WhatsApp ChatAnalyser with NLP

Mohammed Zahan

Btech Computer Science Engineering (Artificial Intelligence and Machine Learning).

Sri Ramachandra Engineering and Technology

Chennai,India

[e0121033@sret.edu.in](mailto:e0121033@sret.edu.in)

Abstract-This paper presents a comprehensive analysis of WhatsApp chat data using a custom-built analyser application. The application, developed using Streamlit, processes chat logs to extract meaningful statistics and visualizations. Key metrics such as the number of messages, words, media, and links shared are computed, along with user activity trends. The results provide insights into communication patterns and user behavior on WhatsApp.

Keywords-NLP, WhatsApp, Chat Analysis, Data Visualization, User Interaction, Streamlit

I. Introduction

WhatsApp is a widely used messaging platform with millions of users exchanging billions of messages daily. Analyzing these chat logs can provide valuable insights for

Sivane S

Btech Computer Science Engineering (Artificial Intelligence and Machine Learning).

Sri Ramachandra Engineering and Technology

Chennai,India

e0121027@sret.edu.in

development of a WhatsApp chat analyser using NLP techniques and Streamlit for an interactive interface. Our goals were threefold: ease of use, powerful analytics, and an engaging user experience.

The application is designed to handle WhatsApp chat exports, preprocess the data, and generate various analytical metrics and visualizations. By using Streamlit, the application offers an interactive and user-friendly interface, making it accessible for users with varying levels of technical expertise.

II. Methodology

The WhatsApp analyser application is developed using Streamlit, a Python library for creating web applications. The main functionalities of the application include data preprocessing, filtering, and visualization. The following subsections detail the process involved in each of these functionalities

A. Data Preprocessing

The preprocessing module is responsible for converting raw chat data into a structured format that can be easily analyzed. The raw data, typically exported from WhatsApp in a text file format, contains messages with timestamps and sender information. The preprocessing involves the following steps:

1. Extracting Timestamps and Messages: Regular expressions are used to identify and extract timestamps and corresponding messages from the raw data.

2. Cleaning and Structuring Data: The extracted data is cleaned to remove any extraneous characters and is structured into a pandas DataFrame. This DataFrame includes columns for user messages, message dates, users, and individual messages.

B. Data Filtering and Analysis

Once the data is preprocessed, it can be filtered and analyzed based on user selections. The filtering module allows users to select specific users or analyze overall chat statistics. The following metrics are calculated:

1. Number of Messages: The total number of messages exchanged.

2. Total Words: The total number of words used in the messages.

3. Media Shared: The number of media files shared.

4. Links Shared: The number of links shared in the chat.

C. Data Visualization

The application generates various visualizations to represent the chat data graphically. These visualizations include:

