02476 Machine Learning Operations

Nicki Skafte Detlefsen

Reproducibility and software

What is in this presentation?

♻ The reproducibility crisis

- What is it?
- How bad is it?

💻 How can software help?

- What to use when?

```
// Dear programmer:
// When I wrote this code, only god and
// I knew how it worked.
// Now, only god knows it!
//
// Therefore, if you are trying to optimize
// this routine and it fails (most surely),
// please increase this counter as a
// warning for the next person:
//
// total_hours_wasted_here = 254
//

```

What is reproducibility?

💡 Reproducibility is the ability of **an entire experiment** or study to be duplicated, either by the same researcher or **by someone else working independently**.

💡 Reproducible data - **repeatability** which is the degree of agreement of tests or measurements on replicate specimens by the same observer in the same laboratory.

💡 Computationally reproducible research - the idea that the ultimate product of **academic research** is the paper along with the **full computational environment** used to produce the results in the paper such as the code, data, etc. that can be used to reproduce the results and create new work based on the research.

Why do we need it?

I would argue, because of this

What does reproducibility have to do with MLOps?

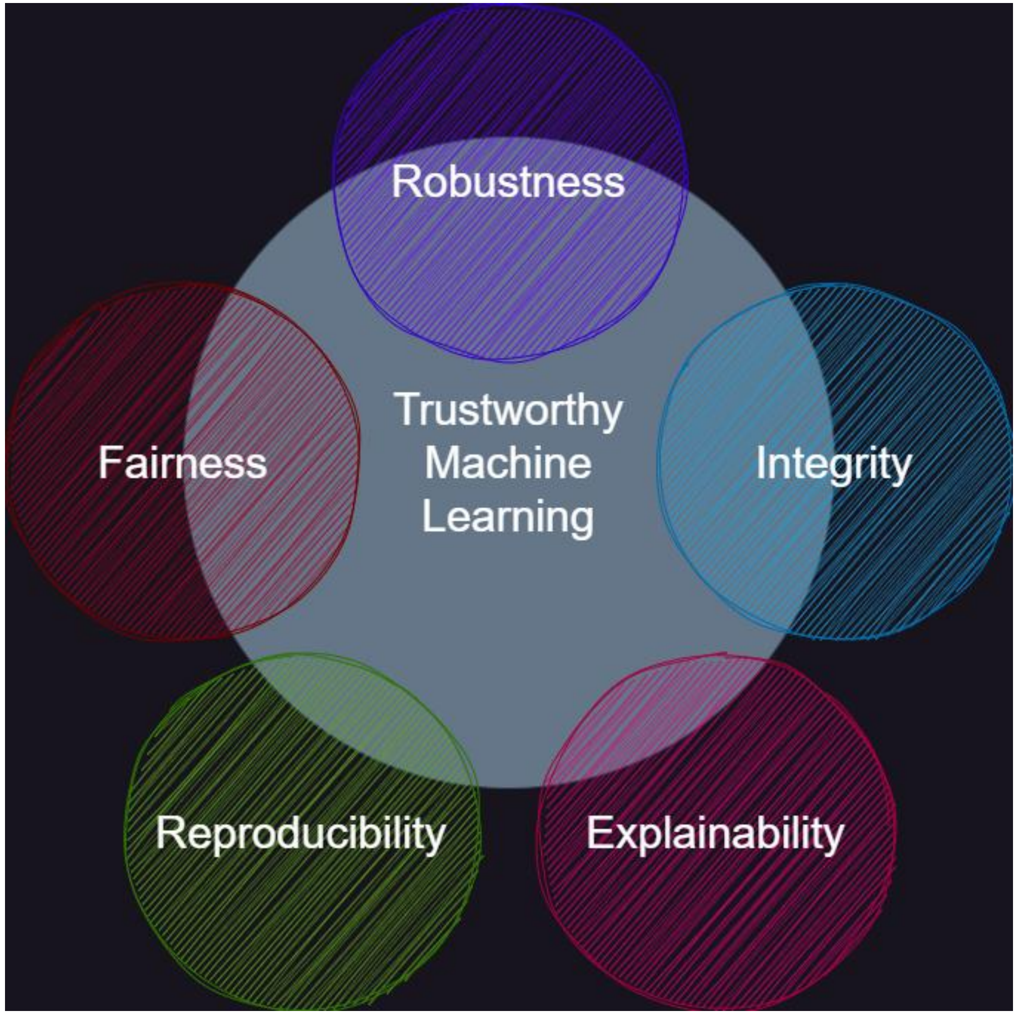
- Knowledge Preservation
  - If other cannot reproduce your work, knowledge can be lost
- Transparency and Accountability
  - To make sure that others can verify your claims before going into production
- Regulatory Compliance
  - To secure the correct documentation to make sure everything is compliant
- Continuous Improvement
  - Making sure that improvements to the pipeline are real and not artifacts of random effects

Trustworthy ML

Reproducibility is a key component in *Trustworthy ML*

⚠ Case:

