

Sentiment analysis of ChatGPT Android App

Objective:

To analyze user reviews and ratings of the ChatGPT Android App using data analysis techniques and basic machine learning models. The project aims to clean and preprocess the data, normalize textual content, and extract insights about user engagement, sentiment patterns, temporal trends, and behavioral characteristics. This analysis provides a structured understanding of user feedback to support data-driven evaluations and improvements.

Dataset Description:

The dataset utilized for this project was sourced from Kaggle, contributed by Ashish Kumar. It contains a collection of daily-updated user reviews and ratings for the ChatGPT Android App, along with metadata such as review content, user scores, timestamps, app versions, and relevancy information. The dataset provides a rich foundation for exploring user feedback and conducting sentiment and trend analyses. Access to the dataset is available through the Kaggle link: [ChatGPT Reviews Dataset](#).

And the dataset consisted of the following columns:

reviewID: Unique identifier for each review.

userName: Name or alias of the user providing the review.

content: Text of the user's review.

score: Rating given by the user (1-5).

thumbsUpCount: Number of upvotes received for the review.

reviewCreatedVersion: App version at the time of review.

at: Timestamp indicating the date and time of the review.

appVersion: Detailed app version information.

This data was collected and updated dynamically to capture user feedback.

Tools and Technologies Used:

Languages and Libraries:

Python (pandas, numpy, matplotlib, seaborn, re, nltk, BeautifulSoup, TextBlob).

Frameworks: NLP processing for sentiment analysis.

Visualization Tools: Matplotlib and Seaborn for plotting graphs and showing trends.

Team Structure and Roles:

1. Dineshbabu

A. **Tasks:** Data Quality, User Engagement, and Temporal Trends Analysis.

B. **Contributions:**

- Handled missing values and removed duplicate reviews to ensure data quality.
- Explored score distributions and analyzed review patterns by app version.
- Examined trends across dates and correlated them with app updates.

2. Omkar Dodamani

A. **Tasks:** Text Normalization, Sentiment Analysis, and User Behavior Analysis.

B. **Contributions:**

- Addressed punctuation and special characters in review content.
- Performed polarity-based sentiment analysis and visualized changes over time.
- Analyzed reviewer types and their scoring patterns.

Methodology :

1. Data Preprocessing and Cleaning:

- Addressed missing values in columns like `userName`, `content`, and `reviewCreatedVersion`.
- Removed duplicate reviews to enhance dataset quality.
- Normalized text in the `content` column by removing special characters, HTML tags, and redundant spaces.

2. Feature Engineering:

- Derived new temporal features from the `at` column, such as year, month, and weekday.
- Extracted major app versions from the `appVersion` column.

3. Sentiment Analysis:

- Used polarity scores to classify reviews into positive, negative, or neutral categories.
- Analyzed the frequency of key terms for each sentiment type using text tokenization and filtering.

4. User Behavior Analysis:

- Differentiated between first-time reviewers and repeat reviewers based on `reviewID` and `userName`.
- Studied consistency in ratings for frequent reviewers.

5. Trend Analysis:

- Mapped temporal patterns in review frequency and sentiment shifts using the at column.
- Evaluated the impact of app updates on user ratings and thumbs-up counts.

Key Insights and Results:

- **Rating Distribution:** Most users rated the app positively, but a subset of reviews indicated dissatisfaction, often correlating with specific app versions.
- **Common Words:** Positive reviews frequently highlighted "usability" and "accuracy," while negative reviews cited "errors" and "lags."
- **Temporal Patterns:** Review frequency spiked post-major app updates, with sentiments fluctuating based on feature changes.
- **User Behavior:** Repeat reviewers provided more critical feedback, while first-time reviewers were generally more lenient.

Challenges and Limitations:

1. Data Quality and Missing Values:

Addressed incomplete values in critical columns like at and appVersion.

2. Preprocessing Complexities:

Handling sarcasm and mixed sentiments in text reviews required additional manual adjustments.

3. Scalability Issues:

Efficiently processing large datasets without compromising on speed and accuracy.

Conclusion and Future Scope:

- This project successfully identified critical user sentiments and behavioral trends, providing actionable insights for improving the ChatGPT Android App.
- In future sentiment detection accuracy can be enhanced by incorporating advanced **NLP models like BERT(bidirectional Encoder representations from Transformers)**.
- Additional data like user demographics could provide deeper insights into user experiences.

Acknowledgment

We express our gratitude to Kaggle and [Ashish Kumar](#) for providing the comprehensive dataset that formed the foundation of this analysis. Special thanks to the collaborative efforts of team members: [Omkar Dodamani](#), who contributed to text normalization, sentiment analysis, and user behavior studies, and [Dineshbabu](#), who handled data cleaning, user engagement analysis, and temporal trends exploration. Their combined expertise and dedication were instrumental in overcoming challenges and successfully completing this project.

References

Python Documentation: <https://www.python.org/doc/>

pandas: <https://pandas.pydata.org/>

TextBlob: <https://textblob.readthedocs.io/>

NLTK: <https://www.nltk.org/>

BeautifulSoup: <https://www.crummy.com/software/BeautifulSoup/>