

Whatsapp Chat Analyzer (ML Based Web App)

A MAJOR PROJECT REPORT

Submitted by

MD NADIM

2020-310-118

**in partial fulfilment for the award of the degree of
B.TECH COMPUTER SCIENCE AND ENGINEERING**

Under the supervision of

MS. ANAM SAIYEDA



Department of Computer Science & Engineering

School of Engineering Sciences & Technology

JAMIA HAMDARD

New Delhi-110062

(2024)



JAMIA HAMDARD

Phone : 011-2605 9588 (12 Lines)
E-mail inquiry@jamiahamdard.edu
Website : www.jamiahamdard.edu

(Declared as Deemed-to-be University under Section 3 of
The UGC Act, 1956 vide Notification No. F.9-18/85-U3
(Accredited by NAAC in "A+" Category)

CERTIFICATE

On the basis of the declaration submitted by **Md Nadim (Enrolment No: 2020-310-118)** a student of **B.Tech (Bachelor of Technology (CSE))**, I hereby certify that the dissertation titled "**Whatsapp Chat Analyzer (ML Based Web App)**" being submitted to the Department of Computer Science & Engineering, Jamia Hamdard, New Delhi in partial fulfillment of the requirement for the award of the degree of **B.Tech (Bachelor of Technology (CSE))**, is carried out by him under my supervision.

Prof. Farheen Siddiqui
Head, Department of CSE

Ms. Anam Saiyeda
(Supervisor)

DECLARATION

I, **MD NADIM** a student of Bachelors in Technology Computer Science & Engineering (**B.Tech CSE**), (**Enrolment No: 2020-310-118**) hereby declare that the Project/Dissertation entitled “**Whatsapp Chat Analyzer (ML Based Web App)**” which is being submitted by me to the Department of Computer Science, Jamia Hamdard, New Delhi in partial fulfilment of the requirement for the award of the degree of Bachelors in Technology Computer Science and Engineering (B.Tech CSE) , is my original work and has not been submitted anywhere else for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

MD NADIM

Date:

(Signature and Name of the Applicant)

Place: New Delhi

Jamia Hamdard

ACKNOWLEDGEMENT

I would want to express my gratitude to a number of persons who have encouraged and assisted me in the preparation of this Machine learning and web application project, both directly and indirectly. It allows me to look back and reflect on the support I've had during this process.

I would like to express my gratitude to **Ms Anam Saiyeda**, my mentor, as well as the professors of Jamia Hamdard University, for their invaluable recommendations, innovative criticisms, and support throughout the writing process. I would like to offer my heartfelt gratitude to the entire faculty of Jamia Hamdard University (SEST).

My heartfelt gratitude goes out to my family and friends, as well as my esteemed university, Jamia Hamdard, for providing me with the chance and infrastructure to complete this project. Finally, I want to express my gratitude to everyone who assisted me in gathering data during the creation of the project, without whom it would not have been possible.

INDEX

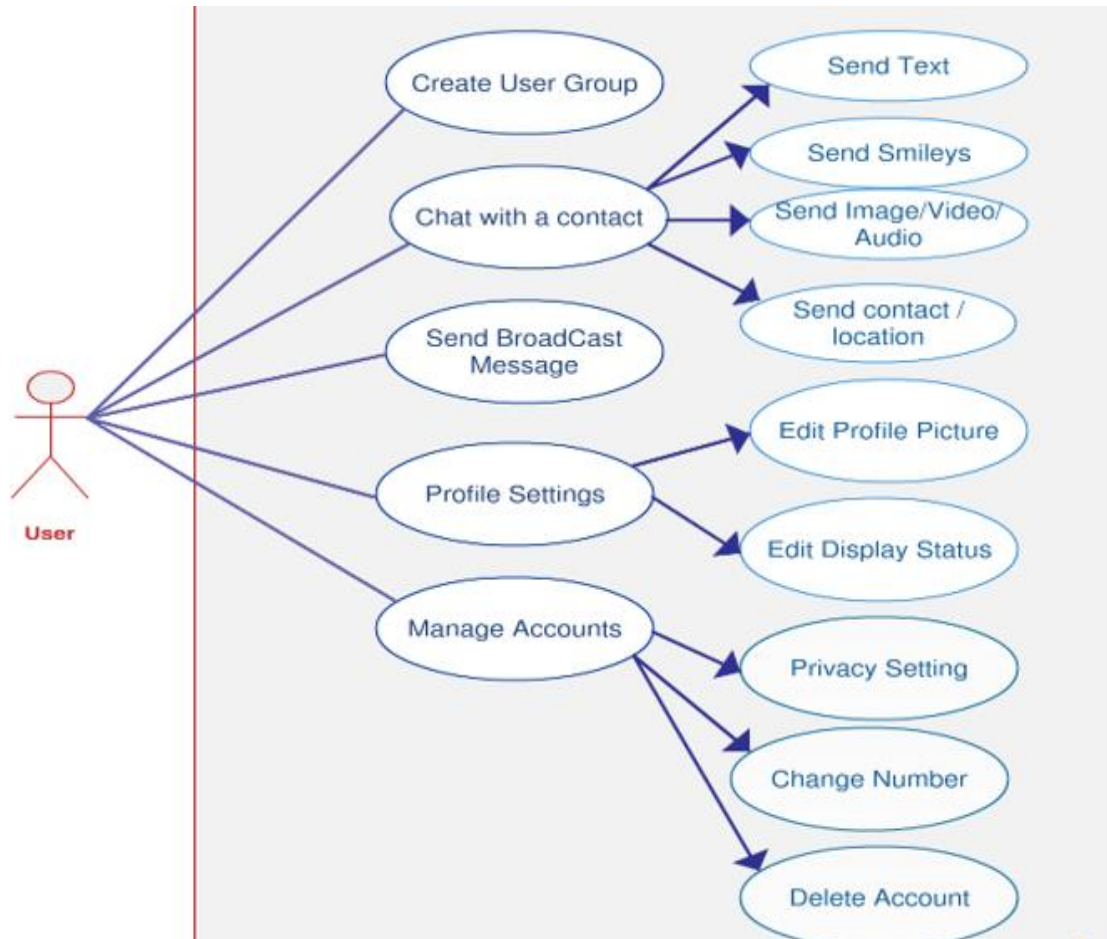
SR. NO.	TOPIC	PAGE NO.
1.	Abstract.....	7
2.	Introduction.....	8-9
3.	Objective.....	10
4.	Need of Whatsapp Chat Analysis.....	11-12
5.	Problem Statement.....	13
6.	Methodology.....	14-18
7.	Used component in Whatsapp chat analysis...	19-21
8.	Python Libraries used in Web App.....	22-23
9.	Snapshots of Project.....	24-29
10.	Conclusion.....	30
11.	Bibliography.....	31

