Sentiment Analysis of Political tweets using Naive Bayes Classifier

Thesis submitted to
Visvesvaraya National Institute of Technology,
Nagpur
In partial fulfillment of requirement for the
Award of degree of

Bachelor of Technology
in
Computer Science and Engineering

by

Chinmay Dorlikar	**BT15CSE026**
Gaurav Nagrale	**BT15CSE027**
Shreyas Dikshit	**BT15CSE082**

Under the guidance of
Dr. (Mrs.) Shital A. Raut

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

May, 2019

Sentiment Analysis of Political tweets
using Naive Bayes Classifier

Thesis submitted to
Visvesvaraya National Institute of Technology,
Nagpur
In partial fulfillment of requirement for the
Award of degree of

Bachelor of Technology
in
Computer Science and Engineering

by

Chinmay Dorlikar	**BT15CSE026**
Gaurav Nagrale	**BT15CSE027**
Shreyas Dikshit	**BT15CSE082**

Under the guidance of

Dr. (Mrs.) Shital A. Raut

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

May, 2019

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology, Nagpur.

Declaration

We , Mr. Chinmay Dorlikar, Mr. Gaurav Nagral e and Mr. Shreyas Dikshit, hereby declare that this project work titled "Sentiment Analysis of Political tweets using Naive Bayes Classifier" is carried out by us in the Department of Computer Science and Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution / University.

Sr. No.	Enrollment No.	Names	Signature
1	BT15CSE026	Chinmay Dorlikar	
2	BT15CSE027	Gaurav Nagrale	
3	BT15CSE082	Shreyas Dikshit	

Date:

Certificate

This is to certify that the project titled "Sentiment Analysis of Political tweets using Naive Bayes Classifier", submitted by **Mr. Chinmay Dorlikar, Mr. Gaurav Nagrale** and **Mr. Shreyas Dikshit** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

Prof. U. A. Deshpande
Head,
Computer Science and Engg. Dept.,
VNIT, Nagpur.

Dr. (Mrs.) Shital A. Raut
Assistant Professor,
Computer Science and Engg Dept.,
VNIT, Nagpur.

Date:

Acknowledgement

We would like to express our sincere thanks and gratitude to our guide, Dr. (Mrs.) Shital A. Raut for guiding us thoughtfully and efficiently throughout this project, giving us an opportunity to work at our own pace, while giving us directions whenever necessary.

Next, we would like to thank the Department of Computer Science and Engineering for providing us an opportunity to work on this project and improve our theoretical as well as practical knowledge.

Last but not the least we would like to acknowledge the contribution of those who have constantly supported and encouraged us which helped in the successful completion of our project.

Mr. Chinmay Dorlikar (BT15CSE026)
Mr. Gaurav Nagrale (BT15CSE027)
Mr. Shreyas Dikshit (BT15CSE082)

ABSTRACT

Sentiment Analysis, the robotized extraction of articulations of positive or negative frames of mind from content has gotten significant consideration from scientists amid the previous decade. What's more, the notoriety of web clients has been developing quick parallel to rising advancements; that effectively utilize online review sites, social media and individual websites to express their assessments. They harbor constructive and contrary dispositions about individuals, associations, spots, occasions, and thoughts. The tools given by natural language processing and machine learning alongside different ways to deal with work with huge volumes of content, makes it conceivable to start extricating opinions from web based life. Promising outcomes has demonstrated that the methodology can be additionally created to cook business condition needs through opinion examination in online life. Individuals make decisions about their general surroundings when they are living in the general public. They make positive and negative opinions about individuals, items, spots and occasions. These sorts of frames of mind can be considered as sentiments. Sentiment Analysis is the investigation of mechanized strategies for separating opinions from composed dialects. Development of web based life has brought about a blast of freely accessible, client produced message on the World Wide Web. These information and data can possibly be used to give ongoing bits of knowledge into the slants of individuals. Sites, online discussions, remark areas on media locales and long range informal communication destinations, for example, Facebook and Twitter all can be considered as internet based life. These online networking can catch a large number of people groups' perspectives or informal. Correspondence and the accessibility of these constant feelings from individuals around the globe make a transformation in computational phonetics and interpersonal organization examination. Online networking is turning into an inexorably progressively significant wellspring of data for a venture. Then again individuals are all the more eager and glad to share the certainties about their lives, learning, encounters and considerations with the whole world through online networking like never before previously. They effectively take an interest in occasions by communicating their feelings and expressing their remarks that happen in the public eye. Along these lines of imparting their insight and feelings to society and web-based social networking drives the organizations to gather more data about their organizations, items and to realize how presumed they are among the general population and in this manner take choices to go on with their organizations viably. In this project, we talk about a portion of the difficulties in opinion extraction, a portion of the methodologies that have been taken to address these difficulties and our methodology that investigations sentiments from Twitter data."

LIST OF FIGURES

LIST OF TABLES

INDEX

CHAPTER 1
INTRODUCTION

Sentiment Analysis (or opinion mining) is the study of emotions conveyed from a piece of writing. The content is delegated positive, negative or impartial, contingent on the examination. These compositions can incorporate anything from a plain content report to a tweet. Sentiment Analysis utilizes semantic tones, hashtags, setting and so forth to dissect a given bit of substance and recognize the feeling passed on [1]. Sentiment Analysis has now included a large group of new courses through which organizations can break down activities across various channels and settle on information driven choices that aim at improving consumer satisfaction. Estimating social sentiment is a significant piece of any social media monitoring plan. It encourages you to comprehend what somebody behind an online article is feeling. Knowing the feeling behind an article can give significant setting to how you continue and react.

Why is sentiment analysis so important in today's world?

- **Provides audience insight -**
  Advertisers profit by having as much data about their audience as conceivable. Understanding your audience's responses to your articles causes you to plan deliberately for future campaigns and articles.
  Suppose, after the launch of a company's product, their social team notices that there is considerable amount of the negative sentiment about the product online. Since their team paid attention to this social sentiment, they can take action (removing) the product from stores, adjust a few features accordingly. Monitoring sentiment also helps the brand refrain from sharing any tone-deaf messaging.

- **Supports customer service -**
  Monitoring sentiment is crucial for customer service and administration. Users in these roles can screen disappointment and respond viably before additional negative notion spreads. This is particularly significant for B2C [business-to-consumer] associations that are assessed on the web.
  This serves another purpose as well—you have an extraordinary opportunity to transform a terrible client experience into a positive one.

- **Informs brand messaging -**
  Public relations and corporate communications experts should know about brand recognition. When you investigate a brand with a similar audience to yours, you're able to investigate what your intended interest group reacts to. Let’s assume there is a company that makes clothes. They might want to take a look at the social sentiment of brands like Gucci, Armani, or Versace to see what they’re doing to bring in positive reviews. And more significantly, what they're doing to get the negative sentiment. What types of online articles are audiences responding to? Are they happy? Sad? Serious? How can they emulate the content with positive sentiment on their social account? On the flip side: What types of posts receives negative feedback? How can you avoid this in your own articles? Sentiment analysis is also helpful when checking keywords. In addition to seeing what the general public has to say, you can find influencers and thought pioneers relevant to your industry.

- **Competitor monitoring -**
  There's a great deal to be gained from the competition, not so they can copy them but rather so you can consider more intelligent approaches to contact the audience they both are focusing on. They can use what they learn from monitoring the competition to improve their own plans in several ways:
  - Work smartly to improve their engagement if they see it dropping or holding steady while a competitor’s is increasing.
  - See how well do they know their audience and if they are talking to them in the right ways.
  - See if they’re implementing what they’re realizing from the competition and if they are doing a lot of marketing and getting a positive reaction.

Sentiment Analysis has moved past merely a fascinating, innovative impulse, and will soon become a basic device for all organizations of the modern technology age. Eventually, opinion mining will empower us to gather new experiences, comprehend our clients better, and enable our very own teams to work more effectively.

CHAPTER 2
LITERATURE SURVEY

In this chapter, we discuss about the scope, further importance and existing methodologies and algorithms to implement sentiment analysis on a piece of data.

2.1 What is an "opinion"?

Textual data can be extensively classified into: facts and opinions. Facts are objective articulations about something. Opinions are normally emotional articulations that portray individuals' sentiments toward a specific theme [2].

Sentiment Analysis, similarly as other NLP problems, can be modeled as a classification problem where two sub problems ought to be settled:

- Classifying a sentence as subjective or objective, known as subjectivity classification.
- Classifying a sentence as expressing a positive, negative or neutral opinion, known as polarity classification.

In an opinion, the entity the content talks about can be an object, its features. It could also be a product, an individual, a company, an event, or any issue. As an example, take a look at the opinion below:

"The battery life of this mobile is too good."

A positive opinion is expressed about a feature (battery life) of an entity (mobile).

2.1.1 Direct vs. Comparative Opinions

Direct opinions give an opinion about an entity directly, for e.g.:

"The battery life of mobile 'XYZ' is poor."

This direct opinion states a negative opinion about *camera 'XYZ'*.

In comparative opinions, the opinion is expressed by comparing an entity with another, for e.g.:

"The battery life of A is better than that of camera B."

Usually, comparative opinions express similarities or differences between entities using a comparative or superlative form of an adjective or adverb. In the previous example, there's a positive opinion about *mobile A* and, conversely, a negative opinion about *mobile B.*

2.1.2 Explicit vs. Implicit Opinions

An explicit opinion on an issue is an opinion expressed directly in a sentence. The following sentence expresses an explicit positive opinion:

“The camera of this phone is stupendous.”

An implicit opinion on a subject is not stated directly but can be understood by the overall sentiment of the sentence. The following sentence expresses an implicit negative opinion:

“The airpods stopped working in two days.”

Here, the airpods not working after two days implies that they were not of good quality and hence a negative opinion is generated about them.

Within implicit opinions we could include *metaphors* that may be the most difficult type of opinions to analyze as they include a lot of semantic information.

2.2 Scope of Sentiment Analysis

Sentiment analysis can be applied at various levels:

- **Document level** sentiment analysis is mainly used to summarize the overall opinion of a document.
- **Sentence level** sentiment analysis can be used when one sentence corresponds to a particular person or group of people. Gives us individual opinions.
- **Sub-sentence level** sentiment analysis obtains the sentiment of sub-expressions within a sentence.

2.3 Types of Sentiment Analysis

There are numerous sorts and kinds of sentiment analysis relying on the domain used and the purpose it serves. Sentiment analysis tools range from frameworks that focus on polarity (positive, negative, neutral) to frameworks that detect feelings and emotions (angry, happy, sad, etc.) or identify intentions (e.g. interested v. not interested) [2]. In the following section, we’ll look into a few important ones:

2.3.1 Fine-grained Sentiment Analysis

Sometimes you may be wanting to find more precise information about the level of polarity of the opinion, so instead of just talking about positive, neutral, or negative opinions, you could consider the following categories:

- Very positive
- Positive
- Neutral

- Negative
- Very negative

A few frameworks likewise give distinctive kinds of extremity by distinguishing if the positive or negative assumption is related with a specific inclination, for example, outrage, bitterness, or sadness (for example negative emotions) or satisfaction, love, or enthusiasm (i.e. positive sentiments).

