Monitoring & dashboarding

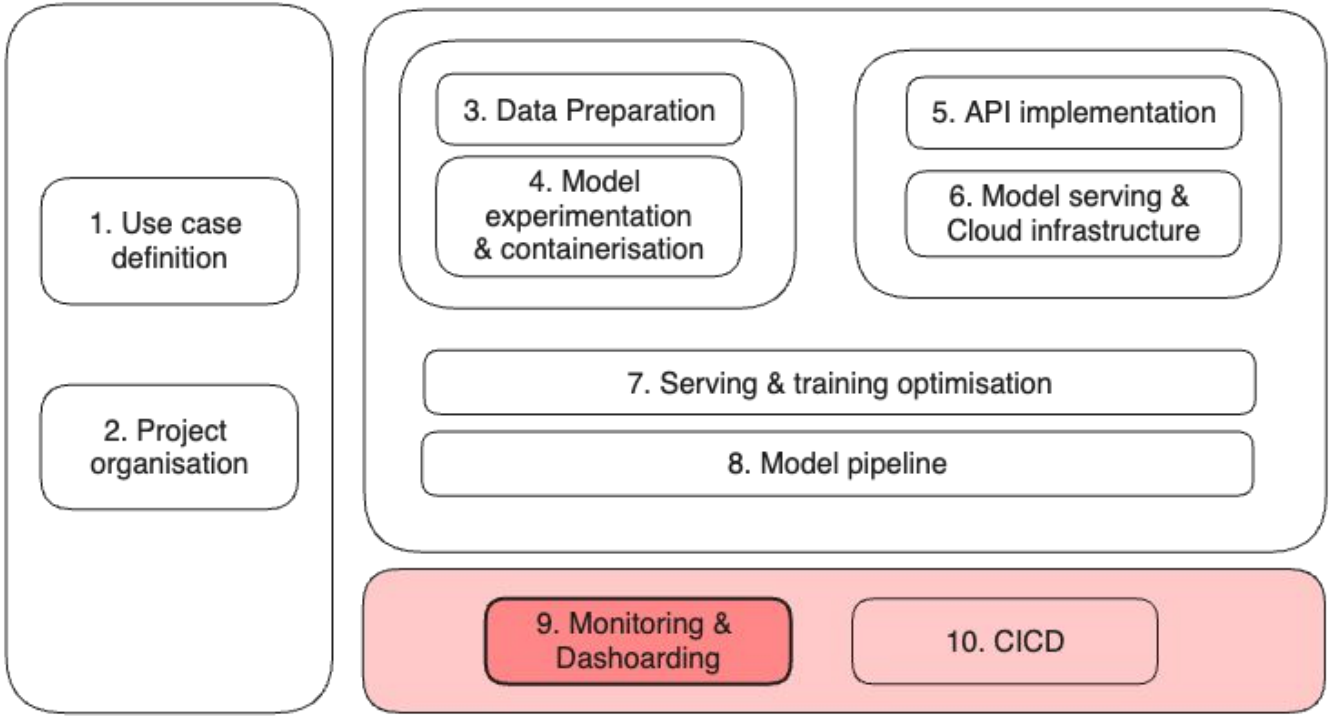
Sprint 5 - Week 09

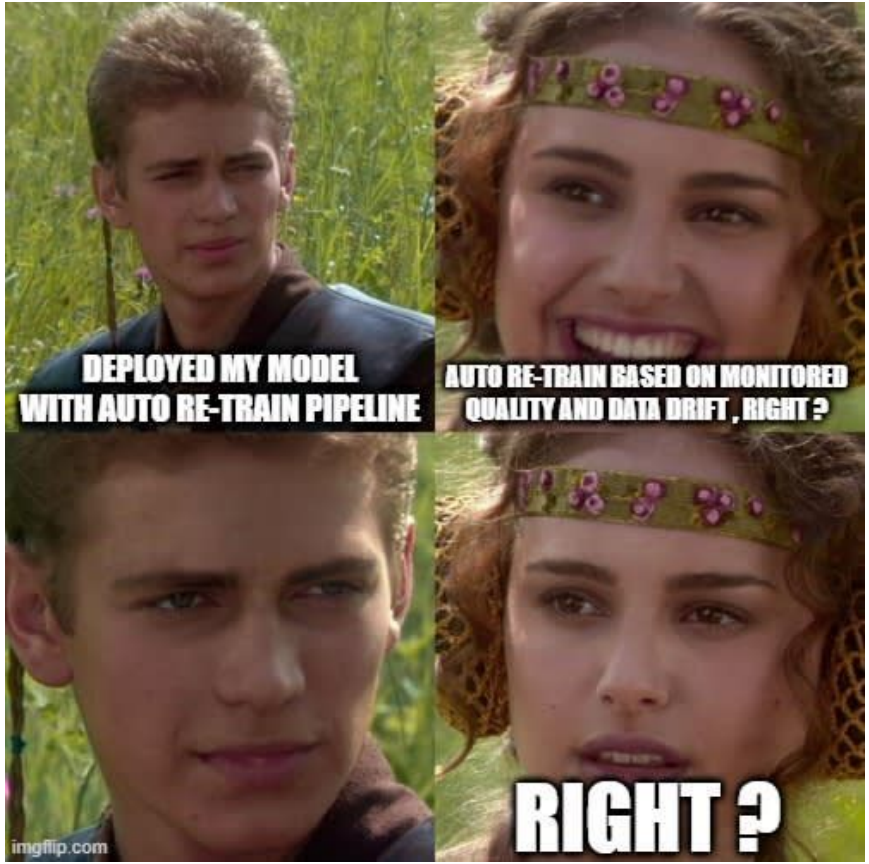
INFO 9023 - Machine Learning Systems Design

2024 H1

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

Status on our overall course roadmap




DEPLOYED MY MODEL
WITH AUTO RE-TRAIN PIPELINE
AUTO RE-TRAIN BASED ON MONITORED
QUALITY AND DATA DRIFT , RIGHT ?
RIGHT ?
imgflip.com

Agenda

What will we talk about today

Lab (20 min)

1. Google Cloud Storage (GCS)

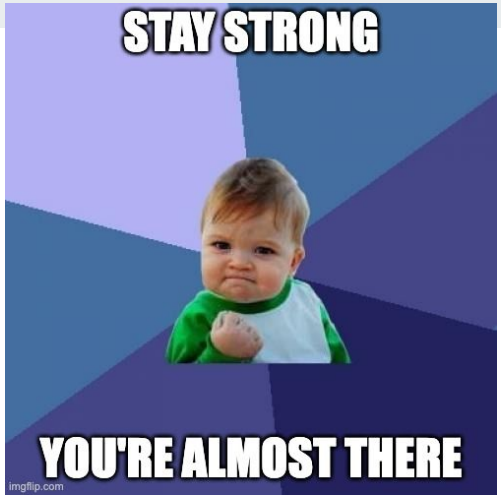
Lecture (1h min)

2. Monitoring
3. Dashboarding

Lab (40 min)

4. Streamlit

Exam info



Exam

- Oral exam: Answer **questions** based on a **use case**
- 30 min preparation + 15 min oral exam

Practical

- Date: 31/05
- We'll send a link for you to say at what time you want to pass it
- We'll put another practice exam on Github
- Exam is 30% of the final grade, 70% is for the project

Tip

- Start by looking at the practice exam to know what types of questions will be asked
- Look at the table at the end of each lectures which states what needs to be known
- *Motivate* why you take certain design choices

Project ~~update~~ - reminder from last week

⚠️ Decision to **combine Milestone 2 and 3 !** ⚠️

There will be only one presentation, taking place in our last **course** (13/05/2024) !

How? (Info is also on github)

1. Same content as MS 3: Present your overall project in 10 min presentation + 5 min QA
2. It will be in classroom ! You're very welcomed to attend other project presentations
3. Send codes (link to pull request) by email **before the presentation**
4. Agenda for the presentations is in this sheet (let us know if it does not work for you)
5. Topics from the last "bonus" lecture are going to be spread over other lectures

Why?

👫 Nicer to let everyone attend all presentations

⚙️ Simpler organisation as there is already a class block in your agenda, don't take more of your time

📃 Avoid having to repeat topics between

Lab: Google Cloud Storage (GCS)

Monitoring

Logging vs Monitoring

Logging	Monitoring

● **Qualitative** (and typically extensive) information about current application state, events, errors, etc.
● (typically) Time based
	● **Quantitative**
	  ○ Captures the rate of change of numerical data
	● Time-series based
	  ○ We look at something that happens consistently over time

Why do we need monitoring?

Alerting

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

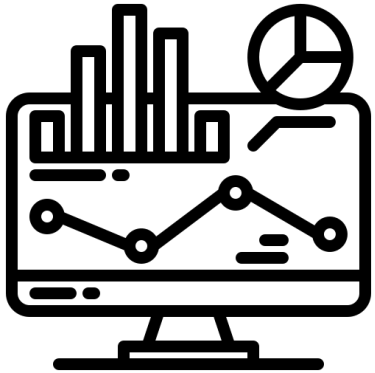
Debugging

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

Trending

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

Model monitoring perspectives



Resource level

1. Monitor the usage of resources used by the model serving
2. Ensuring the model is running correctly in the (production) environment
3. Key questions:
    a. Is the system alive? How much is it being used by users?
    b. Are the CPU, RAM, network usage, and disk space as expected?
    c. What are the Cloud costs?
    d. Are requests being processed at the expected rate?
    e. What is the system uptime? Some maintenance contract depend on it.

Performance level

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

Prof. Filippo Lanubile - University of Bari - https://upcommons.upc.edu/bitstream/handle/2117/390805/ICSE_SEET_2023_MLOps.pdf

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.

https://sre.google/sre-book/monitoring-distributed-systems/

Resource level monitoring

Most Cloud products have a monitoring dashboard.

