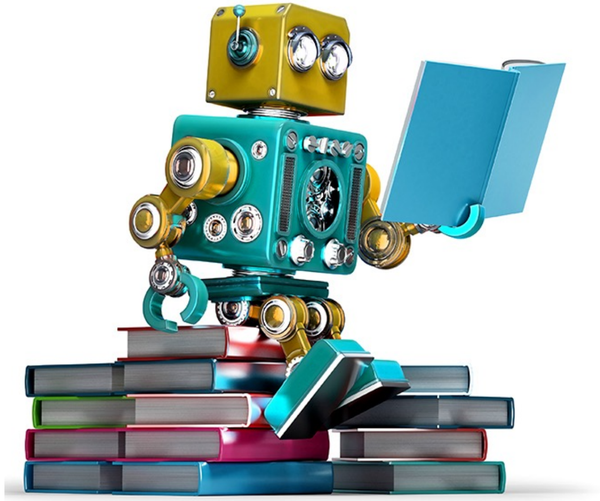
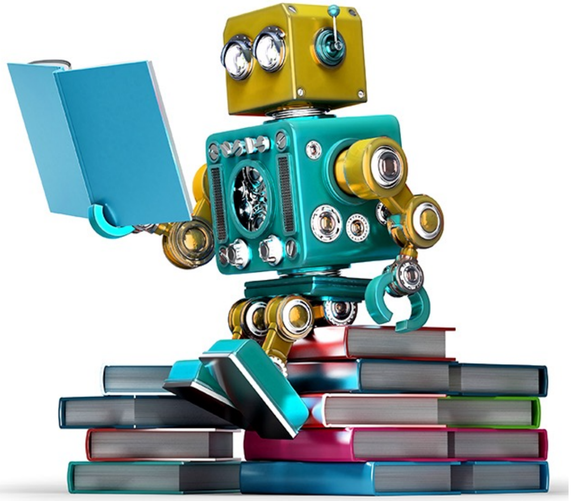
How to drive scientific progress with community-driven open source projects: the scikit-learn approach





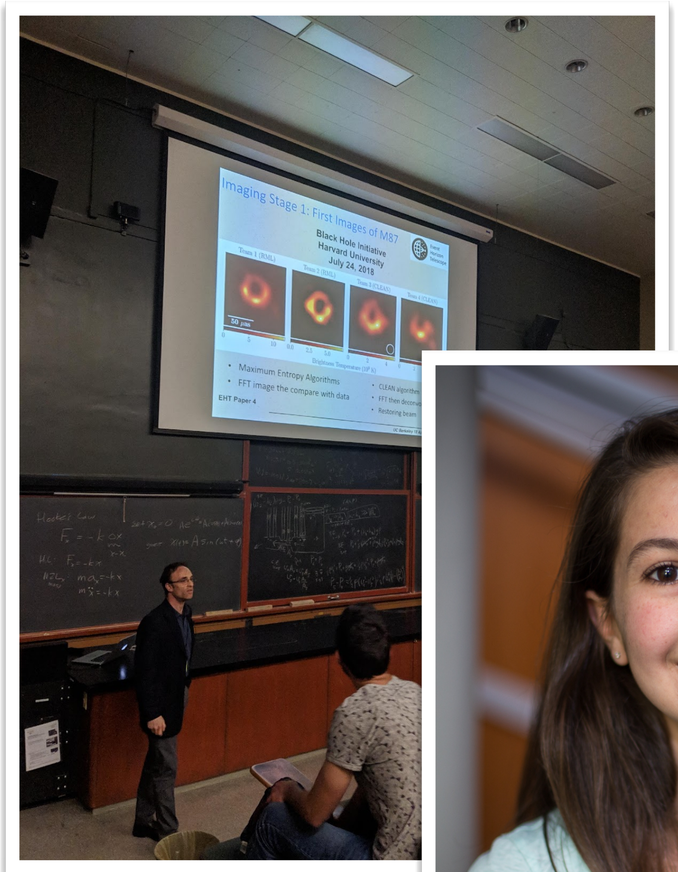
Alexandre Gramfort

http://alexandre.gramfort.net

GitHub : @agramfort

Twitter : @agramfort

2019: First image of black holes!



Shep Doeleman
Harvard

Katie Bouman
MIT/Caltech

THE ASTROPHYSICAL JOURNAL LETTERS

OPEN ACCESS

First M87 Event Horizon Telescope Results. III. Data Processing and Calibration

The Event Horizon Telescope Collaboration, Kazunori Akiyama[1,2,3,4], Antxon Alberdi[5], Walter Alef[6], Keiichi Asada[7], Rebecca Azulay[8,9,6], Anne-Kathrin Baczko[6], David Ball[10], Mislav Baloković[4,11], John Barrett[2] +Show full author list

Published 2019 April 10 • © 2019. The American Astronomical Society.
The Astrophysical Journal Letters, Volume 875, Number 1

Software: DiFX (Deller et al. 2011), CALC, PolConvert (Martí-Vidal et al. 2016), HOPS (Whitney et al. 2004), CASA (McMullin et al. 2007), AIPS (Greisen 2003), ParselTongue (Kettenis et al. 2006), GNU Parallel (Tange 2011), GILDAS, eht-imaging (Chael et al. 2016, 2018), Numpy (van der Walt et al. 2011), Scipy (Jones et al. 2001), Pandas (McKinney 2010), Astropy (The Astropy Collaboration et al. 2013, 2018), Jupyter (Kluyver et al. 2016), Matplotlib (Hunter 2007).

Event Horizon Telescope

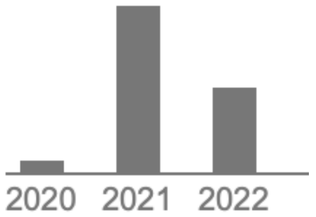
Review Article | Open Access | Published: 16 September 2020

Array programming with NumPy

Charles R. Harris, K. Jarrod Millman ✉, Stéfan J. van der Walt ✉, Ralf Gommers ✉, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke & Travis E. Oliphant — Show fewer authors

Nature **585**, 357–362 (2020) | Cite this article

Total citations Cited by 4699



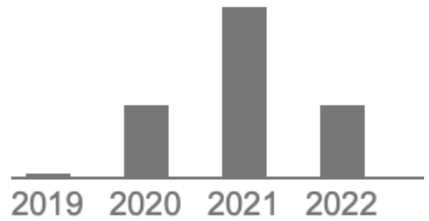
Perspective | Open Access | Published: 03 February 2020

SciPy 1.0: fundamental algorithms for scientific computing in Python

Pauli Virtanen, Ralf Gommers ✉, Travis E. Oliphant, Matt Haberland ✉, Tyler Reddy ✉, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt & SciPy 1.0 Contributors — Show fewer authors

Nature Methods **17**, 261–272 (2020) | Cite this article

Total citations Cited by 10087



scikit-learn.org

scikit learn | Install | User Guide | API | Examples | Community | More | Go

scikit-learn

Machine Learning in Python

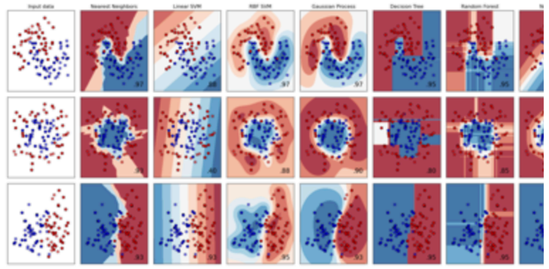
Getting Started | Release Highlights for 1.0 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

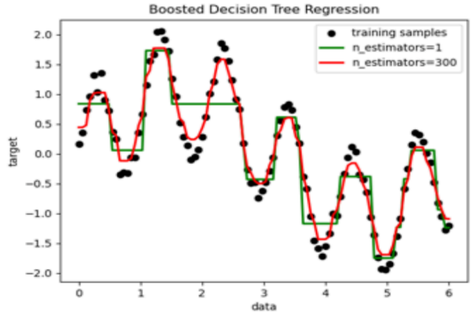


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

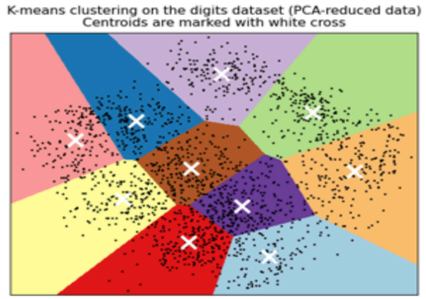


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

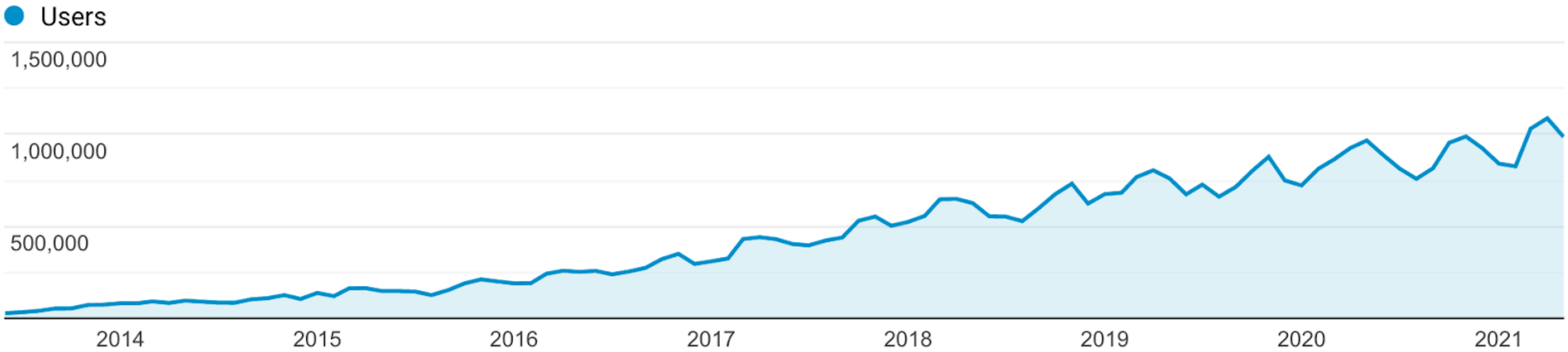


Examples

https://scikit-learn.org


What is the impact of scikit-learn?

