Labels

The list of labels that can be attached to issues or pull requests is available here.

- Module labels: eg "module:ensemble". They identify the scikit-learn module concerned by the issue or the pull request. Pull requests are automatically labeled via a github action scanning the files modified by the pull request itself. Issues cannot be easily automatically labeled, triagers can do that manually.

  Why they are useful? For follow-up reasons, they can help to identify which modules are the most used or which are less maintained (issues or pull requests systematically unanswered). This is a very common approach in a number of other projects (numpy, scipy, matplotlib, ...)

- "Documentation": issues or pull request related to the documentation. Issues open with the documentation template are automatically labeled. Sometimes "Bug Triage" issues are documentation issues and need to be manually relabeled after discussion. Pull requests opened with titles containing [DOC] are automatically labeled as Documentation.

- "Build / CI": issues or pull requests related to the build of the library or the continuous integration process. The label is automatically assigned via a github action when the title contains [CI].

- "pypy": issues or pull requests related to pypy python implementation.

- Architecture labels: eg "arch:arm"... perhaps this could be generalized (adding 32 and 64 bit?). It identifies the architecture of the system where the issue has been experienced.

- "Large Scale": issues and pull requests related to large datasets. This kind of issues are difficult to reproduce, the contributor willing to help should have access to significant computing resources.

- "Breaking Change": Issue resolution would not be easily handled by the usual deprecation cycle. Those issues and pull requests will be allowed only in major release changes. It is very likely to correspond to an "API Change" entry in the Changelog.

- Type of change: those labels are useful to track the type of modification applied and will help the contributor in the creation of the corresponding entry in the change log.
  - "New Feature": feature request issues are automatically labeled as "New Feature", pull request labeled as "New Feature" are meant to be added to the change log as "Major Feature" or "Feature".
  - "Performance": performance issues and correspondent pull requests meant to solve them. It should be applied manually and merged pull request should be referred in the change log as "Efficiency" entries.

  - "Enhancement": miscellaneous improvements corresponding to an "Enhancement" entry in the change log.
  - "Bug": confirmed bug (something not working as expected), pull requests labeled as "Bug" should be referred in the change log as a "Fix". Also, sometime bugs can be detailed with the labels:
    * "segfault": the bug produces a segmentation fault, ie memory access violation.
    * "Regression": the bug has been introduced recently and was not present in an identified previous version.
  - "Bug triage": this is the default label for issues open as bug, meaning that they need confirmation from the team. Ideally this label should be modified **before** a pull request is proposed.
  - "Question": usage or general questions, good candidates for being moved to discussions

- Difficulty level: in principle those labels do not suit to pull requests. They are useful to give to the contributor an idea of the requested knowledge to solve the issue.

  - "Easy": easy, well-defined issues, basic python knowledge requested, very well documented and straightforward way to solve them. Ideally, clear instructions should be given in the issue to help contributors. These can often be tagged as "good first issues". Before tagging an issue as easy, remember that "easy" doesn't mean the same thing for a casual contributor and for a maintainer: in doubt, be conservative and use "moderate".
  - "Moderate": Moderate level of difficulty, for anything that requires some knowledge of our conventions and practices (e.g. deprecations, API conventions, etc.). Most issues will fall under the moderate category.
  - "Hard": Hard level of difficulty, for issues that span a multiple areas or require expert knowledge.

- Need a contributor, community involvement: some issues are particularly suited to improve community engagement.

  - "help wanted": external contributors are welcome to propose a pull request. Before assigning this label it is strongly suggested to ask the users that opened the issue if they want to propose a pull request.
  - "good first issue": working on this kind of issue gives useful insights on the library and their solution is quite straightforward and well documented
  - "sprint": when a [[sprint | Upcoming-events]] is planned some issues are reserved to be tackled at the sprint

- Criticality level: those labels are quite difficult to assign.

  - "Blocker": a bug or a feature that could block a release, should be fixed as soon as possible.

  - "High Priority": high interesting feature or critical bug.
  - "Low Priority": it could correspond to a "Won't fix" in other projects.

- General discussions: those labels define general discussions among maintainers which can lead to open more specific issues.
  - "API": discussions about the scikit-learn application programming interface.
  - "RFC": request for comments about general propositions.

- Workflow labels:
  - Expected actions from maintainers:
    * "Waiting for Reviewer": a pull request is awaiting for review since quite some time (~1 month?).
    * "Needs Decision": there is no consensus among maintainers. They are expected to take a decision at some point.
  - Expected actions from contributors or issue reporters:
    * "Needs Response": the contributor or reporter is asked for clarifications since quite some time (~1 month?).
    * "Needs Benchmarks": the contributor is asked for benchmarking the proposed pull request.
    * "Needs Work": the contributor is asked for consistent reworking of the pull request.
  - "Stalled": the contributor is no longer responding, the pull request is stalling.
  - "Superseded": the pull request has been superseded by another one.
  - "No Changelog Needed": allows the changelog to be skipped for pull requests that do not need an entry in the what's new file.