The ML Workflow in practice.

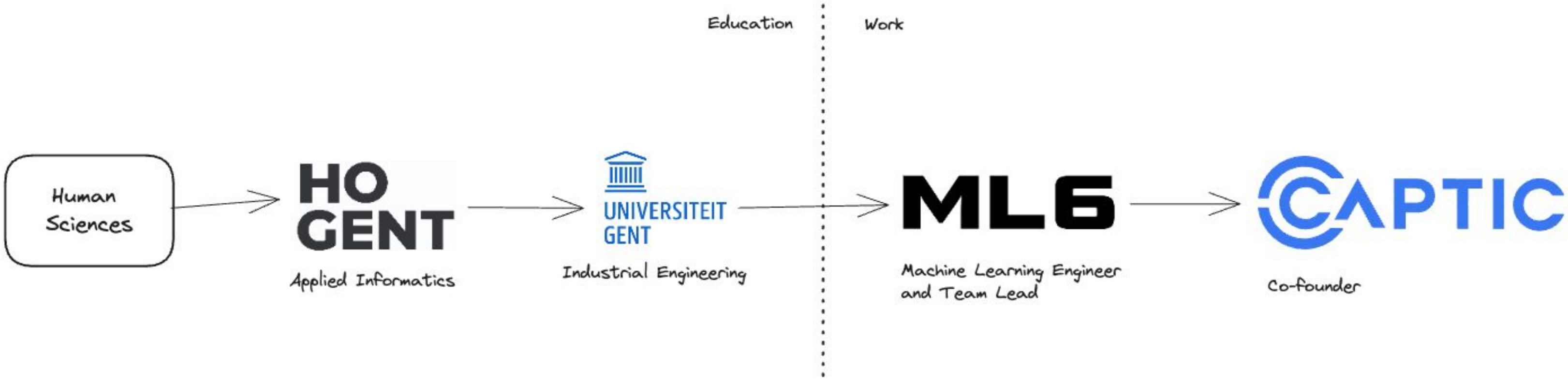
Our learnings at Captic distilled for the new generation of ML Engineers

AI driven robotics and inspection against labor scarcity.

Part of SKYHAUS

Who’s this guy?

Tim De Smet
Co-founder & CTO at Captic



AI driven robotics and inspection against labor scarcity.

Part of SKYHAUS

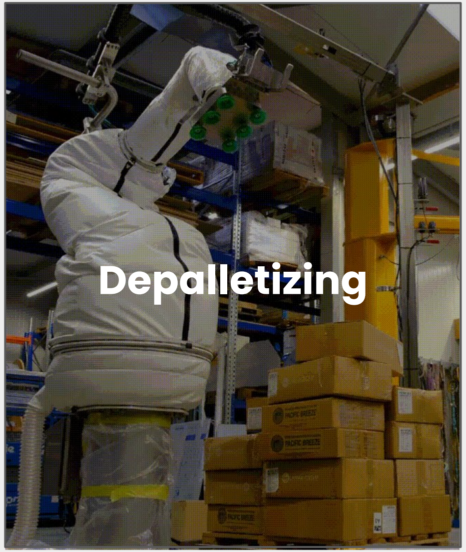
Leverage the incredible power of Captic's AI technology.

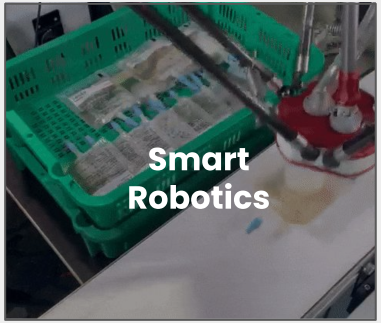
Produce more with fewer people

Make fewer mistakes

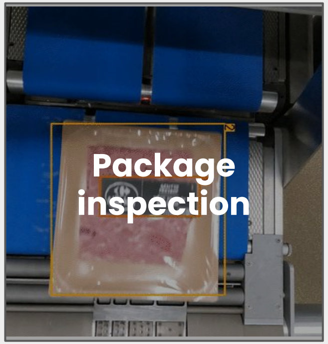
Leverage the incredible power of Captic’s AI technology.

Produce more with fewer people





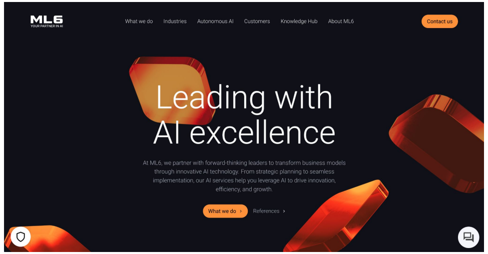
Make fewer mistakes



Did you know...

Captic's mother company is ML6

- Leading AI firm
- 150 employees
- HQ in Ghent, Belgium



Our customers include:

Belgian Pork Group · What's Cooking? · P&G · Greenyard · Jules Destrooper · Wyzo · agrifirm · Milcobel · Takeda · Holcim · Wienerberger · balta · Pfizer

We're partners of:

Microsoft · NVIDIA · FANUC · SCHMALZ · ADVANTECH

Building on a decade of R&D, Captic now delivers the incredible power of AI to industry leaders.

ML6

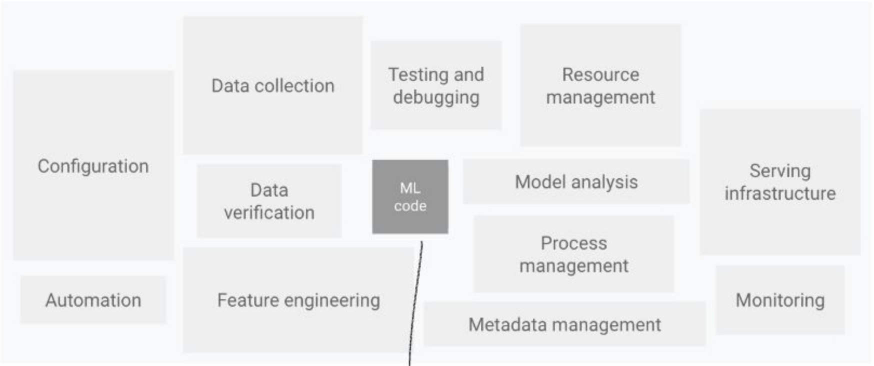
And more...

And more...

Why MLOps?

