Unbabel

The rise of MLOps

Catarina Silva

This presentation was not AI generated, yet. Bear with us.

The rise of MLOps

- ML in real world
- MLOps problems
- MLOps tools
- MLOps maturity

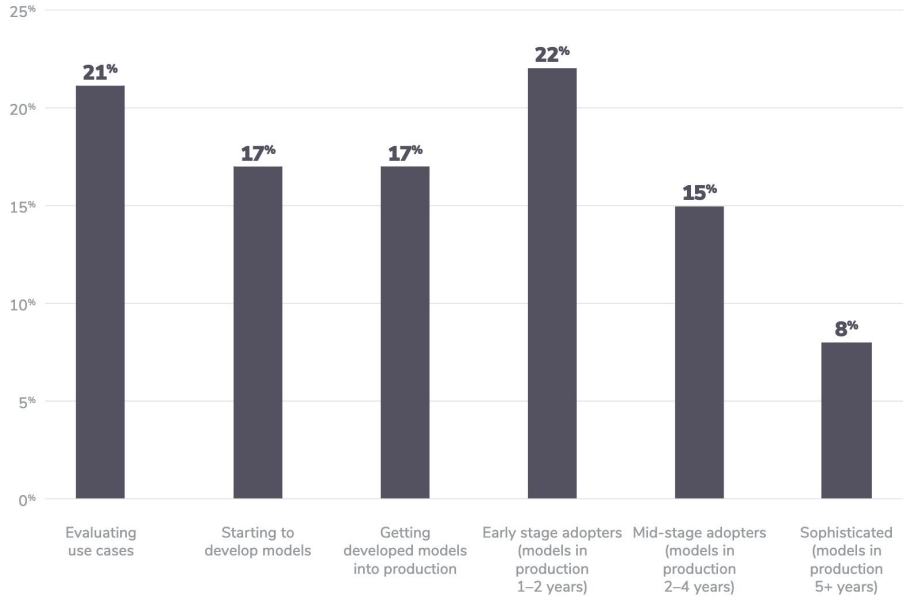
The era of Machine Learning

- Recent improvements in several different fields
- New techniques to improve resource usage
- Huge economical motivation
- Hype - AI for the sake of AI

The era of Machine Learning

Influx of new job titles such as

- Machine Learning Engineer
- ML Developer
- ML Architect
- Data Engineer
- ML Operations
- AI Ops



https://info.algorithmia.com/hubfs/2019/Whitepapers/The-State-of-Enterprise-ML-2020/Algorithmia_2020_State_of_Enterprise_ML.pdf

Around 90% of machine learning projects never make it into production

Main blockers for ML Success

- Data comes with its own set of issues

Main blockers for ML Success

- Data comes with its own set of issues
- Models have different sets of requirements

Main blockers for ML Success

- Data comes with its own set of issues
- Models have different sets of requirements
- There is a gap between research processes and production-focused processes

Main blockers for ML Success

- Data comes with its own set of issues
- Models have different sets of requirements
- There is a gap between research processes and production-focused processes
- AI is currently like a one hit wonder - deploy once with no follow up

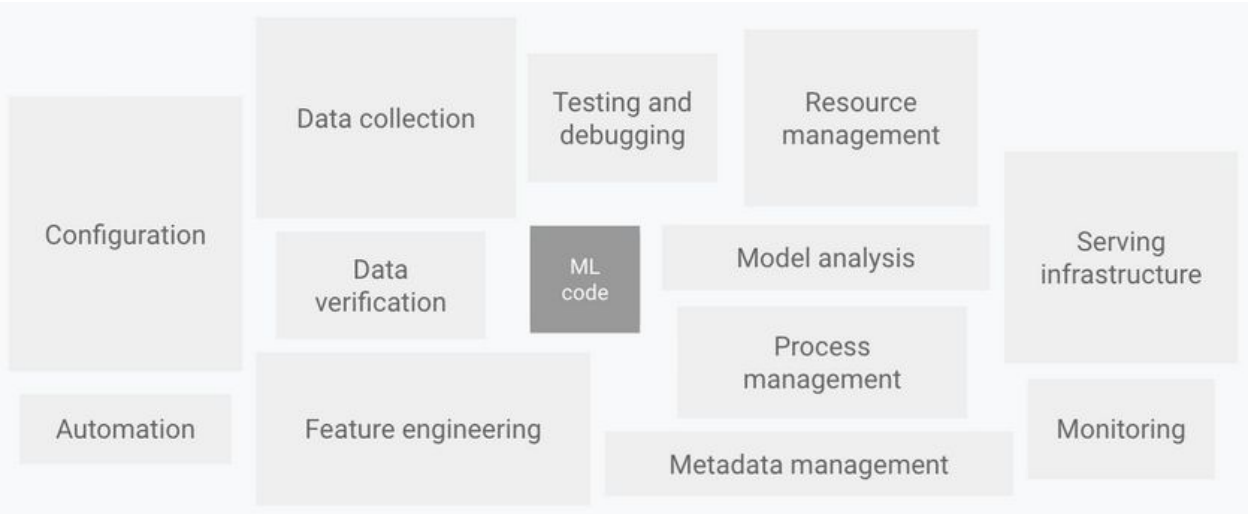
Main blockers for ML Success

- Data comes with its own set of issues
- Models have different sets of requirements
- There is a gap between research processes and production-focused processes
- AI is currently like a one hit wonder - deploy once with no follow up
- Get one of the pieces wrong, and the others won't fit
    - High interest debt accumulator

Main blockers for ML Success

- Data comes with its own set of issues
- Models have different sets of requirements
- There is a gap between research processes and production-focused processes
- AI is currently like a one hit wonder - deploy once with no follow up
- Get one of the pieces wrong, and the others won't fit
    - High interest debt accumulator
- Silos between data scientists building models and remaining teams
    - People are very attached to their ways of working

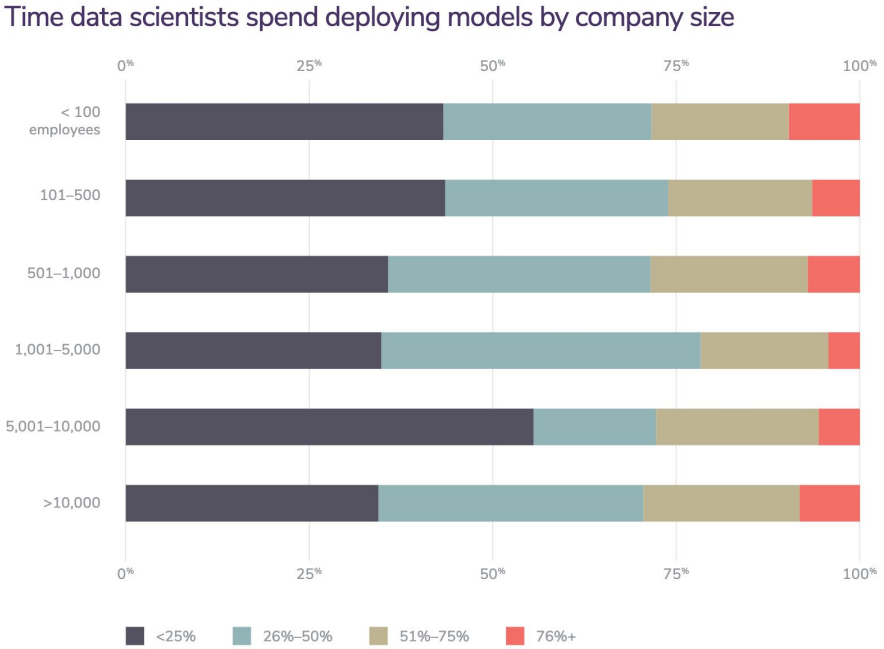
Main blockers for ML Success



https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Main blockers for ML Success

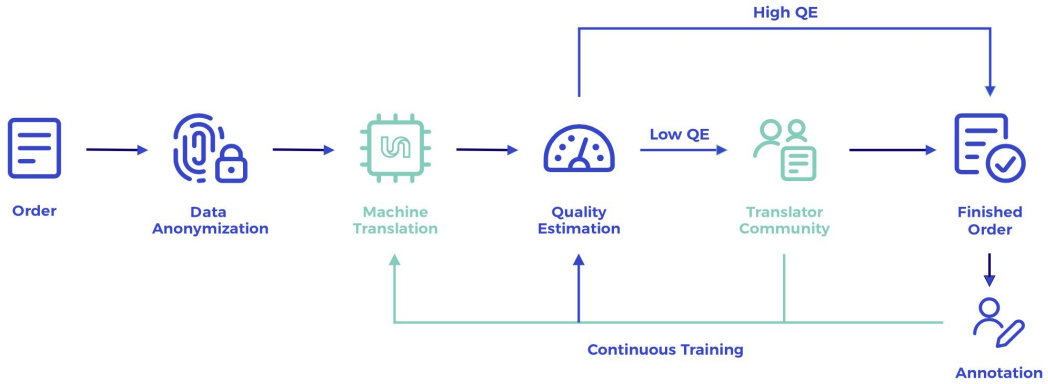
- Deploying a model should be as easy as deploying code
- DevOps brought a change to code deployment, can MLOps bring the same change?



