PROJECT REPORT

CS322 : Data Science

TITLE : WhatsApp Chat Analyzer

GitHub Repository Link: https://github.com/pkp2207/Chat-Analysis-DS
Deployed Website Link: https://chat-analysis-ds.streamlit.app/

Submitted By:

U22CS008 Mohammed Fowzan

U22CS018 Suchi Desai

U22CS023 Param Pathak

U22CS047 Keya Patel

U22CS056 Diya Shah

B. Tech. Year- 3rd and Sem- VI
Division : A

(Year: 2022-26)

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY

Surat- 395007, Gujarat, India

2025

ABSTRACT

This project presents a fully interactive, web-based **WhatsApp Chat Analyzer** built using **Streamlit** and Python, enabling users to upload and explore their WhatsApp chats through dynamic visualizations and intelligent metrics. The analyzer processes exported chat logs using regular expression parsing and timestamp formatting to convert unstructured text into structured data.

Leveraging powerful data analysis libraries such as **Pandas**, **Matplotlib**, and **Seaborn**, the tool provides insights into messaging behavior, frequency trends, and user engagement. Advanced **Natural Language Processing (NLP)** techniques, including **TF-IDF** analysis, are employed to identify distinct vocabulary usage across users, while integrated **Sentiment Analysis** detects the emotional tone of messages.

The application supports group-wise and user-specific breakdowns, with features like emoji usage analytics, word clouds, and heatmaps for weekly activity. With a clean and modular structure, the analyzer not only serves as a robust academic project but also highlights the practical application of data science in social communication analysis.

INTRODUCTION

WhatsApp is one of the most widely used messaging platforms globally, boasting over two billion active users. It facilitates billions of conversations daily, ranging from casual personal chats to critical professional communication. These chat logs, though informal, contain rich information about human interaction patterns, message dynamics, emotional tone, and user behavior.

Analyzing WhatsApp chats offers valuable insights not just from a data science perspective but also from behavioral, psychological, and business standpoints. For instance, businesses can study customer interactions for feedback analysis, digital forensic teams can investigate conversation timelines, and social scientists can analyze group behavior and emotional trends over time. With the rise of text-based digital communication, chat analytics is becoming increasingly relevant in both research and industry.

Real-Life Use Cases of Chat Analytics:

- **Behavioral Analysis**: Understanding individual or group communication styles.
- **Mental Health Insights**: Tracking emotional tone shifts over time.
- **Forensics and Investigations**: Tracing communication during specific events.
- **Marketing and Feedback**: Analyzing customer sentiment from chat transcripts.
- **Educational Research**: Studying collaborative learning or group study behavior.

PROBLEM STATEMENT

Exported WhatsApp chat data, though rich in information, is inherently **unstructured** and inconsistent across platforms (Android vs. iPhone). These exported logs often include system messages, varied timestamp formats, embedded media indicators, and informal or emoji-heavy text — making them difficult to analyze using traditional tools. Parsing and interpreting such raw data requires significant preprocessing and domain-specific logic.

Despite the increasing importance of chat analytics in domains such as digital forensics, behavioral research, and social network analysis, there is a lack of **accessible, intelligent, and visually-driven tools** for individuals and non-technical users. Existing solutions are either overly technical or fail to deliver actionable insights.

Challenges:

- Parsing multi-format exported chats with inconsistent syntax
- Distinguishing real messages from system or media notices
- Handling emojis, special characters, and non-English text
- Presenting findings in a way that is both **meaningful and visually intuitive**

Project Goal:

The primary goal of this project is to build a **modular, visual, and interactive WhatsApp Chat Analyzer** using Python and Streamlit that:

- Automatically **parses and structures** exported chat logs
- Performs advanced **text and sentiment analysis**
- Offers **insightful visualizations** of communication patterns

TOOLS AND TECHNOLOGIES USED

This project is developed using **Python** and leverages a suite of powerful libraries and frameworks for data processing, visualization, and web-based interaction.

Core Technologies

- **Python**: Chosen for its simplicity and rich ecosystem in data science and wenb development.

Data Processing & Analysis

- **Pandas**: For structuring, filtering, and analyzing chat data.
- **re (Regular Expressions)**: Used to parse and extract meaningful components (timestamps, usernames, messages) from raw text.

Visualization

- **Matplotlib & Seaborn**: For creating line plots, bar charts, and activity heatmaps.
- **WordCloud**: To visually represent frequently used words.

Natural Language Processing

- **TF-IDF (via Scikit-learn)**: Identifies the most significant and unique terms per user.
- **TextBlob / NLTK / VADER**: Used for sentiment analysis to classify messages as positive, neutral, or negative.

Web Interface

- **Streamlit**: Converts Python scripts into an interactive web app with file uploads, dynamic visuals, and real-time user input.

Development Tools

- **VS Code / Jupyter Notebook**: For writing, testing, and debugging code.
- **Git**: For version control and collaboration.

SYSTEM ARCHITECTURE / PROJECT STRUCTURE

```
WhatsApp-Chat-Analyzer/
│
├── .devcontainer/        # (Optional) Development environment configuration
├── app.py                # Streamlit app – main entry point, handles UI and logic
├── helper.py             # Core analysis functions – TF-IDF, sentiment, plots
├── preprocessor.py       # Preprocessing raw chat data into structured format
├── requirements.txt      # List of dependencies for easy setup
├── stopword.txt          # List of stopwords used to clean textual data
├── nltk.txt              # Optional: may contain token or NLP-related text data
├── cmd.txt               # Optional: command reference or logs
├── README.md             # Project documentation
├── LICENSE               # Project license
├── .gitignore            # Git ignore rules
└── .gitpod.yml           # Gitpod setup for cloud IDE development
```

Functional Modules

1. `app.py` – **Frontend Interface**
   - Built with **Streamlit**
   - Handles file uploads, user selections, and display of all analytics
2. `preprocessor.py` – **Data Structuring**
   - Detects chat format (Android/iPhone)
   - Parses timestamps, senders, and messages
   - Extracts time-based features (hour, day, month)
3. `helper.py` – **Analysis & Visualization**
   - Statistical summaries (messages, media, links)
   - **TF-IDF scoring**

