A PROJECT REPORT ON

SENTIMENTAL ANALYSIS

SUBMITTED IN PARTIAL FULFILLMENT FOR AWARD DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

ADITYA KUMAR & HRISHABH RAJ

(1906910 / 1906943)

(304 / 338)

UNDER THE GUIDANCE OF

ER. AMANJOT KAUR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MALOUT INSTITUTE OF MANAGEMENT & INFORMATION TECHNOLOGY, MALOUT

Jan-June 2022

DECLARATION

I hereby declare that the project entitled **"SENTIMENTAL ANALYSIS"** submitted for the B.Tech. CSE 6th is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Aditya Kumar (Signature)

Hrishabh Raj (Signature)

Place: MIMIT MALOUT

Date: _____

CERTIFICATE

We hereby certify that the work which is being presented in the Project-I report entitled "**SENTIMENTAL ANALYSIS**" by us in partial fulfilment of requirements for the award of degree of B.Tech. (CSE) submitted in the Department of Computer Science & Engineering at **MIMIT Malout** under **IKG PUNJAB TECHNICAL UNIVERSITY, KAPURTHALA** is an authentic record of my/our own work carried out during a period from "15TH FEB 2022" to "13TH JUNE 2022". The matter presented in this Project-I report has not been submitted by us in any other University / Institute for the award of any Degree or Diploma.

Signature of the Student/s

ADITYA KUMAR (1906910 / 304)

HRISHABH RAJ (1906943 / 338)

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

Signature of the Project-I Guide

Er. AMANJOT KAUR

Assistant Professor (CSE)

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. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behaviours. Whenever we need to make a decision, we want to hear others' opinions. Second, it presents many challenging research problems, which had never been attempted before the year 2000. Part of the reason for the lack of study before was that there was little opinionated text in digital forms. It is thus no surprise that the inception and the rapid growth of the field coincide with those of the social media on the Web.

In fact, the research has also spread outside of computer science to management sciences and social sciences due to its importance to business and society as a whole. In this talk, I will start with the discussion of the mainstream sentiment analysis research and then move on to describe some recent work on modelling comments, discussions, and debates, which represents another kind of analysis of sentiments and opinion.

ACKNOWLEDGEMENT

The authors are highly grateful to the Dr. JASKARAN SINGH BHULLAR Director, MIMIT Malout, for providing this opportunity to carry out the present Project-I work.

The constant guidance and encouragement received from**, Dr. SONIA SHARMA**, **HEAD CSE DEPARTMENT, MIMIT MALOUT**, has been of great help in carrying out the present work and is acknowledged with reverential thanks.

The authors would like to express a deep sense of gratitude and thanks profusely to **Er. AMANJOT KAUR**, who was our Project-I guides. Without the wise counsel and able guidance, it would have been impossible to complete the project in this manner.

The author express gratitude to other faculty members of Computer Science & Engineering Department, MIMIT Malout for their intellectual support throughout the course of this work.

Finally, the authors are indebted to all whosoever have contributed in this Project-I work.

Name of the Student/s

Aditya Kumar & Hrishabh Raj

TABLE OF CONTENS

LIST OF TABLES

LIST OF FIGURES

CHAPTER - 1: INTRODUCTION

1.1 General Introduction:

Sentiment Analysis of Social media Posts using Machine Learning Approaches to detect Sentiments like Depression and all.

Sentiment analysis is the task of finding the opinions and affinity of people towards specific topics of interest. Be it a product or a movie, opinions of people matter, and it affects the decision-making process of people. The first thing a person does when he or she wants to buy a product on-line, is to see the kind of reviews and opinions that people have written. Social media such as Twitter, blogs, twitter have become a place where people post their opinions on certain topics. The sentiment of the tweets of a particular subject has multiple usage, including stock market analysis of a company, movie reviews, in psychology to analyse the mood of people that has a variety of applications, and so on.

The sentiment analysis calculates the overall feeling or mood of users as reflected in social media toward a specific-emotions and determine whether they are viewed positively or negatively. This involves high cost in labour and time. There are few individual studies that have applied SVM, KNN, Decision Tree and Ensemble separately. There are no well-known studies that have combined all these techniques together at same dataset to investigate the variations in technique-based findings. There is no significant study that has applied the above-mentioned machine learning techniques on Twitter data for depression detection.

In this Project, we aim to analyse data from different social media platforms to detect any factors that may reflect the depression of relevant Twitter's users. Various machine

learning techniques are employed for such purpose. Considering the key objective of this study, the following are subsequent research challenges addressed in paper.

- Define what depression is and what are the common factors contributing toward depression.
- What are the factors to look for depression detection in Twitter comments?
- How to extract these factors from Twitter comments?
- What is the relationship between these factors and attitudes toward depression?
- When is the most influential time to communicate within depressive Indicative Twitter user?
- What are the most influential machine learning techniques for detection of depression in Twitter comments?

Users express their feeling as a post or comments in the Twitter platform, sometimes their posts and comments refer to as emotional state such as 'joy', 'sadness', 'fear', 'anger', or 'surprise'. We analyse various features of Twitter comments by collecting data through an effective method of machine learning classification techniques and to make overall judgements regarding their various parts. In this study, we used publicly available Twitter data (from bipolar, depression and anxiety Twitter page) containing users' comments.

1.2 Purpose:

Social networks have been developed as a great point for its users to communicate with their interested friends and share their opinions, photos, and videos reflecting their moods, feelings and sentiments. This creates an opportunity to analyse social

network data for user's feelings and sentiments to investigate their moods and attitudes when they are communicating via these online tools.

Although diagnosis of depression using social networks data has picked an established position globally, there are several dimensions that are yet to be detected. In this study, we aim to perform depression analysis on Twitter data collected from an online public source. To investigate the effect of depression detection, we propose machine learning technique as an efficient and scalable method.

We report an implementation of the proposed method. We have evaluated the efficiency of our proposed method using a set of various psycholinguistic features. We show that our proposed method can significantly improve the accuracy and classification error rate. In addition, the result shows that in different experiments Decision Tree (DT) gives the highest accuracy than other ML approaches to find the depression.

