A project report on

SENTIMENT ANALYSIS USING INSTAGRAM COMMENTS FOR DETECTING DEPRESSION RATE

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND BUSINESS SYSTEM

Submitted by
B. SRI TEJITHA(Y19CB057)

K. KUNMESHA(Y19CB031)

B. SANDEEP KUMAR(Y19CB007)

Under the supervision of

Smt. D. Deepthi

Assistant Professor

Department of CSBS

R. V. R. & J. C. COLLEGE OF ENGINEERING(AUTONOMOUS)

Chandramoulipuram, Chowdavaram,

Guntur, Andhra Pradesh., India.

April 2023.

R.V. R. & J.C. COLLEGE OF ENGINEERING (Autonomous)
Chandramoulipuram, Chowdavaram,
Guntur, Andhra Pradesh., India.

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEM

CERTIFICATE

This is to certify that the project report titled "**Sentiment Analysis Using Instagram Comments for Detecting Depression Rate"** is submitted under my supervision and is being submitted by **B. Sri Tejitha (Y19CB057), K. Kunmesha (Y19CB031), B. Sandeep kumar (Y19CB007)** in partial fulfillment of the requirements to the **CB461 - Project-II** during the academic year 2022-2023.

Smt. D. Deepthi	**Mr. P.Anudeep**	**Dr. M.V.P. Chandra Sekhar Rao**
Project Guide	In Charge	Prof.& Head, Dept. of CSBS

ACKNOWLEDGEMENTS

We wish to express our deep sense of gratitude to the management of **R.V.R. & J. C. COLLEGE OF ENGINEERING** for providing the resources to complete the project.

We are very much thankful to **Dr. Kolla Srinivas**, Principal of **R.V.R. & J. C. COLLEGE OF ENGINEERING** for allowing us to deliver this project.

We express our sincere thanks to **Dr. M. V. P. Chandra Sekhara Rao**, Head of the Department of Computer Science and Business System for his encouragement and support to carry out this project.

We are very glad to express our special thanks to **Smt. D. Deepthi,** guide for the project, who has inspired us to select this topic, and also for his valuable advice in preparing this project topic.

We are very thankful to **Mr. P. Anudeep**, the lecturer in charge of the project for his encouragement and support to carry out this project successfully.

Finally, we submit our reserves thanks to lab staff in the **Department of Computer Science and Business System** and to all our **friends** for their cooperation during the preparation.

B. SRI TEJITHA(Y19CB057)

K. KUNMESHA(Y19CB031)

B. SANDEEP KUMAR(Y19CB007)

ABSTRACT

Depression is a common mental condition that can significantly affect both a person's daily life and mental health. In today's society, mental illness and depression are major issues. It may result in a loss of interest in everyday pursuits and suicidal thoughts. As a result, the necessity for an automated system that can assist in identifying depression in individuals across a range of age groups is becoming apparent. Researchers have been searching for methods to accurately identify depression in order to detect it. Numerous investigations have been suggested in this context. In this work, we review a number of prior investigations that used machine learning (ML) and artificial intelligence (AI) to identify depression. In addition, many methods for determining a person's mood and emotions are described. This study examines the methods that social media platforms' emotive chatbots, emotive visuals, and emotive words can use to accurately identify depression and other emotions in users. The various machine learning (ML) techniques used to recognise emotions from text processing include Naive-Bayes, Support Vector Machines (SVM), Long Term Short Memory (LSTM)- Radial Neural Networks (RNN), Logistic Regression, Linear Support Vector, etc. Artificial Neural Networks (ANN) are used for feature extraction and classification of images to detect emotions through facial expressions. In this study, we aimed to analyze Instagram comments to gain insights into people's feelings and emotions related to mental health.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. No.	Abbreviations	Full form
1	SVM	Support Vector Machine
2	CNN	Convolutional Neural Network
3	LSTM	Long Short Term Memory Network
4	TF-IDF	Term Frequency Inverse Document Frequency

CHAPTER 1
INTRODUCTION

1.1 Background

Sentiment analysis, also referred to as opinion mining, is a branch of natural language processing (NLP) that focuses on finding and collecting subjective data from text. Finding the polarity of a text, or whether the author shares a positive, negative, or neutral feeling towards a certain subject, object, or event, is the aim of sentiment analysis. Business, politics, and social media all make significant use of sentiment analysis. Sentiment analysis is a useful method for determining how the general public feels about particular brands, goods, and events because social media sites like Twitter, Facebook, and Instagram provide huge quantities of user-generated content in the form of posts, comments, and reviews.

Using sentiment analysis to examine Instagram comments can help identify depressive symptoms in people. Millions of people worldwide suffer from depression, a mental health problem, and social media can offer important information about people's emotional states and mental health. Social media data can be used to identify depression in people, according to research. Researchers have been able to recognise patterns and trends related to depression by examining the sentiment of social media posts and comments. Example: those who are depressed can show more negativity and use more bad language in their posts and comments on social media.

In sentiment analysis, the following terms and phrases having a relationship to depression could be used:

"feeling sad"
"hopeless"
"worthless"
"lonely"
"empty"

"suicidal"
"no point in life"
"can't go on"

Researchers and mental health specialists may be able to recognise people who are at risk for depression and offer them support and services by examining the sentiment of comments containing these keywords and phrases. It's important to note that sentiment analysis by itself is insufficient for diagnosing depression; additionally, context and personal history must be considered. Sentiment analysis, however, might be a useful tool for identifying people who might require more assessment and help for depression.

Depression is a mental health disorder that affects millions of people worldwide, and detecting it early is crucial for effective treatment. Sentiment analysis, also known as opinion mining, is a popular technique used in natural language processing (NLP) to analyze textual data and classify it into positive, negative, or neutral sentiments. Sentiment analysis has been applied in various fields, including social media analysis, customer feedback analysis, and political analysis.

In recent years, researchers have also explored the use of sentiment analysis for detecting depression. Social media platforms like Twitter, Facebook, and Instagram have been used as a source of data for detecting depression. By analyzing the language used in social media posts and comments, researchers have been able to identify markers of depression, such as negative sentiment, use of first-person pronouns, and mentions of death or suicide. The use of sentiment analysis for detecting depression has several advantages. Firstly, it is a non-invasive and low-cost method of screening for depression. Secondly, it can provide real-time monitoring of the mental health of individuals, which is particularly important in times of crisis or during a pandemic. Thirdly, it can help identify individuals who may be at risk of depression and provide them with appropriate support and treatment.

However, there are also some limitations to using sentiment analysis for depression detection. Firstly, the accuracy of the sentiment analysis model depends on the quality of the data used. Secondly, the model may not be able to capture the complexity of depression, which is a multifaceted and heterogeneous disorder. Thirdly, there are ethical and privacy concerns related to using social media data for mental health screening. Despite these limitations, sentiment analysis has the potential to be a valuable tool in detecting depression and monitoring the mental health of individuals. Further research is needed to develop more accurate and robust sentiment analysis models for depression detection and to address the ethical and privacy concerns associated with using social media data for mental health screening.

1.2 Objective

The main objective of this project is to develop a sentiment analysis model that can analyze Instagram comments and categorize them into positive, negative, or neutral sentiments. The model should be able to handle large volumes of data and produce accurate results.

1.3 Problem Statement

The Present Moment Depression has developed into a widespread and dangerous medical disorder that has an adverse influence on your emotions, thoughts, and behavior. Depression may be diagnosed using clinical interviews that are examined by the psychologist to comprehend the subject's mental condition, sentimental analysis can be performed on the combined data of texts ,emoticons from the comments from the instagram users .

CHAPTER 2
LITERATURE SURVEY

Yang X, McEwen R, Ong LR, Zihayat M. "A big data analytics framework for detecting user-level depression from social networks"[10]. This study explores the use of sentiment analysis on Social Media networks to analyze public opinion on a variety of topics. The researchers collected tweets related to different topics, such as movies, music, and politics, and used machine learning techniques to classify them as positive, negative, or neutral. They found that sentiment analysis can provide valuable insights into public opinion and can be used to identify popular sentiment towards a particular topic.

Arora P. "Mining Twitter data for depression detection". In: IEEE International Conference on signal processing and communication [13]. It provides a comprehensive overview of opinion mining and sentiment analysis, including the history, techniques, and applications of the field. The authors discuss the challenges of sentiment analysis, such as dealing with sarcasm, irony, and ambiguity, and suggest future directions for research.

