DEPRESSION TWEET ANALYSIS

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

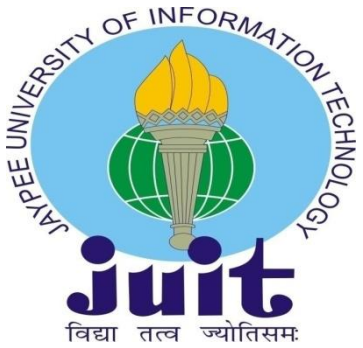
By

Akshay Kumar (181400)

UNDER SUPERVISON OF

Dr. Pankaj Dhiman

To



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

CERTIFICATE

I hereby declare that the work presented in this report entitled **"Depression Tweet Analysis"** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2022 to May 2022 under the supervision of Dr. Pankaj Dhiman, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Akshay Kumar (181400)

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

Dr. Pankaj Dhiman
Assistant Professor (SG)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat

AKCNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully. I really grateful and wish my profound my indebtedness to Supervisor Dr. Pankaj Dhiman **,** Designation, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of Machine learning and AI, carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project. I would like to express my heartiest gratitude to Dr. Pankaj Dhiman ,Department of CSE, for his kind help to finish my project. I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Akshay Kumar (181400)

Table of Contents

Content Page No.

Table of Figures

Figure Page No.

ABSTRACT

Facebook, Twitter, and Instagram, among other social media platforms, have irrevocably changed our world. People are more linked than ever before, and they have developed a digital character. Although social media offers a number of appealing qualities, it also has a number of drawbacks. Recent research has found a link between excessive use of social networking platforms and greater depression. The goal of this research is to apply machine learning techniques to identify a likely sad Twitter user based on his or her network behaviour and tweets. We used data taken from a user's network activity and tweets to train and test classifiers to determine if he or she is sad or not. The findings revealed that the bigger the number of characteristics employed, the better the accuracy and F-measure scores in recognising sad consumers. This is a data-driven, predictive strategy for detecting depression and other mental diseases early on.The key contribution of this work is the examination of the traits and their influence on identifying depression levels.

CHAPTER 1: INTRODUCTION

1.1 Introduction

AnalysingthecontentofTweetshasbecomeanincreasinglymorepopularmethodtounderstandandmakepredictionsabouthumansocial behaviours. Given the wide use of Twitter by the general public, it is a valuable source of data that may be used to investigate a range of these behaviours. For this study, we decided to concentrate our efforts on how obvious it is from a person's Tweets that they are depressed. Depression is a fascinating topic since it is a mental ailment that affects a huge portion of the world's population (350 million people), and it is frequently connected with other mental disorders. Because depression is a multifaceted disorder, it affects various people in different ways and to different degrees.

Depression (Major Depressive Disorder) is a widespread and important medical condition that can adversely affect how you feel, think and behave. It is also, fortunately, treatable. Depression and / or loss of interest in previously appreciated activities. It reduces your ability to work at work and at home along with mental and physical problems.
Symptoms of depression range from mild to severe and include:

1. Sadness is a sad state of mind
2. Loss of interest or pleasure in previously appreciated activities
3. Changes in appetite - Weight loss or weight gain unrelated to dieting
4. Sleepiness or oversleeping
5. Increased fatigue or loss of energy
6. Increased involuntary physical activity (e.g., inability to sit still, move and work with the hand) or slowed movement or speech (these actions should be intense enough for others to notice)
7. Feeling worthless or remorseful

Figure 1 feeling

Because the labels "not-depressed/depressed" cannot be equated with the labels "happy/sad," detecting depression with NLP is more complicated than simple sentiment analysis. Furthermore, depressive symptoms in tweets are frequently mild, and hence not immediately apparent to the human reader. These nuanced clues, on the other hand, may be represented in the intricacies of a person's language, which we believe can be caught by a number of deep learning methods. We intend to construct a robust model that is sensitive to the differences of depression on an individual basis by using a large number of samples and cutting-edge NLP approaches.

Figure 2: Twitter

The findings of this study will be beneficial in predicting depression in people who are unwilling to talk to a professional about their problems. Health professionals will be more aware of which populations are more susceptible to depression, and may develop depression awareness/prevention messages to assist those groups.

People suffering from untreated depression or those who hide their mental illness for various reasons can use social media as a single platform in machine learning architecture, thanks to the development of new, advanced medical techniques to create it all in a trajectory of public disclosure. And technologies. To be clear, billions of people around the world are connected via the Internet is a place where countless conversations, take place every passing second; This information and content is responsible for analyzing a large amount of data that can be used to generate useful information for a variety of industries, including telecommunications, business, health, government security and other multinational and technology sectors.

In addition, there are various platforms for users to read, write post, watch video and catch many new features and techniques in specific media space that respond to social media websites, Twitch, printerest, amazon and online platform. Every one of these platforms represents a specific type of user analysis that can be used in many other technologies.

To be clear, if we select an ad and purchase a product that is advertised in Facebook ads, the meta tag and other restrictions will eventually recommend the next ad as the previous ad, because the user's pattern is determined by machine learning algorithms. Observed and analyzed. Working on the background system.

Similarly, by enhancing the power of data analytics, technology can be developed that will help assess the various insights among social media users. This technology works if we break down their structure according to the most shared customer posts, comments, tweets, job type, position, eating habits and other data posted publicly on social media websites such as those that can be found on Instagram. . , Snapchat, Facebook, Twitter and Tumblr, we may find various unfavorable as well as beneficial results in the future. For example, the tourism and travel industry can analyze data collection through user posts on social media with visited locations, food and hotel options and other relevant information, and obtain various statistics used to determine costs and food. To improve customer needs. , And many other factors.

1.2 Problem Statement

The following are the objectives of our project:

• Construct neural-based architectures (RNN, CNN) to predict sadness using a dataset of labelled tweets generated from scratch

• Generate labeled tweets dataset from scratch

• Fine-tune model settings for best results

• Compare and contrast the performance of character-based and word-based models.

• Compare the performance of pre-trained and learnt embeddings.

Figure 3 Neural network

A Logistic Regression can be thought of as a single perceptron (or neuron). At each layer of an Artificial Neural Network, or ANN, there are many perceptrons/neurons. Because inputs are exclusively processed in the forward direction, an ANN is also known as a Feed-Forward Neural Network:

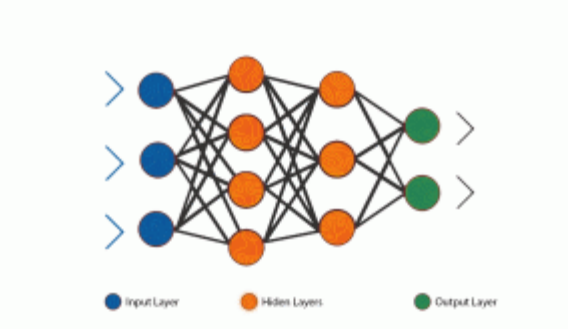


Figure 4 ANN

In the deep learning community, convolutional neural networks (CNN) are all the rage right now. These CNN models are employed in a variety of applications and domains, but they're particularly common in image and video processing projects.

Filters, often known as kernels, are the building components of CNNs. The convolution technique is used to extract meaningful information from the input using kernels. Let's look at how important filters with photos as input data are. A feature map is created by combining a picture with filters:

Figure 5 Output of Convolution

1.3 Objectives

The primary objective of this project is to help and detect depression in peoples tweet this can save life and also by doing this we can provide concealing.

As depression leads to suicide and deaths, hence we can save many people can detect their problem.

Using a variety of common NLP techniques, a lot of past work on sentiment analysis of Tweets has been done. Many of these articles specifically highlighted the idea of creating embeddings for words and/or characters and then feeding those embeddings into the model for training purposes. One research discusses utilising a deep convolutional network to analyse sentiment in brief texts, such as Tweets. The fundamental problem with this form of research is that many Tweets are devoid of context, necessitating a mix of text analysis and prior knowledge to properly classify a piece of text. It collects word and letter level embeddings for all Tweets first, and then uses this information to train neural layers to build sentence level features by examining all of the sentence's windows. This sentence-level feature vector is sent via two standard neural network layers.

Some authors took a more imaginative approach to solving the challenge of Tweet analysis. A publication from MIT [6] coined the acronym Tweet2Vec to describe a technique for learning Tweet embeddings. A character-level CNN-LSTM encoder-decoder is used to learn these embeddings. At its core, it encodes data by extracting features at the character level with a succession of convolutional layers and then successively sending these features through an LSTM layer. It runs the data through two LSTM layers to decode it, and the output can be used to forecast the next character in the sequence.

1.4 Methodology

1.4.1 – Baseline – Logistic Regression and SVM

We used scikitlearn logistic regression with the following parameters as our baseline algorithm.

• Solver = 'lbfgs', Regularization C = 0.1, and SVM classifiers with the following parameters

• To conduct binary classification, use Kernel="rbf" and Penalty for the error term C = 0.025. First, we build word embeddings for each word within each tweet in the dataset using the Word2Vec function from the gensim python library, which is represented as a 100-dimensional feature vector. The feature vector for each tweet is then obtained by averaging all of the words vectors contained within the tweet. Finally, we train and evaluate our baseline models by applying our baseline classifiers to a collection of tweet feature vectors and labels.

1.4.2 – Word Based RNN Model

The first model we implemented after establishing our baseline was a simple RNN model. Because they analyze text in a sequential fashion and can take in any length of input, RNNs are useful for addressing NLP jobs.

The train.csv and dev.csv files were preprocessed so that the RNN model's input was a list of (tweet, label) pairings. Each tweet was represented as a word vector, with each word having a discrete ID that can be used to search up its Word2Vec word embedding. Each tweet was lengthened to a maximum of 30 characters.

We fed a batch of words that corresponded to the tth word into the RNN for each timestept. for each of the current batch's tweets The prediction was produced using the last hidden state, h, which was fed via another neural network layer, followed by the softmax function. The cross-entropy loss between the predicted and true labels was calculated.

