Frequently Asked Questions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 7)

Unknown directive type "currentmodule".

```
.. currentmodule:: sklearn
```

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

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.

How do you pronounce the project name?

sy-kit learn. sci stands for science!

Why scikit?

There are multiple scikits, which are scientific toolboxes built around SciPy. Apart from scikit-learn, another popular one is scikit-image.

How can I contribute to scikit-learn?

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

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 27); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 27); *backlink*

Unknown interpreted text role "ref".

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

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

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

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

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

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

https://stackoverflow.com/help/mcve

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

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

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

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

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

See :ref:`model_persistence`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 71); *backlink*

Unknown interpreted text role "ref".

How can I create a bunch object?

Bunch objects are sometimes used as an output for functions and methods. They extend dictionaries by enabling values to be accessed by key, *bunch["value_key"]*, or by an attribute, *bunch.value_key*.

They should not be used as an input; therefore you almost never need to create a `Bunch` object, unless you are extending the scikit-learn's API.

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 :ref:`loading external datasets <external_datasets>`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 90); *backlink*

Unknown interpreted text role "ref".

What are the inclusion criteria for new algorithms ?

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

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

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

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

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

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

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 125); *backlink*

Unknown interpreted text role "ref".

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

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

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 137); *backlink*

Unknown interpreted text role "ref".

Why did you remove HMMs from scikit-learn?

See :ref:`adding_graphical_models`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 151); *backlink*

Unknown interpreted text role "ref".

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 projects 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)

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.

Do you support PyPy?

scikit-learn is regularly tested and maintained to work with PyPy (an alternative Python implementation with a built-in just-in-time compiler).

Note however that this support is still considered experimental and specific components might behave slightly differently. Please refer to the test suite of a the specific module of interest for more details.

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 :ref:`text_feature_extraction` for the built-in *text vectorizers*. For more general feature extraction from any kind of data, see :ref:`dict_feature_extraction` and :ref:`feature_hashing`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 208); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 208); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 208); *backlink*

Unknown interpreted text role "ref".

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

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 an 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 main 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.

Why does my job use more cores than specified with n_jobs?

This is because `n_jobs` only controls the number of jobs for routines that are parallelized with `joblib`, but parallel code can come from other sources:

- some routines may be parallelized with OpenMP (for code written in C or Cython).
- scikit-learn relies a lot on numpy, which in turn may rely on numerical libraries like MKL, OpenBLAS or BLIS which can provide parallel implementations.

For more details, please refer to our :ref:`Parallelism notes <parallelism>`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, **line 315**); ***backlink***
>
> Unknown interpreted text role "ref".

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

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

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

Note that scikit-learn currently implements a simple multilayer perceptron in :mod:`sklearn.neural_network`. We will only accept bug fixes for this module. If you want to implement more complex deep learning models, please turn to popular deep learning frameworks such as tensorflow, keras and pytorch.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, **line 331**); ***backlink***
>
> Unknown interpreted text role "mod".

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.

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

Please refer to :ref:`randomness`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, **line 360**); ***backlink***
>
> Unknown interpreted text role "ref".

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

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

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, **line 365**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, **line 365**); ***backlink***
>
> Unknown interpreted text role "ref".

Why does Scikit-learn not directly work with, for example,

pandas.DataFrame?

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

Do you plan to implement transform for target y in a pipeline?

Currently transform only works for features X in a pipeline. There's a long-standing discussion about not being able to transform y in a pipeline. Follow on github issue #4143. Meanwhile check out :class:`~compose.TransformedTargetRegressor`, pipegraph, imbalanced-learn. Note that Scikit-learn solved for the case where y has an invertible transformation applied before training and inverted after prediction. Scikit-learn intends to solve for use cases where y should be transformed at training time and not at test time, for resampling and similar uses, like at *imbalanced-learn*. In general, these use cases can be solved with a custom meta estimator rather than a Pipeline

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 384); *backlink***
>
> Unknown interpreted text role "class".

Why are there so many different estimators for linear models?

Usually, there is one classifier and one regressor per model type, e.g. :class:`~ensemble.GradientBoostingClassifier` and :class:`~ensemble.GradientBoostingRegressor`. Both have similar options and both have the parameter *loss*, which is especially useful in the regression case as it enables the estimation of conditional mean as well as conditional quantiles.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 404); *backlink***
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 404); *backlink***
>
> Unknown interpreted text role "class".

For linear models, there are many estimator classes which are very close to each other. Let us have a look at

- :class:`~linear_model.LinearRegression`, no penalty

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 414); *backlink***
  >
  > Unknown interpreted text role "class".

- :class:`~linear_model.Ridge`, L2 penalty

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 415); *backlink***
  >
  > Unknown interpreted text role "class".

- :class:`~linear_model.Lasso`, L1 penalty (sparse models)

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 416); *backlink***
  >
  > Unknown interpreted text role "class".

- :class:`~linear_model.ElasticNet`, L1 + L2 penalty (less sparse models)

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, line 417); *backlink***
  >
  > Unknown interpreted text role "class".

- :class:`~linear_model.SGDRegressor` with *loss='squared_loss'*

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\doc\[scikit-learn-main][doc]faq.rst`, **line 418)**; ***backlink***

Unknown interpreted text role "class".

Maintainer perspective: They all do in principle the same and are different only by the penalty they impose. This, however, has a large impact on the way the underlying optimization problem is solved. In the end, this amounts to usage of different methods and tricks from linear algebra. A special case is *SGDRegressor* which comprises all 4 previous models and is different by the optimization procedure. A further side effect is that the different estimators favor different data layouts (*X* c-contiguous or f-contiguous, sparse csr or csc). This complexity of the seemingly simple linear models is the reason for having different estimator classes for different penalties.

User perspective: First, the current design is inspired by the scientific literature where linear regression models with different regularization/penalty were given different names, e.g. *ridge regression*. Having different model classes with according names makes it easier for users to find those regression models. Secondly, if all the 5 above mentioned linear models were unified into a single class, there would be parameters with a lot of options like the `solver` parameter. On top of that, there would be a lot of exclusive interactions between different parameters. For example, the possible options of the parameters `solver`, `precompute` and `selection` would depend on the chosen values of the penalty parameters `alpha` and `l1_ratio`.