2.3.2 Emotion detection

Emotion detection means identifying feelings like, happiness, sadness etc. A lot of such systems resort to lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms.

One of the drawbacks of falling back on vocabularies is that the manner in which individuals express their feelings fluctuates a great deal thus do the lexical things they use. A few words that would regularly express outrage like *"kill"* (e.g. "*your customer support is killing me"*) might also express happiness (e.g. *"you are killing it"*).

2.3.3 Aspect-based Sentiment Analysis

For the most part, while separating the assumptions in some subject, for example a thing, you may be keen on not just seeing if the audience is conversing with a positive, neutral, or negative tone about the thing, yet additionally which specific attributes or highlights of the thing the audience talk about. That's what aspect-based sentiment analysis is about. In our previous example:

"The battery life of this mobile is too short."

The above line is expressing a negative opinion about the mobile, but more, about the battery life, which is a particular feature of the mobile.

2.3.4 Intent Analysis

Intent analysis basically detects what people want to do with a text rather than what people say with that text. Look at the following examples:

"Your customer support is a disaster. I've been on hold for 20 minutes".
"I would like to know how to replace the parcel".
"Can you help me fill out this application form?"

An individual has no issues distinguishing the protest in the firstline, the inquiry in the second line, and the solicitation in the third line. In any case, machines can have a few issues to recognize those. Some of the times, the sentiment can be construed from the content, yet some of the times, deriving it requires some logical information.

2.3.5 Multilingual Sentiment Analysis

Multilingual Sentiment Analysis oversees perceiving the general feeling of substance when parts of the content are written in various dialects. It very well may be a troublesome errand. Normally, a great deal of preprocessing is required and that preprocessing utilizes various assets. A huge bit of these assets are accessible online"(e.g. feeling lexicons),"but numerous others must be made (for example deciphered corpora or noise discovery algos). The usage of the assets accessible requires a great deal of coding background and can take long to execute.

An option in contrast to that would be to distinguish language in writings naturally, at that point train a custom model for the language of your choice(if writings are not written in English), lastly, perform the analysis.

2.4 Advantages of sentiment analysis

It's estimated that 80% of the world's information is unstructured and not composed in a pre-defined manner. Most of this comes from text data, like emails, chats, social media, surveys, articles, and documents. These texts are usually difficult, time-consuming and expensive to analyze, understand, and sort through.

Sentiment analysis systems enables organizations to comprehend this tremendous sea of unstructured information via mechanizing business procedures, getting noteworthy bits of knowledge, and sparing hours of manual information preparing, at the end of the day, by making teams progressively effective.

Some of the advantages of sentiment analysis include the following:

- **Scalability -**
  one can't envision physically dealing with a large number of tweets client bolster discussions, or client audits. There's only a ton of information to process physically. Sentiment Analysis permits to process information in a proficient and savvy way.
- **Real-time analysis -**
  We can use sentiment analysis to understand critical information that allows awareness during specific scenarios in real-time. eg.: *"Is there an angry customer that is about to churn?"*
  A sentiment analysis system can help you immediately identify these kinds of situations and take action.
- **Consistent Criteria -**
  People don't watch clear criteria for assessing the assumption of a bit of content. It's evaluated that diverse individuals just concur around 60-65% of the occasions when making a decision about the notion for a specific bit of content. It's a subjective task which is vigorously impacted by personal encounters, contemplations, and convictions. By utilizing a collective supposition examination framework, organizations can apply the equivalent.

2.5 How does sentiment analysis work?

There are already existing implementation methods and algorithms to perform sentiment analysis on a piece of data. They can be broadly classified into the following:

2.5.1 Rule based/ Lexical based analysis -

This procedure is represented by the utilization of a word reference comprising pre-labeled dictionaries. The info content is changed over to tokens by the Tokenizer. Each new token encountered is then coordinated for the vocabulary in the word reference. On the off chance that there is a positive match, the score is added to the complete pool of score for the info content. For example in the event that "dramatic" is a positive match in the lexicon then the all out score of the content is increased. Generally the score is decremented or the word is labeled as negative.

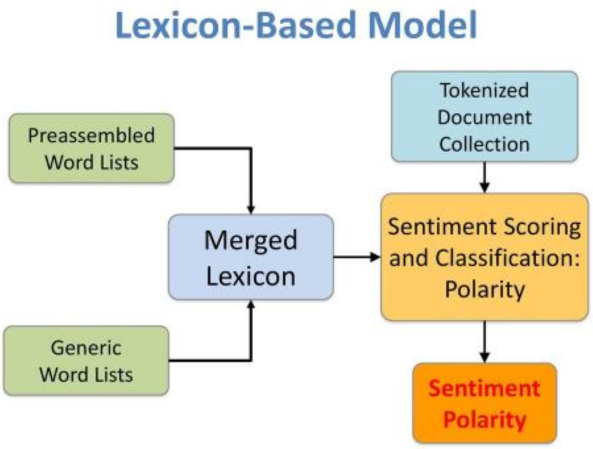


Fig. 2.1 Lexical-Based Model

The characterization of a content relies upon the all out score it accomplishes. Significant measure of work has been dedicated for estimating which best lexical data works. A precision of about 80% on single expressions can be accomplished by the utilization of hand labeled dictionaries included just descriptive words, which are pivotal for choosing the subjectivity of an evaluative content as exhibited by [3]. The author of [4] expanded this work utilizing same approach and tried a database of film reviews, reached a precision of 62%. Other than the hand labeled dictionary approach, [5] thought of a variation by using web crawler for denoting the extremity of words incorporated into work of [4]. They utilized two AltaVista web index questions: target word + "good‟ and other target word + "bad‟. The score was assessed by the inquiry that yielded the maximum number of hits, which answered to improve the earlier precision from 62% to 65%.

2.5.2 Machine learning based analysis -

Machine learning is a standout amongst the most prominent methods picking up enthusiasm of analysts because of its versatility and exactness. In assessment investigation, for the most part the regulated learning variations of this procedure are utilized. It includes three phases: Data collection, Pre-processing of the data, Training the data, Classification and plotting results.

A model is made dependent on the training data set which is utilized over the new/inconspicuous content for order reason. In machine learning method, the way to exactness of a classifier is the determination of fitting highlights. By and large, unigrams (single word phrases), bi-grams (two back to back expressions), tri-grams (three sequential expressions) are chosen as highlight vectors. There are an assortment of proposed includes in particular number of positive words, number of negative words, length of the report, Support Vector Machines (SVM) ([6], [7]), and Naïve Bayes (NB) calculation [8]. Exactness is accounted for to shift from 63% to 80% on the mix of different highlights chosen.

Following steps are involved in this technique:

1. **Data Collecting –** In this stage information to be investigated is slithered from different sources like Blogs, Social systems (Twitter, MySpace, and so forth.) depending on the territory of use.
2. **Pre-processing –** In this stage, the procured information is cleaned and made prepared for nourishing it into the classifier. Cleaning incorporates extraction of catchphrases and images. For example – Emoticons are the smiley utilized in literary structure to represent feelings for example ":-)", ":)", "=)", ":D", ":- (", ":(", "=(", ";(", and so forth.

   Revising the all capitalized and all lowercase to a typical case, evacuating the non-English (or proffered language writings), expelling pointless white spaces and tabs, and so on.
3. **Training Data –** A hand-labeled collection of information is set up by most ordinarily utilized publicly supporting strategy. This information is the fuel for the classifier; it will be nourished to the calculation for learning reason.
4. **Classification –** This is the core of the entire system. Depending on the prerequisite of the application SVM or Naïve Bayes is conveyed for investigation. The classifier (subsequent to finishing the training) is prepared to be conveyed to the constant tweets/content for sentiment extraction.
5. **Results –** Results are plotted dependent on the kind of representation chosen for example outlines, charts, and so forth. Performance tuning is done before the algorithm is released.

2.5.3 Hybrid analysis -

The progress in sentiment analysis tricked analysts to investigate the likelihood of a hybrid methodology which could collectively show the precision of an machine learning approach and the speed of lexical methodology. In [9] authors utilize two-word dictionaries and an unlabeled information, separating these two-word vocabularies in two discrete classes negative and positive. Pseudo archives incorporating every one of the words from the arrangement of picked dictionaries are made. At that point registered the cosine likeness among the pseudo archives and the unlabeled records. Depending on the proportion of comparability, the reports were either allocated a positive or a negative assumption. This preparation data set was then nourished to a Naive Bayes classifier for preparing reason. Another methodology introduced by [10], inferred a "brought together structure" utilizing foundation lexical data as word class affiliations. Creators restored data for specific zones utilizing the accessible data sets or preparing precedents and proposed a classifier called as Polling Multinomial Classifier (PMC) (otherwise called the multinomial Naive Bayes). Physically named information was fused for preparing reason. They asserted that creation utilization of lexical information improved execution. Another variation of this methodology was displayed by [11]. In any case, so far just [10] have had the capacity to guarantee great outcomes.

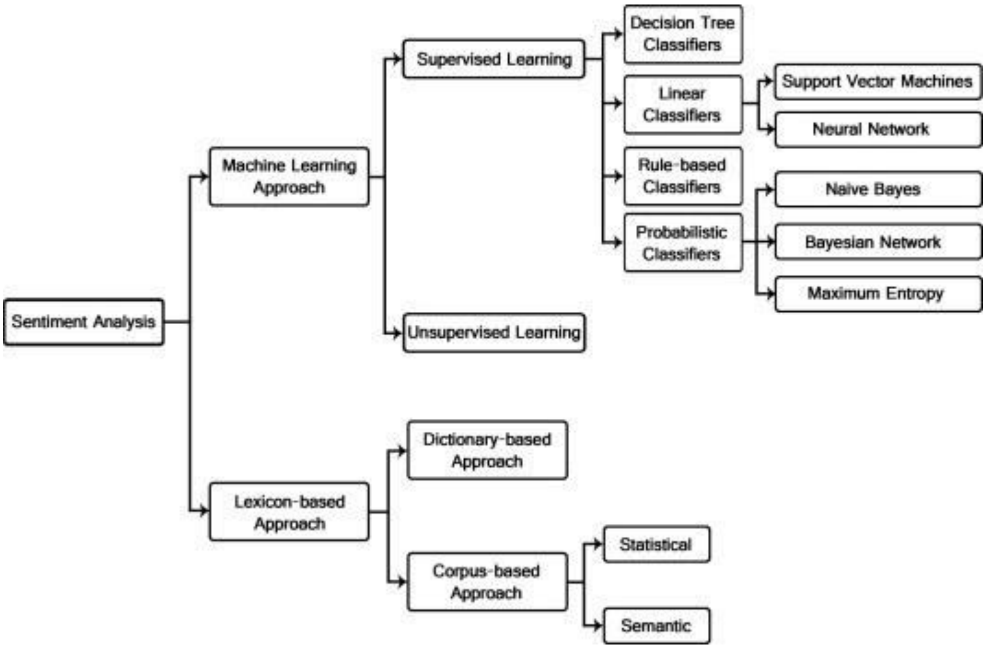


Fig. 2.2 Different approaches for Sentiment Analysis

We have picked a Naive Bayes approach i.e. a Machine learning based procedure for our model and subsequently would now talk about the different Machine learning approaches that exist.

