THERAPIXEL

TECHNOLOGY. FOR LIFE. FOR ALL.

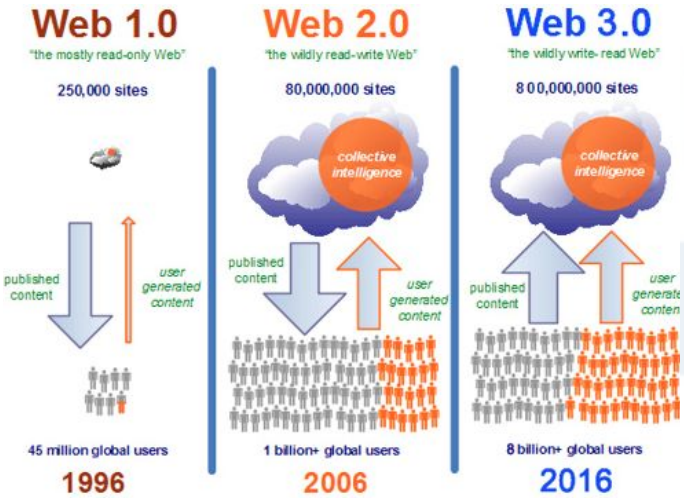
ML/AI applications as software product

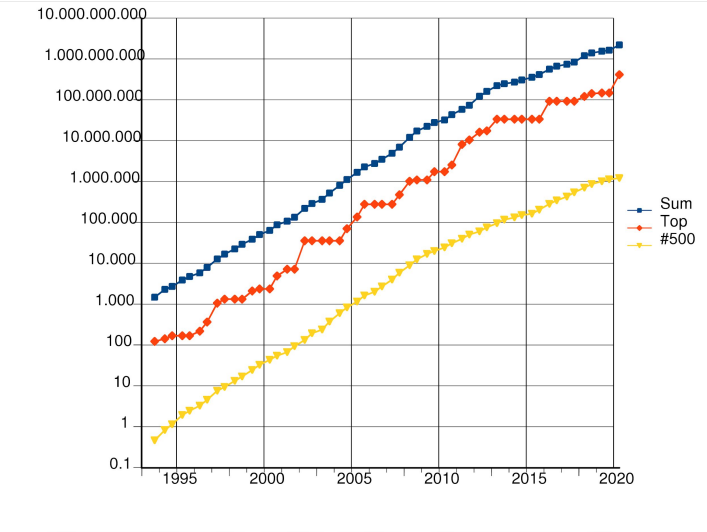
February 22th, 2021

Special for MVA

by Yaroslav Nikulin @ Therapixel AI team

How did we arrive here?



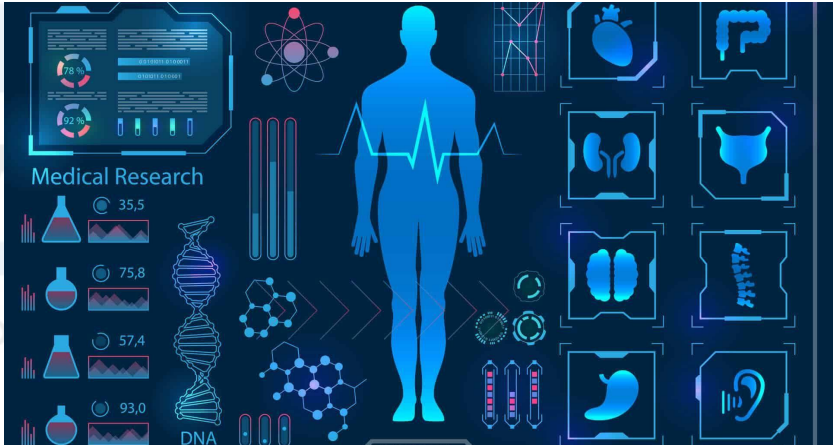


- Web 1.0 = a few content creators, a lot of content consumers
- Web 2.0 = a lot of peer content creators, wildly read-write Web
- Web 3.0 = Web 2.0 + scale + ML/AI

- Top500 = the most powerful supercomputers of the world, updated twice per year
- NVidia V100 = 10^5 GFLOPs