WHATSAPP CHAT ANALYZER

(MACHINE LEARNING BASED Web APPLICATION)



USED CASE-DIAGRAM OF WHATSAPP CHAT ANALYZER **(ML Based WEB APPLICATION)**



ABSTRACT

The WhatsApp chat analyzer project is a web application developed using PyCharm IDE. It is designed to analyze the chat past of WhatsApp conversations, extract important information and provide insights into the chat data. The application allows the user to upload the WhatsApp chat data in a text file format, preprocess the data, and perform various analysis on the data, such as word frequency analysis, user activity analysis, and time-based analysis. The project also includes a feature to export the analyzed data into a CSV file format for further processing.

The application makes use of various Python libraries, such as Pandas, Matplotlib, and NLTK, for data analysis and visualization. The project also includes a user interface developed using Flask web framework,. The user interface is designed to be intuitive and user- friendly, allowing the user to upload the chat data, view the analyzed data, and download the results in a CSV file format.

Overall, the WhatsApp chat analyzer project is a useful tool for individuals and organizations to gain insights into their WhatsApp conversations, analyze their communication patterns, and improve their communication efficiency.

INTRODUCTION



With the increasing popularity of messaging apps like WhatsApp, the volume of text conversations has increased extremely. These conversations can contain a wealth of information about people's preferences, opinions, and sentiments. Analyzing WhatsApp chats can provide valuable insights into various aspects of communication and human behavior. WhatsApp chat analysis involves parsing and interpreting the contents of a chat and presenting it in a structured and meaningful way.

The analysis of WhatsApp chats can provide valuable insights into various aspects of communication and human behavior. For example,

it can help us understand the language used by people in different contexts, the topics that are discussed, the sentiment of the conversation, and the patterns in the communication. This information can be useful for a wide range of

applications, including market research, social science research, and psychological studies.

In recent years, there has been an increasing interest in developing tools and techniques for chat analysis. These tools aim to make the analysis of chats more accessible and easier for researchers, businesses, and individuals. In this project, we aim to develop a WhatsApp chat analyzer that can provide a comprehensive analysis of WhatsApp chats. We will use Python and PyCharm as the primary programming language and development environment, respectively.

OBJECTIVE

The objective of this project is to develop a web application that can perform a comprehensive analysis of WhatsApp chat data. The main objective is to provide users with an easy-to-use tool that can help them gain insights into their WhatsApp conversations. The following are the specific objectives of this project:

1. **Data Preprocessing:** The application should be able to clean, format and preprocess the raw WhatsApp chat data. This involves removing irrelevant data such as media files, system messages, and emojis, as well as structuring the data in a tabular form that can be easily analyzed.
2. **Chat Analysis:** The application should provide a comprehensive analysis of the chat data, including chat frequency, message length, most active users, most common words, and time of day when most chats occur.
3. **Visualizations:** The application should provide visualizations of the chat data analysis to help users better understand the results. These visualizations can include graphs, charts, and other data representations.
4. **User Interface:** The application should have a user-friendly interface that allows users to easily input their WhatsApp chat data and view the results of the analysis. The interface should be easy to navigate, intuitive and aesthetically pleasing.

Overall, the objective of this project is to develop a robust, user-friendly, and accurate WhatsApp chat analysis web application that can provide users with meaningful insights into their chat conversations.

NEED OF WHATSAPP CHAT ANALYSIS



The need for WhatsApp chat analysis arises due to the massive amount of data that is exchanged through the platform. With over 2 billion active users, WhatsApp has become one of the most popular messaging apps globally. People use WhatsApp to communicate with family, friends, and colleagues, making it an essential communication tool.

However, with this vast amount of data comes the need for efficient analysis. WhatsApp chat analysis can provide valuable insights into user behavior, sentiment, and patterns of communication. By analyzing WhatsApp chat data, it is possible to identify trends, track progress, and make informed decisions based on the data.

Moreover, WhatsApp chat analysis can also be used for legal and compliance purposes. With the rise of electronic communication, it has become increasingly important to keep track of conversations for regulatory purposes. WhatsApp chat analysis can help organizations to identify any potential issues or risks, such as compliance violations or data breaches.

In summary, the need for WhatsApp chat analysis is driven by the need to make sense of the vast amount of data that is exchanged through the platform. It can be

provided valuable insights into user behavior, sentiment, and patterns of communication.

WhatsApp chat analysis has become increasingly important in today's world due to its wide usage as a messaging platform. The need for WhatsApp chat analysis can be driven by several reasons, including monitoring employee communication, customer service, personal use, and research purposes.

In the case of companies, monitoring employee communication on WhatsApp is essential to ensure compliance with company policies, protect sensitive information, and prevent any misuse of company resources. Analyzing chat data can help identify any potential misconduct or policy violations.

Moreover, companies can use WhatsApp to provide customer support services, which is becoming increasingly popular. Analyzing chat data can help identify common customer queries, improve response time, and enhance the overall customer experience.

For individuals, analyzing their WhatsApp chats can be beneficial in identifying patterns in their communication with friends and family, monitoring their own behavior, or simply reminisce.

Researchers can also use WhatsApp chat data for social network analysis, sentiment analysis, or other types of research. This can provide valuable insights into how people communicate and interact with each other.

In conclusion, WhatsApp chat analysis is a valuable tool that can provide insights into communication patterns, identify trends, and areas for improvement, leading to better communication and a more efficient use of resources.

PROBLEM STATEMENT

The problem with WhatsApp chats is that they are often disorganized and difficult to navigate. The messages are not structured in any particular order, and the app does not provide any tools to analyze the chat's data. This makes it difficult to understand the chat's content, especially when the chat history is long.