2.6 Classification algorithms

2.6.1 Naive Bayes

Naive Bayes is a group of probabilistic calculations that exploit likelihood hypothesis and Bayes' Theorem to anticipate the tag of a content (like a bit of news or a client audit). They are probabilistic, which implies that they compute the likelihood of each tag for a given content, and after that yield the tag with the most elevated one ([8],[12]).

Anyway for this situation we don't have any numeric highlights. We simply have content, which we have to by one way or another proselyte into numbers that we can do calculations on.

In this manner, we use word frequencies. That is, we disregard word order and sentence development, regarding each record as a lot of the words it contains. Our highlights will be the checks of each one of these words. Despite the fact that it might appear to be too oversimplified a methodology, it works shockingly well.

Bayes’ Theorem is useful when working with conditional probabilities (like we are doing here), because it provides us with a way to reverse them:

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

So here comes the Naive part: we expect that each word in a sentence is free of the other ones. This implies we're no longer seeing whole sentences, yet rather at individual words. So for our motivations, "this was a fun gathering" is equivalent to "this party was fun" and "party fun was this".

Our model is based on this particular algorithm and hence it is explained in detail in a later chapter.

2.6.2 Support Vector Machines (SVM)

The nuts and bolts of Support Vector Machines and how it functions are best comprehended with a straightforward model ([6][7][13]). Let us envision that we have two labels: red and blue, and our information has two highlights: x and y. We need a classifier that, given a couple of (x,y) facilitates, yields if it's either red or blue. We plot our already labeled training data on a plane:

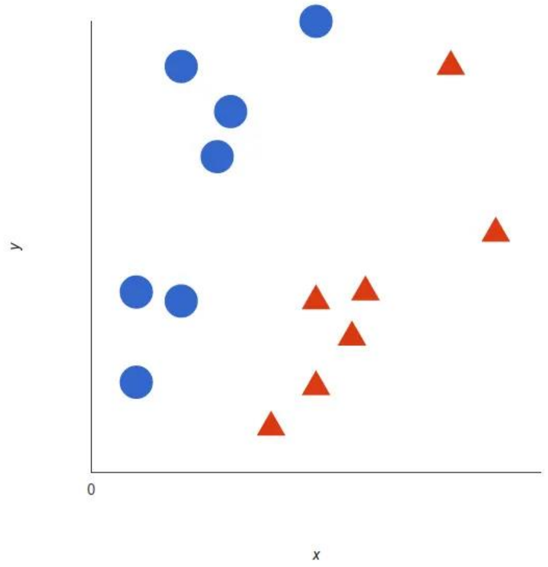


Fig. 2.3 Example of scattered data points

A support vector machine takes these information focuses and yields the hyperplane (which in two dimensions is just a line) that best isolates the labels. This line is the choice limit: whatever tumbles to the other side of it we will group as blue, and anything that tumbles to the next as red.

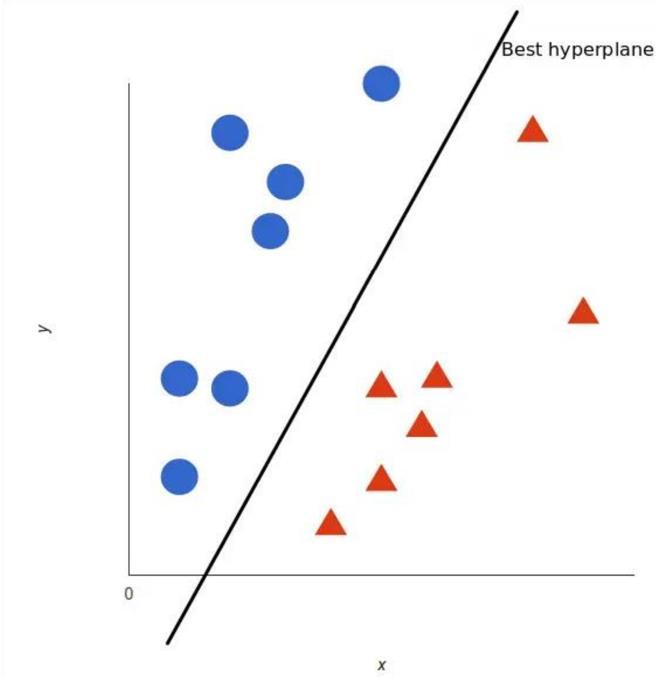
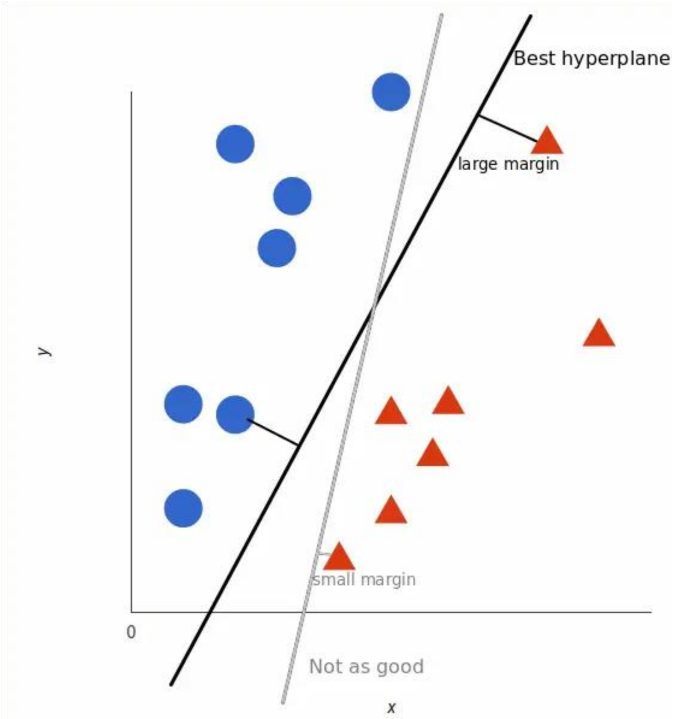


Fig. 2.4 Hyperplane across the scattered points

However, what can be defined as the best hyperplane? For SVM, the one maximizes the margins from the two labels. As it were: the hyperplane (recollect it's a line for this situation) whose separation to the closest component of each tag is the biggest.



Fig, 2.5 Best hyperplane in SVM

Just like we did in Naive Bayes, we take word frequencies as the feature. This implies we treat a content as a sack of words, and for each word that shows up in that pack we have an element. The estimation of that component will be the means by how frequent that word is in the content.

This technique comes down to simply tallying how often every word shows up in a content and dividing it by the all out number of words. So in the sentence “All monkeys are primates but not all primates are monkeys” the word monkeys has a frequency of 2/10 = 0.2, and the word but has a frequency of 1/10 = 0.1 .

After that is done, each text in our dataset is seen as a vector with thousands (or several thousands) of measurements, each one speaking to the recurrence one of the expressions of the content. This is the thing that we feed to SVM for training. We can improve this by utilizing pre-processing methods, such as stemming, removing stopwords, and utilizing n-grams.

Since we have the feature vectors, the main thing left to do is picking a kernel function for our model. Each issue is unique, and the piece work relies upon what the information resembles.

Back in our model, we had two highlights. Some genuine uses of SVM in different fields may utilize tens or even several highlights. In the interim, NLP classifiers utilize a huge number of highlights, since they can have up to one for each word that shows up in the preparation information. This progressions the issue a tad: while utilizing nonlinear pieces might be a smart thought in different cases, having this numerous highlights will finish up making nonlinear parts overfit the information. In this way, it's ideal to simply adhere to a decent old straight piece, which really results in the best execution in these cases.

Contrasted with more up to date calculations like neural systems, SVMs have two fundamental points of interest: higher speed and better execution with a set number of tests (in the thousands). This makes the calculation truly appropriate for text classification issues, where it's basic to approach a dataset of at most two or three thousand labeled samples.

2.6.3 Linear Regression

Linear regression is one of the strongest tools available in statistics and machine learning and can be used to predict some value (Y) given a set of traits or features (X). For eg:

What is your expected income from your years of education? What is expected final exam results given your previous marks?

To illustrate how linear regression works, we may examine a common problem students face when attending university. What is my expected final exam mark, given my previous results in the subject?

This problem can be mathematically defined as some function between our independent variables (X) and the corresponding final exam mark (Y).

$$\widehat{Y} = f(X) + \epsilon$$

where,

X (input) = Assignment Results
Y (output) = Final Exam Mark
f = function which describes the relationship between X and Y
e (epsilon) = Random error term (positive or negative) with a mean zero (there are move assumptions for our residuals, however we won't be covering them)

2.7 Sentiment Analysis Tools

There are numerous APIs that can be used for sentiment analysis of a textual data. Most of these APIs serve a general purpose of sentiment analysis. We will discuss a few such existing tools and their working here [14].

2.7.1 Hootsuite Insights

Rapidly and effectively filter mentions and sort by sentiment utilizing Hootsuite Insights. You can likewise follow estimation by catchphrases and set up computerized assignments by picked keywords.

For eg., you could set up your Twitter specifies on Hootsuite to examine for tweets containing positive terms, for example, "brilliant" "love" and "amazing" You can likewise make sure to look for slant flagging emoticons, for example, the approval or smiley face.

Hootsuite Insights furnishes a review of assessment with a simple-to-use meter. This enables you to rapidly perceive how your image is getting along from an sentiment perspective, and screen for any changes.

