

Whatsapp Chat Analyzer Using Stremlit

M. Jayababu ¹, K. Kiran Paul ², C. Sekhar ³, K. Karthikeya Achari ⁴

^{1, 2, 3, 4} Department of Artificial Intelligence, G. Pullaiah College of Engineering and Technology, Kurnool, India

Abstract:- Communication patterns have evolved significantly with the emergence of messaging applications, with WhatsApp standing as one of the most prevalent platforms in recent years. This research introduces a specialized tool designed to extract meaningful insights from WhatsApp conversation data. The developed application performs in-depth analysis of chat exchanges regardless of subject matter or conversation context. By leveraging fundamental Python libraries that include pandas for the data manipulation, the matplotlib as well as seaborn for the visualization, sentiment analysis techniques, our tool processes conversational datasets effectively. The implementation via Flutter application ensures optimal resource utilization while maintaining performance efficiency, allowing seamless handling of substantial data volumes. This approach creates new opportunities for understanding communication patterns in digital exchanges.

Keywords: Conversational data analysis, Python data processing, Visualization techniques, Sentiment evaluation, Mobile application implementation.

1. Introduction

The foundation of this research centers on data processing and analytical methodologies. When implementing machine learning (ML) systems, establishing appropriate learning parameters represents a crucial initial step toward model refinement. Data preprocessing constitutes a fundamental component in ML workflows. To enhance model effectiveness, substantial datasets are required, which directed our attention toward Facebook's messaging platform WhatsApp, representing one of today's largest data generation sources. According to official statistics, approximately 55 billion messages traverse WhatsApp daily. The average user dedicates 195 minutes weekly to WhatsApp activities while participating in multiple group conversations. Given this wealth of untapped information readily available, developing analytical frameworks to extract meaningful patterns from these digital exchanges becomes increasingly valuable.

1.1 Problem Statement

This project introduces a statistical analysis framework for the WhatsApp communication data. Operating on exportable chat files from the WhatsApp platform, the system generates diverse visualizations that reveal patterns such as interaction frequencies between participants. Our approach employs dataset manipulation techniques to provide enhanced understanding of WhatsApp conversations stored on users' devices. Furthermore, this study will help in understanding message patterns, time-based trends, and user engagement within groups. By utilizing machine learning algorithms, insights such as message sentiment and peak activity hours can be determined to improve user engagement analytics.

1.2 Existing System

Significant evolution has occurred in messaging platforms over time. Earlier iterations lacked capabilities such as status displays, document sharing, and location transmission functionalities. Contemporary versions incorporate these features comprehensively. Previous implementations restricted image sharing through document formats. The current ecosystem allows Windows access through the WhatsApp web application via QR code authentication. Additionally, the platform includes an "export chat" function enabling users to transmit conversation records for analytical purposes through various channels, including email and other messaging services. However, while these features enable communication, they do not provide deeper analytical insights into

user behavior, engagement frequency, or sentiment trends. Existing research in this area is limited, and there is a need for more structured analysis to explore WhatsApp's messaging impact.

1.3 Proposed System

Data Preprocessing

Initial project phase focuses on understanding and implementing various Python modules. This process illuminates the advantages of utilizing pre-built libraries rather than developing equivalent functionality from scratch. These modules enhance code representation and improve user comprehension. The implementation incorporates libraries that include pandas, sklearn, csv, numpy, matplotlib, emoji, scipy, nltk, sys, re, and seaborn. These libraries help in handling text-based data, extracting keywords, analyzing emojis, and performing sentiment analysis on WhatsApp messages.

Exploratory Data Analysis

The first analytical step applies sentiment analysis algorithms to identify positive, negative, and neutral communication elements, which form the basis for pie chart representations. Additional visualizations include:

- Line graphs displaying author-specific message counts by date
- Message frequency distribution by individual participant
- Chronological message distribution patterns
- Media sharing statistics by author
- Identification of messages without attributed authors
- Hourly message distribution patterns

By implementing these analysis techniques, we can uncover messaging behavior trends, detect frequently used words, and determine the impact of multimedia messages within group discussions.

