The pytorch ecosystem

Machine Learning Operations

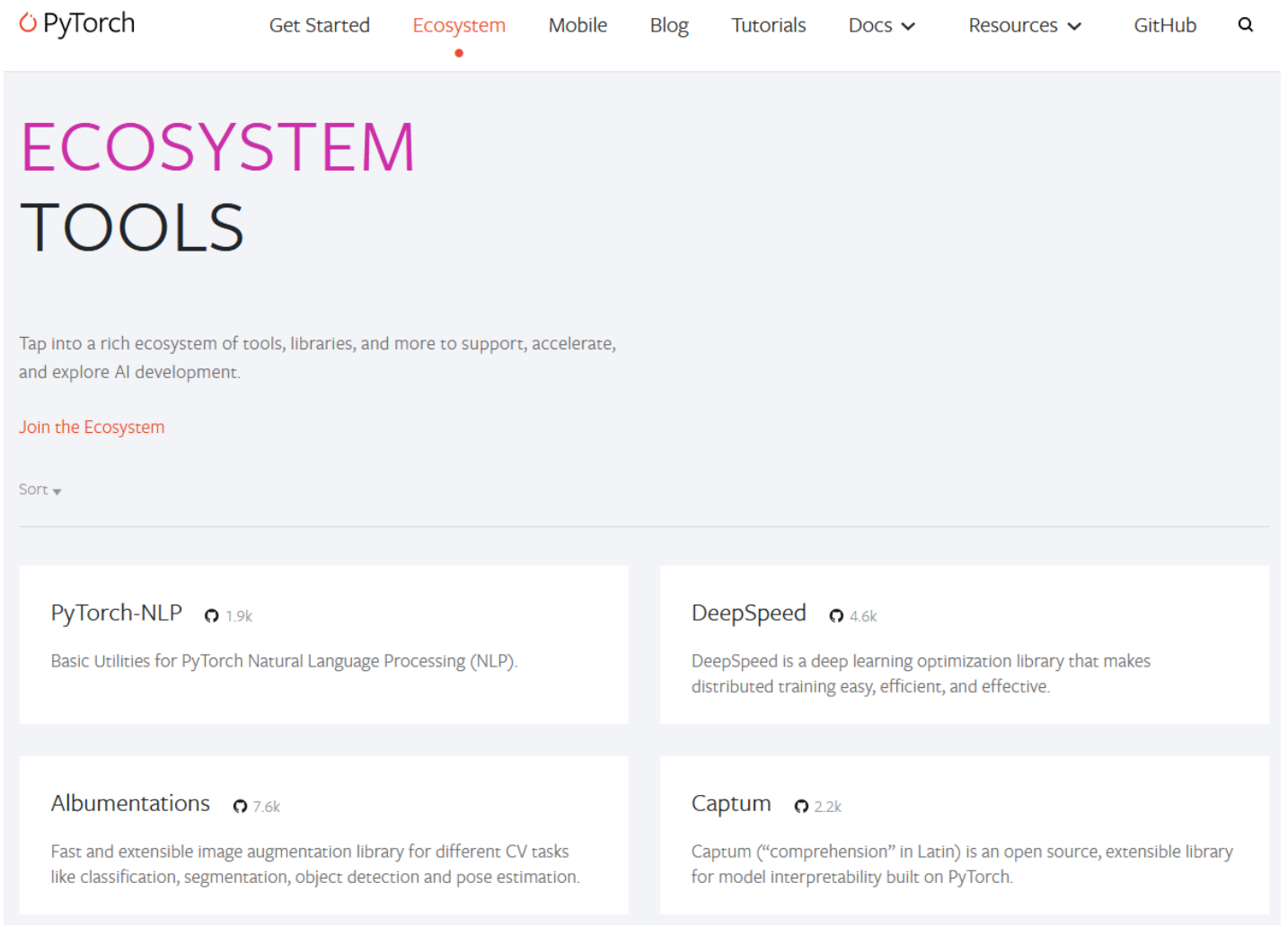
Nicki Skafte Detlefsen,

Postdoc

DTU Compute

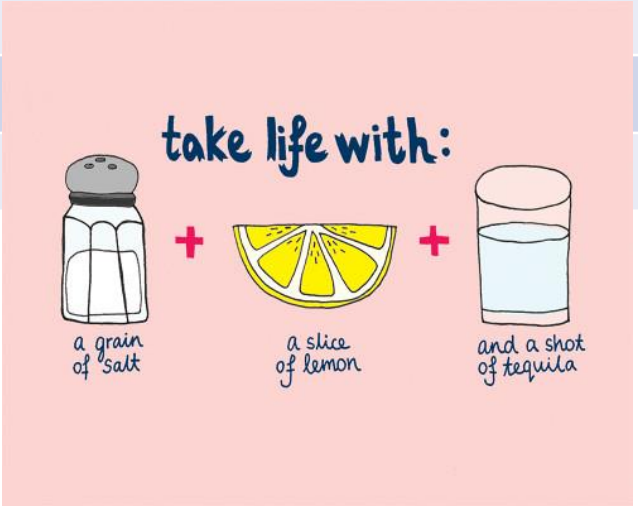
The ecosystem

Collection of frameworks build to be used in combination with Pytorch



Framework categorizing

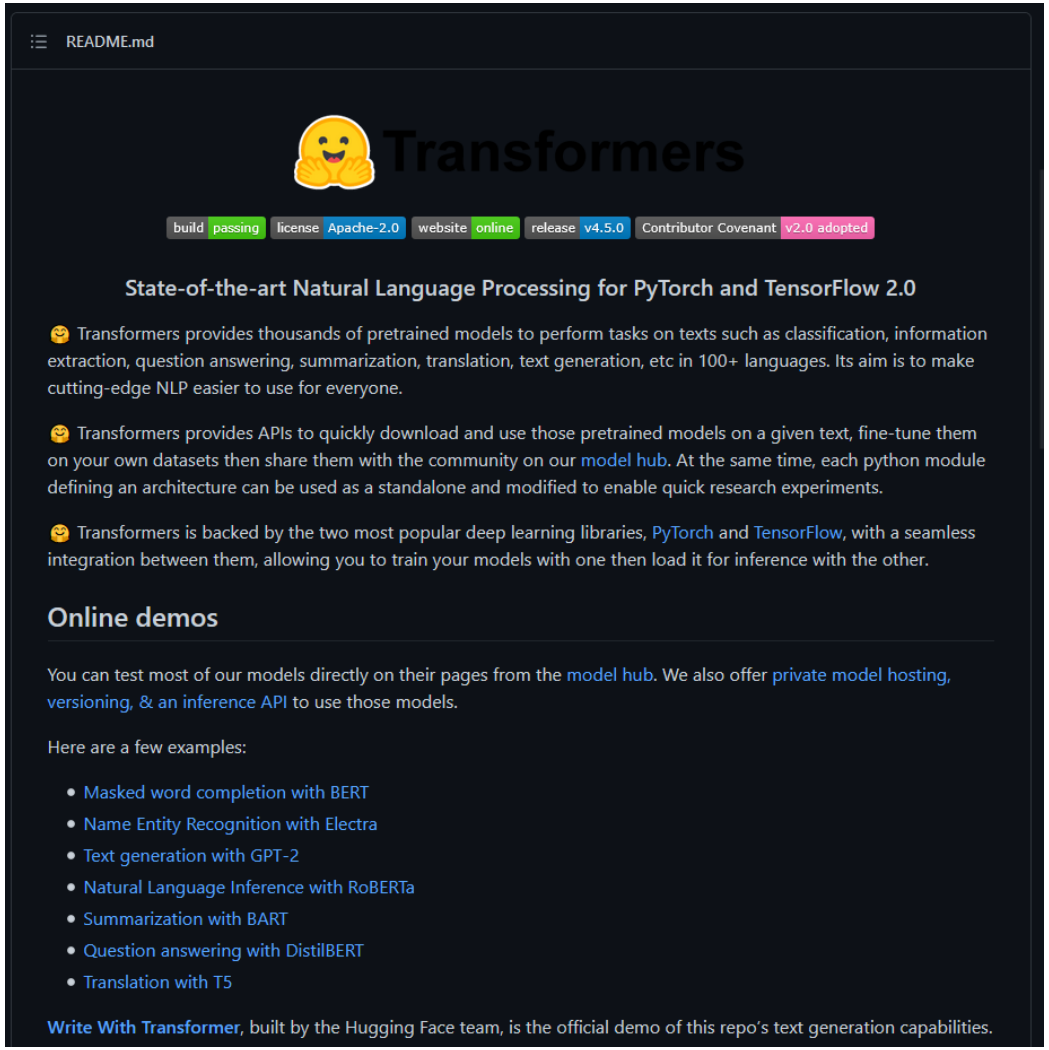
Data specific frameworks	Training frameworks	Utility fremeworks
Transformers	fastai	Albumentations
Detectron2	Ray	PySyft
Pytorch geometric	Pytorch Lightning	Pyro
Flair	Horovod	Optuna
AllenNLP	DeepSpeed	Hydra
ParlAI	ONNX Runtime	Pytorch Metric Learning
DGL	skorch	Einops
PyTorch3D	Ignite	
MMF	Polyaxon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



