

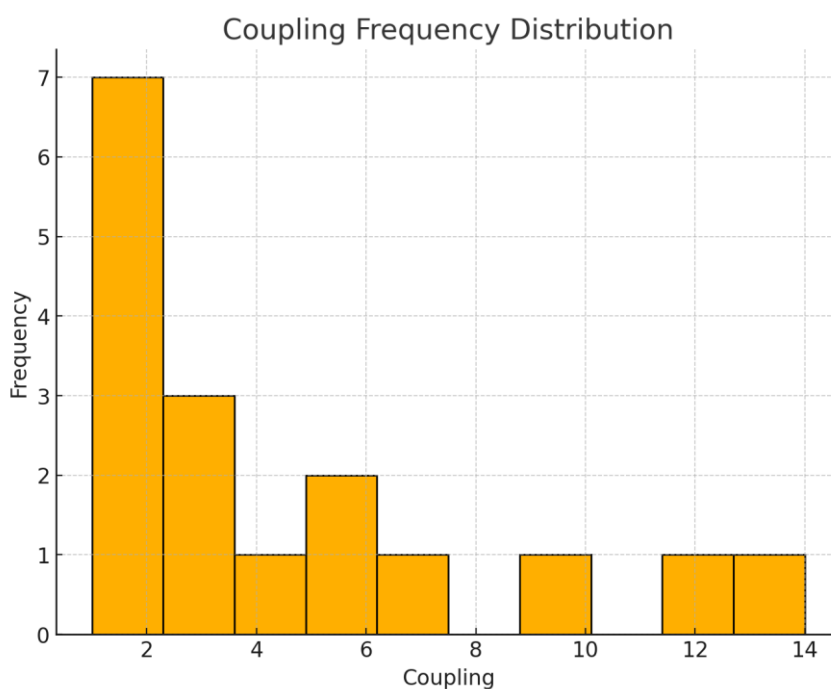
Assignment-2

Introduction

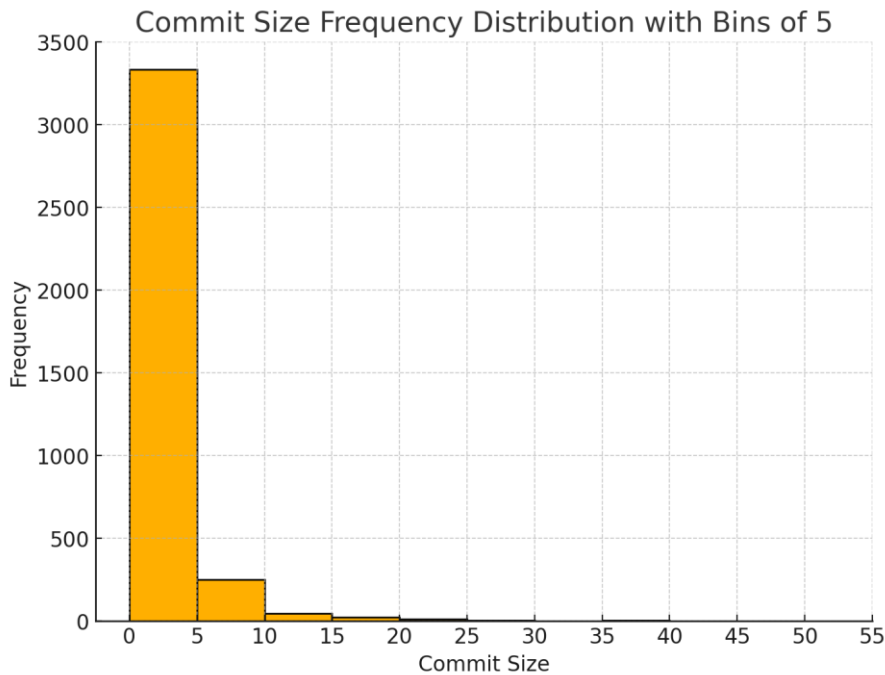
For this analysis, I have chosen three open-source Python projects: Flask, SQLAlchemy, and Fail2Ban. Flask, developed by the Pallets team, is a lightweight web framework commonly used for building web applications. SQLAlchemy is an automated tool for SQL injection and database takeover. Fail2Ban is a log-parsing tool designed to prevent brute force attacks by banning suspicious IP addresses. These projects represent a diverse set of applications, making them ideal for evaluating their architectural quality.