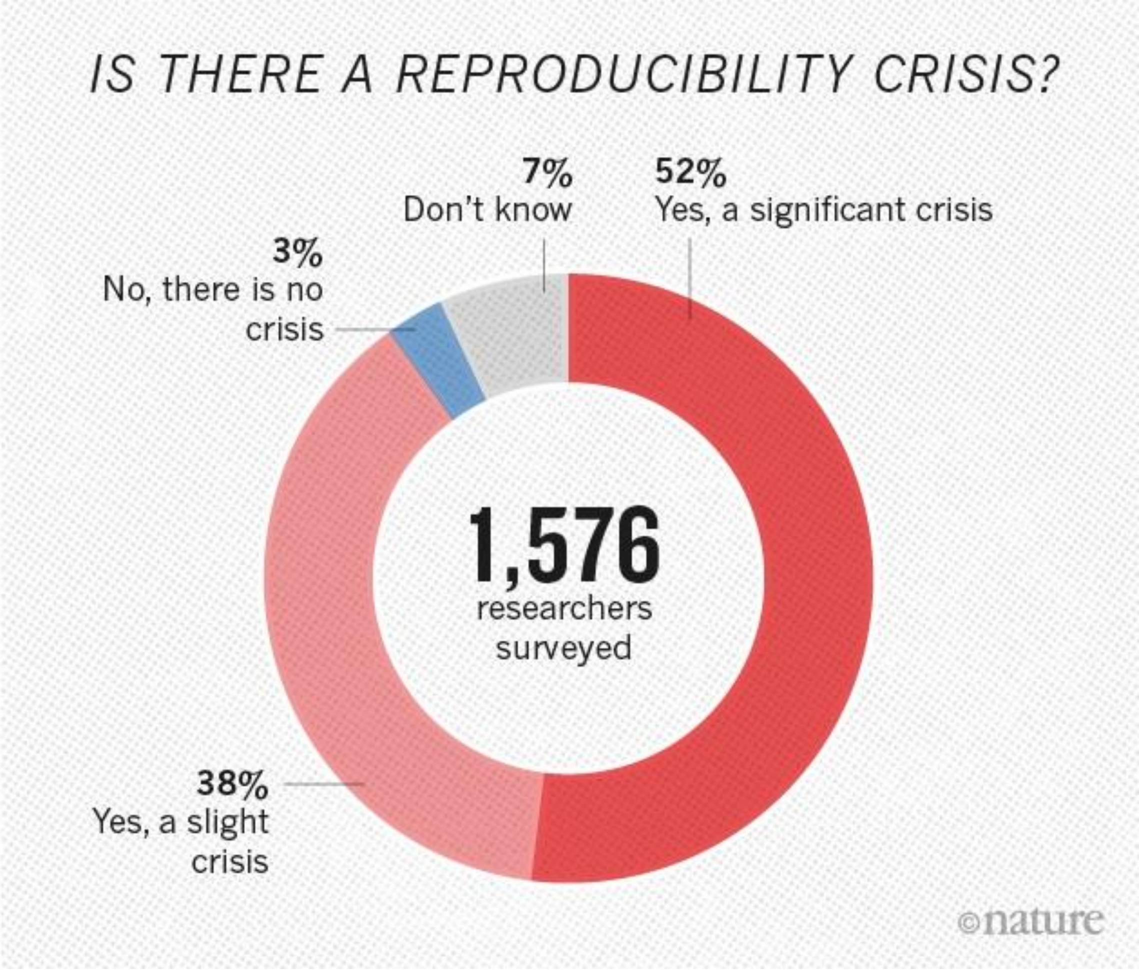
Imaging an AI agent used for diagnostics. Without reproducibility two persons with the exact same symptoms could get different diagnosis



We are in a crisis!

There is growing alarm about results that cannot be reproduced. Explanations include

- Increased levels of scrutiny
- Complexity of experiments and statistics
- Pressures on researchers



An example: Trouble in the lab

The biotech company Amgen had a team of about 100 scientists trying to reproduce the findings of 53 “landmark” articles in cancer research published by reputable labs in top journals.
Only 6 of the 53 studies were reproduced (about 10%).

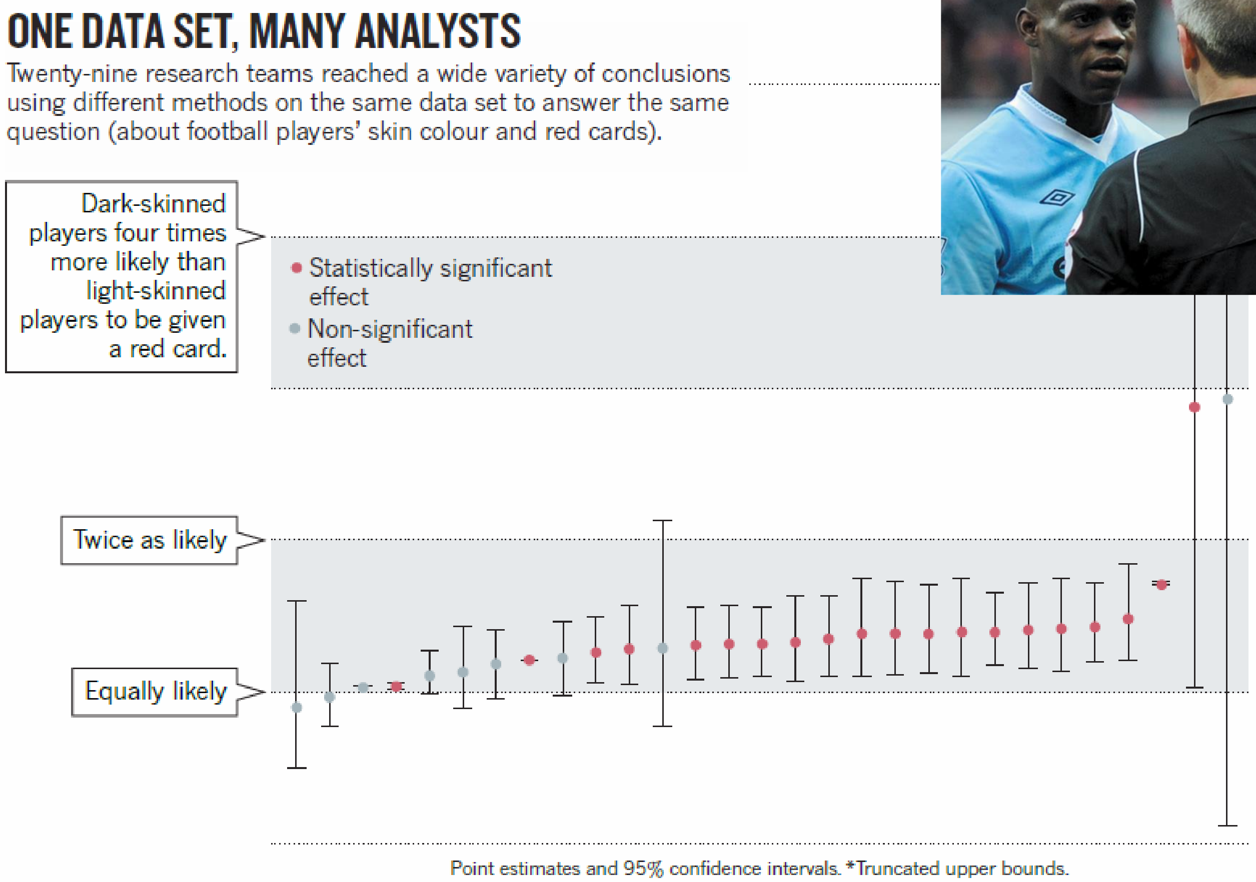
REPRODUCIBILITY OF RESEARCH FINDINGS
Preclinical research generates many secondary publications, even when results cannot be reproduced.