1.4.3 – Word based GRU Model

The model was then updated by switching the basic cell unit from an RNN cell to a GRU cell while keeping the word-level embeddings. GRUs can create shortcut connections between text separated by an arbitrary number of timesteps, allowing the model's memory to capture more long-term dependencies in text. They are more powerful than vanilla RNN cells because they incorporate two additional gates (update gate and reset gate), which gives the model more flexibility in processing input. GRU cells outperform LSTM (long short term memory) cells in terms of computing efficiency.

1.4.5 – Character Based GRU Model

We wanted to see how successful character level embeddings were in addition to training models that employed word embeddings. Several papers have suggested that using character level embeddings, particularly for content like Tweets, captures more cases where users repeat letters to indicate emphasis (e.g. "hellllooooOOo"), as well as the fact that a hashtag can indicate something important about the sentence; our word-based model had cleaned out hashtags, but this character, among others, was used in this character-based model.

The character embeddings (for the regular ASCII range) were inspired by and generated from the GloVE word embeddings. The embedding for a specific character is calculated as follows: We infer the character's embedding from the parent embedding every time it appears. The final character embedding is obtained by adding all of these "inferred" character embeddings throughout the entire corpus and dividing by the character's frequency.

1.4.6 – HyperParameters

We employed the following hyperparameters for the three RNN-based models (basic RNN with words, GRU with words, and GRU with chara- caters): Learning Rate:.001, Epochs: 10, Dropout Probability: 0.5, Minibatch Size: 32, Hidden State Size: 300, Learning

Rate:.001, Epochs: 10, Dropout Probability: 0.5, Minibatch Size: 32, Hidden State Size: 300.

1.4.7 – Using a Word Based CNN Model to Generate Sentence Embeddings

We created a convolutional neural network as the final model, which was influenced by1 and 2. CNNs have lately been applied to NLP applications and have demonstrated promising results in text classification, despite their popularity in computer vision.

The train.csv and dev.csv files are preprocessed in the same way that the word-based RNN and GRU models are. This CNN differs from other RNN-based models in that it learns the embedding matrix W during training, whereas RNN-based models employed pre-trained embeddings.

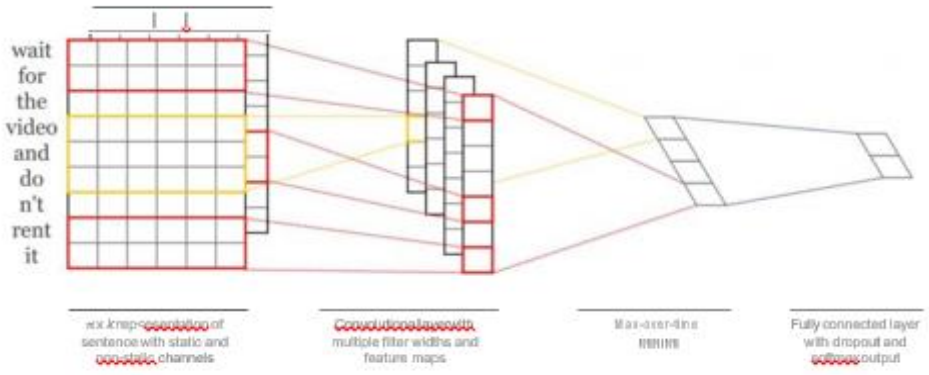


Figure 6 CNN

A simplified depiction of the CNN architecture is shown in Figure 2. To slide over 3, 4, or 5 words at a time, we utilize different filter sizes (3, 4, 5). We calculate the convolution layer outputs for each filter size and apply ReLU nonlinearity and max-pooling to these outputs. On the last layer, we merge all of these pooled features from each filter size to generate our feature vector. On this feature vector, we apply a dropout layer with a probability of 0.5 and calculate the loss using cross entropy.

In this part we will explain the organization of the chapter wise formatting of this whole report. In chapter one we have already seen the Introduction, Problem Statement. After that we saw the objectives and the methodology of the report. Now we stand in the Organization section that would explain the chapters and topics involved further down in this report.

Chapter 2 will be presenting a brief overview of the literature survey process. This will include some of the resources we used in the past to gain necessary knowledge in view of this project and other work we came across through the community that set us forward to reach to this stage where we were able to complete this project and develop a base work to solve these type of problems in the future. Then will follow

Chapter 3 where we would be discussing about how our model is designed and developed. First we would discuss about the dataset that we used in making this report and see some basic features. We will discuss about the basic framework which help us lead to random forest algorithm that include decision tree and feature selection. We would take a look at the various models we applied to draw comparisons with the random forest and their respective accuracies. We have explained them a bit to give the reader a better understanding of them. After that we have also listed some of the mathematical formulas used in this report.

In chapter 4 we have shown how our system performed and also compared them with the other systems used earlier like linear regression ,decision tree etc. Then we begin to showcase the outputs at various stages of our project work that include various graphs and figures. This would conclude with the final output of our model.

Finally, in chapter 5 we started the conclusion part and explained about what we managed to achieve in the project and how this model was the perfect way for us to learn and explore in this field. After that we proposed some future plans that can be achieved through this project and made an extension to it.

CHAPTER 2: LITERATURE SURVEY

This chapter addresses the previous study of the relevant research work related to the purpose, methodology, results, strengths and weaknesses of this study. Also, see the specifics of any changes made to previous research for future use. Finally, a comparative analysis based on a literature review is presented in the form of a table to better understand the content required in previous research publications.

To start with, one of the most important study papers. Aiming to diagnose depression through social media, it describes a serious condition called "depression" that affects the personal and professional lives of millions of people. In both cases there is damage. They use crowdsourcing to explore social media to search the exact structure for which they measure the exact number of people living with depression in comparison to the proportion of people who cannot receive effective treatment. The use of surveys and the collection of Twitter profile access from people with depressive symptoms helped them match their surveys and capture tweets from publicly exposed profiles.

They measure depression as a disorder called MDD or Major Depressive Disorder. He used crowded labor on Twitter to conduct the CES-D (Center for Epidemiological Studies Depression Scale) screening test for MDD patients. They reduced the noise from the dataset, which means they removed the records that they thought would lead to biased results. The authors also collected information about patients' medical histories. They gathered many attributes from data such as size, answer, question, activity, follower, dominance, and so on. He used supervised learning methods and the support vector machine algorithm of all the monitored algorithms he tried gave the best results to his dataset. He developed and evaluated his model by combining all the features as well as the individual features and reducing the size of the features. They double-checked their results using the t-statistical test.

The 476 user samples, on the other hand, are insufficient to generalize depressed behavior across all groups. In addition, big and diverse samples from a variety of age groups can aid in the research of various behavioral traits in people of all ages and genders. Most Twitter

users use emoticons to communicate their feelings, which can be used as one of the features to help detect the current state of the user's emotions.

According to a report, depression is one of the fastest growing mental health disorders. This study aims to look at the level of frustration in people who use social media to express themselves. They used social media indexing to rank their depression levels after collecting data from users on Twitter and applying monitored learning algorithms. They collected data by building the AMT platform and filling their data files with crowdsourcing methods. To assess the severity of the depression, they used the CES-D questionnaire. They form a positive section and a negative section containing data on people suffering from depression, which usually includes data from depressed users. Several tools have also been developed to assist in classifying locations. These features were developed using two methods: post-centered and user-centered methods.

Indicates different conclusions for work error; However, within the limits many characteristics can be inferred from this study by evaluating models of user activity, developmental models of post-depression, and the language used by the affected individuals in their early stages. Furthermore, as the authors point out, some people on the Internet use private Twitter, making it difficult for all users to track activity and diagnose the disease. In addition to Twitter, people can add more social media channels like Facebook and Instagram to share their thoughts.

Furthermore, in a research paper the authors proposed the idea of jealousy and greed introduced through the conscious use of social media platforms. Many social media users choose to publish personal and professional updates in a way that puts them ahead of the competition. This scenario has created an imaginary race among consumers and a sense of competition among them. This constant stress of wanting to look perfect to others can have detrimental consequences for mental health, including depression. The authors analyzed diaries and experimental research taking into account individual users and the authors proposed a link between Facebook usage and frustration. Diary research by Steers and colleagues found that disgust contributes to depression compared to social media.

CHAPTER-3 SYSTEM DEVELOPMENT

This chapter discusses the dataset’s origins, collecting process, strengths, and drawbacks. Furthermore, a dataset analysis is carried out in order to provide a clearer picture of all previous datasets that are available in tabular form. Finally, the implementation procedure for the algorithm in the dataset and computing them with the metrics .

Depression is one of the most common mental health problems that can affect a person's daily routine and lead to serious conditions such as suicide. People with depression may not be aware of the difficulties they are experiencing, but their social activities and behaviors can be used to diagnose depression in advance. Following this understanding, many educators began using social media features to find samples of affected and suspected patients on social media. The language or vocabulary of depressed patients may have a direct or indirect association with contextual symptoms.

Depression. In recent years, digital text analysis has made it possible to compute vast data stores in a matter of minutes. U.S. According to, people who suffer from depressive symptoms when using social media have many symptoms. Examples of these symptoms include over-checking of email, use of social media during peak nights, words and abuses associated with negative impact, or showing a bad mood in a person's virtual social circle.

People in today's world are ready to post updates on social media in minutes or seconds. Social media has evolved into a repository for human feelings and intellectual concepts as a result of its widespread use. For e.g , if we tweet "I've been having sleepless nights since my spouse died," we may be sure that the individual is suffering from mental illness and is on the verge of losing their mind. Between 2012 and 2016, researchers from the University of Pennsylvania collected approximately 400 million tweets from individuals in their nation.

They put folks who have written about 'being alone' more than five times into a bucket.

Those who report these types of tweets have the opportunity to talk about their daily life difficulties [uneasiness, alcoholism, drugs, insomnia and tweeting regularly at night]. Is. In both cases the dictionary buckets used by the users were also examined. Those who spoke most about 'kindness' also posted phrases such as 'prayer', 'miracle' and 'family', while another group reported 'feeling,' "myself," and 'no longer'. More Posted About In addition to their results, preliminary research has suggested that the lexical aspects of the language of depressed consumers may play an important role in distinguishing them from those in a better position.

