# Sentimental Analysis

**Dolon Banik**
**Student Id: 011 142 029**

**Shifat Naznin**
**Student Id: 011 142 059**

**Md. Abdullah Al Emran**
**Student Id: 011 142 150**

**Durre Shahriar Srabony**
**Student Id: 011 131 037**

A thesis in the Department of Computer Science and Engineering presented
in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering


United International University
Dhaka, Bangladesh
April 2019

# Declaration

We, (Dolon Banik, Shifat Naznin, Md. Abdullah Al Emran, Durre Shahriar Srabony), declare that this thesis titled, Thesis Title and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a BSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

_____

Dolon Banik
011142029
Dept. of CSE
United International University
Dhaka-1212, Bangladesh

_____

Shifat Naznin
011142059
Dept. of CSE
United International University
Dhaka-1212, Bangladesh

_____

Md. Abdullah Al Emran
011142150
Dept. of CSE
United International University
Dhaka-1212, Bangladesh

_____

Durre Shahriar Srabony
011131037
Dept. of CSE
United International University
Dhaka-1212, Bangladesh

# Certificate

I do hereby declare that the research works embodied in this thesis entitled “Sentiment Analysis” is the outcome of an original work carried out by [Dolon Banik, Shifat Naznin, Md. Abdullah Al Emran, Durre Shahriar Srabony] under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of BSc in Computer Science and Engineering.

_____

Dr. Mohammad Nurul Huda
Professor and Director - MSCSE
Dept. of CSE
United International University
Dhaka-1212, Bangladesh

# Abstract

The web has opened the new entryways for information exchange. The development of internet-based life has expanded step by step that makes remarkable open doors for natives to freely raise their opinion. But it has genuine bottlenecks with regard to the investigation of these feelings. Sentiment analysis helps to improve the client experience over an interpersonal, organization or framework interface. The learning algorithm will take in what feelings are from measurable information at that point deciding the mood. Assume that if anyone is exhausted or sad in the case of social networks, one thing the computer could do that will be more increasingly suggestive of things that lighten mood and change connections like the background colors. At that point, the site automatically takes a stab at suggesting interactions with individuals and applications that assistance for improving the state of mind. The brief about sentiment analysis, including presentation of sentiment analysis, past history of sentiment analysis, Sentiment analysis using machine learning tools, mathematical tools. The project on sentiment analysis means to actualize those in the organizational network community as well as the services and interfaces of frameworks while making the background more extravagant and productive and lives better.

# Table of Contents

# Chapter 1

## Introduction

The process of mined valuable data from a vast arrangement of data is called Data mining. In this procedure, various analysis tools of data mining like clustering, classification, regression and so on can be utilized for sentiment analysis. Sentiment Analysis which is known as opinion/supposition/thought mining refers to the utilization of the natural language processing toolkit (NLTK). Sentiment mining is a standout amongst the most vital viewpoints for the information mining process where important/most vital data can be mined dependent on the positive or the negative senses of the gathered data. We discovered sentiment in criticism, feedback, comments, studies or critiques. The source materials of opinions, reviews, comments given in different person to person communication destinations. Those sentiment gives valuable demonstrators to a wide range of purposes and can be sorted by polarity. By the polarity can find out that a review/comment is by and large a positive one or a negative one. By the utilization of sentiment analysis, the environment is developing step by step industrially. Presently a-days the increasing number of brand following, showcasing and marketing organizations are offering this service. Services resemble evaluating market buzz, the action of a contender, customer feedback, style, and fashion, estimating the public response to an action or organization related issue and so on. In this paper for Sentiment Analysis, we are utilizing five supervised Machine Learning algorithms: (1) Naive Bayes, (2) Neural Network, (3) Support Vector Machine, (4) K-Nearest Neighbor, (5) Logistic Regression to calculate the accuracies. Likewise Clearness of the positive and negative corpus. Here we can scale the accuracy and efficiency of various algorithms. The challenges in Sentiment Analysis are that an opinion word which is treated as positive at times it might be considered as negative in another circumstance. For our thesis purposes gather corpus information we utilize a movie review dataset of more than a thousand words. At that point utilize this data to determine a client's state of mind. With the goal that we can feature whether they are in a positive state of mind or negative inclination.

# Chapter 2

## Background

In this report, the Background work for the sentiment analysis has been done the world over. We think about and discuss various paper's explorations and exchange subjects. In methodology, we discuss our arrangements for thesis progress and how the project is planned to move along. What are the objectives we have focused on the thesis and what results would we say we are searching for? In methodology and Sentiment analysis utilizing AI techniques, we examine the review on how the software was implemented. How we prepared our classifiers and in Experimental Result we talk about the results we acquired. Demonstrated the precision of the classifiers gets from the distinctive methods below. In Conclusion and Future Works we examine the overview and brief of the task what we will do in the future.

In this report, the Background work for the sentiment analysis has been done far and wide. We examine and talk about various paper's examination and exchange topics. Sentiment Analysis is a procedure for determining a bit of composing content like a movie review, product review, tweets and so on which is positive or negative. Sentiment Analysis utilized by advertisers to research, public opinion and popular assessments of their organizations and items and furthermore to break down consumer satisfaction. Organizations additionally use sentiment analysis to assemble basic input, gather critical feedback about issues in discharging items. Sentiment analysis always helps companies to see how they’re doing with their clients. It moreover offers them a vastly improved picture of any way they pull together against their competitors. Like if any organization has a 20% negative sentiment, is that bad?

