SENTIMENT ANALYSIS

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

BACHELOR OF TECHNOLOGY

IN

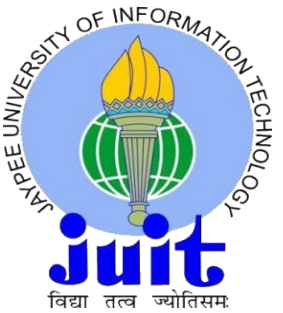
COMPUTER SCIENCE AND ENGINEERING

BY

Dawa Tenzin (171388)

UNDER THE GUIDANCE OF

Dr. Aman Sharma



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,WAKNAGHAT

May 2021

TABLE OF CONTENTS

DECLARATION

I 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 directionof**Dr.Aman Sharma.**I have not succumbed this work elsewhere for any other degree or diploma.

Dawa Tenzin
171388

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

Signature of the SupervisorHead of the Department/Project
Dr. Aman Sharma
Date:

ACKNOWLEDGEMENT

I would firstly like to thank our controllerDr.Aman Sharma at the Department of Computer Science andEngineering at Jaypee University of Information Technology, where this project has been conducted. I would like to thank our supervisor for the help, he has been giving throughout this work.

I have grown both academically and personally from this experience and are very grateful for having had the opportunity to conduct this study.

I am alsothankfultoallotherfacultymembersfortheirconstantmotivationandhelping us to bring improvements in thisproject.

Finally, I would like to thank our family and friends for their constant support.Without their contribution it would have been impossible to complete our work.

Dawa Tenzin

171388

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Sentiment analysis or opinion mining is the computational study of people'sopinions,sentiments, attitudes, and emotions expressed in written language. It is one of the most active researchareas in natural language processing and text mining in recent years. Its popularity is mainly due to tworeasons. First, it has a wide range of applications because opinions are central to almost all humanactivities and are key influencers of our behaviors. Whenever we need to make a decision, we want tohear others' opinions. Second, it presents many challenging research problems, which had never beenattempted before the year 2000. Part of the reason for the lack of study before was that there was littleopinionated text in digital forms. It is thus no surprise that the inception and the rapid growth of the fieldcoincide with those of the social media on the Web. In fact, the research has also spread outside ofcomputer science to management sciences and social sciences due to its importance to business andsociety as a whole. In this talk, I will start with the discussion of the mainstream sentiment analysisresearch and then move on to describe some recent work on modeling comments, discussions, anddebates, which represents another kind of analysis of sentiments and opinions.

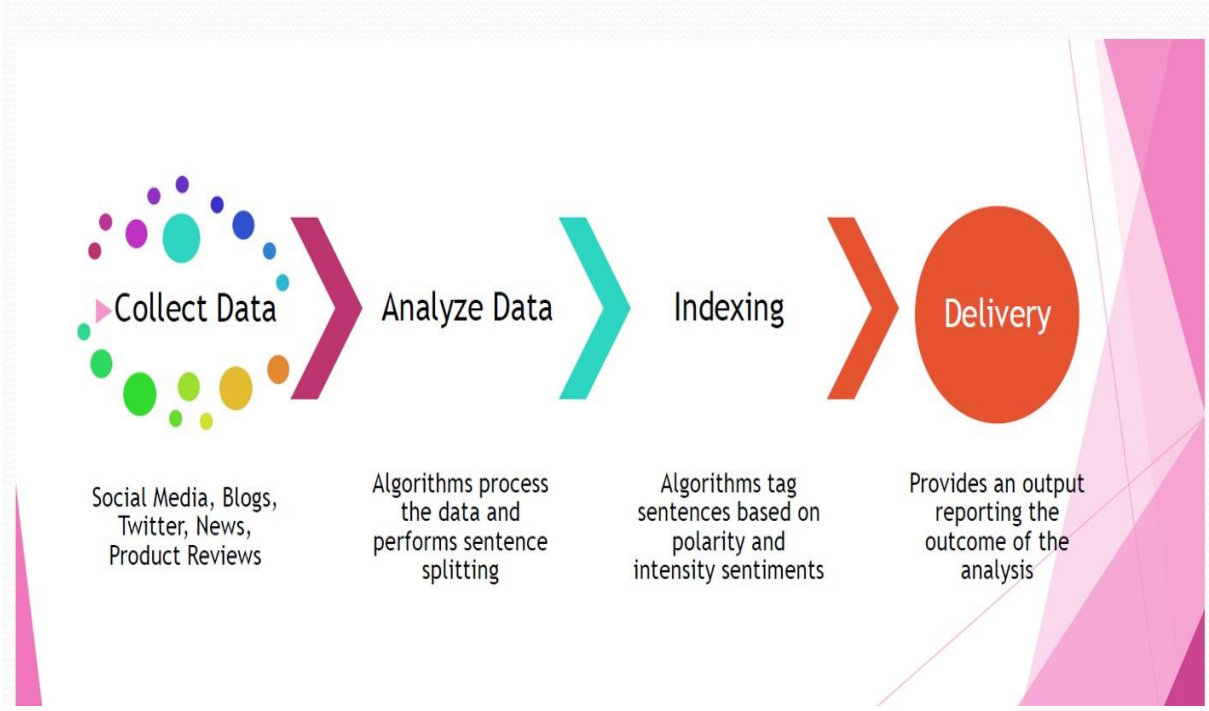


Fig.1 Block Diagram

CHAPTER 1

INTRODUCTION

1.1 Motivation

The rise in the most recent decade of web-based media stages, for example, Twitter, Facebook, and Instagram, empowered individuals to participate in social exercises to communicate their sentiments, musings, what's more, feelings on an assortment of points. On such stages, a lot of information are delivered (e.g: 6000 tweets for each second), this speaking to an open door for organizations to evaluate their social impact and individuals suppositions towards their products[1]. Thus, a computational system is attractive to perform feeling mining and supposition investigation which

can adjust to the action space of the client.

