MLOPS Will Change Machine Learning

MAGDALENA STENIUS - VALOHAI

About the speaker

- GitHub @magdapoppins
- Engineer at Valohai since 2019
- Valohai's mission is to accelerate AI adaption by building the best data science tools
- Valohai features such as Bayesian optimization and spot instance support

Agenda

1. Evolution and impact of DevOPS on software engineering
2. How MLOPS extends DevOPS practices to the machine learning context
3. Future visions for MLOPS
4. Demo time: setting up training, deployments and pipelines for a simple classifier

DevOPS

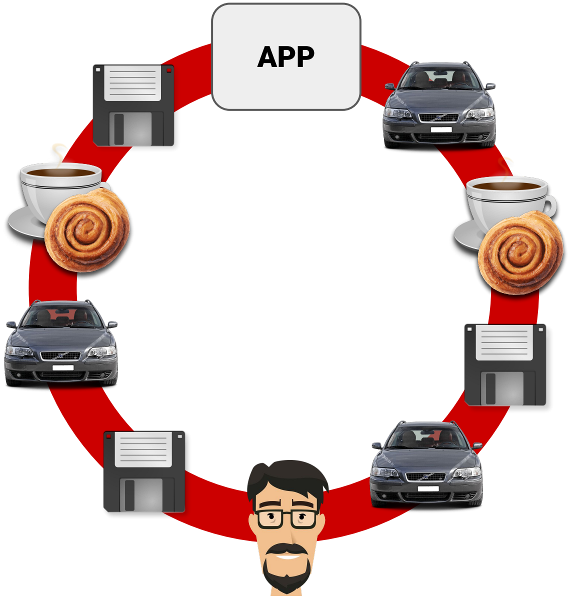
From experiment to production

- Needs to be fail-safe and reproducible
- Moving through environments
- Options for moving between versions
- Possibility for rapid rollback
- Security

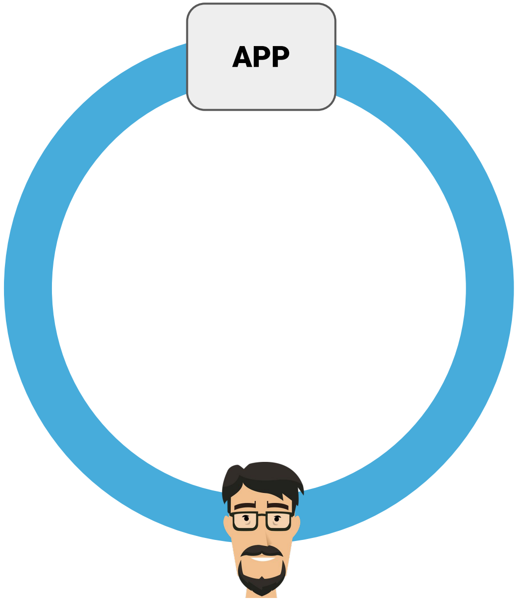
What is DevOPS

- Methodology of bringing together development and operations processes
- Automating different phases of the DevOPS lifecycle
- Aims
    - Continuous integration and continuous delivery for increased speed and quality assurance
    - Feedback and visibility through monitoring
    - Infrastructure reproducibility and consistency through infrastructure as code

