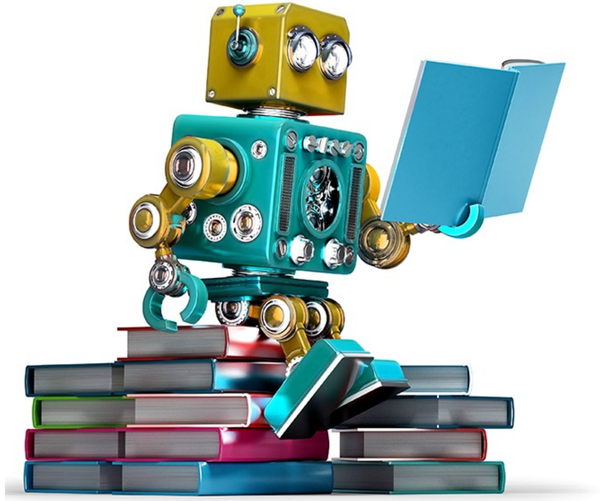
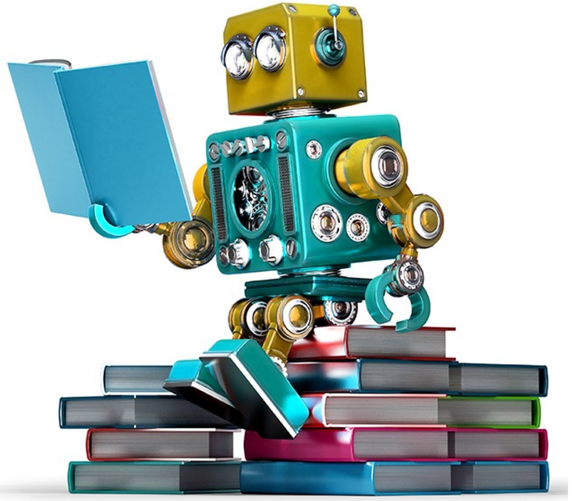
How to build a successful scientific software package: the scikit-learn model



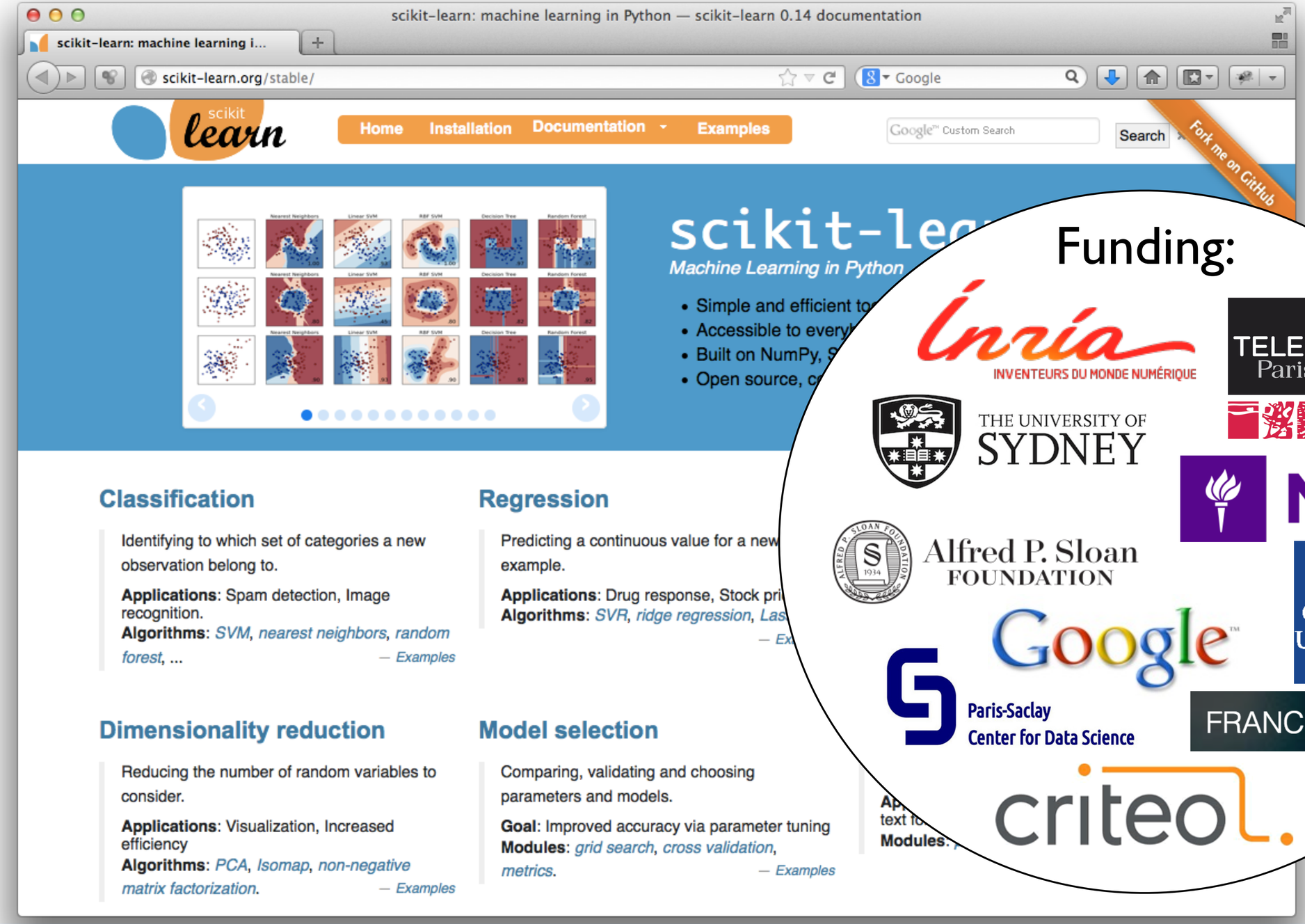


Alexandre Gramfort

http://alexandre.gramfort.net

GitHub : @agramfort

Twitter : @agramfort



http://scikit-learn.org

REPOSITORIES 14,539 | Language: All | Sort: Best match

tensorflow / tensorflow

★ 91.6k

Computation using data flow graphs for scalable machine learning

tensorflow | machine-learning | python | deep-learning | deep-neural-networks

C++ Updated 4 hours ago

keras-team / keras

★ 26.5k

Deep Learning for humans

deep-learning | tensorflow | neural-networks | machine-learning | data-science | python

Python Updated 17 minutes ago 10 issues need help

scikit-learn / scikit-learn

★ 26.3k

scikit-learn: machine learning in Python

machine-learning | python | statistics | data-science | data-analysis

Python Updated an hour ago 192 issues need help

Machine learning

Machine learning is the practice of teaching a computer to learn. The concept uses pattern recognition, as well as other forms of predictive algorithms, to make judgments on incoming data. This field is closely related to artificial intelligence and computational statistics.