https://info.algorithmia.com/hubfs/2019/Whitepapers/The-State-of-Enterprise-ML-2020/Algorithmia_2020_State_of_Enterprise_ML.pdf

Where I’m coming from

- A company where AI is one of the central pieces
- We have high cardinality of models (several models / several different projects)
- We’ve scaled and are still scaling - growing pains



Where I’m coming from

- I’m an instructor here
- Teaching model deployment and other concepts around
- Much much lower scale
- Completely different type of applications

LISBON DATA SCIENCE ACADEMY

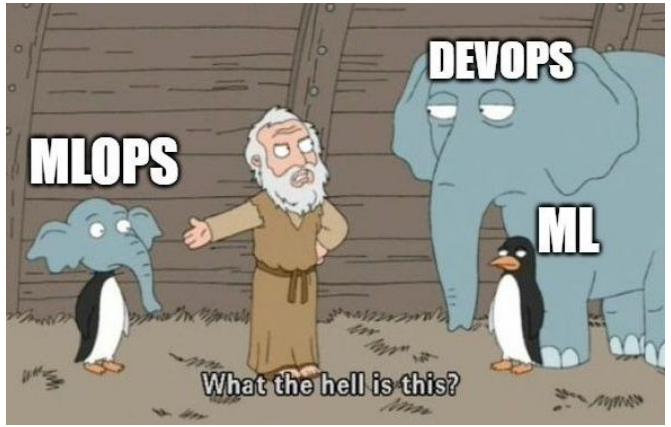
Specialization #6 - Data Science in the Real World			
BLU13 Basic model Deployment	BLU14 Deployment in the real world	BLU15 Model CSI	Hackathon #6

What is MLOps?

What is MLOps ?

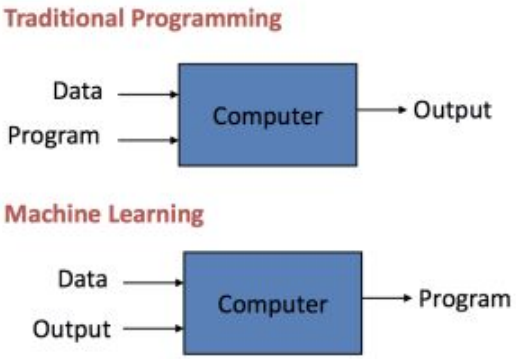
MLOps is the set of Machine Learning and Engineering culture and practices that have as a goal to unify the development and operation of Machine Learning systems.

Similarly to devops, it aims to shorten the systems development life cycle and increase an organization's ability to deliver high quality machine learning systems quickly and regularly.



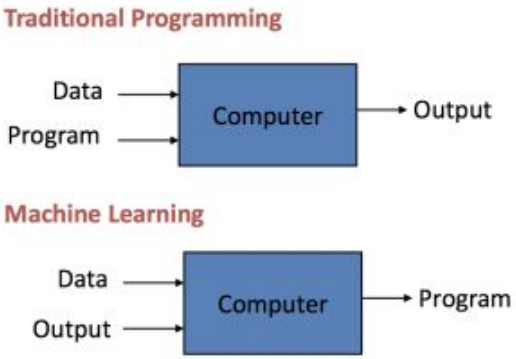
MLOps differs from DevOps where Machine Learning differs from Software Engineering

Traditional vs. ML Engineering



https://machinelearningmastery.com/basic-concepts-in-machine-learning/

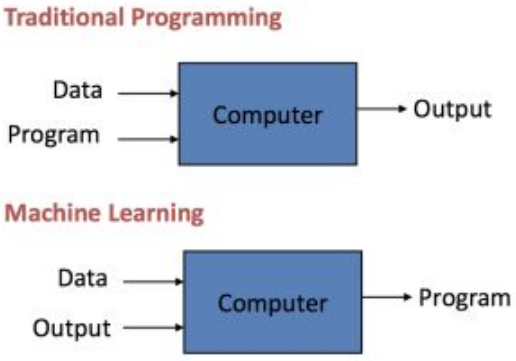
Traditional vs. ML Engineering



- ✔ Desired behavior is well defined
- ✔ Program is well known

https://machinelearningmastery.com/basic-concepts-in-machine-learning/

Traditional vs. ML Engineering



https://machinelearningmastery.com/basic-concepts-in-machine-learning/

✔️ Desired behavior is well defined

✔️ Program is well known

⚠️ Desired behavior is not 100% defined

⚠️ Program is the result of the algorithm

Traditional Software Engineering

SoftEng

Traditional Software Engineering

SoftEng

- ✔ Continuous development
- ✔ Continuous testing
- ✔ Continuous integration

Traditional Software Engineering

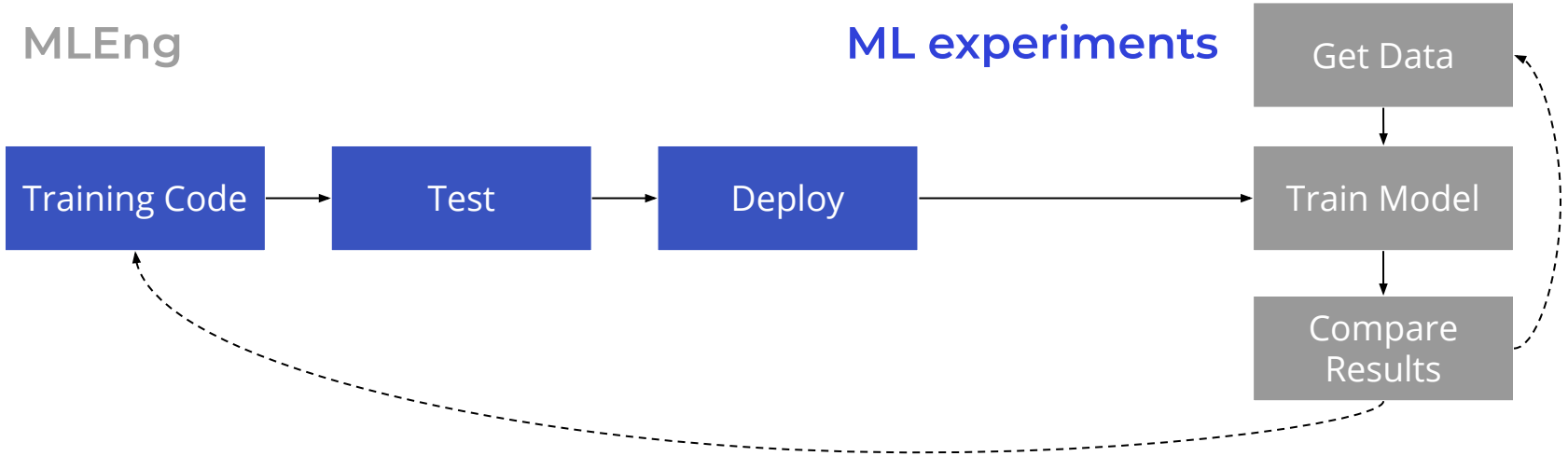
SoftEng

- ✔ Continuous development
- ✔ Continuous testing
- ✔ Continuous integration
- ✔ Continuous deployment
- ✔ Continuous monitoring
- ✔ Infrastructure as code

