

Introduction to MLOps

Sprint 1 | Week 1

INFO9023 - ML Systems Design

Thomas Vrancken (t.vrancken@uliege.be)
Matthias Pirlet (matthias.pirlet@uliege.be)

Agenda

1. Introduction to MLOps

- What is MLOps and why does it matter?
- Key concepts of MLOps
- A zoom on LLMOps
- Real world use case

2. ML project organisation

- Roles around ML projects
- Project definition framework
- Agile way of working

3. MLSD - Course structure

Introduction to the staff



Thomas Vrancken

(Instructor)

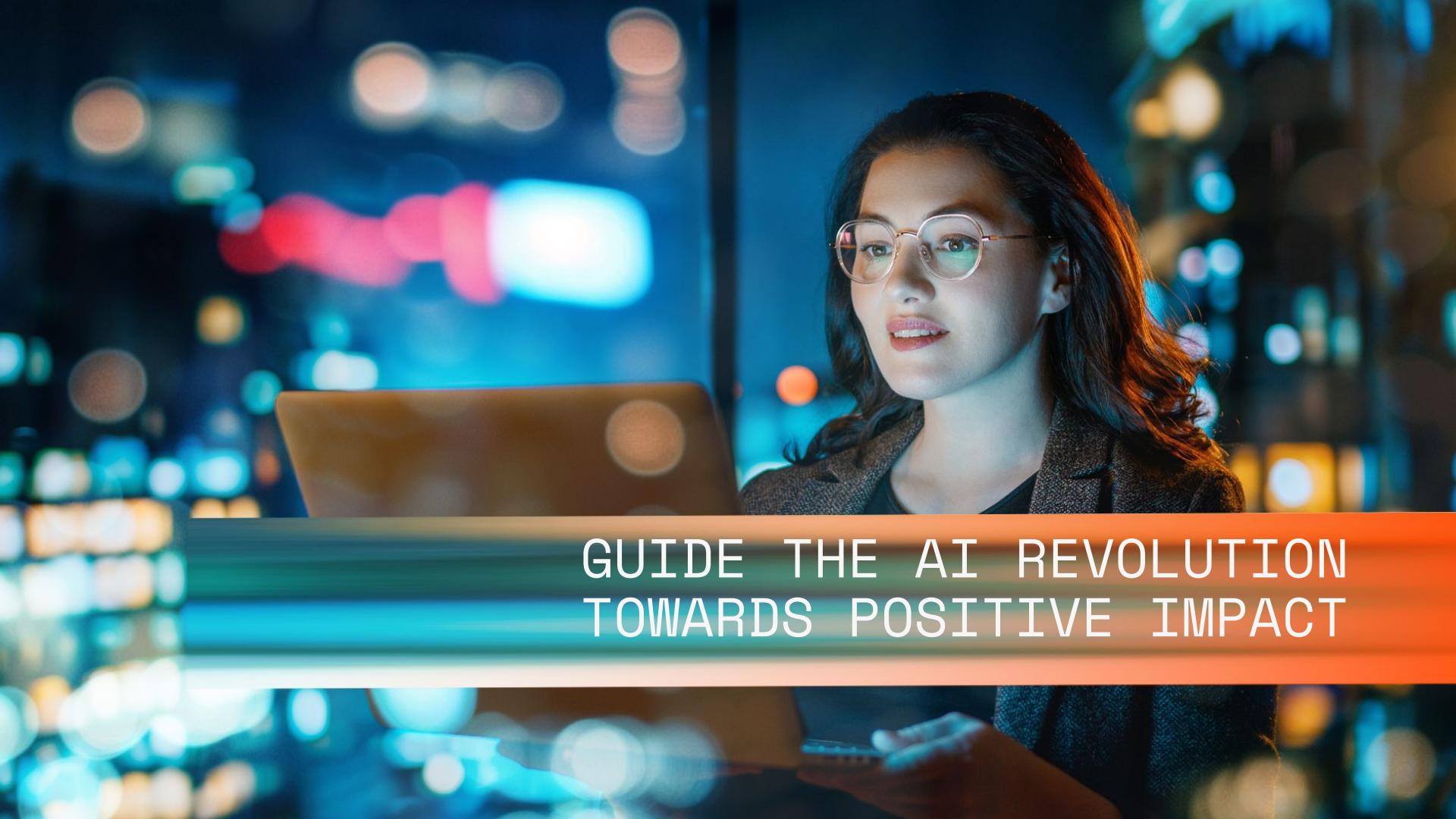
t.vrancken@uliege.be



Matthias Pirlet

(Teaching assistant)

matthias.pirlet@uliege.be



GUIDE THE AI REVOLUTION
TOWARDS POSITIVE IMPACT

INTRODUCTION TO ML6.

One of the largest and fastest growing AI engineering teams since 2013.

120+

Experts spread over 4 different EU locations.

150+

Clients across multiple industries.

300+

Projects delivered.

10+

Applications each day - we are a talent magnet.

100+M

open source downloads p.a., 4.5k stars

250+

Publications & blog posts.

17%

Of engineer time dedicated to research

ISO 27001

ISO Certified since 2020 + Security, Legal & Ethical AI experts.



TRUSTED BY LEADERS ACROSS INDUSTRIES.



6



RECOGNIZED FOR OUR TALENT & GROWTH.

Don't just take our word for it!



#386 (EU) | #4 BE



AI Innovator of the
year 2020



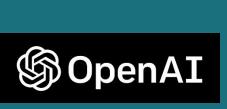
Multiple nominations &
one award win



Nominated in 2022,
2021, 2020, 2019,
2018



Google Cloud partner of
the year BeNeLux
2021-2023



1 of the first 6 official
service partners of
OpenAI in Europe



Scale-up of the year
finalists in 2023



What is MLOps and why does it matter?



AI is everywhere!

A few example of ML applications.

Facial
recognition



Product
recommendation



Email spam
filtering



Autocomplete



Finance
predictions



Healthcare
imaging

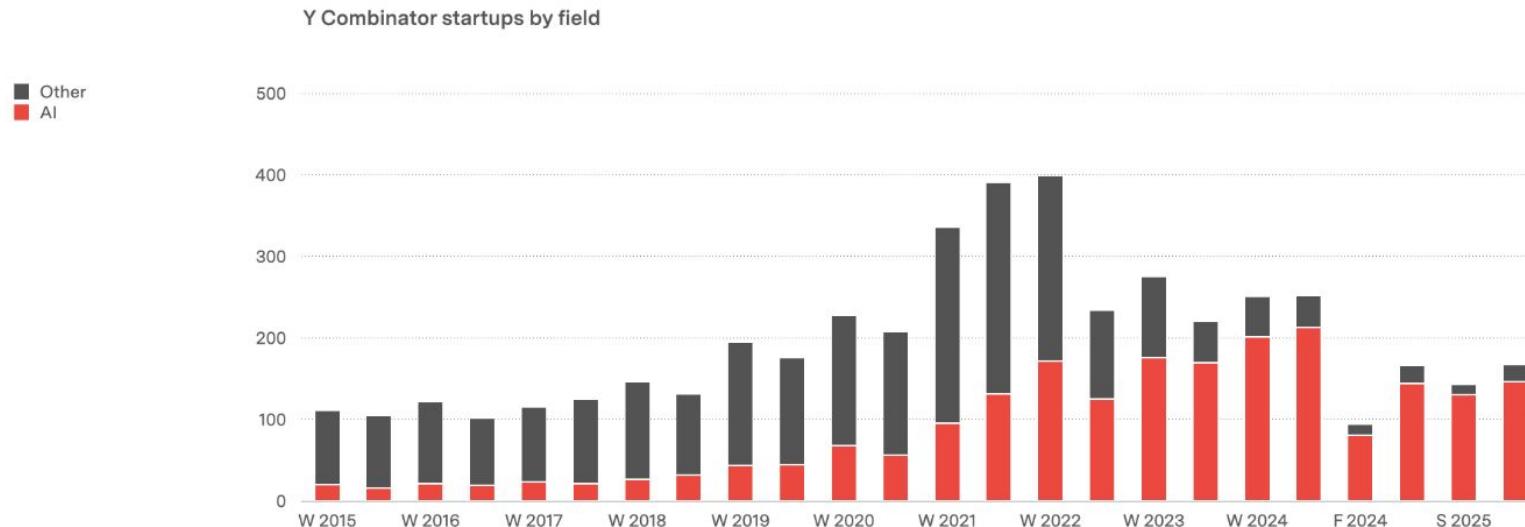


Weather
forecast

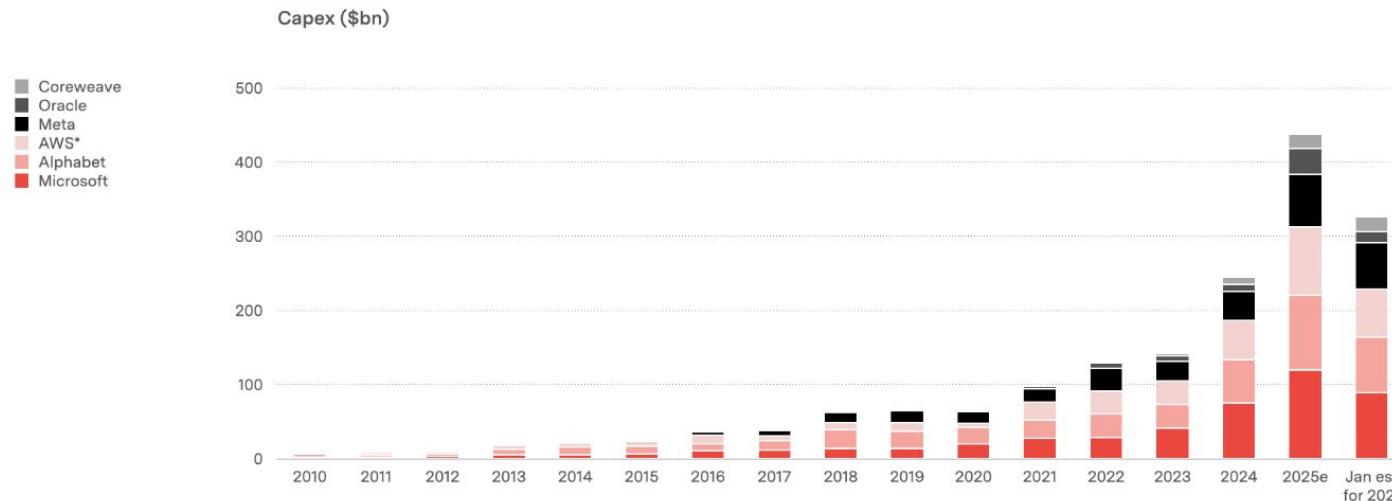


...

Investments are massively focused around AI.



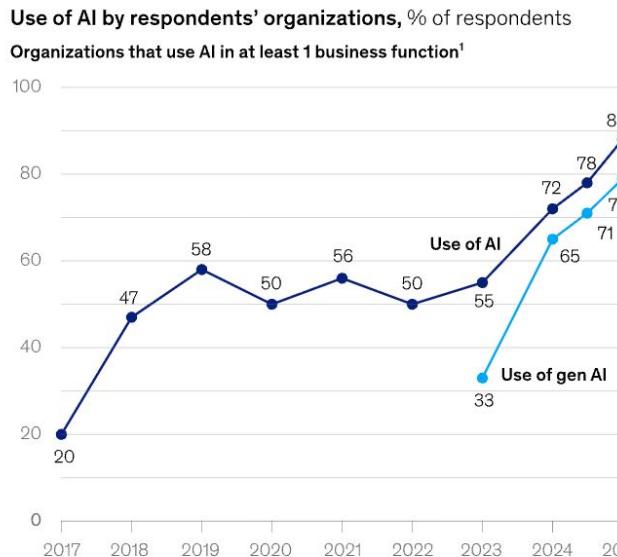
As a consequence, there is a growth in investments (expenditures) in cloud (data centers).



Source: Companies, company guidance. Includes capital leases
* Amazon does not break out AWS capex but reports it as 'the majority'

Benedict Evans -- November 2025 18

Adoption of AI has skyrocketed in the last years.



Phase of AI use among organizations using AI in 2025

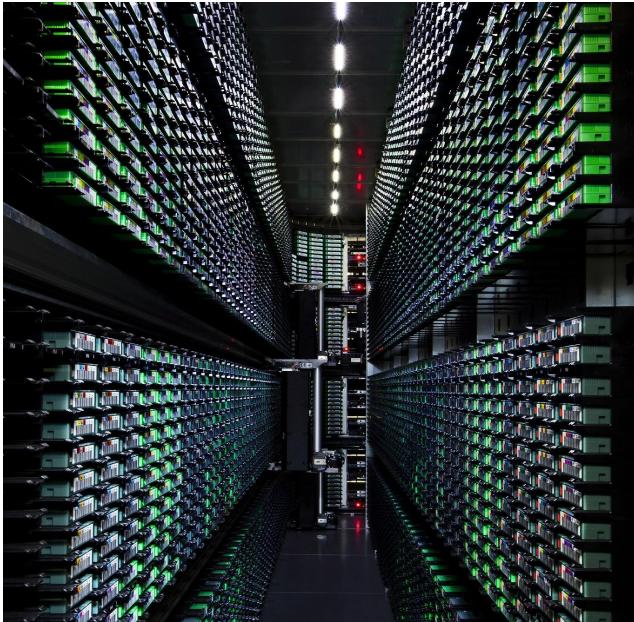


62% are still not in production!

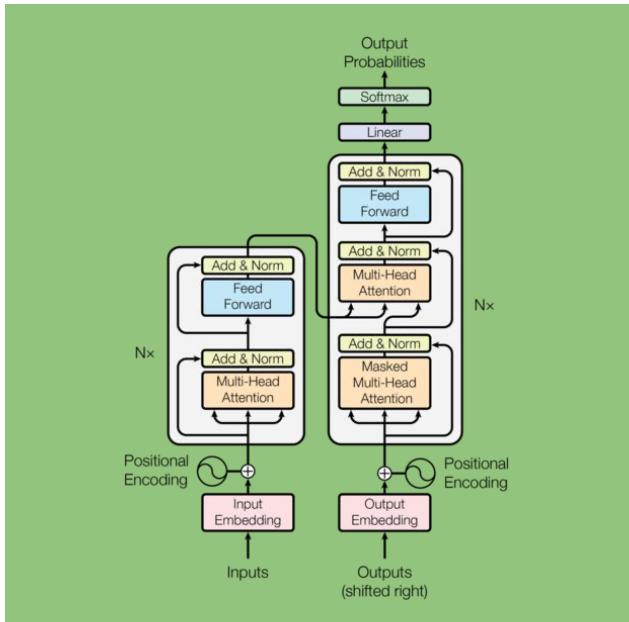
¹In 2017, the definition for AI use was using AI in a core part of the organization's business or at scale. In 2018–19, the definition was embedding at least 1 AI capability in business processes or products. From 2020, the definition was that the organization has adopted AI in at least 1 function, and in 2025, the definition was regular use of AI in at least 1 function.

Source: McKinsey Global Survey on the state of AI, 2017–25

Why now?

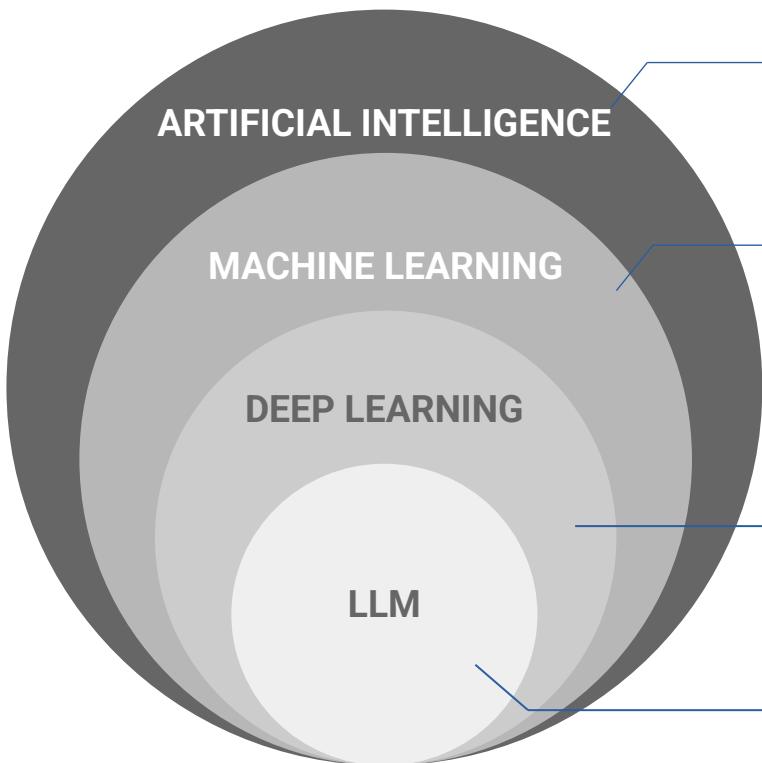


Large Datasets



Lots of Computation

Declinaisons of AI.



ARTIFICIAL INTELLIGENCE

Ability of a machine to perform cognitive functions.

MACHINE LEARNING

AI techniques that give machines the ability to learn from data without being explicitly programmed.

DEEP LEARNING

Type of Machine Learning based on deep neural networks or attention mechanisms.

LARGE LANGUAGE MODELS

Generalised models trained on massive amounts of data, mostly used through an API or open sourced model.

Why do we need ML Systems Design?

Building a ML application means implementing much more than just your ML model.