Additionally, it is almost impossible to grasp and interpret crucial data from WhatsApp chat history without the aid of a WhatsApp chat analyzer web app. Currently, WhatsApp users spend considerable time scrolling through chat history to find the message they need, which is a frustrating and time-consuming process.

Furthermore, people who use WhatsApp for business communication have to keep track of their conversations with clients, suppliers, and partners. These users need a more efficient way to organize their conversations and analyze the data to make better business decisions.

Therefore, the need for a WhatsApp chat analyzer web app that can help users quickly analyze, understand and organize their chat history is essential. This app will save users valuable time and provide valuable insights into the conversations they've had with their contacts.

METHODOLOGY

The methodology for developing a WhatsApp chat analyzer web app using PyCharm involves several steps. Here is a detailed overview of the methodology

1. **Problem statement:** The first step is to clearly define the problem statement and identify the objectives of the project. This involves determining the purpose of the chat analysis, such as monitoring employee communication or improving customer service.
2. **Data collection:** The next step is to collect the data to be analyzed. In the case of a WhatsApp chat analyzer, this involves exporting the chat data from the WhatsApp app and saving it in a suitable format for analysis.



3. **Data preprocessing:** Before performing any analysis, the raw chat data needs to be preprocessed to clean and transform the data into a format suitable for analysis. This involves tasks such as removing irrelevant messages, separating message components, and handling missing or inconsistent data.

05/09/22, 8:22 pm - Rafe added +91 97170 93323
 08/09/22, 2:59 pm - +91 94522 24306: Come to LT 15
 08/09/22, 2:59 pm - Rafe: Okay sir
 13/09/22, 3:23 pm - +91 94522 24306: Machine Learning For Absolute Beginners by Oliver Theobald
 15/09/22, 2:51 pm - +91 94522 24306: Come to LT15
 15/09/22, 4:15 pm - Rafe: Syllabus...
 17/09/22, 9:56 am - +91 94522 24306: I am on leave for next week...
 26/09/22, 12:54 pm - +91 94522 24306: Class from 2:00 PM
 26/09/22, 12:58 pm - Rafe: ok sir
 26/09/22, 1:58 pm - +91 94522 24306: Come 409
 26/09/22, 1:58 pm - Rafe: ok sir
 26/09/22, 4:29 pm - +91 94522 24306: <Media omitted>
 26/09/22, 4:30 pm - +91 94522 24306: Change in your copies
 26/09/22, 5:38 pm - Rafe: okay sir...|
 30/09/22, 5:26 pm - +91 94522 24306: <Media omitted>
 30/09/22, 5:26 pm - +91 94522 24306: <Media omitted>
 30/09/22, 5:26 pm - Rafe: thank you sir@
 06/10/22, 10:26 am - +91 94522 24306: <Media omitted>
 06/10/22, 10:27 am - +91 94522 24306: Use class notes for the remaining.
 06/10/22, 10:28 am - +91 94522 24306: Syllabus till taught.
 06/10/22, 11:17 am - Rafe: ok sir...
 10/10/22, 1:55 pm - +91 94522 24306: 68, 70, 73, 74, 78, 84, 90, 94, 95, 100, 104, 105, 106, 107, 108, 111, 112, 115, and LE8, 11, 18
 10/10/22, 1:55 pm - +91 94522 24306: These students meet me.
 10/10/22, 1:55 pm - +91 94522 24306: For attendance
 10/10/22, 1:57 pm - +91 94522 24306: Else will not be allowed for exam.
 13/10/22, 3:35 pm - Rafe: Sir, The paper was very lengthy which is impossible to solve in just 1 hour....
 13/10/22, 4:03 pm - Zaifan: This message was deleted
 17/10/22, 12:48 pm - +91 94522 24306: I am busy in verification today... No class
 17/10/22, 12:48 pm - Shabab Sec B: Ok sir
 17/10/22, 12:48 pm - Rafe: Okay sir
 17/10/22, 12:52 pm - +91 70111 88053: Sir Aaj ki attendance to milegi na?
 17/10/22, 12:56 pm - Zaifan: +1
 17/10/22, 1:05 pm - Himanshu: Ok sir
 20/10/22, 1:05 pm - +91 94522 24306: All of you should attend the IEEE event, It's important for all of you.
 28/10/22, 11:51 am - +91 94522 24306: <Media omitted>
 28/10/22, 11:53 am - +91 94522 24306: Submit on 3 November
 28/10/22, 12:14 pm - Rafe: Okay sir...
 28/10/22, 12:31 pm - +91 94522 24306: Also include those questions which I given in class.
 28/10/22, 12:32 pm - Zaifan: Sir it should be handwritten or we can submit word file?
 28/10/22, 12:32 pm - +91 94522 24306: Hand written only
 28/10/22, 12:37 pm - +91 94522 24306: Make separate file for assignment...
 28/10/22, 12:39 pm - Rafe: Sir, please also share the answers of these questions....
 28/10/22, 12:40 pm - Zaifan: ?

4. Exploratory data analysis: With the preprocessed data, exploratory data analysis techniques can be used to gain insights into the data. This involves techniques such as data visualization, summary statistics, and correlation analysis.

