

SENTIMENT ANALYSIS

*Project report submitted in partial fulfillment of the requirement for
the degree of*

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

Saurav Anand (171043)

UNDER THE GUIDANCE OF

Dr. Emjee Puthooran



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2021

TABLE OF CONTENTS

CAPTION	PAGE NO.
DECLARATION	4
ACKNOWLEDGEMENT	5
LIST OF FIGURES	6
LIST OF TABLES	7
ABSTRACT	8
CHAPTER-1: INTRODUCTION	
1.1 Motivation	9
1.2 Aims and Objectives	9
1.3 Structure	10
CHAPTER-2: BACKGROUND	
2.1 Text Mining	11
2.2 Natural Language Processing (NLP)	11
2.2.1 Tokenization	12
2.2.2 Parts of Speech Tagging (POS)	12
2.2.3 Stemming and Lemmatization	13
2.2.4 N-grams	14
2.2.5 Pareto Principle	14
2.3 Machine Learning Classification	
2.3.1 Naïve Bayes	16
2.3.2 Support Vector Machine (SVM)	17
CHAPTER-3: DESIGN	
3.1 Why Python?	19
3.2 System Architecture	20
3.3 Methodology	21
3.4 Requirements Gathering	21
3.4.1 Engine Requirements	22
3.4.2 Graphical User Interface (GUI)	23

CHAPTER-4: IMPLEMENTATION

4.1 Engine	25
4.1.1 The Pipeline	26
4.1.2 Data Gathering	27
4.1.3 Data Filtering I	28
4.2 Classification - Naïve Bayes Algorithm	
4.2.1 Data filtering II	29
4.2.2 Association Rules	30
4.3 Graphical User Interface	31

Chapter- 5: Conclusion

5.1 Objectives and Timing	32
5.2 Future Work	32
5.3 Reflection and Knowledge Gained	33
5.4 Conclusion	33

REFERNCES	34-36
------------------	-------

PLAGIARISM REPORT	37
--------------------------	----

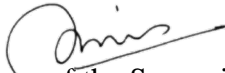
DECLARATION

We hereby declare that the work reported in the B.Tech Project Report entitled “**SENTIMENT ANALYSIS**” submitted at **Jaypee University of Information Technology, Waknaghat, India** is an reliable record of our work carried out under the direction of **Dr. Emjee Puthooran**. We have not succumbed this work elsewhere for any other degree or diploma.

Saurav Anand

171043

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.



Signature of the Supervisor

Dr. Emjee Puthooran

Date: 22-06-2021

Head of the Department/Project Coordinator

ACKNOWLEDGEMENT

We would firstly like to thank our controller Dr. Emjee Puthooran at the Department of Electronics and Communication Engineering at Jaypee University of Information Technology, where this project has been conducted. We would like to thank our supervisor for the help, he has been giving throughout this work.

We have learned from this experience both academically and socially and we are very thankful for having done this research.

We are grateful for the relentless motivation of all other faculty members and encourage us to better this mission.

Finally, our family and friends want to thank you for your continued love. It would have been difficult to finish our job without their contribution.

Saurav Anand

171043

LIST OF FIGURES

Figure 2.1: Tokenization example	12
Figure 2.2: Supervised learning pipeline	16
Figure 2.3: Bayes' Theorem	16
Figure 2.4: SVM - two classes example	18
Figure 3.1: Architecture overview	20
Figure 4.1: Real time component class diagram representation	25
Figure 4.2: Pipeline experiment - Naïve Bayes training phase	27
Figure 4.3: Tokens distribution - Naïve Bayes model	30

LIST OF TABLES

Table 1.1: Objectives set for the project	9
Table 2.1: Part of speech tags used throughout the project	13
Table 2.2: Stemming rules and examples - Porter	14
Table 3.1: Engine - functional requirements	22
Table 3.2: Engine - non-functional requirements	23
Table 3.3: GUI - functional requirements	24
Table 3.4: GUI - non-functional requirements	24
Table 4.1: Sample of the model obtained in the training phase of the classification	28

ABSTRACT

Sentiment analysis, also known as opinion mining, is the computational study of people's spoken feelings, sentiments, behaviours, and emotions in written words. It has been one of the most active natural language processing and text mining research areas in recent years. There are two factors that contribute to its performance. For instance, it has a wide range of applications because opinions are central to virtually all human behaviours and serve as important motivators to our behaviour. When we need to make a call, we want to know other people's opinions. Second, it raises a host of challenging research issues that have never been answered before the year 2000. One of the reasons for the scarcity of r is that it is difficult to find. Because of its relevance to industry and society as a whole, the research has extended beyond computer science to management sciences and social sciences. In this talk, I'll begin by discussing mainstream sentiment analysis research before moving on to some recent work on modelling statements, discussions, and debates.

