**PURBANCHAL UNIVERSITY**



**DEPARTMENT OF COMPUTER ENGINEERING**

**KHWOPA ENGINEERING COLLEGE**  
 **LIBALI-2, BHAKTAPUR**

**A PROJECT REPORT**

**ON**

**A COMPARATIVE STUDY OF ALGORITHMS USED IN SENTIMENT ANALYSIS**

Project work submitted in partial fulfillment of requirements for the award of the degree of Bachelor of Engineering in Computer Engineering (Seventh Semester)

**SUBMITTED BY**

1. Arun Prajapati (740305)

2. Neetu Phaiju (740324)

3. Rabin Phaiju (740329)

4. Rodip Duwal (740334)

**SUBMITTED TO**

DEPARTMENT OF COMPUTER ENGINEERING

KHWOPA ENGINEERING COLLEGE

**UNDER THE SUPERVISION OF**

Er. Milan Chikanbanjar

27th February 2021

# **ACKNOWLEDGEMENT**

First of all, we would like to express our sincere thanks to the department of Computer Engineering for providing us the opportunity to work on the project named “**A Comparative Study of algorithms used in Sentiment Analysis”** to explore our abilities and knowledge in the field of Artificial Intelligence and Machine Learning.

Finally, we would like to thank our supervisor **Er. Milan Chikanbanjar** who inspired us to gather different information and conduct proper research on our study of sentiment analysis.

**Project Team Members:**

1. Arun Prajapati (740305)

2. Neetu Phaiju (740324)

3. Rabin Phaiju (740329)

4. Rodip Duwal (740334)

# **ABSTRACT**

Sentiment analysis is the computational study of people’s opinions, sentiments, attitudes and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years. It has a wide range of applications because opinions are key influencers of our behavior. In decision making, the opinions of others have a significant effect. The approaches of text sentiment analysis typically work at a particular level like phrase, sentence or document level. This system aims at analyzing a solution for the sentiment classification.

Keywords: Opinions, Sentiment Analysis, Sentiment Classification, Term Frequency, Inverse Document Frequency

# **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig. No** | **Title** | **Page no.** |
| 1.1 | Sentiment Classification | 1 |
| 1.2 | Sentiment analysis procedure | 2 |
| 3.1 | Work break down structure | 7 |
| 4.1 | The proposed methodology for sentiment analysis | 8 |
| 4.2 | Porter Stemmer examples | 9 |
| 4.3 | Feature Extraction Procedure | 10 |
| 4.4 | Accuracy Comparison between SVM with TF-IDF and SVM with Doc2Vec | 11 |
| 4.5 | Datasets used for our research | 14 |
| 4.6 | Flowchart for Sentiment Analysis | 15 |
| 5.1 | Comparison of different performance measure of 4 different Algorithms for Conservation dataset | 17 |
| 5.2 | Comparison of different performance measure of 4 different Algorithms for Movie dataset | 17 |
| 5.3 | Screenshot of Sentiment Analysis Application | 20 |

# **LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No** | **Title** | **Page no.** |
| 5.1 | Article Summary for Sentiment analysis using Naïve Bayes | 19 |
| 5.2 | Article Summary for Sentiment analysis using Decision Tree | 19 |
| 5.3 | Article Summary for Sentiment analysis using Logistic Regression | 19 |
| 5.4 | Article Summary for Sentiment analysis using KNN | 19 |

# **LIST OF ABBREVIATION**

1. IDF Inverse Document Frequency
2. KNN K-Nearest Neighbor
3. NLP Natural Language Processing
4. POS Part of Speech
5. SA Sentiment Analysis
6. SVC Support Vector Classifier
7. SVM Support Vector Machine
8. TF Term Frequency

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENT ii](#_Toc89770455)

[ABSTRACT iii](#_Toc89770456)

[LIST OF FIGURES iv](#_Toc89770457)

[LIST OF TABLES v](#_Toc89770458)

[LIST OF ABBREVIATION vi](#_Toc89770459)

[CHAPTER 1 1](#_Toc89770460)

[INTRODUCTION 1](#_Toc89770461)

[1.1 Background 1](#_Toc89770462)

[1.2 Motivation 3](#_Toc89770463)

[1.3 Statement of Problems 3](#_Toc89770464)

[1.4 Objectives 3](#_Toc89770465)

[1.5 Scope & Limitation 4](#_Toc89770466)

[CHAPTER 2 5](#_Toc89770467)

[LITERATURE REVIEW 5](#_Toc89770468)

[CHAPTER 3 7](#_Toc89770469)

[PROJECT MANAGEMENT 7](#_Toc89770470)

[3.1 Team Members 7](#_Toc89770471)

[3.2 Feasibility Study 7](#_Toc89770472)

[3.3 Work break down structure 7](#_Toc89770473)

[CHAPTER 4 8](#_Toc89770474)

[METHODOLOGY 8](#_Toc89770475)

[4.1 Background 8](#_Toc89770476)

[4.2 Datasets 13](#_Toc89770477)

[4.3 Comparison Measure 14](#_Toc89770478)

[4.4 Flowchart 15](#_Toc89770479)

[4.5 Tools and Platform 16](#_Toc89770480)

[CHAPTER 5 17](#_Toc89770481)

[RESULT AND DISCUSSION 17](#_Toc89770482)

[5.1 RESULT ANALYSIS 17](#_Toc89770483)

[5.2 RESEARCH ANALYSIS 19](#_Toc89770484)

[5.3 THE SENTIMENT ANALYZER 20](#_Toc89770485)

[5.4 REMAINING WORK 20](#_Toc89770486)

[REFERENCES 21](#_Toc89770487)

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Background**

Sentiment analysis (or opinion mining) is a natural language processing technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs [1].

