HSLU Lucerne University of Applied Sciences and Arts

Introduction to MLOps

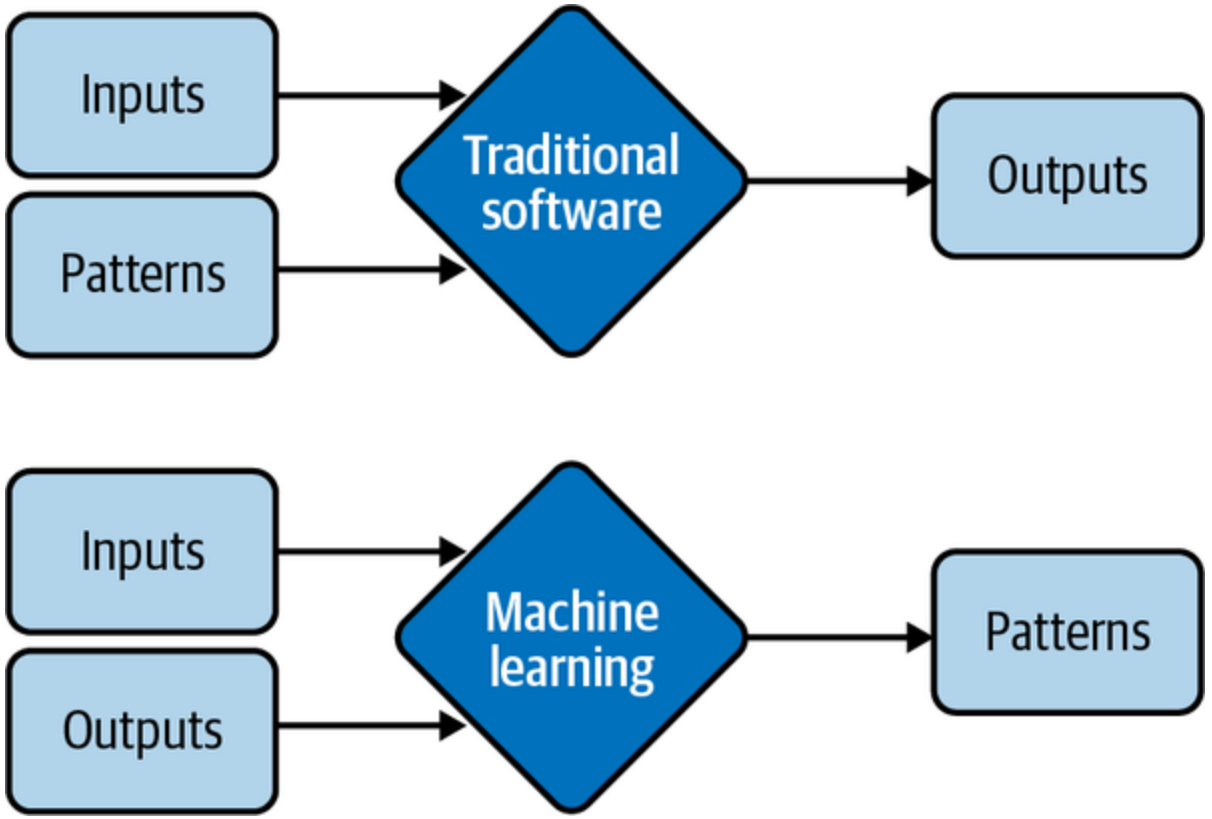
Andreas Marfurt
MLOps

Information Technology
18.09.2025

Overview

- When to use ML
- What is MLOps?
- Why MLOps?
- Challenges
- ML systems lifecycle
- Further reading

When to use ML



When to use ML

Machine learning is an approach to (1) learn (2) complex patterns from (3) existing data and use these patterns to make (4) predictions on (5) unseen data.

When to use ML

Machine learning is an approach to (1) learn (2) complex patterns from (3) existing data and use these patterns to make (4) predictions on (5) unseen data.

(1) learn: the system must be capable of learning, and there must be relationships (inputs-outputs) to learn

When to use ML

Machine learning is an approach to (1) learn (2) complex patterns from (3) existing data and use these patterns to make (4) predictions on (5) unseen data.

(2) complex patterns: easy patterns can be captured by rules, non-existing patterns cannot be learned

When to use ML

Machine learning is an approach to (1) learn (2) complex patterns from (3) existing data and use these patterns to make (4) predictions on (5) unseen data.

(3) existing data: data needs to be available (or you should start collecting it now!)

When to use ML

Machine learning is an approach to (1) learn (2) complex patterns from (3) existing data and use these patterns to make (4) predictions on (5) unseen data.

(4) predictions: the problem requires probabilistic predictions (instead of definite answers), the more predictions the better to use an (automated) system for them

When to use ML

Machine learning is an approach to (1) learn (2) complex patterns from (3) existing data and use these patterns to make (4) predictions on (5) unseen data.