Wikipedia

deep-learning | python

neural-network | data-science

3rd most popular machine learning software on GitHub

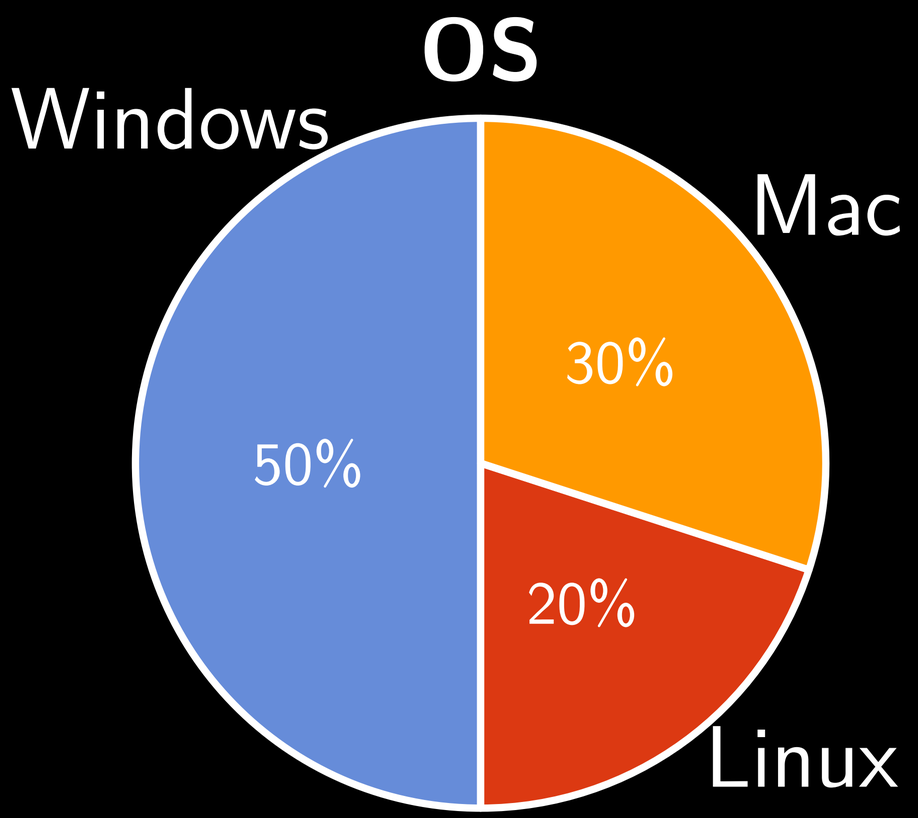
scikit-learn the origin in 2010

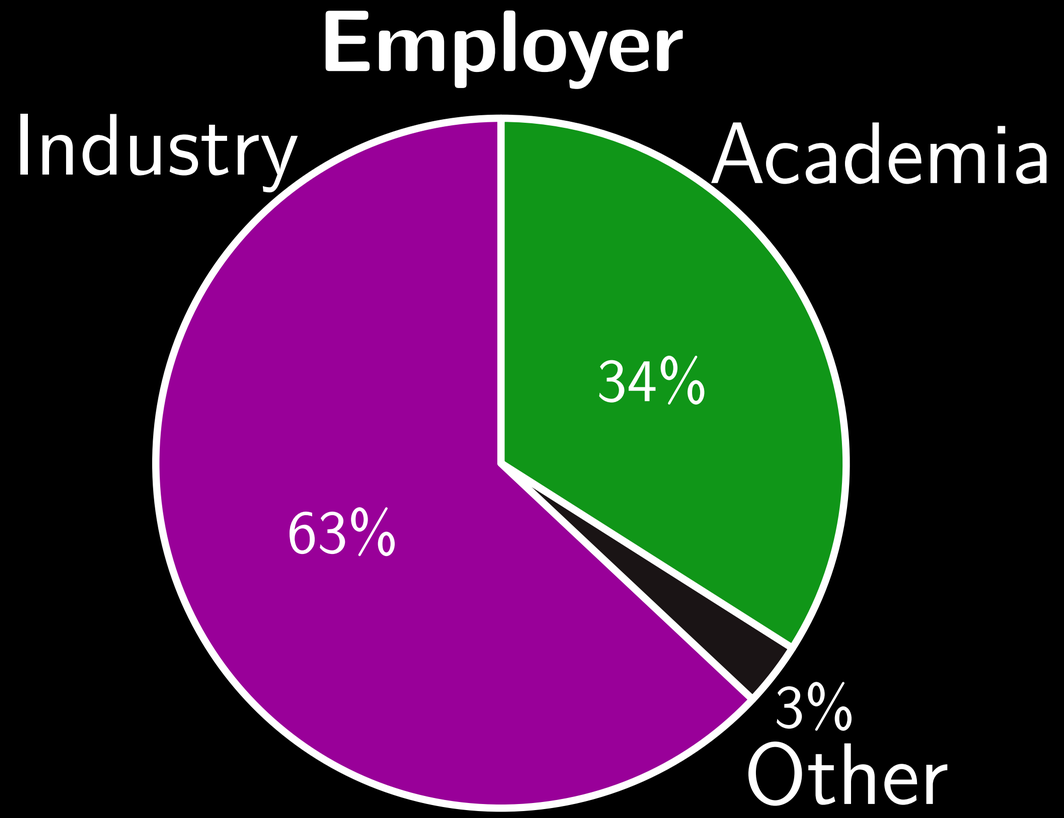
Really started with Inria funding in 2010 in PARIETAL team

Outline

- How successful is actually scikit-learn?
- 10 reasons why it worked
- What's next?

4 million unique visitors in 2017 on scikit-learn.org
Estimation of 500,000 regular users based on web statistics

