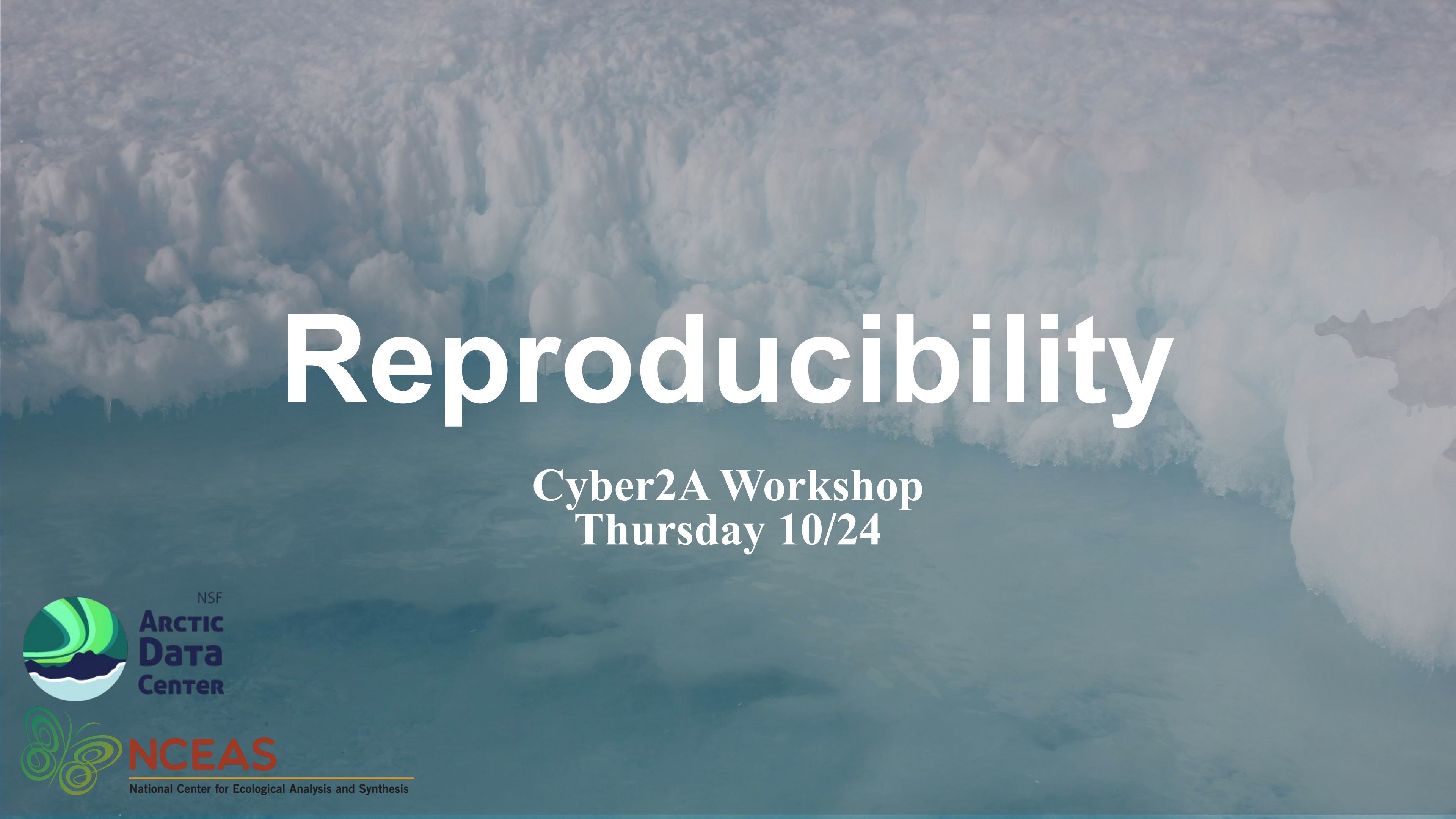


Reproducibility

Cyber2A Workshop
Thursday 10/24

NSF Arctic Data Center

NCEAS
National Center for Ecological Analysis and Synthesis

Hi, I’m Nicole Greco :)

Who am I?

Community Engagement & Outreach Coordinator for the Arctic Data Center & NCEAS Learning Hub

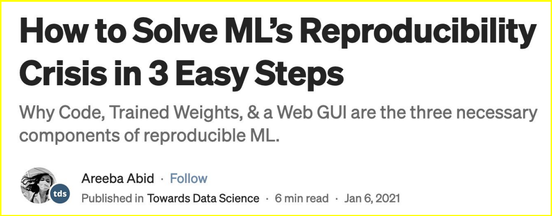
- Plug: If you’ve recently submitted data to the ADC and want to be featured on our website, let me know!

Background & Side Project:

- Glacial Sedimentologist / Marine Geologist
  - Experienced in coding (Python), not ML/AI

Introduction

Reproducibility in AI is not a new topic but is an issue

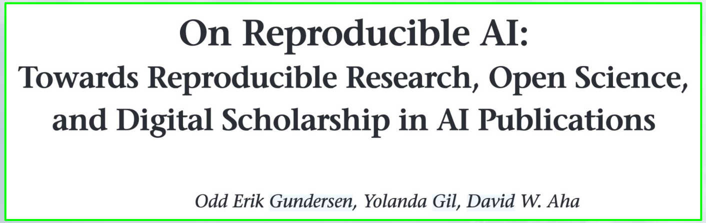


How to Solve ML's Reproducibility Crisis in 3 Easy Steps

Why Code, Trained Weights, & a Web GUI are the three necessary components of reproducible ML.