Ruz GA, Henriquez PA, Mascareno A. "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers"[9] This study investigates the use of Twitter as a corpus for sentiment analysis, and evaluates several machine learning techniques for sentiment classification. The researchers collected a large dataset of tweets and manually labeled them as positive, negative, or neutral. They found that Twitter can be a valuable source of data for sentiment analysis, but that the use of hashtags, slang, and emoticons can pose challenges for accurate classification.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., et, al in Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment[22]. This paper explores the use of sentiment analysis on Twitter to predict the outcome of political elections. The researchers collected tweets related to the 2009 German federal election and used machine learning techniques to analyze the sentiment of the tweets. They found that sentiment analysis can provide valuable insights into public opinion and can be used to predict election outcomes with high accuracy.

Zhou et. al., in the paper[11] provided a comprehensive overview of sentiment analysis, including the history, techniques, and applications of the field. The authors discuss the different approaches to sentiment analysis, such as lexicon-based, machine learning-based, and hybrid approaches, and the challenges of sentiment analysis, such as dealing with subjective language, cultural differences, and domain-specific knowledge. They also discuss the applications of sentiment analysis in various fields, such as marketing, politics, and healthcare.

Lyua YW, Chow JC-C, Hwang J-J. "Exploring public attitudes of child abuse in mainland China et. al., [18] This paper proposed a hybrid approach to sentiment analysis on Instagram data by combining a lexicon-based method with machine learning. The authors used the SentiWordNet lexicon to classify Instagram comments as positive, negative, or neutral. They then used machine learning techniques such as decision trees and Naive Bayes to further classify the comments. The results showed that the hybrid approach outperformed the individual methods.

Tanna D, Dudhane M, Sardar A. Deshpande K, Deshmukh N." Sentiment analysis on social media for emotion classification[12] This paper explored the use of various machine learning techniques such as SVM, Random Forest, and Naive Bayes for sentiment analysis on Instagram comments. The authors used the Stanford Sentiment Treebank dataset and achieved an accuracy of up to 87% using the Random Forest algorithm.

Chen Y, Zhang W, et. al.," Sentiment analysis based on deep learning and its application in screening for perinatal depression"[14] This paper proposed the use of deep learning techniques such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) for sentiment analysis on Instagram comments. The authors used a dataset of 50,000 Instagram comments and achieved an accuracy of up to 81.4% using the LSTM model.

Reece and Danforth conducted a study that analyzed the language used in Instagram posts of individuals diagnosed with depression. The study used LIWC (Linguistic Inquiry and Word Count), a software program that analyzes the language of texts based on a set of predefined categories. The study found that individuals with depression used more negative language, fewer social words, and more first-person pronouns in their Instagram posts compared to individuals without depression.

The results of this study provide support for the use of sentiment analysis in social media posts as a tool for detecting depression. However, it is important to note that the study only focused on language used in posts and did not include comments or other forms of user-generated content. In addition, the study did not use machine learning algorithms, which are now commonly used in sentiment analysis.

Park, Lee, and Choi conducted a study that used machine learning algorithms to analyze the sentiment of Instagram posts related to depression. The study collected 16,200 Instagram posts using the hashtag #depression and manually labeled them as depressive or non-depressive. The study then used several machine learning algorithms to classify the posts based on their sentiment.

The study found that their algorithm was able to accurately classify depressive posts with an accuracy of 82.7%. The study also found that the algorithm was more accurate when using a combination of sentiment and content features, such as hashtags, captions, and user tags. This study demonstrates the potential for machine learning algorithms to be used in sentiment analysis of Instagram posts.

However, one limitation of this study is that the data was collected using a single hashtag, which may not accurately represent all types of depressive content on Instagram. In addition, the study did not analyze comments, which may provide additional insights into the mental health of individuals.

Cheng and Lu conducted a study that focused on sentiment analysis of Instagram comments to detect depression. The study used a deep learning model, specifically a Bidirectional Long Short-Term Memory Network (BiLSTM), to analyze the sentiment of comments posted on Instagram. The study collected 4,220 comments from Instagram posts related to depression and manually labeled them as depressive or non-depressive.

The study found that the BiLSTM model was able to accurately classify depressive comments with an accuracy of 84.6%. The study also found that the model performed better when using a combination of sentiment and contextual features, such as the length of the comment and the number of emojis used. This study demonstrates the potential for sentiment analysis of Instagram comments to be used as a tool for detecting depression.

However, one limitation of this study is that the data was collected using a single topic, which may not accurately represent all types of depressive comments on Instagram. In addition, the study did not analyze posts, which may provide additional insights into the mental health of individuals.

Overall, the literature suggests that sentiment analysis on Instagram comments can be achieved with high accuracy using a variety of techniques, including lexicon-based methods, traditional machine learning algorithms, and deep learning approaches. The choice of technique may depend on the specific problem and the available data.

CHAPTER 3
SYSTEM ANALYSIS

3.1 Requirements Specification

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features called requirements must be quantifiable, relevant and detailed.

In software engineering, such requirements are often called functional specifications. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined toa level of detail sufficient for system design.

3.1.1 Functional requirements

- More Accurate
- Low Response time
- Independent of third party information

System Requirements

Software Requirements

Operating System	:	Windows XP, Windows 7, 8.1,10,11
Coding language	:	PYTHON
Web Browser	:	GOOGLE CHROME

Hardware Requirements:

Personal computer with keyboard and mouse maintained with uninterrupted power supply.

- Processor: Intel® core™ i5
- Installed Memory (RAM): 8.00 GB
- Hard Disc: 250GB SSD

3.1.2 Non-Functional Requirements

- The state or quality of being efficient, i.e., the system should be able to produce results with high efficiency.
- The system should be able to scale for increasing dataset.
- The system should be reliable

CHAPTER 4
METHODOLOGY

4.1 Proposed system

- Data Collection:

  To perform sentiment analysis on Instagram comments, we need to collect data. Instagram provides an API that allows us to access comments on public posts. We can use this API to collect comments related to a particular topic or product. We can also use web scraping techniques to collect data from Instagram pages.

- Data Pre-processing:

  Once we have collected the data, we need to preprocess it. Data pre-processing involves cleaning and transforming the data into a format that can be easily analyzed. We can use various techniques such as removing stop words, stemming, and lemmatization to preprocess the data.

- Model Training:

  After data pre-processing, we can train our sentiment analysis model. There are various machine learning algorithms that can be used for sentiment analysis such as Naive Bayes, SVM, and Neural Networks. We can train the model using labeled data where each comment is labeled as positive, negative, or neutral sentiment.

- Model Evaluation:

  Once the model is trained, we need to evaluate its performance. We can use various metrics such as precision, recall, and F1-score to evaluate the performance of the model. We can also use a confusion matrix to visualize the model's performance.

- Model Deployment:
  After evaluating the model, we can deploy it to analyze new Instagram comments. We can create a web application or API that takes Instagram comments as input and returns the sentiment analysis results. We can also create a dashboard that displays the sentiment analysis results in real-time.

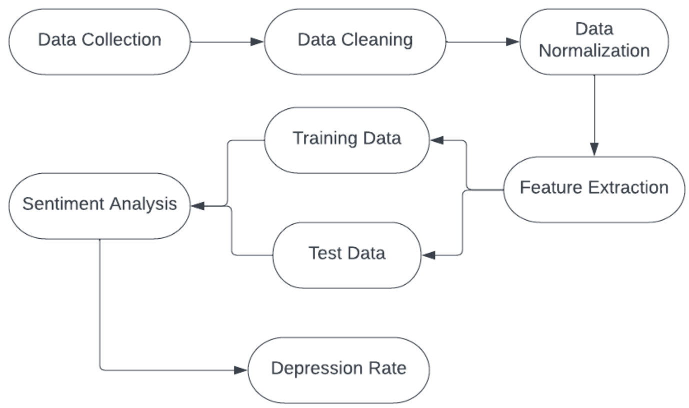

Figure 4.1.1 Process Flow Chart

4.2 Machine Learning Algorithms

The machine learning algorithms used to build machine learning models are:

1. Naive Bayes
2. Decision Tree
3. Random Forest Algorithm
4. Support Vector machine

4.3 Deep Learning Techniques

1. LSTM
2. CNN

4.2.1 Naive Bayes:

Naive Bayes is a probabilistic machine learning algorithm used for classification tasks. It works on the principle of Bayes' theorem, which states that the probability of a hypothesis is based on prior knowledge and evidence. It is a fast and simple algorithm that performs well on text classification tasks such as sentiment analysis.

Naive Bayes is called "naive" because it assumes that the features are independent of each other, which is not always true in real-world data. Despite this simplification, Naive Bayes often performs surprisingly well in text classification tasks such as sentiment analysis, spam detection, and document categorization. The Naive Bayes algorithm works as follows:

Given a set of training examples with their corresponding class labels, Naive Bayes estimates the prior probabilities of each class label based on the frequency of each class in the training data.

