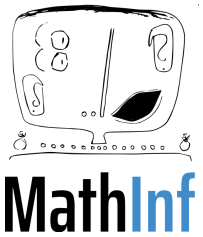

MathInf
Training and Monitoring AI -
Drift Detection
Thomas Viehmann
MathInf GmbH
goto Copenhagen
October 2022

Hello, I’m Thomas!



About me:

PyTorch core dev and at 170 features and bugfixes over 5.5 years as an independent contributor

Co-Author of *Deep Learning with PyTorch*

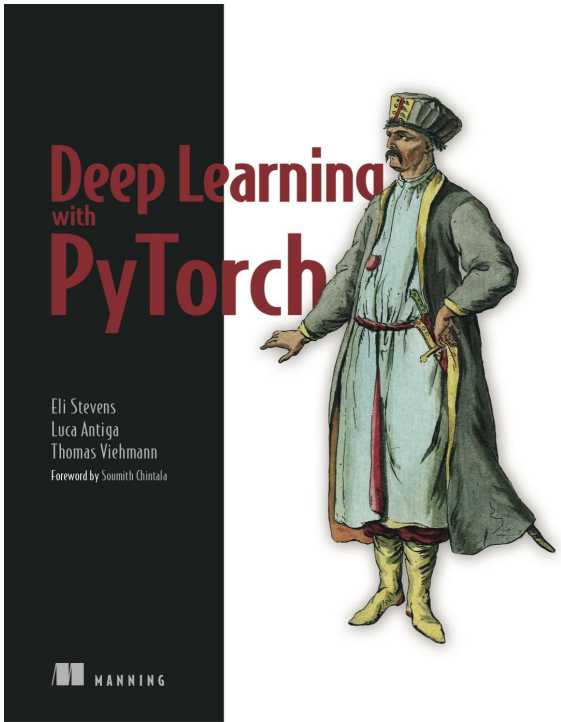
Founded MathInf GmbH in 2018 to do PyTorch Training and Consulting

- integrated prototyping + know-how building
- from slow to fast
- from does not quite work to works
- recently more traditional software startup with lots of drift detection...

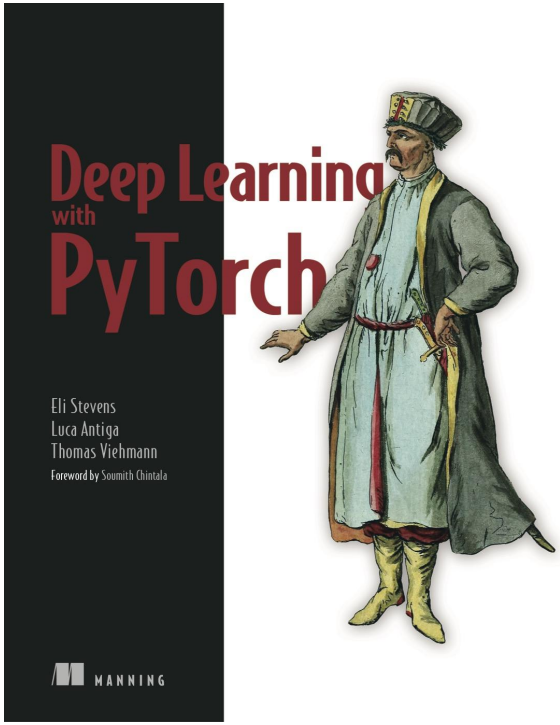
Ph.D. in pen and paper Mathematics

Tech blog: at https://lernapparat.de

Open Source: TorchDrift.org, TVM.Apache.org, some LibreOffice,
Debian Developer Emeritus...



About the book

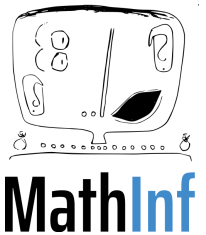


In part 1, we introduce PyTorch and convolutional neural nets. Part 2 takes us through an end-to-end project working with lung CT scans looking for nodules.
Part 3 adds a bit about deployment.

Probably one of the deep learning books with the fewest formulas.

It didn't make the book table, but factumbooks is kind enough to offer a 30% discount for you. Thank you John!
https://factumbooks.dk/?search_string=9781617295263

Agenda



Drift detection as an essential exercise for deployment.

1. Why?
2. How does it work?
3. What do we need to do?

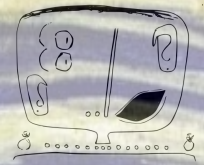
Why detect drift?