The metrics selected for comparison are coupling, maintainability index, and commit size. Coupling measures the dependency between different modules in a project. High coupling can lead to challenges when making modifications, as changes in one module may require changes in several others. Maintainability Index (MI) assesses how easy it is to modify, extend, or maintain the source code over time. A higher MI indicates better code quality in terms of maintainability. Lastly, commit size represents the number of files changed in each commit, reflecting the modifiability or changeability of the project. Large commit sizes might indicate a complex codebase that requires significant changes for each new feature or bug fix.

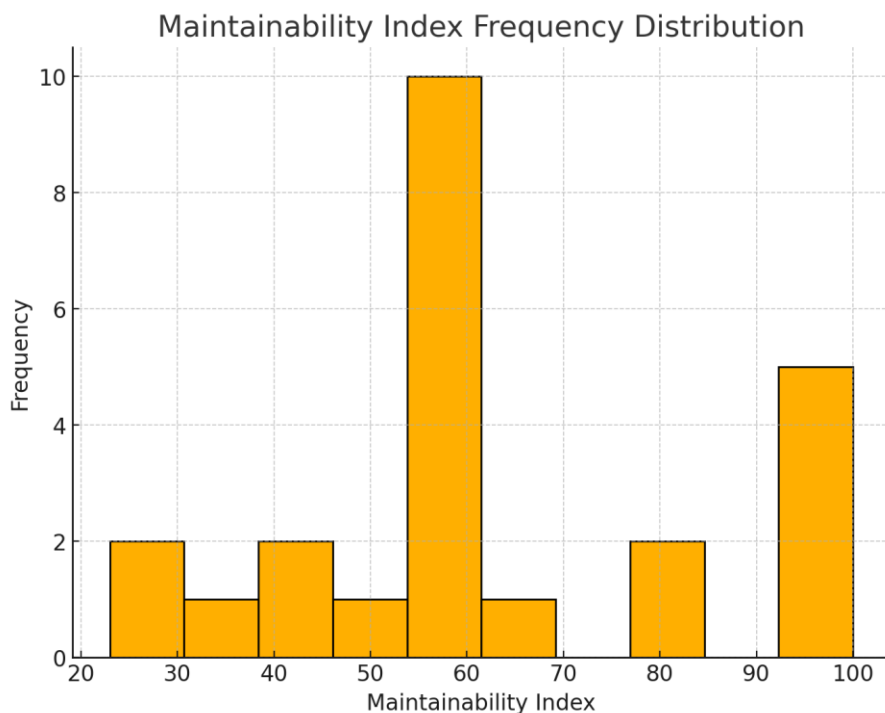
Flask



The coupling histogram shows that most modules in Flask have low coupling values, meaning they are not heavily dependent on other modules. This indicates a well-structured design with limited interdependencies, which simplifies maintenance and enhances modularity. A few modules exhibit slightly higher coupling, but these are rare. Overall, Flask demonstrates a favorable architecture with low coupling, allowing for easier updates and enhancements.

