WHATSAPP CHAT ANALYZER USING PYTHON & ML(Regression)

A PROJECT REPORT

Submitted by

Arvind Gupta

Shilpa Kumari

Table of Contents

CHAPTER 1

INTRODUCTION

This tool is based on data analysis and processing. The first step in implementing a machine learning algorithms is to understand the right learning experience from which the model starts improving. Data preprocessing plays a major role when it comes to machine learning. In order to make the model more efficient we need lots of data, we turned our focus primarily on one of the large-scale data producers owned by Facebook which is nothing but WhatsApp. WhatsApp claims that nearly 55 billion messages are sent each day. The average user spends 195 minutes per week on WhatsApp, and is member of plenty groups. With this treasure house of data right under very noisy, it is imperative that we embark on a mission to gain insight on the messages which our phones are forced to bear witness to. A list that uses pie charts and diagrams to represent the interesting data that it collects after analyzing your WhatsApp chats. You know the drill by now. You will take a backup of your chat to an email id listed on the site.

WhatsApp chat Analyzer focus on:

- The WhatsApp Chat Analyzer project is designed to transform raw chat data from WhatsApp conversations into insightful metrics and visualizations.

- With the growing use of WhatsApp as a primary communication tool, understanding the patterns and behaviors within these conversations can provide valuable insights into social interactions, user activity, and communication trends.

- This project leverages the capabilities of Python and its extensive libraries to achieve this analysis effectively.

PROBLEM STATEMENT

WhatsApp-Analyzer statistical analysis tool for WhatsApp chats Working on the chat files that can be exported from WhatsApp it generates various plots showing for example, which participants active to the most. Communication between people using the internet becomes part of their daily life. People used to communicate with each other wing the online chat system to transfer their messages. We propose to employ dataset manipulation techniques to have a better understanding of WhatsApp chat present in our phones. It shows most used and word which repeatedly most times. It tracks our conversation and analyses how much time we are spending.

OBJECTIVE

- This project to provide a better understanding towards various types of chats. This analysis proves to be better input to machine learning models which essentially explore the chat data. It requires proper learning instances which provides better accuracy for these models. Our project ensures to provide an in-depth exploratory data analysis on various types of WhatsApp chats.
- Exporting text file of a WhatsApp chat does not require a lot of space and internet data.
- The metrics include the average number of messages per user, total messages, total users, average user time, most active user, word cloud, showing busiest day and analysis by date of chats and more.
- WhatsApp Chat Analyzer is more focused on emojis and word than other metrics but no less fun than other WhatsApp Chat Analyzer.
- This application does not store data. It ensures privacy and security of user's chat.

Chapter-2

REQUIREMENT ANALYSIS

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Requirement's analysis involves all the tasks that are conducted to identify the needs of different stakeholders.

Platform Specification

The Platform Initialization Specification (PT Specification) is a specification published by the Unified EFT Forum that describes the internal interfaces between different parts of computer platform firmware. This allows for more interoperability between firmware components from different sources.

System Specification

A System Requirements Specification is a structured collection of information that embodies the requirements a system.

Hardware specification

Describe logical and physical characteristics of each interface between the software and the hardware components of the system.

- **Hardware Required:**
  Any web browser supported device.

- **Supported device types:**
  The software is developed for Windows 32-bit/64-bit or android etc.

- **Nature of the data and control interactions between the software and the hardware:** Internet connection

Software Specification

- **The connections of your software with other operating systems:** the software is developed for all operating system.

The connections of your software with other libraries:

- Streamlit
- Numpy
- Pandas
- Wordcloud
- Seaborn
- Regression
- Matplotlib

Libraries and its Purpose: -

Streamlit:

Introduction to Streamlit:

Streamlit is an open-source Python library that simplifies the creation of custom web applications for data science and machine learning projects. It enables users to turn Python scripts into interactive web apps quickly and easily, without requiring extensive web development knowledge.

Purpose of Using Streamlit

1. **Interactive Data Visualization:**
    - **Purpose:** Create interactive and dynamic web applications.
    - **Detail:** Streamlit allows for the creation of interactive dashboards that can display chat data visualizations and allow users to explore the data in real-time.

2. **Real-Time Data Analysis:**
    - **Purpose:** Provide real-time data analysis and updates.
    - **Detail**: Streamlit can dynamically update visualizations and analyses as data changes, making it possible to interact with and explore the most current data.

3. **Integration with Python Libraries:**
    - **Purpose:** Seamlessly integrate with other data science libraries.
    - **Detail**: Streamlit works well with libraries like Pandas, NumPy, Matplotlib, and Seaborn, allowing for easy integration of data manipulation and visualization tools in the web app.

4. **User-Friendly Interface:**
    - **Purpose**: Offer an intuitive interface for users.
    - **Detail:** Streamlit provides widgets like sliders, buttons, and file uploaders that make it easy for users to interact with the data and customize their analysis.

5. **Deployment:**
    - **Purpose**: Facilitate easy deployment of data apps.
    - **Detail**: Streamlit applications can be deployed easily, allowing the chat Analyzer tool to be shared and accessed by others over the web.

NumPy

Introduction:

NumPy, short for Numerical Python, is a fundamental library for numerical computing in Python. It provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is widely used in data science, machine learning, and scientific computing.

Purpose of Using NumPy

1. **Handling Large Data:**
    - **Purpose**: Manage big sets of chat data efficiently.
    - **Detail**: NumPy can quickly handle and process large amounts of data, which is essential for analyzing extensive chat logs.

2. **Math Operations:**
    - **Purpose**: Perform calculations on the chat data.
    - **Detail:** NumPy makes it easy to do math operations like finding averages or totals, which helps in analyzing chat data.

3. **Data Transformation:**
    - **Purpose:** Clean and prepare the data for analysis.
    - **Detail:** NumPy helps in reshaping and preparing data quickly so it's ready for deeper analysis.

4. **Statistical Analysis**:
    - **Purpose:** Calculate statistics.
    - **Detail:** NumPy can easily compute statistics like mean and standard deviation to understand the chat data better.

5. **Speed and Performance:**
    - **Purpose**: Make calculations fast.
    - **Detail**: NumPy is built for speed, making it much faster than using regular Python for large data sets.

Pandas

Introduction:

Pandas is an open-source data manipulation and analysis library for Python, providing data structures and functions needed to work with structured data seamlessly. It is particularly well-suited for handling tabular data, such as data from spreadsheets or databases, and is widely used in data science, data analysis, and machine learning.

Purpose of Using Pandas

1. **Data Handling:**
    - **Purpose:** Manage and manipulate chat data easily.
    - **Detail**: Pandas makes it simple to read, clean, and organize large sets of chat data.

2. **Data Cleaning:**
    - **Purpose:** Prepare data for analysis.
    - **Detail**: Pandas provides tools to clean and preprocess data, such as removing unnecessary text and handling missing values.

3. **Data Analysis:**
    - **Purpose**: Perform detailed analysis.
    - **Detail**: Pandas allows for quick computation of statistics and aggregation of data to analyze patterns and trends in chat data.

4. **Data Transformation:**
    - **Purpose**: Reshape and reorganize data.
    - **Detail:** Pandas offers functionalities to pivot, merge, and reshape data, making it easier to prepare data for further analysis and visualization.

5. **Visualization Support:**
    - **Purpose:** Simplify data visualization.
    - **Detail:** Pandas works with Matplotlib and Seaborn to create plots directly from DataFrames, making it easier to visualize data insights.

Matplotlib

Introduction:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It provides a wide range of plotting functions and customization options, making it suitable for various visualization needs in data science, engineering, and scientific research.

Purpose of Using Matplotlib

1. **Data Visualization:**
    a. **Purpose**: Create visual representations of chat data.
    b. **Detail**: Matplotlib helps in generating plots and charts that make it easier to understand trends and patterns in the chat data.

2. **Insightful Analysis:**
    - **Purpose**: Gain insights from data through visual means.
    - **Detail**: Visualizing data can reveal insights that might not be obvious from raw data alone, such as peaks in messaging activity or patterns over time.

3. **Customization:**
    - **Purpose**: Customize the appearance of plots.

- **Detail**: Matplotlib provides extensive options to customize graphs, such as changing colors, labels, and styles, allowing for tailored visual presentations.

4. **Integration with Pandas:**
   - **Purpose**: Seamlessly integrate with Pandas DataFrames.
   - **Detail:** Matplotlib works well with Pandas, enabling direct plotting from DataFrames, which simplifies the process of creating visualizations from cleaned and processed data.

5. **Comparison and Trends:**
   - **Purpose:** Compare different data sets and observe trends.
   - **Detail:** Matplotlib allows for plotting multiple data sets on the same graph, making it easy to compare user activity or message frequency over different periods.

Seaborn

Introduction to Seaborn:

Seaborn is a Python data visualization library based on Matplotlib, providing a high-level interface for creating attractive and informative statistical graphics. It simplifies the process of generating complex visualizations by offering intuitive functions for exploring relationships in datasets and showcasing patterns and trends effectively.

Purpose of Using Seaborn

1. **Enhanced Data Visualization:**
    - **Purpose**: Create more attractive and informative visualizations.
    - **Detail**: Seaborn provides beautiful default styles and color palettes that make plots more visually appealing and informative.

2. **Statistical Plots:**
    - **Purpose**: Generate complex statistical graphics easily.
    - **Detail:** Seaborn simplifies the creation of advanced statistical plots such as heatmaps, violin plots, and pair plots, which can provide deeper insights into chat data.

3. **Integration with Pandas:**
    - **Purpose**: Work seamlessly with Pandas DataFrames.