(5) unseen data: the data that serves as input to predictions should have the same characteristics (distribution) and patterns as the training data

When to use ML

Other properties that make ML shine

- The more data available the better
- Cost of wrong predictions is cheap
- Task must be executed many times (amortization)
- Task is repetitive (human concentration suffers)
- Patterns are changing (hand-coded rules don't work)

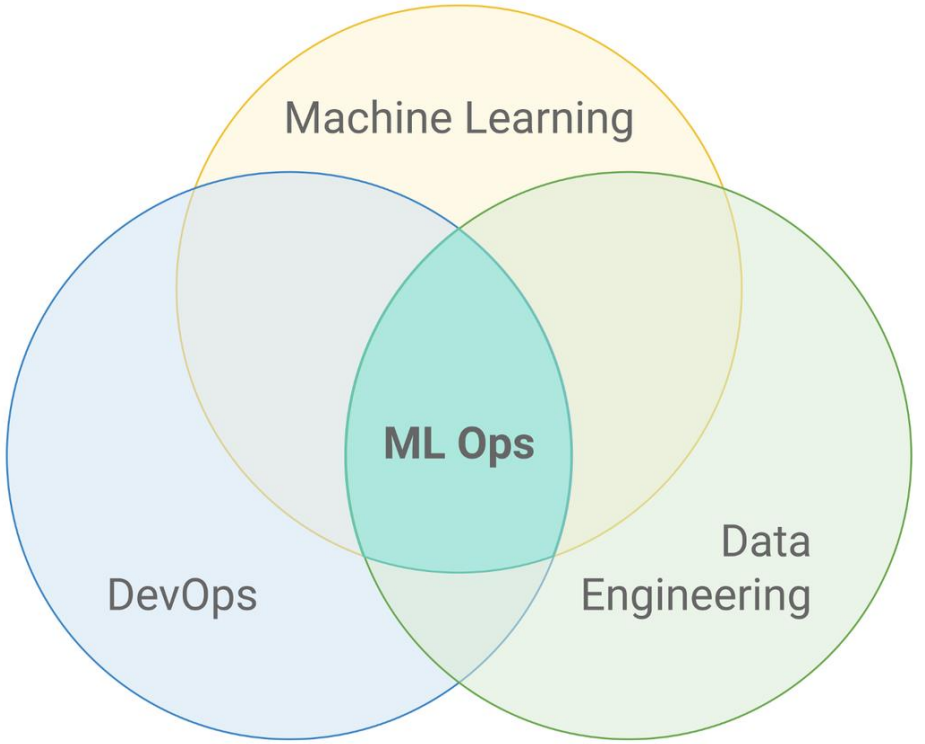
When *not* to use ML

- If it's illegal or unethical (see lecture on governance)
- Simple solutions also work
- It's not cost-effective
  - (but maybe subproblems could be solved by ML?)

What is MLOps?

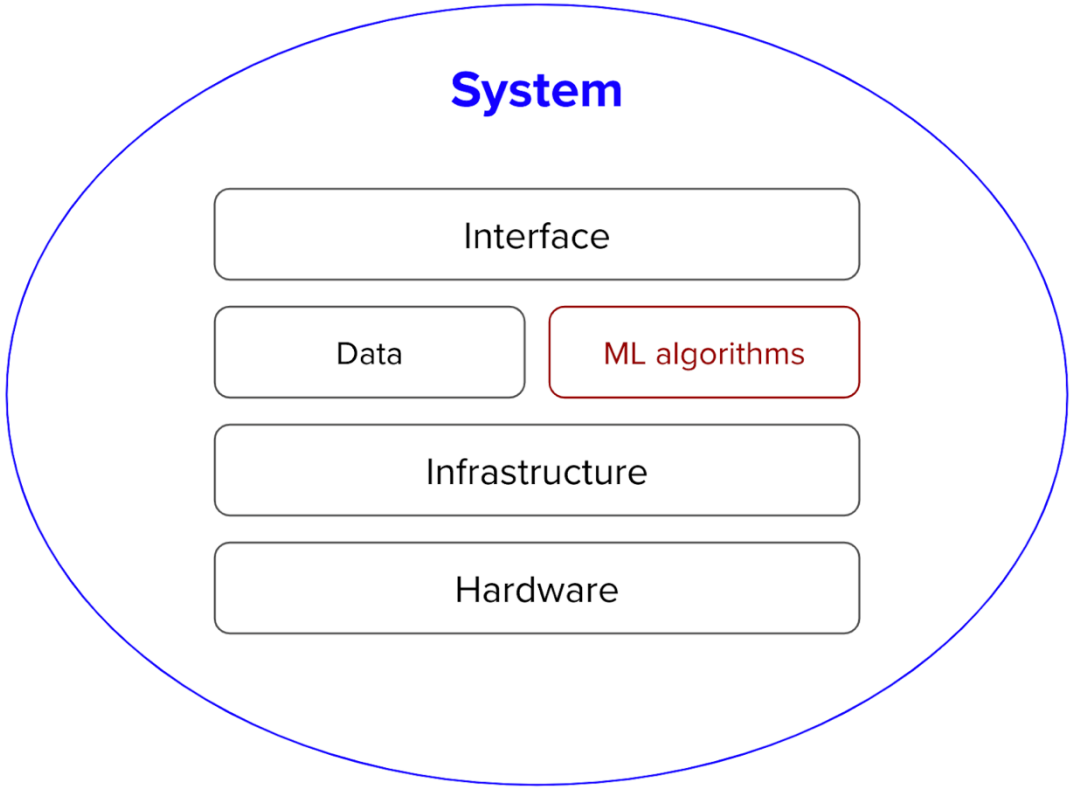
a paradigm that aims to deploy and maintain machine learning models in production reliably and efficiently
– *Wikipedia*