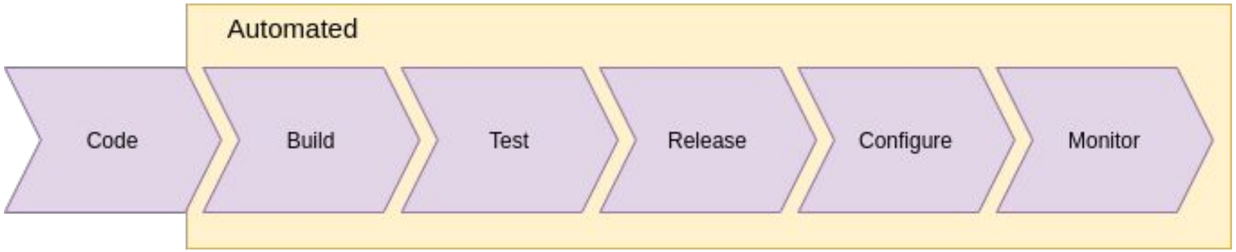
DevOPS in the 90s

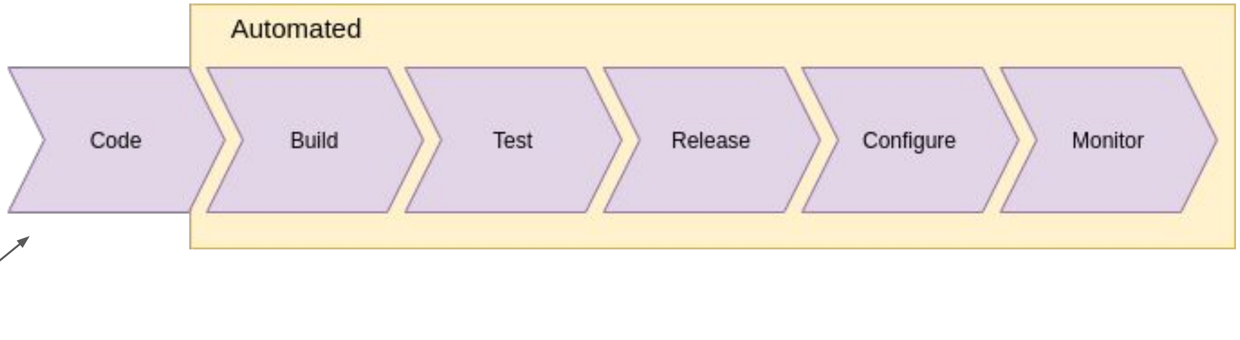


DevOps '95

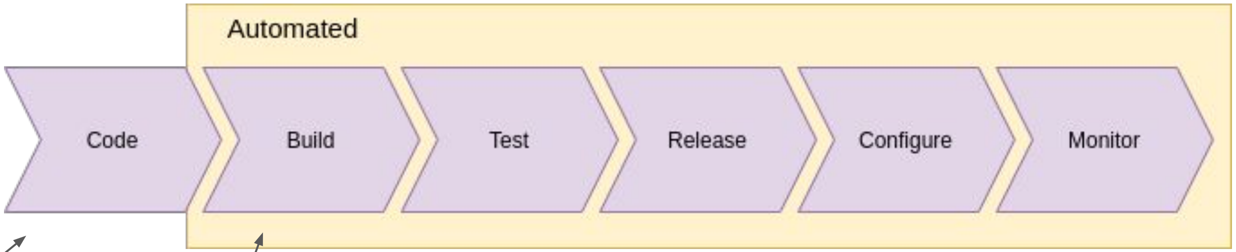


DevOps '21


Automated
Code
Build
Test
Release
Configure
Monitor

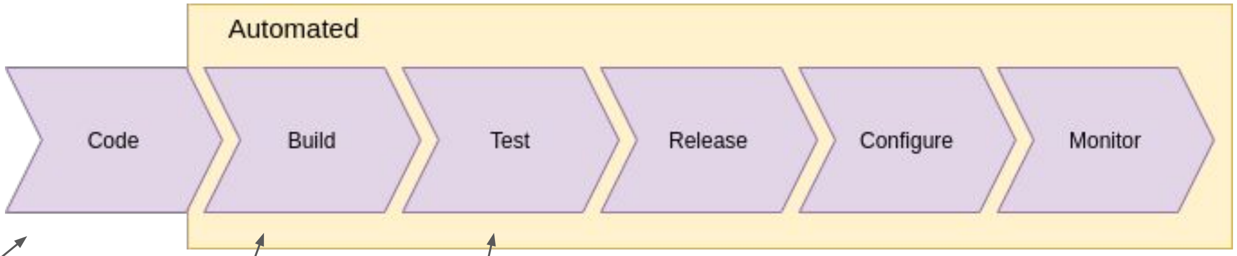


Code review tools
Code review policies
Version control
Remote repository
Test coverage goals