Depression is a disorder that is affecting more and more people. Its impact can also be seen in the form of postings on social media. Of these, 534 are speaking. To examine the effects of depression, not only the vocabulary of the tweets but sentence and grammatical structure are also required. Consequently, this study addressed five main sentence level features to examine the difference between depressed and depressed. People. As a result, detailed architecture is illustrated in the illustrations based on the motivation and observation of the above criteria.

3.1 DESIGN

To interact with the user we created a web app which take tweets as input from the user and show the result as follow using the streamlit.

3.1.1 STREAMLIT

Streamlit is an open-source Python module that simplifies and beautifies the building of online applications. This library, like many other Python libraries, helps data scientists with a difficult and complex task. Creating a front-end for a developer is not as big of a deal as it was before.

Before we start building the front-end, there are a few things to consider. I think sketching out the demo of app that we want it to look like is a fantastic idea. It could be useful to know what Streamlit can do in terms of layout at this time. Streamlit presents items in a linear fashion by default. As a result, whatever you want to display goes underneath the preceding item.

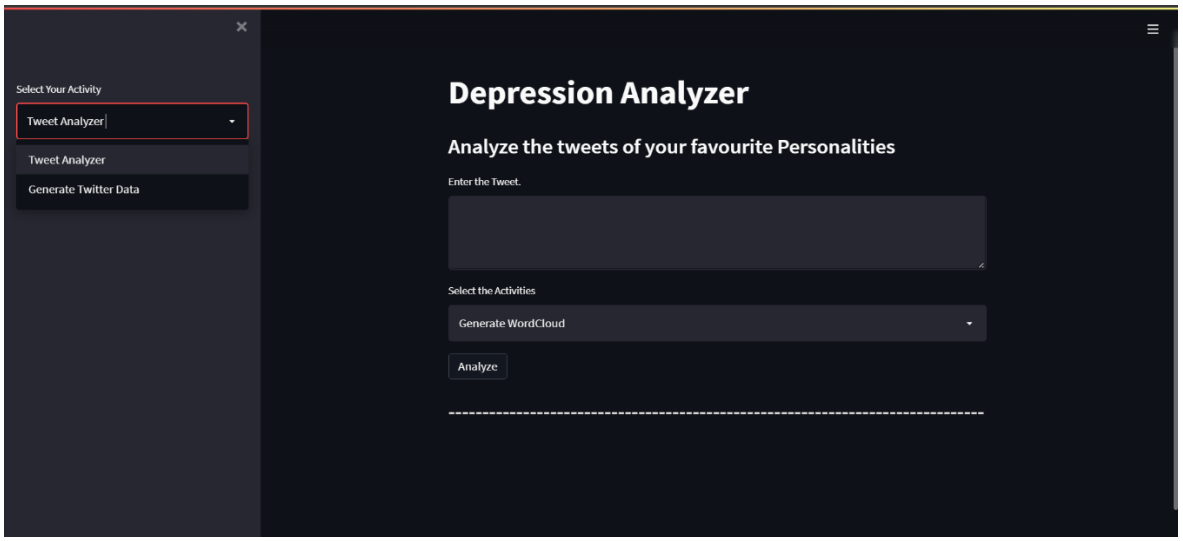


Figure 7 Web App

We added some text in the web app to instruct the user and navigate through it. A text box is created where user input its text which then process through the model to check the depression. The visualization is also given which visualize the text in the word cloud for user to check how is data is given to model to train it. At the end there is the analyze button which on clicking gives the result in depressed or non-depressed. A side nav bar is given to Change the input method to multiple input according to the user choice.

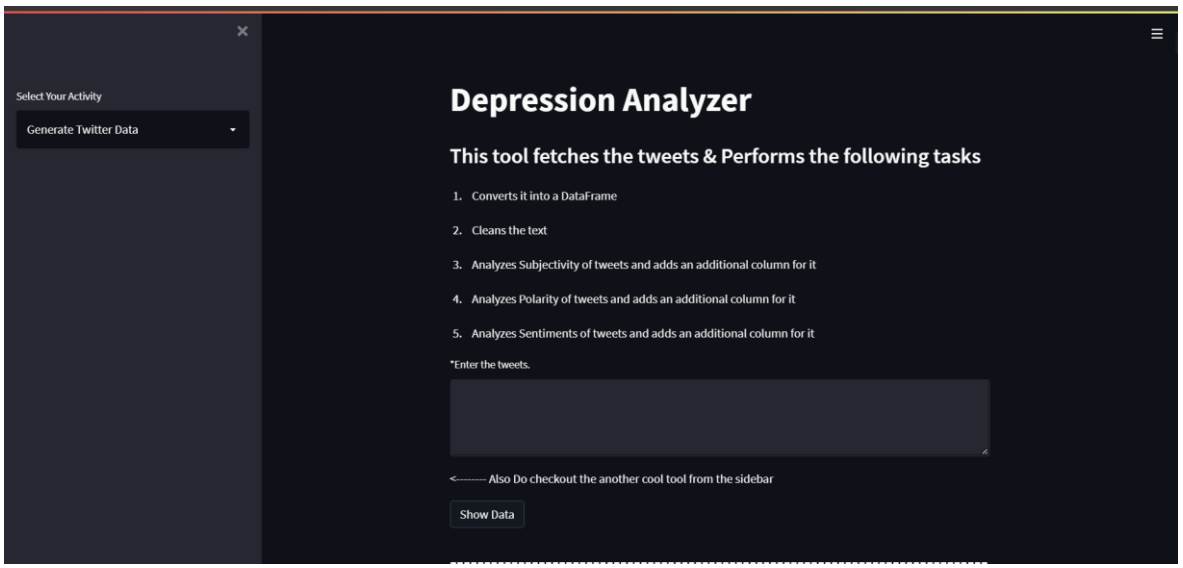


Figure 8 Web App2

3.2 ALGORITHM

Using a variety of common NLP approaches, a lot of past work on sentiment analysis of Tweets has been done. Many of these articles specifically highlighted the idea of creating embedding's for words and/or characters and then feeding those embedding's into the model for training purposes. One research discusses utilizing a deep convolutional network to analyze sentiment in brief messages, such as Tweets. The fundamental problem with this form of research is that many Tweets are devoid of context, necessitating a mix of text analysis and prior knowledge to properly identify a piece of text. It collects word and letter level embedding's for all Tweets first, and then uses this information to train neural layers to build sentence level features by examining all of the sentence's windows. This vector of sentence-level information is sent through two regular neural network layers, with the final layer's output serving as the prediction. Because it detailed successful techniques of leveraging character embeddings in creating sentence-level embeddings, this study was helpful in establishing our strategy.

Some authors take a more imaginative approach to tackling the challenge of Tweet analysis. A publication from MIT coined the acronym Tweet2Vec to describe a strategy for learning Tweet embeddings. A character-level CNN-LSTM encoder-decoder is used to learn these embeddings. At its heart, it encodes data by extracting features at the character level with a succession of convolutional layers and then successively sending these features through an LSTM layer. It runs the data through two LSTM layers to decode it, and the result may be used to forecast the next character in the sequence.

So, CNN_LSTM is going to be main algorithm of this project . We are going to use CNN_LSTM to detect depression tweets from the dataset and compare it other algorithm like logistic regression , RNN and find out how effect it is in detecting the depressed tweets as compare to other algorithms. Before entering into its implementation here is some insight of the algorithm.

3.2.1 CNN_LSTM:-

The CNN Long Short-Term Memory Network, or CNN LSTM, is an LSTM architecture intended primarily for sequence prediction issues using spatial inputs such as pictures or videos. Convolutional Neural Network (CNN) layers for feature extraction on input data are paired with LSTMs to facilitate sequence prediction in the CNN LSTM architecture.

CNN LSTMs were created to solve visual time series prediction issues and to generate textual descriptions from picture sequences (e.g. videos).

In particular, the issues of:

- Activity Recognition is the process of creating a textual description of an activity shown in a series of photographs.
- Creating a written description for a single image is known as image description.
- Generating a written description of a succession of pictures in a video.

Although we will refer to LSTMs that employ a CNN as a front end as "CNN LSTM" in this course, this architecture was initially referred to as a Long-term Recurrent Convolutional Network or LRCN model.The duty of creating textual descriptions of photographs is handled by this architecture. The usage of a CNN that has been pre-trained on a difficult picture classification assignment and then repurposed as a feature extractor for the caption producing challenge is crucial.

CNNs have been utilized as feature extractors for LSTMs on audio and textual input data in this architecture, which has been applied on voice recognition and natural language processing challenges.