- Started as best practices
- Now an independent approach to lifecycle management of an ML system



https://commons.wikimedia.org/wiki/File:MLOps_venn_diagram.png

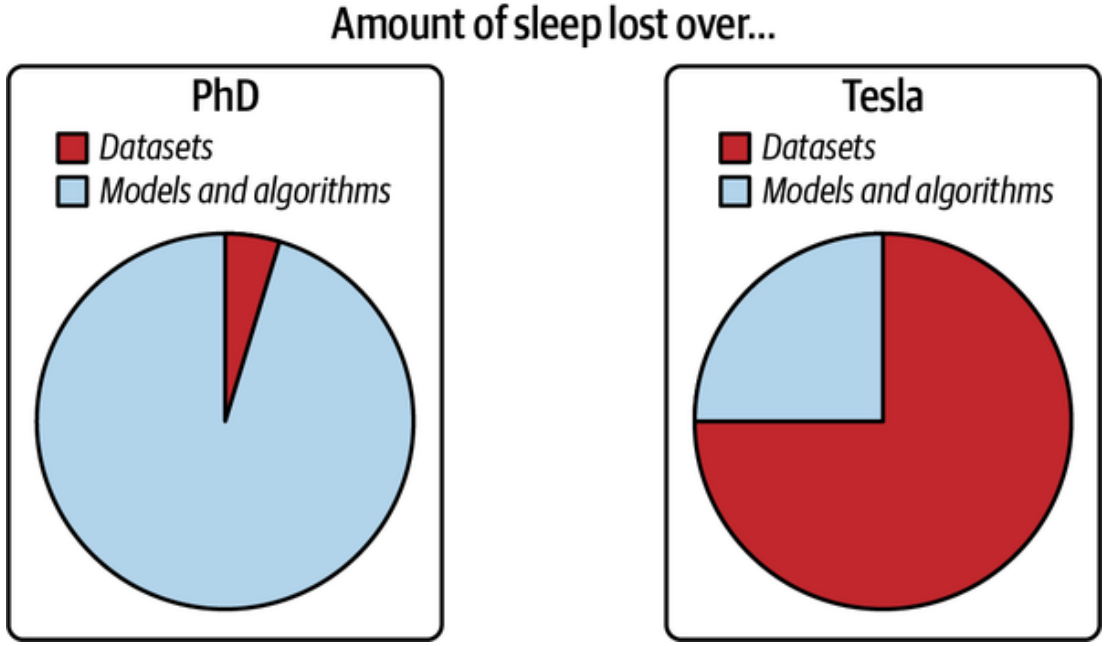
An ML system is more than algorithms



Stanford CS329s lecture notes

Why MLOps?