There are many researches that focus on sentiment classiﬁcation, with different goals and different supervised and unsupervised methods. These are usually based on machine learning models (e.g., support vector machines, maximum entropy, time series, Bayes, Hoeffding trees, artiﬁcial neural networks, etc.). Some of them use lexical resources and different features (e.g., unigrams, bigrams, etc.). The accuracy of such methods is incrementally better, being their main goal, the extraction of the sentiment based on the subjectivity and linguistic features of the words used in an unstructured text. Thus, it is quite common the division into two well differentiated types of methods for SA:

• Machine learning based methods - These methods are also divided in groups: supervised and unsupervised techniques [11], together with semi-supervised learning, a hybrid between them.

• Lexicon based methods - These techniques are based on dictionaries of words annotated with their semantic polarity [20].

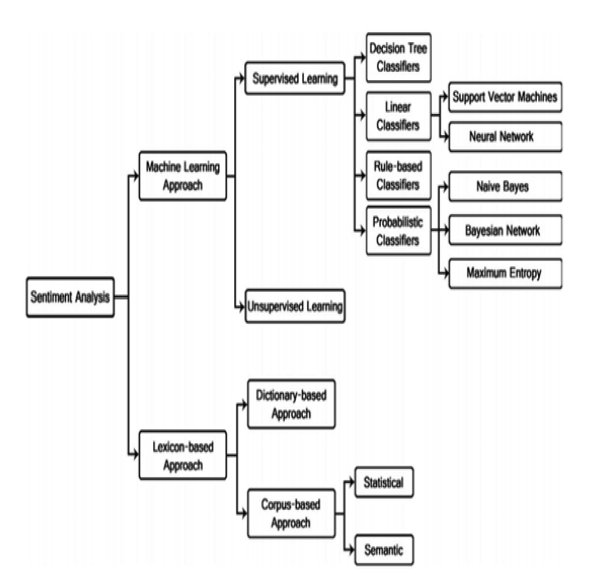


Fig 1.1: Sentiment Classification

Sentiment Analysis can be considered a classification process as illustrated in Fig. 1.2. There are three main classification levels in SA: document-level, sentence-level, and aspect-level SA. Document-level SA aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document a basic information unit (talking about one topic). Sentence-level SA aims to classify sentiment expressed in each sentence. The first step is to identify whether the sentence is subjective or objective. If the sentence is subjective, Sentence-level SA will determine whether the sentence expresses positive or negative opinions. Wilson et al. [20] have pointed out that sentiment expressions are not necessarily subjective in nature. However, there is no fundamental difference between document and sentence level classifications because sentences are just short documents [10]. Classifying text at the document level or at the sentence level does not provide the necessary detail needed opinions on all aspects of the entity which is needed in many applications, to obtain these details; we need to go to the aspect level. Aspect-level SA aims to classify the sentiment with respect to the specific aspects of entities. The first step is to identify the entities and their aspects. The opinion holders can give different opinions for different aspects of the same entity like this sentence “The voice quality of this phone is not good, but the battery life is long”. This survey tackles the first two kinds of SA.

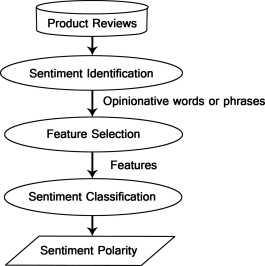


Fig 1.2: Sentiment analysis procedure

In this paper we have presented a comparative study of most commonly used algorithms for sentimental analysis. The task of classification is a very vital task in any system. We present a study of algorithms viz.

1. Naïve Bayes
2. Decision tree
3. Logistic regression
4. KNN
5. Random Forest
6. SVC

We showcase the basic theory behind the algorithms, when they are generally used and their pros and cons including their accuracy, F-score, Recall, Precision etc.

The data sets used in SA are an important issue in this field. The main sources of data are from the product reviews. These reviews are important to the business holders as they can take business decisions according to the analysis results of users’ opinions about their products. The reviews sources are mainly review sites. SA is not only applied on product reviews but can also be applied on stock markets, news articles, or political debates. In political debates for example, we could figure out people’s opinions on a certain election candidates or political parties. The election results can also be predicted from political posts. The social network sites and micro-blogging sites are considered a very good source of information because people share and discuss their opinions about a certain topic freely. They are also used as data sources in the SA process that performs sentiment analysis. In this paper proposes a model to analyses sentiment of different dataset of different source such as political statements, reviews in comments, posts from twitter, Facebook and/or other social media. One of the main tasks to do is the preparation and the process of the corpus as well as the selection of the features that finally will support the process of classification of the comments in positive, neutral, and negative.

### **1.2 Motivation**

Due to the exponential growth of the social network, the sentiment analysis has been applied to analyze the user’s opinions. The majority of the data that is constantly generated in the social networks could contain valuable information like perceptions and tendencies from the users to the objects, personalities, or services.

Due to its popularity nowadays, SA has been expanded to other fields like education, medicine, politics, and others. But reading all reviews is both time and money consuming therefore instead of spending times in reading and figuring out the positivity and negativity of text we can use automated techniques for sentiment analysis.

### **1.3 Statement of Problems**

A major benefit of social media is that we can see the good and bad things people say about the particular brand or personality.

The bigger your company gets difficult it becomes to keep a handle on how everyone feels about your brand. For large companies or a public figure with thousands of daily mentions on social media, news sites and blogs, it’s extremely difficult to do this manually.

To combat this problem, sentimental analysis is necessary. It can be used to evaluate the people's sentiment about particular brand or personality.

### **1.4 Objectives**

* To evaluate persons opinion in certain cases and the performance parameter of different algorithms used for sentiment analysis.

### **1.5 Scope & Limitation**

Sentiment analysis is the area which deals with judgments, responses as well as feelings, which is generated from texts, being extensively used in fields like data mining, web mining, and social media analytics because sentiments are the most essential characteristics to judge the human behavior [2].

