iscte

UNIVERSITY
INSTITUTE
OF LISBON

Sentiment Analysis in the Stock Market Based on Twitter Data

José Maria Guerreiro Ferreira Félix do Sacramento

Master in Computer Science and Business Management

Supervisor:
PhD Adriano Martins Lopes, Invited Assistant Professor,
ISCTE-IUL

November, 2021

[This page has been intentionally left blank]

iscte
TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Sentiment Analysis in the Stock Market Based on Twitter Data

José Maria Guerreiro Ferreira Félix do Sacramento

Master in Computer Science and Business Management

Supervisor:
PhD Adriano Martins Lopes, Invited Assistant Professor,
ISCTE-IUL

November, 2021

[This page has been intentionally left blank]

Sentiment Analysis in the Stock Market Based on Twitter Data

[This page has been intentionally left blank]

To my parents and my sister.

[This page has been intentionally left blank]

Abstract

In this dissertation, we discuss how Twitter can help detecting public sentiment towards companies listed in the stock market, in particular listed in the S&P 500 index (S&P 500). The collection of data is done through a web scrapper that collects tweets from Twitter, using advanced search features based on queries related to the companies under scrutiny. The content of tweets are classified as positive, neutral or negative sentiments and the outcome is then compared against stock market prices. To do so, it is proposed and implemented a framework with different Sentiment Analysis (SA) models and Machine Learning (ML) techniques. Also, to establish which models are more appropriate in detecting and classifying sentiments, a series of visual representations were created to evaluate and compare results.

As a conclusion, the results obtained show that an increase in the volume of tweets leads to oscillations in both stock price and trading volume. Furthermore, the data analysis performed in relation to some companies under scope shows that the use of moving averages of sentiment scores makes the analysis clearer and more insightful, which is particular useful when measuring the strength or weakness of the price of a stock. In the end, it can be perceived as a momentum indicator for the stock market.

Keywords: Sentiment Analysis, Social Networks, Twitter, Stock Market, Polarity Detection.

[This page has been intentionally left blank]

Resumo

Nesta dissertação, é analisada a forma como a plataforma Twitter pode ajudar a detectar sentimento público relativamente a empresas cotadas em bolsa, com foco em empresas que fazem parte do indíce americano S&P 500. A obtenção de dados é feita através de um *web scrapper*, que recolhe *tweets* através de funções de pesquisa avançada, baseada em *queries* associadas às empresas em análise. O conteúdo dos tweets são classificados como positivos, neutros ou negativos, sendo os resultados comparados de seguida com os preços das ações. Nesse sentido, é proposta um arquitectura de trabalho, com a respetiva implementação, que inclui vários modelos de análise de sentimento e técnicas de *Machine Learning*. Por outro lado, de modo a estabelecer quais são os modelos mais adequados para detectar e classificar sentimentos, são criados várias representações visuais para avaliar e comparar resultados.

Como conclusão, os resultados obtidos mostram que um aumento do número de *tweets* conduz a oscilações, quer no preço, quer na quantidade de ações transacionadas. Além disso, a análise de dados levada a cabo relativamente a algumas empresas em estudo, mostra que a utilização de médias móveis de resultados de sentimento torna a leitura da análise mais clara e evidente, o que é bastante útil para medir a força ou fraqueza do preço de determinada ação. Acima de tudo, tal poderá ser percecionado como um indicador de momento para o mercado de capitais.

Palavras-chave: Análise de Sentimentos, Redes Sociais , *Twitter*, Mercados Financeiros, Detecção de Polaridade.

[This page has been intentionally left blank]

Contents

List of Figures

[This page has been intentionally left blank]

List of Tables

[This page has been intentionally left blank]

Acronyms

BoW	Bag-of-Words.
DJIA	DJIA index.
FAANG	Facebook, Amazon, Apple, Netflix, and Alphabet (parent company of Google).
Flair	Flair.
ML	Machine Learning.
NASDAQ	NASDAQ index.
NB	Naïve Bayes.
NLP	Natural Language Processing.
NLTK	NLTK library.
OHLC	Open–High–Low–Close.
RF	Random Forest.
S&P 500	S&P 500 index.
SA	Sentiment Analysis.
scikit-learn	Machine Learning in Python.
Spark ML	Spark ML.
SVM	Support Vector Machine.
Textblob	Textblob.
US stock market	United States Stock Market.
VADER	Valence Aware Dictionary for sEntiment Reasoner.

[This page has been intentionally left blank]

Chapter 1.

Introduction

Contents

This chapter starts by introducing the motivation for this research and describing the problem, as well as setting the scope and methodological approach to be followed. Then, the research objectives are presented. Finally, it highlights the main contributions of this dissertation and presents the structure of the document.

[This page has been intentionally left blank]

Chapter 1
Introduction

1.1 Overview

The rapid growth of the Internet in recent years reshaped the way we connect and interact with the world. The emergence of social networks, as a result of Web 2.0 launch, allowed anyone to create and share content in a broadcast way. The democratisation of the internet unleashed the amount of data generated online and has revolutionized the world in terms of data analysis and its applications. As stated in [4], Web 2.0 includes a wide range of applications, like wikis, blogs, social networking, and content hosting services. Out of these applications, social networking, more specifically the social media platform Twitter deserves particular attention, at least in this research.

Twitter has emerged as a major social media platform with more than 100 million users, generating over 500 million tweets per day. Twitter's primary purpose is to connect people and to allow them to share their opinions and to discuss a variety of topics. The ceaseless growth of people sharing their opinions made Twitter of one the most popular and browsed website. This brought the attention of many researchers from various fields, including politics, healthcare, and financial markets, to name a few.

Indeed, Twitter stands out with regards to the spread of information in comparison to others like Facebook, Instragram or LinkedIn. As mentioned in [21], instantaneous updates, and small message size makes Twitter the ideal platform to source and disseminate information about any given topic or opining, news or announcements, from companies, politicians, and your next-door investor neighbour.

Let us take the example of the U.S. 2016 elections. Authors in [11], using people's opinion captured via Twitter, tried to predict who would win. To this aim, the authors gathered tweets related to either Donald Trump or Hillary Clinton, since during their campaign both candidates took to Twitter to disseminate information related to their policies or to harm their opponent´s campaign.The authors stated that many people took to Twitter to express their views on the presidential candidates and defend their choice or debase their opponent. To that extent, primary research goal relied on determine whether Twitter can be an effective polling method when compared against traditional polling methods such the IBD/TIPP tracking poll. More recently, in [27], the authors stated that Twitter is the most effective polling method when compared to the results of three different polls sources. Cases as those mentioned above, made researchers focus their attention on Twitter, aiming to use it as the basis of a predictive tool in different areas, through the analysis of people's opinions [27].

In the case of financial markets, some researchers believe that by processing user´s opinion from Twitter it makes feasible for someone to gather relevant information about the stock market and so to use it to predict stock price movements. This does not come as surprise.

Indeed, from the very early days of stock markets, investors have always tried to have an edge so to profit from it. Not illegally e.g. via inside trading but using information cleverly, and scarce to some extent. There are abundant examples: people which job was specifically reading newspapers to grasp valuable information and sentiment out there, or in more distant times, the case of the financier Rothschild, who, thanks to his network of carrier pigeons knew that England had defeated France at Waterloo before anyone else in London. So he profit from it by betting in the opposite direction from other traders.

From a different perspective, in [27] the author further adds that due to the risk and complexity that involves investing in the financial market several techniques are being developed to minimize investor's losses and increase their profits. The author's concept behind these techniques is to use ML algorithms to correlate people's opinions about the financial market and stock price movements, aiming in the end to predict possible future changes.

1.2 Motivation and Objectives

The object of study in this dissertation is the impact of Twitter in financial markets. More specifically, the goal is to understand how tweets nexus to a particular company listed in a stock exchange may affect its stock price movement.

If we collect tweets and financial information related to companies of interest, and then to process such information with the help of a framework specifically designed to infer a sentiment score sentiment of a tweet, followed by a comparison against its stock price movement, we may be able to build a useful momentum indicator for the stock market. Notice that, as any other momentum indicator for the stock market, it will be an indicator that is going to be used alongside others. And from the very beginning, we should emphasize that counting the number of tweets is not enough despite being useful. We have to go further and analyse its content.

Other useful hints to be considered would be:

- Having a list of pre-defined companies and extracting tweets nexus to these companies, the better approach is one company-one analysis, meaning that each search query to be executed refers only to one company.
- To design and implement various models and/or techniques, so to better evaluate the relationship between the sentiment expressed (score) vs. the stock price movement.
- To detect and list which companies are more exposed to Twitter.

To address these issues, the proposed frameworks will collect and classify tweets according to their polarity. The sentiment expressed in a tweet can be labelled according to its polarity, that is positive, neutral, or negative. Once tweets are classified, a daily score is calculated based on the arithmetic average between the number of tweets and their respective scores. Then, after computing these values, the goal is to evaluate the relationship between the daily scores and their corresponding stock's prices. Furthermore, it is worth considering a correlation analysis e.g. to be shown via heatmaps highlighting correlation between different data features.

Having said that, the main research questions we are particularly interest on are the following:

- How tweets related to a particular company listed in the S&P 500 may affect its stock price movement?
- Can the SA scores be used as a momentum indicator to measure strength or weakness of a stock's price, even alongside other indicators?

As mentioned above, we must evaluate if there is a relationship, either positive, neutral or negative, between the number of tweets and its stock price movement. Otherwise we will not be able to answer the research questions.

Once a relationship is detected, further investigation is required to understand if there is a latency between the sentiment score and its corresponding impact, if any. A crucial aspect of this research is to understand if the sentiment score can help to identify the direction of movement or trends towards a given stock price.

1.3 Methodology

As inferred from the sections above, the methodological approach for this study consists first in collecting data from Twitter, the tweets, and classifying them according to sentiment polarity expressed. On the other hand, as the size and spectrum of the financial stock markets available worldwide is too large, this study will use a list of pre-defined companies listed in the S&P 500 index [1]. The reason behind this choice relies on the fact that this index is one of most popular and relevant indices worldwide, being regarded as the best single gauge of large-cap U.S. equities, and also because their companies experience high levels of popularity towards multiple social media platforms and their financial communities.

The programming language used in this research is Python. Moreover, the scripts developed throughout the different modules were implemented using Jupyter Notebook and the Spyder IDE. Additionally, the visual representations of results were created using Microsoft Power BI (free version), a business analytics tool useful for data visualization.

To collect tweets, one can use the official Twitter API. But this presents many limitations and restrictions and it is not suitable for this study, given the huge amount of information we are looking for. That being said, one of the biggest challenges of this study concerns the sourcing of Twitter data given the limited options. To surpass this limitation, we choose Twin, a comprehensive and reliable web scrapper developed in Python.

1.4 Document Structure

This document is composed of five chapters, with each one describing the necessary steps to achieve the goals set in Section 1.2. Apart from this Chapter 1 – *Introduction* – the remain

[1] As of December 2020, an estimated USD 13.5 trillion is indexed or bench-marked to the index, with index assets comprising approximately USD 5.4 trillion of this total.

chapters are as follows:

- Chapter 2 – *Related Work* – outlines the state of the art in relation to the subject of interest, SA. The key aspects are its related concepts, the use of the social network Twitter in such context, as well as different algorithmic approaches to work with, namely ML techniques.
- Chapter 3 – *System Architecture and Framework* – proposes the underlying system architecture of the solution implemented, and the framework to build upon, that supports the work of SA in relation to stock market information. It introduces four main blocks of data processing and visualization, as part of the overall architecture, and specifies correspondent SA models and ML techniques to be used.
- Chapter 4 – *Evaluation* – outlines the evaluation and experiments undertaken. It presents the data collection process and the subsequent data processing steps, with focus on 10 selected companies. With SA results computed, particular attention is devote to two of those companies, in order to highlight the research findings.
- Chapter 5 – *Conclusions* – summarizes the main contributions of this dissertation and presents possible research paths for further development in the future.

Chapter 2.

Related Work

Contents

This chapter outlines the state of the art in relation to the subject of interest, SA. We start by introducing the problem and its underlying concepts. Then we focus on the use of the social network Twitter as a useful tool in the SA field. Lastly, we provide a glance of different algorithmic approaches to work with, namely ML techniques.

[This page has been intentionally left blank]

Chapter 2
Related Work

2.1 Sentiment Analysis

SA, also called opinion mining, is the field of study that analyses people's opinions, sentiments, views, attitudes and emotions toward entities such as products, services, organizations, individuals, events, topics and their attributes [19].

The incessantly increasing amount of information accessible online in terms of volume, velocity, and opinion-rich information made the research domain of SA a trending topic among academics and professionals due its practicals applications, which facilitates decision support and deliver targeted information to domain analysts [17]. Text mining models define the process to transform and substitute this unstructured data into structured data for knowledge discovery. Usage of classification algorithms to intelligently mine text has been studied extensively across literature [14]. SA, established as a typical text classification task, is defined as the computational study of people's opinions, attitudes and emotions towards an entity [18]. It offers a technology-based solution to understand people's reactions, views and opinion polarities e.g. positive, negative or neutral in textual content available over social media sources [17].

Research studies and practical applications have scaled in the past decade with the transformation and expansion of the Web, moving from a passive content provider to an active socially-aware distributor of collective knowledge. This new collaborative Web, the so-called Web 2.0, was extended by Web-based technologies like comments, blogs, wikis, and social media portals like Twitter or Facebook. It allows to build social networks based on professional relationships, interests, etc. and encourages a wider range of expressive capabilities, so facilitating more collaborative ways of working, enabling community creation, dialogue and knowledge sharing, as well as providing a setting for learners to attract authentic audiences via different tools and technologies. Furthermore, the convergence of the four technologies – Social media, Mobile, Analytics and Cloud services – has offered the new SMAC technology paradigm, which has notably transformed the operative environment and user engagement on the Web [17].

On the other hand, despite SA research has become very popular in recent years, most companies and researchers still approach it simply as a polarity detection problem. As a matter of fact, SA is a *suitcase problem* that requires tackling many Natural Language Processing (NLP) subtasks, including microtext analysis, sarcasm detection, anaphora resolution, subjectivity detection, and aspect extraction [14].

In general, there are three categories of strategies to deal with affective computing and sentiment analysis: knowledge-based techniques, statistical methods, and hybrid approaches [17]. Also, it is categorised in three levels of granularity: document level, sentence level, entity and aspect level.

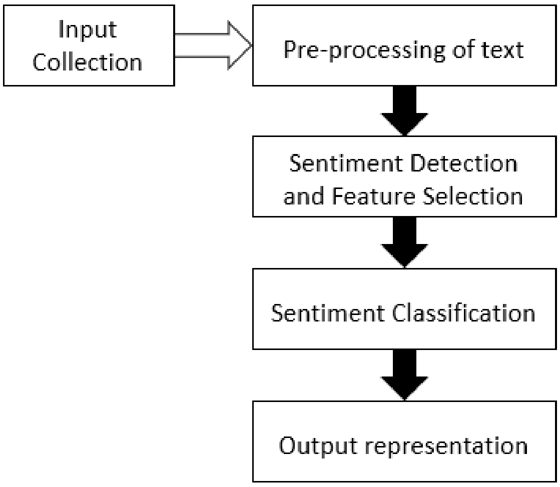


Figure 2.1: Generic SA Processing Workflow

According to the work in [19], the three main categories in SA are as follows:

Document level: At this level, the task is to classify whether a whole opinion document expresses a positive or negative sentiment. This level of analysis assumes that each document expresses opinions on a single entity (i.e. a single product). For example, given a product review, the system determines whether a review expresses an overall positive or negative opinion about the product. Thus, it is not applicable to documents that evaluate or compare multiple entities.

Sentence level: At this level, the task goes to the sentences and determines whether each sentence expresses a positive, negative, or neutral opinion. This level of analysis is closely related to subjective classification, which, as mentioned in [7] distinguish sentences (called *objective sentences*) that express factual information from sentences (called *subjective sentences*) that express subjective views and opinions. However, we should note that subjectivity is not equivalent to sentiment as many objective sentences can imply opinion. For example, that is the case of *"we bought the car last month and honestly, couldn't be more happy."*.

Entity and aspect level: Lastly, the aspect level also called feature level, performs a finer-grained analysis. Both document and sentence level analysis do not discover what exactly people like and did not like. Instead of looking at language constructs (documents, paragraphs, sentences, clauses, or phrases), the aspect level goes further and looks directly at the opinion itself. This approach is based on the idea that an opinion consists of a sentiment (positive or negative) and a target (of opinion). The goal here is to discover sentiments on entities and/or their aspects. For example, the sentence *"The Iphone's call quality is good, but its battery life is short"* evaluates two aspects: call quality is positive and battery life, of Iphone (entity). The sentiment on Iphone´s call quality is positive but the sentiment on its battery life is negative.

The diagram depicted in Figure 2.2 summarizes the definitions mentioned above.

