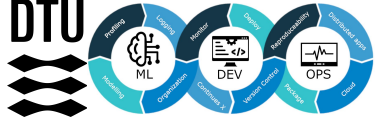
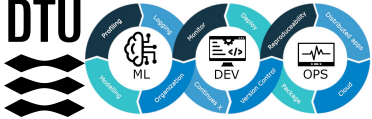


Projects

02476 Machine Learning Operations
Nicki Skafte Detlefsen



The “case”

You are just hired as an MLOps engineer at an start-up.

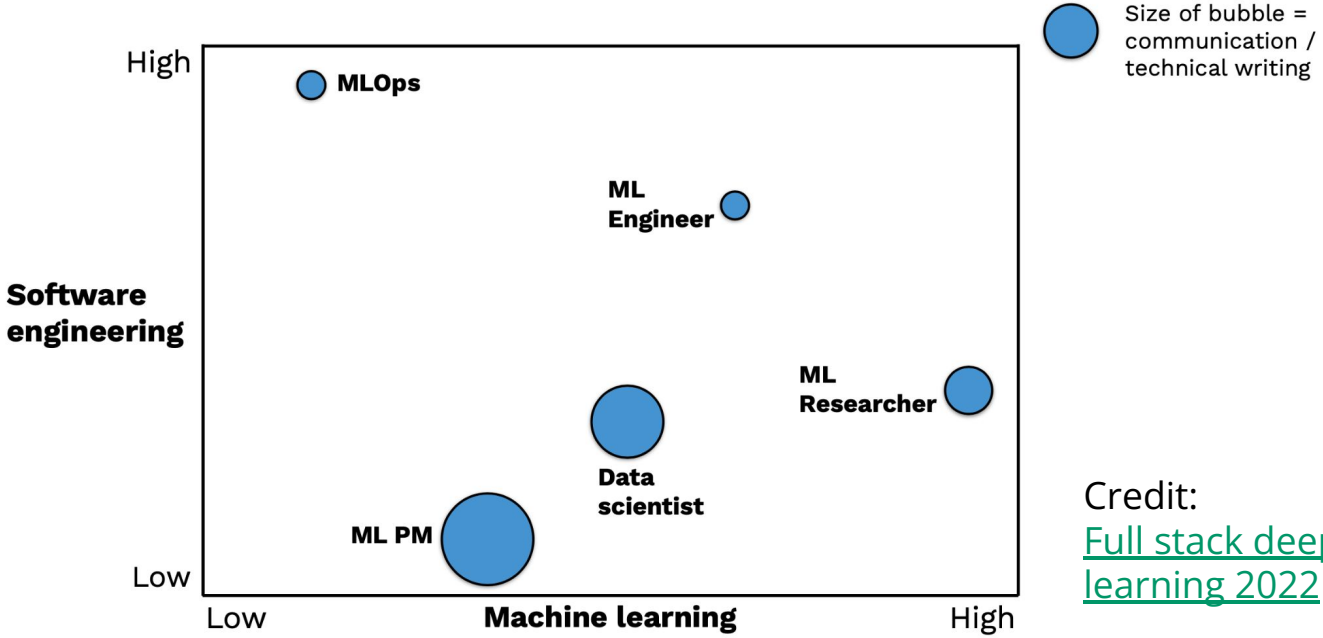
Your first job:

> *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 should not care about the model

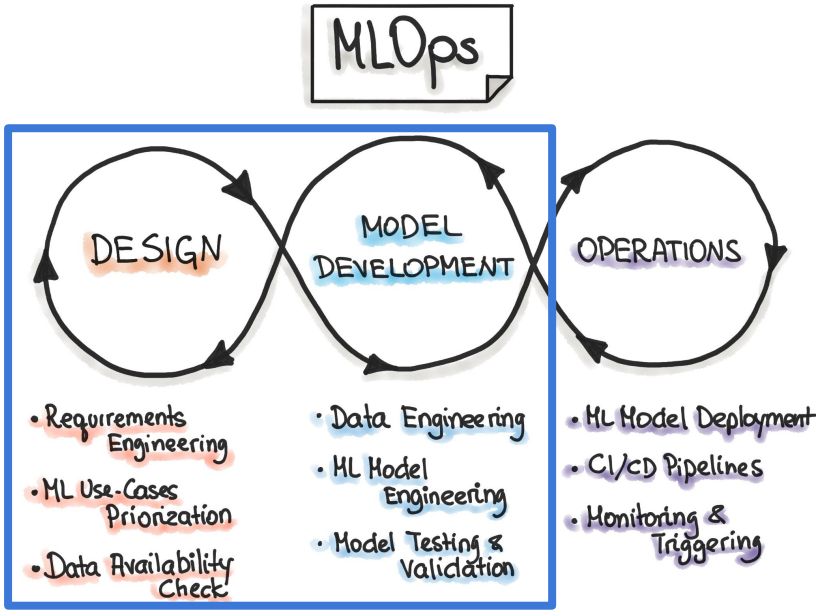
You do not need to focus on the model, that's the ML researchers job



