Using Naïve Bayes and Genetic Algorithms to Find Influential Twitter Users to Forecast the S&P 500

Daniel Jorge Martins Sousa

Thesis to obtain the Master of Science Degree in Electrical and Computer Engineering

Supervisors: Prof. Rui Fuentecilla Maia Ferreira Neves
Prof. Nuno Cavaco Gomes Horta

Examination Committee
Chairperson: Prof. António Manuel Raminhos Cordeiro Grilo
Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves
Members of Committee: Prof. Nuno Filipe Valentim Roma

May 2017
To my grandmother and my girlfriend, for all the support and guidance.
Acknowledgements

Firstly, I would like to thank my supervisors Professor Rui Neves and Professor Nuno Horta for all the support and time dedicated. Their passion is inspirational and the dedication they put in each student is outstanding and helped me a lot.

In addition, I would like to thank my mother, father and brother for all the support and specially my grandmother, Avó São, for providing me the education and conditions that allowed me to achieve my degree. I will never be able to thank you enough for that.

Furthermore, I would like to thank my girlfriend, Soraia Picoito, for celebrating with me at each moment of success and for all the motivation and support given when things did not go as well. Her understanding and help since the beginning of my degree, were key to the course completion and for that, I thank her.

Finally, I would also like to thank all my great companions in my academic journey at Instituto Superior Técnico, specifically: António Mendes, Hugo Pereira, Jessy Neves, Miguel Monteiro, Miguel Rodrigues, Pedro Figueiredo, Rodolfo Carapau and Rúben Borralho. Despite being great friends, they have taught me so much that I hope to never lose touch with any of them.
Abstract

Information available on Twitter about stock market is increasing. Through tweets, users express their predictions or opinions about companies or events. These tweets can influence the behavior of those who read them. With the cashtags creation, this information has become easier to identify and therefore to use. The main objective of this work is use this information to forecast stock market. For that purpose, it is suggested to characterize each user through several numerical features related to his Twitter account. A Genetic Algorithm is used to optimize and return values that best describe an influential user in the stock market. To check the impact of each user, the tweets sentiment is analyzed and compared to the change in the stock price of the mentioned company. In order to make this identification, a classifier was created using Naïve Bayes, with an accuracy of 60.65%. This classifier, when applied to a tweet already pre-processed and filtered by the system, returns whether the tweet is positive, negative or neutral. Based on these classifications an investment is made. The results indicate that the characteristics which best describe an influential user are the account time and the account activity, followed by the number of likes and retweets obtained by tweet and the low tendency of the user to talk to other users using tweets. The results were very interesting, since in the best case, a profit of 14.7% was obtained for a test period of 6 months.

Keywords

Twitter; Influent Users; Genetic Algorithm; Naïve Bayes; Stock Market.
Resumo

A informação disponível no Twitter sobre mercados financeiros é cada vez maior. Os utilizadores através dos tweets exprimem as suas previsões ou opiniões acerca de empresas ou acontecimentos. Estes tweets podem influenciar o comportamento de quem os lê. Com a criação dos cashtags, esta informação tornou-se mais fácil de identificar e portanto, de utilizar. Este trabalho tem como principal objetivo a utilização desta informação para prever o mercado de ações. Para tal, propõe-se a caracterização de cada utilizador através de vários parâmetros numéricos relacionados com a sua conta de Twitter. Utiliza-se um Algoritmo Genético para fazer a sua otimização e retornar os valores que melhor descrevem um utilizador influente no mercado de ações. Para verificar o impacto de cada utilizador, é analisado o sentimento presente nos seus tweets e comparada com a variação do preço das ações da empresa mencionada. Para fazer esta identificação, foi criado um classificador utilizando Naïve Bayes com uma precisão de 60.65%. Este ao ser aplicado a um tweet já pré-processado e filtrado pelo sistema, retorna se este é positivo, negativo ou neutro. Com base nestas classificações é feito um investimento. Os resultados referem que as características que melhor descrevem um utilizador influente são o tempo e a atividade da conta, seguidos pelo número de likes e retweets obtidos por tweet e da pouca tendência do utilizador estabelecer conversas via tweets. Os resultados foram bastante interessantes, uma vez que no melhor caso foi obtido um lucro de 14.7% para um período de teste de 6 meses.

Palavras-chave

Twitter; Utilizadores Influences; Algoritmo Genético; Naïve Bayes; Mercado das Ações.
Table of Contents

Acknowledgements ......................................................................................... v
Abstract ........................................................................................................... vii
Resumo ........................................................................................................... viii
Table of Contents ........................................................................................... ix
List of Figures ................................................................................................ xii
List of Tables .................................................................................................. xiv
List of Acronyms ............................................................................................. xv
List of Software ............................................................................................. xvi

1 Introduction ................................................................................................. 1
  1.1 Overview and Motivation ......................................................................... 2
  1.2 Work’s Purpose ..................................................................................... 3
  1.3 Main Contributions ............................................................................... 4
  1.4 Document Structure ............................................................................. 4

2 Related Work ............................................................................................... 5
  2.1 Twitter Basic Concepts ......................................................................... 6
  2.2 Twitter as a Forecasting Tool ................................................................. 7
  2.3 Tweet Content Analysis ........................................................................ 8
    2.3.1 Data Pre-processing ....................................................................... 8
    2.3.2 Sentiment Analysis ..................................................................... 9
  2.4 Influential Users on Twitter ................................................................. 11
  2.5 Stock Prediction Using Twitter ............................................................ 14
  2.6 Naïve Bayes Classifier .......................................................................... 15
  2.7 Genetic Algorithms ............................................................................... 16
    2.7.1 Representation of Individuals ..................................................... 17
    2.7.2 Initial Generation ....................................................................... 18
    2.7.3 Evaluation .................................................................................. 18
2.7.4 Selection .............................................................................................................. 18
2.7.5 Crossover ............................................................................................................ 18
2.7.6 Mutation ............................................................................................................. 19
2.8 Chi-square Test .................................................................................................... 20
2.9 Synthesis ............................................................................................................. 20
3 Methodology and Architecture ................................................................................. 24
3.1 System Overview .................................................................................................. 25
3.2 Architecture .......................................................................................................... 25
3.3 Tweets Data Module ............................................................................................. 28
3.4 User Details Data Module .................................................................................... 29
3.5 Financial Information Module ............................................................................... 31
3.6 Content Analysis Module ...................................................................................... 32
3.6.1 Data Pre-processing ........................................................................................... 32
3.6.2 Sentiment Analysis ............................................................................................ 34
3.7 User Features Module .......................................................................................... 39
3.7.1 User Features .................................................................................................... 39
3.7.2 Genetic Algorithm ............................................................................................. 42
3.8 Investment Strategy Module .................................................................................. 46
3.8.1 Long Strategies .................................................................................................. 46
3.8.2 Short Strategies .................................................................................................. 47
3.9 Synthesis ............................................................................................................. 48
4 System Validation .................................................................................................... 49
4.1 Data Collection ...................................................................................................... 50
4.1.1 Tweets Collection .............................................................................................. 50
4.1.2 User Details Collection ..................................................................................... 50
4.1.3 Financial Information Collection ....................................................................... 51
4.1.4 Classifier Training Tweets Collection ............................................................... 51
4.2 Simulation Environment ......................................................................................... 52
4.3 Evaluation Metric .................................................................................................. 52
4.4 Train and Test Sets ............................................................................................... 54
4.5 Case Study I – Simple approach ......................................................................... 55
4.6 Case Study II – Implementation of GA ................................................................. 59
4.7 Case Study III – Improved GA using statistics .................................................... 65
4.8 Synthesis ............................................................................................................. 69
5 Conclusions ............................................................................................................ 70
References

75
List of Figures

Figure 1.1. Number of monthly active Twitter users worldwide from 1st quarter 2010 to 4th quarter 2016 (in millions) [1]. .......................................................... 2

Figure 2.1. Announcement of hashtags by Twitter. .......................................................... 6

Figure 2.2. Comparison between Twitter's Streaming API and Twitter's Firehose for tweets collected between December 14th, 2011 and January 10th, 2012 [2]. .............. 7

Figure 2.3. Spam tweet example .................................................................................... 8

Figure 2.4. Output tokens of a tokenization process by word example [4]. .................... 9

Figure 2.5. Simple network scheme with a passive user (in red) preventing contents from spreading over the network. Only users in blue would have access to the content (adapted from [32]). .................................................. 12

Figure 2.6. Venn diagram of the top-100 influential users across the three measures, normalized to 100% [35]. ...................................................................................... 13

Figure 2.7. GA flowchart ............................................................................................. 17

Figure 2.8. Crossover diagram ...................................................................................... 19

Figure 2.9. Mutation diagram ....................................................................................... 19

Figure 3.1. Layered architecture of the system. ............................................................ 26

Figure 3.2. Detailed architecture with the modules present in the system ..................... 27

Figure 3.3. Robot based on GOT pseudocode. ............................................................. 29

Figure 3.4. Spam tweet without any kind of useful content ....................................... 33

Figure 3.5. Example of a tweet with a company cashtag that not gives any information about it. .............................................................................................. 33

Figure 3.6. Time elapsed by the classifier identifying the sentiment polarity of the tweet using different percentage amounts of words. ................................................ 36

Figure 3.7. Accuracy of the classifier identifying the sentiment polarity of the tweet using different percentage amounts of words. .................................................. 37

Figure 3.8. Accuracy achieved by the 3% top words classifier in the dataset classification for each neutral boundary tested value (blue) and accuracy achieved in the same dataset classification without taking into account the neutral sentiment (red). ........ 38

Figure 3.9. Content Analysis diagram. ........................................................................ 39

Figure 3.10. Representation of the ranks for the various features, where each rank represents a set of users. ..................................................................................... 43

Figure 3.11. Translation process example of each rank into the range of values that they correspond to. ..................................................................................... 43

Figure 3.12. Double Point Crossover example .............................................................. 46

Figure 4.1. S&P 500 during the simulation period (July 1 to December 31, 2016). ............ 53

Figure 4.2. Sliding window diagram for an example period of 9 weeks. ......................... 54

Figure 4.3. Daily ROI of S&P 500 and Profit, during the simulation period, for "Most popular companies" long strategy and "Influent users" short strategy, first case study. 56

Figure 4.4. Daily ROI of S&P 500 and Profit, during the simulation period, for "Most popular companies" long strategy and "After a certain number of days" short strategy, first case study. ................................. 56

Figure 4.5. Daily ROI of S&P 500 and Profit, during the simulation period, for "Most popular companies" long strategy and "Crossing the average of the last days" short strategy, first case study. ................................. 57
Figure 4.6. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Influent users” short strategy, first case study.

Figure 4.7. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “After a certain number of days” short strategy, first case study.

Figure 4.8. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Crossing the average of the last days” short strategy, first case study.

Figure 4.9. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Influent users” short strategy, second case study.

Figure 4.10. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “After a certain number of days” short strategy, second case study.

Figure 4.11. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Crossing the average of the last days” short strategy, second case study.

Figure 4.12. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Influent users” short strategy, second case study.

Figure 4.13. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “After a certain number of days” short strategy, second case study.

Figure 4.14. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Crossing the average of the last days” short strategy, second case study.

Figure 4.15. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Influent users” short strategy, third case study.

Figure 4.16. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “After a certain number of days” short strategy, third case study.

Figure 4.17. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Crossing the average of the last days” short strategy, third case study.

Figure 4.18. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Influent users” short strategy, third case study.

Figure 4.19. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “After a certain number of days” short strategy, third case study.

Figure 4.20. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Crossing the average of the last days” short strategy, third case study.

Figure 5.1. Best results strategy. “Influent users” - “Crossing the average of the last days” investment strategy combination for the last 5 days, applied to the third case study.
List of Tables

Table 2.1. Comparison between 8 popular sentimental analysis methods [21]................................. 10
Table 2.2. Parallelism between natural evolution and problem solving in evolution computation (adapted from [46])............................................................... 16
Table 2.3. Synthesis about Twitter Basic Concepts.............................................................................. 21
Table 2.4. Synthesis about Twitter as a Forecasting Tool................................................................. 21
Table 2.5. Synthesis about Tweet Content Analysis............................................................................ 21
Table 2.6. Synthesis about Influent Users on Twitter......................................................................... 22
Table 2.7. Synthesis about Stock Prediction Using Twitter.............................................................. 23
Table 3.1. User Details acquired and calculated by the system......................................................... 31
Table 3.2. Pre-processing tasks performed in the Content Analysis Module.................................... 34
Table 3.3. Emotion hashtags used as search terms in the collection of tweets to train the Naive Bayes Classifier................................................................. 35
Table 3.4. Details summary about the classifier used......................................................................... 38
Table 3.5. Brief description of each calculated feature used in the system....................................... 42
Table 4.1. Tweets collection results................................................................................................. 50
Table 4.2. User details collection results.......................................................................................... 51
Table 4.3. Tweets collection results for the classifier training........................................................... 51
Table 4.4. Start and end dates for train and test periods, of each sliding window............................ 55
Table 4.5. Features range values outputted by the GA for 5 out of the 11 features............................. 59
Table 4.6. Features range values outputted by the GA for the last 6 out of the 11 features.............. 60
Table 4.7. Selected features in each sliding window.......................................................................... 65
Table 5.1. ROI values for each case study and for each combination of investment strategies. 72
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-Separated Values</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GOT</td>
<td>Get Old Tweets</td>
</tr>
<tr>
<td>GPOMS</td>
<td>Google-Profile of Mood States</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>NASDAQ</td>
<td>National Association of Securities Dealers Automated Quotations</td>
</tr>
<tr>
<td>NB</td>
<td>Naïve Bayes</td>
</tr>
<tr>
<td>NLTK</td>
<td>Natural Language Toolkit</td>
</tr>
<tr>
<td>NYSE</td>
<td>New York Stock Exchange</td>
</tr>
<tr>
<td>ROI</td>
<td>Return of Investment</td>
</tr>
<tr>
<td>S&amp;P 500</td>
<td>Standard and Poor's 500</td>
</tr>
<tr>
<td>SOFNN</td>
<td>Self-Organizing Fuzzy Neural Network</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>UTC</td>
<td>Universal Time Coordinated</td>
</tr>
</tbody>
</table>
List of Software

Draw.io
- Online images, schemes and flowchart development

LiClipse 3.1.0
- Python Integrated Development Environment

FileZilla
- File Transfer Protocol client application

GitHub
- Version control repository and Internet hosting service

Mendeley Desktop
- Reference management tool

Microsoft Excel 2016
- Spreadsheet and graph application

Microsoft Word 2016
- Word processor

Putty
- SSH, TelNet and Rlogin client
Chapter 1
Introduction

In this chapter it is given an overview about the proposed topic for the thesis and the motivation for it. The purpose and structure of the document are described and the main goals are detailed. The main contributions to this scientific area are also provided.
1.1 Overview and Motivation

Due to the way transactions are made today in the financial market, making a transaction a fraction of
second later than intended may represent huge costs. This has made it clear that there is a need to
create and use tools that can access and analyze information as quickly as possible. With the technology
advancement, this idea became real but eventually came up with a new problem: how to do this analysis.
The solution may be through the use of extremely fast algorithms, capable of evaluating in different
conditions the most varied situations and make the best possible decisions.

However, to do it, it is necessary to take into account also a great diversity of factors that can influence
the decision success or failure. One of these factors may be the opinion of certain entities that may
somehow, have access to inside information, have impact in the stock market or simply, have a more
accurate predictive capability.

Nowadays social networks are one of the best places to collect opinions. These online networks have
been expanding increasingly and gaining tremendous importance in the everyday life of any ordinary
individual, whether to voice an opinion, read news or simply to have a conversation.

Created in 2006 and with 319 million users at the end of 2017 (see Figure 1.1), Twitter is one of the
examples of this type of online social network. As the other social networks, this one has its peculiarities
such as restricting the number of characters to a maximum of 140 per publication. These publications
are called tweets and may express feelings, opinions, moods or even irrelevant content. Tweets may
include hashtags or cashtags which are keywords for certain subjects, written in a row and without
spaces, and preceded by the symbol “#” and “$”, respectively. However, collect old tweets can be a
difficult task since Twitter has a lot of restrictions on them.