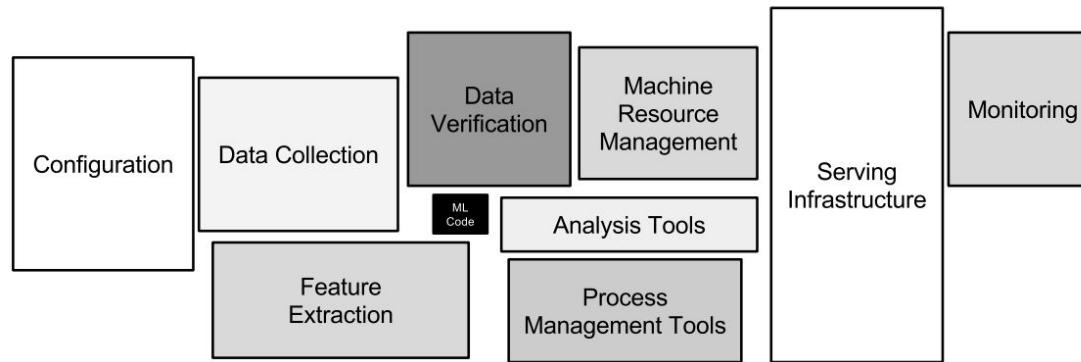


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Key definitions.

ML Application: The final solution or program powered by a Machine Learning model.

ML System: All the components responsible for the implementation and management of the data and models powering an ML application.

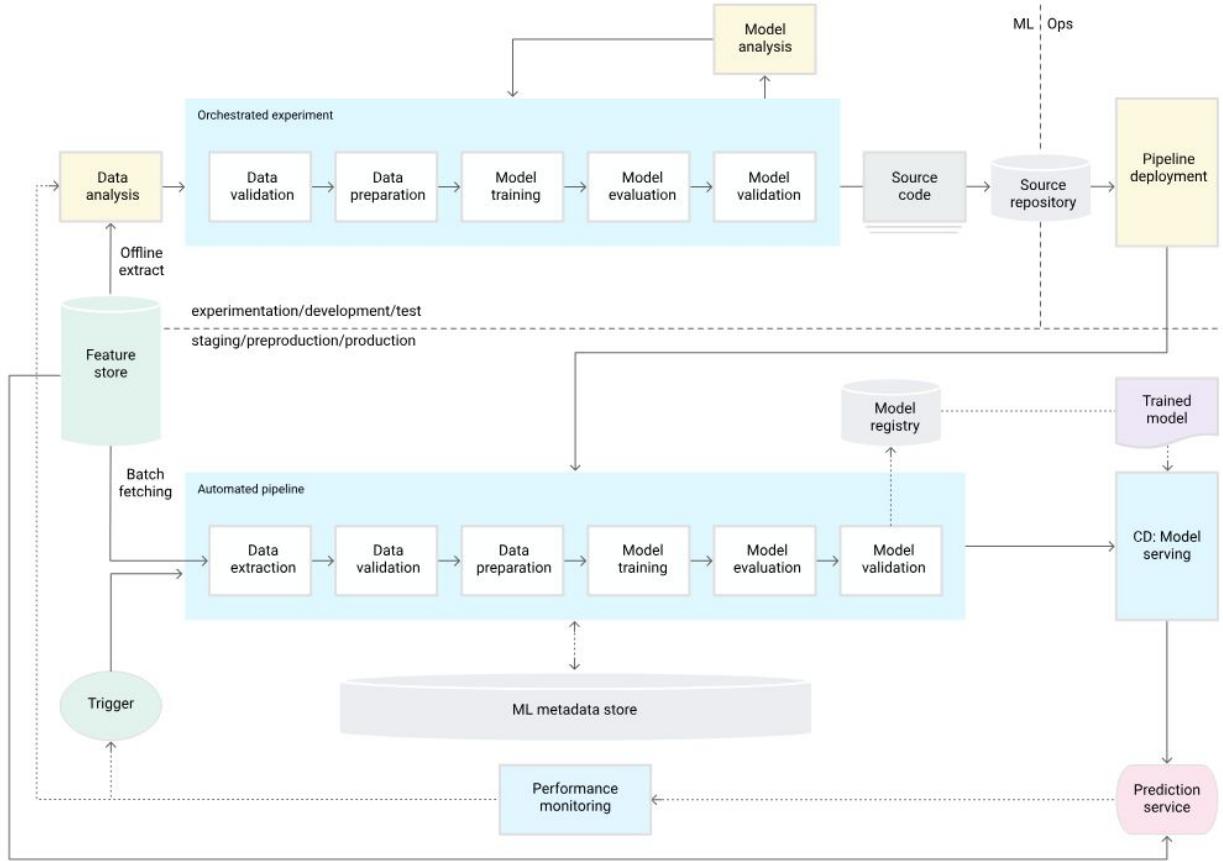
ML Systems Design: The act of designing the architecture and implementing an ML System.

MLOps: A set of practices and tools to enhance and accelerate the entire ML model development life cycle.

LLM Ops: A branch of MLOps specifically focusing on LLM application development life cycle.

Key concepts of MLOps

Typical architecture of an ML system



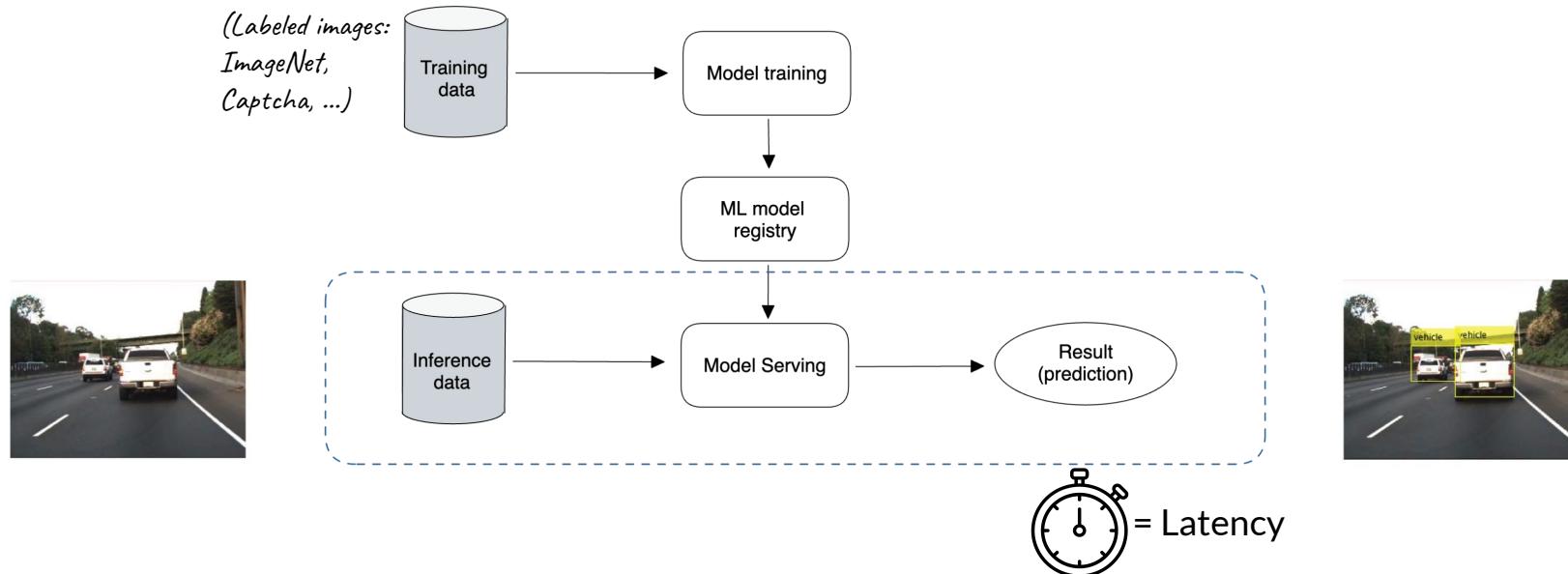
Data preparation

It all starts with data. How to go through all these steps efficiently and effectively.



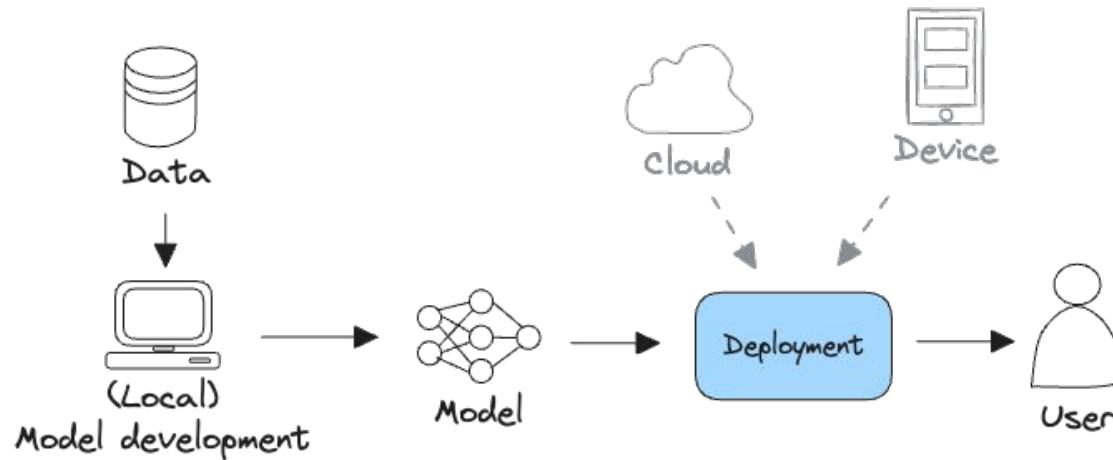
ML model serving

How to efficiently serve ML model to client.



ML model deployment

How to efficiently deploy your model for serving.



Containerisation

Containers encapsulate an application as a **single executable package** that contains all the information to **run it on any hardware**:

- Application code
- configuration files
- libraries
- dependencies

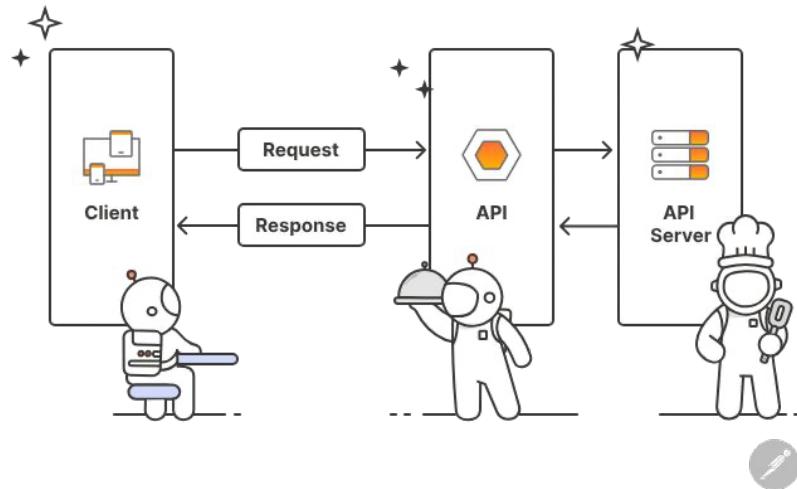
Abstracts the application from its **host operating system**.

Containers can be easily transported from a desktop computer to a virtual machine (VM) or from a Linux to a Windows operating system, and they will run consistently on virtualized infrastructures or on traditional “bare metal” servers, either on-premise or in the cloud.



APIs

Allow other services to call your model or application.



An **Application Programming Interface (API)** is a set of protocols that enable different software components to communicate and transfer data.

Developers use APIs to bridge the gaps between small, discrete chunks of code in order to create applications that are powerful, resilient, secure, and able to meet user needs.

Cloud infrastructure

Cloud infrastructure allow for data storage, compute allocation, training and deploying model, monitoring, ...

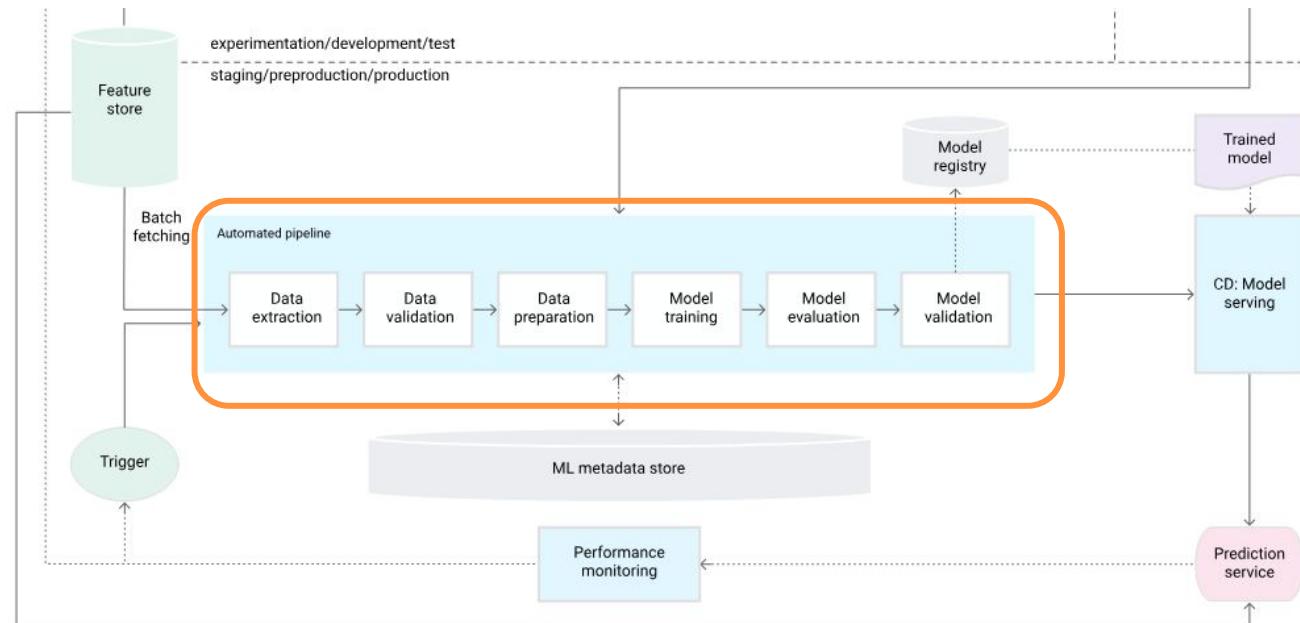


Google Cloud



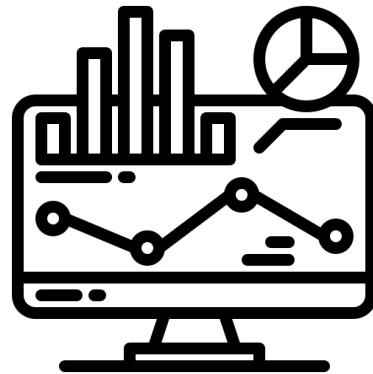
ML Pipeline

Orchestrates components to prepare data, train, evaluate and deploy ML models
(among other things)



Monitoring

Ensuring that models in production are performing well.



Resource level (performance and usage of resources)

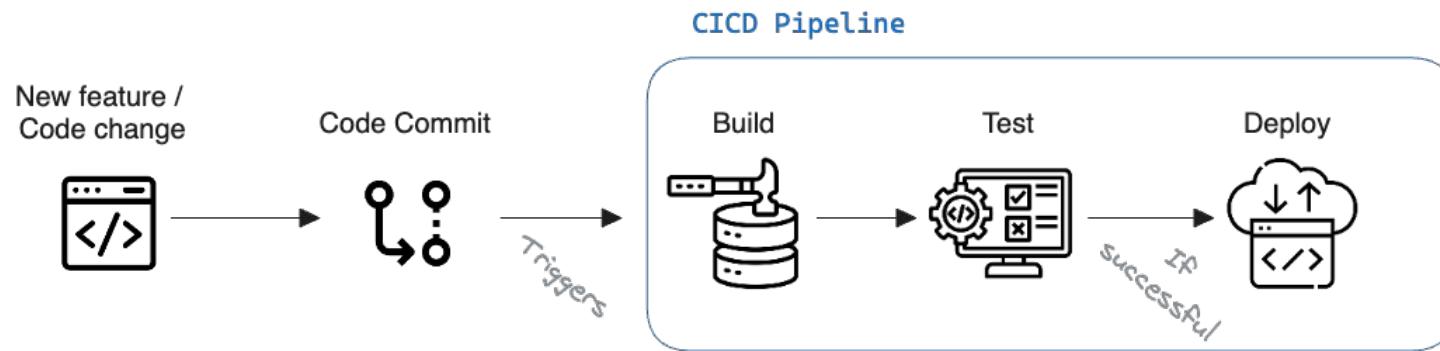
- How much is it being used by users?
- Are the CPU, RAM, network usage, and disk space as expected?
- What are the Cloud costs?
- Are requests being processed at the expected rate?
- What is the system uptime? Some maintenance contract depend on it.

Performance level (performance/accuracy of the model over time)

- Is the model still doing accurate predictions with the new data coming in?
- Is the data distribution changing?
- Is the target variable changing?
- Are concepts around the model changing?