It depends on the off chance that your competitors have an around five hundredth positive and 100% negative sentiment, whereas yours is two hundredth negative, that merits a great deal of discovery to get a handle on the drivers of those opinions. Realizing the opinions related to competitors helps organizations to evaluate their own performance and search for ways how to improve. What are the objectives we have focused on the thesis and what results would we say we are searching for? Sentiment analysis using

machine learning methods we examine the diagram on how the product was implemented and how we trained our classifiers. In Experimental Result we discuss the outcomes we acquired. Demonstrated the accuracy of the classifiers gets from different techniques.

Sentiment analysis has been slowly completing more and more accepted. We are able to seem that with the development of web based mostly business and computerized advances, feeling investigation is remodeling into the factor. Sentiment analysis is employed to get the creator's conduct towards something. A few instruments classify bits of composing as positive, negative, or neutral. Sentiment analysis and conclusion mining have numerous applications going from web based mostly business, advertising, to governmental problems and a few alternative research. This is the way organizations can notice emptor mentalities towards them, their things, administrations, or showcasing efforts on exchange gatherings, survey locales, Facebook, Twitter, and other alternative overtly accessible sources. Now-a-days, purchasers utilize web based mostly life to share each their positive and negative encounters with brands. Feeling examination apparatuses will determine the two notices passing on terribly positive bits of substance showing of an item, or an administration and negative notices, terrible surveys, or specialized issues clients expound on the web. The science behind sentiment analysis depends on calculations utilizing a common language getting ready to classify bits of composing as positive or negative. The algorithm is meant to acknowledge positive and negative words, for instance, "phenomenal", "lovely", "baffling", "awful", and so forth. Owing to language complication, sentiment analysis needs to look somewhere around two or three issues. Another enormous issue sentiment analysis algorithms face is named-substance acknowledgment. Words with the sett have diverse importance. In Conclusion and Future Works we discuss the overview and brief of the project what we will do in the future.

## Chapter 3

# Previous Study

Schukla, A., at [1] describes, a tool which makes a decision about the nature of content depending on annotations on scientific papers. Its procedure gathers the opinions of comments in two methodologies. It checks all the explanation creates the records and calculates all out assessment scores. Its concern proclaims in a connection between explanations that is mind boggling. The system needs a major inquiry learning base containing metadata.

Kasper, W. & Vela, M., at [2] proposed a "Web Based Opinion Mining system" for online user's audits. The paper presented an assessment framework for online client surveys what's more, remarks to help quality controls in hotel management system. It is able of identifying and recovering surveys on the web and manages German audits. Furthermore, the multi-point cases distinguished between their corpus. It is most shortcoming represent in not dealing with certain cases of multi-point fragments.

Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, Sweta Tiwari, at [3] discuss about several analysis tools of data mining where important data can be mined dependent on the positive or negative senses of the gathered information.

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, at [4] they consider the issue of arranging records not by point, yet by and large sentiment, determining whether a survey is positive or negative. Utilizing movie review as information, find that standard AI systems definitively outperformed human-delivered baselines. The AI strategies utilized don't execute too on opinion classification as on conventional theme based arrangement.

Zhang, L., Hua, K., Wang, H., and Qian, G., at [5] describes about A Machine learning algorithm researched the accuracy of the Naive Bayes algorithm. Moreover, the exploration made a judgment on the item quality, and status in the market is worthwhile.

Godbole, N., Srinivasaiah, M., and Skiena, S., at [6] this paper breaks down news sentiment and online journals. It's part earlier work with regard to their particular task

into two classifications. First techniques for consequently creating sentiment lexicon and the second one which identifies with frameworks that analyze sentiment for whole file.

Bhumika M. Jadav, Vimalkumar B. Vaghela, at [7] this paper describes about Sentiment Analysis utilizing Support Vector Machine algorithm to arrange these reviews dependent on its sentiment as either positive or negative class. SVM is utilized to group surveys where RBF kernel SVM is adjusted by its hyper parameters which are delicate edge consistent C, Gamma γ. Improved SVM gives great outcome than SVM and Naive Bayes.

Mohsen Farhadloo and Erik Rolland, at [8] this paper discuss about the issue of distinguishing individuals conclusions communicated in composed language is a moderately new and dynamic field of research. Start the chapter of a brief contextual introduction to the issue of sentiment analysis and supposition mining and expand our presentation with a portion of its applications in various spaces. The primary difficulties in sentiment analysis and supposition mining are talked about, and distinctive existing ways to deal with location these difficulties are clarified.

Meena Rambocas, Joao Gama, at [9] they utilized the keyword based way to deal with arrange sentiment. He took a shot at recognizing keywords basically descriptors which demonstrate the opinion. Such markers can be arranged physically or got from Wordnet.

Brody, S., & Elhadad, N., at [10] this paper basically gathered information from the online sites, the researchers moved toward the related assignment of recognizing a feeling extremity in audits by means of regulated learning approaches. Their baseline tasks to demonstrate that people may not generally have the best instinct for picking segregating words. While they experimented with a lot of various highlights of the past examine their basic spotlight was not on highlight building.

Munir Ahmad, Shabib Aftab, Syed Shah Muhammad, Sarfraz Ahmad, at [11] describe multiple instruments and systems are accessible today for programmed sentiment analysis for this client created information. The methodologies are utilized for this reason Lexicon based systems, Machine Learning based strategies and half and half procedures which consolidates vocabulary based and AI based methodology.

# Chapter 4

## Methodology

For thesis purpose, we tried a new approach that is just tagging positive and negative data. We have tried such emotions as happy, funny, sad, and angry. First of all, to track this data, a new corpus will be created by positive and negative sentences. Then in the corpus, we approached a binary method to detect these emotions. Under positive data, we can detect happy or funny sentences and under negative data, we detect sad and angry sentences.

