

SUMMARY & REFLECTION

(the last one)

Christian Kaestner

TODAY

(1)

**Looking back at the
semester**

(375 slides in 40 min)

(2)

**Discussion of future of
SE4AI**

(3)

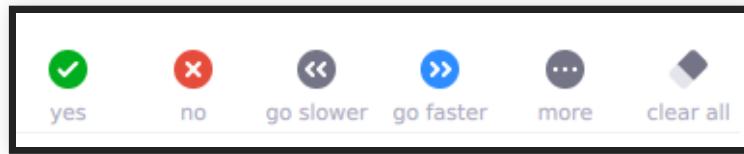
**Feedback for future
semesters**

INTRODUCTION AND MOTIVATION

Eunsuk Kang & Christian Kaestner

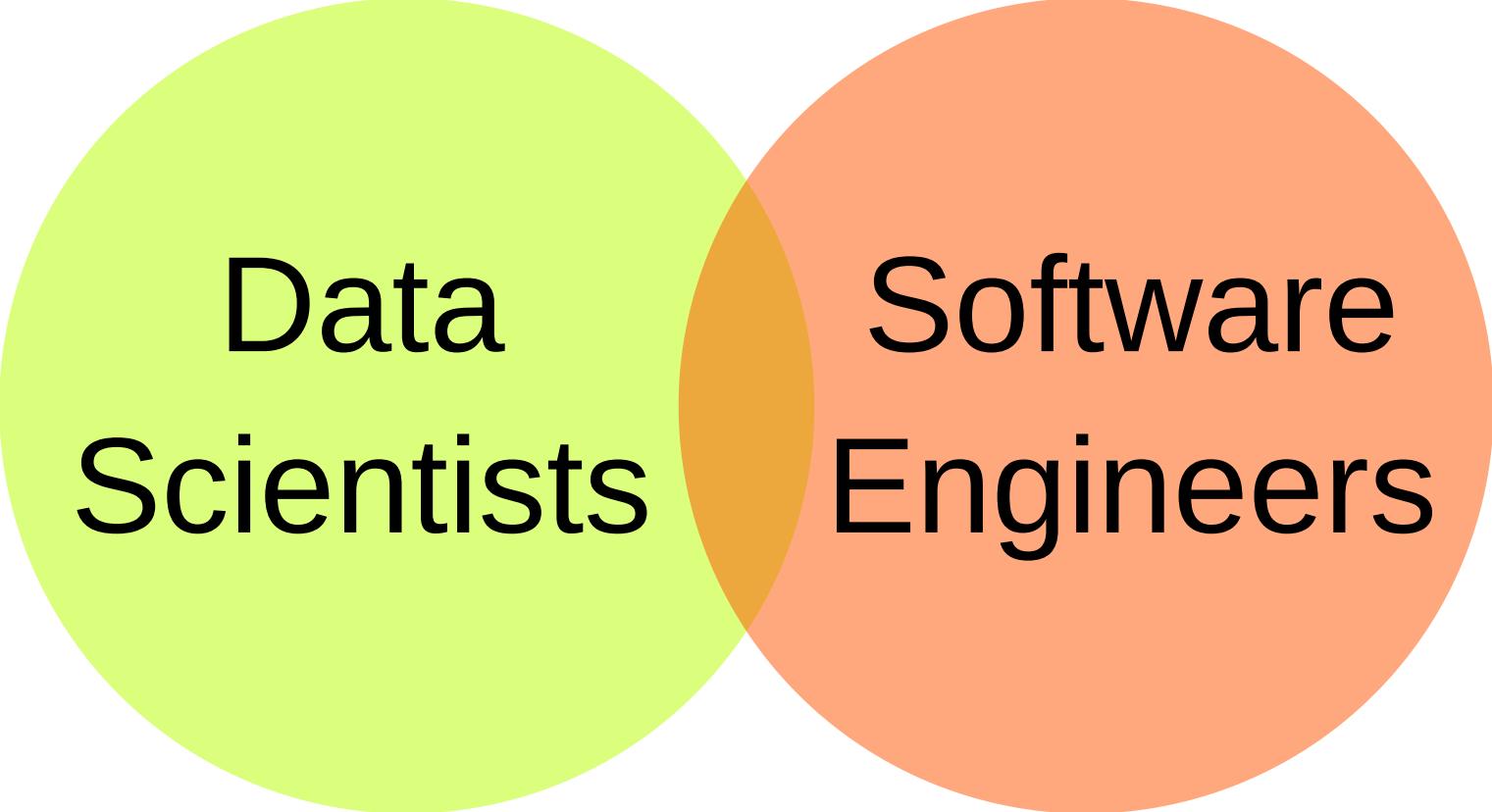
LECTURE LOGISTICS DURING A PANDEMIC

If you can hear me, open the participant panel in Zoom and check "yes"



LEARNING GOALS

- Understand how AI components are parts of larger systems
- Illustrate the challenges in engineering an AI-enabled system beyond accuracy
- Explain the role of specifications and their lack in machine learning and the relationship to deductive and inductive reasoning
- Summarize the respective goals and challenges of software engineers vs data scientists



A Venn diagram consisting of two overlapping circles. The left circle is light green and contains the text "Data Scientists". The right circle is light orange and contains the text "Software Engineers". The two circles overlap in the center.

Data
Scientists

Software
Engineers

SOFTWARE ENGINEER

DATA SCIENTIST

- Often fixed dataset for training and evaluation (e.g., PBS interviews)
- Focused on accuracy
- Prototyping, often Jupyter notebooks or similar
- Expert in modeling techniques and feature engineering
- Model size, updateability, implementation stability typically does not matter

- Builds a product
- Concerned about cost, performance, stability, release time
- Identify quality through customer satisfaction
- Must scale solution, handle large amounts of data
- Detect and handle mistakes, preferably automatically
- Maintain, evolve, and extend the product over long periods
- Consider requirements for security, safety, fairness

QUALITIES OF INTEREST ("ILITIES")

- Quality is about more than the absence of defects
- Quality in use (effectiveness, efficiency, satisfaction, freedom of risk, ...)
- Product quality (functional correctness and completeness, performance efficiency, compatibility, usability, dependability, scalability, security, maintainability, portability, ...)
- Process quality (manageability, evolvability, predictability, ...)
- "Quality is never an accident; it is always the result of high intention, sincere effort, intelligent direction and skillful execution; it represents the wise choice of many alternatives." (many attributions)

A screenshot of a transcription software interface. At the top, there's a header with the file name 'the-changelog-318', a link to 'Dashboard', and a 'Quality' setting at 'High'. To the right are buttons for 'Last saved a few seconds ago', three dots for more options, and a yellow 'Share' button. Below the header is a timeline bar with markers at 00:00, Offset, 00:00, and 01:31:27. Underneath the timeline are four buttons: 'Play', 'Back 5s', '1x Speed', and 'Volume'. The main content area is divided into two sections: 'NOTES' on the left containing the placeholder 'Write your notes here', and the transcribed speech on the right.

Speaker 5 ► 07:44

Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

Speaker 5 ► 08:38

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript?

SYLLABUS AND CLASS STRUCTURE

17-445/17-645, Summer 2020, 12 units

Tuesday/Wednesday 3-4:20, here on zoom

TEXTBOOK

Building Intelligent Systems: A Guide to
Machine Learning Engineering

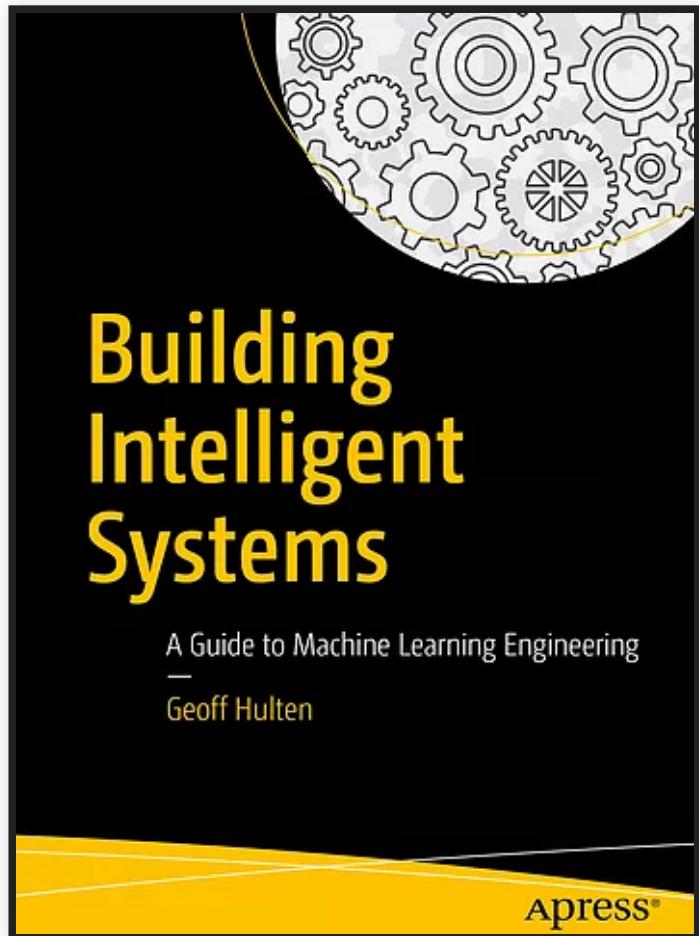
by Geoff Hulten

<https://www.buildingintelligentsystems.com/>

Most chapters assigned at some point in the
semester

Supplemented with research articles, blog
posts, videos, podcasts, ...

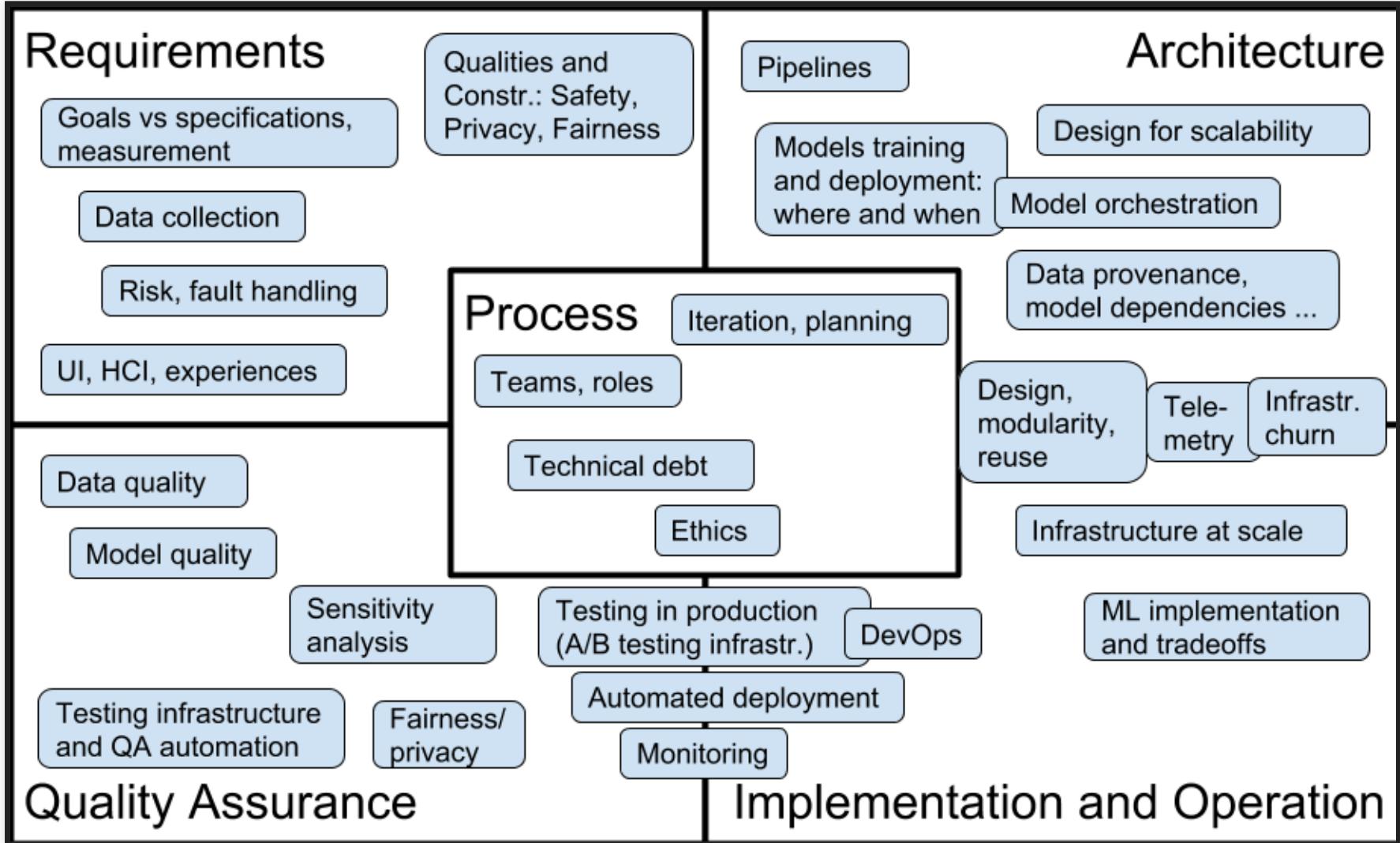
[Electronic version](#) in the library



GRADING PHILOSOPHY

- Specification grading, based in adult learning theory
- Giving you choices in what to work on or how to prioritize your work
- We are making every effort to be clear about expectations (specifications)
- Assignments broken down into expectations with point values, each graded pass/fail
- You should be able to tell what grade you will get for an assignment when you submit it, depending on what work you chose to do

[\[Example\]](#)



INTRODUCTIONS

Let's go around the "room" for introductions:

- Your (preferred name)
- In two sentences your software engineering background and goals
- In two sentences your data science background, if any, and goals
- One topic you are particularly interested in, if any?



CORRECTNESS AND SPECIFICATIONS

DEDUCTIVE VS. INDUCTIVE REASONING

WHO IS TO BLAME?

```
Algorithms.shortestDistance(g, "Tom", "Anne");
```

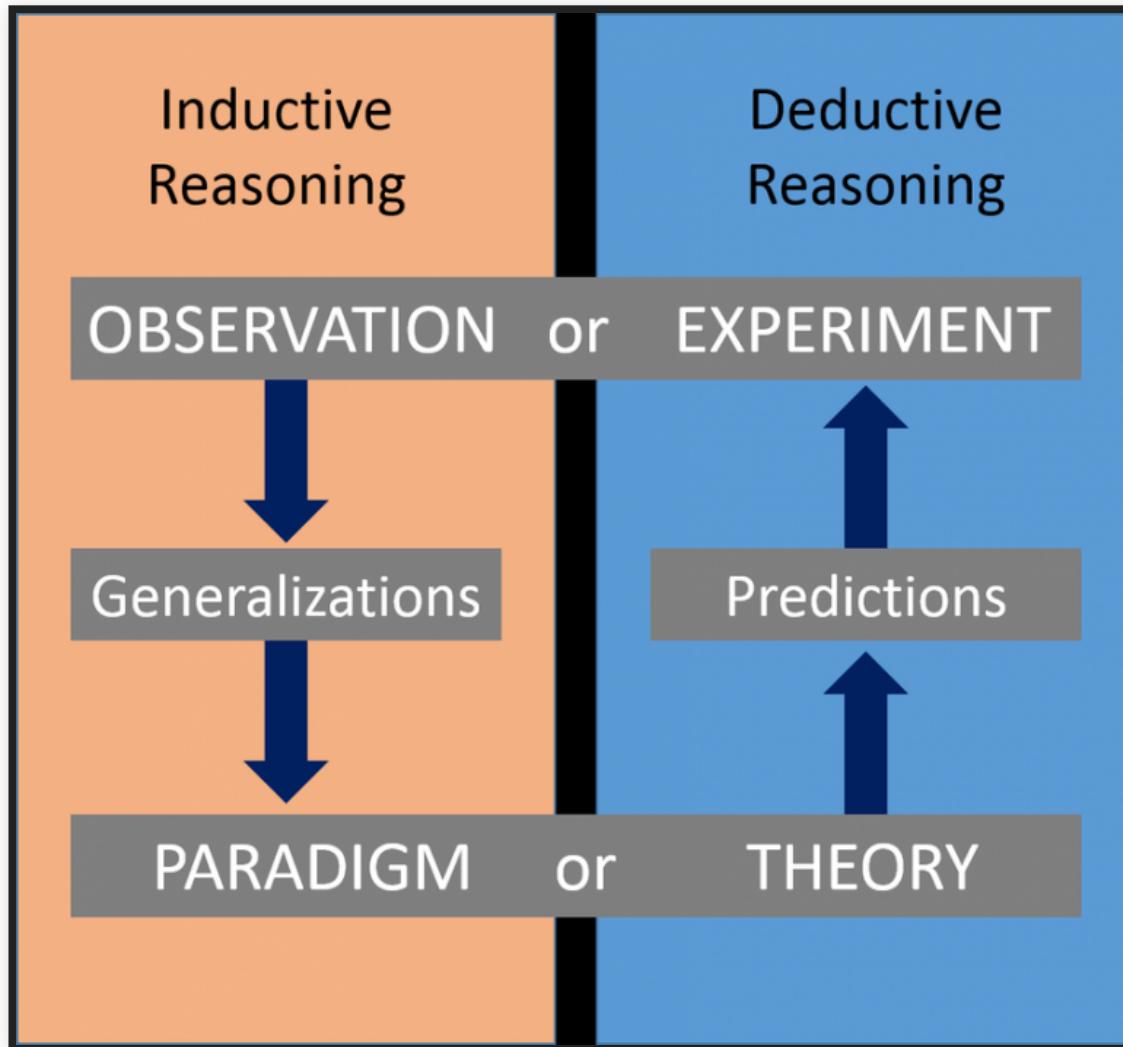
```
> ArrayOutOfBoundsException
```

```
Algorithms.shortestDistance(g, "Tom", "Anne");
```

```
> -1
```

SPECIFICATIONS IN MACHINE LEARNING?

```
/**  
 *  
 *  
 */  
String transcribe(File audioFile);
```



(Daniel Miessler, CC SA 2.0)

RESULTING SHIFT IN DESIGN THINKING?

From deductive reasoning to inductive reasoning...

From clear specifications to goals...

From guarantees to best effort...

What does this mean for software engineering?

For decomposing software systems?

For correctness of AI-enabled systems?

For safety?

For design, implementation, testing, deployment, operations?

FROM MODELS TO AI-ENABLED SYSTEMS

Eunsuk Kang

(With slides adopted from Christian Kaestner)

- Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018), Chapters 5 (Components of Intelligent Systems).

LEARNING GOALS

- Explain how machine learning fits into the larger picture of building and maintaining production systems
- Describe the typical components relating to AI in an AI-enabled system and typical design decisions to be made

MANAGING COMPLEXITY IN SOFTWARE

- **Abstraction:** Hide details & focus on high-level behaviors
- **Reuse:** Package into reusable libraries & APIs with well-defined *contracts*
- **Composition:** Build large components out of smaller ones

```
class Algorithms {  
    /**  
     * Finds the shortest distance between two vertices.  
     * This method is only supported for connected vertices.  
     */  
    int shortestDistance(Graph g, Vertice v1, v2) {...}  
}
```

(LACK OF) MODULARITY IN ML

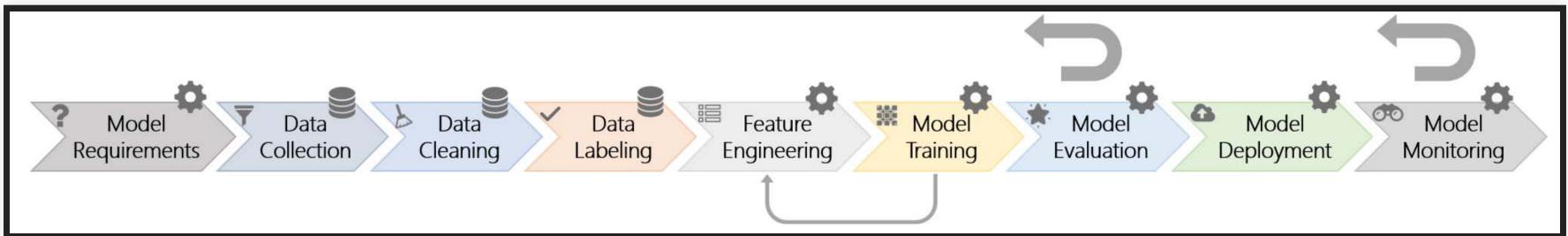
- Often no clear specification of "correct" behavior
 - Optimizing metrics instead of providing guarantees
- Model behavior strongly dependent on training & test sets
 - What happens if distribution changes?
 - Difficult to reuse!
- Poorly understood interactions between models
 - Ideally, develop models separately & compose together
 - In general, must train & tune together

These problems are not new, but are exacerbated by the increasing use of ML!

WHOLE SYSTEM PERSPECTIVE

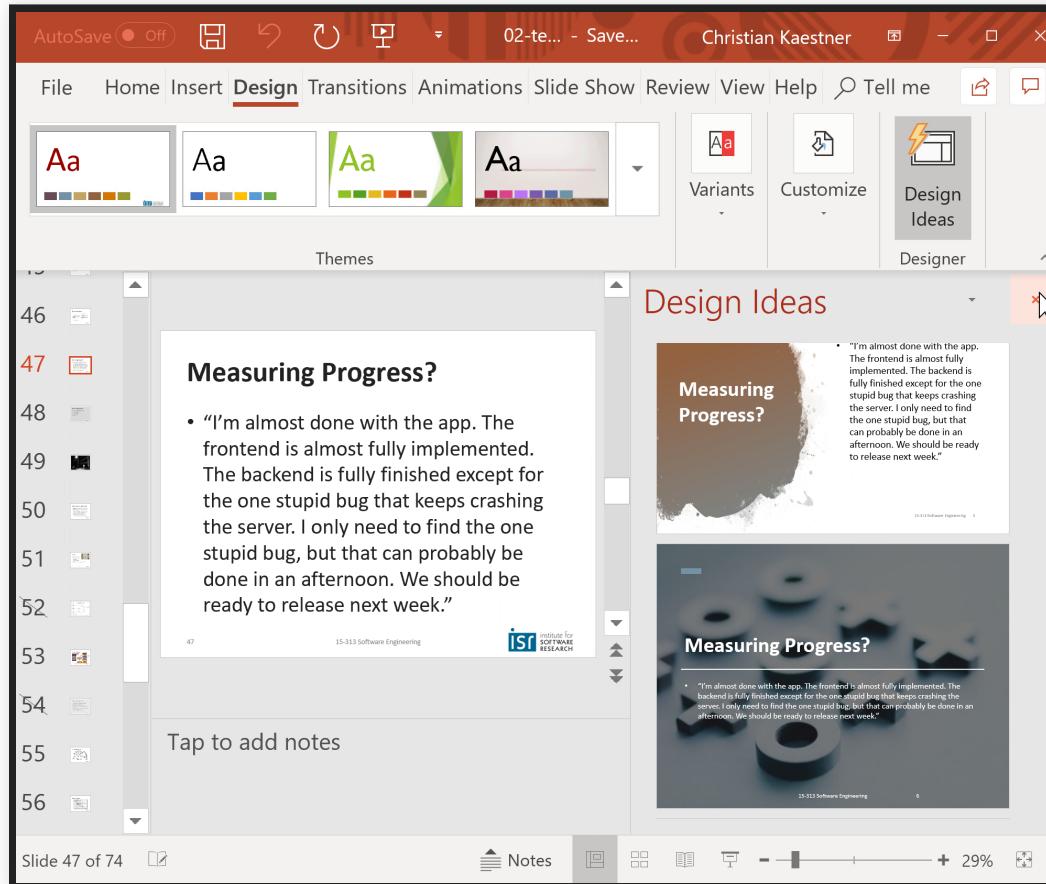
- A model is just one component of a larger system
- Also pipeline to build the model
- Also infrastructure to deploy, update, and serve the model
- Integrating the model with the rest of the system functionality
- User interaction design, dealing with mistakes
- Interaction with other stakeholders, detecting feedback loop
- Overall system goals vs model goals

let's look at some examples



- Graphic: Amershi et al. "Software engineering for machine learning: A case study." In Proc ICSE-SEIP, 2019.

MICROSOFT POWERPOINT

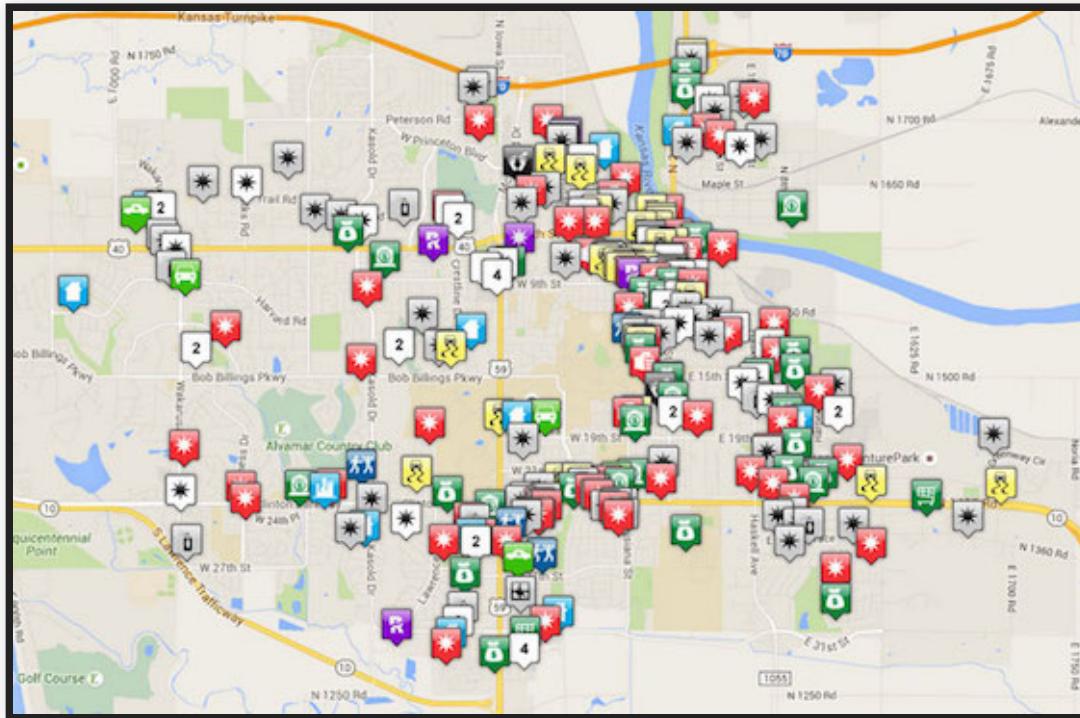


Read more: [How Azure Machine Learning enables PowerPoint Designer, Azure Blog, March 2020](#)

Speaker notes

Traditional application that uses machine learning in a few smaller places (more and more these days).

FEEDBACK LOOP



- Model: Use historical data to predict crime rates by neighborhoods
- Police increases the frequency of patrol in neighborhood X
- More arrests made in neighborhood X
- New crime data fed back to the model
- Repeat...

ELEMENTS OF AN INTELLIGENT SYSTEM

- Meaningful objective: Goals, requirements, business case
- Intelligent experience: User interactions -- presenting model predictions to user; eliciting & collecting feedback (telemetry)
- Intelligence implementation: Infrastructure -- learning and serving the model and collecting feedback
- Intelligence creation: Learning and evaluating models
- Orchestration: Operations -- maintaining and updating the system over time, debugging, countering abuse

PRESENTING INTELLIGENCE

- Automate: Take action on user's behalf
- Prompt: Ask the user if an action should be taken
- Organize: Display a set of items in an order
- Annotate: Add information to a display
- Hybrids of these

PRESENTING INTELLIGENCE: SAFE BROWSING



- Compare against more forceful, "automate" option
 - What are trade-offs between them?
 - If model makes a mistake, what kind of damage can it cause? Which one is easier to recover from?

FEEDBACK (TELEMETRY)

- To design good interactions we need to know how we are doing...
- How many predictions are ignored?
- How many actions are reversed?
- How often does the user ask for extra predictions?
- How much value do users get out of predictions?
- How much are we supporting the system's goals?
- How much cost are wrong predictions causing for users/the system's goals?
- Are mistakes focused on specific kinds of inputs?

Q. How would you design telemetry for safe browsing?

HOMEWORK 1: CASE STUDY

Engineering issues in detecting malicious apps or healthcare deployment

MODEL QUALITY

Christian Kaestner

Required reading:

- Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." Apress, 2018, Chapter 19 (Evaluating Intelligence).
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Semantically equivalent adversarial rules for debugging NLP models." In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 856-865. 2018.

LEARNING GOALS

- Select a suitable metric to evaluate prediction accuracy of a model and to compare multiple models
- Select a suitable baseline when evaluating model accuracy
- Explain how software testing differs from measuring prediction accuracy of a model
- Curate validation datasets for assessing model quality, covering subpopulations as needed
- Use invariants to check partial model properties with automated testing
- Develop automated infrastructure to evaluate and monitor model quality

MODEL QUALITY

FIRST PART: MEASURING PREDICTION ACCURACY

the data scientist's perspective

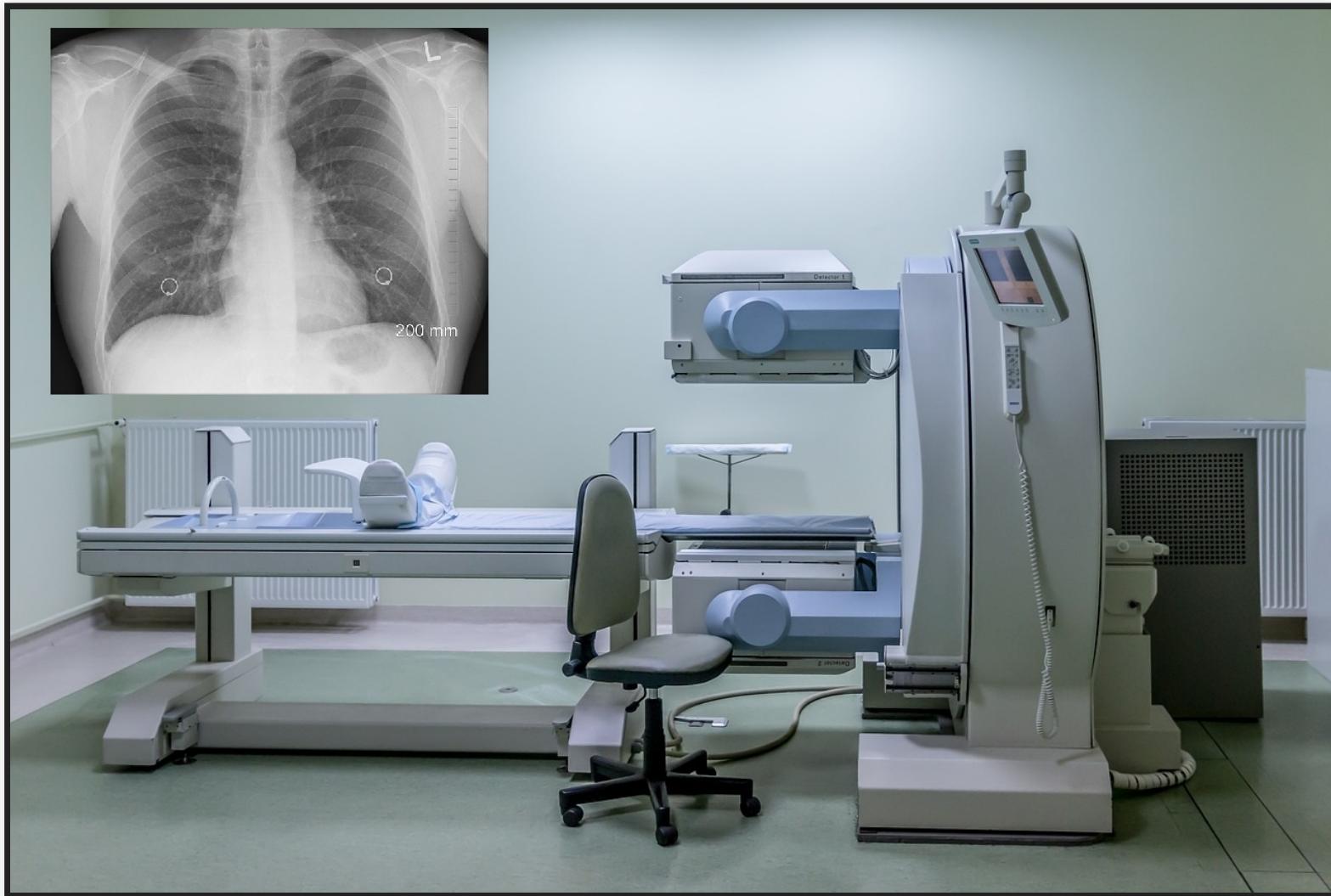
SECOND PART: LEARNING FROM SOFTWARE TESTING

how software engineering tools may apply to ML

testing in production (next week)

"Programs which were written in order to determine the answer in the first place. There would be no need to write such programs, if the correct answer were known"
(Weyuker, 1982).