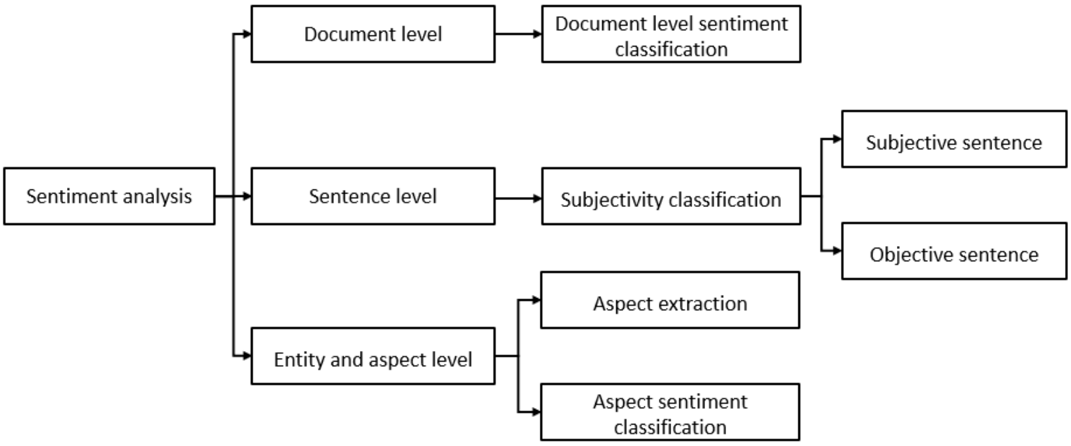


Figure 2.2: SA Different Levels

2.2 Stock Markets and Investing Psychology

There are various characteristics of human behaviour that have large influence in investing in the stock market. Some even argue that we are our own enemies when it comes to correct investing decisions, particularly due to emotion and bias.

As expected, the way prices in the stock market move is largely the result of behaviour and actions of participants, namely the investors. We know that the stock market is by nature a discount mechanism. That is, investors are trying to predict what is going to happen in the near future, let us say in the next 6 to 12 months [1]. To do so, some investors may look preferentially at the fundamentals of economies and companies – mostly leading indicators – or looking just at past prices of assets – mostly lagging indicators. These two fields are commonly known as fundamental analysis (i.e. studying the economics) and technical analysis, respectively, the latter via studying data plots. Not surprisingly, there are hybrid approaches as well, in particular when technical analysis is used for timing decisions. Some practitioners go further by claiming that markets are efficient in the sense that investors as a group take rational decisions, that is, all known information is priced in. [2]

However, it may not be the case. Indeed, behavioral financial in economics has drawn some attention in the recent past. Investors are irrational, and probably not just once. For example, investors, as human beings, particularly men, may be overconfident on their ability to take wise decisions and overoptimistic so they would underestimate risks in their predictions. [20]

Also, there is a phenomenon of herd behaviour to consider. Fair to be said that, in general, groups tend to make better decisions than individuals. Let us call it the wisdom of the crowd: The more information is shared, the more discussion is held, the better is the decision-making

[1] During the recent COVID-19 economic meltdown, the stock market rebounded spectacularly once the discovery of a vaccine was announced. So, at the time we were witnessing that employment and economy were in disarray due to the lockdown imposed, but the stock market was already roaring.

[2] We leave aside illegal behaviour such as inside trading.

process. But there is the madness of crowd behaviour as well. Otherwise how would we explain the 21st century bubble of internet stocks, or the 17th century bubble of tulip bulbs in Netherlands, to name a few? There is an idea of self-fulfilling and reinforcement of group thinking. And this is getting greater as more and more social media plays a bigger role in our lives.

There is another aspect worth mentioning: loss aversion. The reality is that losses are considered far more undesirable than the equivalent gains are desirable. This frames the way choices are made, where human behaviour depends on values assigned to both gains and losses.

To finish, we draw attention to the fact that emotions of pride and regret affect behaviour as well. In general, investors have difficulty to admit they were wrong, and even the regret is worse when is involving loved ones. But when they were correct they may tell everybody about the accomplishment. So it would be no surprise if they were holding losing stock positions just to avoid regret, which is wrong. With a non-emotional mindset, some will argue that a correct approach would be sell the losers and hold the winners. [20]

2.3 Sentiment Analysis in Social Media

With digitalization and the onset of web technologies, the growth of sharing, and expressing opinions over the Internet has been unprecedented [13]. All this relies mainly on social networking sites, including Twitter, Instagram, Facebook, YouTube and, more recently, Chinese networking sites like YUBO. Indeed, social networks, microblogs and other platforms generate massive amounts of information in the Web. So governments, consumers and brands exploit these platforms to share promotional deals, exchange ideas, run campaigns to increase awareness on social issues, and to promote products and services [30]. With large volume of data flowing over such platforms, new means of understanding consumer perceptions have paved the way for business, and they strive to apply algorithms for analysing opinions and sentiments of people [28]. There are different ways to evaluate content on social media for business analytics and intelligence, monitoring fraudulent activities and to grasp sentiment analysis consumer feedback.

SA is about identifying and extracting human sentiments from unstructured text using ML and NLP capabilities [23]. The most common approach is ML as it facilities the training and understanding of the dataset gathered from social media. Also, rule-based and lexicon-based techniques are widely used in practice and mentioned in the literature.

2.3.1 Twitter Sentiment Analysis

Twitter is a social media platform, launched in 2006, that allows registered users to follow and communicate with other user's via size-limited messages. The message, the so called *tweet*, are limited to 280 characters[3]. Twitter allows one to follow or to be followed by any account. The *follow* feature, which does not requires approval, is one of the main differences between Twitter

[3] https://developer.twitter.com/en/docs/counting-characters

and other social media platforms, such as Facebook, LinkedIn or Instagram, where users need to approve social connections.

Twitter platform is mostly used to share information about a variety of topics in real time. Twitter has been growing exponentially since it was launched and caught the attention of several users and entities in different areas. At the time of writing, it has an average of 330 million active users monthly, generating over 600 million tweets per day [4]. The list of users ranges from influencers to brands, including names like Elon Musk, Bill Gates, Amazon, or Apple [17].

The user's profile contains, among other information, the following: biography, photo, website, location, number of followers and followers. The tweets can include hyperlinks and, as mentioned above, have a 280 characters limit. Since they can be sent in real time they are often classified as instant messages. However, the difference resides in the fact that tweets are posted on Twitter´s website so making them permanent, searchable and accessible to everyone, whether they are members or not. Features like the *Retweet* allow any user to re-post tweets. Hashtags, designated as (#), can be included in tweets and are used to denote a topic of a conversation and can be extremely helpful to search any tweet based on the topic. Mentions, designated as (@), can be used to reference a user by his username. Twitter has also an additional feature intended for financial markets called Cashtag ($), which is used to identify a company´s stock symbol [27]. We should draw attention to the fact that this last feature plays a crucial role in the scope and context of our study since it is used in the search criteria.

For the sake of context, in the following we illustrate some examples of using Twitter in the different domains.

Politics. The works in [11] and in [3] relate to predictions of U.S 2016 presidential elections and Brexit Voting, respectively. In [11], the authors seek to determine whether Twitter can be used as an effective polling method. For that, the authors developed a system that incorporates two approaches: tweet volume and sentiment. Results showed that when using volume, Twitter is not a good resource for polling and predicting popular vote. This result is unexpected, as previous studies have found the volume to be a good predictor. On the other hand, when using the sentiment score as pooling tool, authors stated that the polling results achieved similar results as the IBD/TIPP tracking poll, which is considered one of the most accurate tools. In [3], authors used Twitter for the EU referendum in the UK to predict the Brexit vote. They developed a system based on user-generated labels known as hashtags (#) to build training sets related to the Leave/Remain campaign and subsequently implemented a Support Vector Machine (SVM) algorithm to classify tweets. Results suggested that Twitter has the potential to be a suitable substitute for Internet polls and be a useful complement for telephone polls.

Health care. Twitter can also be used for health care. The COVID-19 pandemic caused a significant public health crisis triggering issues such as economic crisis and mental anxiety. In [1], the authors developed a tool to detect and track top involved users´ sentiments and sentimental

[4] https://financesonline.com/number-of-twitter-users/

clusters over time. Experiments were conducted focusing on the topics that have most trendiness on Twitter. More particularly, the paper proposes a model to identify users´sentiment dynamics for top-k trending sub-topics related to COVID-19, and it also detects the top active users based on their involvement scores on the underlying trending topics. Therefore, this study successfully derived a model that calculates user´s involvement scores towards Query topics and determines the top 20 involved users to analyze their corresponding sentiment.

Financial Markets. When it comes investing, there are always risks either when wagering or exiting the stock market [27]. In [26] authors investigated whether measurements of collective mood derived from large-scale Twitter feeds are correlated over time to the value of the most famous stock market index, the DJIA index (DJIA). The authors analysed the daily content of tweets using two mood tracking tools: Opinion finder to measure positive versus negative mood and GPOMS to measure mood in terms of six dimensions (Calm, Alert, Sure, Vital, Kind and Happy). Results indicate that the accuracy of DJIA predictions can be significantly improved by the inclusion of specific public mood dimensions. The authors stated an accuracy of 87.6% in predicting the daily up and down changes in the closing values of the DJIA and a reduction of the Mean Average Percentage Error by more than 6%. Following this work, authors in [24] used Twitter as a tool for forecasting stock market movements. The study involved performing a collection of correlation and regression analyses to compare daily mood with daily changes in the price of the British FTSE 100 index. Findings suggest evidence of causation between public sentiment and the stock market movements, in terms of the relationship between mood and the daily closing price. To conclude, results show promise for using SA on Twitter data for forecasting market movements.

2.4 Algorithmic Approaches

ML approaches are based on a variety of machine learning algorithms and a collection of syntactic and dialectal features. SA is considered as a consistent text classification challenge by this method [21]. This approach uses the sentiment polairty as primary input data and performs statistical analysis to predict the output. It is further divided into supervised and unsupervised learning techniques. Machine learning approach uses word-based characteristics to learn a model that can classify feelings, subjectivity or reactions [29]. Before employing it to the real information set, this approach operates by training an algorithm in which a certain specific inputs have known outputs and later working with the new unknown information [6].

ML is a predictive method based on previous findings that the current information classification is predicted. There are many ML algorithms in use today, and new ones continue to emerge as data analytics directly produce advances in technology [10]. These algorithms can also be implemented to classify text effectively.

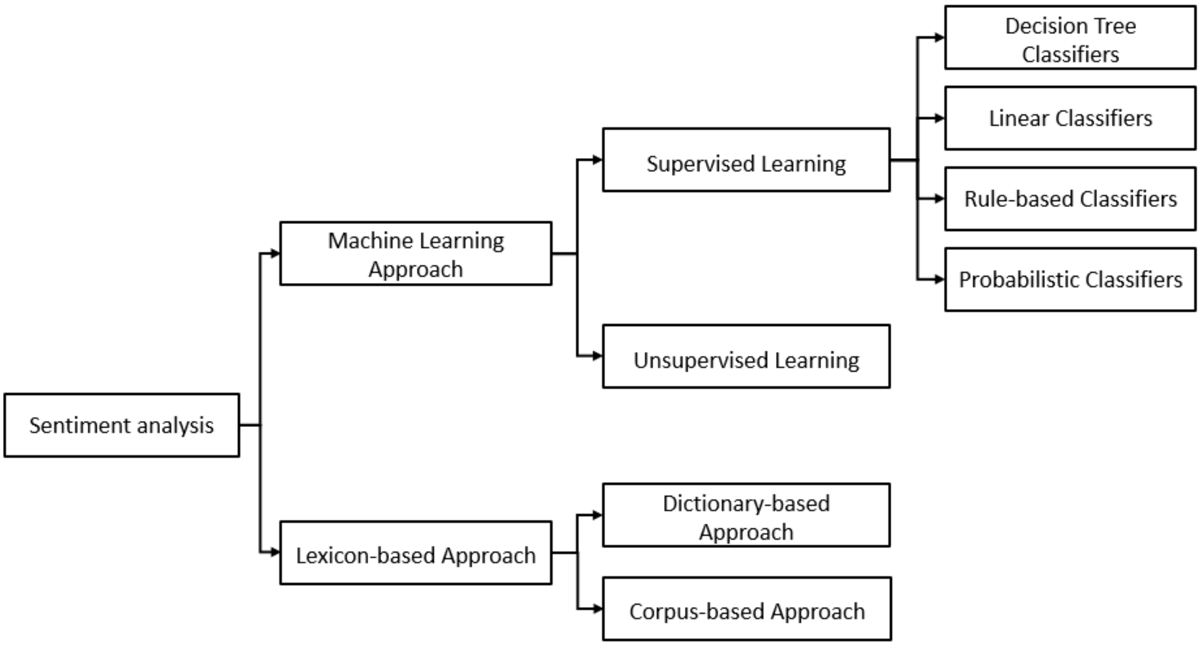


Figure 2.3: Sentiment classification techniques

2.4.1 Sentiment Classification

SA is defined by the process of identifying sentiments in text and labelling them as positive, neutral, or neutral, based on the emotions expressed. Using NLP techniques to interpret subjective and unstructured data through the use of sentiment analysis can help one to understand how a particular subject or group of subjects feel about any topic.

In this study, several sentiment classification approaches will be taken into consideration to see which one presents the best accuracy and reliability. For the ML approach, we focus on NB and classifiers, from scratch. For the Lexicon-based approach, we elect predesign state of art classifiers such as Textblob, VADER and Flair and we aim to compare results so to evaluate which approach betters detects the underlying sentiment expressed in a tweet.

There are many tools that offers the means of extracting advanced features from text. However, most of these tools are complex to handle and requires time before one can explore their full potential. In the present work, VADER [12] is going to be used to determine the polarity of tweets and to classify them according to multiclass sentiment analyses [8].

2.4.2 Machine Learning Approaches

In [6], the authors define the ML approaches as the set of strategies encompassing the training of an algorithm with a training data set before applying it to a target data set, also known as testing data.

In the supervised learning approach, algorithms are first trained with labelled data, whose inputs and outputs are known, to then classify the unknown data. Two of the most predominant and popular supervised learning techniques are the NB Classifiers and Decisions Trees Classifiers [16]. In[15] the authors describe the NB classifiers as a simple probabilistic model based on the Bayes rules and involving a conditional independence assumption. This assumption has a minimal impact on the accuracy of our classifier, on the other hand, it makes the algorithm much faster in terms of classification, being therefore widely used in this sphere.

On the other hand, [15] describes the Decision Trees Classifiers as a tree in which internal nodes are represented by features, edges represent tests to be done at feature weights and leaf nodes represent categories that results from the above. It categorizes a document by starting at the tree root and moving successfully downward via the branches (whose conditions are satisfied by the document) until a leaf node is reached. The document is then classified in the category that labels the leaf node. Decision Trees have been used in many applications in speech and language processing. Per definition, Random Forest is a supervised classification algorithm, and it is an ensemble learning technique based on decision tree algorithms that have become popular in the recent year due to the performance and robustness that this algorithm has when compared to similar machine learning algorithms, such as SVM or even NB [5]. This ensemble technique combines the predictions of some base estimators constructed with decision tree algorithm to enhance robustness over an individual estimator. RF grows a lot of classification trees, which is called forest. If we want to classify new data, each tree gives its category prediction as one vote. The forest chooses the category that has majority voting. In general, the more trees in the random forest the higher accuracy results given [9].

2.4.3 Lexicon-based Approaches

In scenarios where training data is scarce, classic supervised models may become impractical [4]. Lexicon-based systems require a pre-compiled sentiment lexicon corpus, where each word has an assigned sentiment value [22]. These lexicons can be either manually or automatically generated.

Additionally, the lexical based approach depends on constructing a Lexicon, described as a "structure that keeps track of words and possibly information about them" where the words are referred to as "lexicon items". Once the lexicon is constructed, the overall polarity of the text is then found by a weighted count of those lexical items [25].

In [4] the authors identify two types of lexicon-based approaches: 1) unsupervised models that do not require a training corpus of labelled documents, and 2) mixed models that combine lexical knowledge with labelled documents. Since that we are going to implement two ML-based algorithms from scratch, this study will use predesign state of the art lexicon-based algorithms to compare results and evaluate which approach is more suitable to classify Twitter data. That being said, we choose to use the following lexicon-based algorithms: Textblob, VADER and Flair.

VADER is a rule-based model for general sentiment analysis and compared its effectiveness to 11 typical state-of-the-practice benchmarks, including Affective Norms for English Words (ANEW), Linguistic Inquiry and Word Count (LIWC), the General Inquire, Senti WordNet, and machine learning oriented techniques that rely on NB, Maximum Entropy, and SVM algorithms [12]. In [8], the author describes the development, validation, and evaluation of VADER. The researcher used a combination of qualitative and quantitative methods to produce and validate a sentiment lexicon that is used in the social media domain. VADER uses a parsimonious rule-based model to assess the sentiment of tweets. The study showed that VADER improved the benefits of traditional sentiment lexicons, such as LIWC. VADER was differentiated from LIWC because it was more sensitive to sentiment expressions in social media contexts [12] [8].

Flair is a Natural Processing Language (NLP) framework designed to facilitate training and distribution of state-of-the-art sequence labelling, text classification and language models. The core idea of the framework is to present a simple, unified interface for conceptually very different types of word and document embeddings [2]. Lastly, the Textblob is a Python library designed for text mining, text analysis and text processing. Textblob is performed at a sentence level, i.e. first, it takes a dateset as the input then it splits the review into sentences. This is specially important when it comes to social media text processing. Textblob returns a tuple with two parameters called polarity and subjectivity, with the polarity being the count of positive and negative sentences and subjectivity indicating the amount of factual information present in a sentence [4].