In context of existing techniques, there are inadequate accuracy, incapability to deal with complex sentences and inability to perform well in different domains.

# **CHAPTER 2**

## **LITERATURE REVIEW**

In recent years a lot of work has been done in the field of “Sentiment Analysis” by number of researchers. In its early stage it was intended for binary classification which assigns opinions or reviews to bipolar classes such as positive or negative only.

Pak and Paroubek (2010) [5] proposed a model to classify the tweets as objective, positive and negative. They created a twitter corpus by collecting tweets using Twitter API and automatically annotating those tweets using emoticons. Using that corpus, they developed a sentiment classifier based on the multinomial Naive Bayes method that uses features like N gram and POS-tags. The training set they used was less efficient since it contains only tweets having emoticons.

Barbosa et al. (2010) [19] designed a two-phase automatic sentiment analysis method for classifying tweets. They classified tweets as objective or subjective and then in second phase, the subjective tweets were classified as positive or negative. The feature space used included retweets, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS.

In 2011, Han-Xiao Shi, Xiao-Jun Li created a sentiment analysis model for hotel reviews based on supervised learning approach using unigram feature with two types of information (frequency and TF-IDF) to realize polarity classification of documents [17].

Agarwal et al. (2011) [7] developed a 3-way model for classifying sentiment into positive, negative and neutral classes. They experimented with models such as: unigram model, a feature based model and a tree kernel based model. For tree kernel based model uses 100 features and the unigram model uses over 10,000 features. They arrived on a conclusion that features which combine prior polarity of words with their parts-of-speech(pos) tags are most important and plays a major role in the classification task. The tree kernel based model outperformed the other two models.

Po-Wei Liang et. al. (2014) [28] used Twitter API to collect twitter data. Their training data falls in three different categories (camera, movie, mobile). The data is labeled as positive, negative and non-opinions. Tweets containing opinions were filtered. Unigram Naïve Bayes model was implemented and the Naïve Bayes simplifying independence assumption was employed. They also eliminated useless features by using the Mutual Information and Chi square feature extraction method. Finally the orientation of an tweet is predicted i.e. positive or negative.

Pablo et. al. [25] presented variations of Naïve Bayes classifiers for detecting polarity of English tweets. Two different variants of Naïve Bayes classifiers were built namely Baseline(trained to classify tweets as positive, negative and neutral), and Binary (makes use of a polarity lexicon and classifies as positive and negative. Neutral tweets neglected). The features considered by classifiers were Lemmas (nouns, verbs, adjectives and adverbs), Polarity Lexicons, and Multiword from different sources and Valence Shifters.

Xia et al. [29] used an ensemble framework for Sentiment Classification which is obtained by combining various feature sets and classification techniques. In their work, they used two types of feature sets (Part-of-speech information and Word-relations) and three base classifiers (Naïve Bayes, Maximum Entropy and Support Vector Machines). They applied ensemble approaches like fixed combination, weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy.

Sentiment analysis is the process of computationally identifying and categorizing opinions from piece of text, and determine whether the writer’s attitude towards a particular topic or the product, is positive, negative, or neutral. Finding the sentiments of a person (also called the polarization of a text) is a classical NLP problem. This problem brings a relatively new perspective by considering twitter tweets and Facebook posts, which are effectively a dialect of the English language. This problem has been considered already by a few authors. Since it is a necessary step but is not central to our questions, we plan to reuse some filters already available. If these detectors do not work as well as expected, we plan to use a robust binary classifier.

We will compare most popular algorithms such as Naïve Bayes, Decision Tree, Logistic Regression, KNN, Random Forest and SVC on sentiment analysis. We will use datasets to find the polarity as positive or negative.

In 2019, Saad and Yang have aimed for giving a complete tweet sentiment analysis on the basis of ordinal regression with machine learning algorithms. The suggested model included pre-processing tweets as first step and with the feature extraction model, an effective feature was generated [3].

# **CHAPTER 3**

## **PROJECT MANAGEMENT**

In order to design our system on **Sentiment Analysis,** first we will collect the related information. Then we will plan the work for the success of the concerning concept.

### **3.1 Team Members**

For this project, we have a group of four members:

1. Arun Prajapati (740305)
2. Neetu Phaiju (740324)
3. Rabin Phaiju (740329)
4. Rodip Duwal (740334)

### **3.2 Feasibility Study**

The aim of feasibility study is to understand thoroughly all aspects of a concept, or plan. During the study, problems in the system are determined. So, it’s always good to have a contingency plan that test to make sure it’s a viable alternative in case the first plan fails. This study will be reliable for forecasting the detail of the system.

### **3.3 Work break down structure**

All the team members will work on different modules. During the course of work, team members will be in touch so that no problems will arise in future. After the completion of individual work, whole work will be combined to develop a proposed system.

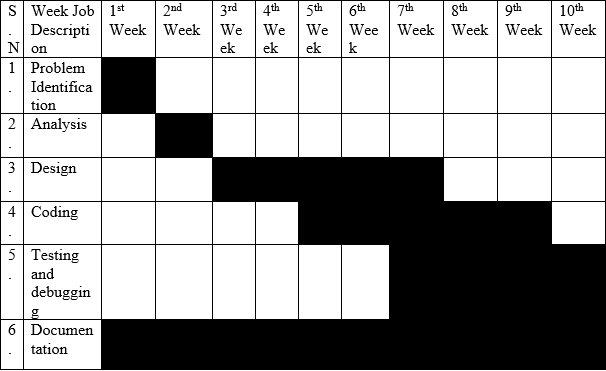


Fig 3.1: Work break down structure of our project

# **CHAPTER 4**

## **METHODOLOGY**

### **4.1 Background**

Sentiment Analysis task is considered a sentiment classification problem. The first step in the SC problem is to extract and select text features. This section presents survey of the related work performed on feature extraction in Sentiment Analysis. We have reviewed nearly 20 publications and categorically summarized their main techniques and contributions in different sections. Major feature extraction and manipulation steps and techniques, identified from cited publications are summarized in below sections.

