Intro to contemporary MLOps

Vladislav Goncharenko,

Harbour Space & UTCC, fall 2023

What is MLOps

girafe
ai

01

What is MLOps for?

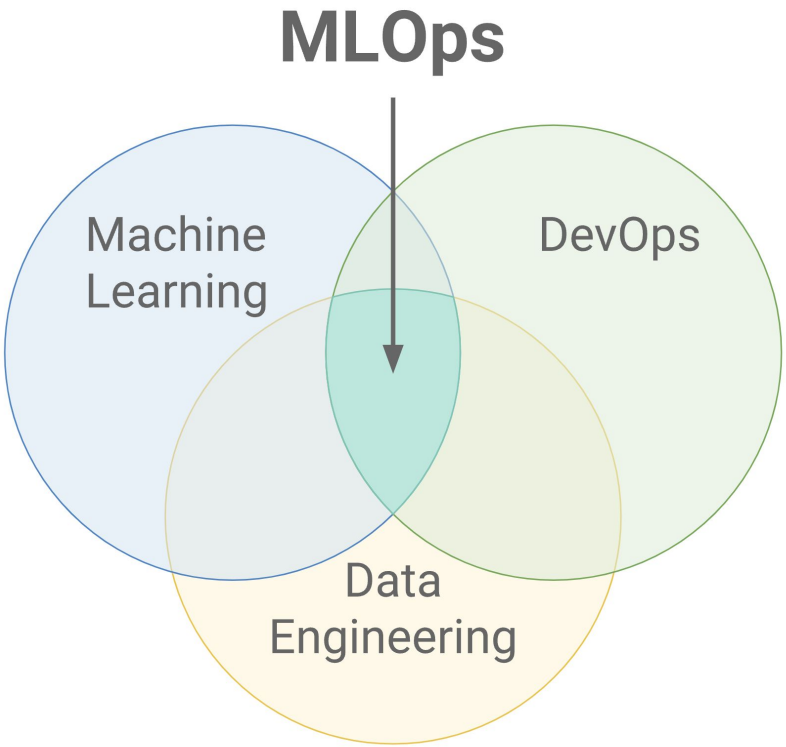
- Resources spent on ML models are growing
  - data mining (cpu)
  - labeling (people)
  - models training (gpu)
- Trainings reproducibility
  - Not only in industry, also in academia
- Models delivery
  - shorting time-to-market
  - reduce routine
- Competencies separation
  - division of labor

Definition

MLOps is a paradigm that aims to deploy and maintain machine learning models in production reliably and efficiently.

MLOps seeks to increase automation and improve the quality of production models, while also focusing on business and regulatory requirements

Well-known common knowledge site



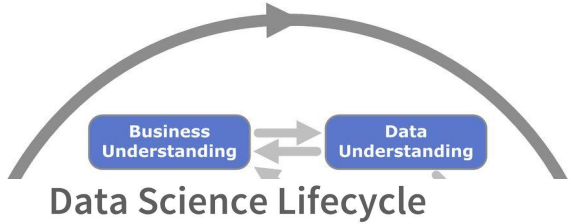
Why do you need MLOps

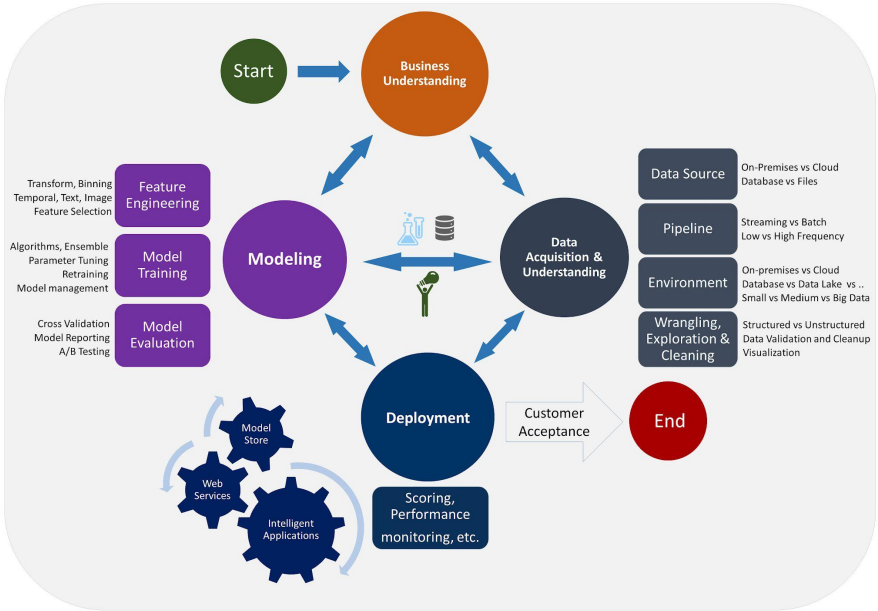
because no one will do it for you!!!

