Community management in scikit-learn

Loïc Estève

Outline

- scikit-learn project overview
- scikit-learn community aspects
- challenges ahead

scikit-learn overview

Vision: an enabler

- Machine learning for everyone
- High quality pythonic library
- Community-driven development

scikit-learn strengths

Easy to use:

```
from sklearn import svm
classifier = svm.SVC()
classifier.fit(X_train, y_train)
y_test = classifier.predict(X_test)
```

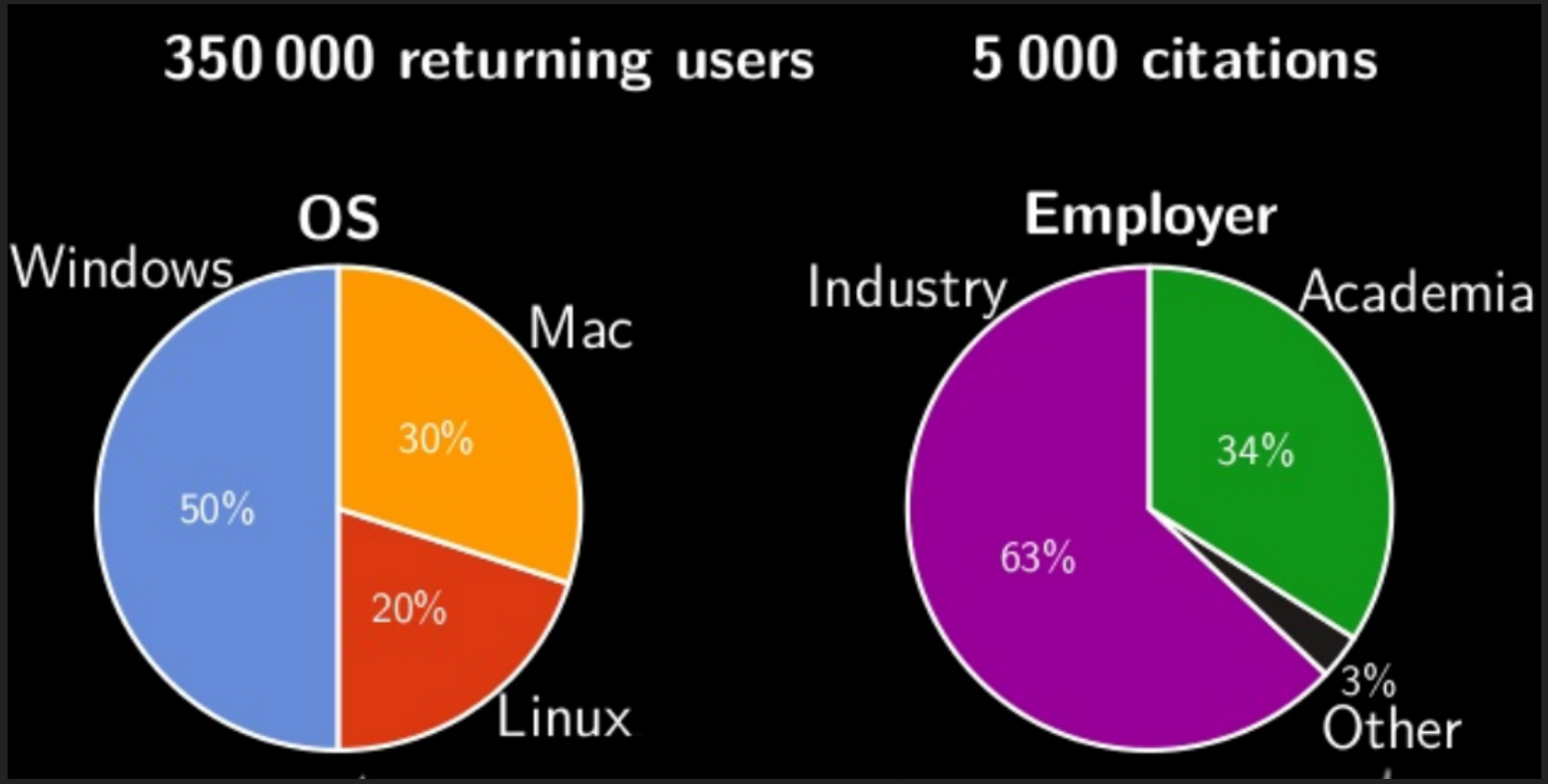
Consistent API for estimators

Optimised for speed: Numpy and Cython

Great documentation

Examples gallery (sphinx-gallery)

Users


350 000 returning users
OS
Windows
Mac
Linux
50%
30%
20%
5 000 citations
Employer
Industry
Academia
Other
63%
34%
3%

Users (industry)