Neural network deployments

- Very successful in applications as diverse as medical imaging, areal surveillance, defect detection / quality inspection, autonomous driving…
- Many applications have a strong requirement for accuracy in various shapes (false positives vs. false negatives).
- We don’t typically know whether a neural network works by looking at it. Instead, we need to test, test, test.

Q: Were these really the right inputs to test with?

Things going out of spec...



MathInf

Neural nets work a lot less well on shifted inputs.
Many good reasons to not let that happen!

Safety

In AI assisted medical diagnostics, we might miss important cues.

When using neural nets for steering in control-loops it can be outright dangerous to have wrong inputs.

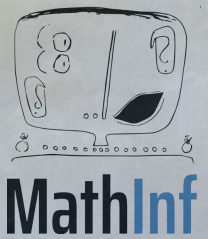
Business

When performance deteriorates, so does the business value.

Compliance

In case things go wrong, people will ask what we did to ensure they're working, e.g. in medicine, traffic or other heavily regulated sectors.

Things going out of spec...



Recommender

We get feedback – but how fast until we notice declining sales performance?

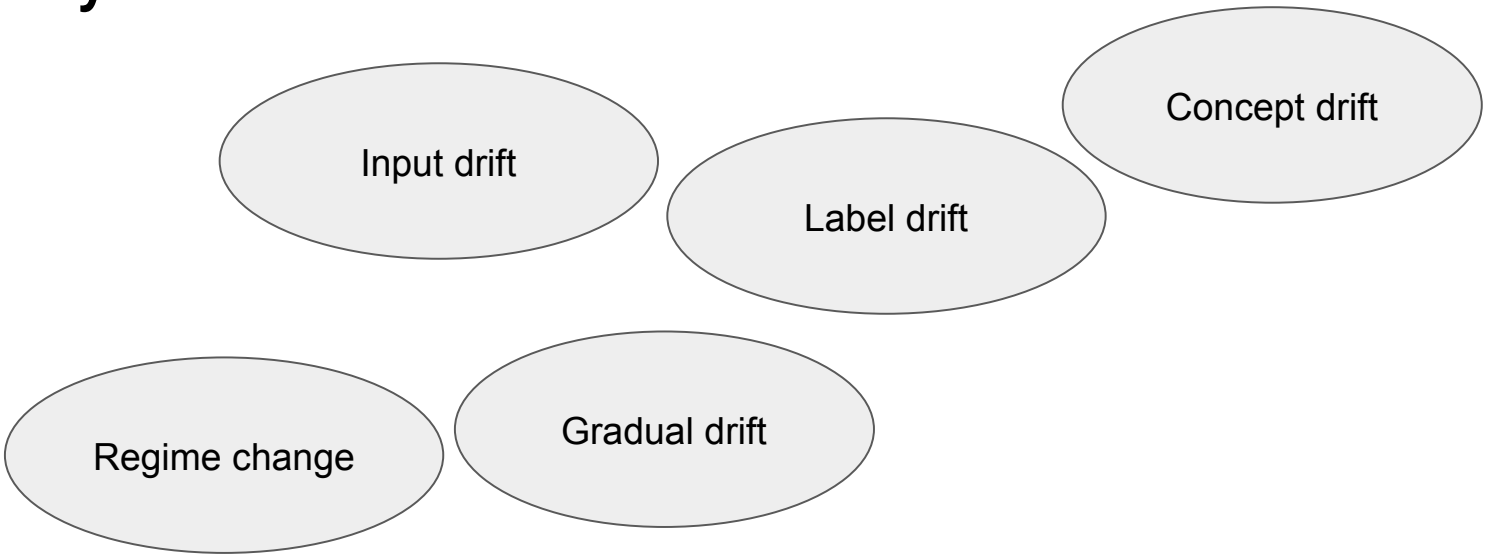
Medical

A new MRT machine or “decoder” (from sensor to voxel images) changes output levels.

Visual Q&A

A differently colored part reduces model sensitivity to defective parts.

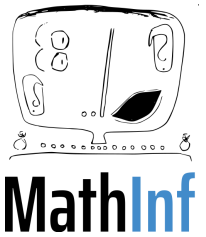
Why and what is Drift?



Lot’s of fancy concepts, but the key question always is:

Do the assurances of the model validation apply in our deployment?