1.3 Sections in this Project:

Sentiments of tweets can be categorized into many categories like positive, negative, neutral, extremely positive, extremely negative, and so on. The two types of sentiments considered in this classification experiment are positive and negative sentiments. The data, being labelled by humans, has a lot of noise, and it's hard to achieve good accuracy?

Currently, the best results are obtained by Support Vector Machine (SVM) for a feature set containing stemming, and Diagram that gives an accuracy of 82.55. The main algorithms used in this project are SVM and Naive bayes and we would be comparing these in the upcoming sections.

The report is organized in the following way.

Section 1 gives the basic idea about what the project is and, the rest of the flow of the project.

Section 2 talks about the related work done in this area and describes the twitter data that was used in this project. Description of the data include the statistical details, as well as the datasets used for testing and training.

Section 3 is a detailed description about the methodology used. The four main important topics include data pre-processing, the machine learning algorithms used, the tools required to execute the project as well as the feature extraction techniques along with features used.

Section 4 reports the results thus far obtained based on the data processing performed and the algorithms used. Discussion is based on two sets of data; one is the smaller set of data on which all the feature set was tested. The other set of data includes the entire dataset of 1.5million tweets. Only a few details on this dataset are reported as the data is very huge to be run even on high memory machines.

Section 5 talks about the future work to be done in this area.

The appendix contains few of the tests performed which did not add much value to the classification results, including numeric features and tf-idf.

CHAPTER - 2: LITERATURE SURVEY

2.1. Related Work:

Research work in the area of Sentiment analysis is numerous. Some of the early results on Sentiment Analysis of twitter data are by Go et al. who used distant learning to acquire sentiment data. They used tweets with positive emoticons like ":)" and ";)" as positive, and tweets with negative emoticons like ":(" as negative. Sentiment Analysis on twitter data has been done previously by Go et al. where they have built the model using Naive Bayes, MaxEnt and SVM classifiers, where they report SVM is better than all other classifiers.

Website: https://towardsdatascience.com/quick-introduction-to-sentiment-analysis-74bd3dfb536c

Sentiment analysis is the automated process of determining whether a text expresses a positive, negative, or neutral opinion about a product or topic. By using sentiment analysis, companies don't have to spend endless hours tagging customer data such as survey responses, reviews, support tickets, and social media comments. Sentiment analysis helps companies monitor their brand reputation on social media, gain insights from customer feedback, and much more!

On the features, they have used Unigram, Bigram, along with Part-of-speech (POS) tagging. They note that unigram feature outperforms all other models and also mention that bigrams and POS tagging does not help. They also perform some pre-processing of the data that was used in modelling the pre-processing techniques used in this project. The text processing, they perform includes removal of URLs, username references and repeated characters in words.

A survey report from Pang and Lee on Opinion mining and sentiment analysis gives a comprehensive study in the area with respect to sentiment analysis of blogs, reviews etc. Algorithms used in the survey include Maximum Entropy, SVM and Naive Bayes.

Website: https://www.sciencedirect.com/science/article/abs/pii/S0950705115002336

As the twitter data is noisy with lot of slang and short words some of the pre-processing techniques using the slang dictionary is mentioned in the paper Apoorva et al., along with its removal of the stop words. They also use the emoticon dictionary which has been implemented in this project to be used in numeric features. They also implement a prior polarity scoring which scores many English words between 1(Negative) to 3(Positive).

From the algorithm point of view, they provide a tree kernel and feature based models. Unigram baseline model is combined with other features and modelled in this paper. Different combination of the features is selected. Part-of-speech tagging and emoticons list from Wikipedia are used for the features.

2.2. Data Used:

Tweets are short length messages and have a maximum length of 140 characters. This limits the amount of information that the user can share with every message. Due to this reason, users use a lot of acronyms, hashtags, emoticons, slang and special characters. Acronyms and slang such as 2moro for tomorrow and so on are used to keep sentences within the word limit. People also refer to other users using the @ operator. Users also post URLs of webpages to share information. Emoticons are a great way to express emotions without having to say much. More details on these are explained in the next section.

The ***data used for this project*** is based out of Sentiment 140 and contains about 1.5 million classified tweets, each row is marked as 1 for positive sentiment and 0 for negative sentiment.

More details about the data are following as under:

Table 2.1 Data Statistics

Type	Count
Positive tweets	790185
Negative tweets	788440
Positive Emoticons	14727
Negative Emoticons	6275
Total words	20952530
Total words without stop-words	13363438
Stop words	7589092

Along with the twitter data, the project also required other datasets like stop words, a dictionary of negative and positive words, an emoticon dictionary3 and an acronym dictionary for twitter slang words. The use of these is described in the next section.

Dictionary of negative and positive words.

The dictionary of negative and positive words is a dataset containing around 6800 negative and positive words. This dataset is used to determine the numeric features of number

of negative and positive words in the tweets, based on which sentiment classification is done. The process of stemming, as explained below is also performed on this dataset, so that it maps to the training and test dataset.

Some negative and positive words from the dataset are shown as below:

Table – 2.2: *Negative and positive words dataset*

Type	Count
Abnormal	Negative
Bothered	Negative
Dangerous	Negative
Dejection	Negative
Aspirations	Positive
Excitement	Positive
Happiness	Positive
Genuine	Positive
Fun	Positive

Emoticons.

Emoticons are a great way to express emotions, especially given the restriction on the length of tweets. Emoticons also form an effective way in determining the sentiment of the tweet. In this project, the emoticons are used as numeric features - positive and negative emotions.

Some of the positive and negative emotions are shown in the table below:

Table 2.3: *Negative and positive emoticons*

Type	Emoticons
Negative	:-/ : :'(:[= :/ :@ :'-(:c ;(=/ v.v
Positive	:-| =p :] :-P ;) :p :3 =] :b :-) 8) :') ;-) :-p :S

The dataset was used in parts and in stages. In the beginning stage, a training set of size 60000 tweets and test-set of size 40000 tweets was used.

This enabled validation, as well as helped in tuning parameters for the algorithm.

