**MAULANA AZAD**
**NATIONAL INSTITUTE OF TECHNOLOGY**
**BHOPAL, INDIA, 462003**

---


---

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

# WhatsApp Chat Sentiment Analysis

## Minor Project Report

### Semester VI

**Submitted by:**

| | |
|---|---|
| **Pawan Patel** | **191112280** |
| **Mridul Karan** | **191112273** |
| **Himanshu Kumar** | **191112248** |
| **Priyanshu Mangal** | **191112250** |

**Under the Guidance of**
Dr. Deepak Singh Tomar
Associate Professor
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
**Session: 2021-22**

**MAULANA AZAD**
**NATIONAL INSTITUTE OF TECHNOLOGY**
**BHOPAL, INDIA, 462003**

---


---

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## CERTIFICATE

This is to certify that the project report carried out on "WhatsApp Chat Sentiment Analysis" by the 3rd year students:

| | |
|---|---|
| **Pawan Patel** | **191112280** |
| **Mridul Karan** | **191112273** |
| **Himanshu Kumar** | **191112248** |
| **Priyanshu Mangal** | **191112250** |

Have successfully completed their project in partial fulfilment of their Degree in Bachelor of Technology in Computer Science and Engineering.

**Dr. Deepak Singh Tomar**
**Associate Professor**
**(Minor Project Supervisor)**

# DECLARATION

We, hereby declare that the following report which is being presented in the Minor Project Documentation Entitled as “WhatsApp Chat Sentiment Analysis” is an authentic documentation of our own original work and to the best of our knowledge. The following project and its report, in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization. Any contribution made to the research by others, with whom we have worked at Maulana Azad National Institute of Technology, Bhopal or elsewhere, is explicitly acknowledged in the report.

| | |
|---|---|
| **Pawan Patel** | **191112280** |
| **Mridul Karan** | **191112273** |
| **Himanshu Kumar** | **191112248** |
| **Priyanshu Mangal** | **191112250** |

# ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide and coordinator Prof. Deepak Singh Tomar, for his valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully.

It is imperative for us to mention the fact that the report of the minor project could not have been accomplished without the periodic suggestions and advice of our project guide Dr. Deepak Singh Tomar and project coordinators Dr. Dhirendra Pratap Singh and Dr. Jaytrilok Choudhary.

We are also grateful to our respected director Dr. N. S. Raghuwanshi for permitting us to utilize all the necessary facilities of the college.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help. Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing the much-needed support and encouragement.

# ABSTRACT

The most used and efficient method of communication in recent times is an application called WhatsApp. WhatsApp chats consist of various kinds of conversations held among groups of people. This chat consists of various topics. This information can provide lots of data for the latest technologies such as machine learning. The most important thing for a machine learning model is to provide the right learning experience which is indirectly affected by the data that we provide to the model. This tool aims to provide in-depth analysis of this data which is provided by WhatsApp. Irrespective of whichever topic the conversation is based on, our developed code can be applied to obtain a better understanding of the data. The advantage of this tool is that It is implemented using simple python modules such as pandas, matplotlib, seaborn and sentiment analysis which are used to create data frames and plot different graphs, where then it is displayed in the using Streamlit web app which is efficient and can be modified easily, therefore it can be easily applied to large dataset.

## Contents

# List of Figures

# List of Tables

# 1. Introduction

Chat Analysis and Sentiment Analysis are the two elements of this research. This application is built around data analysis and processing. The first stage in implementing a machine learning algorithm is determining the appropriate learning experience from which the model will begin to improve. When it comes to machine learning, data pre-processing is crucial. We needed a lot of data to make the model more efficient, so we focused mostly on one of Facebook's big scale data providers, WhatsApp. WhatsApp promises to send approximately 55 billion messages every day. The average WhatsApp user spends 195 minutes each week on the app and is a member of several groups. With this treasure trove of data right under our noses, we must begin on a journey to obtain insights into the messages that our phones are obligated to record.

The statistical study of people's views, feelings, emotions, assessments, and attitudes regarding things such as products, services, organizations, persons, issues, events, themes, and their qualities is known as sentiment analysis or opinion mining. The field's birth and quick expansion correlate with those of the Web's social media, such as reviews, forum debates, blogs, micro-blogs, Twitter, and social networks, since, for the first time in human history, we have a massive amount of opinionated material stored in digital formats. Sentiment analysis has evolved to be one of the most active study fields in natural language processing since the early 2000s (NLP). It is also extensively researched in data mining, Web mining, text mining, and information retrieval. Because of its relevance to business and society as a whole, it has moved from computer science to management sciences and social sciences such as marketing, finance, political science, communications, health science, and even history.

This growth is owing to the fact that views are fundamental to practically all human activities and have a significant role in shaping our behaviour.

Our views and perceptions of reality, as well as the decisions we make, are heavily influenced by how others see and interpret the world. As a result, anytime we need to make a decision, we frequently seek the advice of others. This is true not only for individuals, but also for companies.

