Day5 - Project

02476 Machine Learning Operations

Nicki Skafte Detlefsen, Associate Professor, DTU Compute

January 2026

The job

💡 You (and your group) are just hired as an MLOps engineers at a start-up.

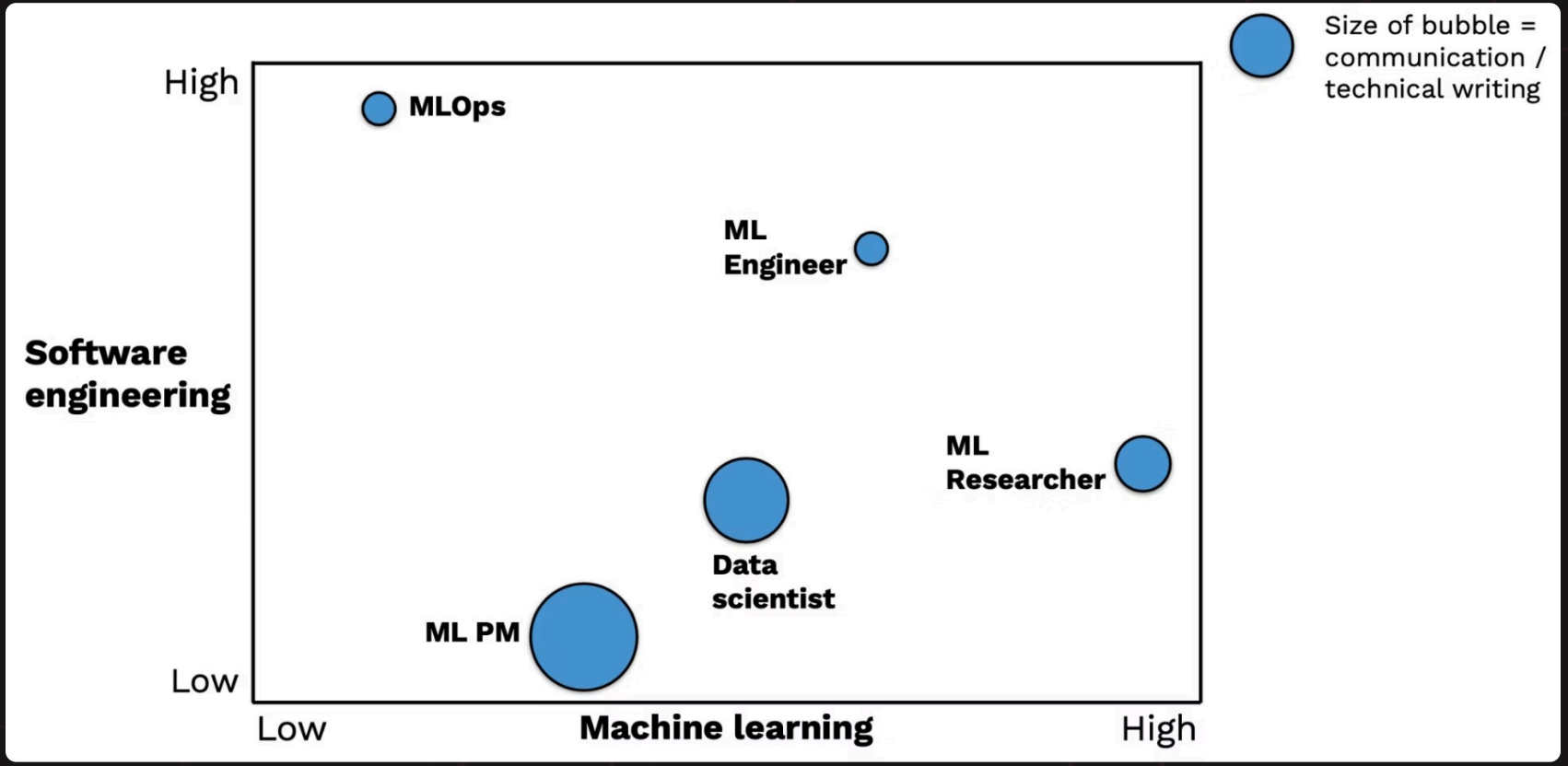
Your first job:

- *Develop an MLOps pipeline to solve a specific task for the company*

💡 Importantly: You are judged not by how great the model is but how fast you can setup a pipeline to solve the task.

Why you do not need to care about the model?