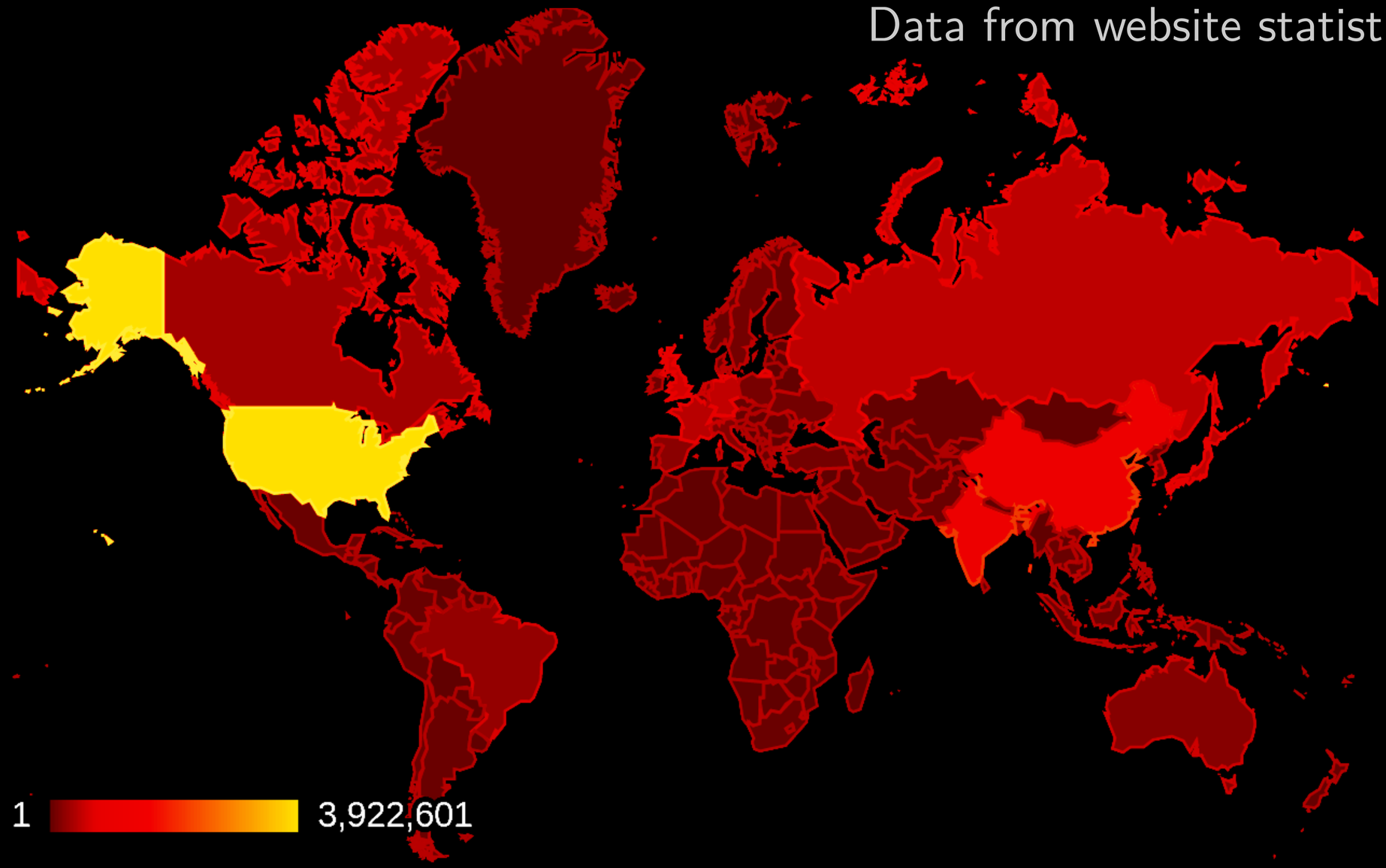




Data from website statistics

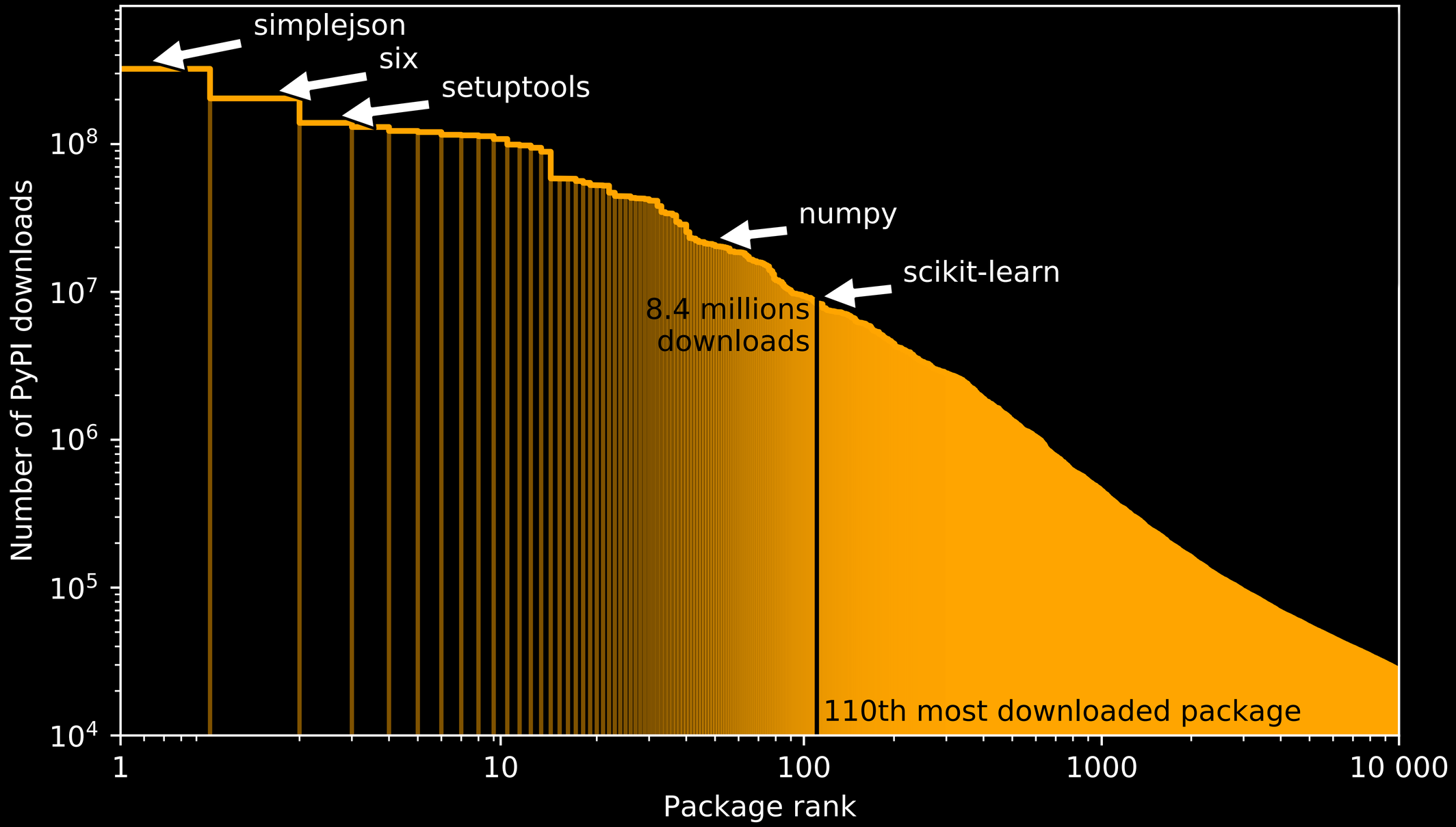
Data from user survey in 2016

Data from website statistics



Global impact:

usage reflects size of tech population per country



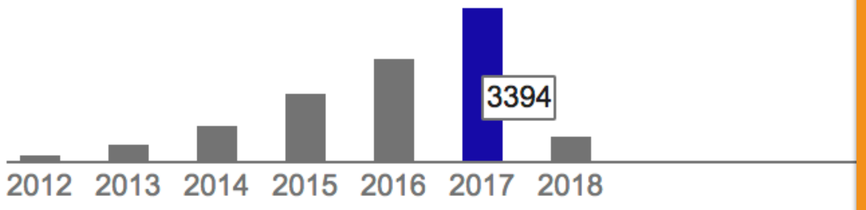
Scikit-learn: a core package supporting the ecosysem

Academic impact

Scikit-learn: Machine learning in Python

[PDF] à partir de jmlr.org

Auteurs	Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay
Date de publication	2011
Revue	Journal of machine learning research
Volume	12
Numéro	Oct
Nombre total de citations	Cité 8914 fois



Cited in ML / Stats papers but massively in applied fields: physics, chemistry, neuroscience etc…


Why did it work?

Reason 0: Simple & consistent API

Import model:
```
>>> from sklearn.xxx import Model
```

Create model:
```
>>> model = Model(param=10)
```

Train model:
```
>>> model.fit(X_train, y_train)
```

Train on batch:
```
>>> model.partial_fit(X_train, y_train)
```

Predict:
```
>>> model.predict(X_test)
```

Transform:
```
>>> model.transform(X_test)
```

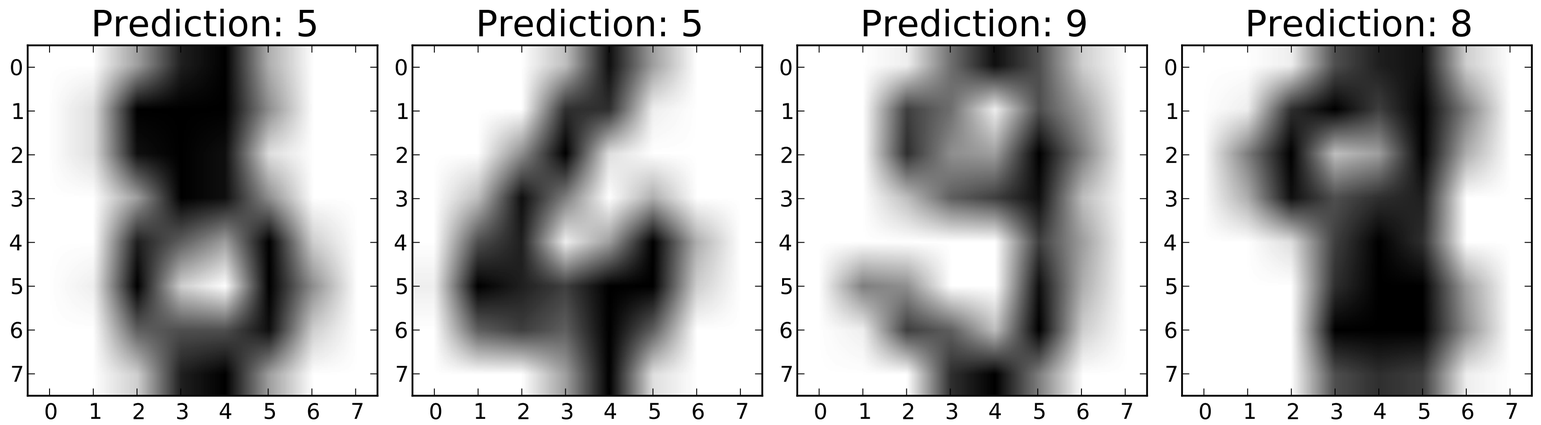
Simple API : Short learning curve !

[API design for machine learning software: experiences from the scikit-learn project
Lars Buitinck , Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake Vanderplas, Arnaud Joly, Brian Holt, Gaël Varoquaux,
https://arxiv.org/abs/1309.0238]

Reason 0: Simple & consistent API

Classification of images of digits in a few lines of code

```
import matplotlib.pyplot as plt
from sklearn import datasets, svm
# Load data
digits = datasets.load_digits()
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))
# Learn ie. fit
classifier = svm.SVC()
classifier.fit(data[:n_samples // 2], digits.target[:n_samples // 2])
# Predict and plot
for index, image in enumerate(digits.images[n_samples // 2:n_samples // 2 + 4]):
    plt.subplot(1, 4, index)
    plt.imshow(image, cmap=plt.cm.gray_r)
    plt.title('Prediction: %i' % classifier.predict(image.ravel()), fontsize=20)
```



Simple API often copied

XGBoost *https://github.com/dmlc/xgboost/blob/master/demo/guide-python/sklearn_examples.py*

```
>>> import xgboost as xgb
>>> xgb_model = xgb.XGBClassifier()
>>> xgb_model.fit(X[train_index], y[train_index])
>>> predictions = xgb_model.predict(X[test_index])
```

Tensorflow + scikit-learn http://terrytangyuan.github.io/2016/03/14/scikit-flow-intro/

