**An Industry-Oriented Mini Project on**

**WhatsApp Chat Analysis using Machine Language**

A report submitted in partial fulfillment of the requirements for the award of

The B. Tech Degree

By

D Kranthi Kumar (20EG105109)

M Vedansh Reddy (20EG105118)

J Naveen Kumar (20EG105138)

Thoundur Preethi (20EG105151)

**Under the Guidance of**

Mr. P Rajasekhar Reddy

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ANURAG UNIVERSITY**

**VENKATAPUR – 500088**

**TELANGANA**

**YEAR 2023-24**

**DECLARATION**

We hereby declare that the Report entitled “WhatsApp Chat Analysis using Machine learning” submitted for the award of Bachelor of technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Anurag University, Hyderabad

D Kranthi Kumar

(20EG105109)

M Vedansh Reddy

(20EG105118)

J Naveen Kumar

(20EG105138)

Thoundur Preethi

(20EG105151)

**CERTIFICATE**

This is to certify that the Report / dissertation entitled “WhatsApp Chat Analysis using Machine Learning” that is being submitted by D Kranthi Kumar (20EG05109), M Vedansh Reddy (20EG105118), J Naveen Kumar (20EG105138), Thoundur Preethi (20EG105151) in partial fulfilment for the award of B. Tech in Computer Science and Engineering to the Anurag University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this Report have not been submitted to any other university or Institute for the award of any degree or diploma

.

**head of the department**

**Signature of Supervisor**

**P Rajasekhar Reddy**

**(ASSISTANT PROFFESOR)**

**ACKNOWLEDGMENT**

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mr. P Rajasekhar Reddy** (Assistant Professor) for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like express my special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for their encouragement and timely support in

our B. Tech program.

We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. We also express my deep sense of gratitude to **Dr**. **V V S S S Balaram,** Academic Coordinator, **Dr. Pallam Ravi**, Project In-Charge, **Dr. G Prabhakar Raju** Project Coordinator and Project Review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage our project work.

**ABSTRACT**

WhatsApp has become a popular platform for communication, and its chat data contains a wealth of information that can be leveraged for various purposes. Users can use this platform to provide the chat analyzer with exported WhatsApp (.txt file) as input.

The objective of this research is to explore the application of NLP algorithms to extract meaningful insights from WhatsApp conversations. It provides a precise picture of total number of messages, activity of each person, total words, shared links, busiest day and more. These techniques enable the identification of key patterns, trends, and themes within the conversations. The analysis focuses on several aspects. Firstly, sentiment analysis is performed to evaluate the overall sentiment of the chats, allowing for an understanding of the emotional tone of the conversations. Secondly, named entity recognition is used to identify important entities mentioned in the chats, such as people, and dates. The results of the analysis shed light on various aspects of the WhatsApp chat data. They reveal the prevalent sentiment within the chats, the frequently mentioned entities. They can be leveraged for customer feedback analysis, social media monitoring, market research, and understanding user behavior. The insights derived from the WhatsApp chat analysis using NLP techniques can aid in decision-making processes, enhance communication strategies, and provide valuable feedback for organizations and individuals.

Overall, this study demonstrates the effectiveness of NLP techniques in analyzing WhatsApp chat data, enabling a deeper understanding of the conversations and extracting meaningful insights. We are developing an application using Machine Learning and NLP, which analyzes the WhatsApp conversation and makes appropriate recommendation.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 1 | Importing Modules | 7 |
| 2 | Importing Data | 8 |
| 3 | Splitting the Data | 8 |
| 4 | Converting Data into Data Frames | 9 |
| 5 | Splitting the Messages | 9 |
| 6 | Extracting unique user’s data | 18 |
| 7 | Pie Chart for unique users and messages sent | 19 |
| 8 | Data Analysis | 19 |
| 9 | Pie Chart for Data Analysis | 20 |
| 10 | Word Cloud Analysis | 21 |
| 11 | Analysis based on Date Time | 22 |
| 12 | Graph on Monthly Peaks | 23 |
| 13 | Timeline based analysis | 23 |
| 14 | Heatmap of a period of a day | 24 |
| 15 | Importing Modules | 24 |
| 16 | Extracting the Data | 25 |
| 17 | Bar Graph based on emotions | 26 |
| 18 | Extract sentiment of data | 26 |
| 19 | Emotion and sentiment relation | 27 |
| 20 | Bar graph emotion and sentiment relation | 27 |
| 21 | Data Cleaning | 28 |
| 22 | Model Creation | 29 |
| 23 | Model Prediction | 30 |
| 24 | Data Visualization | 30 |
| 25 | Users Usage Chart | 32 |
| 26 | Day-Wise Usage chart | 32 |
| 27 | Month-Wise Usage chart | 33 |
| 28 | Emotion Detection Chart | 33 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TABLE NAME** | **PAGE NO.** |
| 1 | Test Cases and Results | 34 |

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **CONTENT** | **PAGE NO.** |
| 1 | Introduction  1.1 Project Introduction  1.2 Scope  1.3 Project Overview  1.4 Objectives | 8  8  9  9 |
| 2 | Literature Review  2.1 Existing System  2.2 Proposed System | 10  11  11 |
| 3 | Implementation and Results  3.1 Language Used  3.2 Steps Involved in Implementation  3.2.1 Constructing Dataset  3.2.2 Adding Libraries  3.2.3 Data Preprocessing  3.2.4 Model Creation  3.2.5 Data Visualization  3.3 Sample Code  3.4 Emotion Detection  3.5 Output Screens | 12  12  12  13  13  13  14  14  24  32 |
| 4 | Testing: Test Cases and Results | 34 |
| 5 | Conclusion and Future Scope | 36 |
| 6 | References | 37 |

1. **INTRODUCTION**

**1.1 Introduction**

One of the most widely used instant messaging services globally is WhatsApp. With over 2 billion active users, it has become a significant source of communication between people. WhatsApp chat analysis involves the examination of the content of these conversations, the participants involved, and the patterns of communication. One of the primary uses of WhatsApp chat analysis is for research purposes. Researchers can gather data from WhatsApp conversations to analyse various aspects of human behaviour, such as language use, social dynamics, and communication patterns. By studying WhatsApp chats, researchers can gain insights into the way people interact with each other, how they express themselves, and how they form relationships. Another use of WhatsApp chat analysis is in the business world. Companies can use WhatsApp to communicate with their customers and gather valuable feedback. By analysing these conversations, businesses can better understand their customers' needs and preferences, and tailor their products or services accordingly. Additionally, companies can monitor employee WhatsApp conversations to ensure that they comply with company policies and regulations. WhatsApp chat analysis can also be useful in personal relationships. By examining WhatsApp conversations, individuals can gain insight into their communication patterns with their friends or family members. They can identify areas where they may need to improve their communication skills or areas where they need to work on building stronger relationships. WhatsApp chat analysis is a valuable tool for researchers, businesses, and individuals. By examining WhatsApp conversations, one can gain valuable insights into human behaviour and communication patterns. Whether for research, business or personal use, analysing WhatsApp chats can provide valuable information for improving relationships and achieving better communication.

