Mining Software Repositories to Improve Refactoring Assistants

Diogo Luís Araújo de Faria { up201907014@up.pt }

Abstract. Software development faces an ever-constant evolution, increasing its complexity over time, and with it, particular needs start to arise. Among them, some of the most important can be considered the need for efficient and maintainable code. Refactoring, as "the process of changing a software system in a way that does not alter the external behaviour of the code yet improves its internal structure", plays a role in addressing some of these needs. However, traditional approaches to refactoring can be perceived as cumbersome, tedious, and time-consuming. Traditionally, refactoring requires a developer to identify opportunities and then decide which refactoring they should apply. After it is performed, they still need to ensure that the refactoring preserves the overall system behaviour. In the context of an existing refactoring recommendation plugin, LiveRef, we explored an approach that uses machine learning to improve the identification and suggestion process, particularly on the Extract Method and Extract Class refactorings. Within this plugin, we replace the conventional threshold-based method with a dynamic classification model, allowing it to continuously learn as it is being used and adapt to new software contexts. We leveraged the power of data mining in existing software repositories to extract real-life data, creating a rich and varied dataset that enhances the model's training and ensures that it applies to diverse real-world scenarios. To validate what we developed, an automated analysis was performed in order to be able to compare the number of suggestions the plugin recommended, using both the original plugin version and the one after our changes, with real-life developer refactoring data. We believe that the use of real-life data is crucial in the refactoring suggestions process, as a textbook refactoring isn't necessarily what developers are going to perform in their day-to-day operations. Thus a refactoring recommendation tool that is able to take into account a larger variety of contexts while adapting to its user over time will increase a developer's use of it, leading to better-maintained software systems.

Supervisor: Ademar Aguiar; **Second Supervisor:** Sara Fernandes.

Keywords: Refactoring. Code Quality Metrics. Mining Software Repositories. Data Mining.

ACM Computing Classification System:

- Software and its engineering → Software creation and management → Software post-development issues → Software reverse engineering
- Software and its engineering → Software notation and tools → Software maintenance tools