Research methodology of this study consists of some following steps:

- Marking off research questions
- Selection of keywords for query string
- Identification of search space
- Outlining the selection criteria
- Quality assessment
- Data adjustment and extraction
- Experimental results

Used Tools:

- Python NLTK
- Movie review dataset as Corpus
- User Interface

For our task, we utilized the Naive Bayes method, Support Vector Machine method, Neural Network method, Nearest Neighbor method, Logistic Regression method. For those, we used python NLTK, which has an opulent library to program. At first, this may not achieve much accuracy, but by incorporating the above-suggested methods it achieves better accuracy. By taking data sets of positive or negative data will bias the data to ensure proper data set incorporation, we must use both in our data set training.

The important things of machine learning are that machine learning algorithms can memorize the data. When we want to use it with new data it has a poor performance. The poor performance behavior is called over-fit. To avoid this problem we worked with test-driven methodology. There each data set is divided into two parts. We collected corpus data from a website. From the data, we used 70 %data for training and 30 %for testing data.

• **Training Dataset (70%):** From the data set used to feed the machine learning algorithm for the learning process.

• **Testing Dataset (30%):** To evaluate the performance of the algorithm.

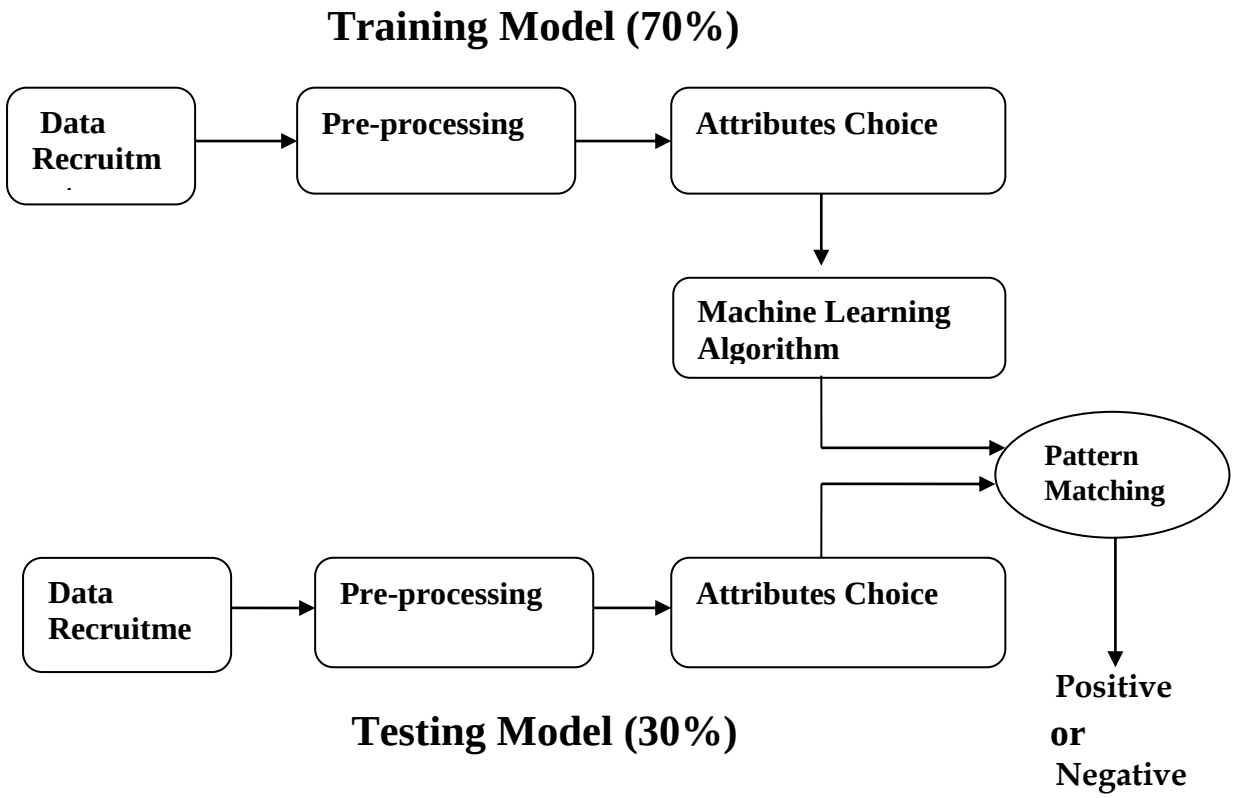



**Figure 4.1: System Diagram**

## Research methodology for sentiment analysis

- **Text pre-processing**
- **Feature generation**
- **Initial classification**
- **Final classification**

**Text pre-processing** first to initiation of a text classification task with the proper content pre-processing which implies that can sift through a point of view of the content that is probably not going to give our model any extra data. It additionally permits to streamline the content to give your model more data. A few models include:

1. Replacing all positive emotions with 1 in a document (csv). For instance, the sentence “This is good news for us – 1 and replacing all negative emotions with 0. For example “This is sad news – 0”.

2. Lower case letters.

3. Removing stop-words, for example, "to", "and", "do". These happen in all respects as often as possible and can mess the model, making it perform more terrible.

**Feature generation** after pre-processed the content and it is prepared to be put into a model. At that point which is expected to produce includes that will be put into the model. Which implies that it converts the content of each report into any numerical format that can be deciphered by the model. There are various prevalent features for Natural Language Processing (NLP) tasks:

1. **Word embedding’s** these are the representations of words, joining word comparability. It can discover pre-trained word set that can utilize to make highlight vectors for records.