**1.2 Project Scope**

Dealing with huge data such as business chats or organizational group chats might be difficult. We can overcome this problem by using Python Programming alongside the Machine Learning algorithms for faster analysis and for accurate insights of the chats. Machine Learning is becoming increasingly important for WhatsApp chat analysis due to its ability to automate and streamline analysis, provide more accurate insights, and enable personalized and predictive analysis. As WhatsApp continues to be a popular platform for communication, the use of ML for chat analysis will likely become even more critical for businesses and individuals alike. We have used Python Programming for data analysis and Machine learning for detecting the emotion of the individual from their chats.

**1.3 Project Overview**

With the help of Python Programming, we have pre-processed the exported WhatsApp chat file. We have used pandas to convert the chat data into data frames. Using the data frames, we have analysed the timelines, count of messages and count of users. We used the external stop words dataset to remove the commonly used words for emotion detection. We later detected the emotion of the individual or group by using Sentiment analysis.

**1.4 Objectives**

The main objective of the project is to analyse thousands of WhatsApp chats in minutes. This project helps the business analysts to understand their customer behaviour and get the insights of the product or the services offered by them. In today's world WhatsApp has become the major media for communication be it for personal, professional or business purposes. It has become a great media for crimes as well which needs to be monitored. With our project we can identify the suspects based on their chat behaviour. It also helps to prevents suicides and find the people suffering with depression in life.

The Objectives of this project are:

•The main objective of this project is to develop a WhatsApp Chat Analyzer; the system can extract the words and emojis to predict the emotions.

•This system aims to exploit machine learning techniques to assist in the prediction of emotions and recommendations.

•Easy to analyse the WhatsApp chats and provide personalized recommendations.

•To reduce the time to analyse the WhatsApp chats.

•To detect any crime activities in prior.

**2. LITERATURE REVIEW**

● Wang, Yongming. "Using Machine Learning and Natural Language Processing to Analyse Library Chat Reference Transcripts." Information Technology and Libraries 41, no. 3 (2022). Improves library services by applying AI and machine learning techniques to library data. Chat reference in libraries generates a large amount of data in the form of transcripts. This study uses machine learning and natural language processing methods to analyse one academic library’s chat transcripts over a period of eight years. The built machine learning model tries to classify chat questions into a category of reference or nonreference questions. The purpose is to predict the category of future questions by the model with the hope that incoming questions can be channelled to appropriate library departments or staff.

● Ahmad, Zishan, Raghav Jindal, Asif Ekbal, and Pushpak Bhattacharyya. "Borrow from rich cousin: transfer learning for emotion detection using cross lingual embedding." Expert Systems with Applications 139 (2020): 112851. This paper proposes an efficient technique to mitigate the problem of resource scarcity for emotion detection in Hindi by leveraging information from a resource-rich language like English. The model follows the methods of  a [deep transfer learning](https://www.sciencedirect.com/topics/computer-science/deep-transfer-learning) framework which efficiently captures relevant information through the shared space of two languages, showing significantly better performance compared to the monolingual scenario that learns in the vector space of only one language. And uses CNN and Bi-LSTM as base learning models.

● Akhilesh Kumar, Bhavna Bajpai, Rachit Adhvaryu, Suthar Dhruvi Pankaj Kumar, Prajapati Parth Kumar Gordhanbhai, and Atul Kumar. "An Efficient Approach of Product Recommendation System using NLP Technique." Materials Today: Proceedings (2021). This is a system, which can recommend the products which are like the searched products. This will help the consumer to find out another product in case the item is unavailable, or the searched product is not good enough, or when they would like to look through different similar products. A good recommendation system has been found out to be financially beneficial for the companies also. It is found out that consumer is 35% more likely to buy a product if the recommendation is good enough for consumers. NLP technologies and CNN to help in predicting similar products. CNN used at last to create a feature vector from product images, and use this vector combined with all the other vectors, for prediction. VGG-16 architecture used to extract the features from the images.

● Shashank and Pushpak Bhattacharya. “Emotion Analysis of Internet Chat”. A system for Emotion Analysis of Instant Messages (IM). Using Instance Based classifier we have shown that our system can outperform similar systems in the IM domain. Tagged instant messages and elaborate feature engineering can help a lot in increasing the performance of text classification of unstructured, ungrammatical text. The impact of class imbalance on classification has been studied and demonstration has been made of how under sampling can help mitigate this problem.

**2.1 Existing System**

In olden days there was no analysis for WhatsApp chat. One had to look after each and every message from the chat. We didn’t have any system which could analyse thousands of messages and extract emotion from it. There isn't a CSV file available for analysis if someone wants to do it. WhatsApp Application provides an export txt file which is in raw format. It is very complicated for analysis. Therefore, we must disregard that system and instead use the WhatsApp Chat Analyzer.

Disadvantages of Existing system:

● It was time consuming

● Required man power

● Raw data

● Difficult to analyse

● Emotion could not be extracted

**2.2 Proposed system**

The “WhatsApp Chat Analysis Using Machine Learning” provides a platform to the user which enables users to analyse WhatsApp chats. We can get the insights of the chat. It becomes easy to understand the data with the help of the graphs and charts which are used to represent the data. It reduces a lot of time for analysis. Most importantly our system helps to detect the emotion of the individual by extracting the words used and running them through the Machine learning model that we have developed. It provides an upper hand to the businesses by making it easy for them to understand the customers, gaining insights into the customers interests helping them to increase their repeated customers count. It also provides the typical information such as the number of users, Number of chats, busiest days, most active hours, most active users.

Advantages of Proposed System:

● Compatible with all devices

● Time efficient

● Easy to use

● Emotion detection

● Free to use

**3. IMPLEMENTATION AND RESULTS**

**3.1 Languages Used**

1. **Python**: Python is a high-level, interpreted programming language that Guido van Rossum developed and was made available in 1991. Since then, it has grown in popularity and is now one of the most widely used programming languages, with uses ranging from web development to data analysis and scientific computing. Python's simplicity and usability are among its most important characteristics. The syntax is simple and straightforward, making the language easy to learn and use. It is also designed to be readable and intuitive. With a sizable standard library that offers a wide range of built-in functions and modules for a number of applications, Python is also incredibly adaptable. Because Python is an interpreted language, code can be run without first needing to be compiled.