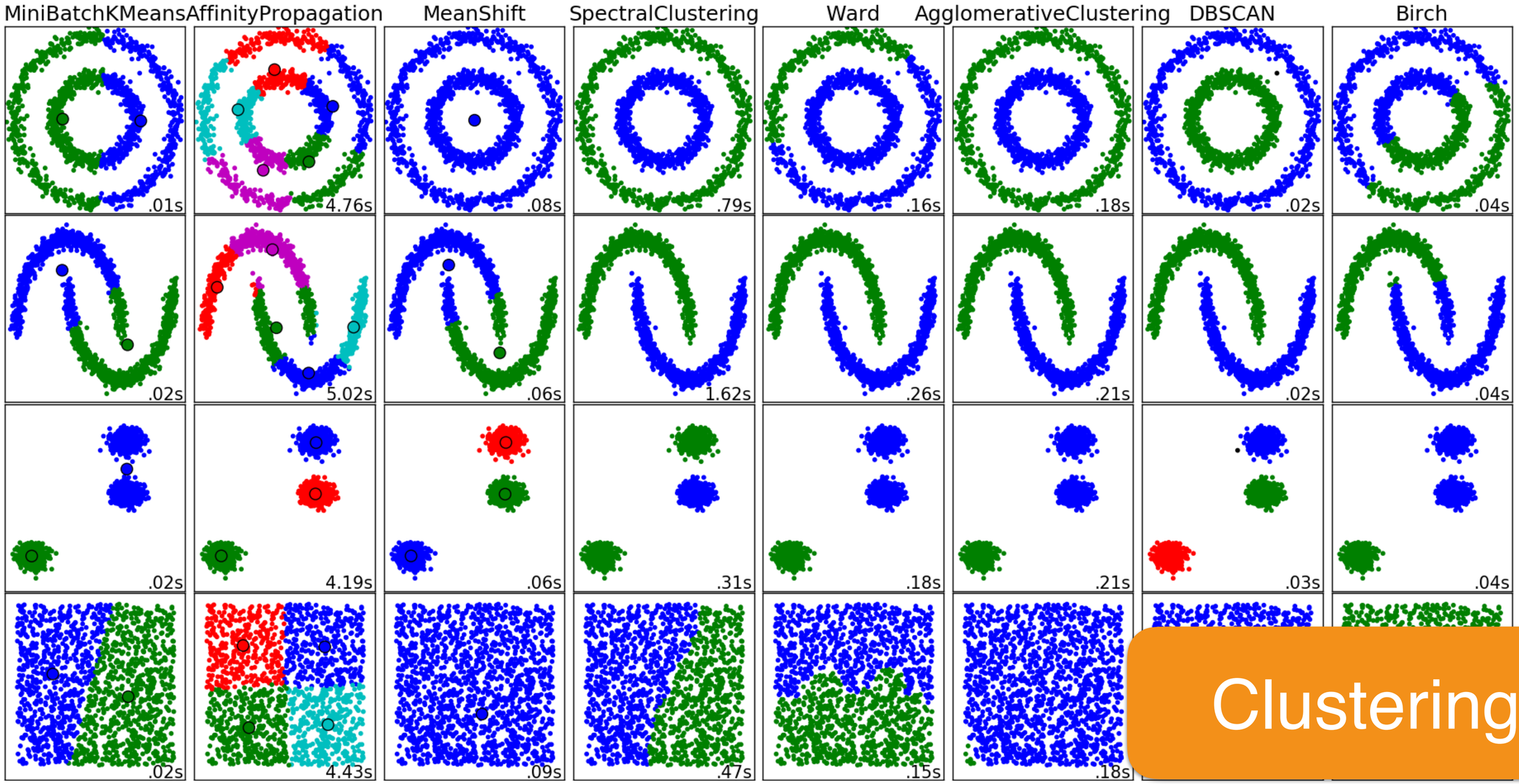
```
>>> import tensorflow.contrib.learn as skflow
>>> from sklearn import metrics
>>> classifier = skflow.TensorFlowDNNClassifier(
      hidden_units=[10, 20, 10], n_classes=3)
>>> classifier.fit(X, y)
>>> score = metrics.accuracy_score(iris.target,
classifier.predict(iris.data))
>>> print("Accuracy: %f" % score)
```

Simple API often copied

Spark *http://spark.apache.org/docs/2.0.0/ml-classification-regression.html#naive-bayes*

```
import org.apache.spark.ml.classification.NaiveBayes
import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator

// Load the data stored in LIBSVM format as a DataFrame.
val data = spark.read.format("libsvm").load("sample_libsvm_data.txt")

// Split the data into training and test sets (30% held out for testing)
val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3), seed=1234L)

// Train a NaiveBayes model.
val model = new NaiveBayes().fit(trainingData)

// Select example rows to display.
val predictions = model.transform(testData)
predictions.show()
```