- **Detail:** Seaborn is designed to work well with Pandas, making it easy to create plots directly from DataFrames and handle data efficiently.

4. **Customization*:**
   - **Purpose**: Customize and refine plots.
   - **Detail**: Seaborn allows extensive customization options to tweak the visual aspects of plots, helping to highlight key points and make the visualizations more effective.

5. **Visualizing Relationships:**
   - **Purpose**: Explore and visualize relationships between variables.
   - **Detail:** Seaborn's functions like sns.pairplot and sns.heatmap are great for visualizing the relationships between different variables in the chat data, such as message length and frequency over time.

Chapter-3
DESIGN

SOFTWARE REQUIREMENT SPECIFICATION

Software requirement specification (SRS) is a technical specification of requirements for the software product. SRS represents an overview of products, features and summaries the processing environments for development operation and maintenance of the product.

Requirement Specification-

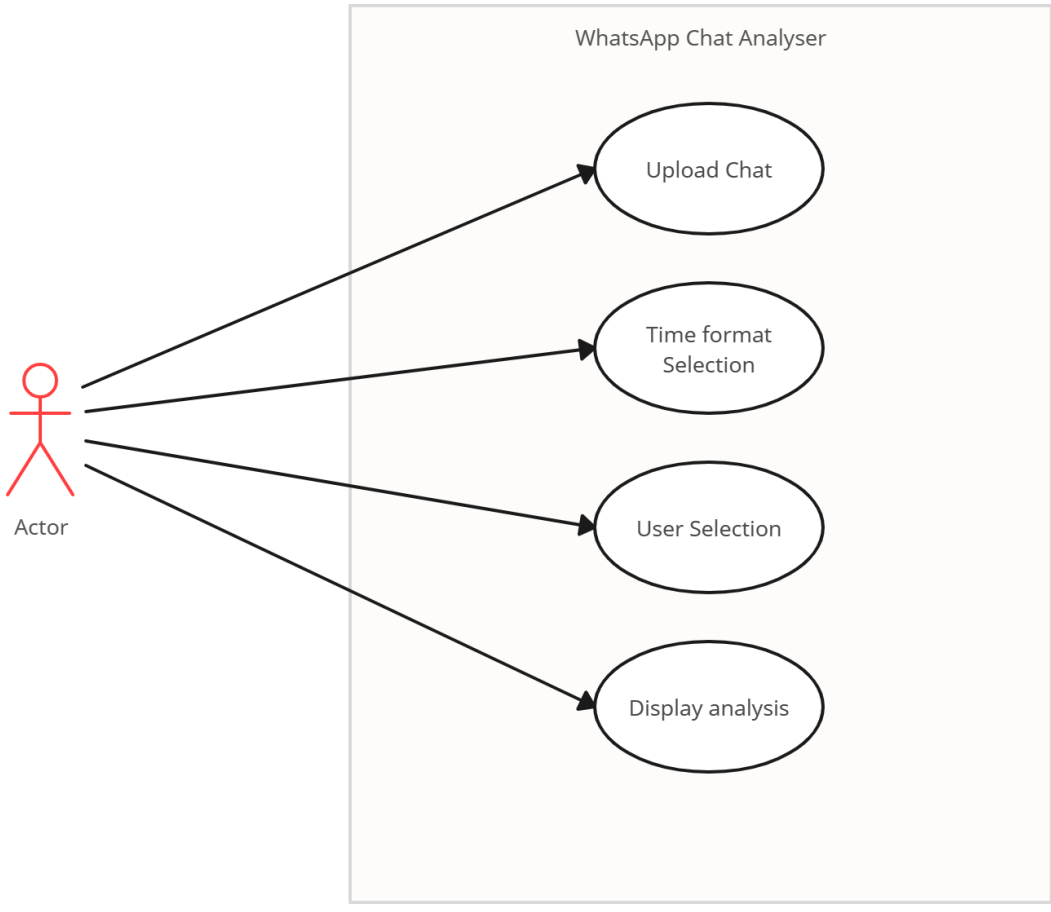
Conceptually every SRS should have the components

- Functionality
- Performance
- Design constraints imposed on
- Implementation External interfaces

USE CASE MODEL

- In the same case diagram, the actor in User.
- Users can make use of chat upload use cases to give input to the system.
- Select time format use case describes that user can input the time format of the file in the system
- Select user use case is to select whose analysis result is desired.

- User can make use of Show analysis the result of the entire analyses done by the system.



Use Case Model

Description

The class diagram has following two classes with their respective attributes and method.

1. DataFrame

- **Attributes:** User, message, date, time, year, month, day, dayname, dayofweek, weekname, hour, minute.

- **Methods:** separateDateTime

2. Generate report

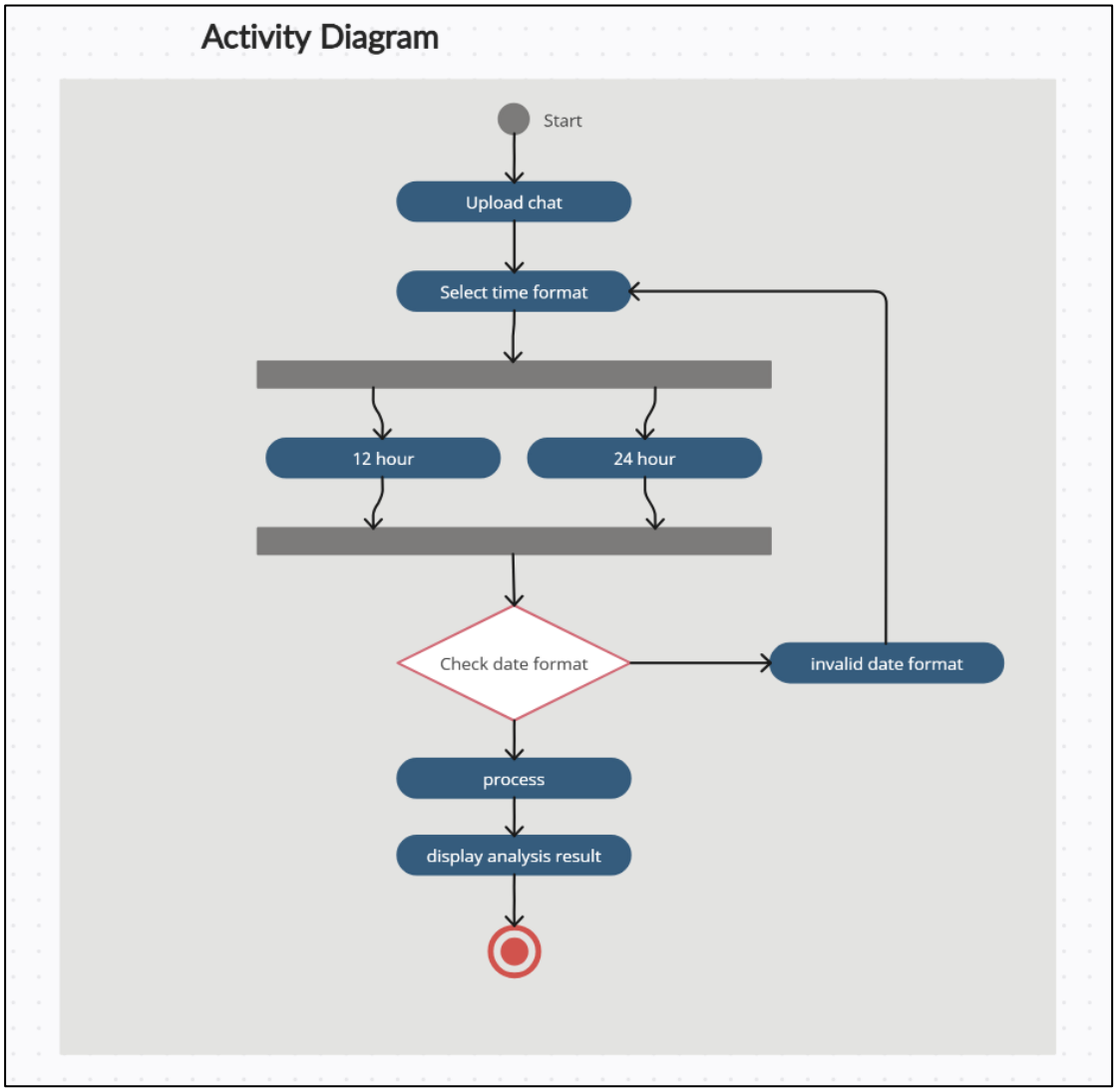
- **Attributes:** selectedUser, message, dataFrame, timeFormat

- **Methods:** Fetch_stats, most_talkative, hourly_timeline, daily_timeline, weekly_timeline. Most_busy_day, most_busy_month, most_common_words, user_chat_percentage, create worldcloud, show error

The class Dataframe is creating the class Generate report so Dataframe class includes Generate report class.

ACTIVITY DIAGRAM

- In the activity diagram as the initial activity starts user will upload the file as input which is action and in the next action time format will be selected.
- The decision box check chat format represents the validity of the time format of the file.
- If the time format is correct then analysis will be done and process will end.
- If the time format is wrong user will have to again check for the correct format.

