Open Source Community Analysis and Contribution - SKlearn
CSCI - 5576, HPSC
Fall '19
Vachan Daffedar Aswathanarayana

Introduction:

SciKit-Learn (Sklearn) is a free open source python library that provides implementations of various machine learning models, algorithms and statistical models. The Sklearn package is built on top of SciPy, NumPy and, maplotlib. The Sklearn package is available to be used under the BSD license. It was designed with the notion of interoperability between the various SciPy modules. The package has a thriving international community of contributors and developers. The community follows a 3 monthly release cycle.

History:

Scikit-Learn started off as scikits.learn project as a spawn of the Google Summer of Code by David Cournapeau in 2007. The basename of the package is indicative of its homage to the SciPy package. In 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel belonging to the INRIA (French Institute for Research in Computer Science and Automation) took over the project. The first official public release of the project happened on Feb 1st, 2010.

Organization and Governance:

The project currently recognizes 19 developers as Core-Contributors (authors) and around 30 people as Core-Developers. The project follows a meritocratic, consensus-based approach for governance and decision-making. Anyone with an interest in the project can join the community, contribute to the project design and participate in the decision-making process. The community is organized into different roles based on their involvement in the project.

- Contributor: Contributors are community members who contribute to the project in terms of bug reports, documentation or code changes. Anyone can become a contributor as long as they follow the guidelines.
- Core Developers: Core Developers are community members who have shown dedicated involvement in the project over a period of time. Being a core developer allows contributors to more easily carry on with their project-related activities by giving them direct access to the project's repository and is represented as being an organization member on the SkLearn GitHub organization. Core developers are expected to review code contributions, can merge approved pull requests, can cast votes for and against merging a pull-request and can be involved in deciding major changes to the API.
- Technical Committee: The Technical Committee (TC) members are core developers who have additional responsibilities to ensure the smooth running of the project. TC members are expected to participate in strategic planning, and approve changes to the governance model.

Decisions about the future of the project are made through discussion with all members of the community. All non-sensitive project management discussion takes place on the project contributors' mailing list and the issue tracker. Occasionally, sensitive discussion occurs on a private list. Any decision taken is done by taking a consensus from all the community members by taking votes, which is open for a set period of time. Any major API change proposal needs to accompanied by a SLEP (SkLearn Enhancement Proposal).

Funding:

All the funding to the community is handled by NumFocus, a non-profit organization. The received donations for the Sklearn project mostly will go towards covering travel-expenses for code sprints, as well as towards the organization budget of the project. Apart from this, INRIA actively funds the project and helps in organizing coding sprints and other events. It also has funded 3 core-contributors to work full-time between 2010-2017. Other institutes like Paris-Saclay Center for Data Science, NYU Moore-Sloan Data Science Environment, Columbia University, the University of Sydney, etc have funded other developers to work fulltime on the project. Google has sponsored 7 developers over the years through Google Summer of Code program.

Goals and Achievements:

Scikit-learn remains very popular in practice for trying out canonical machine learning techniques, particularly for applications in experimental science and in data science. The current goals of Sklearn are three-fold:

- Continue maintaining a high standard of quality and well-documented toolset.
- Increase the ease of onboarding for the developers for external components.
- Improve interoperability between current fleet of data science tools like Pandas, etc.

The achievements of the Sklearn community include getting branded as a "well-maintained and popular" package for machine learning in the book - *SciPy and NumPy: an overview for developers*, O'Reilly, in November 2012. The package has also received public testimonials from a number of tech giants including J.P. Morgan, Spotify, Evernote, Booking.com, etc for being a state-of-the-art machine learning package and for ease of usage. The package has over 500k downloads via anaconda. The Sklearn paper published about the package in 2011 has close to 1.5k citations.

Contribution:

I started looking over the current open issues of the project to pick a good issue for contributing with the intention of finding an issue which is the realm of current capabilities. I skimmed over a few issues marked as "good first issues" and shortlisted a few issues like:

- 15127 - "GBDT support custom validation set", which is about making the changes to the Gradient Boosting classifier and regressor to take in a custom validation set.
- 15409 - "Add dtype parameter to KBinsDiscretizer", which is about maintaining and enforcing the data type of the input across the KBinsDiscretizer in order to prevent mismatches later on.

I am currently working on the above issues and will try to add a few more issue contributions.