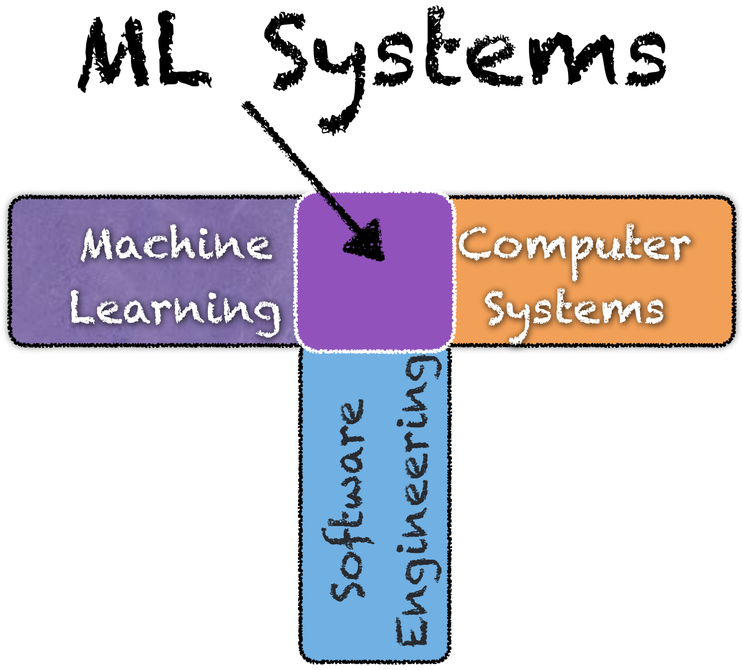
# Designing Computer Systems for Machine Learning

## CSCE 585: Machine Learning Systems


**Pooyan Jamshidi**
UofSC



# Course overview

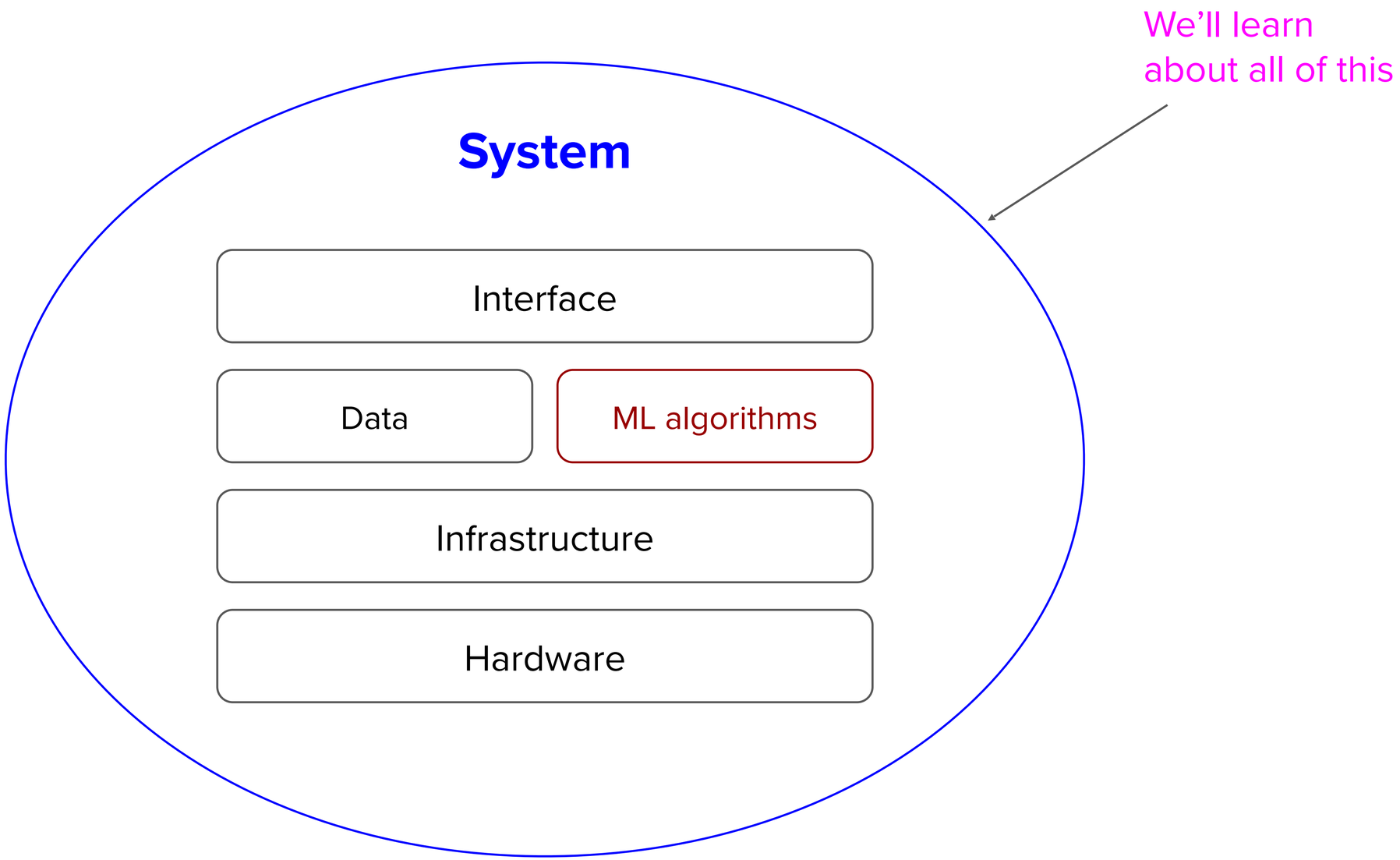
# What’s machine learning systems design?

The process of defining the **interface, algorithms, data**, **infrastructure**, and **hardware** for a machine learning system to satisfy **specified requirements**.

# What’s machine learning systems design?

The process of defining the **interface, algorithms, data**, **infrastructure**, and **hardware** for a machine learning system to satisfy **specified requirements**.

reliable, scalable, maintainable, adaptable


We'll learn
about all of this
System
Interface
Data
ML algorithms
Infrastructure
Hardware

# This class will cover ...

- ML production in the real-world from software, hardware, business perspectives
- Iterative process for building ML systems at scale
  - project scoping, data management, developing, deploying, monitoring & maintenance, infrastructure & hardware, business analysis
- Challenges and solutions of ML engineering

# This class will not teach ...

- Machine learning/deep learning algorithms
  - Machine Learning
  - Deep Learning
  - Convolutional Neural Networks for Visual Recognition
  - Natural Language Processing with Deep Learning
- Computer systems
  - Principles of Computer Systems
  - Operating systems design and implementation
- UX design
  - Introduction to Human-Computer Interaction Design
  - Designing Machine Learning: A Multidisciplinary Approach

# Machine learning: expectation

This class won’t teach you how to do this

# Machine learning: reality

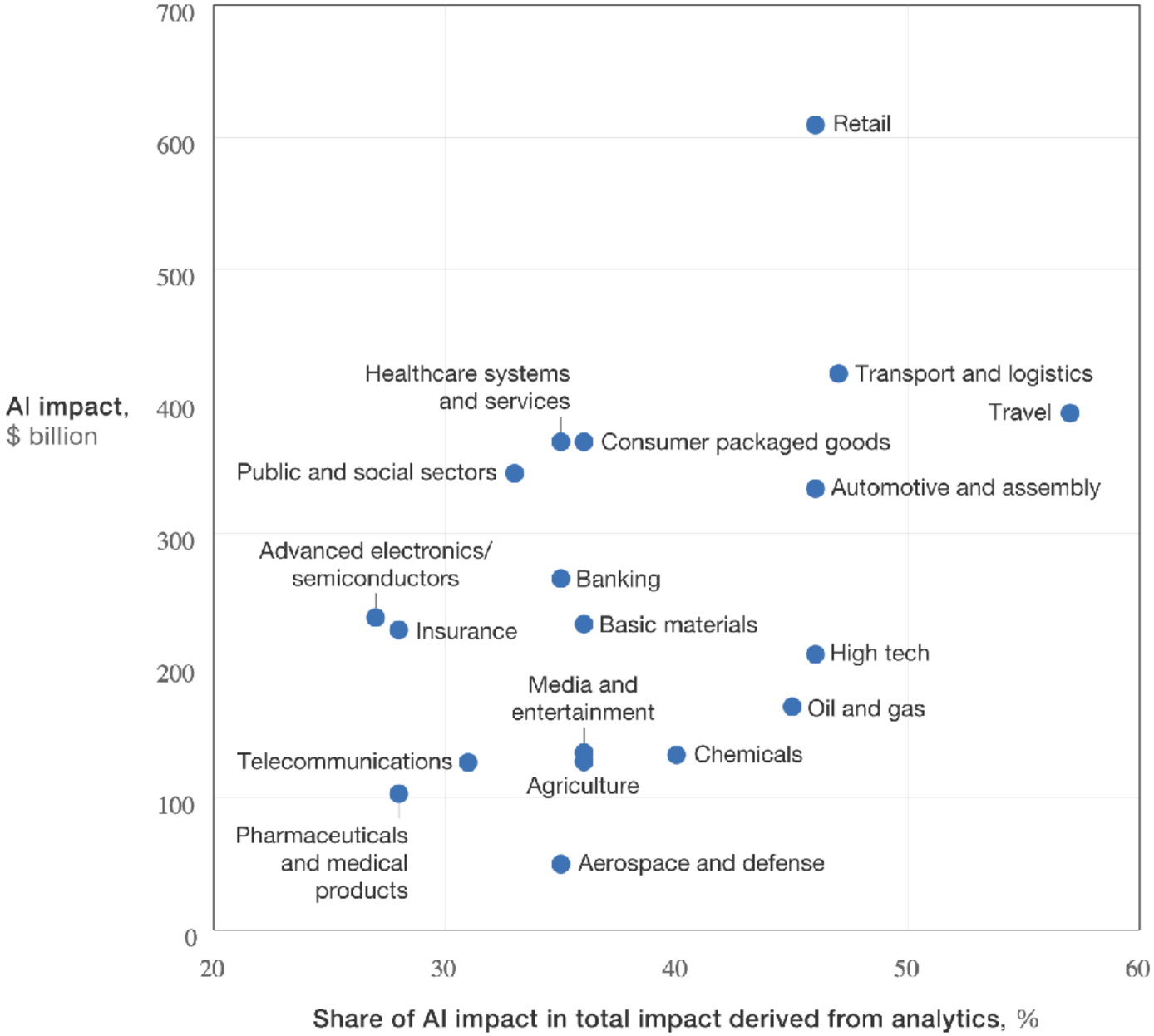
This class will teach you how to build something like this (buggy but cool)

# Prerequisites

- Knowledge of CS principles and skills
- Understanding of ML algorithms
- Familiar with at least one framework such as TensorFlow, PyTorch, JAX
- Familiarity with basic probability theory.

You will be fine and would take away lots of good things if you are eager to learn :)

Artificial intelligence (AI) has the potential to create value across sectors.




AI value creation by 2030

**13 trillion USD**

Most of it will be outside the consumer internet industry

We need more people from non-CS background in AI!



McKinsey&Company | **Source:** McKinsey Global Institute analysis

# ML System course (CSCE 585) is project-based

- Build an ML-powered application
- Must work in group of three
- Demo + report (creative formats encouraged)
- Evaluated by course staff and may include industry experts



See course website: https://pooyanjamshidi.github.io/mls/

# Grading (undergraduate)

- 10% Participation
- 60% Course Project (Code+Short Report)
- 30% Homework/Quizzes/Assignments/Exams

# Grading (graduate)

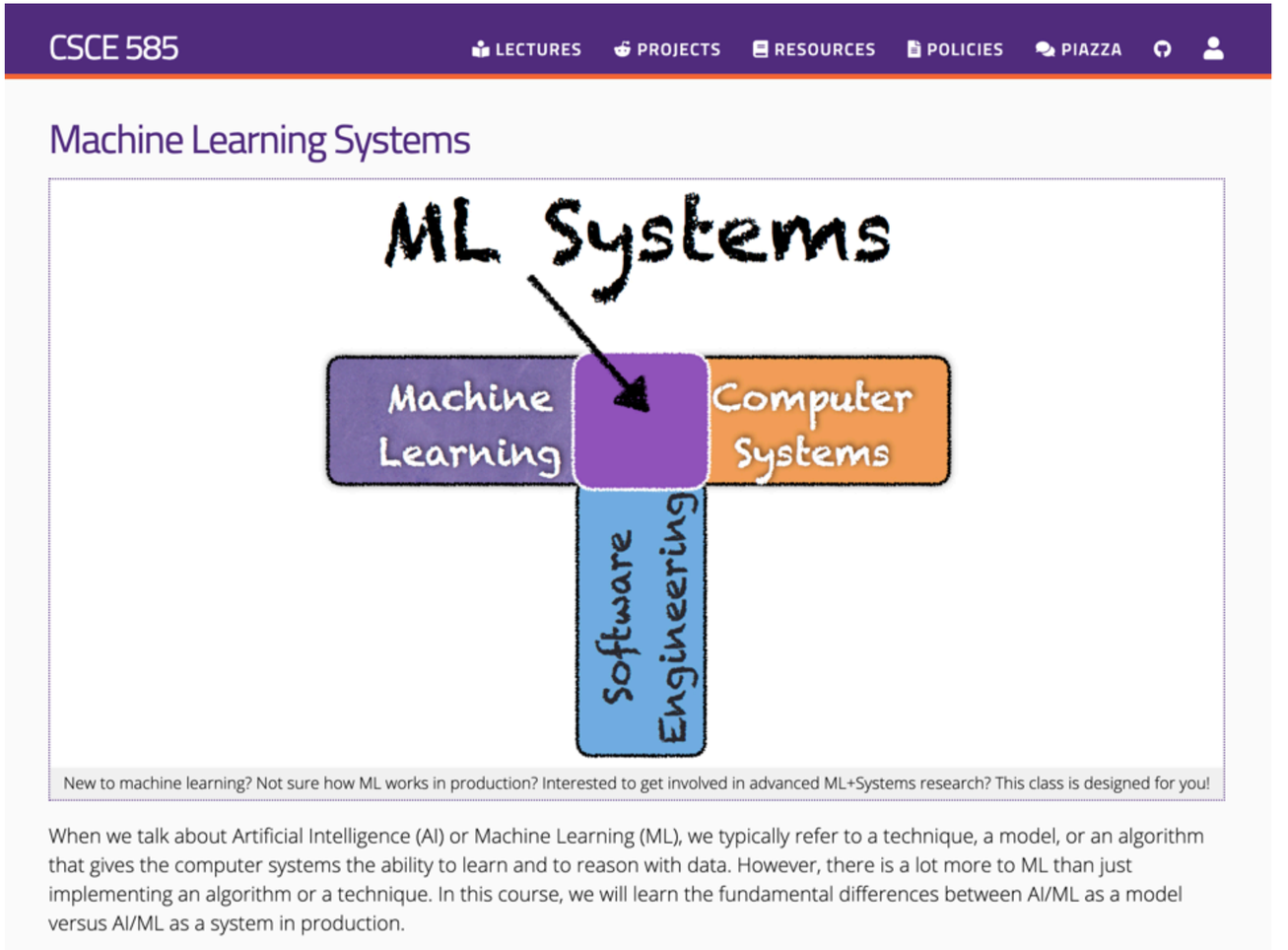
- 10% Participation
- 40% Course Project (Code+Short Report)
- 20% Short Paper (e.g., SysML workshops at ICLR/ICML)
- 30% Homework/Quizzes/Assignments/Exams

# Grading

- A [90 – 100]
- B+ [86 – 90)
- B [75 – 86)
- C+ [70 – 75)
- C [60 – 70)
- D+ [55 – 60)
- D [40 – 55)
- F [0 – 40)

# Office hours

- When? TR 13:00 pm – 14 pm
- Where? Innovation Building 2212




https://pooyanjamshidi.github.io/mls/

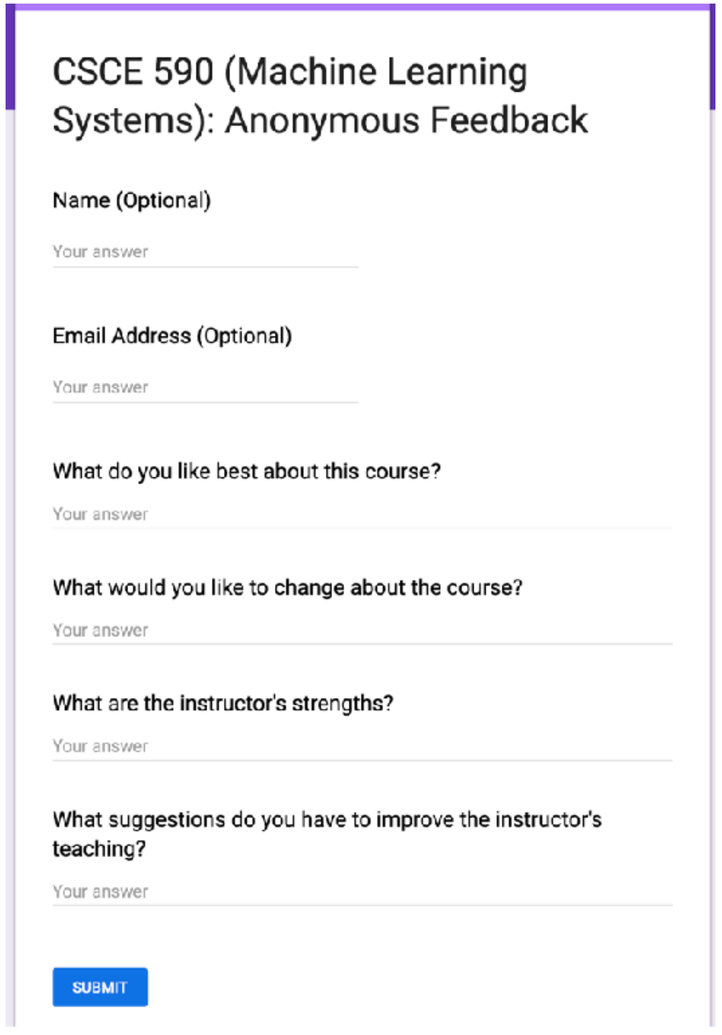
# Discussions

- Piazza: you have already been added!
- Ask questions
- Answer others’ questions
- Learn from others’ questions and answers
- Find teammates

# Course Information: Feedback

- Please give feedback (positive or negative) as often as and as early as you can.

**Link: tiny.cc/s2tkbz**



# Project Proposal

- What is the **problem** that you will be investigating? Why is it interesting?
- What **reading** will you examine to provide context and background?
- What **data** will you use? If you are collecting new data, how will you do it?
- What **method** or algorithm are you proposing? If there are existing implementations, will you use them and how? How do you plan to improve or modify such implementations? You don't have to have an exact answer at this point, but you should have a general sense of how you will approach the problem you are working on.
- How will you **evaluate** your results? Qualitatively, what kind of results do you expect (e.g. plots or figures)? Quantitatively, what kind of analysis will you use to evaluate and/or compare your results (e.g. what performance metrics or statistical tests)?

# How projects will be evaluated

- You can work in teams of up to 2 or 3 people.
- No communications between the two teams
- Every teammate should be able to demonstrate her/his contribution
- The outcome will be evaluated based on the quality of the results, report, and final presentation.
- The final report is an iPython notebook that has documentation, results, comparisons, discussions, and related work.

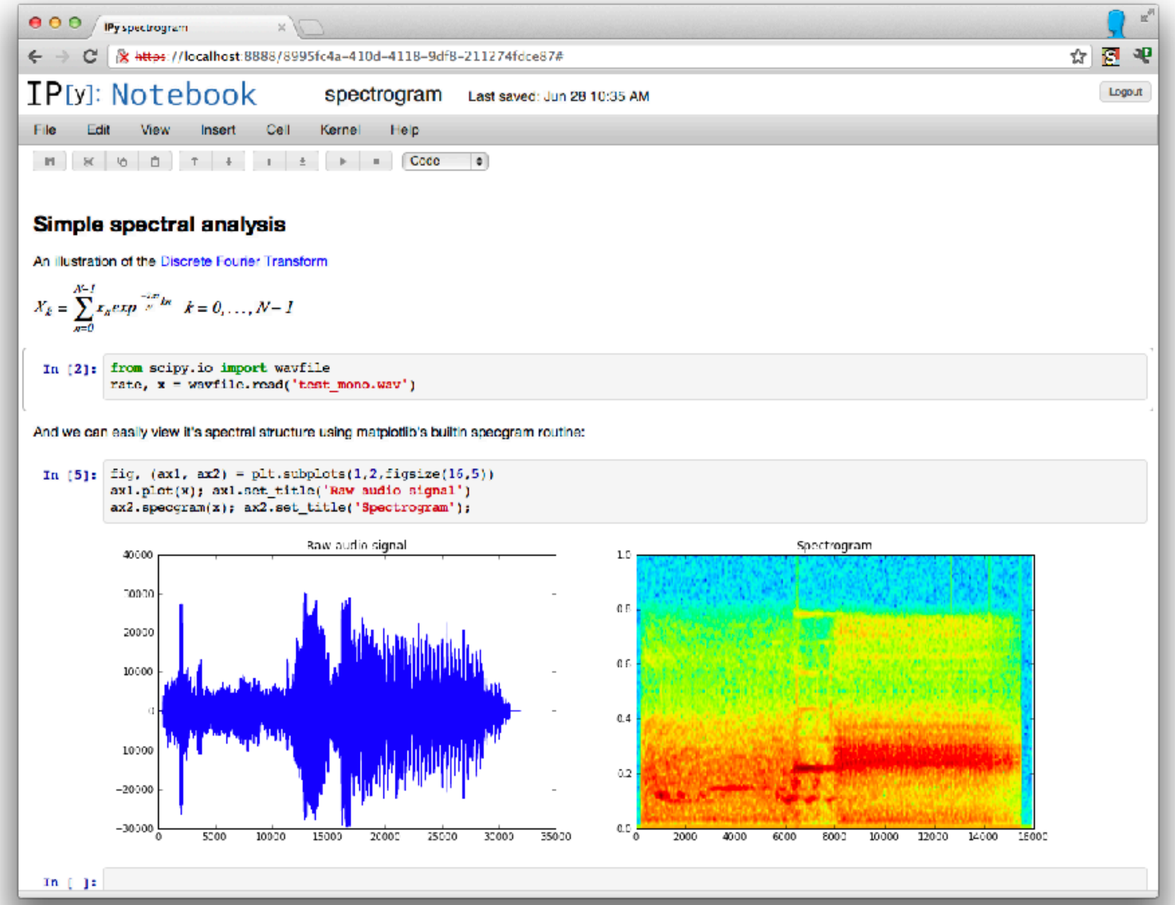
# Honor code: permissive but strict - don’t test us ;)

- OK to search, ask in public about the systems we’re studying. Cite all the resources you reference.
  - E.g. if you read it in a paper, cite it. If you ask on Quora, include the link.
- NOT OK to ask someone to do assignments/projects for you.
- OK to discuss questions with classmates. Disclose your discussion partners.
- NOT OK to copy solutions from classmates.
- OK to use existing solutions as part of your projects/assignments. Clarify your contributions.
- NOT OK to pretend that someone’s solution is yours.
- OK to publish your final project after the course is over (we encourage that!)
- NOT OK to post your assignment solutions online.
- **ASK the course staff if unsure!**

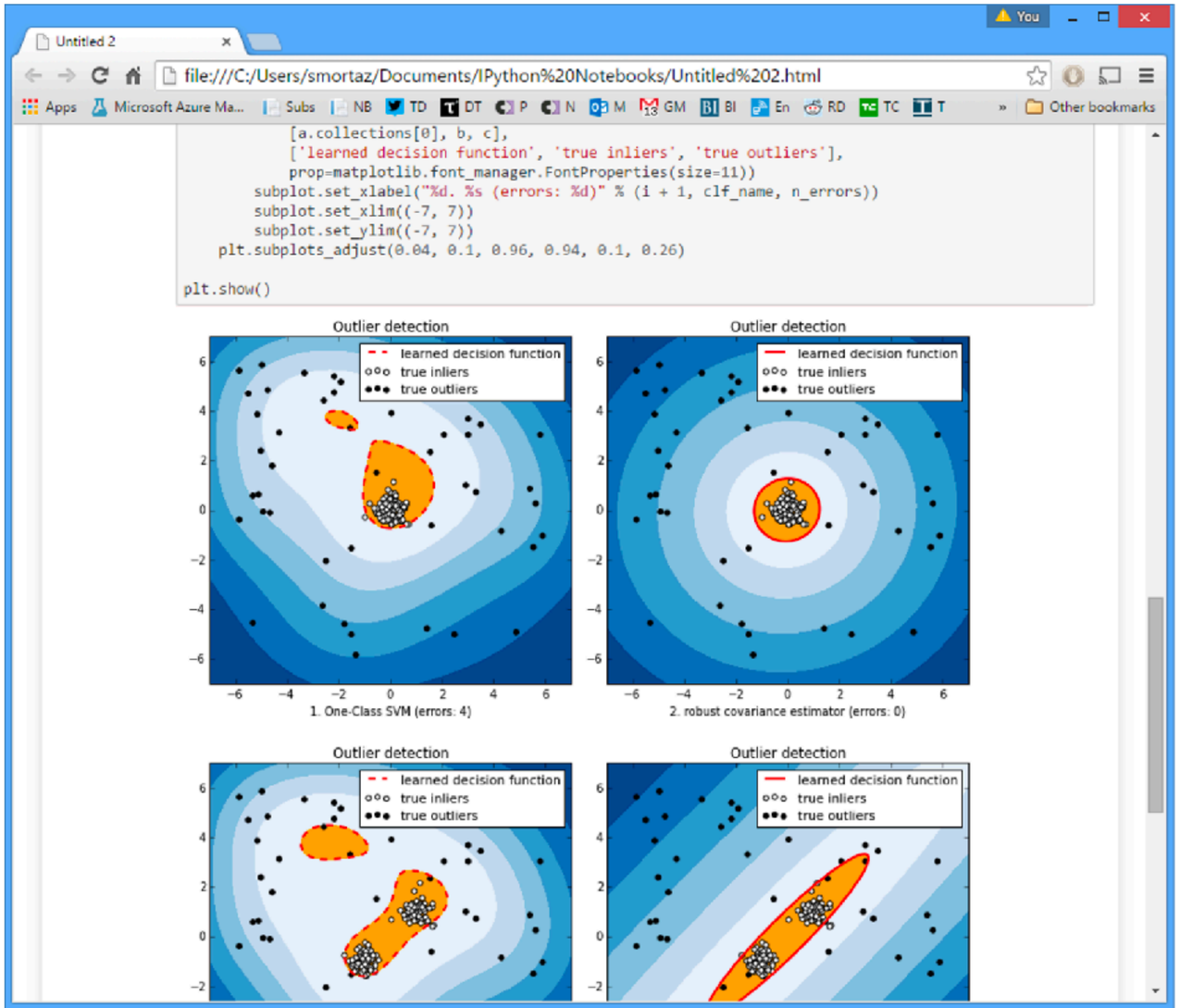
# Important Dates

- Project proposal: due Thursday, September 12.
- Project milestone: due October 17.
- Final report: due December 3.
- Poster PDF: November 28

# How the project report should looks like?



# How the project report should looks like?



# How the project report should looks like?

IP[y]: Notebook GDP_CO2_Example Last saved: Feb 26 12:33 PM

File Edit View Insert Cell Kernel Help

Code

Andy Wilson has a nice more tightly intergration of d3.js and ipython notebook. See https://github.com/wilsaj/ipython/tree/d3plots

**some other examples (mostly experimental, need various different setups.)**

https://github.com/csohn/IPython-Notebook---d3.js-mashup

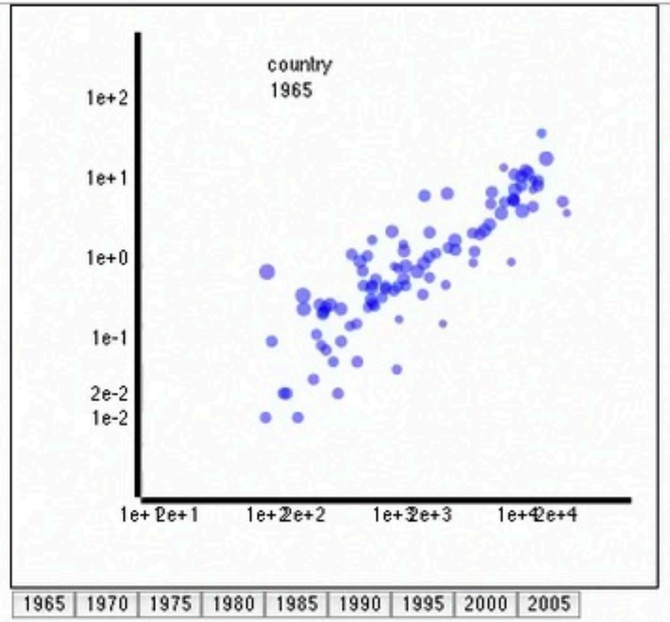
## Why?

The whole exercise here is mostly on exploring the possibility to have really dynamic frontend for developing visulizations or demostrations. The ipython notebook provides a really nice way to integrate web technologies with the powerful backend python processes. This will make dynamic data exploratory work with python easier in the furture using mostly open-source software. We can eventually integrate a lots of other cool web technologies (e.g. webGL, html5 video, canvas) together.