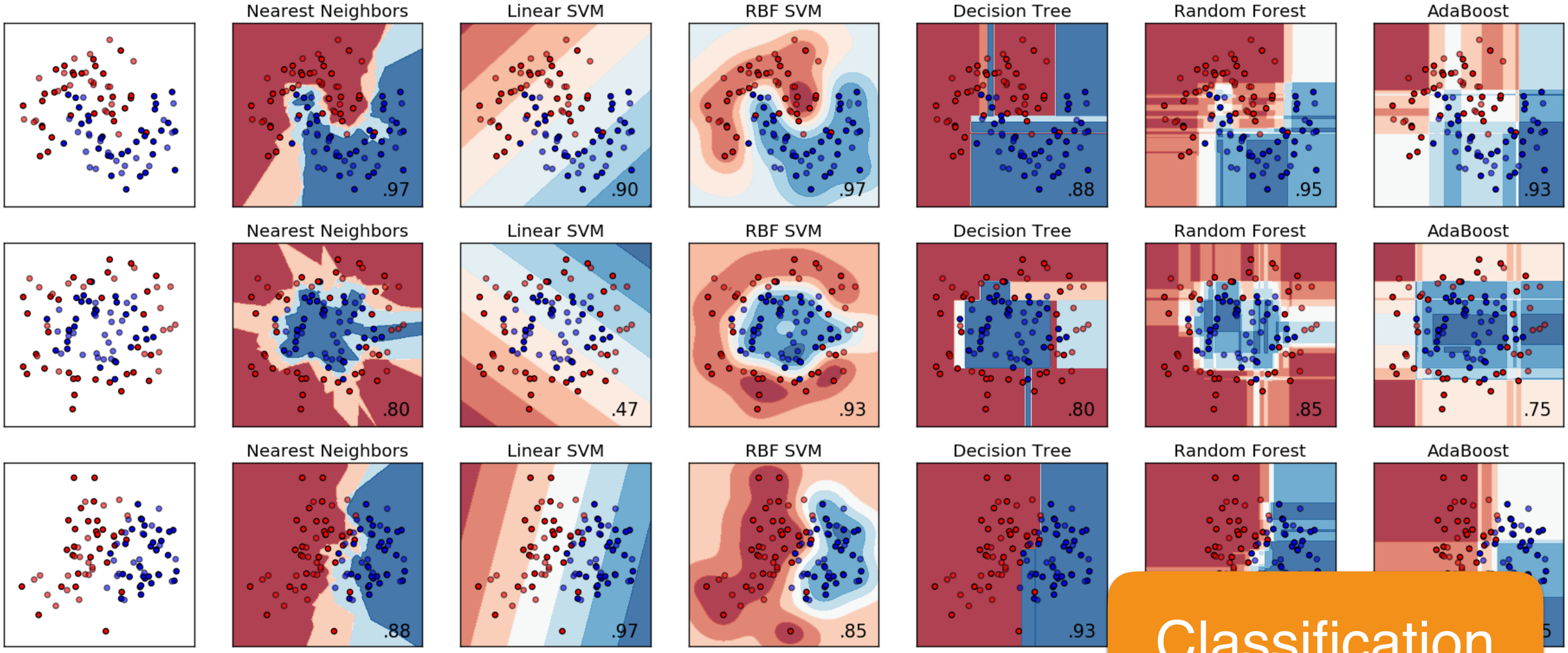
Reason 1: Rich feature set



Clustering

http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

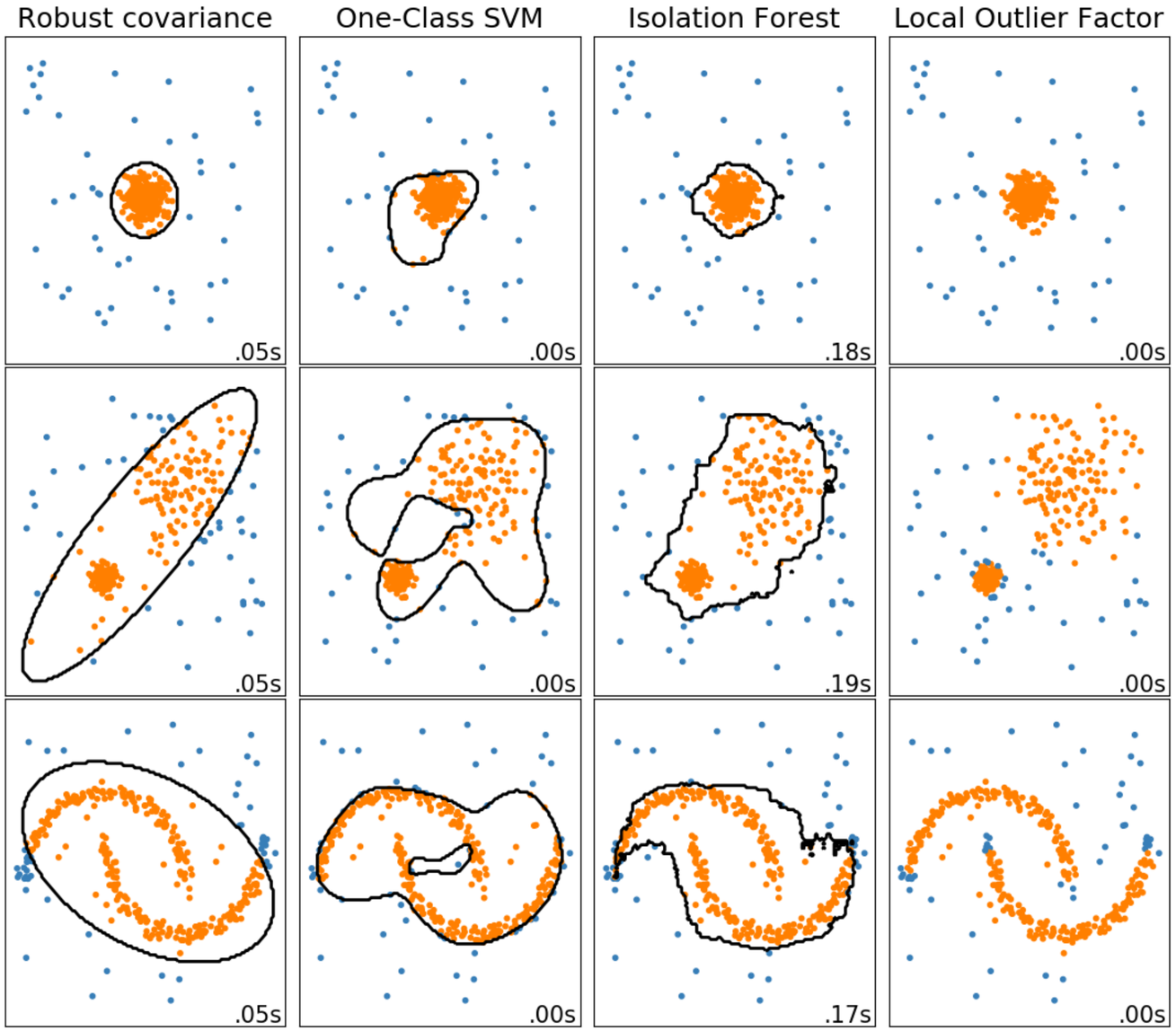
Reason 1: Rich feature set



Classification

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Reason 1: Rich feature set



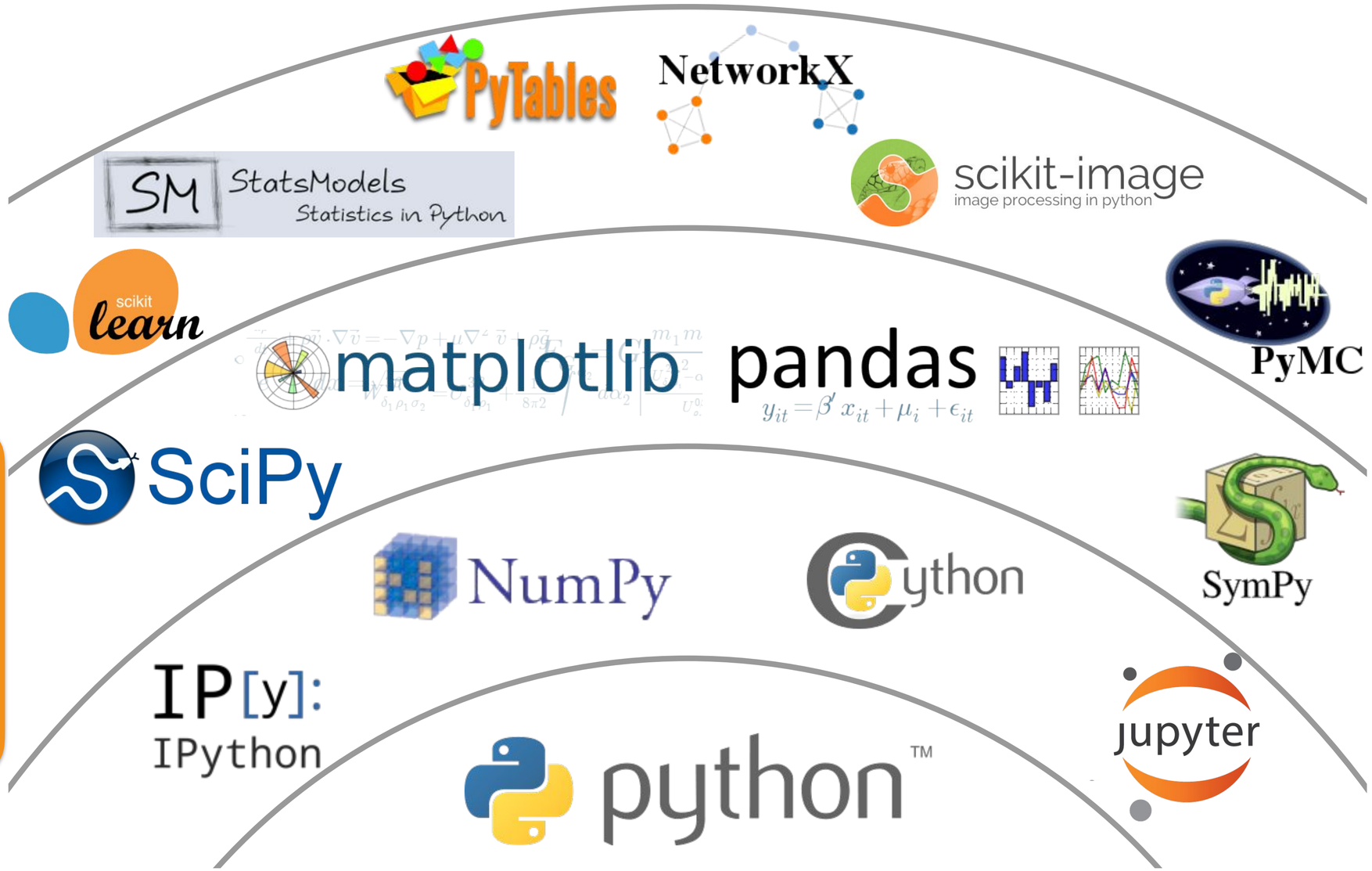
Anomaly / Novelty detection

http://scikit-learn.org/dev/auto_examples/plot_anomaly_comparison.html

Reason 2: Limited feature set / Clear Scope

We provide one component in the Python ecosystem

New models if they have many citations and demonstrate extra empirical benefit



Reason 3: Contributing guidelines

Developer's Guide

Contributing

- Submitting a bug report
- Ways to contribute
- Retrieving the latest code
- ▶ Contributing code
- ▶ Coding guidelines
- Code Review Guidelines
- ▶ APIs of scikit-learn objects
- ▶ Rolling your own estimator

Developers' Tips and Tricks

- ▶ Productivity and sanity-preserving tips
- Debugging memory errors in Cython wi

Utiliti

Make the rules clear

sklearn/cluster/_inertia.pyx

```
...  ...  @@ -21,9 +36,9 @@ def compute_ward_dist(np.ndarray[D
21   36       for i in range(size_max):
22   37           row = coord_row[i]
23   38           col = coord_col[i]
24       -       n = (m_1[row] * m_1[col]) / (m_1[row] + m_1
     39  +       n = 1.0 / (1.0 / m_1[row] + 1.0 / m_1[col])
```

2

agramfort repo collab

i am afraid this is numerically less stable. it is justified by speed?

jmetzen

you are right, I reverted it to the old implementation

Provide feedback with code reviews

Reason 4: Automate review process

All checks have passed Hide all checks

8 successful checks

- ✓ **ci/circleci: python3** — Your tests passed on CircleCI! Details
- ✓ **codecov/patch** — 98.9% of diff hit (target 95.01%) Details
- ✓ **codecov/project** — 95.02% (+<.01%) compared to 2aba6e2 Details
- ✓ **continuous-integration/appveyor/pr** — AppVeyor build succeeded Details
- ✓ **continuous-integration/travis-ci/pr** — The Travis CI build passed Details

This branch has no conflicts with the base branch

Merging can be performed automatically.

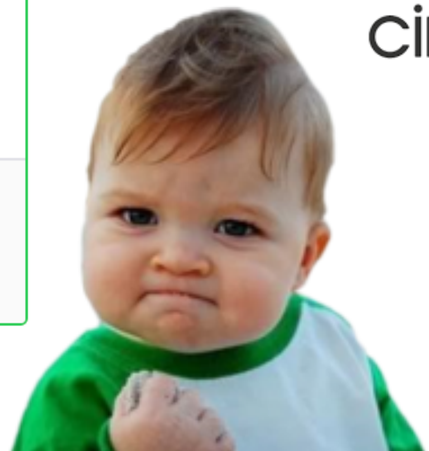
Squash and merge ▾ You can also open this in GitHub Desktop or view command line instructions.

