FEBRUARY 15, 2023

REDDIT ANALYZER

Final Project Report

KARAN SANGHA
UNIVERSITY OF COLORADO - BOULDER

Table of Contents

Project Description

The Reddit Analyzer app is a web-based tool that allows users to view statistics of a subreddit and extract data from Reddit. The app uses the Python Reddit API Wrapper (PRAW) to collect data and provides statistical analysis of the posts and comments, and to view analytics and charts about the selected content.

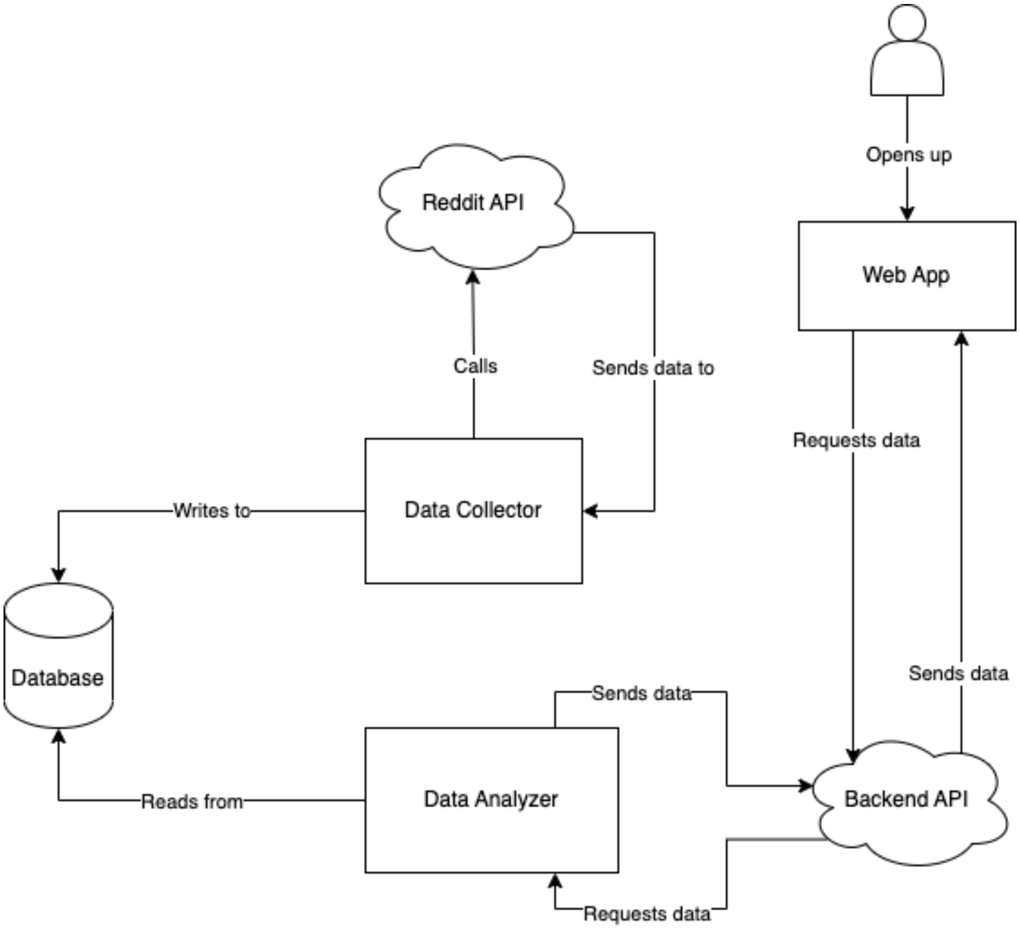
The app provides the following functionalities:

1. Data Extraction: Users can extract data from a subreddit and save it in a database. The app uses PRAW to extract the data and stores it in a SQLite database. Users can then view the stored data by using the provided API endpoints.
2. Statistics: Users can view various statistics for a given subreddit, such as the number of posts per day, the number of comments per day, and the average number of upvotes per day. The app provides a visualization service that analyzes data from the database and generates charts using the Highcharts library for these statistics.

The Reddit Analyzer app is built using Flask, a popular Python web framework, and is containerized using Docker. It utilizes various Python packages, including PRAW, pika, flask, PyTest and SQLite3 to extract and analyze data from Reddit. The app's web interface is built using HTML, CSS, and JavaScript. Currently, the app can produce the following charts: the number of posts per day, the number of comments per day, and the average number of upvotes per day.

Overall, the Reddit Analyzer app is a powerful tool for social media analysts, marketers, and anyone else looking to gain insights into popularity of content on Reddit. With its user-friendly interface and advanced analysis features, the app makes it easy to get meaningful insights from the vast amount of data available on the platform.

