Contributing

This project welcomes and encourages all forms of contributions, including but not limited to:

- Pushing patches.
- Code review of pull requests.
- Documentation, examples and test cases.
- Readability improvement, e.g., improvement on docstr and comments.
- Community participation in issues, discussions, and discord.
- Tutorials, blog posts, talks that promote the project.
- Sharing application scenarios and/or related research.

You can take a look at the Roadmap for Upcoming Features to identify potential things to work on.

Most contributions require you to agree to a Contributor License Agreement (CLA) declaring that you have the right to, and actually do, grant us the rights to use your contribution. For details, visit https://cla.opensource.microsoft.com.

If you are new to GitHub here is a detailed help source on getting involved with development on GitHub.

When you submit a pull request, a CLA bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately (e.g., status check, comment). Simply follow the instructions provided by the bot. You will only need to do this once across all repos using our CLA.

This project has adopted the Microsoft Open Source Code of Conduct. For more information see the Code of Conduct FAQ or contact opencode@microsoft.com with any additional questions or comments.

How to make a good bug report

When you submit an issue to GitHub, please do your best to follow these guidelines! This will make it a lot easier to provide you with good feedback:

- The ideal bug report contains a short reproducible code snippet. This way anyone can try to reproduce the bug easily (see this for more details). If your snippet is longer than around 50 lines, please link to a gist or a GitHub repo.

- If an exception is raised, please **provide the full traceback**.

- Please include your **operating system type and version number**, as well as your **Python, flaml, scikit-learn versions**. The version of flaml can be found by running the following code snippet:

```
import flaml
print(flaml.__version__)
```

- Please ensure all **code snippets and error messages are formatted in appropriate code blocks**. See Creating and highlighting code blocks for more details.

Becoming a Reviewer

There is currently no formal reviewer solicitation process. Current reviewers identify reviewers from active contributors. If you are willing to become a reviewer, you are welcome to let us know on discord.

Developing

Setup

```
git clone https://github.com/microsoft/FLAML.git
pip install -e FLAML[notebook,autogen]
```

In case the `pip install` command fails, try escaping the brackets such as `pip install -e FLAML\[notebook,autogen\]`.

Docker

We provide a simple Dockerfile.

```
docker build https://github.com/microsoft/FLAML.git#main -t flaml-dev
docker run -it flaml-dev
```

Develop in Remote Container

If you use vscode, you can open the FLAML folder in a Container. We have provided the configuration in devcontainer.

Pre-commit

Run `pre-commit install` to install pre-commit into your git hooks. Before you commit, run `pre-commit run` to check if you meet the pre-commit requirements. If you use Windows (without WSL) and can't commit after installing pre-commit, you can run `pre-commit uninstall` to uninstall the hook. In WSL or Linux this is supposed to work.

Coverage

Any code you commit should not decrease coverage. To run all unit tests, install the [test] option under FLAML/:

```
pip install -e."[test]"
coverage run -m pytest test
```

Then you can see the coverage report by `coverage report -m` or `coverage html`.

Documentation

To build and test documentation locally, install Node.js. For example,

```
nvm install --lts
```

Then:

```
npm install --global yarn  # skip if you use the dev container we provided
pip install pydoc-markdown==4.5.0  # skip if you use the dev container we provided
cd website
yarn install --frozen-lockfile --ignore-engines
pydoc-markdown
yarn start
```

The last command starts a local development server and opens up a browser window. Most changes are reflected live without having to restart the server.

Note: some tips in this guide are based off the contributor guide from ray, scikit-learn, or hummingbird.

Edit this page

Previous
« Tune - PyTorch

Next
Research »