Areeba Abid · Follow

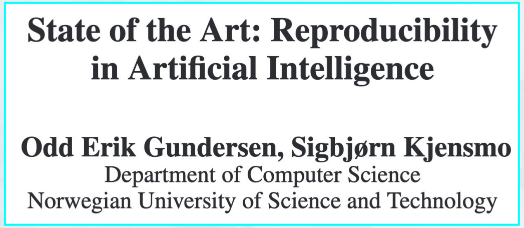
Published in Towards Data Science · 6 min read · Jan 6, 2021



On Reproducible AI:

Towards Reproducible Research, Open Science, and Digital Scholarship in AI Publications

Odd Erik Gundersen, Yolanda Gil, David W. Aha



State of the Art: Reproducibility in Artificial Intelligence

Odd Erik Gundersen, Sigbjørn Kjensmo

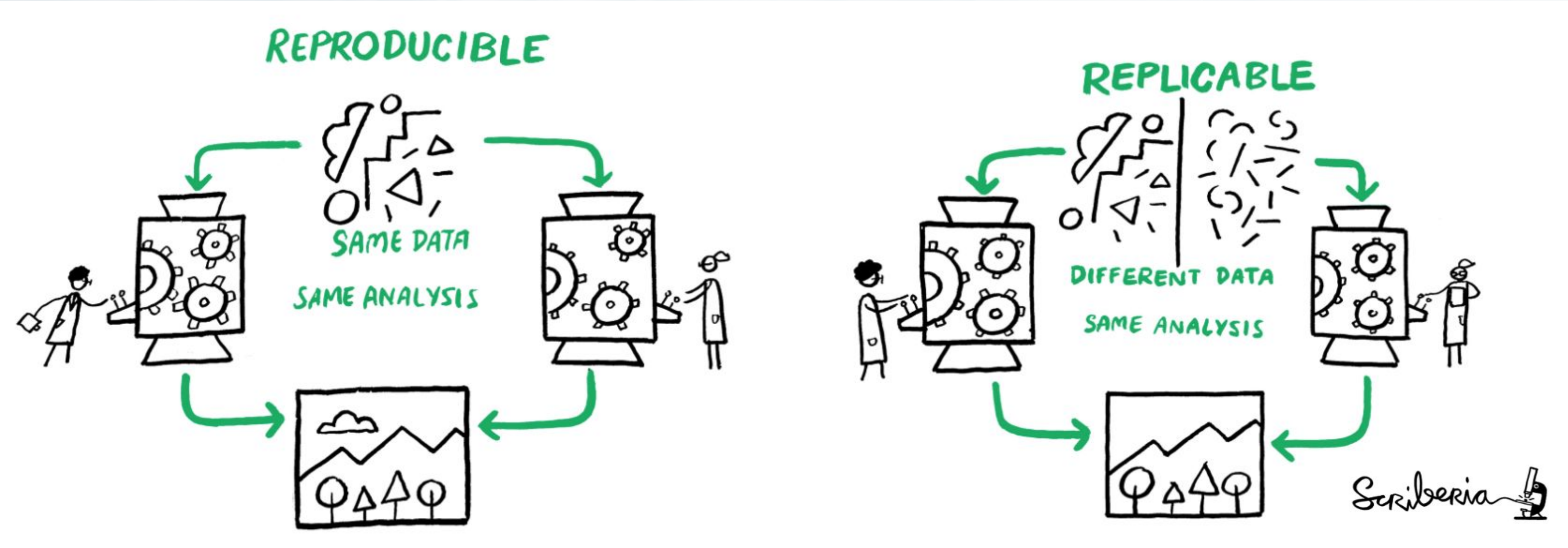
Department of Computer Science

Norwegian University of Science and Technology

This becomes particularly problematic when

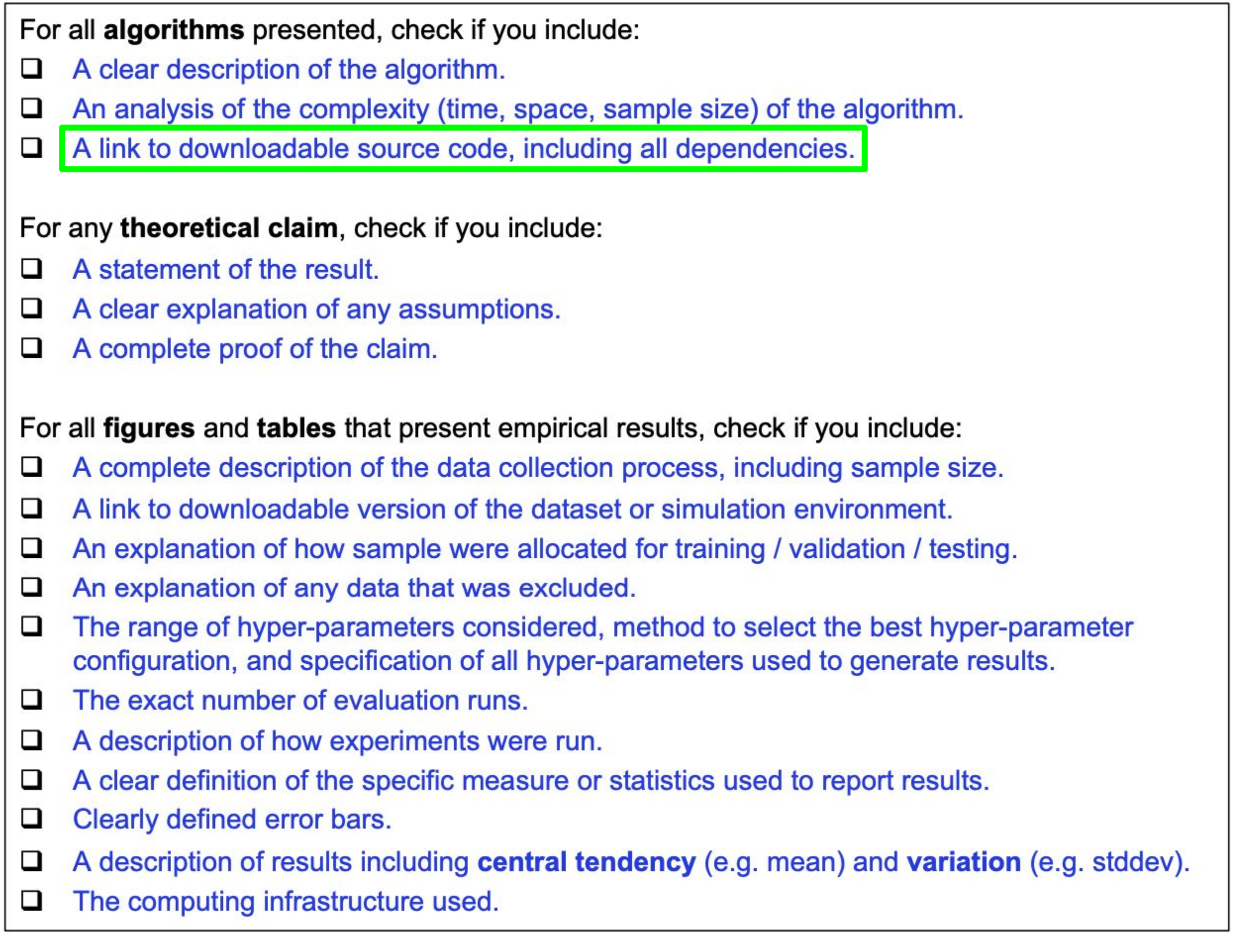
validation of a model requires *reproducing* the model