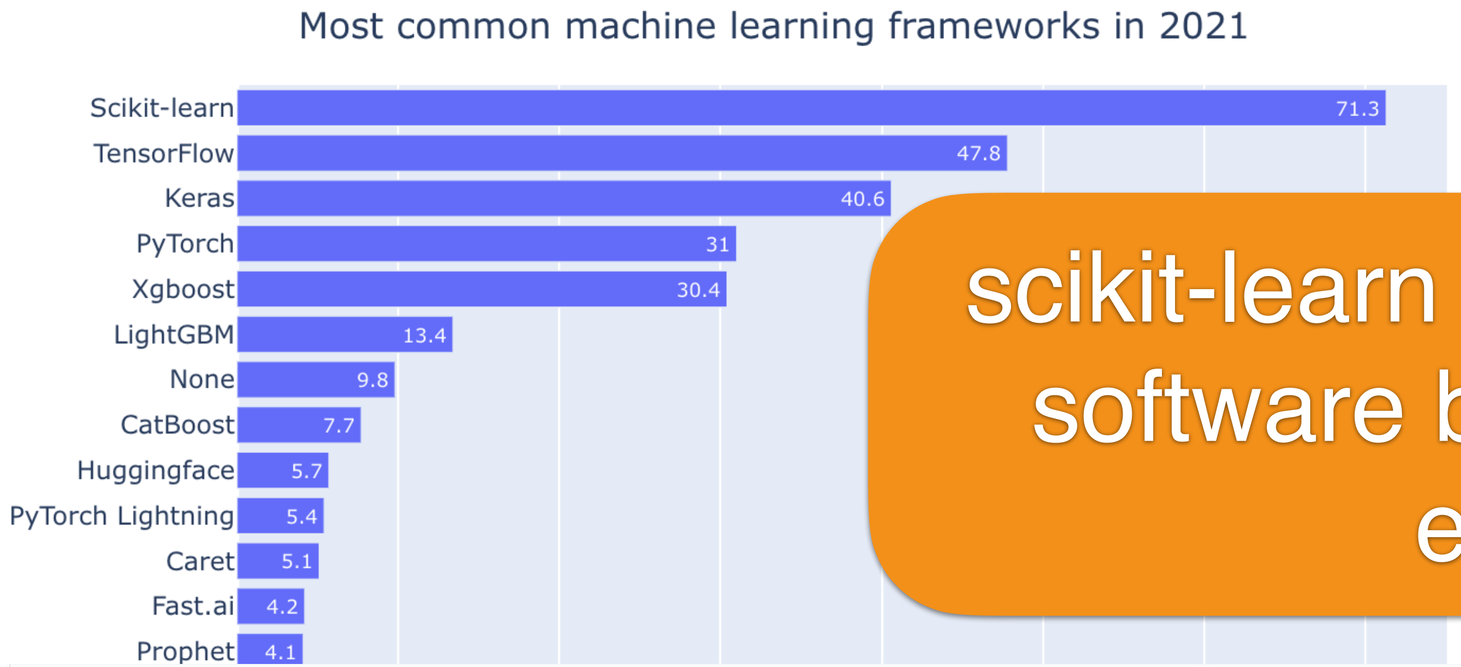
Visits of scikit-learn.org



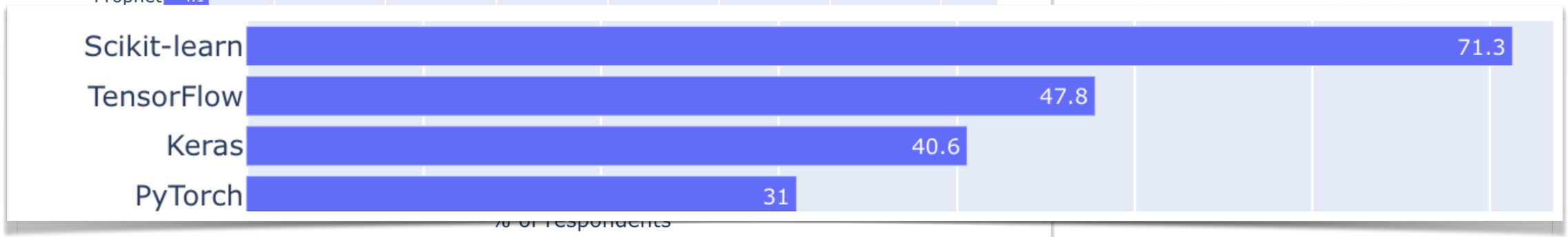
More than 1 million monthly returning users

2021 Kaggle Data Science & Machine Learning Survey

Notebook Data Logs Comments (5)



scikit-learn is the most used software by data science experts



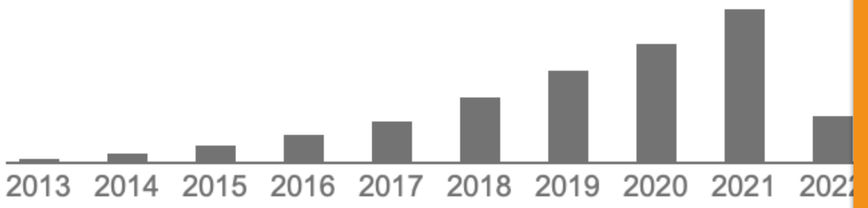
https://www.kaggle.com/paultimothymooney/2021-kaggle-data-science-machine-learning-survey

Academic impact

Scikit-learn: Machine learning in Python

[PDF] from jmlr.org

Authors	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, Edouard Duchesnay
Publication date	2011/11/1
Journal	the Journal of machine Learning research
Volume	12
Pages	2825-2830
Publisher	JMLR. org
Total citations	Cited by 54786



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

Outline

- Past: How did we get there?
- Future: How do we keep it going?
- Can one replicate the success in other scientific disciplines?


How did it start?

scikit-learn the origin in 2010

scikit-learn the origin in 2010

Really started with Inria funding in 2010 in PARIETAL team

http://www.youtube.com/watch?v=uhCF-zlk0jY

courtesy of Gael Varoquaux

http://www.youtube.com/watch?v=uhCF-zlk0jY

courtesy of Gael Varoquaux

Current Biology 21, 1641–1646, October 11, 2011 ©2011 Elsevier Ltd All rights reserved DOI 10.1016/j.cub.2011.08.031

Report

Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies

Presented clip

Clip reconstructed from brain activity

http://www.youtube.com/watch?v=nsjDnYxJ0bo

Current Biology 21, 1641–1646, October 11, 2011 ©2011 Elsevier Ltd All rights reserved DOI 10.1016/j.cub.2011.08.031

Report

Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies

Presented clip

Clip reconstructed from brain activity

http://www.youtube.com/watch?v=nsjDnYxJ0bo


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]

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()
```

Simple API often copied

```python
import numpy as np
from sklearn.datasets import make_classification
from torch import nn
from skorch import NeuralNetClassifier


X, y = make_classification(1000, 20, n_informative=10)
X = X.astype(np.float32)
y = y.astype(np.int64)

class MyModule(nn.Module):
    def __init__(self, num_units=10, nonlin=nn.ReLU()):
        super(MyModule, self).__init__()

        self.dense0 = nn.Linear(20, num_units)
        self.nonlin = nonlin
        self.dropout = nn.Dropout(0.5)
        self.dense1 = nn.Linear(num_units, num_units)
        self.output = nn.Linear(num_units, 2)
        self.softmax = nn.Softmax(dim=-1)

    def forward(self, X, **kwargs):
        X = self.nonlin(self.dense0(X))
        X = self.dropout(X)
        X = self.nonlin(self.dense1(X))
        X = self.softmax(self.output(X))
        return X
```

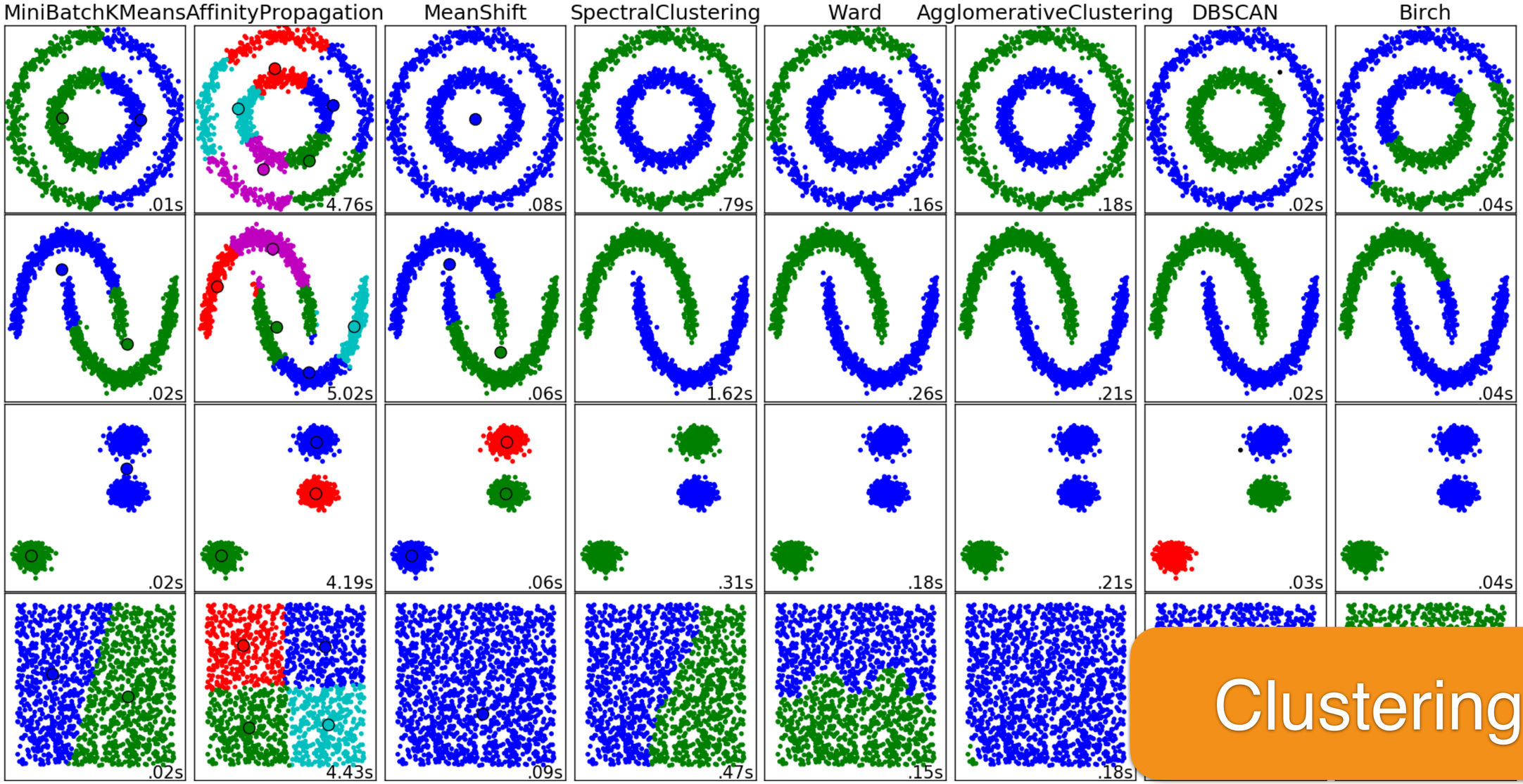
```python
net = NeuralNetClassifier(
    MyModule,
    max_epochs=10,
    lr=0.1,
    # Shuffle training data on each epoch
    iterator_train__shuffle=True,
)