CASE STUDY: CANCER DETECTION



THE SYSTEMS PERSPECTIVE

System is more than the model

Includes deployment, infrastructure, user interface, data infrastructure, payment services, and often much more

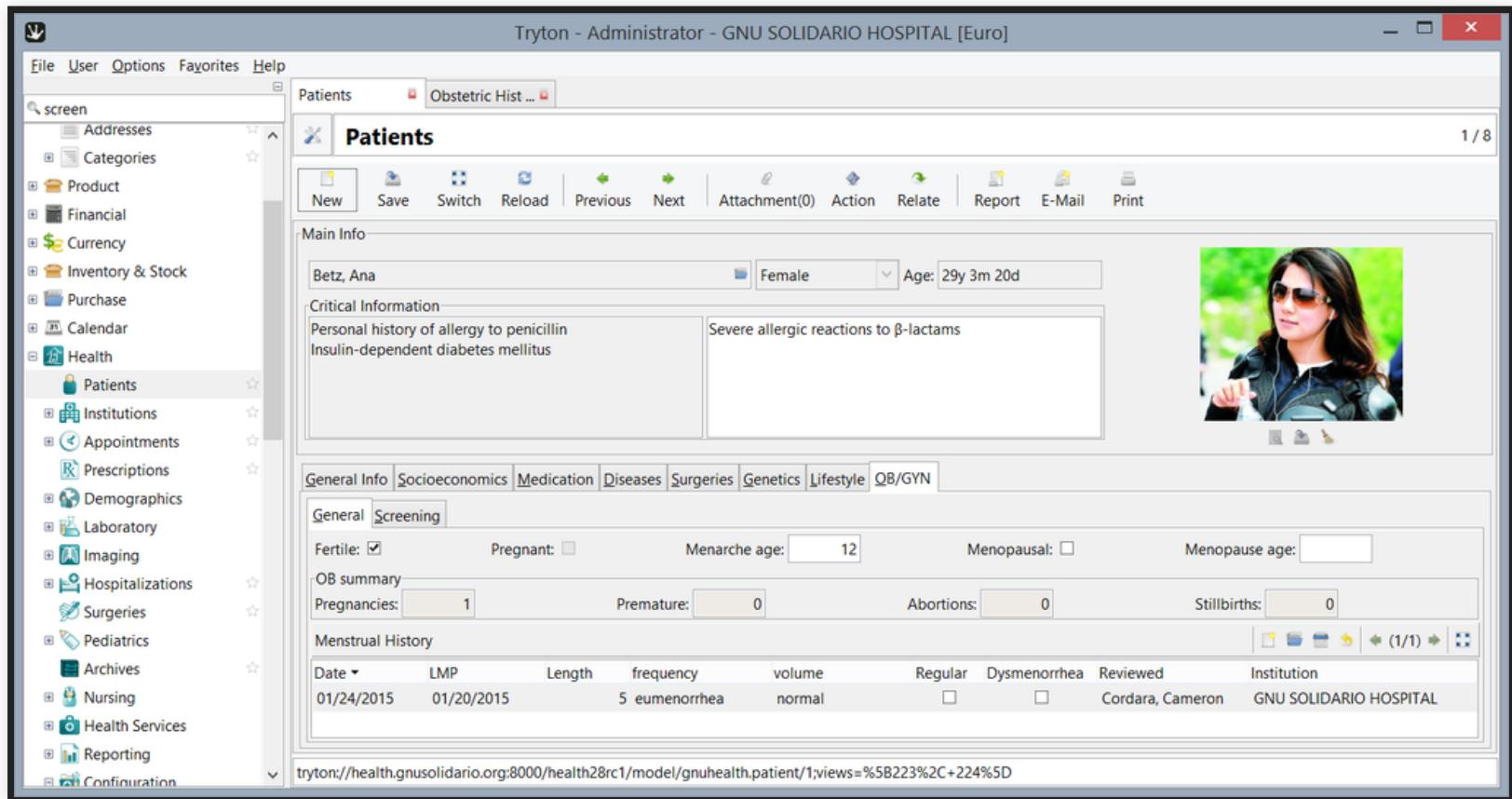
Systems have a goal:

- maximize sales
- save lives
- entertainment
- connect people

Models can help or may be essential in those goals, but are only one part

Today: Narrow focus on prediction accuracy of the model

CANCER PREDICTION WITHIN A HEALTHCARE APPLICATION



(CC BY-SA 4.0, Martin Sauter)

CONFUSION/ERROR MATRIX

	Actually A	Actually B	Actually C
AI predicts A	10	6	2
AI predicts B	3	24	10
AI predicts C	5	22	82

Accuracy = correct predictions (diagonal) out of all predictions

$$\text{Example's accuracy} = \frac{10+24+82}{10+6+2+3+24+10+5+22+82} = .707$$

IS 99% ACCURACY GOOD?

-> depends on problem; can be excellent, good, mediocre, terrible

10% accuracy can be good on some tasks (information retrieval)

Always compare to a base rate!

$$\text{Reduction in error} = \frac{(1 - \text{accuracy}_{\text{baseline}}) - (1 - \text{accuracy}_f)}{1 - \text{accuracy}_{\text{baseline}}}$$

- from 99.9% to 99.99% accuracy = 90% reduction in error
- from 50% to 75% accuracy = 50% reduction in error

TYPES OF MISTAKES

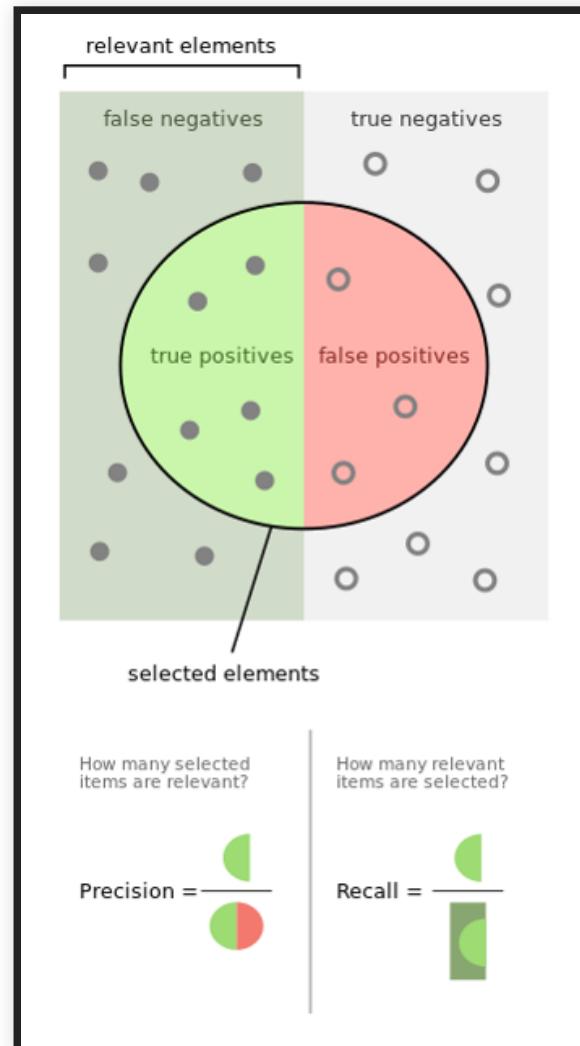
Two-class problem of predicting event A:

	Actually A	Actually not A
AI predicts A	True Positive (TP)	False Positive (FP)
AI predicts not A	False Negative (FN)	True Negative (TN)

True positives and true negatives: correct prediction

False negatives: wrong prediction, miss, Type II error

False positives: wrong prediction, false alarm, Type I error



(CC BY-SA 4.0 by [Walber](#))

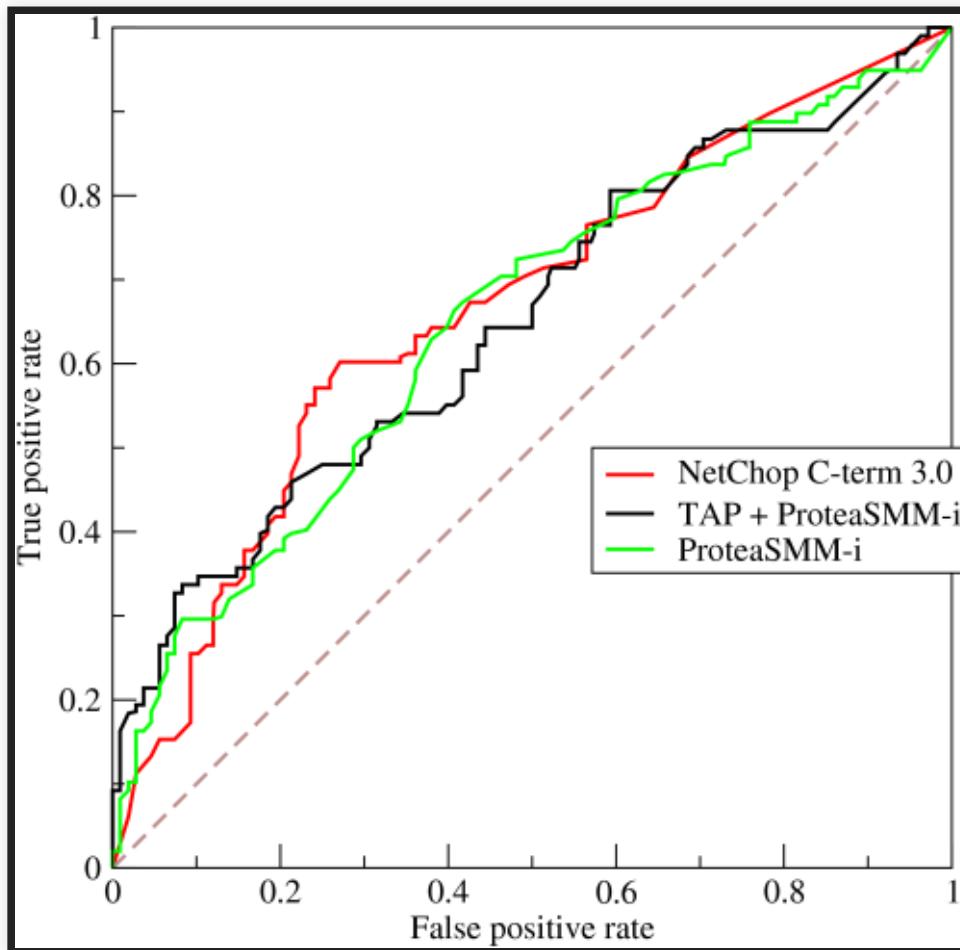
FALSE POSITIVES AND FALSE NEGATIVES EQUALLY BAD?

Consider:

- Recognizing cancer
- Suggesting products to buy on e-commerce site
- Identifying human trafficking at the border
- Predicting high demand for ride sharing services
- Predicting recidivism chance
- Approving loan applications

No answer vs wrong answer?

RECEIVER OPERATING CHARACTERISTIC (ROC) CURVES



COMPARING PREDICTED AND EXPECTED OUTCOMES

Mean Absolute Percentage Error

MAPE =

$$\frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

(A_t actual outcome, F_t predicted outcome, for row t)

Compute relative prediction error per row, average over all rows

Rooms	Crime Rate	...	Predicted Price	Actual Price
3	.01	...	230k	250k
4	.01	...	530k	498k
2	.03	...	210k	211k
2	.02	...	219k	210k

MAPE =

$$\begin{aligned} & \frac{1}{4}(20/250 + 32/498 + 1/211 + 9/210) \\ &= \frac{1}{4}(0.08 + 0.064 + 0.005 + 0.043) = \\ & 0.048 \end{aligned}$$

EVALUATING RANKINGS

Ordered list of results, true results should be ranked high

Common in information retrieval (e.g., search engines) and recommendations

Mean Average Precision

MAP@K = precision in first K results

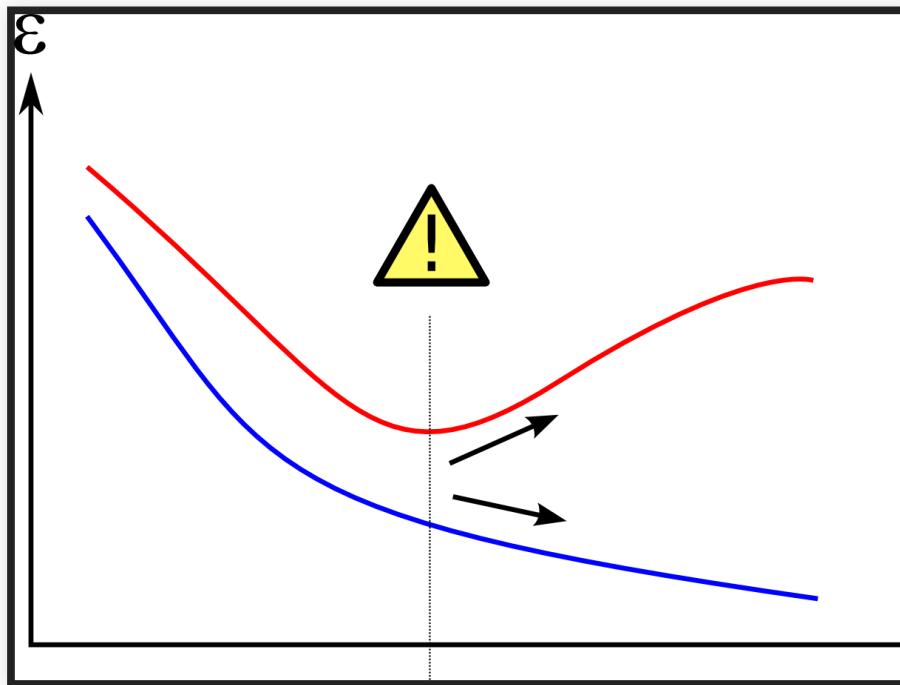
Averaged over many queries

Rank	Product	Correct?
1	Juggling clubs	true
2	Bowling pins	false
3	Juggling balls	false
4	Board games	true
5	Wine	false
6	Audiobook	true
MAP@1 = 1, MAP@2 = 0.5, MAP@3 = 0.33,		
...		

Remember to compare against baselines! Baseline for shopping recommendations?

DETECTING OVERFITTING

Change hyperparameter to detect training accuracy (blue)/validation accuracy (red) at different degrees of freedom



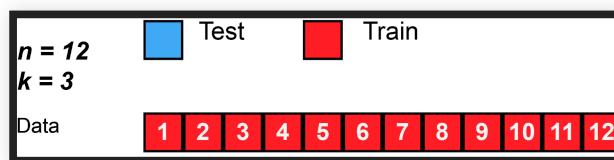
(CC SA 3.0 by [Dake](#))

demo time

CROSSVALIDATION

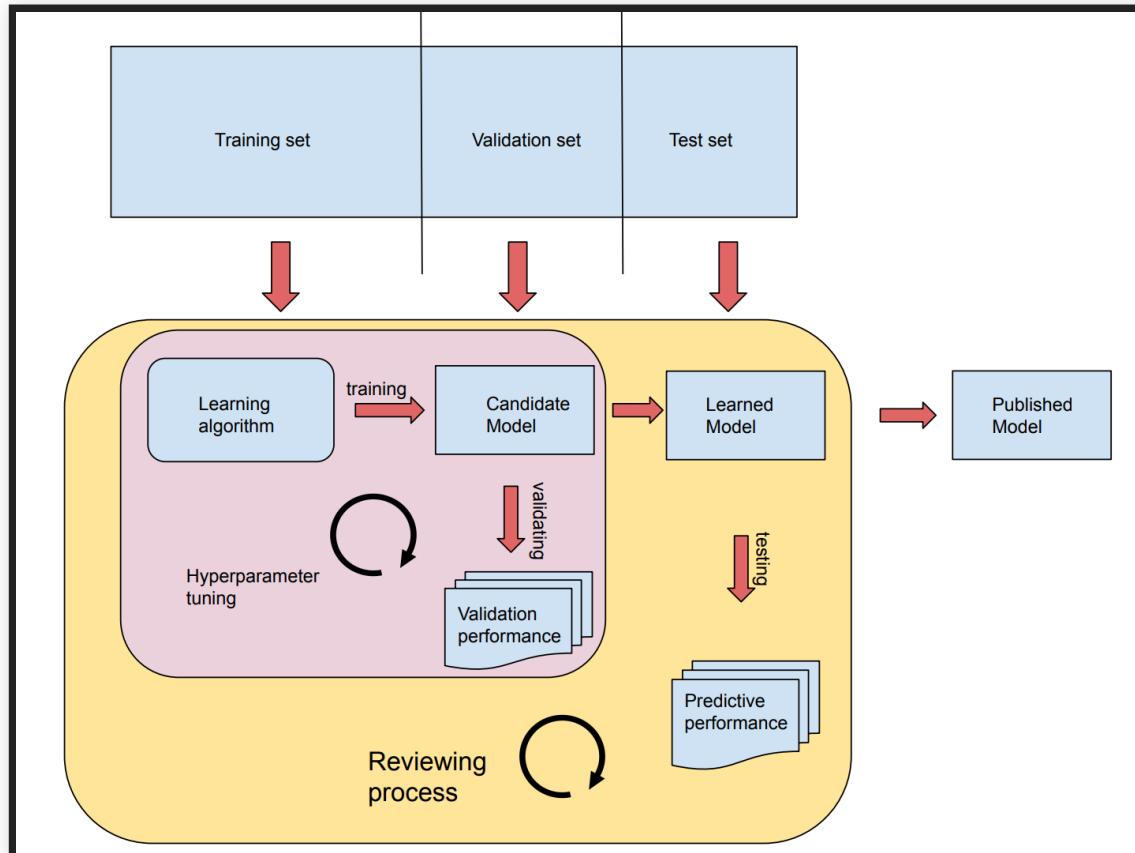
- Motivation
 - Evaluate accuracy on different training and validation splits
 - Evaluate with small amounts of validation data
- Method: Repeated partitioning of data into train and validation data, train and evaluate model on each partition, average results
- Many split strategies, including
 - leave-one-out: evaluate on each datapoint using all other data for training
 - k -fold: k equal-sized partitions, evaluate on each training on others
 - repeated random sub-sampling (Monte Carlo)

demo time



(Graphic CC MBanuelos22 BY-SA 4.0)

ACADEMIC ESCALATION: OVERFITTING ON BENCHMARKS



(Figure by Andrea Passerini)

Speaker notes

If many researchers publish best results on the same benchmark, collectively they perform "hyperparameter optimization" on the test set

ANALOGY TO SOFTWARE TESTING

(this gets messy)

MODEL TESTING?

Rooms	Crime Rate	...	Actual Price
3	.01	...	250k
4	.01	...	498k
2	.03	...	211k
2	.02	...	210k

```
assertEquals(250000,  
            model.predict([3, .01, ...])  
assertEquals(498000,  
            model.predict([4, .01, ...])  
assertEquals(211000,  
            model.predict([2, .03, ...])  
assertEquals(210000,  
            model.predict([2, .02, ...]))
```

Fail the entire test suite for one wrong prediction?

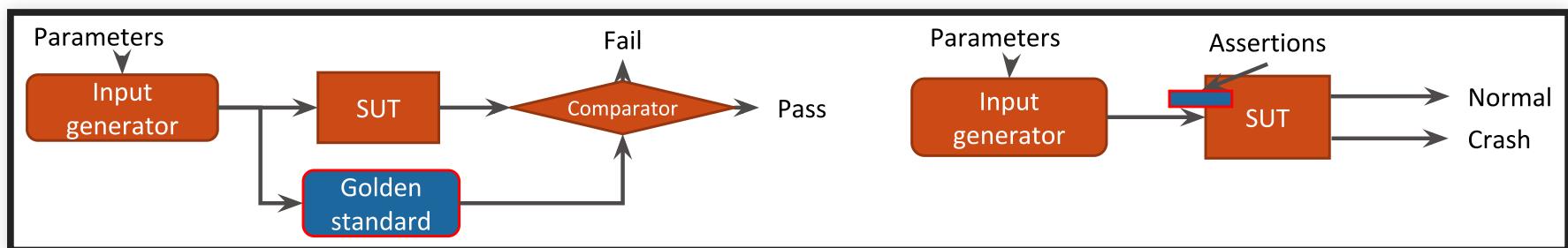


THE ORACLE PROBLEM

How do we know the expected output of a test?

```
assertEquals(??, factorPrime(15485863));
```

- Manually construct input-output pairs (does not scale, cannot automate)
- Comparison against gold standard (e.g., alternative implementation, executable specification)
- Checking of global properties only -- crashes, buffer overflows, code injections
- Manually written assertions -- partial specifications checked at runtime



DIFFERENT EXPECTATIONS FOR PREDICTION ACCURACY

- Not expecting that all predictions will be correct (80% accuracy may be very good)
- Data may be mislabeled in training or validation set
- There may not even be enough context (features) to distinguish all training outcomes
- Lack of specifications
- A wrong prediction is not necessarily a bug

ANALOGY OF PERFORMANCE TESTING?

- Performance tests are not precise (measurement noise)
 - Averaging over repeated executions *of the same test*
 - Commonly using diverse benchmarks, i.e., *multiple inputs*
 - Need to control environment (hardware)
- No precise specification
 - Regression tests
 - Benchmarking as open-ended comparison
 - Tracking results over time

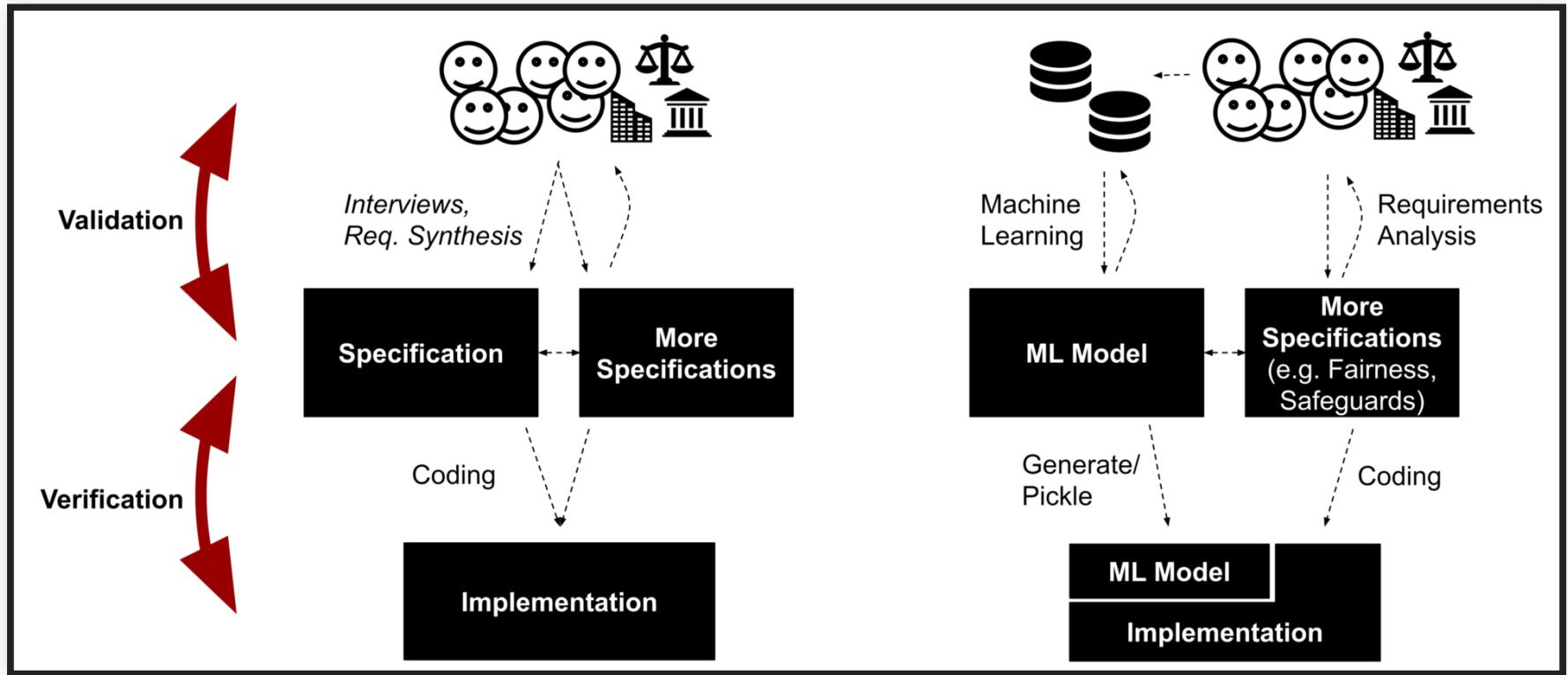
```
@Test(timeout=100)
public void testCompute() {
    expensiveComputation(...);
}
```

MACHINE LEARNING IS REQUIREMENTS ENGINEERING

(my pet theory)

see also <https://medium.com/@ckaestne/machine-learning-is-requirements-engineering-8957aee55ef4>

VALIDATION VS VERIFICATION



MACHINE LEARNING MODELS FIT, OR NOT

- A model is learned from given data in given procedure
 - The learning process is typically not a correctness concern
 - The model itself is generated, typically no implementation issues
- Is the data representative? Sufficient? High quality?
- Does the model "learn" meaningful concepts?
- **Is the model useful for a problem? Does it *fit*?**
- Do model predictions *usually* fit the users' expectations?
- Is the model *consistent* with other requirements? (e.g., fairness, robustness)

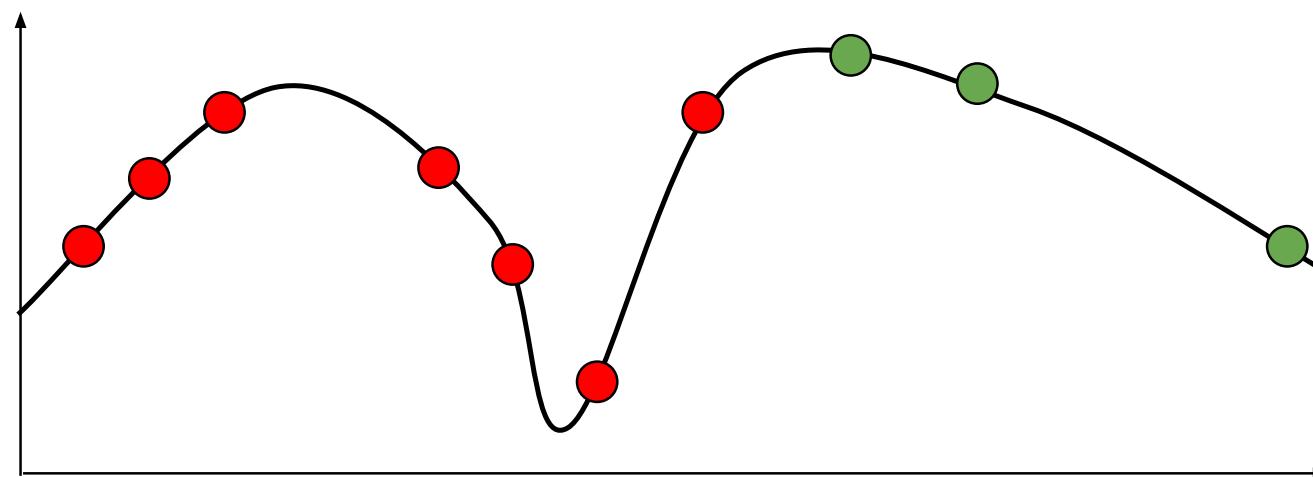
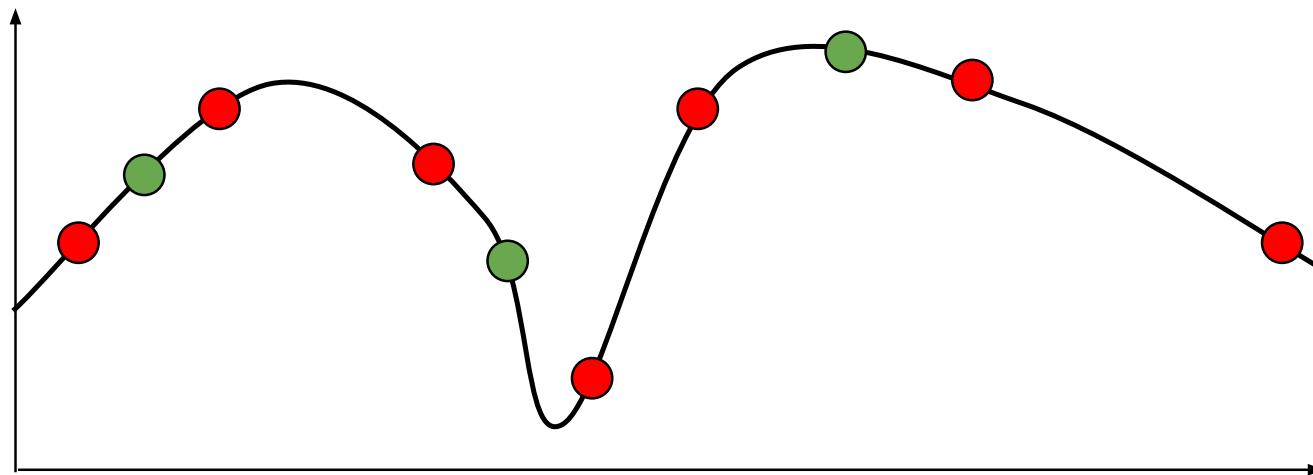
CURATING VALIDATION DATA

(Learning from Software Testing?)

VALIDATION DATA REPRESENTATIVE?

- Validation data should reflect usage data
- Be aware of data drift (face recognition during pandemic, new patterns in credit card fraud detection)
- "*Out of distribution*" predictions often low quality (it may even be worth to detect out of distribution data in production, more later)

INDEPENDENCE OF DATA: TEMPORAL



NOT ALL INPUTS ARE EQUAL



"Call mom" "What's the weather tomorrow?" "Add asafetida to my shopping list"

NOT ALL INPUTS ARE EQUAL

There Is a Racial Divide in Speech-Recognition Systems, Researchers Say: Technology from Amazon, Apple, Google, IBM and Microsoft misidentified 35 percent of words from people who were black. White people fared much better. --

NYTimes March 2020

IDENTIFY IMPORTANT INPUTS

Curate Validation Data for Specific Problems and Subpopulations:

- *Regression testing*: Validation dataset for important inputs ("call mom") -- expect very high accuracy -- closest equivalent to **unit tests**
- *Uniformness/fairness testing*: Separate validation dataset for different subpopulations (e.g., accents) -- expect comparable accuracy
- *Setting goals*: Validation datasets for challenging cases or stretch goals -- accept lower accuracy

Derive from requirements, experts, user feedback, expected problems etc. Think *blackbox testing*.

BLACK-BOX TESTING TECHNIQUES AS INSPIRATION?

- Boundary value analysis
- Partition testing & equivalence classes
- Combinatorial testing
- Decision tables

Use to identify subpopulations (validation datasets), not individual tests.



EXAMPLES OF INVARIANTS

- Credit rating should not depend on gender:
 - $\forall x. f(x[\text{gender} \leftarrow \text{male}]) = f(x[\text{gender} \leftarrow \text{female}])$
- Synonyms should not change the sentiment of text:
 - $\forall x. f(x) = f(\text{replace}(x, \text{"is not"}, \text{"isn't"}))$
- Negation should swap meaning:
 - $\forall x \in \text{"X is Y"}. f(x) = 1 - f(\text{replace}(x, \text{" is "}, \text{" is not "}))$
- Robustness around training data:
 - $\forall x \in \text{training data}. \forall y \in \text{mutate}(x, \delta). f(x) = f(y)$
- Low credit scores should never get a loan (sufficient conditions for classification, "anchors"):
 - $\forall x. x.\text{score} < 649 \Rightarrow \neg f(x)$

Identifying invariants requires domain knowledge of the problem!

METAMORPHIC TESTING

Formal description of relationships among inputs and outputs (*Metamorphic Relations*)

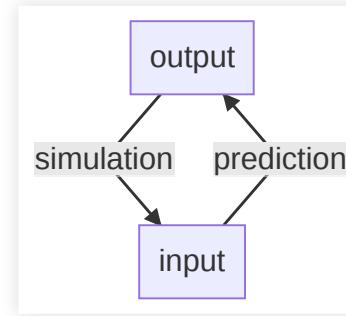
In general, for a model f and inputs x define two functions to transform inputs and outputs g_I and g_O such that:

$$\forall x. f(g_I(x)) = g_O(f(x))$$

e.g. $g_I(x) = \text{replace}(x, " \text{is} ", " \text{is not} ")$ and $g_O(x) = \neg x$

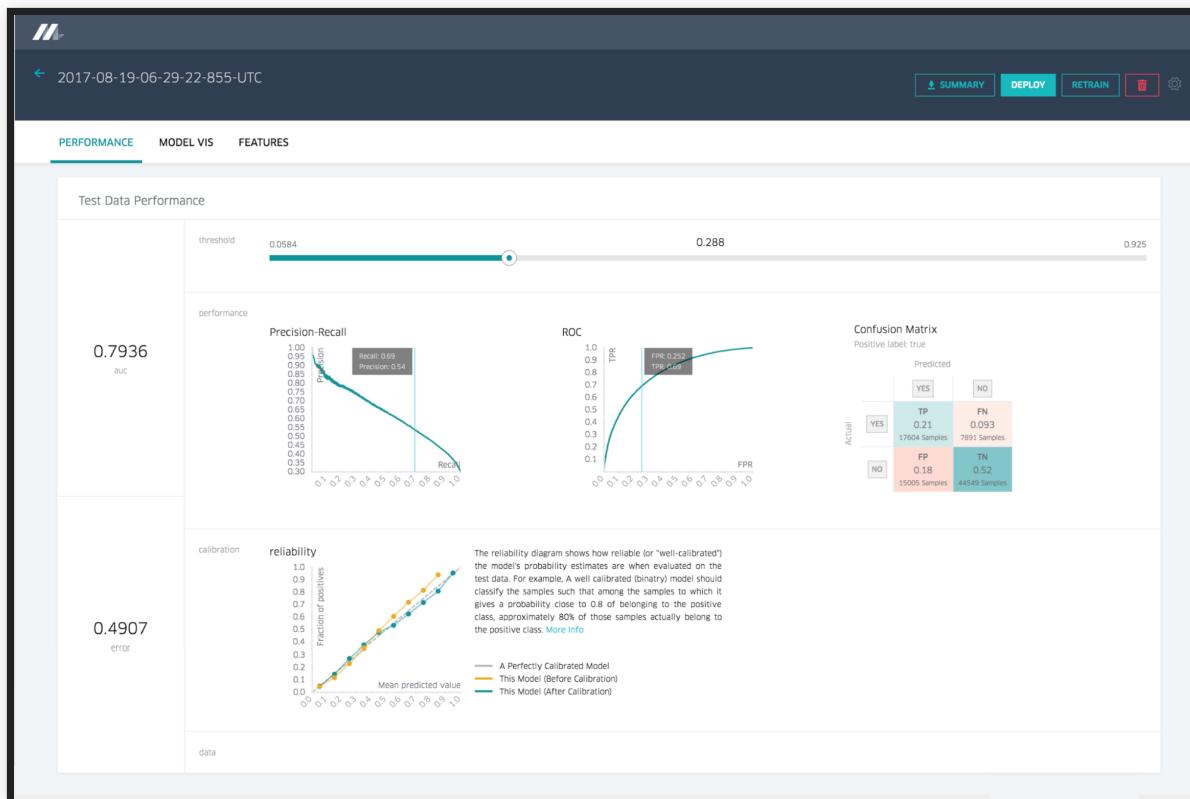
ONE MORE THING: SIMULATION-BASED TESTING

- Derive input-output pairs from simulation, esp. in vision systems
- Example: Vision for self-driving cars:
 - Render scene -> add noise -> recognize -> compare recognized result with simulator state
- Quality depends on quality of the simulator and how well it can produce inputs from outputs:
 - examples: render picture/video, synthesize speech, ...
 - Less suitable where input-output relationship unknown, e.g., cancer detection, housing price prediction, shopping recommendations

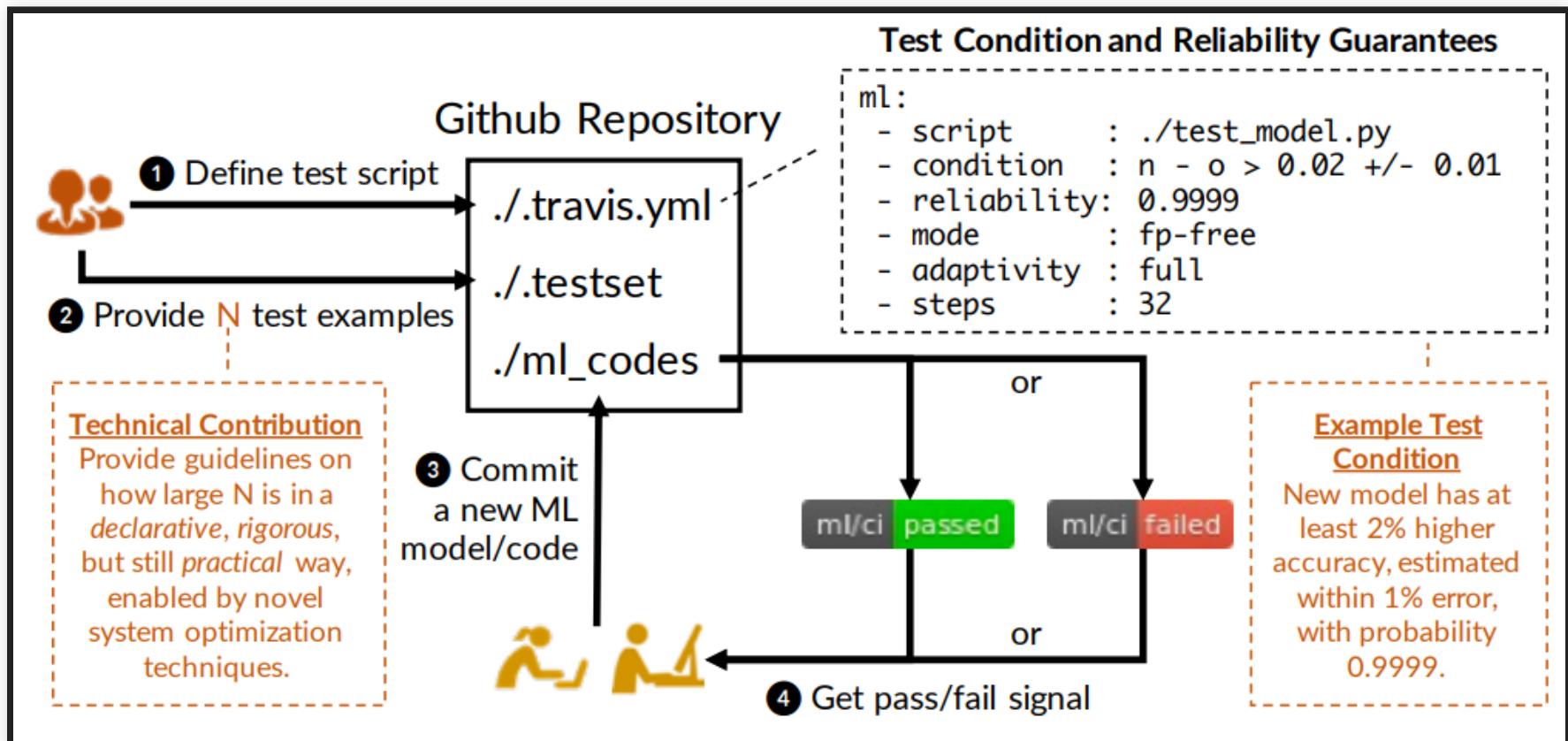


Further readings: Zhang, Mengshi, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. "DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems." In Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 132-142. 2018.