Journal impact factor	Number of articles	Mean number of citations of non-reproduced articles*	Mean number of citations of reproduced articles
>20	21	248 (range 3–800)	231 (range 82–519)
5–19	32	169 (range 6–1,909)	13 (range 3–24)

Results from ten-year retrospective analysis of experiments performed prospectively. The term ‘non-reproduced’ was assigned on the basis of findings not being sufficiently robust to drive a drug-development programme.
*Source of citations: Google Scholar, May 2011.

Another example: We are only humans

💭 The (subjective) choices we take during research may impact the conclusion



DTU

What are we trying to do within research?

https://paperswithcode.com/

ML Reproducibility Challenge 2021 Spring

RC2021Spring

Online Jul 20 2021 https://paperswithcode.com/rc2020 reproducibility.challenge@gmail.com

Please see the venue website for more information.
Submission Start: Apr 01 2021 12:00AM UTC-0, End: TBD UTC-0

Add: ML Reproducibility Challenge 2021 Spring Submission

All Papers | Claimed

Search by paper title and metadata | Conference: All Conferences

ToTTo: A Controlled Table-To-Text Generation Dataset
Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, Dipanjan Das
24 Nov 2020 EMNLP 2020 Readers: Everyone 0 Replies

Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!
Suzanna Sia, Ayush Dalmia, Sabrina J. Mielke
24 Nov 2020 EMNLP 2020 Readers: Everyone 0 Replies

Towards Interpreting BERT for Reading Comprehension Based QA
Sahana Ramnath, Preksha Nema, Deep Sahni, Mitesh M. Khapra
24 Nov 2020 EMNLP 2020 Readers: Everyone 1 Reply

Dialogue Response Ranking Training with Large-Scale Human Feedback Data
Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, Bill Dolan
24 Nov 2020 EMNLP 2020 Readers: Everyone 0 Replies

Data Rejuvenation: Exploiting Inactive Training Examples for Neural Machine Translation
Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael R. Lyu, Zhaopeng Tu
24 Nov 2020 EMNLP 2020 Readers: Everyone 0 Replies

Checklist for conferences

3. If you ran experiments...

- (a) Did you include the code, data, and instructions needed to **reproduce** the main experimental results (either in the supplemental material or as a URL)?
  - The instructions should contain the exact command and environment needed to run to reproduce the results.
  - Please see the NeurIPS code and data submission guidelines for more details.
  - Main experimental results include your new method and baselines. You should try to capture as many of the minor experiments in the paper as possible. If a subset of experiments are reproducible, you should state which ones are.
  - While we encourage release of code and data, we understand that this might not be possible, so "no because the code is proprietary" is an acceptable answer.
  - At submission time, to preserve anonymity, remember to release anonymized versions.
- (b) Did you specify all the **training details** (e.g., data splits, hyperparameters, how they were chosen)?
  - The full details can be provided with the code, but the important details should be in the main paper.
- (c) Did you report **error bars** (e.g., with respect to the random seed after running experiments multiple times)?
  - Answer "yes" if you report error bars, confidence intervals, or statistical significance tests for your main experiments.
- (d) Did you include the amount of **compute** and the type of **resources** used (e.g., type of GPUs, internal cluster, or cloud provider)?
  - Ideally, you would provide the compute required for each of the individual experimental runs as well as the total compute.
  - Note that your full research project might have required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper). The total compute used may be harder to characterize, but if you can do that, that would be even better.
  - You are also encouraged to use a CO2 emissions tracker and provide that information. See, for example, the experiment impact tracker (Henderson et al.), the ML CO2 impact calculator (Lacoste et al.), and CodeCarbon.

A closer look at machine learning

Re-Implementation of 255 paper. Hypothesis testing on what “paper features” have an effect on reproducibility.

A Step Toward Quantifying Independently Reproducible Machine Learning Research

Edward Raff
Booz Allen Hamilton
raff_edward@bah.com
University of Maryland, Baltimore County
raff.edward@umbc.edu

Abstract

What makes a paper independently reproducible? Debates on reproducibility center around intuition or assumptions but lack empirical results. Our field focuses on releasing code, which is important, but is not sufficient for determining reproducibility. We take the first step toward a quantifiable answer by manually attempting to implement 255 papers published from 1984 until 2017, recording features of each paper, and performing statistical analysis of the results. For each paper, we did not look at the authors code, if released, in order to prevent bias toward discrepancies between code and paper.

Table 1: Significance test of which paper properties impact reproducibility. Results significant at $\alpha \leq 0.05$ marked with“*”.

Feature	p-value
Year Published	0.964
Year First Attempted	0.674
Venue Type	0.631
Rigor vs Empirical*	1.55×10^{-9}
Has Appendix	0.330
Looks Intimidating	0.829
Readability*	9.68×10^{-25}
Algorithm Difficulty*	2.94×10^{-5}
Pseudo Code*	2.31×10^{-4}
Primary Topic*	7.039×10^{-4}
Exemplar Problem	0.720
Compute Specified	0.257
Hyperparameters Specified*	8.45×10^{-6}
Compute Needed*	8.75×10^{-5}
Authors Reply*	6.01×10^{-8}
Code Available	0.213
Pages	0.364
Publication Venue	0.342
Number of References	0.740
Number Equations*	0.004
Number Proofs	0.130
Number Tables*	0.010
Number Graphs/Plots	0.139
Number Other Figures	0.217
Conceptualization Figures	0.365
Number of Authors	0.497