    - **Sentiment analysis**
    - Plots: bar charts, timelines, heatmaps, emoji charts, word clouds

4. **`stopword.txt` – Language Cleaning**
    - List of common stopwords to exclude during word analysis
5. **`requirements.txt` – Environment Setup**
    - Contains all necessary packages for running the project

Workflow Overview

1. **User uploads** exported WhatsApp `.txt` file
2. `preprocessor.py` converts it into a structured DataFrame
3. `helper.py` processes the data to extract insights
4. **Streamlit app (`app.py`)** renders the results visually and interactively

DATA COLLECTION

The WhatsApp Chat Analyzer processes chat data exported directly from the WhatsApp mobile application. The chat export feature allows users to generate a `.txt` file containing the full conversation history from an individual or group chat.

Data Source

- Exported via **WhatsApp > Chat > Export Chat**
- Two possible formats based on device:
  - **Android**: `dd/mm/yyyy, hh:mm - Name: message`
  - **iPhone**: `[dd/mm/yyyy, hh:mm:ss] Name: message`

Users can choose to export chats **with or without media**. This project is designed to handle **text-only exports**, where media files are replaced with placeholders like `<Media omitted>`.

Characteristics of Raw Data

- Includes system notifications (e.g., "Messages are end-to-end encrypted")
- Messages may contain emojis, links, or special characters
- Supports multilingual text (dependent on system encoding)
- Timestamp accuracy is preserved in both formats

Initial Filtering

- Raw `.txt` file is uploaded via the Streamlit interface
- The file is read and decoded (`UTF-8`)

- `preprocessor.py` filters and parses the content using **regex** to extract:
    - **Date & Time**
    - **Sender (User)**
    - **Message content**

Invalid or malformed entries are ignored to ensure the accuracy of the structured data.

DATA PREPROCESSING

WhatsApp chat exports are raw, unstructured text files that vary in format based on the device used. To transform this data into a usable format, a dedicated preprocessing module was developed to extract timestamps, usernames, and messages while also generating additional time-based features.

Format Handling

The preprocessor intelligently detects and processes two common formats:

- **Android**: `dd/mm/yyyy, hh:mm - Name: message`
- **iPhone**: `[dd/mm/yyyy, hh:mm:ss] Name: message`

Using Python's `re` (regular expressions) module, the relevant parts of each message—**date**, **user**, and **message text**—are extracted accurately regardless of the format.

Data Transformation

Once parsed, the data is structured into a `Pandas DataFrame` with the following primary columns:

- `date` – timestamp of the message
- `user` – sender name
- `user_message` – actual text message

The module also filters out incomplete or malformed lines by dropping rows with invalid date values.

Time-based Feature Engineering

To enable time-series analysis, the following additional features are derived from the message timestamps:

- `year`, `month`, `month_num`
- `only_date` (YYYY-MM-DD format)
- `day`, `day_name` (e.g., Monday)
- `hour`, `minute`
- `period` – Time intervals like `14-15`, used for heatmap generation

These features are critical for identifying **daily patterns**, **peak activity hours**, and **seasonal trends** within conversations.

Cleaning Considerations

- System messages (e.g., “Messages and calls are end-to-end encrypted”) are excluded from user lists
- Media messages (`<Media omitted>`) are identified but not processed further
- Multi-line messages are handled by associating them with the correct sender using the initial timestamped line

FEATURE-WISE ANALYSIS

The core of the WhatsApp Chat Analyzer lies in its ability to extract meaningful patterns, trends, and metrics from structured chat data. Below is a breakdown of each feature provided by the system.

9.1 Message Statistics

- Calculates the total number of messages in the selected chat
- Counts total words exchanged
- Identifies number of media messages shared (e.g., images, videos)
- Tracks links shared across messages

9.2 TF-IDF Term Significance

- Implements Term Frequency-Inverse Document Frequency to score words based on uniqueness and frequency
- Highlights distinct vocabulary patterns for individual users
- Visualized using bar charts and detailed data tables

9.3 Timeline Analysis

- **Monthly Timeline**: Plots message trends over each month to detect long-term engagement or inactivity
- **Daily Timeline**: Tracks day-by-day message frequency, ideal for identifying specific event-related activity

9.4 Activity Mapping

- **Most Busy Day**: Identifies which days of the week have the highest message volume
- **Most Busy Month**: Highlights months with peak activity

- **Heatmap**: Displays user activity based on hour and weekday, helping identify active periods

9.5 User-Based Analysis

- Displays most active participants in a group chat
- Ranks users based on message contribution
- Supports both visual (bar chart) and tabular comparisons

9.6 Word Cloud

- Generates a visual representation of the most frequently used words
- Stopwords are removed using a custom list to improve relevance
- Helps understand dominant conversation topics

9.7 Most Common Words

- Lists top recurring words in the chat along with their frequencies
- Displayed using a horizontal bar graph
- Offers insights into frequently discussed themes or expressions

9.8 Emoji Analysis

- Detects and counts emojis used across the chat
- Displays a pie chart of top emojis and a table of emoji usage frequency
- Useful for understanding tone and sentiment in informal conversations

9.9 Sentiment Analysis

- Categorizes messages as positive, neutral, or negative using NLP libraries
- Displays overall sentiment distribution as metrics

- Includes sentiment trends over time and comparison across users
- Filters high-confidence sentiment-tagged messages for review

VISUALIZATIONS

Data visualization plays a key role in transforming raw chat data into digestible insights. The WhatsApp Chat Analyzer integrates a variety of chart types to represent message trends, user behaviors, and conversational dynamics in a meaningful way.