The step involved for feature extraction are described as follows:

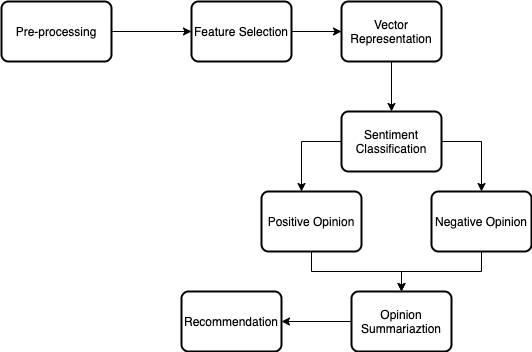


Fig 4.1: The proposed methodology for sentiment analysis.

**4.1.1.** **Preprocessing**: In Preprocessing the unanalyzed data is handled for feature extraction. In this phase, the several techniques like POS tagging, Stemming and Stop word removal are applied to data set for noise reduction and facilitating feature extraction [24].

It is further divided into below steps:

* **Tokenization**: White spaces, symbols and special characters are removed and a sentence is divided into words. Tokenization helps to filter out unnecessary tokens. For example, a document into paragraphs or sentences into words. For our paper we tokenized each sentence for making stemming and stop word removal easier.
* **Cleaning**: For this we created a cleaner class that:
  + Converts uppercase letters to lowercase
  + Normalization: Convert number to its equivalent word or completely removes numbers.
  + Removes punctuations, accent marks and whitespaces etc.
  + Removes contractions.
* **Stemming and Lemmatization:** Stemming and Lemmatization are two essential morphological processes of preprocessing module during feature extraction. It has been seen that most of the times the morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. Since the meaning is same but the word form is different it is necessary to identify each word form with its base form. To do this a variety of stemming algorithms have been developed [8].

In our paper we used Porter stemmer, is as of now one of the most popular stemming methods proposed in 1980. This reduces all words with the same root to a single form, the stem, by stripping the root of its derivational and inflectional affixes [27].

There are mainly total of 5 rules: The rule looks like the following:

<condition> <suffix> → <new suffix>

For example, a rule (m>0) EED → EE means “if the word has at least one vowel and consonant plus EED ending, change the ending to EE”. So “agreed” becomes “agree” while “feed” remains unchanged.[4]

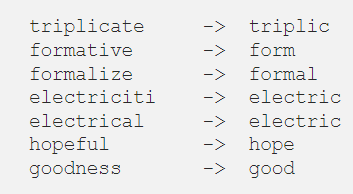


Fig 4.2: Porter Stemmer examples

* **Stop Word Removal:** Stop word concept was first introduced by Hans Luhn, H.P. Stop words are common and high frequency words like “a”, “the”, “of”, “and”, “an”. ]; ultimately enhance performance of feature extraction algorithm.

The stop words removal reduces dimensionality of the data sets and thus key words left in the review corpus can be identified more easily by the automatic feature extraction techniques. Words to be removed are taken from a commonly available list of stop words. Savoy had given huge collection stop word list. At simplest level stop words are iterated in chosen word list and removed from text. We made use of list that contains stop words and filter out the words for sentence that matches those stop words.

**4.1.2.**  **Feature extraction:** Feature extraction handles the following task:

* Feature Type: In this step features are identified like the term frequencies, term cooccurrences, Opinion word, OS information, Negation Syntactic Dependencies.
* Selection of Feature: Good features are selected for classification using the following ways like Information gain, Document frequency, Odd ratio and Mutual Information.
* Feature Weighting Mechanism: The features are ranked by computing the weight using term presence, term frequency and Inverse document frequencies.
* Reduction of Feature: To optimize the classifiers performance the vector size is reduced.

Two techniques are used for feature extraction -TF-IDF and Doc2vec. Figure 4.3 shows the steps involved in extracting features using both the techniques.

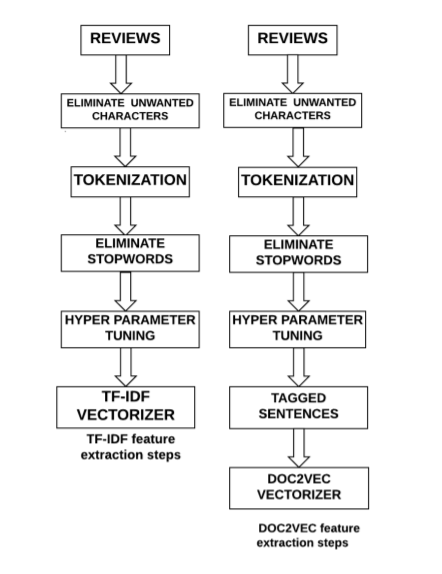


Fig. 4.3: Feature Extraction Procedure

But we only used TF-IDF for our project because the performance of Doc2vec is poor for short documents and according to study of performance comparison between these two models TF-IDF with SVM performed better than Doc2vec with SVM [12].

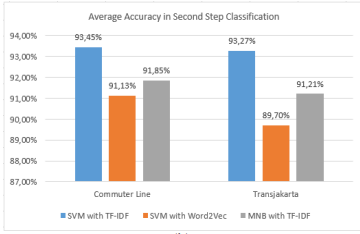


Fig. 4.4: Accuracy Comparison Between SVM with TF-IDF and SVM with Doc2Vec

**TF-IDF**

TF-IDF is short form for term frequency-inverse document frequency.TF-IDF is one of the largely used methods in information retrieval and text mining. TF-IDF is a weight metric which determines the importance of word for that document [11].

**TF**

Term Frequency measures number of times a particular term occurred in a document d. Frequency increases when the term has occurred multiple times. TF is calculated by taking ratio of frequency of term t in document d to number of terms in that particular document d.