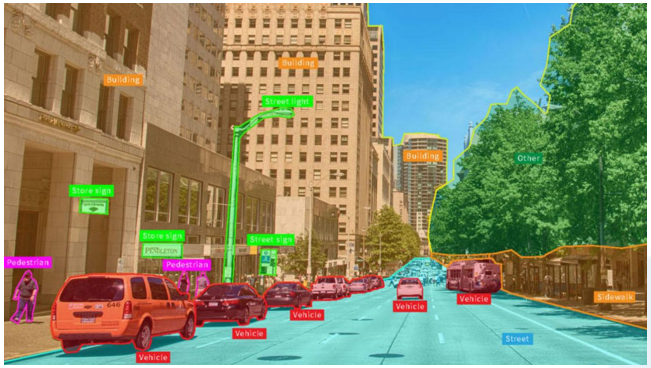
Industrial race for the best AI

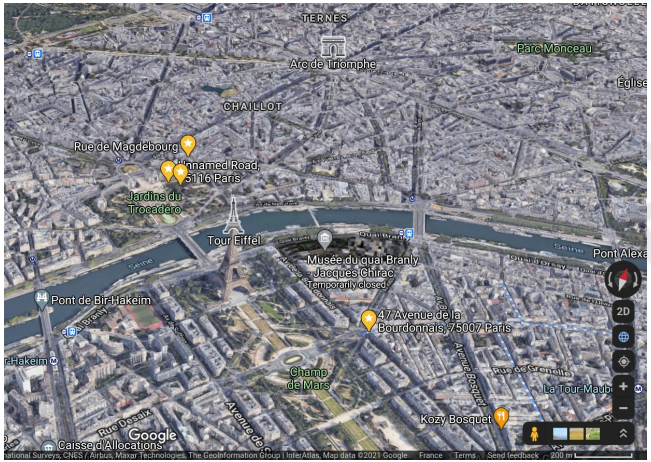
Image credit: viatech.com

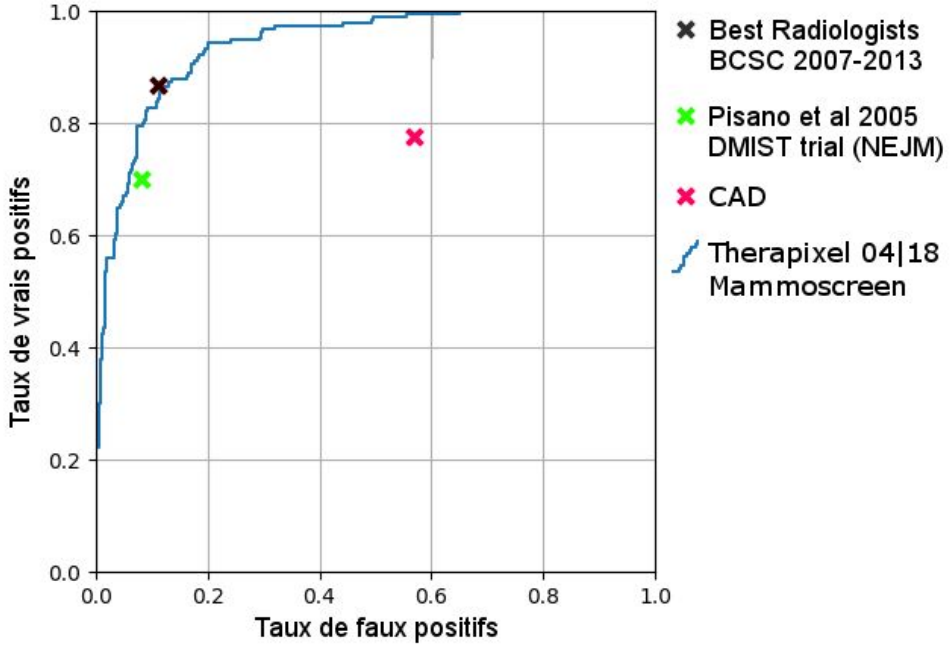


- Different market players have different advantages
- Current general AI = well functioning AI teams
- How many applications can object detection have?

AI as smooth interaction with data

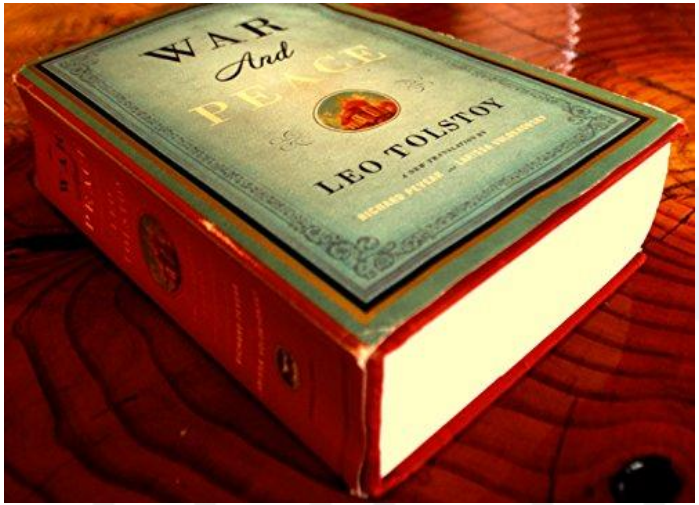


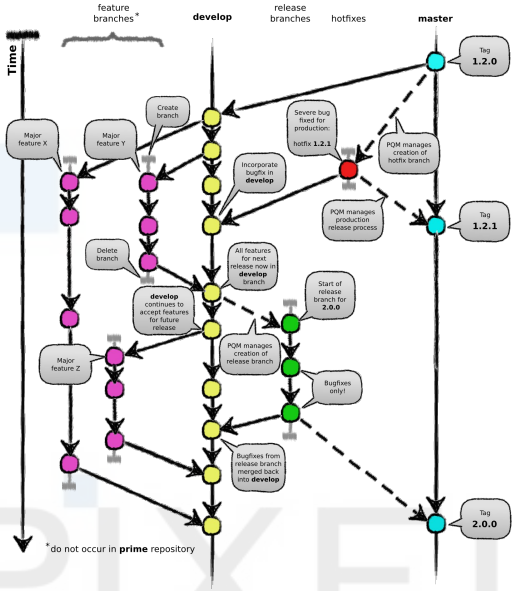




- How to smoothly change the recall rate at hospitals?
- With AI one just needs to tweak some high-level parameters