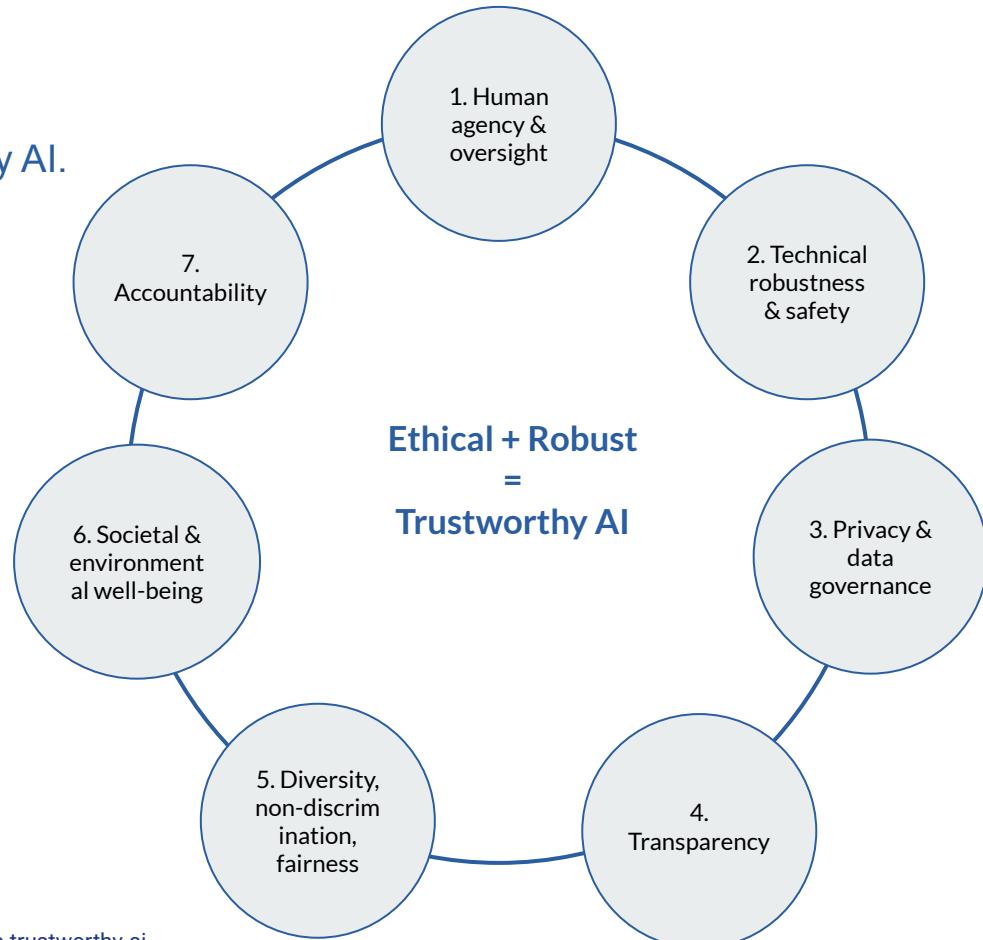
Continuous Integration and Continuous Delivery (CICD)

Allows you to continuously work on your application and efficiently deploy new changes to it.



Key concept: Ethical AI

Guidelines & legislation on building trustworthy AI.

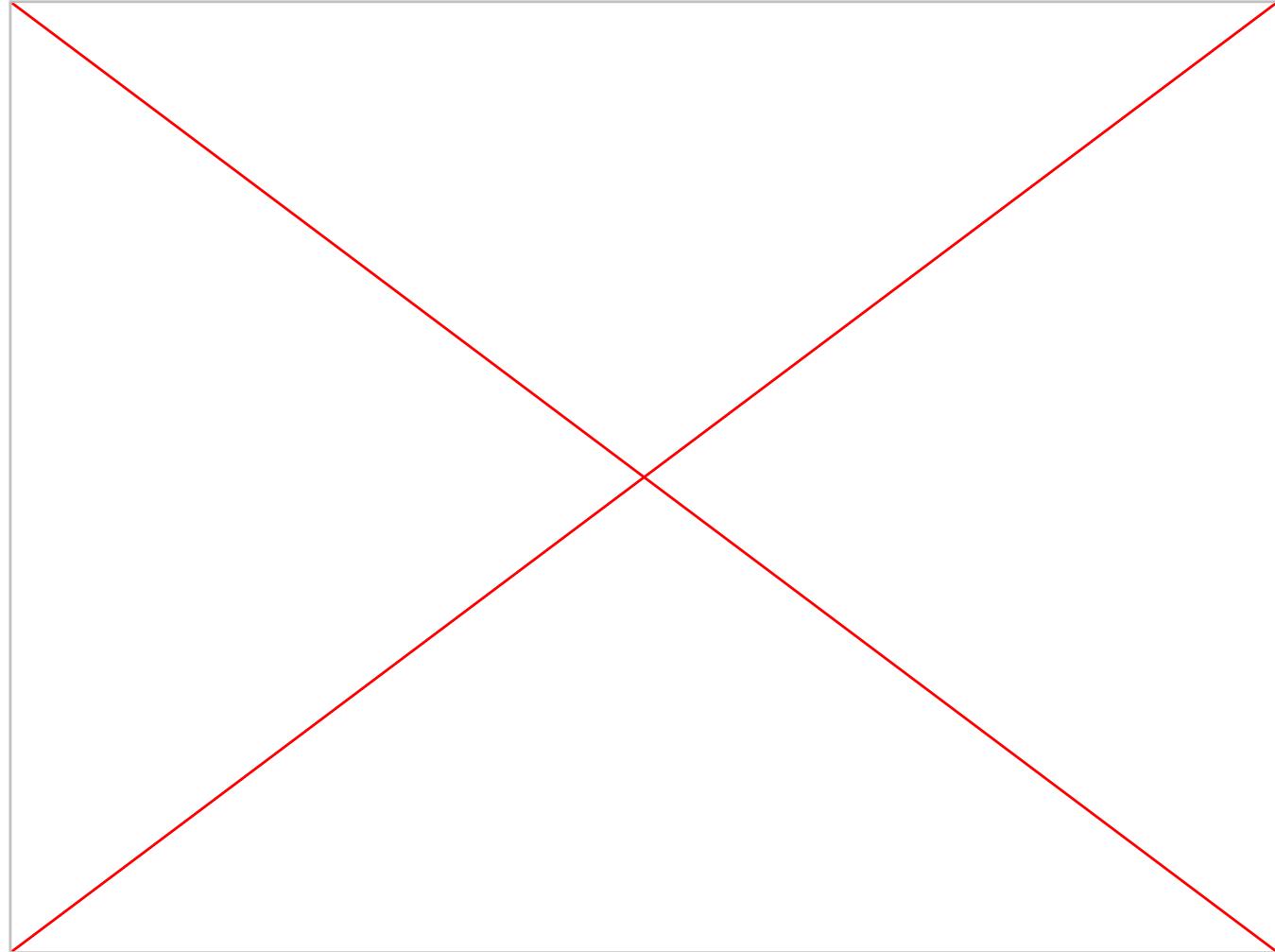


Teams can adopt different MLOps maturity levels



Level	Highlights	Technology
Level 0 No MLOps	<ul style="list-style-type: none">Difficult to manage full ML model lifecycleTeams are disparate and releases are painful"black boxes," little feedback during/post deployment	<ul style="list-style-type: none">Manual training, builds and deploymentsManual testing of model and applicationNo centralized tracking of model performance
Level 1 DevOps but no MLOps	<ul style="list-style-type: none">Releases are less painful than No MLOpsLimited feedback on how well a model performs in productionDifficult to trace/reproduce results	<ul style="list-style-type: none">Automated buildsAutomated tests for application code
Level 2 Automated Training	<ul style="list-style-type: none">Training environment is fully managed and traceableEasy to reproduce modelReleases are manual, but low friction	<ul style="list-style-type: none">Automated model trainingCentralized tracking of model training performanceModel management
Level 3 Automated Deployment	<ul style="list-style-type: none">Releases are low friction and automaticFull traceability from deployment back to original dataEntire environment managed: dev > test > production	<ul style="list-style-type: none">Integrated A/B testing of model performanceAutomated tests for all codeCentralized tracking of model training performance
Level 4 Full MLOps	<ul style="list-style-type: none">Full system automated and easily monitoredAutomated feedback collection and retrainingClose to zero-downtime	<ul style="list-style-type: none">Automated model training and testingVerbose, centralized metrics from deployed model

Going from
standard ML
Engineer to
MLOps master...



A zoom on LLMOps

What defines LLMOps?

LLMOps is a subset of MLOps aimed at Generative AI (GenAI) and Large Language Model (LLM) applications.

- It aims at unifying LLM system development (LLM/Dev) and operation (Ops).
- It provides **practices** and **tools** to enhance and accelerate the entire LLM application development life cycle.
- It enables developer teams to implement LLM applications in a more **efficient, consistent, reproducible, and secure** way.

Why LLMOps is important now?



The **potential** and **value** added of GenAI is skyrocketing. Organisations are keen to enable business value by using LLMs in production in an efficient, robust and safe way.



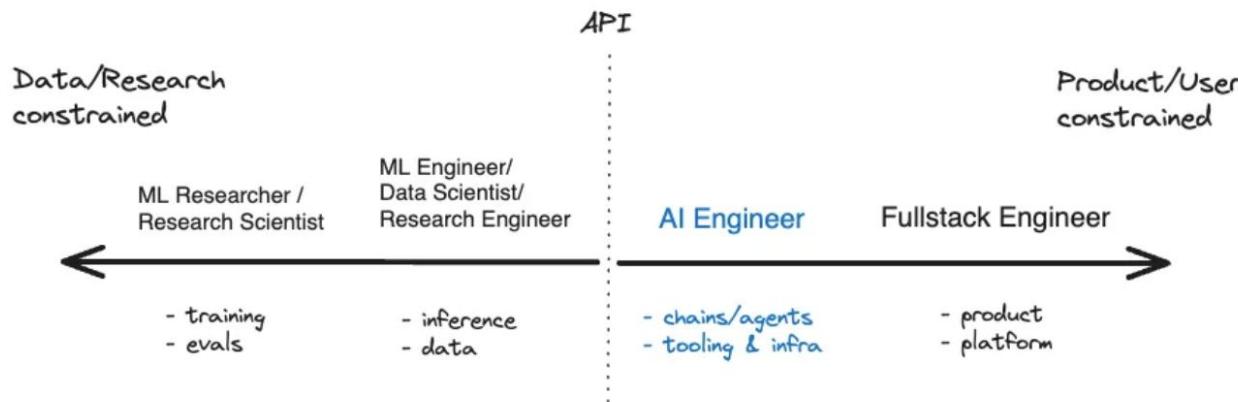
Technologies around GenAI are evolving every day. This is why a proper infrastructure is a must have, to enable an effective and productive ML team



Having a proper ecosystem, enables organizations to **accelerate development** and iterate faster, limits human error and creates a safe and seamless environment to nurture new experiments.

LLMs introduced the need for a new role: AI Engineers.

AI Engineering primarily involves working on GenAI and LLM applications, which uses pre-trained models accessed via APIs or open source. This excludes custom model training, but introduces a new set of unique engineering practices.



LLMs introduced a new set of engineering practices.

A few examples...

LLM HOSTING

LLMs are heavier and more challenging to host than traditional ML model. Hosting and optimisation requires new techniques.

CONTEXT ENGINEERING

Optimise prompt and context provided to the LLMs can lead to significant differences in latency, cost and performance.

EVALUATION

Evaluating LLM applications is inherently difficult. Different techniques exist for this purpose.

TOOLS / MCP

Implementing tools and optimising their usage by LLMs agents is key to building modern AI applications.

GENAI GATEWAY

Providing centralized access control, governance and observability for LLM applications.

GUARDRAILS MANAGEMENT

Protect the input and output data of an LLM application with different guardrail techniques.

What defines MLOps?

MLOps in a nutshell:

MLOps aims at unifying ML system development (ML/Dev) and operation (Ops).

It provides practices and tools to enhance and accelerate the entire ML model development life cycle.

It enables developer teams to implement ML applications in a more efficient, consistent, reproducible, and secure way.

KPIs unlocked by MLOps:

Shorter time
to deployment

Increased
uptime

Accelerate
application
development

Decreased
error rate

Optimise
IT infra cost

Increase
traceability &
observability

What defines LLMOps?

The core value of MLOps still holds for LLMOps

Just like MLOps, LLMOps is focused on bringing solutions to production efficiently & robustly. The KPIs remain applicable.

Shorter time
to deployment

Increased
uptime

Accelerate
application
development

Decreased
error rate

Optimise
IT infra cost

Increase
traceability &
observability

GenAI
applications have
their own
dynamics

Unlike traditional ML models, GenAI applications leverage the power of large language models, introducing new challenges across the different development and deployment cycles.

This shift requires adapting your current practices and adopting new ones to ensure effective handling of data, experiments, and deployments.

When comparing LLMOps to MLOps:

- PRINCIPLES ARE THE SAME
- BENEFITS ARE THE SAME
- BUT SKILLS ARE DIFFERENT

When comparing LLMOps to MLOps:

MLOps

- Preparing data for model development (cleaning, feature engineering, etc...)
- Sourcing and labelling of data

DATA MANAGEMENT

- Experiments focused on altering model parameters and architecture, resulting in new model
- Evaluation on hold out test sets with precise metrics

EXPERIMENTS & DEVELOPMENT

- Automated deployment of code and models across different environments
- Versioning and tracking of models
- Set up of A/B experiments

DEPLOYMENT

- Logging and monitoring of data drifts and model performance

CONTINUOUS MONITORING & LOGGING

LLMOps

- New data stack, with elements like vector databases, embedding models, document parsing, etc...
- Data access and structure gain relevance for retrieval augmented generation (RAG)

- Experiments revolve around prompt engineering and agentic workflows
- Importance model choice and future-proofing by enabling model swap
- Evaluation is ambiguous and challenging

- Tracing of LLM agents logs
- Versioning and tracking of prompts
- Guardrails and defensive mechanisms
- User testing and validation before release

- Monitoring for biases, hallucination and other risks
- Use LLM-as-a-Judge (LAAJ)
- Latency & cost optimisation - wide range of motion

Real-world use case:

Grid system imbalance
forecasting

1 Use case: What is System Imbalance?

2 Initial ML model iteration

3 MLOps solution



What is System Imbalance?



Balancing the Grid is Crucial

Therefore any imbalance must be counteracted

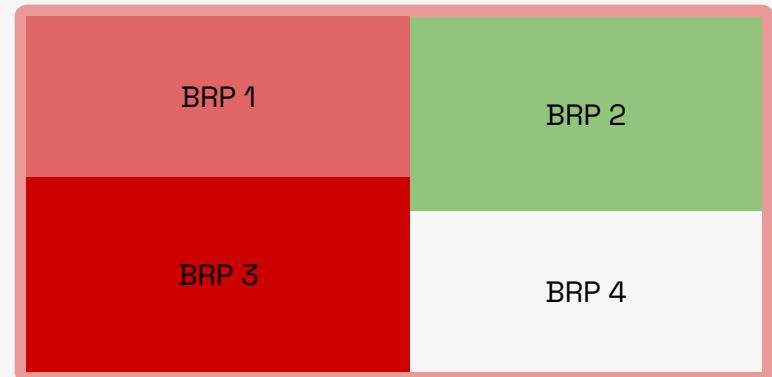
- Electricity grid frequency must be stable at 50hz
- Task of Belgium's Transmission System Operator → Elia
- System imbalance (SI) is the unintended difference between generation and load after balancing actions were performed



SI Forecasting: What is the purpose?

And why is it relevant for ML6?

- Accurate SI forecasts useful for Balancing Responsible Parties (BRPs)
- BRPs are mainly energy suppliers/retailers (e.g. ENGIE, Luminus)
- BRPs must adapt their own energy asset portfolio based on SI



TSO area

red - negative system imbalance
green - positive system imbalance



System Imbalance Forecasting

Goal of the project:

We worked with Luminus to implement ML models that forecast System Imbalance (SI) with higher accuracy than their existing solution.



SI Forecasting: Key Challenges

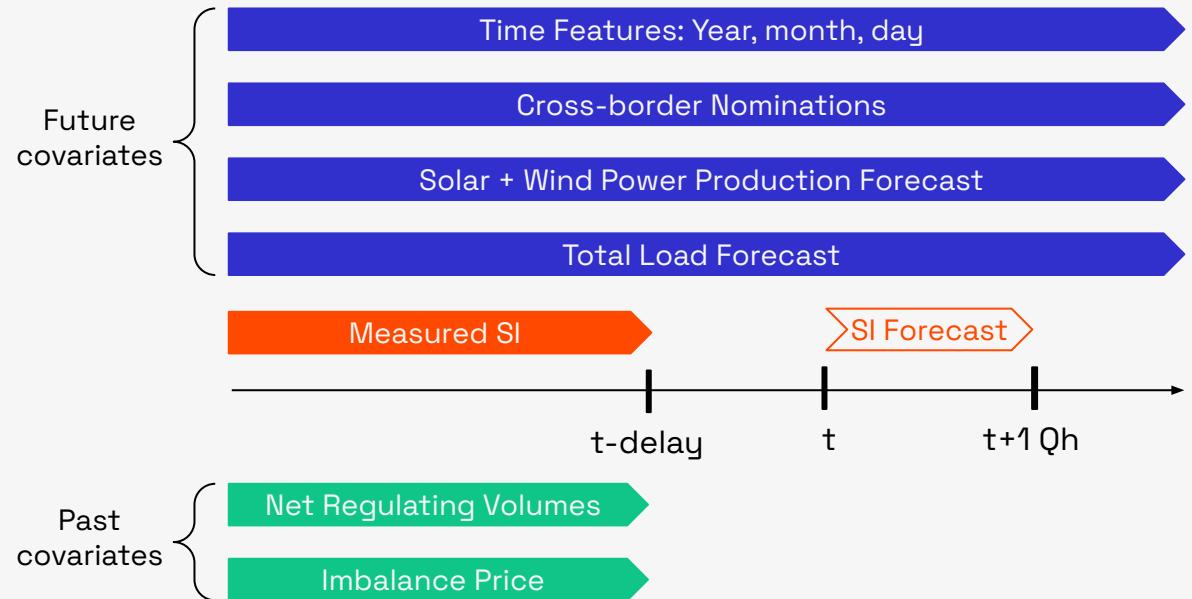


Initial ML model iteration



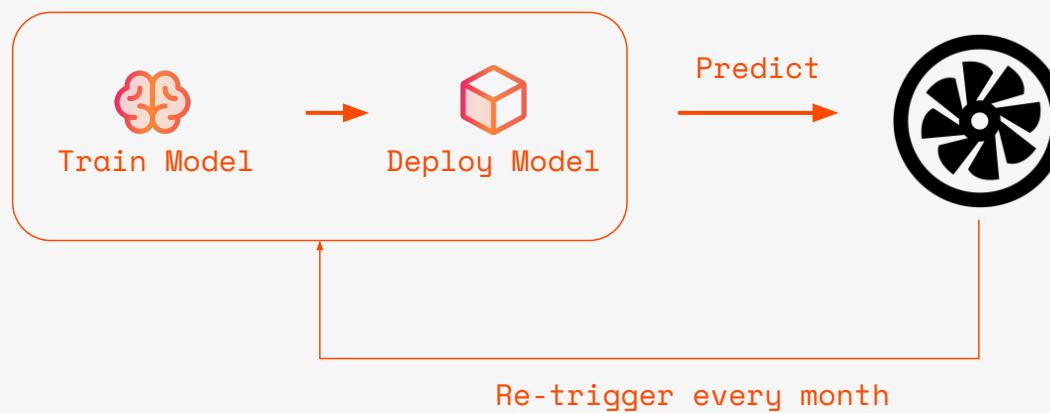
SI Forecasting: Input Data

- Data Source: Elia Open Data Platform
- Past covariates have a data delay
- Goal: Forecast future SI using SI measurement, future and past covariates
- Model: XGBoost



What happens if we don't implement MLOps?