For example, while using Linear SVM, the parameter C was tuned to get maximum accuracy. The default value of C was 1 and the value of this parameter giving maximum accuracy was found to be 0.032. For the final experiment, the entire data set of 1.5 million tweets were used with 75% of the data was used for training and 25% for testing.

At different steps of pre-processing, the data was tested using machine learning algorithms such as Naive Bayes, SVM and MaxEnt.

CHAPTER - 3: METHODOLOGY

The main approach involved in this project are the various data pre-processing steps, the machine learning classifiers and feature extraction. The main machine learning algorithms used are Naive Bayes, Support Vector Machines (SVM) and Maximum Entropy (MaxEnt).

The main data pre-processing steps include URL and username filtering, twitter slang removal, stopping words removal and stemming.

Feature extraction includes POS tagging, unigram, bigram (all the above in various combinations) and numeric features all of which are described below.

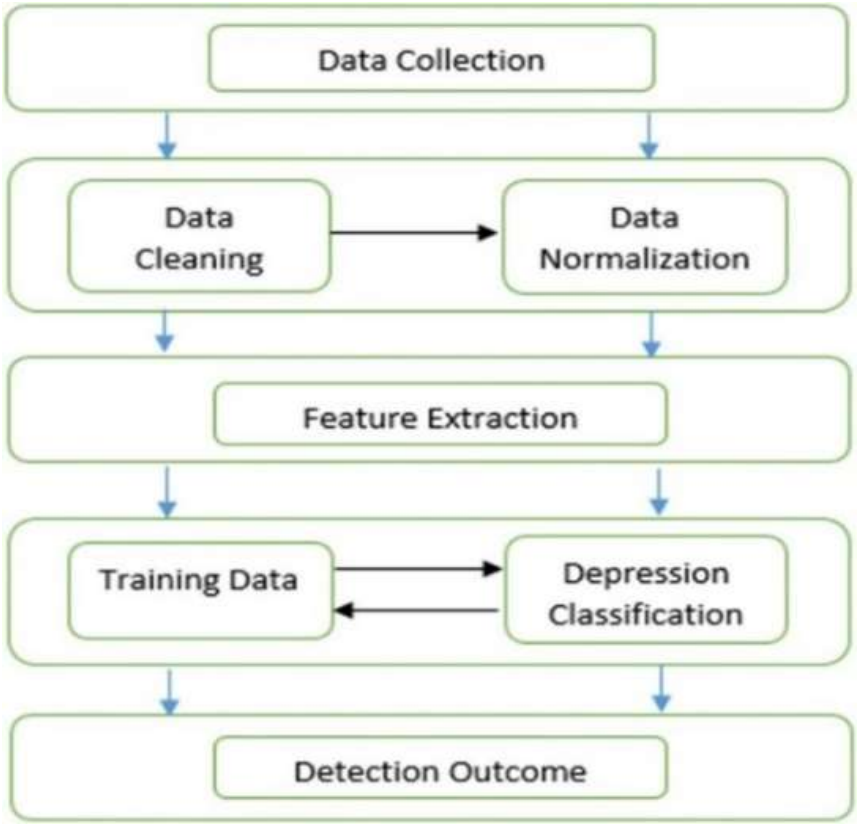

Fig 3.1. Methodology

Table 3.1: *Data*

Item ID	Sentiment	Sentiment Source	Sentiment Text
106	0	Sentiment 140	really wanted Safina to pull out a win; to lose like that...
166	1	Sentiment 140	..... hot Choco is the best!
160	0	Sentiment 140	"RIP, HR."
174	1	Sentiment 140	" :-D))).. What an amazing night! Miss u guys!"

Proposed working methodology are following as under.

3.1 Data Pre-processing:

Data pre-processing is a data mining technique which is used to transform the raw data in a useful and efficient format.

Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn't have a regular, uniform design. Machines like to process nice and tidy information – they read data as 1s and 0s. So, calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

We will collect data from twitter for training and testing purpose to develop our model.

For that we can use either APIs, datasets from Kaggle, input of opinions.

3.1.1. Filtering:

3.1.1.1. URLs:

People use twitter not only for expressing their opinions but also, for sharing information with others. Given the short maximum length of tweets, one way of sharing is using links.

Tweets include various links or URLs and these do not con- tribute to the sentiment of the tweet. The URLs in the data used in this project are of the form http://plurk.com/p/116r50. These do not contribute to the sentiment of the tweet. Hence these were parsed and replaced by a common word, URL.

3.1.1.2. Usernames:

Tweets often refer to other users and such references begin with the @ symbol. These again do not contribute to the sentiment and hence are replaced by the generic word USERNAME.

3.1.1.3. Duplicates or repeated characters:

People use a lot of casual language on twitter. For example, 'happy' is used in the form of 'haaaaaaappy'. Though this implies the same word 'happy', the classifiers consider these as two different words.

Table 3.2: *Data Filtering*

Tweets Containing	Replaced By
http://plurk.com/p/116r50	URL
@abc	Username
cooooooooool	cool
baaaaaad	bad

3.1.2. Twitter Slang Removal:

As mentioned in the previous statement, tweets contain a lot of casual language. Also, given that the maximum length of a tweet is 140 characters, people tend to use abbreviations or some short forms for words. These short words are replaced by the actual words that they represent to improve performance of the learning algorithms.

Table 3.1.2: *Slang Removal*

Twitter Slang	Actual Word
2gethr	Together
bff	Best friend forever
1dering	Wondering
2moro	Tomorrow
2morrow	Tomorrow
tomo	Tomorrow
tmoro	Tomorrow
lol	laugh out loud

The advantage of doing this is evident from the above table. The word *Tomorrow* is used by people using many short forms like 2moro, 2morrow, tomo, tmoro and so on. If these are not mapped to the common original word, then training on them would not produce good accuracy and may also cause overfitting, as these might not be found in the test data.

3.1.3. Stop-words removal:

In information retrieval, there exists many words that are added as conjunctions in sentences. For example, words like *the, and, before, while,* and so on do not contribute to the sentiment of the tweet. Also, these words do not help in classifying the tweets as

they appear in all classes of tweets. These words are removed from the data so as to avoid using them as features.