net.fit(X, y)
y_proba = net.predict_proba(X)
```

SKORCH

https://github.com/skorch-dev/skorch

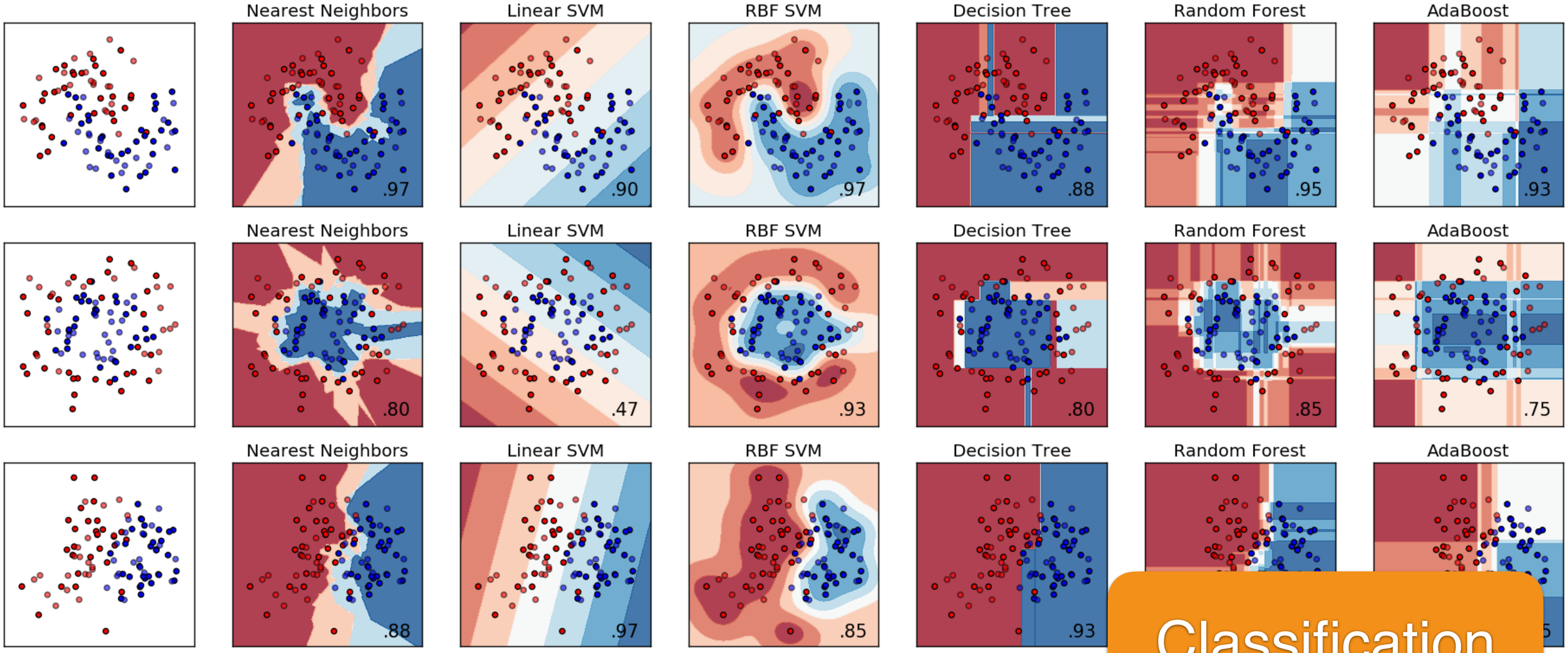
Reason 1: Rich feature set



Clustering

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

Reason 1: Rich feature set



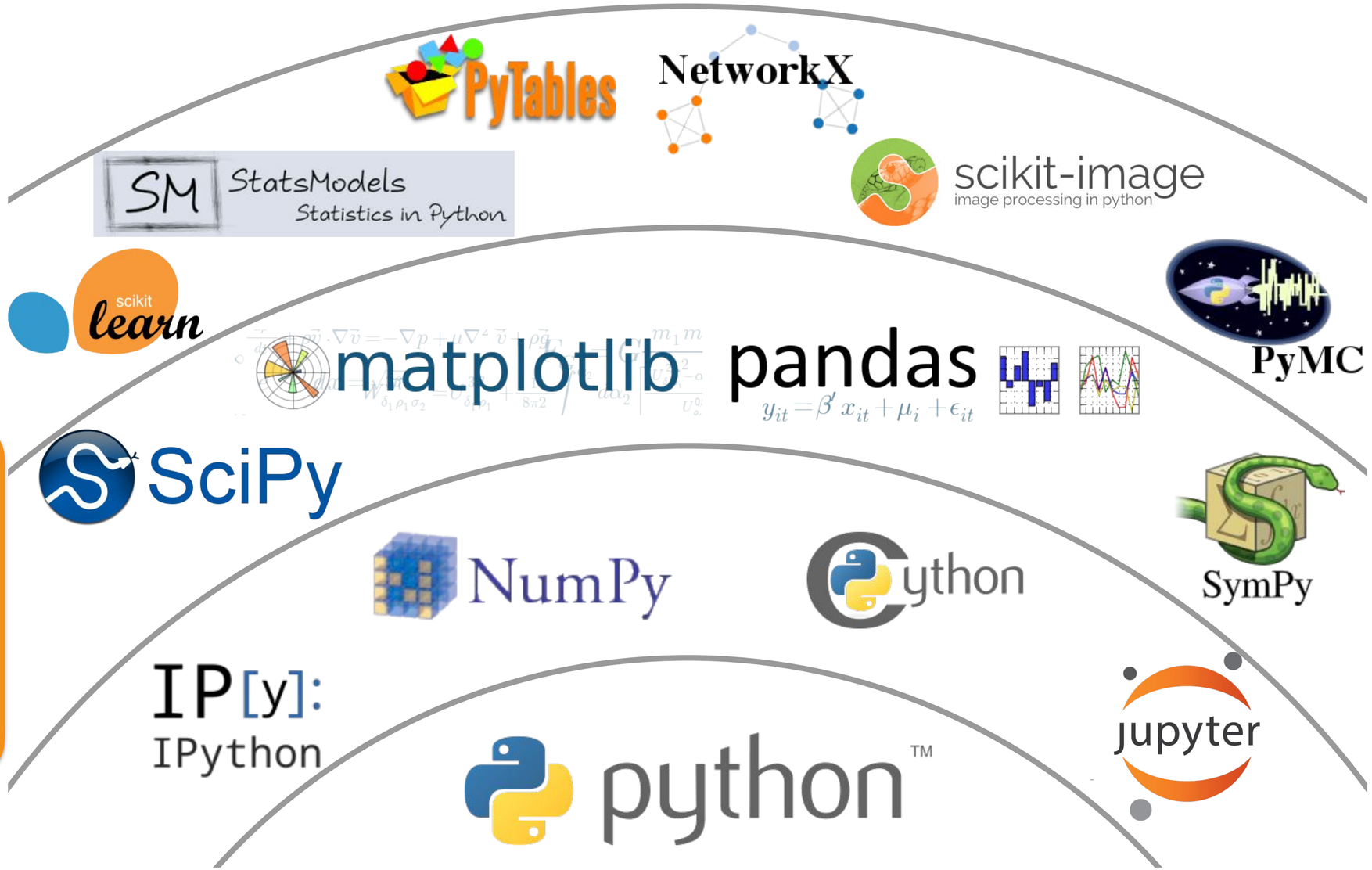
Classification

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

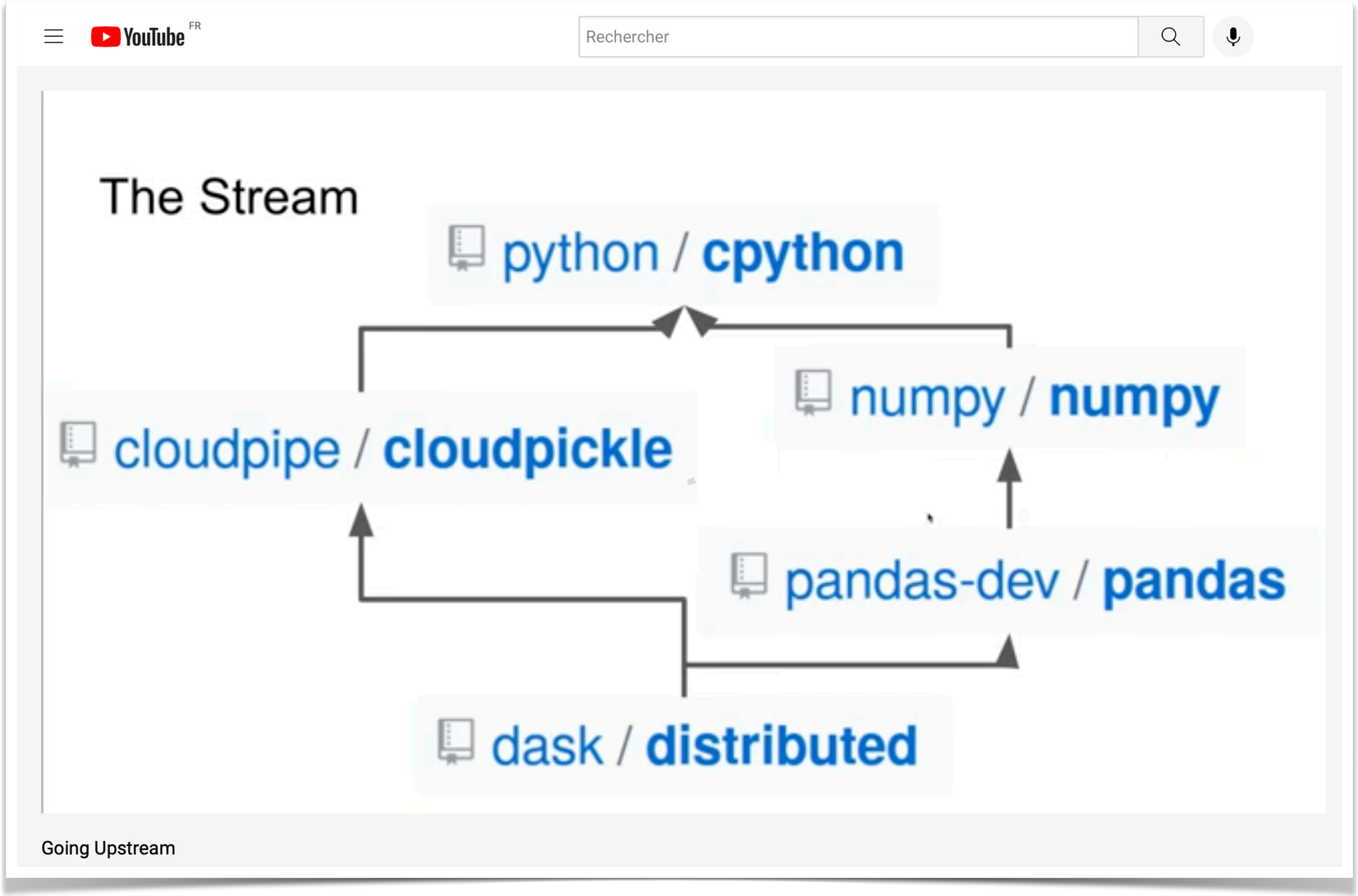
Reason 2: Clear Scope

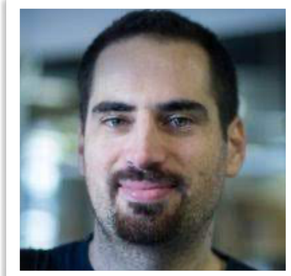
We provide one component in the Python ecosystem

New models if they are well established and demonstrate extra empirical benefit



“Going Upstream” by Olivier Grisel





@ogrisel

https://www.youtube.com/watch?v=PzuvBFWKwcA&t=1102s

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

Make the rules clear

Developers' Tips and Tricks

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

Utilities for Developers

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 3: Contributing guidelines

Previous sklearn.learn... | Next Contributing

scikit-learn v0.19.1
Other versions

Please **cite us** if you use the software.

Developer's Guide

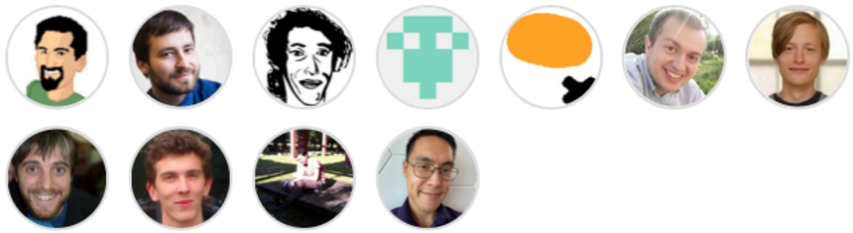
Developer's Guide

Contributing

- Submitting a bug report
- Ways to contribute
- Retrieving the latest code

Make the rules clear

Contributors 2,161



+ 2,150 contributors

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

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

Provide feedback with code reviews

Previous

Reason 4: Automate review process

All checks have passed
8 successful checks

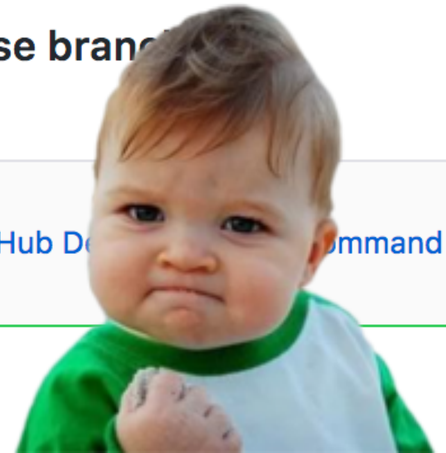
