

Monitoring & dashboarding

Sprint 5 - Week 10

INFO 9023 - Machine Learning Systems Design

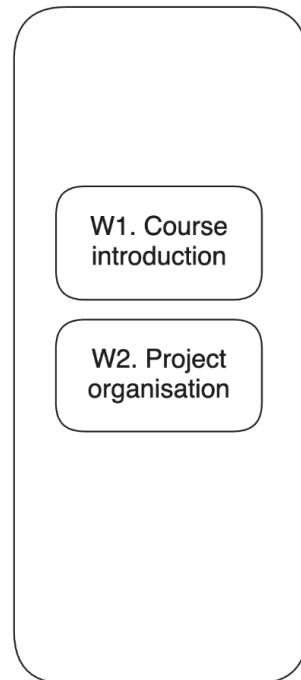
Thomas Vrancken (t.vrancken@uliege.be)

Matthias Pirlet (matthias.pirlet@uliege.be)

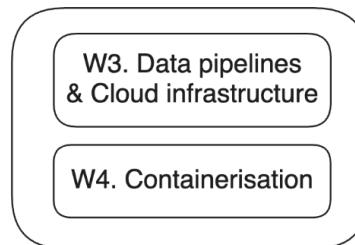
2025 Spring

Status on our overall course roadmap

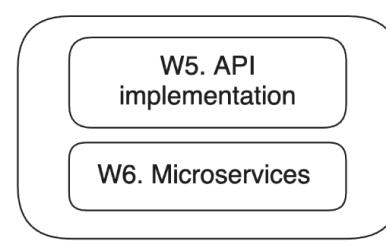
Sprint 1:
Project organisation



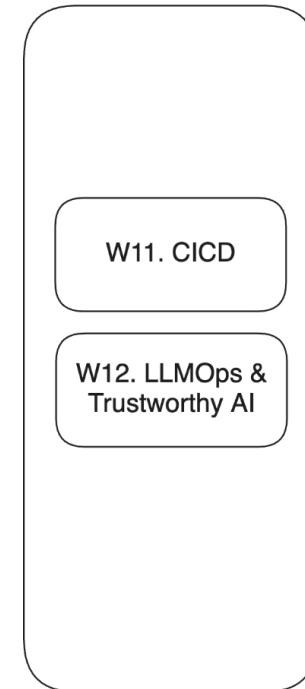
Sprint 2:
Cloud & containerisation



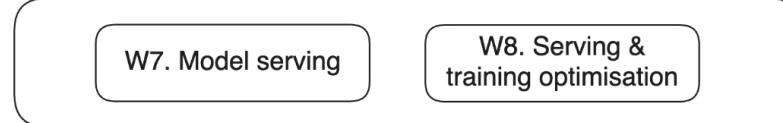
Sprint 3:
API implementation



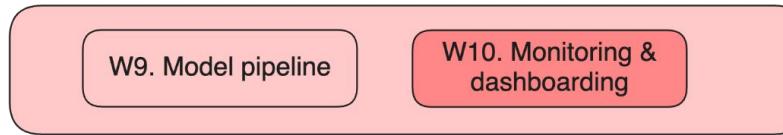
Sprint 6:
CICD



Sprint 4: Model serving & optimisation



Sprint 5: Pipeline & monitoring



Agenda

What will we talk about today

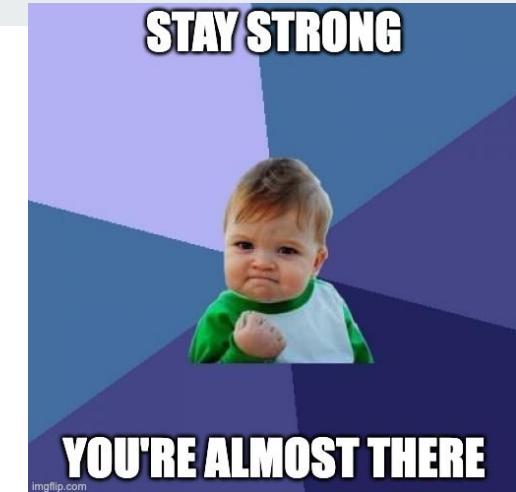
Lecture

- Monitoring
 - Resource monitoring
 - Performance monitoring
 - Drift types: data, target and concept
- Dashboarding

Lab

- Streamlit

STAY STRONG



Exam info

Exam

- Oral exam: Answer **questions** based on a **use case**
- 30 min preparation + 20 min oral exam

Practical

- Date: 2nd of June
- We'll send a link for you to book a specific slot to pass it that day
- We'll put another practice exam on Github
- Exam is 20% of the final grade. Total grade is split between
 - 60% group project (individual grade which takes into account personal participation)
 - 20% directed works
 - **20% exam**

Tip

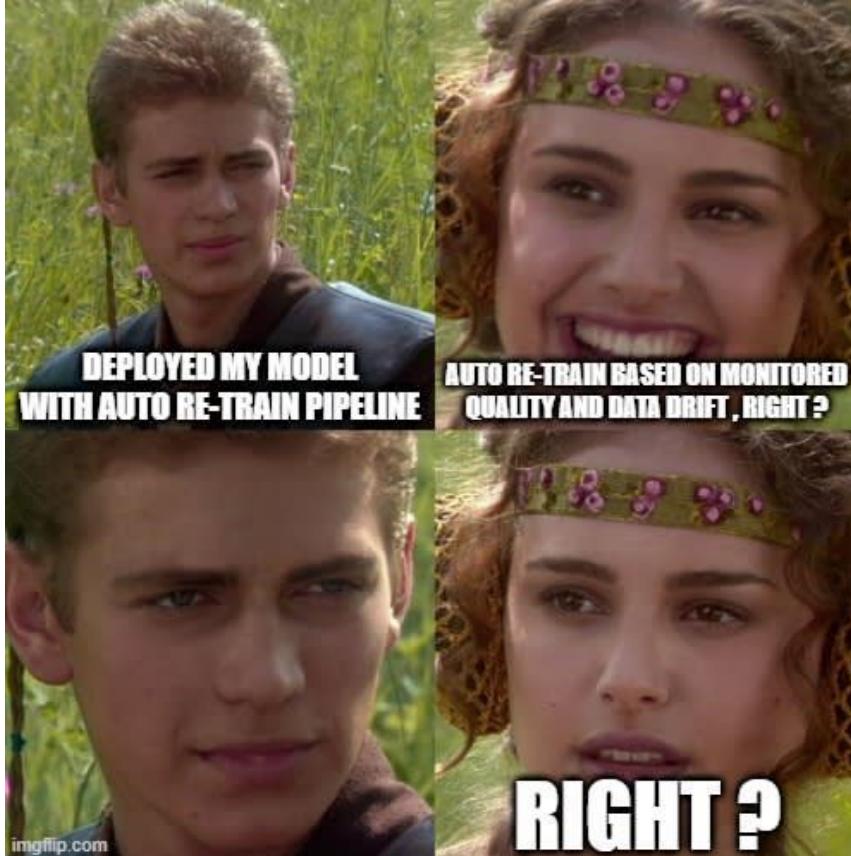
- Carefully read the exam's use case description. Some *preferences* and *factors* will heavily influence your decision
- Make sure to know key concepts of the course and use them in your answers
- *Motivate* why you take certain design choices

MS 3 | The final presentation



Milestone 3 presentation will be different: You will present your results in front of the whole class!

- On the 12th of May: No lecture, only team presentation of MS 3 in front of the whole class
- Each group will present their whole project
 - Focus on key parts of the application. Use your time (and the audience's time) wisely.
- Bring a **demo!** The audience (and the teaching staff) should be able to interact with your system through a publicly available dashboard (or APIs)
- Split talking time and questions per person
- **⚠ Beware of time: 15min to present + 5min QA**



imgflip.com

A word on ML6...

Our experience & expertise make us leading AI specialists.

EXCEPTIONAL TALENT & SKILLS



110+ experts spread over 3 different EU locations.



Known for technical expertise
Loved for our business results



Talent magnet: 16 applications each day

STATE-OF-THE-ART TECH KNOWLEDGE & ASSETS



6 Mio downloads per month of our open source packages

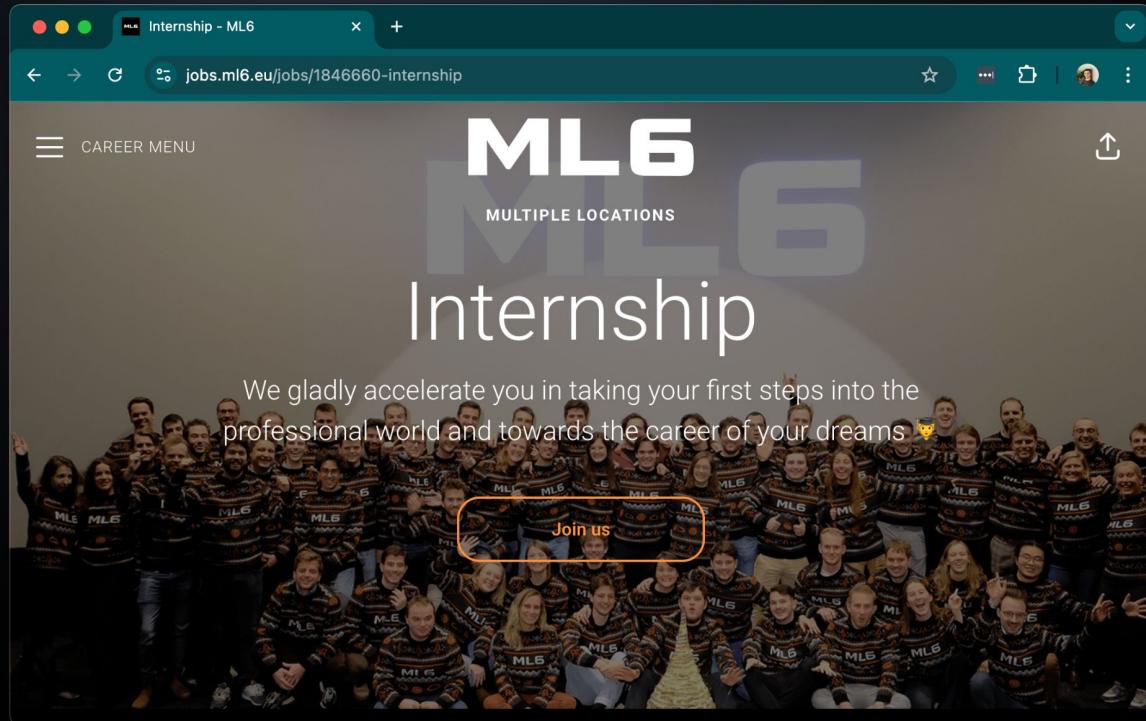


150+ clients, 300+ projects, 3 spin-offs
17% of time in R&D, 250+ publications,
15 awards, avg time to value 11.5 weeks



Security, Legal and Ethical AI experts

We offer internships!