The stopping words corpus was obtained from NLTK. Some modifications were required to this as the corpus also had some negative words such as nor, not, neither which are important in identifying negative sentiments and should not be removed.

3.1.4. Stemming:

In information retrieval, stemming is the process of reducing a word to its root form. For example, walking, walker, walked all these words are derived from the root word walk. Hence, the stemmed form of all the above words is walk.

NLTK (*Natural Language Toolkit*) provides various packages for stemming such as the Porter-Stemmer, Lancaster-Stemmer and so on. The Porter-Stemmer was used in this project which uses various rules for suffix stripping. In addition to stemming the train and test data, the positive and negative word corpus was also stemmed. Stemming reduces the feature space as many derived words are reduced to the same root form. Multiple features now point to the same word and hence it increases the probability of the word.

Table 3.1.3: *Data Filtering*

Twitter Slang	Actual Word
amazed	amaze
amazing	amaze
amazement	amaze

As we will see in the results section, stemming gives a good increase in accuracy. By stemming, different derived words are mapped to their root words and this allows more matching between the tweets in the test and training set.

3.2. MACHINE LEARNING ALGORITHMS USED:

3.2.1 Baseline:

This experiment uses Naive Bayes with *Unigrams* as a baseline.

3.2.2 Naive Bayes:

The *Naive Bayes classifier* is one of the basic text classification algorithms. It is a simple classifier based on Bayes theorem and makes naive independence assumptions of the feature variables. Despite this very naive assumption, it is seen to perform very well in many real-world problems.

Mathematical representation: Consider attributes X_1, X_2 …. X_n to be conditionally independent of each other given a class Y. This assumption gives us,

$$P(X_1....X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

By Bayes theorem, we have,

$$P(Y|X_i) = \frac{P(X_i|Y)P(Y)}{P(X_i)}$$

Using Bayes theorem in the previous equation, we can find the probability of predicting the class Y given the features Xi. The class that gives the maximum probability that the given features predict it, is the class that the tweet will belong to.

In this experiment, the Naïve-Bayes Classifier from NLTK was used to train and test the data.

3.2.3 Support Vector Machine

Support Vector Machines is another popular classification technique. A support vector machine constructs a hyperplane or set of hyperplanes in a high-dimensional space such that the separation is maximum. This is the reason the SVM is also called the maximum margin classifier. The **hyperplane** identifies certain examples close to the plane which are called as support vectors. Linear SVC from sci-kit learn, which is a python package, is used to classify the tweets.

3.2.4 MaxEnt:

The Max Entropy classifier is a discriminative classifier commonly used in Natural Language Processing, Speech and Information Retrieval problems. The max entropy classifier uses a model very similar to the Naive bayes model but it does not make any independence assumption, unlike Naïve Bayes.

The max Ent classifier is based on the principle of maximum entropy and from all the models, chooses the once which has the maximum entropy. The goal is to classify the text (tweet, document, reviews) to a particular class, given unigrams, bigrams or others as features. If w_1, w_2, …. w_m are the words that can appear in a document, according to bag-of words model, each document can be represented by 1s and 0s indicating if the word w_i is present in the document or not.

The parametric form of the MaxEnt model can be represented as below:

$$P(c|d,\lambda) = \frac{exp[\sum_i \lambda_i f_i(c,d)]}{\sum_c [\sum_i \lambda_i f_i(c,d)]}$$

Here, c is the class to be predicted, d is the tweet, and lambda is the weight vector. The weight vector defines the importance of a feature. Higher weight means that the feature is a strong indicator for the class c. The parameters are chosen by iterative optimization,

and for the same reason, this classifier takes a long time to learn when training size, features are large.

3.3. FEATURE EXTRACTION:

3.3.1. Unigrams:

Unigrams are the simplest features that can be used for learning tweets. The bag-of-words model is a powerful technique in sentiment analysis. This technique involves collecting all words in the document and using them as features. The features can either be the frequency of words, or simply 0s and 1s to indicate if the word is present in the document or not. In this project, 0s and 1s are used to indicate the absence or presence of a word in the tweet.

3.3.2. Bigrams:

Bigrams are features consisting of sets of two adjacent words in a sentence. Unigram sometimes cannot capture phrases and multi-word expressions, effectively disregarding any word order dependence. For example, words like 'not happy', 'not good' clearly say that the sentiment is negative, but a unigram might fail to identify this. In such cases, bigrams help in recognizing the correct sentiment of the tweet.

3.3.3. POS-Tagging:

Part-of-speech tagging in linguistics and information retrieval is the process of tagging each word in a sentence to a particular part of speech. There are many parts of speech such as noun, adjective, pronoun, preposition, adverb, and so on. A word can take different meanings in different sentences, *i.e.,* a word can act as a noun in one sentence, and as an adjective in another.

For this project, the tagger model maxent treebank pos-tagger provided by NLTK was used. Below tree shows a POS- tagging.

NLTK POS-tagging:

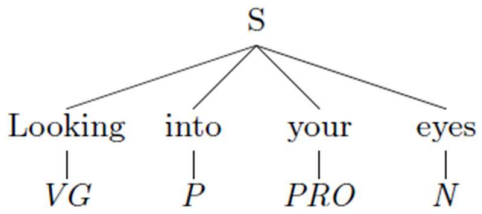

Fig. 3.2 NLTK POS-tagging

3.4. TOOLS:

3.4.1. Natural Language Toolkit:

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. In this project, NLTK was used extensively for tokenizing (tokenizing the tweets), POS tagging, the tagger model being maxent treebank pos-tagger, stemming (as described above, it used the Porter-Stemmer of NLTK), and classification.

The NLTK classifiers used were Naïve Bayes Classifier and the MaxEnt Classifier.

3.4.2. Ipython:

Ipython is a command shell for interactive computing mainly for Python. Few of its main features are its input history across sessions, tab completion, support for visualization and use of GUI tool kits. The IPython also offers a rich text web interface called the IPython

notebook. This project used IPython and IPython notebook extensively for data processing, learning, analysis and visualization, the results of which are discussed in the next section.

