MLOps Done Right*

* according to us.

FUZZY LABS

Table of Contents

What is MLOps

MLOps is a modern approach to machine learning that reduces the pain of productionising models, giving you the most value out of machine learning as quickly as possible while reducing risk.

Why is MLOps a big deal right now?

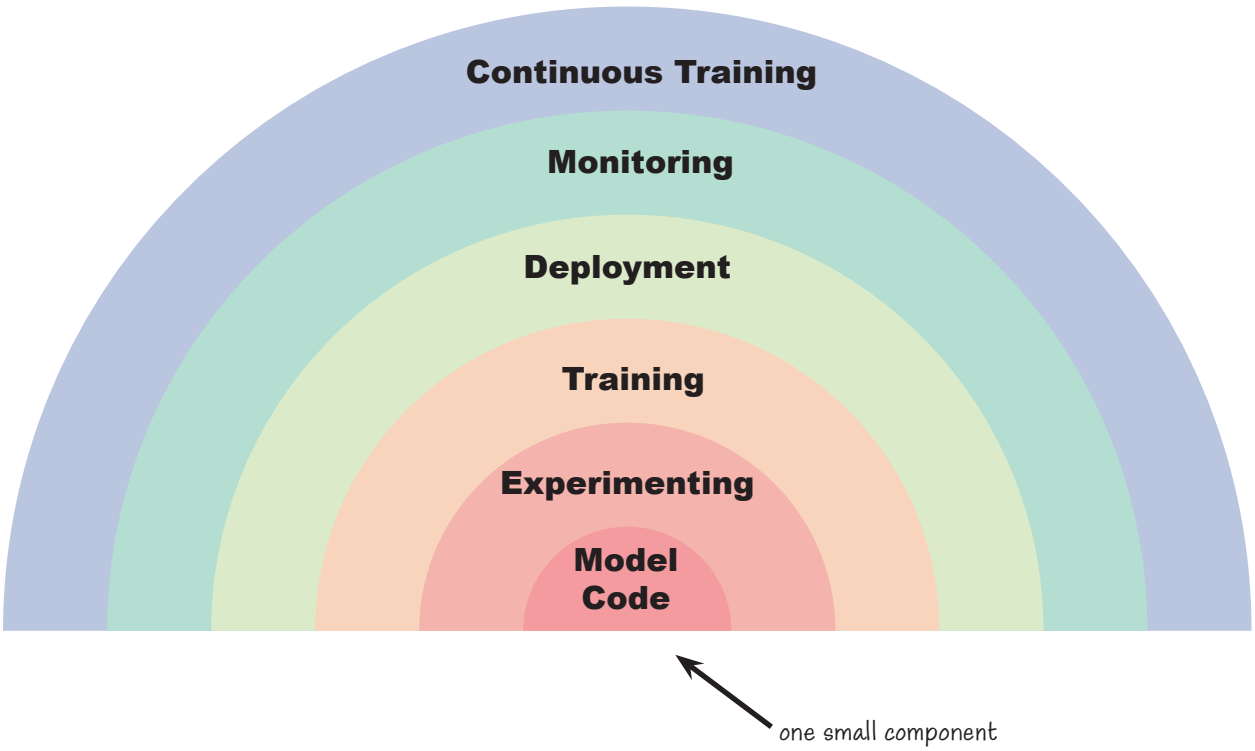
Increasingly more businesses are trying to build ML capabilities.

But they are encountering certain common difficulties along the way of taking machine learning models into production.

At the recent launch of Vertex AI Google said that “Machine learning in the enterprise is in crisis”. Companies are investing in ML but not getting value out of it.

ML models have and need an efficient lifecycle. Building your first model is just the start of the journey.

Data scientists build models, but the model code is only one small component of the overall machine learning capability.



FUZZY LABS

Life Without MLOps

Failure to Collaborate

- With no centralised way to track code, data and experiments, it's difficult for even a small team to work together effectively.
- Lack of experiment tracking means that team members are unable to reproduce one-another's results.
- A lack of collaborative tools and processes means that people can't collaborate well even when they want to.

Loss of knowledge

- Without a single source of truth for code or data, there's no way to track the history of those assets.
- There's lots of experimental code, yet nothing has been standardised.
- Nothing is repeatable, so we can't have confidence that the team can continue to operate effectively in the future.