Whiteboard Architecture



Processes and Services

The Reddit Analyzer app consists of multiple processes and services that work together to provide a seamless experience for the user. Here is an overview of each of the key components:

Web Server

The Web Server process is responsible for serving the Flask application that powers the Reddit Analyzer app. The Flask application provides the user interface for the app, allowing users to enter a subreddit name and view statistics on that subreddit. The Web Server process listens on port 5000 and communicates with the database and visualization service to handle user requests.

Database

The Database is responsible for storing all the data for the Reddit Analyzer app. It stores the top posts data retrieved from the Reddit API for a subreddit. The database is built using SQLite, a lightweight, serverless, and self-contained database engine that is easy to set up and use.

Data Collection and Analysis

The visualization service first queries the app's database to collect the data needed for the charts. Specifically, the service collects the number of posts, comments, and upvotes for each day. It then uses this data to calculate the average number of upvotes, comments and upvotes per subreddit for each day.

Once the data has been collected and analyzed, the visualization service uses the Highcharts library to create interactive charts that display the data in an intuitive and easy-to-understand way. The charts are displayed on a separate page within the app, and users can interact with the charts to explore the data in more detail.

Testing
The Testing process is responsible for running the automated tests for the Reddit Analyzer app. The tests are written using the PyTest framework and test the functionality of the Flask application, data collector module, data analysis module and the database. The tests run automatically when the code is pushed to the main branch in the app's source code repository, ensuring that the app is always working as expected.

Metrics Collection
The Metrics Collection process is responsible for collecting and aggregating performance metrics for the Reddit Analyzer app. The flask-prometheus-metrics library is used to automatically collect and expose performance metrics for the app, including request counts, request latency, and response codes. The metrics are stored in a Prometheus database and can be visualized using Grafana or other similar visualization tools. These metrics provide insight into the performance and usage of the Reddit Analyzer app and can be used to optimize and scale the app as needed.

Justifications for design decisions

- Flask for Web Framework - Flask is a lightweight web framework that allows us to build a simple REST API for the Reddit Analyzer app. Flask is also easy to configure, making it easy to set up routes and endpoints for the app.

- PRAW for Reddit API - PRAW (Python Reddit API Wrapper) was chosen for accessing the Reddit API due to its simplicity and ease of use. PRAW provides a Pythonic interface for accessing Reddit's API, allowing us to easily retrieve information on posts, comments, and subreddits.

- SQLite for Database - SQLite was chosen as the database for the Reddit Analyzer app due to its lightweight and portable nature. SQLite is a self-contained, serverless database that is easy to set up and requires no configuration.

- Flask-Prometheus-Metrics for Metrics - Flask-Prometheus-Metrics was chosen to provide metrics on the Reddit Analyzer app due to its ease of use and compatibility with Prometheus. Flask-Prometheus-Metrics automatically provides metrics on various Flask endpoints, making it easy to monitor the app's performance and usage.

- Visualization Service for Charts - A visualization service was included in the Reddit Analyzer app to provide easy-to-understand charts and graphs for data retrieved from the SQLite database. Highcharts was chosen as the charting library due to its easy integration with Flask, customization options, and wide range of chart types.

- PyTest for Testing – PyTest was chosen as the testing framework for the Reddit Analyzer app due to its ease of use, extensibility, and compatibility with Flask. PyTest makes it easy to write and run tests for the app's various endpoints, ensuring that they are working correctly.

- RabbitMQ for Collaboration Messaging: RabbitMQ is an open-source message broker that was used for inter-process communication between the web server and the data processing service. It was chosen for its flexibility, reliability, and ease of use.

- Continuous Integration & Deployment: Github Actions were used to automate the testing and deployment process. The workflow comprises of a testing suite and push mechanism to Heroku. The code is pushed to Heroku only if the test suite passes. This ensures that the application is always deployed with the latest code changes, and that any issues are caught early in the development cycle.

System Requirements and Testability

System Requirements

The Reddit Analyzer App has several system requirements that must be met for the application to function correctly. These requirements include:

- Python 3.8 or higher
- Docker and Docker Compose
- RabbitMQ
- SQLite database

Testability

The Reddit Analyzer App was designed with testability in mind. The app has several unit tests and integration tests built in using the PyTest testing framework. These tests ensure that the app functions correctly and meets the requirements of the end-users.