3.4.3. Amazon Elastic Compute Cloud:

The Amazon Elastic Compute Cloud or the EC2 is the main component of Amazon's cloud computing platform, Amazon Web Services (AWS). It is a web service that allows users to rent virtual machines to run their applications. To use the large amount of data available for this sentiment analysis task, a high memory, high CPU system was required which led to the need for virtual machines on EC2.

3.4.4. Pandas:

Pandas is a software library written for Python and is used for data analysis and manipulation. Pandas is an open-source library and it also interoperates with the IPython and other Python libraries.

3.4.5. Sci-kit Learn:

Sci-kit Learn is an open-source machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, logistic regression, naive Bayes, random forests, gradient boosting and k-means, and is designed to interoperate with the Python numerical and scientific libraries.

3.4.6. Flask and Web Technologies (HTML, CSS, JavaScript):

Flask is a lightweight web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. Front end technology is particularly vital for making web apps and web pages, being used to craft the design, structure, and animations users see when loading up a website or opening a web application.

CHAPTER - 4: RESULTS

The first set of results show the analysis of the feature set and algorithms for the smaller dataset of 60000 training tweets and 40000 test tweets. When the entire dataset is used, the accuracies are scaled up to a great extent. In general terms, below is the comparison obtained for the smaller dataset.

4.1 Parameter tuning:

The Linear SVC classifier has various parameters that can be tuned depending on the noise in the data. The default value of C is 1. The tuning of parameter C is done while training with smaller set of the data. This was done by running the Linear SVC classifier for multiple C values, the output of which is shown below:

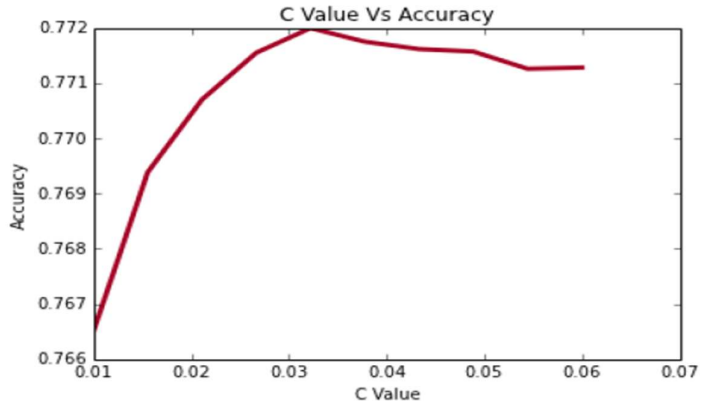

Fig.4.1 Linear SVC classification

We see that the maximum accuracy is obtained when C=0.032. Hence, all the results that are shown in this section use the value of C as 0.032.

4.2 Results with smaller dataset:

The following are the results obtained by data processing, analysis and visualization on a smaller portion of the data, *i.e.,* using 60900 tweets for training and 44000 tweets for tasting.

Table 4.1: *Results with small dataset*

Algorithm	Unigram	Unigram and Bigram	Stemming Unigram	Stemming Bigram	Cleaned Data and Slang removal	Slang removal, stemming and Bigram
Naive Bayes	73.45	75.12	75.35	77.10	74.38	77.01
SVM	77.13	78.29	77.29	78.95	77.62	79.62

4.2.1 Result Analysis using Naive Bayes:

4.2.1.1 *Baseline:*

The Baseline, *i.e.,* Naive Bayes with Unigram gives more details on the most informative features after the classifier was run. The below table shows these details.

Table 4.2: *Naive Bayes (Unigram) most informative features*

Throat	neg: pos = 52.5:1.0
sad.	neg: pos = 47.0:1.0
sad!	neg: pos = 27.4:1.0
welcome!	pos: neg = 25.2:1.0
Rip	neg: pos = 24.8:1.0
Follow-Friday	pos: neg = 22.5:1.0
cancelled	neg: pos = 21.5:1.0
Sad	neg: pos = 20.1:1.0
congratulations	pos: neg = 19.8:1.0
:'(	neg: pos = 17.6:1.0

4.2.1.2 *Effect of Stop-words:*

The algorithms were first run on the dataset without any data pre-processing. When Naive Bayes was run, it gave an accuracy of 73.45 percent, which is considered as the baseline result. The next thing used was stop-word removal. When stop-words were removed and Naive Bayes was run, it gave an accuracy of 73.67 percent.

Table 4.3: *Effect of Stop-words*

Algorithm	Accuracy
Naive Bayes Unigram	73.45
Naive Bayes Stop-words removed	73.67

The results are almost identical, this was the case even with Linear SVC. This shows that stop words do not really affect the predictions much. An intuition to this can be obtained from the fact that given the short length of tweets, people generally avoid the use of stop-words such as and, while, before, after and so on. Thus, removal of stop-words does not make a lot of difference to the accuracy.

4.2.1.3 *Effect of Bigram as a feature:*

Bigram uses a combination of two words as a feature. Bigram effectively captures some features in the data that unigram fails to capture. For example, words like 'not happy', 'not good' clearly say that the sentiment is negative. This effect can be clearly seen from the increase in accuracy from 73.45 (Unigram) to 75.12 percent which is almost a 2% increase. The below table gives the most informative features for Naive Bayes with Bigrams as features.

Table 4.4: *Naive Bayes (Bigram)*

Throat	neg: pos = 52.5:1.0
sad.	neg: pos = 47.0:1.0
sad!	neg: pos = 27.4:1.0
welcome!	pos: neg = 25.2:1.0
('that', 'sucks')	neg: pos = 37.3:1.0
('lost', 'my')	pos: neg = 36.0:1.0
('once', 'you')	neg: pos = 36.0:1.0
('you', 'add')	neg: pos = 34.6:1.0
('so', 'sad')	pos: neg = 32.0:1.0

4.2.1.4 *Effect of Trigrams:*

Running Naive Bayes using Trigrams, bigrams and unigrams together gave an accuracy of 75.19 percent which is almost the same as the accuracy obtained when Bigrams were used as a feature. Also, this feature combination bloats up the feature space exponentially and the execution becomes extremely slow. Hence for further analysis, the trigrams are not considered as they do not have a noticeable impact on the accuracy.

