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.

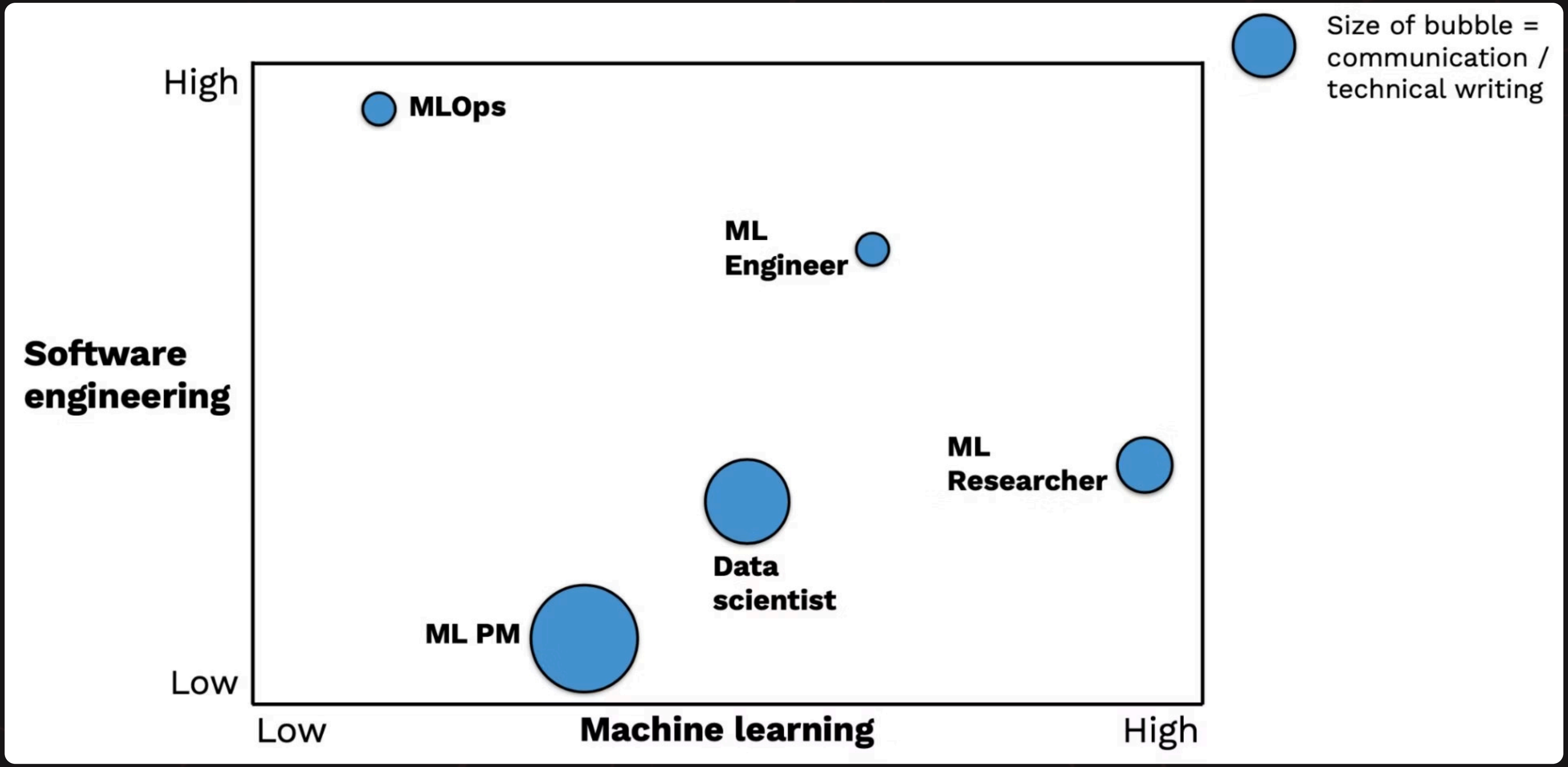
Your first job:

- *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

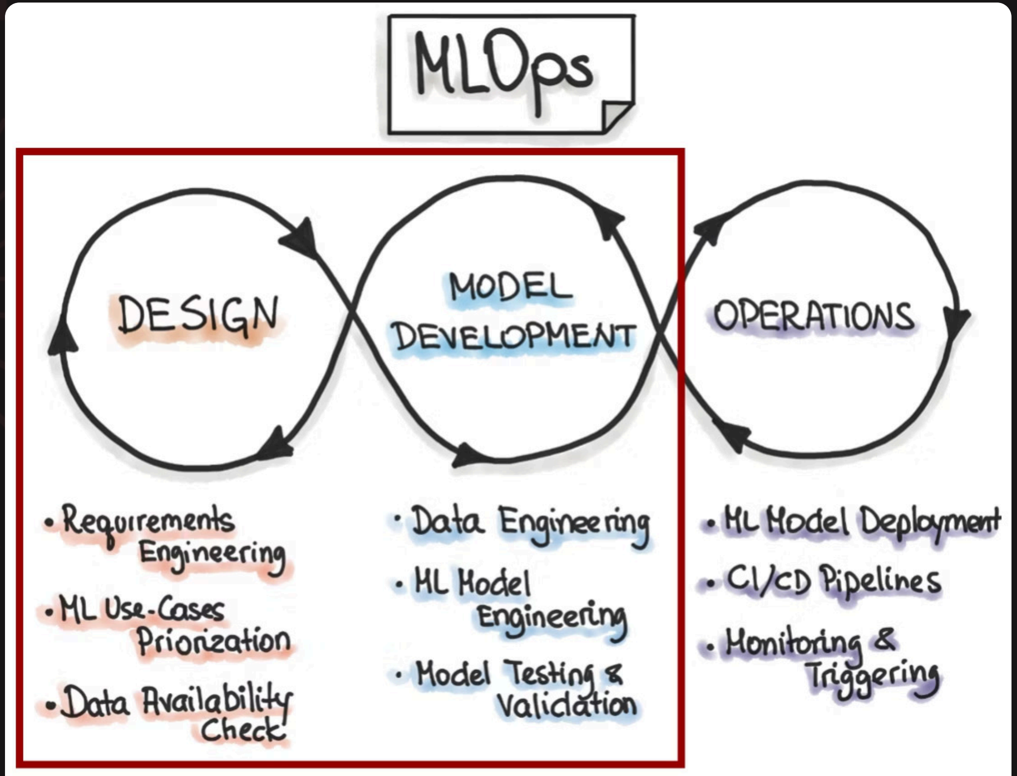


How to solve the problem?