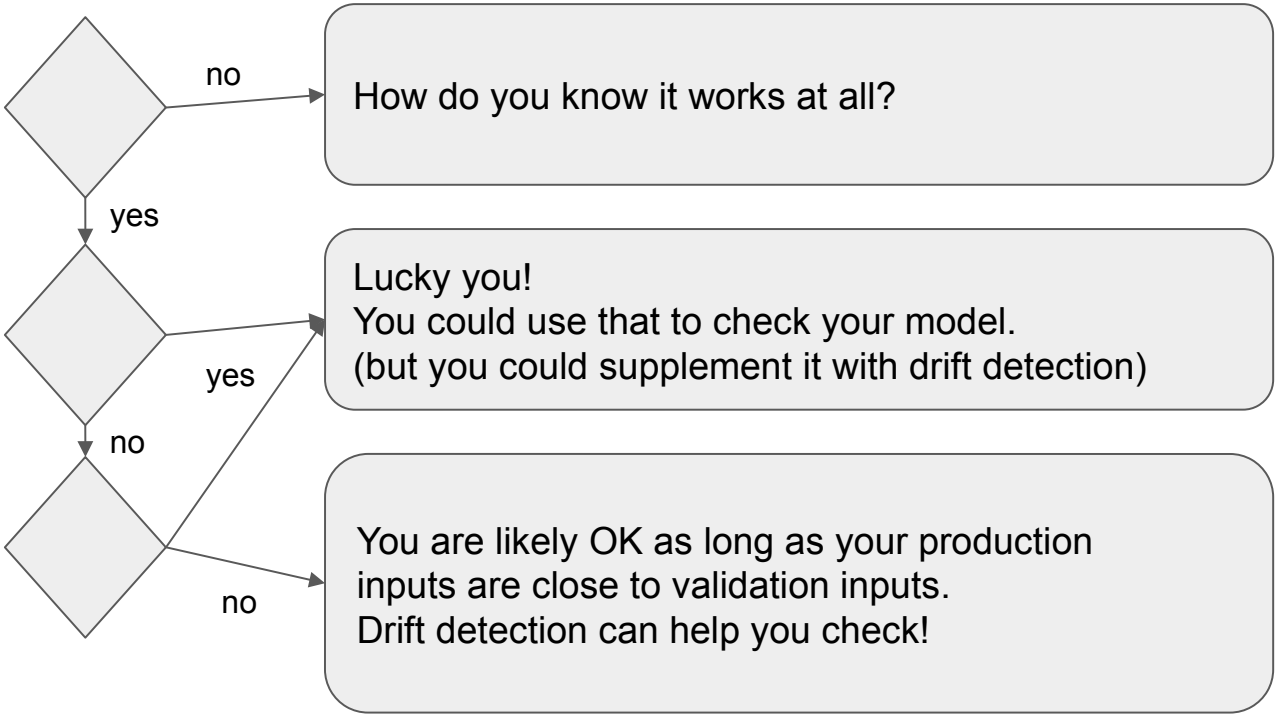
Can you trust your model to work?



Did you validate your model (with early “production” data)?

Do you have labelled data in deployment (cheaply)?

Can your model say “I don’t know” when It does not know? (eg. being Bayesian)

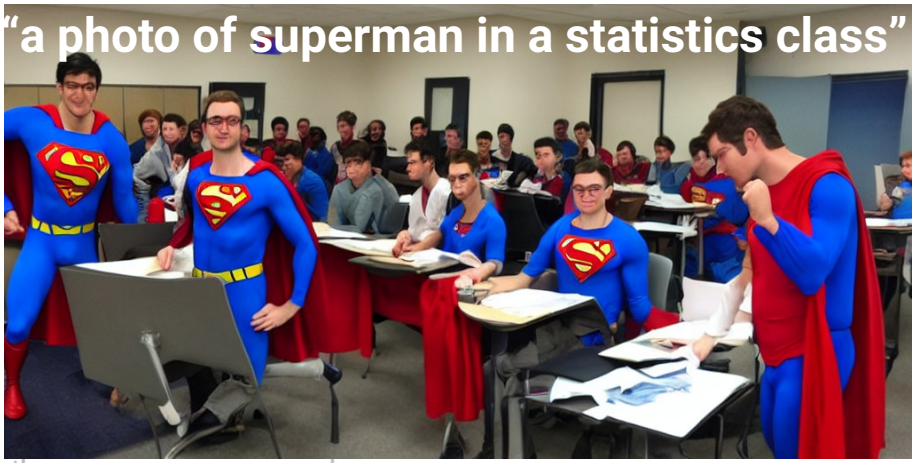


Statistics of detecting drift

“a painting of a sailing boat by claude monet.”