Table 4.5: *Effect of Trigram*

Algorithm	**Accuracy**
Naive Bayes Unigram	75.12
Naive Bayes Stop-words removed	75.19

The below graph shows very clearly the effect on accuracy when stop-words and trigrams are considered. The dashed line shows the bigram, trigram combination which is almost close to the accuracy for bigram. The dash-dot line shows the effect of stop

words on the accuracy. As in the figure, this almost aligns with the Naive Bayes unigram accuracy.

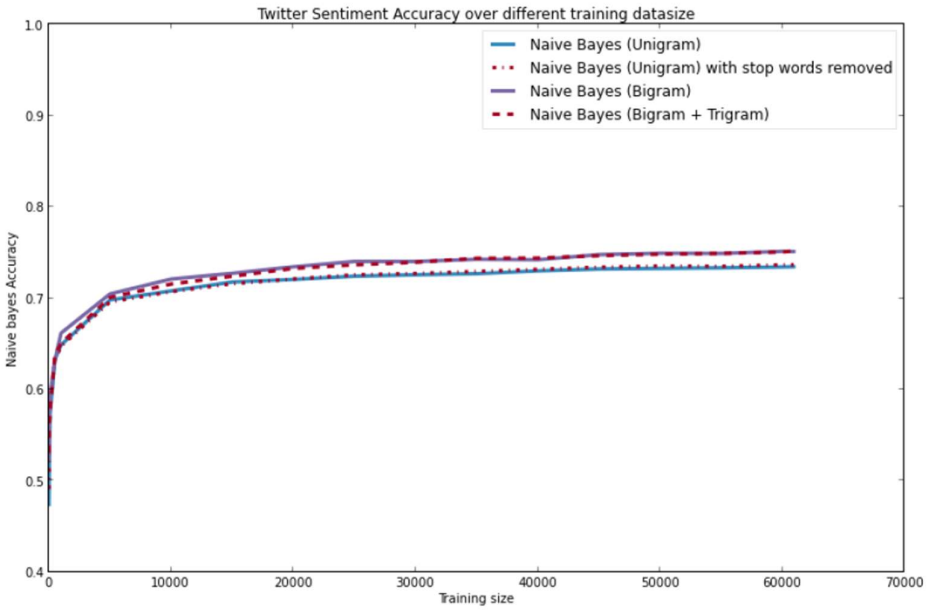

Fig. 4.2 Naive Bayes Accuracy

4.2.1.5 *Effect of Stemming:*

As can be seen in the Results table, stemming has a remarkable effect on the features used. When Unigram is used as a feature with stemmed words, the accuracy of Naïve Bayes increases from 73.45 percent to 75.35 percent, an increase of almost 2 percent. On similar lines, for bigram features, stemming increases the accuracy from 75.12 percent to 77.10 percent. The reason, as mentioned above, is that stemming reduces the feature space as many derived words are reduced to the same root form and multiple features point to the same word, thus increases the probability of the word. In case of SVM, though the increase is not substantial with stemming, the accuracy does improve by a small value.

The below graph shows a comparison of accuracies of Naive Bayes and SVM for the different feature set combinations.

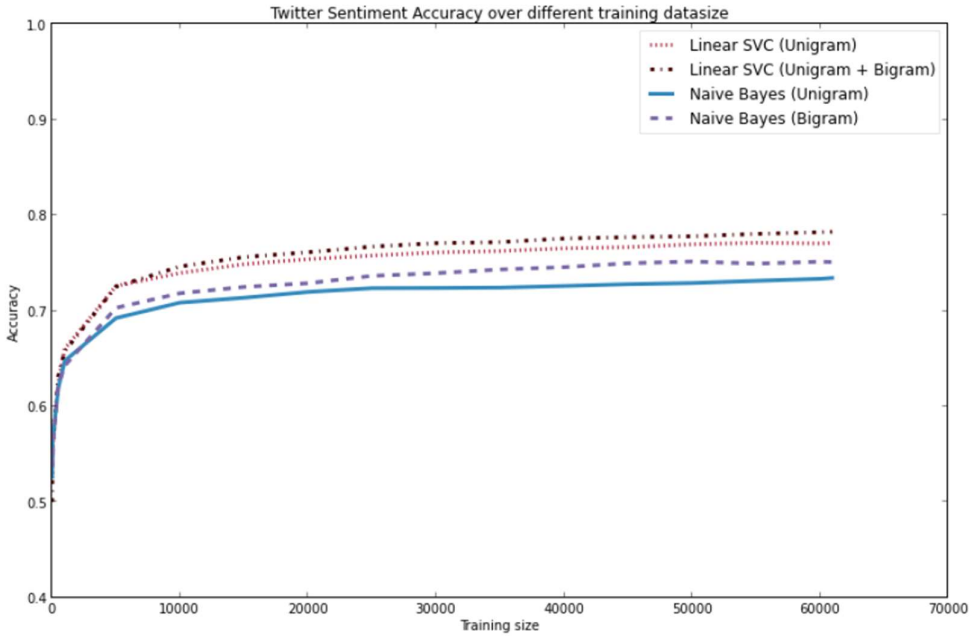

Fig. 4.3 Graph shows a comparison of accuracies of Naive Bayes and SVM

Data filtering and slang removal as an independent feature does give a small improvement in accuracy with an increase from 73.45 percent to 74.3 percent in case of Naive Bayes and increase from 77.13 percent to 77.62 percent in SVM. With all the features considered, the results show that SVM outperforms Naive Bayes in all cases. In particular, the feature combination of Slang removal, stemming and Bigram gives the maximum accuracy of 79.63 with SVM.

Maximum Entropy model gives an accuracy consistently in-between Naive Bayes and SVM. Also, it runs iteratively and takes a large amount of time to run. Hence MaxEnt was not used for all the feature combinations.

4.3 Results with entire dataset:

As the table shows, when the processing, analysis was done on the bigger dataset, the accuracy scaled up to a great extent. Naive Bayes baseline scaled up to 76.39 and SVM scaled up to 80.02 percent.

