

Documentation

experience from scikit-learn

Olivier Grisel - Inria dev meeting - April 2015

Why documentation

- Make it easy for existing users to solve their problems using your software
- Make project discoverable to potential users who don't know about your project yet
- Make it explicit for contributors to understand the scope of the project

Doc as marketing

- Target: potential users who do not know yet about your project
- Doc is full of keywords related to your project
- Make it easy for people answering questions on StackOverflow with a link for details.
- Organic SEO: best marketing tool

Web

Vidéos

Images

Livres

Actualités

Plus ▾

Outils de recherche

Environ 7 540 000 résultats (0,22 secondes)

Articles universitaires correspondant aux termes **support vector machine**

Support vector machines - Hearst - Cité 887 fois**Support vector machines** - Steinwart - Cité 771 fois**Support vector machines** - Shawe-Taylor - Cité 462 fois

Machine à vecteurs de support — Wikipédia

fr.wikipedia.org/wiki/Machine_à_vecteurs_de_support ▾

Aller à [Les SVM pour la régression](#) - [modifier | modifier le code]. Cette section est vide, insuffisamment détaillée ou incomplète. Votre aide est la ...

[Historique](#) - [Résumé intuitif](#) - [Principe général](#) - [Extensions](#)

Support vector machine - Wikipedia, the free encyclopedia

en.wikipedia.org/wiki/Support_vector_machine ▾ [Traduire cette page](#)

In machine learning, **support vector machines** (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that ...

[Kernel method](#) - [Vladimir Vapnik](#) - [Hyperplane](#) - [Quadratic programming](#)

[...]

Support Vector Machines (SVM) - StatSoft

www.statsoft.com/textbook/support-vector-machines ▾ [Traduire cette page](#)

Support Vector Machines are based on the concept of decision planes that define decision boundaries.

1.4. Support Vector Machines — scikit-learn 0.16.1 ...

scikit-learn.org/stable/modules/svm.html ▾ [Traduire cette page](#)

The **support vector machines** in scikit-learn support both dense (`numpy.ndarray` and convertible to that by `numpy.asarray`) and sparse (any `scipy.sparse`) sample ...



support vector machine python



Web

Shopping

Vidéos

Images

Actualités

Plus ▾

Outils de recherche

Environ 264 000 résultats (0,28 secondes)

Articles universitaires correspondant aux termes **support vector machine python**

A practical guide to **support vector** classification - Hsu - Cité 3744 fois

LIBSVM: a library for **support vector machines** - Chang - Cité 21708 fois

Scikit-learn: **Machine** learning in **Python** - Pedregosa - Cité 1321 fois

1.4. Support Vector Machines — scikit-learn 0.16.1 ...

scikit-learn.org/stable/modules/svm.html ▾ Traduire cette page

Using **Python** functions as kernels; 1.4.6.1.2. ... **Support vector machines** (SVMs) are a set of supervised learning methods used for classification, regression and ...
`sklearn.svm.SVC` - `sklearn.svm.LinearSVC` - Plot different SVM classifiers in ...

`sklearn.svm.SVC` — scikit-learn 0.16.1 documentation

scikit-learn.org/stable/modules/.../sklearn.svm.SVC.ht... ▾ Traduire cette page

`sklearn.svm.SVC`¶. class `sklearn.svm.SVC`(C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, ...

LIBSVM -- A Library for Support Vector Machines

www.csie.ntu.edu.tw/~cjlin/libsvm/ ▾ Traduire cette page

15 nov. 2014 - A practical guide to **SVM** classification is available now! (mainly written for ... To use this tool, you also need to install **python** and **gnuplot**.

What documentation

- Narrative documentation
 - Tutorial / Quick-start
 - Detailed explanation of a module / set of features
- Runnable code examples
- API reference extracted from in-code docstrings (e.g. sphinx-doc.org for Python & C/C++ projects)



Documentation of scikit-learn 0.16.1

Quick Start

A very short introduction into machine learning problems and how to solve them using scikit-learn. Introduced basic concepts and conventions.

Tutorials

Useful tutorials for developing a feel for some of scikit-learn's applications in the machine learning field.

Contributing

Information on how to contribute. This also contains useful information for advanced users, for example how to build their own estimators.

User Guide

The main documentation. This contains an in-depth description of all algorithms and how to apply them.

API

The exact API of all functions and classes, as given by the docstrings. The API documents expected types and allowed features for all functions, and all parameters available for the algorithms.

Flow Chart

A graphical overview of basic areas of machine learning, and guidance which kind of algorithms to use in a given situation.

Other Versions

- [scikit-learn 0.17.dev0 \(development\)](#)
- [scikit-learn 0.16 \(stable\)](#)
- [scikit-learn 0.15](#)
- [scikit-learn 0.14](#)
- [scikit-learn 0.13](#)
- Older versions

Additional Resources

Talks given, slide-sets and other information relevant to scikit-learn.

FAQ

Frequently asked questions about the project and contributing.

How: make it mandatory

- Make it a requirement for all code contributions
 - Make it explicit in the documentation of the contribution process
 - Never merge a pull-request if it does not include the new / updated doc
- Automated tests to check that the documentation inline examples are aligned with the code (e.g. `doctest` in Python)

How to write: think as a reader

- 2 main possible user intentions
 - New user want to “get started” without a particular use-case in mind:
 - tutorial / getting started section
 - Existing user want to solve a specific problem:
 - narrative documentation / runnable examples

How to write: empathy

- Do not start from the generic theory of everything & let the reader use logic / inference to derive specific case of interest.
- Choose a specific example that is similar to the use case of 80% of the intended users of that module
 - In-line code example to get the idea
 - Explain typical use case where module is suitable
 - Main parameters that need to be adjusted, tips and tricks
 - Then at the end: mathematical definitions, links to reference

Stuff to document

- Which components work well together and how
- How to set important parameters (rules of thumb)
- Why choose this instead of an alternative module / option (pros and cons).
- Complexity scaling: time (CPU) and space (RAM) with typical numbers

Internal linking

- Increase discoverability & reader rerouting:
 - API reference is not a how-to solve a problem
 - Tutorial should not be exhaustive / give details
- Link to API reference from narrative and source of runnable examples
- “See also” links to similar classes and functions in API doc
- Link to runnable examples from narrative doc
- Back-links to narrative from examples

Automate publishing

- Build the documentation for the developer version and publish it automatically:
 - For Python: <http://readthedoc.org> or travis / jenkins
 - Make it faster / easier to spot HTML formatting issues
 - Less boring work to do at release time.

Thank you!