1.2 Aims and Objectives

The venture expects to create continuous opinion examination related with a scope of brands, items and themes. The task's degree isn't just to have static assessment examination for past information, yet additionally supposition grouping and detailing continuously. Thusly, the framework ought to

naturally gather and break down information from Twitter, the essential information hotspot for this undertaking.For instance, the sentence "Brand An is marvelous" has positive slant for Brand A. More advanced structures can be worked, for instance, the sentence "Brand An is alright yet Brand B is incredible" has unbiased notion for Brand A, and positive assessment for Brand B. Before the finish of the undertaking the objective is to create up to the moment assumption esteems for brands and subjects. As such, a framework which decides the extremity of tweets (Twitter messages) by utilizing machine learning calculations and characteristic language handling methods is proposed. Table 1.1 presents the destinations set, organized by the degree of commitment to the extent of the venture.

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 report contains six parts. In the following part, foundation information is examined, alongside the strategies and calculations utilized being developed. The third section - Design, presents an elevated level perspective on the framework delivered. It likewise covers the plan designs applied, the necessities gathering measure, both useful and non-practical, and what strategies have been utilized. At that point, the Implementation Chapter exhibits a low level perspective on the framework, covering the usage challenges, and what heuristics have been proposed to experience them. In expansion, the framework is surveyed in the Testing section, trailed by the Evaluation and Results section, where the framework's exhibition is contrasted with other comparative instruments accessible,also, accomplishments are introduced. Finally, the decision is an intelligent section, where the finish of the goals set at the beginning of the task is surveyed by the circumstance particulars. Also, the information picked up while chipping away at the venture is illustrated, and future work is proposed.

CHAPTER 2
BACKGROUND

Overview: The following chapter aims to clarify the techniques used throughout the project. A broad definition will be given for the core concepts involved in the development of the artefacts: Text Mining, Natural Language Processing and Machine Learning. Furthermore, specialized terminology will be explained.

2.1 Text Mining

Text mining alludes to the examination of information contained in normal language text, (for example messages recovered from twitter). It very well may be characterized as the act of removing important information from unstructured content sources. The application area of text mining shifts from biomedical applications to promoting applications and slant investigation. In advertising, text mining is pertinent to the investigation of the client relationship the executives. This way an organization can improve their prescient investigation models for client turnover (monitor client feelings).The principle objective of text mining is to handle information into an organized arrangement prepared for investigation, by means of use of regular language handling and other scientific methods.[2] Albeit, there are numerous perspectives inside the field of investigation of text mining, data extraction (IE) is pertinent for this task. Thusly, the accompanying material means to clarify the difficulties and phrasing related with data extraction and resulting preparing.

2.2Natural Language Processing (NLP)

To clarify further ideas, Twitter will be utilized as a running model. The information recovered from twitter presents a specific measure of organizing, as in the greatest length of a tweet is 140 characters in length. The upside of as far as possible is reflected in the unpredictability of the examination for an individual bit of text. Nonetheless, this undertaking plans to examine information in a consistent way, where a lot of information (e.g: 200 tweets every moment) will be dissected. Besides, there is no sureness that all the tweets will follow a formal structure, neither that they will be linguistically right. It is additionally anticipated that contractions and short types of words, just as slang will be experienced in the content investigated. Besides, sentences portraying the equivalent or comparable thoughts may have altogether different linguistic structure and utilize totally different vocabularies [2]. Given the previously mentioned literary constraint, a predefined

printed design must be created at preparing time. The methods introduced beneath were utilized in the advancement of the venture.

2.2.1Tokenization

The main undertaking, that must be finished before any handling can happen, is to isolate the literary information into more modest segments. This is a typical advance in a Natural Language Preparing (NLP) application, known as tokenization. At a more significant level, the book is at first partitioned into sections and sentences. As an outcome of the length constraint of 140 characters forced by twitter, it is infrequently the situation that a tweet will contain more than a passage. In such matters, the task focus on this progression is to accurately distinguish sentences. This should be possible by deciphering the accentuation stamps, for example, a period mark ".", inside the content examined. The subsequent stage is to extricate the words (tokens) from sentences. The test at this step is to deal with the orthography inside a sentence. Thusly, spelling blunders must be adjusted, URLs and accentuation will be prohibited from the subsequent arrangement of tokens. As it can be seen in Figure 2.1, in the wake of tokenizing a tweet the returned result is an exhibit containing a bunch of string.

Fig 2.1: Tokenization Example

2.2.2 Part of Speech Tagging (POS)

To comprehend the total significance of a sentence, the connection between its words must be set up. This should be possible by allocating each word a classification that recognizes syntactic usefulness of that word. Otherwise called grammatical form labeling (POS), this progression can be viewed as a helper necessity for n-grams choice and lemmatization. Table 2.1 covers the grammatical form documentations utilized in the undertaking.

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

The objective of both stemming and lemmatization is to diminish inflectional structures and determinationsof a word to a typical base form[3]. For instance the accompanying words: "association", "associations", "connective", "associated", "interfacing" will have a similar base, which is "associate". Stemming, is a rough heuristic cycle that slashes off the closures of words, so that just the base structure is kept [3]. Conversely, lemmatization utilizes the morphological investigation of the words, restoring their word reference structure (base), normally alluded as the lemma. In any case, for a language like English rather than all the more morphologically rich dialects, this cycle depends on a word reference being accessible. Likewise, a lemmatizer can present uncertainty by proposing all potential lemmas for a word structure, or by picking some unacceptable proposition from two contending lemmas (e.g., is tomahawks the plural of hatchet or of hub?). Considering the contentions introduced above, calculation picked for this assignment is Porter's stemming calculation. It comprises of five stages, where word decreases are performed. For each stage, rules and shows to apply them are characterized [3]. Table 2.2 represents the rules of the principal period of the calculation:

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-grams is a common technique in text mining, where word subsets of length n within a sentence are formed. From the sentence “This is a six words sentence!” the following n-grams can be formed:

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”

As such, the example sentence above will produce 6 unigrams, 5 bigrams, and 4 trigrams. On a bigger data set, producing bigrams and trigrams will considerably contribute to the size of the data set, consequently, slowing down the system. An approach to solve this challenge is detailed in Chapter 4.

