scikit-learn user guide

Release 0.18.2

scikit-learn developers

Jun 28, 2017

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 (>= 2.6 or >= 3.3),
- NumPy (>= 1.6.1),
- SciPy (>= 0.9).

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.

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.

For installation instructions for particular operating systems or for compiling the bleeding edge version, see the *Advanced installation instructions*.

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 build 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.

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:

http://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 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`.

1.2.7 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*.

1.2.8 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.

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 list it under *Related Projects*.

1.2.9 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. Also see this thread on the mailing list.

1.2.10 Why did you remove HMMs from scikit-learn?

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

1.2.11 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)

1.2.12 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.

1.2.13 Do you support PyPy?

In case you didn't know, PyPy is the new, fast, just-in-time compiling Python implementation. We don't support it. When the NumPy support in PyPy is complete or near-complete, and SciPy is ported over as well, we can start thinking of a port. We use too much of NumPy to work with a partial implementation.

1.2.14 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]])
>>> dbscan(X, metric=lev_metric, eps=5, min_samples=2)
([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.

1.2.15 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.

1.2.16 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.

1.2.17 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.

1.2.18 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.

1.3 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: http://webchat.freenode.net

1.3.5 Documentation resources

This documentation is relative to 0.18.2. Documentation for other versions can be found here:

- 0.17
- 0.16
- 0.15

Printable pdf documentation for all versions can be found here.

1.4 Related Projects

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.

- ML Frontend provides dataset management and SVM fitting/prediction through web-based and programmatic interfaces.
- sklearn_pandas bridge for scikit-learn pipelines and pandas data frame with dedicated transformers.
- 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.
- 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.
- sklearn-pmml Serialization of (some) scikit-learn estimators into PMML.
- sklearn2pmml Serialization of a wide variety of scikit-learn estimators and transformers into PMML with the help of JPMML-SkLearn library.

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.

- 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
- lightning Fast state-of-the-art linear model solvers (SDCA, AdaGrad, SVRG, SAG, etc...).
- 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.
- py-earth Multivariate adaptive regression splines
- sklearn-compiledtrees Generate a C++ implementation of the predict function for decision trees (and ensembles) trained by sklearn. Useful for latency-sensitive production environments.
- lda: Fast implementation of Latent Dirichlet Allocation in Cython.
- Sparse Filtering Unsupervised feature learning based on sparse-filtering
- Kernel Regression Implementation of Nadaraya-Watson kernel regression with automatic bandwidth selection
- gplearn Genetic Programming for symbolic regression tasks.
- nolearn A number of wrappers and abstractions around existing neural network libraries
- sparkit-learn Scikit-learn functionality and API on PySpark.
- keras Theano-based Deep Learning library.
- mlxtend Includes a number of additional estimators as well as model visualization utilities.
- 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.
- lasagne A lightweight library to build and train neural networks in Theano.
- multiisotonic Isotonic regression on multidimensional features.
- spherecluster Spherical K-means and mixture of von Mises Fisher clustering routines for data on the unit hypersphere.

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.
- REP Environment for conducting data-driven research in a consistent and reproducible way
- Sacred Tool to help you configure, organize, log and reproduce experiments
- gensim A library for topic modelling, document indexing and similarity retrieval
- 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.
- NiLearn Machine learning for neuro-imaging.
- AstroML Machine learning for astronomy.
- MSMBuilder Machine learning for protein conformational dynamics time series.

1.4.4 Snippets and tidbits

The wiki has more!

1.5 About us

This is a community effort, and as such many people have contributed to it over the years.

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 People

The following people have been core contributors to scikit-learn's development and maintenance:

- Mathieu Blondel
- Matthieu Brucher
- Lars Buitinck
- David Cournapeau
- Noel Dawe
- Vincent Dubourg
- Edouard Duchesnay
- Tom Dupré la Tour
- Alexander Fabisch
- Virgile Fritsch
- Satra Ghosh
- Angel Soler Gollonet
- Chris Filo Gorgolewski
- Alexandre Gramfort
- Olivier Grisel
- Jaques Grobler
- Yaroslav Halchenko
- Brian Holt
- Arnaud Joly
- Thouis (Ray) Jones