1. Monthly Timeline: A line graph showing the number of messages sent each month.

2. Most Active Users: A bar chart displaying the most active users based on the number of messages sent.

3. Most Common Words: A bar chart of the most frequently used words in the chat, excluding common stop words.

4. Daily Timeline: A line graph showing the number of messages sent each day.

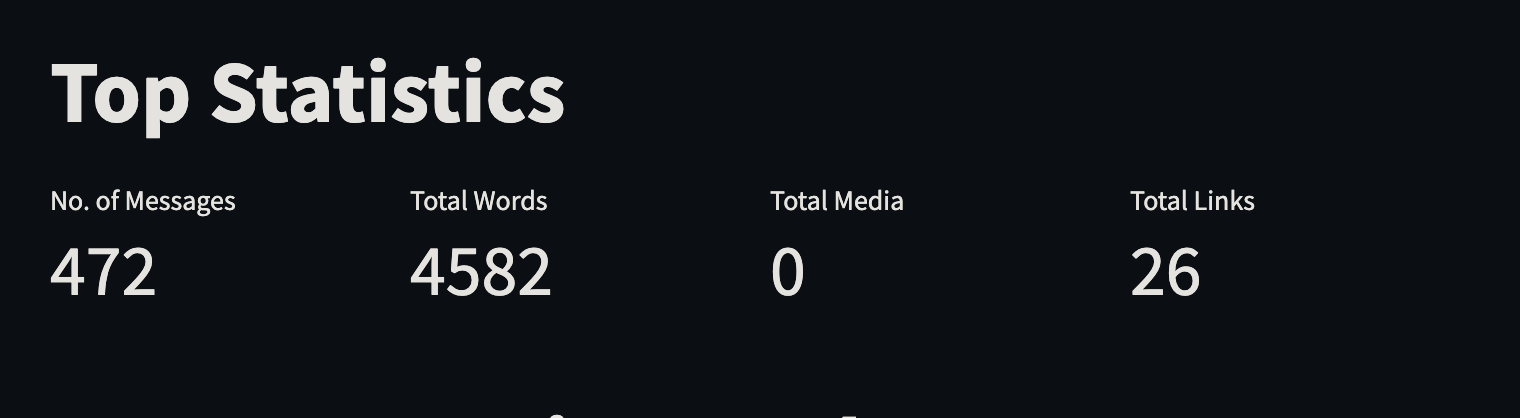
5. Activity Map: Bar charts showing the most active days of the week and the busiest months.

These visualizations help in understanding communication patterns and identifying trends in user activity.

