**Does your pull-request have what it takes to get accepted?**

Pull-based development method [1] has revolutionized software development practices. Integration of this new way of code contribution [2], code review [3] and process automation [4] has allowed developers to work collaboratively from different parts of the world. GitHub has emerged as the most popular project hosting, mirroring and collaboration platform for this matter over the past couple of years [1].

In this project, we would dive deep into an extensive dataset [5] of 3 million pull request records. The dataset in question has 120 metrics related to each pull-request[[1]](#footnote-1). The metrics are further separated into 3 categories- contributor characteristic, project characteristic and pull request characteristic. Besides the technical report[[2]](#footnote-2) and Table 1 [5], the MySQL table structure[[3]](#footnote-3) has to be heavily referenced throughout the timeline of the project as they provide definitions and other valuable information about the metrics.

Our goals from this project are to establish a data science pipeline which would establish if a particular pull-request is more likely to be accepted. The current plan for this is to use a Gradient Boost algorithm, i.e., XGBoost. Furthermore, if our model predicts the rejection of a pull-request, we deploy the SHapley Additive ExPlanation or SHAP to establish the bad actors for that particular pull-request. Finally, we use BEAM search, a heuristic search algorithm, to find the closest best alternative of the bad actor to increase the acceptance rate of the pull-request.

However, since there are 120 different metrics, we start the project by focusing on the metrics that holds the most weight. We refer to Table 1 [5] for this matter. Moreover, processing 3 million data would be exhaustive and thus the decision to start small-scale has been made. The initial processing would have around 30,000 pull-request that could be considered. If time permits, we move on and deploy our pipeline on the entire dataset.

Although, there does not exist a metric that states if a pull-request is successful or not, there exists other metrics which gives us ample information to make an educated guess about the acceptance likelihood. Nevertheless, the determining factors are yet to be established on this one and can only be done after pre-processing stage.

The following table provides an insight of the estimated timeline of the stages of the project.

|  |  |  |
| --- | --- | --- |
| **Sprint Dates** | **Stage of Project** | **Status** |
| June 27 – July 8 | EDA | Ongoing |
| June 27 – July 8 | Feature Selection | Ongoing |
| July 4 – July 15 | Building the acceptance likelihood formula | TBD |
| July 4 – July 15 | Selection 30,000 pull-request among >3 million pull-request.  \*\*Must include both accepted and rejected pull-requests | TBD |
| July 18 – July 22 | XGBoost | TBD |
| July 25 – July 29 | SHAP | TBD |
| August 1 – August 12 | BEAM | TBD |
| August 15 – August 29 | Deploying the pipeline on the entire project | - |
| July 4 – August 29 | Iteratively write the dissertation report. | TBD |

***References:***

[1] Georgios Gousios, Martin Pinzger, and Arie van Deursen. 2014. An Exploratory Study of the Pull-Based Software Development Model. In Proceedings of the 36th International Conference on Software Engineering (Hyderabad, India) (ICSE 2014). Association for Computing Machinery, New York, NY, USA, 345–355. <https://doi.org/10.1145/2568225.2568260>

[2] Jiaxin Zhu, Minghui Zhou, and Audris Mockus. 2016. Effectiveness of code contribution: From patch-based to pull-request-based tools. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering. 871–882

[3] Y. Yu, H. Wang, G. Yin, and C. X. Ling. 2014. Who Should Review this PullRequest: Reviewer Recommendation to Expedite Crowd Collaboration. In 2014 21st Asia-Pacific Software Engineering Conference, Vol. 1. 335–342. https://doi. org/10.1109/APSEC.2014.57

[4] Bogdan Vasilescu, Yue Yu, Huaimin Wang, Premkumar Devanbu, and Vladimir Filkov. 2015. Quality and Productivity Outcomes Relating to Continuous Integration in GitHub. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (Bergamo, Italy) (ESEC/FSE 2015). Association for Computing Machinery, New York, NY, USA, 805–816. <https://doi.org/10.1145/2786805.2786850>

[5] Xunhui Zhang, Ayushi Rastogi, and Yue Yu. 2020. On the Shoulders of Giants: A New Dataset for Pull-based Development Research. Proceedings of the 17th International Conference on Mining Software Repositories. Association for Computing Machinery, New York, NY, USA, 543–547. https://doi.org/10.1145/3379597.3387489

1. Request to use the restricted dataset was made and granted by the authors. [↑](#footnote-ref-1)
2. [new\_pullreq\_msr2020/technical\_report.pdf at master · zhangxunhui/new\_pullreq\_msr2020 · GitHub](https://github.com/zhangxunhui/new_pullreq_msr2020/blob/master/technical_report.pdf) [↑](#footnote-ref-2)
3. [new\_pullreq\_msr2020/table\_structure.pdf at master · zhangxunhui/new\_pullreq\_msr2020 · GitHub](https://github.com/zhangxunhui/new_pullreq_msr2020/blob/master/table_structure.pdf) [↑](#footnote-ref-3)