Contributing to Surprise

========================

Disclamer: please note that starting from version 1.1.0, only bugfixes and

documentation improvements are considered. We will not accept new features.

Before submitting a new pull request, please make sure that:

\* Your code is [clean](https://www.youtube.com/watch?v=wf-BqAjZb8M),

[pythonic](https://www.youtube.com/watch?v=OSGv2VnC0go), well commented and

also well documented (see below for building the docs).

\* The tests are passing. Also, write some tests for the changes you're

proposing. If you're not willing to write tests, it's best not to submit a PR

(it's just a waste of time for everyone).

\* Your code follows [PEP 8](https://www.python.org/dev/peps/pep-0008/) as much

as possible. Coding style is automatically checked when tests are run. About

line length: it's best to respect to 80 columns constraint, but tests will

pass as long as the length is less than 88.

\* For new prediction algorithms or similarity metrics, please submit a

relevent benchmark outlining the performance of the new feature (in terms of

accuracy, computation time, etc.). You can take a look at

[`examples/benchmarks`](https://github.com/NicolasHug/Surprise/blob/master/examples/benchmark.py)

for inspiration.

Set up

------

It's highly recommended to use a virtual environment. All the packages needed

for the development of Surprise (sphinx, flake8, etc...) can be installed by

running

pip install -r requirements\_dev.txt

Then, you can install your local copy of the repo:

pip install -e .

Any change to the code should now be immediately reflected during execution. If

you're modifying Cython code (`.pyx` files), you'll need to compile the code in

order to see the chanes. This can be achieved by running `pip install -e .`

again.

Running and writing tests

-------------------------

Our testing tool is [pytest](http://doc.pytest.org/en/latest/). Running the tests is as

simple as running

pytest

in the root directory.

For writing new tests, check out pytest getting started guide and / or take

inspiration from the current tests in the `tests` directory.

Building the docs locally

-------------------------

The docs can be compiled with

cd doc

make html

You can check the results in `doc/build/html`. Please make sure that the docs

compile without errors. Run `make clean` from time to time in order to avoid

hidden warnings. You can check spelling mistakes by running

make spelling

Legit words that are not recognized can be added in the

`source/spelling\_wordlist.txt` file.