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

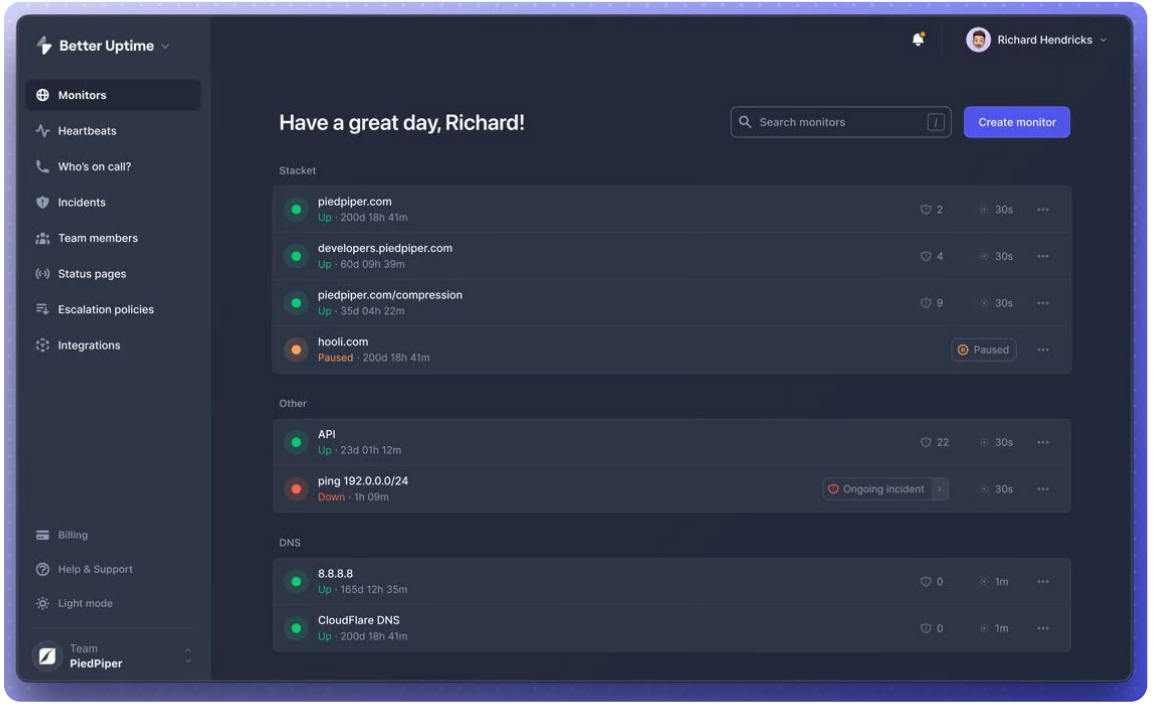
Metric	Services	Jobs
Billable container instance time	✓	✓
Container startup latency	✓	✓
Container CPU utilisation	✓	✓
Container memory utilisation	✓	✓
Sent bytes	✓	✓
Received bytes	✓	✓
Request count	✓	
Request latencies	✓	
Container instance count	✓	
Maximum concurrent requests	✓	
Completed executions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

https://cloud.google.com/run/docs/monitoring

Resource level monitoring: (Better) Uptime

Checking whether a service, such as a website or an application, is available. Uptime monitoring spots issues and alerts the right person on the development team.

An HTTP status code monitor checks your URLs every X seconds for an HTTP success status code ??? . If the URL doesn't return this code, the monitor creates an Incident and alerts the current on-call person. If this person doesn't respond to the incident, an entire team is alerted.

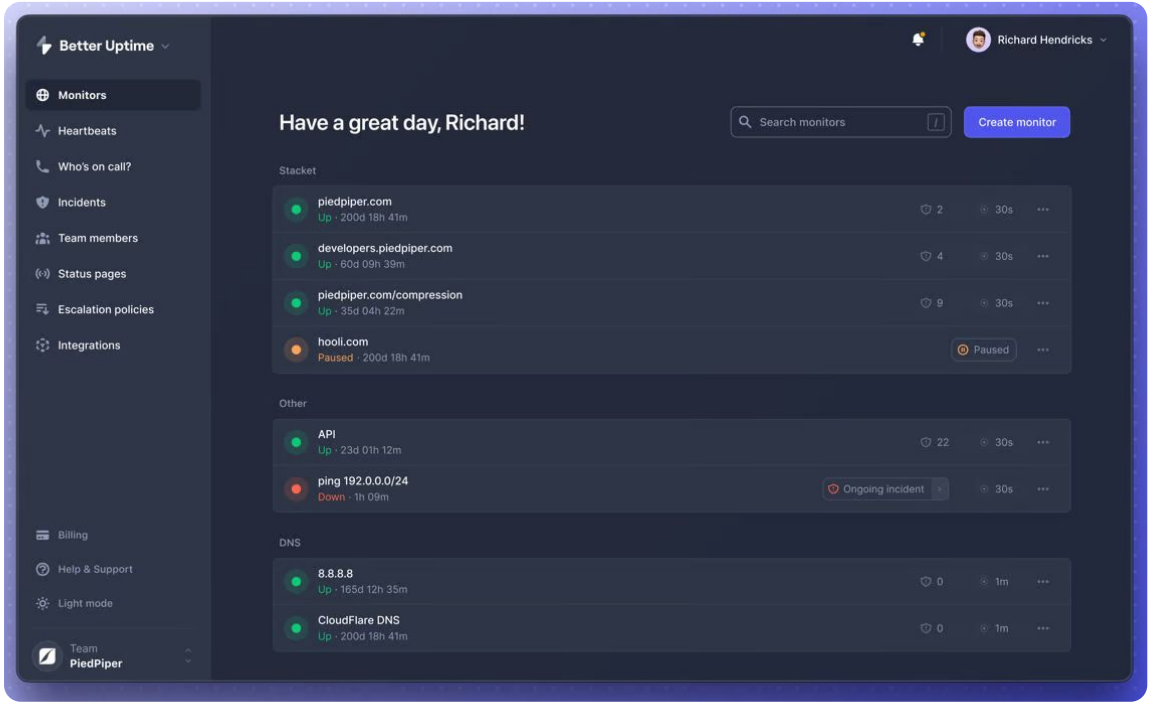


https://betterstack.com/docs/uptime/monitoring-start/

Resource level monitoring: (Better) Uptime

Checking whether a service, such as a website or an application, is available. Uptime monitoring spots issues and alerts the right person on the development team.

An HTTP status code monitor checks your URLs every X seconds for an HTTP success status code (2XX). If the URL doesn't return this code, the monitor creates an Incident and alerts the current on-call person. If this person doesn't respond to the incident, an entire team is alerted.



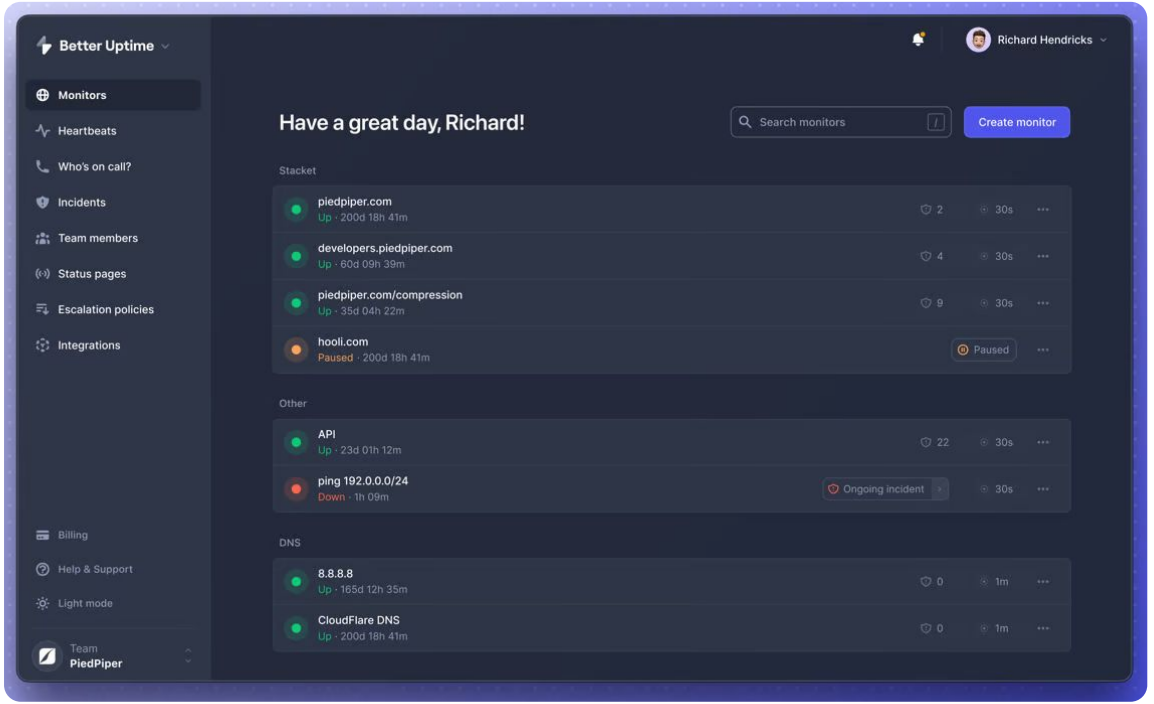
https://betterstack.com/docs/uptime/monitoring-start/

Resource level monitoring: (Better) Uptime

Steps:

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

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



https://betterstack.com/docs/uptime/monitoring-start/

Resource monitoring & testing: Locust

Locust (https://github.com/locustio/locust)

1. **Open source performance/load testing tool** for HTTP and other protocols.
2. Developer friendly approach → define your tests in regular Python code.
3. Locust tests can be run from **command line** or using its **web-based UI**.
4. Monitor:
   a. Throughput
   b. Latency
   c. Errors
5. Real time monitoring and/or exported for later analysis.

Resource monitoring with Locust

LOCUST

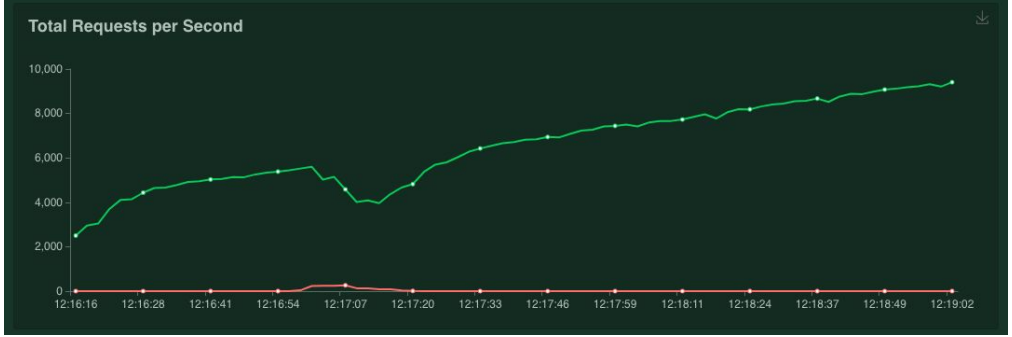
HOST https://your-app.com | STATUS SPAWNING 6000 users Edit | WORKERS 75 | RPS 9688.9 | FAILURES 1% | STOP | Reset Stats

Statistics | Charts | Failures | Exceptions | Download Data | Workers

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
PUT	accept_delivery	1974	2	270	300	277	10	2014	1508	15.1	0
POST	add_device_status	93266	50	23	28	28	7	4617	0	703	2.3
PUT	cancel_delivery	2007	0	270	300	274	236	1424	1510	14.1	0
PUT	close_delivery	1941	1	270	300	277	236	2015	1509	16.7	0
GET	get_auth_current	35	0	8	51	25	4	95	87	0.2	0
GET	get_deliveries_by_id	47309	24	13	33	25	4	59880	1517	358.7	0.9

HOST https://your-app.com | STATUS SPAWNING 5700 users Edit | WORKERS 75 | RPS 9406.9 | FAILURES 0% | STOP | Reset Stats

