December 2020

Use of Machine Learning To Generate Estimates of Code Review Time and Effort

Marko Ivanković

Goran Petrovic

Use of Machine Learning To Generate Estimates of Code Review Time and Effort

ABSTRACT

Providing accurate estimates of time required to perform code reviews can enable practitioners to prioritize reviews and can help improve productivity. Currently, factors such as the number of lines of code to be reviewed, or the number of code files, are provided to the reviewer. Such factors are relatively poor predictors of the actual effort required. This disclosure describes the use of machine learning techniques to generate effort estimates of the actual human time required to perform a code review. The machine learning model is trained on a dataset that is generated based on historical code modifications and corresponding reviews. The target variable for the machine learning model is to predict the actual time taken for code review by a practitioner. Multiple data points from each historical code review are used to obtain features that are used to train the machine learning model. The model is periodically retrained to ensure accuracy of predictions.

KEYWORDS

- Code review
- Peer review
- Effort estimation
- Changelist
- Software development
- Software engineering
- Machine learning

BACKGROUND

Code review is a systematic examination of software source code (generally conducted by peers), intended to find bugs and to improve the code quality. The code review process typically covers aspects such as error detection, vulnerability identification, malware discovery, adherence to best practices, etc. When notifying practitioners that they have been assigned a piece of code for review, it is useful to provide an estimate of the time required to conduct the review. Providing accurate estimates can enable practitioners to prioritize reviews and can play a significant role in improving productivity.

Unfortunately, current techniques for providing effort estimates are limited to showing reviewers related information such as the lines of code to be reviewed, or number of files to be reviewed, as an indication of the likely complexity of the review. However, such indicators are difficult to use for efficient prioritization, since the time taken to review the same number of lines of code can vary widely.

DESCRIPTION

This disclosure describes the use of machine learning techniques to generate effort estimates of the actual human time required to perform a code review. The machine learning model is trained on a dataset that is generated based on historical code modifications and corresponding reviews. The target variable for the machine learning model is to predict the actual time taken for code review by a practitioner. Multiple data points from each historical code review are used to obtain features that are used to train the machine learning model.

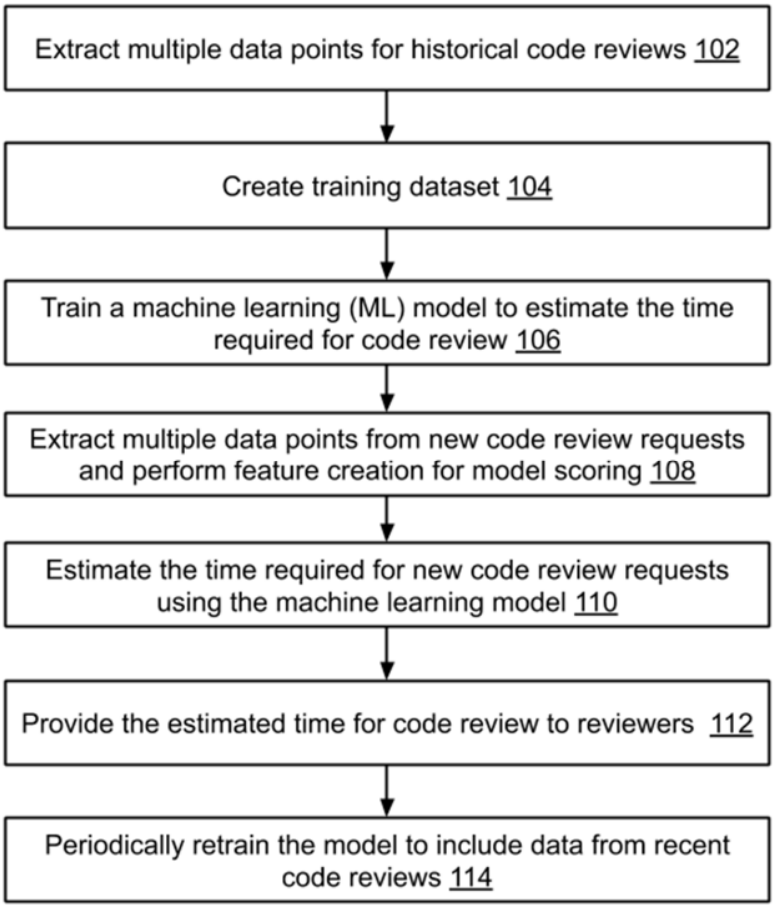

Fig. 1: Example process for estimation of time required for code review

Fig. 1 illustrates an example process flow to train and utilize a machine learning model to generate an estimate of the time required for code review, per techniques described herein. From a historical dataset of code reviews conducted by practitioners, multiple data points are extracted (102) from each code review along with the time taken for the code review. These data points are used to create (104) a training dataset for a machine learning model to estimate the time taken for code review.

A machine learning model is trained (106) using this dataset to estimate the time required for conducting code review. Standard machine learning practices of feature engineering, hyperparameter tuning, out of sample validation, and model performance evaluation are followed

to select a suitable model that can be used in production to estimate the time required for conducting new code reviews.

Multiple data points are extracted (108) from new code review requests to perform feature creation for model scoring. An estimate of the time required to conduct code review is obtained (110) by applying the model on the data record created from the new code review request. The time estimate can be displayed to the code reviewers (112) along with the code review requests via interfaces such as the code review UI, as part of a browser extension (or other application) that notifies the reviewers of incoming reviews, etc. The time estimate can also be made available as a data point that can be used by other systems.

Widespread adoption of the described techniques can lead to efficient prioritization of code reviews, and consequently a reduction in time taken for code review, thus contributing to enhancing practitioner productivity. As a result, it is possible that the output of the model and the consequent change in code review activities affect the very metric that the model is trained to predict. To account for this, the model can be retrained periodically (114), e.g., daily/weekly, retrained (114), by updating the training dataset with features generated from recent code reviews.

CONCLUSION

This disclosure describes the use of machine learning techniques to generate effort estimates of the actual human time required to perform a code review. The machine learning model is trained on a dataset that is generated based on historical code modifications and corresponding reviews. The target variable for the machine learning model is to predict the actual time taken for code review by a practitioner. Multiple data points from each historical

code review are used to obtain features that are used to train the machine learning model. The model is periodically retrained to ensure accuracy of predictions.