Top Statistics

Total
Messages

211



Total
Words

1152

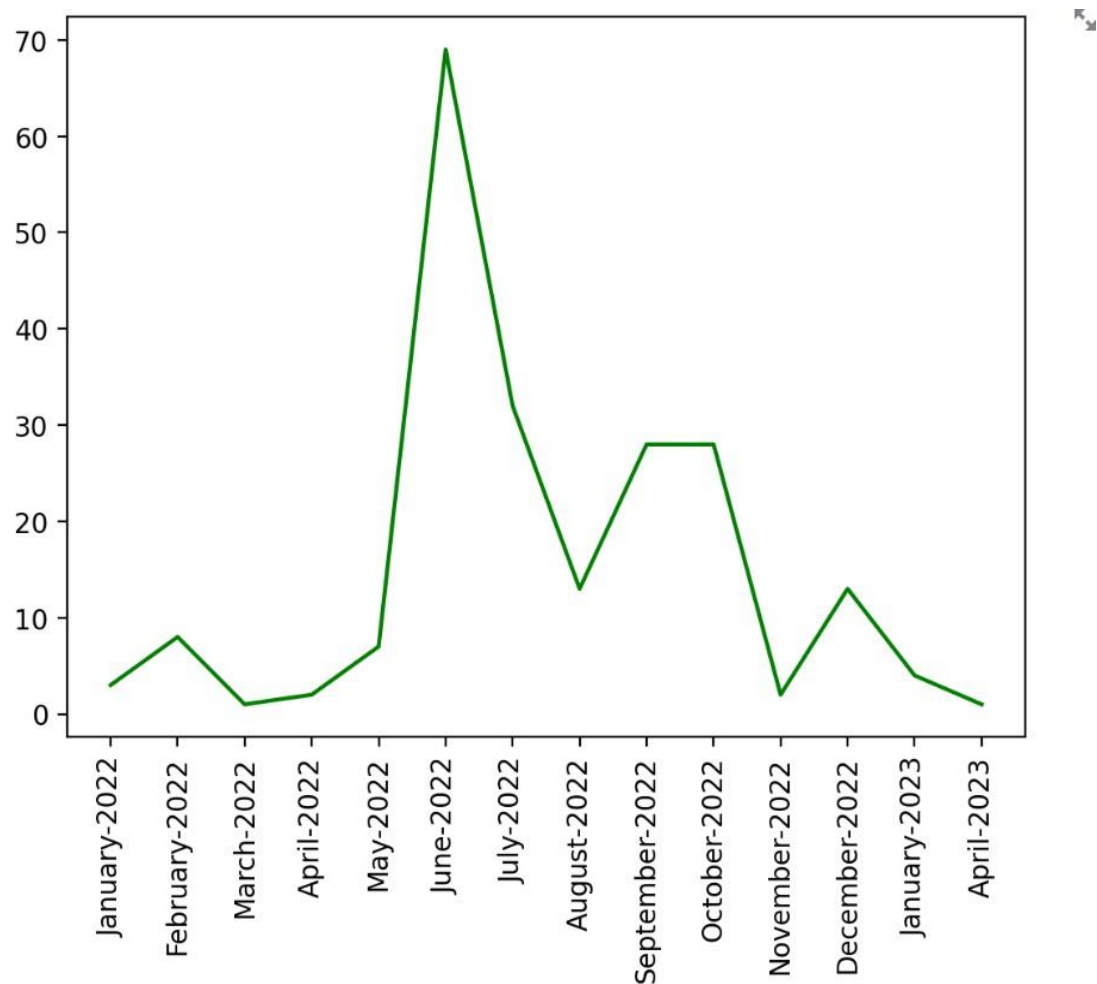
Media
Shared

14

Links
Shared

3

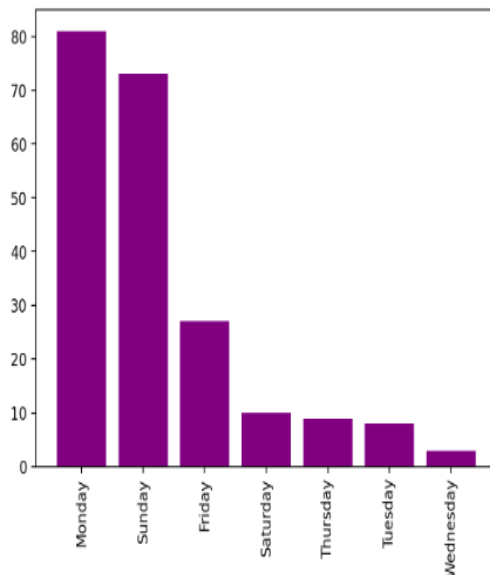
Monthly Timeline



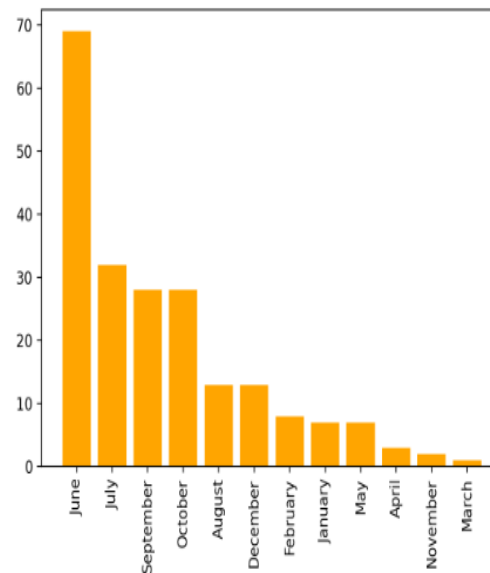
5. Text preprocessing: Since the chat messages are textual in nature, text preprocessing techniques such as tokenization, stemming, and stop word removal can be applied to prepare the text data for analysis.

Activity Map

Most busy day

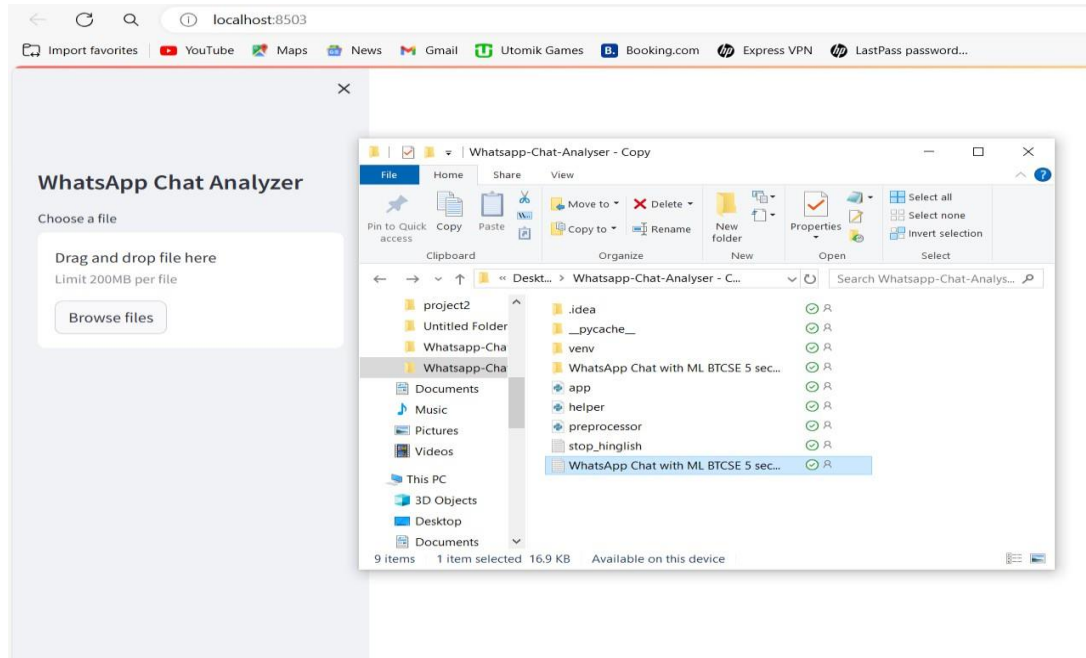


Most busy month



6. Text classification: Text classification techniques can be used to classify messages into predefined categories. For example, messages related to customer support can be classified into different categories such as technical support, billing queries, etc.