Travis CI

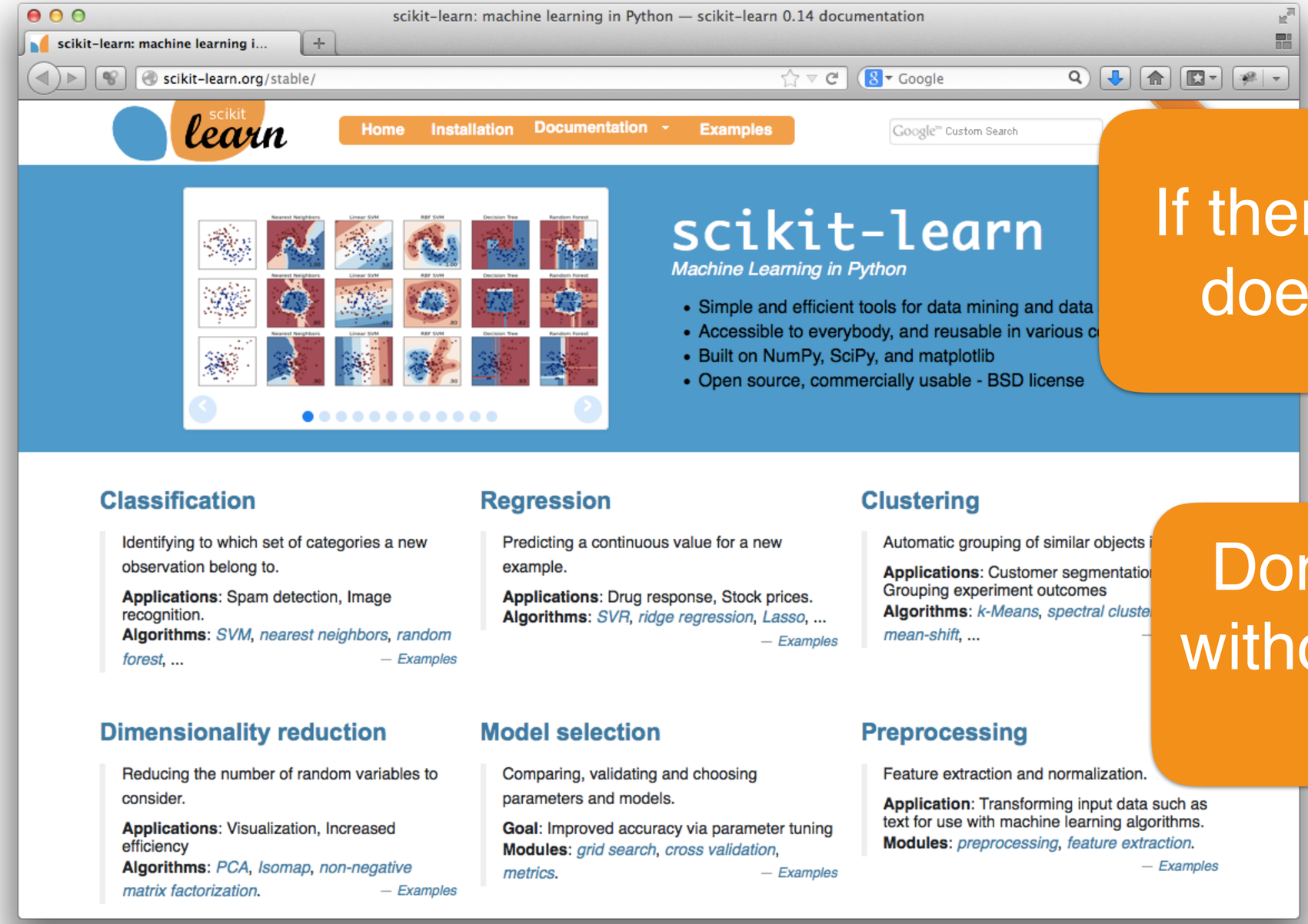
AppVeyor

ANACONDA

circleci



Reason 5: Documentation & Examples



If there is no doc it does not exist !

Don’t accept code without doc and one example

Reason 5: Documentation & Examples

Sphinx-Gallery

latest

Search docs

Getting Started with Sphinx-Gallery
Configuration
Frequently Asked Questions
Sphinx-Gallery Syntax
Sphinx-Gallery Utilities
Sphinx-Gallery API Reference

Gallery of Examples

- General examples
- The sin function
- Examples which don't produce image output

Secondary gallery
Change Log
Fork sphinx-gallery on Github

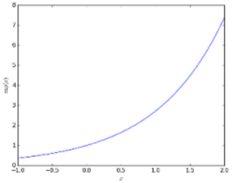
Docs » Gallery of Examples View page source

Gallery of Examples

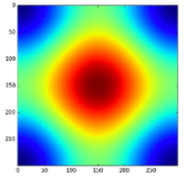
General examples

General-purpose and introductory examples from the sphinx-gallery

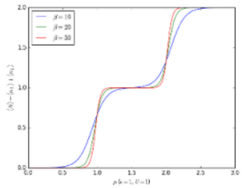
Using sys.argv in examples



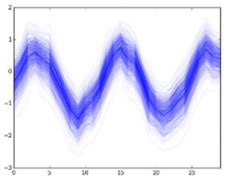
Plotting the exponential function



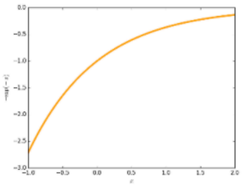
Colormaps alter your perception



Some Quantum Mechanics, filling an



Seaborn example



Choosing the thumbnail figure

SPHINX
Sphinx-Gallery

sphinx-gallery:
Write doc by writing
Python code

Started in scikit-learn
and made possible by
CDS Paris-Saclay

Paris-Saclay
Center for Data Science

Reason 6: Don't own the project

Release history

Version 0.19.1

October, 2017

This is a bug-fix release wi

Note there may be minor d
have equal distance to som

Don't make the package a product of your institution ~~http://scikit-learn.inria.fr~~

Changelog

API changes

- Reverted the addition of `metrics.ndcg_score` and `metrics.dcg_score` which had been merged into version 0.19.0 by error. The implementations were broken and undocumented.
- `return_train_score` which was added to **`model_selection.GridSe`** **`model_selection.RandomizedSearchCV`** and **`model_selection.cro`** its default value from True to False in version 0.21. We found that calcu cross validation runtime in some cases. Users should explicitly set `ret` ing functions are slow, resulting in a deleterious effect on C V runtime, o scores. #9677 by Kumar Ashutosh and Joel Nothman.
- `correlation_models` and `regression_models` from th legacy gaussian processes implementation have been be-

Give credit to people

Reason 7: Open community

Discussions are online not at coffee machine

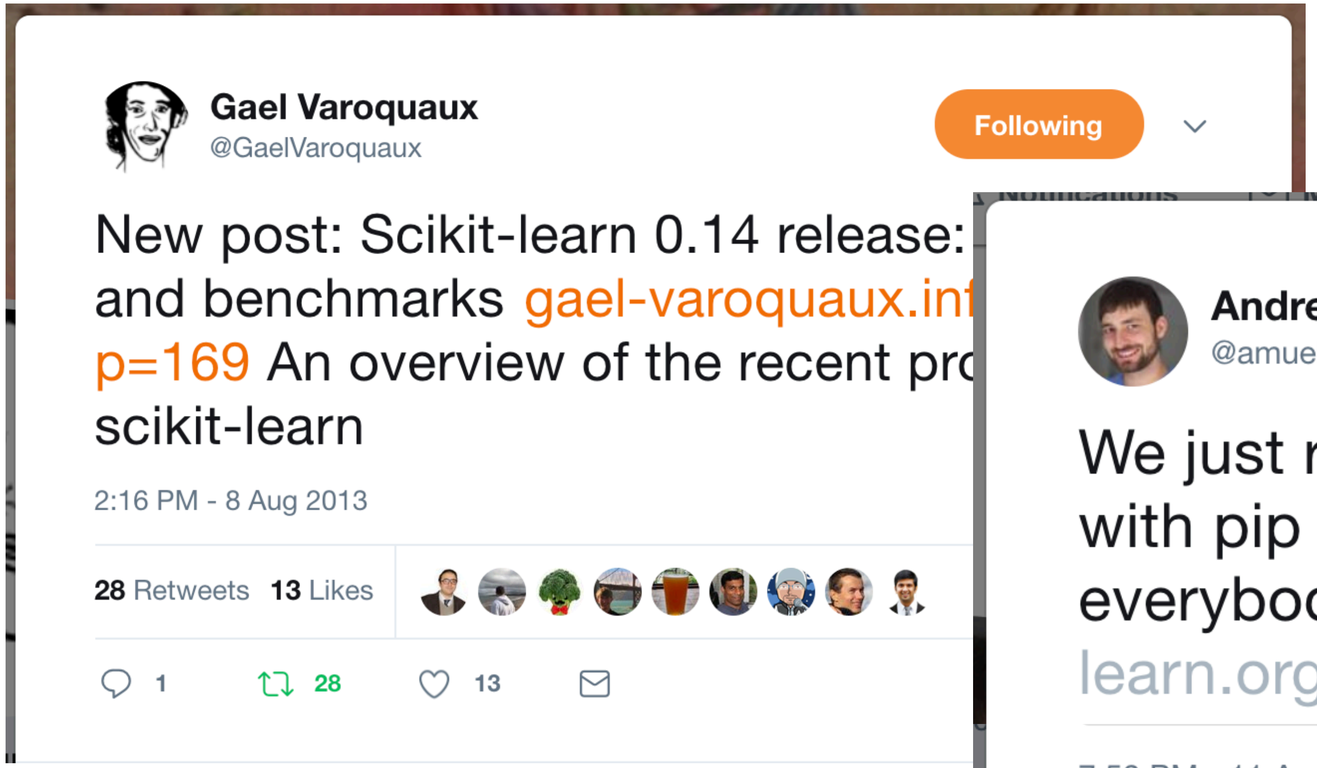
We use a **very** permissive license : BSD