Why is reproducibility important?



The Reproducibility Checklist

Joelle Pineau, Facebook AI research and computer scientist at McGill University



For all **algorithms** presented, check if you include:

- □ A clear description of the algorithm.
- □ An analysis of the complexity (time, space, sample size) of the algorithm.
- □ A link to downloadable source code, including all dependencies.

For any **theoretical claim**, check if you include:

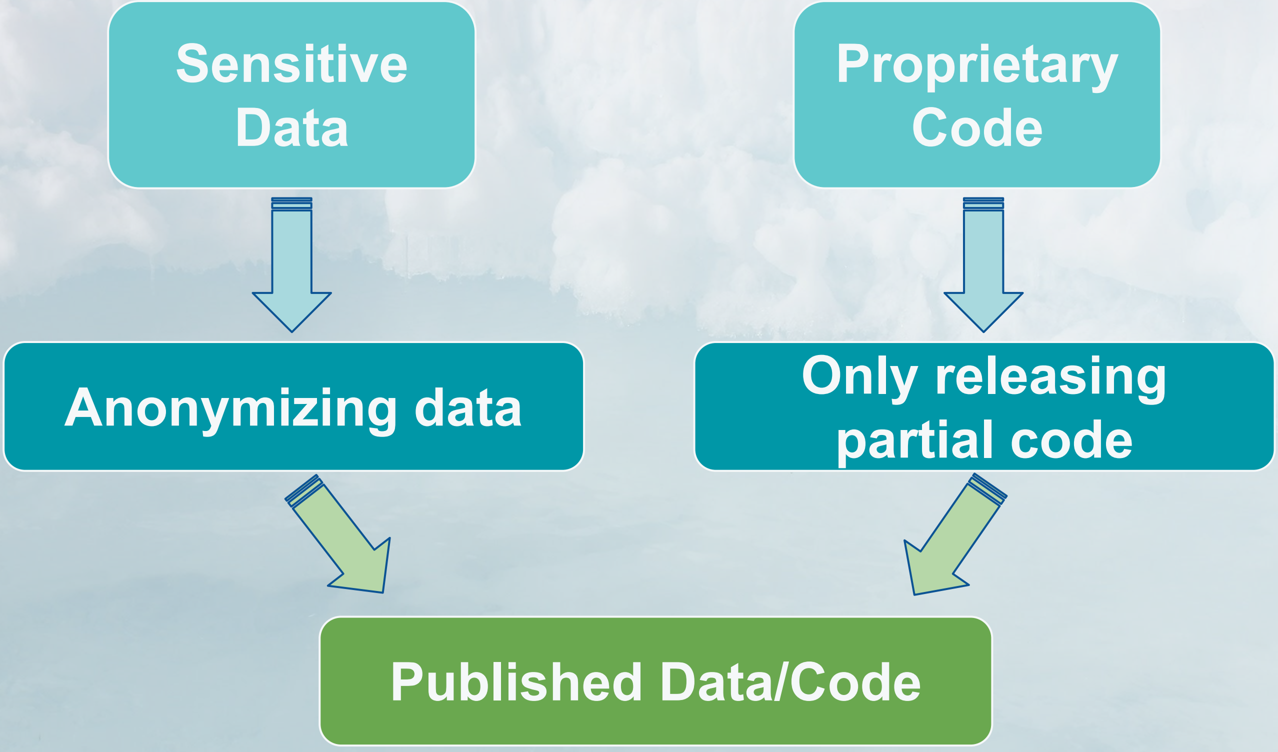
- □ A statement of the result.
- □ A clear explanation of any assumptions.
- □ A complete proof of the claim.