Hide all checks

- ✓ **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 bran
Merging can be performed automatically.

Squash and merge

You can also open this in GitHub D mmand line instructions.



Travis CI

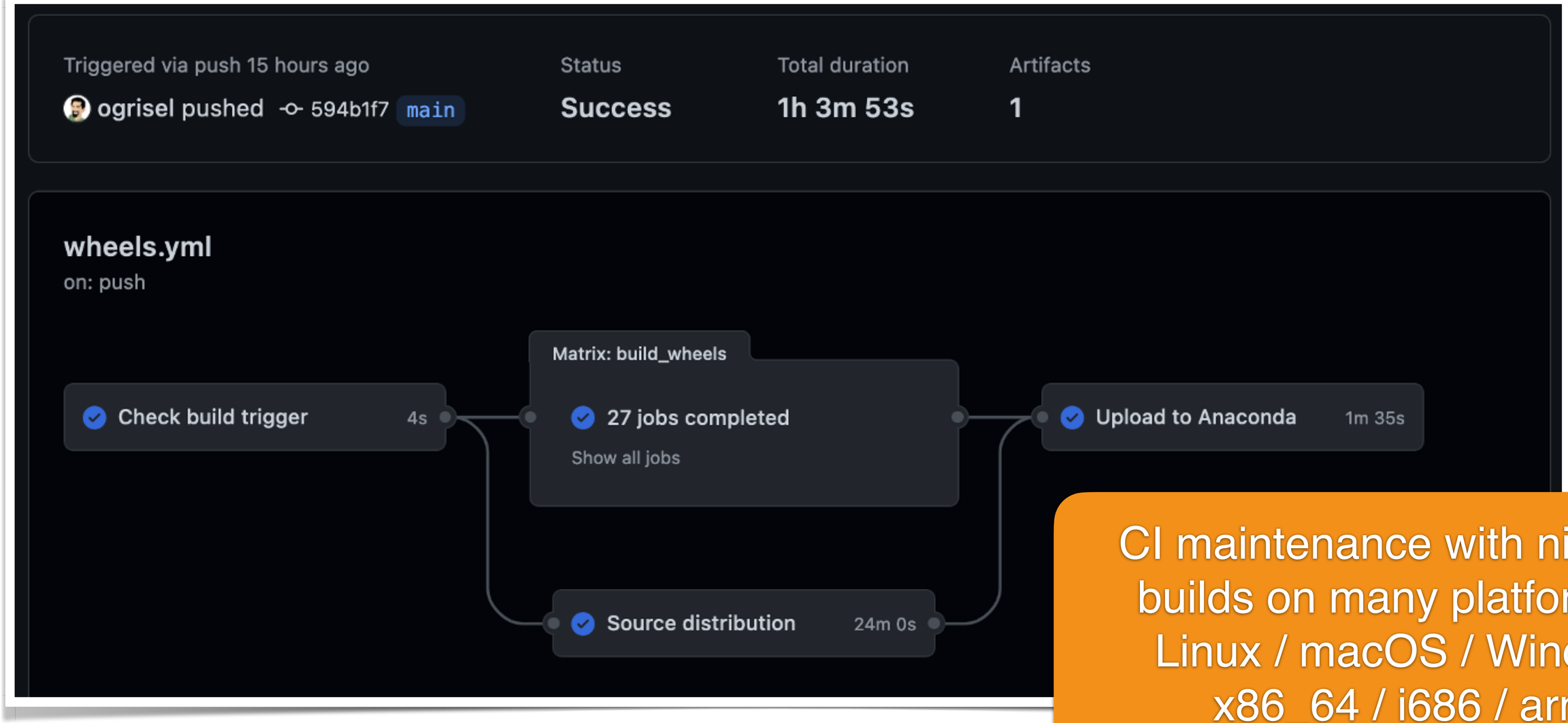
Azure Pipelines

ANACONDA

circleci

GitHub Actions

Reason 4bis: Automate release process



CI maintenance with nightly builds on many platforms:
Linux / macOS / Windows
x86_64 / i686 / arm64 / ppc64le

Reason 5: Documentation & Examples

scikit-learn.org

Install User Guide API Examples Community More Go

scikit-learn

Machine Learning in Python

Getting Started | Release Highlights for 1.0 | GitHub

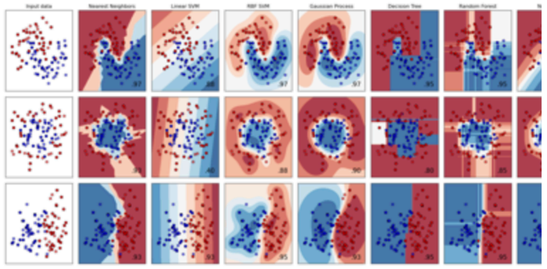
- Simple and efficient tools for pred
- Accessible to everybody, and reu
- Built on NumPy, SciPy, and matpl
- Open source, commercially usabl

If there is no doc it does not exist !

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

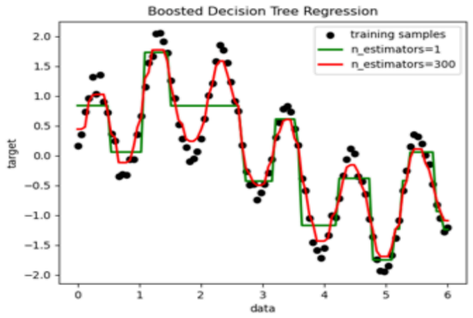


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...



Examples

Cluster

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Mea
shift, and more...

Examples

Don’t accept code without doc and one example

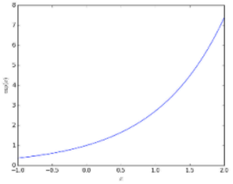
Reason 5: Documentation & Examples

Gallery of Examples

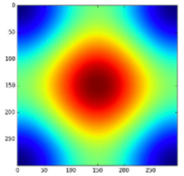
General examples

General-purpose and introductory examples from the sphinx-gallery

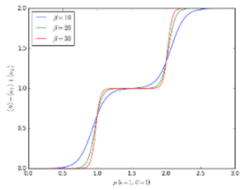
Using sys.argv in examples



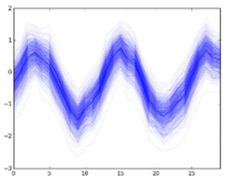
Plotting the exponential function



