CHAPTER

ONE

WELCOME TO SCIKIT-LEARN

1.1 Installing scikit-learn

Note: If you wish to contribute to the project, it's recommended you *install the latest development version*.

1.1.1 Installing the latest release

Scikit-learn requires:

- Python (>= 3.5)
- NumPy (>= 1.11.0)
- SciPy (>= 0.17.0)
- joblib (>= 0.11)

Scikit-learn plotting capabilities (i.e., functions start with "**plot_**") require Matplotlib (>= 1.5.1). Some of the scikit-learn examples might require one or more extra dependencies: scikit-image (>= 0.12.3), pandas (>= 0.18.0).

Warning: Scikit-learn 0.20 was the last version to support Python 2.7 and Python 3.4. Scikit-learn now requires Python 3.5 or newer.

If you already have a working installation of numpy and scipy, the easiest way to install scikit-learn is using `pip`

```
pip install -U scikit-learn
```

or `conda`:

```
conda install scikit-learn
```

If you have not installed NumPy or SciPy yet, you can also install these using conda or pip. When using pip, please ensure that *binary wheels* are used, and NumPy and SciPy are not recompiled from source, which can happen when using particular configurations of operating system and hardware (such as Linux on a Raspberry Pi). Building numpy and scipy from source can be complex (especially on Windows) and requires careful configuration to ensure that they link against an optimized implementation of linear algebra routines. Instead, use a third-party distribution as described below.

If you must install scikit-learn and its dependencies with pip, you can install it as `scikit-learn[alldeps]`. The most common use case for this is in a `requirements.txt` file used as part of an automated build process for a PaaS application or a Docker image. This option is not intended for manual installation from the command line.

Note: For installing on PyPy, PyPy3-v5.10+, Numpy 1.14.0+, and scipy 1.1.0+ are required.

For installation instructions for more distributions see other distributions. For compiling the development version from source, or building the package if no distribution is available for your architecture, see the *Advanced installation instructions*.

1.1.2 Third-party Distributions

If you don't already have a python installation with numpy and scipy, we recommend to install either via your package manager or via a python bundle. These come with numpy, scipy, scikit-learn, matplotlib and many other helpful scientific and data processing libraries.

Available options are:

Canopy and Anaconda for all supported platforms

Canopy and Anaconda both ship a recent version of scikit-learn, in addition to a large set of scientific python library for Windows, Mac OSX and Linux.

Anaconda offers scikit-learn as part of its free distribution.

Warning: To upgrade or uninstall scikit-learn installed with Anaconda or `conda` you **should not use the pip command**. Instead:

To upgrade `scikit-learn`:

```
conda update scikit-learn
```

To uninstall `scikit-learn`:

```
conda remove scikit-learn
```

Upgrading with `pip install -U scikit-learn` or uninstalling `pip uninstall scikit-learn` is likely fail to properly remove files installed by the `conda` command.

pip upgrade and uninstall operations only work on packages installed via `pip install`.

WinPython for Windows

The WinPython project distributes scikit-learn as an additional plugin.

1.2 Frequently Asked Questions

Here we try to give some answers to questions that regularly pop up on the mailing list.

1.2.1 What is the project name (a lot of people get it wrong)?

scikit-learn, but not scikit or SciKit nor sci-kit learn. Also not scikits.learn or scikits-learn, which were previously used.

1.2.2 How do you pronounce the project name?

sy-kit learn. sci stands for science!

1.2.3 Why scikit?

There are multiple scikits, which are scientific toolboxes built around SciPy. You can find a list at https://scikits.appspot.com/scikits. Apart from scikit-learn, another popular one is scikit-image.

1.2.4 How can I contribute to scikit-learn?

See *Contributing*. Before wanting to add a new algorithm, which is usually a major and lengthy undertaking, it is recommended to start with *known issues*. Please do not contact the contributors of scikit-learn directly regarding contributing to scikit-learn.

1.2.5 What's the best way to get help on scikit-learn usage?

For general machine learning questions, please use Cross Validated with the `[machine-learning]` tag.

For scikit-learn usage questions, please use Stack Overflow with the `[scikit-learn]` and `[python]` tags. You can alternatively use the mailing list.

Please make sure to include a minimal reproduction code snippet (ideally shorter than 10 lines) that highlights your problem on a toy dataset (for instance from `sklearn.datasets` or randomly generated with functions of `numpy.random` with a fixed random seed). Please remove any line of code that is not necessary to reproduce your problem.

The problem should be reproducible by simply copy-pasting your code snippet in a Python shell with scikit-learn installed. Do not forget to include the import statements.

More guidance to write good reproduction code snippets can be found at:

https://stackoverflow.com/help/mcve

If your problem raises an exception that you do not understand (even after googling it), please make sure to include the full traceback that you obtain when running the reproduction script.

For bug reports or feature requests, please make use of the issue tracker on GitHub.

There is also a scikit-learn Gitter channel where some users and developers might be found.

Please do not email any authors directly to ask for assistance, report bugs, or for any other issue related to scikit-learn.

1.2.6 How should I save, export or deploy estimators for production?

See *Model persistence*.

1.2.7 How can I create a bunch object?

Don't make a bunch object! They are not part of the scikit-learn API. Bunch objects are just a way to package some numpy arrays. As a scikit-learn user you only ever need numpy arrays to feed your model with data.

For instance to train a classifier, all you need is a 2D array `X` for the input variables and a 1D array `y` for the target variables. The array `X` holds the features as columns and samples as rows . The array `y` contains integer values to encode the class membership of each sample in `X`.

1.2.8 How can I load my own datasets into a format usable by scikit-learn?

Generally, scikit-learn works on any numeric data stored as numpy arrays or scipy sparse matrices. Other types that are convertible to numeric arrays such as pandas DataFrame are also acceptable.

For more information on loading your data files into these usable data structures, please refer to *loading external datasets*.

1.2.9 What are the inclusion criteria for new algorithms ?

We only consider well-established algorithms for inclusion. A rule of thumb is at least 3 years since publication, 200+ citations and wide use and usefulness. A technique that provides a clear-cut improvement (e.g. an enhanced data structure or a more efficient approximation technique) on a widely-used method will also be considered for inclusion.

From the algorithms or techniques that meet the above criteria, only those which fit well within the current API of scikit-learn, that is a `fit`, `predict/transform` interface and ordinarily having input/output that is a numpy array or sparse matrix, are accepted.

The contributor should support the importance of the proposed addition with research papers and/or implementations in other similar packages, demonstrate its usefulness via common use-cases/applications and corroborate performance improvements, if any, with benchmarks and/or plots. It is expected that the proposed algorithm should outperform the methods that are already implemented in scikit-learn at least in some areas.

Inclusion of a new algorithm speeding up an existing model is easier if:

- it does not introduce new hyper-parameters (as it makes the library more future-proof),
- it is easy to document clearly when the contribution improves the speed and when it does not, for instance "when n_features >> n_samples",
- benchmarks clearly show a speed up.

Also note that your implementation need not be in scikit-learn to be used together with scikit-learn tools. You can implement your favorite algorithm in a scikit-learn compatible way, upload it to GitHub and let us know. We will be happy to list it under *Related Projects*. If you already have a package on GitHub following the scikit-learn API, you may also be interested to look at scikit-learn-contrib.

1.2.10 Why are you so selective on what algorithms you include in scikit-learn?

Code is maintenance cost, and we need to balance the amount of code we have with the size of the team (and add to this the fact that complexity scales non linearly with the number of features). The package relies on core developers using their free time to fix bugs, maintain code and review contributions. Any algorithm that is added needs future attention by the developers, at which point the original author might long have lost interest. See also *What are the inclusion criteria for new algorithms ?*. For a great read about long-term maintenance issues in open-source software, look at the Executive Summary of Roads and Bridges

1.2.11 Why did you remove HMMs from scikit-learn?

See *Will you add graphical models or sequence prediction to scikit-learn?*.

1.2.12 Will you add graphical models or sequence prediction to scikit-learn?

Not in the foreseeable future. scikit-learn tries to provide a unified API for the basic tasks in machine learning, with pipelines and meta-algorithms like grid search to tie everything together. The required concepts, APIs, algorithms and expertise required for structured learning are different from what scikit-learn has to offer. If we started doing arbitrary structured learning, we'd need to redesign the whole package and the project would likely collapse under its own weight.

There are two project with API similar to scikit-learn that do structured prediction:

- pystruct handles general structured learning (focuses on SSVMs on arbitrary graph structures with approximate inference; defines the notion of sample as an instance of the graph structure)
- seqlearn handles sequences only (focuses on exact inference; has HMMs, but mostly for the sake of completeness; treats a feature vector as a sample and uses an offset encoding for the dependencies between feature vectors)

1.2.13 Will you add GPU support?

No, or at least not in the near future. The main reason is that GPU support will introduce many software dependencies and introduce platform specific issues. scikit-learn is designed to be easy to install on a wide variety of platforms. Outside of neural networks, GPUs don't play a large role in machine learning today, and much larger gains in speed can often be achieved by a careful choice of algorithms.

1.2.14 Do you support PyPy?

In case you didn't know, PyPy is an alternative Python implementation with a built-in just-in-time compiler. Experimental support for PyPy3-v5.10+ has been added, which requires Numpy 1.14.0+, and scipy 1.1.0+.

1.2.15 How do I deal with string data (or trees, graphs. . .)?

scikit-learn estimators assume you'll feed them real-valued feature vectors. This assumption is hard-coded in pretty much all of the library. However, you can feed non-numerical inputs to estimators in several ways.

If you have text documents, you can use a term frequency features; see *Text feature extraction* for the built-in *text vectorizers*. For more general feature extraction from any kind of data, see *Loading features from dicts* and *Feature hashing*.

Another common case is when you have non-numerical data and a custom distance (or similarity) metric on these data. Examples include strings with edit distance (aka. Levenshtein distance; e.g., DNA or RNA sequences). These can be encoded as numbers, but doing so is painful and error-prone. Working with distance metrics on arbitrary data can be done in two ways.

Firstly, many estimators take precomputed distance/similarity matrices, so if the dataset is not too large, you can compute distances for all pairs of inputs. If the dataset is large, you can use feature vectors with only one "feature", which is an index into a separate data structure, and supply a custom metric function that looks up the actual data in this data structure. E.g., to use DBSCAN with Levenshtein distances:

```
>>> from leven import levenshtein
>>> import numpy as np
>>> from sklearn.cluster import dbscan
>>> data = ["ACCTCCTAGAAG", "ACCTACTAGAAGTT", "GAATATTAGGCCGA"]
>>> def lev_metric(x, y):
...     i, j = int(x[0]), int(y[0])     # extract indices
...     return levenshtein(data[i], data[j])
...
>>> X = np.arange(len(data)).reshape(-1, 1)
>>> X
array([[0],
       [1],
       [2]])
>>> # We need to specify algoritum='brute' as the default assumes
>>> # a continuous feature space.
>>> dbscan(X, metric=lev_metric, eps=5, min_samples=2, algorithm='brute')
...
([0, 1], array([ 0,  0, -1]))
```

(This uses the third-party edit distance package `leven`.)

Similar tricks can be used, with some care, for tree kernels, graph kernels, etc.

1.2.16 Why do I sometime get a crash/freeze with n_jobs > 1 under OSX or Linux?

Several scikit-learn tools such as `GridSearchCV` and `cross_val_score` rely internally on Python's `multiprocessing` module to parallelize execution onto several Python processes by passing `n_jobs > 1` as argument.

The problem is that Python `multiprocessing` does a `fork` system call without following it with an `exec` system call for performance reasons. Many libraries like (some versions of) Accelerate / vecLib under OSX, (some versions of) MKL, the OpenMP runtime of GCC, nvidia's Cuda (and probably many others), manage their own internal thread pool. Upon a call to `fork`, the thread pool state in the child process is corrupted: the thread pool believes it has many threads while only the main thread state has been forked. It is possible to change the libraries to make them detect when a fork happens and reinitialize the thread pool in that case: we did that for OpenBLAS (merged upstream in master since 0.2.10) and we contributed a patch to GCC's OpenMP runtime (not yet reviewed).

But in the end the real culprit is Python's `multiprocessing` that does `fork` without `exec` to reduce the overhead of starting and using new Python processes for parallel computing. Unfortunately this is a violation of the POSIX standard and therefore some software editors like Apple refuse to consider the lack of fork-safety in Accelerate / vecLib as a bug.

In Python 3.4+ it is now possible to configure `multiprocessing` to use the 'forkserver' or 'spawn' start methods (instead of the default 'fork') to manage the process pools. To work around this issue when using scikit-learn, you can set the `JOBLIB_START_METHOD` environment variable to 'forkserver'. However the user should be aware that using the 'forkserver' method prevents joblib.Parallel to call function interactively defined in a shell session.

If you have custom code that uses `multiprocessing` directly instead of using it via joblib you can enable the 'forkserver' mode globally for your program: Insert the following instructions in your main script:

```
import multiprocessing

# other imports, custom code, load data, define model...

if __name__ == '__main__':
    multiprocessing.set_start_method('forkserver')
    # call scikit-learn utils with n_jobs > 1 here
```

You can find more default on the new start methods in the multiprocessing documentation.

1.2.17 Why does my job use more cores than specified with n_jobs under OSX or Linux?

This happens when vectorized numpy operations are handled by libraries such as MKL or OpenBLAS.

While scikit-learn adheres to the limit set by `n_jobs`, numpy operations vectorized using MKL (or OpenBLAS) will make use of multiple threads within each scikit-learn job (thread or process).

The number of threads used by the BLAS library can be set via an environment variable. For example, to set the maximum number of threads to some integer value `N`, the following environment variables should be set:

- For MKL: `export MKL_NUM_THREADS=N`
- For OpenBLAS: `export OPENBLAS_NUM_THREADS=N`

1.2.18 Why is there no support for deep or reinforcement learning / Will there be support for deep or reinforcement learning in scikit-learn?

Deep learning and reinforcement learning both require a rich vocabulary to define an architecture, with deep learning additionally requiring GPUs for efficient computing. However, neither of these fit within the design constraints of scikit-learn; as a result, deep learning and reinforcement learning are currently out of scope for what scikit-learn seeks to achieve.

You can find more information about addition of gpu support at *Will you add GPU support?*.

1.2.19 Why is my pull request not getting any attention?

The scikit-learn review process takes a significant amount of time, and contributors should not be discouraged by a lack of activity or review on their pull request. We care a lot about getting things right the first time, as maintenance and later change comes at a high cost. We rarely release any “experimental” code, so all of our contributions will be subject to high use immediately and should be of the highest quality possible initially.

Beyond that, scikit-learn is limited in its reviewing bandwidth; many of the reviewers and core developers are working on scikit-learn on their own time. If a review of your pull request comes slowly, it is likely because the reviewers are busy. We ask for your understanding and request that you not close your pull request or discontinue your work solely because of this reason.

1.2.20 How do I set a `random_state` for an entire execution?

For testing and replicability, it is often important to have the entire execution controlled by a single seed for the pseudo-random number generator used in algorithms that have a randomized component. Scikit-learn does not use its own global random state; whenever a RandomState instance or an integer random seed is not provided as an argument, it relies on the numpy global random state, which can be set using `numpy.random.seed`. For example, to set an execution’s numpy global random state to 42, one could execute the following in his or her script:

```
import numpy as np
np.random.seed(42)
```

However, a global random state is prone to modification by other code during execution. Thus, the only way to ensure replicability is to pass `RandomState` instances everywhere and ensure that both estimators and cross-validation splitters have their `random_state` parameter set.

1.2.21 Why do categorical variables need preprocessing in scikit-learn, compared to other tools?

Most of scikit-learn assumes data is in NumPy arrays or SciPy sparse matrices of a single numeric dtype. These do not explicitly represent categorical variables at present. Thus, unlike R's data.frames or pandas.DataFrame, we require explicit conversion of categorical features to numeric values, as discussed in *Encoding categorical features*. See also *Column Transformer with Mixed Types* for an example of working with heterogeneous (e.g. categorical and numeric) data.

1.2.22 Why does Scikit-learn not directly work with, for example, pandas.DataFrame?

The homogeneous NumPy and SciPy data objects currently expected are most efficient to process for most operations. Extensive work would also be needed to support Pandas categorical types. Restricting input to homogeneous types therefore reduces maintenance cost and encourages usage of efficient data structures.

1.3 Support

There are several ways to get in touch with the developers.

1.3.1 Mailing List

- The main mailing list is scikit-learn.
- There is also a commit list scikit-learn-commits, where updates to the main repository and test failures get notified.

1.3.2 User questions

- Some scikit-learn developers support users on StackOverflow using the [scikit-learn] tag.
- For general theoretical or methodological Machine Learning questions stack exchange is probably a more suitable venue.

In both cases please use a descriptive question in the title field (e.g. no "Please help with scikit-learn!" as this is not a question) and put details on what you tried to achieve, what were the expected results and what you observed instead in the details field.

Code and data snippets are welcome. Minimalistic (up to ~20 lines long) reproduction script very helpful.

Please describe the nature of your data and the how you preprocessed it: what is the number of samples, what is the number and type of features (i.d. categorical or numerical) and for supervised learning tasks, what target are your trying to predict: binary, multiclass (1 out of `n_classes`) or multilabel (`k` out of `n_classes`) classification or continuous variable regression.

1.3.3 Bug tracker

If you think you've encountered a bug, please report it to the issue tracker:

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

Don't forget to include:

- steps (or better script) to reproduce,
- expected outcome,
- observed outcome or python (or gdb) tracebacks

To help developers fix your bug faster, please link to a https://gist.github.com holding a standalone minimalistic python script that reproduces your bug and optionally a minimalistic subsample of your dataset (for instance exported as CSV files using `numpy.savetxt`).

Note: gists are git cloneable repositories and thus you can use git to push datafiles to them.

1.3.4 IRC

Some developers like to hang out on channel `#scikit-learn` on `irc.freenode.net`.

If you do not have an IRC client or are behind a firewall this web client works fine: https://webchat.freenode.net

1.3.5 Documentation resources

This documentation is relative to 0.21.3. Documentation for other versions can be found here.

Printable pdf documentation for old versions can be found here.

1.4 Related Projects

Projects implementing the scikit-learn estimator API are encouraged to use the scikit-learn-contrib template which facilitates best practices for testing and documenting estimators. The scikit-learn-contrib GitHub organisation also accepts high-quality contributions of repositories conforming to this template.

Below is a list of sister-projects, extensions and domain specific packages.

1.4.1 Interoperability and framework enhancements

These tools adapt scikit-learn for use with other technologies or otherwise enhance the functionality of scikit-learn's estimators.

Data formats

- sklearn_pandas bridge for scikit-learn pipelines and pandas data frame with dedicated transformers.
- sklearn_xarray provides compatibility of scikit-learn estimators with xarray data structures.

Auto-ML

- auto_ml Automated machine learning for production and analytics, built on scikit-learn and related projects. Trains a pipeline wth all the standard machine learning steps. Tuned for prediction speed and ease of transfer to production environments.
- auto-sklearn An automated machine learning toolkit and a drop-in replacement for a scikit-learn estimator

- TPOT An automated machine learning toolkit that optimizes a series of scikit-learn operators to design a machine learning pipeline, including data and feature preprocessors as well as the estimators. Works as a drop-in replacement for a scikit-learn estimator.
- scikit-optimize A library to minimize (very) expensive and noisy black-box functions. It implements several methods for sequential model-based optimization, and includes a replacement for `GridSearchCV` or `RandomizedSearchCV` to do cross-validated parameter search using any of these strategies.

Experimentation frameworks

- REP Environment for conducting data-driven research in a consistent and reproducible way
- ML Frontend provides dataset management and SVM fitting/prediction through web-based and programmatic interfaces.
- Scikit-Learn Laboratory A command-line wrapper around scikit-learn that makes it easy to run machine learning experiments with multiple learners and large feature sets.
- Xcessiv is a notebook-like application for quick, scalable, and automated hyperparameter tuning and stacked ensembling. Provides a framework for keeping track of model-hyperparameter combinations.

Model inspection and visualisation

- eli5 A library for debugging/inspecting machine learning models and explaining their predictions.
- mlxtend Includes model visualization utilities.
- scikit-plot A visualization library for quick and easy generation of common plots in data analysis and machine learning.
- yellowbrick A suite of custom matplotlib visualizers for scikit-learn estimators to support visual feature analysis, model selection, evaluation, and diagnostics.

Model export for production

- onnxmltools Serializes many Scikit-learn pipelines to ONNX for interchange and prediction.
- sklearn2pmml Serialization of a wide variety of scikit-learn estimators and transformers into PMML with the help of JPMML-SkLearn library.
- sklearn-porter Transpile trained scikit-learn models to C, Java, Javascript and others.
- sklearn-compiledtrees Generate a C++ implementation of the predict function for decision trees (and ensembles) trained by sklearn. Useful for latency-sensitive production environments.

1.4.2 Other estimators and tasks

Not everything belongs or is mature enough for the central scikit-learn project. The following are projects providing interfaces similar to scikit-learn for additional learning algorithms, infrastructures and tasks.

Structured learning

- Seqlearn Sequence classification using HMMs or structured perceptron.
- HMMLearn Implementation of hidden markov models that was previously part of scikit-learn.
- PyStruct General conditional random fields and structured prediction.
- pomegranate Probabilistic modelling for Python, with an emphasis on hidden Markov models.
- sklearn-crfsuite Linear-chain conditional random fields (CRFsuite wrapper with sklearn-like API).

Deep neural networks etc.

- pylearn2 A deep learning and neural network library build on theano with scikit-learn like interface.

- sklearn_theano scikit-learn compatible estimators, transformers, and datasets which use Theano internally
- nolearn A number of wrappers and abstractions around existing neural network libraries
- keras Deep Learning library capable of running on top of either TensorFlow or Theano.
- lasagne A lightweight library to build and train neural networks in Theano.
- skorch A scikit-learn compatible neural network library that wraps PyTorch.

Broad scope

- mlxtend Includes a number of additional estimators as well as model visualization utilities.
- sparkit-learn Scikit-learn API and functionality for PySpark's distributed modelling.

Other regression and classification

- xgboost Optimised gradient boosted decision tree library.
- ML-Ensemble Generalized ensemble learning (stacking, blending, subsemble, deep ensembles, etc.).
- lightning Fast state-of-the-art linear model solvers (SDCA, AdaGrad, SVRG, SAG, etc. . .).
- py-earth Multivariate adaptive regression splines
- Kernel Regression Implementation of Nadaraya-Watson kernel regression with automatic bandwidth selection
- gplearn Genetic Programming for symbolic regression tasks.
- multiisotonic Isotonic regression on multidimensional features.
- scikit-multilearn Multi-label classification with focus on label space manipulation.
- seglearn Time series and sequence learning using sliding window segmentation.

Decomposition and clustering

- lda: Fast implementation of latent Dirichlet allocation in Cython which uses Gibbs sampling to sample from the true posterior distribution. (scikit-learn's `sklearn.decomposition.LatentDirichletAllocation` implementation uses variational inference to sample from a tractable approximation of a topic model's posterior distribution.)
- Sparse Filtering Unsupervised feature learning based on sparse-filtering
- kmodes k-modes clustering algorithm for categorical data, and several of its variations.
- hdbscan HDBSCAN and Robust Single Linkage clustering algorithms for robust variable density clustering.
- spherecluster Spherical K-means and mixture of von Mises Fisher clustering routines for data on the unit hypersphere.

Pre-processing

- categorical-encoding A library of sklearn compatible categorical variable encoders.
- imbalanced-learn Various methods to under- and over-sample datasets.

1.4.3 Statistical learning with Python

Other packages useful for data analysis and machine learning.

- Pandas Tools for working with heterogeneous and columnar data, relational queries, time series and basic statistics.
- theano A CPU/GPU array processing framework geared towards deep learning research.

- statsmodels Estimating and analysing statistical models. More focused on statistical tests and less on prediction than scikit-learn.
- PyMC Bayesian statistical models and fitting algorithms.
- Sacred Tool to help you configure, organize, log and reproduce experiments
- Seaborn Visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.
- Deep Learning A curated list of deep learning software libraries.

Domain specific packages

- scikit-image Image processing and computer vision in python.
- Natural language toolkit (nltk) Natural language processing and some machine learning.
- gensim A library for topic modelling, document indexing and similarity retrieval
- NiLearn Machine learning for neuro-imaging.
- AstroML Machine learning for astronomy.
- MSMBuilder Machine learning for protein conformational dynamics time series.
- scikit-surprise A scikit for building and evaluating recommender systems.

1.4.4 Snippets and tidbits

The wiki has more!

1.5 About us

1.5.1 History

This project was started in 2007 as a Google Summer of Code project by David Cournapeau. Later that year, Matthieu Brucher started work on this project as part of his thesis.

In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel of INRIA took leadership of the project and made the first public release, February the 1st 2010. Since then, several releases have appeared following a ~3 month cycle, and a thriving international community has been leading the development.

1.5.2 Governance

The decision making process and governance structure of scikit-learn is laid out in the *governance document*.

1.5.3 Authors

The following people are currently core contributors to scikit-learn's development and maintenance:

Please do not email the authors directly to ask for assistance or report issues. Instead, please see What's the best way to ask questions about scikit-learn in the FAQ.

See also:

How you can contribute to the project

1.5.4 Emeritus Core Developers

The following people have been active contributors in the past, but are no longer active in the project

- Alexander Fabisch
- Alexandre Passos
- Angel Soler Gollonet
- Arnaud Joly
- Chris Gorgolewski
- David Cournapeau
- David Warde-Farley
- Eduard Duchesnay
- Fabian Pedragosa
- Gilles Louppe
- Jacob Schreiber
- Jake Vanderplas
- Jaques Grobler
- Jarrod Millman
- Kyle Kastner
- Lars Buitinck
- Manoj Kumar
- Mathieu Blondel
- Matthieu Brucher
- Noel Dawe
- Paolo Losi
- Peter Prettenhofer
- Raghav Rajagopalan
- Robert Layton
- Ron Weiss
- Satrajit Ghosh
- Shiqiao Du
- Thouis (Ray) Jones
- Vincent Dubourg
- Vincent Michel
- Virgile Fritsch
- Wei Li

1.5.5 Citing scikit-learn

If you use scikit-learn in a scientific publication, we would appreciate citations to the following paper:

> Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
>
> Bibtex entry:

```
@article{scikit-learn,
 title={Scikit-learn: Machine Learning in {P}ython},
 author={Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V.
         and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P.
         and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and
         Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E.},
 journal={Journal of Machine Learning Research},
 volume={12},
 pages={2825--2830},
 year={2011}
}
```

If you want to cite scikit-learn for its API or design, you may also want to consider the following paper:

> API design for machine learning software: experiences from the scikit-learn project, Buitinck *et al.*, 2013.
>
> Bibtex entry:

```
@inproceedings{sklearn_api,
  author    = {Lars Buitinck and Gilles Louppe and Mathieu Blondel and
               Fabian Pedregosa and Andreas Mueller and Olivier Grisel and
               Vlad Niculae and Peter Prettenhofer and Alexandre Gramfort
               and Jaques Grobler and Robert Layton and Jake VanderPlas and
               Arnaud Joly and Brian Holt and Ga{\"{e}}l Varoquaux},
  title     = {{API} design for machine learning software: experiences from␣
↪the scikit-learn
               project},
  booktitle = {ECML PKDD Workshop: Languages for Data Mining and Machine␣
↪Learning},
  year      = {2013},
  pages = {108--122},
}
```

1.5.6 Artwork

High quality PNG and SVG logos are available in the doc/logos/ source directory.

1.5.7 Funding

INRIA actively supports this project. It has provided funding for Fabian Pedregosa (2010-2012), Jaques Grobler (2012-2013) and Olivier Grisel (2013-2017) to work on this project full-time. It also hosts coding sprints and other events. Paris-Saclay Center for Data Science funded one year for a developer to work on the project full-time (2014-2015) and 50% of the time of Guillaume Lemaitre (2016-2017). NYU Moore-Sloan Data Science Environment funded Andreas Mueller (2014-2016) to work on this project. The Moore-Sloan Data Science Environment also funds several students to work on the project part-time. Télécom Paristech funded Manoj Kumar (2014), Tom Dupré la Tour (2015), Raghav RV (2015-2017), Thierry Guillemot (2016-2017) and Albert Thomas (2017) to work on scikit-learn. Columbia University funds Andreas Müller since 2016. Andreas Müller also received a grant to improve scikit-learn from the Alfred P. Sloan Foundation in 2017. The University of

Sydney funds Joel Nothman since July 2017.

The Labex DigiCosme funded Nicolas Goix (2015-2016), Tom Dupré la Tour (2015-2016 and 2017-2018), Mathurin Massias (2018-2019) to work part time on scikit-learn during their PhDs. It also funded a scikit-learn coding sprint in 2015.

The following students were sponsored by Google to work on scikit-learn through the Google Summer of Code program.

- 2007 - David Cournapeau
- 2011 - Vlad Niculae
- 2012 - Vlad Niculae, Immanuel Bayer.
- 2013 - Kemal Eren, Nicolas Trésegnie
- 2014 - Hamzeh Alsalhi, Issam Laradji, Maheshakya Wijewardena, Manoj Kumar.
- 2015 - Raghav RV, Wei Xue
- 2016 - Nelson Liu, YenChen Lin

It also provided funding for sprints and events around scikit-learn. If you would like to participate in the next Google Summer of code program, please see this page.

The NeuroDebian project providing Debian packaging and contributions is supported by Dr. James V. Haxby (Dartmouth College).

The PSF helped find and manage funding for our 2011 Granada sprint. More information can be found here

tinyclues funded the 2011 international Granada sprint.

Donating to the project

If you are interested in donating to the project or to one of our code-sprints, you can use the *Paypal* button below or the NumFOCUS Donations Page (if you use the latter, please indicate that you are donating for the scikit-learn project).

All donations will be handled by NumFOCUS, a non-profit-organization which is managed by a board of Scipy community members. NumFOCUS's mission is to foster scientific computing software, in particular in Python. As a fiscal home of scikit-learn, it ensures that money is available when needed to keep the project funded and available while in compliance with tax regulations.

The received donations for the scikit-learn project mostly will go towards covering travel-expenses for code sprints, as well as towards the organization budget of the project[1].

[1] Regarding the organization budget in particular, we might use some of the donated funds to pay for other project expenses such as DNS, hosting or continuous integration services.

Notes

The 2013 Paris international sprint

Fig. 1.1: IAP VII/19 - DYSCO

For more information on this sprint, see here

1.5.8 Infrastructure support

- We would like to thank Rackspace for providing us with a free Rackspace Cloud account to automatically build the documentation and the example gallery from for the development version of scikit-learn using this tool.
- We would also like to thank Shining Panda for free CPU time on their Continuous Integration server.

1.6 Who is using scikit-learn?

1.6.1 J.P.Morgan

Scikit-learn is an indispensable part of the Python machine learning toolkit at JPMorgan. It is very widely used across all parts of the bank for classification, predictive analytics, and very many other machine learning tasks. Its straightforward API, its breadth of algorithms, and the quality of its documentation combine to make scikit-learn simultaneously very approachable and very powerful.

Stephen Simmons, VP, Athena Research, JPMorgan

1.6.2 Spotify

Scikit-learn provides a toolbox with solid implementations of a bunch of state-of-the-art models and makes it easy to plug them into existing applications. We've been using it quite a lot for music recommendations at Spotify and I think it's the most well-designed ML package I've seen so far.

Erik Bernhardsson, Engineering Manager Music Discovery & Machine Learning, Spotify

1.6.3 Inria

At INRIA, we use scikit-learn to support leading-edge basic research in many teams: Parietal for neuroimaging, Lear for computer vision, Visages for medical image analysis, Privatics for security. The project is a fantastic tool to address difficult applications of machine learning in an academic environment as it is performant and versatile, but all easy-to-use and well documented, which makes it well suited to grad students.

Gaël Varoquaux, research at Parietal

1.6.4 betaworks

Betaworks is a NYC-based startup studio that builds new products, grows companies, and invests in others. Over the past 8 years we've launched a handful of social data analytics-driven services, such as Bitly, Chartbeat, digg and Scale Model. Consistently the betaworks data science team uses Scikit-learn for a variety of tasks. From exploratory analysis, to product development, it is an essential part of our toolkit. Recent uses are included in digg's new video recommender system, and Poncho's dynamic heuristic subspace clustering.

Gilad Lotan, Chief Data Scientist

1.6.5 Hugging Face

At Hugging Face we're using NLP and probabilistic models to generate conversational Artificial intelligences that are fun to chat with. Despite using deep neural nets for a few of our NLP tasks, scikit-learn is still the bread-and-butter of our daily machine learning routine. The ease of use and predictability of the interface, as well as the straightforward mathematical explanations that are here when you need them, is the killer feature. We use a variety of scikit-learn models in production and they are also operationally very pleasant to work with.

Julien Chaumond, Chief Technology Officer

1.6.6 Evernote

Building a classifier is typically an iterative process of exploring the data, selecting the features (the attributes of the data believed to be predictive in some way), training the models, and finally evaluating them. For many of these tasks, we relied on the excellent scikit-learn package for Python.

Read more

Mark Ayzenshtat, VP, Augmented Intelligence

1.6.7 Télécom ParisTech

At Telecom ParisTech, scikit-learn is used for hands-on sessions and home assignments in introductory and advanced machine learning courses. The classes are for undergrads and masters students. The great benefit of scikit-learn is its fast learning curve that allows students to quickly start working on interesting and motivating problems.

Alexandre Gramfort, Assistant Professor

1.6.8 Booking.com

At Booking.com, we use machine learning algorithms for many different applications, such as recommending hotels and destinations to our customers, detecting fraudulent reservations, or scheduling our customer service agents. Scikit-learn is one of the tools we use when implementing standard algorithms for prediction tasks. Its API and documentations are excellent and make it easy to use. The scikit-learn developers do a great job of incorporating state of the art implementations and new algorithms into the package. Thus, scikit-learn provides convenient access to a wide spectrum of algorithms, and allows us to readily find the right tool for the right job.

Melanie Mueller, Data Scientist

1.6.9 AWeber

The scikit-learn toolkit is indispensable for the Data Analysis and Management team at AWeber. It allows us to do AWesome stuff we would not otherwise have the time or resources to accomplish. The documentation is excellent, allowing new engineers to quickly evaluate and apply many different algorithms to our data. The text feature extraction utilities are useful when working with the large volume of email content we have at AWeber. The RandomizedPCA implementation, along with Pipelining and FeatureUnions, allows us to develop complex machine learning algorithms efficiently and reliably.

Anyone interested in learning more about how AWeber deploys scikit-learn in a production environment should check out talks from PyData Boston by AWeber's Michael Becker available at https://github.com/mdbecker/pydata_2013

Michael Becker, Software Engineer, Data Analysis and Management Ninjas

1.6.10 Yhat

The combination of consistent APIs, thorough documentation, and top notch implementation make scikit-learn our favorite machine learning package in Python. scikit-learn makes doing advanced analysis in Python accessible to anyone. At Yhat, we make it easy to integrate these models into your production applications. Thus eliminating the unnecessary dev time encountered productionizing analytical work.

Greg Lamp, Co-founder Yhat

1.6.11 Rangespan

The Python scikit-learn toolkit is a core tool in the data science group at Rangespan. Its large collection of well documented models and algorithms allow our team of data scientists to prototype fast and quickly iterate to find the right solution to our learning problems. We find that scikit-learn is not only the right tool for prototyping, but its careful and well tested implementation give us the confidence to run scikit-learn models in production.

Jurgen Van Gael, Data Science Director at Rangespan Ltd

1.6.12 Birchbox

At Birchbox, we face a range of machine learning problems typical to E-commerce: product recommendation, user clustering, inventory prediction, trends detection, etc. Scikit-learn lets us experiment with many models, especially in the exploration phase of a new project: the data can be passed around in a consistent way; models are easy to save and reuse; updates keep us informed of new developments from the pattern discovery research community. Scikit-learn is an important tool for our team, built the right way in the right language.

Thierry Bertin-Mahieux, Birchbox, Data Scientist

1.6.13 Bestofmedia Group

Scikit-learn is our #1 toolkit for all things machine learning at Bestofmedia. We use it for a variety of tasks (e.g. spam fighting, ad click prediction, various ranking models) thanks to the varied, state-of-the-art algorithm implementations packaged into it. In the lab it accelerates prototyping of complex pipelines. In production I can say it has proven to be robust and efficient enough to be deployed for business critical components.

Eustache Diemert, Lead Scientist Bestofmedia Group

1.6.14 Change.org

At change.org we automate the use of scikit-learn's RandomForestClassifier in our production systems to drive email targeting that reaches millions of users across the world each week. In the lab, scikit-learn's ease-of-use, performance, and overall variety of algorithms implemented has proved invaluable in giving us a single reliable source to turn to for our machine-learning needs.

Vijay Ramesh, Software Engineer in Data/science at Change.org

1.6.15 PHIMECA Engineering

At PHIMECA Engineering, we use scikit-learn estimators as surrogates for expensive-to-evaluate numerical models (mostly but not exclusively finite-element mechanical models) for speeding up the intensive post-processing operations involved in our simulation-based decision making framework. Scikit-learn's fit/predict API together with its efficient cross-validation tools considerably eases the task of selecting the best-fit estimator. We are also using scikit-learn for illustrating concepts in our training sessions. Trainees are always impressed by the ease-of-use of scikit-learn despite the apparent theoretical complexity of machine learning.

Vincent Dubourg, PHIMECA Engineering, PhD Engineer

1.6.16 HowAboutWe

At HowAboutWe, scikit-learn lets us implement a wide array of machine learning techniques in analysis and in production, despite having a small team. We use scikit-learn's classification algorithms to predict user behavior, enabling us to (for example) estimate the value of leads from a given traffic source early in the lead's tenure on our site. Also, our users' profiles consist of primarily unstructured data (answers to open-ended questions), so we use scikit-learn's feature extraction and dimensionality reduction tools to translate these unstructured data into inputs for our matchmaking system.

Daniel Weitzenfeld, Senior Data Scientist at HowAboutWe

1.6.17 PeerIndex

At PeerIndex we use scientific methodology to build the Influence Graph - a unique dataset that allows us to identify who's really influential and in which context. To do this, we have to tackle a range of machine learning and predictive modeling problems. Scikit-learn has emerged as our primary tool for developing prototypes and making quick progress. From predicting missing data and classifying tweets to clustering communities of social media users, scikit-learn proved useful in a variety of applications. Its very intuitive interface and excellent compatibility with other python tools makes it and indispensable tool in our daily research efforts.

Ferenc Huszar - Senior Data Scientist at Peerindex

1.6.18 DataRobot

DataRobot is building next generation predictive analytics software to make data scientists more productive, and scikit-learn is an integral part of our system. The variety of machine learning techniques in combination with the solid implementations that scikit-learn offers makes it a one-stop-shopping library for machine learning in Python. Moreover, its consistent API, well-tested code and permissive licensing allow us to use it in a production environment. Scikit-learn has literally saved us years of work we would have had to do ourselves to bring our product to market.

Jeremy Achin, CEO & Co-founder DataRobot Inc.

1.6.19 OkCupid

We're using scikit-learn at OkCupid to evaluate and improve our matchmaking system. The range of features it has, especially preprocessing utilities, means we can use it for a wide variety of projects, and it's performant enough to handle the volume of data that we need to sort through. The documentation is really thorough, as well, which makes the library quite easy to use.

David Koh - Senior Data Scientist at OkCupid

1.6.20 Lovely

At Lovely, we strive to deliver the best apartment marketplace, with respect to our users and our listings. From understanding user behavior, improving data quality, and detecting fraud, scikit-learn is a regular tool for gathering insights, predictive modeling and improving our product. The easy-to-read documentation and intuitive architecture of the API makes machine learning both explorable and accessible to a wide range of python developers. I'm constantly recommending that more developers and scientists try scikit-learn.

Simon Frid - Data Scientist, Lead at Lovely

1.6.21 Data Publica

Data Publica builds a new predictive sales tool for commercial and marketing teams called C-Radar. We extensively use scikit-learn to build segmentations of customers through clustering, and to predict future customers based on past partnerships success or failure. We also categorize companies using their website communication thanks to scikit-learn and its machine learning algorithm implementations. Eventually, machine learning makes it possible to detect weak signals that traditional tools cannot see. All these complex tasks are performed in an easy and straightforward way thanks to the great quality of the scikit-learn framework.

Guillaume Lebourgeois & Samuel Charron - Data Scientists at Data Publica

1.6.22 Machinalis

Scikit-learn is the cornerstone of all the machine learning projects carried at Machinalis. It has a consistent API, a wide selection of algorithms and lots of auxiliary tools to deal with the boilerplate. We have used it in production environments on a variety of projects including click-through rate prediction, information extraction, and even counting sheep!

In fact, we use it so much that we've started to freeze our common use cases into Python packages, some of them open-sourced, like FeatureForge . Scikit-learn in one word: Awesome.

Rafael Carrascosa, Lead developer

1.6.23 solido

Scikit-learn is helping to drive Moore's Law, via Solido. Solido creates computer-aided design tools used by the majority of top-20 semiconductor companies and fabs, to design the bleeding-edge chips inside smartphones, automobiles, and more. Scikit-learn helps to power Solido's algorithms for rare-event estimation, worst-case verification, optimization, and more. At Solido, we are particularly fond of scikit-learn's libraries for Gaussian Process models, large-scale regularized linear regression, and classification. Scikit-learn has increased our productivity, because for many ML problems we no longer need to "roll our own" code. This PyData 2014 talk has details.

Trent McConaghy, founder, Solido Design Automation Inc.

1.6.24 INFONEA

We employ scikit-learn for rapid prototyping and custom-made Data Science solutions within our in-memory based Business Intelligence Software INFONEA®. As a well-documented and comprehensive collection of state-of-the-art algorithms and pipelining methods, scikit-learn enables us to provide flexible and scalable scientific analysis solutions. Thus, scikit-learn is immensely valuable in realizing a powerful integration of Data Science technology within self-service business analytics.

Thorsten Kranz, Data Scientist, Coma Soft AG.

1.6.25 Dataiku

Our software, Data Science Studio (DSS), enables users to create data services that combine ETL with Machine Learning. Our Machine Learning module integrates many scikit-learn algorithms. The scikit-learn library is a perfect integration with DSS because it offers algorithms for virtually all business cases. Our goal is to offer a transparent and flexible tool that makes it easier to optimize time consuming aspects of building a data service, preparing data, and training machine learning algorithms on all types of data.

Florian Douetteau, CEO, Dataiku

1.6.26 Otto Group

Here at Otto Group, one of global Big Five B2C online retailers, we are using scikit-learn in all aspects of our daily work from data exploration to development of machine learning application to the productive deployment of those services. It helps us to tackle machine learning problems ranging from e-commerce to logistics. It consistent APIs enabled us to build the Palladium REST-API framework around it and continuously deliver scikit-learn based services.

Christian Rammig, Head of Data Science, Otto Group

1.6.27 Zopa

At Zopa, the first ever Peer-to-Peer lending platform, we extensively use scikit-learn to run the business and optimize our users' experience. It powers our Machine Learning models involved in credit risk, fraud risk, marketing, and pricing, and has been used for originating at least 1 billion GBP worth of Zopa loans. It is very well documented, powerful, and simple to use. We are grateful for the capabilities it has provided, and for allowing us to deliver on our mission of making money simple and fair.

Vlasios Vasileiou, Head of Data Science, Zopa

1.6.28 MARS

Scikit-Learn is integral to the Machine Learning Ecosystem at Mars. Whether we're designing better recipes for petfood or closely analysing our cocoa supply chain, Scikit-Learn is used as a tool for rapidly prototyping ideas and taking them to production. This allows us to better understand and meet the needs of our consumers worldwide. Scikit-Learn's feature-rich toolset is easy to use and equips our associates with the capabilities they need to solve the business challenges they face every day.

Michael Fitzke Next Generation Technologies Sr Leader, Mars Inc.

1.7 Release History

Release notes for current and recent releases are detailed on this page, with *previous releases* linked below.

Tip: Subscribe to scikit-learn releases on libraries.io to be notified when new versions are released.

1.7.1 Legend for changelogs

- [Major Feature]: something big that you couldn't do before.
- [Feature]: something that you couldn't do before.
- [Efficiency]: an existing feature now may not require as much computation or memory.
- [Enhancement]: a miscellaneous minor improvement.
- [Fix]: something that previously didn't work as documentated – or according to reasonable expectations – should now work.
- [API Change]: you will need to change your code to have the same effect in the future; or a feature will be removed in the future.

1.8 Version 0.21.3

July 30, 2019

1.8.1 Changed models

The following estimators and functions, when fit with the same data and parameters, may produce different models from the previous version. This often occurs due to changes in the modelling logic (bug fixes or enhancements), or in random sampling procedures.

- The v0.20.0 release notes failed to mention a backwards incompatibility in *metrics.make_scorer* when needs_proba=True and y_true is binary. Now, the scorer function is supposed to accept a 1D y_pred (i.e., probability of the positive class, shape (n_samples,)), instead of a 2D y_pred (i.e., shape (n_samples, 2)).