That is a job for the ML research not MLOps engineer

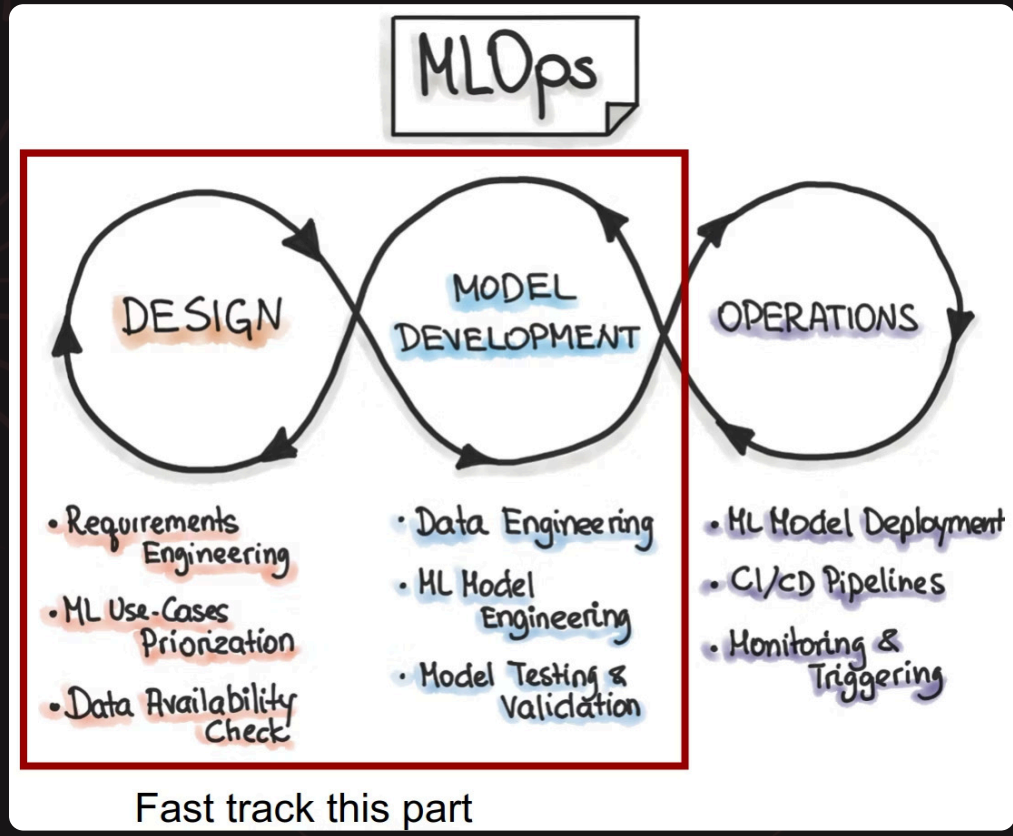


How to solve the problem?

💡You already have all the tools for the pipeline, you just need a good starting model.

💡Your base framework is Pytorch

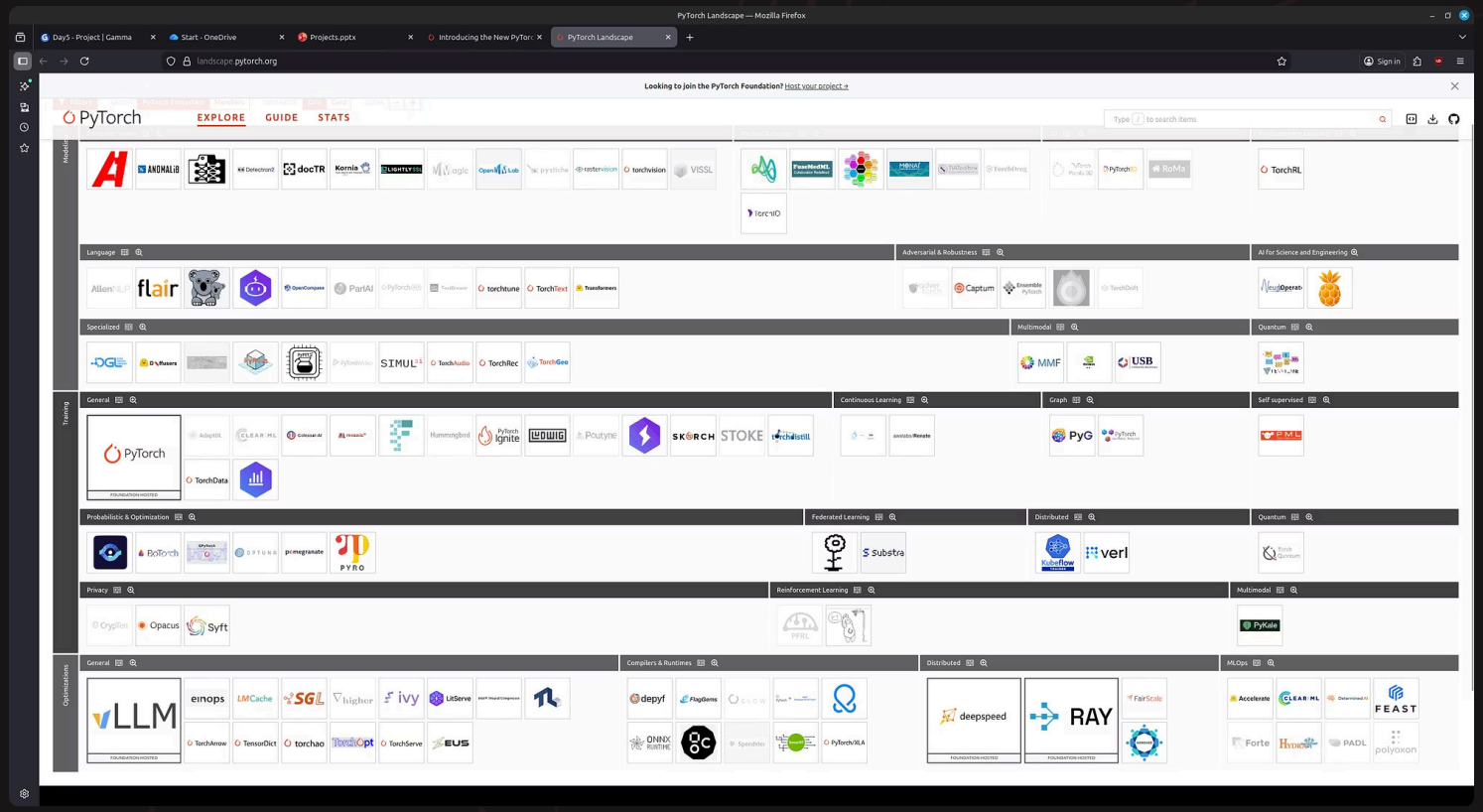
💡You turn your attention towards open-source projects build on top of Pytorch