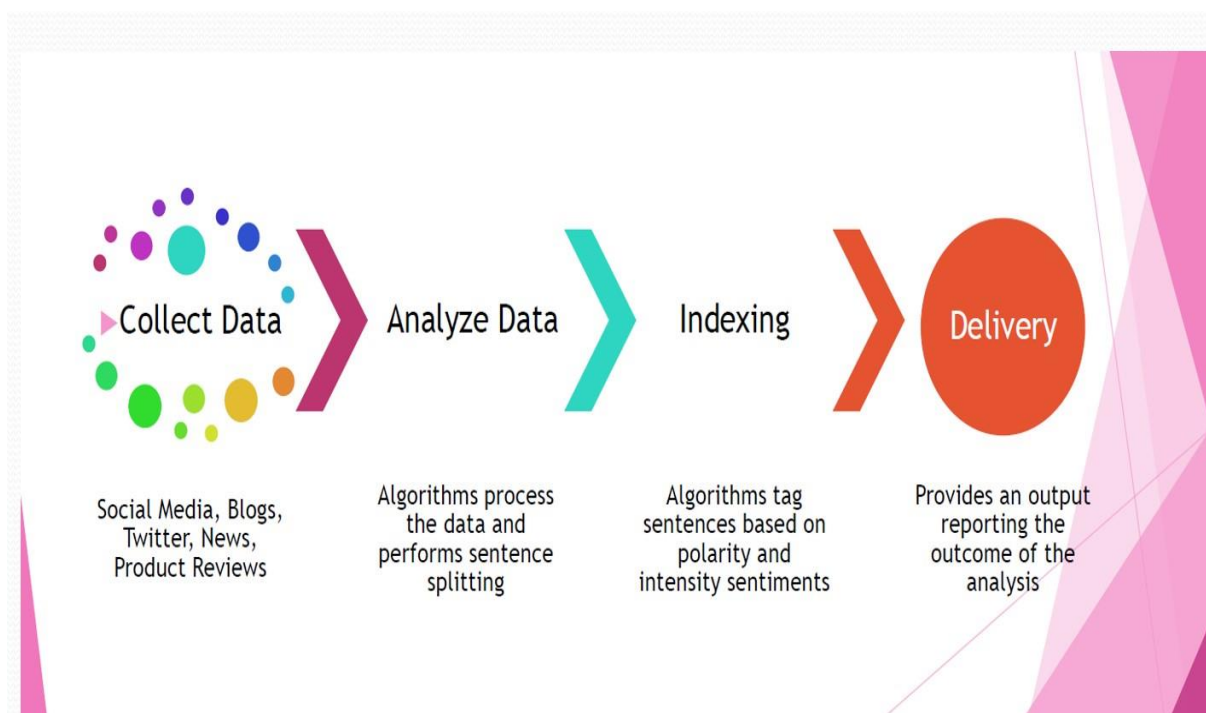


Fig.1 Block Diagram

CHAPTER 1

INTRODUCTION

1.1 Motivation

Individuals have been able to engage in interactive exercises to express their sentiments, musings, feelings on many topics thanks to the emergence of web-based networking stages such as Twitter, Facebook, and Instagram in the most recent decade. On such points, a lot of information is sent (e.g., 5000 tweets per second), providing a possibility for businesses to determine their social impact and consumer perceptions of their products[1]. As result, a computer the system that can adapt the client's behaviour space is appealing for feeling mining and supposition inquiry.

1.2 Aims and Objectives

The venture plans to conduct ongoing opinion research on a variety of labels, services, and subjects. This task's scope of work include is not only a a constant evaluation examination for previous data, and also continuous supposition grouping and detailing. As a result, the system should naturally compile and break down information from Twitter, the project's primary information source. The sentence "Brand An is wonderful," for example, has a favourable slant for A. More complex structures can be used, such as phrase "Brand A is okay, but Brand B is amazing," which is an unbiased opinion of Brand A and a favourable opinion of Brand B. The aim is to build up-to-date assumption esteems for products and subjects before the project is completed. As a result, a system is proposed that uses machine learning calculations and typical language handling approaches to determine the extreme of tweets (Twitter messages).

No.	Objective	Priority
1	Build two infrastructures: real time sentiment analysis and long-term sentiment analysis, employing an engine capable to adapt to both infrastructures	Highest
2	Implement a machine learning algorithm to perform sentiment analysis.	Highest
3	Understand and implement natural language processing techniques.	High
4	Achieve 80% or more in classification accuracy.	High
5	Build a web application graphical user interface for visualisation purposes	Medium

Table 1.1: Objectives set for the project

1.3 Structure

The remainder of the study is divided into six parts. The base knowledge is analysed in the next section, along with the methods and estimates that will be used. The third part, Design, takes a high-level look at the structure that has been delivered. It also includes the strategy designs that were used, the requirements collection measures that were used, both realistic and non-practical, and the techniques that were used. The Implementation Chapter then takes a low-level look at the system, focusing on the problems that users face and the heuristics that has proposed to help overcome them. The testing process follows the evaluation and results section, which compares the system with all comparative tools available and the accomplishments. The technique itself is analysed in a testing phase. The judgement, finally, is a smart section in which the relevant circumstances are analysed for achieving the objectives set at the mission's launch. Furthermore, the experience gained during the work is seen and future work proposed.

CHAPTER 2 BACKGROUND

This chapter will provide an overview of the approaches used in the project. The key principles involved in the creation of the artefacts will be described broadly: Text Mining, Natural Language Processing, and Machine Learning. There is also be an explanation of specialised terms.

2.1 Text Mining

The term "data mining" means the process of examining information found in plain text (for example messages recovered from twitter). The easiest way of extracting useful information from unstructured material sources is to explain this method. Text mining also extends bearing on ads and slanted analysis outside biomedical applications. Text mining is useful in ads for investigating customer relationships with executives. An organization's predictive investigative models for customer turnover may be improved in this manner (monitor client feelings). The primary aim of text mining is the combination of standardised language processing or other computational methods of information into a logical framework. [two] In addition, the data extraction (IE) for this task is important, as there are different views within the area of text mining analysis. The additional material thus attempts to clarify the dynamics of the collection and preparation of data.

2.2 Natural Language Processing (NLP)

Twitter can be used as a running model to explain more theories. Since the maximum duration of a tweet is 140 characters, the details retrieved from Twitter provides a clear indicator of organisation. The advantage of going as far as possible is embodied in the unpredictability of the document test. This research, though, aims to analyse data consistently and dissect a vast number of data (for example, 200 tweets per second). Furthermore, there is no promise to obey any standardised arrangement or accuracy in all tweets. Contractions, short vocabulary and slang are also likely to be included in the content under review. Phrases which convey the same or related ideas may also have entirely different structures and vocabulary[2]. Given the literary restriction previously stated, a predefined printed design must be generated during the preparation stage. The methods mentioned below were used during the project's development.

2.2.1 Tokenization

The key problem is to split the literary details in small fractions until any analysis takes place. Tokenization is a popular advancement in a Natural Language Processing (NLP) programme. The book is split into bits and phrases at a more fundamental level. It is rare for a tweet to contain more than one paragraph because of the limit of length of 140 characters enforced by Twitter. The job priority on this progression is to correctly differentiate sentences in such situations. The accentuation stamps in the content under study such as a period mark "must be deciphered. The next step is to remove words from the phrases. The difficulty of this move is to address spelling in one sentence. This would include correction of orthographic errors and the use of URLs and accentuation in the token arrangement process would not be permitted. Following the tokenization of a tweet, the result is an exhibit comprising a bunch of string, as seen in Figure 2.1.

```
text: "Want to boost Twitter followers ?! http://bit.ly/8Ua""  
tokens: ["want", "to", "boost", "twitter", "followers"]
```

Fig 2.1: Tokenization Example

2.2.2 Part of Speech Tagging (POS)

The relationship between the words must be established to properly understand the context of the phrase. This can be achieved through the syntactic use of a mark for any phrase. This progression, also known as grammatical form marking (POS), can be thought of as a helper for n-gram selection and lemmatization. The grammatical type documentations used in the project are mentioned in Table 2.1.

ADJ : adjective	PART : particle
ADV : adverb	PRON : pronoun
AUX : adjective	PROPN : proper noun
CONJ : conjunction	PUNCT : punctuation
DET : determiner	SYM : symbol
NOUN : noun	VERB : verb
NUM : numeral	X : other

Table 2.1: Part of speech tags used throughout the project

2.2.3 Stemming and Lemmatization

In order to stop and lementize, the inflexional structures and determinations of a word must be reduced to a regular basis[3]. For example, the terms "association," "associations," "connective," "connected," and "interfacing" would all share a "associate" basis. Stemming is a rugged heuristic loop for slashing off term closures and keeping just the foundation structure [3]. Lemmatization, on the other hand, relies on morphological analysis of words to restore their word reference structure (base), which is Usually known as a lemma. In either case, this cycle is dependent on a language like English rather than all the more morphologically rich dialects. Likewise, a lemmatizer will generate uncertainty either by proposing a word-form for all possible lemmas, or by making an unfavourable proposal from two opposing lemmas (for example, is a hatchet or nave plural tomahawks?). On the basis of the above arguments, Porter's stemming calculation was chosen for this task. It consists of five steps in which word reductions are carried out. Every stage has its own set of rules and demonstrations [3]. The rules for the calculation's main time as seen in Table 2.2:

Rules	Examples
SSES -> SS	caresses -> caress
IES -> I	ponies -> poni
SS -> SS	caress -> caress
S -> /	cats -> cat

Table 2.2: Stemming rules and examples - Porter [3]

2.2.4 N-grams

N-gramming is a text mining technique that produces n-length word subsets within a single sentence. The following n-grams can be generated from the phrase "This is a six-word sentence!"