Levels of reproducibility

Reproducibility is not binary, it's a spectrum

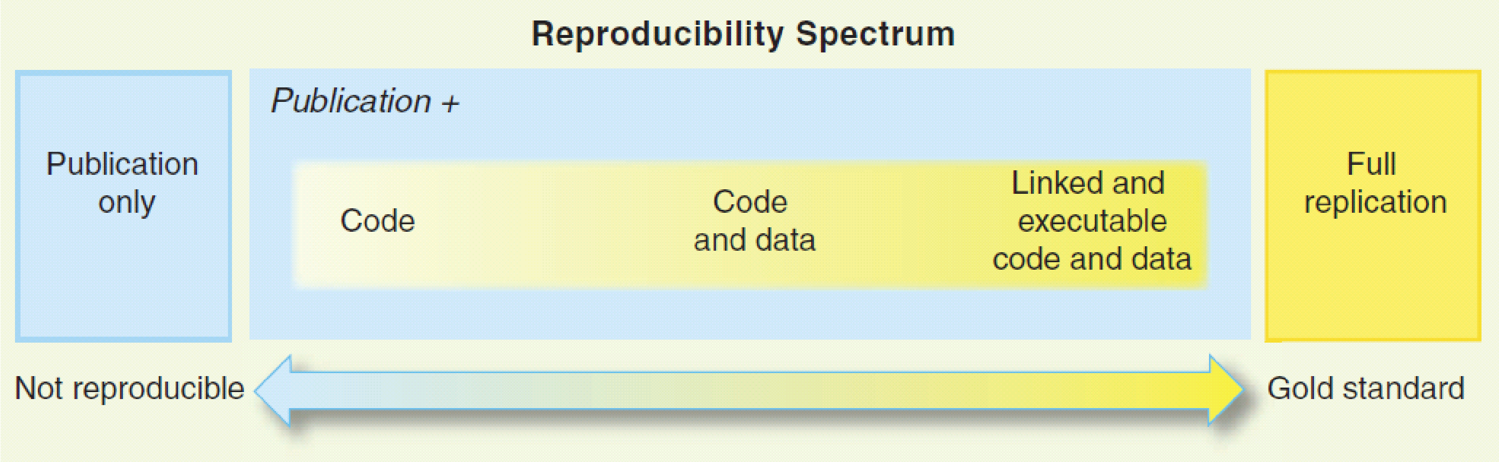


Fig. 1. The spectrum of reproducibility.

Example from ML

$$\begin{Bmatrix} w_1 \\ \vdots \\ w_n \end{Bmatrix} == \begin{Bmatrix} w'_1 \\ \vdots \\ w'_n \end{Bmatrix}$$

m1.ckpt m2.ckpt

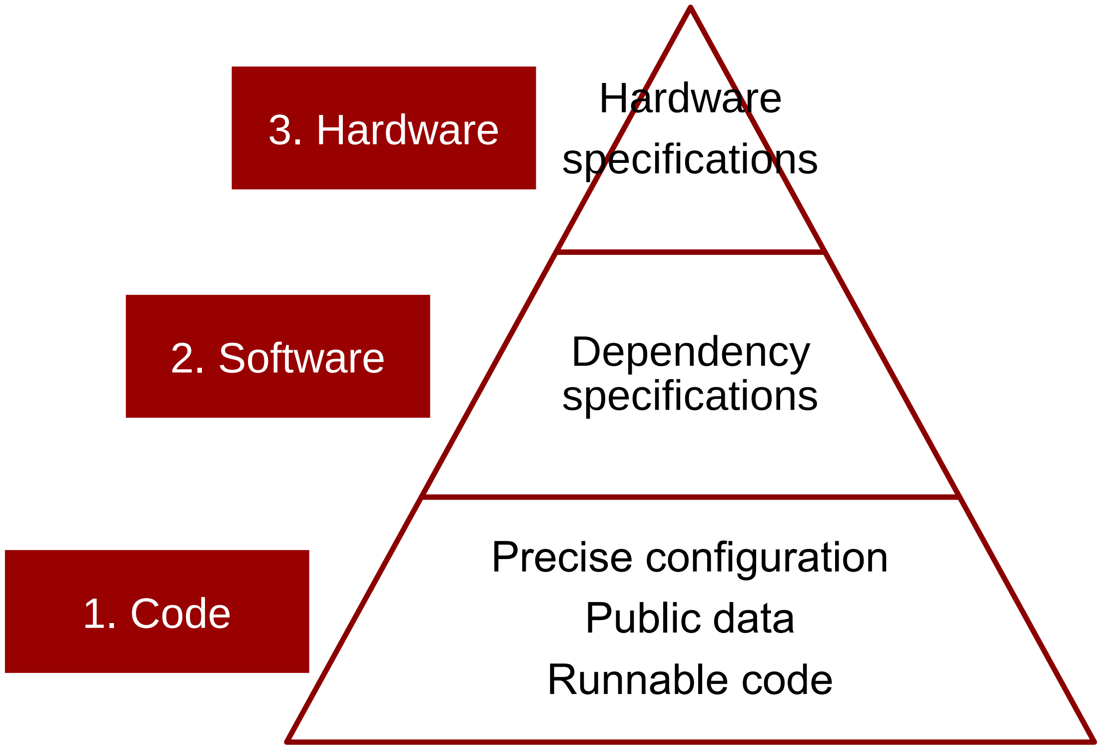
VS

Dataset	Model Architecture	Random Init	Transfer	Parameters	ImageNet Top5
Retina	Resnet-50	96.4% ± 0.05	96.7% ± 0.04	23570408	92.% ± 0.06
Retina	Inception-v3	96.6% ± 0.13	96.7% ± 0.05	22881424	93.9%
Retina	CBR-LargeT	96.2% ± 0.04	96.2% ± 0.04	8532480	77.5% ± 0.03
Retina	CBR-LargeW	95.8% ± 0.04	95.8% ± 0.05	8432128	75.1% ± 0.3
Retina	CBR-Small	95.7% ± 0.04	95.8% ± 0.01	2108672	67.6% ± 0.3
Retina	CBR-Tiny	95.8% ± 0.03	95.8% ± 0.01	1076480	73.5% ± 0.05

What can we do about it ?

