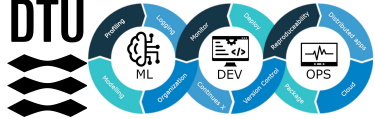


The reproducibility crisis

02476 Machine Learning Operations

Nicki Skafte Detlefsen

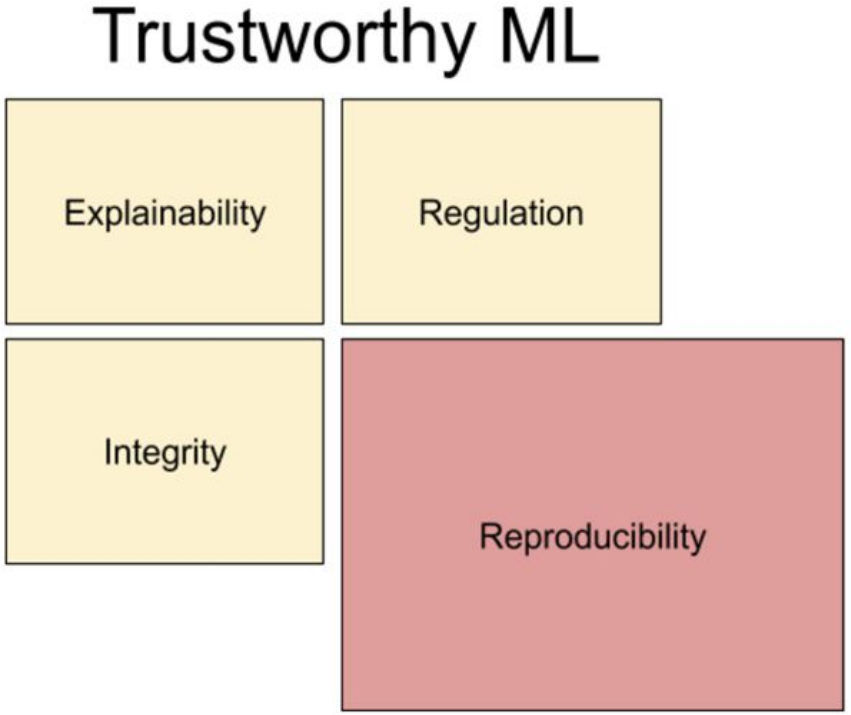
What is reproducibility?

- Being able to reproduce other peoples experimental results is an essential part of the scientific method
- Well known problem throughout most fields (physics, chemistry, biology and computer science)
- With the rise of deep learning, the problem has only been made worse due to competition

Figure credit: https://www.australianenvironmentaleducation.com.au/education-resources/what-is-the-scientific-method/

Why to we need reproducibility?

Reproducibility is a key component in *Trustworthy ML*



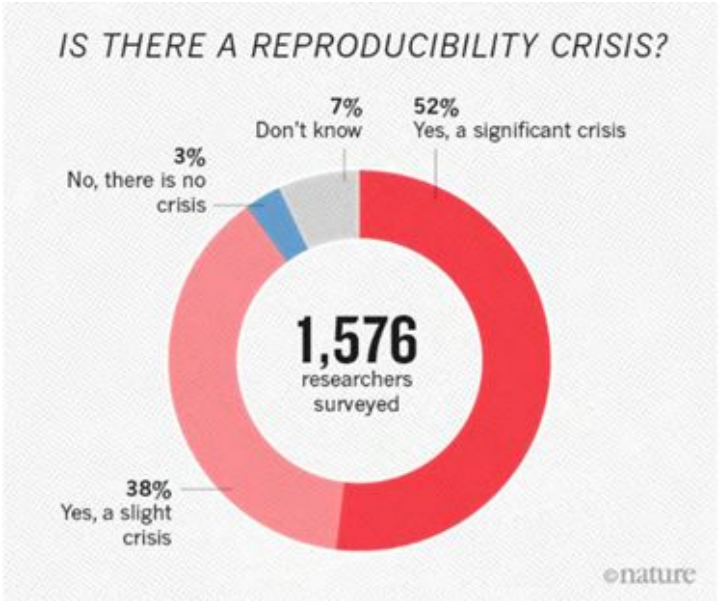
Case:

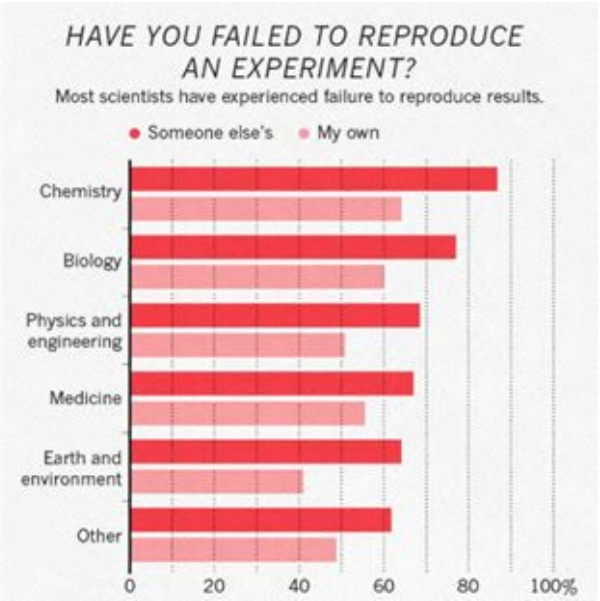
Imaging an AI agent used for diagnostics. Without reproducibility two persons with the exact same symptoms could get different diagnosis

Figure credit: https://towardsdatascience.com/reproducible-machine-learning-cf1841606805

How bad is it?

Wow this is bad...





Machine Learning around [22%]

Figure credit: https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970

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

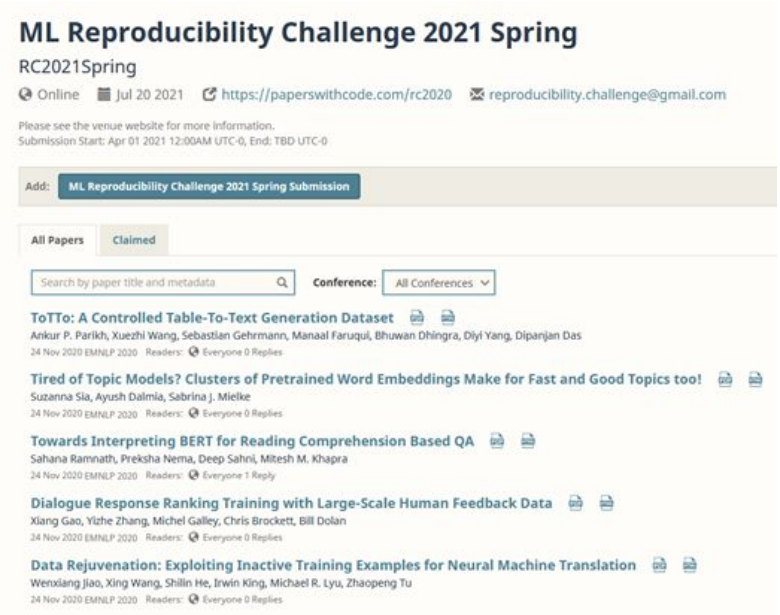
What makes a paper independently reproducible? Debates on reproducibility center around intuition or assumptions but lack empirical results. Our field focuses on releasing code, which is important, but is not sufficient for determining reproducibility. We take the first step toward a quantifiable answer 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

What are the field trying to do?

https://paperswithcode.com/

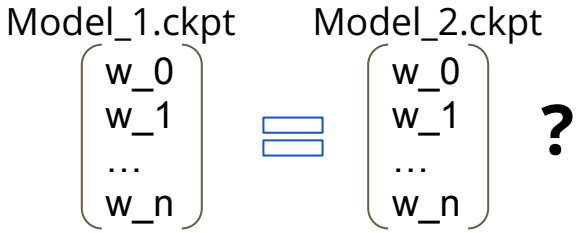


Neurips checklist:

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.