1.8.2 Changelog

sklearn.cluster

- [FIX] Fixed a bug in *cluster.KMeans* where computation with init='random' was single threaded for n_jobs > 1 or n_jobs = -1. #12955 by Prabakaran Kumaresshan.
- [FIX] Fixed a bug in *cluster.OPTICS* where users were unable to pass float min_samples and min_cluster_size. #14496 by Fabian Klopfer and Hanmin Qin.

sklearn.compose

- [FIX] Fixed an issue in *compose.ColumnTransformer* where using DataFrames whose column order differs between :func:fit and :func:transform could lead to silently passing incorrect columns to the remainder transformer. #14237 by Andreas Schuderer.

sklearn.datasets

- [FIX] *datasets.fetch_california_housing*, *datasets.fetch_covtype*, *datasets.fetch_kddcup99*, *datasets.fetch_olivetti_faces*, *datasets.fetch_rcv1*, and *datasets.fetch_species_distributions* try to persist the previously cache using the new joblib if the cached data was persisted using the deprecated sklearn.externals.joblib. This behavior is set to be deprecated and removed in v0.23. #14197 by Adrin Jalali.

sklearn.ensemble

- [FIX] Fix zero division error in HistGradientBoostingClassifier and HistGradientBoostingRegressor. #14024 by Nicolas Hug.

sklearn.impute

- [FIX] Fixed a bug in *impute.SimpleImputer* and *impute.IterativeImputer* so that no errors are thrown when there are missing values in training data. #13974 by Frank Hoang.

sklearn.inspection

- [FIX] Fixed a bug in *inspection.plot_partial_dependence* where target parameter was not being taken into account for multiclass problems. #14393 by Guillem G. Subies.

sklearn.linear_model

- [FIX] Fixed a bug in *linear_model.LogisticRegressionCV* where refit=False would fail depending on the 'multiclass' and 'penalty' parameters (regression introduced in 0.21). #14087 by Nicolas Hug.
- [FIX] Compatibility fix for *linear_model.ARDRegression* and Scipy>=1.3.0. Adapts to upstream changes to the default pinvh cutoff threshold which otherwise results in poor accuracy in some cases. #14067 by Tim Staley.

sklearn.neighbors

- [FIX] Fixed a bug in *neighbors.NeighborhoodComponentsAnalysis* where the validation of initial parameters n_components, max_iter and tol required too strict types. #14092 by Jérémie du Boisberranger.

sklearn.tree

- [FIX] Fixed bug in *tree.export_text* when the tree has one feature and a single feature name is passed in. #14053 by Thomas Fan.
- [FIX] Fixed an issue with plot_tree where it displayed entropy calculations even for gini criterion in DecisionTreeClassifiers. #13947 by Frank Hoang.

1.9 Version 0.21.2

24 May 2019

1.9.1 Changelog

sklearn.decomposition

- [FIX] Fixed a bug in *cross_decomposition.CCA* improving numerical stability when *Y* is close to zero. #13903 by Thomas Fan.

sklearn.metrics

- [FIX] Fixed a bug in *metrics.pairwise.euclidean_distances* where a part of the distance matrix was left un-instanciated for suffiently large float32 datasets (regression introduced in 0.21). #13910 by Jérémie du Boisberranger.

sklearn.preprocessing

- [Fix] Fixed a bug in *preprocessing.OneHotEncoder* where the new drop parameter was not reflected in *get_feature_names*. #13894 by James Myatt.

sklearn.utils.sparsefuncs

- [Fix] Fixed a bug where min_max_axis would fail on 32-bit systems for certain large inputs. This affects *preprocessing.MaxAbsScaler*, *preprocessing.normalize* and *preprocessing.LabelBinarizer*. #13741 by Roddy MacSween.

1.10 Version 0.21.1

17 May 2019

This is a bug-fix release to primarily resolve some packaging issues in version 0.21.0. It also includes minor documentation improvements and some bug fixes.

1.10.1 Changelog

sklearn.inspection

- [Fix] Fixed a bug in *inspection.partial_dependence* to only check classifier and not regressor for the multiclass-multioutput case. #14309 by Guillaume Lemaitre.

sklearn.metrics

- [Fix] Fixed a bug in *metrics.pairwise_distances* where it would raise AttributeError for boolean metrics when X had a boolean dtype and Y == None. #13864 by Paresh Mathur.
- [Fix] Fixed two bugs in *metrics.pairwise_distances* when n_jobs > 1. First it used to return a distance matrix with same dtype as input, even for integer dtype. Then the diagonal was not zeros for euclidean metric when Y is X. #13877 by Jérémie du Boisberranger.

sklearn.neighbors

- [Fix] Fixed a bug in *neighbors.KernelDensity* which could not be restored from a pickle if sample_weight had been used. #13772 by Aditya Vyas.

1.11 Version 0.21.0

May 2019

1.11.1 Changed models

The following estimators and functions, when fit with the same data and parameters, may produce different models from the previous version. This often occurs due to changes in the modelling logic (bug fixes or enhancements), or in random sampling procedures.

- *discriminant_analysis.LinearDiscriminantAnalysis* for multiclass classification. [FIX]
- *discriminant_analysis.LinearDiscriminantAnalysis* with 'eigen' solver. [FIX]
- *linear_model.BayesianRidge* [FIX]
- Decision trees and derived ensembles when both `max_depth` and `max_leaf_nodes` are set. [FIX]
- *linear_model.LogisticRegression* and *linear_model.LogisticRegressionCV* with 'saga' solver. [FIX]
- *ensemble.GradientBoostingClassifier* [FIX]
- *sklearn.feature_extraction.text.HashingVectorizer*, *sklearn.feature_extraction.text.TfidfVectorizer*, and *sklearn.feature_extraction.text.CountVectorizer* [FIX]
- *neural_network.MLPClassifier* [FIX]
- *svm.SVC.decision_function* and *multiclass.OneVsOneClassifier.decision_function*. [FIX]
- *linear_model.SGDClassifier* and any derived classifiers. [FIX]
- Any model using the `linear_model.sag.sag_solver` function with a `0` seed, including *linear_model.LogisticRegression*, *linear_model.LogisticRegressionCV*, *linear_model.Ridge*, and *linear_model.RidgeCV* with 'sag' solver. [FIX]
- *linear_model.RidgeCV* when using generalized cross-validation with sparse inputs. [FIX]

Details are listed in the changelog below.

(While we are trying to better inform users by providing this information, we cannot assure that this list is complete.)

1.11.2 Known Major Bugs

- The default *max_iter* for *linear_model.LogisticRegression* is too small for many solvers given the default `tol`. In particular, we accidentally changed the default *max_iter* for the liblinear solver from 1000 to 100 iterations in #3591 released in version 0.16. In a future release we hope to choose better default *max_iter* and `tol` heuristically depending on the solver (see #13317).

1.11.3 Changelog

Support for Python 3.4 and below has been officially dropped.

sklearn.base

- [API CHANGE] The R2 score used when calling `score` on a regressor will use `multioutput='uniform_average'` from version 0.23 to keep consistent with *metrics.r2_score*. This will influence the `score` method of all the multioutput regressors (except for *multioutput.MultiOutputRegressor*). #13157 by Hanmin Qin.

sklearn.calibration

- [Enhancement] Added support to bin the data passed into *calibration.calibration_curve* by quantiles instead of uniformly between 0 and 1. #13086 by Scott Cole.
- [Enhancement] Allow n-dimensional arrays as input for *calibration.CalibratedClassifierCV*. #13485 by William de Vazelhes.

sklearn.cluster

- [Major Feature] A new clustering algorithm: *cluster.OPTICS*: an algoritm related to *cluster.DBSCAN*, that has hyperparameters easier to set and that scales better, by Shane, Adrin Jalali, Erich Schubert, Hanmin Qin, and Assia Benbihi.
- [Fix] Fixed a bug where *cluster.Birch* could occasionally raise an AttributeError. #13651 by Joel Nothman.
- [Fix] Fixed a bug in *cluster.KMeans* where empty clusters weren't correctly relocated when using sample weights. #13486 by Jérémie du Boisberranger.
- [API Change] The `n_components_` attribute in *cluster.AgglomerativeClustering* and *cluster.FeatureAgglomeration* has been renamed to `n_connected_components_`. #13427 by Stephane Couvreur.
- [Enhancement] *cluster.AgglomerativeClustering* and *cluster.FeatureAgglomeration* now accept a `distance_threshold` parameter which can be used to find the clusters instead of `n_clusters`. #9069 by Vathsala Achar and Adrin Jalali.

sklearn.compose

- [API Change] *compose.ColumnTransformer* is no longer an experimental feature. #13835 by Hanmin Qin.

sklearn.datasets

- [Fix] Added support for 64-bit group IDs and pointers in SVMLight files. #10727 by Bryan K Woods.
- [Fix] *datasets.load_sample_images* returns images with a deterministic order. #13250 by Thomas Fan.

sklearn.decomposition

- [Enhancement] *decomposition.KernelPCA* now has deterministic output (resolved sign ambiguity in eigenvalue decomposition of the kernel matrix). #13241 by Aurélien Bellet.
- [Fix] Fixed a bug in *decomposition.KernelPCA*, `fit().transform()` now produces the correct output (the same as *fit_transform()*) in case of non-removed zero eigenvalues (`remove_zero_eig=False`). `fit_inverse_transform` was also accelerated by using the same trick as *fit_transform* to compute the transform of *X*. #12143 by Sylvain Marié
- [Fix] Fixed a bug in *decomposition.NMF* where `init = 'nndsvd'`, `init = 'nndsvda'`, and `init = 'nndsvdar'` are allowed when `n_components < n_features` instead of `n_components <= min(n_samples, n_features)`. #11650 by Hossein Pourbozorg and Zijie (ZJ) Poh.

- [API CHANGE] The default value of the init argument in *decomposition.non_negative_factorization* will change from random to None in version 0.23 to make it consistent with *decomposition.NMF*. A FutureWarning is raised when the default value is used. #12988 by Zijie (ZJ) Poh.

sklearn.discriminant_analysis

- [ENHANCEMENT] *discriminant_analysis.LinearDiscriminantAnalysis* now preserves float32 and float64 dtypes. #8769 and #11000 by Thibault Sejourne
- [FIX] A ChangedBehaviourWarning is now raised when *discriminant_analysis.LinearDiscriminantAnalysis* is given as parameter n_components > min(n_features, n_classes - 1), and n_components is changed to min(n_features, n_classes - 1) if so. Previously the change was made, but silently. #11526 by William de Vazelhes.
- [FIX] Fixed a bug in *discriminant_analysis.LinearDiscriminantAnalysis* where the predicted probabilities would be incorrectly computed in the multiclass case. #6848, by Agamemnon Krasoulis and Guillaume Lemaitre.
- [FIX] Fixed a bug in *discriminant_analysis.LinearDiscriminantAnalysis* where the predicted probabilities would be incorrectly computed with eigen solver. #11727, by Agamemnon Krasoulis.

sklearn.dummy

- [FIX] Fixed a bug in *dummy.DummyClassifier* where the predict_proba method was returning int32 array instead of float64 for the stratified strategy. #13266 by Christos Aridas.
- [FIX] Fixed a bug in *dummy.DummyClassifier* where it was throwing a dimension mismatch error in prediction time if a column vector y with shape=(n, 1) was given at fit time. #13545 by Nick Sorros and Adrin Jalali.

sklearn.ensemble

- [MAJOR FEATURE] Add two new implementations of gradient boosting trees: *ensemble.HistGradientBoostingClassifier* and *ensemble.HistGradientBoostingRegressor*. The implementation of these estimators is inspired by LightGBM and can be orders of magnitude faster than *ensemble.GradientBoostingRegressor* and *ensemble.GradientBoostingClassifier* when the number of samples is larger than tens of thousands of samples. The API of these new estimators is slightly different, and some of the features from *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor* are not yet supported.

  These new estimators are experimental, which means that their results or their API might change without any deprecation cycle. To use them, you need to explicitly import enable_hist_gradient_boosting:

  ```
  >>> # explicitly require this experimental feature
  >>> from sklearn.experimental import enable_hist_gradient_boosting  # noqa
  >>> # now you can import normally from sklearn.ensemble
  >>> from sklearn.ensemble import HistGradientBoostingClassifier
  ```

  #12807 by Nicolas Hug.

- [FEATURE] Add *ensemble.VotingRegressor* which provides an equivalent of *ensemble.VotingClassifier* for regression problems. #12513 by Ramil Nugmanov and Mohamed Ali Jamaoui.

- [Efficiency] Make *ensemble.IsolationForest* prefer threads over processes when running with `n_jobs > 1` as the underlying decision tree fit calls do release the GIL. This changes reduces memory usage and communication overhead. #12543 by Isaac Storch and Olivier Grisel.
- [Efficiency] Make *ensemble.IsolationForest* more memory efficient by avoiding keeping in memory each tree prediction. #13260 by Nicolas Goix.
- [Efficiency] *ensemble.IsolationForest* now uses chunks of data at prediction step, thus capping the memory usage. #13283 by Nicolas Goix.
- [Efficiency] *sklearn.ensemble.GradientBoostingClassifier* and *sklearn.ensemble.GradientBoostingRegressor* now keep the input `y` as `float64` to avoid it being copied internally by trees. #13524 by Adrin Jalali.
- [Enhancement] Minimized the validation of X in *ensemble.AdaBoostClassifier* and *ensemble.AdaBoostRegressor* #13174 by Christos Aridas.
- [Enhancement] *ensemble.IsolationForest* now exposes `warm_start` parameter, allowing iterative addition of trees to an isolation forest. #13496 by Peter Marko.
- [Fix] The values of `feature_importances_` in all random forest based models (i.e. *ensemble.RandomForestClassifier*, *ensemble.RandomForestRegressor*, *ensemble.ExtraTreesClassifier*, *ensemble.ExtraTreesRegressor*, *ensemble.RandomTreesEmbedding*, *ensemble.GradientBoostingClassifier*, and *ensemble.GradientBoostingRegressor*) now:
  - sum up to `1`
  - all the single node trees in feature importance calculation are ignored
  - in case all trees have only one single node (i.e. a root node), feature importances will be an array of all zeros.

  #13636 and #13620 by Adrin Jalali.
- [Fix] Fixed a bug in *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor*, which didn't support scikit-learn estimators as the initial estimator. Also added support of initial estimator which does not support sample weights. #12436 by Jérémie du Boisberranger and #12983 by Nicolas Hug.
- [Fix] Fixed the output of the average path length computed in *ensemble.IsolationForest* when the input is either 0, 1 or 2. #13251 by Albert Thomas and joshuakennethjones.
- [Fix] Fixed a bug in *ensemble.GradientBoostingClassifier* where the gradients would be incorrectly computed in multiclass classification problems. #12715 by Nicolas Hug.
- [Fix] Fixed a bug in *ensemble.GradientBoostingClassifier* where validation sets for early stopping were not sampled with stratification. #13164 by Nicolas Hug.
- [Fix] Fixed a bug in *ensemble.GradientBoostingClassifier* where the default initial prediction of a multiclass classifier would predict the classes priors instead of the log of the priors. #12983 by Nicolas Hug.
- [Fix] Fixed a bug in *ensemble.RandomForestClassifier* where the `predict` method would error for multiclass multioutput forests models if any targets were strings. #12834 by Elizabeth Sander.
- [Fix] Fixed a bug in `ensemble.gradient_boosting.LossFunction` and `ensemble.gradient_boosting.LeastSquaresError` where the default value of `learning_rate` in `update_terminal_regions` is not consistent with the document and the caller functions. Note however that directly using these loss functions is deprecated. #6463 by movelikeriver.
- [Fix] `ensemble.partial_dependence` (and consequently the new version *sklearn.inspection.partial_dependence*) now takes sample weights into account for the partial dependence computation when the gradient boosting model has been trained with sample weights. #13193 by Samuel O. Ronsin.

- [API Change] ensemble.partial_dependence and ensemble.plot_partial_dependence are now deprecated in favor of *inspection.partial_dependence* and *inspection.plot_partial_dependence*. #12599 by Trevor Stephens and Nicolas Hug.
- [Fix] *ensemble.VotingClassifier* and *ensemble.VotingRegressor* were failing during fit in one of the estimators was set to None and sample_weight was not None. #13779 by Guillaume Lemaitre.
- [API Change] *ensemble.VotingClassifier* and *ensemble.VotingRegressor* accept 'drop' to disable an estimator in addition to None to be consistent with other estimators (i.e., *pipeline.FeatureUnion* and *compose.ColumnTransformer*). #13780 by Guillaume Lemaitre.

sklearn.externals

- [API Change] Deprecated externals.six since we have dropped support for Python 2.7. #12916 by Hanmin Qin.

sklearn.feature_extraction

- [Fix] If input='file' or input='filename', and a callable is given as the analyzer, *sklearn.feature_extraction.text.HashingVectorizer*, *sklearn.feature_extraction.text.TfidfVectorizer*, and *sklearn.feature_extraction.text.CountVectorizer* now read the data from the file(s) and then pass it to the given analyzer, instead of passing the file name(s) or the file object(s) to the analyzer. #13641 by Adrin Jalali.

sklearn.impute

- [Major Feature] Added *impute.IterativeImputer*, which is a strategy for imputing missing values by modeling each feature with missing values as a function of other features in a round-robin fashion. #8478 and #12177 by Sergey Feldman and Ben Lawson.

  The API of IterativeImputer is experimental and subject to change without any deprecation cycle. To use them, you need to explicitly import enable_iterative_imputer:

  ```
  >>> from sklearn.experimental import enable_iterative_imputer  # noqa
  >>> # now you can import normally from sklearn.impute
  >>> from sklearn.impute import IterativeImputer
  ```

- [Feature] The *impute.SimpleImputer* and *impute.IterativeImputer* have a new parameter 'add_indicator', which simply stacks a *impute.MissingIndicator* transform into the output of the imputer's transform. That allows a predictive estimator to account for missingness. #12583, #13601 by Danylo Baibak.
- [Fix] In *impute.MissingIndicator* avoid implicit densification by raising an exception if input is sparse add missing_values property is set to 0. #13240 by Bartosz Telenczuk.
- [Fix] Fixed two bugs in *impute.MissingIndicator*. First, when X is sparse, all the non-zero non missing values used to become explicit False in the transformed data. Then, when features='missing-only', all features used to be kept if there were no missing values at all. #13562 by Jérémie du Boisberranger.

sklearn.inspection

(new subpackage)

- [FEATURE] Partial dependence plots (*inspection.plot_partial_dependence*) are now supported for any regressor or classifier (provided that they have a *predict_proba* method). #12599 by Trevor Stephens and Nicolas Hug.

sklearn.isotonic

- [FEATURE] Allow different dtypes (such as float32) in *isotonic.IsotonicRegression*. #8769 by Vlad Niculae

sklearn.linear_model

- [ENHANCEMENT] *linear_model.Ridge* now preserves `float32` and `float64` dtypes. #8769 and #11000 by Guillaume Lemaitre, and Joan Massich
- [FEATURE] *linear_model.LogisticRegression* and *linear_model.LogisticRegressionCV* now support Elastic-Net penalty, with the 'saga' solver. #11646 by Nicolas Hug.
- [FEATURE] Added *linear_model.lars_path_gram*, which is *linear_model.lars_path* in the sufficient stats mode, allowing users to compute *linear_model.lars_path* without providing `X` and `y`. #11699 by Kuai Yu.
- [EFFICIENCY] `linear_model.make_dataset` now preserves `float32` and `float64` dtypes, reducing memory consumption in stochastic gradient, SAG and SAGA solvers. #8769 and #11000 by Nelle Varoquaux, Arthur Imbert, Guillaume Lemaitre, and Joan Massich
- [ENHANCEMENT] *linear_model.LogisticRegression* now supports an unregularized objective when `penalty='none'` is passed. This is equivalent to setting `C=np.inf` with l2 regularization. Not supported by the liblinear solver. #12860 by Nicolas Hug.
- [ENHANCEMENT] `sparse_cg` solver in *linear_model.Ridge* now supports fitting the intercept (i.e. `fit_intercept=True`) when inputs are sparse. #13336 by Bartosz Telenczuk.
- [ENHANCEMENT] The coordinate descent solver used in *Lasso*, *ElasticNet*, etc. now issues a *ConvergenceWarning* when it completes without meeting the desired toleranbce. #11754 and #13397 by Brent Fagan and Adrin Jalali.
- [FIX] Fixed a bug in *linear_model.LogisticRegression* and *linear_model.LogisticRegressionCV* with 'saga' solver, where the weights would not be correctly updated in some cases. #11646 by Tom Dupre la Tour.
- [FIX] Fixed the posterior mean, posterior covariance and returned regularization parameters in *linear_model.BayesianRidge*. The posterior mean and the posterior covariance were not the ones computed with the last update of the regularization parameters and the returned regularization parameters were not the final ones. Also fixed the formula of the log marginal likelihood used to compute the score when `compute_score=True`. #12174 by Albert Thomas.
- [FIX] Fixed a bug in *linear_model.LassoLarsIC*, where user input `copy_X=False` at instance creation would be overridden by default parameter value `copy_X=True` in `fit`. #12972 by Lucio Fernandez-Arjona
- [FIX] Fixed a bug in *linear_model.LinearRegression* that was not returning the same coeffecients and intercepts with `fit_intercept=True` in sparse and dense case. #13279 by Alexandre Gramfort
- [FIX] Fixed a bug in *linear_model.HuberRegressor* that was broken when `X` was of dtype bool. #13328 by Alexandre Gramfort.

- [FIX] Fixed a performance issue of saga and sag solvers when called in a joblib.Parallel setting with n_jobs > 1 and backend="threading", causing them to perform worse than in the sequential case. #13389 by Pierre Glaser.
- [FIX] Fixed a bug in linear_model.stochastic_gradient.BaseSGDClassifier that was not deterministic when trained in a multi-class setting on several threads. #13422 by Clément Doumouro.
- [FIX] Fixed bug in *linear_model.ridge_regression*, *linear_model.Ridge* and *linear_model.RidgeClassifier* that caused unhandled exception for arguments return_intercept=True and solver=auto (default) or any other solver different from sag. #13363 by Bartosz Telenczuk
- [FIX] *linear_model.ridge_regression* will now raise an exception if return_intercept=True and solver is different from sag. Previously, only warning was issued. #13363 by Bartosz Telenczuk
- [FIX] *linear_model.ridge_regression* will choose sparse_cg solver for sparse inputs when solver=auto and sample_weight is provided (previously cholesky solver was selected). #13363 by Bartosz Telenczuk
- [API CHANGE] The use of *linear_model.lars_path* with X=None while passing Gram is deprecated in version 0.21 and will be removed in version 0.23. Use *linear_model.lars_path_gram* instead. #11699 by Kuai Yu.
- [API CHANGE] *linear_model.logistic_regression_path* is deprecated in version 0.21 and will be removed in version 0.23. #12821 by Nicolas Hug.
- [FIX] *linear_model.RidgeCV* with generalized cross-validation now correctly fits an intercept when fit_intercept=True and the design matrix is sparse. #13350 by Jérôme Dockès

sklearn.manifold

- [EFFICIENCY] Make manifold.tsne.trustworthiness use an inverted index instead of an np.where lookup to find the rank of neighbors in the input space. This improves efficiency in particular when computed with lots of neighbors and/or small datasets. #9907 by William de Vazelhes.

sklearn.metrics

- [FEATURE] Added the *metrics.max_error* metric and a corresponding 'max_error' scorer for single output regression. #12232 by Krishna Sangeeth.
- [FEATURE] Add *metrics.multilabel_confusion_matrix*, which calculates a confusion matrix with true positive, false positive, false negative and true negative counts for each class. This facilitates the calculation of set-wise metrics such as recall, specificity, fall out and miss rate. #11179 by Shangwu Yao and Joel Nothman.
- [FEATURE] *metrics.jaccard_score* has been added to calculate the Jaccard coefficient as an evaluation metric for binary, multilabel and multiclass tasks, with an interface analogous to *metrics.f1_score*. #13151 by Gaurav Dhingra and Joel Nothman.
- [FEATURE] Added *metrics.pairwise.haversine_distances* which can be accessed with metric='pairwise' through *metrics.pairwise_distances* and estimators. (Haversine distance was previously available for nearest neighbors calculation.) #12568 by Wei Xue, Emmanuel Arias and Joel Nothman.
- [EFFICIENCY] Faster *metrics.pairwise_distances* with *n_jobs* > 1 by using a thread-based backend, instead of process-based backends. #8216 by Pierre Glaser and Romuald Menuet
- [EFFICIENCY] The pairwise manhattan distances with sparse input now uses the BLAS shipped with scipy instead of the bundled BLAS. #12732 by Jérémie du Boisberranger

- [Enhancement] Use label `accuracy` instead of `micro-average` on *metrics.classification_report* to avoid confusion. `micro-average` is only shown for multi-label or multi-class with a subset of classes because it is otherwise identical to accuracy. #12334 by Emmanuel Arias, Joel Nothman and Andreas Müller
- [Enhancement] Added `beta` parameter to *metrics.homogeneity_completeness_v_measure* and *metrics.v_measure_score* to configure the tradeoff between homogeneity and completeness. #13607 by Stephane Couvreur and and Ivan Sanchez.
- [Fix] The metric *metrics.r2_score* is degenerate with a single sample and now it returns NaN and raises *exceptions.UndefinedMetricWarning*. #12855 by Pawel Sendyk.
- [Fix] Fixed a bug where *metrics.brier_score_loss* will sometimes return incorrect result when there's only one class in `y_true`. #13628 by Hanmin Qin.
- [Fix] Fixed a bug in *metrics.label_ranking_average_precision_score* where sample_weight wasn't taken into account for samples with degenerate labels. #13447 by Dan Ellis.
- [API Change] The parameter `labels` in *metrics.hamming_loss* is deprecated in version 0.21 and will be removed in version 0.23. #10580 by Reshama Shaikh and Sandra Mitrovic.
- [Fix] The function *metrics.pairwise.euclidean_distances*, and therefore several estimators with `metric='euclidean'`, suffered from numerical precision issues with `float32` features. Precision has been increased at the cost of a small drop of performance. #13554 by @Celelibi and Jérémie du Boisberranger.
- [API Change] *metrics.jaccard_similarity_score* is deprecated in favour of the more consistent *metrics.jaccard_score*. The former behavior for binary and multiclass targets is broken. #13151 by Joel Nothman.

sklearn.mixture

- [Fix] Fixed a bug in `mixture.BaseMixture` and therefore on estimators based on it, i.e. *mixture.GaussianMixture* and *mixture.BayesianGaussianMixture*, where `fit_predict` and `fit.predict` were not equivalent. #13142 by Jérémie du Boisberranger.

sklearn.model_selection

- [Feature] Classes *GridSearchCV* and *RandomizedSearchCV* now allow for refit=callable to add flexibility in identifying the best estimator. See *Balance model complexity and cross-validated score*. #11354 by Wenhao Zhang, Joel Nothman and Adrin Jalali.
- [Enhancement] Classes *GridSearchCV*, *RandomizedSearchCV*, and methods *cross_val_score*, *cross_val_predict*, *cross_validate*, now print train scores when `return_train_scores` is True and *verbose* > 2. For *learning_curve*, and *validation_curve* only the latter is required. #12613 and #12669 by Marc Torrellas.
- [Enhancement] Some *CV splitter* classes and *model_selection.train_test_split* now raise `ValueError` when the resulting training set is empty. #12861 by Nicolas Hug.
- [Fix] Fixed a bug where *model_selection.StratifiedKFold* shuffles each class's samples with the same `random_state`, making `shuffle=True` ineffective. #13124 by Hanmin Qin.
- [Fix] Added ability for *model_selection.cross_val_predict* to handle multi-label (and multioutput-multiclass) targets with `predict_proba`-type methods. #8773 by Stephen Hoover.
- [Fix] Fixed an issue in *cross_val_predict* where `method="predict_proba"` returned always `0.0` when one of the classes was excluded in a cross-validation fold. #13366 by Guillaume Fournier

sklearn.multiclass

- [Fix] Fixed an issue in *multiclass.OneVsOneClassifier.decision_function* where the decision_function value of a given sample was different depending on whether the decision_function was evaluated on the sample alone or on a batch containing this same sample due to the scaling used in decision_function. #10440 by Jonathan Ohayon.

sklearn.multioutput

- [Fix] Fixed a bug in *multioutput.MultiOutputClassifier* where the *predict_proba* method incorrectly checked for *predict_proba* attribute in the estimator object. #12222 by Rebekah Kim

sklearn.neighbors

- [Major Feature] Added *neighbors.NeighborhoodComponentsAnalysis* for metric learning, which implements the Neighborhood Components Analysis algorithm. #10058 by William de Vazelhes and John Chiotellis.
- [API Change] Methods in *neighbors.NearestNeighbors* : *kneighbors*, *radius_neighbors*, *kneighbors_graph*, *radius_neighbors_graph* now raise `NotFittedError`, rather than `AttributeError`, when called before `fit` #12279 by Krishna Sangeeth.

sklearn.neural_network

- [Fix] Fixed a bug in *neural_network.MLPClassifier* and *neural_network.MLPRegressor* where the option `shuffle=False` was being ignored. #12582 by Sam Waterbury.
- [Fix] Fixed a bug in *neural_network.MLPClassifier* where validation sets for early stopping were not sampled with stratification. In the multilabel case however, splits are still not stratified. #13164 by Nicolas Hug.

sklearn.pipeline

- [Feature] *pipeline.Pipeline* can now use indexing notation (e.g. `my_pipeline[0:-1]`) to extract a subsequence of steps as another Pipeline instance. A Pipeline can also be indexed directly to extract a particular step (e.g. `my_pipeline['svc']`), rather than accessing `named_steps`. #2568 by Joel Nothman.
- [Feature] Added optional parameter `verbose` in *pipeline.Pipeline*, *compose.ColumnTransformer* and *pipeline.FeatureUnion* and corresponding `make_` helpers for showing progress and timing of each step. #11364 by Baze Petrushev, Karan Desai, Joel Nothman, and Thomas Fan.
- [Enhancement] *pipeline.Pipeline* now supports using `'passthrough'` as a transformer, with the same effect as `None`. #11144 by Thomas Fan.
- [Enhancement] *pipeline.Pipeline* implements `__len__` and therefore `len(pipeline)` returns the number of steps in the pipeline. #13439 by Lakshya KD.

sklearn.preprocessing

- [Feature] *preprocessing.OneHotEncoder* now supports dropping one feature per category with a new drop parameter. #12908 by Drew Johnston.
- [Efficiency] *preprocessing.OneHotEncoder* and *preprocessing.OrdinalEncoder* now handle pandas DataFrames more efficiently. #13253 by @maikia.

- [Efficiency] Make *preprocessing.MultiLabelBinarizer* cache class mappings instead of calculating it every time on the fly. #12116 by Ekaterina Krivich and Joel Nothman.
- [Efficiency] *preprocessing.PolynomialFeatures* now supports compressed sparse row (CSR) matrices as input for degrees 2 and 3. This is typically much faster than the dense case as it scales with matrix density and expansion degree (on the order of density^degree), and is much, much faster than the compressed sparse column (CSC) case. #12197 by Andrew Nystrom.
- [Efficiency] Speed improvement in *preprocessing.PolynomialFeatures*, in the dense case. Also added a new parameter `order` which controls output order for further speed performances. #12251 by Tom Dupre la Tour.
- [Fix] Fixed the calculation overflow when using a float16 dtype with *preprocessing.StandardScaler*. #13007 by Raffaello Baluyot
- [Fix] Fixed a bug in *preprocessing.QuantileTransformer* and *preprocessing.quantile_transform* to force n_quantiles to be at most equal to n_samples. Values of n_quantiles larger than n_samples were either useless or resulting in a wrong approximation of the cumulative distribution function estimator. #13333 by Albert Thomas.
- [API Change] The default value of `copy` in *preprocessing.quantile_transform* will change from False to True in 0.23 in order to make it more consistent with the default `copy` values of other functions in `preprocessing` and prevent unexpected side effects by modifying the value of *X* inplace. #13459 by Hunter McGushion.

`sklearn.svm`

- [Fix] Fixed an issue in *svm.SVC.decision_function* when `decision_function_shape='ovr'`. The decision_function value of a given sample was different depending on whether the decision_function was evaluated on the sample alone or on a batch containing this same sample due to the scaling used in decision_function. #10440 by Jonathan Ohayon.

`sklearn.tree`

- [Feature] Decision Trees can now be plotted with matplotlib using *tree.plot_tree* without relying on the `dot` library, removing a hard-to-install dependency. #8508 by Andreas Müller.
- [Feature] Decision Trees can now be exported in a human readable textual format using *tree.export_text*. #6261 by `Giuseppe Vettigli`.
- [Feature] `get_n_leaves()` and `get_depth()` have been added to `tree.BaseDecisionTree` and consequently all estimators based on it, including *tree.DecisionTreeClassifier*, *tree.DecisionTreeRegressor*, *tree.ExtraTreeClassifier*, and *tree.ExtraTreeRegressor*. #12300 by Adrin Jalali.
- [Fix] Trees and forests did not previously *predict* multi-output classification targets with string labels, despite accepting them in *fit*. #11458 by Mitar Milutinovic.
- [Fix] Fixed an issue with `tree.BaseDecisionTree` and consequently all estimators based on it, including *tree.DecisionTreeClassifier*, *tree.DecisionTreeRegressor*, *tree.ExtraTreeClassifier*, and *tree.ExtraTreeRegressor*, where they used to exceed the given `max_depth` by 1 while expanding the tree if `max_leaf_nodes` and `max_depth` were both specified by the user. Please note that this also affects all ensemble methods using decision trees. #12344 by Adrin Jalali.

sklearn.utils

- [Feature] *utils.resample* now accepts a stratify parameter for sampling according to class distributions. #13549 by Nicolas Hug.
- [API Change] Deprecated warn_on_dtype parameter from *utils.check_array* and *utils.check_X_y*. Added explicit warning for dtype conversion in check_pairwise_arrays if the metric being passed is a pairwise boolean metric. #13382 by Prathmesh Savale.

Multiple modules

- [Major Feature] The __repr__() method of all estimators (used when calling print(estimator)) has been entirely re-written, building on Python's pretty printing standard library. All parameters are printed by default, but this can be altered with the print_changed_only option in *sklearn.set_config*. #11705 by Nicolas Hug.
- [Major Feature] Add estimators tags: these are annotations of estimators that allow programmatic inspection of their capabilities, such as sparse matrix support, supported output types and supported methods. Estimator tags also determine the tests that are run on an estimator when *check_estimator* is called. Read more in the *User Guide*. #8022 by Andreas Müller.
- [Efficiency] Memory copies are avoided when casting arrays to a different dtype in multiple estimators. #11973 by Roman Yurchak.
- [Fix] Fixed a bug in the implementation of the our_rand_r helper function that was not behaving consistently across platforms. #13422 by Madhura Parikh and Clément Doumouro.

Miscellaneous

- [Enhancement] Joblib is no longer vendored in scikit-learn, and becomes a dependency. Minimal supported version is joblib 0.11, however using version >= 0.13 is strongly recommended. #13531 by Roman Yurchak.

1.11.4 Changes to estimator checks

These changes mostly affect library developers.

- Add check_fit_idempotent to *check_estimator*, which checks that when *fit* is called twice with the same data, the ouput of *predict*, *predict_proba*, *transform*, and *decision_function* does not change. #12328 by Nicolas Hug
- Many checks can now be disabled or configured with *Estimator Tags*. #8022 by Andreas Müller.

1.11.5 Code and Documentation Contributors

Thanks to everyone who has contributed to the maintenance and improvement of the project since version 0.20, including:

adanhawth, Aditya Vyas, Adrin Jalali, Agamemnon Krasoulis, Albert Thomas, Alberto Torres, Alexandre Gramfort, amourav, Andrea Navarrete, Andreas Mueller, Andrew Nystrom, assiaben, Aurélien Bellet, Bartosz Michałowski, Bartosz Telenczuk, bauks, BenjaStudio, bertrandhaut, Bharat Raghunathan, brentfagan, Bryan Woods, Cat Chenal, Cheuk Ting Ho, Chris Choe, Christos Aridas, Clément Doumouro, Cole Smith, Connossor, Corey Levinson, Dan Ellis, Dan Stine, Danylo Baibak, daten-kieker, Denis Kataev, Didi Bar-Zev, Dillon Gardner, Dmitry Mottl, Dmitry Vukolov, Dougal J. Sutherland, Dowon, drewmjohnston, Dror Atariah, Edward J Brown, Ekaterina Krivich, Elizabeth Sander, Emmanuel Arias, Eric Chang, Eric Larson, Erich Schubert, esvhd, Falak, Feda Curic, Federico Caselli, Frank Hoang, Fibinse Xavier', Finn O'Shea, Gabriel Marzinotto, Gabriel Vacaliuc, Gabriele Calvo, Gael Varoquaux, GauravAhlawat, Giuseppe Vettigli, Greg Gandenberger, Guillaume Fournier, Guillaume Lemaitre, Gustavo De Mari Pereira, Hanmin Qin, haroldfox, hhu-luqi, Hunter McGushion, Ian Sanders, JackLangerman, Jacopo Notarstefano, jakirkham, James Bourbeau, Jan Koch, Jan S, janvanrijn, Jarrod Millman, jdethurens, jeremiedbb, JF, joaak, Joan Massich, Joel Nothman, Jonathan Ohayon, Joris Van den Bossche, josephsalmon, Jérémie Méhault, Katrin Leinweber, ken, kms15, Koen, Kossori Aruku, Krishna Sangeeth, Kuai Yu, Kulbear, Kushal Chauhan, Kyle Jackson, Lakshya KD, Leandro Hermida, Lee Yi Jie Joel, Lily Xiong, Lisa Sarah Thomas, Loic Esteve, louib, luk-f-a, maikia, mail-liam, Manimaran, Manuel López-Ibáñez, Marc Torrellas, Marco Gaido, Marco Gorelli, MarcoGorelli, marineLM, Mark Hannel, Martin Gubri, Masstran, mathurinm, Matthew Roeschke, Max Copeland, melsyt, mferrari3, Mickaël Schoentgen, Ming Li, Mitar, Mohammad Aftab, Mohammed AbdelAal, Mohammed Ibraheem, Muhammad Hassaan Rafique, mwestt, Naoya Iijima, Nicholas Smith, Nicolas Goix, Nicolas Hug, Nikolay Shebanov, Oleksandr Pavlyk, Oliver Rausch, Olivier Grisel, Orestis, Osman, Owen Flanagan, Paul Paczuski, Pavel Soriano, pavlos kallis, Pawel Sendyk, peay, Peter, Peter Cock, Peter Hausamann, Peter Marko, Pierre Glaser, pierretallotte, Pim de Haan, Piotr Szymański, Prabakaran Kumaresshan, Pradeep Reddy Raamana, Prathmesh Savale, Pulkit Maloo, Quentin Batista, Radostin Stoyanov, Raf Baluyot, Rajdeep Dua, Ramil Nugmanov, Raúl García Calvo, Rebekah Kim, Reshama Shaikh, Rohan Lekhwani, Rohan Singh, Rohan Varma, Rohit Kapoor, Roman Feldbauer, Roman Yurchak, Romuald M, Roopam Sharma, Ryan, Rüdiger Busche, Sam Waterbury, Samuel O. Ronsin, SandroCasagrande, Scott Cole, Scott Lowe, Sebastian Raschka, Shangwu Yao, Shivam Kotwalia, Shiyu Duan, smarie, Sriharsha Hatwar, Stephen Hoover, Stephen Tierney, Stéphane Couvreur, surgan12, SylvainLan, TakingItCasual, Tashay Green, thibsej, Thomas Fan, Thomas J Fan, Thomas Moreau, Tom Dupré la Tour, Tommy, Tulio Casagrande, Umar Farouk Umar, Utkarsh Upadhyay, Vinayak Mehta, Vishaal Kapoor, Vivek Kumar, Vlad Niculae, vqean3, Wenhao Zhang, William de Vazelhes, xhan, Xing Han Lu, xinyuliu12, Yaroslav Halchenko, Zach Griffith, Zach Miller, Zayd Hammoudeh, Zhuyi Xue, Zijie (ZJ) Poh, ^__^

1.12 Version 0.20.4

July 30, 2019

This is a bug-fix release with some bug fixes applied to version 0.20.3.

1.12.1 Changelog

The bundled version of joblib was upgraded from 0.13.0 to 0.13.2.

`sklearn.cluster`

- [Fix] Fixed a bug in *`cluster.KMeans`* where KMeans++ initialisation could rarely result in an IndexError. #11756 by Joel Nothman.

`sklearn.compose`

- [Fix] Fixed an issue in *`compose.ColumnTransformer`* where using DataFrames whose column order differs between :func:`fit` and :func:`transform` could lead to silently passing incorrect columns to the `remainder` transformer. #14237 by `Andreas Schuderer`.

`sklearn.model_selection`

- [Fix] Fixed a bug where *`model_selection.StratifiedKFold`* shuffles each class's samples with the same `random_state`, making `shuffle=True` ineffective. #13124 by Hanmin Qin.

sklearn.neighbors

- [FIX] Fixed a bug in *neighbors.KernelDensity* which could not be restored from a pickle if `sample_weight` had been used. #13772 by Aditya Vyas.

1.13 Version 0.20.3

March 1, 2019

This is a bug-fix release with some minor documentation improvements and enhancements to features released in 0.20.0.

1.13.1 Changelog

sklearn.cluster

- [FIX] Fixed a bug in *cluster.KMeans* where computation was single threaded when `n_jobs > 1` or `n_jobs = -1`. #12949 by Prabakaran Kumaresshan.

sklearn.compose

- [FIX] Fixed a bug in *compose.ColumnTransformer* to handle negative indexes in the columns list of the transformers. #12946 by Pierre Tallotte.

sklearn.covariance

- [FIX] Fixed a regression in *covariance.graphical_lasso* so that the case `n_features=2` is handled correctly. #13276 by Aurélien Bellet.

sklearn.decomposition

- [FIX] Fixed a bug in *decomposition.sparse_encode* where computation was single threaded when `n_jobs > 1` or `n_jobs = -1`. #13005 by Prabakaran Kumaresshan.

sklearn.datasets

- [EFFICIENCY] *sklearn.datasets.fetch_openml* now loads data by streaming, avoiding high memory usage. #13312 by Joris Van den Bossche.

sklearn.feature_extraction

- [FIX] Fixed a bug in *feature_extraction.text.CountVectorizer* which would result in the sparse feature matrix having conflicting `indptr` and `indices` precisions under very large vocabularies. #11295 by Gabriel Vacaliuc.

sklearn.impute

- [Fix] add support for non-numeric data in *sklearn.impute.MissingIndicator* which was not supported while *sklearn.impute.SimpleImputer* was supporting this for some imputation strategies. #13046 by Guillaume Lemaitre.

sklearn.linear_model

- [Fix] Fixed a bug in *linear_model.MultiTaskElasticNet* and *linear_model.MultiTaskLasso* which were breaking when `warm_start = True`. #12360 by Aakanksha Joshi.

sklearn.preprocessing

- [Fix] Fixed a bug in *preprocessing.KBinsDiscretizer* where `strategy='kmeans'` fails with an error during transformation due to unsorted bin edges. #13134 by Sandro Casagrande.
- [Fix] Fixed a bug in *preprocessing.OneHotEncoder* where the deprecation of `categorical_features` was handled incorrectly in combination with `handle_unknown='ignore'`. #12881 by Joris Van den Bossche.
- [Fix] Bins whose width are too small (i.e., <= 1e-8) are removed with a warning in *preprocessing.KBinsDiscretizer*. #13165 by Hanmin Qin.

sklearn.svm

- [Fix] Fixed a bug in *svm.SVC*, *svm.NuSVC*, *svm.SVR*, *svm.NuSVR* and *svm.OneClassSVM* where the `scale` option of parameter `gamma` is erroneously defined as `1 / (n_features * X.std())`. It's now defined as `1 / (n_features * X.var())`. #13221 by Hanmin Qin.

1.13.2 Code and Documentation Contributors

With thanks to:

Adrin Jalali, Agamemnon Krasoulis, Albert Thomas, Andreas Mueller, Aurélien Bellet, bertrandhaut, Bharat Raghunathan, Dowon, Emmanuel Arias, Fibinse Xavier, Finn O'Shea, Gabriel Vacaliuc, Gael Varoquaux, Guillaume Lemaitre, Hanmin Qin, joaak, Joel Nothman, Joris Van den Bossche, Jérémie Méhault, kms15, Kossori Aruku, Lakshya KD, maikia, Manuel López-Ibáñez, Marco Gorelli, MarcoGorelli, mferrari3, Mickaël Schoentgen, Nicolas Hug, pavlos kallis, Pierre Glaser, pierretallotte, Prabakaran Kumaresshan, Reshama Shaikh, Rohit Kapoor, Roman Yurchak, SandroCasagrande, Tashay Green, Thomas Fan, Vishaal Kapoor, Zhuyi Xue, Zijie (ZJ) Poh

1.14 Version 0.20.2

December 20, 2018

This is a bug-fix release with some minor documentation improvements and enhancements to features released in 0.20.0.

1.14.1 Changed models

The following estimators and functions, when fit with the same data and parameters, may produce different models from the previous version. This often occurs due to changes in the modelling logic (bug fixes or enhancements), or in random sampling procedures.

- *sklearn.neighbors* when `metric=='jaccard'` (bug fix)
- use of `'seuclidean'` or `'mahalanobis'` metrics in some cases (bug fix)

1.14.2 Changelog

`sklearn.compose`

- [Fix] Fixed an issue in *compose.make_column_transformer* which raises unexpected error when columns is pandas Index or pandas Series. #12704 by Hanmin Qin.

`sklearn.metrics`

- [Fix] Fixed a bug in *metrics.pairwise_distances* and *metrics.pairwise_distances_chunked* where parameters `V` of `"seuclidean"` and `VI` of `"mahalanobis"` metrics were computed after the data was split into chunks instead of being pre-computed on whole data. #12701 by Jeremie du Boisberranger.

`sklearn.neighbors`

- [Fix] Fixed *sklearn.neighbors.DistanceMetric* jaccard distance function to return 0 when two all-zero vectors are compared. #12685 by Thomas Fan.

`sklearn.utils`

- [Fix] Calling *utils.check_array* on `pandas.Series` with categorical data, which raised an error in 0.20.0, now returns the expected output again. #12699 by Joris Van den Bossche.

1.14.3 Code and Documentation Contributors

With thanks to:

adanhawth, Adrin Jalali, Albert Thomas, Andreas Mueller, Dan Stine, Feda Curic, Hanmin Qin, Jan S, jeremiedbb, Joel Nothman, Joris Van den Bossche, josephsalmon, Katrin Leinweber, Loic Esteve, Muhammad Hassaan Rafique, Nicolas Hug, Olivier Grisel, Paul Paczuski, Reshama Shaikh, Sam Waterbury, Shivam Kotwalia, Thomas Fan

1.15 Version 0.20.1

November 21, 2018

This is a bug-fix release with some minor documentation improvements and enhancements to features released in 0.20.0. Note that we also include some API changes in this release, so you might get some extra warnings after updating from 0.20.0 to 0.20.1.