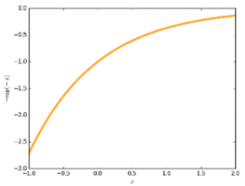
Colormaps alter your perception



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
- Changelog
  - API changes
  - Bug fixes
  - Enhancements
- Code and Documentation Contributors

Version 0.19
- Highlights
- Changed models
- Changelog
  - New features
  - Enhancements
  - Bug fixes
- API changes summary
- Code and Documentation Contributors

Version 0.18.2
- Changelog

Release history

Version 0.19.1

October, 2017

This is a bug-fix release with some minor documentation improvements and enhancements to features released in 0.19.0.

Note there may be minor differences in TSNE output in this release (due to #9623), in the case where multiple samples have equal distance to some sample.

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 CV 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 6: Don't own the project

Release history
Version 0.19.1
- Changelog
  - API changes
  - Bug fixes
  - Enhancements
- Code and Documentation Contributors

Version 0.19
- Highlights
- Changed models
- Changelog
  - New features
  - Enhancements
  - Bug fixes
- API changes summary
- Code and Documentation Contributors

Version 0.18.2
- Changelog

Release history

Version 0.19.1

October, 2017

This is a bug-fix release wit

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 CV 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

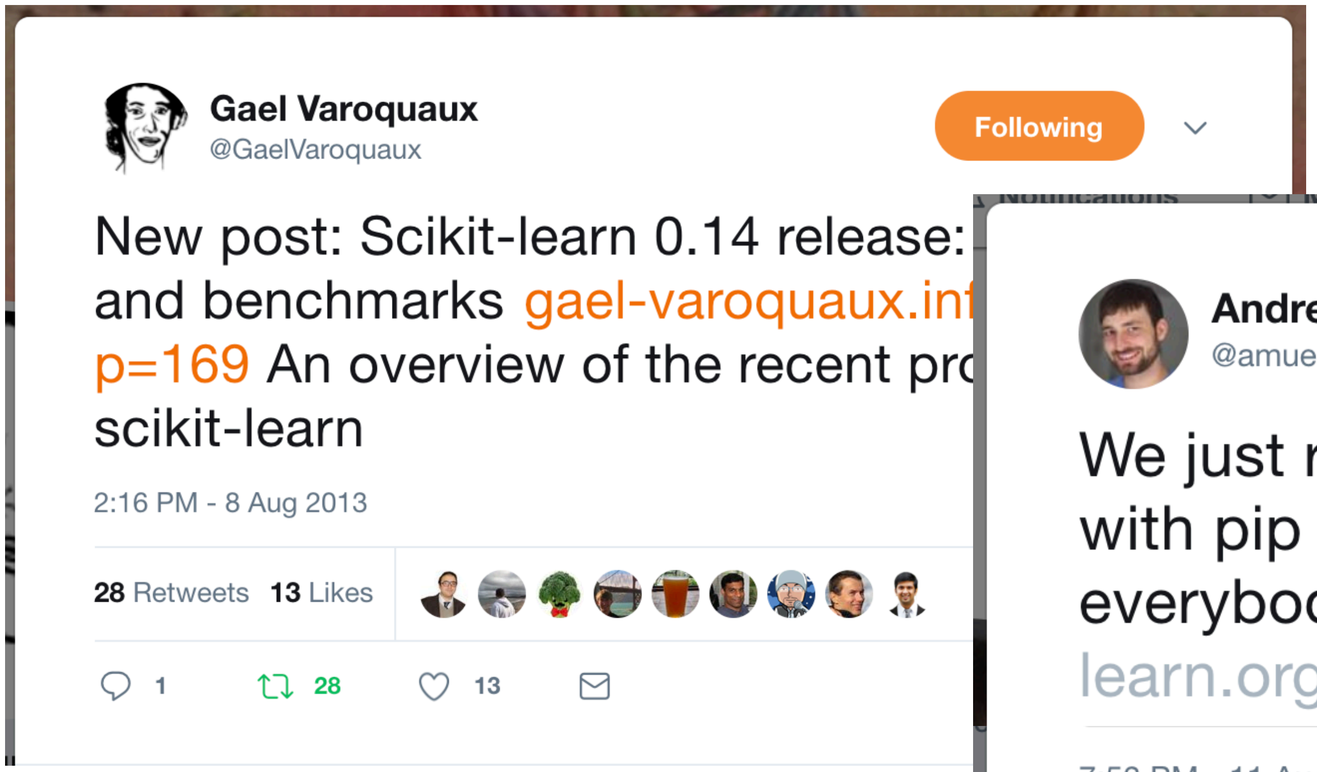
It’s a team effort beyond France and the Inria Consortium

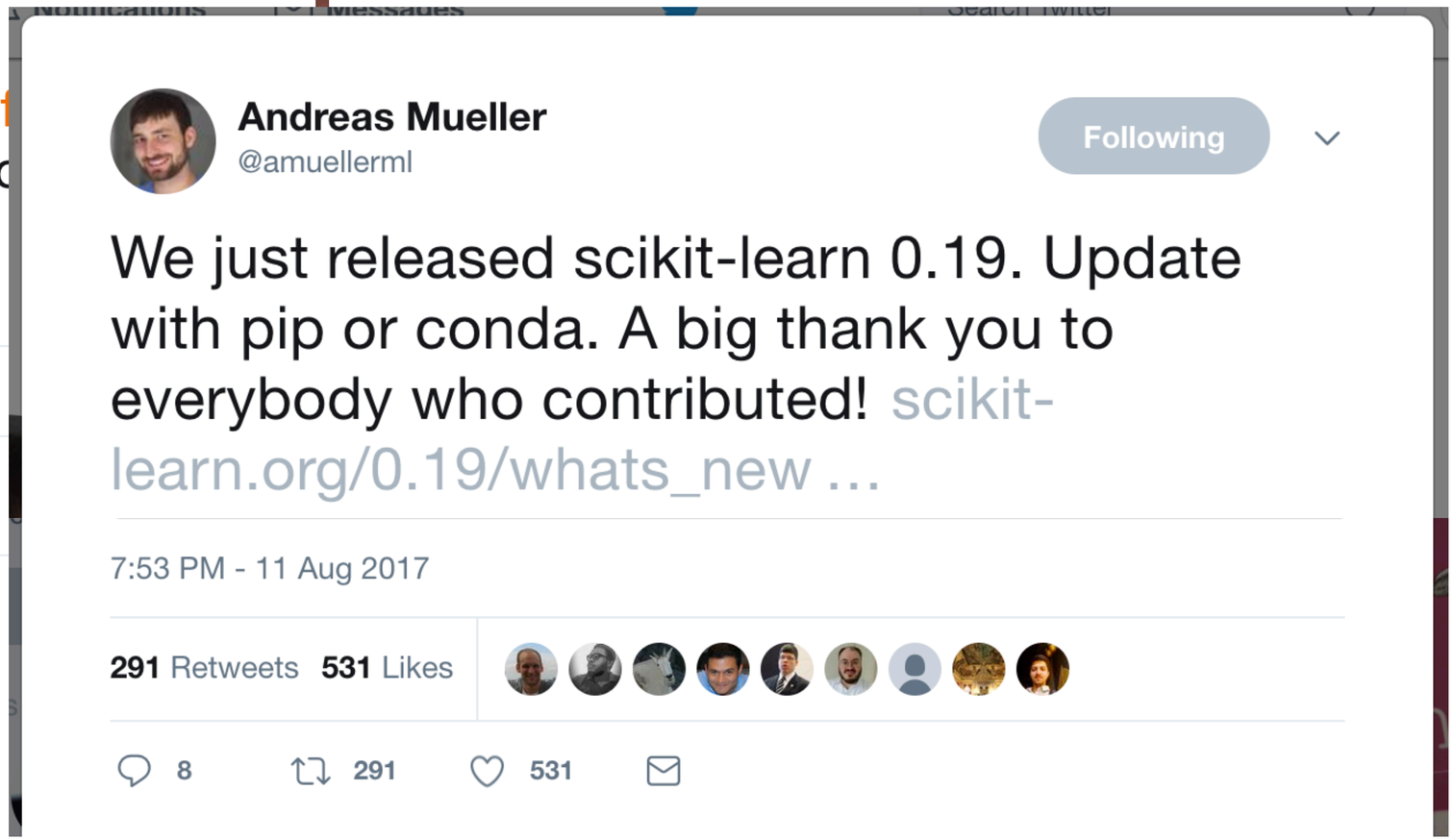
Reason 8: Fight code complexity

- 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
```