Virality and readers engagement

Fraud detection

Personalized radios

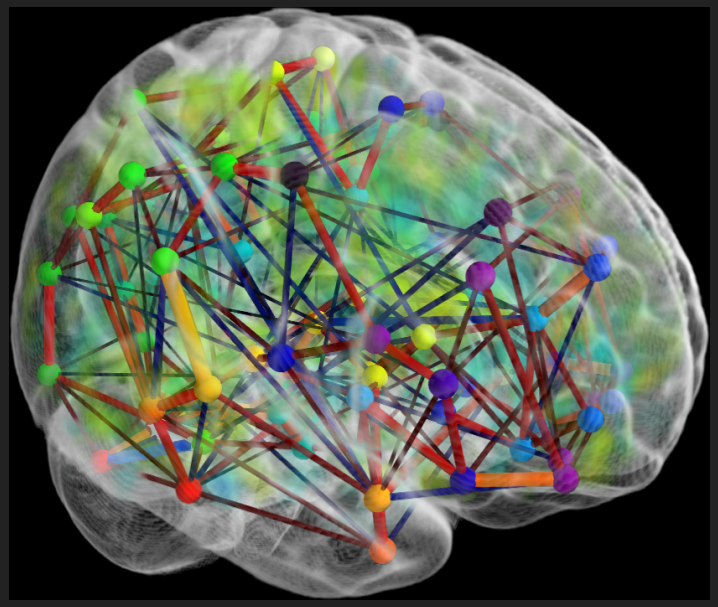
Inventory forecasting & trends detection

Predictive maintenance

Personality matching

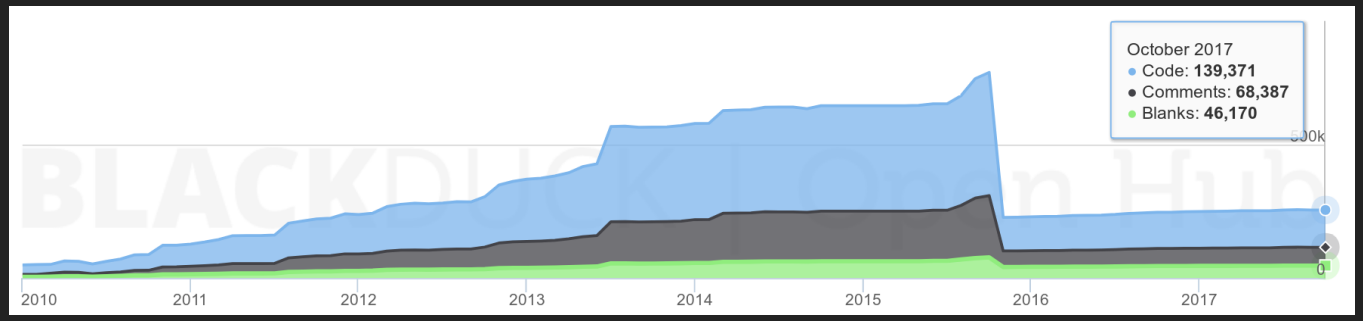
Users (academia)

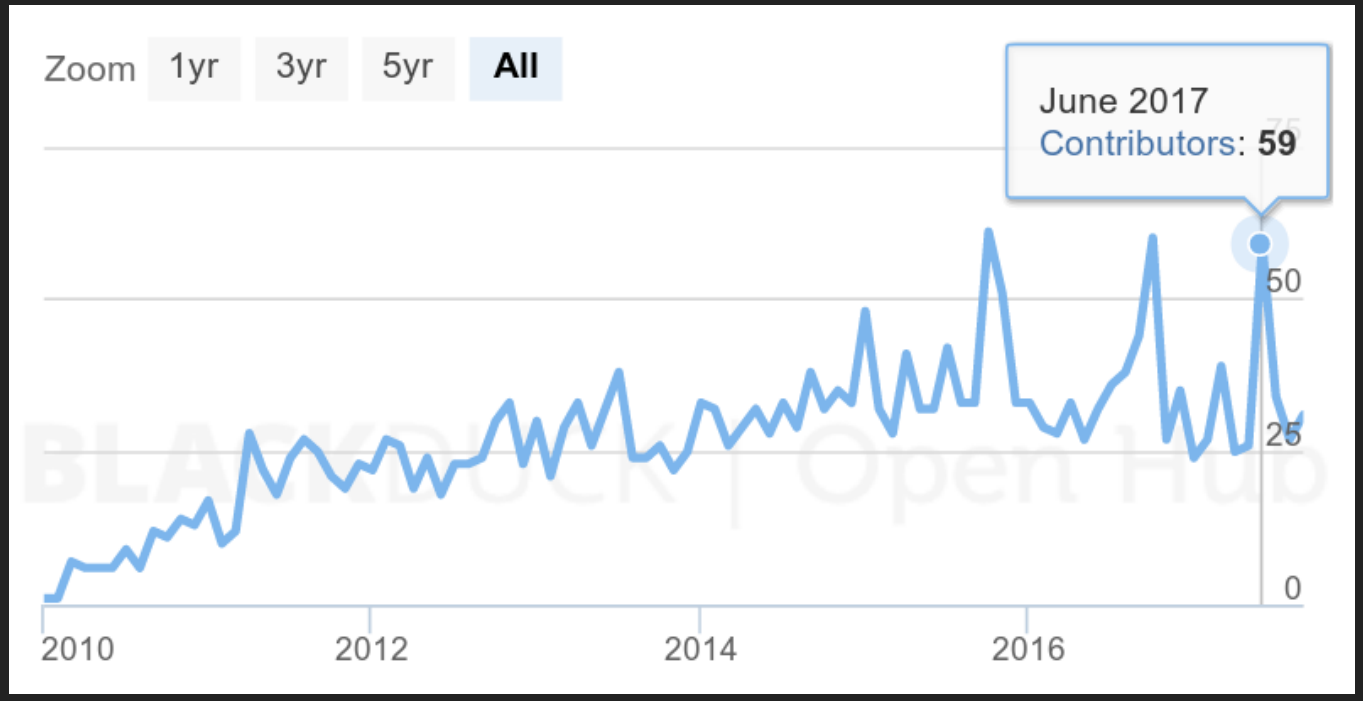
In Parietal: decode activity of the brain recorded via fMRI