Project 1: NLP

Framework: Transformers (Huggingface)

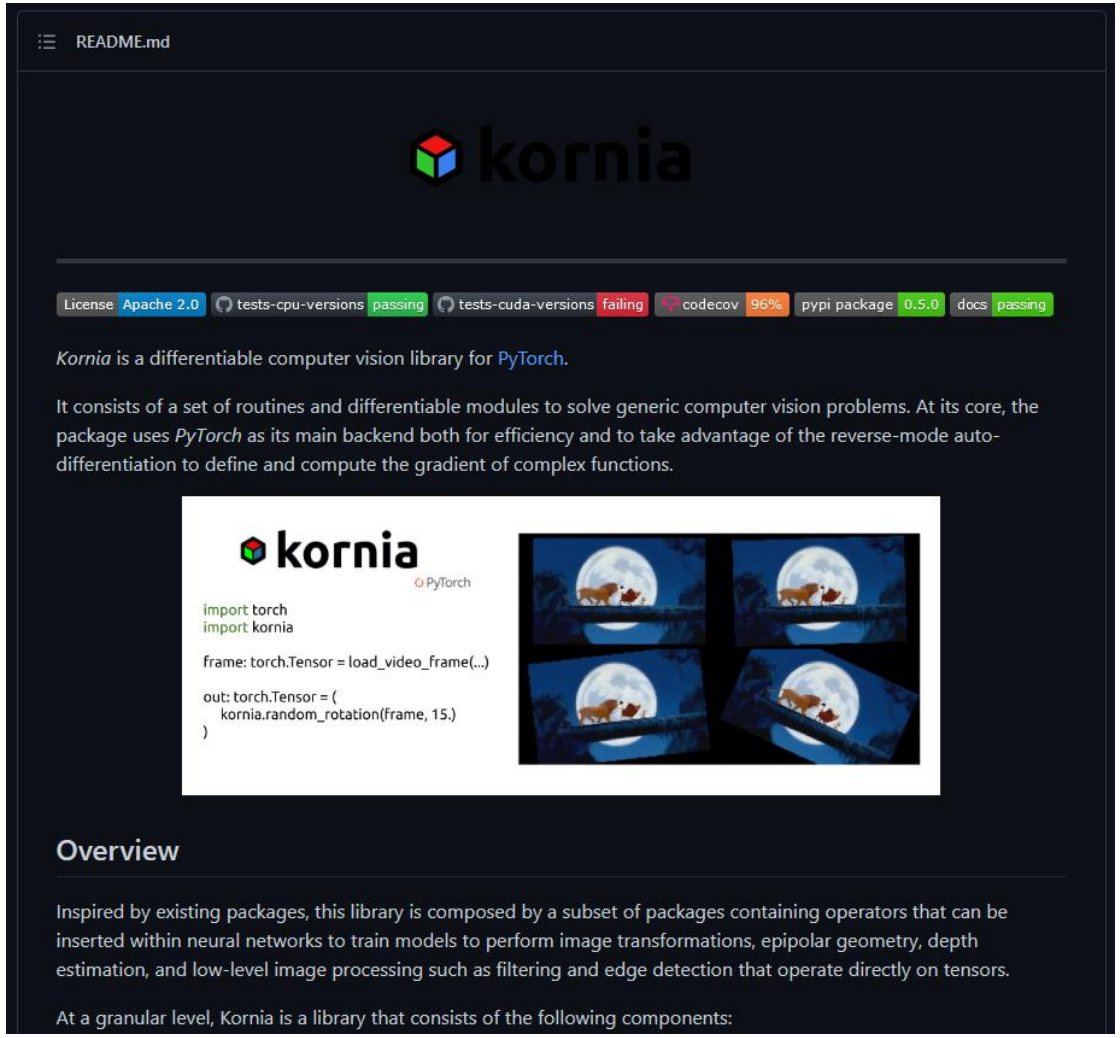
- https://github.com/huggingface/transformers
- State-of-the-art NLP models
- Most starred framework in the ecosystem