AI decomposed as a software product

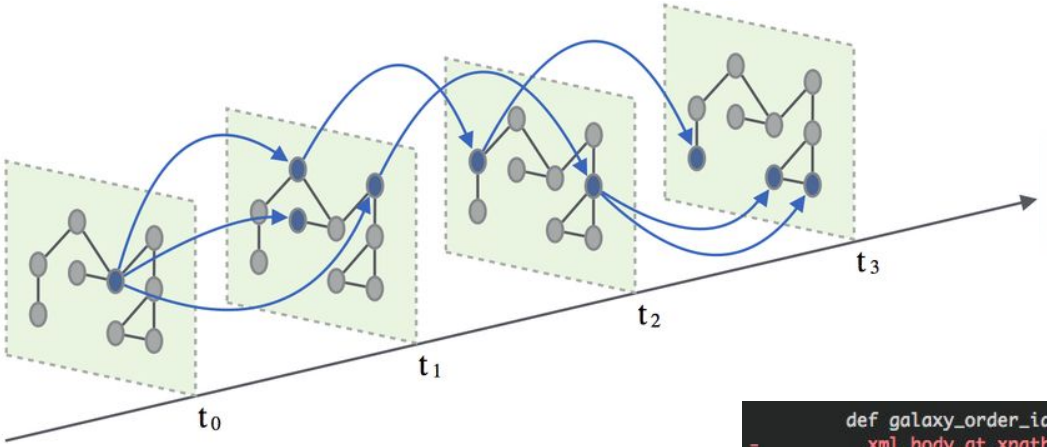




- War and Peace ~ 587k English words
- 587k English words ~ 3M characters
- Each action = new/modified text (usually a lot)

- Source versioning tools
- Master branch, dev branch, etc.
- Actions on branches: merge/rebase

Development tools and methodology = product spine



- Git
- Unit tests
- Jira/Redmine
- Code review

```
        def galaxy_order_id
-         xml_body.at_xpath('GalaxyOrderID').try(:text).try(:to_i)
+         xml_body.at_xpath('GalaxyOrderID')&.text&.to_i
        end

        def errors
@@ -42,8 +46,8 @@ module Egalaxy
        class Error
          def self.from_xml(xml)
            new(
-             code: xml.at_xpath('ErrorCode').try(:text).try(:to_i),
-             text: xml.at_xpath('ErrorText').try(:text)
+             code: xml.at_xpath('ErrorCode')&.text&.to_i,
+             text: xml.at_xpath('ErrorText')&.text
            )
          end
```

AI projects specificities: (new incoming) data

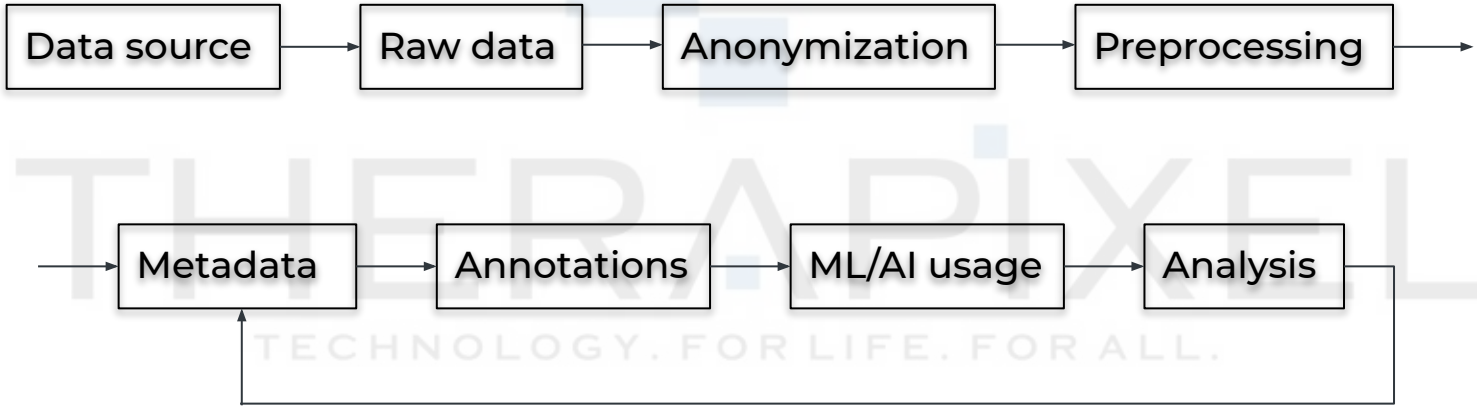
```
trainset = torchvision.datasets.CIFAR10(root='./data', train=True,
                                        download=True, transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=4,
                                          shuffle=True, num_workers=2)

testset = torchvision.datasets.CIFAR10(root='./data', train=False,
                                       download=True, transform=transform)
testloader = torch.utils.data.DataLoader(testset, batch_size=4,
                                         shuffle=False, num_workers=2)

classes = ('plane', 'car', 'bird', 'cat',
           'deer', 'dog', 'frog', 'horse', 'ship', 'truck')
```