7. Web app development: Finally, a web app can be developed using a framework such as Streamlit to visualize the results of the analysis and provide an interactive interface for users to explore the chat data.



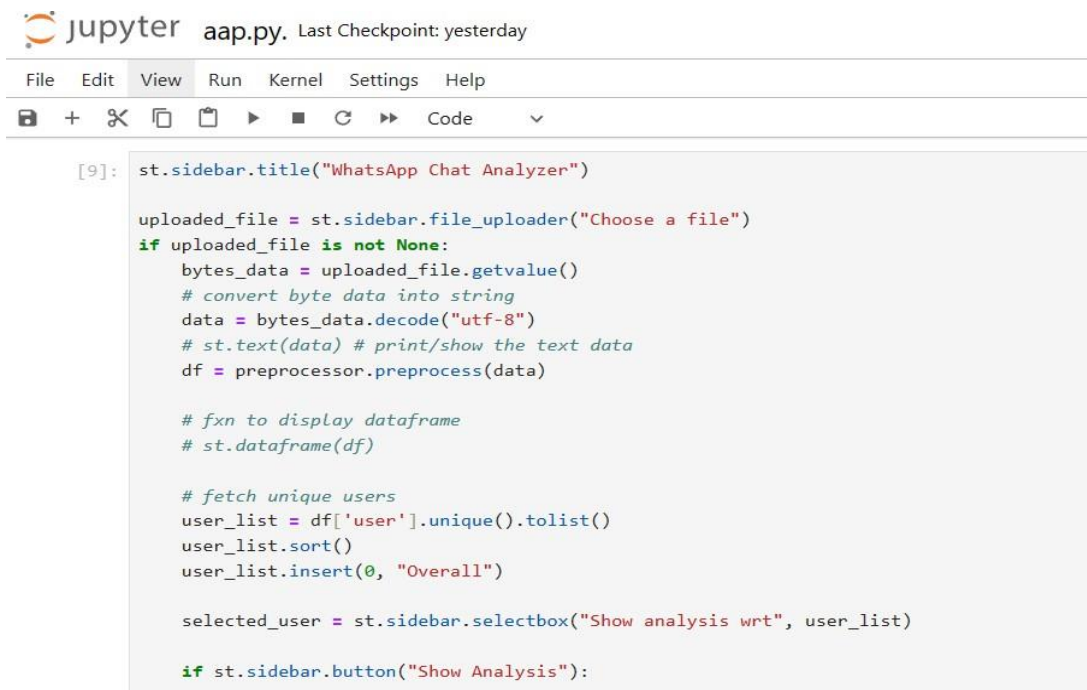
By following this methodology, a comprehensive WhatsApp chat analyzer web app can be developed using PyCharm

USED COMPONENTS IN WHATSAPP CHAT ANALZER WEB APP

JUPYTOR NOTEBOOK - Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It is widely used by data scientists, researchers, and students to perform data analysis, machine learning, and scientific computing tasks.

Jupyter Notebook provides an interactive computing environment where users can write and execute code in multiple programming languages, including Python, R, Julia, and others. The code is executed in cells, which can be run independently or in a specific order, allowing users to experiment and iterate quickly.

Jupyter Notebook also supports the creation of rich, interactive visualizations using libraries such as Matplotlib, Seaborn, and Plotly. These visualizations can be used to explore and communicate insights from data in a more intuitive and engaging way.



```
[9]: st.sidebar.title("WhatsApp Chat Analyzer")

uploaded_file = st.sidebar.file_uploader("Choose a file")
if uploaded_file is not None:
    bytes_data = uploaded_file.getvalue()
    # convert byte data into string
    data = bytes_data.decode("utf-8")
    # st.text(data) # print/show the text data
    df = preprocessor.preprocess(data)

    # fxn to display dataframe
    # st.dataframe(df)

    # fetch unique users
    user_list = df['user'].unique().tolist()
    user_list.sort()
    user_list.insert(0, "Overall")

    selected_user = st.sidebar.selectbox("Show analysis wrt", user_list)

    if st.sidebar.button("Show Analysis"):
```

PYTHON- Python is a high-level programming language that was released in 1991 by Guido van Rossum. It is an interpreted, dynamically typed language that is easy to learn and has a large standard library. Python is open-source and runs on various platforms such as Windows, Linux, and macOS.

Python is widely used in various industries such as web development, scientific computing, data analysis, artificial intelligence, and machine learning. It has numerous libraries and frameworks, including Django, Flask, NumPy, Pandas, Matplotlib, TensorFlow, and PyTorch, among others.

Python is known for its ease of use, versatility, and speed. It allows for rapid prototyping, making it a popular language in research and development. It also has a large user base and community, making it easy to find support and resources.

Overall, Python is a powerful language with a wide range of applications and is well-suited for both beginners and experienced programmers.