Statistics to the rescue

- In general, we don't have labels in production.
- Use statistical testing at the core
- (Adapted) two-sample variant most relevant: "could the production data and the reference data be samples of the same distribution"
- fancy kernel-based non-parametric methods



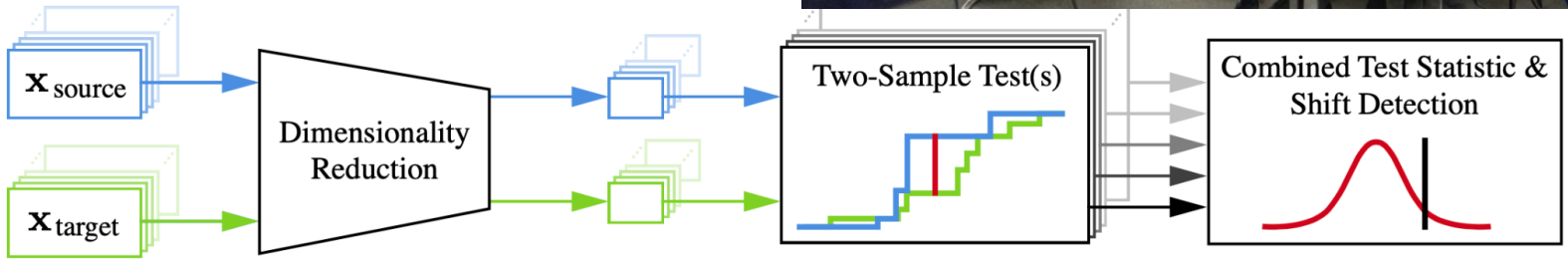
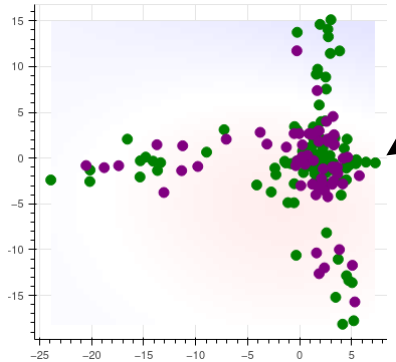


Diagram: S. Rabanser et al: Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift

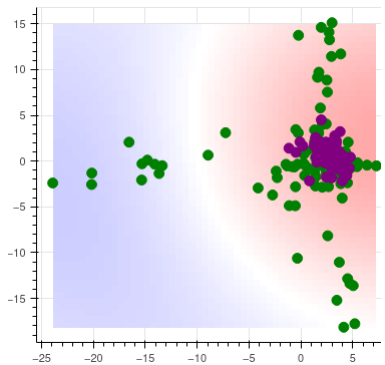
Outlier detection is not enough

- Outlier detection more well-established task, maybe you are doing it already. But...
- OD has weaker statistical power than i.e. later detection of shifts both methods see
- Some important types of shift (features becoming more “bland”) are not be detectable at all by looking at individual samples



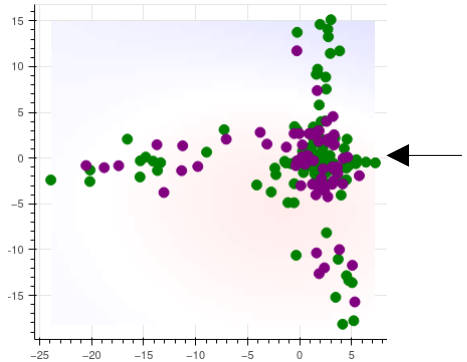
Looks normal (green similar to purple)

The clustering looks suspicious, even if each point individually is well within the expected range

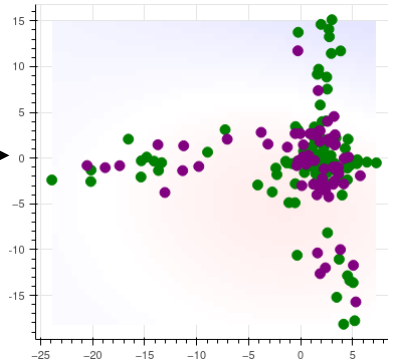


We do not do adversarial detection

- Adversarial detection likely interesting on the single-sample case (so more like OD)
- Much harder, as the inputs are deliberately close to “good” inputs
- In many applications (industrial, medical), adversarial inputs are not part of the “threat model



The essence of adversarial perturbations is that they’re so small that they are not noticeable.