The Pytorch Landscape♻️

💡Collection of frameworks build to be used in collaboration with Pytorch https://landscape.pytorch.org/

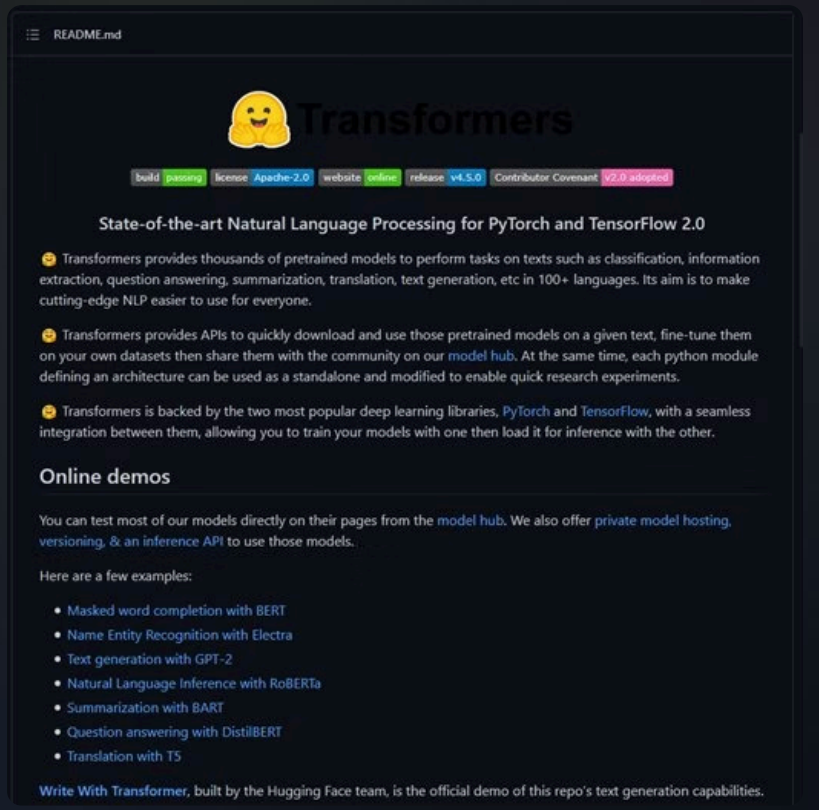
💡It is not a complete list of all great frameworks



Example 1: Transformers

https://github.com/huggingface/transformers

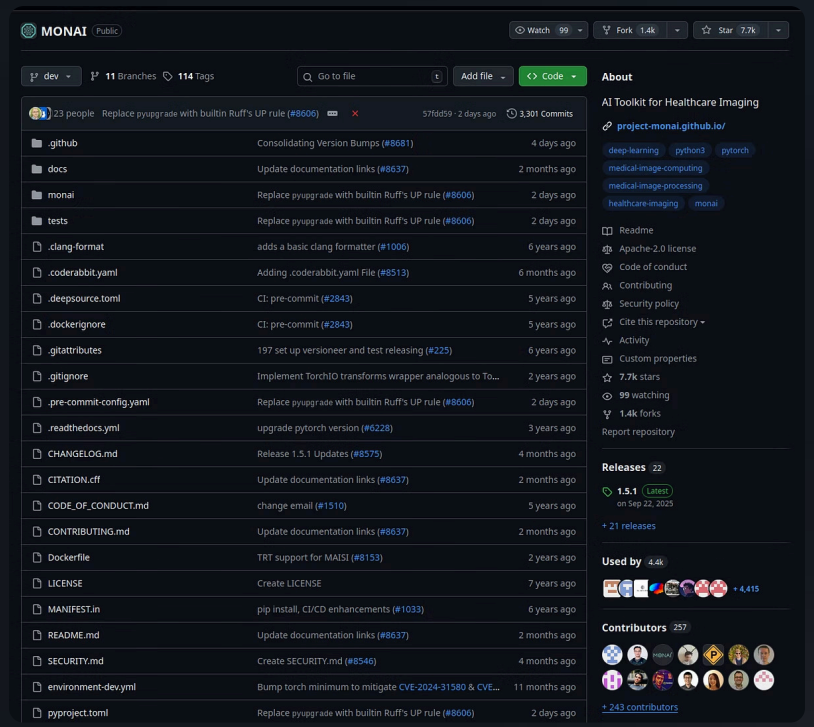
Provides state-of-the-art NLP models for both Pytorch, Jax and Tensorflow.



