

**SCHOOL OF COMPUTER SCIENCE ANDAPPLICATIONS**

A Project Report

on

TWITTER Sentiment Analysis using NLP

Submitted in Partial Fulfillment of the Requirements for Award of the Degree of

Bachelor of Science (Research) in Computer Science

Cloud Computing and Big data

Submitted by

Mr. JAYANT BISHNOI

(R16BS034)

Under the Guidance of

Mrs. Shreetha Bhat

InternalGuide External Guide

May 2020

Rukmini Knowledge Park,Kattigenahalli*,* Yelahanka, Bengaluru – 560064

www.reva.edu.in

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**SCHOOL OF COMPUTER SCIENCE AND APPLICATIONS**

**CERTIFICATE**

Certified that the project work entitled **TWITTER Sentiment Analysis using NLP** carried out under our guidance by **Jayant Bishnoi** , **R16BS034**, a bonafide student of REVA University during the academic year 2019-20, is submitting the project report in partial fulfillment for the award of **Bachelor of Science (Research) in Computer Science –Cloud Computing and Big Data** during the academic year **2019–20**. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

**Signature with date Signature with date Signature with date**

**Dr.S .Senthil (Director)**

**Mrs. Shrettha Bhat**

**Internal Guide**

**<External Guide name>**

**External Guide**

**Signature with date**

**­­­­­­­­­­­­­­­­­­­**

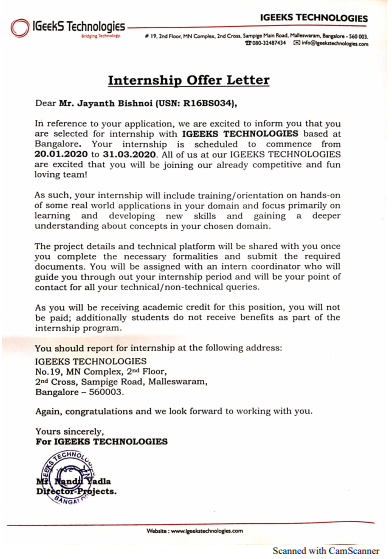
**<Vice Chancellor>**

**Name of the Examiner with affiliation Signature with Date**

1.

2.

COMPANY CERTIFICATE



**DECLARATION**

I, Mr. Jayant Bishnoi student of Bachelor of Science (Research) in Computer Science –Cloud Computing and Big Data belong in to School of Computer Science and Applications, REVA University, declare that this Project work entitled “TWITTER Sentiment Analysis using NLP ” is the result of the Project workdone by me under the supervision of Prof. Shreetha Bhat ,School of CSA

This Project work is submitted in partial fulfillment of the requirements forthe award of the degree Bachelor of Science (Research) in Computer Science –Cloud Computing and Big Data by the REVA University, Bangalore during the academic year 2019-20.

I further declare that this Project report or any part of it has not been submitted for award of any other Degree/ Diploma of this University or any other University/ Institution.

*(Signature of the candidate)*

*Signed by me on:* < date, month and year >

*Certified that this project work submitted by Jayant Bishnoi has been carried out under our guidance and the declaration made by the candidate is true to the best of my knowledge.*

*Signature of Internal Guide Signature of External Guide,*

*Date :………..Date :………..*

*Signature of Director of School*

*Date :………..*

*Official Seal of the School*

**ACKNOWLEDGEMENT**

# 

# I would like to acknowledge all whose guidance and encouragement that has made us to do what is done so far, I avail the opportunity to express our deep sense of gratitude and sincere thanks to our School of Computer Science and Applications which is always been a tremendous source for inspiration.

# I express my sincere gratitude to Dr. P. Shyama Raju, Honorable Chancellor, REVA University, Bengaluru for providing us the facilities.

I am thankful to Dr. Surendra Rao Shankapal, Vice Chancellor (In-charge), Dr. M. Dhanamjaya, Registrar, Dr. Yadavalli Lakshman Kumar, Pro Vice-Chancellor, Dr. Sunil Kumar S. Manvi, Dean, Faculty of Engineering and Technology, Dr. N. Ramesh, Dean, Faculty of Science and Technology, REVA University, Bengaluru for their support and encouragement.

I take this opportunity to express my heartfelt sincere thanks to Dr. S. Senthil, Professor & Director School of CSA, REVA University and Dr. Revina Rebecca , Associate Professor, BSc(Research) Programme Coordinator, School of CSA, REVA University whose encouragement and best wishes provided impetus for this project.

# My sincere gratitude to my internal guide,(Shreetha bhatt,) for this valuable suggestion and her/his ever-following encouragement towards the completion of this project. I am deeply indebted to my external guide, ( , ) for giving me permission to commence this project in the first instance, to do the necessary research work and to use the organizational data.

# 

# Last but not the least I also thank our parents for being supportive in all our activities and carrier without whom it wouldn’t be possible for us to reach successful completion of this project.

**ABSTRACT**

Our day to day life has always been influenced by what people think. Ideas and opinions of others have always affected our own opinions .The explosion of Web 2.0 has led to increased activity in podcasting , Blogging and Tagging ,Contributing to RSS , Social Bookmarking and Social Networking.

As a result there has been an eruption of interest in people to mine these vast resources of data for opinions.Sentiment Analysis or Opinion- Mining is the Computational Treatment of Opinions , sentiments and subjectivity of text.

In this report, we discuss various approaches to perform computational treatment of sentiments or opinions.There are various Supervised or Data-Driven Techniques to Sentiment Analysis , like Naive Bayes ,SVM(Support Vector machines) and Senti WordNet approach to Sentiment Analysis Natural language processing(NLP).

Twitter, one of the largest social media site receives tweets in millions every day.

This huge amount of raw data can be used for industrial or business purpose by organizing according to our requirement and processing. This paper provides a way of sentiment analysis using Natural language processing which will process the huge amount of data on a machine learning model faster in real time

IN our project we have attempted to study the classification of data based on the acerbity of the text that is being thrown in the form of tweets by various users.

and made an attempt to understand the implication of hastags and their associations with the general sentiment of the masses.

Furthermore, an attempt has been made at the classification of the of data that is present in the dataset , which has been segregated on the basis of the basic human emotional palette.

Keywords --

.Machine Learning

.Sentiment analysis,

.NLP(natural language processing),

.Naive Bayes

.TF-IDF,

.Wordcloud.

**TABLE OF CONTENTS**

**CHAPTERS** **PAGE#**

1. **INTRODUCTION**

1.1 INTRODUCTION TO PROJECT ...........................................................

1.2 STATEMENT OF THE PROBLEM ........................................................

- BRIEF DESCRIPTION OF THE PROJECT

1. **LITERATURE SURVEY**
2. **SYSTEM ANALYSIS**

3.1 EXISTING SYSTEM .............................................................................

3.1.1 LEXICON BASED APPROACH

3.1.2 MACHINE LEARNING BASED

3.2 PROPOSED SYSTEM ...........................................................................

* + 1. NAIVE BAYES
    2. NATURAL LANGUAGE PROCESSING
    3. SUPPORT VECTOR MACHINE
    4. ADVANTAGES OF PROPOSED SYSTEM

3.3 FEASIBILITY ANALYSIS

3.3.1 Technical Feasibility.....................................................................................

3.3.2 Operational Feasibility...............................................................................…

3.3.3 Economic Feasibility ...................................................................................

3.4 REQUIREMENT DEFINATION .................................................................

3.4.1 Functional Requirements.........................................................................………..

3.4.2 Non-Functional Requirements.................................................................………..

**4.SYSTEM DESIGN AND DEVELOPEMENT**

4.1System Flow Diagram …………………………………………………………

4.2Activity Diagram ………………………………………………..

4.3 Data Flow Diagram …………………………………………………….

4.4Use Case Diagram………………………………………………….

**5.TOOLS AND TECHNOLOGIES**

5.1 HARDWARE REQUIREMENT ............................................................

5.2 SOFTWARE REQUIREMENT ..........................................................…

**6. IMPLEMENTATION**

6.1 COLLECTING THE DATA ..................................…………………….

6.2 PREPROCESSING AND CLEANING OF DATA.................................

6.3 STORY GENERATION AND VISUALIZATION ............................…

6.4 EXTRACTING FEATURES FROM CLEAND TWEETS .................…

6.5 BUILDING THE MODEL .......................………………………………

**7.CODING**

7.1 PSEUDO CODE…………………………………………………

**8.SOFTWARE TESTING(Test cases)**

8.1.Unit Testing…………………………………………………………………

8.2. System Testing...............................................................................................

8.3. Performance Testing.......................................................................................

8.4. Verification and Validation…………………………………………………

**9.CONCLUSION**

--FUTURE ENHANCEMENT …………………………………………….

**10.BIBILOGRAPHY** ..........................................................................………………….

**CHAPTER 1 : INTRODUCTION**

**CHAPTER: 1**

**INTRODUCTION**

* 1. **INTRODUCTION TOTHE PROJECT:**

Sentiment analysis is a text analysis method that detects polarity (e.g. a positive or negative opinion) within text, whether a whole document, paragraph, sentence, or clause.

Understanding people’s emotions is essential for businesses since customers are able to express their thoughts and feelings more openly than ever before. [By automatically analyzing customer feedback](https://monkeylearn.com/blog/customer-feedback-analysis/" \t "https://monkeylearn.com/sentiment-analysis/_blank), from survey responses to social media conversations, brands are able to listen attentively to their customers, and tailor products and services to meet their needs.

For example, using sentiment analysis to automatically analyze 4,000+ reviews about your product could help you discover if customers are happy about your pricing plans and customer service.

### Types of Sentiment Analysis

Sentiment analysis models focus on polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc), and even on intentions (e.g. interested v. not interested).

Here are some of the most popular types of sentiment analysis:

#### **Fine-grained Sentiment Analysis**

If polarity precision is important to your business, you might consider expanding your polarity categories to include:

* Very positive
* Positive
* Neutral
* Negative
* Very negative

This is usually referred to as fine-grained sentiment analysis, and could be used to interpret 5-star ratings in a review, for example:

* Very Positive = 5 stars
* Very Negative = 1 star

#### **Emotion detection**

This type of sentiment analysis aims at detecting emotions, like happiness, frustration, anger, sadness, and so on. Many emotion detection systems use lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms.

One of the downsides of using lexicons is that people express emotions in different ways. Some words that typically express anger, like bad or kill (e.g. your product is so bad or your customer support is killing me) might also express happiness (e.g. this is bad ass or you are killing it).

#### **Aspect-based Sentiment Analysis**

Usually, when analyzing sentiments of texts, let’s say product reviews, you’ll want to know which particular aspects or features people are mentioning in a positive, neutral, or negative way. That's where [aspect-based sentiment analysis](https://monkeylearn.com/blog/aspect-based-sentiment-analysis/" \t "https://monkeylearn.com/sentiment-analysis/_blank) can help, for example in this text: "The battery life of this camera is too short", an aspect-based classifier would be able to determine that the sentence expresses a negative opinion about the feature battery life.