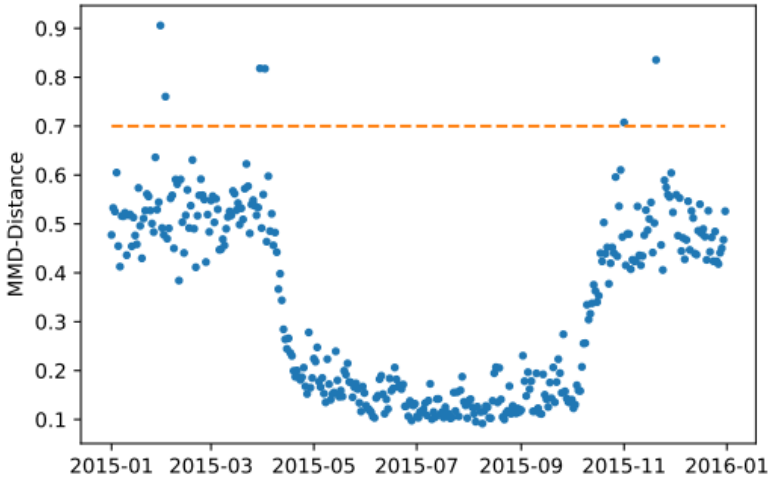


Expected fluctuations

Often there are expected fluctuations

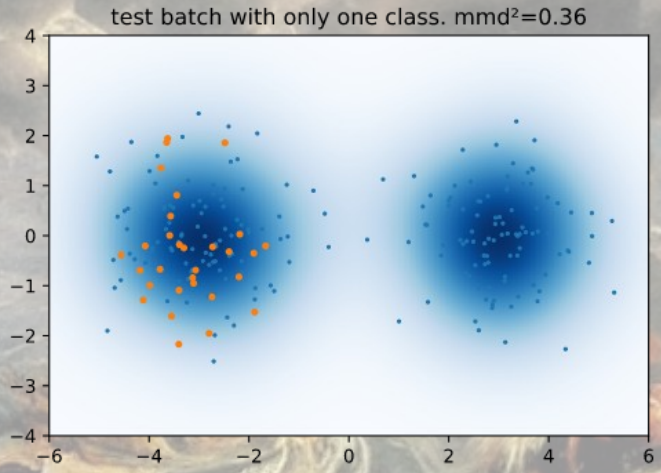
- Lighting over time of day / season
- Batches of different parts at the same QA station, …

Bottom: Real example from the vessel detection calibrated on a summer day.
People take the boats out in winter (not a malfunction of the detector...).



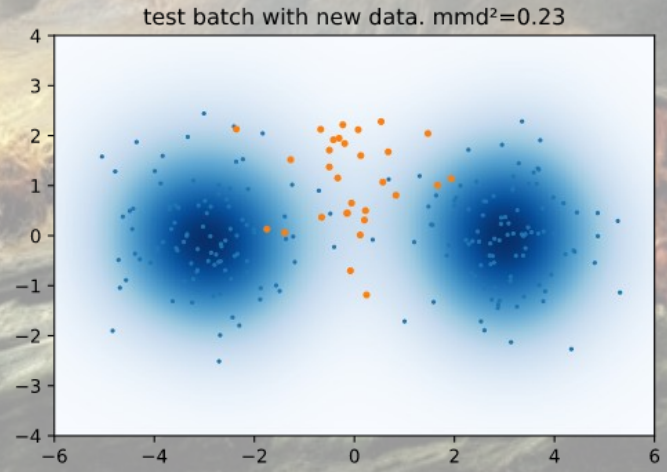
How to deal with expected fluctuations?

...in a toy two-mode model. What if it is OK to only see data from one of the modes?



Conventional answer: Don't sound the alarm at 0.36 then...

...but that means we'll miss legitimately drifted data!

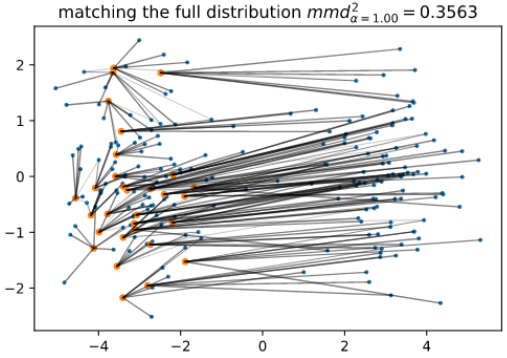


How to deal with expected fluctuations?