![Figure 1.1. Number of monthly active Twitter users worldwide from 1st quarter 2010 to 4th quarter 2016 (in millions) [1].](image)
This network also allows a user to follow and be followed by another one, without having to send a request for it, which is not the case in most other social networks. Therefore, a user can choose which users can put content in his feed, that is, a kind of screen that is being updated whenever a new tweet is published. All of these characteristics have made Twitter one of the largest and most powerful social networks today.

One of the topics that has always attracted much attention is the financial market behavior, which has consequently made it one of the most popular topics on Twitter, with all kinds of entities to issue and to be influenced by other opinions and information. In this way, access to this type of information and its real-time analysis can be crucial to make a good decision in the financial market.

Despite the high popularity of the subject and the huge amount of works in this area, it continues to be a topic of much discussion and with results that fall short of expectations.

1.2 Work’s Purpose

The main objective of this work is to develop a system capable of choosing among a group of Twitter users those with greater impact in the stock market, more specifically, in S&P 500. This is an American stock market index based on the market capitalizations of 500 large companies having common stock listed on the NYSE and NASDAQ.

The principle behind is that a tweet posted about a certain company will have an impact on its stock price, that is, a positive tweet opinion about that company will be spread across the community and will lead to an increase in its stock price while a negative, will result on a decrease.

It is intended to choose users bases on their Twitter account features, understanding if it is possible to find their optimal values that claim that a user is influential. For that purpose, it will be necessary to collect tweets to train and test the system. However, in order to have free access to those tweets, it will be necessary to develop a tool that collects this data in another way, since Twitter does not allow the collection of tweets posted more than a week ago.

These tweets will be processed, filtered and applied to a sentimental classifier to understand its opinion polarity. This sentiment classifier, created based on Naïve Bayes techniques, will be trained to classify each tweet as positive, neutral or negative.

Finally, a Genetic Algorithm will be implemented in order to choose the optimal values for each feature and even which features are positively related with an influential user.

This work presents a new approach in this area of data science, combining different other researches in other areas such as opinion mining, evolutionary computation and stock market analysis. The way tweets will be collected is completely innovator, fast and effective.
1.3 Main Contributions

The main contributions made in this work are:

- A tool capable of collecting tweets, given the keywords and the period of time, without the restrictions of the official API of Twitter, without time and date restrictions and without quantity limits;
- The implementation of a tweet sentiment classifier for ternary classification: positive, neutral and negative;
- The creation of techniques to identify Twitter users with an impact in the stock market;
- The use of a GA to optimize the techniques and select the best ones.

1.4 Document Structure

The present work is composed of 5 chapters:

- Chapter 1 – Introduction;
- Chapter 2 – Related Work;
- Chapter 3 – Methodology and Architecture;
- Chapter 4 – System Validation;
- Chapter 5 – Conclusions.

This first chapter presents an introduction and the main goals of the work as well as the main contributions and the structure of the document.

In the second chapter is described and discussed the previous work related to the most important topics of the system such as sentiment and content analysis of tweets, identification of Tweet users as influential and forecasting the stock market using Twitter. The algorithms and techniques used by the system are also detailed.

The Chapter 3 presents the system architecture and provides a detailed explanation about the data flow and the methodology used in the system.

The fourth chapter describes the way the evaluation was performed to the system and presents in detail each tested case study.

In the fifth and final chapter, an analysis of the complete work is made, the conclusions are drawn and the future work, which is thought to improve the scientific research in this area, is proposed.
Chapter 2

Related Work

This chapter provides an overview of the relevant work already develop in each important topic of the system. Firstly, it is provided a background on fundamental concepts of Twitter and the main challenges when using it as a forecasting tool. Then, it is described several approaches about tweet content analysis including the sentimental analysis. The relevant work detecting influent users on Twitter is also mentioned, as well as the different approaches to predict stock market variations using Twitter. It is presented the theory about Naïve Bayes classifiers, Genetic Algorithms and the probabilistic test chi-square.
2.1 Twitter Basic Concepts

Launched in July 2006, Twitter is a social network service where users can post their opinions restricted to 140 characters. Tweets are accessible by everyone, but just registered users can post them.

Twitter has not strictly privacy rules as Facebook or Instagram. A user just have to “follow” another one to get access to his contents, without being required an approval.

Another important feature of Twitter is “retweets”, which allows a user to share quickly another user’s tweet, making the information flow fast and efficiently. This is the great and most appreciated advantage of Twitter.

Users can mention other users too, using ‘@’ preceding their usernames in tweets or keep track of a certain topic by using ‘#’ preceding its name, which are commonly known as hashtags. In 2012, Twitter has announced via a tweet (see Figure 2.1), a new feature similar to hashtag called “cashtag”, which take users to the search results about the companies’ finances and stocks, that made them very used nowadays for tweets about stock market. To create a cashtag, user just need to put the “$” symbol before the topic name, which is usually the stock symbol of the company.

![Twitter](https://via.placeholder.com/150)

Now you can click on ticker symbols like $GE on twitter.com to see search results about stocks and companies

Figure 2.1. Announcement of cashtags by Twitter.

For developers, Twitter has released two different APIs for two different proposes. One of them, the Twitter Search API, allows developers to access to recent Tweets published in the last 7 days subject to a limit of 180 requests every 15 minutes, where each request can provide a maximum number of 100 tweets or information about 100 users. The other one is the Twitter Streaming API that works under the same limits of the first one but allows real-time tweets collection, with Twitter’s documentation for this API promising 1% of all data.

Another way to access Twitter data is using Twitter Firehose which guarantees 100% delivery of tweets that match the given search parameters but besides having to pay for the access, it is needed a great amount of disk space and computing power. In [2], the authors compared Twitter Streaming API and the payed Twitter Firehose, concluding that when the number of tweets increases, the Streaming API efficiency decreases as is showed in Figure 2.2. They suggest more specific parameter sets instead of search by keyword.
2.2 Twitter as a Forecasting Tool

Since Twitter was created, it has been used for many different proposes than just a social network. In [3], the authors proposed tracking the H1N1 flu pandemic by monitoring Twitter. They relied their study on the 24 weeks daily analysis of tweets in the United Kingdom, searching for symptom-related statements to turn into statistical data to be compared to the Health Protection Agency information. They obtained a statistically significant linear correlation greater than 95%. The authors mentioned that the general concept of their work could also be applied for learning tendencies in other types of contexts such as politics, finance and public opinion.

Another relevant study carried out in this area, aimed at measure public opinion about presidential candidates in the 2012 U.S.A. election. The study’s authors, in [4], defined keywords related to which candidate and collect tweets with them. Then, a pre-processing task was done to filter URLs, emoticons, phone numbers, HTML tags, twitter mentions, hashtags, numbers with fractions and decimals, repetition of symbols and Unicode characters. After this, to create a sentiment classifier, it was used the crowdsourcing Amazon Mechanical Turk, where was asked to anonymous people to classify each tweet using a simple interface developed by the authors. With this data, they created the statistical classifier based in a Naïve Bayes model on unigram features, being which feature result of the tokenization of the tweets. This classifier was used to identify the sentiment (positive, negative, neutral or unsure) of which tweet acquired and then measure the candidate’s popularity and its variation. The authors concluded that tweet volume is largely driven by campaign events and according to them the method is generic and could be easily adopted and extended to other domains.
In [5], authors tried to forecast box-office revenues for movies. They collect tweets related to 24 different movies from Twitter Search API and classified them using a sentiment classifier based on DynamicLMClassifier, which is a language model classifier that accepts training events of categorized character sequences. The training data to build this model was collected, like in [4], using Amazon Mechanical Turk. The authors concluded that the success of box-office revenues of a movie was directly related with the amount of tweets associated to the movie.

2.3 Tweet Content Analysis

2.3.1 Data Pre-processing

Data pre-processing is the first stage of the content analysis. Since Twitter created cashtag, becomes much easier get more specific content from a company. However, it is not certain that the content has the desired quality, because it is not guaranteed that all collected tweets be about the intended company. For example, A, CAT, MMM and GAS are stock symbols but could be used in tweets with another connotation [6]. To solve this problem, in [7], the authors used more keywords in the search query. For instance, to filter content about Yahoo!, besides cashtag (“$YHOO”), they used “#YHOO” and “#Yahoo”.

Another limitation is the great amount of spam tweets using a lot of well-known cashtags aiming to get public attention (Figure 2.3). These kind of tweets are posted generally by robot accounts. The authors said in [8], that tweets’ content could vary drastically between excellent and spam.

Figure 2.3. Spam tweet example.

Therefore, the best way is to deal with spam. In [9], it was used a FireFox add-on, Clean Tweets, removing tweets that contain three or more trending topics of Twitter and filtering tweets from users whose accounts have less than a day of duration.

After getting companies’ related tweets and eliminating the spamming ones, it is necessary a data treatment to apply it to a sentiment analysis. Most of this data contains abbreviations, slang, emoticons, hashtags, cashtags, URLs, special characters, stopwords or it may have also been written in a language
other than English.

The first step in almost all references is doing the tokenization, which is the process of splitting a tweet or a document into tokens that could be a word (Figure 2.4), an emoticon or even a set of words correlated.

<table>
<thead>
<tr>
<th>Tweet</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAAAAHH!!! RT @politico: Romney: Santorum's 'dirty tricks' could steal Michigan: <a href="http://t.co/q6s1Ym">http://t.co/q6s1Ym</a> #Miprimary #tcot #teaparty #GOP</td>
<td>WAAAAHH!!! RT @politico: Romney: Santorum’s ‘dirty tricks’ could steal Michigan: <a href="http://politico.co/wYUz7m">http://politico.co/wYUz7m</a> #Miprimary #tcot #teaparty #GOP</td>
</tr>
</tbody>
</table>

Figure 2.4. Output tokens of a tokenization process by word example [4].

There are dictionaries used to correct the spelling and slang such as WordNet [10], SpellCheck [11] and JSpell [12] but are only available online which cause some performance issues.

Twitter APIs provide a feature which allows developer to choose the language of tweets that he wants to acquire. However, in [13], the authors collected tweets with no language restrictions and used Google Translate to convert non English tweets into English. They also convert “lol” expression to “laughing out loud” and some negative expressions like “never” or “cannot” to “not”. Long sequences of repeated character were replaced by “||U||” and “||T||” respectively. In [3], the authors just removed stop words, and then applied Porter’s algorithm [14]. In [15], it was used Porter’s algorithm too, but in this case the authors removed punctuation, symbols, stopwords, single characters and mentions.

### 2.3.2 Sentiment Analysis

Sometimes referred to as opinion mining, sentiment analysis has become a trend on computer science research area. Microblogging websites like Twitter have become a source of real-time information about users’ opinions on a variety of topics. Complaining, discussing or expressing their opinions, users are generating a lot of data which has a great value for companies that for instance sell products that people use in daily life [16].

The biggest challenge is to detect and summarize an overall sentiment relied not on facts but on subjective impressions, and in Twitter case, these subjective impressions must be obtained from a set composed by texts with 140 characters maximum length. This unique characteristic of Twitter leads to new problems for current sentiment analysis methods, which originally focused on large opinionated corpora [16].

In [15], the authors proposed a sentimental analysis using neural networks after the data pre-processing described in section 2.3.1. It was used a set of 200 tweets, 100 positive and 100 negative. The output words (280 different words) of this pre-processing was used as training input, obtaining a two neurons output representing the probability of the tweet being positive or negative. This classifier obtained an accuracy of 74.15% when applied to a set of 100 tweets labelled positive and negative manually.
An algorithm called Convolutional Neural Networks was used in [17]. This algorithm is very similar to the ordinary Neural networks, however is more efficient to implement and vastly reduce the amount of parameters in the network. It was applied for two corpora: the Stanford Sentiment Treebank, which contains sentences from movie reviews and the Stanford Twitter Sentiment corpus, which contains Twitter messages. For the first corpus, the author’s approach achieves 85.7% accuracy in binary classification (positive/negative). For the second corpus, it achieves a sentiment prediction accuracy of 86.4% for the same binary options.

In [18], the authors demonstrated the effectiveness of context-based Neural Network model for this task. Most of the researchers have been reporting that Support Vector Machines has higher accuracy than other algorithms, but it also has limitations, like it was said in [19]. Trying to overcome some other algorithms techniques, the authors focused on the use of Neural Networks in sentiment classification and analysis. Their study suggested that Neural Networks implementations would result in improved classification, combining the best characteristics of neural network with fuzzy logic.

As it was mentioned in [20], Neural Networks have been attracted a small audience as a sentimental analysis approach. However, the authors mentioned that experiments indicated that Neural Networks produced superior results to SVM when tested on several data sets. They confirmed some potential limitations of both models, which have been not so much discussed in the literature of sentiment analysis, like the computational cost of Support Vector Machines at the running time and Neural Networks at the training time.

Another approach is use sentimental analysis methods already developed and available online, instead of training a model. In [21], the authors used this approach presenting a comparison (in Table 2.1) between 8 popular methods, however excluding two very popular: Profile of Mood States[22] and OpinionFinder [23].

<table>
<thead>
<tr>
<th>Metric</th>
<th>PANAS-t</th>
<th>Emoticons</th>
<th>SASSA</th>
<th>Sentic-Net</th>
<th>Senti-WordNet</th>
<th>Happiness</th>
<th>Senti-Strength</th>
<th>LJWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>0.614</td>
<td>0.856</td>
<td>0.648</td>
<td>0.562</td>
<td>0.601</td>
<td>0.571</td>
<td>0.767</td>
<td>0.153</td>
</tr>
<tr>
<td>Precision</td>
<td>0.741</td>
<td>0.867</td>
<td>0.667</td>
<td>0.934</td>
<td>0.786</td>
<td>0.945</td>
<td>0.780</td>
<td>0.846</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.677</td>
<td>0.817</td>
<td>0.649</td>
<td>0.590</td>
<td>0.643</td>
<td>0.639</td>
<td>0.815</td>
<td>0.675</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.632</td>
<td>0.846</td>
<td>0.627</td>
<td>0.658</td>
<td>0.646</td>
<td>0.665</td>
<td>0.765</td>
<td>0.689</td>
</tr>
</tbody>
</table>

Analyzing accuracy metric presented in Table 2.1, the best method is Emoticons but it is not so feasible to use as it looks due to the fact that there are not so much tweets containing emoticons. In [24], the authors collected a large set of tweets and concluded that only 7% of them contained emoticons.

Another method often used in sentiment analysis is Naïve Bayes. In [25], the authors proposed a sentiment detection in a Spanish tweets’ data set. Their approach was based on polarity classification, classifying tweets into one of six different categories: strong positive, positive, neutral, negative, strong negative or without a sentiment. They detected the tweet’s polarity searching for polarity words on it, resulting in an accuracy of 67%. These results were very satisfactory since the more categories are
used, the worse the accuracy.

A similar approach was used but for English tweets by Gamallo et al. in [26], implementing two different Naïve Bayes classifiers (baseline and binary). In addition of tweets being very small to be linguistically analyzed, the English tweets require greater care on its classification because they are based on the uncertain human subjectivity [27]. However, the authors tried to classify as positive, negative or neutral. The baseline classifier was trained and then used to classify tweets without any changes after the training. On the other hand, the binary classifier uses a polarity lexicon to train and then, to classify tweets as positive or negative, only finding polarity words in tweets. If no positive or negative word is found, the tweet is classified as neutral. The binary classifier obtained an accuracy of 80%, being the best of both.

In [28] the authors proposed a sentiment classification using a Naïve Bayes classifier preceded by a pre-processing of the tweets data set, where they used a more complex lexicon, obtaining a F-score of 59.26.