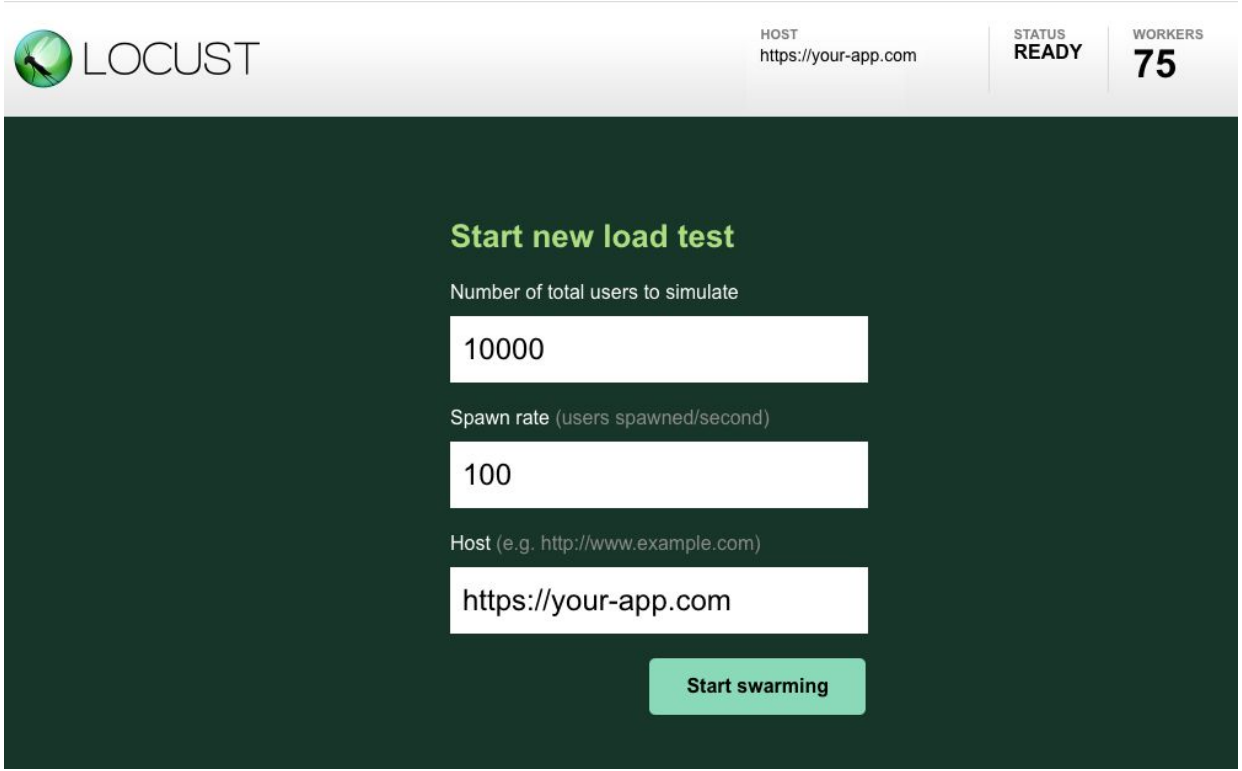
eptions | Download Data | Workers



LIÈGE université

Running load tests with Locust

(Not strictly *monitoring* but still interesting)



Write user test scenarios in plain old Python with Locust

```python
from locust import HttpUser, task, between

class QuickstartUser(HttpUser):
    wait_time = between(1, 2)

    def on_start(self):
        self.client.post("/login", json={"username":"foo",
"password":"bar"})

    @task
    def hello_world(self):
        self.client.get("/hello")
        self.client.get("/world")

    @task(3)
    def view_item(self):
        for item_id in range(10):
            self.client.get(f"/item?id={item_id}", name="/item")
```

Resource level monitoring: Common metric types

Counter

Value can only be increased or reset to zero.

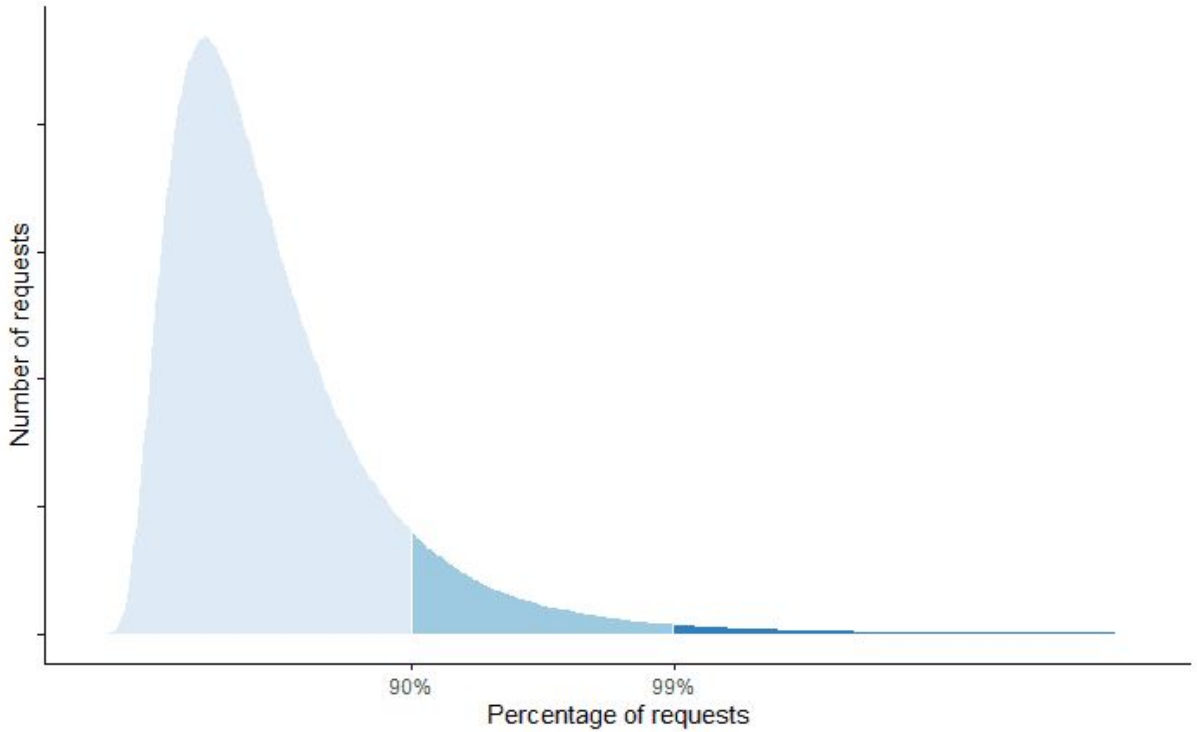
1. Number of requests, errors, tasks completed
2. `request_count_total {http_status="200"}`- Shows the count of requests that have status 200

Gauge

Value can go up or down

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

Resource level monitoring: Look at the tail of your distribution



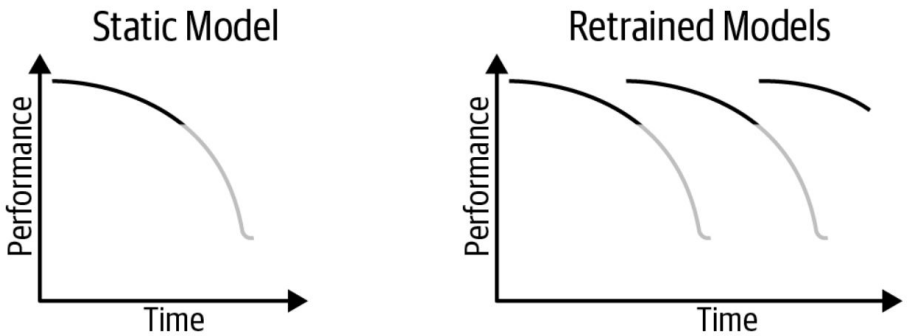
https://robertovitillo.com/why-you-should-measure-tail-latencies/

Performance Level Monitoring

Model performance monitoring

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

1. Triggered by capturing **prediction inputs** and **outputs** and comparing with **ground truth**
2. Needed to **retrain models**
3. Present if the system experiences **drifts**



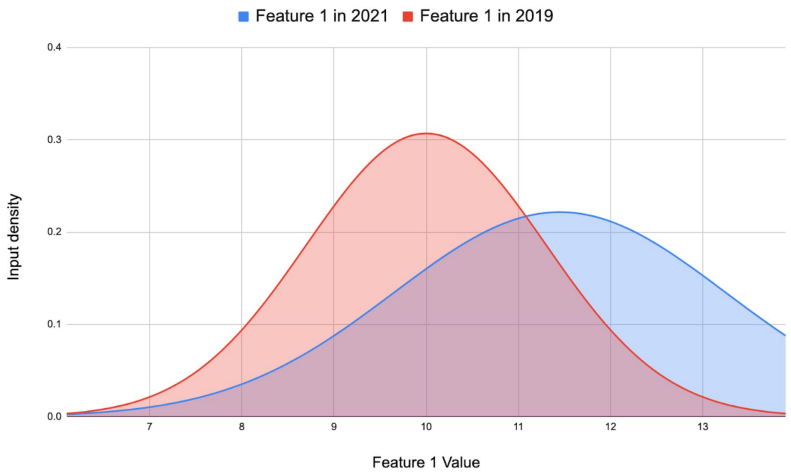
One of the reasons to have an automatic (scheduled) retraining pipeline!

Drift

Need to understand the different types of issues that can cause model's performance to decay

1. **Data drift**
2. **Target drift**
3. **Concept drift**

Data drift

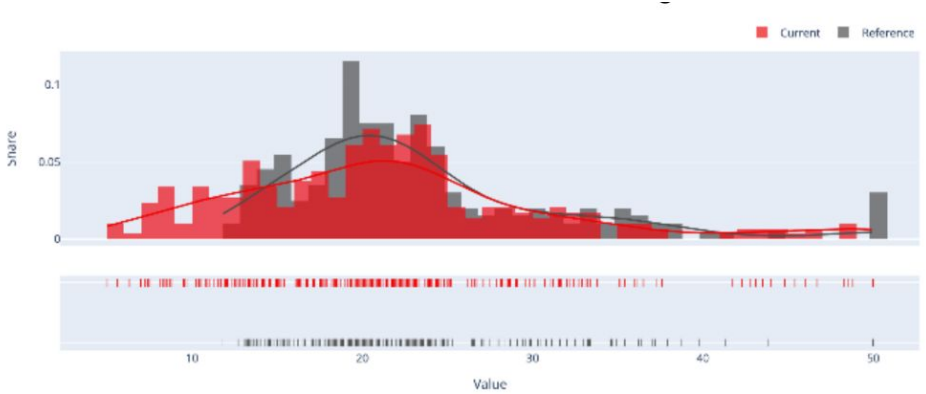


1. Also known as **feature drift** or **covariate shift**
2. Drift in the data inputs (features)
    a. The distribution of the *production* data is different (or evolved differently) to the data the model was trained on
3. Highlights the importance to not only monitor **model performance** but also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

https://www.picsellia.com/post/what-is-data-drift-and-how-to-detect-it-with-mlops