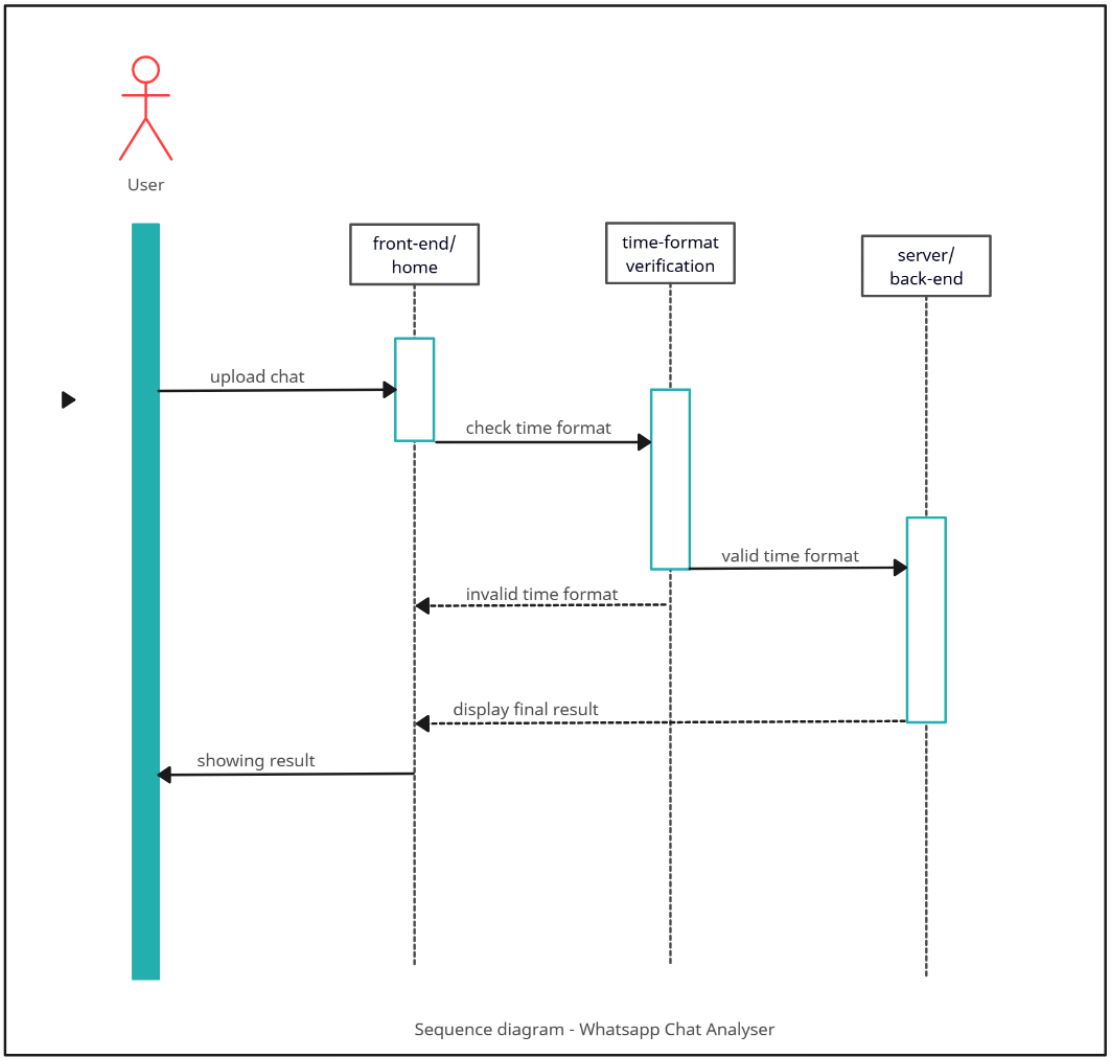


Chapter-4

SYSTEM MODELING

SEQUENCE DIAGRAM

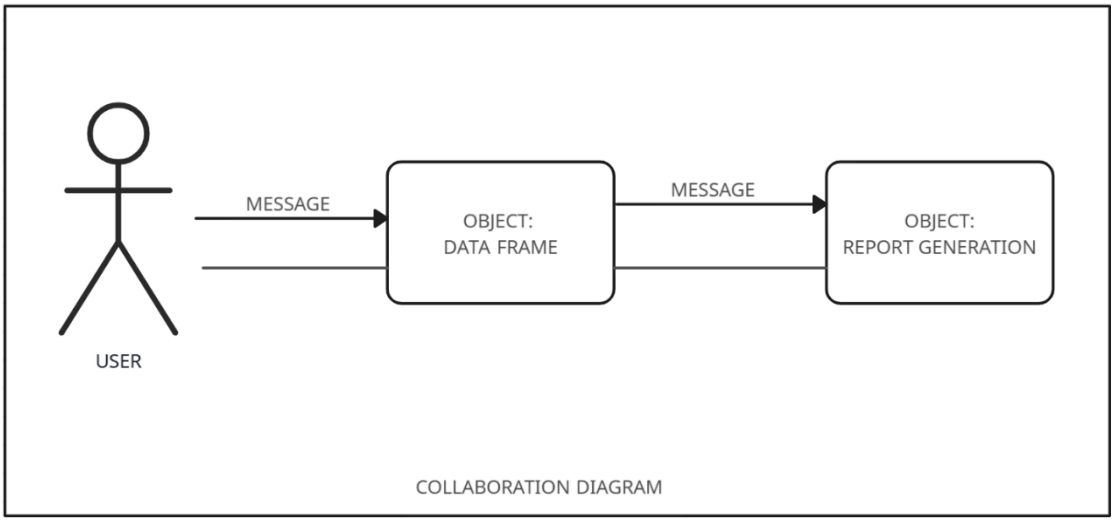
- The Sequence diagram start with upload chat in front-end then check time format will be ex it will match time format of chat upload with time format user selected than it goes to server than server perform analysis operation and send back to result in user end
- If time format of chat and user select time format sot match it will display an invalid time format select error.



Sequence diagram - Whatsapp Chat Analyser

COLLABORATION DIAGRAM

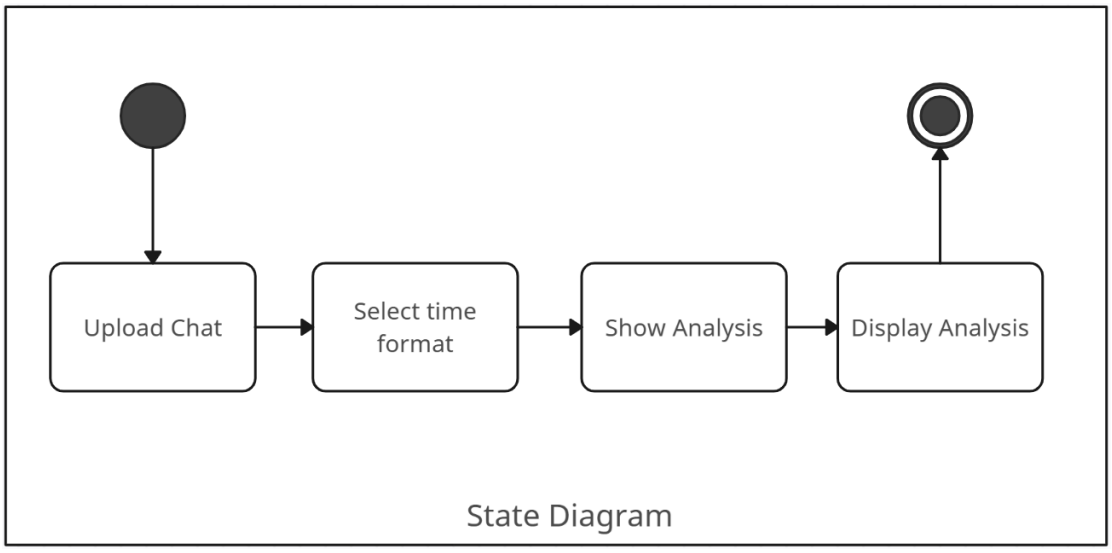
- This collaboration diagram shows the relationship between the objects in a system.
- An object consists of several features.
- Multiple objects present in the system are connected to each other



CONCEPTUAL LEVEL STATE DIAGRAM

- The state diagram starts with the uploading of the file and after that in the next state time format will be selected if the time format in valid them in the next analysis will be done. The analysis state will complete when the overall result will be shows on the user interface.

- In the analysis state the user can select the option of whose analysis he/she wants to see and this will give corresponding next state of display result.