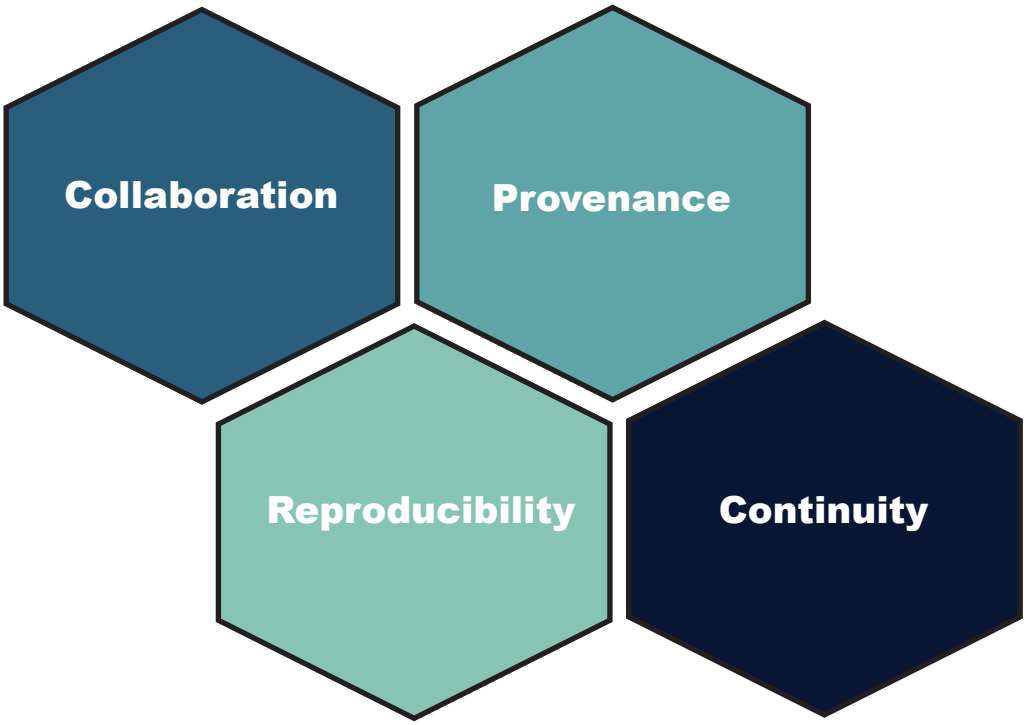
Failures in production

- Lack of automated provisioning means that we can't be confident in what's been deployed.
- When failures do happen, we don't always know about them due to a lack of monitoring.
- The technical team are spending too much time on repetitive tasks. Repetition invites errors to creep in.

Regulatory and ethical risk

- Without monitoring, we don't know whether models give reasonable answers all the time, and we can't be sure of whether biases exist.
- In some industries, regulations might require us to audit models in production and be able to explain the answers which the model gives.

MLOps Principles



Collaboration

People with different specialities are empowered to work together effectively.

Provenance

For any model, we're able to track the code version, data version and parameters that went into making that model.

Reproducibility

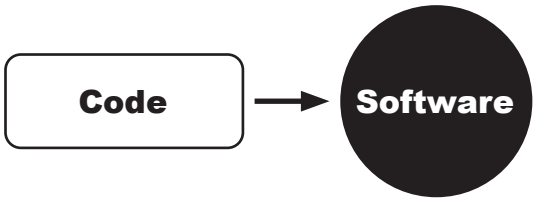
Both experimental and production models can be reproduced easily on-demand.

Continuity

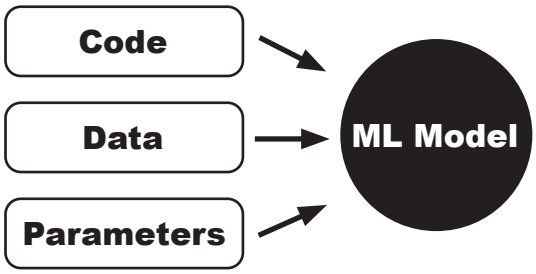
We can build, test, and deploy models and associated infrastructure automatically and consistently.