III Results

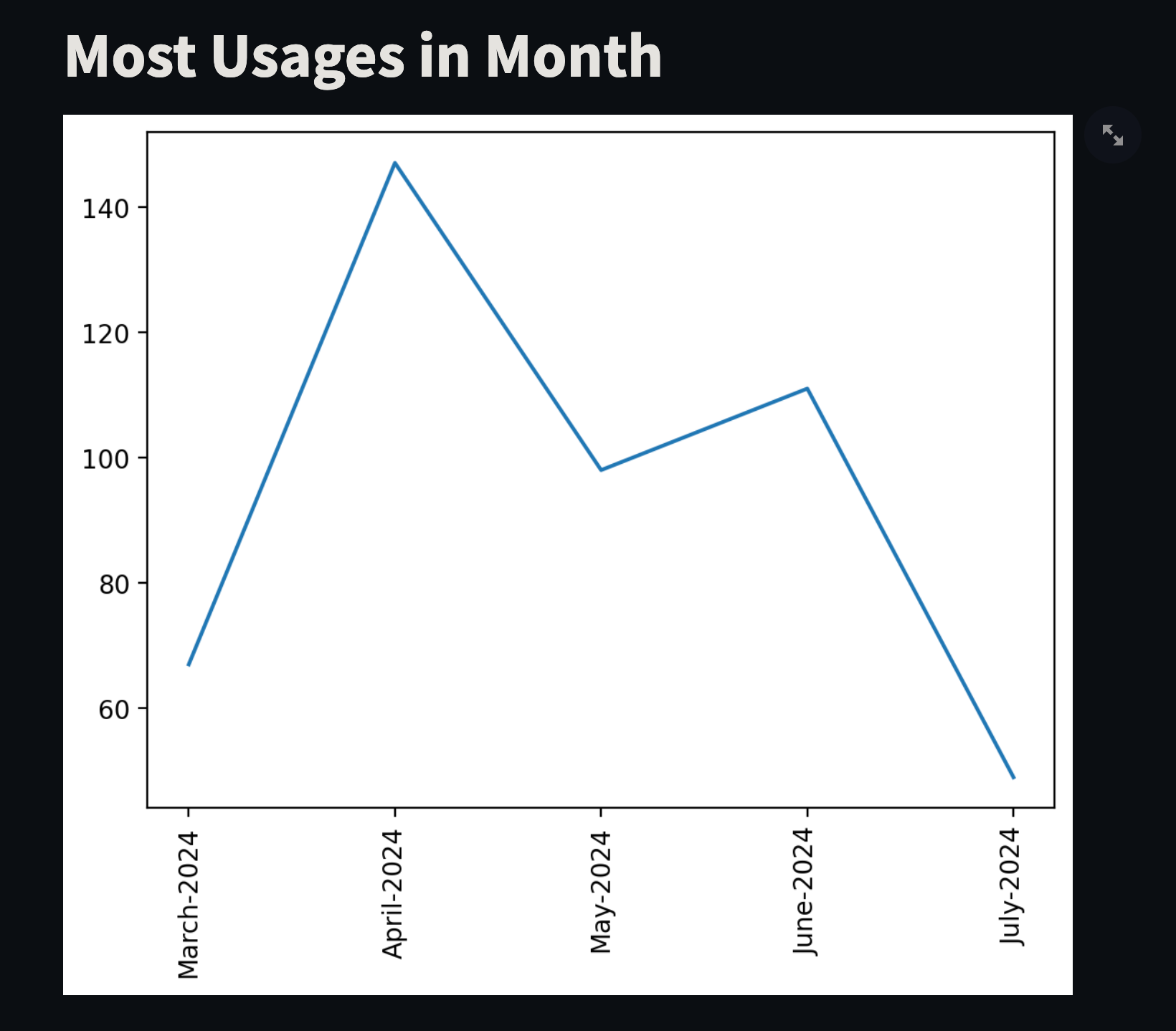
A. Top Statistics

The application provides an overview of key statistics, including the total number of messages, words, media files, and links shared. This overview helps users quickly understand the scale of the chat data.