[This page has been intentionally left blank]

Chapter 3.

System Architecture and Framework

Contents

This chapter proposes the underlying system architecture of a solution, and the framework to build upon, that supports our work of SA in the stock market. At its core, it introduces four main blocks of data processing and visualization, as part of the overall architecture. It specifies the SA models and ML techniques used and provides a detailed explanation of the implementation steps.

[This page has been intentionally left blank]

Chapter 3
System Architecture and Framework

3.1 Introduction

A crucial research aspect of this work is to understand how tweets that are somehow related to a particular company listed in a stock market exchange may affect its stock price movement. Among thousands of companies we may choose from, in this study we will focus on a pre-defined set of companies listed in the United States Stock Market (US stock market). To be more specific, they are components of the S&P 500. The reason is that this index is by far the one that better represents the largest stock market in the world, at least from a investor's point of view [1].

It is worth pointing out from the outset that the bulk of effort relates to collecting and processing twitter financial information, first through search queries targeting companies listed in the S&P 500, and subsequently via application of SA techniques. Overall, tweets are classified into three categories: positive, negative, and neutral. It is expected that a positive tweet will lead to an increase of the company's stock price, while a negative tweet will lead to a opposite effect; neutral tweets are the ones that are considered as not having a significant impact on the stock price movement.

Hence, in the following sections we will introduce and discuss the general architecture to work upon. It means we will discuss in detail its main processing blocks and how they are intertwined to each other. As expected, the architecture is aligned with classic stages we find in a data science project.

3.2 General Architecture

Given the research goals set, we have to delineate a strategy to dealing with tweets, stock prices and then the associated data analysis. It is clear that such strategy must fit into a broader data processing pipeline, so we borrow the idea of a data pipeline from the Cross-industry standard process for data mining, also known as CRISP-DM. Hence, the system architecture we propose for the scope of this study comprises four different data processing blocks, that are linked to each other and data will flow through them. The combination of these processing blocks [2] in a layered manner leads to the architecture illustrated in Figure 3.1, which can be summarised as follows:

[1] The others equally important that can be considered for the US stock market would be the DJIA and the NASDAQ index (NASDAQ).

[2] We also call them *modules* interchangeably.

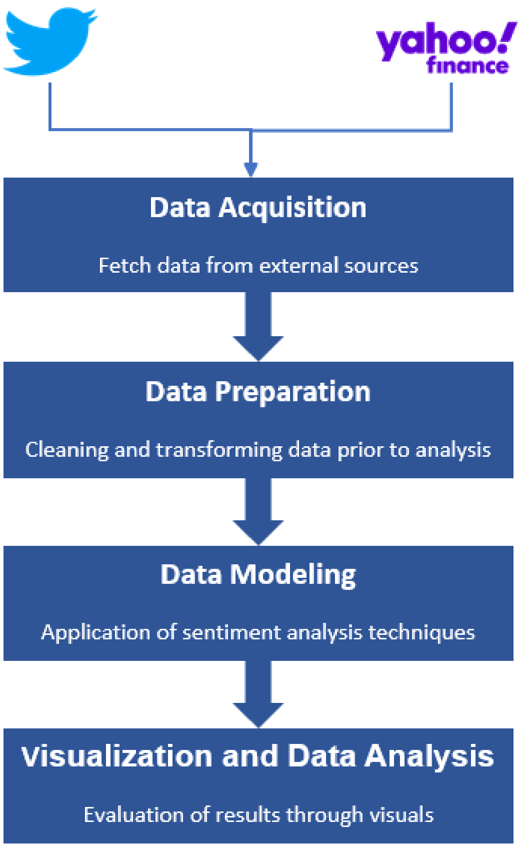

yahoo! finance
Data Acquisition
Fetch data from external sources
Data Preparation
Cleaning and transforming data prior to analysis
Data Modeling
Application of sentiment analysis techniques
Visualization and Data Analysis
Evaluation of results through visuals

Figure 3.1: Proposed layered architecture with four distinct data processing blocks.

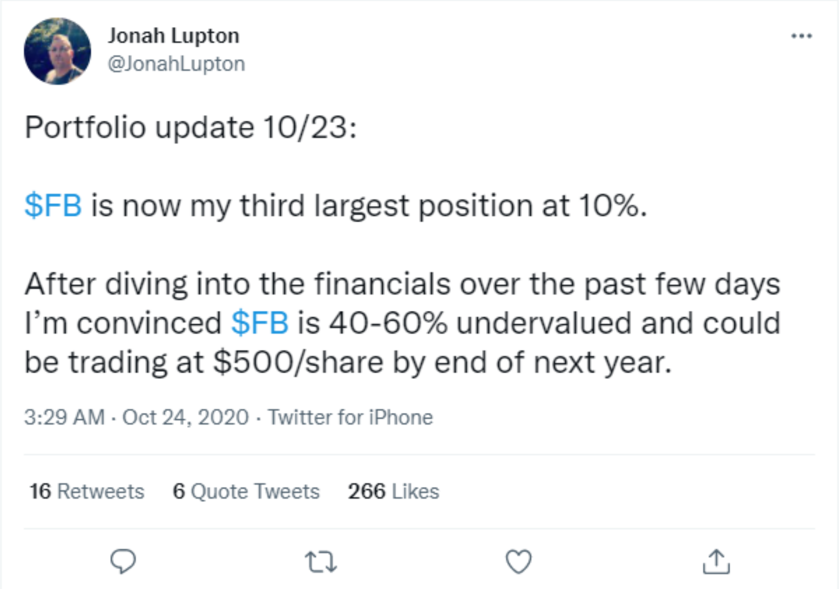
1. **Data Acquisition** – Module responsible for obtaining the raw data that is required and to validate if the data that has been received is what it was asked for. Taking into consideration the scope of this study, the data sources under consideration are:
   - **Twitter**, to retrieve tweets using a scrapper fit for the purpose.
   - **Yahoo Finance**, to gather information about price quotes of stocks listed in the S&P 500.
2. **Data Preparation** – Module responsible for cleaning and structuring the data so it can be ready for creating a SA model.
3. **Data Modelling** – Module responsible for setting up a SA model that will describe the data we want to analyse, based on SA techniques. In general, the model is supported by ML algorithms.
4. **Visualization and Data Analysis** – Module responsible for providing visual idioms that allow us to interpret and assess the results of SA.

3.3 Data Acquisition

This processing block is responsible for acquiring and storing the data extracted from external sources that are going to be used downstream. In that respect, as mentioned before, there are two types of data to be collected: Twitter data and stock market data. For each type of data, there will be a suitable Jupyter notebook that it is implemented accordingly, so the task of acquiring and storing data can be accomplished. Further details about this module are presented in the following sections.

3.3.1 Twitter Data Collection

This module is responsible for collecting tweets and creating our own Twitter database. Figure 3.2 shows the kind of tweet we are looking for.



Jonah Lupton
@JonahLupton

Portfolio update 10/23:

$FB is now my third largest position at 10%.

After diving into the financials over the past few days I'm convinced $FB is 40-60% undervalued and could be trading at $500/share by end of next year.

3:29 AM · Oct 24, 2020 · Twitter for iPhone

16 Retweets 6 Quote Tweets 266 Likes

Figure 3.2: Example of a tweet we may be interested on.

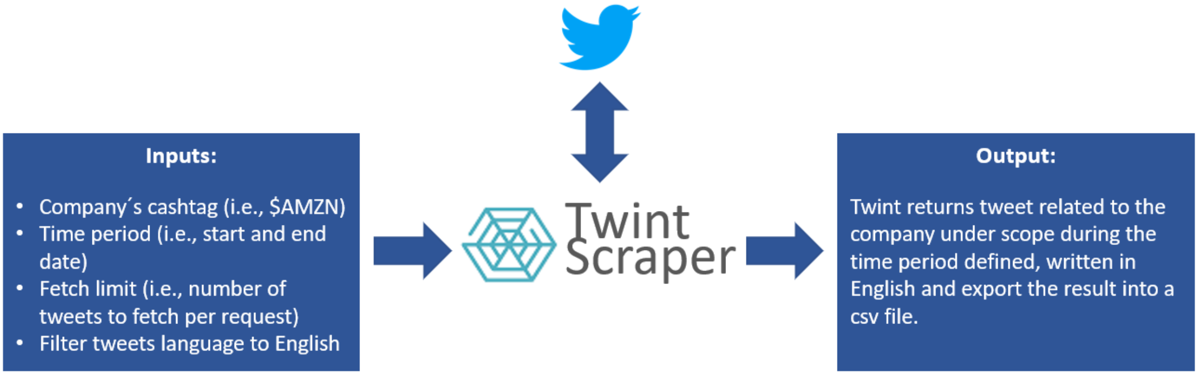
At the time of the implementation of this module, Twitter provided an official Twitter search API offering three types of subscription: *Standard*, *Premium* and *Enterprise*. The *Standard* subscription is free, however it only allows (i) simple queries against the indices of recent or popular tweets and (ii) limits search, just for the last seven days. So really this option is not suitable since it requires a lot of time and long periods of preparation in advance to collect enough information. The other two subscription options bypass these limitations but they are paid tools.

With such context in mind, we have decided to use an open-source web scraper called Twint. Twint is an advanced Twitter scraping tool written in Python that allows one to scrape a user's followers, following, tweets and more, while evading most of Twitter API limitations. By using Twitter's search operators, Twint offers a high level of customization when querying tweets from specific users, tweets relating to certain topics, hashtags, trends, or sort out sensitive information like e-mails and phone numbers. In the following we highlight the main benefits of using Twint *vs* the Twitter API. They are:

- Twitter API has a limit of fetching only the most recent 3200 tweets for the last seven days, while Twint has no limit of downloading tweets.
- Easy to use, very fast, free and no rate limitations.

- No initial sign-in or configurations required for fetching data.

The process of Twitter data collection is illustrated in Figure 3.3, showing how Twint is used to retrieve tweets. In its essence, this robot receives the cashtag symbol of a company under scrutiny (e.g. $FB for Facebook) as input and scrapes the Twitter's official page searching information, in this case tweets, about the company's finances and stocks performance. The mechanism used by this robot for fetching data is very similar to the one when a user uses the Twitter's search to find any specific information. The results are displayed, yet they continue to be loaded by scrolling down the page. Based on the input that has been set – the search query – for the company of interest, in this case Facebook, the robot searches tweets with the cashtag symbol $FB, and saves all related tweets by scrolling down through the Twitter's search results page.



Inputs:

- Company´s cashtag (i.e., $AMZN)
- Time period (i.e., start and end date)
- Fetch limit (i.e., number of tweets to fetch per request)
- Filter tweets language to English

Output:

Twint returns tweet related to the company under scope during the time period defined, written in English and export the result into a csv file.

Figure 3.3: Process related to data acquisition of tweets.

Table 3.1 shows part of the data schema underlying the Tweet object data model, with the most interesting attributes.

Attribute	Description
Id	Unique identifier of the Tweet
Date	Creation date of the Tweet
Username	Unique name that a user identifies themselves with
Tweet	The actual text of the Tweet
Language	Language of the Tweet
Mentions	User's mention included in the Tweet body preceded by the @ symbol
URLs	Wrapped URL for the media link embedded in the Tweet
Hastags	Combination of keywords or phrases preceded by the # symbol
Cashtag	Identifier of a company ticker symbol preceded by the $ sign
Retweet	Is a re-posting of a Tweet
Geo	Contains place details in GeoJSON format

Table 3.1: Partial data schema underlying the Tweet object data model.

3.3.2 Yahoo Data Collection

The data regarding companies' stock quotes are collected via a Yahoo API, called Yahoo Finance, that accesses information about financial news, stock quotes, press releases and financial reports. All the data provided by Yahoo Finance is free of charge, with more than five years of daily Open–High–Low–Close (OHLC) prices available.

The process of fetching financial data is illustrated in Figure 3.4. It is supported by a Python library to deal with the Yahoo Finance API, and stock quotes are collected using the Ticker module, which allows to get market and meta data for a specific stock.

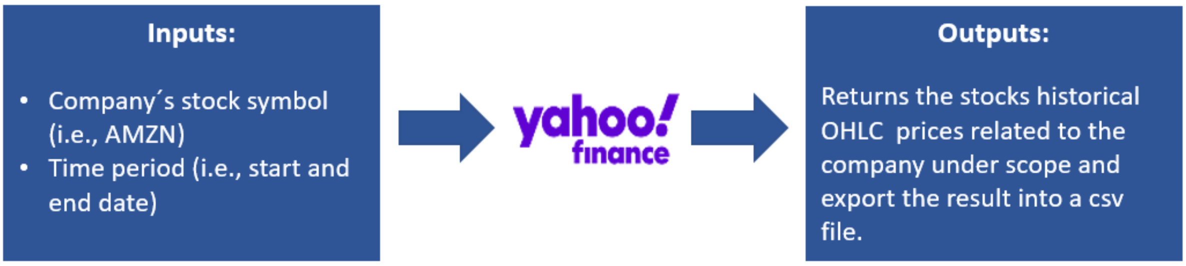

Inputs:
• Company´s stock symbol (i.e., AMZN)
• Time period (i.e., start and end date)
yahoo! finance
Outputs:
Returns the stocks historical OHLC prices related to the company under scope and export the result into a csv file.

Figure 3.4: Process related to data acquisition of financial information via Yahoo Finance.

The financial variables that are useful for finding relationship and/or measure of the impact of tweets in the stock market are the `Open`, `Close`, `High`, `Low`, `Adj. Close` and `Volume`. As mentioned above, the Ticker module enables us to collect this information via the Yahoo Finance API. As in the case of tweets, the stock data is retrieved in a tabular format, with columns representing the stock data attributes and rows representing each record of information for a single day. Table 3.2 shows the data attributes available for each record of information.

Attribute	Description
Date	Day of the stock trading
Open	Price at which the stock began trading, for a given day
Close	Price at which the stock ended trading, for a given day
High	Stock highest price, for a given day
Low	Stock lowest price, for a given day
Adj. Close	Adjusted price at which the stock ended trading, for a given day
Volume	Number of shares traded, for a given day

Table 3.2: Stock data attributes of interest provided by Yahoo Finance.

3.4 Data Preparation

This module is responsible for gathering, compiling and transforming the data that has been collected from the data sources. Its primary purpose is to ensure that the raw data is carefully adjusted and/or enhanced prior to any relevant processing and analysis. It works as a kind of first sieve of information. This is even more important in the case of unstructured data, like the twitter data we are dealing with.

It is worth pointing out that the downloaded stock market data does not require special preparation to be workable further down the line. But there is one aspect worth mentioning at this point: Whereas we may have tweets every day, the stock market is not always open for business, like in the weekends or holidays, so we may have missing stock quotes for a particular day. In the end, if one wants to merge both types of data by day, as we do, data has to be adjusted accordingly. But notice that a proxy for a missing stock quote is not a proper data point [3].

Having said that, this Section hereafter is all about processing of twitter data. Indeed, it is a major challenge for this module and its main job to accomplish. Recall that tweets may contain emoticons, pictures, mentions, URLs, as well as irrelevant content.

As a first hurdle to overcome, it is required that a set of pre-processing operations have to be applied to the twitter data, in order to ensure that data is properly formatted, with consistency, and therefore it can be used for analysis down the line. It is a cleansing task. Figure 3.5 highlights these pre-processing operations that are at stake here.

[3] For example, a moving average of a stock price should not include data points from days that the stock market exchange is closed.

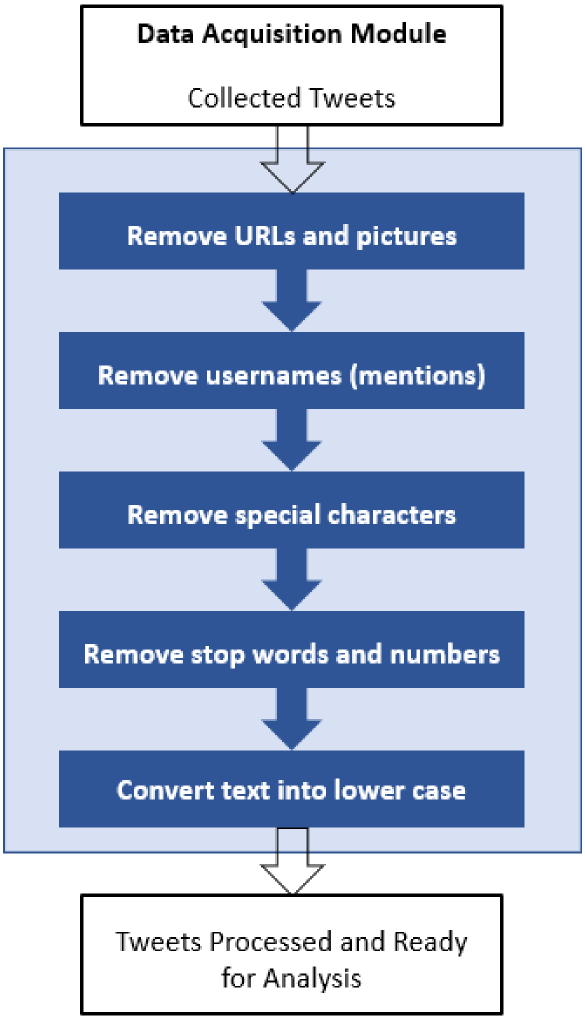

Data Acquisition Module
Collected Tweets
Remove URLs and pictures
Remove usernames (mentions)
Remove special characters
Remove stop words and numbers
Convert text into lower case
Tweets Processed and Ready for Analysis

Figure 3.5: Process related to data preparation of tweets, focussing on the cleansing of data.