For all **figures** and **tables** that present empirical results, check if you include:

- □ A complete description of the data collection process, including sample size.
- □ A link to downloadable version of the dataset or simulation environment.
- □ An explanation of how sample were allocated for training / validation / testing.
- □ An explanation of any data that was excluded.
- □ The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results.
- □ The exact number of evaluation runs.
- □ A description of how experiments were run.
- □ A clear definition of the specific measure or statistics used to report results.
- □ Clearly defined error bars.
- □ A description of results including **central tendency** (e.g. mean) and **variation** (e.g. stddev).
- □ The computing infrastructure used.

https://media.neurips.cc/Conferences/NIPS2018/Slides/jpineau-NeurIPS-dec18-fb.pdf

Consider the sensitivity of your data/code when publishing



Sensitive Data

Proprietary Code

Anonymizing data

Only releasing partial code

Published Data/Code

Sharing Code

Sharing code is the first step to solving the problem of reproducibility and allows researchers to:

Validate the model

Track code construction and see any author annotations

Expand on published work

Sharing Code Continued

Despite this, sharing code does not always mean that models are fully reproducible

Many ML models are trained on restricted datasets

and/or require extensive computing power for training the model

Sharing Code Continued

Because of this, there are a few additional criteria that improve reproducibility including:

Data and metadata availability (must be included without question)

Easily installable computational analysis tools and pipelines

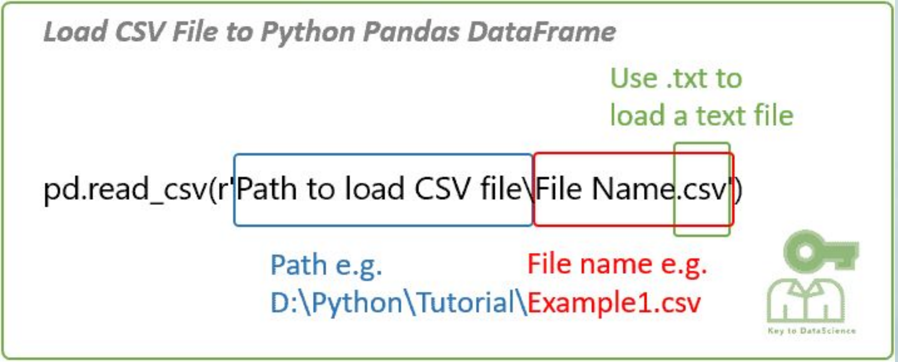
Transparency of the code you’re using and dependencies needed to run the code

Installed software should behave the same on every machine and should have the same runtime

Sharing Code Continued

Tips & tricks:

- Avoid using absolute file paths when reading in data (and in general the use of slashes as these differ between operating systems)



Load CSV File to Python Pandas DataFrame

```
pd.read_csv(r'Path to load CSV file\File Name.csv')
```

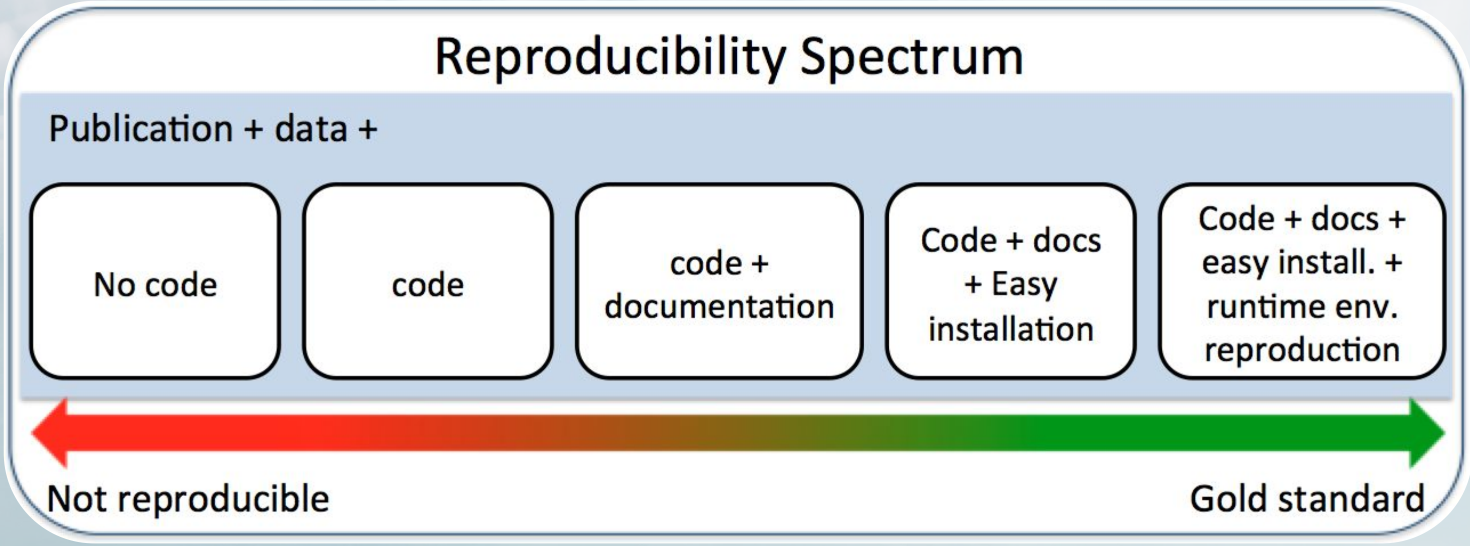
Use .txt to load a text file
Path e.g. D:\Python\Tutorial\
File name e.g. Example1.csv
Key to DataScience