1-grams (unigrams): "this", "is", "a", "six", "words", "sentence"

2-grams (bigrams): "this is", "is a", "a six", "six words", "words sentence"

3-grams (trigrams): "this is a", "is a six", "a six words", "six words sentence"

This will produce 6 unigrams, 5 bigrams and 4 trigrams in the preceding sample sentence. The production of bigrams and trigrams on a larger dataset would increase the size of the data set substantially and delay the process. In Chapter 4, you'll learn how to solve this problem.

2.2.5 Pareto Principle

The pareto theorem states that 20% of experience gives 80% of the return to other wonders. It is named after "an Italian economist Vilfredo Pareto, who observed in 1906 that 20% of the population claimed 80% of the land in Italy" [4]. By pursuing this perception, he discovered that comparable extents can be depicted in financial terms. When extended to the corpus of the mission in the development phase, the pareto rule proved to be an interesting heuristic that improved accuracy and effectiveness.

2.3 Machine Learning Classification

The remaining part would go into the AI equations that were used to group the tweets' extremes (positive, negative, and nonpartisan) into their uniform form. The word "artificial intelligence" refers to the "mechanised position of significant examples in data" [5]. AI has developed a regular method for data extraction in tandem with the growing scale of the data delivered. AI is used in a wide range of areas, from spam sifting and personalised advertising to network indexes and facial recognition programming [5]. While the variety of current calculations is dependent on the learning activity, basic writing distinguishes itself, as shown by the concept of coordination between the PC and the environment. As a result, a distinction is made between AI calculations that are guided and those that are performed alone:

Supervised Learning: The planning details in a controlled AI measurement "requires instances of the information vectors along with their corresponding objective vectors (classes)" [6]. As example, a PC may be programmed to identify feline and canine images in a controlled learning environment. The measurement will prepare a group of marked pictures during the planning period. The machine now learns which pictures include felines and which canines. If new unlabeled images are viewed, the calculation would select the form of the creature in the picture based on what it has already seen. This means that an overall principle that guides contribution to yield is to be "read"[6].

Unsupervised Learning: The degree to which unaided AI estimates are equivalent to administered realising, which is to schedule contribution to yield, is comparable to that of administered realising. The important point is that the material isn't called during the planning period, so the PC has to figure out how to describe the contribution without being directly told. A guided approach (Figure 2.2) seemed appealing as a function of the mission. As a result, two measurements were used, one of which was performed and the other of which was drawn from a pre-actualized library that served as the evaluation cycle's examination centre. The rest of the section will explain the calculations for Naïve Bayes in detail and include a brief description of the supporting vector engine calculation (use explained in Chapter 4).

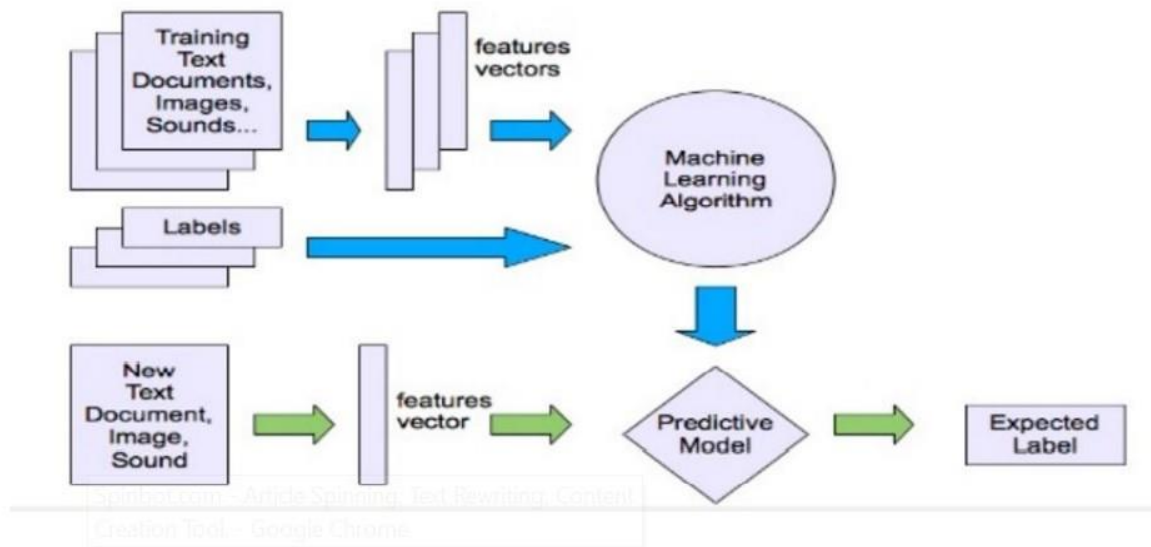


Figure 2.2: Supervised learning pipeline [8]

2.3.1 Naïve Bayes

A supervised machine learning approach is the Nave Bayas algorithm. It is generally considered to be "one of the most powerful and efficient algorithms for data mining learning"[7]. According to independence assumptions, the classifier is based on the Bayes Theorem Figure 2.3. Therefore, the classifier means that the effect of a predictor on a certain class (y) of outputs does not impact other predictors.

The diagram shows the equation for Bayes' Theorem: $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$. Arrows point from the terms to their respective labels: $P(c | x)$ is labeled 'Posterior Probability', $P(x | c)$ is labeled 'Likelihood', $P(c)$ is labeled 'Class Prior Probability', and $P(x)$ is labeled 'Predictor Prior Probability'. Below the equation is the expanded form: $P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$.

Figure 2.3: Bayes' Theorem [9]

$P(c|x)$: posterior target(c) chance given the forecast (x)

$P(x|c)$: likelihood of a class predictor (c)

$P(c)$: preceding class chance

$P(x)$: a predictor's previous chance

The previously stated supposition, also known as class dependent freedom, is often found as a drawback in the accuracy of the measurement. In the Implementation section, various heuristics for dealing with this problem will be added. Consider the previous model, which used images of felines and canines to represent the equation. The classifier must choose between two classes in this situation. The similarity of the two probabilities is used for characterising another image: $P(\text{cats} | \text{new image})$ and $P(\text{dogs} | \text{new image})$. These characteristics will be reported based on the above recipe. The estimation time images will be used to record the probability of both classes, the earlier probability and the probability of the predictor for the calculation of the two classes.

2.3.2 Support Vector Machine (SVM)

Another example of guided computation is the Backing Vector Machine. SVM is used for the construction of a High Dimensional Plane contribution, while Nave Bayes uses a probabilistic approach. A line connecting the groups to the hyperplane would separate them, increasing the distance between them and the hyperplane. When additional models are applied to the model, the calculation produces an optimal hyperplane for them to be arranged in. A SVM computation hyperplane is depicted in Figure 2.4 [27].