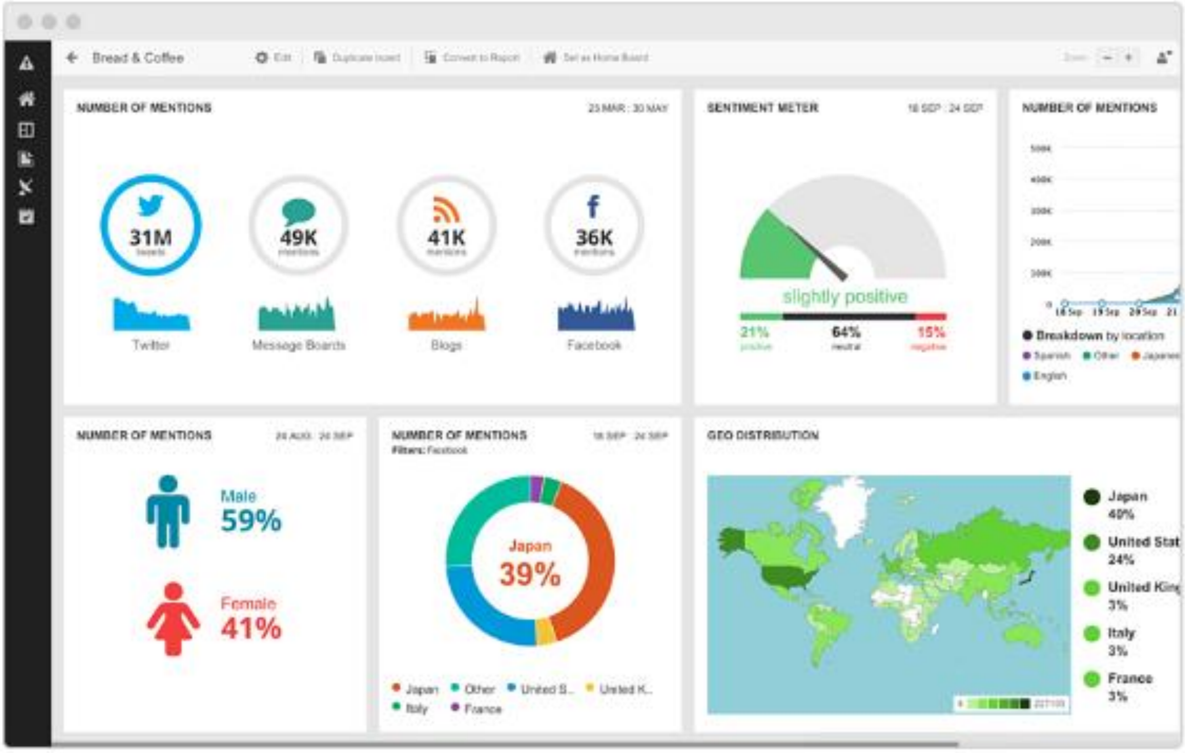
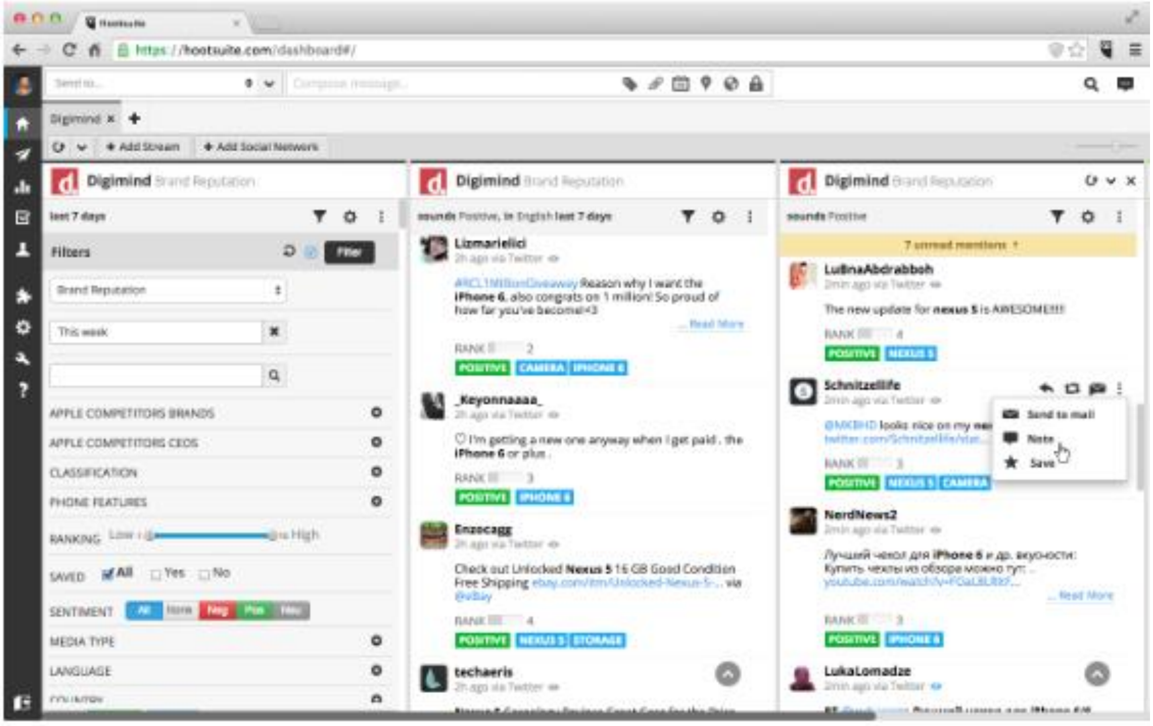


Fig. 2.6 HootSuite Insights

2.7.2 Digimind

Digimind causes you intently screen your web-based social networking nearness by recognizing and breaking down all the significant discussions about your image and competitors. It pulls data from in excess of 850 million web sources, so you realize you're getting a far reaching perspective on opinion toward your brand.

You can likewise examine makes reference to and apply channels to exceptionally tweak your notion investigation process. Digimind is additionally accessible inside the Hootsuite application Directory and can be added to your Hootsuite dashboard.



Lizmarielici
2h ago via Twitter
#RCL1MillionGiveaway Reason why I want the iPhone 6, also congrats on 1 million! So proud of how far you've become!<3
RANK 2

Keyonnaaaa
2h ago via Twitter
♡ I'm getting a new one anyway when I get paid , the iPhone 6 or plus .
RANK 3

Enzocagg
2h ago via Twitter
Check out Unlocked Nexus 5 16 GB Good Condition Free Shipping ebay.com/itm/Unlocked-Nexus-5-... via @eBay
RANK 4

techaeris
2h ago via Twitter

LuBnaAbdrabboh
2min ago via Twitter
The new update for nexus 5 is AWESOME!!!!
RANK 4

Schnitzellife
2min ago via Twitter
@MKBHD looks nice on my ne... twitter.com/Schnitzellife/stat...
RANK 3

NerdNews2
2min ago via Twitter
Лучший чехол для iPhone 6 и др. аксесуари: Купить чехлы из обзора можно тут: ... youtube.com/watch?v=...
RANK 3

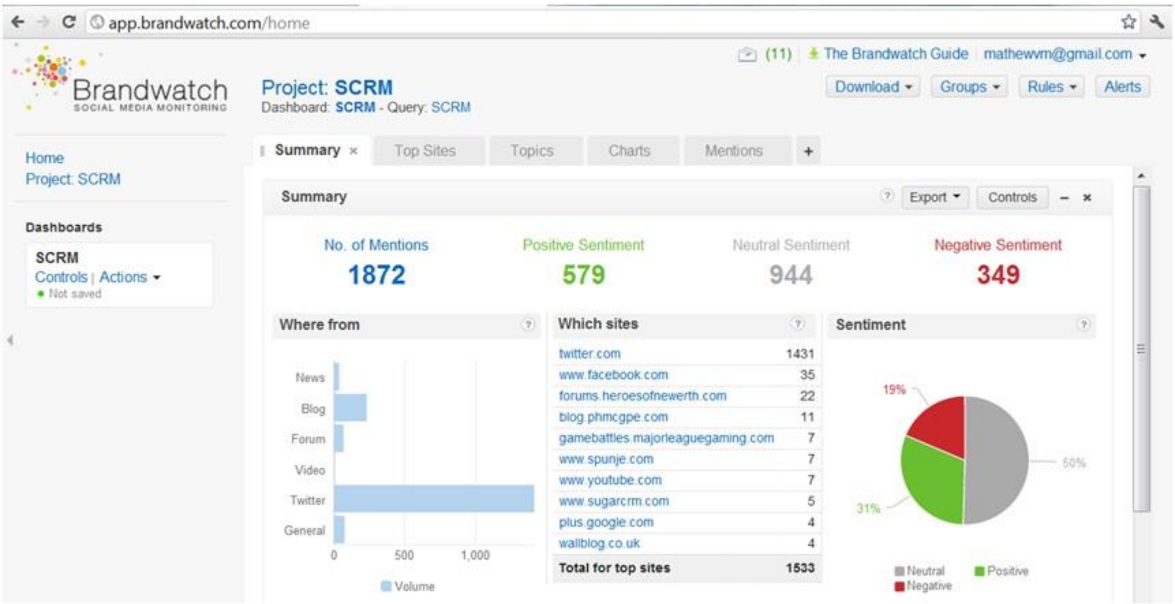
LukaLomadze
2min ago via Twitter

Fig. 2.7 Digimind

2.7.3 Brandwatch

Brandwatch was established and propelled by Giles Palmer in August 2007. Giles keeps on driving the organization at the situation of CEO, and is helped by Fabrice Retkowsky who is the Chief Technical Officer and Bryan Tookey who is the Chief Operating Officer. Based out of Brighton in the UK, Brandwatch has attempted consistent item improvement and is as of now in its 6th form.

Brandwatch gives clients access to bits of knowledge and information from more than 70 million traffic sources. Alongside commitment and observing capacities, you can rapidly observe the tone of posts and information being checked. This enables you to break down how certain substance performs, and the sort of responses it gets.



Summary

No. of Mentions	Positive Sentiment	Neutral Sentiment	Negative Sentiment
1872	579	944	349

Which sites	
twitter.com	1431
www.facebook.com	35
forums.heroesofnewerth.com	22
blog.phmcgpe.com	11
gamebattles.majorleaguegaming.com	7
www.spunje.com	7
www.youtube.com	7
www.sugarcrm.com	5
plus.google.com	4
wallblog.co.uk	4
Total for top sites	**1533**

Fig. 2.8 Brandwatch

2.7.4 Rapidminer

RapidMiner is an information science programming stage created by the organization of a similar name that gives an incorporated domain to information readiness, AI, profound learning, content mining, and prescient investigation. It is utilized for business and business applications just as for research, instruction, preparing, quick prototyping, and application improvement and supports all means of the AI procedure including information readiness, results representation, model approval and streamlining. RapidMiner is created on an open center model. Business valuing begins at $2,500 and is accessible from the developer.

Rapidminer utilizes a forte content mining way to deal with assistance brands lead supposition examination. With Rapidminer, unstructured substance sources, for example, online audits and web based life posts, are broke down, alongside organized sources, for example, official productions and reports. This enables you to recognize territories for business development and accumulate input from item dispatches. The more information you have about your group of onlookers and industry, the better shot of progress you have.

CHAPTER 3
PROJECT WORK

In this chapter, we discuss the project implementation in detail. The implementation includes three phases, Problem Statement, Proposed Approach and the Working model.

3.1 Problem Statement

3.1.1 Problem Introduction

The best businesses understand sentiments of its customers, the best political parties understand opinions of its people, the best agents understand sentiments of its clients - what they are saying, how they are saying it, what do they mean. Sentiment Analysis aims at understanding these emotions using an algorithm to computationally identify and categorize opinions in order to determine the writer's emotions towards a particular topic. Millions of people use micro-blogging services like Twitter to express their views and opinions on a day to day affairs. The existing approaches for Sentiment Analysis have proved to be essential and have achieved good results. But, almost all approaches have fallen prey in calculating the polarity of a sentiment and categorizing it into positive, negative and neutral, rather than finding the individual / organization to which the sentiment has been expressed. These approaches yield performance up to ~65 - 80 percent. We believe, using our approach, in addition to calculating the polarity, it also discovers the individual / organization the sentiment has been referred to, which results up to 70 percent can be realized (see chapter 4).