Code review tools
Code review policies
Version control
Remote repository
Test coverage goals

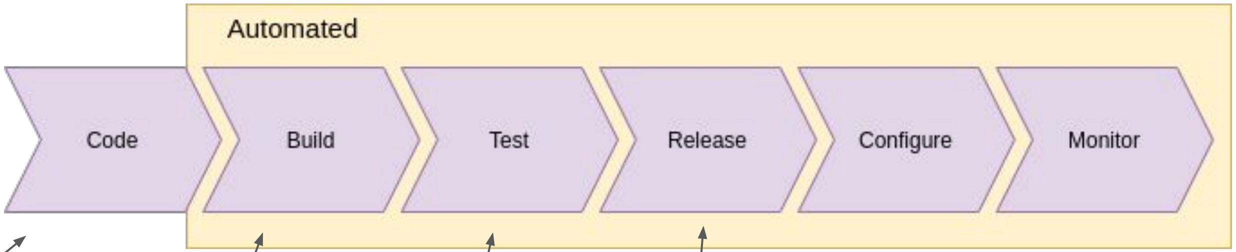
Creation and storage
of artifacts, e.g.
image



Code review tools
Code review policies
Version control
Remote repository
Test coverage goals

Creation and storage
of artifacts, e.g.
image

Test run configuration
Test environment

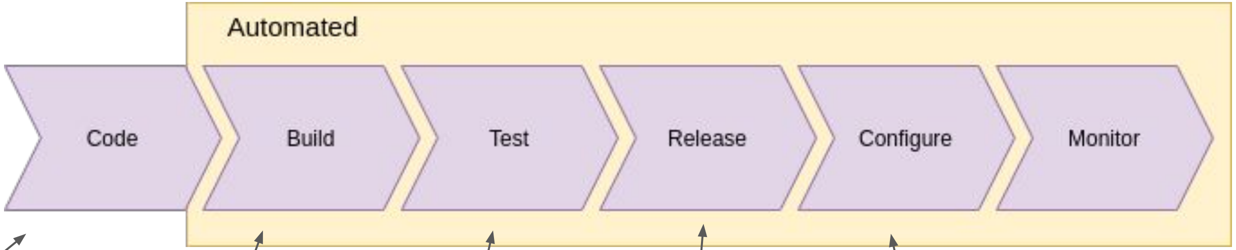


Code review tools
Code review policies
Version control
Remote repository
Test coverage goals

