

Project Charter

Project Title: GitHub Issue Analysis Tool for Python Poetry

Project Start Date: Mar 20, 2025

Projected Finish Date: May 20, 2025

Project Manager: Palak Gupta, pgupta13@umd.edu

Developers:

- Bhavna Kumari, bhavna@umd.edu
- Bolla Sai Saketh, sakethb@umd.edu
- Souhardya Pal, spal05@umd.edu

Project Description/Objectives: The project aims to analyze GitHub issues from the Python Poetry repository to generate insights about contributor activity, issue categories, and trends over time. The analysis will involve extracting data using the GitHub API, designing an application with visualizations (charts), implementing features for insights generation, and presenting findings.

Main Project Success Criteria (MOV): The project will be successful if at least 90% test coverage is achieved, while delivering all milestones on time, successful extraction of issue data, present actionable insights from issue data.

Approach:

The project involves analyzing GitHub issues from the Python Poetry repository to extract meaningful insights about contributor activity, issue trends, and project patterns. The approach begins by leveraging the GitHub API to extract issue data, which is formatted into a structured JSON file for further analysis. Using tools like Mermaid.js, the team will design an Entity-Relationship Diagram (ERD) and class diagrams to model the data structure and application architecture. The project then transitions into the implementation phase, where an application is developed to perform three distinct analyses on the issues, supported by visualizations created with Matplotlib. These analyses may include metrics such as issue resolution times, contributor activity levels, or popular labels.

The project will also focus on software quality by writing unit tests to achieve at least 90% code coverage, ensuring the robustness of the application. Finally, findings will be presented through a live demonstration of the application, highlighting key insights and the overall development process. Throughout the project, GitHub Issues and Kanban boards will be used for task tracking and collaboration, ensuring transparency and efficient workflow management. This systematic approach balances technical implementation with clear communication and deliverable-focused milestones.

Assumptions/Risks:

- Assumption: GitHub API will provide uninterrupted access for data extraction within rate limits.
- Risk: Team collaboration challenges may arise due to uneven participation or technical difficulties during implementation/testing phases.

Roles and Responsibilities

Role	Name	Contact Information
Project Manager - Oversee project progress and ensure deadlines are met	Palak Gupta	pgupta13@umd.edu
Tester/Analyst - Write unit tests and verify code coverage	Souhardya Pal	spal05@umd.edu
Developers - Implement application features/Extract GitHub issues and format JSON file	Bhavna Kumari	bhavna@umd.edu
Developers - Create ERD and class diagrams using Mermaid.js	Bolla Sai Saketh	sakethb@umd.edu

Milestone 2:

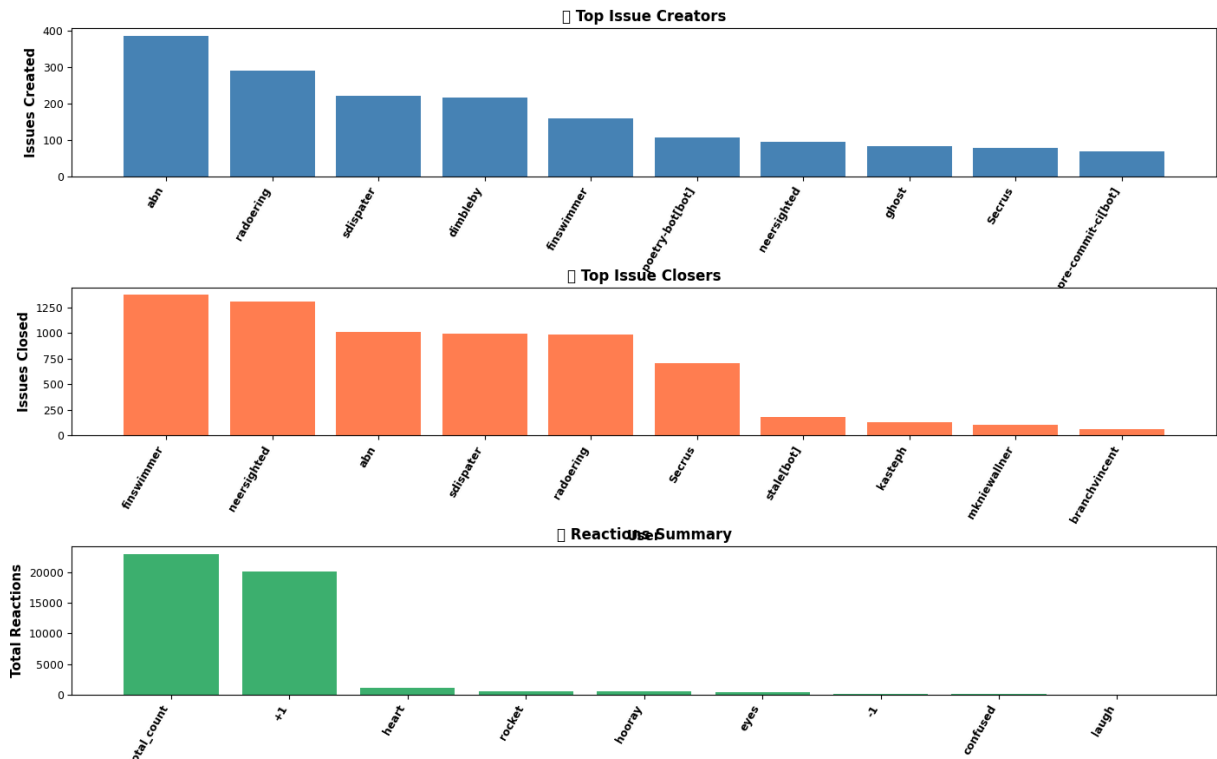
We implemented three analytical features using issue data from the GitHub repository of the Poetry project. Each feature extracts insights from `poetry_data.json` and visualizes patterns related to labels, contributors, and resolution times.