**IDF**

TF measures only the frequency of a term t. Some terms like stop words occur multiple times but may not be useful. Hence Inverse Document Frequency (IDF) is used to measure term’s importance. IDF gives more importance to the rarely occurring terms in the document d. IDF is calculated as:

The final weight for a term t in a document d is calculated as:

**4.1.3.** **Sentiment Classification:** Sentiment classification uses two approaches to classify the nature of documents/sentence. They are Machine Learning Approaches and Lexicon Based Approaches.

* The Machine Learning belongs to supervised learning and classification of text in particular. So, it is called as ―Supervised Learning.
* Lexicon Based approach consist of Dictionary based and corpus based.

Classifiers are trained on training data sets with the features obtained from the feature extraction techniques TF-IDF and Doc2vec and with the corresponding output labels from the dataset. The test data is tested with the trained classifiers to predict the sentiments and the accuracy is measured for test data. For our purpose we used 6 popular classifiers, they are: Naïve Bayes, Decision tree, Logistic regression, KNN, Random Forest, SVC.

**4.1.3.1. Naïve Bayes**

Bayesian classifiers are based around the Bayes rule, a way of looking at conditional probabilities that allows you to flip the condition around in a convenient way. A conditional probably is a probably that event X will occur, given the evidence Y. That is normally written P(X | Y). The Bayes rule allows us to determine this probability when all we have is the probability of the opposite result and of the two components individually:

This restatement can be very helpful when we are trying to estimate the probability of something based on examples of it occurring.

In our case, we are trying to estimate the probability that a document is positive or negative, given its contents. We can estimate the probability of a word occurring, given a positive or negative sentiment by looking through a series of examples of positive and negative sentiments and counting how often it occurs in each class. This is what makes this supervised learning, the requirement for pre-classified examples to train on.

**4.1.3.2. Decision Tree**

This classifier worked on the basis of dividing the working area repetitively into multiple sub parts. Decision Tree can possibly produce important insight into interactions between variables. Because, there are two different regions of same class. Class is either divided on the basis of purity in classification or impurity in classification. Clear separation of work classes is done easily. But when there is impurity in classes. It is to be solved by the concept of Entropy. Entropy works on the basis degree of how particular element is occurred randomly [13]. Mathematically it is calculated using probability of element as:

P(x) is a probability of element x.

**4.1.3.3. Logistic Regression**

Logistic Regression belongs to the family of classifiers known as the exponential or log-linear classifiers. It works by extracting some set of weighted features from the input, taking logs, and combining them linearly (meaning that each feature is multiplied by a weight and then added up). Technically, logistic regression refers to a classifier that classifies an observation into one of two classes, and multinomial logistic regression is used when classifying into more than two classes [22]. Logistic Regression is a discriminative classifier. Recall that the job of a probabilistic classifier is to choose which output label y to assign an input x, choosing the y that maximizes P(y|x).

**4.1.3.4. KNN**

KNN uses feature similarity where it assigns a data point based on how close it is to its neighbor. The algorithm for the KNN shown below is used for the classification of the data.

Algorithm:

Step 1: Load dataset

Step 2: Select the value of k

Step 3: Calculate the distance between each data point using Euclidean distance

Step 4: Sort data point according to the distance calculated

Step 5: Select the top k row

Step 6: Assign data point on the most frequent class

Step 7: End

To calculate the distance of the data point in the KNN algorithm Euclidean distance is calculated.

**4.1.3.5. Random Forest**

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Random forest takes the prediction from each tree and based on the majority votes of predictions and it predicts the final output. The greater number of trees in the forest leads to higher accuracy.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

Algorithm:

Step 1: Select random k data points from the training set.

Step 2: Build the decision trees associated with the selected data points.

Step 3: Choose the number N for decision trees that we want to build.

Step 4: Repeat Step 1 & 2.

Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

**4.1.3.6. SVC**

The Linear Support Vector Classifier (SVC) method applies a linear kernel function to perform classification and it performs well with a large number of samples. Support Vector Classification (SVC) can be extended to solve regression problem. That extended method is called Support Vector Regression (SVR).

Linear SVC is based on liblinear and only supports a linear kernel. So SVC (kernel = ‘linear’) is in theory “equivalent” to: LinearSVC( ). The objective of a Linear SVC is to fit to the data provided returning a best fit hyperplane that divides, or categorizes, the data.

**4.1.4.** **Visualization**: An application software that takes text or voice input and shows the sentiment behind each sentence, Also the software will compare the different plots to show the frequency of words in the customer tweets and the sentiment scores.

### **4.2 Datasets**

We employ two datasets in this paper.

**4.2.1 Movie reviews**

This is a dataset for binary sentiment classification containing 40K movie reviews for natural language processing or Text analytics. This dataset consists of 20,000 highly polar movie reviews for training and 20,000 for testing. This dataset was collected by Stanford research [21].

**4.2.2 Conversation Dataset**

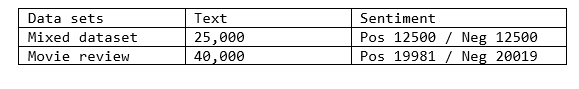
This dataset consists data of casual conversations to train model for sentiment analysis. This dataset consists of 25000 conversations.

Fig 4.5: Datasets used for our research.