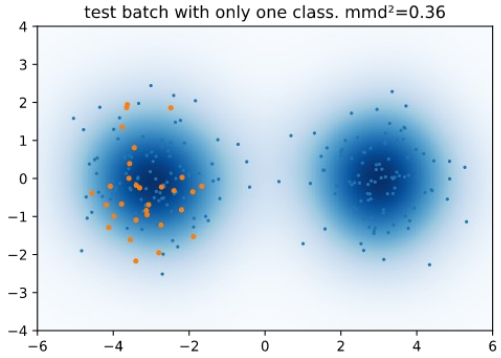
...good opportunity for some statistics nerd-out.

- Plain two-sample testing is “could the production data be a representative sample of the reference data distribution”
- Outlier detection is “could the production datapoint be a reasonable draw of the reference data distribution”

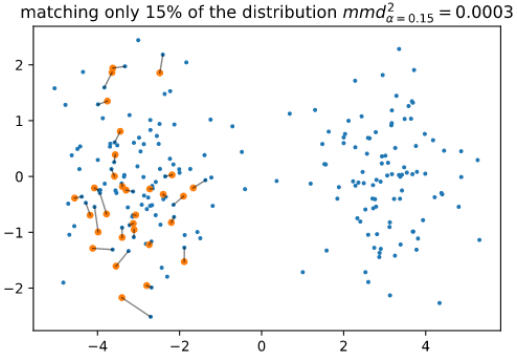
Idea: Interpolate between the two: “could the production data be a representative sample **of part** of the reference data distribution”



Match all



Match part



Practical drift detection

Drift detection challenges in practice

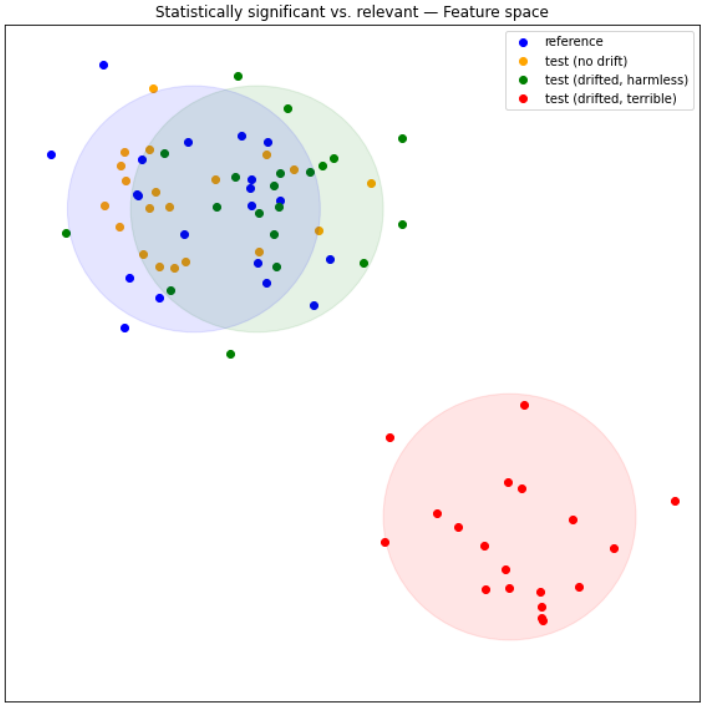
Which features to use?

- Model blindness vs. extracting relevant bits
- Curse of Dimensionality

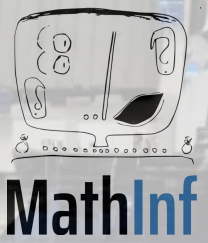
When to sound the alarm?

- Sensitivity-Specificity trade-off when you test many times.
- Region in which the model does well might be different to what statistical testing identifies.

How to integrate into the operation?



Operationalization – two step process



1. **Baselining** (fit reference set)
   - Either as a preparatory step or
   - directly from production
   - configure “alarms”
2. **Monitoring** (run during normal operation)
   - Independently (on input / intermediates / output)
   - From a hook with separate reporting / alerting
   - Integrated into the deployed model with an additional output processed by the model output’s consumer

What if the alarm goes off?

Or, in fancy speak, how to embed into governance.

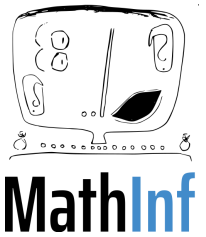
All is good.
Can take the day off.