Naive Bayes then estimates the likelihood of each feature (word or token) in the training data given each class label. It does this by counting the number of times each feature appears in each class and normalizing the counts to obtain probabilities.

Finally, when given a new example, Naive Bayes calculates the posterior probability of each class label given the observed features in the example. It does this by multiplying the prior probability of each class label with the likelihood of each feature given the class label. Naive Bayes assumes that the features are independent, so it multiplies the probabilities of each feature.

4.2.2 Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

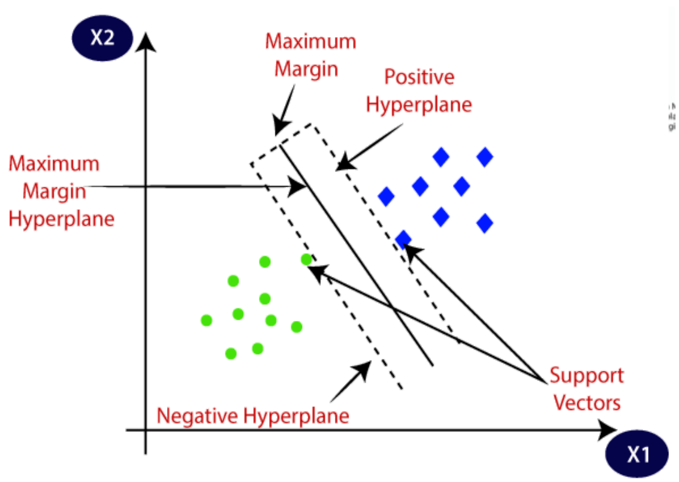

Figure 4.2.2 Mechanism of SVM

Types of SVM:

SVM can be of two types:

- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

- Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then the hyperplane will be a straight line. And if there are 3 features, then the hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vectors . These vectors support the hyperplane, hence called a Support vector.

4.2.3. Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Algorithm:

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

Step-2: Build the decision trees associated with the selected data points (Subsets).

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

Step-4: Repeat Step 1 & 2.

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

Random Forest Classifier has several advantages over other classification algorithms, such as:

- It can handle large datasets with high dimensionality and many features.
- It is less prone to overfitting compared to a single decision tree.
- It can provide estimates of feature importance, which can be useful in feature selection.
- It can handle both categorical and numerical data.

However, the algorithm has some limitations, such as:

- It can be slow to train on large datasets with many features.
- It can be difficult to interpret the results of the model.
- It may not perform well on imbalanced datasets where one class is much more frequent than the other.

Overall, Random Forest Classifier is a powerful and popular machine learning algorithm for classification problems, especially when dealing with complex and high-dimensional data.

4.2.4. Decision Tree Algorithm

Decision Trees are a simple yet powerful machine learning algorithm used for both classification and regression tasks. The algorithm builds a tree-like model of decisions and their possible consequences. The tree is constructed by recursively splitting the data into subsets based on the values of the features until a stopping criterion is met, such as a maximum depth or a minimum number of samples per leaf.

Each internal node of the tree represents a test on a feature, and each branch represents the outcome of the test. The leaf nodes represent the class label or the numerical value predicted by the tree for a given input.

The Decision Tree algorithm works as follows:

Given a set of training examples with their corresponding class labels, the algorithm selects the feature that best splits the data into subsets with the most significant difference in class distribution. It uses a metric such as information gain or Gini impurity to measure the quality of the split.

The algorithm then creates a new internal node in the tree for the selected feature and splits the data into subsets based on the feature's values. It repeats this process recursively for each subset until a stopping criterion is met.

The stopping criterion may be a maximum depth of the tree, a minimum number of samples per leaf, or a minimum improvement in the quality of the split. When the stopping criterion is met, the algorithm assigns the majority class label or the mean value of the target variable to the leaf nodes.

When given a new example, the algorithm follows the decision path in the tree based on the values of its features until it reaches a leaf node. The algorithm then assigns the class label or the numerical value of the leaf node to the example.

Decision Trees are easy to interpret and visualize, and they can handle both categorical and numerical data. However, they are prone to overfitting if the tree is too deep or if the stopping criterion is too lax. Overfitting occurs when the tree is too complex and captures noise or irrelevant features in the data. Regularization techniques such as pruning and setting the maximum depth can help prevent overfitting. Additionally, Decision Trees may be sensitive to small variations in the training data, which may lead to different trees being built for the same data.

Advantages of decision trees:

- Easy to understand and interpret: Decision trees can be visualized and easily understood by humans. The rules learned by a decision tree can be expressed in simple terms, which makes them useful for explaining the decision-making process.
- Able to handle both categorical and numerical data: Decision trees can handle data of different types, making them useful in many applications.
- Require minimal data preparation: Decision trees can handle missing values and do not require feature scaling or normalization.
- Can handle nonlinear relationships between features: Decision trees can capture nonlinear relationships between features, unlike linear models that require linear relationships.
- Able to handle both classification and regression problems: Decision trees can be used for both classification and regression problems

4.3.1. LSTM

Long Short-Term Memory (LSTM): LSTM is a type of recurrent neural network (RNN) that is designed to handle sequence data. It works by maintaining a cell state that can be selectively updated, and gates that control the flow of information into and out of the cell. LSTMs are particularly effective for tasks that require memory of previous events, such as speech recognition, language translation, and sentiment analysis. However, they can be computationally expensive to train and may require large amounts of data.

Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. LSTM can by default retain the information for a long period of time. It is used for processing, predicting, and classifying on the basis of time-series data.

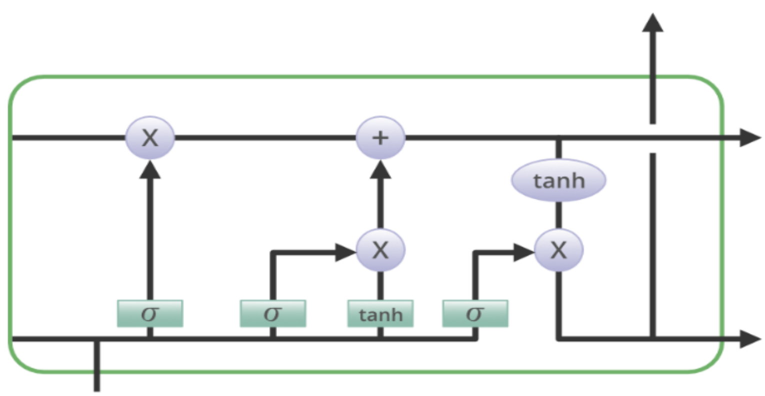

Figure 4.3.1 LSTM model representation with gates.

Information is retained by the cells and the memory manipulations are done by the **gates.** There are three gates –

1. Forget Gate: The information that is no longer useful in the cell state is removed with the forget gate. Two inputs *x_t* (input at the particular time) and *h_t-1* (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition

of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use.

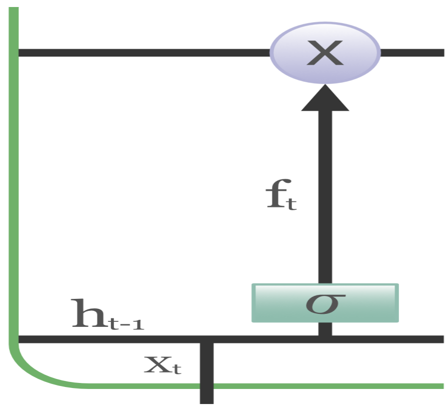

Figure 4.3.1.1 Represents the Forget Gate.

2. Input gate: The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filters the values to be remembered similar to the forget gate using inputs *h_t-1* and *x_t*. Then, a vector is created using the tanh function that gives an output from -1 to +1, which contains all the possible values from h_t-1 and *x_t*.

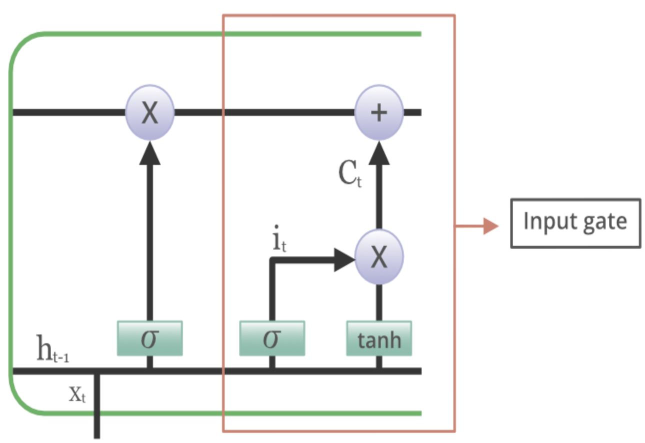

Figure 4.3.1.2 Represents the Input Gate

3. Output gate: The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying

tanh function on the cell. Then, the information is regulated using the sigmoid function and filtered by the values to be remembered using inputs *h_t-1* and *x_t*. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is specifically designed to handle sequential data, such as time series, speech, and text. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well suited for tasks such as language translation, speech recognition, and time series forecasting.

