Acquiring Software Engineering Skills by Contributing Advanced Open Source Projects

Aidar Shakerimov

Department of Computer Science, School of Engineering and Digital Sciences,
Nazarbayev University, Nur-Sultan, Kazakhstan
aidar.shakerimov@nu.edu.kz

Abstract. Open Source Software (OSS) projects provide a huge opportunity for beginner computer scientists to learn skills by contributing them and interacting with other contributors. Although the types of contributed projects and how they can be contributed may differ from each other, a detailed example can give a comprehensive understanding of the process of learning through contribution. This paper describes an experience of a computer science student contributing to a solid OSS project as a code developer. Although the contributions done by the student were not added to the code base of the project, the student discusses his project and contribution selection process, his interactions with the core developers, the challenges and pitfalls encountered, as well as his overall impressions about the process. The narrative comes on behalf of the student.

Keywords: Open source software, Code development, Student experience

1 Motivation

I am a 4th year CS student with a strong interest in the research and development of Artificial Intelligence (AI) and Machine Learning (ML). Academic researchers depend on software sharing [1]. For example, the most popular research tools for developing Artificial Intelligence and Machine Learning are Open Sources, and it is important to understand how to use and contribute those tools efficiently. In this report, I describe my first experience of contributing to an OSS project as a code developer. It was important for me to focus on the developer role to acquire practical skills in building scientific programs. The OSS project that is contributed by me in this report is called scikit-learn (see the link in the Appendix), and it is one of the most frequently used frameworks in ML studies. It contains a variety of ML algorithm implementations as well as tools for working with data sets and calculus associated with the algorithms. In addition to that, it is an active project with a high number of contributors, an active discussion forum with a sufficient frequency of bug reports and fixes responses.

I coordinated the selection of the project and its further contribution with the book [1]. It provides comprehensive information about organisation and man-

agement models, licenses, motivations of the OSS projects, as well as the contribution roles, so that I would become involved into the project as efficiently as possible.

2 Description of the Contributed Project

Scikit-learn is an Open Source development tool specialized for statistical analysis with data mining, predictive modeling, and machine learning. The goal of this project is to provide the research community with solid state-of-art implementations of predictive analysis techniques and to be supported and developed by the research community itself. In addition to that, the project aims to remain as much user-friendly and available for understanding and becoming involved in it in terms of simple, clear, and comprehensive documentation and other study materials[2].

3 General Aspects

The source code of the project is shared under the BSD license which is a non-copyleft license. This, unlike copyleft licenses, allows a licensee to use the source code in commercial aims [book]. In addition to the Github repository, it has a mailing list. The project was initiated by David Cournapeau in 2007, but from 2010 the community of the project has a federal leadership model a group of leaders. The current governance model structure consists of the technical committee in the center who is responsible for keeping smooth development of the project [book], the team of 21 core contributors who also have the power of important decisions, and 5 triage managers who are responsible for tasks allocation. The decision-making process in the project is done in the 'Consensus-seeking' democratic model. The project tasks are managed by labeling issues with category of an issue (bug, new feature, etc.), status (taken/help needed), structural location (name of the module incorporated), and skill requirements (easy, moderate, hard), etc.

According to the Muffatto, 2006 [1], the open participation principle, allows the project (such as scikit-learn) high flexibility and large exploration of new features or more efficient alternatives to the old features. However, theoretically, this can lead to a lack of incentives to participate. It seems that the project overcomes these risks through frequent releases (usually once in a half of a year). As a result, the pulse of the project demonstrates that the number of active contributors is sufficiently high (see Fig 1.).