#### **Multilingual sentiment analysis**

Multilingual sentiment analysis can be difficult. It involves a lot of preprocessing and resources. Most of these resources are available online (e.g. sentiment lexicons), while others need to be created (e.g. translated corpora or noise detection algorithms), but you’ll need to know how to code to use them.

**1.2 STATEMENT OF THE PROBLEM:**

**The** textual data on the internet is growing at a rapid pace. Different industries are trying to use this huge textual data for extracting the people’s views towards their products.

Social media is a vital source of information in this case. It is impossible to manually analyze the large amount of data.

This is where the need of automatic categorization becomes apparent. Subjective data is analyzed generally in this case.

There are a large number of social media websites that enable users to contribute, modify and grade the content. Users have an opportunity to express their personal opinions about specific topics.

The example of such websites include blogs, forums, product reviews sites, and social networks. In this case, twitter data is used.

Sites like twitter contain prevalently short comments, like status messages on social networks like twitter or article reviews on Digg.

Additionally many web sites allow rating the popularity of the messages which can be related to the opinion expressed by the author. The focus of our project is to assign the polarity to each tweet I.e. whether the author express positive or negative opinion.

It’s estimated that [80% of the world’s data is unstructured](https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/" \t "https://monkeylearn.com/sentiment-analysis/_blank), in other words it’s unorganized. Huge volumes of text data (emails, support tickets, chats, social media conversations, surveys, articles, documents, etc), is created every day but it’s hard to analyze, understand, and sort through, not to mention time-consuming and expensive.

Sentiment analysis, however, helps businesses make sense of all this unstructured text by automatically tagging it.

Despite the availability of software to extract data regarding a person’s sentiment on a specific product or service,organizations and other data workers still face issues regarding the data extraction.

Sentiment Analysis of Web Based Applications Focus on

Single Tweet Only

.

With the rapid growth of the World Wide Web, people are using social media such as Twitter which generates big volumes of opinion texts in the form of tweets which is available for the sentiment analysis . This translates to a huge volume of information from a human viewpoint which

make it difficult to extract a sentences, read them, analyze tweet by tweet, summarize them and organize them into an understandable format in a timely manner .

**•Difficulty of Sentiment Analysis with inappropriate English**

Informal language refers to the use of colloquialisms and slang in communication, employing the conventions of spoken language such as ‘would not’ and ‘wouldn’t’. Not all

systems are able to detect sentiment from use of informal language and this could hanker the analysis and decision-making process. Emoticons, are a pictorial representation of human facial expressions , which in the absence of body language and prosody serve to draw a receiver's attention to the tenor or temper of a sender's nominal verbal communication, improving and changing its interpretation . For example,

☺ indicates a happy state of mind. Systems currently in place do not have sufficient data to allow them to draw feelings out of the emoticons.

As humans often turn to emoticons to properly express what they cannot put into words . Not

being able to analyze this puts the organization at a loss. Short-form is widely used even with short message service (SMS).

The usage of short-form will be used more frequently on Twitter so as to help to minimize the characters used. This is because Twitter has pu t a limit on its characters t o 1 4 0

. F o r e x a m p l e , ‘Tba’ refers to be announced.

**Benefits of sentiment analysis include:**

1 .Sorting Data at Scale Can you imagine manually sorting through thousands of tweets, customer support conversations, or surveys? There’s just too much data to process manually. Sentiment analysis helps businesses process huge amounts of data in an efficient and cost-effective way.

2.Real-Time Analysis Sentiment analysis can identify critical issues in real-time, for example is a PR crisis on social media escalating?

Is an angry customer about to churn?

Sentiment analysis models can help you immediately identify these kinds of situations and [gauge brand sentiment](https://monkeylearn.com/blog/brand-sentiment/" \t "https://monkeylearn.com/sentiment-analysis/_blank), so you can take action right away.

3.Consistent criteria It’s estimated that people only agree around 60-65% of the time when determining the sentiment of a particular text. Tagging text by sentiment is highly subjective, influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all of their data, helping them improve accuracy and gain better insights.

Sentiment analysis uses various Natural Language Processing (NLP) methods and algorithms,

**The main types of algorithms used include:**

****>Rule-based**** systems that perform sentiment analysis based on a set of manually crafted rules.

****>Automatic**** systems that rely on machine learning techniques to learn from data.

****> Hybrid**** systems that combine both rule-based and automatic approaches.

#### **Rule-based Approach**

Usually, a rule-based system uses a set of human-crafted rules to help identify subjectivity, polarity, or the subject of an opinion.

These rules may include various techniques developed in computational linguistics, such as:

* Stemming, tokenization, part-of-speech tagging and parsing.
* Lexicons (i.e. lists of words and expressions).

HERE IS A BASIC EXAMPLE ,

1.Define two lists of polarized words (e.g. negative words such as bad, worst, ugly, etc and positive words such as good, best, beautiful, etc).

1. Counts the number of positive and negative words that appear in a given text.
2. If the number of positive word appearances is greater than the number of negative word appearances, the system returns a positive sentiment, and vice versa. If the numbers are even, the system will return a neutral sentiment.

Rule-based systems are very naive since they don't take into account how words are combined in a sequence. Of course, more advanced processing techniques can be used, and new rules added to support new expressions and vocabulary. However, adding new rules may affect previous results, and the whole system can get very complex. Since rule-based systems often require fine-tuning and maintenance, they’ll also need regular investments.

#### **Automatic Approache**

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on [machine learning](https://monkeylearn.com/blog/gentle-guide-to-machine-learning/" \t "https://monkeylearn.com/sentiment-analysis/_blank) techniques. A sentiment analysis task is usually modeled as a classification problem, whereby a classifier is fed a text and returns a category, e.g. positive, negative, or neutral.

IN this project we will try to create an analysis model based on a rule based approach

To keep things less complicated, so this model can be optimized to function with lesser resources and the purpose is to achieve maximum accuracy we will use NLP and basic ML algorithms and libraries to do this task.

.

**CHAPTER 2 : LITERATURE SURVEY**

**CHAPTER 2**

**LITERATURE SURVEY**

* 1. **: LITERATURE SURVEY**

A study related to automatic text summarization which is gaining popularity lately. For covering all the important contents and general information, a compressed version of documents is created. Few features are used to score sentences within extractive text summarization.

In past few studies, large numbers of features network based techniques are proposed. In order to score the sentences, each of the features which use metrics and idea of complex network have been reviewed. Discussion of experimental results on single component and combinations of various features are made. The assessments being performed on DUe 2002 data sets include the quantitative and qualitative aspects.

For summarization shortest ways were provided using which the highest scores for increasing the quality were achieved. The results that were achieved by integrating similar kinds of network properties were another contribution of this approach. The sentences were chosen on the basis of this incredible influence. The two categorizations of text summarization techniques are extractive and abstractive methods.

This paper presented a comprehensive survey of both of these techniques used for text summarization. This paper studied the different summarization techniques. An effective summary that has less redundancy and includes grammatically correct sentences is to be generated through the summarization approach. The users can use extractive and abstractive methods from which efficient results are achieved. For generating compressed and readable information for users, the hybridization technique proposed here proves to be highly efficient as per the test results.

Mihai Dascălu, et.al, (2011)proposed an automatic approach with respect to NLP and used distributing figuring to improve the runtime performance .

Further, with respect to the multilayered engineering an exceptional grading component is provided. A replicated worker design is sent to improve the speed of this process. In this approach, along with increment in performance level, there are two important aspects to be considered which are, load balancing and fault tolerance.

The corpus of chats can be assessed in a timely manner resulting is providing a quick access to the participants’ feedbacks as per this demonstration of framework. By investigating huge corpuses in very less time, a solution is provided through which the general performance is improved. This is done by utilizing the subtle elements on a distributed form of instrument. Under different conditions and loads, the performance of proposed work is better as compared to existing approaches.WirajUdaraWickramaarachchi et.al, (2017) studied that the chats, comments on images, status and other data uploaded by people on social networks is one important method being used by them to express their opinions and emotions

. Due to the lack of any face to face interaction amongst the users, the effectiveness of communication is increased . Any kinds of emotions that are expressed through content are distinguished using content processing strategies.

From handling the content uploaded in the form of images to the general emotions mentioned in content are handled in this paper by proposing a new approach.

The previous works were extended to develop a new methodology.

LSA, a Startup Collected data from the twitter using the twippy API

1. Input data and tokanize the data

2. Remove Links and smilies

3. remove stop workds Apply N-gram algorithm form the feature extraction Apply SVM algorithm for the classification

Information Systems & Business Network (ISN) light weight approach, was utilized along with the change and extensions in this proposed work.

The online emotion testing approaches also need to be light weight as per the applications. It is seen that in comparison to existing approaches, the proposed provided a prototype of GUI which provided better performance results.

YeongWai Chung (2018) Did the study on big data and open challenges of sentiment analysis. Twitter generates 175 millions tweets daily basis.world already generates 1 zettabyte of data. The effectiveness and applicability of the machine learning algorithm is shown by the extensive literature survey.

This potential effectiveness also invites proportional increase in complexity. It is shown more than 80% of social media data can be used for analysis.

Ankita Gupta, JyotikaPruthi, Neha Sahu . The proposed hybrid model is curated in three steps. The first step is the preprocessing stage where impurities correction and multi aspect based filtration is applied. The spell correction, stemming, abbreviation expansion, stop words removal is defined in this stage to normalize the input tweets. The second stage takes the filtered text for processing in order to define statistical features. The initial training and testing sets are converted into feature sets.

The features obtained in the second step are evaluated via the use of the hybrid classifier. Sentiment prediction is the final stage of this model. In classification stage, the probabilistic predictive decision is applied for selection of KNN or SVM classifier for individual instance.

**A. Opining Mining**

Opinion mining refers to the broad area of natural language processing, text mining, computational linguistics, which involves the computational study of sentiments, opinions and emotions expressed in text .

Although, view or attitude based on emotion instead of reason is often colloquially referred to as a

sentiment . Hence, lending to an equivalent for opinion mining or sentiment analysis. stated that opinion mining has many application domains including accounting, law, research, entertainment,

education, technology, politics, and marketing.

In earlier days many social media have given web users avenue for opening up to express and share their thoughts and opinions .

**B. Twitter**

Twitter is a popular real time microblogging service that allows users to share short information known as tweets which are limited to 140 characters . Users write tweets to express their opinion about various topics relating to their daily lives. Twitter is an ideal platform for the extraction of

general public opinion on specific issues.

A collection of tweets is used as the primary corpus for sentiment analysis, which refers to the use of opinion mining or natural language processing .