1.4 Objective

Contemporary technological advancements increasingly depend on data resources. Obtaining relevant data requires targeted research aligned with specific tool requirements. As ML practitioners develop models addressing diverse challenges, the demand for large-scale, appropriate data continues to grow. This project aims to enhance understanding of various conversation patterns within digital communications. The resulting analysis provides valuable input for ML models exploring conversational data. These models require properly structured learning instances to achieve optimal accuracy. Our project delivers comprehensive exploratory data analysis across diverse WhatsApp conversation types, offering practical applications in behavioral analysis, sentiment prediction, and engagement tracking.

2. Literature Survey

Usage and Impact Analysis of WhatsApp Messenger

Several investigations have examined WhatsApp usage patterns and their societal effects. Some research focuses on student impacts, while others target general populations in specific regions.

A study conducted in southern India focused on individuals between 18-23 years to investigate WhatsApp's significance among younger users. Findings revealed that students allocated approximately 8 hours daily to WhatsApp usage while remaining online for nearly 16 hours. All participants confirmed using WhatsApp for friend communications, exchanging multimedia content including images, audio, and video files. The research established WhatsApp as the predominant application during smartphone engagement among youth. Methodologies included analyzing usage intensity, identifying popular features, and determining positive and negative impact levels.

Content Analysis of WhatsApp Conversations

An analytical examination evaluated WhatsApp's effectiveness in Karachi, providing important research on the application's emergence as a major mobile messaging platform in the Pakistan. With digital technology advancement along with mobile phone proliferation in Pakistan, communication paradigms have transformed completely. The growing prevalence of smartphones and social networking applications has accelerated communication processes unprecedented in history. Economic constraints notwithstanding, mobile phone ownership for maintaining connections with family, friends, and customers has become widespread. This evolution has driven increased implementation of quantitative and qualitative research methodologies.

WhatsApp Group Data Analysis with R

Dataset examined comprised one year of group chat records (May 2015-May 2016) containing 5,563 entries with various measurable characteristics including usage duration, response patterns, message types (emoji, text, multimedia), demographic activity levels, and temporal distribution patterns.

Forensic Analysis of WhatsApp Messenger

WhatsApp supports diverse communication modalities, including direct user exchanges, broadcast messages, along with group conversations. During interactions, users exchange text messages, multimedia files (that is images, audio, video), contact information, geographical coordinates. Every user maintains a profile containing name, status information, and avatar imagery.

3. Software Requirement Analysis

3.1. Feasibility Study

Primary objective of the feasibility assessment has been evaluating technical, operational, along with economic viability of application development. Feasibility determines project worthiness through structured evaluation processes.

3.1.1 Technical Feasibility

This measures specific technical solutions and resource/expertise availability. The proposed system utilizes Jupyter software, created by a non-profit organization creating open-source tools and standards for interactive computing throughout several programming languages. The implementation focuses on data processing code using Python to extract meaningful insights from WhatsApp group conversations. The system requires computational resources with at least 8GB RAM and a processing speed of 2.5 GHz or above for efficient execution of data analysis operations.

3.1.2 Operational Feasibility

Operational feasibility addresses system utilization following development, potential user resistance, and impact on anticipated benefits. The system provides multiple benefits, displaying WhatsApp user statistics and communication patterns through pie charts and bar graph visualizations. It ensures that users with minimal technical expertise can access and interpret the results. The system will be tested in various WhatsApp groups to confirm adaptability across different conversation formats.

3.1.3 Economic Feasibility

It represents the most commonly applied methodology for estimating system effectiveness, measuring cost-effectiveness of information system solutions. Economic analysis assesses proposed system's effectiveness with cost-benefit analysis as its central component. This particular project carries limited economic considerations as it primarily facilitates data exchange between devices. The software is built on open-source technologies, reducing development and maintenance costs. The only costs associated may be related to cloud storage for large datasets or additional computational resources for large-scale analysis.