Overall, the details concerning such cleansing operation in the context of this study are as follows:

1. **Removing URLs and pictures** – A tweet do often contain URL links and pictures. Since these aspects do not add sentimental value in a meaningful way, they are removed and replaced by empty spaces.

2. **Removing usernames (mentions)** – A tweet can include usernames as *@username*, often referred to as mentions. Again, they are removed since they do not express any kind of sentiment.

3. **Removing special characters** – Special characters like emoticons and punctuation marks are removed. Notice that emoticons can be extremely useful to determine the underlying sentiment of a tweet however, these are highly complex to interpret and therefore considered to be noisy labels, which can affect the performance of our sentiment models. These characters are replaced by empty spaces.

4. **Removing stop words and numbers** – The text of a tweet may contain a lot of words, often called stop words, that do not express any kind of emotion nor adding any value at all but noise to the sentiment classification. And numbers also do not add value for the purpose of this study. So they are removed. Examples are words like *the*, *of*, *a*, or *12*.

5. **Convert text into lower case** – A tweet is converted into lower case to contribute to data consistency for downstream use. This step also adds robustness to the model at later stages of training and classification, since it avoids that a given algorithm may classify identical words wrongly due to case sensitive issues.

In the following we present some examples of tweets we have collected, both in raw and cleansed versions.

```
2020 In 12 Stunning Charts $GE $WMT $AAPL $AMZN $CSCO $INTC $MSFT $FB $SPX
 ↪  $GOOGL  https://t.co/LlilteDj4S
```

```
in  stunning charts ge wmt aapl amzn csco intc msft fb spx googl
```

```
$NXTD BOUNCING AFTERHOURS! Adding heavy here  $SPY $SPX $QQQ $ES_F $NQ_F
 ↪  $RTY_F $ZB_F $GC_F $NDX $RUT $AAPL $NFLX $AMZN $TSLA $FB $MSFT $DIA $NDX
 ↪  $IWM $QCOM $GDX $DAX $BYND $TWTR $GLD $SLV $GE_F $BABA $TLT $LYFT $VXX
 ↪  $TVIX $VIX $XLE $XOM $JPM $GS $GOOG $DIS $IBM
```

```
nxtd bouncing afterhours adding heavy here  spy spx qqq esf nqf rtyf zbf gcf
 ↪  ndx rut aapl nflx amzn tsla fb msft dia ndx iwm qcom gdx dax bynd twtr
 ↪  gld slv gef baba tlt lyft vxx tvix vix xle xom jpm gs goog dis ibm
```

```
Net margin (%) among largest #stocks $SPX $SPY 1.  APPLE INC. $AAPL: 20.9 2.
 ↪  MICROSOFT CORPORATIO. $MSFT: 32.3 3.   https://t.co/D6MvHgxMPe, INC.
 ↪  $AMZN: 5.0 4.  ALPHABET INC. $GOOGL: 20.8 5.  FACEBOOK, INC. $FB: 32.0
```

```
largest stocks spx spy   apple inc aapl    microsoft corporatio msft       inc
 ↪  amzn    alphabet inc googl    facebook inc fb
```

Finally, notice that the preparation of data could have included other tasks. For example, to investigate the cleansed data by drawing some preliminary statistics, with the purpose of enriching the data itself. However, we believe that the operations described above concerning data cleansing are enough to secure a proper input for the next stages in the pipeline.

3.5 Twitter Data Modelling

Once data input for the model is available, we draw now attention to the specifics of the model to classify tweets. But before going further, we have to set boundaries. First, there will be different models available so we can compare results and performance of various algorithms. Secondly, there will be models that are largely customised and therefore should be built almost from scratch. On the other hand, there will be models that are easily configured but relying on well-known NLP packages.

Having that in mind, the provided models are based on the following algorithms and/or packages:

- NB.
- RF.
- Textblob.
- VADER.
- Flair.

Both NB and RF algorithms are incorporated into two custom models. One of them is relying on the Machine Learning in Python (scikit-learn) toolkit and uses the so-called BoW technique. The other one relies on Spark ML, which means in this case we will have a more powerful data distributed solution. Then, in a different group, we are using popular NLP packages suitable for SA: Simple rule-based Textblob and VADER, and embedding based Flair. Notice that while the rule-based approach only focus on individual words not context, the embedding based approach also focus on the idea of word closeness by aggregating similar words in a n-dimensional space. Further details about the models are presented in the following sections.

In the case of the custom models mentioned above, we should take particular attention to the specifics of a ML workflow, a concept depicted in Figure 3.6, in order to reach better outcomes.

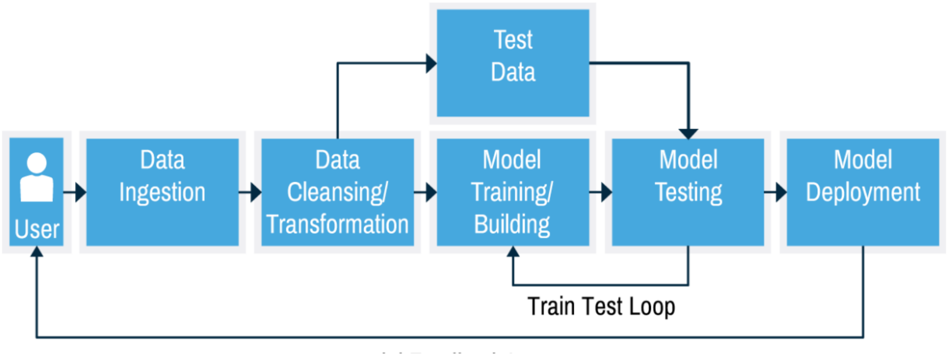


Figure 3.6: Classic ML workflow, including the mentioning of feedback loops.

In that respect, a critical task is to train and evaluate the model, which usually requires splitting the data into two sets, e.g. following a 70-to-30% train-test rule and then applying the model to both of them but separately. Figure 3.7 illustrates the concept.

However, we are taking a different approach: we are not driven by random spilt of our own

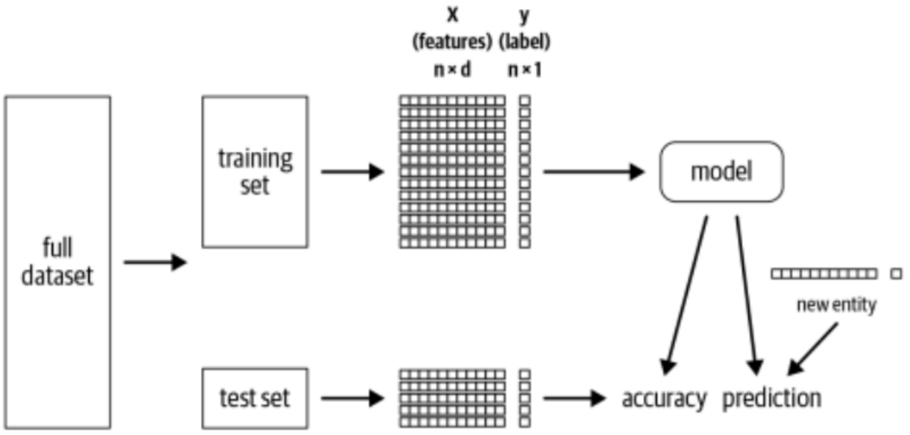


Figure 3.7: Classic splitting of data into training and testing sets in order to train and evaluate the model.

downloaded tweets, as we have no idea about their SA (that is our goal), but rather using information related to tweets that have been already classified elsewhere. The key restriction is that such classified tweets must be somehow related to the ones we aim to classify, e.g. written in English, etc. Nevertheless, it is advisable to pay attention to the size of each part in the split, as it would be nice to hold comparable sizes of the classic 70%-to-30% train-test rule.

In the end, in the context of SA, we may predict different emotions attached to a tweet. But for this study we are considering just three particular sentiments: positive, negative, and neutral. If there is no prediction established yet, we use the label unknown.

Lastly, it is important to emphasize that the training data used to train our classifiers came from the NLTK library (NLTK) [4] library. This library provides over than 50 corpora and lexical resources, within the natural language processing field. For this study, we used a corpora specially designed for the Twitter classification task. This corpora contained 5.970 pre-classified tweets into Positive, Negative or Neutral. It is worthily to point out however, that the classification is not equally distributed. This imbalance must be noted and can introduce some bias to our classifiers. The impact of this imbalance will be assessed in the sections below.

3.5.1 Bag-of-Words Model

This model is based on building a vector of features describing each cleansed tweet that it is going to the classified by estimators included in a ML pipeline. More specifically, by ML algorithms overall. It comprises the following main steps, in sequence:

1. **Tokenization** – The goal is to break up a sentence or paragraph into specific tokens or words so that we can have a more abstract representation of human language for computers to work with. It can be done sentence by sentence, splitting of just words.

[4] https://www.nltk.org/.

2. **Stemming** – It is the process of finding the root of words and removing the endings from words so we can get rid of things like tenses or plurality. Stemming is a rule-based approach that reveals some inconsistencies when cutting suffixes in words, accordingly to certain rules. But special care is required when breaking down words using stemming to avoid overstemming – when words are over-truncated – or understemming, when two different words are stemmed from the same root.

3. **Word list dictionary** – This data dictionary is built by counting the occurrences of every unique word across the data [5]. Though, before building the final word list for the model, we should first have a look to what we have at this point. Usually, the most common words are the typical stop-words, which were already filtered out. But, as the purpose of this analysis is to determine sentiment from tweets, words like *not* and *n't* can have a great influence too. Therefore, such words will be whitelisted. Still, there are some words occurring too many times that should also be filtered out. As a final outcome, the word list is saved to a *csv* file, so the words can be used later on. Figure 3.8 highlights the kind of word list we may have.

4. **Feature engineering as a BoW** – Building a vector of features that resembles the text to be classified. The features will derive from the word list dictionary mentioned above, and we will end up knowing whether the text contains a particular word from the word list or not. The BoW is ultimately converted into numbers as the ML algorithms to be used in the next classification stage cannot ingest raw text.

   Notice that we could have added extra features but we are short on that. Indeed, we believe that the extra computational cost required would not pay off. For example, we could have added features such the counting of uppercase words, of question marks, of hashtags, of quotes, etc.

5. **Prediction** – Use of estimators in the ML pipeline to predict each tweet category, given the numeric version of BoW. Recall that the estimators here will uphold the strategy we have set in relation to the classic splitting of data into training and testing sets mentioned above at the beginning of 3.5.

6. **SA Classification** – Establishing the final sentiment for each tweet, given the predictions obtained in the previous step.

Both the tasks of tokenization and stemming are carried out using NLTK. This library was developed with the purpose of helping to create Python programs to deal with human language data. More specifically, the tweets are tokenized using the *nlkt.word_tokenize* feature and then, stemming is done using *PorterStemmer*. Recall that the tweets we are using are written in English.

Figure 3.9 highlights some of the most common words found in a set of tweets (BoW), alongside related SA classification. As a note, it seems skewed data distribution will be a problem to

[5] This is a training data set.

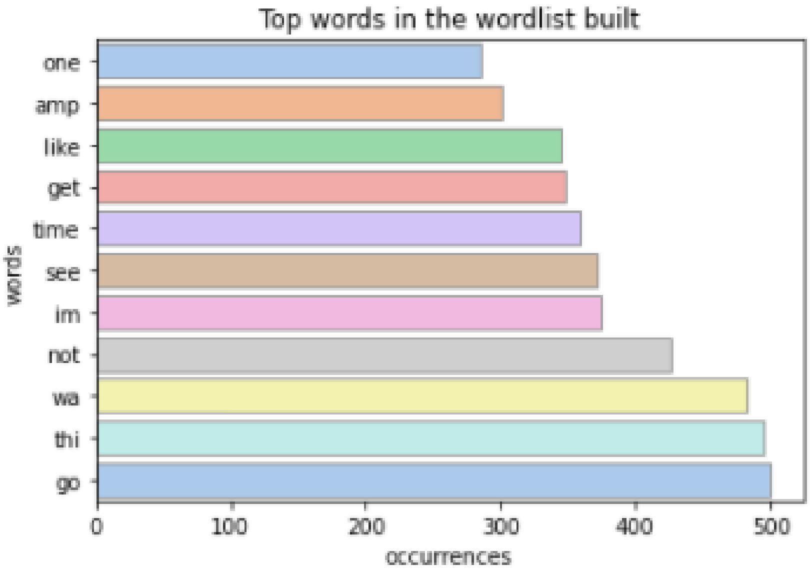


Figure 3.8: Example of top words in word list dictionary, given some tweets.

distinguish negative sentiments from other classifying classes, which surely depends upon the profile of input data.

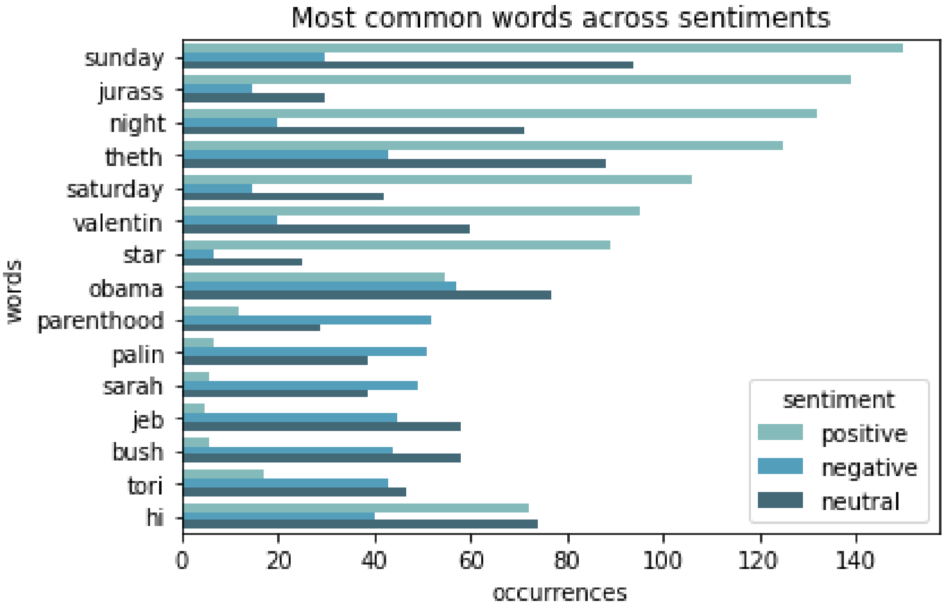


Figure 3.9: BoW and associated classification of sentiment.

As far as ML algorithms are concerned, we have considered two algorithms: NB and RF. We present a short description of them in the following paragraphs.

Naïve Bayes (Bernoulli). NB is one of the simplest and fastest classification algorithms for large amounts of data, in this case for millions of posted tweets. NB methods are a set of supervised learning algorithms based on applying Bayes' theorem, as described in Section 2.4.2, with the *naive* assumption of conditional independence between every pair of features given the value of the class variable. There are a lot of such methods available, each one fitted according to the underlying classification task. The most suitable method in our case is the Bernoulli Naïve. It is based on the Bernoulli distribution, where features are independent binary variables, thus being the most appropriated classifier since the goal here is to classify tweets within the binary interval of $[-1, 1]$, with -1 being negative and 1 being positive.

Random Forest. RF is also a very popular supervised learning algorithm used to solve classification problems. The classifier is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. In other words, RF works by initially creating decisions trees based on randomly selected data sample, then it gets a prediction for each tree, and finally selects the best solution through a voting mechanism. The classification output is similar to the NB, whose goal is to label sentiment polarity within the binary interval of $[-1, 1]$, with -1 being negative and 1 being positive.

3.5.2 Spark ML Model

As a custom model, this model resembles the BoW model but mainly in concept. As a matter of a fact, the divide line between the two of them is that, whereas the BoW model relies mostly on some Python packages for its implementation, e.g. NLTK, so leading to a centralized solution, this Spark ML model relies primarily on Apache Spark and its ML library, so leading to a distributed solution. Very briefly, Apache Spark is a popular open-source distributed cluster-computing framework designed for large-scale distributed data processing [31]. Hence, as a distributed solution, with implementation of ML algorithms based on parallel programming, we may afford to classify larger sets of tweets, and faster, in comparison to the BoW model. Furthermore, we can deploy the code in various modes, namely locally – one machine / one node – or distributed, in a cluster of various machines / nodes.

Also, the specification of the ML workflow here goes further in the sense that we can chain multiple operations of transforming data and learning algorithms (classifiers) into a single pipeline. It is really a powerful Spark ML tool at our disposal. And in this particular case, we are using the same kind of algorithms that are used in the BoW model, that is, NB and RF. So for the time being we keep a similar learning strategy.

Overall, the model comprises the following main steps, in sequence:

1. **Tokenization and Steaming** – Likewise the BoW, although we could have used the Spark ML tokenizer instead [6].