2. **Dictionary/Lexicons** Dictionary is the words that appear most frequently in a given class. Dictionary can use something like NLTK to separate the most

regularly happening words in positive content, and the most commonly occurring words in negative content. Then you can use as a feature, the count of the number of positive/negative lexicons in each record.

**Initial classification** when setup is finished of highlight exhibit, then can feed it into models to make predictions. First, need to choose how that's going to utilize data to do this. Here are a few alternatives:

1. **Training:** Divide information up into two sections. We used 70% of the data for training the model, and 30% for testing. The outcomes get on testing is the last outcomes.

2. **Cross-validation:** This includes iteratively making different train/test parts over the data; end up with various diverse scores from testing. At that point need to average the score. This has the benefit of catching the outcomes over every one of the information without over-fitting. Each time a train/test is known as a fold.

3. **Stratified cross-approval**: Equivalent to the above mentioned, aside from it guarantees that each overlay has a similar distribution of classes. Like, on the off chance that 60% of information is classed as positive and 40% negative, at that point each fold will have 60% positive records and 40% negative.

Presently to run all features through a classifier, utilizing decision of train/test rules. There are numerous classifiers to choose from, and it is great to attempt to demonstrate with a group of them. Here are some good examples of the classifier:

i. Naive Bayes

ii. Neural Networks

iii. Support Vector Machine

iv. Nearest Neighbor

v. Logistic Regression

All above classifier and trying them out with different sets of features, from those one model with the best score.

**Final classification** presently is a keenly intellective thought to re-implement last tuned model to get the last score for a classification system to get the best score.

# Chapter 5

## Sentiment Analysis Using Machine Learning Tools

We used five different machine learning methods for sentiment analysis problem, where one of them is based on neural networks explained below.

### 5.1 Naive Bayes

Naive Bayes is a basic technique which depends on Bayes rule. According to Naive Bayes Classifier initially given by Thomas Bayes that is anything but difficult to actualize. The fundamental thought of the Naive Bayes method is to discover the probabilities of classes assigned to writings by utilizing the joint probabilities of words and classes. The likelihood of each element contributes independently to the final probability to be a class; each one has its circulation. The method performs progressively and more efficiently compared to other machine learning algorithms. The best effective and efficient learning algorithm for AI and data mining is Naive Bayes. Naive Bayes Classifier is a supervised classifier as it used to calculate the probability of a data to be positive or negative. The hypothesis is with an assumption of independence among predictors. In the present reality, its application is competing for performance in a classification that is shockingly once in a while evident. Naive Bayes Model is exceptionally helpful for huge datasets. We consider the positive class and negative clauses to evaluate the accuracy by preparing the Naive Bayes Algorithm using 1000 sentences and got 0.8499313186813187(85%) precision. The significant focal point of Naive Bayes classification methods for sentiment analysis is that anything but difficult to clarify and the outcomes are determined efficiently. While having the assumptions of attributes being independent is a drawback of this as it probably won't be legitimate constantly. Form the thought of general terms in Naive Bayes classifier the setting of sentiment classification. For a document **s** and class **r.**

The equation is

$$\mathrm{P(r \mid s)} = \frac{P(s \mid r)P(r)}{P(s)}$$

- The probability of the target given attribute / Posterior probability = P(c | s)
- Prior probability of class = P(r)
- The probability of predictor class = P(s | r)
- Predictor prior probability = P(r)

### 5.2 Neural Networks

The neural network is a machine learning technique which models itself after the human brain. The artificial neural network algorithm enables the computer to learn by arranging new information. Neural Networks might be a methodology that endeavors to upgrade a few loads, the body of the vegetative cell that is expanded by the vector of choices. Multilayer Neural Networks (MLP) can be utilized accordingly or as a component to the following arrangement of neurons. The forecast is the consequence of this duplication made by this neuron, the axon terminal.

