Project Milestone 2 Report

YAOJUN LIU, University of British Columbia, Canada
SASIVIMOL SIRIJANGKAPATTANA, University of British Columbia, Canada

1. Introduction

This Milestone 2 report documents progress on the first research question (RQ1) of the project. The objective is to examine whether repository- and user-level characteristics predict pull request (PR) resolution time.

The project investigates three research questions:

- **RQ1:** How do repository popularity (stars, forks) and user influence (followers, account age) relate to PR resolution time?
- **RQ2:** How do text characteristics of PRs (length and sentiment) vary across repositories of different popularity tiers and users of different influence levels?
- **RQ3:** How does AI coding agent contribution behavior vary across repositories with different popularity levels?

This milestone focuses exclusively on RQ1. RQ2 and RQ3 will be completed for the final report.

2. Data Wrangling and Feature Construction

Three datasets were used: pull requests, repositories, and users. All timestamps were converted to `datetime` format and merged using `repo_id` and `user_id`. Only closed PRs were retained to compute resolution time.

2.1 Resolution Time

PR resolution time was computed as:

$$\text{resolution\_time\_hours} = \frac{\text{closed\_at} - \text{created\_at}}{3600}.$$

User account age was defined as:

$$\text{user\_account\_age\_days} = \text{created\_at(PR)} - \text{created\_at\_user}.$$

2.2 Zero-Inflated Predictors

Repository popularity (stars, forks) and user influence (followers) are heavily zero-inflated (60–80% zeros). To preserve dataset representativeness, zero values were retained. All predictors and the outcome were log-transformed using $\log(1 + x)$ to reduce skew.

3. Preliminary Results for RQ1

3.1 Correlation Analysis

A correlation matrix of log-transformed variables shows weak associations between predictor variables and PR resolution time:

- $\text{corr}(\log(\text{resolution time}), \log(\text{stars})) \approx 0.20$
- $\text{corr}(\log(\text{resolution time}), \log(\text{forks})) \approx 0.22$
- $\text{corr}(\log(\text{resolution time}), \log(\text{followers})) \approx 0.079$

Authors' Contact Information: Yaojun Liu, University of British Columbia, Department of Computer Science, Okanagan, BC, Canada; Sasivimol Sirijangkapattana, University of British Columbia, Department of Computer Science, Okanagan, BC, Canada.

- $\mathrm{corr}(\log(\text{resolution time}), \log(\text{user age})) \approx 0.038$

All correlations are below 0.25, indicating negligible linear relationships between popularity/influence and PR processing time.

3.2 Regression Analysis

A log-linear regression model was fitted using 817,668 closed pull requests:

$$\log(1+Y) = \beta_0 + \beta_1 \log(1+\text{stars}) + \beta_2 \log(1+\text{forks}) + \beta_3 \log(1+\text{followers}) + \beta_4 \log(1+\text{user age}) + \varepsilon.$$

Model performance:

$$R^2 = 0.049, \quad \text{Adjusted } R^2 = 0.049.$$

Thus, only about 4.9% of the variability in PR resolution time is explained by these predictors.
Estimated coefficients (all $p < 0.001$):

- β_1 (log-stars): $+0.0163$
- β_2 (log-forks): $+0.2223$
- β_3 (log-followers): -0.0120
- β_4 (log-account age): $+0.0099$

Although statistically significant due to the large sample size, these effects are extremely small in magnitude.

3.3 Interpretation

These findings indicate that:

- Repository popularity has little to no predictive power for PR resolution time.
- User influence (followers, account age) also has negligible predictive impact.
- The small coefficient sizes and low R^2 suggest that PR resolution time is primarily driven by unobserved factors such as maintainer availability, CI outcomes, review workload, and project-specific processes.

This aligns with prior software engineering studies reporting that PR review speed is difficult to predict from repository- or user-level metadata alone.

4. Limitations

- Resolution time is highly skewed, with many short-duration PRs and a long tail of very long ones.
- Important complexity indicators (changed files, additions, deletions, comments) were not used in this milestone but may explain additional variance.
- Zero-inflation limits the predictive strength of popularity and influence metrics.

5. Next Steps: RQ2 and RQ3

RQ2: How do text characteristics of pull requests vary across repositories of different popularity levels and users of different influence levels?

We will analyze whether PR text features—such as sentiment and text length—differ across repositories with varying popularity (low, medium, high) and across users with different influence levels (based on follower count). This analysis will help determine whether more popular repositories or high-influence users tend to produce PRs with distinct linguistic or stylistic characteristics.

Methodology. PR, repository, and user data are merged using `repo_id` and `user_id`. Text-based features are computed by concatenating PR titles and bodies into a single field, from which `text_length` and sentiment polarity are derived using the `TextBlob` library. Repository popularity tiers are generated using star count quantiles, while user influence tiers are based on follower quantiles. Exploratory visualizations (e.g., boxplots) will then be used to compare sentiment and text length across popularity and influence levels.

Features Used. The analysis will rely on the following fields:

- `title`, body (raw PR text)
- `full_text` (combined text used for NLP features)
- `text_length` (character length of PR text)
- `sentiment` (sentiment polarity extracted via TextBlob)
- `stars` (repository popularity)
- `followers` (user influence)
- `repo_id`, `user_id` (merge keys)

RQ3: How do AI agents behave across repositories of different popularity levels?

We will analyze how AI-generated pull requests differ when submitted to repositories with varying popularity levels. The focus is on understanding whether AI agents contribute differently in terms of submission volume, merge outcomes, and PR lifetime across low, medium, and high-popularity repositories.

Methodology. PR metadata will be merged with repository information using `repo_id`. Repository popularity will be categorized into three tiers (low, medium, high) based on star count quantiles. For all PRs authored by AI agents, we will compute behavioral metrics—AI PR volume, closure/merge rate, and PR lifetime (closed_at − created_at)—within each popularity tier. Exploratory data analysis, including summary statistics and visualizations, will be used to compare AI behavior across these tiers.

Features Used. The analysis will rely on the following fields:

- `agent` (AI-agent: Claude_Code, Copilot,..)
- `repo_id` (repository identifier)
- `stars` (repository popularity measure)
- `state` (merged/closed/open)
- `created_at` and `closed_at` (timestamps used to compute PR lifetime)
- `id` (pull request identifier)

6. GitHub Repository

Code and data processing pipeline:
https://github.com/liuyaojun1/542.git