JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

SECTOR-62, NOIDA

WORK SUMMARY REPORT

SUBMITTED TO-

MEMBERS-

VANSHIKA JALHOTRA – 21103119

VIDHI RASTOGI – 21103121

HIMRAL GARG – 21103128

MOTIVATION

Understanding Communication Patterns: Studying chat data helps reveal several communication patterns among individuals or groups. This can be beneficial for linguistics, social sciences, or even marketing strategies to understand how people interact and convey information.

Improving User Experience: Companies often use chat analysers to enhance user experience in customer support. By analysing chats, they can identify common issues, improve response times, and train support agents for better service.

Sentiment Analysis: Analysing chats can reveal sentiment trends, helping businesses gauge customer satisfaction or detect potential issues early on. It assists in understanding how users feel about products, services, or experiences.

Training AI Models: Chat analysers can be used to train and fine-tune AI models, especially in natural language processing (NLP). They help machines understand context, tone, and intent in human conversations.

Security and Compliance: Analysing chats can aid in monitoring for compliance purposes, ensuring that conversations adhere to legal and ethical guidelines. It can also be used in security contexts to detect anomalies or potential threats in communication. Personal Development: On an individual level, analysing one's own chat data can offer insights into personal communication habits, helping individuals improve their communication skills or better understand their social interactions. The motivation behind a specific chat analyser project can vary based on these or other objectives. Whether it's for research, business optimization, personal growth, or any other purpose, the aim is typically to derive meaningful insights from conversational data.

TYPE OF PROJECT-

Research cum development project- The Chat Analyzer project combines research into conversation dynamics with software development. It studies language patterns, sentiment analysis, and communication nuances, using these findings to design and build an effective chat analysis tool. The process involves iterative refinement, where research informs development and vice versa, leading to continuous enhancement of the analyser's accuracy and usability through testing and improvement cycles.

FEATURES BUILT

1. Sentiment Analysis: Assessing emotions expressed in conversations.
2. Keyword Extraction: Pinpointing key terms or topics within discussions.
3. Conversation Flow Analysis: Understanding the progression and coherence of dialogue.
4. User Profiling: Creating profiles based on communication habits.
5. Summarization: Condensing lengthy conversations into brief overviews.

6. Anomaly Detection: Noticing unusual patterns or behaviours in chats.
7. Real-time Monitoring: Analysing ongoing chats in live environments.
8. Customization: Allowing users to tailor analysis based on specific criteria or goals. These features collectively aim to enhance understanding, improve communication, and derive valuable insights from conversations across various platforms and contexts.

CRITICAL ANALYSIS OF RESEARCH PAPER

1. https://www.analyticsvidhya.com/blog/2022/01/sentiment-analysis-with-lstm/

The article on Analytics Vidhya introduces sentiment analysis using Long Short-Term Memory (LSTM) neural networks. It explains LSTM's role in capturing sequential information from text data for sentiment classification. The tutorial guides readers through preprocessing steps, LSTM model building, and sentiment prediction, demonstrating LSTM's effectiveness in analyzing sentiment in textual data.

2. https://www.researchgate.net/publication/358942212_SENTIMENT_ANALYSIS_AND_CLASSIFICATION_OF_ASUU_WHATSAPP_GROUP_POST_USING_DATA_MINING

The research paper employs data mining for sentiment analysis on an ASUU WhatsApp group. It lacks detailed methodology, statistical analysis, or validation metrics, raising concerns about robustness. Despite its attempt to explore sentiment, the absence of rigorous methodology and analysis weakens its credibility and generalizability.

3. https://www.researchgate.net/publication/370147223_A_Survey_of_Various_Sentiment_Analysis_Techniques_of_Whatsapp?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19

The ResearchGate survey explores diverse sentiment analysis techniques for WhatsApp. It examines methods like machine learning, lexicon-based, and hybrid approaches. While comprehensive, it lacks depth in individual technique analysis, needing more comparative evaluations to establish their effectiveness within WhatsApp's specific context.