If one wishes to buy a consumer goods, one is no longer confined to asking friends and family for advice because there are many user reviews and debates about the product in public forums on the Internet. Because there is an excess of such information publicly available, an organization may no longer need to undertake surveys, opinion polls, and focus groups in order to acquire public opinions. In recent years, we have seen how opinionated social media comments have helped restructure businesses and sway public feelings and emotions, all of which have had a tremendous impact on our social and political institutions. Such postings have also organized people in support of political changes, such as those that occurred in various Arab nations in 2011.As a result, gathering and analysing opinions has become a requirement.

However, due to the development of varied sites, locating and monitoring opinion sites on the Web, as well as distilling the information contained in them, remains a difficult endeavour. Each site often has a large amount of opinion content, which is not always easy to comprehend in lengthy blogs and forum discussions

# 2. Literature Review and Survey

## Statistical analysis:

In a study of the southern part of India was conducted on the age group of between 18 to 23 years to investigate the importance of WhatsApp among youth. Though, this study, it was found that students spent 8 hours per day using WhatsApp and remained online almost 16 hours a day. All the respondents agreed that they are using WhatsApp for communicating with their friends. They also exchange images, audio and video files with their friends using WhatsApp. It was also proved that the only application that the youth uses when they are spending time on their smartphone is WhatsApp. Methods used in this survey is to analyse the intensity of WhatsApp usage and its popular services and to identify the degree of positive or negative impacts of using WhatsApp.

## Sentiment Analysis:

There are various machine learning algorithms which are being used for sentiment analysis and giving nearly satisfied results.

In [1] Suppala K, Rao N developed the sentiment analysis using Naive Bayes Classifier. Here, they tried to predict the emotion of the sentence whether it is positive, negative or neutral.In [2], They used K-Nearest Neighbour Classifier to do the Sentiment Analysis of Online Movies based on the reviews. In [3], They used Support Vector Machine(SVM) to find the Sentiments from the customer reviews on a product. However Deep Neural Networks(DNN) are also doing well in the field of Natural Language Processing(NLP). language modeling [4], sentiment analysis [5], syntactic parsing [6], and machine interpretation [7]. A recurrent neural network(RNN) is a different kind of neural network, which creates a directed cycle by making the connections between neurons, which makes it unique among other models. Long Short Term Memory(LSTM) is one of the special variations of RNN. There are projects going on LSTM for sentiment analysis on various research facilities, but there is still a lot of work to do on this model and this field. That's why, an efficient sentimental analysis methodology is presented in this work

The purpose of this proposed methodology is to predict the sentiments of people in the group or in the personal conversation. The proposed model has four modules namely, pre-processing of data, feature engineering, chat analysis, and prediction of emotions based on the polarity of that sentence. In the first step, the data is gathered using the chat export feature in WhatsApp and then text pre-processing is done on the obtained text file to improve the data quality. Then lemmatization, word embedding and vectorization is done. Then, the chat analysis is shown in different types of graphs, and then the extracted features are given to the models to predict the emotions of the chat.

There are different types of sentiment analysis models that have been developed. Some of them are discussed here. The paper [8] discusses how the spreading of fake news and rumour problems

can be solved using natural language processing. They used bag-of-words, n-grams, count vectorizer methods and TF-IDF model in order to solve the problem. They classified the data in five classifiers to check which of them is closely related to the specific news statements.

In [9], Tanjim Ul Haque; Nudrat Nawal Saber; Faisal Muhammad Shah, they discussed about how the process of selection of best product can be made easier on online platform by doing the sentiment analysis of reviews of that product using Natural Language Processing concepts. They build a model which polarizes the reviews and learns from it. They trained this model using supervised learning methods on a large-scale dataset of an ecommerce website to get the best accuracy.

Sheng et al. [10] In this, the model which finds rumors based on consumer opinion was discussed. They discussed the sentimental characteristic of the comment and along with that they also discussed the huge amount of information on comments. They used a Convolution Neural Network (CNN) with LSTM, where LSTM is connected to both CNN and LSTM networks.

Zhigang et al. [11] explained that a stock closing forecast depends on sentiment analysis. To improve the stock forecasting accuracy of the model they included the perception of investors on forecasting. Getting a good accuracy in stock forecasting is not an easy task because of varying time fluctuations giving a hard-to-understand time series of stock price sequence. They first introduced a gradual decomposition of the stock price complex by adding the empirical model decomposition (EMD) which resulted in better prediction accuracy. After that, they implemented the LSTM model because of its memory specialty to incorporate the time series data. Because of memory function, it not only improved the result a lot but it also helped in reducing the delay

