Sentiment Analysis Project Functional Specification

Supervisor
Greg Doyle

Student
Mantas Macionis

Academic Year
2023/2024

Contents

Abstract

This project aims to develop a web application that conducts traditional sentiment analysis on Reddit posts and provides a comparative assessment with AI-driven sentiment analysis using the ChatGPT API. The application aims to summarize the sentiments expressed on Reddit topics, offering users valuable insights and time savings.

By leveraging traditional sentiment analysis techniques and AI-powered sentiment analysis, the application will provide users with an understanding of public opinion on topics of their choice discussed on Reddit. This will enable users to make informed decisions based on accurate and timely data, improving their knowledge of public perception and assisting in decision-making processes.

Introduction

This document will cover the key factors relating to the functionality and development of the Sentiment Analysis web application. FURPS will cover development related headings and additional headings are included to cover topics relating to the obstacles in development and the target audience for this application. A system use case diagram shows the interactions between the users and the system, and both brief and detailed use cases are included to demonstrate functionalities in practice.

Project FURPS

FURPS is a framework used to evaluate the quality of a software product in terms of its Functionality, Usability, Reliability, Performance, and Supportability.

Functionality

The core functionality of this project is to allow the user to either choose a general, or board specific search, and a term which they would like to analyse. The Web application will then output analysed sentiment on the topic a user has chosen. The output will include an analysis obtained using traditional sentiment analysis techniques, and an analysis generated by the ChatGPT API. Once the output has been generated, it will be saved to a database. The user will be able to navigate through the website to the history section and view the past sentiment generated and the time and date it was generated.

Usability

To be most effective, the web application UI will be very simple, With basic menu systems and large buttons for key functionality. This will allow users to quickly navigate through the website and generate sentiment analysis on the topics of their choice. In areas where misinterpretation is possible, brief descriptions will be displayed to assist users who may be having issues. The key function, which is

sentiment analysis, will be available immediately to a user once they have logged in to the site.

Reliability

The success of this web application relies heavily on the availability and performance of the reddit website for retrieving sentiment analysis data. With an estimated 52 million daily users, reddit is a highly popular platform, and as such, downtimes are rare, and any site-wide issues are typically resolved swiftly.

To ensure continued access to past sentiment analysis results in the event of downtime, to store past outputs, a database is used that links user accounts with their respective data. This approach not only enhances the overall reliability of the system, but also provides users with a convenient way to access their previously analysed information.

Performance

The performance of this website will be highly important to the overall user experience. Research suggests that users expect web pages to load within 2-3 seconds, anything beyond this can lead to frustration and abandonment. The speed of retrieving posts from the reddit API will also be another key element of performance. To ensure the API retrieves posts fast, the number of posts to be retrieved by the API will be controlled, A balance between effective sampling and API limitations is needed, as the reddit API has limits and exceeding them will lead to throttling and possible errors.

Supportability

This website will be a web-application, web applications have several advantages including:

- Accessibility: A web application can be accessed from any device with an internet connection, making it accessible to users regardless of their location or device type
- Maintenance: Web applications are easier to maintain then mobile applications, as updates are pushed to all users simultaneously.
- User Experience: A web application can provide a consistent user experience across devices. Negating possible issues which can arise mobile platforms.
- Cost: There are costs involved with deploying an application on the two most popular mobile platforms, iOS and Android.

Target Users

The scenarios where this web application can provide usefulness to users include:

- Brand Monitoring
- Competitor Analysis
- Political Analysis
- Product Development
- Content Recommendations
- Social Trend Prediction

The target audience for this web application is broad, individuals and companies looking to utilise the reddit platform for analysis will see the value this application can bring.

Obstacles

Obstacles to developing a sentiment analysis web application include choosing the right analysis type and creating effective predictions for different topics. The accuracy and usefulness of the app can be impacted by the analysis type, such as Basic Sentiment analysis struggling with double sentiment. Aspect-Based Sentiment analysis is more thorough but can be time-consuming. Additionally, the accuracy of predictions can be affected by the use of sarcasm and topic-specific terms, as seen in stock market discussions.

