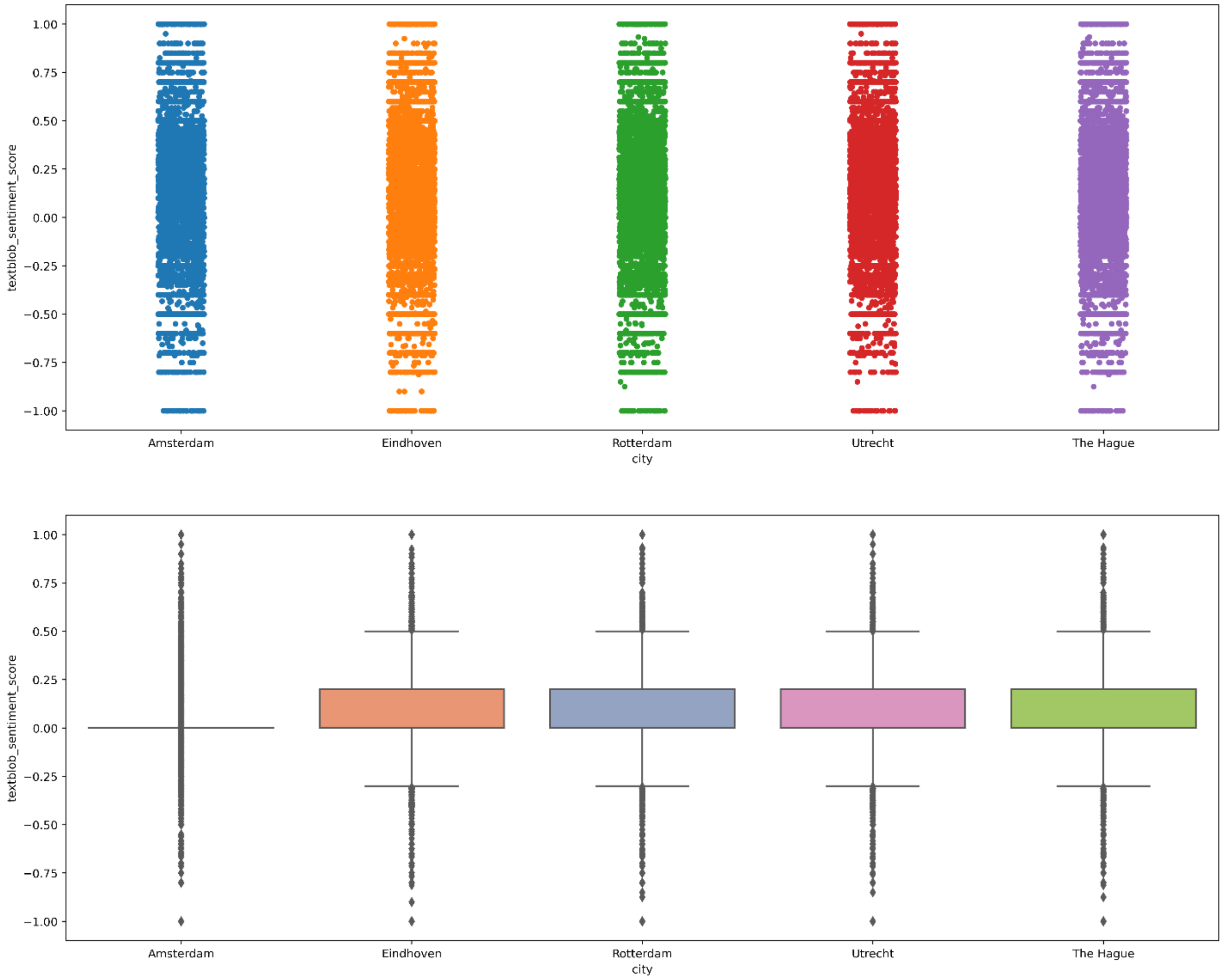


Figure 11 Class distribution per city TextBlob

The overall sentiment scores for both VADER and TextBlob, have been visualized using Seaborn and Matplotlib libraries. The figures show that most Tweets are scattered across the entire range, while the sentiment value, based on the TextBlob approach, has its mean closer to zero compared to the VADER approach. The VADER approach has a wider interquartile range, meaning it has more diverse results. The boxplot of Tweets also express a general positive sentiment with a value inbetween [0-0.25] for TextBlob and a value between [0-0.5] for VADER. This is also supported by the scatterplot, which is more centered in the TextBlob Figure 11 than in the VADER Figure 12. Unfortunately, all five cities indicate a similar sentiment composition, which implies no significant difference in sentiment per city (given that the maximum publication, posting to the Twitter platform, of Tweets within the dataset is a week).

