How to contribute

Unless you explicitly state otherwise, any contribution intentionally submitted for inclusion in this project by you, as defined in the Apache License, shall be dual-licensed as above, without any additional terms or conditions.

1. Join slack
2. Become familiar with docs and code by reading the ramp-up guide
3. **Ask the owner of the repository to add your GitHub username to the repository** so that you can work on issues and be able to create your own branches and not needing to fork the repo
4. Pick an open issue or a task in the corresponding project for the repo that you'll be working on. You can see good for first issues that you can pick from
5. **Assign the issues you are working on to you and move them to the corresponding status column as you are progressing on them. If the taks is not an issue yet, convert it to issue first**
6. Make the changes in your branch
7. Add docs as they apply
8. Add tests, benchmarks and examples for your changes, if applicable
9. `cargo fmt --all` to format the code. You can configure your `IDE` to do this on save, RustRover and VSCode
10. `cargo clippy --all --release` and fix any errors
11. **DON'T INCREASE THE VERSION NUMBER IN `Cargo.toml`, WE WILL DO THAN WHEN RELEASING**
12. Create a `git` `commit hook` file in `.git/hooks/pre-commit` with this content on `Linux` and `MacOS`, and this on `Windows`. Make it executable in Linux and macOS with `chmod +x .git/hooks/pre-commit` .This will run when you do `git commit` and will make the commit to be quite slow, but please give it time to complete as this helps to fix any issues locally and not relying just on running `ci` on GitHub when you create the PR
13. Commit your changes and if there are any errors fix them before you push them
14. Push your changes and create a `PR` back to the `parent` repo targeting the `main` branch and request review from owners of the repository
15. Monitor the checks (GitHub actions runs) and fix the code if they are failing
16. Respond to any comments
17. **DON'T MERGE THE PR YOURSELF, LEAVE THAT TO REPOSITORY OWNERS**
18. In the end, ideally, it will be merged to `main`