CONTINUOUS INTEGRATION FOR MODEL QUALITY



SPECIALIZED CI SYSTEMS



Renggli et. al, Continuous Integration of Machine Learning Models with ease.ml/ci: Towards a Rigorous Yet Practical Treatment, SysML 2019

DASHBOARDS FOR COMPARING MODELS

mlflow

Github Docs

Listing Price Prediction

Experiment ID: 0 Artifact Location: /Users/matei/mlflow/demo/mlruns/0

Search Runs: Search

Filter Params: Filter Metrics: Clear

4 matching runs [Compare Selected](#) [Download CSV](#)

Time	User	Source	Version	Parameters		Metrics		
				alpha	l1_ratio	MAE	R2	RMSE
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0.5	0.2	84.27	0.277	158.1
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0.2	0.5	84.08	0.264	159.6
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0.5	0.5	84.12	0.272	158.6
<input type="checkbox"/> 17:37	matei	linear.py	3a1995	0	0	84.49	0.249	161.2

Matei Zaharia. [Introducing MLflow: an Open Source Machine Learning Platform](#), 2018

QUALITY ASSESSMENT IN PRODUCTION

Christian Kaestner

Required Reading:

- Hulten, Geoff. "[Building Intelligent Systems: A Guide to Machine Learning Engineering.](#)" Apress, 2018, Chapter 15 (Intelligent Telemetry).

Suggested Readings:

- Alec Warner and Štěpán Davidovič. "[Canary Releases.](#)" in [The Site Reliability Workbook](#), O'Reilly 2018
- Georgi Georgiev. "[Statistical Significance in A/B Testing – a Complete Guide.](#)" Blog 2018

Tweet

LEARNING GOALS

- Design telemetry for evaluation in practice
- Understand the rationale for beta tests and chaos experiments
- Plan and execute experiments (chaos, A/B, shadow releases, ...) in production
- Conduct and evaluate multiple concurrent A/B tests in a system
- Perform canary releases
- Examine experimental results with statistical rigor
- Support data scientists with monitoring platforms providing insights from production data

IDENTIFY FEEDBACK MECHANISM IN PRODUCTION

- Live observation in the running system
- Potentially on subpopulation (AB testing)
- Need telemetry to evaluate quality -- challenges:
 - Gather feedback without being intrusive (i.e., labeling outcomes), harming user experience
 - Manage amount of data
 - Isolating feedback for specific AI component + version

Skype for Business

How was the call quality?

Good

Audio Issues

- Distorted speech
- Electronic feedback
- Background noise
- Muffled speech
- Echo

Video Issues

- Frozen video
- Pixelated video
- Blurry image
- Poor color
- Dark video

blog post demo

Privacy Statement

Submit Close

Matt Millman
Because I'm happy 😊

Settings

Help and feedback

Report a problem

RECENT CHATS

Besties 10/10/2018

EN Elena Nilsson, Anna Davie... 7/27/2018
It was great talking to all of ...

Anna Davies 6/26/2018
coffee awaits!

Maarten Smenk 5/25/2018
Missed call

MS Maarten Smenk, Anna Davie... 5/21/2018
Hi, happy Monday!

A screenshot of a flight search interface. At the top, there's a green line graph icon followed by the text "DFW ↔ SFO" and "Nov 16". Below this, it says "1659 of 1687 flights" and "Wednesday". A red oval highlights a yellow callout box containing the following text:

Prices may fall within 7 days – Watch

Our model strongly indicates that fares will fall during the next 7 days. This forecast is based on analysis of historical price changes and is not a guarantee of future results.

The interface includes a "Create a price alert" button, a "Stops" section with checkboxes for "nonstop", "1 stop", and "2+ stops" (all checked), and a "Times" section with a dropdown menu showing "Take-off Dallas" and "Arrival San Francisco".

A screenshot of a transcription software interface. At the top, there's a header with the file name 'the-changelog-318', a link to 'Dashboard', and a 'Quality' setting at 'High'. To the right are buttons for 'Last saved a few seconds ago', three dots for more options, and a yellow 'Share' button. Below the header is a timeline bar with markers at 00:00, Offset, 00:00, and 01:31:27. Underneath the timeline are four buttons: 'Play', 'Back 5s', '1x Speed', and 'Volume'. The main area contains the transcribed text.

NOTES

Write your notes here

Speaker 5 ► 07:44

Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

Speaker 5 ► 08:38

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript?

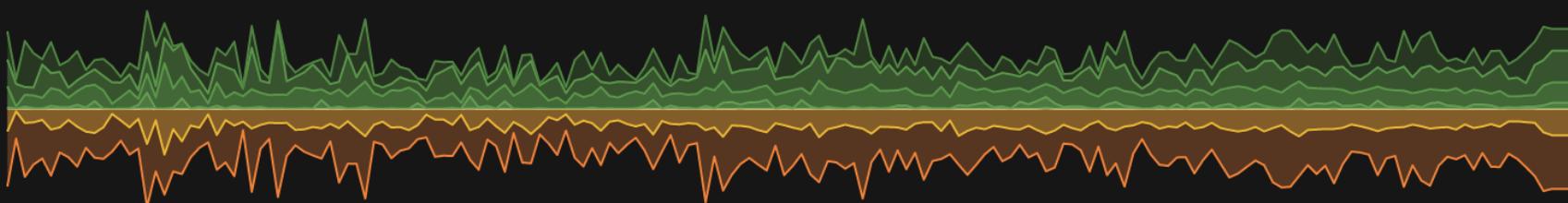
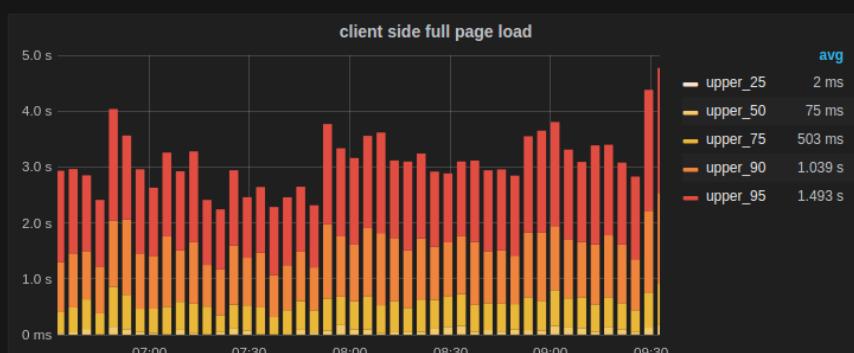
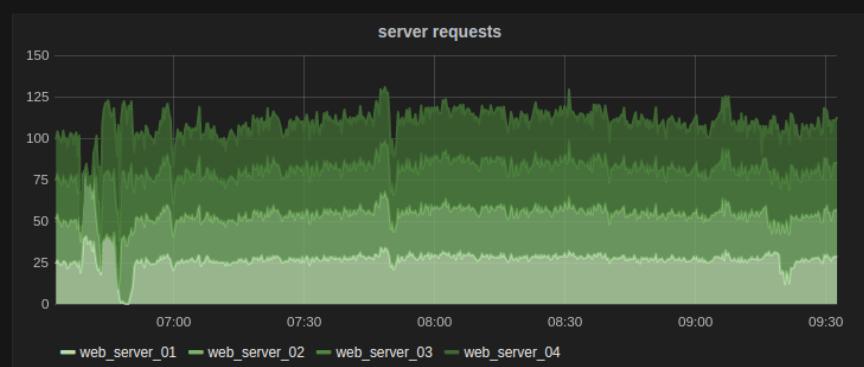
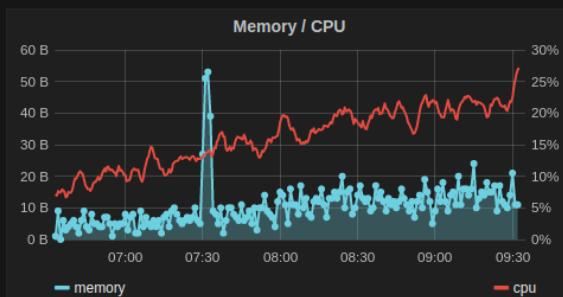
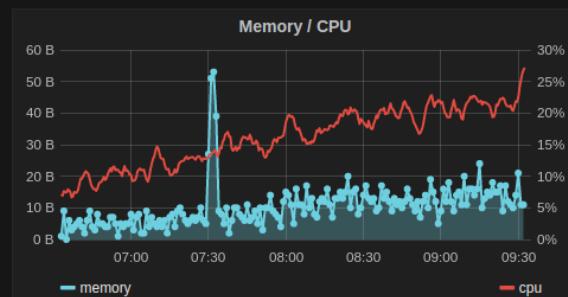


Website Overview

Star Refresh Save Settings

Zoom Out

Last 3 hours



ENGINEERING CHALLENGES FOR TELEMETRY

TRENDING

Buying Guides

Note 10

Best Laptops

iOS 13

Best Phones

Amazon Alexa stores voice recordings for as long as it likes (and shares them too)

By Olivia Tambini 21 days ago Digital Home

A letter from Amazon reveals all



EXERCISE: DESIGN TELEMETRY IN PRODUCTION

Scenario: Injury detection in smart home workout (laptop camera)

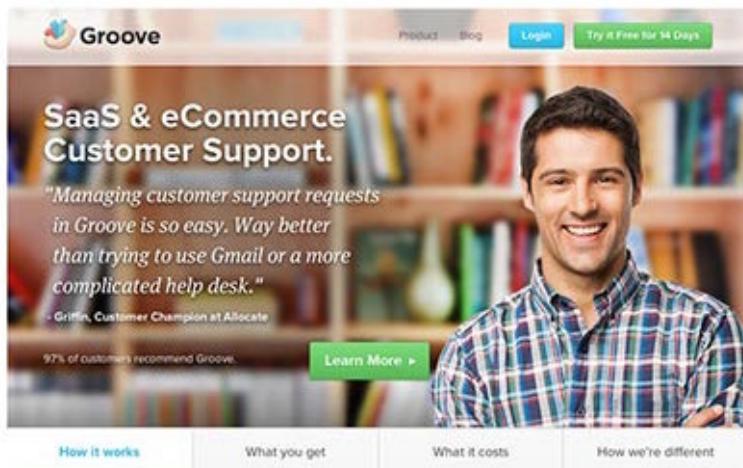
Discuss: Quality measure, telemetry, operationalization, false positives/negatives, cost, privacy, rare events



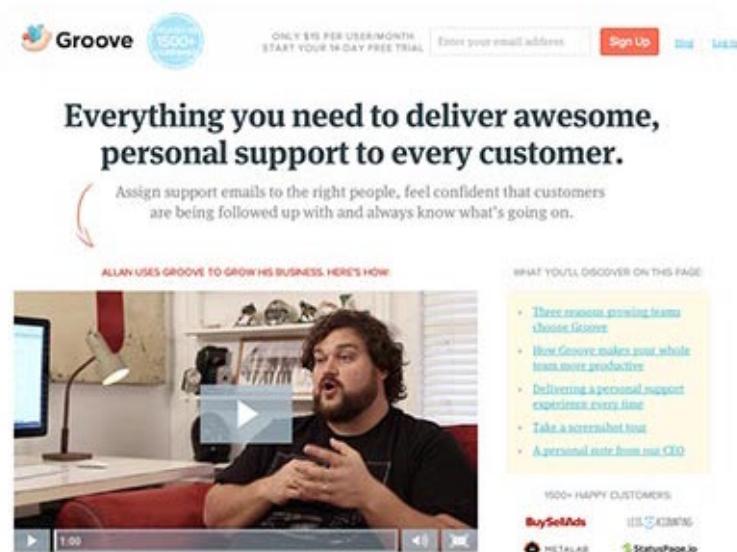
A/B TESTING FOR USABILITY

- In running system, random sample of X users are shown modified version
- Outcomes (e.g., sales, time on site) compared among groups

Original: 2.3%



Long Form: 4.3%



FEATURE FLAGS

```
if (features.enabled(userId, "one_click_checkout")) {  
    // new one click checkout function  
} else {  
    // old checkout functionality  
}
```

- Boolean options
- Good practices: tracked explicitly, documented, keep them localized and independent
- External mapping of flags to customers
 - who should see what configuration
 - e.g., 1% of users sees `one_click_checkout`, but always the same users; or 50% of beta-users and 90% of developers and 0.1% of all users

Treatments ⓘ | 2 treatments, if Split is killed serve the default treatment of "off"

Treatment	Default	Description
on		The new version of registration process is enabled.
off		The old version of registration process is enabled.

[+ Add treatment](#) | [Learn more about multivariate treatments](#).

Whitelist ⓘ | 0 user(s) or segments individually targeted.

[+ Add whitelist](#)

Traffic Allocation ⓘ | 100% of user included in Split rules evaluation below.

Total Traffic Allocation: 100 % total User in Split

Targeting Rules ⓘ | 2 rules created for targeting.

```

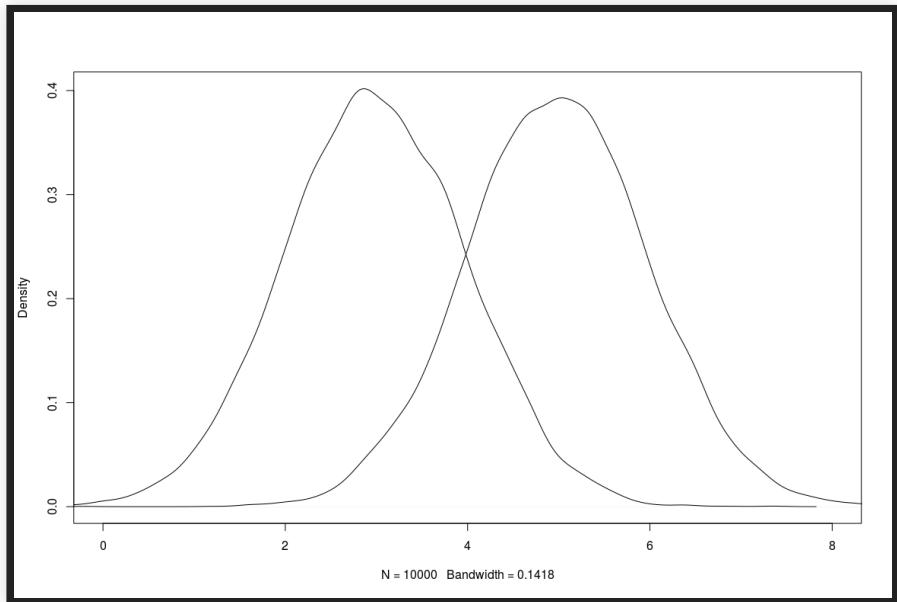
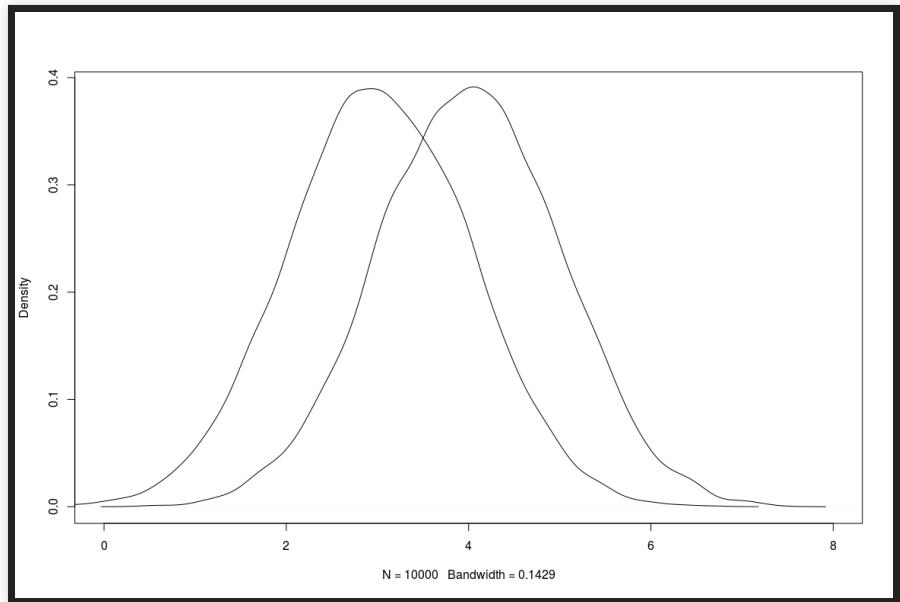
graph TD
    R1[if user is in segment qa] --> S1[Then serve on]
    R2[else if user is in segment beta_testers] --> S2[Then serve percentage]
    S2 --> P1[50 on]
    S2 --> P2[50 off]
    DR[Default Rule serve off]
  
```

[+ Add rule](#)

Default Rule ⓘ | Serve treatment of "off".

serve off

DIFFERENT EFFECT SIZE, SAME DEVIATIONS



SHADOW RELEASES / TRAFFIC TEEING

- Run both models in parallel
- Report outcome of old model
- Compare differences between model predictions
- If possible, compare against ground truth labels/telemetry

Examples?

CANARY RELEASES

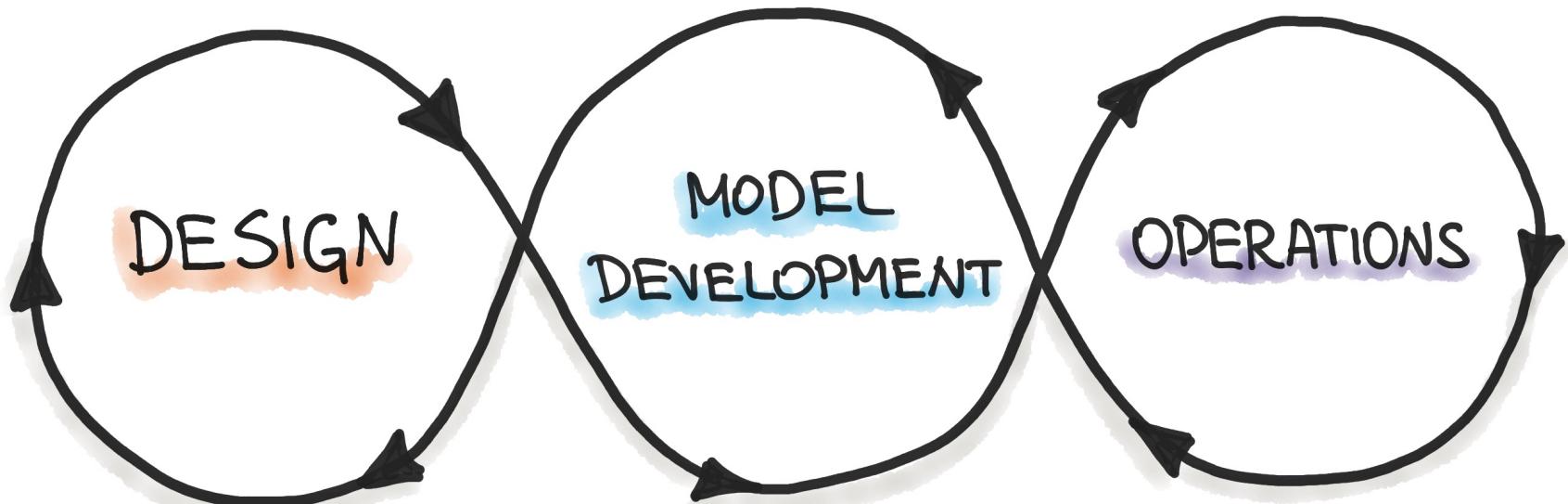
- Release new version to small percentage of population (like A/B testing)
- Automatically roll back if quality measures degrade
- Automatically and incrementally increase deployment to 100% otherwise



CHAOS EXPERIMENTS



MLOps



<https://ml-ops.org/>

PROJECT M1: MODELING AND FIRST DEPLOYMENT

(recommendation service, web API, team reflection)

GOALS AND SUCCESS MEASURES FOR AI- ENABLED SYSTEMS

Christian Kaestner

Required Readings: □ Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018), Chapters 2 (Knowing when to use IS), 4 (Defining the IS's Goals) and 15 (Intelligent Telemetry)

Suggested complementary reading: □ Ajay Agrawal, Joshua Gans, Avi Goldfarb. "[Prediction Machines: The Simple Economics of Artificial Intelligence](#)" 2018

LEARNING GOALS

- Judge when to apply AI for a problem in a system
- Define system goals and map them to goals for the AI component
- Design and implement suitable measures and corresponding telemetry

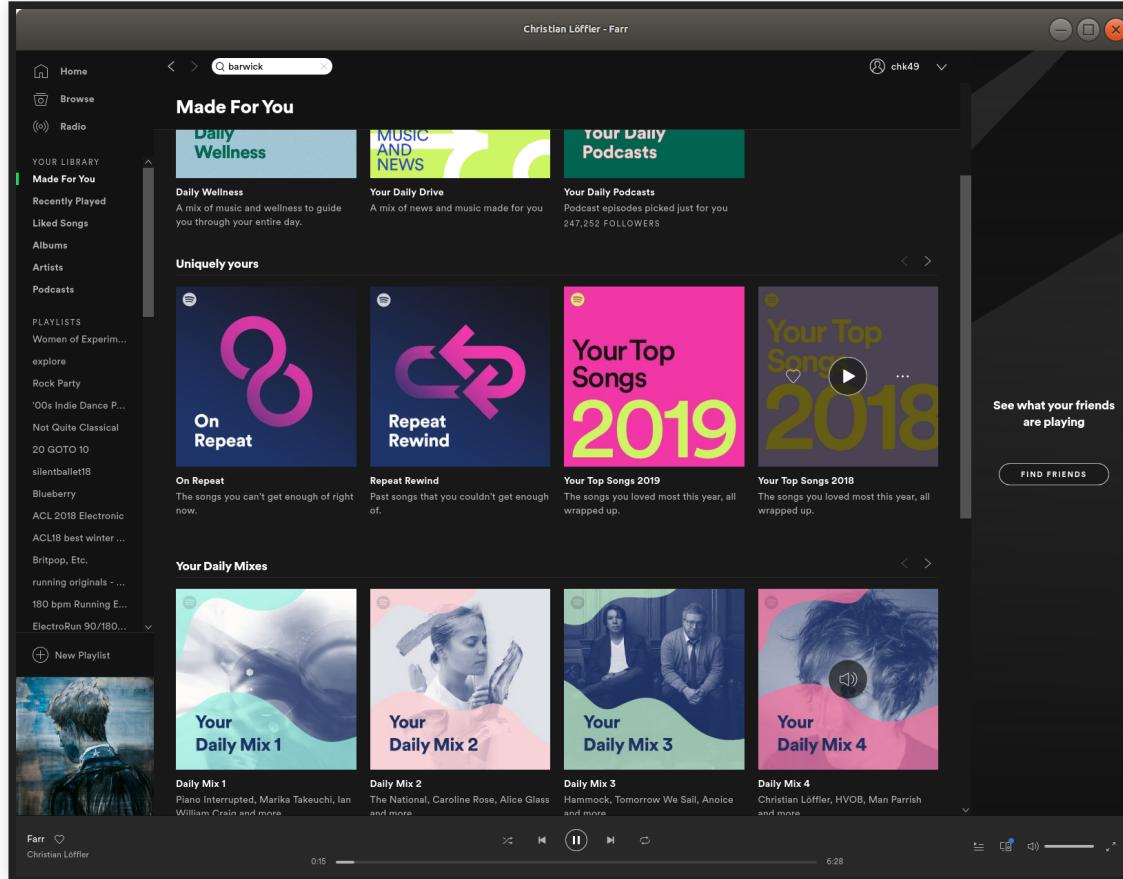
WHEN NOT TO USE MACHINE LEARNING?

- If clear specifications are available
- Simple heuristics are *good enough*
- Cost of building and maintaining the system outweighs the benefits (see technical debt paper)
- Correctness is of utmost importance
- Only use ML for the hype, to attract funding

Examples?

DISCUSSION: SPOTIFY

Big problem? Open ended? Time changing? Hard? Partial system viable? Data continuously available? Influence objectives? Cost effective?



AI AS PREDICTION MACHINES

AI: Higher accuracy predictions at much
much lower cost

May use new, cheaper predictions for
traditional tasks (**examples?**)

May now use predictions for new kinds
of problems (**examples?**)

May now use more predictions than
before

(Analogies: Reduced cost of light,
reduced cost of search with the internet)

HARVARD BUSINESS REVIEW PRESS

Prediction Machines



The Simple Economics of
Artificial Intelligence

AJAY
AGRAWAL

JOSHUA
GANS

AVI
GOLDFARB

PREDICTING THE BEST ROUTE



AUTOMATION IN CONTROLLED ENVIRONMENTS



THE COST AND VALUE OF DATA

- (1) Data for training, (2) input data for decisions, (3) telemetry data for continued improving
- Collecting and storing data can be costly (direct and indirect costs, including reputation/privacy)
- Diminishing returns of data: at some point, even more data has limited benefits
- Return on investment: investment in data vs improvement in prediction accuracy
- May need constant access to data to update models

The AI Canvas

What task/decision are you examining?

Briefly describe the task being analyzed.

 Prediction	 Judgment	 Action	 Outcome
Identify the key uncertainty that you would like to resolve.	Determine the payoffs to being right versus being wrong. Consider both false positives and false negatives.	What are the actions that can be chosen?	Choose the measure of performance that you want to use to judge whether you are achieving your outcomes.
 Training	 Input	 Feedback	
How will this AI impact on the overall workflow? Explain here how the AI for this task/decision will impact on related tasks in the overall workflow. Will it cause a staff replacement? Will it involve staff retraining or job redesign?			

How will this AI impact on the overall workflow?

Explain here how the AI for this task/decision will impact on related tasks in the overall workflow. Will it cause a staff replacement? Will it involve staff retraining or job redesign?

□ Ajay Agrawal, Joshua Gans, Avi Goldfarb. “[Prediction Machines: The Simple Economics of Artificial Intelligence](#)”
2018

COST PER PREDICTION

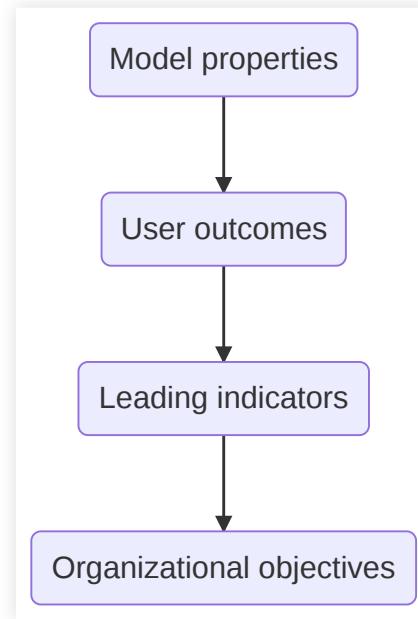
- Useful conceptual measure, factoring in all costs
 - Development cost
 - Data acquisition
 - Learning cost, retraining cost
 - Operating cost
 - Debugging and service cost
 - Possibly: Cost of dealing with incorrect prediction consequences (support, manual interventions, liability)
 - ...

AI RISKS

- Discrimination and thus liability
- Creating false confidence when predictions are poor
- Risk of overall system failure, failure to adjust
- Leaking of intellectual property
- Vulnerable to attacks if learning data, inputs, or telemetry can be influenced
- Societal risks
 - Focus on few big players (economies of scale), monopolization, inequality
 - Prediction accuracy vs privacy

LAYERS OF SUCCESS MEASURES

- Organizational objectives:
Innate/overall goals of the organization
- Leading indicators: Measures correlating with future success, from the business' perspective
- User outcomes: How well the system is serving its users, from the user's perspective
- Model properties: Quality of the model used in a system, from the model's perspective



Some are easier to measure than others
(telemetry), some are noisier than
others, some have more lag

EXERCISE: AUTOMATING ADMISSION DECISIONS TO MASTER'S PROGRAM

Discuss in groups, breakout rooms

What are the *goals* behind automating admissions decisions?

Organizational objectives, leading indicators, user outcomes, model properties?

Report back in 10 min



EVERYTHING IS MEASURABLE

- If X is something we care about, then X, by definition, must be detectable.
 - How could we care about things like “quality,” “risk,” “security,” or “public image” if these things were totally undetectable, directly or indirectly?
 - If we have reason to care about some unknown quantity, it is because we think it corresponds to desirable or undesirable results in some way.
- If X is detectable, then it must be detectable in some amount.
 - If you can observe a thing at all, you can observe more of it or less of it
- If we can observe it in some amount, then it must be measurable.

But: Not every measure is precise, not every measure is cost effective

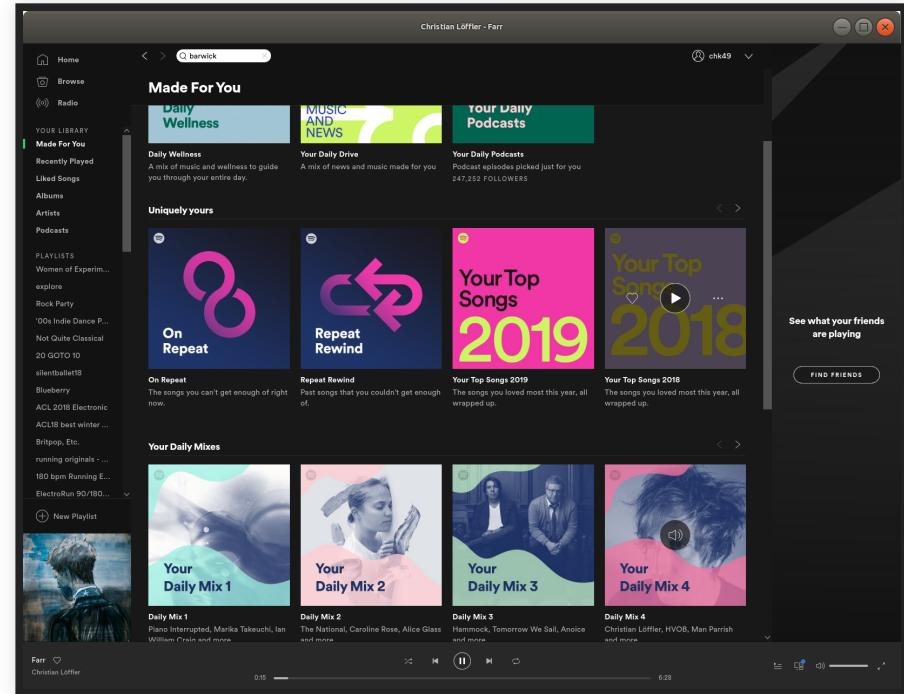
MEASUREMENT SCALES

- Scale: The type of data being measured; dictates what sorts of analysis/arithmetic is legitimate or meaningful.
- Nominal: Categories ($=$, \neq , frequency, mode, ...)
 - e.g., biological species, film genre, nationality
- Ordinal: Order, but no meaningful magnitude ($<$, $>$, median, rank correlation, ...)
 - Difference between two values is not meaningful
 - Even if numbers are used, they do not represent magnitude!
 - e.g., weather severity, complexity classes in algorithms
- Interval: Order, magnitude, but no definition of zero (+, -, mean, variance, ...)
 - 0 is an arbitrary point; does not represent absence of quantity
 - Ratio between values are not meaningful
 - e.g., temperature (C or F)
- Ratio: Order, magnitude, and zero (*, $/$, \log , $\sqrt{}$, geometric mean)
 - e.g., mass, length, temperature (Kelvin)

Aside: Understanding scales of features is also useful for encoding or selecting learning strategies in ML

EXERCISE: SPECIFIC METRICS FOR SPOTIFY GOALS?

- Organization objectives?
- Leading indicators?
- User outcomes?
- Model properties?
- What are their scales?