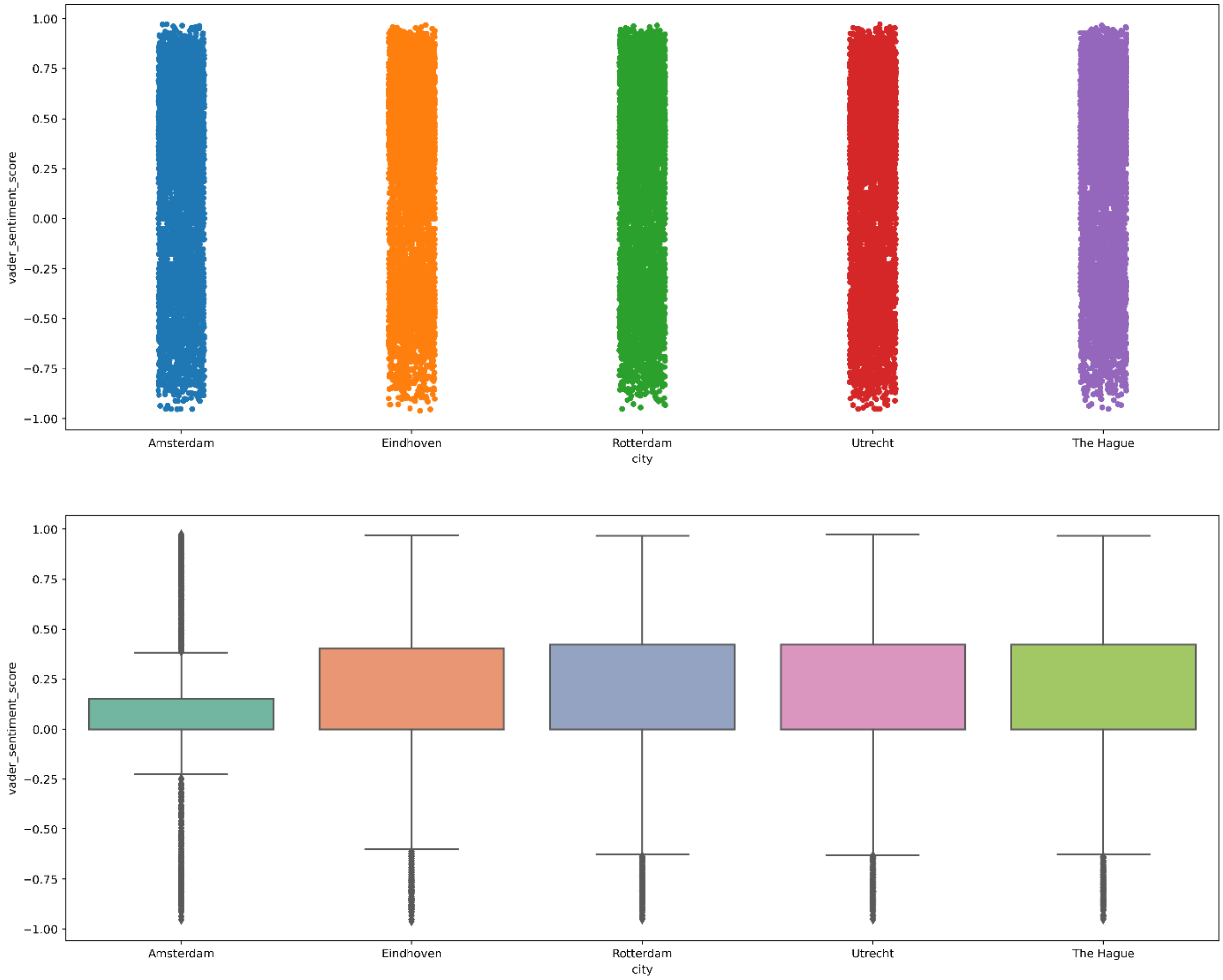


Figure 12 Class distribution per city VADER

In general, the class distribution for both lexicon approaches are similar fairly similar, but it can be stated that VADER is a more suitable approach regarding this dataset as it has been able to detect more distinct polarity and therefore produces a more varied distribution of classes, see Table 13 and Appendix:

	VADER		**TextBlob**	
Neutral	32841	0.476	39227	0.568
Positive	25599	0.371	21610	0.313
Negative	10573	0.153	8176	0.118

Table 13 Distribution of classes lexicon approaches

Even though the is no distinctive difference between the sentiment per city, see Table 14, the VADER approach has been identified as the lexicon based baseline for the classification approaches (in combination with the MNB machine learning classifier).

	Amsterdam		Rotterdam		The Hague		Utrecht		Eindhoven	
Neutral	10137	0.595	5730	0.433	5720	0.437	4814	0.411	6440	0.462
Positive	4556	0.267	5421	0.41	5354	0.409	4859	0.415	5409	0.388
Negative	2345	0.138	2073	0.157	2029	0.155	2028	0.173	2098	0.150
Total	17038	1.000	13224	1.000	13103	1.000	11701	1.000	13947	1.000

Table 14 Distribution per city

5.1 Classification approaches

This study established several analyzes to enquire the machine learning methods in order to classify Dutch based Tweets into neutral, positive or negative classes. At first, the classification experiment was conducted using the BOW feature to evaluate the performance of the chosen supervised machine learning models. Subsequently this experiment was repeated using the TF-IDF feature. The experiments have both been conducted ten times to overcome bias. The (average) results are shown, in Table 15, where the best most relevant scores are listed. A more elaborate overview is provided in the Appendix.

	BOW	**TFIDF**	**BOW**	**TFIDF**
	F-1 macro score		Recall macro score	
MNB	0.79	0.74	0.74	0.72
SVM	0.83	0.84	0.81	0.81
LR	0.83	0.84	0.81	0.81
ET	0.86	0.77	0.83	0.73
GB	0.63	0.68	0.60	0.65
AB	0.42	0.42	0.42	0.46

Table 15 Feature evaluation

The VADER algorithm has been used to provide the data to the classifiers and has formed the baseline with the MNB model. MNB is a good fit for the baseline algorithm as it does not require specific parameter selection and it is traditionally used as a baseline algorithm (Kabakchieva, 2013). As mentioned in the evaluation section of chapter 4, the F1-macro score is a suitable evaluation metric for imbalanced datasets. The scores for the F1-macro score considering the BOW-feature in the Twitter dataset range from 0.42 to 0.86, of which the ET classifier performed best. The scores for the F1-macro considering the TF-IDF-feature in the Twitter dataset range from 0.42 to 0.84, where the SVM classifier as the LR classifier both performed best. The baseline classifier scores a F1-macro score of 0.79 accounting for the BOW-feature and a F1-macro score of 0.74 involving the TF-IDF-feature.

Second the recall scores will be analyzed for the Twitter dataset. It can be stated that there is a lot more variation present compared to the F1-macro score. Accounting for the BOW-feature, the recall scores ranges from 0.07 to 1 for specific classes and from 0.45 to 0.83 for the macro average score. The bottom score is attained by the AB algorithm and the top score is claimed by the ET classifier. The range for the macro recall score accounting for the TF-IDF-feature 0.46-0.81, and for specific classes the range is 0.08 to 1, prompting for a high and a low false negative rate. The base algorithm reached a recall of 0.56 to 0.87 with a macro average of 0.74 for the BOW-feature, and a range of 0.39-0.85 with a macro average recall score of 0.72. Hence, the baseline algorithm attains a higher rate of false negatives using the TF-IDF feature.

When the results of TF-IDF and BOW are compared based on the evaluation metrics depicted in Table X, it can be stated that the overall precision increases when TFIDF-feature is applied, while the impact of either feature cannot be generalized to all classifiers. The linear classifiers attain a fairly constant accuracy and F1-macro average score, whereas the ET classifier encounters problem classifying the negative class when the TF-IDF feature is applied. Furthermore, ET outperforms all models in BOW in terms of accuracy and the SVM classifier gives the most significant performance on TF-IDF in terms of accuracy. The GB classifier attains the highest increase in F-1 average macro score as compared to BOW since TF-IDF provides more weighted features than BoW which provide simply the occurrence of the words. A noteworthy finding, founded on the model results, encompasses the significantly better performance of the ET classifier over the ensemble boosting classifiers.

Chapter 6 Conclusion and future research

The final chapter will commence with a discussion involving the results of the classifications regarding the formulated research questions. Hereinafter, the problems statement will be answered and this section will end with feasible remarks concerning future research.

6.1 Conclusion

The public's opinion has been identified by extracting data on Twitter. In particular, the public's opinion within the boundaries of a certain area, has been extracted using the Python open source platform. The written code used can be found in the Appendix. This code has utilized the Twitter API to gather the data and attributes needed to analyze the influence of the public opinion on the picket process. The data attributes consist of location specificity, date, and text. Also, it is noteworthy to mention that a search term has not been used due to the potential impact of bias. Tweets could have contained emotionally laden text associated with the search term, indirectly influencing the results of this work.

Furthermore, a ground-truth needed to be established in order to support the sentiment expressed by the public's opinion. The ground truth has been created using two different lexicon based approach which are widely adopted in the field of text mining and TSA. The approaches are Textblob and VADER (Hutto and Gilbert, 2014; Loria, 2018). A comparative analysis should than the results produced by VADER were more suitable to the dataset extracted from Twitter. Therefore the VADER lexicon based approach was identified as beneficial to form a baseline in combination with a traditional, also commonly used as a baseline algorithm, machine learning approach. This supervised machine learning approach is the MNB (Kabakchieva, 2013). Unfortunately, the comparative analysis based on the lexicon based approaches also proved the lack of significant diversity in sentiment for the five research regions. These regions were the top five cities of the Netherlands. Due to the relatively small geographical distance it can be explained that the associated sentiment is not significantly different. Nonetheless, it can be stated that VADER is a decent lexicon based approach to assign sentiment in order to measure the public's opinion. Moreover, six different machine learning models, based on the BoW and TF-IDF feature extraction methods, have been applied to address English Tweet classification. MNB, SVM, LR, ET, GB and AB algorithms have been trained to classify Tweets in either negative, neutral or positive classes. The results indicate that most have been classified in the positive and neutral class. As a consequence, we can conclude that the overall sentiment and discussion on Twitter was positive. Additionally, it depends on the classification model which feature is better suitable to classify Tweets, as the results did not significantly imply a best best performing feature. However, it can be stated that the machine learning models can be helpful in predicting future sentiment.