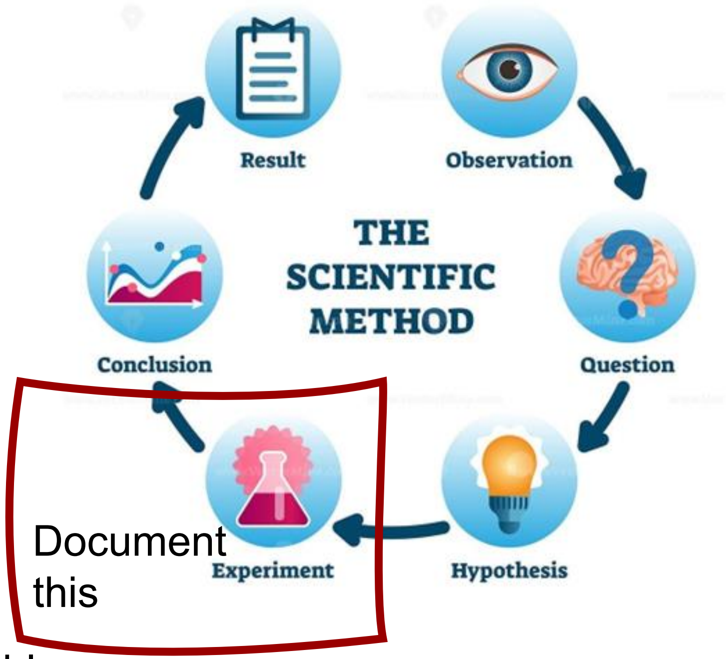
Make sure that you document everything about your experiments

Nicki’s hierarchy of reproducibility of ML



⚠ Hard to ship
☑ Highly reproducible

☑ Easy to ship
⚠ Less reproducible



Reproducibility level 1

⚡ There is a lot of subjective choices that we do when running experiments in machine learning, most notable the hyperparameters.

Parameters in scripts	Argument parser	Config files

<pre>class hparams:
    lr = 0.1
    batch_size = 16
    num_layers = 5</pre>
	<pre>python my_script.py \\
	    --lr 0.1 \\
	    --batch_size 16 \\
	    --num_layers 5</pre>
		<pre>experiment1.yaml
		
		lr: 0.001
		batch_size: 16
		num_layers: 5
		
		python my_script.py \\
		    config=experiment1.yaml</pre>

☑ Easy to code ⚠ Not easy to configure on the run ⚠ Experimental info may be lost if not careful	☑ Easy to configure ⚠ Falls on user to save the config	☑ Highly configurable ☑ Configuration is systematically saved (and version controlled)

Reproducibility level 1

Hydra is a framework for elegantly configuring complex (ML) applications

https://github.com/facebookresearch/hydra

Example:

```
├── conf
│   ├── config.yaml
│   └── dataset
│       ├── cifar10.yaml
│       └── imagenet.yaml
└── my_app.py
```

```python
import hydra
from omegaconf import DictConfig

@hydra.main(config_path="config.yaml")
def my_app(cfg: DictConfig) -> None:
    print(cfg.pretty())

if __name__ == "__main__":
    my_app()
```

Other options

- https://github.com/IDSIA/sacred
- https://mlflow.org/

Reproducibility level 2

For python: Just use a package management system

Examples:

- Conda (what I like)
- Pipenv
- venv
- pyenv

```
(lightning) C:\Users\nsde\Documents\metrics>conda env list
# conda environments:
#
base                     C:\Users\nsde\Anaconda3
ensemble                 C:\Users\nsde\Anaconda3\envs\ensemble
laplace                  C:\Users\nsde\Anaconda3\envs\laplace
lightning             *  C:\Users\nsde\Anaconda3\envs\lightning
mixerensemble            C:\Users\nsde\Anaconda3\envs\mixerensemble
mlops                    C:\Users\nsde\Anaconda3\envs\mlops
protein                  C:\Users\nsde\Anaconda3\envs\protein
pvae                     C:\Users\nsde\Anaconda3\envs\pvae
stochman                 C:\Users\nsde\Anaconda3\envs\stochman


(lightning) C:\Users\nsde\Documents\metrics>
```

```
(lightning) C:\Users\nsde\Documents\metrics>conda list
# packages in environment at C:\Users\nsde\Anaconda3\envs\lightning:
#
# Name                    Version                   Build  Channel
absl-py                   1.2.0                    pypi_0    pypi
aiohttp                   3.8.3                    pypi_0    pypi
aiosignal                 1.2.0                    pypi_0    pypi
alabaster                 0.7.12                   pypi_0    pypi
asttokens                 2.0.5              pyhd3eb1b0_0
async-timeout             4.0.2                    pypi_0    pypi
atomicwrites              1.4.1                    pypi_0    pypi
attrs                     22.1.0                   pypi_0    pypi
babel                     2.10.3                   pypi_0    pypi
backcall                  0.2.0              pyhd3eb1b0_0
beautifulsoup4            4.11.1                   pypi_0    pypi
black                     22.8.0                   pypi_0    pypi
blas                      2.116                       mkl    conda-forge
blas-devel                3.9.0              16_win64_mkl    conda-forge
bleach                    5.0.1                    pypi_0    pypi
brotlipy                  0.7.0           py38h294d835_1004    conda-forge
build                     0.8.0                    pypi_0    pypi
ca-certificates           2022.07.19           haa95532_0
cachetools                5.2.0                    pypi_0    pypi
certifi                   2022.9.14        py38haa95532_0
cffi                      1.15.1           py38hd8c33c5_0    conda-forge
cfgv                      3.3.1                    pypi_0    pypi
charset-normalizer        2.1.1              pyhd8ed1ab_0    conda-forge
check-manifest            0.48                     pypi_0    pypi
click                     8.1.3                    pypi_0    pypi
cloudpickle               2.2.0                    pypi_0    pypi
colorama                  0.4.5            py38haa95532_0
commonmark                0.9.1                    pypi_0    pypi
contourpy                 1.0.5                    pypi_0    pypi
coverage                  6.4.4                    pypi_0    pypi
cryptography              37.0.4           py38hb7941b4_0    conda-forge
cudatoolkit               11.6.0              hc0ea762_10    conda-forge
cycler                    0.11.0                   pypi_0    pypi
decorator                 5.1.1              pyhd3eb1b0_0
defusedxml                0.7.1                    pypi_0    pypi
distlib                   0.3.6                    pypi_0    pypi
docutils                  0.17.1                   pypi_0    pypi
dython                    0.7.2                    pypi_0    pypi
```

Reproducibility level 3

💡 The easiest way for someone to reproduce your work, would be to just hand over your computer.