Visualization Techniques Used

- **Bar Charts**
    - Used to display:
        - Most active users
        - Most common words
        - Activity by day and month
- **Line Graphs**
    - Employed in:
        - Monthly timeline analysis
        - Daily message trends
    - Helps visualize communication patterns over time
- **Heatmaps**
    - Visualizes user activity across days of the week and hours of the day
    - Highlights peak usage periods in a matrix format
- **Word Cloud**
    - Displays the most frequently used words in a visually engaging format

    - Emphasizes more frequent words with larger font size
    - Filters out stopwords for clarity
- **Pie Charts**
    - Used for:
        - Emoji distribution analysis
        - Sentiment breakdown (if applicable)
    - Offers a high-level overview of usage proportions
- **Tables and DataFrames**
    - Used to support all analytical visuals with raw numbers
    - Especially useful in TF-IDF scoring, sentiment-tagged message samples, and user-wise activity logs

Tools Used for Visualization

- **Matplotlib**: Core plotting library used for all static charts
- **Seaborn**: Applied for aesthetically styled charts and heatmaps
- **Streamlit**: Facilitates interactive rendering of visuals within the web app interface

Each visualization is dynamically generated based on user selection, ensuring flexibility in individual and group-level analysis.

CHALLENGES FACED

While building the WhatsApp Chat Analyzer, several challenges emerged in terms of data handling, user experience, and feature implementation. These were addressed through custom logic, iterative testing, and the use of robust libraries.

1. Parsing Multi-Format Chat Logs

- WhatsApp exports differ between Android and iPhone, with variations in timestamp formats and line structure.
- Solution: Implemented dual regex patterns to detect and process both formats automatically.

2. Handling Multi-Line and Irregular Messages

- Messages that span multiple lines or include special formatting (e.g., forwarded content) required careful reconstruction.
- Solution: Associated continuation lines with the last valid timestamp to maintain message integrity.

3. Filtering Out System Notifications

- System-generated texts such as encryption notices and group updates needed exclusion to prevent skewing the data.
- Solution: Filtered such entries during preprocessing based on known patterns.

4. Timezone and Encoding Variability

- Some exported chats contained encoding issues or locale-specific date/time formats.
- Solution: Standardized decoding using UTF-8 and used `errors='coerce'` in `pandas.to_datetime()` to manage parsing errors gracefully.

5. Large Chat Files and Performance Bottlenecks

- Group chats with thousands of messages slowed down rendering and computations.
- Solution: Optimized DataFrame operations and used selective rendering in Streamlit to improve performance.

6. Sentiment Analysis on Informal Text

- Short, emoji-heavy or slang-filled messages posed difficulties for NLP models.
- Solution: Applied confidence-based filtering to only display high-certainty sentiment predictions.

CONCLUSION

The WhatsApp Chat Analyzer successfully transforms unstructured chat data into meaningful, actionable insights through a combination of natural language processing, statistical analysis, and intuitive visualizations. Built with Python and Streamlit, the tool provides a seamless user experience for both group-level and individual-level chat analysis.

By automating the parsing, structuring, and evaluation of exported WhatsApp messages, this project demonstrates the practical application of data science in analyzing real-world communication patterns. From identifying the most active users to exploring sentiment trends and text uniqueness via TF-IDF, the analyzer offers a comprehensive understanding of conversational behavior.

The modular and extensible design ensures that the tool can be adapted for additional platforms or enhanced with new analytical capabilities. Whether used for academic research, behavioral analysis, or personal curiosity, the WhatsApp Chat Analyzer is a functional and insightful solution for modern text analytics.

FUTURE SCOPE

While the WhatsApp Chat Analyzer currently provides a robust foundation for static chat analysis, there are several opportunities to enhance its capabilities and expand its applicability in future iterations:

1. Real-Time Chat Monitoring

- Integrate with **WhatsApp Web APIs** or browser automation tools to enable real-time message tracking and live updates.

2. Cross-Platform Support

- Extend parsing logic to support chat exports from other messaging platforms such as:
    - **Telegram**
    - **Facebook Messenger**
    - **Discord**

3. Enhanced NLP Capabilities

- Use advanced models like **spaCy**, **BERT**, or **transformer-based sentiment classifiers** for deeper emotion and intent detection.
- Add **topic modeling** (e.g., LDA) to uncover conversation themes.

4. Interactive Dashboards

- Export analyzed results to **PDF reports** or **dashboard interfaces** using tools like Plotly Dash or Tableau for non-technical users.

5. User Behavior Profiling

- Implement **user clustering**, **engagement scoring**, or **influence mapping** based on interaction patterns and message types.

6. Privacy and Anonymization Features

- Add functionality to **anonymize usernames** and redact sensitive content for privacy-preserving analysis and safe public sharing.

REFERENCES

Python Libraries & Documentation

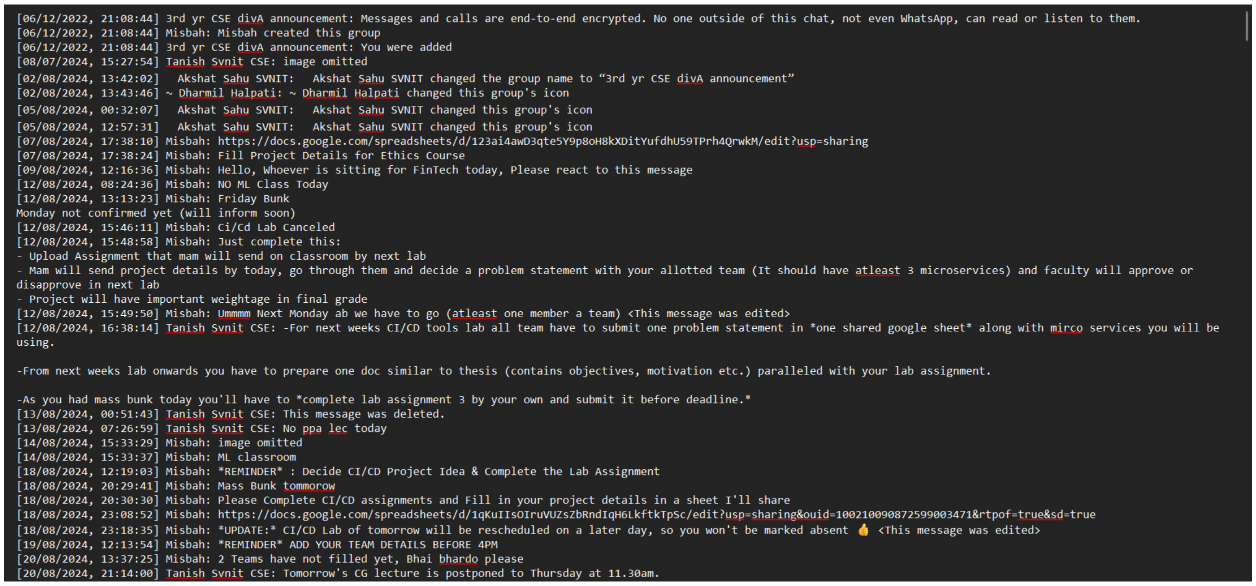
- Pandas Documentation
- Matplotlib Documentation
- Seaborn Documentation
- Streamlit Documentation
- Scikit-learn: TF-IDF Vectorizer
- WordCloud Library
- TextBlob
- NLTK
- VADER Sentiment Analysis (NLTK)

