Table of Contents:

1. Executive Summary:

The purpose of the project report is to offer a comprehensive overview of the News Sentiment Analysis project, covering various aspects such as objectives, methodology, achievements, challenges, and future recommendations. The primary goal of the project was to develop a web application that performs sentiment analysis on news articles sourced from different platforms. The application would provide visualizations and insights into public opinion regarding specific topics or events.

The project aimed to address the following key objectives:

1. Develop a user-friendly web interface: The project sought to create an intuitive and user-friendly web interface using either the Django or Flask framework. The interface would allow users to input and analyze news articles from diverse sources.
2. Perform sentiment analysis on news articles: Natural Language Processing (NLP) techniques were employed to extract sentiment from news articles. The sentiment analysis would determine the overall sentiment towards chosen topics or events. Python was utilized for the development of the sentiment analysis code.
3. Generate sentiment scores and calculate overall sentiment: The sentiment analysis code generated sentiment scores for each analyzed article. These scores were then aggregated to calculate the overall sentiment for each topic or event.
4. Implement visualization techniques: The project included the implementation of visualization techniques to present the results of the sentiment analysis. Appropriate charting or graphing libraries were used to create visual representations that would provide insightful interpretations of the sentiment analysis results.
5. Verify news articles from reliable sources: To ensure the authenticity of news articles, the project incorporated a web scraping mechanism to verify if the articles were covered by reliable sources. Two to three reputable websites were identified per topic, and the scraping mechanism searched for the presence of the articles on these sources. Unverified or potentially unreliable articles were indicated as such.

The achievements of the project include:

- Development of a user-friendly web interface for inputting and analyzing news articles.
- Successful implementation of sentiment analysis using NLP techniques.
- Generation of sentiment scores for each article and calculation of overall sentiment for each topic.
- Design and implementation of visualizations to display sentiment analysis results.
- Integration of reliable sources for news article verification.
- Calculation of match scores between social media sentiment and reliable news.
- Provision of insights and visualizations on public opinion towards specific topics or events.

The project also encountered several challenges, such as API authentication and access, data retrieval and parsing, rate limits and API quotas, data privacy and user permissions, platform-specific differences, and maintenance and future updates. These challenges were addressed through appropriate research, implementation strategies, and continuous adaptation.

Future recommendations for the project include the implementation of real-time sentiment analysis for up-to-date insights, expanding the range of reliable sources to include international platforms, enhancing the matching algorithm to handle complex sentiment patterns, incorporating user feedback and sentiment data for more accurate analysis, improving scalability and performance to handle larger data volumes, exploring machine learning techniques for advanced sentiment analysis, and conducting regular maintenance and updates to adapt to API changes and security enhancements.

2. Introduction:

The News Sentiment Analysis project had a primary focus on analyzing news articles sourced from different platforms and extracting sentiment to gain insights into public opinion. Flask, a lightweight and flexible web framework, was chosen as the foundation for the project's web application. Python, a versatile and popular programming language, was utilized for data analysis and visualization tasks. The estimated project duration was set at 8-10 weeks.

Analyzing news articles involved processing the textual data to determine the sentiment expressed within the articles. The sentiment analysis aimed to identify whether the sentiment expressed in the articles was positive, negative, or neutral. This analysis provided a quantitative measure of the overall sentiment towards specific topics or events discussed in the news articles.

Flask, known for its simplicity and flexibility, was chosen as the web framework for the project. It allowed for more control and freedom in designing the application structure. Flask provided the necessary tools and infrastructure to develop a user-friendly web interface, handle HTTP requests, and implement the required functionality for the sentiment analysis platform.

Python, a widely used programming language in the data analysis and machine learning domains, was employed for the data analysis and visualization tasks. Python offers a rich ecosystem of libraries and tools that are well-suited for natural language processing (NLP), sentiment analysis, and data visualization. It provided the necessary capabilities to extract sentiment from news articles, calculate sentiment scores, and determine the overall sentiment for each topic or event.

The estimated project duration of 8-10 weeks allowed for a systematic and phased approach to complete the various stages of the project. The initial weeks focused on project setup and topic selection. During this phase, the project team familiarized themselves with the requirements, technologies, and tools involved. The development environment was set up using Flask, and 2-3 topics were selected for sentiment analysis.

Subsequent weeks involved the development of the web interface. The team designed the layout, navigation flow, and input functionality for news articles. Flask's flexibility allowed for customization and the creation of a user-friendly

interface that facilitated the input and analysis of news articles from different sources.