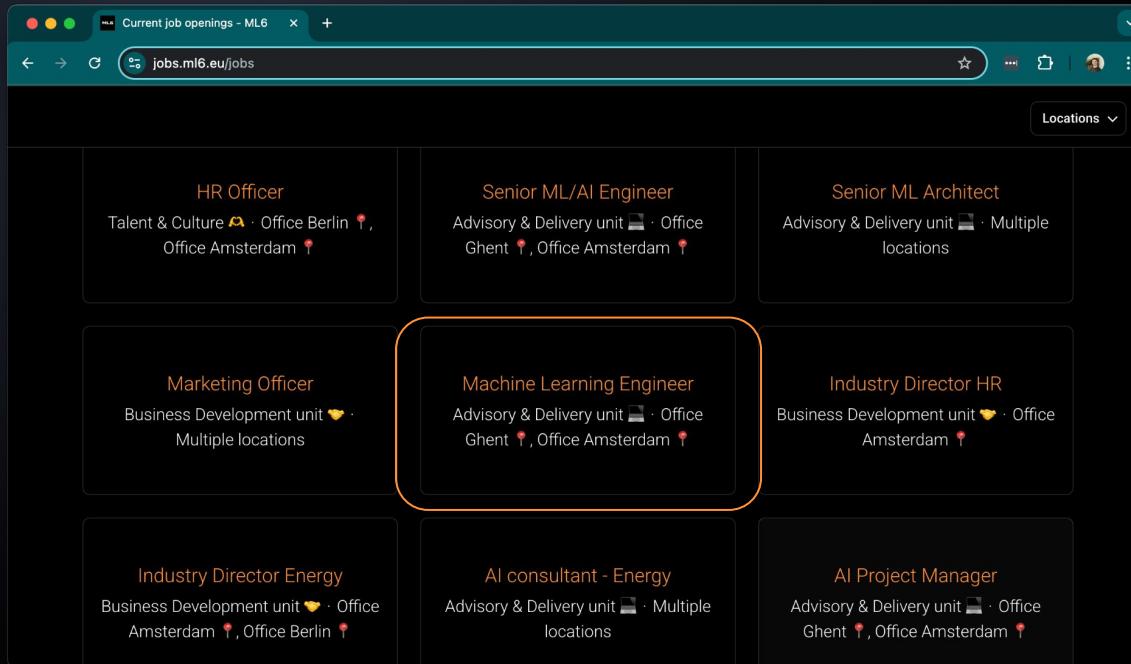


The screenshot shows a web browser window for the ML6 internship application. The title bar says "Internship - ML6". The URL in the address bar is "jobs.ml6.eu/jobs/1846660-internship". The main content features the ML6 logo with "MULTIPLE LOCATIONS" below it. A large image of a group of people wearing ML6 jumpers is displayed, with a call-to-action button labeled "Join us" overlaid. To the right of the image, there is a vertical "CAREER MENU" icon. Below the image, a text block reads: "We gladly accelerate you in taking your first steps into the professional world and towards the career of your dreams." At the bottom of the page, the URL "https://jobs.ml6.eu/jobs/1846660-internship" is shown.

Topics:

- Information Management 
- GenAI NLP 
- GenAI Vision 
- Digital Twins - Industry 4.0 
- AI in Bio 
- Energy 
- Ops 
- Hybrid Solutions 
- Business Development 
- Ethics & AI 

And we have some open positions...



The screenshot shows a web browser window titled "Current job openings - ML6" with the URL "jobs.ml6.eu/jobs". The page displays several job listings in a grid format:

- HR Officer**
Talent & Culture 🏢 · Office Berlin ⓘ, Office Amsterdam ⓘ
- Senior ML/AI Engineer**
Advisory & Delivery unit 🏢 · Office Ghent ⓘ, Office Amsterdam ⓘ
- Senior ML Architect**
Advisory & Delivery unit 🏢 · Multiple locations
- Marketing Officer**
Business Development unit 🎨 · Multiple locations
- Machine Learning Engineer**
Advisory & Delivery unit 🏢 · Office Ghent ⓘ, Office Amsterdam ⓘ
- Industry Director HR**
Business Development unit 🎨 · Office Amsterdam ⓘ
- Industry Director Energy**
Business Development unit 🎨 · Office Amsterdam ⓘ, Office Berlin ⓘ
- AI consultant - Energy**
Advisory & Delivery unit 🏢 · Multiple locations
- AI Project Manager**
Advisory & Delivery unit 🏢 · Office Ghent ⓘ, Office Amsterdam ⓘ

<https://jobs.ml6.eu/jobs>

And we have some open positions...



Badei Alrahal
Ex-MLSD student
ML Engineer @ ML6

"To the future data scientists of ULG,

As soon as I completed my master's in data science at ULG, I was looking for an environment where intellectual challenge meets real-world impact. That's exactly what I found at ML6: a passionate team, innovative projects, and a setting that fosters personal growth.

In short, ML6 is truly a very "cool" company, and I strongly encourage you to consider it among your top choices for a future job or internship. I wish you all the best in your journey!"

Monitoring

Logging vs Monitoring

Logging	Monitoring
<ul style="list-style-type: none">• Qualitative (and typically extensive) information about current application state, events, errors, etc.• (typically) Time based	<ul style="list-style-type: none">• Quantitative<ul style="list-style-type: none">◦ Captures the rate of change of numerical data• Aggregated<ul style="list-style-type: none">◦ We look at something that happens consistently over time

Why do we need monitoring?

Alerting

- to know when something goes wrong (failures or problems such as application running slow)

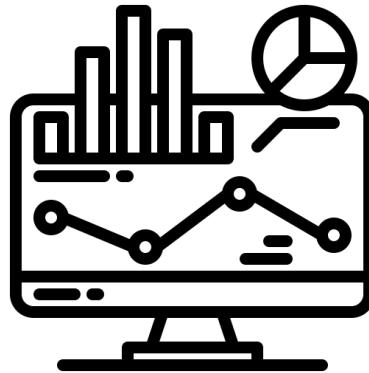
Debugging

- to have system information to reason about (inspecting code might not be enough)

Trending

- to know how your system is being used (for capacity planning and design decisions)

Model monitoring perspectives



Resource level

- Monitor the usage of infrastructure and resources
- Ensuring the model is running correctly
- Key questions:
 - Is the system alive? 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

- Monitor the performance/accuracy of the model over time
- Key questions
 - Is the model still doing accurate predictions with the new data coming in?
 - Is it performing as good as during the development phase?

Resource Level Monitoring

The four golden signals of resource monitoring

Latency

The time it takes to service a request.

Average, max, N-percentiles, ...

Traffic

A measure of how much demand is being placed on your system, measured in a high-level system-specific metric.

Errors

The rate of requests that fail

Explicitly (e.g., HTTP 500s).

or

Implicitly (wrong content)

or

Policy (own rule for failure - e.g. more than 1 sec for response).

Saturation

How "full" your service is. A measure of your system fraction, emphasizing the resources that are most constrained.

Includes auto scaling and load balancing metrics.

Resource level monitoring

Most Cloud products have a monitoring dashboard.

For example, Cloud Run allows you to monitor various metrics.

Metric	Services	Jobs
Billable container instance time	✓	✓
Container startup latency	✓	✓
Container CPU utilisation	✓	✓
Container memory utilisation	✓	✓
Sent bytes	✓	✓
Received bytes	✓	✓
Request count	✓	
Request latencies	✓	
Container instance count	✓	
Maximum concurrent requests	✓	
Completed executions		✓
Running executions		✓
Completed task attempts		✓
Running task attempts		✓

Resource level monitoring: (Better) Uptime

- Checking a service **availability**
- An HTTP request is sent every X seconds to the system and Uptime and checks if the return *status code is a success*
 - I.e. code: ???
- Uptime can be sent to send alerts (notification, email...) to an on-call developer in case the system is down

The screenshot shows the Better Uptime web interface. At the top right, there's a user profile for "Richard Hendricks". The main header says "Have a great day, Richard!". On the left, a sidebar menu includes "Monitors" (selected), "Heartbeats", "Who's on call?", "Incidents", "Team members", "Status pages", "Escalation policies", and "Integrations". Below the sidebar are sections for "Stacker", "Other", and "DNS".

- Stacker:**
 - piedpiper.com (Up · 200d 18h 41m)
 - developers.piedpiper.com (Up · 60d 09h 39m)
 - piedpiper.com/compression (Up · 35d 04h 22m)
 - hooli.com (Paused · 200d 18h 41m)
- Other:**
 - API (Up · 23d 01h 12m)
 - ping 192.0.0.0/24 (Down · 1h 09m) - marked as an "Ongoing incident"
- DNS:**
 - 8.8.8.8 (Up · 165d 12h 35m)
 - CloudFlare DNS (Up · 200d 18h 41m)

Resource level monitoring: (Better) Uptime

- Checking a service **availability**
- An HTTP request is sent every X seconds to the system and Uptime and checks if the return *status code is a success*
 - I.e. code: 2XX
- Uptime can be sent to send alerts (notification, email...) to an on-call developer in case the system is down

The screenshot shows the Better Uptime web interface. At the top right, there's a user profile for "Richard Hendricks". The main header says "Have a great day, Richard!". On the left, a sidebar menu includes "Monitors" (selected), "Heartbeats", "Who's on call?", "Incidents", "Team members", "Status pages", "Escalation policies", "Integrations", "Billing", "Help & Support", "Light mode", and "Team PiedPiper". The main content area is divided into sections: "Stacker" (listing "piedpiper.com" as up for 200 days 18 hours 41 minutes, "developers.piedpiper.com" as up for 60 days 09 hours 39 minutes, "piedpiper.com/compression" as up for 35 days 04 hours 22 minutes, and "hooli.com" as paused for 200 days 18 hours 41 minutes), "Other" (listing "API" as up for 23 days 01 hours 12 minutes and "ping 192.0.0.0/24" as down for 1 hour 09 minutes with an "Ongoing incident"), and "DNS" (listing "8.8.8.8" as up for 165 days 12 hours 35 minutes and "CloudFlare DNS" as up for 200 days 18 hours 41 minutes). A search bar at the top right says "Search monitors" and a blue button says "Create monitor".

Resource level monitoring: (Better) Uptime

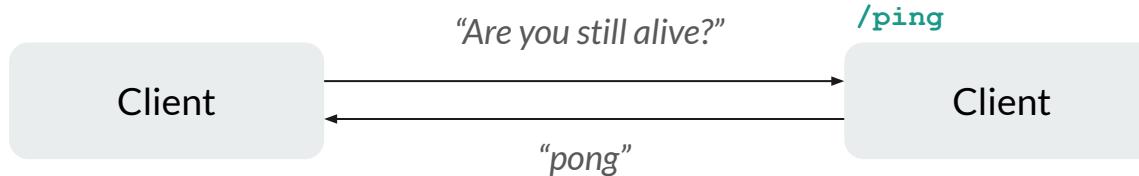
Steps:

- Step 1: Choosing the URL to monitor
- Step 2: Choosing the alerting options
 - E-mail, Slack, phone call, SMS,
 - ...
- Step 3: Choosing the escalation options
- Step 4: Creating a monitor

(This is a Pull monitoring example - more options on Uptime)

The screenshot shows the Better Uptime web interface. At the top right, there's a user profile for "Richard Hendricks". Below the header, a search bar and a "Create monitor" button are visible. The main area is divided into sections: "Stacket" (containing monitors for piedpiper.com, developers.piedpiper.com, piedpiper.com/compression, and hooli.com), "Other" (containing API and ping 192.0.0.0/24), and "DNS" (containing 8.8.8.8 and CloudFlare DNS). Each monitor entry includes status (green for up, orange for paused, red for down), uptime, and a "Details" button.

Side note: 🎾 Ping Pong Protocol: The API Health Check



Endpoint: `/ping`

- A minimalist handshake
- Returns a basic response (often just "pong")
- Lets you know the server is up and listening
- Like knocking on a door to see if someone's home 🖐

Resource testing: Locust

Before releasing & monitoring your system, you should (load) test it.

Locust

- **Open source load testing tool** for HTTP and other protocols
- Enables developers to define user behavior and simulate millions of concurrent users to test the performance of web applications and services
- Developer friendly approach → define your tests in regular Python code
- Locust tests can be run from **command line** or using its **web-based UI**
- Monitor:
 - Throughput
 - Latency
 - Errors
- Real time monitoring and/or exported for later analysis.