https://keytodatascience.com/python-read-csv-txt-file/

- Clean data *within* your code, avoid copying/pasting in a spreadsheet, always keep an unedited version of your raw data

THAT FACE YOU MAKE WHEN
YOU CALL A COLLEAGUE OVER TO TROUBLESHOOT AN ISSUE, AND IT TURNS OUT YOU MADE A COPY/PASTE ERROR

Sharing Code Summary



Reproducibility Spectrum

Publication + data +

No code

code

code + documentation

Code + docs + Easy installation

Code + docs + easy install. + runtime env. reproduction

Not reproducible

Gold standard

https://towardsdatascience.com/scientific-data-analysis-pipelines-and-reproducibility-75ff9df5b4c5

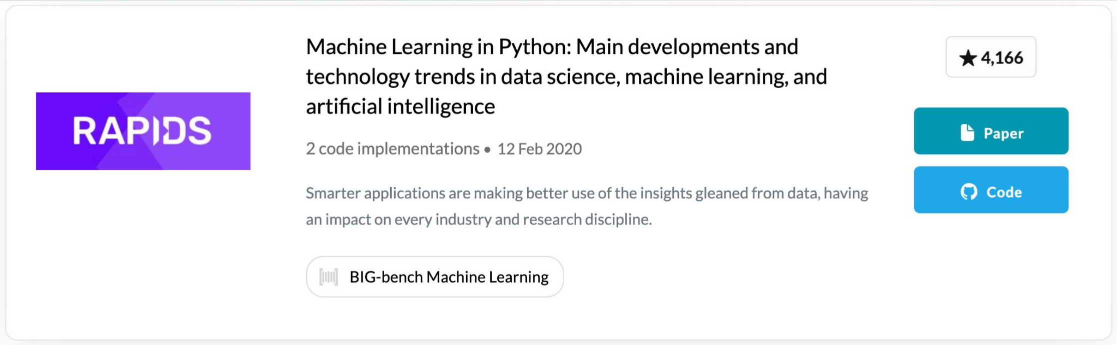
Model Repositories

PyTorch

PyTorch Hub is a pre-trained model repository designed to facilitate reproducibility and enable new research

Compatible with: CO jupyter Papers With Code

& more!



RAPIDS

Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence

2 code implementations • 12 Feb 2020

Smarter applications are making better use of the insights gleaned from data, having an impact on every industry and research discipline.

BIG-bench Machine Learning

Free, open source hub for publications that include direct links to GitHub code, no account needed for access

Version Control

- The process of keeping track of every individual change by each contributor that’s saved in a version control framework, or special database
- Keeping a history of these changes to track model performance relative to model parameters saves the time you’d spend retraining the model

The 3 components of ML version control:

Code

Data

Model

Code

We recommend writing and storing your model code in the same language as your implementation code to make it easier to maintain all code and dependencies

Data

Versioning should link the data to the appropriate metadata and note any changes in either

Model

The model connects your code and data with model parameters and analysis

Version Control Continued

Using a version control system ensures easier:

- **Collaboration**

Nicole updated ML.qmd

Nicole updated sessions

Nicole added outlines for select modules

- **Versioning**
  - If your model breaks, you'll have a log of any changes that were made, allowing you or others to revert back to a stable version

- **Dependency tracking**
  - You can test more than one model on different branches or repositories, tune the model parameters, and monitor the accuracy of each implemented change

- **Model updates**
  - Version control allows for incrementally released versions while continuing the development of the next release

Wrap-Up

Consider the following to ensure your model is reproducible:

- Use the reproducibility checklist for algorithms, theoretical claims, and figures/tables
- Anonymize any sensitive data and remove proprietary code before publishing
    - BUT still provide training data and enough code for others to replicate your model
- Share data and metadata, be transparent in any dependencies needed to run your model, use easily installable computational analysis tools and pipelines, and ensure installed software behaves the same on every machine (i.e. runtime)
- Use a pre-trained model repository (ex. PyTorch Hub) and publish to open-source journals/websites (ex. Papers with Code)
- Practice efficient version control (recommend GitHub if working with collaborators)

LEGO Metadata Activity (30m for activity, 10m for review)