2. **Machine Learning**: In the quickly expanding subject of machine learning, algorithms and models are created that can recognise patterns in data and predict outcomes without explicit programming. Python is a well-known programming language with a large community of tools and libraries that makes it ideal for machine learning. Python offers a variety of potent machine learning libraries and frameworks, such as NumPy, SciPy, Pandas, Scikit-learn, and TensorFlow. In terms of data preprocessing, feature extraction, model selection, and model evaluation, these technologies offer a wide range of functionalities. With support for massive, multi-dimensional arrays and matrices as well as a variety of mathematical operations, NumPy is a well-known Python library for numerical computing. A library called SciPy, which is based on NumPy, adds support for operations like optimisation, integration, and interpolation in scientific computing.

**3.2 Steps involved in Implementation**

**3.2.1 Constructing Dataset**

A dataset of words and their emotions must first be compiled. This can be done by observing the terms that are frequently used in the chats and assigning the appropriate emotion to them. We need a diverse set of terms and their spelling in order to obtain representative data. It is appropriate to refer to numerous chats with other people to gain a thorough understanding of the words, sentences, and emotions. Before analysing the chat, we must also make a list of stop words, such as articles, punctuation, and prepositions, to remove them from the conversation.

**3.2.2 Importing Libraries**

In this project, we're utilizing a variety of libraries, including TensorFlow, Matplotlib, NumPy, SK-Learn and Pandas is used to extract the chat and transform them into data frames.

**TensorFlow**: TensorFlow is an open-source machine learning and artificial intelligence software library. It was developed by Google and is used for a variety of machine learning operations, including image classification, natural language processing, and predictive modelling.

**Matplotlib**: Matplotlib is a Python module for data visualization. It is used to deliver dynamic, animated, and interactive visualizations in Python. With Matplotlib, you can create a wide range of visualizations, including line plots, scatter plots, bar plots, error bars, histograms, bar charts, pie charts, box plots, and more.

**Pandas**: An open-source Python library for data analysis and manipulation is called Pandas. It offers tools for data analysis and working with structured data that are simple to use. Series, a one-dimensional labelled array, and DataFrame, a two-dimensional labelled array, are the two primary data structures in Pandas. For activities including data wrangling, data cleansing, data exploration, and data analysis, Pandas is frequently employed. As well as handling missing data, reshaping and pivoting data, and merging and joining datasets, it offers functionality for these tasks. In a variety of disciplines, including finance, economics, social sciences, and more, Pandas is a popular choice for data analysis.

**3.2.3 Data Pre-Processing**

The user inputs the Exported WhatsApp chat text file to the system. This text file consists of raw data with date, usernames, message, media data and group notification messages. The main objective of this data processing is to split the data into different categories and represent the whole set of data separately. We use the data frames to represent the data where each column consists of a particular set of data. The stop words are removed from the messages before inputting the words into the machine learning model for the emotion detection. The words other than the stop words which help in detecting the emotion of the individual user are recorded into a text file.

**3.2.4 Model Creation**

The machine learning model is created using the dataset of the words and the emotions. We use the Multinomial Naive Bayes theorem to build the model which is trained on 70% data of the dataset and is tested on the remaining 30% of the data. This model takes the text file created and runs the naive bayes algorithm in order to predict the emotion from the trained data and later predicts the overall emotion of the user and outputs the results to the user in the form of pie chart.

**3.2.5 Data Visualization**

The data which is analysed such as the busiest days, most active users, count of messages, count of media messages, top active users, emotion of the individual etc are given as a result to the users in the form of graphs and charts.

**3.3 Sample Code**

**1. Importing Required Libraries:**

**Import re**: The functions in this module allow you to determine whether a given string matches a given regular expression.

**Import os**: Python offers a mechanism to communicate with the underlying operating system using its "os" module. It offers tools for managing processes, environment variables, and many other things in addition to working with files and directories.

**Import pandas as pd**: This library provides data structures to deal with different types of data.

**Import matplotlib. pyplot as plt**: This library helps in data visualization like plotting graphs, bar charts.

**Import numpy as np**: This library provides function to deal with arrays in python

**Import url extract**: This library provides functions to deal with URL and work on the URL’s.

**Fig. 1 Importing Libraries**

**2. Importing a Data file:**

**f= open(‘data/WhatsApp chat with WAGDemoGroup (4).txt’,’r’,encoding=’utf-8’):** This line opens the file in the specified location in the read mode and assigns the it to the variable f

**data=f.read():** This line reads the data from the file and assigns to ‘data’ variable

**print(data):** This line displays the data read from the file.

** Fig.2 Splitting the Data**

**3. Data Pre-processing:**

**pattern='\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s-\s':** This line helps to create a pattern to split the data from the imported data

**messages = re.split(pattern,data)[1:]:** This line helps to split the data according to the pattern

**dates=re.findall(pattern,data**): This line helps to find all the data which satisfies the pattern.

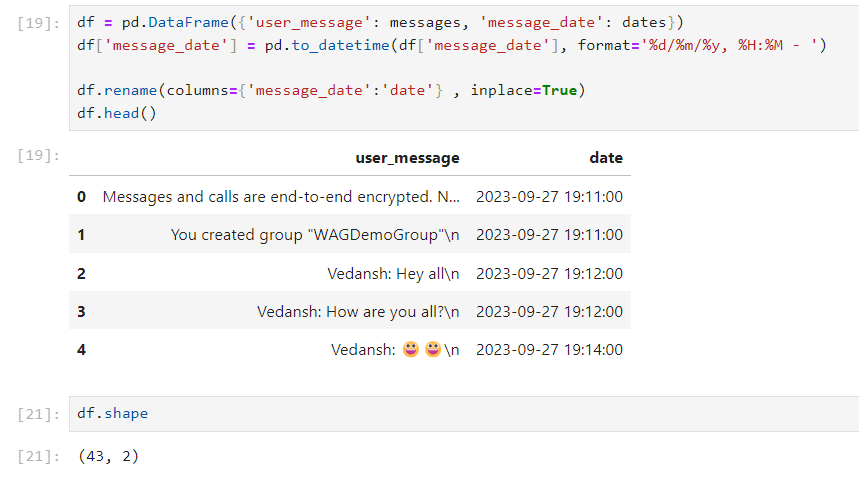
**df = pd.DataFrame({'user\_message': messages, 'message\_date': dates}):** This line helps to convert data to data frames to access and modify the data easily

**df['message\_date'] = pd.to\_datetime(df['message\_date'], format='%d/%m/%y, %H:%M - '):** This line helps to convert the datetime data into proper sequence.