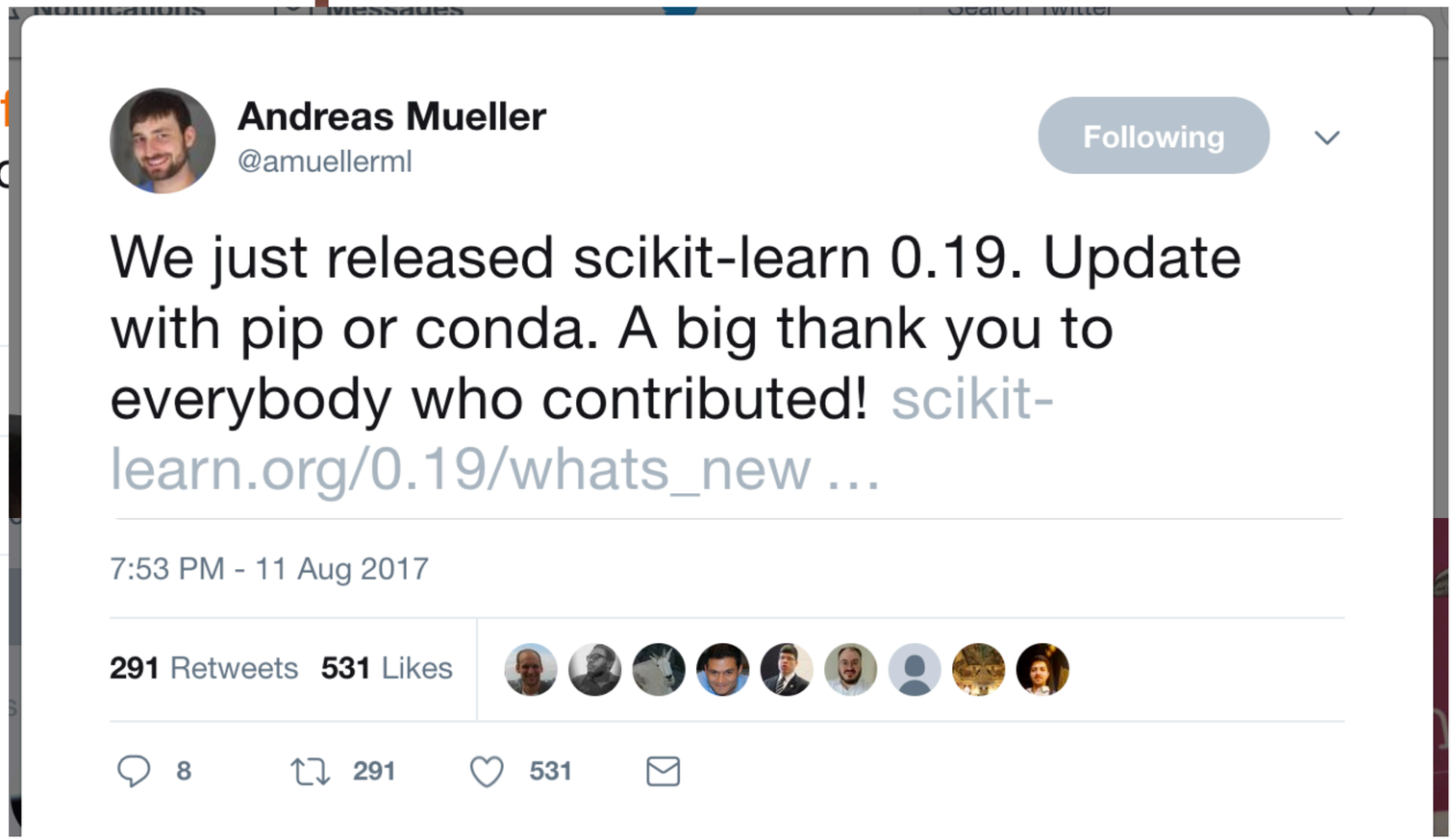
Reason 8: Fight code complexity

- Keep bounds on **technical difficulty**
- **Minimize dependencies:** limit install problems
- **Limit maintenance:** 2k loc. for a tiny feature is bad for long term

Write dumb code !

Reason 9: Communication





Reason 10

```
>>> print(success_reasons[10])
IndexError: list index out of range
```

Can we replicate/clone the model?

http://martinos.org/mne/

MNE Install Documentation API Examples Contribute Search

MNE

MEG + EEG ANALYSIS & VISUALIZATION

Speed

Multi-core CPU & GPU.

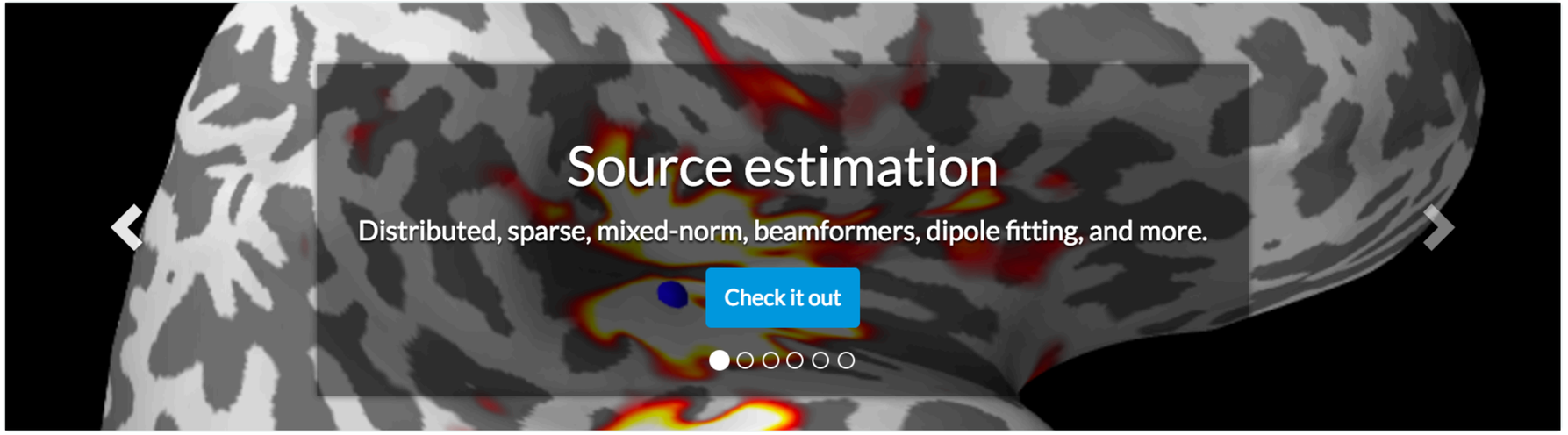
Usability

Clean scripting & visualization.

Flexibility

Broad data format & analysis support.

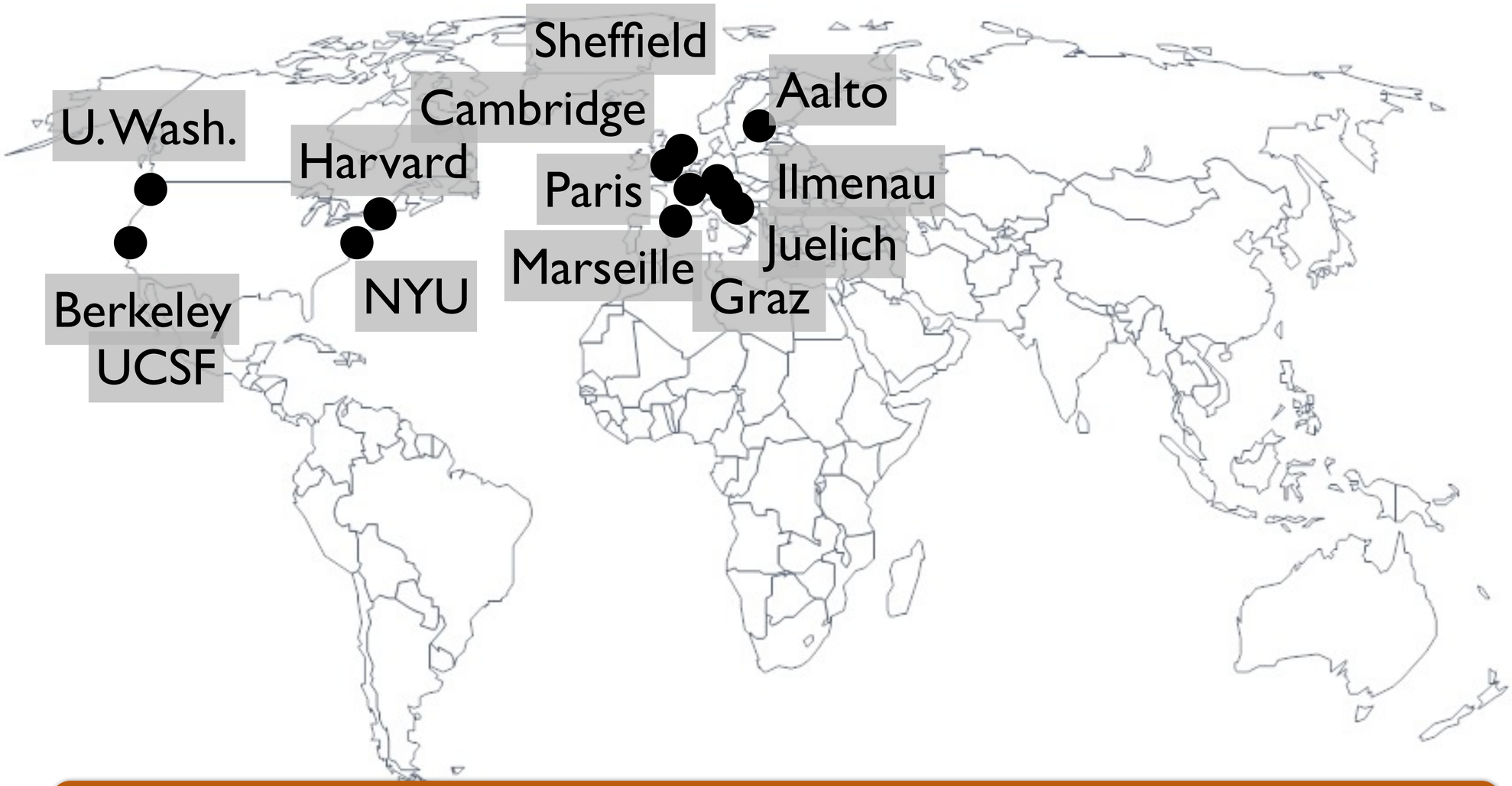
Open-source Python software for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, and more.