Resource testing: Locust

```
from locust import HttpUser, task, between

class ImageClassificationUser(HttpUser):
    wait_time = between(1, 3)

    @task
    def classify_image(self):
        payload = {"img_url": "https://example.com/sample_image.jpg"}
        headers = {'Content-Type': 'application/json'}
        self.client.post("/predict", json=payload, headers=headers)
```

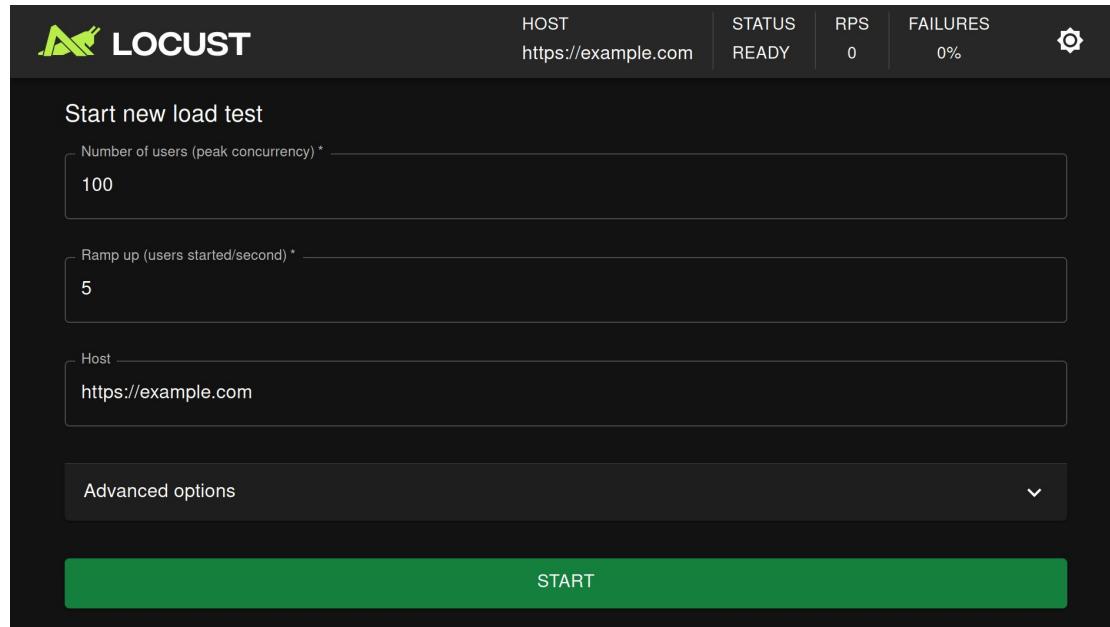
Defines a `user` behavior where the `/predict` endpoint is repeatedly called with a sample image URL.

Each user will wait randomly 1 to 3 seconds between requests.

URL is defined by the `--host` parameter in the `locust start` command (your model serving needs to run on that URL)

```
locust -f locustfile.py --host=http://127.0.0.1:8000
```

Resource testing: Locust



The screenshot shows the Locust UI interface for starting a new load test. At the top, it displays the host URL as `https://example.com`, status as `READY`, RPS as `0`, and failures as `0%`. Below this, there's a configuration section with fields for the number of users (peak concurrency), ramp up rate, and host URL. A large green "START" button is at the bottom.

Setting	Value
Number of users (peak concurrency) *	100
Ramp up (users started/second) *	5
Host	<code>https://example.com</code>

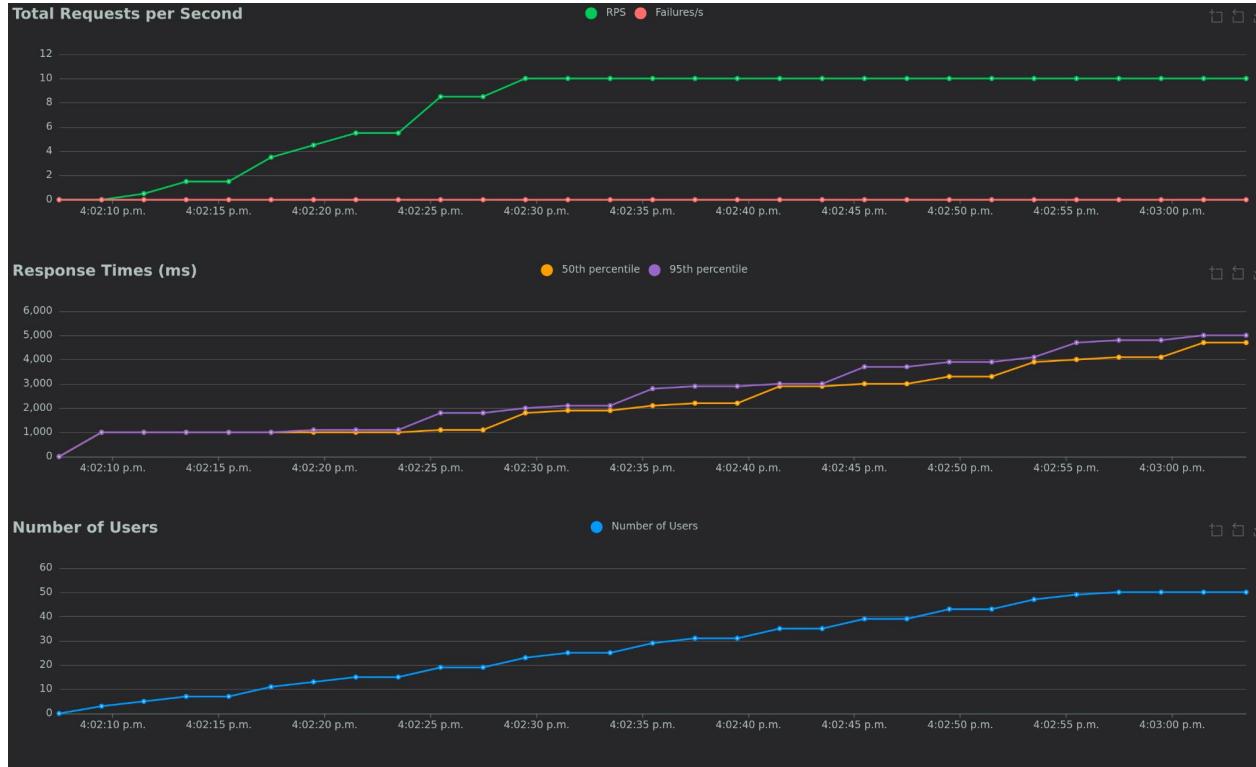
START

Navigate to the locust UI at
<http://127.0.0.1:8089/>

Select the maximum number of users (e.g. 100) and spawn rate (e.g. 5 users per second)

Check your results...

Resource testing: Locust



Resource testing: Locust

 **LOCUST**

HOST
http://0.0.0.0:8089

STATUS
STOPPED

RPS
38.5

FAILURES
100%

NEW **RESET** 

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS 

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/hello	445	445	2	3	4	2.09	1	15	207	19.2	19.2
GET	/world	444	444	42	43	43	41.64	41	43	207	19.3	19.3
	Aggregated	889	889	15	42	43	21.84	1	43	207	38.5	38.5

Resource testing: Locust

LOCUST		HOST http://0.0.0.0:8089	STATUS RUNNING	USERS 300	WORKERS 10	RPS 390.5	FAILURES 100%	EDIT	STOP	RESET	⚙️
STATISTICS	CHARTS	FAILURES	EXCEPTIONS	CURRENT RATIO	DOWNLOAD DATA	LOGS	WORKERS				
Worker											
Worker				State	# users	CPU usage		Memory usage			
0ee7e39e5a524b6f9a69283186b1ecbc				running	30	4.4		53.91 MB			
1ab8aa75f0ee49029b72fc7f555bfe3c				running	30	4.2		53.45 MB			
26147ee1d88a46ab8e9f629cedd8803f				running	30	4.4		53.79 MB			
4364389b7cccd44fa84671540b8531d76				running	30	4.6		53.78 MB			
43fd22591ef641adb4e9cc051b94e45d				running	30	4.6		53.79 MB			
49f08a6f8ae34e3eabf656b3e5edf19d				running	30	4.2		53.54 MB			
68525e334ed345ea8245e9a73dc4cc82				running	30	4.4		53.53 MB			
77cb6fc6f8dc46208f841019ba5a7173				running	30	4.6		53.28 MB			
78a81c72e7524b0391794ac93705a5aa				running	30	4.6		53.29 MB			
a4dc45692f9347d0a6127719766d8784				running	30	4.2		53.75 MB			

Resource level monitoring: Common metric types

Counter

Value can only be increased or reset to zero.

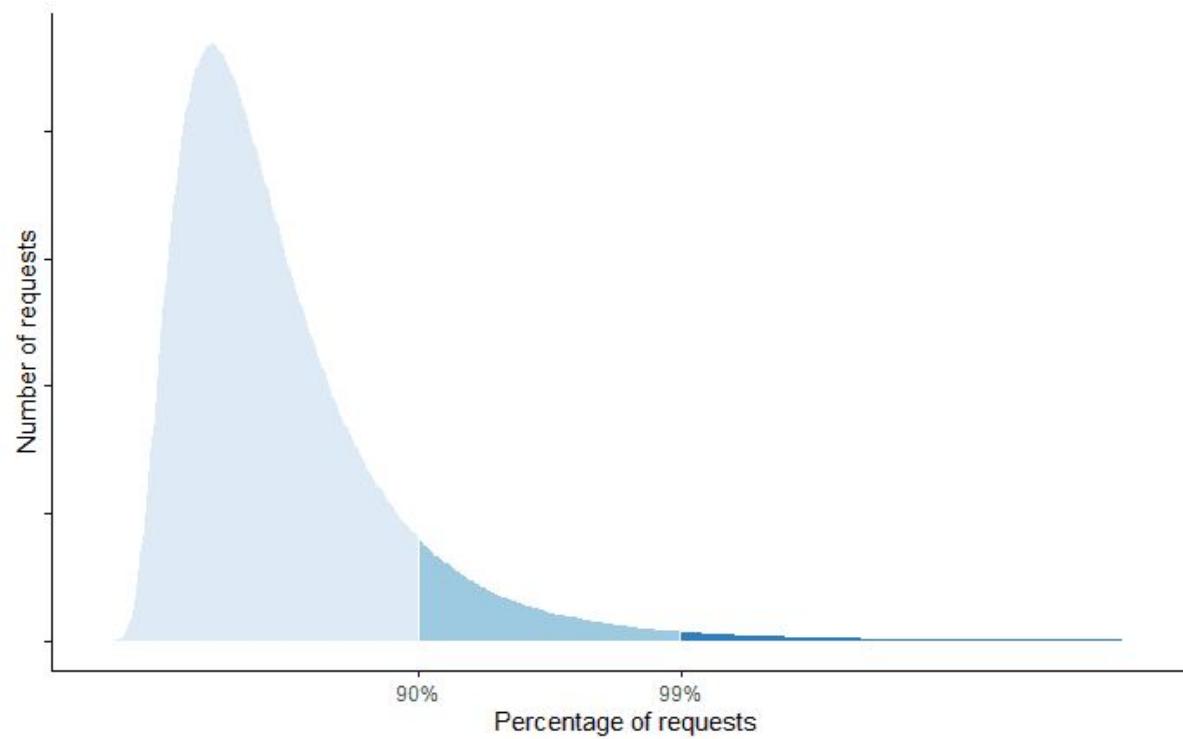
- Number of requests, errors, tasks completed
- `request_count_total {http_status="200"}`- Shows the count of requests that have status 200

Gauge

Value can go up or down

- Number of concurrent requests, running containers
- `rate(request_latency_seconds_count[1h])`- Shows the request latency over 1 hour