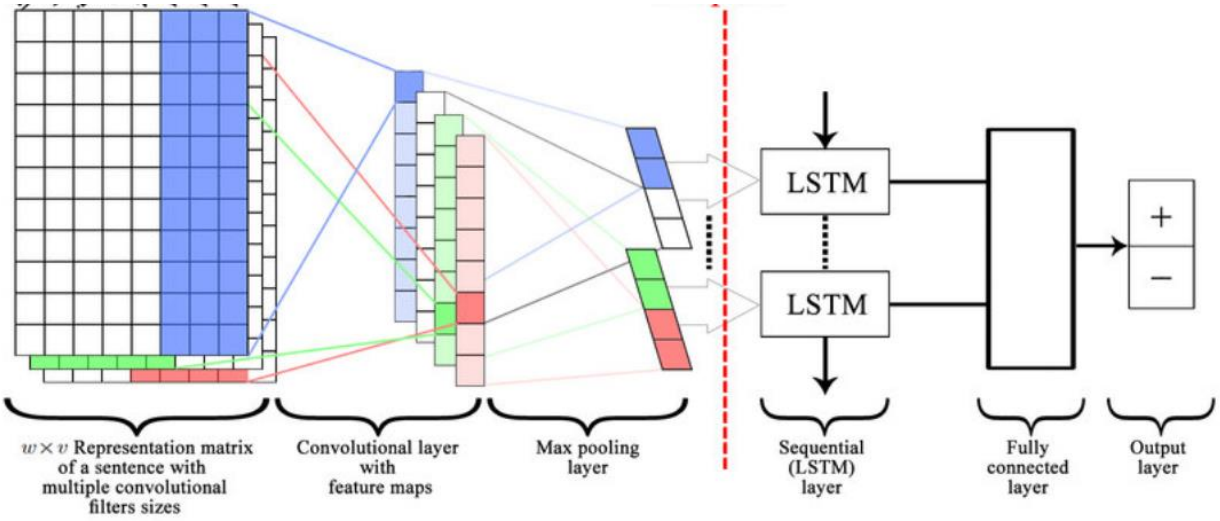


Figure 9 CNN-LSTM

This design is suitable for the following problems:

- Have spatial structure in their input, such as 2D pixels in a picture or 1D words in a phrase, paragraph, or text.
- Have temporal structure in their input, such as the order of pictures in a video or words in text, or necessitate the development of temporal structure in their output, such as words in a textual description

3.2.2 RNN: -

RNN works based on the unstable nature of the perceptron. In addition, they cooperate in the model through a mixture of timely connections and proximity connectivity. They can hold a considerably wider set of instructions gleaned from the efficacy of previous occurrences. However, a non-linear gesture workflow allows for a variety of complex approaches to adjust the concealed layer for updating. Recurrent neural networks, on the other hand, may generate a variety of complex sequels by storing information units in a large number of little pieces and running programmers alongside the ensuing unique new transformations.

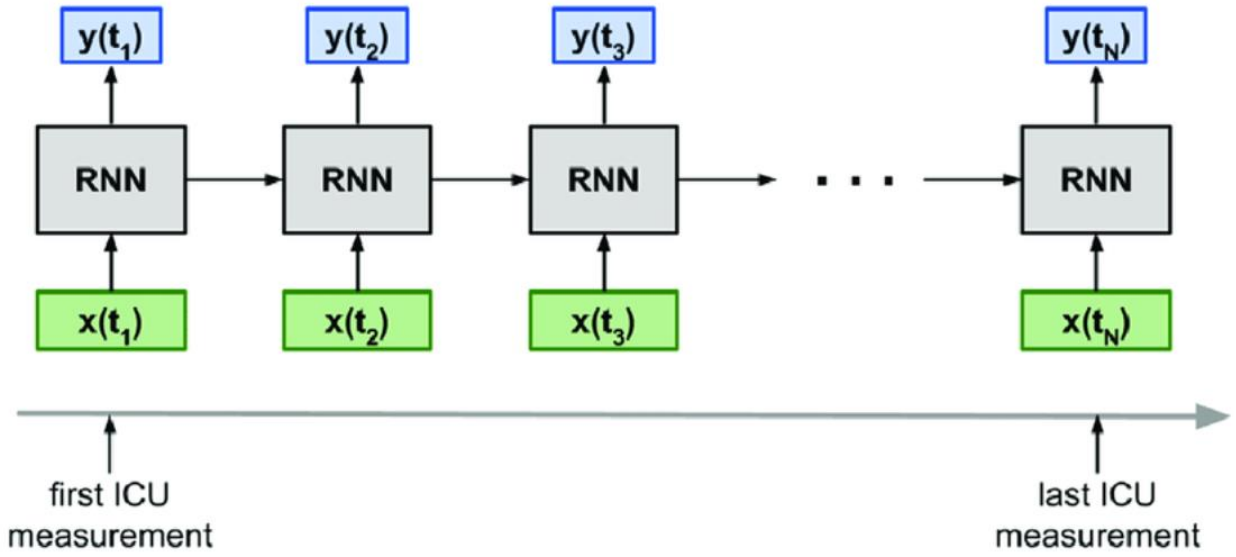


Figure 10 RNN

3.2.3 Logistic regression:

A classification strategy for selecting separate classes in a collection and allocating data to each selected category, as opposed to linear regression, which outputs persistent numerical values. Furthermore, in Logistic regression, a sigmoid function is used to provide a 119 estimation of the probability of working on the calibration of two or more different classes. Depending on the use of one or more predictive parameters, the logistic regression method produces only two results (dichotomy) .It is a specific case of linear regression in which the output belongs to a single category and is created by applying a logarithmic function to the projected probability.

The prediction equation for the logistic regression for two variables can be defined as:

Output = exp (a0 + a1 *x)/(1+ exp (a0 +a1 *x))

3.3 MODEL DEVELOPMENT

3.3.1 Baseline Model :

Logistic Regression:-We used scikitlearn logistic regression with the following settings as our baseline method

- Regularization *C*= 0.1 and Solver='lbfgs'.

to classify data into binary categories .First, we build word embeddings for each word within each tweet in the dataset using the Word2Vec function from the gensim python package, which is represented as a 100-dimensional feature vector. The feature vector for each tweet is then obtained by averaging all of the words vectors included inside the tweet. Finally, we train and evaluate our baseline models by applying our baseline classifiers to a set of tweet feature vectors and labels.

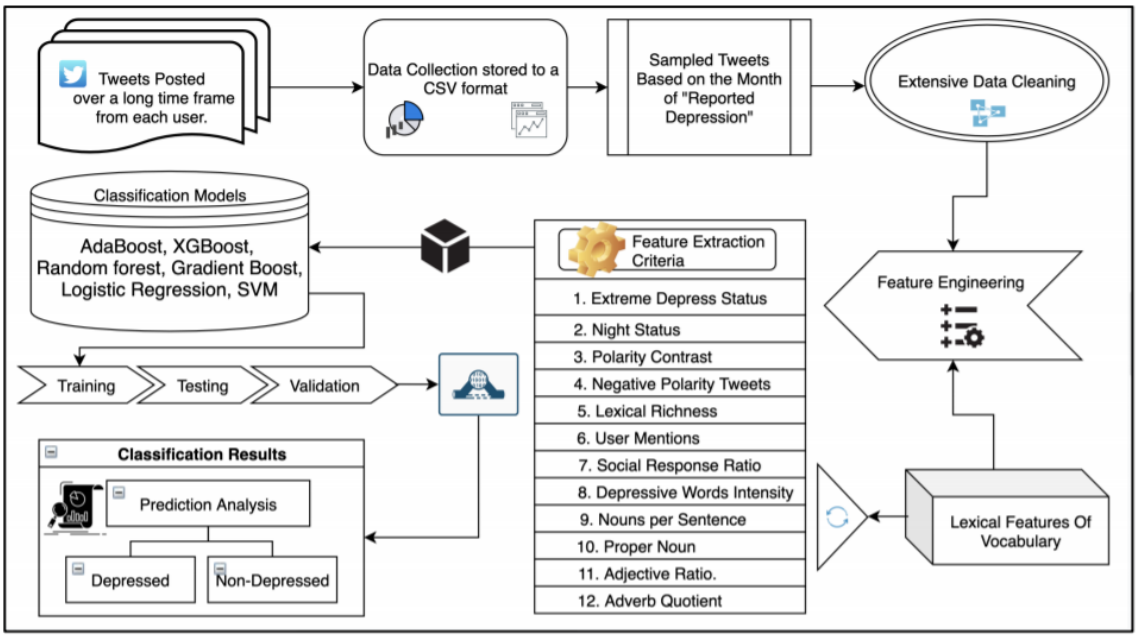


Figure 11 Classification Model

3.3.2 Word-based RNN Model:

The first model we constructed after establishing our baseline was a basic RNN model. Because they analyze text in a sequential fashion and can take in any amount of input, RNNs are useful for addressing NLP task.

The train.csv and dev.csv files were preprocessed such that the RNN model's input was a list of (tweet, label) pairs. Each tweet was represented as a word vector, with each word having a discrete ID that can be used to search up its Word2Vec word embedding. Each tweet was lengthened to a maximum of 30 characters.

We input a batch of words that corresponded to the t^{th} word into the RNN for each timestep for each of the current batch's tweets. The prediction was produced using the last hidden state, h, which was passed via another neural network layer, followed by the softmax function. The cross-entropy loss between the predicted and true labels was calculated. During training, we employed the Adam optimizer to reduce this loss as follows:

$$h^{(t)} = W_t h^{(t-1)} + W_x x_t + b_1$$
$$o = U h^{(t_{final})} + b_2$$
$$softmax = \frac{e^o}{\sum_{j=1}^{n} e^j}$$
$$CE = \sum_{i=1}^{n} y_i log(\hat{y}_i)$$

3.3.3 Word-based CNN Model to Generate Sentence Embeddings

We created a convolutional neural network as the final model, which was influenced by 1 and 2. CNNs have lately been applied to NLP applications and have demonstrated promising results in text classification, despite their popularity in computer vision.

The train.csv and dev.csv files are preprocessed in the same way as the word-based RNN. This CNN differs from previous RNN-based models in that it learns the embedding matrix W during training, whereas RNN-based models employed pre-trained embeddings.

A simplified depiction of the CNN architecture is shown in Figure .To scroll over 3, 4, or 5 words at a time, we utilize different filter sizes (3, 4, 5).We calculate the convolution layer outputs for each filter size and apply ReLU nonlinearity and max-pooling to these outputs. On the last layer, we merge all of these pooled features from each filter size to generate our feature vector. On this feature vector, we apply a dropout layer with a probability of 0.5 and quantify the loss using cross entropy.