The best result tested thus far, was obtained when SVM was used on a feature set of a combination of Unigram, Bigram with stemming, giving an accuracy of 82.55. MaxEnt also performed well and gave an accuracy of 77.18 when stop-words was removed.

Table 4.6: *Results with Full data*

Algorithm	Unigram	Stop words removed	Unigram and Bigram	Stemming Unigram and Bigram
Naive Bayes	76.39	76.98	78.65	79.46
SVM	80.02	79.22	82.07	**82.55**

4.4 Screenshots:

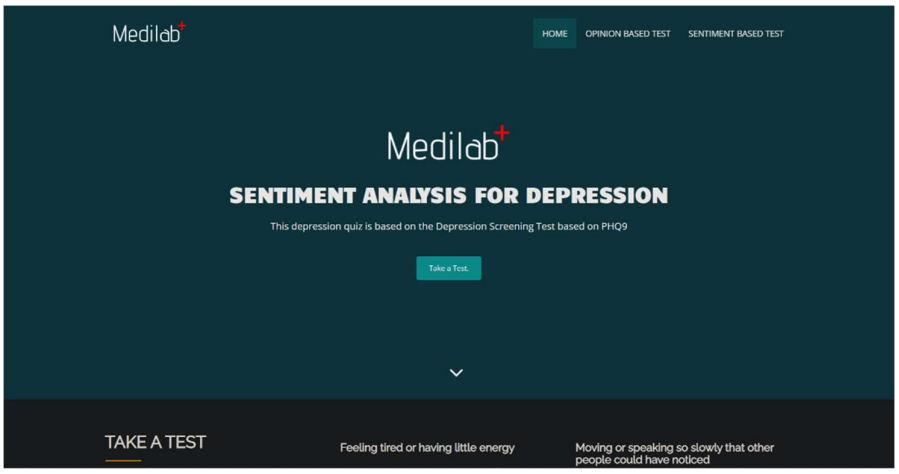

Fig. 4.4 *Home Page*

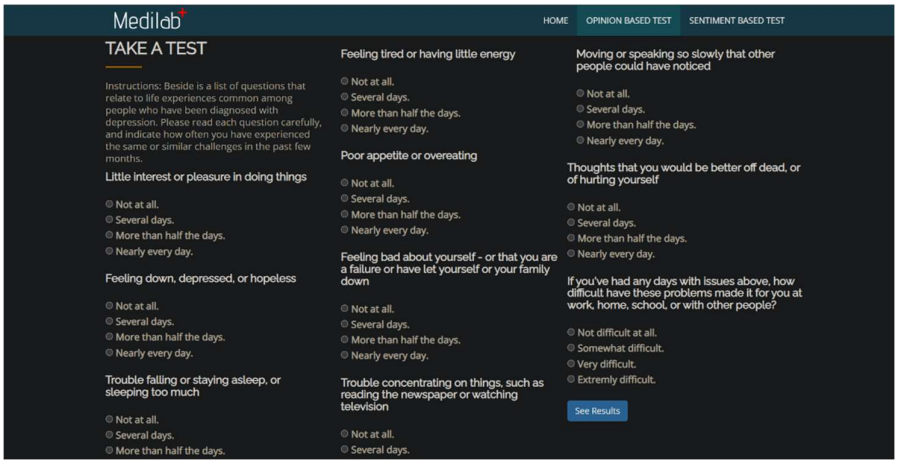

Fig. 4.5 *Opinion Based Analysis*

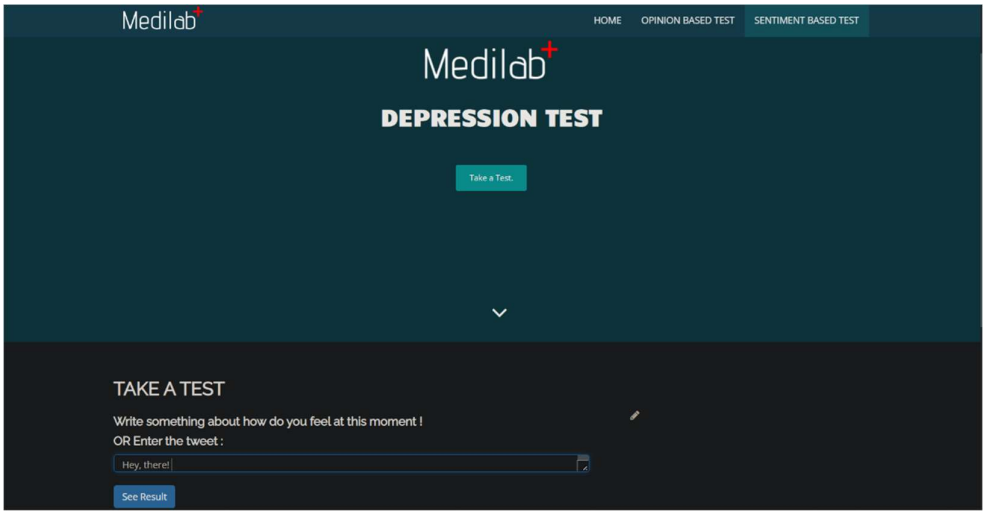

Fig. 4.6 *Feeling / Tweet Based Analysis*

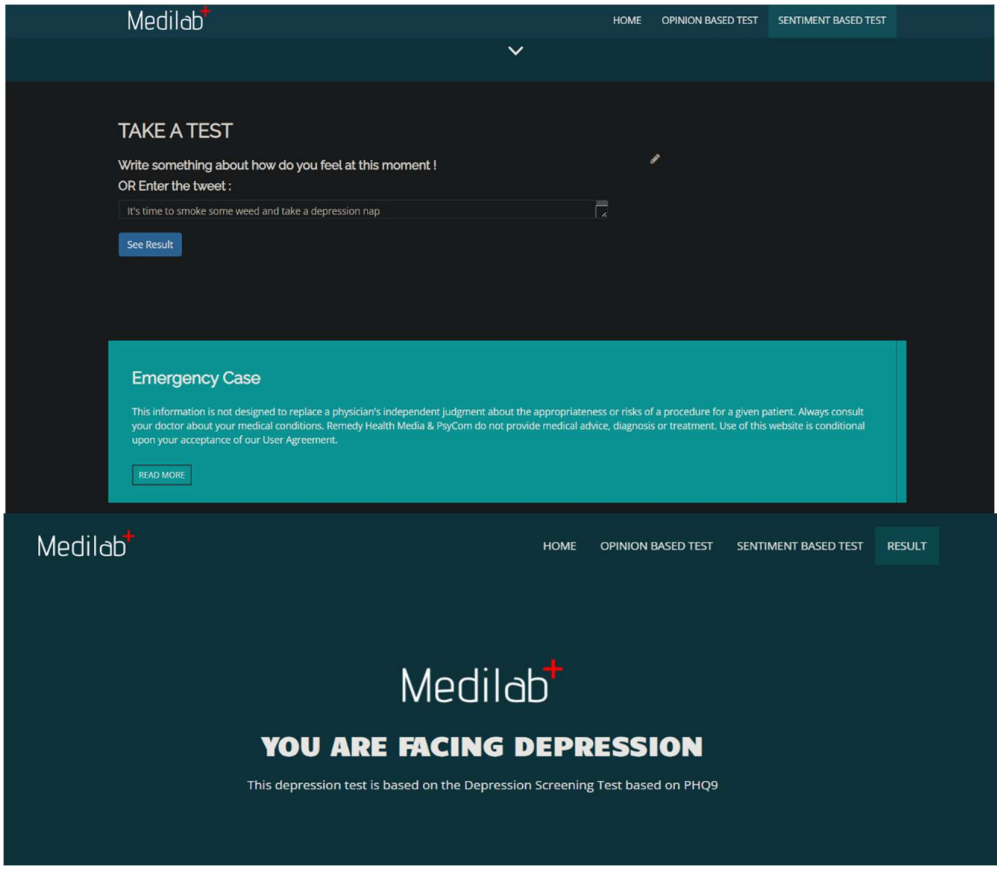

Fig. 4.7 *Test & Result*

CHAPTER - 5: CONCLUSION AND FUTURE WORK

5.1 *Conclusion*:

Twitter is a source of vast unstructured and noisy data sets that can be processed to locate interesting patterns and trends. Apache Spark proved prolific in extracting live streams of data and has further capability to store batches of data in HDFS and other major conventional storages. The processing capabilities of Spark makes the project flexible to further extend to multiple nodes, thereby supporting distributed computing. Real time data analysis makes it possible for business organizations to keep track of their services and generates opportunities to promote, advertise and improve from time to time.

Our heartfelt appreciation goes to Professor Imran Ahmad with regards to his feedback across the course of project from the initial proposal up to the conclusion and for the valuable lessons learned along the way including collaboration within a group and the challenges involved in a large-scale software development effort.

5.2 *Multi-class classification*:

Till now, I have only dealt with binary classification of tweets, either as positive or negative sentiment. There are many tweets, for instance, those with URL's which do not have any sentiment, or, are neutral. These tweets are mainly for sharing some useful information with people, and not necessarily for raising an opinion. As a part of our future work, we would like to explore multi-class classification into various levels of sentiment such as Extremely positive, positive, neutral, negative and extremely negative.

5.3 *More numeric features:*