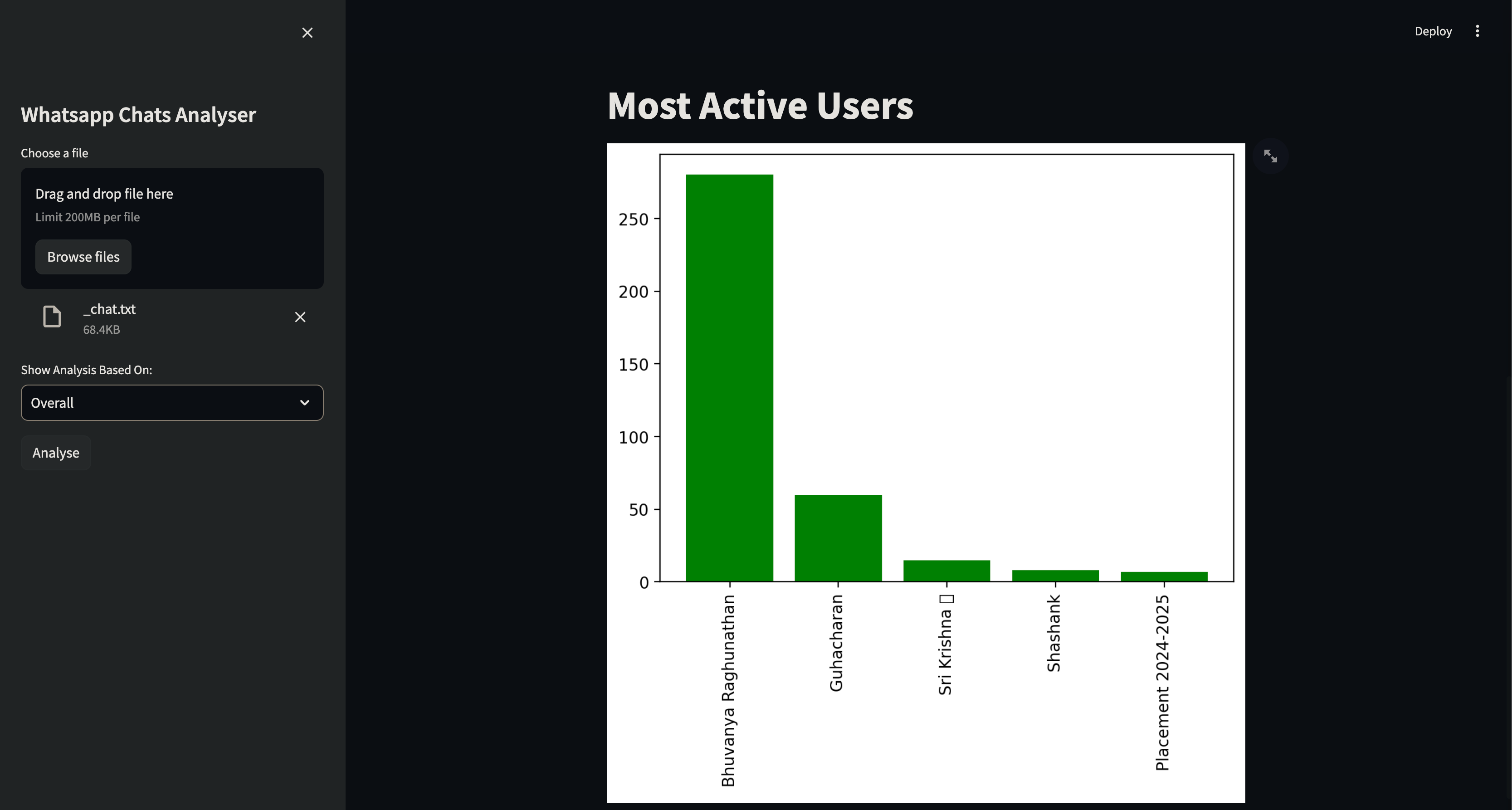
B. Monthly Timeline

The monthly timeline graph highlights trends in user activity over time. It shows the number of messages sent each month, allowing users to identify periods of high or low activity.