### 2.4 Influent Users on Twitter

Nowadays there are so much data flooding Internet that became harder to distinguish clearly its credibility. An intuitive way to solve this problem is trust in some entities such as an ordinary Twitter user or even corporate Twitter accounts that, in some way, are considered influent and authoritarian.

This approach might be relatively intuitive but the challenge is which criterions should be used to differentiate. In [29], influence on Twitter was defined as “the potential of an action of a user to initiate a further action by another user”. The authors explored Twitter as an information broadcast platform reaching some interesting findings such as that celebrities with a large number of followers tend to feed more Twitter conversations (using Twitter mention feature) than to provide content that other users want to retweet. They also concluded that regardless of the number of followers, Twitter accounts associated with well-known media sources have a huge tendency to influence their followers to retweet their content. Despite this, it was mentioned that an independent user could even be more influent than a popular media source.

This article also mentioned that one of the most important attributes contributing to the users’ influence is the ability to get their content forwarded as far as possible in the users’ network. To achieve a good coverage, it is necessary that user’s followers are also active users, as well as the followers of these and so on.

This task is more complicated than it seems due to the fact that most of Twitter users act like passive. This sentence is supported by [30], where the authors developed a graph based algorithm which despite of being computationally infeasible for near real-time scenarios (as explained in [31]), shows that the
The correlation between popularity and influence is weaker than expected. The authors also analyzed passivity for a 22 million dataset of tweets collected using Twitter Search API during 300 hours. This process started on 10 Sep 2009 and were collect all tweets containing the string “http”. They concluded that an average Twitter user retweets only 1 in 318 tweets and the small number of the most active users play an important role in spreading the information unlike passive users.

A passive user is a user who absorbs content rather than shares it, preventing it from spreading over the network. Figure 2.5 illustrates this phenomenon, where the user in red represents a passive user blocking content from reaching users in black.

![Figure 2.5](image)

Figure 2.5. Simple network scheme with a passive user (in red) preventing contents from spreading over the network. Only users in blue would have access to the content (adapted from [32]).

H. Kwak et al. published a paper in 2010 where it was analyzed in detail, the way contents are spread in Twitter, aiming to identify who were the influential users in a network [9]. One of their findings was that approximately 22.1% of Twitter users have reciprocal relationship between them, which is a low value when compared to other social networking services. Previous studies have reported 68% on Flickr [33] and 84% on Yahoo! 360 [34]. Another important finding is that there was an account that deserved special attention which name is “mashable”, as was also mentioned in [29]. This account belongs to an entity which purpose is to spread news over social network services and as the authors proved, it is the most retweeted account in the test period, beating CNN and The New York Times accounts for example, despite having less followers.

The authors of [35] proposed another way of measuring user’s influence, using three different points of view: “indegree”, “retweets” and “mentions”. “Indegree” estimates the user’s influence just considering his followers’ number. “Retweets”, the second point of view, uses the number of tweets spread across the network where it is mentioned the user, aiming to found a relation between popularity or influence and the user’s capacity to provide spreadable content. The last one, uses mentions done to the user to check his capability to generate dialogues about him that could be interpreted as an influence factor.
For each of this three different measures, the authors created a set with the 100 most influential users, from a sample of 6 million users. Then, crossing the three top-100, the authors created a Venn diagram (Figure 2.6) where it is possible to conclude that only 7.1% of users are in the three top-100 at same time.

![Venn diagram of the top-100 influential users across the three measures, normalized to 100%](image)

In order to investigate how the three measures are correlated, the authors used Spearman’s rank correlation coefficients to measure the strength of the association between the three measures. They concluded that a user with a high indegree influence, i.e., with many followers, could not have a good capability to have his tweets well spread in the network, as well as could not be mentioned regularly by other users.

In this paper, it was also analyzed the ability of a user to cover several popular topics at same time, such as H1N1, Michael Jackson due to his death and Iran elections. The authors used Spearman’s rank correlation coefficients again for this purpose and applied it to the three top-100 of influential. They concluded that it is possible that some users can extend their influence to many topics.

In the same way, M. Furini and M. Montangero in [32] went further and identify the most influential users in several specific topics, developing an algorithm which they called “TRank” using three metrics, as in [35], based on followers, retweets and favorites. The first metric was the followers’ number of a user. The second one, retweets, was the retweets’ number that a user had at that time and the last one, favorites, counted the number of times the user had been added to a favorites list of other user. TRank started collecting all tweets from a specific cashtag for a certain period of time and then for each tweet, computed a user score for each metric mentioned above, selecting at the end the users with best scores.

Despite the positive results and the identification of some real influential accounts, these methods based in simple analysis are not very sensitive to the real topics’ authorities, giving more prominence to celebrities as stated in [31].

In that paper, [31], the authors proposed a different approach, leaving aside slow and heavy algorithms
based on graph analysis or rankings creation taking into account the amount of retweets, tweets, likes and other general Twitter details like that. This new approach used these details to compute more complex features. Some of the features were for example “Topical Signal” that estimated how much an author was involved with the topic irrespective of the types of tweets posted by her, “Retweet Impact” that indicated the impact of the content generated by the author or “Network Score”, that was the raw number of topically active users around the author. The probabilistic method was Gaussian Mixture Model [36], used for the purpose of create users clusters based on their features’ scores. The results obtained were quite satisfactory once the same tests were done manually by people and the results were very similar.

Qi He et al. in [37], presented an algorithm which they called “TwitterRank” that included the tweet topic analysis and took into account the link structures in users network with the objective of finding and identifying which users were influential in a certain subject (based on PageRank [38]). This algorithm used Latent Dirichlet Allocation method [39] to create a probability distribution on the various subjects for each user. After this, the authors generated a users graph for each subject and determined the subject and finally, calculated the weight of each user in the given subject. Despite the good indicators, this method becomes quite heavy because it was based on graph analysis.

2.5 Stock Prediction Using Twitter

In [40], the authors made a description of the most important works to date with a special focus on forecasting financial market indicators using Twitter. The authors associated certain words with two types of feelings: positive and negative. Then, they tested three different hypotheses of data analysis in order to compare it to the indicators. Two of the conclusions obtained were that users tend to post more tweets when they feel positive and that when there is some uncertainty in the stock market they tend to express feelings like fear or hope. However, when there is not a predominant feeling then it is because markets are likely to be stable. Zhang et al. also concluded that being aware of the general sentiment on Twitter would allow predicting short-term financial market behavior.

Another important work in this area was published by Bollen et al. in which the financial market was analyzed taking into account the public mood on Twitter [23]. In this paper, the authors used a tool called OpinionFinder, which analyzes the mood of a text, classifying it as positive or negative. They used also the Google-Profile of Mood States (GPOMS) that has the same objective of OpinionFinder but unlike it, this one measures public mood in terms of 6 dimensions. The objective was to collect and analyze each tweet from the point of view of its sentiment to after compare it to the financial market, using Granger’s Causality. Finally, they used the Self-Organizing Fuzzy Neural Network (SOFNN) algorithm to verify the possibility of doing predictions for the market, based on the feelings obtained through the tweets. It was verified a huge increase in forecasting capacity, reaching a success rate of 86.7%.

Goel et al. in [41] used a tweets dataset to find the public mood that they then used along with past
market values, to predict the behavior of the financial market [35]. The tool used, as in [19], ran the SOFNN algorithm, as well as OpinionFinder and Google-Profile of Mood States, for data collected from Yahoo! Finance. Using Granger's Causality, the authors found that calm and happiness sentiments had better predictive records. Then, Goel et al. implemented four algorithms to test the most accurate: Linear Regression, Logistic Regression, Support Vector Machine (SVM) and SOFNN. They found that the Logistic Regression and SVM methods had low accuracy because they presented the same percentage value in all sentiments, which did not allow any conclusion. The Linear Regression and SOFNN methods presented both good results.

2.6 Naïve Bayes Classifier

A Naïve Bayes classifier (NB) is a well-known probabilistic classifier that has been employed in many applications [42]. It is very simple to implement but very effective [43], which makes it the most commonly used classifier [20]. It was shown in [44] that NB under zero-one loss, despite of the independence assumption, performs well in many domains.

NB is based on Bayes' theorem, applying it with strong (naïve) independence assumptions between the features given the context of the class. The objective it to predict the probability that a given feature set belongs to a certain label (see 2.1).

\[
P(label|features) = \frac{P(label) \times P(features|label)}{P(features)}.
\]  

(2.1)

\(P(label)\) should be interpreted as the prior probability of a label and \(P(features|label)\) as the prior probability that a certain feature set is being classified as a label. \(P(features)\) is the prior probability that a given feature set occurred. Taking into account the assumption mentioned in the previous paragraph, which says that all features are independent, (2.1) could be reformulated to:

\[
P(label|features) = \frac{P(label) \times (P(f_1|label) \times ... \times P(f_n|label))}{P(features)},
\]  

(2.2)

with \(n\) being the number of features. This assumption makes NB computation very fast and efficient.

To use NB to classify is necessary to give it a prior training. In [45] the authors proposed using a sentence set previously labelled to identify the sentiment of a certain sentence. It is essential that the training set be correctly labelled and its size be big enough to allow the classifier to learn and get as many features as it can. However, the more features it learns, the more slower it gets. Thus, it is crucial to find an equilibrium point between the number of classifier features and the classifier accuracy.

For instance, if well trained, this model could pick a sentence and based on its words distribution,
compute the posterior probability of a class with simple tasks that do not require much computational resources.

### 2.7 Genetic Algorithms

Charles Darwin’s theory of evolution was the inspiration for Genetic Algorithm (GA). This algorithm relies on bio-inspired concepts such as mutation, crossover and natural selection.

Cells are the basic building blocks of all biological organisms. In each cells’ nucleus, there are structures called chromosomes which are composed by genes. Genes are made up of DNA and can reproduce. In this task, parent genes are combined using crossover to form a new chromosome with a new combination of genes.

The study of this biological and natural principles are in the origin of great human inventions. Humans mimic dolphins and bats to invent radar and fishes inspired the invention of submarines, for example. Species natural evolution can be considered a learning process.

As an example, let’s consider an environment filled with a certain species population, where each individual strives to survive and reproduce. The fitness of each individual will be determined by the environment, dictating how well they succeed. The best individuals will survival and reproduce while the worst will not, making the next generation stronger and more adapted to the environment than the previous one, optimizing the species fitness.

Thus, it is not surprising that so many scientists chosen as a source of inspiration the natural evolution, as it was mention in [46]. This powerful evolution is the fundamental idea of evolutionary algorithms, which are specially designed for solving complex optimization problems in a feasible time. The metaphor between evolutionary computing and natural evolution is represented in Table 2.2, where it is possible to understand the parallelism in a problem solving situation.

<table>
<thead>
<tr>
<th>Natural Evolution</th>
<th>Problem Solving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>Problem</td>
</tr>
<tr>
<td>Individual</td>
<td>Candidate Solution</td>
</tr>
<tr>
<td>Fitness</td>
<td>Quality</td>
</tr>
</tbody>
</table>

GA is basically a search algorithm for potential solutions inside a specific search space. The way this search is made, saves a lot of time since it does not need to test all the possible combinations. However,
it does not guarantee that the best solution is found because it may have a tendency to converge towards local optima, rather than the global optimum of the problem, as the authors said in [47]. This means that GA cannot sacrifice a short-term solution to gain in longer-term. Due to this, it is very important to maintain diversity in population in each cycle of the process.

The number of cycles should be defined taking into the account the problem to solve. On one hand, it should not be too high because it can become redundant and lose its purpose, but on the other hand, if it is too low, GA could not find a good solution.

GA flowchart is presented in Figure 2.7 and each step are explained below in more detail.

![GA flowchart](image)

**Figure 2.7. GA flowchart.**

### 2.7.1 Representation of Individuals

This first step is one of the most important since it is where the design of the problem is done. The output of the GA is a set of individuals, therefore it is important to choose well how they are constituted to solve the problem from the computer perspective.
Each individual, just like a chromosome, has a set of features (genes). Each individual represents a potential solution to the problem.

2.7.2 Initial Generation

A GA starts with a several number of individuals which is called population of individuals. In most cases, this initial set has its values randomly generated aiming to create a great diversity. However, if there are any prior knowledge about the desired solution, it should be considered in the formation of the initial population generation, as mentioned in [48].

The size of it is usually a constant value and depends on the study case. It does not change while the algorithm is running, aiming to increase the competition between individuals and the accuracy of the algorithm.

2.7.3 Evaluation

This process occurs in each algorithm cycle and evaluate the fitness value for each individual of the population. This value will be higher as the potential solution is better to solve the problem than the others. To compute this value it is necessary to define a function that evaluates each potential solution according to its capacity to solve the problem.

2.7.4 Selection

This operator selects individuals in the population for reproduction based on their fitness evaluation. The fitter the individual, the more times it is likely to be selected to reproduce and pass their genes to the next generation of individuals.

There are several selection methods discussed in [49] being the most popular “Elitism”, “Tournament” and “Rank Selection”.

2.7.5 Crossover

The crossover operator picks two individuals from the population, chosen by selection operator and combine them, swapping some values, creating new individuals in the population, as illustrated in Figure 2.8. After some iterations of the GA, every individual of the population has a high fitness and at that time, crossover plays an important role, combining those individuals and trying to generate a new better potential solution.

This method is inspired in a sexual reproduction phase, where is observed the recombination phenomenon between two distinct chromosomes.
2.7.6 Mutation

Mutation is a procedure that randomly picks a part of an individual and changes its genes values to generate a new individual, as presented in Figure 2.9. The basic idea of mutation is to bring some randomness to the population, creating new solutions inside the search space without compromising the progress already made by GA successive searches. Otherwise any combination of solutions could be created from the initial generation.

The mutation probability is defined as “rate of mutation” and generally has a small probability to occur. However, it is a powerful operator to achieve a good solution with GA.

This method is also inspired in reproduction. In that scope, mutations result from errors during DNA replication, changing the DNA sequence.
2.8 Chi-square Test

The chi-square test has the objective of calculating the probability that a sample distribution is due to chance [50].

There are two types of chi-square tests, both using the chi-square statistic and distribution. The first one intends to determine if a certain sample matches a population. The second one, which is used in this work and therefore is more detailed below in this section, compares two variables to verify if they are significantly related [51].

The chi-square output value is a single number that describes the difference between the train data and the results and can be calculated using (2.3).

\[ \chi^2 = \sum_{i=0}^{n} \frac{(O_i - E_i)^2}{E_i} , \]  

(2.3)

with \( n \) being the sample size, \( O \) the observed value and \( E \) the expected value.

A small output value means that the train data fits the results extremely well, that is, there is a relationship between them. On the other hand, a large value means that there is no relationship between the train data and the results.

2.9 Synthesis

From this chapter it is possible to conclude that using Twitter to forecast the stock market is feasible. However, there are a lot of different approaches that could be used in the various stages of this process.

After a thorough research, there were a lot of papers that have positively influenced the methodology that was used in this work. For example, tweets collection was done using cashtags which proved to be a good choice due to the quite satisfactory number of tweets collected. The tweets pre-processing approach used in this work was inspired too on the literature, as well as the sentiment model used, leading to good results in this stage.

However, there were methodologies that were not inspired by the literature. One of them was the way the tweets were collected. The proposed approaches imply a payout or are subject to the rate limits imposed by the Twitter API, that limits the number of possible collectible tweets. To solve this problem, based on an existing method, an innovative way of acquiring tweets (presented in the next chapter) has been developed, which allows to do this collection free of charge and without being subject to the rate limits.

Another methodology not totally inspired by the literature was the identification of influent users. In this process, the use of GA was proposed in order to optimize the users’ characteristics and choose their
best values. As genetic algorithms and evolutionary computation are increasingly becoming a trend, the aim of this choice is to test an approach that has not yet been used in a relevant way in this area.

A summarize of the literature are divided by main areas and presented in Table 2.3, Table 2.4, Table 2.5, Table 2.6 and Table 2.7.