Project 2: CV

Framework: Kornia

- https://github.com/kornia/kornia
- Differentiable computer vision algorithms

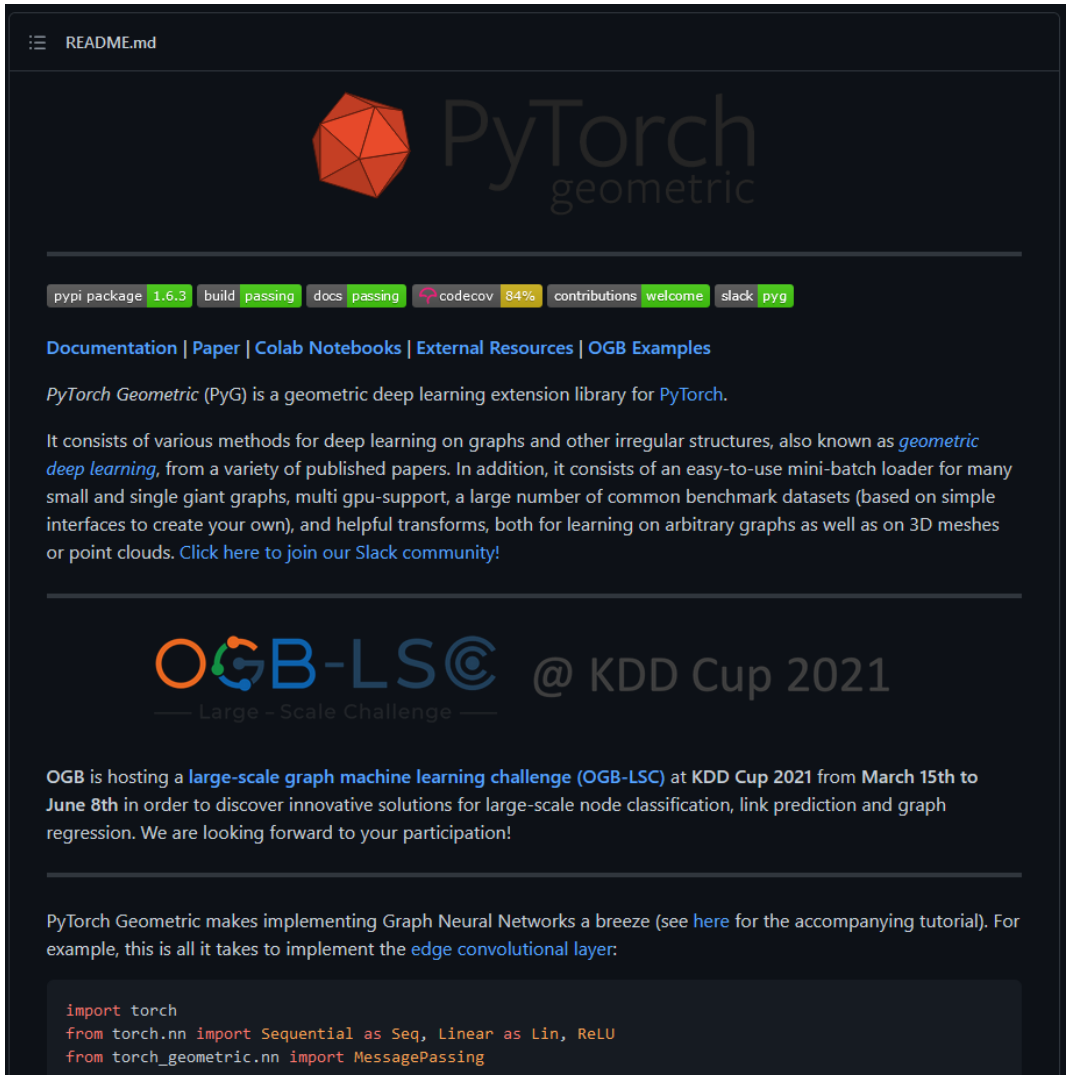


Project 3: Graphs and points

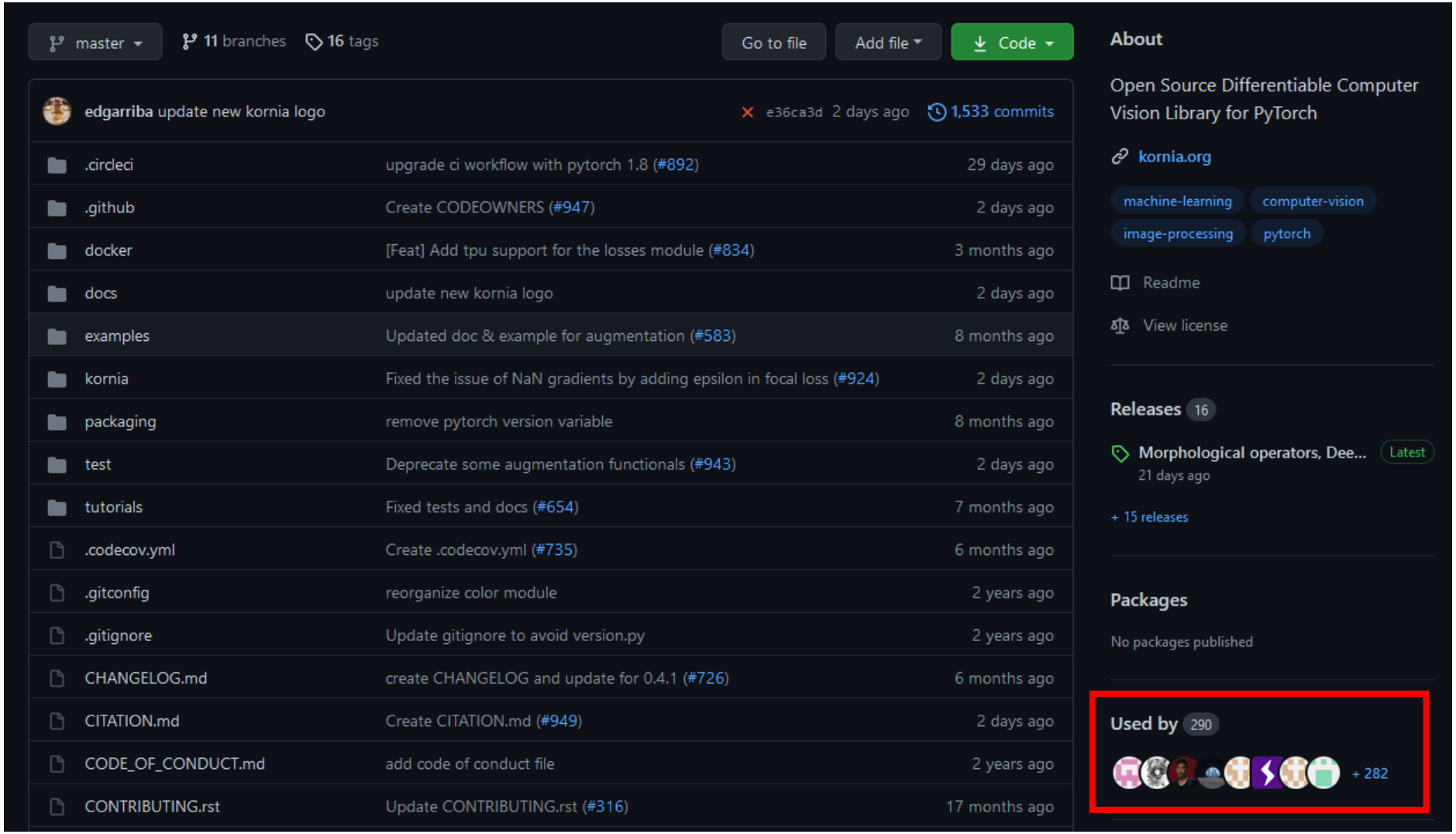
Framework: Pytorch Geometric

- https://github.com/rusty1s/pytorch_geometric

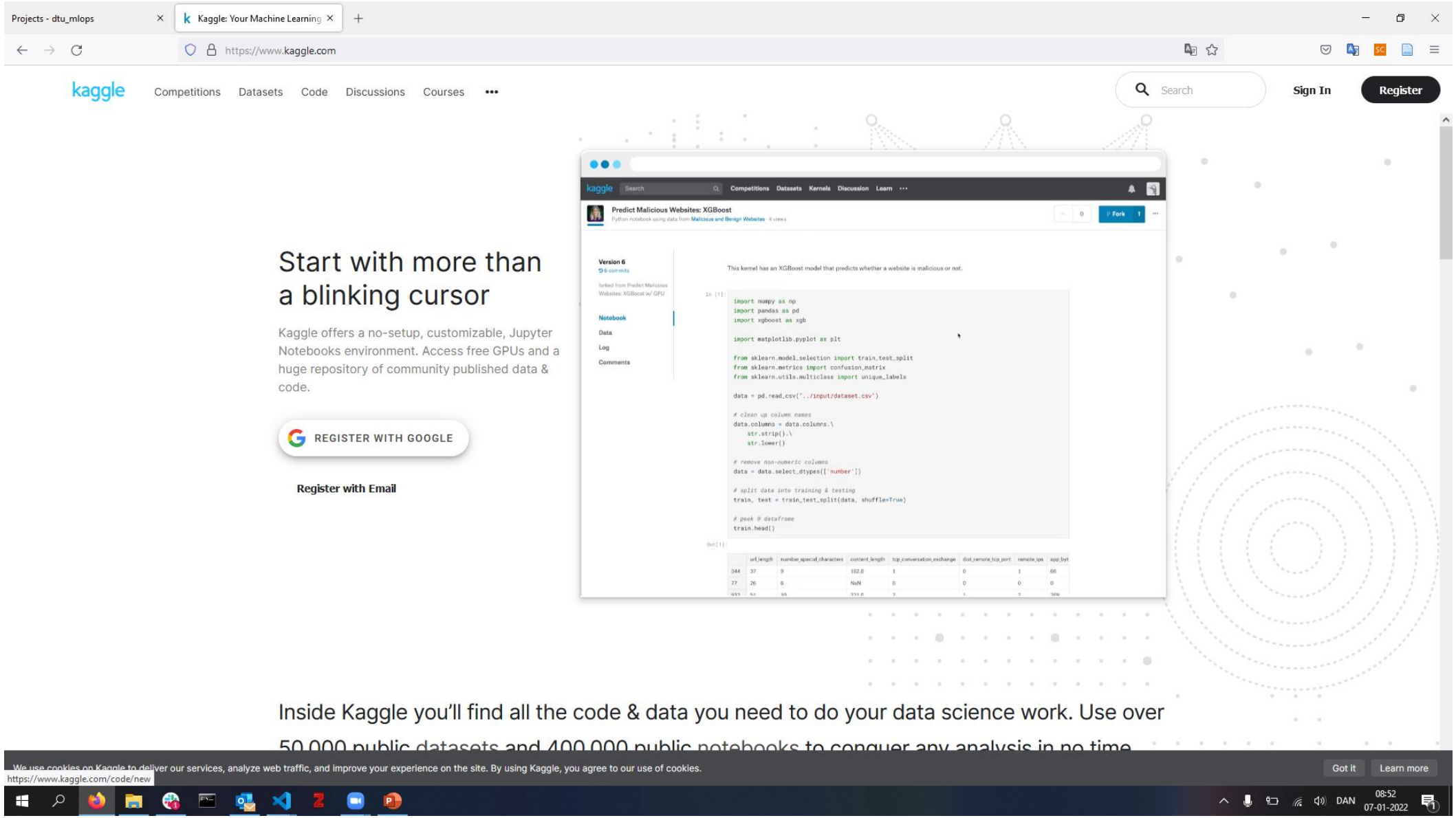
- Neural networks on graphs and point clouds



Getting a good idea



Getting a good idea



Summary

- Pick a framework (try running their notebooks/examples!):
  - Project 1: NLP
  - Project 2: CV
  - 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 (also in todays readme)

- - [ ] 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

Hand-in

- By 17:00 today handin link to github reposatory on DTU Learn

Exam format

Thursday 24/6 – evaluation by Nicki and Søren

- Group presentation
  - 6 minutes of powerpoint showcase
  - 10 minutes of discussion

- What you will be evaluated on:
  - How well you have included what we teach you in the course

- What you will NOT be evaluated on
  - How epic your deep learning model is

Some good advice

1. Document everything
   - Take screenshots of your work

2. Parallize work
   - Many of the checkpoints are independent of each other

Meme of the day