Use case

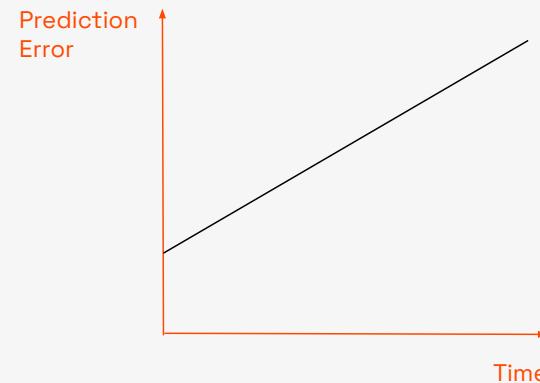


What happens if we don't implement MLOps?

Initially things worked fine



Until they didn't



Setting the scene

What happens if we don't implement MLOps?

When and why did it fail?

- Did the training data start drifting?
- Did the re-training pipeline fail?
- Was the model poorly designed in the first place?

If

-
-
-
-
-
-



MLOPS

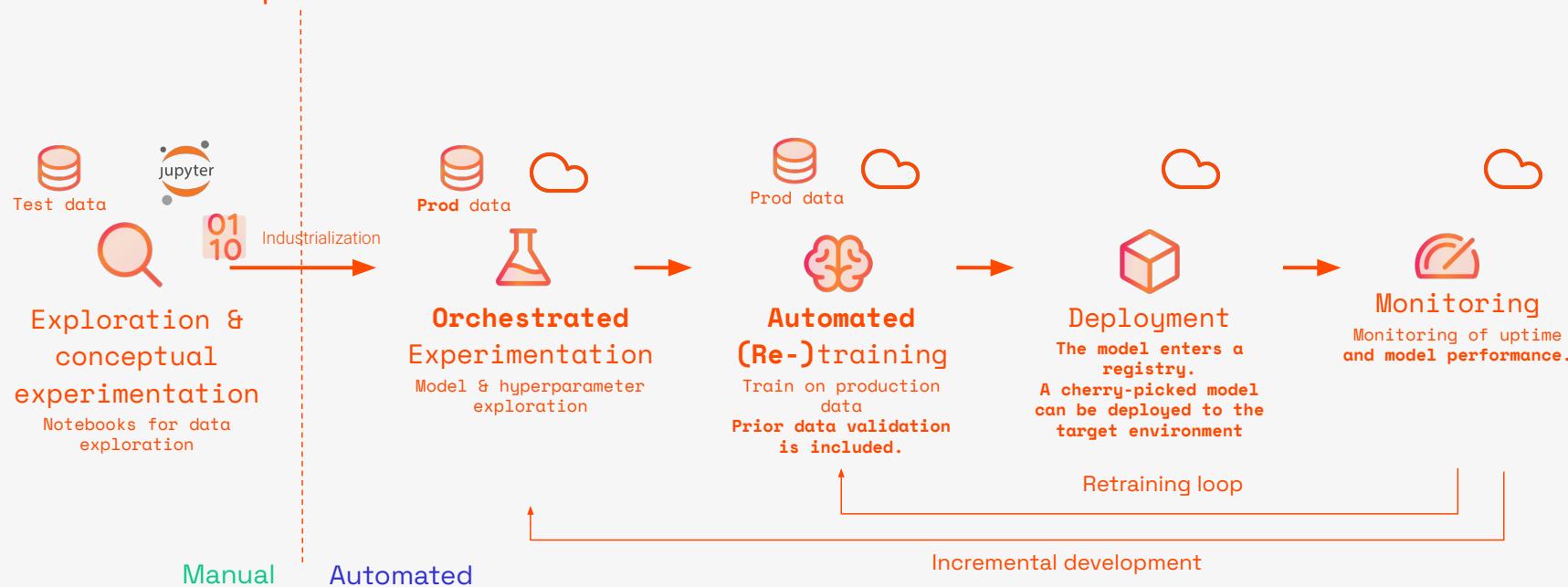
atistics

e our



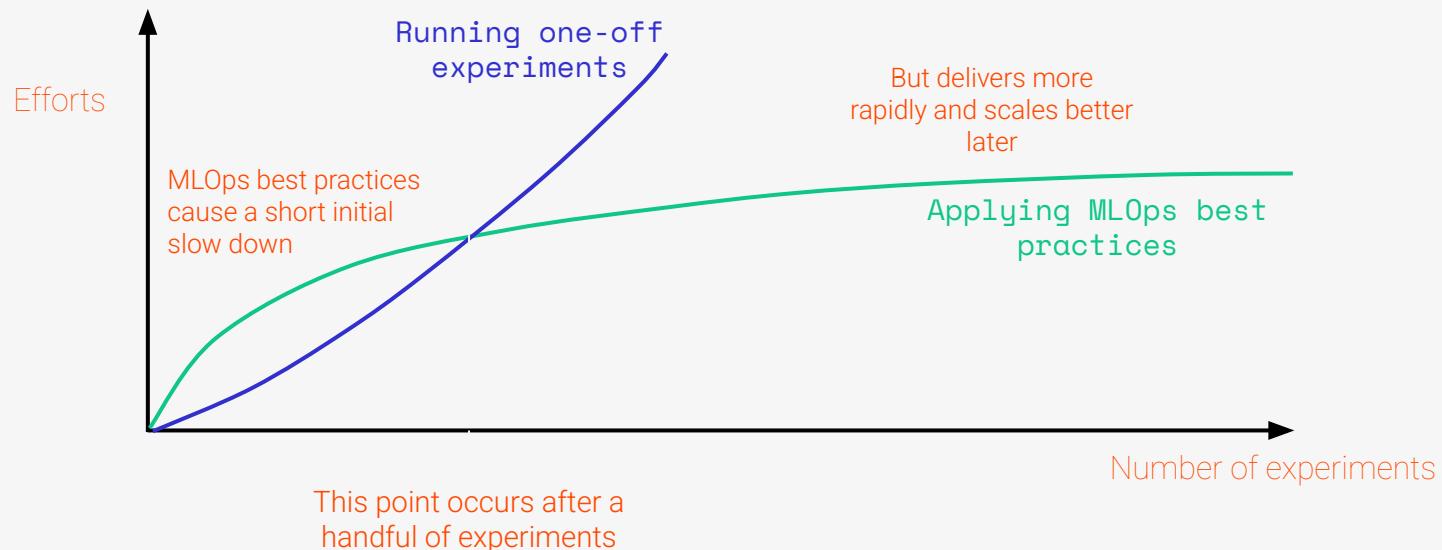
How does MLOps work?

Streamlined MLOps



Why MLOps?

It pays off in the long term!

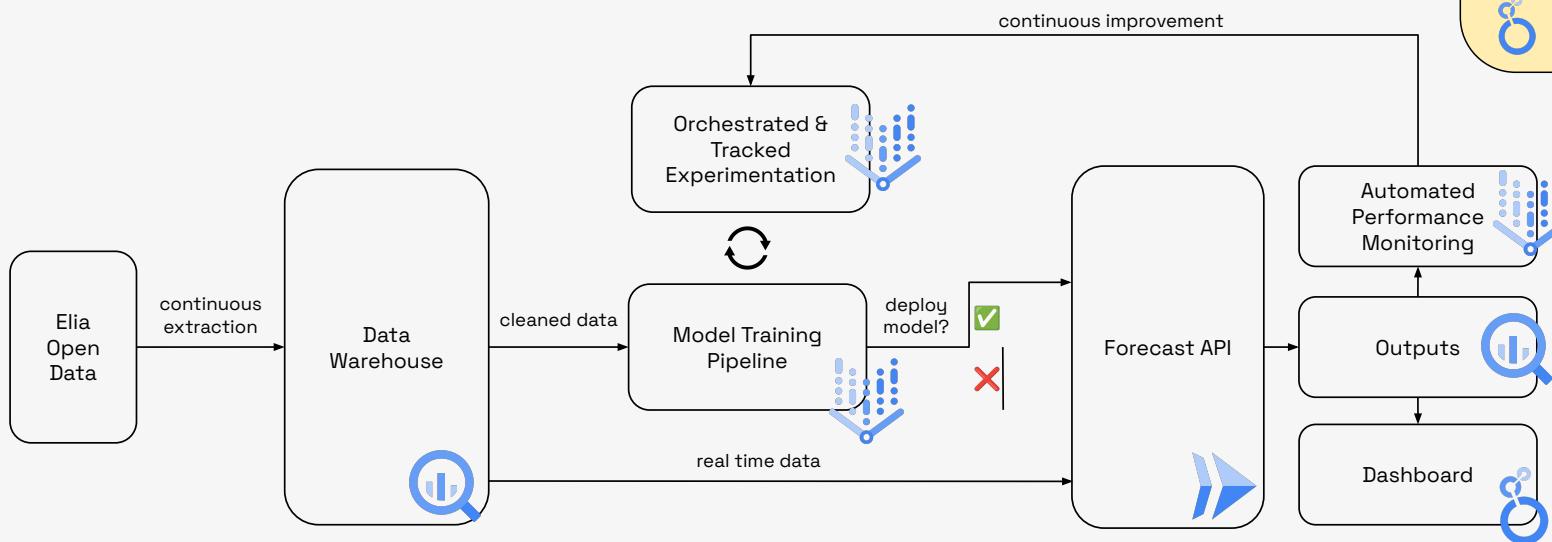


MLOps solution



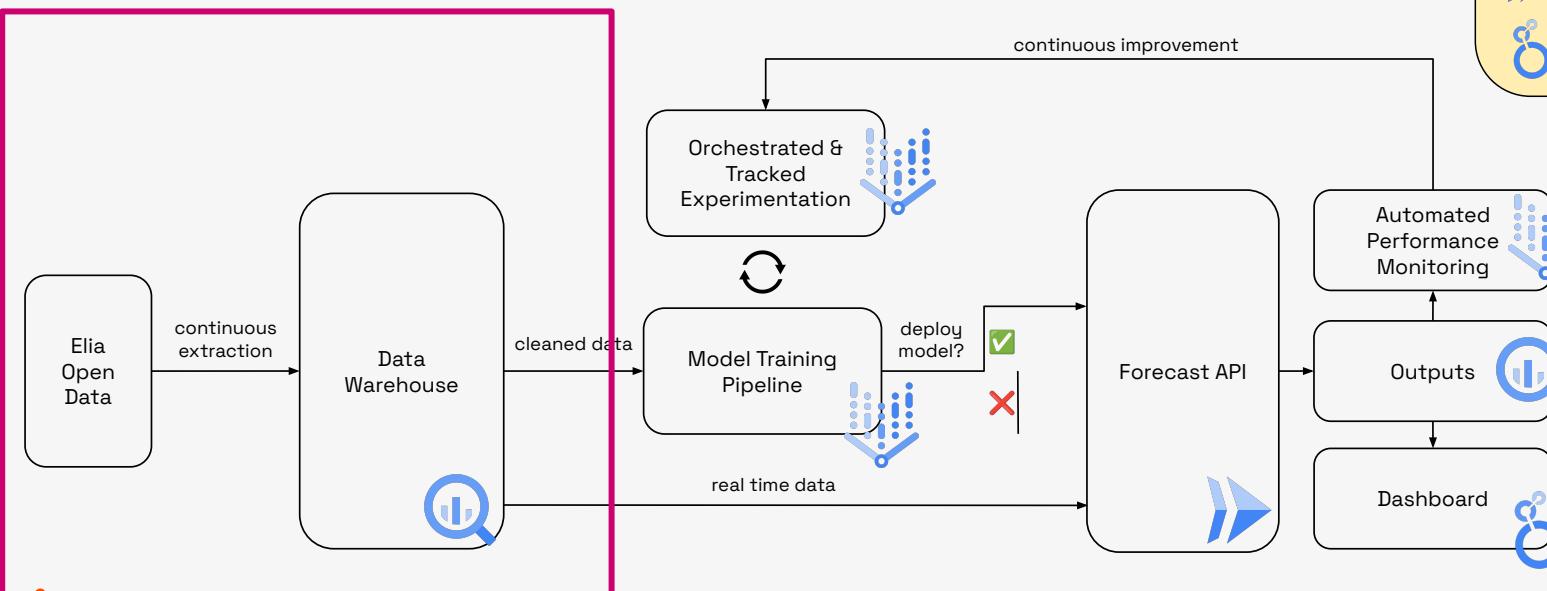
Overview of the solution

Automated model training & inference in Google Cloud

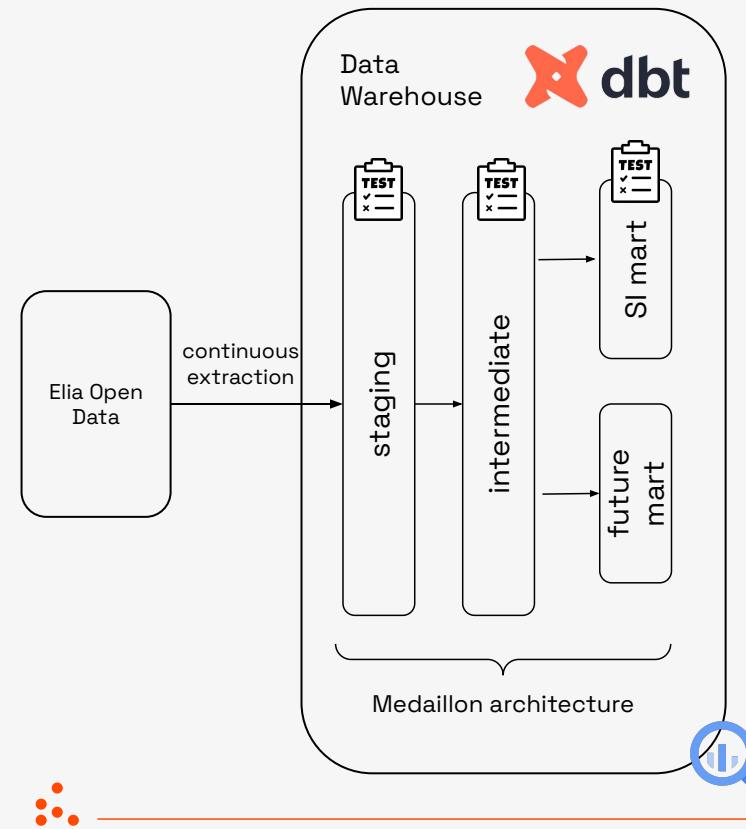


Overview of the solution

Automated model training & inference in Google Cloud



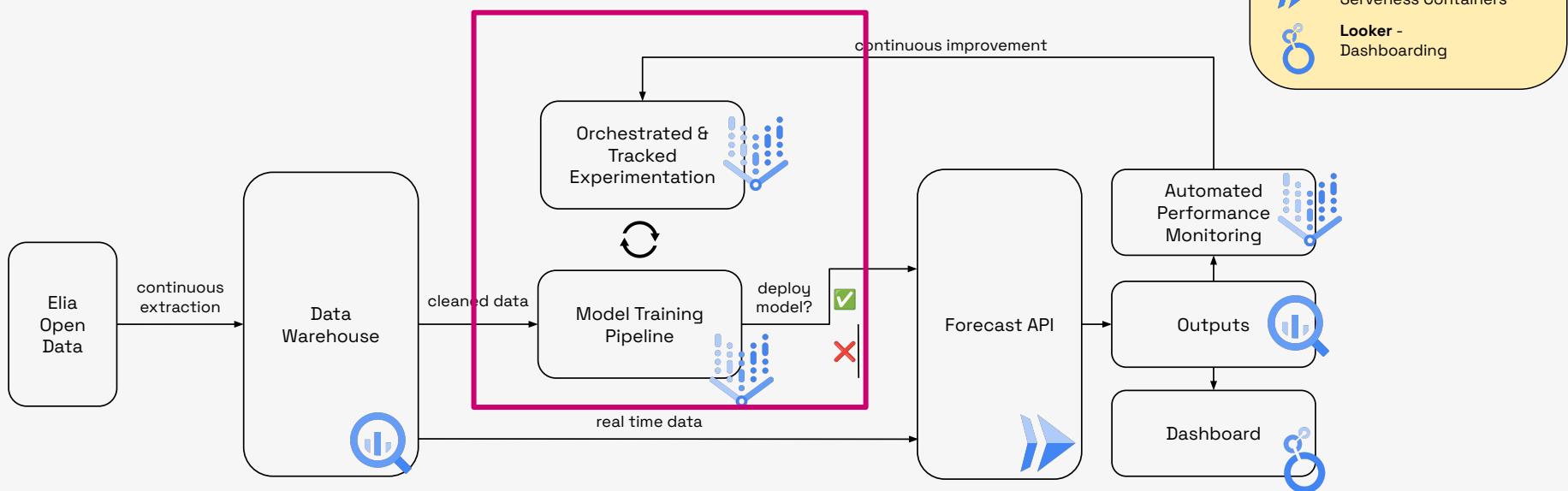
Data Layer



Data Layer Requirements	Solution
Receive live input data	CRON triggered Cloud functions
Data must be cleaned, transformed and exposed	Data warehouse medaillon architecture
Data must have certain formats	Verify via dbt tests