Additionally, the reddit API from which the original posts will be obtained, is subject to rate limits and other restrictions. Code involving the API will need to be optimal to ensure data is retrieved in a timely manner and without error.

Use Case Diagram
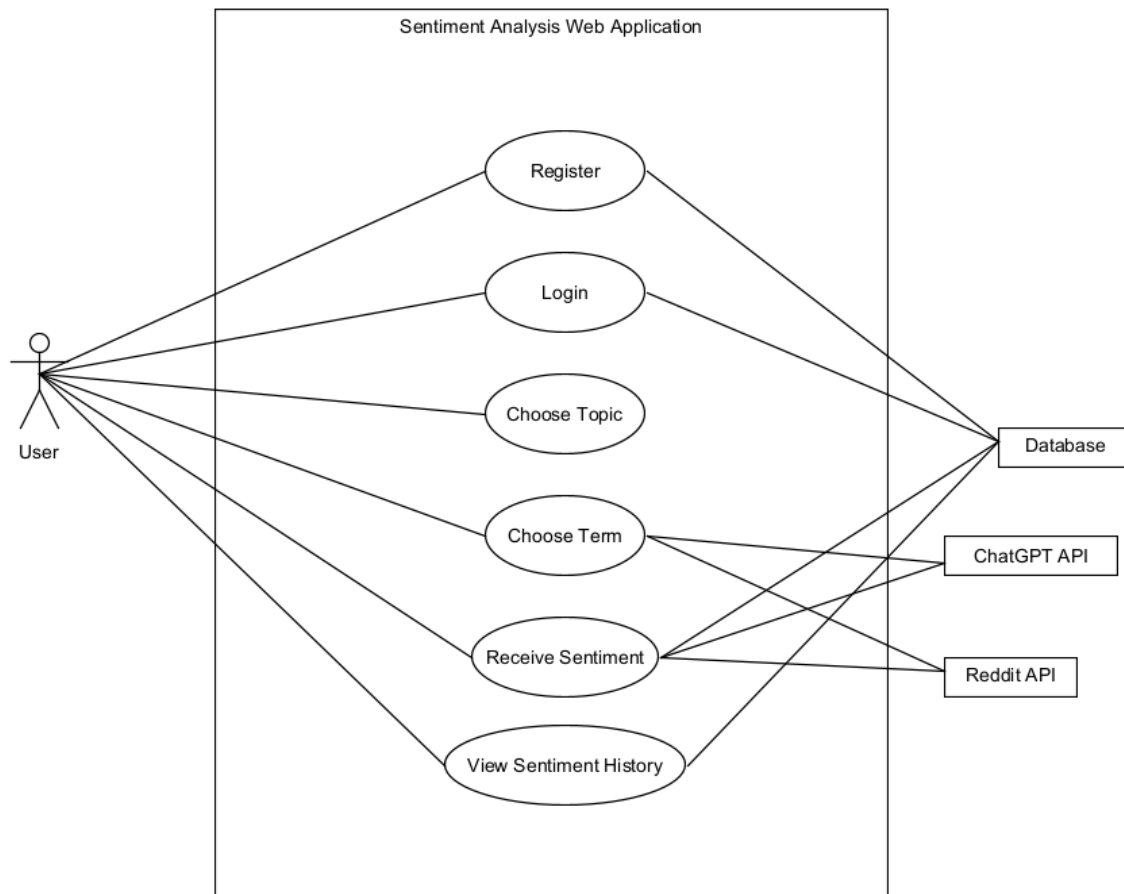
Brief Use Cases

Use Case: Register

Actors: User, Database

Description: This use case begins when the user enters the website, they are brought to a login page and choose to register, the user enters their information, Username and Password. Their information is then stored on the database.

Use Case: Login

Actors: User, Database

Description: User, Database

Description: This use case begins when the user enters the website and chooses to log into their account, they end their username and password, which is verified by the database, after verification they are signed into their account.

Use Case: Choose Topic

Actors: User

Description: Before the user enters the term, they would like to receive sentiment about, they choose the topic which best suits it. This is so the most accurate word dataset can be used to give predictions.