2. **Training/Testing Data Split** – Likewise the BoW, the training data relates only to tweets whose emotion/category is different from `unknown`. That is, the ones we believe we can

[6] The thinking was that we would like to compare primarily the classifiers from different models and somehow these two tasks were kind of *in the border* to the data cleansing stage.

set its classification from well-known sources in advance [7]. Conversely, the testing data relates to tweets whose emotion/category is so far unknown.

3. **ML Pipeline Configuration** – Setting the pipeline upon which data will be fit and then predictions are computed. It includes dealing with feature engineering and setting classifiers. The implementation of this step looks like the content of the Listing 3.1. (We are using Python code for the sake of explanation, and as it is implemented.)

4. **ML Model Training** – To fit the training data into the ML pipeline. Notice that, as in the BoW model and according to the training/testing split mentioned above, we are not using a classic random split but we are driven by data that we already consider as properly classified.

5. **Prediction** – To apply the testing data, that is, the tweets of interest, to the ML pipeline (a transformation) in order to get predictions and probabilities associated.

6. **SA Classification** – Setting out the final sentiment for each tweet, given the results obtained in the previous step.

```python
# ...
hastft = HashingTF(numFeatures=2**16, inputCol="new_text", outputCol='tf')

# compute the Inverse Document Frequency
idf = IDF(inputCol='tf', outputCol="features")
# ...
label_string_idx = StringIndexer(inputCol = "emotion", outputCol = "label",
    handleInvalid = "keep")

# =======================
# Classifier to be used
# =======================
# 1. Naive Bayes
# supported options: multinomial (default), bernoulli and gaussian
classifier = NaiveBayes(smoothing=1.0, modelType="multinomial")

# 2. Random Forest
# classifier = RandomForestClassifier(numTrees=10)

pipeline = Pipeline(stages=[hashtf, idf, label_string_idx, classifier])
```

Listing 3.1: Example of ML pipeline configuration in the Spark ML model.

3.5.3 Textblob Model

The Textblob model is a natural language processing library used to process textual data. It returns the polarity and subjectivity of a sentence. Polarity is measured in the interval of $[-1, 1]$, with -1 being negative sentiment and 1 being positive sentiment. Subjectivity is measured

[7] We have a source of 5421 tweets already classified that we believe are trustworthy to be used in the context of this study, and considering as well the kind of tweets we aim to classify.

between $[0, 1]$ and quantifies the amount of factual information contained in the text, with 0 meaning personal opinion and 1 meaning objective opinion.

```
# Import Model
from textblob import TextBlob

# ================
# Classification
# ================
for sentence in df['Tweet']:
    blob = TextBlob(sentence)
    polarity.append(blob.polarity)
    subjectivity.append(blob.subjectivity)
```

Listing 3.2: Example of ML pipeline configuration in the Textblob model.

3.5.4 Vader Model

The VADER model is part of the package and it is used for text sentiment analysis. It can be applied directly to unable data and classifies text according to polarity (negative/positive) and in intensity (sentiment compound) of emotion. Polarity is measured in the interval of $[-1, 1]$, with -1 being negative sentiment and 1 being positive sentiment. On the other hand, intensity, commonly referred as sentiment compound gives us the sentiment score of a given tweet by summing up the intensity of each word in the tweet.

```
# Import Model
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()

# =================
# Classification
# =================
for i in range(df['Tweet'].shape[0]):
    compound = analyzer.polarity_scores(df['Tweet'][i])["compound"]
```

Listing 3.3: Example of ML pipeline configuration in the VADER model.

3.5.5 Flair Model

It is a framework used for natural language processing. Flair relies in a neural network architecture (long short-term memory) and delivers state of the art performance for text classification. Contrarily to VADER and Textblob, whose focus is on polarity classification, the goal here is to determine if a given sentence tweet is objective or subjective, fact or opinion. This classifier takes into the account several parameters like, the sequence of words, the sequence of letters and the user of intensifiers ("too", "very", etc.) when classifying a tweet. The classification output however is similar to the remaining algorithms, it given being classified against their polarity (negative/positive). One caveat regarding Flair is that only classifies tweets into positive or negative, whereas the remaining others classifies a tweet into positive, negative or neutral.

```python
# Import flair

sentiment_model = flair.models.TextClassifier.load('en-sentiment')

# =================
# Classification
# =================
for sentence in df['Tweet']:
        sample = flair.data.Sentence(sentence)
        sentiment_model.predict(sample)
```

Listing 3.4: Example of ML pipeline configuration in the model.

3.6 Visualization and Data Analysis

Once the ML model was built and then used to classify the tweets of interest, yielding to a conclusion about the inherent sentiment – positive, neutral or negative – it is time to grasp insight from the data we were able to gather, both initially and computed afterwards. That is the purpose of this module.

3.6.1 Visualization Techniques

The visuals generated are based mainly on the tool Microsoft Power BI Desktop (free version), whose main features are its interactive visualizations, business intelligence capabilities and user-friendly interface. For that and since all the previous modules were developed in a Python environment, results had to be exported into a csv file before being uploaded into Power BI. Once the data is uploaded, with Power BI one can create a variety of visual idioms to represent information. Recall that a proper image is a powerful mechanism to convey messages effectively.

3.6.2 Data Analysis

With results from classifiers stored in csv files, and using the visualization tools at disposal, it follows the data analysis. As mentioned, the main goal now is to extract insights from the collected results.

The overall analysis follows a three level strategic approach, as explained below.

The first and second level are similar in a sense that they focus on the prediction results of the different algorithms. But while the first level is purely technical, with the assessment of the ML prediction results, the second level deals with the SA prediction itself, so also concerned about the surrounding context. That is, the main difference between these two levels is that while the first level is only worried about the technical execution of the algorithms, the second level is an extension of the first but with context involved. By context we mean the underlying task to consider, which is to classify twitter data into positive, negative or neutral.

Lastly, after classifying the inherent sentiment of tweets, it follows the third and final level of analysis. The outputs from the previous two levels are in a raw format, meaning that data is organized and displayed at record level. This means that data is in its most granular form, with the number of records being the number of tweets classified and the columns of these records containing the different classifications attributes. To bypass this limitation, data is then aggregated by day and their scores are also averaged daily. This allows two things: first we are able to merge the financial attributes with the remaining twitter related fields in a single file and secondly, because by aggregating the data we are able to gain relevant insights, recognise trends and identify underlying patterns. It is worth remembering that the stock market data was acquired on a daily basis.

3.7 Summary

This Chapter proposes a set of four different processing blocks as composing the system architecture and the framework that supports the quest to achieve our proposed goals. These processing blocks are linked to each other in a way that allows data flowing between them. The implementation is done via Jupyter notebooks and usage of Python libraries, as for example to acquire financial data via Yahoo Finance. In general, a lot of effort is put on pre-processing operations and compiling and transforming data, so to make sure that data is in a standardized format and ready for downstream use.

There are various SA models that have been considered and implemented, four in total and each one relying on different methods. Tables 3.3 and 3.4 highlight the main features of the models proposed. For the BoW, which is based on building a vector of features, the performance drops significantly as the size of vectors increases. This becomes impractical to classifying large amounts of information, such as the number of tweets we expect to deal with. But the ML model bypasses this limitation and reduces the time drastically to predict results. This is mainly due to the fact that ML model relies on Apache Spark, which is suited for large-scale distributed processing. On the other hand, the Textblob, VADER and Flair models were built following the same ML pipeline approach.

Finally, the way data is displayed and analysed should follow a proper framework and guidelines. In that respect, most of the work is carried out using the interactive visualization tool Microsoft Power BI.

	NB	RF
Type	Probabilistic	Decision tree
Output	[POS, NEU, NEG]	[POS, NEU, NEG]
Advantage (+)	Fast and suited for text	Perform both regression and classification tasks
Disadvantage (-)	Assumes all features are independent	Requires much computational power
Interpretation	POS: 1, NEG: -1, NEU: 0	POS: 1, NEG: -1, NEU: 0

Table 3.3: Main characteristics associated to NB and RF models.

	Textblob	VADER	Flair
Type	Rule based, Bag of words	Rule based, Bag of words	Character-level LSTM neural network
Output	[-1,1]	[-1,1]	[POS, NEG]
Advantage (+)	Evaluate subjectivity Heurist	Empirically validated for microblog-like contexts	Recognizes typos, negations, and intensifier
Disadvantage (-)	Does not have the heuristics of Vader typos	Typos (OOV)	Training is computationally expensive
Interpretation	POS: > 0.5, NEG: < -0.5, NEU: rest	POS: > 0.5, NEG: < -0.5, NEU: rest	POS: 1, NEG: -1

Table 3.4: Main characteristics associated to Textblob, VADER and Flair models.

[This page has been intentionally left blank]

Chapter 4.

Evaluation

Contents

This chapter outlines the evaluation and experiments undertaken for this study. First, it is presented the data collection and the subsequent data processing steps, along the considerations that form the scope of our work. There are 10 selected companies. The results are classified by their polarity and validated using visuals designed for the purpose. Finally, a more detailed visualization and data analysis is provided in relation to two of the selected companies.

[This page has been intentionally left blank]

Chapter 4
Evaluation

4.1 Introduction

The answering to the research questions set out in Section 1.2 requires a clear evaluation framework to work with. Although it is important to have a sound computational environment to carry out experiments, it is also very important to design the experiments we are interest on very carefully, and how to carry out them. Then we will be able to draw conclusions properly.

As expected, the experiments and evaluation we are about to describe will follow the guidelines and inherent principles associated to the computational architecture discussed in Chapter 3. Recall that most of its implementation relies on notebooks and the Python language.

Turning now to the evaluation task itself, first and foremost we have to establish the data we are going to collect for our experiments and analysis.

As mentioned in Section 3.1, a crucial aspect of our research is to figure out how tweets related to a particular company may affect its stock price movement. Among the constituents that are part of the S&P 500 index, we have selected 10 companies for the experiment. We could have selected more companies but we believe the number considered is just enough given the research context and tasks ahead. The selection was based on criteria such as the sector the company is part of, its popularity, as well as the relevance for the analysis considering the time frame of interest.

Just for context, as of September 2021, the S&P 500 index includes 505 stocks [1], and it is divided into 11 sectors, each of them with companies operating in similar industries. The respective weightings by market capitalization are as follows: Information Technology (27.6%) Health Care (13.3%), Consumer Discretionary (12.4%), Financials (11.4%), Communication Services (11.3%), Industrials (8.0%), Consumer Staples (5.8%), Energy (2.7%), Real Estate (2.6%), Materials (2.5%), and Utilities (2.5%). And the top companies by index weight are: Apple, Microsoft, Amazon, Facebook, Alphabet (twice, with different type of shares), Tesla, Nvidia, Berkshire Hathaway and JP Morgan Chase.

The time frame we are considering is the year 2020, which turns out to be a very special year, given the effects of the COVID-19 pandemic in our lives, in the world economy and consequently in the stock market. Recall that not all industries have experience the same kind of negative effects. For instance, technology and the so-called companies from the lockdown theme were doing better, whereas, in the extreme opposite side, there were companies badly hit like airlines, from hospitality and tourism industries, or from the oil industry, to name a few.

The selected companies are mostly from top sectors in terms of market capitalization. Notice that high valued companies are typically more popular amongst investors, which matters to

[1] It is 505 instead of 500 because it includes two share classes of stock from five of its component companies.

us. The popularity is even more crucial when it comes to social media, in particular Twitter, as these companies are usually more tweeted. Also, the relevance of the companies and scope were considered. For example, following the temporal context mentioned above, the technology-based Facebook, Amazon, Apple, Netflix, and Alphabet (parent company of Google) (FAANG) [2] have recovered and performed quite well during the pandemic, whereas other companies from other sectors have experienced a very turbulent and difficult year. That being said, the 10 selected companies are as shown in Table 4.1.

Company	Stock Symbol	S&P 500 Sector	Cashtag
American Airlines	AAL	Industrials	$AAL
Amazon	AMZN	Consumer Discretionary	$AMZN
Carnival	CCL	Consumer Discretionary	$CCL
Disney	DIS	Communication Services	$DIS
Ford	F	Consumer Discretionary	$F
Facebook	FB	Communication Services	$FB
General Electric	GE	Industrials	$GE
Alphabet [3] (Google)	GOOGL	Communication Services	$GOOGL
Microsoft	MSFT	Information Technology	$MSFT
Exxon Mobile	XOM	Energy	$XOM

Table 4.1: Selected companies to support the experiments and evaluation.

In the following sections we will present the data – collected and subsequently prepared – in relation to the 10 selected companies. Once that is achieved, we provide a detailed analysis about two of those companies: Facebook and American Airlines.

4.2 Data Acquisition

As mentioned in Section 4.2, data acquisition is a basic module of our system and it is responsible for acquiring and storing the relevant data we need from external sources. Recall that we have implemented two Jupyter notebooks, one to collect information from Twitter and the other one from Yahoo Finance. The first one relies on Twint, an advanced Twitter scrapping tool written in Python, and uses predefined queries to find specific tweet information. The second one collects financial data, e.g. stock quotes from Yahoo Finance using an API from Yahoo, also written in Python.

4.2.1 Twitter Data Collection

Accordingly to the procedure set in Section 3.3.1, tweets are collected using a pre-defined search query, which contains a cashtag feature in order to identify, and extract, the tweets regarding the companies of interest. Recall that each company has its own cashtag, as depicted in Table 4.1.

[2] FAANG is an acronym referring to the stocks of the five most popular and best-performing American technology-based companies: Facebook, Amazon, Apple, Netflix, and Alphabet (parent company of Google). They have a combined market capitalization of nearly USD 7.1 trillion as of August 2021, representing roughly 17.5% of the S&P 500 index.

Regarding the time frame [4], as mentioned before we are considering for this study the year 2020. So the goal is to collect tweets posted between 1 January 2020 and 31 December 2020.

Table 4.2 shows the volume of downloaded tweets for this study.

Company	Tweets (#)	Daily Average (#)	Size (MB)
Amazon	343,347	941	207
Facebook	255,929	701	158
Microsoft	214,815	589	135
General Electric	125,048	343	85
Disney	111,836	306	66
American Airlines	95,907	263	63
Alphabet (Google)	87,200	239	52
Carnival	71,089	195	45
Ford	53,296	146	34
Exxon Mobil	50,585	139	31
	1,409,052	3,860	876

Table 4.2: Volume of tweets downloaded: Number of downloaded tweets for each company, alongside its daily average and size of downloaded data. (Ordered by number of tweets.)

4.2.2 Yahoo Data Collection

The financial data was collected via the Yahoo Finance API (See Section 3.3.2). Such historical data that was collected include the quotes of `Open`, `Close`, `Adj Close` and `Volume` for each stock of interest. But for the purpose of our study only `Adj Close` and `Volume` matters. Likewise the case of downloaded tweets mentioned above, the daily quotes were collected with the time frame set between 1 January 2020 and 31 December 2020. In total, we have downloaded 442 KB of data, being roughly the same size for each stock. (As the data columns and days were the same.)

4.3 Data Preparation

Once data is collected, the next step is to clean and format it properly. Otherwise the next stages in the overall process will not be functioning as planned.

In respect to twitter data, this operation is straightforward: we use a Jupyther notebook to sieve the twitter data as described in Section 3.4. The outcome are csv files with the schema depicted in Table 4.3 for every data set mentioned in Table 4.2.

The stock market data does not require a cleansing operation *per si*. Only to computing extra columns but useful and making sure we have filled data for every single day of the year, as recalled in Section 3.4.

[4] This time frame could be increased however, since the Twint scraper relies on the Twitter advanced search feature, makes the execution times impractical.

Id	Date	Tweet	Category
17951998	2020-06-12	facebooks libra has failed says switzerlands president	Negative
...	...	...	...

Table 4.3: Data schema of prepared twitter data.

Therefore, as far as stock market data is concerned, we end up with csv files holding the schema depicted in Table 4.4 for every company presented in Table 4.1.

Date	Adj Close	Volume
2020-04-27	186.72	67,378,800
...	...	...

Table 4.4: Data schema of prepared stock market data.

4.4 Twitter Data Modelling

Following the models described in Section 3.5, we have carried out some experiments to classify tweets. Recall that we have at our disposal five models: BoW, Spark ML, Textblob, VADER and Flair. The hardware used in the experiments had the following specification:

- Memory: 31,4 GiB
- Processor: Intel Core i7-8705G CPU @ 3.10GHz x 8
- Graphics: AMD Radeon graphics
- Disk: 1,0 TB
- Operating System: Ubuntu 18.04.5 LTS

We were able to use all the clean data sets available for every model except the BoW model. As a matter of fact, only the smallest data set – from Exxon Mobil – containing 50,586 tweets, and volume size of 31 MB, could be processed in the BoW model. It is not a surprise since the feature engineering stage in this model uses a huge Python matrix corresponding to a size of number of features times number of tweets. The management of such array, e.g. copying parts of it, is usually prohibitive in a normal desktop computing set with a simple Python ecosystem, which was the case for this model. Fortunately, we have a similar model but implemented using a sounded technology – the Spark ML model [5].

4.4.1 Tweet Category Prediction

After models are being trained, it follows the prediction of each tweet category. Tables 4.5 and 4.6 present for each experiment the time spent on both training and predicting.