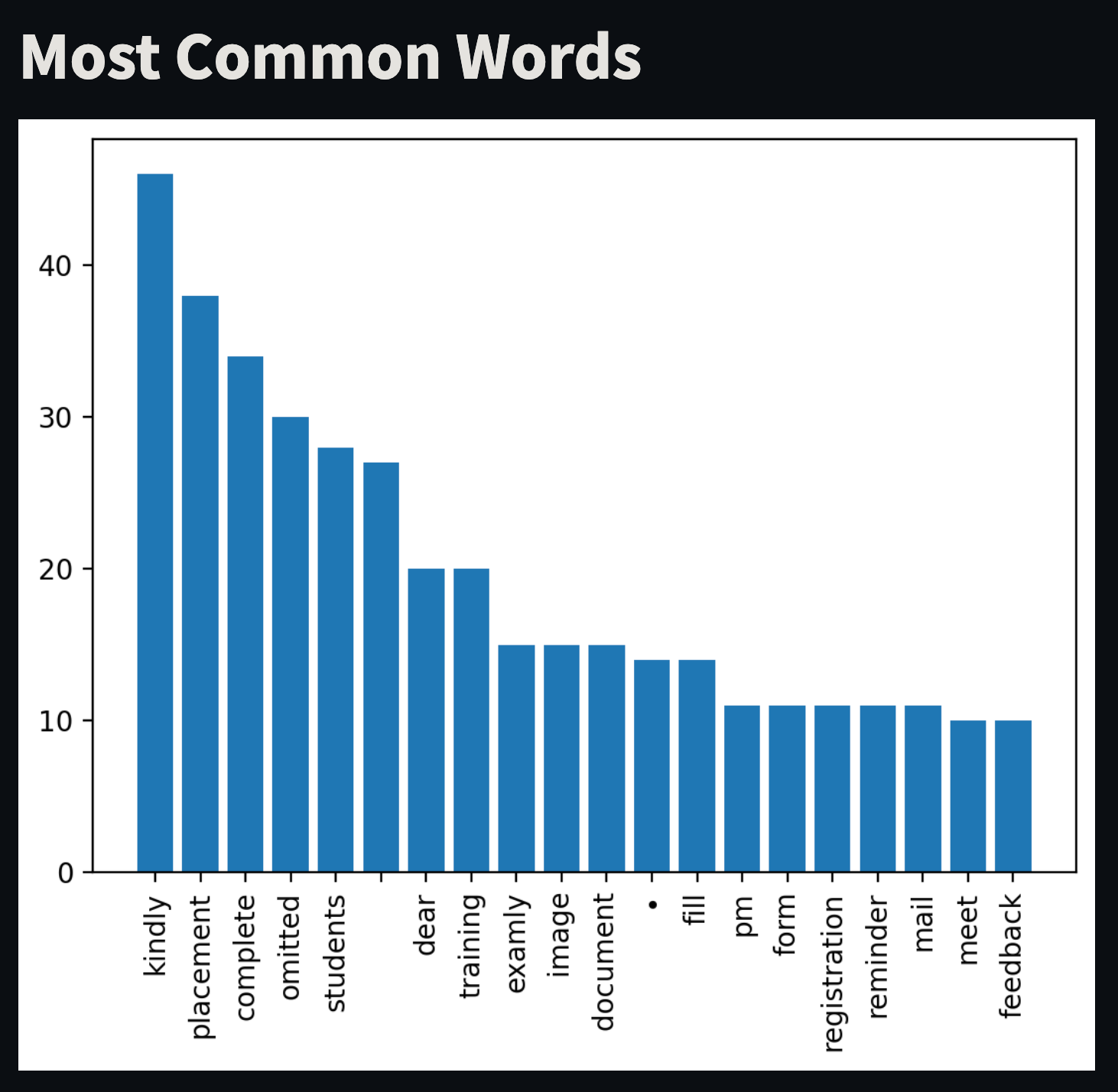
C. Most Active Users

The most active users chart identifies the participants who contribute the most to the conversation. This chart is useful for understanding the dynamics of the chat group and identifying key contributors.