Twitter, with 500 million users and million messages per day, has quickly became a valuable asset for organizations to invigilate their reputation and brands by extracting and analyzing the sentiment of the tweets by the public about their products, services market and even about competitors .

highlighted that, from the social media generated opinions with the mammoth growth of the world wide web, super volumes of opinion texts in the form of tweets, reviews, blogs or any discussion groups and forums are available for analysis,

thus making the world wide web the fastest, most comprising and easily accessible medium for sentiment analysis.

1. **Microblogging with E-commerce**

A microblogging platform such as Twitter is alike to a conventional blogging platform just single posts are shorter

. Twitter has limited for a small number of words which are designed for the quick transmission of information or exchange of opinion . However, small business or large organizations are initiation to the potential of microblogging platform has been developed a few years’ time for promoting

foreign trade website by using a foreign microblogging platform as Twitter marketing.

The instant of sharing, interactive, community-oriented features are opening an e-commerce, launched a new bright spot which it can be shown that microblogging platform has enabled companies do brand image, product important sales channel, improve product sales, talk to consumer for a good interaction and other business activities involved .

in fact, the companies manufacturing such products have started to poll theses microblogs to get a sense of general sentiment for a product. Many times these companies study user reactions and reply to users on microblogs .

**D. Social Media**

Defined a social media as a group of Internet-based applications that create on the ideological and technological foundations of Web2.0 which is allowed to build and exchange of user generated contents. In a discussion of Internet World Start, identified that a trend of internet users is increasing and continuing to spend more time with social media by the total time spent on mobile devices and

social media in the U.S.across PC increased by 37percent to121billion minutes in 2012, compared to 88 billion minutes in 2011. On the other hand, businesses use social networking sites to find and communicate with clients, business can be demonstrated damage to productivity caused by social

networking . As social media can be posted so easily to the public, it can harm private information to spread out in the social world .

On the contrary, discussed that the benefits of participating in social media have gone beyond simply social sharing to build organization’s reputation and bring in career

opportunities and monetary income.

In addition,

mentioned that the social media is also being used for advertisement by companies for promotions, professionals for searching, recruiting, social learning online and electronic commerce.

Electronic commerce or E-commerce refers to the purchase and sale of goods or services online which can via social media, such has Twitter which is convenient due to its 24-hours availability, ease of customer service and global reach .

Among the reasons of why business tends to use more social media is for getting insight into consumer behavioral tendencies, market intelligence and present an opportunity to

learn about customer review and perceptions.

**E.Twitter Sentiment Analysis**

The sentiment can be found in the comments or tweet to provide useful indicators for many different purposes .

A sentiment can be categorized into two groups, which is negative and positive words. Sentiment analysis is a natural language processing techniques to quantify an expressed opinion or sentiment

within a selection of tweets .

**CHAPTER 3: SYSTEM ANALYSIS**

**CHAPTER3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM:**

**Techniques of Sentiment Analysis**

The semantic concepts of entities extracted from tweets can be used to measure the overall correlation of a group of entities with a given sentiment polarity. Polarity refers to the most basic form, which is if a text or sentence is positive or negative

However, sentiment analysis has techniques

in assigning polarity such as:

**1.Natural Language Processing (NLP)**

NLP techniques are based on machine learning and especially statistical learning which uses a general learning algorithm combined with a large sample, a corpus, of data to learn the rules . Sentiment analysis has been handled as a Natural Language Processing denoted NLP, at many levels of granularity. Starting from being a document level classification task , it has been handled at the sentence level and more recently at the phrase level . NLP is a field in computer science which involves making computers derive meaning from human language and input as a way of interacting with the real world.

**2. Case-Based Reasoning (CBR)**

Case-Based Reasoning (CBR) is one of the techniques available to implement sentiment analysis. CBR is known by recalling the past successfully solved problems and use the same solutions to solve the current closely related problems .identified some of the advantages of using CBR that CBR does not require an explicit domain model and so elicitation becomes a task of gathering care histories and CBR system can learn by acquiring new knowledge as cases.

This and the application of database techniques make the maintenance of large columns of information easier .

**3. Artificial Neural Network (ANN)**

Artificial Neural Network (ANN) or known as neural network is a mathematical technique that interconnects group of artificial neurons. It will process information using the connections approach to computation. ANN is used in finding the relationship between input and output or to find patterns in data.

**4.Support Vector Machine(SVM)**

Support Vector Machine is to detect the sentiments of tweets . SVM is able to extract and analyze to obtain upto70%-81.3% of accuracy on the test set. collected training data from three different

Twitter sentiment detection websites which mainly use some pre-built sentiment lexicons to label each tweet as positive or negative. Using SVM trained from these noisy labeled data, they obtained 81.3% in sentiment classification accuracy.

**5.Lexicon-based Approach**

Lexicon-based methods make use of predefined list of words where each word is associated with a specific sentiment . The lexicon methods vary according to the context in which they were created and involve calculating orientation for a document from the semantic orientation of texts or

phrases in the documents .

Besides, also states that a lexicon sentiment is to detect word-carrying opinion in the

corpus and then to predict opinion expressed in the text.has shown the lexicon methods which have a basic paradigm

which are:

i .Preprocess each tweet, post by remove punctuation

ii.Initialize a total polarity score (s) equal 0 -> s=0

iii.Check if token is present in a dictionary, then

If token is positive, s will be positive (+)

If token is negative, s will be negative (-)

iv. Look at the total polarity score of tweet post

If s > threshold, tweet post as positive

If s < threshold, tweet post as negative

However, one **advantage** of leaning-based method, is that it has the ability to adapt and create trained

models for specific purposes and contexts.

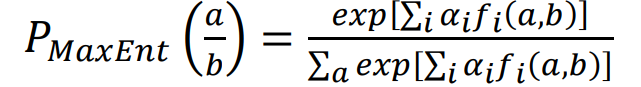
In contrast, an availability of labeled data and hence the low applicability of

the method of new data which is cause labeling data might be

costly or even prohibitive for some tasks.

**6.Maximum Entropy**

The Maximum Entropy (MaxEnt) classifier estimates the conditional distribution of a class marked a given a record b utilizing a type of exponential family with one weight for every constraint. The model with maximum entropy is the one in the parametric family  **pmax ent. (a/b)** that maximizes the likelihood. Numerical methods such as iterative scaling and quasi-Newton optimization are usually employed to solve the optimization problem. The model is represented by the followig



Where a is the class, b is the predictor. The weight of vector is denoted as (alpha i).

**3.2 PROPOSED SYSTEM**

**There are primarily two approaches for the sentiment classification of opinionated texts**

1. Using a machine learning based text classifier such as Naive Bayes, Support Vector Machine(SVM)
2. Using Natural Language Processing (NLP)

**3.2.1**

**MACHINE LEARNING**

The text is classified into classes by the machine learning based approach using classification techniques. The two broader categorizations of these machine learning techniques are:

1. Unsupervisedlearning : There is no category involved and the targets are not provided by them at all. Thus, clustering is considered to be an important factor here.
2. Supervised learning : The labeled dataset is used to develop this method. When the classification approach is to be designed, the labels are provided to the model. For getting significant outputs when going through decision making these labeled datasets are trained .

The determination and extraction of particular sets of features such that the sentiments can be detected is the success of both of these learning techniques. Naive Bayes (NB), Maximum Entropy (ME), and Support Vector machines (SVM) are few amongst the widely used machine learning techniques for sentiment classification. When having an initial set of labeled opinions is unrealistic for training the classifier, the semi-supervised and unsupervised techniques are designed.

1. **Naive Bayes Classifier**

This is a classification method that relies on Bayes' Theorem with strong (naive) independence assumptions between the features. A Naive Bayes classifier expects that the closeness of a specific feature (element) in a class is disconnected to the closeness of some other elements. For instance, an organic fruit might be considered to be an apple if its color is red, its shape is round and it measures approximately three inches in breadth. Regardless of whether these features are dependent upon one another or upon the presence of other features, a Naïve Bayes classifier would consider these properties independent due to the likelihood that this natural fruit is an apple. Alongside effortlessness, the Naive Bayes is known to out-perform even exceedingly.

The Naive Bayes is widely used in the task of classifying texts into multiple classes and was recently utilized for sentiment analysis classification.

A considerable numbers of features are utilized in feature vector through Naive Bayes classifier .

Since these features are independent equally, analyzing them exclusively is important. The mathematical representation of conditional probability for Naive Bayes is given as:



A feature vector denoted by "x" is included here which is defined by X={x1,x2,....xm}. The class label is represented by‘c’. The classification of different types of independent features such as positive and negative keywords, emoticons and emotional keywords is done efficiently using Naive Bayes. The relationships amongst features are not considered in Naive Bayes classifier. Thus, the relationships which exist among emotional keyword, negation words and speech tag are not utilized in it.

1. **SUPPORT VECTOR MACHINES (SVM):**

The support vector machine (SVM) is known to perform well in sentiment analysis. SVM investigates information, characterizes choice limits and uses the components for the calculation, which are performed in the input space. The vital information is presented in two arrangements of vectors, each of size m. At this point, each datum (expressed as a vector) is ordered into a class.

Next, the machine identifies the boundary between the two classes that is far from any place in the training samples . The separate characterizes the classification edge, expanding the edge lessens ambivalent choices. SVM has been proven to perform more effectively than the Naïve Bayes classifier in various text classification problems.

Many researchers of sentiment analysis found Support Vector Machines to be

very effective. When compared to Naive Bayes Classifier in text categorization, SVM outperformed NBC )

.In contrast to NBC and Maximum Entropy, SVM is large margin classifier, rather than probabilistic classifier. In case of two class label, the procedure for training is finding a hyperplane that is represented by the vector ,which separates, not only document classes of

the two classes, but for the margin, as large as possible.

The corresponding constrained optimization search problem is:

*Let cj be the class and cj ∈ {1,-1} (1 for positive and -1 for negative).*

*Let dj be the document it belongs to the class cj.*

*The solution for the above search problem can be put as:*

*: −∑αjcj j, αj>0*

where we can obtain the αj’s from the solutions of dual optimization problem.The documents j whose αj values are greater than 0 are nothing but support vectors.These are the document vectors that only contribute to vector αj. The classification of test instances could be made by considering the side of , hyperplane they fall on.

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges.

However,

it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well **(look at the below snapshot)**.



**3.2.2**

**NLP(NATURAL LANGUAGE PROCESSING):**

Natural language processing (NLP) is a field of artificial intelligence in which computers analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, [sentiment analysis](https://algorithmia.com/blog/introduction-sentiment-analysis/), speech recognition, and topic segmentation.

Apart from common word processor operations that treat text like a mere sequence of symbols, NLP considers the hierarchical structure of language: several words make a phrase, several phrases make a sentence and, ultimately, sentences convey ideas,

NLP systems have long filled useful roles, such as correcting grammar, converting speech to text and automatically translating between languages.