Reproducibility are not a fixed concept

- Reproducibility of model



- Reproducibility of results

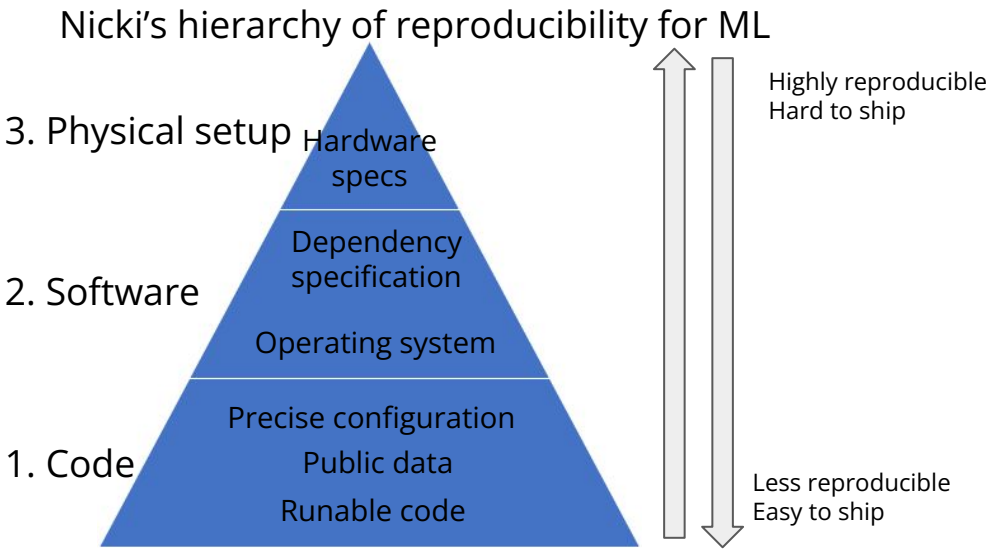
Dataset	Model Architecture	Random Init	Transfer	Parameters	ImageNet Top5
Retina	Resnet-50	96.4% ± 0.05	96.7% ± 0.04	23570408	92.% ± 0.06
Retina	Inception-v3	96.6% ± 0.13	96.7% ± 0.05	22881424	93.9%
Retina	CBR-LargeT	96.2% ± 0.04	96.2% ± 0.04	8532480	77.5% ± 0.03
Retina	CBR-LargeW	95.8% ± 0.04	95.8% ± 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 is the difference?

Table credit: https://ahmdtaha.medium.com/understanding-transfer-learning-for-medical-imaging-d3c1c088ea1b

What can you do about it?

Make sure to document everything about your experiments



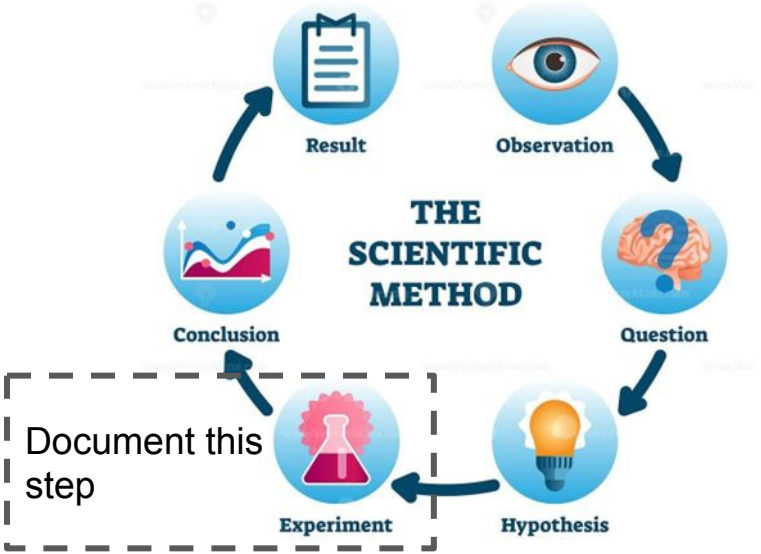


Figure credit: https://www.australianenvironmentaleducation.com.au/education-resources/what-is-the-scientific-method/