Python was utilized to implement the sentiment analysis functionality. The team researched and selected appropriate NLP techniques and developed Python code to perform sentiment analysis on the news articles. Rigorous testing and validation were conducted to ensure the accuracy and reliability of the sentiment analysis results.

The project also encompassed the implementation of data visualization techniques to present the sentiment analysis results in an informative and visually appealing manner. Python's rich ecosystem of libraries and tools facilitated the integration of charting or graphing libraries to generate insightful visual representations of the sentiment analysis outcomes.

Throughout the project, regular testing, feedback collection, and refinement were performed to ensure the reliability, performance, and user experience of the web application. The estimated project duration provided ample time for thorough testing and iteration to address any challenges or issues encountered during the development process.

3. Objectives:

The primary objectives of the project were as follows:

- Develop a user-friendly web interface to input and analyze news articles from different sources.
- Perform sentiment analysis on news articles using natural language processing (NLP) techniques.
- Generate sentiment scores for each article and calculate the overall sentiment for each topic.
- Implement visualization techniques to display sentiment analysis results.
- Verify news articles from reliable sources to ensure authenticity.
- Calculate match scores between social media sentiment and reliable news.
- Provide insights and visualizations to users on public opinion towards specific topics or events.

4. Methodology:

The project followed a systematic approach to achieve the objectives:

- Project Setup and Topic Selection
    - Familiarized with project requirements and technologies.
    - Set up the development environment with Django framework.
    - Selected and finalized 2-3 topics for sentiment analysis.
- Web Interface Development
    - Designed the user interface layout and navigation flow.
    - Implemented the input functionality for news articles.
    - Integrated Django framework and developed the basic structure of the web application.
- Sentiment Analysis Implementation
    - Researched and selected appropriate sentiment analysis techniques.
    - Developed Python code to perform sentiment analysis on news articles.
    - Tested and validated the sentiment analysis results.
- Visualization and Insights
    - Designed visualizations to display sentiment analysis results.
    - Implemented charting or graphing libraries for generating insightful visual representations.
    - Tested the visualization functionality and refined as needed.
- Reliable Sources Integration
    - Identified and finalized reliable websites for each topic.
    - Implemented scraping mechanisms to verify news coverage from reliable sources.
    - Handled cases where news articles were not found in reliable sources.
- Match Score Calculation
    - Designed an algorithm for calculating match scores between social media sentiment and reliable news.

    - Implemented the match score calculation logic.
    - Tested and validated the match score calculation for different scenarios.
- Refinement and Testing
    - Reviewed the web interface, functionality, and user experience.
    - Refined and improved the application based on feedback and testing results.
    - Conducted comprehensive testing to identify and fix any bugs or issues.
- Documentation and Finalization
    - Created technical documentation, including specifications and user guide.
    - Finalized the project by performing code review and ensuring compliance with coding standards.
    - Prepared the final project submission, including source code, documentation, and additional artifacts.

5. Achievements:

The project achieved the following milestones:

- Developed a user-friendly web interface for inputting and analyzing news articles.
- Implemented sentiment analysis using natural language processing techniques.
- Generated sentiment scores for each article and calculated the overall sentiment for each topic.
- Designed visualizations to display sentiment analysis results.
- Integrated reliable sources to verify news articles.
- Calculated match scores between social media sentiment and reliable news.
- Provided insights and visualizations on public opinion towards specific topics or events.

6. Challenges:

The project encountered several challenges, including:

- API Authentication and Access: Overcoming the complexities of authenticating and accessing media platforms' APIs.
- Data Retrieval and Parsing: Efficiently handling large amounts of data in various formats.
- Rate Limits and API Quotas: Managing and preventing restrictions imposed by media platforms' rate limits.
- Data Privacy and User Permissions: Ensuring compliance with data privacy policies and obtaining necessary user permissions.
- Platform-Specific Differences: Adapting code to accommodate differences in APIs, data structures, and limitations.
- Maintenance and Future Updates: Addressing API changes and keeping the integration code up to date.