- Kyle Kastner
- Manoj Kumar
- Robert Layton
- Wei Li
- Paolo Losi
- Gilles Louppe
- Jan Hendrik Metzen
- Vincent Michel
- Jarrod Millman
- Andreas Müller (release manager)
- Vlad Niculae
- Joel Nothman
- Alexandre Passos
- Fabian Pedregosa
- Peter Prettenhofer
- Bertrand Thirion
- Jake VanderPlas
- Nelle Varoquaux
- Gael Varoquaux
- Ron Weiss

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

1.5.3 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},
}
```

1.5.4 Artwork

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

1.5.5 Funding

INRIA actively supports this project. It has provided funding for Fabian Pedregosa (2010-2012), Jaques Grobler (2012-2013) and Olivier Grisel (2013-2015) 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).

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 funds Manoj Kumar (2014), Tom Dupré la Tour

(2015), Raghav RV (2015-2016) and Thierry Guillemot (2016) to work on scikit-learn.

Columbia University funds Andreas Mueller since 2016. 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

1.5.6 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 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

1.6.2 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

1.6.3 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

1.6.4 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

1.6.5 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

1.6.6 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

1.6.7 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

1.6.8 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

1.6.9 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

1.6.10 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

1.6.11 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

1.6.12 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

1.6.13 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

1.6.14 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

1.6.15 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

1.6.16 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.

1.6.17 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

1.6.18 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

1.6.19 Data Publica

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

1.6.20 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

1.6.21 solido

solido
DESIGN AUTOMATION

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.

1.6.22 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.

1.6.23 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

1.6.24 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

1.7 Release history

1.7.1 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

1.7.2 Version 0.18.1

November 11, 2016

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

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.

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

Code Contributors