Target drift

1. Also known as **prediction drift**
2. Drift in the **target variable**
    a. Can be a shift in the *distribution* of the target or the addition/removal of categorical variables
        i. E.g. a new product is added or deprecated in demand forecasting



https://docs.evidentlyai.com/reports/num-target-drift

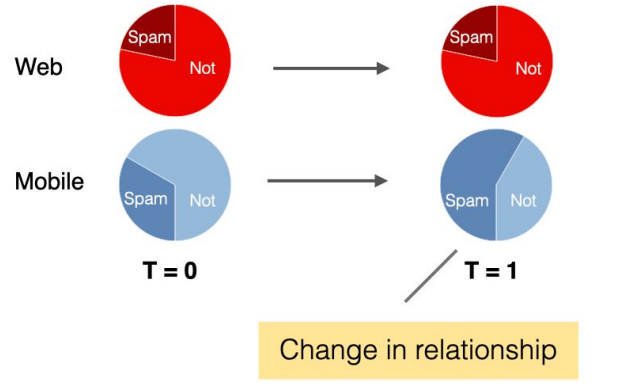
Concept drift

1. The relationship between the input data and the target has changed
2. The patterns the model learned to map between the original inputs and outputs are no longer relevant

⇒ Importance to retrain models

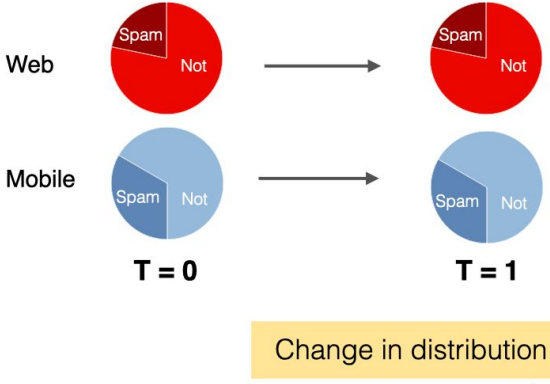
https://madewithml.com/courses/mlops/monitoring/

Concept drift



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

Data drift



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

https://www.evidentlyai.com/ml-in-production/concept-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 = features (e.g., location, ride length, …), Y = 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 = 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
  - ?

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

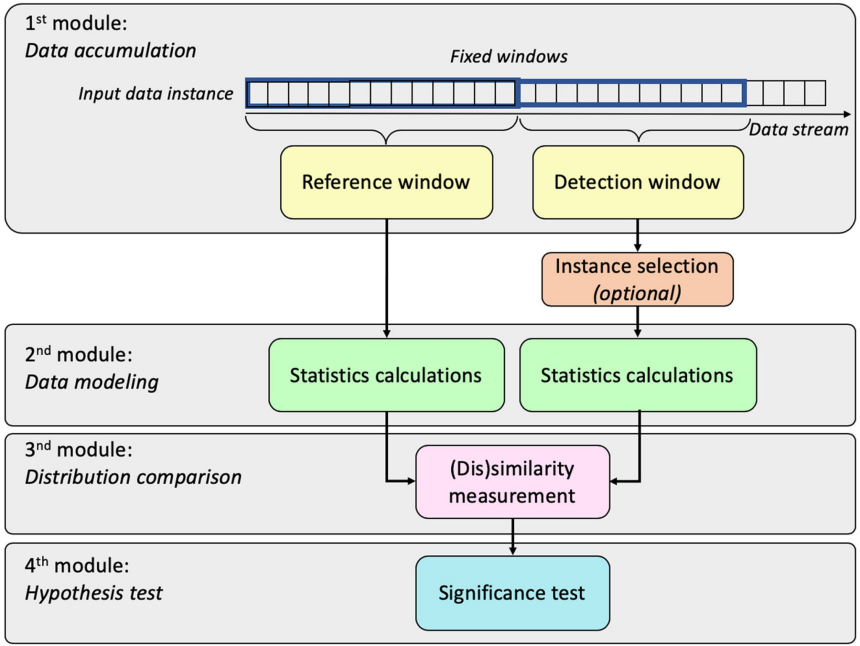
Project: Binary classification task: predict whether a passenger in a NYC taxi ride will give the driver a “reasonable” tip (>10% of fare)

- 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



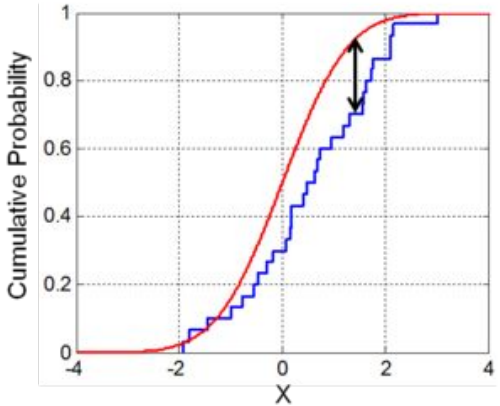
https://wires.onlinelibrary.wiley.com/doi/full/10.1002/widm.1381

Similarity measurement on continuous feature

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.



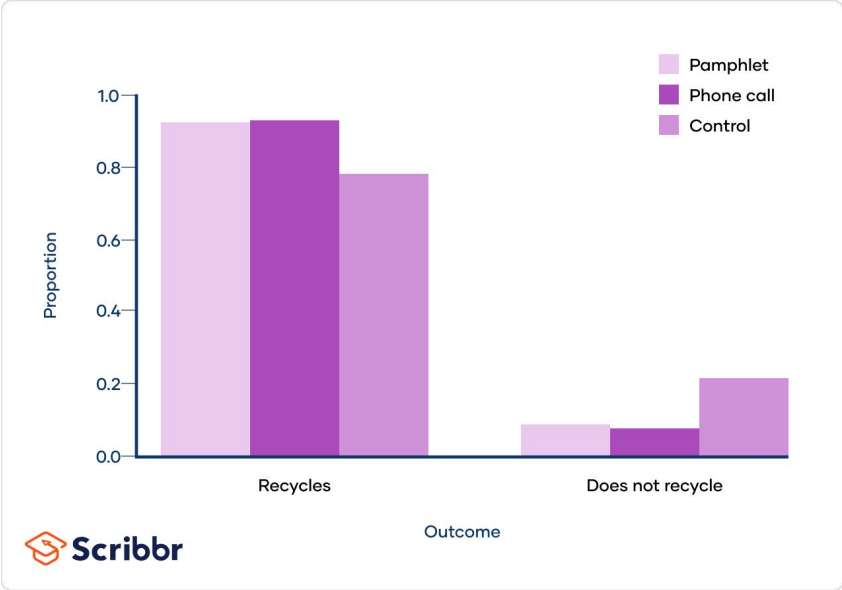
- Expected distribution
- Actual distribution

https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test

Similarity measurement on categorical feature

Categorical data

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



https://www.scribbr.com/statistics/chi-square-test-of-independence/

Drift detection - level of sophistication

Straw-man approach 🍹

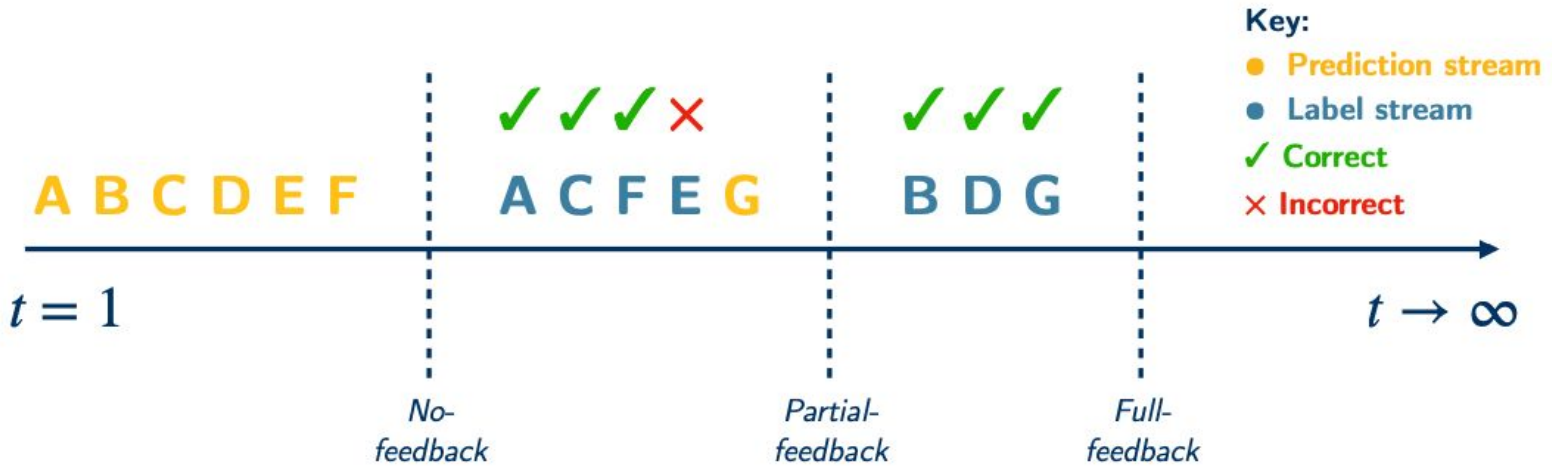
- Tracking means & quantiles of features and outputs

“I know stats” approach 👩‍🎓

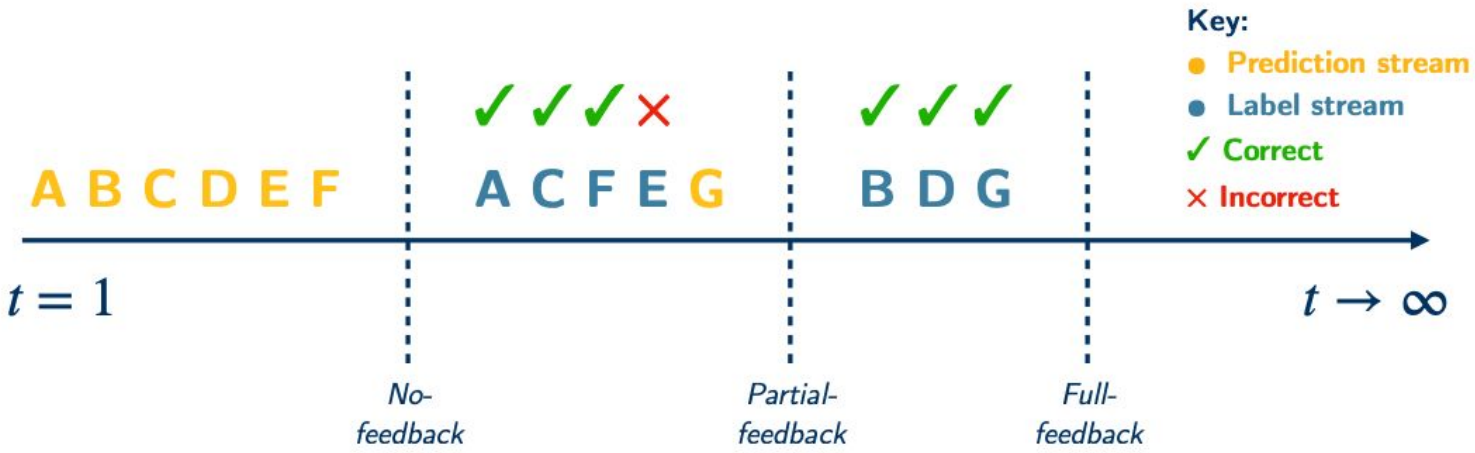
- Tracking MMD, KS & Chi-Square test statistics, etc

⇒ Both don’t require labeled data!