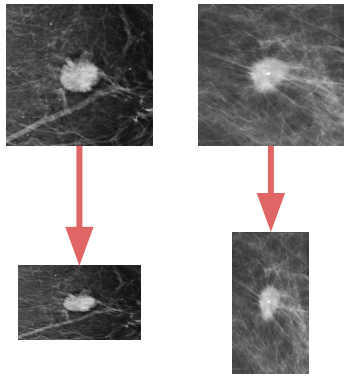
New roles:

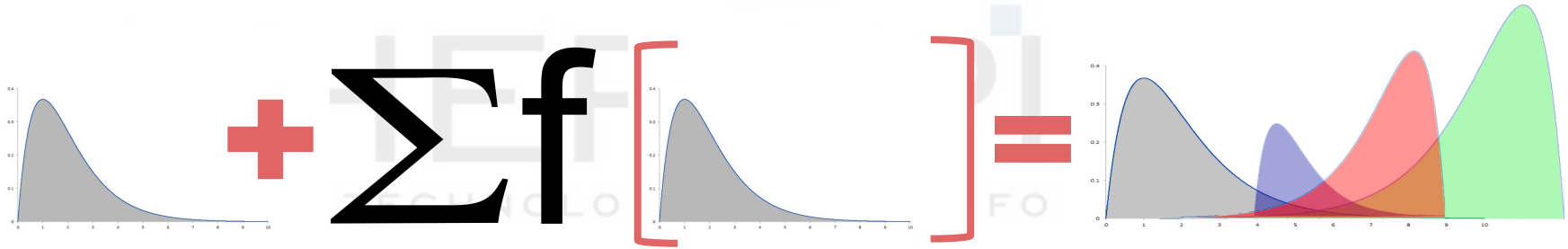
- Data engineer
- Data architect
- Data manager
- Chief Data Officer



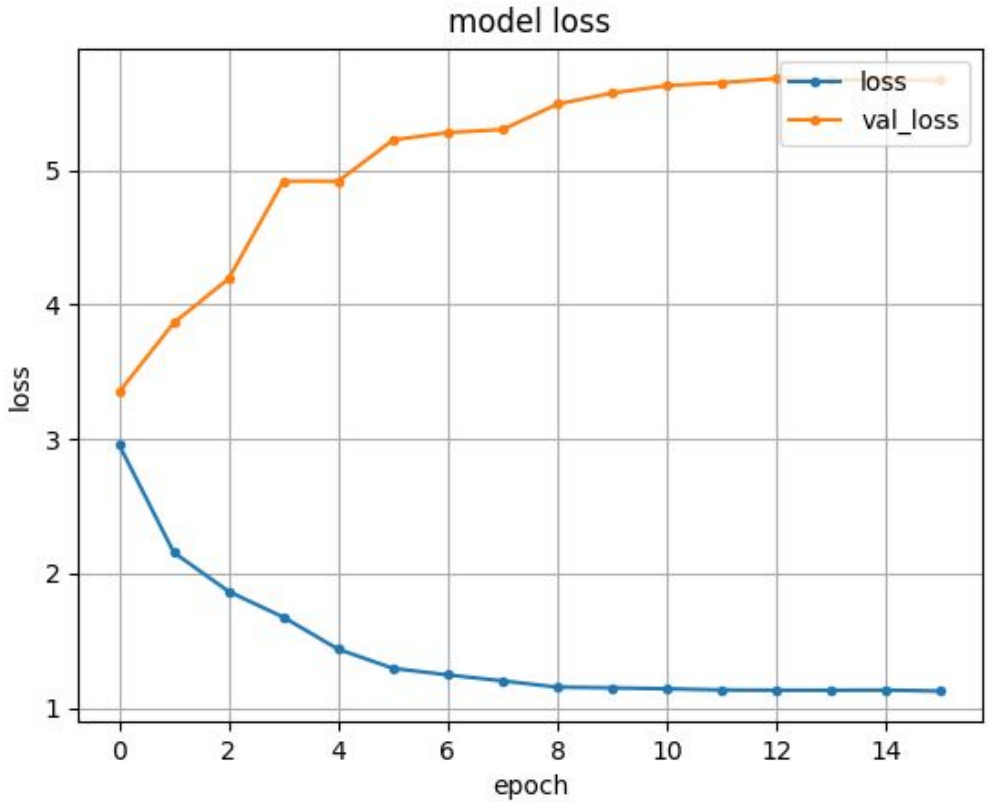
AI projects specificities: see data as (a very complex) distribution