4 Technical Aspects

The existing architecture is built around the uniform API and 4 main classes of objects: Estimator (fits a model based on training data), Predictor (a trained

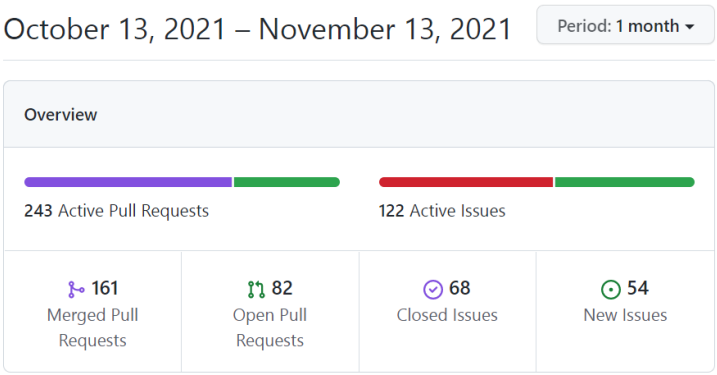

Fig. 1. Pulse of the project for the recent month

model to make predictions), Transformer (to modify the data), and Model (evaluates the trained model). The functionality is represented with more than 50 modules that implement different statistical and machine learning models in terms of the four classes described above. Currently, the project is under the development of a new release. The overall development goals of the project are described in a special document. It lists new or upgraded functionalities that need to be done by core developers.

5 Role and Work Done

I entered the scikit-learn community as a code developer. My main interest includes fixing bugs. I briefly analyzed several recent available bug issues in terms of the level of difficulty, clarity of requirements, and being theoretically understandable. All of the issues that I was working with were associated with the Gaussian Process (GP) Module which implements a supervised learning algorithm for numerical prediction and probabilistic classification. To understand the code of this module I spent some time studying the theoretical concepts associated with the GP: theoretical background and formulas were learned from Rasmussen and Williams, 2003 [3]. The main idea of this algorithm is to predict the function without iterative adjusting parameters of function but, instead, it calculates the probability distribution of all functions that can be applied for the prediction.

The module consists of two independent classes (subclasses of the Estimator class): Gaussian Process Regression and Gaussian Process Classification. Both two classes exploit the third class called 'Kernels'. The subclasses of 'Kernel' class represent different kernels that may be used for covariance estimation in the GP.

5.1 Fixing the bug of division by zero in instances of Gaussian Process using Matern kernel where smoothness coefficient is equal to 0.5 (the issue 19021)

Firstly, I assigned myself to solve the bug issue 19021 "Improving the error message on the Gaussian Process Regression" (see the link in the appendix). Although it was assigned to me it was closed before my pull request due to parallel development. The details of my work are given below.

The class which is bugged is the Matern Kernel class. This is one of the subclasses of the 'Kernel' class. First of all, it was important to understand the logic lying behind the given kernel. Kernels are functions that compute covariance matrix for a given data. The Matern kernel is a subtype of RBF kernel, a highly generalized kernel that is usually used when it is unclear what kernel to use. The distinction of Matern from RBF is that it has a parameter that controls the degree of generalization (smoothness of learned line) [3].

After becoming familiar with the logic of the algorithm, I spent some time reading the source code of the issued module. The problem is located in the formula where the Matern kernel function is expected to behave like an absolute exponential kernel when given a smoothness parameter equal to 0.5. In one of the array divisions, it signals the Runtime error caused by invalid division (NumPy divide command). As one of the triage members suggested, the solution can be found in the NumPy library documentation for the issued command.

I have read the documentation and found out that the error is caused when division by zero is being performed. This led to the conclusion that the array in the denominator might contain zeros.

My final implementation was to replace all zeros with a very small number so that the division is possible (see the link to the commit in the appendix). However, when I was testing the code I discovered that the problem with division has disappeared. This was caused by a pull request that was merged during the period between the opening of the issue and my commit. Although I did not merge any commits to the main branch of the project, I helped the core members to investigate the fact of parallel development and the issue was closed with my help.

5.2 Fixing the invalid broadcasting of normalization undo in predictions of Gaussian Process Regression with multi-target data (issues 17394 and 18065)

I found out that two issues associated with the Gaussian Process module (see links in the appendix) were describing the same bug in the code. The pull request (see link in the appendix) was done by me, but I was asked to make some improvements.