A closer look

Use software to abstract away problem

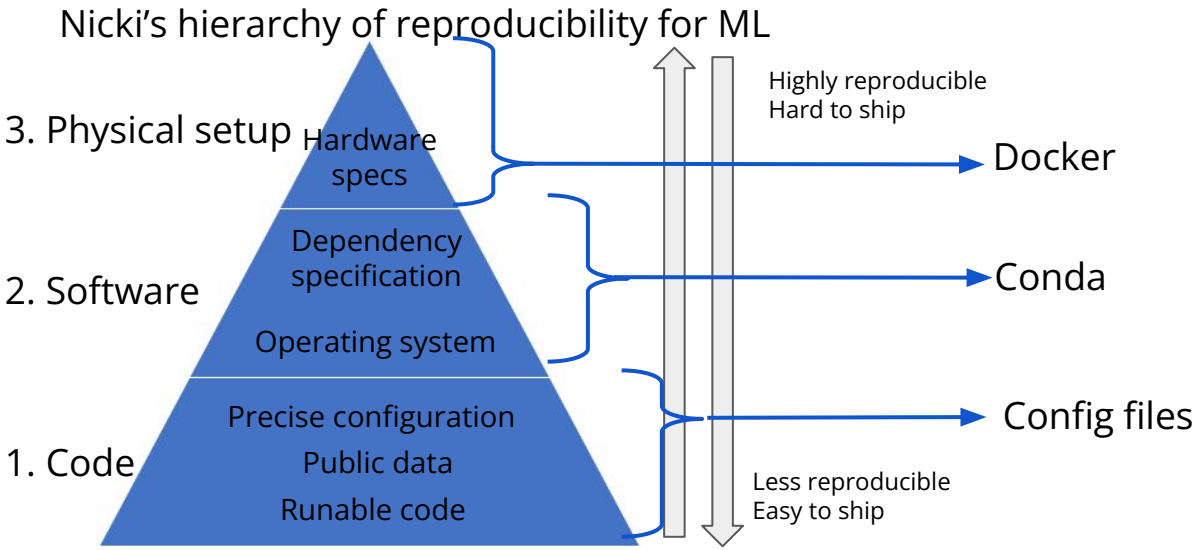
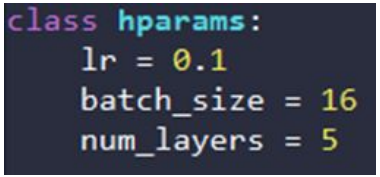
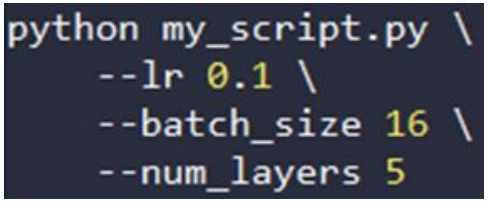


Table 1: Significance test of which paper properties impact reproducibility. Results significant at $\alpha \leq 0.05$ marked with"*".

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

Reproducibility 1: Code level

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

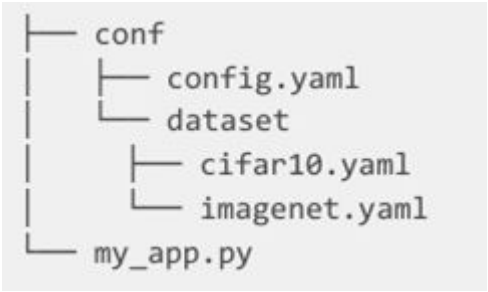
Parameter in script	Argparser	Config files
		
• No easy configuration • Experiments may be lost if not careful	• Easy to configure new experiment • Falls on user to save the config	• Moderate level of configurability • Parameters are systematically saved with experiments

Reproducibility 1: Code level

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

https://github.com/facebookresearch/hydra

Example:




```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 2: Software level

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 3: Hardware level

The easiest would be to hand over your machine. In practise not that possible.

Instead lets hand over a “virtual machine”.

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

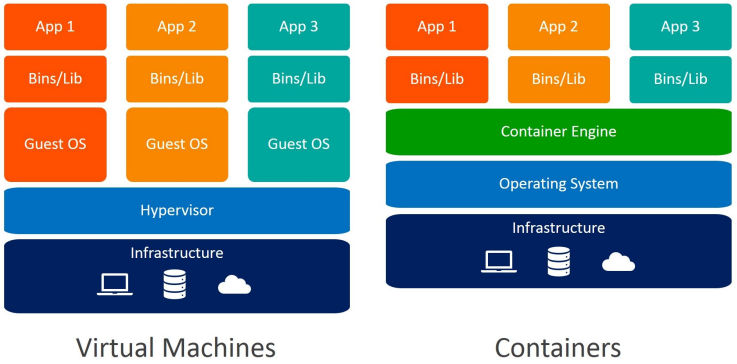
Reproducibility 3: Hardware level

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

Virtual machine by stable diffusion



Virtual Machines

Containers

Reproducibility 3: Hardware level

A way to create containerized applications = specialized VMs

build

docker

run

GEEK FLARE

Dockerfile

How to construct the VM

Docker Image

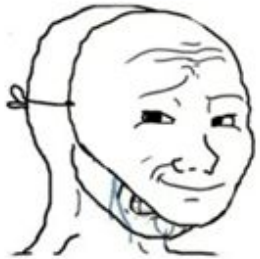
File containing the VM

Docker Container

File containing the VM

Meme of the day

Programmers



This code is unreadable and your dataset is flawed. No one will be able to reproduce your results!

It's not my fault the legacy environment is messed up! We still have 97.3% unit test coverage.

Scientist

This code is unreadable and your dataset is flawed. No one will be able to reproduce your results!

I know :)