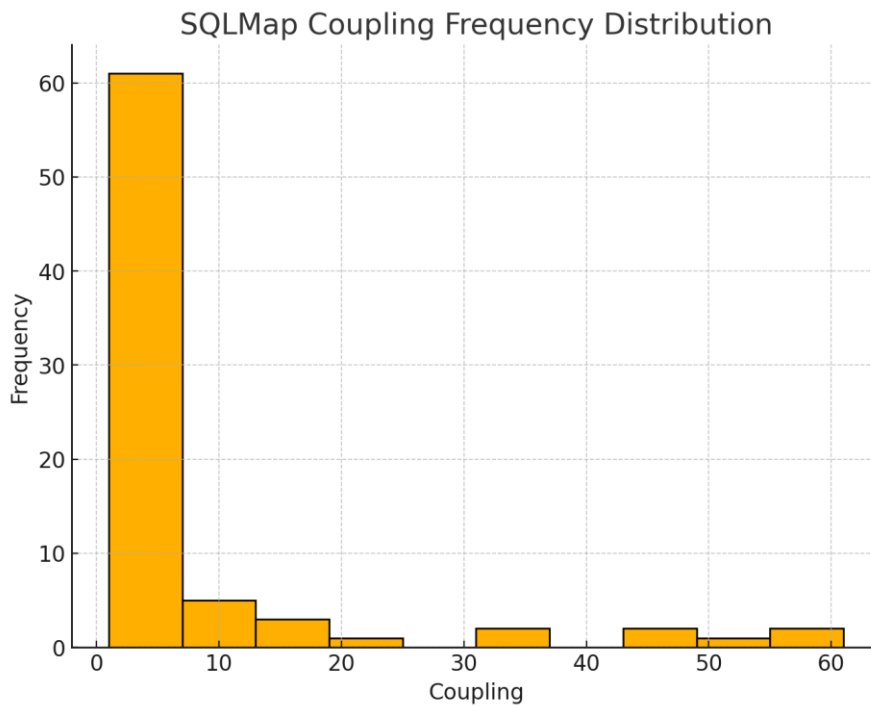


The commit size distribution reveals that the majority of commits involve small changes, typically affecting fewer than 5 files. This reflects a codebase where changes are localized and relatively easy to manage. Very few commits involve more than 50 files, indicating that large-scale changes are rare, further supporting modifiability. This localized change pattern suggests that adding new features or fixing bugs in Flask generally requires minimal effort.

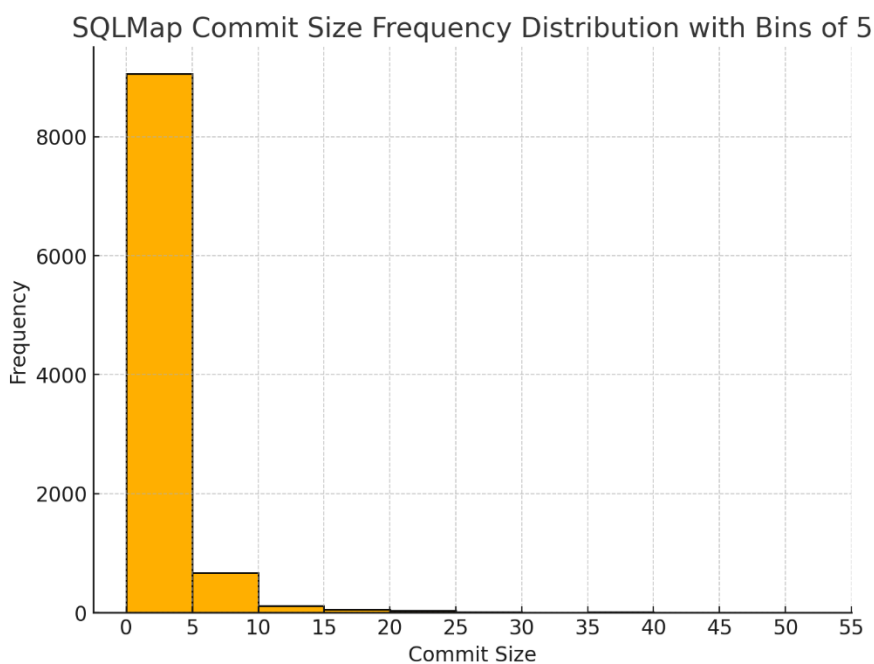


The maintainability index histogram shows that most of Flask's modules have a high index value, with many above 80. This suggests that the project is highly maintainable, with code that is easy to support and modify. A few modules have lower scores, which may require additional attention, but the overall maintainability remains strong. This high index indicates that Flask is well-suited for long-term development and upkeep.