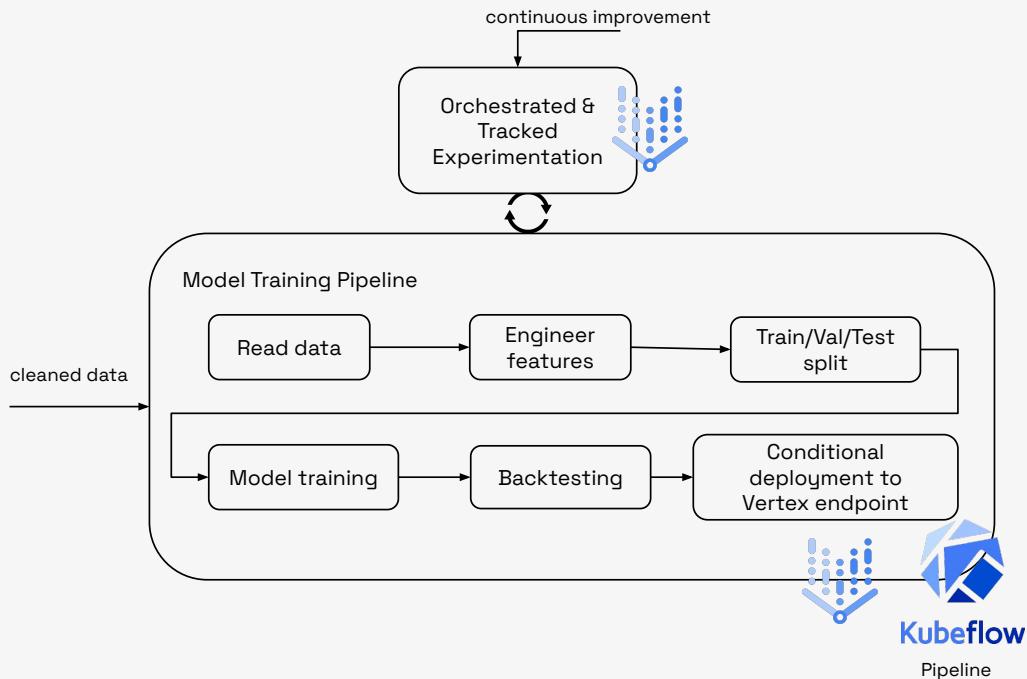
Overview of the solution

Automated model training & inference in Google Cloud



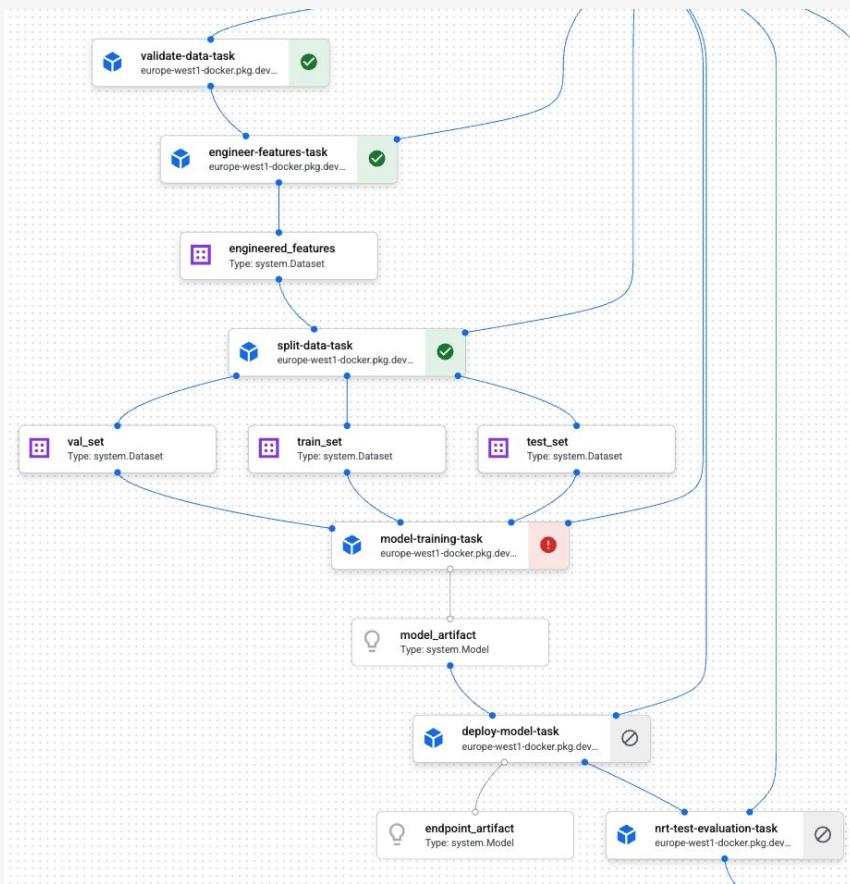
ML Pipeline

Conceptual flow



- Automated
- Reproducible
- Traceable
- Modular
- Improves collaboration



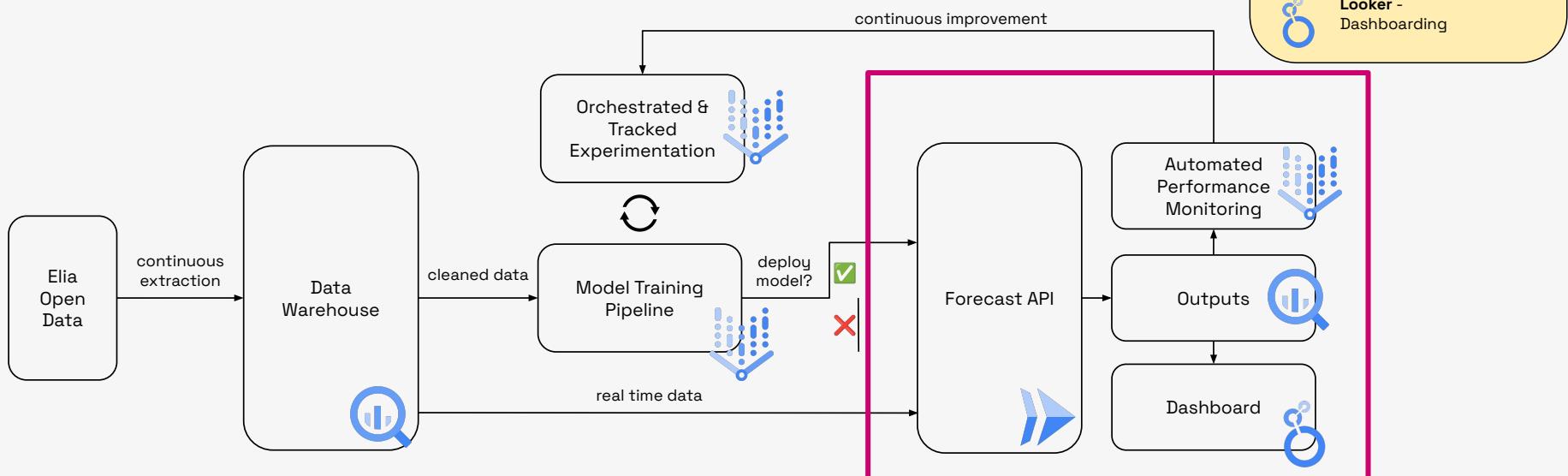


ML Pipeline

Overview of their Vertex Pipeline

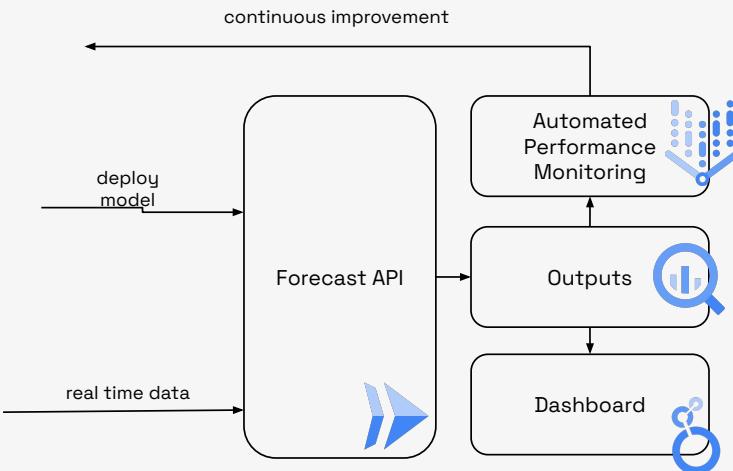
Overview of the solution

Automated model training & inference in Google Cloud



Model serving

REST API - deployed in a Cloud Run instance



- REST API is used to serve the ML models
- It is packaged in a Docker container
- Which is deployed in a Cloud Run instance

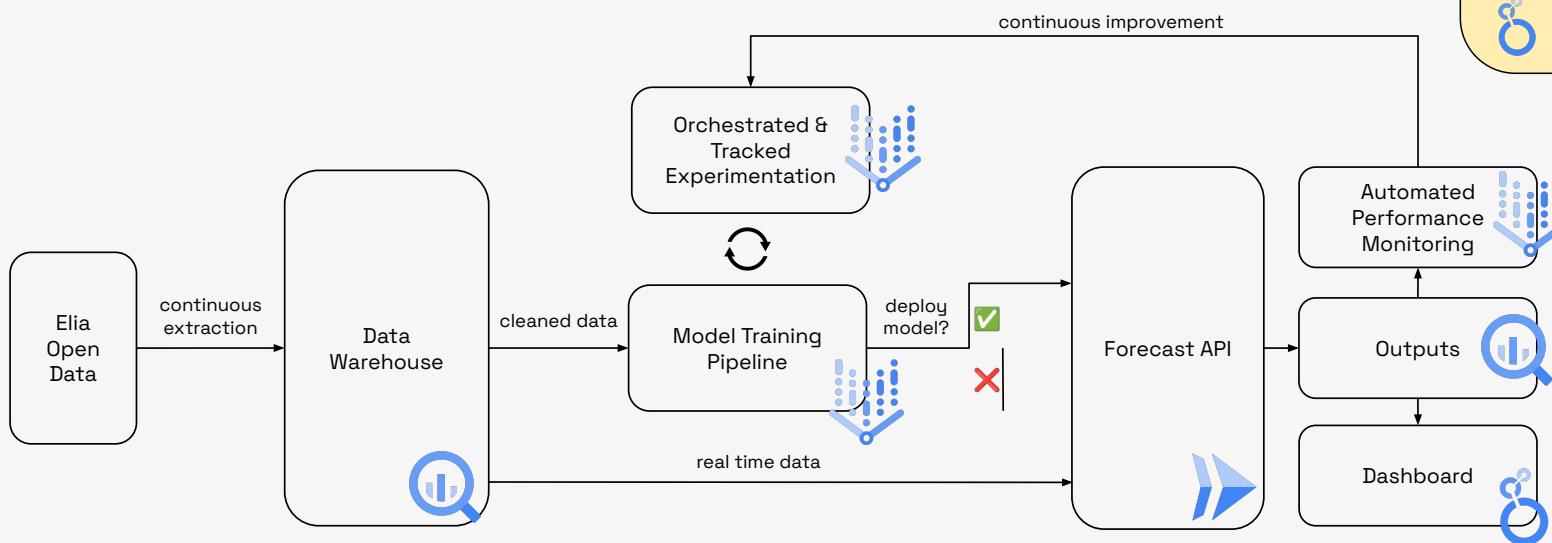
Downstream systems can make a “request” to the API with input data and get forecasts. The API is served in “real time” (online serving).

Each outputs are stored. A monitoring dashboard is built using **Looker**.

(In this course we will see Streamlit for dashboarding)

Overview of the solution

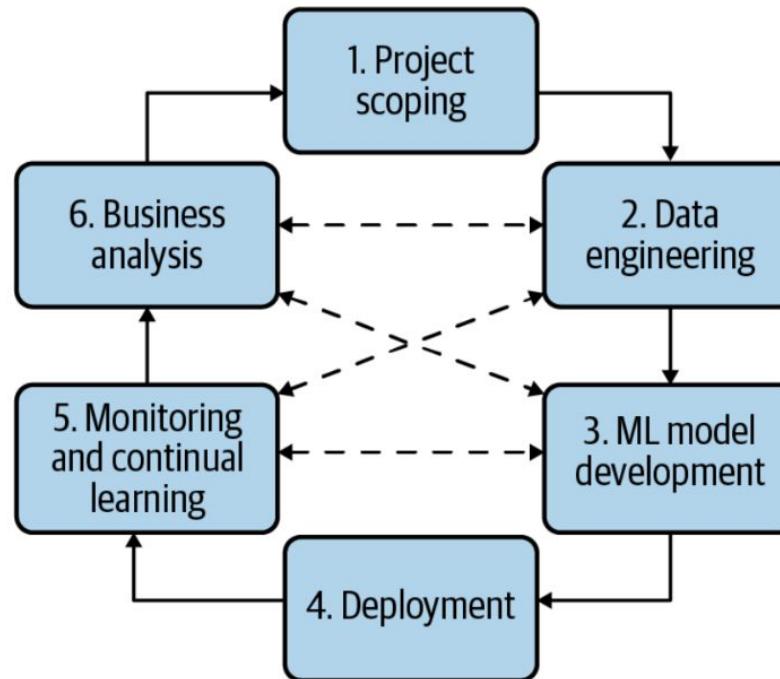
Automated model training & inference in Google Cloud



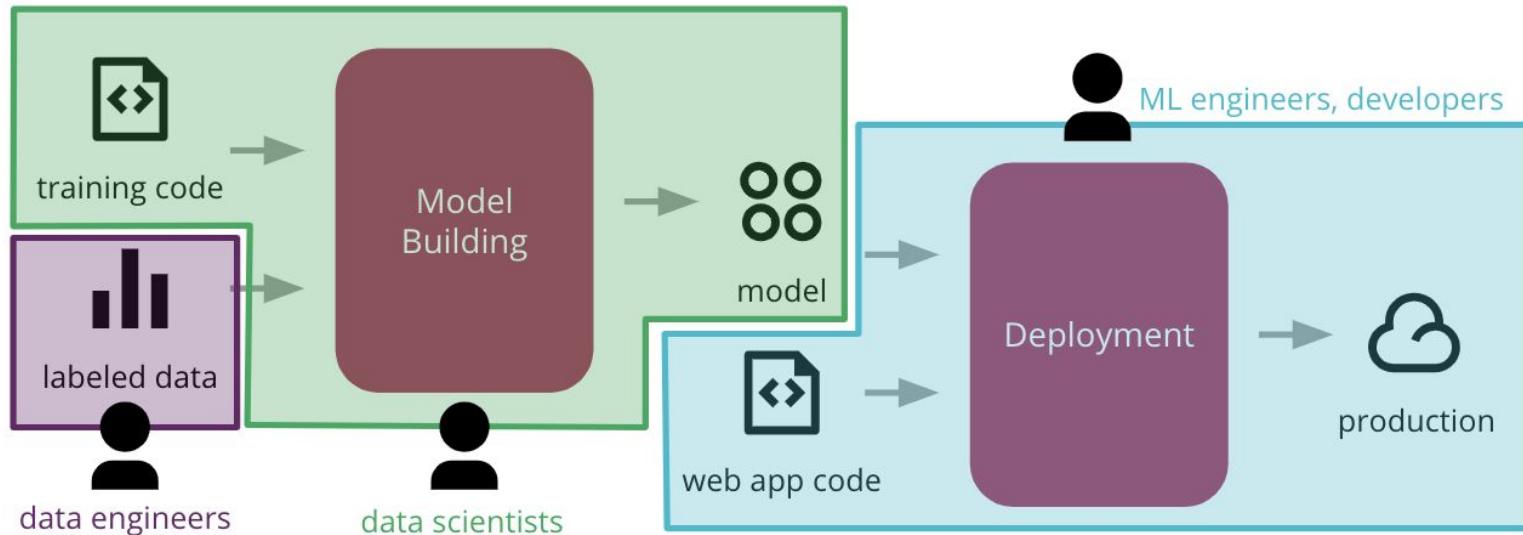
ML project organisation

Roles around ML projects

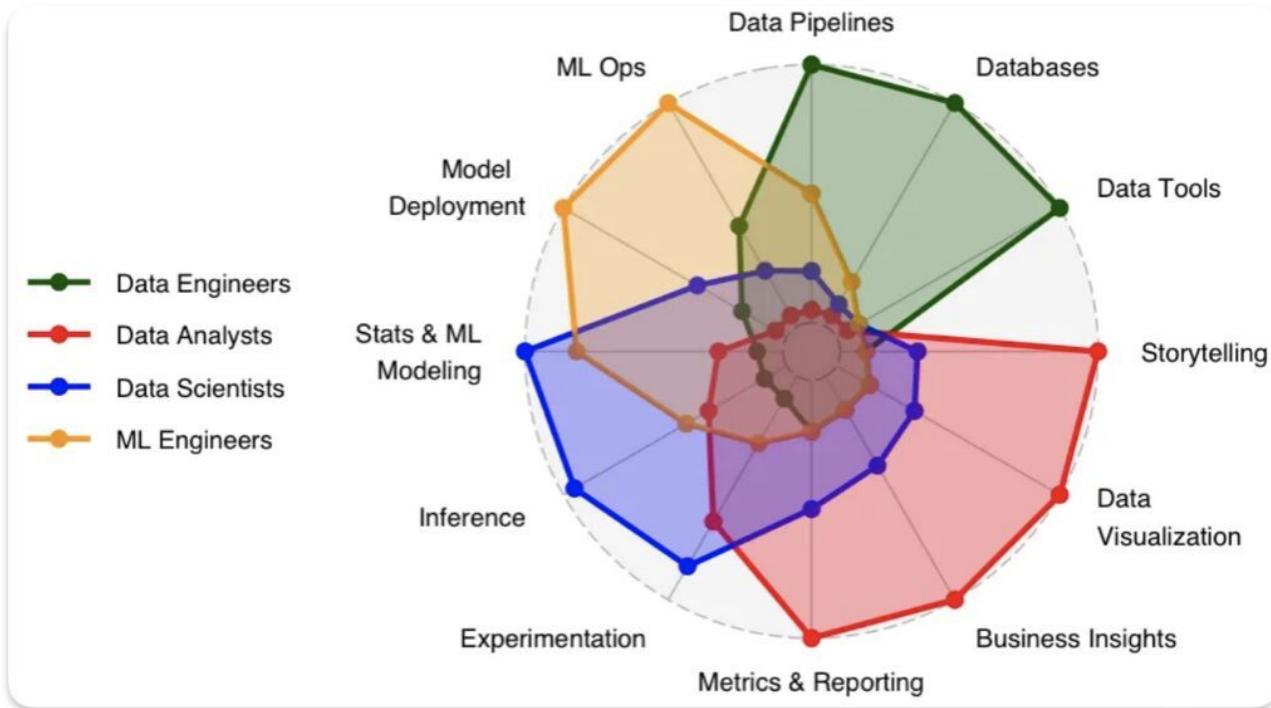
Typical ML project lifecycle.