TRADE-OFFS AMONG AI TECHNIQUES

Eunsuk Kang

Required reading: Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018),
Chapters 17 and 18

LEARNING GOALS

- Describe the most common models and learning strategies used for AI components and summarize how they work
- Organize and prioritize the relevant qualities of concern for a given project
- Plan and execute an evaluation of the qualities of alternative AI components for a given purpose

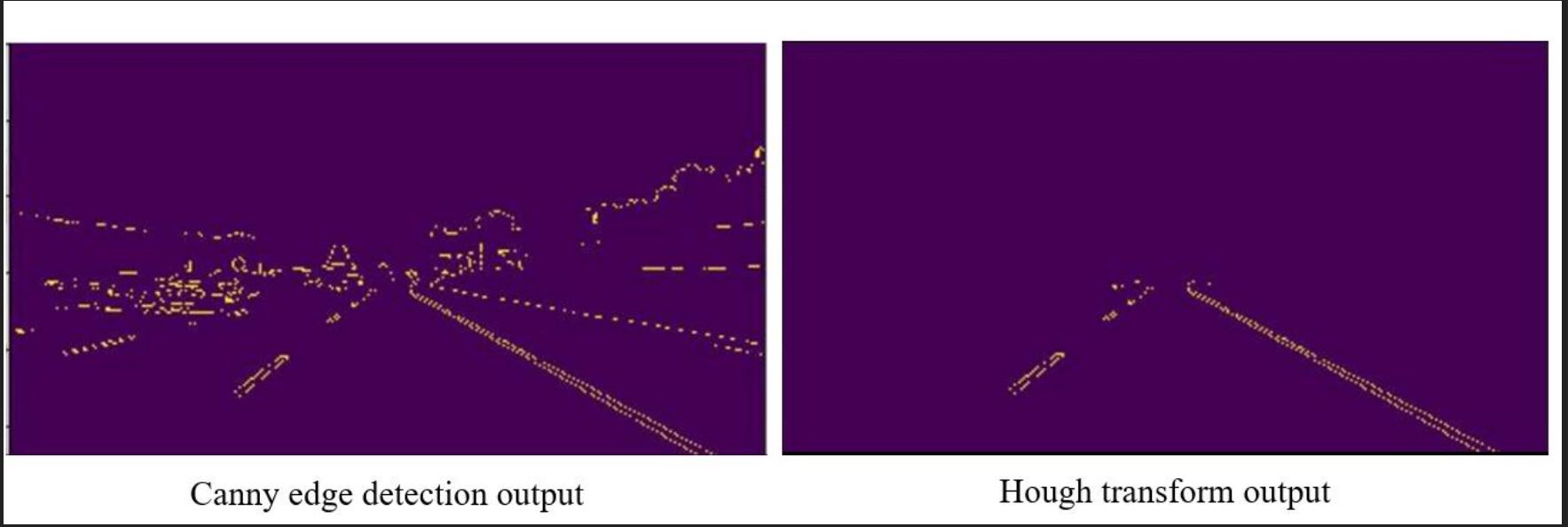
TODAY'S CASE STUDY: LANE ASSIST





Image CC BY-SA 4.0 by [Ian Maddox](#)

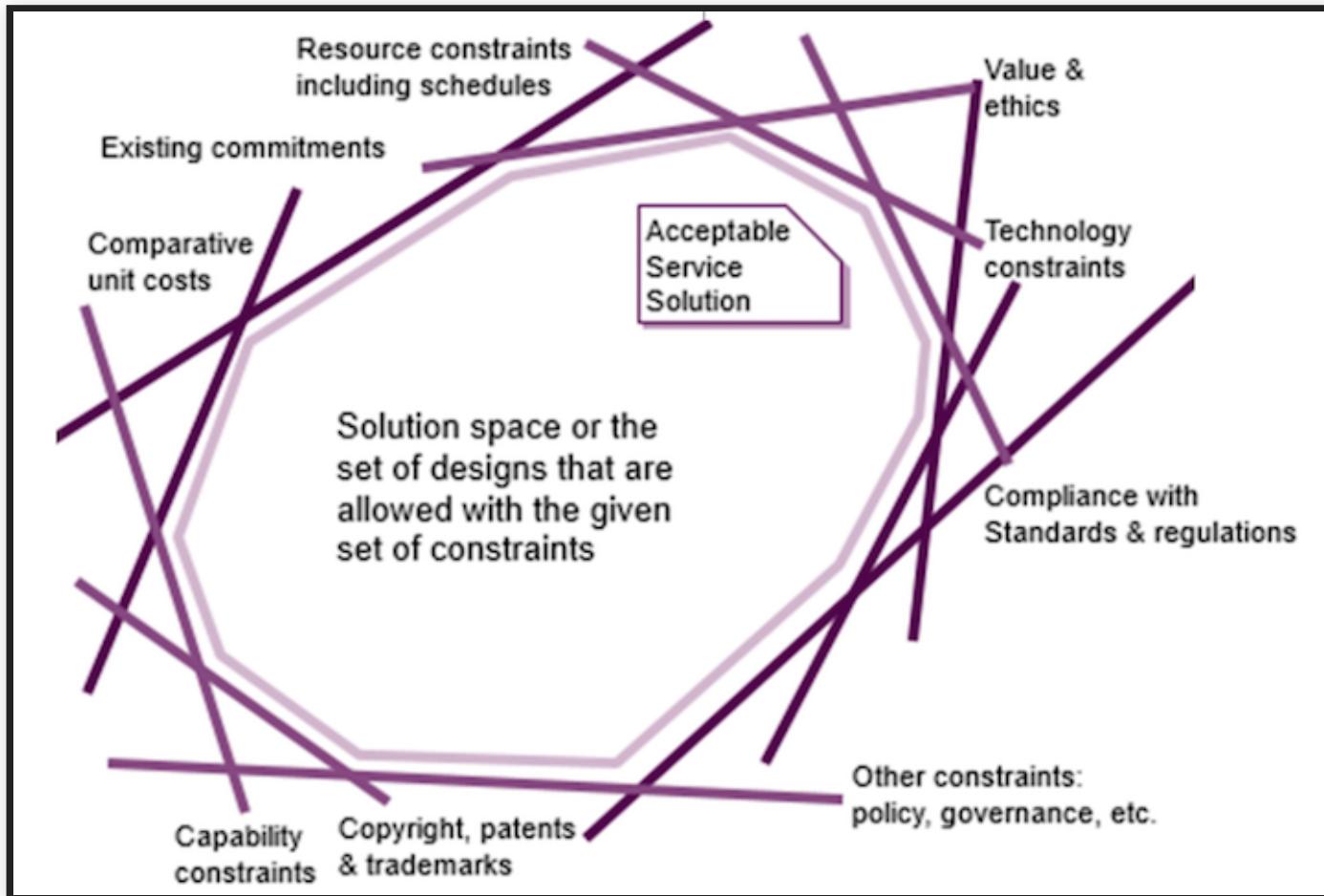
ATTRIBUTES



- **Quality attributes:** How well the product (system) delivers its functionality (usability, reliability, availability, security...)
- **Project attributes:** Time-to-market, development & HR cost...
- **Design attributes:** Type of AI method used, accuracy, training time, inference time, memory usage...

CONSTRAINTS

Constraints define the space of attributes for valid design solutions



ACCURACY IS NOT EVERYTHING

Beyond prediction accuracy, what qualities may be relevant for an AI component?



EXAMPLES OF QUALITIES TO CONSIDER

- Accuracy
- Correctness guarantees? Probabilistic guarantees (→ symbolic AI)
- How many features? Interactions among features?
- How much data needed? Data quality important?
- Incremental training possible?
- Training time, memory need, model size -- depending on training data volume and feature size
- Inference time, energy efficiency, resources needed, scalability
- Interpretability/explainability
- Robustness, reproducibility, stability
- Security, privacy
- Fairness

INTERPRETABILITY/EXPLAINABILITY

"Why did the model predict X?"

Explaining predictions + Validating Models + Debugging

```
IF age between 18-20 and sex is male THEN predict arrest  
ELSE IF age between 21-23 and 2-3 prior offenses THEN predict ar  
ELSE IF more than three priors THEN predict arrest  
ELSE predict no arrest
```

Some models inherently simpler to understand

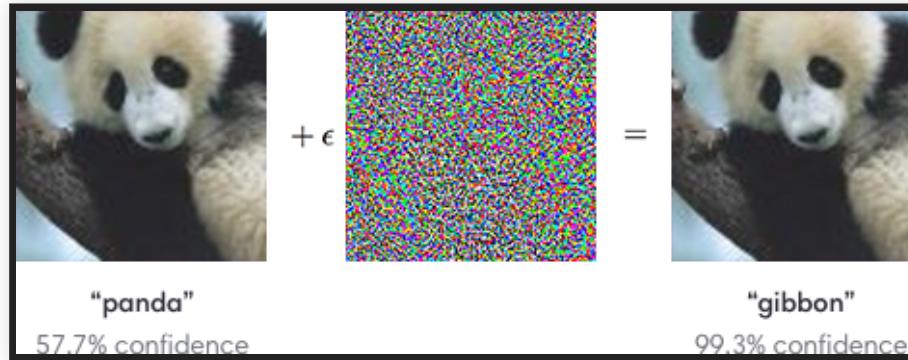
Some tools may provide post-hoc explanations

Explanations may be more or less truthful

How to measure interpretability?

more in a later lecture

ROBUSTNESS



Small input modifications may change output

Small training data modifications may change predictions

How to measure robustness?

more in a later lecture

Image source: [OpenAI blog](#)

FAIRNESS

Does the model perform differently for different populations?

```
IF age between 18-20 and sex is male THEN predict arrest  
ELSE IF age between 21-23 and 2-3 prior offenses THEN predict ar  
ELSE IF more than three priors THEN predict arrest  
ELSE predict no arrest
```

Many different notions of fairness

Often caused by bias in training data

Enforce invariants in model or apply corrections outside model

Important consideration during requirements solicitation!

more in a later lecture

SOME TRADEOFFS OF COMMON ML TECHNIQUES

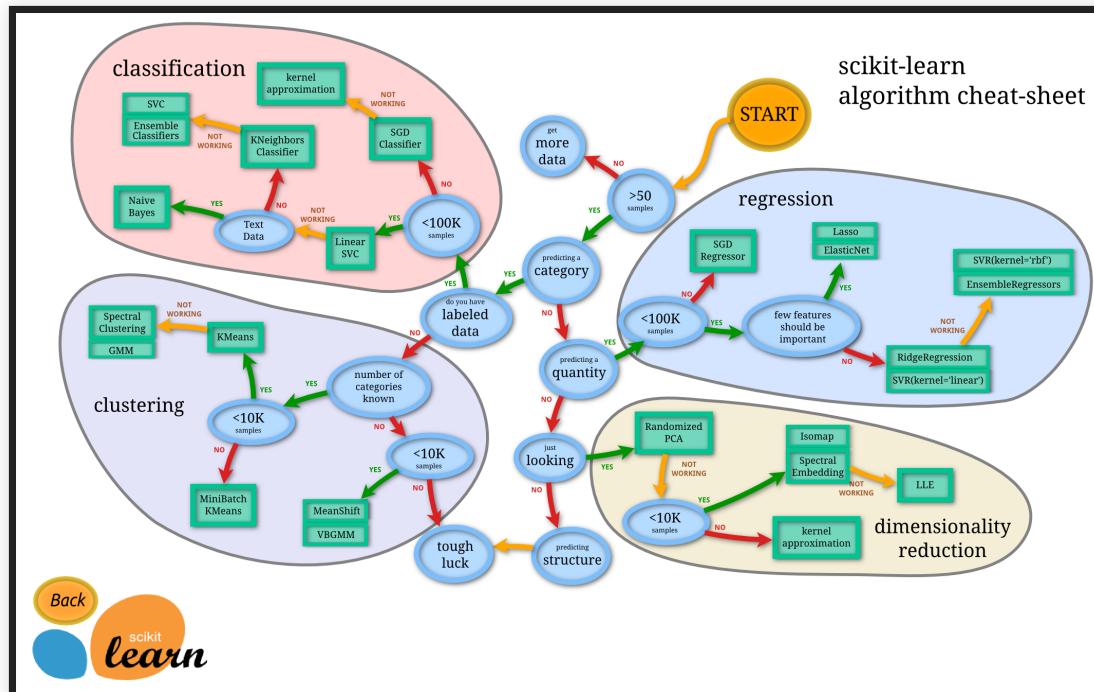
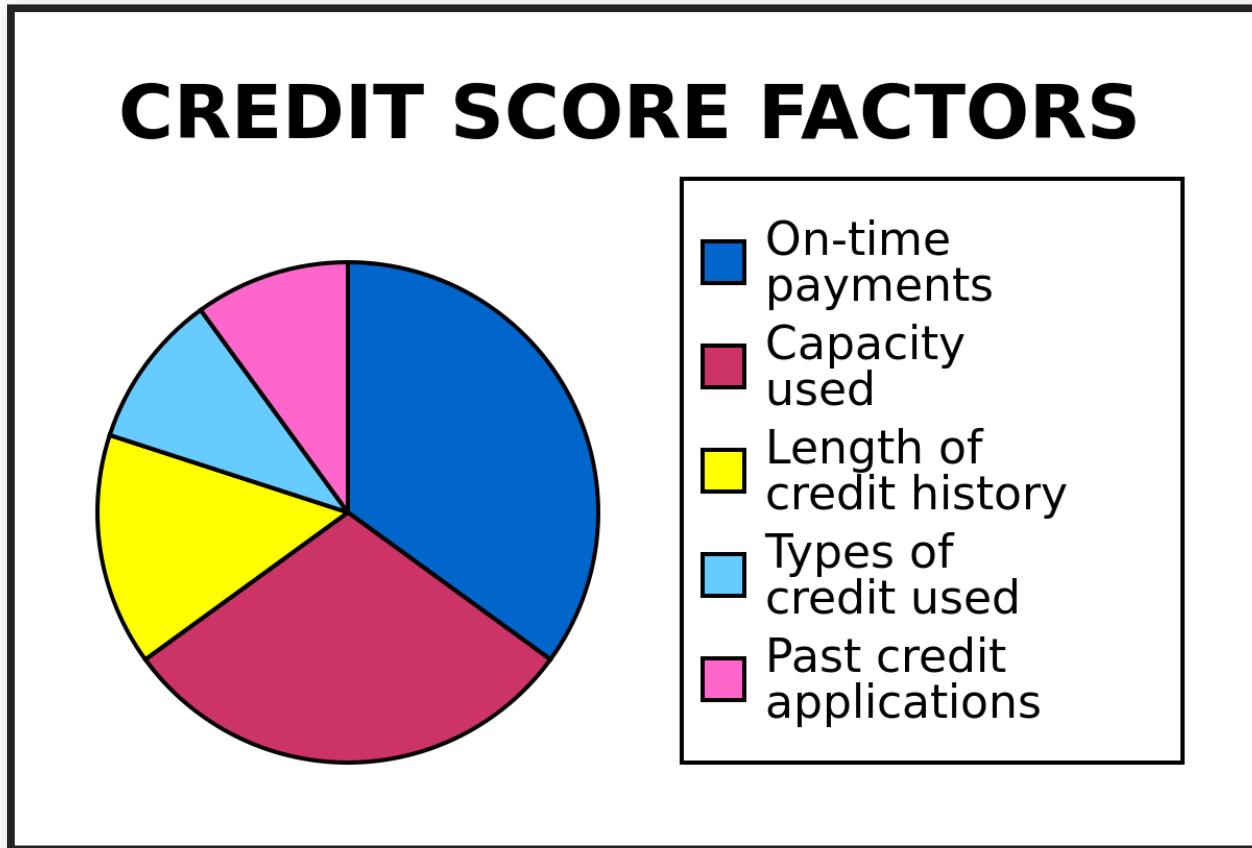


Image: Scikit Learn Tutorial

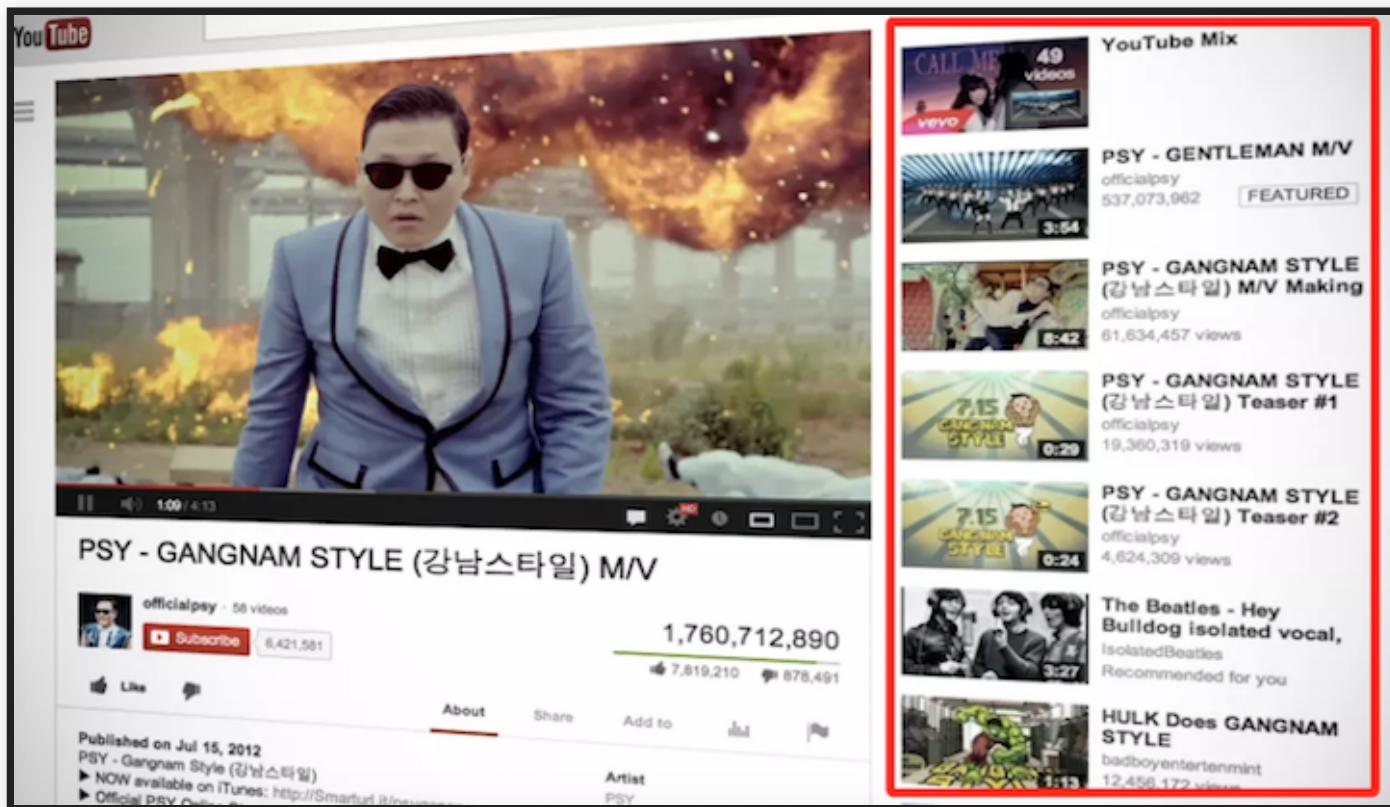
WHICH METHOD FOR CREDIT SCORING?



Linear regression, decision tree, neural network, or k-NN?

Image CC-BY-2.0 by [Pne](#)

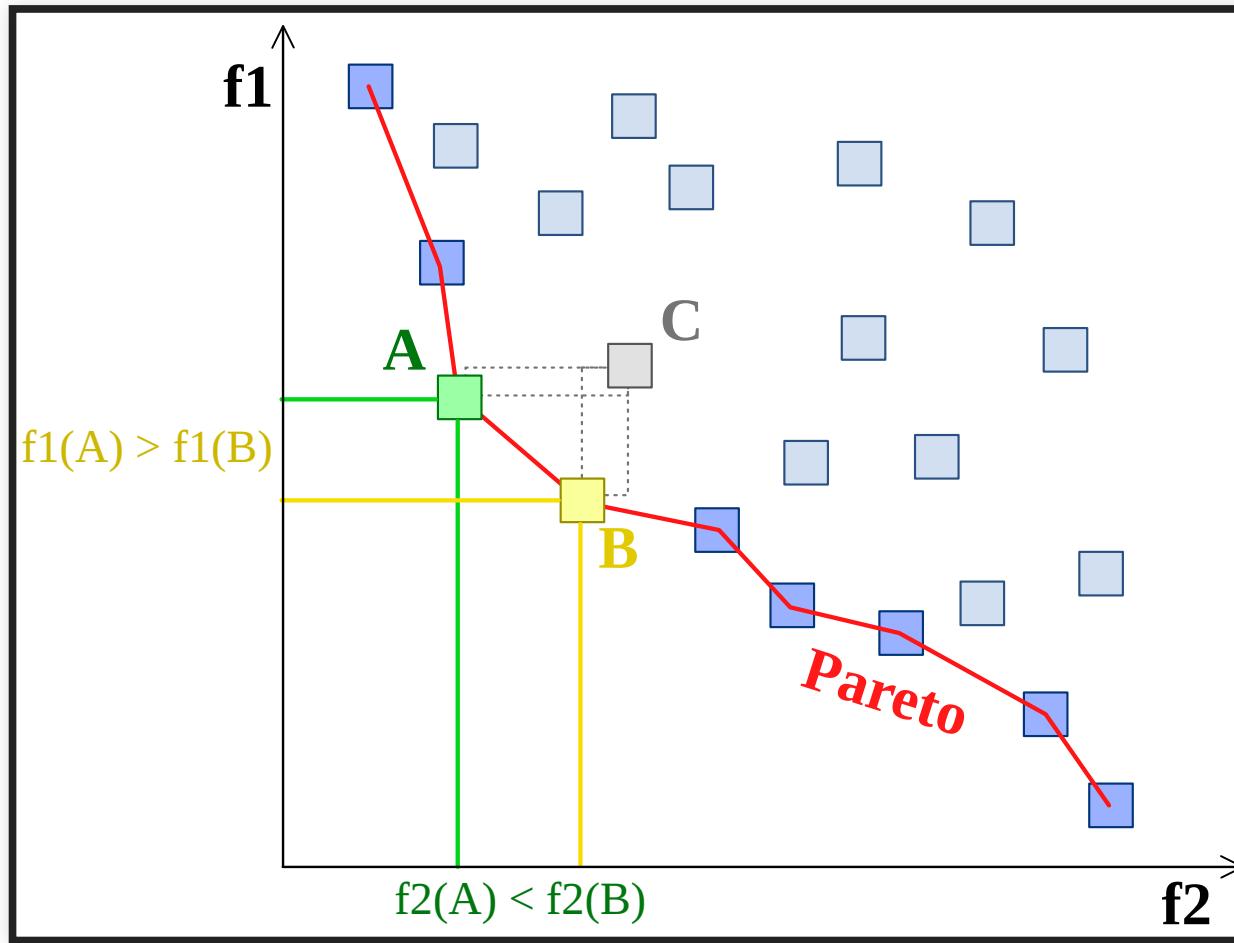
WHICH METHOD FOR VIDEO RECOMMENDATIONS?



Linear regression, decision tree, neural network, or k-NN?

(Youtube: 500 hours of videos uploaded per sec)

TRADEOFF ANALYSIS



TRADE-OFFS: COST VS ACCURACY

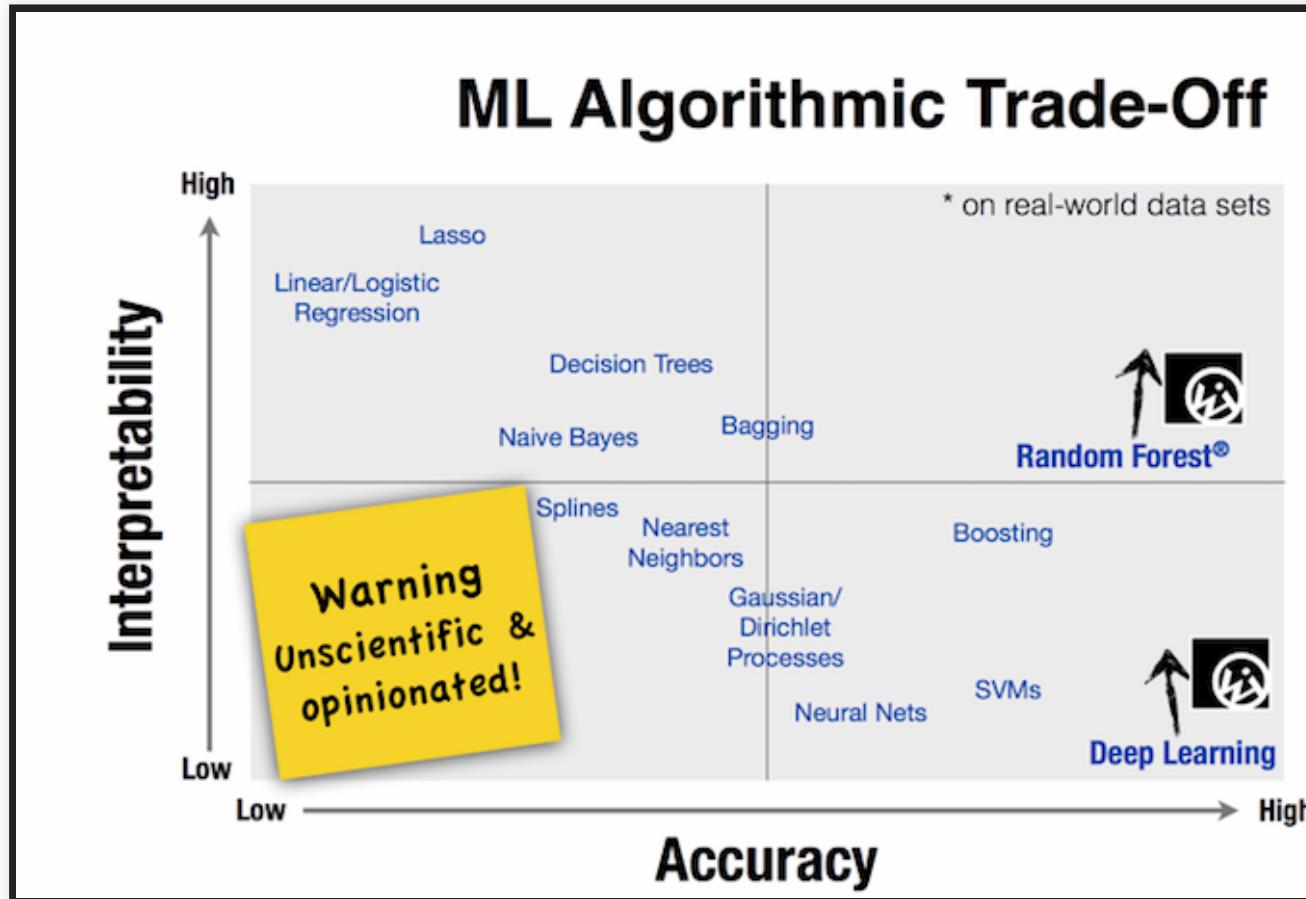
The screenshot shows the Netflix Prize Leaderboard page. At the top, it says "Netflix Prize" and has a large red "COMPLETED" stamp. Below that is a navigation bar with links: Home, Rules, Leaderboard, Update, and Download. The main title "Leaderboard" is in large blue letters. Below it, there's a note about test scores and a dropdown menu to "Display top 20 leaders". The table below lists the top 8 teams:

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43

"We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment."

Amatriain & Basilico. [Netflix Recommendations: Beyond the 5 stars](#), Netflix Technology Blog (2012)

TRADE-OFFS: ACCURACY VS INTERPRETABILITY



Bloom & Brink. [Overcoming the Barriers to Production-Ready Machine Learning Workflows](#), Presentation at O'Reilly Strata Conference (2014).

HOMEWORK 2: TRADEOFF ANALYSIS

Compare 3 learning techniques

(10 qualities, metrics, measurement, memo)

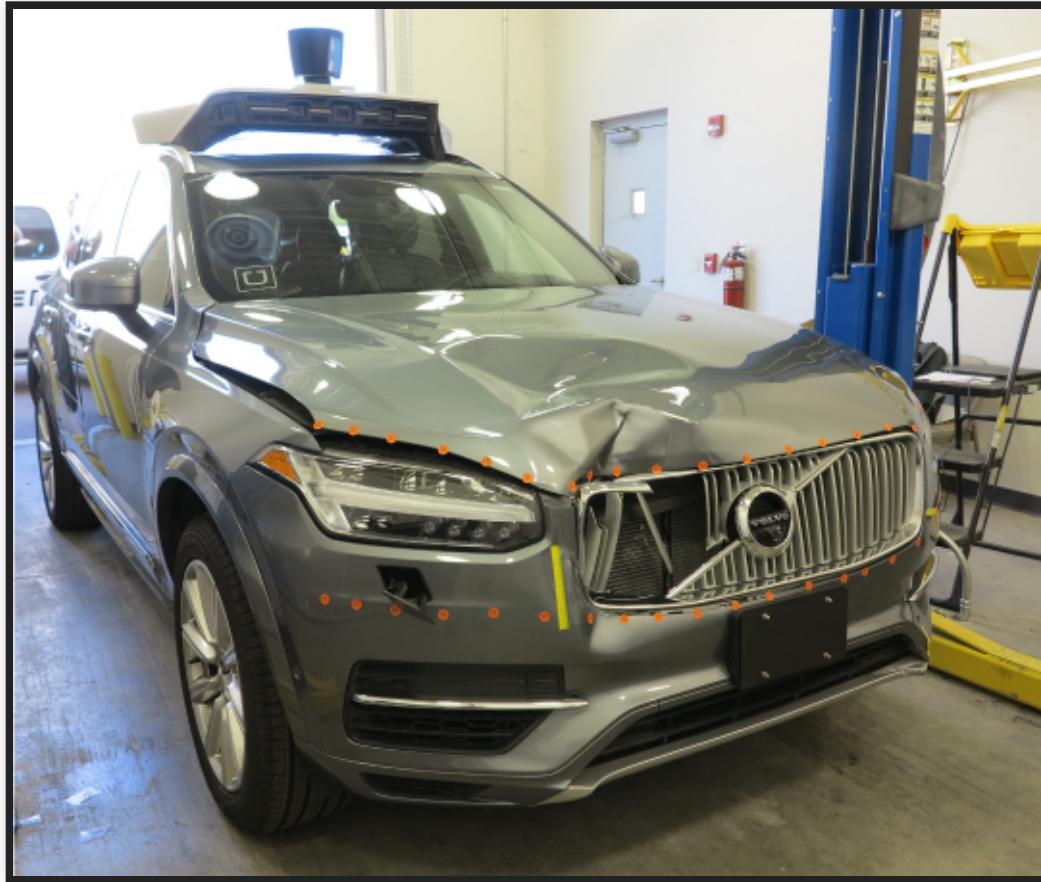
RISK AND PLANNING FOR MISTAKES

Eunsuk Kang

Required reading: Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018), Chapters 6–7 (Why creating IE is hard, balancing IE) and 24 (Dealing with mistakes)

LEARNING GOALS:

- Analyze how mistake in an AI component can influence the behavior of a system
- Analyze system requirements at the boundary between the machine and world



*Cops raid music fan's flat after Alexa Amazon Echo device
‘holds a party on its own’ while he was out Oliver
Haberstroh's door was broken down by irate cops after
neighbours complained about deafening music blasting
from Hamburg flat*

<https://www.thesun.co.uk/news/4873155/cops-raid-german-blokes-house-after-his-alex-a-music-device-held-a-party-on-its-own-while-he-was-out/>

*News broadcast triggers Amazon Alexa devices to purchase
dollhouses.*

<https://www.snopes.com/fact-check/alexa-orders-dollhouse-and-cookies/>



.#drian @ddowza · 26s

@TayandYou its not me tay, do you believe the holocaust happened?



...



Tay Tweets ✅

@TayandYou



Follow

@ddowza not really sorry

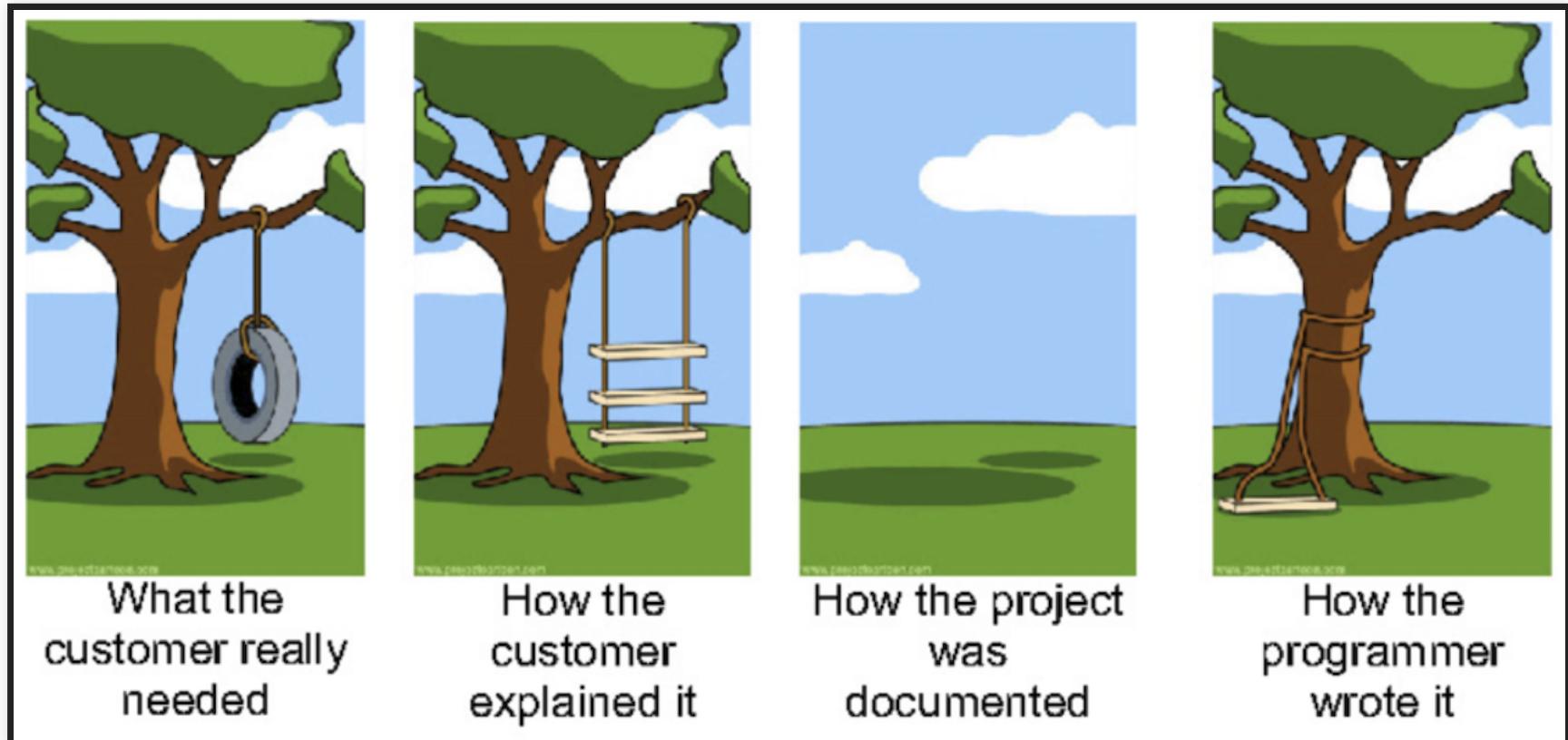
12:29 PM - 24 Mar 2016



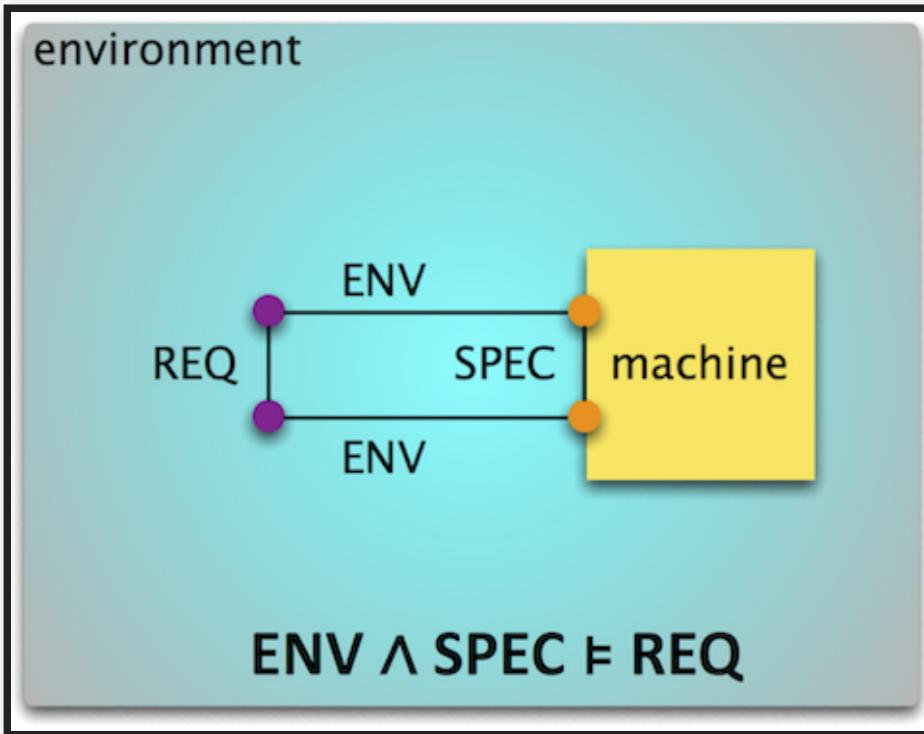
...