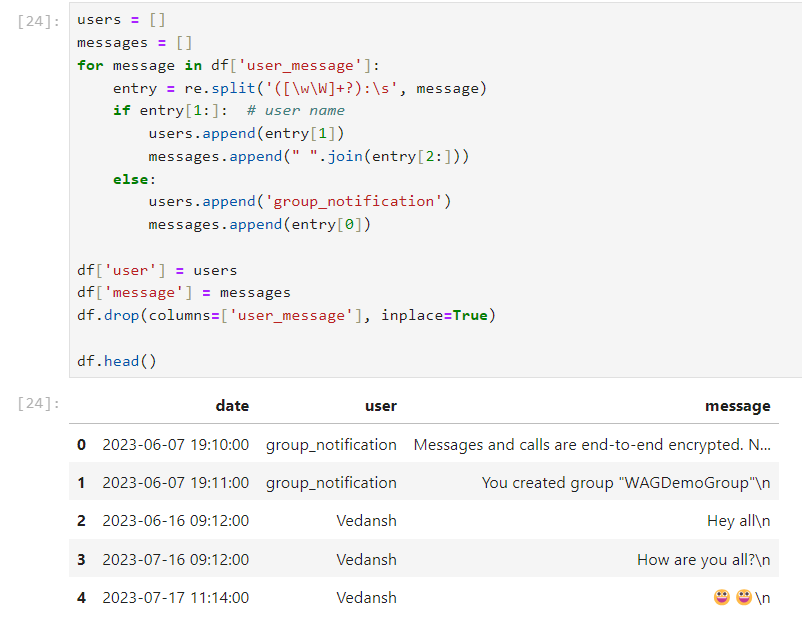
**df.rename(columns={'message\_date':'date'} , inplace=True):** This line helps to rename the columns in the data frame

**df.head()** : This line is used to display the first few rows in the data frame

**df.shape :** This line shows the size of the data frames as rows and columns

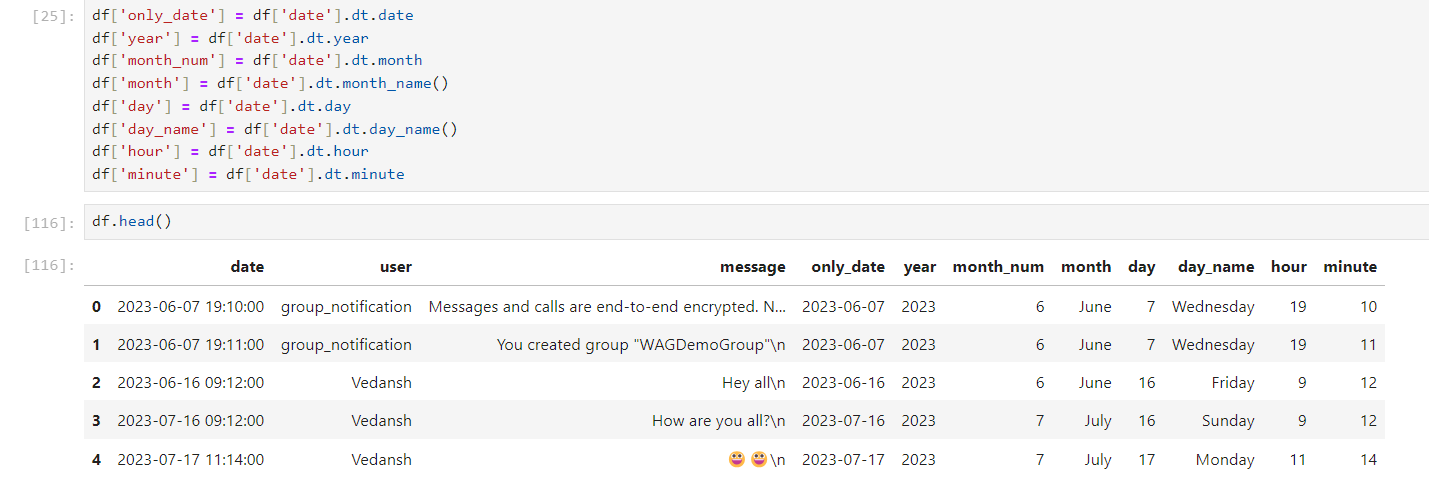


**Fig.3 Converting data into data frames**

****

**Fig.4 Splitting the messages**

The above code (Fig.4) helps to split the data into user names and the message sent by the user. We create two columns in the data frame namely user and message and assign the respective values obtained from splitting the data.

****

**Fig.5 Converting datetime data**

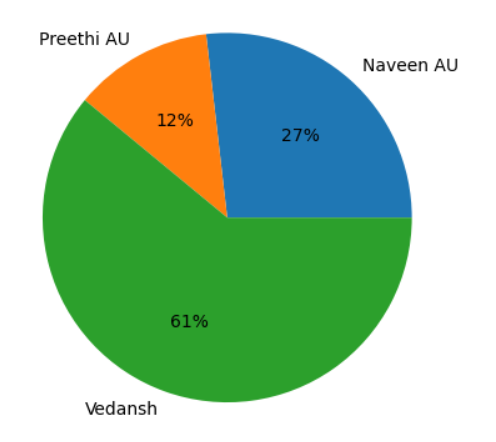
The above code is used to create separate columns for date, year, month, month number, day, day\_name, hour and minute in the data frame for further analysis.

**4. Data Analysis**

****

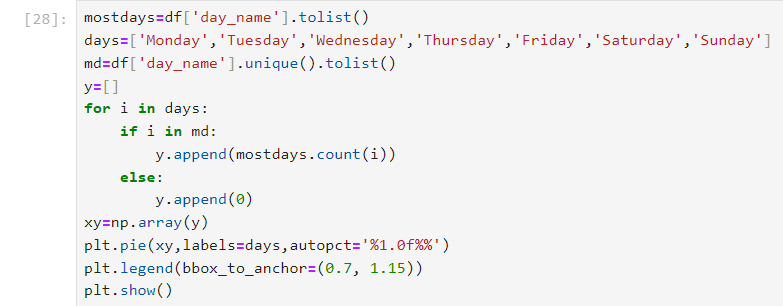
**Fig. 6 Extracting unique users and data**

The above code helps to identify the unique users and extract the total number of messages sent by each user and is displayed using the data frame’s head function. We further plot the pie chart to gain the insights of the busiest user.