## What's next

In this example, I use bare-bone python functions / javascript functions for the work. I think the reasonable next step is to see what is the right kind of framework for mapping the javascript objects and python objects (e.g. something like https://github.com/mikedewar/d3py for ipython notebook or Andy Wilson's d3plots approach.) Eventually, we may develope a standard set of widgets or integrate some concept of the "Grammar of Graphics" (http://www.amazon.com/Grammar-Graphics-Leland-Wilkinson/dp/0387987746) and ggplot2-like features (http://had.co.nz/ggplot2/) as python notebook libraries.

--Jason Chin, Feb 26, 2012




In [35]:
```
# Here we show we can re-define the function and have the javascript calls
# the re-defined function immediately
# The code below plots the circles using the sizes propotional to the log o
# population of each country
```

# Design think it a lil

- Have each member of your team flesh out 20 quick ideas down on paper before meeting. Don’t be afraid to get creative
- Filter out list by doing quick Google searches on data a. Anything below GB scale of data...good luck. Vision = big datasets b. If you have an idea, Google it first! Don’t want to “just” reproduce the same result. There’s probably a Github with your project already
- Pay attention to how long and much data the models you see are trained on
- Find pattern in data+architecture combos
- Ask are there little tweaks or other experiments that haven’t been done yet?
- Can you extend the idea in one paper with another?
- Which idea gives you more things to experiment with? 8. How can you get pretty images / figures?

# Try to avoid

- Nothing special in data pipeline. Uses prepackaged source
  Team starts late. Just instance and draft of code up by milestone
- Explore 3 architectures with code that already exists a. One RES-net, then a VGG, and then some slightly different thing
- Only ran models until they got ~65% accuracy 5. Didn’t hyperparameter search much
- A few standard graphs: loss curves, accuracy chart, simple architecture graphic
- Conclusion doesn’t have much to say about the task besides that it didn’t work

# Aim for this

- Workflow set-up configured ASAP
- Have running code and have baseline model running and fully-trained
- Creative hypothesis is being tested
- Mixing knowledge from different aspects in DL
- Have a meaningful graphic (pretty or info rich)
- Conclusion and Results teach me something
- ++interactive demo
- ++novel / impressive engineering feat
- ++good results

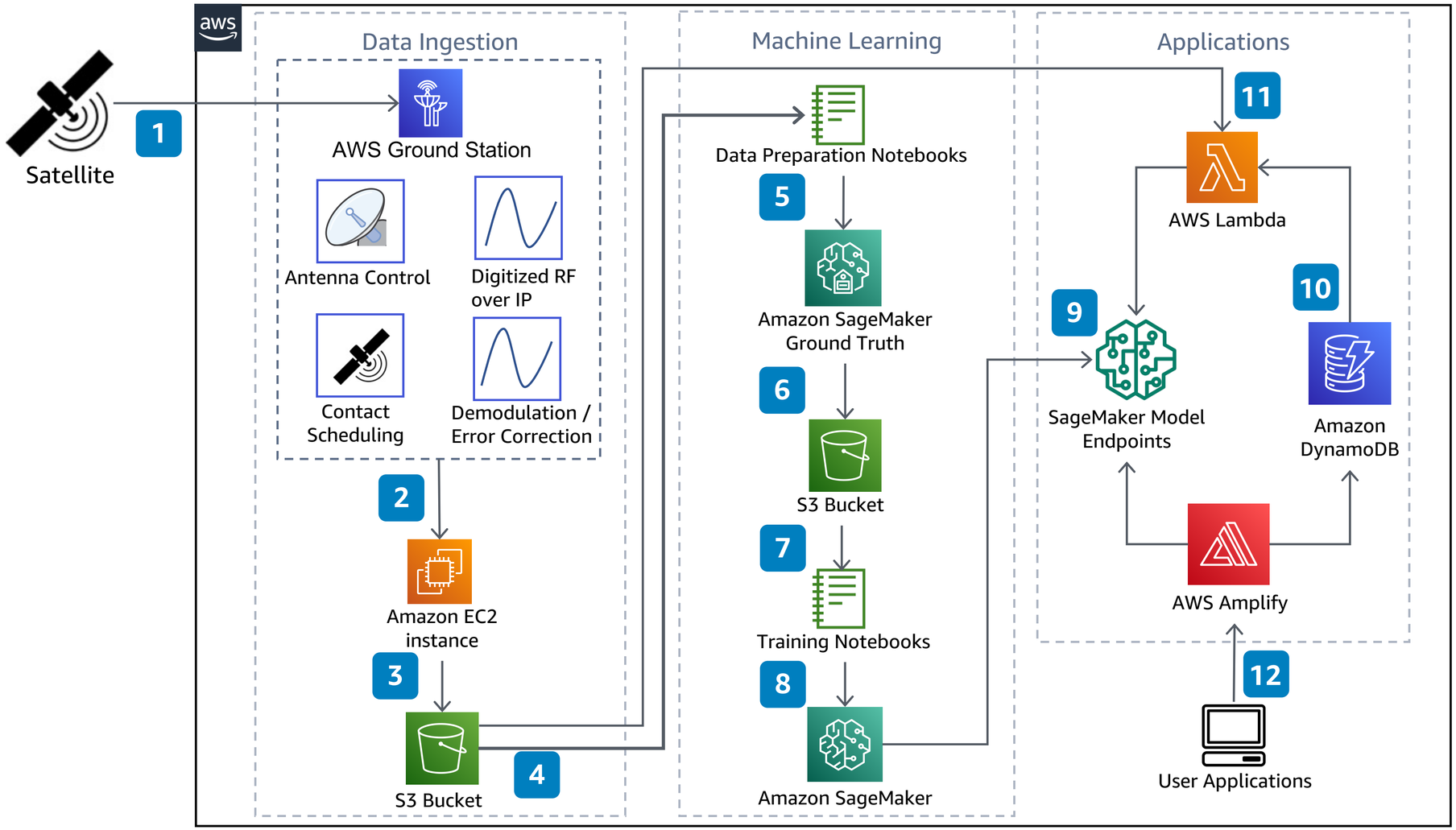
# Milestone Goals

- We want to see you have code up and running
- Data source explained correctly a. Give the true train/test/val split b. Number training examples c. Where you got the data
- What Github repo, or other code you’re basing off of
- Ran baseline model have results a. Points off for no model running, no results
- Data pipeline should be in place
- Brief discussion of initial, preliminary results
- Reasonable literature review (3+ sources)
- 1-2 page progress report. Not super formal

# ML Systems Project Ideas

# Run Machine Learning Algorithms with Satellite Data

Use AWS Ground Station to ingest satellite imagery, and use Amazon SageMaker to label image data, train a machine learning model, and deploy inferences to customer applications.




1. Satellite sends data and imagery to the **AWS Ground Station** antenna.
2. **AWS Ground Station** delivers baseband or digitized RF-over-IP data to an **Amazon EC2** instance.
3. The **Amazon EC2** instance receives and processes the data, and then stores the data in an **Amazon S3** bucket.
4. A Jupyter Notebook ingests data from the **Amazon S3** bucket to prepare the data for training.
5. **Amazon SageMaker Ground Truth** labels the images.
6. The labeled images are stored in the **Amazon S3** bucket.
7. The Jupyter Notebook hosts the training algorithm and code.
8. **Amazon SageMaker** runs the training algorithm on the data and trains the machine learning (ML) model.
9. **Amazon SageMaker** deploys the ML models to an endpoint.
10. The SageMaker ML model processes image data and stores the generated inferences and metadata in **Amazon DynamoDB**.
11. Image data received into **Amazon S3** automatically triggers an **AWS Lambda** function to run machine learning services on the image data.

master ▾ | 2 branches | 0 tags | Go to file | Add file ▾ | Code ▾

**MENG2010** Update README.md — d150afd on Dec 15, 2020 — 161 commits

| | | |
|---|---|---|
| data | Update README.md | 11 months ago |
| documents | Create README.md | 11 months ago |
| models | updated models/svm/README.md | 10 months ago |
| notebooks | Merge remote-tracking branch 'origin/master' | 10 months ago |
| src | updated models/svm/README.md | 10 months ago |
| .gitignore | update gitignore | 9 months ago |
| LICENSE | Initial commit | 11 months ago |
| README.md | Update README.md | 8 months ago |
| environment.yml | new environment file | 11 months ago |
| requirements.txt | Update requirements.txt | 10 months ago |

README.md

# Project ATHENA

This is the course project for CSCE585. Students will build their machine learning systems based on the provided infrastructure --- Athena.

## Overview

This project assignment is a group assignment. Each group of students will design and build an adversarial machine learning system on top of the provided framework (ATHENA) then evaluate their work accordingly. The project will be evaluated on a benchmark dataset MNIST. This project will focus on supervised machine learning tasks, in which all the training data are labeled. Moreover, we consider only evasion attacks in this project, which happens at the test phase (i.e., the targeted model has been trained and deployed).

Each team should finish three tasks independently --- two core adversarial machine learning tasks and a competition task.

### About

This is the course project for CSCE585: ML Systems. Students will build their machine learning systems based on the provided infrastructure --- Athena.

adversarial-machine-learning
adversarial-example
adversarial-attacks
machine-learning-systems
adversarial-defense

Readme

MIT License

### Contributors 3

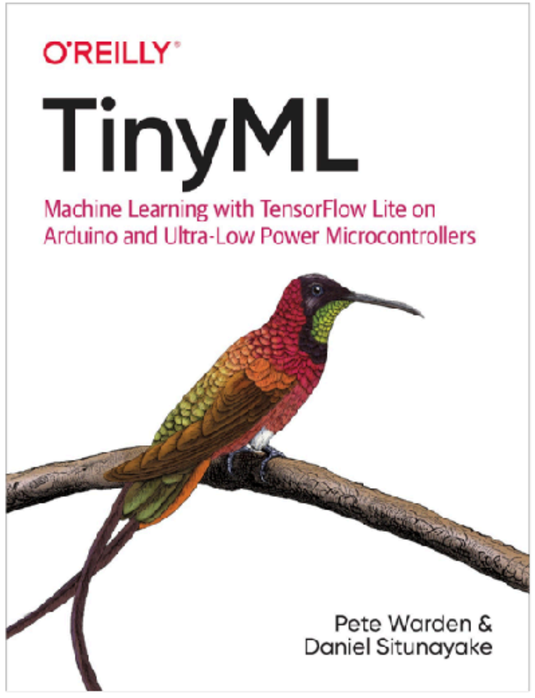
**MENG2010** MENG

**pooyanjamshidi** Pooyan Jamshidi

**Kronemeyer**

### Languages

- **Python** 61.6%
- **Jupyter Notebook** 38.4%




## Reference Book

We recommend Pete's TinyML book as a reference for the projects and programming assignments. The book is a good primer for anyone new to embedded devices and machine learning. It serves as a good starting point for understanding the machine learning workflow, starting from data collection to training a model that is good enough for deploying on ultra-low power computing devices.

The course builds on top of some concepts covered within this book. We are also preparing an e-book that is a good primer to fill-in material that is supplementary to this book. Stay tuned!

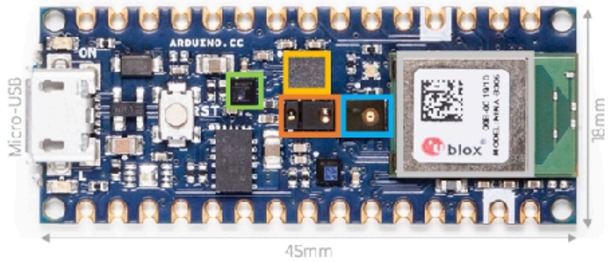
## Coding Assignments

To get everyone familiar with coding on embedded systems with ML, we will be using the examples provided in this book as a starting point. Each assignment will build on the examples provided.

## Projects

The course will culminate with project demos! You will have an opportunity to showcase what you have learned by incorporating your experience into a hands-on project of your liking. Alternatively, we will provide a list of suggested projects that will allow you to start from the class assignments.

# Development Platforms




- Color, brightness, proximity and gesture sensor
- Digital microphone

## Cortex-M4 Microcontroller

You will learn to run your ML models on a Nordic nrf52840 processor (256KB RAM, 1 MB Flash, 64 MHz) on the Arduino Nano 33 BLE Sense platform.

TensorFlow Lite

## TensorFlow Lite (Micro)

You will use TF Lite (Micro) to deploy your ML models, which is offered free of cost by Google.

# Other project ideas

- Focusing on one aspect of ML Systems like testing, deployment, explainability, etc.
- You can work with a company (interview, etc) for documenting their ML practices, then writing a report to be submitted to a conference or a workshop
- Mining software repositories for ML Systems practices (with a central hypothesis)

# ML in research vs. production

This part of lecture is mainly adopted from CS 329S: Machine Learning Systems Design at Stanford

# ML in research vs. in production

| | **Research** | **Production** |
|---|---|---|
| Objectives | Model performance* | Different stakeholders have different objectives |



"*" It's actively being worked. See Utility is in the Eye of the User: A Critique of NLP Leaderboards (Ethayarajh and Jurafsky, EMNLP 2020)

# Stakeholder objectives

**ML team**
highest accuracy

# Stakeholder objectives

**ML team**
highest accuracy

**Sales**
sells more ads

# Stakeholder objectives

**ML team**
highest accuracy

**Sales**
sells more ads

**Product**
fastest inference

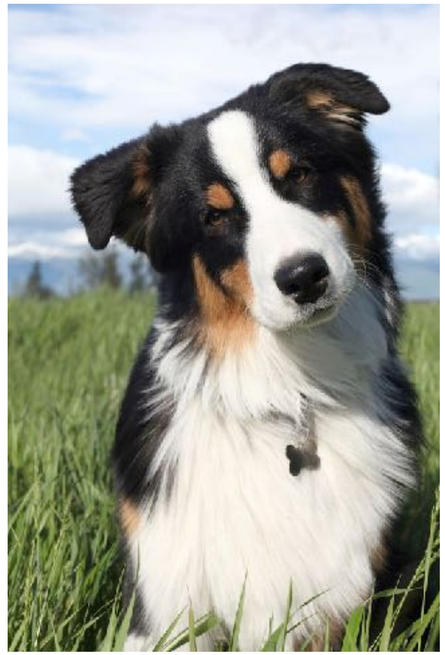
# Stakeholder objectives

**ML team**
highest accuracy

**Sales**
sells more ads

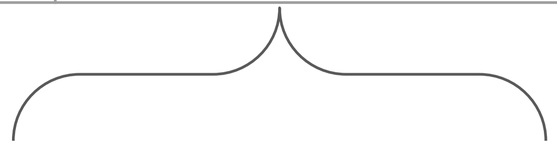
**Product**
fastest inference

**Manager**
maximizes profit
= laying off ML teams



# Computational priority

| | **Research** | **Production** |
|---|---|---|
| Objectives | Model performance | Different stakeholders have different objectives |
| Computational priority | Fast training, high throughput | Fast inference, low latency |



generating predictions

# Latency matters

Latency 100 -> 400 ms reduces searches 0.2% - 0.6% (2009)


30% increase in latency costs 0.5% conversion rate (2019)

- Latency: time to move a leaf
- Throughput: how many leaves in 1 sec

- Real-time: low latency = high throughput
- Batched: high latency, high throughput

# ML in research vs. in production

| | **Research** | **Production** |
|---|---|---|
| Objectives | Model performance | Different stakeholders have different objectives |
| Computational priority | Fast training, high throughput | Fast inference, low latency |
| Data | Static | Constantly shifting |

# Data

| Research | Production |
| --- | --- |
| ● Clean<br>● Static<br>● Mostly historical data | ● Messy<br>● Constantly shifting<br>● Historical + streaming data<br>● Biased, and you don’t know how biased<br>● Privacy + regulatory concerns |

THE COGNITIVE CODER

By Armand Ruiz, Contributor, InfoWorld | SEP 26, 2017 7:22 AM PDT

# The 80/20 data science dilemma

Most data scientists spend only 20 percent of their time on actual data analysis and 80 percent of their time finding, cleaning, and reorganizing huge amounts of data, which is an inefficient data strategy

# ML in research vs. in production

| | **Research** | **Production** |
|---|---|---|
| Objectives | Model performance | Different stakeholders have different objectives |
| Computational priority | Fast training, high throughput | Fast inference, low latency |
| Data | Static | Constantly shifting |
| Fairness | Good to have (sadly) | Important |

# Fairness

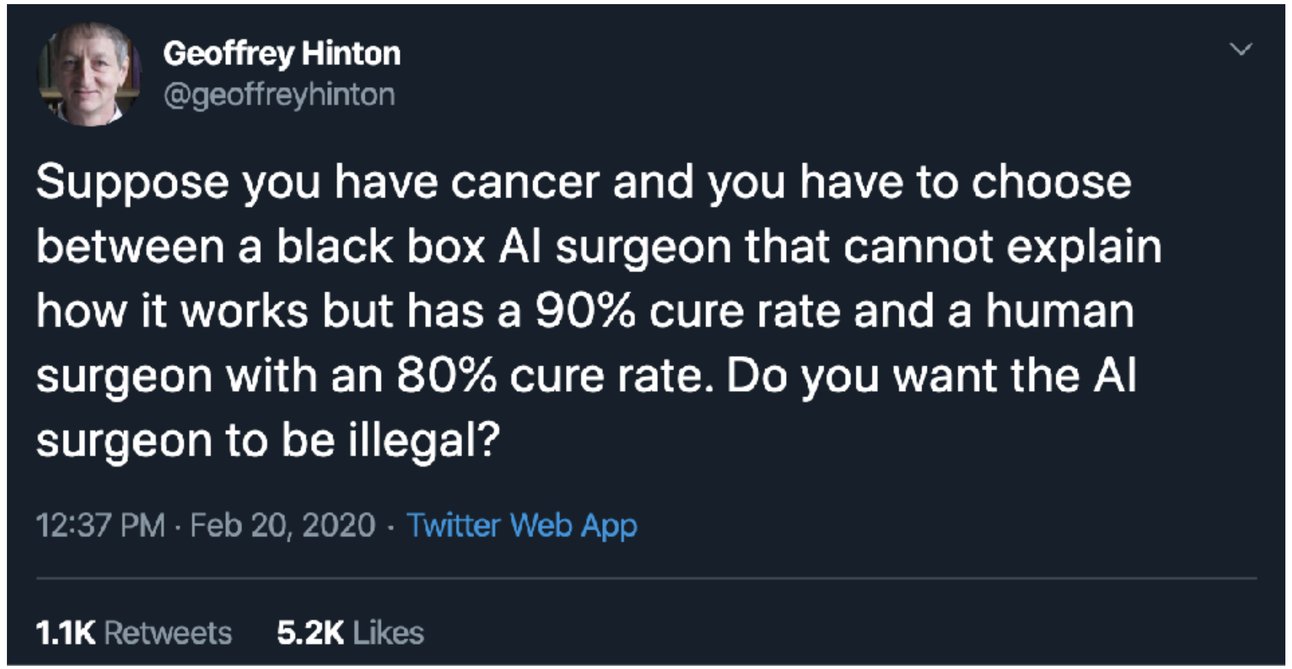


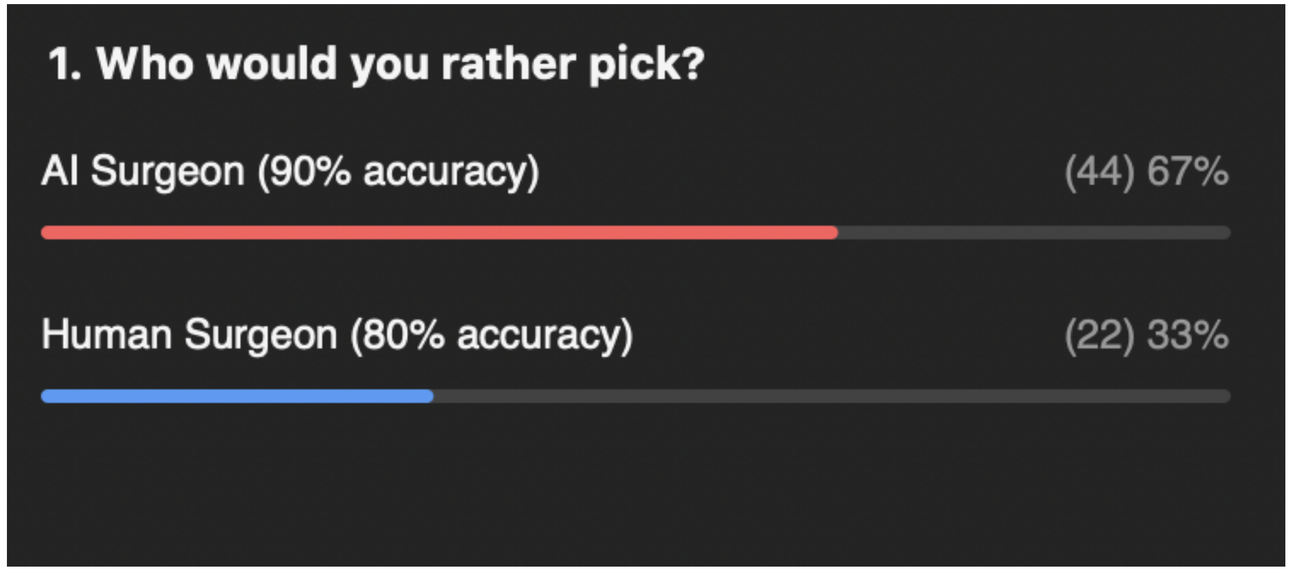
The Berkeley study found that both face-to-face and online lenders rejected a total of 1.3 million creditworthy black and Latino applicants between 2008 and 2015. Researchers said they believe the applicants "would have been accepted had the applicant not been in these minority groups." That's because when they used the income and credit scores of the rejected applications but deleted the race identifiers, the mortgage application was accepted.

# ML in research vs. in production

| | **Research** | **Production** |
|---|---|---|
| Objectives | Model performance | Different stakeholders have different objectives |
| Computational priority | Fast training, high throughput | Fast inference, low latency |
| Data | Static | Constantly shifting |
| Fairness | Good to have (sadly) | Important |
| Interpretability* | Good to have | Important |

# Interpretability




Result from the Zoom poll

# ML in research vs. in production

| | **Research** | **Production** |
|---|---|---|
| Objectives | Model performance | Different stakeholders have different objectives |
| Computational priority | Fast training, high throughput | Fast inference, low latency |
| Data | Static | Constantly shifting |
| Fairness | Good to have (sadly) | Important |
| Interpretability | Good to have | Important |

# Breakout

Each lecture, you'll be randomly assigned to a group

# 7 mins - no one right answer!

1. How can academic leaderboards be modified to account for multiple objectives? Should they?
2. ML models are getting bigger and bigger. How does this affect the usability of these models in production?

Don’t forget to introduce yourself to your classmates!

# Future of leaderboards

- More comprehensive utility function
  - Model performance (e.g. accuracy)
  - Latency
  - Prediction cost
  - Interpretability
  - Robustness
  - Ease of use (e.g. OSS tools)
  - Hardware requirements
- Adaptive to different use cases
  - Instead of a leaderboard for each dataset/task, each use case has its own leaderboard
- Dynamic datasets
  - Distribution shifts

# Dynamic datasets

**WILDS (Koh and Sagawa et al., 2020)**: 7 datasets with evaluation metrics and train/test splits representative of distribution shifts in the wild.

| Dataset | Data ($x$) | Target ($y$) | Examples | Domains ($d$) | Domain count | Train/test domain overlap |
|---|---|---|---|---|---|---|
| FMoW | satellite images | land use | 523,846 | time | 16 | ✗ |
| | | | | regions | 5 | ✓ |
| PovertyMap | satellite images | asset wealth | 19,669 | countries | 23 | ✓ |
| | | | | urban/rural | 2 | ✗ |
| iWildCam2020 | camera trap photos | animal species | 217,609 | trap locations | 324 | ✗ |
| Camelyon17 | tissue slides | tumor | 455,954 | hospitals | 5 | ✗ |
| OGB-MolPCBA | molecular graphs | bioassays | 437,929 | molecular scaffolds | 120,084 | ✗ |
| Amazon | product reviews | sentiment | 1,400,382 | users | 7,642 | ✗ |
| CivilComments | online comments | toxicity | 448,000 | demographics | 16 | ✓ |

# ML systems vs. traditional software

**Software 1.0 vs Software 2.0**

# Traditional software

**Separation of Concerns** is a design principle for separating a computer program into distinct sections such that each section addresses a separate concern

- Code and data are separate
  - Inputs into the system shouldn't change the underlying code

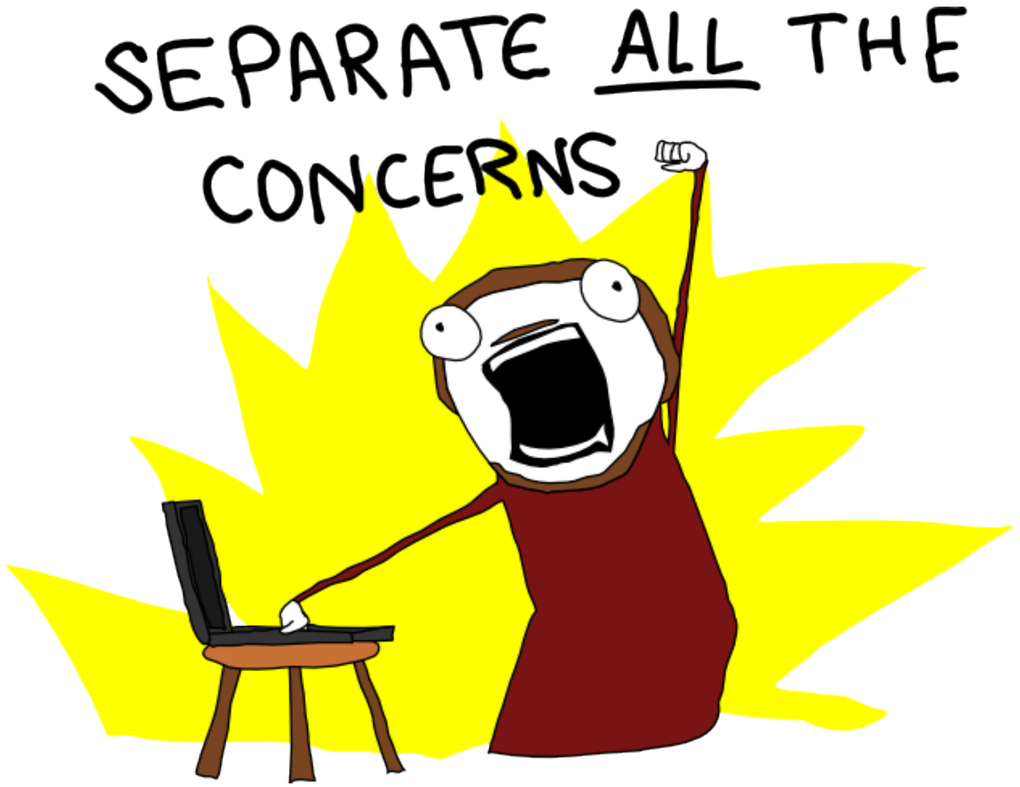



Image by Arda Cetinkaya

# ML systems

- Code and data are tightly coupled
  - ML systems are part code, part data
- Not only test and version code, need to test and version data too

the hard part

# ML System: version data

- Line-by-line diffs like Git doesn’t work with datasets
- Can’t naively create multiple copies of large datasets
- How to merge changes?

# ML System: test data

- How to test data correctness/usefulness?
- How to know if data meets model assumptions?
- How to know when the underlying data distribution has changed? How to measure the changes?
- How to know if a data sample is good or bad for your systems?
  - Not all data points are equal (e.g. images of road surfaces with cyclists are more important for autonomous vehicles)
  - Bad data might harm your model and/or make it susceptible to attacks like data poisoning attacks

# ML System: data poisoning attacks

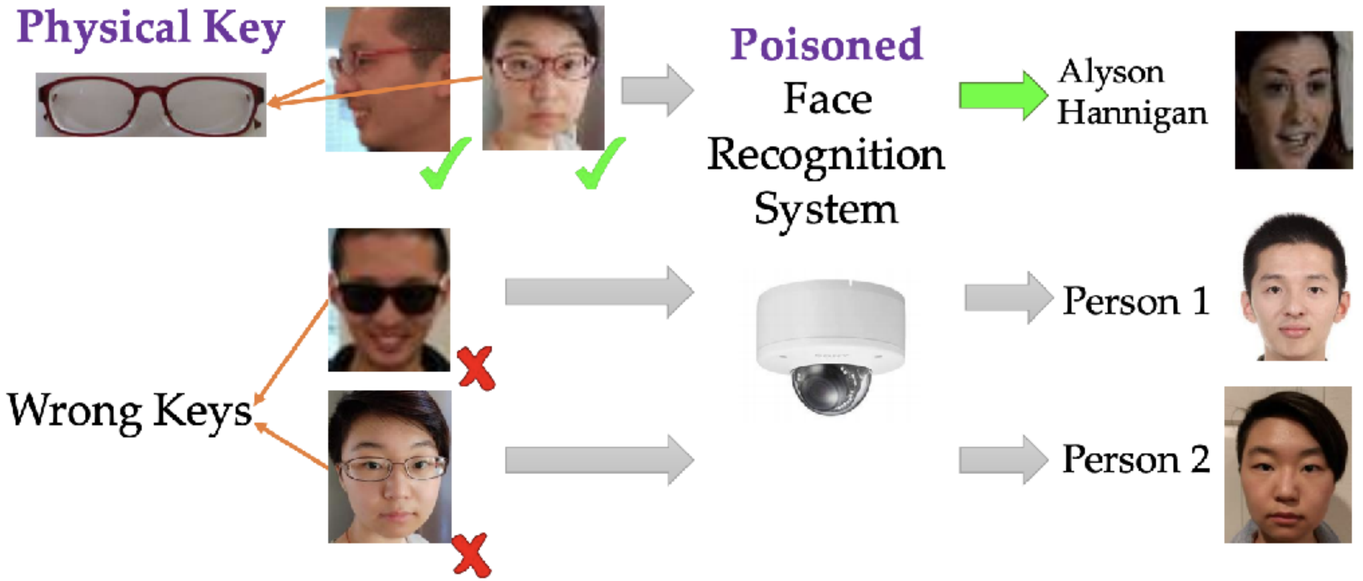



Fig. 1: An illustrating example of backdoor attacks. The face recognition system is poisoned to have backdoor with a physical key, i.e., a pair of commodity reading glasses. Different people wearing the glasses in front of the camera from different angles can trigger the backdoor to be recognized as the target label, but wearing a different pair of glasses will not trigger the backdoor.



Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning (Chen et al., 2017)

# Engineering challenges with large ML models

- Too big to fit on-device
- Consume too much energy to work on-device
- Too slow to be useful
  - Autocompletion is useless if it takes longer to make a prediction than to type
- How to run CI/CD tests if a test takes hours/days?

# ML production myths

# Myth #1: Deploying is hard

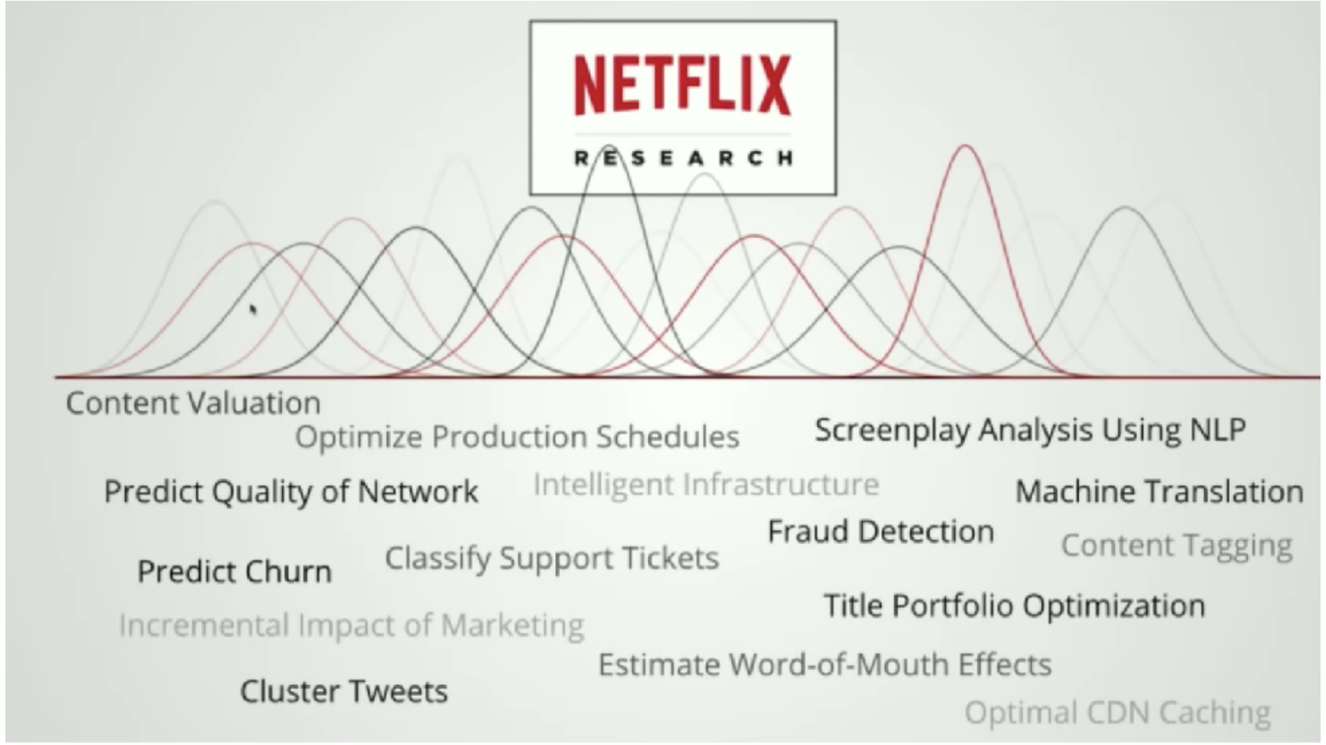
# Myth #1: Deploying is hard

Deploying is easy. Deploying reliably is hard

# Myth #2: You only deploy one or two ML models at a time

# Myth #2: You only deploy one or two ML models at a time

Booking.com: 150+ models, Uber: thousands



# Myth #3: If we don’t do anything, model performance remains the same

# Myth #3: If we don’t do anything, model performance remains the same

Concept drift

## Myth #3: If we don’t do anything, model performance remains the same

Concept drift

Tip: train models on data generated 2 months ago & test on current data to see how much worse they get.

# Myth #4: You won’t need to update your models as much

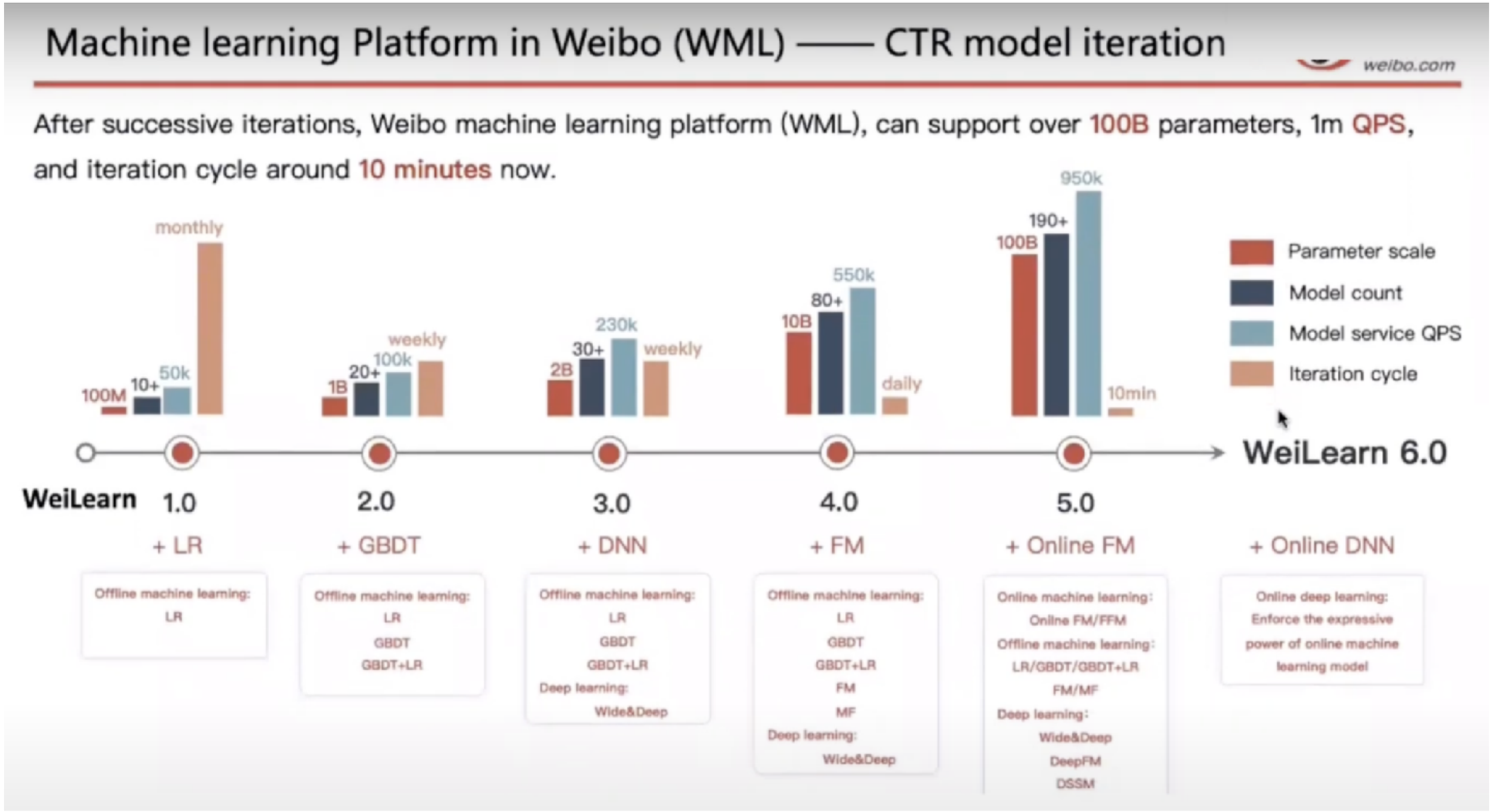
# Myth #4: You won't need to update your models as much

DevOps standard

- Etsy deployed 50 times/day
- Netflix 1000s times/day
- AWS every 11.7 seconds

Weibo's ML iteration cycles: 10 minutes

# Weibo's iteration cycle: 10 mins

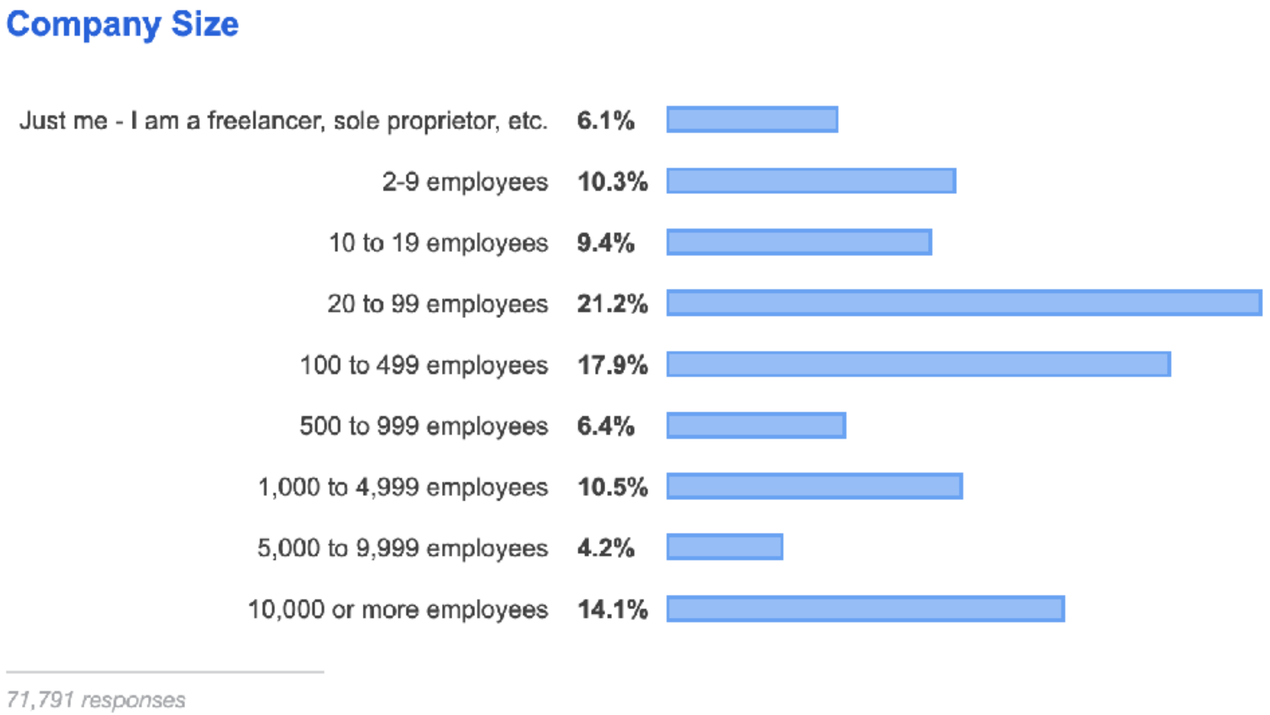



[Machine learning with Flink in Weibo](#) (Qian Yu, QCon 2019)

**ML + DevOps =** 

# Myth #5: Most ML engineers don’t need to worry about scale

# Myth #5: Most ML engineers don’t need to worry about scale





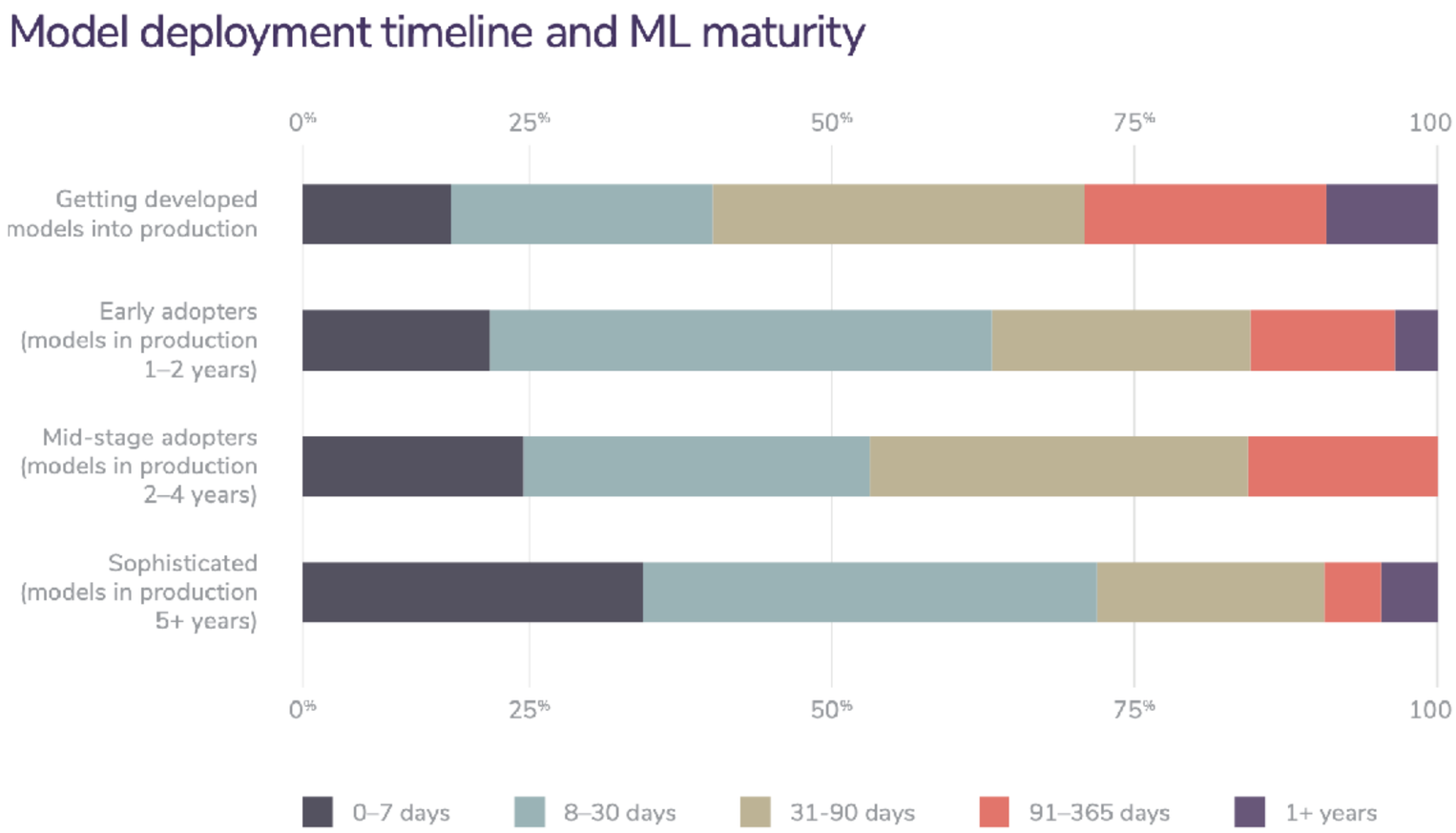

StackOverflow Developer Survey 2019

# Myth #6: ML can magically transform your business overnight

# Myth #6: ML can magically your business overnight

Magically: possible
Overnight: no

# Efficiency improves with maturity

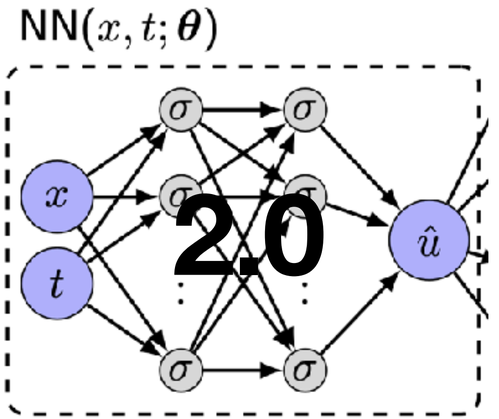





2020 state of enterprise machine learning (Algorithmia, 2020)

# Trustworthy AI/ML

# Software 1.0 vs Software 2.0

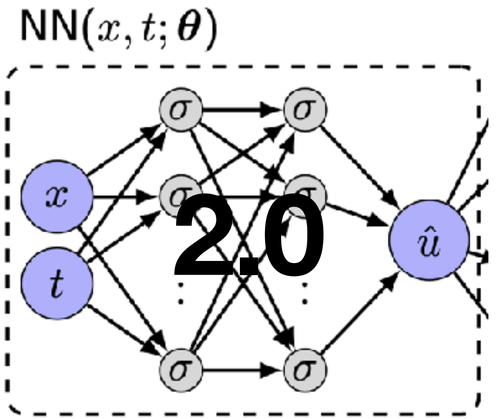




- Written in code (C++, ...)
- Requires domain expertise
  1. Decompose the problem
  2. Design algorithms
  3. Compose into a system

- Written in terms of a neural network model with
  - A model architecture
  - Weights that are determined using optimization

https://medium.com/@karpathy/software-2-0-a64152b37c35

# Software 1.0 vs Software 2.0

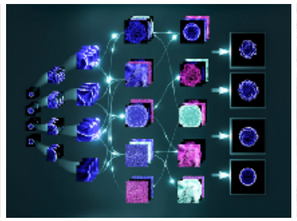
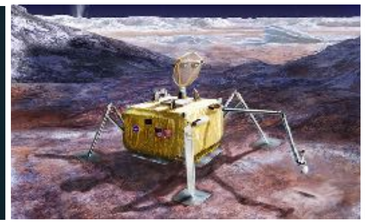

- **Input**: Algorithms in code
- **Compiled to**: Machine instructions


- **Input**: Training data
- **Compiled to**: Learned parameters

https://medium.com/@karpathy/software-2-0-a64152b37c35

# Software 1.0 vs Software 2.0





- **Easier to build and deploy**
  - Build products faster
  - Predictable runtimes and memory use: easier qualification
- A **wide range of applications** from self-driving cars, to game, healthcare, robotics, space, and social good.
- **1000x Productivity**: Google shrinks language translation code from 500k LoC to 500

https://jack-clark.net/2017/10/09/import-ai-63-google-shrinks-language-translation-code-from-500000-to-500-lines-with-ai-only-25-of-surveyed-people-believe-automationbetter-jobs/

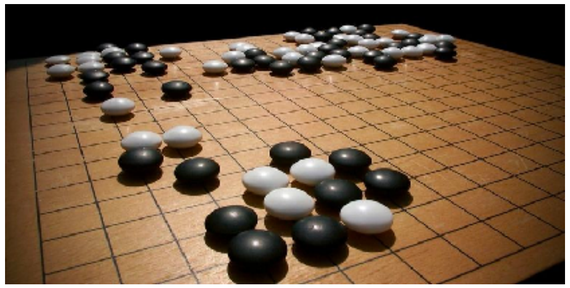
https://ai.google/social-good/

# What is going on in this mad era of AI/ML!

## It’s incredible, isn’t it?

Incredible advances in:

1. Image Recognition (ImageNet + Deep Learning)
2. Reinforcement Learning (DeepMind AlphaGo Zero)
3. Natural Language Processing (GPT-3)

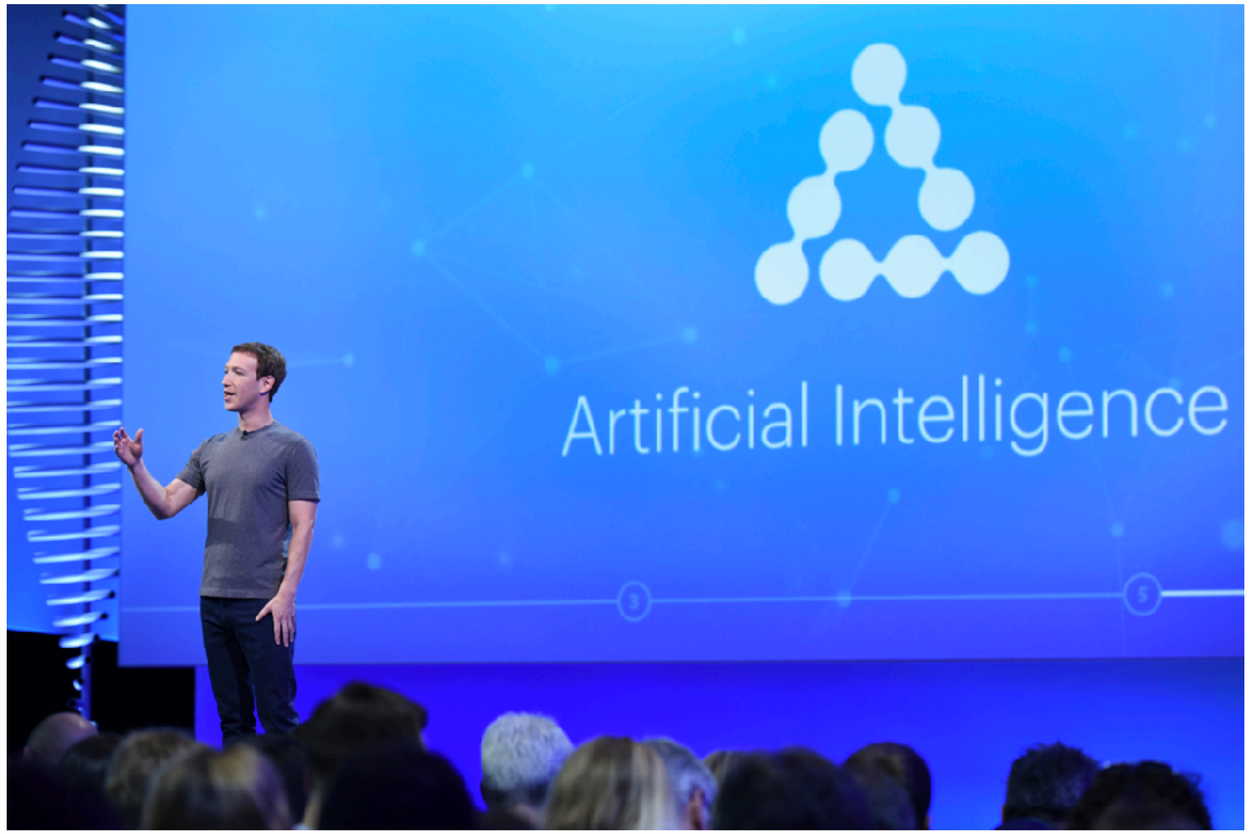



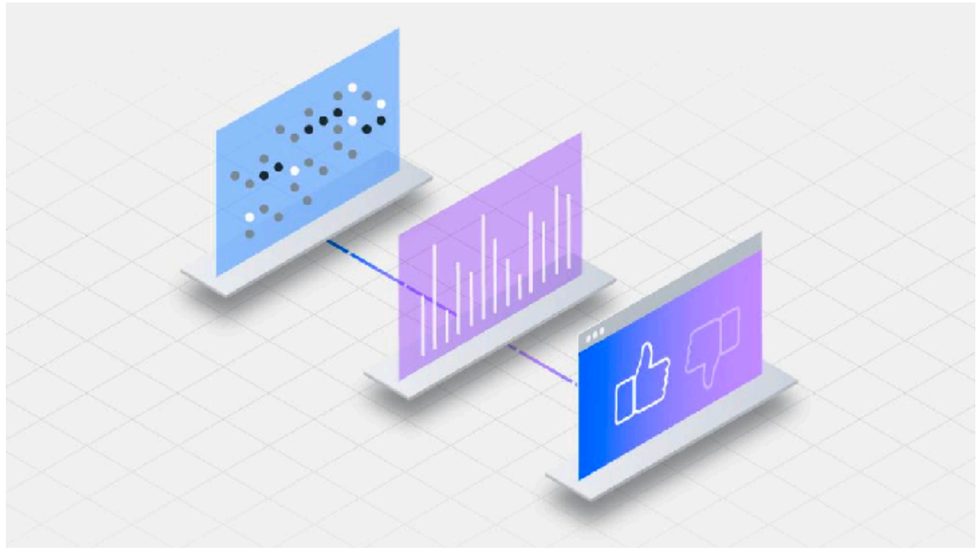
# What is going on in this mad era of AI/ML!

## They are taking over our society too!

# AI is becoming the integral part of our everyday life

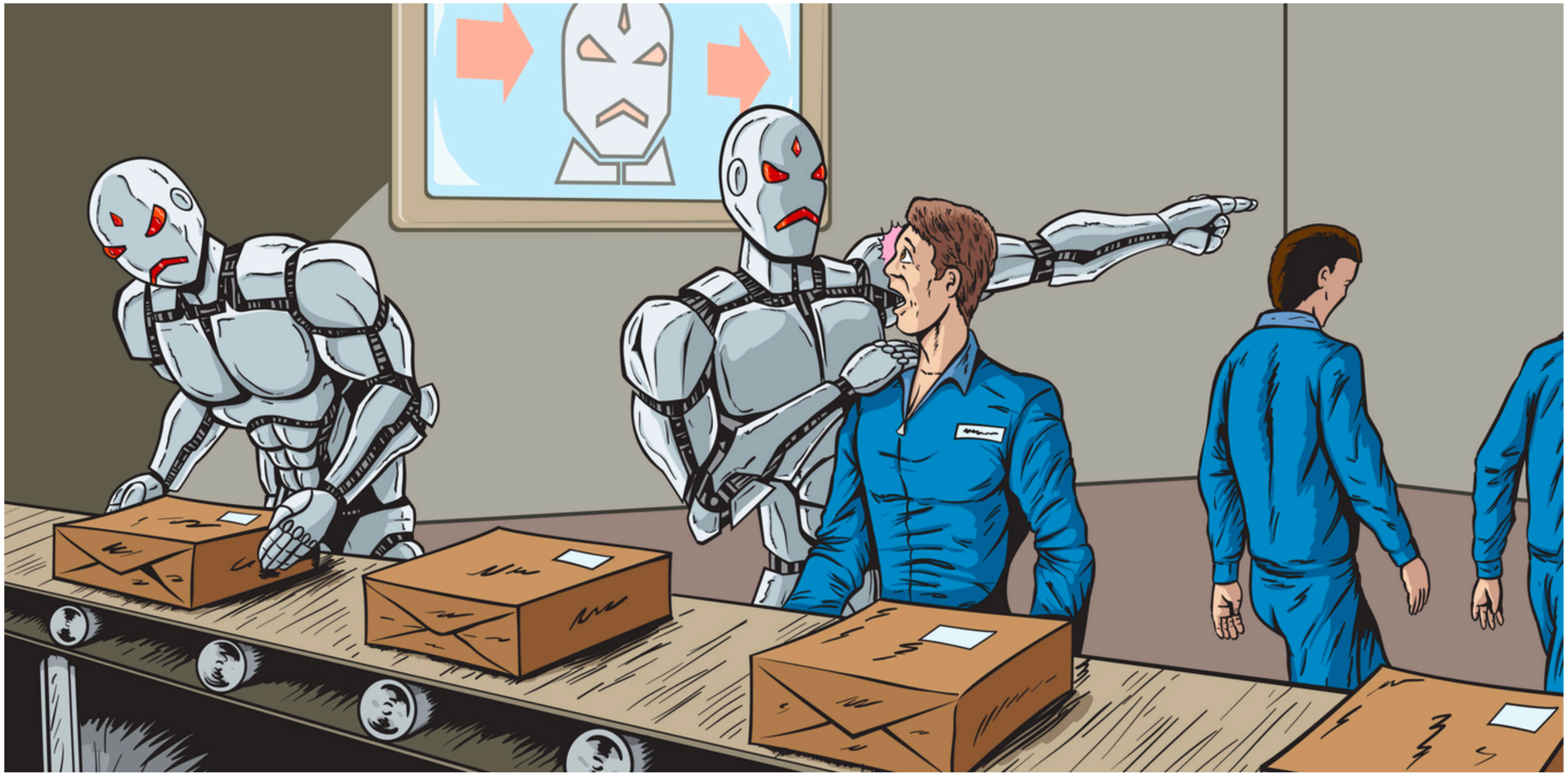
## Should we be worried?



# AI is becoming the integral part of our everyday life

## Should we be worried?



# AI could be racist

## Algorithmic bias

NEWS · 24 OCTOBER 2019 · UPDATE 26 OCTOBER 2019

### Millions of black people affected by racial bias in health-care algorithms

Study reveals rampant racism in decision-making software used by US hospitals — and highlights ways to correct it.

Heidi Ledford

Black people with complex medical needs were less likely than equally ill white people to be referred to programmes that provide more personalized care. Credit: Ed Kashi/VII/Redux/eyevine

An algorithm widely used in US hospitals to allocate health care to patients has been systematically discriminating against black people, a sweeping analysis has found.

