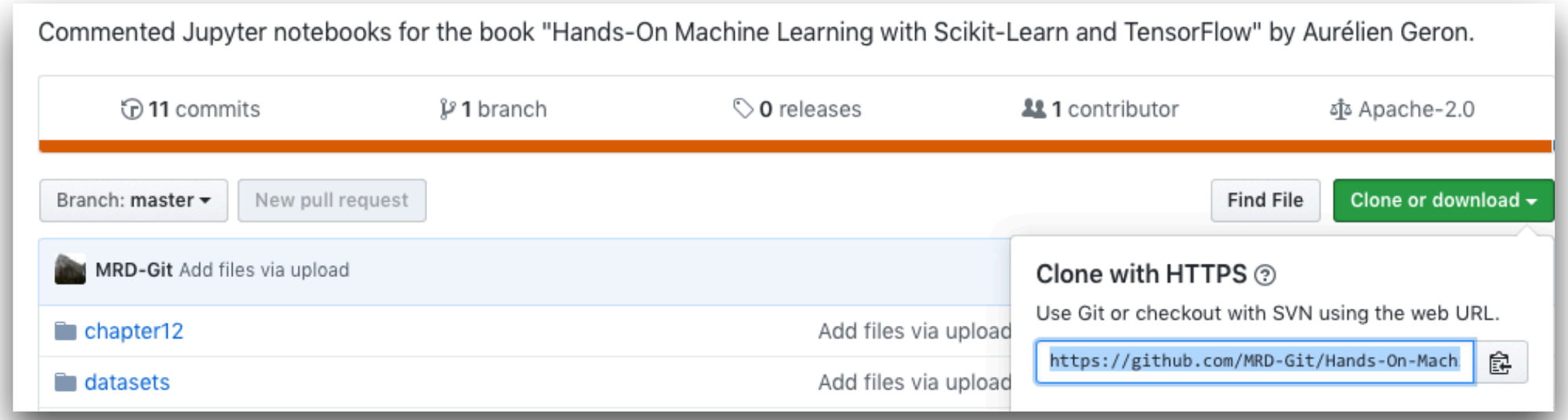
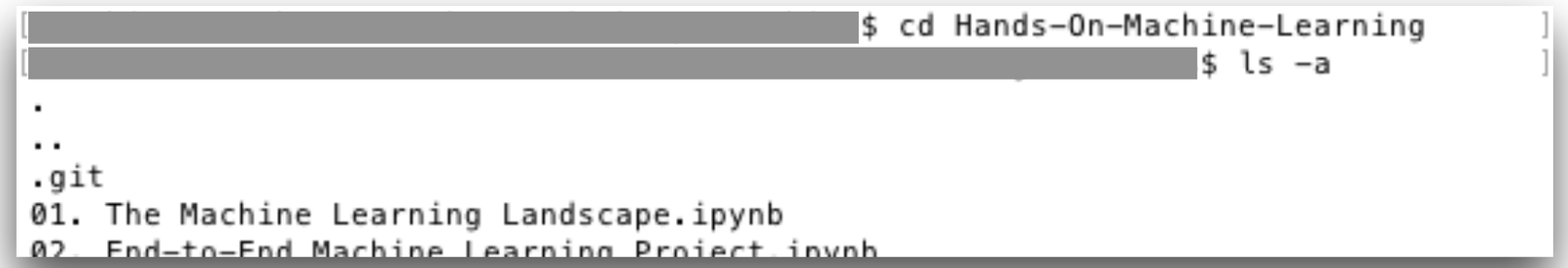
Cloning and pushing repositories from/to GitHub using *git*

0. If git is not installed on your machine, get it from here: https://git-scm.com.
   The version of git can be checked by running *git --version* in the terminal

1. Cloning a remote repository ("repo") from GitHub on a local machine (see also Ref. [1]):
   # copy the repo URL as shown below
   # open a terminal, navigate to the folder into which the repo shall be cloned, and run *git clone <repo URL>* in the terminal (here, one would run *git clone https://github.com/MRD-Git/Hands-On-Machine-Learning.git*)