💡 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

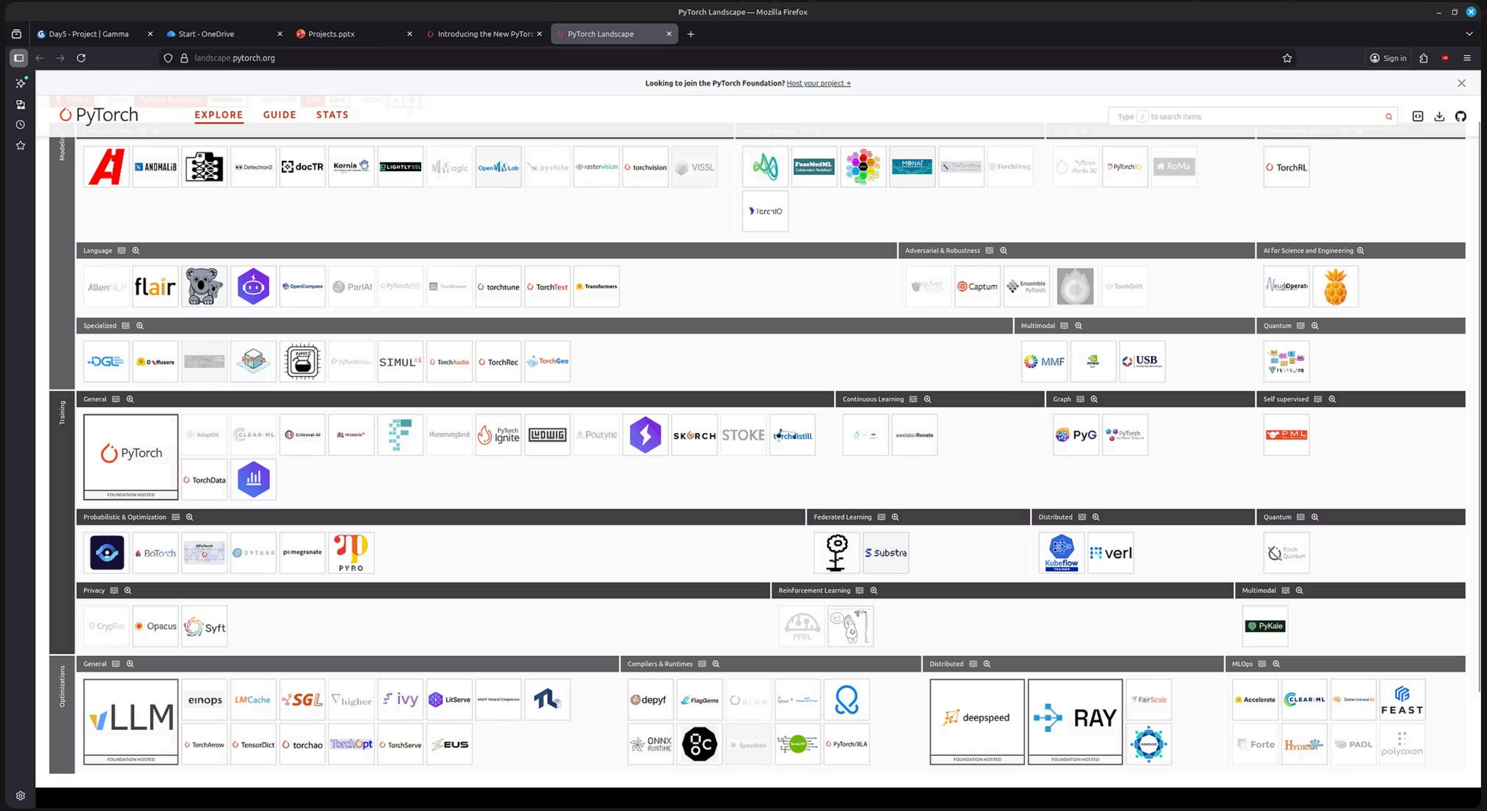

MLOps
DESIGN
MODEL DEVELOPMENT
OPERATIONS
• Requirements Engineering
• ML Use-Cases Priorization
• Data Availability Check
• Data Engineering
• ML Model Engineering
• Model Testing & Validation
• ML Model Deployment
• CI/CD Pipelines
• Monitoring & Triggering

Fast track this part

The Pytorch Landscape♻️

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

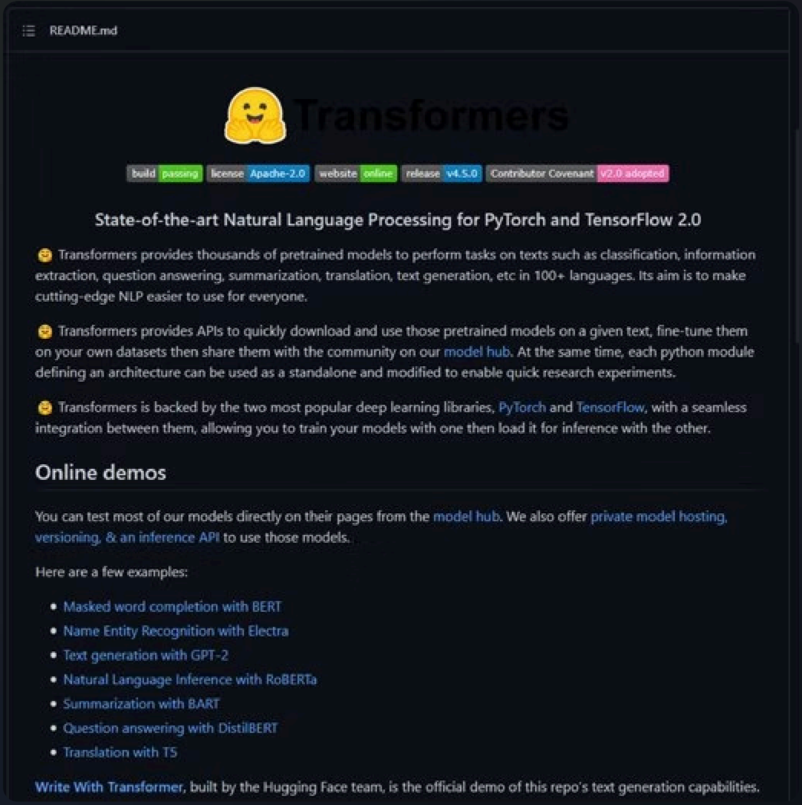
💡It is not a complete list of all great frameworks



Example 1: Transformers

https://github.com/huggingface/transformers

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



README.md

🤗 Transformers

build passing | license Apache-2.0 | website online | release v4.5.0 | Contributor Covenant v2.0 adopted

State-of-the-art Natural Language Processing for PyTorch and TensorFlow 2.0

🤗 Transformers provides thousands of pretrained models to perform tasks on texts such as classification, information extraction, question answering, summarization, translation, text generation, etc in 100+ languages. Its aim is to make cutting-edge NLP easier to use for everyone.

🤗 Transformers provides APIs to quickly download and use those pretrained models on a given text, fine-tune them on your own datasets then share them with the community on our model hub. At the same time, each python module defining an architecture can be used as a standalone and modified to enable quick research experiments.