Credit: [Full stack deep learning 2022]

How to solve the problem

- You already have all the tools for the pipeline, you just need a good starting model.
- You base framework is Pytorch
- You turn your attention towards open-source projects build on top of Pytorch

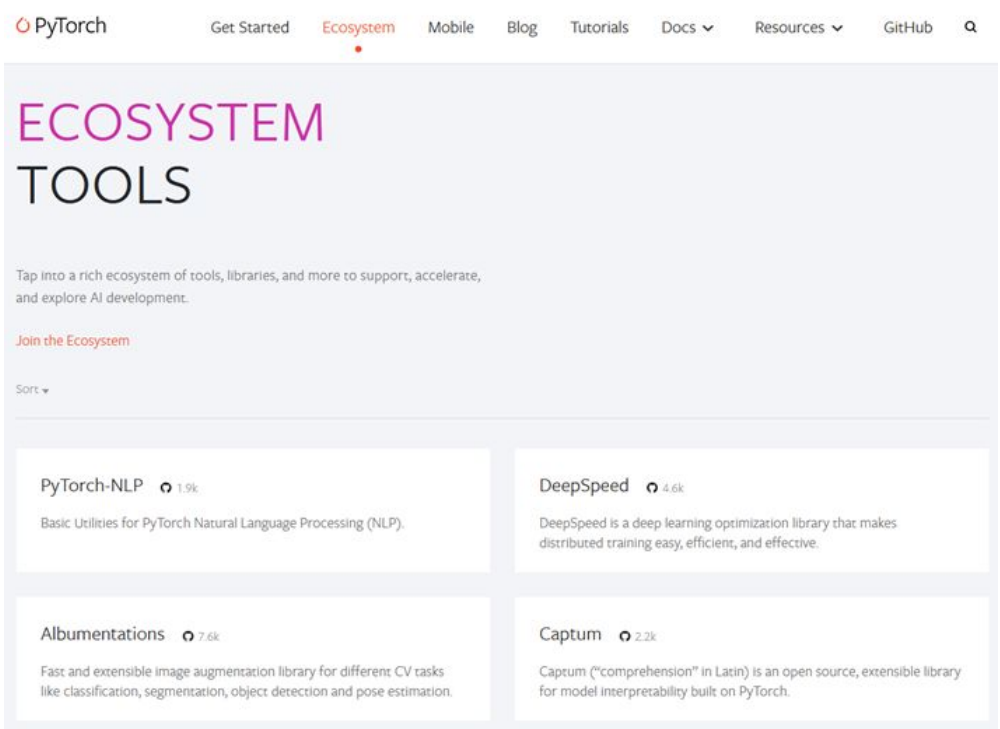


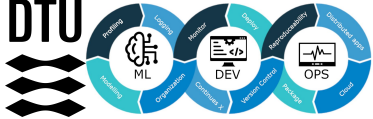
Trying to fast track this part

The Pytorch Ecosystem

Collection of frameworks build to be used in combination with Pytorch

Note: Not a complete list of all frameworks





Framework

My own biased division

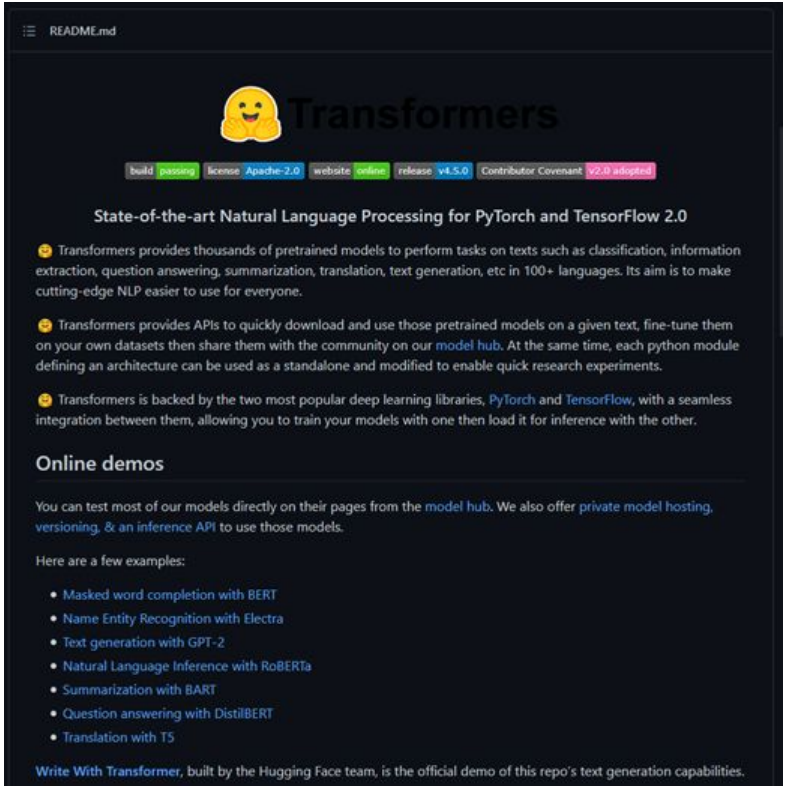
Data specific frameworks	Training frameworks	Utility fremeworks