Machine Learning Vs Traditional Software

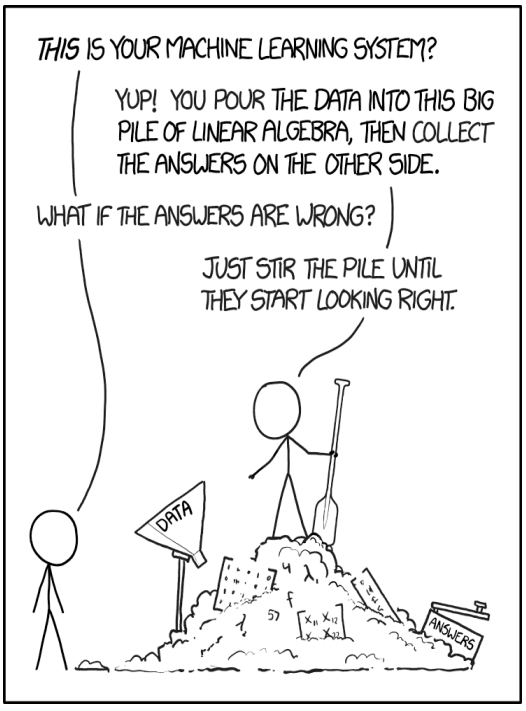
In standard software engineering the software components that we deploy into production are based purely on some code that lives in version control.



Whereas, with machine learning we have code + data + model parameters.



While software engineering is all about telling the computer how to do something, ML is about asking the computer to learn on its own how to do something. As a result, ML involves much more experimentation; it's not an exact science.



In software engineering, the code changes that you make are typically aimed at fixing a bug or adding a feature. The same applies for ML, but additionally you're taking into account changes to the data that you started with, potentially re-training that model periodically for updated data. The model can even have an effect on the data that it gets trained on in the future (through affecting customer behaviour, for example).

What about DevOps?

You may have heard the term devops.

Devops is a culture and a practice adopted by the software engineering community.

10 years ago you would hear a lot of developers say “but it works on my machine”. The trouble was, what worked on a developer’s machine often didn’t work when you tried to productionise it, and getting something to production involved a lot of guesswork.

As a result, software engineers adopted DevOps, and this transformed the way software was and is productionised.

Today the same problem exists for ML. Data Scientists have models that work on their machine … but so what?

The data science / ML world hasn’t had their DevOps moment yet, but it’s emerging, its name is MLOps.

The challenge for the data scientist coming to MLOps is understanding the tools, starting from no prior knowledge.

The challenge for the devops specialist coming to MLOps is understanding which of the tools and techniques that they already know are transferable.

It's Not A Phase ...

As we said at the beginning, MLOps is an approach to machine learning that reduces the pain of productionising models. The model is one part of the whole. For us, the keys to success in machine learning are: Collaboration, Provenance, Reproducibility and Continuity.

The diagram below highlights these values, alongside 5 phases that ML projects go through, from experimentation through to full continuous training. We'll describe each phase in detail.

FUZZY LABS

Experimenting

What is it

At the start of a new ML project, you don't usually know what the best way to solve your problems will be. As a result, the data scientist will try out a number of different ideas before they have something that they're confident will work in real life.

Why do you do it

As we mentioned before, machine learning differs from software engineering because in machine learning we want the computer to learn how to do something by itself. There's no obvious right way to do this at the start, so that's why we need to experiment.

Who typically does it

Data scientists will be doing the experiments, typically using a notebook environment, which is something that a data scientist will be very familiar with.

What does it typically involve

For each experiment, there will be a set of data, some code, and some parameters.

The data typically contains examples related to a problem that you're trying to solve with ML, the code determines how the computer will approach learning for this problem, and parameters represent the specifics of each experiment.

What are the key MLOps concepts

One of the first MLOps concepts we introduce is an experiment tracker. This allows every data scientist on the team to log their experiments, along with the version of the data, version of the code, and particular parameters, to a central location, and means that any historical experiment can be found later on.

Training

What is it

Training is how we build an AI model; this model represents a capability that the computer has learned. Through the training process the computer learns from patterns in the data, following the code and the training parameters.

Why do you do it

Training actually begins in the experimentation phase, but in that phase it will be typically localised to a data scientist's machine.

In order to productionise training we have to create a training pipeline. The pipeline contains all of the steps necessary to go from the original data, along with model parameters and code, and create a model that's ready to be deployed.

Who typically does it

Because this phase is about starting to productionise the experimental work, data scientists will have a hand in creating the training pipeline, but would typically work with ML engineers who provide operational support.

With effective MLOps tooling, data scientists should be able to do this by themselves, so that ML engineers can be hands-off, just providing the tools, letting the data scientists self-serve.