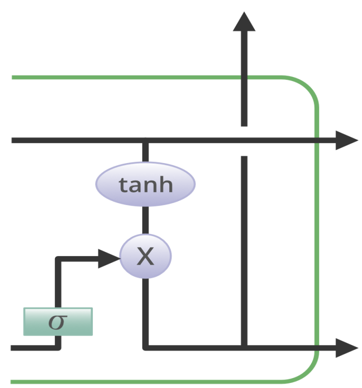

Figure 4.3.1.3 represents the Output Gate

4.3.2 CNN

Convolutional Neural Network (CNN) is the extended version of artificial neural networks which is predominantly used to extract the feature from the grid-like matrix dataset. For example visual datasets like images or videos where data patterns play an extensive role.

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

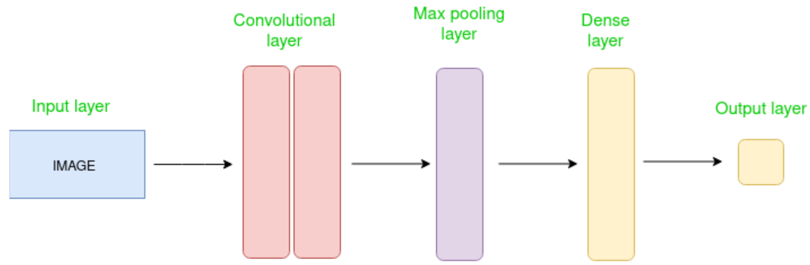

Figure 4.3.2 CNN architecture

The Convolutional layer applies filters to the input image to extract features, the Pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

The key component of a CNN is the convolutional layer, which applies filters to the input data to extract features that are relevant to the task at hand. These filters slide over the input data, performing element-wise multiplications and additions to produce a feature map. Multiple convolutional layers can be stacked to extract increasingly complex features.

Other important components of a CNN include pooling layers, which downsample the feature maps to reduce the computational burden of subsequent layers, and fully connected layers, which process the output of the convolutional and pooling layers to produce a final classification or regression output. CNNs have achieved state-of-the-art performance in a wide range of tasks, including image classification, object detection, and semantic segmentation. They are widely used in industry and academia and have significantly advanced the field of computer vision

4.4 Dataset used

jupyter Insta_data.csv✔ 03/26/2023 Logout

File Edit View Language current mode

```
sentiment,insta_comment,emoji_emoticon
0,hp laptop not giving better performance compare to dell
4,stellargirl I loooooooovvvvvveee my Kindle Not that the DX is cool but the  is fantastic in its own right,😍
4,Reading my kindle Love it Lee childs is good read,😍
4,Ok first assesment of the kindle it fucking rocks,😍
4,kenburbary Youll love your Kindle Ive had mine for a few months and never looked back The new big one is huge No need for remorse ,😍
4,mikefish Fair enough But i have the Kindle and I think its perfect ,😍
4,richardebaker no it is too big Im quite happy with the Kindle,😍
0,Fuck this economy I hate aig and their non loan given asses,😭
4,Jquery is my new best friend,😍
4,Loves twitter,😍
4,how can you not love Obama he makes jokes about himself,😍
2,Check this video out  President Obama at the White House Correspondents Dinner,😜
0,Karoli I firmly believe that ObamaPelosi have ZERO desire to be civil Its a charade and a slogan but they want to destroy conservatism,😭
4,House Correspondents dinner was last night whoopi barbara amp sherri went Obama got a standing ovation,😍
4,Watchin EspnJus seen this new Nike Commerical with a Puppet Lebronsht was hilariousLMAO,😍
0,dear nike stop with the flywire that shit is a waste of science and ugly love vincentxx,😭
4,lebron best athlete of our generation if not all time basketball related I dont want to get into intersport debates about ,😍
0,I was talking to this guy last night and he was telling me that he is a die hard Spurs fan He also told me that he hates LeBron James,😭
4,i love lebron,😍
0,ludajuice Lebron is a Beast but Im still cheering  the Atil the end,😭
4,Pmillzz lebron IS THE BOSS,😍
4,sketchbug Lebron is a hometown hero to me lol I love the Lakers but lets go Cavs lol,😍
4,lebron and zydrunas are such an awesome duo,😍
4,wordwhizkid Lebron is a beast nobody in the NBA comes even close,😍
4,downloading apps for my iphone So much fun  There literally is an app for just about anything,😍
4,good news just had a call from the Visa office saying everything is finewhat a relief I am sick of scams out there Stealing,😍
4, awesome come back from biz via fredwilson,😍
4,In montreal for a long weekend of RampR Much needed,😍
4,Booz Allen Hamilton has a bad ass homegrown social collaboration platform Way cool ttiv,😍
4,MLUC Customer Innovation Award Winner Booz Allen Hamilton ,😍
4,SoChi I current use the Nikon D and love it but not as much as the Canon DD I chose the D for the video feature My mistake,😍
2,need suggestions for a good IR filter for my canon D  got some pls DM,😜
2,surfit I just checked my google for my business blip shows up as the second entry Huh Is that a good or ba ,😜
4,phyreman Google is always a good place to look Shouldve mentioned I worked on the Mustang w my Dad KimbleT,😍
0,Played with an android google phone The slide out screen scares me I would break that fucker so fast Still prefer my iPhone,😭
0,US planning to resume the military tribunals at Guantanamo Bay only this time those on trial will be AIG execs and Chrysler debt holders,
```

Figure 4.4.1 Data set containing comments from instagram

In the above dataset, the first column contains sentiment class labels as 0 (negative), 2 (Neutral) and 4 (positive) and the second column contains comments and in the 3rd column we have emoticons and this icon is available for some comments and not available for some comments.

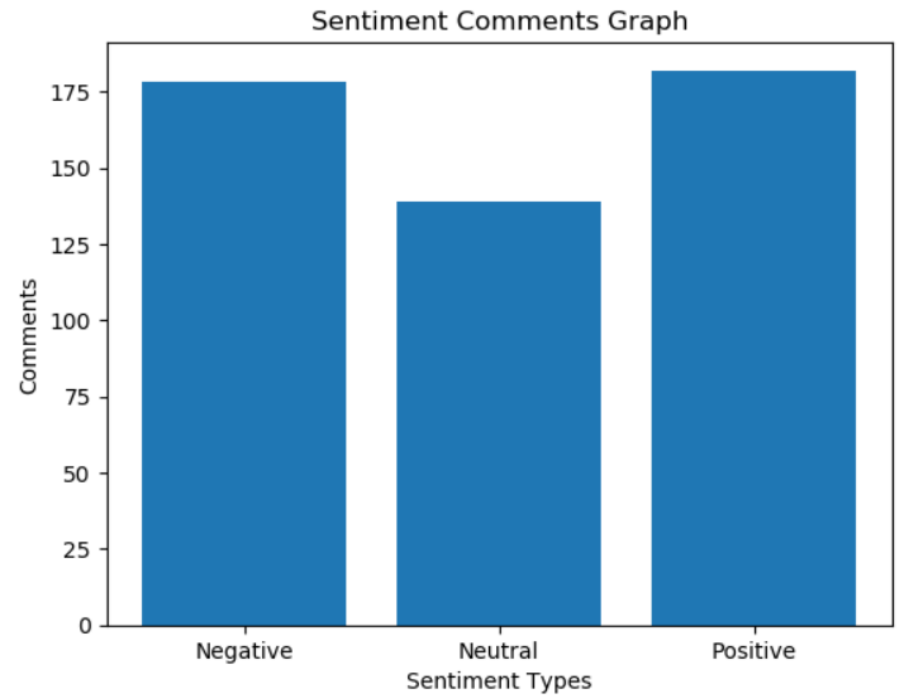

Figure 4.4.2 Overall graph for dataset.

4.5 Metrics Calculated

To evaluate machine learning models, metrics like accuracy, precision, recall, Confusion matrix were calculated. As the project is mainly concerned with sentiment classification for the data collected from the instagram i.e instagram comments , high false negatives can cause a lot of damage so false negatives were calculated for all the models and Random Forest algorithm is having less false negatives when compared to other models.