Resource level monitoring: Look at the tail of your distribution

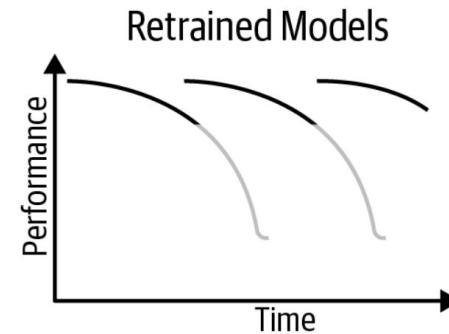
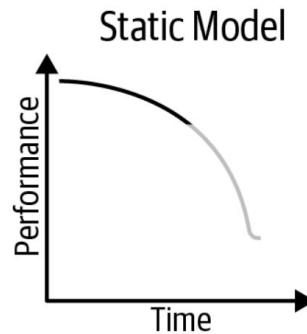


Performance Level Monitoring

Model performance monitoring

Models may become stale over time, which negatively impacts the quality of predictions

- Need to **retrain models**
- Present if the system experiences **drifts**
- Triggered by capturing **prediction inputs** and **outputs** and comparing with **ground truth**



*One of the reasons to have an automatic
(scheduled) retraining pipeline!*

Drift

Different types of drift:

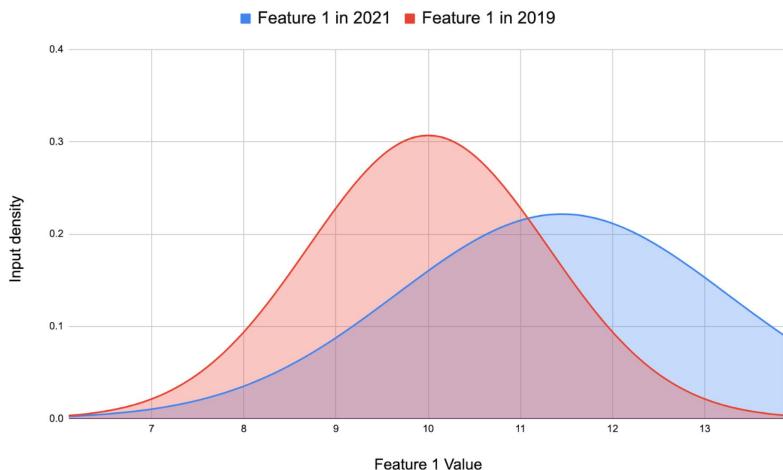
- *Data drift*
- *Target drift*
- *Concept drift*

Different occurrences of drift:

- *Sudden*
- *Gradual*
- *Incremental*
- *Reoccurring*

Different types of drift

Data drift

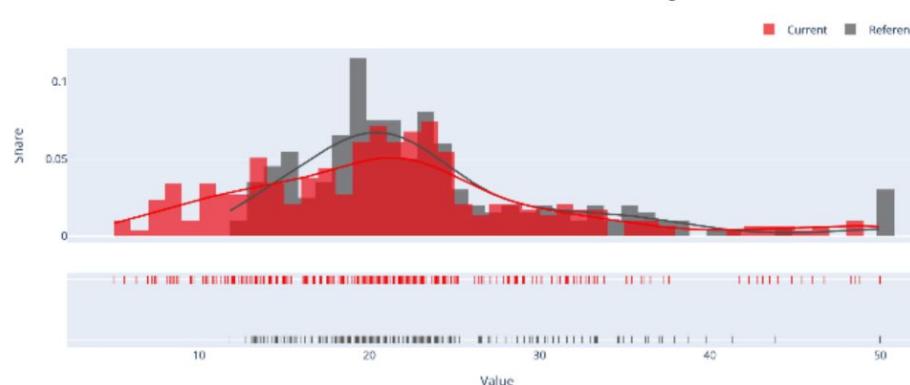


- Also known as **feature drift** or **covariate shift**
- Drift in the data inputs (features)
 - The distribution of the *production* data is different (or evolved differently) to the data the model was trained on
- Highlights the importance to not only monitor **model performance** but also **data distribution**

Different types of drift

Target drift

- Also known as **prediction drift**
- Drift in the **target variable**
 - Can be a shift in the *distribution* of the target (continuous or categorical target)
 - or the addition/removal of categories (categorical target)
 - E.g. a new product is added or deprecated in demand forecasting

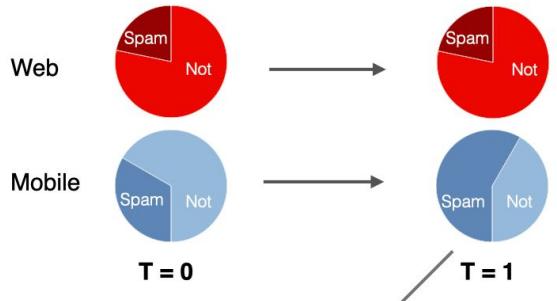


Different types of drift

Concept drift

- The relationship between the input data and the target has changed
 - The patterns the model learned to map between the original inputs and outputs are no longer relevant
- ⇒ Important to retrain models

Concept drift



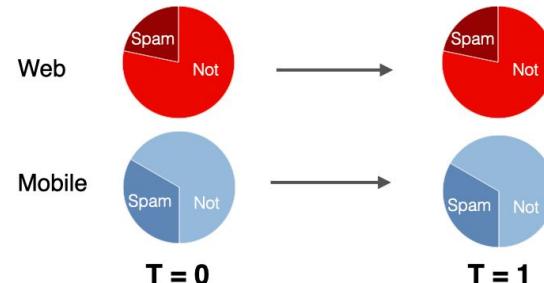
Change in relationship

“Device type” feature distribution over time



$P(Y | X)$ changes but
 $P(X)$ is the same

Data drift



Change in distribution

“Device type” feature distribution over time

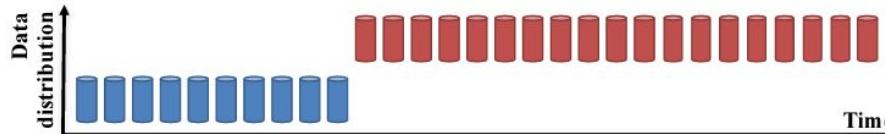


$P(Y | X)$ is the same
but $P(X)$ changes

Different occurrences of drift

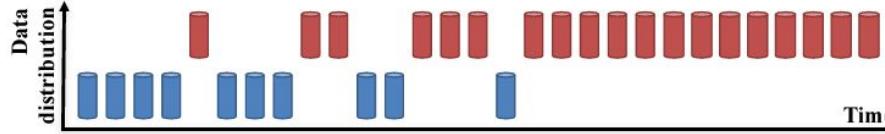
Sudden Drift:

A new concept occurs within a short time.



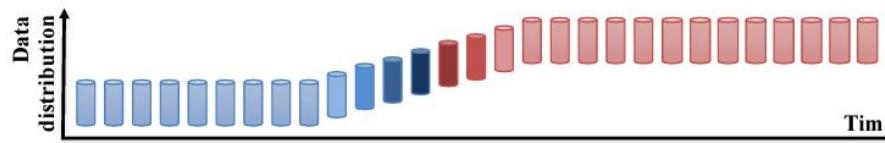
Gradual Drift:

A new concept gradually replaces an old one over a period of time.



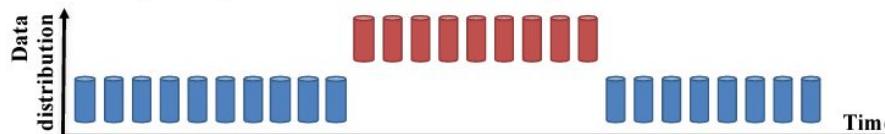
Incremental Drift:

An old concept incrementally changes to a new concept over a period of time.



Reoccurring Concepts:

An old concept may reoccur after some time.



Drift example: Predicting taxi tips

Project: Binary classification task: predict whether a passenger in a NYC taxi ride will give the driver a “reasonable” tip (>10% of fare)

- $X = \text{features}$ (e.g., location, ride length, ...), $Y = \text{labels}$ (high tip indicator)

Example of drifts:

- Data drift
 - ?

Drift example: Predicting taxi tips

Project: Binary classification task: predict whether a passenger in a NYC taxi ride will give the driver a “reasonable” tip (>10% of fare)

- $X = \text{features}$ (e.g., location, ride length, ...), $Y = \text{labels}$ (high tip indicator)

Example of drifts:

- Data drift
 - More taxi rides in Midtown area around NYE 
- Label shift
 - ?

Drift example: Predicting taxi tips

Project: Binary classification task: predict whether a passenger in a NYC taxi ride will give the driver a “reasonable” tip (>10% of fare)

- X = features (e.g., location, ride length, ...), Y = labels (high tip indicator)

Example of drifts:

- Data drift
 - More taxi rides in Midtown area around NYE 
- Label shift
 - Stimulus check causes people to tip more 
- Concept shift
 - ?

Drift example: Predicting taxi tips

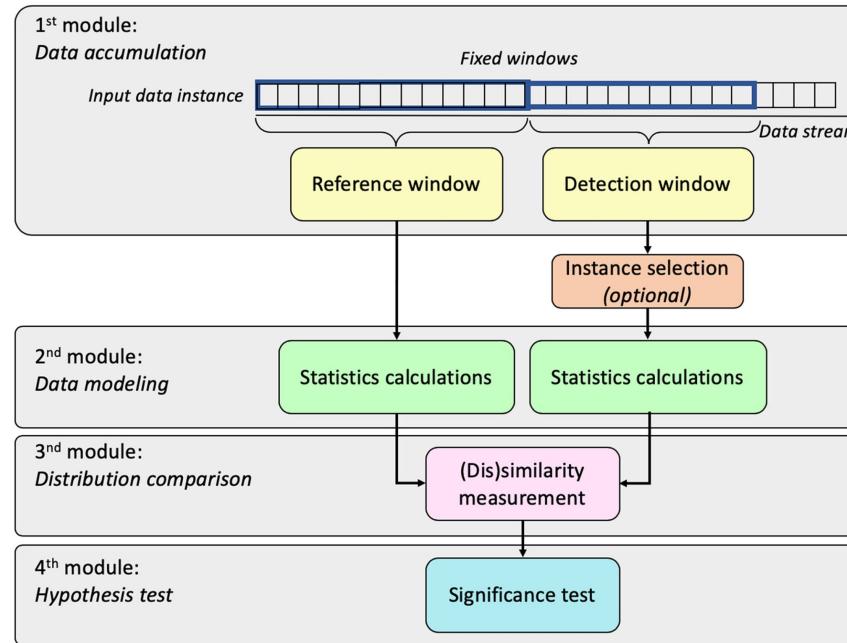
Project: Binary classification task: predict whether a passenger in a NYC taxi ride will give the driver a “reasonable” tip (>10% of fare)

- X = features (e.g., location, ride length, ...), Y = labels (high tip indicator)

Example of drifts:

- Data drift
 - More taxi rides in Midtown area around NYE 
- Label shift
 - Stimulus check causes people to tip more 
- Concept shift
 - Heavy construction in certain areas causes people to tip less 

Drift detection



Drift detection - level of sophistication

Straw-man approach 🍹

- Tracking means & quantiles of features and outputs

“I know stats” approach 🎓

- Tracking MMD, KS & Chi-Square test statistics, etc

⇒ None of those require labeled data!

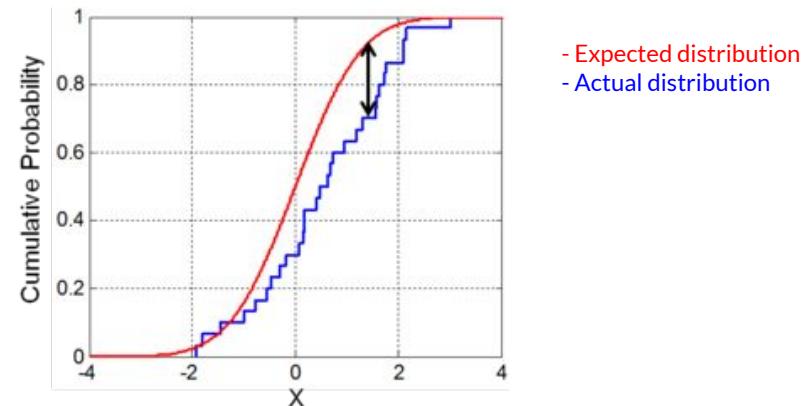
Similarity measurement

Kolmogorov-Smirnov test (KS)

Continuous data

- **Kolmogorov-Smirnov test (KS):** Determines the maximum distance between two distributions' cumulative density functions

The Kolmogorov–Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples.