### **4.3 Comparison Measure**

Let a represent the number of messages correctly classiﬁed as positive (i.e., true positive), b the number of negative messages classiﬁed as positive (i.e., false positive), c the number of positive messages classiﬁed as negative (i.e., false negative), and d the number of messages correctly classiﬁed as negative (i.e., true negative). In order to compare and evaluate the methods, we consider the following metrics, commonly used in information retrieval: true positive rate or recall: R=a/(a+c), false positive rate or precision: P=a/(a+b), accuracy: A= (a+d)/(a+b+c+d), and F-measure: F= 2 ·(P·R)/(P+R). We will in many cases simply use the F-measure, as it is a measure of a test’s accuracy and relies on both precision and recall. We report all the metrics listed above since they have direct interpretation in practice. The true positive rate or re-call can be understood as the rate at which positive messages are predicted to be positive (R), whereas the true negative rate is the rate at which negative messages are predicted to be negative. The accuracy represents the rate at which the method predicts results correctly (A). The precision rate, also called the positive predictive rate, calculates how close the measured values are to each other (P). We also use the F-measure to compare results, since it is a standard way of summarizing precision and recall (F). Ideally, a polarity identiﬁcation method reaches the maximum value of the F-measure, which is 1, meaning that its polarity classiﬁcation is perfect.

### **4.4 Flowchart**

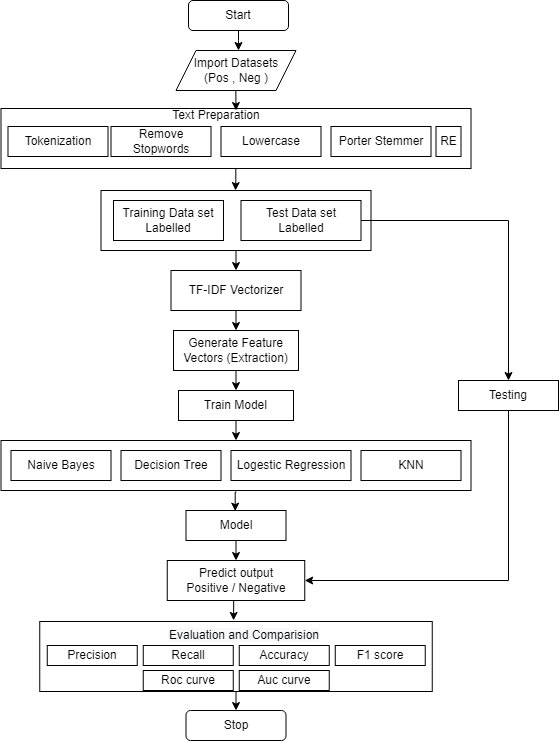


Fig 4.6: Flowchart for process of sentiment analysis

### **4.5 Tools and Platform**

1. Python
2. Google Colab
3. Jupyter Notebook
4. VS code
5. Hardware

# **CHAPTER 5**

## **RESULT AND DISCUSSION**

### **5.1 RESULT ANALYSIS**

In this section we analyzed different comparison measures for 6 popular algorithms of Sentiment Analysis. Fig 5.1 and Fig 5.2 shows the measure i.e., Accuracy, F1 Score, Recall and Precision of different algorithms for SA. The analysis of Fig.5.1 was performed using conversation dataset and Fig 5.2 using Movie Review Dataset. We trained our own model for this analysis. The figure shows that among all these 4 algorithms Logistic regression gave higher value for all comparison measures.

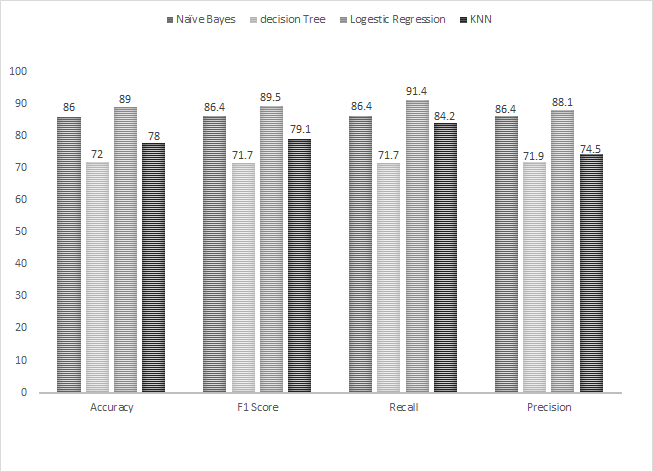


Fig. 5.1: Comparison of different performance measure of 4 different Algorithms for Conversation dataset.

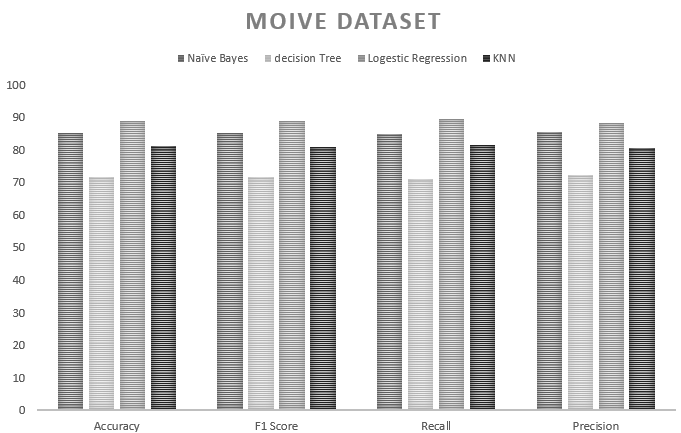


Fig. 5.2: Comparison of different performance measure of 4 different Algorithms for Movie Dataset.

Some people have this preconceived notion that logistic regression shouldn’t perform well because it is too simple. While that might be true for certain types of complex problems, but in our study, we only used datasets with binary classifier, true and false. So, that might be the reason as to why our model with logistic regression gave higher measures.

Another reason as to why Logistic regression algorithm compared better may be due to the fact that the Logistic Regression doesn’t make as many assumptions as of other algorithm such as naïve bayes.

But in the world of fine-grained Sentiment Analysis, Naïve Bayes and SVC are the widely used and popular ones. Our comparison study focused on detecting the polarity of content (i.e., positive and negative aﬀects) and does not yet consider other types of sentiments (e.g., psychological processes such as anger or calmness)). For this case these two algorithms might be the best option [9 13 18].