2.2.5 Pareto Principle

The pareto guideline expresses that, for some marvels, 20% of the info produces 80% of the yield. It is named after "an Italian Economist Vilfredo Pareto who in 1906 saw that 80% of the land in Italy was claimed by 20% of the populace" [4]. By seeking after this perception he seen comparative extents can be portrayed in financial aspects. The pareto rule is fairly an fascinating heuristic which created enhancements in the precision and the effectiveness when it was applied to the corpus of the task in the advancement stages.

2.3 Machine Learning Classification

The remainder of this part will introduce AI calculations used to group the extremity (positive, negative, nonpartisan) of the tweets in their standardized structure. The term AI alludes to the "mechanized location of significant examples in information" [5]. Close by with the developing size of the information delivered, AI has gotten a regular method for data extraction. From spam sifting and customized promoting, to web indexes and face identification programming, AI is applied in a wide scope of spaces [5]. While the assortment of present calculations relies upon the learning task, specific writing makes the differentiation as indicated by the idea of communication between the PC and the climate. Accordingly, the partition is made between directed and solo AI calculations:

Supervised Learning: In a regulated AI calculation the preparation information "includes instances of the information vectors alongside their relating objective vectors (classes)" [6]. For instance, in a regulated learning way a PC can be idea to recognize pictures of felines and pictures of canines. In the preparation stage, a bunch of marked pictures will be prepared by the calculation. Now, the PC 'knows' which pictures contain felines and which contain canines. When given new unlabelled pictures, the calculation will choose dependent on what it 'saw' previously, the sort of creature in the picture. Thus, the objective is to 'learn' an overall principle that guides contribution to yield [6].

Unsupervised Learning: Unaided AI calculations have a similar degree as administered realizing, which is to plan contribution to yield. Nonetheless, the thing that matters is that in the preparation stage the information isn't named, thusly, the PC needs to discover structure in the contribution, without explicitly being advised how to characterize. As a feature of the task, a directed methodology (Figure 2.2) was attractive. Thusly, two calculations were utilized, one executed and the other taken from a pre-actualized library which fills in as an examination base for the assessment cycle. The remainder of this part will clarify in detail the Naïve Bayes calculation (usage is clarified in Chapter 4), and offer a short depiction of the Support Vector Machine calculation.

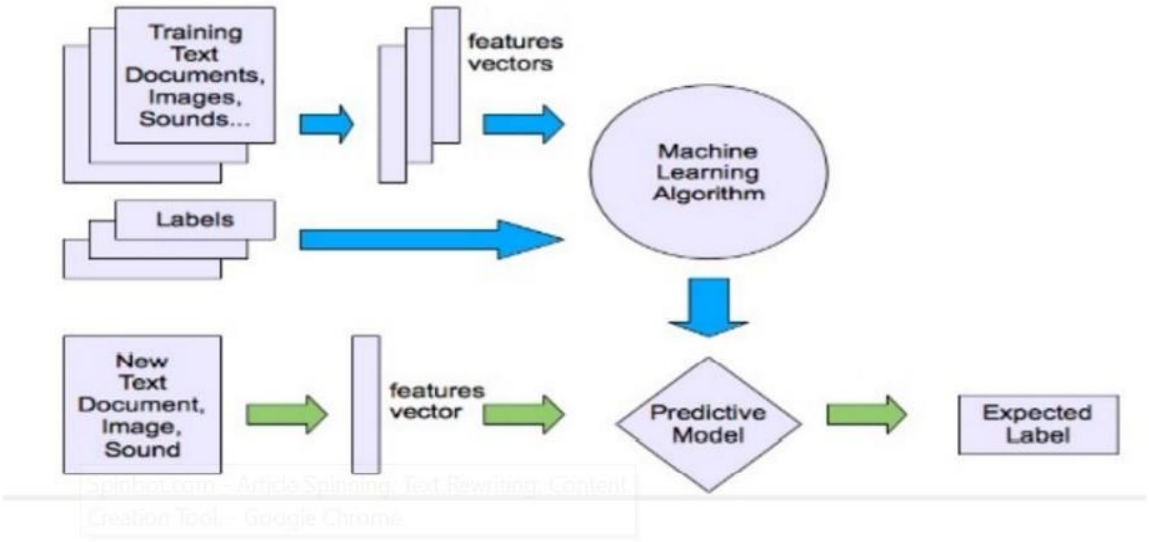
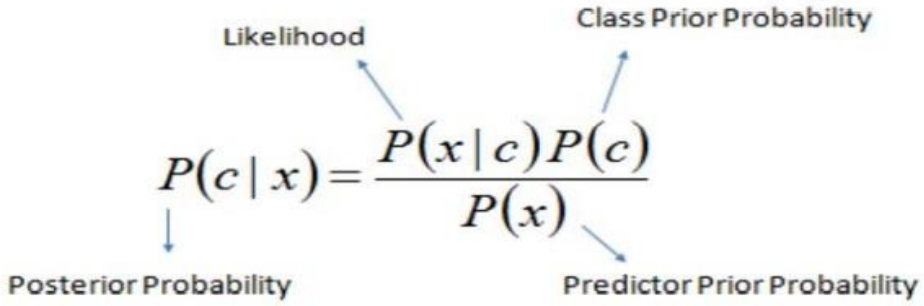


Figure 2.2: Supervised learning pipeline [8]

2.3.1 Naïve Bayes

Naïve Bayes is a supervised machine learning algorithm. It is widely recognized as one of the "most efficient and effective learning algorithms for data mining" [7]. The classifier is build using Bayes' theorem - Figure 2.3, with independence assumptions. As such, the classifier assumes that the effect of a predictor (x) over a given output class (y) is independent of the value of other predictors.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 2.3: Bayes' Theorem [9]

P(c|x): posterior probability of the target(c) given a predictor (x)

P(x|c): probability of a predictor given a class(c)

P(c): prior probability of a class

P(x): prior probability of a predictor

Otherwise called class contingent freedom, the previously mentioned supposition that is frequently observed as a downside in the exactness of the calculation. Various heuristics intended to deal with this challenge will be introduced in the Implementation part. To embody the equation, think about the past model with pictures of felines and canines. In this case, the classifier needs to choose two classes. Characterizing another picture is the likeness looking at between the two probabilities: P(cats | new_image) and P(dogs | new_image). These qualities will be registered dependent on the above recipe. In the calculation, the pictures utilized in the preparation period of the calculation will be utilized to register the likelihood of the two classes, the earlier likelihood of the indicators, and the likelihood of an indicator given the two classes.