For diving deeper into industrial systems design I suggest system design primer

Standards for ML/DS projects

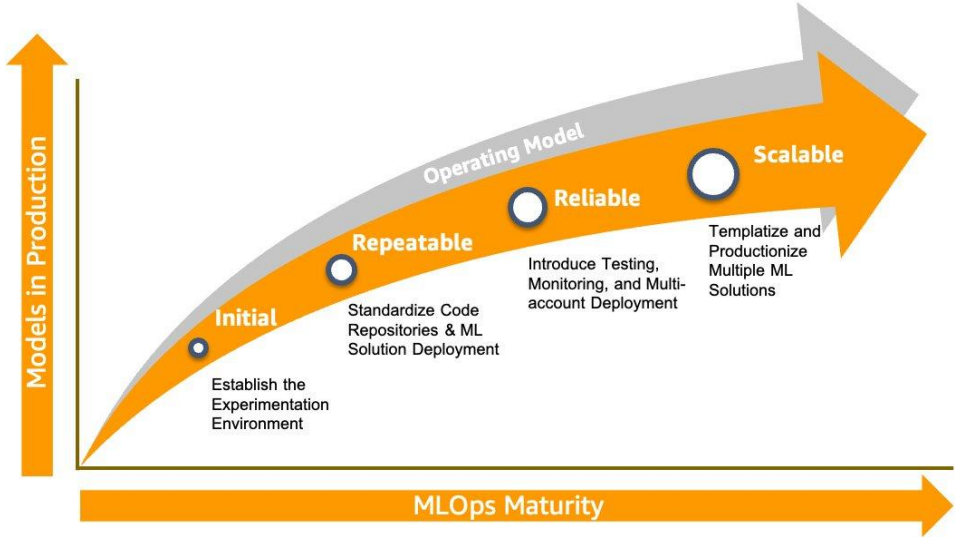
- **CRISP DM**
  - Cross-industry standard process for data mining
  - proposed in 1999
  - upgraded to ASUM-DM in 2015
- **TDSP**
  - Team standard Data Science Process
  - by Microsoft





MLOps at big tech

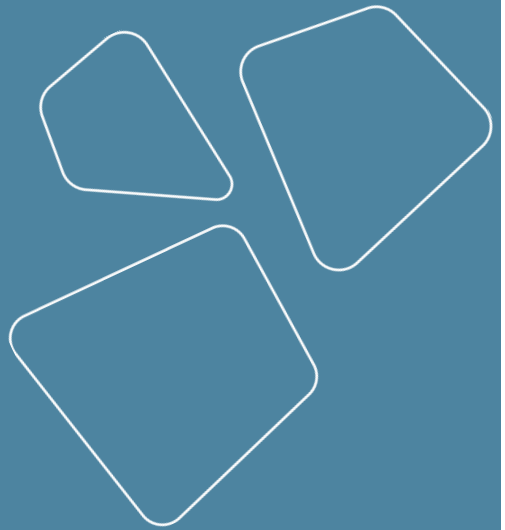
- Amazon https://aws.amazon.com/sagemaker/mlops/
- Google https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning
- Nvidia https://blogs.nvidia.com/blog/2020/09/03/what-is-mlops/
- Databricks
- Yandex: YTSaurus + Nirvana + Toloka



Even Gucci!!!