By providing structured analysis techniques, this project ensures a cost-effective solution for exploring WhatsApp communication patterns while leveraging modern data analysis methods.

4. System Implementation

Python

Originally released in the year 1991, this high-level, interpreted general-purpose programming language was developed by Guido Van Rossum. Its structural elements as well as object-oriented methodology support programmers in developing clear, logical code for applications at various scales. Python applications include software development, workflow automation, database connectivity, web development (server-side), mathematical operations, file manipulation, big data processing, complex calculations, rapid prototyping, and production software development.

Robust frameworks that include Django and Flask for web development, TensorFlow and PyTorch for ML, NumPy and Pandas for data analysis, a wealth of scientific computing libraries are all part of Python's ecosystem. The language emphasizes readability with significant whitespace and a comprehensive standard library, following a "batteries included" philosophy. Python 3.x introduced improved Unicode support, function annotations, and asynchronous programming capabilities. The Python Package Index (PyPI), that hosts more than 300,000 packages, enabling developers to leverage community-developed solutions for specialized tasks. Python's interpreter implementation varies between CPython (reference implementation), PyPy (JIT compilation for performance), Jython (Java integration), and IronPython (.NET framework integration).

JavaScript Object Notation (JSON)

JSON is an open standard file format along with a data transfer protocol that stores and transmits array data structures and attribute-value pairs utilizing human-readable text. This widely implemented data format serves numerous applications, including XML replacement in AJAX systems. JSON maintains language independence despite JavaScript origins, with most modern programming languages supporting JSON generation and parsing. The official Internet media type is application/json with .json file extensions. Browser-server data exchanges require text formatting, which JSON provides. JavaScript objects convert to JSON for server transmission, and received JSON transforms into JavaScript objects, eliminating complex parsing operations.

JSON's lightweight structure makes it particularly suitable for RESTful API communications, configuration files, and application state management. The format supports 6 data types: objects, strings, arrays, booleans, numbers, and null values. Unlike XML, JSON excludes comments and schema definitions to maintain simplicity. JSON Schema provides validation capabilities for ensuring structural integrity of JSON documents. Advanced implementations include JSONP (JSON with Padding) for cross-domain requests, JSON-LD for linked data representation, and GeoJSON for encoding geographic data structures. Modern frameworks often include built-in serialization/deserialization utilities for converting between native objects and JSON representations with minimal developer intervention.

DART

A client-optimized programming language supporting multi-platform application development. Developed by the Google, it powers desktop, server, mobile, along with several web applications. Dart implements object-oriented, class-based architecture with garbage collection and C-style syntax. Compilation targets include native code or else JavaScript. The language supports interfaces, abstract classes, refined generics, mixins, type inference. Web browser compatibility depends on source-to-source JavaScript compilation. According to official documentation, Dart had been "designed to be easy to write development tools for, well-suited to modern app development, and capable of high-performance implementations." Browser execution involves JavaScript precompilation using the dart2js compiler, ensuring compatibility across major browsers without browser-specific adaptation. Through optimized JavaScript output that eliminates expensive operations, Dart code sometimes outperforms equivalent hand-written JavaScript implementations. Dart's sound type system helps catch errors during development rather than runtime, improving application stability. The language features an ahead-of-time (AOT) compiler for producing efficient native code and a just-in-time (JIT) compiler enabling hot reload during development. Dart's asynchronous programming model uses Future and Stream classes with async/await syntax for managing concurrent operations without callback complexity. The Dart package ecosystem is managed through pub.dev,

providing reusable libraries for common development needs. Version 2.0 introduced strong mode typing, while subsequent releases enhanced null safety, extension methods, and spread operators. Dart's standard library includes collections, async utilities, IO operations, and math functions, reducing dependency on external packages for core functionality.