2.3.2 Support Vector Machine (SVM)

Backing Vector Machine is another illustration of directed calculation. While Naïve Bayes employments a probabilistic model, SVM is utilized to plan the contribution to a high dimensional plane. A line speaking to the hyperplane will isolate the classes, with the goal that the distance between the classes and the hyperplane is augmented. At the point when new models are added to the model, the calculation yields an ideal hyperplane to arrange them. Figure 2.4 is an illustration of a SVM calculation hyperplane [27].

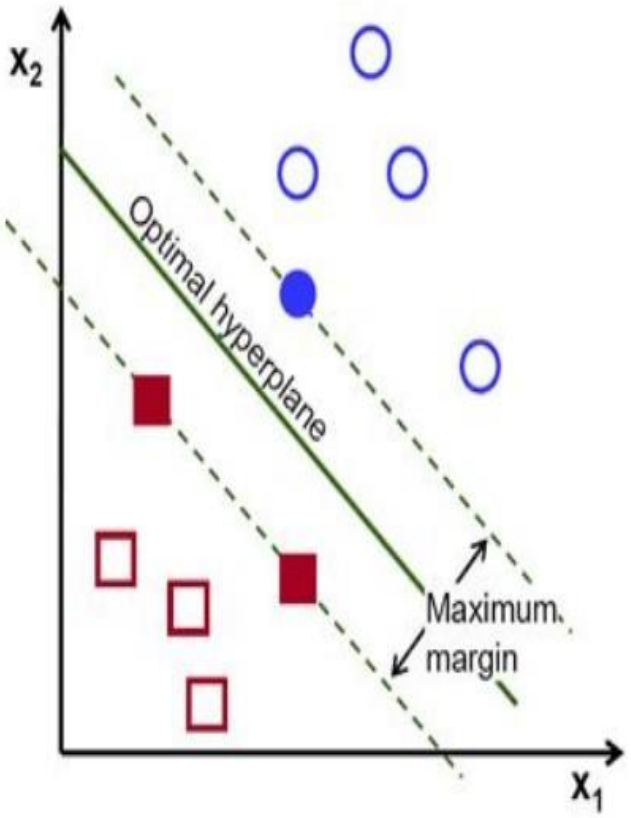


Figure 2.4: SVM - two classes example [10]

When no such line can be established, a kernel function is used to map the data into a higher dimensional plane, where the data can be separated by its component classes. When visualised in a lower dimensional plane, the line separating the classes is curved.

CHAPTER 3

DESIGN

Overview: This chapter describes the design patterns used during the development of the project. It presents a high level view of the system, what methodology was applied, and which third party technologies have been adopted to deliver the final product. More details about implementation will be demonstrated in Chapter 4.

3.1 Why Python?

Given the setting of preparing a lot of information, memory the board became prioritary. Therefore, a programming language which is skilled to deal with preparing and capacity of these measures of information was basic. An iterative framework approach, when preparing a rundown of things, needs to initially store it, which requires memory. In such matters, Python gives generators, which are especially valuable when preparing a lot of information, passing the source information through the preparing chain, each thing in turn, putting away as it were the aftereffects of the preparing chain [11]. Thinking about the above contention, Python demonstrates prepared to do productively overseeing memory, a task pivotal for the 'continuous' part of the undertaking. A disadvantage of Python is that it is an deciphered language, which paradoxically, is more slow than incorporated dialects, (for example, C, Java, and so forth) The engineers network thought about the inconvenience, and proposed various ways to improve Python's speed. Thusly, ventures like Numba and PyPy are suitable arrangements. The maker of Python, Guido van Rossum perceives the upgrades added to Python and states that PyPy is the most ideal approach to acquire superior frameworks while utilizing Python [12]. Besides, progressed libraries for information preparing, for example, NumPy and SciPy were created by established researchers and area specialists [29]. Such devices end up being supportive during the improvement stage, and strengthened the thinking of picking Python.

3.2 System Architecture

In order to perform sentiment analysis, data manipulation is required via a processing chain. On this matter, in the early stage of the project, a support module was developed. The support module, referred from now on as the pipeline, had to be capable of integrating and testing the following components:

1. Data Gathering Modules
2. Data Filtering Modules
3. Association Rules Modules
4. Sentiment Classification Modules

Together, the pipeline and the aforementioned components form the engine of the system. One of the early objectives of the project was developing a working model of the engine. As a consequence, the user interface was produced in later development stages. The architectural overview is illustrated in Figure 3.1. The following subsections will cover in detail the design and the requirements gathering for each component of the system.