Transformers	fastai	Albumentations
Detectron2	Ray	PySyft
Pytorch geometric	Pytorch Lightning	Pyro
Flair	Horovod	Optuna
AllenNLP	DeepSpeed	Hydra
ParlAI	ONNX Runtime	Pytorch Metric Learning
DGL	skorch	Einops
PyTorch3D	Ignite	
MMF	Polyaxon	
Kornia		

Project case 1: Natural Language Processing

Framework: Transformers

https://github.com/huggingface/transformers

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

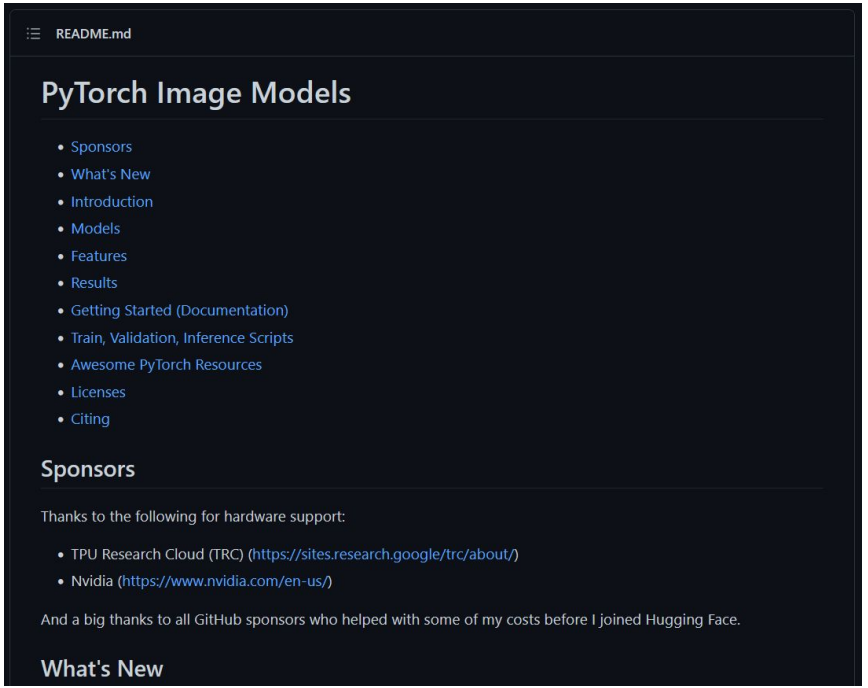


Project case 2: Computer vision

Framework: Pytorch Image Models

https://github.com/rwightman/pytorch-image-models

Also known as TIMM. Image models, scripts, pretrained weights.