3.1.2 Problem Definition

To propose an approach to yield competitive results by analysing sentiment of users captured in an open social network Twitter using Bayes Theorem for exploring public opinion.

3.2 Proposed Approach

The superiority of Naive Bayes Classification is not unknown to the world. This conditional probabilistic model has worked quite well in many complex real-world situations. A comprehensive comparison of Naive Bayes Classifier with other classification algorithms shows that it outperformed many other approaches. Naive Bayes Classifier delivers good results in categorizing the sentiments, however, it doesn't categorizes the sentiment to a particular individual / organization.

Our goal is to propose an approach that not only gives the polarity of the sentiment i.e., the intensity of positive, negative and neutral values but also categorizes the sentiment on the basis of individual / organization to which it has been referred to. For this purpose, we choose to employ a Naive Bayes Classifier over a data labeled for a two party scenario (discussed in detail in section 3.2.1). Our model follows the following four steps namely: Data collection, Preprocessing, Training the Classifier and Classification. Through the following sections, we will discuss each step in detail, one at a time. The following figure (fig. 3.1) shows the system architecture of our approach.

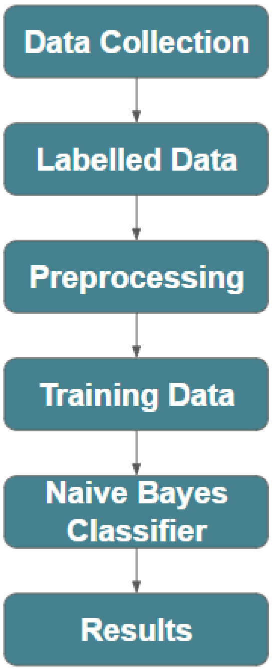


Fig. 3.1 System Architecture

3.2.1 Data Collection: Tweepy

For training the classifier, we are using Twitter data. For this purpose, we are using Tweepy to extract the data. Tweepy is an open-sourced tool that enables Python to communicate the Twitter platform and use its API. It is hosted on Github. For the demonstration of our model, we collect the tweets concerning the "US Presidential Election 2016".

3.2.2 Preprocessing

The tweets obtained from Twitter are a mixture of unwanted i.e., unsentimental data like hashtags #, annotations @, emoticons, links and retweets "RT". Preprocessing basically refers to the transformations applied to data before passing it to the classifier. We have tried using self-written function to preprocess our data. The function includes lowercasing the alphabets, removing non-required symbols, removing links and removing stop-words. Simple string operations and regular expressions have been used to achieve this. In information retrieval, it is a common tactic to remove very common words like "a", "an", "the", "about", etc.

since their appearance / removal doesn't deviate the sentiment of the tweet. The data has been handpicked to maintain it to one language only, English.

3.2.3 Training the data

Pre-labeled twitter training data is not available freely and it doesn't allow open / free sharing its contents, the data collected has been hand flagged in order to get the dataset in the required form. We plan to train the classifier using 80 percent of this pre flagged data and the rest of the data for testing and checking the accuracy.

Labeled data

Since we do not have direct access to pre-labeled Twitter data, we planned to crawl it manually. The data is labeled according to the following table 3.1:

Table 3.1 Labeling of data

Type of Tweet	Label
Tweet in favour of Hillary Clinton and Democratic Party (or) Tweet against Donald Trump and Republican Party	-1
Tweet in favour of Donald Trump and Republican Party (or) Tweet against Hillary Clinton and Democratic Party	+1
Neutral Tweet	0

The data has been labeled to provide a two-party scenario where, a sentiment in favour of one party is a sentiment against the other. Now, we feed this data to the classifier for training.

3.2.4 Sentiment Analysis: Naive Bayes Classifier

Naive Bayes is a probabilistic machine learning model that's used for classification purpose. The basis of the classifier is Bayes Theorem.

Bayes Theorem

Bayes Theorem or Bayes Law based on prior knowledge of conditions, describes the probability of an event. Bayes Theorem is mathematically represented in the following way:

$$\boldsymbol{P(A|B) = \frac{P(B|\,A)\,*\,P(A)}{P(B)}}$$

where,

A and B are events and P(B) ≠ 0.
P(A | B) is conditional probability defined as probability of A
given that B is true.
P(B | A) is conditional probability defined as probability of B given that A is true.
P(A) and P(B) are the independent probabilities of occurrence of corresponding events.

The process of how this equation has been used in the model is defined below.

To explain this concept, let's take an example. For an instance, we have a new tweet to be classified into one of the three defined classes. Since the new tweet's class is not known, the problem is to estimate correctly the class that the tweet is to be categorized in. The probability of a given Tweet belonging to a class can be determined using Bayes Rule.

$$\boldsymbol{P(class \mid tweet) = \frac{P(tweet|\,class)\,*\,P(class)}{P(tweet)}}$$

Defining the above equation in words states that the probability of a class given the tweet is conditional probability of tweet given the class (is true) multiplied by the probability of the class divided by the probability of the tweet. As P(Tweet) remains the same for the entire data i.e.,

$$\boldsymbol{P(tweet) = \frac{1}{Total\,number\,of\,tweets}}$$

we remove the denominator section of the eq. 3.2, thus resulting to the final equation stated below:

$$\boldsymbol{P(class \mid tweet) = P(tweet|\,class)\,*\,P(class)}$$

The conditional probability of a tweet given the class (is true) is defined as:

$$\boldsymbol{P(tweet|\,class) = \pi P(word \mid class)}$$

In words, conditional probability of a tweet for a given class is defined as the

product of posterior probability of each word given the class.
Now, the posterior probability of a word in a given class is defined as:

$$P(word \mid class) \ = \frac{Frequency\ of word\ in\ the\ given\ class +\ 1}{Total\ number\ of\ words\ in\ the\ given\ class}$$

If the word is not listed in the list of words belonging to the given class, the probability of that word becomes 1/(Total Number of words in the given class) so as to not make the probability zero resulting in the entire conditional probability to be zero, hence value “1” is called Laplacian Smoothing.

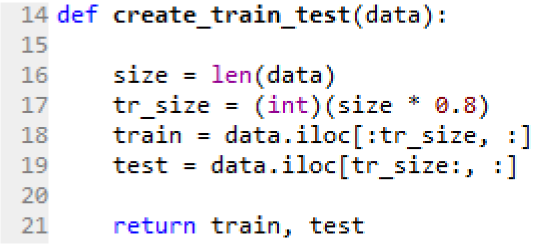
In this way, the probability of a given tweet will be calculated for each class. The polarity of the tweet, thus will be the maximum of the probability value among all the classes.

3.3 Working model

In this section, we will go through our working model and observe how various equations and methods discussed above are defined.

3.3.1 Training Data:

The partition of the labeled data into training and testing is shown in **Fig. 3.2**. 80 percent of the data has been used to train the model. The rest of the tweets are used to check the accuracy.

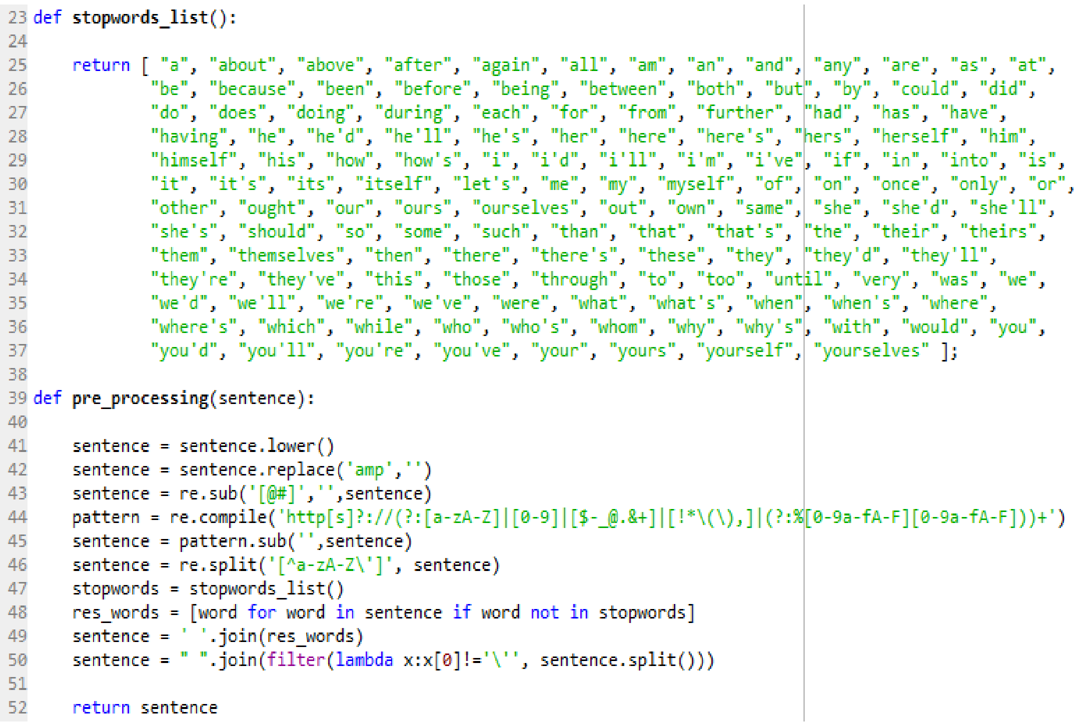


```
def create_train_test(data):

    size = len(data)
    tr_size = (int)(size * 0.8)
    train = data.iloc[:tr_size, :]
    test = data.iloc[tr_size:, :]

    return train, test
```

Fig. 3.2 Partitioning of data

3.3.2 Preprocessing:

Processing of the data is carried out using self-defined function shown in **Fig. 3.3.** A list of stop-words is maintained to remove the unwanted and non-sentimental words from the data. This process involves essentially removing these stop words and https tags.