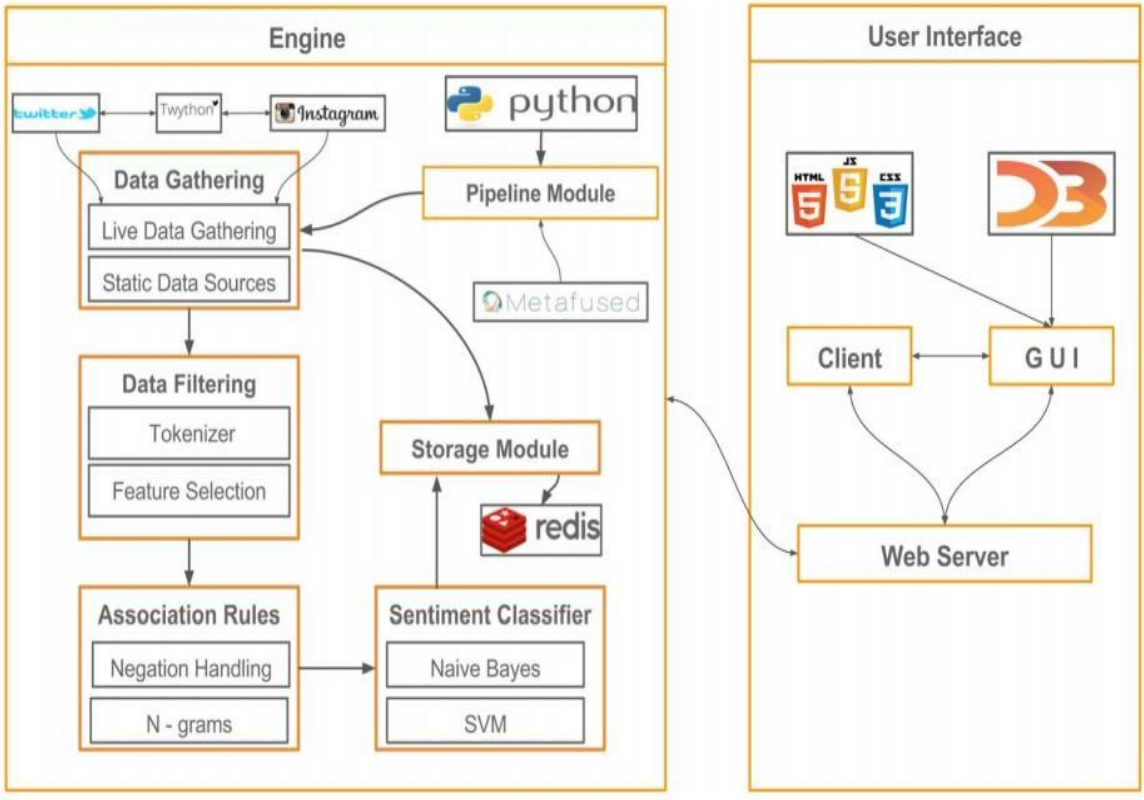


Figure 3.1: Architecture overview

3.3 Methodology

The task's plan and improvement stages were finished in little emphasess. This is a center part in thedeft procedure which was broadly applied all through the undertaking. The time accessible foradvancement was separated into more modest pieces looking like the emphasess, each havingacutoff time and a bunch of assignments to be finished [15]. A smaller than expected arrangement covering a rundown for every cycle can be found in Appendix A. Besides, GitHub was utilized as a rendition control stage. Toward the finish of every cycle a new branch containing the undertakings done was made. In the event that the code breezed through the assessments, the current branch was converged with the expert branch. In the event that the code didn't finish the assessments, the code remained unmerged until blunders/clashes were settled. To assist with the advancement following, 16 JIRA, an outsider programming offering a usage of the "Assignments Board" spry practice, was utilized.

3.4 Requirements gathering

Necessities, both useful and non-useful, were accumulated at various stages during the advancement cycle, in opposition to a traditional in advance methodology. Nonetheless, characterizing a base set of prerequisites and expected conduct was important to start improvement. In that capacity, the motor must be versatile, permitting mix of new modules without influencing its execution. Moreover, a standard structure of the prepared information was characterized, utilizing JSON (JavaScript Object Notation) arranging, where information objects were put away utilizing characteristic worth sets. A non-practical necessity before the advancement stage, was exploring the condition of the craftsmanship in slant examination. From this viewpoint the precision of the calculations utilized for arrangement, the adaptability of executing heuristics on such calculations, and the time needed for execution and testing were surveyed. For instance, a framework utilizing Neural Organizations requires more opportunity for execution contrasted with one where Naïve Bayes is utilized. Moreover, Neural Network calculations need inside and out comprehension to perform heuristics, while a probabilistic model like the Naïve Bayes calculation is more adaptable [13]. More subtleties on how directed AI calculations are contrasted with one another in particular writing will be introduced in Chapter 5 .

3.4.1 Engine Requirements

The majority of the necessities for the motor are of a useful sort. Consequently, heuristics on improving the AI calculation were needed, just as the utilization of outsider programming. An underlying rundown of utilitarian necessities positioned by need and number of hours needed for fruition is introduced in Table 3.1. To accomplish the continuous conduct wanted, a solid Twitter stream programming must be utilized in the improvement of the motor, as Twitter doesn't give a local Python API. In these respects, a "unadulterated Python covering for Twitter API" [14] - Twython was utilized. As there are various Python coverings accessible for Twitter's API (e.g: Tweepy, Twitter

Search, Birdy, and so forth), the component which made Twython most engaging was the capacity of supporting a multi-strung usage. Along these lines, the application bolsters more than each client in turn, and furthermore to deal with numerous solicitations from a similar client without having to run various cases of the motor [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 important design problem was data storage. Here, two database architectures were considered: a lightweight database - Redis - using a file system stored in RAM, and a relational database - PostgreSQL. Furthermore, the engine has to support both the real time component, and the long-term sentiment analysis component. Consequently, Redis was the first choice, as it proved to be an efficient solution for both temporary and permanent storage, being particularly useful at caching operations in the long-term component. A more detailed list of non-functional requirements is outlined 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)

As one of the main objective of the project is performing highly accurate sentiment analysis, the graphical user interface was not within the initial scope. However, a graphical user interface turned out to be imperative, such that users can interact with the engine in a more intuitive manner. As such, a set of functional requirements was developed in the GUI as evidenced 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

The non-functional requirements for the graphical user interface were set following three principles: responsiveness, readability, and usability. As such, Table 3.4 presents the non-functional requirements developed for the graphical user interface.

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

An important challenge was to make the interface robust, so that a user can follow the sentiment evolution of two or more topics at the same time. This was accomplished using Flask - a third party micro framework in Python, which offers a build-in development server with restful request dispatching [20]. As such, every time the user places a request for a topic, a new URL which contains the name of the topic is generated. For example, if the topics of interest are “Brand A” and “Brand B”, two charts will be displayed in the same window, resembling real time sentiment analysis for those, with the following URL address: “http://localhost:8000/Brand_A+Brand_B”

CHAPTER 4
Implementation

Overview: This chapter follows the approach described in the Design section, making a clear distinction between the back-end (the engine of the system) and front-end (the graphical user interface). More in depth technical details will be presented for both of the components, as well as how third party software was employed in development.