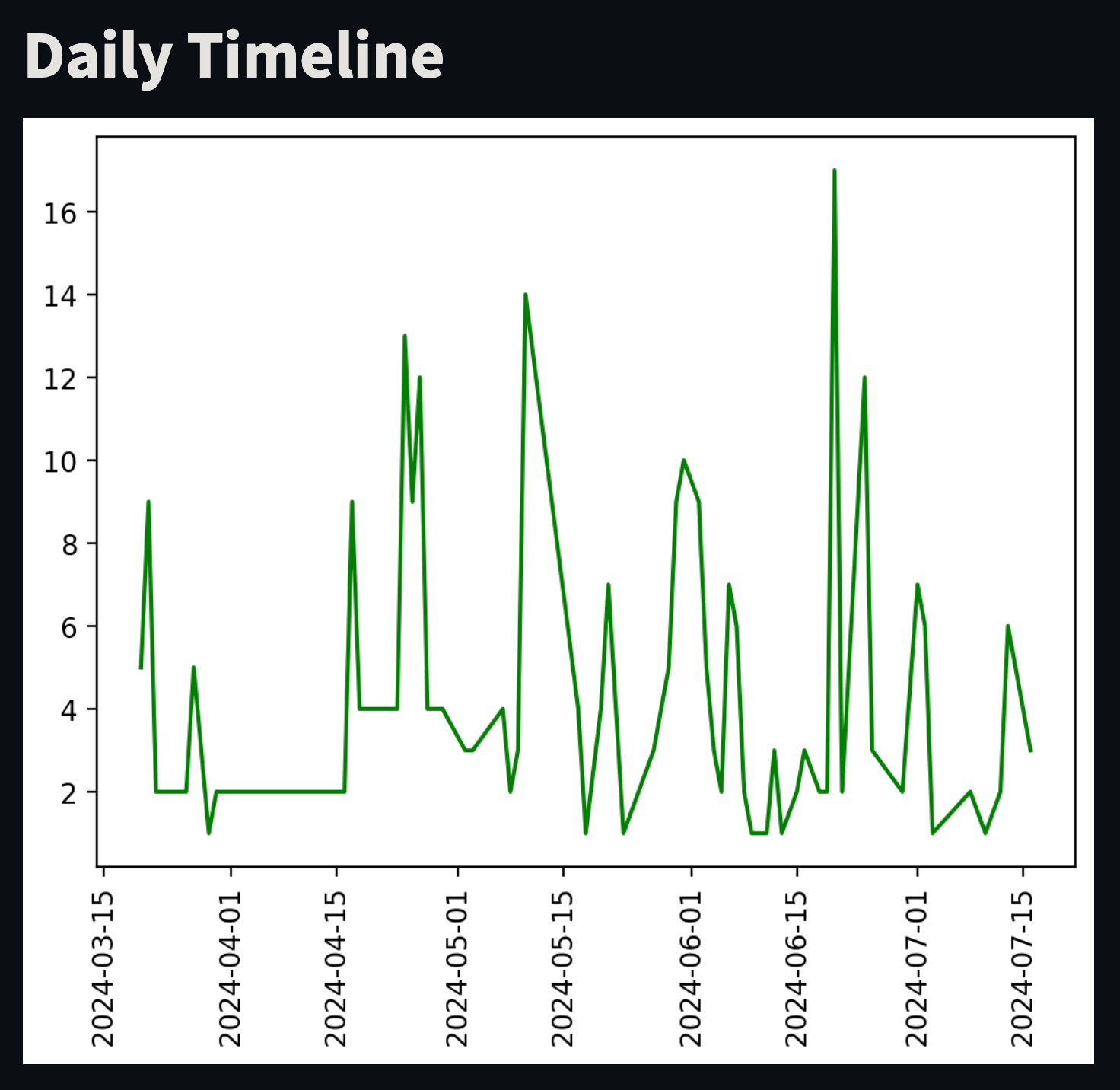
D. Most Common Words

The most common words chart provides insights into the topics and themes prevalent in the chat. By excluding common stop words, this chart highlights the specific words that are most frequently used by the participants.