Fu et al. [12], discussed Lexicon - enhanced LSTM model using an attention mechanism. This research discusses how quality improvement in word embedding can be reflected in the sentiment classifier's accuracy using the sentiment lexicon. Combining the word embedding and sentiment embedding makes the representation of the word more accurate. He tested this proposed method on English and Chinese datasets named Yelp2013, MR, IMDB and NB4000 of respective languages. In comparison with WALE-LSTM and ALE-LSMT this method showed comparative better results than existing models, with the highest obtained accuracy of 89%, 79.0%, 93% and 96%, when compared with

Guixian et al. [13] proposed an improved method of word representation. The distributed word representation only considers the semantic information of a word ignoring the information about the sentiment of the word. Therefore a better way of word representation is presented here which integrates the contribution of sentiment with the traditional Term Frequency-Inverse Document Frequency (IF-IDF) algorithms and generates weighted word vectors. The Weighted vector is input into a Bi-Directional long short term memory (BiLSTM) network to analyze the information about the context. Here Relu activation function in order to overcome gradient vanishing problem and overfitting.

Da'U and Salim[22] made a neural attention-based recommender system which comprises LSTM encoder, semi-supervised topic model, co-attention, analysis layer for prediction of rating of users. This model analyzes the sentiment lexicons and increases the efficiency of the system. Sailunaz and Allhajj [23] wrote about sentiment analysis of comments on Twitter. This uses the Naive Bayes classifier and is topic based general recommendation and user-based recommendations.

Shoieb and Ajit had done sentiment analysis on web data based on emoticons. In pre-processing POS tagging, removal of stopwords, lemmatization were done. Then derivation of sentiwordnet was done and then classification was done using SVM, IBK, MLP and Naive bias. Naive Bayes classifier had the highest precision of 84.7%(college dataset) and 83.3%(hospital dataset).Zeeshan et al. [25] presented a lexicon and ANN based Sentiment analysis. Using a movie review dataset containing two labels positive and negative the network training was done with an accuracy of 91%.

# 3. Gaps Identified

In the previous works they have implemented the sentiment analysis for Twitter comments, Amazon Reviews and IMDB reviews etc., but there is very less work on WhatsApp Chat Sentiment Analysis. WhatsApp Chat Analysis is not similar to above mentioned because we know the topic of discussion beforehand but that's not case with WhatsApp and also Talks on Twitter, Amazon and IMDB are professional and the whole statement is written at once fully, but in case of WhatsApp the statements may not be complete and is may in short form or unprofessional.

# 4. Proposed work

Almost everyone has chat apps installed in their phones and devices in today's world. It is manually impossible to understand the statistical and sentimental inferences behind each text. It becomes a prominent problem in fields such as forensics where the intent and activity of a person has to be observed and recorded.

There have been many advancements in the study of chat analysis such as multi-chat analysis, sentiment analysis and opinion mining. This is of utmost use for many government agencies and also companies who on this day are dependent on customer reviews and satisfaction.

Data preprocessing is the initial part of the project. This task can be accomplished by using built-in python modules and is better than writing the functions from scratch. These various modules provide better code representation and user understandability. The following libraries are used such as NumPy, SciPy pandas, csv, sklearn, matplotlib, sys, re, emoji, nltk seaborn etc. are used for exploratory data analysis. After pre-processing, the first step is to apply a sentiment analysis algorithm which provides positive, negative and neutral parts of the chat and is used to plot pie charts based on these parameters. To plot a line graph which shows author and message count of each date, to plot a line graph which shows author and message count of each author, Ordered graph of date vs message count, media sent by authors and their count, Display the message which is do not have authors, plot graph of hour vs message count.

This paper emphasizes on the usage of different classifiers like naves bayes classifier, Long Short Term Memory(LSTM), Random Forest Classifiers and the predictions of these methods are compared . Here we are using the concepts of Natural Language Processing(NLP). NLP is the field of Artificial Intelligence(AI) which deals with understanding and making insights from human speeches and texts. Sentiment Analysis is an application of NLP.

NLP basically has following parts :

## Lexical Analysis:

In this step we break the text into a series of tokens for the ease of analysis. Here the whole chunk of text into paragraphs is divided into paragraphs, sentences and words.

We can also say that words are the tokens of the sentence and sentences are the tokens of the paragraph. Here the mapping directory is created which maps each word with some unique index.

## Syntactic and Semantic Analysis:

In this step the arrangement of words in sentences are taken care such that the words are in their right order so that they make sense of the statement. It also checks if the words are grammatically correct or incorrect by converting the words to their root words.

To check the grammar there are two methods: stemming and lemmatization. In both of the methods we convert the words to their root form. For example, word ‘studying’ gets converted to ‘study’, word ‘playing’ gets converted to ‘play’.

Stemming is the process by which a word is lowered into grammatical categories to form its root word or stem (basic form). Following are the most famous stemmers in NLTK: Porter Stemmer, Snowball Stemmer, Lancaster Stemmer, Regex Stemmer. Stemming is a rule-based approach.

Lemmatization is similar to stemming but it is a dictionary-based approach but aims for morphological analysis of words and removing inflectional endings to get the root word or *lemma*.