4.1 Engine

The engine was built using “the pipeline” (introduced in Chapter 3) and it had to adapt for two purposes. One is to serve the real time interface (described in future subsections). The second purpose is to be run continuously in a pre-established interval of time in order to perform in depth sentiment analysis for a spectrum of brands and topics. This is reflected in the separation in two modules of the data gathering component.

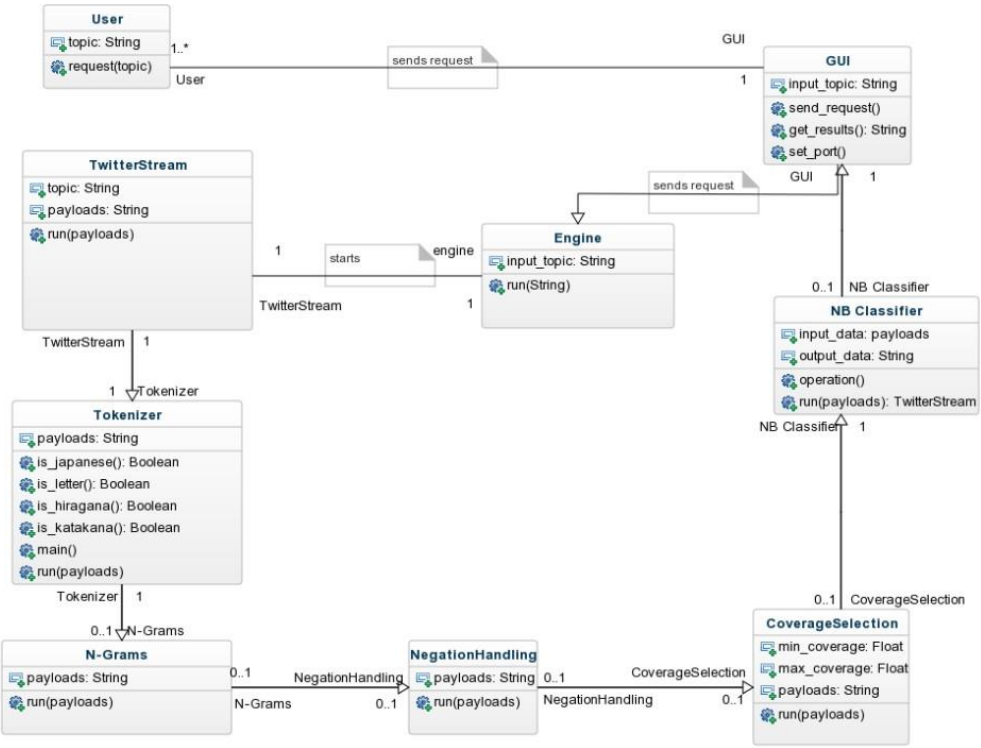


Figure 4.1: Real time component class diagram representation

The engine uses the pipeline to add a series of functionalities to the processing chain. The following components build up the final version of the engine:

1. Data Gathering Modules
2. Data Filtering Modules
3. Sentiment Classification Module
4. Association Rules Modules

The rest of this subsection will be focused on explaining the implementation steps and technicalities of both, the pipeline and the aforementioned modules. A simplified class diagram of the real time system is presented in Figure 4.1.

4.1.1 The Pipeline

The pipeline needed to give a climate prepared to deal with information, while incorporating the other

modules. Moreover, the pipeline must be adaptable with the goal that it could undoubtedly adjust information streams. One normal issue when handling information streams with assistant utilization of information structures, is that the arbitrary access memory space accessible may be surpassed. This can lead to framework disappointments which are least attractive. Applicable to this issue, was an inherent component of Python: capacities generators. Rather than restoring a bunch of qualities put away in an information structure, a capacity can be composed so it creates the qualities as they are required by other parts in the handling chain [30]. While utilizing generators presents a bunch of preferences, there is one disadvantage to be referenced. Whenever information was created for utilization it must be called once. Not as terrible as it sounds, this obliged the code to be written in a cautious way by maintaining a strategic distance from redundancies, helper factors and information structures, to protect the proficient memory use. Another significant component of the pipeline is to guarantee the appropriateness of an item for the reason it was at first intended for. From this viewpoint, the pipeline needed to serve information structures starting with one module then onto the next, while guaranteeing that those items saved their properties.The outcome of the pipeline is a class which once started up fills in as help for the different modules in the preparing chain. Figure 4.2 is an illustration of the pipeline utilization and how modules are added to it. The initial step is to make an example of the pipeline class. Once done, this occurrence can consolidate different modules (e.g.: line 7 to line 12 in the code scrap). After the modules are added utilizing the add_step strategy, the run technique is approached the pipeline to begin the handling chain.

```
#Coverage Selection Parameters
min_cov = float(sys.argv[1])
max_cov = float(sys.argv[2])

#Training
pipeline = MFPipeline()
engine = pipeline.add_step(MFTwitterSentiment140Loader('training_data'))
engine = pipeline.add_step(MFTokenize5())
engine = pipeline.add_step(MFTokenAddBigrams())
engine = pipeline.add_step(MFNegationHandling())
engine = pipeline.add_step(MFCoverageSelectionTool(min_cov, max_cov))
engine = pipeline.add_step(MFSentimentTrainModel('trained_model.csv'))

#Run
pipeline.run()
```