Table 2.3. Synthesis about Twitter Basic Concepts.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Rate Limit</th>
<th>Free</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter Streaming API</td>
<td>1% to 40% of all data</td>
<td>Yes</td>
<td>Just in real-time</td>
</tr>
<tr>
<td>Twitter Search API</td>
<td>72k tweets per hour at most</td>
<td>Yes</td>
<td>400 keywords, 5k tweets per keyword, 7 days ago at the most</td>
</tr>
<tr>
<td>Twitter Firehose</td>
<td>None</td>
<td>No. Starts at 1000$ for 40 days [52]</td>
<td>Not free</td>
</tr>
</tbody>
</table>

Table 2.4. Synthesis about Twitter as a Forecasting Tool.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Approach to Filter Tweets</th>
<th>Dataset Size</th>
<th>Tools</th>
<th>Best Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>2010</td>
<td>Twitter geolocation feature</td>
<td>27M tweets</td>
<td>LASSO; Twitter Streaming API</td>
<td>Correlation greater than 95%</td>
</tr>
<tr>
<td>[4]</td>
<td>2012</td>
<td>Keywords and hashtags</td>
<td>17k tweets</td>
<td>Twitter Firehose</td>
<td>59% accuracy on Sentiment Analysis</td>
</tr>
<tr>
<td>[5]</td>
<td>2010</td>
<td>Keywords</td>
<td>3M tweets</td>
<td>Twitter Search API</td>
<td>98% accuracy on Sentiment Analysis</td>
</tr>
</tbody>
</table>

Table 2.5. Synthesis about Tweet Content Analysis.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Approach to filter tweets</th>
<th>Dataset size</th>
<th>Tools</th>
<th>Best Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6]</td>
<td>2012</td>
<td>Cashtags</td>
<td>-</td>
<td>Twitter Search API</td>
<td>-</td>
</tr>
<tr>
<td>[7]</td>
<td>2012</td>
<td>Hashtags and cashtags</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[13]</td>
<td>2011</td>
<td>-</td>
<td>12k tweets</td>
<td>Tree Kernels; Unigram; 100 Senti-features</td>
<td>75.39% accuracy on Sentiment Analysis</td>
</tr>
<tr>
<td>[15]</td>
<td>2015</td>
<td>Keywords</td>
<td>300 tweets</td>
<td>Neural Networks; Twitter Search API</td>
<td>74.15% accuracy on Sentiment Analysis</td>
</tr>
<tr>
<td>[17]</td>
<td>2014</td>
<td></td>
<td>107.5k tweets</td>
<td>Deep Convolutional Neural Networks</td>
<td>86.4% accuracy on Sentiment Analysis</td>
</tr>
</tbody>
</table>
### Table 2.5 (cont). Synthesis about Tweet Content Analysis.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Approach to filter tweets</th>
<th>Dataset size</th>
<th>Tools</th>
<th>Best Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>2016</td>
<td>Hashtags</td>
<td>18k tweets</td>
<td>Neural Networks; Twitter Streaming API</td>
<td>92.27% macro-F score</td>
</tr>
<tr>
<td>[21]</td>
<td>2015</td>
<td>-</td>
<td>1800M tweets</td>
<td>Emoticons Model; LIWC; SentiStrenght; SASA; PANAS-t; Happiness Index; SenticNet; SentiWordNet</td>
<td>81.7% accuracy on Sentiment Analysis</td>
</tr>
<tr>
<td>[24]</td>
<td>2013</td>
<td>-</td>
<td>1.1G tweets</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[25]</td>
<td>2013</td>
<td>-</td>
<td>7k tweets</td>
<td>Naive Bayes</td>
<td>67% accuracy on Sentiment Analysis</td>
</tr>
<tr>
<td>[26]</td>
<td>2014</td>
<td>-</td>
<td>11.5k tweets</td>
<td>Naive Bayes</td>
<td>63% F-score on Sentiment Analysis</td>
</tr>
<tr>
<td>[28]</td>
<td>2015</td>
<td>-</td>
<td>8k tweets</td>
<td>Naive Bayes</td>
<td>59.26% F-score on Sentiment Analysis</td>
</tr>
</tbody>
</table>

### Table 2.6. Synthesis about Influential Users on Twitter.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>User Features Tested</th>
<th>Approach to Filter Tweets</th>
<th>Dataset Size</th>
<th>Tools</th>
<th>Best Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>[29]</td>
<td>2009</td>
<td>Number of F followers, Number of Retweets</td>
<td>keywords</td>
<td>135k tweets</td>
<td>Twitter Search API</td>
<td>-</td>
</tr>
<tr>
<td>[30]</td>
<td>2011</td>
<td>Number of followers and Number of Retweets</td>
<td>URL</td>
<td>22M tweets</td>
<td>IP Algorithm; Twitter Search API</td>
<td>Number of Retweets</td>
</tr>
<tr>
<td>[9]</td>
<td>2010</td>
<td>Keywords</td>
<td>Clean Tweets Firefox add-on</td>
<td>106M tweets</td>
<td>Twitter Search API; PageRank</td>
<td>-</td>
</tr>
<tr>
<td>[35]</td>
<td>2010</td>
<td>Number of followers, Number of Retweets and Number of Mentions</td>
<td>keywords</td>
<td>1963M tweets</td>
<td>Twitter Streaming API</td>
<td>Number of Retweets Number of Mentions</td>
</tr>
<tr>
<td>[32]</td>
<td>2015</td>
<td>Number of Followers, Number of Retweets and Number of Favorites</td>
<td>hashtags</td>
<td>-</td>
<td>-</td>
<td>Number of Retweets Number of Favorites</td>
</tr>
</tbody>
</table>
Table 2.6 (cont). Synthesis about Influent Users on Twitter.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>User Features Tested</th>
<th>Approach to Filter Tweets</th>
<th>Dataset Size</th>
<th>Tools</th>
<th>Best Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>[31]</td>
<td>2011</td>
<td>17 features</td>
<td>keywords</td>
<td>90M tweets</td>
<td>Gaussian Mixture Model; Twitter Firehose</td>
<td>-</td>
</tr>
<tr>
<td>[37]</td>
<td>2010</td>
<td>FF Ratio</td>
<td>keywords</td>
<td>1M tweets</td>
<td>Twitter Streaming API; Twitter Search API; TwitterRank; PageRank; Indegree; PageRank</td>
<td>TwitterRank</td>
</tr>
</tbody>
</table>

Table 2.7. Synthesis about Stock Prediction Using Twitter.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Approach to filter tweets</th>
<th>Dataset size</th>
<th>Tools</th>
<th>Best Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>[40]</td>
<td>2011</td>
<td>-</td>
<td>2.8M tweets</td>
<td>Bag-of-Words Model</td>
<td>-</td>
</tr>
<tr>
<td>[23]</td>
<td>2011</td>
<td>-</td>
<td>10M tweets</td>
<td>OpinionFinder; Self Organizing Fuzzy Neural Networks; GPOMS</td>
<td>86.7% accuracy in Prediction</td>
</tr>
<tr>
<td>[41]</td>
<td>2012</td>
<td>keywords</td>
<td>476M tweets</td>
<td>Self Organizing Fuzzy Neural Networks; GPOMS</td>
<td>75.56% accuracy</td>
</tr>
</tbody>
</table>
Chapter 3

Methodology and Architecture

This chapter describes in detail the architecture of the system, as well as each module that compose it. It is presented all the modules to collect data from Twitter and information from Yahoo! Finance and the way it is done. It is also showed how the tweet sentiment analysis is done and the way GA is employed by the system to optimize features values. The investment strategies are also described.
### 3.1 System Overview

The main objective of this work is to identify users who may have an impact on the stock market in some way. The main idea is that a tweet made by an influential account about a certain subject or company may have impact on its stock price. Therefore, the goal is to identify users that can cause an increase in the stock price of a certain company just by posting a positive tweet about it and also can cause a decrease in its stock price, posting a negative tweet.

More specifically, it is intended to develop a system that is capable of performing opinion mining on companies' stocks. This process should be done automatically, monitoring identified users and new tweets about companies and interpreting its emotional content, in order to predict the stock price variation.

For that purpose, it was used Python to develop all the software. This language is becoming increasingly popular, it is easy to use, powerful, versatile, has a lot of open source packages available for free and it is very efficient to manage Big Data.

### 3.2 Architecture

To achieve the purpose of the systems, it is divided into different modules that individually handle each step. Despite of working individually, these modules are not isolated from each other, since they exchange data throughout the process.

These modules can be grouped into layers according to their purposes. In Figure 3.1 is represented the layered architecture of the system as well as two additional layers with the inputs and outputs.

The Data Sources Layer represent the external sources where the Data Collection and Processing Layer collects the data. These external sources are Twitter, in order to get tweets and user details, and Yahoo! Finance that is an online tool owned by Yahoo! that provides users financial news, data and commentaries.

The Data Collection and Processing Layer allows data from the Data Sources Layer to be collected, processed and stored.

Then there is the Data Analysis Layer which contains the modules that analyze and use data from the layer below, the Data Collection and Processing Layer, and provide it already processed by the algorithms used to the above layer, the Portfolio Management Layer.

In this last system layer, are made investment decisions based on the various strategies chosen by the user. The system provides several long and short strategies to be applied, with further details given in
section 3.8. Afterwards the results are presented to the user, as an output of the system, in the Output Data Layer.

In this outer layer the results are presented in graphical form, presenting a comparison between the user’s portfolio and the S&P 500 index. It is also presented the most influent users and the best features values.

Figure 3.1. Layered architecture of the system.

The schematic in Figure 3.1 is just a theoretical way to present modules grouped because, in fact, the data does not flow like this. In Figure 3.2 is illustrated in more detail, the data flow between the various modules of the system. It is presented below a brief description of each module.

The Tweets Data Module does the collection of tweets for the given time period and for the given set of keywords, which in this work, is a set of ticker symbols of the companies from the S&P 500. For each tweet, it is stored its text and several other details in the tweets database.

The User Details Module collects, for each author present in the tweets database, their account details, creating and storing the users database.
The Financial Information Module gather the relevant financial information about each company of S&P 500 and for the time period used in the collection of tweets, with a safety margin of 15-day against its start and end dates. This information is stored in the stocks database.

The Content Analysis Module does the tweets processing which is divided into two parts, being the first one the Data Pre-processing and the second one the Sentiment Analysis. In the first part, it is done the tweets analyzes, processing them to remove some words and to correct other. In the second part, each tweet is classified using a Naïve Bayes classifier and tagged as positive, negative or neutral, updating the tweets database.

The User Features Module uses the tweets database and the users database to calculate the users features that are explained in detail in section 3.7. Using these calculated features, this module is also...
responsible for the Genetic Algorithm that optimizes their values and identifies the influential users.

The Investment Strategy Module manages the portfolio investments, making short and long decisions about the stocks according to the specific rules of each investment strategy.

Finally, it is presented the system output, i.e. the investment results for the chosen strategy, the most influential users and the optimal values range of their features.

### 3.3 Tweets Data Module

This module, as mentioned above, is responsible for the tweets collection. Since there are a lot of tweets posted daily about the S&P 500 companies, it is vitally important that this process be done efficiently and effectively. Using Twitter APIs, the best choice would be the Twitter Search API, once Twitter Firehose is payed and the Twitter Streaming API just gives access to tweets that are being posted in real-time. However, as mentioned in section 2.1, this solution just provides tweets posted in the last 7 days, which is not enough data to train and test the system. To be enough, it would be necessary a weekly collection of data for a long period, which is unfeasible due to the time period defined for this work. In addition, the collection of tweets is subject to the Twitter rate limits which leads to large time constraints.

To solve this issue, it is used the Python package “GetOldTweets” (GOT) [53]. Twitter website has a search box in the top right corner of the page. This box allows to insert keywords, as “$AAPL”, and search for tweets using them and the more scroll down, the older tweets are presented. Twitter also allows to insert search operators such as “lang:en” that present tweets written in English, “until:2016-01-01” or “since:2016-01-01” presenting tweets posted until or since that date and even allows to combine all of them in a single query.

What GOT does is act like a browser and use that search box, inserting the keywords and scrolling down automatically to get the old tweets, all through calls to a JavaScript Object Notation (JSON) provider. So, applying this procedure for each company, using it stock symbol as cashtag, tweets about them are easily collected without time constraints. It is necessary to run it for each company because if all the cashtags were inserted at the same time in the search box, Twitter would interpret that the user wanted tweets with all the cashtags present in each tweet. Therefore, it has to be run 505 times since the S&P 500 is composed by 505 companies.

However, Twitter prevents tweets from being collected by this way through security mechanisms. Therefore, it was developed a robot based on GOT to deal with this constraint. This robot basically checks, after the tweets collection for each company by GOT, if there are tweets posted in more than 50% of the period days, since there are a huge volume of tweets posted daily about each company. If there is not, the robot closes the connection and tries again to the company that was processing when the error occurred, every 20 minutes. When it is able again to get the tweets properly, the process continues normally for the companies that still need to be processed.
The robot pseudocode is presented in Figure 3.3.

```plaintext
procedure robot based on GOT

for company in S&P500_companies:
    create connection
    tweets_set = run GOT(company)
    tweets_date_set = distinct dates in tweets_set

    if size(tweets_date_set) < 0.5*size(period):
        close connection

    while size(tweets_date_set) < 0.5*size(period):
        wait 20 minutes
        create connection
        tweets_set = run GOT(company)
        tweets_date_set = distinct dates in tweet_set
        close connection
    return tweets_set
```

Figure 3.3. Robot based on GOT pseudocode.

### 3.4 User Details Data Module

In addition to the collection of tweets, it is also relevant to collect information about the users that posted them, since this information is the basis of its features calculation. There are two types of details: the details that are calculated, analyzing the tweets content, and the details that are retrieved from the Twitter.

One of the details that are retrieved from the Twitter is the number of followings or in another words, the number of different Twitter accounts that a user is following to receive their content in his feed. In the opposite direction, is also acquired the number of followers, which is the number of different Twitter accounts that follow the user in order to receive his content in their feed.

Other important details that are also collected are the number of tweets posted so far by the user and the date when users created their account, which allows the calculation of the age of their accounts, for example.

To obtain these details from Twitter, it is used the Twitter Search API because the robot used in tweets collection in section 3.3, does not allows the collection of specific parameters of the users, such as the number of followers or the number of tweets ever done. Twitter Streaming API also does not make sense to use because it works in real-time and Twitter Firehose because it is payed.
Essentially, the system makes requests to the Twitter Search API in order to access the details with the required search term, which is the account username. Since the tweets are collected using the robot described in 3.3, it is possible to use Twitter Search API to collect the user details in parallel with it. Thus, it makes possible a more efficient data collection.

If for instance, the Tweets Data Module was running once every hour, collecting 18000 tweets each time, with each one posted by a different user, then the User Details Data Module would gather the details about the users in a few seconds, because the window rate limit would not be exceeded.

In fact, 18,000 requests is the limit value that allows an instant Twitter response for each 15 minute interval. However, this is an unrealistic scenario for the 505 companies of the S&P 500 index, that generally do not generate this amount of tweets related to them in a 15 minutes interval.

Additionally, there are also details that are calculated from the tweets content analyzes and are presented below. Some of them are calculated separately for each company and the others are calculated based on all the tweets in the dataset. This information is in the Table 3.1, where are presented all the user details. In this table, the Source means where the details are obtained, the Level indicates whether the details are calculated for each company separately or for all companies together and the Time Period represents the period for which the details are calculated or collected.

One of these details that needs to be calculated is the number of tweets about the company during the period and it is the total amount of tweets that the user posted about the company in the period defined by the system’s user. It is also a user detail the total number of tweets posted by him during the period.

Other user details are the number of likes received during the period and the total number of likes made by the user during the period.

The number of different cashtags used in the tweets is an important detail too, as the number of tweets that have been retweeted by other users and the number of different users who retweeted.

Related to mentions which were detailed in 2.1, there are also some important details such as the number of mentions done to the user, the number of mentions done by the user, the number of different users that mentioned the user and the number of different users mentioned by the user.