Purely informational:
Wait until someone sees the yellow light came on.

Actually alert people:
So that the model will be looked at in a timely manner.

Stop the machines:
Don’t use the model until it has been inspected.

Where to run the drift detection?



Out-of-band: get drift feedback via dashboard, hooks etc.

In-band: get drift feedback with prediction

Drift detection as a service

- Quickest to deploy.
- Easiest to get help with calibration etc.
- But would you send your data to a service?

Drift detection on your cloud / cloud provider

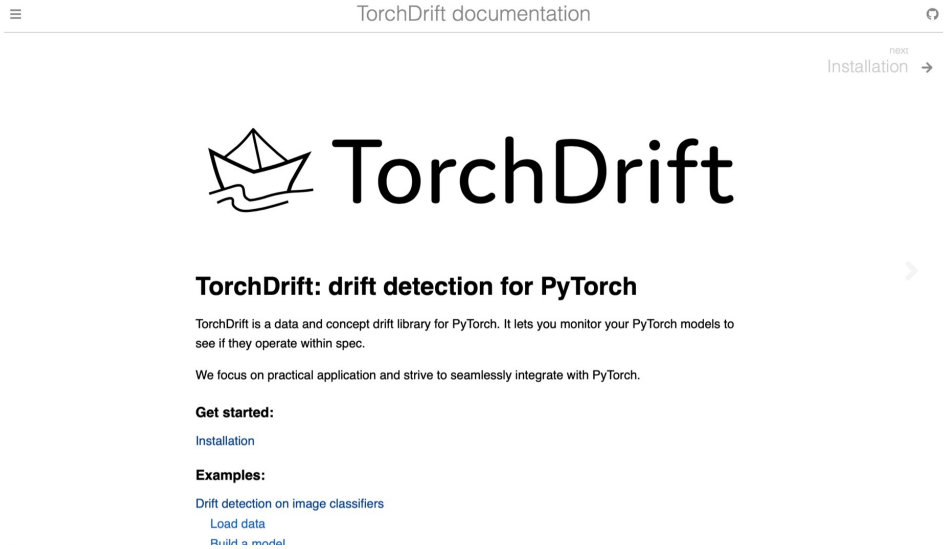
- Integrates nicely with your other monitoring.
- If you want to monitor models running in different locations.

Drift detection on premise / on the edge

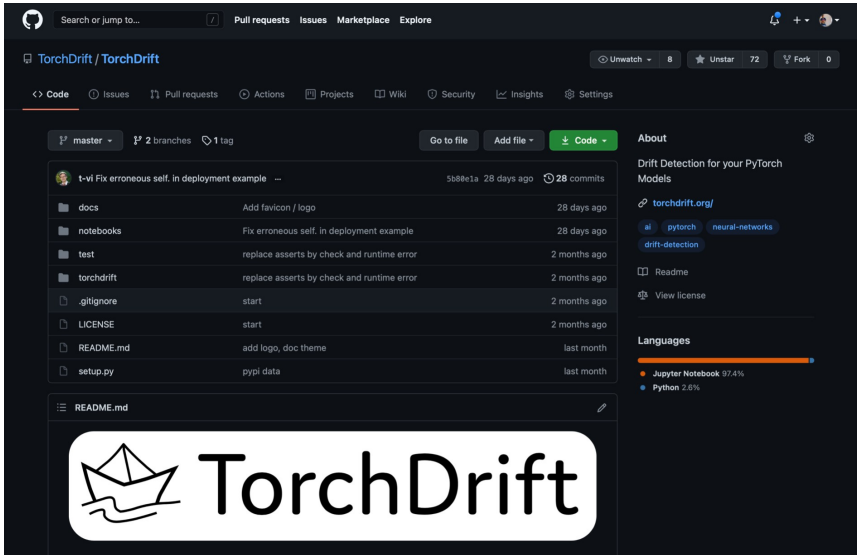
- If you need to keep your data close to yourself.
- If you don’t have connectivity.
- If you want immediate feedback, e.g. for “stop the machines”.