Example 2: Monai

https://github.com/Project-MONAI/MONAI

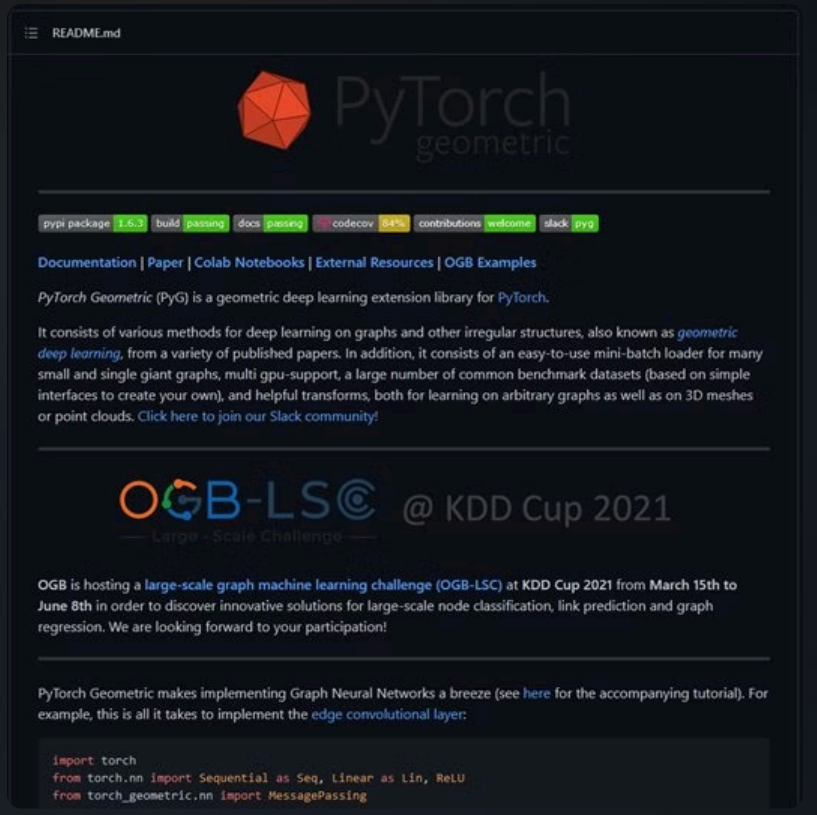
Models for healtcare imaging



Example 3: PyTorch geometric

https://github.com/pyg-team/pytorch_geometric

Graph Neural Network Library for PyTorch to work on irregular data such as graphs and points.

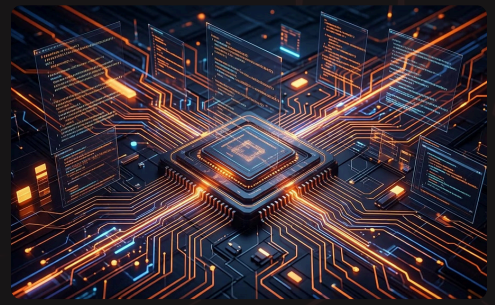


A open-source framework can usually get you 80% of the way

Open-source frameworks provide a robust foundation for MLOps, covering most common functionalities and significantly reducing development effort.

Pre-built Models & Algorithms

Access state-of-art, often pre-trained, models and algorithms ready for fine-tuning.



Battle-tested Code

Community-maintained, extensively tested, and optimized for higher reliability.

Strong Community Support

Extensive documentation, tutorials, and forums make learning and troubleshooting accessible.

The remaining 20% focuses on unique differentiation:

Customization

Tailor models to unique business logic, data types, and performance needs.

Integration

Connect ML pipelines with existing systems, data sources, and applications.

Deployment

Adapt to infrastructure, set up monitoring, scaling, and security protocols.

Your first task

Find a Dataset

Identify a compelling dataset that aligns with your interests and project goals. Consider its structure, size, and relevance.

Choose a Model

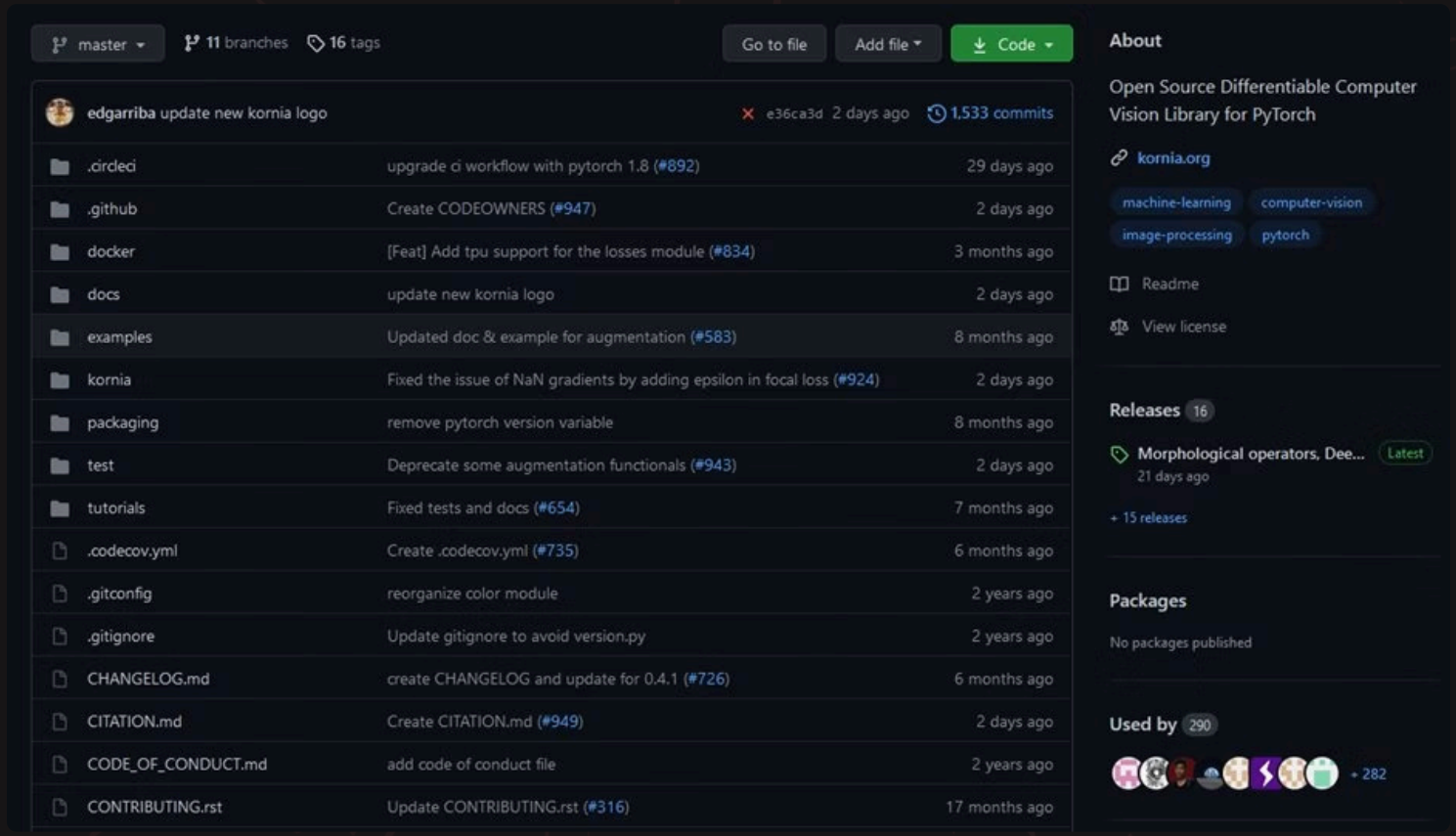
Select a suitable machine learning or deep learning model. Explore various architectures and their applications for your chosen dataset.