Detecting performance issues: Feedback Delays

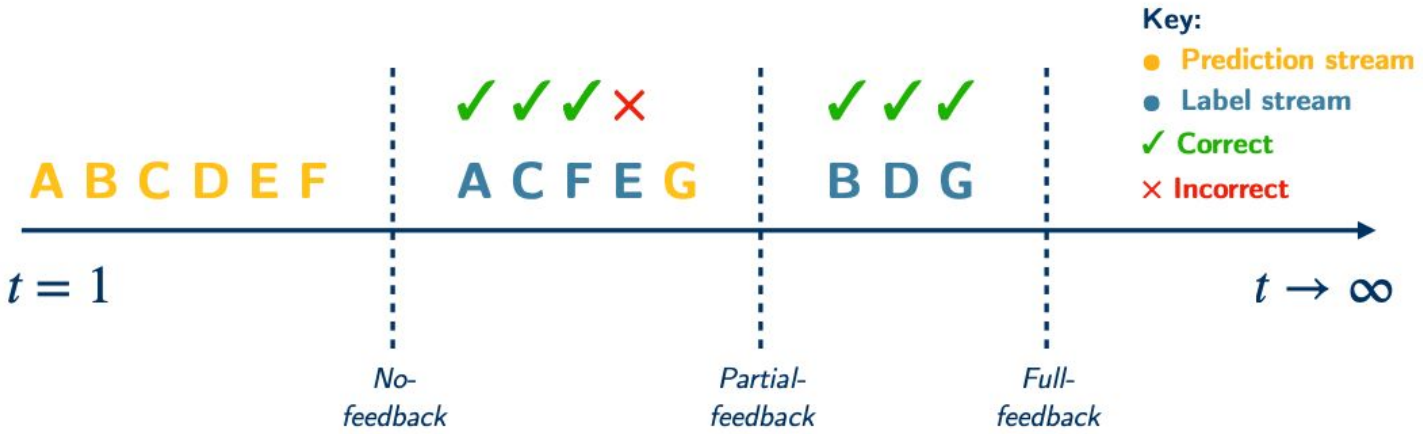


Detecting performance issues: Feedback Delays



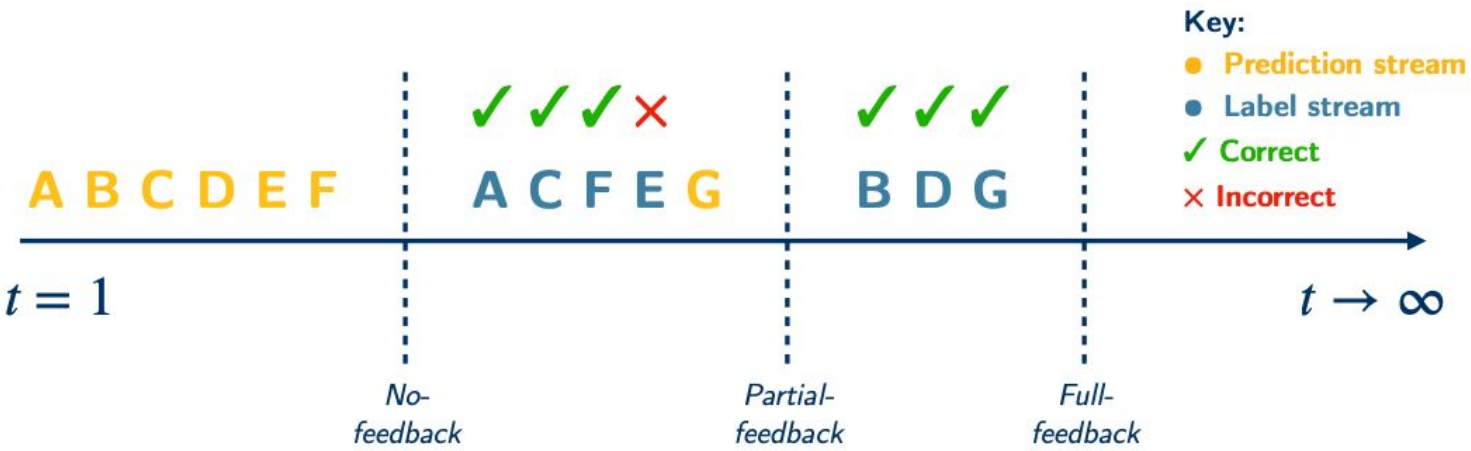
Accuracy: ?? 🤔

Detecting performance issues: Feedback Delays



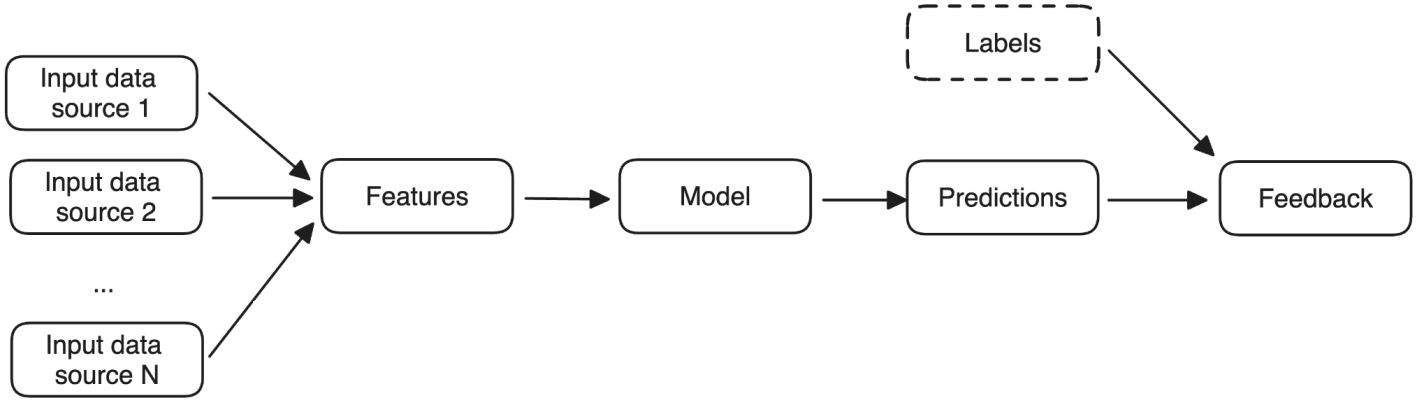
Accuracy: 75% ?? 🤔

Detecting performance issues: Feedback Delays

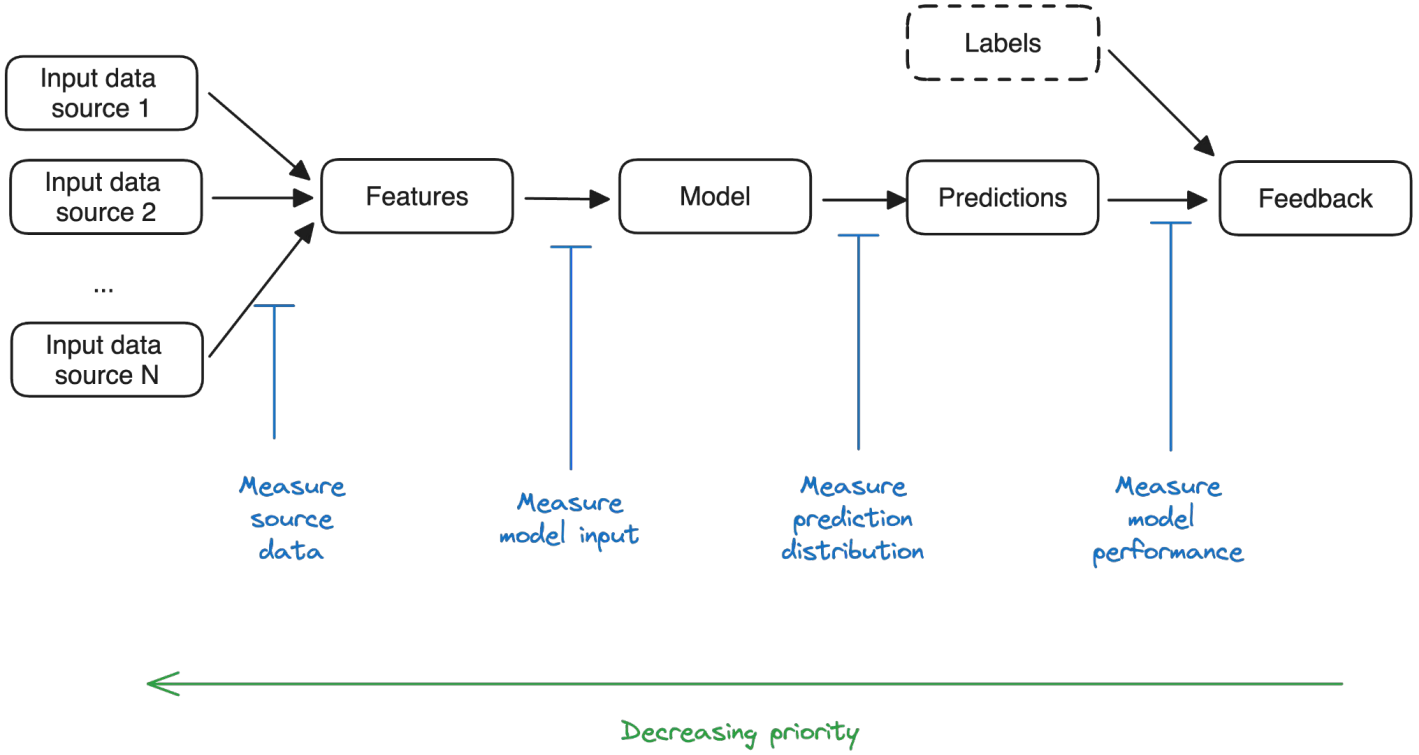


Accuracy: 86% 😀

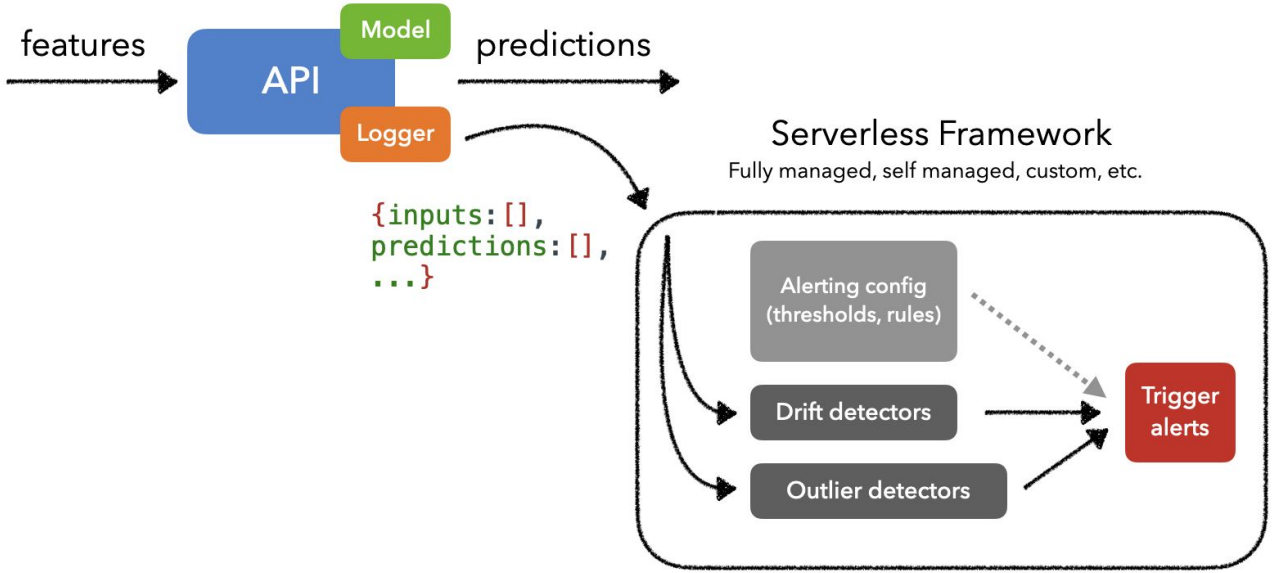
Different level of performance measurements for ML



Different level of performance measurements for ML



You can combine custom drift detection methods into your monitoring system



https://madewithml.com/courses/mlops/monitoring/

Breakout exercise

LIÈGE université

Find drifts and ways of detecting it

Exercise

1. By groups of 3-5 try to answer the following questions
2. 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:

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

Breakout exercise

LIÈGE université

Dashboarding

LIÈGE université