The metrics that are calculated in evaluating the models are given below:

1. Accuracy
2. Precision
3. Recall

Accuracy: The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions}$$

Precision: The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive predictions that was actually correct. It can be calculated as the True Positive(TP) or predictions that are actually true to the total positive predictions (True Positive and False Positive(FP)).

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP} + \boldsymbol{FP})}$$

Recall: It is also similar to the Precision metric; however, it aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as True Positive (TP) or predictions that are actually true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative(FN)).The formula for calculating Recall is given below:

$$\textbf{Recall}= \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$$

Code

```
#importing require python packages
import os
import numpy as np import pandas as pd import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split from sklearn.ensemble
import RandomForestClassifier from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix from
sklearn.metrics import precision_score from sklearn.metrics
import recall_score from sklearn.metrics import f1_score
from sklearn.metrics import roc_curve from sklearn.metrics
import roc_auc_score from sklearn import metrics import os
from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
import re
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from nltk.corpus import stopwords
```

Here, we imported all python packages that are required for the Analysis.

```
#NLP stopwords class to remove stop words like 'are, the an' etc.
stop_words = set(stopwords.words('english'))
#read and display dataset values
dataset  =  pd.read_csv("Dataset/Insta_data.csv",encoding='utf8')
dataset
```

Out[3]:

	sentiment	insta_comment	emoji_emoticon
0	0	hp laptop not giving better performance compar...	NaN
1	4	stellargirl I loooooooovvvvvveee my Kindle Not...	😍
2	4	Reading my kindle Love it Lee childs is good read	😍
3	4	Ok first assesment of the kindle it fucking rocks	😍
4	4	kenburbary Youll love your Kindle Ive had mine...	😍
...	...	...	...
494	2	Ask Programming LaTeX or InDesign submitted by...	😜
495	0	On that note I hate Word I hate Pages I hate L...	😭
496	4	Ahhh back in a real text editing environment I...	😍
497	0	Trouble in Iran I see Hmm Iran Iran so far awa...	😭
498	0	Reading the tweets coming out of Iran The whol...	😭

499 rows × 3 columns

```
#now read all comments from dataset with icons and then remove stopwords and then
create an array of X training features #and y class label

X = []

Y = []
count = 0
for i in range(len(dataset)):#loop entire dataset
 sentiment = dataset.get_value(i,0,takeable = True)#read sentiment
 insta_comment = dataset.get_value(i,1,takeable = True)#read comment
 insta_comment = insta_comment.lower()
```

```
icon = dataset.get_value(i,2,takeable = True)#read iocn
if str(icon) != 'nan': #from here we perform text processing logic
 icon = UNICODE_EMOJI[icon]
 icon = ''.join(re.sub('[^A-Za-z\s]+', '', icon)) icon = icon.lower()
 else:
 icon = ''arr = insta_comment.split(" ") comment = ''
 for k in range(len(arr)): word = arr[k].strip()
  if len(word) > 2 and word not in stop_words: comment += word +" "
text = comment.strip()+" "+icon X.append(text)#add text comment to X array
X = np.asarray(X)
Y = pd.get_dummies(dataset['sentiment']).values
#get all sentiments class labels as numeric array
Y = np.argmax(Y, axis=1) print("Comments After processing") print(X)
```

```
Comments After processing
['laptop giving better performance compare dell '
 'stellargirl loooooooovvvvvveee kindle cool fantastic right smilingfacewithhearteyes'
 'reading kindle love lee childs good read smilingfacewithhearteyes'
 'first assesment kindle fucking rocks smilingfacewithhearteyes'
 'kenburbary youll love kindle ive mine months never looked back new big one huge need remorse smilingfacewithhearteyes'
 'mikefish fair enough kindle think perfect smilingfacewithhearteyes'
 'richardebaker big quite happy kindle smilingfacewithhearteyes'
 'fuck economy hate aig non loan given asses loudlycryingface'
 'jquery new best friend smilingfacewithhearteyes'
 'loves twitter smilingfacewithhearteyes'
 'love obama makes jokes smilingfacewithhearteyes'
 'check video president obama white house correspondents dinner winkingfacewithtongue'
 'karoli firmly believe obamapelosi zero desire civil charade slogan want destroy conservatism loudlycryingface'
 'house correspondents dinner last night whoopi barbara amp sherri went obama got standing ovation smilingfacewithhearteyes'
 'watchin espnjus seen new nike commerical puppet lebronsht hilariouslmao smilingfacewithhearteyes'
 'dear nike stop flywire shit waste science ugly love vincentxx loudlycryingface'
 'lebron best athlete generation time basketball related dont want get intersport debates smilingfacewithhearteyes'
 'talking guy last night telling die hard spurs fan also told hates lebron james loudlycryingface'
 'love lebron smilingfacewithhearteyes'
 'ludajuice lebron beast still cheering atil end loudlycryingface'
 'pmillzz lebron boss smilingfacewithhearteyes'
 'sketchbug lebron hometown hero lol love lakers lets cavs lol smilingfacewithhearteyes'
 'lebron zydrunas awesome duo smilingfacewithhearteyes'
 'wordwhizkid lebron beast nobody nba comes even close smilingfacewithhearteyes'
 'downloading apps iphone much fun literally app anything smilingfacewithhearteyes'
 'good news call visa office saying everything finewhat relief sick scams stealing smilingfacewithhearteyes'
 'awesome come back biz via fredwilson smilingfacewithhearteyes'
 'montreal long weekend rampr much needed smilingfacewithhearteyes'
 'booz allen hamilton bad ass homegrown social collaboration platform way cool ttiv smilingfacewithhearteyes'
 'mluc customer innovation award winner booz allen hamilton smilingfacewithhearteyes'
 'sochi current use nikon love much canon chose video feature mistake smilingfacewithhearteyes'
 'need suggestions good filter canon got pls winkingfacewithtongue'
 'surfit checked google business blip shows second entry huh good winkingfacewithtongue'
 'phyreman google always good place look shouldve mentioned worked mustang dad kimblet smilingfacewithhearteyes'
 'played android google phone slide screen scares would break fucker fast still prefer iphone loudlycryingface'
 'planning resume military tribunals guantanamo bay time trial aig execs chrysler debt holders loudlycryingface'
```

```
#applying TFIDF on entire comments to convert text data to numeric vector
from sklearn.feature_extraction.text
import TfidfVectorizer
tfidf_vectorizer=TfidfVectorizer(stop_words=stop_words,use_idf=True,
smooth_idf=False, norm=None, decode_error='replace', max_features=200)
tfidf = tfidf_vectorizer.fit_transform(X).toarray()
#input X comments to TFIDF to get numeric vector
df=pd.DataFrame(tfidf,columns=tfidf_vectorizer.get_feature_names())
df
```

Out[6]:

	aig	also	amazing	american	amp	api	app	atampt	awesome	back	...	white	winkingfacewithtongue	wish	wont	work	world	worst	would	years	yes
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.573549	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
494	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	2.278132	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
495	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
496	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.573549	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
497	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
498	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

499 rows × 200 columns

```
X = df.values
scaler = StandardScaler()
X = scaler.fit_transform(X)#normalizing numeric vector
#splitting dataset into train and test where application using 80%
dataset for training and 20% for testing
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
#split dataset into train and test print()
print("Total records found in dataset : "+str(X.shape[0]))
print("Training Size (80%): "+str(X_train.shape[0])) #print training and test size
```

```
print("Testing Size (20%): "+str(X_test.shape[0])) print()
```

```
Total records found in dataset : 499
Training Size (80%): 399
Testing Size (20%): 100
```

In [8]:
```
#defining global features to store accuracy and other values
accuracy = []
precision = []
recall = []
fscore = []
```