The results depicted in the Appendix, shown that decent accuracy scores may be obtained, for instance ET classifier attains an overall accuracy of 89 percent accounting for the Bow-feature. However, in support of the findings by Gangangwar (2012), who stated that models have a tendency to be biased in relation to the majority class in imbalanced datasets, are the results of the boosting classifiers used in this study as the algorithms attained recall scores for the negative class (which was the least represented class in the test and training set) of 0.26 and 0.39 for the GB algorithm, and a value of 0.07 and 0.08 for the AB algorithm. Moreover the linear classification methods, LR and SVM, had the least variation in their predictive performance and overall scored an accuracy of 0.87. The best performing algorithms for the BoW feature was the ET classifier with a F1 macro average score of 0.86 and the linear classifiers, SVM and LR, with a F1 average macro score of 0.84 for the TF-IDF-feature.

Regrettably, a clear and argumentative answer to the primary research question cannot be provided due to the lack of regional difference within the sentiment. The primary research questions was listed as:

"How can sentiment analysis (SA) be used to measure the influence of the public's opinion on the picket process?"

The sentiment cannot be generalized to the country as whole due to regional focus of the picket process. If the sentiment could be aligned to the overall number of picket notifications, the case company would not know where in the country the number of notifications would increase, and consequently assign more employees. Furthermore, another limitation, is the inability to detect slang. In further research slang detection needs to be tackled accordingly in order to improve the predictive results. Therefore, future research should aim at identifying the public sentiment within a given area.

6.2 Future research

The public sentiment in a given area may be identified based on the detection of certain events. For instance, a minimum number of people following, contributing or attending an (online) event may use a similar hashtag, Retweet or phrase. This phrase does not need to be a trending topic (most used hashtag or word for usually a given country) on Twitter as a relatively simple code can detect the trending topic within a specific area. Furthermore, areas can be quickly scanned using the code in the appendix and subsequently a wordcloud or word n-grams can be created to address the search term for event detection.

To conclude, a search term for particular events could not be applied in this research due to the lack of events. The lack of events is a direct result of the corona measures in the Netherlands during the time of this research, which could have resulted in more variation in terms of sentiment for each city. Hence, it could be a proper future research topic to analyze Twitter data based on events in the biggest cities of the

Netherlands to detect the public's opinion, subsequently this may cause the number of notifications of the picket process to alter.

References

Agarwal, A., Xie, B., Vovsha, I., RamBoW, O., & Passonneau, R. J. (2011, June). Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)* (pp. 30-38).

Agarwal, A., Singh, R., & Toshniwal, D. (2018). Geospatial sentiment analysis using twitter data for UK-EU referendum. *Journal of Information and Optimization Sciences*, *39*(1), 303-317.

Ahmad, M., Aftab, S., Bashir, M. S., Hameed, N., Ali, I., & Nawaz, Z. (2018). SVM optimization for sentiment analysis. *Int. J. Adv. Comput. Sci. Appl*, *9*(4), 393-398.

Ahmed Ibrahim M, Salim N (2013) Opinion analysis for twitter and Arabic tweets: a systematic literature review. J Theor Appl Inf Technol 56(3):338–348

Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). The impact of features extraction on the sentiment analysis. *Procedia Computer Science*, *152*, 341-348.

Aisopos, F., Papadakis, G., & Varvarigou, T. (2011, November). Sentiment analysis of social media content using n-gram graphs. In *Proceedings of the 3rd ACM SIGMM international workshop on Social media* (pp. 9-14).

Al-Moslmi T, Omar N, Abdullah S, Albared M (2017) Approaches to cross-domain sentiment analysis: a systematic literature review. IEEE Access 5:16173–16192

Anwar, A., Ilyas, H., Yaqub, U., & Zaman, S. (2021, June). Analyzing QAnon on Twitter in Context of US Elections 2020: Analysis of User Messages and Profiles Using VADER and BERT Topic modeling. In *DG. O2021: The 22nd Annual International Conference on Digital Government Research* (pp. 82-88).

Apté, C., & Weiss, S. (1997). Data mining with decision trees and decision rules. *Future generation computer systems*, *13*(2-3), 197-210.

Asiaee T, A., Tepper, M., Banerjee, A., & Sapiro, G. (2012, October). If you are happy and you know it... tweet. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1602-1606).

Bakliwal, A., Arora, P., Madhappan, S., Kapre, N., Singh, M., & Varma, V. (2012, July). Mining sentiments from tweets. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis* (pp. 11-18).

Barbosa, L., & Feng, J. (2010, August). Robust sentiment detection on twitter from biased and noisy data. In *Coling 2010: Posters* (pp. 36-44).

Benevenuto, F., Magno, G., Rodrigues, T., & Almeida, V. (2010, 01 01). Detecting spammers on Twitter. Computer Science Department .

Bermingham, A., & Smeaton, A. F. (2010, October). Classifying sentiment in microblogs: is brevity an advantage?. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 1833-1836).

Bonta, V., & Janardhan, N. K. N. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), 1-6.

Brody, S., & Diakopoulos, N. (2011, July). Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! using word lengthening to detect sentiment in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 562-570).

Browne, M. W. (2000). Cross-validation methods. *Journal of mathematical psychology*, *44*(1), 108-132.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*.

Carvalho J, Plastino A (2016) An assessment study of feature and meta-level features in twitter sentiment analysis. In: Proceedings of the 22nd European conference on artifcial intelligence. IOS Press, pp 769–777

Carvalho, J., & Plastino, A. (2021). On the evaluation and combination of state-of-the-art features in Twitter sentiment analysis. *Artificial Intelligence Review*, *54*(3), 1887-1936.

Da Silva, N. F., Hruschka, E. R., & Hruschka Jr, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, *66*, 170-179.

De Oliveira Lima T, Colaco Junior M, Nunes MASN (2018) Mining on line general opinions about sustainability of hotels: a systematic literature mapping. In: Gervasi O, Murgante B, Misra S, Stankova E, Torre CM, Rocha AMAC, Taniar D, Apduhan BO, Tarantino E, Ryu Y (eds) Computational science and ıts applications–ICCSA 2018. Springer, New York, pp 558–574

De Oliveira, T. H. M., & Painho, M. (2021). Open Geospatial Data Contribution Towards Sentiment Analysis Within the Human Dimension of Smart Cities. *In Open Source Geospatial Science for Urban Studies* (pp. 75-95). Springer, Cham.

De Vries, E., Schoonvelde, M., & Schumacher, G. (2018). No longer lost in translation: Evidence that Google Translate works for comparative bag-of-words text applications. *Political Analysis*, *26*(4), 417-430.

El Hannach, H., & Benkhalifa, M. (2018). Wordnet based implicit aspect sentiment analysis for crime identification from twitter. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *9*(12).

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, *56*(4), 82-89.

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering , 42-47.

Genc-Nayebi N, Abran A (2017) A systematic literature review: opinion mining studies from mobile app store user reviews. J Syst Softw 125:207–219. https://doi.org/10.1016/j.jss.2016.11.027

Ghag, K., & Shah, K. (2014). SentiTFIDF–Sentiment classification using relative term frequency inverse document frequency. *International Journal of Advanced Computer Science and Applications*, *5*(2).

Giachanou, A., & Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, *49*(2), 1-41.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12), 2009.