🤗 Transformers is backed by the two most popular deep learning libraries, PyTorch and TensorFlow, with a seamless integration between them, allowing you to train your models with one then load it for inference with the other.

Online demos

You can test most of our models directly on their pages from the model hub. We also offer private model hosting, versioning, & an inference API to use those models.

Here are a few examples:

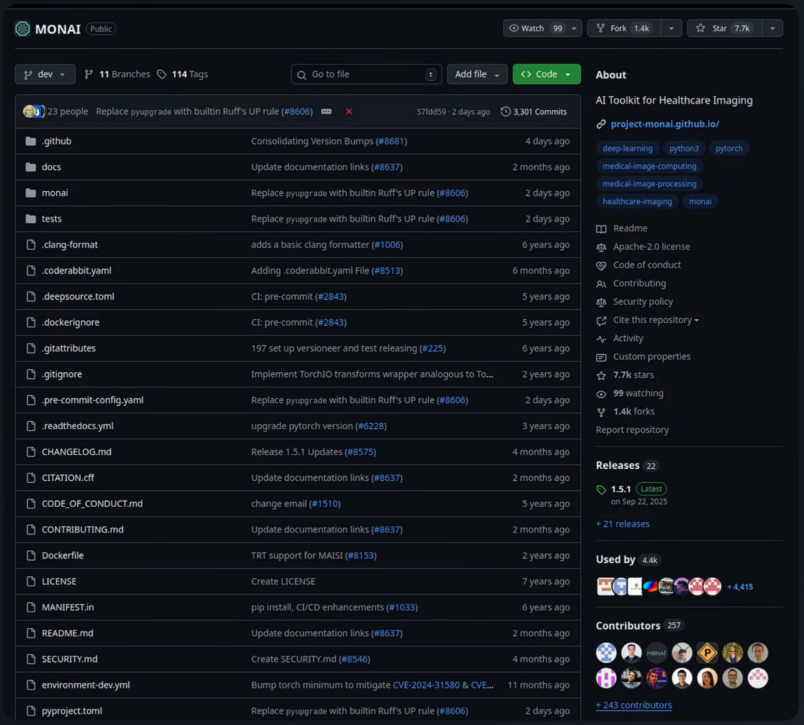
- Masked word completion with BERT
- Name Entity Recognition with Electra
- Text generation with GPT-2
- Natural Language Inference with RoBERTa
- Summarization with BART
- Question answering with DistilBERT
- Translation with T5

Write With Transformer, built by the Hugging Face team, is the official demo of this repo's text generation capabilities.

Example 2: Monai

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

Models for healtcare imaging



23 people Replace pyupgrade with builtin Ruff's UP rule (#8606) 57fdd59 · 2 days ago 3,301 Commits

.github	Consolidating Version Bumps (#8681)	4 days ago
docs	Update documentation links (#8637)	2 months ago
monai	Replace pyupgrade with builtin Ruff's UP rule (#8606)	2 days ago
tests	Replace pyupgrade with builtin Ruff's UP rule (#8606)	2 days ago
.clang-format	adds a basic clang formatter (#1006)	6 years ago
.coderabbit.yaml	Adding .coderabbit.yaml File (#8513)	6 months ago
.deepsource.toml	CI: pre-commit (#2843)	5 years ago
.dockerignore	CI: pre-commit (#2843)	5 years ago
.gitattributes	197 set up versioneer and test releasing (#225)	6 years ago
.gitignore	Implement TorchIO transforms wrapper analogous to To...	2 years ago
.pre-commit-config.yaml	Replace pyupgrade with builtin Ruff's UP rule (#8606)	2 days ago
.readthedocs.yml	upgrade pytorch version (#6228)	3 years ago
CHANGELOG.md	Release 1.5.1 Updates (#8575)	4 months ago
CITATION.cff	Update documentation links (#8637)	2 months ago
CODE_OF_CONDUCT.md	change email (#1510)	5 years ago
CONTRIBUTING.md	Update documentation links (#8637)	2 months ago
Dockerfile	TRT support for MAISI (#8153)	2 years ago
LICENSE	Create LICENSE	7 years ago
MANIFEST.in	pip install, CI/CD enhancements (#1033)	6 years ago
README.md	Update documentation links (#8637)	2 months ago
SECURITY.md	Create SECURITY.md (#8546)	4 months ago
environment-dev.yml	Bump torch minimum to mitigate CVE-2024-31580 & CVE...	11 months ago
pyproject.toml	Replace pyupgrade with builtin Ruff's UP rule (#8606)	2 days ago

About

AI Toolkit for Healthcare Imaging

project-monai.github.io/

deep-learning python3 pytorch medical-image-computing medical-image-processing healthcare-imaging monai

Readme
Apache-2.0 license
Code of conduct
Contributing
Security policy
Cite this repository
Activity
Custom properties
7.7k stars
99 watching
1.4k forks
Report repository

Releases 22

1.5.1 Latest
on Sep 22, 2025

+ 21 releases

Used by 4.4k

+ 4,415

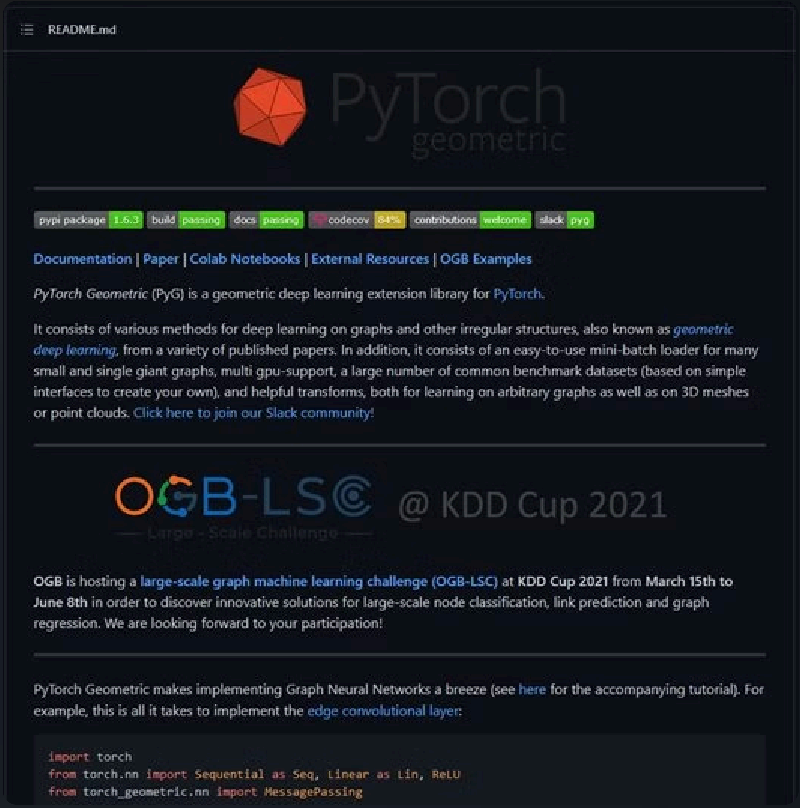
Contributors 257

+ 243 contributors

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.