README.md

PyTorch Image Models

- Sponsors
- What's New
- Introduction
- Models
- Features
- Results
- Getting Started (Documentation)
- Train, Validation, Inference Scripts
- Awesome PyTorch Resources
- Licenses
- Citing

Sponsors

Thanks to the following for hardware support:

- TPU Research Cloud (TRC) (https://sites.research.google/trc/about/)
- Nvidia (https://www.nvidia.com/en-us/)

And a big thanks to all GitHub sponsors who helped with some of my costs before I joined Hugging Face.

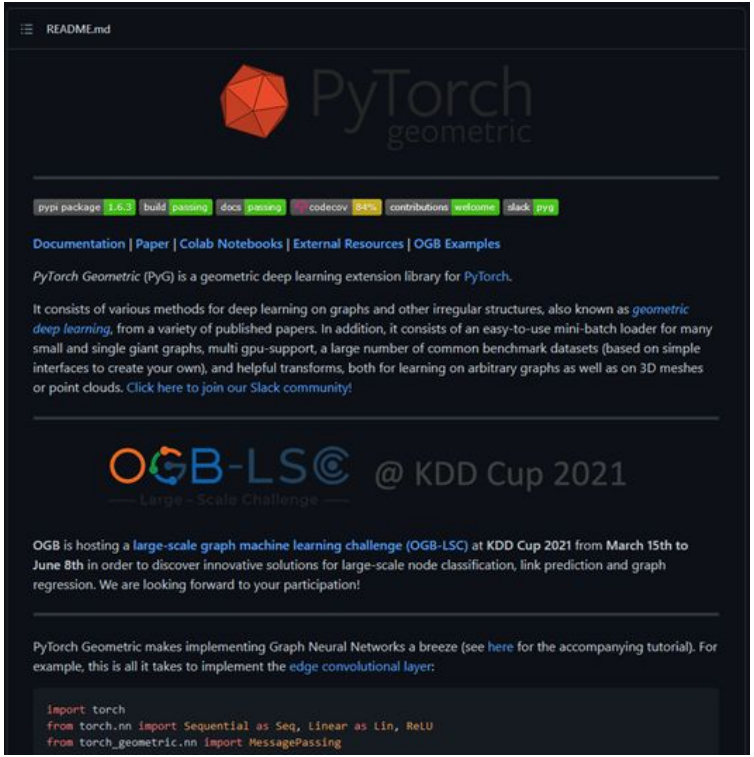
What's New

Project case 3: Graphs and points

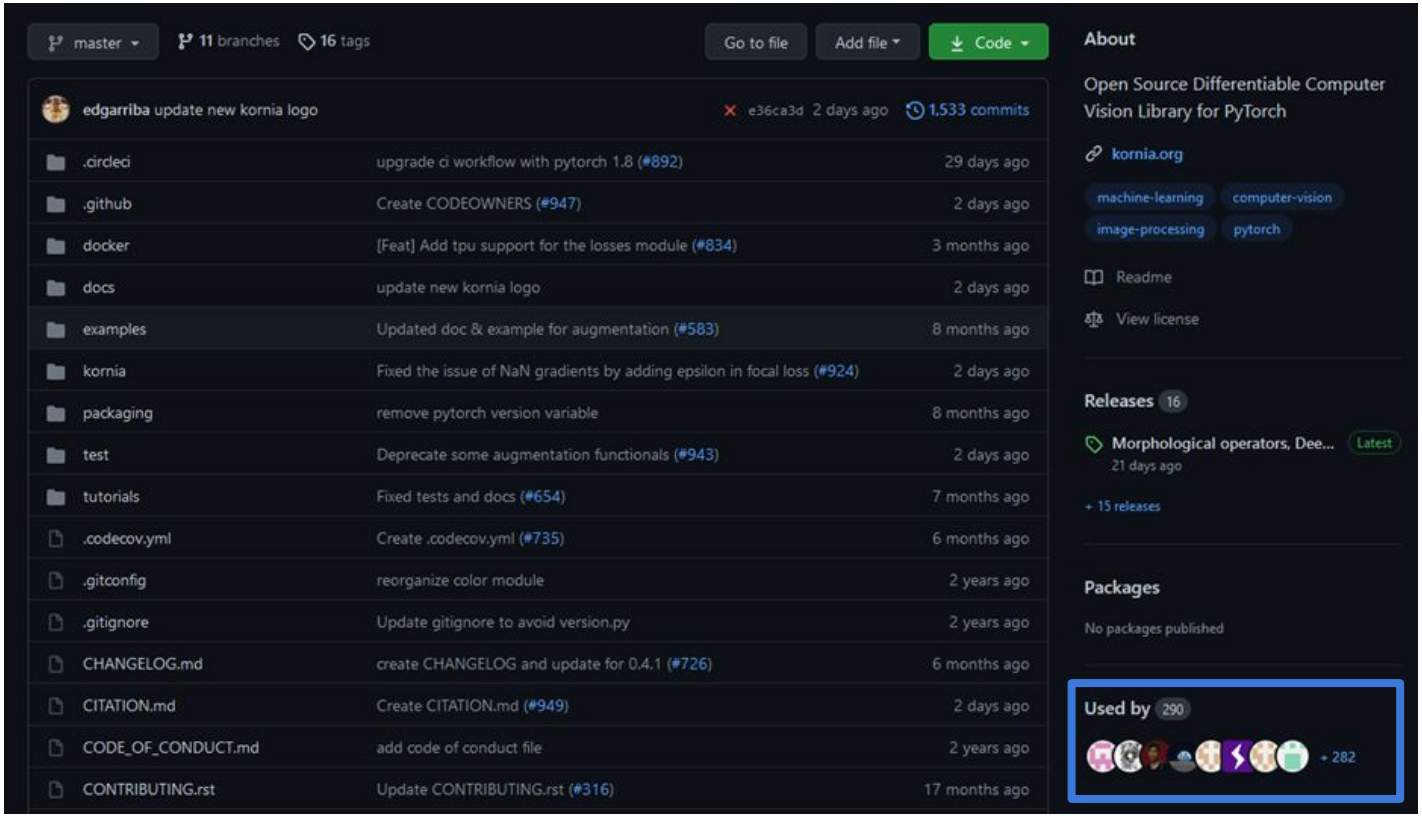
Framework: Pytorch Geometric

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

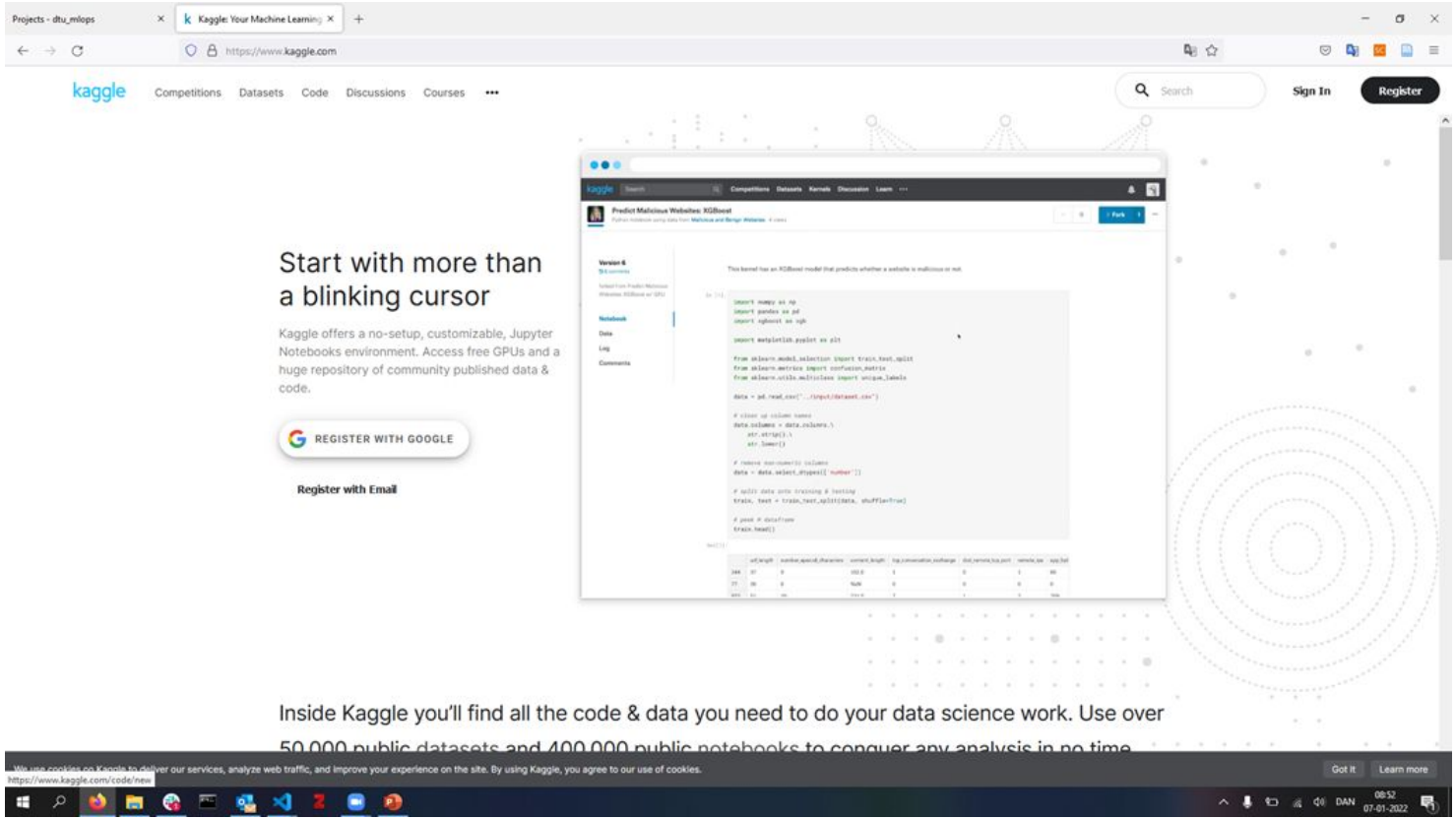
Graph Neural Network Library for PyTorch to work on irregular data such as graphs and points.