Figure 4.2: Pipeline experiment - Naïve Bayes training phase

4.1.2 Data Gathering

The fundamental information source utilized all through the venture is the online interpersonal interaction administration, Twitter. Two information gathering modules were actualized utilizing Twython (presented in Chapter 3). One segment was intended to be utilized for the ongoing interface. When a solicitation is set, it will additionally send the solicitation to Twython, which thus will answer with a nonstop stream of tweets. The other segment gathers information inside a pre-set up time frame. As advertising is worried about turn of events and usage of a special technique, the time factor

must be painstakingly viewed as when performing investigation. Ten minutes worth of information on a mission probably won't legitimize the achievement pace of it. Considering the measure of time accessible for the improvement of the undertaking, holding up a whole week to perform examination over a bunch of information was not attainable by the same token. To meet the necessities, while having sufficient opportunity to test what's more, change the conclusion examination measure, the subsequent module was utilized to store information inside the accompanying stretches: ten minutes, 60 minutes, and two days. Thusly, changes of other parts is conceivable in a proficient measure of time, while the measure of information dissected will convey an exact achievement assessment of a mission. Besides, a helper module was created to arrange the marked information acquired from a

outsider source - "Sentiment140" [16]

4.1.3 Data Filtering I

For information sifting one module was fabricated, which performs tokenization parting literary contribution to tokens (words). The tokenization module actualizes assistant strategies to distinguish and dispose of tweets which are not completely made out of English characters (e.g.: Japanese, Hiragana, Katakana, and so forth) Nonetheless, tweets containing different images (e.g.: specifies set apart with the image "@" and hashtags, set apart with the image "#") are sifted in a different JSON field.

4.2 Classification - Naïve Bayes Algorithm

Whenever information was procured and sifted, the subsequent stage was the execution of the grouping calculation, the primary module of the motor. As illustrated in Section 3, the module executed the Naïve Bayes calculation, which comprises of two stages: preparing and testing. In the preparation stage, marked information gained from Sentiment140 was utilized. Accordingly, 1.2 million tweets out of the 1.6 million size of the corpus was utilized in preparing. The yield is a model comprising of token-esteem sets, where the qualities are: number of events over the whole dataset, number of events in certain named tweets, and number of events in negative marked tweets. This is exemplified in Table 4.1.

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 testing stage, concealed named information and the probabilistic model created in the preparation stage were utilized to gauge the precision of the calculation. Therefore, the calculation created a characterization precision of 71%. For the advancement stage when it was actualized, such precision was normal, yet further enhancements were required. 24 As the informational index utilized in preparing had marks just for positive and negative tweets, yet the objective is to recognize the three: positive, negative, impartial, further heuristic was required. In expansion, an estimation assessment scale: - 1 (negative) to 1 (positive) was presented, where tweets of which slant score was in the scope of: [-0.05, 0.05] were viewed as unbiased. The constraints of the reach were set up after a manual examination of the tweets investigated. The accompanying subsection portrays the execution of heuristics to improve the execution and the characterization exactness of the calculation.

4.2.1 Data filtering II

Later in the development, as a solution to improve the accuracy of the classifier, another filtering module was build to be used in the training stage of the classification. The problem identified was the large number of tokens produced by the tokenization module, which equaled almost 1 million.

This made the model to be computationally expensive, reducing the speed performance of the algorithm. In addition, two approaches were considered:

1. Using Porter’s Stemming algorithm to perform word reductions.
2. Excluding tokens with low sentiment value.

While Porter's calculation decreased the corpus size to 60% of its underlying size after tokenization, this was not adequate, as a critical number of the tokens left, held no specific opinion esteem. Checking physically which tokens were common in certain or negative settings was definitely not an answer, because of the high measure of manual work. A fascinating new heuristics utilizing the pareto standard (clarified in Chapter 2) was applied. Taking a gander at the tokens dissemination of the preparing model - Figure 4.3, it tends to be seen that words, for example, "I" and "the" have the most elevated event all through the entire information. At the furthest edge, are words which though, they may introduce a type of assumption esteem, the quantity of all out events in the corpus is low, subsequently the effect over the module is practically immaterial. Be that as it may, when isolated from their specific situation, those words don't hold any assessment esteem. Conversely, words arranged on the bend's incline in the figure, seemed to have a solid feeling esteem (e.g.: "love", "scorn", "astonishing" and so forth) Following the pareto standard, the corpus was sliced to 20% of its size, so just the tokens with a high opinion esteem were kept. A manual examination was needed to figure out where tokens with low feeling esteem begin to show up in the conveyance, to decide the cut focuses. Therefore, the last size of the model was decreased down to 40.000 tokens. Subsequently, the exactness was improved by 7% because of commotion (undesirable information) expulsion.