More time spent on productionizing an ML system than developing it



Adapted from Andrej Karpathy

THE COGNITIVE CODER

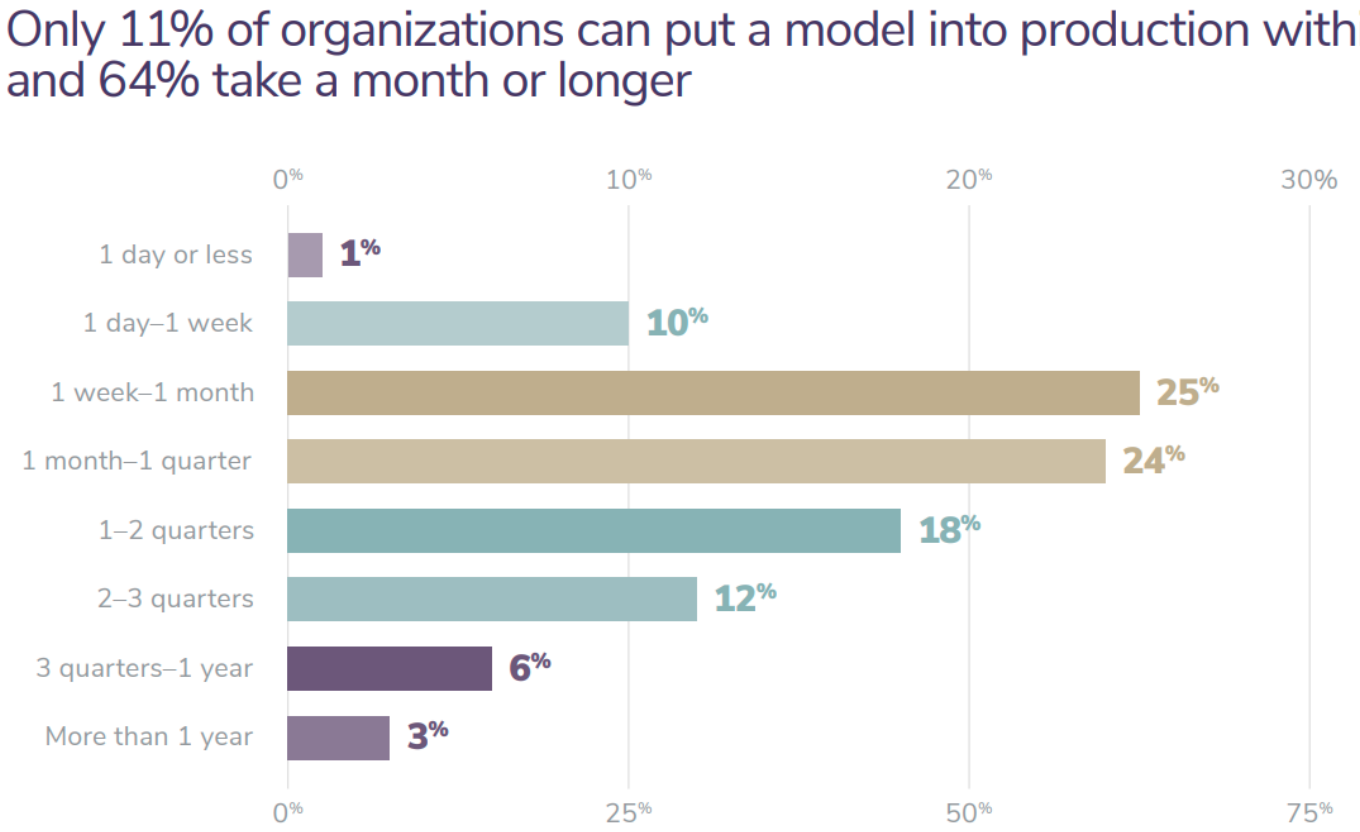
By Armand Ruiz, Contributor, InfoWorld | SEP 26, 2017 7:22 AM PDT

The 80/20 data science dilemma

Most data scientists spend only 20 percent of their time on actual data analysis and 80 percent of their time finding, cleaning, and reorganizing huge amounts of data, which is an inefficient data strategy

Why MLOps?

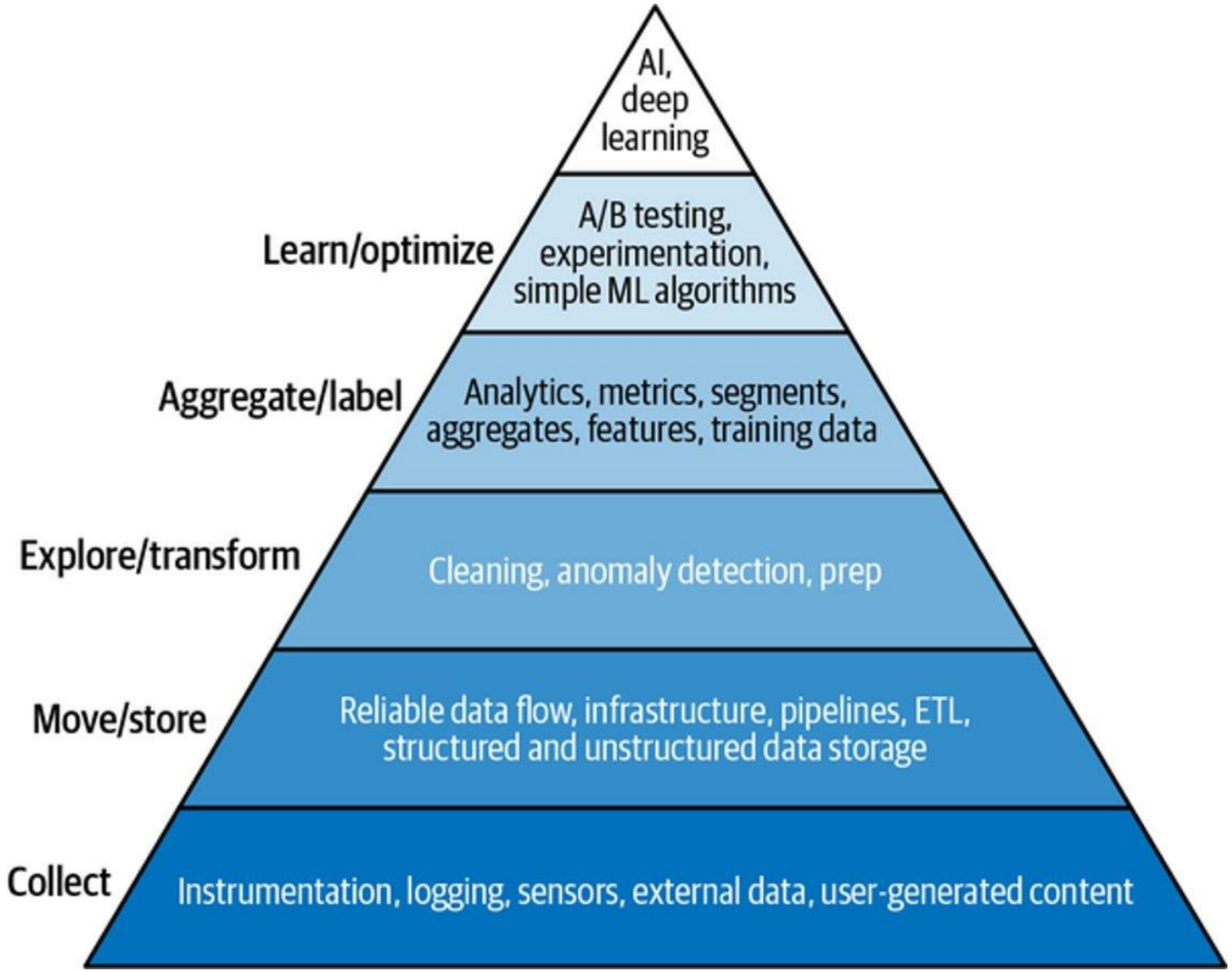
Automating the ML pipeline gets models into production faster