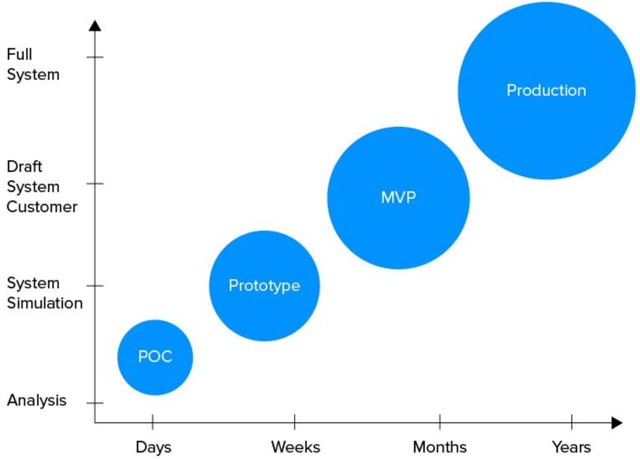
Why MLOps is a crucial topic for any ML Engineer.

ML by itself isn't enough.



Becoming easier everyday

But this is just a Proof of Concept (PoC) by itself

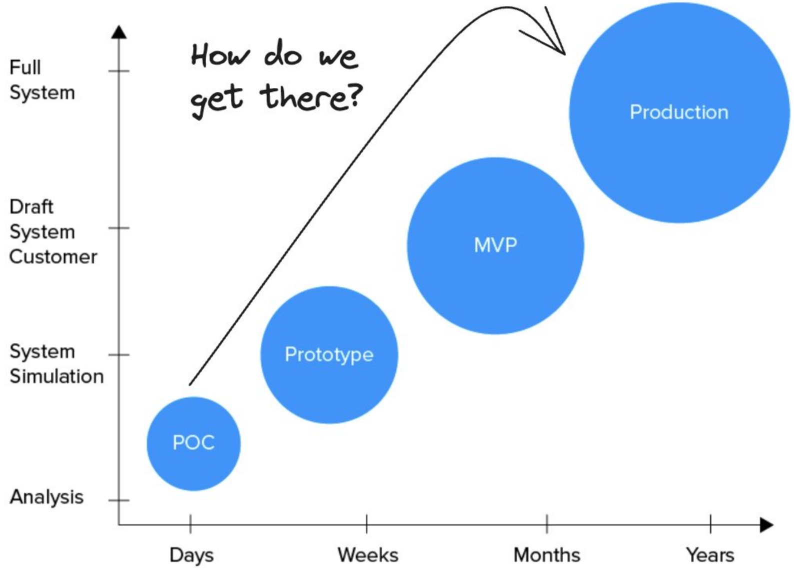


Why you're taught MLOps.

MLOps = Standardization and streamlining of ML lifecycle management

→ allows you to **get to** - **and stay in** - **production**.

ML becomes valuable in production, not before.
That's when it's no longer a gimmick.



Production = continuous effort.

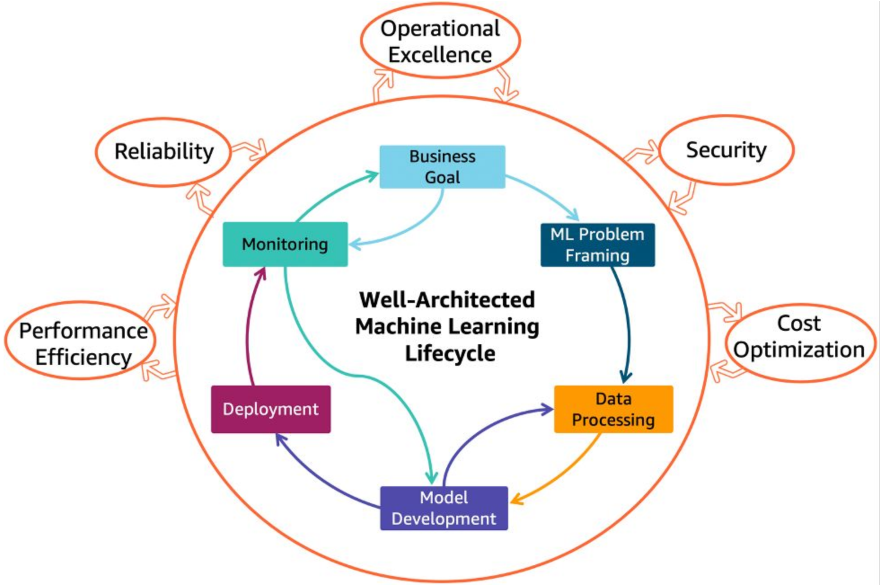
Getting to / staying in production is a continuous effort.
It requires iterations through the ML Lifecycle.

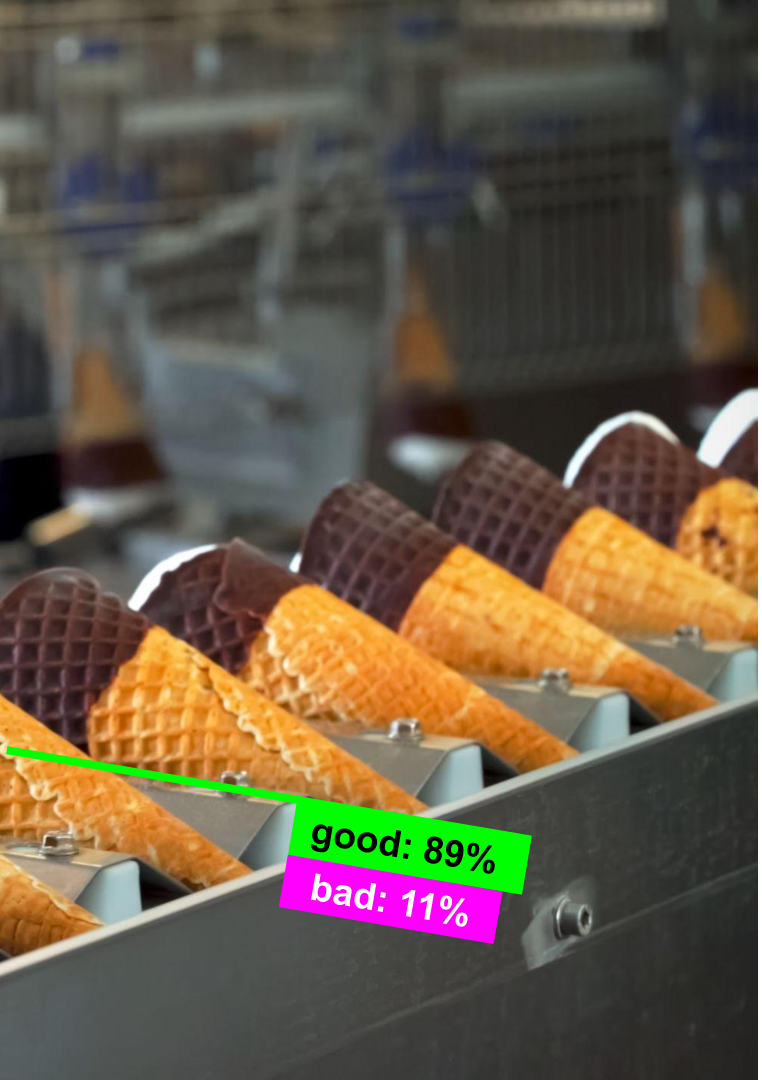
MLOps will allow you to:

- work professionally
- Streamline processes for efficiency

During this lecture, we will:

- Walk through the ML Lifecycle
- Illustrate with a real-world use case
- Share MLOps tips /tricks.