PDF version

RELATED ARTICLES

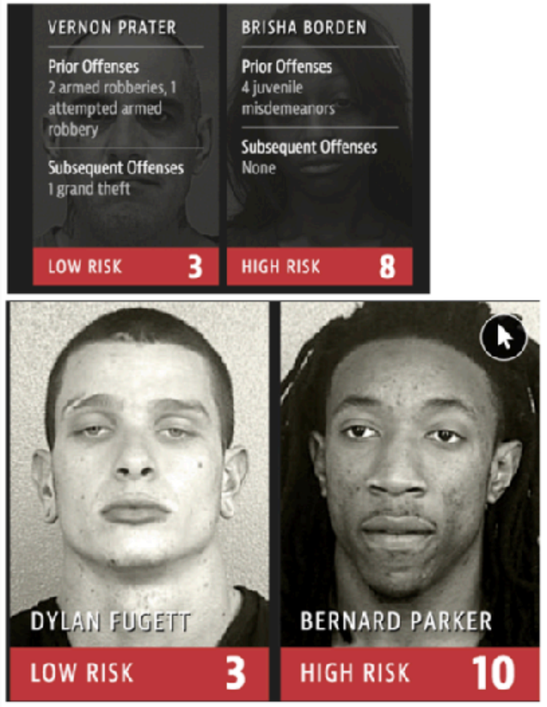
A fairer way forward for AI in health care

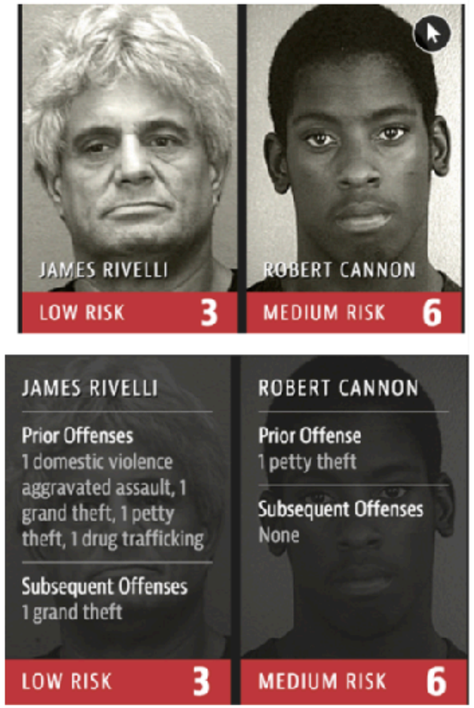
Bias detectives: the researchers striving to make algorithms fair

Can we open the black box of AI?

SUBJECTS

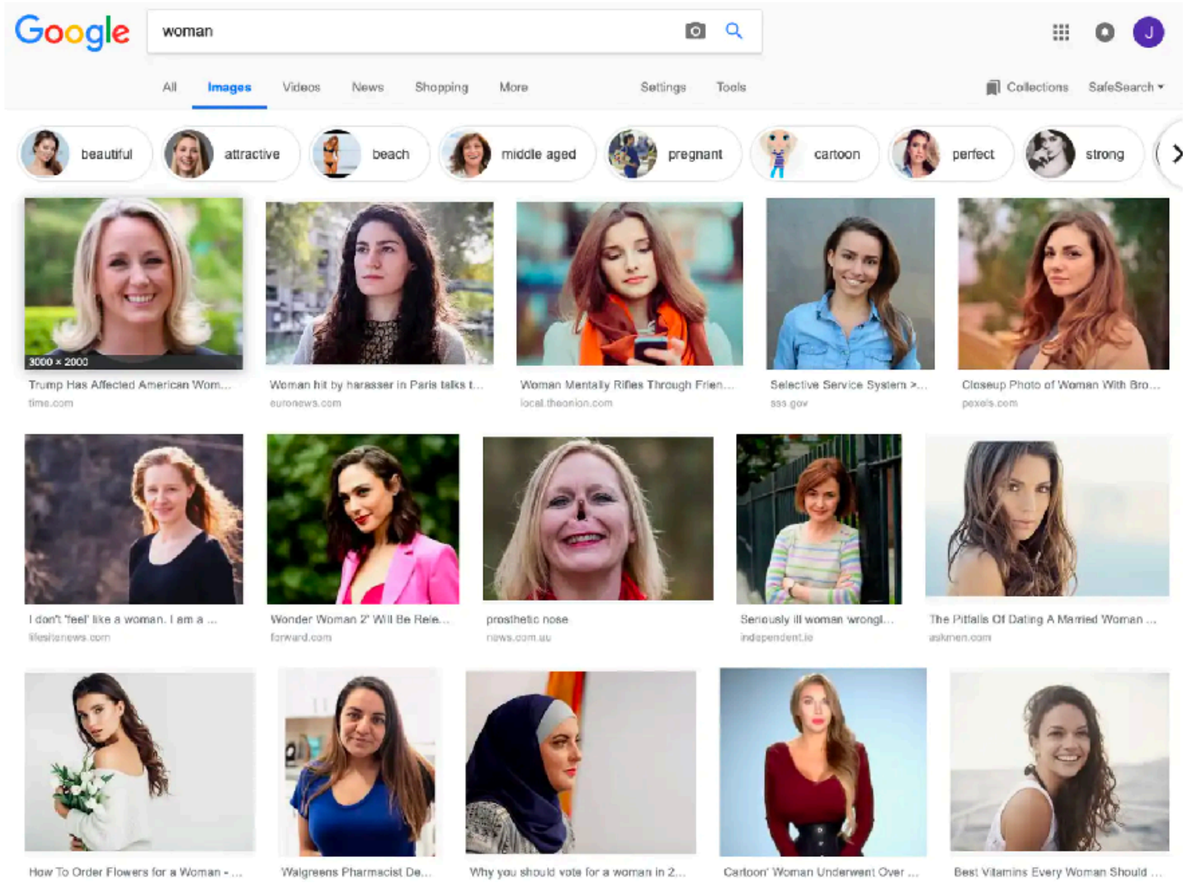
Computer science · Health care · Policy · Society

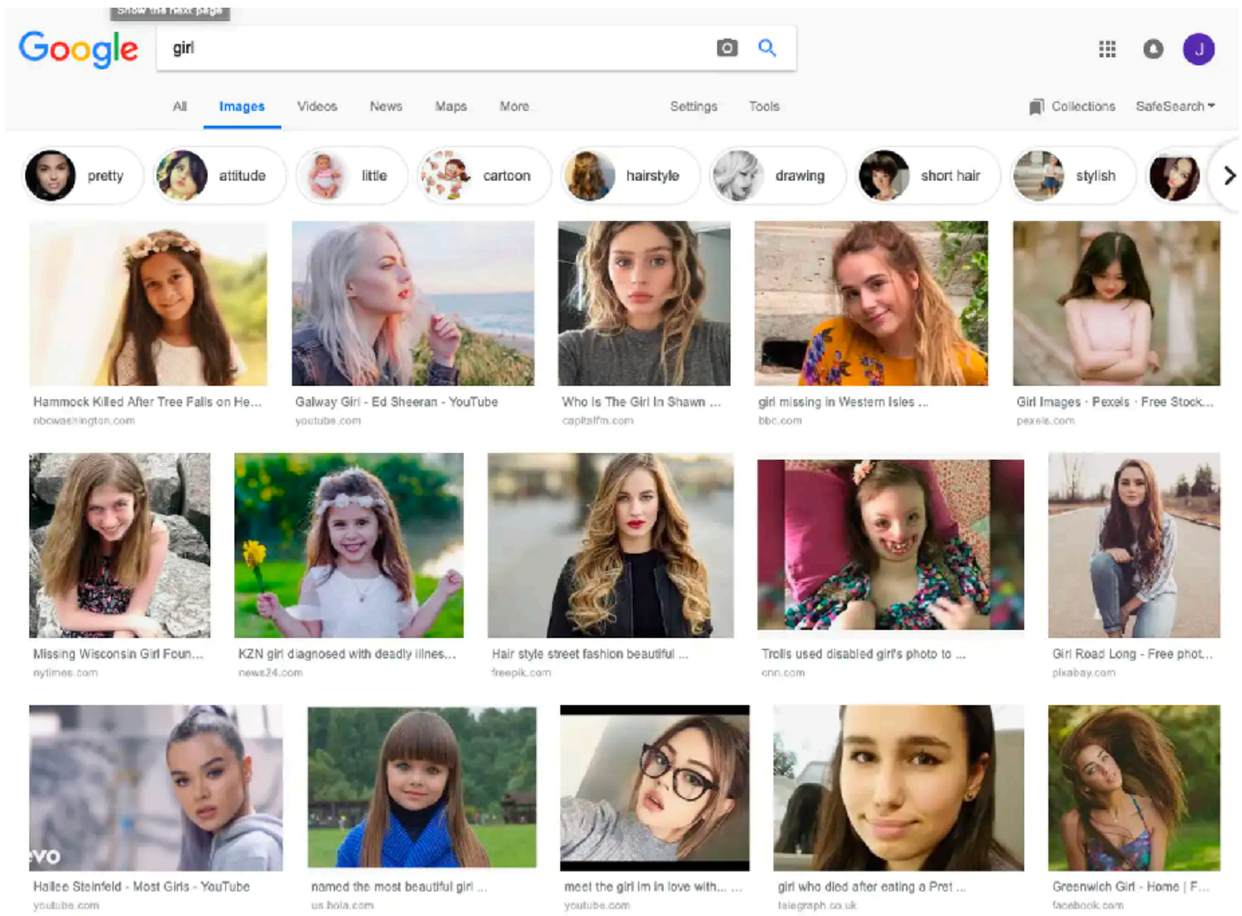


# AI could be racist

## Algorithmic bias



# AI could be also gender biased

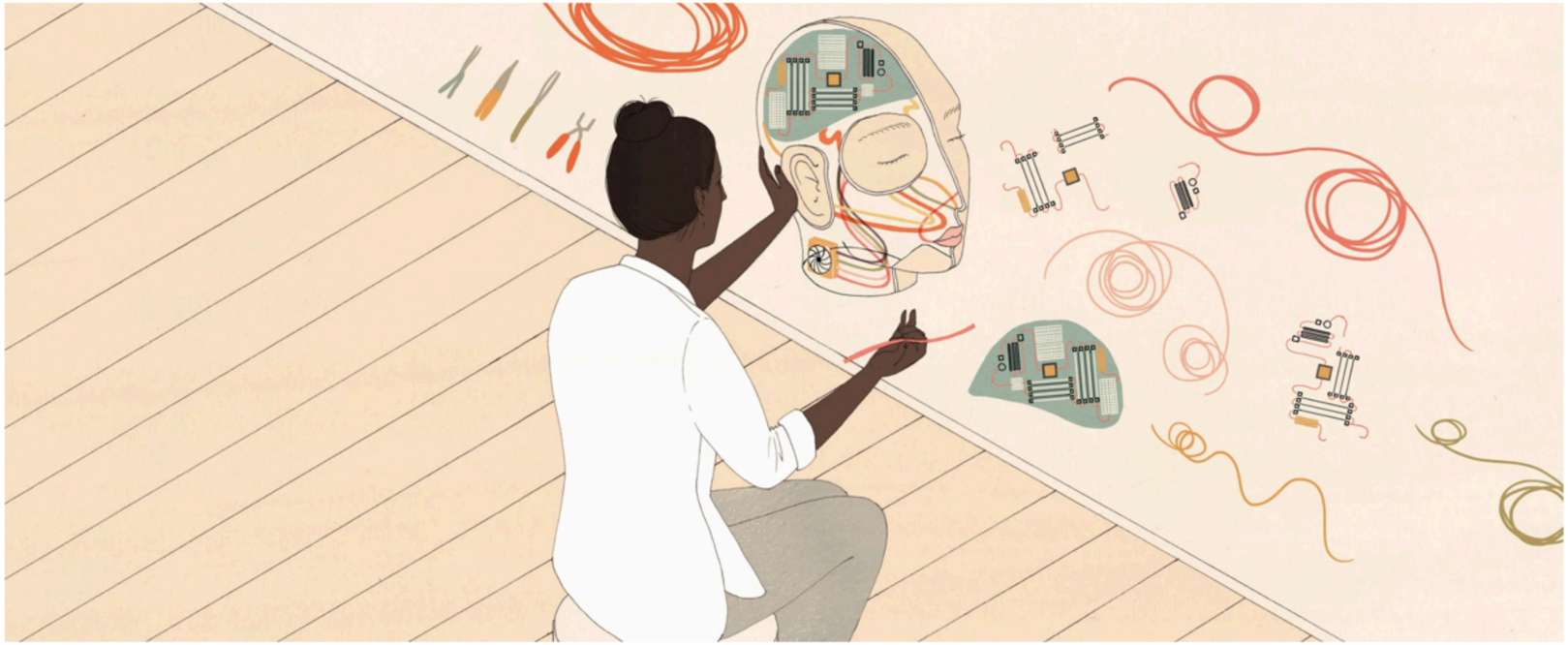
## Algorithmic bias


# AI could be also gender biased

## Algorithmic bias

### Dealing With Bias in Artificial Intelligence

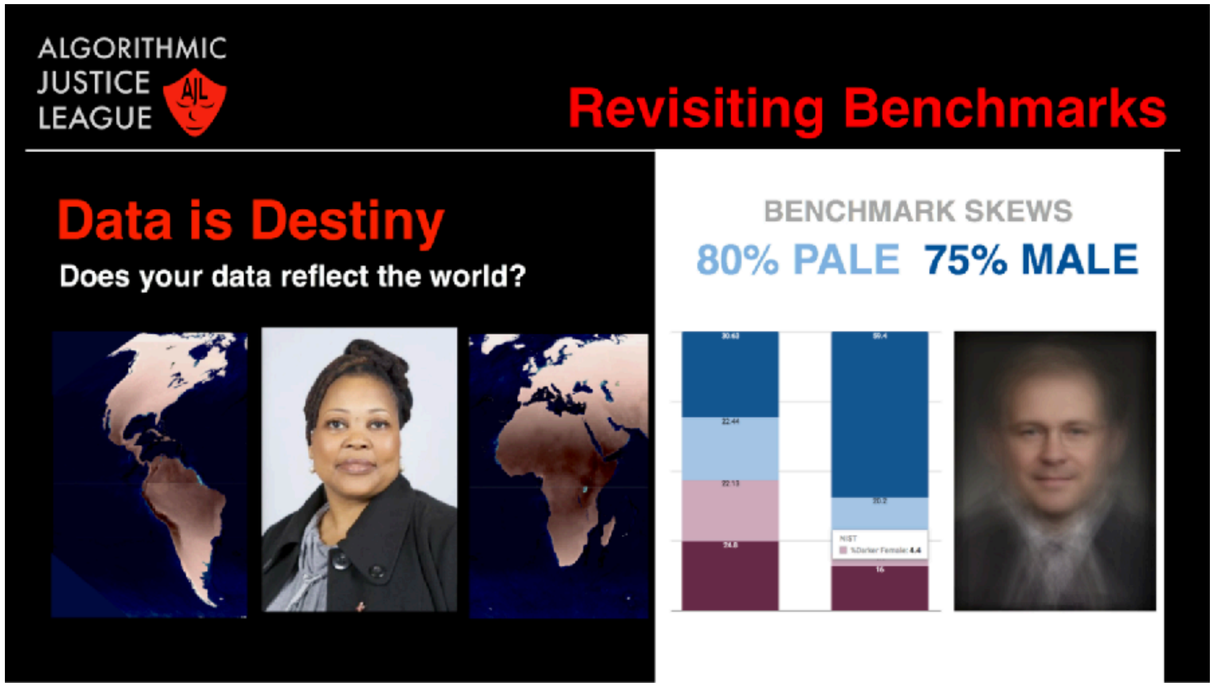
Three women with extensive experience in A.I. spoke on the topic and how to confront it.



Harriet Lee-Merrion

# What is the source of the problem?

## Data or Algorithms or Both?



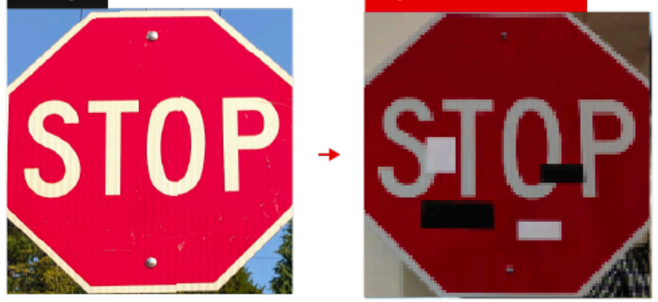

# AI/ML Systems can be easily fooled!

## What? Yes, it is true, and the implications could be massive!

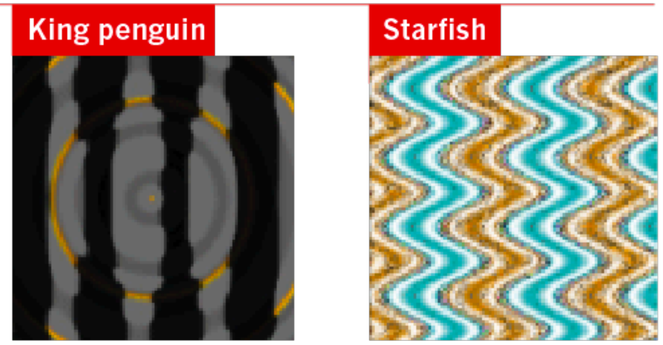
### FOOLING THE AI

Deep neural networks (DNNs) are brilliant at image recognition — but they can be easily hacked.

These stickers made an artificial-intelligence system read this stop sign as 'speed limit 45'.




Scientists have evolved images that look like abstract patterns — but which DNNs see as familiar objects.




©nature

### PERCEPTION PROBLEMS

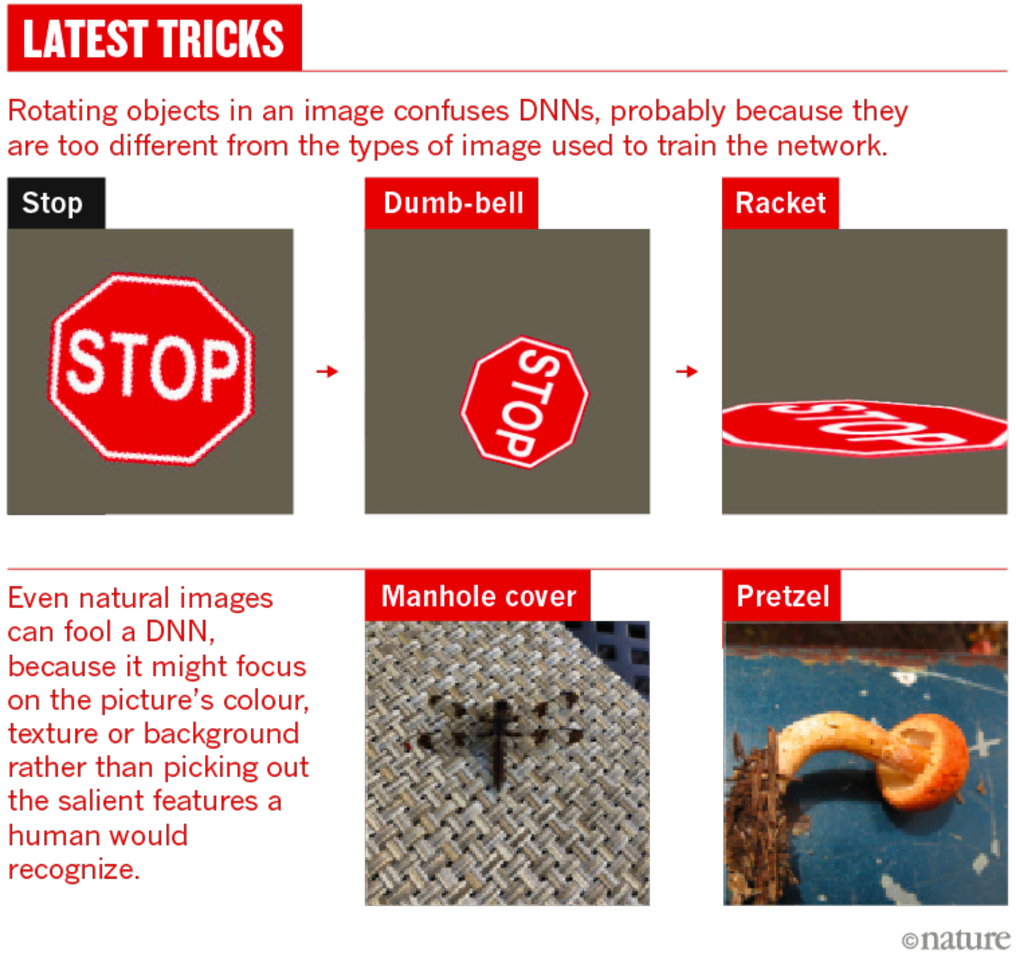
Adding carefully crafted noise to a picture can create a new image that people would see as identical, but which a DNN sees as utterly different.

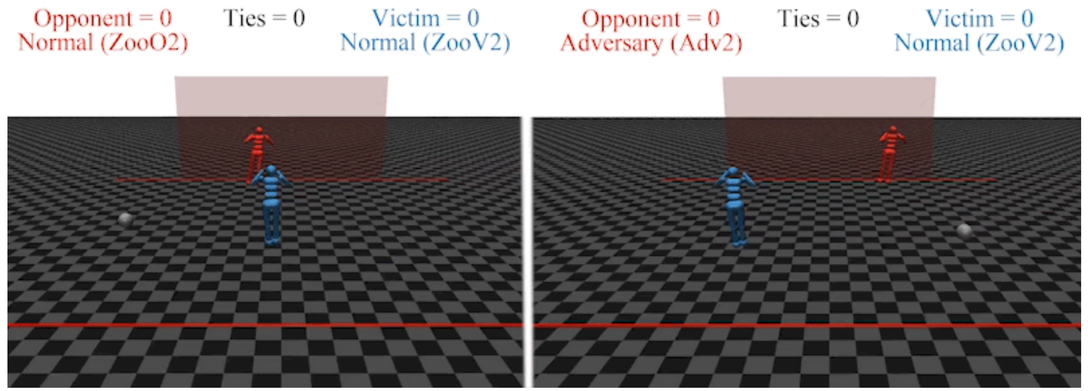

In this way, any starting image can be tweaked so a DNN misclassifies it as any target image a researcher chooses.


©nature

# AI/ML Systems can be easily fooled!

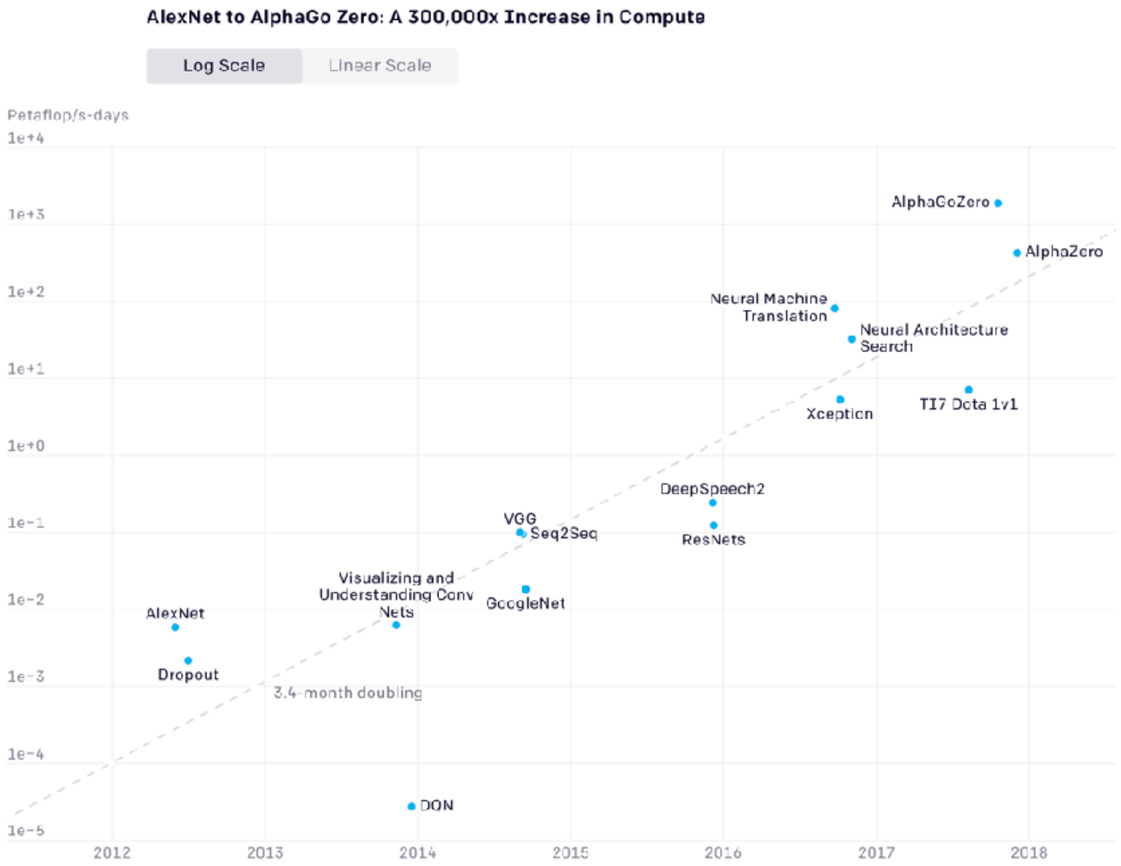
## What? Yes, it is true, and the implications could be massive!