[NLP is used to analyze text](https://blog.algorithmia.com/2015/09/getting-started-with-natural-language-processing/), allowing machines to [understand how human’s speak](https://en.wikipedia.org/wiki/Artificial_intelligence" \l "Natural_language_processing_.28communication.29" \t "https://algorithmia.com/blog/_blank). This human-computer interaction enables real-world applications like [automatic text summarization](https://en.wikipedia.org/wiki/Automatic_summarization" \t "https://algorithmia.com/blog/_blank), [sentiment analysis](https://en.wikipedia.org/wiki/Sentiment_analysis" \t "https://algorithmia.com/blog/_blank), [topic extraction](https://en.wikipedia.org/wiki/Terminology_extraction" \t "https://algorithmia.com/blog/_blank), [named entity recognition](https://en.wikipedia.org/wiki/Named-entity_recognition" \t "https://algorithmia.com/blog/_blank), [parts-of-speech tagging](https://en.wikipedia.org/wiki/Part-of-speech_tagging" \t "https://algorithmia.com/blog/_blank), [relationship extraction](https://en.wikipedia.org/wiki/Relationship_extraction" \t "https://algorithmia.com/blog/_blank), [stemming](https://en.wikipedia.org/wiki/Stemming" \t "https://algorithmia.com/blog/_blank), and more. NLP is commonly used for [text mining](https://en.wikipedia.org/wiki/Text_mining" \t "https://algorithmia.com/blog/_blank), [machine translation](https://en.wikipedia.org/wiki/Machine_translation" \t "https://algorithmia.com/blog/_blank), and [automated question answering](https://en.wikipedia.org/wiki/Question_answering" \t "https://algorithmia.com/blog/_blank).

NLP is characterized as a difficult problem in computer science. Human language is rarely precise, or plainly spoken. To understand human language is to understand not only the words, but the concepts and how they’re [linked together to create meaning](http://research.microsoft.com/en-us/groups/nlp/" \t "https://algorithmia.com/blog/_blank). Despite language being one of the easiest things for the human mind to learn, the ambiguity of language is what makes natural language processing a difficult problem for computers to master.

* **NLP LIBRARIES ALLOW YOU TO:**
* **Summarize blocks of text** using [Summarizer](https://algorithmia.com/algorithms/nlp/Summarizer?utm_source=blog&utm_medium=post&utm_campaign=nlp" \t "https://algorithmia.com/blog/_blank) to extract the most important and central ideas while ignoring irrelevant information.
* Create a****chat bot**** using [Parsey McParseface](https://algorithmia.com/algorithms/deeplearning/Parsey), a language parsing deep learning model made by [Google that uses Point-of-Speech tagging](https://research.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html" \t "https://algorithmia.com/blog/_blank).
* **Automatically generate keyword tags** from content using [AutoTag](https://algorithmia.com/algorithms/nlp/AutoTag?utm_source=blog&utm_medium=post&utm_campaign=nlp" \t "https://algorithmia.com/blog/_blank), which leverages LDA, a technique that discovers topics contained within a body of text.
* **Identify the type of entity extracted**, such as it being a person, place, or organization using [Named Entity Recognition](https://algorithmia.com/algorithms/StanfordNLP/NamedEntityRecognition?utm_source=blog&utm_medium=post&utm_campaign=nlp" \t "https://algorithmia.com/blog/_blank).
* Use [Sentiment Analysis](https://algorithmia.com/algorithms/nlp/SentimentAnalysis?utm_source=blog&utm_medium=post&utm_campaign=nlp" \t "https://algorithmia.com/blog/_blank) to **identify the sentiment of a string of text**, from very negative to neutral to very positive.
* **Reduce words to their root**, or stem, using [PorterStemmer](https://algorithmia.com/algorithms/codeb34v3r/PorterStemmer?utm_source=blog&utm_medium=post&utm_campaign=nlp" \t "https://algorithmia.com/blog/_blank), or **break up text into tokens** using [Tokenizer](https://algorithmia.com/algorithms/ApacheOpenNLP/Tokenizer?utm_source=blog&utm_medium=post&utm_campaign=nlp" \t "https://algorithmia.com/blog/_blank).
* In our project we have made use of the NLTK LIBRARY(natural language toolkit).

The system we are using in our project is based on the natural language processing method along with which we will be using a basic naive bayes Machine leaning algorithm ,Which has already been explained above.

The task of preprocessing the data and cleaning of the tweets using stemming, and tokenizing needs to be done to further prepare features that we will use to train our machine learning model.

The implementation of all these steps will be discussed in detail in the below sections.

Our main aim is to get a good precision score and f-1 score, and most importantly to be able to get a good accuracy on the developed model.

**3.3 FEASIBILITY ANALYSIS**

A feasibility study is a preliminary study which investigates the information of prospective users and determines the resources requirements, costs, benefits and feasibility of proposed system.

A feasibility study takes into account various constraints within which the system should be implemented and operated. In this stage, the resource needed for the implementation such as computing equipment, manpower and costs are estimated. The estimated are compared with available resources and a cost benefit analysis of the system is made.

The feasibility analysis activity involves the analysis of the problem and collection of all relevant

information relating to the project.

The main objectives of the feasibility study are to determine whether the project would be feasible in terms of economic feasibility, technical feasibility and operational feasibility and schedule feasibility or not. It is to make sure that the input data which are required for the project are available.

Thus we evaluated the feasibility of the system in terms of the following categories:

 Technical feasibility

 Operational feasibility

 Economic feasibility

**3.3.1 Technical Feasibility**

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at the point in time there is no any detailed designed of the system, making it difficult to access issues like performance, costs (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis; understand the different technologies involved in the proposed system.

Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system. Is the required technology available? Our system is technically feasible since all the required tools are easily available. Python with

can be easily handled. Although all tools seems to be easily available

there are challenges too.

**3.3.2 Operational Feasibility**

Proposed project is beneficial only if it can be turned into information systems that will meet the operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation?

The proposed was to make a simplified machine learning model that can be implemented further. It is simpler to operate and can be integrated in any web pages. It is free and not costly to operate.

**3.3.3 Economic Feasibility**

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case.

In addition, this proves to be useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These

could increase improvement in product quality, better decision making, and timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information.

This is a web based application. Creation of application is not costly.

**3.4 Requirement Definition**

After the extensive analysis of the problems in the system, we are familiarized with

the requirement that the current system needs. The requirement that the system

needs is categorized into the functional and non-functional requirements. These

requirements are listed below:

**3.41 Functional Requirements**

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows:

System should be able to process new tweets stored in repositories after retrieval

 System should be able to analyze data and classify each tweet polarity

**3.4.2 Non-Functional Requirements**

Non-functional requirements is a description of features, characteristics and

attribute of the system as well as any constraints that may limit the boundaries of the proposed system.

The non-functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Based on these the non-functional requirements are as follows:

Should be based on an simple yet efficient model

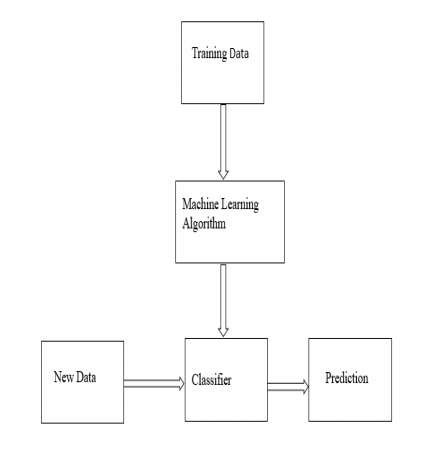
System should provide better accuracy

**CHAPTER 4: SYSTEM DESIGN AND DEVELOPEMENT**

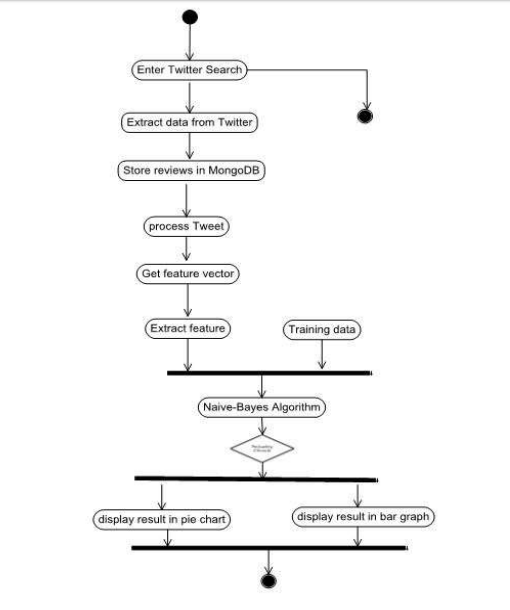
**CHAPTER 4:**

**SYSTEM DESIGN AND DEVELOPEMENT**

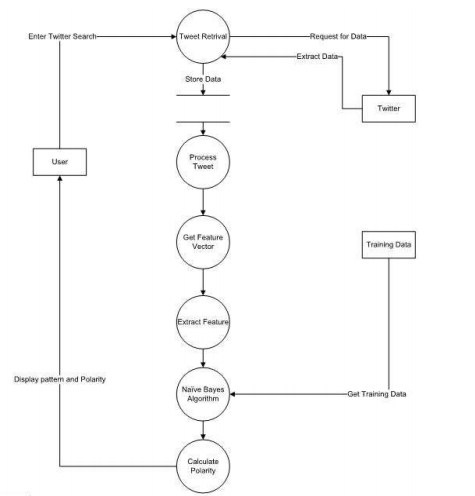
4.1 SYSTEM FLOW DIAGRAM:

****

4.2 ACTIVITY DIAGRAM:

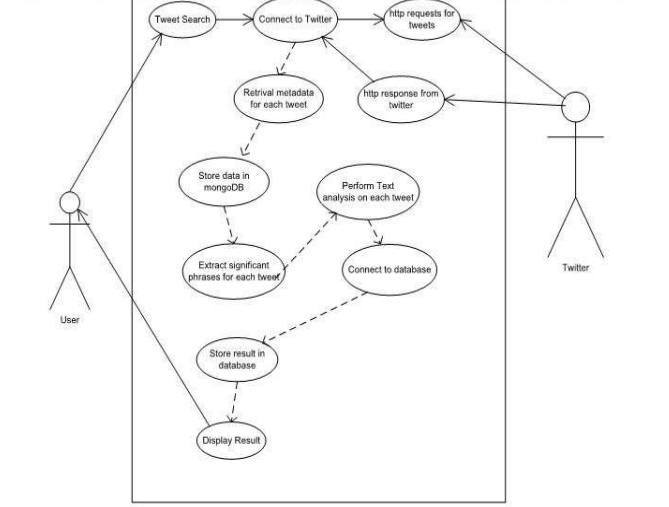


4.3 DATAFLOW DIAGRAM:





4.4 USE CASE DIAGRAM:



**CHAPTER 5: TOOLS AND TECHNOLOGIES**

**CHAPTER 5**

**TOOLS AND TECHNOLOGIES**

**5.1 HARDWARE REQUIREMENTS:**

Hardware requirements for implementing our specified system:

• Ubuntu machine/Windows 10

- RAM: 16 GB

- SSD: 120 GB (minimum)

- GPU: Nvidia GTX 10 series /AMD -a10/intel core( I-5 or I-7)

- Jupyter notebook : Provided by Anaconda Navigator

-Alternatively you can also use Google-Co lab for running your code

• High speed internet for best performance

• Functional peripheral I/O devices

**5.2 SOFTWARE REQUIREMENTS**:

Software requirements for implementing our specified system:

• **Packages:**

- SciKit learn

- Seaborn

- Regular Expression

- NLTK (natural language toolkit)

- Pandas

- NumPy

- Matplot\_lib

- wordcloud

• Google Chrome or Chromium browser (version 65 or above)

**SERVICES:**

**• GOOGLE COLAB**

- Colab is ideal for everything from improving your Python coding skills to 

working with deep learning libraries, like PyTorch, Keras, TensorFlow, and

OpenCV.