State Diagram

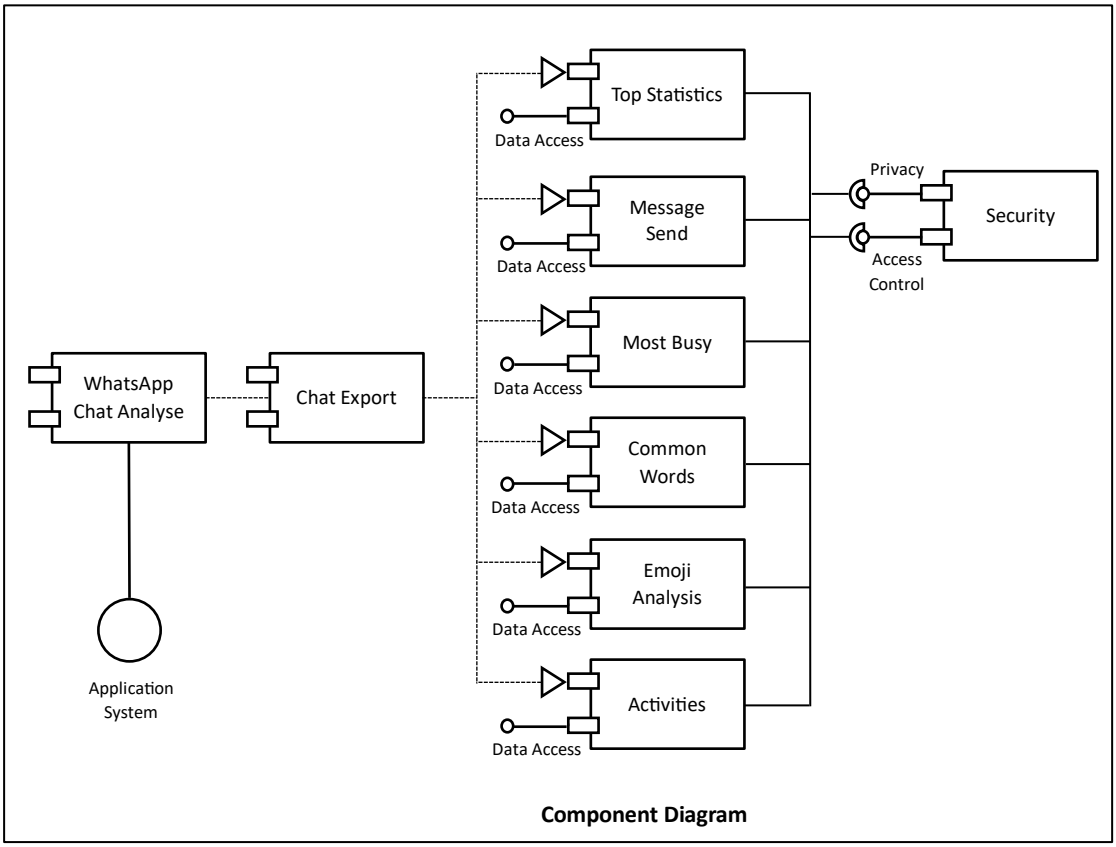
CONCEPTUAL LEVEL COMPONENT DIAGRAM

WhatsApp chat Analyzer has following components:

Chat export which connected with other components via input file.

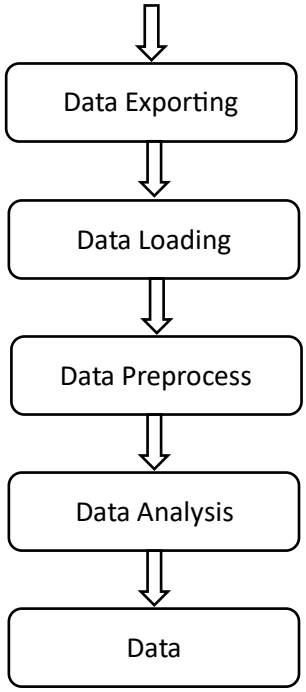
The data of the input file will be accessed by following components

- Total message
- Message sent
- Most busy
- Most common words
- Emoji analysis
- Weekly activity
- Monthly activity



Component Diagram

WORKING:


Data Exporting
Data Loading
Data Preprocess
Data Analysis
Data

Software requirement for developing application

- Jupyter notebook
- PyCharm Technologies
- Python and its libraries
- ML. algorithm

Jupyter notebook:

Introduction of Jupyter Notebook:

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It is widely used in data science, machine learning, and academic research due to its interactive and versatile nature.

Purpose of Using Jupyter Notebook

1. **Interactive Data Analysis:**
   - **Purpose**: Provide an interactive environment for data analysis.
   - **Detail**: Jupyter Notebooks allow you to write and execute code in cells, making it easy to test and iterate on data analysis steps interactively.

2. **Visualization Integration:**
   - **Purpose**: Embed visualizations directly in the workflow.
   - **Detail:** Jupyter supports inline plotting with libraries like Matplotlib and Seaborn, enabling you to visualize data and results within the notebook itself.

3. **Documentation and Code Together:**
   - **Purpose:** Combine code, text, and visuals.
   - **Detail**: Jupyter allows you to mix code, markdown, and visualizations, making it easy to document the analysis

process, write explanations, and present results all in one place.

4. **Ease of Use:**
   - **Purpose**: Simplify the coding and analysis process.
   - **Detail**: The interactive nature and user-friendly interface of Jupyter make it accessible for beginners and powerful for experts, facilitating easy data manipulation and exploration.

5. **Collaboration:**
   - **Purpose**: Facilitate sharing and collaboration.
   - **Detail:** Jupyter Notebooks can be easily shared with others, allowing collaborators to review, comment, and contribute to the analysis directly.

PyCharm

Introduction of PyCharm:

PyCharm is a powerful Integrated Development Environment (IDE) developed by JetBrains, designed specifically for Python development. It offers a range of tools and features that streamline the development process, making it an ideal choice for projects involving complex data analysis and visualization, such as the WhatsApp Chat Analyzer.

Purpose of Using PyCharm

1. **Integrated Development Environment (IDE):**
   - **Purpose:** Provide a comprehensive development environment.
   - **Detail**: PyCharm offers a robust IDE with features like code completion, syntax highlighting, and debugging tools, enhancing productivity and code quality.

2. **Project Management:**
   - **Purpose**: Manage complex projects efficiently.
   - **Detail:** PyCharm's project management features help organize and navigate large codebases, making it easier to manage files, dependencies, and settings.

3. **Debugging and Testing:**
   - **Purpose**: Facilitate efficient debugging and testing.

- **Detail:** PyCharm includes powerful debugging tools and test runners, which help in identifying and fixing bugs and ensuring code reliability through automated tests.

4. **Code Quality Tools:**
   - **Purpose:** Improve code quality and maintainability.
   - **Detail**: PyCharm integrates tools for code analysis, refactoring, and linting, helping maintain high standards of code quality and readability.

5. **Version Control Integration:**
   - **Purpose:** Simplify version control management.
   - **Detail**: PyCharm seamlessly integrates with version control systems like Git, making it easy to manage code versions, collaborate with others, and maintain a history of changes.

Python: -

Introduction to Python:

Python is a high-level, interpreted programming language known for its simplicity and readability. Created by Guido van Rossum and first released in 1991, Python emphasizes code readability with its notable use of significant whitespace. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Purpose of Using Python

1. **Ease of Use:**
    - **Purpose**: Make coding simple and easy to understand.
    - **Detail:** Python has a straightforward syntax that is easy to learn and read, which helps in quickly writing and understanding the project code.

2. **Extensive Libraries:**
    - **Purpose:** Utilize powerful tools and libraries.
    - **Detail**: Python has many libraries like Pandas for data manipulation, NumPy for numerical operations, and Matplotlib for creating charts, which are essential for analyzing and visualizing chat data.

3. **Data Handling:**
    - **Purpose:** Efficiently manage and process data.
    - **Detail:** Python's libraries provide robust support for handling large datasets, making it easier to load, clean, and analyze chat logs.

4. **Community Support:**
    - **Purpose:** Access help and resources.
    - **Detail**: Python has a large community, so finding tutorials, documentation, and support is easy, helping to solve problems and learn new techniques.

5. **Rapid Development:**
    - **Purpose:** Speed up the development process.
    - **Detail:** Python's simplicity and the availability of ready-to-use libraries allow for faster development and iteration of the project.

Regression

Introduction to Regression:

Regression analysis is a statistical technique used to model the relationship between a dependent variable and one or more independent variables. It aims to predict the value of the dependent variable based on the values of the independent variables. Regression analysis is widely used in various fields, including economics, finance, medicine, and machine learning, to understand and predict the behaviour of variables.

Purpose of Using Regression

1. **Identify Relationships:**
    - **Purpose**: Determine the relationship between variables.
    - **Detail:** Regression analysis helps to understand how one variable (e.g., time) affects another variable (e.g., message frequency), revealing trends and patterns in the chat data.

2. **Predictive Analysis:**
    - **Purpose**: Make predictions based on historical data.
    - **Detail**: Regression models can predict future chat activity based on past trends, such as predicting peak messaging times.

3. **Trend Analysis:**
    - **Purpose:** Analyze trends over time.
    - **Detail**: Regression helps to identify and quantify trends in chat activity over time, such as increasing or decreasing messaging frequency.

4. **Quantify Impact:**
    - **Purpose:** Measure the impact of different factors.
    - **Detail:** Regression can quantify how different factors, such as time of day or day of the week, impact the volume of messages sent.

5. **Anomaly Detection:**
    - **Purpose**: Detect outliers and unusual patterns.
    - **Detail**: By modeling expected behavior, regression can help identify outliers or anomalies in the chat data that deviate from the norm.

TESTING

Testing is the major quality control that can be used during software development. Its basic function is to detect the errors in the software. During requirement analysis and design, the output is a document that is usually textual and non-executable. After the coding phare, a computer program is available that can be executed for testing purposes.

Testing Objectives

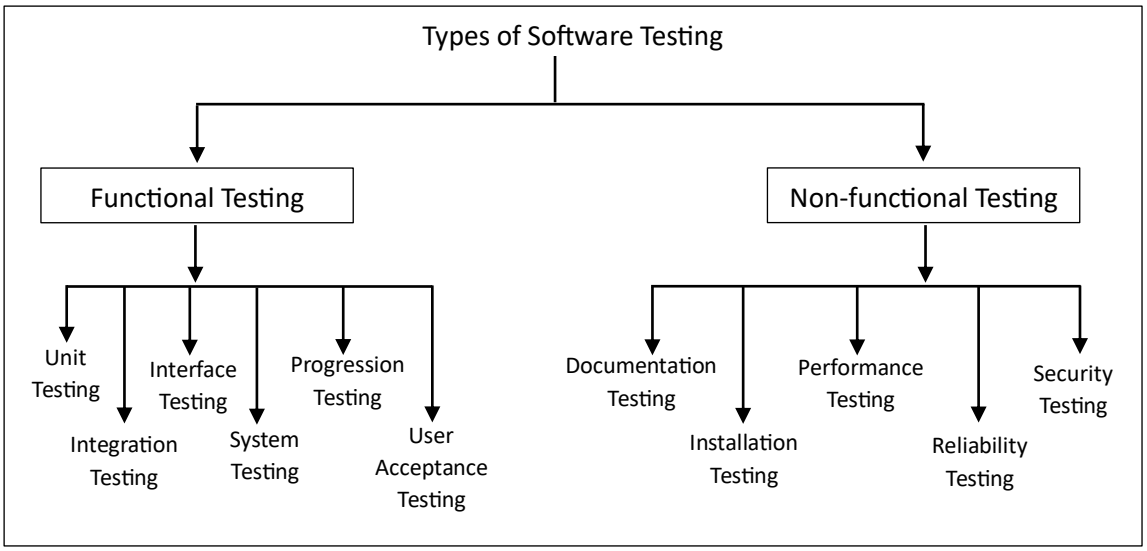
- To check if the application is working as expected.
- To check the error of different scenarios by using different test cases.