Go, A., Huang, L., & Bhayani, R. (2009). Twitter sentiment analysis. *Entropy*, *17*, 252.

Gull, R., Shoaib, U., Rasheed, S., Abid, W., & Zahoor, B. (2016). Pre processing of twitter's data for opinion mining in political context. *Procedia Computer Science*, *96*, 1560-1570.

Gupta, D., Khanna, A., Bhattacharyya, S., Hassanien, A. E., Anand, S., & Jaiswal, A. (Eds.). (2020). *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 2* (Vol. 1166). Springer Nature.

Gupta, P., Kumar, S., Suman, R. R., & Kumar, V. (2020). Sentiment Analysis of Lockdown in India During COVID-19: A Case Study on Twitter. *IEEE Transactions on Computational Social Systems*.

Hamdan, H., Béchet, F., & Bellot, P. (2013, June). Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (pp. 455-459).

Hazarika, D., Konwar, G., Deb, S., & Bora, D. J. (2020). Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing.

He, W., Zha, S., & Li, L. (2013). Social media competitive analysis and text mining: A case study in the pizza industry. *International journal of information management*, *33*(3), 464-472.

Ho, Y. C., & Pepyne, D. L. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of optimization theory and applications*, 115(3), 549-570.

Hu, X., Tang, J., Gao, H., & Liu, H. (2013, May). Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 607-618).

Hutto, C., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 8, No. 1).

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011, June). Target-dependent twitter sentiment classification. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* (pp. 151-160).

Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *5*, 2870-2879.

Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, *6*, 23253-23260.

Kabakchieva, D. (2013). Predicting student performance by using data mining methods for classification. *Cybernetics and information technologies*, *13*(1), 61-72.

Kasmuri E, Basiron H (2017) Subjectivity analysis in opinion mining—a systematic literature review. Int J Adv Soft Comput Appl 9(3):132–159

Kaur, C., & Sharma, A. (2019). Twitter Sentiment Analysis-A Review Study. *International Journal of Engineering, Applied and Management Sciences Paradigms (IJEAM).*

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, *50*, 723-762.

Kouloumpis, E., Wilson, T., & Moore, J. (2011, July). Twitter sentiment analysis: The good the bad and the omg!. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 5, No. 1).

Kumar, Akshi, and Teeja Mary Sebastian. (2012) "Sentiment analysis on twitter." *International Journal of Computer Science Issues (IJCSI)* 9, no. 4: 372.

Kumar A, Garg G (2019) Systematic literature review on context-based sentiment analysis in social multimedia. Multimed Tools Appl 79:15349–15380

Kumar A, Jaiswal A (2020) Systematic literature review of sentiment analysis on Twitter using soft computing techniques. Concurr Comput Pract Exp 32(1):e5107

Kumar A, Sharma A (2017) Systematic literature review on opinion mining of big data for government intelligence. Webology 14(2):6–47

Li, J., He, Z., Plaza, J., Li, S., Chen, J., Wu, H., ... & Liu, Y. (2017). Social media: New perspectives to improve remote sensing for emergency response. *Proceedings of the IEEE*, *105*(10), 1900-1912.

Li, X., Bai, Y., & Kang, Y. (2021). Exploring the social influence of Kaggle virtual community on the M5 competition. *arXiv preprint arXiv:2103.00501*.

Ligthart, A., Catal, C. & Tekinerdogan, B. Systematic reviews in sentiment analysis: a tertiary study. *Artif Intell Rev* (2021). https://doi.org/10.1007/s10462-021-09973-3

Lin, C., & He, Y. (2009, November). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 375-384).

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, *2*(2010), 627-666.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5*(1), 1-167.

Lim, W. L., Ho, C. C., & Ting, C. Y. (2020). Tweet sentiment analysis using deep learning with nearby locations as features. In *Computational Science and Technology* (pp. 291-299). Springer, Singapore.

Loper, E., & Bird, S. (2002). Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Loria, S., Keen, P., Honnibal, M., Yankovsky, R., Karesh, D., & Dempsey, E. (2014). Textblob: simplified text processing. *Secondary TextBlob: simplified text processing*, *3*.

Loria, S. (2018). textblob Documentation. *Release 0.15*, *2*.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 142-150).

Madhala P, Jussila J, Aramo-Immonen H, Suominen A (2018) Systematic literature review on customer emotions in social media. In: ECSM 2018 5th European conference on social media. Academic Conferences and publishing limited, South Oxfordshire, pp 154–162

Manguri, K. H., Ramadhan, R. N., & Amin, P. R. M. (2020). Twitter sentiment analysis on worldwide COVID-19 outbreaks. *Kurdistan Journal of Applied Research*, 54-65.

McKinney, W. (2010, June). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51-56).

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, *5*(4), 1093-1113.

Meyer, D., & Wien, F. T. (2015). Support vector machines. *The Interface to libsvm in package e1071*, *28*.

Mite-Baidal K, Delgado-Vera C, Solís-Avilés E, Espinoza AH, Ortiz-Zambrano J, Varela-Tapia E (2018) Sentiment analysis in education domain: a systematic literature review. Commun Comput Inf Sci 883:285–297. https://doi.org/10.1007/978-3-030-00940-3_21

Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Montangero, M., & Furini, M. (2015, January). Trank: Ranking twitter users according to specific topics. In 2015 12th *annual IEEE consumer communications and networking conference* (CCNC) (pp. 767-772). IEEE

Musto, C., Semeraro, G., & Polignano, M. (2014, December). A Comparison of Lexicon based Approaches for Sentiment Analysis of Microblog Posts. In *DART@ AI* IA* (pp. 59-68).

Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based deep intelligent contextual embedding for twitter sentiment analysis. *Future Generation Computer Systems*, *113*, 58-69.

Narr, S., Hulfenhaus, M., & Albayrak, S. (2012). Language-independent twitter sentiment analysis. *Knowledge discovery and machine learning (KDML), LWA*, 12-14.

Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70-77).

Neethu, M. S., & Rajasree, R. (2013, July). Sentiment analysis in twitter using machine learning techniques. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (pp. 1-5). IEEE.

Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1, p. 85). USA: Trelgol Publishing.

Ortega, R., Fonseca, A., & Montoyo, A. (2013, June). SSA-UO: unsupervised Twitter sentiment analysis. In *Second joint conference on lexical and computational semantics (* SEM)* (Vol. 2, pp. 501-507).

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., & Smith, N. A. (2013, June). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 380-390).

Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc* (Vol. 10, No. 2010, pp. 1320-1326).

Pawar, S., Jacques, T., Deshpande, K., Pusapati, R., & Meguerdichian, M. J. (2018). Evaluation of cognitive load and emotional states during multidisciplinary critical care simulation sessions. *BMJ Simulation and Technology Enhanced Learning*, *4*(2).

Plisson, J., Lavrac, N., & Mladenic, D. (2004, May). A rule based approach to word lemmatization. In *Proceedings of IS* (Vol. 3, pp. 83-86).

Prabhakar, E., Santhosh, M., Krishnan, A. H., Kumar, T., & Sudhakar, R. (2019). Sentiment analysis of US airline twitter data using new adaboost approach. *International Journal of Engineering Research & Technology (IJERT)*, *7*(1), 1-6.

PraBoWo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*(2), 143-157.

Qazi, A., Raj, R. G., Hardaker, G., & Standing, C. (2017). A systematic literature review on opinion types and sentiment analysis techniques. *Internet Research*.

Rane, A., & Kumar, A. (2018, July). Sentiment classification system of Twitter data for US airline service analysis. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 1, pp. 769-773). IEEE.