APPENDIX

A. Sample Chat Input

- Screenshot or snippet of a sample WhatsApp `.txt` file (Android/iPhone export)

- Demonstrates structure before parsing



```
[06/12/2022, 21:08:44] 3rd yr CSE divA announcement: Messages and calls are end-to-end encrypted. No one outside of this chat, not even WhatsApp, can read or listen to them.
[06/12/2022, 21:08:44] Misbah: Misbah created this group
[06/12/2022, 21:08:44] 3rd yr CSE divA announcement: You were added
[08/07/2024, 15:27:54] Tanish Svnit CSE: image omitted
[02/08/2024, 13:42:02]   Akshat Sahu SVNIT:   Akshat Sahu SVNIT changed the group name to “3rd yr CSE divA announcement”
[02/08/2024, 13:43:46] ~ Dharmil Halpati: ~ Dharmil Halpati changed this group's icon
[05/08/2024, 00:32:07]   Akshat Sahu SVNIT:   Akshat Sahu SVNIT changed this group's icon
[05/08/2024, 12:57:31]   Akshat Sahu SVNIT:   Akshat Sahu SVNIT changed this group's icon
[07/08/2024, 17:38:10] Misbah: https://docs.google.com/spreadsheets/d/123ai4awD3qte5Y9p8oH8kXDitYufdhU59TPrh4QrwkM/edit?usp=sharing
[07/08/2024, 17:38:24] Misbah: Fill Project Details for Ethics Course
[09/08/2024, 12:16:36] Misbah: Hello, Whoever is sitting for FinTech today, Please react to this message
[12/08/2024, 08:24:36] Misbah: NO ML Class Today
[12/08/2024, 13:13:23] Misbah: Friday Bunk
Monday not confirmed yet (will inform soon)
[12/08/2024, 15:46:11] Misbah: Ci/Cd Lab Canceled
[12/08/2024, 15:48:58] Misbah: Just complete this:
- Upload Assignment that mam will send on classroom by next lab
- Mam will send project details by today, go through them and decide a problem statement with your allotted team (It should have atleast 3 microservices) and faculty will approve or
disapprove in next lab
- Project will have important weightage in final grade
[12/08/2024, 15:49:50] Misbah: Ummmm Next Monday ab we have to go (atleast one member a team) <This message was edited>
[12/08/2024, 16:38:14] Tanish Svnit CSE: -For next weeks CI/CD tools lab all team have to submit one problem statement in *one shared google sheet* along with mirco services you will be
using.

-From next weeks lab onwards you have to prepare one doc similar to thesis (contains objectives, motivation etc.) paralleled with your lab assignment.

-As you had mass bunk today you'll have to *complete lab assignment 3 by your own and submit it before deadline.*
[13/08/2024, 00:51:43] Tanish Svnit CSE: This message was deleted.
[13/08/2024, 07:26:59] Tanish Svnit CSE: No ppa lec today
[14/08/2024, 15:33:29] Misbah: image omitted
[14/08/2024, 15:33:37] Misbah: ML classroom
[18/08/2024, 12:19:03] Misbah: *REMINDER* : Decide CI/CD Project Idea & Complete the Lab Assignment
[18/08/2024, 20:29:41] Misbah: Mass Bunk tommorow
[18/08/2024, 20:30:30] Misbah: Please Complete CI/CD assignments and Fill in your project details in a sheet I'll share
[18/08/2024, 23:08:52] Misbah: https://docs.google.com/spreadsheets/d/1qKuIIsOIruVUZsZbRndIqH6LkftkTpSc/edit?usp=sharing&ouid=100210090872599003471&rtpof=true&sd=true
[18/08/2024, 23:18:35] Misbah: *UPDATE:* CI/CD Lab of tomorrow will be rescheduled on a later day, so you won't be marked absent 👍 <This message was edited>
[19/08/2024, 12:13:54] Misbah: *REMINDER* ADD YOUR TEAM DETAILS BEFORE 4PM
[20/08/2024, 13:37:25] Misbah: 2 Teams have not filled yet, Bhai bhardo please
[20/08/2024, 21:14:00] Tanish Svnit CSE: Tomorrow's CG lecture is postponed to Thursday at 11.30am.
```

