

02476 Machine Learning Operations
Nicki Skafte Detlefsen

Projects

The “case”

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

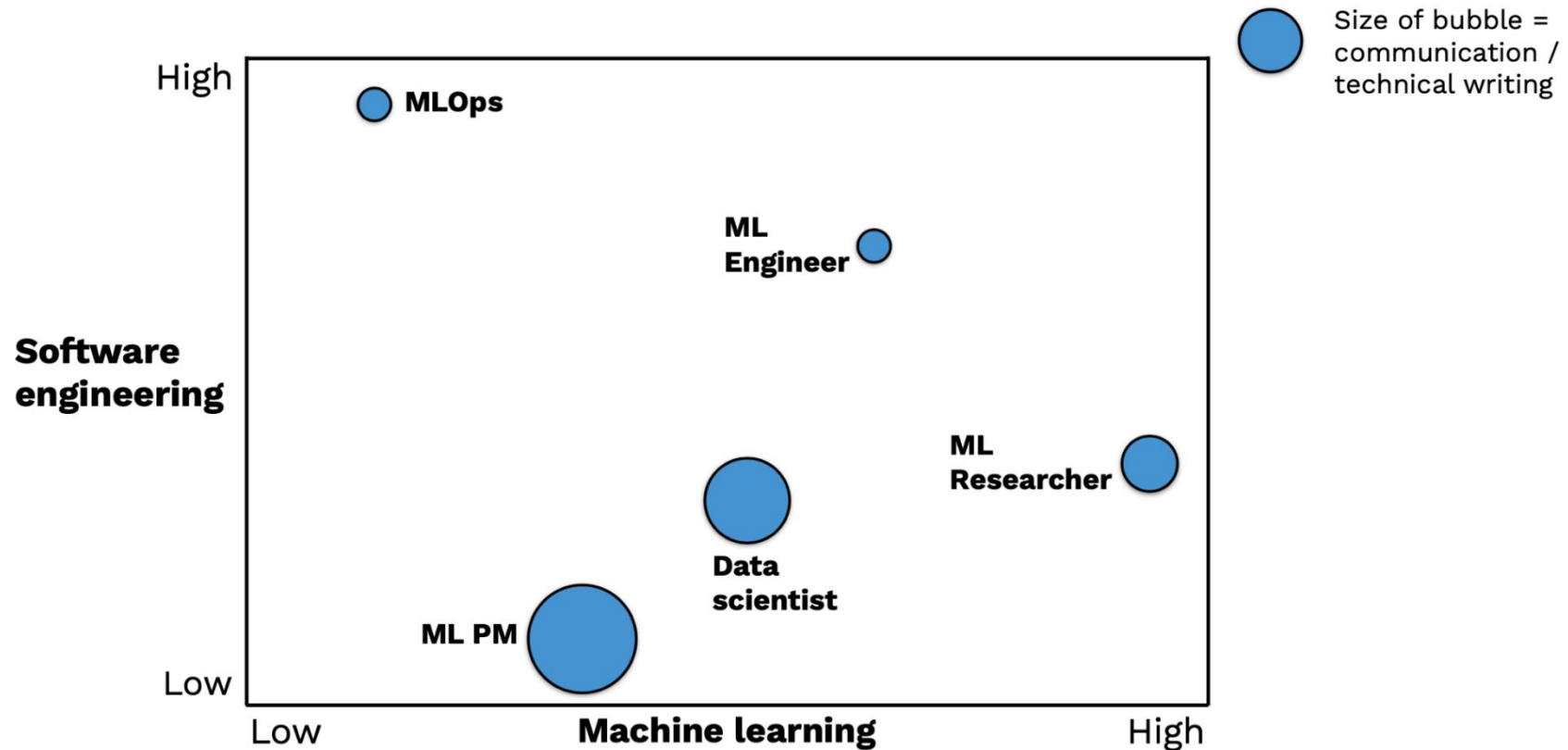
Your first job:

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 do not need to care about the model?