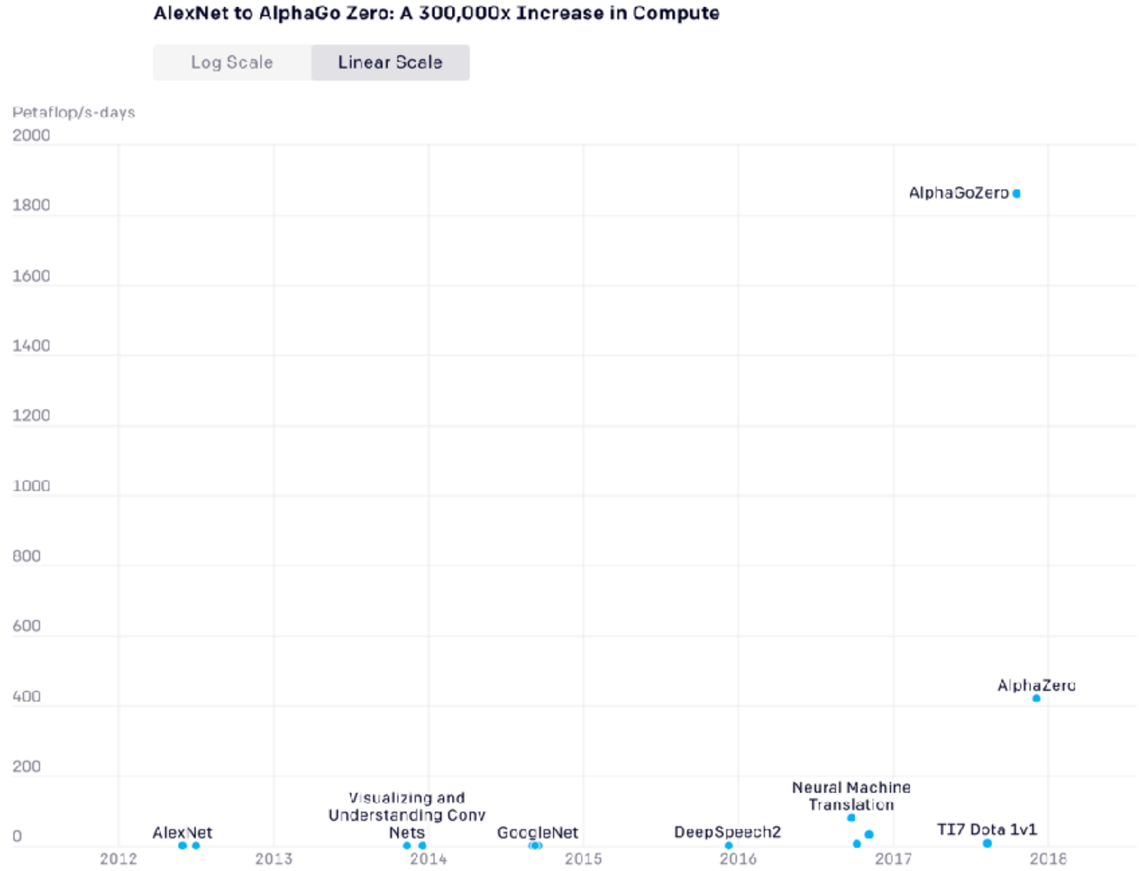


# AI and Compute

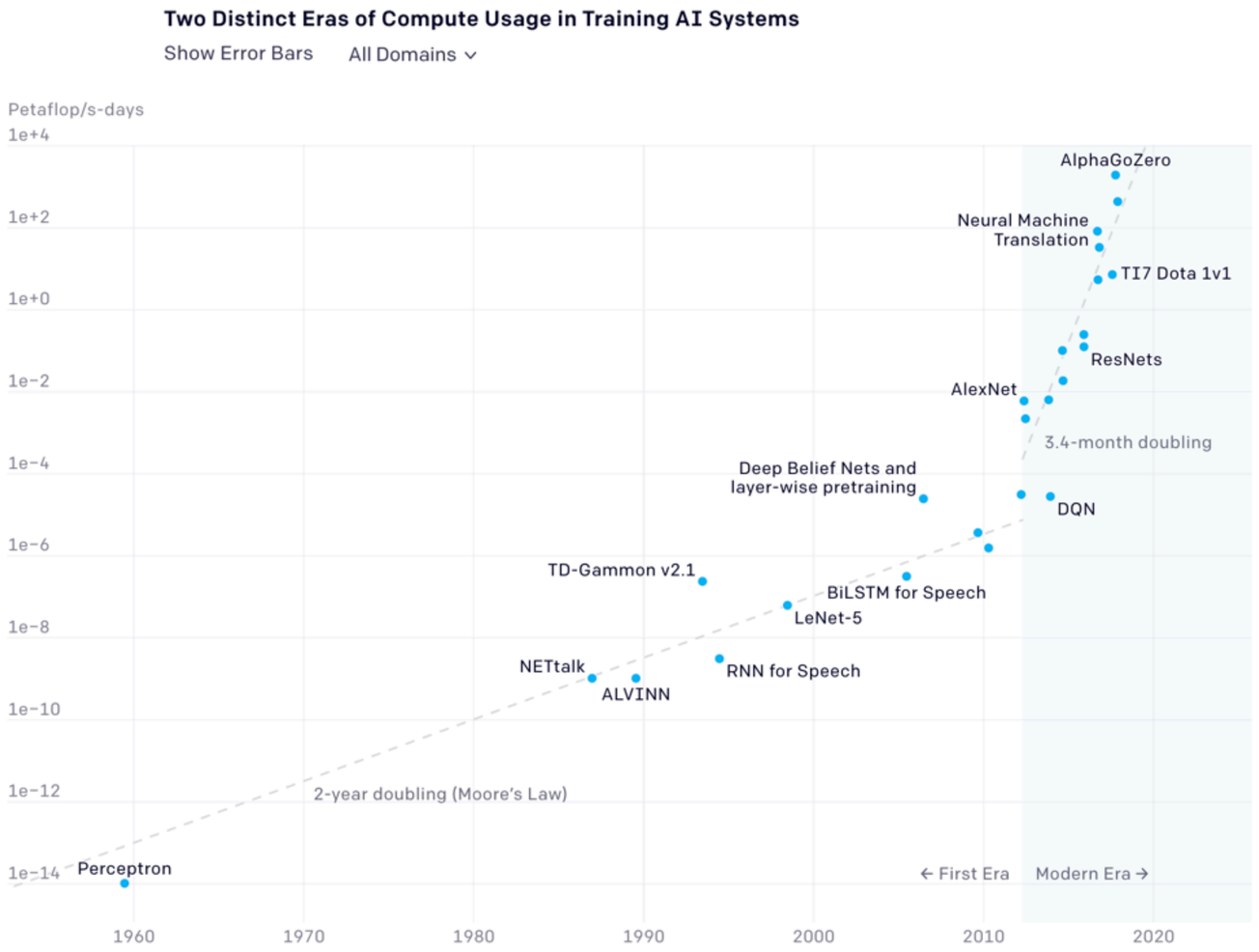
**The amount of compute used in the largest AI training runs has been increasing exponentially with a 3.4-month doubling time (by comparison, Moore's Law had a 2-year doubling period).**



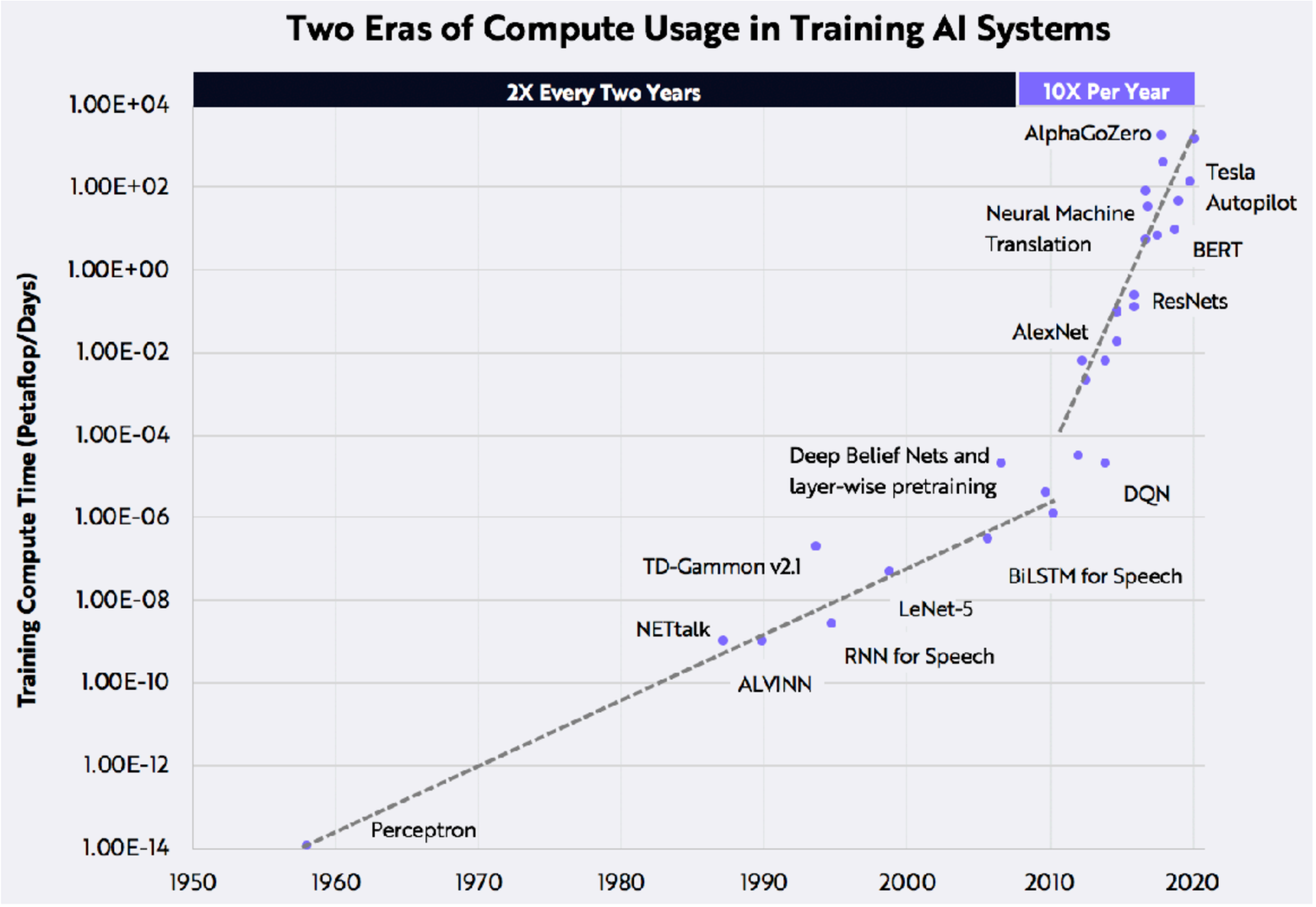
# AI and Compute

## Two Distinct Eras of Compute Usage in Training AI Systems



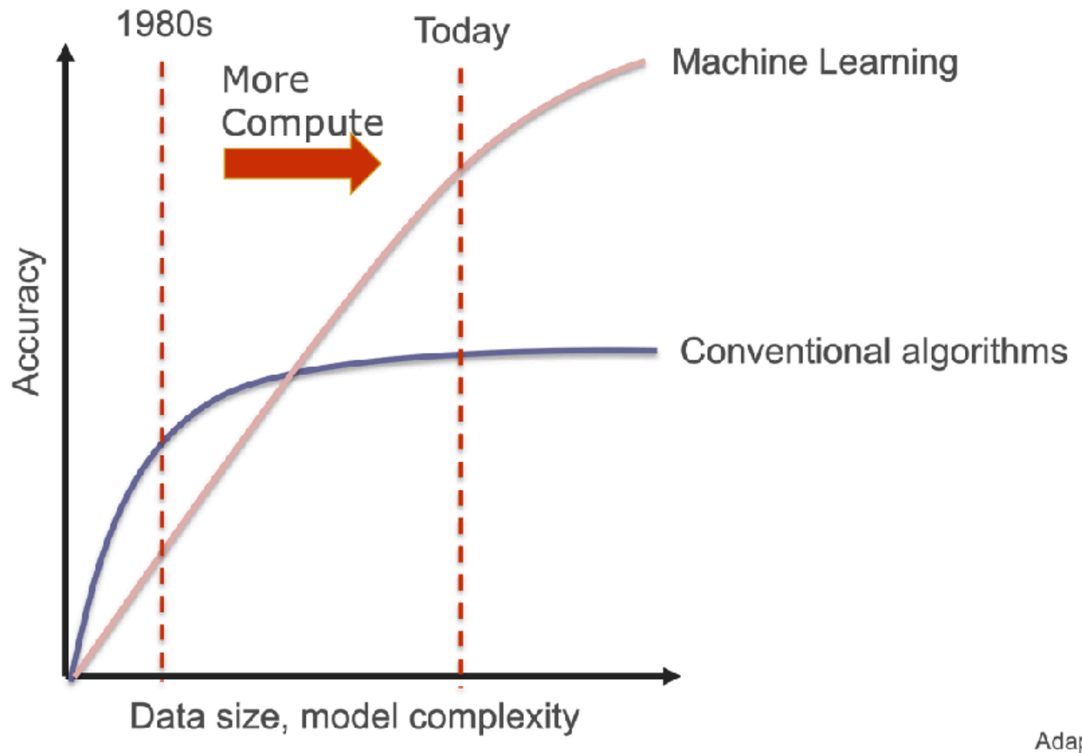
# AI and Compute

## Two Distinct Eras of Compute Usage in Training AI Systems



# AI and Compute

## Two Distinct Eras of Compute Usage in Training AI Systems




Adapted from Jeff Dean
HotChips 2017

# Machine Learning Systems

## Algorithmic Bias

NEWS · 24 OCTOBER 2019 · UPDATE 26 OCTOBER 2019

### Millions of black people affected by racial bias in health-care algorithms

Study reveals rampant racism in decision-making software used by US hospitals — and highlights ways to correct it.

Heidi Ledford

Black people with complex medical needs were less likely than equally ill white people to be referred to programmes that provide more personalized care. Credit: Ed Kashi/VII/Redux/eyevine

An algorithm widely used in US hospitals to allocate health care to patients has been systematically discriminating against black people, a sweeping analysis has found.

PDF version

RELATED ARTICLES

A fairer way forward for AI in health care

Bias detectives: the researchers striving to make algorithms fair

Can we open the black box of AI?

SUBJECTS

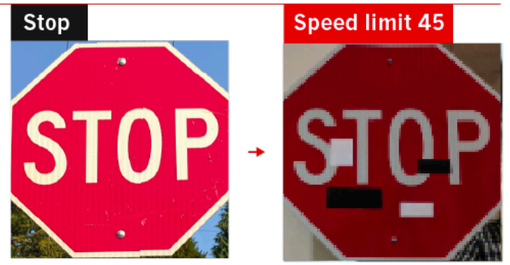
Computer science · Health care · Policy · Society

## Fooling AI Systems

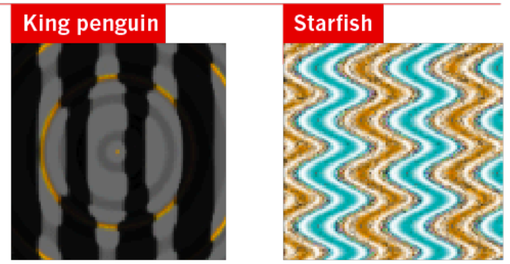
**FOOLING THE AI**

Deep neural networks (DNNs) are brilliant at image recognition — but they can be easily hacked.

These stickers made an artificial-intelligence system read this stop sign as 'speed limit 45'.

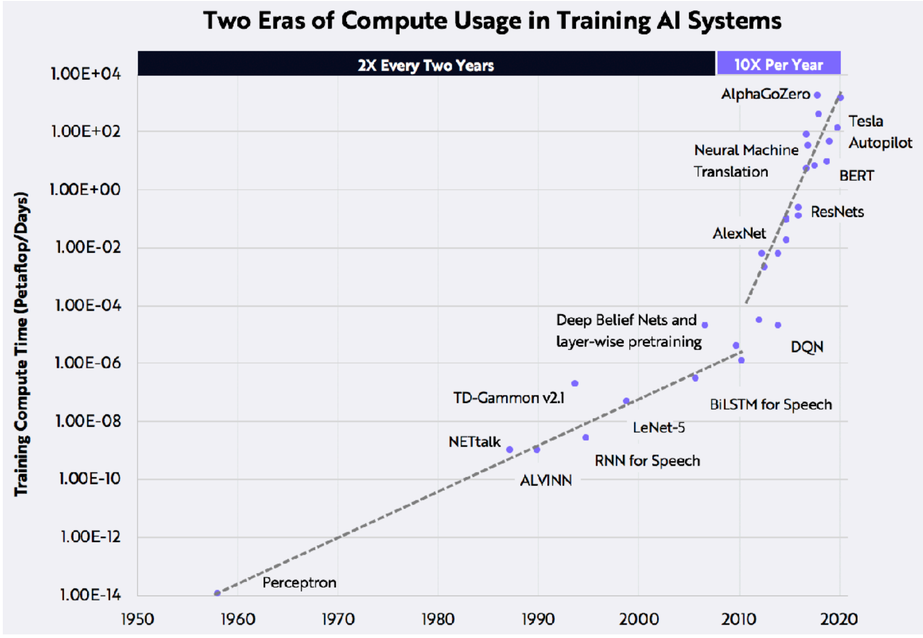



Scientists have evolved images that look like abstract patterns — but which DNNs see as familiar objects.

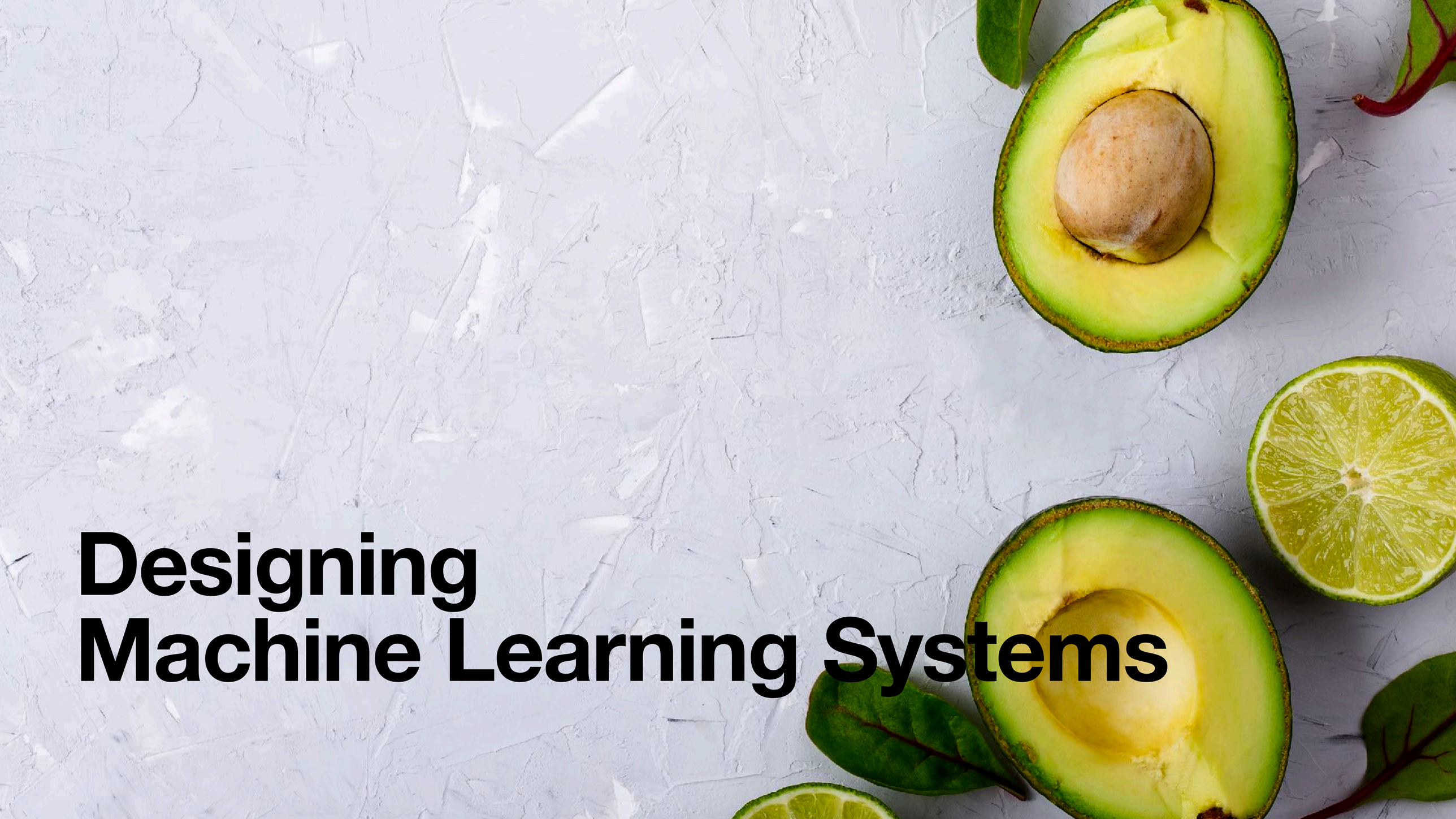



©nature

## AI and Compute

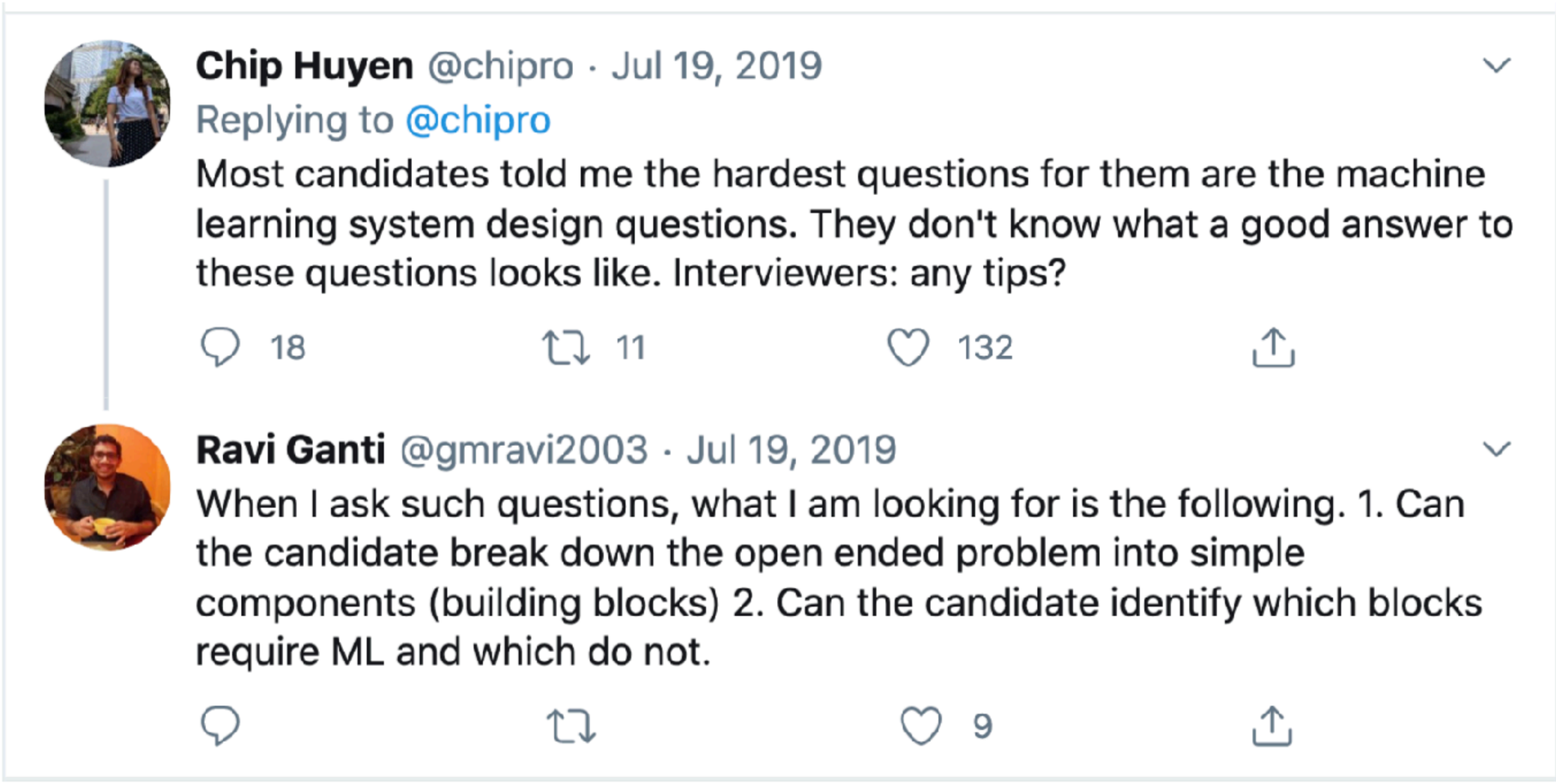


# Designing Machine Learning Systems



# What is missing?

## The gap between ML Research and Production



**Chip Huyen** @chipro · Jul 19, 2019

Replying to @chipro

Most candidates told me the hardest questions for them are the machine learning system design questions. They don't know what a good answer to these questions looks like. Interviewers: any tips?

18 11 132

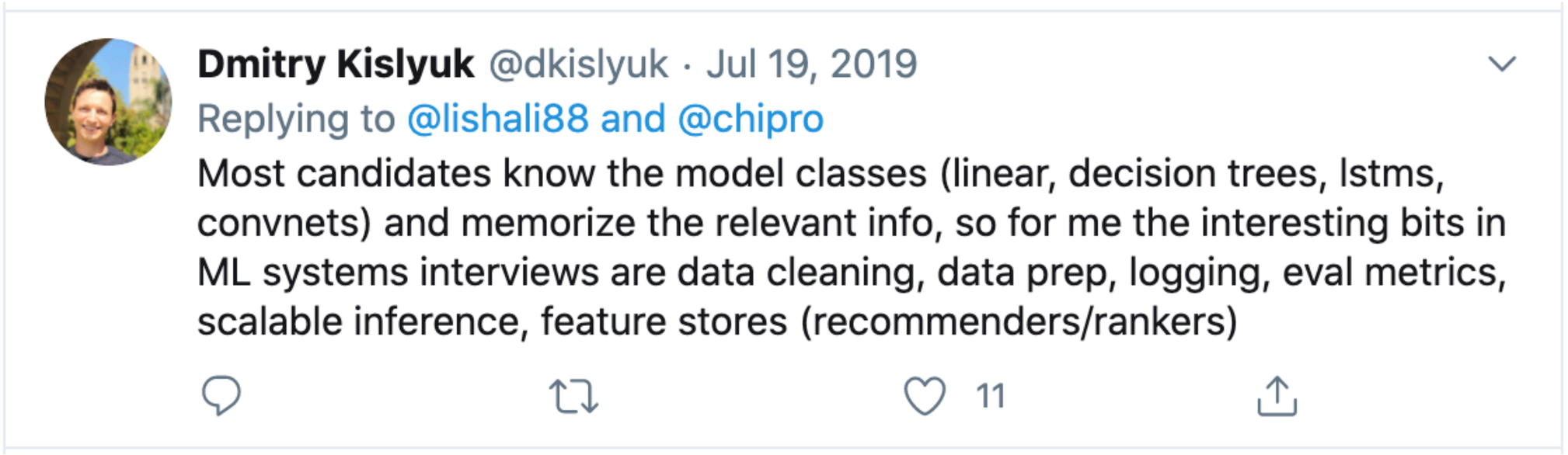
**Ravi Ganti** @gmravi2003 · Jul 19, 2019

When I ask such questions, what I am looking for is the following. 1. Can the candidate break down the open ended problem into simple components (building blocks) 2. Can the candidate identify which blocks require ML and which do not.

9

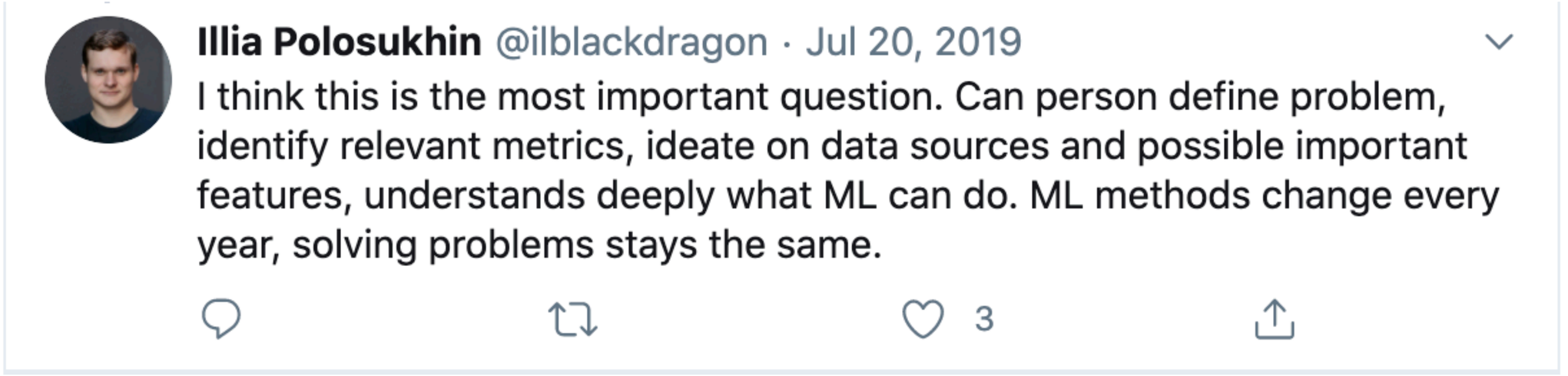
# What is missing?

## The gap between ML Research and Production



# What is missing?

## The gap between ML Research and Production



# In ML Systems, only a small fraction is comprised of actual ML code

## Hidden Technical Debt in Machine Learning Systems

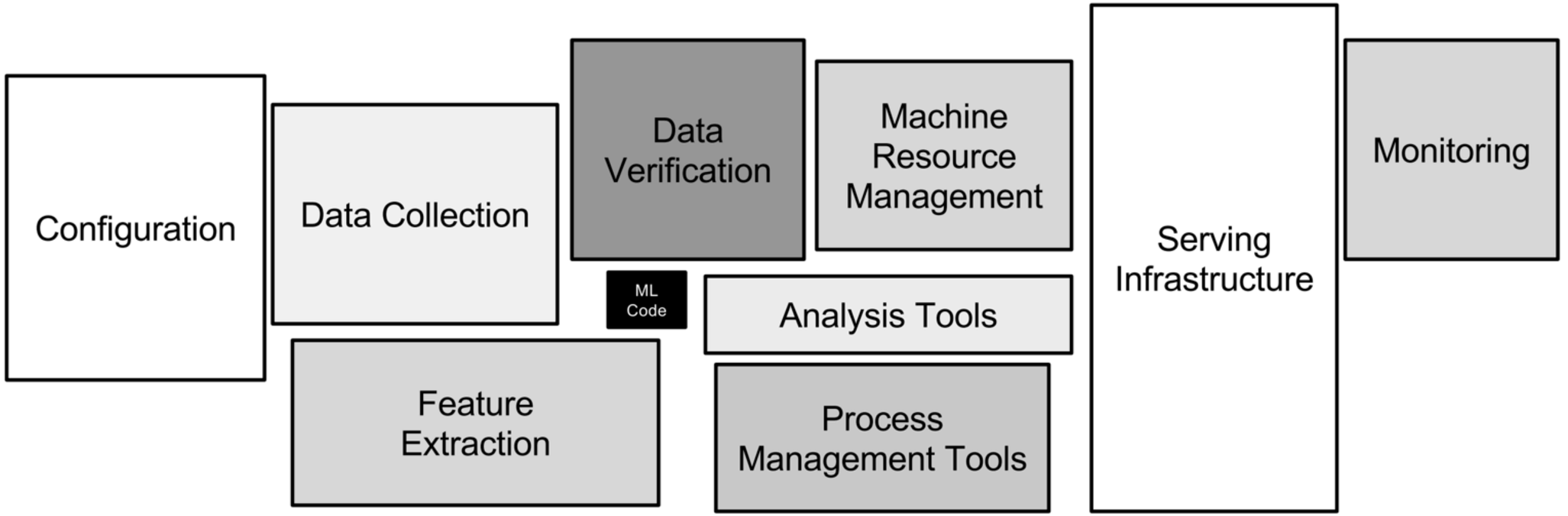
**D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips**
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.

**Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison**
{ebner,vchaudhary,mwyoung,jfcrespo,dennison}@google.com
Google, Inc.

### Abstract

Machine learning offers a fantastically powerful toolkit for building useful complex prediction systems quickly. This paper argues it is dangerous to think of these quick wins as coming for free. Using the software engineering framework of *technical debt*, we find it is common to incur massive ongoing maintenance costs in real-world ML systems. We explore several ML-specific risk factors to account for in system design. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and a variety of system-level anti-patterns.