- You can create notebooks in Colab, upload notebooks, store notebooks,

share notebooks, mount your Google Drive and use whatever you’ve got

stored in there, import most of your favorite directories, upload your

personal Jupyter Notebooks, upload notebooks directly from GitHub, upload

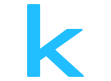
Kaggle files, download your notebooks, and do just about everything else

that you might want to be able to do.

**ANACONDA NAVIGATOR**  

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

* **KAGGLE** 

The best place to discover and seamlessly analyze open data. You can find any dataset of your interest and apply analytical methods to fine tune your analytical capabilities.

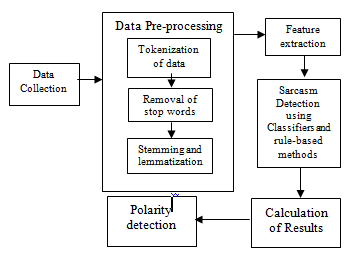
It is a platform for predictive modelling and analytics competitions in which companies and researchers post data and statisticians and data miners compete to produce the best models for predicting and describing the data.

**CHAPTER 6: IMPLEMENTATION**

**CHAPTER 6**

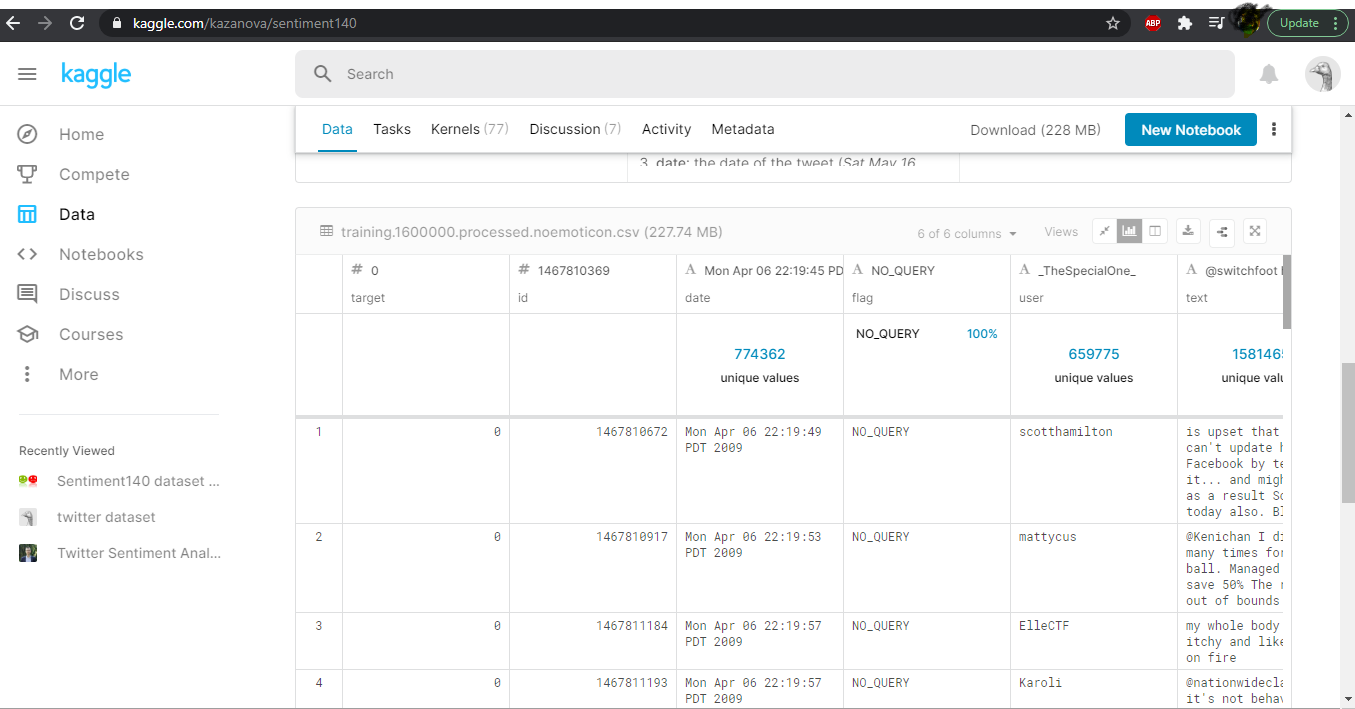
**IMPLEMENTATION**

**GENERAL WORKFLOW:**

****

**6.1 collecting the data:**

* Openup your chrome browser and seach for kaggle
* In kaggle search bar type twitter sentiment dataset and you will find your data set.



* Now click on the download icon to start downloading your dataset.
* You will find all your data ina .csv format.

**6.2 PREPROCESSING AND CLEANING OF DATA**:

The extraction of keywords becomes difficult due to the presence of slangs and incorrect spellings in tweets. Thus, a preprocessing step is performed for filtering out the slang words and misspellings before extracting the features. Any slang words present in the tweets are replaced with their relevant meanings using the slang word dictionary. The slang word dictionary is created using the domain information.

If the data is arranged in a structured format then it becomes easier to find the right information.

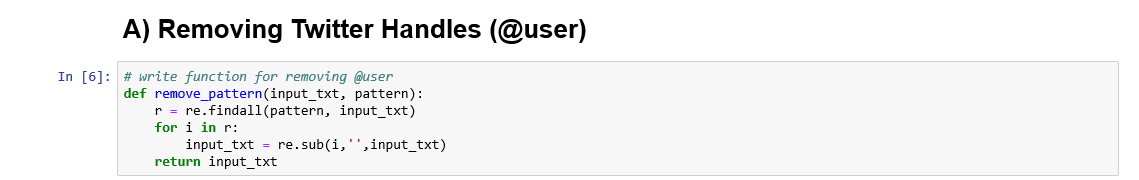
The preprocessing of the text data is an essential step as it makes the raw text ready for mining, i.e., it becomes easier to extract information from the text and apply machine learning algorithms to it. If we skip this step then there is a higher chance that you are working with noisy and inconsistent data. The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don’t carry much weightage in context to the text.

In one of the later stages, we will be extracting numeric features from our Twitter text data. This feature space is created using all the unique words present in the entire data. So, if we preprocess our data well, then we would be able to get a better quality feature space.

* **Removing ‘@’ symbols from user handles:**

We have to remove all such unnecessary symbols that don’t add credibility to the text for classification of any sort.

As mentioned above, the tweets contain lots of twitter handles (@user), that is how a Twitter user acknowledged on Twitter. We will remove all these twitter handles from the data as they don’t convey much information.

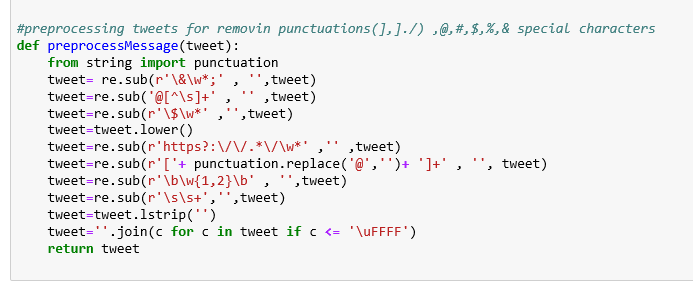


* **Removing Punctuations, Numbers and Special Characters**

punctuations, numbers and special characters do not help much. It is better to remove them from the text just as we removed the twitter handles. Here we will replace everything except characters and hashtags with spaces.

Annotation 3

* You can also create a function to substitute all such text with white space using REGULAR EXPRESSION



### **Removing Short Words**

We have to be a little careful here in selecting the length of the words which we want to remove. So, I have decided to remove all the words having length 3 or less. For example, terms like “hmm”, “oh” are of very little use. It is better to get rid of them.



* **TOKENIZATION.**

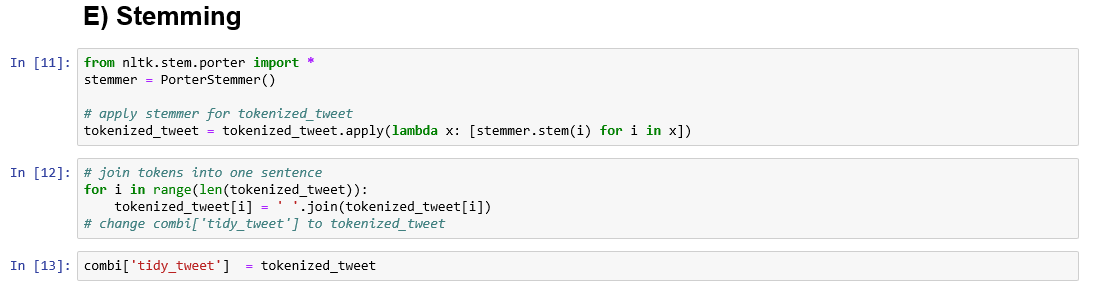
we will tokenize all the cleaned tweets in our dataset. Tokens are individual terms or words, and tokenization is the process of splitting a string of text into tokens.

Annotation TOKEN

* **STEMMING:**

Stemming is a rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word. For example, For example – “play”, “player”, “played”, “plays” and “playing” are the different variations of the word – “play”.

In our project we are using “porter stemmer” which is a stemming tool used to filter out all ing at the end of words , which are basically just unnnecessary letters that have little or no impact whatsoever on the base classification .



**6.3 STORY GENERATION AND VISUALIZATION OF TWEETS**

* What are the most common words in the entire dataset?
* What are the most common words in the dataset for negative and positive tweets, respectively?
* How many hashtags are there in a tweet?
* Which trends are associated with my dataset?
* Which trends are associated with either of the sentiments? Are they compatible with the sentiments?

These are some of the basic questions whose answers we need to find via visualization

We are using WORDCLOUD library to visualize these categories.

A wordcloud is a visualization wherein the most frequent words appear in large size and the less frequent words appear in smaller sizes.