Always has been.
It’s all about data…
ML
imgflip.com


TRUST

innocens

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

UZA

ML6

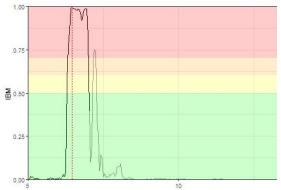
IBM

Universiteit Antwerpen

A trustworthy AI based Clinical Decision Support Platform.

Prerequisites

A Machine Learning Model

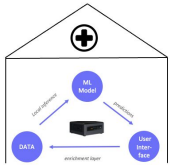


High sensitivity
Detect (most) sepsis episodes

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

innocens

ML probablity LOS

RECALL (sensitivity)
=TP/TP+FN

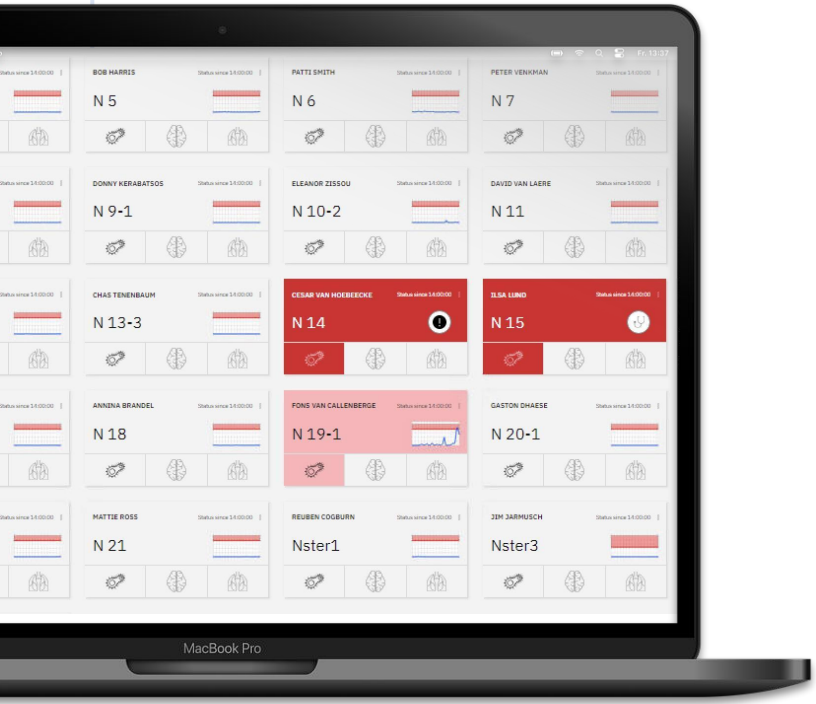
PRECISION (PPV)
=TP/ TP+FP

TRUE POSIT
Any spike < 2
before sepsi

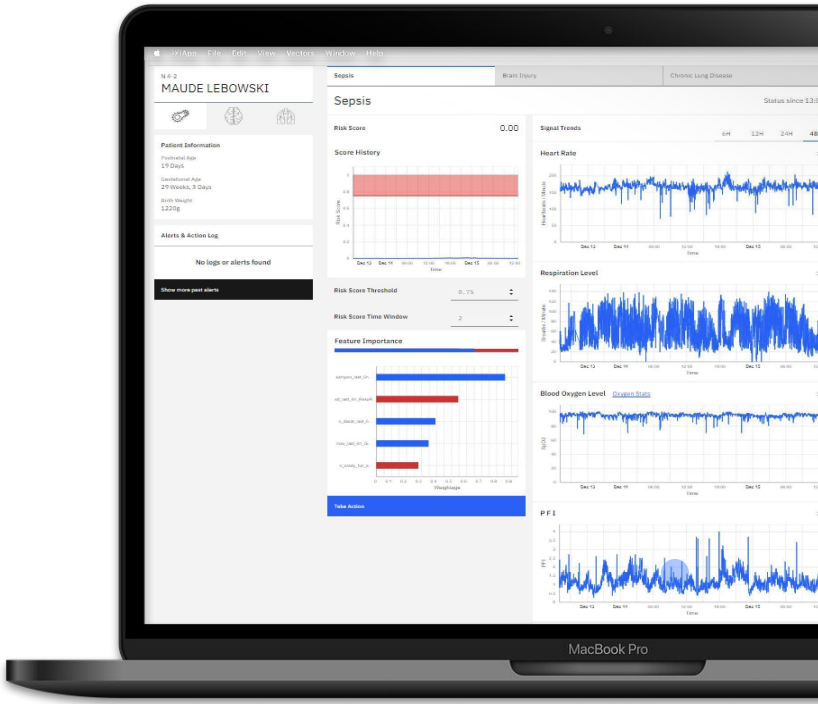
Explainable AI.

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

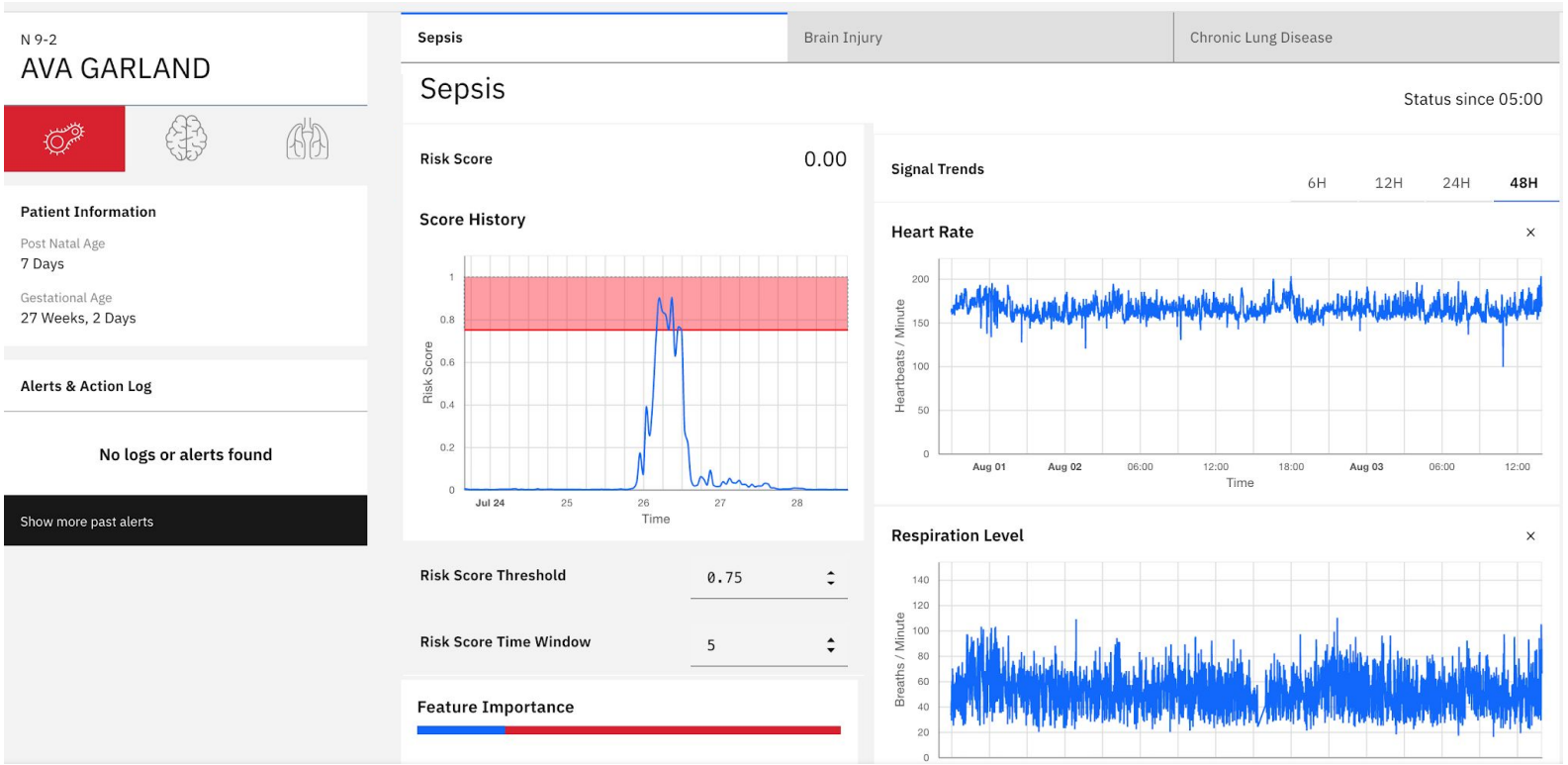
Ward governance at a glance



Feature importance



Innocens in action.



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

ML6

Detecting impact of the covid crisis on job demand

And actually communicating it...