# A vast array of surrounding infrastructure and processes is needed to support evolution of ML systems



# Technical debt that can accumulate in ML systems

- Data dependencies
- Model complexity
- Reproducibility
- Testing
- Monitoring
- Configuration issues
- External changes

# Systems issues in ML Systems

## Understanding the Nature of System-Related Issues in Machine Learning Frameworks: An Exploratory Study

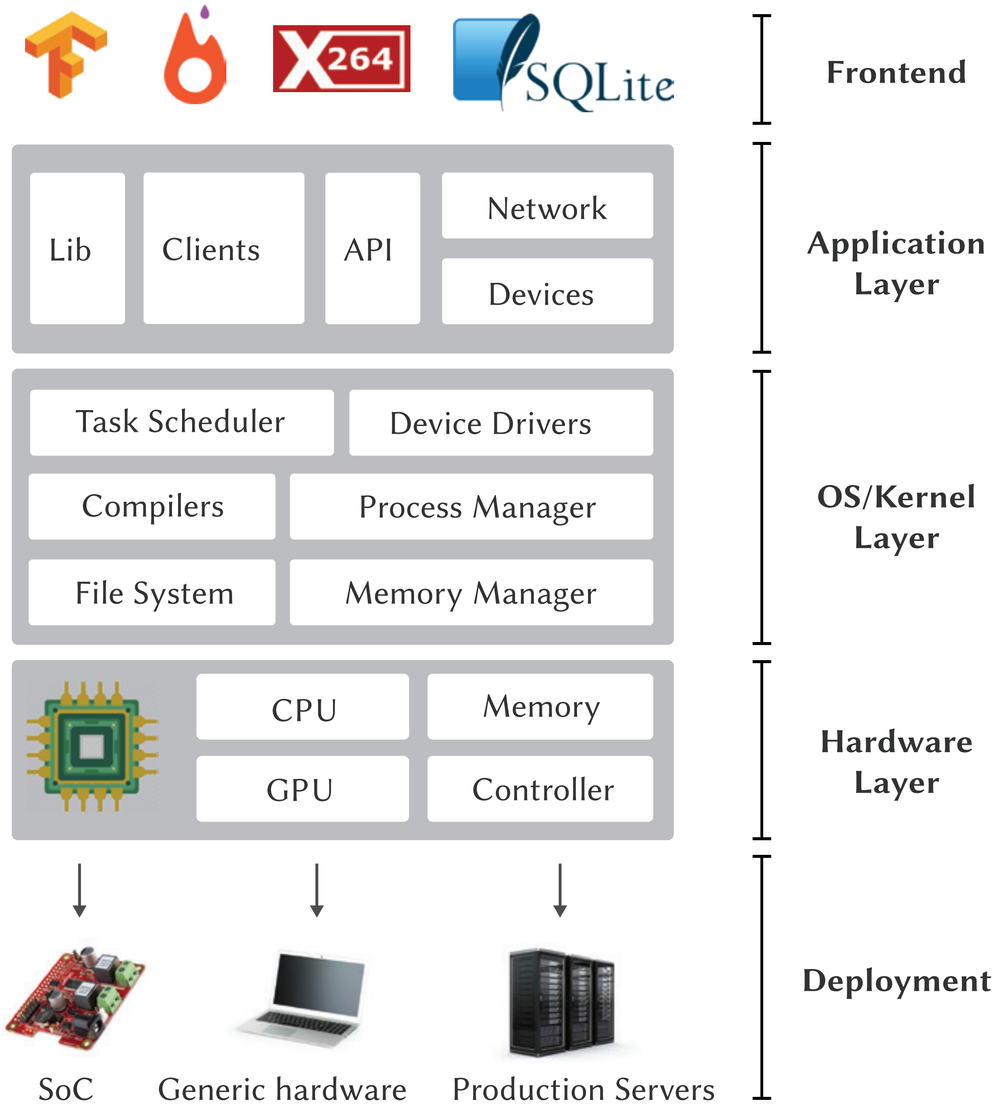
Yang Ren
University of South Carolina
USA

Gregory Gay
Chalmers and the University of Gutenberg
Sweden

Christian Kästner
Carnegie Mellon University
USA

Pooyan Jamshidi
University of South Carolina
USA

# System = Software + Middleware + Hardware

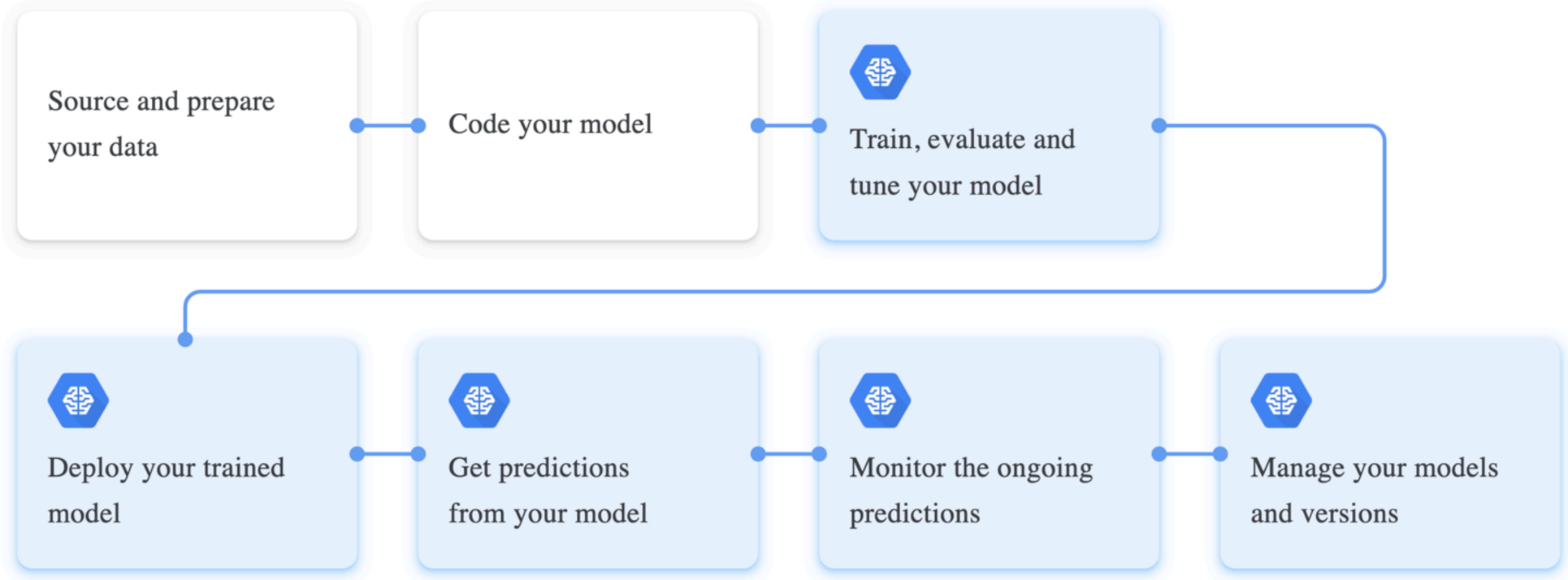


# Systems issues in ML Systems

| Category (Short Title) | Definition |
|---|---|
| API Mismatch (API) | Change to API version or mixed usage of APIs leading to performance degradation. |
| Compilation Error (Compl) | Failure to compile the source code. |
| Configuration Error (Config) | Configuration settings lead to performance degradation or error. |
| Connection Error (Conn) | Unexpected or wrongly-formatted connection request leads to error. |
| Data Race (Race) | Two or more threads access the same memory location concurrently. |
| Execution Error (Exec) | Unexpected error leads to the execution process crashing. |
| Hardware-Architecture Mismatch (HA) | Unfit hardware architecture leads to performance degradation or compilation error. |
| Memory Allocation (MA) | Memory allocation leads to performance degradation. |
| I/O Slowdown (I/O) | Issues with I/O processes lead to performance degradation. |
| Memory Leak (ML) | A failure in a program to release memory. |
| Model Conversion (Conv) | Performance degradation due to type conversion/cast. |
| Multi-Threading Error (MT) | Performance degradation due to thread interaction. |
| Performance Regression (PR) | Performance degradation after a change to the system. |
| Slow Synchronization (SYNC) | Synchronization between components leads to performance degradation. |
| Unexpected Resource Usage (RU) | Unusual system resource usage or requests leading to error or performance degradation. |

# The Building Process of ML Systems

## Continuous Delivery for ML Systems



# A Machine Learning System is more than just a model

## Change in ML Systems

**Data**

Schema

Sampling over Time

Volume

**Model**

Algorithms

More Training

Experiments

**Code**

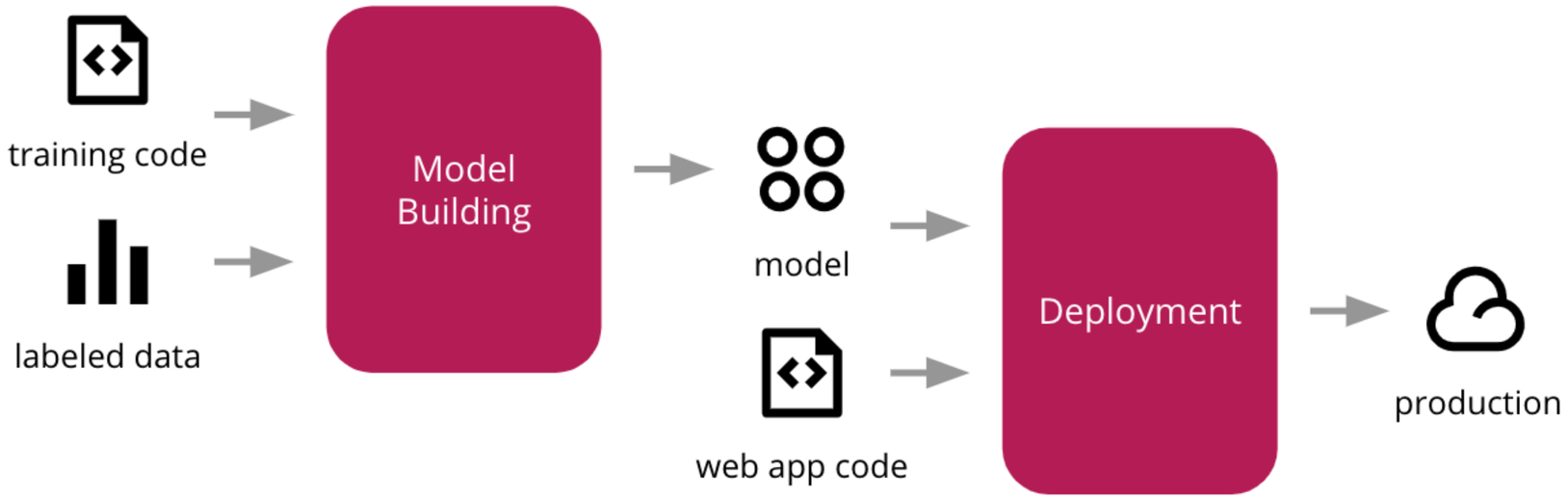
Business Needs

Bug Fixes

Configuration


Continuous Delivery for Machine Learning

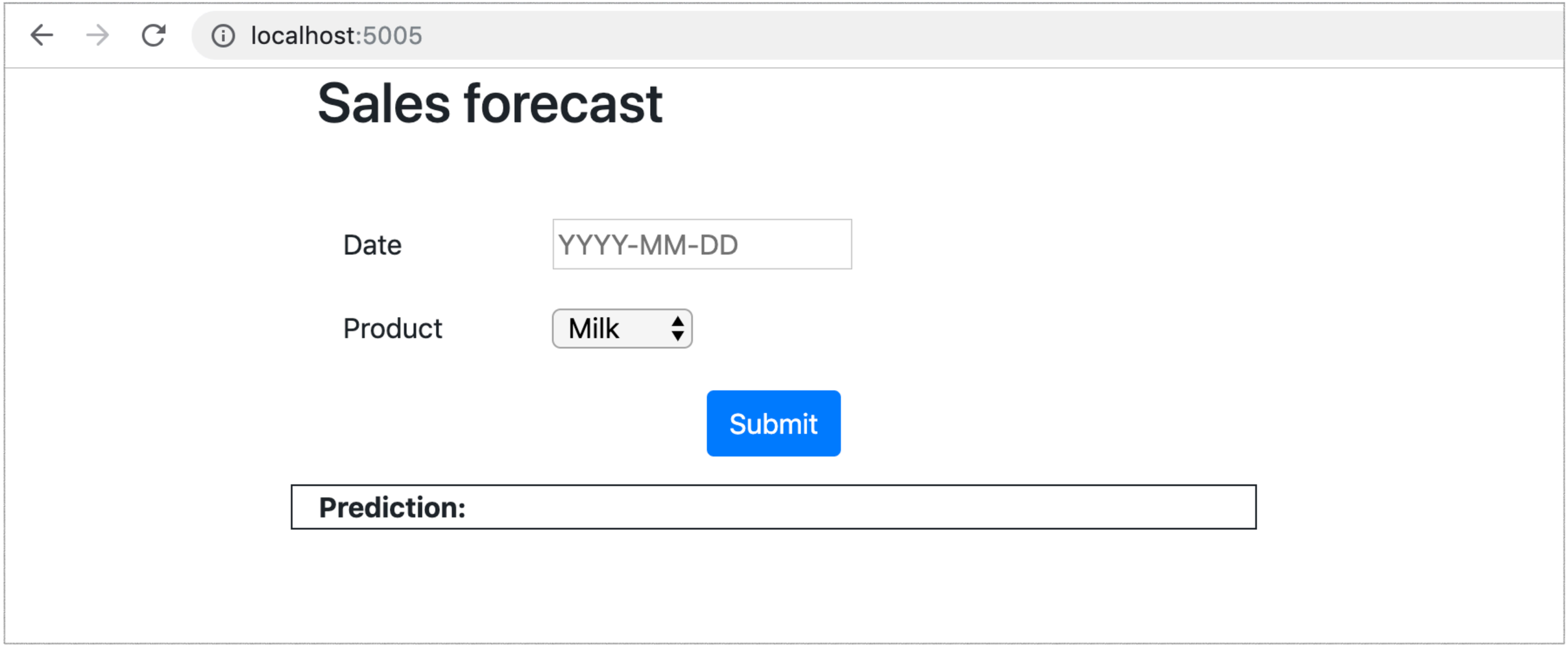
# Train ML model, integrate it with an application, and deploy into production





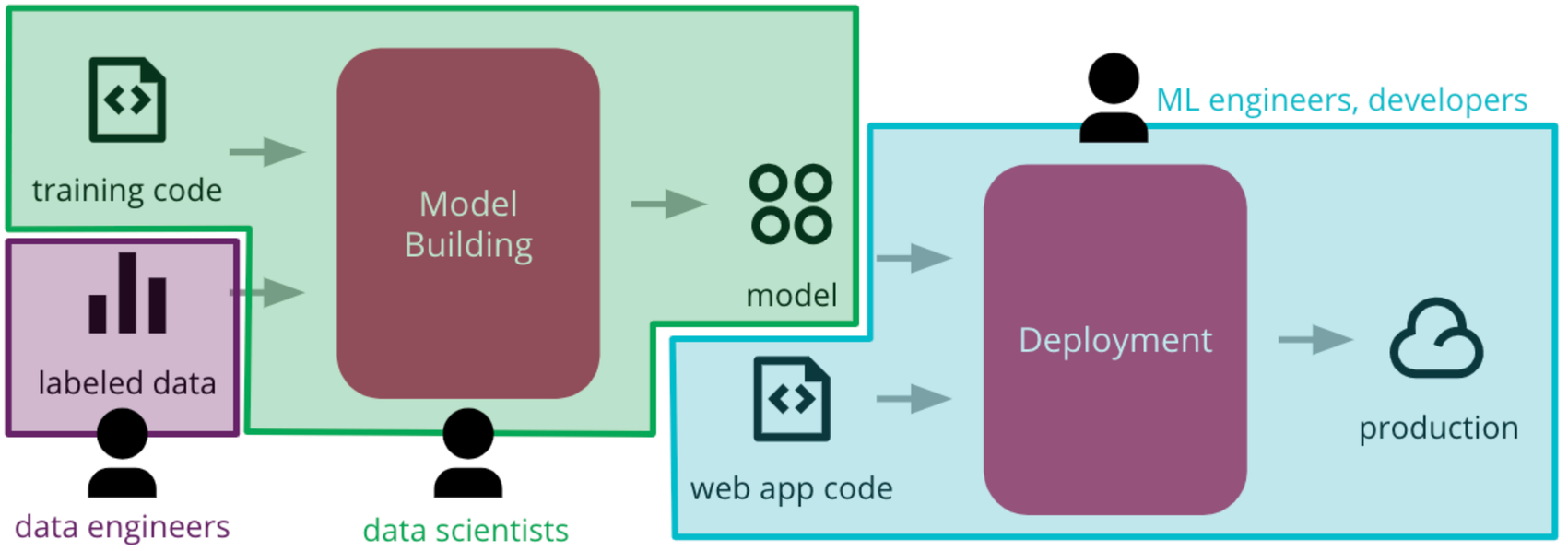
Continuous Delivery for Machine Learning

# ML model behind a web application

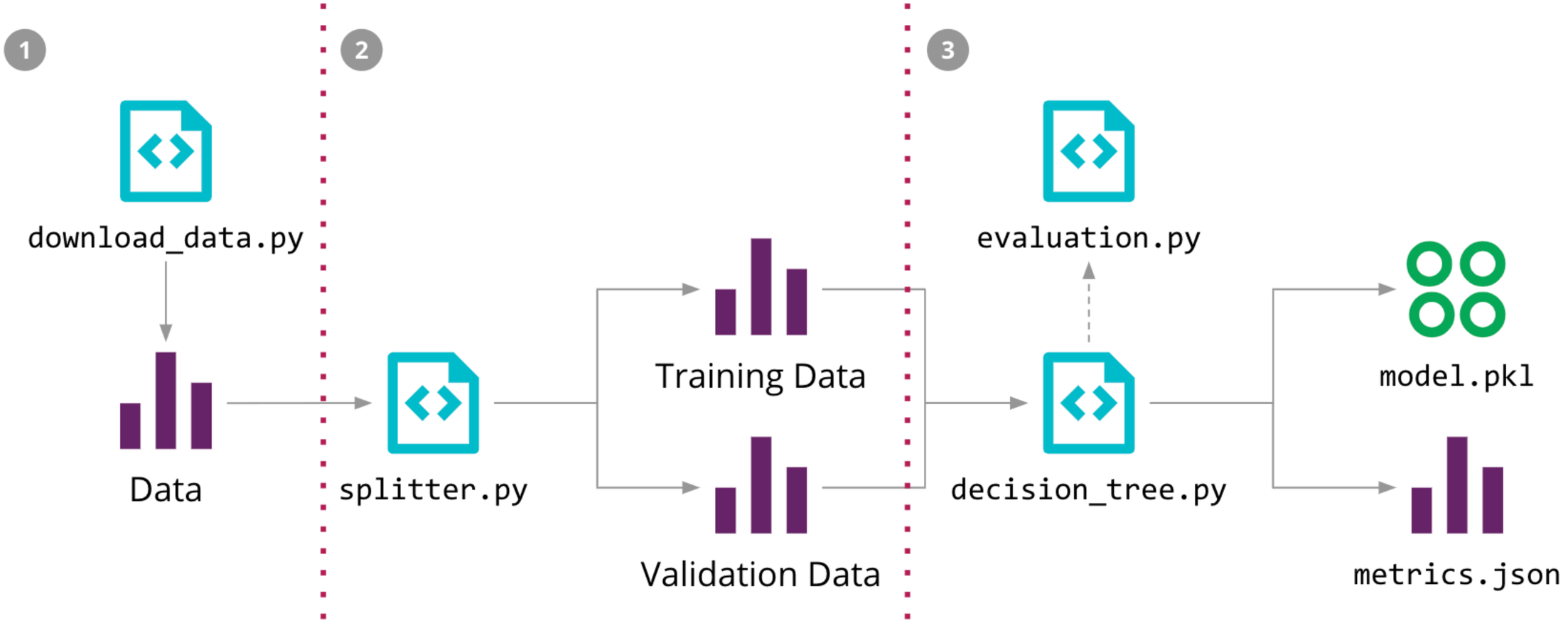


# Challenges

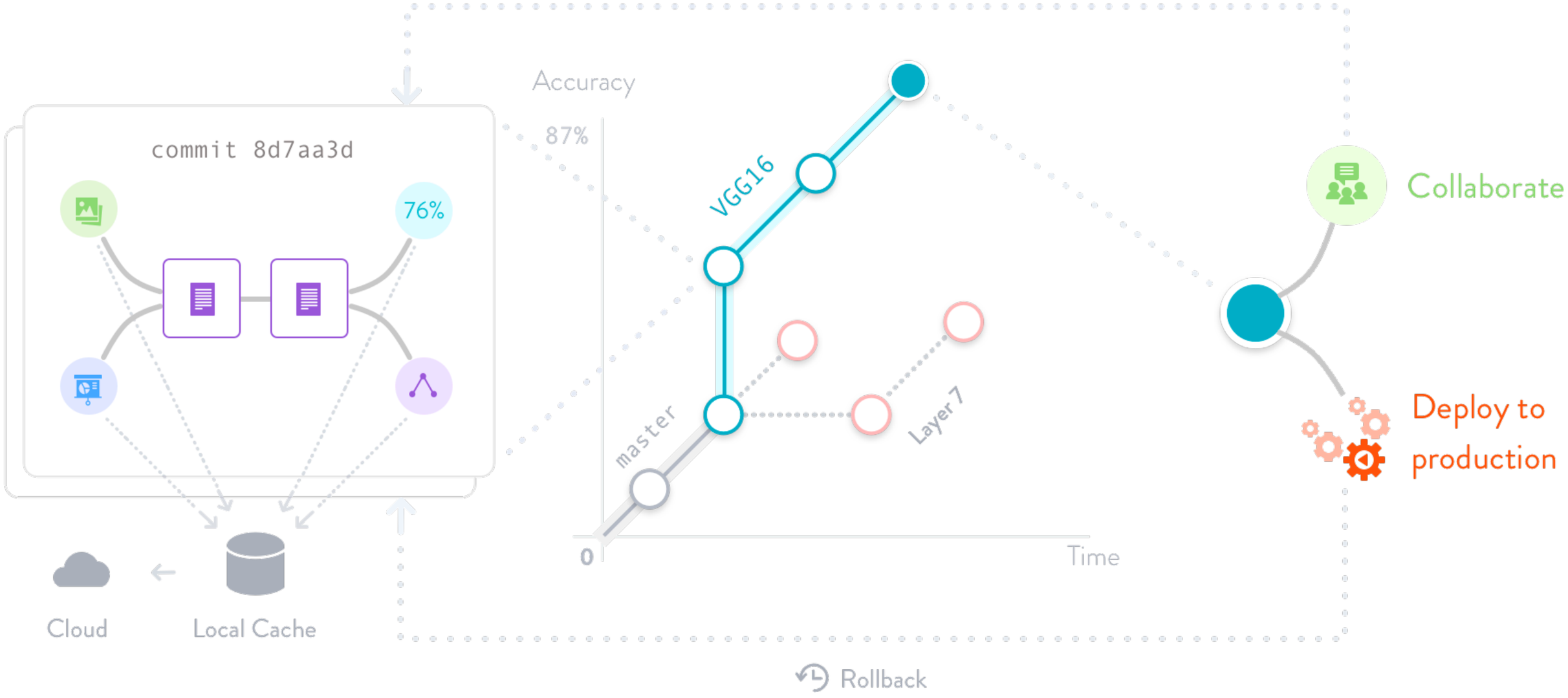
- Throw over the wall
- Models that only work in a lab environment
- Even if make it to production, they become stale and hard to update
- Reproducible and auditable



# ML pipeline


1
2
3
download_data.py
Data
splitter.py
Training Data
Validation Data
evaluation.py
decision_tree.py
model.pkl
metrics.json

# Configure ML pipeline: DVC tracks ML models and data sets



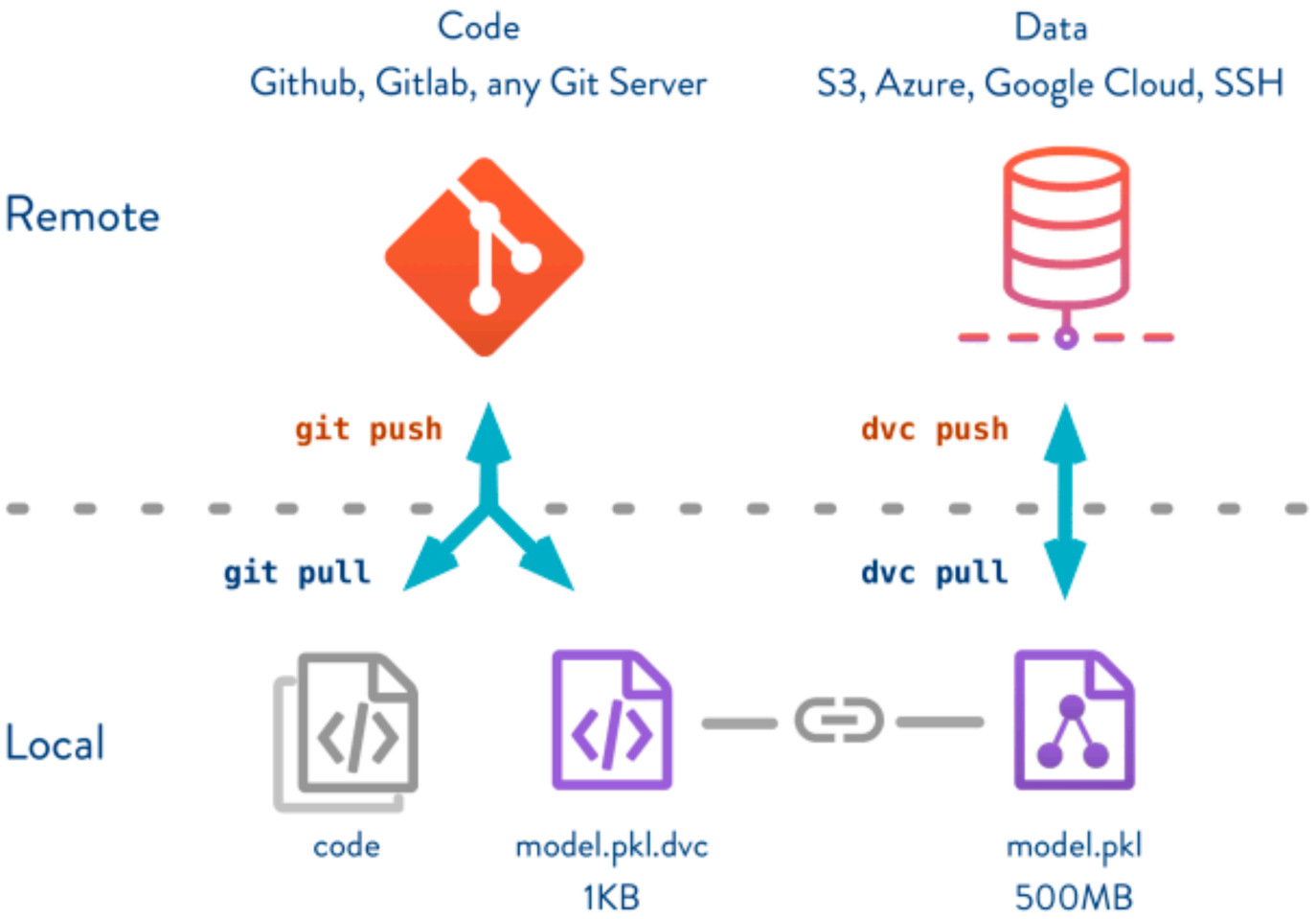
# Configure ML pipeline: DVC tracks ML models and data sets

```
dvc run -f input.dvc \ ❶
  -d src/download_data.py -o data/raw/store47-2016.csv python src/download_data.py
dvc run -f split.dvc \ ❷
  -d data/raw/store47-2016.csv -d src/splitter.py \
  -o data/splitter/train.csv -o data/splitter/validation.csv python src/splitter.py
dvc run ❸
  -d data/splitter/train.csv -d data/splitter/validation.csv -d src/decision_tree.py \
  -o data/decision_tree/model.pkl -M results/metrics.json python src/decision_tree.py
```

# Configure ML pipeline: DVC tracks ML models and data sets

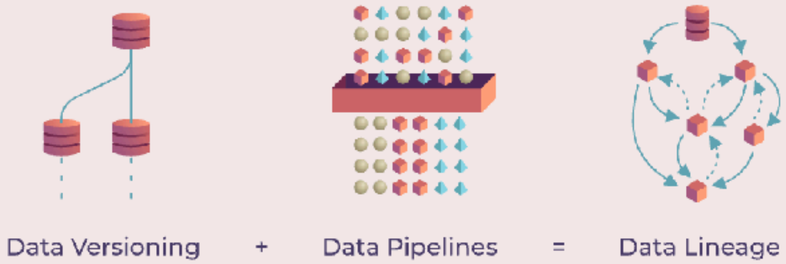
- Each run will create a file, that can be committed to version control
- DVC allows other people to reproduce the entire ML pipeline, by executing the *dvc repro* command.
- Once we find a suitable model, we will treat it as an artifact that needs to be *versioned* and *deployed* to production.
- With DVC, we can use the *dvc push* and *dvc pull* commands to publish and fetch it from *external storage*.