THE PROGRAMMING LANGUAGES USED

Python

LIBRARIES-

streamlit

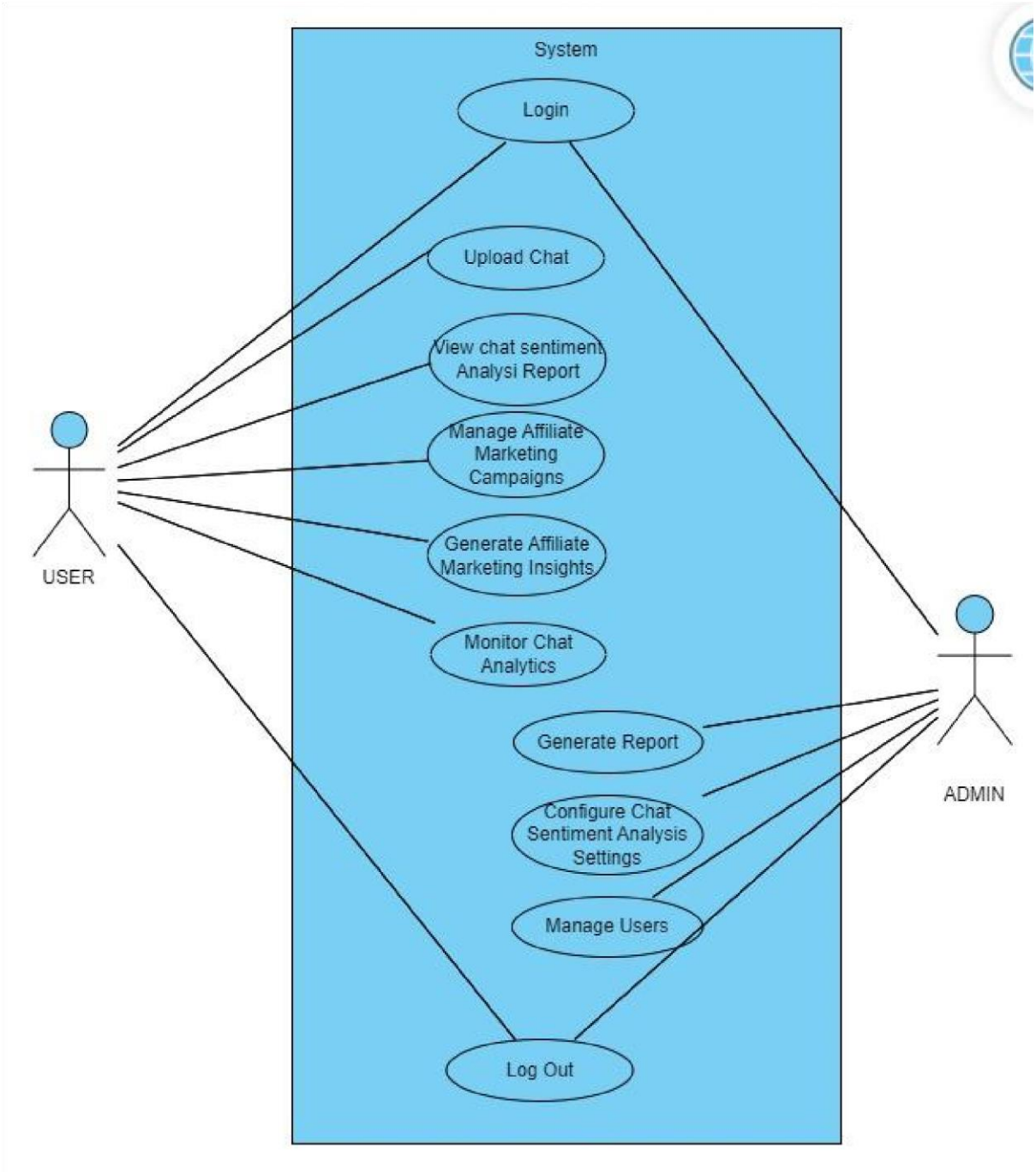
preprocessor

helper

matplotlib

seaborn

USE CASE DIAGRAM



PROPOSED METHODOLOGY

1. **File Upload:**
   - Users are prompted to upload a file containing WhatsApp chat data.
   - The uploaded file is processed to extract the text data.