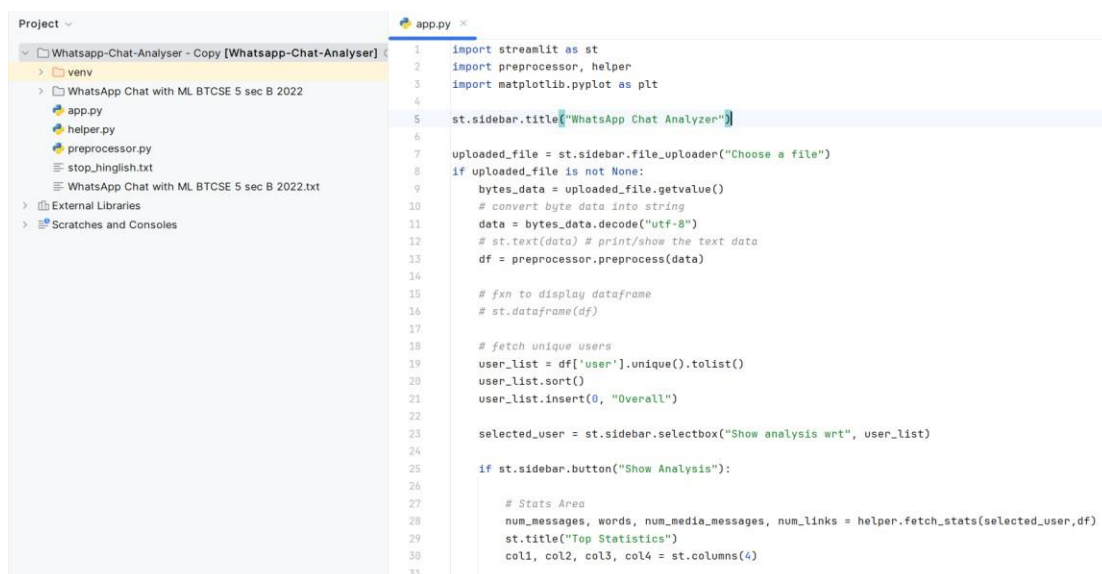
PYCHARM- PyCharm is an Integrated Development Environment (IDE) used for Python programming. It is developed by JetBrains, and it is a cross-platform IDE that works on Windows, macOS, and Linux operating systems. PyCharm provides an intelligent code editor, which includes features such as syntax highlighting, code completion, and error detection.

PyCharm also comes with several tools and features that make development in Python much easier and more efficient. These tools include a debugger, profiler, testing framework, version control system integration, and database support. PyCharm also supports web development with frameworks such as Django and Flask, and it includes support for JavaScript, HTML, and CSS.

In addition to its features for Python development, PyCharm also includes support for other programming languages such as JavaScript, TypeScript, SQL, HTML, CSS, and more. This makes it a powerful IDE for full-stack development.

PyCharm is available in two editions: the Community Edition and the Professional Edition. The Community Edition is free and open-source, while the Professional Edition is a paid version that includes additional features such as remote development, scientific tools, and database tools.

Overall, PyCharm is a powerful IDE for Python development that provides a range of features to make the development process more efficient and productive.



The screenshot displays the PyCharm IDE interface. On the left, the 'Project' sidebar shows a tree view of the project structure, including a 'venv' directory and a 'WhatsApp Chat with ML BTCSE 5 sec B 2022' directory containing files like 'app.py', 'helper.py', and 'preprocessor.py'. The main editor window on the right shows the code for 'app.py'. The code imports 'streamlit' as 'st', 'preprocessor' as 'helper', and 'matplotlib.pyplot' as 'plt'. It defines a sidebar titled 'WhatsApp Chat Analyzer' with a file uploader and a text area. The code handles file uploads, decodes the data, and processes it using the 'preprocessor' module. It also fetches unique users and displays a dataframe. The code is as follows:

```
1 import streamlit as st
2 import preprocessor, helper
3 import matplotlib.pyplot as plt
4
5 st.sidebar.title("WhatsApp Chat Analyzer")
6
7 uploaded_file = st.sidebar.file_uploader("Choose a file")
8 if uploaded_file is not None:
9     bytes_data = uploaded_file.getvalue()
10    # convert byte data into string
11    data = bytes_data.decode("utf-8")
12    # st.text(data) # print/show the text data
13    df = preprocessor.preprocess(data)
14
15    # fxn to display dataframe
16    # st.dataframe(df)
17
18    # fetch unique users
19    user_list = df['user'].unique().tolist()
20    user_list.sort()
21    user_list.insert(0, "Overall")
22
23    selected_user = st.sidebar.selectbox("Show analysis wrt", user_list)
24
25    if st.sidebar.button("Show Analysis"):
26
27        # Stats Area
28        num_messages, words, num_media_messages, num_links = helper.fetch_stats(selected_user, df)
29        st.title("Top Statistics")
30        col1, col2, col3, col4 = st.columns(4)
```

PYTHON LIBRARIES USED IN CHAT ANALYSIS

1. Streamlit- Streamlit is an open-source Python library that allows you to create web applications in a few lines of code. It was specifically designed to create data science and machine learning applications, but can also be used for a wide range of other web applications. Streamlit allows you to quickly and easily create user interfaces for your Python code, making it easy to share your data analysis and machine learning models with others.
2. Pandas: Pandas is a very popular library for data manipulation and analysis. It is used to read the WhatsApp chat data, preprocess it, and perform exploratory data analysis.
3. Numpy: Numpy is a library for scientific computing in Python. It is used for numerical operations and array manipulation.
4. Matplotlib: Matplotlib is a plotting library used for creating visualizations in Python. It is used to create various graphs and charts to represent the chat data.
5. NLTK: The Natural Language Toolkit (NLTK) is a library used for natural language processing in Python. It is used for sentiment analysis, text classification, tokenization, and other NLP tasks.
6. TextBlob: TextBlob is a library built on top of NLTK. It provides a modest API for common NLP tasks such as sentiment analysis, part-of- speech tagging, and noun phrase extraction.
7. Wordcloud: Wordcloud is a library used for generating word clouds. It is used to visualize the most frequent words in the chat data.

8. Seaborn: Seaborn is a useful data visualization library based on Matplotlib. It provides a high-level interface for creating smart statistical graphics.
9. Emoji: The emoji library in Python is a lightweight library that allows developers to work with Unicode emoji characters seamlessly. This library provides a simple way to detect, extract and manipulate emoji in text. It can be used to clean emoji from text or to extract them to visualize their usage.
10. Urlextract: The urlextract library in Python is a library that allows developers to extract URLs from text. It uses regular expressions to extract URLs from text, including URLs that are hidden behind shortened URLs. This library can be used to extract URLs from text messages in WhatsApp chats, which is useful in analyzing how URLs are being shared among chat participants.