Use Case: Choose Term

Actors: User, Reddit API

Description: The user inputs the term which they want to analyse the sentiment of, once they have entered the term, the reddit API is utilized to get information about their topic of choice and forward it to the algorithm and dataset used to make a prediction.

Use Case: Receive Sentiment

Actors: User, Database

Description: This use case begins after the user has chosen their topic and term, sentiment analysis is output to the user and the record is stored.

Use Case: Receive AI generated Sentiment.

Actors: User, Reddit API, ChatGPT API, database

Description: This use case begins after the user has chosen their topic of choice and term, sentiment analysis generated by the ChatGPT API displayed to the user and the record is stored.

Use Case: View Sentiment History

Actors: User, Database

Description: This use case begins when the user decides to check the history of their sentiment analysis requests, the terms they have chosen to receive sentiment about are kept in a record, when they decide to view their historical searches, the database is utilized to output the information.

Detailed Use Cases

Use Case: Register
Actors: User, Database

Brief Description: This use case begins when the user wants to register an account on the sentiment analysis website.

Main Success Scenario:

1. The User wants to register.
2. The User enters their Username, Email and Password
3. The WebApp takes this information and sends it to a secure database.
4. The user is directed to the main page.

Alternatives:

- The username already exists, the password is bad, or the email is already in use.
    1. The user is given an error message personalised to their issue.
    2. The user is redirected to enter their details again.

Use Case: Login

Actors: User, Database

Brief Description: This use case begins when the user wishes to login to their account on the website.

Main Success Scenario:

1. The User wants to login.
2. The User enters their username and password.
3. Their input is compared to login info stored by the database, it is verified.
4. The user is logged in and directed to the main page.

Alternatives:

- The user enters incorrect login information.
    1. An error message appears notifying the user their details are incorrect.
    2. The user is redirected to attempt to enter their details again.

Use Case: Choose Topic

Actors: User

Brief Description: This use case begins when the user is asked to choose the topic which suits their intended topic of analysis.

Main Success Scenario:

1. The user has a term in mind they want to research.
2. The user chooses from a dropdown menu the topic which best suits their term.
3. Depending on their choice, the dataset used to make predictions is altered for more accurate results.

Alternatives:

- The user does not see a topic which describes their term accurately.
    1. The user can choose an option called "Other" which will allow them to search any topic with a broader dataset.

Use Case: Choose Term

Actors: User, Reddit API

Brief Description: The user enters the term of which they would like the sentiment analysed.

Main Success Scenario:

1. The User enters the term they want an analysis of

2. The reddit API is utilized by the web app to retrieve relevant posts to the entered term.

Alternatives:

- The user enters a term with invalid characters.
    1. An error message appears notifying the user they have entered invalid characters.
    2. The user begins at step 1 again and must re-enter the term they wish to analyse.

Use Case: Receive Sentiment
Actors: User, Database

Brief Description: This use case begins when the user has entered their term of choice, and an analysis is being generated for them.

Main Success Scenario:

1. The user receives an analysis of the term they have chosen.
2. The analysis is added to the database.

Use Case: Receive AI generated Sentiment.
Actors: User, ChatGPT API, database

Brief Description: This use case begins when the user has entered their term of choice, and an AI generated analysis is created for them.

Main Success Scenario:

1. The user receives an AI generated analysis of the term they have chosen.
2. The analysis is added to the database.

Use Case: View Sentiment History
Actors: User, Database

Brief Description: The user wishes to view the results of their past sentiment searches.

Main Success Scenario:

1. The user enters the View sentiment history page.
2. The can either choose from a table or their searches or use a search bar to use check their previous searches.
3. When they have found the record, they are looking for, they click on it and view the sentiment which was generated previously.

Alternatives:

- The user has never searched for any terms, or they have chosen to delete their old records.
    1. A message is displayed to the user notifying them they have no historical records.

2. The user is given an option to go straight to the main page to begin the process of analysing sentiment on a topic of their choice, to generate a record.