2. **Text Preprocessing:**
   - The extracted text data undergoes preprocessing using the `preprocessor` module.
   - The preprocessing may involve tasks like removing special characters, links, and other noise from the text.

3. **Sentiment Analysis:**
   - The VADER sentiment analysis tool from the NLTK library is utilized for sentiment analysis.
   - Three sentiment scores (positive, negative, neutral) are computed for each message.

4. **Categorizing Sentiments:**
   - A custom function `sentiment` is defined to categorize each message into positive, negative, or neutral based on the sentiment scores.

5. **Data Enrichment:**
   - Additional columns (`po`, `ne`, `nu`, `value`) are added to the Data Frame to store positive, negative, neutral scores, and the categorized sentiment, respectively.

6. **User Interaction:**
   - Users can select a specific user or choose to view overall statistics.
   - The selected user is used for further analysis.

7. **Monthly Activity Map:**
   - Monthly activity maps are generated for positive, neutral, and negative sentiments using the `helper. month_activity_map` function.
   - Bar plots are created using Matplotlib and displayed in the Streamlit app.

8. **Daily Activity Map:**
   - Daily activity maps are generated for positive, neutral, and negative sentiments using the `helper. week_activity_map` function.
   - Bar plots are created and displayed.

9. **Weekly Activity Map:**
   - Weekly activity maps are generated for positive, neutral, and negative sentiments using the `helper.activity_heatmap` function.
   - Heatmaps are created using Seaborn and displayed.

10. **Daily Timeline:**
    - Daily timelines are displayed for positive, neutral, and negative sentiments using the `helper. daily_timeline` function.
    - Line plots are created and shown in the app.

11. **Monthly Timeline:**
- Monthly timelines are displayed for positive, neutral, and negative sentiments using the `helper.monthly_timeline` function.
- Line plots are created and presented.

12. **Percentage Contribution:**
- If the selected user is "Overall," the most positive, neutral, and negative contributions are displayed using the `helper. Percentage` function.

13. **User Sentiment Analysis**:
- If the selected user is "Overall," the most positive, neutral, and negative users are displayed based on their contributions.

14. **Word Clouds:**
- Word clouds are generated for positive, neutral, and negative sentiments using the `helper.create_wordcloud` function.
- The generated word clouds are displayed in the app.

15. **Most Common Words:**
- The most common positive, neutral, and negative words are displayed using the `helper.most_common_words` function.
- Horizontal bar plots are created and presented.

DESCRIPTION OF THE WORK

The code implements a Streamlit web app for WhatsApp chat analysis. Users upload a chat file, undergo text preprocessing, and perform sentiment analysis using VADER. The app generates visualizations including activity maps, timelines, and word clouds for positive, neutral, and negative sentiments. Users can select specific users for analysis, and overall statistics are available. The methodology includes data enrichment, user interaction, and various visualizations using Matplotlib and Seaborn.

DIVISION OF WORK

VIDHI:
- Readings of research paper.
- Implementing most active users, busy time of week and month for both quantitative and sentiment analysis.

HIMRAL:
-Readings of research paper.
-data preprocessing and implementation of timeline graph for both quantitative and sentiment analysis.

VANSHIKA:
-Readings of research paper.
-Heat maps and word cloud for both quantitative and sentiment analysis

RESULT-

The web application analyses WhatsApp chats, starting with quantitative insights such as message counts, word frequency, and user activity visualizations. It seamlessly integrates sentiment analysis using VADER to reveal emotional tones over time, showcasing positive, neutral, and negative sentiments. The combined tool provides a holistic view of chat dynamics, offering both quantitative metrics and sentiment trends, enhancing the understanding of communication patterns within the WhatsApp conversations.

CONCLUSION-

In summary, leveraging the WhatsApp application's functionalities alongside Python in Jupyter notebook presents a robust tool for analysing diverse datasets. This exploration covered exporting and preparing WhatsApp data, converting it into a labelled dataset, and visualizing it. Utilizing Jupyter notebook and Python libraries like NumPy, Pandas, Matplotlib, and Seaborn, the analysis involved employing text classification techniques for examination.