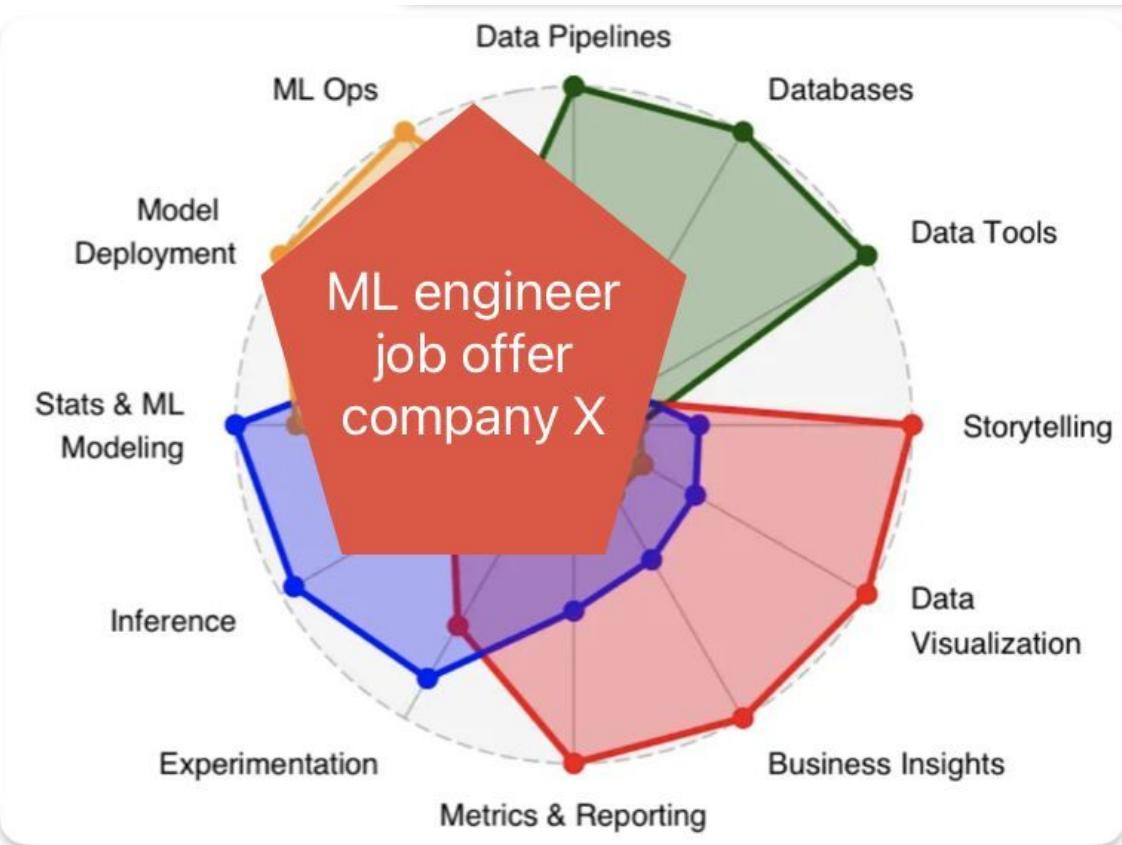
Roles around a ML system implementation.



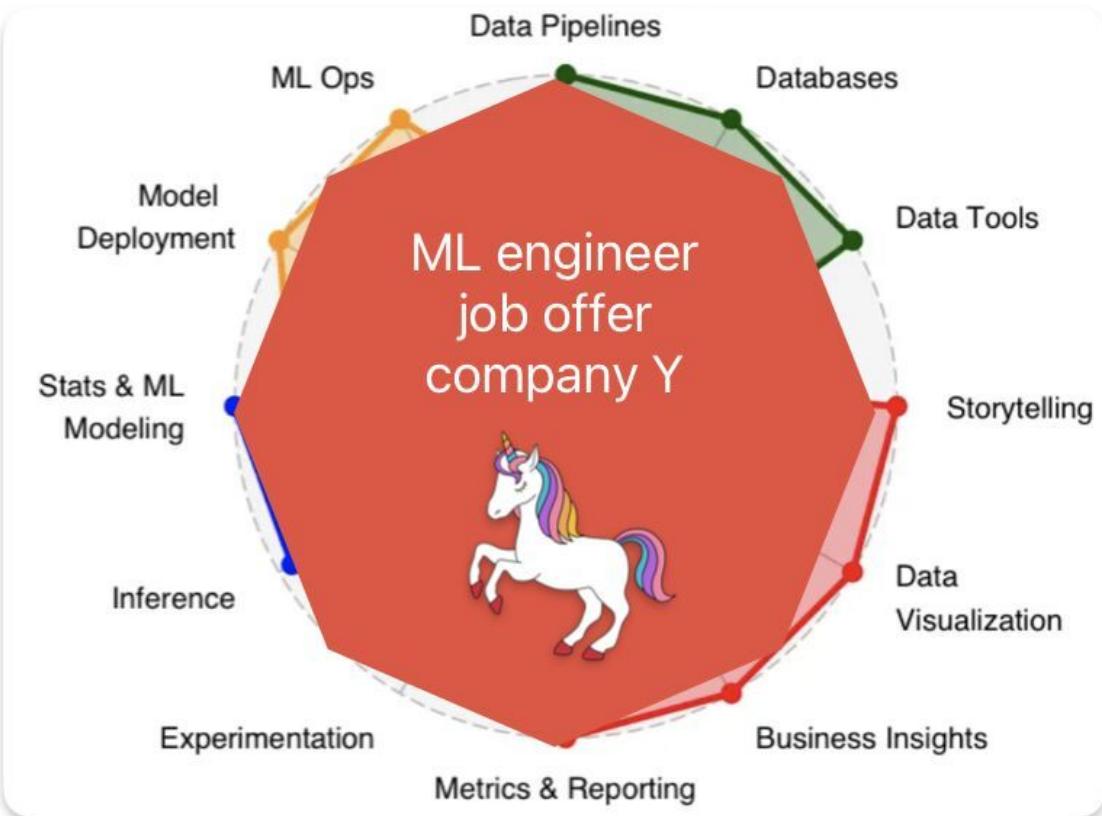
Different set of skills per roles.



In reality it's a bit
blurry...



In reality it's a bit
blurry...



ML Engineering skills are in high demand

Chip Huyen @chipro · Oct 12, 2020
Machine learning engineering is 10% machine learning and 90% engineering.
88 608 7.6K

You Retweeted
Elon Musk @elonmusk
Replies to @chipro
Yeah
11:09 PM · Oct 12, 2020 · Twitter for iPhone
93 Retweets 16 Quote Tweets 5,293 Likes



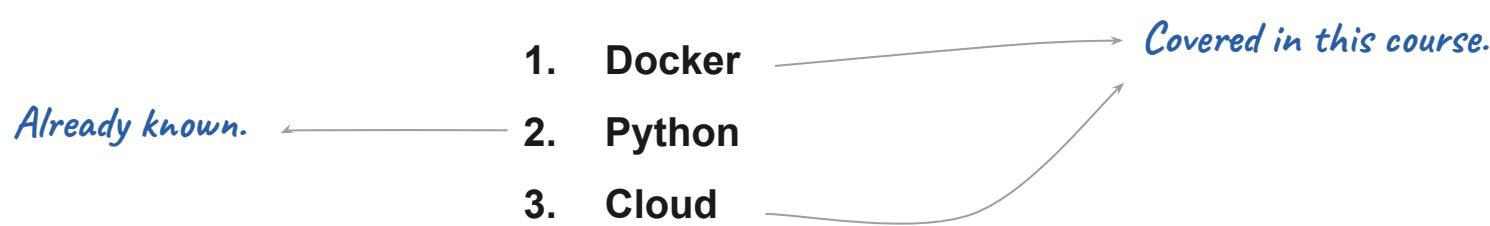
Andrej Karpathy · Following
(Former) Director of AI at Tesla, Op...
1yr • Edited

I am hiring Deep Learning Engineers for the Tesla AI team. Strong software engineering is the primary requirement. Except for the scientist role, deep learning interest or knowledge is only a bonus (we will teach you). For the deep learning scientist role any domain outside of computer vision (e.g. speech, NLP, etc.) works great too.

Study on demanded skills for MLOps engineers.

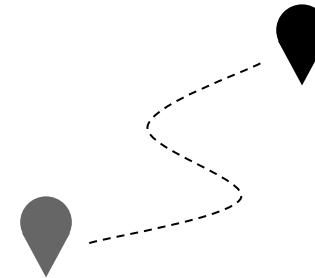
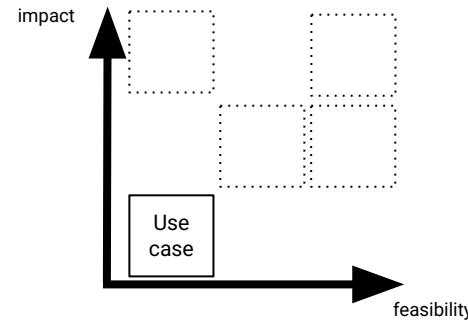
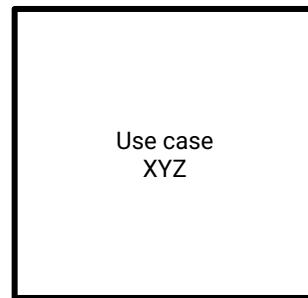
Looking at 310 job offers on MLOps/ML Engineering

Top 3 highest demanded skills:



Project definition framework

A process for ideating new AI project.



1 Identify AI opportunities

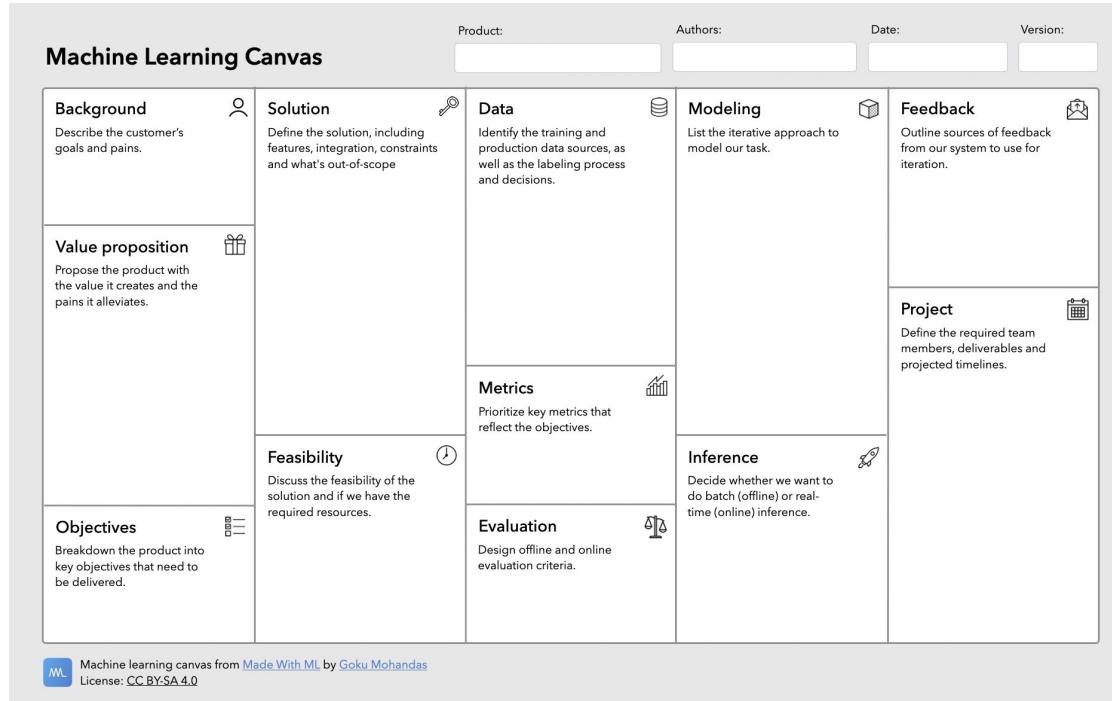
2 Refine selected use cases and their feasibility

3 Prioritize the use cases, select the one(s) to be implemented.

4 Scope the project. Build a technical design and implementation roadmap.

Define and scope your project.

Product design template



Build different stages of your solution

Proof of Concept

Use easily available data to show that your model or solution can work.
Low efforts.
Prove the feasibility and value.
Iterate fast.

Minimum Viable Product

Just enough features for a small set of users to start using it.
Gather feedback and make sure that it is designed in an optimal way.

Productionisation / scaling

Build the infrastructure to finally deploy your solution and let users use it.
Gradual roll-out to more and more users in more and more markets.
Deploy better models, attract more users, go to new markets, maintain the solution, ...

Maintenance

Keep the solution up and running.
Monitor resources and performance.
Update packages and dependencies (software around solution change).
Security and up-time.

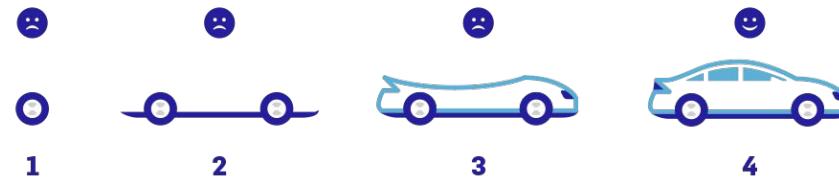
POC	MVP	Productionisation / scaling	Maintenance	...
2 weeks	2 months	6 months	As long as it's up...	

Build different stages of your solution

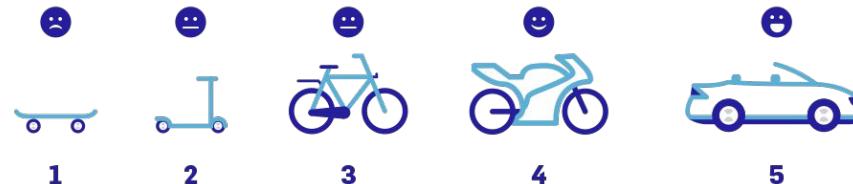


At each stage, your product should be usable

NOT LIKE THIS!



LIKE THIS!



Data science projects are challenging to bring to production

Breaking the myth

Forbes

“87% of data science projects never make it into production...”

VentureBeat

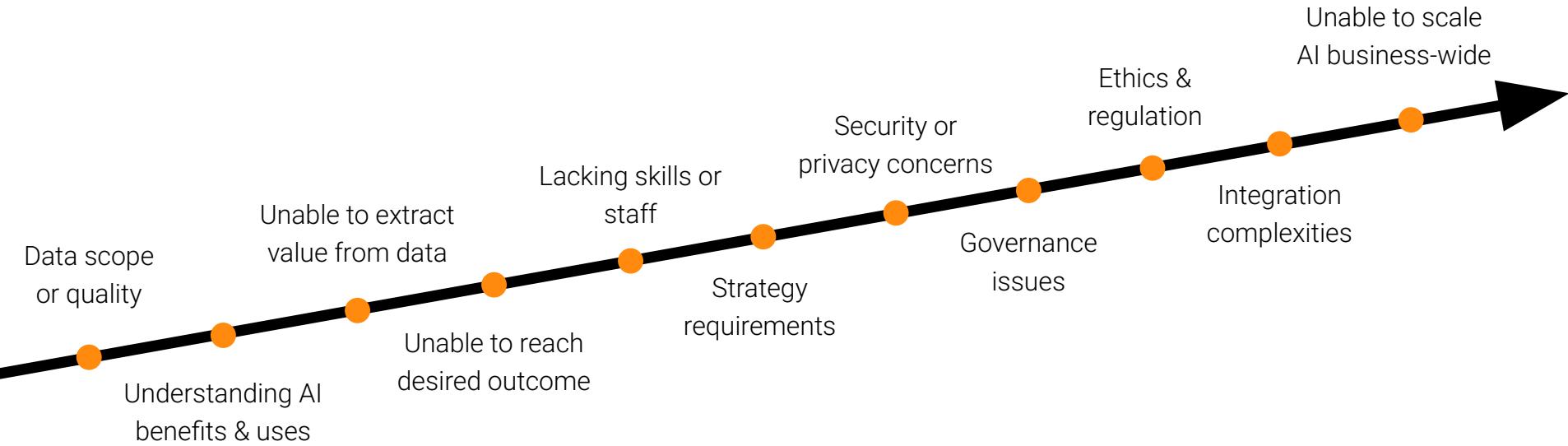


(Might not be a factual number...)

But data science project are still challenging to actually roll-out to the real world!

AI Journey Challenges.

While AI is an enabler for strategic priorities, it doesn't come without its challenges.



When not to use Machine Learning?

It's not always the right solution...

- Clear specifications are available
- Simple heuristics are good enough
- Cost of building and maintaining the ML system outweighs its benefits
- Correctness is of utmost importance
- ML is used only for the hype (e.g., to attract funding)

Examples of these?

(Really) accurate predictions might not even be that important

The over-optimizing paradox

- "Good enough" may be good enough
- Prediction critical for system success or just an gimmick?
- Better predictions may come at excessive costs
 - Data is often the bottleneck
 - Cost of producing more data (labeling, infra, collection, ...)
- Better user interface ("experience") may mitigate many problems
 - Explain decisions to users with Explainable AI (XAI)
- Use only high-confidence predictions?