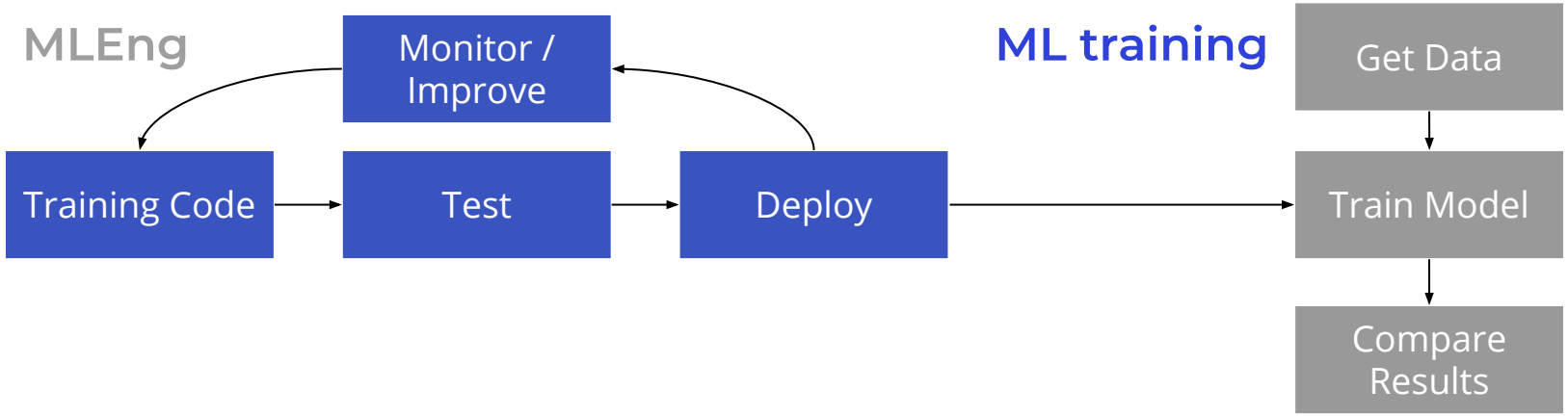
Machine Learning Engineering

MLEng

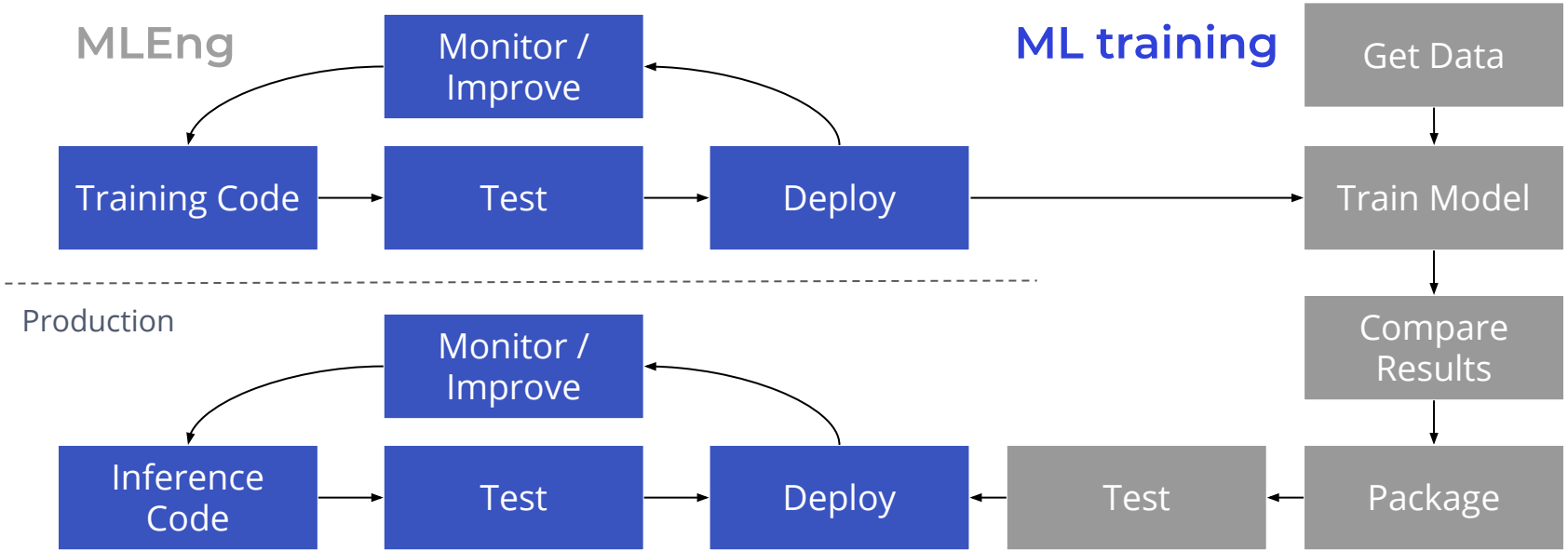
Machine Learning Engineering



Machine Learning Engineering



Machine Learning Engineering



Machine Learning is essentially experimental

Models are often black boxes

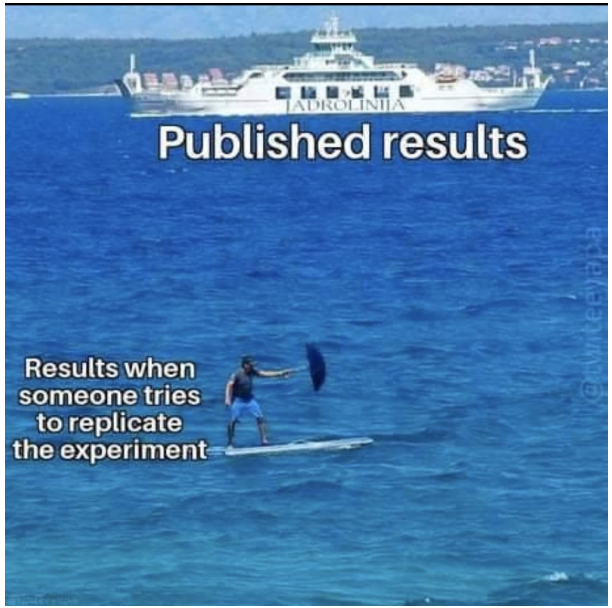
- You can’t differentiate between two models, so how do you compare them?
- Evolving surroundings have an impact on performance

https://xkcd.com/1838/

Models are hard to scale

- If a model works great in a small environment it doesn't mean it will work great in any case
- Usually datasets are very big
- Models are also somewhat big, and getting bigger everyday
- Inference can be slow

Models are hard to reproduce



Models are a combination of data, parameters and algorithms, including random seed pick

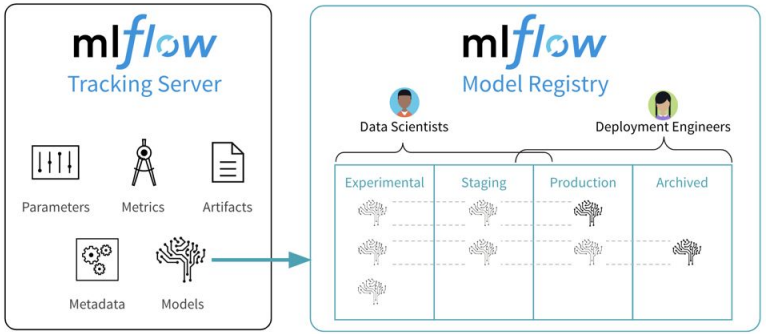
Results may not scale well from one scenario to the other, even when they are close

Reproducibility leads to efficiency

- ✔️ Experiment tracking
- ✔️ Model and dataset management
- ✔️ Data Provenance

Reproducibility leads to efficiency

- ✔ Experiment tracking
- ✔ Model and dataset management
- ✔ Data Provenance

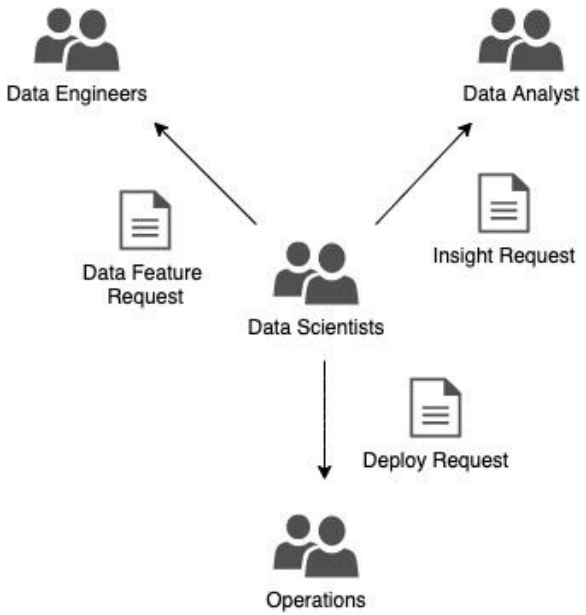


https://databricks.com/product/managed-mlflow

Models are only as good as the data they see

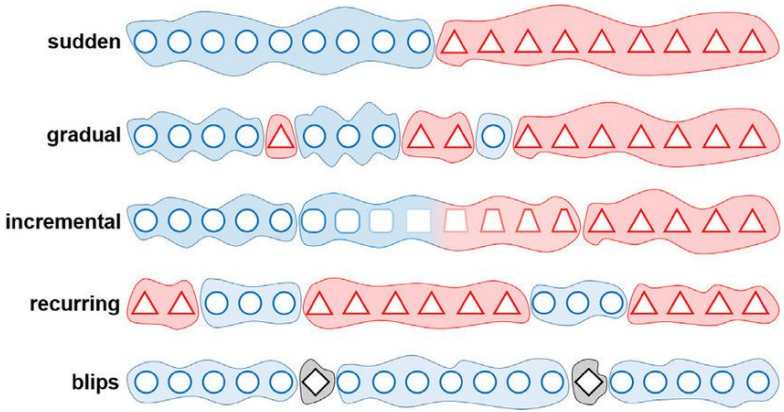
Data ingestion

- Data comes from many different sources
- Sometimes there are processes that block access to data
- A solid data platform can be a game changer for data science teams