Algorithmia - 2021 enterprise trends in machine learning

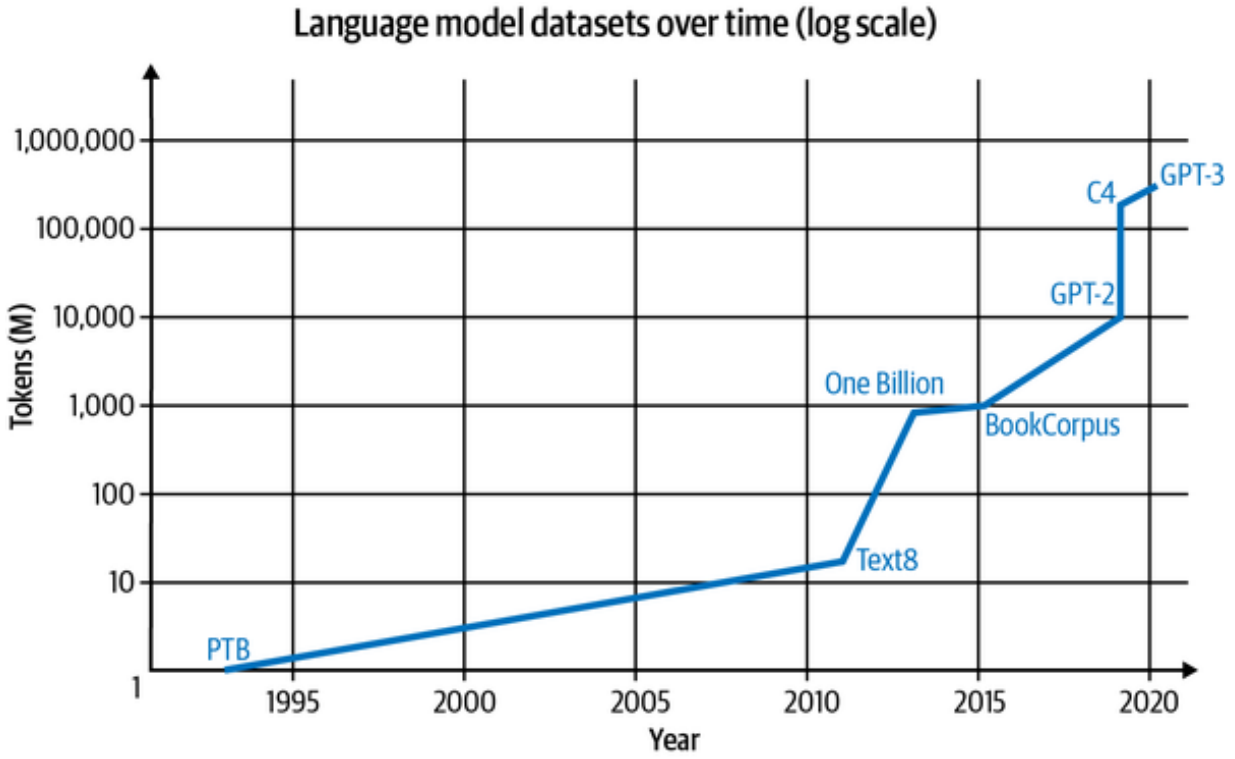
Why MLOps?

Get the fundamentals (data, preprocessing, metrics) in place before working on complicated algorithms



Why MLOps?