That is a job for the ML research not MLOps engineer

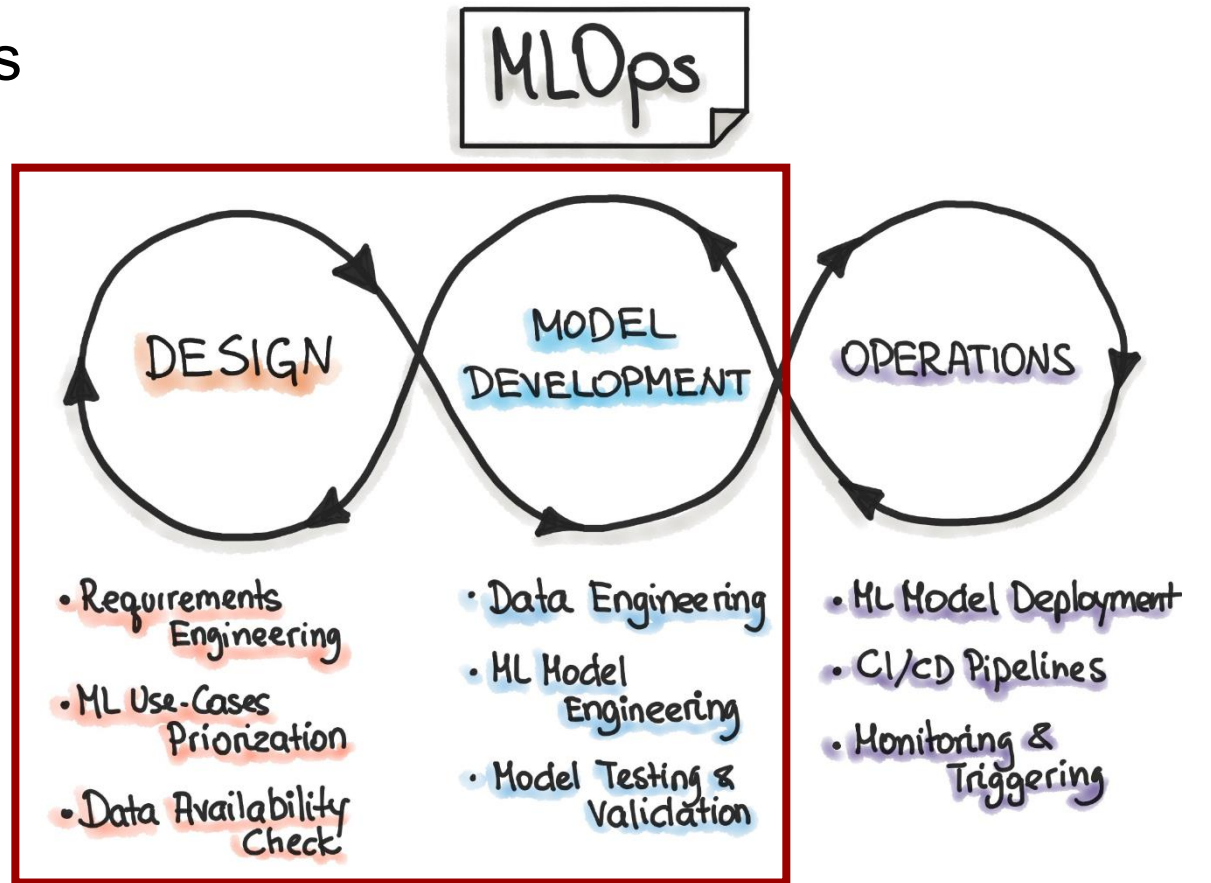


How to solve the problem?