Youtube video

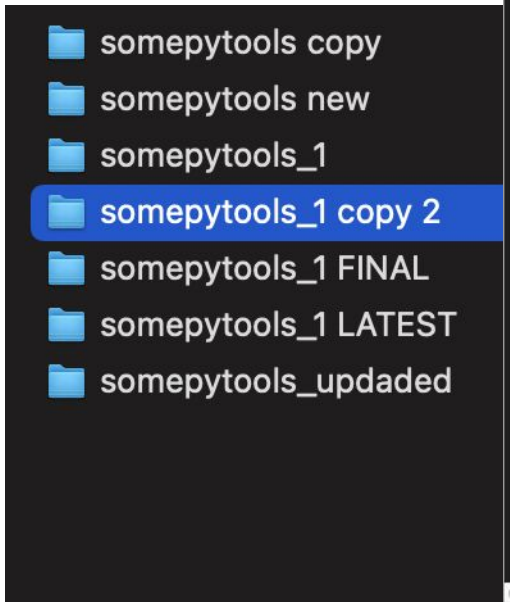
Topics for today

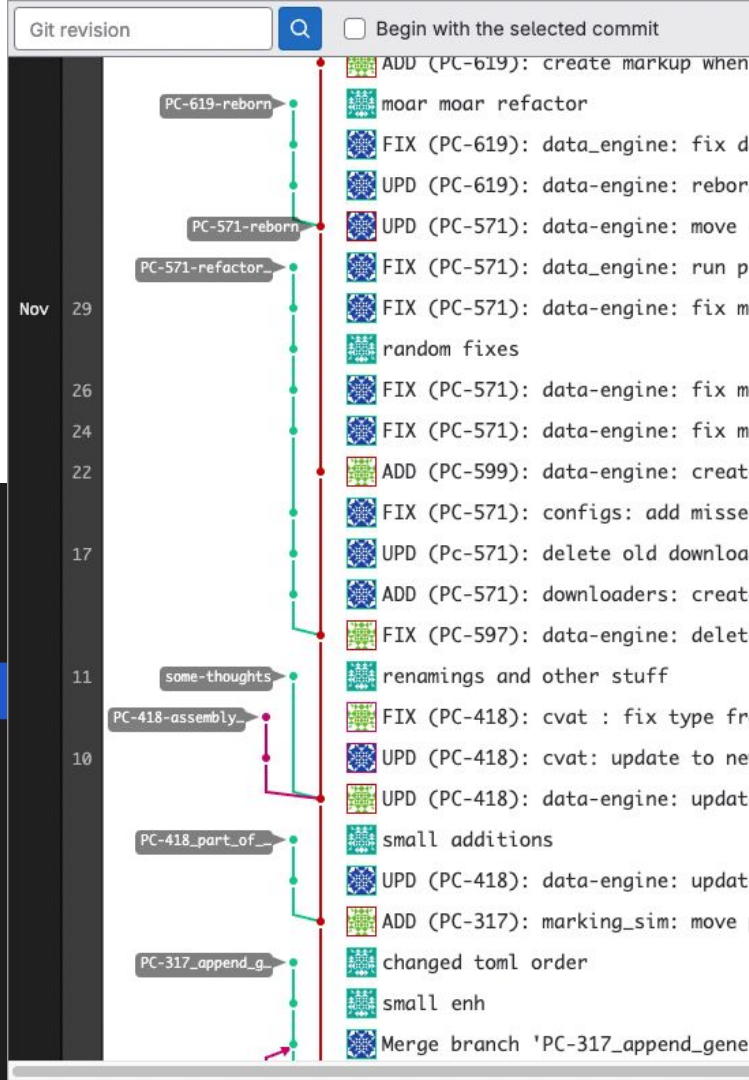


- Code management
- Data management
- Computation models
- Logging and visualization
- Repeatable tasks
- Labeling