Set the Right Scope

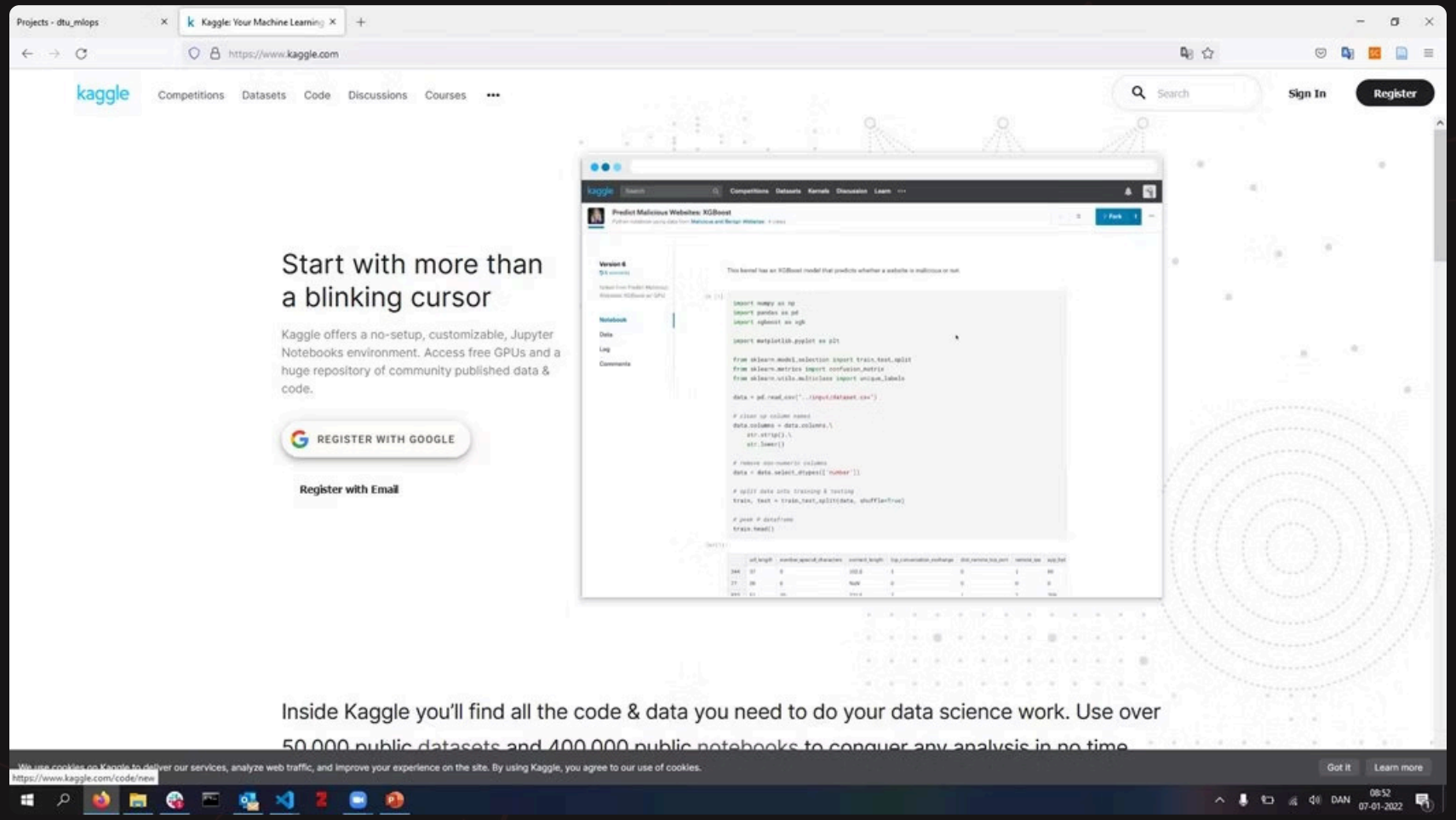
Aim for a challenge that is harder than basic benchmarks (e.g., MNIST, CIFAR) but easier than training a large language model (LLM) from scratch. Find your sweet spot!

How to get an good idea?