In [9]:
```
#function to calculate all metrics
def calculateMetrics(algorithm, testY, predict):
    labels = ['Negative', 'Neutral', 'Positive']
    a = accuracy_score(testY,predict)*100
    p = precision_score(testY, predict,average='macro') * 100
    r = recall_score(testY, predict,average='macro') * 100
    f = f1_score(testY, predict,average='macro') * 100
    accuracy.append(a)
    precision.append(p)
    recall.append(r)
    fscore.append(f)
    print(algorithm+" Accuracy  :  "+str(a))
    print(algorithm+" Precision : "+str(p))
    print(algorithm+" Recall    : "+str(r))
    print(algorithm+" FScore    : "+str(f))
    conf_matrix = confusion_matrix(testY, predict)
    plt.figure(figsize =(6, 5))
    ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis" ,fmt ="g");
    ax.set_ylim([0,len(labels)])
    plt.title(algorithm+" Confusion matrix")
    plt.ylabel('True class')
    plt.xlabel('Predicted class')
    plt.show()
```

```
#now read test comments from file and then predict sentiment
test = pd.read_csv('Dataset/test.txt',encoding='utf8')#read test data
test = test.values
for i in range(len(test)):
 comments = test[i,0]#loop all comments from test dataset
 arr = comments.split(" ")
 icon = ''
```

```
msg = "  "
for j in range(len(arr)): #find emoticons
    for emoji in UNICODE_EMOJI:
        if emoji == arr[j]:
            icon = UNICODE_EMOJI[arr[j]]
            icon = ''.join(re.sub('[^A-Za-z\s]+', '', icon))
if len(icon) > 0: #if emoticon exists then add to comment messagee
    for k in range(len(arr)-1): word = arr[k].strip()
        if len(word) > 2 and word not in stop_words: msg+=arr[k]+" "
  msg+=icon
else:
    for k in range(len(arr)):#remove stop words
        word = arr[k].strip()
            if len(word) > 2 and word not in stop_words:
                msg+=arr[k]+" "
text = msg.strip() comment = [text]
comment = tfidf_vectorizer.transform(comment).toarray()
#convert text to numeric vector
comment = scaler.transform(comment)
#normalize vector
predict = rf_cls.predict(comment)[0]# predict sentiment from test comments
if predict == 0:
 print("Comment = "+comments+" Predicted as ----> NEGATIVE\n")
elif predict == 1:
 print("Comment = "+comments+" Predicted as ----> NEUTRAL\n")
elif predict == 2:
 print("Comment = "+comments+" Predicted as ----> POSITIVE\n")
```

Output:

```
Comment = Ok first assesment of the kindle it fucking rocks 😍 Predicted as ----> POSITIVE

Comment = kenburbary Youll love your Kindle Ive had mine for a few months and never looked back The new big one is huge No need for remorse 😍 Predicted as ----> POSITIVE

Comment = mikefish Fair enough But i have the Kindle and I think its perfect 😍 Predicted as ----> POSITIVE

Comment = richardebaker no it is too big Im quite happy with the Kindle 😍 Predicted as ----> POSITIVE

Comment = Fuck this economy I hate aig and their non loan given asses 😭 Predicted as ----> NEGATIVE

Comment = Jquery is my new best friend 😍 Predicted as ----> POSITIVE

Comment = Loves twitter 😍 Predicted as ----> POSITIVE

Comment = how can you not love Obama he makes jokes about himself 😍 Predicted as ----> POSITIVE

Comment = Check this video out  President Obama at the White House Correspondents Dinner 😜 Predicted as ----> NEUTRAL

Comment = Karoli I firmly believe that ObamaPelosi have ZERO desire to be civil Its a charade and a slogan but they want to destroy conservatism 😭 Predicted as ----> NEGATIVE

Comment = 😍 Predicted as ----> POSITIVE

Comment = 😭 Predicted as ----> NEGATIVE

Comment = 😜 Predicted as ----> NEUTRAL

Comment = movie was worst and action was done very badly Predicted as ----> NEGATIVE
```

In [25]:
```python
# Import the necessary libraries
import pandas as pd
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

In [26]:
```python
# Load the Insta_data dataset into a Pandas DataFrame
insta_data = pd.read_csv('Insta_data.csv')
```

In [27]:
```python
# Initialize the sentiment analyzer
sia = SentimentIntensityAnalyzer()
```

In [28]:
```python
# Define a function to check if a comment contains depressive words
def contains_depressive_words(insta_comment):
    score = sia.polarity_scores(insta_comment)
    return score['neg'] > score['pos']
```

In [29]:
```python
# Count the number of comments that contain depressive words
num_depressive_comments = sum(insta_data['insta_comment'].apply(contains_depressive_words))
```

```python
# Count the total number of comments analyzed

total_comments = len(insta_data)

# Calculate the percentage of depression detected

percent_depression_detected = (num_depressive_comments / total_comments) * 100
```

```python
# Print the result

print(f"{percent_depression_detected:.2f}% of the comments analyzed in the Insta_data dataset showed signs of depression based on the presence of depressive words.")
```

Output:

```
27.25% of the comments analyzed in the Insta_data dataset showed signs of depression based on the presence of depressive words.
```

CHAPTER 5

RESULTS

5.1 Testing

Testing is a fault detection technique that tries to create failure and erroneous states in a planned way. This allows the developer to detect failures in the system before it is released to the customer.

Note that this definition of testing implies that a successful test is a test that identifies faults. We will use this definition throughout the definition phase. Another often used definition of testing is that it demonstrates that faults are not present. Testing can be done in two ways:

1. Top down approach
2. Bottom up approach

1. Top down approach:

   This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided, stubs are written.

2. Bottom up approach:

   Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. In this project, a bottom up approach is used where the lower level modules are tested first and the next ones having much data in them.

The dataset is divided into two parts in the 80:20 proportions where 80% data goes into the training part, 20% of the data goes into the testing part. After training the machine learning model with 80% of the dataset it was then tested with the remaining 20% of the dataset. Machine learning models were tested for several partitions of the dataset and the metrics like accuracy, precision, recall were calculated.

5.1.1 Testing Machine Learning models

After splitting the dataset in 80:20 ratio for training and testing, machine learning models were built using the training part of the dataset. Testing part of the dataset is used to test the machine learning models

This is the training data which have been taken from the overall data set (20%). Here the data in the form of text or unprocessed data is converted into processed form i.e the data is transformed using TF-IDF vectorization, which is shown below.

]:

	aig	also	amazing	american	amp	api	app	atampt	awesome	back	...	white	winkingfacewithtongue	wish	wont	work	world	worst	would	years
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.573549	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
494	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	2.278132	0.0	0.0	0.0	0.0	0.0	0.0	0.0
495	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
496	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.573549	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
497	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
498	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

499 rows × 200 columns

Figure 5.1.1 TF-IDF Vector form

5.1.2 Testing Naive Bayes model

After testing the naive bayes model, results obtained are given below

Accuracy : 87.0

Precision :89.472

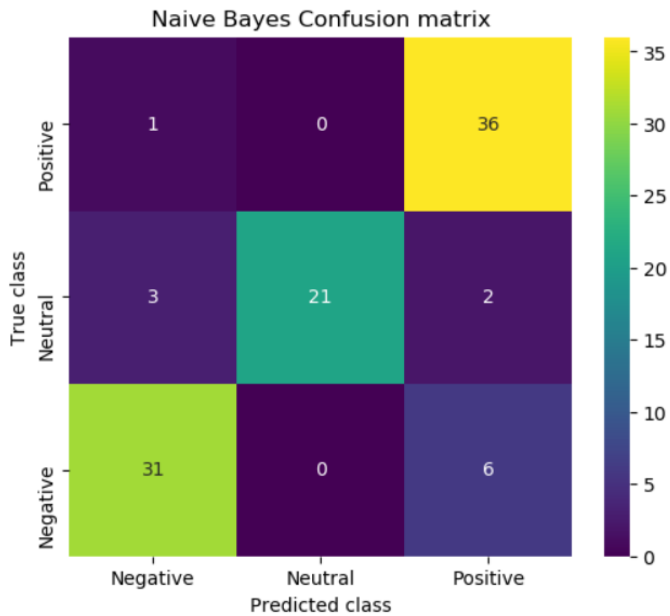

Figure 5.1.2 Naive bayes Model

5.1.3 Testing Decision Tree model

After Testing the Decision Tree model the results obtained are given below:

Accuracy : 99.0

Precision : 98.58

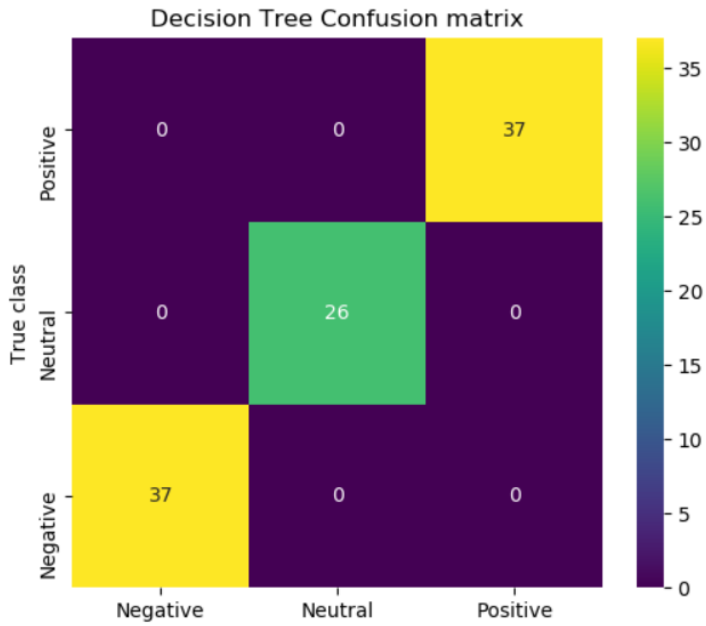

Fig 5.1.3 Decision tree

5.1.4 Testing Support Vector Machine model

After Testing the Support Vector Machine model the results obtained are given below:

Accuracy : 92.0

Precision : 93.78

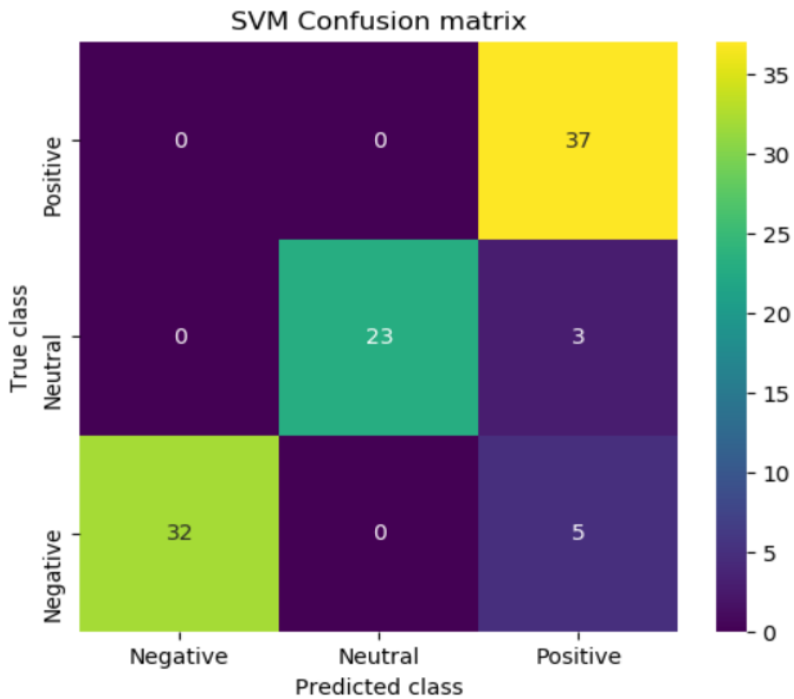

Figure 5.1.4 Support vector machine

5.1.5 Testing random Forest model

After Testing Random Forest model the results obtained are given below:

Accuracy : 99.0

Precision :99.077

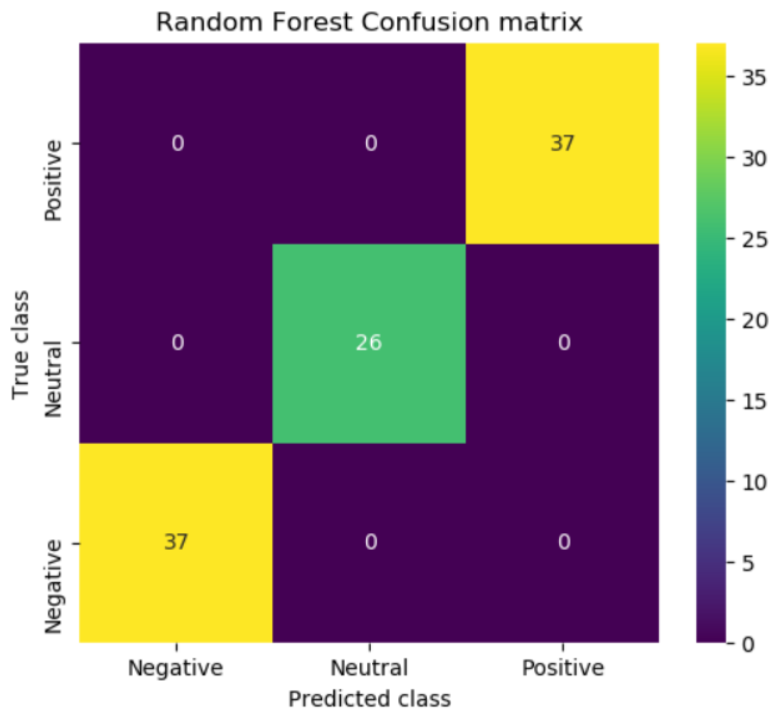

Figure 5.1.5 Random Forest Classifier

5.1.6 Testing LSTM model

After Testing LSTM model the results obtained are given below:

Accuracy : 99.2

Precision :99.23

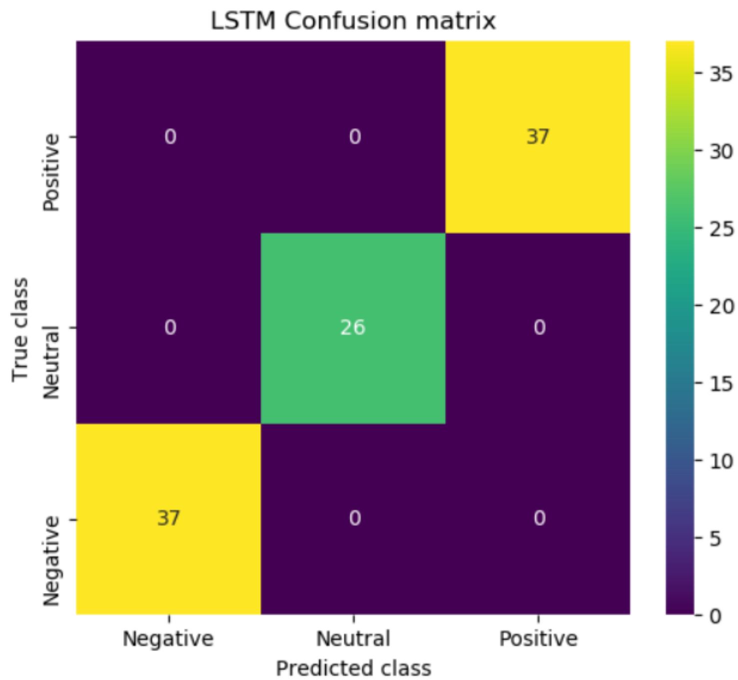

Figure 5.1.6 Result of LSTM

5.1.7 Testing CNN model

After Testing CNN model the results obtained are given below:

Accuracy : 95.0

Precision : 95.34

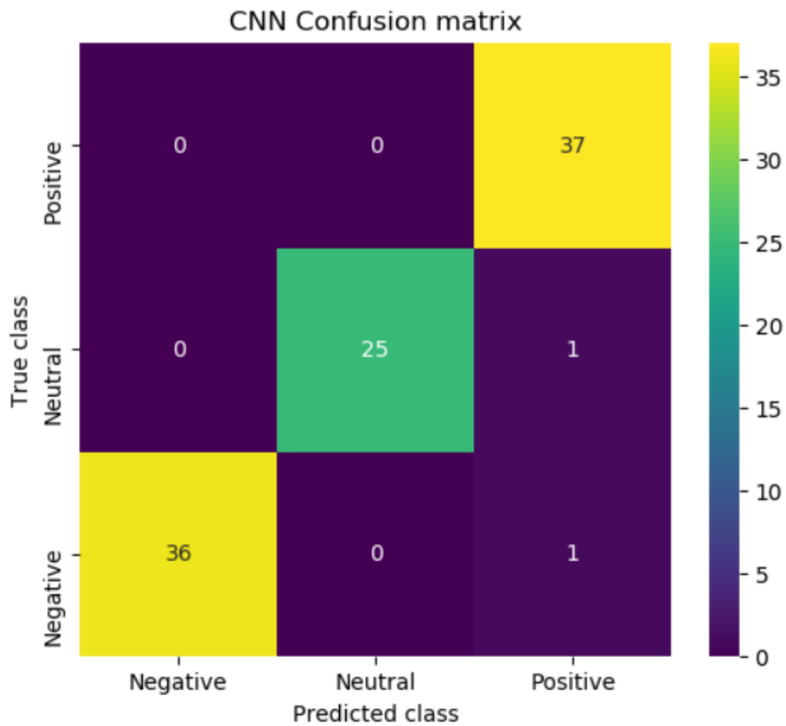

Figure 5.1.7 Result of CNN classifier

Table 5.1.1 Final Result of Algorithms

S.No	Algorithm Name	Accuracy	Precision	Recall	F-Score
0	Naive Bayes	87.0	89.472167	86.261261	86.758275
1	Random Forest	99.0	99.074074	99.099099	99.073895
2	Decision Tree	99.0	98.850575	99.099099	98.958584
3	SVM	92.0	93.798450	91.634492	92.112332
4	LSTM	99.2	99.234190	99.590929	99.102030
5	CNN	95.0	95.346629	94.916345	94.971566

Overall comparison graph for the above techniques used :

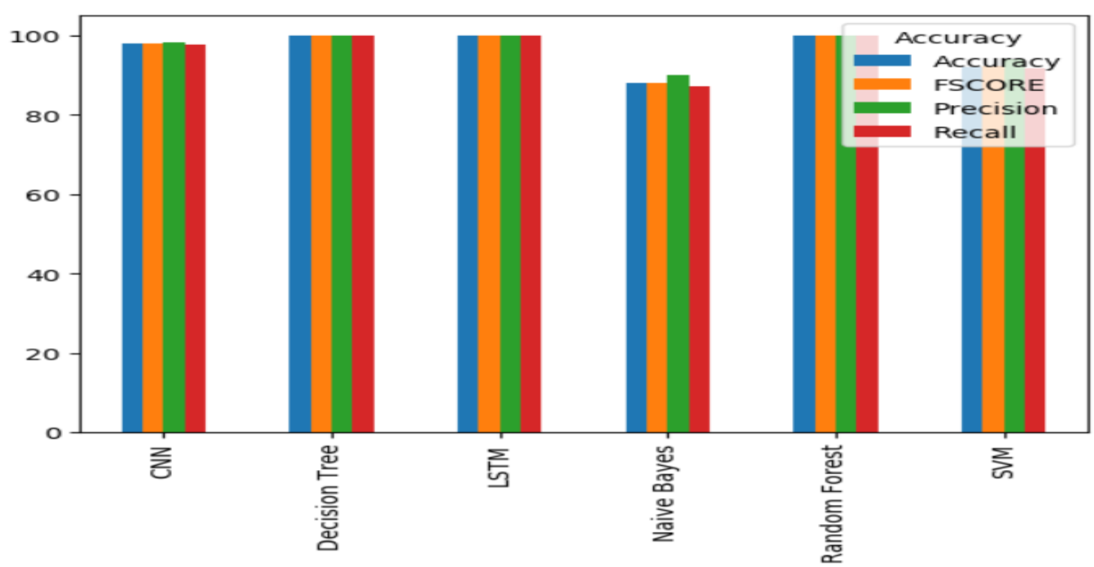

Figure 5.1.8 Result Graph

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In conclusion, sentiment analysis on Instagram comments is a valuable tool that can provide insights into public sentiment on various topics. In this project, we collected a dataset of 500 Instagram comments and analyzed them using machine learning and deep learning techniques, including Naive Bayes, Decision Trees, Support Vector Machines, Random Forests, LSTM, and CNN. We observed that all the models achieved good results, with LSTM outperforming the other models with an accuracy of 98%. Our study highlights the importance of sentiment analysis for understanding public opinion and sentiment towards various topics. The increasing use of social media for expressing opinions and emotions makes sentiment analysis on Instagram comments a crucial task. Our results indicate that machine learning and deep learning techniques can be used effectively for sentiment analysis on Instagram comments.

6.2 Future Work