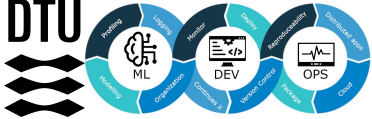


How to get an good idea?



How to get an good idea?





Summary

Pick a framework (try running their notebooks/examples!):

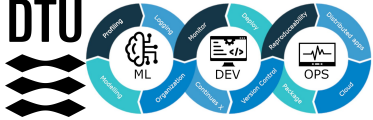
- Project 1: Natural Language Processing
- Project 2: Computer vision
- Project 3: Graphs and points

Brainstorm a project.

- It does not have to be particular big as you only have **4½** full days for working on it

Write a small (max 1 page) project description including:

- What model do intent to implement
- What data are you going to use
- How you think the chosen framework can be incorporated



Checklist

Week 1

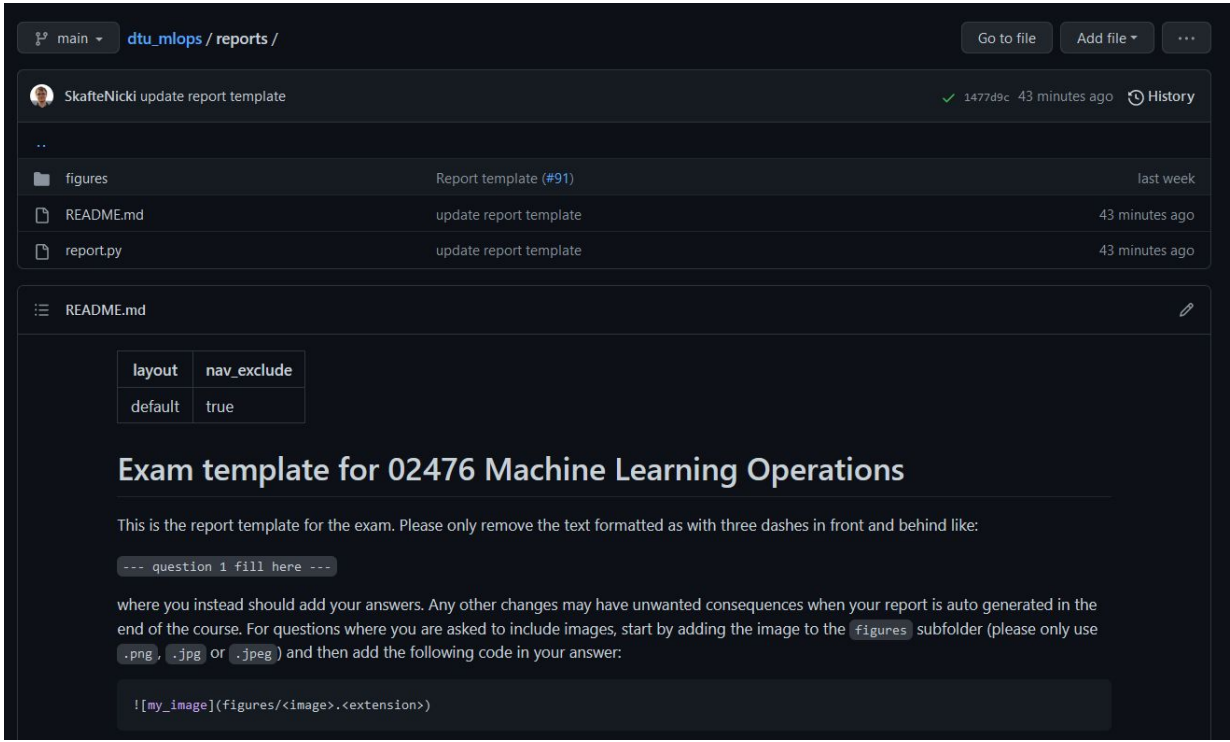
- [ ] Create a git repository