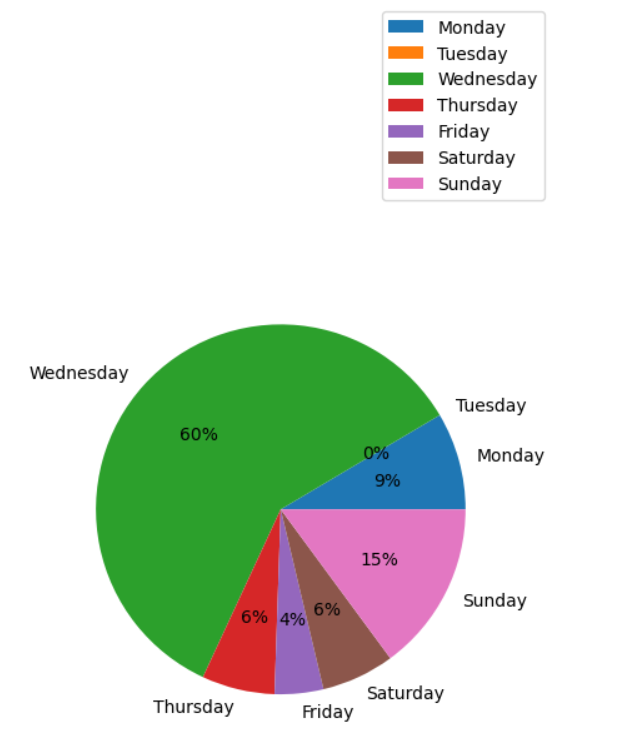
****

**Fig. 7 Pie chart of Unique users and messages sent**

The below code enables us to analyse the chat data based on the dates. Here we can extract the busiest days and number of messages sent on each day of a week. We then plot a pie chart based on the percentage of messages on each day in a week.

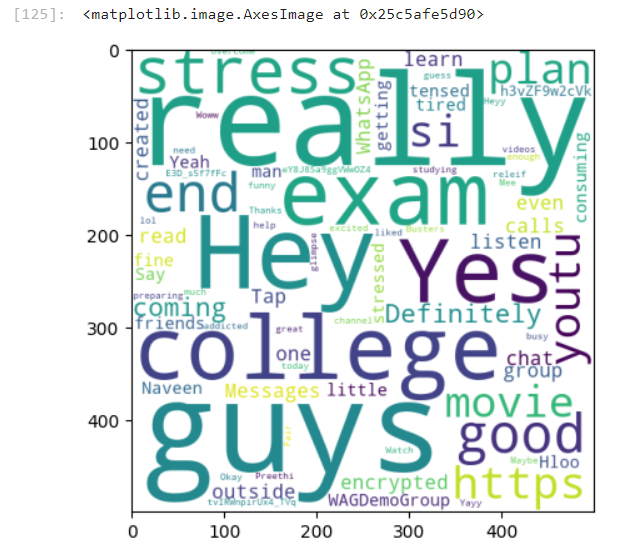
****

**Fig. 8 Daily Analysis**

****

**Fig. 9 Pie Chart of Daily Analysis**

****

****

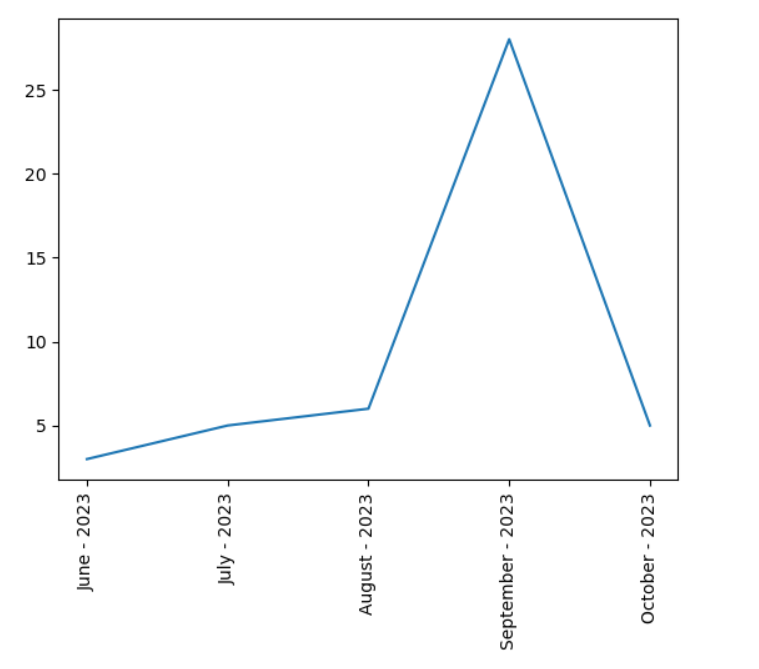
**Fig. 10 WordCloud Analysis**

The above code splits the messages into words . We use the ‘stop words.txt’ data set to remove the commonly used stop words from the messages . Then we use the WordCloud function from the module wordcloud and create a word cloud which represents the most commonly used words in the chats .The generate function in the word cloud is used to generate the word cloud chart . We then display the word cloud chart using the image show function .

****

**Fig. 11 Analysis based on Datetime**

In the above code we extract the date year and month data to analyse the chats based on the months. We plot a timeline graph in order to get monthly usage of WhatsApp chats.

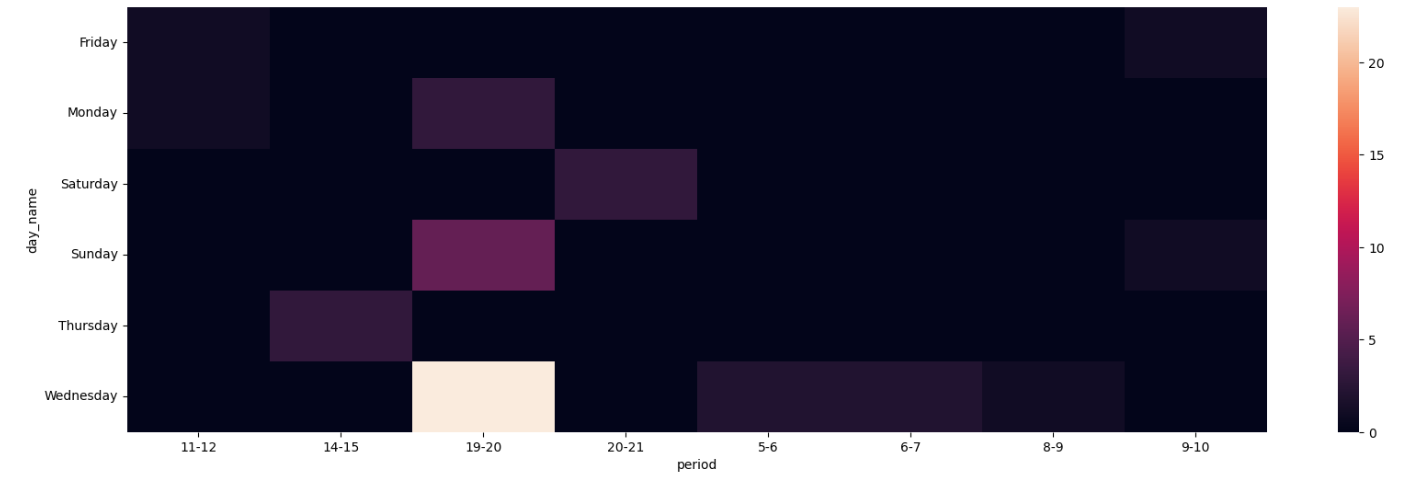
****

**Fig. 11 Graph on Monthly Peak**

The below code helps to divide the messages based on time. We divide the timelines of the messages into a fixed frequency and then plot a heatmap in order to get insights of the data based on timelines of a particular day in a week.