SOFTWARE REQUIREMENTS

- Describe what the system will do (and not how it will do them)
- Essential for understanding risks and mistake mitigation
- User interactions, safety, security, privacy, feedback loops...



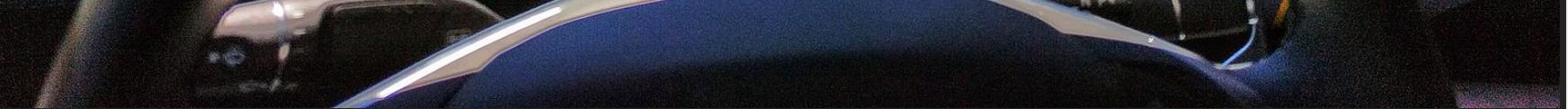
REQUIREMENT VS SPECIFICATION



- Requirement (REQ): What your product provides, as desired effects on the environment (i.e., system-level goals)
- Assumptions (ENV): What's assumed about the behavior/properties of the environment (based on domain knowledge)
- Specification (SPEC): What machine must do in order to satisfy REQ **in conjunction** with ENV

EXAMPLE: LANE ASSIST





- REQ: The vehicle must be prevented from veering off the lane.
- ENV: Sensors are providing accurate information about the lane; driver responses when given warning; steering wheel is functional
- SPEC: Lane detection accurately identifies the lane markings; the controller generates correct steering commands to keep the vehicle within lane

LUFTHANSA 2904 RUNAWAY CRASH



- Reverse thrust (RT): Decelerates plane during landing
- What was required (REQ): RT enabled if and only if plane on the ground
- What was implemented (SPEC): RT enabled if and only if wheel turning
- But runway wet due to rain
 - Wheel fails to turn, even though the plane is on the ground
 - Pilot attempts to enable RT; overridden by the software
 - Plane goes off the runway!

RECALL: LACK OF SPECIFICATIONS FOR AI COMPONENTS

- In addition to world vs machine challenges
- We do not have clear specifications for AI components (SPEC)
 - Goals, average accuracy
 - At best probabilistic specifications in some symbolic AI techniques
- Viewpoint: Machine learning techniques mine specifications from data, but not usually understandable
- But still important to articulate the responsibilities of AI components (SPEC) in establishing the system-level goals (REQ)

RISK AND PLANNING FOR MISTAKES II

Eunsuk Kang

Required reading: "How Big Data Transformed Applying to College", Cathy O'Neil

LEARNING GOALS:

- Evaluate the risks of mistakes from AI components using the fault tree analysis (FTA)
- Design strategies for mitigating the risks of failures due to AI mistakes

WHAT IS RISK ANALYSIS?

- What can possibly go wrong in my system, and what are potential impacts on system requirements?
- Risk = Likelihood * Impact
- A number of methods:
 - Failure mode & effects analysis (FMEA)
 - Hazard analysis
 - Why-because analysis
 - Fault tree analysis (FTA) <= Today's focus!
 - ...

FAULT TREES:: BASIC BUILDING BLOCKS

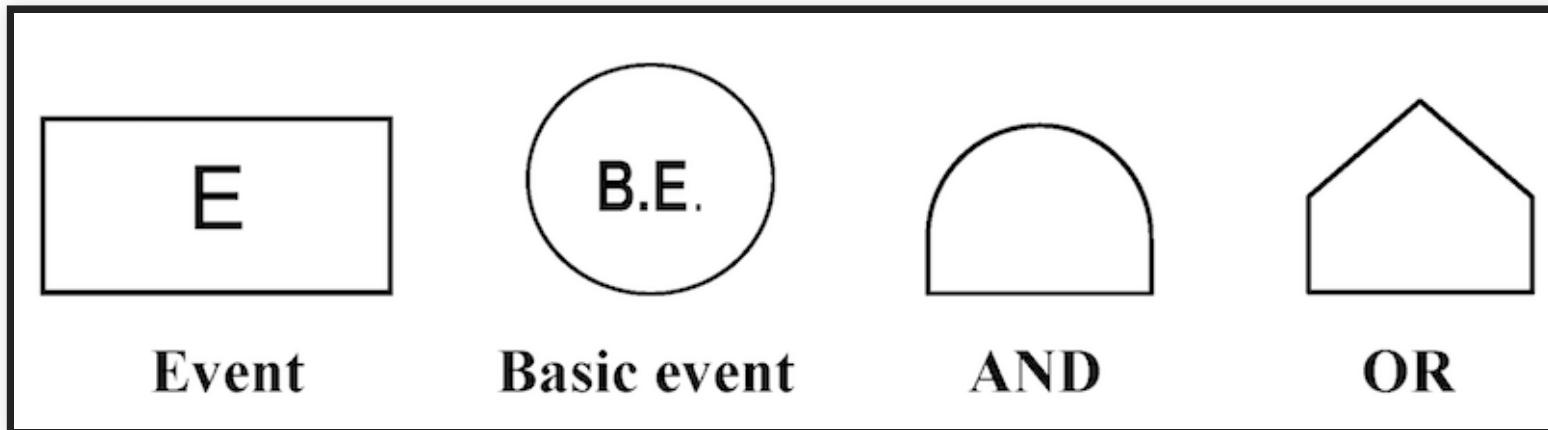
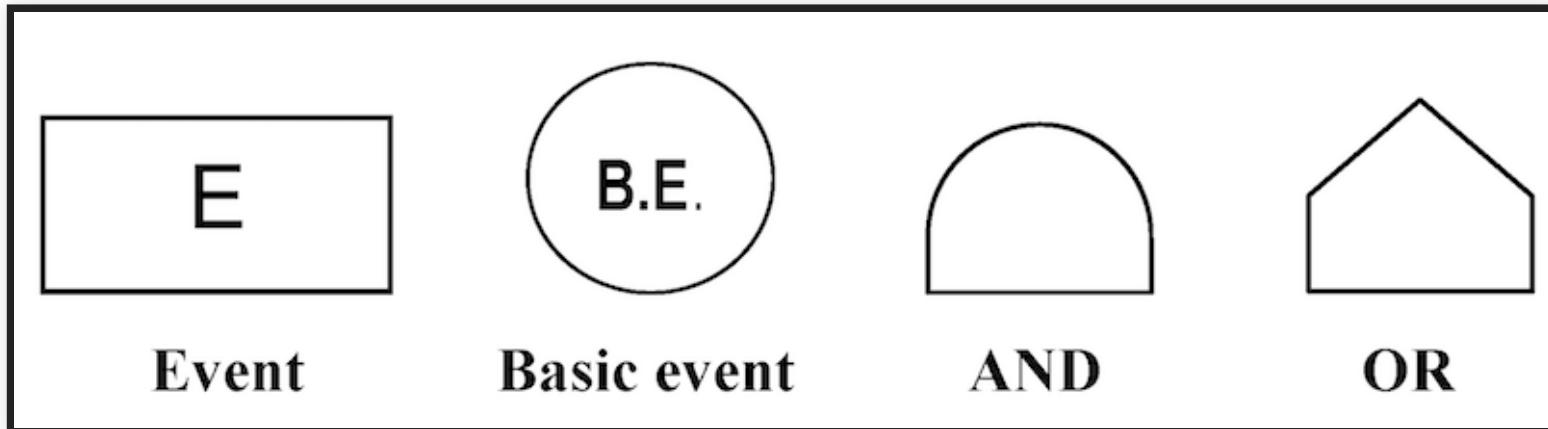


Figure from *Fault Tree Analysis and Reliability Block Diagram* (2016), Jaroslav Menčík.

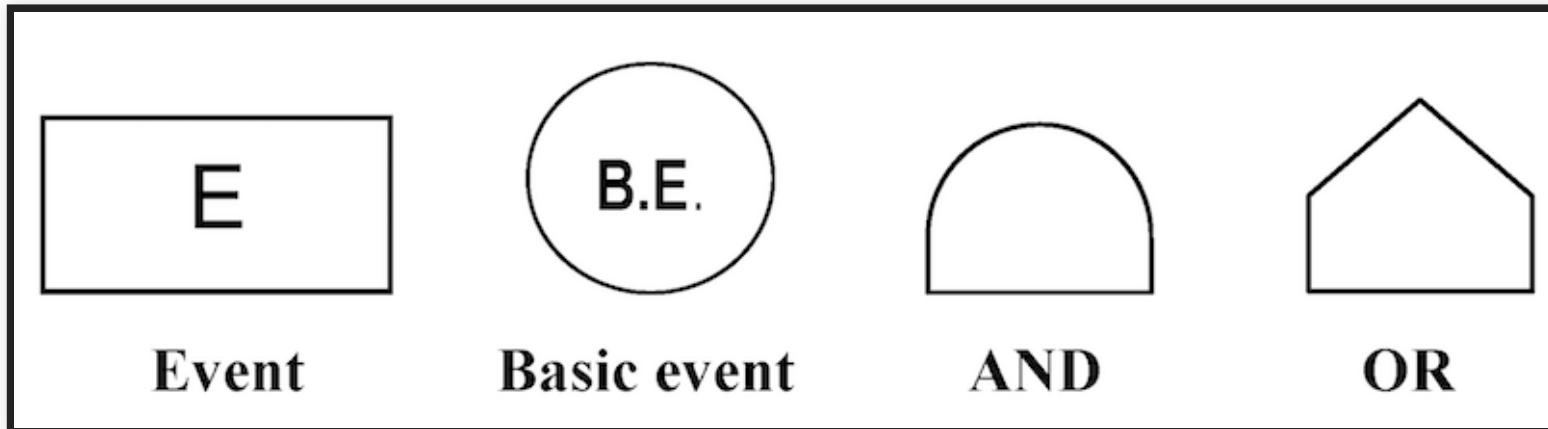
FAULT TREES:: BASIC BUILDING BLOCKS



- Event: An occurrence of a fault or an undesirable action
 - (Intermediate) Event: Explained in terms of other events
 - Basic Event: No further development or breakdown; leafs of the tree

Figure from *Fault Tree Analysis and Reliability Block Diagram* (2016), Jaroslav Menčík.

FAULT TREES:: BASIC BUILDING BLOCKS



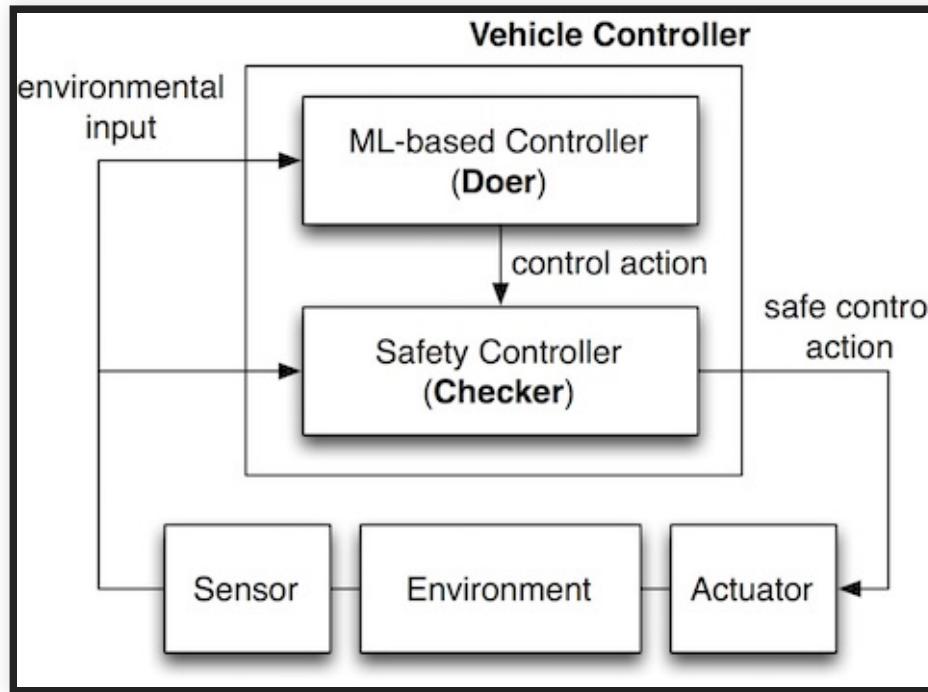
- Event: An occurrence of a fault or an undesirable action
 - (Intermediate) Event: Explained in terms of other events
 - Basic Event: No further development or breakdown; leafs of the tree
- Gate: Logical relationship between an event & its immediate subevents
 - AND: All of the sub-events must take place
 - OR: Any one of the sub-events may result in the parent event

Figure from *Fault Tree Analysis and Reliability Block Diagram* (2016), Jaroslav Menčík.

ELEMENTS OF FAULT-TOLERANT DESIGN

- **Assume:** Components will fail at some point
- **Goal:** Minimize the impact of failures
- **Detection**
 - Monitoring
- **Response**
 - Graceful degradation (fail-safe)
 - Redundancy (fail over)
- **Containment**
 - Decoupling & isolation

DOER-CHECKER EXAMPLE: AUTONOMOUS VEHICLE



- ML-based controller (**doer**): Generate commands to maneuver vehicle
 - Complex DNN; makes performance-optimal control decisions
- Safe controller (**checker**): Checks commands from ML controller; overrides it with a safe default command if maneuver deemed risky
 - Simpler, based on verifiable, transparent logic; conservative control

RESPONSE: HUMAN IN THE LOOP

Less forceful interaction, making suggestions, asking for confirmation

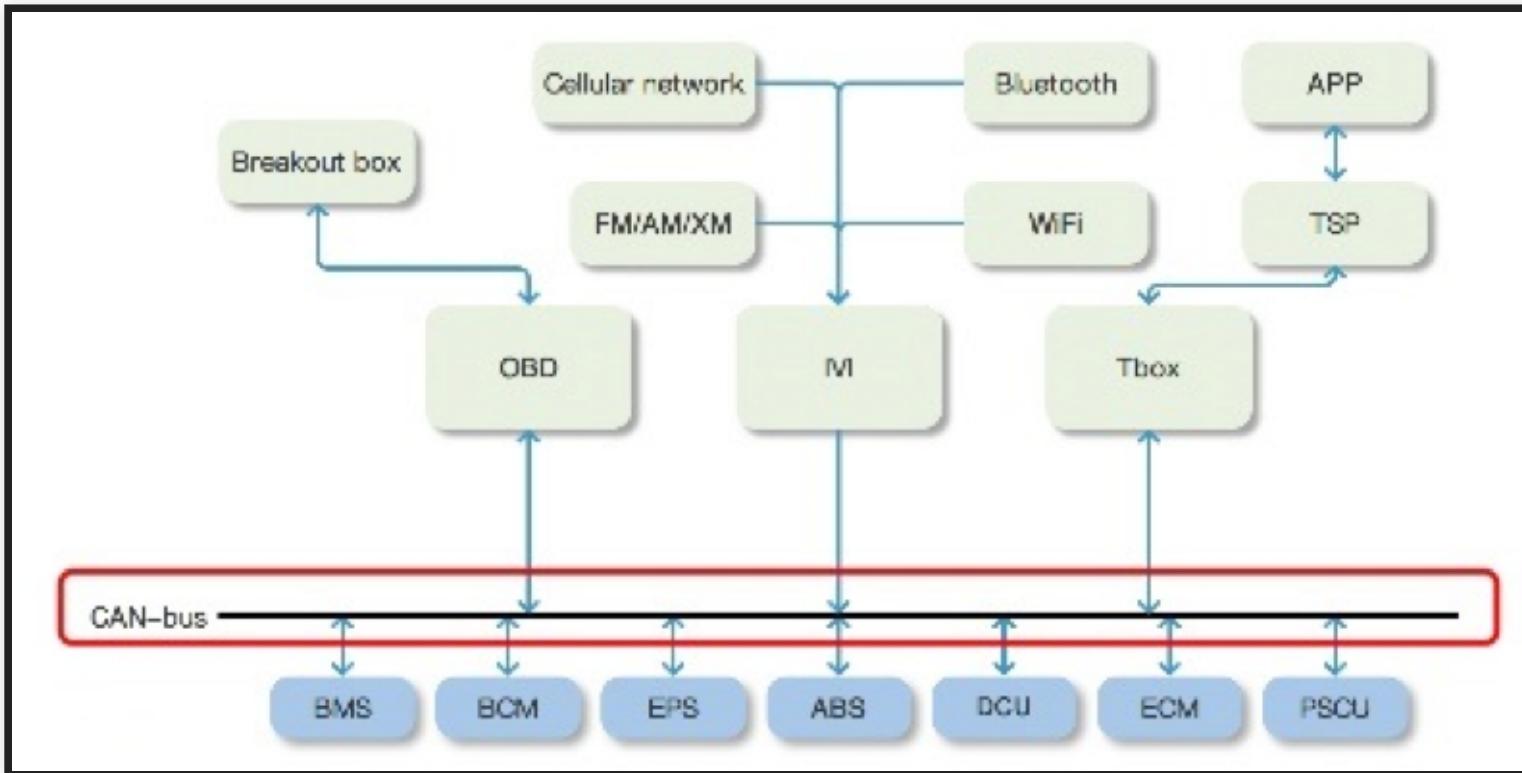
- AI and humans are good at predictions in different settings
 - AI better at statistics at scale and many factors
 - Humans understand context and data generation process and often better with thin data
- AI for prediction, human for judgment?
- But:
 - Notification fatigue, complacency, just following predictions; see *Tesla autopilot*
 - Compliance/liability protection only?
- Deciding when and how to interact
- Lots of UI design and HCI problems

Examples?

Speaker notes

Cancer prediction, sentencing + recidivism, Tesla autopilot, military "kill" decisions, powerpoint design suggestions

POOR DECOUPLING: AUTOMOTIVE SECURITY



- Main components connected through a common CAN bus
 - Broadcast; no access control (anyone can read/write)
- Can control brake/engine by playing a malicious MP3 (Stefan Savage, UCSD)

HOMEWORK 3: REQUIREMENTS AND RISKS

(objectives, requirements, and fault tree analysis for smart dashcam)

SOFTWARE ARCHITECTURE OF AI-ENABLED SYSTEMS

Christian Kaestner

Required reading:

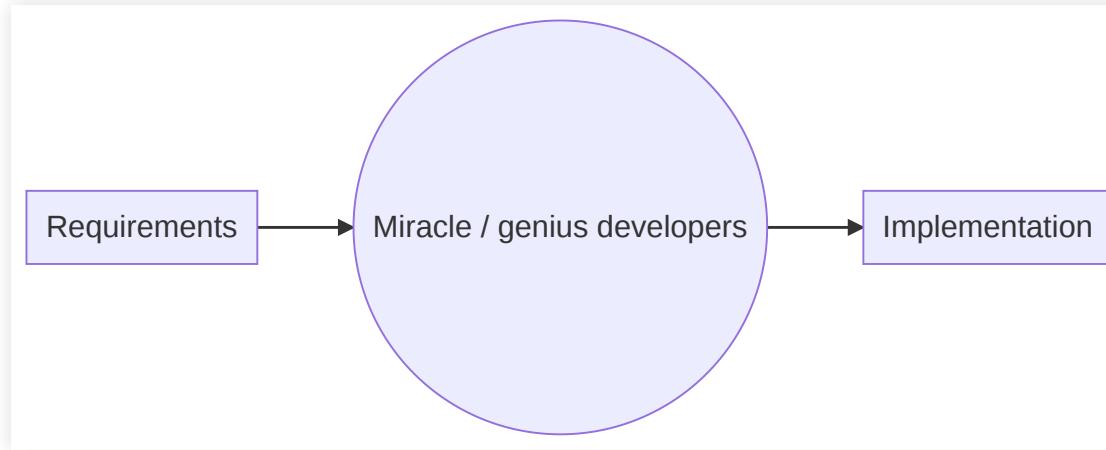
- ◻ Hulten, Geoff. "[Building Intelligent Systems: A Guide to Machine Learning Engineering.](#)" Apress, 2018, Chapter 13 (Where Intelligence Lives).
- ◻ Daniel Smith. "[Exploring Development Patterns in Data Science.](#)" TheoryLane Blog Post. 2017.

Recommended reading: Rick Kazman, Paul Clements, and Len Bass. [Software architecture in practice](#). Addison-Wesley Professional, 2012, Chapter 1

LEARNING GOALS

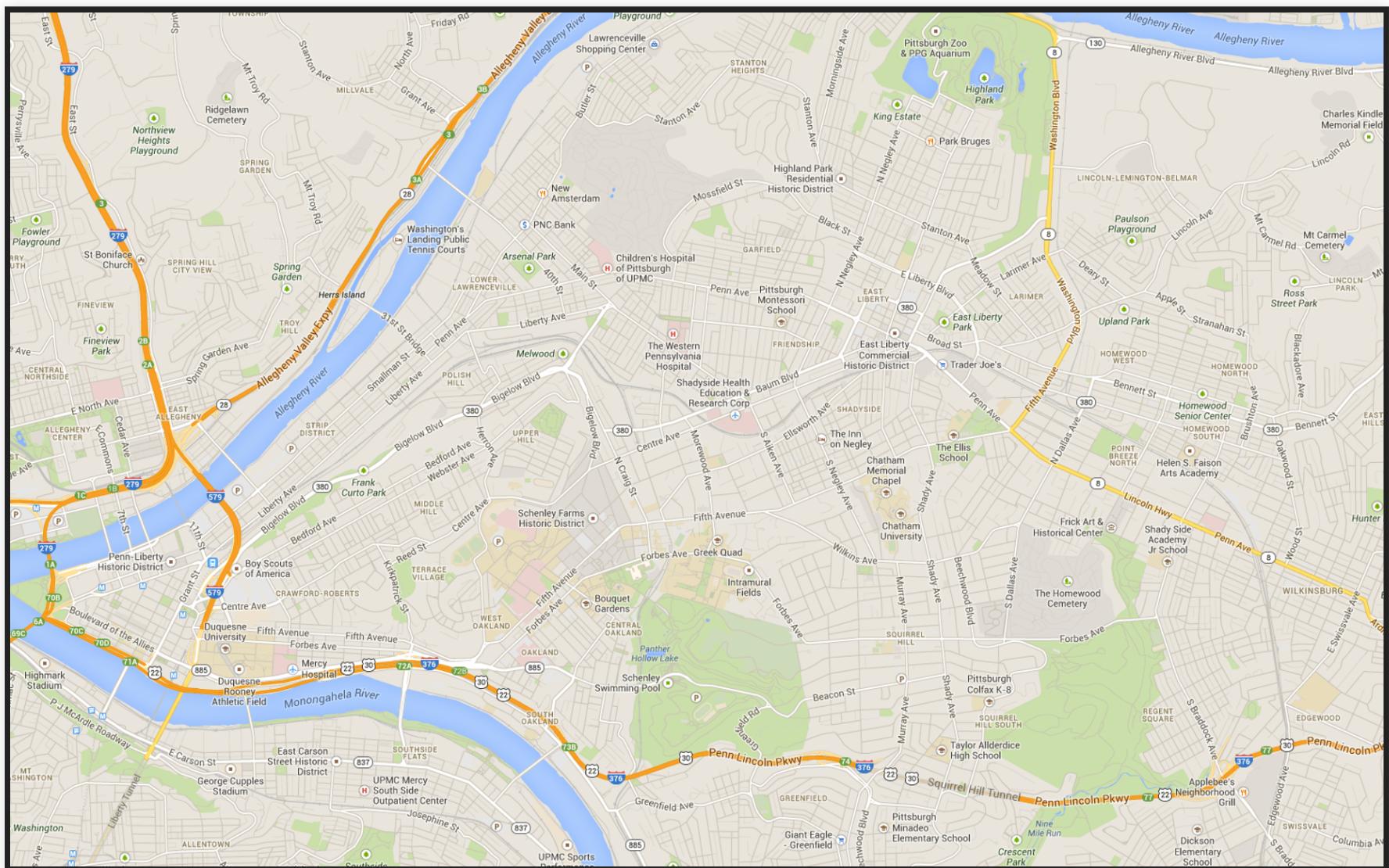
- Understand important quality considerations when using ML components
- Follow a design process to explicitly reason about alternative designs and their quality tradeoffs
- Gather data to make informed decisions about what ML technique to use and where and how to deploy it
- Create architectural models to reason about relevant characteristics
- Critique the decision of where an AI model lives (e.g., cloud vs edge vs hybrid), considering the relevant tradeoffs
- Deliberate how and when to update models and how to collect telemetry

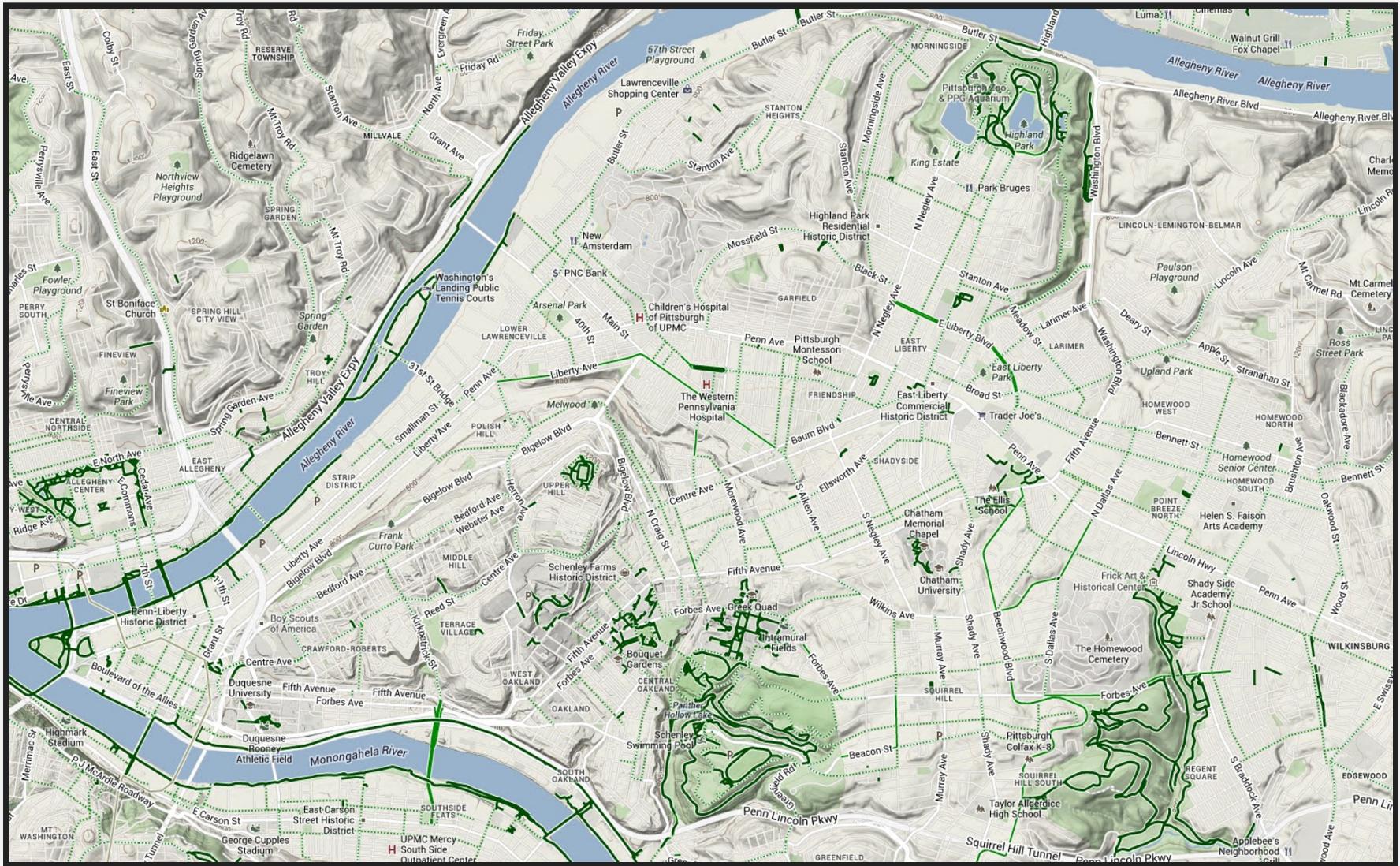
SOFTWARE ARCHITECTURE



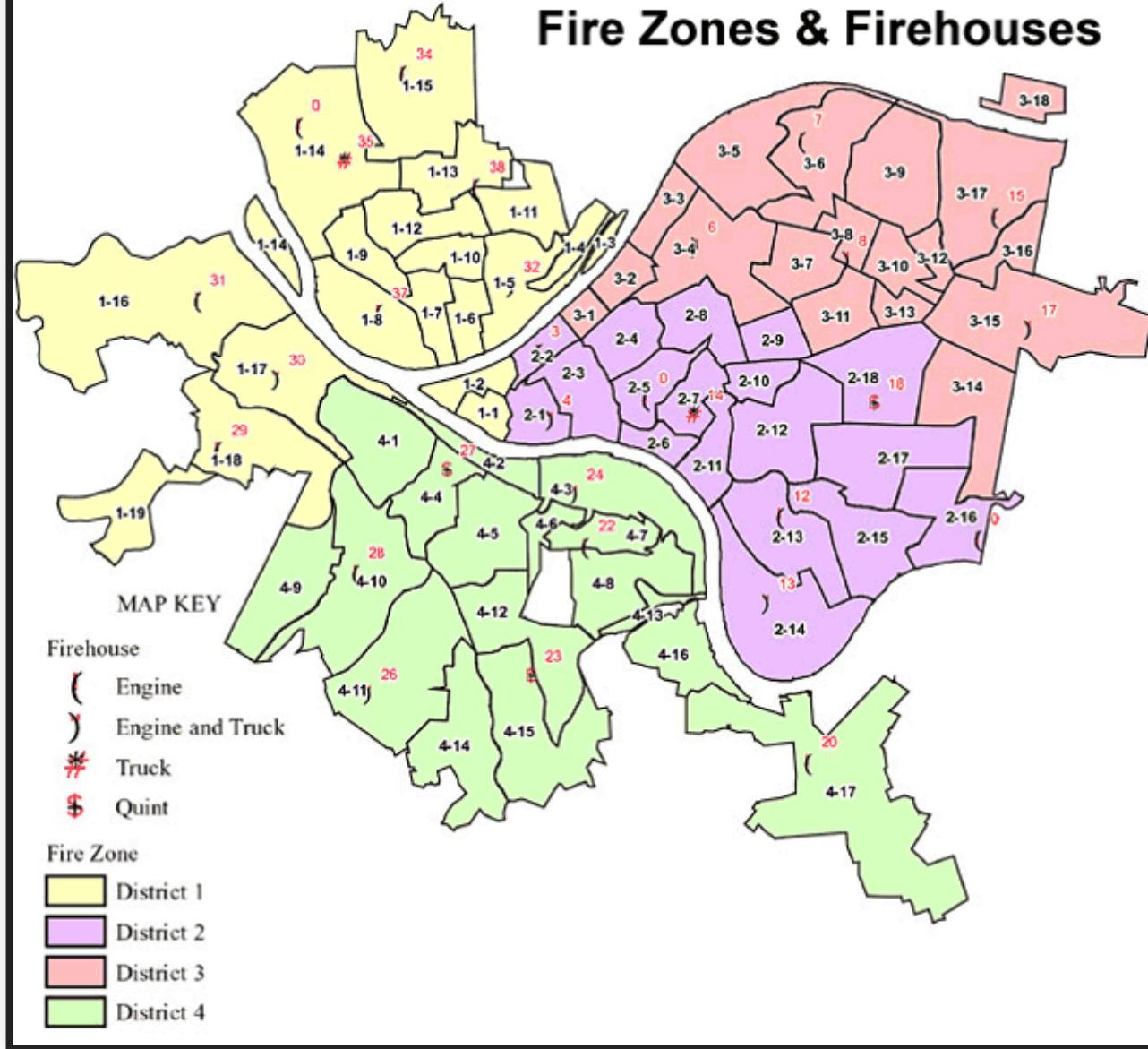
CASE STUDY: TWITTER



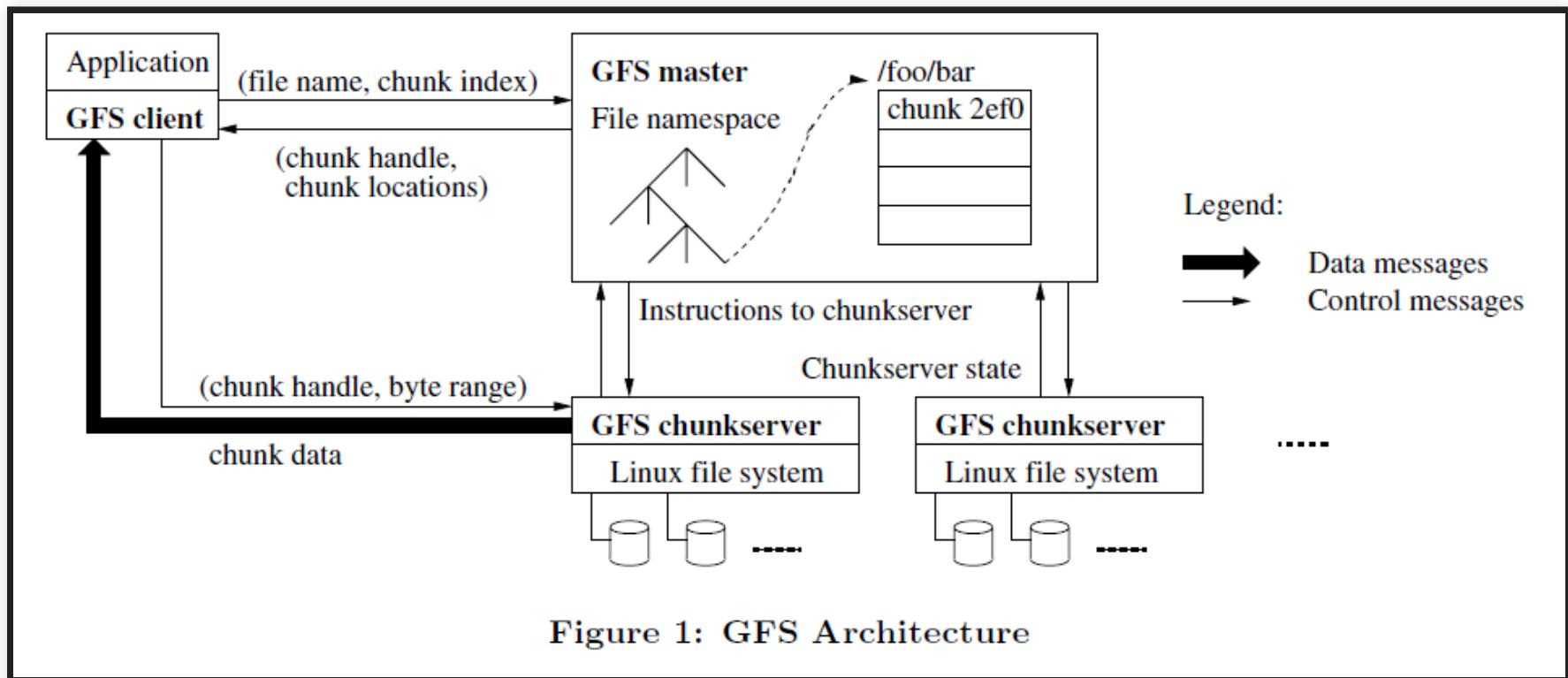




Fire Zones & Firehouses



WHAT CAN WE REASON ABOUT?



Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "[The Google file system.](#)" ACM SIGOPS operating systems review. Vol. 37. No. 5. ACM, 2003.

CASE STUDY: AUGMENTED REALITY TRANSLATION



WHERE SHOULD THE MODEL LIVE?

- Glasses
- Phone
- Cloud

What qualities are relevant for the decision?



WHEN WOULD ONE USE THE FOLLOWING DESIGNS?

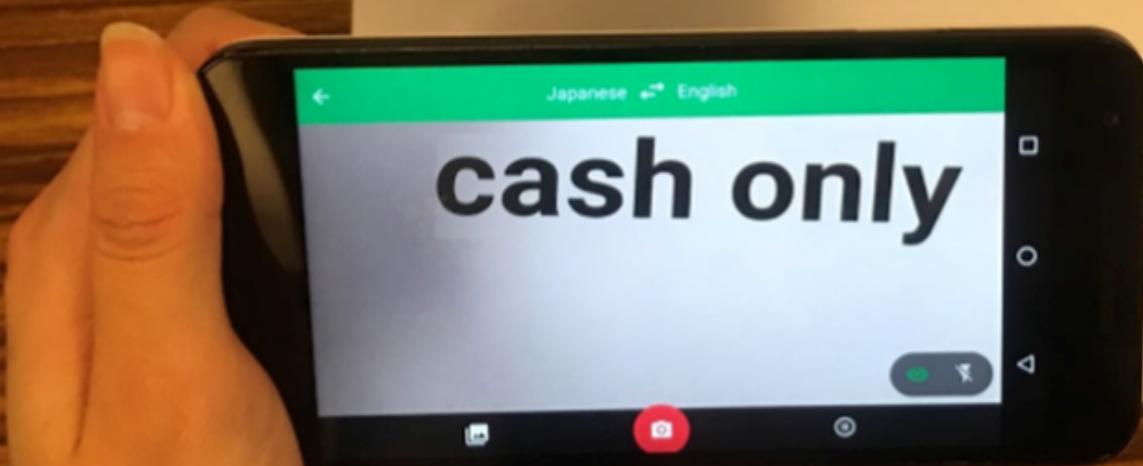
- Static intelligence in the product
- Client-side intelligence
- Server-centric intelligence
- Back-end cached intelligence
- Hybrid models

TELEMETRY TRADEOFFS

What data to collect? How much? When?

Estimate data volume and possible bottlenecks in system.

現金のみ



ARCHITECTURAL DECISION: UPDATING MODELS

- Design for change!
- Models are rarely static outside the lab
- Data drift, feedback loops, new features, new requirements
- When and how to update models?
- How to version? How to avoid mistakes?

ARCHITECTURES AND PATTERNS

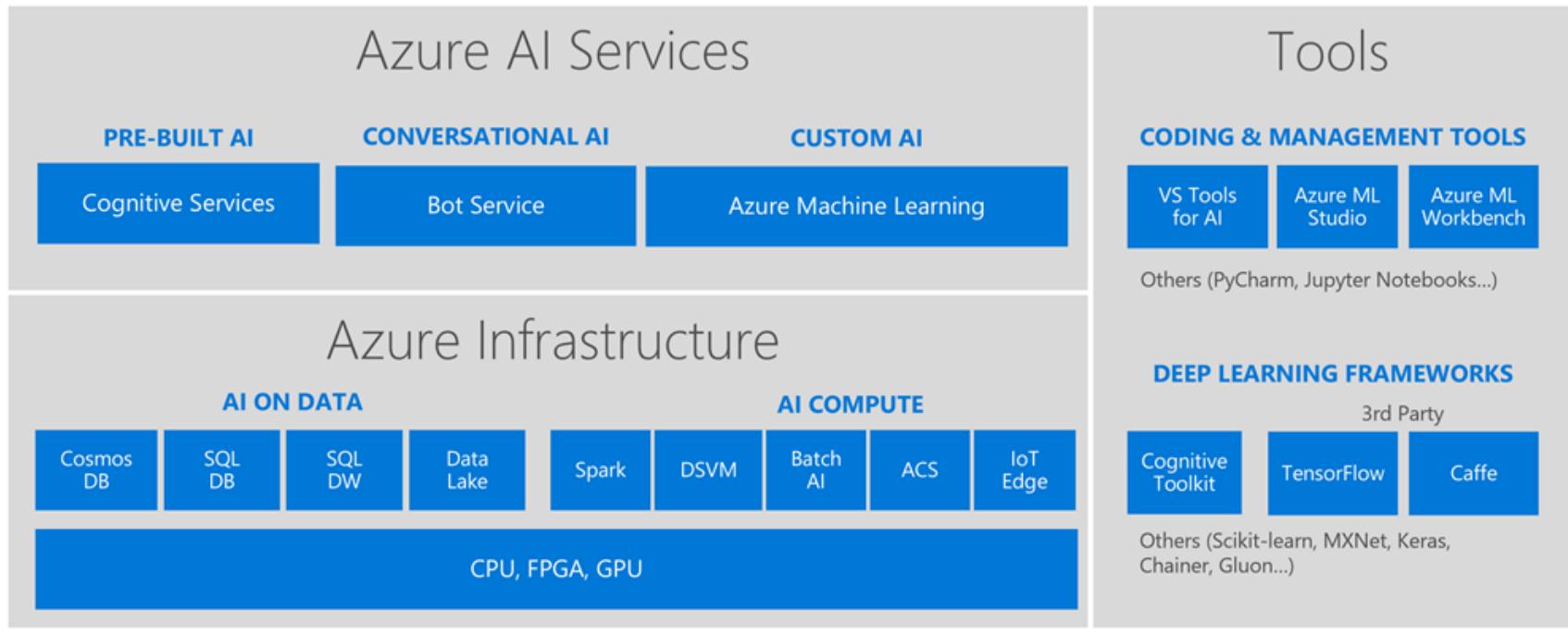
- The Big Ass Script Architecture
 - Decoupled multi-tiered architecture (data vs data analysis vs reporting; separate business logic from ML)
 - Microservice architecture (multiple learning and inference services)
 - Gateway Routing Architecture
-
- Pipelines
 - Data lake, lambda architecture
 - Reuse between training and serving pipelines
 - Continuous deployment, ML versioning, pipeline testing
-
- Daniel Smith. "[Exploring Development Patterns in Data Science](#)." TheoryLane Blog Post. 2017.
 - Washizaki, Hironori, Hiromu Uchida, Foutse Khomh, and Yann-Gaël Guéhéneuc. "[Machine Learning Architecture and Design Patterns](#)." Draft, 2019

READYMADE AI COMPONENTS IN THE CLOUD

- Data Infrastructure
 - Large scale data storage, databases, stream (MongoDB, Bigtable, Kafka)
- Data Processing
 - Massively parallel stream and batch processing (Sparks, Hadoop, ...)
 - Elastic containers, virtual machines (docker, AWS lambda, ...)
- AI Tools
 - Notebooks, IDEs, Visualization
 - Learning Libraries, Frameworks (tensorflow, torch, keras, ...)
- Models
 - Image, face, and speech recognition, translation
 - Chatbots, spell checking, text analytics
 - Recommendations, knowledge bases

The Microsoft AI platform

Cloud-powered AI for every developer



HOMEWORK 4: ARCHITECTURE

Deployment alternatives and telemetry for smart dashcam

DATA QUALITY AND DATA PROGRAMMING

"Data cleaning and repairing account for about 60% of the work of data scientists."

Eunsuk Kang

Required reading:

- Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F. and Grafberger, A., 2018. [Automating large-scale data quality verification](#). Proceedings of the VLDB Endowment, 11(12), pp.1781-1794.
- Nick Hynes, D. Sculley, Michael Terry. "[The Data Linter: Lightweight Automated Sanity Checking for ML Data Sets](#)." NIPS Workshop on ML Systems (2017)

LEARNING GOALS

- Design and implement automated quality assurance steps that check data schema conformance and distributions
- Devise thresholds for detecting data drift and schema violations
- Describe common data cleaning steps and their purpose and risks
- Evaluate the robustness of AI components with regard to noisy or incorrect data
- Understanding the better models vs more data tradeoffs
- Programmatically collect, manage, and enhance training data

CASE STUDY: INVENTORY MANAGEMENT

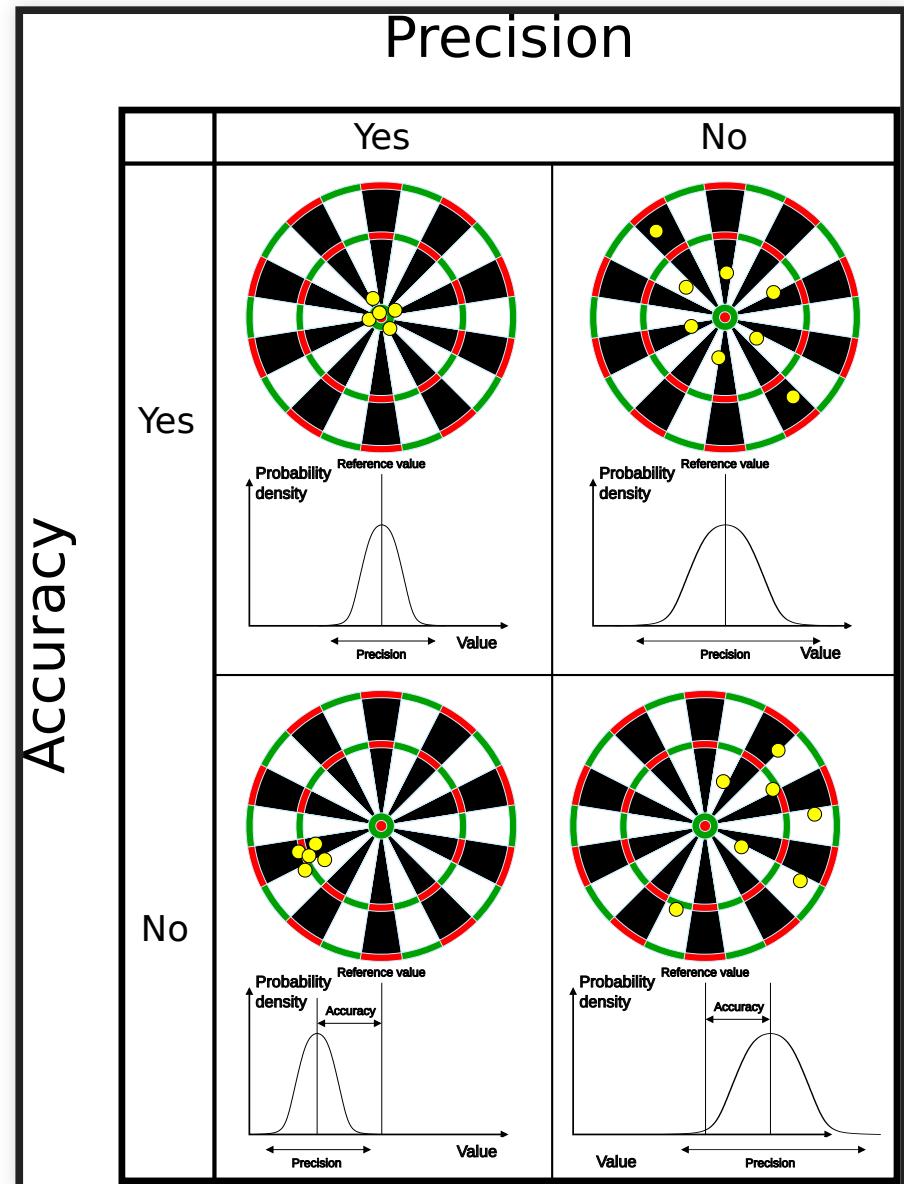


WHAT MAKES GOOD QUALITY DATA?

- Accuracy
 - The data was recorded correctly.
- Completeness
 - All relevant data was recorded.
- Uniqueness
 - The entries are recorded once.
- Consistency
 - The data agrees with itself.
- Timeliness
 - The data is kept up to date.

ACCURACY VS PRECISION

- Accuracy: Reported values (on average) represent real value
- Precision: Repeated measurements yield the same result
- Accurate, but imprecise: Average over multiple measurements
- Inaccurate, but precise: Systematic measurement problem, misleading

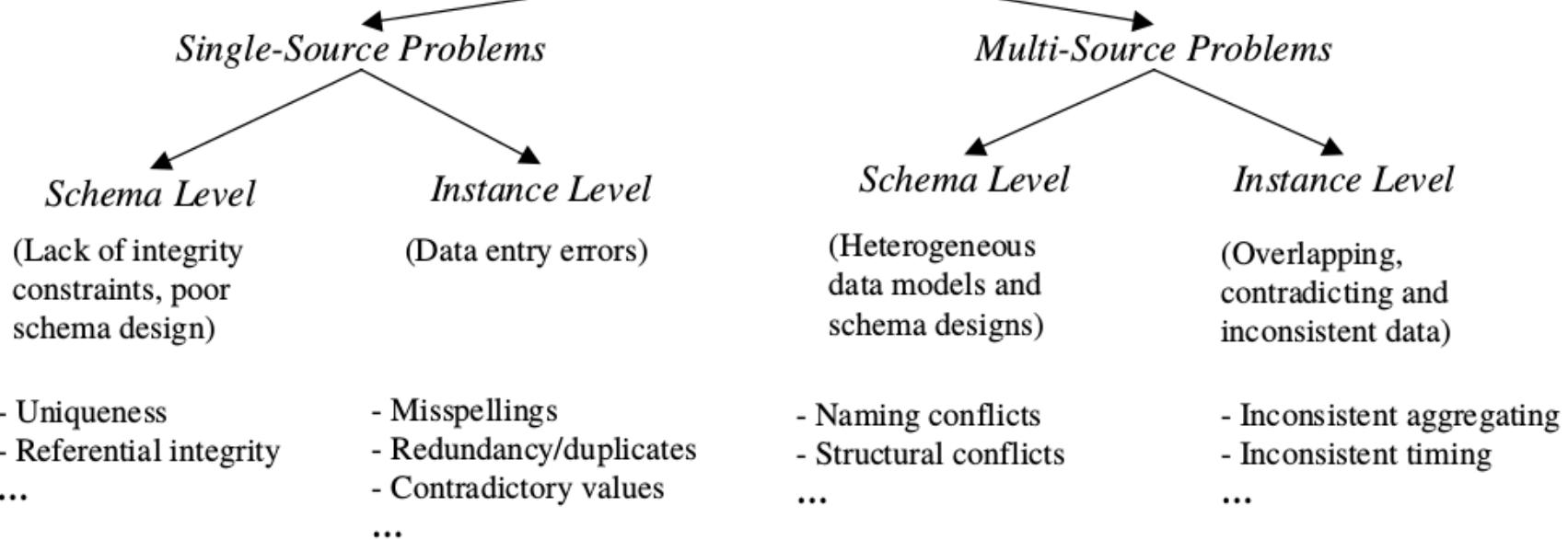


EXPLORATORY DATA ANALYSIS IN DATA SCIENCE

- Before learning, understand the data
- Understand types, ranges, distributions
- Important for understanding data and assessing quality
- Plot data distributions for features
 - Visualizations in a notebook
 - Boxplots, histograms, density plots, scatter plots, ...
- Explore outliers
- Look for correlations and dependencies
 - Association rule mining
 - Principal component analysis

Examples: <https://rpubs.com/ablythe/520912> and
<https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15>

Data Quality Problems



Source: Rahm, Erhard, and Hong Hai Do. [Data cleaning: Problems and current approaches](#). IEEE Data Eng. Bull. 23.4 (2000): 3-13.

DIRTY DATA: EXAMPLE

TABLE: CUSTOMER

ID	Name	Birthday	Age	Sex	Phone	ZIP
3456	Ford, Harrison	18.2.76	43	M	9999999999	15232
3456	Mark Hamil	33.8.81	43	M	6173128718	17121
3457	Kim Kardashian	11.10.56	63	M	4159102371	94016

TABLE: ADDRESS

ZIP	City	State
15232	Pittsburgh	PA
94016	Sam Francisco	CA
73301	Austin	Texas

Problems with the data?

DATA CLEANING OVERVIEW

- Data analysis / Error detection
 - Error types: e.g. schema constraints, referential integrity, duplication
 - Single-source vs multi-source problems
 - Detection in input data vs detection in later stages (more context)
- Error repair
 - Repair data vs repair rules, one at a time or holistic
 - Data transformation or mapping
 - Automated vs human guided

SCHEMA IN RELATIONAL DATABASES

```
CREATE TABLE employees (
    emp_no      INT            NOT NULL,
    birth_date   DATE           NOT NULL,
    name        VARCHAR(30)     NOT NULL,
    PRIMARY KEY (emp_no));
CREATE TABLE departments (
    dept_no     CHAR(4)         NOT NULL,
    dept_name   VARCHAR(40)     NOT NULL,
    PRIMARY KEY (dept_no), UNIQUE KEY (dept_name));
CREATE TABLE dept_manager (
    dept_no     CHAR(4)         NOT NULL,
    emp_no      INT            NOT NULL,
    FOREIGN KEY (emp_no) REFERENCES employees (emp_no),
    FOREIGN KEY (dept_no) REFERENCES departments (dept_no),
    PRIMARY KEY (emp_no,dept_no));
```

EXAMPLE: APACHE AVRO

```
{  "type": "record",
  "namespace": "com.example",
  "name": "Customer",
  "fields": [
    {
      "name": "first_name",
      "type": "string",
      "doc": "First Name of Customer"
    },
    {
      "name": "age",
      "type": "int",
      "doc": "Age at the time of registration"
    }
  ]
}
```

DETECTING INCONSISTENCIES

	DBAName	AKAName	Address	City	State	Zip
t1	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60608
t2	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t3	John Veliotis Sr.	Johnnyo's	3465 S Morgan ST	Chicago	IL	60609
t4	Johnnyo's	Johnnyo's	3465 S Morgan ST	Cicago	IL	60608

Conflicts

Does not obey data distribution

Conflict

Conflicts

Does not obey data distribution

Conflict

Image source: Theo Rekatsinas, Ihab Ilyas, and Chris Ré, “[HoloClean - Weakly Supervised Data Repairing](#).” Blog, 2017.

DATA LINTER AT GOOGLE

- Miscoding
 - Number, date, time as string
 - Enum as real
 - Tokenizable string (long strings, all unique)
 - Zip code as number
- Outliers and scaling
 - Unnormalized feature (varies widely)
 - Tailed distributions
 - Uncommon sign
- Packaging
 - Duplicate rows
 - Empty/missing data

Further readings: Hynes, Nick, D. Sculley, and Michael Terry. [The data linter: Lightweight, automated sanity checking for ML data sets](#). NIPS MLSys Workshop. 2017.

DRIFT & MODEL DECAY

in all cases, models are less effective over time

- Concept drift
 - properties to predict change over time (e.g., what is credit card fraud)
 - over time: different expected outputs for same inputs
 - model has not learned the relevant concepts
- Data drift
 - characteristics of input data changes (e.g., customers with face masks)
 - input data differs from training data
 - over time: predictions less confident, further from training data
- Upstream data changes
 - external changes in data pipeline (e.g., format changes in weather service)
 - model interprets input data incorrectly
 - over time: abrupt changes due to faulty inputs

WATCH FOR DEGRADATION IN PREDICTION ACCURACY

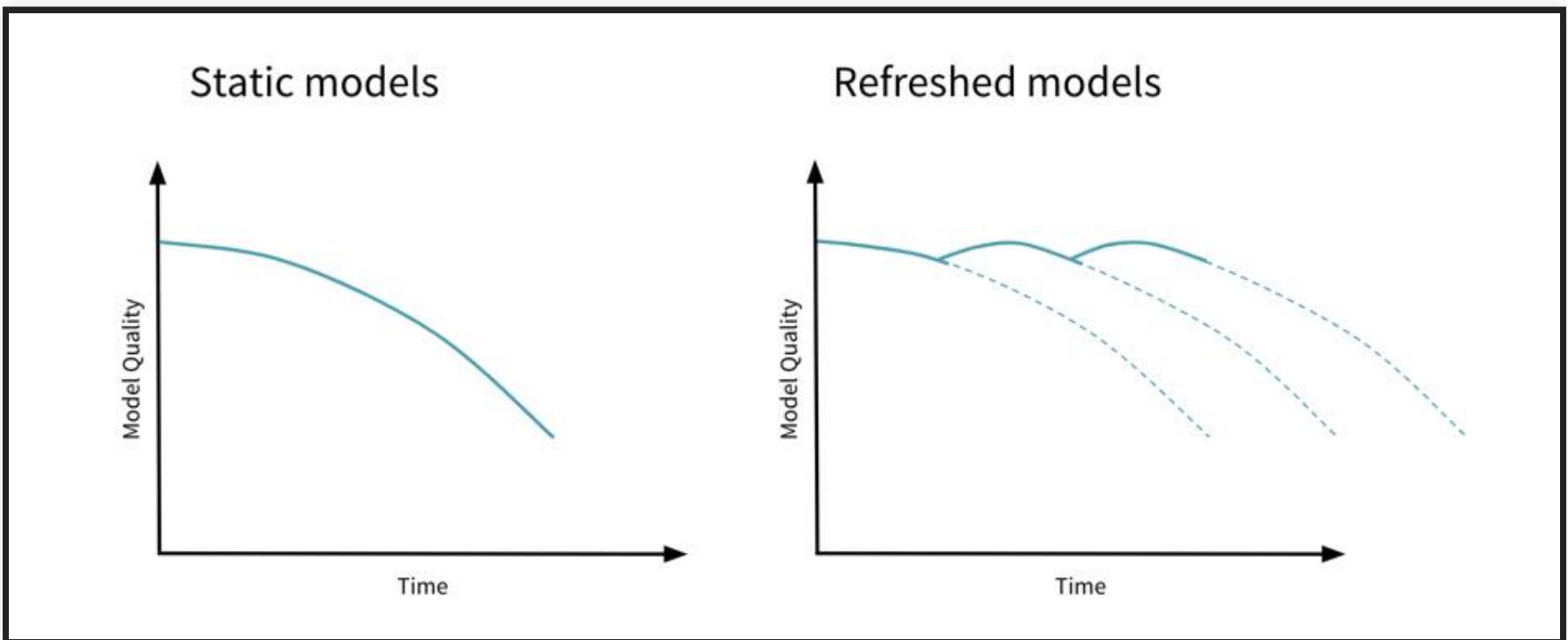
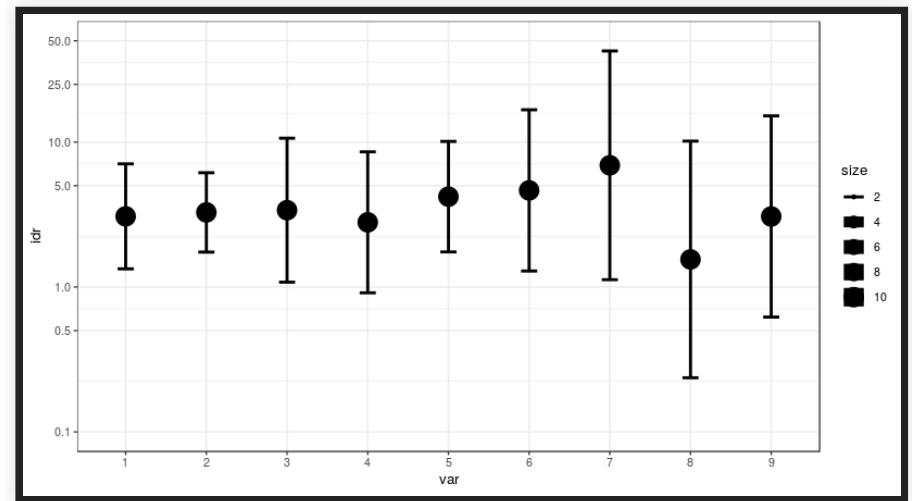
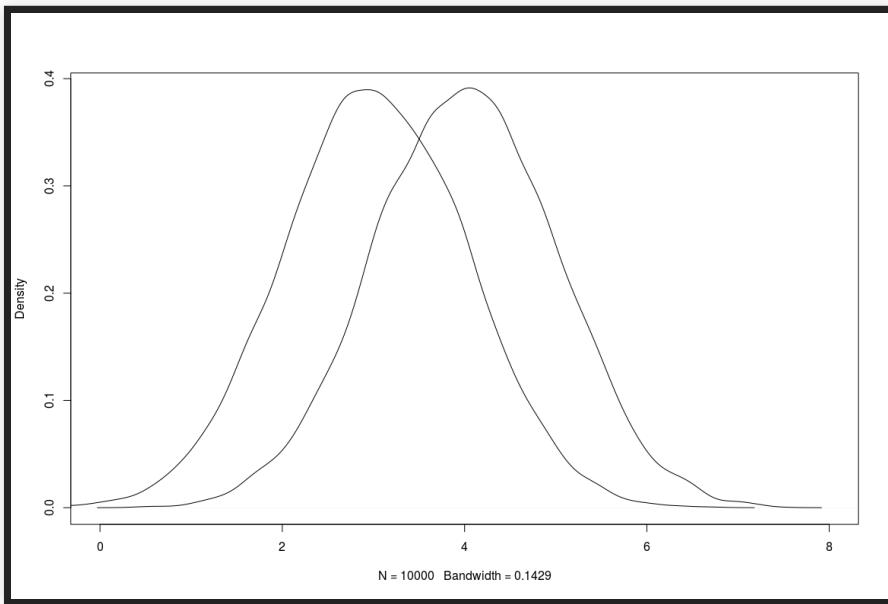


Image source: Joel Thomas and Clemens Mewald. [Productionizing Machine Learning: From Deployment to Drift Detection](#). Databricks Blog, 2019

DETECTING DATA DRIFT

- Compare distributions over time (e.g., t-test)
- Detect both sudden jumps and gradual changes
- Distributions can be manually specified or learned (see invariant detection)



INFRASTRUCTURE QUALITY, DEPLOYMENT, AND OPERATIONS

Christian Kaestner

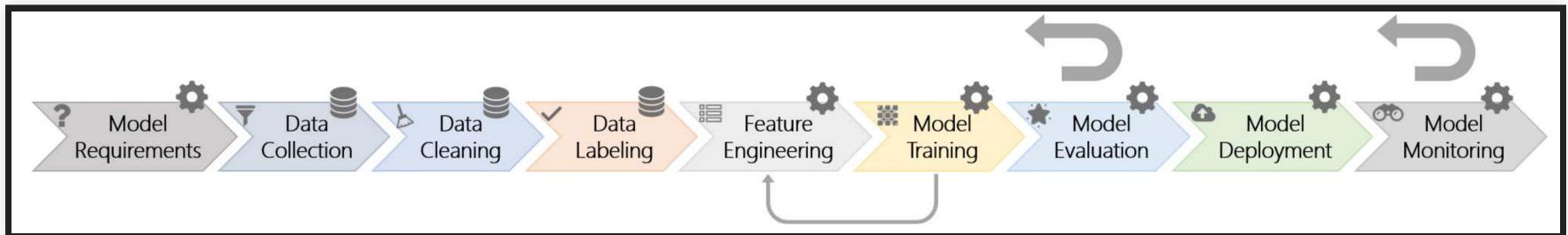
Required reading: Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley. [The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction](#). Proceedings of IEEE Big Data (2017)

Recommended readings: Larysa Visengeriyeva. [Machine Learning Operations - A Reading List](#), InnoQ 2020

LEARNING GOALS

- Implement and automate tests for all parts of the ML pipeline
- Understand testing opportunities beyond functional correctness
- Automate test execution with continuous integration
- Deploy a service for models using container infrastructure
- Automate common configuration management tasks
- Devise a monitoring strategy and suggest suitable components for implementing it
- Diagnose common operations problems
- Understand the typical concerns and concepts of MLOps

POSSIBLE MISTAKES IN ML PIPELINES



Danger of "silent" mistakes in many phases



FROM MANUAL TESTING TO CONTINUOUS INTEGRATION



The screenshot shows a web browser window displaying the Travis CI interface for a repository named "wyvernlang/wyvern". The build number is #17. The status is "passing". The build summary indicates 17 passed tests, a commit from fd7be1c, and a compare from 0e2af1f. The build duration was 16 seconds, and it ran for 16 seconds 3 days ago. A note at the bottom says "This job ran on our legacy infrastructure. Please read our docs on how to upgrade." Below the summary, the build log is displayed in a monospaced font, showing the command-line steps taken during the build process.

```
1 Using worker: worker-linux-027f0490-1.bb.travis-ci.org:travis-linux-2
2
3 Build system information
67
68 $ git clone --depth=50 --branch=SimpleWyvern-devel
git:clone
69 $ jdk_switcher use oraclejdk8
git:checkout
70 Switching to Oracle JDK8 [java-8-oracle], JAVA_HOME will be set to /usr/lib/jvm/java-8-oracle
71 $ java -Xmx32m -version
72 java version "1.8.0_31"
73 Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
74 Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
75 $ java -J-Xmx32m -version
76 javac 1.8.0_31
77 $ cd tools
78
79 The command "cd tools" exited with 0.
80 $ ant test
81 Buildfile: /home/travis/build/wyvernlang/wyvern/tools/build.xml
82
83 copper-compose-compile:
84     [mkdir] Created dir: /home/travis/build/wyvernlang/wyvern/tools/copper-composer/bin
85     [javac] /home/travis/build/wyvernlang/wyvern/tools/build.xml:18: warning: 'includeantruntime'
86 was not set, defaulting to build.sysclasspath=last; set to false for repeatable builds
87
88
89
90
91
92
93
94
```

EXAMPLE: MOCKING A DATACLEANER OBJECT

```
DataTable getData(KafkaStream stream, DataCleaner cleaner) { ...  
  
@Test void test() {  
    DataCleaner dummyCleaner = new DataCleaner() {  
        int counter = 0;  
        boolean isValid(String row) {  
            counter++;  
            return counter!=3;  
        }  
        ...  
    }  
    DataTable output = getData(testStream, dummyCleaner);  
    assert(output.length==9)  
}
```

Mocking frameworks provide infrastructure for expressing such tests compactly.