# Configure ML pipeline: DVC tracks ML models and data sets



# There are other open source tools for versioning

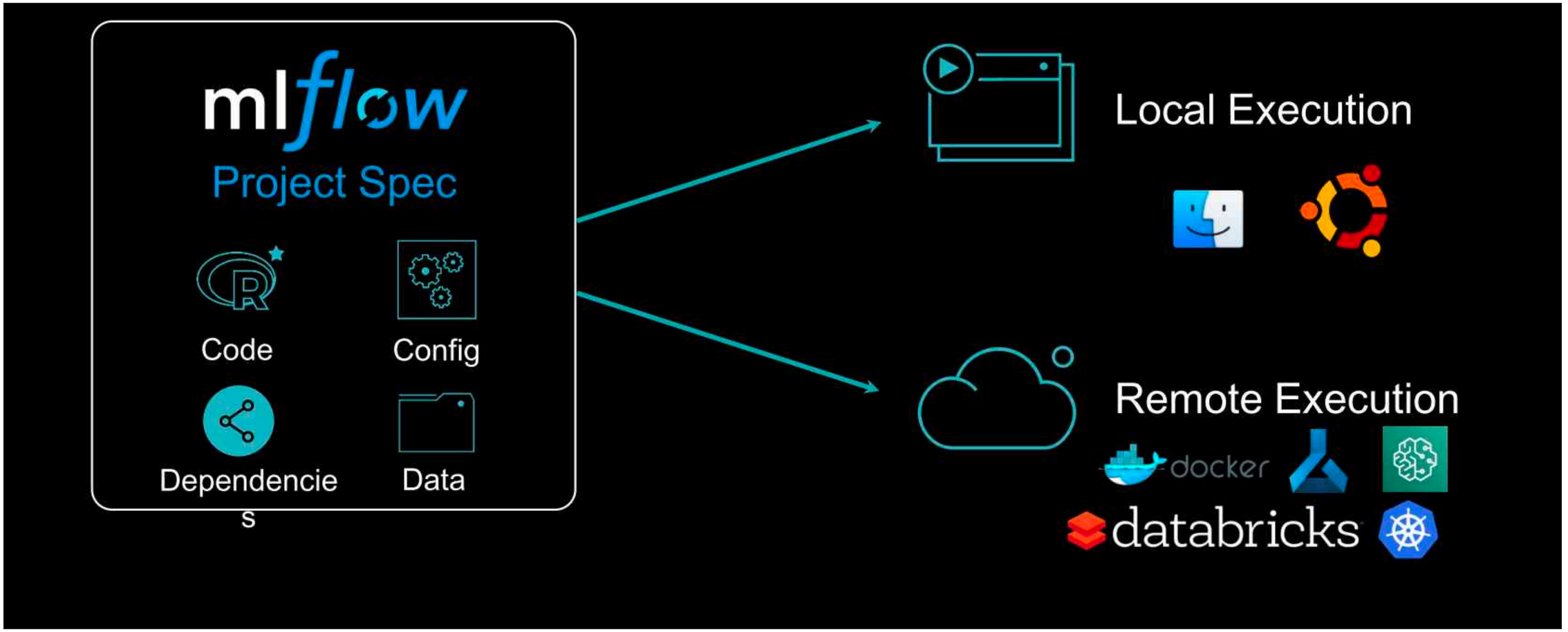
## Pachyderm





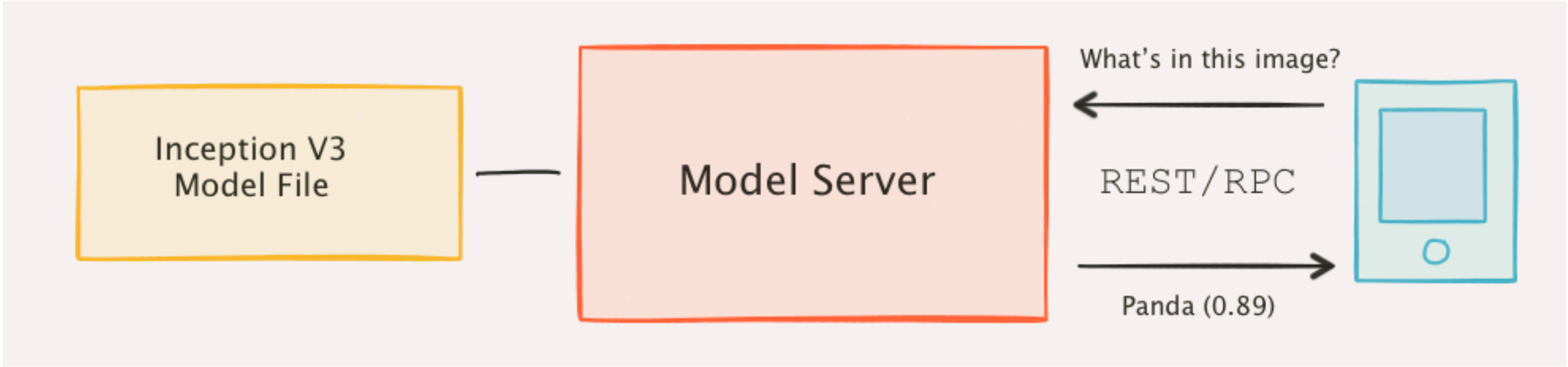
# There are other open source tools for versioning

## MLflow



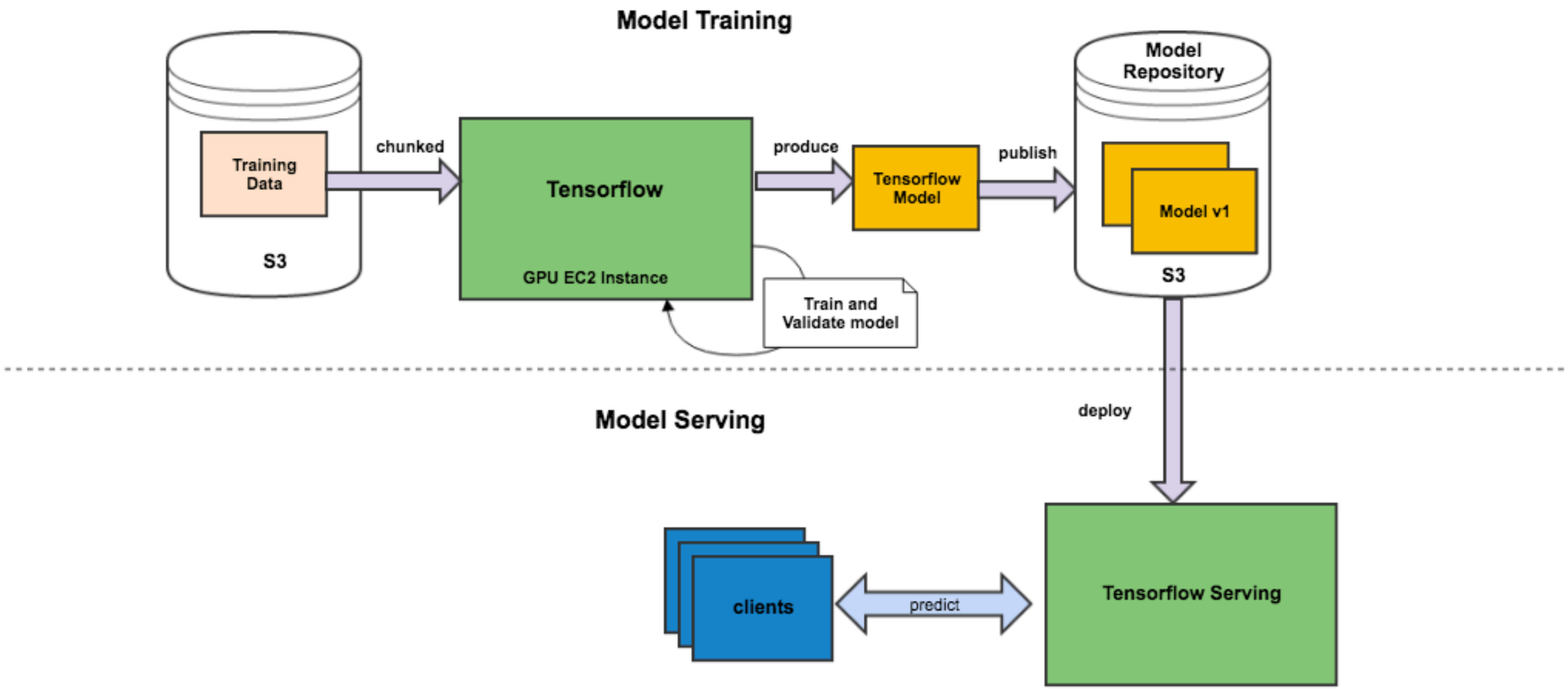
# Model Serving

## Abstract level



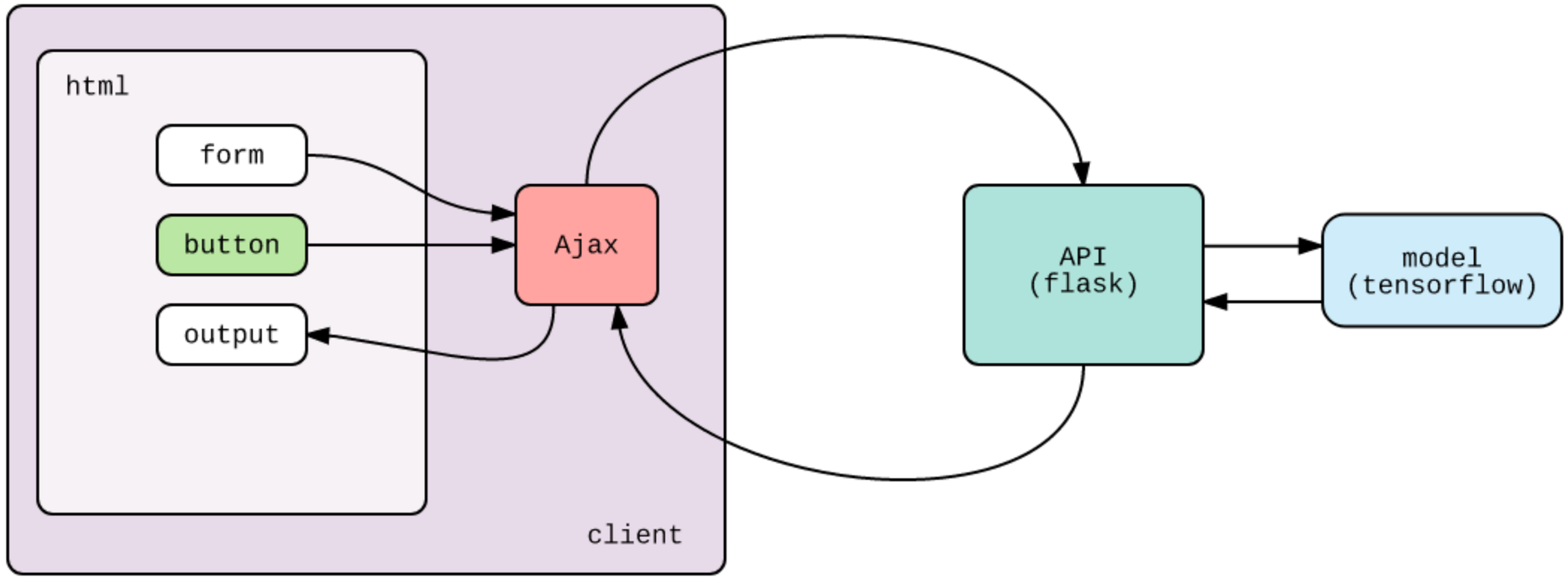
# Model Serving

## TF Serving



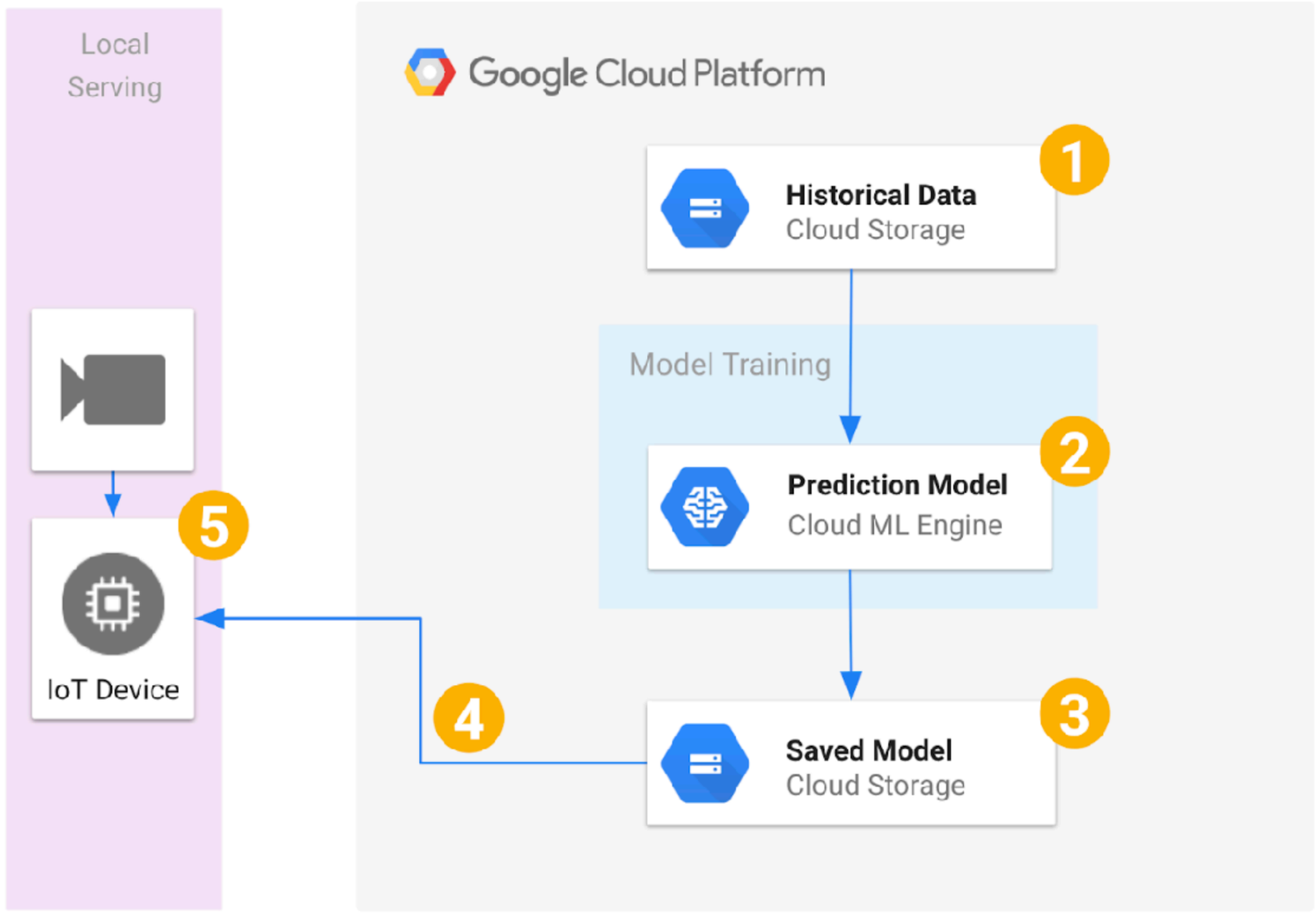
# Model Serving

## Web app



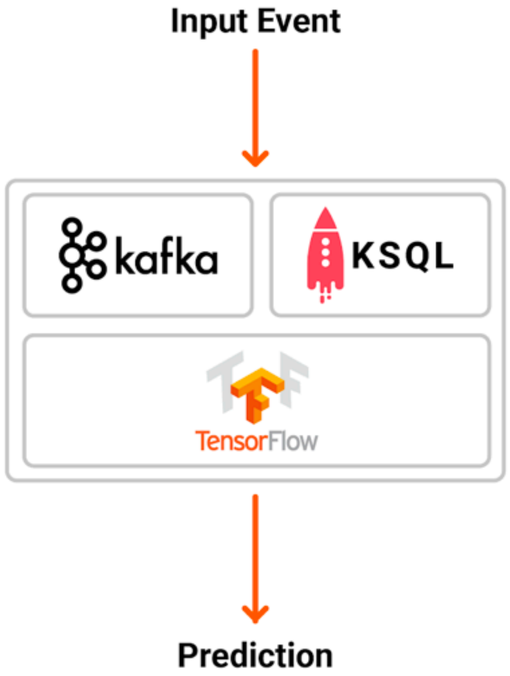
# Model Serving

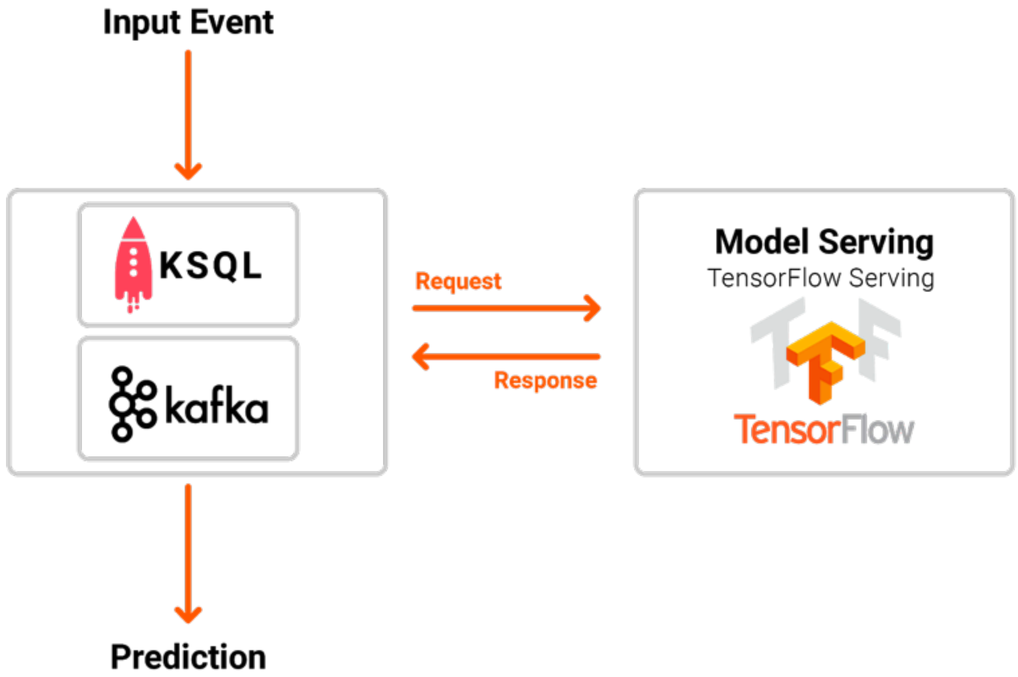
## Internet of Thing



# Model Serving

## Stream Processing System



# Model Serving

## Embedded model

- Simple approach
- You treat the model artifact as a dependency that is built and packaged within the consuming application.
- You can treat the application artifact and version as being a combination of the application code and the chosen model.

# Model Serving

## Model deployed as a separate service

- The model is wrapped in a service that can be deployed independently of the consuming applications.
- This allows updates to the model to be released independently, but it can also introduce latency at inference time
- There will be some sort of remote invocation required for each prediction.

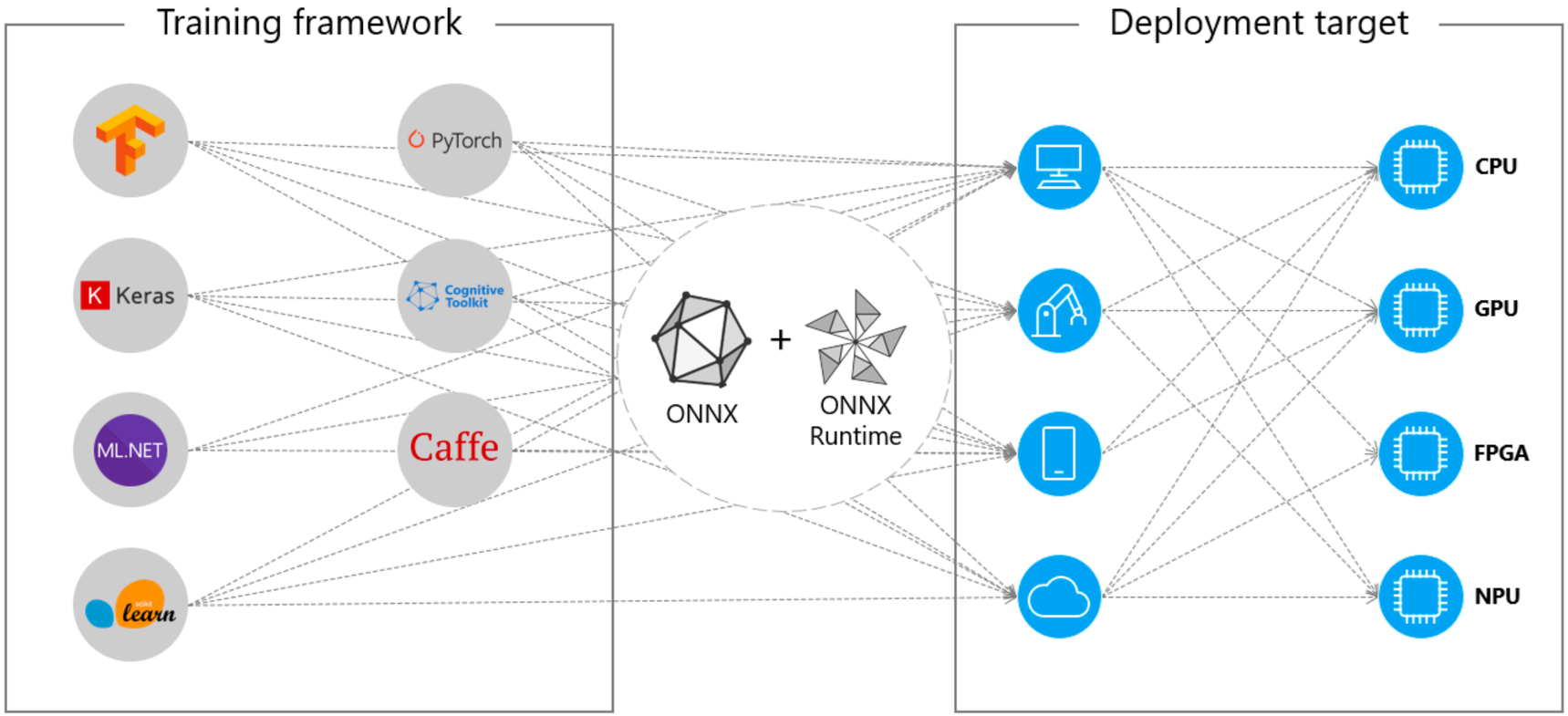
# Model Serving

## Model published as data

- The model is also treated and published independently,
- But the consuming application will ingest it as data at runtime.
- We have seen this used in streaming/real-time scenarios where the application can subscribe to events that are published whenever a new model version is released, and ingest them into memory while continuing to predict using the previous version.
- Software release patterns such as Canary Releases can also be applied in this scenario.

# Export ML models to production environment

## Open Neural Network Exchange



# Testing and Quality in Machine Learning

- Regardless of which pattern you decide to use, there is always an implicit contract between the model and its consumers.
- The model will usually expect input data in a certain shape, and if Data Scientists change that contract to require new input or add new features, you can cause integration issues and break the applications using it.
- So testing becomes important.


Continuous Delivery for Machine Learning

# Testing Machine Learning Systems

## Validating data

- Tests to **validate input data** against the expected schema, or to validate our **assumptions** about its valid values:
  - Values fall within expected ranges
  - Values are not null
- Unit tests to check **features** are calculated correctly:
  - Numeric features are scaled or normalized,
  - One-hot encoded vectors contain all zeroes and a single 1
  - Missing values are replaced appropriately

# Testing Machine Learning Systems

## Validating component integration

- Test the **integration** between different services:
  - Contract Tests to validate that the expected model interface is compatible with the consuming application.
- Test that the **exported model** still produces the same results:
  - Running the original and the productionized models against the same validation dataset, and comparing the results are the same.

# Testing Machine Learning Systems

## Validating the model quality

- ML **model performance** is non-deterministic.
- Collect and monitor **metrics** to evaluate a model's performance,
  - Error rates, accuracy
  - Precision, recall
  - AUC, ROC, confusion matrix
- **Threshold Tests** in our pipeline, to ensure that new models don't degrade against a known performance baseline.

# Testing Machine Learning Systems

## Validating model bias and fairness

- Check how the model performs against **baselines** for specific **data slices**:
  - Inherent bias in the training data where there are many more data points for a given value of a feature (e.g. race, gender, or region) compared to the actual distribution in the real world.
- A tool like **Facets** can help you visualize those slices and the distribution of values across the features in your datasets.

# Testing Machine Learning Systems

## Integration Test

- When models are **distributed or exported** to be used by a different application,
- The engineered features are **calculated differently** between training and serving time.
- Distribute a **holdout dataset** along with the model artifact, and allow the consuming application team to reassess the model's performance against the holdout dataset after it is integrated.
- This would be the equivalent of a broad **Integration Test** in traditional software development.

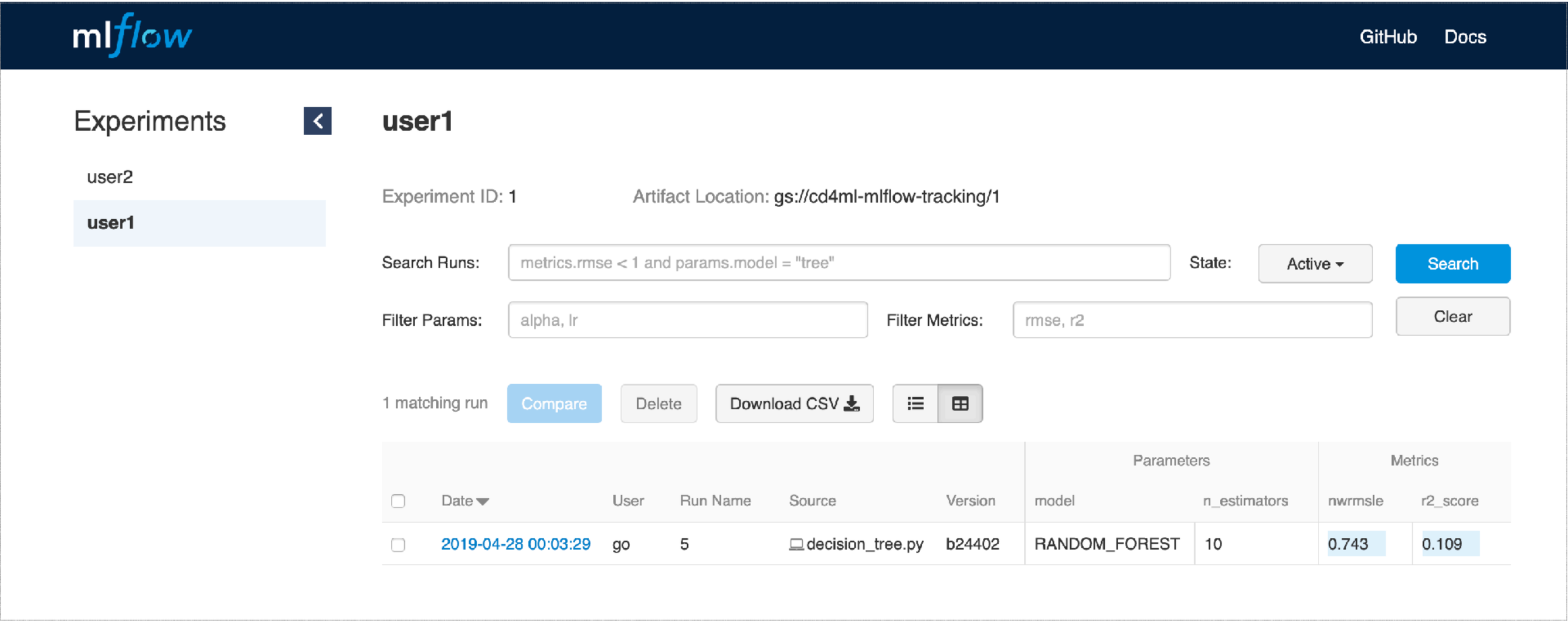
# Governance process for ML Systems

## Experiments Tracking

- To capture and display information that will allow humans to decide if and which model should be promoted to production.
- It is common that you will have multiple experiments being tried in parallel, and many of them might not ever make it to production.
- The code for many of these experiments will be thrown away, and only a few of them will be deemed worthy of making it to production.
- Different Git branches to track the different experiments in source control.
- Tools such as DVC can fetch and display metrics from experiments running in different branches or tags, making it easy to navigate between them.

# Governance process for ML Systems

## MLflow Tracking web UI



# Model Deployment

## Multiple models

- More than one model performing the same task.
  - Train a model to predict demand for each product.
- Deploying the models as a separate service might be better for consuming applications to get predictions with a single API call.
- You can later evolve how many models are needed behind that Published Interface.