💡 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

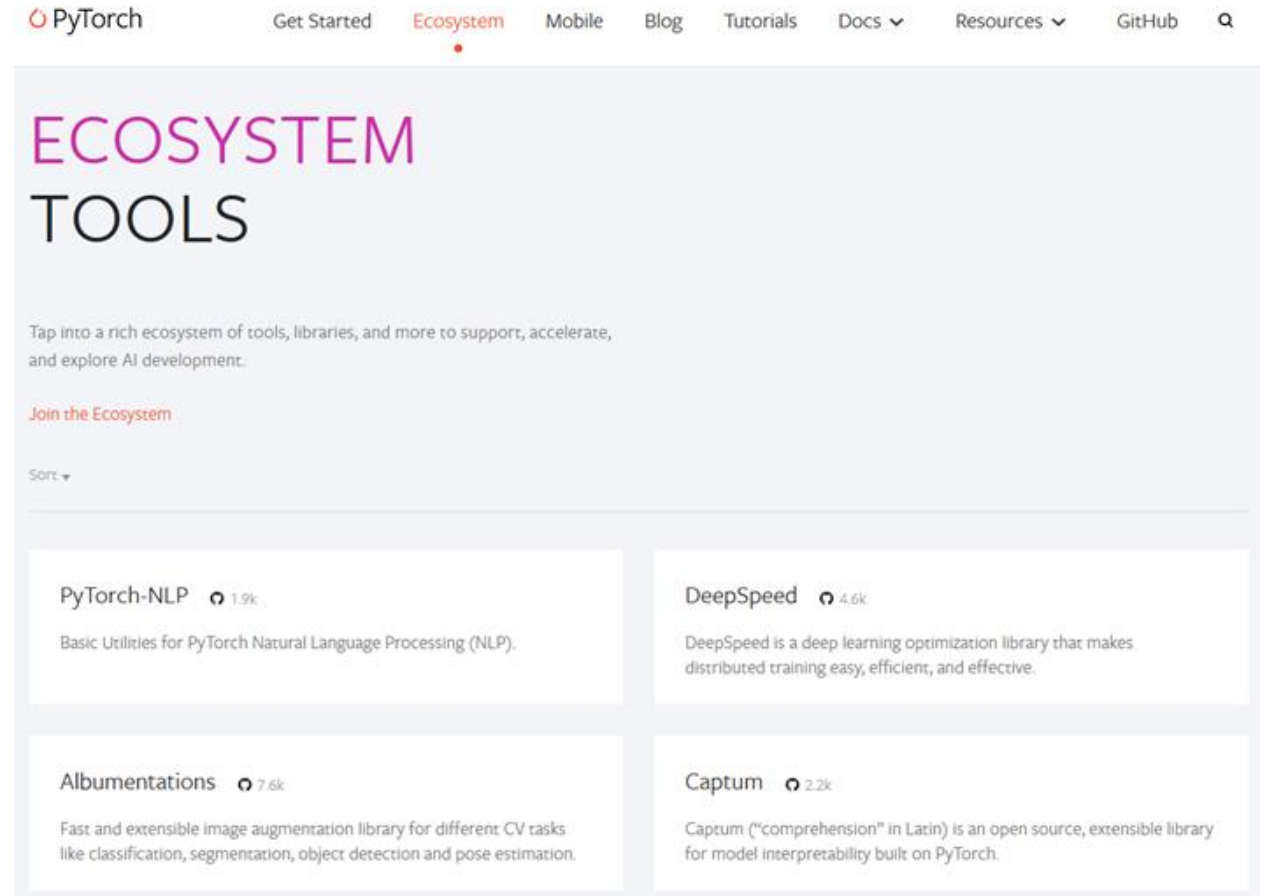


Fast track this part

The Pytorch Ecosystem

💡 Collection of frameworks build to be used in collaboration with Pytorch

💡 It is not a complete list of all great frameworks



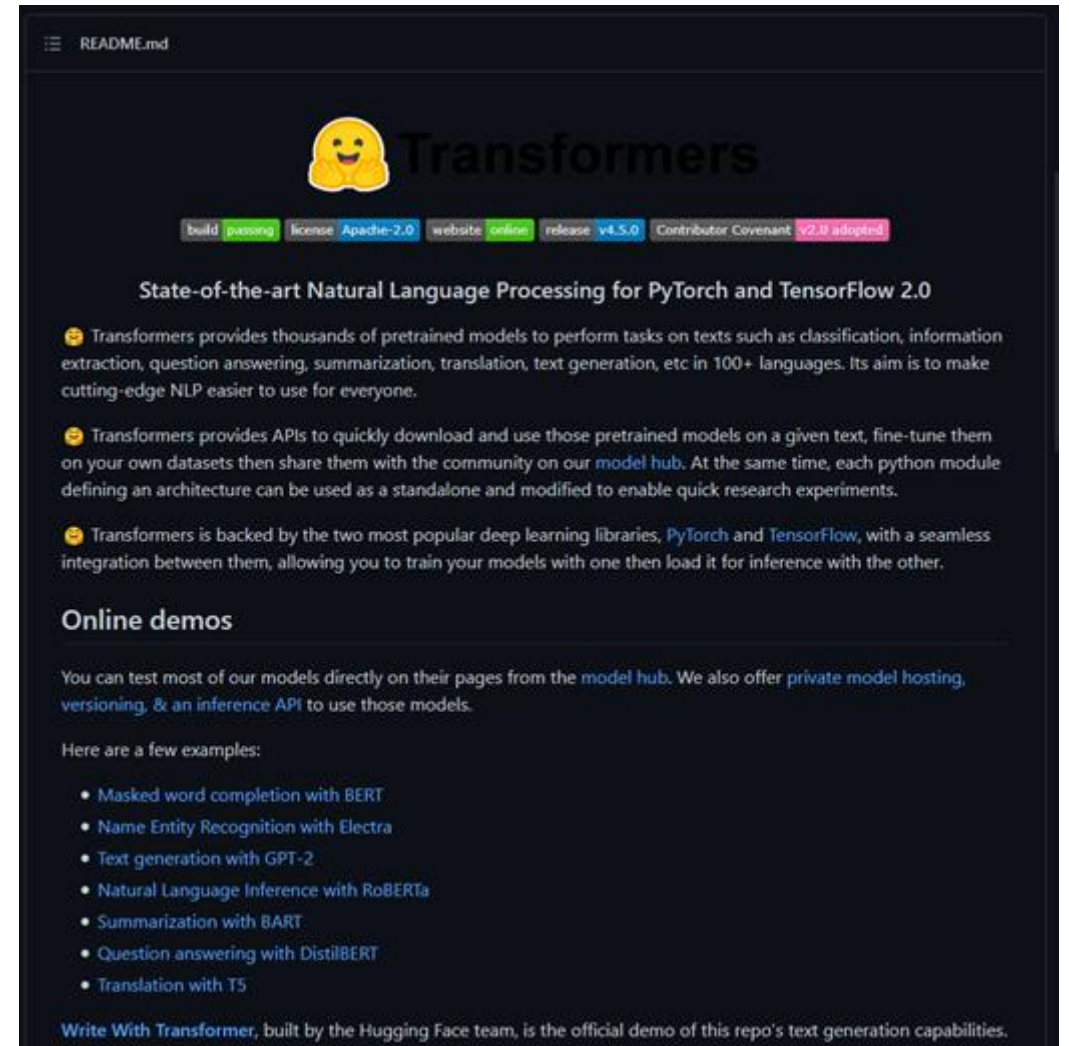
The screenshot shows the PyTorch Ecosystem Tools page. The navigation bar includes links for Get Started, Ecosystem (highlighted), Mobile, Blog, Tutorials, Docs, Resources, and GitHub. The main heading is "ECOSYSTEM TOOLS". Below the heading, there is a sub-heading "Tap into a rich ecosystem of tools, libraries, and more to support, accelerate, and explore AI development." and a link "Join the Ecosystem". A "Sort" dropdown menu is visible. The page displays four tool cards:

- PyTorch-NLP** (1.9k): Basic Utilities for PyTorch Natural Language Processing (NLP).
- DeepSpeed** (4.6k): DeepSpeed is a deep learning optimization library that makes distributed training easy, efficient, and effective.
- Albumentations** (7.6k): Fast and extensible image augmentation library for different CV tasks like classification, segmentation, object detection and pose estimation.
- Captum** (2.2k): Captum ("comprehension" in Latin) is an open source, extensible library for model interpretability built on PyTorch.