Creation and storage of artifacts, e.g. image

Environments used
Rollback strategy

Test run configuration
Test environment



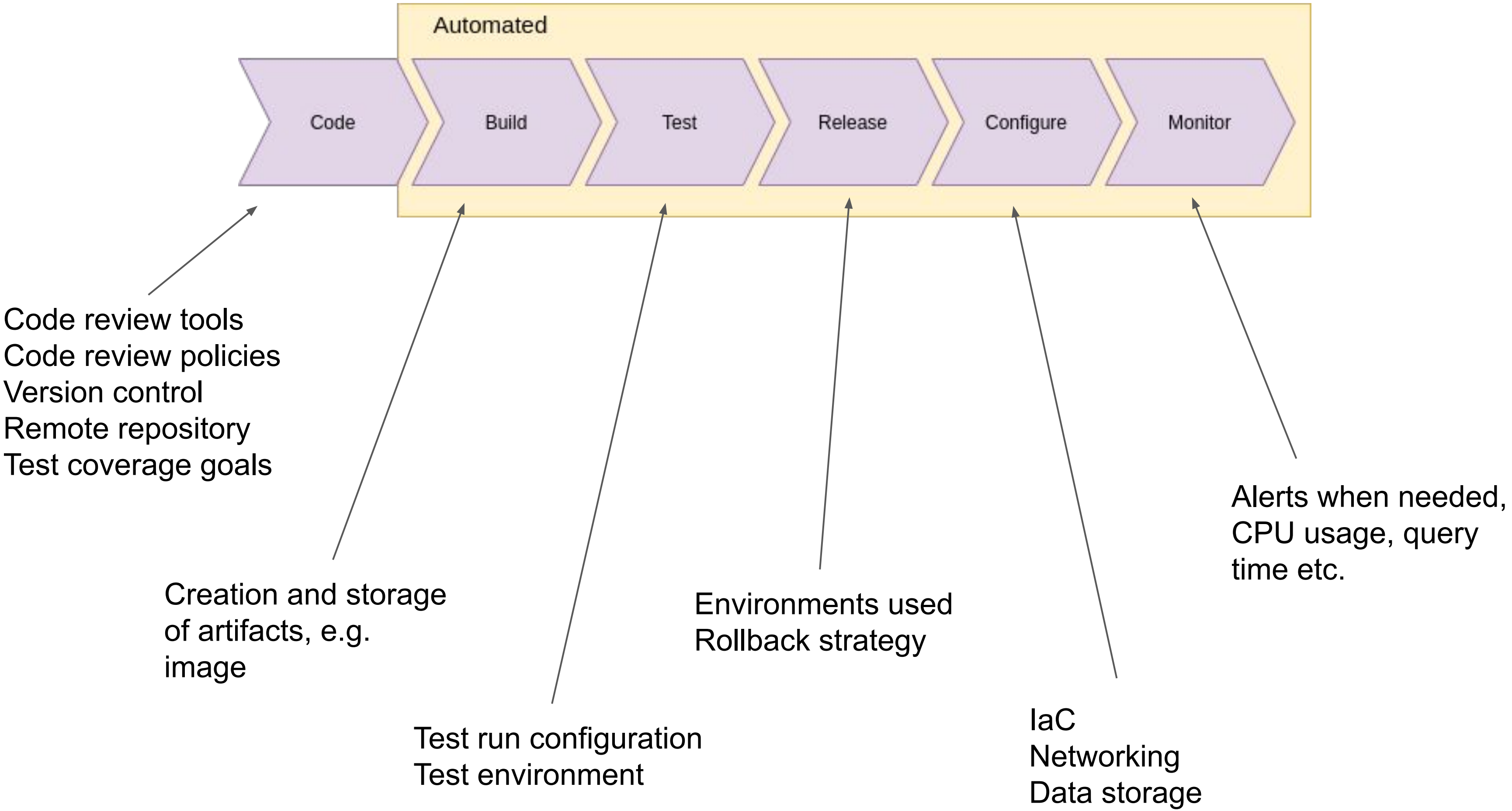
Code review tools
Code review policies
Version control
Remote repository
Test coverage goals

Creation and storage of artifacts, e.g. image

Environments used
Rollback strategy

Test run configuration
Test environment

IaC
Networking
Data storage


Automated
Code
Build
Test
Release
Configure
Monitor
Code review tools
Code review policies
Version control
Remote repository
Test coverage goals
Creation and storage of artifacts, e.g. image
Test run configuration
Test environment
Environments used
Rollback strategy
IaC
Networking
Data storage
Alerts when needed, CPU usage, query time etc.

Containers

- One of the main topics in DevOPS has been virtualization
- Addition of a virtual layer of hardware, os and storage enable apps to be run agnostic to the underlying system
- Os level virtualization is usually done using containers
- Containers are created using images containing everything that is needed to run the container (os, compilers, software)

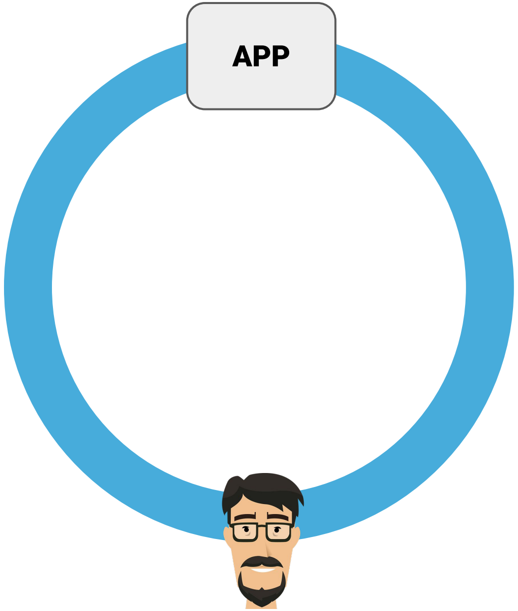
Container orchestration

- Multiple interconnected and dynamic container workloads
  - How to start containers?
  - Network addresses of different containers?
  - How to connect a container service with its storage?
- Kubernetes
  - Load balancing
  - Self-healing (automatic replacement of containers)
  - Automatic rollout or rollback of services