Read, J. (2005, June). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop* (pp. 43-48).

Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., & Benevenuto, F. (2016). Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, *5*(1), 1-29.

Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. S. (2019). Tweets classification on the base of sentiments for US airline companies. *Entropy*, *21*(11), 1078.

Rustam, F., Khalid, M., Aslam, W., Rupapara, V., Mehmood, A., & Choi, G. S. (2021). A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. *Plos one*, *16*(2), e0245909.

Saif, H., He, Y., & Alani, H. (2012). Alleviating data sparsity for twitter sentiment analysis. CEUR Workshop Proceedings (CEUR-WS. org).

Saif, H., He, Y., Fernandez, M., & Alani, H. (2014, October). Semantic patterns for sentiment analysis of Twitter. In *International Semantic Web Conference* (pp. 324-340). Springer, Cham.

Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management*, *52*(1), 5-19.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010, April). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web* (pp. 851-860).

Salah Z, Al-Ghuwairi A-RF, Baarah A, Aloqaily A, Qadoumi B, Alhayek M, Alhijawi B (2019) A systematic review on opinion mining and sentiment analysis in social media. Int J Bus Inf Syst 31(4):530– 554. https://doi.org/10.1504/IJBIS.2019.101585

Salton, G. (1986). Another look at automatic text-retrieval systems. *Communications of the ACM*, *29*(7), 648-656.

Samal, B., Behera, A. K., & Panda, M. (2017, May). Performance analysis of supervised machine learning techniques for sentiment analysis. In *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)* (pp. 128-133). IEEE

Sazzed, S., & Jayarathna, S. (2021). Ssentia: a self-supervised sentiment analyzer for classification from unlabeled data. *Machine Learning with Applications*, *4*, 100026.).

Schlosser, S., Toninelli, D., & Cameletti, M. (2021). Comparing Methods to Collect and Geolocate Tweets in Great Britain. *Journal of Open Innovation: Technology, Market, and Complexity*, *7*(1), 44.

Schwartz, H. A., & Ungar, L. H. (2015). Data-driven content analysis of social media: a systematic overview of automated methods. *Annals of the American Academy of Political and Social Science*, 659(1), 78–94. https://doi.org/10.1177/0002716215569197

Severyn, A., & Moschitti, A. (2015, August). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 959-962).

Siahaan, H., Mawengkang, H., Efendi, S., Wanto, A., & Windarto, A. P. (2019, June). Application of Classification Method C4. 5 on Selection of Exemplary Teachers. In *Journal of Physics: Conference Series* (Vol. 1235, No. 1, p. 012005). IOP Publishing.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427-437.

Shayaa S, Jaafar NI, Bahri S, Sulaiman A, Seuk Wai P, Wai Chung Y, Piprani AZ, Al-Garadi MA (2018) Sentiment analysis of big data: Methods, applications, and open challenges. IEEE Access 6:37807–37827. https://doi.org/10.1109/ACCESS.2018.2851311

Statistics Netherlands (Centraal Bureau voor de Statistiek). (2020). Bevolkingsgroei. cbs.nl *https://www.cbs.nl/nl-nl/visualisaties/dashboard-bevolking/bevolkingsgroei/groei*

Stefanidis, A., Vraga, E., Lamprianidis, G., Radzikowski, J., Delamater, P. L., Jacobsen, K. H., … Crooks, A. (2017). Zika in twitter: temporal variations of locations, actors, and concepts. *Jmir Public Health and Surveillance*, 3(2), 22. https://doi.org/10.2196/publichealth.6925

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014, June). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1555-1565).

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, *63*(1), 163-173.

Wakade, S., Shekar, C., Liszka, K. J., & Chan, C. C. (2012). Text mining for sentiment analysis of Twitter data. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Whitelaw, C., Garg, N., & Argamon, S. (2005, October). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 625-631).

Wiebe, J., Bruce, R., & O'Hara, T. P. (1999, June). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics* (pp. 246-253).

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, *39*(2), 165-210.

Wijayanti, R., & Arisal, A. (2021). Automatic Indonesian Sentiment Lexicon Curation with Sentiment Valence Tuning for Social Media Sentiment Analysis. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, *20*(1), 1-16.

Wijayanto, U. W., & Sarno, R. (2018, September). An experimental study of supervised sentiment analysis using Gaussian Naïve Bayes. In *2018 International Seminar on Application for Technology of Information and Communication* (pp. 476-481). IEEE.

Wu, L., Morstatter, F., & Liu, H. (2018). SlangSD: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification. *Language Resources and Evaluation*, *52*(3), 839-852.

Yan, Y., & Yan, J. (2018). *Hands-On Data Science with Anaconda: Utilize the right mix of tools to create high-performance data science applications*. Packt Publishing Ltd.

Zahoor, S., & Rohilla, R. (2020, June). Twitter Sentiment Analysis Using Lexical or Rule Based Approach: A Case Study. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 537-542). IEEE.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). Combining lexicon based and learning-based methods for Twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, *89*.

Zuo, Z. (2018). Sentiment analysis of steam review datasets using naive bayes and decision tree classifier.

Appendices

Appendix 1 CRISP-DM

1.6.1 Business Understanding

The first phase commences with research on the background of the research subject. A systematic literature review towards sentiment techniques is performed in order to retrieve a suitable understanding of Twitter and according domain by researching relevant data mining literature. Based on the literature research a business case is prompted which proposes that the RvR can improve their understanding of the public's opinion on the organization as whole, but in particular can derive value from the public's opinion in the form of better understanding its relation to the planning and matching of picket lawyers and litigants. Events influencing the picket process are identified based on location based Twitter analysis.

1.6.2 Data understanding

The next phase of the CRISP-DM process is the understanding of data. The initial data is collected from Twitter, subsequently the data is explored, described by its properties and then the quality of data is verified. The Twitter data used in this research is extracted from the Twitter Application Programming Interface (API) by using an open source Python package named Tweepy. In this research, sentiments are classified in three different categories: negative, neutral and positive based on the polarity score assigned. The Tools Vader and Text Blob have been used to calculate the polarity of each tweet. These are other powerful tools accessible in Python. A total of approximately 100000 Tweets have been collected from Twitter using Tweepy. This has been done to establish location based public's opinion. In order to apply machine learning algorithms, the dataset needs to be split for training and testing and subsequently the model needs to be trained using different classification methods just as Naïve Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT). The classifiers will be compared on different parameters: accuracy, precision, recall and F1 scores.

1.6.3 Data preparation

In the data preparation phase the relevant data is included whereas the irrelevant data is excluded depending on the data report from the data understanding phase. Hereafter the data is cleaned to remove any useless information that is present within the dataset. The cleaning of data takes the form of removing duplicates, stop word removal and other pre-processing steps discussed in detail in a later chapter. The data is stored in the proper format for further analysis after the cleaning of the data.

1.6.4 Modeling

The first classification model that has been built is based on the data retrieved from Twitter, the data of the model is based on the first pre-processing steps and uses lexicon based approaches to assign sentiment to the Tweets. This algorithms are widely adopted and often used in combination with the NB classification algorithm, which is scalable and useful for small and large datasets. It is a simple algorithm that is very efficient and is known to outperform highly sophisticated and complex algorithms (Chen, Huang, Tian, & Qu, 2009; H. Zhang, 2004). Moreover, two features will be extracted and trained on. Also an initial classification model is built, this model is then assessed and any action required to fine-tune the model is taken. These actions include the evaluation of the model and taking necessary steps to improve the model or try new models when previous models are insufficient.