What are you doing now
to make it last?

Big Effort on Usability

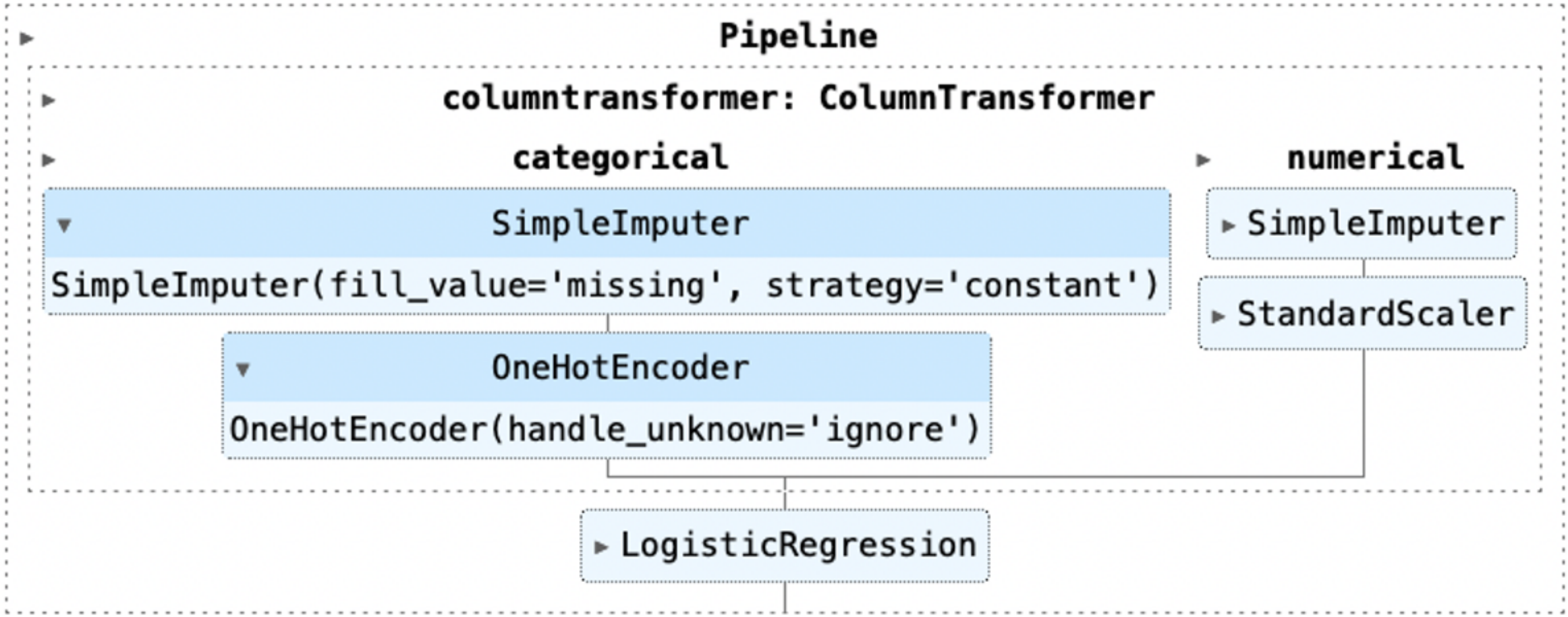
scikit-learn 1.0:

```
ValueError: Input contains NaN, infinity or a value too large for
dtype('float64')
```

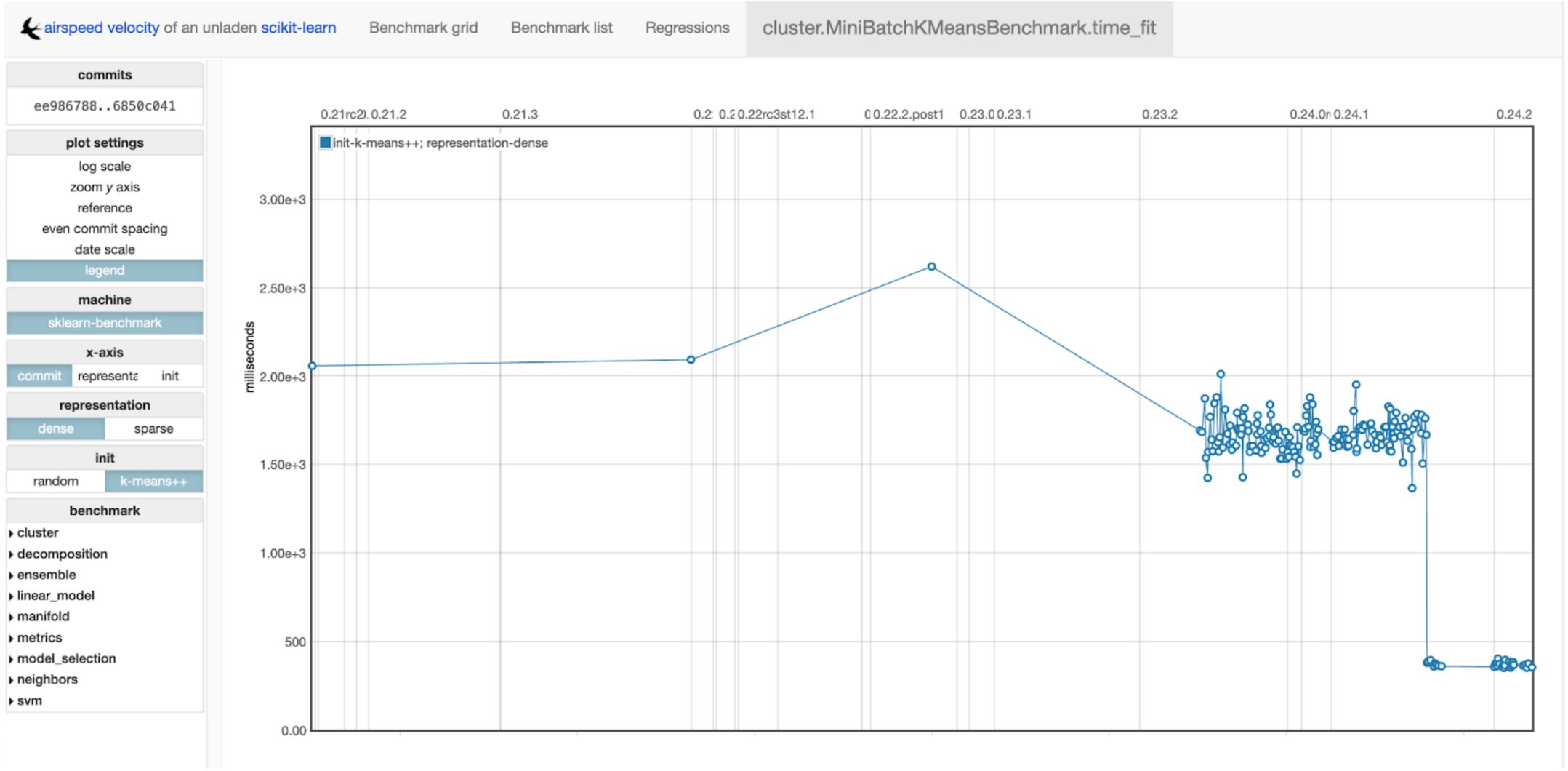
scikit-learn 1.1:

```
ValueError: Input X contains NaN. LogisticRegression does not accept missing
values encoded as NaN natively. For supervised learning, you might want to
consider sklearn.ensemble.HistGradientBoostingClassifier and Regressor which
accept missing values encoded as NaNs natively. Alternatively, it is possible
to preprocess the data, for instance by using an imputer transformer in a
pipeline or drop samples with missing values. See https://scikit-learn.org/
stable/modules/impute.html
```