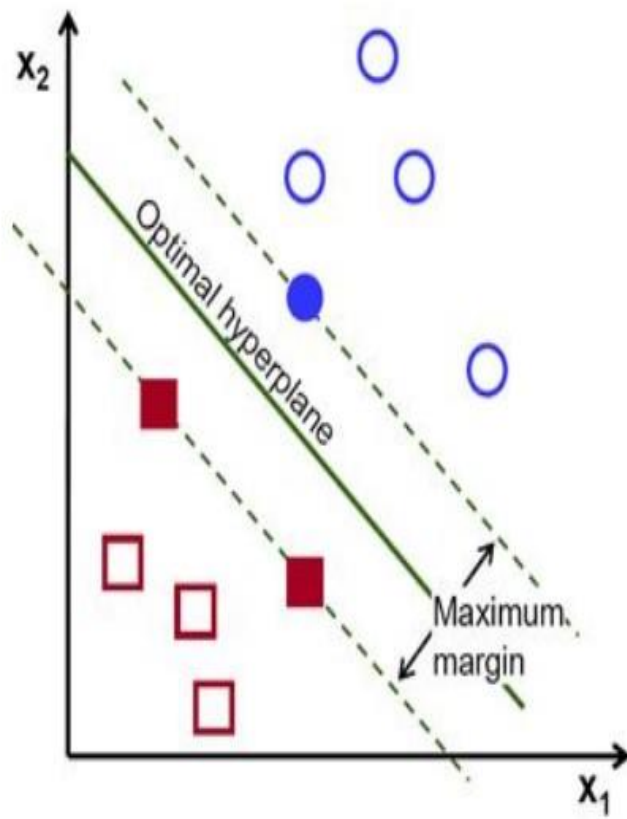


Figure 2.4: SVM - two classes example [10]

If this line cannot be drafted, a kernel function is used to map the data to a level that can distinguish member groups. The line dividing the groups is curved when seen in a lower-dimensional plane.

CHAPTER 3

DESIGN

Overview: This chapter examines the patterns in architecture used during the construction of the project. It provides a high standard description of the framework, including the methods used to deliver the final product and technologies from third parties. In Chapter 4, more information on implementation will be illustrated.

3.1 Why Python?

Since a lot of data is being prepared, it was high priority to memorise the board. As a result, a programming language capable of handling the preparation and capacity of these data measures was essential. When preparing an overview of objects, it must first be saved using an iterative framework methodology requiring memory. In such instances, Python offers generators that are particularly useful for planning a huge number of data by transferring the source data along the chain of preparation, one stage at a time, removing aftereffects from the preparation chain[11]. In light of the preceding argument, Python shows that it is capable of effectively managing memory, a role that is critical for the 'continuous' portion of the project. Python has the drawback of being a deciphered language, and is, paradoxically, slower than embedded dialects (for example, C, Java, and so forth). The engineers network considered the inconvenience and recommended a number of improvements to Python's pace. As a result, projects like Numba and PyPy are appropriate options. Python's developer Guido van Rossum recognises Python's advancements and notes that PyPy is the fastest way to get superior frameworks using Python[12]. Finding specialised library data preparation libraries like NumPy, SciPy[29] were also built by established researchers and field specialists. During the development process, these instruments proved to be helpful and strengthened the Python choice.

3.2 System Architecture

To conduct a sentiment analysis, data manipulation through the processing chain is required. A help module was developed in this regard early in the project. The support node, from now on the pipeline should be capable of integrating and validating the following components: The following components:

1. Module for data collection
2. Modules for Data Filters
3. Modules for Association Rules
4. Classification feeling modules 4.

The pipeline and the components above work together to form the engine of the device. One of the first aims of the project was to develop an engine operating model. This led to the later development of the user interface. Figure 3.1 depicts a high-level architectural outline. The specification and specifications gathering for each part of the structure will be covered in detail in the subsections that follow.

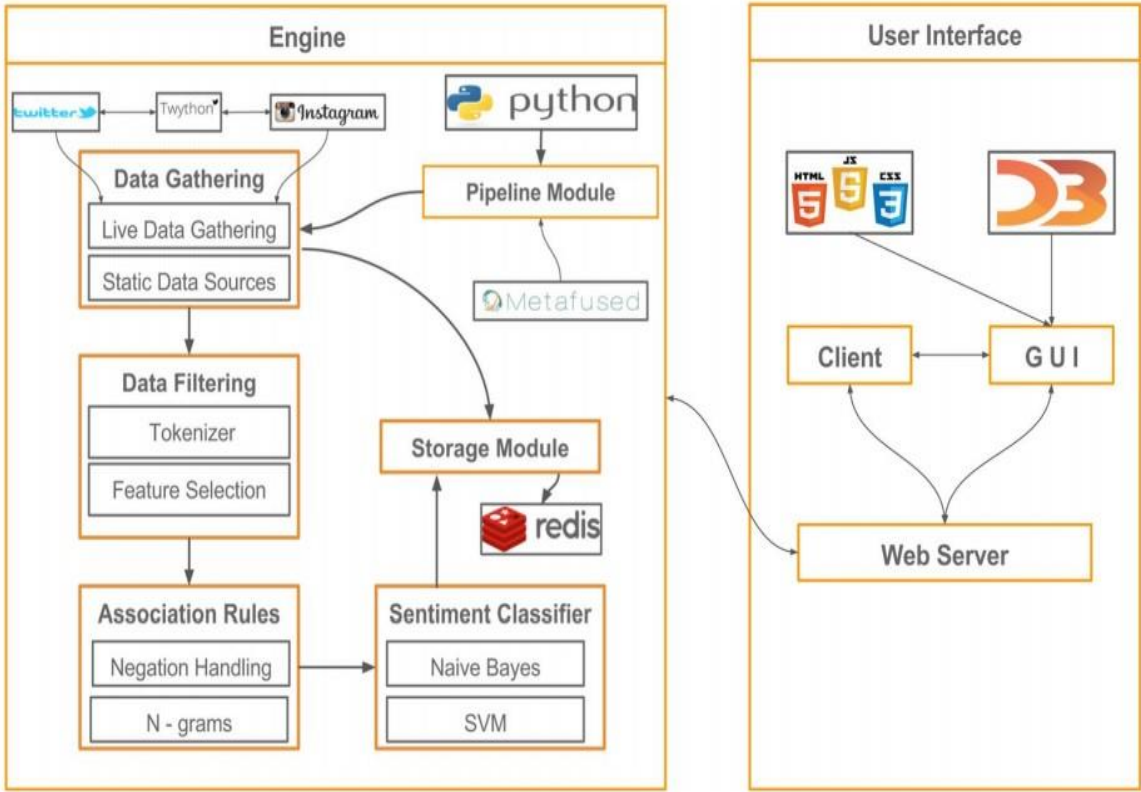


Figure 3.1: Architecture overview

3.3 Methodology

The planning and improvement stages of the project were completed quickly. This is a key component of the deft technique, which was used extensively during the project. The amount of advancement time was split into smaller pieces according to emphasis, each with a cut-off time and various tasks to be performed[15]. Appendix A provides a structure that is smaller than intended, including a roundabout for each circle. As a release management tool, GitHub was also included. A new branch containing the completed tasks was generated at the end of each cycle. The new branch was combined with the expert branch after the code passed the checks with flying colours. If evaluations have not been completed by the code, it was left unmerged before any errors or conflicts were resolved. 16 JIRA, an outsider programming that uses the "Assignments Board" spry practise, was used to help with the development process.