****

**Fig. 12 Timeline based Analysis**

****

**Fig. 13 Heatmap of a period of a day**

**3.4 Emotion Detection**

****

**Fig. 14 Importing Modules**

**1. Importing the modules:**

**Import pandas:** This library helps to deal with huge amounts of data by providing special data structures

**Import numpy as np**: This library helps to deal with arrays and for converting the data into arrays

**Import seaborn as sns**: This library provides functions for data visualization using the data frames

**Import Sklearn**: This library provides the machine learning functions to create a model.

**2. Data Extraction:**

**variable ‘data data\_set = "data/emotion\_dataset\_2.csv" :** This line assigns the path of the data file to the \_set’

**df = pd.read\_csv(data\_set) :** This line is used to read the csv file from the path assigned to ‘data\_set’ into a data frame

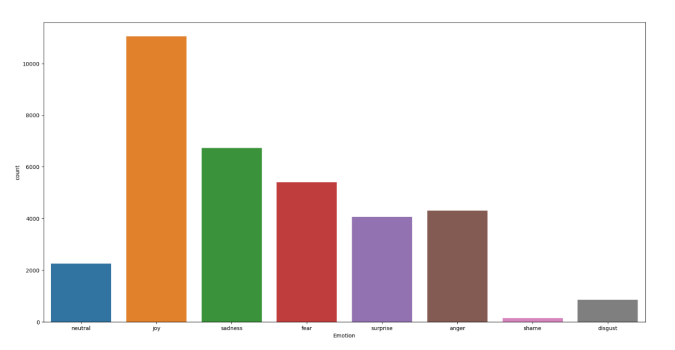
**df.shape :** It provides the size of the data frame

**df.dtypes :** This line returns the list of datatypes of all the columns of the dataframe.

****

**Fig. 15 Extracting the Data**

The above code gives us the data which is grouped based on different emotions. We can also plot the Bar graph using the data from the data frames. We use the seaborn library’s function countplot in order to plot a graph for the count of number of records of each emotion .

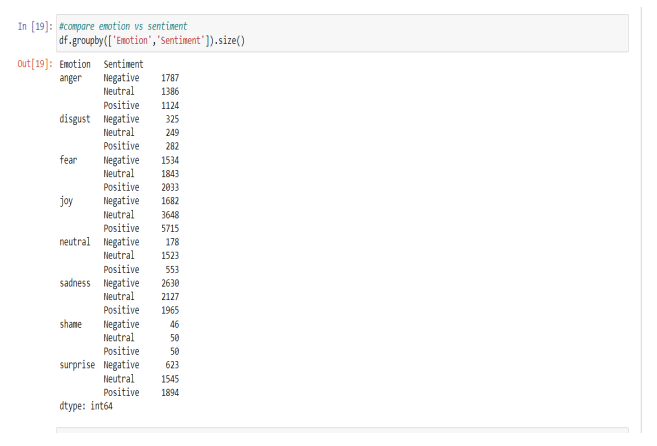
****

**Fig. 16 Bar graph of data based on emotions**

The below function ‘get\_sentiment’ is used to extract the sentiment of the input. The sentiments are of three categories namely Positive, Neutral and Negative. We use the TextBlob function to get the polarity of the sentence. If the polarity is greater than 0 then it is considered Positive, if the polarity is less than 0 then it is considered Negative and if the polarity is 0 it is considered as Neutral. We add another column in the data frame namely sentiment to record the sentiment of each row.

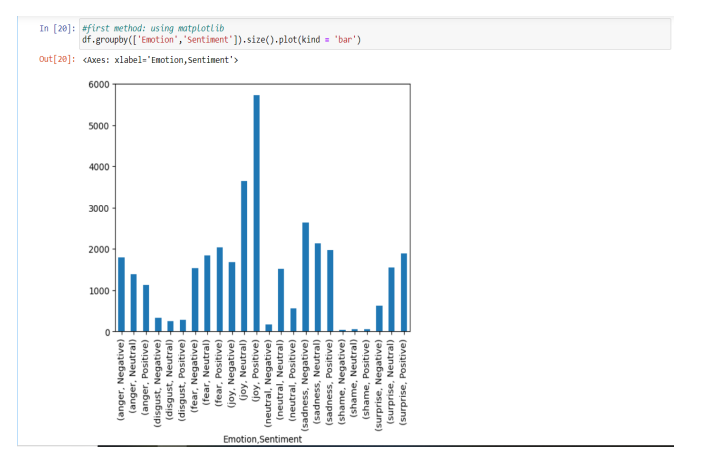
****

**Fig. 17 Extract sentiment of the data**

****

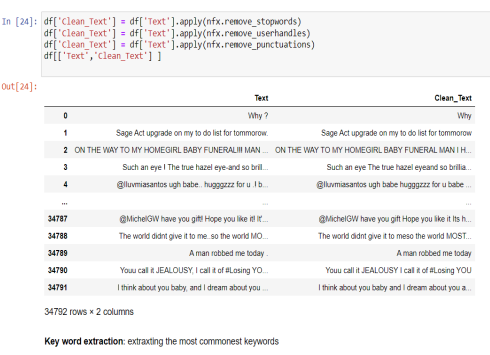
**Fig. 18 Emotion and Sentiment Relation**

The above code helps to group the sentiments of each emotion. Then we plot a bar graph based on the grouped data.

****

**Fig. 19 Bar Graph of Emotion and Sentiment Relation**

**3. Data Pre-processing:**

****

**Fig. 20 Data Cleaning**

**df['Clean\_Text'] = df['Text']. apply(nfx.remove\_stopwords) :** It is used to remove the stop words from the messages

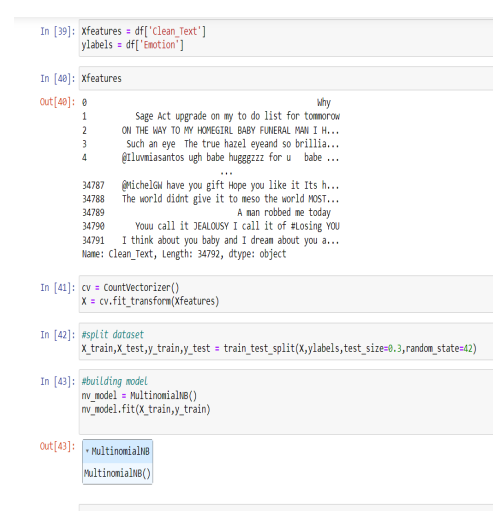
**df['Clean\_Text'] = df['Text'].apply(nfx.remove\_userhandles):** It is used to remove usernames, URL’s from the data

**df['Clean\_Text'] = df['Text'].apply(nfx.remove\_punctuations)** : It is used to eliminate punctuations from the data

**df[['Text','Clean\_Text'] ] :** The cleaned text is replaced with the previous text

**4. Model Creation:**

We assign the text to x\_featues and the respective emotions to y\_labels as input to the model. CountVectorizer is used to convert the message into a numerical array We perform the training and testing of the model with 70% and 30% of the data respectively. We create a Multinomial Naive Bayes model and then feed the training data and testing data into the model.