Big Effort on Usability

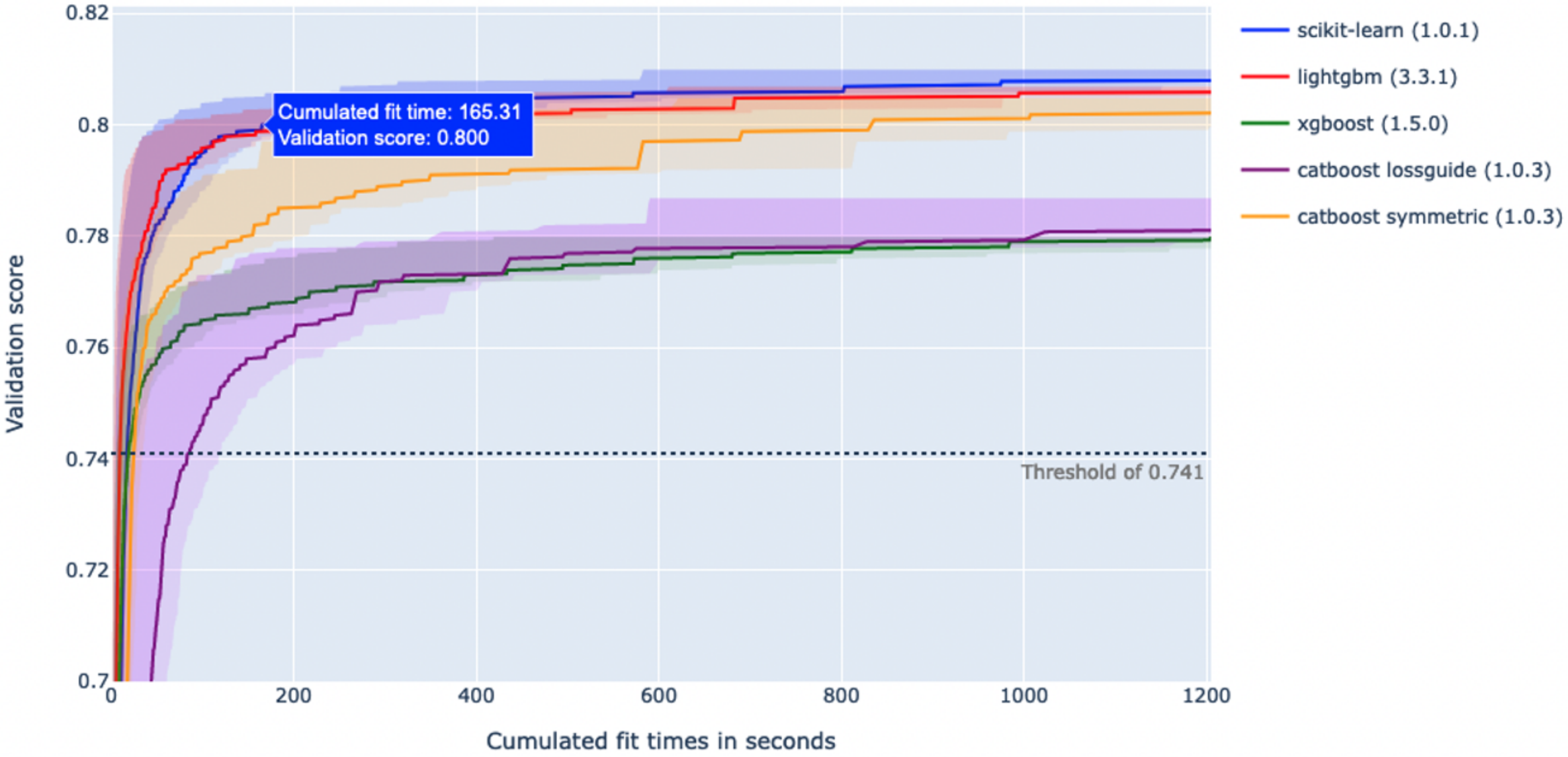


Big Effort on Performance



https://scikit-learn.org/scikit-learn-benchmarks/

Big Effort on Performance



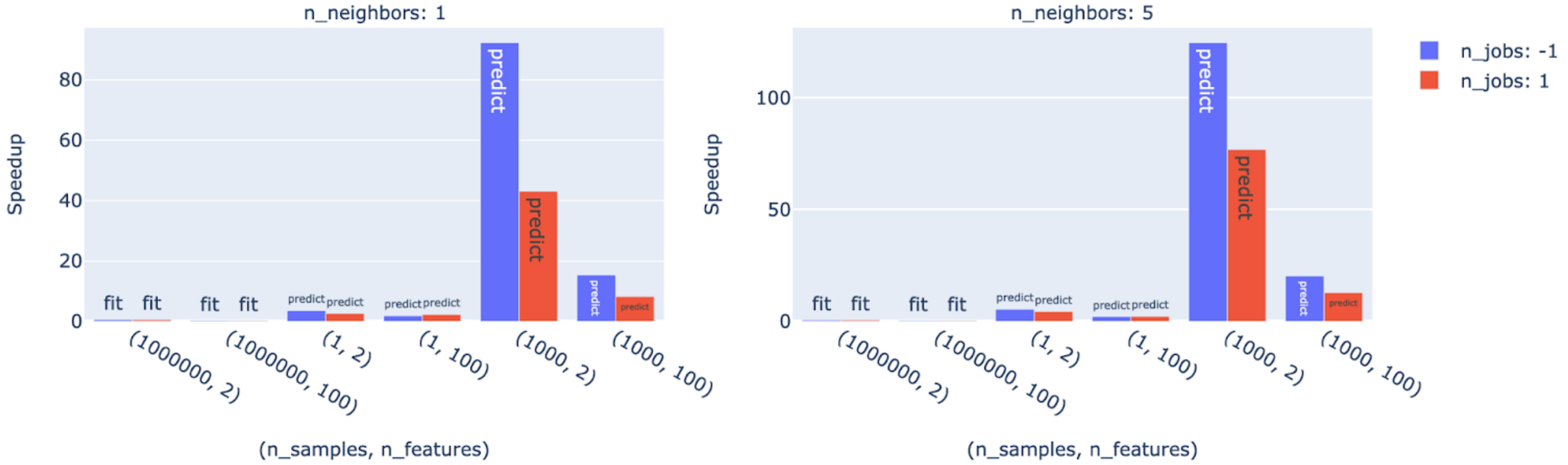
Comparative performance measurements

KNeighborsClassifier vs daal4py

All estimators share the following hyperparameters:

	value
algorithm	brute

https://github.com/mbatoul/sklearn_benchmarks



New features: Generalized Additive Models (GAMs)

Spline-based #3482

- Smooth decision function
- Performance can be challenging but R's mgcv / bam can scale

GBRT-based #19914

- First PoC as interaction constraints in Hist GBRT #19148
- Need to be aligned with more general interaction constraints definition
- Very fast and accurate but non-smooth and still newish among GAM users

FUN MOOC

Search for courses

ogrisel | Help | English

News | Courses | Organizations

You are here: Home › Courses › Machine learning in python with scikit-learn

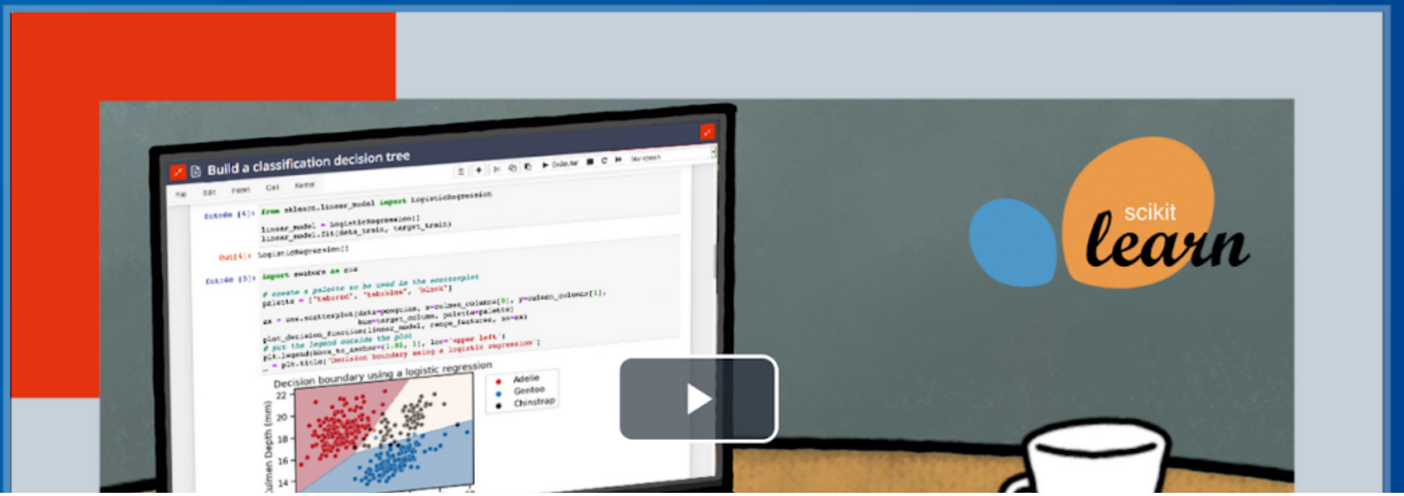
Computer science and programming | Digital and technology

Machine learning in Python with scikit-learn

Ref. 41026

Duration: 8 weeks | Effort: 35 hours | Pace: ~4h15/week

Build predictive models with scikit-learn and gain a practical understanding of the strengths and limitations of machine learning!



Inria

Enrollment
From April 19, 2021 to July 13, 2021

Course
From May 18, 2021 to July 14, 2021

Languages
English

Go to course

You are enrolled in this course run

https://www.fun-mooc.fr/fr/cours/machine-learning-python-scikit-learn/

https://scikit-learn.fondation-inria.fr/

Scikit-learn @ Inria Foundation

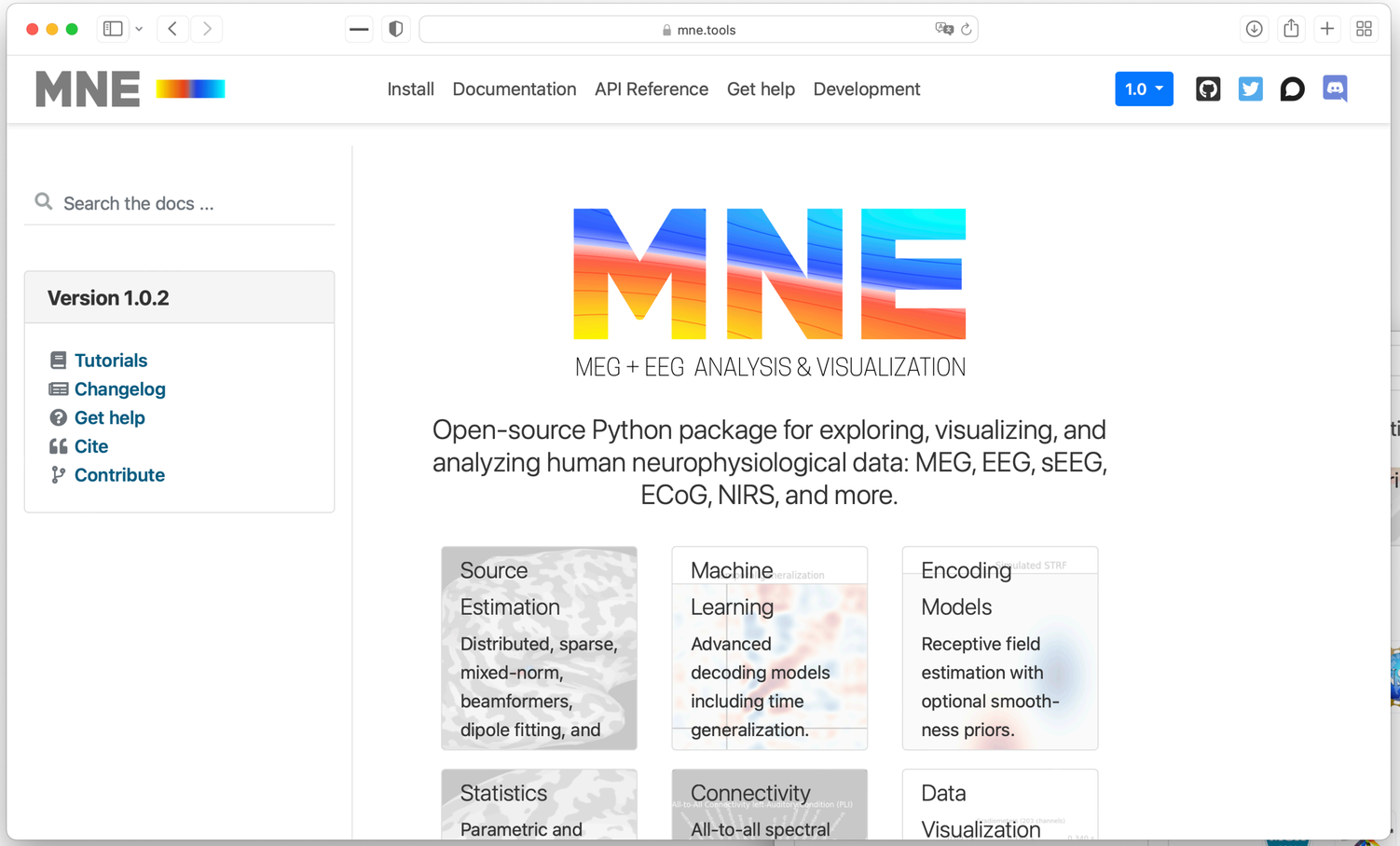
Fostering growth and sustainability of the reference machine-learning toolbox