SNAPSHOTS OF THE PROJECT

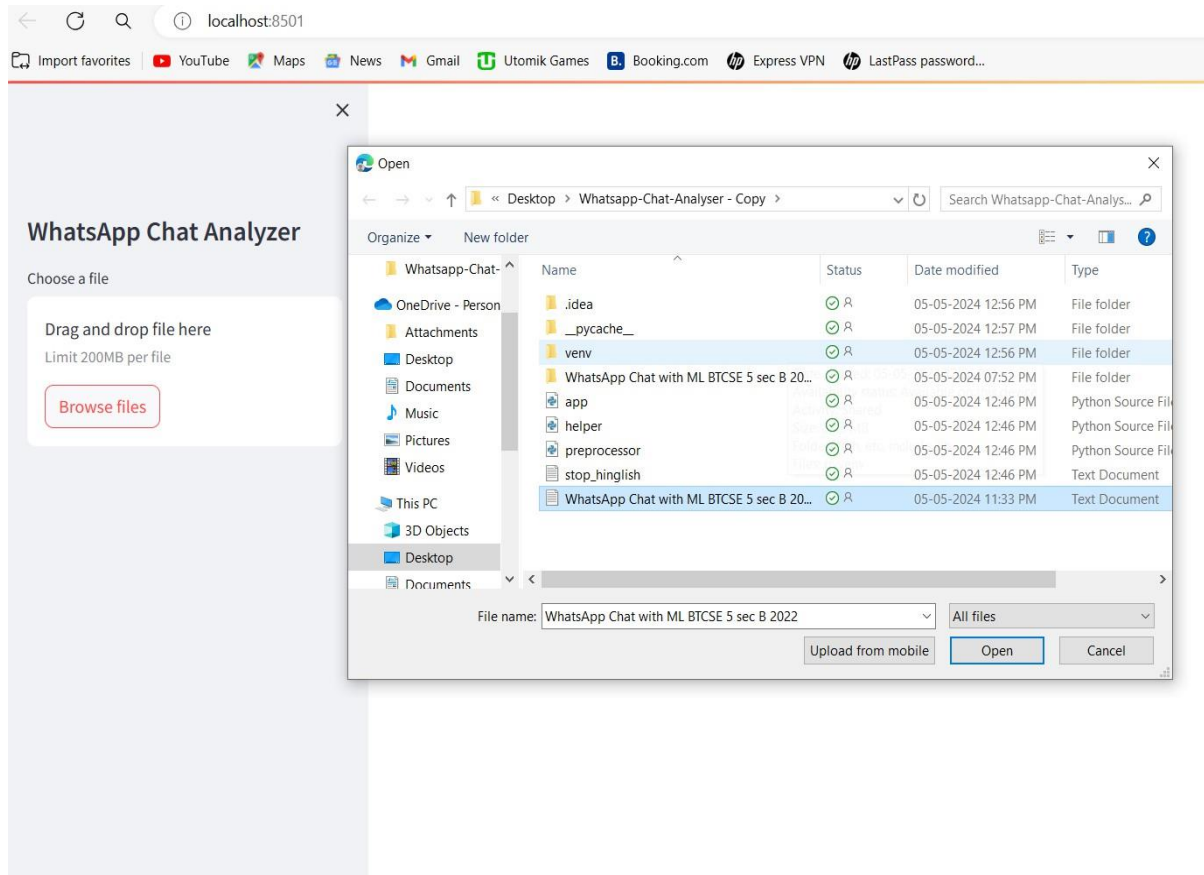


FIGURE-1

Top Statistics

Total Messages	Total Words	Media Shared	Links Shared
211	1152	14	3

Monthly Timeline

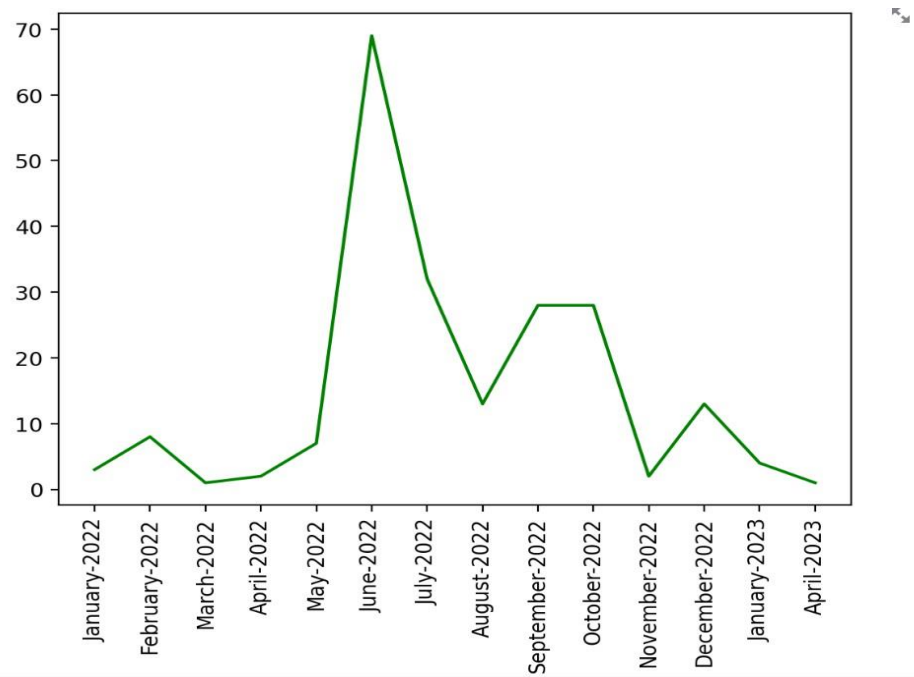
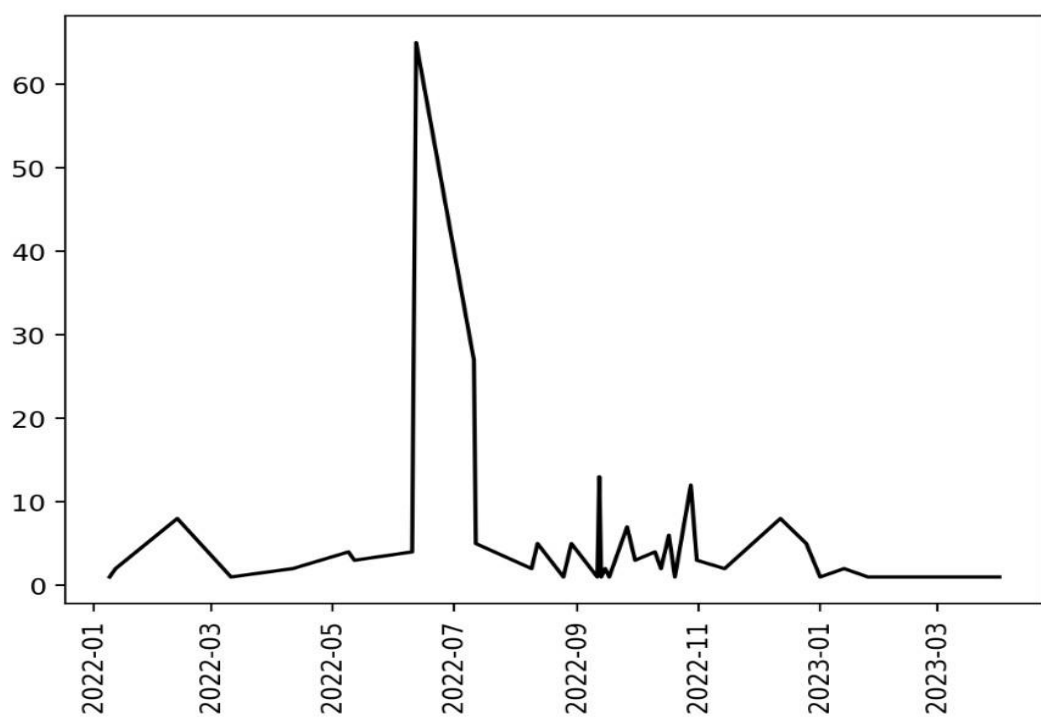