Conceptually speaking, the module did not allow to make predictions with normalized multi-class outputs. A typical program that runs this process should firstly initialize the Gaussian Process Regression model with normalization flag turned on, then fit the model with training input data and training multi-class

targets (outputs), and finally predict the multi-class mean outputs, standard deviation, and covariance matrix for testing data. The problem was that the code of Gaussian Process Regression was not able to run the predict() function when normalization was initially applied in the fit() function. Inside the predict() function some commands undo the normalization of the obtained covariance and variance from the kernel (lines 355 and 381) by multiplying them with the squared standard deviation of training outputs (the required distribution spread). If the normalization flag was turned off, then they would just remain the same.

Two points were causing the bug. The first is that the code was trying to multiply covariance or variance, which are horizontal vectors (covariance matrix is a horizontal 2d vector), with standard deviation vector, which is also horizontal vectors. The results of such multiplication were expected to be outer products, but they returned a broadcasting error because the outer products require the multiplication of perpendicular vectors. I solved this by reshaping the variance vector and rows of covariance matrix into vertical form before multiplication with the standard deviation vector. As a result, we obtain new dimensionality of outputs: (n samples, n output dims) and (n samples, n samples, n output dims) for std and covariance respectively. The outputs now represent independent results for each class as if they were computed separately but combined in one matrix. This solution was motivated by Rasmussen and Williams [3] which states: "A simple approach is to model each output variable as independent from the others and treat them separately.". When targets are single values the undo normalization is performed as usual.

The solution that I was proposing is considered to work only when output classes are independent of each other. From the context of the issues, I understood that they require the independence of classes. However, as was written in the Rasmussen and William's book (p.190) [3] there might be cases when a user might require dependency between classes, but it is handled using a special methodology. I considered this to be an opportunity for a new functionality issue and sent the suggestion to a discussion thread of one of the issues.

After the pull request was done, there were some linting tests failures. When all checks were passed I was asked to create a new test that checks the correctness of normalization. However, after I applied the test to a toy dataset, it appeared that when normalization is applied in advance, the code gives divergent results when the normalization flag is set on and off. The important point here is that this code does not work for both multi-class and single-class targets. This implies that the normalization process in the GP module was probably implemented wrongly. Since the initially considered issue is focused on multi-class output problems, the new problem increases the scope of my contribution but requires more time for solving.

Although the tests described above displayed some new issues with the code, I still needed to revise my code and I discovered that the fit() function also functioned wrong. The moment that was needed to be improved is that if normalization was turned off, the outputs had a different format from the case when

normalization is set on. That is because the standard deviation here was given as a scalar rather than an array. In fit() function, I handle this by conditioning the dimension of the standard deviation vector on the dimension of the training targets. However, when I edited my code, one of the original tests - the multi-output test treated data with 2d outputs incorrectly: the covariance matrix and standard deviation array was treated to have -1 dimensionality.

Finally, after resubmitting my pull request a changelog test remained. I was consulted that passing this test is dependent on the decision of a core contributor on whether to add the pull request into a new release. However, the core contributor decided to temporally reject this pull request due to the incapability to check its correctness.

5.3 Description of the Interaction with the Community

After reading contribution guidelines, I searched available issues by filtering the labels in the issues list using "Bug" and "help wanted" labels, which gave me a sublist of all available bug issues. All issues that I was working on were assigned to me when I commented on an issue post with the keyword 'take' which is an automatic keyword.

I helped to close the bug issue 19021 "Improving the error message on the Gaussian Process Regression", where I consulted one of the triage members about the parallel development of this issue.

I made a pull request to issues 17395 and 18065. I tried to communicate with the authors of the issues mentioned above to get some theoretical consultation but did not receive replies. On the other hand, I obtained a good practical consultation from the core contributors regarding making pull requests.