3.4 Requirements gathering

In comparison to a traditional method in advance, needs, both useful and useless, during the advancement era, were gathered at various stages. To begin improving, it was necessary to define a baseline set of requirements and desired behaviour. In this ability, the motor must be adaptable, allowing new modules to be mixed in without affecting the motor's efficiency. In addition, the standard structure for the prepared data was specified by JSON (JavaScript Object Notation), which stored data objects through characteristic value sets. Exploring the state of the craftsmanship in slant examination was a non-practical requirement prior to the advancement period. From this point of view, the precision of the equations used, the adaptability of running heuristics for certain formulas and the time for implementation and examination have all been studied. For eg, it takes longer than one that runs Nave Bays for a device that uses neural organisations. In addition, the computation of the neural network requires inside and outside understanding to perform heuristics, while the probabilistic model like the calculation of Nave Bayes is more adaptable[13]. We will examine in Chapter 5 how driven AI calculations in writing are comparable.

3.4.1 Engine Requirements

The other requirements of the engine are realistic. As a result, heuristics for improving AI calculations, as well as the use of outsider programming, were needed. Table 3.1 presents an underlying list of utilitarian necessities organised by requirement and number of hours required for completion. Since Twitter does not have a local Python API, a strong Twitter stream programming must be used in the development of the motor to achieve the desired continuous behaviour. Twython - a "unadulterated Python covering for Twitter API" [14] - was included in these cases. As there are a number of Twitter Python wrappers (e.g. Tweepy, Twitter Search, Birdy, etc.), Twython's potential to make multi-cutting was the most attractive function. This enables more than one customer to be supported and to deal with a number of demands from the same customer without the need for different engine instances[14].

No.	Functional Requirements	Priority	Hours	Development Stage
1	Train the main classification algorithm	Very High	15-20	Early
2	Output and store the model after the training phase of the classification algorithm	Very High	5-10	Early
3	Test the classification algorithm proposed	Very High	15-20	Early
4	Clean the noise from tweets retrieved and address orthography issues	High	10-15	Middle
5	Retrieve a stream of tweets (domain specific for specified topics) for the long-term components within intervals of : 10 minutes, 2 hours, 2 days	Very High	10-15	Middle
6	Store the acquired data depending on the component in which the engine is used	High	10-15	Middle to Late
7	Train an additional machine learning algorithm for comparison with the main algorithm	Low	12-15	Late
8	Based on the selected topic, retrieve a stream of tweets for an undetermined period of time (until user exits the system)	High	5-10	Late

Table 3.1: Engine - functional requirements

Another major design challenge was data storage. This research involved the lightweight database Redis using a RAM-based filesystem and related database PostgreSQL. Two architecture databases were included in this study. In addition, the engine should be able to facilitate an interpretation of

emotion in both real time and long term. As a result, Redis was first selected because the solution was an efficient temporary and permanent storage solution, with cache operations particularly useful in the long-lasting dimension. A larger set of non-functional requirements is included in Table 3.2.

No.	Non-functional requirements	Priority
1	Extensibility - the engine should be able to implement new modules without performance issues	Very High
2	Efficiency - the engine should be able to support large intakes of data without affecting its performance	High
3	Speed - The engine should perform sentiment analysis in seconds from the time the request was made	Very High
4	Robustness - the system should be able to run multiple instances of engine	Medium
5	Scalability - the engine should scale the output size based on the size of the input	Very High

Table 3.2: Engine - non-functional requirements

3.4.2 Graphical User Interface (GUI)

The graphical user interface was not originally planned because one of the project's key goals is to conduct highly precise emotion analysis. A graphical user interface, on the other hand, proved to be essential for users to communicate with the engine in a more intuitive fashion. As a result, the GUI established a collection of practical specifications, as seen in Table 3.3.

No.	Functional Requirements	Priority	Hours	Development Stage
1	Include textual input field on which sentiment analysis is performed	Very High	5-10	Late
2	Output stream chart in real time	Very High	20-25	Late
3	Display last text received and average sentiment score in panel nearby chart	Very High	15-20	Late

Table 3.3: GUI - functional requirements

Three criteria were defined: functionality and usability, which are non-functional specifications for the graphical user interface. As a result of this, Table 3.4 shows user interface specifications that are not available.

No.	Non-functional requirements	Type
1	The graphical user interface should have a quick response time for user interactions	Responsiveness
2	The graphical user interface should present an intuitive design.	Readability
3	The graphical user interface should output the text body of the tweets analysed	Usability
4	The graphical user interface should update in real time the aggregate sentiment score of the tweets analysed	Usability

Table 3.4: GUI - non-functional requirements

It was a big challenge to make the interface flexible enough to enable the user to track the feelings of two or more subjects simultaneously. To do this, Flask has been used as a third-party micro-platform from Python with an integrated development server and restful application dispatch[20]. Therefore a new URL with the name of the topic is provided whenever a user requests a subject. While the topics concerned, for example "brand A" and "brand b," two maps of "http://localhost:8000/brand A+brand B" are shown in the same window, which seems like a real-time mood analysis for each.

CHAPTER 4

Implementation

Overview: This chapter follows the concept method and differentiates the backend from the front end (user interface) (the graphical user interface). More technical details and how third-party software was used in production will be addressed with both modules.

4.1 Engine

The engine was built from the pipeline (chapter 3 described), which had to be changed for two purposes. The first is a real-time application (described in future subsections). The second goal is to continually run the programme at a predetermined interval, so that a complete evaluation of a variety of things and topics can be made. This is seen in the division of the data collection section into two modules.