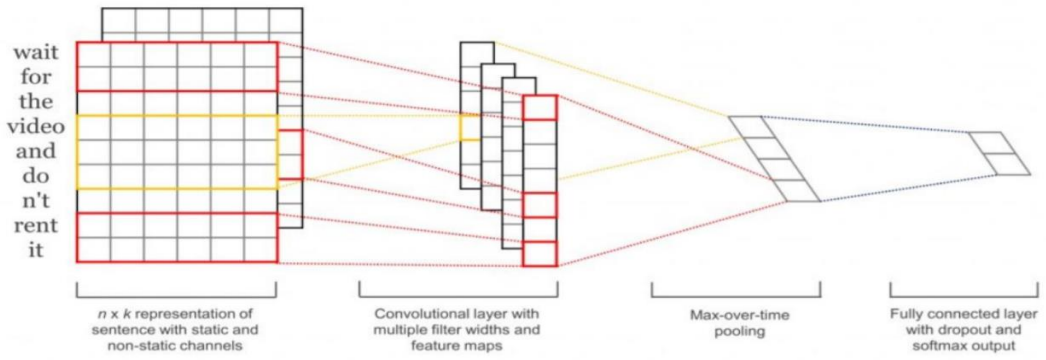


Figure 12 CNN-Model

3.4 EXPERIMENTAL SETUP

This part covers the technical components of the research, such as setting up the libraries in Python, data collecting, LDA, word embedding combinations, description of ensemble learning models using pseudo codes, and techniques for calculating the findings employed in this study.

3.4.1 Loading Libraries

The initial step in implementing the experimental module is to import a number of Python libraries that will help load loads of statistical and graphical analysis, plot sequences, and training and testing of the algorithms used in this study. Some important libraries are Tweep, Pandas, Seaborn, NLTK, Numpy & Keros, Genesim, Geopi, Tensor Flow and Schlern.

3.4.2 DATASET

Our dataset included 13,385 tweets collected from various Twitter pages, each of which was tagged with a 1 or 0 to indicate whether the individual who sent the tweet suffers from depression or not. The train set (train.csv with 10,708 tweets) and the dev set (dev.csv with 2,677 tweets) are jumbled and chosen at random from the aforesaid dataset.

	0	1	2	3	4	5	6	7
0	989292962323615744	2018-04-25	23:59:57	Eastern Standard Time	whosalli	The lack of this understanding is a small but ...	1	0
1	989292959844663296	2018-04-25	23:59:56	Eastern Standard Time	estermnunes	i just told my parents about my depression and...	1	0
2	989292951716155392	2018-04-25	23:59:54	Eastern Standard Time	TheAlphaAries	depression is something i don't speak about ev...	0	0
3	989292873664393218	2018-04-25	23:59:35	Eastern Standard Time	_ojhodgson	Made myself a tortilla filled with pb&j. My de...	1	0
4	989292856119472128	2018-04-25	23:59:31	Eastern Standard Time	DMiller96371630	@WorldofOutlaws I am gonna need depression med...	0	0

Figure 13 Dataset

3.4.3 Collection of Dataset

We gathered the data using a number online Twitter scraper programmers, including .All collected tweets were labelled in the start.

To gather possibly positive examples, we looked for sites on Twitter that dealt with depression in some way; particularly, the term "depression" had to appear either in the user name or title of the page. In our dataset collection, sites like "Depression Quotes," "Depression Notes," and "Damn Depression" were all relevant. Because many of these pages were not owned by individuals who were possibly depressed, we avoided scraping Tweets from pages that were more aimed toward depression awareness or those administered by organizations that are striving to help people with depression. We gathered their accompanying tweets and organized them into a list of possibly good instances after analyzing the content of the pages we had nominated as "legitimate" pages.

We gathered samples from a wide variety of pages (news, sports, dance teams, etc.) for possibly negative instances, as well as pages dealing with other emotions (anger, happiness, etc.) so that our model could discriminate between depression-related emotions and other emotions.

We manually tagged the tweets after collecting them; to facilitate with labeling, we built a script that calculates a polarity score for each tweet using the Textblob python module; based on these values, the script classifies each tweet as 1 for depressed, 0 for no depressed. After running this script, we manually comb over all of the tweets to see if the Textblob algorithm's predictions are accurate, and we change the labels appropriately.

This was an example of a "depressed" tweet:

"I attempted suicide depressed"
"I am anxiety patient"
"I am suffering from anxiety"
"I am feeling suicidal thoughts"
"I Blurred no idea of life stressed"
"I have been diagnosed with depressed"
"I have been diagnosed with depression"
"I am on depression medications"
"I am suffering from depression"
"I am poor mental health"
"I sleepless nights depression"
"I irritate depressed alone"
"I attack nausea episodes"
"I overweight depression"
"I am under treatment depressed"
"I am on antidepressants"
"I take drugs depressed"
"I am under chemotherapy depressed"
"I have been diagnosed with Postpartum depression"
"I am suffering from postpartum depression"

Figure 14 Tweets

3.4.4 Cleaning The Dataset

When using data, most people will agree that your insights and analytics are only as good as the data you are using. In particular, the worst analysis is out of the worst data. Also known as data cleaning, data cleaning and data scrubbing, it is one of the most important steps for your organization if you want to build a culture around quality data decision making. Data cleaning is the process of fixing or deleting faulty, corrupted or corrupted data. Malformed, duplicated or incomplete data in the dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or labeled incorrectly. If the data are incorrect, the results and algorithms may appear to be correct but may not be reliable. There is no absolute way to determine the exact steps in a data cleaning process,

as the procedures vary from dataset to dataset. But it is very important to set up a template for your data cleaning process, so you know that you are doing it right every time. converting tweets to a supported data structure (essentially a list), and then eliminating extraneous data from the tweets. The following routines were created for various data cleaning processes:

The data cleaning procedure begins by going over each document or tweet character by character, skipping all unnecessary information except the words that are critical to the sentence's composition. The techniques for removing superfluous information are as follows:

(i) Punctuation Removal: Punctuation marks are symbols that help in the formation of complex phrases. The functional module "() * _'|, +, -. /: ; = >?,! " # $ percent &, @ []" " contains punctuation that matches the raw text.

(ii) Deletion of HTML tags: Most real-world data scraped from Internet resources contains HTML elements such as p>, <h1> and
, which add noise to the data and also consume information, creating memory and space. Recovery phase. The 'Beautifulsoup' package, which assists in the data cleaning process by extracting data from HTML and XML context-based files, is used to remove this unusable and unnecessary context.

(iii) Stop Words removal : Stop words are the most often occurring or repeating words in sentences. “And, or, not, like, could, have, desire, where,..." are examples of popular stop words. Stop words are deleted during the data cleaning process to filter out relevant context in natural language. This might reduce the amount of the dataset and perhaps the calculation time, as well as improve categorization using machine learning models. The Natural Language Toolkit in Python provides a list of stop words from over 16 languages that are taken into account when eliminating them from corpora.

(iv) URL removal: Regular expression in NLTK is used to delete links to posts and photographs that do not provide insight into the user's attitude.

(v) Emoji’s and extra words removal: The emoji’s used in different tweets were deleted from the tweets, as were unnecessary repeating terms found in the corpus, such as 'sometimes, whom, maybe, com..'

Uncleaned Sentences (Depressed Class)	Cleaned Sentences (Non-depressed Class)
['4 years ago I attempted suicide, was extremely depressed, doing a lot of drugs. After a lot of hard work and patience, I am clean, I love myself, I am the happiest I've ever been. Never lose hope & put in the work ❤️ #BellLetsTalk',]	['4 years ago attempted suicide extremely depressed lot drugs after lot hard work patience clean love happiest ever never lose hope put work bellletstalk',]
'When they say it gets better, they aren't lying. This time 3 years ago I attempted suicide bc I was so depressed and hopeless. Flash forward to today and I have a whole ass husband and two babies. Yes, I still struggle, but I'm also happier and more loved than I've ever been.'	'when better arent lying time 3 years ago attempted suicide depressed hopeless flash forward today whole ass husband two babies yes still struggle im also happier loved ever',
'You can dm us both if need be. I had a "friend" years ago try to fake an attempted suicide and it turned out just like this. All signs point to it\'s Chris. He\'s depressed and he wants reassurance without directly asking for it. He needs mental help.'	'you us need friend years ago try fake attempted suicide turned all signs point chris hes depressed wants reassurance without directly asking needs mental help'

Figure 15 Random Tweets

3.4.5 Information Retrieval

This section discusses how to convert tweets to vectors by combining Word2Vec and TF-IDF.

To begin the analysis of natural language the raw sentences must be converted into numerical representation to serve as inputs to the machine learning algorithms. Each word in the English language can be represented numerically in a particular way. Each word is assumed to be distributed in n-space. Dimensions or vector space, each dimension value has its own unique value, which gives weight to the meaning of the word in machine learning processing. One-hot encoding is one of the ways to convert phrases into vector form. However, this method is limited to giving the value of '1' if the word appears in the text, otherwise '0'. Other parameters such as word frequency, relevance of individual words, phrasal semantics and word-to-word similarity to other words in the corpus are not taken into account by this method.

(i). Bag of Words: One method for retrieving information and natural language processing methods (NLP) is to divide the phrase into words and remove stop words, punctuation, and

grammatical loads in sequence from one to z. In a dataset, BOW can be used to estimate the total number of words and their frequencies. In addition, BW assists in the execution of subject-dependent tasks, as well as utilizing a collection of very important words converted into vectors as inputs to the neural network learning approach for better data classification training.

(ii). TF-IDF: Term Frequency-Inverse Document Frequency (TF-IDF) is an acronym for Term Frequency-Inverse Document Frequency. The most important aim of this tool, as the name implies, is to assess the pertinence 107 of words in a document. It's a metric for determining the importance of a word inside a document. It is made up of two terms: term frequency, which indicates how many times a word appears in a phrase or document, and inverse document frequency, which indicates how many times the word appears in all documents. For example, if a word appears in all papers, its TF-IDF score will be 0, indicating that the term may be discarded and does not play a significant role in distinguishing documents.

The TF-IDF is calculated by multiplying TF by IDF.

Term frequency of a word = no of time word appear

Inverse frequency = $\log \frac{\text{Total no of documen}}{\text{Number of time word appear in the document}}$

(iii). Word2Vec Embeddings: Word2Vec collects input data using cascade flow to memorize word patterns and trains word embedding to restart the attenuation function using gradient decent. Word2Vec, on the other hand, calculates the damage function by averaging the ability to estimate the words around a word. For example, "Mop and hat are in the car," for example.