The numeric features that were used in this experiment include number of negative and positive words, emoticons, length of tweets and number of special characters such as exclamations, hashtags and so on. The numeric features did not yield good accuracy and gave around 63 percent accuracy. Hence, as a part of my future work on this, I would like to generate more as well as smarter numeric features.

5.4 *Use more classifiers:*

In this project, Naive Bayes, SVM and MaxEnt were used extensively. I would also like to explore another machine learning algorithms like Artificial Neural networks. Also, generation of more numeric features will allow me to use more binary classifiers such as logistic regression and so on.

5.5 *Use Hadoop Framework:*

As we have seen, using more data scales up the accuracy. But despite using the amazon EC2, the data still turned out to be very memory and CPU intensive and I was unable to run the SVM and Naive Bayes with POS Tagging and also unable to run MaxEnt algorithms on it. As a part of the future work, I would like to make use of Hadoop for processing large data of this kind.

REFERENCES

A. Books and Research Papers:

[1] Alec Go, Richa Bhavani and Lei Huang,

Twitter Sentiment Classification using Distant Supervision

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze,

Introduction to Information Retrieval

[3] Bo Pang and Lillian Lee,

Opinion mining and sentiment analysis

[4] Tom M. Mitchell,

Generative and Discriminative classifiers: Naive Bayes and Logistic Regression

[5] Christopher M. Bishop,

Pattern Recognition and Machine Learning

B. Websites:

1. https://github.com/topics/sentiment-analysis?o=asc&s=stars/ 18/02/2022,10:11PM(IST)
2. https://github.com/Rakshit-Shetty/Real-Time-Sentiment-Analysis-Using-Twiiter-API/blob/master/Sentiment%20Analysis.py/ 25/02/2022, 11:00PM(IST)
3. https://towardsdatascience.com/a-complete-sentiment-analysis-project-using-pythons-scikit-learn-b9ccbb0405c2/ 03/03/2022,10:30AM(IST)
4. https://arxiv.org/pdf/1602.07563v2.pdf 10/03/2022,01:15PM(IST)
5. https://towardsdatascience.com/quick-introduction-to-sentiment-analysis-74bd3dfb536c/ 18/03/2022,10:17PM(IST)
6. https://github.com/pranitbose/sentiment-analysis/ 22/03/2022,10:30PM(IST)
7. https://github.com/sharmaroshan/Twitter-Sentiment-Analysis/ 24/03/2022,10:17PM(IST)
8. https://monkeylearn.com/blog/sentiment-analysis-machine-learning/ 18/05/2022,08:17PM(IST)
9. https://www.sciencedirect.com/science/article/abs/pii/S0950705115002336/ 01/06/2022,08:17PM(IST)
10. https://patient.info/doctor/patient-health-questionnaire-phq-9 02/06/2022,08:17PM(IST)