Model Monitoring

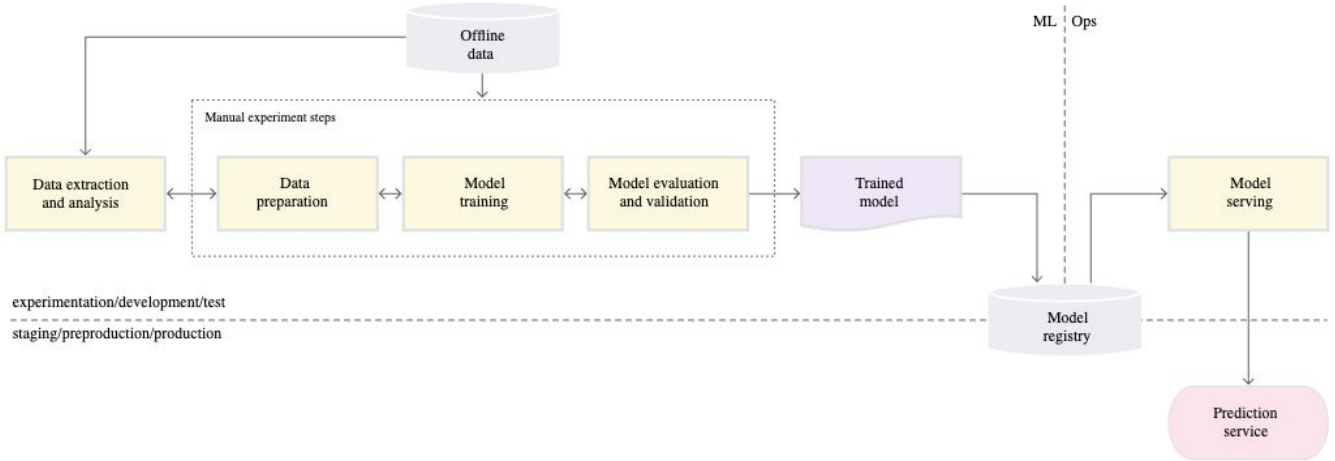
- Real world data is constantly changing
- **Model Degradation** can be caused by data shifts



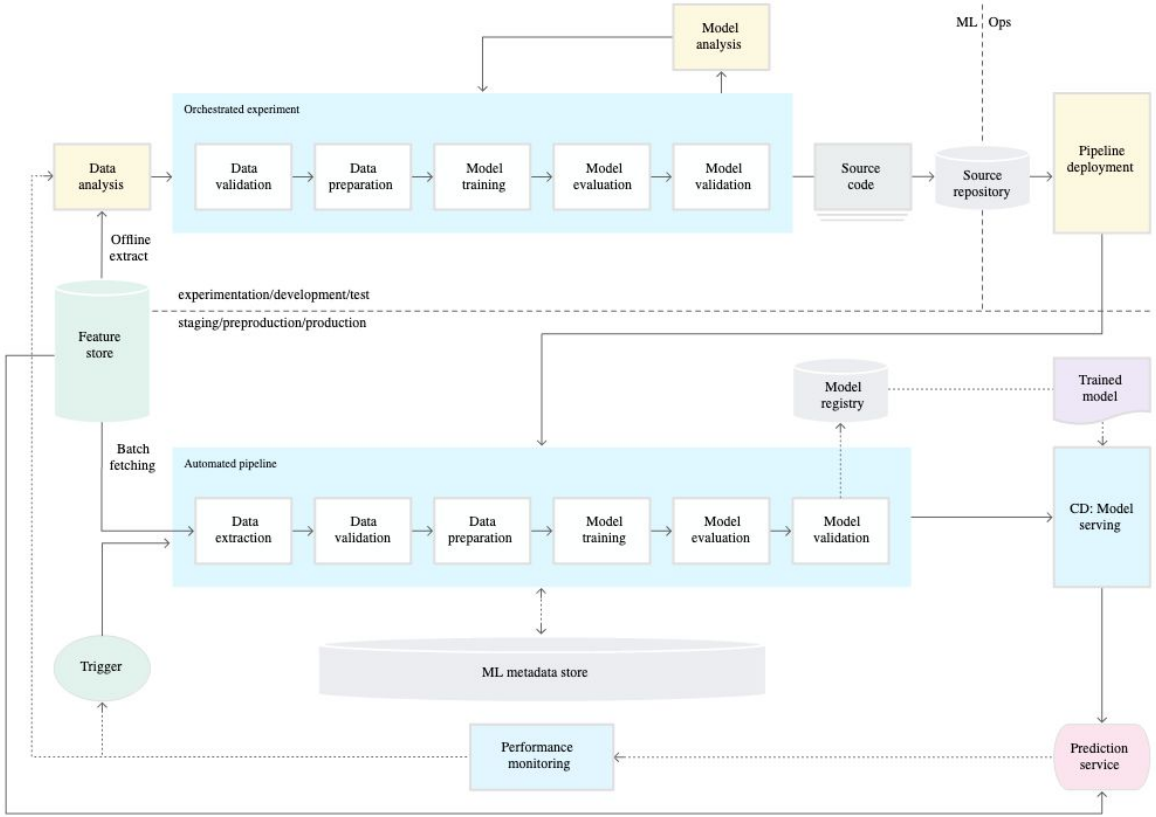
Krawczyk, Bartosz & Cano, Alberto. (2018). Online Ensemble Learning with Abstaining Classifiers for Drifting and Noisy Data Streams. Applied Soft Computing. 68. 677-692. 10.1016/j.asoc.2017.12.008.

Continuous Training

A regular experimental pipeline, if well established, is easy to run manually. However, if we just want to re-run every day/week/etc with the newest data, it becomes inefficient.



https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning



https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Training infrastructure

- Most machine learning applications need different infrastructure requirements - eg. GPU
- Usually costly and has peaks of usage
- Typically shared among several researchers and relies on some queue system

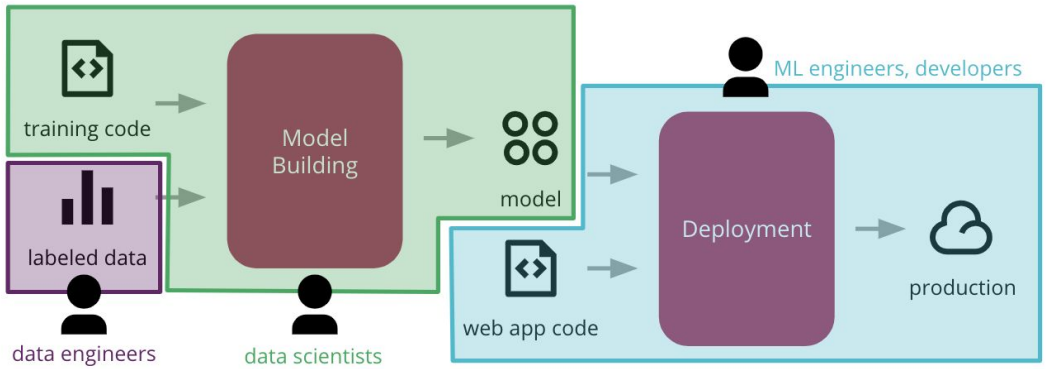
Continuous Training is key

- ✔️ Data accessibility and Platform thinking
- ✔️ Continuous training and pipeline monitoring
- ✔️ Continuous ML monitoring
- ✔️ Model degradation detection
- ✔️ Training infrastructure management 💰 💰

It takes a village to deliver successful machine learning projects

MLOps

Putting all these tools and practices in place is not the work only of ML Engineers, but of the whole team. This means that ML engineers are not solely responsible by these pipelines, but are empowered to do their job better and are more aware and accountable for the models they produce.



https://martinfowler.com/articles/cd4ml.html

MLOps

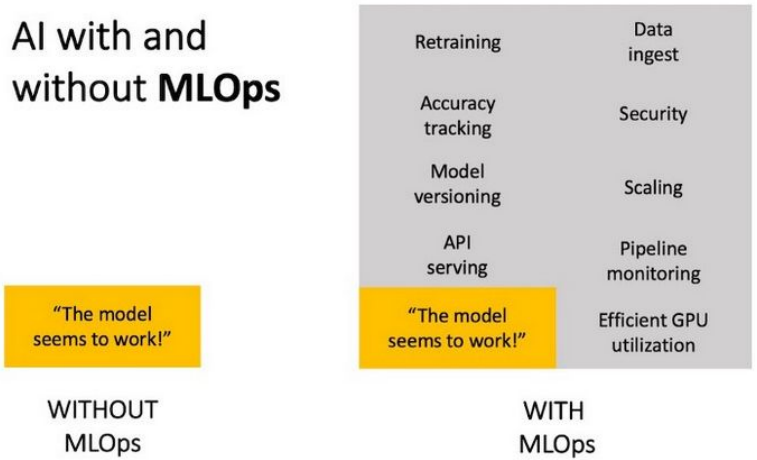
- ☑️ Continuous development
- ☑️ Continuous testing
- ☑️ Continuous integration
- ☑️ Continuous deployment
- ☑️ Continuous monitoring
- ☑️ Infrastructure as code

