ADALTAS

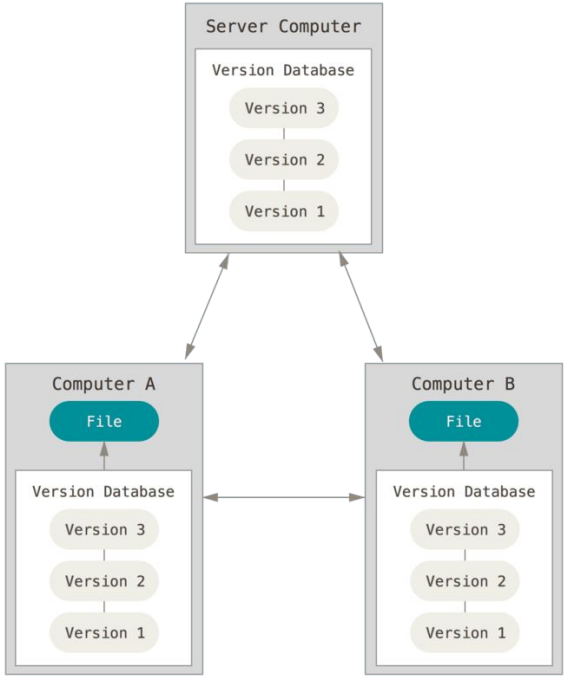
2. Versioning

2.1 Git refresher

Git refresher

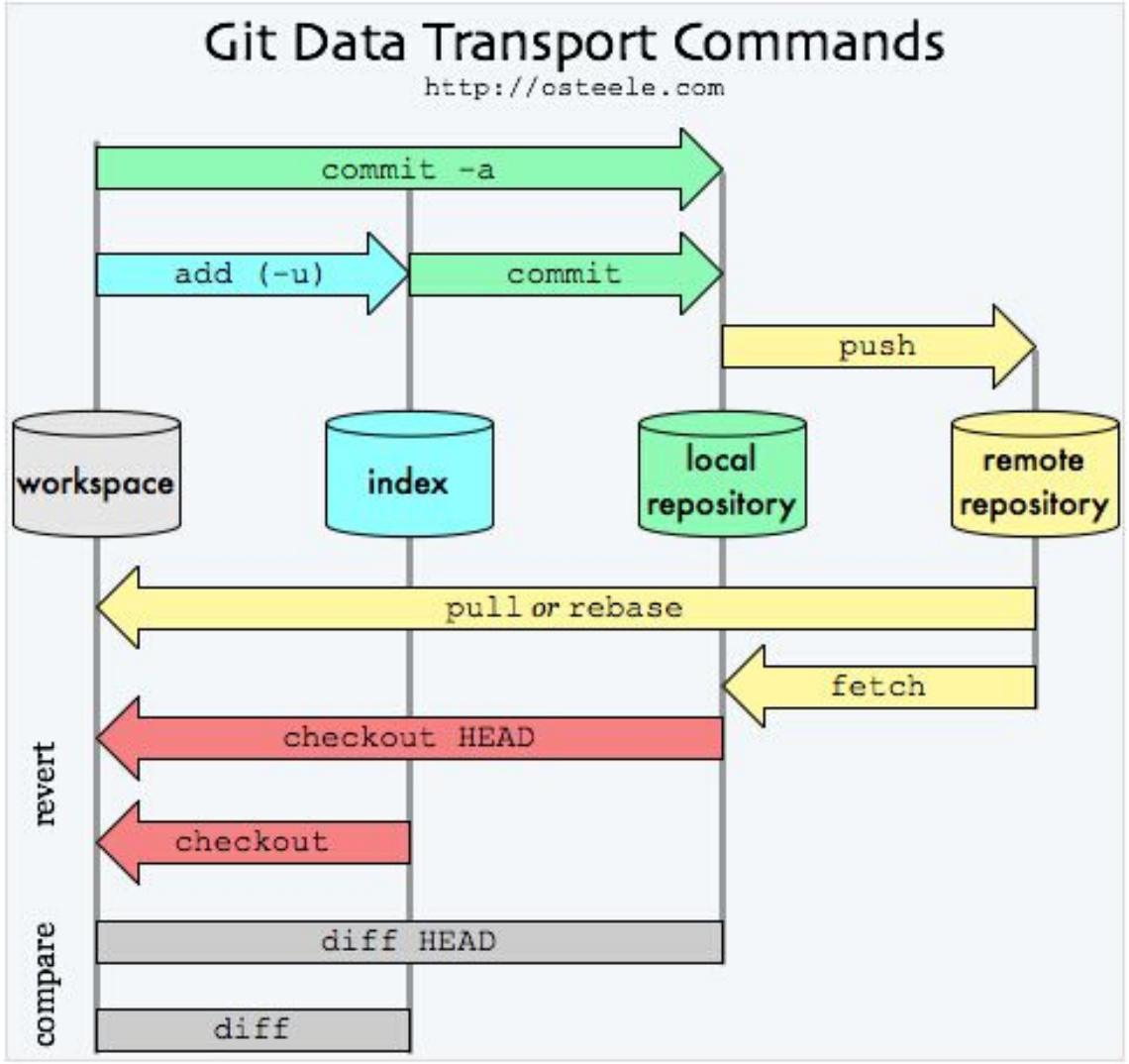
ADALTAS

- **Git:** Decentralized Version Control System



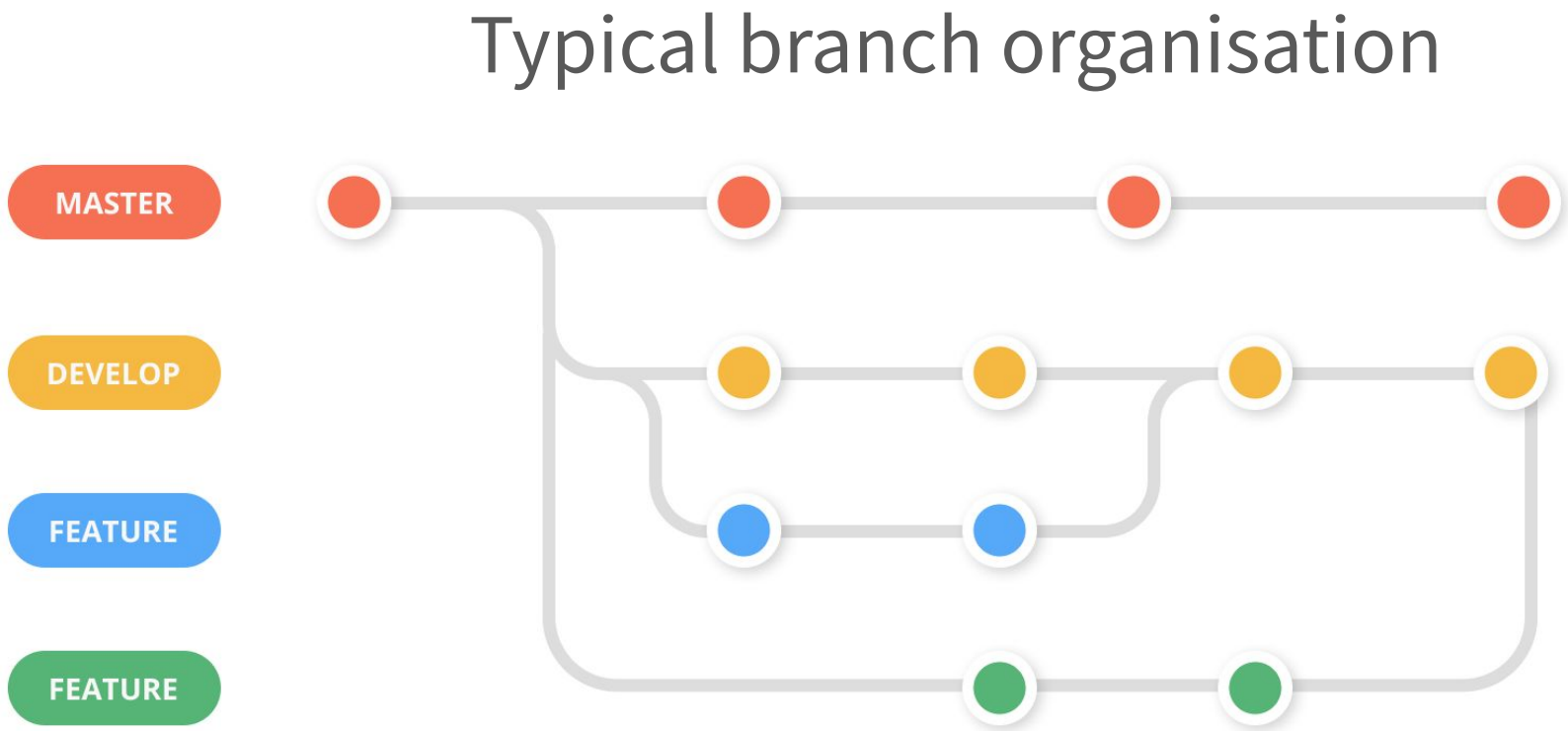
- **Git repository:**
  - a directory of versioned files => saves the history of all changes
  - it exists as a local copy with “.git” at the root of the project
  - the user can decide which files to version (.gitignore file)
  - providers: GitHub, GitLab; self-managed