Let’s visualize all the **COMMON** **WORDS** in our data using the wordcloud plot.

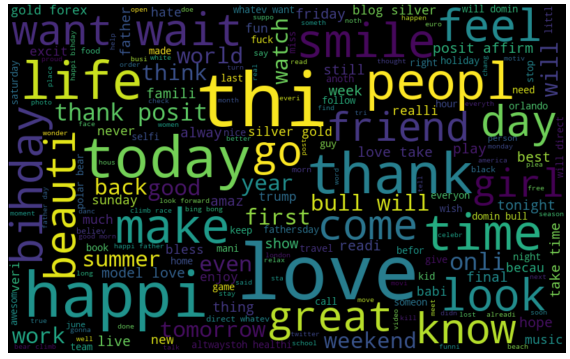
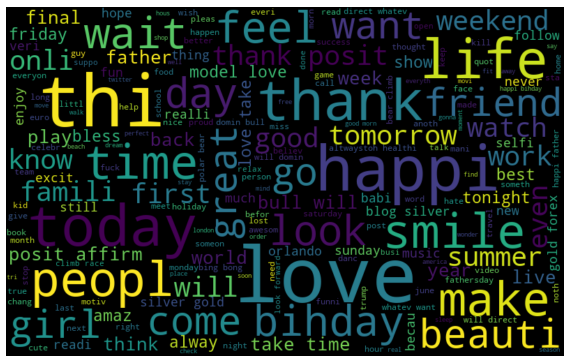


Fig- depicting the most common words in a dataset of TWEETS.

We can see most of the words are positive or neutral. With *happy* and *love* being the most frequent ones. It doesn’t give us any idea about the words associated with the racist/sexist tweets. Hence, we will plot separate wordclouds for both the classes(racist/sexist or not) in our train data.

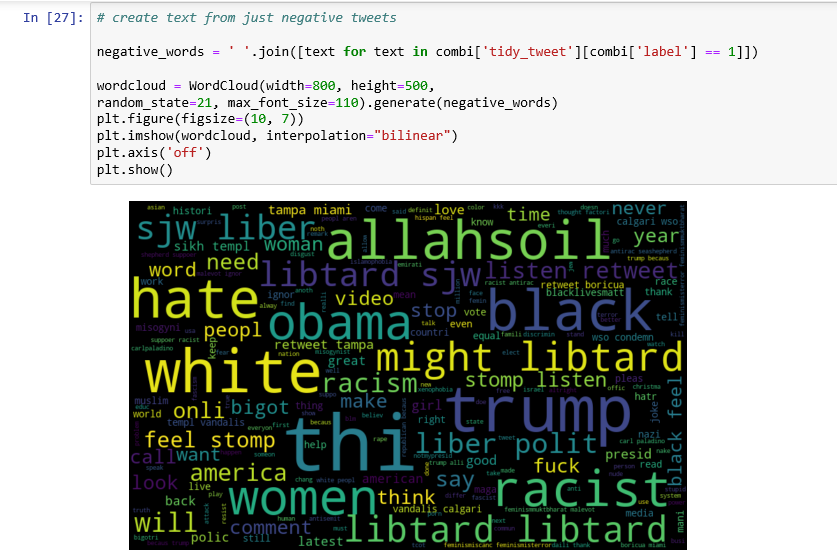
* **NON RACIST/NON SEXIST TWEETS**





We can see most of the words are positive or neutral. With thank*, love,*and today being the most frequent ones. Hence, most of the frequent words are compatible with the sentiment which is non racist/sexists tweets. Similarly, we will plot the word cloud for the other sentiment. Expect to see negative, racist, and sexist terms.

* **RACIST/SEXIST TWEETS.**

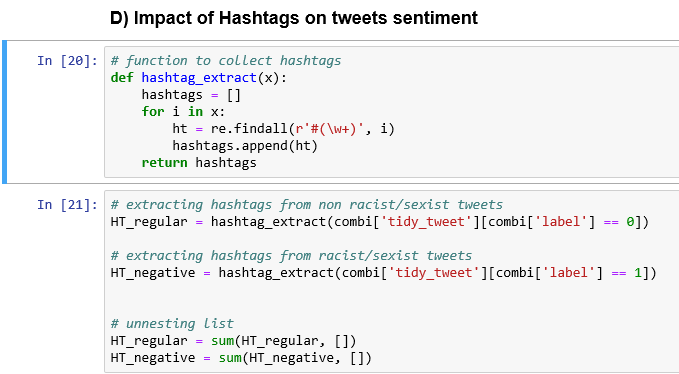


As we can clearly see, most of the words have negative connotations. So, it seems we have a pretty good text data to work on. Next we will the hashtags/trends in our twitter data.

### **D) Understanding the impact of Hashtags on tweets sentiment**

Hashtags in twitter are synonymous with the ongoing trends on twitter at any particular point in time. We should try to check whether these hashtags add any value to our sentiment analysis task, i.e., they help in distinguishing tweets into the different sentiments.

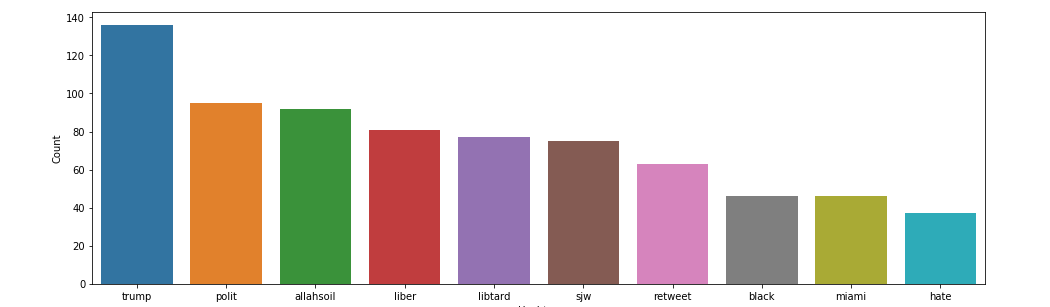
We will now create a function to collect hashtags , for simplification of understanding ,we will create two seperate lists one for non-racist/non-sexist tweets and the other for racist/sexist tweets



Now that we have prepared our lists of hashtags for both the sentiments, we can plot the top n hashtags. So, first let’s check the hashtags in the non-racist/sexist tweets.

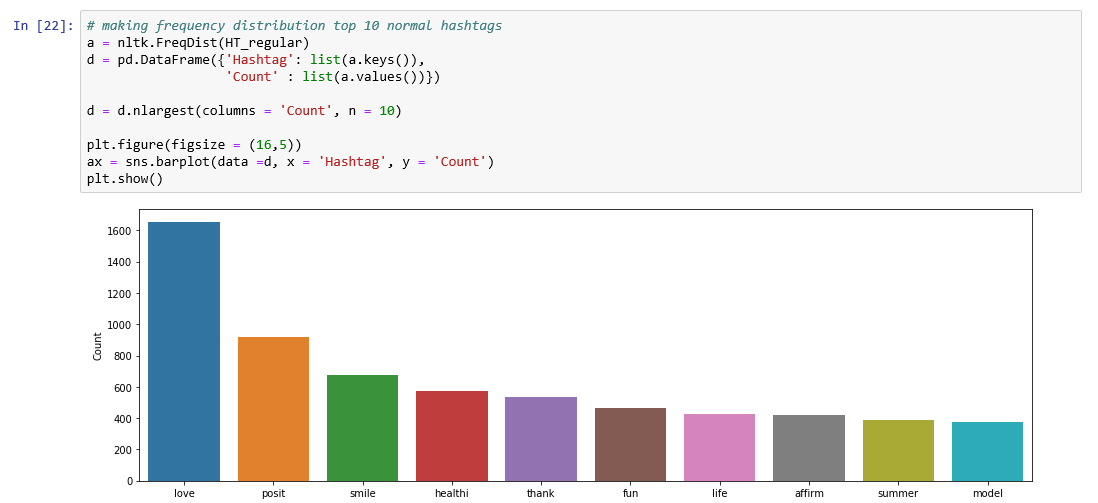
**RACIST/SEXIST TWEETS:**

* Plotting a frequency distribution barplot for the most negative tweets in the given dataset
* most of the terms are negative with a few neutral terms as well. So, it’s not a bad idea to keep these hashtags in our data as they contain useful information. Next, we will try to extract features from the tokenized tweets.



**NON-RACIST/NON-SEXIST TWEETS:**

* Plotting a frequency distribution barplot for the most positive tweets in the given dataset
* these hashtags are positive and it makes sense.

****

**6.4 EXTRACTING FEATURES FROM THE CLEANED TWEETS:**

**BAG OF WORDS**

A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms.

The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents.

A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

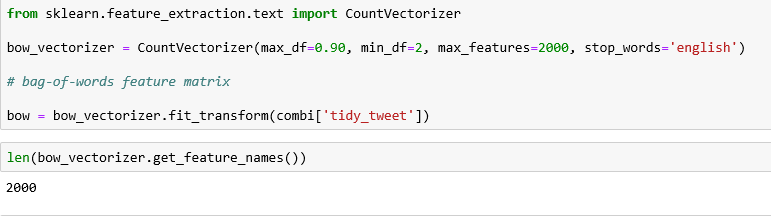
1. A vocabulary of known words.

2. A measure of the presence of known words.

It is called a “bag” of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

The intuition is that documents are similar if they have similar content. Further, that from the content alone we can learn something about the meaning of the document.

The bag-of-words can be as simple or complex as you like. The complexity comes both in deciding how to design the vocabulary of known words (or tokens) and how to score the presence of known words



### **TF-IDF:**

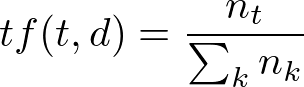
A problem with scoring word frequency is that highly frequent words start to dominate in the document (e.g. larger score), but may not contain as much “informational content” to the model as rarer but perhaps domain specific words.

One approach is to rescale the frequency of words by how often they appear in all documents, so that the scores for frequent words like “the” that are also frequent across all documents are penalized.

This approach to scoring is called Term Frequency – Inverse Document Frequency, or TF-IDF for short, where:

****Term Frequency****: is a scoring of the frequency of the word in the current document.

Term Frequency is just ratio number of current word to the number of all words in document/string/etc.



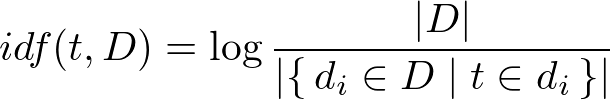
Term Frequency Formula

Frequency of term t\_i, where n\_t — the number of t\_i in current document/string, the sum of n\_k is the number of all terms in current document/string.

****Inverse Document Frequency****: is a scoring of how rare the word is across documents.

The scores are a weighting where not all words are equally as important or interesting.

Inverse Document Frequency is a log of the ratio of the number of all documents/string in the corpus to the number of documents with term t\_i.