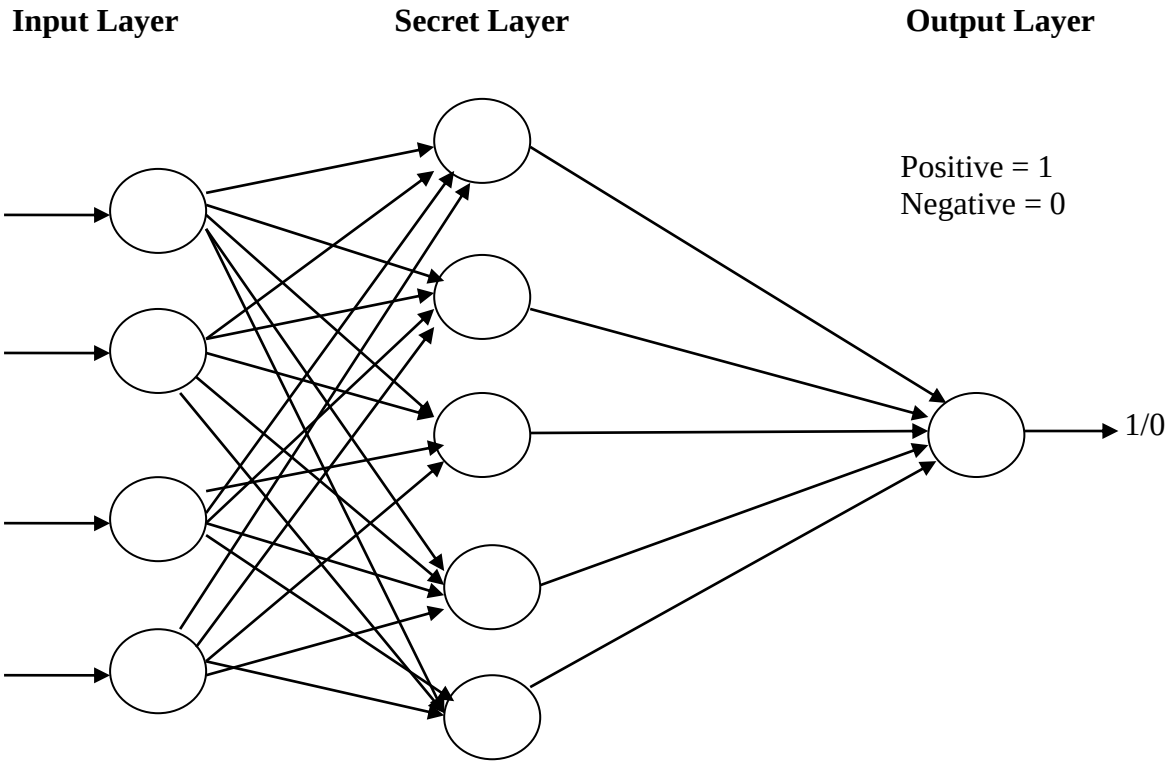



**Figure 5.1: Neural network**

The content classification issue can be seen as a transient appropriation of highlights. The goal is to train by the interior loads utilizing the Gradient Descent and Back-engendering technique. There is a cost, capacity is processed and the outcome is stretching out back to the neurons loads that get refreshed to limit the target work on each round. We utilization of Recurrent Neural Networks (RNN). It is a unique instance of Neural Networks which utilizes an interior memory on every neuron and speaks to the middle of the road understanding between highlights that can be amassed or overlook by the neuron. We have utilized our execution utilizing Tensorflow for RNN.

### 5.3 Support Vector Machines

Support Vector Machines (SVM) could be an administered learning model. With a related learning algorithm, it breaks down information utilized in order and relapse examination. SVM works by mapping data with a high-dimensional component space so information focuses can be arranged, regardless of whether the information is not directly distinct. Support vector machines think about that each arrangement of highlights speaks to a situation inside a hyperspace, and then the SVM endeavors to divide it using a hyper plane maximizing the distance between this hyper plane and each vector. At that point, it limits the goal work. This space division is once in a while unimaginable and difficult to achieve. For this, the SVM can utilize an edge that permits to misclassify and expands the general execution. Sentiment analysis is treated as a classification task since it groups the introduction of content into either positive or negative. The experimental results which are connected Support Vector Machine (SVM) on informational indexes to prepare a sentiment classifier. To extract a different weighting plan was used for the most classical features. By the exploratory examination uncovers that by exploitation Chi-Square element, the decision gives a noteworthy enhancement for characterization precision.

### 5.4 K Nearest Neighbors

The k-nearest neighbor algorithm (k-NN) is a non-parametric technique which is utilized for classification and regression. KNN formula is utilized to arrange by finding the K nearest matches in coaching information. For text classification, we used the Natural Language Toolkit (NLTK) library to create equivalent words and use closeness scores

among writings. We identify the K nearest neighbors which have the most elevated comparability score among the training data set.

The input consists of the k nearest coaching examples inside the component space. If k = 1, at that point the article is simply distributed to the classification of that single Nearest Neighbor. At that point, it utilizes the name of the nearest matches to foresee. Generally, separate, for example, geometrician is utilized to look out the nearest match.

In K-NN the separations between the unknown sample and training set can be figured. Let K =1. The distance with the smallest value assembles to the data in the training dataset closest to the unknown data. The unknown sample ordered depends on the arrangement of this nearest neighbor. K-NN is a simple algorithm to understand and actualize. It's a powerful tool for sentiment analysis. KNN is powerful because it does not assume anything about the data, as opposed to its measure remove that can be determined reliably between two examples. Exactness, Precision, and review are the strategies that utilized for assessing the execution of opinion mining.

$$\text{Accuracy} = \frac{p+s}{p+q+r+s}$$

$$\text{Recall (Positive)} = \frac{p}{p+r} \qquad \text{Precision (Positive)} = \frac{p}{p+q}$$

$$\text{Recall (Negative)} = \frac{s}{q+s} \qquad \text{Precision (Negative)} = \frac{s}{r+s}$$

- Accuracy = The overall accuracy of certain sentiment models
- Recall (Positive) = The ratio true positive reviews
- Precision (Positive) = Precision ratio for true positive reviews
- Recall (Negative) = The ratio true negative reviews
- Precision (Negative) = Precision ratio for true negative reviews

### 5.5 Logistic Regression

Logistic Regression refers to a language processing as most extreme entropy displaying which has a place with the group of classifiers known as the log-linear or exponential classifiers. Logistic Regression is mostly liked naive Bayes. It works by log-linear classifier extracting some arrangement of weighted highlights of the information input, taking logs, and joining them straightly. The classifier classifies/arranges by a perception into one of two classes, and multinomial logistic regression is accustomed to characterizing into more than two classes.