README.md

Documentation | Paper | Colab Notebooks | External Resources | OGB Examples

PyTorch Geometric (PyG) is a geometric deep learning extension library for PyTorch.

It consists of various methods for deep learning on graphs and other irregular structures, also known as *geometric deep learning*, from a variety of published papers. In addition, it consists of an easy-to-use mini-batch loader for many small and single giant graphs, multi gpu-support, a large number of common benchmark datasets (based on simple interfaces to create your own), and helpful transforms, both for learning on arbitrary graphs as well as on 3D meshes or point clouds. Click here to join our Slack community!

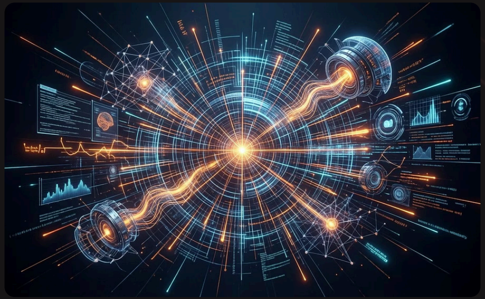
OGB is hosting a large-scale graph machine learning challenge (OGB-LSC) at **KDD Cup 2021** from **March 15th to June 8th** in order to discover innovative solutions for large-scale node classification, link prediction and graph regression. We are looking forward to your participation!

PyTorch Geometric makes implementing Graph Neural Networks a breeze (see here for the accompanying tutorial). For example, this is all it takes to implement the edge convolutional layer:

```
import torch
from torch.nn import Sequential as Seq, Linear as Lin, ReLU
from torch_geometric.nn import MessagePassing
```

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

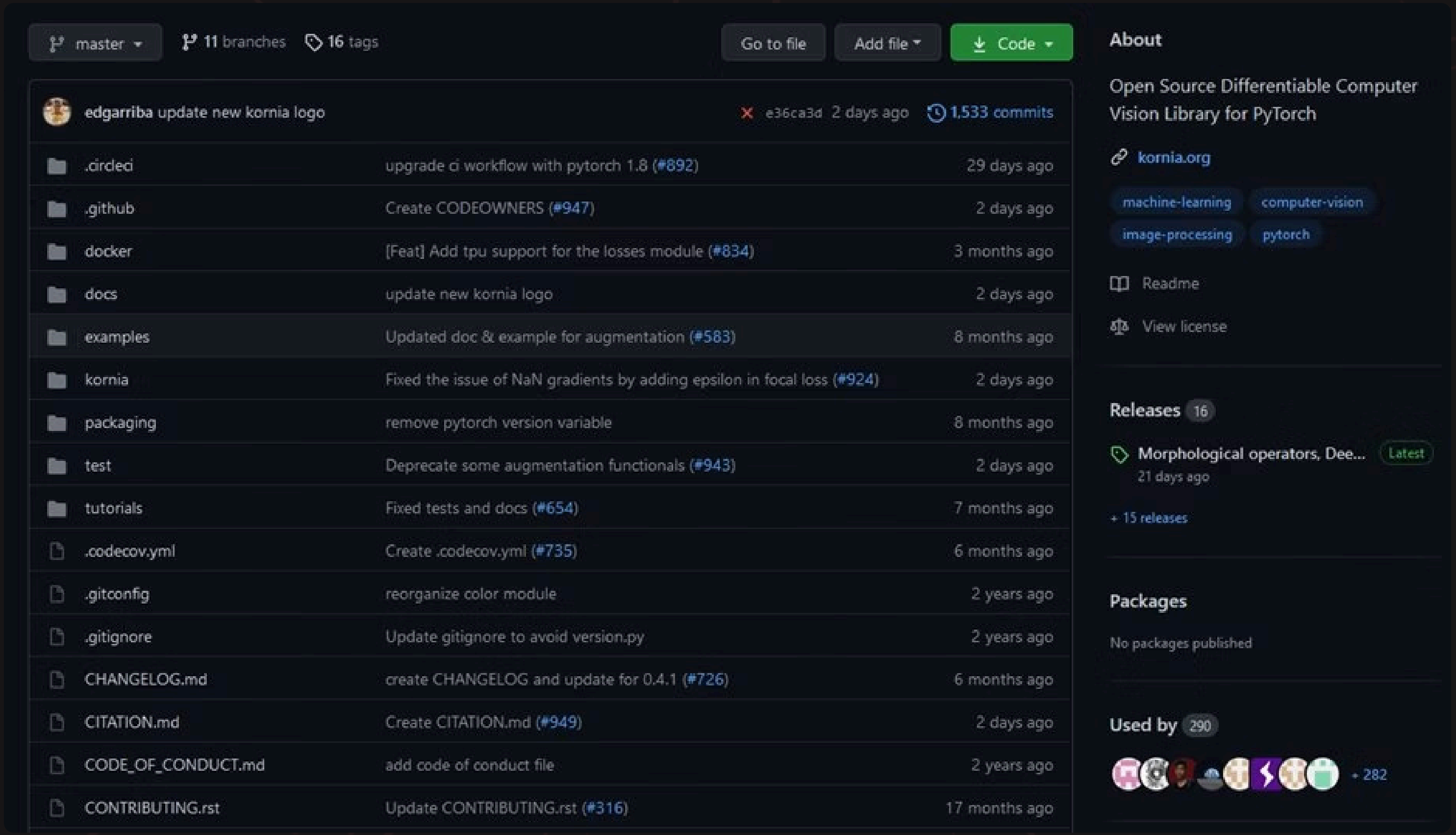
Select a suitable machine learning or deep learning model. Explore various architectures and their applications for your chosen dataset.