💡 Instead of doing this, lets hand over a virtual copy of our machine

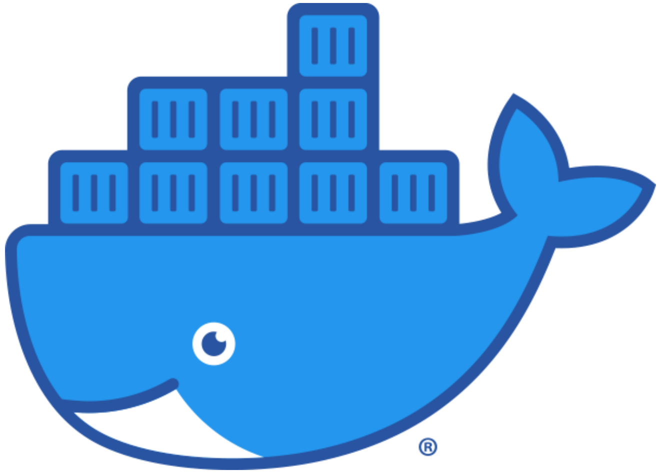
Virtual machine works by taking hardware from the host and creates virtual CPU, RAM, storage for each virtual machine. The virtual machines are completely independent from the host

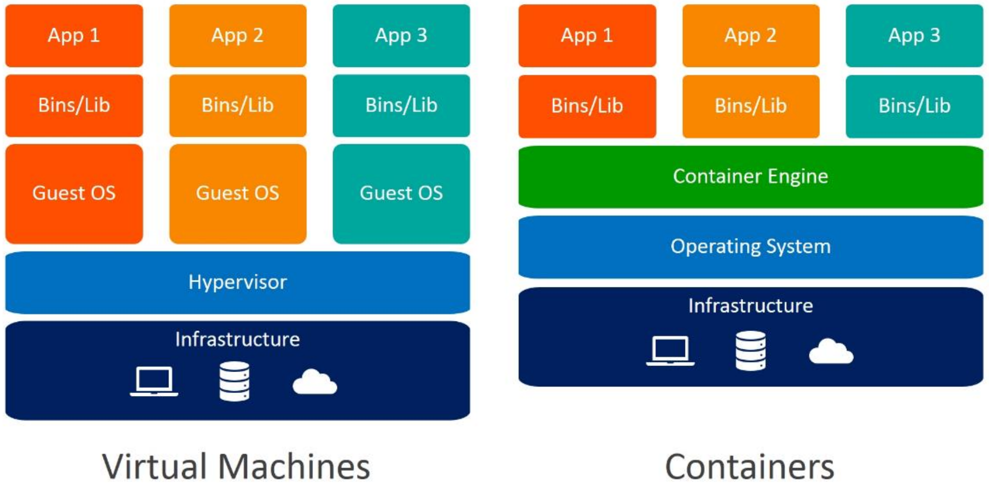
Reproducibility level 3

- The core advantage of a VM is that it in principal can run on any host without changes, because it is independent.

- Docker can be seen as an lightweight version of full VMs.

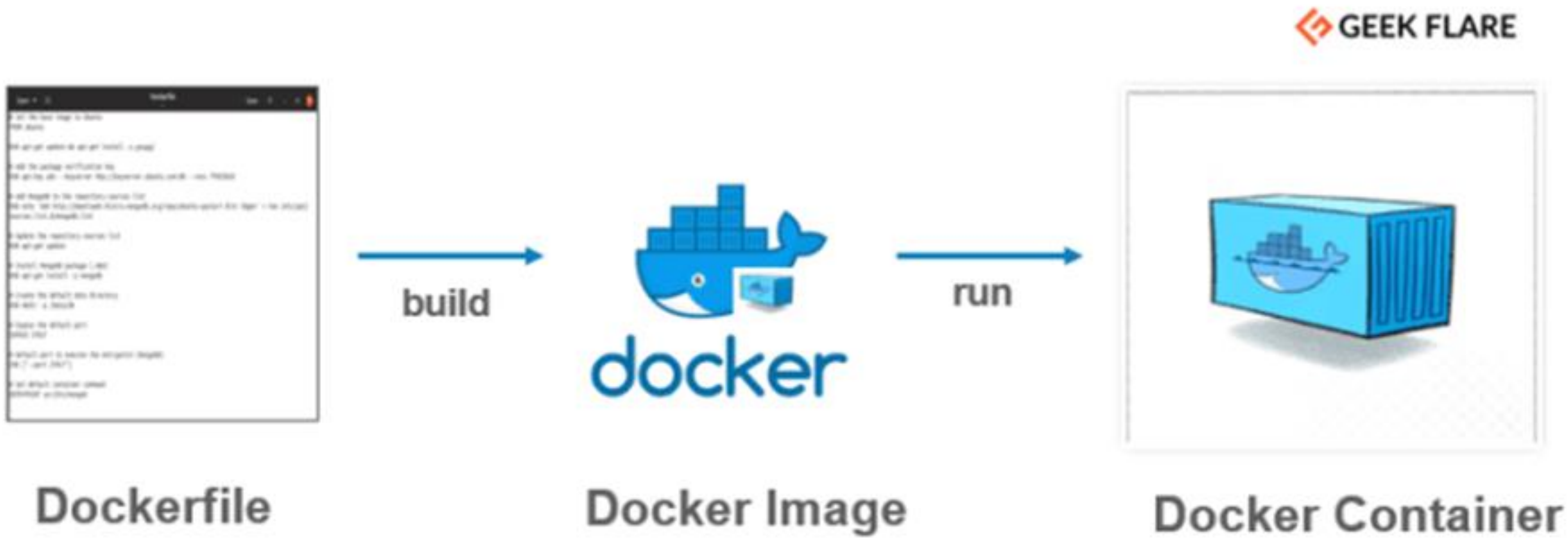
- *VM is isolation of machines, while Containers is isolation of processes*





Reproducibility level 3

A way to create containerized applications = low overhead virtual machines (VMs)