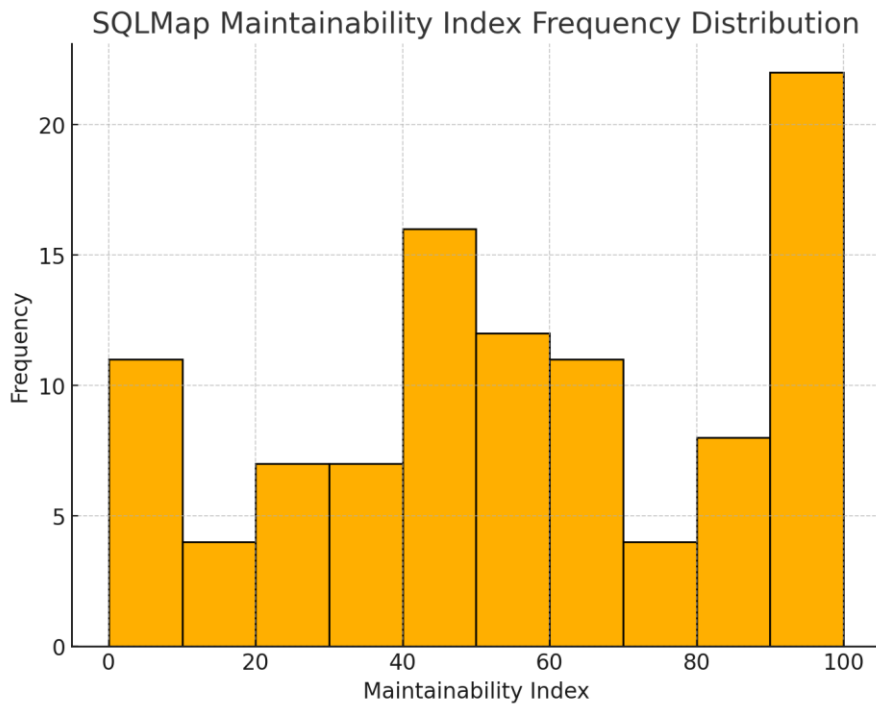
Sqlmap



The coupling histogram shows a wide distribution, indicating that some modules have a higher level of interdependency. This could suggest more complex relationships between different parts of

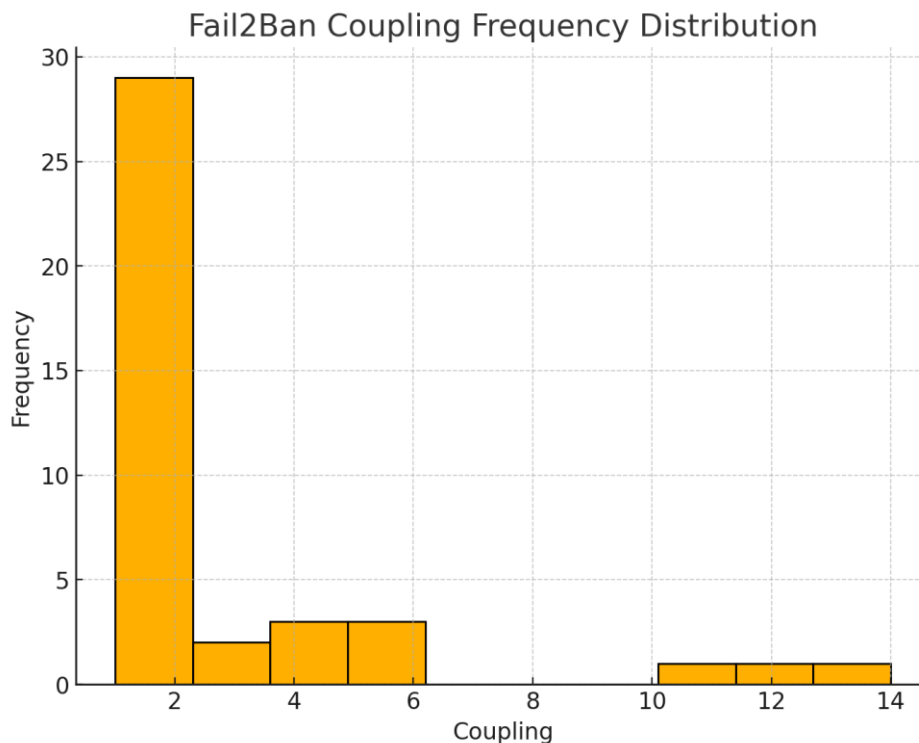


The commit size histogram reflects that most changes involve fewer than 5 files, with very few large commits exceeding 50 files. This suggests that changes are generally localized, reducing the complexity of modifications and making the system more adaptable.

