The scikit-learn package and community

Claire Savard
CSCI 5576
Fall 2019

Introduction and community overview:

Scikit-learn is an open source package in python that provides machine learning algorithms along with data mining and pre-processing techniques (figure 1). This package is built on SciPy with close connection to both NumPy and matplotlib. It was initially developed by David Cournapeau during a Google Summer of Code internship and was first released online in 2010. Since then, the scikit-learn community has grown tremendously to over 1000 contributors. It is a very active community with more than 25 issues posted on the master GitHub branch per week and 19 core contributors that revise and approve pull requests daily. There are detailed guidelines on the scikit-learn webpage that describe how people can contribute to the project, facilitating the contribution process so that is easy for people of all levels. Similarly, the issues posted on the GitHub page range from simple fixes in documentation, to ideas for new algorithm features, allowing everyone to contribute in whichever way they please.

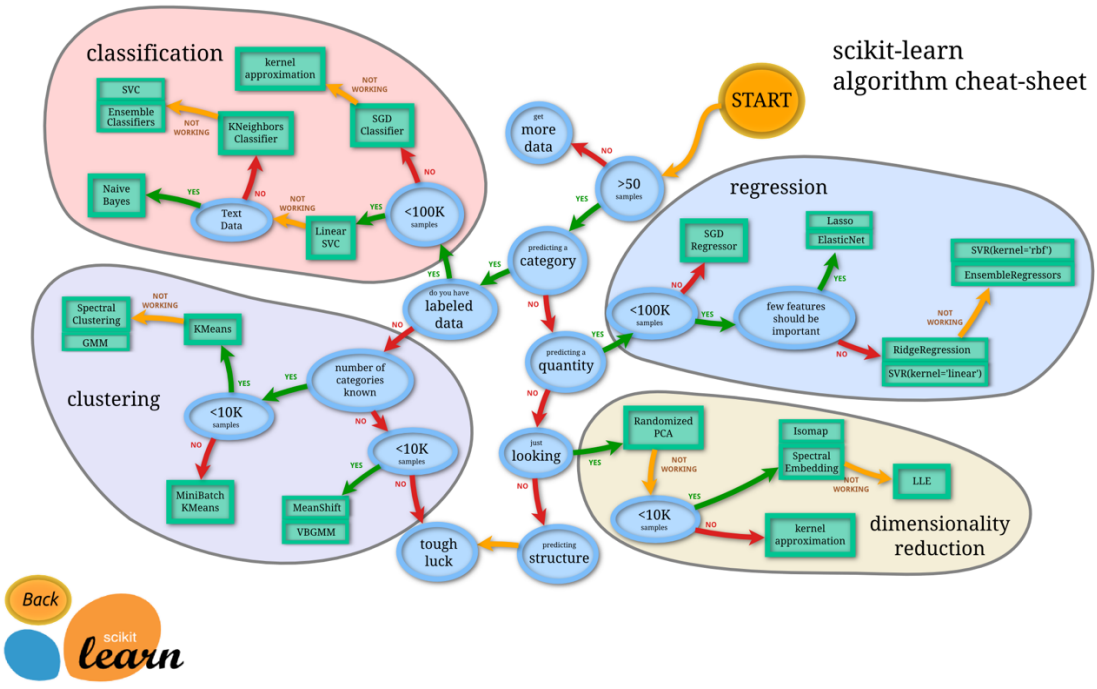

Figure 1: A look at what is offered in the scikit-learn package and when someone may want to use a certain class or function from the package.

The use of scikit-learn:

Scikit-learn is used widely across the globe by various companies, institutions, and private groups. JP Morgan, Spotify, and Booking.com are 3 examples of companies that use the scikit-learn regularly. This package is well respected as a machine learning tool due to the state-

of-art models that it provides which run efficiently and effectively. It emphasizes the use of NumPy and SciPy for vectorization, instead of creating for loops to avoid wasting time with the python interpreter. It also makes use of Cython to wrap around C/C++ files when more efficiency may be needed. In general, the community is cautious to ensure that code is fully optimized before releasing it into the public, ensuring that the performance users are getting from the scikit-learn algorithms are the best that they can be.

The scikit-learn package is not only efficient, but flexible and extremely easy to use. It was built so that anyone can implement machine learning algorithms within a few minutes, given their data. Inputs and outputs of most functions are in a NumPy array format, which is also a common package for managing data with well-developed mathematical functions. The package has a website that is easy to read through and gives in-depth detail on all of the classes available for use. This documentation includes what each class does, a detailed list of attributes for the class, an overview of each function call that can be made with the class, and an example of how the class can be used. For more experienced coders, it is also simple to enter the source code and make any changes you wish to the original algorithms.

Current focuses:

The community has many goals for the future of scikit-learn. Most resources are currently being put towards two major tasks: improving the ease of users to develop and publish external components and improving inter-operability with modern data science tools infrastructure such as Pandas and Dask. Specific examples of how these two tasks are being conquered can be found here. These examples include supporting pipelining and pre-trained models better and improving the handling of missing data and categorical features.

This past month, the community has been busy working on a new release of the package. For this reason, the core contributors have been more active on larger issues that they wish to include in the upcoming release. Most of these issues deal with an addition to current classes. There has been a lot of talk over how new additions may break the current API and should, therefore, wait until the updated API. Thus, it seems that a large task for the future will be the API update.

My community contribution:

When I first started using this package in my high-energy physics research, I was frustrated when some classes that I wished to use did not have example code on how to use them in the documentation. Therefore, I decided that I would find the classes which lacked examples and add them into the documentation. After starting this project, however, I realized that all of the classes without examples on the webpage were currently being created for the next release. I managed to get an example merge, but had trouble finding another that was not already being worked on. Thus, I then started to work on the documentation in another way. I realized that the default values for all attributes in certain classes were either inconsistent with the code or were not indicated on the webpage. I am now working through a couple classes to ensure that the default values are documented correctly. Both of my pull requests working on this issue for different classes have now been merged, and I plan to work on one or two more by the end of the semester.