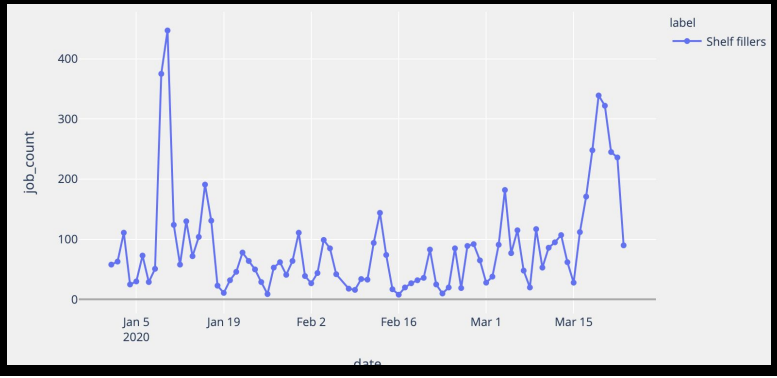
Communicating results: Performance metrics

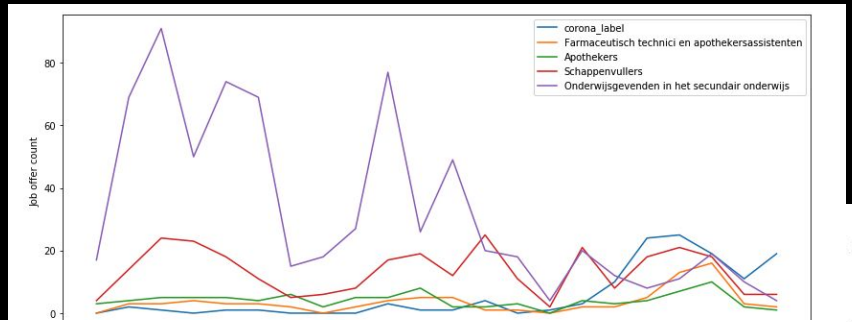
```
Accuracy: 79.67%
F1 score: 83.93%
Precision: 85.33%
Recall: 82.58%

           Negative  Positive
True\Pred
Negative         64        22
Positive         27       128
```

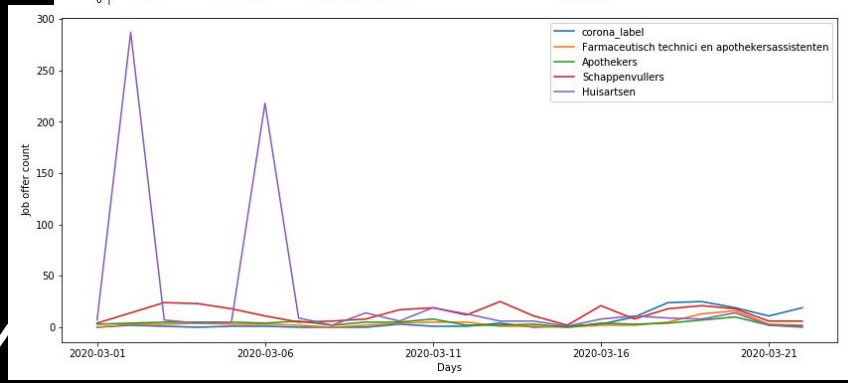

Detecting impact of the covid crisis on job demand

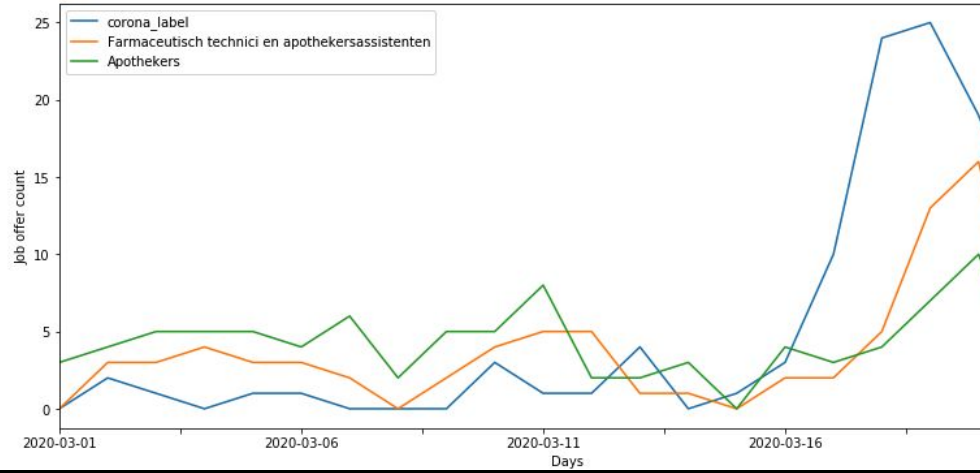
And actually communicating it...

Communicating results: Static plots









Detecting impact of the covid crisis on job demand

And actually communicating it...

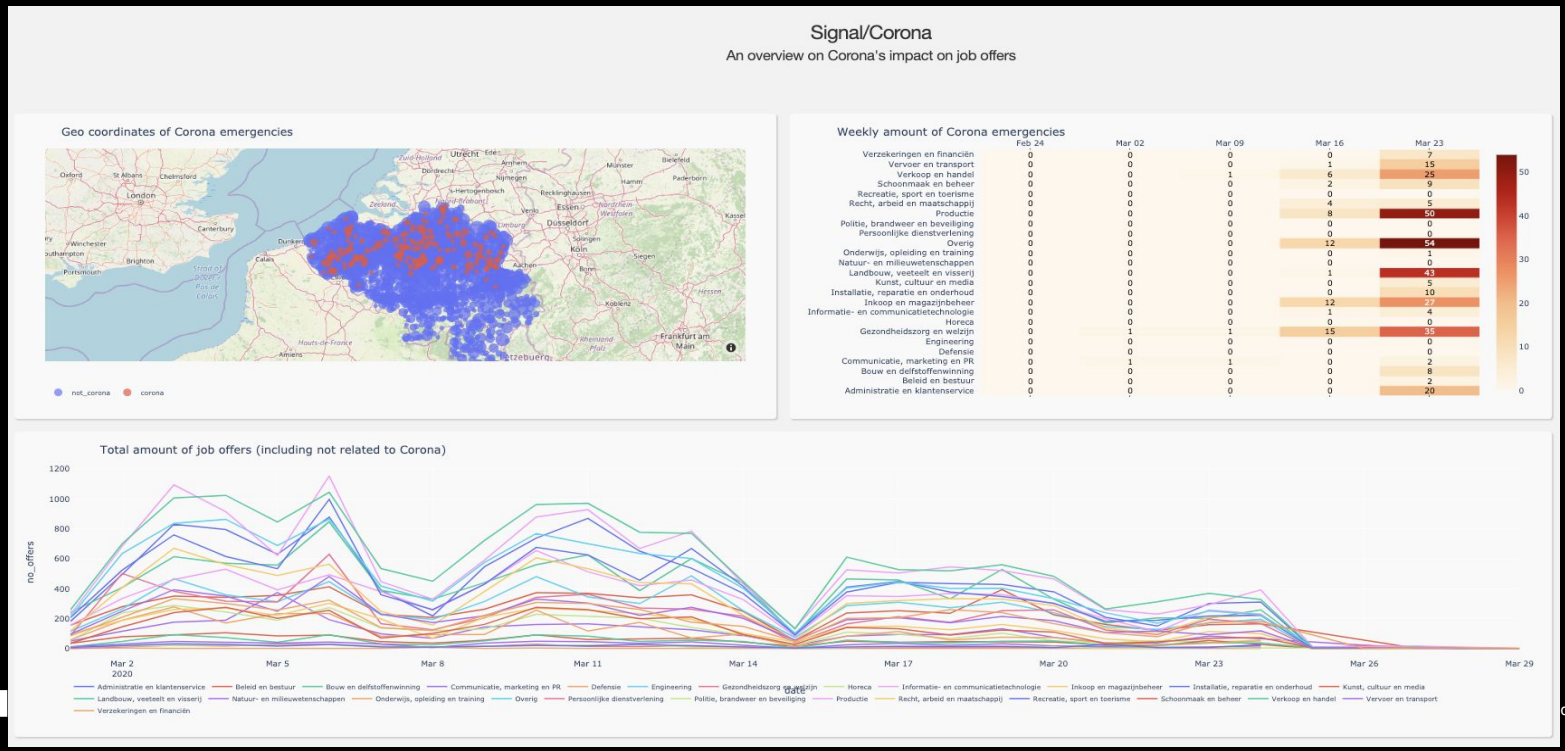
Communicating results: Track per customs

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...

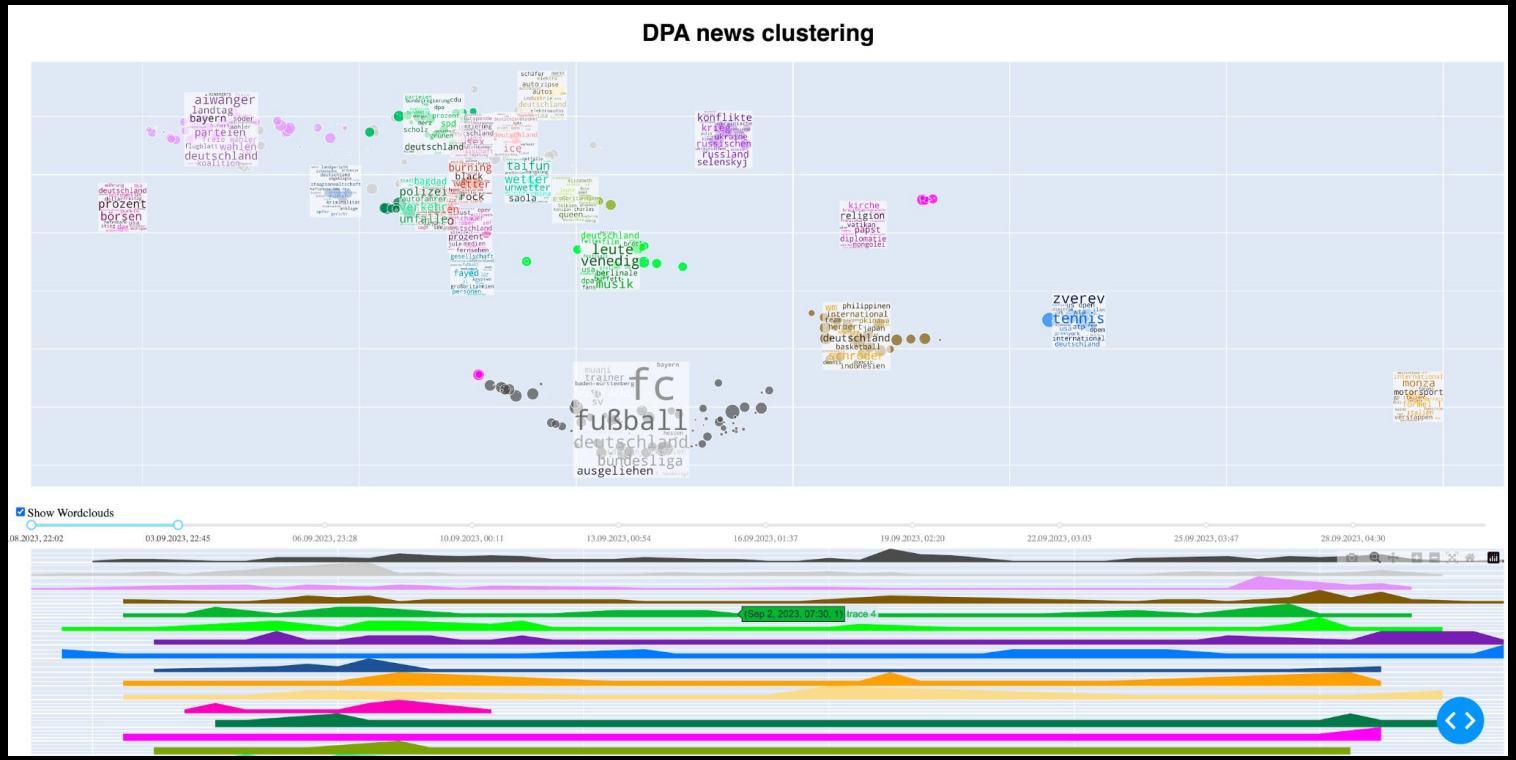
Communicating results: Interactive dashboard