1.15.1 Changed models

The following estimators and functions, when fit with the same data and parameters, may produce different models from the previous version. This often occurs due to changes in the modelling logic (bug fixes or enhancements), or in random sampling procedures.

- *decomposition.IncrementalPCA* (bug fix)

1.15.2 Changelog

sklearn.cluster

- [EFFICIENCY] make *cluster.MeanShift* no longer try to do nested parallelism as the overhead would hurt performance significantly when `n_jobs > 1`. #12159 by Olivier Grisel.
- [FIX] Fixed a bug in *cluster.DBSCAN* with precomputed sparse neighbors graph, which would add explicitly zeros on the diagonal even when already present. #12105 by Tom Dupre la Tour.

sklearn.compose

- [FIX] Fixed an issue in *compose.ColumnTransformer* when stacking columns with types not convertible to a numeric. #11912 by Adrin Jalali.
- [API CHANGE] *compose.ColumnTransformer* now applies the `sparse_threshold` even if all transformation results are sparse. #12304 by Andreas Müller.
- [API CHANGE] *compose.make_column_transformer* now expects `(transformer, columns)` instead of `(columns, transformer)` to keep consistent with *compose.ColumnTransformer*. #12339 by Adrin Jalali.

sklearn.datasets

- [FIX] *datasets.fetch_openml* to correctly use the local cache. #12246 by Jan N. van Rijn.
- [FIX] *datasets.fetch_openml* to correctly handle ignore attributes and row id attributes. #12330 by Jan N. van Rijn.
- [FIX] Fixed integer overflow in *datasets.make_classification* for values of `n_informative` parameter larger than 64. #10811 by Roman Feldbauer.
- [FIX] Fixed olivetti faces dataset `DESCR` attribute to point to the right location in *datasets.fetch_olivetti_faces*. #12441 by Jérémie du Boisberranger
- [FIX] *datasets.fetch_openml* to retry downloading when reading from local cache fails. #12517 by Thomas Fan.

sklearn.decomposition

- [FIX] Fixed a regression in *decomposition.IncrementalPCA* where 0.20.0 raised an error if the number of samples in the final batch for fitting IncrementalPCA was smaller than n_components. #12234 by Ming Li.

sklearn.ensemble

- [Fix] Fixed a bug mostly affecting *ensemble.RandomForestClassifier* where class_weight='balanced_subsample' failed with more than 32 classes. #12165 by Joel Nothman.
- [Fix] Fixed a bug affecting *ensemble.BaggingClassifier*, *ensemble.BaggingRegressor* and *ensemble.IsolationForest*, where max_features was sometimes rounded down to zero. #12388 by Connor Tann.

sklearn.feature_extraction

- [Fix] Fixed a regression in v0.20.0 where *feature_extraction.text.CountVectorizer* and other text vectorizers could error during stop words validation with custom preprocessors or tokenizers. #12393 by Roman Yurchak.

sklearn.linear_model

- [Fix] *linear_model.SGDClassifier* and variants with early_stopping=True would not use a consistent validation split in the multiclass case and this would cause a crash when using those estimators as part of parallel parameter search or cross-validation. #12122 by Olivier Grisel.
- [Fix] Fixed a bug affecting SGDClassifier in the multiclass case. Each one-versus-all step is run in a joblib.Parallel call and mutating a common parameter, causing a segmentation fault if called within a backend using processes and not threads. We now use require=sharedmem at the joblib.Parallel instance creation. #12518 by Pierre Glaser and Olivier Grisel.

sklearn.metrics

- [Fix] Fixed a bug in metrics.pairwise.pairwise_distances_argmin_min which returned the square root of the distance when the metric parameter was set to "euclidean". #12481 by Jérémie du Boisberranger.
- [Fix] Fixed a bug in metrics.pairwise.pairwise_distances_chunked which didn't ensure the diagonal is zero for euclidean distances. #12612 by Andreas Müller.
- [API Change] The *metrics.calinski_harabaz_score* has been renamed to *metrics.calinski_harabasz_score* and will be removed in version 0.23. #12211 by Lisa Thomas, Mark Hannel and Melissa Ferrari.

sklearn.mixture

- [Fix] Ensure that the fit_predict method of *mixture.GaussianMixture* and *mixture.BayesianGaussianMixture* always yield assignments consistent with fit followed by predict even if the convergence criterion is too loose or not met. #12451 by Olivier Grisel.

sklearn.neighbors

- [Fix] force the parallelism backend to threading for *neighbors.KDTree* and *neighbors.BallTree* in Python 2.7 to avoid pickling errors caused by the serialization of their methods. #12171 by Thomas Moreau.

sklearn.preprocessing

- [Fix] Fixed bug in *preprocessing.OrdinalEncoder* when passing manually specified categories. #12365 by Joris Van den Bossche.
- [Fix] Fixed bug in *preprocessing.KBinsDiscretizer* where the transform method mutates the _encoder attribute. The transform method is now thread safe. #12514 by Hanmin Qin.
- [Fix] Fixed a bug in *preprocessing.PowerTransformer* where the Yeo-Johnson transform was incorrect for lambda parameters outside of [0, 2] #12522 by Nicolas Hug.
- [Fix] Fixed a bug in *preprocessing.OneHotEncoder* where transform failed when set to ignore unknown numpy strings of different lengths #12471 by Gabriel Marzinotto.
- [API Change] The default value of the method argument in *preprocessing.power_transform* will be changed from box-cox to yeo-johnson to match *preprocessing.PowerTransformer* in version 0.23. A FutureWarning is raised when the default value is used. #12317 by Eric Chang.

sklearn.utils

- [Fix] Use float64 for mean accumulator to avoid floating point precision issues in *preprocessing.StandardScaler* and *decomposition.IncrementalPCA* when using float32 datasets. #12338 by bauks.
- [Fix] Calling *utils.check_array* on pandas.Series, which raised an error in 0.20.0, now returns the expected output again. #12625 by Andreas Müller

Miscellaneous

- [Fix] When using site joblib by setting the environment variable SKLEARN_SITE_JOBLIB, added compatibility with joblib 0.11 in addition to 0.12+. #12350 by Joel Nothman and Roman Yurchak.
- [Fix] Make sure to avoid raising FutureWarning when calling np.vstack with numpy 1.16 and later (use list comprehensions instead of generator expressions in many locations of the scikit-learn code base). #12467 by Olivier Grisel.
- [API Change] Removed all mentions of sklearn.externals.joblib, and deprecated joblib methods exposed in sklearn.utils, except for *utils.parallel_backend* and *utils.register_parallel_backend*, which allow users to configure parallel computation in scikit-learn. Other functionalities are part of joblib. package and should be used directly, by installing it. The goal of this change is to prepare for unvendoring joblib in future version of scikit-learn. #12345 by Thomas Moreau

1.15.3 Code and Documentation Contributors

With thanks to:

^__^, Adrin Jalali, Andrea Navarrete, Andreas Mueller, bauks, BenjaStudio, Cheuk Ting Ho, Connossor, Corey Levinson, Dan Stine, daten-kieker, Denis Kataev, Dillon Gardner, Dmitry Vukolov, Dougal J. Sutherland, Edward J Brown, Eric Chang, Federico Caselli, Gabriel Marzinotto, Gael Varoquaux, GauravAhlawat, Gustavo De Mari Pereira, Hanmin Qin, haroldfox, JackLangerman, Jacopo Notarstefano, janvanrijn, jdethurens, jeremiedbb, Joel Nothman, Joris Van den Bossche, Koen, Kushal Chauhan, Lee Yi Jie Joel, Lily Xiong, mail-liam, Mark Hannel, melsyt, Ming Li, Nicholas Smith, Nicolas Hug, Nikolay Shebanov, Oleksandr Pavlyk, Olivier Grisel, Peter Hausamann, Pierre Glaser, Pulkit Maloo, Quentin Batista, Radostin Stoyanov, Ramil Nugmanov, Rebekah Kim, Reshama Shaikh, Rohan Singh, Roman Feldbauer, Roman Yurchak, Roopam Sharma, Sam Waterbury, Scott Lowe, Sebastian Raschka, Stephen Tierney, SylvainLan, TakingItCasual, Thomas Fan, Thomas Moreau, Tom Dupré la Tour, Tulio Casagrande, Utkarsh Upadhyay, Xing Han Lu, Yaroslav Halchenko, Zach Miller

1.16 Version 0.20.0

September 25, 2018

This release packs in a mountain of bug fixes, features and enhancements for the Scikit-learn library, and improvements to the documentation and examples. Thanks to our contributors!

This release is dedicated to the memory of Raghav Rajagopalan.

> **Warning:** Version 0.20 is the last version of scikit-learn to support Python 2.7 and Python 3.4. Scikit-learn 0.21 will require Python 3.5 or higher.

1.16.1 Highlights

We have tried to improve our support for common data-science use-cases including missing values, categorical variables, heterogeneous data, and features/targets with unusual distributions. Missing values in features, represented by NaNs, are now accepted in column-wise preprocessing such as scalers. Each feature is fitted disregarding NaNs, and data containing NaNs can be transformed. The new `impute` module provides estimators for learning despite missing data.

`ColumnTransformer` handles the case where different features or columns of a pandas.DataFrame need different preprocessing. String or pandas Categorical columns can now be encoded with *`OneHotEncoder`* or *`OrdinalEncoder`*.

`TransformedTargetRegressor` helps when the regression target needs to be transformed to be modeled. *`PowerTransformer`* and *`KBinsDiscretizer`* join *`QuantileTransformer`* as non-linear transformations.

Beyond this, we have added *sample_weight* support to several estimators (including *`KMeans`*, *`BayesianRidge`* and *`KernelDensity`*) and improved stopping criteria in others (including *`MLPRegressor`*, *`GradientBoostingRegressor`* and *`SGDRegressor`*).

This release is also the first to be accompanied by a *Glossary of Common Terms and API Elements* developed by Joel Nothman. The glossary is a reference resource to help users and contributors become familiar with the terminology and conventions used in Scikit-learn.

Sorry if your contribution didn't make it into the highlights. There's a lot here...

1.16.2 Changed models

The following estimators and functions, when fit with the same data and parameters, may produce different models from the previous version. This often occurs due to changes in the modelling logic (bug fixes or enhancements), or in random sampling procedures.

- *`cluster.MeanShift`* (bug fix)
- *`decomposition.IncrementalPCA`* in Python 2 (bug fix)
- *`decomposition.SparsePCA`* (bug fix)
- *`ensemble.GradientBoostingClassifier`* (bug fix affecting feature importances)
- *`isotonic.IsotonicRegression`* (bug fix)
- *`linear_model.ARDRegression`* (bug fix)
- *`linear_model.LogisticRegressionCV`* (bug fix)
- *`linear_model.OrthogonalMatchingPursuit`* (bug fix)

- *linear_model.PassiveAggressiveClassifier* (bug fix)
- *linear_model.PassiveAggressiveRegressor* (bug fix)
- *linear_model.Perceptron* (bug fix)
- *linear_model.SGDClassifier* (bug fix)
- *linear_model.SGDRegressor* (bug fix)
- *metrics.roc_auc_score* (bug fix)
- *metrics.roc_curve* (bug fix)
- neural_network.BaseMultilayerPerceptron (bug fix)
- *neural_network.MLPClassifier* (bug fix)
- *neural_network.MLPRegressor* (bug fix)
- The v0.19.0 release notes failed to mention a backwards incompatibility with *model_selection.StratifiedKFold* when shuffle=True due to #7823.

Details are listed in the changelog below.

(While we are trying to better inform users by providing this information, we cannot assure that this list is complete.)

1.16.3 Known Major Bugs

- #11924: *linear_model.LogisticRegressionCV* with solver='lbfgs' and multi_class='multinomial' may be non-deterministic or otherwise broken on macOS. This appears to be the case on Travis CI servers, but has not been confirmed on personal MacBooks! This issue has been present in previous releases.
- #9354: *metrics.pairwise.euclidean_distances* (which is used several times throughout the library) gives results with poor precision, which particularly affects its use with 32-bit float inputs. This became more problematic in versions 0.18 and 0.19 when some algorithms were changed to avoid casting 32-bit data into 64-bit.

1.16.4 Changelog

Support for Python 3.3 has been officially dropped.

sklearn.cluster

- [Major Feature] *cluster.AgglomerativeClustering* now supports Single Linkage clustering via linkage='single'. #9372 by Leland McInnes and Steve Astels.
- [Feature] *cluster.KMeans* and *cluster.MiniBatchKMeans* now support sample weights via new parameter sample_weight in fit function. #10933 by Johannes Hansen.
- [Efficiency] *cluster.KMeans*, *cluster.MiniBatchKMeans* and *cluster.k_means* passed with algorithm='full' now enforces row-major ordering, improving runtime. #10471 by Gaurav Dhingra.
- [Efficiency] *cluster.DBSCAN* now is parallelized according to n_jobs regardless of algorithm. #8003 by Joël Billaud.
- [Enhancement] *cluster.KMeans* now gives a warning if the number of distinct clusters found is smaller than n_clusters. This may occur when the number of distinct points in the data set is actually smaller than the number of cluster one is looking for. #10059 by Christian Braune.

- [Fix] Fixed a bug where the fit method of *cluster.AffinityPropagation* stored cluster centers as 3d array instead of 2d array in case of non-convergence. For the same class, fixed undefined and arbitrary behavior in case of training data where all samples had equal similarity. #9612. By Jonatan Samoocha.
- [Fix] Fixed a bug in *cluster.spectral_clustering* where the normalization of the spectrum was using a division instead of a multiplication. #8129 by Jan Margeta, Guillaume Lemaitre, and Devansh D..
- [Fix] Fixed a bug in cluster.k_means_elkan where the returned iteration was 1 less than the correct value. Also added the missing n_iter_ attribute in the docstring of *cluster.KMeans*. #11353 by Jeremie du Boisberranger.
- [Fix] Fixed a bug in *cluster.mean_shift* where the assigned labels were not deterministic if there were multiple clusters with the same intensities. #11901 by Adrin Jalali.
- [API Change] Deprecate pooling_func unused parameter in *cluster.AgglomerativeClustering*. #9875 by Kumar Ashutosh.

sklearn.compose

- New module.
- [Major Feature] Added *compose.ColumnTransformer*, which allows to apply different transformers to different columns of arrays or pandas DataFrames. #9012 by Andreas Müller and Joris Van den Bossche, and #11315 by Thomas Fan.
- [Major Feature] Added the *compose.TransformedTargetRegressor* which transforms the target y before fitting a regression model. The predictions are mapped back to the original space via an inverse transform. #9041 by Andreas Müller and Guillaume Lemaitre.

sklearn.covariance

- [Efficiency] Runtime improvements to *covariance.GraphicalLasso*. #9858 by Steven Brown.
- [API Change] The *covariance.graph_lasso*, *covariance.GraphLasso* and *covariance.GraphLassoCV* have been renamed to *covariance.graphical_lasso*, *covariance.GraphicalLasso* and *covariance.GraphicalLassoCV* respectively and will be removed in version 0.22. #9993 by Artiem Krinitsyn

sklearn.datasets

- [Major Feature] Added *datasets.fetch_openml* to fetch datasets from OpenML. OpenML is a free, open data sharing platform and will be used instead of mldata as it provides better service availability. #9908 by Andreas Müller and Jan N. van Rijn.
- [Feature] In *datasets.make_blobs*, one can now pass a list to the n_samples parameter to indicate the number of samples to generate per cluster. #8617 by Maskani Filali Mohamed and Konstantinos Katrioplas.
- [Feature] Add filename attribute to datasets that have a CSV file. #9101 by alex-33 and Maskani Filali Mohamed.
- [Feature] return_X_y parameter has been added to several dataset loaders. #10774 by Chris Catalfo.
- [Fix] Fixed a bug in *datasets.load_boston* which had a wrong data point. #10795 by Takeshi Yoshizawa.
- [Fix] Fixed a bug in *datasets.load_iris* which had two wrong data points. #11082 by Sadhana Srinivasan and Hanmin Qin.

- [FIX] Fixed a bug in *datasets.fetch_kddcup99*, where data were not properly shuffled. #9731 by Nicolas Goix.
- [FIX] Fixed a bug in *datasets.make_circles*, where no odd number of data points could be generated. #10045 by Christian Braune.
- [API CHANGE] Deprecated *sklearn.datasets.fetch_mldata* to be removed in version 0.22. mldata.org is no longer operational. Until removal it will remain possible to load cached datasets. #11466 by Joel Nothman.

sklearn.decomposition

- [FEATURE] *decomposition.dict_learning* functions and models now support positivity constraints. This applies to the dictionary and sparse code. #6374 by John Kirkham.
- [FEATURE] [FIX] *decomposition.SparsePCA* now exposes `normalize_components`. When set to True, the train and test data are centered with the train mean repsectively during the fit phase and the transform phase. This fixes the behavior of SparsePCA. When set to False, which is the default, the previous abnormal behaviour still holds. The False value is for backward compatibility and should not be used. #11585 by Ivan Panico.
- [EFFICIENCY] Efficiency improvements in *decomposition.dict_learning*. #11420 and others by John Kirkham.
- [FIX] Fix for uninformative error in *decomposition.IncrementalPCA*: now an error is raised if the number of components is larger than the chosen batch size. The `n_components=None` case was adapted accordingly. #6452. By Wally Gauze.
- [FIX] Fixed a bug where the `partial_fit` method of *decomposition.IncrementalPCA* used integer division instead of float division on Python 2. #9492 by James Bourbeau.
- [FIX] In *decomposition.PCA* selecting a n_components parameter greater than the number of samples now raises an error. Similarly, the `n_components=None` case now selects the minimum of `n_samples` and `n_features`. #8484 by Wally Gauze.
- [FIX] Fixed a bug in *decomposition.PCA* where users will get unexpected error with large datasets when `n_components='mle'` on Python 3 versions. #9886 by Hanmin Qin.
- [FIX] Fixed an underflow in calculating KL-divergence for *decomposition.NMF* #10142 by Tom Dupre la Tour.
- [FIX] Fixed a bug in *decomposition.SparseCoder* when running OMP sparse coding in parallel using read-only memory mapped datastructures. #5956 by Vighnesh Birodkar and Olivier Grisel.

sklearn.discriminant_analysis

- [EFFICIENCY] Memory usage improvement for `_class_means` and `_class_cov` in `discriminant_analysis`. #10898 by Nanxin Chen.

sklearn.dummy

- [FEATURE] *dummy.DummyRegressor* now has a `return_std` option in its `predict` method. The returned standard deviations will be zeros.
- [FEATURE] *dummy.DummyClassifier* and *dummy.DummyRegressor* now only require X to be an object with finite length or shape. #9832 by Vrishank Bhardwaj.

- [FEATURE] *dummy.DummyClassifier* and *dummy.DummyRegressor* can now be scored without supplying test samples. #11951 by Rüdiger Busche.

sklearn.ensemble

- [FEATURE] *ensemble.BaggingRegressor* and *ensemble.BaggingClassifier* can now be fit with missing/non-finite values in X and/or multi-output Y to support wrapping pipelines that perform their own imputation. #9707 by Jimmy Wan.
- [FEATURE] *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor* now support early stopping via `n_iter_no_change`, `validation_fraction` and `tol`. #7071 by Raghav RV
- [FEATURE] Added `named_estimators_` parameter in *ensemble.VotingClassifier* to access fitted estimators. #9157 by Herilalaina Rakotoarison.
- [FIX] Fixed a bug when fitting *ensemble.GradientBoostingClassifier* or *ensemble.GradientBoostingRegressor* with `warm_start=True` which previously raised a segmentation fault due to a non-conversion of CSC matrix into CSR format expected by `decision_function`. Similarly, Fortran-ordered arrays are converted to C-ordered arrays in the dense case. #9991 by Guillaume Lemaitre.
- [FIX] Fixed a bug in *ensemble.GradientBoostingRegressor* and *ensemble.GradientBoostingClassifier* to have feature importances summed and then normalized, rather than normalizing on a per-tree basis. The previous behavior over-weighted the Gini importance of features that appear in later stages. This issue only affected feature importances. #11176 by Gil Forsyth.
- [API CHANGE] The default value of the `n_estimators` parameter of *ensemble.RandomForestClassifier*, *ensemble.RandomForestRegressor*, *ensemble.ExtraTreesClassifier*, *ensemble.ExtraTreesRegressor*, and *ensemble.RandomTreesEmbedding* will change from 10 in version 0.20 to 100 in 0.22. A FutureWarning is raised when the default value is used. #11542 by Anna Ayzenshtat.
- [API CHANGE] Classes derived from `ensemble.BaseBagging`. The attribute `estimators_samples_` will return a list of arrays containing the indices selected for each bootstrap instead of a list of arrays containing the mask of the samples selected for each bootstrap. Indices allows to repeat samples while mask does not allow this functionality. #9524 by Guillaume Lemaitre.
- [FIX] `ensemble.BaseBagging` where one could not deterministically reproduce `fit` result using the object attributes when `random_state` is set. #9723 by Guillaume Lemaitre.

sklearn.feature_extraction

- [FEATURE] Enable the call to *get_feature_names* in unfitted *feature_extraction.text.CountVectorizer* initialized with a vocabulary. #10908 by Mohamed Maskani.
- [ENHANCEMENT] `idf_` can now be set on a *feature_extraction.text.TfidfTransformer*. #10899 by Sergey Melderis.
- [FIX] Fixed a bug in *feature_extraction.image.extract_patches_2d* which would throw an exception if `max_patches` was greater than or equal to the number of all possible patches rather than simply returning the number of possible patches. #10101 by Varun Agrawal
- [FIX] Fixed a bug in *feature_extraction.text.CountVectorizer*, *feature_extraction.text.TfidfVectorizer*, *feature_extraction.text.HashingVectorizer* to support 64 bit sparse array indexing necessary to process large datasets with more than $2 \cdot 10^9$ tokens (words or n-grams). #9147 by Claes-Fredrik Mannby and Roman Yurchak.

- [Fix] Fixed bug in *feature_extraction.text.TfidfVectorizer* which was ignoring the parameter dtype. In addition, *feature_extraction.text.TfidfTransformer* will preserve dtype for floating and raise a warning if dtype requested is integer. #10441 by Mayur Kulkarni and Guillaume Lemaitre.

sklearn.feature_selection

- [Feature] Added select K best features functionality to *feature_selection.SelectFromModel*. #6689 by Nihar Sheth and Quazi Rahman.
- [Feature] Added min_features_to_select parameter to *feature_selection.RFECV* to bound evaluated features counts. #11293 by Brent Yi.
- [Feature] *feature_selection.RFECV*'s fit method now supports *groups*. #9656 by Adam Greenhall.
- [Fix] Fixed computation of n_features_to_compute for edge case with tied CV scores in *feature_selection.RFECV*. #9222 by Nick Hoh.

sklearn.gaussian_process

- [Efficiency] In *gaussian_process.GaussianProcessRegressor*, method predict is faster when using return_std=True in particular more when called several times in a row. #9234 by andrewww and Minghui Liu.

sklearn.impute

- New module, adopting preprocessing.Imputer as *impute.SimpleImputer* with minor changes (see under preprocessing below).
- [Major Feature] Added *impute.MissingIndicator* which generates a binary indicator for missing values. #8075 by Maniteja Nandana and Guillaume Lemaitre.
- [Feature] The *impute.SimpleImputer* has a new strategy, 'constant', to complete missing values with a fixed one, given by the fill_value parameter. This strategy supports numeric and non-numeric data, and so does the 'most_frequent' strategy now. #11211 by Jeremie du Boisberranger.

sklearn.isotonic

- [Fix] Fixed a bug in *isotonic.IsotonicRegression* which incorrectly combined weights when fitting a model to data involving points with identical X values. #9484 by Dallas Card

sklearn.linear_model

- [Feature] *linear_model.SGDClassifier*, *linear_model.SGDRegressor*, *linear_model.PassiveAggressiveClassifier*, *linear_model.PassiveAggressiveRegressor* and *linear_model.Perceptron* now expose early_stopping, validation_fraction and n_iter_no_change parameters, to stop optimization monitoring the score on a validation set. A new learning rate "adaptive" strategy divides the learning rate by 5 each time n_iter_no_change consecutive epochs fail to improve the model. #9043 by Tom Dupre la Tour.
- [Feature] Add *sample_weight* parameter to the fit method of *linear_model.BayesianRidge* for weighted linear regression. #10112 by Peter St. John.

- [FIX] Fixed a bug in logistic.logistic_regression_path to ensure that the returned coefficients are correct when multiclass='multinomial'. Previously, some of the coefficients would override each other, leading to incorrect results in *linear_model.LogisticRegressionCV*. #11724 by Nicolas Hug.
- [FIX] Fixed a bug in *linear_model.LogisticRegression* where when using the parameter multi_class='multinomial', the predict_proba method was returning incorrect probabilities in the case of binary outcomes. #9939 by Roger Westover.
- [FIX] Fixed a bug in *linear_model.LogisticRegressionCV* where the score method always computes accuracy, not the metric given by the scoring parameter. #10998 by Thomas Fan.
- [FIX] Fixed a bug in *linear_model.LogisticRegressionCV* where the 'ovr' strategy was always used to compute cross-validation scores in the multiclass setting, even if 'multinomial' was set. #8720 by William de Vazelhes.
- [FIX] Fixed a bug in *linear_model.OrthogonalMatchingPursuit* that was broken when setting normalize=False. #10071 by Alexandre Gramfort.
- [FIX] Fixed a bug in *linear_model.ARDRegression* which caused incorrectly updated estimates for the standard deviation and the coefficients. #10153 by Jörg Döpfert.
- [FIX] Fixed a bug in *linear_model.ARDRegression* and *linear_model.BayesianRidge* which caused NaN predictions when fitted with a constant target. #10095 by Jörg Döpfert.
- [FIX] Fixed a bug in *linear_model.RidgeClassifierCV* where the parameter store_cv_values was not implemented though it was documented in cv_values as a way to set up the storage of cross-validation values for different alphas. #10297 by Mabel Villalba-Jiménez.
- [FIX] Fixed a bug in *linear_model.ElasticNet* which caused the input to be overridden when using parameter copy_X=True and check_input=False. #10581 by Yacine Mazari.
- [FIX] Fixed a bug in *sklearn.linear_model.Lasso* where the coefficient had wrong shape when fit_intercept=False. #10687 by Martin Hahn.
- [FIX] Fixed a bug in *sklearn.linear_model.LogisticRegression* where the multi_class='multinomial' with binary output with warm_start=True #10836 by Aishwarya Srinivasan.
- [FIX] Fixed a bug in *linear_model.RidgeCV* where using integer alphas raised an error. #10397 by Mabel Villalba-Jiménez.
- [FIX] Fixed condition triggering gap computation in *linear_model.Lasso* and *linear_model.ElasticNet* when working with sparse matrices. #10992 by Alexandre Gramfort.
- [FIX] Fixed a bug in *linear_model.SGDClassifier*, *linear_model.SGDRegressor*, *linear_model.PassiveAggressiveClassifier*, *linear_model.PassiveAggressiveRegressor* and *linear_model.Perceptron*, where the stopping criterion was stopping the algorithm before convergence. A parameter n_iter_no_change was added and set by default to 5. Previous behavior is equivalent to setting the parameter to 1. #9043 by Tom Dupre la Tour.
- [FIX] Fixed a bug where liblinear and libsvm-based estimators would segfault if passed a scipy.sparse matrix with 64-bit indices. They now raise a ValueError. #11327 by Karan Dhingra and Joel Nothman.
- [API CHANGE] The default values of the solver and multi_class parameters of *linear_model.LogisticRegression* will change respectively from 'liblinear' and 'ovr' in version 0.20 to 'lbfgs' and 'auto' in version 0.22. A FutureWarning is raised when the default values are used. #11905 by Tom Dupre la Tour and Joel Nothman.
- [API CHANGE] Deprecate positive=True option in *linear_model.Lars* as the underlying implementation is broken. Use *linear_model.Lasso* instead. #9837 by Alexandre Gramfort.

- [API CHANGE] n_iter_ may vary from previous releases in *linear_model.LogisticRegression* with solver='lbfgs' and *linear_model.HuberRegressor*. For Scipy <= 1.0.0, the optimizer could perform more than the requested maximum number of iterations. Now both estimators will report at most max_iter iterations even if more were performed. #10723 by Joel Nothman.

sklearn.manifold

- [EFFICIENCY] Speed improvements for both 'exact' and 'barnes_hut' methods in *manifold.TSNE*. #10593 and #10610 by Tom Dupre la Tour.
- [FEATURE] Support sparse input in *manifold.Isomap.fit*. #8554 by Leland McInnes.
- [FEATURE] manifold.t_sne.trustworthiness accepts metrics other than Euclidean. #9775 by William de Vazelhes.
- [FIX] Fixed a bug in *manifold.spectral_embedding* where the normalization of the spectrum was using a division instead of a multiplication. #8129 by Jan Margeta, Guillaume Lemaitre, and Devansh D..
- [API CHANGE] [FEATURE] Deprecate precomputed parameter in function manifold.t_sne.trustworthiness. Instead, the new parameter metric should be used with any compatible metric including 'precomputed', in which case the input matrix X should be a matrix of pairwise distances or squared distances. #9775 by William de Vazelhes.
- [API CHANGE] Deprecate precomputed parameter in function manifold.t_sne.trustworthiness. Instead, the new parameter metric should be used with any compatible metric including 'precomputed', in which case the input matrix X should be a matrix of pairwise distances or squared distances. #9775 by William de Vazelhes.

sklearn.metrics

- [MAJOR FEATURE] Added the *metrics.davies_bouldin_score* metric for evaluation of clustering models without a ground truth. #10827 by Luis Osa.
- [MAJOR FEATURE] Added the *metrics.balanced_accuracy_score* metric and a corresponding 'balanced_accuracy' scorer for binary and multiclass classification. #8066 by @xyguo and Aman Dalmia, and #10587 by Joel Nothman.
- [FEATURE] Partial AUC is available via max_fpr parameter in *metrics.roc_auc_score*. #3840 by Alexander Niederbühl.
- [FEATURE] A scorer based on *metrics.brier_score_loss* is also available. #9521 by Hanmin Qin.
- [FEATURE] Added control over the normalization in *metrics.normalized_mutual_info_score* and *metrics.adjusted_mutual_info_score* via the average_method parameter. In version 0.22, the default normalizer for each will become the *arithmetic* mean of the entropies of each clustering. #11124 by Arya McCarthy.
- [FEATURE] Added output_dict parameter in *metrics.classification_report* to return classification statistics as dictionary. #11160 by Dan Barkhorn.
- [FEATURE] *metrics.classification_report* now reports all applicable averages on the given data, including micro, macro and weighted average as well as samples average for multilabel data. #11679 by Alexander Pacha.
- [FEATURE] *metrics.average_precision_score* now supports binary y_true other than {0, 1} or {-1, 1} through pos_label parameter. #9980 by Hanmin Qin.
- [FEATURE] *metrics.label_ranking_average_precision_score* now supports sample_weight. #10845 by Jose Perez-Parras Toledano.

- [Feature] Add dense_output parameter to *metrics.pairwise.linear_kernel*. When False and both inputs are sparse, will return a sparse matrix. #10999 by Taylor G Smith.
- [Efficiency] *metrics.silhouette_score* and *metrics.silhouette_samples* are more memory efficient and run faster. This avoids some reported freezes and MemoryErrors. #11135 by Joel Nothman.
- [Fix] Fixed a bug in *metrics.precision_recall_fscore_support* when truncated range(n_labels) is passed as value for labels. #10377 by Gaurav Dhingra.
- [Fix] Fixed a bug due to floating point error in *metrics.roc_auc_score* with non-integer sample weights. #9786 by Hanmin Qin.
- [Fix] Fixed a bug where *metrics.roc_curve* sometimes starts on y-axis instead of (0, 0), which is inconsistent with the document and other implementations. Note that this will not influence the result from *metrics.roc_auc_score* #10093 by alexryndin and Hanmin Qin.
- [Fix] Fixed a bug to avoid integer overflow. Casted product to 64 bits integer in *metrics.mutual_info_score*. #9772 by Kumar Ashutosh.
- [Fix] Fixed a bug where *metrics.average_precision_score* will sometimes return nan when sample_weight contains 0. #9980 by Hanmin Qin.
- [Fix] Fixed a bug in *metrics.fowlkes_mallows_score* to avoid integer overflow. Casted return value of *Contingency Matrix* to int64 and computed product of square roots rather than square root of product. #9515 by Alan Liddell and Manh Dao.
- [API Change] Deprecate reorder parameter in *metrics.auc* as it's no longer required for *metrics.roc_auc_score*. Moreover using reorder=True can hide bugs due to floating point error in the input. #9851 by Hanmin Qin.
- [API Change] In *metrics.normalized_mutual_info_score* and *metrics.adjusted_mutual_info_score*, warn that average_method will have a new default value. In version 0.22, the default normalizer for each will become the *arithmetic* mean of the entropies of each clustering. Currently, *metrics.normalized_mutual_info_score* uses the default of average_method='geometric', and *metrics.adjusted_mutual_info_score* uses the default of average_method='max' to match their behaviors in version 0.19. #11124 by Arya McCarthy.
- [API Change] The batch_size parameter to *metrics.pairwise_distances_argmin_min* and *metrics.pairwise_distances_argmin* is deprecated to be removed in v0.22. It no longer has any effect, as batch size is determined by global working_memory config. See *Limiting Working Memory*. #10280 by Joel Nothman and Aman Dalmia.

sklearn.mixture

- [Feature] Added function *fit_predict* to *mixture.GaussianMixture* and *mixture.GaussianMixture*, which is essentially equivalent to calling *fit* and *predict*. #10336 by Shu Haoran and Andrew Peng.
- [Fix] Fixed a bug in mixture.BaseMixture where the reported *n_iter_* was missing an iteration. It affected *mixture.GaussianMixture* and *mixture.BayesianGaussianMixture*. #10740 by Erich Schubert and Guillaume Lemaitre.
- [Fix] Fixed a bug in mixture.BaseMixture and its subclasses *mixture.GaussianMixture* and *mixture.BayesianGaussianMixture* where the lower_bound_ was not the max lower bound across all initializations (when n_init > 1), but just the lower bound of the last initialization. #10869 by Aurélien Géron.

sklearn.model_selection

- [FEATURE] Add return_estimator parameter in *model_selection.cross_validate* to return estimators fitted on each split. #9686 by Aurélien Bellet.
- [FEATURE] New refit_time_ attribute will be stored in *model_selection.GridSearchCV* and *model_selection.RandomizedSearchCV* if refit is set to True. This will allow measuring the complete time it takes to perform hyperparameter optimization and refitting the best model on the whole dataset. #11310 by Matthias Feurer.
- [FEATURE] Expose error_score parameter in *model_selection.cross_validate*, *model_selection.cross_val_score*, *model_selection.learning_curve* and *model_selection.validation_curve* to control the behavior triggered when an error occurs in model_selection._fit_and_score. #11576 by Samuel O. Ronsin.
- [FEATURE] BaseSearchCV now has an experimental, private interface to support customized parameter search strategies, through its _run_search method. See the implementations in *model_selection.GridSearchCV* and *model_selection.RandomizedSearchCV* and please provide feedback if you use this. Note that we do not assure the stability of this API beyond version 0.20. #9599 by Joel Nothman
- [ENHANCEMENT] Add improved error message in *model_selection.cross_val_score* when multiple metrics are passed in scoring keyword. #11006 by Ming Li.
- [API CHANGE] The default number of cross-validation folds cv and the default number of splits n_splits in the *model_selection.KFold*-like splitters will change from 3 to 5 in 0.22 as 3-fold has a lot of variance. #11557 by Alexandre Boucaud.
- [API CHANGE] The default of iid parameter of *model_selection.GridSearchCV* and *model_selection.RandomizedSearchCV* will change from True to False in version 0.22 to correspond to the standard definition of cross-validation, and the parameter will be removed in version 0.24 altogether. This parameter is of greatest practical significance where the sizes of different test sets in cross-validation were very unequal, i.e. in group-based CV strategies. #9085 by Laurent Direr and Andreas Müller.
- [API CHANGE] The default value of the error_score parameter in *model_selection.GridSearchCV* and *model_selection.RandomizedSearchCV* will change to np.NaN in version 0.22. #10677 by Kirill Zhdanovich.
- [API CHANGE] Changed ValueError exception raised in *model_selection.ParameterSampler* to a UserWarning for case where the class is instantiated with a greater value of n_iter than the total space of parameters in the parameter grid. n_iter now acts as an upper bound on iterations. #10982 by Juliet Lawton
- [API CHANGE] Invalid input for *model_selection.ParameterGrid* now raises TypeError. #10928 by Solutus Immensus

sklearn.multioutput

- [MAJOR FEATURE] Added *multioutput.RegressorChain* for multi-target regression. #9257 by Kumar Ashutosh.

sklearn.naive_bayes

- [MAJOR FEATURE] Added *naive_bayes.ComplementNB*, which implements the Complement Naive Bayes classifier described in Rennie et al. (2003). #8190 by Michael A. Alcorn.
- [FEATURE] Add var_smoothing parameter in *naive_bayes.GaussianNB* to give a precise control over variances calculation. #9681 by Dmitry Mottl.

- [FIX] Fixed a bug in *naive_bayes.GaussianNB* which incorrectly raised error for prior list which summed to 1. #10005 by Gaurav Dhingra.
- [FIX] Fixed a bug in *naive_bayes.MultinomialNB* which did not accept vector valued pseudocounts (alpha). #10346 by Tobias Madsen

sklearn.neighbors

- [EFFICIENCY] *neighbors.RadiusNeighborsRegressor* and *neighbors.RadiusNeighborsClassifier* are now parallelized according to n_jobs regardless of algorithm. #10887 by Joël Billaud.
- [EFFICIENCY] Nearest neighbors query methods are now more memory efficient when algorithm='brute'. #11136 by Joel Nothman and Aman Dalmia.
- [FEATURE] Add sample_weight parameter to the fit method of *neighbors.KernelDensity* to enable weighting in kernel density estimation. #4394 by Samuel O. Ronsin.
- [FEATURE] Novelty detection with *neighbors.LocalOutlierFactor*: Add a novelty parameter to *neighbors.LocalOutlierFactor*. When novelty is set to True, *neighbors.LocalOutlierFactor* can then be used for novelty detection, i.e. predict on new unseen data. Available prediction methods are predict, decision_function and score_samples. By default, novelty is set to False, and only the fit_predict method is avaiable. By Albert Thomas.
- [FIX] Fixed a bug in *neighbors.NearestNeighbors* where fitting a NearestNeighbors model fails when a) the distance metric used is a callable and b) the input to the NearestNeighbors model is sparse. #9579 by Thomas Kober.
- [FIX] Fixed a bug so predict in *neighbors.RadiusNeighborsRegressor* can handle empty neighbor set when using non uniform weights. Also raises a new warning when no neighbors are found for samples. #9655 by Andreas Bjerre-Nielsen.
- [FIX] [EFFICIENCY] Fixed a bug in KDTree construction that results in faster construction and querying times. #11556 by Jake VanderPlas
- [FIX] Fixed a bug in *neighbors.KDTree* and *neighbors.BallTree* where pickled tree objects would change their type to the super class BinaryTree. #11774 by Nicolas Hug.

sklearn.neural_network

- [FEATURE] Add *n_iter_no_change* parameter in neural_network.BaseMultilayerPerceptron, *neural_network.MLPRegressor*, and *neural_network.MLPClassifier* to give control over maximum number of epochs to not meet tol improvement. #9456 by Nicholas Nadeau.
- [FIX] Fixed a bug in neural_network.BaseMultilayerPerceptron, *neural_network.MLPRegressor*, and *neural_network.MLPClassifier* with new n_iter_no_change parameter now at 10 from previously hardcoded 2. #9456 by Nicholas Nadeau.
- [FIX] Fixed a bug in *neural_network.MLPRegressor* where fitting quit unexpectedly early due to local minima or fluctuations. #9456 by Nicholas Nadeau

sklearn.pipeline

- [FEATURE] The predict method of *pipeline.Pipeline* now passes keyword arguments on to the pipeline's last estimator, enabling the use of parameters such as return_std in a pipeline with caution. #9304 by Breno Freitas.

- [API CHANGE] *pipeline.FeatureUnion* now supports 'drop' as a transformer to drop features. #11144 by Thomas Fan.

sklearn.preprocessing

- [MAJOR FEATURE] Expanded *preprocessing.OneHotEncoder* to allow to encode categorical string features as a numeric array using a one-hot (or dummy) encoding scheme, and added *preprocessing.OrdinalEncoder* to convert to ordinal integers. Those two classes now handle encoding of all feature types (also handles string-valued features) and derives the categories based on the unique values in the features instead of the maximum value in the features. #9151 and #10521 by Vighnesh Birodkar and Joris Van den Bossche.
- [MAJOR FEATURE] Added *preprocessing.KBinsDiscretizer* for turning continuous features into categorical or one-hot encoded features. #7668, #9647, #10195, #10192, #11272, #11467 and #11505. by Henry Lin, Hanmin Qin, Tom Dupre la Tour and Giovanni Giuseppe Costa.
- [MAJOR FEATURE] Added *preprocessing.PowerTransformer*, which implements the Yeo-Johnson and Box-Cox power transformations. Power transformations try to find a set of feature-wise parametric transformations to approximately map data to a Gaussian distribution centered at zero and with unit variance. This is useful as a variance-stabilizing transformation in situations where normality and homoscedasticity are desirable. #10210 by Eric Chang and Maniteja Nandana, and #11520 by Nicolas Hug.
- [MAJOR FEATURE] NaN values are ignored and handled in the following preprocessing methods: *preprocessing.MaxAbsScaler*, *preprocessing.MinMaxScaler*, *preprocessing.RobustScaler*, *preprocessing.StandardScaler*, *preprocessing.PowerTransformer*, *preprocessing.QuantileTransformer* classes and *preprocessing.maxabs_scale*, *preprocessing.minmax_scale*, *preprocessing.robust_scale*, *preprocessing.scale*, *preprocessing.power_transform*, *preprocessing.quantile_transform* functions respectively addressed in issues #11011, #11005, #11308, #11206, #11306, and #10437. By Lucija Gregov and Guillaume Lemaitre.
- [FEATURE] *preprocessing.PolynomialFeatures* now supports sparse input. #10452 by Aman Dalmia and Joel Nothman.
- [FEATURE] *preprocessing.RobustScaler* and *preprocessing.robust_scale* can be fitted using sparse matrices. #11308 by Guillaume Lemaitre.
- [FEATURE] *preprocessing.OneHotEncoder* now supports the *get_feature_names* method to obtain the transformed feature names. #10181 by Nirvan Anjirbag and Joris Van den Bossche.
- [FEATURE] A parameter `check_inverse` was added to *preprocessing.FunctionTransformer* to ensure that `func` and `inverse_func` are the inverse of each other. #9399 by Guillaume Lemaitre.
- [FEATURE] The `transform` method of *sklearn.preprocessing.MultiLabelBinarizer* now ignores any unknown classes. A warning is raised stating the unknown classes classes found which are ignored. #10913 by Rodrigo Agundez.
- [FIX] Fixed bugs in *preprocessing.LabelEncoder* which would sometimes throw errors when `transform` or `inverse_transform` was called with empty arrays. #10458 by Mayur Kulkarni.
- [FIX] Fix ValueError in *preprocessing.LabelEncoder* when using `inverse_transform` on unseen labels. #9816 by Charlie Newey.
- [FIX] Fix bug in *preprocessing.OneHotEncoder* which discarded the `dtype` when returning a sparse matrix output. #11042 by Daniel Morales.
- [FIX] Fix `fit` and `partial_fit` in *preprocessing.StandardScaler* in the rare case when `with_mean=False` and `with_std=False` which was crashing by calling `fit` more than once and giving inconsistent results for `mean_` whether the input was a sparse or a dense matrix. `mean_` will be set to `None`

with both sparse and dense inputs. n_samples_seen_ will be also reported for both input types. #11235 by Guillaume Lemaitre.

- [API Change] Deprecate n_values and categorical_features parameters and active_features_, feature_indices_ and n_values_ attributes of *preprocessing.OneHotEncoder*. The n_values parameter can be replaced with the new categories parameter, and the attributes with the new categories_ attribute. Selecting the categorical features with the categorical_features parameter is now better supported using the *compose.ColumnTransformer*. #10521 by Joris Van den Bossche.
- [API Change] Deprecate *preprocessing.Imputer* and move the corresponding module to *impute.SimpleImputer*. #9726 by Kumar Ashutosh.
- [API Change] The axis parameter that was in *preprocessing.Imputer* is no longer present in *impute.SimpleImputer*. The behavior is equivalent to axis=0 (impute along columns). Row-wise imputation can be performed with FunctionTransformer (e.g., FunctionTransformer(lambda X: SimpleImputer().fit_transform(X.T).T)). #10829 by Guillaume Lemaitre and Gilberto Olimpio.
- [API Change] The NaN marker for the missing values has been changed between the *preprocessing.Imputer* and the *impute.SimpleImputer*. missing_values='NaN' should now be missing_values=np.nan. #11211 by Jeremie du Boisberranger.
- [API Change] In *preprocessing.FunctionTransformer*, the default of validate will be from True to False in 0.22. #10655 by Guillaume Lemaitre.

sklearn.svm

- [Fix] Fixed a bug in *svm.SVC* where when the argument kernel is unicode in Python2, the predict_proba method was raising an unexpected TypeError given dense inputs. #10412 by Jiongyan Zhang.
- [API Change] Deprecate random_state parameter in *svm.OneClassSVM* as the underlying implementation is not random. #9497 by Albert Thomas.
- [API Change] The default value of gamma parameter of *svm.SVC*, *NuSVC*, *SVR*, *NuSVR*, *OneClassSVM* will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. #8361 by Gaurav Dhingra and Ting Neo.

sklearn.tree

- [Enhancement] Although private (and hence not assured API stability), tree._criterion.ClassificationCriterion and tree._criterion.RegressionCriterion may now be cimported and extended. #10325 by Camil Staps.
- [Fix] Fixed a bug in tree.BaseDecisionTree with splitter="best" where split threshold could become infinite when values in X were near infinite. #10536 by Jonathan Ohayon.
- [Fix] Fixed a bug in tree.MAE to ensure sample weights are being used during the calculation of tree MAE impurity. Previous behaviour could cause suboptimal splits to be chosen since the impurity calculation considered all samples to be of equal weight importance. #11464 by John Stott.

sklearn.utils

- [Feature] *utils.check_array* and *utils.check_X_y* now have accept_large_sparse to control whether scipy.sparse matrices with 64-bit indices should be rejected. #11327 by Karan Dhingra and Joel Nothman.