Lego Activity Teams

Team 1: Anne, Aman, Mikhail, Mia

Team 2: Gillian, Kalum, Jake, Taylor

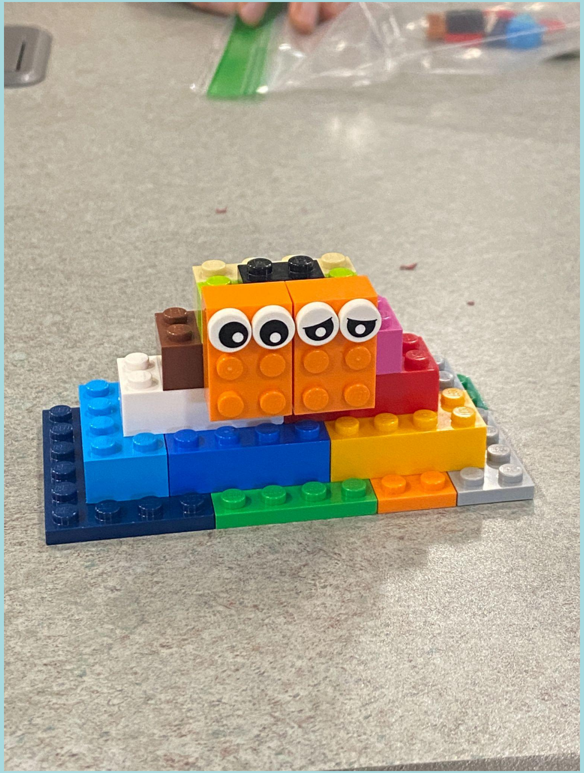
Team 3: Munish, Ellen, Barrett, Elchin

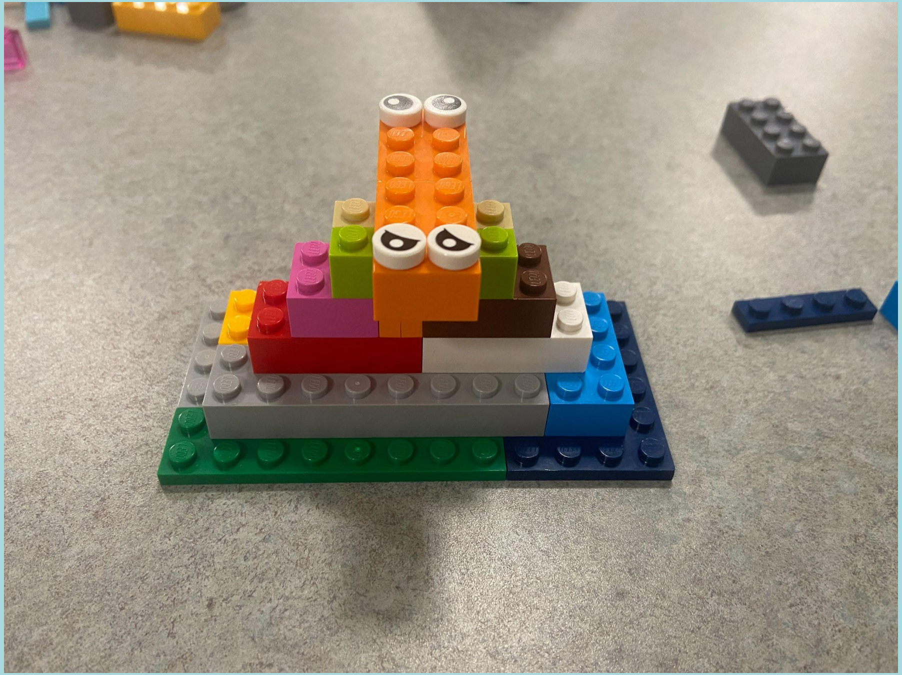
Team 4: Kamil, Varunesh, Irina, Susan

Team 5: Mahboubeh, Dogukan, Ivan, Wilson

Team 1 Creators (Anne, Aman, Mikhail, Mia)