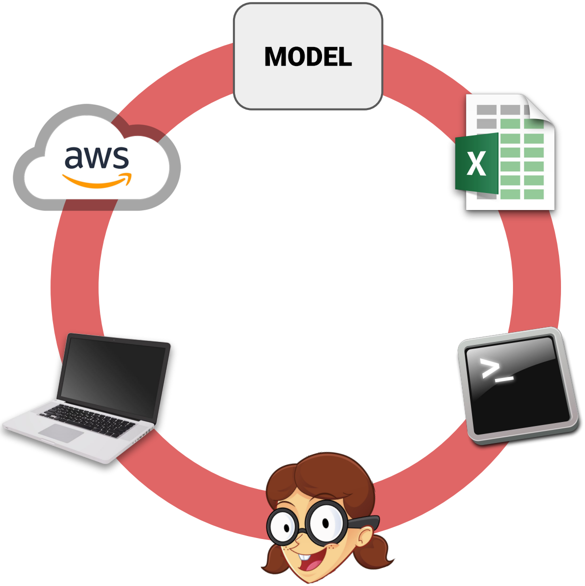
The impact of DevOPS

- Speed up software development cycle
- Increased quality due to replacing humans with automation
- Decrease in maintenance cost

MLOPS



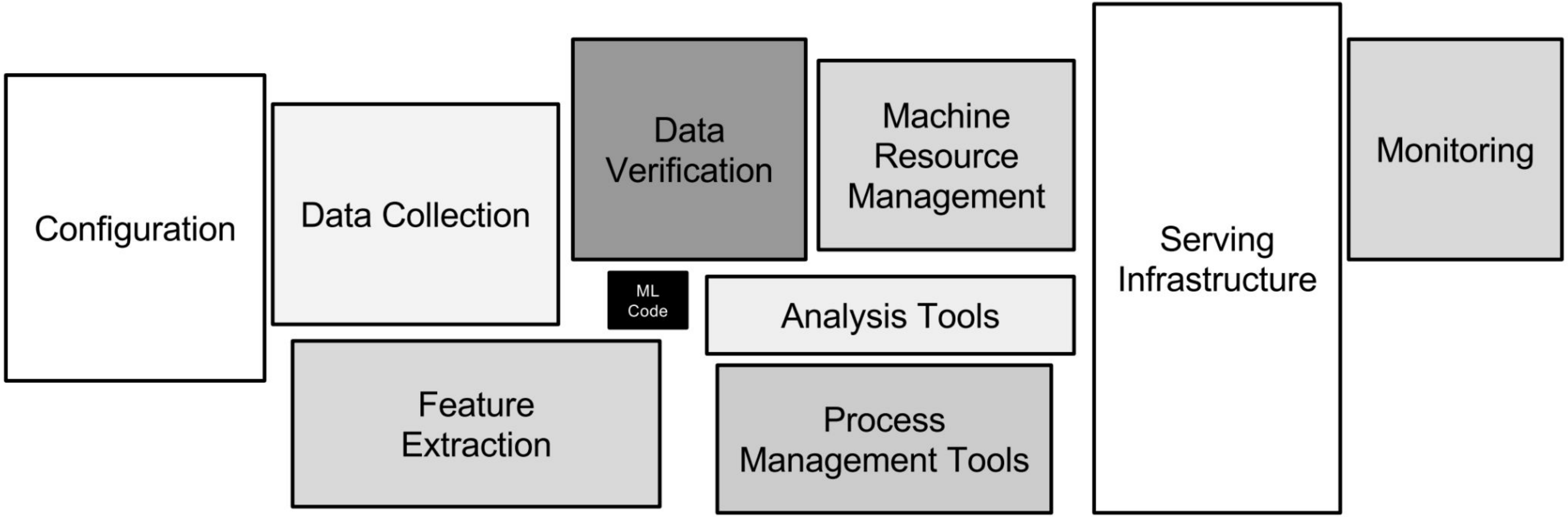
DevOps ‘21



MLOps ‘21

Why is ML development slow?