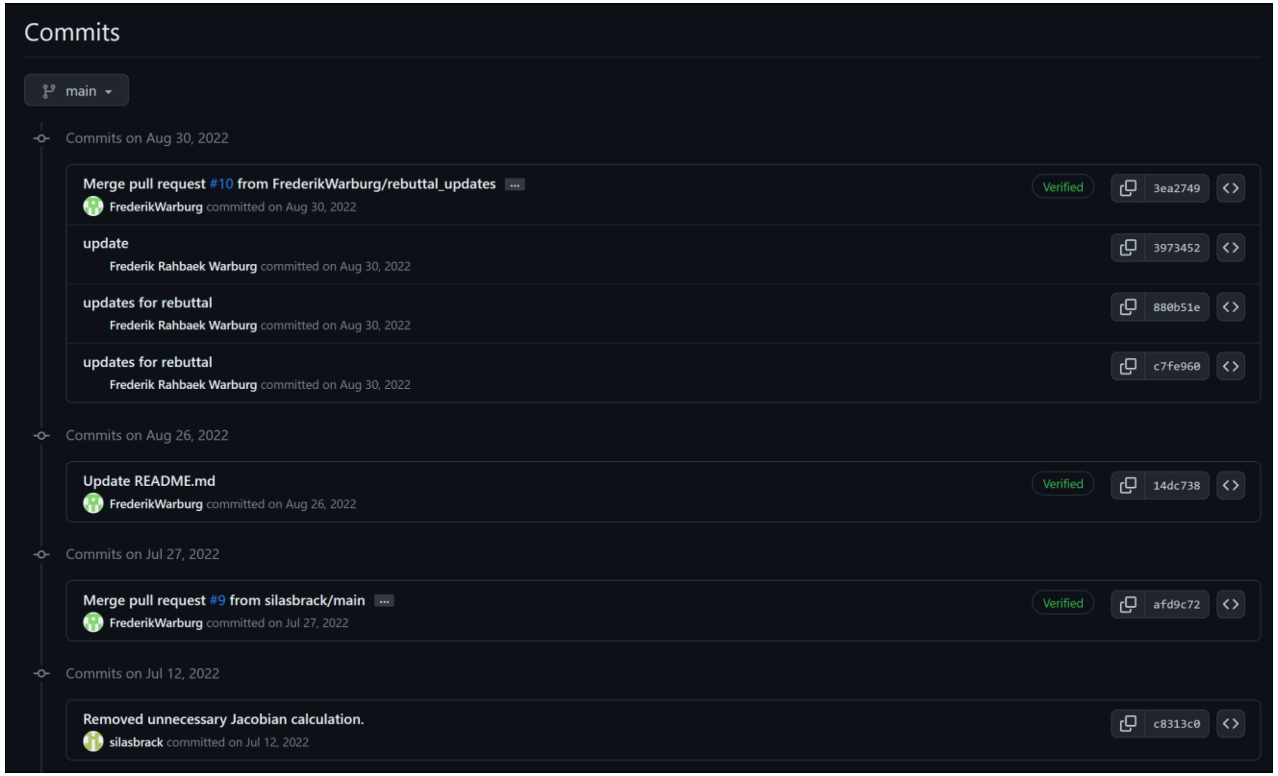
Instructions for creating the container

File containing the everything

A running instance of our application

A 6-step process for reproducible software

Use version control



A 6-step process for reproducible software

Use version control

Use templates

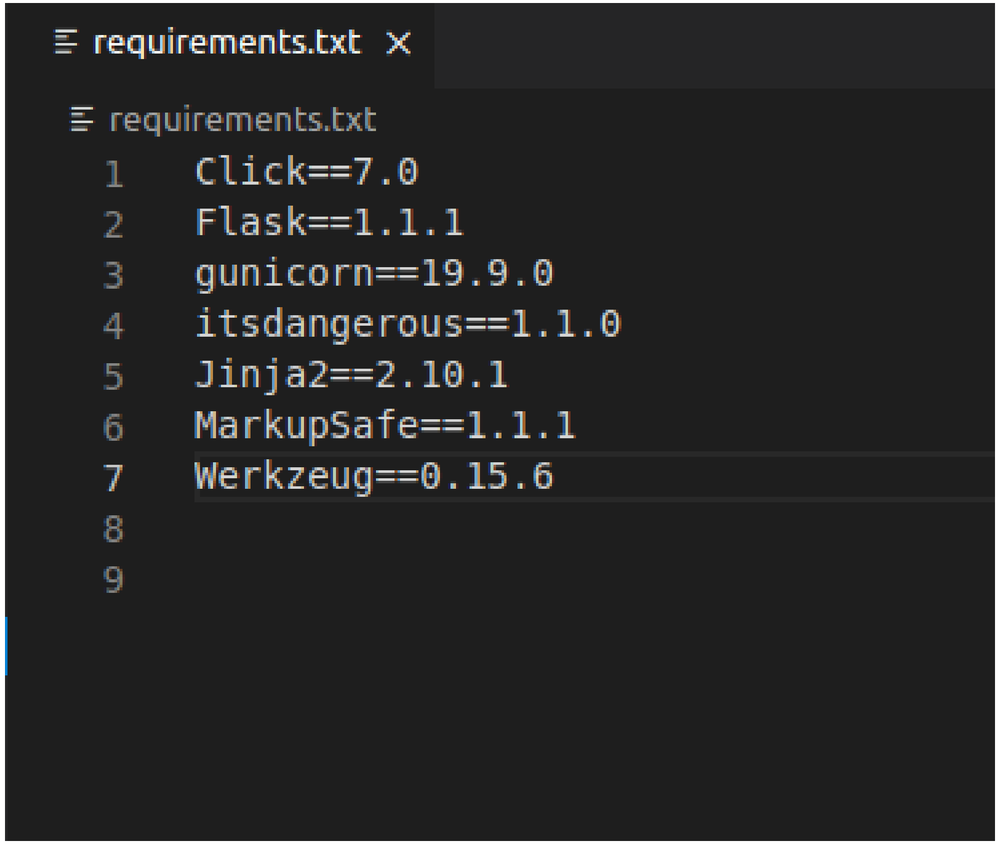
```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── setup.py           <- makes project pip installable (pip install -e .) so src can be imported
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.readthedocs.io
```

A 6-step process for reproducible software

Use version control

Use templates

Write down your dependencies
(and use virtual environments)



A 6-step process for reproducible software

Use version control

Use templates

Write down your dependencies
(and use virtual environments)

Document your code!