- All breast lesions in the world form a (very complex) distribution
- Eliminate everything not linked to natural variability
- Example: device1 -> 3600x2400, device2 -> 3600x3600
- When both image sets rescaled to 1200x800 -> 2 modes





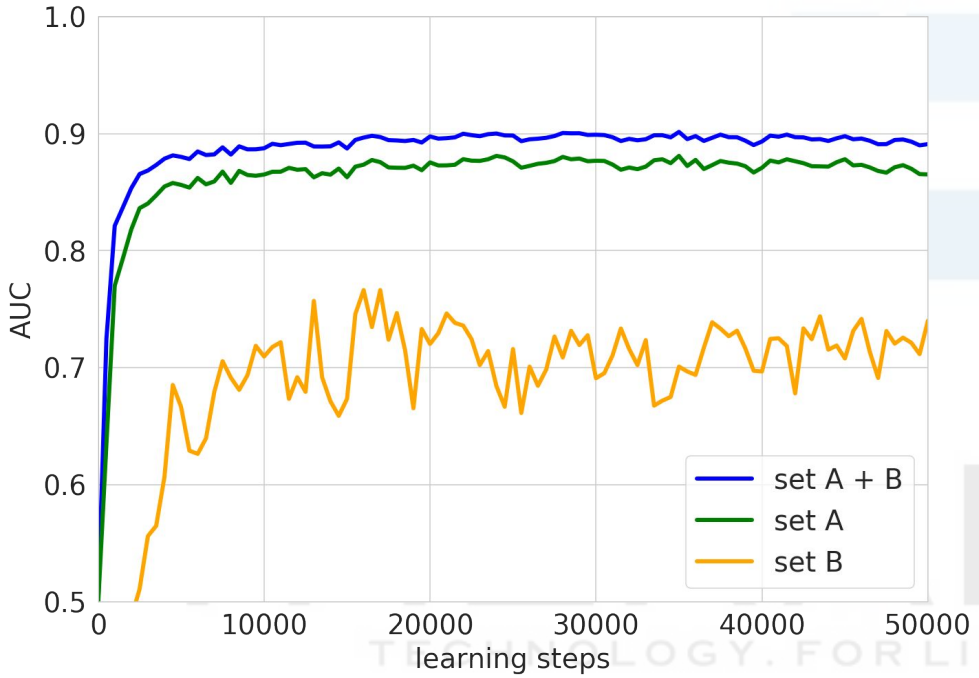
AI projects specificities: ML aspects



- “Classical” software bugs - usually evident failure
- ML/AI bugs - all ok in terms of code, but nothing works
- Machine epsilon + Randomness
- Default values can change ! Avoid blind usage of them.

```
>>> 7./3 - 4./3 - 1
2.220446049250313e-16
```

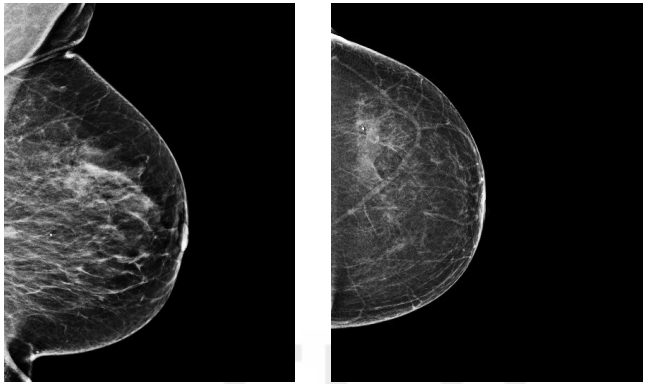
AI projects specificities: Metrics



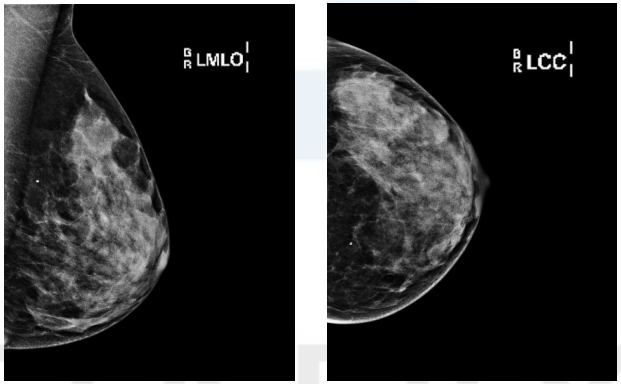
- Global stats are good, but we need to go deeper
- Analyze different sub Distributions
- Fairness metrics to eliminate bias