Datasets grow larger exponentially



Challenges

	Studies	Production
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting
Fairness	Good to have	Important
Interpretability	Good to have	Important

Challenges: Reproducibility

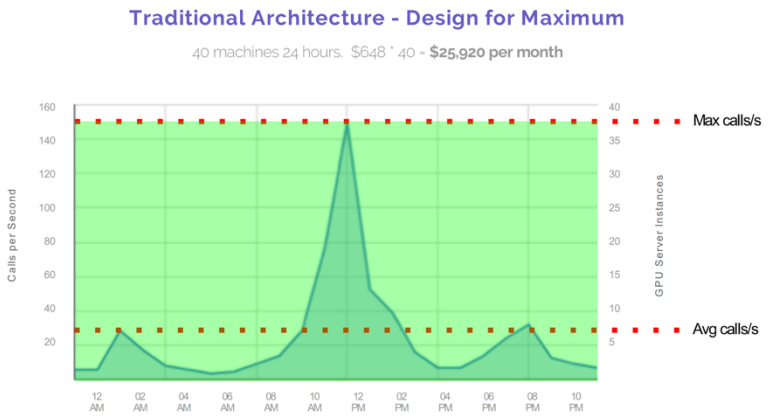
- Want to be able to reproduce results → versioning
- Code: Git
- Data: How?
  - New version for every new data point, every experiment?
  - How are versions saved efficiently?
- Models: Different training results for changes in initialization, batch size, order of training data points, training hardware, ...
- Experiments: Hyperparameters, metrics, which data/model versions were used?
- ML pipeline: Order of (pre/post)processing steps

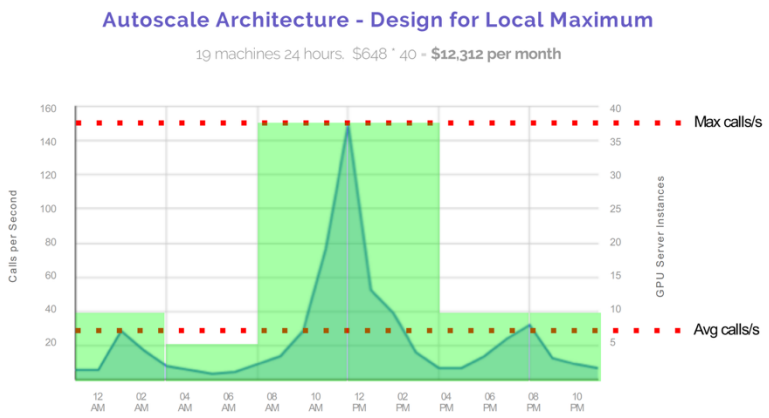
Challenges: Monitoring

- Data: distribution shift?
- Model
  - Metrics: does performance match expectations?
  - Numerical stability
  - Memory leaks
  - Speed of inference, GPU utilization, …
- Tools
  - Training: TensorBoard, Weights and Biases
  - Production: AWS CloudWatch

Challenges: Varying load

- When and how to add/remove servers/virtual machines
- Alternative: serverless (function as a service, FaaS)





Delegate more responsibility to infrastructure provider

MLOps tasks

- Automation
- Versioning
- Reproducibility
- Development
- Testing
- Deployment
- Monitoring
- Security & Access Management
- Governance