In some cases, the temporal performance achieved can be further analysed taken into consideration the sub-tasks involved. Then we can figure out which are the most expensive computations

[5] We reckon that the limitation we have faced while using the BoW model has motivated us to use the Spark ecosystem as well.

		Spark ML				
Company	Tweets (#)	NB	RF	Textblob	VADER	Flair
Amazon	343,348	04:12	04:41	04:09	02:13	01:56:35
Facebook	255,930	03:18	03:44	03:08	01:43	01:32:19
Microsoft	214,816	10:04	10:29	02:40	01:26	01:18:33
General Electric	125,049	01:30	01:54	01:30	00:45	41:19
Disney	111,837	01:27	01:52	01:21	00:42	37:56
American Airlines	95,908	01:22	01:53	01:12	00:42	39:03
Alphabet (Google)	87,201	04:02	04:28	01:03	00:32	00:29
Carnival	71,090	00:59	01:22	00:52	00:29	26:44
Ford	53,297	00:47	01:12	00:39	00:20	19:15
Exxon Mobil	50,586	02:34	02:58	00:38	00:19	17:51

Table 4.5: Temporal performance of models – Time spent by each predictor to predict the tweets' category, including any time spent on initial settings. (in the hh:mm:ss format.)

and even how much they are in terms of the overall time spent. For example, let us consider the case of Facebook and the Spark ML model. While running the code, we figured out that:

- The initial operations of setting up the dataset (corpora plus the downloaded tweets) and subsequent cleasing, tokenization and stemming operations, as well as preparing the data structures to hold the data took approximately 2m 58s. [6] Notice that this stage is common to both NB and RF algorithms.
- The most expensive single operation from the initial operations was stemming (1m 17.3s), followed by tokenization. (30.7s) The cost time of the others were individually marginal.
- For NB, setting the pipeline and fitting the train data cost roughly 3.6s; the prediction itself took about 4.79s to perform, and then the storing of results was around 13s.
- For RF, setting the pipeline and fitting the train data cost roughly 27s; the prediction itself took about 4.82s to perform, and then the storing of results was around 14s.

These values are comparatively low in relation to the ones got from other models, in particular in the case of the prediction task itself. Of course, the Apache Spark technology and the distributed implementations available of the algorithms makes the difference. It is worth, as long as we have large datasets to process, and more so if the algorithms require complex data structures to work with.

As for the BoW model, the performance is poor. Not only considering the allowable size of datasets – only the smallest Exxon Mobil for that matter – but the temporal performance, shown in Table 4.6. Even if we compare its temporal performance against other models (Table 4.5) still, the result is poorer. We should emphasize that some models only really are advantageous in terms of execution time when datasets are of considerable size, like in the situation of Spark ML for example. And that was not the case of Exxon Mobil.

[6] This is the clean dataset that is going to used by the estimator, with schema [`Id, Category, Tweet`].

		BoW	
Company	Tweets (#)	NB	RF
Exxon Mobil	50,585	04:27	04:49

Table 4.6: Temporal performance of the BoW model – Time spent by each predictor to predict the tweets' category, including any time spent on initial settings. (in the hh:mm:ss format.)

4.4.2 Sentiment Analysis Classification

The final SA classification is obtained given the predictions computed before by each model of concerning. Tables 4.7 and 4.8 present the associated outcome in terms of profile of classifications, that is, the relative percentage of positive, neutral and negative sentiments.

But before looking at numbers of positive, neutral and negative sentiments, there are a few aspects we should draw attention to, as predictions and/or final SA classifications from any model/algorithm may be different in nature or form.

Recall that, while the NB and Flair classifiers only return categorical values such as positive or negative, the remaining classifiers return the sentiment polarity within a numerical scale, ranging between the interval of [-1,1], with -1 being negative and 1 being positive. To surpass this limitation, a query was developed to convert categorical values into numerical, otherwise it would not be possible to evaluate and compare the results of each technique. This step is critical to guarantee consistency in our analysis as scores must be within the same interval to allow comparisons and draw conclusions.

		Spark ML				
Company	Tweets (#)	NB	RF	Textblob	VADER	Flair
Amazon	343,347	34 | 36 | 30	99 | 1 | 0	9 | 90 | 1	22 | 74 | 4	48 | - | 52
Facebook	255,929	65 | 17 | 18	99 | 1 | 0	8 | 91 | 1	26 | 69 | 5	43 | - | 57
Microsoft	214,815	39 | 45 | 16	99 | 1 | 0	9 | 90 | 1	27 | 69 | 4	48 | - | 52
General Electric	125,048	22 | 29 | 48	100 | 0 | 0	7 | 92 | 1	16 | 74 | 9	49 | - | 51
Disney	111,836	66 | 16 | 18	99 | 1 | 0	9 | 90 | 1	21 | 74 | 5	36 | - | 64
American Airlines	95,907	51 | 12 | 37	100 | 0 | 0	8 | 91 | 1	40 | 57 | 3	51 | - | 49
Alphabet (Google)	87,200	34 | 50 | 16	90 | 1 | 0	8 | 91 | 1	19 | 76 | 4	47 | - | 53
Carnival	71,089	26 | 25 | 49	99 | 1 | 0	11 | 88 | 1	30 | 66 | 4	56 | - | 44
Ford	53,297	11 | 59 | 30	99 | 1 | 0	9 | 90 | 1	21 | 75 | 4	54 | - | 46
Exxon Mobil	50,585	23 | 37 | 40	99 | 1 | 0	9 | 89 | 1	20 | 75 | 5	50 | - | 50

Table 4.7: Prediction outcome computed by models. (% of positives | neutrals | negatives, out of number of tweets)

		BoW	
Company	Tweets (#)	NB	RF
Exxon Mobil	50,585	61 | 25 | 14	76 | 22 | 3

Table 4.8: Prediction outcome from BoW model (% of positives | neutrals | negatives, out of number of tweets)

4.5 Sentiment Analysis Evaluation

With all the results of computations in place, it is time to deliver data analysis and visualization, in particular the evaluation of SA. As outlined in previous sections, a crucial research aspect of this work is to understand how tweets about a given company may affect its stock price movement. More specially, this study aims to learn if the use of difference sentiment indicators can provide useful insights regarding stock price movements. As mentioned in Section 2.2, most of the traders in the stock market use indicators, particularly from technical analysis when timing is critical to get a successful trade. So a sentiment analysis indicator can be perceived as one more in their own package of tools. Actually, as briefly mentioned in the introductory Section 1.1, there are some commercial trading applications that provide sentiment indicators, although it seems not very complex in their design. [7] For example, an indicator would be to figure out how much more or less on a daily basis, in percentage, a particular listed company has experienced a number of related tweets.

Turning to the analysis itself, we have decided to use just two companies out of the scope of 10 companies available to run an extensive evaluation. The reason is two-fold: Firstly, it seems no significant extra value would be reached to providing answers to the research questions of concern. Secondly, it would be almost impossible to consider the 10 companies in this document given the scope and research framework of this dissertation.

Hence, the two chosen companies are Facebook and America Airlines. This decision is mainly due to the fact that those companies are from very different sectors, they are at different stages of growth and have shown distinct market behaviour in the recent past. One is from the *new* economy, the other one is from the *not so new* economy. As a FAANG company, Facebook is somehow a proxy the the other FAANG companies we have include in the set of 10. For the time frame considered, which coincides with the COVID-19 pandemic, all of them have experienced extraordinary gains and similar price trajectories in the stock market. But in the same period, companies from aviation and the tourism sector as a whole have had bad times, with great volatility in the stock market. America Airlines was one of them.

Recall that, as shown in Table 4.2, there are 255,930 tweets related to Facebook, so a huge number, and 95,908 tweets related to American Airlines. In the following sections we will present our findings about these two companies. To do so, we will rely primarily on results from four models/algorithms described before: Spark ML/NB, Textblob, VADER and Flair.

[7] We wonder if that is somehow related to compliance issues in public trading.

4.5.1 Facebook

The main findings in respect to Facebook can be drawn from Figures 4.1 and 4.2, alongside a correlation matrix in relation to various variables that is depicted in Figure 4.3, as a heatmap [8].

Starting with Figure 4.1, we have the volume of traded shares vs. respectively, the adjusted closing share price (top) and the number of tweets (bottom). Then, in Figure 4.2, we have the adjusted closing share price vs. respectively, the score of SA classifiers i.e. the sentiment indicators – Spark ML/NB, Textblob, VADER and Flair – (top) and the 20 simple moving average [9] from the same classifiers. (bottom)

From Figure 4.1 we can observe that 2020 was indeed a very turbulent year in terms of trading volume, with plenty of spikes and troughs. The reason behind this behaviour is definitely attributed to the COVID-19 pandemic, as it sparked a chain reaction in global markets, so becoming the dominant market force ever since and bringing a lot of uncertainty to the trading environment. Facebook hit its all time low towards the end of March, coincidentally at the time that major economies entered in lockdown. Then the stock price rebounded later, thanks to major stimulus packages, specially in the U.S. The spikes in trading volume observed between June and August may be due to the overall optimism towards the news about the vaccination rollout and the ease of lockdown restrictions.

Also, from the same Figure 4.1, (bottom) we can conclude that spikes in trading volumes are generally followed by a rise in the number of tweets, for a given day. This may be due the fact that investors remain hyper-sensitive to any COVID-19 news, whether positive or negative. We know that Twitter plays an important role within the financial community and spread of news. This trend however does not indicate us if the sentiment behind the spike is positive, negative, or neutral. It just shows that, somehow, there is a relation between the trading volume of stocks and the number of tweets having the Facebook cashtag. On the other hand, we can see that spikes in the number of tweets matches the spikes in the adjusted closing share price. We will delve into more detail the relationship between these two variables in the correlation analysis map below.

Figure 4.2 highlights that the four sentiment indicators illustrate different behaviours and patterns. While Spark ML/NB and Flair show great sentiment oscillations, Textblob and VADER are steadier in terms of the same oscillation. And overall, Spark ML/NB shows a higher level of optimism, followed by VADER, Textblob and lastly, Flair. The reason behind the Spark ML/NB higher scores may rely in the fact that it was the only one using our own corpora to training the data before predicting the sentiment. As mentioned before, the imbalance in the distribution of classified tweets may have introduced some bias to the predictor.

Overall, we can see that, apart from Flair, there was a growth in positive sentiment throughout the year. However, there is a lot of sentiment volatility increasing the visual noise present in

[8] Heatmap is a data visualization technique aiming to show the magnitude of variable as color, in 2D.

[9] A window of 20 for a moving average is one of the commonly used by traders in the stock market. Others are 9, 50, 100, 150 and 200.

the image. This may reduce the quality of insights that can be drawn from since no patterns or conclusions are easily identified.

Fortunately, the moving averages drawn surpass the limitation mentioned above [10]. So we were able to smooth the sentiment scores and therefore, making the analysis clearer and more insightful. The key focus for this visualization is measuring and assessing the rate of the rise or fall, also known as momentum, of sentiment scores and stock prices. Spark ML/NB and VADER are the ones following similar behaviour to stock prices movement. Apart from some shifts in the trajectory, both sentiment indicators could be useful in pointing out the strength and direction of the stock price. On the other hand, Textblob has a very steady performance showing a small reaction with regards to stock prices movements and lastly, Flair, despite showing a similar pattern, it is prone to negative sentiments.

Finally, the correlation matrix depicted in Figure 4.3 will also help us to draw some conclusions.

At a first glance, its possible to see that both trading volume and adjusted closing share price have a positive correlation, of 0.38 and 0.42 respectively, against the number of tweets for a given day. This may indicate that an oscillation in the number of tweets nexus to Facebook will lead to an oscillation to both stock price and trading volume.

On the other hand, when comparing trading volume against different classifiers, it is possible to infer that they have a neutral, slightly negative correlation. For the adjusted closing share price, we can see that the scores of Spark ML/NB and Flair show a similar level of correlation, and with VADER (`VaderCompound`) presenting the highest correlation of 0.62. Yet, if we look at the adjusted closing share price against the 20-days simple moving average of each classifier, we observe that there is a general increase in their correlation, outperforming the daily scores. Similar to the daily scores, VADER shows again the highest correlation of 0.77.

The last conclusion is a major one and may indicate that the moving averages of sentiment scores could be used as a momentum indicator to assess stock price variations.

[10] Indeed, for same reason moving averages are widely used by traders as a means to suppress noise in plots.

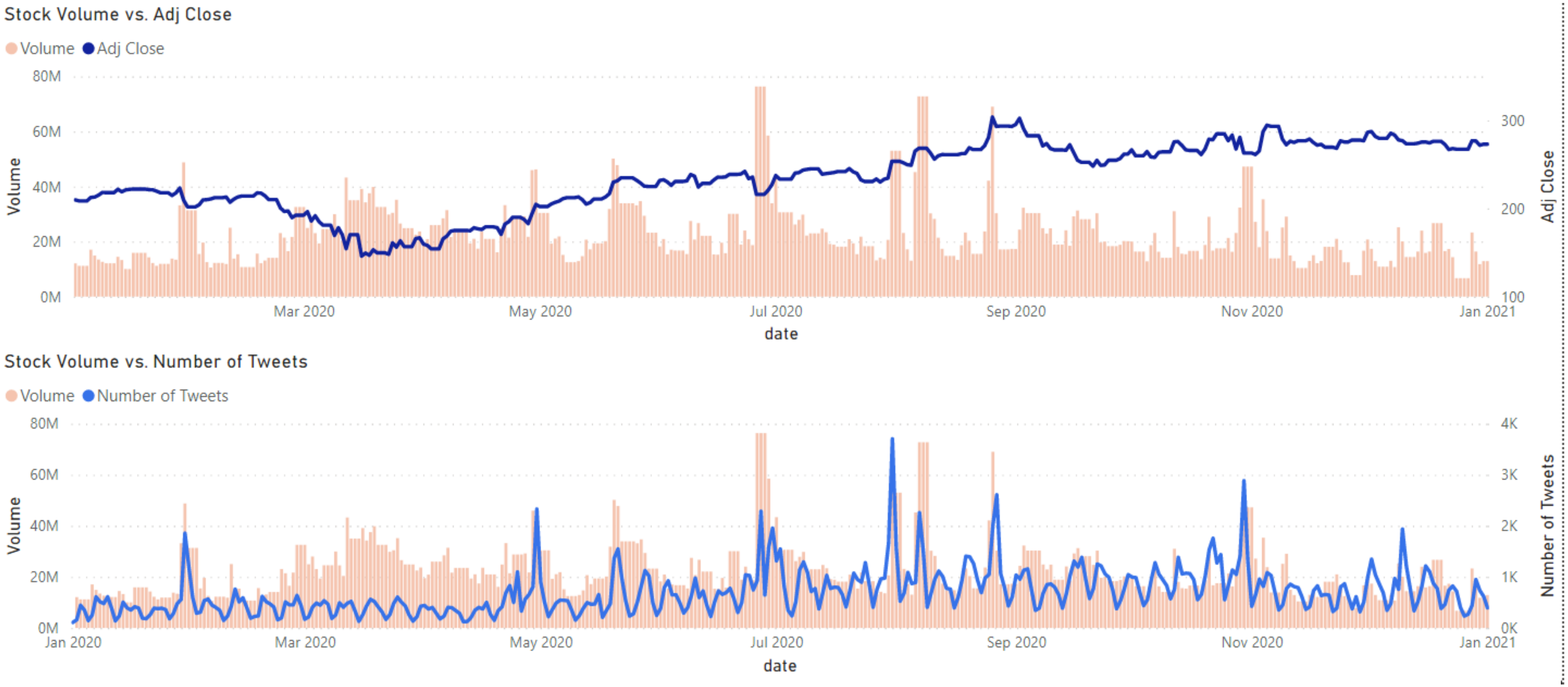


Figure 4.1: Facebook – Trading volume vs. respectively, adjusted closing share price (top) and number of tweets. (bottom)