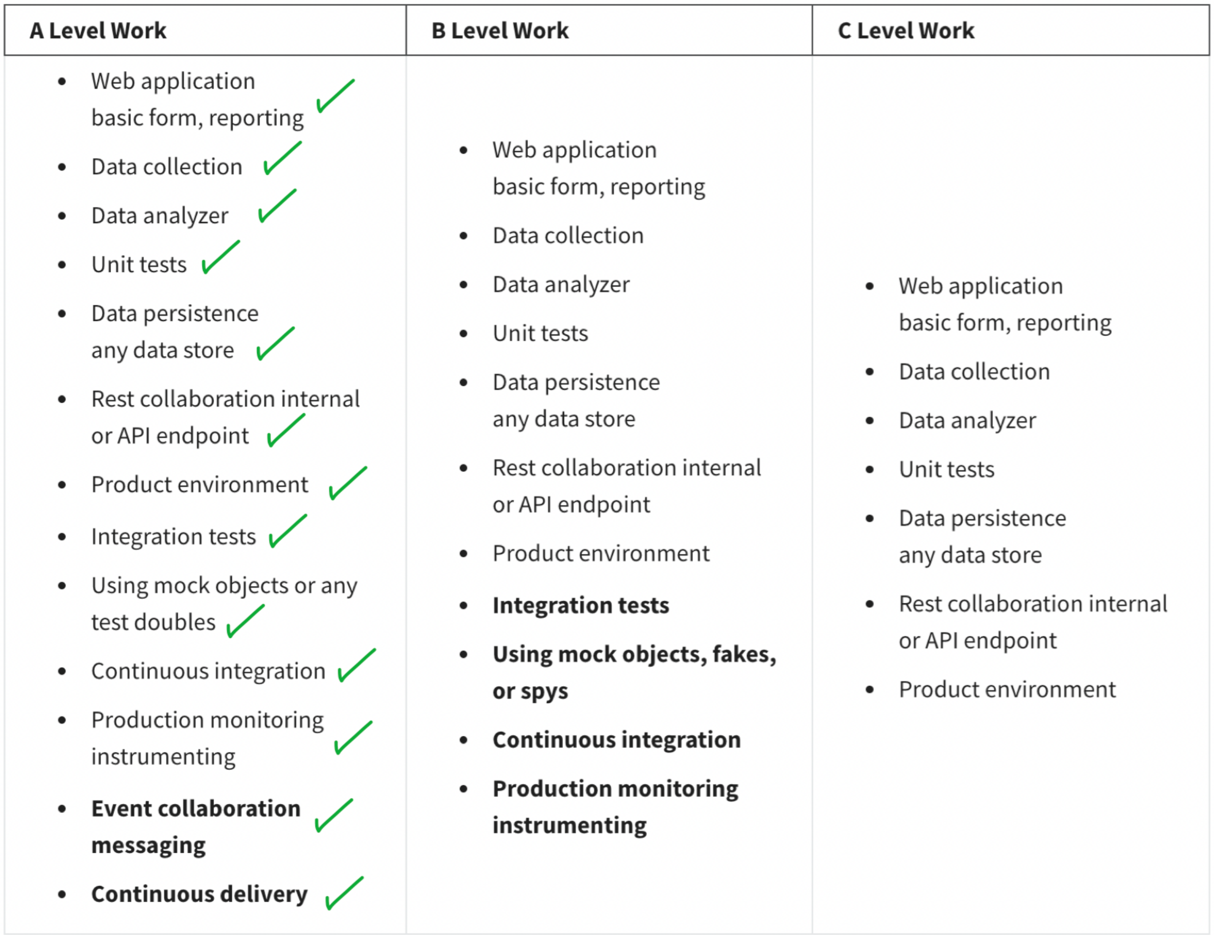
The app also includes continuous integration and deployment using GitHub Actions. The continuous integration process includes running the test suite before deploying the code to Heroku. This ensures that any new code that is deployed to the production environment is tested and functional.

Additionally, the app includes a Docker Compose configuration file, which can be used to spin up a local development environment that mimics the production environment. This makes it

easy for developers to work on the app and test new features locally before pushing changes to the production environment.
Overall, the Reddit Analyzer App was designed with testability in mind to ensure that the app is functional and meets the needs of the end-users.

Rubric Self-Check

I believe I have checked-off all the items in the "A Level Work" section of the rubric. Off course, the peer reviewer holds all the power to review the existence of each work item.



A Level Work	**B Level Work**	**C Level Work**
• Web application basic form, reporting ✓ • Data collection ✓ • Data analyzer ✓ • Unit tests ✓ • Data persistence any data store ✓ • Rest collaboration internal or API endpoint ✓ • Product environment ✓ • Integration tests ✓ • Using mock objects or any test doubles ✓ • Continuous integration ✓ • Production monitoring instrumenting ✓ • **Event collaboration messaging** ✓ • **Continuous delivery** ✓	• Web application basic form, reporting • Data collection • Data analyzer • Unit tests • Data persistence any data store • Rest collaboration internal or API endpoint • Product environment • **Integration tests** • **Using mock objects, fakes, or spys** • **Continuous integration** • **Production monitoring instrumenting**	• Web application basic form, reporting • Data collection • Data analyzer • Unit tests • Data persistence any data store • Rest collaboration internal or API endpoint • Product environment