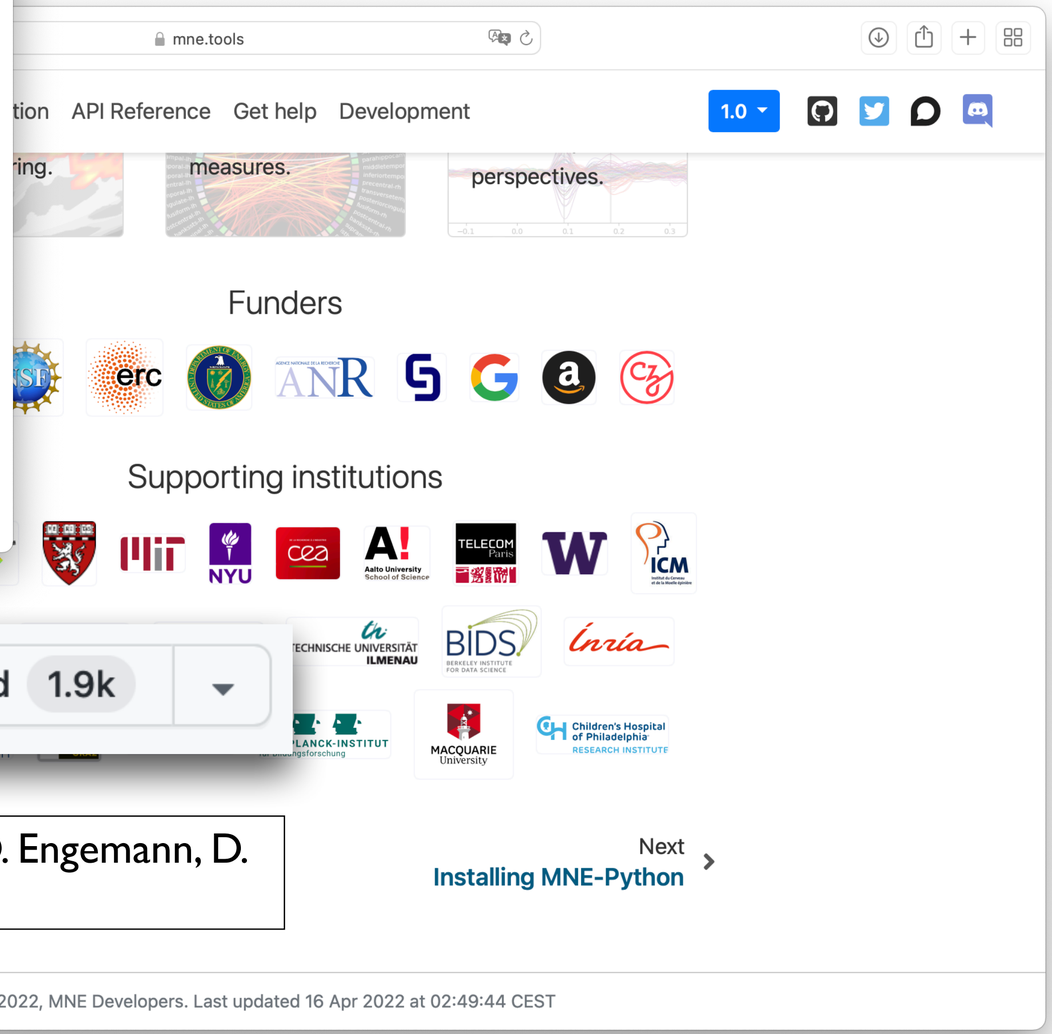
with the help of our partners

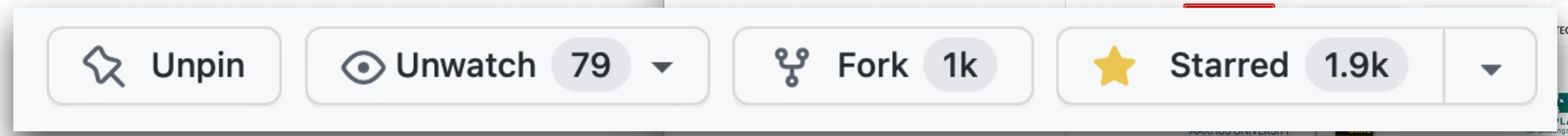
BCG GAMMA · Microsoft · BNP PARIBAS CARDIF · AXA · FUJITSU · ASSISTANCE PUBLIQUE HÔPITAUX DE PARIS · data iku


Can we replicate/clone the model?

https://mne.tools/







MNE software for processing MEG and EEG data, A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, M. Hämäläinen, Neuroimage 2013

https://nilearn.github.io

nilearn.github.io

Nilearn:

Statistics for NeuroImaging in Python

SVM Ward clustering
Searchlight ICA
Nifti IO Datasets

ENHANCED BY Google

Nilearn Home | User Guide | Examples | Reference | Glossary | Bibliography | Nilearn: Statistical Analysis for NeuroImaging in Python Nipy ecosystem

This is the *stable* documentation for the latest release of Nilearn, the current development version is available here.

Nilearn enables **approachable and versatile analyses of brain volumes**. It provides statistical and machine-learning tools, with **instructive documentation & open community**.

Unpin | Unwatch 78 | Fork 456 | Starred 827

Nilearn now includes the functionality of Nistats. Here's a guide to replacing Nistats imports to work in Nilearn.

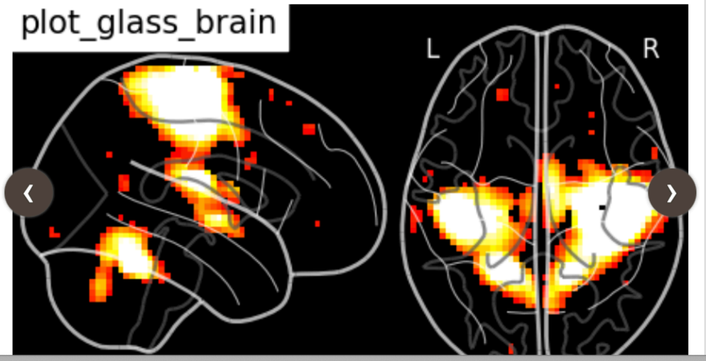
First Steps

Get started with nilearn

Examples

Visit our example gallery

User Guide



News

April 2022: Nilearn 0.9.1 released
January 2022: Nilearn 0.9.0 released
September 2021: Nilearn 0.8.1 released

Ongoing: What's new.

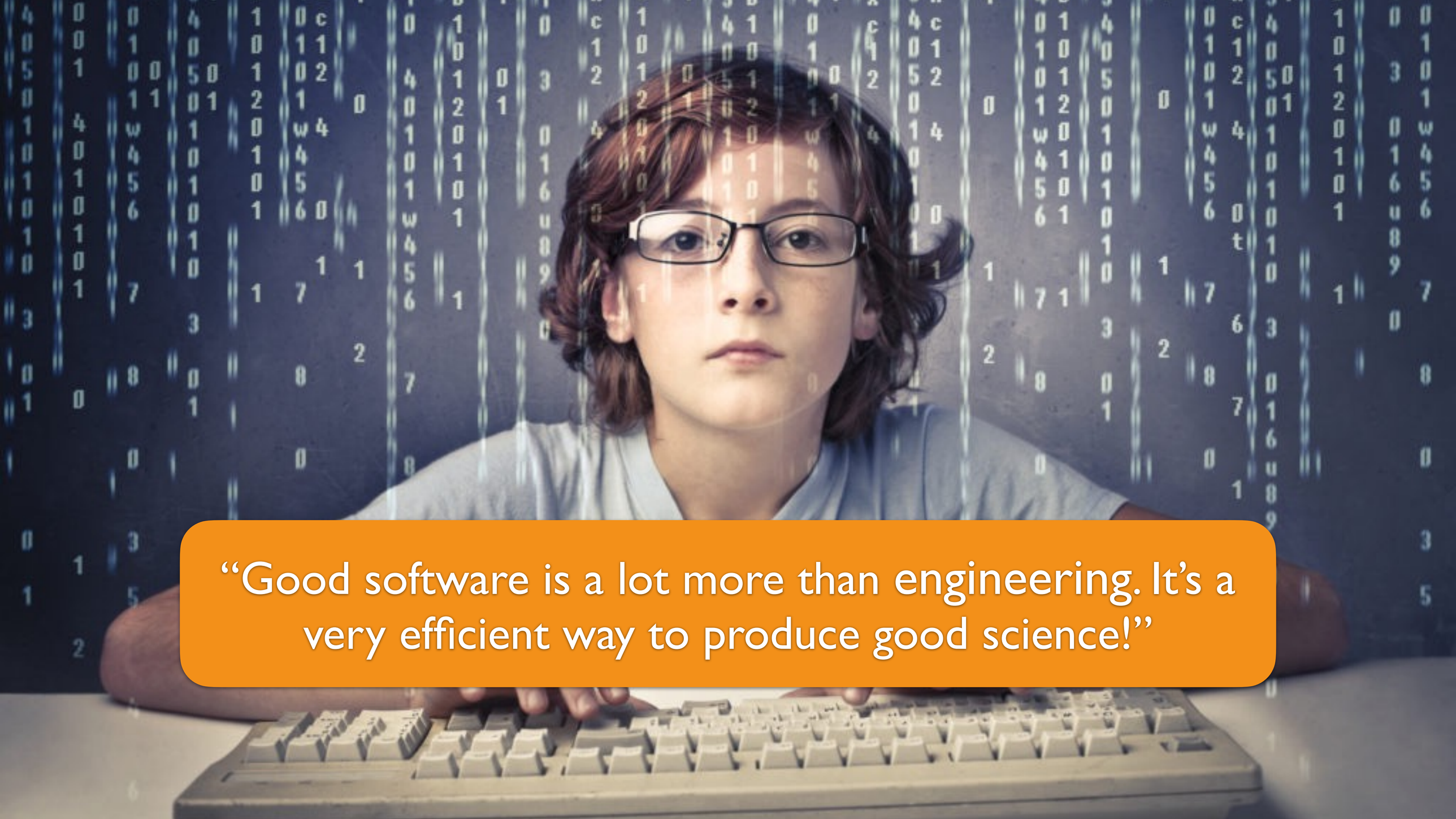
Software

Installation

Development

Nilearn on GitHub
All material Free Software: **BSD license** (3 clause).
Authors
Contributing

Giving credit


"Good software is a lot more than engineering. It's a very efficient way to produce good science!"

Contact:

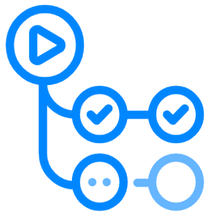
Alexandre Gramfort
http://alexandre.gramfort.net

GitHub : @agramfort

Twitter : @agramfort

Thanks !

Sphinx-Gallery



GitHub Actions

Azure Pipelines