1.6.5 Evaluation

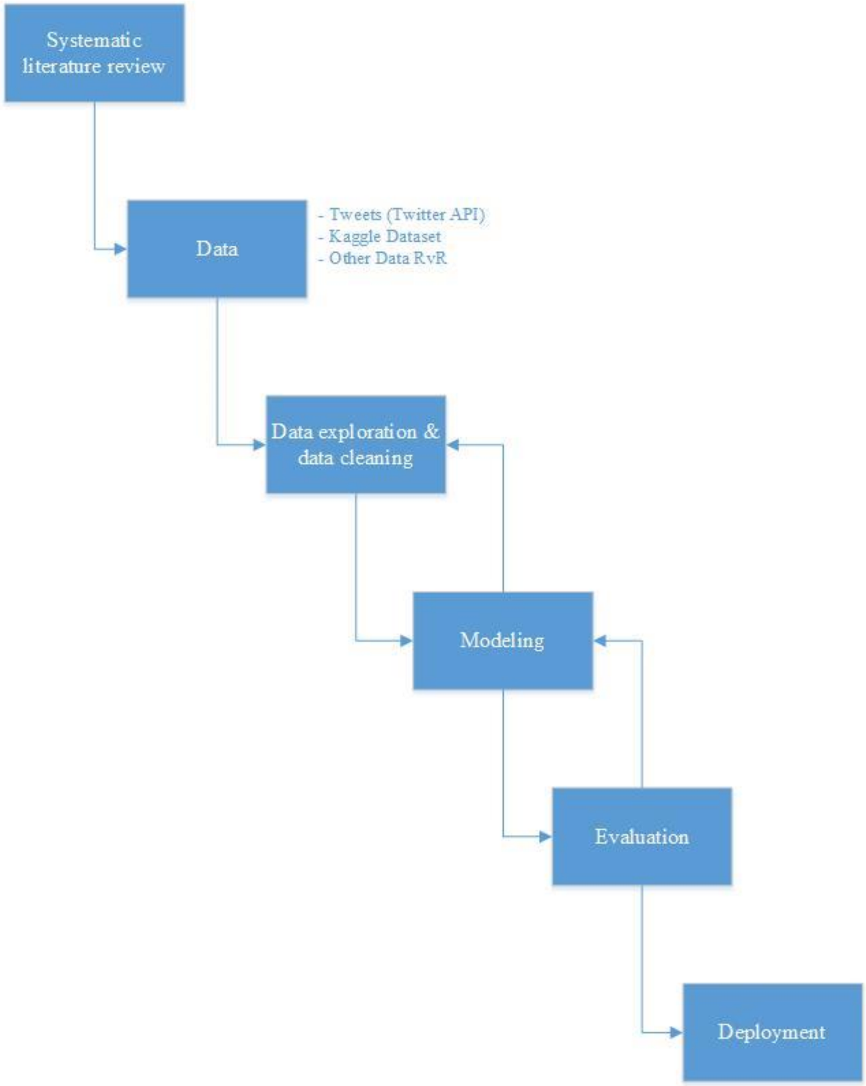
In this phase the outcomes of the model are evaluated based on the accuracy, precision, recall and F1 score of the model. Furthermore, the data mining process is evaluated. The evaluation will lead to changes and recommendations for the iterative steps of the process.

1.6.6 Iteration

After the evaluation of the first classification model, improved models are built, these models try to enhance the classification task by evaluation of the previous phase. Furthermore, new pre-processing steps are performed when considered required. The new models are tested on the extracted and filtered Twitter data.

1.6.7 Deployment

The knowledge extracted from the model built is structured and analysed, elaborated on and eventually presented. An analysis will be made of the picket process events that will be detected and subsequently a trend comparison will take place. This comparison will focus on the sentiment trend, which will be the output of the best performing model. The outcome, conclusions and insights originating from the comparison of the sentiment trend and the picket events are presented in the deployment phase. The thesis research design is indicated in the following Figure.



Appendix 3 Timeline corona measures Netherlands, extracted from www.rijksoverheid.nl

Date	Main corona measures
Jan-20	January 2020: First corona signals
Feb-20	February 2020: First corona contamination in the Netherlands
Mar-20	March 2020: Measures against the spread of the corona virus, intelligent lockdown
Apr-20	April 2020: Extension of measures announcement and test policy
May-20	May 2020: Economic impacts, financial support and relaxation of measures
Jun-20	June 2020: Relaxations of corona measures and testing for all
Jul-20	July 2020: A 'summer and a half' and slowly rising infections
Aug-20	August 2020: 'We are done with the virus, but the virus is not done with us yet'.
Sep-20	September 2020: Tightened measures still needed
Oct-20	October 2020: Second wave and partial lockdown
Nov-20	November 2020: Strengthening and extension of partial lockdown
Dec-20	December 2020: Lockdown during holidays and mutation of virus appears in UK

Appendix 3 Libraries and packages

Libraries/packages
accuracy_score
AdaBoostClassifier
autocorrect
classification_report
collections
confusion_matrix
counter
CountVectorizer
datetime
defaultdict
ExtraTreesClassifier
GradientBoostingClassifier
itertools
LogisticRegression
matplotlib
MultinomialNB
nltk.corpus
nltk.stem
nltk.stem.porter
nltk.tokenize
numpy
packages
pandas
RandomForestClassifier
re
seaborn
SentimentIntensityAnalyzer
sklearn.ensemble
sklearn.feature_extraction.text
sklearn.metrics
sklearn.model_selection
sklearn.naive_bayes
sklearn.svm
spacy
spell
stopwords
string
SVC
TextBlob
TfidfVectorizer
ToktokTokenizer
train_test_split
vaderSentiment
word_tokenize
WordCloud
wordnet

Appendix 3 Twitter data extraction

```
#Twitter Developer Account Access Keys
```

```
consumer_key= 'SlDU79rIgrpVPQCjj2ut5f8Xn'
consumer_secret= 'UED0E74Crkr4FjlgMr87cNv9tcD0GO81V69pxVWtjaQx6N6KOr'
access_token='1379417669020098565-OiKOYNhgC4KuOjOqGiVw9M7RsWfWoT'
access_token_secret='95jEGqNpQ5utFMGGy5XCzLLxafHBCIoFQEV7Sz8oWylo4'

# Connection with Twiter Developer Account to extract the Tweets
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tw.API(auth, wait_on_rate_limit=True)

#Search English Tweets, location, user, text and data.

search_words = ""
new_search = search_words + " -filter:retweets"
date_since = "2020-01-01"
geoc="52.070499,4.300700,25mi"
tweets = tw.Cursor(api.search,
                   geocode=geoc,
                   lang="en",
                   since=date_since).items(x)

users_locs = [[tweet.user.screen_name, tweet.user.location,tweet.text,tweet.created_at.date()] for tweet in twee
ts]

data = pd.DataFrame(data=users_locs,
              columns=['user', "location","text","date"])


stg=geoc.split(",")

data["long"]=stg[0]
data["lat"]=stg[1]
```