The last detail calculated is the number of conversational tweets done by the user. A conversational tweet is identified when the first word is a mention to another user account.

These details about each user who posted a tweet within the specified period are stored according to the date they were collected.
### Table 3.1. User Details acquired and calculated by the system.

<table>
<thead>
<tr>
<th>User Detail</th>
<th>Source</th>
<th>Level</th>
<th>Time Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of followings</td>
<td>Twitter API</td>
<td>General</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of followers</td>
<td>Twitter API</td>
<td>General</td>
<td>Ever</td>
</tr>
<tr>
<td>Total number of tweets ever done</td>
<td>Twitter API</td>
<td>General</td>
<td>Ever</td>
</tr>
<tr>
<td>Account creation date</td>
<td>Twitter API</td>
<td>General</td>
<td>Ever</td>
</tr>
<tr>
<td>Number of tweets about the company during the period</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of tweets during the period</td>
<td>Calculated</td>
<td>General</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of likes received during the period</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Total number of likes ever made</td>
<td>Twitter API</td>
<td>General</td>
<td>Ever</td>
</tr>
<tr>
<td>Number of different hashtags used</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of different users who retweeted the user</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of tweets that have been retweeted by other users</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of mentions done to the user</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of mentions done by the user</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of different users that mentioned the user</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of different users mentioned by the user</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
<tr>
<td>Number of conversation tweets done by the user</td>
<td>Calculated</td>
<td>Individual company</td>
<td>Limited</td>
</tr>
</tbody>
</table>

### 3.5 Financial Information Module

The responsibilities of this module are to obtain and process the information about the S&P 500 index companies to, later in section 3.6, be identified which users have had impact on the stock variation. This information includes the companies’ stocks open and close prices for each day of the time period defined. It is also collected the S&P 500 index value for that time period from the Yahoo! Finance python package [54]. This package provides a way for developers to access to information about the stock market, using an Internet connection and without storing any data.

There are two ways of verifying the variation of stock prices between two certain days: the opening price of both days or the adjusted closing price of both days. The adjusted closing price of a stock has always the same value of the closing price, except on any given trading day that has been amended to include any distributions and corporate actions.

The stock price variation in percentage terms of a company C in date $d_1$, between that date and a date $d_2$, for each way, is calculated through the formula (3.1) for adjusted closing price and through the formula (3.2) for the opening price.
%Variation_{\text{adjClosingPrice}}(C,d_1,d_2) = \frac{\text{AdjustedClosingPrice}(C,d_1) - \text{AdjustedClosingPrice}(C,d_2)}{\text{AdjustedClosingPrice}(C,d_2)} \times 100. \quad (3.1)

%Variation_{\text{OpeningPrice}}(C,d_1,d_2) = \frac{\text{OpeningPrice}(C,d_1) - \text{OpeningPrice}(C,d_2)}{\text{OpeningPrice}(C,d_2)} \times 100. \quad (3.2)

3.6 Content Analysis Module

The purpose of this module is to analyze the sentiment polarity of tweets. It is decomposed into two phases: Data Pre-processing and Sentiment Analysis.

Data Pre-processing is the first phase and is where the tweets are processed, in order to remove some content that would only represent noise and that could lead to less accuracy in the sentimental analysis task. The objective is to make this data suitable for a reliable analysis.

After the tweets processing, they can be applied to the Sentiment Analysis, the second phase, which aims to determine the attitude of the tweets author with respect to a certain company. This analysis task requires the execution of many complex processes because tweets, or natural language texts, are a highly unstructured and meaningless set of words, from the computer perspective. This second phase is an important part of the system since the output is used to identify the potential influential users.

3.6.1 Data Pre-processing

As mentioned in 3.3, only tweets in English and with a cashtag of a company from the S&P 500 index are collected. It is not analyzed emoticons because in addition to appearing in a few tweets (in [24] it is mentioned 7% of tweets), often can lead to a wrong decision of the sentiment.

Since there is a lot of meaningless content in tweets, some pre-processing tasks have been created in order to remove it.

The first pre-processing task is to remove any kind of URLs, pictures, words with numbers or mentioned users. These contents cannot be taken into account in a sentiment analysis since they do not add any kind of sentiment content to it.

After this first filtering phase, the tweets are converted to be all lower case. This is done because the Naïve Bayes Classifier is case sensitive and would consider as different words, for example, “dog” and “Dog”, which is not the objective.

Additionally, cashtags are removed since they are related to companies. It would be possible to remove only the “$” character, however the stock symbol would remain in the tweet and would be considered as a positive or negative word and it is not the purpose. The real objective is that the sentiment analysis of a certain company be made through the sentimental analysis of tweets related to it and not through...
the sentimental classification of the company name, as a word.

Withal, in the case of hashtags, only the special character “#” is removed because, unlike the cashtags, hashtags are often used to express personal feelings, emotions or the context around a given message.

Then, aiming to remove stop-words and words that do not add sentimental information to the tweet, all words that have 3 or less characters are removed from it, except for the words “sad”, “mad”, “joy” and “lol”. These 4 words are not removed because they are strongly related to several feelings.

Finally, special characters are removed, intending to exclude some words written in a language other than English or spam tweets created by the “web-robots”, for example. An example of a spam tweet is presented in Figure 3.4, where the user only uses cashtags.

Figure 3.4. Spam tweet without any kind of useful content.

The order in which the various pre-processing tasks are performed, must be this or have only minor changes, since if, for example, the special characters would be removed before the cashtags removal, then the cashtags could not be identified because of the missing of the “$”.

This data pre-processing is not 100% accurate, since some meaningless content remains in tweets. However, it is impossible to identify all content of this kind. Another issue is that it is not guaranteed that a tweet, with a certain stock symbol, gives information about that company, once many Twitter users post tweets about a certain company product using its cashtag, as in Figure 3.5.

Figure 3.5. Example of a tweet with a company cashtag that not gives any information about it.

All the pre-processing tasks are presented in Table 3.2, followed by its pattern match and an example. The “” means that the word is removed.
Table 3.2. Pre-processing tasks performed in the Content Analysis Module.

<table>
<thead>
<tr>
<th>Pre-processing Task</th>
<th>Pattern Match</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove URLs</td>
<td>Strings beginning with “http”</td>
<td>“<a href="http://aaa.pt%E2%80%9D">http://aaa.pt”</a> to “”</td>
</tr>
<tr>
<td>Remove pictures</td>
<td>Strings beginning with “pic.twitter”</td>
<td>“pic.twitter.com/aaa” to “”</td>
</tr>
<tr>
<td>Remove words with numbers</td>
<td>Strings with numbers</td>
<td>“Formula1” to “”</td>
</tr>
<tr>
<td>Remove mentioned users</td>
<td>Strings beginning with “@”</td>
<td>“@Barack” to “”</td>
</tr>
<tr>
<td>Lower case conversion</td>
<td>All strings</td>
<td>“Dog” to “dog”</td>
</tr>
<tr>
<td>Remove cashtags</td>
<td>Strings beginning with “$”</td>
<td>“$AAPL” to “”</td>
</tr>
<tr>
<td>Remove “#” of hashtags</td>
<td>Strings beginning with “#”</td>
<td>“#happy” to “happy”</td>
</tr>
<tr>
<td>Remove words having 3 or less characters</td>
<td>Strings having 3 or less characters</td>
<td>“and” to “”</td>
</tr>
<tr>
<td>Remove special characters</td>
<td>Every character except letters and blank spaces</td>
<td>“&amp;gm#%” to “gm”</td>
</tr>
</tbody>
</table>

3.6.2 Sentiment Analysis

The implementation of the sentiment analysis is done with the aid of the natural language toolkit NLTK [55] for Python. The purpose of this implementation is to automatically classify each tweet, using Naïve Bayes to create a classifier to identify its sentimental polarity.

A. Classifier Training

To classify each tweet with the Naïve Bayes Classifier, it needs to be trained beforehand. Therefore, it is necessary a big set of sentences with each one labelled as positive, negative or neutral. Since it takes a lot of work to classify manually a huge set of sentences, these kind of sets that are available for free on the Internet are very few and very small.

Because a large sample is needed, the solution to this problem is to collect tweets that contain words associated with positive and negative feelings and use this set of tweets to train the classifier. For that purpose, it is picked 86 emotion hashtags and used as search terms to collect all the tweets in the period between 15 November 2014 and 17 November 2016, using the robot based on GOT described in 3.3. Most of these hashtags were suggested in [56] and are presented in Table 3.3. However, it is not guaranteed that every tweet collected match its sentiment.

After getting the tweets labelled, there are two sets: the positive and the negative. It is applied the pre-processing described in 3.6.1 to them and then, they are merged, creating a single set of tuples, each containing two elements. The first one, containing the remaining words of the tweet and the second, the label of its sentiment.

Then with this set, it is created a list of words ordered by its frequency of appearance to initialize the classifier, because the Naïve Bayes Classifier uses the prior probability of each sentiment label and the
contribution on it of each word.

Table 3.3. Emotion hashtags used as search terms in the collection of tweets to train the Naïve Bayes Classifier.

<table>
<thead>
<tr>
<th>Label</th>
<th>Hashtags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>#related, #overjoyed, #enjoy, #excited, #proud, #joyful, #feelhappy, #sohappy, #veryhappy, #happy, #superhappy, #happytweet, #feelblessed, #blessed, #amazing, #wonderful, #excellent, #delighted, #enthusiastic, #calm, #calming, #peaceful, #quiet, #silent, #serene, #convinced, #contented, #contentment, #satisfied, #relax, #relaxed, #relaxing, #sleepy, #sleepyhead, #asleep, #resting, #restful, #placid</td>
</tr>
<tr>
<td>Negative</td>
<td>#nervous, #anxious, #tension, #afraid, #fearful, #angry, #annoyed, #annoying, #stress, #distressed, #distress, #stressful, #stressed, #worried, #tense, #bothered, #disturbed, #irritated, #mad, #furious, #sad, #ifeelsad, #feelsad, #sosad, #verysad, #sorrow, #disappointed, #supersad, #miserable, #hopeless, #depress, #depressed, #depression, #fatigued, #gloomy, #nothappy, #unhappy, #suicidal, #downhearted, #hapless, #dispirited, #hope, #fear, #worry, #upset, #positive, #negative</td>
</tr>
</tbody>
</table>

For example, assuming that there are 100 positive and 100 negative tweets to train the classifier, that the word “happy” appears in 20 positive tweets and there is a tweet with the word “happy” to classify. Then, the contribution of the word “happy” to the positive coefficient of the tweet would be obtained using (2.1), presented in section 2.6. That is, multiplying the probability of the positive label, which has the value of 0.5 (100 tweets out of 200), by the probability of the word “happy” appears in a positive tweet, which is 0.2 (20 tweets out of 100).

The global positive sentiment of the tweet would be calculated throughout the formula (2.2), presented in section 2.6, i.e. multiplying the value obtained previously by the probabilities of the other words appear in a positive tweets. For example, now assuming that the tweet to be analyzed is “happy birthday”, that the word “birthday” appears in 10 positive tweets and that it is maintained the same above assumption for the word “happy”. Then, the global positive sentiment of the tweet is obtained multiplying 0.5 by 0.2, for the same reasons explained in the last paragraph, and finally multiplying that value by the probability of the word “birthday” appears in a positive tweet, which is 0.1.

B. Classification

With the classifier initialized, to classify the tweets it is necessary to apply them the pre-processing task, using in the classification only the words that remain after this process. It is defined that any word that the classifier does not know, or in another words, that it is not in the classifier’s list of words, is discarded of the analysis.

After this, the classifier finds the log probability, in base 2, for each label (positive and negative). Using the same example given before, the log probability would be -1 for both positive and negative, since each label have the same number of tweets. Then, the log probability of each word given labels, is added to the log probability of each respective label i.e., for each label, the classifier go through the words that remains of the tweet and adds the log probability of each item to the log probability of that
same label. In the end, there are two log probabilities, one for each label and therefore, the sentiment coefficient of the tweet is the positive log probability minus the negative log probability and it is showed in the (3.3).

\[
\text{sentiment coefficient} = \text{positive log probability} - \text{negative log probability} . \tag{3.3}
\]

At this moment, this classifier only outputs a sentiment coefficient that if it is greater than 0, the tweet will be considered positive, and negative otherwise. In the following section, the classifier is improved to be able to classify neutral tweets also.

C. Classification Improvement

To improve the classifier accuracy and efficiency, it is removed 97% of the words set, which is the list ordered by its frequency of appearance. The 3% of words that remained are the most frequent ones and consequently the most representative of each label. This removal is done since there are many tweets to initialize the classifier, its list of words would have many words that are not useful to the classification. The 3% value was chosen based on an empirical process illustrated in Figure 3.6 and Figure 3.7. In Figure 3.6 it is possible to verify the decrease of the classification time with the decrease of the number of words used by the classifier. In the Figure 3.7, it is verified that the best percentage of words to use is the 3% that appears most, achieving an accuracy of 72.16% classifying the sentences just as positive or negative. The dataset used for this experience is available in [57] and is constituted by 372184 different words and 7086 sentences, where there are 3995 positive sentences and 3091 negative.

![Figure 3.6. Time elapsed by the classifier identifying the sentiment polarity of the tweet using different percentage amounts of words.](image-url)
However, in an investment strategy, it is important to consider a neutral sentiment which indicates that a certain tweet does not give any important information and so, there is no need to buy or sell any stocks, for example. As it is very unlikely that the output coefficient be exactly 0, it is necessary to define a limit value from which the tweet is considered neutral, which is called the neutral boundary. For example, if the neutral boundary is defined as 0.2 then, any tweet with a sentiment coefficient between -0.2 and 0.2 is considered neutral.

For this purpose, by an empirical way and using the dataset available in [58], a neutral boundary was defined for the most accurate classifier, the one with only the 3% top words. This dataset has 5513 sentences labelled manually but 1786 are considered irrelevant, like tweets that are written in another language other than English. Then, 3727 sentences left out of the 5513 that are labelled as positive, neutral or negative.

In this approach, it was tested different values for the neutral boundary, proving that the most accurate value for it is 0.22. In Figure 3.8 is represented the accuracy achieved in the dataset classification for each neutral boundary tested value. The accuracy of the same classifier for the same dataset is also represented, but only using positive and negative sentences, i.e. not using the neutral sentiment as hypothesis.

It was predictable that accuracy would drop once a new sentiment was introduced. However, it is expected that the results on the stock market will be better.
Figure 3.8. Accuracy achieved by the 3% top words classifier in the dataset classification for each neutral boundary tested value (blue) and accuracy achieved in the same dataset classification without taking into account the neutral sentiment (red).

The summary of the classifier settings are presented in Table 3.4.

Table 3.4. Details summary about the classifier used.

<table>
<thead>
<tr>
<th>Classifier Summary</th>
<th>Naïve Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm used</td>
<td>Naïve Bayes</td>
</tr>
<tr>
<td>Percentage of words retained from the training set</td>
<td>3%</td>
</tr>
<tr>
<td>Total amount of words retained from the training set</td>
<td>11165</td>
</tr>
<tr>
<td>Neutral Boundary</td>
<td>0.22</td>
</tr>
<tr>
<td>Neutral Sentiment Range</td>
<td>-0.22 ≤ sentiment coefficient ≤ 0.22</td>
</tr>
<tr>
<td>Positive Sentiment Range</td>
<td>0.22 &lt; sentiment coefficient</td>
</tr>
<tr>
<td>Negative Sentiment Range</td>
<td>sentiment coefficient &lt; -0.22</td>
</tr>
<tr>
<td>Accuracy on [57] dataset for positive or negative classification</td>
<td>72.16%</td>
</tr>
<tr>
<td>Accuracy on [58] dataset for positive or negative classification</td>
<td>75.45%</td>
</tr>
<tr>
<td>Accuracy on [58] dataset for positive, neutral or negative classification</td>
<td>60.65%</td>
</tr>
</tbody>
</table>

The content analysis task is illustrated in Figure 3.9.
3.7 User Features Module

The User Features Module is another important part of the system, since it is where the features that reveal a user influence are selected, as well as their ideal values. Based on these values, it is identified the influent users using the Genetic Algorithm.