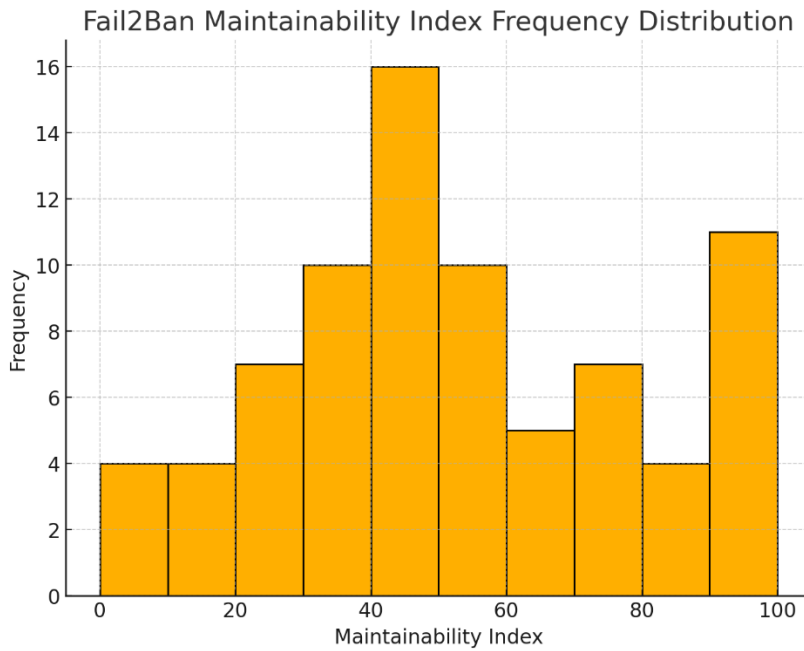


The maintainability index histogram shows a diverse range of values, with some modules having very high scores (100), indicating excellent maintainability, while others have lower scores, suggesting that parts of the codebase may be harder to maintain or modify.

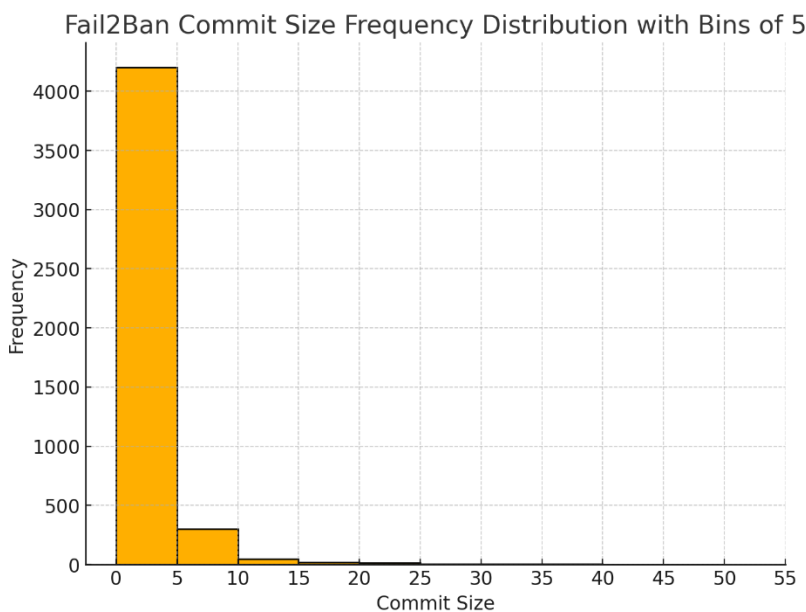
Fail2ban



The histogram indicates that most modules in Fail2Ban have low coupling, with a small number of modules showing slightly higher interdependency. This design pattern minimizes the complexity of changes, making it easier to maintain and modify individual components without widespread impact.



The maintainability index histogram reveals a wide range of scores. Many modules show high maintainability (scores near 100), but there are a few outliers with low maintainability, which may require more significant effort to support and modify.



Most commits involve small changes (typically affecting fewer than five files), reflecting localized updates to the codebase. Larger commits are rare, suggesting that Fail2Ban is well-structured and allows for incremental, non-disruptive changes to the project.

Comparison

Comparing the three project Flask, SQLAlchemy, and Fail2Ban across the metrics of Coupling, Commit Size, and Maintainability Index reveals key insights into their structural quality. In terms of Coupling, Flask and Fail2Ban exhibit relatively low coupling, indicating a modular design with minimal dependencies between components. SQLAlchemy, on the other hand, shows a broader distribution of coupling, suggesting that some parts of the system may have higher interdependency, which could complicate modifications. For Commit Size, all three projects

generally display localized changes, with the majority of commits involving fewer than five files. This indicates that changes are typically incremental and well-contained, which is a positive sign of good modifiability. However, SQLAlchemy and Fail2Ban have a few instances of significantly larger commits, suggesting more complex updates on occasion. Regarding Maintainability Index, Flask shows consistently high scores, making it the most maintainable of the three projects. Fail2Ban has a mix of high and low scores, indicating that while many modules are highly maintainable, some require additional attention. SQLAlchemy, meanwhile, presents the widest range of scores, suggesting variability in maintainability across different parts of the codebase. Overall, Flask demonstrates the strongest architecture in terms of modularity and maintainability, while SQLAlchemy and Fail2Ban show more variability, particularly in maintainability and coupling, though they remain manageable.