Flutter

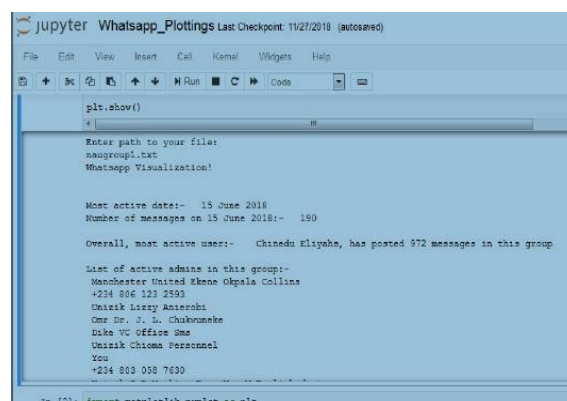
An application SDK for developing high-fidelity as well as high-performance applications for web (beta), iOS, Android, desktop from unified codebases. This Framework aims to allow natural-feeling applications across different platforms while respecting platform-specific behaviors including scrolling mechanics, typography, iconography, and interface elements. Flutter enhances developer productivity through single-codebase development for iOS and Android. Its modern, expressive language and declarative approach reduce code volume even for single-platform development. The framework facilitates prototyping and iteration through code changes with real-time reloading during execution, maintains debugging continuity after crashes, and enables highly customized user experiences through comprehensive widget libraries incorporating Material Design and Cupertino components. These capabilities support beautiful, brand-aligned implementations without conventional widget set limitations.

Flutter's architecture employs a layered design including the framework layer (widget hierarchy, rendering, animation), engine layer (low-level implementation in C/C++), and platform-specific embedder layer. The rendering system uses Skia, a 2D graphics engine, to create consistent visuals across platforms. Flutter's widget system embraces composition over inheritance, with everything from buttons to padding implemented as widgets. State management approaches range from `setState` for simple cases to Provider, Bloc, Redux, and MobX patterns for complex applications. Flutter DevTools provides performance profiling, widget inspection, and memory analysis. The widget tree rebuilds efficiently through diff algorithms that minimize repainting. Flutter supports internationalization, accessibility features, and responsive layouts through MediaQuery and LayoutBuilder widgets. The ecosystem includes Firebase integration, camera access, geolocation, and numerous third-party packages expanding framework capabilities.

5. Result Analysis

The outcomes of the work displayed a number of activities on particular days as determined by the system at the designated time. According to the findings, June 15, in the year 2018, was the most active date. On that most active date, 190 messages were sent. Additionally, a record of the most active user overall revealed that the person had posted more than 972 messages to the group. Emojis and the list of active authors were also recorded by the system. Additionally, it was displayed that there were 230 users in that group overall. Additionally, a complete list of every user on the platform was generated, complete with their name or phone number and the number of times they had posted. Additionally, "the" was listed as the most frequently utilized word, appearing 43313 times. The figure below displays a snapshot of the screen presenting the analysis output.

The output of the Python analysis performed on the specified group conversation is presented in the subsequent.



```
jupyter Whatsapp_Plotting Last Checkpoint: 11/27/2018 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
+ + + + + Run C + Code
plt.show()
Enter path to your file:
whatsapp.txt
Whatsapp Visualization!

Most active date:- 15 June 2018
Number of messages on 15 June 2018:- 190

Overall, most active user:- Chinedu Eliyahu, has posted 972 messages in this group

List of active users in this group:-
Manchester United Ebene Obwala Collins
+234 806 123 2593
Umasik Lizzy Anserohi
Oma Dr. J. L. Chabumeke
Dike VC Office Sme
Umasik Chimes Personnel
You
+234 803 050 7630

In [2]: import matplotlib.pyplot as plt
```

Fig 5.1: Sample output of the WhatsApp plot.

