**Case Study on Scikit-learn**

Scikit-learn is an open-source machine learning application. It has simple and efficient tools for data mining and data analysis. It’s easy accessibility and reusability in various contexts makes Scikit a popular tool to use. It is distributed under the 3-clause BSD license.

**Advantages of Scikit-learn**

* **Free Platform**

Because scikit-learn is released with a BSD license, it can be used for free by everyone. This license has minimal restrictions; therefore, users can utilize it to design their applications and platforms with little worry over limitations.

* **Industrial Use**

Scikit-learn is a helpful platform that can predict consumer behavior, identify abusive actions in the cloud, create neuroimages, and more. It is being used extensively by commercial and research organizations around the world, a testament to its ease of use and overall advantage.

* **Collaborative Library**

Scikit-learn began as a one-man mission but now it is being built by numerous authors from INRIA spearheaded by Fabian Pedregosa and individual contributors who are not attached to teams or organizations. This makes the module a well-updated one, releasing updates several times a year. Users can also look forward to assistance from an international community, in case they have queries or if they hit snags in development using the module.

* **Ease of Use**

Commercial entities and research organizations alike have employed scikit-learn in their processes. They all agree that the module is easy-to-use, thereby allowing them to perform a multitude of processes with nary a problem.

* **API Documentation**

Scikit-learn ensures that users old and new alike get the assistance they need in integrating the machine learning module into their own platforms. That is why a documentation detailing the use of its API exists that users can access anytime on the website. This makes certain developers can implement machine learning algorithms offered by the tool seamlessly.

* **Technology & Platform used for development**

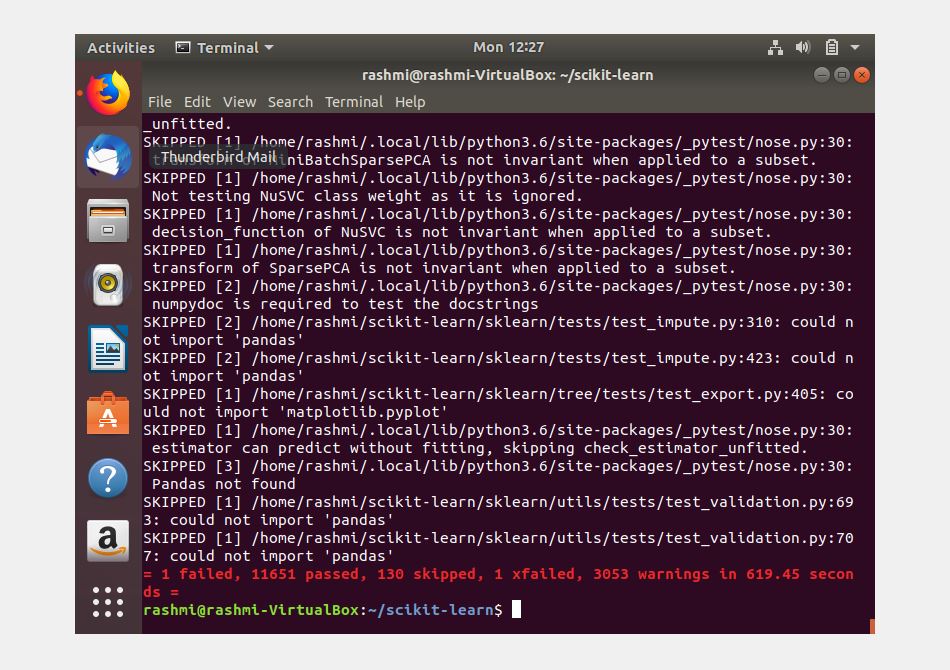
Scikit-learn is built in Python. The latest version of scikit requires Python 3.5 or above. The choice of Python is appropriate due to its rich math and scientific libraries like Numpy, Scipy and matplotlib. Scikit uses Numpy 1.11.0 or above and Scipy 0.17.0 or above.