- Divergence of the scientist and engineering roles?
- The role of data



Sculley, D & Holt, Gary & Golovin, Daniel & Davydov, Eugene & Phillips, Todd & Ebner, Dietmar & Chaudhary, Vinay & Young, Michael & Dennison, Dan. (2015). Hidden Technical Debt in Machine Learning Systems. NIPS. 2494-2502.

Traditional software

Version control of code
Monitoring the application
Tests
Deployment
CI/CD pipeline

Software with ML elements

Version control of code
Monitoring the application
Tests
Deployment
CI/CD pipeline

Monitoring your data
Versioning data
Cyclical dependencies
Data validation
Model quality evaluation

Ingredients of the machine learning pipeline

- Data engineering
  - ETL
  - Feature engineering

Ingredients of the machine learning pipeline

- Data engineering
    - ETL
    - Feature engineering
- Training pipeline
    - Training and hyperparameter tuning
    - Evaluation of trained model metadata

Ingredients of the machine learning pipeline

- Data engineering
  - ETL
  - Feature engineering
- Training pipeline
  - Training and hyperparameter tuning
  - Evaluation of trained model metadata
- Serving
  - Making the model accessible for its end user, e.g. over TCP though an inference endpoint or via batch inference
  - Model monitoring

Outside of the pipeline?

- Exploration!

Artefacts of a MLOPS system

- Snapshot of code
- Data used for training
- Hyperparameters

For a reproducible result, all of these need to be version controlled.

Title	Sample Run Wednesday
Environment	Microsoft Azure F2s v2 (No GPU) (azure-westeurope-f2sv2)
Commit	4d34124 (4 mo ago)
Step	Train model (MNIST)
Image	tensorflow/tensorflow:1.13.1-py3
Command	python train.py {parameters}
Interpolated	python train.py --max_steps=300 --learning_rate=0.001 --dropout=0.9 --batch_size=200
Inputs	test-set-images https://valohaidemo.blob.core.windows.net/mnist/t10k-images-idx3-ubyte.gz test-set-labels https://valohaidemo.blob.core.windows.net/mnist/t10k-labels-idx1-ubyte.gz training-set-images https://valohaidemo.blob.core.windows.net/mnist/train-images-idx3-ubyte.gz training-set-labels https://valohaidemo.blob.core.windows.net/mnist/train-labels-idx1-ubyte.gz
Parameters	**batch_size** 200 **dropout** 0.9 **learning_rate** 0.001 **max_steps** 300
Created	2 minutes ago by **magda**
Executor	azwesteuropef2sv2-poeomurh
Duration	32 seconds
Price	US$0.00090657
Tags	Select...

Proprietary solutions

Solutions offered by large cloud providers, e.g. sagemaker.

- Upside: ease of use
- Downside: heavy vendor lock-in
- Downside: not available on-premise
- Downside: limited customization

The alternative: build MLOPS yourself?

- Skillset mismatch between data scientists and ops

- Cost of time

- Cost of work happiness

- Crafting one pipeline => managing a system with multiple production pipelines

The Future?