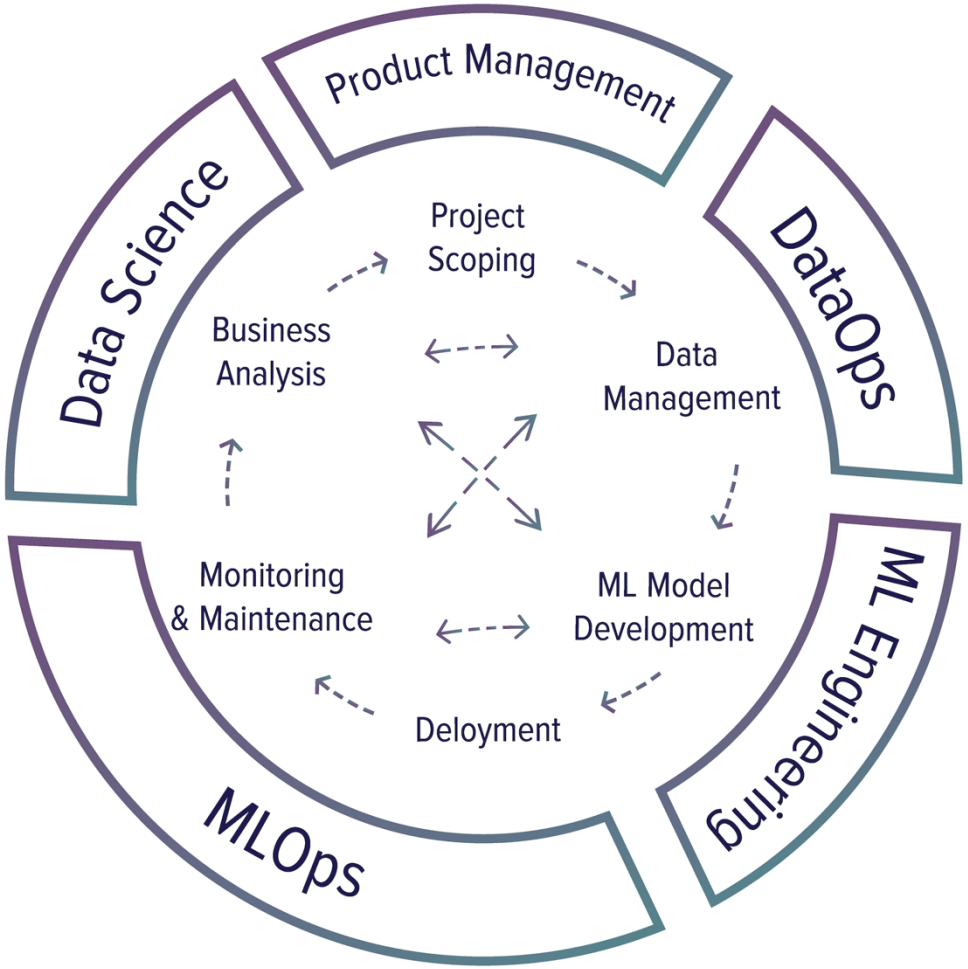
ML in production: Expectation

- Collect data
- Train model
- Deploy model

ML in production: Reality

- Choose a metric to optimize
- Collect data
- Train model
- Realize many labels are wrong -> relabel data
- Train model
- Model performs poorly on one class -> collect more data for that class
- Train model
- Model performs poorly on most recent data -> collect more recent data
- Train model
- Deploy model
- Dream about $$$
- Wake up at 2am to complaints that model biases against one group -> revert to older version
- Get more data, train more, do more testing
- Deploy model
- Pray
- Model performs well but revenue decreasing
- Cry
- Choose a different metric
- Start over