Feature 1: Top Contributors and Community Reactions

Developed by Bolla Sai Saketh

The bar chart shows three unique dimensions of issue-related activity in the Poetry repository:

- Primary Issue Contributors: The people who started the most issues.
- Principal issue resolvers: the persons who solved the most problems.
- Reaction Overview: An overall count of emoji reactions summed across all issues. Each bar represents user activity or reaction type, helping to understand both contributor engagement and community sentiment.



Key Findings:

Highly Engaged Participants:

The first panel shows users like abn, raddening, and sdispater at the center of issue generation, representing their continuous involvement or contribution.

Responsible maintainers:

The second panel highlights finswimmer and neersighted as major closers, suggesting high involvement in triage and maintenance tasks.

Evident Community Support:

The third panel shows that +1 and heart reactions responses are common, suggesting that the subjects of debate elicit positive comments and engagement.

Fair Participation:

Some users appear in both the creator and closer lists (e.g., abn), reflecting their overall involvement during the issue's lifecycle.

Feature 2: Most Frequently Used Labels with Average Resolution Time

Developed by Souhardya Pal

This feature provides a dual-perspective analysis of issue labeling in the Poetry repository. It visualizes:

- **Issue Count per Label:** Indicates how often a label is used across issues.
- **Average Resolution Time:** Represented as annotations above each bar, this shows the average number of days it takes to resolve issues tagged with the corresponding label.

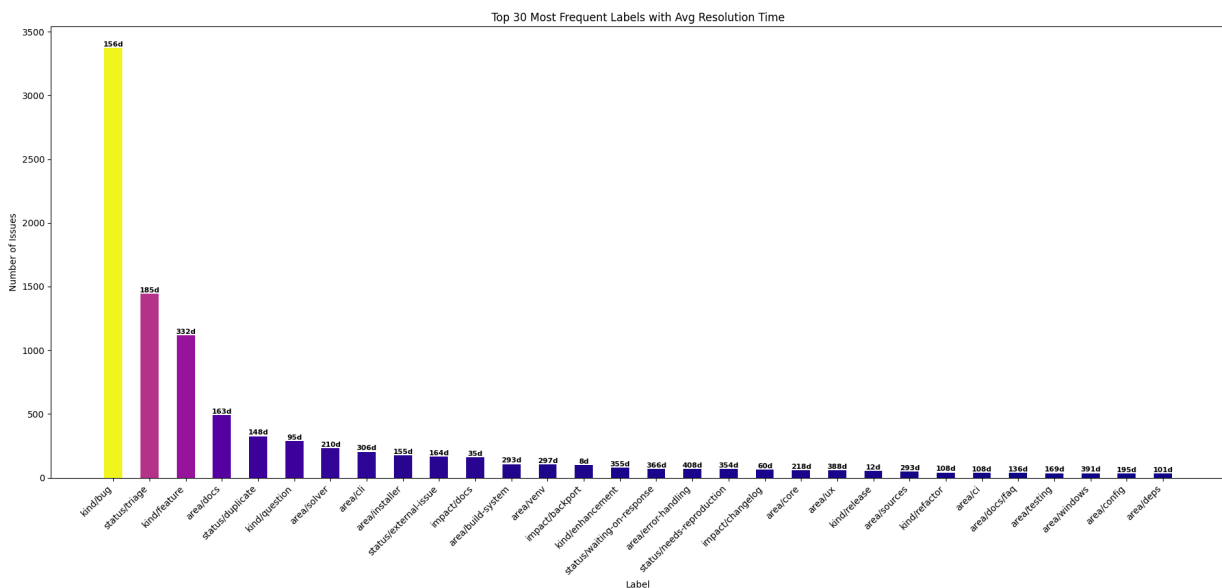
Each bar in the chart represents a label, with its height corresponding to the number of issues tagged with that label. The number above each bar shows the average time it took to resolve (close) those issues. It offers quick insight into which types of issues tend to linger and which are resolved quickly.