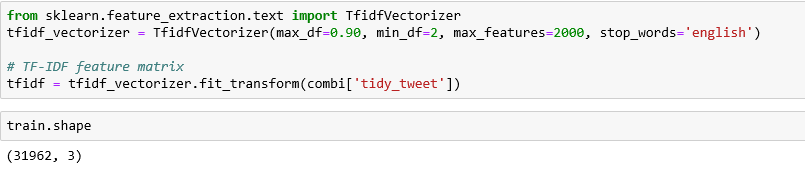


Inverse Document Frequency Formula

tf-idf(t, d, D) is the product tf(t, d) to idf(t, D). .

IMG_259

The scores have the effect of highlighting words that are distinct (contain useful information) in a given document.



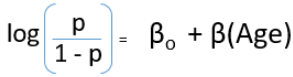
**Figs. - feature extraction using BOW and TF-IDF vectorizer respectively**

**6.5 BUILDING THE MODEL**

We are now done with all the pre-modeling stages required to get the data in the proper form and shape. Now we will be building predictive models on the dataset using the two feature set — Bag-of-Words and TF-IDF.

We will use logistic regression to build the models. It predicts the probability of occurrence of an event by fitting data to a logit function.

The following equation is used in Logistic Regression:

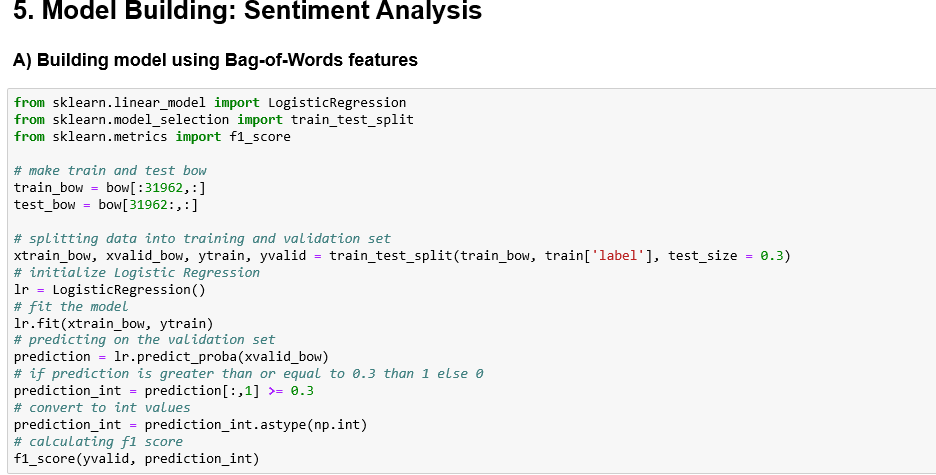


in our project we have used two different approaches to see which one gives better accuracy and f-1 score.

* f1-score=2\*(precision\*recall)/(precision+recall)
* good f1 scores means low false positives and negatives

APPROACH 1

BUILDING THE MODEL USING SENTIMENT ANALYSIS:

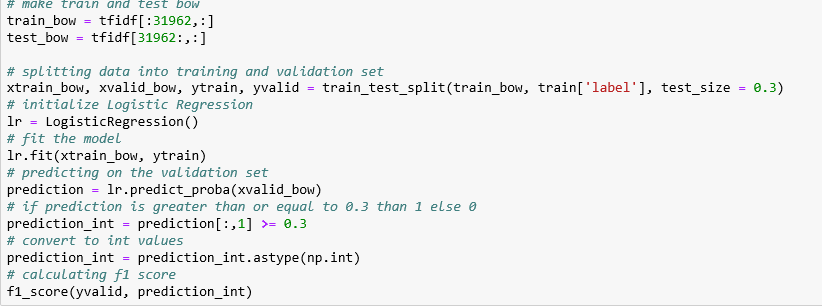


**APPROACH 2**

**BUILDING MODEL USING TF-IDF FEATURES:**

#### This is another method which is based on the frequency method but it is different to the bag-of-words approach in the sense that it takes into account, not just the occurrence of a word in a single document (or tweet) but in the entire corpus.

#### TF-IDF works by penalizing the common words by assigning them lower weights while giving importance to words which are rare in the entire corpus but appear in good numbers in few documents.



**CHAPTER 7: TEST CASES**

**CHAPTER 7:**

**TESTING**

**7.1 Unit Testing**

Unit testing is performed for testing modules against detailed design. Inputs to the process are usually compiled modules from the coding process. Each modules are assembled into a larger unit during the unit testing process.

Testing has been performed on each phase of project design and coding.

We carry out the testing of module interface to ensure the proper flow of information into and out of the program unit while testing. We make sure that the temporarily stored data maintains its integrity throughout the algorithm's execution by examining the local data structure.

Finally, all error-handling paths are also tested.

**7.2. System Testing**

We usually perform system testing to find errors resulting from unanticipated interaction between the sub-system and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, series of test cases must be developed which ultimately uncover all the possibly existing errors.

Different software techniques can be used for this process. These techniques provide systematic guidance for designing test that

Exercise the internal logic of the software components,

Exercise the input and output domains of a program to uncover errors in program function, behavior and performance.

We test the software using two methods:

*White Box testing****:*** Internal program logic is exercised using this test case design techniques.

*Black Box testing*: Software requirements are exercised using this test case design techniques.

29Both techniques help in finding maximum number of errors with minimal effort and

time.

**7.3. Performance Testing**

It is done to test the run-time performance of the software within the context of integrated system. These tests are carried out throughout the testing process. For example, the performance of individual module are accessed during white box testing under unit testing.

**7.4. Verification and Validation**

The testing process is a part of broader subject referring to verification and validation. We have to acknowledge the system specifications and try to meet the customer’s requirements and for this sole purpose, we have to verify and validate the product to make sure everything is in place. Verification and validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and other is done to ensure if the customer requirements are

properly met or not by the end product.

Verification is more like 'are we building the product right?' and validation is

more like 'are we building the right product?'.

**7.5 **Confusion Matrix:****  
A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.



*Fig: depicting the quadrants of a confusion matrix*

Here,

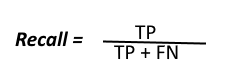
* Class 1 : Positive
* Class 2 : Negative

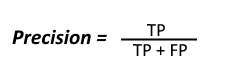
**Definition of the Terms:**

* Positive (P) : Observation is positive (for example: is an apple).
* Negative (N) : Observation is not positive (for example: is not an apple).
* True Positive (TP) : Observation is positive, and is predicted to be positive.
* False Negative (FN) : Observation is positive, but is predicted negative.
* True Negative (TN) : Observation is negative, and is predicted to be negative.
* False Positive (FP) : Observation is negative, but is predicted positive.

**Classification Rate/Accuracy:**  
Classification Rate or Accuracy is given by the relation:  


However, there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

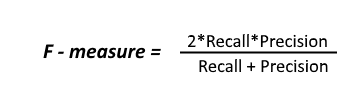
**Recall:**  
  
Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).

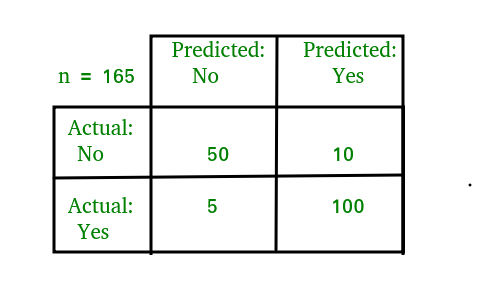
**Precision:**  
  
To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labelled as positive is indeed positive (a small number of FP).

**High recall, low precision:**This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.

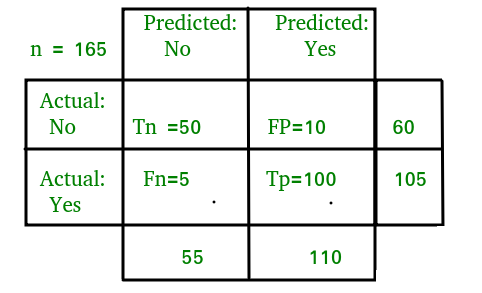
**Low recall, high precision:**This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)

1. **measure:**  
   Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

The F-Measure will always be nearer to the smaller value of Precision or Recall.  


Let’s consider an example now, in which we have infinite data elements of class B and a single element of class A and the model is predicting class A against all the instances in the test data.  
Here,  
Precision : 0.0Recall : 1.0

Now:  
Arithmetic mean: 0.5  
Harmonic mean: 0.0  
When taking the arithmetic mean, it would have 50% correct. Despite being the worst possible outcome! While taking the harmonic mean, the F-measure is 0.

**Example to interpret confusion matrix:**  
  
For the simplification of the above confusion matrix I have added all the terms like TP, FP, etc and the row and column totals in the following image:  
  
Now,

**Classification Rate/Accuracy:**  
Accuracy = (TP + TN) / (TP + TN + FP + FN) = (100 + 50) /(100 + 5 + 10 + 50) = 0.90

**Recall:** Recall gives us an idea about when it’s actually yes, how often does it predict yes.  
Recall = TP / (TP + FN) = 100 / (100 + 5) = 0.95

**Precision:** Precsion tells us about when it predicts yes, how often is it correct.  
Precision = TP / (TP + FP)=100/ (100+10) = 0.91

**F-measure:**  
Fmeasure = (2 \* Recall \* Precision) / (Recall + Presision) = (2 \* 0.95 \* 0.91) / (0.91 + 0.95) = 0.92

**CHAPTER 8: CODING**

**CHAPTER 8**

**CODING**

**8.1- PSEUDO CODE**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import string

import nltk

import warnings

warnings.filterwarnings("ignore", category=DeprecationWarning)

import matplotlib.pyplot as plt

import sklearn

from sklearn.model\_selection import train\_test\_split

from sklearn import preprocessing

from sklearn.preprocessing import LabelEncoder #convert target into numbers

from sklearn import metrics #performance of the model

import seaborn as sn

from sklearn.metrics import confusion\_matrix

import re #regular expression find replace etc word or letter in a string

from sklearn import model\_selection , preprocessing, naive\_bayes, metrics #naive bayes

from sklearn.feature\_extraction.text import TfidfVectorizer

#converts target data into numbers remover strings in numbers

**#term frequency creates vrctor of word assigned as a number to its frequency.**

%matplotlib inline

train = pd.read\_csv('train\_tweets.csv')

test = pd.read\_csv('test\_tweets.csv')

train.head()

combi = train.append(test, ignore\_index = True)

combi.head()

**# write function for removing @user**

def remove\_pattern(input\_txt, pattern):

r = re.findall(pattern, input\_txt)

for i in r:

input\_txt = re.sub(i,'',input\_txt)

return input\_txt

import re

combi['tidy\_tweet'] = np.vectorize(remove\_pattern)(combi['tweet'], '@[\w]\*')

**# remove special characters, numbers, punctuations**

combi['tidy\_tweet'] = combi['tidy\_tweet'].str.replace('[^a-zA-Z#]+',' ')