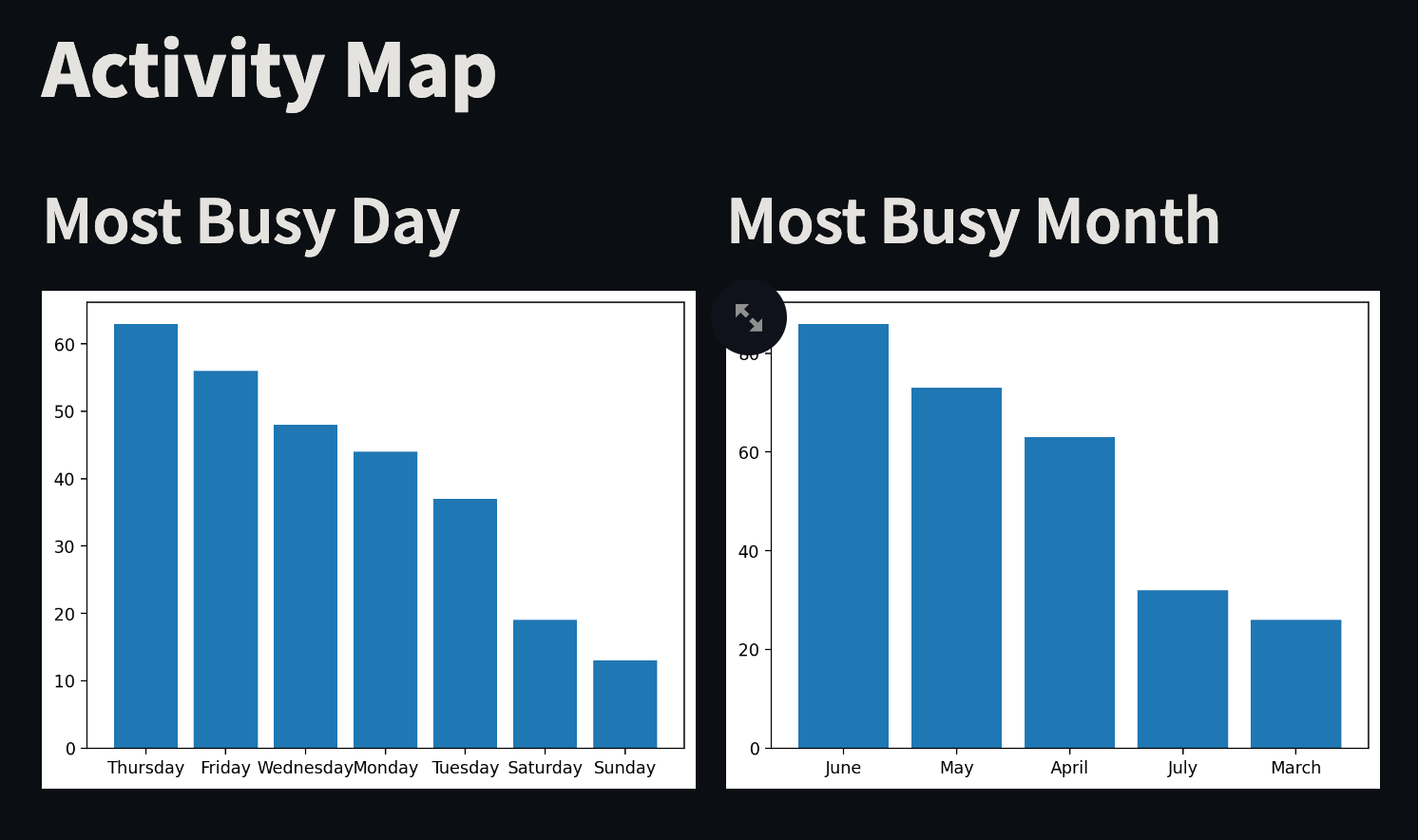
E. Daily Timeline

The daily timeline graph shows the number of messages sent each day, offering a detailed view of user activity over time. This graph helps in understanding daily communication patterns and identifying any recurring trends.



F. Activity Map

The activity maps show the most active days of the week and the busiest months. These maps provide a visual representation of when the chat group is most active, helping users understand temporal communication patterns.



IV. Discussion

The WhatsApp analyser provides a user-friendly interface for extracting and visualizing chat data. It allows users to gain insights into their communication patterns, identify active participants, and understand content trends. The application can be extended to include more advanced analytics and support for other messaging platforms.

The analysis of WhatsApp chat data can have various applications, including monitoring group activities, understanding social dynamics, and even for academic research in fields like linguistics and social sciences. The visualizations generated by the application provide a clear and concise summary of chat data, making it easier for users to interpret and derive meaningful conclusions.

One of the key features of the application is its ability to handle large datasets efficiently. This ensures that users can analyze extensive chat histories without facing performance issues. The use of pandas for data manipulation and matplotlib for visualization ensures that the application is both powerful and flexible.

Future improvements could include sentiment analysis to determine the emotional tone of messages, network analysis to map relationships between participants, and integration with other data sources for more comprehensive insights. Additionally, enhancing the application’s ability to handle different languages and character sets would make it more versatile and useful for a global audience.

VI. Conclusion

The WhatsApp chat analyser developed using Streamlit provides an effective tool for analyzing and visualizing chat data. By preprocessing the chat logs and generating various metrics and visualizations, the application helps users understand their communication patterns and user interactions. The simplicity and interactivity of the application make it accessible to users with varying levels of technical expertise.

This project demonstrates the potential of data science techniques in extracting valuable insights from everyday communication data. The analyser not only serves as a practical tool for individuals and groups but also opens up possibilities for further research and development in the field of chat data analysis.

Acknowledgment

We would like to thank the developers of Streamlit, pandas, and matplotlib for providing the tools necessary to develop this application. We also acknowledge the support and guidance provided by our institution and peers throughout the development process.

References

1. McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 51-56).

2. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.

3. Streamlit. (2023). Streamlit Documentation. Retrieved from <https://docs.streamlit.io/>