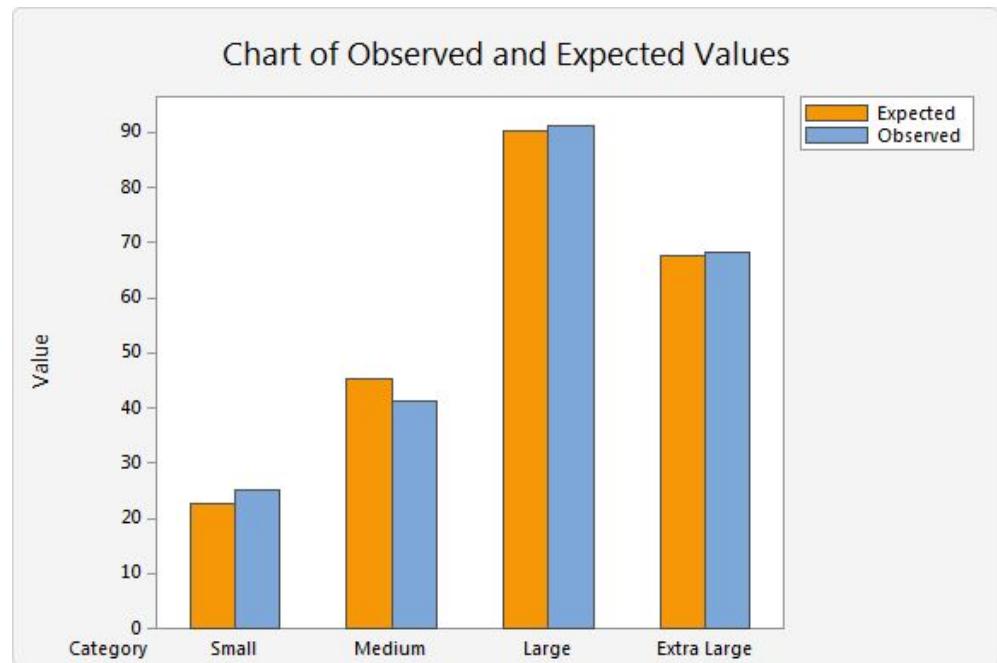


Similarity measurement

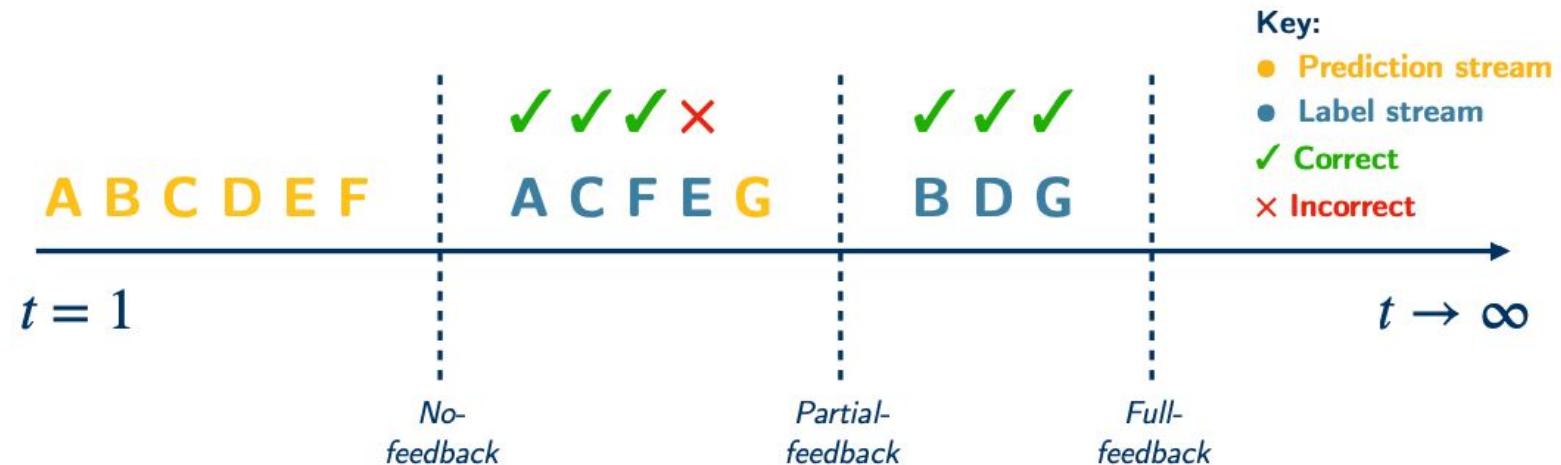
Pearson chi-square

Categorical data

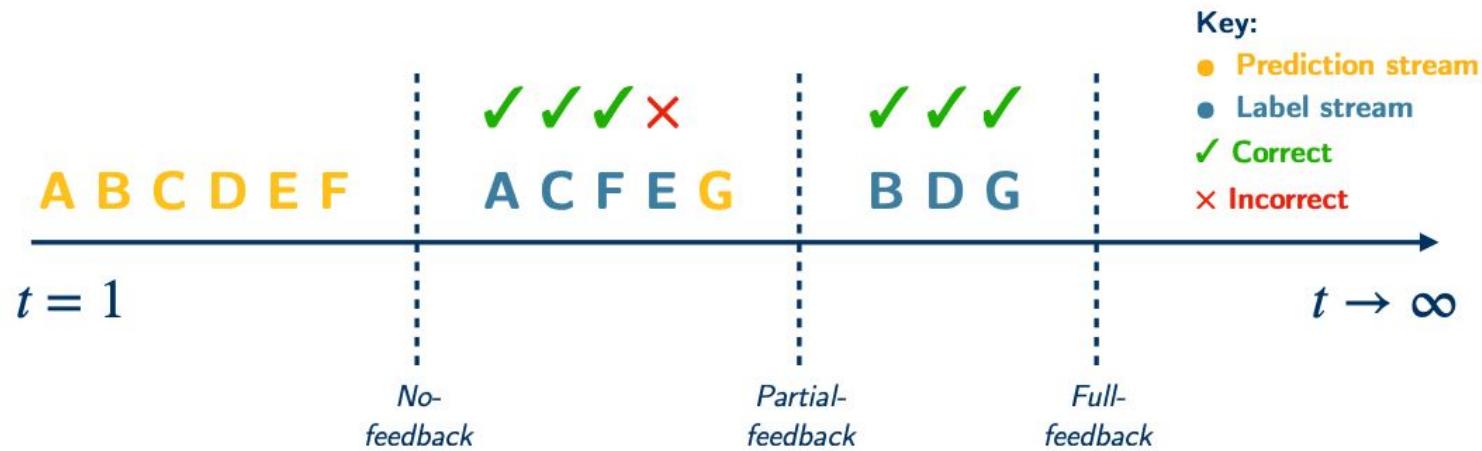
- **Pearson chi-square test:** Determines if a frequency of events in production is consistent with a reference distribution
 - E.g. measure if there is any drift on the size of a target variable