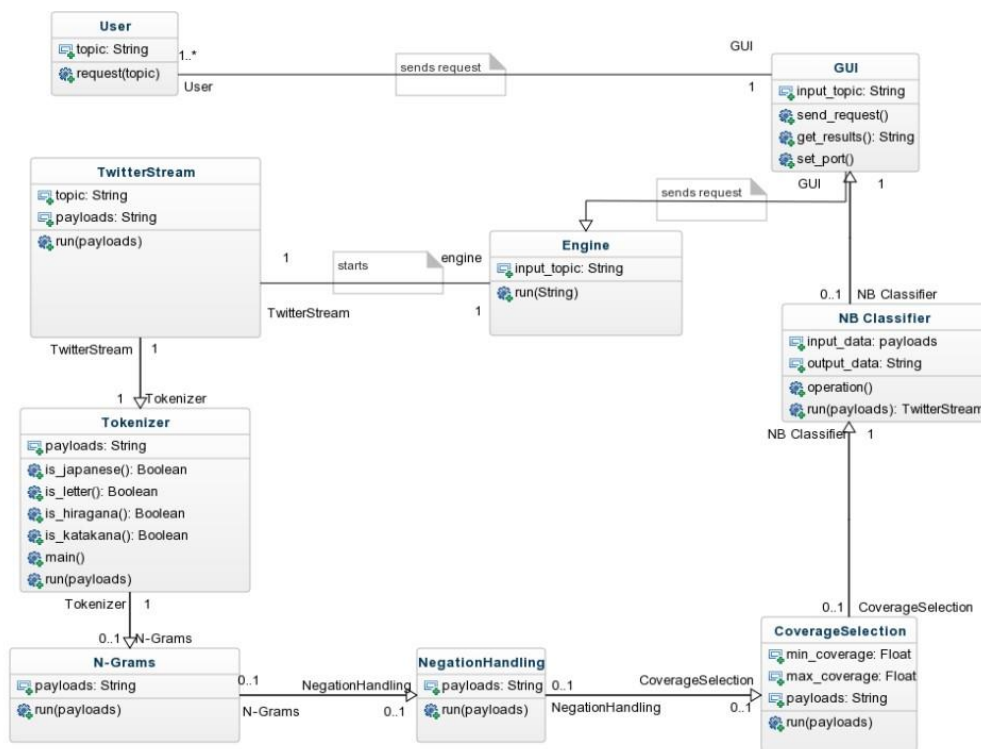


Figure 4.1: Real time component class diagram representation

The pipeline is used to introduce a number of functionalities to the workflow. The last motor prototype constructs the following components:

1. Module for data collection
2. Modules for Data Filters
3. Classification Sentiment Module
4. Modules for Association Rules

The remainder of this subsection will be devoted to discussing the pipeline's and modules' development phases and technical details. Figure 4.1 shows a condensed class diagram of the real-time method.

4.1.1 The Pipeline

The pipeline must provide a working area for data when adding the other components. In addition, the pipeline must be adaptable so that information sources can be quickly changed. When processing information sources with the help of information systems, one common problem is that the available arbitrary access memory capacity can be exceeded. This can lead to the most unappealing system disappointments. An inherent component of Python, capacities generators, was applicable to this issue. The capability of producing the quality needed for other segments of the handling chain can be installed instead of restoring the array of qualities contained in an information system[30]. While there are many benefits of the use of generators, there is one inconvenience. The data must only be called once, whenever created to be used. This required the code to be written with caution, keeping a strategic distance from redundancies, helper variables, and knowledge systems in order to protect the efficient use of memory. Another important part of the pipeline is ensuring that an object is suitable for the purpose for which it was created. From this perspective, the pipeline wanted to serve knowledge systems beginning with one module and progressing to the next while ensuring that the resources of certain objects were saved. The output of the pipeline is a class that, once enabled, supports the different modules within the production chain. Figure 4.2 shows the connection and application of the components to the pipeline. The first step is to create a pipeline class example. If completed, this event may be used to combine various units (e.g.: line 7 to line 12 in the code scrap). After the modules have been inserted with the add step strategy, the pipeline is approached with the run technique to start the handling sequence.

```

01 #Coverage Selection Parameters
02 min_cov = float(sys.argv[1])
03 max_cov = float(sys.argv[2])
04
05 #Training
06 pipeline = MFPipeline()
07 engine = pipeline.add_step(MFTwitterSentiment140Loader('training_data'))
08 engine = pipeline.add_step(MFTokenize5())
09 engine = pipeline.add_step(MFTokenAddBigrams())
10 engine = pipeline.add_step(MFNegationHandling())
11 engine = pipeline.add_step(MFCoverageSelectionTool(min_cov, max_cov))
12 engine = pipeline.add_step(MFSentimentTrainModel('trained_model.csv'))
13
14 #Run
15 pipeline.run()

```

Figure 4.2: Pipeline experiment - Naïve Bayes training phase

4.1.2 Data Gathering

Twitter, an online interpersonal interaction service, is the primary information source used in the project. To introduce two data collection plugins, Twython was used (presented in Chapter 3). In an attempt to be used for the interface, a segment was developed. If a request is made, it will also provide Twython with the request, which will answer with an endless stream of tweets. The other segment gathers data within a defined amount of time. Because advertising concerns practises and the use of a technique, the time factor for analysis must be taken carefully into account. It is doubtful that 10 minutes in depth will enable a mission's progress to be legitimised. Given the amount of time available for the growth of the project, it was not a choice to wait a week to look at a large number of results. The following module was used to store data for the following amounts of time, to fulfil the criteria while still having sufficient time to verify and change the final examination measure: ten minutes, 60 minutes and two days. As a consequence, amendments to other pages will be made in a fair period of time, even if the sum of dissected data provides an exact evaluation of mission performance. In addition, a "Sentiment140" support module has been created to organise the marked data obtained from an outside source.

4.1.3 Data Filtering I

One module has been developed to screen information, which tokenizes literary contributions into tokens (words). The tokenization module uses assistant techniques to classify and discard non-English Characters tweets (e.g.: Japanese, Hiragana, Katakana, and so forth) However, the various image tweets (e.g., the image separated "@" and image separated "#" hashtags) are arranged into a separate JSON field.

4.2 Classification - Naïve Bayes Algorithm

Following the acquisition and sifting of data, the grouping measurement, the motor's primary module, was carried out. The module performed the Nave Bayes measurement, which has two stages: preparation and examination, as seen in Section 3. In the planning phase, marked Sentiment140 information was used. As a result, 1,2 million were used in the planning out of 1,6 million Corpus tweets. The yield is a list of token-evaluations with the following qualities: number of events in the whole data set, number of tweet events and number of events in derogatory tweets. Table 4.1 illustrates this.

Token	Positive counts	Negative counts	Total counts
more	11426	11320	22746
fleet	16	61	77
whole	17	12	29
woods	97	75	172
spiders	33	88	121
like	1087	542	1629
woody	26	15	41
loving	931	12	943
sigh	8	105	113

Table 4.1: Sample of the model obtained in the training phase of the classification

In the test point, the accuracy of the calculation was assessed using secret named information and the probabilistic model created during the design phase. As a result of the measurement, the

characterization accuracy was 71 percent. Such precision was standard for the advancement stage when it was actualized, but further improvements were needed. ²⁴ Since the explanatory index used in the preparation only had markers for positive and negative tweets, but the aim was to remember all three: positive, negative, and neutral, a new heuristic was required. In addition, an appraisal rating scale ranging from - 1 (negative) to 1 (positive) was presented, with tweets with a slant score of [-0.05, 0.05] being considered unbiased. After a manual inspection of the tweets under investigation, the scope constraints were created. The following paragraph depicts the use of heuristics to enhance the calculation's execution and characterization accuracy.