AI projects specificities: different biases

Screening center
(1% of malignant images)



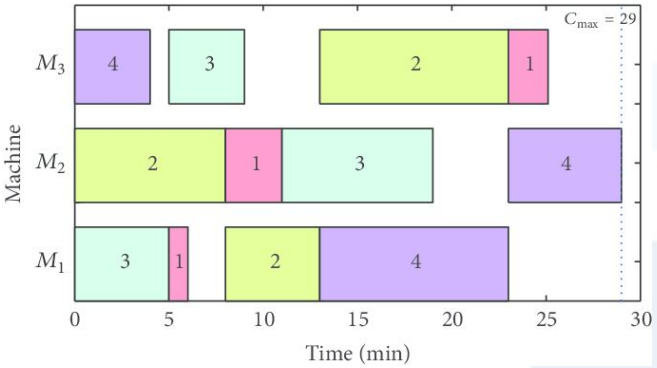
Diagnostic center
(50% of malignant images)



- Biases can be of very different nature
- Biases create wrong correlations
- Biases are most dangerous when correlating with prediction target
- Learnt biases can be detected via fairness metrics

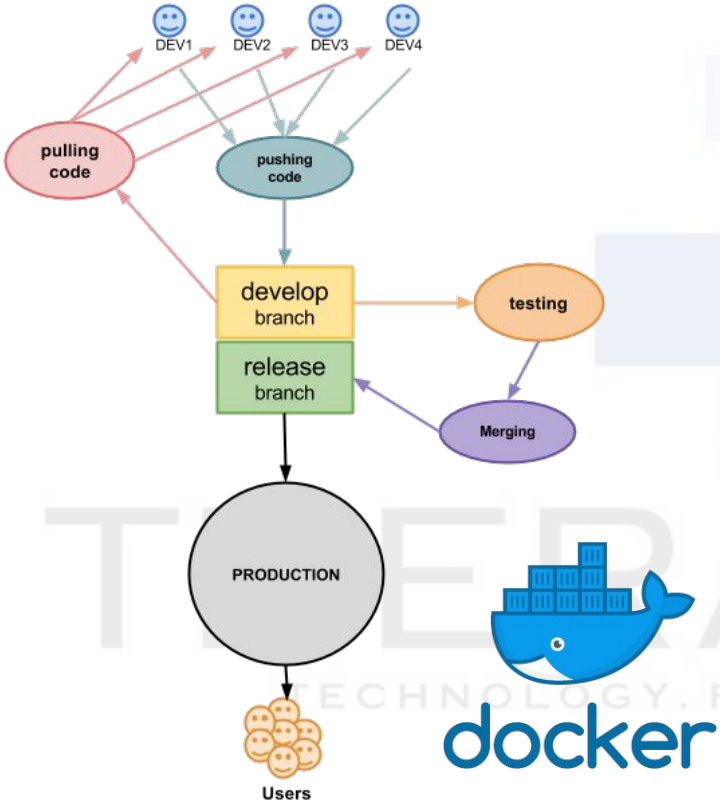
read about how new type of data (high-res mammography) can be solved by DL (DREAM Challenges Therapixel won): write-up link.