More important, we try to show that the prediction performance of method vary largely across datasets. It can be inferred from the results of this study that, using sufficient data set for predictive model construction can lead to better accuracy and the model’s ability to generalized. Although, due to vagueness that surrounds the size of the dataset, it is difficult to say precisely when a dataset can be considered to be big [17].

We also found that among these four no single methods is always best across diﬀerent text sources. As mentioned, for only detecting the polarity of a text, Logistic regression gave higher accuracy. Furthermore, for diverse categories of sentiment beyond positive and negative polarity logistic might not be the best option.

### **5.2 RESEARCH ANALYSIS**

The articles of sentiment analysis using naïve bayes presented in the survey are summarized in table 5.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| References | Year | Data Scope | Accuracy | Precision | Recall | F-score |
| [9] | 2016 | Twitter Feeds | 0.89 | 0.88 | 0.885 | 0.88 |
| [18] | 2013 | Facebook status | 0.78 | 0.77 | 0.68 | 0.72 |
| [31] | 2019 | Twitter Comments | 0.8343 | 0.5629 | 0.6513 | 0.5768 |
| [33] | 2020 | Amazon Review | 0.88 | 0.8679 | 0.9019 | 0.8846 |

Table 5.1: Article Summary for Sentiment analysis using Naïve Bayes

The articles of sentiment analysis using decision tree presented in the survey are summarized in table 5.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| References | Year | Data Scope | Accuracy | Precision | Recall | F-score |
| [13] | 2019 | Tweets | 0.63 |  |  |  |
| [23] | 2017 | Tweets | 0.8466 | 0.9596 |  |  |
| [31] | 2019 | Twitter comments | 0.8291 | 0.8291 | 1 | 0.9066 |

Table 5.2: Article Summary for Sentiment analysis using Decision Tree

The articles of sentiment analysis using logistic regression presented in the survey are summarized in table 5.3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| References | Year | Data Scope | Accuracy | Precision | Recall | F-score |
| [1] | 2017 | Food Review | 0.934 | 0.93 | 0.93 | 0.93 |
| [4] | 2020 | Yelp Review | 0.949 |  |  | 0.902 |
| [22] | 2021 | Food Review | 0.8235 | 0.8878 | 0.8764 |  |

Table 5.3: Article Summary for Sentiment analysis using Logistic Regression

The articles of sentiment analysis using KNN presented in the survey are summarized in table 5.4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| References | Year | Data Scope | Accuracy | Precision | Recall | F-score |
| [13] | 2019 | Tweets | 0.67 |  |  |  |
| [23] | 2017 | Tweeter data | 0.84 | 0.85 | 0.81 | 0.83 |
| [26] | 2017 | Movie Review | 0.5530 |  |  |  |
| [34] | 2021 | User Review | 0.5625 | 0.64 | 0.56 | 0.54 |

Table 5.4: Article Summary for Sentiment analysis using KNN

The articles of sentiment analysis using Random Forest presented in the survey are summarized in table 5.5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| References | Year | Data Scope | Accuracy | Precision | Recall | F-score |
| [9] | 2016 | Tweets | 0.8291 | 0.56 | 0.333 | 0.30 |
| [30] | 2020 | Social Media | 0.8263 | 0.70 | 0.785 |  |
| [32] | 2020 | Hindi Tweets | 0.902 |  |  | 0.6635 |

Table 5.5: Article Summary for Sentiment analysis using Random Forest

The articles of sentiment analysis using SVC presented in the survey are summarized in table 5.6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| References | Year | Data Scope | Accuracy | Precision | Recall | F-score |
| [30] | 2020 | Social Media | 0.798 | 0.508 | 0.866 |  |
| [33] | 2020 | Amazon Review | 0.91 | 0.9166 | 0.8979 | 0.9072 |
| [34] | 2021 | User Review | 0.875 | 0.88 | 0.88 | 0.88 |

Table 5.6: Article Summary for Sentiment analysis using SVC

The research data also proves our result analysis that the Logistic regression is better in term of accuracy, but again all these papers used different datasets and preprocessing techniques that hugely impact the accuracy but in general, Logistic Regression shows higher accuracy. The reason as to why might be similar to as explained above in this section.

### **5.3 THE SENTIMENT ANALYZER**

Finally, having compared the different sentiment methods and tested different performance parameter, we have created a python-based desktop application. This application takes text or voice (voice-to-text) as an input. A paragraph is divided as multiple sentences of which sentiment is calculated using 4 different algorithms as shown in Fig. 5.3. Also, the result tab will estimate the performance parameters of each algorithm used.

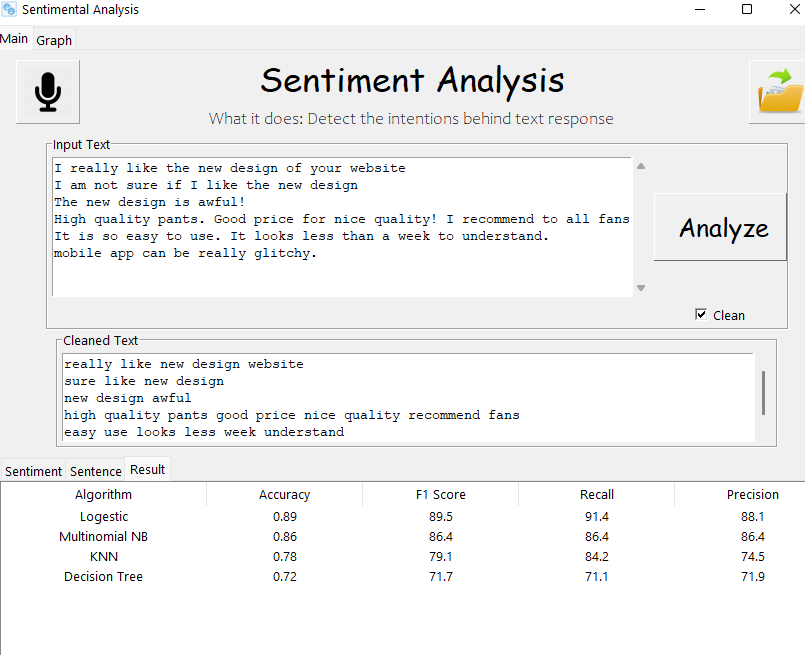


Fig 5.3: Screenshot of Sentiment Analysis Application

### **5.4 REMAINING WORK**

In this study we have to further go through implementation of SVC and Random Forest algorithm on sentiment analysis. And we will develop a REST API that will use our own models and using that API we will develop a web page that will clearly visualize different performance parameters of different algorithms.