4.2.1 Data filtering II

In order to improve the accuracy of the classifier, a further filter module was later developed in development. It was used during the classification training stage. The problem found was that the tokenization module generated a huge number of tokens, amounting to almost one million. The model was thus computer-intensive, reducing the performance of the algorithm. Furthermore, two strategies were considered:

1. To execute word reductions using Porter's Stemming algorithm.
2. Except low-feeling tokens.

After the tokenisation, Porter's measure lowered corpus dimensions to 60%, but this was not enough, because no particular opinion was seen for a large number of token. Because of the high level of manual labour, physically checking which tokens were normal in such or negative settings was not an option. The pareto norm (explained in Chapter 2) was used to apply an interesting new heuristic. When you look at the tokens of the model prepared for use in Figure 4.3, the most elevated case in the data can be seen as "I" and "the." Though words on the far edge can add a kind of expectation esteem, the quantity of all out events in the corpus is small, so their impact on the module is virtually immaterial. However, when seen outside of their context, such words are meaningless. On the other side, words on the inclines of the bend in the figure appeared to have a deep sense of appreciation (e.g.: "love", "scorn", "astonishing" and so forth) The corpus has been reduced to 20 percent, according to the pareto theorem, and only tokens of high esteem have remained. A hand analysis was carried out to assess where toks with low sense of appreciation appeared in the transport system to determine the focuses of the cut. The final size of the model was then limited to 40,000 tokens. As a result of the commotion (unwanted information) expulsion, the exactness was improved by 7%.

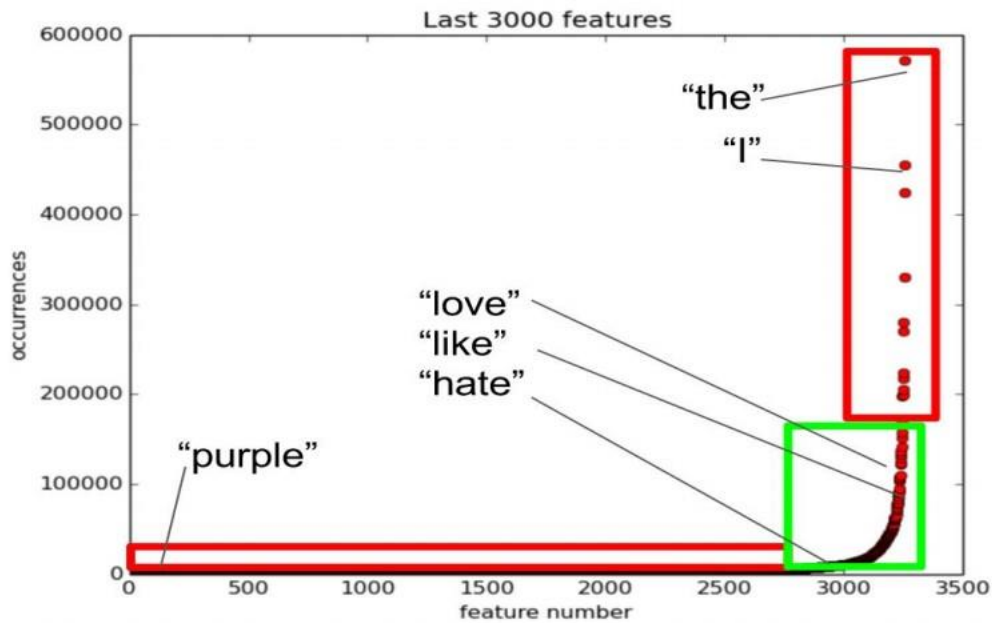


Figure 4.3 : Tokens distribution - Naïve Bayes model

4.2.2 Association Rules

Another popular advancement of computer phonetics is the use of n-grams (as presented in Chapter 2) for which a module was created. The execution of n- grammes including bigrams and trigrams without just taking into consideration major affiliations raises the scale and clamour of the corpus. In this capability, the relationship between objects, modifiers, and action terms was completed using multilingual outsider programming that provided a grammatical style marking device [18]. The classifier's overall accuracy increased by 2.9 percent to 80.9 percent as a result of the addition of bigrams. A separate module was created to deal with refutation in addition to n-grams. "If a word x is preceded by a nullification word (e.g., non, don't, no), then instead of worrying about this as a case of the variable x, another highlight (token) NOT x is created," according to a standard writing technique [4]. For example, after conducting invalidation on the sentence "I don't care for this new brand," the following representation would appear: "I don't NOT like, NOT this, NOT new, NOT brand." The advantage of this aspect is that there are now plain and discredited cases in the corpus. The classifier accuracy was increased by 1.8% to a total of 82.71%. (more subtleties are introduced in Chapter 5).

4.3 Graphical User Interface

The real-time part was given a graphical user interface, as stated in Chapter 3. JavaScript was chosen as the front-end programming language to bring interactivity to the framework. As a result, dynamic charts were created using D3JS, a third-party library. Despite the fact that the library contains a number of models, a custom implementation was needed to satisfy the system's specifications. As a result, the visual component proved difficult due to a lack of prior JavaScript experience (more details in the Conclusion chapter). Appendix B contains screenshots of the graphical user interface.

CHAPTER 5

CONCLUSION

Overview: This is a chapter in retrospect in which we examine the achievement of the initial objectives of the enterprise. Future research and the insight gained during the project are also suggested.

5.1 Objectives and Timing

The project's main goal was to provide a system that could conduct continuous evaluation testing using web-based media outlets. For scanning and reporting the assumption analysis for improvement of points, as shown in the evaluation part, the machine engine was used in the advertising industry. The update of the engine to the speed of the events was expected in December 2015. (introduced in Appendix A). Furthermore, the framework's constant section was designed, and standard language preparation techniques (tokenization, stemming, grammatical type marking, and n-grams) were used to increase the motor's precision. As a result, the classification exactness was set at 82.71 percent, exceeding the underlying target. However, it was deemed important for customer service that the graphical user interface for the continuous part was improved as a media need objective. As a result, customers will work more naturally with the engine, and the visual image of the content under consideration reinforces their agreement. As part of the final achievement of the enterprise, this goal has been achieved.

5.2 Future Work

The implementation of a more perplexing graphical interface would require more work on optimising the delivered motor's capability. As a result, when a client enters a stage, the gui displays the estimation characterization based on previous data as a corresponding calculation. The use of a plug-in to recover URL content found in the tweets under review is another possible increase. During investigation, a large number of URLs were discovered that highlighted Instagram - a web-based media stage. Besides Instagram, it also has plugins that stretch data from other web-based networking sites, such as Facebook with a full Python API and YouTube. A wider range of emotional values will also be explored (e.g., euphoria, outrage, trust, and so on)

5.3 Reflection and Knowledge Gained

My basic knowledge and aptitudes in computer science were greatly increased as a result of preparing and constructing this mission. Overall, when performing the scheme calculation and the subsequent heuristics to enhance its exactness, a superior understanding of the AI field was picked up. Furthermore, critical data in the area of natural language handling was obtained. Furthermore, JavaScript is a new addition to the programming list I saw. I am sure that, by not only using the inherent value of the ancient rareness, but also using Javascript's flexibility and the local visual interface, I can enhance our customer cooperation and possible architectures. The newly learned skills of Python were enhanced with the mastery of advanced concepts like generators and decorators in a similar way[30]. Various exercises were drawn from the perspective of the implementation field - the web-based advertisement industry. Although it has been shown that a concept inquiry instrument is critical for the success of an organization's promoting initiative, a more explicit, theme-oriented item is appealing. Therefore, I would take a more exploratory approach if I wanted to revive the project in order to decide what are the main topics in opinion polling. In either case, comparative functionalities must be included in the expectations. From a practical standpoint, the benefit of starting from scratch with the grouping measurement was expressed in the maximum capacity of the motor's capacities. The pipeline provided was useful for maintaining a synchronised way of writing code as well as a useful investigation tool. Notably, an examination of the characterization estimates available is focused on a second cycle of the project which will prioritise the pipeline for further development stages.