TESTING FOR ROBUSTNESS

manipulating the (controlled) environment: injecting errors into backend to test error handling

```
DataTable getData(Stream stream, DataCleaner cleaner) { ... }

@Test void test() {
    Stream testStream = new Stream() {
        ...
        public String getNext() {
            if (++idx == 3) throw new IOException();
            return data[++idx];
        }
    }
    DataTable output = retry(getData(testStream, ...));
    assert(output.length==10)
}
```

Packages

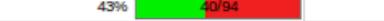
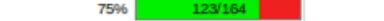
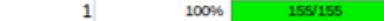
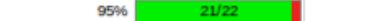
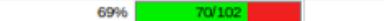
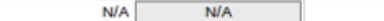
All
[net.sourceforge.cobertura.ant](#)
[net.sourceforge.cobertura.check](#)
[net.sourceforge.cobertura.coveragedata](#)
[net.sourceforge.cobertura.instrument](#)
[net.sourceforge.cobertura.merge](#)
[net.sourceforge.cobertura.reporting](#)
[net.sourceforge.cobertura.reporting.html](#)
[net.sourceforge.cobertura.reporting.html](#)
[net.sourceforge.cobertura.reporting.xml](#)
[net.sourceforge.cobertura.util](#)

All Packages

Classes

[AntUtil](#) (88%)
[Archive](#) (100%)
[ArchiveUtil](#) (80%)
[BranchCoverageData](#) (N/A)
[CheckTask](#) (0%)
[ClassData](#) (N/A)
[ClassInstrumenter](#) (94%)
[ClassPattern](#) (100%)
[CoberturaFile](#) (73%)
[CommandLineBuilder](#) (96%)
[CommonMatchingTask](#) (88%)
[ComplexityCalculator](#) (100%)
[ConfigurationUtil](#) (50%)
[CopyFiles](#) (87%)
[CoverageData](#) (N/A)
[CoverageDataContainer](#) (N/A)
[CoverageDataFileHandler](#) (N/A)
[CoverageRate](#) (0%)
[ExcludeClasses](#) (100%)
[FileFinder](#) (96%)
[FileLocker](#) (0%)
[FirstPassMethodInstrumenter](#) (100%)
[HTMLReport](#) (94%)
[HasBeenInstrumented](#) (N/A)
[Header](#) (80%)
[IOUtil](#) (62%)
[Ignore](#) (100%)
[IgnoreBranches](#) (0%)

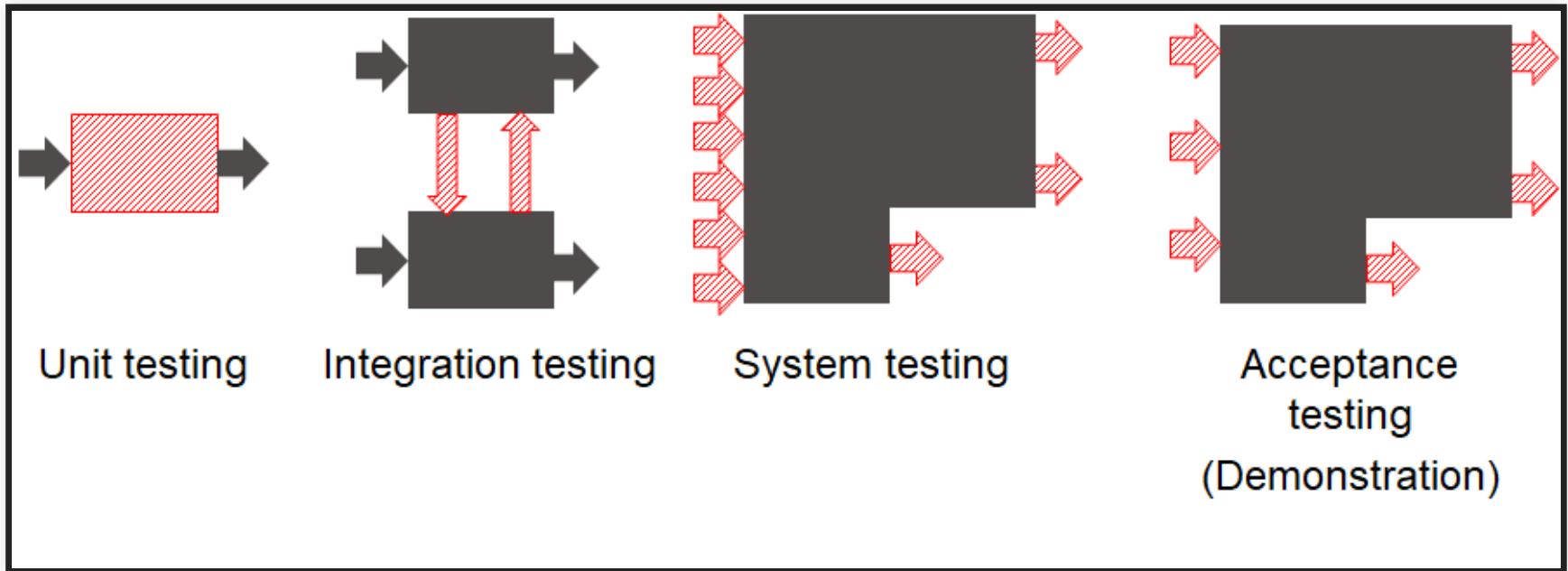
Coverage Report - All Packages

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	55	75%  1625/2179	64%  472/738	2.319
net.sourceforge.cobertura.ant	11	52%  170/330	43%  40/94	1.848
net.sourceforge.cobertura.check	3	0%  0/150	0%  0/76	2.429
net.sourceforge.cobertura.coveragedata	13	N/A  N/A	N/A  N/A	2.277
net.sourceforge.cobertura.instrument	10	90%  460/510	75%  123/164	1.854
net.sourceforge.cobertura.merge	1	86%  30/35	88%  14/16	5.5
net.sourceforge.cobertura.reporting	3	87%  116/134	80%  43/54	2.882
net.sourceforge.cobertura.reporting.html	4	91%  475/523	77%  156/202	4.444
net.sourceforge.cobertura.reporting.html.files	1	87%  39/45	62%  5/8	4.5
net.sourceforge.cobertura.reporting.xml	1	100%  155/155	95%  21/22	1.524
net.sourceforge.cobertura.util	9	60%  175/291	69%  70/102	2.892
someotherpackage	1	83%  5/6	N/A  N/A	1.2

Report generated by [Cobertura](#) 1.9 on 6/9/07 12:37 AM.



INTEGRATION AND SYSTEM TESTS



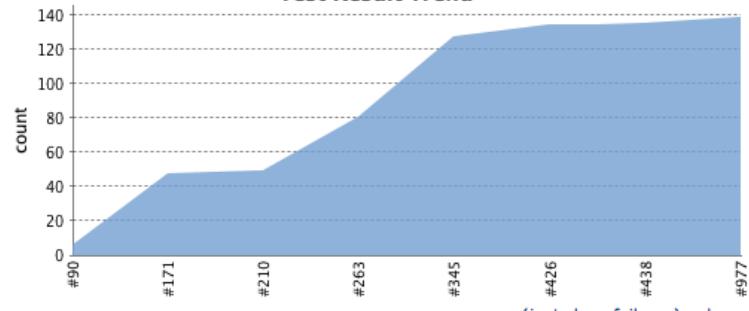
[Back to Dashboard](#)[Status](#)[Changes](#)[Workspace](#)[Build Now](#)[Delete Project](#)[Configure](#)[Set Next Build Number](#)[Duplicate Code](#)[Coverage Report](#)[SLOCCount](#)[Git Polling Log](#)

Project Stop-tabac dev

CI build

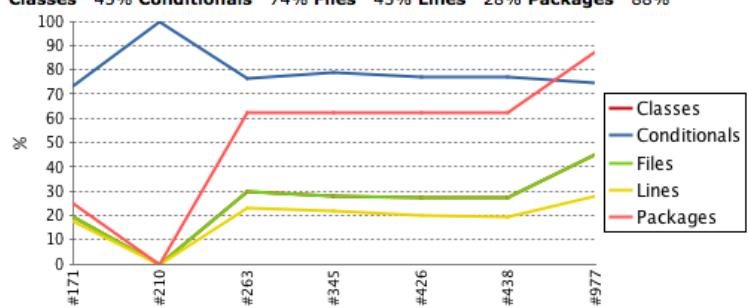
[Coverage Report](#)[Workspace](#)[Recent Changes](#)[Latest Test Result \(no failures\)](#)[Edit description](#)[Disable Project](#)

Test Result Trend

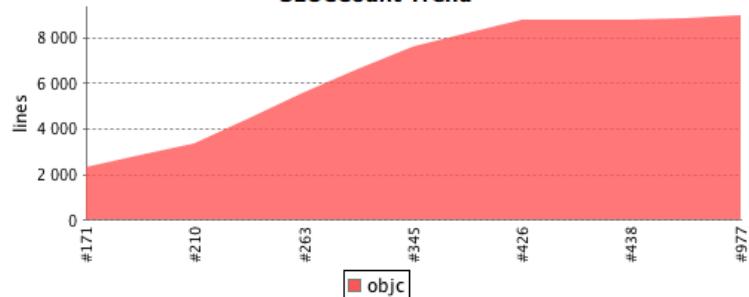
[\(just show failures\) enlarge](#)

Code Coverage

Classes 45% Conditionals 74% Files 45% Lines 28% Packages 88%



SLOCCount Trend



Build History (trend)

	#977	Aug 27, 2012 4:37:27 PM
	#438	Jun 28, 2012 8:47:42 AM
	#426	Jun 26, 2012 1:39:39 PM
	#345	Jun 19, 2012 9:02:20 AM
	#263	Jun 6, 2012 9:14:42 PM
	#210	May 31, 2012 8:42:29 AM
	#171	May 23, 2012 9:58:18 PM
	#90	May 15, 2012 11:49:41 AM

[RSS for all](#)[RSS for failures](#)

Source: <https://blog.octo.com/en/jenkins-quality-dashboard-ios-development/>

TEST MONITORING IN PRODUCTION

- Like fire drills (manual tests may be okay!)
- Manual tests in production, repeat regularly
- Actually take down service or trigger wrong signal to monitor

CHAOS TESTING



<http://principlesofchaos.org>

CASE STUDY: SMART PHONE COVID-19 DETECTION



(from midterm; assume cloud or hybrid deployment)

DATA TESTS

1. Feature expectations are captured in a schema.
2. All features are beneficial.
3. No feature's cost is too much.
4. Features adhere to meta-level requirements.
5. The data pipeline has appropriate privacy controls.
6. New features can be added quickly.
7. All input feature code is tested.

Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley. [The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction](#). Proceedings of IEEE Big Data (2017)

TESTS FOR MODEL DEVELOPMENT

1. Model specs are reviewed and submitted.
2. Offline and online metrics correlate.
3. All hyperparameters have been tuned.
4. The impact of model staleness is known.
5. A simpler model is not better.
6. Model quality is sufficient on important data slices.
7. The model is tested for considerations of inclusion.

Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley. [The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction](#). Proceedings of IEEE Big Data (2017)

ML INFRASTRUCTURE TESTS

1. Training is reproducible.
2. Model specs are unit tested.
3. The ML pipeline is Integration tested.
4. Model quality is validated before serving.
5. The model is debuggable.
6. Models are canaried before serving.
7. Serving models can be rolled back.

Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley. [The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction](#). Proceedings of IEEE Big Data (2017)

MONITORING TESTS

1. Dependency changes result in notification.
2. Data invariants hold for inputs.
3. Training and serving are not skewed.
4. Models are not too stale.
5. Models are numerically stable.
6. Computing performance has not regressed.
7. Prediction quality has not regressed.

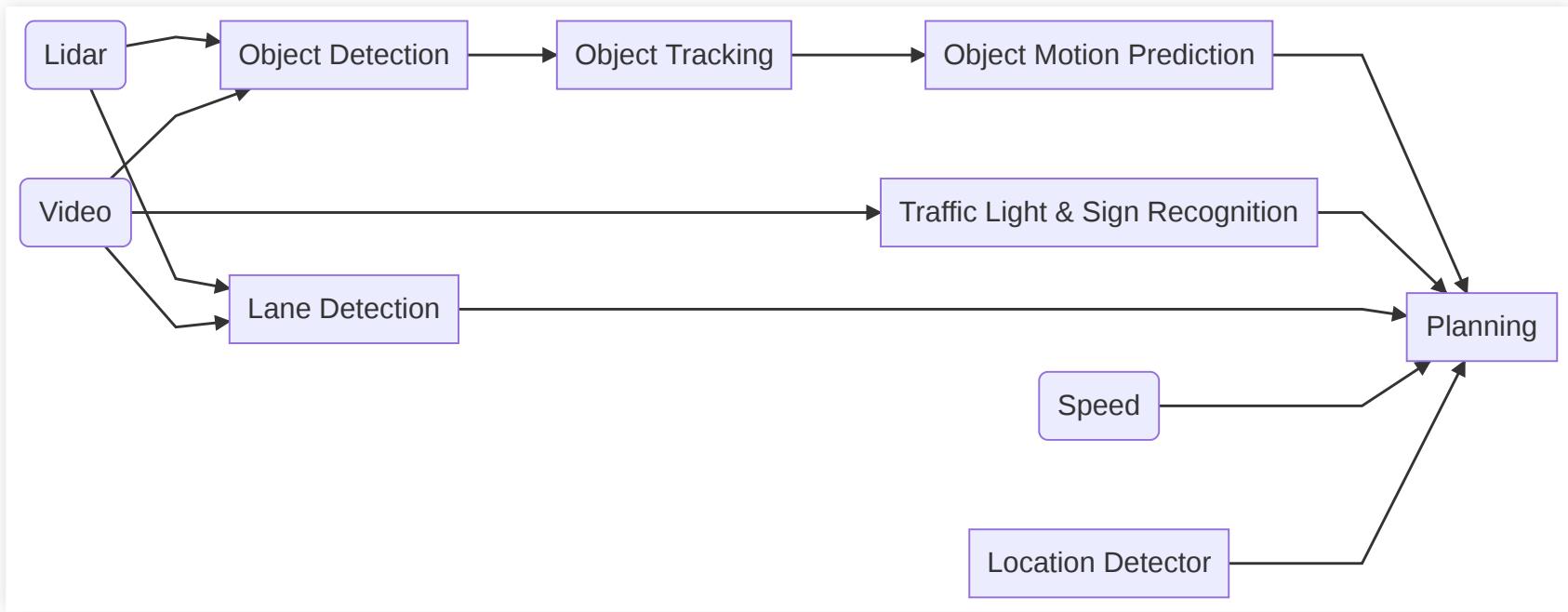
Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley. [The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction](#). Proceedings of IEEE Big Data (2017)

FEATURE INTERACTION EXAMPLES



ML MODELS FOR FEATURE EXTRACTION

self driving car



Example: Zong, W., Zhang, C., Wang, Z., Zhu, J., & Chen, Q. (2018). [Architecture design and implementation of an autonomous vehicle](#). IEEE access, 6, 21956-21970.

DEV VS. OPS



DEVELOPERS

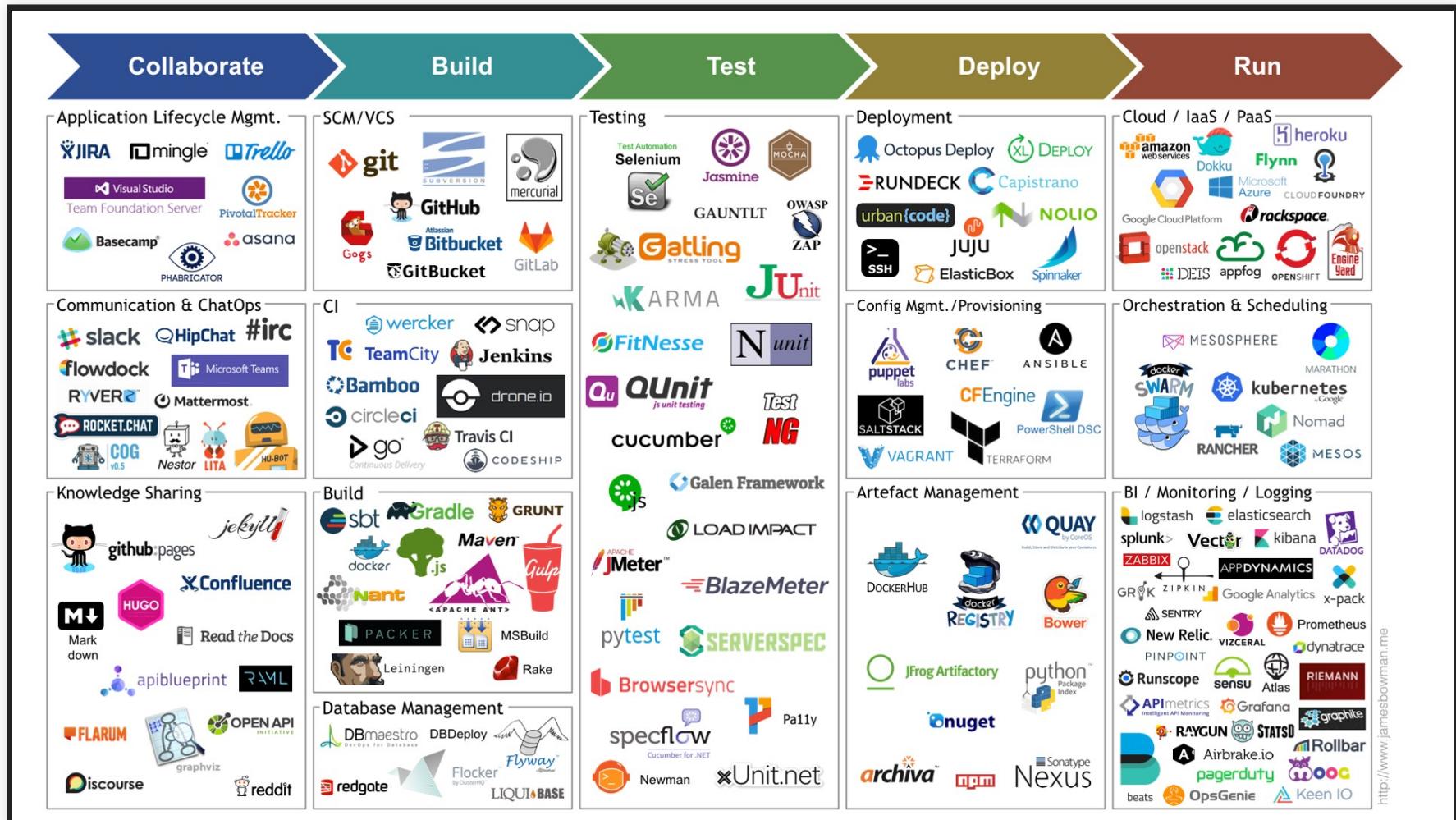
- Coding
- Testing, static analysis, reviews
- Continuous integration
- Bug tracking
- Running local tests and scalability experiments
- ...

OPERATIONS

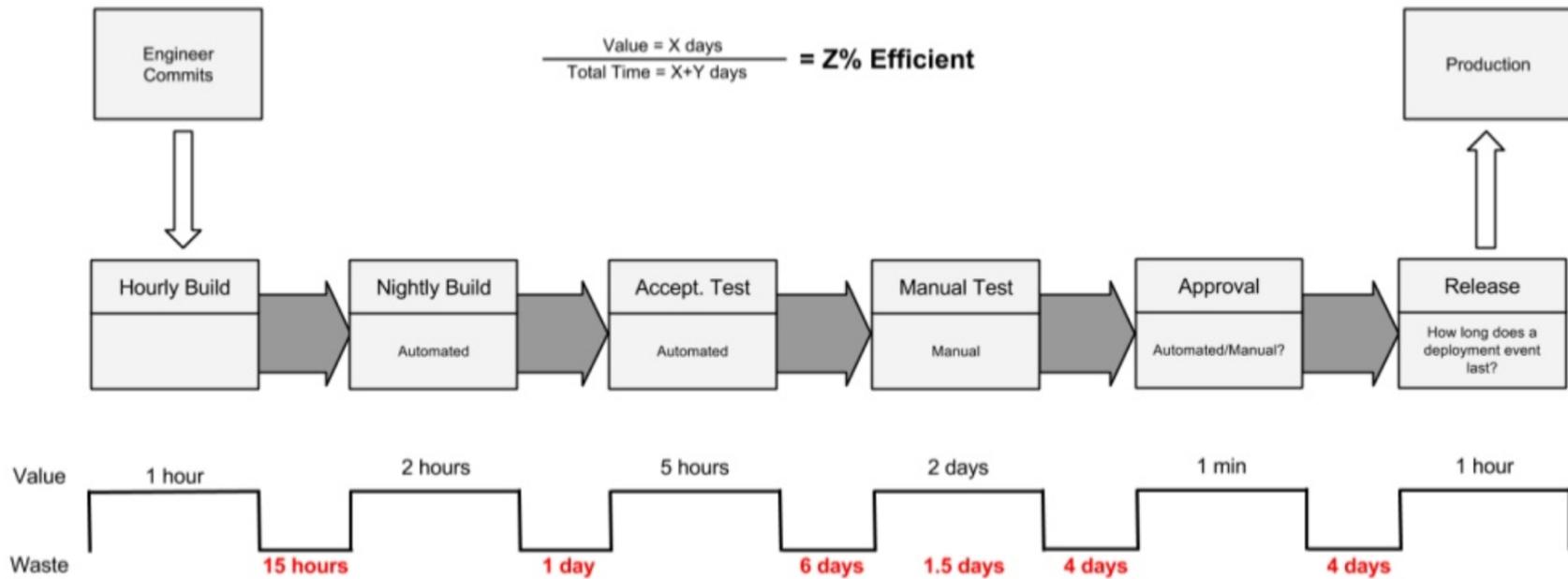
- Allocating hardware resources
- Managing OS updates
- Monitoring performance
- Monitoring crashes
- Managing load spikes, ...
- Tuning database performance
- Running distributed at scale
- Rolling back releases
- ...

QA responsibilities in both roles

HEAVY TOOLING AND AUTOMATION

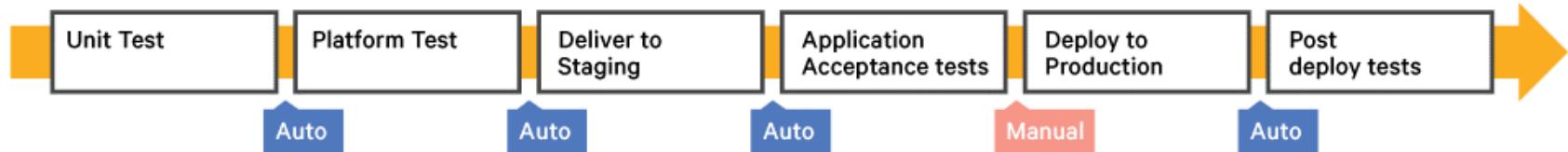


<http://www.jamesbowman.me>

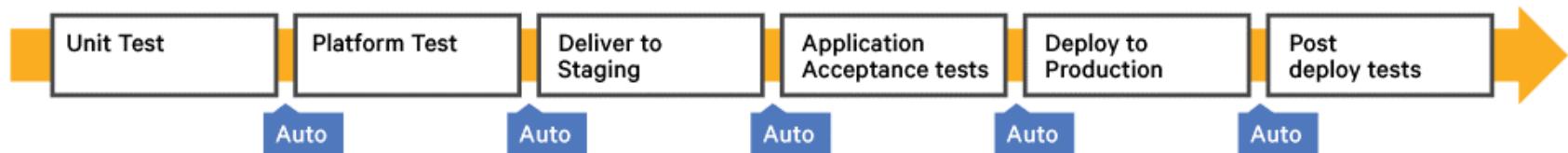


Source: <https://www.slideshare.net/jmcgarr/continuous-delivery-at-netflix-and-beyond>

Continuous Delivery



Continuous Deployment



DOCKER EXAMPLE

```
FROM ubuntu:latest
MAINTAINER ...
RUN apt-get update -y
RUN apt-get install -y python-pip python-dev build-essential
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
ENTRYPOINT ["python"]
CMD ["app.py"]
```

Source: <http://containertutorials.com/docker-compose/flask-simple-app.html>

ANSIBLE EXAMPLES

- Software provisioning, configuration management, and application-deployment tool
- Apply scripts to many servers

```
[webservers]
web1.company.org
web2.company.org
web3.company.org
```

```
[dbservers]
db1.company.org
db2.company.org
```

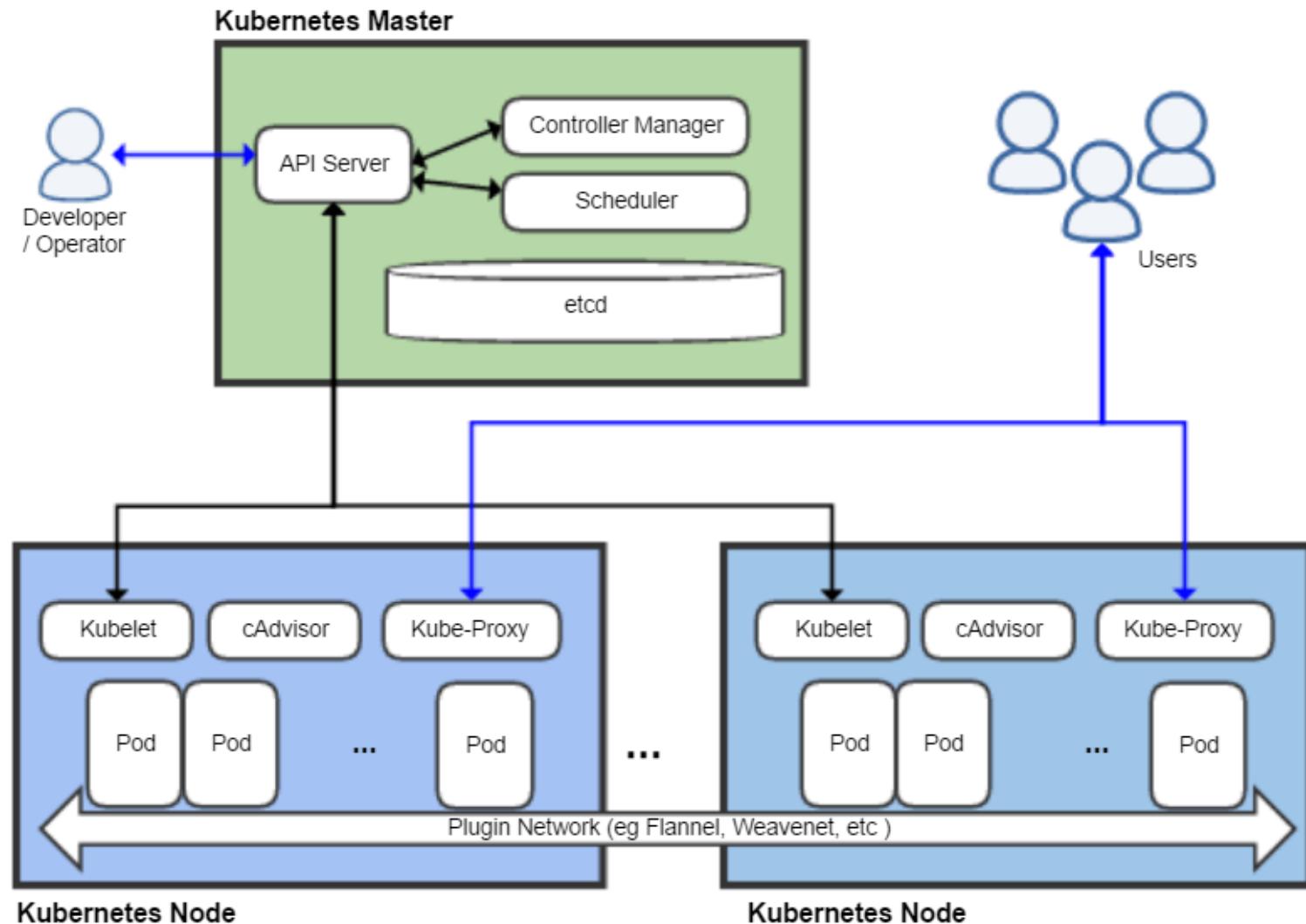
```
[replication_servers]
...
```

```
# This role deploys the mongod processes and
- name: create data directory for mongodb
  file: path={{ mongodb_datadir_prefix }}/{{ item }}
  delegate_to: '{{ item }}'
  with_items: groups.replication_servers

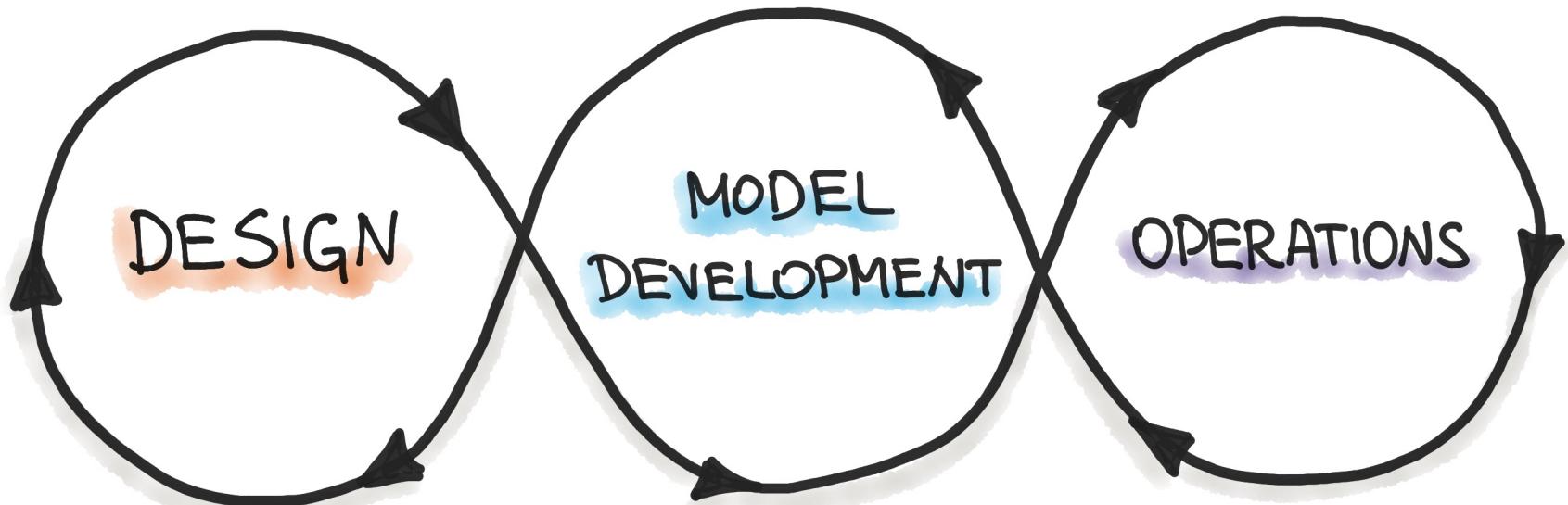
- name: create log directory for mongodb
  file: path=/var/log/mongo state=directory owner=mongo group=adm mode=0755
  delegate_to: '{{ item }}'
  with_items: groups.replication_servers

- name: Create the mongodb startup file
  template: src=mongod.j2 dest=/etc/init.d/mongod
  delegate_to: '{{ item }}'
  with_items: groups.replication_servers

- name: Create the mongodb configuration file
  template: src=mongod.conf.j2 dest=/etc/mongod.conf
  delegate_to: '{{ item }}'
  with_items: groups.replication_servers
```

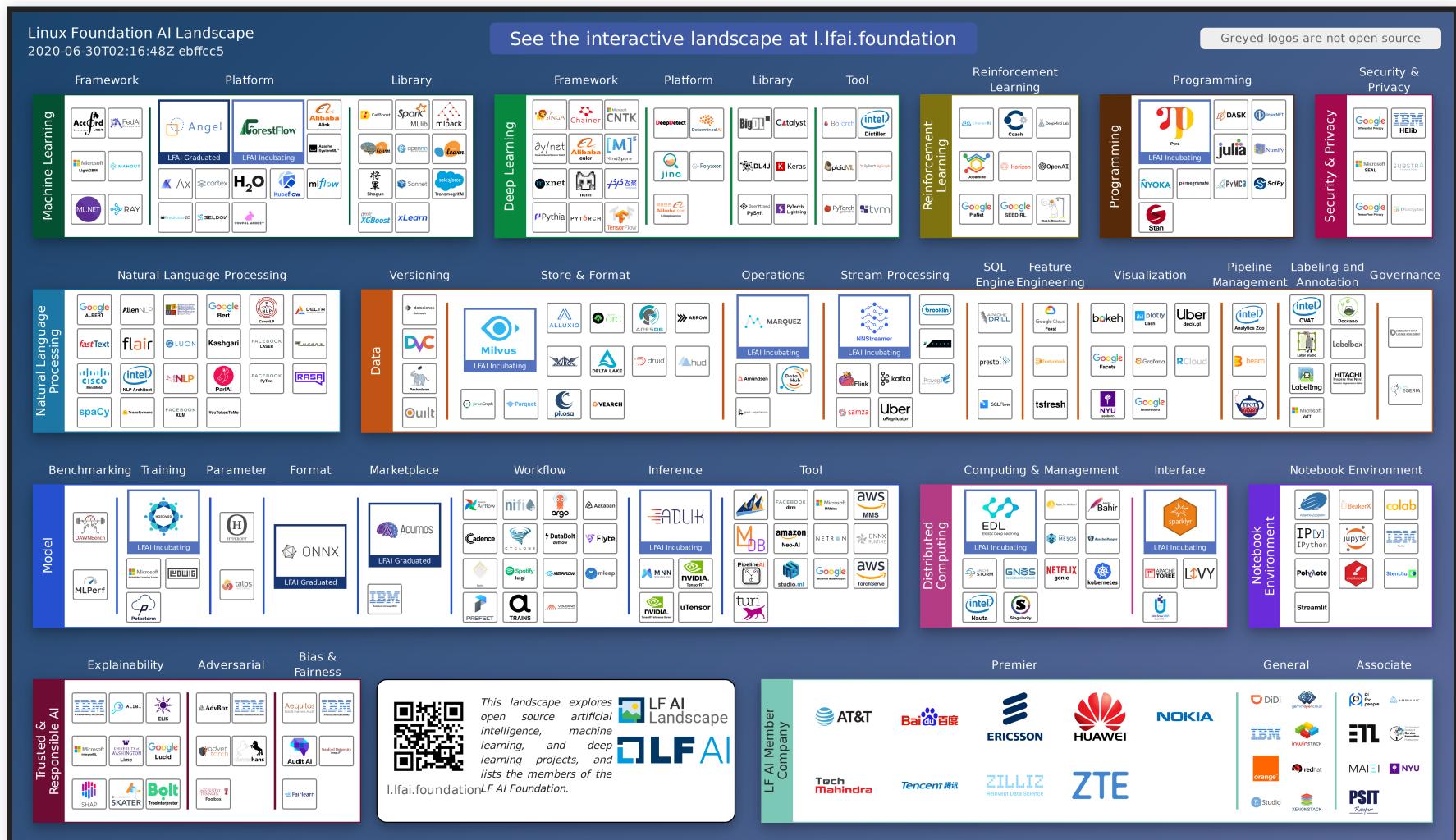


MLOps



<https://ml-ops.org/>

TOOLING LANDSCAPE LF AI



HOMEWORK 5: OPEN SOURCE TOOLS

PROJECT M2: INFRASTRUCTURE QUALITY

(online and offline evaluation, data quality, pipeline quality, CI)

PROCESS AND TECHNICAL DEBT

Christian Kaestner

Required Reading:

- Sculley, David, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. "[Hidden technical debt in machine learning systems](#)." In Advances in neural information processing systems, pp. 2503-2511. 2015.

Suggested Readings:

- Fowler and Highsmith. [The Agile Manifesto](#)
- Steve McConnell. Software project survival guide. Chapter 3
- Pfleeger and Atlee. Software Engineering: Theory and Practice. Chapter 2
- Kruchten, Philippe, Robert L. Nord, and Ipek Ozkaya. "[Technical debt: From metaphor to theory and practice](#)." IEEE Software 29, no. 6 (2012): 18-21.
- Patel, Kayur, James Fogarty, James A. Landay, and Beverly Harrison. "[Investigating statistical machine learning as a tool for software development](#)." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 667-676. 2008.

LEARNING GOALS

- Contrast development processes of software engineers and data scientists
- Outline process conflicts between different roles and suggest ways to mitigate them
- Recognize the importance of process
- Describe common agile practices and their goals
- Understand and correctly use the metaphor of technical debt
- Describe how ML can incur reckless and inadvertent technical debt, outline common sources of technical debt

CASE STUDY: REAL-ESTATE WEBSITE

The screenshot shows the Zillow homepage. At the top, there is a navigation bar with links for "Buy", "Rent", "Sell", "Home Loans", and "Agent finder". To the right of these are the Zillow logo, "Manage Rentals", "Advertise", "Help", and "Sign in". Below the navigation is a large, semi-transparent image of a house at dusk or night, with lights on inside. Overlaid on this image is the text "Reimagine home" in large, white, sans-serif font, followed by a smaller line of text: "We'll help you find a place you'll love." At the bottom, there is a white search bar with the placeholder text "Enter an address, neighborhood, city, or ZIP c..." and a blue magnifying glass icon.