MLOPS today

- Cloud resources and accelerators (GPU/TPU/FPGA) in training
- Automated deployment
- Distinct staging and production environments
- Model monitoring
- Manual feature engineering/data processing
- Manual model tweaking
- Continuous Integration/Continuous Delivery/Continuous Training

MLOPS tomorrow

- Feature stores
- Automated retraining
- Self-healing systems (reacting to monitoring, e.g. data drift)
- Self-improving systems
- From manual feature engineering to self-learning from raw data
- Automated hyperparameter tuning
- Training-while-consuming

Data Centric

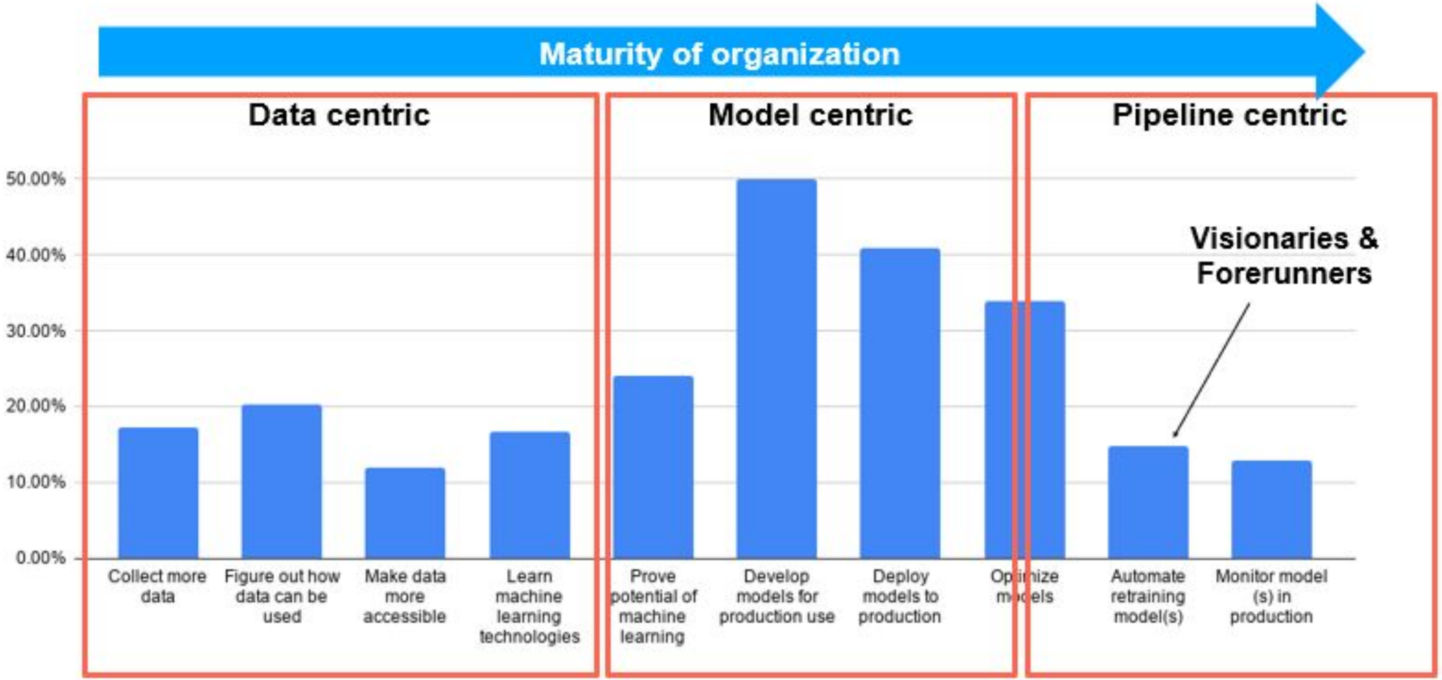
- Taking the first steps towards using ML
- Data preparation and engineering takes up lots of time
- Managing data is the largest pain
- Data lakes

Model Centric

- Data is available and ready for use in ML
- Using notebooks to achieve first results and initial versions of models
- Model deployment requires new systems to be built or purchased

Pipeline Centric

- First models have proven value but need to be maintained
- Models need to be tweaked and retrained
- Teams are growing and new members need to be onboarded



Our ML in 2020 questionare N=~350 data scientists in actual companies
“What are you trying to accomplish in the next 3 months? (Max 3)”

Concluding remarks: what change will MLOPS bring?

- Significant speed-up in model development and delivery iterations
- While DevOPS is dependent on a manual development process, ML can be automated even further
- The data scientists role shifting even further into understanding and explaining automated systems