Code management

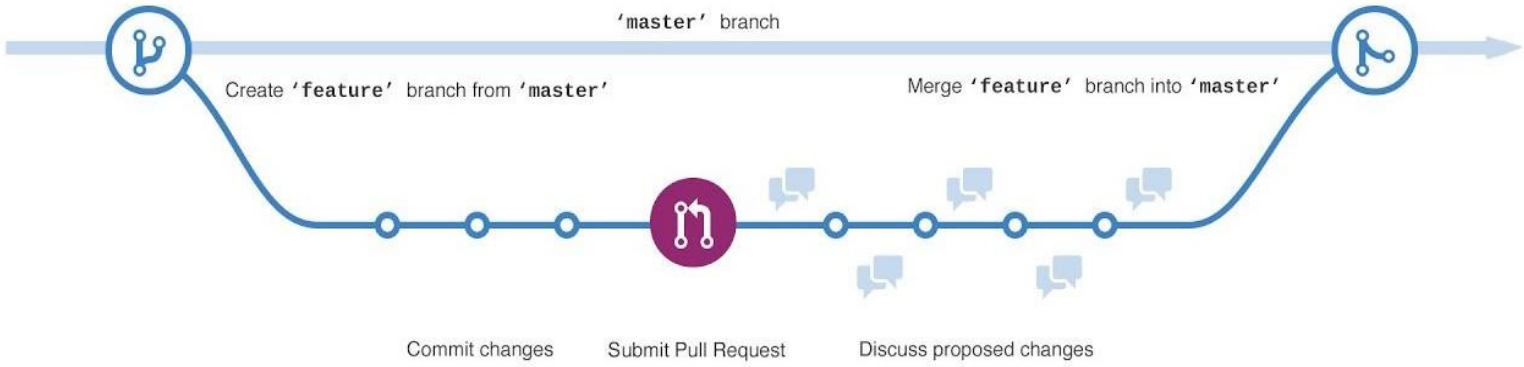
- One local copy
- Many local copies
- Remote copies
- Version control system (VCS e.g. git, svn, etc.)





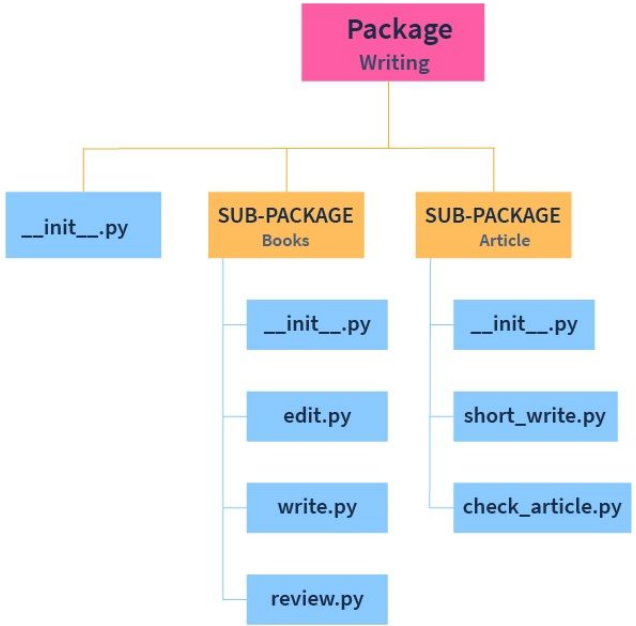
Version Control System (VCS)

- Source code - git
- Cloud remote - gitlab or github
- Best practice - merge requests (pull requests)



Code distribution

- Source code
  - clone project on runtime machine
  - update using git pull
  - don’t need code structure
- Packages
  - install using pip install
  - update via pip update
  - requires structure
  - for simple packaging poetry is a great tool



