

Fundamentals of Machine Learning - 2022

Practice 1 - Git & GitHub

August 16 2022

Karina Laneri, Laila D. Kazimierski, Martín Onetto, Luis G. Moyano

This practice is intended for your own exercise and **does not** need to be turned in. It assumes you have git installed and a properly configured GitHub account. You will need this for some of the exercises below.

1. Questions

1. What is the stage area? How is it different from the working directory? How is it different from the index?
2. What is the difference between the commands `init` and `clone`?
3. When you checkout a branch, what happens with those files that are not in that branch?
4. Why is the commit so important?
5. What is 'origin'? Can there be more than one? Is the name mandatory?
6. What is a '3-way merge'?
7. Why one should not rewrite history?
8. What's the difference between a hard and a soft `reset`?
9. What does the `stash` command do?

2. Exercises

2.1. Local

1. Initialize a toy repository and make a few commits in a `main` branch, adding files and modifying them (mimic your everyday workflow).

2. Create and checkout a new branch (we suggest you choose expressive names for branches!), also making a few commits there, putting new files and modifying existing files. Importantly, make sure you modify some file also present in the `main` branch. Changes in this file should include new lines and changes to lines existing in `main`, thus forcing a merge conflict.
3. Merge both branches. A conflict message should appear and, once you (manually) solve it, finish merging.
4. Inspect the log and see if you can identify the different changes you made (use `diffs`!), as if you were inspecting someone else's repo history.

2.2. Remote

1. Clone this course's repo to your local drive. Compare what you see local vs. what you see remote (via the `status` command). Try committing some changes and repeat `status`, to see the difference.
2. Create a new repo in your GitHub account, choose whatever name, it'll be just a sandbox.
 - a) Clone the repo locally.
 - b) Make a few commits and push everything to Github.
 - c) Now go to Github and make a change there (e.g., upload a file or edit something). Now the remote repo differs from the local repo.
 - d) Add a new file (without committing this change) in your local repo. Now pull from the remote. Why the error?
 - e) Now commit locally, and pull again. Why no error?

2.3. Distributed

In pairs, one of you creates a sandbox repo in GitHub (or uses a previous one), and both clone this repo locally. The repo should have the appropriate permissions so both can push (pull) to (from) it.

1. One of you pushes a file to the repo.
2. The other pulls this file from the repo.
3. Now both change (locally) the same line in the same file. One of you pushes the changed file to the repo.
4. Afterwards, the other one pulls from the repo, resolving any conflict and pushing it again.
5. Repeat these steps changing roles.