3.7.1 User Features

These features are in the base of the approach to identify the influent users. They are calculated from the parameters detailed in Table 3.1. There are 11 features studied in this work, some of them inspired in a prior work of A. Pal et al. [31]. All features are scaled to have values starting at 0 and the higher this value, the more influential the user.

The first feature is the Age of the Account which is calculate through the formula (3.4) and it is measured in days. This feature is important because it is expected that a user that has an older account could
spread his content further in the network than a user with a recent account, reaching a bigger audience.

\[
\text{Age of the Account} = \text{current date} - \text{account creation date}.
\]  \tag{3.4}

Another feature is the Account Activity which shows how often a certain user posts tweets, since he creates his account. The expectation is that an influential user posts more tweets than a non-influential and therefore, has a large feature value. However, a spam account would post a lot of tweets and should be detected by this feature, having a large value on it, which formula is:

\[
\text{Account Activity} = \frac{\text{total number of tweets ever done}}{\text{Age of the Account}}.
\]  \tag{3.5}

The Popularity feature is given by the formula (3.6). A value nearly 1 on this feature means that the user has a lot of followers, or in another words, that there are many people interested in his content. Additionally, it also reveals that the user is following few people, which can be associated to an influential user. If the value is nearly 0, then the user has few followers and it may be an indication that the account is a spammer one, for example.

\[
\text{Popularity} = \frac{\text{number of followers}}{\text{number of followers} + \text{number of followings}}.
\]  \tag{3.6}

The Company Connection is a feature that intends to measure how much the user is involved with a certain topic, which in this case is a cashtag related to a certain company. A value nearly 1 indicates that the user is interested in this company and may be related with an influential user in this topic. Otherwise, a small value means that the user tweets much more about other topics and may not be influential on this one. The formula is:

\[
\text{Company Connection} = \frac{\text{number of tweets about the company during the period}}{\text{number of tweets during the period}}.
\]  \tag{3.7}

Another feature that is associated to influence is called Likes and intends to measure if a user receives a lot of likes, which may indicate that he is an influential user. As it is not feasible to monitor the likes that the user is doing, the denominator intends to calculate an average of likes that the user could do during the period. So a large value means that the user received many likes on his tweets during the period, which is a sign of influence.

\[
\text{Likes} = \frac{\text{number of likes received during the period}}{\text{total number of likes ever done} \times \frac{\text{period size}}{\text{Age of the Account}}}
\]  \tag{3.8}

The Cashtags feature represents the variety of cashtags that a certain user uses. A value nearly 0 means that the user post tweets with a lot of cashtags, which could be a spammer account or a non-influential user. However, an influential user may post about several topics too.
There are two features retweets-related which indicate the impact of the user content on the others. The first one is Retweets by Tweet given by (3.10) and is used to understand if the user’s tweets are often retweeted by other users or not. A value nearly 1 indicates that all the tweets of the user were retweeted and this may be related to an influential user. However, if the value is nearly 0, then it means that no tweet has been retweeted.

\[
\text{Retweets by Tweet} = \frac{\text{number of tweets that have been retweeted by other users}}{\text{number of tweets about the company during the period}}. \tag{3.10}
\]

The other feature related to retweets is the Retweets by User, which is represented in (3.11) and if its value is large, it means that there are many different users who retweeted the user’s content, which may be related to an influential user. A small value or nearly 0 means that no users retweet his contents.

\[
\text{Retweets by User} = \frac{\text{number of different users who retweeted the user}}{\text{number of tweets about the company during the period}}. \tag{3.11}
\]

As the retweets features, there are also two features about mentions. The Mentions by Tweet intends to measure whether the number of times the user is mentioned by the other users is greater than the number of times that he mentions them. In this case, it has a large value greater than 1, but in an opposite scenario, it is nearly 0. It is expected that an influential user has many mentions done to him and a small amount of mentions done to other users.

\[
\text{Mentions by Tweet} = \frac{\text{number of mentions done to the user}}{\text{number of mentions done by the user} + 1}. \tag{3.12}
\]

The second feature about mentions is the Mentions by User. A large value means that there are many users mentioning the user account, which is a sign of influence. Otherwise, a value nearly 0 shows that the user mentions a lot of other users but it is not reciprocal.

\[
\text{Mentions by User} = \frac{\text{number of different users that mentioned the user}}{\text{number of different users mentioned by the user} + 1}. \tag{3.13}
\]

The Talk feature shows how much the user digresses into conversations with other users. The 3 factor intends to put more weight in a conversational tweet done by the user, since an influential user should not chat very often with the others. A large value means that the user is more focused posting about the company than in chat, and a value nearly 0, shows the opposite.

\[
\text{Talk} = \frac{\text{number of tweets about the company during the period}}{3 \times \text{number of conversation tweets done by the user} + 1}. \tag{3.14}
\]
All the features described above are presented in Table 3.5.

Table 3.5. Brief description of each calculated feature used in the system.

<table>
<thead>
<tr>
<th>Features</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age of the Account</td>
<td>Days since the account was created</td>
</tr>
<tr>
<td>Account Activity</td>
<td>How often the user post tweets</td>
</tr>
<tr>
<td>Popularity</td>
<td>Ratio between number of followers and the number of followings</td>
</tr>
<tr>
<td>Company Connection</td>
<td>How much the user is involved with a particular company</td>
</tr>
<tr>
<td>Likes</td>
<td>Amount of likes the user receives per tweet</td>
</tr>
<tr>
<td>Cashtags</td>
<td>Inverse of the number of different cashtags that the user includes per tweet</td>
</tr>
<tr>
<td>Retweets by Tweet</td>
<td>Proportion of tweets that are retweeted</td>
</tr>
<tr>
<td>Retweets by User</td>
<td>Number of different users that retweeted the user per tweet</td>
</tr>
<tr>
<td>Mentions by Tweet</td>
<td>Relationship between number of mentions done and received</td>
</tr>
<tr>
<td>Mentions by User</td>
<td>Relationship between number of different users that mentioned the user and number of different users mentioned by him</td>
</tr>
<tr>
<td>Talk</td>
<td>How much the user digresses into conversations with other users</td>
</tr>
</tbody>
</table>

3.7.2 Genetic Algorithm

The objective of using the GA is to optimize the users’ features, to find the values which better describes a Twitter influent user on the stock market. For this purpose, several settings and tasks are done, which are described in detail in the present section.

A. Representation of Individuals

Each individual is constituted by 11 genes, where each gene is a feature calculated in section 3.7.1 and presented in Table 3.5.

Since each user have different values in each feature compared to other users, there is a great dispersion which causes some problems to the GA to find the best result. The solution for this problem is, rather than associate each individual to each user, group several users and associate them to a single individual. These groups are called ranks and the user of the system has to choose how much ranks he wants to create.

The criterion used to do this association is the similarity of the features values between the users. Firstly, for each feature, all the values are sorted decreasingly by its value creating 11 sets where the users are ordered in each one by the value that they have on that specific feature. Then, each one of the 11 sets is divided in n ranks, with n being the number of ranks that the system’s user chosen, as illustrated in Figure 3.10.
In these ranks, the features of the user for the various companies are used individually, as the user may have certain features in relation to a company that may not have to another. Thus, the same user may appear more than once on the ranks of the same feature, since he has tweeted about more than one company.

For example, supposing there are 1500 different users in which 1000 tweeted about two companies and the other 500 just about one. If the system’s user choose 25 ranks then, for each feature, there are 25 sets constituted by 100 users each one, where the 100 users with the highest values on that feature are in the first set and the 100 users with the lowest values on that feature, are included in the last set, the twenty-fifth. This process is executed for each feature. Bear in mind that a user may be included in the first rank of a certain feature and yet, for another feature, be included in the last rank.

By ranking the users, the GA is able to choose, for each feature, the rank subset that represent the best solution and that has more impact on the stock market. This approach allows GA to choose the fittest rank of each feature in an independent way, analyzing the features apart from one another, which is a great advantage.

Therefore, the output of the GA is the optimal individual that have the best rank of each feature and then, each rank is translated into the range of values that it corresponds to, as shown in Figure 3.11. These values are the ones that better describes the features of the influent users.

Figure 3.11. Translation process example of each rank into the range of values that they correspond to.
B. Initial Generation

The initial generation size is the number of ranks chosen by the system's user, which is always kept the same throughout the process. This value must be between 10 and 50. It must be greater than 10 because it is pointless to have ranks with so many users, since the solution would be a great range of values. It must be less than 50 because if many ranks were used, it would lose its purpose of eliminating the dispersion, having ranks with few users and small value ranges.

It is randomly initialized, creating individuals constituted by randomly assigned ranks. Each time the algorithm is executed from the beginning, its initial population is initialized again by the same process. The main idea is that after all iterations, the solution be the same if the input data was the same. For this purpose, the number of generations is 200, which allows the GA to find the best solution spending only a few seconds.

C. Evaluation

The evaluation is made by a fitness function that measures if the users of a certain rank, as a group, had an impact on the stock market and it is calculated through the formula:

$$
\text{fitness}_{\text{individual}} = \sum_{\text{user}=1}^{x} \sum_{\text{day}=1}^{y} \left( \% \text{Variation}_{m}(C, \text{dci}, \text{day}) \times \sum_{\text{tweet}=1}^{z} \text{sentiment coefficient}_{\text{tweet}} \right), \quad (3.15)
$$

with $m$ being the opening or the adjusted closing price, $C$ the company, $\text{dci}$ the number of days after the day the tweet was posted for which the system’s user wants to check the impact of a user, $x$ the number of users present in the rank, $y$ the working days of the chosen period and $z$ the number of tweets posted by the user in that day.

The inner summation gives the user sentiment coefficient about the company $C$ in the day $d$. This value is multiplied by the variation of the company $C$ stock price between the day and the $\text{day}+\text{dci}$, obtaining the score of that individual user. This score allows to differentiate a user which tweets had an impact in the stock variation for a user who had not. These results also allow to differentiate two users whose tweets indicated the stock variation with different level of emphasis. If it is positive, it means that the user was in line with the stock variation. On the other hand, when it is negative, the user’s prediction about the company $C$ was not according to the stock variation. This score is calculated for each day of the period defined by the system’s user and for all the users who are part of the rank, obtaining the individual fitness of it.

The calculated fitness allows the GA to choose a certain rank over the other, giving more chances to the rank with the group of users whose indications overall were more correct.

Since the historical data available about the stock market prices is only at the opening and the closing time of the working days, it is crucial to take into account the timestamp when each tweet was posted, because if a tweet posted after the market opening is considered to make an investment based on the
that day opening price, its results could be misleading. For example, considering that some terrible event for a particular company occurred and a user posted a tweet about it with a negative sentiment. Then, when this tweet is analyzed, its indications are to sell any stock of this company and therefore, the system register the sale for the opening price which leads to get inflected results.

To solve this issue, only tweets made between 2:30 p.m. the previous working day and 2:30 p.m. the same day are considered at that day's opening price. The same happens for the closing price but between 9:00 p.m. of both days. Also, tweets posted on weekends or holydays are associated to the next working day.

The reason is that the companies of the S&P 500 index are listed on the New York Stock Exchange or NASDAQ, which both open at precisely 9:30 a.m. UTC-05:00 and close at the 4:00 p.m. UTC-05:00, excluding extended hours trading. However, tweets are collected in the time zone UTC±00:00 and therefore it is necessary to do the conversion which is done adding 5 hours to the opening and closing time.

D. Selection

Selection is responsible for selecting the fittest individuals to be copied over into the next generation, which happens after calculated the fitness of each individual.

For this purpose, it is employed the Tournament Selection method. This method picks randomly k individuals from the population creating a subgroup, where they compete against each other. The fittest individual of this subgroup survives, passing to the next generation and being able to reproduce. The others, are returned to the population and can be chosen again.

If the tournament size is larger, weak individuals have a smaller chance to be selected. Since it is important to create diversity, it is crucial to give chances to the weak individuals to be included in the next generations. Due to that, the value used for the tournament size is 3, which is a small value compared to the size of the population (10-50 individuals).

E. Crossover

With the parents chosen in the Selection task and to create diversity, it is applied the crossover to generate their offspring. The method used is Double Point Crossover, which chooses randomly two crossover points and the genes between them are copied from the first parent to the child. The rest of its genes are copied from the second parent, as it is illustrated in Figure 3.12.
F. Mutation

The mutation is another task to create diversity in the population. Since the mutation is a slow process in a GA, it is applied the most commonly used mutation rate with the probability of $1/n$, where $n$ is the number of genes of each individual [46].

3.8 Investment Strategy Module

After the User Features Module be executed, the Genetic Algorithm returns the range values of the features that best describes the Twitter users that had more impact on the Stock Market during the defined period.

Since it is rare to find users whose features values are all in the optimal ranges, users who have at least 5 of their features values within these ranges are selected as influents. This set of users then serves as basis for decisions to go long or short on stocks, i.e. making decisions based on their opinion, hoping that they may have an impact on the stock market.

Since no limitation is considered in the budget for the buy of stocks and in the number of different stocks that can be held, all those that are suggested by influential users are bought. However, it is only allowed to have in portfolio a stock of each company. For this purpose, there are 2 long strategies and 3 short strategies that are all different from each other and are detailed below.

The association criterion of each tweet to its respective day is mentioned in section 3.7.2.

3.8.1 Long Strategies

Although the main objective is to invest based on the opinion of the influential users, an analyzing
strategy of the general sentiment of all Twitter users that posted during the defined period is also tested.

A. Most popular companies

This is an alternative to the benchmark strategy. In this case, instead of considering the sentiment of the influent users, long decisions are based on the public mood on Twitter about each company.

For each day, it is computed the general sentiment coefficient about each company and are bought stocks of the top-20 companies.

B. Influent users

This is the benchmark strategy to buy and it is applied to each company separately. For each one and for each day of the period, it is computed the average of the sentiment coefficient of all the tweets posted on that day by the influent users of that company. If that value is greater than 0.22, defined in section 3.6.2, this company stock is bought, since there is no stock of this company already in the portfolio.

3.8.2 Short Strategies

The short strategies are 3 different approaches to sell a certain stock and, they are only available to be applied when some buy strategy has been already used and there are stocks in the portfolio.

A. Influent users

In sell strategies, this is the benchmark strategy and it is also applied separately to each company. As the buy strategy similar to this one, it is applied for each company and for each day of the period. It is computed the average of the sentiment coefficient of the all tweets posted on that day by the influent users of that company. If that value is smaller than -0.22, defined in section 3.6.2, this company stock is sold since there is already a stock of this company in the portfolio.

B. After a certain number of days

As alternative, it is possible to sell the stock after a certain number of working day chosen by the system’s user. Even if the day to sell is outside the period, it is also accounted for, in the results.

This strategy does not depend on any type of influence or decision, because it is always triggered as soon as the number of days chosen has passed, after the stock has been bought.
C. Crossing the average of the last days

The third sell strategy is triggered when the stock price is equal to or less than the average of a certain number of working days before. The number of working days can be chosen by the system's user but must be at least 3, because does not make sense a value smaller than that.

3.9 Synthesis

In this chapter it was described in detail the architecture of the system. Each composing module was presented and detailed not only as an individual module, but also from the system point of view and the way it is connected to the other modules.

The data collection was briefly described, with special focus on the innovative way to collect tweets without being subjected to the Twitter rate limits and without compromising the data flow and its quality.

It was also explained how the content analysis is done, with special attention to the sentiment analysis, which is an import part of the system, as well as the users features and the way in which the generic algorithm is applied to them to ascertain their ideal values.