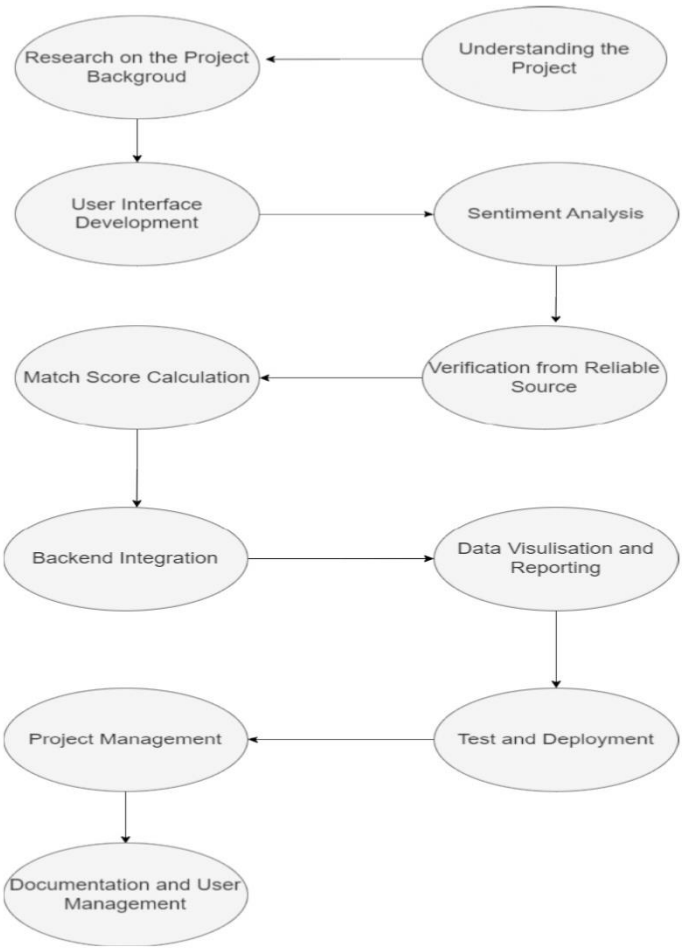
7. Future Recommendations:

To further improve the News Sentiment Analysis project, the following recommendations are proposed:

- Implement real-time sentiment analysis to provide up-to-date insights.
- Expand the range of reliable sources and include international sources.
- Enhance the matching algorithm to handle complex sentiment patterns.
- Incorporate user feedback and sentiment data for more accurate analysis.
- Improve scalability and performance to handle increasing data volumes.
- Explore machine learning techniques for advanced sentiment analysis.
- Conduct regular maintenance and updates to adapt to API changes and security enhancements.