Testing Methods & Strategies used along with Test Data

Software Testing Strategies

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrast to the actual requirements. It can be either done manually or using automated tools.
In simple terms, Software Testing means Verification of Application under Test (AUT)

- Functional Testing
- Non-Functional Testing.



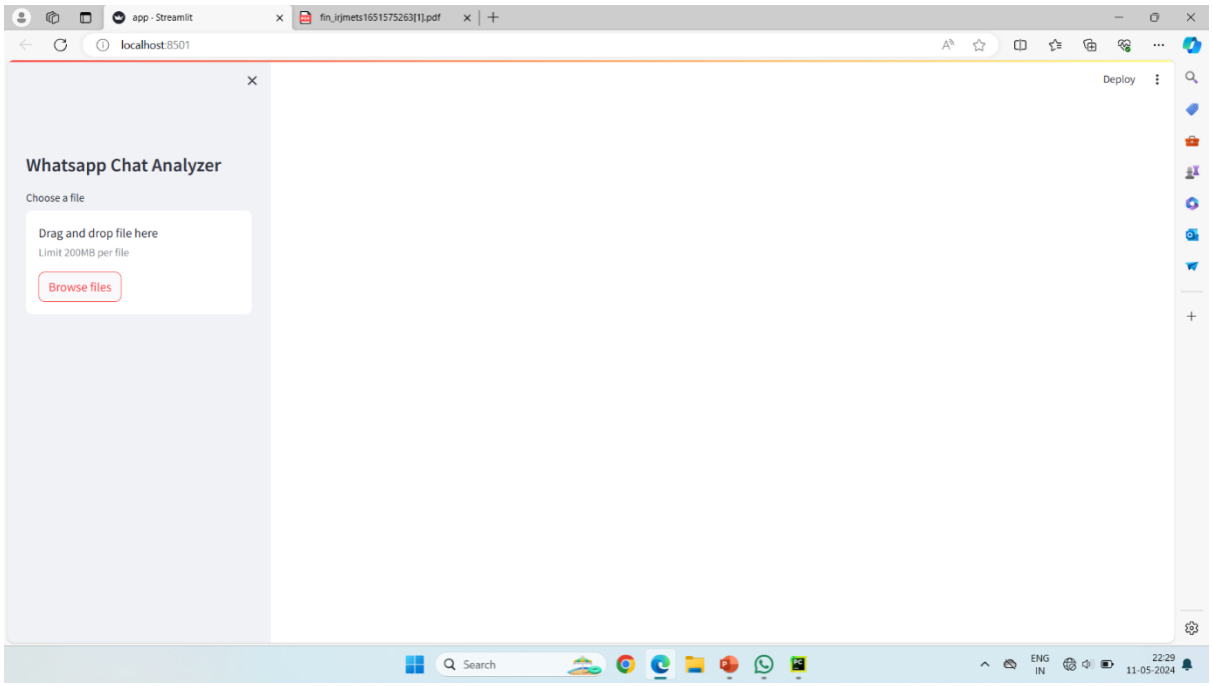
TEST CASES

S. No.	Test Case	Domain
1	Application load time is about 10 to 15 s	Performance Testing
2	Application should run in any browser	Compatibility Testing
3	Minimum Storage in browser 30MB	Scalability Testing

OUTPUT:

Pc or laptop:

INDEX PAGE

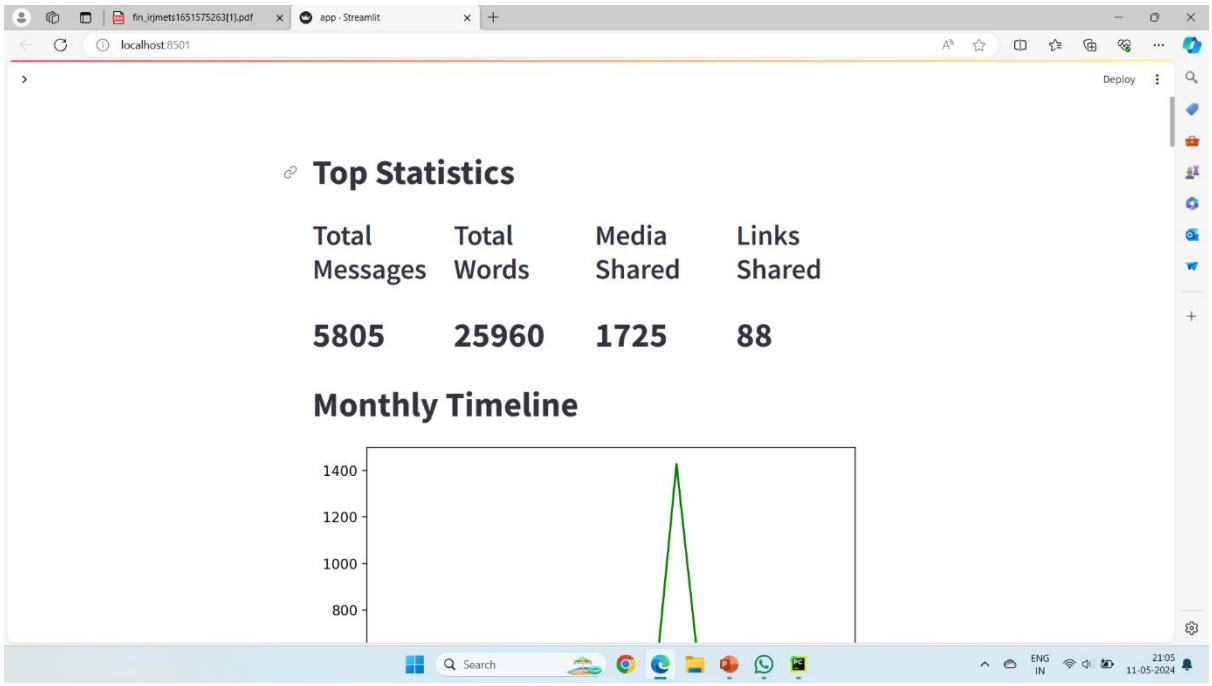


This is the opening page of the project.

1. **Title:** The project is titled “WhatsApp Chat Analyzer”, indicating that it’s designed to analyse WhatsApp chat data.
2. **File Upload Interface:** The interface has an area where users can either drag and drop files or click on the “Browse files” button to upload their WhatsApp chat files for analysis. There’s a limit of 100MB per file for the upload.

This interface provides a user-friendly way for users to upload their WhatsApp chat data for analysis.

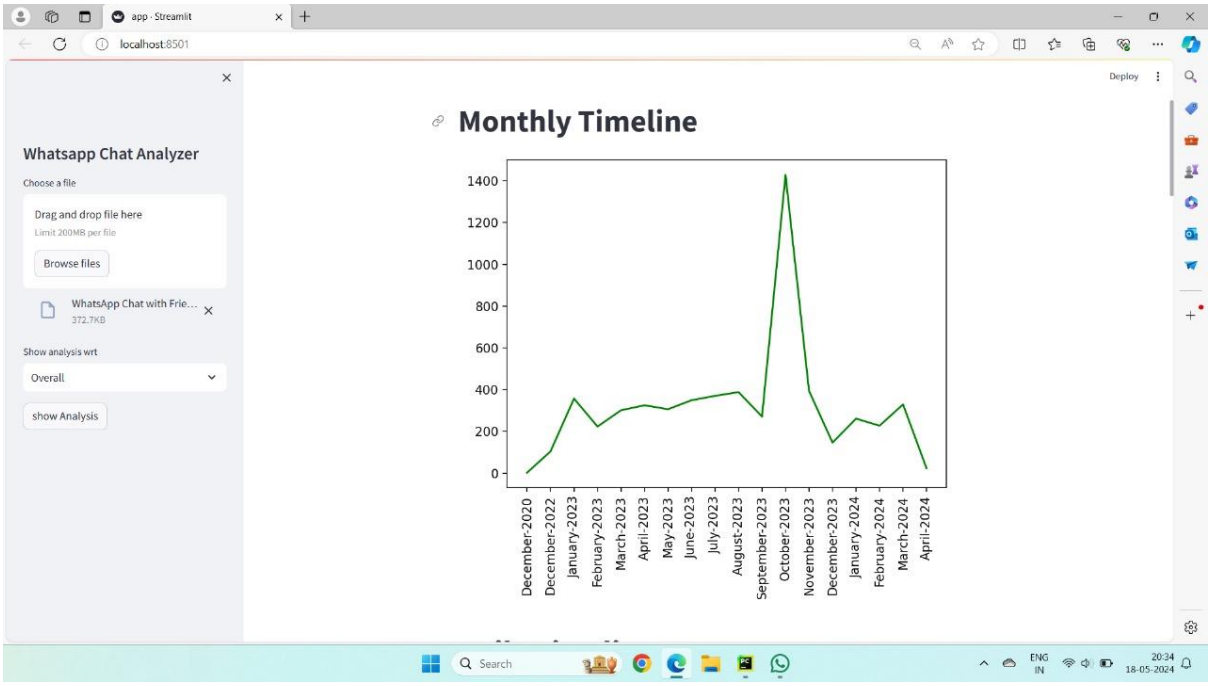
DATA VISUALISATION



"Top Statistics" shows the statistics like total messages, words, images links shared.

