A Reproducible Research Pipeline

Using Git and Data Version Control (dvc)

Lukas Erhard

2025-02-19

University of Stuttgart, CSS Lab

The Big Problem

How can we work on the same research project collaboratively?

Working on research together has major advantages

- The research is faster
- The code has less bugs
- It keeps the research reproducible

What do we need to work together?

1. Same code
2. Same data
3. Same environment
4. Code needs to be run in the same order

Problem 1

Having the same code

This is what git is for, so we can skip this issue...

Knowledge requirement check

- Do we all know what a `.gitignore` file is?

Problem 2

Having the same data

aka “oh no, my data is too big for git”

What is dvc?

- Tool (written in Python) that augments the functionality of git
- DVC stands for: Data Version Control
- provides the commmand: dvc

command: dvc add [path/to/file]

- adds any file or folder to a `.gitignore` file
- creates a `.dvc` file instead which is still tracked by git
  - contains the md5-hash of the original file
  - is used to keep dvc and git in sync
- moves the file to `.dvc/cache/` and links to it from its original position

command: dvc commit [path/to/file]

- if a file changes and you want to add the changes, run dvc commit to update it

command: dvc push

- pushes all dvc tracked data to a specified remote
- There are many backends available, we have our own

Consequences of this approach

The workflow for using git + dvc changes to:

1. dvc add / commit
2. git add
3. git commit
4. dvc push
5. git push

Demo Time: Introducing MinIO S3 Storage

Problem 3

Same environment

Problem 4

Ensure the order of execution

aka “Why are my results from today different from yesterday?”

Using the dvc DAG

TODO: Explain dvc.yaml and the DAG