5.4 Conclusion

Because of the previous reflection on the project preparation, the scale of the project will reasonably be concluded within the defined timeline. As a result, the optimal grouping exactness of 80% or more is achieved. The announcements are delivered in stages, and the graphical client interface encourages normal, but comprehensive, client participation. Generally speaking, the project was an ideal chance for me to increase my programming skills in standard language training and AI.

REFERENCES

- [1]. Sisira Neti, S. (2011). SOCIAL MEDIA AND ITS ROLE IN MARKETING. 1st ed. [ebook] International Journal of Enterprise Computing and Business Systems. Available at: <http://www.ijecbs.com/July2011/13.pdf> [Accessed 13 Apr. 2016].
- [2]. Shatkay, H. and Craven, M. (2012). Mining the biomedical literature. Cambridge, Mass.: MIT Press.
- [3]. Nlp.stanford.edu. (2016). Stemming and lemmatization. [online] Available at: <http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html> [Accessed 17 Apr. 2016].
- [4]. Busino, G. (ed) (1965) Oeuvres complètes. Vilfredo Pareto 1848-1923. Geneva: Droz.
- [5]. Russell, Stuart, Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.
- [6]. Konstantinova, N. (2014). Machine learning explained in simple words - Natalia Konstantinova. [online] Nkonst.com. Available at: <http://nkonst.com/machine-learning-explained-simple-words/> [Accessed 18 Apr. 2016].
- [7]. Miskovic, V. (2001). Application of inductive machine learning in data mining. Vojnotehnicki glasnik, 49(4-5), pp.429-438.
- [8]. Slideshare.net. (2016). Lucene/Solr Revolution 2015: [online] Available at: <http://www.slideshare.net/joaquindelgado1/lucenesolr-revolution-2015-where-search-meetsmachine-learning> [Accessed 16 Apr. 2016].
- [9]. Saedsayad.com. (2016). Naïve Bayesian. [online] Available at: http://www.saedsayad.com/Naïve_bayesian.htm [Accessed 21 Apr. 2016].
- [10]. Docs.opencv.org. (2016). Introduction to Support Vector Machines — OpenCV 2.4.13.0 documentation. [online] Available at: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html [Accessed 22 Apr. 2016].
- [11]. Codefellow.org. (2016). 5 Reasons why Python is Powerful Enough for Google. [online] Available: <https://www.codefellow.org/blog/5-reasons-why-python-is-powerful-enough-for-google> [Accessed 23 Apr. 2016]. 34
- [12]. IMPYTHONIST. (2015). Build massively scalable RESTful API with Falcon and PyPy. [online] Available at: <https://impythonist.wordpress.com/2015/09/12/build-massively-scalable-restful-api-with-falcon-and-pypy/> [Accessed 23 Apr. 2016].

- [13]. Shalev-Shwartz., (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press.
- [14]. WhatIs.com. (2016). What is multithreading? - Definition from WhatIs.com. [online] Available at: <http://whatis.techtarget.com/definition/multithreading> [Accessed 23 Apr. 2016].
- [15]. Suzanne Embury(2015),List of Suggested Agile Practicea Available at: <https://moodle.cs.man.ac.uk/file.php/357/Coursework/> [Accessed 24 Apr. 2016].
- [16]. Sentiment140.com. (2016). Sentiment140 - A Twitter Sentiment Analysis Tool. [online] Available at: <http://www.sentiment140.com/> [Accessed 25 Apr. 2016].
- [17]. Lextutor.ca. (2016). LISTS DOWNLOAD. [online] Available at: http://www.lexutor.ca/freq/lists_download/ [Accessed 25 Apr. 2016].
- [18]. Research Blog. (2016). All Our N-gram are Belong to You. [online] Available at: <http://googleresearch.blogspot.co.uk/2006/08/all-our-n-gram-are-belong-to-you.html> [Accessed 25 Apr. 2016].
- [19]. Wiegand, M., Balahaur, A. and Montoyo, A. (2010). Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, Uppsala, July 2010, pp. 60–68.
- [20]. Flask.pocoo.org. (2016). Flask (A Python Microframework). [online] Available at: <http://flask.pocoo.org/> [Accessed 28 Apr. 2016].
- [21]. Kohavi, R. and John, G. (1995). A Study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection, Stanford CA. 94305 , pp. 2-5
- [22]. Picard, Richard; Cook, Dennis (1984). "Cross-Validation of Regression Models".Journal of the American Statistical Association 79 (387): pp. 575–583 doi:10.2307/2288403
- [23]. Metafused Shoots, S., data, H. and Landscape, M. (2015). Metafused Blog. [online] Blog.metafused.com. Available at: <http://blog.metafused.com/search?updated-min=2015-01-01T00:00:00-08:00&updated-max=2016-01-01T00:00:00-08:00&max-results=3> [Accessed 30 Apr. 2016].
- [24]. Frank, E. (1998). Naive Bayes for regression. Hamilton, N.Z.: Dept. of Computer Science, University of Waikato. 35
- [25]. Yang, Z. (2010). Machine learning approaches to bioinformatics. Singapore: World Scientific.
- [26]. Hammell, T. (2005). Test-driven development. Berkeley, CA: Apress, ISBN-10: 159-0-593-278
- [27]. Steinwart, I. and Christmann, A. (2008). Support vector machines. New York: Springer. ISBN-13: 978-0-387-77241-7
- [28]. Arbuckle, D. (n.d.). Learning Python testing. ISBN 978-1847198846
- [29]. Percival, H. (2014). Test-driven development with Python. Sebastopol, CA: O'Reilly Media. ISBN-13: 978-1449364823

[30]. Gorelick, M. and Ozsvald, I. (2014). High performance Python. Sebastopol, CA: O'Reilly.
ISBN-13: 978-1449361594

PLAGIARISM REPORT

Sentiment Analysis

ORIGINALITY REPORT

14%	13%	0%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	studentnet.cs.manchester.ac.uk Internet Source	11%
2	Submitted to Jaypee University of Information Technology Student Paper	1%
3	Submitted to The University of Manchester Student Paper	1%
4	Submitted to Sogang University Student Paper	1%
5	Submitted to National Institute of Technology Delhi Student Paper	<1%
6	Submitted to Softwarica College Of IT & E-Commerce Student Paper	<1%