DATA542 Project Plan

Author: Zihao Sheng, Wei Li from Group 19

The GitHub pull request dataset contains about **940,000 PR records** stored in a main table linked to auxiliary tables such as pr_commit, pr_commit_file, pr_review, and repository. The main table provides core metadata (timestamps, merge status, task type, repo ID), while the auxiliary tables supply commit details, file changes, review activities, and repository attributes. After joining these sources, each PR contains roughly **90–100 features**.

Preliminary inspection shows many PRs are incomplete or uninformative—such as PRs with **no commits**, long-idle PRs, and trivial or bot-generated PRs. These records do not reflect meaningful behavior and are removed prior to analysis.

Research Topic 1: Factors Predicting the Quality of Agent-Generated PRs

This topic aims to identify which PR-level characteristics best predict the quality of AI-generated PRs.

1. **Construct a quality score：** Normalize turnaround time, review activity, and merge status into a 0–100 metric.
2. **Filter PRs：** Retain only PRs with ≥1 commit; flag or remove empty or incomplete PRs.
3. **Assemble predictors：** Extract commit count, additions/deletions, patch size metrics, task type, programming language, repository metadata, and AI agent type.
4. **Model quality predictors：** Use linear regression and stepwise regression to identify which factors significantly influence the quality score.

Research Topic 2: Common Failure Patterns in Agent-Generated PRs

This topic aims to identify and characterize recurring failure patterns in AI-generated pull requests, including incomplete, idle, or structurally problematic contributions.

1. **Identify failure outcomes:** Define failure signals such as non-merged PRs, long inactivity, unbalanced or oversized patches, and PRs with no commits.
2. **Categorize failed PRs:** Group failures into empty patches, oversized or chaotic patches, high-churn edits, and long-idle PRs lacking review progress.
3. **Extract failure-related features:** Collect indicators including commit count, additions/deletions, patch scale, task type, language, agent type, and repository attributes.
4. **Analyze failure patterns:** Compare distributions and proportions across agents and use simple clustering or heatmaps to reveal recurring structural failure modes.

Research Topic 3: Repository Collaboration Activity and Interaction Structures

1. **Identify activity patterns:** Measure repository activity using PR volume, merge rate, contributor count, and review intensity to capture variation across projects.
2. **Build collaboration networks:** Construct repo–repo interaction graphs based on shared contributors and cross-project PR behaviors.
3. **Extract network features:** Collect centrality, connectivity, and clustering metrics to describe structural roles of each repository.
4. **Analyze collaboration dynamic:** Compare network patterns across languages and activity levels to reveal hubs, isolated repos, and cross-project cohesion.