Finally it was presented the various investment strategies whose results are presented in the following chapter.
Chapter 4

System Validation

This chapter describes the evaluation performed to the system. The data collection is detailed, as well as the environment of each simulation made to each proposed case study. These case studies are detailed and evaluated with different investment strategies combined. The evaluation metric is given and the way the period is divided to be tested is also described.
4.1 Data Collection

It was necessary to collect a large amount of data to validate system results in a clear and unambiguous way. This collection was made in several modules according to the type of data required by the system. The output of each collection was stored in a comma-separated values (CSV) file, where each line of it is a data record using the comma as a field separator.

These outputs, as well as their settings and specifications, are detailed below in the following sections.

4.1.1 Tweets Collection

Tweets about the companies present in S&P 500 index were collected in Tweets Data Module, using their stock symbols converted into cashtags as keywords input for the robot.

It was collected tweets posted between July 1, 2016 and December 31, 2016. However, this collection was executed in six distinct phases, each corresponding to one month of this period as shown in Table 4.1.

<table>
<thead>
<tr>
<th>Period</th>
<th>Number of Tweets Collected</th>
<th>Data Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2016</td>
<td>298536</td>
<td>49.7 MB</td>
</tr>
<tr>
<td>August 2016</td>
<td>321943</td>
<td>54.1 MB</td>
</tr>
<tr>
<td>September 2016</td>
<td>297425</td>
<td>50.5 MB</td>
</tr>
<tr>
<td>October 2016</td>
<td>364529</td>
<td>62.1 MB</td>
</tr>
<tr>
<td>November 2016</td>
<td>361356</td>
<td>61.4 MB</td>
</tr>
<tr>
<td>December 2016</td>
<td>280653</td>
<td>47.9 MB</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1924442</strong></td>
<td><strong>325.7 MB</strong></td>
</tr>
</tbody>
</table>

During six months, it was collected a total amount of 1924442 tweets, all related to the S&P 500 companies, leading to a total data volume of 325.7 MB.

4.1.2 User Details Collection

Using the tweets database created from the tweets collection, the details of each user who posted during this period were retrieved using the Twitter Search API in the User Details Data Module. Note that there are several details that are not possible to collect from Twitter Search API, as described in section 3.4. Thus, the results of the details collection for those that are collectible are presented in Table 4.2, since the other details are computed on the local machine, when they are needed.
Table 4.2. User details collection results.

<table>
<thead>
<tr>
<th>Period</th>
<th>Number of User Details Sets Collected</th>
<th>Data Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2016</td>
<td>97137</td>
<td>5.5 MB</td>
</tr>
<tr>
<td>August 2016</td>
<td>97652</td>
<td>5.6 MB</td>
</tr>
<tr>
<td>September 2016</td>
<td>90072</td>
<td>5.1 MB</td>
</tr>
<tr>
<td>October 2016</td>
<td>116574</td>
<td>6.6 MB</td>
</tr>
<tr>
<td>November 2016</td>
<td>109192</td>
<td>6.1 MB</td>
</tr>
<tr>
<td>December 2016</td>
<td>86886</td>
<td>4.9 MB</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>597513</strong></td>
<td><strong>33.8 MB</strong></td>
</tr>
</tbody>
</table>

Since each company is individually processed, a user is allowed to be counted more than once because his details may differ from one company to another.

4.1.3 Financial Information Collection

Financial information was collected for the period between June 15, 2016 and January 15, 2017 in order to have a 15-day security margin regarding the tweets collection dates, which were collected between July 1, 2016 and December 31, 2016.

It was collect the open values for each working day of this period, for each company of S&P 500, as well as the S&P500 index values during the period.

4.1.4 Classifier Training Tweets Collection

For the purpose of training the classifier, it was collected tweets according to the specifications described in section 3.6.2.A, and in Table 4.3, it is described the details about each set collected in the period between November 15, 2014 and November 17, 2016.

Table 4.3. Tweets collection results for the classifier training.

<table>
<thead>
<tr>
<th>Label</th>
<th>Number of Tweets Collected</th>
<th>Data Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>204228</td>
<td>32.3 MB</td>
</tr>
<tr>
<td>Negative</td>
<td>237169</td>
<td>37.1 MB</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>441397</strong></td>
<td><strong>69.4 MB</strong></td>
</tr>
</tbody>
</table>

These tweets sets were submitted to the Content Analysis Module, in order to train the sentiment classifier.
4.2 Simulation Environment

The simulations refer to the period going from July 1, 2016 to December 31, 2016, which is imposed by the tweets dataset, since it is constituted only by tweets posted during this period.

In the simulation, it is assumed that there is purchase power to buy any stock at any time. However, it is established that only a maximum of one stock per company is allowed at the same time in the portfolio, and the minimum holding period for any stock is one day, that is, buy at open and sell at next day open time.

It is only tested the possibility to buy and sell at market open time and price, not being tested buy at the open time and sell at the close time or vice versa, neither buy and sell at market close. Since data processing takes a lot of time and there are many tweets posted during each trading session, acting only on the open time gives a lot of computing time to the system that allows it to make more complex decisions than if it acts at the close time.

The objective is search for opportunities to enter the market on a daily basis, always on long positions and at the same time, trying to anticipate a decrease in the stock price to go on short position. Furthermore, dividends are not included in the results calculation.

The number of days after the day a tweet was posted, defined to check the impact of a user tweet in this simulation, is 5 working days which represents a working week. In [59], the authors tested periods of 10 and 1 days to check the returns, with better results for the 10-days period. Since the results were not as good as expected, the approach in this simulation is to test for a 5-days period, as used in [60].

Nine companies were excluded from the simulation, due to the fact that they do not have all the information available on Yahoo! Finance for the simulation period. The S&P 500 list of companies undergoes frequent changes and these companies have been acquired or replaced during the simulation period.

There are three different case studies further detailed later in this chapter, which are combined with the long/short strategies described in section 3.8. The objective is to find out which are the best combination.

For the short strategy “After a certain number of days” it were tested 2, 5 and 10 days, and for the short strategy “Cossing the average of the last days”, 3, 5 and 10 days were tested.

4.3 Evaluation Metric

In order to evaluate the performance of the proposed solution, it is defined the Return of Investment (ROI) metric which relates, in percentage, the amount gained or lost to the total amount invested, as is shown in (4.1).
For example, if a certain stock is bought for $100 and is sold for $120, the ROI has the value of 20% which means that this investment leads to a 20% profit over the initial investment.

Then, the Profit (%) during the period is given by calculating the ROI average of all bought stocks in the period, as presented in (4.2).

\[
\text{Profit} \, (\%) = \frac{\sum_{k=1}^{n} \text{ROI}_k \,(\%)}{n}.
\]  

However, it is not sufficient to validate the system only taking into the account the ROI of the proposed solution. Therefore, in this study, to understand the quality of its performance, it is established a comparison between ROI of the proposed solution and the ROI of S&P 500 index, since the study is applied to its companies. The ROI of S&P 500 index supposes that someone buys all the companies present in the index and holds them, during the simulation period.

Figure 4.1 shows S&P 500 index curve throughout the simulation period, with each horizontal axis label spaced 15 working days.
4.4 Train and Test Sets

Since users influence can vary over large periods of time, separating the entire period into two sets of training and testing may not be the best approach to validate the system. The approach used in this study, is to create a sliding window that allows to train and test the system for smaller and consecutive periods.

In this system validation, 3 weeks are used to train the system, identifying potential influential users and the following 2 weeks are used to test the system, that is, make decisions based on the influential users and verify their impact at the end. After that, the 5-weeks window slides two weeks, including the previously test period on the new train period and defining a new 2-weeks testing set never used before.

The selected users with impact on a particular period of time, may not be the same users at a different time, since their influence is calculated for each sliding window.

In Figure 4.2 is illustrated the sliding window approach used.

![Sliding Window Diagram](image)

Figure 4.2. Sliding window diagram for an example period of 9 weeks.

In all there are 11 sliding windows in the simulation period, that are described in Table 4.4.
4.5 Case Study I – Simple approach

In this first case study, users which sentiment’s polarity of their tweets were directly related to the stock’s price variation 5 days later, are selected as influents and form the basis of decisions that are taken. Therefore, Genetic Algorithm is not used, as well as chi-square test.

A. “Most popular companies” Long Strategy – “Influent users” Short Strategy

Using the combination of the long strategy “Most popular companies” and the short strategy “Influent users”, the results are less than 1, resulting in a loss of 4.1% (Figure 4.3). Whenever S&P 500 goes down, the system’s profit goes down even more, since probably the influent users are slow to tweet about what happened. Whenever S&P 500 goes up, the system’s profit follows this trend in the same way, since the companies that have influence on it, would be very popular on Twitter that day, so the system identifies them and then buys those stocks, influencing the results.

<table>
<thead>
<tr>
<th>Sliding Window Number</th>
<th>Train Period</th>
<th>Test Period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start Date</td>
<td>End Date</td>
</tr>
<tr>
<td>1</td>
<td>05/07/2016</td>
<td>22/07/2016</td>
</tr>
<tr>
<td>2</td>
<td>18/07/2016</td>
<td>05/08/2016</td>
</tr>
<tr>
<td>3</td>
<td>01/08/2016</td>
<td>19/08/2016</td>
</tr>
<tr>
<td>4</td>
<td>15/08/2016</td>
<td>02/09/2016</td>
</tr>
<tr>
<td>6</td>
<td>12/09/2016</td>
<td>30/09/2016</td>
</tr>
<tr>
<td>7</td>
<td>26/09/2016</td>
<td>14/10/2016</td>
</tr>
<tr>
<td>8</td>
<td>10/10/2016</td>
<td>28/10/2016</td>
</tr>
</tbody>
</table>
B. “Most popular companies” Long Strategy – “After a certain number of days” Short Strategy

Using the combination of the long strategy “Most popular companies” and the short strategy “After a certain number of days”, the results for 2, 5 and 10 days are losses (Figure 4.4). As mentioned in section A, whenever S&P 500 goes up, the system’s profit follows this trend in the same way, since the companies that have influence on it, would be very popular on Twitter that day, so the system identifies them and then buys that stocks. As the strategy to short is independent of the market, the results tend to be poor, because when the stocks’ price are declining, the system continues holding them until the date to sell.

Figure 4.4. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “After a certain number of days” short strategy, first case study.
C. “Most popular companies” Long Strategy – “Crossing the average of the last days” Short Strategy

For the combination of the long strategy “Most popular companies” and the short strategy “After a certain number of days”, the result for the last 3 days is loss, however, for the last 5 and 10 days, are profit (Figure 4.5). As mentioned in section A and B, whenever S&P 500 goes up, the system’s profit follows this trend. However, the strategy to short prevents the results to be poor, because when the stocks’ price are declining, the system sells them instantly.

![Daily ROI during the simulation period](image)

Figure 4.5. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Crossing the average of the last days” short strategy, first case study.

D. “Influent users” Long Strategy – “Influent users” Short Strategy

This case is distinct from the previous ones because the long strategy used is based on influent users, instead of using the general sentiment present in Twitter. As it is possible to see in Figure 4.6, although the good behavior on the most of times during the period, this strategy have a tendency to have great losses whenever the S&P 500 goes down. As mentioned in A, this probably happens due to the time that the influent users take to tweet about events that lead to prices’ decrease. Another possible reason is that the majority of this kind of events occur when the market is closed. Therefore, when the market opens in the following day, stocks’ prices are already much lower than the previous day and then, the unique option is to sell them at these prices.
Figure 4.6. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Influent users” short strategy, first case study.

E. “Influent users” Long Strategy – “After a certain number of days” Short Strategy

As the previous one, this case is based on influent users’ opinion, instead of using the general sentiment present in Twitter. It is possible to see in Figure 4.7, although the good behavior of the profit line whenever S&P 500 is stable, this strategy have a tendency to have great losses when the S&P 500 goes down. This may happen because, as mentioned in B, the strategy to short is independent of the market and therefore, when the stocks’ price are declining, the system continues holding them until the date to sell, result in great decreases when the S&P 500 goes down.

Figure 4.7. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “After a certain number of days” short strategy, first case study.
F. “Influent users” Long Strategy – “Crossing the average of the last days” Short Strategy

This strategy combination, as the two previous ones, based their long decisions on the influent users’ opinion, leading to good results when the S&P 500 is stable. The strategy to short also prevents the results to be poor, because when the stocks’ price are declining, the system sells them instantly. However, since the long decision in this case study is based on all Twitter users which sentiment’s polarity of their tweets were directly related to the stock’s price variation 5 days later, the results are positive but low, reaching a maximum result of 2% profit for the last 5 days at the end of the period (Figure 4.8).

![Daily ROI during the simulation period](image)

Figure 4.8. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Crossing the average of the last days” short strategy, first case study.

4.6 Case Study II – Implementation of GA

In order to improve the results of the first case study, which reached a maximum profit of 2%, the second one introduces some complexity using users’ features and implementing the GA. The purpose is to optimize them to identify the best values for each one, that better describes a Twitter user with impact on the stock market, as described in 3.7.

It is used 50 ranks to test because it was used in [61] with good results. It was also defined that for a user to be identified as influential, he would have to have at least 5 feature values in the 11 features ranges outputted by the GA, since it is almost impossible to find users with all their values, or almost all, in the optimal ranges.

This approach was applied to each sliding window. Therefore it is not guaranteed that for all of them,
the same features range values were chosen, since it depends on the training period information. That is exactly the main goal: use the training information at that specific time, in order to be able to adapt to the test moment trend, since the users with impact on a particular period of time may not be the same users at a different time, due to political, social and economic factors.

The features range values that were chosen are presented in Table 4.5 and Table 4.6. As it is possible to verify, there is a pattern in Age of the Account feature, as well as for the Talk and the Account Activity features, having values’ ranges very similar.

Table 4.5. Features range values outputted by the GA for 5 out of the 11 features.

<table>
<thead>
<tr>
<th>Sliding Window</th>
<th>Age of the Account</th>
<th>Popularity</th>
<th>Company Connection</th>
<th>Likes</th>
<th>Cashtags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[388;428]</td>
<td>[0.993;0.994]</td>
<td>[0.004;0.005]</td>
<td>[0.987;1]</td>
<td>[0.001;0.001]</td>
</tr>
<tr>
<td>2</td>
<td>[395;441]</td>
<td>[0.992;0.993]</td>
<td>[0.000;0.285]</td>
<td>[3.727;7869]</td>
<td>[0.003;0.003]</td>
</tr>
<tr>
<td>3</td>
<td>[399;476]</td>
<td>[0.992;0.994]</td>
<td>[0.002;0.002]</td>
<td>[0.005]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>4</td>
<td>[388;414]</td>
<td>[0.991;0.992]</td>
<td>[0.05;0.058]</td>
<td>[0.004]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>5</td>
<td>[2089;2155]</td>
<td>[0.906;0.917]</td>
<td>[0.004;0.005]</td>
<td>[4.132;3724]</td>
<td>[0.055;0.07]</td>
</tr>
<tr>
<td>6</td>
<td>[388;399]</td>
<td>[0.992;0.994]</td>
<td>[0.6;1.0]</td>
<td>[3.878;2660]</td>
<td>[0.001;0.001]</td>
</tr>
<tr>
<td>7</td>
<td>[2074;2138]</td>
<td>[0.485;0.501]</td>
<td>[0.2;0.227]</td>
<td>[4;2808]</td>
<td>[0.066;0.076]</td>
</tr>
<tr>
<td>8</td>
<td>[2782;2829]</td>
<td>[0.792;0.820]</td>
<td>[0.5;0.666]</td>
<td>[3.994;2530]</td>
<td>[0.090;0.1]</td>
</tr>
<tr>
<td>9</td>
<td>[2128;2182]</td>
<td>[0.905;0.914]</td>
<td>[0.003;0.003]</td>
<td>[2.354;4221]</td>
<td>[0.075;0.089]</td>
</tr>
<tr>
<td>10</td>
<td>[2231;2303]</td>
<td>[0.992;0.994]</td>
<td>[0.002;0.002]</td>
<td>[3.687;975.2]</td>
<td>[0.072;0.082]</td>
</tr>
<tr>
<td>11</td>
<td>[2136;2166]</td>
<td>[0.992;0.994]</td>
<td>[0.003;0.004]</td>
<td>[3.653;1156]</td>
<td>[0.081;0.094]</td>
</tr>
</tbody>
</table>

Table 4.6. Features range values outputted by the GA for the last 6 out of the 11 features.