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



edgarriba update new kornia logo ✕ e36ca3d 2 days ago 1,533 commits

.circleci	upgrade ci workflow with pytorch 1.8 (#892)	29 days ago
.github	Create CODEOWNERS (#947)	2 days ago
docker	[Feat] Add tpu support for the losses module (#834)	3 months ago
docs	update new kornia logo	2 days ago
examples	Updated doc & example for augmentation (#583)	8 months ago
kornia	Fixed the issue of NaN gradients by adding epsilon in focal loss (#924)	2 days ago
packaging	remove pytorch version variable	8 months ago
test	Deprecate some augmentation functionals (#943)	2 days ago
tutorials	Fixed tests and docs (#654)	7 months ago
.codecov.yml	Create .codecov.yml (#735)	6 months ago
.gitconfig	reorganize color module	2 years ago
.gitignore	Update gitignore to avoid version.py	2 years ago
CHANGELOG.md	create CHANGELOG and update for 0.4.1 (#726)	6 months ago
CITATION.md	Create CITATION.md (#949)	2 days ago
CODE_OF_CONDUCT.md	add code of conduct file	2 years ago
CONTRIBUTING.rst	Update CONTRIBUTING.rst (#316)	17 months ago

About

Open Source Differentiable Computer Vision Library for PyTorch

kornia.org

machine-learning computer-vision image-processing pytorch

Readme

View license

Releases 16

Morphological operators, Dee... Latest
21 days ago

+ 15 releases

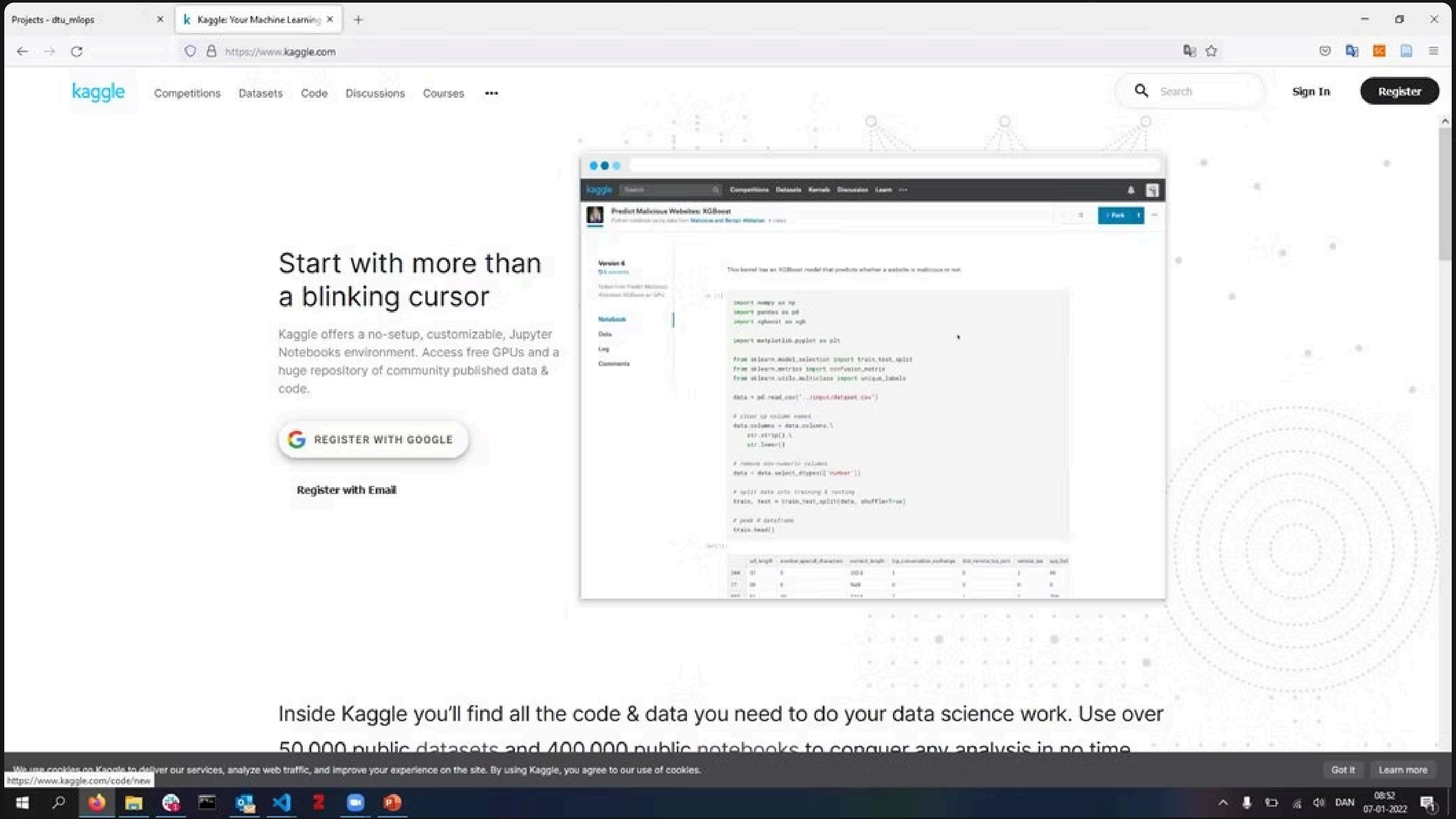
Packages

No packages published

Used by 290

+ 282

How to get an good idea?



Start with more than a blinking cursor

Kaggle offers a no-setup, customizable, Jupyter Notebooks environment. Access free GPUs and a huge repository of community published data & code.

Predict Malicious Websites: XGBoost

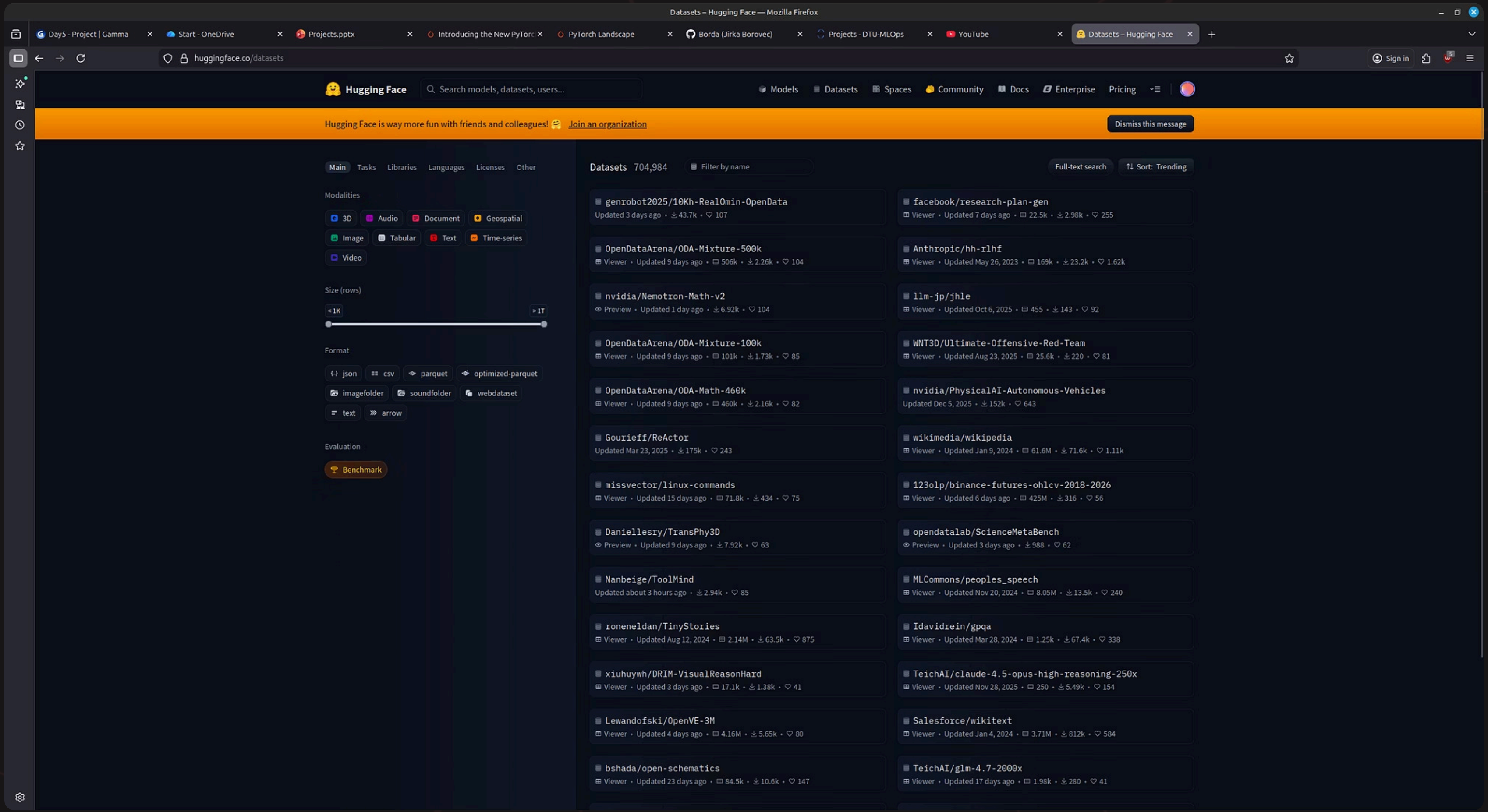
This kernel has an XGBoost model that predicts whether a website is malicious or not.

```
import numpy as np
import pandas as pd
import xgboost as xgb

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels

data = pd.read_csv('../input/dataset.csv')

# clean up column names
data.columns = data.columns.\
    str.strip().\
    str.lower()

# remove non-numeric columns
data = data.select_dtypes(['number'])

# split data into training & testing
train, test = train_test_split(data, shuffle=True)

# peek @ dataframe
train.head()
```