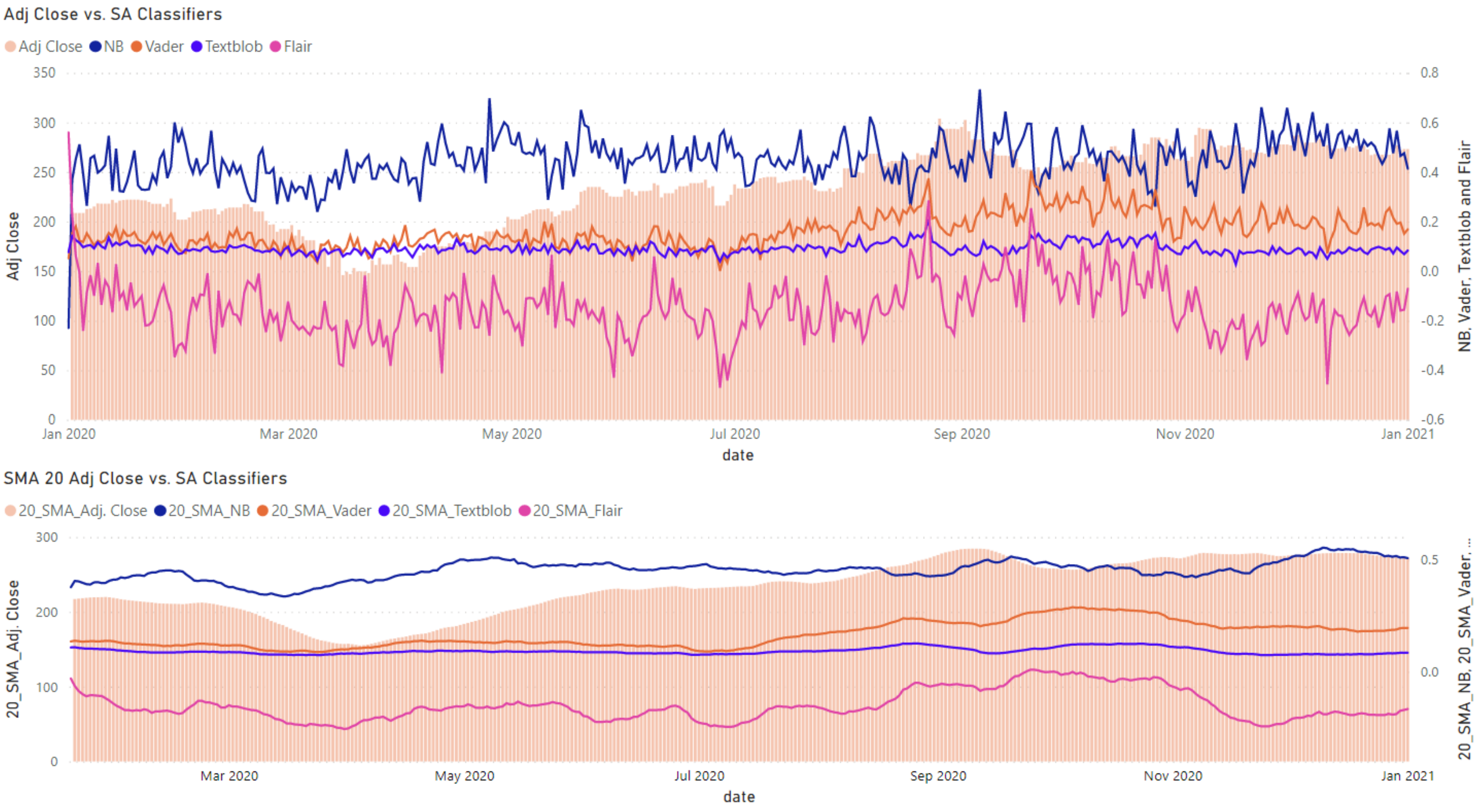


Figure 4.2: Facebook – Sentiment indicators, with adjusted closing share price vs. respectively, the score of SA classifiers – Spark ML/NB, Textblob, VADER and Flair – (top) and the 20 simple moving average from the same classifiers. (bottom)

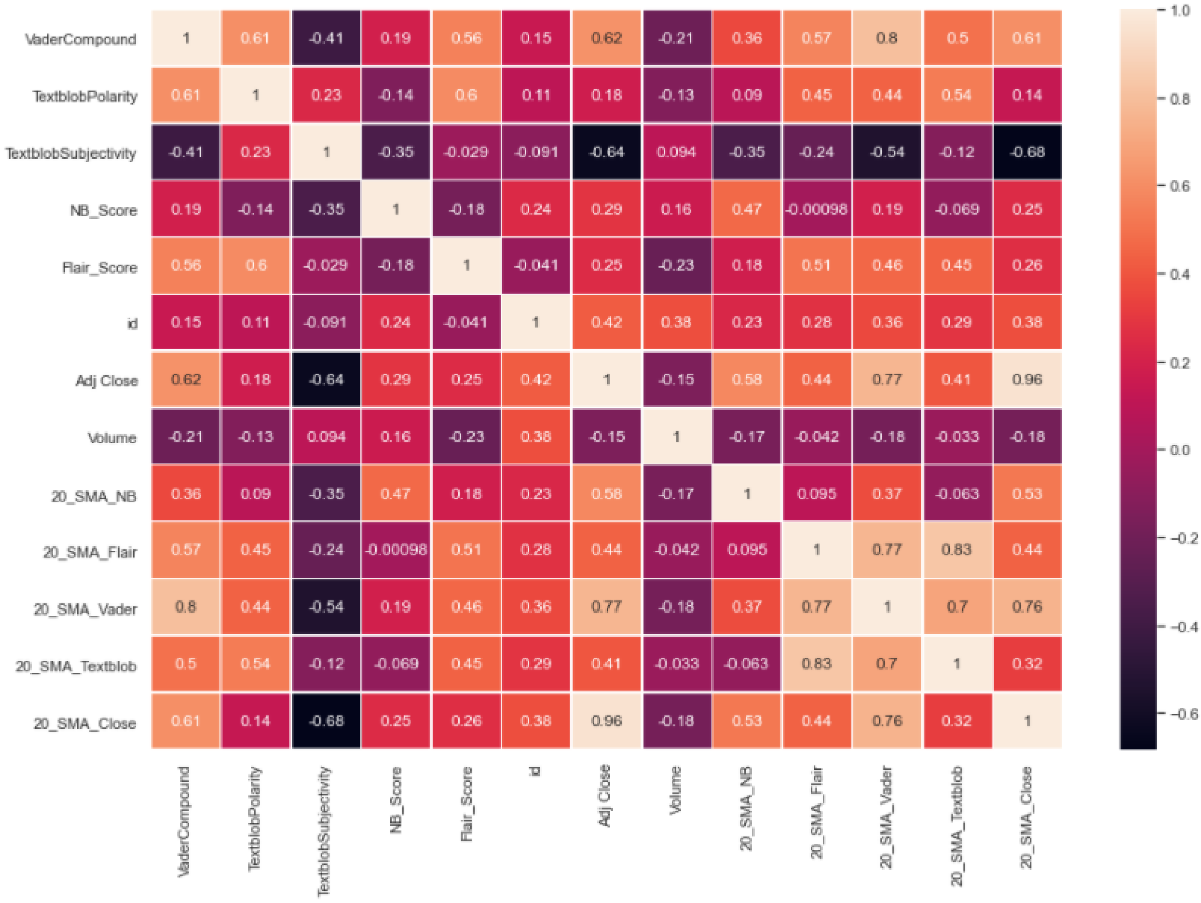


Figure 4.3: Facebook – Heatmap showing correlations among various variables.

4.5.2 American Airlines

Similarly to Section 4.5.1, the main findings in respect to American Airlines can be drawn from Figures 4.4 and 4.5, alongside a correlation matrix in relation to various variables that is depicted in Figure 4.6, as a heatmap [11].

Starting with Figure 4.4, we have the volume of traded shares vs. respectively, the adjusted closing share price (top) and the number of tweets (bottom). Then, in Figure 4.5, we have the adjusted closing share price vs. respectively, the score of SA classifiers i.e. the sentiment indicators – Spark ML/NB, Textblob, VADER and Flair – (top) and the 20 simple moving average [12] from the same classifiers. (bottom)

From Figure 4.4 we can observe that 2020 was an atypical year for the airlines companies, with tremendous impact on the industry. COVID-19 decimated the sector resulting in huge financial losses, in the other hand government's stimulus packages attenuated some of this loses. Still in Figure 4.4 we can note that until March 2020, date when the pandemic was declared, American Airlines had a pretty stable price and low volume of stocks being transacted. However, in March their stock prices stumbled due the beginning of pandemic resulting in an entirely airline industry virtually grounded, with a few exceptions such air cargo suppliers, causing jitters in the market. The spikes and also the increase in volume observed between April and August may relate to the approval of the 14 billion dollars U.S government aid packaged and to the arrival of the stimulus checks with a lot of millennials turning to the stock market to invest and make quick money, resulting in a phenomenon called stonks.

In Figure 4.4, it is also showed the stock volume versus the number of tweets containing the American Airlines cashtag, designed as $AAL. The first observation we can draw from the visual is that spikes in volumes almost matches the spikes in the number of tweets for a given day. This may be due the fact that American Airlines stock became highly attractive amongst the millennials, creating almost a herd mentality within the market. Another import factor relies on the fact that most of these investors use social media platforms to search and share investment ideas, with Twitter being one of the most used platform for this end. Similar to Figure 4.1, this trend does not show us the sentiment polarity behind the spike. Nevertheless, it shows that there is a relationship between the volume of stocks and the number of tweets trends. We will delve into more detail the relationship between these two variables in the correlation analysis map below.

Figure 4.5 highlights that the four sentiment indicators illustrate different behaviours and patterns. While Spark ML/NB and Flair show great sentiment oscillations, Textblob and VADER are steadier in terms of the same oscillation. And overall, Spark ML/NB shows a higher level of optimism, followed by VADER, Textblob and lastly, Flair. The reason behind the Spark ML/NB higher scores may rely in the fact that it was the only one using our own corpora to training the data before predicting the sentiment. As mentioned before, the imbalance in the distribution

[11] Heatmap is a data visualization technique aiming to show the magnitude of variable as color, in 2D.

[12] A window of 20 for a moving average is one of the commonly used by traders in the stock market. Others are 9, 50, 100, 150 and 200.

of classified tweets may have introduced some bias to the predictor.

Overall, we can see that the algorithms show similar patterns, again with Spark ML/NB and Flair showing greater sentiment oscillations, while Textblob and VADER are more steadier. In terms of sentiment scores, their results shows a slightly level of optimism, with Flair being the only one showing a negative score. As in Section 4.2, the huge oscillations in sentiment scores increases the amount of noise present in the visual. This makes the visual almost unintelligible dragging the quality of insights that can be drawn since no patterns or conclusions can be identified.

In Figure 4.5, and using the moving averages described in Section 4.2, we were able to smooth the sentiment scores and therefore, making the analysis clearer and more insightful. The focus here is in measuring the rate of the rise or fall, also known as momentum, of sentiment scores and stock adj. prices. Flair shows a similar pattern as the adj. close however, its prone to negative sentiments can giving a misleading lead. Spark ML/NB and Vader have similar behaviors with Vader showing a greater and steady optimism through the year and Spark ML/NBBayes having more oscillations but always in a positive trajectory. On the other hand, Textblob has a very steady performance showing a small reaction with regards to stock prices movements. This behavior is in line with the results in Figure 4.2, leading to the conclusion that the indicators that best represent and relate to the stock adj close price is VADER and Spark ML/NB.

Finally, the correlation matrix depicted in Figure 4.6 will also help us to draw some conclusions.

At a first glance, its possible to see that the trading volume have a positive correlation, of 0.51, against the number of tweets for a given day. This may indicate that an oscillation in the number of tweets nexus to American Airlines will lead to an oscillation to both stock price and trading volume.

On the other hand, when comparing trading volume against different classifiers, it is possible to infer that they have a neutral, slightly negative correlation. For the adjusted closing share price, we can see that the scores of Textblob and Flair show a similar level of correlation, and with VADER (`VaderCompound`) presenting the lowest correlation of -0.34, contrasting with Figure 4.3. Additionally, if we look at the adjusted closing share price against the 20 simple moving average of each classifier, we observe that there is no impact in the correlation results, which may indicate that for this case the moving averages might no be the most suitable indicator.

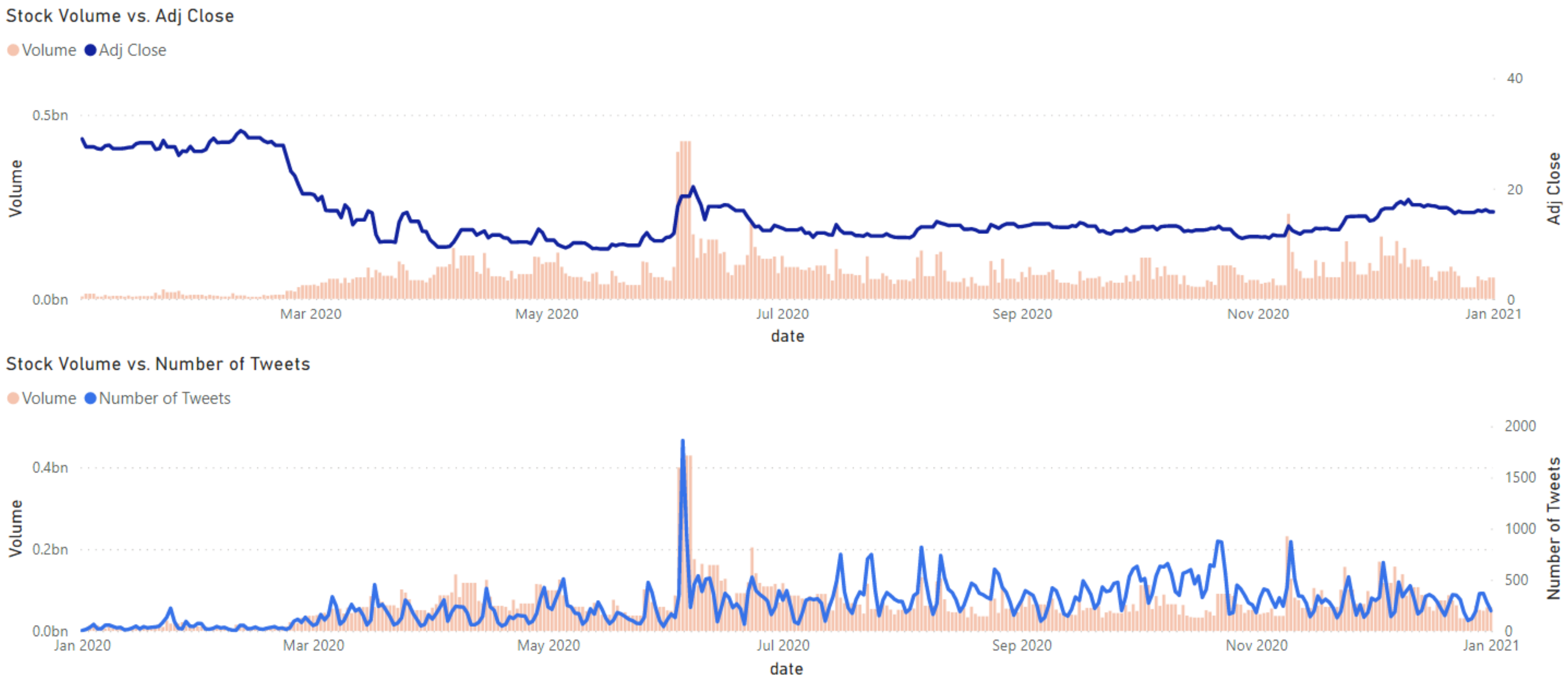


Figure 4.4: American Airlines – Trading volume vs. respectively, adjusted closing share price (top) and number of tweets. (bottom)

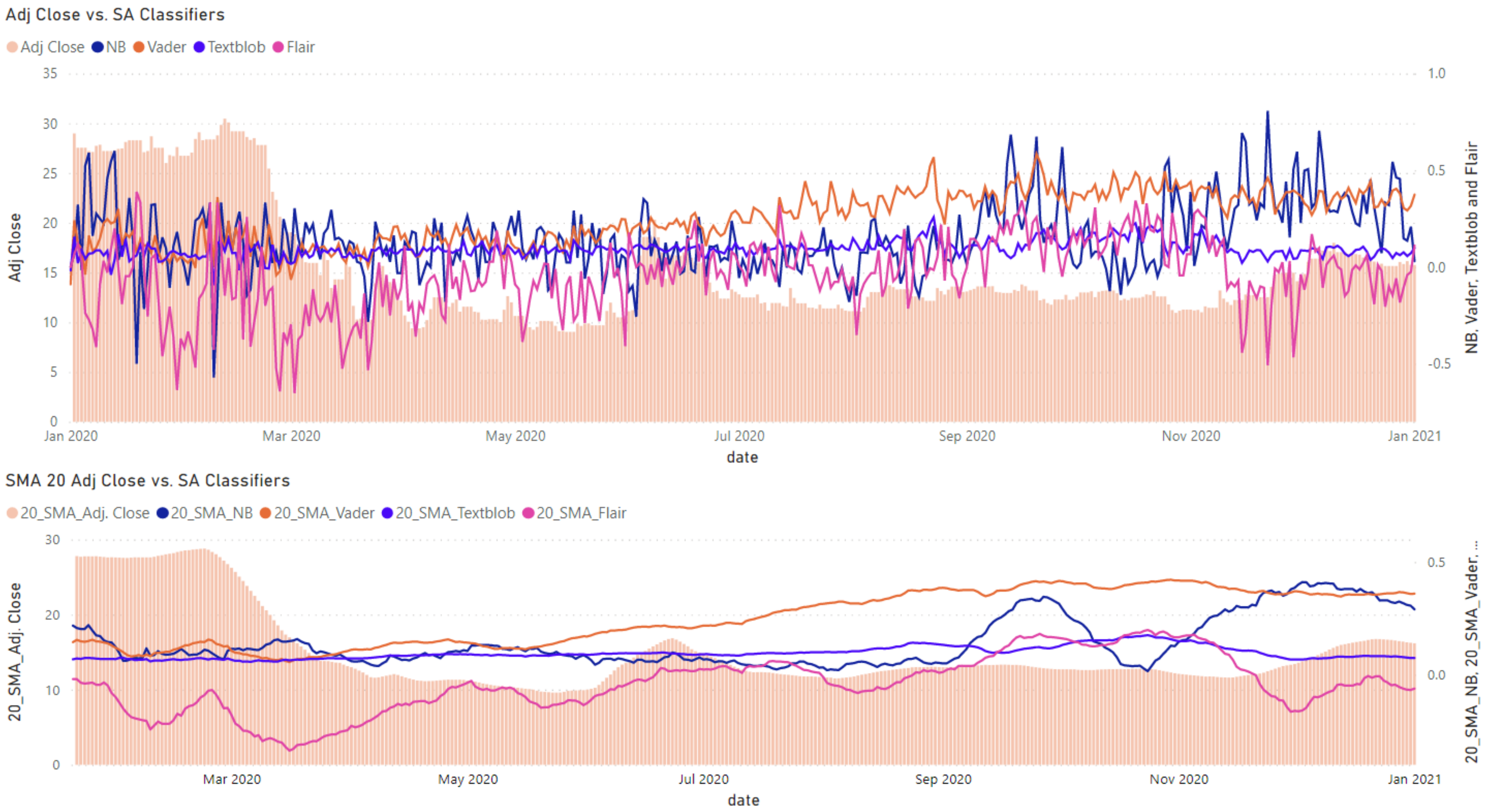


Figure 4.5: American Airlines – Sentiment indicators, with adjusted closing share price vs. respectively, the score of SA classifiers – Spark ML/NB, Textblob, VADER and Flair – (top) and the 20 simple moving average from the same classifiers. (bottom)