AI projects specificities: HPC and long training times



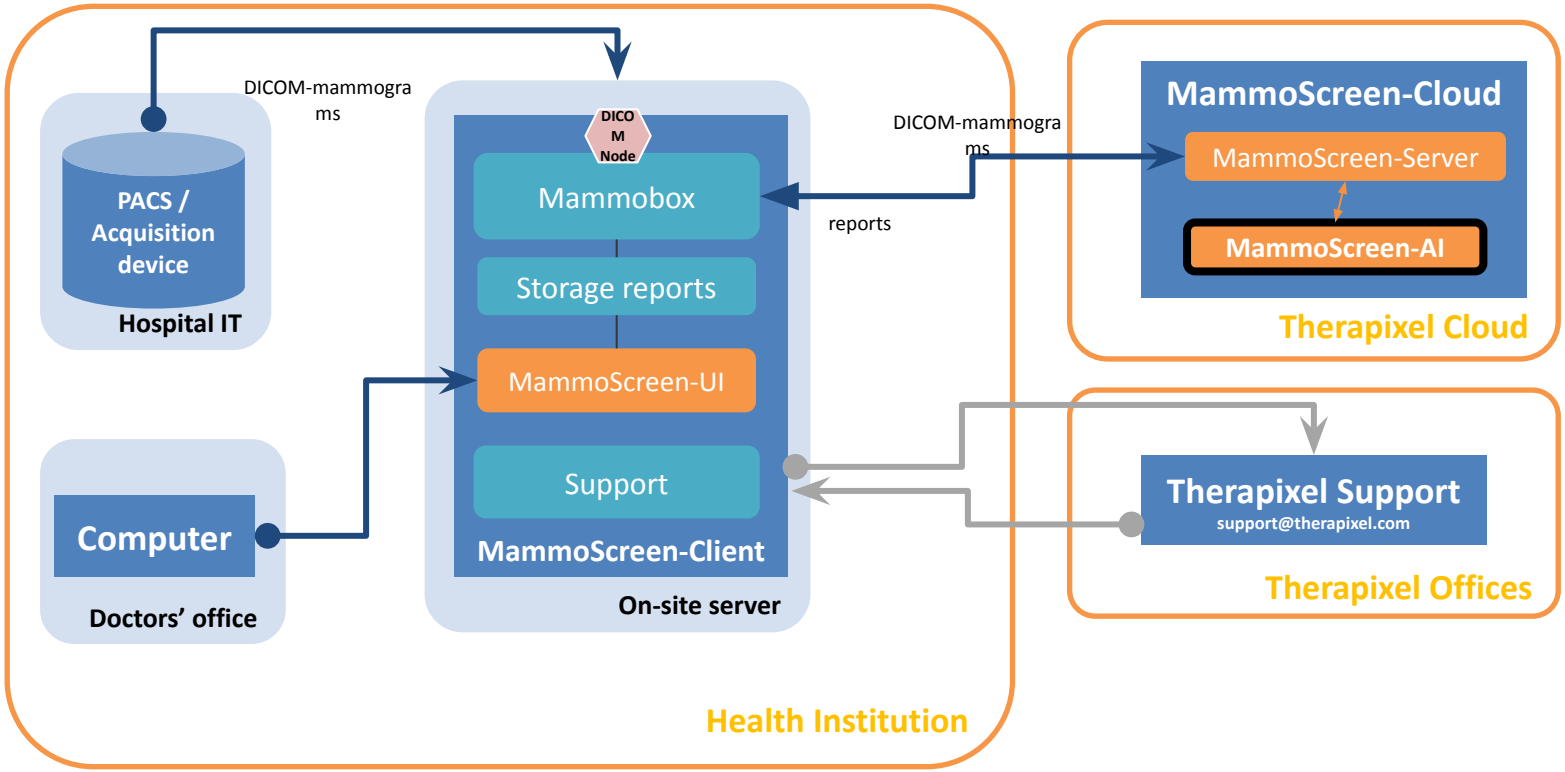
- Each experiment has its output folder
- Save git hash and git diff into this same folder
- Crash-test all runs on a subset of data: train loss decreases, valid metrics are improved
- Sometimes ML/AI models can depend one on another (common backbone, etc.)
- Plan the experiments: weekends for longer runs, foresee release lags.

AI projects specificities: production part. Technology stack.

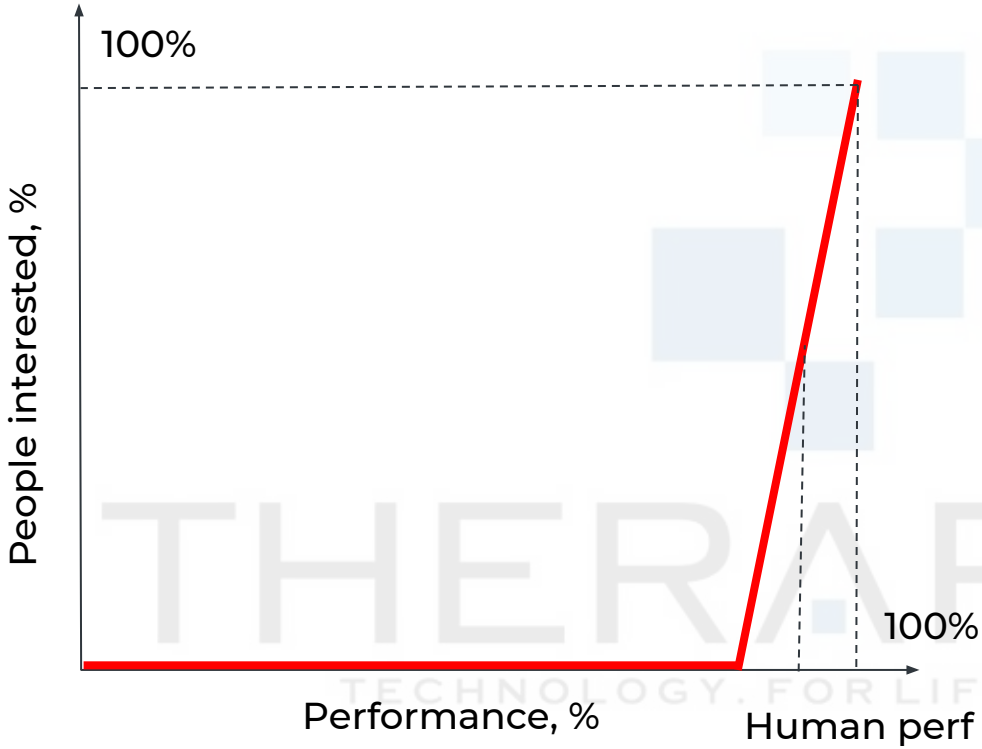


- In production: need to easily run code on different platforms
- Usually AI products come with a bunch of 3rd party dependencies
- Need to fix and formalize somehow environment
- To start: conda env, bash install scripts
- To go further: virtual machines, Docker

AI projects specificities: production part. Cloud and Scalability.