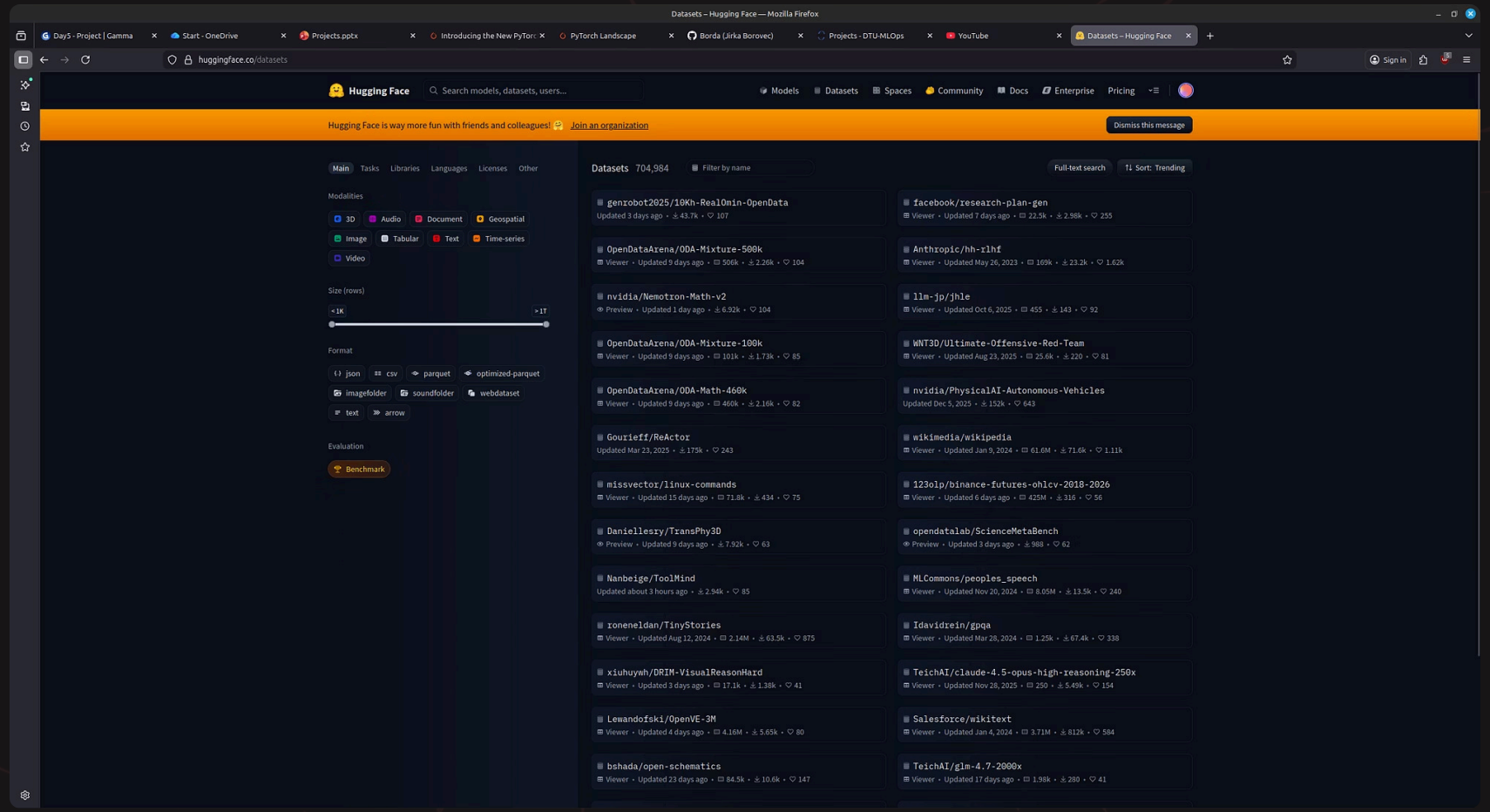
Look at the **used by** section on github



How to get an good idea?



How to get an good idea?



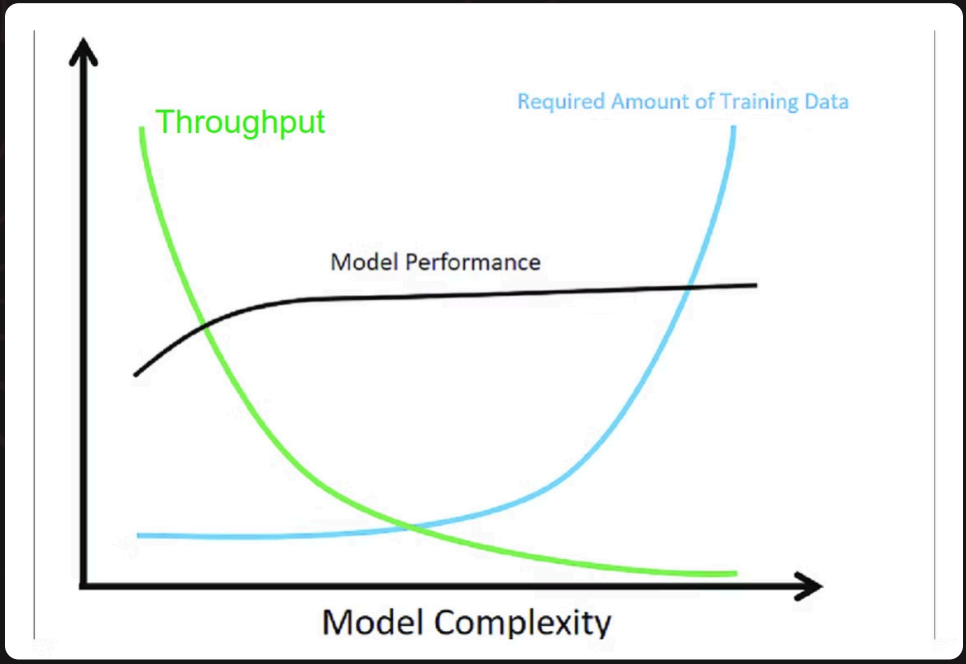
General recommendations

📊Data

- Choose where data loading is not too complex
- <10 GB (else work on a subset)