Data I/O | Preprocessing | Visualization

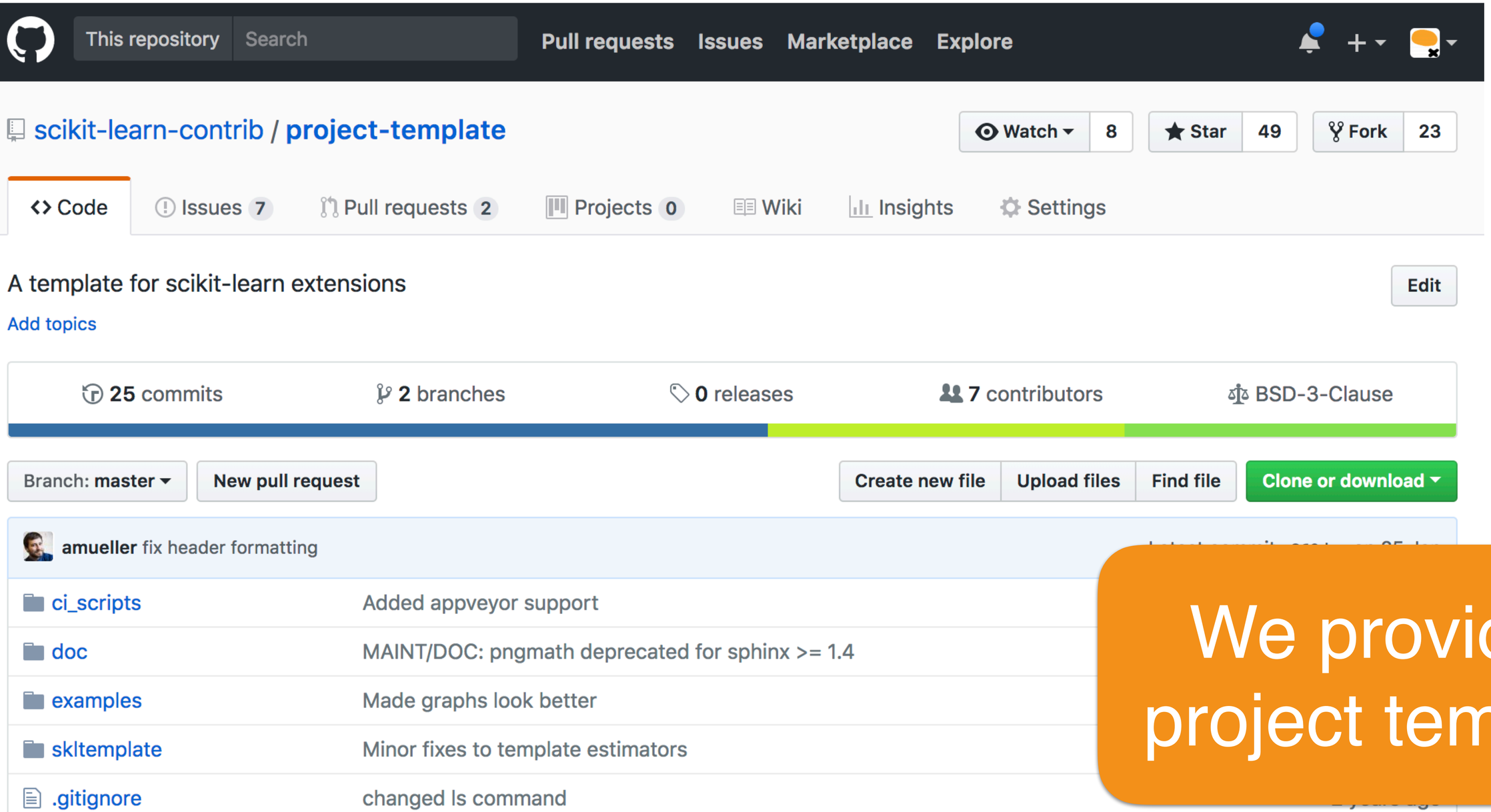
Domain specific package for processing of neural time series

Distributed development



Vision: Implement, share, document the best methods from all labs

https://github.com/scikit-learn-contrib/project-template



We provide a project template


Where is scikit-learn going?

scikit-learn-contrib

Scaling the scikit-learn universe quicker

https://github.com/scikit-learn-contrib

Affiliated packages with the same API and tests

Curation is done outside of the core team

⇒ growth without burdening the team

scikit-learn-contrib projects

lightning

Large-scale linear classification, regression and ranking.

Maintained by Mathieu Blondel and Fabian Pedregosa.

py-earth

A Python implementation of Jerome Friedman's Multivariate Adaptive Regression Splines.

Maintained by Jason Rudy and Mehdi.

imbalanced-learn

Python module to perform under sampling and over sampling with various techniques.

Maintained by Guillaume Lemaitre, Fernando Nogueira, Dayvid Oliveira and Christos Aridas.

polylearn

Factorization machines and polynomial networks for classification and regression in Python.

Maintained by Vlad Niculae.

forest-confidence-interval

Confidence intervals for scikit-learn forest algorithms.

Maintained by Ariel Rokem, Kivan Polimis and Bryna Hazelton.

hdbscan

A high performance implementation of HDBSCAN clustering.

Maintained by Leland McInnes, jc-healy, c-north and Steve Astels.

categorical-encoding

A library of sklearn compatible categorical variable encoders.

Maintained by Will McGinnis

boruta_py

Python implementations of the Boruta all-relevant feature selection method.

Maintained by Daniel Homola

sklearn-pandas

Pandas integration with sklearn.

Maintained by Israel Saeta Pérez

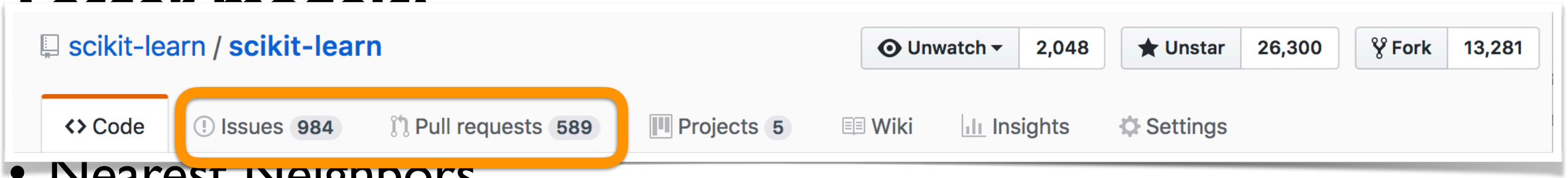
skope-rules

Machine learning with logical rules in Python.

Maintained by Florian Gardin, Ronan Gautier, Nicolas Goix and Jean-Matthieu Schertzer.

What's next?

- **Better integration columnar data:**
  - e.g. with Pandas DataFrame object with `ColumnsTransformer`
- **Faster models:**
  - Nearest Neighbors
  - Linear models
- **Scaling out:**
  - Distributed computing with Joblib / Dask integration



Funding and sustainability

- New York (A. Mueller):
  - $350,000 Moore-Sloan grant
  - A. Mueller (full time). Students: M. Kumar, V. Birodkar
- Telecom ParisTech (A. Gramfort):
  - 200 000€ WendelinIA grant + 12 000€ CDS
  - Programmers: T. Guillemot, T. Dupré
  - Students: M. Kumar, D. Sullivan, V.R. Rajagopalan, N. Goix
- Inria Parietal (G. Varoquaux):
  - 120 000€ Inria + 100 000€ WendelinIA
  - + 50 000€ ANR + 30 000€ CDS
  - Programmers: O. Grisel, L. Esteve, G. Lemaitre, J. Van den Bossche
  - Students: A. Mensch, J. Schreiber, G. Patrini

Total spending in 2015-2016 > 400k€

Concluding Thoughts…

… on contributing to open source as a researcher

- ***Alone you go fast, together we go far!***
  - With 3 developers, for 1 contribution you get 2. Imagine with 40…
- **Good software means better science**
- **Open source works best for building blocks packages** (not cutting edge research)
- **Publish academic papers** on software to get research credit (e.g. JMLR MLOSS, …)

"All models are wrong but some come with good and documented open source implementation so use those"

Contact:

Alexandre Gramfort
http://alexandre.gramfort.net

GitHub : @agramfort

Twitter : @agramfort

Thanks !

Travis CI

Sphinx-Gallery