Appendix 4 Cleaning and preprocessing

```
# check column names
print(data.columns)
### removing duplicate Tweets
data.duplicated()
data.drop_duplicates(keep='first', inplace=True)
### removing ReTweets
data = data[data.text.str.contains("RT") == False]
### conversion of ’ to '
# define pattern
patterncomma = r'’'
# replacing regex functions based on column Tweet
data.text.replace({patterncomma: "'"},inplace=True, regex=True)
### hasthag removal
# define pattern
patternhash = r'#'
# replacing regex functions based on column Tweet
data.text.replace({patternhash: ""},inplace=True, regex=True)
### removal of usernames
# define pattern
patternuser = r'(@\w+)'
# replacing regex functions based on column Tweet
data.text.replace({patternuser: ""},inplace=True, regex=True)
#### removing capital letters
data['text'] = data['text'].str.lower()
### HTML REMOVING
# define pattern \S anything but a space, + match one or more
patternhtml = r'http\S+'
# replacing regex functions based on column Tweet
data.text.replace({patternhtml: ""},inplace=True, regex=True)
##### Apostrophe removing (contraction words)
#calling dictionary of apostrophe contracted words
dictionary = {}
#checking index and resetting (working)
data.reset_index(inplace=True, drop=True)
```

```
# replacing contraction words
counter = list(range(0,2950)) ## place last index value here of datasets last position
counter2 = range(0,10000) ## value that makes it possible to iterate over all dictionary values
counter3 = [counter2] ## make list out of values
for j in counter:
  words = data['text'][j].split()
  print(words)
  for x in counter3:
    reformed = [dictionary[x] if x in dictionary else x for x in words]
    reformed = " ".join(reformed)
    data.at[j, 'text'] = reformed
    counter2 = 0
### Removing punctuation
counter = list(range(0,2950)) ## place last index value here of datasets last position
for k in counter:
  for c in string.punctuation:
    punct = data['text'][k].replace(c," ")
    data.at[k, 'text'] = punct
### cleaning van slang
#dictionary = slangdict
# replacing slang words (working)
counter = list(range(0,2950)) ## place last index value here of datasets last position
counter2 = range(0,10000) ## value that makes it possible to iterate over all dictionary values
counter3 = [counter2] ## make list out of values
for i in counter:
  words = data['text'][i].split()
  for z in counter3:
    reformed = [dictionary[z] if z in dictionary else z for z in words]
    reformed = " ".join(reformed)
    data.at[i, 'text'] = reformed
### standardizing words
counter = list(range(0,2950))
for g in counter:
  words = data['text'][g]
  words = ".join(".join(s)[:2] for _, s in itertools.groupby(words))
  words = ".join(words)
  data.at[g, 'text'] = words
### spelling correction
counter = list(range(0, 2950))

for k in counter:
  words = data['text'][k].split()
  for l in range(len(words)):
    correctword = spell(words[l])
    if correctword != words[l]:
      words[l] = correctword
      words = " ".join(words)
      data.at[k, 'text'] = words
### filter out stopwords
counter = list(range(0,2950)) ## place last index value here of datasets last position
counter2 = range(0,10000) ## value that makes it possible to iterate over all dictionary values
counter3 = [counter2] ## make list out of values
stopwords_list = set(stopwords.words('english'))
print(stopwords_list)

for m in counter:
  words = data['text'][m].split()
  for c in counter3:
    words = [c for c in words if c not in stopwords_list]
    words = " ".join(words)
    data.at[m, 'text'] = words
#### removing capital letters
data['text'] = data['text'].str.lower()
### Lemmitization
lemmatizer = WordNetLemmatizer()
counter = list(range(0,2950)) ## place last index value here
for g in counter:
  words = data['text'][g].split()
  for f in range(len(words)):
    lemmaword = lemmatizer.lemmatize(words[f])
    if lemmaword != words[f]:
      words[f] = lemmaword
```

```
        words = " ".join(words)
        data.at[g, 'text'] = words
```

Appendix 5.1 TextBlob sentiment analysis

```
#Sentiment analysis using TextBlob
data['TextBlob Score']=""
data['TextBlob Sentiment']=""
#df2 = pd.DataFrame(columns=['text', 'sentiment', 'score'])
data['cleanText']=data['cleanText'].astype(str)
for i in range(len(data)):
    sentiment = TextBlob(data['cleanText'][i])
    a=sentiment.sentiment.polarity
    #df2.loc[i] = [data['cleanText'][i]]+[str(0)]+ [a]
    data["TextBlob Score"][i]=a

for i in range(len(data)):
    if(data['TextBlob Score'][i]>0.05):
        data['TextBlob Sentiment'][i]="Positive"
    elif(data['TextBlob Score'][i]<0.05):
        data['TextBlob Sentiment'][i]="Negative"
    else:
        data['TextBlob Sentiment'][i]="Neutral"
```

Appendix 5.2 VADER sentiment analysis

```
data["Vader Score"]=0
data["Vader Sentiment"]=0
sid_obj = SentimentIntensityAnalyzer()
# function to print sentiments
# of the sentence.
for i in range(len(data)):

    # Create a SentimentIntensityAnalyzer object.
    # polarity_scores method of SentimentIntensityAnalyzer
    # object gives a sentiment dictionary.
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(data["cleanText"][i])
    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        data["Vader Score"][i]= str(sentiment_dict['compound'])
        data["Vader Sentiment"][i]= "Positive"



    elif sentiment_dict['compound'] <= - 0.05 :
        data["Vader Score"][i]= str(sentiment_dict['compound'])
        data["Vader Sentiment"][i]= "Negative"

    else :
        data["Vader Score"][i]= str(sentiment_dict['compound'])
        data["Vader Sentiment"][i]= "Neutral"
```

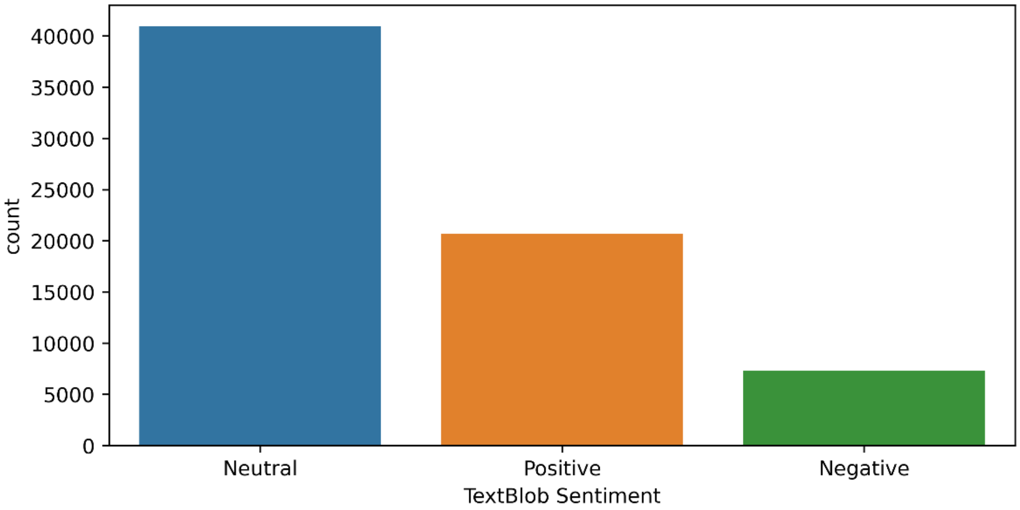
Appendix 6 Classification

```
# Splitting dataset into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(data["text"].astype(str), data["Vader Sentiment"], test_size=0.25, shuffle=True)

# learn training data vocabulary, then use it to create a document-term matrix
vect = CountVectorizer(max_features=2000)
# 3. fit
# 4. transform training data
X_train_dtf = vect.fit_transform(X_train)
X_test_dtf = vect.transform(X_test)

vectorizer = TfidfVectorizer(max_features=2000)
X_train_tf = vectorizer.fit_transform(X_train)
X_test_tf = vectorizer.transform(X_test)

print("MNB")
mnb = MultinomialNB()
mnb.fit(X_train_dtf,y_train)
```