**Build System used**

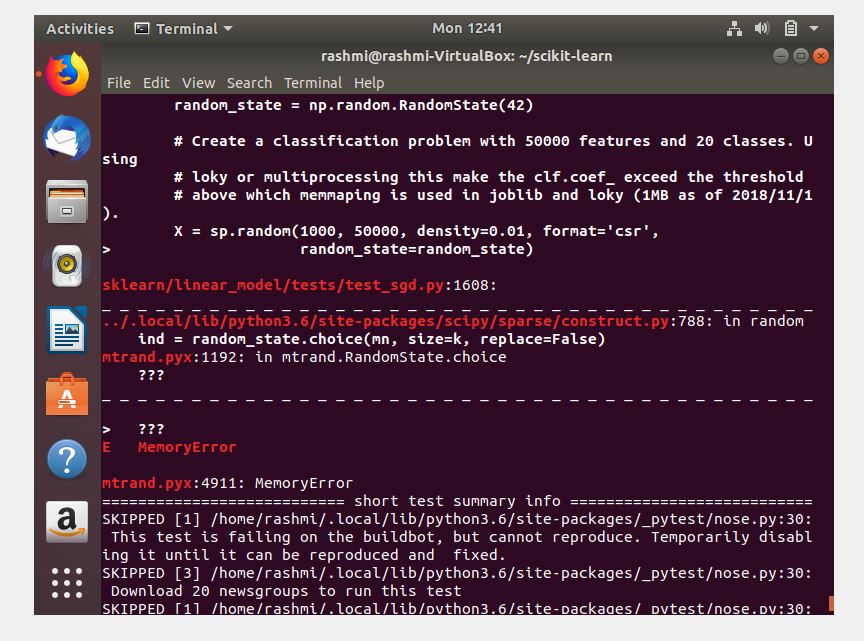
It uses cmake. Building Scikit-learn requires Cython 0.23 or above.

**Tests**

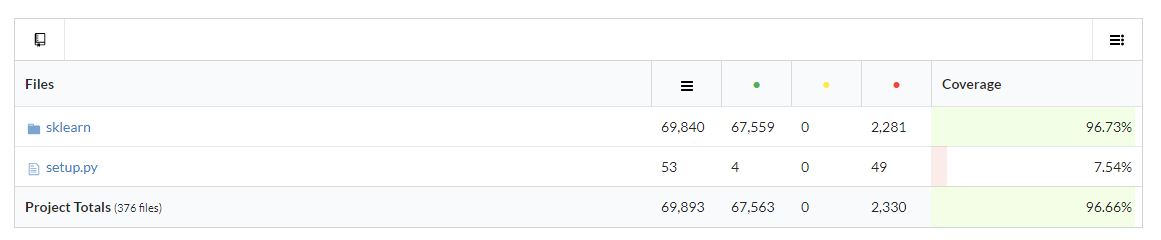
Running tests requires Pytest 3.3.0 or above. Some tests also require pandas. They have Travis CI platform integrated in their github. They test windows, linux and MAC computing platforms. Screenshot of test results is as follows:

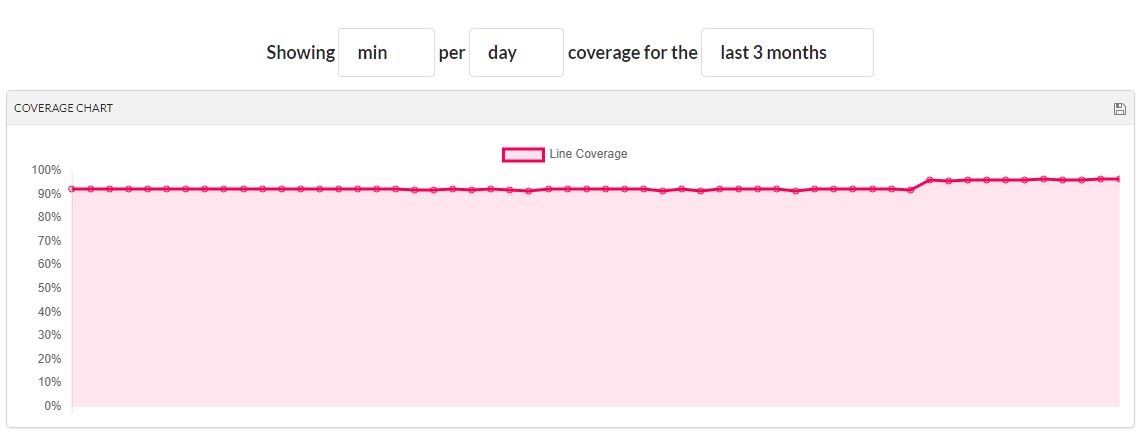


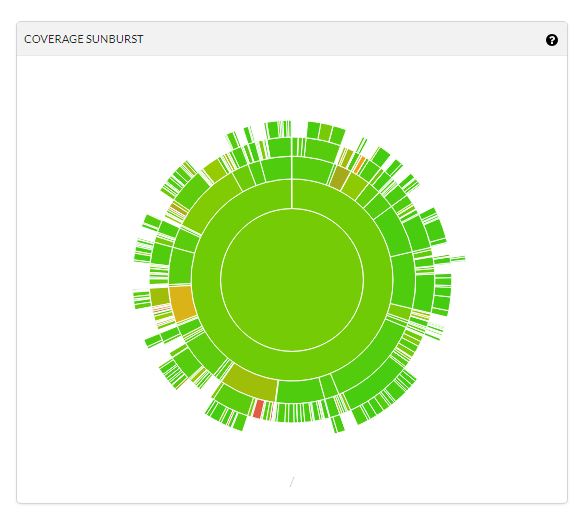
Investigating further into the failed test case, I found this:



Code coverage gives an analytical idea on how well the code or an application has been tested. Tests might return positive results across the board, but if they cover only 30-40% of the code, it’s really hard to gain confidence about the application. Hence, higher the value of code coverage, the better is the end product. Higher code coverage value indicates a more thorough testing of the application and less show-stopper bugs. Their github also has code coverage metrics. Screen shots of the code coverage are as follows:







**Software Architecture**

Scikit is basically an objected oriented implementation of various functional components. Six major functional components include classification, clustering, regression, dimensionality reduction, model selection and preprocessing. A high-level system diagram is as follows:



Scikit-learn has a simple, coherent API built around Estimator objects. All the six functional components are implemented as six basic classes with base functions. Various algorithms are implemented for each of these functional components. Scikit-learn provides documentation for the use of its API if you want to integrate your app. No other integration information is provided. 4 tools on stackshare integrate with Scikit i.e. Keras, Xcessiv and Comet.ml

**Defects**

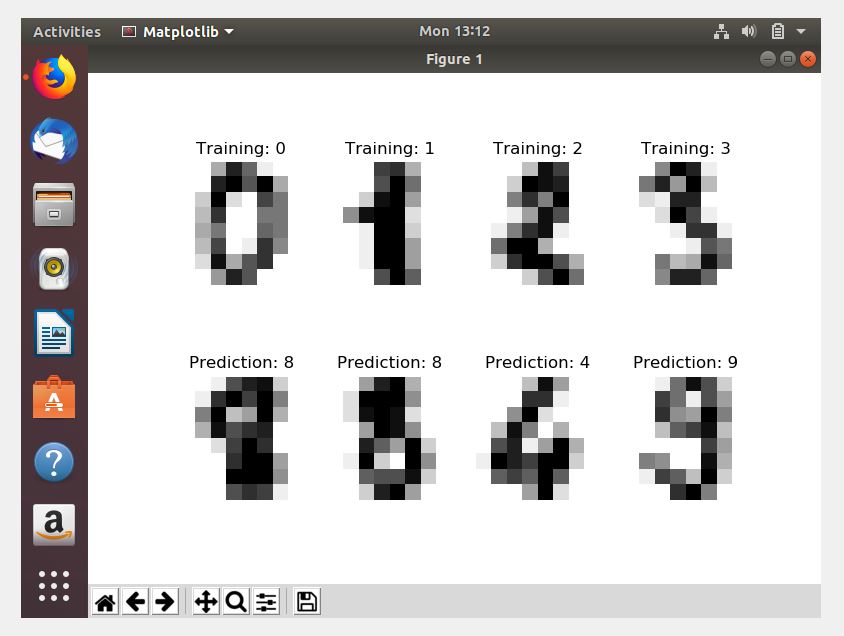
Scikit-learn recommends opening an issue in the issue tracker. This enables contributors to get some feedback from core developers. Advantage of this approach is that it avoids work duplication and avoids adding non-trivial features to the existing system. They also suggest going through the issue list or the pull request list before opening an issue.

Most of the issues in Scikit are related to feature update or adding a new functionality. I couldn’t spot an issue which requires an architecture change.

**Demonstration:**

After code installation and setup, I build the project and was able to run examples. Two sample demonstrations are as follow:

1. Classification example (to predict mnist numbers correctly):



1. Clustering example (K-means clustering with n=6):