🤖Model

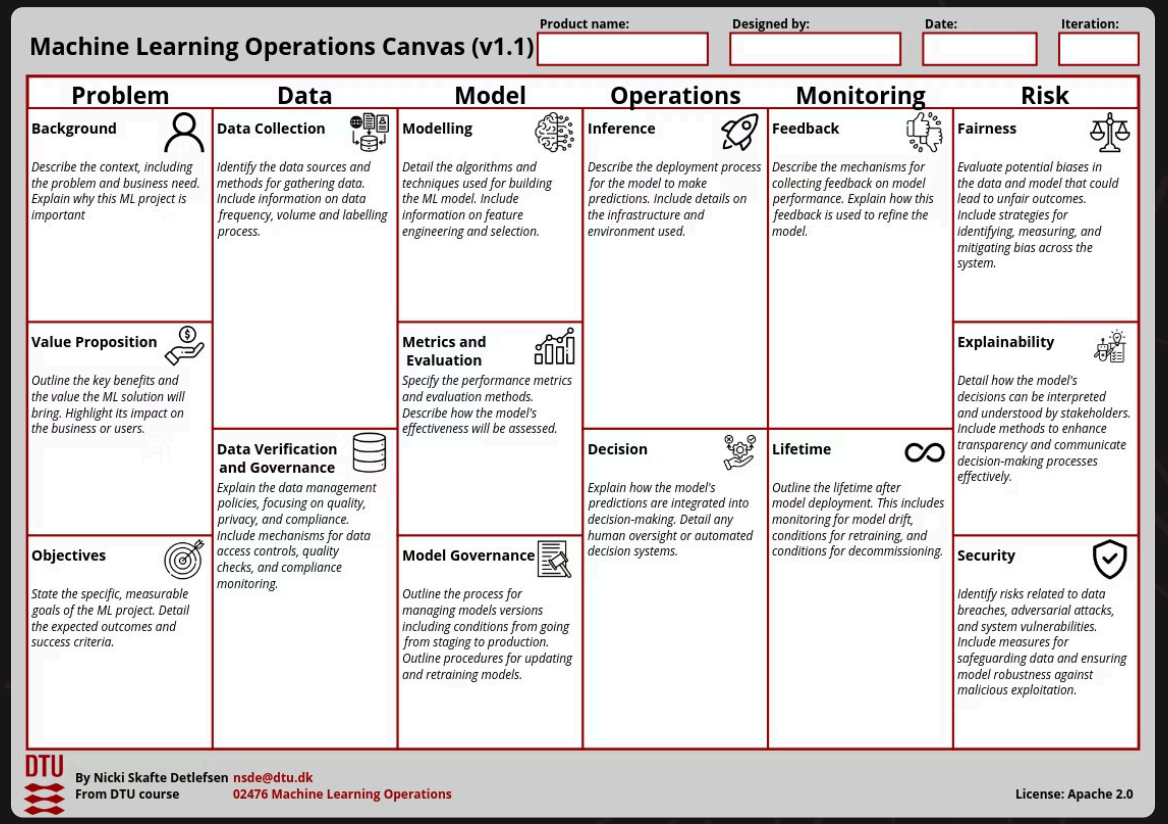
- Start out with a public baseline model if possible
- Choose smaller models over large models



Summary

1. Pick a dataset you would like to work with
2. Pick a model you would like to work with
3. Write a small project description containing
    a. Overall goal of the project
    b. What data are you going to run on (initially, may change). Describe overall number of samples, size, modality...
    c. What models do you expect to use
4. Create project repository
5. Upload project description as part of **README.md** file
6. Work on the rest of project...

ML Canvas for staying organized and thinking ahead



Machine Learning Operations Canvas (v1.1)

Product name: | Designed by: | Date: | Iteration:

Problem	Data	Model	Operations	Monitoring	Risk
Background *Describe the context, including the problem and business need. Explain why this ML project is important*	**Data Collection** *Identify the data sources and methods for gathering data. Include information on data frequency, volume and labelling process.*	**Modelling** *Detail the algorithms and techniques used for building the ML model. Include information on feature engineering and selection.*	**Inference** *Describe the deployment process for the model to make predictions. Include details on the infrastructure and environment used.*	**Feedback** *Describe the mechanisms for collecting feedback on model performance. Explain how this feedback is used to refine the model.*	**Fairness** *Evaluate potential biases in the data and model that could lead to unfair outcomes. Include strategies for identifying, measuring, and mitigating bias across the system.*
Value Proposition *Outline the key benefits and the value the ML solution will bring. Highlight its impact on the business or users.*	**Data Verification and Governance** *Explain the data management policies, focusing on quality, privacy, and compliance. Include mechanisms for data access controls, quality checks, and compliance monitoring.*	**Metrics and Evaluation** *Specify the performance metrics and evaluation methods. Describe how the model's effectiveness will be assessed.*	**Decision** *Explain how the model's predictions are integrated into decision-making. Detail any human oversight or automated decision systems.*	**Lifetime** *Outline the lifetime after model deployment. This includes monitoring for model drift, conditions for retraining, and conditions for decommissioning.*	**Explainability** *Detail how the model's decisions can be interpreted and understood by stakeholders. Include methods to enhance transparency and communicate decision-making processes effectively.*
Objectives *State the specific, measurable goals of the ML project. Detail the expected outcomes and success criteria.*		**Model Governance** *Outline the process for managing models versions including conditions from going from staging to production. Outline procedures for updating and retraining models.*			**Security** *Identify risks related to data breaches, adversarial attacks, and system vulnerabilities. Include measures for safeguarding data and ensuring model robustness against malicious exploitation.*