Inside Kaggle you'll find all the code & data you need to do your data science work. Use over 50,000 public datasets and 400,000 public notebooks to conquer any analysis in no time.

We use cookies on Kaggle to deliver our services, analyze web traffic, and improve your experience on the site. By using Kaggle, you agree to our use of cookies.

How to get an good idea?



Hugging Face is way more fun with friends and colleagues! 🤗 Join an organization

Datasets 704,984

genrobot2025/10Kh-RealOmin-OpenData
Updated 3 days ago • 43.7k • ♡ 107

OpenDataArena/ODA-Mixture-500k
Viewer • Updated 9 days ago • 506k • 2.26k • ♡ 104

nvidia/Nemotron-Math-v2
Preview • Updated 1 day ago • 6.92k • ♡ 104

OpenDataArena/ODA-Mixture-100k
Viewer • Updated 9 days ago • 101k • 1.73k • ♡ 85

OpenDataArena/ODA-Math-460k
Viewer • Updated 9 days ago • 460k • 2.16k • ♡ 82

Gourieff/ReActor
Updated Mar 23, 2025 • 175k • ♡ 243

missvector/linux-commands
Viewer • Updated 15 days ago • 71.8k • 434 • ♡ 75

Daniellesry/TransPhy3D
Preview • Updated 9 days ago • 7.92k • ♡ 63

Nanbeige/ToolMind
Updated about 3 hours ago • 2.94k • ♡ 85

roneneldan/TinyStories
Viewer • Updated Aug 12, 2024 • 2.14M • 63.5k • ♡ 875

xiuhuywh/DRIM-VisualReasonHard
Viewer • Updated 3 days ago • 17.1k • 1.38k • ♡ 41

Lewandofski/OpenVE-3M
Viewer • Updated 4 days ago • 4.16M • 5.65k • ♡ 80

bshada/open-schematics
Viewer • Updated 23 days ago • 84.5k • 10.6k • ♡ 147

facebook/research-plan-gen
Viewer • Updated 7 days ago • 22.5k • 2.98k • ♡ 255

Anthropic/hh-rlhf
Viewer • Updated May 26, 2023 • 169k • 23.2k • ♡ 1.62k

llm-jp/jhle
Viewer • Updated Oct 6, 2025 • 455 • 143 • ♡ 92

WNT3D/Ultimate-Offensive-Red-Team
Viewer • Updated Aug 23, 2025 • 25.6k • 220 • ♡ 81

nvidia/PhysicalAI-Autonomous-Vehicles
Updated Dec 5, 2025 • 152k • ♡ 643

wikimedia/wikipedia
Viewer • Updated Jan 9, 2024 • 61.6M • 71.6k • ♡ 1.11k

123olp/binance-futures-ohlcv-2018-2026
Viewer • Updated 6 days ago • 425M • 316 • ♡ 56

opendatalab/ScienceMetaBench
Preview • Updated 3 days ago • 988 • ♡ 62

MLCommons/peoples_speech
Viewer • Updated Nov 20, 2024 • 8.05M • 13.5k • ♡ 240

Idavidrein/gpqa
Viewer • Updated Mar 28, 2024 • 1.25k • 67.4k • ♡ 338

TeichAI/claude-4.5-opus-high-reasoning-250x
Viewer • Updated Nov 28, 2025 • 250 • 5.49k • ♡ 154