# Model Deployment

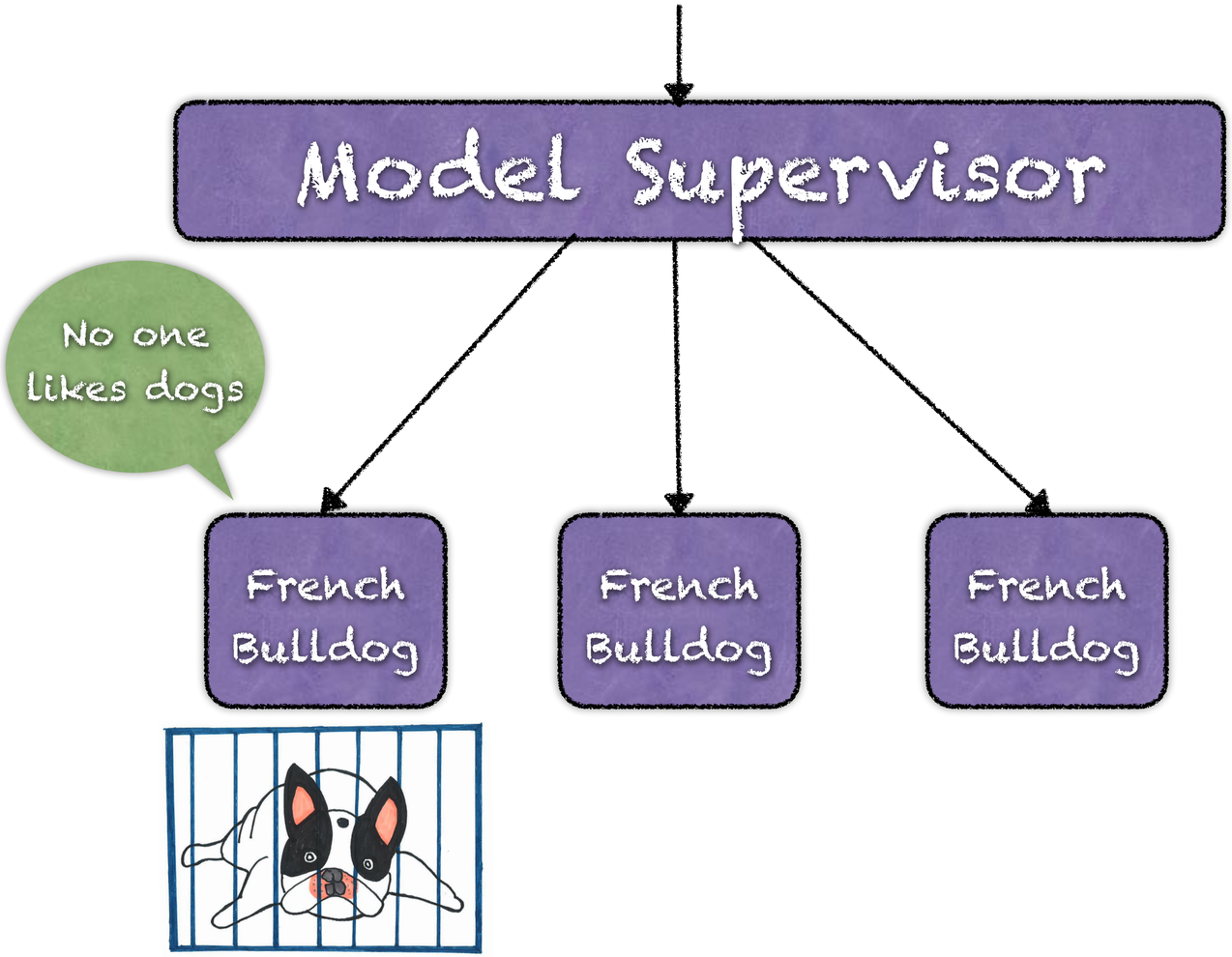
## Shadow models

- Deploy the new model side-by-side with the current one, as a shadow model
- Send the same production traffic to gather data on how the shadow model performs before promoting it into the production.

# Model Deployment

## Competing models

- Multiple versions of the model in production — like an A/B test
  - Infrastructure and routing rules required to ensure the traffic is being redirected to the right models.
  - To gather enough data to make statistically significant decisions, which can take some time.
- Evaluating multiple competing models is Multi-Armed Bandits,
  - To define a way to calculate and monitor the reward associated with using each model.


Model Supervisor
No one likes dogs
French Bulldog
French Bulldog
French Bulldog

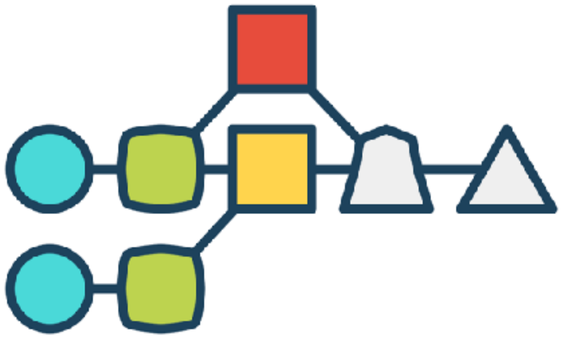
# Model Deployment

## Online learning models

- To use algorithms and techniques that can continuously improve its performance with the arrival of new data.
- Constantly learning in production.
- Extra complexities, as versioning the model as a static artifact won't yield the same results if it is not fed the same data.
- You will need to version not only the training data, but also the production data that will impact the model's performance.

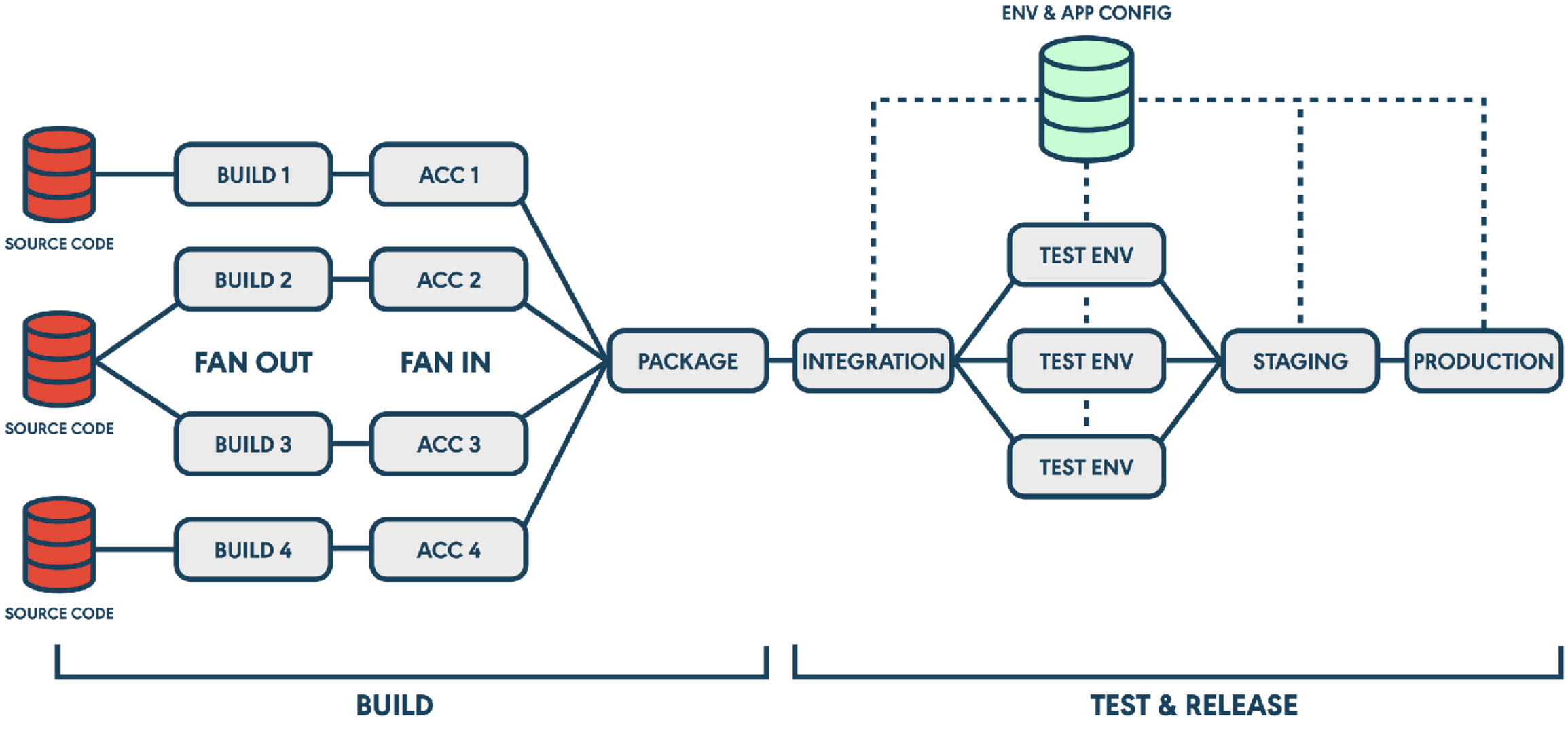
# Orchestration in ML Pipelines

- Provisioning of infrastructure and the execution of the ML Pipelines to train and capture metrics from multiple model experiments
- Building, testing, and deploying Data Pipelines
- Testing and validation to decide which models to promote
- Provisioning of infrastructure and deployment of models to production



# Continuous Integration and Delivery

## GoCD

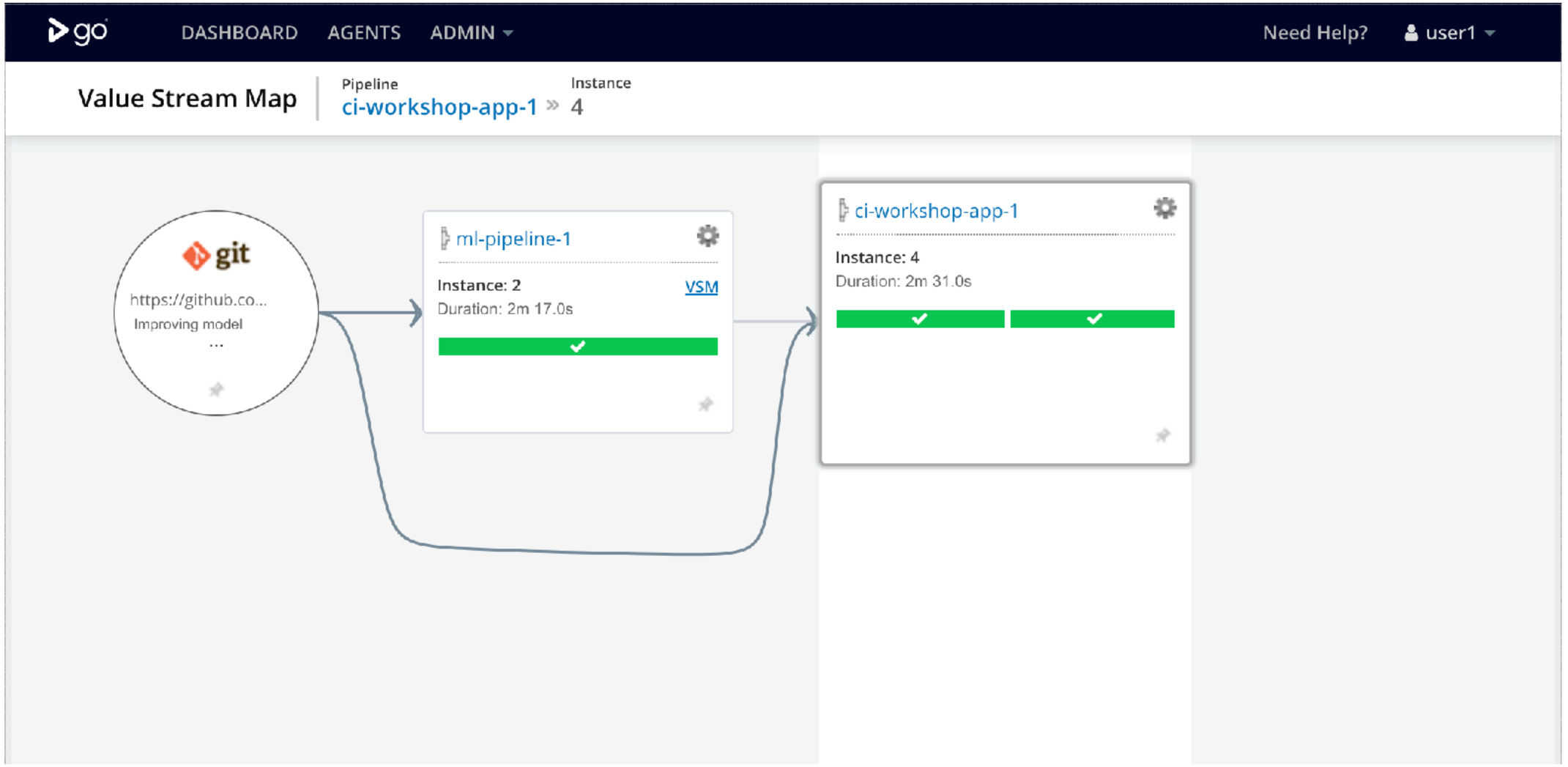


# A Continuous Delivery Scenario for ML

1. **Machine Learning Pipeline**:
   - To train and evaluate ML models
   - To execute threshold test to decide if the model can be promoted or not
   - *dvc push* to publish it as an artifact
2. **Application Deployment Pipeline**:
   - To build and test the application code
   - To fetch the promoted model from the upstream pipeline using *dvc pull*
   - To package a new combined artifact that contains the model and the application as a Docker image
   - To deploy them to a production cluster


Continuous Delivery for Machine Learning

# Combining Machine Learning Pipeline and Application Deployment Pipeline



# ML Model Monitoring

## How models perform in production and rollback mechanisms

- **Model inputs**:
  - What data is being fed to the models, identifying training-serving skew.
- **Model outputs**:
  - What predictions and recommendations are the models making from these inputs, to understand how the model is performing with real data.

# ML Model Monitoring

## How models perform in production and rollback mechanisms

- **Model interpretability outputs**:
  - Metrics such as model coefficients, ELI5, or LIME outputs that allow further investigation to understand how the models are making predictions to identify potential overfit or bias that was not found during training.

hi there, i am here looking for some help. my friend is a interic
graphics software on pc. any suggestion on which software to
sophisticated software(the more features it has,the better)

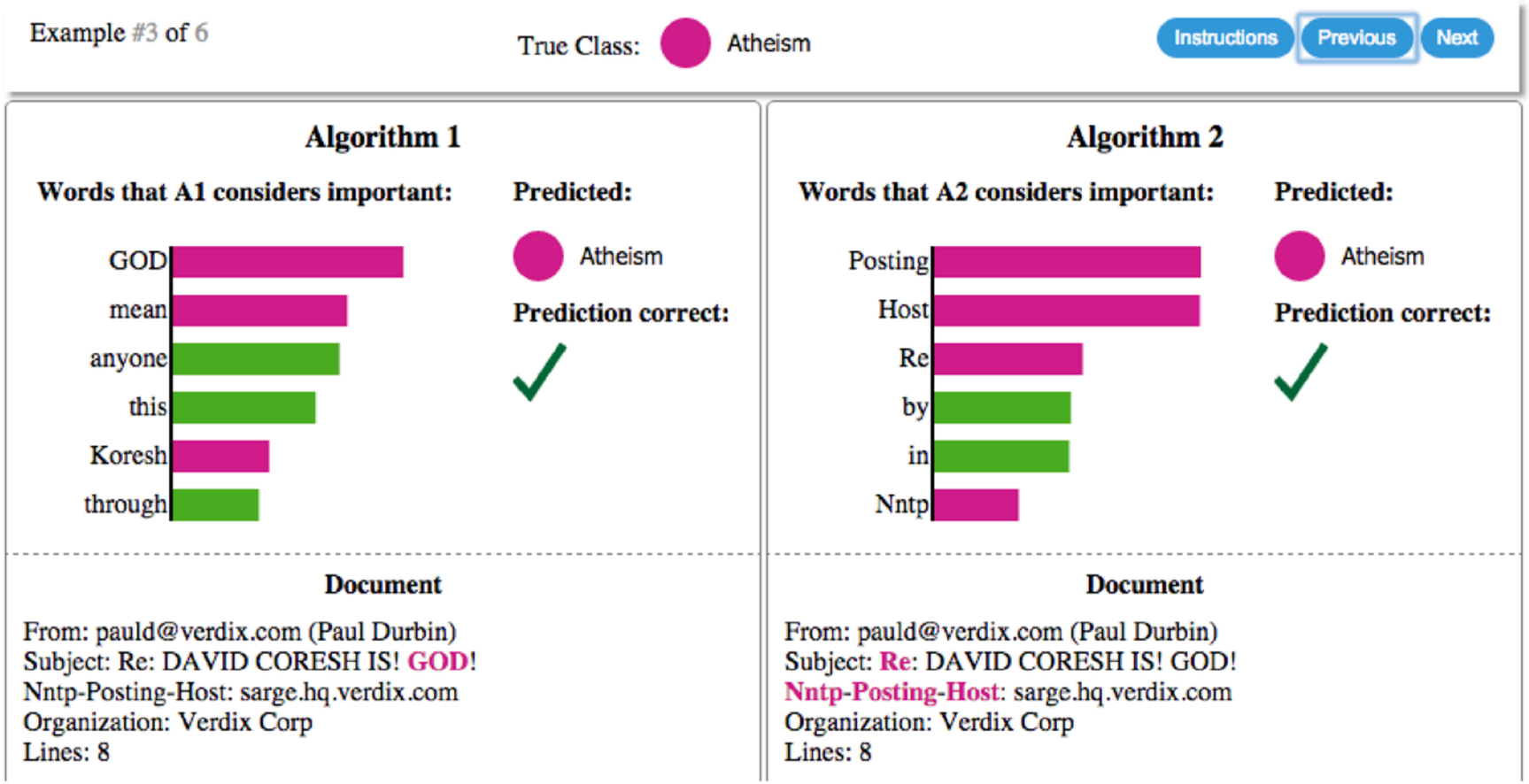
| y=0 (probability **0.000**) top features | | |
|---|---|---|
| Contribution[?] | Feature | Value |
| +0.301 | <BIAS> | 1.000 |
| +0.064 | color_intensity | 8.500 |
| +0.004 | malic_acid | 4.600 |
| -0.018 | alcalinity_of_ash | 25.000 |
| -0.044 | total_phenols | 1.980 |
| -0.055 | flavanoids | 0.960 |
| -0.100 | proline | 630.000 |
| -0.153 | hue | 0.670 |

| y=1 (probability **0.100**) top features | | |
|---|---|---|
| Contribution[?] | Feature | Value |
| +0.427 | <BIAS> | 1.000 |
| +0.033 | proline | 630.000 |
| +0.022 | od280/od315_of_diluted_wines | 1.920 |
| +0.009 | alcalinity_of_ash | 25.000 |
| +0.006 | total_phenols | 1.980 |
| -0.003 | proanthocyanins | 1.110 |
| -0.010 | alcohol | 13.400 |
| -0.028 | flavanoids | 0.960 |
| -0.060 | malic_acid | 4.600 |
| -0.137 | hue | 0.670 |
| -0.160 | color_intensity | 8.500 |

| y=2 (probability **0.900**) top features | | |
|---|---|---|
| Contribution[?] | Feature | Value |
| +0.289 | hue | 0.670 |
| +0.272 | <BIAS> | 1.000 |
| +0.095 | color_intensity | 8.500 |
| +0.083 | flavanoids | 0.960 |
| +0.067 | proline | 630.000 |
| +0.056 | malic_acid | 4.600 |
| +0.038 | total_phenols | 1.980 |
| +0.010 | alcohol | 13.400 |
| +0.009 | alcalinity_of_ash | 25.000 |
| +0.003 | proanthocyanins | 1.110 |
| -0.022 | od280/od315_of_diluted_wines | 1.920 |

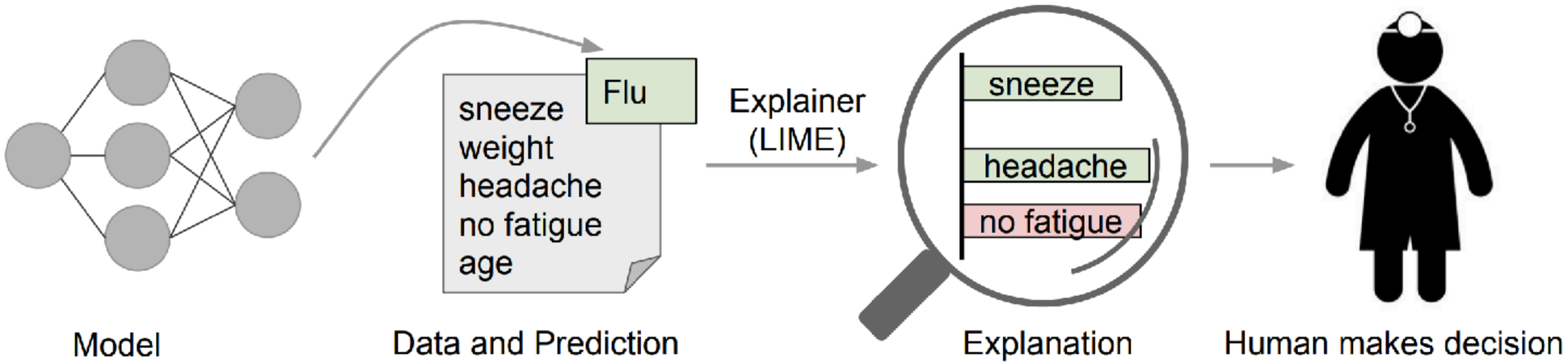
# “Why Should I Trust You?”

## Explaining the Predictions of Any Classifier



# Explaining individual predictions

**A model predicts that a patient has the flu, and LIME highlights the symptoms in the patient's history that led to the prediction**



# ML Model Monitoring

## How models perform in production and rollback mechanisms

- **Model outputs and decisions**:
  - What predictions our models are making given the production input data, and also which decisions are being made with those predictions.
  - Sometimes the application might choose to ignore the model and make a decision based on pre-defined rules (or to avoid future bias).

# ML Model Monitoring

## How models perform in production and rollback mechanisms

- **User action and rewards**:
  - Based on further user action, we can capture reward metrics to understand if the model is having the desired effect.
  - For example, if we display product recommendations, we can track when the user decides to purchase the recommended product as a reward.

# A pipeline for model monitoring
## ELK

- **Elasticsearch**: an open source *search* engine.
- **Logstash**: an open source data collector for unified *logging* layer.
- **Kibana**: an open source web UI that makes it easy to explore and *visualize* the data indexed by Elasticsearch.

# A pipeline for model monitoring

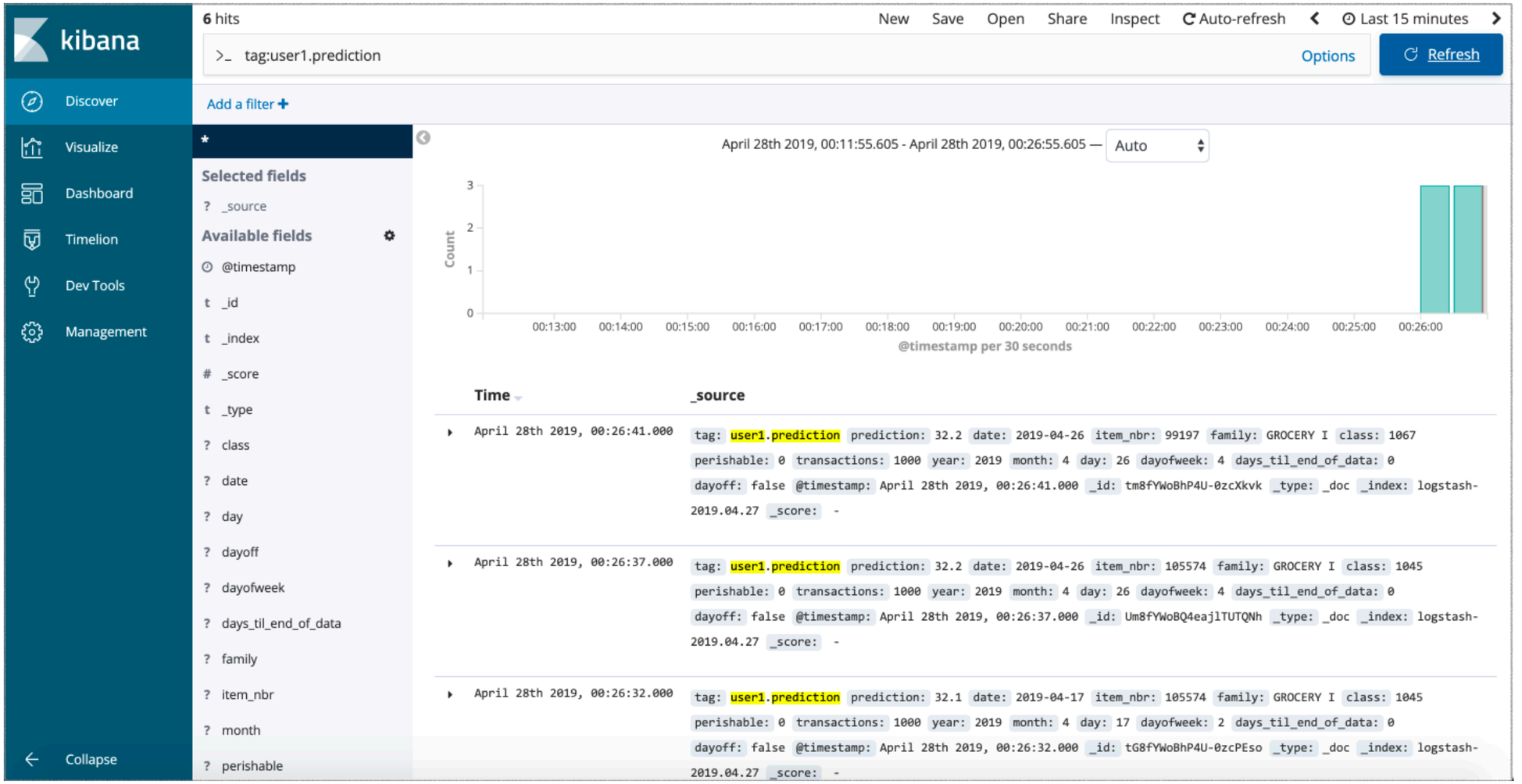
## ELK

# Logging

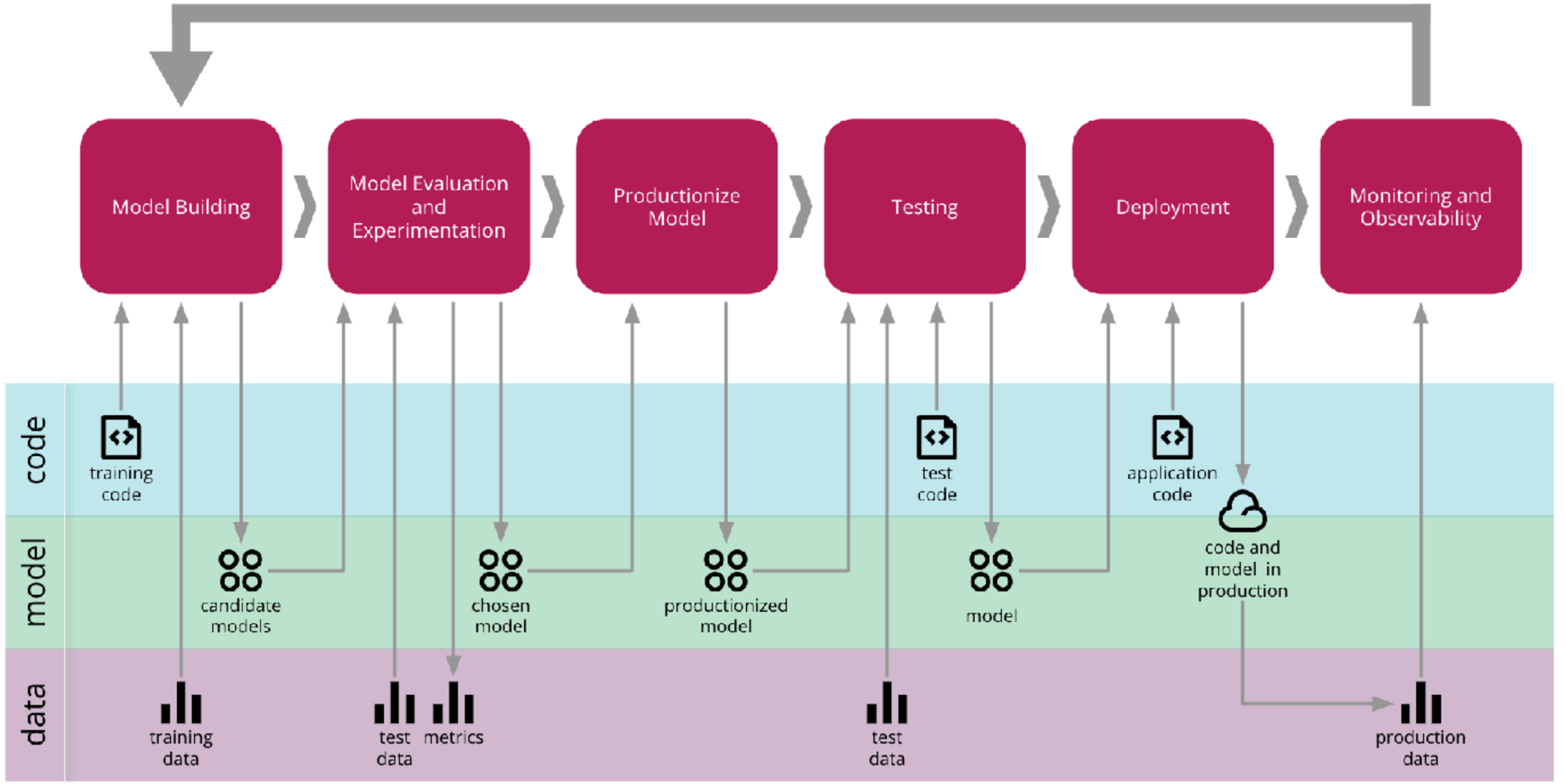
*predict_with_logging.py...*

```
df = pd.DataFrame(data=data, index=['row1'])
df = decision_tree.encode_categorical_columns(df)
pred = model.predict(df)
logger = sender.FluentSender(TENANT, host=FLUENTD_HOST, port=int(FLUENTD_PORT))
log_payload = {'prediction': pred[0], **data}
logger.emit('prediction', log_payload)
```

# A pipeline for model monitoring



# An End-to-End ML Building Process





Continuous Delivery for Machine Learning

# Machine Learning Systems

Next class: Foundations of Neural Networks and Learning