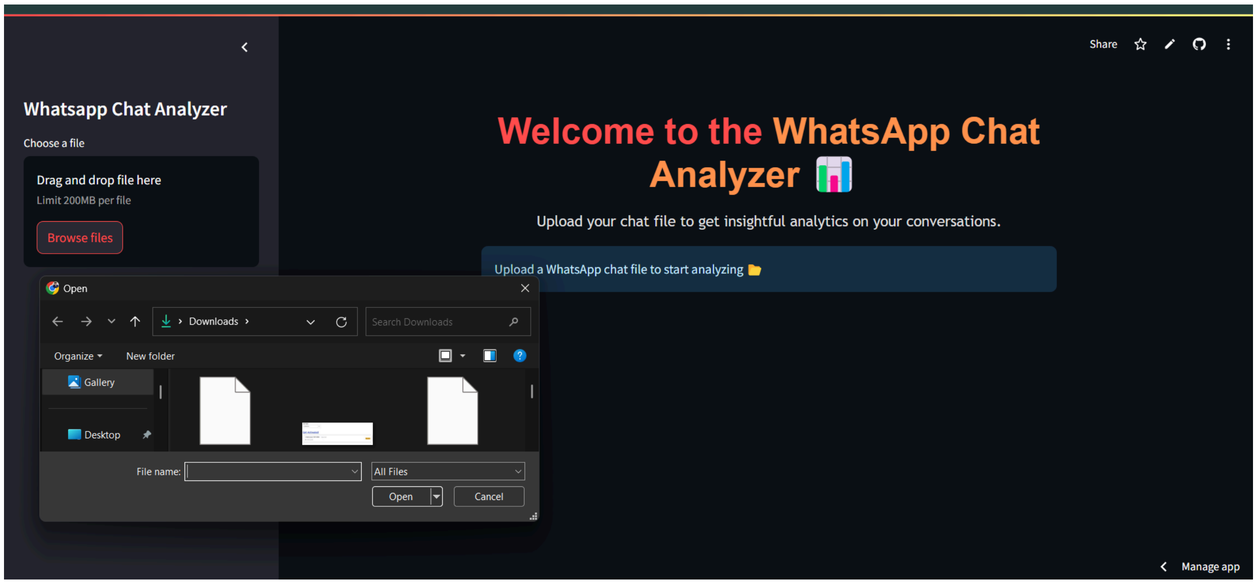
B. Preprocessed Data Sample

- Example `DataFrame` showing columns like:
    - `date`, `user`, `user_message`, `month`, `day_name`, `hour`, `period`

- Helps visualize how raw data was transformed

C. Interface Screenshots

- Streamlit UI: file upload, user selection dropdown

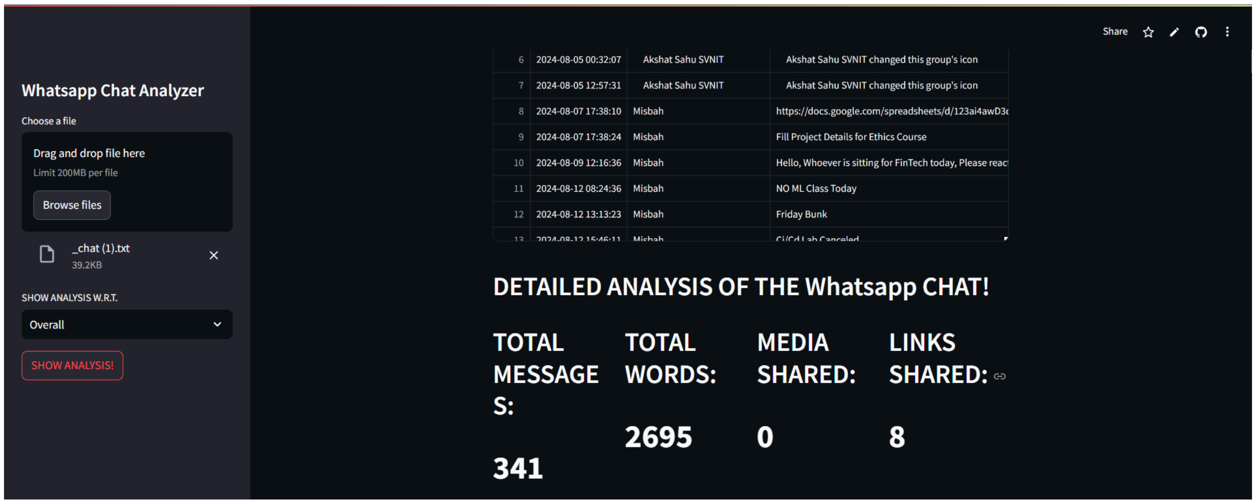


Welcome to the WhatsApp Chat Analyzer 📊

Upload your chat file to get insightful analytics on your conversations.