Detecting performance issues: Feedback Delays

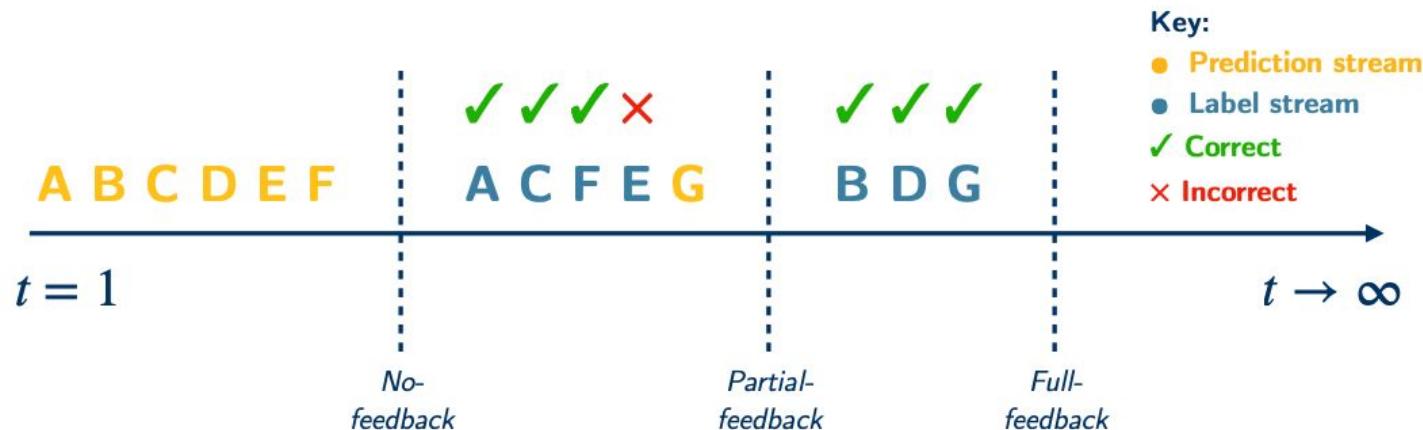


Detecting performance issues: Feedback Delays



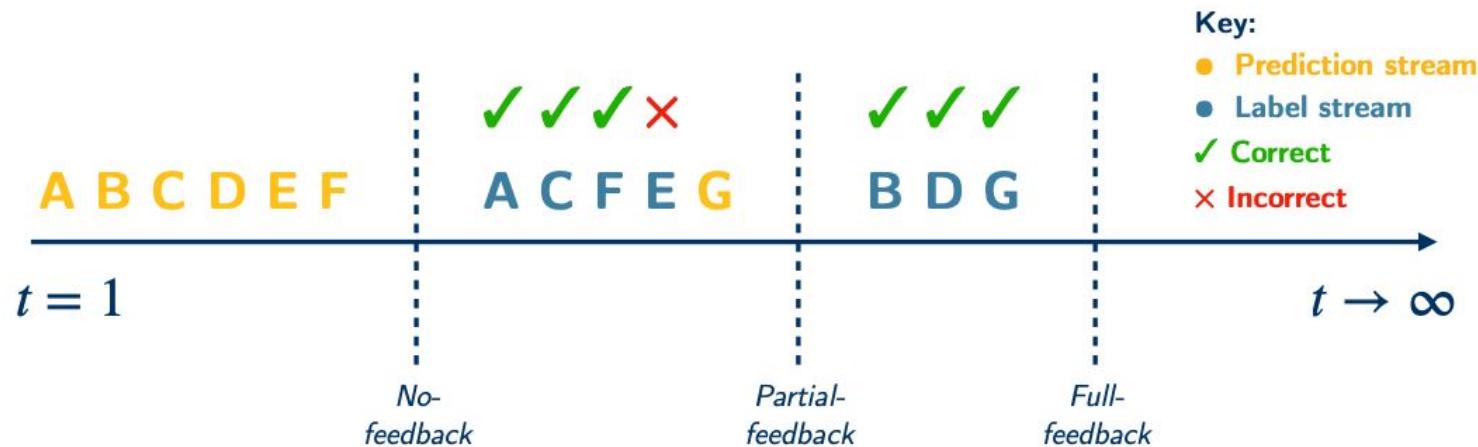
Accuracy: ?? 🤔

Detecting performance issues: Feedback Delays



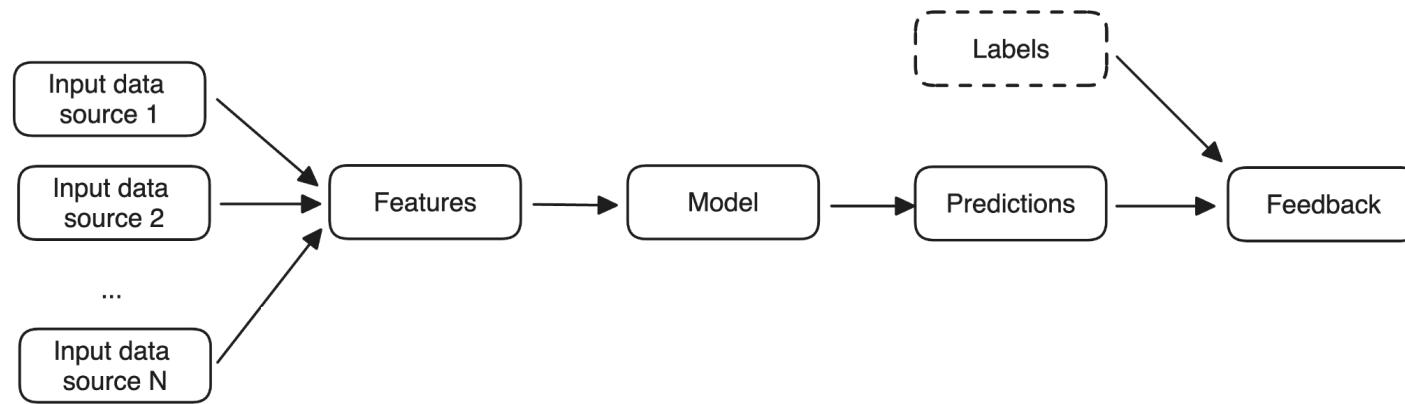
Accuracy: 75% ?? 🤔

Detecting performance issues: Feedback Delays

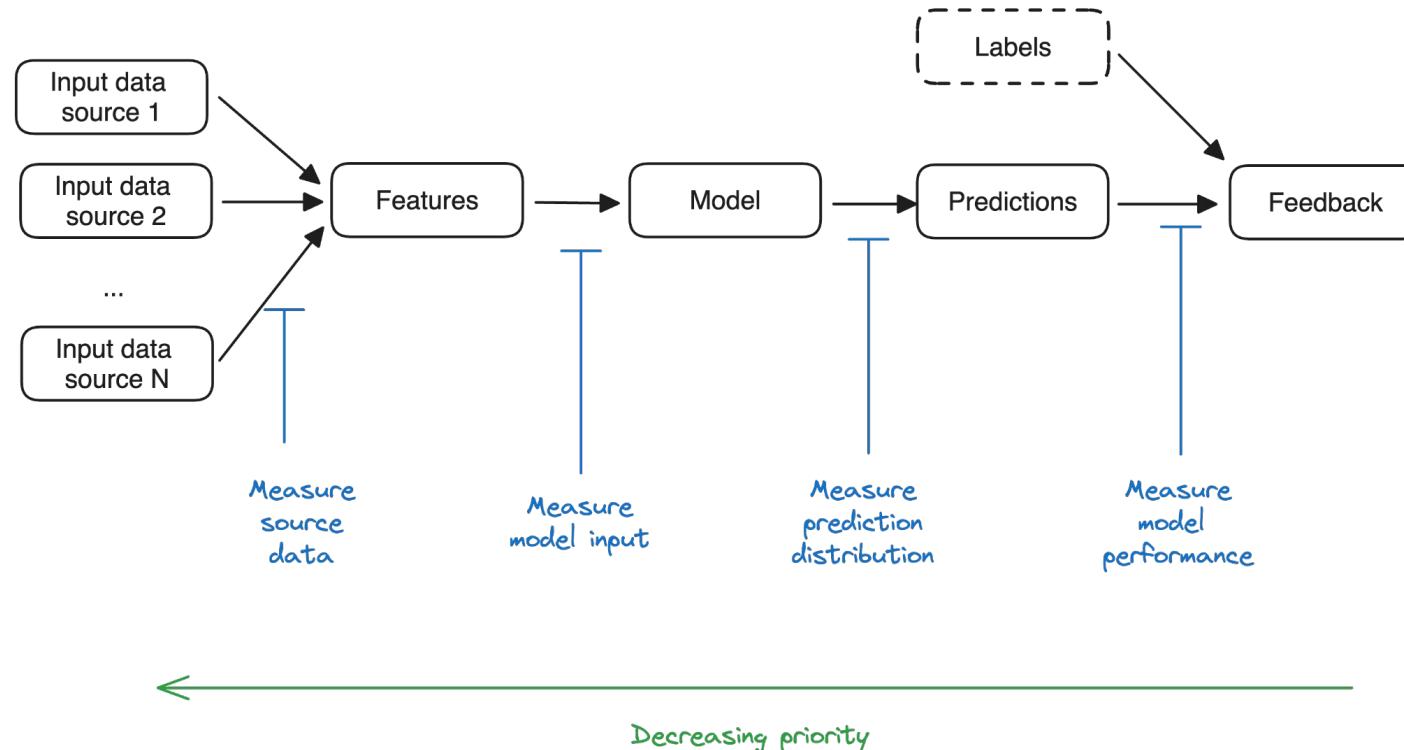


Accuracy: 86% 😊

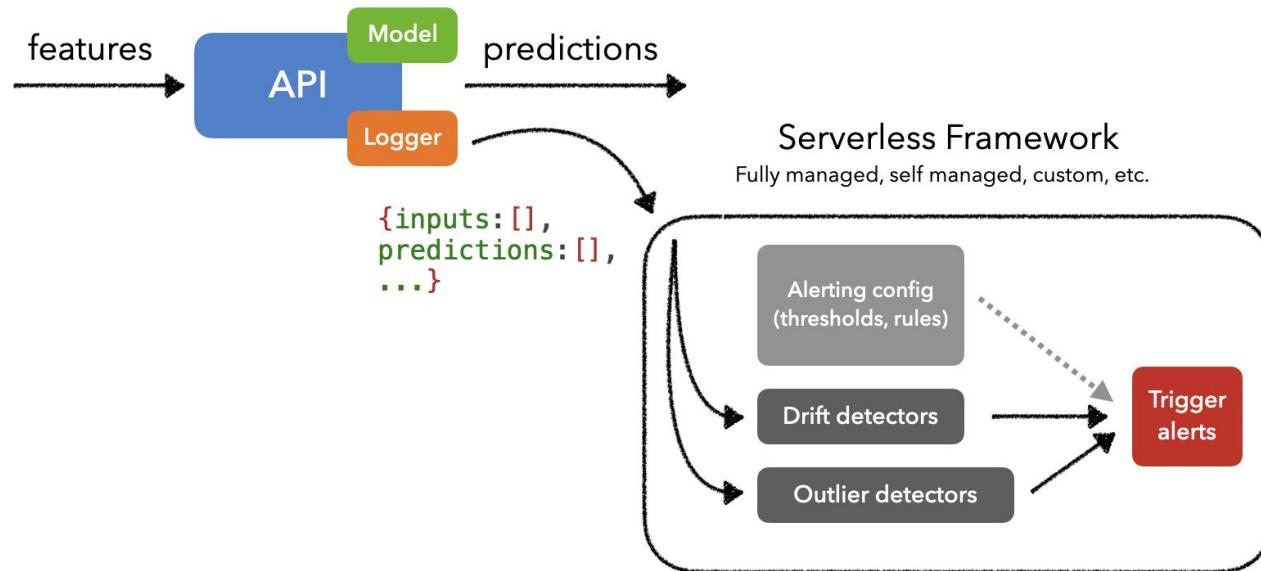
Different level of performance measurements for ML



Different level of performance measurements for ML



You can combine custom drift detection methods into your monitoring system



Breakout exercise

Find drifts and ways of detecting it

Exercise

- By groups of 3-5 try to answer the following questions
- Take 10min to answer them then we'll take 5min as a group to answer them

Use case: A supermarket chain wants you to predict the demand of their products a week ahead.

Questions:

- What kind of monitoring pipeline are we setting up?
- What kind of drift can we see happening?
 - Try to find practical examples for each type of drift
- How can you tackle these drifts in your design

Dashboarding



Always has been.

It's all about data...

ML



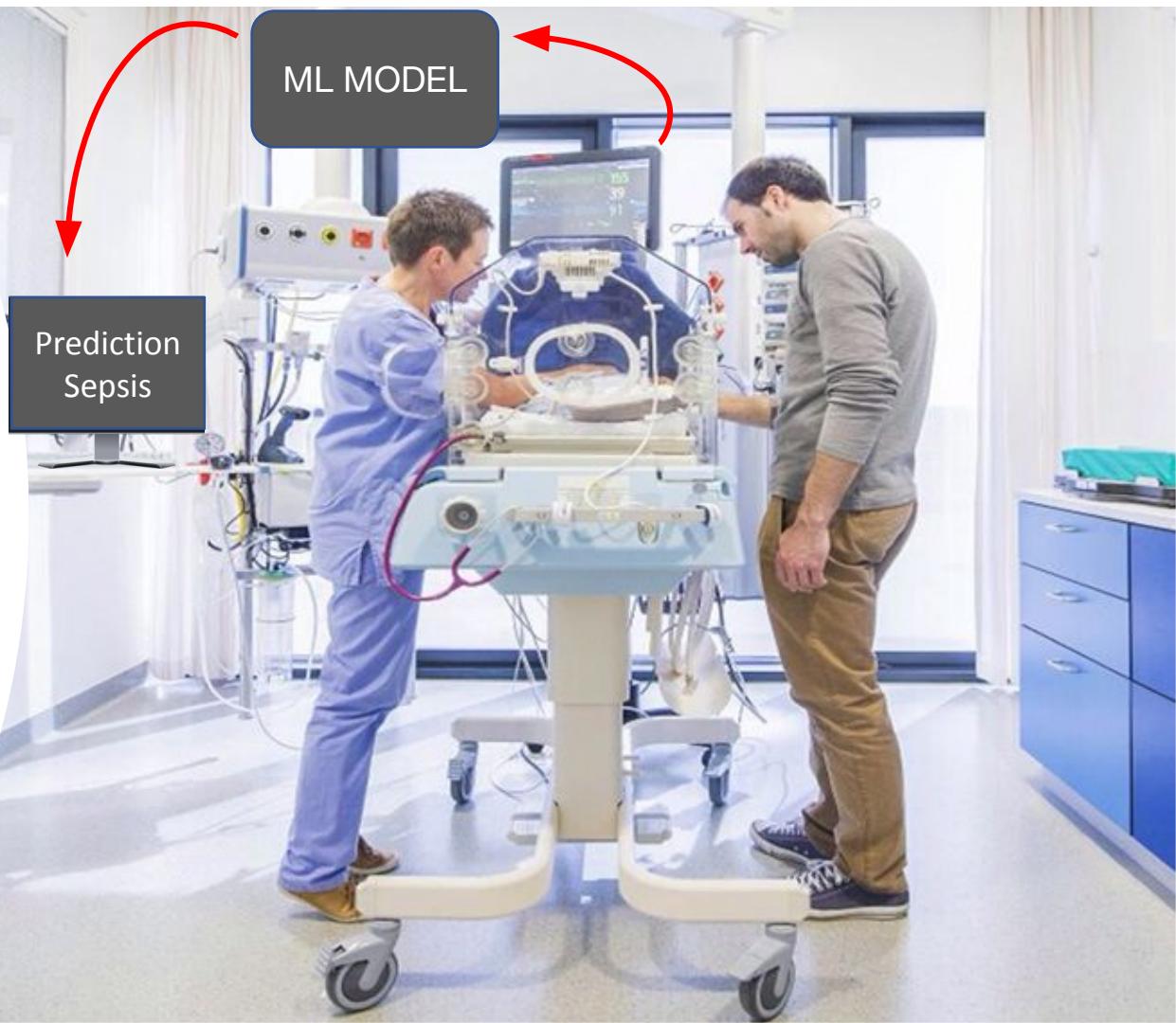
TRUST

ARTIFICIAL INTELLIGENCE DECISION SUPPORT

digital aid at every bed
checking data every 30 mins



alert “real” MD
if sepsis pattern detected



Innocens Project

Research & Development project



Team

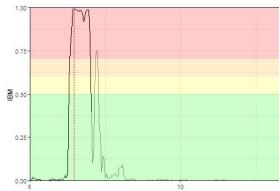


Universiteit
Antwerpen

A trustworthy AI based Clinical Decision Support Platform.