The difference between stemming and lemmatization is that in stemming the root word may or may not make sense but in lemmatization the root word is always meaningful. Lemmatization comes at a higher cost of computation than stemming because it needs to know the context of the word before processing. It affects the accuracy of the analysis given that the meaning of the root word is significant in the process. Though Stemming has better recall, it hurts the precision.

## Discourse Integration:

It is considered as the larger context for any smaller part of Natural Language Structure. NL is so complex and most of the time, sequences of text are dependent on prior discourse. That’s why we are required to analyse the presence of immediate sentences or words so that we can analyse upcoming sentences or words in the paragraph.

## Model 1: Long Short-Term Memory (LSTM):

LSTM has an advantage over RNN. RNN is a supervised deep learning algorithm which has neurons and are interconnected with each other in layers. Here neurons pass information from the previous state to the next state. Hence at each epoch the information output from one neuron is taken as input for another.

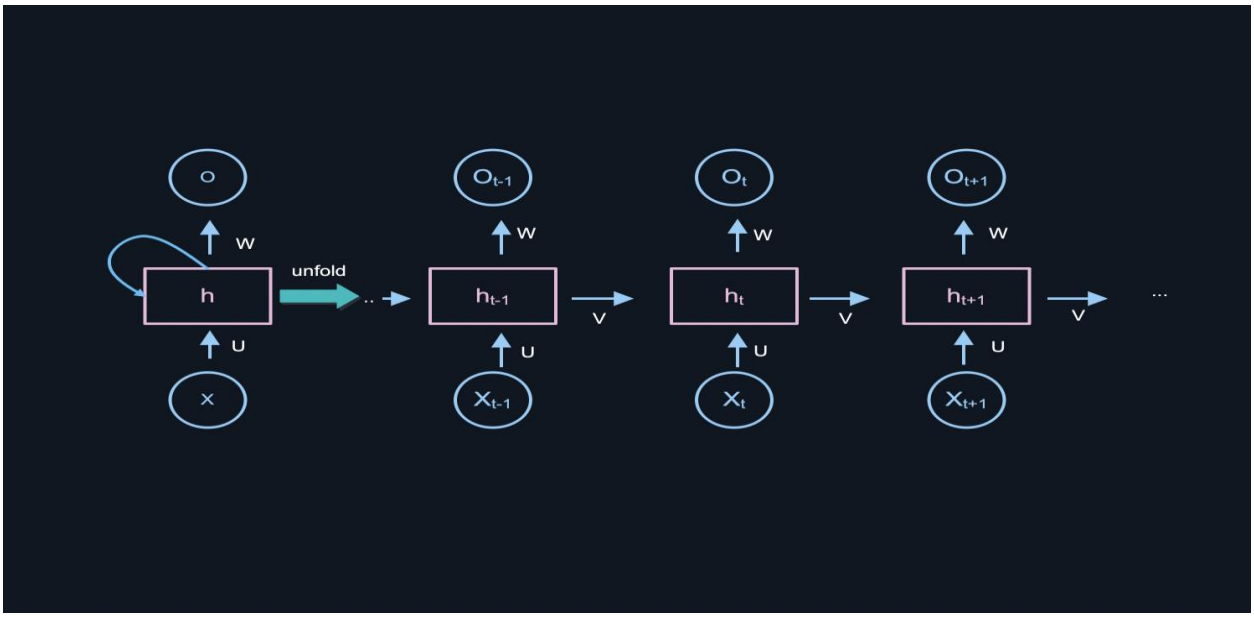


Figure 1 RNN Network

The problem with RNN is the vanishing gradient in back propagation which is weakening the gradient of weight calculation such that the new weight is equivalent to the previous weight. The problem lies in calculating these weights. If the gradient value is very small, then it won't contribute much to the learning process. Talking in normal terms, then RNNs in theory are fully capable of handling problems regarding long term dependency but in reality, it connects with more recent information.