Data flow


Git Data Transport Commands
http://osteele.com
commit -a
add (-u)
commit
push
workspace
index
local repository
remote repository
pull or rebase
fetch
checkout HEAD
checkout
diff HEAD
diff
revert
compare

Git workflow

ADALTAS



https://zepel.io/blog/5-git-workflows-to-improve-development/

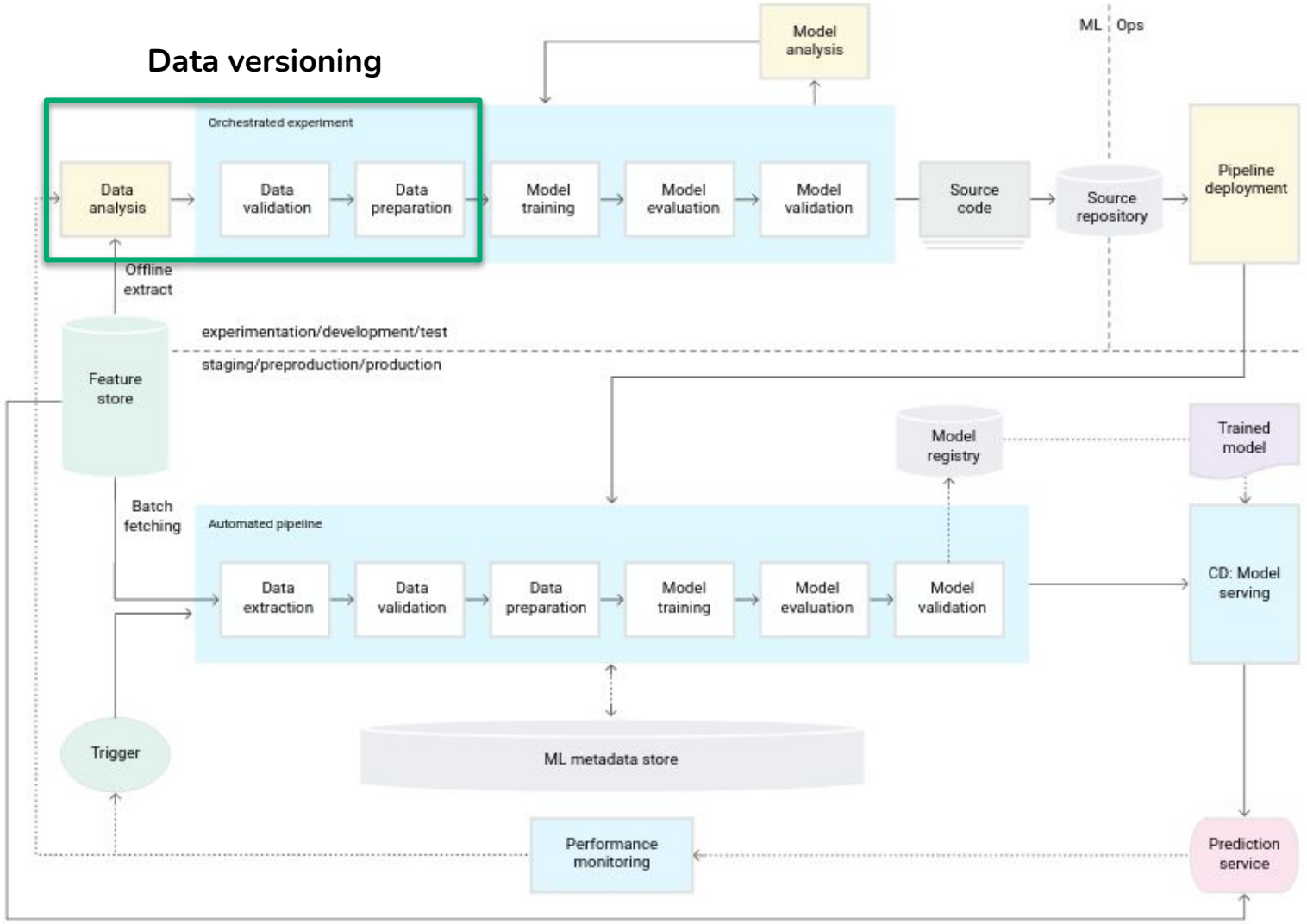
ADALTAS

2.2 Data versioning

git

Roadmap

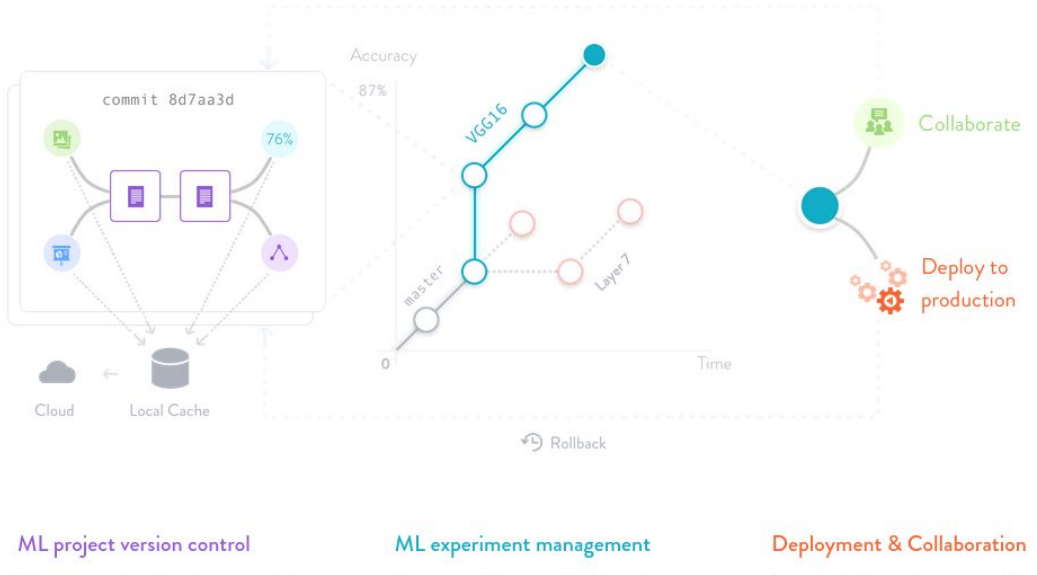
ADALTAS



Problem definition

What do we need to track in Data Science project to be able to reproduce the results?

- Dataset
  - Tabular
  - Images, video, sound, text
- Code
  - Functionality, Hyper-parameters
  - Data pre-processing, Modeling
- Results
  - Numeric (metrics)
  - Plots
- Environment
  - Dependencies



Data versioning use cases

ADALTAS

Data beyond Data Science

- **Data engineering** during data cleaning and dataset preparation
- Test data in **software engineering** to assure the functionality of the software
- Development and testing of **database applications**
- Ontology versioning for **Semantic Web**
- ...

Data is in the heart of any ML project

ADALTAS

Why is the exact version of a dataset important?

- Data quality management
- Reproducible training and re-training
- Automation of testing or deployment
- Use in applications and web services
- Audit

Classical DV solutions and their downsides

ADALTAS

Classical Software Engineering tools are not enough to solve ML reproducibility crisis

- Git
  - Not suitable for big files
  - Difficulties with binary files
- Git-LFS
  - File size limitation (ex.: 2 GB for GitHub free account and 5 GB for GitHub Enterprise Cloud)
  - Data needs to be stored on servers of Git provider
- External storage
  - Checksum calculation for any change in the dataset and placing them under version control