Prerequisites

A Machine Learning Model



High sensitivity

Detect (most) sepsis episodes

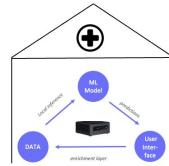
Time gain

Faster than MD

High precision

Low alarm rate

System Architecture



Privacy Compliant

No raw data leaving firewall

User experience



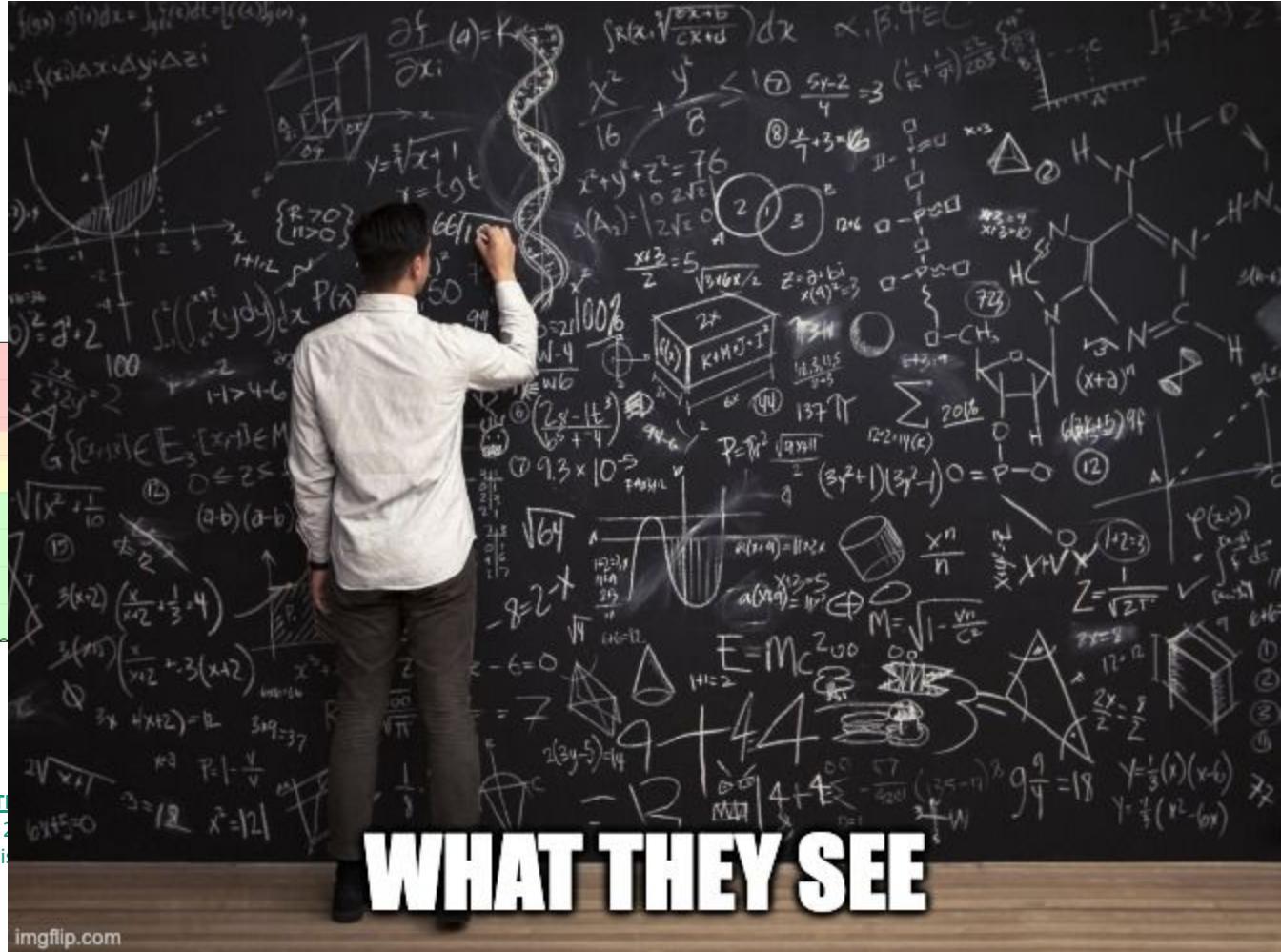
Explainable AI

Keep MD in control

ML probability LOS



TRUE POSIT
Any spike < 2 days
before sepsis



RECALL
(sensitivity)
 $= \text{TP}/(\text{TP} + \text{FN})$

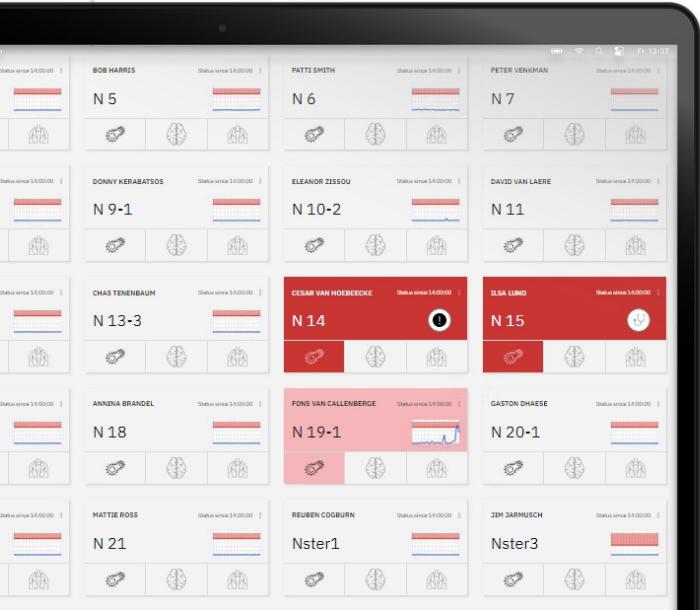
PRECISION
(PPV)
 $= \text{TP}/(\text{TP} + \text{FP})$

Explainable AI.

a digital aid for every bed doing 24/7 hourly rounds

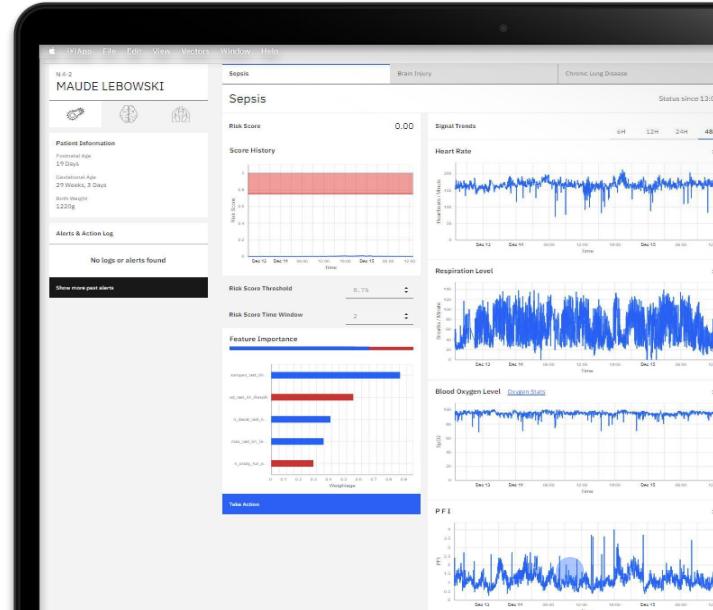


Ward governance at a glance



MacBook Pro

Feature importance



MacBook Pro

Innocens in action.

N 9-2

AVA GARLAND

Patient Information

Post Natal Age
7 Days

Gestational Age
27 Weeks, 2 Days

Alerts & Action Log

No logs or alerts found

Show more past alerts

Risk Score Threshold: 0.75

Risk Score Time Window: 5

Feature Importance:

Sepsis

Brain Injury

Chronic Lung Disease

Status since 05:00

Sepsis

Risk Score: 0.00

Score History

Risk Score

Time

Date	Risk Score
Jul 24	0.00
Jul 25	0.00
Jul 26	0.00
Jul 27	0.00
Jul 28	0.00
Aug 01	0.00
Aug 02	0.00
Aug 03	0.00
Aug 04	0.00

Signal Trends

Heart Rate

Heartbeats / Minute

Time

6H 12H 24H 48H

Respiration Level

Breaths / Minute

Time

Detecting impact of the covid crisis on job demand

And actually communicating it...

Context

- Human resource company helping with recruitment
- Processing hundreds of thousands of jobs per year
- Work offers highly impacted by the covid crisis
- Some jobs actually in high demand due to covid (nurses, cashiers, warehouse workers, ...)

Objective

- Predict which unique job demand is urgent due to covid
- Look at aggregated results to steer efforts

Detecting impact of the covid crisis on job demand

And actually communicating it...



Communicating results: Performance metrics

Accuracy: 79.67%

F1 score: 83.93%

Precision: 85.33%

Recall: 82.58%

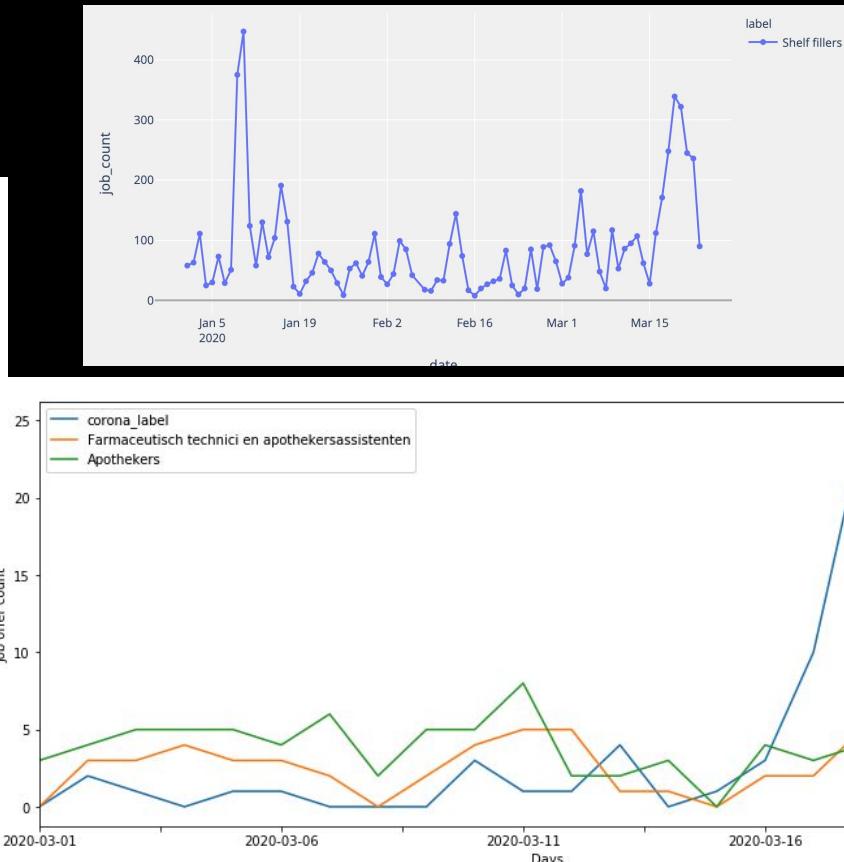
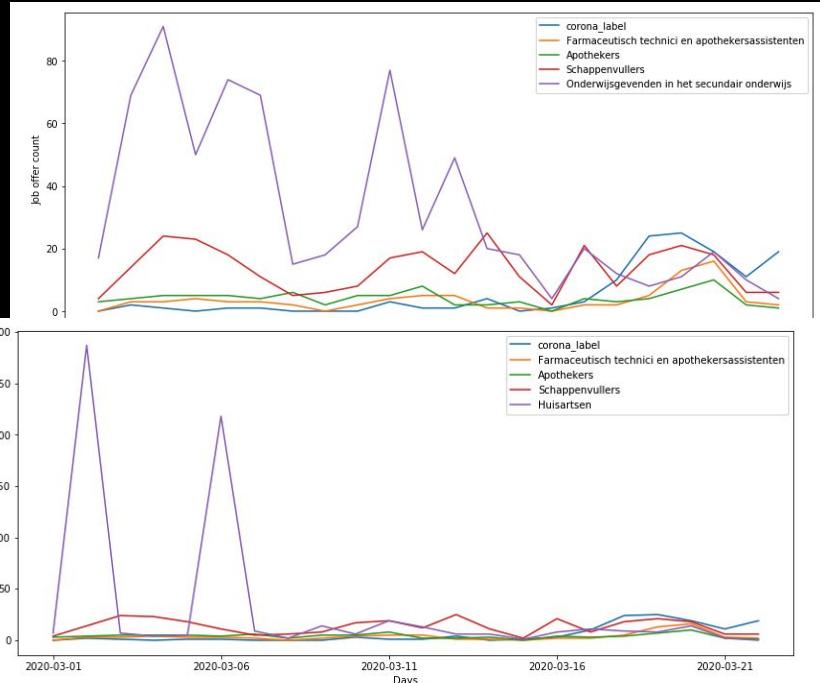
	Negative	Positive
True\Pred		
Negative	64	22
Positive	27	128