Train & Test

To train and test, you can call:

```
cd src;
CUDA_VISIBLE_DEVICES=0 python trainer_[INSERT MODEL].py --config PATH_TO_CONFIG
```

For example to train online LAE on mnist and evaluate on kmnist

```
cd src;
CUDA_VISIBLE_DEVICES=0 python trainer_lae_elbo.py --config ../configs/ood_experiments/mnist/linear/lae_elbo.y
```

and try train a VAE

```
cd src;
CUDA_VISIBLE_DEVICES=0 python trainer_vae.py --config ../configs/ood_experiments/mnist/linear/vae.yaml
```

You can monitor training on tensorboard

```
tensorboard --logdir lightning_log --port 6006
```

To test on missing data imputation experiments, you can call. This require that you have a trained model.

```
cd src/data_imputation;
CUDA_VISIBLE_DEVICES=0 python lae.py
```

or

```
cd src/data_imputation;
CUDA_VISIBLE_DEVICES=0 python vae.py
```

A 6-step process for reproducible software

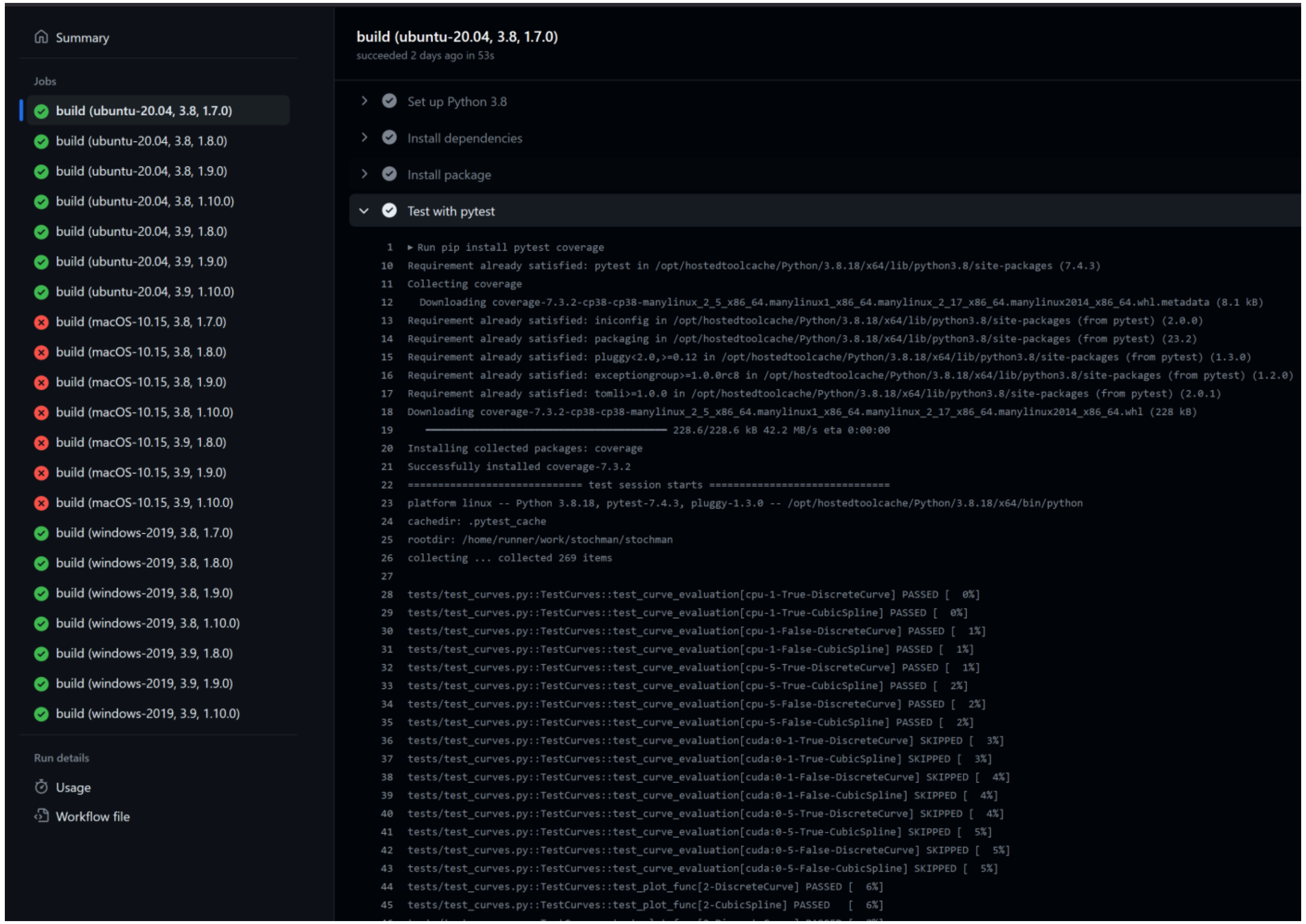
Use version control

Use templates

Write down your dependencies
(and use virtual environments)

Document your code!

Test your code



A 6-step process for reproducible software

- Use version control
- Use templates
- Write down your dependencies (and use virtual environments)
- Document your code!
- Test your code
- Containerize your code

```
# Import a base image so we don't have to start from scratch
#FROM python:3.10-slim
FROM huggingface/transformers-pytorch-cpu

# Use EXPOSE so we can give docker run the appropriate commandline argument (PORT) as:
# docker run predict:latest -e PORT=8000
EXPOSE $PORT
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8

# Run a bunch of linux commands
RUN apt update && \
    apt install --no-install-recommends -y build essential gcc & \
    apt clean & rm -rf /var/lib/apt/lists/*

# Copy the essential files from our folder to docker container.
COPY src/ src/
COPY requirements_predict.txt requirements_predict.txt
COPY setup.py setup.py

RUN pip install -r requirements_predict.txt --no-cache-dir

RUN dvc init --no-scm
RUN dvc remote add -d gcloud_storage gs://mlops-dataset-small
RUN dvc pull

# Set working directory as / and install dependencies
WORKDIR /

RUN mkdir app

# Set entry point, i.e. which file we run with which argument when running the docker container.
# The -u flag makes it print to console rather than the docker log file.
#ENTRYPOINT ["python", "-u", "src/models/predict_model.py"]
CMD exec uvicorn src.models.predict_model:app --host 0.0.0.0 --workers 1 --port $PORT
#ENTRYPOINT ["uvicorn", "src.models.predict_model:app", "--host", "0.0.0.0", "--workers", "1", "--port", $PORT]
```

Meme of the day