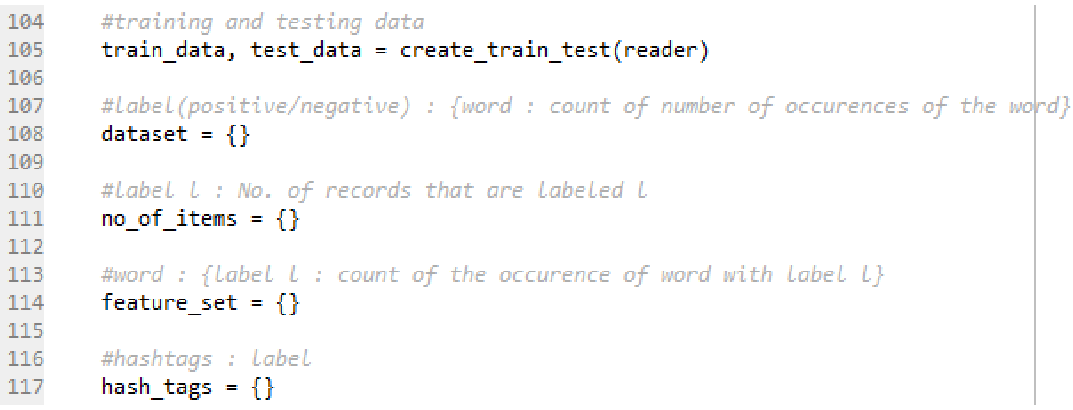
```python
def stopwords_list():

    return [ "a", "about", "above", "after", "again", "all", "am", "an", "and", "any", "are", "as", "at",
             "be", "because", "been", "before", "being", "between", "both", "but", "by", "could", "did",
             "do", "does", "doing", "during", "each", "for", "from", "further", "had", "has", "have",
             "having", "he", "he'd", "he'll", "he's", "her", "here", "here's", "hers", "herself", "him",
             "himself", "his", "how", "how's", "i", "i'd", "i'll", "i'm", "i've", "if", "in", "into", "is",
             "it", "it's", "its", "itself", "let's", "me", "my", "myself", "of", "on", "once", "only", "or",
             "other", "ought", "our", "ours", "ourselves", "out", "own", "same", "she", "she'd", "she'll",
             "she's", "should", "so", "some", "such", "than", "that", "that's", "the", "their", "theirs",
             "them", "themselves", "then", "there", "there's", "these", "they", "they'd", "they'll",
             "they're", "they've", "this", "those", "through", "to", "too", "until", "very", "was", "we",
             "we'd", "we'll", "we're", "we've", "were", "what", "what's", "when", "when's", "where",
             "where's", "which", "while", "who", "who's", "whom", "why", "why's", "with", "would", "you",
             "you'd", "you'll", "you're", "you've", "your", "yours", "yourself", "yourselves" ];

def pre_processing(sentence):

    sentence = sentence.lower()
    sentence = sentence.replace('amp','')
    sentence = re.sub('[@#]','',sentence)
    pattern = re.compile('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    sentence = pattern.sub('',sentence)
    sentence = re.split('[^a-zA-Z\']', sentence)
    stopwords = stopwords_list()
    res_words = [word for word in sentence if word not in stopwords]
    sentence = ' '.join(res_words)
    sentence = " ".join(filter(lambda x:x[0]!='\'', sentence.split()))

    return sentence
```

Fig. 3.3 Preprocessing of data

3.3.3 Feature Variables:

The variables used in the training are stated in **Fig. 3.4** with their descriptions.

In all following discussions, tweets *labeled as 0* are in favour of *Hillary Clinton*, tweets *labeled as 1* are *neutral*, while tweets *labeled as 2* are in favour of *Donald Trump.*

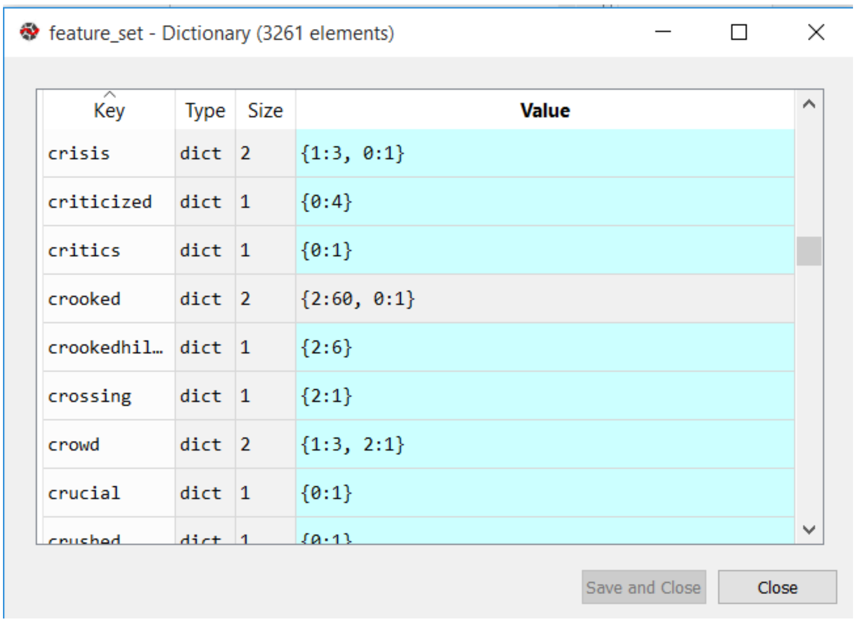


```python
#training and testing data
train_data, test_data = create_train_test(reader)

#label(positive/negative) : {word : count of number of occurences of the word}
dataset = {}

#label l : No. of records that are labeled l
no_of_items = {}

#word : {label l : count of the occurence of word with label l}
feature_set = {}

#hashtags : label
hash_tags = {}
```

Fig. 3.4 Variables of the model

feature_set-

In **Fig. 3.5** we can see that the word *crooked* is stored as a key. *{2:60, 0:1}* implies that the word (crooked) comes 60 times in tweets labeled as 2, while it comes 1 time in tweets labeled as 0.

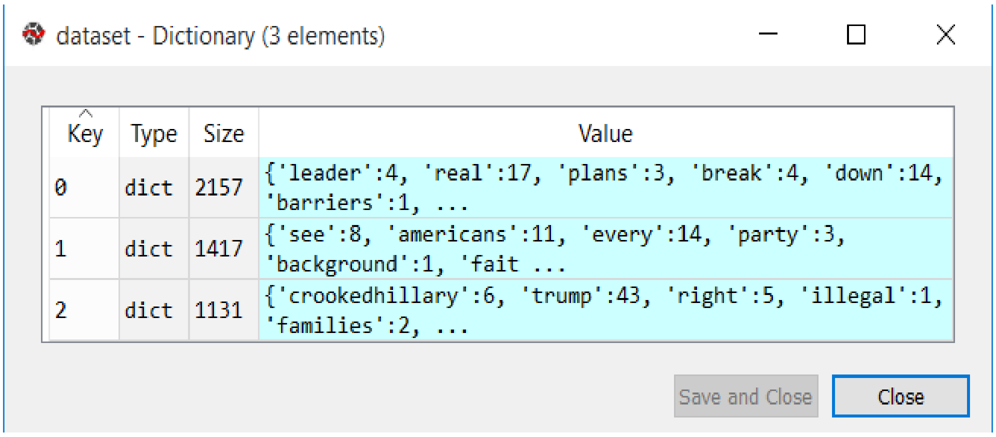


Key	Type	Size	Value
crisis	dict	2	{1:3, 0:1}
criticized	dict	1	{0:4}
critics	dict	1	{0:1}
crooked	dict	2	{2:60, 0:1}
crookedhil…	dict	1	{2:6}
crossing	dict	1	{2:1}
crowd	dict	2	{1:3, 2:1}
crucial	dict	1	{0:1}
crushed	dict	1	{0:1}

Fig. 3.5 feature_set

dataset-

In **Fig. 3.6** it is seen that the *polarity* i.e. *0,1 or 2* is stored as the key. Each word appearing in that particular polarity is stored with it's count. e.g.: *"trump"* appears *43 times* in *polarity 2*.

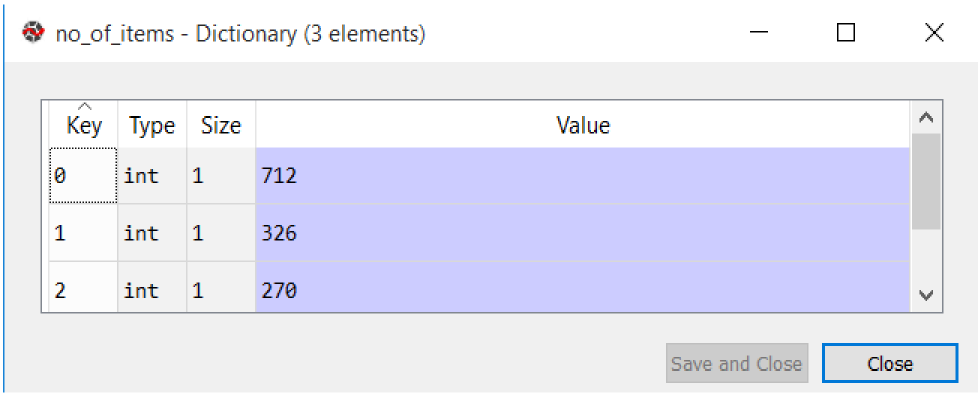


Key	Type	Size	Value
0	dict	2157	{'leader':4, 'real':17, 'plans':3, 'break':4, 'down':14, 'barriers':1, ...
1	dict	1417	{'see':8, 'americans':11, 'every':14, 'party':3, 'background':1, 'fait ...
2	dict	1131	{'crookedhillary':6, 'trump':43, 'right':5, 'illegal':1, 'families':2, ...

Fig. 3.6 dataset

no_of_items-

Fig. 3.7 shows that the *polarity* is stored as the key. The value corresponding the each key is the *number of words appearing in that polarity*.



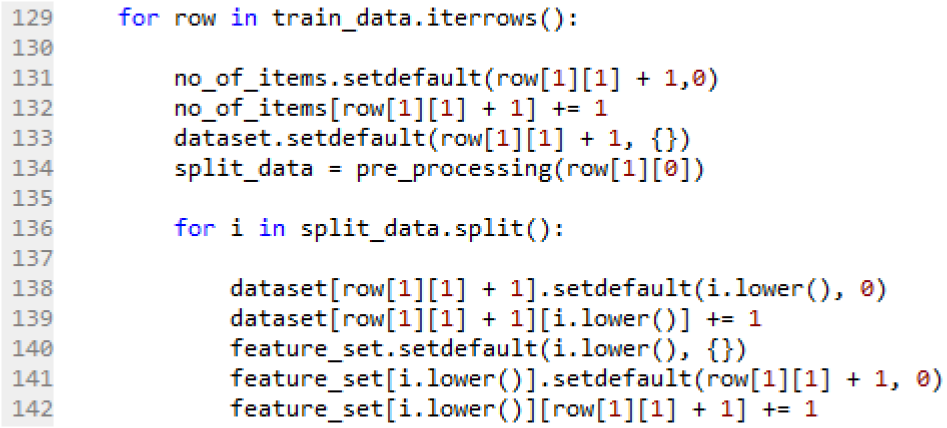
no_of_items - Dictionary (3 elements)

Key	Type	Size	Value
0	int	1	712
1	int	1	326
2	int	1	270

Fig. 3.7 no_of_items

3.3.4 Training:

The training of the model is shown in **Fig. 3.8**. While feeding each tweet in this section, the feature variables discussed above are updated.