What does it typically involve

All of the steps required to train the model - including data preparation tasks, as well the training itself - will be combined together into a script or template that can be run on demand or triggered automatically as part of continuous training.

What are the key MLOps concepts

In addition to a training pipeline, another key concept to understand is the model registry. This is a central location into which every model that is trained via the pipeline can be stored for later retrieval.

Models would be retrieved from the registry when you want to deploy them. Data scientists and other technical team members might also want to retrieve a particular version of a model, so that they can test it locally, perhaps for debugging purposes.

Models in the model registry can be traced back to the experiment tracker, so that we have full provenance for every model.

Deployment

What is it

Deployment is when we take a model that's been trained in the previous phase and make it live, ensure that it meets the business requirements for availability and performance, and test different versions of that model with real users.

Why do you do it

A model needs to be deployed before it can provide useful insights to the business. Most organisations won't accept a model that is anything other than 100% available, and with modern cloud infrastructure this is considered the standard, however there is a trade-off of cost against performance.

As part of deployment, we want the ability to run multiple versions of a model side-by-side. This enables us to try out different variations on the same model against real-world data so we can understand which works best in reality.

Who typically does it

As with the training phase, typically it may be an ML engineer who is responsible for deployment. But with effective MLOps tooling, data scientists should be empowered to deploy on their own with ease, although there may still be a product owner who gives final authorisation to each deployment, because the new model may include new features or behaviours that need to be considered carefully before their release.

What does it typically involve

First we need to determine what kind of infrastructure we want to use for serving the model; we might use dedicated servers, or we might use a specialised AI platform. Next, we install the trained model into this infrastructure, and finally configure the infrastructure according to business requirements, which includes looking at security, reliability and performance.

What are the key MLOps concepts

The portability of models is key to modern MLOps practices. This is the idea that we can package a model up in a generic way. Think of it like shipping containers: every port in the world can process them, every ship can carry them, no matter what they contain.

In the same way, if our models are portable, they will run anywhere we want. This gives us huge efficiency gains: the deployment infrastructure doesn't need to know about the different types of models that it might run, it just needs to know how to work with these generic, containerised models.

Having these portable environments also makes it very easy to configure scalability and run multiple versions of a model side-by-side.

Monitoring

What is it

After we have deployed a model, we need to monitor it. Monitoring is the ongoing observation of models while they run in production.

Why do you do it

First and foremost we monitor the models so as to ensure that they are working properly as well as performing to expectations. Furthermore, we need to validate the results returned by the model so that we are confident that it is returning results that make sense.

The real-world data that goes into a model can subtly shift over time, and subsequently so too can the model results, which means these ongoing observations are critical to maintaining business value from machine learning

We can take actions based on monitoring, like sending somebody an alert, or triggering a new training pipeline to retrain a model.

Who typically does it

Monitoring itself is something that should be happening automatically, once the monitoring process and infrastructure has been put in place, typically by ML engineers.

What does it typically involve

The configuration of appropriate monitoring tools, ensuring that all of the relevant information from the models themselves is made available to these monitoring tools, and then deciding what actions should be taken based on the monitoring.

What are the key MLOps concepts

We want to monitor three aspects of a model: inputs, outputs, and errors.

For inputs, we're interested in checking how similar the inputs are to the original training data; if the inputs drift far from the training data, that tells us our training data is no longer representative.

For outputs, we're looking for either a sudden change, which suggests the model has unexpected and unexplainable behaviour. We're also looking for unexpected biases in the output, which suggests the training should be re-visited.

Error monitoring looks at whether the model emits errors due to such things as unexpected input values, internal software errors.

Finally, we want to periodically make contact with the model to make sure that it is still running, and we want to monitor its performance too.

Continuous Training

What is it

Models are never finished. We always need to re-train our models, either based on manual decisions or automation. Continuous improvement is about enabling a team to rapidly re-train and deploy new versions of models.

Why do you do it

The decision to train a new model comes from either new business requirements, or from changes from the expected model input or output as detected by monitoring systems.

Who typically does it

Everybody involved in a project has an interest and involvement in continuous improvement.

What does it typically involve

In this phase we tie everything together: anybody on the team should in principle be able to build, train, test and deploy a model to production at the press of a button.

What are the key MLOps concepts

There are no new concepts introduced at this stage. We're taking all of the pieces established before and combining them to either deliver new features or keep our model relevant.