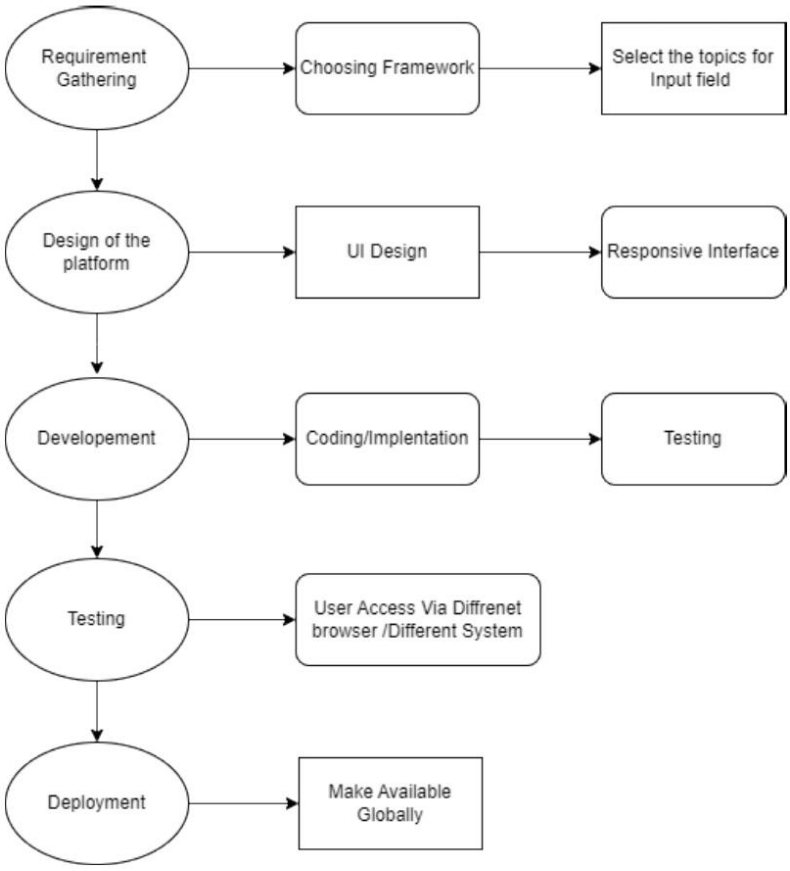
7. **Diagram :**

High Level Diagram of News Sentiment Analysis and Verification Platform

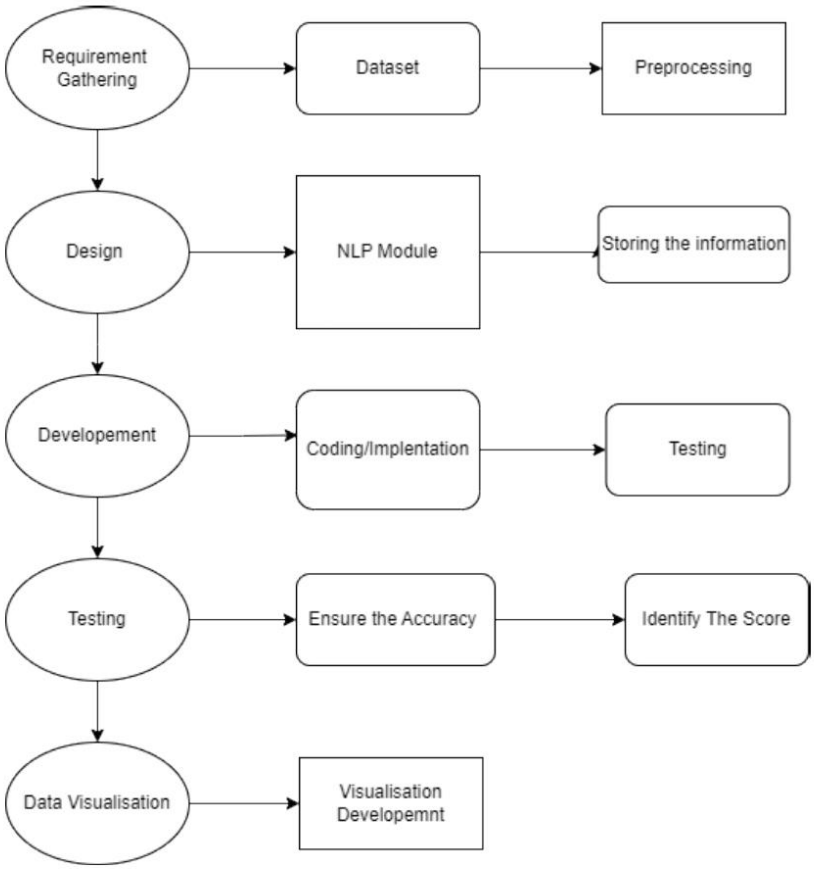


8.