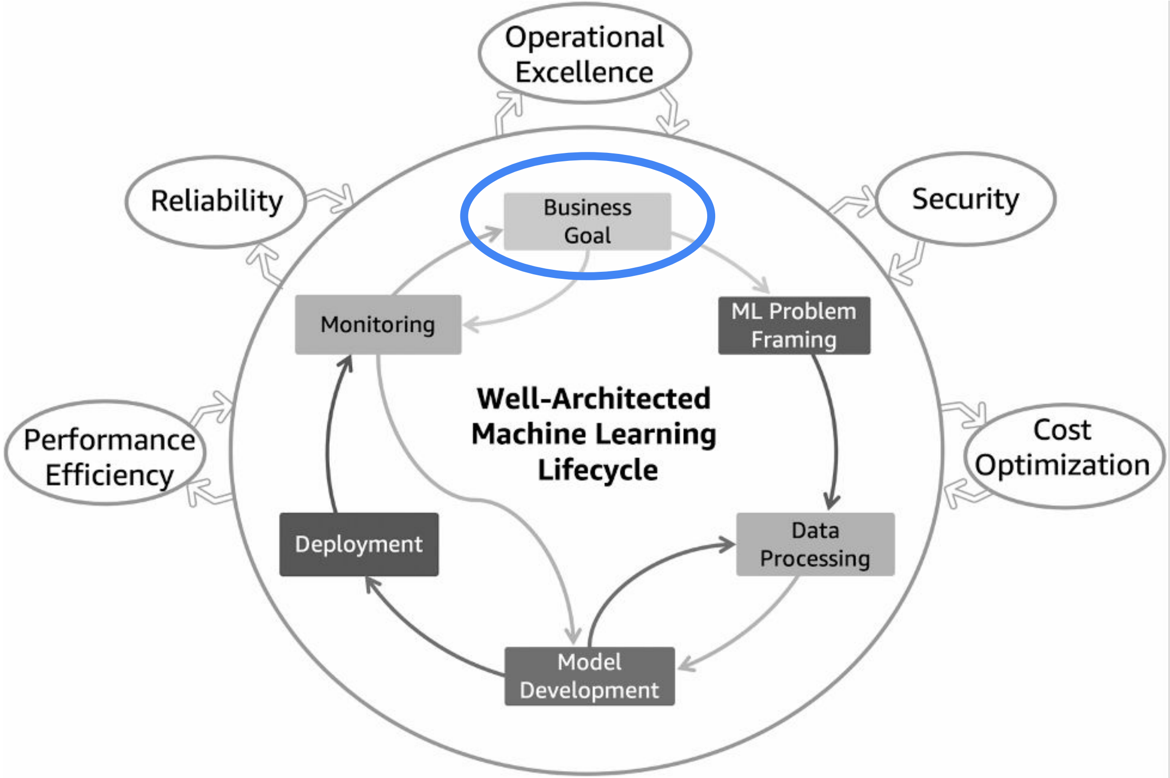




ML Lifecycle IRL.

Let's solve an example **inspection use case** together.

The ML Lifecycle.



Business Goal.

Questions you need to ask yourself:

1. What real problem are we trying to solve? (What is the goal?)
   - Staffing issues?
   - Safety issues?
   - Quality issues?
   - Throughput issues?
   - Waste issues?
   - ...

2. Is ML the best way to solve the problem?
   - No?
   - Yes?

A common pitfall.

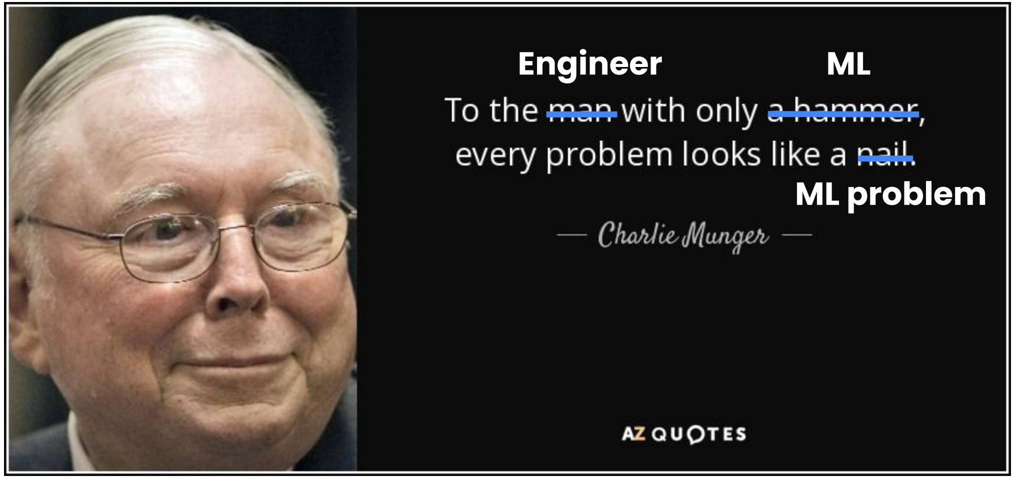
Business is always looking for the:

- Biggest ROI
- Most likely ROI

ML solutions require a lot of:

- Knowledge
- Skill
- Ongoing effort

→ Driving up the cost and risk of failure



Tip #1: If the problem can be solved without ML, then don't use it just because it seems cool
Tip #2: Don't engineer for the sake of engineering

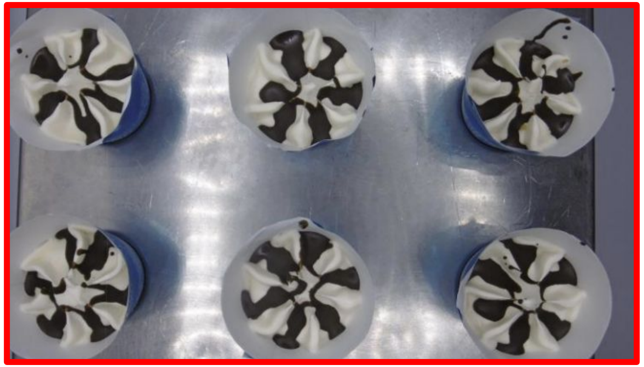
Business Goal IRL. (1/2)