Example 1: Transformers

<https://github.com/huggingface/transformers>

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

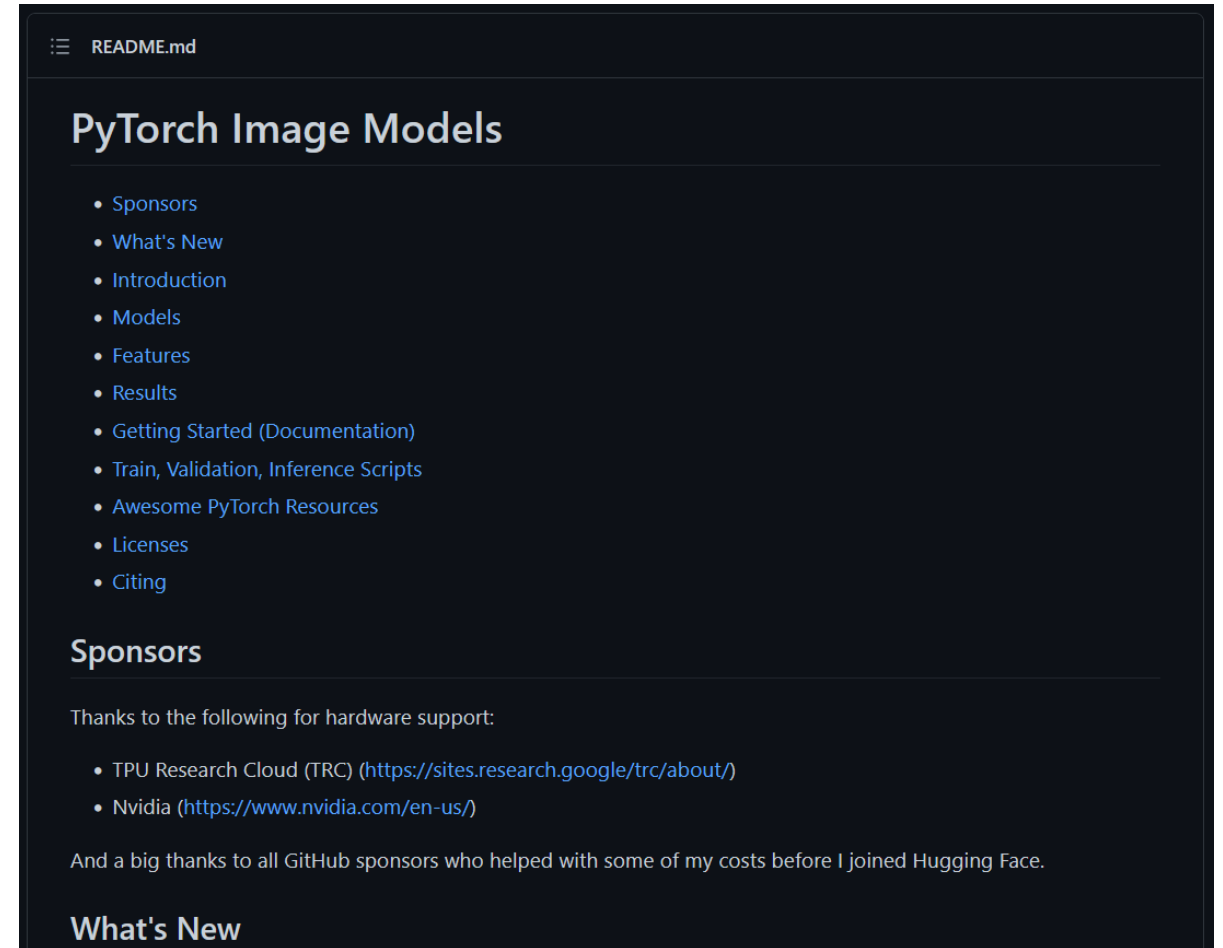


The screenshot shows the README page for the HuggingFace Transformers repository. At the top, there is a navigation menu with 'README.md' selected. Below the navigation is the HuggingFace logo (a yellow smiley face) and the word 'Transformers' in a large, bold font. Underneath the logo, there are several status badges: 'build passing', 'license Apache-2.0', 'website online', 'release v4.5.0', 'Contributor Covenant', and 'v2.0 adopted'. The main heading of the page is 'State-of-the-art Natural Language Processing for PyTorch and TensorFlow 2.0'. Below this heading, there are three paragraphs of text, each starting with a smiley face emoji. The first paragraph describes the library's capabilities in performing various NLP tasks across 100+ languages. The second paragraph explains how to use the library's APIs to download and fine-tune models. The third paragraph mentions the library's integration with PyTorch and TensorFlow. Below the text, there is a section titled 'Online demos' which provides information about testing models and offers private model hosting. A bulleted list of examples follows, including masked word completion, name entity recognition, text generation, natural language inference, summarization, question answering, and translation. At the bottom, there is a link to 'Write With Transformer', the official demo of the library's text generation capabilities.

Example 2: Pytorch-image-models

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

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



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.



How to get a good idea?