Why This Matters:

- **Workflow Optimization:** Helps project maintainers understand which types of issues occur most frequently and how quickly they are resolved.
- **Resource Allocation:** By identifying high-volume labels with longer resolution times, teams can prioritize improvements or assign more contributors to specific areas.
- **Process Monitoring:** The average resolution time per label acts as a benchmark for future performance tracking.

Key Findings (Example Use Case):

- Labels like bug, dependencies, and status/triage have a high issue count.
- Certain labels exhibit longer-than-average resolution times, indicating potential complexity or lower contributor attention.
- Visual annotations help distinguish quick-to-resolve labels from those that may be bottlenecks in the issue triage pipeline.



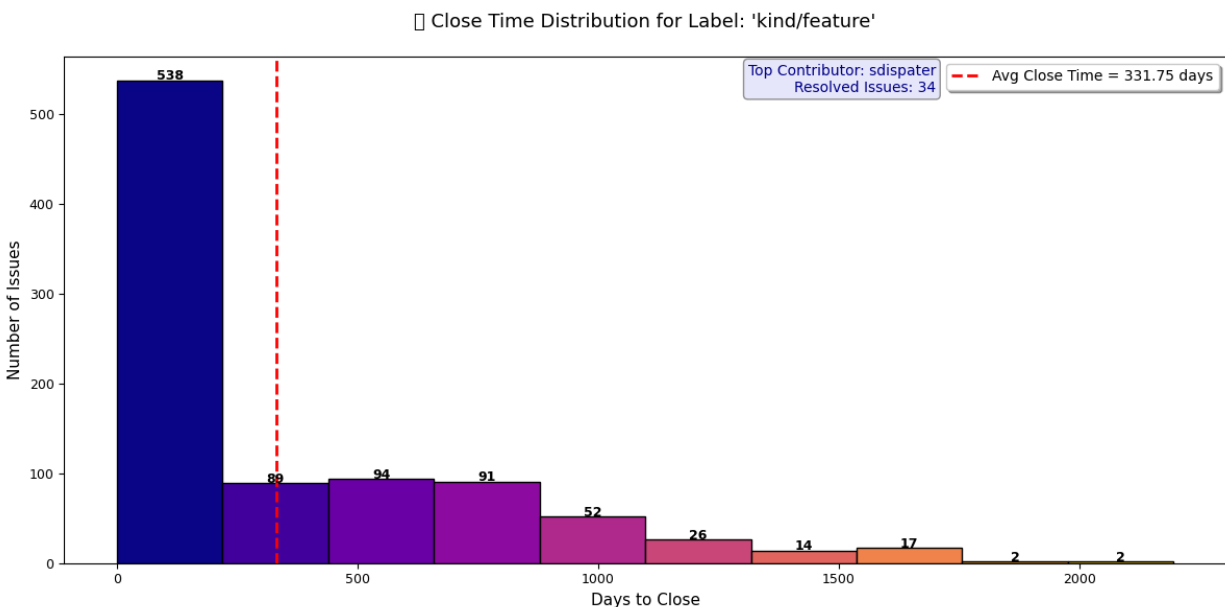
Feature 3: Label-Based Insights

Developed by Bhavna Kumari

Accepts a user-input label and returns metrics like Average close time of the label, No of issues, Top contributor, Resolved issues by Top contributor, Time to close, and Total issues. This histogram visualizes :

How long it took to close GitHub issues tagged with the 'status/triage' label. Each bar represents a time range (in days), showing how many issues were closed within that range. A red dashed line shows the average close time, and a contributor annotation box highlights who resolved the most issues for that label.

Below is an example figure for the label : **kind/feature**



Why This Matters:

Maintainers can identify inconsistencies in triage workflows.

Long-lived triaged issues may indicate:

- Low contributor attention
- Complex problems
- Label misuse

The contributor annotation helps surface leadership patterns and potential bottlenecks.

Sign-off: Palak Gupta