**# remove short words**

combi['tidy\_tweet'] = combi['tidy\_tweet'].apply(lambda x: ' '.join([w for w in x.split() if len(w) > 3]))

**# create new variable tokenized tweet**

tokenized\_tweet = combi['tidy\_tweet'].apply(lambda x: x.split())

from nltk.stem.porter import \*

stemmer = PorterStemmer()

**# apply stemmer for tokenized\_tweet**

tokenized\_tweet = tokenized\_tweet.apply(lambda x: [stemmer.stem(i) for i in x])

**# join tokens into one sentence**

for i in range(len(tokenized\_tweet)):

tokenized\_tweet[i] = ' '.join(tokenized\_tweet[i])

**# change combi['tidy\_tweet'] to tokenized\_tweet**

combi['tidy\_tweet'] = tokenized\_tweet

import sys

print(sys.path)

['', 'C:\\Users\\Jayant\\Anaconda3\\python36.zip', 'C:\\Users\\Jayant\\Anaconda3\\DLLs', 'C:\\Users\\Jayant\\Anaconda3\\lib', 'C:\\Users\\Jayant\\Anaconda3', 'C:\\Users\\Jayant\\Anaconda3\\lib\\site-packages', 'C:\\Users\\Jayant\\Anaconda3\\lib\\site-packages\\win32', 'C:\\Users\\Jayant\\Anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\Users\\Jayant\\Anaconda3\\lib\\site-packages\\Pythonwin', 'C:\\Users\\Jayant\\Anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\Jayant\\.ipython']

import wordcloud

all\_words = ' '.join([text for text in combi['tidy\_tweet']])

from wordcloud import WordCloud

wordcloud = WordCloud(width=800, height=500, random\_state=21, max\_font\_size=110).generate(all\_words)

plt.figure(figsize=(10, 7))

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis('off')

plt.show()

normal\_words = ' '.join([text for text in combi['tidy\_tweet'][combi['label'] == 0]])

wordcloud = WordCloud(width=800, height=500, random\_state=21, max\_font\_size=110).generate(normal\_words)

plt.figure(figsize=(10, 7))

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis('off')

plt.show()

negative\_words = ' '.join([text for text in combi['tidy\_tweet'][combi['label'] == 1]])

wordcloud = WordCloud(width=800, height=500,

random\_state=21, max\_font\_size=110).generate(negative\_words)

plt.figure(figsize=(10, 7))

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis('off')

plt.show()

**# function to collect hashtags**

def hashtag\_extract(x):

hashtags = []

for i in x:

ht = re.findall(r'#(\w+)', i)

hashtags.append(ht)

return hashtags

**# extracting hashtags from non racist/sexist tweets**

HT\_regular = hashtag\_extract(combi['tidy\_tweet'][combi['label'] == 0])

**# extracting hashtags from racist/sexist tweets**

HT\_negative = hashtag\_extract(combi['tidy\_tweet'][combi['label'] == 1])

**# unnesting list**

HT\_regular = sum(HT\_regular, [])

HT\_negative = sum(HT\_negative, [])

**# making frequency distribution top 10 normal hashtags**

a = nltk.FreqDist(HT\_regular)

d = pd.DataFrame({'Hashtag': list(a.keys()),

'Count' : list(a.values())})

d = d.nlargest(columns = 'Count', n = 10)

plt.figure(figsize = (16,5))

ax = sns.barplot(data =d, x = 'Hashtag', y = 'Count')

plt.show()

**# making frequency distribution top 10 negative hashtag**s

a = nltk.FreqDist(HT\_negative)

d = pd.DataFrame({'Hashtag': list(a.keys()),

'Count' : list(a.values())})

d = d.nlargest(columns = 'Count', n = 10)

plt.figure(figsize = (16,5))

ax = sns.barplot(data =d, x = 'Hashtag', y = 'Count')

plt.show()

from sklearn.feature\_extraction.text import CountVectorizer

bow\_vectorizer = CountVectorizer(max\_df=0.90, min\_df=2, max\_features=2000, stop\_words='english')

**# bag-of-words feature matrix**

bow = bow\_vectorizer.fit\_transform(combi['tidy\_tweet'])

len(bow\_vectorizer.get\_feature\_names())

from sklearn.feature\_extraction.text import TfidfVectorizer

tfidf\_vectorizer = TfidfVectorizer(max\_df=0.90, min\_df=2, max\_features=2000, stop\_words='english')

**# TF-IDF feature matrix**

tfidf = tfidf\_vectorizer.fit\_transform(combi['tidy\_tweet'])

train.shape

**#Building model using Bag-of-Words features**

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import f1\_score

# make train and test bow

train\_bow = bow[:31962,:]

test\_bow = bow[31962:,:]

**# splitting data into training and validation set**

xtrain\_bow, xvalid\_bow, ytrain, yvalid = train\_test\_split(train\_bow, train['label'], test\_size = 0.3)

**# initialize Logistic Regression**

lr = LogisticRegression()

**# fit the model**

lr.fit(xtrain\_bow, ytrain)

**# predicting on the validation set**

prediction = lr.predict\_proba(xvalid\_bow)

**# if prediction is greater than or equal to 0.3 than 1 else 0**

prediction\_int = prediction[:,1] >= 0.3

**# convert to int values**

prediction\_int = prediction\_int.astype(np.int)

**# calculating f1 score**

f1\_score(yvalid, prediction\_int)

**# submit your file**

**# writing data to a CSV file**

test\_pred = lr.predict\_proba(test\_bow)

**# if prediction is greater than or equal to 0.3 than 1 else 0**

test\_pred\_int = test\_pred[:,1] >= 0.3

**# convert to int values**

test\_pred\_int = test\_pred\_int.astype(np.int)

test['label'] = test\_pred\_int

submission = test[['id', 'label']]

submission.to\_csv('lr\_bow\_sub.csv', index=False)

**Building model using TF-IDF features**

**# make train and test bow**

train\_bow = tfidf[:31962,:]

test\_bow = tfidf[31962:,:]

**# splitting data into training and validation set**

xtrain\_bow, xvalid\_bow, ytrain, yvalid = train\_test\_split(train\_bow, train['label'], test\_size = 0.3)

**# initialize Logistic Regression**

lr = LogisticRegression()

**# fit the model**

lr.fit(xtrain\_bow, ytrain)

**# predicting on the validation set**

prediction = lr.predict\_proba(xvalid\_bow)

**# if prediction is greater than or equal to 0.3 than 1 else 0**

prediction\_int = prediction[:,1] >= 0.3

**# convert to int values**

prediction\_int = prediction\_int.astype(np.int)

**# calculating f1 score**

f1\_score(yvalid, prediction\_int)

# submit your file

# writing data to a CSV file

test\_pred = lr.predict\_proba(test\_bow)

**# if prediction is greater than or equal to 0.3 than 1 else 0**

test\_pred\_int = test\_pred[:,1] >= 0.3

**# convert to int values**

test\_pred\_int = test\_pred\_int.astype(np.int)

test['label'] = test\_pred\_int

submission = test[['id', 'label']]

submission.to\_csv('lr\_tfidf\_sub.csv', index = False)

**#get accuracy for the model**

metrics.accuracy\_score(yvalid,prediction\_int)

**#get the confusion matrix**

cm=confusion\_matrix(yvalid,prediction\_int)

conf\_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual:1'])

plt.figure(figsize =(8,5))

sn.heatmap(conf\_matrix, annot=True ,fmt='d', cmap="YlGnBu")

**CHAPTER 9: CONCLUSION**

**CHAPTER 9**

**CONCLUSION**

In conclusion sentiment analysis is: it’s a tremendously difficult task even for human beings. That said, sentiment analysis classifiers might not be as precise as other types of classifiers. Remember that inter-annotator agreement is pretty low and that machines learn from the data they are fed with (see above).

Chances are that sentiment analysis predictions will be wrong from time to time, but by using sentiment analysis you will get the opportunity to get it right about 70-80% of the times you submit your texts for classification.

If you or your company have not used sentiment analysis before, then you’ll see some improvement really quickly. For typical use cases, such as ticket routing, brand monitoring, and VoC analysis , this means you will save a lot of time and money -which you are likely to be investing in in-house manual work nowadays,- save your teams some frustration, and increase your (or your company’s) productivity.

#### sentiment analysis can be used to:

* Analyze tweets and/or facebook posts over a period of time to detect sentiment of a

particular audience

* Monitor social media mentions of your brand and automatically categorize by urgency
* Automatically route social media mentions to team members best fit to respond
* Automate any or all of these processes
* Gain deep insights into what’s happening across your social media channels

Sentiment analysis is a very wide branch for research.We have covered some of the important aspects. We plan ahead to improve our algorithm used for determining the sentiment value.

Also the project as of now can also be expanded to other social media platform usages like movie reviews(IMDB reviews), personal blogs. The accuracy achieved is also mentioned below.

Emoticons and the use of hashtags for the sentiment evaluation is a very important inference related to sentiment analysis of social media data.

Our project uses hashtags but the use of emoticons to determine the context of the tweet is not done. Hence with the current limitations the accuracy is found to be

**95%.**

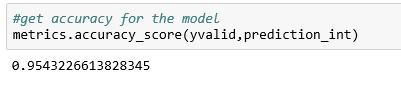
****

Fig.- depicting the accuracy score of the model

**9.1 - FUTURE ENHANCEMENTS**

Based on the models that we have developed and results we have got it is safe to say that it os not the most free flowing task to do sentiment analysis.

To get better model accuracy , however we have seen that if we increase the nimber of features from 1500 to 2000 the accuracy also shoots up.

Meanig the larger the dataset the better the accuracy

Interesting area for future study includes the fluctuations in the performance of sentiment analysis algorithms in cases where multiple features are considered. In other words, combining various features was found to lead to improve the performance in most cases, but substandard performance in others. Thus, an exploration into the causes of these performance instabilities would be an intriguing direction for future works.

Another might be to investigate the data sparsity issue using both ensemble and hybrid approaches. The intention behind this is to measure the robustness of various Twitter sentiment approaches the data sparsity. A further area of study might be the utilization of active learning techniques to detect Twitter sentiments and to increase the confidence of decision makers.

I tentatively conclude that sentiment analysis for Twitter data is not that different from sentiment analysis for other genres. In future work, we will explore even richer linguistic analysis, for example, parsing, semantic analysis and topic modeling.

**CHAPTER 10: BIBLIOGRAPHY**

**WEBSITES :**

- <http://cs229.stanford.edu/>

- <http://www.ijircce.com/>

- [https://www.kaggle.com/](https://www.kaggle.com/search?q=twitter+sentiment)

- <https://nseindia.com/>

- <https://www.analyticsvidhya.com/>

- <https://towardsdatascience.com/>

- <https://ieeexplore.ieee.org/>

-<https://www.semanticscholar.org/>