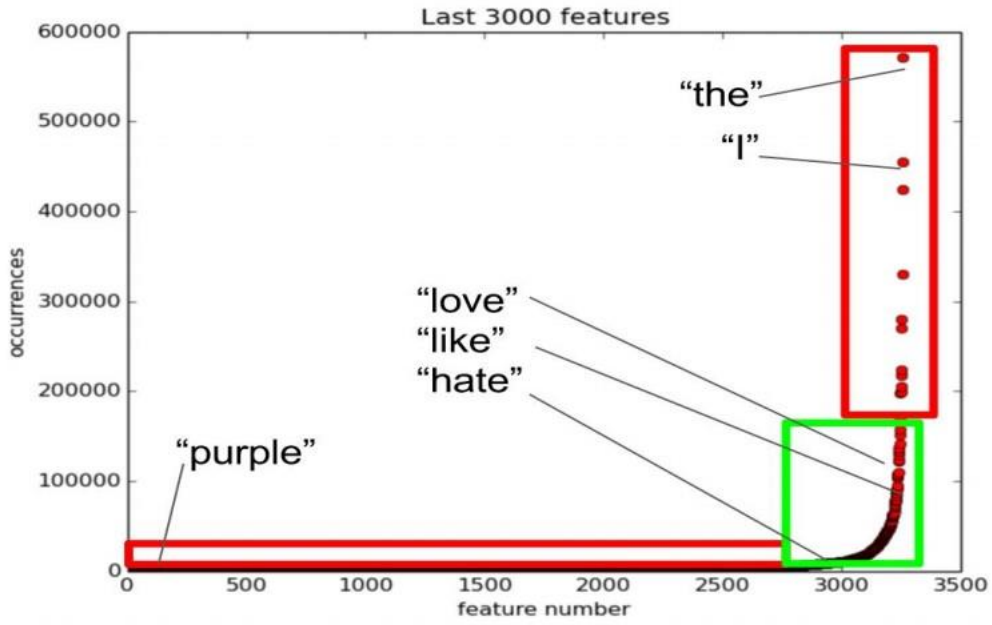


Figure 4.3 : Tokens distribution - Naïve Bayes model

4.2.2 Association Rules

Another normal improvement in computational phonetics is the utilization of n-grams (presented in Chapter 2) for which a module was made. Nonetheless, executing n-grams as bigrams and trigrams, without thinking about just significant affiliations, expands the size of the corpus and furthermore adds clamor to it. In that capacity, relationship between things, modifiers and action words were finished utilizing an outsider programming, multilingual, which offered a grammatical form labeling device [18]. Because of adding bigrams the general exactness of the classifier developed by 2.9% up to 80.9%. Notwithstanding n-grams, a different module was work to deal with refutation. A traditional methodology in particular writing is to add fake words: "if a word x is gone before by a nullification word (e.g: not, don't, no), at that point instead of thinking about this as an event of the component x, another highlight (token) NOT x is made" [4]. For instance subsequent to performing invalidation taking care of the sentence: "I don't care for this new brand" will bring about the accompanying portrayal: "I don't NOT_like, NOT_this, NOT_new, NOT_brand". The upside of this element is that the plain event and the discredited event of the word are presently discernable in the corpus. In expansion, the precision of the classifier was improved by 1.8% up to 82.71% generally exactness

(more subtleties are introduced in Chapter 5).

4.3 Graphical User Interface

As described in Chapter 3, the graphical user interface was implemented for the real time component. In order to add interactivity to the system, the language of choice to build the front-end was JavaScript. Consequently, a third party library - D3JS, was used to build interactive charts. Albeit there is a variety of templates already available within the library, an implementation from scratch was necessary to meet the requirements of the system. As such, the visual component was challenging due to the lack of previous experience in JavaScript (more details in the Conclusion chapter). Captures of the graphical user interface are presented in Appendix B.

CHAPTER5
CONCLUSION

Overview: This is a reflective chapter, where the completion of the objectives set at the start of the project is assessed. In addition, future work is proposed and the knowledge gained while working on the project is demonstrated.

5.1 Objectives and Timing

The fundamental target set for the venture was to build up a framework which can performcontinuous assessment examination via web-based media sources. As confirmed in the Evaluation section, the motor utilized in the framework was utilized in the advertising business, to screen and report supposition examination on a progression of points. From a planning point of view the improvement of the motor in December 2015, which is in accordance with the course of events set (introduced in Appendix A). Moreover, the constant segment of the framework was created and regular language preparing strategies (tokenization, stemming, grammatical form labeling, n-grams) were utilized to improve the exactness of the motor. Thus, the assessment set the grouping exactness at 82.71%, which outperformed the underlying goal. Yet, the improvement of the graphical UI for the constant part was set as a medium need objective, it was viewed as basic for the client's experience. As such, clients can collaborate with the motor in a more natural way, while their agreement is increased with visual portrayals of the information examined. This goal was met as part of the last achievement in the undertaking.

5.2 Future Work

Future work to upgrade the abilities of the motor delivered incorporates the improvement of a more perplexing graphical interface. Thusly, when the client inputs a point, the interface will likewise show the estimation characterization performed on past information, as a subsequent measurement. Another future improvement depends on the usage of a module to recover the substance of the URLs found in the tweets investigated. While investigating the corpus, a critical number of URLs which highlight Instagram - a web-based media stage - were found. Other than Instagram, the motor will incorporate modules to remove data from other web-based media stages: Facebook, which gives a complete Python API, and YouTube. Moreover, a more intricate range of feeling esteems will be presented (e.g., euphoria, outrage, trust, and so on)

5.3 Reflection and Knowledge Gained

By planning and building up this task, my general information and aptitudes in PC science were altogether improved. All things considered, a superior comprehension of the AI field was picked up, while executing the arrangement calculation and the resulting heuristics to improve its exactness. What's more, crucial information in the field of regular language handling was procured. Moreover, an expansion to the arrangement of programming dialects I ran over is JavaScript. I am sure now, that I can additionally upgrade cooperation among clients and future frameworks created, by depending not just on the inherent estimation of the ancient rarities delivered, yet in addition on the flexibility of JavaScript and the local visual systems accessible. Inside a similar note, the recently procured Python aptitudes were honed, by dominating progressed ideas, for example, generators and decorators [30]. From the point of view of the application area - web based advertising industry - various exercises were drawn. Conveying the product in industry, has demonstrated that a notion investigation instrument is major for the accomplishment of an organization's showcasing effort, yet a more explicit, theme arranged, item is attractive. Therefore, if I somehow happened to begin the venture once more, I would consider a more exploratory approach, to build up what are the prioritary themes on which opinion investigation will be performed. Be that as it may, the expectations would incorporate comparative functionalities. From an usage point of view, the advantage of building up the grouping calculation from scratch was reflected by the full power over the motor's capacities. The pipeline delivered was useful to keep a coordinated way of composing code and a helpful investigating apparatus. Notwithstanding, a second cycle of the venture will utilize a more top to bottom examination of the calculations accessible for characterization, deprioritizing the pipeline for later phases of the advancement.

5.4 Conclusion

Thinking about the above reflection on the undertaking's arranging, it tends to be inferred that the degree of the venture was met inside the course of events set. Accordingly, the ideal grouping exactness of 80% or more is reached. The announcing is acted progressively and the graphical client interface offers a natural, yet thorough client cooperation. Generally, the task was a great occasion to additional sharpen my programming aptitudes and grow my insight in the fields of regular language preparing and AI.

REFERENCES

[1]. SisiraNeti, 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. Vojnotehnickiglasnik, 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]. Codefellows.org. (2016). 5 Reasons why Python is Powerful Enough for Google. [online] Available: https://www.codefellows.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-falc on-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.lextutor.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