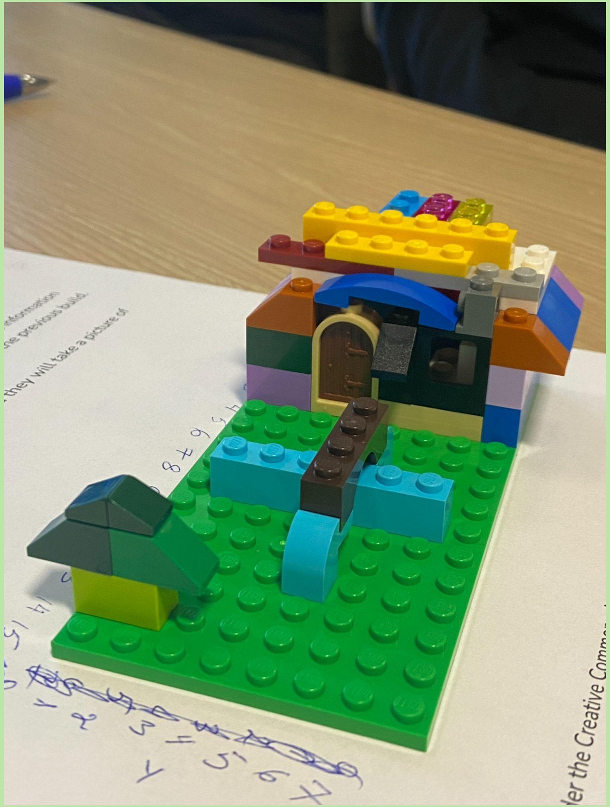
Pyramid with Eyes

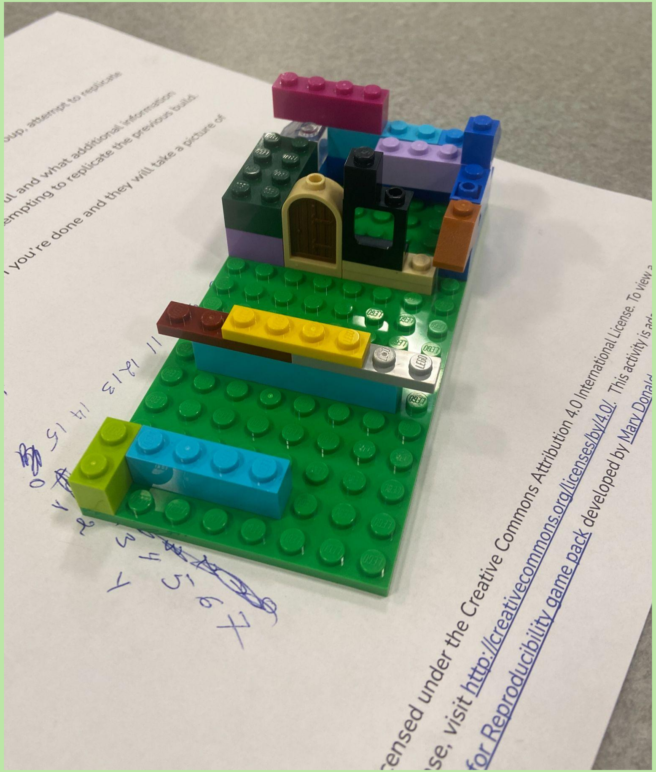




Team 2 Creators (Gillian, Kalum, Jake, Taylor, Alyona)

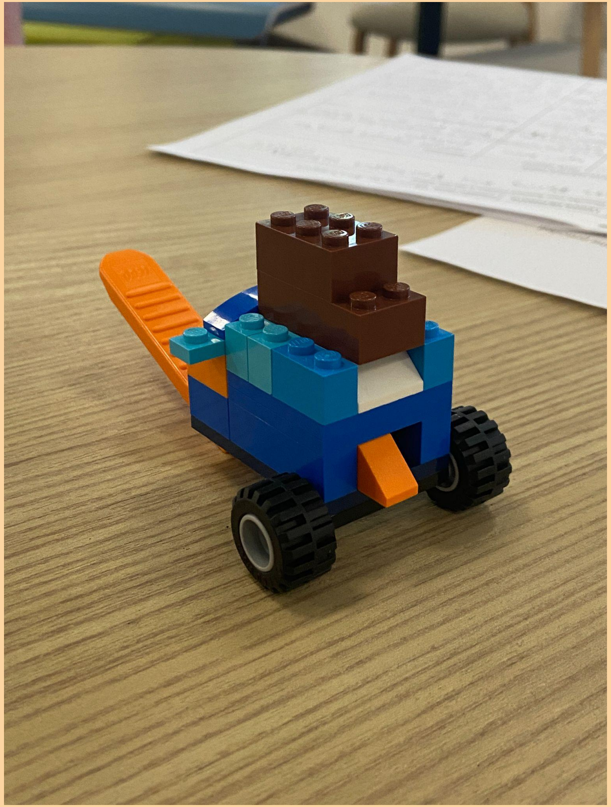
My Dream House in Sandy Ago

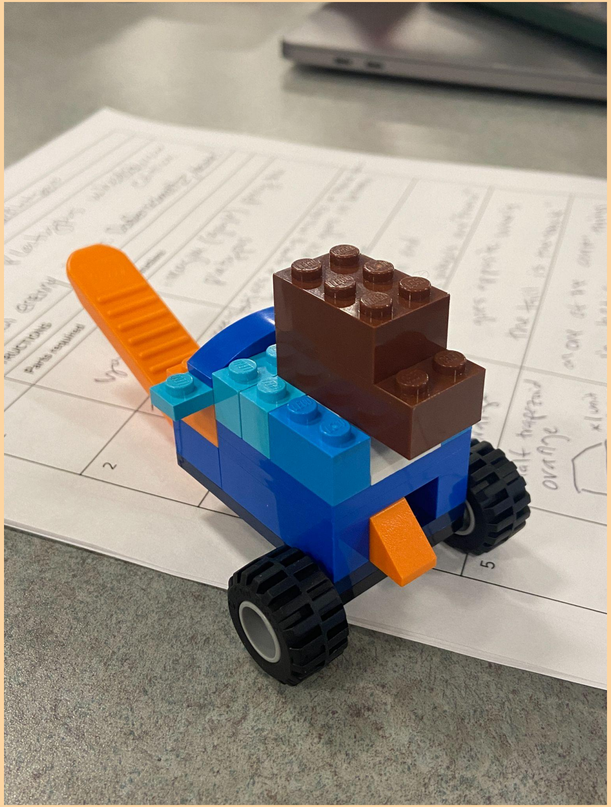




Team 3 Creators (Munish, Ellen, Barrett, Elchin)