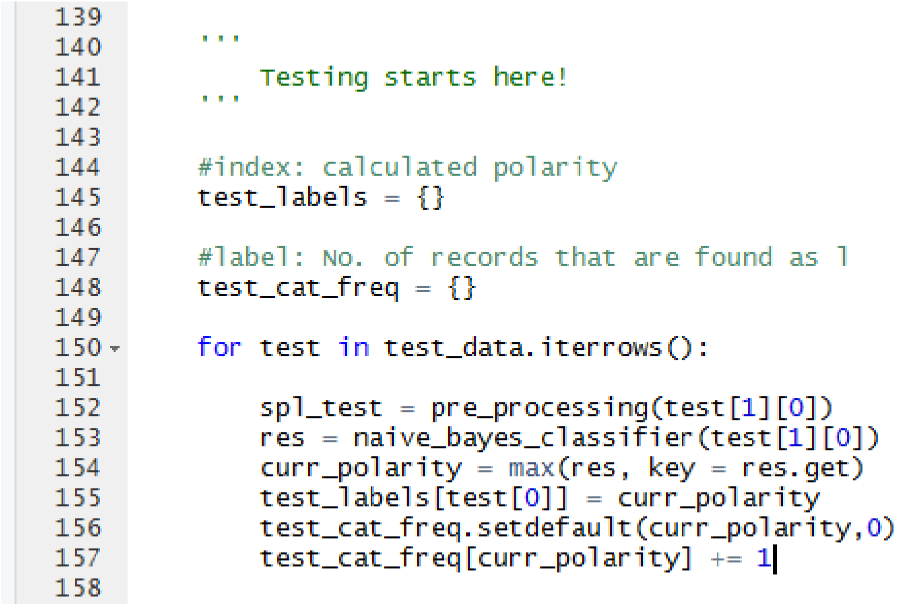


```
for row in train_data.iterrows():

    no_of_items.setdefault(row[1][1] + 1,0)
    no_of_items[row[1][1] + 1] += 1
    dataset.setdefault(row[1][1] + 1, {})
    split_data = pre_processing(row[1][0])

    for i in split_data.split():

        dataset[row[1][1] + 1].setdefault(i.lower(), 0)
        dataset[row[1][1] + 1][i.lower()] += 1
        feature_set.setdefault(i.lower(), {})
        feature_set[i.lower()].setdefault(row[1][1] + 1, 0)
        feature_set[i.lower()][row[1][1] + 1] += 1
```

Fig. 3.8 Training of the model

3.3.5 Testing :

The testing part of the code is shown in **Fig. 3.9**. After preprocessing the given tweet, it is passed to the classifier method to calculate the conditional probability of the tweet in each class.



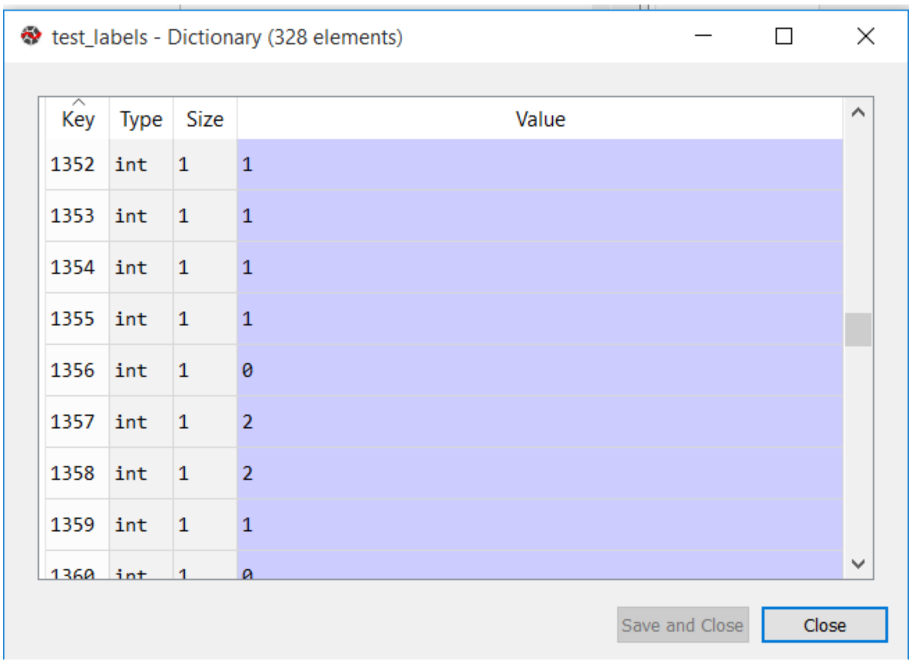
```
'''
    Testing starts here!
'''

#index: calculated polarity
test_labels = {}

#label: No. of records that are found as 1
test_cat_freq = {}

for test in test_data.iterrows():

    spl_test = pre_processing(test[1][0])
    res = naive_bayes_classifier(test[1][0])
    curr_polarity = max(res, key = res.get)
    test_labels[test[0]] = curr_polarity
    test_cat_freq.setdefault(curr_polarity,0)
    test_cat_freq[curr_polarity] += 1
```

Fig. 3.9 Testing the model

In the above figure, *res* stores *P(tweet|polarity)* for each polarity. The *max P* out of polarities 0.1.2 is stored in *curr_polarity* which implies the *overall polarity of the tweet*.

test_labels-

Fig. 3.10 shows that the *tweet number* in the dataset is stored as the key. The value corresponding to a key is the *overall polarity* of the tweet after calculating *res* and *curr_polarity*.

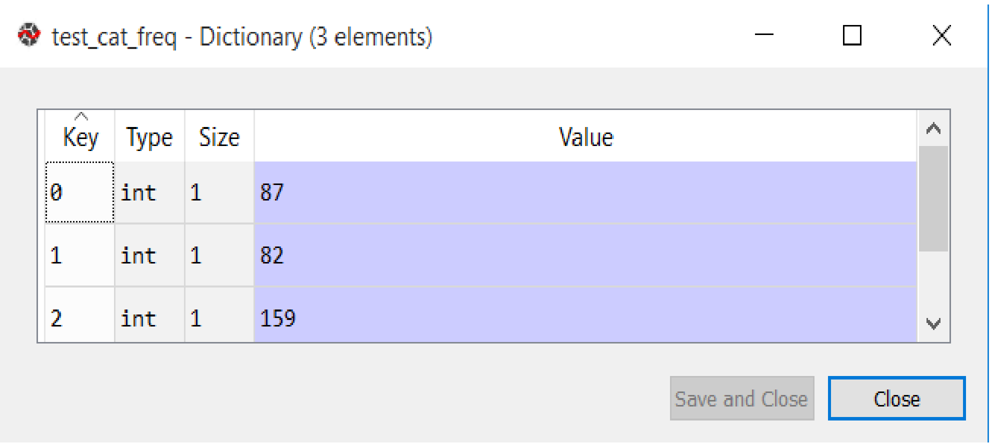


Key	Type	Size	Value
1352	int	1	1
1353	int	1	1
1354	int	1	1
1355	int	1	1
1356	int	1	0
1357	int	1	2
1358	int	1	2
1359	int	1	1
1360	int	1	0

Fig. 3.10 test_labels

test_cat_freq-

Fig, 3.11 shows that the polarity is stored as the key. The total number of tweets falling into each polarity is stored as the value. e.g.: there are 159 tweets with polarity 2.



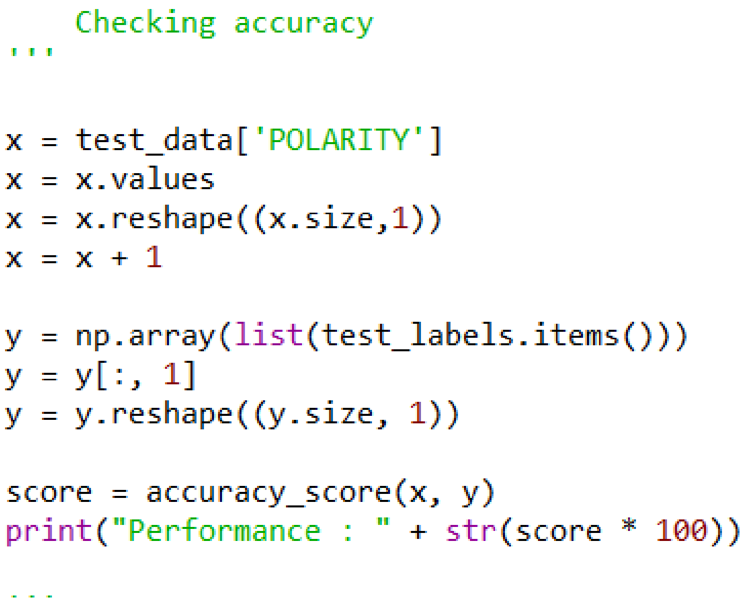
test_cat_freq - Dictionary (3 elements)

Key	Type	Size	Value
0	int	1	87
1	int	1	82
2	int	1	159

Fig. 3.11 test_cat_freq

3.3.6 Checking Accuracy:

The process of estimating the accuracy of the model is shown in the **Fig. 3.12**. We had taken 20% of our data for testing which was already labeled. We check the obtained polarity for every tweet in the testing set against the labels.



```
    Checking accuracy
'''

x = test_data['POLARITY']
x = x.values
x = x.reshape((x.size,1))
x = x + 1

y = np.array(list(test_labels.items()))
y = y[:, 1]
y = y.reshape((y.size, 1))

score = accuracy_score(x, y)
print("Performance : " + str(score * 100))

'''
```

Fig. 3.12 Checking accuracy of the model

3.3.7 Role of hashtags:

Hashtags “#” play a very important role in determining the sentiment of a tweet. Popular tweets which directly point to a particular party or are trending in support of a party can be given higher priority in calculating the conditional probability of a tweet.

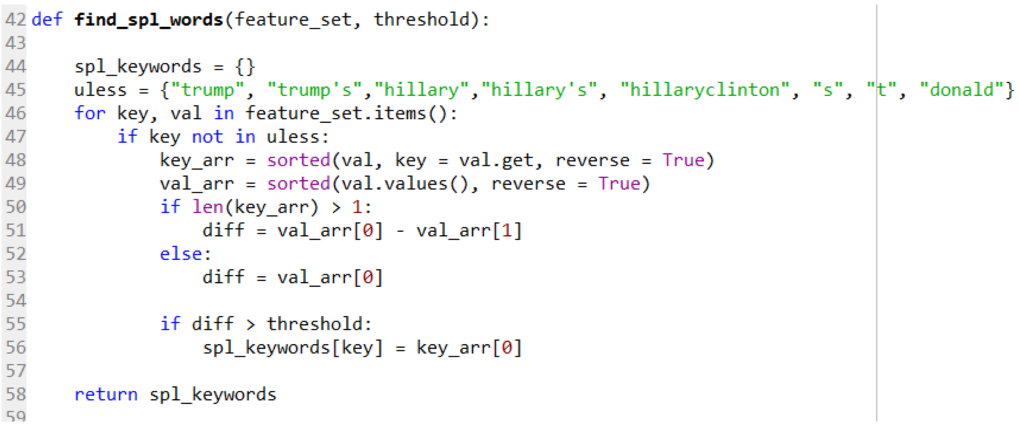
In our model, if a word is a part of list of hashtags for the given class, the posterior probability of that word becomes 1. This makes the model consider the important hashtags which are generally used to calculate polarity of the tweet efficiently.