- [Efficiency] [Fix] Avoid copying the data in *utils.check_array* when the input data is a memmap (and copy=False). #10663 by Arthur Mensch and Loïc Estève.
- [API Change] *utils.check_array* yield a FutureWarning indicating that arrays of bytes/strings will be interpreted as decimal numbers beginning in version 0.22. #10229 by Ryan Lee

Multiple modules

- [Feature] [API Change] More consistent outlier detection API: Add a score_samples method in *svm.OneClassSVM*, *ensemble.IsolationForest*, *neighbors.LocalOutlierFactor*, *covariance.EllipticEnvelope*. It allows to access raw score functions from original papers. A new offset_ parameter allows to link score_samples and decision_function methods. The contamination parameter of *ensemble.IsolationForest* and *neighbors.LocalOutlierFactor* decision_function methods is used to define this offset_ such that outliers (resp. inliers) have negative (resp. positive) decision_function values. By default, contamination is kept unchanged to 0.1 for a deprecation period. In 0.22, it will be set to "auto", thus using method-specific score offsets. In *covariance.EllipticEnvelope* decision_function method, the raw_values parameter is deprecated as the shifted Mahalanobis distance will be always returned in 0.22. #9015 by Nicolas Goix.
- [Feature] [API Change] A behaviour parameter has been introduced in *ensemble.IsolationForest* to ensure backward compatibility. In the old behaviour, the decision_function is independent of the contamination parameter. A threshold attribute depending on the contamination parameter is thus used. In the new behaviour the decision_function is dependent on the contamination parameter, in such a way that 0 becomes its natural threshold to detect outliers. Setting behaviour to "old" is deprecated and will not be possible in version 0.22. Beside, the behaviour parameter will be removed in 0.24. #11553 by Nicolas Goix.
- [API Change] Added convergence warning to *svm.LinearSVC* and *linear_model.LogisticRegression* when verbose is set to 0. #10881 by Alexandre Sevin.
- [API Change] Changed warning type from UserWarning to *exceptions.ConvergenceWarning* for failing convergence in *linear_model.logistic_regression_path*, *linear_model.RANSACRegressor*, *linear_model.ridge_regression*, *gaussian_process.GaussianProcessRegressor*, *gaussian_process.GaussianProcessClassifier*, *decomposition.fastica*, *cross_decomposition.PLSCanonical*, *cluster.AffinityPropagation*, and *cluster.Birch*. #10306 by Jonathan Siebert.

Miscellaneous

- [Major Feature] A new configuration parameter, working_memory was added to control memory consumption limits in chunked operations, such as the new *metrics.pairwise_distances_chunked*. See *Limiting Working Memory*. #10280 by Joel Nothman and Aman Dalmia.
- [Feature] The version of joblib bundled with Scikit-learn is now 0.12. This uses a new default multiprocessing implementation, named loky. While this may incur some memory and communication overhead, it should provide greater cross-platform stability than relying on Python standard library multiprocessing. #11741 by the Joblib developers, especially Thomas Moreau and Olivier Grisel.
- [Feature] An environment variable to use the site joblib instead of the vendored one was added (*Environment variables*). The main API of joblib is now exposed in *sklearn.utils*. #11166 by Gael Varoquaux.
- [Feature] Add almost complete PyPy 3 support. Known unsupported functionalities are *datasets.load_svmlight_file*, *feature_extraction.FeatureHasher* and *feature_extraction.text.HashingVectorizer*. For running on PyPy, PyPy3-v5.10+, Numpy 1.14.0+, and scipy 1.1.0+ are required. #11010 by Ronan Lamy and Roman Yurchak.

- [FEATURE] A utility method *sklearn.show_versions* was added to print out information relevant for debugging. It includes the user system, the Python executable, the version of the main libraries and BLAS binding information. #11596 by Alexandre Boucaud
- [FIX] Fixed a bug when setting parameters on meta-estimator, involving both a wrapped estimator and its parameter. #9999 by Marcus Voss and Joel Nothman.
- [FIX] Fixed a bug where calling *sklearn.base.clone* was not thread safe and could result in a "pop from empty list" error. #9569 by Andreas Müller.
- [API CHANGE] The default value of `n_jobs` is changed from `1` to `None` in all related functions and classes. `n_jobs=None` means `unset`. It will generally be interpreted as `n_jobs=1`, unless the current `joblib.Parallel` backend context specifies otherwise (See *Glossary* for additional information). Note that this change happens immediately (i.e., without a deprecation cycle). #11741 by Olivier Grisel.
- [FIX] Fixed a bug in validation helpers where passing a Dask DataFrame results in an error. #12462 by Zachariah Miller

1.16.5 Changes to estimator checks

These changes mostly affect library developers.

- Checks for transformers now apply if the estimator implements *transform*, regardless of whether it inherits from *sklearn.base.TransformerMixin*. #10474 by Joel Nothman.
- Classifiers are now checked for consistency between *decision_function* and categorical predictions. #10500 by Narine Kokhlikyan.
- Allow tests in *utils.estimator_checks.check_estimator* to test functions that accept pairwise data. #9701 by Kyle Johnson
- Allow *utils.estimator_checks.check_estimator* to check that there is no private settings apart from parameters during estimator initialization. #9378 by Herilalaina Rakotoarison
- The set of checks in *utils.estimator_checks.check_estimator* now includes a `check_set_params` test which checks that `set_params` is equivalent to passing parameters in `__init__` and warns if it encounters parameter validation. #7738 by Alvin Chiang
- Add invariance tests for clustering metrics. #8102 by Ankita Sinha and Guillaume Lemaitre.
- Add `check_methods_subset_invariance` to *check_estimator*, which checks that estimator methods are invariant if applied to a data subset. #10428 by Jonathan Ohayon
- Add tests in *utils.estimator_checks.check_estimator* to check that an estimator can handle read-only memmap input data. #10663 by Arthur Mensch and Loïc Estève.
- `check_sample_weights_pandas_series` now uses 8 rather than 6 samples to accommodate for the default number of clusters in *cluster.KMeans*. #10933 by Johannes Hansen.
- Estimators are now checked for whether `sample_weight=None` equates to `sample_weight=np.ones(...)`. #11558 by Sergul Aydore.

1.16.6 Code and Documentation Contributors

Thanks to everyone who has contributed to the maintenance and improvement of the project since version 0.19, including:

211217613, Aarshay Jain, absolutelyNoWarranty, Adam Greenhall, Adam Kleczewski, Adam Richie-Halford, adelr, AdityaDaflapurkar, Adrin Jalali, Aidan Fitzgerald, aishgrt1, Akash Shivram, Alan Liddell, Alan Yee, Albert Thomas, Alexander Lenail, Alexander-N, Alexandre Boucaud, Alexandre Gramfort, Alexandre Sevin, Alex Egg, Alvaro Perez-Diaz, Amanda, Aman Dalmia, Andreas Bjerre-Nielsen, Andreas Mueller, Andrew Peng, Angus Williams, Aniruddha Dave, annaayzenshtat, Anthony Gitter, Antonio Quinonez, Anubhav Marwaha, Arik Pamnani, Arthur Ozga, Artiem K, Arunava, Arya McCarthy, Attractadore, Aurélien Bellet, Aurélien Geron, Ayush Gupta, Balakumaran Manoharan, Bangda Sun, Barry Hart, Bastian Venthur, Ben Lawson, Benn Roth, Breno Freitas, Brent Yi, brett koonce, Caio Oliveira, Camil Staps, cclauss, Chady Kamar, Charlie Brummitt, Charlie Newey, chris, Chris, Chris Catalfo, Chris Foster, Chris Holdgraf, Christian Braune, Christian Hirsch, Christian Hogan, Christopher Jenness, Clement Joudet, cnx, cwitte, Dallas Card, Dan Barkhorn, Daniel, Daniel Ferreira, Daniel Gomez, Daniel Klevebring, Danielle Shwed, Daniel Mohns, Danil Baibak, Darius Morawiec, David Beach, David Burns, David Kirkby, David Nicholson, David Pickup, Derek, Didi Bar-Zev, diegodlh, Dillon Gardner, Dillon Niederhut, dilutedsauce, dlovell, Dmitry Mottl, Dmitry Petrov, Dor Cohen, Douglas Duhaime, Ekaterina Tuzova, Eric Chang, Eric Dean Sanchez, Erich Schubert, Eunji, Fang-Chieh Chou, FarahSaeed, felix, Félix Raimundo, fenx, filipj8, FrankHui, Franz Wompner, Freija Descamps, frsi, Gabriele Calvo, Gael Varoquaux, Gaurav Dhingra, Georgi Peev, Gil Forsyth, Giovanni Giuseppe Costa, gkevinyen5418, goncalo-rodrigues, Gryllos Prokopis, Guillaume Lemaitre, Guillaume "Vermeille" Sanchez, Gustavo De Mari Pereira, hakaa1, Hanmin Qin, Henry Lin, Hong, Honghe, Hossein Pourbozorg, Hristo, Hunan Rostomyan, iampat, Ivan PANICO, Jaewon Chung, Jake VanderPlas, jakirkham, James Bourbeau, James Malcolm, Jamie Cox, Jan Koch, Jan Margeta, Jan Schlüter, janvanrijn, Jason Wolosonovich, JC Liu, Jeb Bearer, jeremiedbb, Jimmy Wan, Jinkun Wang, Jiongyan Zhang, jjabl, jkleint, Joan Massich, Joël Billaud, Joel Nothman, Johannes Hansen, JohnStott, Jonatan Samoocha, Jonathan Ohayon, Jörg Döpfert, Joris Van den Bossche, Jose Perez-Parras Toledano, josephsalmon, jotasi, jschendel, Julian Kuhlmann, Julien Chaumond, julietcl, Justin Shenk, Karl F, Kasper Primdal Lauritzen, Katrin Leinweber, Kirill, ksemb, Kuai Yu, Kumar Ashutosh, Kyeongpil Kang, Kye Taylor, kyledrogo, Leland McInnes, Léo DS, Liam Geron, Liutong Zhou, Lizao Li, lkjcalc, Loic Esteve, louib, Luciano Viola, Lucija Gregov, Luis Osa, Luis Pedro Coelho, Luke M Craig, Luke Persola, Mabel, Mabel Villalba, Maniteja Nandana, MarkIwanchyshyn, Mark Roth, Markus Müller, MarsGuy, Martin Gubri, martin-hahn, martin-kokos, mathurinm, Matthias Feurer, Max Copeland, Mayur Kulkarni, Meghann Agarwal, Melanie Goetz, Michael A. Alcorn, Minghui Liu, Ming Li, Minh Le, Mohamed Ali Jamaoui, Mohamed Maskani, Mohammad Shahebaz, Muayyad Alsadi, Nabarun Pal, Nagarjuna Kumar, Naoya Kanai, Narendran Santhanam, NarineK, Nathaniel Saul, Nathan Suh, Nicholas Nadeau, P.Eng., AVS, Nick Hoh, Nicolas Goix, Nicolas Hug, Nicolau Werneck, nielsenmarkus11, Nihar Sheth, Nikita Titov, Nilesh Kevlani, Nirvan Anjirbag, notmatthancock, nzw, Oleksandr Pavlyk, oliblum90, Oliver Rausch, Olivier Grisel, Oren Milman, Osaid Rehman Nasir, pasbi, Patrick Fernandes, Patrick Olden, Paul Paczuski, Pedro Morales, Peter, Peter St. John, pierreablin, pietruh, Pinaki Nath Chowdhury, Piotr Szymański, Pradeep Reddy Raamana, Pravar D Mahajan, pravarmahajan, QingYing Chen, Raghav RV, Rajendra arora, RAKOTOARISON Herilalaina, Rameshwar Bhaskaran, RankyLau, Rasul Kerimov, Reiichiro Nakano, Rob, Roman Kosobrodov, Roman Yurchak, Ronan Lamy, rragundez, Rüdiger Busche, Ryan, Sachin Kelkar, Sagnik Bhattacharya, Sailesh Choyal, Sam Radhakrishnan, Sam Steingold, Samuel Bell, Samuel O. Ronsin, Saqib Nizam Shamsi, SATISH J, Saurabh Gupta, Scott Gigante, Sebastian Flennerhag, Sebastian Raschka, Sebastien Dubois, Sébastien Lerique, Sebastin Santy, Sergey Feldman, Sergey Melderis, Sergul Aydore, Shahebaz, Shalil Awaley, Shangwu Yao, Sharad Vijalapuram, Sharan Yalburgi, shenhanc78, Shivam Rastogi, Shu Haoran, siftikha, Sinclert Pérez, SolutusImmensus, Somya Anand, srajan paliwal, Sriharsha Hatwar, Sri Krishna, Stefan van der Walt, Stephen McDowell, Steven Brown, syonekura, Taehoon Lee, Takanori Hayashi, tarcusx, Taylor G Smith, theriley106, Thomas, Thomas Fan, Thomas Heavey, Tobias Madsen, tobycheese, Tom Augspurger, Tom Dupré la Tour, Tommy, Trevor Stephens, Trishnendu Ghorai, Tulio Casagrande, twosigmajab, Umar Farouk Umar, Urvang Patel, Utkarsh Upadhyay, Vadim Markovtsev, Varun Agrawal, Vathsala Achar, Vilhelm von Ehrenheim, Vinayak Mehta, Vinit, Vinod Kumar L, Viraj Mavani, Viraj Navkal, Vivek Kumar, Vlad Niculae, vqean3, Vrishank Bhardwaj, vufg, wallygauze, Warut Vijitbenjaronk, wdevazelhes, Wenhao Zhang, Wes Barnett, Will, William de Vazelhes, Will Rosenfeld, Xin Xiong, Yiming (Paul) Li, ymazari, Yufeng, Zach Griffith, Zé Vinícius, Zhenqing Hu, Zhiqing Xiao, Zijie (ZJ) Poh

1.17 Previous Releases

1.17.1 Version 0.19.2

July, 2018

This release is exclusively in order to support Python 3.7.

Related changes

- `n_iter_` may vary from previous releases in *`linear_model.LogisticRegression`* with `solver='lbfgs'` and *`linear_model.HuberRegressor`*. For Scipy <= 1.0.0, the optimizer could perform more than the requested maximum number of iterations. Now both estimators will report at most `max_iter` iterations even if more were performed. #10723 by Joel Nothman.

1.17.2 Version 0.19.1

October 23, 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.GridSearchCV`*, *`model_selection.RandomizedSearchCV`* and *`model_selection.cross_validate`* in version 0.19.0 will be changing its default value from True to False in version 0.21. We found that calculating training score could have a great effect on cross validation runtime in some cases. Users should explicitly set `return_train_score` to False if prediction or scoring functions are slow, resulting in a deleterious effect on CV runtime, or to True if they wish to use the calculated scores. #9677 by Kumar Ashutosh and Joel Nothman.
- `correlation_models` and `regression_models` from the legacy gaussian processes implementation have been belatedly deprecated. #9717 by Kumar Ashutosh.

Bug fixes

- Avoid integer overflows in *`metrics.matthews_corrcoef`*. #9693 by Sam Steingold.
- Fixed a bug in the objective function for *`manifold.TSNE`* (both exact and with the Barnes-Hut approximation) when `n_components >= 3`. #9711 by @goncalo-rodrigues.
- Fix regression in *`model_selection.cross_val_predict`* where it raised an error with `method='predict_proba'` for some probabilistic classifiers. #9641 by James Bourbeau.
- Fixed a bug where *`datasets.make_classification`* modified its input `weights`. #9865 by Sachin Kelkar.
- *`model_selection.StratifiedShuffleSplit`* now works with multioutput multiclass or multilabel data with more than 1000 columns. #9922 by Charlie Brummitt.
- Fixed a bug with nested and conditional parameter setting, e.g. setting a pipeline step and its parameter at the same time. #9945 by Andreas Müller and Joel Nothman.

Regressions in 0.19.0 fixed in 0.19.1:

- Fixed a bug where parallelised prediction in random forests was not thread-safe and could (rarely) result in arbitrary errors. #9830 by Joel Nothman.
- Fix regression in *model_selection.cross_val_predict* where it no longer accepted X as a list. #9600 by Rasul Kerimov.
- Fixed handling of cross_val_predict for binary classification with method='decision_function'. #9593 by Reiichiro Nakano and core devs.
- Fix regression in *pipeline.Pipeline* where it no longer accepted steps as a tuple. #9604 by Joris Van den Bossche.
- Fix bug where n_iter was not properly deprecated, leaving n_iter unavailable for interim use in *linear_model.SGDClassifier*, *linear_model.SGDRegressor*, *linear_model.PassiveAggressiveClassifier*, *linear_model.PassiveAggressiveRegressor* and *linear_model.Perceptron*. #9558 by Andreas Müller.
- Dataset fetchers make sure temporary files are closed before removing them, which caused errors on Windows. #9847 by Joan Massich.
- Fixed a regression in *manifold.TSNE* where it no longer supported metrics other than 'euclidean' and 'precomputed'. #9623 by Oli Blum.

Enhancements

- Our test suite and utils.estimator_checks.check_estimators can now be run without Nose installed. #9697 by Joan Massich.
- To improve usability of version 0.19's *pipeline.Pipeline* caching, memory now allows joblib.Memory instances. This make use of the new *utils.validation.check_memory* helper. issue:9584 by Kumar Ashutosh
- Some fixes to examples: #9750, #9788, #9815
- Made a FutureWarning in SGD-based estimators less verbose. #9802 by Vrishank Bhardwaj.

Code and Documentation Contributors

With thanks to:

Joel Nothman, Loic Esteve, Andreas Mueller, Kumar Ashutosh, Vrishank Bhardwaj, Hanmin Qin, Rasul Kerimov, James Bourbeau, Nagarjuna Kumar, Nathaniel Saul, Olivier Grisel, Roman Yurchak, Reiichiro Nakano, Sachin Kelkar, Sam Steingold, Yaroslav Halchenko, diegodlh, felix, goncalo-rodrigues, jkleint, oliblum90, pasbi, Anthony Gitter, Ben Lawson, Charlie Brummitt, Didi Bar-Zev, Gael Varoquaux, Joan Massich, Joris Van den Bossche, nielsenmarkus11

1.17.3 Version 0.19

August 12, 2017

Highlights

We are excited to release a number of great new features including *neighbors.LocalOutlierFactor* for anomaly detection, *preprocessing.QuantileTransformer* for robust feature transformation, and the *multioutput.ClassifierChain* meta-estimator to simply account for dependencies between classes in multilabel problems. We have some new algorithms in existing estimators, such as multiplicative update in *decomposition.NMF* and multinomial *linear_model.LogisticRegression* with L1 loss (use solver='saga').

Cross validation is now able to return the results from multiple metric evaluations. The new *model_selection.cross_validate* can return many scores on the test data as well as training set performance and timings, and we have extended the scoring and refit parameters for grid/randomized search *to handle multiple metrics*.

You can also learn faster. For instance, the *new option to cache transformations* in *pipeline.Pipeline* makes grid search over pipelines including slow transformations much more efficient. And you can predict faster: if you're sure you know what you're doing, you can turn off validating that the input is finite using *config_context*.

We've made some important fixes too. We've fixed a longstanding implementation error in *metrics.average_precision_score*, so please be cautious with prior results reported from that function. A number of errors in the *manifold.TSNE* implementation have been fixed, particularly in the default Barnes-Hut approximation. *semi_supervised.LabelSpreading* and *semi_supervised.LabelPropagation* have had substantial fixes. LabelPropagation was previously broken. LabelSpreading should now correctly respect its alpha parameter.

Changed models

The following estimators and functions, when fit with the same data and parameters, may produce different models from the previous version. This often occurs due to changes in the modelling logic (bug fixes or enhancements), or in random sampling procedures.

- *cluster.KMeans* with sparse X and initial centroids given (bug fix)
- *cross_decomposition.PLSRegression* with scale=True (bug fix)
- *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor* where min_impurity_split is used (bug fix)
- gradient boosting loss='quantile' (bug fix)
- *ensemble.IsolationForest* (bug fix)
- *feature_selection.SelectFdr* (bug fix)
- *linear_model.RANSACRegressor* (bug fix)
- *linear_model.LassoLars* (bug fix)
- *linear_model.LassoLarsIC* (bug fix)
- *manifold.TSNE* (bug fix)
- *neighbors.NearestCentroid* (bug fix)
- *semi_supervised.LabelSpreading* (bug fix)
- *semi_supervised.LabelPropagation* (bug fix)
- tree based models where min_weight_fraction_leaf is used (enhancement)
- *model_selection.StratifiedKFold* with shuffle=True (this change, due to #7823 was not mentioned in the release notes at the time)

Details are listed in the changelog below.

(While we are trying to better inform users by providing this information, we cannot assure that this list is complete.)

Changelog

New features

Classifiers and regressors

- Added *multioutput.ClassifierChain* for multi-label classification. By Adam Kleczewski.
- Added solver 'saga' that implements the improved version of Stochastic Average Gradient, in *linear_model.LogisticRegression* and *linear_model.Ridge*. It allows the use of L1 penalty with multinomial logistic loss, and behaves marginally better than 'sag' during the first epochs of ridge and logistic regression. #8446 by Arthur Mensch.

Other estimators

- Added the *neighbors.LocalOutlierFactor* class for anomaly detection based on nearest neighbors. #5279 by Nicolas Goix and Alexandre Gramfort.
- Added *preprocessing.QuantileTransformer* class and *preprocessing.quantile_transform* function for features normalization based on quantiles. #8363 by Denis Engemann, Guillaume Lemaitre, Olivier Grisel, Raghav RV, Thierry Guillemot, and Gael Varoquaux.
- The new solver 'mu' implements a Multiplicate Update in *decomposition.NMF*, allowing the optimization of all beta-divergences, including the Frobenius norm, the generalized Kullback-Leibler divergence and the Itakura-Saito divergence. #5295 by Tom Dupre la Tour.

Model selection and evaluation

- *model_selection.GridSearchCV* and *model_selection.RandomizedSearchCV* now support simultaneous evaluation of multiple metrics. Refer to the *Specifying multiple metrics for evaluation* section of the user guide for more information. #7388 by Raghav RV
- Added the *model_selection.cross_validate* which allows evaluation of multiple metrics. This function returns a dict with more useful information from cross-validation such as the train scores, fit times and score times. Refer to *The cross_validate function and multiple metric evaluation* section of the userguide for more information. #7388 by Raghav RV
- Added *metrics.mean_squared_log_error*, which computes the mean square error of the logarithmic transformation of targets, particularly useful for targets with an exponential trend. #7655 by Karan Desai.
- Added metrics.dcg_score and metrics.ndcg_score, which compute Discounted cumulative gain (DCG) and Normalized discounted cumulative gain (NDCG). #7739 by David Gasquez.
- Added the *model_selection.RepeatedKFold* and *model_selection.RepeatedStratifiedKFold*. #8120 by Neeraj Gangwar.

Miscellaneous

- Validation that input data contains no NaN or inf can now be suppressed using *config_context*, at your own risk. This will save on runtime, and may be particularly useful for prediction time. #7548 by Joel Nothman.
- Added a test to ensure parameter listing in docstrings match the function/class signature. #9206 by Alexandre Gramfort and Raghav RV.

Enhancements

Trees and ensembles

- The min_weight_fraction_leaf constraint in tree construction is now more efficient, taking a fast path to declare a node a leaf if its weight is less than 2 * the minimum. Note that the constructed tree will be different from previous versions where min_weight_fraction_leaf is used. #7441 by Nelson Liu.

- *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor* now support sparse input for prediction. #6101 by Ibraim Ganiev.
- *ensemble.VotingClassifier* now allows changing estimators by using *ensemble.VotingClassifier.set_params*. An estimator can also be removed by setting it to None. #7674 by Yichuan Liu.
- *tree.export_graphviz* now shows configurable number of decimal places. #8698 by Guillaume Lemaitre.
- Added flatten_transform parameter to *ensemble.VotingClassifier* to change output shape of *transform* method to 2 dimensional. #7794 by Ibraim Ganiev and Herilalaina Rakotoarison.

Linear, kernelized and related models

- *linear_model.SGDClassifier*, *linear_model.SGDRegressor*, *linear_model.PassiveAggressiveClassifier*, *linear_model.PassiveAggressiveRegressor* and *linear_model.Perceptron* now expose max_iter and tol parameters, to handle convergence more precisely. n_iter parameter is deprecated, and the fitted estimator exposes a n_iter_ attribute, with actual number of iterations before convergence. #5036 by Tom Dupre la Tour.
- Added average parameter to perform weight averaging in *linear_model.PassiveAggressiveClassifier*. #4939 by Andrea Esuli.
- *linear_model.RANSACRegressor* no longer throws an error when calling fit if no inliers are found in its first iteration. Furthermore, causes of skipped iterations are tracked in newly added attributes, n_skips_*. #7914 by Michael Horrell.
- In *gaussian_process.GaussianProcessRegressor*, method predict is a lot faster with return_std=True. #8591 by Hadrien Bertrand.
- Added return_std to predict method of *linear_model.ARDRegression* and *linear_model.BayesianRidge*. #7838 by Sergey Feldman.
- Memory usage enhancements: Prevent cast from float32 to float64 in: *linear_model.MultiTaskElasticNet*; *linear_model.LogisticRegression* when using newton-cg solver; and *linear_model.Ridge* when using svd, sparse_cg, cholesky or lsqr solvers. #8835, #8061 by Joan Massich and Nicolas Cordier and Thierry Guillemot.

Other predictors

- Custom metrics for the neighbors binary trees now have fewer constraints: they must take two 1d-arrays and return a float. #6288 by Jake Vanderplas.
- algorithm='auto in neighbors estimators now chooses the most appropriate algorithm for all input types and metrics. #9145 by Herilalaina Rakotoarison and Reddy Chinthala.

Decomposition, manifold learning and clustering

- *cluster.MiniBatchKMeans* and *cluster.KMeans* now use significantly less memory when assigning data points to their nearest cluster center. #7721 by Jon Crall.
- *decomposition.PCA*, *decomposition.IncrementalPCA* and *decomposition.TruncatedSVD* now expose the singular values from the underlying SVD. They are stored in the attribute singular_values_, like in *decomposition.IncrementalPCA*. #7685 by Tommy Löfstedt
- *decomposition.NMF* now faster when beta_loss=0. #9277 by @hongkahjun.
- Memory improvements for method barnes_hut in *manifold.TSNE* #7089 by Thomas Moreau and Olivier Grisel.
- Optimization schedule improvements for Barnes-Hut *manifold.TSNE* so the results are closer to the one from the reference implementation lvdmaaten/bhtsne by Thomas Moreau and Olivier Grisel.

- Memory usage enhancements: Prevent cast from float32 to float64 in *decomposition.PCA* and decomposition.randomized_svd_low_rank. #9067 by Raghav RV.

Preprocessing and feature selection

- Added norm_order parameter to *feature_selection.SelectFromModel* to enable selection of the norm order when coef_ is more than 1D. #6181 by Antoine Wendlinger.
- Added ability to use sparse matrices in *feature_selection.f_regression* with center=True. #8065 by Daniel LeJeune.
- Small performance improvement to n-gram creation in feature_extraction.text by binding methods for loops and special-casing unigrams. #7567 by Jaye Doepke
- Relax assumption on the data for the *kernel_approximation.SkewedChi2Sampler*. Since the Skewed-Chi2 kernel is defined on the open interval $(-skewedness; +\infty)^d$, the transform function should not check whether X < 0 but whether X < -self.skewedness. #7573 by Romain Brault.
- Made default kernel parameters kernel-dependent in *kernel_approximation.Nystroem*. #5229 by Saurabh Bansod and Andreas Müller.

Model evaluation and meta-estimators

- *pipeline.Pipeline* is now able to cache transformers within a pipeline by using the memory constructor parameter. #7990 by Guillaume Lemaitre.
- *pipeline.Pipeline* steps can now be accessed as attributes of its named_steps attribute. #8586 by Herilalaina Rakotoarison.
- Added sample_weight parameter to *pipeline.Pipeline.score*. #7723 by Mikhail Korobov.
- Added ability to set n_jobs parameter to *pipeline.make_union*. A TypeError will be raised for any other kwargs. #8028 by Alexander Booth.
- *model_selection.GridSearchCV*, *model_selection.RandomizedSearchCV* and *model_selection.cross_val_score* now allow estimators with callable kernels which were previously prohibited. #8005 by Andreas Müller .
- *model_selection.cross_val_predict* now returns output of the correct shape for all values of the argument method. #7863 by Aman Dalmia.
- Added shuffle and random_state parameters to shuffle training data before taking prefixes of it based on training sizes in *model_selection.learning_curve*. #7506 by Narine Kokhlikyan.
- *model_selection.StratifiedShuffleSplit* now works with multioutput multiclass (or multilabel) data. #9044 by Vlad Niculae.
- Speed improvements to *model_selection.StratifiedShuffleSplit*. #5991 by Arthur Mensch and Joel Nothman.
- Add shuffle parameter to *model_selection.train_test_split*. #8845 by themrmax
- *multioutput.MultiOutputRegressor* and *multioutput.MultiOutputClassifier* now support online learning using partial_fit. :issue: 8053 by Peng Yu.
- Add max_train_size parameter to *model_selection.TimeSeriesSplit* #8282 by Aman Dalmia.
- More clustering metrics are now available through *metrics.get_scorer* and scoring parameters. #8117 by Raghav RV.
- A scorer based on *metrics.explained_variance_score* is also available. #9259 by Hanmin Qin.

Metrics

- *metrics.matthews_corrcoef* now support multiclass classification. #8094 by Jon Crall.

- Add `sample_weight` parameter to *metrics.cohen_kappa_score*. #8335 by Victor Poughon.

Miscellaneous

- `utils.check_estimator` now attempts to ensure that methods transform, predict, etc. do not set attributes on the estimator. #7533 by Ekaterina Krivich.
- Added type checking to the `accept_sparse` parameter in `utils.validation` methods. This parameter now accepts only boolean, string, or list/tuple of strings. `accept_sparse=None` is deprecated and should be replaced by `accept_sparse=False`. #7880 by Josh Karnofsky.
- Make it possible to load a chunk of an svmlight formatted file by passing a range of bytes to *datasets.load_svmlight_file*. #935 by Olivier Grisel.
- *dummy.DummyClassifier* and *dummy.DummyRegressor* now accept non-finite features. #8931 by @Attractadore.

Bug fixes

Trees and ensembles

- Fixed a memory leak in trees when using trees with `criterion='mae'`. #8002 by Raghav RV.
- Fixed a bug where *ensemble.IsolationForest* uses an an incorrect formula for the average path length #8549 by Peter Wang.
- Fixed a bug where *ensemble.AdaBoostClassifier* throws `ZeroDivisionError` while fitting data with single class labels. #7501 by Dominik Krzeminski.
- Fixed a bug in *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor* where a float being compared to `0.0` using `==` caused a divide by zero error. #7970 by He Chen.
- Fix a bug where *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor* ignored the `min_impurity_split` parameter. #8006 by Sebastian Pölsterl.
- Fixed `oob_score` in *ensemble.BaggingClassifier*. #8936 by Michael Lewis
- Fixed excessive memory usage in prediction for random forests estimators. #8672 by Mike Benfield.
- Fixed a bug where `sample_weight` as a list broke random forests in Python 2 #8068 by @xor.
- Fixed a bug where *ensemble.IsolationForest* fails when `max_features` is less than 1. #5732 by Ishank Gulati.
- Fix a bug where gradient boosting with `loss='quantile'` computed negative errors for negative values of `ytrue - ypred` leading to wrong values when calling `__call__`. #8087 by Alexis Mignon
- Fix a bug where *ensemble.VotingClassifier* raises an error when a numpy array is passed in for weights. #7983 by Vincent Pham.
- Fixed a bug where *tree.export_graphviz* raised an error when the length of features_names does not match n_features in the decision tree. #8512 by Li Li.

Linear, kernelized and related models

- Fixed a bug where *linear_model.RANSACRegressor.fit* may run until `max_iter` if it finds a large inlier group early. #8251 by @aivision2020.
- Fixed a bug where *naive_bayes.MultinomialNB* and *naive_bayes.BernoulliNB* failed when `alpha=0`. #5814 by Yichuan Liu and Herilalaina Rakotoarison.

- Fixed a bug where *linear_model.LassoLars* does not give the same result as the LassoLars implementation available in R (lars library). #7849 by Jair Montoya Martinez.
- Fixed a bug in linear_model.RandomizedLasso, *linear_model.Lars*, *linear_model.LassoLars*, *linear_model.LarsCV* and *linear_model.LassoLarsCV*, where the parameter precompute was not used consistently across classes, and some values proposed in the docstring could raise errors. #5359 by Tom Dupre la Tour.
- Fix inconsistent results between *linear_model.RidgeCV* and *linear_model.Ridge* when using normalize=True. #9302 by Alexandre Gramfort.
- Fix a bug where *linear_model.LassoLars.fit* sometimes left coef_ as a list, rather than an ndarray. #8160 by CJ Carey.
- Fix *linear_model.BayesianRidge.fit* to return ridge parameter alpha_ and lambda_ consistent with calculated coefficients coef_ and intercept_. #8224 by Peter Gedeck.
- Fixed a bug in *svm.OneClassSVM* where it returned floats instead of integer classes. #8676 by Vathsala Achar.
- Fix AIC/BIC criterion computation in *linear_model.LassoLarsIC*. #9022 by Alexandre Gramfort and Mehmet Basbug.
- Fixed a memory leak in our LibLinear implementation. #9024 by Sergei Lebedev
- Fix bug where stratified CV splitters did not work with *linear_model.LassoCV*. #8973 by Paulo Haddad.
- Fixed a bug in *gaussian_process.GaussianProcessRegressor* when the standard deviation and covariance predicted without fit would fail with a unmeaningful error by default. #6573 by Quazi Marufur Rahman and Manoj Kumar.

Other predictors

- Fix semi_supervised.BaseLabelPropagation to correctly implement LabelPropagation and LabelSpreading as done in the referenced papers. #9239 by Andre Ambrosio Boechat, Utkarsh Upadhyay, and Joel Nothman.

Decomposition, manifold learning and clustering

- Fixed the implementation of *manifold.TSNE*:
- early_exageration parameter had no effect and is now used for the first 250 optimization iterations.
- Fixed the AssertionError: Tree consistency failed exception reported in #8992.
- Improve the learning schedule to match the one from the reference implementation lvdmaaten/bhtsne. by Thomas Moreau and Olivier Grisel.
- Fix a bug in *decomposition.LatentDirichletAllocation* where the perplexity method was returning incorrect results because the transform method returns normalized document topic distributions as of version 0.18. #7954 by Gary Foreman.
- Fix output shape and bugs with n_jobs > 1 in *decomposition.SparseCoder* transform and *decomposition.sparse_encode* for one-dimensional data and one component. This also impacts the output shape of *decomposition.DictionaryLearning*. #8086 by Andreas Müller.
- Fixed the implementation of explained_variance_ in *decomposition.PCA*, decomposition.RandomizedPCA and *decomposition.IncrementalPCA*. #9105 by Hanmin Qin.
- Fixed the implementation of noise_variance_ in *decomposition.PCA*. #9108 by Hanmin Qin.
- Fixed a bug where *cluster.DBSCAN* gives incorrect result when input is a precomputed sparse matrix with initial rows all zero. #8306 by Akshay Gupta

- Fix a bug regarding fitting `cluster.KMeans` with a sparse array X and initial centroids, where X's means were unnecessarily being subtracted from the centroids. #7872 by Josh Karnofsky.
- Fixes to the input validation in `covariance.EllipticEnvelope`. #8086 by Andreas Müller.
- Fixed a bug in `covariance.MinCovDet` where inputting data that produced a singular covariance matrix would cause the helper method `_c_step` to throw an exception. #3367 by Jeremy Steward
- Fixed a bug in `manifold.TSNE` affecting convergence of the gradient descent. #8768 by David DeTomaso.
- Fixed a bug in `manifold.TSNE` where it stored the incorrect `kl_divergence_`. #6507 by Sebastian Saeger.
- Fixed improper scaling in `cross_decomposition.PLSRegression` with `scale=True`. #7819 by jayzed82.
- `cluster.bicluster.SpectralCoclustering` and `cluster.bicluster.SpectralBiclustering` `fit` method conforms with API by accepting `y` and returning the object. #6126, #7814 by Laurent Direr and Maniteja Nandana.
- Fix bug where `mixture` `sample` methods did not return as many samples as requested. #7702 by Levi John Wolf.
- Fixed the shrinkage implementation in `neighbors.NearestCentroid`. #9219 by Hanmin Qin.

Preprocessing and feature selection

- For sparse matrices, `preprocessing.normalize` with `return_norm=True` will now raise a `NotImplementedError` with 'l1' or 'l2' norm and with norm 'max' the norms returned will be the same as for dense matrices. #7771 by Ang Lu.
- Fix a bug where `feature_selection.SelectFdr` did not exactly implement Benjamini-Hochberg procedure. It formerly may have selected fewer features than it should. #7490 by Peng Meng.
- Fixed a bug where `linear_model.RandomizedLasso` and `linear_model.RandomizedLogisticRegression` breaks for sparse input. #8259 by Aman Dalmia.
- Fix a bug where `feature_extraction.FeatureHasher` mandatorily applied a sparse random projection to the hashed features, preventing the use of `feature_extraction.text.HashingVectorizer` in a pipeline with `feature_extraction.text.TfidfTransformer`. #7565 by Roman Yurchak.
- Fix a bug where `feature_selection.mutual_info_regression` did not correctly use `n_neighbors`. #8181 by Guillaume Lemaitre.

Model evaluation and meta-estimators

- Fixed a bug where `model_selection.BaseSearchCV.inverse_transform` returns `self.best_estimator_.transform()` instead of `self.best_estimator_.inverse_transform()`. #8344 by Akshay Gupta and Rasmus Eriksson.
- Added `classes_` attribute to `model_selection.GridSearchCV`, `model_selection.RandomizedSearchCV`, `grid_search.GridSearchCV`, and `grid_search.RandomizedSearchCV` that matches the `classes_` attribute of `best_estimator_`. #7661 and #8295 by Alyssa Batula, Dylan Werner-Meier, and Stephen Hoover.
- Fixed a bug where `model_selection.validation_curve` reused the same estimator for each parameter value. #7365 by Aleksandr Sandrovskii.
- `model_selection.permutation_test_score` now works with Pandas types. #5697 by Stijn Tonk.
- Several fixes to input validation in `multiclass.OutputCodeClassifier` #8086 by Andreas Müller.
- `multiclass.OneVsOneClassifier`'s `partial_fit` now ensures all classes are provided up-front. #6250 by Asish Panda.

- Fix *multioutput.MultiOutputClassifier.predict_proba* to return a list of 2d arrays, rather than a 3d array. In the case where different target columns had different numbers of classes, a ValueError would be raised on trying to stack matrices with different dimensions. #8093 by Peter Bull.
- Cross validation now works with Pandas datatypes that that have a read-only index. #9507 by Loic Esteve.

Metrics

- *metrics.average_precision_score* no longer linearly interpolates between operating points, and instead weighs precisions by the change in recall since the last operating point, as per the Wikipedia entry. (#7356). By Nick Dingwall and Gael Varoquaux.
- Fix a bug in metrics.classification._check_targets which would return 'binary' if y_true and y_pred were both 'binary' but the union of y_true and y_pred was 'multiclass'. #8377 by Loic Esteve.
- Fixed an integer overflow bug in *metrics.confusion_matrix* and hence *metrics.cohen_kappa_score*. #8354, #7929 by Joel Nothman and Jon Crall.
- Fixed passing of gamma parameter to the chi2 kernel in *metrics.pairwise.pairwise_kernels* #5211 by Nick Rhinehart, Saurabh Bansod and Andreas Müller.

Miscellaneous

- Fixed a bug when *datasets.make_classification* fails when generating more than 30 features. #8159 by Herilalaina Rakotoarison.
- Fixed a bug where *datasets.make_moons* gives an incorrect result when n_samples is odd. #8198 by Josh Levy.
- Some fetch_ functions in datasets were ignoring the download_if_missing keyword. #7944 by Ralf Gommers.
- Fix estimators to accept a sample_weight parameter of type pandas.Series in their fit function. #7825 by Kathleen Chen.
- Fix a bug in cases where numpy.cumsum may be numerically unstable, raising an exception if instability is identified. #7376 and #7331 by Joel Nothman and @yangarbiter.
- Fix a bug where base.BaseEstimator.__getstate__ obstructed pickling customizations of child-classes, when used in a multiple inheritance context. #8316 by Holger Peters.
- Update Sphinx-Gallery from 0.1.4 to 0.1.7 for resolving links in documentation build with Sphinx>1.5 #8010, #7986 by Oscar Najera
- Add data_home parameter to *sklearn.datasets.fetch_kddcup99*. #9289 by Loic Esteve.
- Fix dataset loaders using Python 3 version of makedirs to also work in Python 2. #9284 by Sebastin Santy.
- Several minor issues were fixed with thanks to the alerts of lgtm.com. #9278 by Jean Helie, among others.

API changes summary

Trees and ensembles

- Gradient boosting base models are no longer estimators. By Andreas Müller.
- All tree based estimators now accept a min_impurity_decrease parameter in lieu of the min_impurity_split, which is now deprecated. The min_impurity_decrease helps stop splitting the nodes in which the weighted impurity decrease from splitting is no longer at least min_impurity_decrease. #8449 by Raghav RV.

Linear, kernelized and related models

- n_iter parameter is deprecated in *linear_model.SGDClassifier*, *linear_model.SGDRegressor*, *linear_model.PassiveAggressiveClassifier*, *linear_model.PassiveAggressiveRegressor* and *linear_model.Perceptron*. By Tom Dupre la Tour.

Other predictors

- neighbors.LSHForest has been deprecated and will be removed in 0.21 due to poor performance. #9078 by Laurent Direr.
- *neighbors.NearestCentroid* no longer purports to support metric='precomputed' which now raises an error. #8515 by Sergul Aydore.
- The alpha parameter of *semi_supervised.LabelPropagation* now has no effect and is deprecated to be removed in 0.21. #9239 by Andre Ambrosio Boechat, Utkarsh Upadhyay, and Joel Nothman.

Decomposition, manifold learning and clustering

- Deprecate the doc_topic_distr argument of the perplexity method in *decomposition.LatentDirichletAllocation* because the user no longer has access to the unnormalized document topic distribution needed for the perplexity calculation. #7954 by Gary Foreman.
- The n_topics parameter of *decomposition.LatentDirichletAllocation* has been renamed to n_components and will be removed in version 0.21. #8922 by @Attractadore.
- *decomposition.SparsePCA.transform*'s ridge_alpha parameter is deprecated in preference for class parameter. #8137 by Naoya Kanai.
- *cluster.DBSCAN* now has a metric_params parameter. #8139 by Naoya Kanai.

Preprocessing and feature selection

- *feature_selection.SelectFromModel* now has a partial_fit method only if the underlying estimator does. By Andreas Müller.
- *feature_selection.SelectFromModel* now validates the threshold parameter and sets the threshold_ attribute during the call to fit, and no longer during the call to transform`. By Andreas Müller.
- The non_negative parameter in *feature_extraction.FeatureHasher* has been deprecated, and replaced with a more principled alternative, alternate_sign. #7565 by Roman Yurchak.
- linear_model.RandomizedLogisticRegression, and linear_model.RandomizedLasso have been deprecated and will be removed in version 0.21. #8995 by Ramana.S.

Model evaluation and meta-estimators

- Deprecate the fit_params constructor input to the *model_selection.GridSearchCV* and *model_selection.RandomizedSearchCV* in favor of passing keyword parameters to the fit methods of those classes. Data-dependent parameters needed for model training should be passed as keyword arguments to fit, and conforming to this convention will allow the hyperparameter selection classes to be used with tools such as *model_selection.cross_val_predict*. #2879 by Stephen Hoover.
- In version 0.21, the default behavior of splitters that use the test_size and train_size parameter will change, such that specifying train_size alone will cause test_size to be the remainder. #7459 by Nelson Liu.
- *multiclass.OneVsRestClassifier* now has partial_fit, decision_function and predict_proba methods only when the underlying estimator does. #7812 by Andreas Müller and Mikhail Korobov.
- *multiclass.OneVsRestClassifier* now has a partial_fit method only if the underlying estimator does. By Andreas Müller.

- The decision_function output shape for binary classification in *multiclass.OneVsRestClassifier* and *multiclass.OneVsOneClassifier* is now (n_samples,) to conform to scikit-learn conventions. #9100 by Andreas Müller.
- The *multioutput.MultiOutputClassifier.predict_proba* function used to return a 3d array (n_samples, n_classes, n_outputs). In the case where different target columns had different numbers of classes, a ValueError would be raised on trying to stack matrices with different dimensions. This function now returns a list of arrays where the length of the list is n_outputs, and each array is (n_samples, n_classes) for that particular output. #8093 by Peter Bull.
- Replace attribute named_steps dict to utils.Bunch in *pipeline.Pipeline* to enable tab completion in interactive environment. In the case conflict value on named_steps and dict attribute, dict behavior will be prioritized. #8481 by Herilalaina Rakotoarison.