Critical thinking when doing the project definition

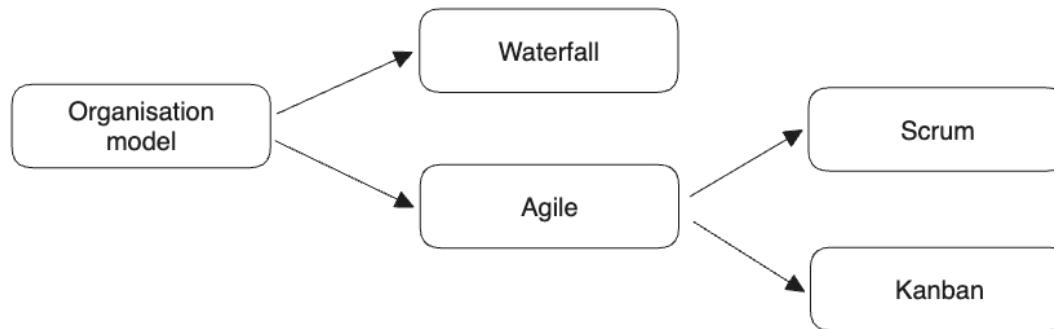
Ask the right questions - make sure you have a solid use case before you start building anything.

- **Baseline:** What is the performance of an alternative to ML? How do simple heuristics or human guess-predictions perform?
- **Probabilistic:** ML is by definition not deterministic. Are probabilities/ranges fine for this use case? E.g. for demand forecasting the model can make errors, for self-driving cars not...
- **Precision / recall:** Are both important? If not, can I make it a success by sacrificing one? E.g. for fraud detection we can raise a warning on false positive, but cannot have false negative...
- **Interpretability:** Do we need to explain why the model makes specific decisions? If yes, can we?
- **Do not reinvent the wheel:** Are there existing open source or 3rd party solutions? Did anybody in my organisation work on something like this?

Working “Agile”



Different models for organising an IT project

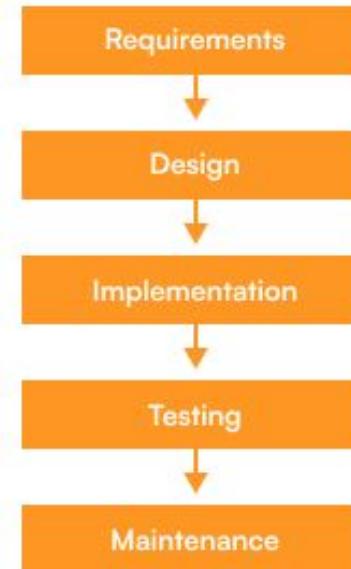


Waterfall model

Models for organising IT projects and teams

Organise a project as a **waterfall**, flowing down **sequentially** through 5 phases. It is **rigid, structured** and **linear** (it is not Agile).

Decision maker team sets a detailed plan for the whole project implementation, with limited flexibility.



Waterfall methodology

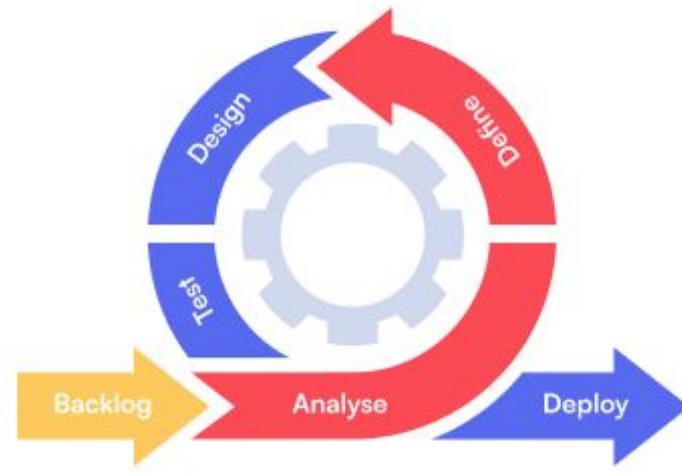
Agile model

Models for organising IT projects and teams

Iterative approach to delivering a project, which focuses on **continuous releases** that incorporate **customer feedback**.

Ability to **adjust** during each iteration promotes **velocity** and **adaptability**.

This approach is different from a linear, waterfall project management approach, which follows a set path with limited deviation.



Agile methodology

Agile: Scrum vs Kanban



Kanban methodology

In Japanese, kanban literally translates to "visual signal."

- Originally comes from *lean manufacturing* in Japan (at Toyota)
- Simple and clean application of Agile
- Matches the amount of Work in Progress (WIP) to the team's capacity - *Just In Time (JIT)*
- Minimal set of rules - easy to pick up for new software engineering teams

Advantages

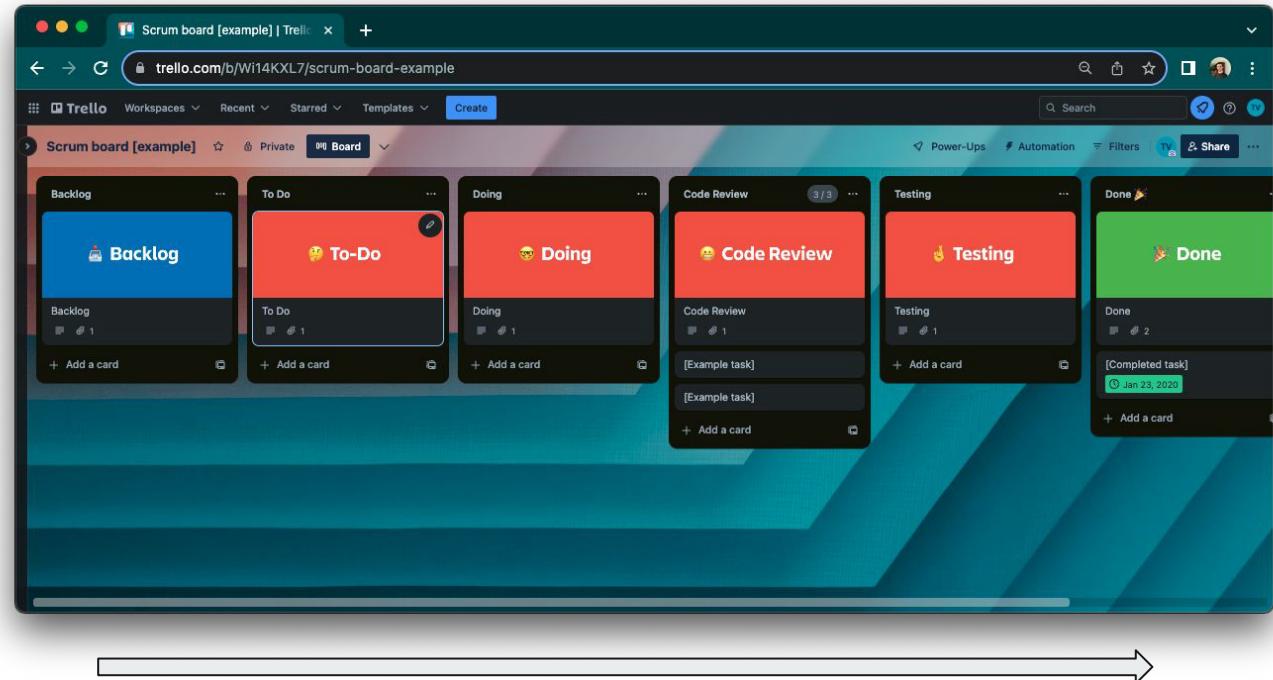
- + Planning flexibility
 - Once done with an item, pick up the next on the backlog
 - Product owner can update the backlog without disrupting the team
- + Short time cycles
 - Time for a unit of work to cycle through the whole process
- + Fewer bottlenecks
- + Visual metrics, transparency

Kanban board

The **kanban board** is filled with **kanban cards** (aka tickets or tasks).

New tasks are added to the backlog.

They are then gradually moved along the board.



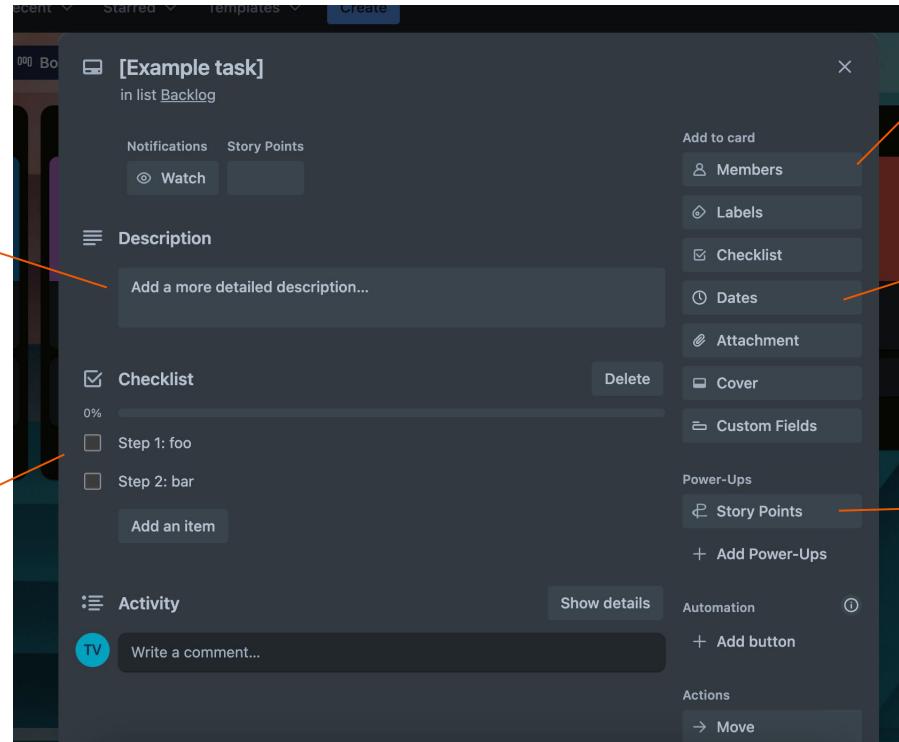
Kanban card

A detailed description of what should be done and how ensures quality of delivery and alignment in the team.

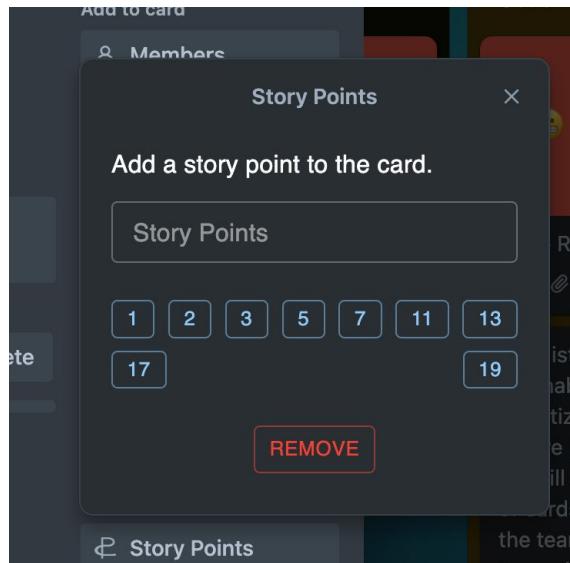
Define task

Include definition of done!

Break up tasks into steps



Kanban card



Story points are a measure of *effort* and *complexity* of a task.

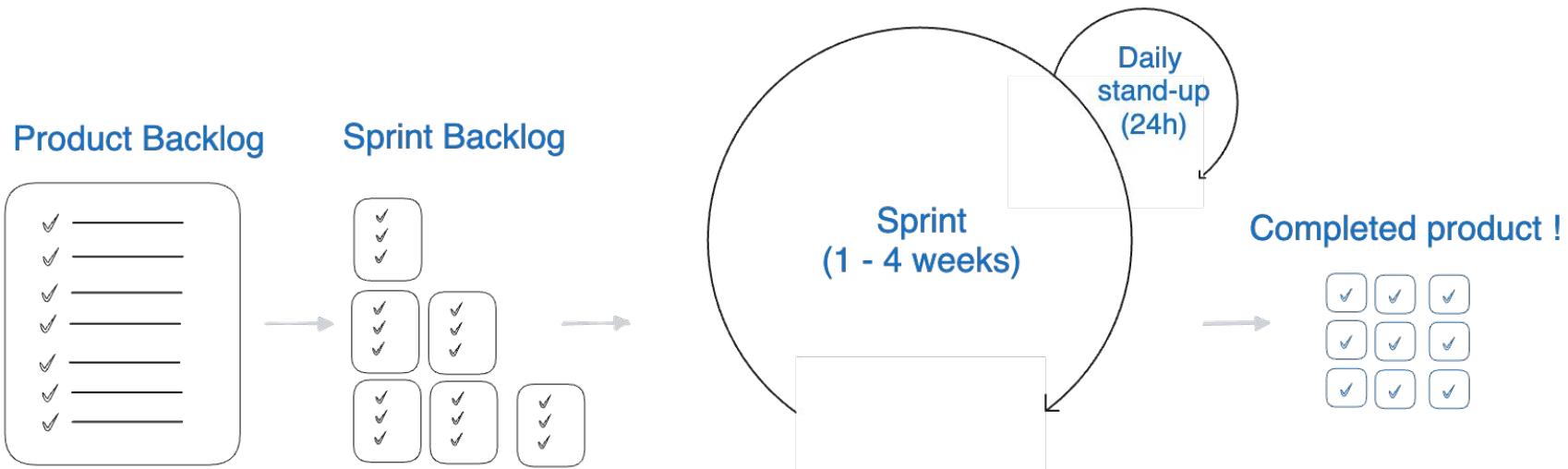
Follows the fibonacci scale.

Sometimes voted on
(e.g. <https://www.pointingpoker.com/>)

If you want more information...



Scrum methodology

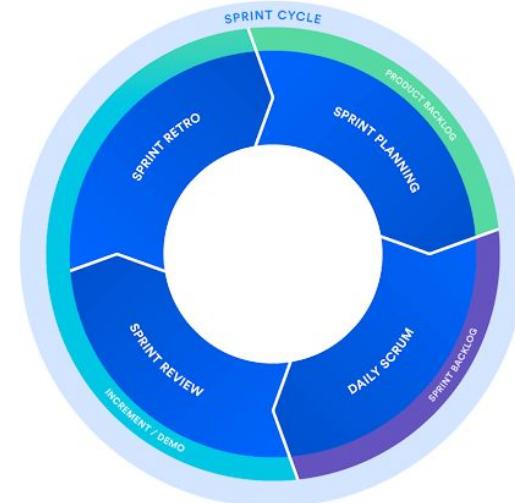


Scrum methodology

Sprint organisation

Each **sprint cycle** will include (1-4 weeks per sprint cycle):

- **Sprint planning:** Before each sprint, plan and add tasks from the [product backlog](#) to the [sprint backlog](#).
- **Refinement session:** The technical team refines each ticket into a concrete implementation plan (approach, dependencies, definition of done, ...).
- **Daily stand-up** (aka daily scrum): During the sprint, have dailies to update the [scrum board](#), similar to kanban board.
Everyone state what they have done the day before, what they will do today and any blockers they might have.
- **Sprint review:** Casually [present](#) the work that was done in the last sprint to the rest of the team (definition, celebration and transparency).
- **Sprint retrospective:** [Feedback](#) on what went well and what can be improved regarding last sprint.



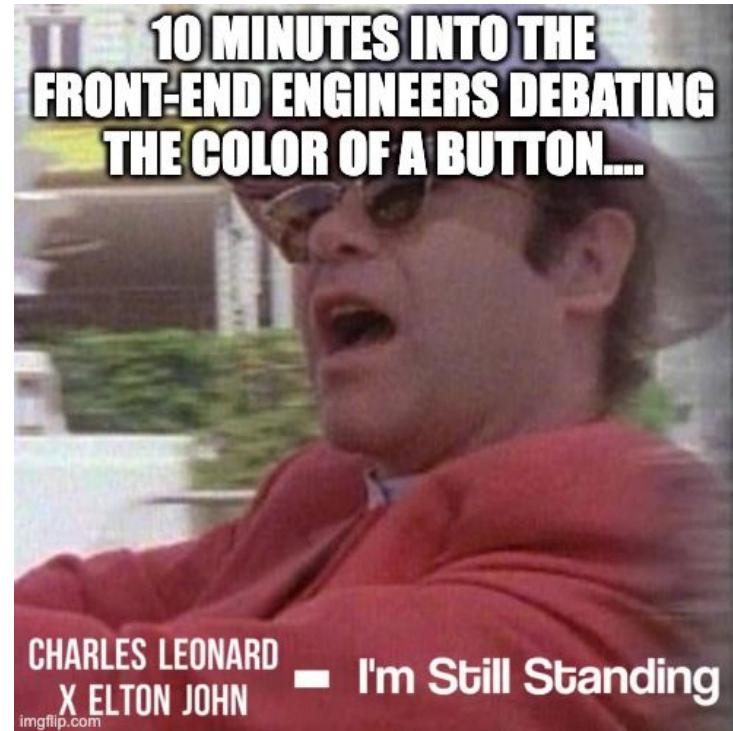
Why are “daily stand-up” meetings called that way?

Why are “daily stand-up” meetings called that way?

Goal is to make the meeting efficient.

All team members have to answer a fixed set of questions such as:

- What did you do yesterday?
- What did you do today?
- What, if anything, is blocking your progress?



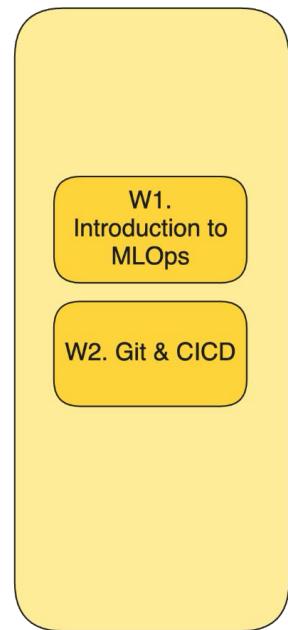
The scrum team

Roles & responsibilities

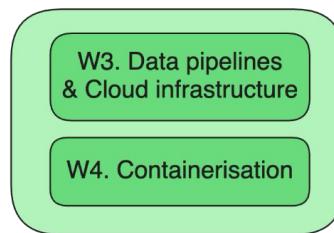
- **Product Owner:** Represents stakeholders' interests. Selects what is relevant to work on considering product objectives. Define user stories, priorities backlog and decide on product's direction.
- **Scrum Master:** Coach for the team. Ensures the best following of the scrum framework. Maintains the scrum board and backlog, facilitates and leads meetings and addresses obstacles.
- **Scrum Development Team:** Cross-functional developer team (data scientists, ML Engineers, data engineers, DevOps, front-end engineers, ...). Delivers a quality product.

