TEXT SENTIMENT ANALYZER APP

AIM

The primary aim of this application is to provide a user-friendly, efficient, and visually engaging platform for real-time sentiment analysis. It targets users who need quick feedback on the emotional tone of text, enabling them to make informed decisions based on the sentiment scores.

ABSTARCT

The Text Sentiment Analyzer is a web-based application designed to provide instant sentiment analysis of user-input text. Leveraging the capabilities of the VADER SentimentIntensityAnalyzer, this tool evaluates the emotional tone behind text snippets, categorizing the sentiment as positive, negative, or neutral. The application is built using Streamlit, an open-source app framework, which facilitates rapid development and deployment of data-driven web applications. With a focus on user experience, the application incorporates dynamic CSS for enhanced visual feedback and utilizes intuitive UI components for user interaction. This tool aims to assist users in understanding the sentiment of text content quickly and visually, making it suitable for social media managers, content creators, customer service representatives, and researchers who require efficient sentiment evaluation in their workflows.

METHODOLOGY

1. Setup and Initialization
   The application uses the streamlit library for the web framework and the vaderSentiment library for analyzing sentiment. Upon initializing the app, the VADER sentiment analyzer is set up to process text inputs.

2. User Interface

- Header: A centered header is displayed using HTML and CSS embedded within Streamlit's markdown function, ensuring it is visually appealing and aligned with the app's theme.
- Text Input: Users can enter the text they want analyzed into a textarea, which is styled for visibility and ease of use.
- Analysis Button: A button triggers the sentiment analysis process.

3. Sentiment Analysis

   When the user submits text:

- Computation: The VADER SentimentIntensityAnalyzer computes sentiment scores, which include positive, negative, neutral, and compound metrics.
- Interpretation: Based on the compound score, the sentiment is categorized as Positive, Negative, or Neutral. This categorization is enhanced visually with corresponding emojis to represent the sentiment effectively.

4. Styling and Effects

- Dynamic CSS: The application uses CSS animations for flashing text and changing backgrounds to draw attention to key results and make the interface lively.
- Custom CSS Properties: Styles are applied directly to elements like the text area and the results display to ensure they are clear and effectively integrated into the overall design.

5. Background and Layout

   The background of the app is set with a fixed image, providing a pleasant aesthetic that does not scroll away as the user interacts with the application, ensuring a consistent visual experience.

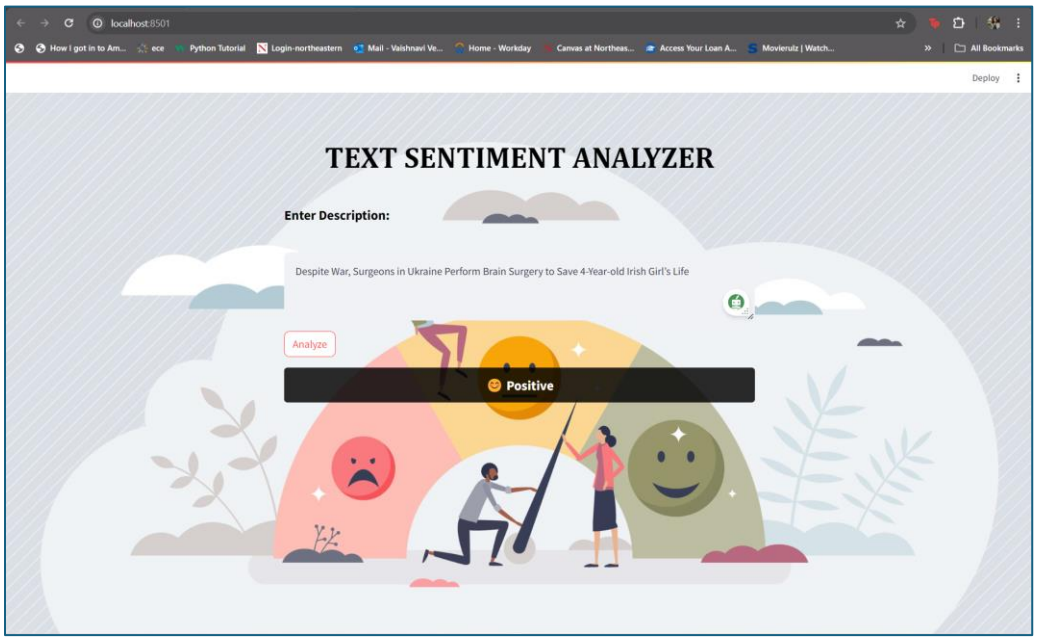
CODE STRUCTURE

The application code is organized into distinct sections:

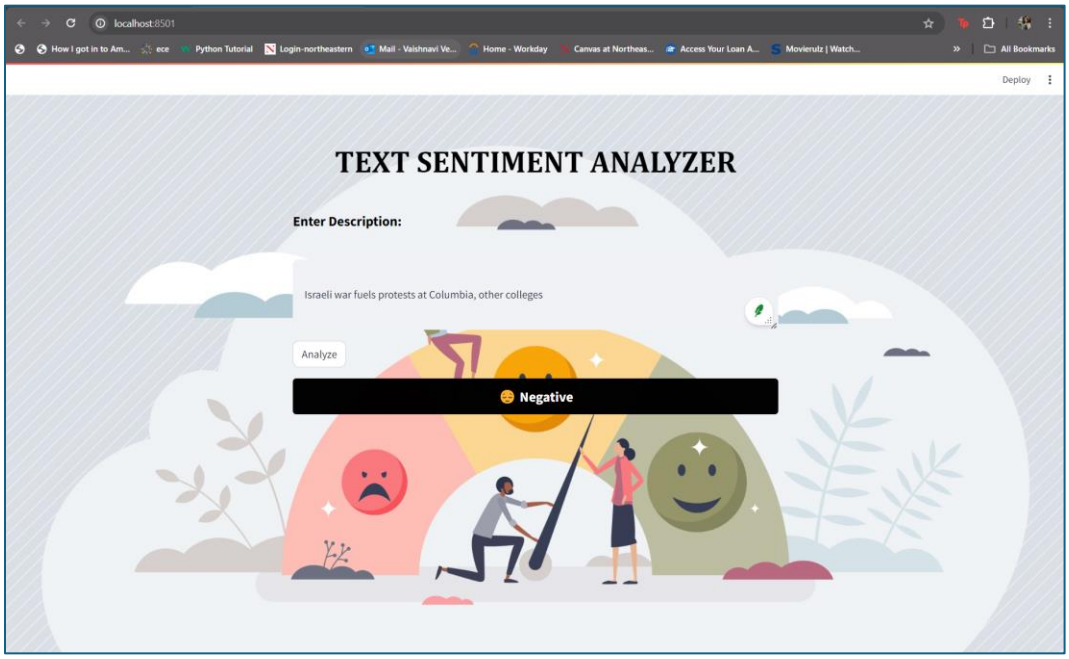
- CSS Definitions: Embedded directly in the Streamlit app using the st.write function with unsafe_allow_html=True to allow custom styles.
- Streamlit Components: Defined using Streamlit's API to maintain a clear and functional layout. Elements like text areas, buttons, and markdown are used to build the interface.
- Sentiment Analysis Logic: Integrated within the button's callback to ensure that sentiment analysis is performed only when the user requests it.

SNIPPETS FROM THE APP

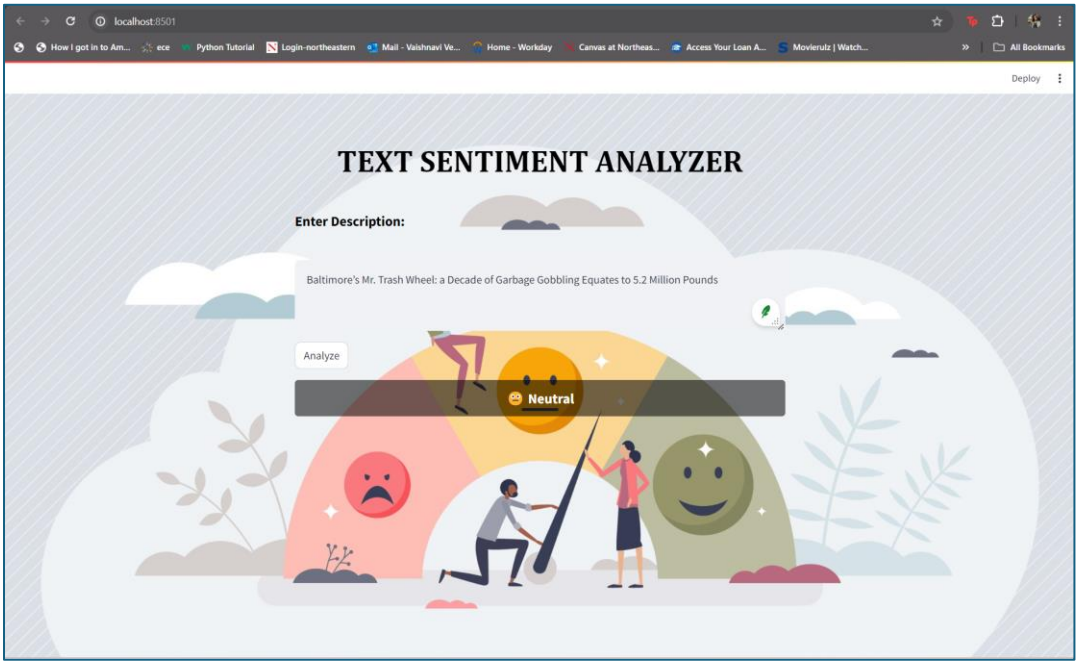
a. Positive Analysis:



b. Negative Analysis:



c. Neutral Analysis:



CONCLUSION

This Streamlit application leverages the VADER sentiment analysis tool to provide instant sentiment assessments in a user-friendly web interface. The application is designed with a focus on accessibility, aesthetics, and practical functionality, making it an excellent tool for casual users and professionals who need quick sentiment analysis of text.