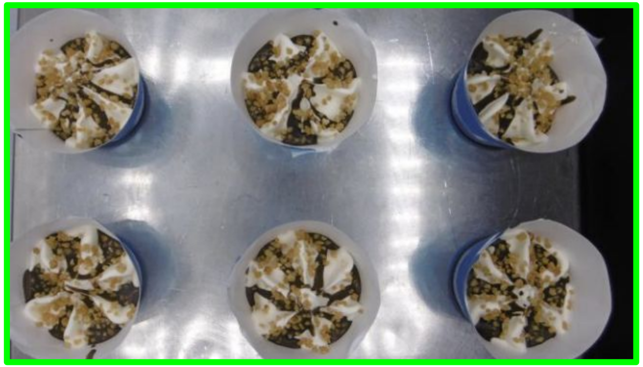
Company that manufactures ice creams

Problem:
- Customer complaints due to missing toppings
- Labor cost keeps increasing

Result:
- Dissatisfied customers
- Shrinking margins due to claims

The goal is **quality assurance**.





Business Goal IRL. (2/2)

Do we need to use ML?

Some kind of sensor is needed.

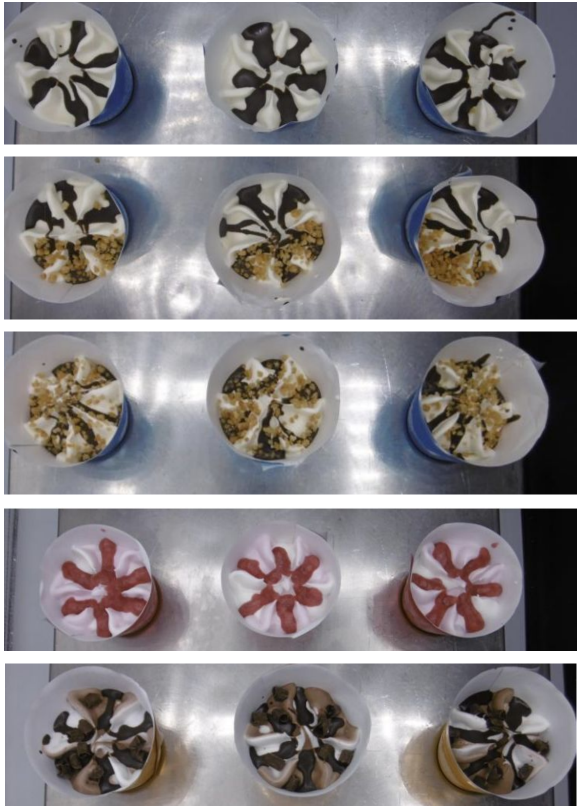
→ Weigher? Too similar
→ Camera? Makes sense

Can we use traditional methods?

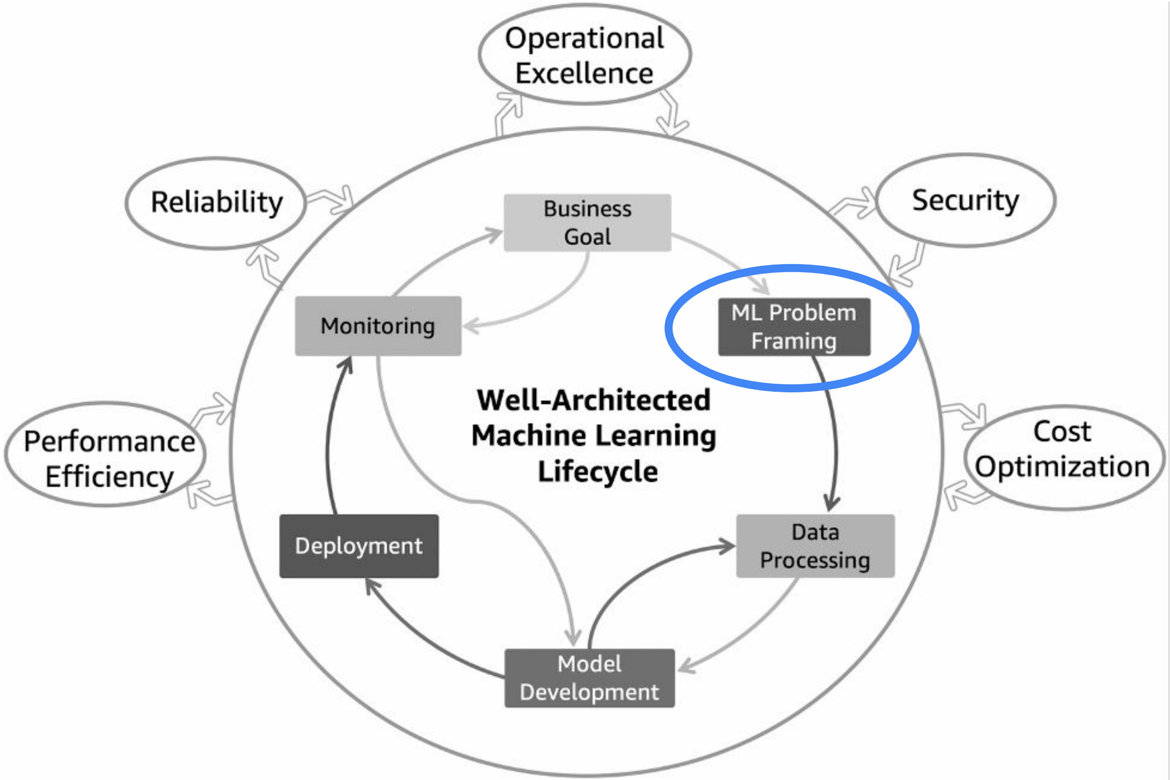
- Color? Could work but won’t handle variety well
- Shape? Too similar

We’ve exhausted all other options
→ We’ll use ML to solve this problem

= Automation through AI-Vision



The ML Lifecycle.



ML Problem Framing.