Used widely in: astronomy, particle physics, genomics, etc ...

Code and contributors





https://www.openhub.net/p/scikit-learn

Community aspects

Communication

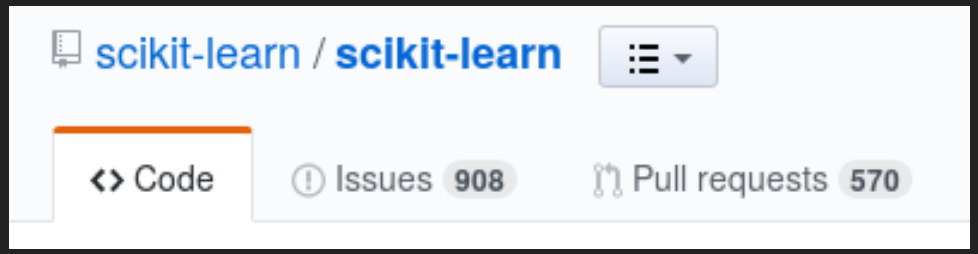
- github for bug report, feature requests, questions
- StackOverflow is recommended for usage questions but hard to enforce
- mailing list, mostly usage questions. Recently: internal mailing list to discuss priorities
- twitter: mostly releases and new features merged
- gitter: dev (mostly active to prepare releases) and main channel

Project management

- "good first issue" + "Easy" tags
- sprint (Scipy, EuroScipy, yearly one-week sprint)
- scikit-learn talks/tutorials in Scipy, EuroScipy, PyCon conferences
- Google Summer of Code: advantages and downsides
- mission d'école doctorale. 10% time of a PhD

Project activity

~50 notifications per day from comments on issues/PRs



October 2, 2017 – November 2, 2017

Period: **1 month**

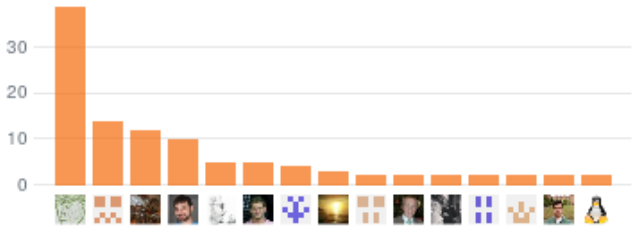
Overview

108 Active Pull Requests

130 Active Issues

80 Merged Pull Requests	**28** Proposed Pull Requests	**94** Closed Issues	**36** New Issues

Excluding merges, **61 authors** have pushed **89 commits** to master and **165 commits** to all branches. On master, **122 files** have changed and there have been **1,910 additions** and **971 deletions**.



Tools to manage the project

- CIs on each Pull Request
- generate documentation on each PR with CircleCI (userscript to add button to the github website)
- flake8_diff.sh (details matter! maintenance: flake8 bug-fix still to be released)
- github issue/PR template. Significantly improved the standard of issues submitted
- Python scientific ecosystem: dev wheels allows to test on numpy, scipy, cython, pandas dev versions to catch regressions sooner

Things we could do

- Give more rights to more people to close issues/add labels (help on triage, PR review)
- using bots (e.g. danger) ? trade-off to find, may this would create tools with heavy maintenance and higher barrier to entry for new contributors
- discourse forum. Free hosting for open-source, but then position compared to mailing list unclear

scikit-learn contrib (1/2)

http://contrib.scikit-learn.org

Not everything can (and has to) go in scikit-learn

For cutting-edge algorithms, quick development, maturation

Benefits:

- nice template to start the project (testing, CIs, etc ...)
- visibility

scikit-learn contrib (2/2)

So far ~10 projects in scikit-learn-contrib

Quite recent, gouvernance questions to sort out:

- who decides what goes in?
- Some well-defined pre-requisites to be included in scikit-learn-contrib (doc, tests, follow scikit-learn API), but precise inclusion strategy yet to define

Challenges ahead

Staying relevant

Quickly moving ecosystem:

- GBRT: xgboost, LightGBM
- deep-learning: tensorflow, PyTorch
- Spark/PySpark. dask + distributed: young and promising projects by Continuum

Sustainable growth

Reviewing is the bottleneck

User support drowns core devs

Finding contributors. Project maturity can offset the fun of contributing

Funding (tragedy of the commons)