The exact version of the dataset can be extracted in through the api

ADALTAS

Like that we always know which data we used for training

&

- solves Git's limitations
- easy to learn

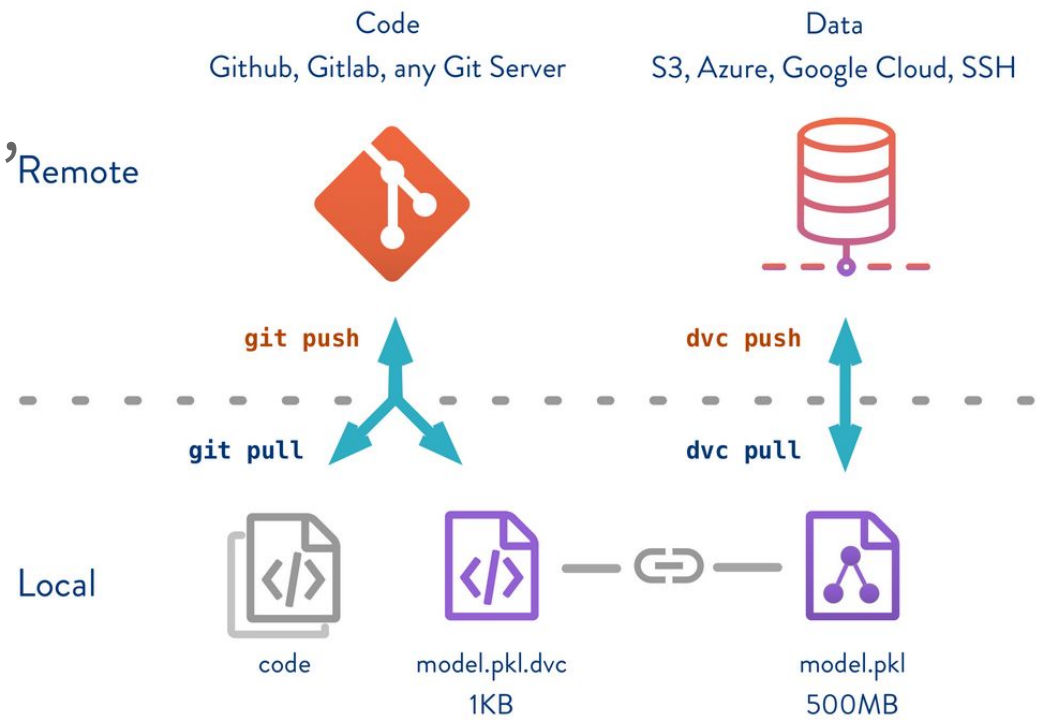
- use dvc api

What is DVC?

https://dvc.org/

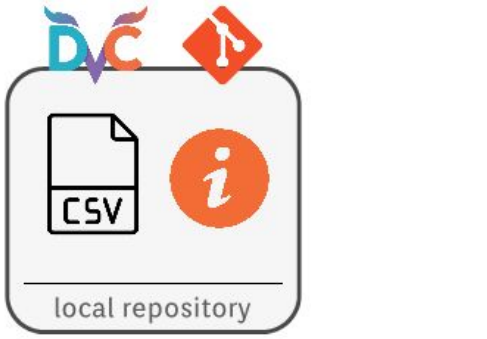
- Tracks datasets and ML projects
- Works with many types of storages (cloud, local, HDFS, HTTP…)
- Runs on top of a Git repository
- Supports building and running pipelines

- We will focus on **dataset tracking**.

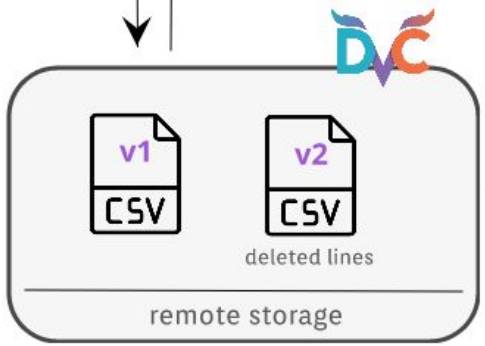


Code and data flows in DVC

Let’s put it in action!



MLflow git repo
https://github.com/mlflow/mlflow
- data: wine-quality.csv
- code: train.py



dvc.api

```python
import dvc.api

resource_url = dvc.api.get_url(
    'data/raw/records.csv',
    repo='https://github.com/aisg/dataset-registry'
    )

print(resource_url)
```