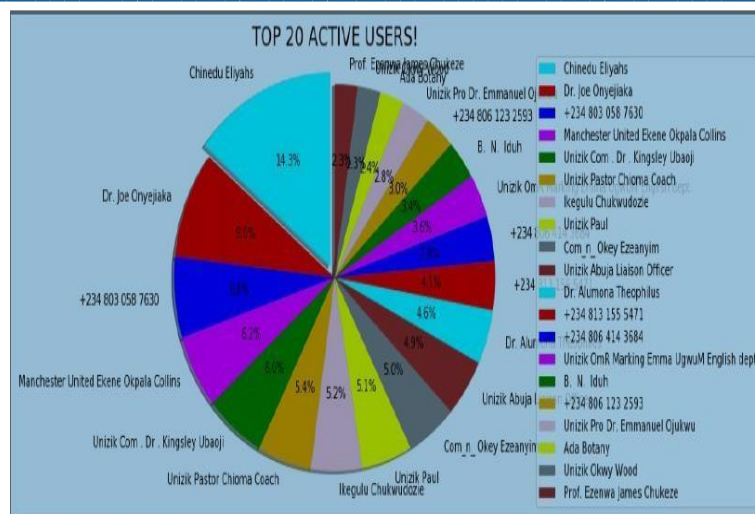


Fig 5.2: Top 20 active users

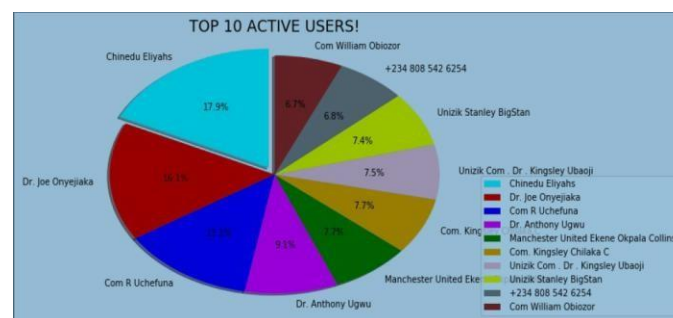


Fig 5.3: Top 10 active users

6. Conclusion

The capabilities of WhatsApp combined with Python's analytical power demonstrate significant potential for network data examination. This research explored WhatsApp application features and supporting libraries to analyze group conversations and visually represent participation patterns among members. Implementation included pseudocode development followed by visualization techniques. The analysis specifically identified the top 10 and top 20 contributors based on activity levels. The implementation utilized Python libraries that include NumPy for the numerical operations, Pandas for the data manipulation, Matplotlib and Seaborn for the visualization. Completed system successfully revealed participation patterns among group members according to expected outcomes. Importantly, the framework demonstrates versatility for analyzing any WhatsApp group data provided as input.

References

- [1] Ahmed, B. (n.d.). *ARE MENA STOCK MARKETS PREDICTABLE ?*
- [2] Al-Khazali, O. M., Ding, D. K., & Pyun, C. S. (2007). A new variance ratio test of random walk in emerging markets: A revisit. *Financial Review*, 42(2), 303–317. <https://doi.org/10.1111/j.1540-6288.2007.00173.x>
- [3] Andr w W. Lo A.Graig Mackinlay. (1988). *Stock market prices do not follow a random walk*. University of Pennsylvania.
- [4] Bouzia, A. (2020). *Les d terminants de la liquidit  du march  des actions au Maroc Determinants of the equity market ' s liquidity in Morocco. 1*, 89–110.
- [5] Bruno H . Solnik. (1973). American Finance Association Note on the Validity of the Random Walk for European Stock Prices. *The Journal of Finance* , Vol . 28 , No . 5 (Dec ., 1973), Pp . 1151-1159 Published by : Wiley for the American Finance, 28(5), 1151–1159.