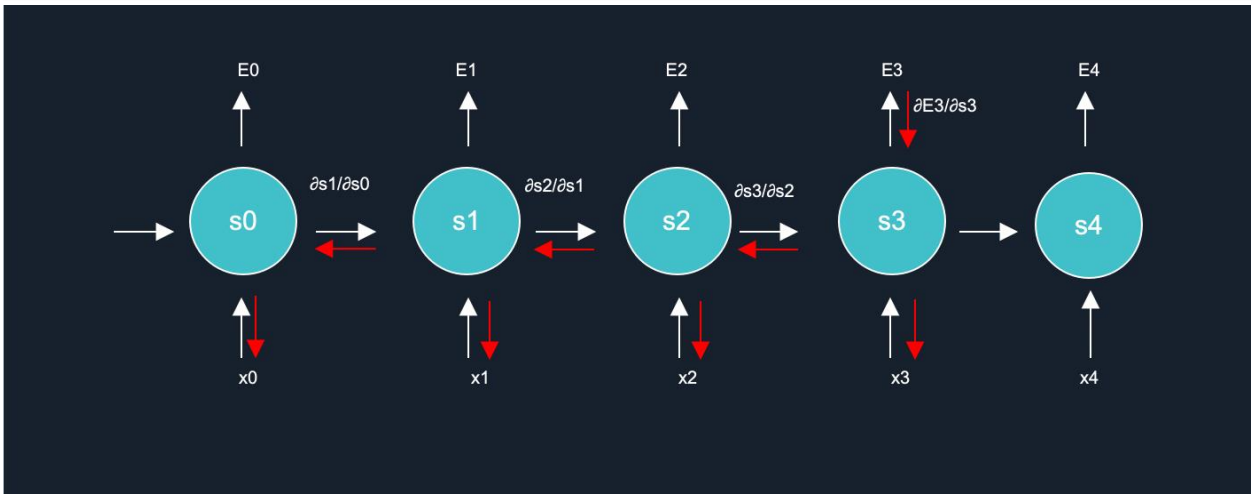


Figure 2 Process of back propagation

The weights are updated using the backpropagation calculus.

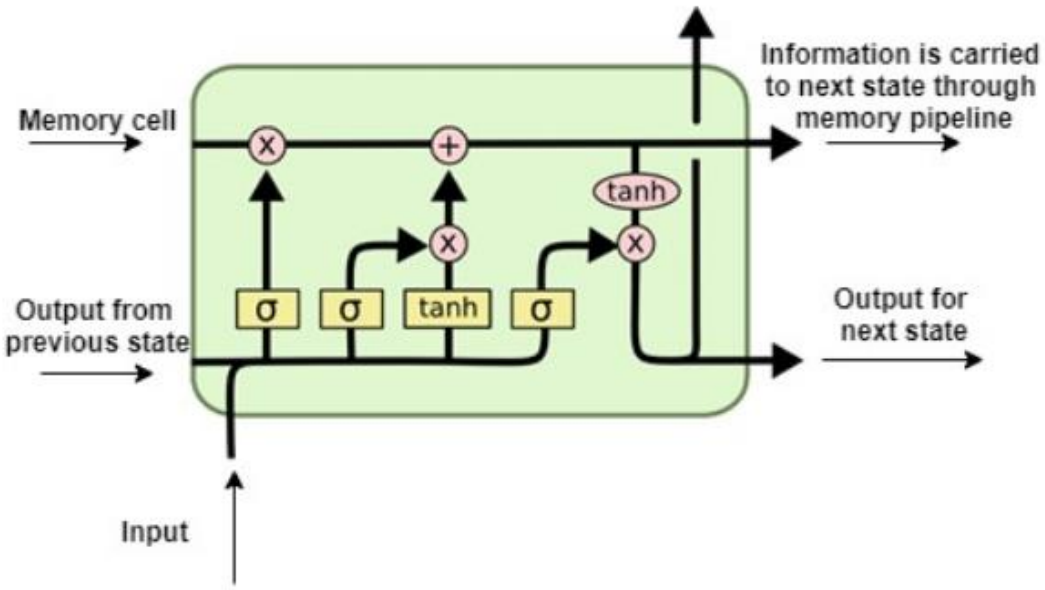


Figure 3 Basic LSTM model

Information is added/removed to the memory cells using valves. The memory cells carry the information from one epoch to another thereby retaining more information from previous states compared to RNN. In this way LSTM deals with vanishing gradient problem. The input for a LSTM layer is fed as input (at current time instance) and the output of hidden layer (previous time instance) and these pass-through valves and activation functions before reaching output.

## Model 2: Multinomial Naive bayes

Naive bayes classifier is a family of classifiers which works on a 'probabilistic model'. The model used here is Multinomial Naive Bayes. These classifiers used the concept of Bayesian probability with an assumption that each feature is conditionally independent with each other. Multinomial Naive Bayes is used in text classification. It is heavily used in industries as it is much faster than other algorithms. Moreover, it is easily scalable.

## Model 3: Random Forest

Random Forest is a supervised learning algorithm, it is used in regression and classification problems. Random Forest is simply called a collection of trees and each tree is different from one another. It constructs multiple decision trees and finally merges them together to gain absolute and stable value, which is mainly used at the time of training and outputting the class. There is a large dataset D consisting of m rows and d columns and since we are using Random Forest means that there will be multiple models of individual decision trees using the concept of Bagging, which means of using multiple machine learning algorithms.

Another thing to note about Random Forest, is that the decision trees consist of two properties – Low Bias and High Variance. Low Bias is defined as a concept where the decision tree created is completed to the end, the result will be more accurate and the training error will be very less.

As compared to Low Bias, High Variance is defined when we get a new test data and apply it on the decision tree, the decision tree has more tendency to give more error. Now, the Random Forest works well by deciding that each Decision Tree will have a high variance, as having new test data but when each of these Trees will be combined and vote will be for majority, the high variance for Decision Trees will be converted to Low Variance.