- ✔️ Experiment tracking
- ✔️ Asset management
- ✔️ Data Provenance
- ✔️ Platform thinking
- ✔️ Continuous training
- ✔️ Continuous ML monitoring
- ✔️ Model degradation detection
- ✔️ Training infrastructure management

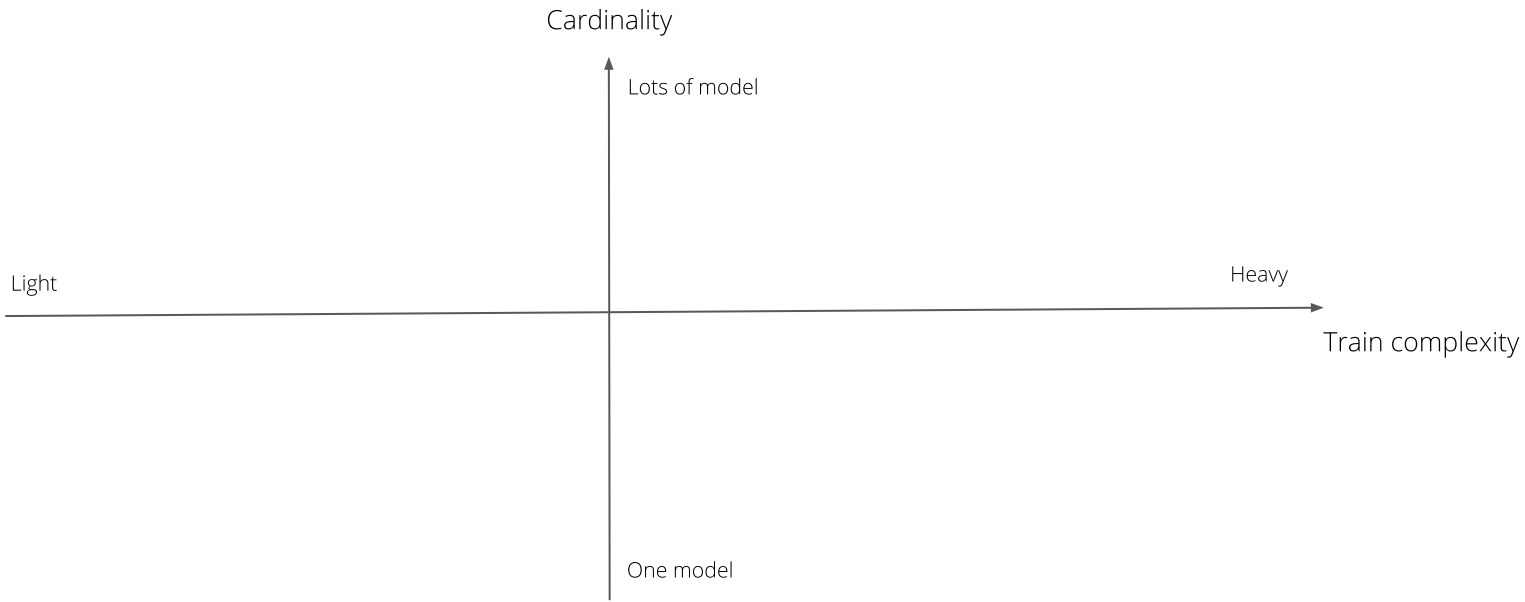
How can I start using MLOps?

Importance of MLOps for you

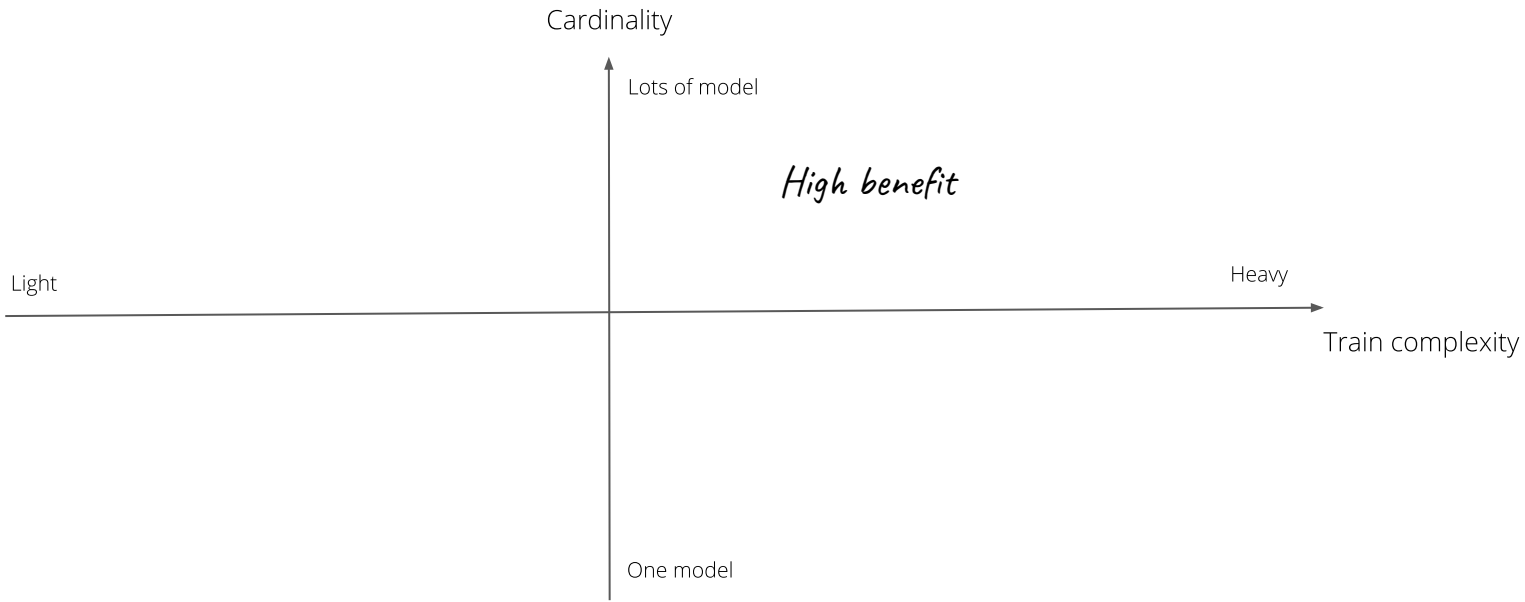
- MLOps can bring high value, but there is not a one-size-fits-all
- Different maturity levels, different approaches
- There's a trade-off between the value it brings and the costs of implementation