Salesforce/wikitext
Viewer • Updated Jan 4, 2024 • 3.71M • 812k • ♡ 584

TeichAI/glm-4.7-2000x
Viewer • Updated 17 days ago • 1.98k • 280 • ♡ 41

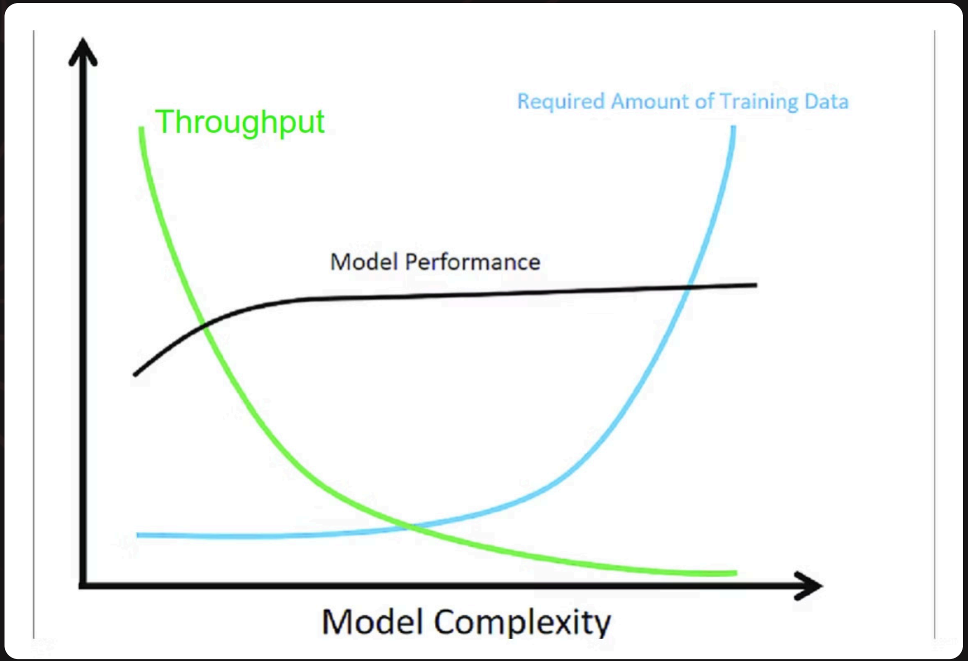
General recommendations

📊Data

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

🤖Model

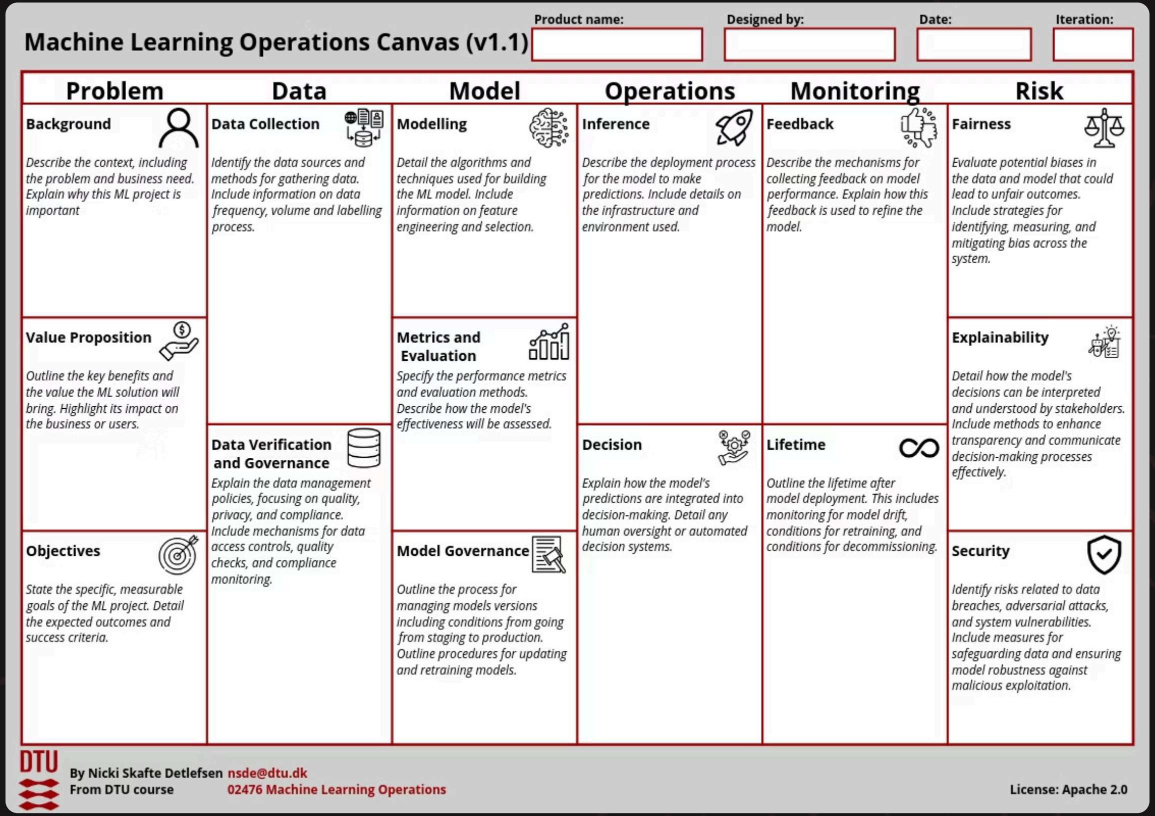
- Start out with a public baseline model if possible
- Choose smaller models over large models



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

Background

Describe the context, including the problem and business need. Explain why this ML project is important

Value Proposition

Outline the key benefits and the value the ML solution will bring. Highlight its impact on the business or users.

Objectives

State the specific, measurable goals of the ML project. Detail the expected outcomes and success criteria.

Data

Data Collection

Identify the data sources and methods for gathering data. Include information on data frequency, volume and labelling process.

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.

Model

Modelling

Detail the algorithms and techniques used for building the ML model. Include information on feature engineering and selection.

Metrics and Evaluation

Specify the performance metrics and evaluation methods. Describe how the model's effectiveness will be assessed.

Model Governance

Outline the process for managing models versions including conditions from going from staging to production. Outline procedures for updating and retraining models.

Operations

Inference

Describe the deployment process for the model to make predictions. Include details on the infrastructure and environment used.

Decision

Explain how the model's predictions are integrated into decision-making. Detail any human oversight or automated decision systems.

Monitoring

Feedback

Describe the mechanisms for collecting feedback on model performance. Explain how this feedback is used to refine the model.

Lifetime

Outline the lifetime after model deployment. This includes monitoring for model drift, conditions for retraining, and conditions for decommissioning.

Risk

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.

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.