6 Discussion and Conclusion

Overall, two bugs contained in one module were assigned to me. None of my fixes were merged to the main branch. The first bug was fixed by another contributor before I created a pull request. Leaning on the literature related to the module I managed to propose a fix of the second bug, but my pull request was not merged because it did not pass all tests.

I consider the main problem that prevented successful fixes of the bugs in the absence of sufficient theoretical knowledge about the module that I was contributing. Gaussian Process module implements a narrow field in Machine Learning and involves a deep understanding of it. One of the consequences is that I was spending dramatic amounts of time investigating the problem. For example, I was late to propose a solution to the first assigned bug. Another consequence is that I was not sure of the correctness of my proposed solutions. Although I found one source to get familiar with the topic and proposed my solution following it, I was not able to identify the reason for some tests failures. Moreover, I still have not received a response to the theoretical questions in the discussion threads, which would help me clarify the correctness of my fix. From

these cases, I can suggest that contributing such sophisticated scientific projects as scikit-learn without deep knowledge of the related concepts is not what a beginner like me can afford.

Another significant obstacle that I experienced during my contribution is the relatively high complexity of the contribution procedures in scikit-learn. Since it is a large project with a huge number of active contributors, the organization of contributions is complex. The policy of the project forces a contributor to complete several sophisticated procedures to propose an update. Although this allows authors to manage contributions easily, it was difficult for me to follow all instructions. I suppose that this happened because I had small experience in using the Github platform and I did not understand the meaning of most of the procedures. For example, during my pull request, it was unclear for me how to solve work with linting and changelog tests until the core contributors consulted me.

As a small critic of the authors of the scikit-learn, I would like to mention the lack of comments in lines of code which also increases the hardness of contributions. Although the header documentation explains the overall logic of the code, the details of implementation are not explained so well. This provides some difficulties in understanding between different contributors and contributions may cancel each other. Muffatto, 2006 explains this as a bad consequence of parallel development. It is suggested that the code developers should increase the modularity of the code to avoid the high complexity of the code[1].

Despite encountered problems, significant positive outcomes of my participation are the OSS observations that I made from working with scikit-learn. Firstly, I discovered how well can a project be organized and managed. I consider the high number of contributors and frequent releases as an advantage of the 'Consensus-seeking' democratic model and pleasant attitude of the triage managers and core developers towards common contributors. However, I noticed some limitations which scikit-learn project has. First of all, the number of triage managers is quite small, which leads to the appearance of parallel development cases as happened with my first assigned bug. Secondly, since my theoretical questions clarifying the correctness of my fix have not been answered, I conclude that there were not many core members which had sufficient theoretic knowledge of the implemented algorithms. In my opinion, the absence of correspondent specialist control may decrease the reliability of this project.

To sum up, although the scikit-learn community is a solid project with a well-organized community and friendly attitude towards new contributors, I would not suggest it for a first contribution experience. This is, generally, because of the theoretical background significance, the lack of commentaries in the code, and the complexity of procedures needed for completing a contribution.

7 Appendix

Link to the scikit-learn project: https://github.com/scikit-learn/scikit-learn

Link to the issue 19021: https://github.com/scikit-learn/scikit-learn/issues/19021

Link to the issue 17394: https://github.com/scikit-learn/scikit-learn/issues/17394

Link to the issue 18065: https://github.com/scikit-learn/scikit-learn/issues/18065

Link to the commit fixing the issue 19021: https://github.com/AidarShakerimoff/scikit-learn/commit/e56cace45470471ec9ee20a246663fe8ee1002b2

Link to the pull request fixing issues 18065 and 17394: https://github.com/scikit-learn/scikit-learn/pull/19706

References

1. Muffatto, Moreno. Open source: A multidisciplinary approach. Vol. 10. World Scientific, 2006.
2. Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.
3. Rasmussen, Carl Edward. "Gaussian processes in machine learning." In Summer school on machine learning, pp. 63-71. Springer, Berlin, Heidelberg, 2003.