Our case: Automation through AI-Vision

Questions to ask yourself:

1. What do we want to predict?
2. Do we have performance expectations?

Answers:

1. For each recipe, we need to be able to determine if the toppings are present

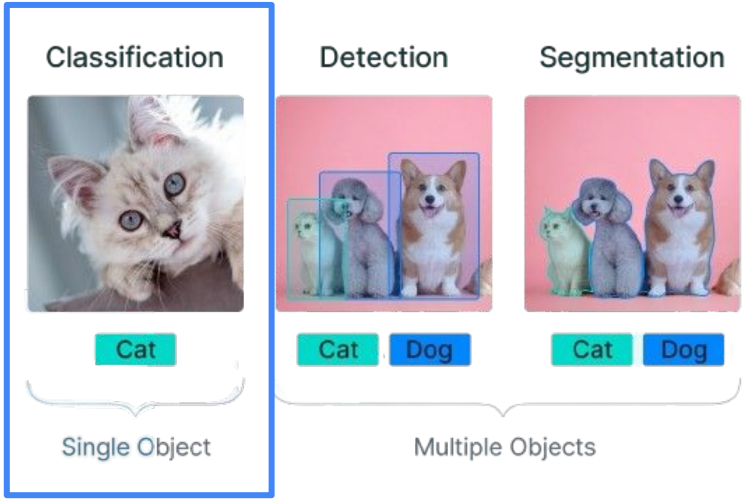
2. Requirements
   a. We will need the system to work real-time → Deployed on the edge
   b. False positives as are as bad as false negatives

ML Problem Framing IRL.

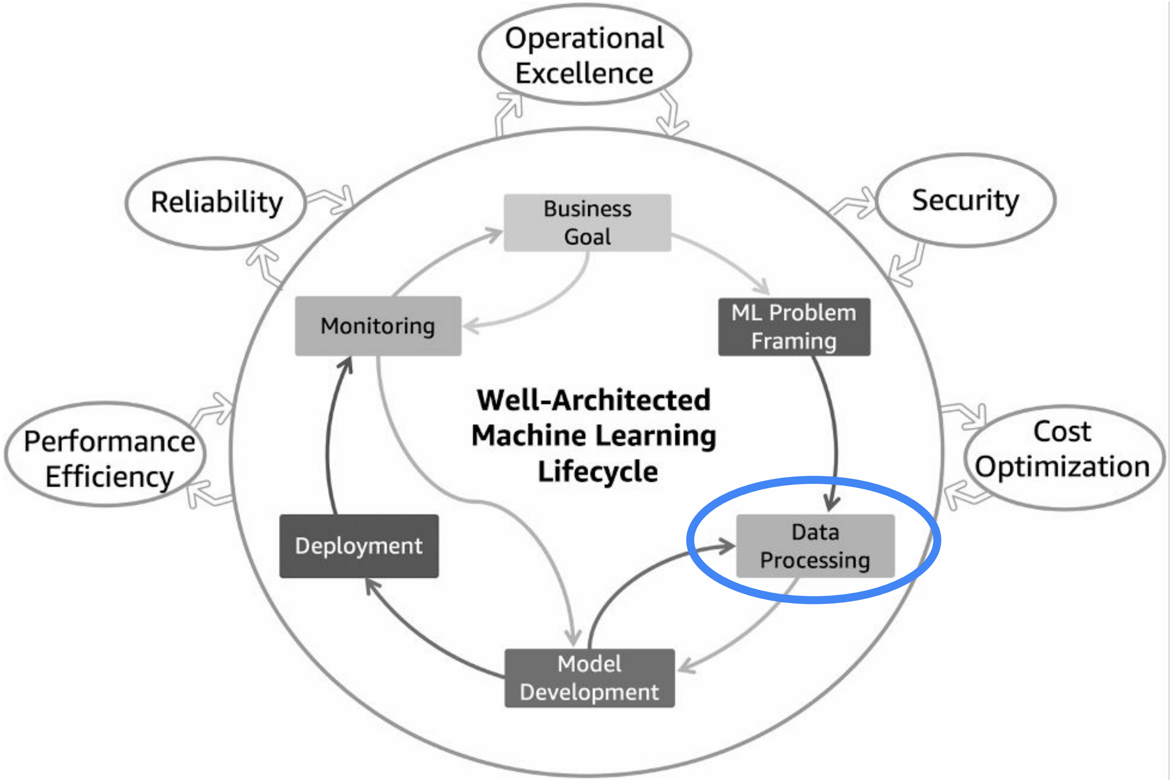
We know we will have to leverage ML to cope with the variety in production.

But what kind of ML?

We need to be able to determine good / bad. → Classification



The ML Lifecycle.

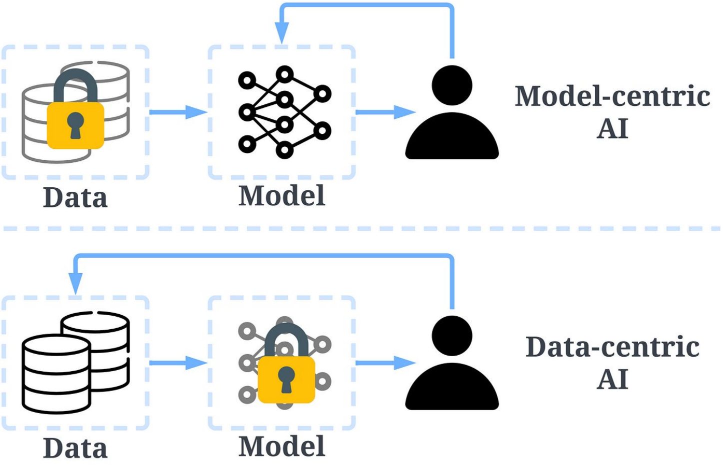


Data processing.

Questions to ask yourself:

- What data do we need?
- How much data do we need?
- Do we need particular examples that are hard to come by?
- How will we collect the data?
- Where do we store the data?
- How will we label it?
- In what format do we store it?
- Do we need to modify the data?
- How long will we keep the data?
- …

Data is hard work:



Data processing IRL.

What data do we need?
→ Images of good & bad cones in their real setting under different circumstances

How much data do we need?
→ Hard to know for sure, our classifier will probably only need a few hundred of each type