TorchDrift: open source drift detection for PyTorch

MathInf GmbH and Orobix s.r.l.



https://torchdrift.org



https://github.com/torchdrift/torchdrift

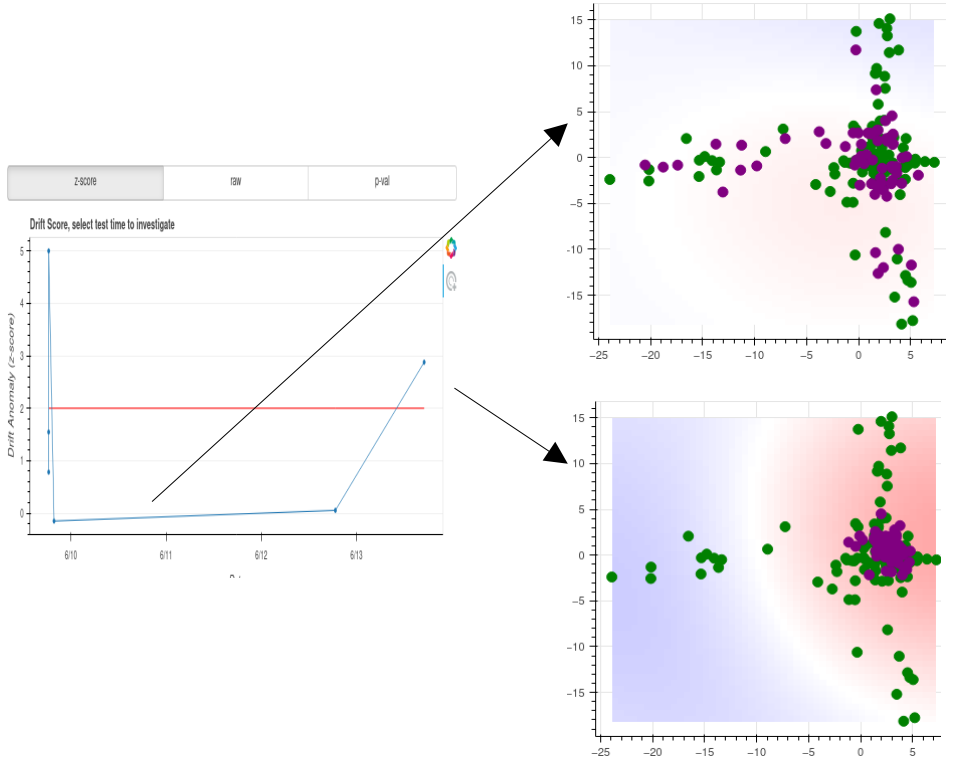
Friendly nod to Seldon Alibi Detect which does a similar thing

DriftDash by MathInf

A one-stop solution for drift detection

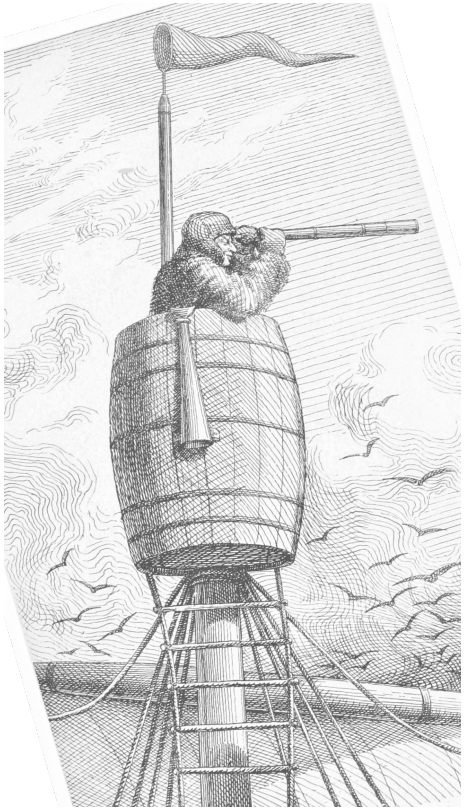
- Standardizes calibration process
- Framework agnostic though HTTP API
- Cloud-based or on prem / on device
- Interactive Dash-Board for Investigating Drift in Time and Feature-Space
- Direct Alerts
- Grafana Integration
- Reports (soon)

https://driftdash.de



Conclusion

Conclusion and outlook



Drift detection provides evidence that your validation assurances still apply

Drift detection is an essential part of deployment best practice

Drift Detection likely gets more important as AI deployments grow

Still a bit too much “art” in there (e.g. what to detect drift on), but lots of progress, too.

Thank you!

Your questions and comments

Contact:
MathInf GmbH
Thomas Viehmann
Gereonstraße 14
48145 Münster
tv@mathinf.eu

(from Graves style handwriting generation)
Thanks to stable diffusion for the images:
https://github.com/CompVis/stable-diffusion