The most important difference between naive Bayes and logistic regression is that naive Bayes is a generative classifier while logistic regression is a discriminative classifier. A discriminative model adopts this immediate strategy, figuring by separating among the different conceivable qualities. While calculating relapse variations in such manner it estimates probabilities. It is as yet like Naive Bayes in being a linear classifier. Logistic regression estimates by extricating some arrangement of highlights from the information, input, joining them straightly. Linearly implies duplicating each element by weight and including them up. At that point applying a function to this combination. The produced qualities from minus infinity to infinity, powers the yield to be a lawful likelihood which lies somewhere in the range between 0 and 1. Actually, since the loads are genuine esteemed, may even be negative.

# Chapter 6

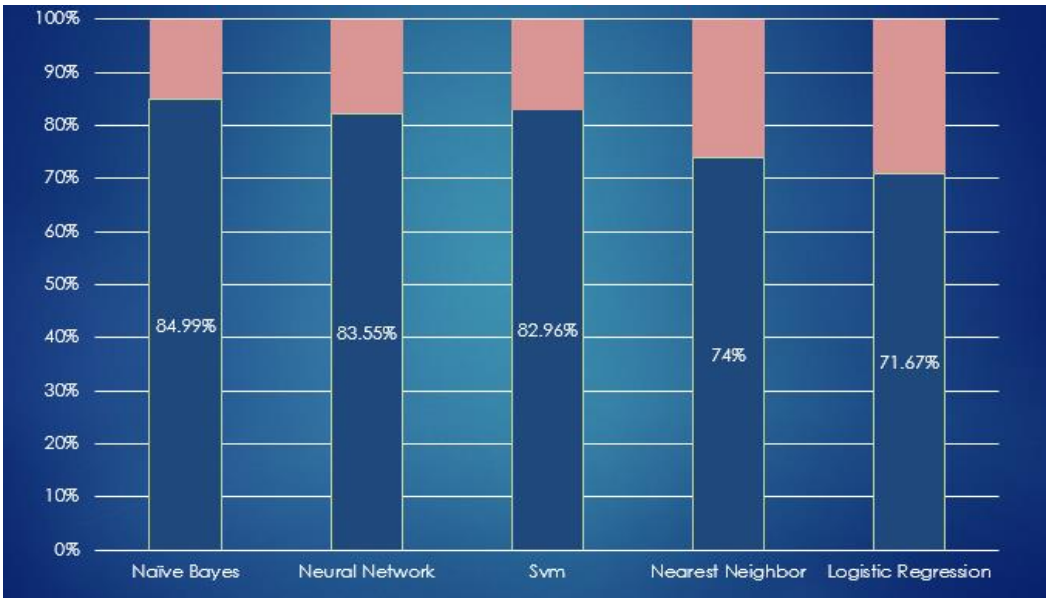
## Experimental Results

We utilized five machine learning tools (Naive Bayes technique, Support Vector Machine technique, Neural Network technique, Nearest Neighbor technique, Logistic Regression technique). By utilizing python NLTK, which has an excessive library to program. Taking informational collections of positive or negative information will predisposition the information to guarantee an appropriate informational index joining, we should utilize both in our informational index preparing. We have attempted such feelings as cheerful, entertaining, funny, angry, tragic, and sad. Above all else, to follow this information, another corpus will be made by positive and negative sentences. At that point in the corpus, we moved toward a paired technique to recognize these feelings. Under positive information, we can identify cheerful or amusing sentences and under negative information, we recognize dismal and irate sentences. The significant things of machine learning are that machine learning algorithms can remember the information. When we need to utilize it with new information, it has a poor act. To keep away from this issue we worked with test-driven system. There every datum set is isolated into two sections. We gathered corpus information from a site. From the information, we utilized 70% information for preparing and 30% for testing information. Then we applied five machine learning tools (Naive Bayes, Support Vector Machine, Neural Network, Nearest Neighbor, Logistic Regression) on the data-set and got some experimental results. Those results show different accuracy of positive and negative sentiment.

The training datasets accuracy's are shown below for the different classifications

- Naive Bayes: Accuracy 84.99%
- Neural Network: Accuracy 83.55 %
- Support Vector Machine: Accuracy 2.96%

- Nearest Neighbor 74%
- Logistic Regression 70.67%




The accuracy of all classifiers for over 1000 sentences data sets. The data sets are mainly based on movie reviews. It showed most 84.99% accuracy where the others showed 83.55%, 82.96, 74% and 70.67% of accuracy respectively. So the most accuracy rate is 84.99% got from the Naive Bayes Classifier.

# Chapter 7

## Conclusion and Future Works

### 7.1 Conclusion

In this chapter, the research study is studied for the analysis of opinion. There are detailed and clear suggestions about findings opinions. The research studies major contribution is to distinguish between the opinions is positive or negative on the basis of explicit opinions. The planning for the research studies center of attention will be on recognizing and by the help of the implicit and explicit features for identifying explicit as well as implicit opinions. So, by the using sentiment analysis techniques, it can be used to fetch the useful information as analyzing the market reputation of particular brands, getting user's feedback about any software, obtaining public opinion before launching a new product and so on. The field of sentiment analysis is an energetic new examination course on account of a sizable measure of real-world applications wherever finding individuals' sentiment is essential in higher decision making. The development of techniques for the archive level sentiment analysis, which is one with all the various pieces of this space. A great deal of investigation is a blessing in writing for many other works gathering opinion from the content. All things considered, there's a substantial extent of the progress of those current sentiment analysis models. Existing conclusion investigation models will be improved more with progressively extra etymology and commonsensical information.

### 7.2 Future Works

The work being done on the subject is endless tight. Here we just location the issue of client sentiment. In the future incorporating with this, the following stage will be in accomplishing better results and also better incorporation with social networking sites. Will attempt to create other facilities and android gadgets which can assist our program to achieve a more far-reaching experience. Utilizing a greater data set to improve accuracy (considering emoticons, sarcasm, etc.). Improving algorithms to deal with multi-language area. So for the future work, it is suggested to perform an altered strategy of the customized techniques with a similar informational collection.