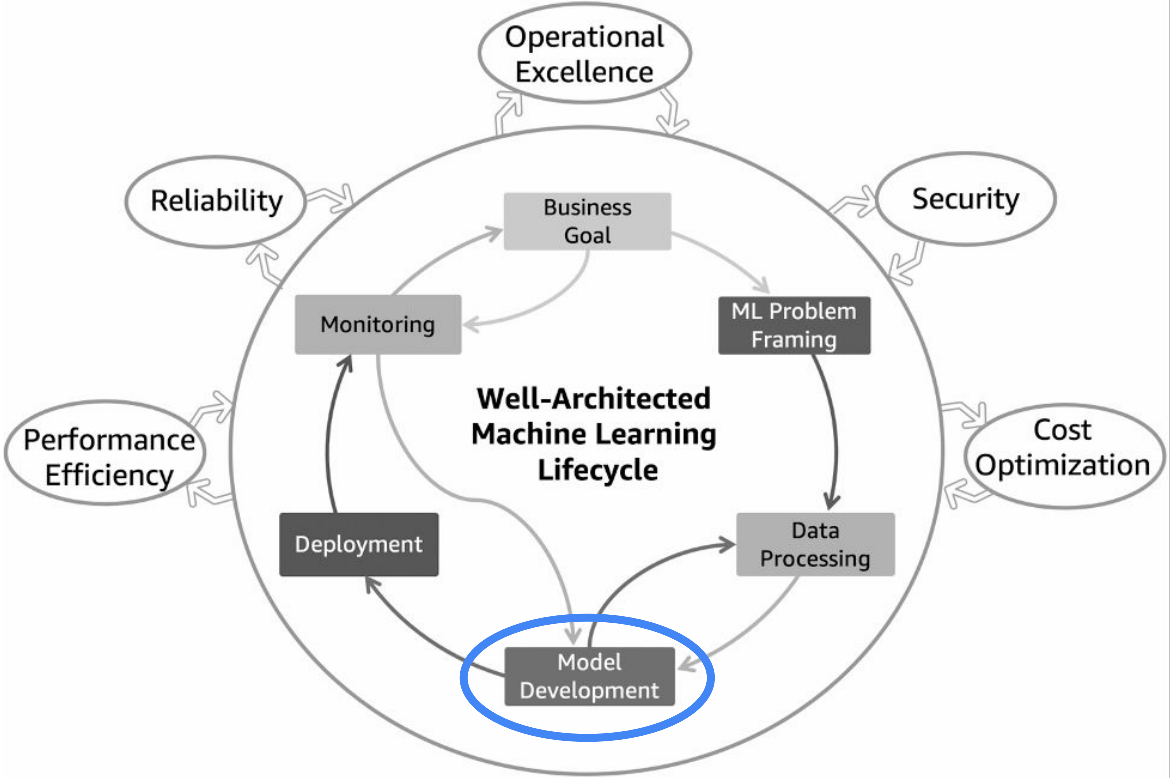
Do we need particular examples that are hard to come by?
→ We need to take into account edge cases (e.g. spillage → we can stage those if needed)

How will we collect the data?
→ Bring a camera to the manufacturing plant

Where do we store the data?
→ We'll go with an SD card for now, but the Cloud is the obvious choice (blob storage)

How will we label the data?
→ There are many tools out there. We use Azure's one, but Label Studio is a great open-source one

The ML Lifecycle.



Model development: Selection.

There are different approaches you can take in terms of model selection:

1. Build from scratch → Not done often

2. Give AutoML a go
   - Tends to be worse than SOTA-models, but perhaps good enough
   - Always good to have a baseline
   - Cloud costs

3. Use existing and proven architecture
   - Find them on Papers With Code
   - Use them
     - GitHub
     - HuggingFace
     - Keras
     - …

Model development: training.

Tip: Training on specialized hardware makes it a lot faster. Try to avoid using your own laptop.
Google Colab gives you free GPU usage.

It's crucial that training is reproducible, this requires you track everything
- What dataset
- What augmentations
- What model
- What parameters
- …

You can use a tool like MLflow for this.

Additional "Hacks":

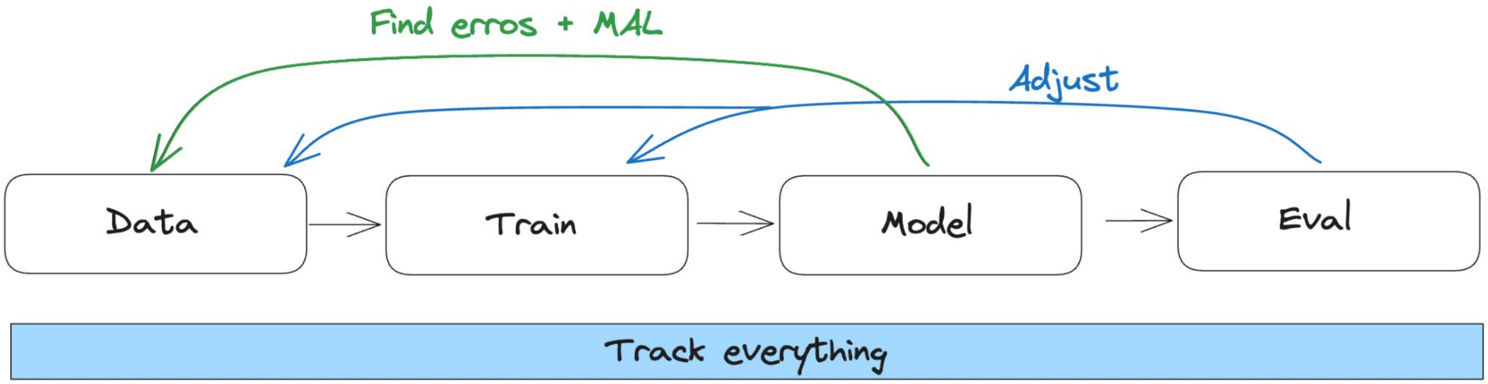
- **Transfer learning**	start from model weights for similar task → Faster (and better)
- **Hyperparameter tuning**	try different parameters for different runs to see which are best
- **Data augmentation**	generate more data by changing the data itself

Model development: evaluation.

Evaluating the model properly is key. Make sure your test dataset is high-quality and balanced.

This step should also be tracked for reproducibility.

Model development is never done → Iterative process



Model development: pipelines.

ML Pipelines = Automated sequence of steps to build, train, evaluate and deploy machine learning models. Used to streamline the end-to-end process.

Why?

- Help organize and automate
- Keeps track of everything → reproducibility
- Can scale as needed in the Cloud
- Splitting into steps allows for collaboration
- Enforces systematic approach

Many flavors:

- Kubeflow
- Prefect
- TFX
- Azure
- …

The Lab: ML Workflow IRL.