1. **Conversion to Data Frame:** The entire chat file is converted into a structured format known as a data frame. A data frame is a two-dimensional labelled data structure with columns potentially of different types. It's similar to a spreadsheet or SQL table, or a dictionary of Series objects.
2. **Separation of Words and Messages:** Once the chat data is in a data frame, the words and messages are separated. This could mean that each message is broken down into individual words, and each word is treated as a separate data point.

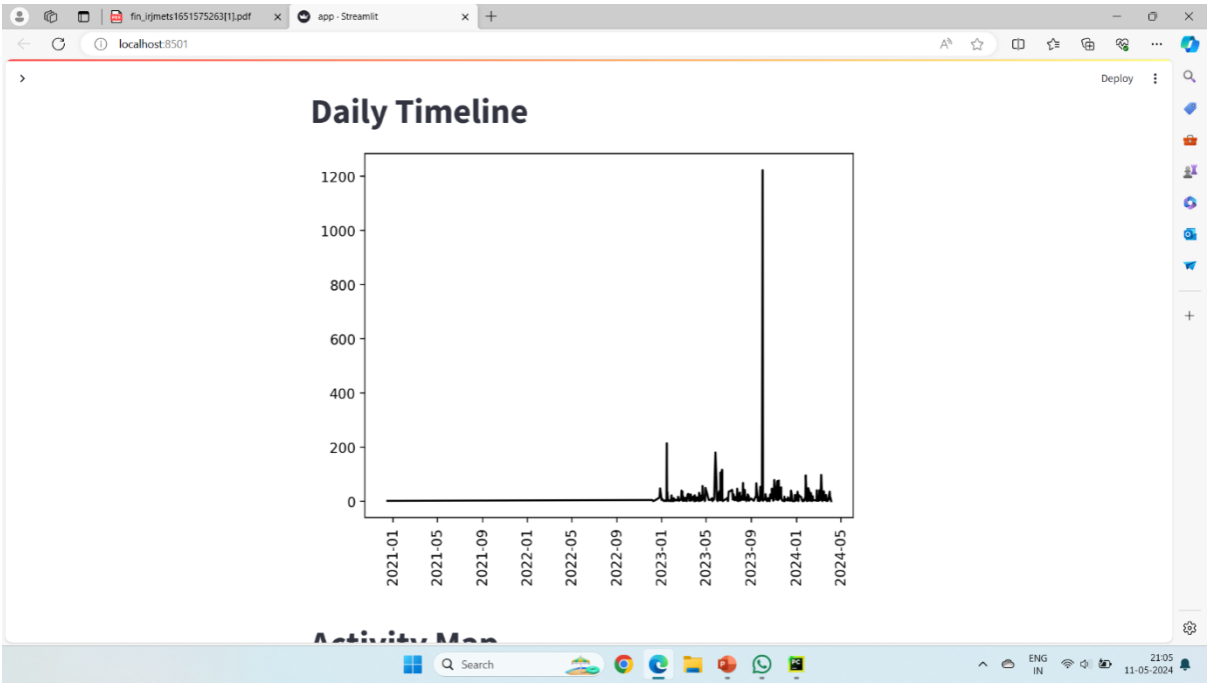
MONTHLY TIMELINE



"Monthly Timeline" gives the frequency of messages in a month.

1. **Graph Title:** The title of the graph appears to have typographical errors and is not clear. It seems to be intended to say "Monthly Timeline" or similar.
2. **Y-Axis (Message Count):** The Y-axis is labelled with numbers ranging from 0 to 1400, indicating the count of messages. This represents the frequency of messages in each month.
3. **X-Axis (Time Period):** The X-axis is labelled with months from September 2020 to April 2021. This represents the time period over which the message frequency is plotted.
4. **Data Representation:** A green line represents the frequency of messages each month. There is a noticeable peak in December 2020 where the message count exceeds 1200. This suggests a significant increase in message activity during that month.

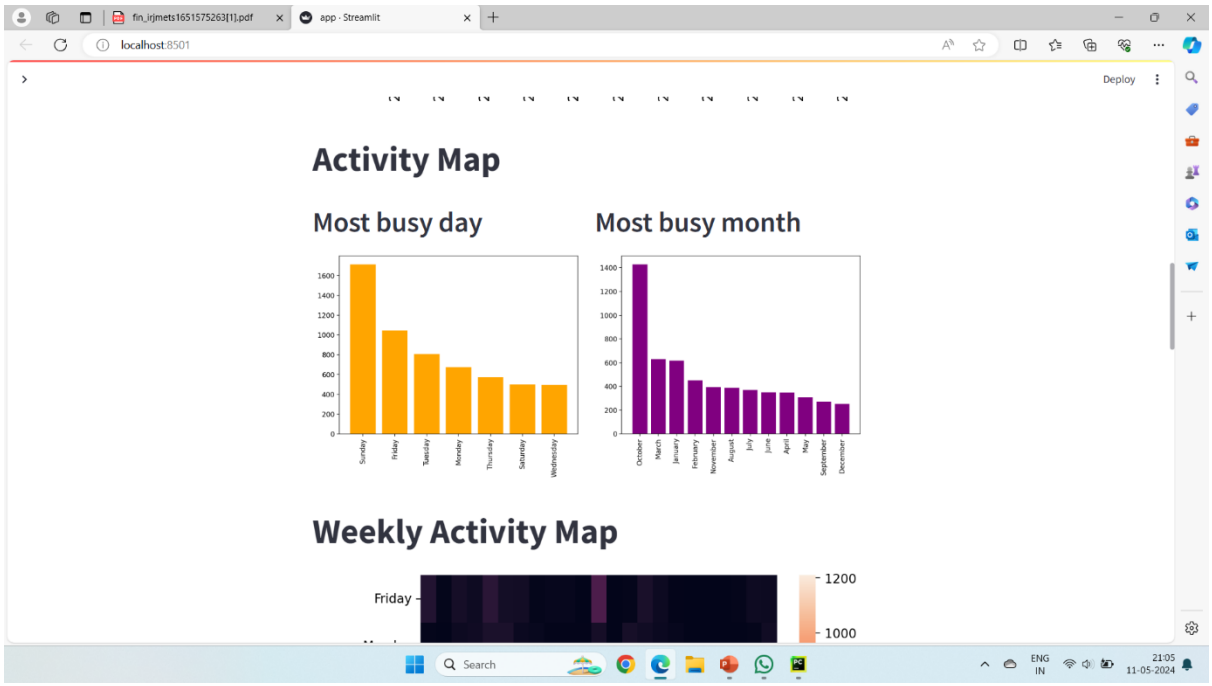
DAILY TIMELINE



"Daily Timeline" represents the frequency of messages per day over a specific period.

1. **Graph Title:** The graph is titled "Daily Timeline", indicating that it shows data over a series of days.
2. **Y-Axis (Message Count):** The Y-axis is labeled with numbers ranging from 0 to 1200, indicating the count of messages. This represents the frequency of messages each day.
3. **X-Axis (Time Period):** The X-axis represents dates, although they are not entirely clear due to resolution limitations. This represents the time period over which the message frequency is plotted.

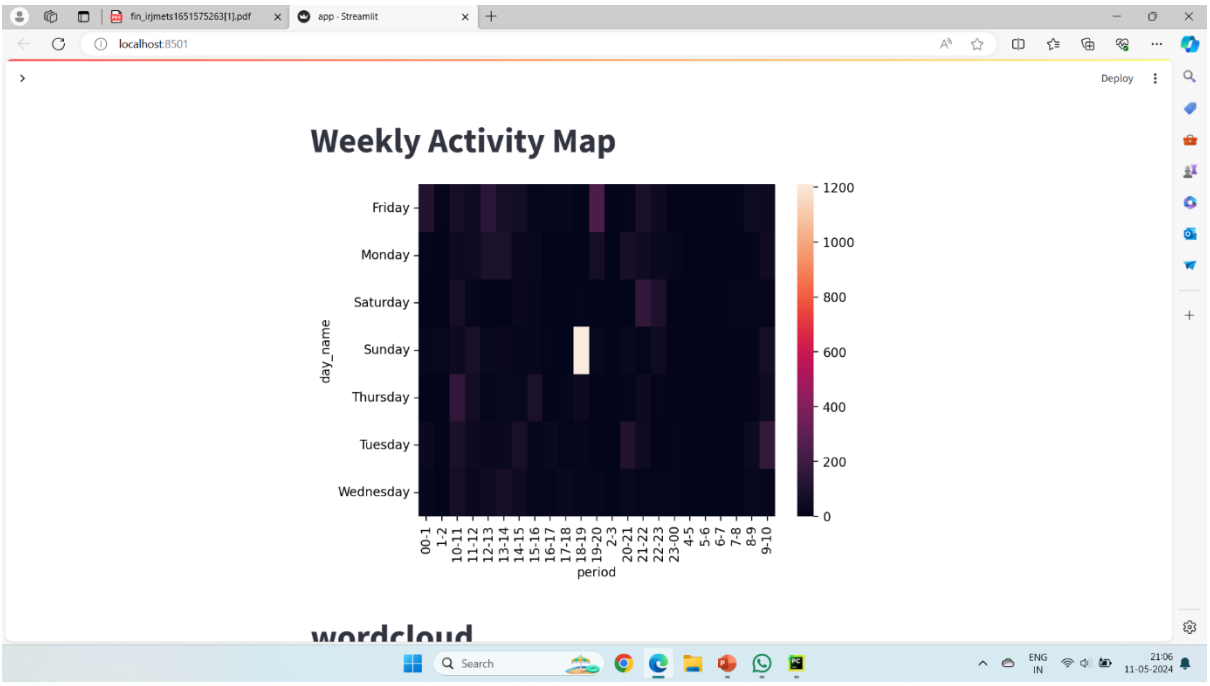
ACTIVITY MAP



"Activity Map" represents the frequency of messages on different days and months.