Detecting impact of the covid crisis on job demand

And actually communicating it...



Communicating results: Static plots



Detecting impact of the covid crisis on job demand

And actually communicating it...



Communicating results: Track per customs

company	Sector	21/03/2020	22/03/2020	23/03/2020	24/03/2020	25/03/2020	26/03/2020
AVIQ MONS	Public						
SPORTSDIRECT.COM MEDIACITE	Retail						
SCARPINA 02	Retail						
CURA VZW	Public						
HIS - IXELLES CUISINE KK	Hotel						
UBEM SURVEYS	Logistics						
GOOD DISPLAY WALLONIE/BXL	Retail						
VERBEKEN EVA-PAB	Retail						
RENEWI EVERGEM 02	Manufacturing						
ARMONEA NV WZC MILLEGEM	Manufacturing						

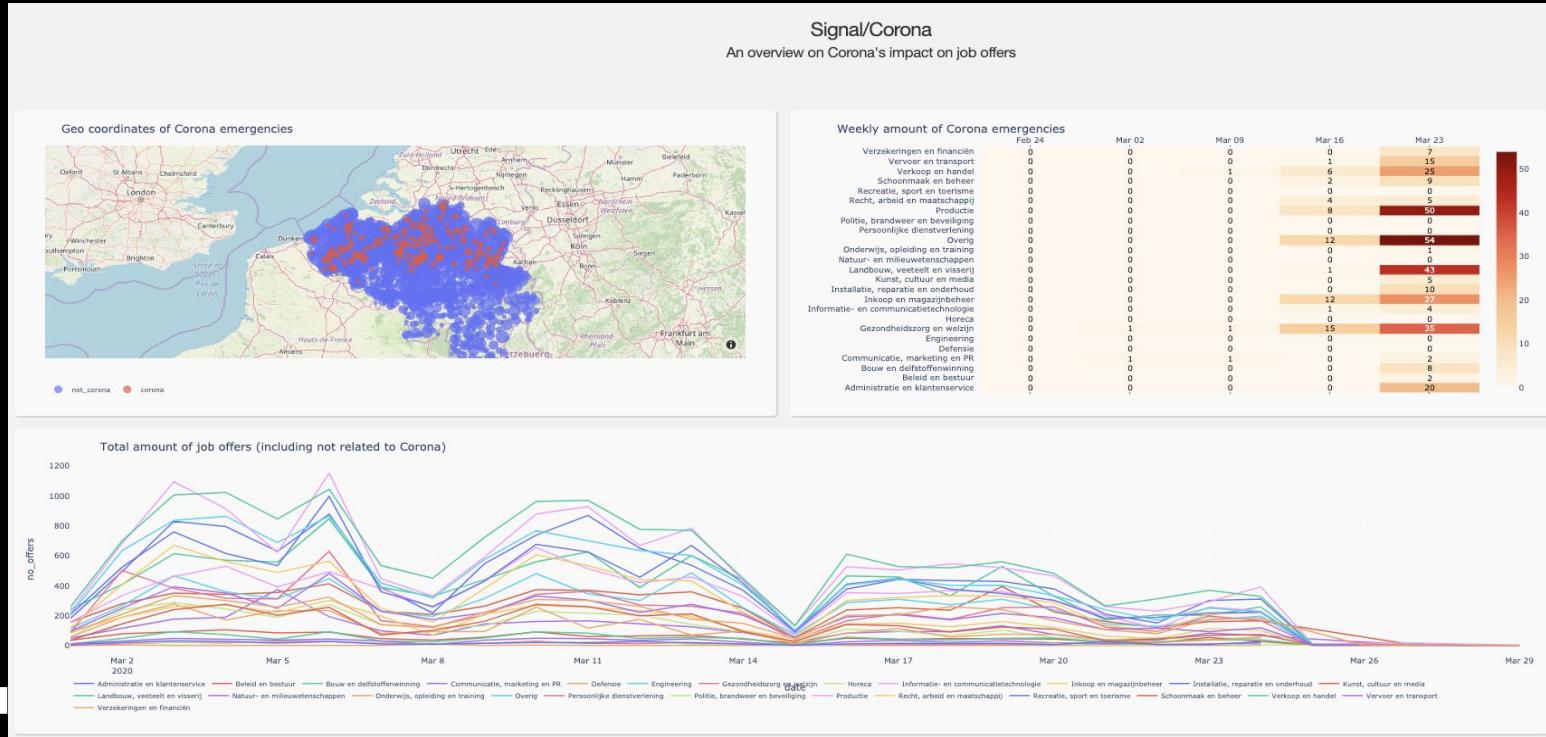
Detecting impact of the covid crisis on job demand

And actually communicating it...



plotly | Dash

Communicating results: Interactive dashboard

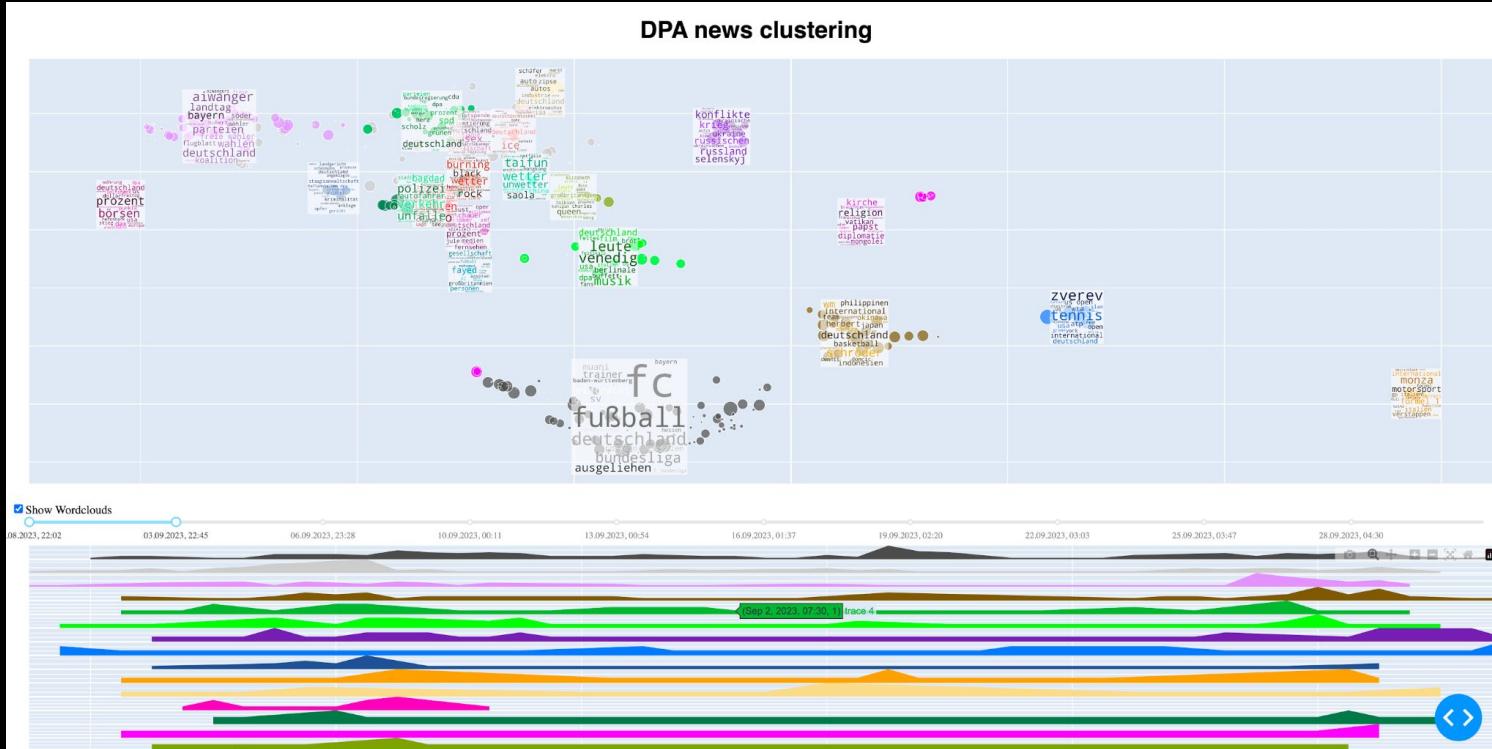


EDA on social media posts

How to actually understand your data with clustering



plotly | Dash



Interactive
dashboard

Wordclouds to
understand
clusters

Timeline

Some dashboarding best practices

- Don't put every possible metric on the dashboard
 - Focus on the top few metrics
 - No more than 5 graphs per console, 5 plots per graph
 - Split out/trim dashboards when they get too complex
- Have units, y-labels, legends and descriptions for your plots!
- Dashboards are not for alerting

Lab: Streamlit

Wrap-up

Lecture summary

Topic	Concepts	To know for...	
		Project	Exam
Monitoring	<ul style="list-style-type: none">• Logging vs monitoring• Why we need monitoring• Resource vs performance monitoring• Model performance monitoring• Drift types: data, target and concept		Yes
Dashboarding	<ul style="list-style-type: none">• Importance of dashboarding• Examples		
Lab: Streamlit	<ul style="list-style-type: none">• How to use it	(Optional)	

Project objective for sprint 5

#	Week	Work package	Requirement
5.1	W09	<p>Run your model training as a job in the Cloud. You can implement this in different ways:</p> <ul style="list-style-type: none">• Containerise your training script and run it on a VM in the Cloud (e.g. on EC2 or on Cloud Run, example,)• Use a managed service such as Vertex Training or Sagemaker Training <p>Attention: This can incur Cloud costs. Make sure to use a platform where you have credits and not burn through them. You can ask for support from the teaching staff in that regard.</p>	Optional
5.2	W10	Build a simple user interface or dashboard to show your results and deploy it on the Cloud.	Optional

That's it for today!