****

**Fig. 21 Model Creation**

The predict\_emotion function takes a sentence and a model as arguments. It converts the sentence into an array using the cv.transform . The transformed sentence is fed into the model.predict function. The emotion with most probability is given as output

****

**Fig. 22 Model Prediction**

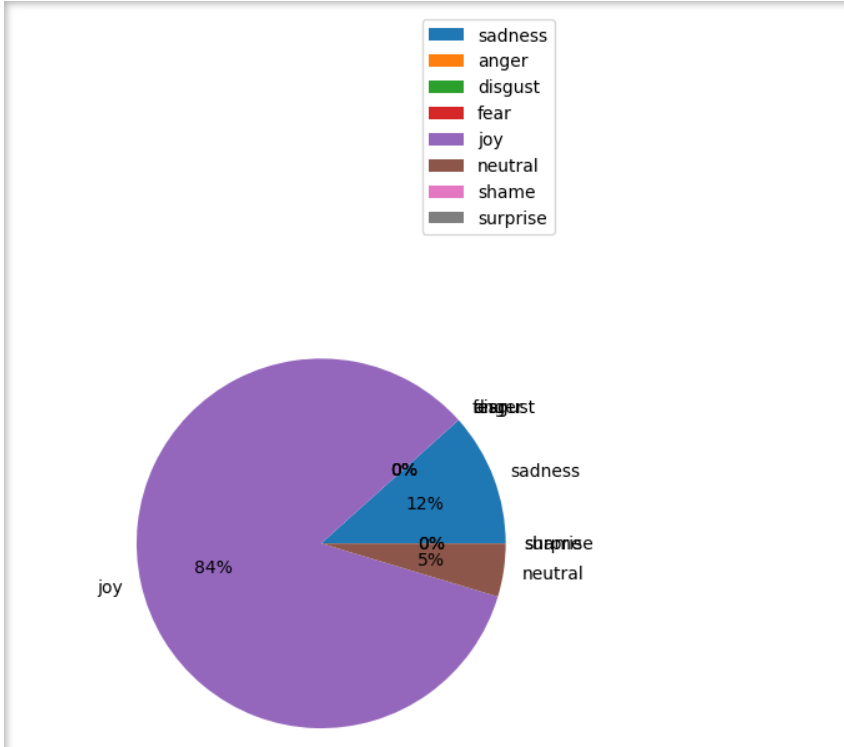
**5. Saving Model:**

****

**Fig. 23 Saving the Model**

The joblib module is used to save the Machine Learning models to extract and use them for predictions. We create a pickle file and write the bytes to the file of the model. We also save the CountVectorizer created for converting the data to arrays.

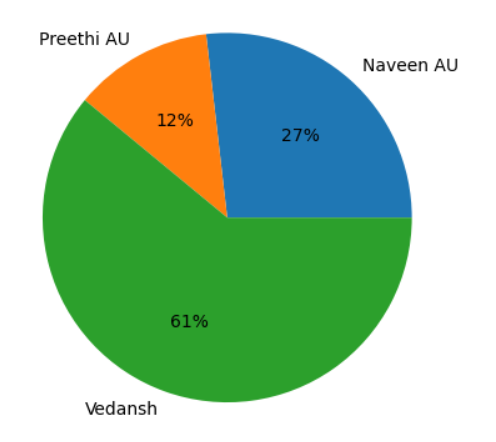
**6. Data Visualisation:**

****

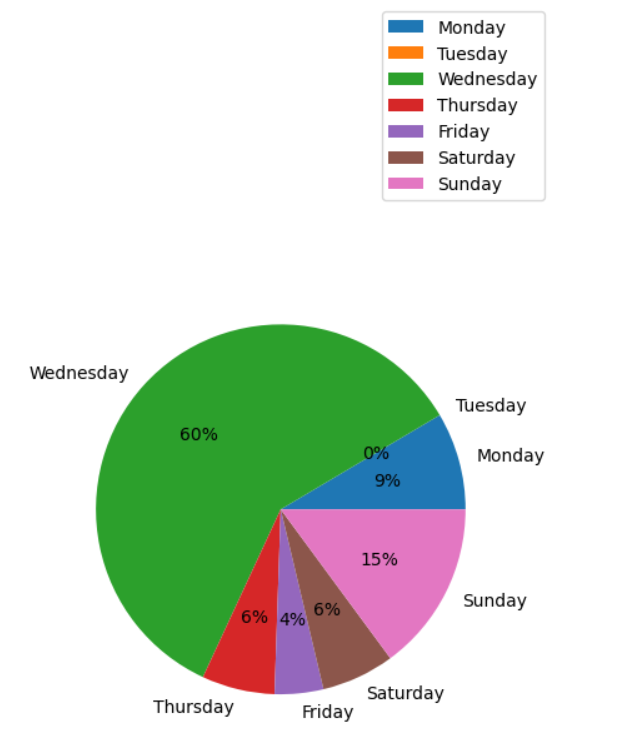
**Fig. 24 Data Visualisation**

In the above code, we use the model we created on the exported WhatsApp chat and we predict the emotion of each message and plot a pie chart from the whole analysis.