Aashi, affanv14, Alexander Junge, Alexandre Gramfort, Aman Dalmia, Andreas Mueller, Andrew Jackson, Andrew Smith, Angus Williams, Artem Golubin, Arthur Douillard, Artsiom, Bertrand Thirion, Bing Tian Dai, Brian Burns, CJ Carey, Charlton Austin, chkoar, Dave Elliott, David Kirkby, Deborah Gertrude Digges, ditenberg, E. Lynch-Klarup, Ekaterina Krivich, Fabian Egli, ferria, fukatani, Gael Varoquaux, Giorgio Patrini, Grzegorz Szpak, He Chen, guoci, Ibraim Ganiev, Iván Vallés, JPFrancoia, Jake VanderPlas, Joel Nothman, Jon Crall, Jonathan Rahn, Jonathan Striebel, Josh Karnofsky, Julien Aubert, Kathy Chen, Kaushik Lakshmikanth, Kevin Yap, Kyle Gilliam, ljwolf, Loic Esteve, Mainak Jas, Maniteja Nandana, Mathieu Blondel, Mehul Ahuja, Michele Lacchia, Mikhail Korobov, Nelle Varoquaux, Nelson Liu, Nicole Vavrova, nuffe, Olivier Grisel, Om Prakash, Patrick Carlson, Pieter Arthur de Jong, polmauri, Rafael Possas, Raghav R V, Ruifeng Zheng, Sam Shleifer, Sebastian Saeger, Sourav Singh, Srivatsan, Thierry Guillemot, toastedcornflakes, Tom Dupré la Tour, vibrantabhi19, waterponey

1.7.3 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-ouput 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 sphx_glr_plot_compare_reduction.py. 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 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.

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

1.7.4 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.

1.7.5 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.PassiveAgressiveClassifier`. 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

1.7.6 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

1.7.7 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.PassiveAgressiveClassifier` and `linear_model.PassiveAgressiveRegressor` 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

1.7.8 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.

1.7.9 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.

1.7.10 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

1.7.11 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

1.7.12 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

1.7.13 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 *Partial Dependence Plots* 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

1.7.14 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

1.7.15 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`* 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

1.7.16 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 models*: 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

1.7.17 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

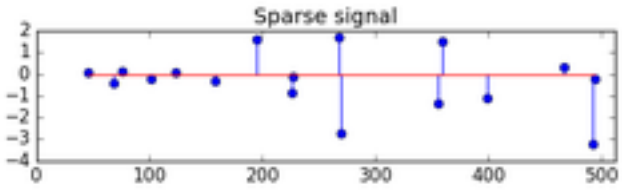
- 1 Félix-Antoine Fortin
- 1 Juan Manuel Caicedo Carvajal
- 1 Nelle Varoquaux
- 1 Nicolas Pinto
- 1 Tiziano Zito
- 1 Xinfan Meng

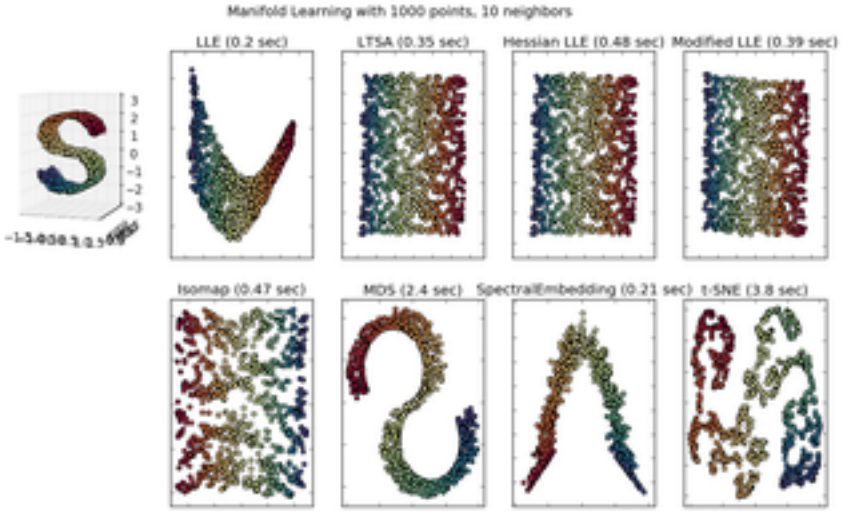
1.7.18 Version 0.9

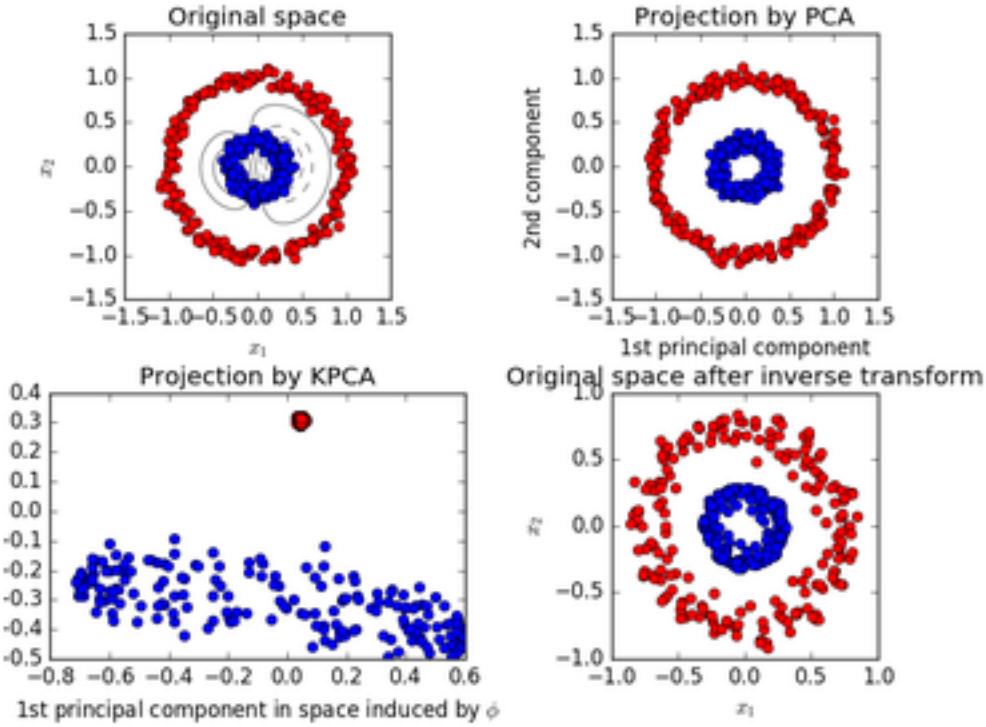
September 21, 2011

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.
- *Sample generators* 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.
- *Downloading datasets from the mldata.org repository* utilities by Pietro Berkes.
- *The Olivetti faces dataset* 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

1.7.19 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
- *The Labeled Faces in the Wild face recognition dataset* 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

1.7.20 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

1.7.21 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

1.7.22 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` (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

1.7.23 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

1.7.24 Earlier versions

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