Upload a WhatsApp chat file to start analyzing 📂

- Visual sections for:
  - Message statistics

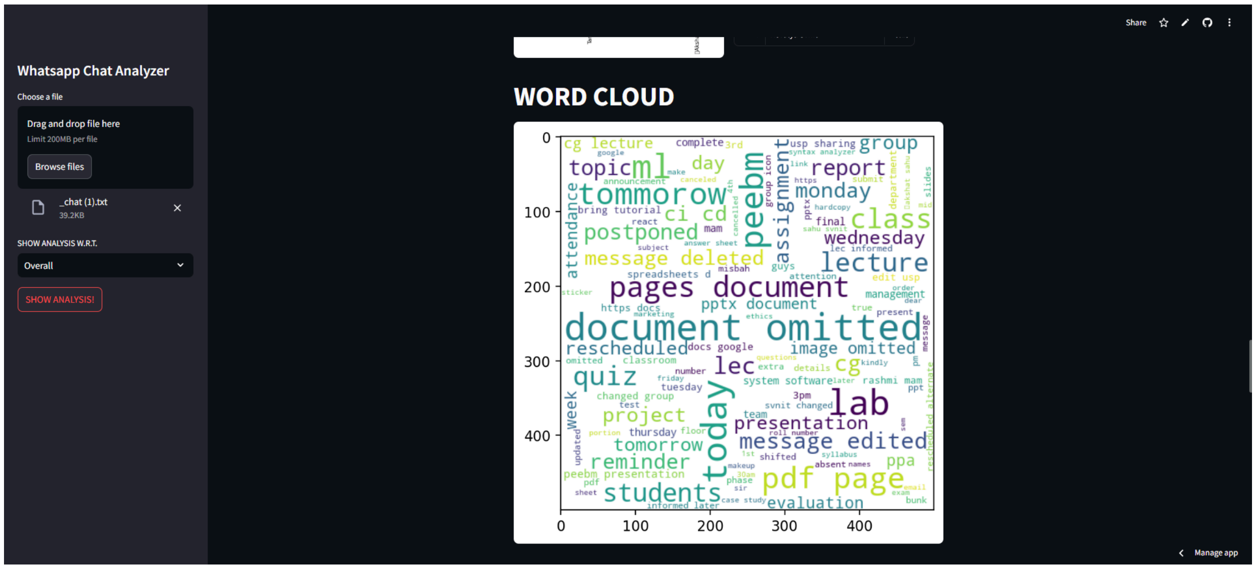


6	2024-08-05 00:32:07	Akshat Sahu SVNIT	Akshat Sahu SVNIT changed this group's icon
7	2024-08-05 12:57:31	Akshat Sahu SVNIT	Akshat Sahu SVNIT changed this group's icon
8	2024-08-07 17:38:10	Misbah	https://docs.google.com/spreadsheets/d/123ai4awD3c
9	2024-08-07 17:38:24	Misbah	Fill Project Details for Ethics Course
10	2024-08-09 12:16:36	Misbah	Hello, Whoever is sitting for FinTech today, Please react
11	2024-08-12 08:24:36	Misbah	NO ML Class Today
12	2024-08-12 13:13:23	Misbah	Friday Bunk
13	2024-08-12 15:46:11	Misbah	Ci/Cd Lab Canceled

DETAILED ANALYSIS OF THE Whatsapp CHAT!

TOTAL MESSAGES:	TOTAL WORDS:	MEDIA SHARED:	LINKS SHARED:
341	2695	0	8

  - Word cloud



Whatsapp Chat Analyzer

Choose a file

Drag and drop file here
Limit 200MB per file

Browse files

_chat (1).txt 39.2KB

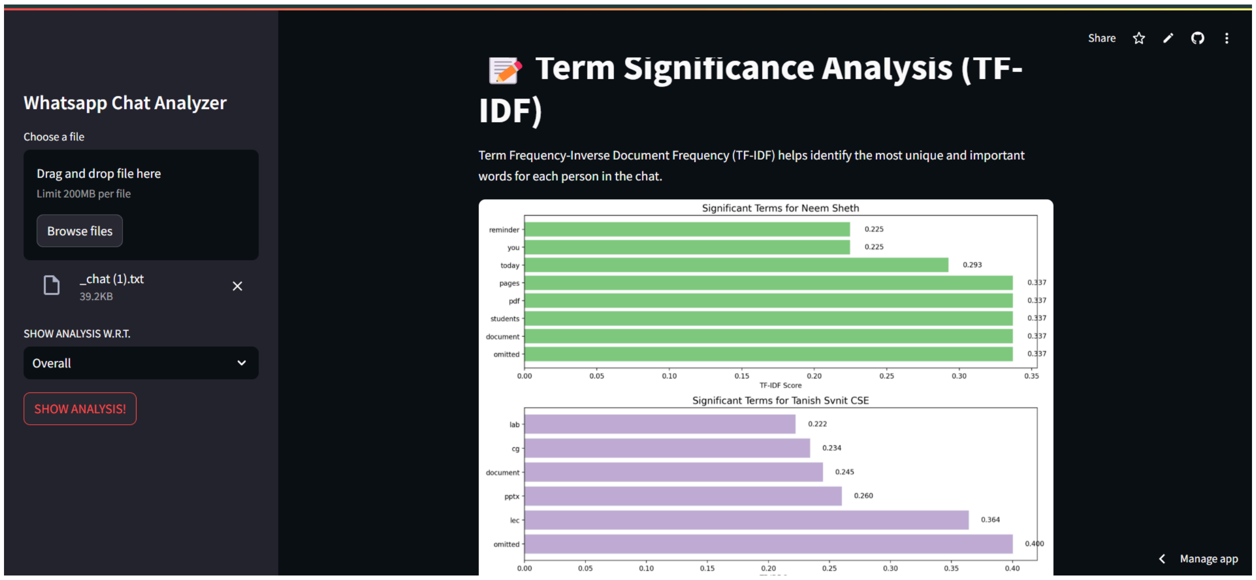
SHOW ANALYSIS W.R.T.

Overall

SHOW ANALYSIS!

WORD CLOUD

- TF-IDF bar chart



Whatsapp Chat Analyzer

Choose a file

Drag and drop file here
Limit 200MB per file

Browse files

_chat (1).txt 39.2KB

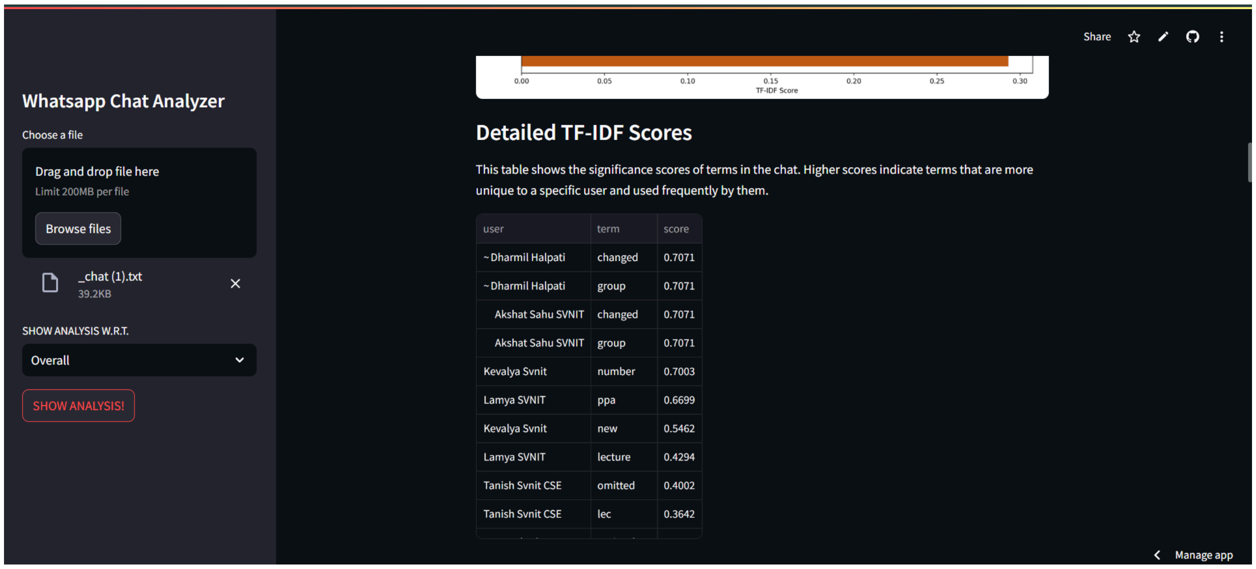
SHOW ANALYSIS W.R.T.