If the background reference window is three-dimensional and the target word is "cap", then to cover the situation "The," "Map," "and," "Array," "In," and "The." Can be used to refer to: a continuous bag of words, aimed at guessing and skipping the word hotspot from contextually adjacent words. Village technology, which serves as a useless education. Model for estimating words in context using word at convergence point.

In addition, the Gensim package was used to train the Word2Vec embedding used in this study. GENSIM makes it possible to train embedding using bespoke datasets instead of pre-built embedding.

The dataset for Word2vec embedding training is provided with the following settings:

(i) Size: It specifies the number of measurements that each word will display. In this study, it was set to 50.

(ii) Window: Specifies the number of words that occur between a given word and the words that surround it. For this study, the value was set to ten.

(iii) Minimum calculation: The term [102] specifies the number of times one should appear for training to take into account its embedding. In this study, the value is constant for one.

(iv) Threads: The total number of threads involved in the process. In the context of this study, the value was set to ten.

(iv). Matrix for Word2Vec Embeddings: Vector embedded data are trained using gensim. The word2vec embeddings are also computed using the same Tweet with 16 words. The whole Tweet is grouped together into a single bucket. Furthermore, as illustrated in Illustration, the weight of the bucket is determined by extracting vector embeddings for every phrase in the tweet.

```
S = csr_matrix(word_vector_embeddings[4])
B = S.todense()
print("word2vec embeddings: \n", B)
```

```
word2vec embeddings:
 [[ 5.9734664   0.6310016  -4.588813    1.651538    1.9868397  -1.7386596
  -4.120953    3.402071    0.5155075  -3.7933753  -6.065419    5.028276
  -2.981268   -4.295137   -1.0704504  -0.8749166  -3.256247    1.9120423
   0.7522198   3.6496274   0.6942794  -1.2072089  -3.4863591   0.04962726
   1.0564413   4.179343   -3.659727   -2.9100568  -1.8187703   0.97587985
  -0.01675744 -1.1130638  -0.09645098 -0.940086    1.9391406   4.739372
  -6.8861837   2.2638113  -1.9229968  -0.08238193 -2.855467   -2.555551
   0.25767124 -0.48622692 -2.2049894  -1.6723586  -1.7160431  -2.2270513
  -0.22170933  2.6303535 ]]
```

Figure 16 Word Matrix

(v). Computation of algorithms using metrics

The Python library allows you to use a variety of classification methods, whose performance and accuracy can be evaluated using metrics like the confusion matrix, accuracy, precision, recall(sensitivity), and specificity. The output of the confusion matrix is defined by four terms: True Positive, True Negative, False Positive, and False Negative.

The following are the definitions for these terms:

	Actual	Actual
Predicted	True Positive	False Positive
Predicted	False Negative	True Negative

True Positive (TP): The total number of positive class labels that have been successfully classified as positive (for example, the model guesses "yes" and the true value is "yes").

True Negative (TN): The total number of negative class labels successfully classified (for example, the model estimates "no" and the actual value is "no")

False Positive (FP): The total number of class labels rated as positive despite being actually negative (for example, the model was rated "yes" but the true value was "no").

False Negative (FN): The total number of class labels expected to be negative even though they are actually positive (for example, the model estimates "no" but the actual value is "yes").

Other metrics including accuracy, precision and recall are obtained using the True Positive, True Negative, False Positive and False Negative output fields, which are examined using the algorithm below.

MEASURE	DERIVATION
SENSITIVITY(RECALL)	TPR = TP / (TP + FN)
SPECIFICITY	TNR = TN / (FP + TN)
PRECISION	PPV = TP / (TP + FP)
NEGATIVE PREDICTIVE VALUE	NPV = TN / (TN + FN)
FALSE POSITIVE RATE	FPR = FP / (FP + TN)
FALSE DISCOVERY RATE	FDR = FP / (FP + TP)
FALSE NEGATIVE RATE	FNR = FN / (FN + TP)
ACCURACY	ACC = (TP + TN) / (TP + TN + FP + FN)
F1 SCORE	F1 = 2TP / (2TP + FP + FN)
MATTHEWS CORRELATION COEFFICIENT	MCC = TP*TN - FP*FN / SQRT((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Sensitivity: As True Positive Rate, it is also known as Recall value, which indicates the model's capacity to evaluate all potential compatible Instances.

Specificity: It's also known as the True Negative Rate, which represents the proportion of true negatives that are accurately evaluated.

Precision: It's also known as the Positive Predictive Value, and it demonstrates the model's capacity to create just the most likely meaningful events.

Negative Predictive Value: This is the ratio of false positives to the total value of true negative examples (FP + TN), which shows the accuracy of the probability of the expected negative result being true negative values to true negative values.

False Positive Rate: False positives are values that are actually negative but the expected result is positive.

False Negative Rate: This refers to the ratio of false negatives to the total value of true positive cases (FN + TP), where false negatives are values that are positive in real life but have negative predictive results..

False Discovery Rate: The ratio of false positives to the total value of all projected positive outcomes (FP + TP) demonstrates this.

Accuracy: When both prediction classes are of equal significance, an important rating is used. It's the proportion of properly recognized expected cases (TP + TN) to the total number of projected outcomes (TP + TN + FP + FN).

F1- Score: Precision and Recall are represented by the harmonic average. It’s used to find the offset between the Recall and Precision values and utilize weights to counterbalance the equation.

Matthews correlation coefficient: It aids in comprehending the relationship between actual values and projected results. Only when both the prediction outcomes for negative and positive events are appropriately identified with a high percentage creates a substantial value. The greater the MCC value, the higher the prediction model's quality.

Chapter-4 PERFORMANCE ANALYSIS

4.1 Quantitative Analysis

We mostly employed Accuracy(Acc), Precision(P), Recall(R), Fl score, and confusion matrix (CM) to statistically assess our models.

$$Acc = \frac{tp + tn}{tp + tn + fp + fn}$$
$$P = \frac{tp}{tp + fp}$$
$$R = \frac{tp}{tp + fn}$$
$$F1 = \frac{2 \times P \times R}{P + R}$$
$$CM = \begin{bmatrix} tn & fp \\ fn & tp \end{bmatrix}$$

The number of true positives, true negatives, false positives, and false negatives is represented as t p, tn, f p, and f n, respectively. We also assessed training and validation accuracy, as well as training and validation loss, for each of the models during the training process.

4.2 Evaluation of Logistic Regression

The generalized confusion matrix for logistic regression using Word2Vec for tweets is shown in the scenario. The decimal format makes sense by looking at the diagonal values, which exceed the level of the exact expected events and the diagonal values. Furthermore, the total value in each row is 1.00, indicating that the benefit of all predictable results for each category is 100%.

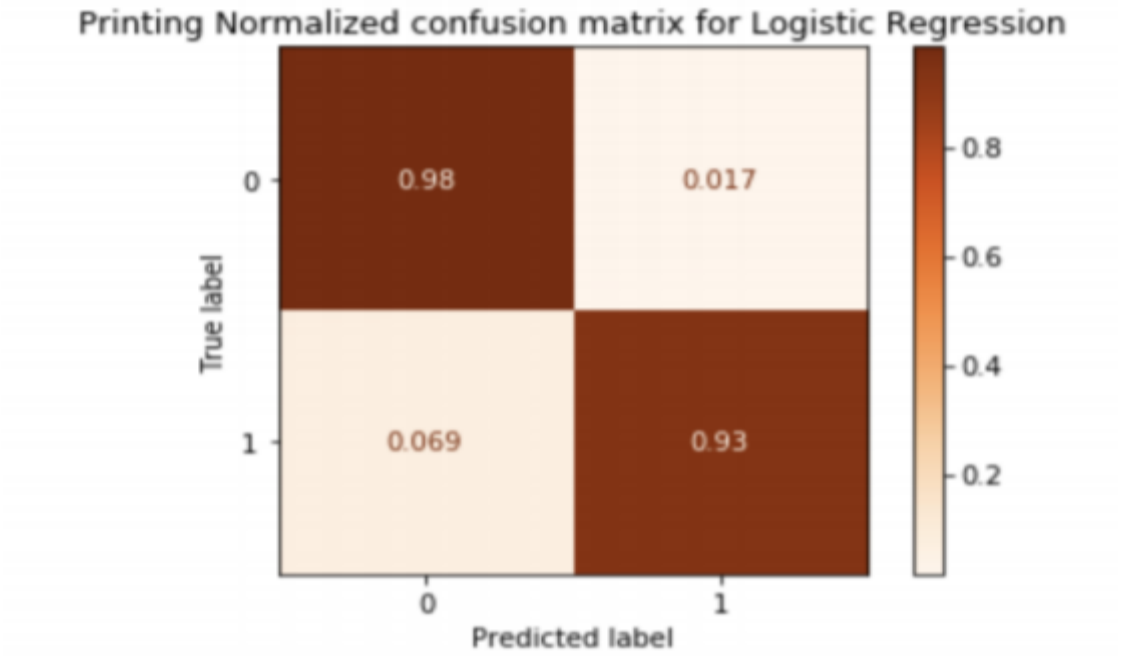


Figure 17 Confusion Matrix

The results are displayed in percentage style and include a variety of measures that make it easier to compare the performance of the model to that of other models.

Accuracy = 98.03% (represents the model's overall accuracy in prediction performance)

Recall (Sensitivity) = 94.15% (percentage of tweets accurately identified as being from the depressed class)

Specificity = 98.% (percentage of tweets accurately identified as belonging to the non-depressed class)

Precision = 94.34% (proportion of significant predicted outcomes out of all potential outcomes)

F1 score = 0.9304 (This figure is the harmonic average of recall and accuracy.)

Furthermore, a decimal number close to 1 denotes a perfect F1 score with balanced high accuracy and high recall, whereas a value of 0 denotes the worst F1 score with poor precision and low recall).

Measure	Value
Accuracy	97.03%
Recall(Sensitivity)	93.15%
Specificity	98.34%
Precision	94.94%
F1 Score	94.03%

4.3 Evaluation of Random Forest

Random Forest using Word2Vec +TF-IDF for Tweet Level. The decimal representation helps to understand the results by observation of the diagonal values showing the degree of accurately predicted occurrences, which are higher than the off diagonal values. Moreover, the sum of values of each row is equal to 1.00 that shows the 100% utility of possible outcomes for each class.