ML system lifecycle

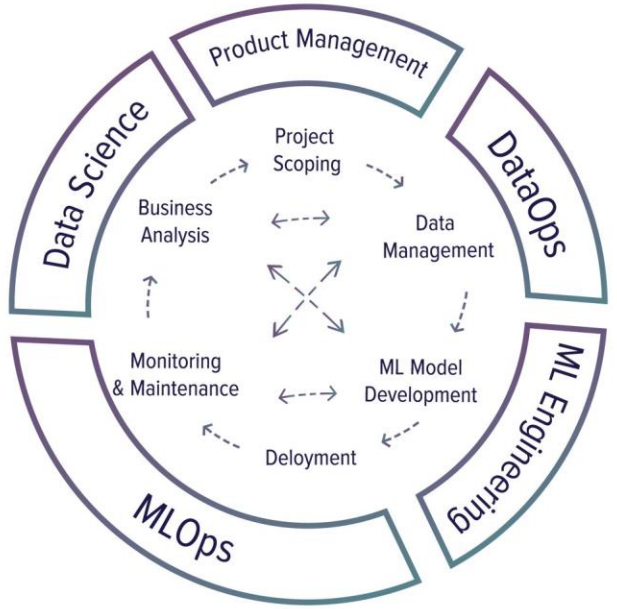


20% of time for initial development

80% for updates and maintenance

Stanford CS329s lecture notes

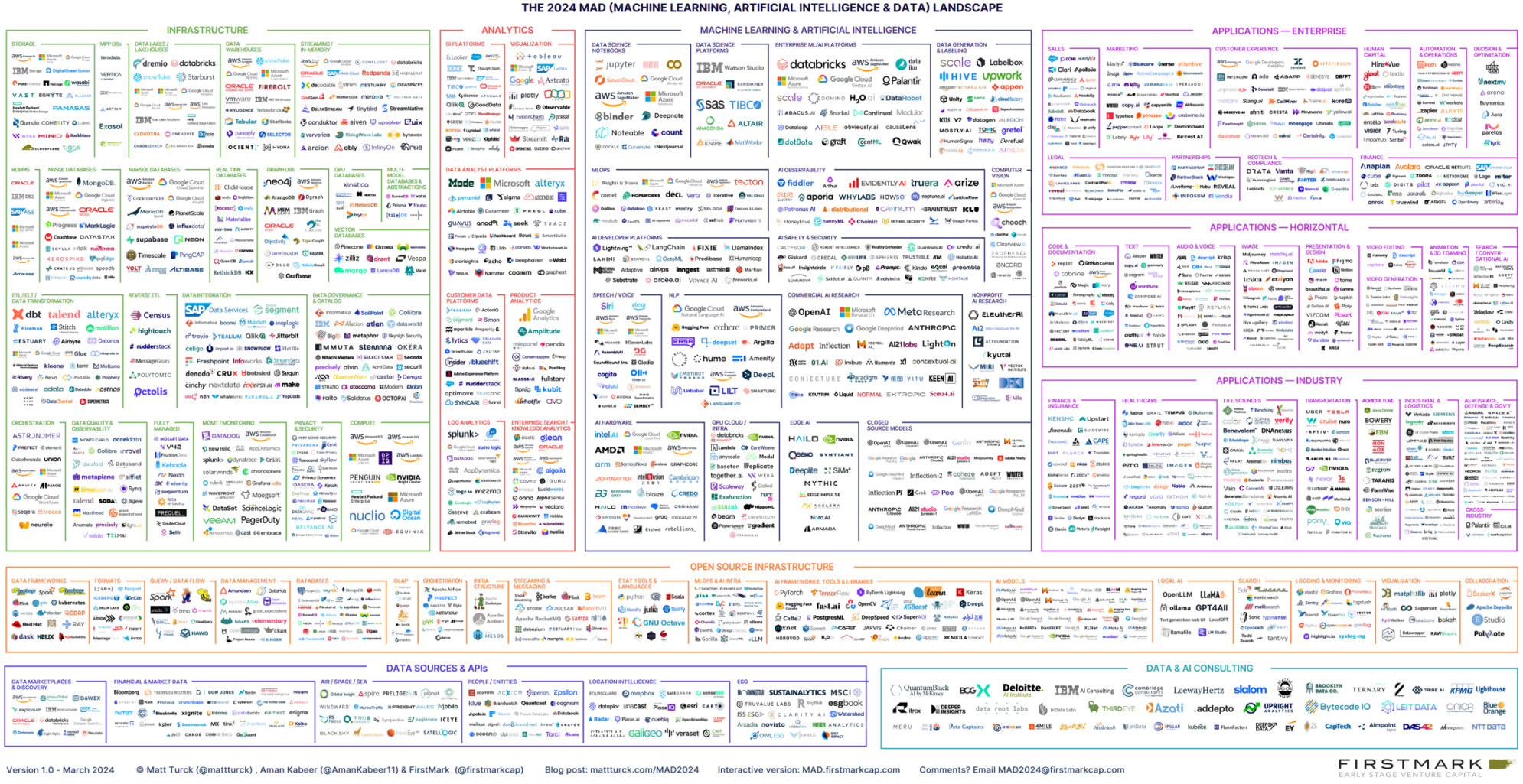
ML system lifecycle



1. Project scoping: identify goals, constraints, stakeholders, resources
2. Data management/engineering: preprocessing, curating, storing
3. ML model development: feature engineering, model selection, training, evaluation
4. Deployment: make model results accessible
5. Monitoring and maintenance: track metrics, update model (maybe with continual learning)
6. Business analysis: business goals satisfied?

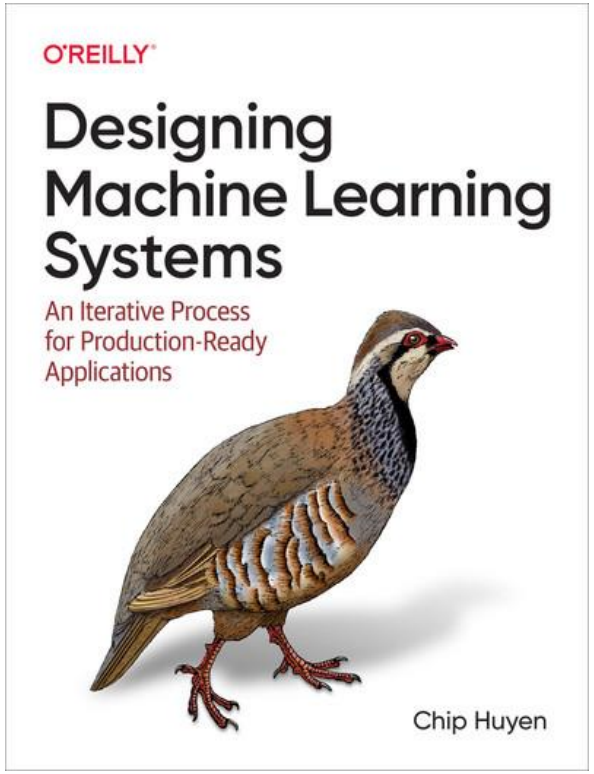
Stanford CS329s lecture notes

ML, AI and Data (MAD) landscape



Course Book

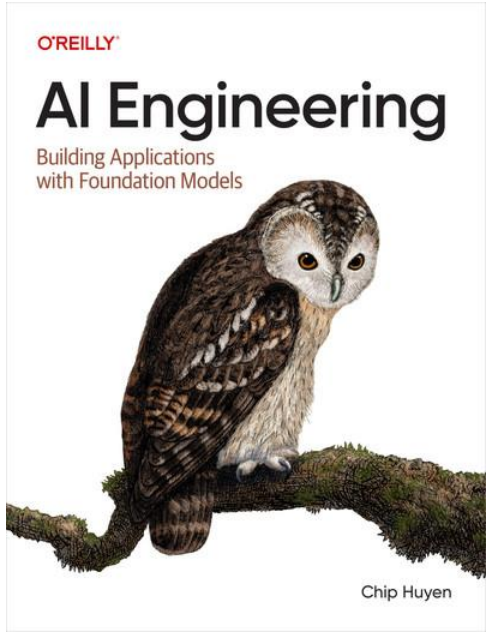
- Designing Machine Learning Systems by Chip Huyen (lecturer at Stanford, previously at Nvidia & Netflix)
  - sign in with your HSLU email address > Bibliotheken der Hochschule Luzern



- This course is also based on Stanford course CS329s

Further reading

- New book by Chip Huyen (Jan 2025): AI Engineering



- Stanford MLSys seminars
- ML engineering for production course on Coursera or DeepLearning.AI
- Full Stack Deep Learning course
- MLOps - tools, best practices, and case studies
- ml-ops.org (collection of principles and definitions/explanations)
- Big collection of MLOps article links
- Huge collection of open source MLOps libraries