Overall

SHOW ANALYSIS!

📝 Term Significance Analysis (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) helps identify the most unique and important words for each person in the chat.

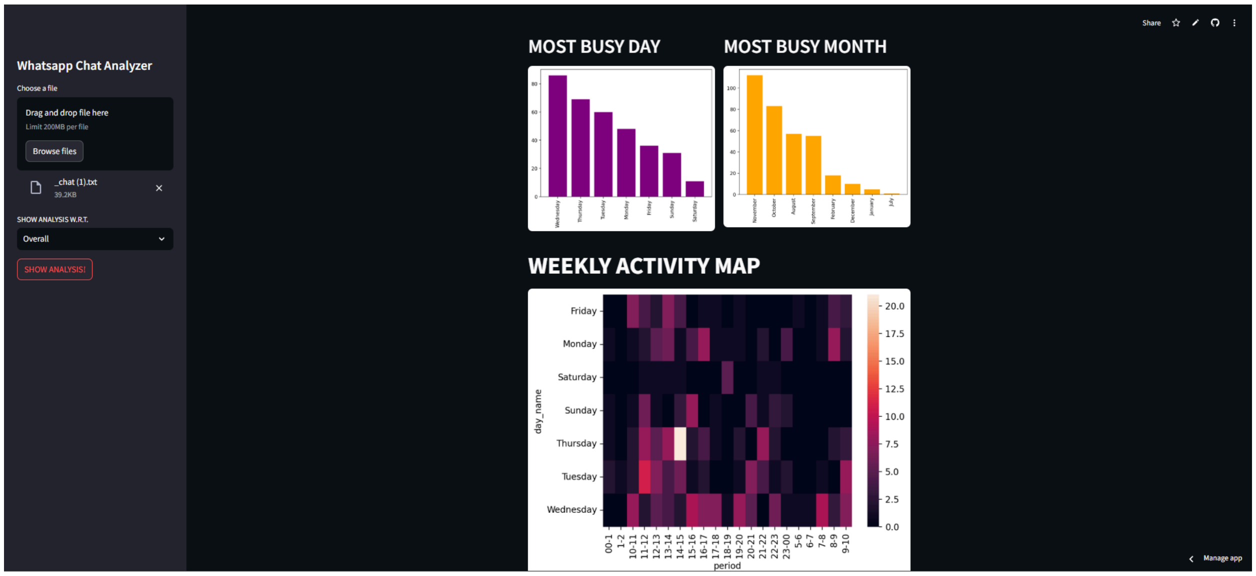


Detailed TF-IDF Scores

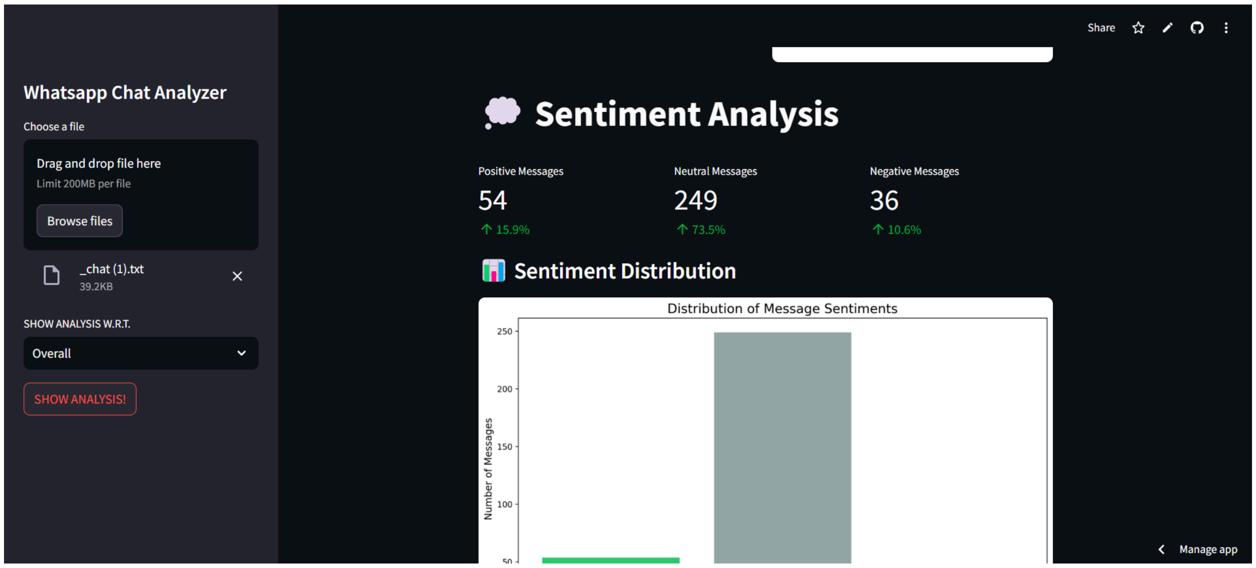
This table shows the significance scores of terms in the chat. Higher scores indicate terms that are more unique to a specific user and used frequently by them.