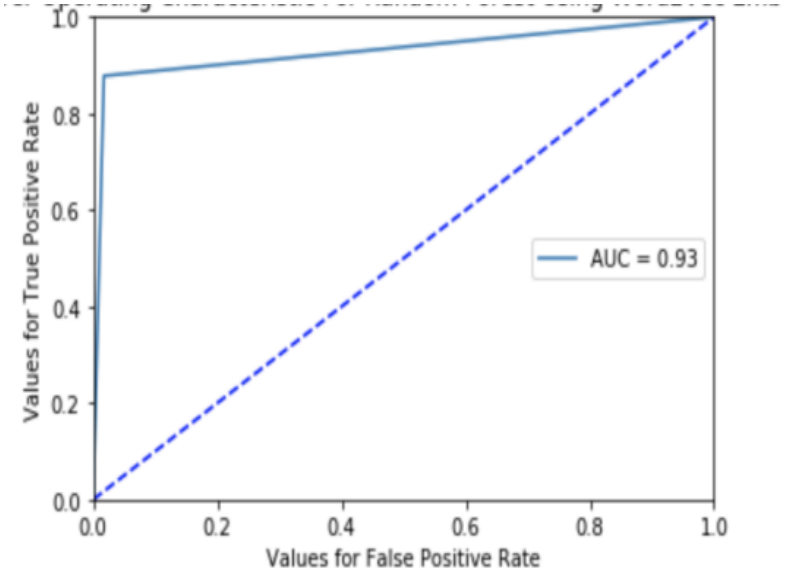


Figure 18 Accuracy Graph

The results are displayed in percentage style and include a variety of measures that make it easier to compare the performance of the model to that of other models.

Accuracy = 95.03% (represents the model's overall accuracy in prediction performance)

Recall (Sensitivity) = 87.23% (percentage of tweets accurately identified as being from the depressed class)

Specificity = 96.72% (percentage of tweets accurately identified as belonging to the non-depressed class)

Precision = 93.34% (proportion of significant predicted outcomes out of all potential outcomes)

F1 score in decimal value = 0.90 (This figure is the harmonic average of recall and accuracy.)

Measure	Value
Accuracy	95.65%
Recall (Sensitivity)	87.81%
Specificity	98.29%
Precision	94.49%
F1 Score	91.03%

4.4 Evaluation of Recurrent Neural Network (RNN)

In this we train.csv and dev.csv files were preprocessed such that the RNN model's input was a list of (tweet, label) pairs. Each tweet was represented as a word vector, with each word having a discrete ID that can be used to search up its Word2Vec word embedding. Each tweet was lengthened to a maximum of 30 characters. We input a batch of words that corresponded to the t^{th} word into the RNN for each timestep t for each of the current batch's tweets. The prediction was produced using the last hidden state, h, which was passed via another neural network layer, followed by the softmax function. The cross-entropy loss between the predicted and true labels was calculated.

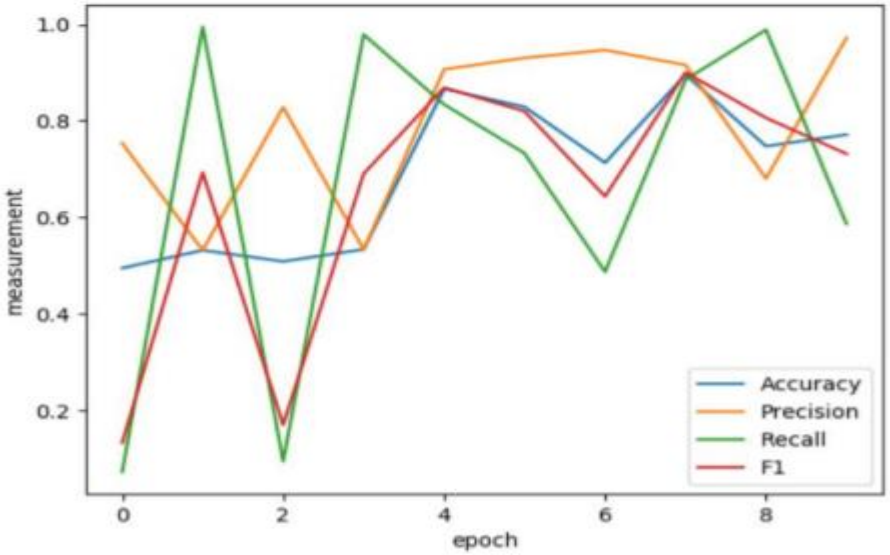


Figure 19 Epoch graph

The results are displayed in percentage style and include a variety of measures that make it easier to compare the performance of the model to that of other classification techniques.

Accuracy = 90.03% (represents the model's overall accuracy in prediction performance)

Recall (Sensitivity) = 89.23% (percentage of tweets accurately identified as being from the depressed class)

Specificity = 90.72% (percentage of tweets accurately identified as belonging to the non-depressed class)

Precision = 92.34% (proportion of significant predicted outcomes out of all potential outcomes)

F1 score in decimal value = 0.90 (This figure is the harmonic average of recall and accuracy.)

Furthermore, a decimal number close to 1 denotes a perfect F1 score with balanced high accuracy and high recall, whereas a value of 0 denotes the worst F1 score with poor precision and low recall).

Label	Acc	P	R	F1
Nondepressed	0.90	0.87	0.91	0.89
Depressed	0.90	0.92	0.89	0.90

4.5 Evaluation of CNN Long Short-Term Memory Network (CNN-LSTM)

In this we train.csv and dev.csv files are preprocessed in the same way as the word-based RNN. This CNN differs from previous RNN-based models in that it learns the embedding matrix W during training, whereas RNN-based models employed pre-trained embeddings.

We calculate the convolution layer outputs for each filter size and apply ReLU nonlinearity and max-pooling to these outputs. On the last layer, we merge all of these pooled features from each filter size to generate our feature vector. On this feature vector, we apply a dropout layer with a probability of 0.5 and quantify the loss using cross entropy.

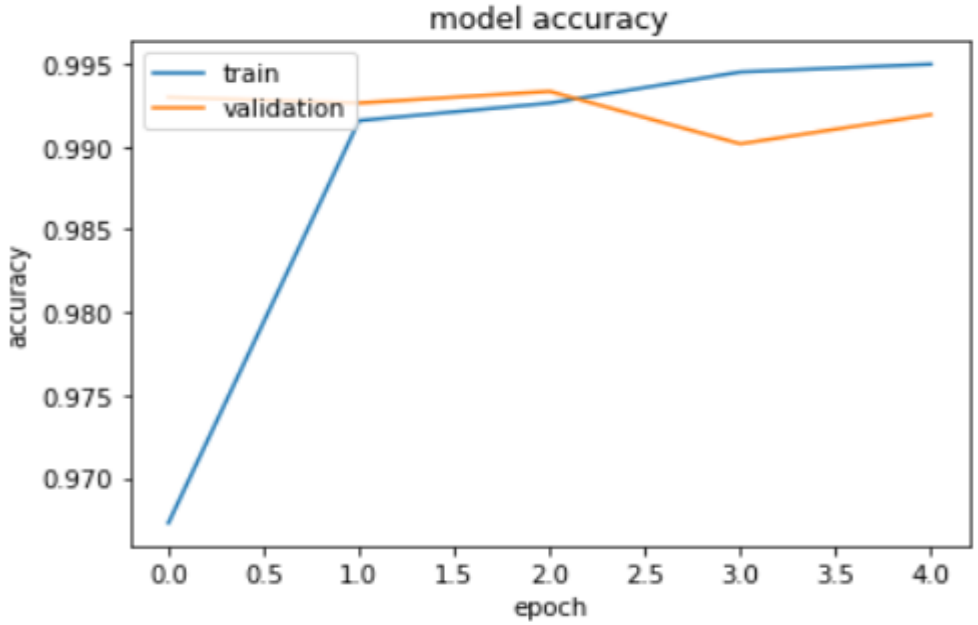


Figure 20 Model Accuracy

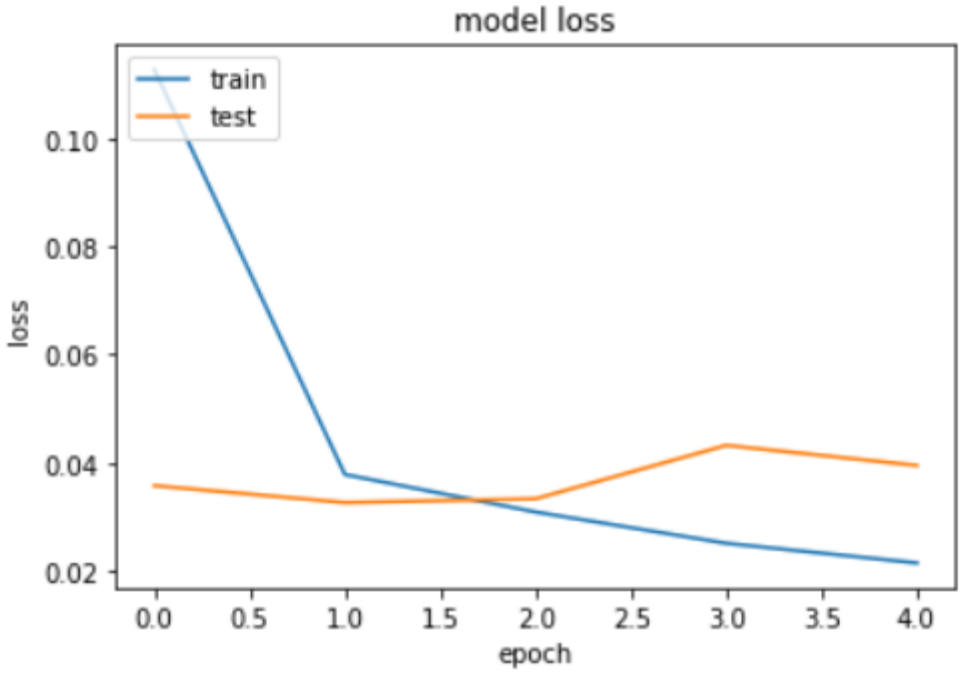


Figure 21 Model Loss

The results are displayed in percentage style and include a variety of measures that make it easier to compare the performance of the model to that of other models.