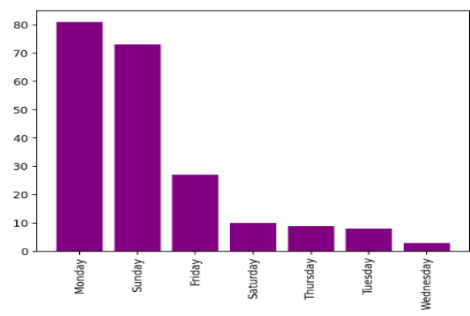
FIGURE-2

Daily Timeline



Activity Map

Most busy day



Most busy month

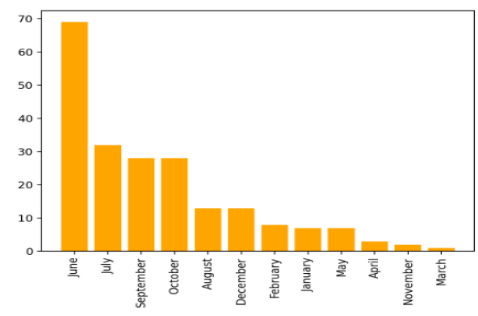
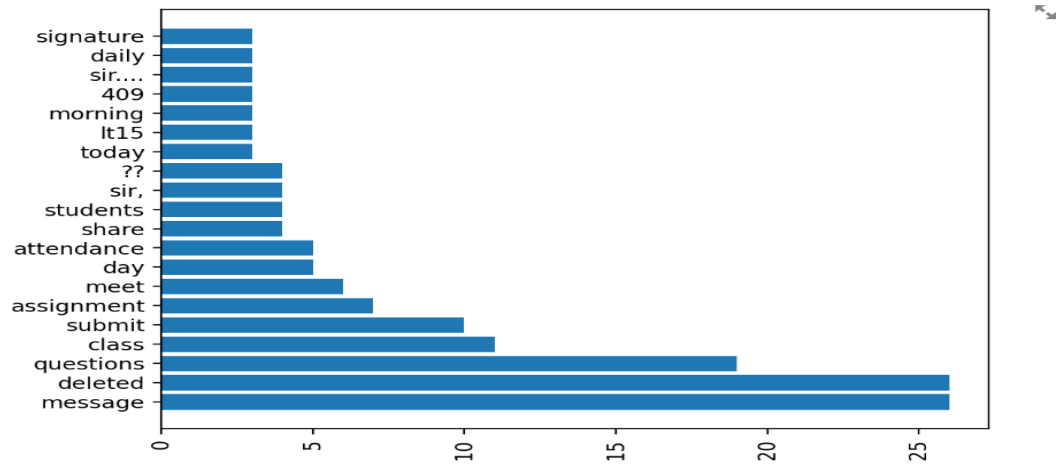


FIGURE-3

Most common words



	0	1
0	message	26
1	deleted	26
2	questions	19
3	class	11
4	submit	10
5	assignment	7
6	meet	6
7	day	5
8	attendance	5
9	share	4

FIGURE-5

Emoji Analysis

	0	1
0	😊	1
1	👉	1
2	📄	1
3	😄	1



FIGURE-6

CONCLUSION

The WhatsApp Chat Analyzer web app developed using PyCharm provides a user-friendly interface for analyzing and visualizing WhatsApp chat data. The app offers various features, including sentiment analysis, word frequency analysis, and keyword extraction, providing valuable insights into the chat data.

The project utilized various Python libraries such as Streamlit, Matplotlib, Seaborn, Urlextract, Wordcloud, Pandas, Emoji, Textblob, and NLTK, which played a significant role in achieving the desired functionalities of the app.

Overall, this project has demonstrated the usefulness of chat analysis in various contexts, including employee monitoring, customer service, personal use, and research purposes. The app can help users identify patterns, trends, and areas for improvement in their communication, leading to better decision-making and improved communication skills.

Future work can involve improving the accuracy of sentiment analysis by utilizing more advanced machine learning techniques, integrating more features, such as named entity recognition and topic modeling, and developing a mobile app version of the web app.

BIBLIOGRAPHY

➤ Software

1. Pycharm
2. Jupyter Notebook
3. Vs Code
4. Localhost
5. Python

➤ Sites

1. www.google.com
2. www.youtube.com
3. www.python.com

PLAGIARISM SCAN REPORT



PLAGIARISM SCAN REPORT



Content Checked For Plagiarism



PLAGIARISM SCAN REPORT



Content Checked For Plagiarism



PLAGIARISM SCAN REPORT



Content Checked For Plagiarism