-
- [6] Case, K. E., & Shiller, R. J. (1989). The efficiency of the market for single-family homes. *American Economic Review*, 79(1), 125–137. <https://doi.org/10.2307/1804778>
- [7] Chow, K. V., & Denning, K. C. (1993). A simple multiple variance ratio test. *Journal of Econometrics*, 58(3), 385–401. [https://doi.org/10.1016/0304-4076\(93\)90051-6](https://doi.org/10.1016/0304-4076(93)90051-6)
- [8] Cootner, P. H. (1964). Cootner-1964.Pdf. In *Industrial Management Review* (pp. 231–252).
- [9] Cowles, A. (1933). Can Stock Market Forecasters Forecast? *Econometrica*, 1(3), 309. <https://doi.org/10.2307/1907042>
- [10] Elhami, M., & Hefnaoui, A. (2018). L'Efficiency du Marché dans les Marchés Émergents et Frontières de la Zone MENA. *Finance and Finance Internationale*, 10, 1–18. <https://doi.org/10.12816/0045344>
- [11] Fama, E. F. (1965). Random Walks in Stock Market Prices. *Financial Analysts Journal*, 51(1), 75–80. <https://doi.org/10.2469/faj.v51.n1.1861>
- [12] Fama, E. F. (1970). Session Topic: Stock Market Price Behavior Session Chairman: Burton G. Malkiel Efficient Capital Markets: A Review Of Theory And Empirical Work. *The Journal of Finance*, 25(2), 383–417.
- [13] FONTAINE, P. (1973). Peut-on prédire l'évolution des marchés d'actions à partir des cours et des dividendes passés ? (tests de marche au hasard et de co-intégration). *Journal de La Société Statistique de Paris, Tome 131, No 1 (1990), p. 16-36, 114, 96–106.*
- [14] French, K. R., & Roll, R. (1986). Stock return variances. *Journal of Financial Economics*, 17(1), 5–26. [https://doi.org/10.1016/0304-405x\(86\)90004-8](https://doi.org/10.1016/0304-405x(86)90004-8)
- [15] Harrison, B., & Moore, W. (2012). Stock Market Efficiency, Non-Linearity, Thin Trading and Asymmetric Information in MENA Stock Markets. *Economic Issues*, 17(1), 77–93.
- [16] Harvey, C. R. (1995). The risk exposure of emerging equity markets. *World Bank Economic Review*, 9(1), 19–50. <https://doi.org/10.1093/wber/9.1.19>
- [17] Jensen, M. C. (1978). Market Efficiency Some Anomalous Evidence Regarding Market Efficiency I believe there is no other proposition in economics which has more solid empirical evidence supporting it than the Efficient Market Hypothesis . That hypothesis has been tested and , w. *Journal of Financial Economics*, 6(July 2002), 95–101.
- [18] Kendall, M. G., & Hill, A. B. (1953). The analysis of economic time-series-part i: Prices. *Journal of the Royal Statistical Society. Series A (General)*, 116(1), 11–34.
- [19] Kupukile, M. B. E. and F. G. and, & Mlambo. (2011). [WIP] Mpr a. *Economic Policy*, 2116, 0–33.
- [20] Lo, A. W., & MacKinlay, A. C. (2014). 2. Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test. *A Non-Random Walk Down Wall Street*, 1(1), 17–46. <https://doi.org/10.1515/9781400829095.17>
- [21] Malkiel, B. G. (2003). *Critics*. 17(1), 59–82.
- [22] Mlambo, Chipu and Biekpe, N. (2010). *Munich Personal RePEc Archive The efficient market hypothesis : Evidence from ten African stock markets The efficient market hypothesis : Evidence from ten African stock markets. 25968.*
- [23] Omran, M., & Farrar, S. V. (2006). Tests of weak form efficiency in the Middle East emerging markets. *Studies in Economics and Finance*, 23(1), 13–26. <https://doi.org/10.1108/10867370610661927>
- [24] Régis Bourbonnais, M. T. (n.d.). *Analyse des séries temporelles.*
- [25] Vuillemeys, G. (2013). *Sur le statut épistémologique de l ' hypothèse d ' efficience des marchés. 14.*