The screenshot shows the GitHub repository page for 'kornia'. The repository is on the 'master' branch, has 11 branches, and 16 tags. The commit history shows a recent commit by 'edgarriba' titled 'update new kornia logo' 2 days ago, with 1,533 commits in total. The file list includes folders like '.circleci', '.github', 'docker', 'docs', 'examples', 'kornia', 'packaging', 'test', and 'tutorials', as well as files like '.codecov.yml', '.gitconfig', '.gitignore', 'CHANGELOG.md', 'CITATION.md', 'CODE_OF_CONDUCT.md', and 'CONTRIBUTING.rst'. The right sidebar shows the repository's description as 'Open Source Differentiable Computer Vision Library for PyTorch', the website 'kornia.org', and tags for 'machine-learning', 'computer-vision', 'image-processing', and 'pytorch'. It also lists 16 releases, with the latest one being 'Morphological operators, Dee...' 21 days ago. At the bottom, a red box highlights the 'Used by' section, which shows 290 users and a row of profile icons with a '+ 282' indicator.

How to get a good idea?

The screenshot shows the Kaggle website interface. At the top, there are navigation links for Competitions, Datasets, Code, Discussions, and Courses. A search bar and 'Sign In'/'Register' buttons are also visible. The main content area features a large heading: "Start with more than a blinking cursor". Below this, it states: "Kaggle offers a no-setup, customizable, Jupyter Notebooks environment. Access free GPUs and a huge repository of community published data & code." There are two buttons: "REGISTER WITH GOOGLE" and "Register with Email".

The central focus is a notebook titled "Predict Malicious Websites: KGBest". The notebook content includes the following code:

```

import numpy as np
import pandas as pd
import sklearn as skl

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

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

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

# remove non-numeric columns
data = data.select_dtypes(include=[object])

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

# save it dataframe
train.head()

```

Below the code, there is a table with columns: url_length, number_of_urls, content_length, top_level_domain, ip_address, ip_port, domain_age, and ip_geo. The table contains three rows of data.

At the bottom of the screenshot, there is a cookie consent banner: "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." with "Got It" and "Learn more" buttons.

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.