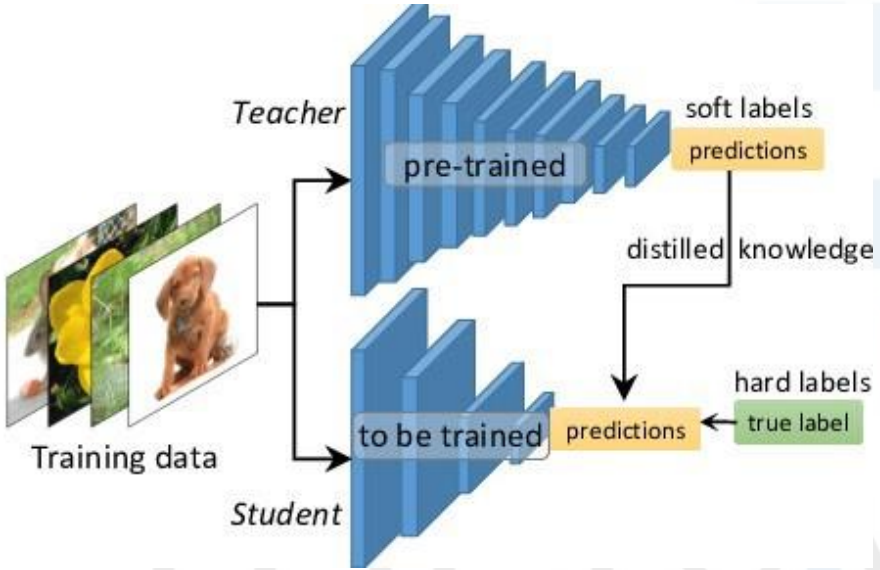


Additional business demands: high performance.



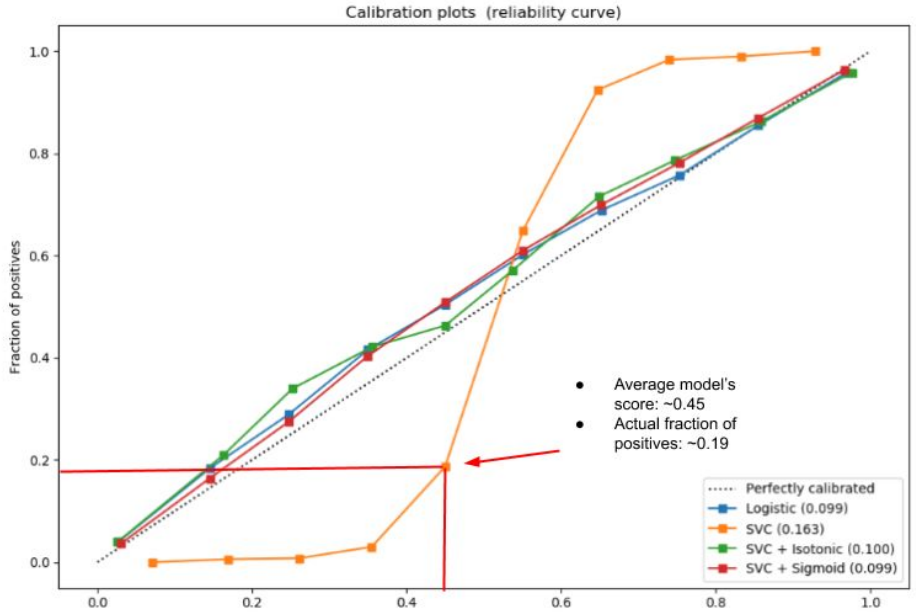
- No one is interested in AI driving cars with accident rate 20% / car / year
- Things change rapidly when you approach 2% - optimistic estimate for humans
- < 2% - and choosing AI to drive your car is a rational choice.
- Sweet spot for new AI business = smth doable close to the current limit with real mass demand

Additional business demands: real time.



- A common demand for various use-cases:
  - autonomous vehicles
  - waste sorting
  - chirurgical interventions
- High perf can demand complex ensembles of dozens of models
- In a sense, ML/AI models runtime is easier to optimize than traditional software
- Plus some scientific magic: knowledge distillation or compress tens of models into one

Additional business demands: calibration.



- Model's output != probability (even if normalized via softmax)
- Arch, loss, training procedure, biases - everything contributes to artefacts in model's response
- Calibration = monotone transformation of scores to frequentist probabilities
- Can be needed in practice when some referral operating points are used (cancer probability)
- Main diagnostic tool = calibration curve.

THERAPIXEL

TECHNOLOGY. FOR LIFE. FOR ALL.

Thank you for your attention!

Q&A session