Miscellaneous

- Deprecate the y parameter in transform and inverse_transform. The method should not accept y parameter, as it's used at the prediction time. #8174 by Tahar Zanouda, Alexandre Gramfort and Raghav RV.
- SciPy >= 0.13.3 and NumPy >= 1.8.2 are now the minimum supported versions for scikit-learn. The following backported functions in utils have been removed or deprecated accordingly. #8854 and #8874 by Naoya Kanai
- The store_covariances and covariances_ parameters of *discriminant_analysis.QuadraticDiscriminantAnalysis* has been renamed to store_covariance and covariance_ to be consistent with the corresponding parameter names of the *discriminant_analysis.LinearDiscriminantAnalysis*. They will be removed in version 0.21. #7998 by Jiacheng

  Removed in 0.19:

  - utils.fixes.argpartition
  - utils.fixes.array_equal
  - utils.fixes.astype
  - utils.fixes.bincount
  - utils.fixes.expit
  - utils.fixes.frombuffer_empty
  - utils.fixes.in1d
  - utils.fixes.norm
  - utils.fixes.rankdata
  - utils.fixes.safe_copy

  Deprecated in 0.19, to be removed in 0.21:

  - utils.arpack.eigs
  - utils.arpack.eigsh
  - utils.arpack.svds
  - utils.extmath.fast_dot
  - utils.extmath.logsumexp
  - utils.extmath.norm
  - utils.extmath.pinvh
  - utils.graph.graph_laplacian

  - utils.random.choice
  - utils.sparsetools.connected_components
  - utils.stats.rankdata

- Estimators with both methods decision_function and predict_proba are now required to have a monotonic relation between them. The method check_decision_proba_consistency has been added in **utils.estimator_checks** to check their consistency. #7578 by Shubham Bhardwaj
- All checks in utils.estimator_checks, in particular *utils.estimator_checks.check_estimator* now accept estimator instances. Most other checks do not accept estimator classes any more. #9019 by Andreas Müller.
- Ensure that estimators' attributes ending with _ are not set in the constructor but only in the fit method. Most notably, ensemble estimators (deriving from ensemble.BaseEnsemble) now only have self.estimators_ available after fit. #7464 by Lars Buitinck and Loic Esteve.

Code and Documentation Contributors

Thanks to everyone who has contributed to the maintenance and improvement of the project since version 0.18, including:

Joel Nothman, Loic Esteve, Andreas Mueller, Guillaume Lemaitre, Olivier Grisel, Hanmin Qin, Raghav RV, Alexandre Gramfort, themrmax, Aman Dalmia, Gael Varoquaux, Naoya Kanai, Tom Dupré la Tour, Rishikesh, Nelson Liu, Taehoon Lee, Nelle Varoquaux, Aashil, Mikhail Korobov, Sebastin Santy, Joan Massich, Roman Yurchak, RAKOTOARISON Herilalaina, Thierry Guillemot, Alexandre Abadie, Carol Willing, Balakumaran Manoharan, Josh Karnofsky, Vlad Niculae, Utkarsh Upadhyay, Dmitry Petrov, Minghui Liu, Srivatsan, Vincent Pham, Albert Thomas, Jake VanderPlas, Attractadore, JC Liu, alexandercbooth, chkoar, Óscar Nájera, Aarshay Jain, Kyle Gilliam, Ramana Subramanyam, CJ Carey, Clement Joudet, David Robles, He Chen, Joris Van den Bossche, Karan Desai, Katie Luangkote, Leland McInnes, Maniteja Nandana, Michele Lacchia, Sergei Lebedev, Shubham Bhardwaj, akshay0724, omtcyfz, rickiepark, waterponey, Vathsala Achar, jbDelafosse, Ralf Gommers, Ekaterina Krivich, Vivek Kumar, Ishank Gulati, Dave Elliott, ldirer, Reiichiro Nakano, Levi John Wolf, Mathieu Blondel, Sid Kapur, Dougal J. Sutherland, midinas, mikebenfield, Sourav Singh, Aseem Bansal, Ibraim Ganiev, Stephen Hoover, AishwaryaRK, Steven C. Howell, Gary Foreman, Neeraj Gangwar, Tahar, Jon Crall, dokato, Kathy Chen, ferria, Thomas Moreau, Charlie Brummitt, Nicolas Goix, Adam Kleczewski, Sam Shleifer, Nikita Singh, Basil Beirouti, Giorgio Patrini, Manoj Kumar, Rafael Possas, James Bourbeau, James A. Bednar, Janine Harper, Jaye, Jean Helie, Jeremy Steward, Artsiom, John Wei, Jonathan LIgo, Jonathan Rahn, seanpwilliams, Arthur Mensch, Josh Levy, Julian Kuhlmann, Julien Aubert, Jörn Hees, Kai, shivamgargsya, Kat Hempstalk, Kaushik Lakshmikanth, Kennedy, Kenneth Lyons, Kenneth Myers, Kevin Yap, Kirill Bobyrev, Konstantin Podshumok, Arthur Imbert, Lee Murray, toastedcornflakes, Lera, Li Li, Arthur Douillard, Mainak Jas, tobycheese, Manraj Singh, Manvendra Singh, Marc Meketon, MarcoFalke, Matthew Brett, Matthias Gilch, Mehul Ahuja, Melanie Goetz, Meng, Peng, Michael Dezube, Michal Baumgartner, vibrantabhi19, Artem Golubin, Milen Paskov, Antonin Carette, Morikko, MrMjauh, NALEPA Emmanuel, Namiya, Antoine Wendlinger, Narine Kokhlikyan, NarineK, Nate Guerin, Angus Williams, Ang Lu, Nicole Vavrova, Nitish Pandey, Okhlopkov Daniil Olegovich, Andy Craze, Om Prakash, Parminder Singh, Patrick Carlson, Patrick Pei, Paul Ganssle, Paulo Haddad, Paweł Lorek, Peng Yu, Pete Bachant, Peter Bull, Peter Csizsek, Peter Wang, Pieter Arthur de Jong, Ping-Yao, Chang, Preston Parry, Puneet Mathur, Quentin Hibon, Andrew Smith, Andrew Jackson, 1kastner, Rameshwar Bhaskaran, Rebecca Bilbro, Remi Rampin, Andrea Esuli, Rob Hall, Robert Bradshaw, Romain Brault, Aman Pratik, Ruifeng Zheng, Russell Smith, Sachin Agarwal, Sailesh Choyal, Samson Tan, Samuël Weber, Sarah Brown, Sebastian Pölsterl, Sebastian Raschka, Sebastian Saeger, Alyssa Batula, Abhyuday Pratap Singh, Sergey Feldman, Sergul Aydore, Sharan Yalburgi, willduan, Siddharth Gupta, Sri Krishna, Almer, Stijn Tonk, Allen Riddell, Theofilos Papapanagiotou, Alison, Alexis Mignon, Tommy Boucher, Tommy Löfstedt, Toshihiro Kamishima, Tyler Folkman, Tyler Lanigan, Alexander Junge, Varun Shenoy, Victor Poughon, Vilhelm von Ehrenheim, Aleksandr Sandrovskii, Alan Yee, Vlasios Vasileiou, Warut Vijitbenjaronk, Yang Zhang, Yaroslav Halchenko, Yichuan Liu, Yuichi Fujikawa, affanv14, aivision2020, xor, andreh7, brady salz, campustrampus, Agamemnon Krasoulis, ditenberg, elena-sharova, filipj8, fukatani, gedeck, guiniol, guoci, hakaa1, hongkahjun, i-am-xhy, jakirkham, jaroslaw-weber, jayzed82, jeroko, jmontoyam, jonathan.striebel, josephsalmon, jschendel, leereeves, martin-hahn, mathurinm, mehak-sachdeva, mlewis1729, mlliou112, mthorrell, ndingwall, nuffe, yangarbiter, plagree, pldtc325, Breno Freitas, Brett Olsen, Brian A. Alfano, Brian Burns, polmauri, Brandon Carter, Charlton Austin, Chayant T15h, Chinmaya Pancholi, Christian Danielsen, Chung Yen, Chyi-Kwei Yau, pravarmahajan, DOHMATOB Elvis, Daniel LeJeune, Daniel Hnyk, Darius Morawiec, David DeTomaso, David Gasquez, David Haberthür, David Heryanto, David Kirkby, David Nicholson, rashchedrin, Deborah Gertrude Digges, Denis Engemann, Devansh D, Dickson, Bob Baxley, Don86, E. Lynch-Klarup, Ed Rogers, Elizabeth Ferriss, Ellen-Co2, Fabian Egli, Fang-Chieh Chou, Bing Tian Dai, Greg Stupp, Grzegorz Szpak, Bertrand Thirion, Hadrien Bertrand, Harizo Rajaona, zxcvbnius, Henry Lin, Holger Peters, Icyblade Dai, Igor Andriushchenko, Ilya, Isaac Laughlin, Iván Vallés, Aurélien Bellet, JPFrancoia, Jacob Schreiber, Asish Mahapatra

1.17.4 Version 0.18.2

June 20, 2017

> **Last release with Python 2.6 support**
>
> Scikit-learn 0.18 is the last major release of scikit-learn to support Python 2.6. Later versions of scikit-learn will require Python 2.7 or above.

Changelog

- Fixes for compatibility with NumPy 1.13.0: #7946 #8355 by Loic Esteve.
- Minor compatibility changes in the examples #9010 #8040 #9149.

Code Contributors

Aman Dalmia, Loic Esteve, Nate Guerin, Sergei Lebedev

1.17.5 Version 0.18.1

November 11, 2016

Changelog

Enhancements

- Improved `sample_without_replacement` speed by utilizing numpy.random.permutation for most cases. As a result, samples may differ in this release for a fixed random state. Affected estimators:
  - *`ensemble.BaggingClassifier`*
  - *`ensemble.BaggingRegressor`*
  - *`linear_model.RANSACRegressor`*
  - *`model_selection.RandomizedSearchCV`*
  - *`random_projection.SparseRandomProjection`*

  This also affects the *`datasets.make_classification`* method.

Bug fixes

- Fix issue where `min_grad_norm` and `n_iter_without_progress` parameters were not being utilised by *manifold.TSNE*. #6497 by Sebastian Säger
- Fix bug for svm's decision values when `decision_function_shape` is `ovr` in *svm.SVC*. *svm.SVC*'s decision_function was incorrect from versions 0.17.0 through 0.18.0. #7724 by Bing Tian Dai
- Attribute `explained_variance_ratio` of *discriminant_analysis.LinearDiscriminantAnalysis* calculated with SVD and Eigen solver are now of the same length. #7632 by JPFrancoia
- Fixes issue in *Univariate feature selection* where score functions were not accepting multi-label targets. #7676 by Mohammed Affan
- Fixed setting parameters when calling `fit` multiple times on *feature_selection.SelectFromModel*. #7756 by Andreas Müller
- Fixes issue in `partial_fit` method of *multiclass.OneVsRestClassifier* when number of classes used in `partial_fit` was less than the total number of classes in the data. #7786 by Srivatsan Ramesh
- Fixes issue in *calibration.CalibratedClassifierCV* where the sum of probabilities of each class for a data was not 1, and `CalibratedClassifierCV` now handles the case where the training set has less number of classes than the total data. #7799 by Srivatsan Ramesh
- Fix a bug where *sklearn.feature_selection.SelectFdr* did not exactly implement Benjamini-Hochberg procedure. It formerly may have selected fewer features than it should. #7490 by Peng Meng.
- *sklearn.manifold.LocallyLinearEmbedding* now correctly handles integer inputs. #6282 by Jake Vanderplas.
- The `min_weight_fraction_leaf` parameter of tree-based classifiers and regressors now assumes uniform sample weights by default if the `sample_weight` argument is not passed to the `fit` function. Previously, the parameter was silently ignored. #7301 by Nelson Liu.
- Numerical issue with *linear_model.RidgeCV* on centered data when `n_features > n_samples`. #6178 by Bertrand Thirion
- Tree splitting criterion classes' cloning/pickling is now memory safe #7680 by Ibraim Ganiev.
- Fixed a bug where *decomposition.NMF* sets its `n_iters_` attribute in *transform()*. #7553 by Ekaterina Krivich.
- *sklearn.linear_model.LogisticRegressionCV* now correctly handles string labels. #5874 by Raghav RV.
- Fixed a bug where *sklearn.model_selection.train_test_split* raised an error when `stratify` is a list of string labels. #7593 by Raghav RV.
- Fixed a bug where *sklearn.model_selection.GridSearchCV* and *sklearn.model_selection.RandomizedSearchCV* were not pickleable because of a pickling bug in `np.ma.MaskedArray`. #7594 by Raghav RV.
- All cross-validation utilities in *sklearn.model_selection* now permit one time cross-validation splitters for the `cv` parameter. Also non-deterministic cross-validation splitters (where multiple calls to `split` produce dissimilar splits) can be used as `cv` parameter. The *sklearn.model_selection.GridSearchCV* will cross-validate each parameter setting on the split produced by the first `split` call to the cross-validation splitter. #7660 by Raghav RV.
- Fix bug where *preprocessing.MultiLabelBinarizer.fit_transform* returned an invalid CSR matrix. #7750 by CJ Carey.

- Fixed a bug where *`metrics.pairwise.cosine_distances`* could return a small negative distance. #7732 by Artsion.

API changes summary

Trees and forests

- The `min_weight_fraction_leaf` parameter of tree-based classifiers and regressors now assumes uniform sample weights by default if the `sample_weight` argument is not passed to the `fit` function. Previously, the parameter was silently ignored. #7301 by Nelson Liu.
- Tree splitting criterion classes' cloning/pickling is now memory safe. #7680 by Ibraim Ganiev.

Linear, kernelized and related models

- Length of `explained_variance_ratio` of *`discriminant_analysis.LinearDiscriminantAnalysis`* changed for both Eigen and SVD solvers. The attribute has now a length of min(n_components, n_classes - 1). #7632 by JPFrancoia
- Numerical issue with *`linear_model.RidgeCV`* on centered data when `n_features > n_samples`. #6178 by Bertrand Thirion

1.17.6 Version 0.18

September 28, 2016

> **Last release with Python 2.6 support**
>
> Scikit-learn 0.18 will be the last version of scikit-learn to support Python 2.6. Later versions of scikit-learn will require Python 2.7 or above.

Model Selection Enhancements and API Changes

- **The model_selection module**

  The new module *`sklearn.model_selection`*, which groups together the functionalities of formerly `sklearn.cross_validation`, `sklearn.grid_search` and `sklearn.learning_curve`, introduces new possibilities such as nested cross-validation and better manipulation of parameter searches with Pandas.

  Many things will stay the same but there are some key differences. Read below to know more about the changes.

- **Data-independent CV splitters enabling nested cross-validation**

  The new cross-validation splitters, defined in the *`sklearn.model_selection`*, are no longer initialized with any data-dependent parameters such as `y`. Instead they expose a `split` method that takes in the data and yields a generator for the different splits.

  This change makes it possible to use the cross-validation splitters to perform nested cross-validation, facilitated by *`model_selection.GridSearchCV`* and *`model_selection.RandomizedSearchCV`* utilities.

- **The enhanced cv_results_ attribute**

  The new `cv_results_` attribute (of *`model_selection.GridSearchCV`* and *`model_selection.RandomizedSearchCV`*) introduced in lieu of the `grid_scores_` attribute is a dict of 1D arrays with elements in each array corresponding to the parameter settings (i.e. search candidates).

The cv_results_ dict can be easily imported into pandas as a DataFrame for exploring the search results.

The cv_results_ arrays include scores for each cross-validation split (with keys such as 'split0_test_score'), as well as their mean ('mean_test_score') and standard deviation ('std_test_score').

The ranks for the search candidates (based on their mean cross-validation score) is available at cv_results_['rank_test_score'].

The parameter values for each parameter is stored separately as numpy masked object arrays. The value, for that search candidate, is masked if the corresponding parameter is not applicable. Additionally a list of all the parameter dicts are stored at cv_results_['params'].

- **Parameters n_folds and n_iter renamed to n_splits**

  Some parameter names have changed: The n_folds parameter in new *model_selection.KFold*, *model_selection.GroupKFold* (see below for the name change), and *model_selection.StratifiedKFold* is now renamed to n_splits. The n_iter parameter in *model_selection.ShuffleSplit*, the new class *model_selection.GroupShuffleSplit* and *model_selection.StratifiedShuffleSplit* is now renamed to n_splits.

- **Rename of splitter classes which accepts group labels along with data**

  The cross-validation splitters LabelKFold, LabelShuffleSplit, LeaveOneLabelOut and LeavePLabelOut have been renamed to *model_selection.GroupKFold*, *model_selection.GroupShuffleSplit*, *model_selection.LeaveOneGroupOut* and *model_selection.LeavePGroupsOut* respectively.

  Note the change from singular to plural form in *model_selection.LeavePGroupsOut*.

- **Fit parameter labels renamed to groups**

  The labels parameter in the split method of the newly renamed splitters *model_selection.GroupKFold*, *model_selection.LeaveOneGroupOut*, *model_selection.LeavePGroupsOut*, *model_selection.GroupShuffleSplit* is renamed to groups following the new nomenclature of their class names.

- **Parameter n_labels renamed to n_groups**

  The parameter n_labels in the newly renamed *model_selection.LeavePGroupsOut* is changed to n_groups.

- Training scores and Timing information

  cv_results_ also includes the training scores for each cross-validation split (with keys such as 'split0_train_score'), as well as their mean ('mean_train_score') and standard deviation ('std_train_score'). To avoid the cost of evaluating training score, set return_train_score=False.

  Additionally the mean and standard deviation of the times taken to split, train and score the model across all the cross-validation splits is available at the key 'mean_time' and 'std_time' respectively.

Changelog

New features

Classifiers and Regressors

- The Gaussian Process module has been reimplemented and now offers classification and regression estimators through *gaussian_process.GaussianProcessClassifier* and *gaussian_process.GaussianProcessRegressor*. Among other things, the new implementation supports kernel engineering, gradient-based hyperparameter optimization or sampling of functions from GP prior and GP posterior. Extensive documentation and examples are provided. By Jan Hendrik Metzen.

- Added new supervised learning algorithm: *Multi-layer Perceptron* #3204 by Issam H. Laradji
- Added `linear_model.HuberRegressor`, a linear model robust to outliers. #5291 by Manoj Kumar.
- Added the `multioutput.MultiOutputRegressor` meta-estimator. It converts single output regressors to multi-output regressors by fitting one regressor per output. By Tim Head.

Other estimators

- New `mixture.GaussianMixture` and `mixture.BayesianGaussianMixture` replace former mixture models, employing faster inference for sounder results. #7295 by Wei Xue and Thierry Guillemot.
- Class `decomposition.RandomizedPCA` is now factored into `decomposition.PCA` and it is available calling with parameter `svd_solver='randomized'`. The default number of `n_iter` for `'randomized'` has changed to 4. The old behavior of PCA is recovered by `svd_solver='full'`. An additional solver calls `arpack` and performs truncated (non-randomized) SVD. By default, the best solver is selected depending on the size of the input and the number of components requested. #5299 by Giorgio Patrini.
- Added two functions for mutual information estimation: `feature_selection.mutual_info_classif` and `feature_selection.mutual_info_regression`. These functions can be used in `feature_selection.SelectKBest` and `feature_selection.SelectPercentile` as score functions. By Andrea Bravi and Nikolay Mayorov.
- Added the `ensemble.IsolationForest` class for anomaly detection based on random forests. By Nicolas Goix.
- Added `algorithm="elkan"` to `cluster.KMeans` implementing Elkan's fast K-Means algorithm. By Andreas Müller.

Model selection and evaluation

- Added `metrics.cluster.fowlkes_mallows_score`, the Fowlkes Mallows Index which measures the similarity of two clusterings of a set of points By Arnaud Fouchet and Thierry Guillemot.
- Added `metrics.calinski_harabaz_score`, which computes the Calinski and Harabaz score to evaluate the resulting clustering of a set of points. By Arnaud Fouchet and Thierry Guillemot.
- Added new cross-validation splitter `model_selection.TimeSeriesSplit` to handle time series data. #6586 by YenChen Lin
- The cross-validation iterators are replaced by cross-validation splitters available from `sklearn.model_selection`, allowing for nested cross-validation. See *Model Selection Enhancements and API Changes* for more information. #4294 by Raghav RV.

Enhancements

Trees and ensembles

- Added a new splitting criterion for `tree.DecisionTreeRegressor`, the mean absolute error. This criterion can also be used in `ensemble.ExtraTreesRegressor`, `ensemble.RandomForestRegressor`, and the gradient boosting estimators. #6667 by Nelson Liu.
- Added weighted impurity-based early stopping criterion for decision tree growth. #6954 by Nelson Liu
- The random forest, extra tree and decision tree estimators now has a method `decision_path` which returns the decision path of samples in the tree. By Arnaud Joly.
- A new example has been added unveiling the decision tree structure. By Arnaud Joly.

- Random forest, extra trees, decision trees and gradient boosting estimator accept the parameter `min_samples_split` and `min_samples_leaf` provided as a percentage of the training samples. By yelite and Arnaud Joly.
- Gradient boosting estimators accept the parameter `criterion` to specify to splitting criterion used in built decision trees. #6667 by Nelson Liu.
- The memory footprint is reduced (sometimes greatly) for `ensemble.bagging.BaseBagging` and classes that inherit from it, i.e, *`ensemble.BaggingClassifier`*, *`ensemble.BaggingRegressor`*, and *`ensemble.IsolationForest`*, by dynamically generating attribute `estimators_samples_` only when it is needed. By David Staub.
- Added `n_jobs` and `sample_weight` parameters for *`ensemble.VotingClassifier`* to fit underlying estimators in parallel. #5805 by Ibraim Ganiev.

Linear, kernelized and related models

- In *`linear_model.LogisticRegression`*, the SAG solver is now available in the multinomial case. #5251 by Tom Dupre la Tour.
- *`linear_model.RANSACRegressor`*, *`svm.LinearSVC`* and *`svm.LinearSVR`* now support `sample_weight`. By Imaculate.
- Add parameter `loss` to *`linear_model.RANSACRegressor`* to measure the error on the samples for every trial. By Manoj Kumar.
- Prediction of out-of-sample events with Isotonic Regression (*`isotonic.IsotonicRegression`*) is now much faster (over 1000x in tests with synthetic data). By Jonathan Arfa.
- Isotonic regression (*`isotonic.IsotonicRegression`*) now uses a better algorithm to avoid `O(n^2)` behavior in pathological cases, and is also generally faster (##6691). By Antony Lee.
- *`naive_bayes.GaussianNB`* now accepts data-independent class-priors through the parameter `priors`. By Guillaume Lemaitre.
- *`linear_model.ElasticNet`* and *`linear_model.Lasso`* now works with `np.float32` input data without converting it into `np.float64`. This allows to reduce the memory consumption. #6913 by YenChen Lin.
- *`semi_supervised.LabelPropagation`* and *`semi_supervised.LabelSpreading`* now accept arbitrary kernel functions in addition to strings `knn` and `rbf`. #5762 by Utkarsh Upadhyay.

Decomposition, manifold learning and clustering

- Added `inverse_transform` function to *`decomposition.NMF`* to compute data matrix of original shape. By Anish Shah.
- *`cluster.KMeans`* and *`cluster.MiniBatchKMeans`* now works with `np.float32` and `np.float64` input data without converting it. This allows to reduce the memory consumption by using `np.float32`. #6846 by Sebastian Säger and YenChen Lin.

Preprocessing and feature selection

- *`preprocessing.RobustScaler`* now accepts `quantile_range` parameter. #5929 by Konstantin Podshumok.
- *`feature_extraction.FeatureHasher`* now accepts string values. #6173 by Ryad Zenine and Devashish Deshpande.
- Keyword arguments can now be supplied to `func` in *`preprocessing.FunctionTransformer`* by means of the `kw_args` parameter. By Brian McFee.
- *`feature_selection.SelectKBest`* and *`feature_selection.SelectPercentile`* now accept score functions that take X, y as input and return only the scores. By Nikolay Mayorov.

Model evaluation and meta-estimators

- *multiclass.OneVsOneClassifier* and *multiclass.OneVsRestClassifier* now support partial_fit. By Asish Panda and Philipp Dowling.
- Added support for substituting or disabling *pipeline.Pipeline* and *pipeline.FeatureUnion* components using the set_params interface that powers sklearn.grid_search. See *Selecting dimensionality reduction with Pipeline and GridSearchCV* By Joel Nothman and Robert McGibbon.
- The new cv_results_ attribute of *model_selection.GridSearchCV* (and *model_selection.RandomizedSearchCV*) can be easily imported into pandas as a DataFrame. Ref *Model Selection Enhancements and API Changes* for more information. #6697 by Raghav RV.
- Generalization of *model_selection.cross_val_predict*. One can pass method names such as *predict_proba* to be used in the cross validation framework instead of the default *predict*. By Ori Ziv and Sears Merritt.
- The training scores and time taken for training followed by scoring for each search candidate are now available at the cv_results_ dict. See *Model Selection Enhancements and API Changes* for more information. #7325 by Eugene Chen and Raghav RV.

Metrics

- Added labels flag to *metrics.log_loss* to explicitly provide the labels when the number of classes in y_true and y_pred differ. #7239 by Hong Guangguo with help from Mads Jensen and Nelson Liu.
- Support sparse contingency matrices in cluster evaluation (metrics.cluster.supervised) to scale to a large number of clusters. #7419 by Gregory Stupp and Joel Nothman.
- Add sample_weight parameter to *metrics.matthews_corrcoef*. By Jatin Shah and Raghav RV.
- Speed up *metrics.silhouette_score* by using vectorized operations. By Manoj Kumar.
- Add sample_weight parameter to *metrics.confusion_matrix*. By Bernardo Stein.

Miscellaneous

- Added n_jobs parameter to *feature_selection.RFECV* to compute the score on the test folds in parallel. By Manoj Kumar
- Codebase does not contain C/C++ cython generated files: they are generated during build. Distribution packages will still contain generated C/C++ files. By Arthur Mensch.
- Reduce the memory usage for 32-bit float input arrays of utils.sparse_func.mean_variance_axis and utils.sparse_func.incr_mean_variance_axis by supporting cython fused types. By YenChen Lin.
- The ignore_warnings now accept a category argument to ignore only the warnings of a specified type. By Thierry Guillemot.
- Added parameter return_X_y and return type (data, target) : tuple option to load_iris dataset #7049, load_breast_cancer dataset #7152, load_digits dataset, load_diabetes dataset, load_linnerud dataset, load_boston dataset #7154 by Manvendra Singh.
- Simplification of the clone function, deprecate support for estimators that modify parameters in __init__. #5540 by Andreas Müller.
- When unpickling a scikit-learn estimator in a different version than the one the estimator was trained with, a UserWarning is raised, see *the documentation on model persistence* for more details. (#7248) By Andreas Müller.

Bug fixes

Trees and ensembles

- Random forest, extra trees, decision trees and gradient boosting won't accept anymore min_samples_split=1 as at least 2 samples are required to split a decision tree node. By Arnaud Joly
- *ensemble.VotingClassifier* now raises NotFittedError if predict, transform or predict_proba are called on the non-fitted estimator. by Sebastian Raschka.
- Fix bug where *ensemble.AdaBoostClassifier* and *ensemble.AdaBoostRegressor* would perform poorly if the random_state was fixed (#7411). By Joel Nothman.
- Fix bug in ensembles with randomization where the ensemble would not set random_state on base estimators in a pipeline or similar nesting. (#7411). Note, results for *ensemble.BaggingClassifier* *ensemble.BaggingRegressor*, *ensemble.AdaBoostClassifier* and *ensemble.AdaBoostRegressor* will now differ from previous versions. By Joel Nothman.

Linear, kernelized and related models

- Fixed incorrect gradient computation for loss='squared_epsilon_insensitive' in *linear_model.SGDClassifier* and *linear_model.SGDRegressor* (#6764). By Wenhua Yang.
- Fix bug in *linear_model.LogisticRegressionCV* where solver='liblinear' did not accept class_weights='balanced. (#6817). By Tom Dupre la Tour.
- Fix bug in *neighbors.RadiusNeighborsClassifier* where an error occurred when there were outliers being labelled and a weight function specified (#6902). By LeonieBorne.
- Fix *linear_model.ElasticNet* sparse decision function to match output with dense in the multioutput case.

Decomposition, manifold learning and clustering

- decomposition.RandomizedPCA default number of iterated_power is 4 instead of 3. #5141 by Giorgio Patrini.
- *utils.extmath.randomized_svd* performs 4 power iterations by default, instead or 0. In practice this is enough for obtaining a good approximation of the true eigenvalues/vectors in the presence of noise. When *n_components* is small (< .1 * min(X.shape)) n_iter is set to 7, unless the user specifies a higher number. This improves precision with few components. #5299 by Giorgio Patrini.
- Whiten/non-whiten inconsistency between components of *decomposition.PCA* and decomposition.RandomizedPCA (now factored into PCA, see the New features) is fixed. *components_* are stored with no whitening. #5299 by Giorgio Patrini.
- Fixed bug in *manifold.spectral_embedding* where diagonal of unnormalized Laplacian matrix was incorrectly set to 1. #4995 by Peter Fischer.
- Fixed incorrect initialization of utils.arpack.eigsh on all occurrences. Affects *cluster.bicluster.SpectralBiclustering*, *decomposition.KernelPCA*, *manifold.LocallyLinearEmbedding*, and *manifold.SpectralEmbedding* (#5012). By Peter Fischer.
- Attribute explained_variance_ratio_ calculated with the SVD solver of *discriminant_analysis.LinearDiscriminantAnalysis* now returns correct results. By JPFrancoia

Preprocessing and feature selection

- preprocessing.data._transform_selected now always passes a copy of X to transform function when copy=True (#7194). By Caio Oliveira.

Model evaluation and meta-estimators

- *model_selection.StratifiedKFold* now raises error if all n_labels for individual classes is less than n_folds. #6182 by Devashish Deshpande.
- Fixed bug in *model_selection.StratifiedShuffleSplit* where train and test sample could overlap in some edge cases, see #6121 for more details. By Loic Esteve.
- Fix in *sklearn.model_selection.StratifiedShuffleSplit* to return splits of size `train_size` and `test_size` in all cases (#6472). By Andreas Müller.
- Cross-validation of `OneVsOneClassifier` and `OneVsRestClassifier` now works with precomputed kernels. #7350 by Russell Smith.
- Fix incomplete `predict_proba` method delegation from *model_selection.GridSearchCV* to *linear_model.SGDClassifier* (#7159) by Yichuan Liu.

Metrics

- Fix bug in *metrics.silhouette_score* in which clusters of size 1 were incorrectly scored. They should get a score of 0. By Joel Nothman.
- Fix bug in *metrics.silhouette_samples* so that it now works with arbitrary labels, not just those ranging from 0 to n_clusters - 1.
- Fix bug where expected and adjusted mutual information were incorrect if cluster contingency cells exceeded `2**16`. By Joel Nothman.
- `metrics.pairwise.pairwise_distances` now converts arrays to boolean arrays when required in `scipy.spatial.distance`. #5460 by Tom Dupre la Tour.
- Fix sparse input support in *metrics.silhouette_score* as well as example examples/text/document_clustering.py. By YenChen Lin.
- *metrics.roc_curve* and *metrics.precision_recall_curve* no longer round `y_score` values when creating ROC curves; this was causing problems for users with very small differences in scores (#7353).

Miscellaneous

- `model_selection.tests._search._check_param_grid` now works correctly with all types that extends/implements `Sequence` (except string), including range (Python 3.x) and xrange (Python 2.x). #7323 by Viacheslav Kovalevskyi.
- *utils.extmath.randomized_range_finder* is more numerically stable when many power iterations are requested, since it applies LU normalization by default. If `n_iter<2` numerical issues are unlikely, thus no normalization is applied. Other normalization options are available: `'none', 'LU'` and `'QR'`. #5141 by Giorgio Patrini.
- Fix a bug where some formats of `scipy.sparse` matrix, and estimators with them as parameters, could not be passed to *base.clone*. By Loic Esteve.
- *datasets.load_svmlight_file* now is able to read long int QID values. #7101 by Ibraim Ganiev.

API changes summary

Linear, kernelized and related models

- `residual_metric` has been deprecated in *linear_model.RANSACRegressor*. Use `loss` instead. By Manoj Kumar.
- Access to public attributes `.X_` and `.y_` has been deprecated in *isotonic.IsotonicRegression*. By Jonathan Arfa.

Decomposition, manifold learning and clustering

- The old mixture.DPGMM is deprecated in favor of the new *mixture.BayesianGaussianMixture* (with the parameter weight_concentration_prior_type='dirichlet_process'). The new class solves the computational problems of the old class and computes the Gaussian mixture with a Dirichlet process prior faster than before. #7295 by Wei Xue and Thierry Guillemot.
- The old mixture.VBGMM is deprecated in favor of the new *mixture.BayesianGaussianMixture* (with the parameter weight_concentration_prior_type='dirichlet_distribution'). The new class solves the computational problems of the old class and computes the Variational Bayesian Gaussian mixture faster than before. #6651 by Wei Xue and Thierry Guillemot.
- The old mixture.GMM is deprecated in favor of the new *mixture.GaussianMixture*. The new class computes the Gaussian mixture faster than before and some of computational problems have been solved. #6666 by Wei Xue and Thierry Guillemot.

Model evaluation and meta-estimators

- The sklearn.cross_validation, sklearn.grid_search and sklearn.learning_curve have been deprecated and the classes and functions have been reorganized into the *sklearn.model_selection* module. Ref *Model Selection Enhancements and API Changes* for more information. #4294 by Raghav RV.
- The grid_scores_ attribute of *model_selection.GridSearchCV* and *model_selection.RandomizedSearchCV* is deprecated in favor of the attribute cv_results_. Ref *Model Selection Enhancements and API Changes* for more information. #6697 by Raghav RV.
- The parameters n_iter or n_folds in old CV splitters are replaced by the new parameter n_splits since it can provide a consistent and unambiguous interface to represent the number of train-test splits. #7187 by YenChen Lin.
- classes parameter was renamed to labels in *metrics.hamming_loss*. #7260 by Sebastián Vanrell.
- The splitter classes LabelKFold, LabelShuffleSplit, LeaveOneLabelOut and LeavePLabelsOut are renamed to *model_selection.GroupKFold*, *model_selection.GroupShuffleSplit*, *model_selection.LeaveOneGroupOut* and *model_selection.LeavePGroupsOut* respectively. Also the parameter labels in the split method of the newly renamed splitters *model_selection.LeaveOneGroupOut* and *model_selection.LeavePGroupsOut* is renamed to groups. Additionally in *model_selection.LeavePGroupsOut*, the parameter n_labels is renamed to n_groups. #6660 by Raghav RV.
- Error and loss names for scoring parameters are now prefixed by 'neg_', such as neg_mean_squared_error. The unprefixed versions are deprecated and will be removed in version 0.20. #7261 by Tim Head.

Code Contributors

Aditya Joshi, Alejandro, Alexander Fabisch, Alexander Loginov, Alexander Minyushkin, Alexander Rudy, Alexandre Abadie, Alexandre Abraham, Alexandre Gramfort, Alexandre Saint, alexfields, Alvaro Ulloa, alyssaq, Amlan Kar, Andreas Mueller, andrew giessel, Andrew Jackson, Andrew McCulloh, Andrew Murray, Anish Shah, Arafat, Archit Sharma, Ariel Rokem, Arnaud Joly, Arnaud Rachez, Arthur Mensch, Ash Hoover, asnt, b0noI, Behzad Tabibian, Bernardo, Bernhard Kratzwald, Bhargav Mangipudi, blakeflei, Boyuan Deng, Brandon Carter, Brett Naul, Brian McFee, Caio Oliveira, Camilo Lamus, Carol Willing, Cass, CeShine Lee, Charles Truong, Chyi-Kwei Yau, CJ Carey, codevig, Colin Ni, Dan Shiebler, Daniel, Daniel Hnyk, David Ellis, David Nicholson, David Staub, David Thaler, David Warshaw, Davide Lasagna, Deborah, definitelyuncertain, Didi Bar-Zev, djipey, dsquareindia, edwinENSAE, Elias Kuthe, Elvis DOHMATOB, Ethan White, Fabian Pedregosa, Fabio Ticconi, fisache, Florian Wilhelm, Francis, Francis O'Donovan, Gael Varoquaux, Ganiev Ibraim, ghg, Gilles Louppe, Giorgio Patrini, Giovanni Cherubin, Giovanni Lanzani, Glenn Qian, Gordon Mohr, govin-vatsan, Graham Clenaghan, Greg Reda, Greg Stupp, Guillaume

Lemaitre, Gustav Mörtberg, halwai, Harizo Rajaona, Harry Mavroforakis, hashcode55, hdmetor, Henry Lin, Hobson Lane, Hugo Bowne-Anderson, Igor Andriushchenko, Imaculate, Inki Hwang, Isaac Sijaranamual, Ishank Gulati, Issam Laradji, Iver Jordal, jackmartin, Jacob Schreiber, Jake Vanderplas, James Fiedler, James Routley, Jan Zikes, Janna Brettingen, jarfa, Jason Laska, jblackburne, jeff levesque, Jeffrey Blackburne, Jeffrey04, Jeremy Hintz, jeremynixon, Jeroen, Jessica Yung, Jill-Jênn Vie, Jimmy Jia, Jiyuan Qian, Joel Nothman, johannah, John, John Boersma, John Kirkham, John Moeller, jonathan.striebel, joncrall, Jordi, Joseph Munoz, Joshua Cook, JPFrancoia, jrfiedler, JulianKahnert, juliathebrave, kaichogami, KamalakerDadi, Kenneth Lyons, Kevin Wang, kingjr, kjell, Konstantin Podshumok, Kornel Kielczewski, Krishna Kalyan, krishnakalyan3, Kvle Putnam, Kyle Jackson, Lars Buitinck, ldavid, LeiG, LeightonZhang, Leland McInnes, Liang-Chi Hsieh, Lilian Besson, lizsz, Loic Esteve, Louis Tiao, Léonie Borne, Mads Jensen, Maniteja Nandana, Manoj Kumar, Manvendra Singh, Marco, Mario Krell, Mark Bao, Mark Szepieniec, Martin Madsen, MartinBpr, MaryanMorel, Massil, Matheus, Mathieu Blondel, Mathieu Dubois, Matteo, Matthias Ekman, Max Moroz, Michael Scherer, michiaki ariga, Mikhail Korobov, Moussa Taifi, mrandrewandrade, Mridul Seth, nadya-p, Naoya Kanai, Nate George, Nelle Varoquaux, Nelson Liu, Nick James, NickleDave, Nico, Nicolas Goix, Nikolay Mayorov, ningchi, nlathia, okbalefthanded, Okhlopkov, Olivier Grisel, Panos Louridas, Paul Strickland, Perrine Letellier, pestrickland, Peter Fischer, Pieter, Ping-Yao, Chang, practicalswift, Preston Parry, Qimu Zheng, Rachit Kansal, Raghav RV, Ralf Gommers, Ramana.S, Rammig, Randy Olson, Rob Alexander, Robert Lutz, Robin Schucker, Rohan Jain, Ruifeng Zheng, Ryan Yu, Rémy Léone, saihttam, Saiwing Yeung, Sam Shleifer, Samuel St-Jean, Sartaj Singh, Sasank Chilamkurthy, saurabh.bansod, Scott Andrews, Scott Lowe, seales, Sebastian Raschka, Sebastian Saeger, Sebastián Vanrell, Sergei Lebedev, shagun Sodhani, shanmuga cv, Shashank Shekhar, shawpan, shengxiduan, Shota, shuckle16, Skipper Seabold, sklearn-ci, SmedbergM, srvanrell, Sébastien Lerique, Taranjeet, themrmax, Thierry, Thierry Guillemot, Thomas, Thomas Hallock, Thomas Moreau, Tim Head, tKammy, toastedcornflakes, Tom, TomDLT, Toshihiro Kamishima, tracer0tong, Trent Hauck, trevorstephens, Tue Vo, Varun, Varun Jewalikar, Viacheslav, Vighnesh Birodkar, Vikram, Villu Ruusmann, Vinayak Mehta, walter, waterponey, Wenhua Yang, Wenjian Huang, Will Welch, wyseguy7, xyguo, yanlend, Yaroslav Halchenko, yelite, Yen, YenChenLin, Yichuan Liu, Yoav Ram, Yoshiki, Zheng RuiFeng, zivori, Óscar Nájera

1.17.7 Version 0.17.1

February 18, 2016

Changelog

Bug fixes

- Upgrade vendored joblib to version 0.9.4 that fixes an important bug in `joblib.Parallel` that can silently yield to wrong results when working on datasets larger than 1MB: https://github.com/joblib/joblib/blob/0.9.4/CHANGES.rst
- Fixed reading of Bunch pickles generated with scikit-learn version <= 0.16. This can affect users who have already downloaded a dataset with scikit-learn 0.16 and are loading it with scikit-learn 0.17. See #6196 for how this affected *`datasets.fetch_20newsgroups`*. By Loic Esteve.
- Fixed a bug that prevented using ROC AUC score to perform grid search on several CPU / cores on large arrays. See #6147 By Olivier Grisel.
- Fixed a bug that prevented to properly set the `presort` parameter in *`ensemble.GradientBoostingRegressor`*. See #5857 By Andrew McCulloh.
- Fixed a joblib error when evaluating the perplexity of a *`decomposition.LatentDirichletAllocation`* model. See #6258 By Chyi-Kwei Yau.

1.17.8 Version 0.17

November 5, 2015

Changelog

New features

- All the Scaler classes but *preprocessing.RobustScaler* can be fitted online by calling *partial_fit*. By Giorgio Patrini.
- The new class *ensemble.VotingClassifier* implements a "majority rule" / "soft voting" ensemble classifier to combine estimators for classification. By Sebastian Raschka.
- The new class *preprocessing.RobustScaler* provides an alternative to *preprocessing.StandardScaler* for feature-wise centering and range normalization that is robust to outliers. By Thomas Unterthiner.
- The new class *preprocessing.MaxAbsScaler* provides an alternative to *preprocessing.MinMaxScaler* for feature-wise range normalization when the data is already centered or sparse. By Thomas Unterthiner.
- The new class *preprocessing.FunctionTransformer* turns a Python function into a `Pipeline`-compatible transformer object. By Joe Jevnik.
- The new classes `cross_validation.LabelKFold` and `cross_validation.LabelShuffleSplit` generate train-test folds, respectively similar to `cross_validation.KFold` and `cross_validation.ShuffleSplit`, except that the folds are conditioned on a label array. By Brian McFee, Jean Kossaifi and Gilles Louppe.
- *decomposition.LatentDirichletAllocation* implements the Latent Dirichlet Allocation topic model with online variational inference. By Chyi-Kwei Yau, with code based on an implementation by Matt Hoffman. (#3659)
- The new solver `sag` implements a Stochastic Average Gradient descent and is available in both *linear_model.LogisticRegression* and *linear_model.Ridge*. This solver is very efficient for large datasets. By Danny Sullivan and Tom Dupre la Tour. (#4738)
- The new solver `cd` implements a Coordinate Descent in *decomposition.NMF*. Previous solver based on Projected Gradient is still available setting new parameter `solver` to `pg`, but is deprecated and will be removed in 0.19, along with `decomposition.ProjectedGradientNMF` and parameters `sparseness`, `eta`, `beta` and `nls_max_iter`. New parameters `alpha` and `l1_ratio` control L1 and L2 regularization, and `shuffle` adds a shuffling step in the `cd` solver. By Tom Dupre la Tour and Mathieu Blondel.

Enhancements

- *manifold.TSNE* now supports approximate optimization via the Barnes-Hut method, leading to much faster fitting. By Christopher Erick Moody. (#4025)
- `cluster.mean_shift_.MeanShift` now supports parallel execution, as implemented in the `mean_shift` function. By Martino Sorbaro.
- *naive_bayes.GaussianNB* now supports fitting with `sample_weight`. By Jan Hendrik Metzen.
- *dummy.DummyClassifier* now supports a prior fitting strategy. By Arnaud Joly.
- Added a `fit_predict` method for `mixture.GMM` and subclasses. By Cory Lorenz.

- Added the *metrics.label_ranking_loss* metric. By Arnaud Joly.
- Added the *metrics.cohen_kappa_score* metric.
- Added a warm_start constructor parameter to the bagging ensemble models to increase the size of the ensemble. By Tim Head.
- Added option to use multi-output regression metrics without averaging. By Konstantin Shmelkov and Michael Eickenberg.
- Added stratify option to cross_validation.train_test_split for stratified splitting. By Miroslav Batchkarov.
- The *tree.export_graphviz* function now supports aesthetic improvements for *tree.DecisionTreeClassifier* and *tree.DecisionTreeRegressor*, including options for coloring nodes by their majority class or impurity, showing variable names, and using node proportions instead of raw sample counts. By Trevor Stephens.
- Improved speed of newton-cg solver in *linear_model.LogisticRegression*, by avoiding loss computation. By Mathieu Blondel and Tom Dupre la Tour.
- The class_weight="auto" heuristic in classifiers supporting class_weight was deprecated and replaced by the class_weight="balanced" option, which has a simpler formula and interpretation. By Hanna Wallach and Andreas Müller.
- Add class_weight parameter to automatically weight samples by class frequency for *linear_model.PassiveAggressiveClassifier*. By Trevor Stephens.
- Added backlinks from the API reference pages to the user guide. By Andreas Müller.
- The labels parameter to *sklearn.metrics.f1_score*, *sklearn.metrics.fbeta_score*, *sklearn.metrics.recall_score* and *sklearn.metrics.precision_score* has been extended. It is now possible to ignore one or more labels, such as where a multiclass problem has a majority class to ignore. By Joel Nothman.
- Add sample_weight support to *linear_model.RidgeClassifier*. By Trevor Stephens.
- Provide an option for sparse output from *sklearn.metrics.pairwise.cosine_similarity*. By Jaidev Deshpande.
- Add minmax_scale to provide a function interface for MinMaxScaler. By Thomas Unterthiner.
- dump_svmlight_file now handles multi-label datasets. By Chih-Wei Chang.
- RCV1 dataset loader (*sklearn.datasets.fetch_rcv1*). By Tom Dupre la Tour.
- The "Wisconsin Breast Cancer" classical two-class classification dataset is now included in scikit-learn, available with sklearn.dataset.load_breast_cancer.
- Upgraded to joblib 0.9.3 to benefit from the new automatic batching of short tasks. This makes it possible for scikit-learn to benefit from parallelism when many very short tasks are executed in parallel, for instance by the grid_search.GridSearchCV meta-estimator with n_jobs > 1 used with a large grid of parameters on a small dataset. By Vlad Niculae, Olivier Grisel and Loic Esteve.
- For more details about changes in joblib 0.9.3 see the release notes: https://github.com/joblib/joblib/blob/master/CHANGES.rst#release-093
- Improved speed (3 times per iteration) of decomposition.DictLearning with coordinate descent method from *linear_model.Lasso*. By Arthur Mensch.
- Parallel processing (threaded) for queries of nearest neighbors (using the ball-tree) by Nikolay Mayorov.
- Allow *datasets.make_multilabel_classification* to output a sparse y. By Kashif Rasul.