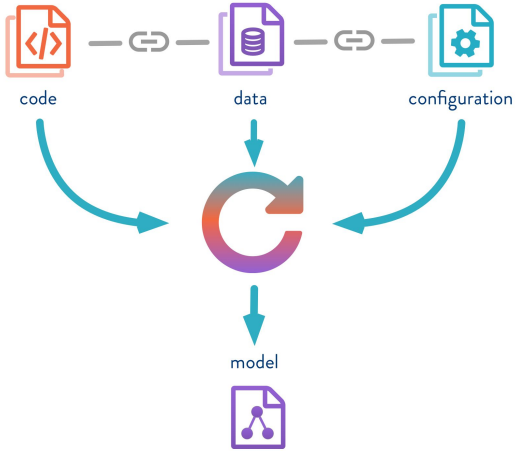
Data management

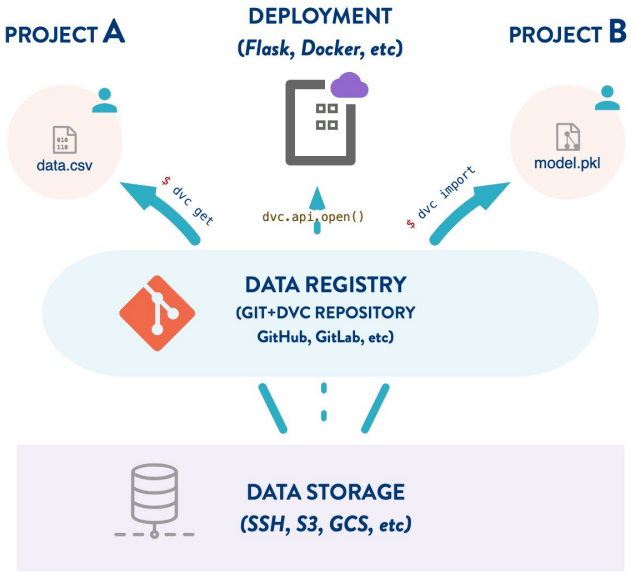
Applicable both to datasets and trained model artifacts

- Locally
- Remotely
- Distributed
  - small data
    - dvc
    - git LFS
  - big data
    - MapReduce paradigm
    - hadoop stack or YTsaurus system
  - data catalog datahubproject.io

Data Version Control (DVC)

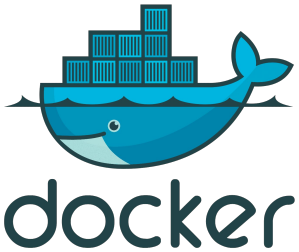
- git for data is DVC (tutorials: one, two)
- Versioning and Access submodules





Computation models

- Bare metal
- Virtual
  - classical VMs: KVM, vmware
  - docker
- Dynamically allocated
  - Task queues (slurm, clearml)
  - MapReduce
  - Kubernetes, k8s (Kubeflow)
  - Serverless computing (Amazon Lambda)



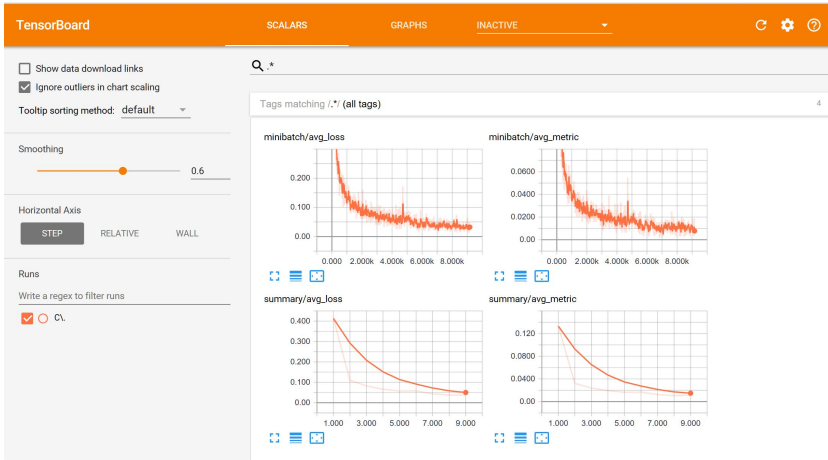
kubernetes

Logging and visualization

Gets important as number and size of experiments grow

Applicable for experiments, not for production logging

- absence
- print
- local service
  - tensorboard
- remote service
  - ml-flow
  - clear-ml
  - kubeflow
  - w&b, neptune and other proprietary soft



Experiments Tracker

- Tensorboard
- **MLFlow**
- ClearML
- sacred
- Kubeflow
- neptune.ai
- weights and biases
- .....