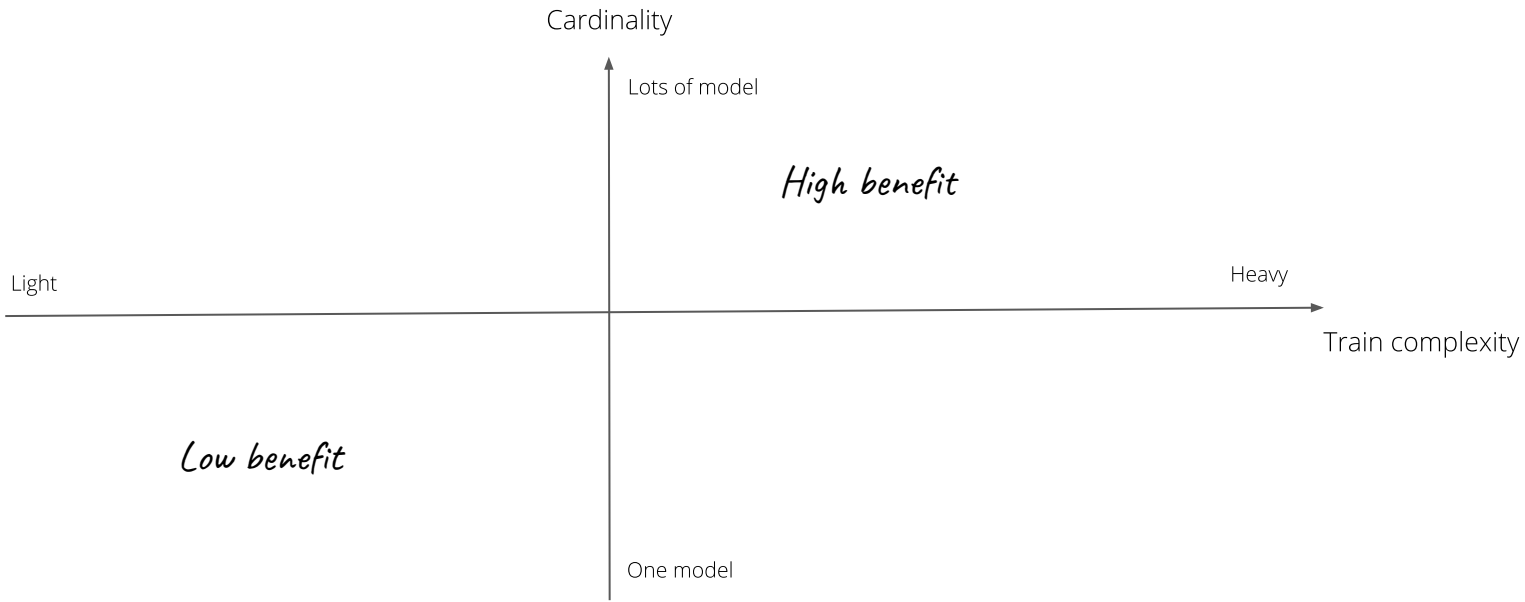
Importance of MLOps for you



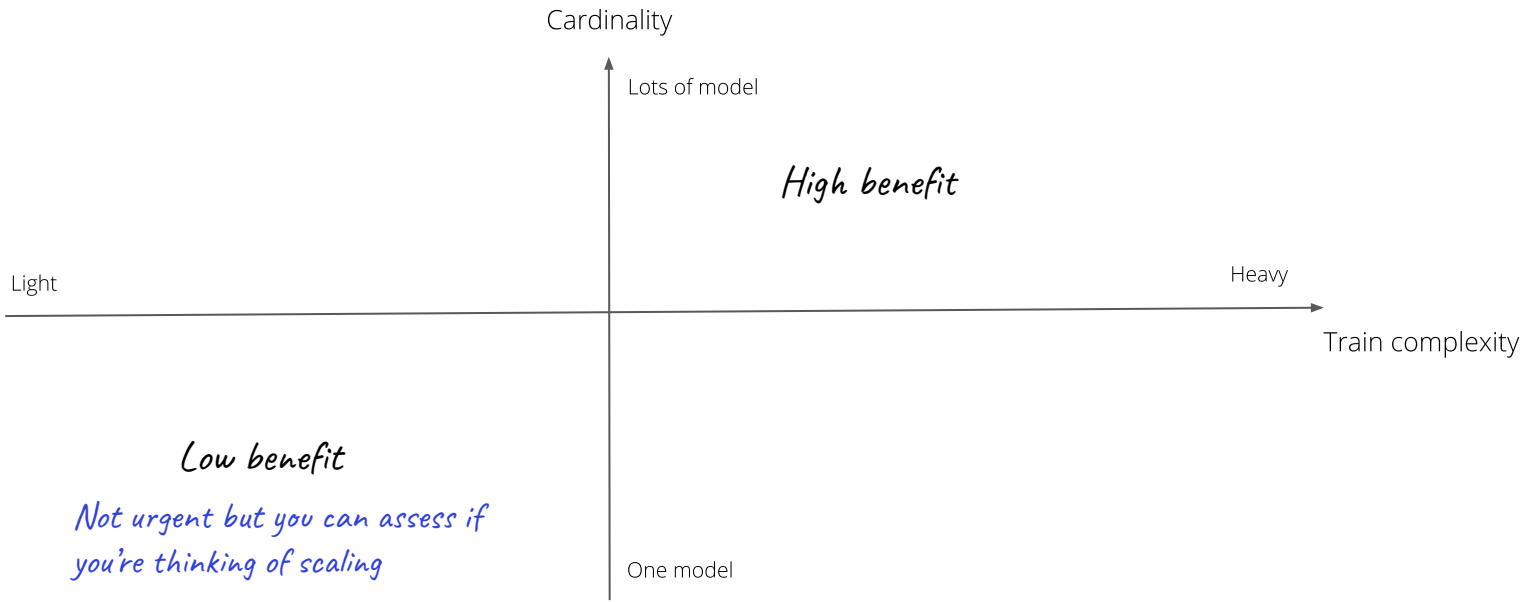
Importance of MLOps for you



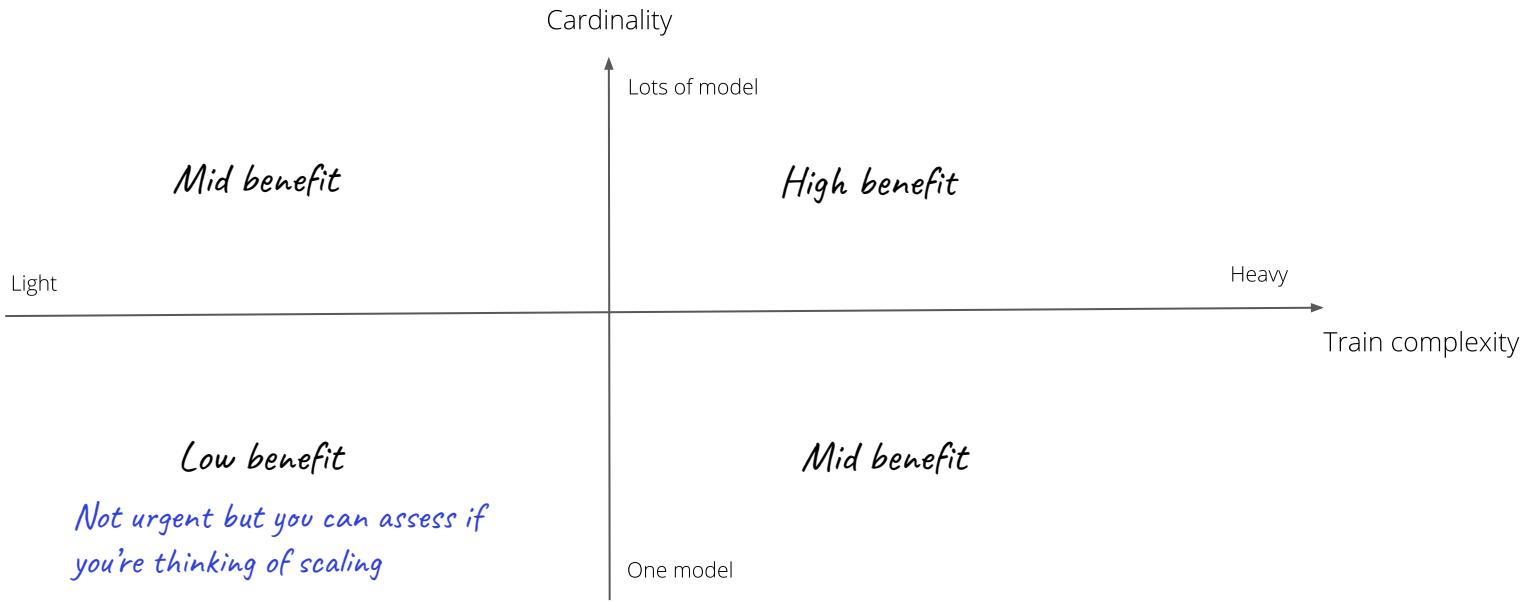
Importance of MLOps for you



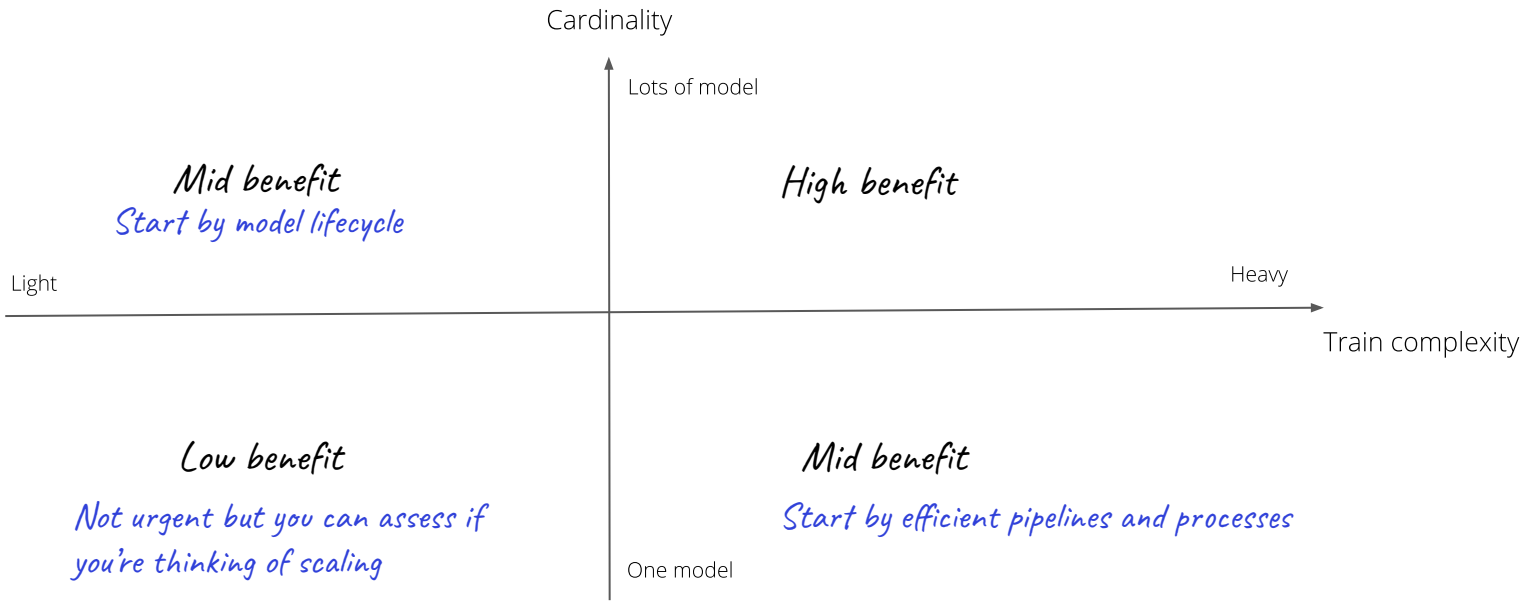
Importance of MLOps for you



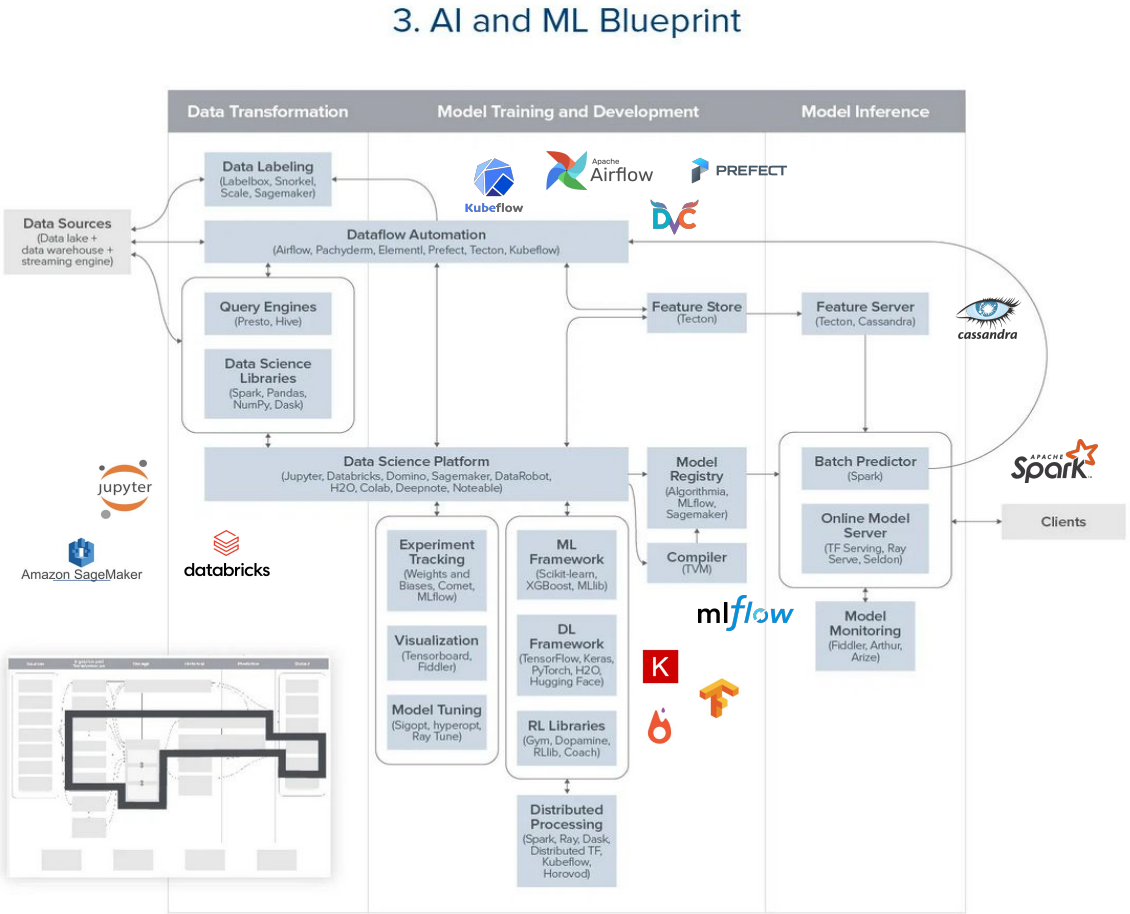
Importance of MLOps for you



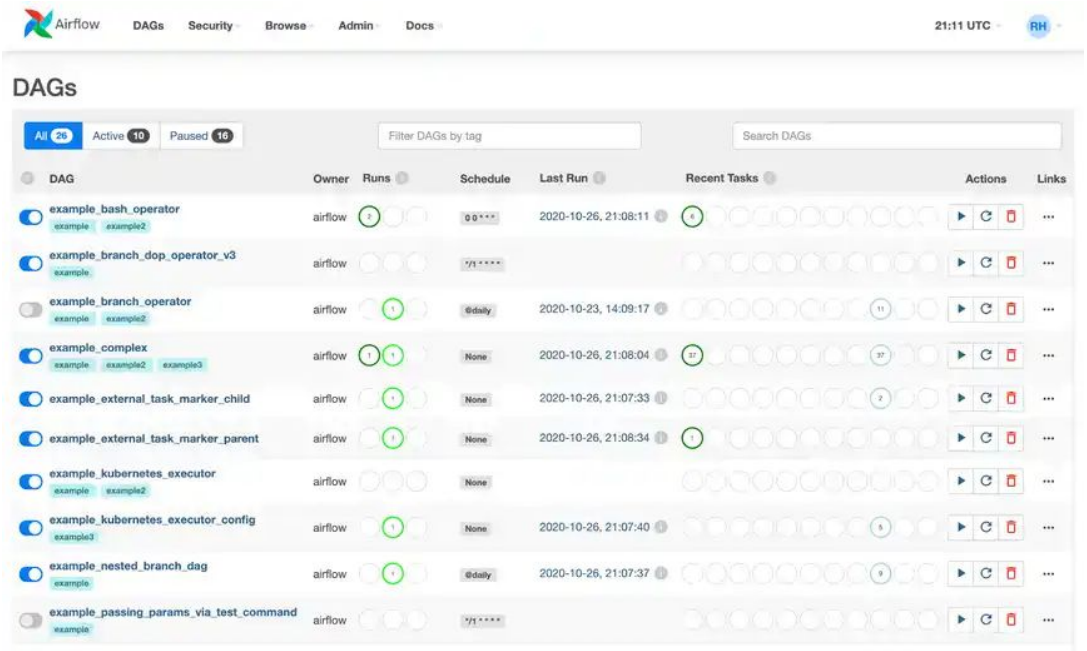
Importance of MLOps for you



Some tools for MLOps

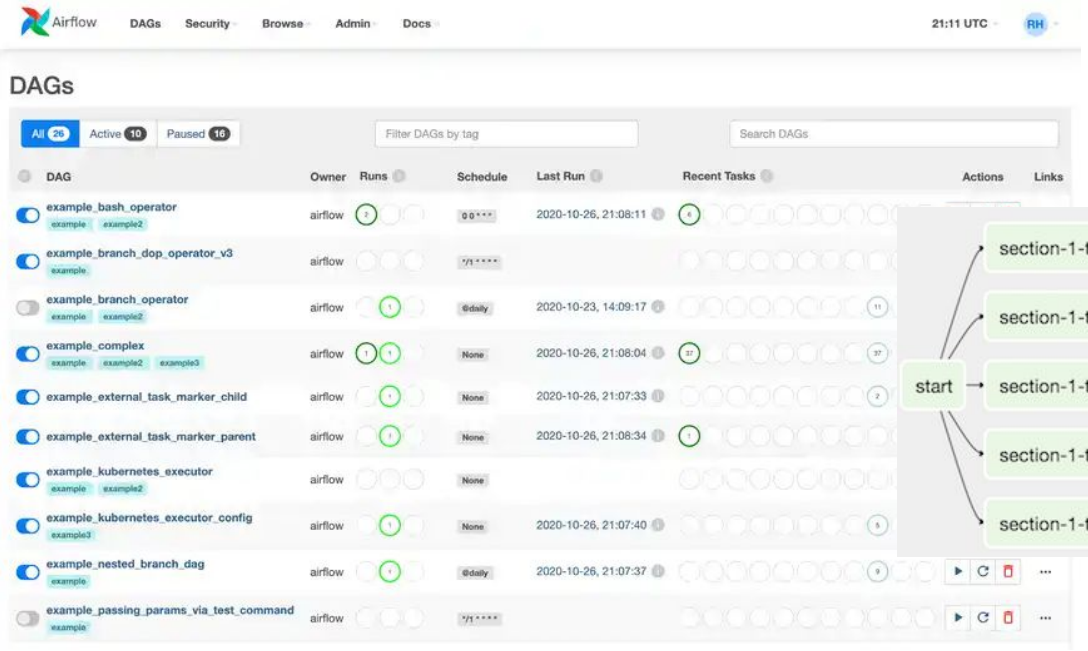


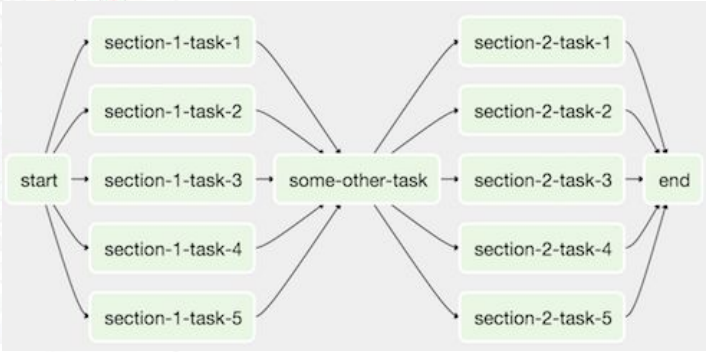
Example: Pipeline management



https://godatadriven.com/blog/apache-airflow-tutorial-for-data-pipelines/

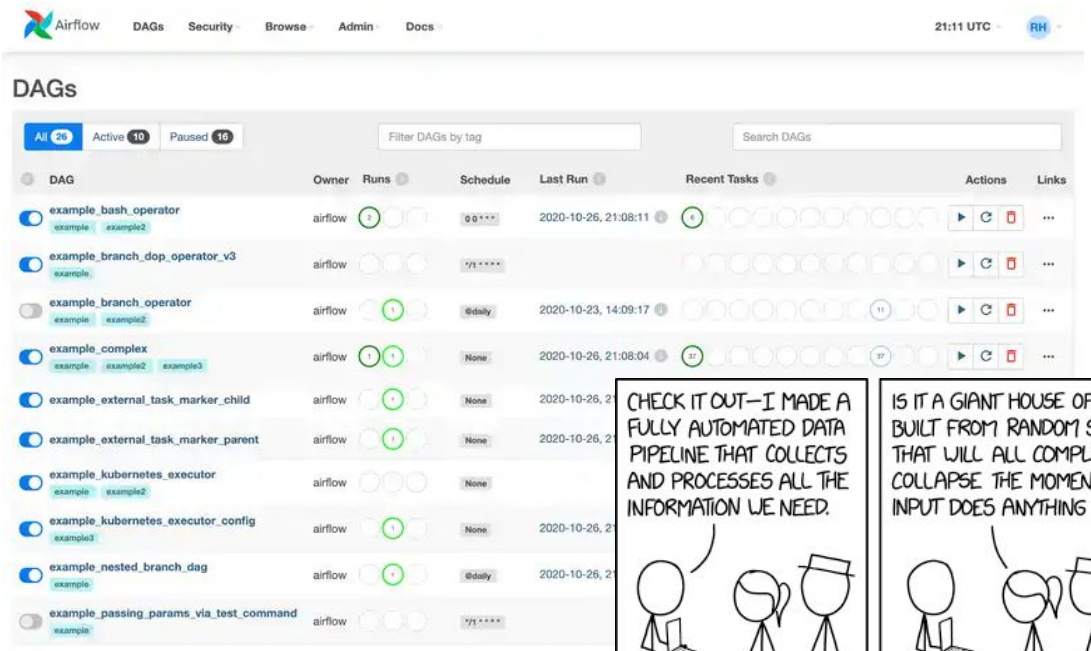
Example: Pipeline management

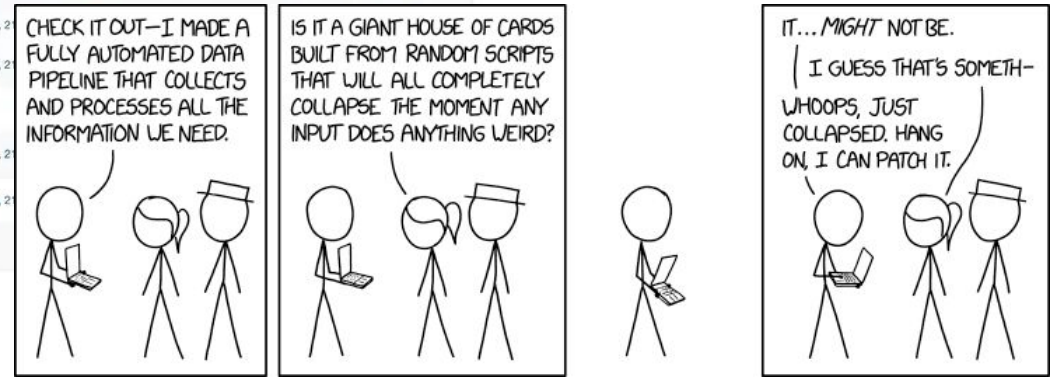




https://godatadriven.com/blog/apache-airflow-tutorial-for-data-pipelines/

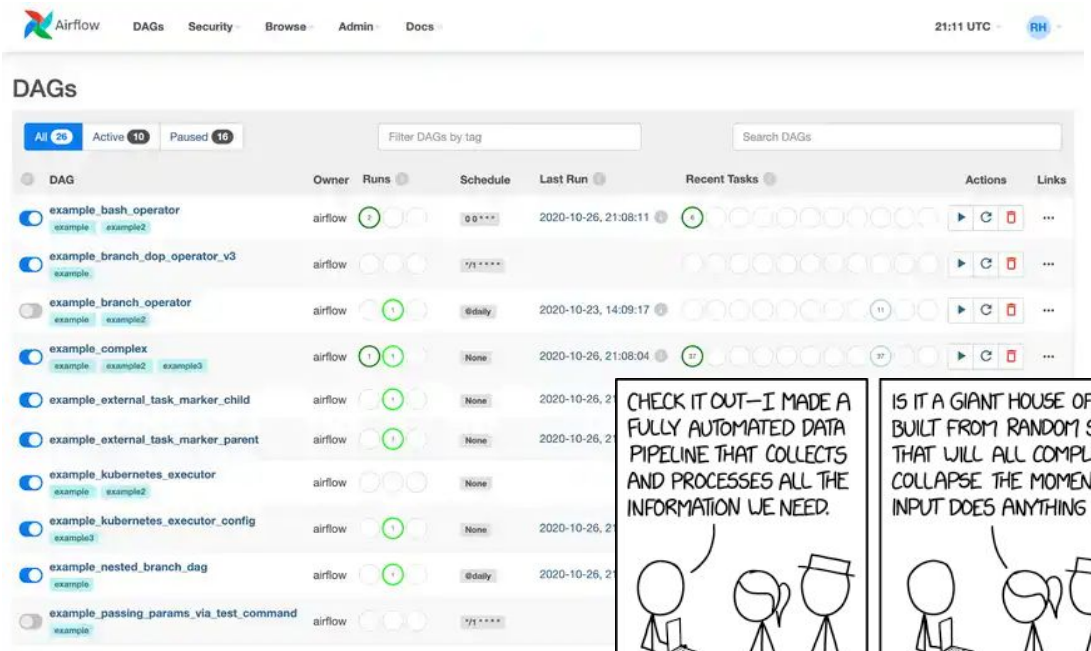
Example: Pipeline management





https://xkcd.com/2054/

Example: Pipeline management