2. Initialize the downloaded folder as a local git repo (if necessary)
   # change the terminal prompt into the cloned / downloaded folder
   # run *ls -a* to show all (including hidden) folder contents; if there is a ".git" folder, then the folder is already a git repo; if there is no ".git" folder, create it by running *git init* in the terminal (while in the cloned / downloaded folder)



3. Get familiar with git by going through the git tutorial here:
   https://git-scm.com/docs/gittutorial.
   important commands:
   # *git init* (to initialize the folder as a git repo, see step 2. on previous page)
   # *git add <file>* (add a file to the next commit)
   # *git add .* (add all files to the next commit)
   # *git diff --cached* (see what would be commited)
   # *git status* (check the status of the next commit)
   # *git commit -a* (perform the commit; *-a* automatically includes all changed files; as
   for the commit message: press *i* to enter input mode, *esc* to exit it,
   and then enter *.wq!* to return to the command line; see also Ref. [2])
   # *git log* (show history or repo)
   # *git branch <new_branch>* (create new branch “new_branch” and switch to it)
   # *git branch* (show all branches; the current one is indicated by an asterisk)
   # *git switch <branch_name>* (switch to branch “branch_name”)
   # *git merge <branch_name>* (merge branch “branch_name” into the current branch)
   # *git diff* (show the difference between the current branch and the master branch)
   # *git branch -d <branch_name>* (will delete branch “branch_name” only if all changes
   have already been included in the current branch)
   # *git branch -D <junk_branch>* (delete branch “junk_branch” no matter what)

4. Make changes to the files (outside the terminal). The changes are automatically logged in the current branch.

5. When done, commit the changes (see step 3).

6. Using remote repos:
   # run *git remote -v* to check for available remote repos (see also Ref. [3])

```
[redacted]$ git remote -v
github  https://github.com/MRD-Git/Hands-On-Machine-Learning.git (fetch)
github  https://github.com/MRD-Git/Hands-On-Machine-Learning.git (push)
```

   # if necessary, run *git remote add <remote_name> <repo_URL>* to add the github repo with URL “repo_URL” (see step 1) under the name “remote_name” to the remote repos

7. Pushing the modified repo back to GitHub:
   # push the branch with name “branch_name” of the local repo to the remote repo with name “remote_name” by running *git push <remote_name> <branch_name>* (see also Ref. [4])

```
[redacted]$ git push github
master
To https://github.com/MRD-Git/Hands-On-Machine-Learning.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/MRD-Git/Hands-On-Machine-
Learning.git'
```

if an error message pops up (as above), one can try to force the push via the *-f* option

```
[redacted]$ git push -f github master
Enumerating objects: 6811, done.
Counting objects: 100% (6811/6811), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6804/6804), done.
Writing objects: 100% (6811/6811), 268.81 MiB | 100.00 KiB/s, done.
Total 6811 (delta 490), reused 0 (delta 0)
remote: Resolving deltas: 100% (490/490), done.
To https://github.com/MRD-Git/Hands-On-Machine-Learning.git
```

References:

[1] https://www.youtube.com/watch?v=wZFAmxEgT3I (on cloning a repo from GitHub)

[2] https://stackoverflow.com/questions/13239368/git-how-to-close-commit-editor/28343418 (on terminal commands for commit messages)

Press i to enter inline insert mode. Type the description at the very top, press esc to **exit** insert mode, then type :x! (now the cursor is at the bottom) and hit enter to **save and exit**. After writing **commit message**, just press Esc Button and then write :wq or :wq! and then Enter to close the unix file. Feb 5, 2015

Git - How to close commit editor? - Stack Overflow

https://stackoverflow.com › questions › git-how-to-close-commit-editor

[3] https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes (on remotes)

[4] https://stackoverflow.com/questions/17291995/push-existing-project-into-github and https://www.youtube.com/watch?v=ibNqauPoicg (on pushing a repo to GitHub)