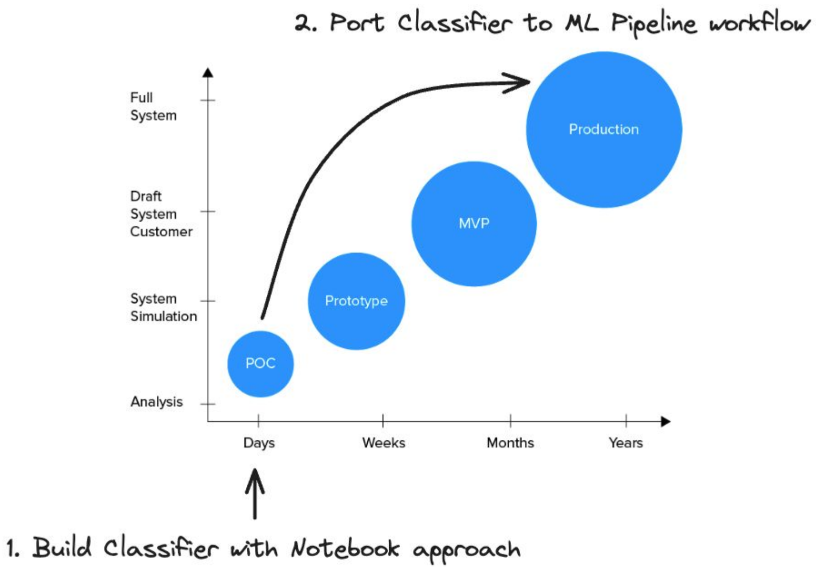
We've already:

- Determined the business goal
- Framed it as an ML Problem

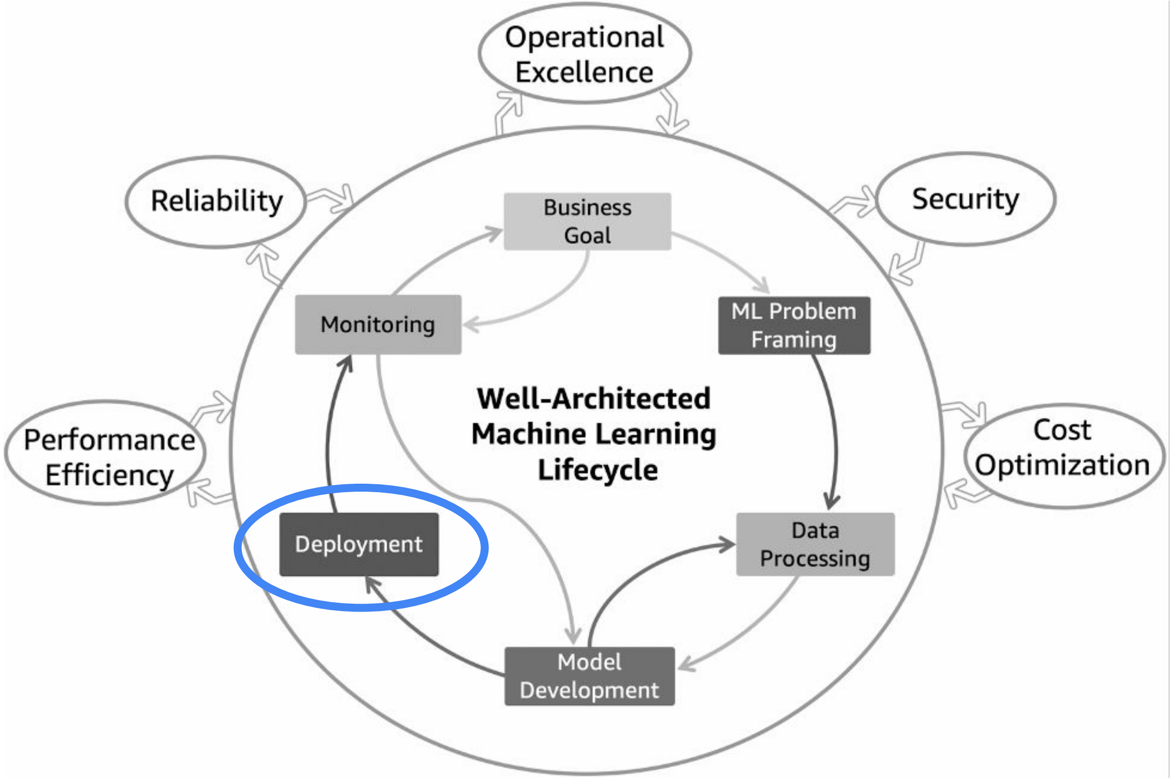
You will be given a toy dataset, so no data work
Note: not a real dataset (tiny & badly stored)

During the lab, you will:

- Build a classifier from scratch in a notebook
- Port to an ML Pipeline workflow
- Apply ML best-practices along the way (=MLOps)



The ML Lifecycle.



Deployment.

There are many ways to deploy an ML model:
- Real-time serving
- Serverless
- Batch processing
- Edge deployments

Questions to ask yourself:
- How do I want to work with my model?
- How fast should I get an answer?
- What’s my budget?
- How do we automate the release process? (CI/CD)
- What and how do we monitor?

Deployment.

We need our Classifier to work real-time. Real-time serving?
→ No, we can't let our production rely on whether we have internet connection.

We need an **Edge Deployment**

There are many options for the hardware:

- Coral
- Raspberry Pi
- Nvidia Jetson
- ...

What do they have in common?
They're tiny, so our model will have to be as well

Edge Deployments.

Since we want our model to run on Edge, we need to make it tiny and fast

→ Best to start with a model that is already pretty small

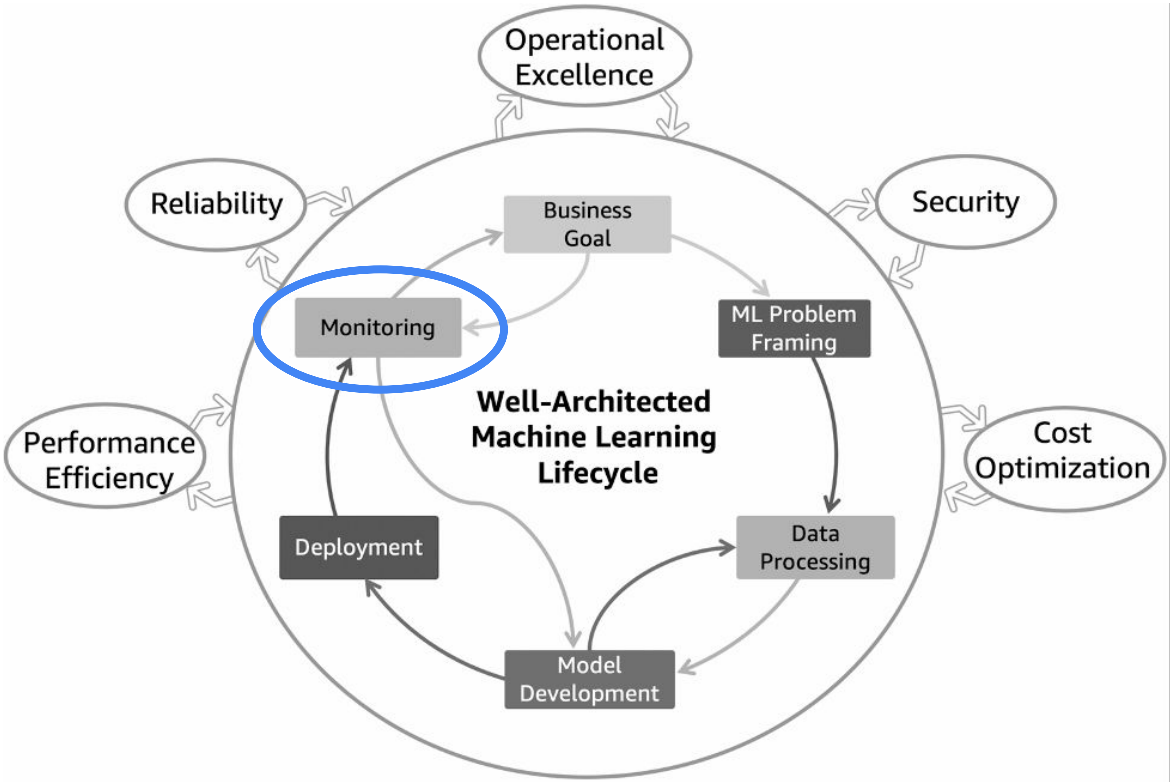
Many techniques for model compression:

- Pruning	Removing unimportant weights
- Quantization	Reducing precision of weights
- Knowledge distillation	Training a smaller student model that learns from the bigger teacher

There are tools that do this for you:

- In TensorFlow
- TensorFlow Lite
- In ONNX
- ...

The ML Lifecycle.

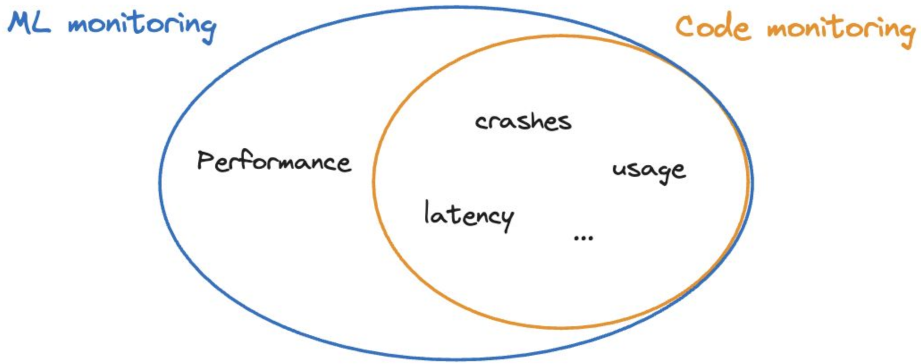


Monitoring.

Why?

- We want to know how our model is doing
- Every model will get bad over time. How fast depends on the use case

ML monitoring is different from normal code monitoring
→ Bad doesn't mean crash (latency, up, …)



Monitoring: Decay.

No model lives forever, but the speed of decay varies.

Usual culprits:
- Data drift = changing of input data
- Concept drift = relationship between input and output has changed

Example of Data drift:
Our lens is dirty so our images look different

Example of Concept drift:
The Quality manager’s interpretation of a certain class changes over time

→ Monitoring allows us to spot model decay and to retrain (back to model development)

Monitoring: Signals for ML.

What can be monitored:

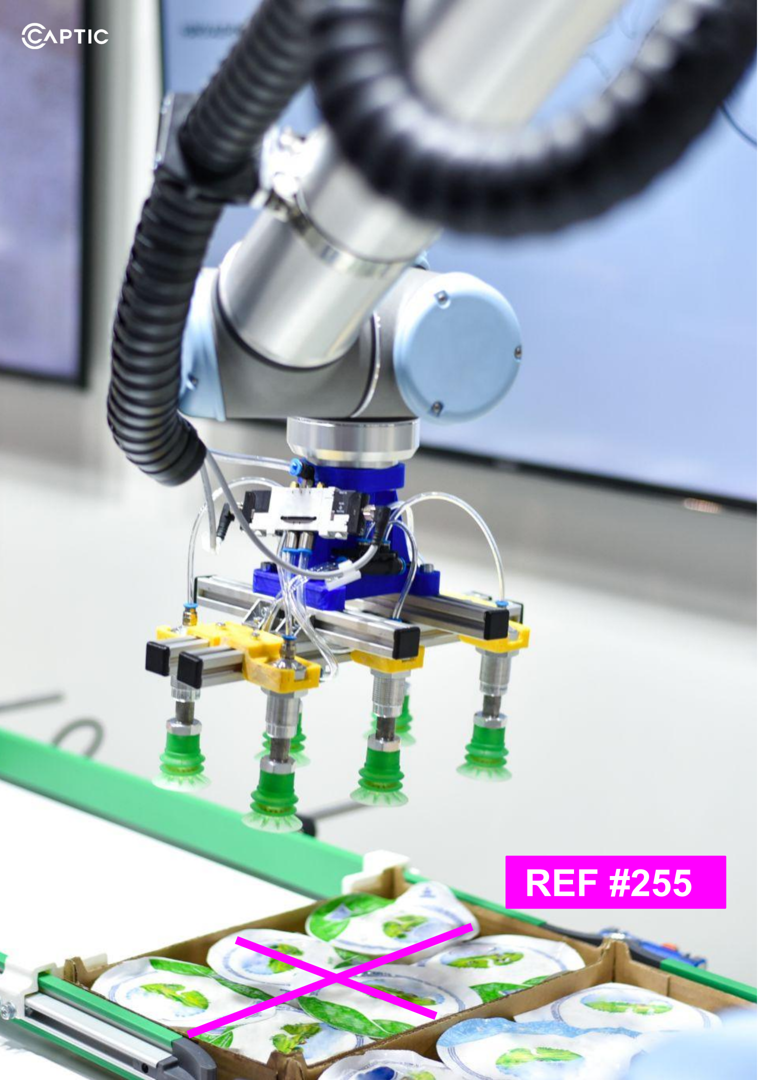
- User feedback collection
  - Instances themselves
  - How many over time

- Output
  - Confidence scores
  - Distributions as expected?
    - Dataset
    - Previous models

- Annotated data

You'll also want to monitor whether you're actually solving the business problem.

(You'll learn more about monitoring in upcoming lessons)



How to get started IRL.

Skills you'll need.

Python

Almost everything is done in python.

Git

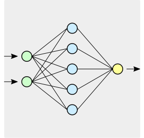
Must have skill in order to work within a team.

Docker

A key tool for any software engineer.

Cloud

The Cloud gives you huge storage, powerful compute and much more.



ML Knowledge

Understanding of the types of ML tasks, models and frameworks, allow you to problem-solve.

(The more experienced you are, the better your chances)

Apply for an internship!

For CV and/or Robotics:

CAPTIC

- ML Delivery
- Anomaly detection
- Robotic cell
- ...

For broader ML topics:

ML6

- LLMs
- Agents
- ...

How to apply? Let's connect!

tim@captic.com