# 5. Methodology and Implementation

Initially the data is in *.txt* format which contains all the information like date, day, name of sender and message. But to extract the data efficiently we need to pre-process the data first.

Begin with separation of the messages using regular expression concepts. To separate messages, a pattern is written according to the data in the text file, Then pattern and text file is passed to the re.split function of the regex library which will separate the data into message, date and name of the sender. A check is performed if any message is really a message or some kind of notification. If it is a notification then drop it. Here, the separated data is stored into a dataframe, where the attributes of dataframe are named as "user" to store the sender's name, "user_message" to store the corresponding user's message and "data" to store the data of exchange of messages.

For the chat analysis following attributes are added and visualized: **word cloud** to see the most frequent words, Monthly and Daily Timeline to see the messages exchanged each month/day, Activity Map Weekly and Monthly to see the active users.

For the Sentiment Analysis, supervised learning is incorporated. For that a dataset containing 16000 messages for training and 4000 messages dataset for testing is used. All these data sets are in a csv file. At start the csv file is opened and then converted into a dataframe and separated the data frame into message and its corresponding emotion into two data frames named X and Y respectively, using the pandas library of python.

Then data preprocessing is performed on each message of training and testing datasets. For this nltk library is imported and then a lemmatizer object is created in order to perform lemmatization. Then for each message of the training and testing data set, for every word of the sentence, it is checked if it is contained in stopwords. If found, then drop the word else, lemmatize that word and append it into a list. Replace the earlier message with the newly created lemmatized list.

```
def lem(x): # lemmatizing the text
  corpus = []
  i=1
  for words in x:
    words = words.split()
    y = [wn.lemmatize(word) for word in words if not word in stopwords.words('english')]
    y =  ' '.join(y)
    corpus.append(y)
  return corpus
x = lem(df['sentence'])
```

Figure 4 Lemmatization

Next step is to import the tensorflow library to do embedding, tokenization and use pre implemented models. First tokenization of each word in the dataset is done and an upper bound is placed on the maximum number of words in the dictionary because of limited computational power. After tokenization, each word is replaced with its corresponding value in the dictionary. As it is known that to do the Embedding the length of each sentence must be the same. To do that,

if the length of any sentence is smaller than required then padding of zeros is done else truncate the sentence.

```
tokenizer = Tokenizer(num_words=10000, split=' ')
tokenizer.fit_on_texts(all)
def conv(all):
    X1 = tokenizer.texts_to_sequences(all)
    X1 = pad_sequences(X1, maxlen=20, padding='post', truncating='post')
    return X1
```

Figure 5 Tokenization

Now, the data-frame is ready for training the model.

## Model 1:

The Sequential LSTM model is used because the work to be done is in various layers, so it is the best choice. Then the embedding comes into play where embedding of each sentence is done, for the embedding layer the input dimension is the same as the size of the dictionary created above in the tokenization step and the output dimension is taken 64 here because for this particular problem it is giving the highest accuracy. Word Embedding is the language modelling process that converts texts into real numbers. Here each word is represented in dimensions, which shows how much is related to other words.

Next, the activation functions in the sequential layer are added which determines whether a neuron should be activated or not. Using mathematical operations, it decides if the neuron's input to the network is important in the prediction process or not. According to the intensity of the input signals it decides if it should be fired or not.

```
model = Sequential()
model.add(Embedding(input_dim=10000,output_dim = 64,input_length=20))
model.add(LSTM(64)) # 64 is number of neurons in the LSTM
model.add(Dense(6,activation='softmax'))
```

Figure 6: Sequential Layer and Activation Function

Then to get to results faster optimization functions are used. Optimizer functions are the algorithms used to change the attributes of the neural network like learning rate and weights which helps in reducing the loss and predicting the most accurate results possible.

```
model.compile(optimizer='rmsprop',loss='mse',metrics=['accuracy'])
```

Figure 7 Optimizer and Loss Function

Now the model is ready to get trained by passing the processed training set. After this the model is ready to predict

```
model.fit(X_train,Y_train,batch_size=32,epochs=10,verbose=2,validation_split=0.2)
```

Figure 8 LSTM Model Fit

Table 1:LSTM Model Statistics on different optimizer functions

| Optimizer Function | Loss Function | Epoch | Accuracy | Loss |
|---|---|---|---|---|
| rmsprop | mse | 10 | 0.8820 | 0.0317 |
| rmsprop | mse | 12 | 0.8875 | 0.0313 |
| rmsprop | mse | 15 | 0.8905 | 0.0312 |
| rmsprop | mse | 17 | 0.8815 | 0.0349 |
| adam | mse | 10 | 0.8915 | 0.0303 |
| adam | mse | 12 | 0.9020 | 0.0279 |
| adam | mse | 15 | 0.8980 | 0.0291 |
| adam | mse | 17 | 0.8972 | 0.0298 |

## Model 2

A Multinomial Naive Bayes was also used to predict the sentiment of the text messages. The pre-processing of data is done in a similar way. To check the accuracy of the model, the data is vectorized using two different vectorizers: tfidf and countVectorizer. Now using the sklearn library to import MultinomialNB.