3.3.8 Role of special words:

We categorize a word as a special word if it appears a number of times in one particular sense i.e. with a particular label. While calculating the polarity of a tweet in the test dataset, sentences which have any of these special words are directly assigned with the polarity of that special word. It is similar to assigning a weight to the a few words.

We can therefore update this list of special words according to the dataset and hence make our analysis better. Moreover, weights can be assigned to such words based on their importance which can also make the results better.

The following **Fig. 3.13** shows how these special words are obtained. We have maintained a variable called *threshold* which can be updated whenever required. As the name suggests, the words appearing in a particular polarity with a *frequency greater than threshold* will be categorized as special words.



```python
def find_spl_words(feature_set, threshold):

    spl_keywords = {}
    uless = {"trump", "trump's","hillary","hillary's", "hillaryclinton", "s", "t", "donald"}
    for key, val in feature_set.items():
        if key not in uless:
            key_arr = sorted(val, key = val.get, reverse = True)
            val_arr = sorted(val.values(), reverse = True)
            if len(key_arr) > 1:
                diff = val_arr[0] - val_arr[1]
            else:
                diff = val_arr[0]

            if diff > threshold:
                spl_keywords[key] = key_arr[0]

    return spl_keywords
```

Fig. 3.13 Special words

CHAPTER 4
RESULTS AND CONCLUSION

Open social networks are the most accurate instances of sociological trust. The trading of messages, followers and changing feelings of users give a rough stage to examine social trust in sentiment analysis domain. Especially the micro blogging platforms like Twitter which allows the user to express his/her opinion in 280 characters. Not only this boosts the idea of compactness but also opens the terrain for normalization of a broader sentiment.

Text classification is a standout amongst the most significant territories in sentiment mining, and sentiment analysis can be performed dependent on content characterization. Sentiment analysis comes under the umbrella of Natural Language Processing. Machine learning approaches have been so far great in conveying exact outcomes. Depending on the application, the accomplishment of any methodology will fluctuate. Lexical methodology is a prepared to-go and doesn't require any earlier data or preparing. While then again machine learning requires a well-structured classifier, tremendous measure of training data collections and execution tuning preceding arrangement.

In our project, we have used an algorithm to calculate the probability of a text being flagged as either in favour of any of the two parties or being neutral as compared to the default algorithms which generally provide only the general sentiment.Naive Bayes is one of the techniques to perform sentiment analysis. The general thought is to do some word tallying and figure a few measurements information to assemble the classifier. Here, we used the default method of Naïve-Bayes algorithm to predict the probability of a pre-processed tweet. The state of art algorithms which use Naïve-Bayes have an accuracy ~65-80% [15], which is also satisfied by our model. The data added to the hand flagged dataset contains the training data of the model as well as contains the data to be tested. The idea is to use 80% data to train the machine learning model and use the rest of the dataset to test the model. The results are not exactly linear but a general growth of the accuracy can be observed over the time as data increases but eventually, the growth can be seen to slow down.

The landmark for our project was the Trump vs. Hillary contest as it contained wide areas of interest and thereby providing a compatible resource for our dataset which were nothing but tweets from different users. Dataset is crucial for any machine learning platform and it wasn’t surprising that we had to hand flag our data in order to get the dataset in the required form. Moving ahead, the selection of Naïve-Bayes as the algorithm for our model was another milestone for us. The performance was merely a 15% in terms of accuracy, as we started with only 100 tweets and it went on to reach

65.7% as we touched 2500 tweets. Fig. 4.1 indicates the performance of the model over the size of dataset.

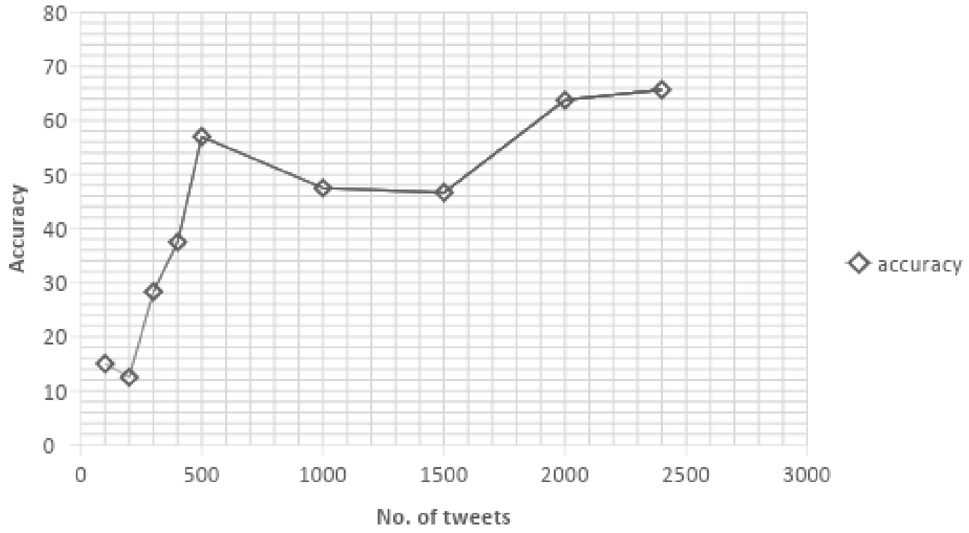


Fig. 4.1: Accuracy vs Dataset

The Naïve-Bayes algorithm focuses largely on probability of each word for a particular label thus, the strength of vocabulary should clearly affect the performance and it does. However, the decrease in the rate of growth indicates that after a certain point, the strength of the vocabulary, i.e., the number of words barely affect the performance but the size of the dataset is still a deal breaker.

After the examination all things considered, it is clear that best outcomes have been seen from machine learning methodologies, and least by lexical methodologies. Be that as it may, with no appropriate preparing of a classifier in machine learning approach results may fall apart radically. Hybrid approach has so far shown positive development to the extent execution is concerned. Work is being carried on cross breed approaches; henceforth so far just restricted data is accessible as far as anyone is concerned.

CHAPTER 5
FUTURE WORK

Although we have made a step towards the analysis and prediction of Naive Bayes implementation, there is still much more work that needs more and deeper research. Our model does well in terms of performance but it has a scope of improvement when compared to the state of art algorithms. Sentiment analysis can be very helpful for political parties to approach various problems in a modified way. Additionally, a very important and especially motivating factor is the potential business benefit that can come from fully functional sentiment analysis systems.

It can also be interesting to see the algorithm to run in a location-specific way which can help a political party to modify its agenda according to the data received from the geographical location. It is very practical and convenient to segregate the dataset based on the location and adding an additional flag which will allow the algorithm to find the general sentiment area-wise.

Some of the features/aspects that we wish to incorporate and/or improve are:

1. The collected sample data is limited. A technique to anticipate or deduct the location of a tweet dependent on the tweet's data and the client's information is ought to be found later on.

2. As of now, the algorithm works for any two competing parties which can be further extended to multiple parties and the general sentiment can be derived in terms of percentage, for or against a party.

3. As discussed earlier, hybrid approaches make the best algorithms for sentiment analysis. We, thereby, positively look forward to implement such approaches by extending the Naïve-Bayes algorithm. This not only improves our performance drastically but also gives us a deeper insight about the context of sentiment analysis.

4. Interpreting Sarcasm: The proposed methodology is as of now unequipped for deciphering sarcasm. When all is said in done sarcasm is the utilization of incongruity to taunt or pass on disdain, with regards to current work, sarcasm changes the extremity of an evidently positive or negative articulation into its inverse. This impediment can be overwhelmed by comprehensive investigation of essentials in "discourse-driven conclusion examination". The principle objective of this methodology is to observationally recognize lexical and down to earth factors that recognize sarcastic, positive and negative utilization of words.

5. Multi-lingual support: Because of the absence of multilingual lexical word reference, it is right now not achievable to build up a multilingual sentiment analyzer. Further research can be done in making the classifiers language autonomous, appeared by [16]. The creators have proposed a supposition examination framework with help vector machines, comparable methodology can be connected for our framework to make it language free.

REFERENCES

1. Sentiment Analysis: Types, Tools and Use Cases https://www.altexsoft.com/blog/business/sentiment-analysis-types-tools-and-use-cases/

2. Sentiment Analysis- everything you need to know https://monkeylearn.com/sentiment-analysis/

3. Hatzivassiloglou, V., Wiebe, J.: Effects of Adjective Orientation and Gradability on Sentence Subjectivity. Proceedings of the 18th International Conference on Computational Linguistics, New Brunswick, NJ (2000)

4. Kennedy, A., Inkpen, D.,: Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters, Computational Intelligence (2006) 110-125

5. Turney, P. D.: Thumbs up or thumbs down?. Semantic orientation applied to Unsupervised classification of reviews. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (2002) 417–424

6. Akshay, J.: A Framework for Modeling Influence, Opinions and Structure in Social Media. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, BC, July (2007) 1933–1934.

7. Durant, K., and Smith M.: Mining Sentiment Classification from Political Web Logs. Proceedings of Workshop on Web Mining and Web Usage Analysis of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Philadelphia (2006).

8. S. Prasad: Micro-blogging sentiment analysis using Bayesian classification methods (2010).

9. A. Pak and P. Paroubek: Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives. Proceeding SemEval '10 Proceedings of the 5th International Workshop on Semantic Evaluation (2010) 436-439.

10. B. Liu, X. Li, W.S. Lee, and P.S. Yu.: Text classification by labeling words. Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA; Cambridge, MA; London.. MIT Press (2004) 425-430.

11. P. Melville, W. Gryc, and R.D. Lawrence: Sentiment analysis of blogs by combining lexical knowledge with text classification. In Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining (2009) 1275-1284.

12. C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, J. Caro: Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning (2013).

13. M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf: Support Vector Machines (1998)

14. A Guide to Social Media Sentiment https://blog.hootsuite.com/social-media-sentiment-analysis-tools/

15. S. Prasad: Micro-blogging sentiment analysis using Bayesian classification methods (2010).

16. S. Narr, M. Hifulfenhaus, and S. Albayrak, Language-independent twitter sentiment analysis.", The 5th SNA-KDD Workshop (2011).

BIBLIOGRAPHY

1. H. Zhang, Di Li: Naive Bayes Text Classifier (2007).

2. L. Povoda, R. Burget, M. K. Dutta: Sentiment analysis based on Support Vector Machine and Big Data (2016).

3. N. Zainuddin, A. Selamat: Sentiment analysis using Support Vector Machine (2014).

4. https://brand24.com/blog/sentiment-analysis/

5. https://www.researchgate.net/post/What_are_the_hardware_software_platforms_used_for_Sentiment_Analysis

6. https://mention.com/blog/social-media-sentiment-analysis/

7. https://www.brandwatch.com/blog/understanding-sentiment-analysis/

8. https://www.researchgate.net/publication/312068124_Sentiment_Classifi_cation_using_Decision_Tree_Based_Feature_Selection

9. https://www.knime.com/knime-applications/social-media-sentiment-analysis