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

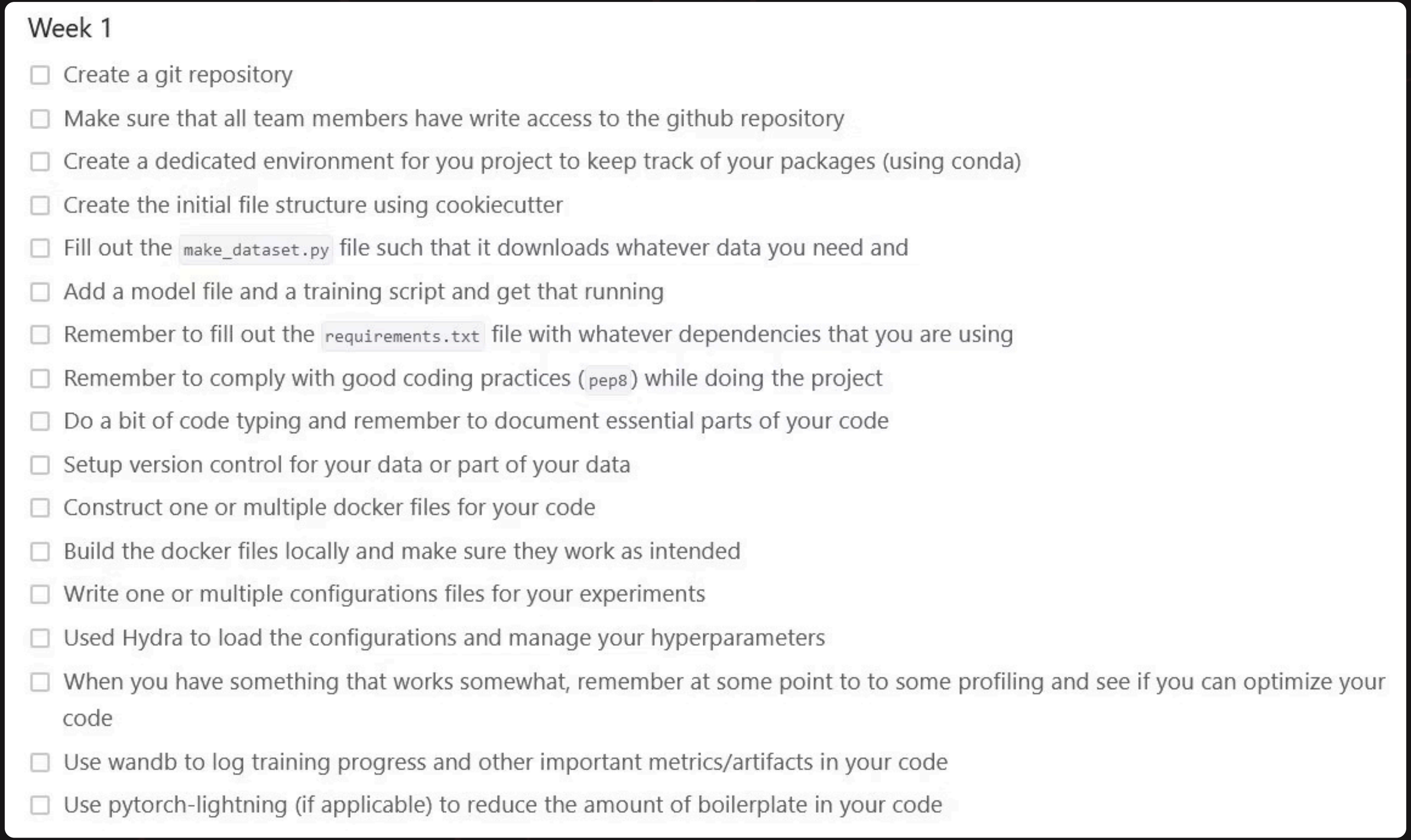
License: Apache 2.0

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

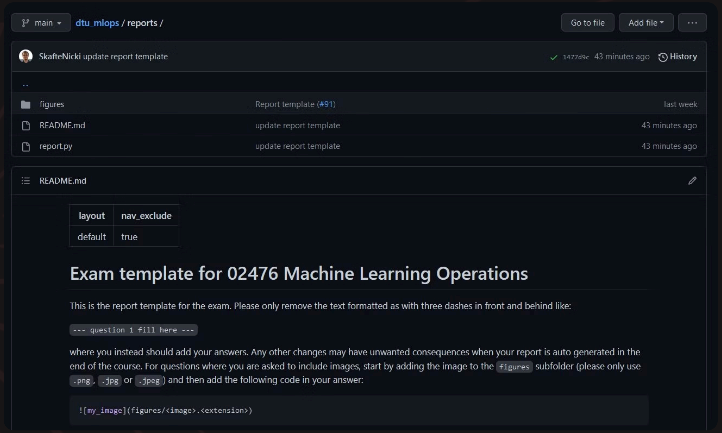
I am specifically looking at

> 💻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
```



dtu_mlops / reports /

SkafteNicki update report template ✓ 1477d9c 43 minutes ago

..		
figures	Report template (#91)	last week
README.md	update report template	43 minutes ago
report.py	update report template	43 minutes ago

README.md

layout	nav_exclude
default	true

Exam template for 02476 Machine Learning Operations

This is the report template for the exam. Please only remove the text formatted as with three dashes in front and behind like:

```
--- question 1 fill here ---
```

where you instead should add your answers. Any other changes may have unwanted consequences when your report is auto generated in the end of the course. For questions where you are asked to include images, start by adding the image to the `figures` subfolder (please only use `.png`, `.jpg` or `.jpeg`) and then add the following code in your answer:

```
![my_image](figures/<image>.<extension>)
```

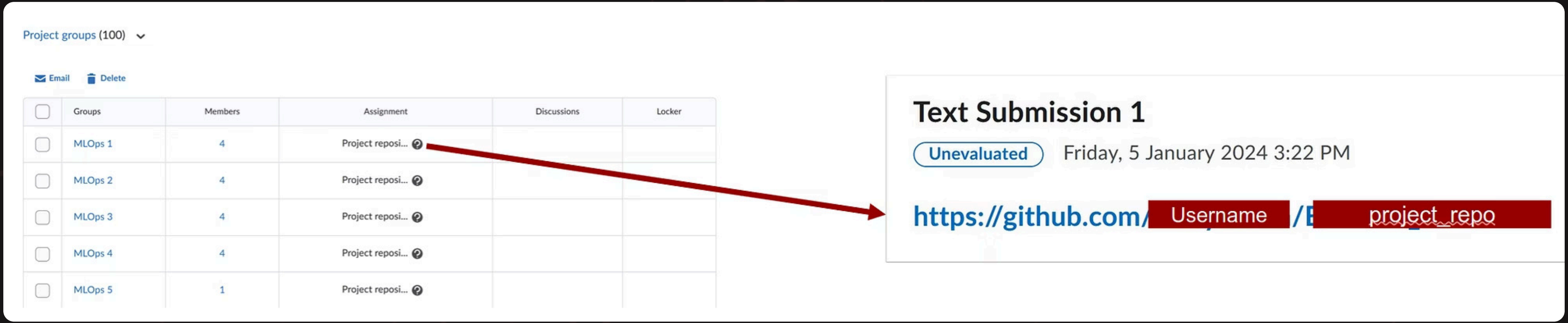
https://github.com/SkafteNicki/dtu_mlops/tree/main/reports

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



Project groups (100)

Email Delete

□	Groups	Members	Assignment	Discussions	Locker
□	MLOps 1	4	Project reposi...		
□	MLOps 2	4	Project reposi...		
□	MLOps 3	4	Project reposi...		
□	MLOps 4	4	Project reposi...		
□	MLOps 5	1	Project reposi...		

Text Submission 1

Unevaluated Friday, 5 January 2024 3:22 PM

https://github.com/Username/project_repo

A little helper for you guys

```
uvx cookiecutter \
    https://github.com/SkafteNicki/mlops_template \
    --checkout code_helpers
```

Comes with

- AGENTS.md: an overall agent helper with basic project commands explained
- .github/agents/dtu_mlops_agent.md: a specific agent for course related questions

If I find time I will update this to be an agent skill that dynamically can interact with the content of the course

Meme of the day

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

When someone asks why you never stops talking about machine learning
it defines who I am