1. **Graphs:** There are two bar graphs at the top; one labeled "Most busy day" and another labeled "Most busy month".These graphs represent the volume of messages, indicating busier days and months.
    a. The "Most busy day" graph shows increasing message activity from left to right.
    b. The "Most busy month" graph shows one particular month having a significantly higher message count.
2. **Weekly Activity Map:** Below these bar graphs, there's a "Weekly Activity Map" which is a heatmap showing the frequency of messages throughout different hours of each weekday. It uses colour intensity to represent message volume, with darker colours indicating higher volumes.

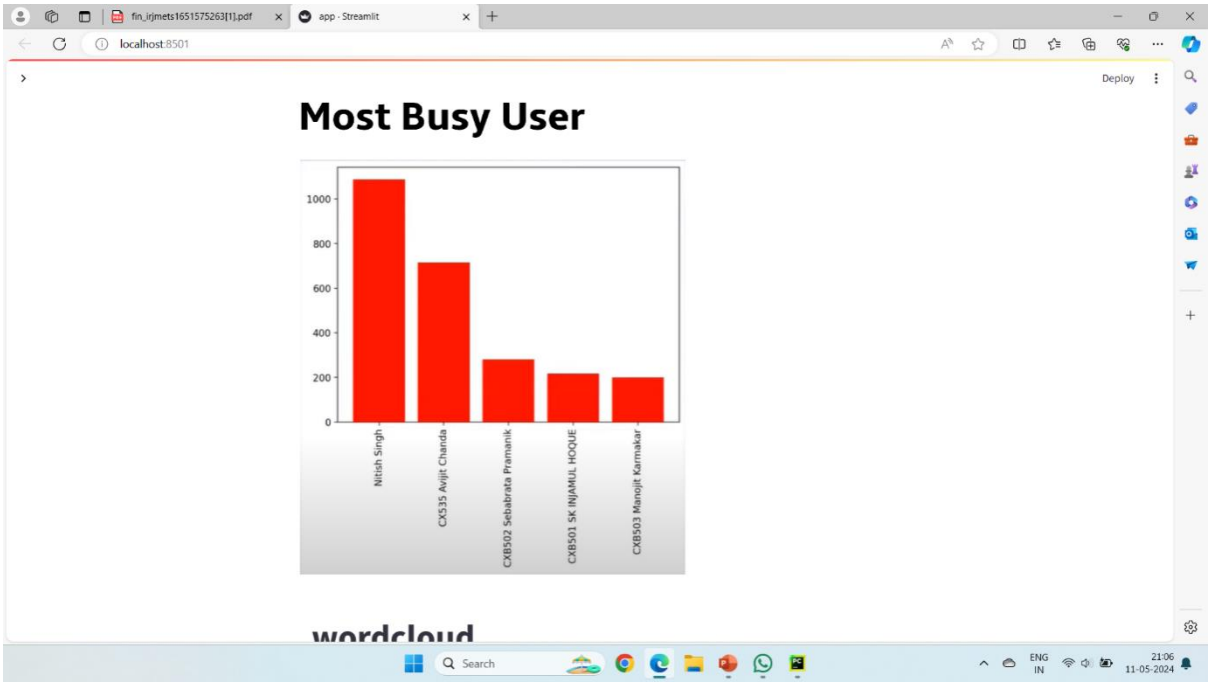
WEEKLY ACTIVITY MAP



"Weekly Activity Map" displays visual representation of activity over different days.

1. **Heatmap:** The main part of the image is a heatmap that represents data for different days of the week, from Friday to Wednesday, listed on the left side. Different shades of purple represent various data points on the heatmap; lighter shades indicate higher values.
2. **Colour Gradient Scale:** There is a colour gradient scale on the right-side indicating values from 0 to 1200. This scale helps interpret the colour intensity in the heatmap.

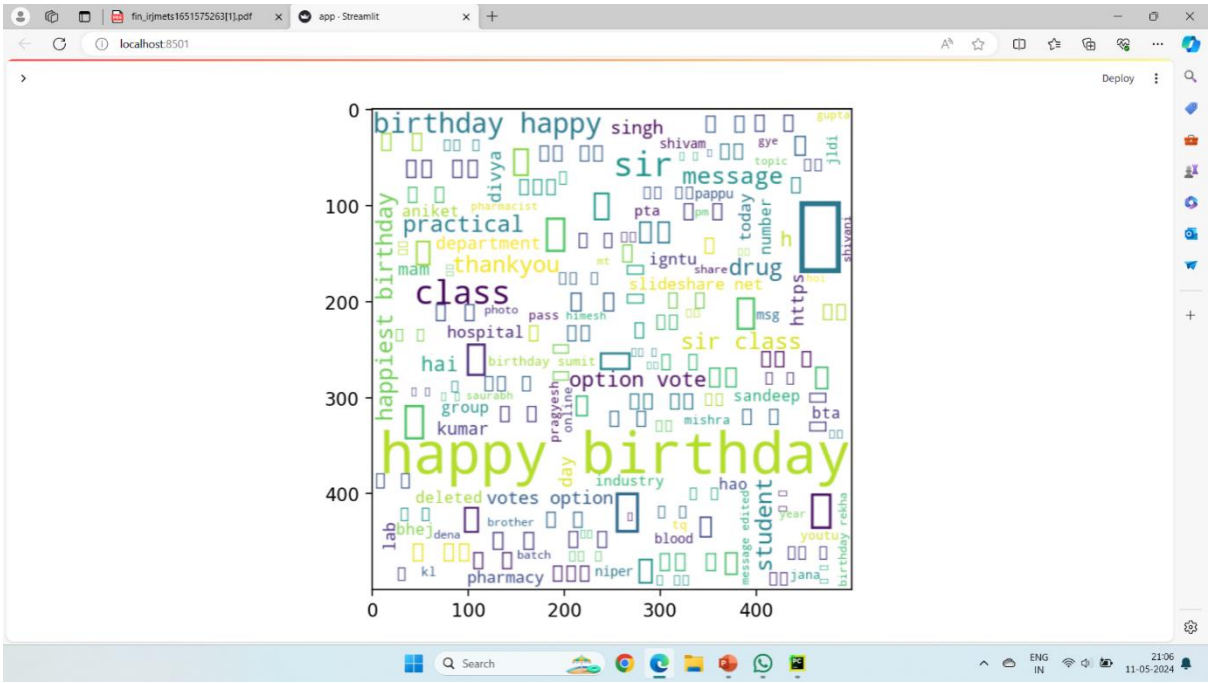
MOST BUSY USERS



"Most Busy User" displays the user who has most sent chats in the group.

1. **Y-Axis (Users):** The Y-axis lists different users in the group. The user "Nitish Singh" stands out with the highest bar, suggesting he is the busiest user in the group.
2. **X-Axis (Level of Activity):** The X-axis represents the level of activity of each user, indicated by the length of the blue bars. The length of the bar corresponding to "Nitish Singh" suggests a significantly higher level of activity compared to others.

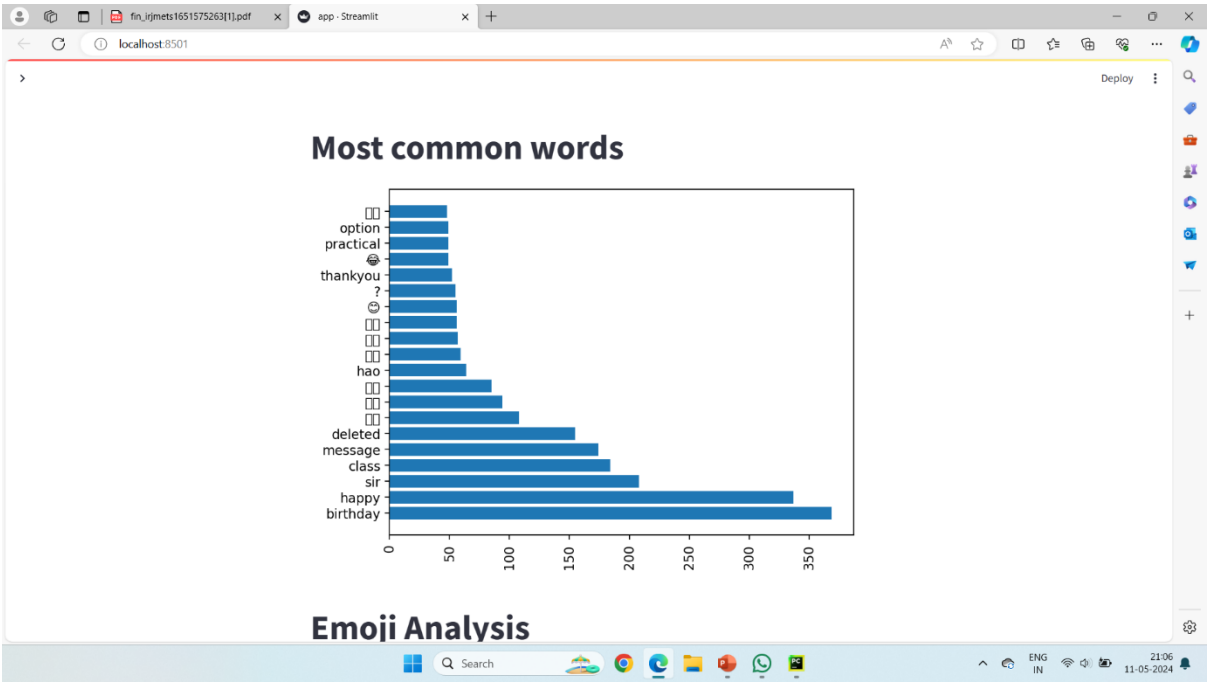
WORD CLOUD



"Word Cloud", which is a graphical representation of text data. In a word cloud, the size of each word indicates its frequency or importance within the dataset.