- `cluster.DBSCAN` now accepts a sparse matrix of precomputed distances, allowing memory-efficient distance precomputation. By Joel Nothman.
- `tree.DecisionTreeClassifier` now exposes an `apply` method for retrieving the leaf indices samples are predicted as. By Daniel Galvez and Gilles Louppe.
- Speed up decision tree regressors, random forest regressors, extra trees regressors and gradient boosting estimators by computing a proxy of the impurity improvement during the tree growth. The proxy quantity is such that the split that maximizes this value also maximizes the impurity improvement. By Arnaud Joly, Jacob Schreiber and Gilles Louppe.
- Speed up tree based methods by reducing the number of computations needed when computing the impurity measure taking into account linear relationship of the computed statistics. The effect is particularly visible with extra trees and on datasets with categorical or sparse features. By Arnaud Joly.
- `ensemble.GradientBoostingRegressor` and `ensemble.GradientBoostingClassifier` now expose an `apply` method for retrieving the leaf indices each sample ends up in under each try. By Jacob Schreiber.
- Add `sample_weight` support to `linear_model.LinearRegression`. By Sonny Hu. (##4881)
- Add `n_iter_without_progress` to `manifold.TSNE` to control the stopping criterion. By Santi Villalba. (#5186)
- Added optional parameter `random_state` in `linear_model.Ridge` , to set the seed of the pseudo random generator used in `sag` solver. By Tom Dupre la Tour.
- Added optional parameter `warm_start` in `linear_model.LogisticRegression`. If set to True, the solvers `lbfgs`, `newton-cg` and `sag` will be initialized with the coefficients computed in the previous fit. By Tom Dupre la Tour.
- Added `sample_weight` support to `linear_model.LogisticRegression` for the `lbfgs`, `newton-cg`, and `sag` solvers. By Valentin Stolbunov. Support added to the `liblinear` solver. By Manoj Kumar.
- Added optional parameter `presort` to `ensemble.GradientBoostingRegressor` and `ensemble.GradientBoostingClassifier`, keeping default behavior the same. This allows gradient boosters to turn off presorting when building deep trees or using sparse data. By Jacob Schreiber.
- Altered `metrics.roc_curve` to drop unnecessary thresholds by default. By Graham Clenaghan.
- Added `feature_selection.SelectFromModel` meta-transformer which can be used along with estimators that have *coef_* or *feature_importances_* attribute to select important features of the input data. By Maheshakya Wijewardena, Joel Nothman and Manoj Kumar.
- Added `metrics.pairwise.laplacian_kernel`. By Clyde Fare.
- `covariance.GraphLasso` allows separate control of the convergence criterion for the Elastic-Net subproblem via the `enet_tol` parameter.
- Improved verbosity in `decomposition.DictionaryLearning`.
- `ensemble.RandomForestClassifier` and `ensemble.RandomForestRegressor` no longer explicitly store the samples used in bagging, resulting in a much reduced memory footprint for storing random forest models.
- Added `positive` option to `linear_model.Lars` and `linear_model.lars_path` to force coefficients to be positive. (#5131)
- Added the `X_norm_squared` parameter to `metrics.pairwise.euclidean_distances` to provide precomputed squared norms for `X`.
- Added the `fit_predict` method to `pipeline.Pipeline`.

- Added the `preprocessing.min_max_scale` function.

Bug fixes

- Fixed non-determinism in *dummy.DummyClassifier* with sparse multi-label output. By Andreas Müller.
- Fixed the output shape of *linear_model.RANSACRegressor* to `(n_samples, )`. By Andreas Müller.
- Fixed bug in `decomposition.DictLearning` when `n_jobs < 0`. By Andreas Müller.
- Fixed bug where `grid_search.RandomizedSearchCV` could consume a lot of memory for large discrete grids. By Joel Nothman.
- Fixed bug in *linear_model.LogisticRegressionCV* where `penalty` was ignored in the final fit. By Manoj Kumar.
- Fixed bug in `ensemble.forest.ForestClassifier` while computing oob_score and X is a sparse.csc_matrix. By Ankur Ankan.
- All regressors now consistently handle and warn when given `y` that is of shape `(n_samples, 1)`. By Andreas Müller and Henry Lin. (#5431)
- Fix in *cluster.KMeans* cluster reassignment for sparse input by Lars Buitinck.
- Fixed a bug in `lda.LDA` that could cause asymmetric covariance matrices when using shrinkage. By Martin Billinger.
- Fixed `cross_validation.cross_val_predict` for estimators with sparse predictions. By Buddha Prakash.
- Fixed the `predict_proba` method of *linear_model.LogisticRegression* to use soft-max instead of one-vs-rest normalization. By Manoj Kumar. (#5182)
- Fixed the `partial_fit` method of *linear_model.SGDClassifier* when called with `average=True`. By Andrew Lamb. (#5282)
- Dataset fetchers use different filenames under Python 2 and Python 3 to avoid pickling compatibility issues. By Olivier Grisel. (#5355)
- Fixed a bug in *naive_bayes.GaussianNB* which caused classification results to depend on scale. By Jake Vanderplas.
- Fixed temporarily *linear_model.Ridge*, which was incorrect when fitting the intercept in the case of sparse data. The fix automatically changes the solver to 'sag' in this case. #5360 by Tom Dupre la Tour.
- Fixed a performance bug in `decomposition.RandomizedPCA` on data with a large number of features and fewer samples. (#4478) By Andreas Müller, Loic Esteve and Giorgio Patrini.
- Fixed bug in `cross_decomposition.PLS` that yielded unstable and platform dependent output, and failed on *fit_transform*. By Arthur Mensch.
- Fixes to the `Bunch` class used to store datasets.
- Fixed `ensemble.plot_partial_dependence` ignoring the `percentiles` parameter.
- Providing a `set` as vocabulary in `CountVectorizer` no longer leads to inconsistent results when pickling.
- Fixed the conditions on when a precomputed Gram matrix needs to be recomputed in *linear_model.LinearRegression*, *linear_model.OrthogonalMatchingPursuit*, *linear_model.Lasso* and *linear_model.ElasticNet*.
- Fixed inconsistent memory layout in the coordinate descent solver that affected `linear_model.DictionaryLearning` and *covariance.GraphLasso*. (#5337) By Olivier Grisel.

- manifold.LocallyLinearEmbedding no longer ignores the reg parameter.
- Nearest Neighbor estimators with custom distance metrics can now be pickled. (#4362)
- Fixed a bug in pipeline.FeatureUnion where transformer_weights were not properly handled when performing grid-searches.
- Fixed a bug in linear_model.LogisticRegression and linear_model.LogisticRegressionCV when using class_weight='balanced' or class_weight='auto'. By Tom Dupre la Tour.
- Fixed bug #5495 when doing OVR(SVC(decision_function_shape=”ovr”)). Fixed by Elvis Dohmatob.

API changes summary

- Attribute data_min, data_max and data_range in preprocessing.MinMaxScaler are deprecated and won’t be available from 0.19. Instead, the class now exposes data_min_, data_max_ and data_range_. By Giorgio Patrini.
- All Scaler classes now have an scale_ attribute, the feature-wise rescaling applied by their *transform* methods. The old attribute std_ in preprocessing.StandardScaler is deprecated and superseded by scale_; it won’t be available in 0.19. By Giorgio Patrini.
- svm.SVC` and svm.NuSVC now have an decision_function_shape parameter to make their decision function of shape (n_samples, n_classes) by setting decision_function_shape='ovr'. This will be the default behavior starting in 0.19. By Andreas Müller.
- Passing 1D data arrays as input to estimators is now deprecated as it caused confusion in how the array elements should be interpreted as features or as samples. All data arrays are now expected to be explicitly shaped (n_samples, n_features). By Vighnesh Birodkar.
- lda.LDA and qda.QDA have been moved to discriminant_analysis.LinearDiscriminantAnalysis and discriminant_analysis.QuadraticDiscriminantAnalysis.
- The store_covariance and tol parameters have been moved from the fit method to the constructor in discriminant_analysis.LinearDiscriminantAnalysis and the store_covariances and tol parameters have been moved from the fit method to the constructor in discriminant_analysis.QuadraticDiscriminantAnalysis.
- Models inheriting from _LearntSelectorMixin will no longer support the transform methods. (i.e, RandomForests, GradientBoosting, LogisticRegression, DecisionTrees, SVMs and SGD related models). Wrap these models around the metatransfomer feature_selection.SelectFromModel to remove features (according to coefs_ or *feature_importances_*) which are below a certain threshold value instead.
- cluster.KMeans re-runs cluster-assignments in case of non-convergence, to ensure consistency of predict(X) and labels_. By Vighnesh Birodkar.
- Classifier and Regressor models are now tagged as such using the _estimator_type attribute.
- Cross-validation iterators always provide indices into training and test set, not boolean masks.
- The decision_function on all regressors was deprecated and will be removed in 0.19. Use predict instead.
- datasets.load_lfw_pairs is deprecated and will be removed in 0.19. Use datasets.fetch_lfw_pairs instead.
- The deprecated hmm module was removed.
- The deprecated Bootstrap cross-validation iterator was removed.

- The deprecated Ward and WardAgglomerative classes have been removed. Use clustering.AgglomerativeClustering instead.
- cross_validation.check_cv is now a public function.
- The property residues_ of *linear_model.LinearRegression* is deprecated and will be removed in 0.19.
- The deprecated n_jobs parameter of *linear_model.LinearRegression* has been moved to the constructor.
- Removed deprecated class_weight parameter from *linear_model.SGDClassifier*'s fit method. Use the construction parameter instead.
- The deprecated support for the sequence of sequences (or list of lists) multilabel format was removed. To convert to and from the supported binary indicator matrix format, use *MultiLabelBinarizer*.
- The behavior of calling the inverse_transform method of Pipeline.pipeline will change in 0.19. It will no longer reshape one-dimensional input to two-dimensional input.
- The deprecated attributes indicator_matrix_, multilabel_ and classes_ of *preprocessing.LabelBinarizer* were removed.
- Using gamma=0 in *svm.SVC* and *svm.SVR* to automatically set the gamma to 1. / n_features is deprecated and will be removed in 0.19. Use gamma="auto" instead.

Code Contributors

Aaron Schumacher, Adithya Ganesh, akitty, Alexandre Gramfort, Alexey Grigorev, Ali Baharev, Allen Riddell, Ando Saabas, Andreas Mueller, Andrew Lamb, Anish Shah, Ankur Ankan, Anthony Erlinger, Ari Rouvinen, Arnaud Joly, Arnaud Rachez, Arthur Mensch, banilo, Barmaley.exe, benjaminirving, Boyuan Deng, Brett Naul, Brian McFee, Buddha Prakash, Chi Zhang, Chih-Wei Chang, Christof Angermueller, Christoph Gohlke, Christophe Bourguignat, Christopher Erick Moody, Chyi-Kwei Yau, Cindy Sridharan, CJ Carey, Clyde-fare, Cory Lorenz, Dan Blanchard, Daniel Galvez, Daniel Kronovet, Danny Sullivan, Data1010, David, David D Lowe, David Dotson, djipey, Dmitry Spikhalskiy, Donne Martin, Dougal J. Sutherland, Dougal Sutherland, edson duarte, Eduardo Caro, Eric Larson, Eric Martin, Erich Schubert, Fernando Carrillo, Frank C. Eckert, Frank Zalkow, Gael Varoquaux, Ganiev Ibraim, Gilles Louppe, Giorgio Patrini, giorgiop, Graham Clenaghan, Gryllos Prokopis, gwulfs, Henry Lin, Hsuan-Tien Lin, Immanuel Bayer, Ishank Gulati, Jack Martin, Jacob Schreiber, Jaidev Deshpande, Jake Vanderplas, Jan Hendrik Metzen, Jean Kossaifi, Jeffrey04, Jeremy, jfraj, Jiali Mei, Joe Jevnik, Joel Nothman, John Kirkham, John Wittenauer, Joseph, Joshua Loyal, Jungkook Park, KamalakerDadi, Kashif Rasul, Keith Goodman, Kian Ho, Konstantin Shmelkov, Kyler Brown, Lars Buitinck, Lilian Besson, Loic Esteve, Louis Tiao, maheshakya, Maheshakya Wijewardena, Manoj Kumar, MarkTab marktab.net, Martin Ku, Martin Spacek, MartinBpr, martinosorb, MaryanMorel, Masafumi Oyamada, Mathieu Blondel, Matt Krump, Matti Lyra, Maxim Kolganov, mbillinger, mhg, Michael Heilman, Michael Patterson, Miroslav Batchkarov, Nelle Varoquaux, Nicolas, Nikolay Mayorov, Olivier Grisel, Omer Katz, Óscar Nájera, Pauli Virtanen, Peter Fischer, Peter Prettenhofer, Phil Roth, pianomania, Preston Parry, Raghav RV, Rob Zinkov, Robert Layton, Rohan Ramanath, Saket Choudhary, Sam Zhang, santi, saurabh.bansod, scls19fr, Sebastian Raschka, Sebastian Saeger, Shivan Sornarajah, SimonPL, sinhrks, Skipper Seabold, Sonny Hu, sseg, Stephen Hoover, Steven De Gryze, Steven Seguin, Theodore Vasiloudis, Thomas Unterthiner, Tiago Freitas Pereira, Tian Wang, Tim Head, Timothy Hopper, tokoroten, Tom Dupré la Tour, Trevor Stephens, Valentin Stolbunov, Vighnesh Birodkar, Vinayak Mehta, Vincent, Vincent Michel, vstolbunov, wangz10, Wei Xue, Yucheng Low, Yury Zhauniarovich, Zac Stewart, zhai_pro, Zichen Wang

1.17.9 Version 0.16.1

April 14, 2015

Changelog

Bug fixes

- Allow input data larger than block_size in *covariance.LedoitWolf* by Andreas Müller.
- Fix a bug in *isotonic.IsotonicRegression* deduplication that caused unstable result in *calibration.CalibratedClassifierCV* by Jan Hendrik Metzen.
- Fix sorting of labels in func:*preprocessing.label_binarize* by Michael Heilman.
- Fix several stability and convergence issues in *cross_decomposition.CCA* and *cross_decomposition.PLSCanonical* by Andreas Müller
- Fix a bug in *cluster.KMeans* when precompute_distances=False on fortran-ordered data.
- Fix a speed regression in *ensemble.RandomForestClassifier*'s predict and predict_proba by Andreas Müller.
- Fix a regression where utils.shuffle converted lists and dataframes to arrays, by Olivier Grisel

1.17.10 Version 0.16

March 26, 2015

Highlights

- Speed improvements (notably in *cluster.DBSCAN*), reduced memory requirements, bug-fixes and better default settings.
- Multinomial Logistic regression and a path algorithm in *linear_model.LogisticRegressionCV*.
- Out-of core learning of PCA via *decomposition.IncrementalPCA*.
- Probability callibration of classifiers using *calibration.CalibratedClassifierCV*.
- *cluster.Birch* clustering method for large-scale datasets.
- Scalable approximate nearest neighbors search with Locality-sensitive hashing forests in neighbors.LSHForest.
- Improved error messages and better validation when using malformed input data.
- More robust integration with pandas dataframes.

Changelog

New features

- The new neighbors.LSHForest implements locality-sensitive hashing for approximate nearest neighbors search. By Maheshakya Wijewardena.
- Added *svm.LinearSVR*. This class uses the liblinear implementation of Support Vector Regression which is much faster for large sample sizes than *svm.SVR* with linear kernel. By Fabian Pedregosa and Qiang Luo.
- Incremental fit for *GaussianNB*.
- Added sample_weight support to *dummy.DummyClassifier* and *dummy.DummyRegressor*. By Arnaud Joly.

- Added the *metrics.label_ranking_average_precision_score* metrics. By Arnaud Joly.
- Add the *metrics.coverage_error* metrics. By Arnaud Joly.
- Added *linear_model.LogisticRegressionCV*. By Manoj Kumar, Fabian Pedregosa, Gael Varoquaux and Alexandre Gramfort.
- Added warm_start constructor parameter to make it possible for any trained forest model to grow additional trees incrementally. By Laurent Direr.
- Added sample_weight support to *ensemble.GradientBoostingClassifier* and *ensemble.GradientBoostingRegressor*. By Peter Prettenhofer.
- Added *decomposition.IncrementalPCA*, an implementation of the PCA algorithm that supports out-of-core learning with a partial_fit method. By Kyle Kastner.
- Averaged SGD for *SGDClassifier* and *SGDRegressor* By Danny Sullivan.
- Added cross_val_predict function which computes cross-validated estimates. By Luis Pedro Coelho
- Added *linear_model.TheilSenRegressor*, a robust generalized-median-based estimator. By Florian Wilhelm.
- Added *metrics.median_absolute_error*, a robust metric. By Gael Varoquaux and Florian Wilhelm.
- Add *cluster.Birch*, an online clustering algorithm. By Manoj Kumar, Alexandre Gramfort and Joel Nothman.
- Added shrinkage support to *discriminant_analysis.LinearDiscriminantAnalysis* using two new solvers. By Clemens Brunner and Martin Billinger.
- Added *kernel_ridge.KernelRidge*, an implementation of kernelized ridge regression. By Mathieu Blondel and Jan Hendrik Metzen.
- All solvers in *linear_model.Ridge* now support *sample_weight*. By Mathieu Blondel.
- Added cross_validation.PredefinedSplit cross-validation for fixed user-provided cross-validation folds. By Thomas Unterthiner.
- Added *calibration.CalibratedClassifierCV*, an approach for calibrating the predicted probabilities of a classifier. By Alexandre Gramfort, Jan Hendrik Metzen, Mathieu Blondel and Balazs Kegl.

Enhancements

- Add option return_distance in hierarchical.ward_tree to return distances between nodes for both structured and unstructured versions of the algorithm. By Matteo Visconti di Oleggio Castello. The same option was added in hierarchical.linkage_tree. By Manoj Kumar
- Add support for sample weights in scorer objects. Metrics with sample weight support will automatically benefit from it. By Noel Dawe and Vlad Niculae.
- Added newton-cg and lbfgs solver support in *linear_model.LogisticRegression*. By Manoj Kumar.
- Add selection="random" parameter to implement stochastic coordinate descent for *linear_model.Lasso*, *linear_model.ElasticNet* and related. By Manoj Kumar.
- Add sample_weight parameter to *metrics.jaccard_similarity_score* and *metrics.log_loss*. By Jatin Shah.
- Support sparse multilabel indicator representation in *preprocessing.LabelBinarizer* and *multiclass.OneVsRestClassifier* (by Hamzeh Alsalhi with thanks to Rohit Sivaprasad), as well as evaluation metrics (by Joel Nothman).

- Add sample_weight parameter to *metrics.jaccard_similarity_score*. By Jatin Shah.
- Add support for multiclass in *metrics.hinge_loss*. Added labels=None as optional parameter. By Saurabh Jha.
- Add sample_weight parameter to *metrics.hinge_loss*. By Saurabh Jha.
- Add multi_class="multinomial" option in *linear_model.LogisticRegression* to implement a Logistic Regression solver that minimizes the cross-entropy or multinomial loss instead of the default One-vs-Rest setting. Supports lbfgs and newton-cg solvers. By Lars Buitinck and Manoj Kumar. Solver option newton-cg by Simon Wu.
- DictVectorizer can now perform fit_transform on an iterable in a single pass, when giving the option sort=False. By Dan Blanchard.
- GridSearchCV and RandomizedSearchCV can now be configured to work with estimators that may fail and raise errors on individual folds. This option is controlled by the error_score parameter. This does not affect errors raised on re-fit. By Michal Romaniuk.
- Add digits parameter to *metrics.classification_report* to allow report to show different precision of floating point numbers. By Ian Gilmore.
- Add a quantile prediction strategy to the *dummy.DummyRegressor*. By Aaron Staple.
- Add handle_unknown option to *preprocessing.OneHotEncoder* to handle unknown categorical features more gracefully during transform. By Manoj Kumar.
- Added support for sparse input data to decision trees and their ensembles. By Fares Hedyati and Arnaud Joly.
- Optimized *cluster.AffinityPropagation* by reducing the number of memory allocations of large temporary data-structures. By Antony Lee.
- Parellization of the computation of feature importances in random forest. By Olivier Grisel and Arnaud Joly.
- Add n_iter_ attribute to estimators that accept a max_iter attribute in their constructor. By Manoj Kumar.
- Added decision function for *multiclass.OneVsOneClassifier* By Raghav RV and Kyle Beauchamp.
- *neighbors.kneighbors_graph* and radius_neighbors_graph support non-Euclidean metrics. By Manoj Kumar
- Parameter connectivity in *cluster.AgglomerativeClustering* and family now accept callables that return a connectivity matrix. By Manoj Kumar.
- Sparse support for paired_distances. By Joel Nothman.
- *cluster.DBSCAN* now supports sparse input and sample weights and has been optimized: the inner loop has been rewritten in Cython and radius neighbors queries are now computed in batch. By Joel Nothman and Lars Buitinck.
- Add class_weight parameter to automatically weight samples by class frequency for *ensemble.RandomForestClassifier*, *tree.DecisionTreeClassifier*, *ensemble.ExtraTreesClassifier* and *tree.ExtraTreeClassifier*. By Trevor Stephens.
- grid_search.RandomizedSearchCV now does sampling without replacement if all parameters are given as lists. By Andreas Müller.
- Parallelized calculation of pairwise_distances is now supported for scipy metrics and custom callables. By Joel Nothman.
- Allow the fitting and scoring of all clustering algorithms in *pipeline.Pipeline*. By Andreas Müller.
- More robust seeding and improved error messages in *cluster.MeanShift* by Andreas Müller.

- Make the stopping criterion for mixture.GMM, mixture.DPGMM and mixture.VBGMM less dependent on the number of samples by thresholding the average log-likelihood change instead of its sum over all samples. By Hervé Bredin.
- The outcome of *manifold.spectral_embedding* was made deterministic by flipping the sign of eigenvectors. By Hasil Sharma.
- Significant performance and memory usage improvements in *preprocessing.PolynomialFeatures*. By Eric Martin.
- Numerical stability improvements for *preprocessing.StandardScaler* and *preprocessing.scale*. By Nicolas Goix
- *svm.SVC* fitted on sparse input now implements decision_function. By Rob Zinkov and Andreas Müller.
- cross_validation.train_test_split now preserves the input type, instead of converting to numpy arrays.

Documentation improvements

- Added example of using FeatureUnion for heterogeneous input. By Matt Terry
- Documentation on scorers was improved, to highlight the handling of loss functions. By Matt Pico.
- A discrepancy between liblinear output and scikit-learn's wrappers is now noted. By Manoj Kumar.
- Improved documentation generation: examples referring to a class or function are now shown in a gallery on the class/function's API reference page. By Joel Nothman.
- More explicit documentation of sample generators and of data transformation. By Joel Nothman.
- *sklearn.neighbors.BallTree* and *sklearn.neighbors.KDTree* used to point to empty pages stating that they are aliases of BinaryTree. This has been fixed to show the correct class docs. By Manoj Kumar.
- Added silhouette plots for analysis of KMeans clustering using *metrics.silhouette_samples* and *metrics.silhouette_score*. See *Selecting the number of clusters with silhouette analysis on KMeans clustering*

Bug fixes

- Metaestimators now support ducktyping for the presence of decision_function, predict_proba and other methods. This fixes behavior of grid_search.GridSearchCV, grid_search.RandomizedSearchCV, *pipeline.Pipeline*, *feature_selection.RFE*, *feature_selection.RFECV* when nested. By Joel Nothman
- The scoring attribute of grid-search and cross-validation methods is no longer ignored when a grid_search.GridSearchCV is given as a base estimator or the base estimator doesn't have predict.
- The function hierarchical.ward_tree now returns the children in the same order for both the structured and unstructured versions. By Matteo Visconti di Oleggio Castello.
- *feature_selection.RFECV* now correctly handles cases when step is not equal to 1. By Nikolay Mayorov
- The *decomposition.PCA* now undoes whitening in its inverse_transform. Also, its components_ now always have unit length. By Michael Eickenberg.
- Fix incomplete download of the dataset when datasets.download_20newsgroups is called. By Manoj Kumar.

- Various fixes to the Gaussian processes subpackage by Vincent Dubourg and Jan Hendrik Metzen.
- Calling `partial_fit` with `class_weight=='auto'` throws an appropriate error message and suggests a work around. By Danny Sullivan.
- *RBFSampler* with `gamma=g` formerly approximated *rbf_kernel* with `gamma=g/2.`; the definition of `gamma` is now consistent, which may substantially change your results if you use a fixed value. (If you cross-validated over `gamma`, it probably doesn't matter too much.) By Dougal Sutherland.
- Pipeline object delegate the `classes_` attribute to the underlying estimator. It allows, for instance, to make bagging of a pipeline object. By Arnaud Joly
- *neighbors.NearestCentroid* now uses the median as the centroid when metric is set to `manhattan`. It was using the mean before. By Manoj Kumar
- Fix numerical stability issues in *linear_model.SGDClassifier* and *linear_model.SGDRegressor* by clipping large gradients and ensuring that weight decay rescaling is always positive (for large l2 regularization and large learning rate values). By Olivier Grisel
- When `compute_full_tree` is set to "auto", the full tree is built when n_clusters is high and is early stopped when n_clusters is low, while the behavior should be vice-versa in *cluster.AgglomerativeClustering* (and friends). This has been fixed By Manoj Kumar
- Fix lazy centering of data in *linear_model.enet_path* and *linear_model.lasso_path*. It was centered around one. It has been changed to be centered around the origin. By Manoj Kumar
- Fix handling of precomputed affinity matrices in *cluster.AgglomerativeClustering* when using connectivity constraints. By Cathy Deng
- Correct `partial_fit` handling of `class_prior` for *sklearn.naive_bayes.MultinomialNB* and *sklearn.naive_bayes.BernoulliNB*. By Trevor Stephens.
- Fixed a crash in *metrics.precision_recall_fscore_support* when using unsorted `labels` in the multi-label setting. By Andreas Müller.
- Avoid skipping the first nearest neighbor in the methods `radius_neighbors`, `kneighbors`, `kneighbors_graph` and `radius_neighbors_graph` in *sklearn.neighbors.NearestNeighbors* and family, when the query data is not the same as fit data. By Manoj Kumar.
- Fix log-density calculation in the `mixture.GMM` with tied covariance. By Will Dawson
- Fixed a scaling error in *feature_selection.SelectFdr* where a factor `n_features` was missing. By Andrew Tulloch
- Fix zero division in *neighbors.KNeighborsRegressor* and related classes when using distance weighting and having identical data points. By Garret-R.
- Fixed round off errors with non positive-definite covariance matrices in GMM. By Alexis Mignon.
- Fixed a error in the computation of conditional probabilities in *naive_bayes.BernoulliNB*. By Hanna Wallach.
- Make the method `radius_neighbors` of *neighbors.NearestNeighbors* return the samples lying on the boundary for `algorithm='brute'`. By Yan Yi.
- Flip sign of `dual_coef_` of *svm.SVC* to make it consistent with the documentation and `decision_function`. By Artem Sobolev.
- Fixed handling of ties in *isotonic.IsotonicRegression*. We now use the weighted average of targets (secondary method). By Andreas Müller and Michael Bommarito.

API changes summary

- GridSearchCV and cross_val_score and other meta-estimators don't convert pandas DataFrames into arrays any more, allowing DataFrame specific operations in custom estimators.
- multiclass.fit_ovr, multiclass.predict_ovr, predict_proba_ovr, multiclass.fit_ovo, multiclass.predict_ovo, multiclass.fit_ecoc and multiclass.predict_ecoc are deprecated. Use the underlying estimators instead.
- Nearest neighbors estimators used to take arbitrary keyword arguments and pass these to their distance metric. This will no longer be supported in scikit-learn 0.18; use the metric_params argument instead.
- ***n_jobs* parameter of the fit method shifted to the constructor of the** LinearRegression class.
- The predict_proba method of *multiclass.OneVsRestClassifier* now returns two probabilities per sample in the multiclass case; this is consistent with other estimators and with the method's documentation, but previous versions accidentally returned only the positive probability. Fixed by Will Lamond and Lars Buitinck.
- Change default value of precompute in ElasticNet and Lasso to False. Setting precompute to "auto" was found to be slower when n_samples > n_features since the computation of the Gram matrix is computationally expensive and outweighs the benefit of fitting the Gram for just one alpha. precompute="auto" is now deprecated and will be removed in 0.18 By Manoj Kumar.
- Expose positive option in *linear_model.enet_path* and *linear_model.enet_path* which constrains coefficients to be positive. By Manoj Kumar.
- Users should now supply an explicit average parameter to *sklearn.metrics.f1_score*, *sklearn.metrics.fbeta_score*, *sklearn.metrics.recall_score* and *sklearn.metrics.precision_score* when performing multiclass or multilabel (i.e. not binary) classification. By Joel Nothman.
- *scoring* parameter for cross validation now accepts 'f1_micro', 'f1_macro' or 'f1_weighted'. 'f1' is now for binary classification only. Similar changes apply to 'precision' and 'recall'. By Joel Nothman.
- The fit_intercept, normalize and return_models parameters in *linear_model.enet_path* and *linear_model.lasso_path* have been removed. They were deprecated since 0.14
- From now onwards, all estimators will uniformly raise NotFittedError (utils.validation.NotFittedError), when any of the predict like methods are called before the model is fit. By Raghav RV.
- Input data validation was refactored for more consistent input validation. The check_arrays function was replaced by check_array and check_X_y. By Andreas Müller.
- Allow X=None in the methods radius_neighbors, kneighbors, kneighbors_graph and radius_neighbors_graph in *sklearn.neighbors.NearestNeighbors* and family. If set to None, then for every sample this avoids setting the sample itself as the first nearest neighbor. By Manoj Kumar.
- Add parameter include_self in *neighbors.kneighbors_graph* and *neighbors.radius_neighbors_graph* which has to be explicitly set by the user. If set to True, then the sample itself is considered as the first nearest neighbor.
- thresh parameter is deprecated in favor of new tol parameter in GMM, DPGMM and VBGMM. See Enhancements section for details. By Hervé Bredin.
- Estimators will treat input with dtype object as numeric when possible. By Andreas Müller
- Estimators now raise ValueError consistently when fitted on empty data (less than 1 sample or less than 1 feature for 2D input). By Olivier Grisel.

- The shuffle option of linear_model.SGDClassifier, linear_model.SGDRegressor, linear_model.Perceptron, linear_model.PassiveAggressiveClassifier and linear_model.PassiveAggressiveRegressor now defaults to True.
- cluster.DBSCAN now uses a deterministic initialization. The *random_state* parameter is deprecated. By Erich Schubert.

Code Contributors

A. Flaxman, Aaron Schumacher, Aaron Staple, abhishek thakur, Akshay, akshayah3, Aldrian Obaja, Alexander Fabisch, Alexandre Gramfort, Alexis Mignon, Anders Aagaard, Andreas Mueller, Andreas van Cranenburgh, Andrew Tulloch, Andrew Walker, Antony Lee, Arnaud Joly, banilo, Barmaley.exe, Ben Davies, Benedikt Koehler, bhsu, Boris Feld, Borja Ayerdi, Boyuan Deng, Brent Pedersen, Brian Wignall, Brooke Osborn, Calvin Giles, Cathy Deng, Celeo, cgohlke, chebee7i, Christian Stade-Schuldt, Christof Angermueller, Chyi-Kwei Yau, CJ Carey, Clemens Brunner, Daiki Aminaka, Dan Blanchard, danfrankj, Danny Sullivan, David Fletcher, Dmitrijs Milajevs, Dougal J. Sutherland, Erich Schubert, Fabian Pedregosa, Florian Wilhelm, floydsoft, Félix-Antoine Fortin, Gael Varoquaux, Garrett-R, Gilles Louppe, gpassino, gwulfs, Hampus Bengtsson, Hamzeh Alsalhi, Hanna Wallach, Harry Mavroforakis, Hasil Sharma, Helder, Herve Bredin, Hsiang-Fu Yu, Hugues SALAMIN, Ian Gilmore, Ilambharathi Kanniah, Imran Haque, isms, Jake VanderPlas, Jan Dlabal, Jan Hendrik Metzen, Jatin Shah, Javier López Peña, jdcaballero, Jean Kossaifi, Jeff Hammerbacher, Joel Nothman, Jonathan Helmus, Joseph, Kaicheng Zhang, Kevin Markham, Kyle Beauchamp, Kyle Kastner, Lagacherie Matthieu, Lars Buitinck, Laurent Direr, leepei, Loic Esteve, Luis Pedro Coelho, Lukas Michelbacher, maheshakya, Manoj Kumar, Manuel, Mario Michael Krell, Martin, Martin Billinger, Martin Ku, Mateusz Susik, Mathieu Blondel, Matt Pico, Matt Terry, Matteo Visconti dOC, Matti Lyra, Max Linke, Mehdi Cherti, Michael Bommarito, Michael Eickenberg, Michal Romaniuk, MLG, mr.Shu, Nelle Varoquaux, Nicola Montecchio, Nicolas, Nikolay Mayorov, Noel Dawe, Okal Billy, Olivier Grisel, Óscar Nájera, Paolo Puggioni, Peter Prettenhofer, Pratap Vardhan, pvnguyen, queqichao, Rafael Carrascosa, Raghav R V, Rahiel Kasim, Randall Mason, Rob Zinkov, Robert Bradshaw, Saket Choudhary, Sam Nicholls, Samuel Charron, Saurabh Jha, sethdandridge, sinhrks, snuderl, Stefan Otte, Stefan van der Walt, Steve Tjoa, swu, Sylvain Zimmer, tejesh95, terrycojones, Thomas Delteil, Thomas Unterthiner, Tomas Kazmar, trevorstephens, tttthomasssss, Tzu-Ming Kuo, ugurcaliskan, ugurthemaster, Vinayak Mehta, Vincent Dubourg, Vjacheslav Murashkin, Vlad Niculae, wadawson, Wei Xue, Will Lamond, Wu Jiang, x0l, Xinfan Meng, Yan Yi, Yu-Chin

1.17.11 Version 0.15.2

September 4, 2014

Bug fixes

- Fixed handling of the p parameter of the Minkowski distance that was previously ignored in nearest neighbors models. By Nikolay Mayorov.
- Fixed duplicated alphas in linear_model.LassoLars with early stopping on 32 bit Python. By Olivier Grisel and Fabian Pedregosa.
- Fixed the build under Windows when scikit-learn is built with MSVC while NumPy is built with MinGW. By Olivier Grisel and Federico Vaggi.
- Fixed an array index overflow bug in the coordinate descent solver. By Gael Varoquaux.
- Better handling of numpy 1.9 deprecation warnings. By Gael Varoquaux.
- Removed unnecessary data copy in cluster.KMeans. By Gael Varoquaux.
- Explicitly close open files to avoid ResourceWarnings under Python 3. By Calvin Giles.

- The `transform` of *`discriminant_analysis.LinearDiscriminantAnalysis`* now projects the input on the most discriminant directions. By Martin Billinger.
- Fixed potential overflow in `_tree.safe_realloc` by Lars Buitinck.
- Performance optimization in *`isotonic.IsotonicRegression`*. By Robert Bradshaw.
- `nose` is non-longer a runtime dependency to import `sklearn`, only for running the tests. By Joel Nothman.
- Many documentation and website fixes by Joel Nothman, Lars Buitinck Matt Pico, and others.

1.17.12 Version 0.15.1

August 1, 2014

Bug fixes

- Made `cross_validation.cross_val_score` use `cross_validation.KFold` instead of `cross_validation.StratifiedKFold` on multi-output classification problems. By Nikolay Mayorov.
- Support unseen labels *`preprocessing.LabelBinarizer`* to restore the default behavior of 0.14.1 for backward compatibility. By Hamzeh Alsalhi.
- Fixed the *`cluster.KMeans`* stopping criterion that prevented early convergence detection. By Edward Raff and Gael Varoquaux.
- Fixed the behavior of *`multiclass.OneVsOneClassifier`*. in case of ties at the per-class vote level by computing the correct per-class sum of prediction scores. By Andreas Müller.
- Made `cross_validation.cross_val_score` and `grid_search.GridSearchCV` accept Python lists as input data. This is especially useful for cross-validation and model selection of text processing pipelines. By Andreas Müller.
- Fixed data input checks of most estimators to accept input data that implements the NumPy `__array__` protocol. This is the case for for `pandas.Series` and `pandas.DataFrame` in recent versions of pandas. By Gael Varoquaux.
- Fixed a regression for *`linear_model.SGDClassifier`* with `class_weight="auto"` on data with non-contiguous labels. By Olivier Grisel.

1.17.13 Version 0.15

July 15, 2014

Highlights

- Many speed and memory improvements all across the code
- Huge speed and memory improvements to random forests (and extra trees) that also benefit better from parallel computing.
- Incremental fit to *`BernoulliRBM`*
- Added *`cluster.AgglomerativeClustering`* for hierarchical agglomerative clustering with average linkage, complete linkage and ward strategies.
- Added *`linear_model.RANSACRegressor`* for robust regression models.

- Added dimensionality reduction with *manifold.TSNE* which can be used to visualize high-dimensional data.

Changelog

New features

- Added *ensemble.BaggingClassifier* and *ensemble.BaggingRegressor* meta-estimators for ensembling any kind of base estimator. See the *Bagging* section of the user guide for details and examples. By Gilles Louppe.
- New unsupervised feature selection algorithm *feature_selection.VarianceThreshold*, by Lars Buitinck.
- Added *linear_model.RANSACRegressor* meta-estimator for the robust fitting of regression models. By Johannes Schönberger.
- Added *cluster.AgglomerativeClustering* for hierarchical agglomerative clustering with average linkage, complete linkage and ward strategies, by Nelle Varoquaux and Gael Varoquaux.
- Shorthand constructors *pipeline.make_pipeline* and *pipeline.make_union* were added by Lars Buitinck.
- Shuffle option for `cross_validation.StratifiedKFold`. By Jeffrey Blackburne.
- Incremental learning (`partial_fit`) for Gaussian Naive Bayes by Imran Haque.
- Added `partial_fit` to *BernoulliRBM* By Danny Sullivan.
- Added `learning_curve` utility to chart performance with respect to training size. See *Plotting Learning Curves*. By Alexander Fabisch.
- Add positive option in *LassoCV* and *ElasticNetCV*. By Brian Wignall and Alexandre Gramfort.
- Added *linear_model.MultiTaskElasticNetCV* and *linear_model.MultiTaskLassoCV*. By Manoj Kumar.
- Added *manifold.TSNE*. By Alexander Fabisch.

Enhancements

- Add sparse input support to *ensemble.AdaBoostClassifier* and *ensemble.AdaBoostRegressor* meta-estimators. By Hamzeh Alsalhi.
- Memory improvements of decision trees, by Arnaud Joly.
- Decision trees can now be built in best-first manner by using `max_leaf_nodes` as the stopping criteria. Refactored the tree code to use either a stack or a priority queue for tree building. By Peter Prettenhofer and Gilles Louppe.
- Decision trees can now be fitted on fortran- and c-style arrays, and non-continuous arrays without the need to make a copy. If the input array has a different dtype than `np.float32`, a fortran- style copy will be made since fortran-style memory layout has speed advantages. By Peter Prettenhofer and Gilles Louppe.
- Speed improvement of regression trees by optimizing the the computation of the mean square error criterion. This lead to speed improvement of the tree, forest and gradient boosting tree modules. By Arnaud Joly
- The `img_to_graph` and `grid_tograph` functions in *sklearn.feature_extraction.image* now return `np.ndarray` instead of `np.matrix` when `return_as=np.ndarray`. See the Notes section for more information on compatibility.

- Changed the internal storage of decision trees to use a struct array. This fixed some small bugs, while improving code and providing a small speed gain. By Joel Nothman.
- Reduce memory usage and overhead when fitting and predicting with forests of randomized trees in parallel with `n_jobs != 1` by leveraging new threading backend of joblib 0.8 and releasing the GIL in the tree fitting Cython code. By Olivier Grisel and Gilles Louppe.
- Speed improvement of the `sklearn.ensemble.gradient_boosting` module. By Gilles Louppe and Peter Prettenhofer.
- Various enhancements to the `sklearn.ensemble.gradient_boosting` module: a `warm_start` argument to fit additional trees, a `max_leaf_nodes` argument to fit GBM style trees, a `monitor` fit argument to inspect the estimator during training, and refactoring of the verbose code. By Peter Prettenhofer.
- Faster `sklearn.ensemble.ExtraTrees` by caching feature values. By Arnaud Joly.
- Faster depth-based tree building algorithm such as decision tree, random forest, extra trees or gradient tree boosting (with depth based growing strategy) by avoiding trying to split on found constant features in the sample subset. By Arnaud Joly.
- Add `min_weight_fraction_leaf` pre-pruning parameter to tree-based methods: the minimum weighted fraction of the input samples required to be at a leaf node. By Noel Dawe.
- Added *`metrics.pairwise_distances_argmin_min`*, by Philippe Gervais.
- Added predict method to *`cluster.AffinityPropagation`* and *`cluster.MeanShift`*, by Mathieu Blondel.
- Vector and matrix multiplications have been optimised throughout the library by Denis Engemann, and Alexandre Gramfort. In particular, they should take less memory with older NumPy versions (prior to 1.7.2).
- Precision-recall and ROC examples now use train_test_split, and have more explanation of why these metrics are useful. By Kyle Kastner
- The training algorithm for *`decomposition.NMF`* is faster for sparse matrices and has much lower memory complexity, meaning it will scale up gracefully to large datasets. By Lars Buitinck.
- Added svd_method option with default value to “randomized” to *`decomposition.FactorAnalysis`* to save memory and significantly speedup computation by Denis Engemann, and Alexandre Gramfort.
- Changed `cross_validation.StratifiedKFold` to try and preserve as much of the original ordering of samples as possible so as not to hide overfitting on datasets with a non-negligible level of samples dependency. By Daniel Nouri and Olivier Grisel.
- Add multi-output support to `gaussian_process.GaussianProcess` by John Novak.
- Support for precomputed distance matrices in nearest neighbor estimators by Robert Layton and Joel Nothman.
- Norm computations optimized for NumPy 1.6 and later versions by Lars Buitinck. In particular, the k-means algorithm no longer needs a temporary data structure the size of its input.
- *`dummy.DummyClassifier`* can now be used to predict a constant output value. By Manoj Kumar.
- *`dummy.DummyRegressor`* has now a strategy parameter which allows to predict the mean, the median of the training set or a constant output value. By Maheshakya Wijewardena.
- Multi-label classification output in multilabel indicator format is now supported by *`metrics.roc_auc_score`* and *`metrics.average_precision_score`* by Arnaud Joly.
- Significant performance improvements (more than 100x speedup for large problems) in *`isotonic.IsotonicRegression`* by Andrew Tulloch.
- Speed and memory usage improvements to the SGD algorithm for linear models: it now uses threads, not separate processes, when `n_jobs>1`. By Lars Buitinck.

- Grid search and cross validation allow NaNs in the input arrays so that preprocessors such as *preprocessing.Imputer* can be trained within the cross validation loop, avoiding potentially skewed results.
- Ridge regression can now deal with sample weights in feature space (only sample space until then). By Michael Eickenberg. Both solutions are provided by the Cholesky solver.
- Several classification and regression metrics now support weighted samples with the new sample_weight argument: *metrics.accuracy_score*, *metrics.zero_one_loss*, *metrics.precision_score*, *metrics.average_precision_score*, *metrics.f1_score*, *metrics.fbeta_score*, *metrics.recall_score*, *metrics.roc_auc_score*, *metrics.explained_variance_score*, *metrics.mean_squared_error*, *metrics.mean_absolute_error*, *metrics.r2_score*. By Noel Dawe.
- Speed up of the sample generator *datasets.make_multilabel_classification*. By Joel Nothman.

Documentation improvements

- The *Working With Text Data* tutorial has now been worked in to the main documentation's tutorial section. Includes exercises and skeletons for tutorial presentation. Original tutorial created by several authors including Olivier Grisel, Lars Buitinck and many others. Tutorial integration into the scikit-learn documentation by Jaques Grobler
- Added *Computational Performance* documentation. Discussion and examples of prediction latency / throughput and different factors that have influence over speed. Additional tips for building faster models and choosing a relevant compromise between speed and predictive power. By Eustache Diemert.

Bug fixes

- Fixed bug in *decomposition.MiniBatchDictionaryLearning* : partial_fit was not working properly.
- Fixed bug in linear_model.stochastic_gradient : l1_ratio was used as (1.0 - l1_ratio) .
- Fixed bug in *multiclass.OneVsOneClassifier* with string labels
- Fixed a bug in *LassoCV* and *ElasticNetCV*: they would not pre-compute the Gram matrix with precompute=True or precompute="auto" and n_samples > n_features. By Manoj Kumar.
- Fixed incorrect estimation of the degrees of freedom in *feature_selection.f_regression* when variates are not centered. By Virgile Fritsch.
- Fixed a race condition in parallel processing with pre_dispatch != "all" (for instance, in cross_val_score). By Olivier Grisel.
- Raise error in *cluster.FeatureAgglomeration* and cluster.WardAgglomeration when no samples are given, rather than returning meaningless clustering.
- Fixed bug in gradient_boosting.GradientBoostingRegressor with loss='huber': gamma might have not been initialized.
- Fixed feature importances as computed with a forest of randomized trees when fit with sample_weight != None and/or with bootstrap=True. By Gilles Louppe.