https://xkcd.com/2054/

Current tool status

- There’s a lot of tools, and it’s a fast changing world
- The maturity level of most tools is still quite low
- There’s a big tendency for some of these tools to try to solve everything

Maturity Framework

0: No MLOps	The teams are disparate and releases are painful	Manual builds, deployments, testing, training
1: DevOps but no MLOps	Releases are less painful Difficult to trace/reproduce results	Automated builds and tests for application
2: Automated Training	Training environment is fully managed/traceable Easy to reproduce model	Automated model training Centralized tracking of model performance
3: Automated Model Deployment	Full traceability from deployment back to data Environment managed: train > test > production	Automated tests for all code Data provenance
4: Fully Automated Operations	Full system automated and easily monitored Production systems providing information and automatically improving models	Automated model training and testing Verbose, centralized metrics

Above all, it's about collaboration

While many of the DevOps patterns shown in this book require automation, **DevOps also requires cultural norms** and an architecture that allows for the shared goals to be achieved throughout the IT value stream. This **goes far beyond just automation**.

The DevOps Handbook

For MLOps to learn from DevOps, **we must center the needs of data scientists** and the people that are impacted by their models first. [...] When looking for a solution to automate, consider if you're only reducing the work required for manual processes or if you're also **enabling data scientists to focus on the hard problems they're trained to tackle.** [1]

What does success look like?

- 🚀 Time to value of new models and projects
- 🚀 Costs of training infrastructure
- 🚀 Efficiency of data scientists/machine learning engineers

Questions?