DTU

By Nicki Skafte Detlefsen nsde@dtu.dk
From DTU course 02476 Machine Learning Operations

License: Apache 2.0

A structural framework for staying organized for large machine learning projects and making sure all the different phases are aligned

https://github.com/SkafteNicki/dtu_mlops/tree/main/canvas

Project checklist

⚠️You do not need to do everything to pass, the list is meant to be exhaustive

Week 1

- [ ] Create a git repository
- [ ] Make sure that all team members have write access to the github repository
- [ ] Create a dedicated environment for you project to keep track of your packages (using conda)
- [ ] Create the initial file structure using cookiecutter
- [ ] Fill out the `make_dataset.py` file such that it downloads whatever data you need and
- [ ] Add a model file and a training script and get that running
- [ ] Remember to fill out the `requirements.txt` file with whatever dependencies that you are using
- [ ] Remember to comply with good coding practices (`pep8`) while doing the project
- [ ] Do a bit of code typing and remember to document essential parts of your code
- [ ] Setup version control for your data or part of your data
- [ ] Construct one or multiple docker files for your code
- [ ] Build the docker files locally and make sure they work as intended
- [ ] Write one or multiple configurations files for your experiments
- [ ] Used Hydra to load the configurations and manage your hyperparameters
- [ ] When you have something that works somewhat, remember at some point to to some profiling and see if you can optimize your code
- [ ] Use wandb to log training progress and other important metrics/artifacts in your code
- [ ] Use pytorch-lightning (if applicable) to reduce the amount of boilerplate in your code

How is the project evaluated?

✅We look at how well you can use the tools and techniques from the material in your project

⚠️We do not look at how good model performance you get

⚠️We do not look at how complex a model and dataset you are using

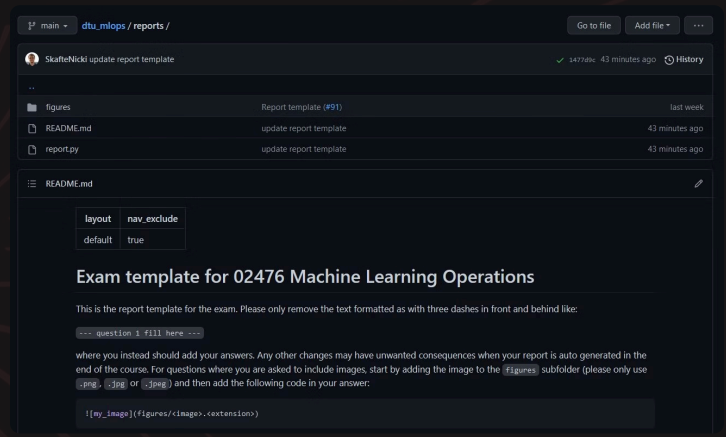
I am specifically looking at

> 💻How well are your code, data, experiments version controlled and reproducible
>
> ♻️Is appropriate continues integration implemented for automatization of tasks
>
> 📦Is a final model deployed online and able to be interacted with a end user
>
> 🤝 How well does it look like you have collaborated on the project

Exam report template

Add this to your public project repository

```
├─── project_repo
│    ├─── src/
│    │    ├─── __init__.py
│    │    ├─── ...
│    ├─── data/
│    │    ├─── raw/
│    │    └─── processed/
│    ├─── ...
│    ├─── reports/
│    │    ├─── figures/  <- for any figures for the report
│    │    ├─── README.md <- YOUR REPORT
│    │    └─── report.py <- helper script
```



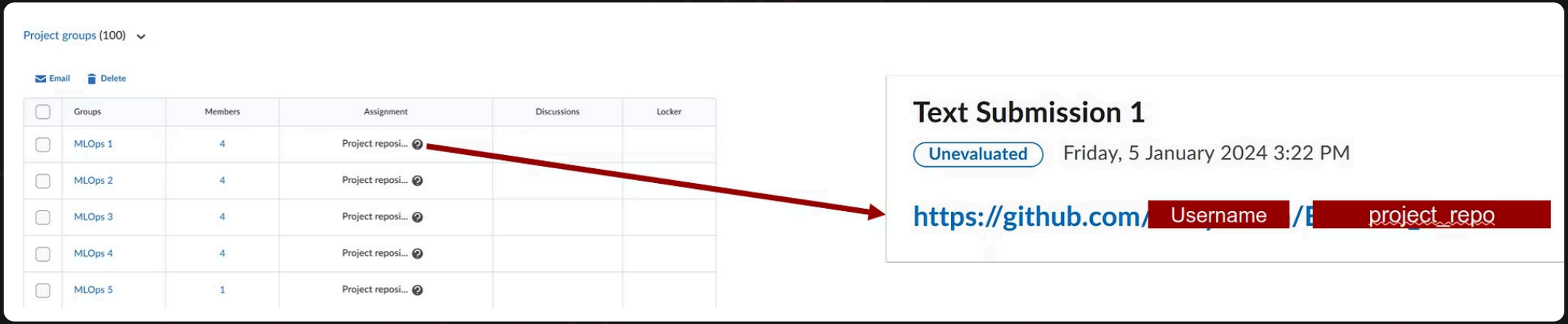
https://github.com/SkafteNicki/dtu_mlops/tree/main/reports

I will scrape you reports and repositories on the 23/1 at 23:59.

Hand-in for today

Should be handed in before midnight today

- If all have access to learn, signup to a group and hand-in
- If only one or more group members are missing from learn, still hand-in as a group and send a email with remaining student ids to me



Meme of the day

https://skaftenicki.github.io/dtu_mlops/pages/projects/