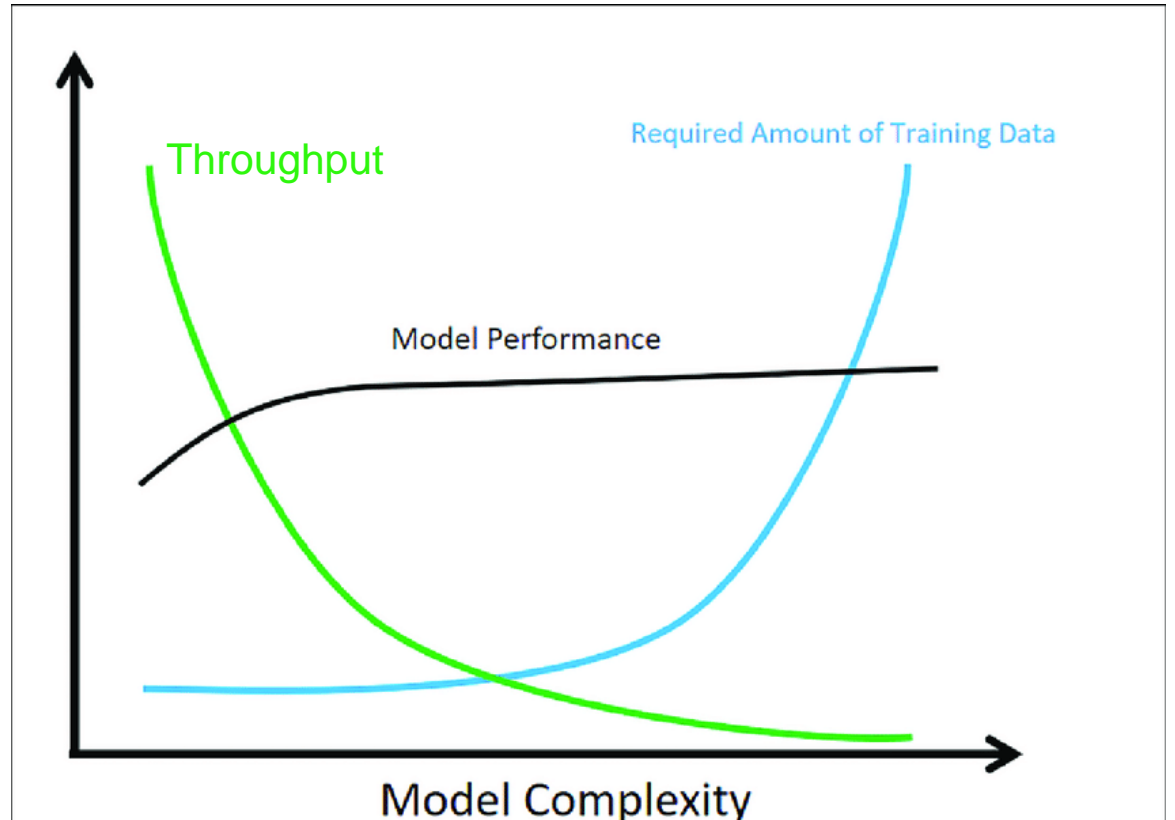
General recommendations

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. Pick any Pytorch-based third-party package (not used in the course) you would like to work with
4. Write a small project description
 - A. Overall goal of the project
 - B. What framework are you going to use and do you intend to include the framework into your project?
 - C. What data are you going to run on (initially, may change)
 - D. What models do you expect to use
5. Create project repository
6. Upload project description as part of README.md file
7. Work on the rest of project...

ML Canvas for staying organized and thinking ahead

💡 Create service to quickly get overview of ML news from many sources



ml-canvas.pdf

Product:
Authors:
Date:
Version:

Machine Learning Canvas

| | | | | |
|--|--|---|--|---|
| <p>Background </p> <p>Describe the customer's goals and pains.</p> <p>User: machine learning engineers Goals: staying up to date for work Pain: too much unlabeled content on the net</p> | <p>Solution </p> <p>Define the solution, including features, integration, constraints and what's out-of-scope</p> <p>Core features: *predict the correct tag for given content *user feedback for incorrectly classified content * workflow to categorize ML content that out model is incorrect/unsure about</p> <p>Integrations: * ML content from reliable sources</p> <p>Alternatives * allow users to add content manually and classify them</p> <p>Constraints: * maintain low latency when classifying * only recommend tags from list of approved tags * avoid duplications</p> <p>Out of scope: * auto discover new tags for documents that does not fit one currently in data</p> | <p>Data </p> <p>Identify the training and production data sources, as well as the labeling process and decisions.</p> <p>Data overall: Title, description, tag</p> <p>Training Large set of scraped data that needs manually labeling. Start small and scale up as needed.</p> <p>Production: Incoming data from users</p> | <p>Modeling </p> <p>List the iterative approach to model our task.</p> <p>Start with simple rulebased system</p> <p>If this does not work, then move to simple ML models</p> <p>If this does not work, then move on to more complex ML models</p> | <p>Feedback </p> <p>Outline sources of feedback from our system to use for iteration.</p> <p>Enforce some human-in-the-loop checks when there is a low confidence in classification</p> <p>Allow users to report issues related to misclassification</p> |
| <p>Value proposition </p> <p>Propose the product with the value it creates and the pains it alleviates.</p> <p>Product that discovers and categorizes and from popular sources (reddit, twitter)</p> <p>Alleviates: display categorized content for users to discover Advantages: save users time by not having to search through lot of content themselves</p> | <p>Feasibility </p> <p>Discuss the feasibility of the solution and if we have the required resources.</p> <p>If open-source data exist then creating the model should be possible Else it will require \$ for API to sites like reddit, twitter to scrape the data ourself</p> <p>Modelling seems feasible as text classification is well understood</p> | <p>Metrics </p> <p>Prioritize key metrics that reflect the objectives.</p> <p>Accuracy, confusion matrix, F1 Standard classification metrics</p> <p>Latency measurements</p> | <p>Inference </p> <p>Decide whether we want to do batch (offline) or real-time (online) inference.</p> <p>Do scraping in batches at a given time point and do the processing in batches to optimize processing</p> | <p>Project </p> <p>Define the required team members, deliverables and projected timelines.</p> <p>Nicki does everything</p> |
| <p>Objectives </p> <p>Breakdown the product into key objectives that need to be delivered.</p> <p>Discover ML content from sources Classify incoming content for users Display categorized content</p> | | | | |

Machine learning canvas from [Made With ML](#) by [Goku Mohandas](#)
License: [CC BY-SA 4.0](#)

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 do 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 looking at

-  How well are your code, data, experiments version controlled and reproducible
-  Is appropriate continuous integration implemented for automatization of tasks
-  Is a final model deployed and able to be interacted with a end user

When stuff does not fit

What if I cannot get framework X to work in my project ?