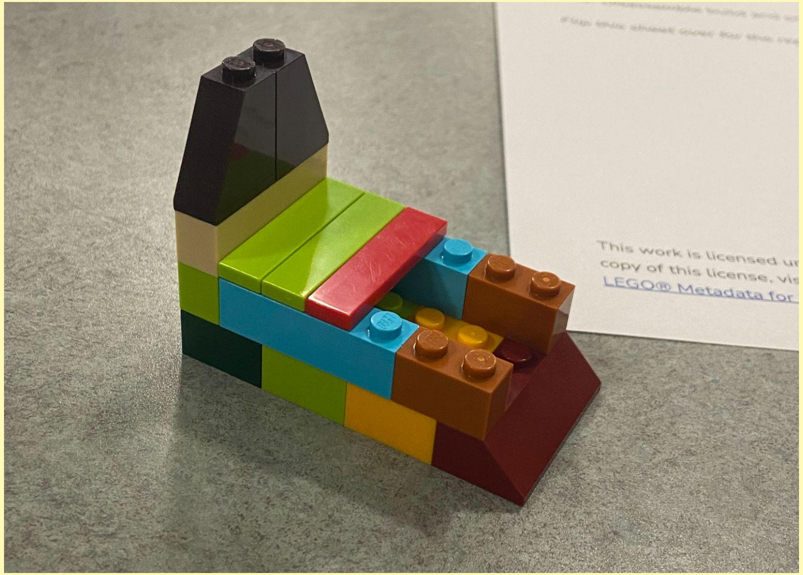
Perry the Platypus

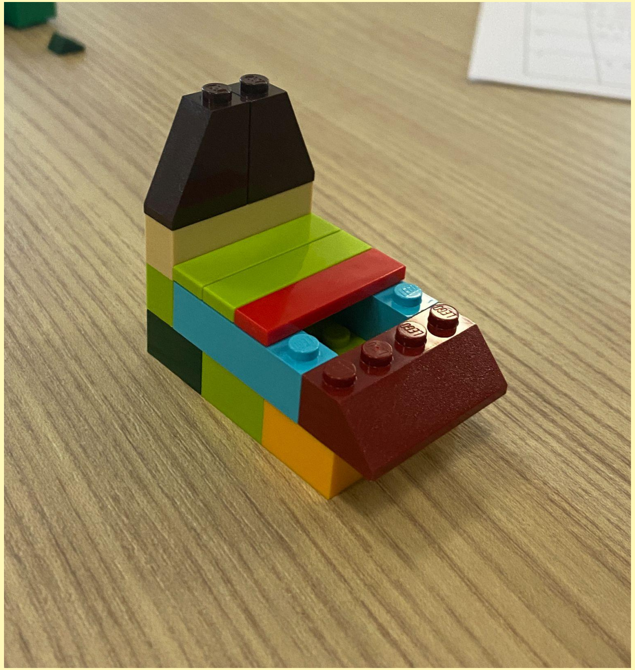




Team 4 Creators (Kamil, Varunesh, Irina, Susan, Minu)

Open Science House





Team 5 Creators (Mahboubeh, Dogukan, Ivan, Wilson, Sandeep)

The Dream House RV

Discussion

- What were some assumptions you made while writing your instructions?
- Were there any unexpected hurdles you encountered when writing your instructions or trying to replicate another group's structure?
- What did you find most difficult about this activity?
- Now that you see how successful or unsuccessful the other group was in recreating your structure, is there anything you would do differently?

References & Resources

1. Gundersen, Odd Erik, and Sigbjørn Kjensmo. 2018. “State of the Art: Reproducibility in Artificial Intelligence”. *Proceedings of the AAAI Conference on Artificial Intelligence* 32 (1). https://doi.org/10.1609/aaai.v32i1.11503.
2. Gundersen, Odd Erik, Yolanda Gil, and David W. Aha. “On Reproducible AI: Towards Reproducible Research, Open Science, and Digital Scholarship in AI Publications.” *AI Magazine* 39, no. 3 (September 28, 2018): 56–68. https://doi.org/10.1609/aimag.v39i3.2816.
3. “How the AI Community Can Get Serious about Reproducibility.” Accessed September 18, 2024. https://ai.meta.com/blog/how-the-ai-community-can-get-serious-about-reproducibility/.
4. Abid, Areeba. “Addressing ML’s Reproducibility Crisis.” Medium, January 7, 2021. https://towardsdatascience.com/addressing-mls-reproducibility-crisis-7d59e9ed050.
5. PyTorch. “Towards Reproducible Research with PyTorch Hub.” Accessed September 18, 2024. https://pytorch.org/blog/towards-reproducible-research-with-pytorch-hub/.
6. Stojnic, Robert. “ML Code Completeness Checklist.” *PapersWithCode* (blog), April 8, 2020. https://medium.com/paperswithcode/ml-code-completeness-checklist-e9127b168501.
7. Akalin, Altuna. “Scientific Data Analysis Pipelines and Reproducibility.” Medium, July 5, 2021. https://towardsdatascience.com/scientific-data-analysis-pipelines-and-reproducibility-75ff9df5b4c5.
8. Hashesh, Ahmed. “Version Control for ML Models: What It Is and How To Implement It.” neptune.ai, July 22, 2022. https://neptune.ai/blog/version-control-for-ml-models.
9. NCEAS Learning Hub: https://www.nceas.ucsb.edu/learning-hub