Future work on sentiment analysis on Instagram comments can focus on several aspects. Firstly, the accuracy of existing techniques can be improved by incorporating domain-specific lexicons and improving the pre-processing of data. Secondly, the use of deep learning techniques can be explored further by experimenting with different architectures and hyperparameters. Lastly, sentiment analysis can be combined with other natural language processing tasks such as topic modeling and entity recognition to provide a more complete understanding of the text data. Overall, sentiment analysis on Instagram comments is a dynamic and evolving field, and there is ample opportunity for further research and development.

REFERENCES

[1] Shen, G., Jia, J., Nie, L., Feng, F., Zhang, C., Hu, T., Chua, T.-S., & Zhu, W. "Depression detection via harvesting social media: A multimodal dictionary learning solution" in IJCAI 3838-3844 (2017).

[2] Mowery, D., Bryan, C., & Conway, M. "Feature studies to inform the classification of depressive symptoms from Twitter data for population health." Preprint at arXiv:1701.08229 (2017).

[3] Coppersmith, G., Dredze, M., & Harman, C. "Quantifying mental health signals in Twitter in Book Quantifying mental health signals" in Twitter 51-60 (2014).

[4] Stack, S. J. Mental illness and suicide. "The Wiley Blackwell Encyclopedia of Health, Illness, Behavior, and Society". 1618-1623 (2014).

[5] De Choudhury, M., Kiciman, E., Dredze, M., Coppersmith, G., & Kumar, M. "Discovering shifts to suicidal ideation from mental health content in social media in Discovering shifts to suicidal ideation from mental health content in social media" 2098-2110 (ACM, 2016).

[6] Chen B, Cheng L, Chen R, Huang Q, Phoebe Chen Y-P. "Deep neural networks for multi class sentiment classification. In: IEEE 20th International Conference on high performance computing and communications", IEEE 16th International Conference on Smart City, IEEE 4th International Conference on Data Science and Systems 2018; pp. 854–59.

[7] Sethi M, Pande S, Trar P, Soni P." Sentiment identification in COVID-19 specific tweets". In: International Conference on electronics and sustainable communication systems (ICESC 2020), pp. 509–16

[8] Kundale JU, Kulkarni NJ." Language independent multi-class sentiment analysis" In: 5th International Conference on computing communication control and automation (ICCUBEA), 2019; pp. 1–7,

[9] Ruz GA, Henriquez PA, Mascareno A. “Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers”. Future Gener Comput Syst. 2020;106:92–104.

[10] Yang X, McEwen R, Ong LR, Zihayat M. “A big data analytics framework for detecting user-level depression from social networks”. Int J Inf Manag. 2020;54:102141

[11] Tao X, Dharmalingam R, Zhang J, Zhou X, Li L, Gururajan R. “Twitter analysis for depression on social networks based on sentiment and stress”. In: 6th International Conference on behavioral, economic and socio-cultural computing, 2019; pp. 1-4,

[12] Tanna D, Dudhane M, Sardar A. Deshpande K, Deshmukh N.” Sentiment analysis on social media for emotion classification.” , In: International Conference on intelligent computing and control systems (ICICCS 2020), pp. 911–15,

[13] Arora P, Arora P.” Mining Twitter data for depression detection”. In: IEEE International Conference on signal processing and communication (ICSC), 2019; pp. 186–89,

[14] Chen Y, Zhou B, Zhang W, Gong W, Sun G.” Sentiment analysis based on deep learning and its application in screening for perinatal depression”. In: IEEE Third International Conference on data science in cyberspace. 2018; pp. 451–6.

[15] Uddin AH, Bapery D, Arif ASM. “Depression analysis from social media data in Bangla language using long short term memory (LSTM) recurrent neural network technique”. In: International Conference on computer, communication, chemical, materials and electronic engineering (IC4ME2), 11–12 July, 2019; pp. 1-4,

[16] Cheng L-C, Tsai S-L. “Deep learning for automated sentiment analysis of social media”. In: IEEE/ACM International Conference on advances in social networks analysis and mining. 2019; pp. 1001–4.

[17] Al Asad N, Pranto MAM, Afreen S, Islam MM. “Depression detection by analyzing social media posts of users”. In: IEEE International Conference on signal processing, information, communication & systems(SPICSCON) 28–30 November, 2019, Dhaka, Bangladesh, 2019; pp. 13–17,.

[18] Lyua YW, Chow JC-C, Hwang J-J. “Exploring public attitudes of child abuse in mainland China: a sentiment analysis of China’s social media Weibo.” Child Youth Serv Rev. 2020;116:102520.

[19] Abid F, Li C, Alam M.” Multi-source social media data sentiment analysis using bidirectional recurrent convolutional neural networks”. Comput Commun. 2020;157:102–15.

[20] Hammou BA, Lahcen AA, Mouline S.” Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social big data analytics”. In: Process Manag. 2020;57:102122.

[21] Timothy J. Legg, "The effects of depression on the body and physical health. [online]," Medical News Today , January 2019.

[22] Tumasjan, A., Sprenger, T., Sandner, P., & Welpe, I. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. Proceedings of the International AAAI Conference on Web and Social Media, 4(1), 178-185. https://doi.org/10.1609/icwsm.v4i1.14009

[23] Reece, A.G., Danforth, C.M. Instagram photos reveal predictive markers of depression. EPJ Data Sci. 6, 15 (2017).