<table>
<thead>
<tr>
<th>Sliding Window</th>
<th>Retweets by Tweet</th>
<th>Retweets by User</th>
<th>Mentions by Tweet</th>
<th>Mentions by User</th>
<th>Account Activity</th>
<th>Talk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[0;0.125]</td>
<td>[0;0.142]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[4.9; 5.8]</td>
<td>[14;784]</td>
</tr>
<tr>
<td>2</td>
<td>[0;0.166]</td>
<td>[0.166;0.333]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[6.3; 7.3]</td>
<td>[16;829]</td>
</tr>
<tr>
<td>3</td>
<td>[0.037;0.2]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[4; 4.9]</td>
<td>[14;1586]</td>
</tr>
<tr>
<td>4</td>
<td>[0;0.08]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[4.6; 5.8]</td>
<td>[15;2282]</td>
</tr>
<tr>
<td>5</td>
<td>[0;0.111]</td>
<td>[0.117]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[7.7; 8.8]</td>
<td>[19;5877]</td>
</tr>
<tr>
<td>6</td>
<td>[0;0.1]</td>
<td>[0.105]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[4.7; 5.4]</td>
<td>[15;3256]</td>
</tr>
<tr>
<td>7</td>
<td>[0.187;0.333]</td>
<td>[0.2;0.333]</td>
<td>[1.70]</td>
<td>[1.37]</td>
<td>[7.5; 8.8]</td>
<td>[13;1788]</td>
</tr>
<tr>
<td>8</td>
<td>[0.571;1]</td>
<td>[0;0.092]</td>
<td>[0;0.25]</td>
<td>[0;0]</td>
<td>[12.1; 14.3]</td>
<td>[14;2063]</td>
</tr>
<tr>
<td>9</td>
<td>[0.666;1]</td>
<td>[1;2]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[8.6; 9.6]</td>
<td>[15;2216]</td>
</tr>
<tr>
<td>10</td>
<td>[0.032;0.2]</td>
<td>[0.033;0.25]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[10; 11.6]</td>
<td>[14;1552]</td>
</tr>
<tr>
<td>11</td>
<td>[0.083;0.25]</td>
<td>[0.083;0.289]</td>
<td>[0;0]</td>
<td>[0;0]</td>
<td>[8.4; 9.7]</td>
<td>[15;1417]</td>
</tr>
</tbody>
</table>
A. “Most popular companies” Long Strategy – “Influent users” Short Strategy

For this combination, although the results at the final of the simulation period were lower than S&P 500 (Figure 4.9), they were improved when compared with the same combination for the first case study. This happen because, in this case, were used as base of decision, the influent users which have their features in the ranges selected by the GA.

![Daily ROI during the simulation period](image)

Figure 4.9. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Influent users” short strategy, second case study.

B. “Most popular companies” Long Strategy – “After a certain number of days”

As in A, although the results at the final of the simulation period were lower than S&P 500 (Figure 4.10), they were improved when compared with the same combination for the case study I. The behavior of the results are much less volatile in the most part of the period. However, the results continue to be lower than S&P 500 at the end of the period of simulation.
C. “Most popular companies” Long Strategy – “Crossing the average of the last days” Short Strategy

Once again, the results for this combination were improved, as it is possible to see in Figure 4.11. However, the volatility has increased when compared with the same combination of the first case study.

Figure 4.10. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “After a certain number of days” short strategy, second case study.

Figure 4.11. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Crossing the average of the last days” short strategy, second case study.
D. “Influent users” Long Strategy – “Influent users” Short Strategy

Since the influent users are now chosen by their features values, it was verified that the results have improved, as it is possible to see in Figure 4.12, when compared with the same combination strategy for the previous case study. In this case, the results at the end of the period have actually be higher than the S&P 500.

![Daily ROI during the simulation period](image.png)

Figure 4.12. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Influent users” short strategy, second case study.

E. “Influent users” Long Strategy – “After a certain number of days” Short Strategy

In this case, although the volatility of the results, for 5 and 10 days, the results were higher than S&P 500, as illustrated in Figure 4.13. Once again is verified that the sell strategy “After a certain number of days” were not well succeed when compared with the other results of this case study, due to the same reasons mentioned in E of the precious case study.
F. “Influent users” Long Strategy – “Crossing the average of the last days” Short Strategy

Once again and for this case study too, this combination was the most successful, even with the influential users’ selection criterion change. It was reached a profit of 9.3%, again for the last 5 days, improving the results as it was proposed in the beginning of this section. This is shown in Figure 4.14.

Figure 4.13. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “After a certain number of days” short strategy, second case study.

Figure 4.14. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Crossing the average of the last days” short strategy, second case study.
4.7 Case Study III – Improved GA using statistics

This case study is the benchmark of this study. As the second case study, it uses the GA but with some restrictions on the features used.

Since features have a huge impact on the system performance, partially or totally irrelevant features can negatively impact the results. As the features were intuitively defined, it is made a feature selection to exclude those that do not contribute positively to the output results, hoping to improve the model accuracy due to the less misleading information inputted. Thus, the overfitting is reduced as well as the training time.

Therefore, the objective is to reduce the number of features by half, selecting only the top-5 features that have the most positive relationship with the output results, using a statistical test called chi-square (\(\chi^2\)), described in section 2.8.

The features selected in each sliding window are shown in Table 4.7. It is easily verified that there are 4 main relevant features in the process of identifying which users have the most impact on the Stock Market, which are Account Activity, Age of the Account, Talk and Likes.

Table 4.7. Selected features in each sliding window.

<table>
<thead>
<tr>
<th>Sliding Window</th>
<th>Selected Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Account Activity, Age of the Account, Talk, Retweets by User, Likes</td>
</tr>
<tr>
<td>2</td>
<td>Account Activity, Age of the Account, Talk, Retweets by User, Likes</td>
</tr>
<tr>
<td>3</td>
<td>Account Activity, Age of the Account, Talk, Likes, Cashtags</td>
</tr>
<tr>
<td>4</td>
<td>Account Activity, Age of the Account, Talk, Retweets by User, Company Connection</td>
</tr>
<tr>
<td>5</td>
<td>Account Activity, Age of the Account, Talk, Retweets by User, Likes</td>
</tr>
<tr>
<td>6</td>
<td>Account Activity, Age of the Account, Talk, Retweets by User, Likes</td>
</tr>
<tr>
<td>7</td>
<td>Account Activity, Age of the Account, Talk, Retweets by User, Likes</td>
</tr>
<tr>
<td>8</td>
<td>Account Activity, Age of the Account, Retweets by User, Likes, Company Connection</td>
</tr>
<tr>
<td>9</td>
<td>Account Activity, Age of the Account, Retweets by User, Likes, Retweets by Tweet</td>
</tr>
<tr>
<td>10</td>
<td>Account Activity, Age of the Account, Talk, Likes, Company Connection</td>
</tr>
<tr>
<td>11</td>
<td>Account Activity, Age of the Account, Talk, Retweets by User, Company Connection</td>
</tr>
</tbody>
</table>

A. “Most popular companies” Long Strategy – “Influent users” Short Strategy

Although the implementation of a selection of the best features, buy based on the “most popular companies” and sell based on the “influent users” continues to be a bad combination, as it is possible to see in Figure 4.15. Besides it achieved a positive result, it was lower than the S&P 500. However, the result was better when compared to this combination for the previous two case studies.
B. “Most popular companies” Long Strategy – “After a certain number of days” Short Strategy

This strategy combination revealed again that could not reach good results when applied, even for the benchmark case study. It only overcame the S&P 500 using the last 10 days. Additionally, it revealed a great volatility, reacting badly to the fluctuations on the S&P 500, as it is easily detected in Figure 4.16. However, the best results for this combination were achieved for this case study, which was the main objective.

Figure 4.15. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Influent users” short strategy, third case study.

Figure 4.16. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “After a certain number of days” short strategy, third case study.
C. “Most popular companies” Long Strategy – “Crossing the average of the last days” Short Strategy

For the combination “most popular companies” to buy and using “crossing the average of the last days” to sell, the two main objectives were achieved: better results when compared to S&P 500 and improved the results of the other case studies. However, it was only reached 3.5% of profit for the last 3 days (Figure 4.17), being this one the best number of days tested. Nevertheless, this case study combined with this strategy, improved the results of the other two case studies.

![Graph](Image)

Figure 4.17. Daily ROI of S&P 500 and Profit, during the simulation period, for “Most popular companies” long strategy and “Crossing the average of the last days” short strategy, third case study.

D. “Influent users” Long Strategy – “Influent users” Short Strategy

In this case, is specially tested the capability of detecting and identifying influent users, since the strategy “influent users” is used to buy and sell. However, it was verified in the last two case studies, that using the influent users’ opinion to sell is not a good option, since they take some time to tweet about negative events. However, it was achieved a great improvement on the results, 6.7% of profit (Figure 4.18), which was a very good result when compared to this same strategy when used in the two first case studies, where it was achieved 2.2% and 3.7% profits. Therefore, it was verified the good capability to identify influent users using this third case study.
E. “Influent users” Long Strategy – “After a certain number of days” Short Strategy

For this combination was proved once again, that the capability of identifying the influential users was improved in this third case study, since the results were much better when compared to the results achieved for this combination of strategies when tested on the last case studies. It is possible to verify in Figure 4.19 that despite of the chosen number of days, the results were always profits and greater than S&P 500. The volatility was low and the results achieved a profit of 6.2% in the best scenario (2 days).
F. “Influent users” Long Strategy – “Crossing the average of the last days” Short Strategy

This strategy revealed to be the best one, for all the case studies. Particularly in this case study, it has achieved 14.7% of profit, for the last 5 days. As it is possible to verify in Figure 4.20, the profit’s line were always above the S&P 500 line since the beginning of the period and was always increasing slowly but steadily, with low oscillation and volatility.

![Daily ROI during the simulation period](image)

Figure 4.20. Daily ROI of S&P 500 and Profit, during the simulation period, for “Influent users” long strategy and “Crossing the average of the last days” short strategy, third case study.

4.8 Synthesis

On this chapter, it was described three different case studies with different investment strategies combined. They were trained and tested during a 6-month period using a sliding window approach. This approach, was used to validate the system in smaller and consecutive periods, training the system in the weeks immediately before and thus, making the system more robust to the temporal variation of influential users’ type.

The first case scenario is result of a simple comparison, between the variation of the stock price of a certain company and its general sentiment on twitter. The second one, take into account users features and applies them the GA, to optimize their values and then select users based on its values.

The third case study worked as benchmark, being the result of a successive attempt to improve both previous case studies. This approach works the same way as the second case study, but only uses the top-5 features selected by the chi-square test. It was proved that this third approach is the best one.
Chapter 5
Conclusions

This chapter summarizes this work, draw conclusions and suggests some aspects to be developed in future work.
In this work, it was implemented an innovator system for the purpose of forecast stock market fluctuations, more precisely for the S&P 500 companies, using public mood collected from Twitter. To address this question, it was done an identification of influent users in each company of S&P 500 index to, based on them, predict those fluctuations.

To make this possible, it was developed an innovator tool that allows to collect tweets without any kind of restriction. Using this tool, it was collected 6 months of posted tweets with S&P 500 companies’ cashtags, between July 1, 2016 and December 31, 2016.

After that, in order to process these tweets, a pre-processing task was done followed by a sentimental classification of each one. This classification was made using another tool, developed based on Naïve Bayes techniques, which classifies tweets as positive, negative or neutral. This classifier was applied to the datasets described in [57] and [58] to check its accuracy. In the first one, that has only positive and negative labeled tweets, the classifier achieves 72.16% of accuracy. In the second one, that has tweets labeled as positive, neutral and negative, this classifier achieves 75.45% classifying tweets only as positive and negative. When classifying tweets as positive, neutral and negative, it reaches an accuracy of 60.65%. These results were very satisfactory, since tweets analysis is very complicated due to the informal way that people writes it and also, due to its limited number of characters.

Then, it was proposed to characterize each user through various numerical features related to his Twitter account, and use GA to optimize those values and return the optimal ones, that better describes a user with impact in stock market. After the optimization, it was concluded that the characteristics which best describe an influential user are the account time and the account activity, followed by the number of likes and retweets obtained by posted tweet and the low tendency of the user to talk to other users using tweets.

Finally, three case studies were made for the same period of 6 months, using a sliding window approach where the system was trained during 3 weeks and then tested during the following 2 weeks. The main objective was take into account training information at that specific time in order to be able to adapt to the test moment trend, since the users with impact on a particular period of time may not be the same users at a different time. In each case study, were combined different long and short strategies and their results are presented in Table 5.1.

The third case study worked as benchmark to prove that the test method was valid, which was confirmed by the results, achieving a ROI of 14.7%. This result was obtained using the long strategy that indicates a company stock buy whenever the influent users’ general opinion is positive about that company. The short strategy used in this case is to wait for the stock company price goes across its average of the last 5 days to short.

The influential users selected to be considered in the third case study, are the ones that have their features values in the optimal ranges, outputted by GA, of the top-5 features that have the most positive relationship with the output results. These features top-5 were obtained using the statistical approach of chi-square.
Analyzing Table 5.1, it is possible to find out some interesting conclusions. The first one is that the ROI values for the “Influent users” long strategy are greater than for the “Most popular companies” long strategy, that is, using the identified influent users’ opinion to go long on a company stock is more accurate than using general Twitter opinion about that company.

A second one is that for the third case study, regardless of the strategy used, the ROI is almost always higher than the market’s tendency.

Another interesting conclusion is that, although the opinion of the influent users is the best way to go long, the same does not occurs when it is to go short. This may happen because stock price declines usually happen due to important events, which does not give time to users to tweet before the fall.

The strategy with the best results was the the “Influent users” - “Crossing the average of the last days” investment strategy combination, for the third case study, and its results are presented in Figure 5.1. Although it depends on the market behavior, profit has almost always been higher than the market ROI. Despite of this strategy does not react well to the falls in the market, it always recovers very well, even ending up growing and increasing the difference to the market curve after a fall of it.

<table>
<thead>
<tr>
<th>Investment Strategies Combination</th>
<th>Case Study I</th>
<th>Case Study II</th>
<th>Case Study III</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Most popular companies” - “Influent users”</td>
<td>0.959</td>
<td>1.003</td>
<td>1.017</td>
</tr>
<tr>
<td>“Most popular companies” - “After a certain number of days”</td>
<td>0.980</td>
<td>0.990</td>
<td>0.984</td>
</tr>
<tr>
<td>“Most popular companies” - “Crossing the average of the last days”</td>
<td>Last 3 days</td>
<td>Last 5 days</td>
<td>Last 10 days</td>
</tr>
<tr>
<td>“Influent users” - “Influent users”</td>
<td>1.022</td>
<td>1.037</td>
<td>1.067</td>
</tr>
<tr>
<td>“Influent users” - “After a certain number of days”</td>
<td>0.955</td>
<td>0.980</td>
<td>0.969</td>
</tr>
<tr>
<td>“Influent users” - “Crossing the average of the last days”</td>
<td>Last 3 days</td>
<td>Last 5 days</td>
<td>Last 10 days</td>
</tr>
</tbody>
</table>
Figure 5.1. Best results strategy. “Influent users” - “Crossing the average of the last days” investment strategy combination for the last 5 days, applied to the third case study.

There are some points that could be improved in future work:

- Take into account the user who first tweeted indicating a possible variation in the price of a certain stock, and give him more weight as an influential user than others who follow that trend;
- Creation of a dictionary with terms related to the stock market, in order to increase the accuracy of the content analyzer, as well as the sentiment analyzer;
- Increase the simulation duration to test results consistency;
- Test the users using a real portfolio, taking into account variables such as a limited available capital and trading taxes.
- Use GA to optimize variables in the investment phase.
References