## **REFERENCES**

**Website**

[1] Sentiment Analysis: A definitive Guide, Retrieved from:

Monkeylearn.com/sentiment-analysis/

[2] Sentiment Analysis: ScienceDirect, Retrived from:

Sciencedirect.com/topics/engineering/sentiment-analysis

[3] S. E. Saad and J. Yang, “Twitter Sentiment Analysis Based on Ordinal Regression”, 2019

[4] Dehao Xhang, Sentiment Classification with Logistic Regression – Analyzing Yelp Reviews, 2020

**Paper in Conference**

[5] A. Pak and P. Paroubek., Twitter as a corpus for Sentiment Analysis and Opinion Mining”, 2010

[6] Ajiboye Abdulraheem, Ruzaini Abdullah Arshah, Evaluating the Effect of Dataset Size on Predictive Model Using Supervised Learning Technique, 2015

[7] Agarwal, B. Xie, I. Voysha, O. Rambow, R. Passonneau, “Sentiment Analysis of Twitter Data”, 2011

[8] Anjali G. Jivani, The Maharaja Sayajirao University, A comparative Study of Stemming Algorithm, 2011

[9] Anmol Nayak, Dr. S Natarajan, Comparative study of Naïve Bayes, Support Vector Machine and Random Forest Classifiers in Sentiment Analysis of Twitter feeds, 2016

[10] Avinash M and Sivasankar E, A Study of Feature Extraction techniques for Sentiment Analysis, 2019

[11] D. Vilares, C. G´omez-Rodrıguez and M.A. Alonso, Universal, unsupervised (rule-Based), uncovered sentiment analysis, Knowledge-Based Systems, 2017

[12] Denis Ekta Cahyanil, Irene PAtasik Performance comparison of TF-IDF and Word2Vec models for emotion text classification, 2021

[13] Dimple Tiwari, NAnhay Singh, Ensemble Approach for Twitter Sentiment Analysis, 2019

[14] East West University, Sentiment Analysis on Twitter Data using KNN and SVM, 2017

[15] Federal University of Minas Gerals, Comparing and combining sentiment analysis methods, 2013

[16] Hafiz Muhammad Ahmed, Mazhar Javed Awam, Nabeel Sabir Khan, Hafiz Muhammad Faisal Shehzad, Sentiment Analysis of Online Food Reviews using Big Data Analytics, 2021

[17] Han-Xiao Shi, Xiao-Jun Li, A Sentiment analysis model for hotel reviews based on supervised learning, 2011

[18] Jaime Caro, Sentiment analysis of Facebook statuses using Naïve Bayes classifier for language learning, 2013

[19] L. Barbosa, J. Feng. “Robust Sentiment Detection on Twitter from Biased and Noisy Data”, 2013

[20] M. Taboada, J. Brooke, M.Tofiloski, K. Voll and M. Stede, Lexicon-based methods for sentiment analysis, 2011

[21] Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher, Learning Word Vectors for Sentiment Analysis, 2011

[22] Mayur Wankhade, A Chandra Sekhara Rao, Suresh Dara, Baijnath Kaushik, A Sentiment Analysis of Food Review using Logistic Regression, 2017

[23] Md Shoeb, Jaweb Ahmed, Sentiment Analysis and Classification of Tweets Using Data Mining, 2017

[24] Muhammad Zubair Asghar, Aurangzeb Khan, Shakeel Ahmad, Fazal Masud Kundi, A Review of Feature Extraction in Sentiment Analysis, 2014

[25] Pablo Gamallo, Marcos Garcia, “Citius: A Naïve-Bayes Strategy for Sentiment Analysis on English Tweets”, 2014

[26] Palak BAid, Apoorva Gupta, Neelam Chaplot, Sentiment Analysis of Movie Reviews using Machine Learning Techniques, 2017

[27] Peter Willett, The porter Stemmer algorithm, 2006

[28] Po-Wei Liang, Bi-Ru Dai, “Opinion Mining on Social Media Data”, 2013

[29] R. Xia, C.Zong, and S. Li, “Ensemble of feature sets and classification algorithms for sentiment classification”, 2011

[30] Shoffan Saifullah, Yuli Fauziah, Agus Sasmito Aribowo, “Comparison of Machine Learning for Sentiment Analysis in Detecting Anxiety based on social media data”, 2020

[31] Veny Amilia Fitri Rachmadita Andreswari, Muhammad Azani Hasibuan, Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm, 2019

[32] Arnav Munshi, Sanchit Sapra, M.Arvindhan, “A Novel Random Forest Implementation of Sentiment Analysis”, 2020

[33] Sandeep Singh Sikarwar, Dr. Nirupma Tiwari, “Analysis The Sentiments of Amazon Reviews Dataset By Using Linear SVC and Voting Classifier”, 2020

[34] Asad Khattak, Muhammad Zubair Asghar, Zain Ishaq, Waqas Haider Bangyal, Ibrahim A Hameed, “Enhanced concept-level sentiment analysis system with expanded ontological relations for efficient classification of user reviews”, 2021