1. **Word Cloud:** The image shows a colorful word cloud with various words displayed in different sizes and orientations. The size of each word in the word cloud indicates its frequency or importance in the dataset.
2. **Prominent Words:** Words like "happy" and "birthday" are the most prominent, indicating their higher frequency or importance. Other visible words include "singh," "sir," "message," "practical," "department," "pta," "drug," "signature," "net," "class," "also," "pass," "hospital," "hai," "birthday," "cum," "sir," "option," "vote0," "group," "kumar," "happy," "birthday," "sandeep," "bta," "industry," "student," "deleted," "votes," "option," "brother," "pharmacy," and "niper."

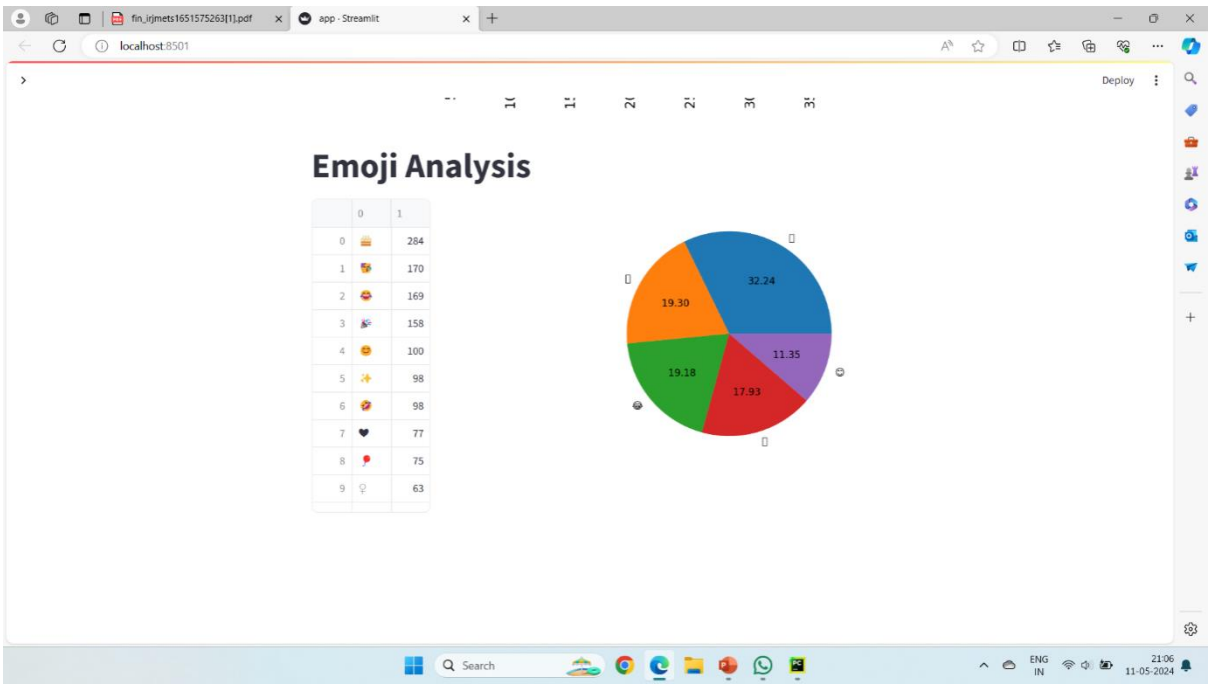
MOST COMMON WORDS



"Most common words" provides a visual representation of word frequency.

1. **Graph Title:** The graph is titled "Most common words", indicating that it shows the frequency of various words.
2. **Y-Axis (Words):** The Y-axis lists words like "birthday", "happy", "class", "sir", and others. These are the most commonly used words in the dataset.
3. **X-Axis (Frequency):** The X-axis represents the frequency of each word, indicated by the length of the blue bars. The word "birthday" has the longest bar, suggesting it's the most frequently used word.

EMOJI ANALYSIS



"Emoji Analysis" represents the frequency of various emojis used in a dataset. Here's a detailed description based on your detail and the image:

1. **Emoji List:** On the left side, there's a list enumerating eight different emojis along with their respective counts. These are the most commonly used emojis in the dataset.
2. **Pie Chart:** On the right side, there's a colourful pie chart illustrating the distribution of these emojis. Each slice of the pie chart is labelled with percentages, representing the proportion of each emoji in the dataset.
3. **Most Common Emojis:** The most common emojis according to the pie chart and the list are 😂, 👍, ♡, and 😭.

Chapter-5

FUTURE ENHANCEMENTS

1. **Advanced Sentiment Analysis:**

   - Objective: Improve the accuracy of sentiment analysis by incorporating more sophisticated natural language processing (NLP) techniques.
   - Enhancement: Use advanced NLP models like BERT or GPT for sentiment classification to better understand the context and nuances of messages, including slang and emojis.

2. **Topic Modeling:**

   - Objective: Identify and analyze topics of conversation within the chat data.
   - Enhancement: Implement topic Modeling techniques such as Latent Dirichlet Allocation (LDA) to discover underlying themes and topics in the chat history. This will help in understanding the main subjects discussed over time.

3. **Enhanced Visualization Options:**

   - Objective: Provide more interactive and detailed visualizations.

- Enhancement: Incorporate advanced visualization libraries such as Plotly or D3.js for creating interactive charts and dashboards. Adding features like zoom, hover-over details, and dynamic filtering can enhance the user experience.

4. Enhanced Privacy and Security Features:

- Objective: Ensure the privacy and security of chat data.
- Enhancement: Implement encryption for data storage and transmission. Add user authentication and access control mechanisms to ensure that only authorized users can access and analyze the chat data.

Chapter-6

LIMITATION OF PROJECT

The WhatsApp Chat Analyzer project has several limitations: it must handle sensitive personal data, posing privacy risks, and relies on the quality of exported chat data, which can be incomplete or corrupted. The analysis lacks contextual understanding, focusing only on textual data without capturing the nuances of conversations. Scalability is an issue with very large datasets, and the project does not support real-time data analysis. Results are specific to WhatsApp and may not generalize to other platforms. Regression models may oversimplify data relationships. While Streamlit offers a user-friendly interface, it lacks advanced customization. The project is dependent on the current export format of WhatsApp, and significant computational resources are required. Lastly, users need basic Python and data analysis knowledge to utilize the tool effectively.

1. Maximum file size to be uploaded in 200MB
2. Only Support English Language
3. Support only text extension

Chapter-7

CONCLUSION

The WhatsApp Chat Analyzer project successfully demonstrates how data science techniques can be applied to understand and gain insights from WhatsApp chat data. By systematically collecting, preprocessing, analyzing, and visualizing chat logs, the project provides valuable information about message frequency, user activity, content trends, and sentiment. The use of Python libraries such as Pandas, Matplotlib, Seaborn, and Streamlit facilitated the creation of an interactive and user-friendly web application.

The project achieved its primary goals of:

- Extracting and cleaning WhatsApp chat data.
- Conducting exploratory data analysis to uncover communication patterns and user behaviors.
- Visualizing the data through interactive dashboards and meaningful charts.
- Applying regression analysis to predict message characteristics based on various factors.

While the current implementation offers significant insights, there are opportunities for future enhancements, such as incorporating advanced sentiment analysis, topic Modeling, and additional machine learning techniques to further enrich the analysis. These improvements could provide deeper understanding and more precise predictions, making the tool even more powerful and versatile.

REFERENCE

My project Link:

https://github.com/shilpa0216/WhatsApp-chat-analyzer.git

https://github.com/arvindanalyst/WhatsApp-chat-Analyzer/upload

YouTube leacher: -

CampusX:

https://youtu.be/Q0QwvZKG_6Q?si=hfic8nFRZ5RQb0Bn

Machine Learning:

https://youtube.com/playlist?list=PLjVLYmrlmjGexLyoCdDrt8Nil1Alg_L3&si=hyow5VkTqwTh8GqE

PANDAS:

https://youtube.com/playlist?list=PLjVLYmrlmjGdEE2jFpL71LsVH5QjDP5s4&si=W1MOzSSlEmPdXWxo

NumPy:

https://youtube.com/playlist?list=PLjVLYmrlmjGfgBKkIFBkMNGG7qyRfo00W&si=YadISwgHi_v8hJ9U

Matplotlib:

https://youtube.com/playlist?list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&si=a2zdvu5I5oWw4IHj