This course follows sprints!

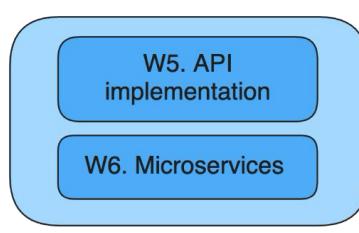
Sprint 1:
Project organisation



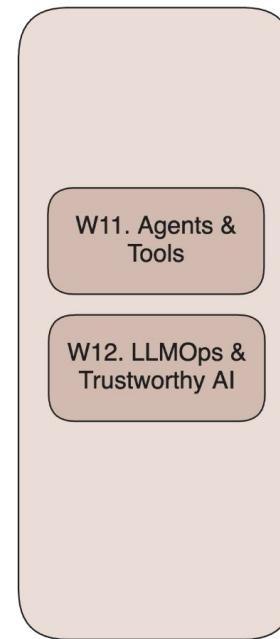
Sprint 2:
Cloud & containerisation



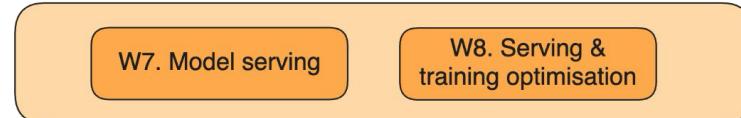
Sprint 3:
API implementation



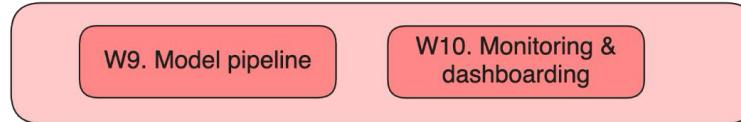
Sprint 6:
LLMOps



Sprint 4: Model serving & optimisation



Sprint 5: Pipeline & monitoring



MLSD Course structure

Objective for this course.

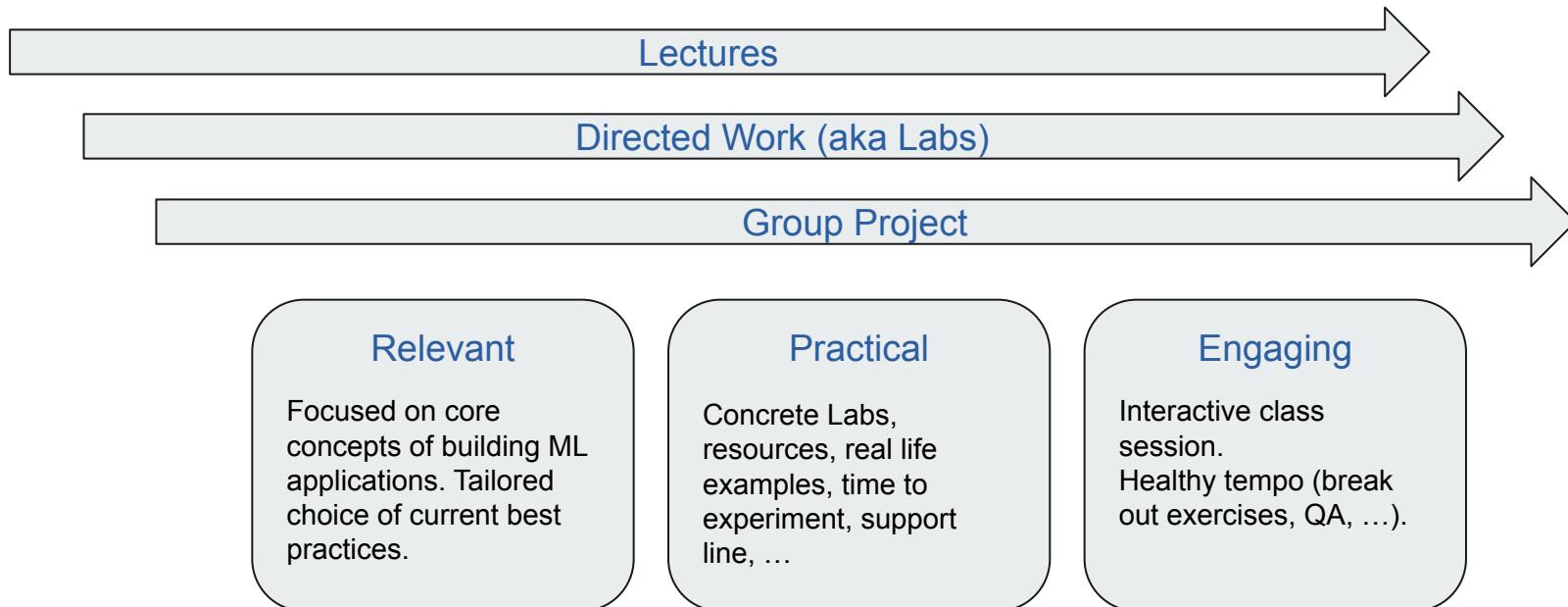
This course will provide you with concepts, tools and practical skill enabling you to **design** and **build** fully operational **ML application** 

The tools covered in this course are selected based on **usability**, **performance**, **popularity** and **accessibility**.

Goal is to provide:

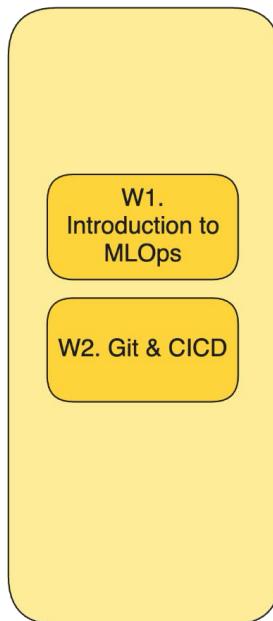
- **Theoretical** concepts
- **Technical** tools & skills
- Practical real world **practices**

Structure of the course

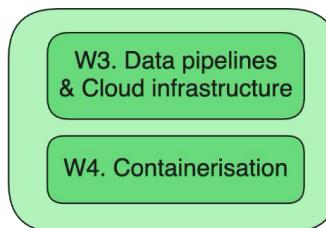


Course outline

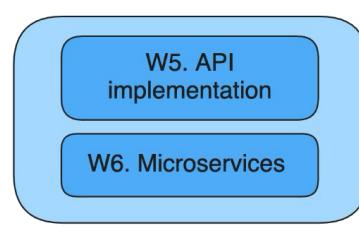
Sprint 1: Project organisation



Sprint 2: Cloud & containerisation



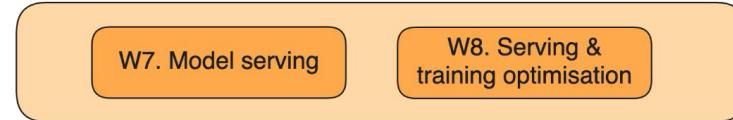
Sprint 3: API implementation



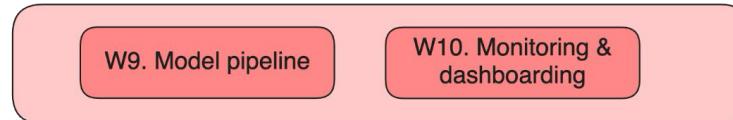
Sprint 6: LLMOps



Sprint 4: Model serving & optimisation



Sprint 5: Pipeline & monitoring



Course material

Everything is on Github!

- Project info
- Lecture & labs (before the class)
- Directed work
- Practice exam (before the exams)

<https://github.com/ThomasVrancken/info9023-mlops>

The screenshot shows a GitHub repository page for 'info9023-mlops'. The repository is public and has 3 branches and 2 tags. The main branch was last updated 2 weeks ago. The repository contains files like 'project', '.gitignore', 'LICENSE', and 'README.md'. The 'README' file is currently selected. The page includes a project card template and a detailed description of the course.

Project Card Template:

- ThomasVrancken Add project card template 648cb5c · 2 weeks ago 224 Commits
- project Add project card template 2 weeks ago
- .gitignore Add local self hosting with vLLM demo 11 months ago
- LICENSE Add license 2 years ago
- README.md Add project card template 2 weeks ago

INFO9023 - Machine Learning Systems Design [spring 2026]

This course equips students with the practical tools and frameworks needed to build production-ready machine learning systems. Covering the complete ML application lifecycle, students will gain hands-on experience with MLOps and LLMOps tools and skills in high demand as industries undergo rapid AI transformation.

Deploying ML to production requires far more than model theoretical knowledge. This course addresses that challenge head-on, providing the technical skills, practices, and tools to bridge the gap between prototype and production. Students will be equipped to make an immediate impact when starting their professional life in the industry.

Practicals & communication

Class practicals

- We meet every Monday from **9:00** to **12:30**
- ~2h30 of lecture/directed work (& break). Remaining of the time can be spent working on your project.

Communication

- Discord: <https://discord.gg/5gucJ9CV>
- Open office hours on **Monday afternoons** (office Number I 77 B in Montefiore)

Type of work & grade

Your final grade is divided in:

- **Exam (40%)**: Oral exam on the topics covered during the lectures
- **Directed work (10%)**: Pass/fail on practical exercises given at the end of the courses.
- **Group project (50%)**: One large group project, see next slides.

Directed work

- You are given an **assignment** to do at the end of the course, in class.
- The teaching staff is there to **support** 🧑‍🤝‍🧑
- You need to upload your results to **Gradescope**
- In most cases **by the end of the day** (we will confirm the deadlines)
- You will get a **pass/fail** grades

Project

Organisation

Build one ML system throughout the course. The application is picked by yourself.

- **Teams:** 2 - 3 students
 - Form group by next week!
 - Let the teaching staff know if you don't have a group and you'll be assigned one
- **Structure**
 - The building blocks to be implemented in the project follow the course's **6 sprints**.
- **You're in the driving seat!**
 - Many building blocks are optional. You are free to choose the overall design and tools used for your project. Experiment and ask questions if you have any.

Project information is also on [Github](#):

Project

Handovers

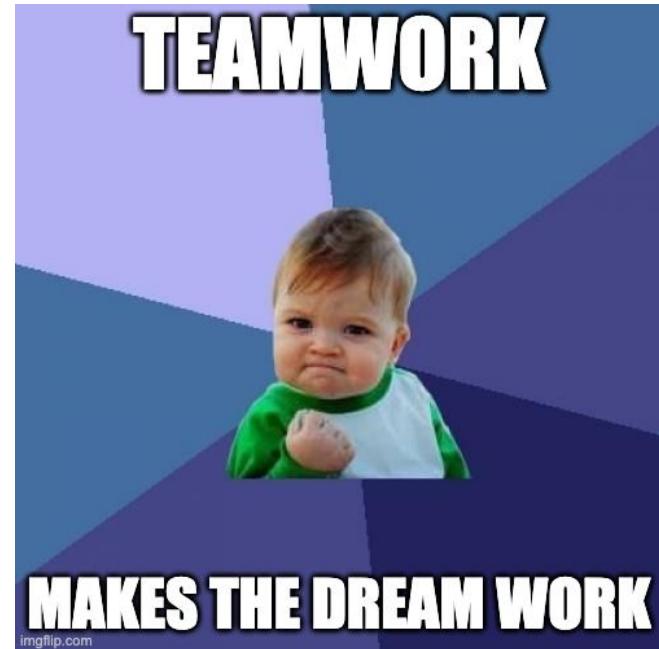
1. There will be **3 milestone meetings** where you can present your results
2. You must **submit the codes** for your implementation **at each milestone**:
 - You do so by creating a Pull Request (PR) from develop → main
 - **⚠ You must follow Gitflow principles: Work on *feature* branches based on *develop*. Once per “release” (aka Milestone), you merge *develop* into *main*.**
 - Send a link to that PR by email to the teaching staff **before the milestone meeting!** We will not accept late PRs
 - Make sure to document your codes:
 - Include Markdown files explaining key components in the `docs/` directory (e.g. docs/DEPLOYMENT.md)
 - Write a clear description of your PR when creating it
 - Write readable, structured and well documented code → Points are deducted if we cannot easily verify how something works
 - We will ask you questions about your codebase during the milestone meeting. If you cannot answer detailed question we will assume it was written by an AI and will deduct some points.

Project

Grading & collaboration

We want to foster **teamwork**!

To incentivise every team member to contribute,
we will give **individual grades**.



Project

Grading & collaboration

$$\text{Individual_grade} = \text{project_grade} * \text{individual_contribution}$$

$$\text{Individual_grade} \in [0, 20]$$

$$\text{project_grade} \in [0, 20]$$

$$\text{Individual_contribution} \in [0, 1]$$

Project

Grading & collaboration

$$\text{Individual_grade} = \boxed{\text{project_grade}} * \text{individual_contribution}$$

$\text{Individual_grade} \in [0, 20]$

$\text{project_grade} \in [0, 20]$

$\text{Individual_contribution} \in [0, 1]$

Project grade will take into account

- The implementation and maturity of each of the work packages (see grid of “work packages” per sprint)
- The quality of your codes and documentatin
- Your presentations

Project

Grading & collaboration

$$\text{Individual_grade} = \text{project_grade} * \text{individual_contribution}$$

$\text{Individual_grade} \in [0, 20]$

$\text{project_grade} \in [0, 20]$

$\text{Individual_contribution} \in [0, 1]$

Individual contribution is calculated based on

- Participation in milestone **presentations** (make sure to spread talking time to represent what you each did)
- Ability to answer **questions**
- Contribution to **code implementation** in github
- A question **form** sent to the team at key moment(s)

Of course, for a healthy project with an even work distribution, all students can get an individual contribution of 1

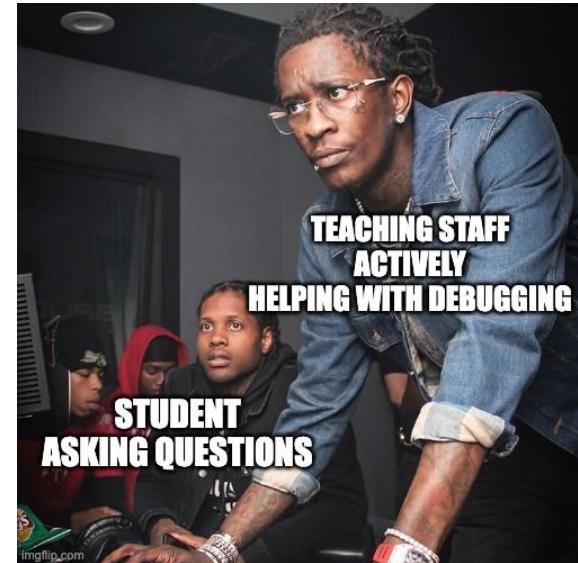
Project

Support

- You will learn new tools!
- At first, some of those tools might be less straightforward to debug
 - Running things in the cloud means that your logs are not directly on your computer
 - There will be new dimensions such as access rights, connections, ...

That is why...

- The teaching staff is there to support!
- We will stay after each lesson to help with any issue, question, bug... you might have!
- **Make sure to attend the class and actively ask questions about the direction of your project, choices you make and any issue/bug you might have!**
 - If you implement everything at the last minute there will be less opportunities to ask us questions



Project

Guiding principles

- Learn, learn and learn!
 - Find an interesting project to work on → with a real world usage
 - Come up with your own design and toolstack
 - Focus on relevant parts of your specific system
 - Motivate your choices
-
- ... And pick a cool name for it ✨🚀



Project

Examples from last year

- Hessian: <https://github.com/alexandre-eymael/HESSIAN>
- ClipMorph: <https://github.com/iSach/clipmorph>
- Triple-P: <https://github.com/lambi702/MLOps-TripleP>
- ...



We strive at improving this course!

Help us help you 🤝

- Quick feedback cycles
- Open communication
- Enthusiasm for trying new things 🖌
- Active support from teaching staff
- Students in the classroom



imgflip.com

Project objective for sprint 1

Project guidelines on [github!](#)

#	Week	Work package
1.1	W01	Pick a team <ul style="list-style-type: none">• Try to mix skills and experience• If you didn't find one let one of the teachers know and we'll allocate you to one
1.2	W01	Select a use case Source options <ul style="list-style-type: none">• Previous course• Kaggle Datasets• ... <p>Make sure to pick a use case where data is available. A requirement for this project is to train and deploy your own ML model. It is not an option to work with LLMs, as there is too much overhead in training and serving those. Note that the ML modeling itself won't be a big part of the course. You can pick data from one of your previous projects if it is available. Ideally pick something with interesting data and a real world application.</p>
1.3	W01	Find a cool name for your project ✨
1.4	W01	Fill in the project card template (docx version is in this course's github repository, under "project/project_card_template.docx") and send it via email to the teaching staff. The teaching staff needs to approve and provide feedback on your project - if the project is deemed not feasible then we will ask you to come up with a new one.

Deliverable: Submit all of your projects by sending in a filled in project card template by email by Thursday 5th of February at 23:59 !

Project deliverable for next week!

Submit your **team** and **project** by filling in the **project card template** and sending it to the teaching staff.

- The template is on github:
https://github.com/ThomasVrancken/info9023-mlops/blob/main/project/project_card_template.docx
- You can download the docx file, fill it in and send it back by email to the teaching staff:
 - t.vrancken@uliege.be
 - Matthias.Pirlet@uliege.be

The teaching staff needs to approve and provide feedback on your project - if the project is deemed not feasible then we will ask you to come up with a new one.

 **Send it by email by to the
teaching staff by Thursday 5th of
February at 23:59**

[Project name]
INFO9023 - Project card
Purpose of this document is to briefly explain what your project will be about so the teaching staff can validate it and provide feedback. It is not part of grading, no need to spend too much time editing it (just a couple of sentences per section).
Make a copy and send it to t.vrancken@uliege.be and Matthias.Pirlet@uliege.be by the 29/02/2024.
Project description
<i>[Short description of your project]</i>
Project data
<i>[Short description of the data you'll use]</i>

That's it for today!