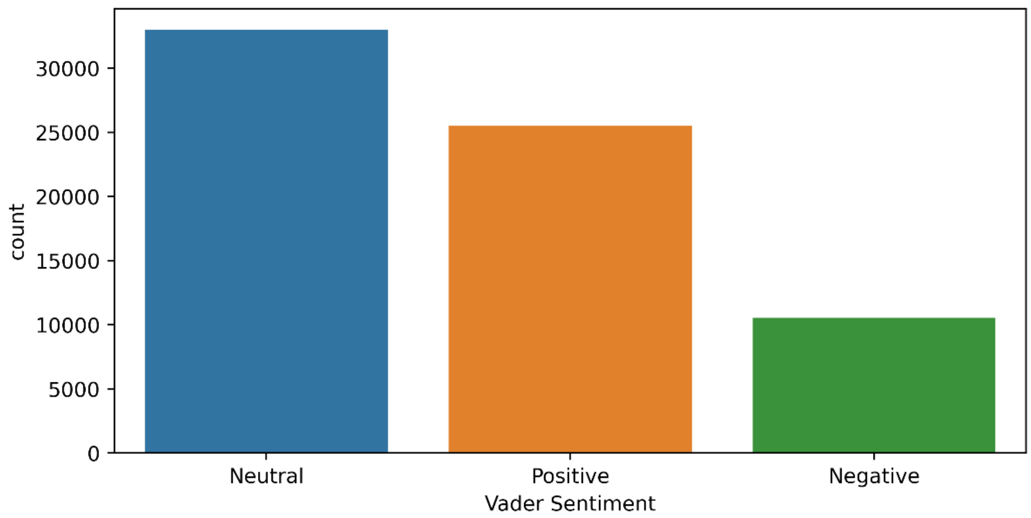
```
y_pred=mnb.predict(X_test_dtf)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

print("SVC")
svm = SVC(kernel='linear', C=1.0, random_state=500)
svm.fit(X_train_dtf,y_train)
y_pred=svm.predict(X_test_dtf)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

print("LR")
logreg = LogisticRegression(random_state=1000,multi_class='multinomial',C=3.0)
logreg.fit(X_train_dtf,y_train)
y_pred=logreg.predict(X_test_dtf)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

etc = ExtraTreesClassifier(n_estimators=300, random_state=5, max_depth=300)
print("ExtraTreesClassifier")
etc.fit(X_train_dtf.toarray(),y_train)
y_pred=etc.predict(X_test_dtf.toarray())
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

a = GradientBoostingClassifier(max_depth=10, learning_rate=0.2, n_estimators=10, random_state=52)
print("GBM")
a.fit(X_train_dtf.toarray(),y_train)
y_pred=a.predict(X_test_dtf.toarray())
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print("ADA")
xgb = AdaBoostClassifier(n_estimators=10, random_state=5)
xgb.fit(X_train_dtf.toarray(),y_train)
y_pred=xgb.predict(X_test_dtf.toarray())
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

print("MNB")
mnb = MultinomialNB()
mnb.fit(X_train_tf,y_train)
y_pred=mnb.predict(X_test_tf)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

print("SVC")
svm = SVC(kernel='linear', C=1.0, random_state=500)
svm.fit(X_train_tf,y_train)
y_pred=svm.predict(X_test_tf)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

print("LR")
logreg = LogisticRegression(random_state=1000,multi_class='multinomial',C=3.0)
logreg.fit(X_train_tf,y_train)
y_pred=logreg.predict(X_test_tf)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

etc = ExtraTreesClassifier(n_estimators=300, random_state=5, max_depth=100)
print("ExtraTreesClassifier")
etc.fit(X_train_tf.toarray(),y_train)
```

```
y_pred=etc.predict(X_test_tf.toarray())
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

a = GradientBoostingClassifier(max_depth=10, learning_rate=0.2, n_estimators=10, random_state=52)
print("GBM")
a.fit(X_train_tf.toarray(),y_train)
y_pred=a.predict(X_test_tf.toarray())
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))

xgb = AdaBoostClassifier(n_estimators=10, random_state=5)
xgb.fit(X_train_tf.toarray(),y_train)
y_pred=xgb.predict(X_test_tf.toarray())
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
```

Appendix 7.1 TextBlob sentiment distribution



Appendix 7.2 Vader sentiment distribution


30000
25000
20000
15000
10000
5000
0
count
Neutral
Positive
Negative
Vader Sentiment

Appendix 8 Performance of the classification algorithms

Feature	**BOW**					**TF-IDF**				
Model	**Accuracy**	**Class**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Support**
MNB	0.79	Negative	0.76	0.56	0.65	0.81	0.91	0.39	0.55	2653
		Neutral	0.85	0.80	0.83		0.81	0.91	0.86	8257
		Positive	0.73	0.87	0.79		0.79	0.85	0.82	6344
		Macro avg	0.78	0.74	0.79		0.84	0.72	0.74	17254
		Weighted avg	0.79	0.79	0.79		0.82	0.81	0.80	17254
SVM	0.87	Negative	0.87	0.59	0.70	0.88	0.89	0.60	0.71	2653
		Neutral	0.84	0.99	0.90		0.83	0.99	0.90	8257
		Positive	0.94	0.85	0.89		0.94	0.85	0.89	6344
		Macro avg	0.88	0.81	0.83		0.89	**0.81**	**0.84**	17254
		Weighted avg	0.88	0.87	0.87		0.88	0.88	0.87	17254
LR	0.87	Negative	0.84	0.61	0.70	0.87	0.87	0.62	0.72	2653
		Neutral	0.85	0.96	0.90		0.84	0.97	0.90	8257
		Positive	0.92	0.86	0.89		0.93	0.85	0.89	6344
		Macro avg	0.87	0.81	0.83		0.88	**0.81**	**0.84**	17254
		Weighted avg	0.87	0.87	0.87		0.88	0.87	0.87	17254
ET	0.89	Negative	0.93	0.64	0.76	0.83	0.97	0.42	0.59	2653
		Neutral	0.85	0.98	0.91		0.76	0.98	0.86	8257
		Positive	0.94	0.88	0.91		0.92	0.79	0.85	6344
		Macro avg	0.91	**0.83**	**0.86**		0.88	0.73	0.77	17254
		Weighted avg	0.90	0.89	0.89		0.85	0.83	0.82	17254
GB	0.72	Negative	0.90	0.26	0.40	0.73	0.98	0.39	0.55	2653
		Neutral	0.65	0.99	0.78		0.64	1.00	0.78	8257
		Positive	0.91	0.56	0.70		0.95	0.56	0.71	6344
		Macro avg	0.82	0.60	0.63		0.85	0.65	0.68	17254
		Weighted avg	0.78	0.72	0.69		0.81	0.73	0.71	17254
AB	0.59	Negative	0.92	0.07	0.13	0.60	0.93	0.08	0.14	2653
		Neutral	0.54	1.00	0.70		0.54	1.00	0.71	8257
		Positive	0.93	0.28	0.43		0.94	0.28	0.43	6344
		Macro avg	0.80	0.45	0.42		0.80	0.46	0.42	17254
		Weighted avg	0.74	0.59	0.51		0.75	0.60	0.51	17254