- [ ] Make sure that all team members have write access to the github repository
- [ ] Create a dedicated environment for you project to keep track of your packages (using conda)
- [ ] Create the initial file structure using cookiecutter
- [ ] Fill out the `make_dataset.py` 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

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

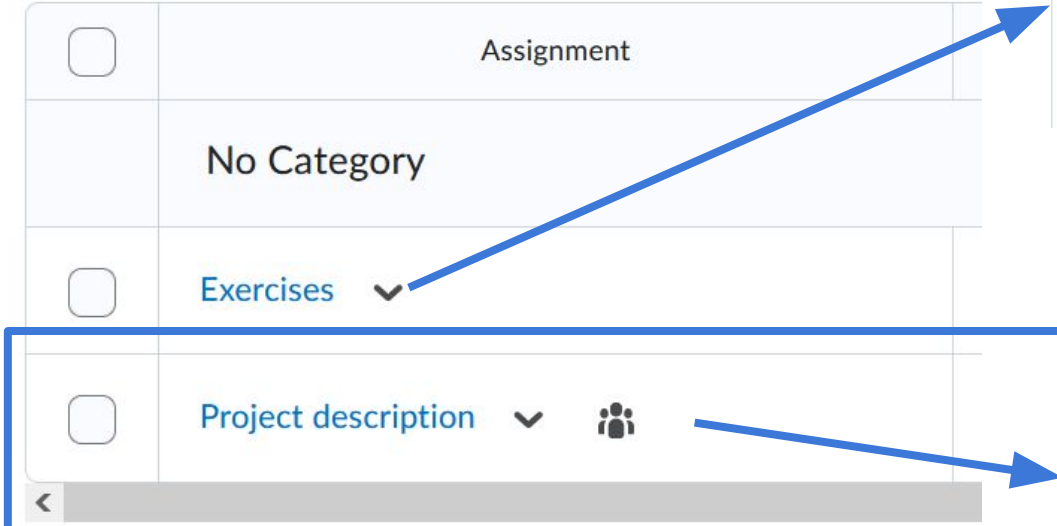
Exam report template

Add this to your **public** project repo.

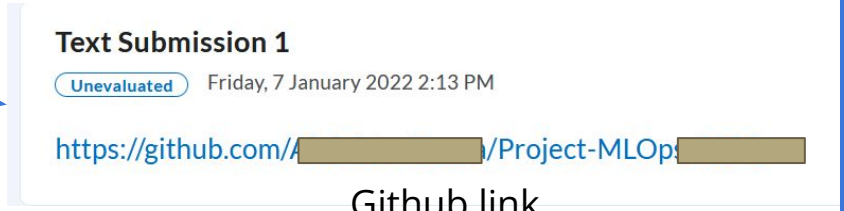
We will scrape automatically at midnight on the 20/1.



Hands-in: Today no later than 17:00



Big zip file with everything + github link



Github link

Meme of the day