API changes summary

- sklearn.hmm is deprecated. Its removal is planned for the 0.17 release.
- Use of covariance.EllipticEnvelop has now been removed after deprecation. Please use *covariance.EllipticEnvelope* instead.
- cluster.Ward is deprecated. Use *cluster.AgglomerativeClustering* instead.
- cluster.WardClustering is deprecated. Use
- *cluster.AgglomerativeClustering* instead.
- cross_validation.Bootstrap is deprecated. cross_validation.KFold or cross_validation.ShuffleSplit are recommended instead.
- Direct support for the sequence of sequences (or list of lists) multilabel format is deprecated. To convert to and from the supported binary indicator matrix format, use *MultiLabelBinarizer*. By Joel Nothman.
- Add score method to *PCA* following the model of probabilistic PCA and deprecate ProbabilisticPCA model whose score implementation is not correct. The computation now also exploits the matrix inversion lemma for faster computation. By Alexandre Gramfort.
- The score method of *FactorAnalysis* now returns the average log-likelihood of the samples. Use score_samples to get log-likelihood of each sample. By Alexandre Gramfort.
- Generating boolean masks (the setting indices=False) from cross-validation generators is deprecated. Support for masks will be removed in 0.17. The generators have produced arrays of indices by default since 0.10. By Joel Nothman.
- 1-d arrays containing strings with dtype=object (as used in Pandas) are now considered valid classification targets. This fixes a regression from version 0.13 in some classifiers. By Joel Nothman.
- Fix wrong explained_variance_ratio_ attribute in RandomizedPCA. By Alexandre Gramfort.
- Fit alphas for each l1_ratio instead of mean_l1_ratio in *linear_model.ElasticNetCV* and *linear_model.LassoCV*. This changes the shape of alphas_ from (n_alphas,) to (n_l1_ratio, n_alphas) if the l1_ratio provided is a 1-D array like object of length greater than one. By Manoj Kumar.
- Fix *linear_model.ElasticNetCV* and *linear_model.LassoCV* when fitting intercept and input data is sparse. The automatic grid of alphas was not computed correctly and the scaling with normalize was wrong. By Manoj Kumar.
- Fix wrong maximal number of features drawn (max_features) at each split for decision trees, random forests and gradient tree boosting. Previously, the count for the number of drawn features started only after one non constant features in the split. This bug fix will affect computational and generalization performance of those algorithms in the presence of constant features. To get back previous generalization performance, you should modify the value of max_features. By Arnaud Joly.
- Fix wrong maximal number of features drawn (max_features) at each split for *ensemble.ExtraTreesClassifier* and *ensemble.ExtraTreesRegressor*. Previously, only non constant features in the split was counted as drawn. Now constant features are counted as drawn. Furthermore at least one feature must be non constant in order to make a valid split. This bug fix will affect computational and generalization performance of extra trees in the presence of constant features. To get back previous generalization performance, you should modify the value of max_features. By Arnaud Joly.
- Fix utils.compute_class_weight when class_weight=="auto". Previously it was broken for input of non-integer dtype and the weighted array that was returned was wrong. By Manoj Kumar.
- Fix cross_validation.Bootstrap to return ValueError when n_train + n_test > n. By Ronald Phlypo.

People

List of contributors for release 0.15 by number of commits.

- 312 Olivier Grisel
- 275 Lars Buitinck
- 221 Gael Varoquaux
- 148 Arnaud Joly
- 134 Johannes Schönberger
- 119 Gilles Louppe
- 113 Joel Nothman
- 111 Alexandre Gramfort
- 95 Jaques Grobler
- 89 Denis Engemann
- 83 Peter Prettenhofer
- 83 Alexander Fabisch
- 62 Mathieu Blondel
- 60 Eustache Diemert
- 60 Nelle Varoquaux
- 49 Michael Bommarito
- 45 Manoj-Kumar-S
- 28 Kyle Kastner
- 26 Andreas Mueller
- 22 Noel Dawe
- 21 Maheshakya Wijewardena
- 21 Brooke Osborn
- 21 Hamzeh Alsalhi
- 21 Jake VanderPlas
- 21 Philippe Gervais
- 19 Bala Subrahmanyam Varanasi
- 12 Ronald Phlypo
- 10 Mikhail Korobov
- 8 Thomas Unterthiner
- 8 Jeffrey Blackburne
- 8 eltermann
- 8 bwignall
- 7 Ankit Agrawal
- 7 CJ Carey

- 6 Daniel Nouri
- 6 Chen Liu
- 6 Michael Eickenberg
- 6 ugurthemaster
- 5 Aaron Schumacher
- 5 Baptiste Lagarde
- 5 Rajat Khanduja
- 5 Robert McGibbon
- 5 Sergio Pascual
- 4 Alexis Metaireau
- 4 Ignacio Rossi
- 4 Virgile Fritsch
- 4 Sebastian Säger
- 4 Ilambharathi Kanniah
- 4 sdenton4
- 4 Robert Layton
- 4 Alyssa
- 4 Amos Waterland
- 3 Andrew Tulloch
- 3 murad
- 3 Steven Maude
- 3 Karol Pysniak
- 3 Jacques Kvam
- 3 cgohlke
- 3 cjlin
- 3 Michael Becker
- 3 hamzeh
- 3 Eric Jacobsen
- 3 john collins
- 3 kaushik94
- 3 Erwin Marsi
- 2 csytracy
- 2 LK
- 2 Vlad Niculae
- 2 Laurent Direr
- 2 Erik Shilts

- 2 Raul Garreta
- 2 Yoshiki Vázquez Baeza
- 2 Yung Siang Liau
- 2 abhishek thakur
- 2 James Yu
- 2 Rohit Sivaprasad
- 2 Roland Szabo
- 2 amormachine
- 2 Alexis Mignon
- 2 Oscar Carlsson
- 2 Nantas Nardelli
- 2 jess010
- 2 kowalski87
- 2 Andrew Clegg
- 2 Federico Vaggi
- 2 Simon Frid
- 2 Félix-Antoine Fortin
- 1 Ralf Gommers
- 1 t-aft
- 1 Ronan Amicel
- 1 Rupesh Kumar Srivastava
- 1 Ryan Wang
- 1 Samuel Charron
- 1 Samuel St-Jean
- 1 Fabian Pedregosa
- 1 Skipper Seabold
- 1 Stefan Walk
- 1 Stefan van der Walt
- 1 Stephan Hoyer
- 1 Allen Riddell
- 1 Valentin Haenel
- 1 Vijay Ramesh
- 1 Will Myers
- 1 Yaroslav Halchenko
- 1 Yoni Ben-Meshulam
- 1 Yury V. Zaytsev

- 1 adrinjalali
- 1 ai8rahim
- 1 alemagnani
- 1 alex
- 1 benjamin wilson
- 1 chalmerlowe
- 1 dzikie drożdże
- 1 jamestwebber
- 1 matrixorz
- 1 popo
- 1 samuela
- 1 François Boulogne
- 1 Alexander Measure
- 1 Ethan White
- 1 Guilherme Trein
- 1 Hendrik Heuer
- 1 IvicaJovic
- 1 Jan Hendrik Metzen
- 1 Jean Michel Rouly
- 1 Eduardo Ariño de la Rubia
- 1 Jelle Zijlstra
- 1 Eddy L O Jansson
- 1 Denis
- 1 John
- 1 John Schmidt
- 1 Jorge Cañardo Alastuey
- 1 Joseph Perla
- 1 Joshua Vredevoogd
- 1 José Ricardo
- 1 Julien Miotte
- 1 Kemal Eren
- 1 Kenta Sato
- 1 David Cournapeau
- 1 Kyle Kelley
- 1 Daniele Medri
- 1 Laurent Luce

- 1 Laurent Pierron
- 1 Luis Pedro Coelho
- 1 DanielWeitzenfeld
- 1 Craig Thompson
- 1 Chyi-Kwei Yau
- 1 Matthew Brett
- 1 Matthias Feurer
- 1 Max Linke
- 1 Chris Filo Gorgolewski
- 1 Charles Earl
- 1 Michael Hanke
- 1 Michele Orrù
- 1 Bryan Lunt
- 1 Brian Kearns
- 1 Paul Butler
- 1 Paweł Mandera
- 1 Peter
- 1 Andrew Ash
- 1 Pietro Zambelli
- 1 staubda

1.17.14 Version 0.14

August 7, 2013

Changelog

- Missing values with sparse and dense matrices can be imputed with the transformer `preprocessing.Imputer` by Nicolas Trésegnie.
- The core implementation of decisions trees has been rewritten from scratch, allowing for faster tree induction and lower memory consumption in all tree-based estimators. By Gilles Louppe.
- Added `ensemble.AdaBoostClassifier` and `ensemble.AdaBoostRegressor`, by Noel Dawe and Gilles Louppe. See the *AdaBoost* section of the user guide for details and examples.
- Added `grid_search.RandomizedSearchCV` and `grid_search.ParameterSampler` for randomized hyperparameter optimization. By Andreas Müller.
- Added *biclustering* algorithms (`sklearn.cluster.bicluster.SpectralCoclustering` and `sklearn.cluster.bicluster.SpectralBiclustering`), data generation methods (`sklearn.datasets.make_biclusters` and `sklearn.datasets.make_checkerboard`), and scoring metrics (`sklearn.metrics.consensus_score`). By Kemal Eren.
- Added *Restricted Boltzmann Machines* (`neural_network.BernoulliRBM`). By Yann Dauphin.

- Python 3 support by Justin Vincent, Lars Buitinck, Subhodeep Moitra and Olivier Grisel. All tests now pass under Python 3.3.
- Ability to pass one penalty (alpha value) per target in *linear_model.Ridge*, by @eickenberg and Mathieu Blondel.
- Fixed sklearn.linear_model.stochastic_gradient.py L2 regularization issue (minor practical significance). By Norbert Crombach and Mathieu Blondel .
- Added an interactive version of Andreas Müller's Machine Learning Cheat Sheet (for scikit-learn) to the documentation. See *Choosing the right estimator*. By Jaques Grobler.
- grid_search.GridSearchCV and cross_validation.cross_val_score now support the use of advanced scoring function such as area under the ROC curve and f-beta scores. See *The scoring parameter: defining model evaluation rules* for details. By Andreas Müller and Lars Buitinck. Passing a function from *sklearn.metrics* as score_func is deprecated.
- Multi-label classification output is now supported by *metrics.accuracy_score*, *metrics.zero_one_loss*, *metrics.f1_score*, *metrics.fbeta_score*, *metrics.classification_report*, *metrics.precision_score* and *metrics.recall_score* by Arnaud Joly.
- Two new metrics *metrics.hamming_loss* and *metrics.jaccard_similarity_score* are added with multi-label support by Arnaud Joly.
- Speed and memory usage improvements in *feature_extraction.text.CountVectorizer* and *feature_extraction.text.TfidfVectorizer*, by Jochen Wersdörfer and Roman Sinayev.
- The min_df parameter in *feature_extraction.text.CountVectorizer* and *feature_extraction.text.TfidfVectorizer*, which used to be 2, has been reset to 1 to avoid unpleasant surprises (empty vocabularies) for novice users who try it out on tiny document collections. A value of at least 2 is still recommended for practical use.
- *svm.LinearSVC*, *linear_model.SGDClassifier* and *linear_model.SGDRegressor* now have a sparsify method that converts their coef_ into a sparse matrix, meaning stored models trained using these estimators can be made much more compact.
- *linear_model.SGDClassifier* now produces multiclass probability estimates when trained under log loss or modified Huber loss.
- Hyperlinks to documentation in example code on the website by Martin Luessi.
- Fixed bug in *preprocessing.MinMaxScaler* causing incorrect scaling of the features for non-default feature_range settings. By Andreas Müller.
- max_features in *tree.DecisionTreeClassifier*, *tree.DecisionTreeRegressor* and all derived ensemble estimators now supports percentage values. By Gilles Louppe.
- Performance improvements in *isotonic.IsotonicRegression* by Nelle Varoquaux.
- *metrics.accuracy_score* has an option normalize to return the fraction or the number of correctly classified sample by Arnaud Joly.
- Added *metrics.log_loss* that computes log loss, aka cross-entropy loss. By Jochen Wersdörfer and Lars Buitinck.
- A bug that caused *ensemble.AdaBoostClassifier*'s to output incorrect probabilities has been fixed.
- Feature selectors now share a mixin providing consistent transform, inverse_transform and get_support methods. By Joel Nothman.
- A fitted grid_search.GridSearchCV or grid_search.RandomizedSearchCV can now generally be pickled. By Joel Nothman.

- Refactored and vectorized implementation of *metrics.roc_curve* and *metrics.precision_recall_curve*. By Joel Nothman.
- The new estimator *sklearn.decomposition.TruncatedSVD* performs dimensionality reduction using SVD on sparse matrices, and can be used for latent semantic analysis (LSA). By Lars Buitinck.
- Added self-contained example of out-of-core learning on text data *Out-of-core classification of text documents*. By Eustache Diemert.
- The default number of components for sklearn.decomposition.RandomizedPCA is now correctly documented to be n_features. This was the default behavior, so programs using it will continue to work as they did.
- *sklearn.cluster.KMeans* now fits several orders of magnitude faster on sparse data (the speedup depends on the sparsity). By Lars Buitinck.
- Reduce memory footprint of FastICA by Denis Engemann and Alexandre Gramfort.
- Verbose output in sklearn.ensemble.gradient_boosting now uses a column format and prints progress in decreasing frequency. It also shows the remaining time. By Peter Prettenhofer.
- sklearn.ensemble.gradient_boosting provides out-of-bag improvement oob_improvement_ rather than the OOB score for model selection. An example that shows how to use OOB estimates to select the number of trees was added. By Peter Prettenhofer.
- Most metrics now support string labels for multiclass classification by Arnaud Joly and Lars Buitinck.
- New OrthogonalMatchingPursuitCV class by Alexandre Gramfort and Vlad Niculae.
- Fixed a bug in *sklearn.covariance.GraphLassoCV*: the 'alphas' parameter now works as expected when given a list of values. By Philippe Gervais.
- Fixed an important bug in *sklearn.covariance.GraphLassoCV* that prevented all folds provided by a CV object to be used (only the first 3 were used). When providing a CV object, execution time may thus increase significantly compared to the previous version (bug results are correct now). By Philippe Gervais.
- cross_validation.cross_val_score and the grid_search module is now tested with multi-output data by Arnaud Joly.
- *datasets.make_multilabel_classification* can now return the output in label indicator multilabel format by Arnaud Joly.
- K-nearest neighbors, *neighbors.KNeighborsRegressor* and *neighbors.RadiusNeighborsRegressor*, and radius neighbors, *neighbors.RadiusNeighborsRegressor* and *neighbors.RadiusNeighborsClassifier* support multioutput data by Arnaud Joly.
- Random state in LibSVM-based estimators (*svm.SVC*, NuSVC, OneClassSVM, *svm.SVR*, *svm.NuSVR*) can now be controlled. This is useful to ensure consistency in the probability estimates for the classifiers trained with probability=True. By Vlad Niculae.
- Out-of-core learning support for discrete naive Bayes classifiers *sklearn.naive_bayes.MultinomialNB* and *sklearn.naive_bayes.BernoulliNB* by adding the partial_fit method by Olivier Grisel.
- New website design and navigation by Gilles Louppe, Nelle Varoquaux, Vincent Michel and Andreas Müller.
- Improved documentation on *multi-class, multi-label and multi-output classification* by Yannick Schwartz and Arnaud Joly.
- Better input and error handling in the metrics module by Arnaud Joly and Joel Nothman.
- Speed optimization of the hmm module by Mikhail Korobov
- Significant speed improvements for *sklearn.cluster.DBSCAN* by cleverless

API changes summary

- The auc_score was renamed roc_auc_score.
- Testing scikit-learn with sklearn.test() is deprecated. Use nosetests sklearn from the command line.
- Feature importances in *tree.DecisionTreeClassifier*, *tree.DecisionTreeRegressor* and all derived ensemble estimators are now computed on the fly when accessing the feature_importances_ attribute. Setting compute_importances=True is no longer required. By Gilles Louppe.
- *linear_model.lasso_path* and *linear_model.enet_path* can return its results in the same format as that of *linear_model.lars_path*. This is done by setting the return_models parameter to False. By Jaques Grobler and Alexandre Gramfort
- grid_search.IterGrid was renamed to grid_search.ParameterGrid.
- Fixed bug in KFold causing imperfect class balance in some cases. By Alexandre Gramfort and Tadej Janež.
- *sklearn.neighbors.BallTree* has been refactored, and a *sklearn.neighbors.KDTree* has been added which shares the same interface. The Ball Tree now works with a wide variety of distance metrics. Both classes have many new methods, including single-tree and dual-tree queries, breadth-first and depth-first searching, and more advanced queries such as kernel density estimation and 2-point correlation functions. By Jake Vanderplas
- Support for scipy.spatial.cKDTree within neighbors queries has been removed, and the functionality replaced with the new KDTree class.
- *sklearn.neighbors.KernelDensity* has been added, which performs efficient kernel density estimation with a variety of kernels.
- *sklearn.decomposition.KernelPCA* now always returns output with n_components components, unless the new parameter remove_zero_eig is set to True. This new behavior is consistent with the way kernel PCA was always documented; previously, the removal of components with zero eigenvalues was tacitly performed on all data.
- gcv_mode="auto" no longer tries to perform SVD on a densified sparse matrix in *sklearn.linear_model.RidgeCV*.
- Sparse matrix support in sklearn.decomposition.RandomizedPCA is now deprecated in favor of the new TruncatedSVD.
- cross_validation.KFold and cross_validation.StratifiedKFold now enforce n_folds >= 2 otherwise a ValueError is raised. By Olivier Grisel.
- *datasets.load_files*'s charset and charset_errors parameters were renamed encoding and decode_errors.
- Attribute oob_score_ in *sklearn.ensemble.GradientBoostingRegressor* and *sklearn.ensemble.GradientBoostingClassifier* is deprecated and has been replaced by oob_improvement_.
- Attributes in OrthogonalMatchingPursuit have been deprecated (copy_X, Gram, …) and precompute_gram renamed precompute for consistency. See #2224.
- *sklearn.preprocessing.StandardScaler* now converts integer input to float, and raises a warning. Previously it rounded for dense integer input.
- *sklearn.multiclass.OneVsRestClassifier* now has a decision_function method. This will return the distance of each sample from the decision boundary for each class, as long as the underlying estimators implement the decision_function method. By Kyle Kastner.
- Better input validation, warning on unexpected shapes for y.

People

List of contributors for release 0.14 by number of commits.

- 277 Gilles Louppe
- 245 Lars Buitinck
- 187 Andreas Mueller
- 124 Arnaud Joly
- 112 Jaques Grobler
- 109 Gael Varoquaux
- 107 Olivier Grisel
- 102 Noel Dawe
- 99 Kemal Eren
- 79 Joel Nothman
- 75 Jake VanderPlas
- 73 Nelle Varoquaux
- 71 Vlad Niculae
- 65 Peter Prettenhofer
- 64 Alexandre Gramfort
- 54 Mathieu Blondel
- 38 Nicolas Trésegnie
- 35 eustache
- 27 Denis Engemann
- 25 Yann N. Dauphin
- 19 Justin Vincent
- 17 Robert Layton
- 15 Doug Coleman
- 14 Michael Eickenberg
- 13 Robert Marchman
- 11 Fabian Pedregosa
- 11 Philippe Gervais
- 10 Jim Holmström
- 10 Tadej Janež
- 10 syhw
- 9 Mikhail Korobov
- 9 Steven De Gryze
- 8 sergeyf
- 7 Ben Root

- 7 Hrishikesh Huilgolkar
- 6 Kyle Kastner
- 6 Martin Luessi
- 6 Rob Speer
- 5 Federico Vaggi
- 5 Raul Garreta
- 5 Rob Zinkov
- 4 Ken Geis
- 3 A. Flaxman
- 3 Denton Cockburn
- 3 Dougal Sutherland
- 3 Ian Ozsvald
- 3 Johannes Schönberger
- 3 Robert McGibbon
- 3 Roman Sinayev
- 3 Szabo Roland
- 2 Diego Molla
- 2 Imran Haque
- 2 Jochen Wersdörfer
- 2 Sergey Karayev
- 2 Yannick Schwartz
- 2 jamestwebber
- 1 Abhijeet Kolhe
- 1 Alexander Fabisch
- 1 Bastiaan van den Berg
- 1 Benjamin Peterson
- 1 Daniel Velkov
- 1 Fazlul Shahriar
- 1 Felix Brockherde
- 1 Félix-Antoine Fortin
- 1 Harikrishnan S
- 1 Jack Hale
- 1 JakeMick
- 1 James McDermott
- 1 John Benediktsson
- 1 John Zwinck

- 1 Joshua Vredevoogd
- 1 Justin Pati
- 1 Kevin Hughes
- 1 Kyle Kelley
- 1 Matthias Ekman
- 1 Miroslav Shubernetskiy
- 1 Naoki Orii
- 1 Norbert Crombach
- 1 Rafael Cunha de Almeida
- 1 Rolando Espinoza La fuente
- 1 Seamus Abshere
- 1 Sergey Feldman
- 1 Sergio Medina
- 1 Stefano Lattarini
- 1 Steve Koch
- 1 Sturla Molden
- 1 Thomas Jarosch
- 1 Yaroslav Halchenko

1.17.15 Version 0.13.1

February 23, 2013

The 0.13.1 release only fixes some bugs and does not add any new functionality.

Changelog

- Fixed a testing error caused by the function `cross_validation.train_test_split` being interpreted as a test by Yaroslav Halchenko.
- Fixed a bug in the reassignment of small clusters in the *`cluster.MiniBatchKMeans`* by Gael Varoquaux.
- Fixed default value of `gamma` in *`decomposition.KernelPCA`* by Lars Buitinck.
- Updated joblib to `0.7.0d` by Gael Varoquaux.
- Fixed scaling of the deviance in *`ensemble.GradientBoostingClassifier`* by Peter Prettenhofer.
- Better tie-breaking in *`multiclass.OneVsOneClassifier`* by Andreas Müller.
- Other small improvements to tests and documentation.

People

List of contributors for release 0.13.1 by number of commits.

- 16 Lars Buitinck
- 12 Andreas Müller
- 8 Gael Varoquaux
- 5 Robert Marchman
- 3 Peter Prettenhofer
- 2 Hrishikesh Huilgolkar
- 1 Bastiaan van den Berg
- 1 Diego Molla
- 1 Gilles Louppe
- 1 Mathieu Blondel
- 1 Nelle Varoquaux
- 1 Rafael Cunha de Almeida
- 1 Rolando Espinoza La fuente
- 1 Vlad Niculae
- 1 Yaroslav Halchenko

1.17.16 Version 0.13

January 21, 2013

New Estimator Classes

- *dummy.DummyClassifier* and *dummy.DummyRegressor*, two data-independent predictors by Mathieu Blondel. Useful to sanity-check your estimators. See *Dummy estimators* in the user guide. Multioutput support added by Arnaud Joly.
- *decomposition.FactorAnalysis*, a transformer implementing the classical factor analysis, by Christian Osendorfer and Alexandre Gramfort. See *Factor Analysis* in the user guide.
- *feature_extraction.FeatureHasher*, a transformer implementing the "hashing trick" for fast, low-memory feature extraction from string fields by Lars Buitinck and *feature_extraction.text.HashingVectorizer* for text documents by Olivier Grisel See *Feature hashing* and *Vectorizing a large text corpus with the hashing trick* for the documentation and sample usage.
- *pipeline.FeatureUnion*, a transformer that concatenates results of several other transformers by Andreas Müller. See *FeatureUnion: composite feature spaces* in the user guide.
- *random_projection.GaussianRandomProjection*, *random_projection.SparseRandomProjection* and the function *random_projection.johnson_lindenstrauss_min_dim*. The first two are transformers implementing Gaussian and sparse random projection matrix by Olivier Grisel and Arnaud Joly. See *Random Projection* in the user guide.
- *kernel_approximation.Nystroem*, a transformer for approximating arbitrary kernels by Andreas Müller. See *Nystroem Method for Kernel Approximation* in the user guide.

- *preprocessing.OneHotEncoder*, a transformer that computes binary encodings of categorical features by Andreas Müller. See *Encoding categorical features* in the user guide.
- *linear_model.PassiveAggressiveClassifier* and *linear_model.PassiveAggressiveRegressor*, predictors implementing an efficient stochastic optimization for linear models by Rob Zinkov and Mathieu Blondel. See *Passive Aggressive Algorithms* in the user guide.
- *ensemble.RandomTreesEmbedding*, a transformer for creating high-dimensional sparse representations using ensembles of totally random trees by Andreas Müller. See *Totally Random Trees Embedding* in the user guide.
- *manifold.SpectralEmbedding* and function *manifold.spectral_embedding*, implementing the "laplacian eigenmaps" transformation for non-linear dimensionality reduction by Wei Li. See *Spectral Embedding* in the user guide.
- *isotonic.IsotonicRegression* by Fabian Pedregosa, Alexandre Gramfort and Nelle Varoquaux,

Changelog

- *metrics.zero_one_loss* (formerly `metrics.zero_one`) now has option for normalized output that reports the fraction of misclassifications, rather than the raw number of misclassifications. By Kyle Beauchamp.
- *tree.DecisionTreeClassifier* and all derived ensemble models now support sample weighting, by Noel Dawe and Gilles Louppe.
- Speedup improvement when using bootstrap samples in forests of randomized trees, by Peter Prettenhofer and Gilles Louppe.
- Partial dependence plots for *Gradient Tree Boosting* in *ensemble.partial_dependence.partial_dependence* by Peter Prettenhofer. See sphx_glr_auto_examples_ensemble_plot_partial_dependence.py for an example.
- The table of contents on the website has now been made expandable by Jaques Grobler.
- *feature_selection.SelectPercentile* now breaks ties deterministically instead of returning all equally ranked features.
- *feature_selection.SelectKBest* and *feature_selection.SelectPercentile* are more numerically stable since they use scores, rather than p-values, to rank results. This means that they might sometimes select different features than they did previously.
- Ridge regression and ridge classification fitting with `sparse_cg` solver no longer has quadratic memory complexity, by Lars Buitinck and Fabian Pedregosa.
- Ridge regression and ridge classification now support a new fast solver called `lsqr`, by Mathieu Blondel.
- Speed up of *metrics.precision_recall_curve* by Conrad Lee.
- Added support for reading/writing svmlight files with pairwise preference attribute (qid in svmlight file format) in *datasets.dump_svmlight_file* and *datasets.load_svmlight_file* by Fabian Pedregosa.
- Faster and more robust *metrics.confusion_matrix* and *Clustering performance evaluation* by Wei Li.
- `cross_validation.cross_val_score` now works with precomputed kernels and affinity matrices, by Andreas Müller.
- LARS algorithm made more numerically stable with heuristics to drop regressors too correlated as well as to stop the path when numerical noise becomes predominant, by Gael Varoquaux.
- Faster implementation of *metrics.precision_recall_curve* by Conrad Lee.
- New kernel `metrics.chi2_kernel` by Andreas Müller, often used in computer vision applications.

- Fix of longstanding bug in *naive_bayes.BernoulliNB* fixed by Shaun Jackman.
- Implemented predict_proba in *multiclass.OneVsRestClassifier*, by Andrew Winterman.
- Improve consistency in gradient boosting: estimators *ensemble.GradientBoostingRegressor* and *ensemble.GradientBoostingClassifier* use the estimator *tree.DecisionTreeRegressor* instead of the tree._tree.Tree data structure by Arnaud Joly.
- Fixed a floating point exception in the *decision trees* module, by Seberg.
- Fix *metrics.roc_curve* fails when y_true has only one class by Wei Li.
- Add the *metrics.mean_absolute_error* function which computes the mean absolute error. The *metrics.mean_squared_error*, *metrics.mean_absolute_error* and *metrics.r2_score* metrics support multioutput by Arnaud Joly.
- Fixed class_weight support in *svm.LinearSVC* and *linear_model.LogisticRegression* by Andreas Müller. The meaning of class_weight was reversed as erroneously higher weight meant less positives of a given class in earlier releases.
- Improve narrative documentation and consistency in *sklearn.metrics* for regression and classification metrics by Arnaud Joly.
- Fixed a bug in *sklearn.svm.SVC* when using csr-matrices with unsorted indices by Xinfan Meng and Andreas Müller.
- MiniBatchKMeans: Add random reassignment of cluster centers with little observations attached to them, by Gael Varoquaux.

API changes summary

- Renamed all occurrences of n_atoms to n_components for consistency. This applies to *decomposition.DictionaryLearning*, *decomposition.MiniBatchDictionaryLearning*, *decomposition.dict_learning*, *decomposition.dict_learning_online*.
- Renamed all occurrences of max_iters to max_iter for consistency. This applies to *semi_supervised.LabelPropagation* and semi_supervised.label_propagation.LabelSpreading.
- Renamed all occurrences of learn_rate to learning_rate for consistency in ensemble.BaseGradientBoosting and *ensemble.GradientBoostingRegressor*.
- The module sklearn.linear_model.sparse is gone. Sparse matrix support was already integrated into the "regular" linear models.
- sklearn.metrics.mean_square_error, which incorrectly returned the accumulated error, was removed. Use mean_squared_error instead.
- Passing class_weight parameters to fit methods is no longer supported. Pass them to estimator constructors instead.
- GMMs no longer have decode and rvs methods. Use the score, predict or sample methods instead.
- The solver fit option in Ridge regression and classification is now deprecated and will be removed in v0.14. Use the constructor option instead.
- feature_extraction.text.DictVectorizer now returns sparse matrices in the CSR format, instead of COO.
- Renamed k in cross_validation.KFold and cross_validation.StratifiedKFold to n_folds, renamed n_bootstraps to n_iter in cross_validation.Bootstrap.

- Renamed all occurrences of n_iterations to n_iter for consistency. This applies to cross_validation.ShuffleSplit, cross_validation.StratifiedShuffleSplit, utils.randomized_range_finder and utils.randomized_svd.
- Replaced rho in *linear_model.ElasticNet* and *linear_model.SGDClassifier* by l1_ratio. The rho parameter had different meanings; l1_ratio was introduced to avoid confusion. It has the same meaning as previously rho in *linear_model.ElasticNet* and (1-rho) in *linear_model.SGDClassifier*.
- *linear_model.LassoLars* and *linear_model.Lars* now store a list of paths in the case of multiple targets, rather than an array of paths.
- The attribute gmm of hmm.GMMHMM was renamed to gmm_ to adhere more strictly with the API.
- cluster.spectral_embedding was moved to *manifold.spectral_embedding*.
- Renamed eig_tol in *manifold.spectral_embedding*, *cluster.SpectralClustering* to eigen_tol, renamed mode to eigen_solver.
- Renamed mode in *manifold.spectral_embedding* and *cluster.SpectralClustering* to eigen_solver.
- classes_ and n_classes_ attributes of *tree.DecisionTreeClassifier* and all derived ensemble models are now flat in case of single output problems and nested in case of multi-output problems.
- The estimators_ attribute of ensemble.gradient_boosting.GradientBoostingRegressor and ensemble.gradient_boosting.GradientBoostingClassifier is now an array of :class:'tree.DecisionTreeRegressor'.
- Renamed chunk_size to batch_size in *decomposition.MiniBatchDictionaryLearning* and *decomposition.MiniBatchSparsePCA* for consistency.
- *svm.SVC* and *svm.NuSVC* now provide a classes_ attribute and support arbitrary dtypes for labels y. Also, the dtype returned by predict now reflects the dtype of y during fit (used to be np.float).
- Changed default test_size in cross_validation.train_test_split to None, added possibility to infer test_size from train_size in cross_validation.ShuffleSplit and cross_validation.StratifiedShuffleSplit.
- Renamed function sklearn.metrics.zero_one to *sklearn.metrics.zero_one_loss*. Be aware that the default behavior in *sklearn.metrics.zero_one_loss* is different from sklearn.metrics.zero_one: normalize=False is changed to normalize=True.
- Renamed function metrics.zero_one_score to *metrics.accuracy_score*.
- *datasets.make_circles* now has the same number of inner and outer points.
- In the Naive Bayes classifiers, the class_prior parameter was moved from fit to __init__.

People

List of contributors for release 0.13 by number of commits.

- 364 Andreas Müller
- 143 Arnaud Joly
- 137 Peter Prettenhofer
- 131 Gael Varoquaux
- 117 Mathieu Blondel
- 108 Lars Buitinck

- 106 Wei Li
- 101 Olivier Grisel
- 65 Vlad Niculae
- 54 Gilles Louppe
- 40 Jaques Grobler
- 38 Alexandre Gramfort
- 30 Rob Zinkov
- 19 Aymeric Masurelle
- 18 Andrew Winterman
- 17 Fabian Pedregosa
- 17 Nelle Varoquaux
- 16 Christian Osendorfer
- 14 Daniel Nouri
- 13 Virgile Fritsch
- 13 syhw
- 12 Satrajit Ghosh
- 10 Corey Lynch
- 10 Kyle Beauchamp
- 9 Brian Cheung
- 9 Immanuel Bayer
- 9 mr.Shu
- 8 Conrad Lee
- 8 James Bergstra
- 7 Tadej Janež
- 6 Brian Cajes
- 6 Jake Vanderplas
- 6 Michael
- 6 Noel Dawe
- 6 Tiago Nunes
- 6 cow
- 5 Anze
- 5 Shiqiao Du
- 4 Christian Jauvin
- 4 Jacques Kvam
- 4 Richard T. Guy
- 4 Robert Layton

- 3 Alexandre Abraham
- 3 Doug Coleman
- 3 Scott Dickerson
- 2 ApproximateIdentity
- 2 John Benediktsson
- 2 Mark Veronda
- 2 Matti Lyra
- 2 Mikhail Korobov
- 2 Xinfan Meng
- 1 Alejandro Weinstein
- 1 Alexandre Passos
- 1 Christoph Deil
- 1 Eugene Nizhibitsky
- 1 Kenneth C. Arnold
- 1 Luis Pedro Coelho
- 1 Miroslav Batchkarov
- 1 Pavel
- 1 Sebastian Berg
- 1 Shaun Jackman
- 1 Subhodeep Moitra
- 1 bob
- 1 dengemann
- 1 emanuele
- 1 x006

1.17.17 Version 0.12.1

October 8, 2012

The 0.12.1 release is a bug-fix release with no additional features, but is instead a set of bug fixes

Changelog

- Improved numerical stability in spectral embedding by Gael Varoquaux
- Doctest under windows 64bit by Gael Varoquaux
- Documentation fixes for elastic net by Andreas Müller and Alexandre Gramfort
- Proper behavior with fortran-ordered NumPy arrays by Gael Varoquaux
- Make GridSearchCV work with non-CSR sparse matrix by Lars Buitinck
- Fix parallel computing in MDS by Gael Varoquaux

- Fix Unicode support in count vectorizer by Andreas Müller
- Fix MinCovDet breaking with X.shape = (3, 1) by Virgile Fritsch
- Fix clone of SGD objects by Peter Prettenhofer
- Stabilize GMM by Virgile Fritsch

People

- 14 Peter Prettenhofer
- 12 Gael Varoquaux
- 10 Andreas Müller
- 5 Lars Buitinck
- 3 Virgile Fritsch
- 1 Alexandre Gramfort
- 1 Gilles Louppe
- 1 Mathieu Blondel

1.17.18 Version 0.12

September 4, 2012

Changelog

- Various speed improvements of the *decision trees* module, by Gilles Louppe.
- *ensemble.GradientBoostingRegressor* and *ensemble.GradientBoostingClassifier* now support feature subsampling via the max_features argument, by Peter Prettenhofer.
- Added Huber and Quantile loss functions to *ensemble.GradientBoostingRegressor*, by Peter Prettenhofer.
- *Decision trees* and *forests of randomized trees* now support multi-output classification and regression problems, by Gilles Louppe.
- Added *preprocessing.LabelEncoder*, a simple utility class to normalize labels or transform non-numerical labels, by Mathieu Blondel.
- Added the epsilon-insensitive loss and the ability to make probabilistic predictions with the modified huber loss in *Stochastic Gradient Descent*, by Mathieu Blondel.
- Added *Multi-dimensional Scaling (MDS)*, by Nelle Varoquaux.
- SVMlight file format loader now detects compressed (gzip/bzip2) files and decompresses them on the fly, by Lars Buitinck.
- SVMlight file format serializer now preserves double precision floating point values, by Olivier Grisel.
- A common testing framework for all estimators was added, by Andreas Müller.
- Understandable error messages for estimators that do not accept sparse input by Gael Varoquaux
- Speedups in hierarchical clustering by Gael Varoquaux. In particular building the tree now supports early stopping. This is useful when the number of clusters is not small compared to the number of samples.

- Add MultiTaskLasso and MultiTaskElasticNet for joint feature selection, by Alexandre Gramfort.
- Added `metrics.auc_score` and *`metrics.average_precision_score`* convenience functions by Andreas Müller.
- Improved sparse matrix support in the *Feature selection* module by Andreas Müller.
- New word boundaries-aware character n-gram analyzer for the *Text feature extraction* module by @kernc.
- Fixed bug in spectral clustering that led to single point clusters by Andreas Müller.
- In *`feature_extraction.text.CountVectorizer`*, added an option to ignore infrequent words, `min_df` by Andreas Müller.
- Add support for multiple targets in some linear models (ElasticNet, Lasso and OrthogonalMatchingPursuit) by Vlad Niculae and Alexandre Gramfort.
- Fixes in `decomposition.ProbabilisticPCA` score function by Wei Li.
- Fixed feature importance computation in *Gradient Tree Boosting*.

API changes summary

- The old `scikits.learn` package has disappeared; all code should import from `sklearn` instead, which was introduced in 0.9.
- In *`metrics.roc_curve`*, the `thresholds` array is now returned with it's order reversed, in order to keep it consistent with the order of the returned `fpr` and `tpr`.
- In `hmm` objects, like `hmm.GaussianHMM`, `hmm.MultinomialHMM`, etc., all parameters must be passed to the object when initialising it and not through `fit`. Now `fit` will only accept the data as an input parameter.
- For all SVM classes, a faulty behavior of `gamma` was fixed. Previously, the default gamma value was only computed the first time `fit` was called and then stored. It is now recalculated on every call to `fit`.
- All `Base` classes are now abstract meta classes so that they can not be instantiated.
- *`cluster.ward_tree`* now also returns the parent array. This is necessary for early-stopping in which case the tree is not completely built.
- In *`feature_extraction.text.CountVectorizer`* the parameters `min_n` and `max_n` were joined to the parameter `n_gram_range` to enable grid-searching both at once.
- In *`feature_extraction.text.CountVectorizer`*, words that appear only in one document are now ignored by default. To reproduce the previous behavior, set `min_df=1`.
- Fixed API inconsistency: *`linear_model.SGDClassifier.predict_proba`* now returns 2d array when fit on two classes.
- Fixed API inconsistency: *`discriminant_analysis.QuadraticDiscriminantAnalysis.decision_function`* and *`discriminant_analysis.LinearDiscriminantAnalysis.decision_function`* now return 1d arrays when fit on two classes.
- Grid of alphas used for fitting *`linear_model.LassoCV`* and *`linear_model.ElasticNetCV`* is now stored in the attribute `alphas_` rather than overriding the init parameter `alphas`.
- Linear models when alpha is estimated by cross-validation store the estimated value in the `alpha_` attribute rather than just `alpha` or `best_alpha`.
- *`ensemble.GradientBoostingClassifier`* now supports *`ensemble.GradientBoostingClassifier.staged_predict_proba`*, and *`ensemble.GradientBoostingClassifier.staged_predict`*.

- svm.sparse.SVC and other sparse SVM classes are now deprecated. The all classes in the *Support Vector Machines* module now automatically select the sparse or dense representation base on the input.
- All clustering algorithms now interpret the array X given to fit as input data, in particular *cluster.SpectralClustering* and *cluster.AffinityPropagation* which previously expected affinity matrices.
- For clustering algorithms that take the desired number of clusters as a parameter, this parameter is now called n_clusters.

People

- 267 Andreas Müller
- 94 Gilles Louppe
- 89 Gael Varoquaux
- 79 Peter Prettenhofer
- 60 Mathieu Blondel
- 57 Alexandre Gramfort
- 52 Vlad Niculae
- 45 Lars Buitinck
- 44 Nelle Varoquaux
- 37 Jaques Grobler
- 30 Alexis Mignon
- 30 Immanuel Bayer
- 27 Olivier Grisel
- 16 Subhodeep Moitra
- 13 Yannick Schwartz
- 12 @kernc
- 11 Virgile Fritsch
- 9 Daniel Duckworth
- 9 Fabian Pedregosa
- 9 Robert Layton
- 8 John Benediktsson
- 7 Marko Burjek
- 5 Nicolas Pinto
- 4 Alexandre Abraham
- 4 Jake Vanderplas
- 3 Brian Holt
- 3 Edouard Duchesnay
- 3 Florian Hoenig

- 3 flyingimmidev
- 2 Francois Savard
- 2 Hannes Schulz
- 2 Peter Welinder
- 2 Yaroslav Halchenko
- 2 Wei Li
- 1 Alex Companioni
- 1 Brandyn A. White
- 1 Bussonnier Matthias
- 1 Charles-Pierre Astolfi
- 1 Dan O'Huiginn
- 1 David Cournapeau
- 1 Keith Goodman
- 1 Ludwig Schwardt
- 1 Olivier Hervieu
- 1 Sergio Medina
- 1 Shiqiao Du
- 1 Tim Sheerman-Chase
- 1 buguen

1.17.19 Version 0.11

May 7, 2012

Changelog

Highlights

- Gradient boosted regression trees (*Gradient Tree Boosting*) for classification and regression by Peter Prettenhofer and Scott White .
- Simple dict-based feature loader with support for categorical variables (`feature_extraction.DictVectorizer`) by Lars Buitinck.
- Added Matthews correlation coefficient (`metrics.matthews_corrcoef`) and added macro and micro average options to `metrics.precision_score`, `metrics.recall_score` and `metrics.f1_score` by Satrajit Ghosh.
- *Out of Bag Estimates* of generalization error for *Ensemble methods* by Andreas Müller.
- Randomized sparse linear models for feature selection, by Alexandre Gramfort and Gael Varoquaux
- *Label Propagation* for semi-supervised learning, by Clay Woolam. **Note** the semi-supervised API is still work in progress, and may change.

- Added BIC/AIC model selection to classical *Gaussian mixture models* and unified the API with the remainder of scikit-learn, by Bertrand Thirion
- Added `sklearn.cross_validation.StratifiedShuffleSplit`, which is a `sklearn.cross_validation.ShuffleSplit` with balanced splits, by Yannick Schwartz.
- *`sklearn.neighbors.NearestCentroid`* classifier added, along with a `shrink_threshold` parameter, which implements **shrunken centroid classification**, by Robert Layton.

Other changes

- Merged dense and sparse implementations of *Stochastic Gradient Descent* module and exposed utility extension types for sequential datasets `seq_dataset` and weight vectors `weight_vector` by Peter Prettenhofer.
- Added `partial_fit` (support for online/minibatch learning) and warm_start to the *Stochastic Gradient Descent* module by Mathieu Blondel.
- Dense and sparse implementations of *Support Vector Machines* classes and *`linear_model.LogisticRegression`* merged by Lars Buitinck.
- Regressors can now be used as base estimator in the *Multiclass and multilabel algorithms* module by Mathieu Blondel.
- Added n_jobs option to `metrics.pairwise.pairwise_distances` and *`metrics.pairwise.pairwise_kernels`* for parallel computation, by Mathieu Blondel.
- *K-means* can now be run in parallel, using the `n_jobs` argument to either *K-means* or `KMeans`, by Robert Layton.
- Improved *Cross-validation: evaluating estimator performance* and *Tuning the hyper-parameters of an estimator* documentation and introduced the new `cross_validation.train_test_split` helper function by Olivier Grisel
- *`svm.SVC`* members `coef_` and `intercept_` changed sign for consistency with `decision_function`; for `kernel==linear`, `coef_` was fixed in the one-vs-one case, by Andreas Müller.
- Performance improvements to efficient leave-one-out cross-validated Ridge regression, esp. for the `n_samples > n_features` case, in *`linear_model.RidgeCV`*, by Reuben Fletcher-Costin.
- Refactoring and simplification of the *Text feature extraction* API and fixed a bug that caused possible negative IDF, by Olivier Grisel.
- Beam pruning option in `_BaseHMM` module has been removed since it is difficult to Cythonize. If you are interested in contributing a Cython version, you can use the python version in the git history as a reference.
- Classes in *Nearest Neighbors* now support arbitrary Minkowski metric for nearest neighbors searches. The metric can be specified by argument `p`.

API changes summary

- `covariance.EllipticEnvelop` is now deprecated - Please use *`covariance.EllipticEnvelope`* instead.
- `NeighborsClassifier` and `NeighborsRegressor` are gone in the module *Nearest Neighbors*. Use the classes `KNeighborsClassifier`, `RadiusNeighborsClassifier`, `KNeighborsRegressor` and/or `RadiusNeighborsRegressor` instead.
- Sparse classes in the *Stochastic Gradient Descent* module are now deprecated.