Repeatable tasks

Defacto airflow is an industry standard

For custom stack usually special tools exist

Apache Airflow

Data labeling

- Self-hosted
  - cvat
  - and many more
- Cloud
  - toloka
  - mturk

amazon
beta
mechanical turk

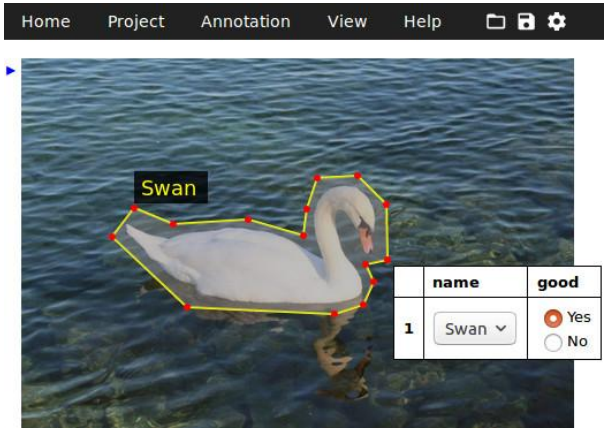
Self-hosted

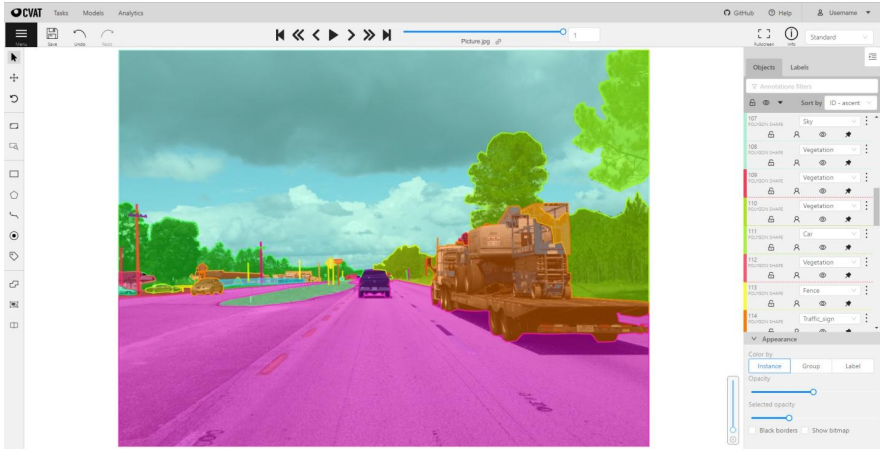
Solutions specific to Computer Vision

- simple cases - VIA (free software, standalone)
- scalable solution - CVAT (free software, server based)
- special cases - hasty.ai (proprietary, server based)

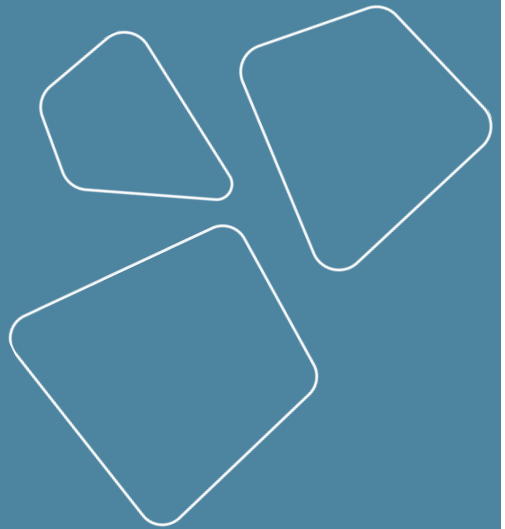
All of them are web based

Suggest your favorite
tools in comments!
(especially for other tasks)





Review



- MLOps motivation
- Code management
- Data management
- Computation models
- Logging and visualization
- Repeatable tasks
- Labeling

Thanks for attention!

Questions?