Accuracy = 98.17% (represents the model's overall accuracy in prediction performance)

Recall (Sensitivity) = 98.15% (percentage of tweets accurately identified as being from the depressed class)

Specificity = 97.34% (percentage of tweets accurately identified as belonging to the non-depressed class)

Precision = 99.02% (proportion of significant predicted outcomes out of all potential outcomes)

F1 score in decimal value = 0.99 (This figure is the harmonic average of recall and accuracy.)

Furthermore, a decimal number close to 1 denotes a perfect F1 score with balanced high accuracy and high recall, whereas a value of 0 denotes the worst F1 score with poor precision and low recall).

Label	Acc	P	R	F1
Nondepressed	0.97	0.97	0.98	0.98
Depressed	0.98	0.99	0.98	0.99

4.6 Analysis of RNN Base Models Vs CNN-LSTM Model

The performance of the word-based CNN-LSTM model may be shown in Figure. We can observe that the performance of the CNN-LSTM is equivalent to that of the top RNN-based models.

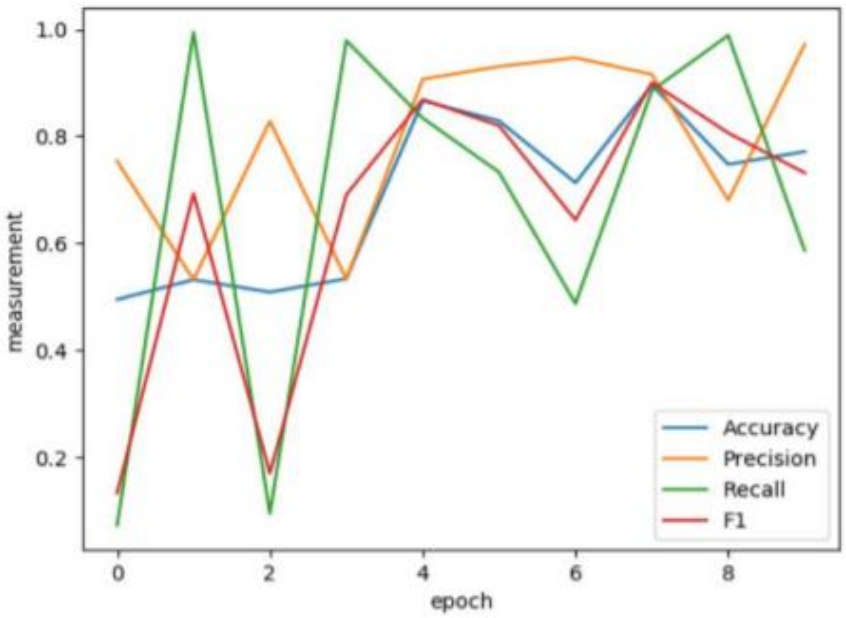


Figure 22 Model Loss2

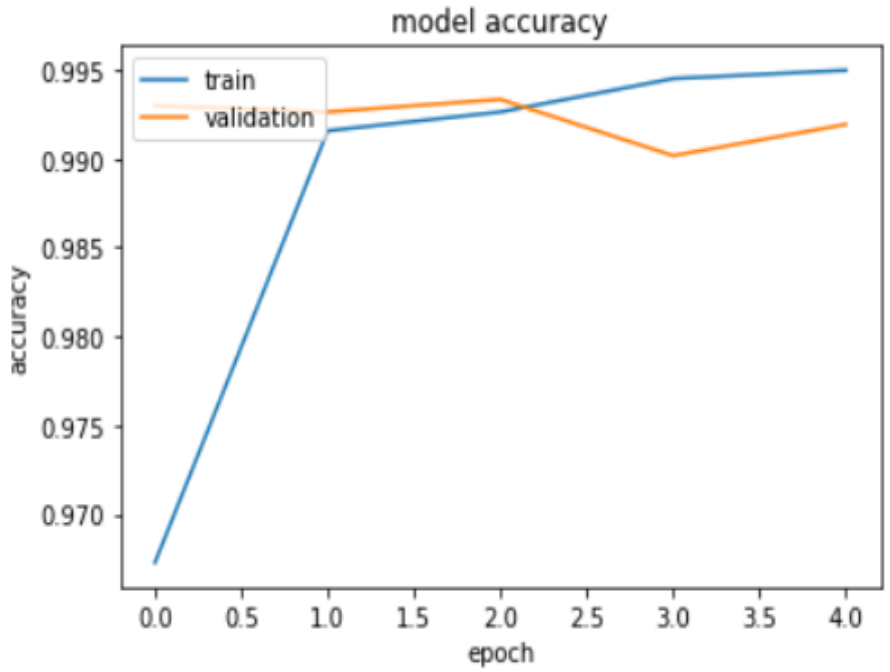


Figure 23 Model acccuracy2

Despite the fact that CNN-LSTM have typically been utilized for image identification, Despite the fact that the CNN does not save temporal information about the text use LSTM to provide memory to store the long information for short term, it achieves high accuracy on our dataset thanks to the convolutions done by variable convolutional filters, which allow the model to learn efficient and meaningful representations of the text. While our CNN model can catch up to 5 n-grams, other models are frequently unable to collect n-grams more than 3 without drastically reducing computing performance. The CNN can analyze the text more quicker because to these convolutions, whereas word-based RNN-based models must read the text word by word.

Chapter-5 CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

We investigate existing methods for identifying and predicting depression among Twitter users in this study. We've also analyzed existing techniques and presented two methods for identifying and forecasting depression in Twitter users. The first method is converting Twitter phrases into Word2Vec vectors and then applying several classification algorithms to them. CNN Long Short-Term Memory has the highest accuracy of 98.92%, precision of 99.02% and recall rate of 98.15%, which is higher than others when input with Word2Vec embedding. On the other hand, the RNN model gives 90.03% accuracy, 92.94% accuracy and 89.23% recall rate when input with Word2Vec embedding. The proposed solution model of this study at the user level can be used in various fields:

At the consumer level the proposed solution model of this study is used in various areas:

1.Big Data Analysis: Other researchers can use this research to assess frustration among users using a massive dataset. In addition, this work can be used to examine a number of issues under the broader feature engineering criteria in the approach suggested by other researchers, as well as patterns of participation of different individuals on social media (Twitter).

2. Mental Health Professionals: Professionals such as psychologists and psychiatrists can use the results of this study to examine their patients' social media usage (Twitter). In addition, this study will help in the classification of each patient based on the strict feature engineering criteria that apply. For his social media posts (tweets). Finally, to assist in assessing depression in patients and to understand the patterns of patients who are unable to express their thoughts and feelings during the interview session. In addition, by evaluating users' social media (Twitter) actions in a previous step, this research can be used to identify frustration in users who know that severe depression-related effects (such as suicide) can be avoided. Proper medical care can be provided.

5.2 FUTURE WORK

The ending of this study will outline some of the future work methodologies and concepts that will be created in response to the various datasets, such the combination of a more complicated hybrid architecture or an AI-operated multimodal feature. structure of engineering One of the most important considerations for appropriate treatment is the diagnosis of depression. People's mental health has been aided by social media's transformation into a tool.to be able to recognize depression This study may be expanded by include numerous other variables. Other than textual information, such as the user's profile picture and sentiments If the post involves photographs or videos, it will be shared. In addition, there are several social media networks. Instagram, like Facebook, may be used to generate a user's history and information. Investigate the user's conduct across multiple social media sites as well. This study effort exceeds several existing techniques in terms of machine learning, however it can still be improved by using rigorous ensemble machine learning. strategies that increase the accuracy of other classification models by building on top of them. As a result, a research of several factors that accelerate depression is being conducted based on the findings. The user's geographic location may be determined, and relevant assistance can be supplied. the patient who has been afflicted Nonetheless, this research might be used to anticipate other events.by changing some of the data gathering methodologies in terms of mental health illnesses intended result Schizophrenia, anxiety, PTSD, and other mental illnesses, for example, can be anticipated. altering the phrases and keywords in data extraction tools in accordance with. The signs of a mental health condition will be investigated. This research was used to study the differences between user models with different mental health disorders, by collecting data in different datasets for each mental health disorder and sending those datasets one by one through the implementation process described below for the proposed solution. Can also. Each dataset gives two different outputs, each of which can be further studied. In addition, the study modifies the database for future use by connecting it to an automated alert messaging system and selecting the output as the target result from the taxonomy model, which directly alerts frustrated users. Are in the early stages. Assessed by their social media activities. Additionally, by inserting a user profile link on the website page, a dynamic website can be developed that can interact with the approach in this research to investigate and assess the level of depression. The interactive website might potentially be used to raise awareness about a variety of mental illnesses. By changing the pre-processing

approach of the input dataset, the output can be adjusted to assess various mental health issues such as stress, anxiety and mental disorders. This leads to enhanced additional aspects of the website, such as the ability to register any user account and input their social media details to receive information about their mental health based on their social media activity patterns.

REFRENCE

1. Britz D. (2015) Implementing a CNN for Text Classification in Tensorflow. WILDML. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/

2. Kim Yoon (2014) Convolutional Neural Networks for Sentence Classification. https://arxiv.org/pdf/1408.5882.pdf

3. Minimax Char embeddings GitHub. https://github.com/minimaxir/char-embeddings

4. Santos C.N. & Gatti M. (2014) Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 6978, Dublin, Ireland, August 23-29 2014. http://www.aclweb.org/anthology/C14-1()()8.

5. Twitterscraper library. https://github.com/taspinar/twitterscraper

6. Vosoughi S. & Vijayaraghavan P. & Roy D. (2016) Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder. ACM. https://arxiv.org/pdf/1607.07514.pdf.