- In mixture.GMM, mixture.DPGMM and mixture.VBGMM, parameters must be passed to an object when initialising it and not through fit. Now fit will only accept the data as an input parameter.
- methods rvs and decode in GMM module are now deprecated. sample and score or predict should be used instead.
- attribute _scores and _pvalues in univariate feature selection objects are now deprecated. scores_ or pvalues_ should be used instead.
- In LogisticRegression, LinearSVC, SVC and NuSVC, the class_weight parameter is now an initialization parameter, not a parameter to fit. This makes grid searches over this parameter possible.
- LFW data is now always shape (n_samples, n_features) to be consistent with the Olivetti faces dataset. Use images and pairs attribute to access the natural images shapes instead.
- In *svm.LinearSVC*, the meaning of the multi_class parameter changed. Options now are 'ovr' and 'crammer_singer', with 'ovr' being the default. This does not change the default behavior but hopefully is less confusing.
- Class feature_selection.text.Vectorizer is deprecated and replaced by feature_selection.text.TfidfVectorizer.
- The preprocessor / analyzer nested structure for text feature extraction has been removed. All those features are now directly passed as flat constructor arguments to feature_selection.text.TfidfVectorizer and feature_selection.text.CountVectorizer, in particular the following parameters are now used:
- analyzer can be 'word' or 'char' to switch the default analysis scheme, or use a specific python callable (as previously).
- tokenizer and preprocessor have been introduced to make it still possible to customize those steps with the new API.
- input explicitly control how to interpret the sequence passed to fit and predict: filenames, file objects or direct (byte or Unicode) strings.
- charset decoding is explicit and strict by default.
- the vocabulary, fitted or not is now stored in the vocabulary_ attribute to be consistent with the project conventions.
- Class feature_selection.text.TfidfVectorizer now derives directly from feature_selection.text.CountVectorizer to make grid search trivial.
- methods rvs in _BaseHMM module are now deprecated. sample should be used instead.
- Beam pruning option in _BaseHMM module is removed since it is difficult to be Cythonized. If you are interested, you can look in the history codes by git.
- The SVMlight format loader now supports files with both zero-based and one-based column indices, since both occur "in the wild".
- Arguments in class ShuffleSplit are now consistent with StratifiedShuffleSplit. Arguments test_fraction and train_fraction are deprecated and renamed to test_size and train_size and can accept both float and int.
- Arguments in class Bootstrap are now consistent with StratifiedShuffleSplit. Arguments n_test and n_train are deprecated and renamed to test_size and train_size and can accept both float and int.
- Argument p added to classes in *Nearest Neighbors* to specify an arbitrary Minkowski metric for nearest neighbors searches.

People

- 282 Andreas Müller
- 239 Peter Prettenhofer
- 198 Gael Varoquaux
- 129 Olivier Grisel
- 114 Mathieu Blondel
- 103 Clay Woolam
- 96 Lars Buitinck
- 88 Jaques Grobler
- 82 Alexandre Gramfort
- 50 Bertrand Thirion
- 42 Robert Layton
- 28 flyingimmidev
- 26 Jake Vanderplas
- 26 Shiqiao Du
- 21 Satrajit Ghosh
- 17 David Marek
- 17 Gilles Louppe
- 14 Vlad Niculae
- 11 Yannick Schwartz
- 10 Fabian Pedregosa
- 9 fcostin
- 7 Nick Wilson
- 5 Adrien Gaidon
- 5 Nicolas Pinto
- 4 David Warde-Farley
- 5 Nelle Varoquaux
- 5 Emmanuelle Gouillart
- 3 Joonas Sillanpää
- 3 Paolo Losi
- 2 Charles McCarthy
- 2 Roy Hyunjin Han
- 2 Scott White
- 2 ibayer
- 1 Brandyn White
- 1 Carlos Scheidegger

- 1 Claire Revillet
- 1 Conrad Lee
- 1 Edouard Duchesnay
- 1 Jan Hendrik Metzen
- 1 Meng Xinfan
- 1 Rob Zinkov
- 1 Shiqiao
- 1 Udi Weinsberg
- 1 Virgile Fritsch
- 1 Xinfan Meng
- 1 Yaroslav Halchenko
- 1 jansoe
- 1 Leon Palafox

1.17.20 Version 0.10

January 11, 2012

Changelog

- Python 2.5 compatibility was dropped; the minimum Python version needed to use scikit-learn is now 2.6.
- *Sparse inverse covariance* estimation using the graph Lasso, with associated cross-validated estimator, by Gael Varoquaux
- New *Tree* module by Brian Holt, Peter Prettenhofer, Satrajit Ghosh and Gilles Louppe. The module comes with complete documentation and examples.
- Fixed a bug in the RFE module by Gilles Louppe (issue #378).
- Fixed a memory leak in *Support Vector Machines* module by Brian Holt (issue #367).
- Faster tests by Fabian Pedregosa and others.
- Silhouette Coefficient cluster analysis evaluation metric added as `sklearn.metrics.silhouette_score` by Robert Layton.
- Fixed a bug in *K-means* in the handling of the `n_init` parameter: the clustering algorithm used to be run `n_init` times but the last solution was retained instead of the best solution by Olivier Grisel.
- Minor refactoring in *Stochastic Gradient Descent* module; consolidated dense and sparse predict methods; Enhanced test time performance by converting model parameters to fortran-style arrays after fitting (only multi-class).
- Adjusted Mutual Information metric added as `sklearn.metrics.adjusted_mutual_info_score` by Robert Layton.
- Models like SVC/SVR/LinearSVC/LogisticRegression from libsvm/liblinear now support scaling of C regularization parameter by the number of samples by Alexandre Gramfort.
- New *Ensemble Methods* module by Gilles Louppe and Brian Holt. The module comes with the random forest algorithm and the extra-trees method, along with documentation and examples.

- *Novelty and Outlier Detection*: outlier and novelty detection, by Virgile Fritsch.
- *Kernel Approximation*: a transform implementing kernel approximation for fast SGD on non-linear kernels by Andreas Müller.
- Fixed a bug due to atom swapping in *Orthogonal Matching Pursuit (OMP)* by Vlad Niculae.
- *Sparse coding with a precomputed dictionary* by Vlad Niculae.
- *Mini Batch K-Means* performance improvements by Olivier Grisel.
- *K-means* support for sparse matrices by Mathieu Blondel.
- Improved documentation for developers and for the `sklearn.utils` module, by Jake Vanderplas.
- Vectorized 20newsgroups dataset loader (`sklearn.datasets.fetch_20newsgroups_vectorized`) by Mathieu Blondel.
- *Multiclass and multilabel algorithms* by Lars Buitinck.
- Utilities for fast computation of mean and variance for sparse matrices by Mathieu Blondel.
- Make `sklearn.preprocessing.scale` and `sklearn.preprocessing.Scaler` work on sparse matrices by Olivier Grisel
- Feature importances using decision trees and/or forest of trees, by Gilles Louppe.
- Parallel implementation of forests of randomized trees by Gilles Louppe.
- `sklearn.cross_validation.ShuffleSplit` can subsample the train sets as well as the test sets by Olivier Grisel.
- Errors in the build of the documentation fixed by Andreas Müller.

API changes summary

Here are the code migration instructions when upgrading from scikit-learn version 0.9:

- Some estimators that may overwrite their inputs to save memory previously had `overwrite_` parameters; these have been replaced with `copy_` parameters with exactly the opposite meaning.

  This particularly affects some of the estimators in `linear_model`. The default behavior is still to copy everything passed in.
- The SVMlight dataset loader `sklearn.datasets.load_svmlight_file` no longer supports loading two files at once; use `load_svmlight_files` instead. Also, the (unused) `buffer_mb` parameter is gone.
- Sparse estimators in the *Stochastic Gradient Descent* module use dense parameter vector `coef_` instead of `sparse_coef_`. This significantly improves test time performance.
- The *Covariance estimation* module now has a robust estimator of covariance, the Minimum Covariance Determinant estimator.
- Cluster evaluation metrics in `metrics.cluster` have been refactored but the changes are backwards compatible. They have been moved to the `metrics.cluster.supervised`, along with `metrics.cluster.unsupervised` which contains the Silhouette Coefficient.
- The `permutation_test_score` function now behaves the same way as `cross_val_score` (i.e. uses the mean score across the folds.)
- Cross Validation generators now use integer indices (`indices=True`) by default instead of boolean masks. This make it more intuitive to use with sparse matrix data.

- The functions used for sparse coding, `sparse_encode` and `sparse_encode_parallel` have been combined into `sklearn.decomposition.sparse_encode`, and the shapes of the arrays have been transposed for consistency with the matrix factorization setting, as opposed to the regression setting.
- Fixed an off-by-one error in the SVMlight/LibSVM file format handling; files generated using `sklearn.datasets.dump_svmlight_file` should be re-generated. (They should continue to work, but accidentally had one extra column of zeros prepended.)
- `BaseDictionaryLearning` class replaced by `SparseCodingMixin`.
- `sklearn.utils.extmath.fast_svd` has been renamed `sklearn.utils.extmath.randomized_svd` and the default oversampling is now fixed to 10 additional random vectors instead of doubling the number of components to extract. The new behavior follows the reference paper.

People

The following people contributed to scikit-learn since last release:

- 246 Andreas Müller
- 242 Olivier Grisel
- 220 Gilles Louppe
- 183 Brian Holt
- 166 Gael Varoquaux
- 144 Lars Buitinck
- 73 Vlad Niculae
- 65 Peter Prettenhofer
- 64 Fabian Pedregosa
- 60 Robert Layton
- 55 Mathieu Blondel
- 52 Jake Vanderplas
- 44 Noel Dawe
- 38 Alexandre Gramfort
- 24 Virgile Fritsch
- 23 Satrajit Ghosh
- 3 Jan Hendrik Metzen
- 3 Kenneth C. Arnold
- 3 Shiqiao Du
- 3 Tim Sheerman-Chase
- 3 Yaroslav Halchenko
- 2 Bala Subrahmanyam Varanasi
- 2 DraXus
- 2 Michael Eickenberg
- 1 Bogdan Trach

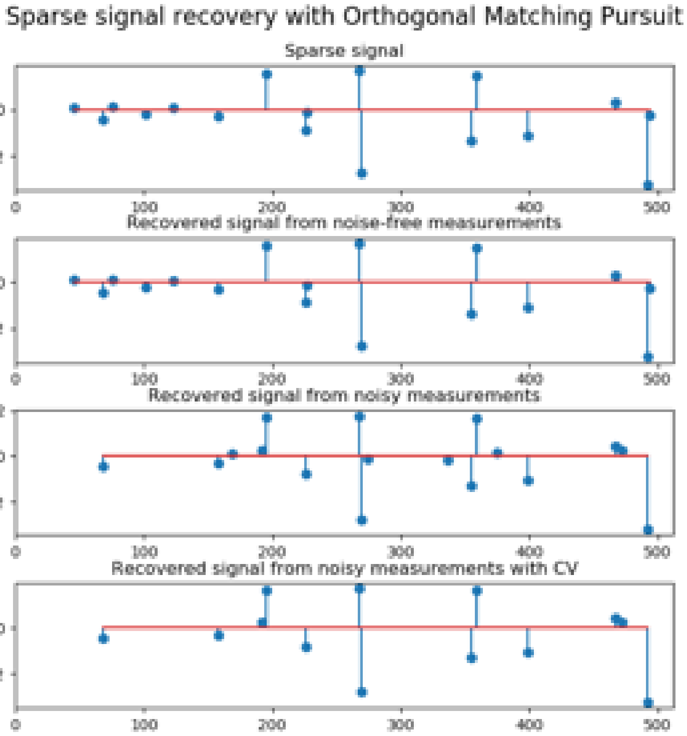
- 1 Félix-Antoine Fortin
- 1 Juan Manuel Caicedo Carvajal
- 1 Nelle Varoquaux
- 1 Nicolas Pinto
- 1 Tiziano Zito
- 1 Xinfan Meng

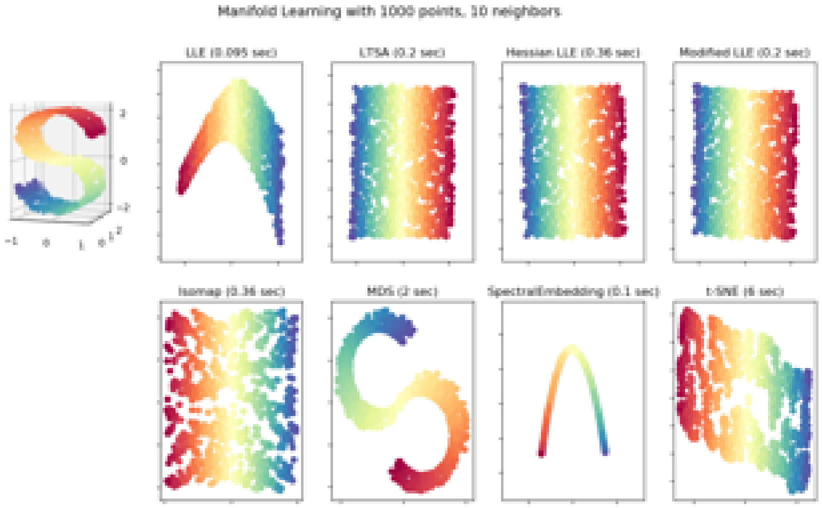
1.17.21 Version 0.9

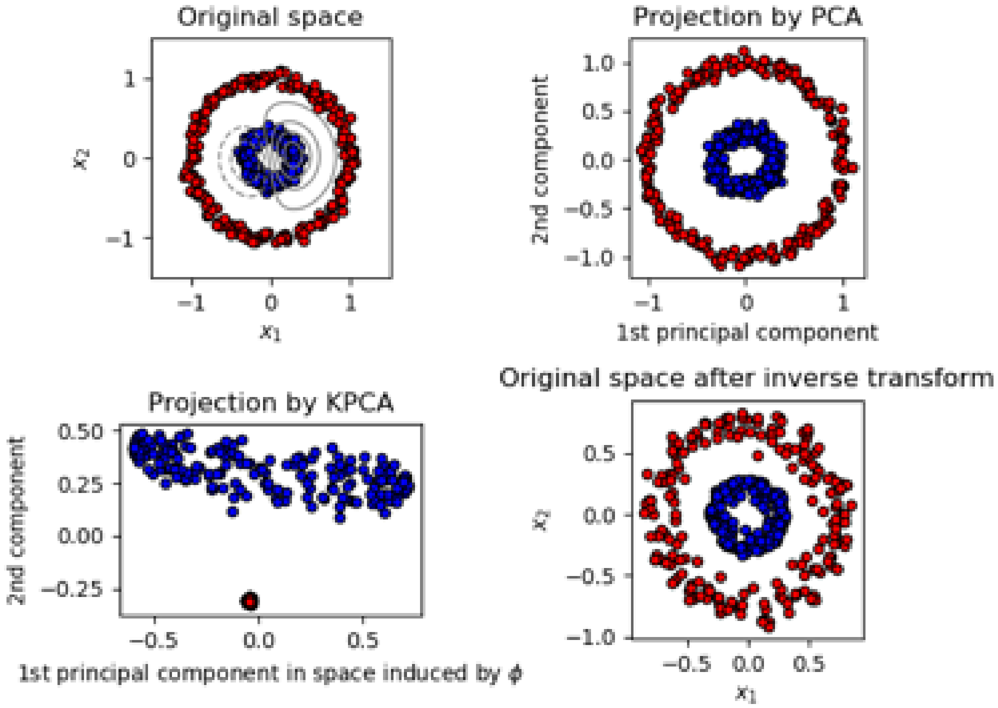
September 21, 2011

scikit-learn 0.9 was released on September 2011, three months after the 0.8 release and includes the new modules *Manifold learning*, *The Dirichlet Process* as well as several new algorithms and documentation improvements.

This release also includes the dictionary-learning work developed by Vlad Niculae as part of the Google Summer of Code program.







Changelog

- New *Manifold learning* module by Jake Vanderplas and Fabian Pedregosa.
- New *Dirichlet Process* Gaussian Mixture Model by Alexandre Passos

- *Nearest Neighbors* module refactoring by Jake Vanderplas : general refactoring, support for sparse matrices in input, speed and documentation improvements. See the next section for a full list of API changes.
- Improvements on the *Feature selection* module by Gilles Louppe : refactoring of the RFE classes, documentation rewrite, increased efficiency and minor API changes.
- *Sparse principal components analysis (SparsePCA and MiniBatchSparsePCA)* by Vlad Niculae, Gael Varoquaux and Alexandre Gramfort
- Printing an estimator now behaves independently of architectures and Python version thanks to Jean Kossaifi.
- *Loader for libsvm/svmlight format* by Mathieu Blondel and Lars Buitinck
- Documentation improvements: thumbnails in example gallery by Fabian Pedregosa.
- Important bugfixes in *Support Vector Machines* module (segfaults, bad performance) by Fabian Pedregosa.
- Added *Multinomial Naive Bayes* and *Bernoulli Naive Bayes* by Lars Buitinck
- Text feature extraction optimizations by Lars Buitinck
- Chi-Square feature selection (`feature_selection.univariate_selection.chi2`) by Lars Buitinck.
- *Generated datasets* module refactoring by Gilles Louppe
- *Multiclass and multilabel algorithms* by Mathieu Blondel
- Ball tree rewrite by Jake Vanderplas
- Implementation of *DBSCAN* algorithm by Robert Layton
- Kmeans predict and transform by Robert Layton
- Preprocessing module refactoring by Olivier Grisel
- Faster mean shift by Conrad Lee
- New `Bootstrap`, *Random permutations cross-validation a.k.a. Shuffle & Split* and various other improvements in cross validation schemes by Olivier Grisel and Gael Varoquaux
- Adjusted Rand index and V-Measure clustering evaluation metrics by Olivier Grisel
- Added *`Orthogonal Matching Pursuit`* by Vlad Niculae
- Added 2D-patch extractor utilities in the *Feature extraction* module by Vlad Niculae
- Implementation of *`linear_model.LassoLarsCV`* (cross-validated Lasso solver using the Lars algorithm) and *`linear_model.LassoLarsIC`* (BIC/AIC model selection in Lars) by Gael Varoquaux and Alexandre Gramfort
- Scalability improvements to *`metrics.roc_curve`* by Olivier Hervieu
- Distance helper functions `metrics.pairwise.pairwise_distances` and *`metrics.pairwise.pairwise_kernels`* by Robert Layton
- *`Mini-Batch K-Means`* by Nelle Varoquaux and Peter Prettenhofer.
- mldata utilities by Pietro Berkes.
- olivetti_faces by David Warde-Farley.

API changes summary

Here are the code migration instructions when upgrading from scikit-learn version 0.8:

- The `scikits.learn` package was renamed `sklearn`. There is still a `scikits.learn` package alias for backward compatibility.

  Third-party projects with a dependency on scikit-learn 0.9+ should upgrade their codebase. For instance, under Linux / MacOSX just run (make a backup first!):

  ```
  find -name "*.py" | xargs sed -i 's/\bscikits.learn\b/sklearn/g'
  ```

- Estimators no longer accept model parameters as `fit` arguments: instead all parameters must be only be passed as constructor arguments or using the now public `set_params` method inherited from *`base.BaseEstimator`*.

  Some estimators can still accept keyword arguments on the `fit` but this is restricted to data-dependent values (e.g. a Gram matrix or an affinity matrix that are precomputed from the `X` data matrix.

- The `cross_val` package has been renamed to `cross_validation` although there is also a `cross_val` package alias in place for backward compatibility.

  Third-party projects with a dependency on scikit-learn 0.9+ should upgrade their codebase. For instance, under Linux / MacOSX just run (make a backup first!):

  ```
  find -name "*.py" | xargs sed -i 's/\bcross_val\b/cross_validation/g'
  ```

- The `score_func` argument of the `sklearn.cross_validation.cross_val_score` function is now expected to accept `y_test` and `y_predicted` as only arguments for classification and regression tasks or `X_test` for unsupervised estimators.
- `gamma` parameter for support vector machine algorithms is set to `1 / n_features` by default, instead of `1 / n_samples`.
- The `sklearn.hmm` has been marked as orphaned: it will be removed from scikit-learn in version 0.11 unless someone steps up to contribute documentation, examples and fix lurking numerical stability issues.
- `sklearn.neighbors` has been made into a submodule. The two previously available estimators, `NeighborsClassifier` and `NeighborsRegressor` have been marked as deprecated. Their functionality has been divided among five new classes: `NearestNeighbors` for unsupervised neighbors searches, `KNeighborsClassifier` & `RadiusNeighborsClassifier` for supervised classification problems, and `KNeighborsRegressor` & `RadiusNeighborsRegressor` for supervised regression problems.
- `sklearn.ball_tree.BallTree` has been moved to `sklearn.neighbors.BallTree`. Using the former will generate a warning.
- `sklearn.linear_model.LARS()` and related classes (LassoLARS, LassoLARSCV, etc.) have been renamed to `sklearn.linear_model.Lars()`.
- All distance metrics and kernels in `sklearn.metrics.pairwise` now have a Y parameter, which by default is None. If not given, the result is the distance (or kernel similarity) between each sample in Y. If given, the result is the pairwise distance (or kernel similarity) between samples in X to Y.
- `sklearn.metrics.pairwise.l1_distance` is now called `manhattan_distance`, and by default returns the pairwise distance. For the component wise distance, set the parameter `sum_over_features` to `False`.

Backward compatibility package aliases and other deprecated classes and functions will be removed in version 0.11.

People

38 people contributed to this release.

- 387 Vlad Niculae

- 320 Olivier Grisel
- 192 Lars Buitinck
- 179 Gael Varoquaux
- 168 Fabian Pedregosa (INRIA, Parietal Team)
- 127 Jake Vanderplas
- 120 Mathieu Blondel
- 85 Alexandre Passos
- 67 Alexandre Gramfort
- 57 Peter Prettenhofer
- 56 Gilles Louppe
- 42 Robert Layton
- 38 Nelle Varoquaux
- 32 Jean Kossaifi
- 30 Conrad Lee
- 22 Pietro Berkes
- 18 andy
- 17 David Warde-Farley
- 12 Brian Holt
- 11 Robert
- 8 Amit Aides
- 8 Virgile Fritsch
- 7 Yaroslav Halchenko
- 6 Salvatore Masecchia
- 5 Paolo Losi
- 4 Vincent Schut
- 3 Alexis Metaireau
- 3 Bryan Silverthorn
- 3 Andreas Müller
- 2 Minwoo Jake Lee
- 1 Emmanuelle Gouillart
- 1 Keith Goodman
- 1 Lucas Wiman
- 1 Nicolas Pinto
- 1 Thouis (Ray) Jones
- 1 Tim Sheerman-Chase

1.17.22 Version 0.8

May 11, 2011

scikit-learn 0.8 was released on May 2011, one month after the first “international” scikit-learn coding sprint and is marked by the inclusion of important modules: *Hierarchical clustering*, *Cross decomposition*, *Non-negative matrix factorization (NMF or NNMF)*, initial support for Python 3 and by important enhancements and bug fixes.

Changelog

Several new modules where introduced during this release:

- New *Hierarchical clustering* module by Vincent Michel, Bertrand Thirion, Alexandre Gramfort and Gael Varoquaux.
- *Kernel PCA* implementation by Mathieu Blondel
- labeled_faces_in_the_wild by Olivier Grisel.
- New *Cross decomposition* module by Edouard Duchesnay.
- *Non-negative matrix factorization (NMF or NNMF)* module Vlad Niculae
- Implementation of the *Oracle Approximating Shrinkage* algorithm by Virgile Fritsch in the *Covariance estimation* module.

Some other modules benefited from significant improvements or cleanups.

- Initial support for Python 3: builds and imports cleanly, some modules are usable while others have failing tests by Fabian Pedregosa.
- `decomposition.PCA` is now usable from the Pipeline object by Olivier Grisel.
- Guide *How to optimize for speed* by Olivier Grisel.
- Fixes for memory leaks in libsvm bindings, 64-bit safer BallTree by Lars Buitinck.
- bug and style fixing in *K-means* algorithm by Jan Schlüter.
- Add attribute converged to Gaussian Mixture Models by Vincent Schut.
- Implemented `transform`, `predict_log_proba` in `discriminant_analysis.LinearDiscriminantAnalysis` By Mathieu Blondel.
- Refactoring in the *Support Vector Machines* module and bug fixes by Fabian Pedregosa, Gael Varoquaux and Amit Aides.
- Refactored SGD module (removed code duplication, better variable naming), added interface for sample weight by Peter Prettenhofer.
- Wrapped BallTree with Cython by Thouis (Ray) Jones.
- Added function `svm.l1_min_c` by Paolo Losi.
- Typos, doc style, etc. by Yaroslav Halchenko, Gael Varoquaux, Olivier Grisel, Yann Malet, Nicolas Pinto, Lars Buitinck and Fabian Pedregosa.

People

People that made this release possible preceded by number of commits:

- 159 Olivier Grisel

- 96 Gael Varoquaux
- 96 Vlad Niculae
- 94 Fabian Pedregosa
- 36 Alexandre Gramfort
- 32 Paolo Losi
- 31 Edouard Duchesnay
- 30 Mathieu Blondel
- 25 Peter Prettenhofer
- 22 Nicolas Pinto
- **11 Virgile Fritsch**
    - 7 Lars Buitinck
    - 6 Vincent Michel
    - 5 Bertrand Thirion
    - 4 Thouis (Ray) Jones
    - 4 Vincent Schut
    - 3 Jan Schlüter
    - 2 Julien Miotte
    - 2 Matthieu Perrot
    - 2 Yann Malet
    - 2 Yaroslav Halchenko
    - 1 Amit Aides
    - 1 Andreas Müller
    - 1 Feth Arezki
    - 1 Meng Xinfan

1.17.23 Version 0.7

March 2, 2011

scikit-learn 0.7 was released in March 2011, roughly three months after the 0.6 release. This release is marked by the speed improvements in existing algorithms like k-Nearest Neighbors and K-Means algorithm and by the inclusion of an efficient algorithm for computing the Ridge Generalized Cross Validation solution. Unlike the preceding release, no new modules where added to this release.

Changelog

- Performance improvements for Gaussian Mixture Model sampling [Jan Schlüter].
- Implementation of efficient leave-one-out cross-validated Ridge in `linear_model.RidgeCV` [Mathieu Blondel]

- Better handling of collinearity and early stopping in `linear_model.lars_path` [Alexandre Gramfort and Fabian Pedregosa].
- Fixes for liblinear ordering of labels and sign of coefficients [Dan Yamins, Paolo Losi, Mathieu Blondel and Fabian Pedregosa].
- Performance improvements for Nearest Neighbors algorithm in high-dimensional spaces [Fabian Pedregosa].
- Performance improvements for `cluster.KMeans` [Gael Varoquaux and James Bergstra].
- Sanity checks for SVM-based classes [Mathieu Blondel].
- Refactoring of `neighbors.NeighborsClassifier` and `neighbors.kneighbors_graph`: added different algorithms for the k-Nearest Neighbor Search and implemented a more stable algorithm for finding barycenter weights. Also added some developer documentation for this module, see notes_neighbors for more information [Fabian Pedregosa].
- Documentation improvements: Added `pca.RandomizedPCA` and `linear_model.LogisticRegression` to the class reference. Also added references of matrices used for clustering and other fixes [Gael Varoquaux, Fabian Pedregosa, Mathieu Blondel, Olivier Grisel, Virgile Fritsch , Emmanuelle Gouillart]
- Binded decision_function in classes that make use of liblinear, dense and sparse variants, like `svm.LinearSVC` or `linear_model.LogisticRegression` [Fabian Pedregosa].
- Performance and API improvements to `metrics.euclidean_distances` and to `pca.RandomizedPCA` [James Bergstra].
- Fix compilation issues under NetBSD [Kamel Ibn Hassen Derouiche]
- Allow input sequences of different lengths in `hmm.GaussianHMM` [Ron Weiss].
- Fix bug in affinity propagation caused by incorrect indexing [Xinfan Meng]

People

People that made this release possible preceded by number of commits:

- 85 Fabian Pedregosa
- 67 Mathieu Blondel
- 20 Alexandre Gramfort
- 19 James Bergstra
- 14 Dan Yamins
- 13 Olivier Grisel
- 12 Gael Varoquaux
- 4 Edouard Duchesnay
- 4 Ron Weiss
- 2 Satrajit Ghosh
- 2 Vincent Dubourg
- 1 Emmanuelle Gouillart
- 1 Kamel Ibn Hassen Derouiche
- 1 Paolo Losi

- 1 VirgileFritsch
- 1 Yaroslav Halchenko
- 1 Xinfan Meng

1.17.24 Version 0.6

December 21, 2010

scikit-learn 0.6 was released on December 2010. It is marked by the inclusion of several new modules and a general renaming of old ones. It is also marked by the inclusion of new example, including applications to real-world datasets.

Changelog

- New stochastic gradient descent module by Peter Prettenhofer. The module comes with complete documentation and examples.
- Improved svm module: memory consumption has been reduced by 50%, heuristic to automatically set class weights, possibility to assign weights to samples (see *SVM: Weighted samples* for an example).
- New *Gaussian Processes* module by Vincent Dubourg. This module also has great documentation and some very neat examples. See example_gaussian_process_plot_gp_regression.py or example_gaussian_process_plot_gp_probabilistic_classification_after_regression.py for a taste of what can be done.
- It is now possible to use liblinear's Multi-class SVC (option multi_class in `svm.LinearSVC`)
- New features and performance improvements of text feature extraction.
- Improved sparse matrix support, both in main classes (`grid_search.GridSearchCV`) as in modules sklearn.svm.sparse and sklearn.linear_model.sparse.
- Lots of cool new examples and a new section that uses real-world datasets was created. These include: *Faces recognition example using eigenfaces and SVMs*, *Species distribution modeling*, *Libsvm GUI*, *Wikipedia principal eigenvector* and others.
- Faster *Least Angle Regression* algorithm. It is now 2x faster than the R version on worst case and up to 10x times faster on some cases.
- Faster coordinate descent algorithm. In particular, the full path version of lasso (`linear_model.lasso_path`) is more than 200x times faster than before.
- It is now possible to get probability estimates from a `linear_model.LogisticRegression` model.
- module renaming: the glm module has been renamed to linear_model, the gmm module has been included into the more general mixture model and the sgd module has been included in linear_model.
- Lots of bug fixes and documentation improvements.

People

People that made this release possible preceded by number of commits:

- 207 Olivier Grisel
- 167 Fabian Pedregosa
- 97 Peter Prettenhofer
- 68 Alexandre Gramfort

- 59 Mathieu Blondel
- 55 Gael Varoquaux
- 33 Vincent Dubourg
- 21 Ron Weiss
- 9 Bertrand Thirion
- 3 Alexandre Passos
- 3 Anne-Laure Fouque
- 2 Ronan Amicel
- 1 Christian Osendorfer

1.17.25 Version 0.5

October 11, 2010

Changelog

New classes

- Support for sparse matrices in some classifiers of modules `svm` and `linear_model` (see `svm.sparse.SVC`, `svm.sparse.SVR`, `svm.sparse.LinearSVC`, `linear_model.sparse.Lasso`, `linear_model.sparse.ElasticNet`)
- New *`pipeline.Pipeline`* object to compose different estimators.
- Recursive Feature Elimination routines in module *Feature selection*.
- Addition of various classes capable of cross validation in the linear_model module (*`linear_model.LassoCV`*, *`linear_model.ElasticNetCV`*, etc.).
- New, more efficient LARS algorithm implementation. The Lasso variant of the algorithm is also implemented. See *`linear_model.lars_path`*, *`linear_model.Lars`* and *`linear_model.LassoLars`*.
- New Hidden Markov Models module (see classes `hmm.GaussianHMM`, `hmm.MultinomialHMM`, `hmm.GMMHMM`)
- New module feature_extraction (see *class reference*)
- New FastICA algorithm in module sklearn.fastica

Documentation

- Improved documentation for many modules, now separating narrative documentation from the class reference. As an example, see documentation for the SVM module and the complete class reference.

Fixes

- API changes: adhere variable names to PEP-8, give more meaningful names.
- Fixes for svm module to run on a shared memory context (multiprocessing).
- It is again possible to generate latex (and thus PDF) from the sphinx docs.

Examples

- new examples using some of the mlcomp datasets: sphx_glr_auto_examples_mlcomp_sparse_document_classif py (since removed) and *Classification of text documents using sparse features*
- Many more examples. See here the full list of examples.

External dependencies

- Joblib is now a dependency of this package, although it is shipped with (sklearn.externals.joblib).

Removed modules

- Module ann (Artificial Neural Networks) has been removed from the distribution. Users wanting this sort of algorithms should take a look into pybrain.

Misc

- New sphinx theme for the web page.

Authors

The following is a list of authors for this release, preceded by number of commits:

- 262 Fabian Pedregosa
- 240 Gael Varoquaux
- 149 Alexandre Gramfort
- 116 Olivier Grisel
- 40 Vincent Michel
- 38 Ron Weiss
- 23 Matthieu Perrot
- 10 Bertrand Thirion
- 7 Yaroslav Halchenko
- 9 VirgileFritsch
- 6 Edouard Duchesnay
- 4 Mathieu Blondel
- 1 Ariel Rokem
- 1 Matthieu Brucher

1.17.26 Version 0.4

August 26, 2010

Changelog

Major changes in this release include:

- Coordinate Descent algorithm (Lasso, ElasticNet) refactoring & speed improvements (roughly 100x times faster).
- Coordinate Descent Refactoring (and bug fixing) for consistency with R's package GLMNET.
- New metrics module.
- New GMM module contributed by Ron Weiss.
- Implementation of the LARS algorithm (without Lasso variant for now).
- feature_selection module redesign.
- Migration to GIT as version control system.
- Removal of obsolete attrselect module.
- Rename of private compiled extensions (added underscore).
- Removal of legacy unmaintained code.
- Documentation improvements (both docstring and rst).
- Improvement of the build system to (optionally) link with MKL. Also, provide a lite BLAS implementation in case no system-wide BLAS is found.
- Lots of new examples.
- Many, many bug fixes …

Authors

The committer list for this release is the following (preceded by number of commits):

- 143 Fabian Pedregosa
- 35 Alexandre Gramfort
- 34 Olivier Grisel
- 11 Gael Varoquaux
- 5 Yaroslav Halchenko
- 2 Vincent Michel
- 1 Chris Filo Gorgolewski

1.17.27 Earlier versions

Earlier versions included contributions by Fred Mailhot, David Cooke, David Huard, Dave Morrill, Ed Schofield, Travis Oliphant, Pearu Peterson.

1.18 Roadmap

1.18.1 Purpose of this document

This document list general directions that core contributors are interested to see developed in scikit-learn. The fact that an item is listed here is in no way a promise that it will happen, as resources are limited. Rather, it is an indication that help is welcomed on this topic.

1.18.2 Statement of purpose: Scikit-learn in 2018

Eleven years after the inception of Scikit-learn, much has changed in the world of machine learning. Key changes include:

- Computational tools: The exploitation of GPUs, distributed programming frameworks like Scala/Spark, etc.
- High-level Python libraries for experimentation, processing and data management: Jupyter notebook, Cython, Pandas, Dask, Numba…
- Changes in the focus of machine learning research: artificial intelligence applications (where input structure is key) with deep learning, representation learning, reinforcement learning, domain transfer, etc.

A more subtle change over the last decade is that, due to changing interests in ML, PhD students in machine learning are more likely to contribute to PyTorch, Dask, etc. than to Scikit-learn, so our contributor pool is very different to a decade ago.

Scikit-learn remains very popular in practice for trying out canonical machine learning techniques, particularly for applications in experimental science and in data science. A lot of what we provide is now very mature. But it can be costly to maintain, and we cannot therefore include arbitrary new implementations. Yet Scikit-learn is also essential in defining an API framework for the development of interoperable machine learning components external to the core library.

Thus our main goals in this era are to:

- continue maintaining a high-quality, well-documented collection of canonical tools for data processing and machine learning within the current scope (i.e. rectangular data largely invariant to column and row order; predicting targets with simple structure)
- improve the ease for users to develop and publish external components
- improve inter-operability with modern data science tools (e.g. Pandas, Dask) and infrastructures (e.g. distributed processing)

Many of the more fine-grained goals can be found under the API tag on the issue tracker.

1.18.3 Architectural / general goals

The list is numbered not as an indication of the order of priority, but to make referring to specific points easier. Please add new entries only at the bottom.

1. Everything in Scikit-learn should conform to our API contract
    - *`Pipeline`* and *`FeatureUnion`* modify their input parameters in fit. Fixing this requires making sure we have a good grasp of their use cases to make sure all current functionality is maintained. #8157 #7382
2. Improved handling of Pandas DataFrames and SparseDataFrames
    - document current handling
    - column reordering issue #7242

   - avoiding unnecessary conversion to ndarray
   - returning DataFrames from transformers #5523
   - getting DataFrames from dataset loaders
   - Sparse currently not considered

3. Improved handling of categorical features
   - Tree-based models should be able to handle both continuous and categorical features #4899
   - In dataset loaders
   - As generic transformers to be used with ColumnTransforms (e.g. ordinal encoding supervised by correlation with target variable)
4. Improved handling of missing data
   - Making sure meta-estimators are lenient towards missing data
   - Non-trivial imputers
   - Learners directly handling missing data
   - An amputation sample generator to make parts of a dataset go missing
   - Handling mixtures of categorical and continuous variables
5. Passing around information that is not (X, y): Sample properties
   - We need to be able to pass sample weights to scorers in cross validation.
   - We should have standard/generalised ways of passing sample-wise properties around in meta-estimators. #4497 #7646
6. Passing around information that is not (X, y): Feature properties
   - Feature names or descriptions should ideally be available to fit for, e.g. . #6425 #6424
   - Per-feature handling (e.g. “is this a nominal / ordinal / English language text?”) should also not need to be provided to estimator constructors, ideally, but should be available as metadata alongside X. #8480
7. Passing around information that is not (X, y): Target information
   - We have problems getting the full set of classes to all components when the data is split/sampled. #6231 #8100
   - We have no way to handle a mixture of categorical and continuous targets.
8. Make it easier for external users to write Scikit-learn-compatible components
   - More flexible estimator checks that do not select by estimator name #6599 #6715
   - Example of how to develop a meta-estimator
   - More self-sufficient running of scikit-learn-contrib or a similar resource
9. Support resampling and sample reduction
   - Allow subsampling of majority classes (in a pipeline?) #3855
   - Implement random forests with resampling #8732
10. Better interfaces for interactive development
    - __repr__ and HTML visualisations of estimators #6323
    - Include plotting tools, not just as examples. #9173

11. Improved tools for model diagnostics and basic inference
    - alternative feature importances implementations (e.g. methods or wrappers)
    - better ways to handle validation sets when fitting
    - better ways to find thresholds / create decision rules #8614
12. Better tools for selecting hyperparameters with transductive estimators
    - Grid search and cross validation are not applicable to most clustering tasks. Stability-based selection is more relevant.
13. Improved tracking of fitting
    - Verbose is not very friendly and should use a standard logging library #6929
    - Callbacks or a similar system would facilitate logging and early stopping
14. Distributed parallelism
    - Joblib can now plug onto several backends, some of them can distribute the computation across computers
    - However, we want to stay high level in scikit-learn
15. A way forward for more out of core
    - Dask enables easy out-of-core computation. While the dask model probably cannot be adaptable to all machine-learning algorithms, most machine learning is on smaller data than ETL, hence we can maybe adapt to very large scale while supporting only a fraction of the patterns.
16. Better support for manual and automatic pipeline building
    - Easier way to construct complex pipelines and valid search spaces #7608 #5082 #8243
    - provide search ranges for common estimators??
    - cf. searchgrid
17. Support for working with pre-trained models
    - Estimator “freezing”. In particular, right now it’s impossible to clone a `CalibratedClassifierCV` with prefit. #8370. #6451
18. Backwards-compatible de/serialization of some estimators
    - Currently serialization (with pickle) breaks across versions. While we may not be able to get around other limitations of pickle re security etc, it would be great to offer cross-version safety from version 1.0. Note: Gael and Olivier think that this can cause heavy maintenance burden and we should manage the trade-offs. A possible alternative is presented in the following point.
19. Documentation and tooling for model lifecycle management
    - Document good practices for model deployments and lifecycle: before deploying a model: snapshot the code versions (numpy, scipy, scikit-learn, custom code repo), the training script and an alias on how to retrieve historical training data + snapshot a copy of a small validation set + snapshot of the predictions (predicted probabilities for classifiers) on that validation set.
    - Document and tools to make it easy to manage upgrade of scikit-learn versions:
      - Try to load the old pickle, if it works, use the validation set prediction snapshot to detect that the serialized model still behave the same;
      - If joblib.load / pickle.load not work, use the versioned control training script + historical training set to retrain the model and use the validation set prediction snapshot to assert that it is possible to recover the previous predictive performance: if this is not the case there is probably a bug in scikit-learn that needs to be reported.

20. (Optional) Improve scikit-learn common tests suite to make sure that (at least for frequently used) models have stable predictions across-versions (to be discussed);

    - Extend documentation to mention how to deploy models in Python-free environments for instance ONNX. and use the above best practices to assess predictive consistency between scikit-learn and ONNX prediction functions on validation set.
    - Document good practices to detect temporal distribution drift for deployed model and good practices for re-training on fresh data without causing catastrophic predictive performance regressions.

21. More didactic documentation

    - More and more options have been added to scikit-learn. As a result, the documentation is crowded which makes it hard for beginners to get the big picture. Some work could be done in prioritizing the information.

1.18.4 Subpackage-specific goals

sklearn.cluster

- kmeans variants for non-Euclidean distances, if we can show these have benefits beyond hierarchical clustering.

sklearn.ensemble

- a stacking implementation

sklearn.model_selection

- multi-metric scoring is slow #9326
- perhaps we want to be able to get back more than multiple metrics
- the handling of random states in CV splitters is a poor design and contradicts the validation of similar parameters in estimators.
- exploit warm-starting and path algorithms so the benefits of `EstimatorCV` objects can be accessed via *GridSearchCV* and used in Pipelines. #1626
- Cross-validation should be able to be replaced by OOB estimates whenever a cross-validation iterator is used.
- Redundant computations in pipelines should be avoided (related to point above) cf daskml

sklearn.neighbors

- Ability to substitute a custom/approximate/precomputed nearest neighbors implementation for ours in all/most contexts that nearest neighbors are used for learning. #10463

sklearn.pipeline

- Performance issues with `Pipeline.memory`
- see "Everything in Scikit-learn should conform to our API contract" above

1.19 Scikit-learn governance and decision-making

The purpose of this document is to formalize the governance process used by the scikit-learn project, to clarify how decisions are made and how the various elements of our community interact. This document establishes a decision-making structure that takes into account feedback from all members of the community and strives to find consensus, while avoiding any deadlocks.

This is a meritocratic, consensus-based community project. Anyone with an interest in the project can join the community, contribute to the project design and participate in the decision making process. This document describes how that participation takes place and how to set about earning merit within the project community.

1.19.1 Roles And Responsibilities

Contributors

Contributors are community members who contribute in concrete ways to the project. Anyone can become a contributor, and contributions can take many forms – not only code – as detailed in the *contributors guide*.

Core developers

Core developers are community members who have shown that they are dedicated to the continued development of the project through ongoing engagement with the community. They have shown they can be trusted to maintain Scikit-learn with care. Being a core developer allows contributors to more easily carry on with their project related activities by giving them direct access to the project's repository and is represented as being an organization member on the scikit-learn GitHub organization. Core developers are expected to review code contributions, can merge approved pull requests, can cast votes for and against merging a pull-request, and can be involved in deciding major changes to the API.

New core developers can be nominated by any existing core developers. Once they have been nominated, there will be a vote by the current core developers. Voting on new core developers is one of the few activities that takes place on the project's private management list. While it is expected that most votes will be unanimous, a two-thirds majority of the cast votes is enough. The vote needs to be open for at least 1 week.

Core developers that have not contributed to the project (commits or GitHub comments) in the past 12 months will be asked if they want to become emeritus core developers and recant their commit and voting rights until they become active again. The list of core developers, active and emeritus (with dates at which they became active) is public on the scikit-learn website.

Technical Committee

The Technical Committee (TC) members are core developers who have additional responsibilities to ensure the smooth running of the project. TC members are expected to participate in strategic planning, and approve changes to the governance model. The purpose of the TC is to ensure a smooth progress from the big-picture perspective. Indeed changes that impact the full project require a synthetic analysis and a consensus that is both explicit and informed. In cases that the core developer community (which includes the TC members) fails to reach such a consensus in the required time frame, the TC is the entity to resolve the issue. Membership of the TC is by nomination by a core developer. A nomination will result in discussion which cannot take more than a month and then a vote by the core developers which will stay open for a week. TC membership votes are subject to a two-third majority of all cast votes as well as a simple majority approval of all the current TC members. TC members who do not actively engage with the TC duties are expected to resign.

The initial Technical Committee of scikit-learn consists of Alexandre Gramfort, Olivier Grisel, Andreas Müller, Joel Nothman, Hanmin Qin, Gaël Varoquaux, and Roman Yurchak.

1.19.2 Decision Making Process

Decisions about the future of the project are made through discussion with all members of the community. All non-sensitive project management discussion takes place on the project contributors' mailing list and the issue tracker. Occasionally, sensitive discussion occurs on a private list.

Scikit-learn uses a “consensus seeking” process for making decisions. The group tries to find a resolution that has no open objections among core developers. At any point during the discussion, any core-developer can call for a vote, which will conclude one month from the call for the vote. Any vote must be backed by a *SLEP*. If no option can gather two thirds of the votes cast, the decision is escalated to the TC, which in turn will use consensus seeking with the fallback option of a simple majority vote if no consensus can be found within a month. This is what we hereafter may refer to as “the decision making process”.

Decisions (in addition to adding core developers and TC membership as above) are made according to the following rules:

- **Minor Documentation changes**, such as typo fixes, or addition / correction of a sentence, but no change of the scikit-learn.org landing page or the “about” page: Requires +1 by a core developer, no -1 by a core developer (lazy consensus), happens on the issue or pull request page. Core developers are expected to give “reasonable time” to others to give their opinion on the pull request if they’re not confident others would agree.
- **Code changes and major documentation changes** require +1 by two core developers, no -1 by a core developer (lazy consensus), happens on the issue of pull-request page.
- **Changes to the API principles and changes to dependencies or supported versions** happen via a *Enhancement proposals (SLEPs)* and follows the decision-making process outlined above.
- **Changes to the governance model** use the same decision process outlined above.

If a veto -1 vote is cast on a lazy consensus, the proposer can appeal to the community and core developers and the change can be approved or rejected using the decision making procedure outlined above.

1.19.3 Enhancement proposals (SLEPs)

For all votes, a proposal must have been made public and discussed before the vote. Such proposal must be a consolidated document, in the form of a ‘Scikit-Learn Enhancement Proposal’ (SLEP), rather than a long discussion on an issue. A SLEP must be submitted as a pull-request to enhancement proposals using the SLEP template.