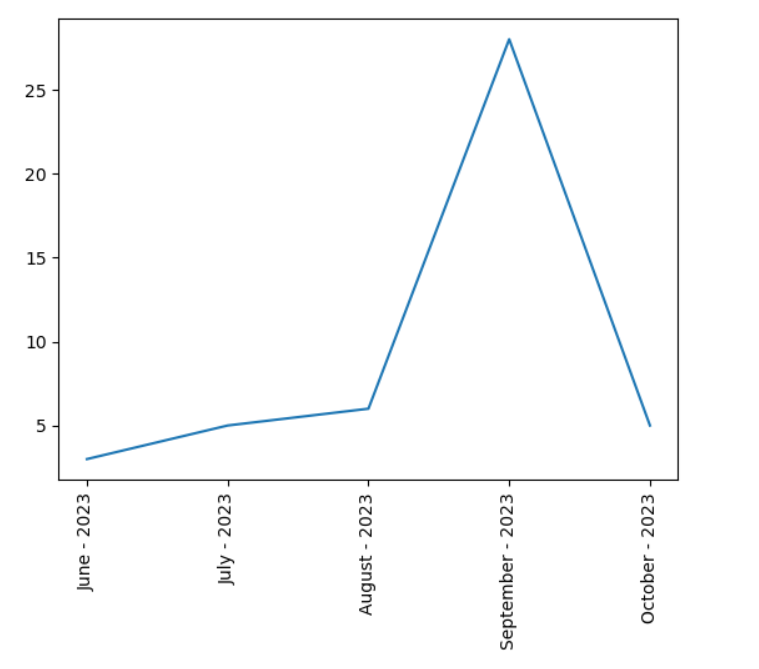
**3.5 Output Screens:**

****

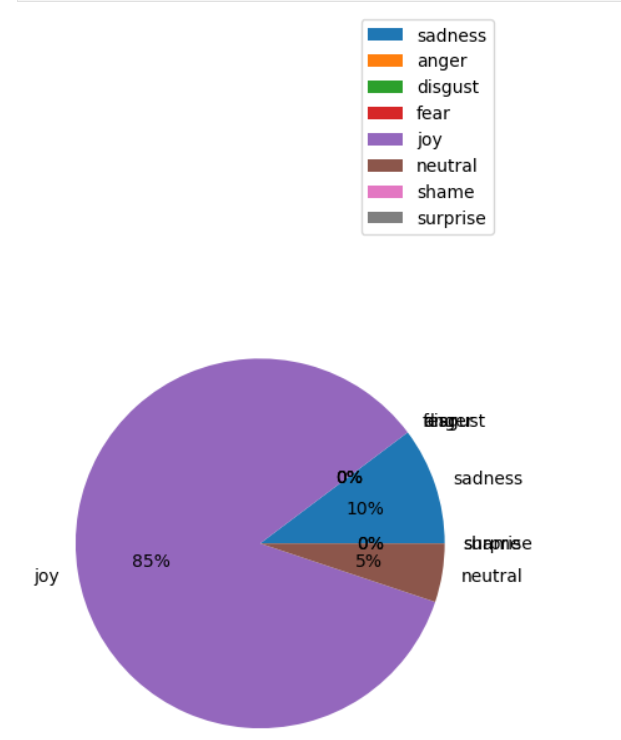
**Fig. 25 Users Usage Chart**

****

**Fig. 26 Day-wise Usage Chart**

****

**Fig. 27 Month Wise Usage Chart**

****

**Fig. 28 Emotion Detection Chart**

**4. TESTING: TEST CASES AND RESULTS**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test  Scenario | TC-ID | Feature  Description | Pre-Requisite | Test  Description | Input | Expected Result | Actual Result | P/F |
| Analysing Chat for frequent words | TC-01 | To obtain most frequent words | Messages should be splitted into words | 1. Input the chat data  2.Extract the messages  3. Split messages to words  4.Run the analysis code | Exported WhatsApp chat | top 10 frequently words in the chat | The analyser displays top 10 frequent words in the chat | P  A  S  S |
| Analysing the chat for most active users | TC-03 | obtaining most active user | The users and count of messages should be calculated | 1.Input the chat data  2. Extract unique users 3. Count the messages for each user | Exported WhatsApp chat | The analyser displays a pie chart for the most active user | The analyser displays a pie chart for the most active user | P  A  S  S |
| Analysing the emotion of the message | TC  -03 | To test the prediction on of emotion by the emotion detection n model | The model should be created and saved for importing | 1.Input the emotion dataset  2. Build a model using Multinomial NB  3.Save the model using joblib | [“I hate you”] | Anger | Anger | P  A  S  S |
| Analysing the emotion of the message | TC  -04 | To test the prediction on of emotion by the emotion detection n model | The model should be created and saved for importing | 1.Input the emotion dataset  2. Build a model using Multinomial NB  3.Save the model using joblib | [“Wow What a catch!”] | Joy | Joy | P  A  S  S |
| Analyse chat for active hours | TC  -05 | To analyse the activity and time spent by each user. | Date and time data Should be converted into the appropriate format. | 1.Give the WhatsApp chat analyser the chat data. 2.Pick the  option for active hours analysis. | Exported WhatsApp chat | messages sent during different hours of the day. | The active Hours are displayed Using charts for each user. | P  A  S  S |

**5. CONCLUSION AND FUTURE SCOPE**

**Conclusion:**

The answer to a machine learning-based analysis of a WhatsApp message relies on the particular research topic and the data set employed. Here are some broad conclusions that may be drawn from such a study, though: To determine the primary subjects mentioned in the WhatsApp discussion, utilize machine learning. This can be helpful for figuring out the chat's overarching topic and locating areas of interest. Machine learning may be used to assess the tone of the chat messages. This might shed light on the participants' general state of mind and feelings. The social network of chat participants, including the frequency and kind of interactions between them, may be analysed using machine learning. This can provide light on the dynamics of the group and the functions played by the various members. The vocabulary, grammar, and syntax used by the participants may all be examined using machine learning. This can provide information about the individuals' educational background, sophistication level, and cultural and language background. Overall, a WhatsApp chat analysis employing machine learning can offer insightful information about the participants' social dynamics and communication styles. A wide range of research issues, from marketing and consumer behaviour to social and political analyses, can be informed by these findings.

**Future Scope:**

WhatsApp Chat Analysis is a major concern as it helps to predict the business and detect any unwanted activities. In the future we would like to add the analysis of WhatsApp media such as images, videos, voice notes, stickers, documents. We can also develop the system to run independently on the cloud so that it can be available for use from any corner of the world. We can also develop the system to provide a PDF of the whole Analysis report to the user which can be shared or stored on the cloud. Detection of spam accounts based on the chat analysis can be notified to the users. We can develop a third-party application so as to work on the smartphones directly by extracting the chats to which permission is granted by the user and send daily / weekly / monthly reports to the user to his WhatsApp or Email automatically.

**6. REFERENCES**

[1] Wang, Yongming. "Using Machine Learning and Natural Language Processing to Analyse Library Chat Reference Transcripts." Information Technology and Libraries 41, no. 3 (2022).

[2] Dahiya, Sonika, Astha Mohta, and Atishay Jain. "Text classification based behavioural analysis of WhatsApp chats." In 2020 5th international conference on communication and electronics systems (ICCES), pp. 717-724. IEEE, 2020.

[3] Ahmad, Zishan, Raghav Jindal, Asif Ekbal, and Pushpak Bhattacharyya. "Borrow from rich cousin: transfer learning for emotion detection using cross lingual embedding." Expert Systems with Applications 139 (2020): 112851.

[4] Akhilesh Kumar, Bhavna Bajpai, Rachit Adhvaryu, Suthar Dhruvi Pankaj Kumar, Prajapati Parth Kumar Gordhanbhai, and Atul Kumar. "An Efficient Approach of Product Recommendation System using NLP Technique." Materials Today: Proceedings (2021).

[5] “Analysis of WhatsApp Chat for Crime Investigation” by M A Rahman and A H M Kamal (2021)