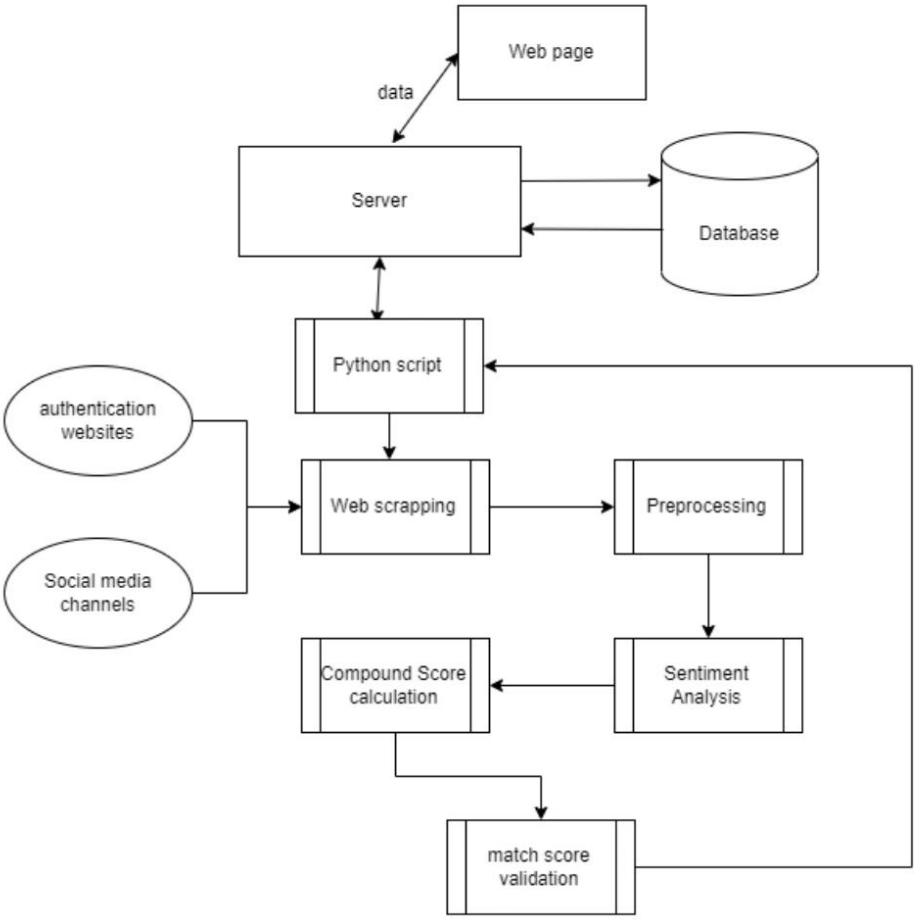
Frontend Development

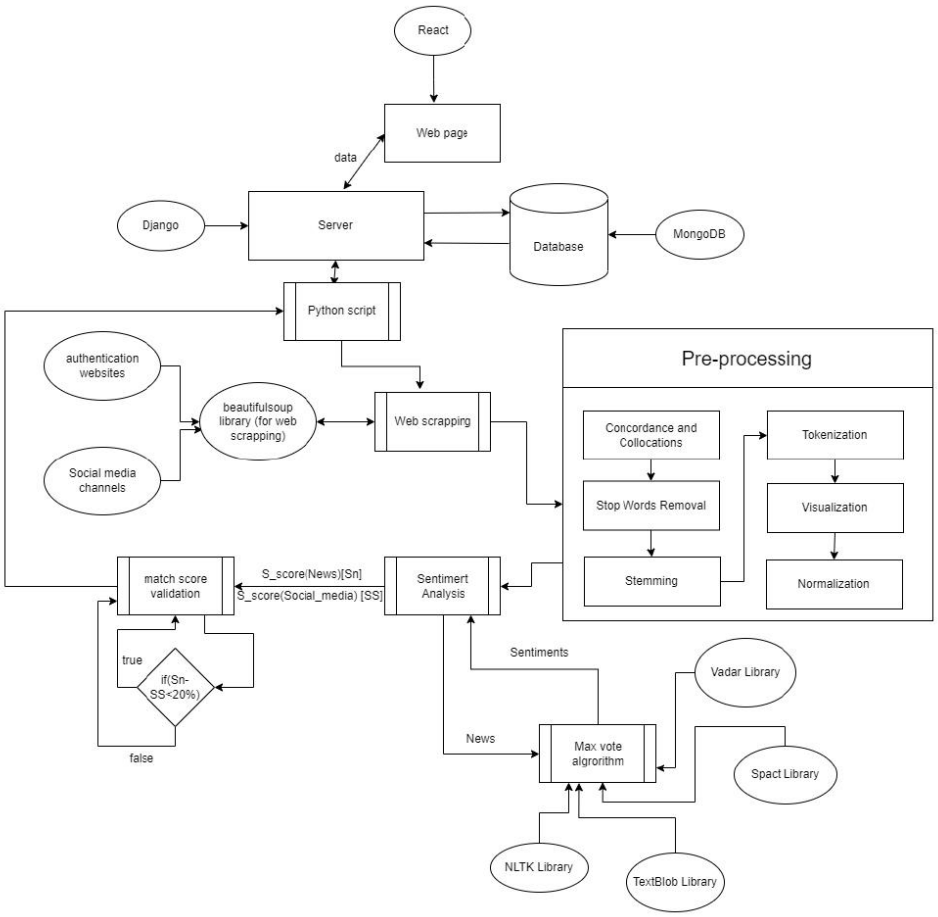


Backend Development



Brief Overview




React
Web page
data
Django
Server
Database
MongoDB
Python script
authentication websites
Social media channels
beautifulsoup library (for web scrapping)
Web scrapping
Pre-processing
Concordance and Collocations
Tokenization
Stop Words Removal
Visualization
Stemming
Normalization
match score validation
S_score(News)[Sn]
S_score(Social_media) [SS]
Sentimert Analysis
true
if(Sn-SS<20%)
false
Sentiments
Vadar Library
News
Max vote algorithm
Spact Library
NLTK Library
TextBlob Library

8. **Conclusion:** The News Sentiment Analysis project successfully developed a web application that performs sentiment analysis on news articles, verifies news from reliable sources, and provides visualizations and insights on public opinion. The project accomplished its objectives within the allocated timeline. The system design and methodology employed contributed to the project's success. With further enhancements and adaptations, the News Sentiment Analysis platform can be a valuable tool for understanding public sentiment towards various topics and events.