💡 That is completely fine, but make sure to either argument why this was not possible, not necessary or why you choose to go with an alternative.

Example:

We did not end up using Weights and Bias for tracking out experiments because the group did already have prior experience with MLflow and therefore opted for using that framework

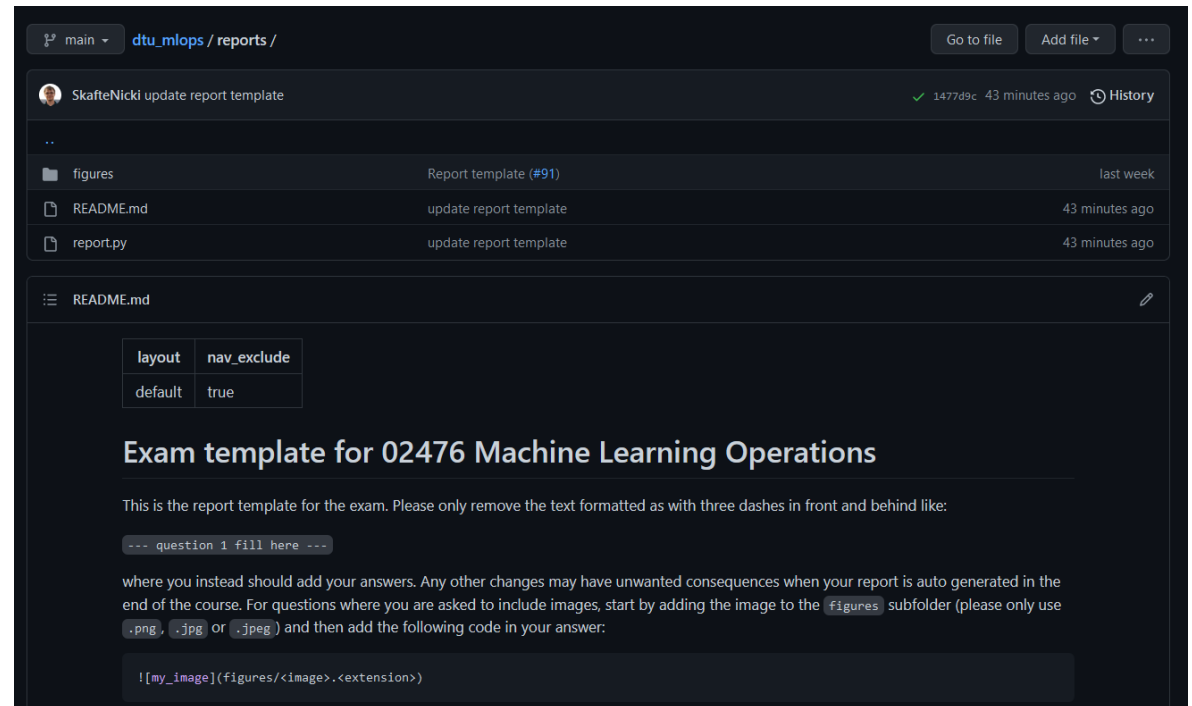
Exam report template

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/SkaftNicki/dtu_mlops/tree/main/reports



I will scrape your report on the 19/1 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 1 have access to learn, signup to a group, hand-in and send email with remaining student ids to me
- If non have access to learn, send email with student ids and project repository, I will send back a group number

Project groups (100) ▾

Email Delete

| <input type="checkbox"/> | Groups | Members | Assignment | Discussions | Locker |
|--------------------------|---------|---------|-------------------|-------------|--------|
| <input type="checkbox"/> | MLOps 1 | 4 | Project reposi... | | |
| <input type="checkbox"/> | MLOps 2 | 4 | Project reposi... | | |
| <input type="checkbox"/> | MLOps 3 | 4 | Project reposi... | | |
| <input type="checkbox"/> | MLOps 4 | 4 | Project reposi... | | |
| <input type="checkbox"/> | MLOps 5 | 1 | Project reposi... | | |

Text Submission 1

Unevaluated

Friday, 5 January 2024 3:22 PM

https://github.com/Username/project_repo

Exam wishes

Fill out this form:

<https://forms.gle/RfXkPvUkHHvpZFy56>

- Participate online (not EUROTOEQ students)
- Request specific timeslot
- Request grade on the 7-point scale
- Something else

Meme of the day

**When someone asks why you never stops
talking about machine learning**