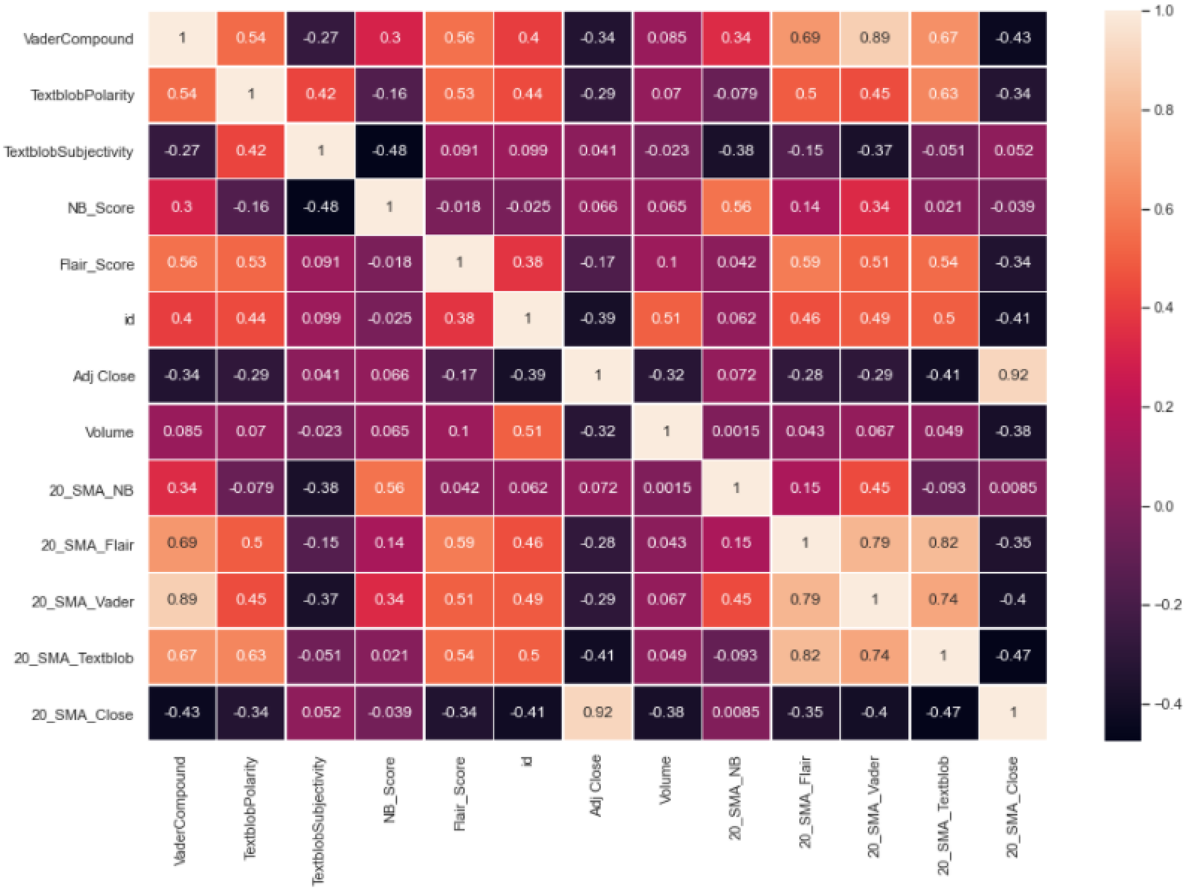


Figure 4.6: American Airlines – Heatmap showing correlations among various variables.

4.6 Summary

This Chapter has described all the experiments that have been carried out to validate our assumptions about the use of tweets in predicting stock price movements. That is, to be used as a useful stock market momentum indicator, depicting public sentiments about a particular company. [13]

Hence, in this case the analysis aims to provide practical insights in relation to two chosen companies, Facebook and American Airlines. But others could have been chosen instead.

To do so, the research work is built upon the framework set and implemented in the previous Chapter. It ranges from selecting companies, downloading both twitter and stock market data of concerning, processing data in the various stages of the pipeline, to applying classification models and, finally, visualizing and analysing the predictions and classifications obtained for the tweets.

[13] Notice that a stock market momentum indicator is always used alongside other indicators.

Chapter 5.

Conclusions

Contents

This chapter summarizes the main contributions of this dissertation and presents possible research paths for further development in the future.

[This page has been intentionally left blank]

Chapter 5
Conclusions

5.1 Introduction

Overall, this dissertation aims to establish a relationship between public sentiment scores and stock price movements, if any. To reach that purpose, we have designed and implemented a framework intended to collect, process, classify and, in the end, to evaluate if the sentiment expressed in data that was fetched from Twitter nexus to a particular company have impact in their stock price movement. Furthermore, this research also proposes to identify the most suitable sentiment techniques that should be used in such context, as well as finding out the most relevant variables for predicting stock price movements.

As described in Chapter 3, the general architecture of the solution implemented is composed of four different data processing blocks, as illustrated in Figure 3.1.

Within the data acquisition block, collecting data from Twitter using a feature called Cashtag ($) was, unexpectedly, a big challenge, that demanded a significant amount of effort. In part because there were few options available, all with their respective limitations. For example, in some cases there was a large percentage of tweets with spam.

In the end, the collected tweets were subject to a set of pre-processing operations, required to ensure consistency across the data set to be used further down the line. Once data was properly cleansed and structured, it followed the usage and testing of different SA models under consideration. Then the sentiment scores obtained were properly visualized and analysed, with the help of the visualization tool Microsoft Power BI.

The key evaluation efforts were presented in Chapter 4. As stated there, there are 10 selected companies but only two of them were considered in detail.

5.2 Main Contributions

Apart the solution that has been designed and implemented, which constitutes a major portion of the work undertaken, the experiments carried out and presented in Chapter 4 allow us to highlight a few points as far as the research questions initially set are concerned. Recall that, for the purpose of highlighting the research findings, there were two companies of interest for evaluation: Facebook and American Airlines.

Just to put into context, the rationale was to pick distinct companies, from different sectors and experiencing different realities so to broaden the spectrum of our analysis and comparison. Financially speaking, in the considered time frame and despite the COVID-19 pandemic, Facebook has experienced a great year, hitting all-time highs, contrasting with American Airlines that has experienced one of her worst years. The analysis of results from Facebook indicate a possible connection between the number of tweets and stock volume and stock close price, scoring a correlation of 0.38 and 0.42, respectively. Still on Facebook, another conclusion indicates that the moving averages of the sentiment scores have a higher correlation when compared to the daily scores. On the other hand, results from American Airlines indicate that the number of tweets does not influence the stock close prince but influence the stock volume, with a correlation of 0.51. The results scores from the SA classifiers did not showed any influence in the financial indicators.

Firstly, we can conclude that tweets related to a particular company may have an impact in their stock price performance. By this, we do not state that there is a direct relationship, yet data has showed that an increase in the volume of tweets leads to an oscillation in both stock price and trading volume. Actually, this can seen as a corroboration of why some trading platforms provide similar type of sentiment indicators, simply based on number of tweets but not having proper SA evaluation, which should include the processing of the content of every tweet.

Secondly, the data analysis carried out in relation to the two selected companies shows that using moving averages of sentiment scores makes the analysis clearer and more insightful. This is particular useful when measuring the strength or weakness of the price of a stock.

5.3 Conclusions and Future Work

Despite the objectives set for this dissertation were in general accomplished, there are a few aspects that could be addressed in the future. Indeed, it would be interesting to see this work further extended, with the goal of improving a sentiment indicator as a momentum indicator for the stock market.

Hence, we should point out the following aspects:

- To refine the quality of tweets extracted, for example looking at only verified accounts but making sure there are enough tweets to process, that is, to have a large amount of tweets.
- Increasing the time period of studying in order to obtain a wider range of tweets and events, so to overcame the specifics of a particular time frame like the COVID-19 pandemic we have considered in this research.
- Deployment of an end-to-end Apache Spark based solution, with improved ML algorithms, and therefore leveraging the distributed clustering computing and reducing the processing time.

Bibliography

[1] M. S. Ahmed, T. T. Aurpa, and M. M. Anwar. "Detecting sentiment dynamics and clusters of Twitter users for trending topics in COVID-19 pandemic." In: *PLoS ONE* 16.8 August (2021), pp. 1–20. issn: 19326203. doi: 10.1371/journal.pone.0253300.

[2] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf. "FLAIR: An easy-to-use framework for state-of-the-art NLP." In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Demonstrations Session* (2019), pp. 54–59.

[3] J. C. Amador Diaz Lopez, S. Collignon-Delmar, K. Benoit, and A. Matsuo. "Predicting the Brexit Vote by Tracking and Classifying Public Opinion Using Twitter Data." In: *Statistics, Politics and Policy* 8.1 (2017). issn: 2194-6299. doi: 10.1515/spp-2017-0006.

[4] V. Bonta, N. Kumaresh, and N. Janardhan. "A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis." In: *Asian Journal of Computer Science and Technology* 8.S2 (2019), pp. 1–6. issn: 2249-0701. doi: 10.51983/ajcst-2019.8.s2.2037.

[5] L. Breiman. "Machine Learning, Volume 45, Number 1 - SpringerLink." In: *Machine Learning* 45 (Oct. 2001), pp. 5–32. doi: 10.1023/A:1010933404324.

[6] M. D. Devika, C. Sunitha, and A. Ganesh. "Sentiment Analysis: A Comparative Study on Different Approaches." In: *Procedia Computer Science* 87 (2016), pp. 44–49. issn: 18770509. doi: 10.1016/j.procs.2016.05.124.

[7] G. Dubey, A. Rana, and J. Ranjan. "A research study of sentiment analysis and various techniques of sentiment classification." In: *International Journal of Data Analysis Techniques and Strategies* 8.2 (2016), pp. 122–142. issn: 17558069. doi: 10.1504/IJDATS.2016.077485.

[8] S. Elbagir and J. Yang. "Sentiment Analysis on Twitter with Python's Natural Language Toolkit and VADER Sentiment Analyzer." In: *Lecture Notes in Engineering and Computer Science*. Vol. 2239. 2020, pp. 63–80. isbn: 9789881404855. doi: 10.1142/9789811215094_0005.

[9] M. A. Fauzi. "Random forest approach fo sentiment analysis in Indonesian language." In: *Indonesian Journal of Electrical Engineering and Computer Science* 12.1 (2018), pp. 46–50. issn: 25024760. doi: 10.11591/ijeecs.v12.i1.pp46-50.

[10] A. Gandomi and M. Haider. "Beyond the hype: Big data concepts, methods, and analytics." In: *International Journal of Information Management* 35.2 (2015), pp. 137–144. issn: 02684012. doi: 10.1016/j.ijinfomgt.2014.10.007.

[11] B. Heredia, J. Prusa, and T. Khoshgoftaar. "Exploring the Effectiveness of Twitter at Polling the United States 2016 Presidential Election." In: *Proceedings - 2017 IEEE 3rd International Conference on Collaboration and Internet Computing, CIC 2017* 2017-January (2017), pp. 283–290. DOI: 10.1109/CIC.2017.00045.

[12] E Hutto, C.J. and Gilbert. "VADER: A Parsimonious Rule-based Model for." In: *Eighth International AAAI Conference on Weblogs and Social Media* (2014), p. 18. arXiv: nan.

[13] F. Iqbal, J. M. Hashmi, B. C. Fung, R. Batool, A. M. Khattak, S. Aleem, and P. C. Hung. "A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction." In: *IEEE Access* 7 (2019), pp. 14637–14652. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2892852.

[14] A. Joshi, P. Bhattacharyya, and S. Ahire. *Sentiment Resources: Lexicons and Datasets*. 2017, pp. 85–106. ISBN: 9783319553924. DOI: 10.1007/978-3-319-55394-8_5.

[15] N. Joshi and S. Itkat. "A Survey on Feature Level Sentiment Analysis." In: *International Journal of Computer Science and Information Technologies* 5.4 (2014), pp. 5422–5425. ISSN: 00274909.

[16] S. Kamath S., A. Bagalkotkar, A. Khandelwal, S. Pandey, and K. Poornima. "Sentiment analysis based approaches for understanding user context in Web content." In: *Proceedings - 2013 International Conference on Communication Systems and Network Technologies, CSNT 2013* May 2015 (2013), pp. 607–611. DOI: 10.1109/CSNT.2013.130.

[17] A. Kumar and A. Jaiswal. "Systematic literature review of sentiment analysis on Twitter using soft computing techniques." In: *Concurrency and Computation: Practice and Experience*. Vol. 32. 1. 2020, pp. 1–29. DOI: 10.1002/cpe.5107.

[18] A. Kumar and T. Sebastian. "Sentiment Analysis: A Perspective on its Past, Present and Future." In: *International Journal of Intelligent Systems and Applications* 4 (Sept. 2012). DOI: 10.5815/ijisa.2012.10.01.

[19] B. Liu. "Sentiment analysis and opinion mining." In: *Synthesis Lectures on Human Language Technologies* 5.1 (2012), pp. 1–184. ISSN: 19474040. DOI: 10.2200/S00416ED1V01Y201204HLT016.

[20] B. G. Malkiel. "Behavioral Finance." In: *A Random Walk Down Wall Street*. W. W. Norton & Company, 2019. Chap. 10.

[21] W. Medhat, A. Hassan, and H. Korashy. "Sentiment analysis algorithms and applications: A survey." In: *Ain Shams Engineering Journal* 5.4 (2014), pp. 1093–1113. ISSN: 20904479. DOI: 10.1016/j.asej.2014.04.011.

[22] A. Mudinas and D. Zhang. "Adaptive Sentiment Analysis *6mm." In: December (2018).

[23] N. Naw. "Twitter Sentiment Analysis Using Support Vector Machine and K-NN Classifiers." In: *International Journal of Scientific and Research Publications (IJSRP)* 8.10 (2018), pp. 407–411. DOI: 10.29322/ijsrp.8.10.2018.p8252.

[24] T. M. Nisar and M. Yeung. "Twitter as a tool for forecasting stock market movements: A short-window event study." In: *Journal of Finance and Data Science* 4.2 (2018), pp. 101–119. ISSN: 24059188. DOI: 10.1016/j.jfds.2017.11.002.

[25] O. Of, E. Of, and O. Of. “Sentiment Analysis : Text P Re -Processing , Reader Views.” In: *Pre-processing* (2015).

[26] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi. “Sentiment analysis of Twitter data for predicting stock market movements.” In: *International Conference on Signal Processing, Communication, Power and Embedded System, SCOPES 2016 - Proceedings* (2017), pp. 1345–1350. DOI: 10.1109/SCOPES.2016.7955659. arXiv: 1610.09225.

[27] S. Prof, R. Fuentecilla, and M. Ferreira. “Detecting Popularity and Innovation on Twitter to Find the Best stocks in SP500 João da Silva Franco Thesis to obtain the Master of Science Degree in Electrical and Computer Engineering October 2018.” In: October (2018).

[28] S. Rani and P. Kumar. “A sentiment analysis system for social media using machine learning techniques: Social enablement.” In: *Digit. Scholarsh. Humanit.* 34 (2019), pp. 569–581.

[29] R. Rodrigues, C. Camilo-Junior, and T. Couto. “A Taxonomy for Sentiment Analysis Field.” In: *International Journal of Web Information Systems* 14 (2018), p. 0. DOI: 10.1108/IJWIS-07-2017-0048.

[30] D. Sudarsa, S. Kumar.P, and L Jagajeevan Rao. “Sentiment Analysis for Social Networks Using Machine Learning Techniques.” In: *International Journal of Engineering & Technology* 7.2.32 (2018), p. 473. DOI: 10.14419/ijet.v7i2.32.16271.

[31] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica. “Apache Spark: a unified engine for big data processing.” In: *Commun. ACM* 59.11 (2016), pp. 56–65. DOI: 10.1145/2934664.

[This page has been intentionally left blank]

Sentiment Analysis in the Stock Market Based on Twitter Data

José Sacramento