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
```

Figure 9 Multinomial Naïve Bayes

The model is fit on the training data which has already been vectorized. Now the model is run on testing data. The accuracy turns out to be 71.0%.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vec = TfidfVectorizer(stop_words='english')
XTrain = vec.fit_transform(x).toarray()
XTest = vec.transform(x_test).toarray()
model.fit(XTrain, y)
y_pred = model.predict(XTest)
model.score(XTest, y_test)
```

```
0.71
```

Figure 10 Accuracy of Testing Data

An important observation is that when the vectorizer is changed to countVectorizer(), there is an additional increase in accuracy on the same testing data. The accuracy boosts upto 79.7%.

```
from sklearn.feature_extraction.text import CountVectorizer
vec = CountVectorizer(stop_words='english')
XTrain = vec.fit_transform(x).toarray()
XTest = vec.transform(x_test).toarray()
model.fit(XTrain, y)
y_pred = model.predict(XTest)
model.score(XTest, y_test)
```

```
0.797
```

Figure 11 Naïve Bayes Model Score and Accuracy

Table 2 Statics based on hyper-tuned Naïve Bayes Model with different vectorizers

| **TF-IDF** | Count Vectorizer |
|---|---|
| 71.0% | 79.7% |

## Model 3:

A Random Forest Model is also implemented. The preprocessing of data is done in a similar way. To check the accuracy of the model, the data is vectorized using two different vectorizers: tfidf and countVectorizer but it is prefered to use tf-idf Vectotrizer because TF-IDF is better than Countvectorizer because **it not only focuses on the frequency of words present in the corpus but also provides the importance of the words**. Removal of the words that are less important for analysis is carried out, hence making the model building less complex by reducing the input dimensions. Using the sklearn library to import ensemble. RandomForest Classifier, because random forest is an ensemble of many decision trees.

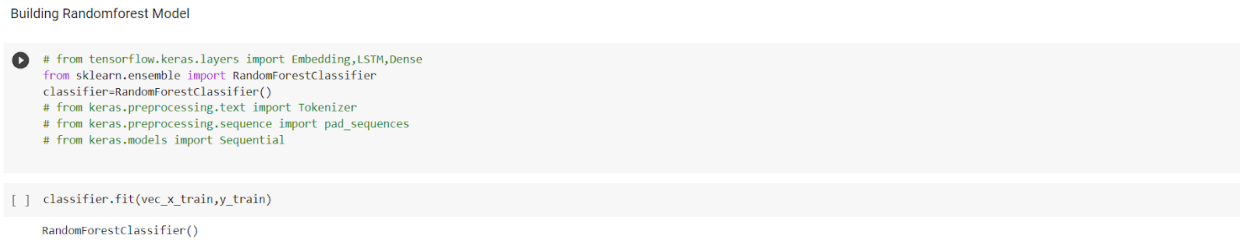


Figure 12 Random Forest Model

The model is fit on the training data which has already been vectorized. Now the model is run on testing data. The accuracy turns out to be 88.55%.


In this model, we apply here the best vectorized technique for the sentiment analysis, and also use precision and recall techniques for accuracy purposes.

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

```
print(confusion_matrix(y_test,prediction))
print(classification_report(y_test,prediction))
print(accuracy_score(y_test,prediction))
```

```
[[248   8   8   1   9   1]
 [  6 196   1   1  12   8]
 [  8   5 644  24   7   7]
 [  2   0  42 112   1   2]
 [ 13   9  21   5 531   2]
 [  0  14  12   0   0  40]]
              precision    recall  f1-score   support

       anger       0.90      0.90      0.90       275
        fear       0.84      0.88      0.86       224
         joy       0.88      0.93      0.91       695
        love       0.78      0.70      0.74       159
     sadness       0.95      0.91      0.93       581
    surprise       0.67      0.61      0.63        66

    accuracy                           0.89      2000
   macro avg       0.84      0.82      0.83      2000
weighted avg       0.88      0.89      0.88      2000

0.8855
```

Figure 13 Metrics of Random Forest Classifier

# 6. TECHNOLOGY USED AND SYSTEM REQUIREMENT

## Technology Used :

**Python**: It is an interpreted, high-level general-purpose programming language. Created by Guido Van Rossum and first released in 1991. Its language constructs and objects-oriented approach aim to help programmers with clear, logical code for small and large-scale tools. Python is used for web development (server-side), software development, mathematics, it can be used alongside software to create workflows, it can connect to database systems, it can also read and modify files, it can be used to handle big data and perform complex mathematics and can be used for rapid prototyping, or for production-ready software development.