## References

[1] Schukla, A., “Sentiment analysis of document based on annotation”, CORR Journal, Vol. abs/1111.1648, 2011.

[2] Kasper, W. & Vela, M., “Sentiment analysis for hotel reviews”, proceedings of the computational linguistics-applications, Jacharanka Conference, 2011.

[3] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, Sweta Tiwari, “Sentiment Analysis of Review Datasets using Naive Bayes’ and K-NN Classifier”, Information Engineering and Electronic Business, 2016.

[4] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques”, EMNLP, 2002**.**

[5] Zhang, L., Hua, K., Wang, H., and Qian, G., “Sentiments reviews for mobile devices products”, The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014) , procedia computer science, Volume 34, 2014.

[6] Godbole, N., Srinivasaiah, M., and Skiena, S., “Large-Scale Sentiment Analysis for News and Blogs”, ICWSM’2007 Boulder, Colorado, USA, 2007.

[7] Bhumika M. Jadav, Vimalkumar B. Vaghela, “Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis”, International Journal of Computer Applications, July 2016.

[8] Mohsen Farhadloo and Erik Rolland, “Fundamental of Sentiment Analysis and its Application”, Sentiment Analysis and Ontology Engineering, Springer International Publishing Switzerland, 2016

[9] Meena Rambocas, Joao Gama, “Marketing Research: The Role of Sentiment Analysis”, ISSN: 0870-8541, April 2013.

[10] Brody, S., & Elhadad, N., “An unsupervised aspect-sentiment model for online reviews”, Los Angeles, California, Association for Computational Linguistics: 804-812, 2010.

[11] Munir Ahmad, Shabib Aftab, Syed Shah Muhammad, Sarfraz Ahmad, “Machine Learning Techniques for Sentiment Analysis: A Review”, INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING, APRIL 2017.

[12] Data Preprocessing using python (https://towardsdatascience.com/data-preprocessing-in-python-6f04e6c2cb70)

[13] Download NLTK 2.0 (http://www.nltk.org/download)

[14] Python 3.6.7 (https://www.python.org/downloads/release/python-367/)

# Appendix

## Codes

## Naive Bayes

```
import pandas as pd
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn import naive_bayes
from sklearn.metrics import roc_auc_score
import numpy as np
# reading dataset
df = pd.read_csv("imdb_labelled.txt",sep='\t',names=['sentence','result'])
# preparing vectorization method
stopset = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(use_idf=True, lowercase=True,strip_accents='ascii',

# taking class values in y variable
y = df.result
# vectorizing feature data(Converting sentences into vectors) and taking them in
X = vectorizer.fit_transform(df.sentence)
# suppose, 70% data train variable e nibe, 30% data test variable e nibe
# train data dia algorithm ta train korbe, test data dia algorithm chaliye accur
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

# preparing naive bayes algorithm
clf = naive_bayes.MultinomialNB()

# Trainging train dataset with naive bayes algortithm
clf.fit(X_train,y_train)

# obtaining accuracy for naive bayes algorithm
print('Accuracy: ',roc_auc_score(y_test,clf.predict_proba(X_test)[:,1]))
a = np.array([input('User Input: ')])
a_vector = vectorizer.transform(a)

# if prediction of user input is 1 the result positive, if 0 then result negetiv
if(clf.predict(a_vector)[0]==1):
 print('Positive')
else:
 print('Negetive')
```

## Neural Network

```
import pandas as pd
from keras.preprocessing.text import Tokenizer
from sklearn.model_selection import train_test_split
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import RandomizedSearchCV
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras import layers
epochs = 20
embedding_dim = 50
maxlen = 100
output_file = 'data/output.txt'

filepath_dict = {'yelp':   'data/sentiment_analysis/yelp_labelled.txt','amazon':
                 'imdb':   'data/sentiment_analysis/imdb_labelled.txt'}
df_list = []
for source, filepath in filepath_dict.items():
    df = pd.read_csv(filepath, names=['sentence', 'label'], sep='\t')
    df['source'] = source
    df_list.append(df)
df = pd.concat(df_list)
def create_model(num_filters, kernel_size, vocab_size, embedding_dim, maxlen):
    model = Sequential()
    model.add(layers.Embedding(vocab_size, embedding_dim, input_length=maxlen))
    model.add(layers.Conv1D(num_filters, kernel_size, activation='relu'))
    model.add(layers.GlobalMaxPooling1D())
    model.add(layers.Dense(10, activation='relu'))
    model.add(layers.Dense(1, activation='sigmoid'))
    model.compile(optimizer='adam',
                  loss='binary_crossentropy',
                  metrics=['accuracy'])
    return model
    # Run grid search for each source (yelp, amazon, imdb)
    for source, frame in df.groupby('source'):
    print('Running grid search for data set :', source)
    sentences = df['sentence'].values
    y = df['label'].values
    # Train-test split
    sentences_train, sentences_test, y_train, y_test = train_test_split(
        sentences, y, test_size=0.25, random_state=1000)
```

```
    # Tokenize words
    tokenizer = Tokenizer(num_words=5000)
    tokenizer.fit_on_texts(sentences_train)
    X_train = tokenizer.texts_to_sequences(sentences_train)
    X_test = tokenizer.texts_to_sequences(sentences_test)

    # Adding 1 because of reserved 0 index
    vocab_size = len(tokenizer.word_index) + 1

    # Pad sequences with zeros
    X_train = pad_sequences(X_train, padding='post', maxlen=maxlen)
    X_test = pad_sequences(X_test, padding='post', maxlen=maxlen)

    # Parameter grid for grid search
    param_grid = dict(num_filters=[32, 64, 128],
                      kernel_size=[3, 5, 7],
                      vocab_size=[vocab_size],
                      embedding_dim=[embedding_dim],
                      maxlen=[maxlen])
    model = KerasClassifier(build_fn=create_model,
                            epochs=epochs, batch_size=10,
                            verbose=False)
    grid = RandomizedSearchCV(estimator=model, param_distributions=param_grid,
                              cv=4, verbose=1, n_iter=5)
    grid_result = grid.fit(X_train, y_train)
    # Evaluate testing set
    test_accuracy = grid.score(X_test, y_test)
#    prompt = input('finished {source}; write to file and proceed? [y/n]')
    with open(output_file, 'a') as f:
        s = ('Running {} data set\nBest Accuracy : '
             '{:.4f}\n{}\nTest Accuracy : {:.4f}\n\n')
        output_string = s.format(
            source,
            grid_result.best_score_,
            grid_result.best_params_,
            test_accuracy)
        print(output_string)
        f.write(output_string)
```