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

LIÈGE université

Wrap-up

Lecture summary

Topic	Concepts	To know for…	
		Project	Exam
Monitoring	• Logging vs monitoring • Why we need monitoring • Resource vs performance monitoring • Model performance monitoring • Drift types: data, target and concept		Yes
Dashboarding	• Importance of dashboarding • Examples		
Lab: Streamlit	• How to use it	(Optional)	

Project objective for sprint 5

You can decide which steps of your CICD pipeline are relevant to implement.

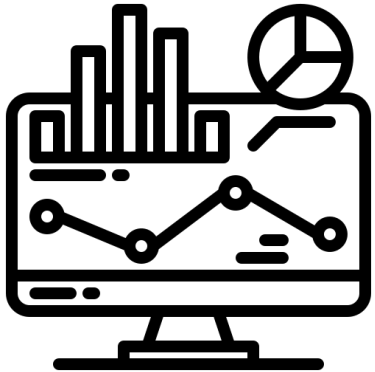
#	Week	Work package	Requirement
5.1	W09	Build a dashboard that runs either locally on in the Cloud to show your results	Optional
5.2	W10	Build a CICD pipeline using Github Actions (or other tool) to automatically run some of the following steps. Include at least one step. The rest is optional. Up to you to decide what is relevant.	Required
5.3	W10	Include step in CICD: Automatically launch model training pipeline	Optional
5.4	W10	Include step in CICD: Automatically launch model deployment	Optional
5.5	W10	Include step in CICD: Pylint	Optional
5.6	W10	Include step in CICD: Pytest for any unit test you think is relevant	Optional

APPENDIX

LIÈGE université

Key concept: Monitoring

Ensuring that models in production are performing well.



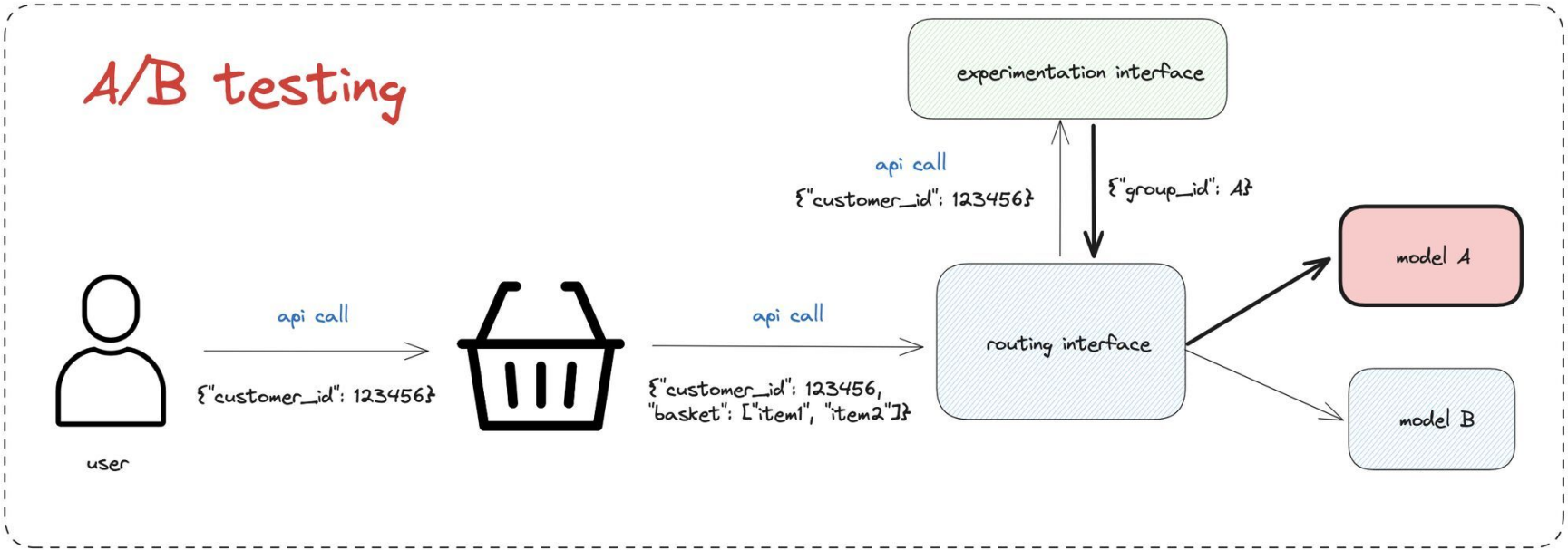
Resource level (performance and usage of resources used by the model serving)

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

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

1. Is the model still doing accurate predictions with the new data coming in?
2. Is the data distribution changing?
3. Is the environment around the model changing?
4. Is it performing as well as during the development phase?

A / B testing


A/B testing
experimentation interface
api call
{"customer_id": 123456}
{"group_id": A}
model A
api call
api call
routing interface
model B
{"customer_id": 123456}
{"customer_id": 123456,
"basket": ["item1", "item2"]}
user

Resource level monitoring: Push vs Pull

Pull

1. Expose metrics from your application via an API, and let your monitoring tool pull it
1. Typically used when your app is going to be a long running process

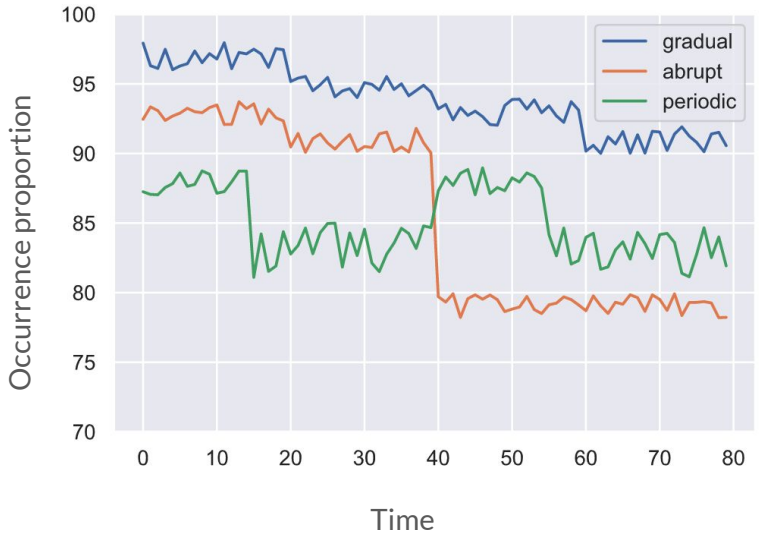
Push

1. Push metrics from your application / CI pipeline to a “push gateway” app
2. You need to run push gateway as a separate service of your application
3. Your monitoring tool will poll the push gateway to fetch metrics
4. Typically used when
    a. You have a short-lived process
    b. The metrics aren’t going to change that often

Concept drift

1. The relationship between the input data and the target has changed
2. The patterns the model learned to map between the original inputs and outputs are no longer relevant

⇒ Importance to retrain models



https://madewithml.com/courses/mlops/monitoring/

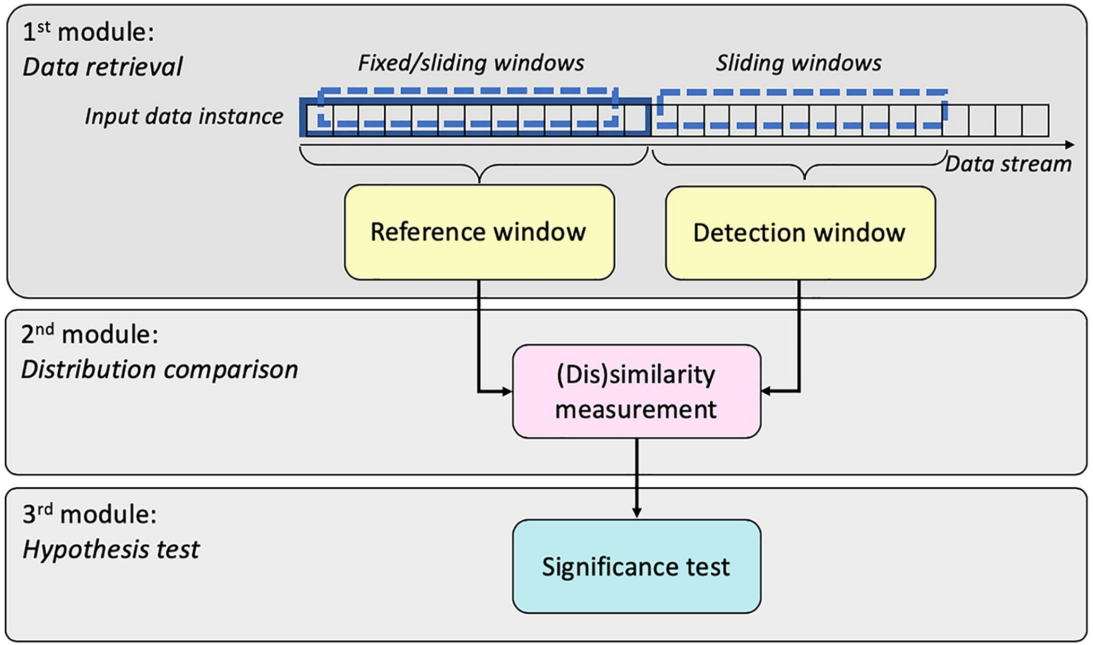
Detecting drift: different methods

Two main families of drift detection:

1. Batch drift detection
2. Online drift detection

https://wires.onlinelibrary.wiley.com/doi/full/10.1002/widm.1381

Online based drift detection



https://wires.onlinelibrary.wiley.com/doi/full/10.1002/widm.1381