**Pandas**: It is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open-source data analysis / manipulation tool available in any language.It can be used for converting a list into a 2d matrix data in table form.

**NLTK**: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, Corpus, Stopwords , WordLemattizer along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

**Tensorflow**: TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications. TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. TensorFlow provides stable Python and C++ APIs, as well as non-guaranteed backward compatible API for other languages.

**Keras**: Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.It is an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and giving efficient,fast solutions to machine learning problems.

**Jupyter Notebook**: Jupyter notebook is an easy-to-use, interactive data science environment which works as an IDE and helps in graphs, plots, visualizations in its notebook called jupyter notebook. It runs the code in segments which helps in easily error solving, code manipulation and it also increases readability of the code.

**Streamlit**: It turns data scripts into shareable web apps in minutes. It’s all Python, open-source, and free! And once the app is created it can also be deployed on the cloud platform.

**Seaborn**: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

## System Requirement :

Minimum of 4GB RAM with Intel i3 processor is required to run this project.

# 7. Conclusion

In conclusion, it can be said that the capabilities of the WhatsApp application and the power of the python programming language in implementing whatever network data analysis intended, cannot be overemphasized. This work was able to discuss the WhatsApp application and its libraries, to create an analysis of a WhatsApp group chat and visually represent the top 10 and top 20 users in the chat groups. A pseudocode of the plot was given and at the end, visual representation of the plot was implemented. Also, an analysis of the top 10 and top 20 users were done. The system was done with python, and the python libraries that were implemented includes, NumPy, Pandas, Matplotlib and Seaborn. At the end of the work expected results were obtained and the analysis was able to show the level of participation of the various individuals on the given WhatsApp group. On a serious note this system has the ability to analyse any WhatsApp group data input into it.

# 8. References


[1] Suppala K, Rao N (2019) Sentiment analysis using naïve bayes classifier. Int J Innov Technol Explor Eng 8(8):246–269

[2] Liu B (2012) Sentiment analysis and opinion mining. Morgan & Claypool, Williston

[3] Arora T, Dhawan S, Singh K (2016) Sentiment analysis of online movies' reviews using Improved k-Nearest Neighbor Classifier. Adv Comput Sci Inf Technol (ACSIT) 3(4):241–245

[4] Zainuddin N, Selamat A (2014) Sentiment analysis using a support vector machine. In 2014 International Conference on Computer, Communications, and Control Technology (I4CT), pp 333–337. IEEE

[5] Mikolov T, Karafiat M, Burget L, Cernocký J, Khudanpur S (2010) Recurrent neural network based language model. Interspeech 2:3

[6] Legrand J, Collobert R (2016) Deep neural networks for syntactic parsing of morphologically rich languages. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, no.CONF

[7] Hemalatha I, Saradhi Varma GP, Govardhan A (2013) Sentiment analysis tool using machine learning algorithms. Int J Emerg Trends Technol Comput Sci (IJETTCS) 2(2):105–109

[8] INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019, ICRTAC 2019 Analysis of Classifiers for Fake News Detection Vasu Agarwal , H.Parveen Sultana ,Srijan Malhotra , AmitrajitSarkarb

[9] Sentiment analysis on large scale Amazon product reviews, IEEE International Conference on Innovative Research and Development (ICIRD)

[10] Lv S, Zhang H, He H, Chen B (2020) MicroblogRumor detection based on comment sentiment and CNN-LSTM. J Artif Intell China.

[11] Jin Z, Yang Y, Liu Y (2019) Stock closing price prediction based on sentiment analysis and LSTM.

[12] Xianghua Fu, Yang J, Li J, Fang M, Wang H (2018) Lexicon-enhanced LSTM with attention for general sentiment analysis. IEEE Access.

[13] Guixian Xu, Meng Y, Qiu X, Ziheng Yu, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. IEEE Access.

[14] Graber F, Kallumadi S, Malberg H, Zaunseder S (2018) Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In: Proceedings of the 2018 International Conference on Digital Health, pp 121–125

[15] Da'u A, Salim N (2019) Sentiment-aware deep recommender system with neural attention networks. In: 2169–3536 2019 IEEE. Translations and content mining, vol. 7.

[16] Sailunaz K, Alhajj R (2019) Emotion and sentiment analysis from Twitter text. J Comput Sci.

[17] Ahamed S, Danti A (2018) Effective emoticon based framework for sentimental analysis of web data. In: International Conference on Recent Trends in Image Processing and Pattern Recognition. Springer, Singapore, pp 622–633

[18] Shaukat Z, AhadZulfiqar A, Xiao C, Azeem M, Mahmood T (2020) Sentiment analysis on IMDB using lexicon and neural networks. SN Appl Sci 2(2):1–10

[19] Ankita, Saleena N (2018) An ensemble classification system for Twitter sentiment analysis. In: Peer-review under responsibility of the scientific committee of the International Conference on Computational Intelligence and Data Science. Published by Elsevier Ltd.