## Support Vector Machine

```
import numpy as np
import pandas as pd
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
from nltk.tokenize import TweetTokenizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split, StratifiedKFold, cross_va
from sklearn.pipeline import make_pipeline, Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer, accuracy_score, f1_score
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import confusion_matrix, roc_auc_score, recall_score, prec

df = pd.read_csv("imdb_labelled.txt",sep='\t',names=['sentence','result'])
data = list(df.sentence)
data_labels = list(df.result)
X_train, X_test, y_train, y_test  = train_test_split(data, data_labels,train_si

def tokenize(text):
    tknzr = TweetTokenizer()
    return tknzr.tokenize(text)
def stem(doc):
    return (stemmer.stem(w) for w in analyzer(doc))
en_stopwords = set(stopwords.words("english"))
vectorizer = CountVectorizer(analyzer = 'word', tokenizer = tokenize,
    lowercase = True, ngram_range=(1, 1),
    stop_words = en_stopwords)
kfolds = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)
np.random.seed(1)
pipeline_svm = make_pipeline(vectorizer,
                             SVC(probability=True, kernel="linear", class_weight
grid_svm = GridSearchCV(pipeline_svm,
                    param_grid = {'svc__C': [0.01, 0.1, 1]},cv = kfolds,
                    scoring="roc_auc",verbose=1,n_jobs=-1)
```

```
def report_results(model, X, y):
    pred_proba = model.predict_proba(X)[:, 1]
    pred = model.predict(X)
    auc = roc_auc_score(y, pred_proba)
    acc = accuracy_score(y, pred)
    f1 = f1_score(y, pred)
    prec = precision_score(y, pred)
    rec = recall_score(y, pred)
    result = {'auc': auc, 'f1': f1, 'acc': acc, 'precision': prec, 'recall': re
    return result
grid_svm.fit(X_train, y_train)
print('Accuracy',report_results(grid_svm.best_estimator_, X_test, y_test)['acc']
while True:
    s = input('Input: ')
    if grid_svm.predict([s])==1:
        print('Positive')
    else:
        print('Negetive')
```

## Nearest Neighbor

```
import pandas as pd
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier

df = pd.read_csv("imdb_labelled.txt",sep='\t',names=['sentence','result'])
y = df.result
X = df.sentence

#Using CountVectorizer to convert text into tokens/features
vect = CountVectorizer(stop_words='english', ngram_range = (1,1), max_df = .80,
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size= 0.2,random_st
#Using training data to transform text into counts of features for each message
vect.fit(X_train)
X_train_dtm = vect.transform(X_train)
X_test_dtm = vect.transform(X_test)


#Accuracy using KNN Model
KNN = KNeighborsClassifier(n_neighbors = 3)
KNN.fit(X_train_dtm, y_train)
y_pred = KNN.predict(X_test_dtm)
accuracy = 'Accuracy : {:.2f} %'
print(accuracy.format(metrics.accuracy_score(y_test,y_pred)*100))


while True:
    test = []
    test.append(input('Input: '))
    test_dtm = vect.transform(test)
    predLabel = KNN.predict(test_dtm)
    tags = ['Negative','Positive']
    #Display Output
    print(tags[predLabel[0]])
```

## Logistic Regression

File Edit Format Run Options Window Help

```
import pandas as pd
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression

df = pd.read_csv("imdb_labelled.txt",sep='\t',names=['sentence','result'])
y = df.result
X = df.sentence

X_train, X_test, y_train, y_test = train_test_split(X,y,random_state=1, test_siz

vect = CountVectorizer(stop_words='english', ngram_range = (1,1), max_df = .80,
vect.fit(X)
X_dtm = vect.transform(X)


LR = LogisticRegression()
LR.fit(vect.transform(X_train), y_train)
y_pred = LR.predict(vect.transform(X_test))
print('\nLogistic Regression')
print('Accuracy Score: ',metrics.accuracy_score(y_test,y_pred)*100,'%',sep='')


LR_complete = LogisticRegression()
LR_complete.fit(X_dtm, y)

while True:
    test = []
    test.append(input('Input: '))
    test_dtm = vect.transform(test)
    predLabel = LR_complete.predict(test_dtm)
    tags = ['Negative','Positive']
    #Display Output
    print(tags[predLabel[0]])
```