user	term	score
~ Dharmil Halpati	changed	0.7071
~ Dharmil Halpati	group	0.7071
Akshat Sahu SVNIT	changed	0.7071
Akshat Sahu SVNIT	group	0.7071
Kevalya Svnit	number	0.7003
Lamya SVNIT	ppa	0.6699
Kevalya Svnit	new	0.5462
Lamya SVNIT	lecture	0.4294
Tanish Svnit CSE	omitted	0.4002
Tanish Svnit CSE	lec	0.3642

○ Activity heatmap



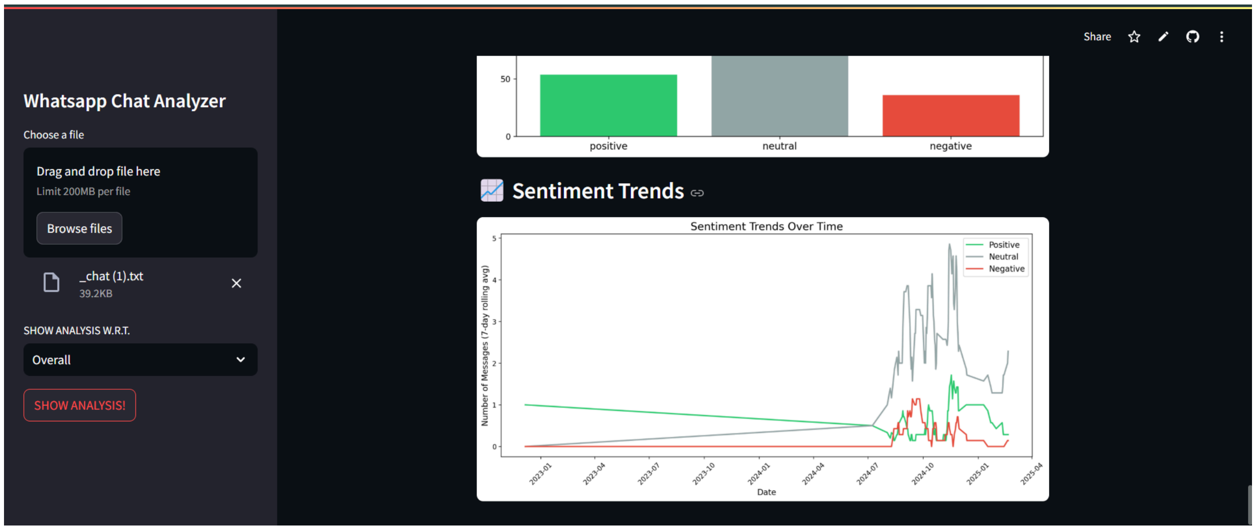
○ Sentiment analysis panel



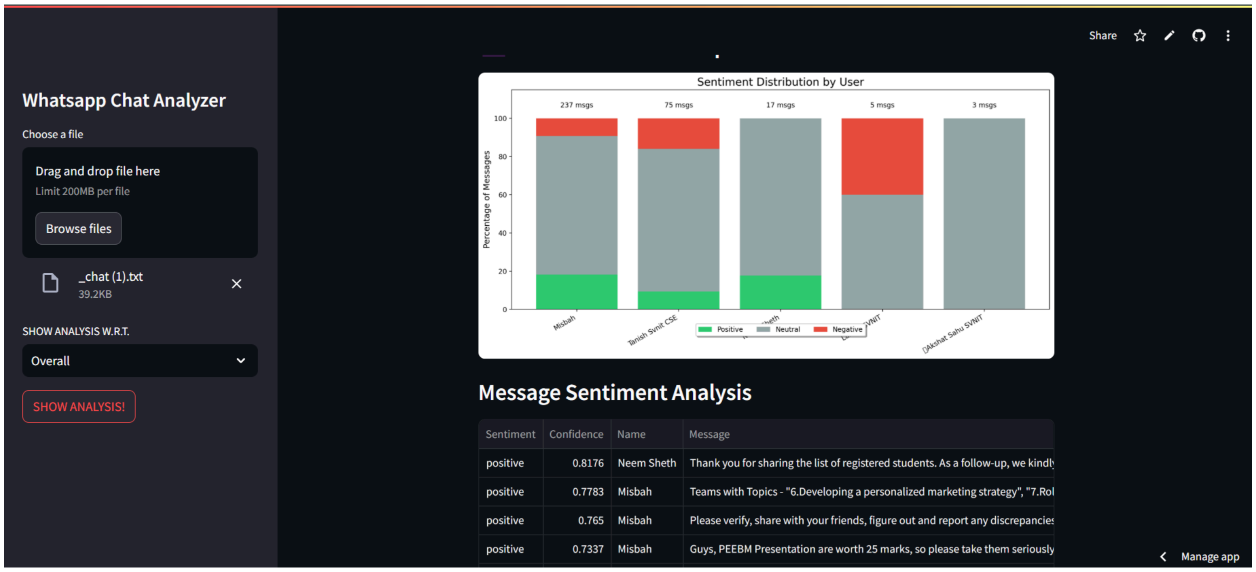
Sentiment Analysis

Positive Messages	Neutral Messages	Negative Messages
54	249	36
↑ 15.9%	↑ 73.5%	↑ 10.6%

Sentiment Distribution



Sentiment Trends



Message Sentiment Analysis

Sentiment	Confidence	Name	Message
positive	0.8176	Neem Sheth	Thank you for sharing the list of registered students. As a follow-up, we kindly
positive	0.7783	Misbah	Teams with Topics - "6.Developing a personalized marketing strategy", "7.Rol
positive	0.765	Misbah	Please verify, share with your friends, figure out and report any discrepancies
positive	0.7337	Misbah	Guys, PEEBM Presentation are worth 25 marks, so please take them seriously

D. Code Snippets

- Key logic from:
  - `preprocessor.py` (regex + DataFrame creation)
  - `helper.py` (TF-IDF, sentiment, emoji)
  - `app.py` (Streamlit layout and control flow)

E. Requirements File

- List of dependencies for easy project setup using `requirements.txt`

```
streamlit
pandas
matplotlib
seaborn
wordcloud
textblob
scikit-learn
nltk
```