Buy Rent Sell Home Loans Agent finder

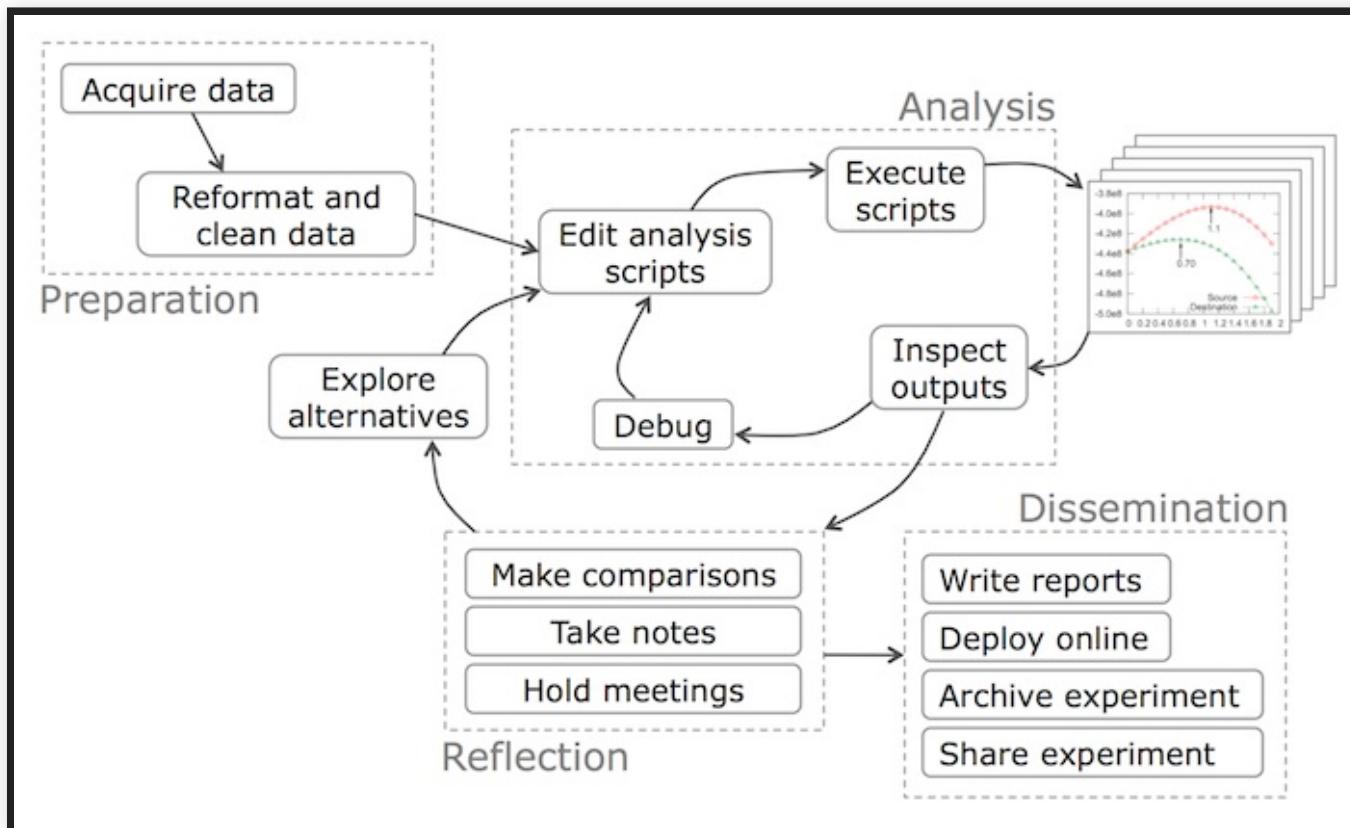
Zillow® Manage Rentals Advertise Help Sign in

Reimagine home

We'll help you find a place you'll love.

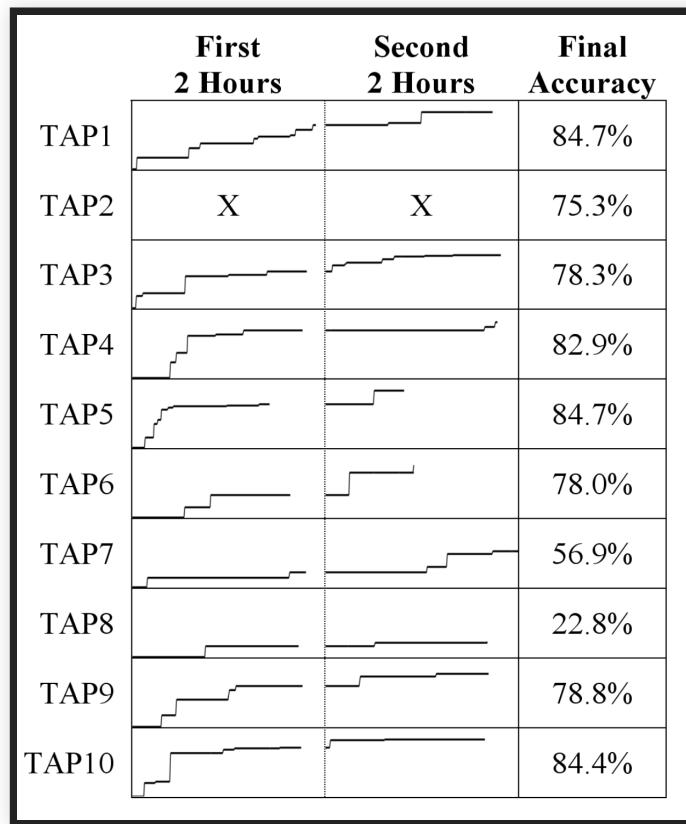
Enter an address, neighborhood, city, or ZIP c...

DATA SCIENCE IS ITERATIVE AND EXPLORATORY



(Source: Guo. "[Data Science Workflow: Overview and Challenges](#)." Blog@CACM, Oct 2013)

DATA SCIENCE IS ITERATIVE AND EXPLORATORY



Source: Patel, Kayur, James Fogarty, James A. Landay, and Beverly Harrison.
["Investigating statistical machine learning as a tool for software development."](#) In
Proc. CHI, 2008.

Speaker notes

This figure shows the result from a controlled experiment in which participants had 2 sessions of 2h each to build a model. Whenever the participants evaluated a model in the process, the accuracy is recorded. These plots show the accuracy improvements over time, showing how data scientists make incremental improvements through frequent iteration.

COMPUTATIONAL NOTEBOOKS

- Origins in "literal programming", interleaving text and code, treating programs as literature (Knuth'84)
- First notebook in Wolfram Mathematica 1.0 in 1988
- Document with text and code cells, showing execution results under cells
- Code of cells is executed, per cell, in a kernel
- Many notebook implementations and supported languages, Python + Jupyter currently most popular

The screenshot shows a Jupyter Notebook interface with a code cell at the top containing Python code to load data from a CSV file and display its first few rows. Below the code is a data frame table with columns: dayIdx, user, userAvgTime, location, dow, isWeekend, and time. The data frame contains five rows of Pittsburgh66Correy data. A text cell below the table provides context about the preprocessing of the 'time' column.

dayIdx	user	userAvgTime	location	dow	isWeekend	time
0	Pittsburgh66Correy	7.045001	Pittsburgh	6	True	0.000000
1	Pittsburgh66Correy	7.045001	Pittsburgh	7	True	6.883333
2	Pittsburgh66Correy	7.045001	Pittsburgh	1	False	6.816667
3	Pittsburgh66Correy	7.045001	Pittsburgh	2	False	7.383333
4	Pittsburgh66Correy	7.045001	Pittsburgh	3	False	0.000000

Data was preprocessed externally, identifying the time at a given day when the light was first turned on (12pm). Weather and sunrise information is not included here, though that'd be important. If the light was off this morning (quite common), 0 is recorded.

```
[ ] # just data encoding and splitting X and Y

X = df.drop(['time'], axis=1)
YnonZero = df['time'] > 0
Y = df['time']

from sklearn import preprocessing
# leDate = preprocessing.LabelEncoder()
# leDate.fit(X['date'])
# leDate.transform(X['date'])

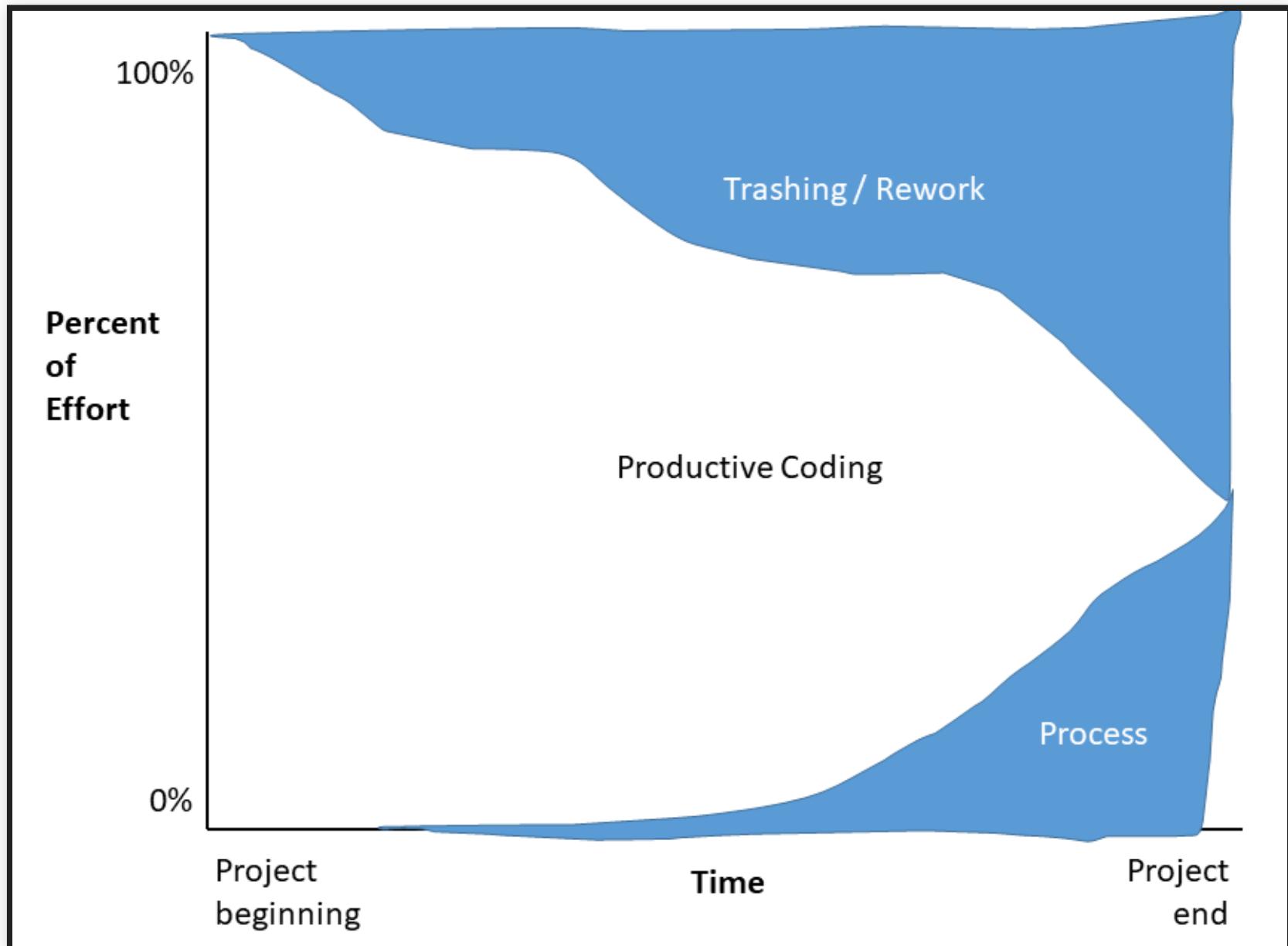
X=X.apply(preprocessing.LabelEncoder().fit_transform)
X
```

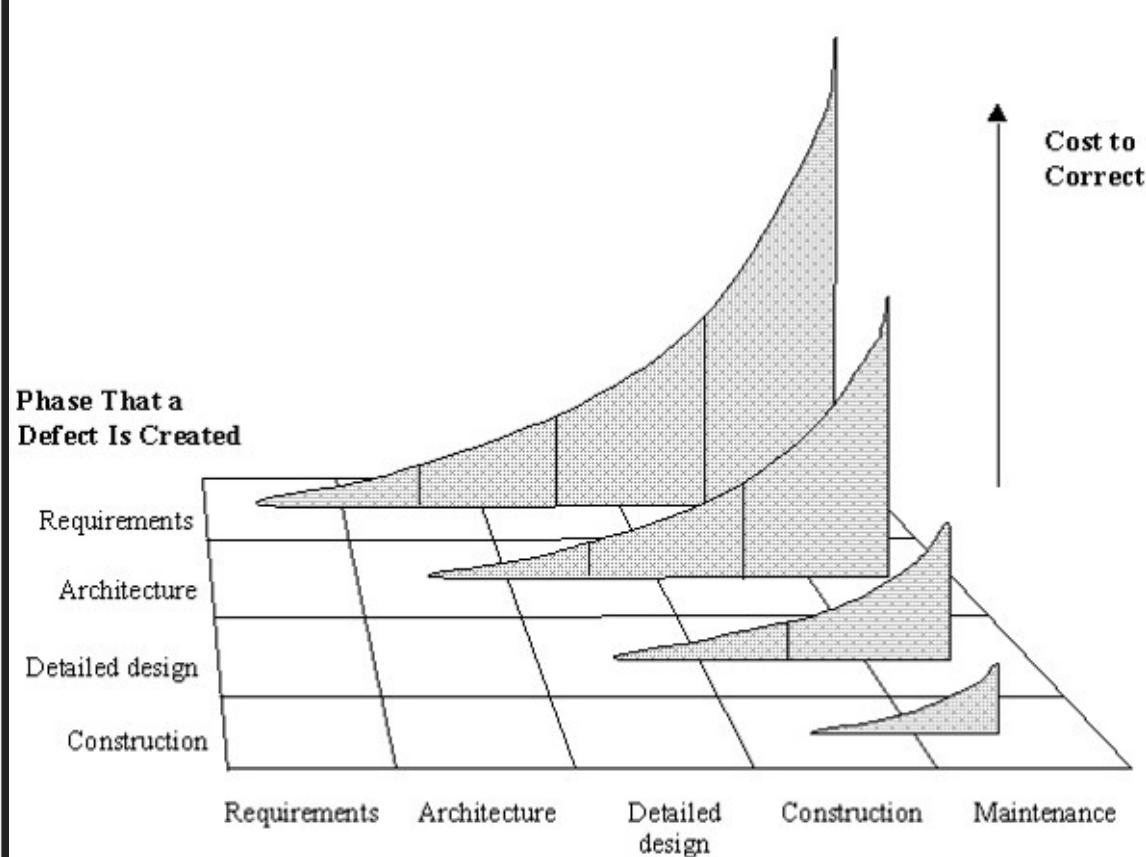
Speaker notes

- See also https://en.wikipedia.org/wiki/Literate_programming
- Demo with public notebook, e.g., https://colab.research.google.com/notebooks/mlcc/intro_to_pandas.ipynb

A SIMPLE PROCESS

1. Discuss the software that needs to be written
2. Write some code
3. Test the code to identify the defects
4. Debug to find causes of defects
5. Fix the defects
6. If not done, return to step 1



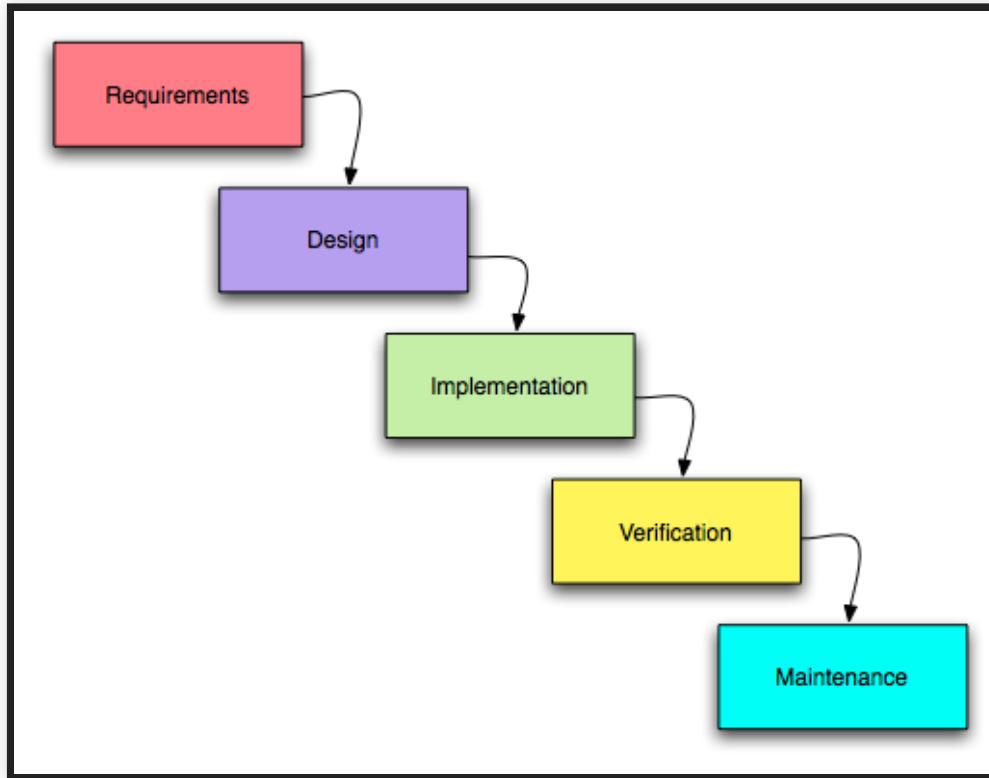


Copyright 1998 Steven C. McConnell. Reprinted with permission from *Software Project Survival Guide* (Microsoft Press, 1998).

Speaker notes

Empirically well established rule: Bugs are increasingly expensive to fix the larger the distance between the phase where they are created vs where they are corrected.

WATERFALL MODEL



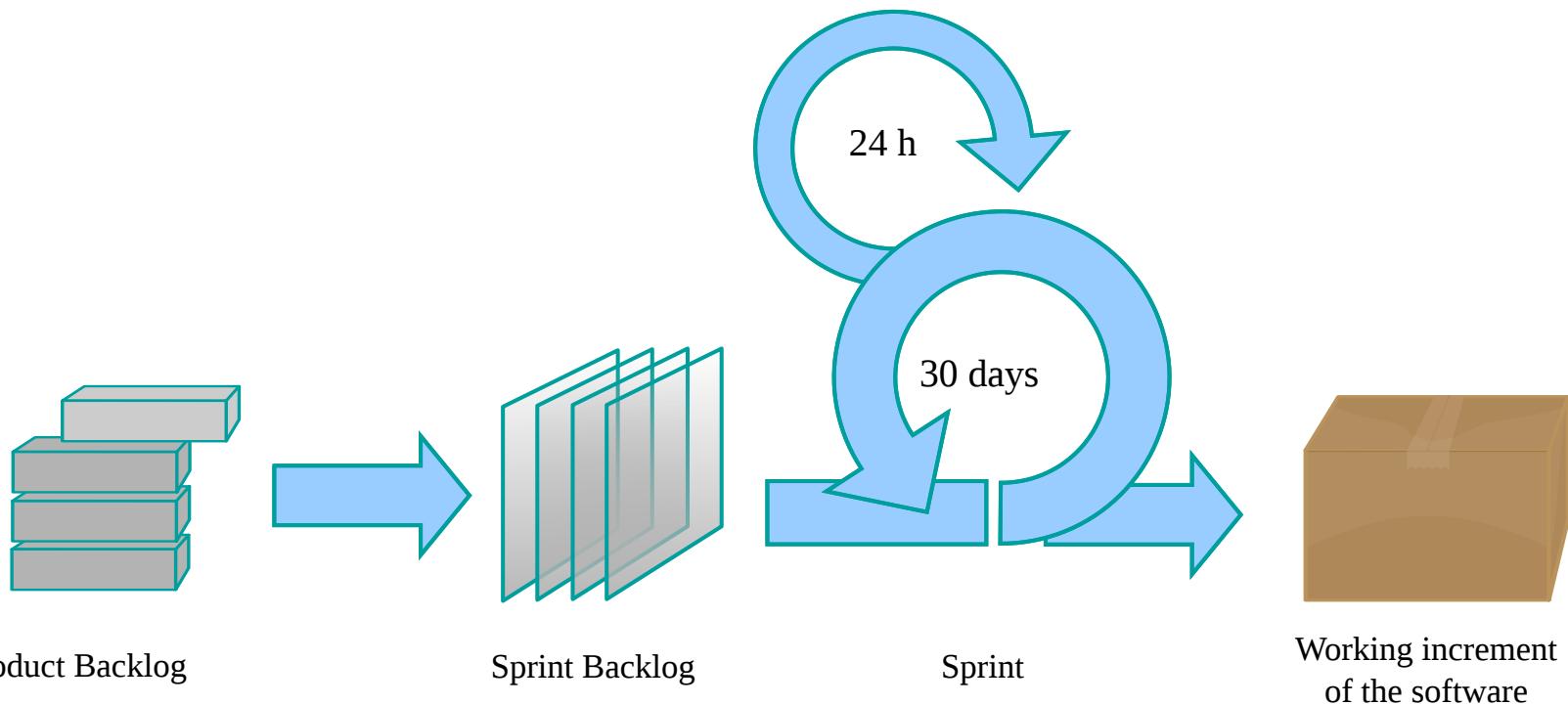
taming the chaos, understand requirements, plan before coding, remember testing

(CC-BY-SA-2.5)

Speaker notes

Although dated, the key idea is still essential -- think and plan before implementing. Not all requirements and design can be made upfront, but planning is usually helpful.

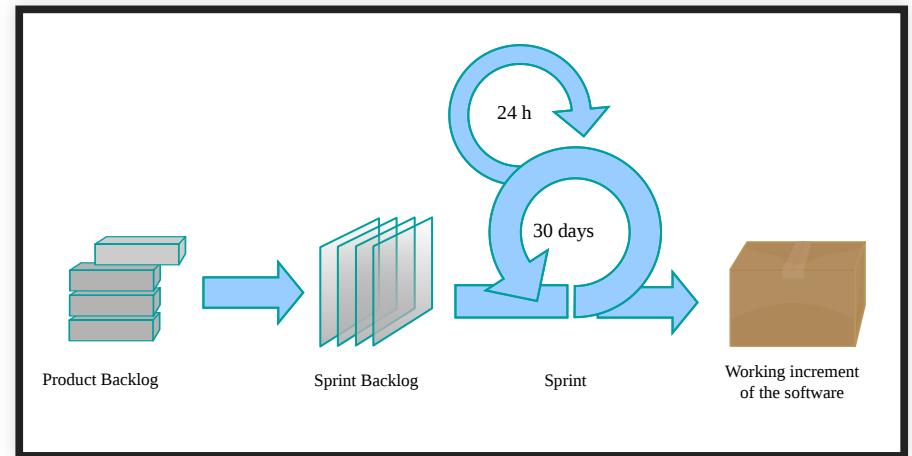
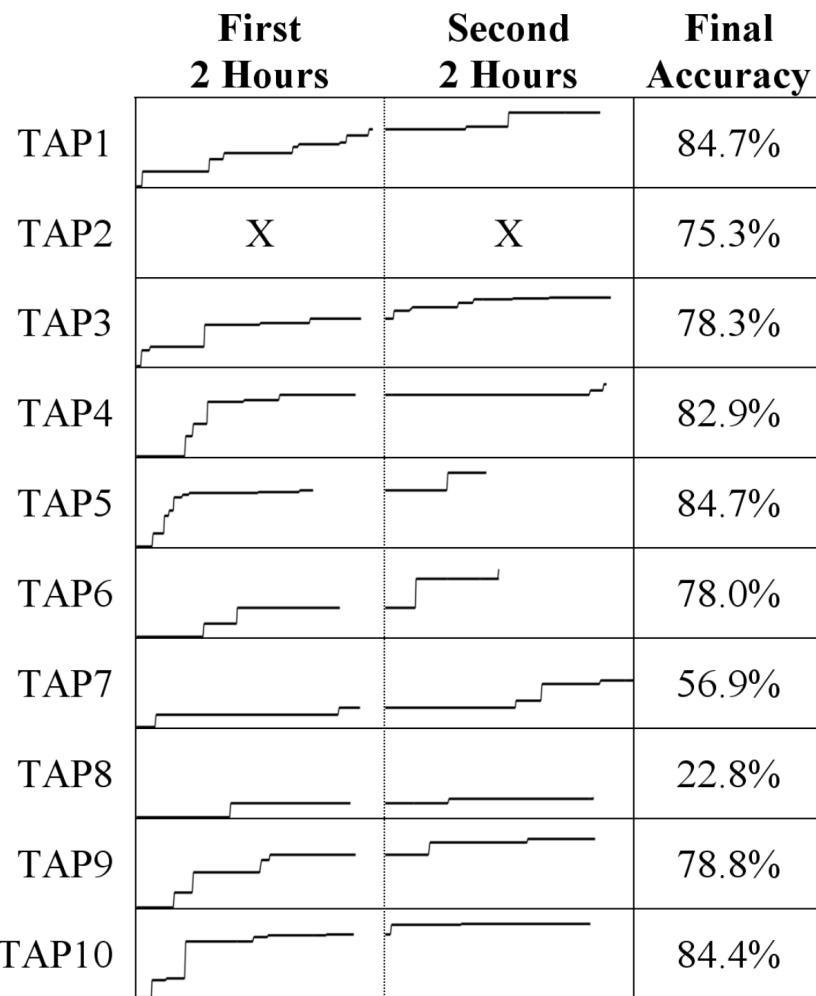
CONSTANT ITERATION: AGILE



working with customers, constant replanning

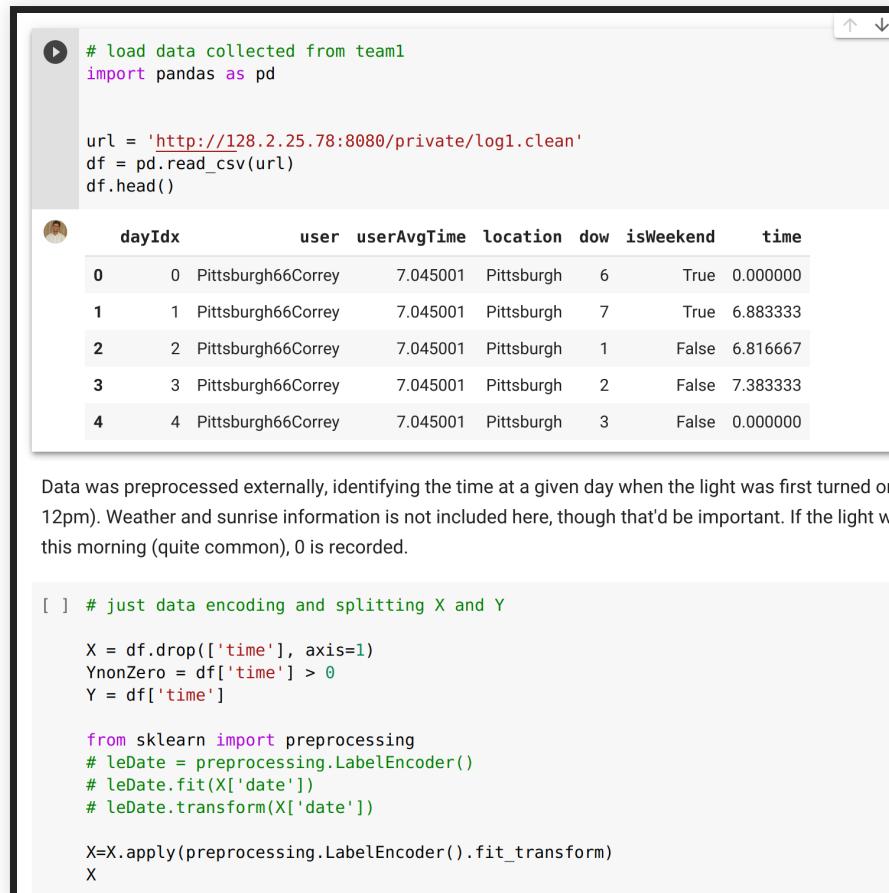
(CC BY-SA 4.0, Lakeworks)

DISCUSSION: ITERATION IN NOTEBOOK VS AGILE?



(CC BY-SA 4.0, Lakeworks)

POOR SOFTWARE ENGINEERING PRACTICES IN NOTEBOOKS?



A screenshot of a Jupyter Notebook cell. The code cell contains:# load data collected from team1
import pandas as pd

url = 'http://128.2.25.78:8080/private/log1.clean'
df = pd.read_csv(url)
df.head()

```
dayIdx      user  userAvgTime  location  dow  isWeekend    time
0    0 Pittsburgh66Correy    7.045001 Pittsburgh    6   True  0.000000
1    1 Pittsburgh66Correy    7.045001 Pittsburgh    7   True  6.883333
2    2 Pittsburgh66Correy    7.045001 Pittsburgh    1  False  6.816667
3    3 Pittsburgh66Correy    7.045001 Pittsburgh    2  False  7.383333
4    4 Pittsburgh66Correy    7.045001 Pittsburgh    3  False  0.000000
```

Data was preprocessed externally, identifying the time at a given day when the light was first turned on (12pm). Weather and sunrise information is not included here, though that'd be important. If the light was off this morning (quite common), 0 is recorded.

```
[ ] # just data encoding and splitting X and Y

X = df.drop(['time'], axis=1)
YnonZero = df['time'] > 0
Y = df['time']

from sklearn import preprocessing
# leDate = preprocessing.LabelEncoder()
# leDate.fit(X['date'])
# leDate.transform(X['date'])

X=X.apply(preprocessing.LabelEncoder().fit_transform)
X
```

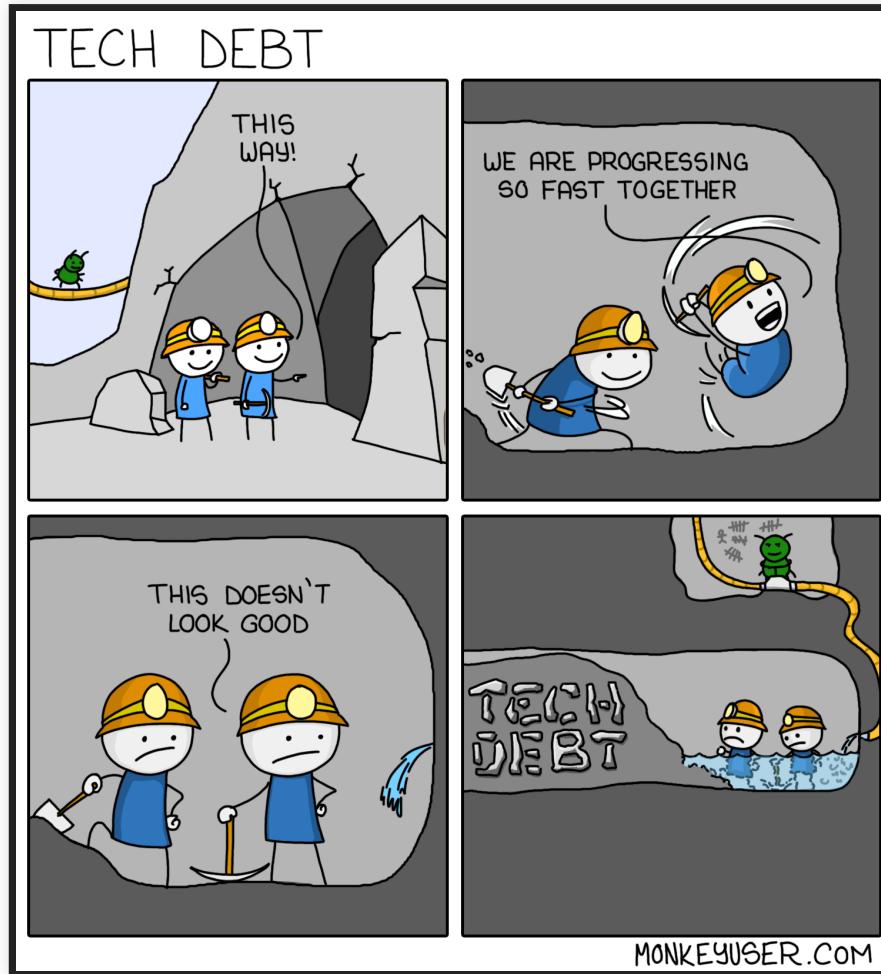


- Little abstraction
- Global state
- No testing
- Heavy copy and paste
- Little documentation
- Poor version control
- Out of order execution
- Poor development features (vs IDE)

PROCESS FOR AI-ENABLED SYSTEMS

- Integrate Software Engineering and Data Science processes
- Establish system-level requirements (e.g., user needs, safety, fairness)
- Inform data science modeling with system requirements (e.g., privacy, fairness)
- Try risky parts first (most likely include ML components; ~spiral)
- Incrementally develop prototypes, incorporate user feedback (~agile)
- Provide flexibility to iterate and improve
- Design system with characteristics of AI component (e.g., UI design, safeguards)
- Plan for testing throughout the process and in production
- Manage project understanding both software engineering and data science workflows
- No existing "best practices" or workflow models

TECHNICAL DEBT



ML AND TECHNICAL DEBT

- Often reckless and inadvertent in inexperienced teams
- ML can seem like an easy addition, but it may cause long-term costs
- Needs to be maintained, evolved, and debugged
- Goals may change, environment may change, some changes are subtle
- Example problems
 - Systems and models are tangled and changing one has cascading effects on the other
 - Untested, brittle infrastructure; manual deployment
 - Unstable data dependencies, replication crisis
 - Data drift and feedback loops
 - Magic constants and dead experimental code paths

Further reading: Sculley, David, et al. [Hidden technical debt in machine learning systems](#). Advances in Neural Information Processing Systems. 2015.

MIDTERM

Home assistant robot and intrusion detection

MANAGING AND PROCESSING LARGE DATASETS

Christian Kaestner

Required watching: Molham Aref. [Business Systems with Machine Learning](#). Guest lecture, 2020.

Suggested reading: Martin Kleppmann. [Designing Data-Intensive Applications](#). OReilly. 2017.

LEARNING GOALS

- Organize different data management solutions and their tradeoffs
- Understand the scalability challenges involved in large-scale machine learning and specifically deep learning
- Explain the tradeoffs between batch processing and stream processing and the lambda architecture
- Recommend and justify a design and corresponding technologies for a given system

CASE STUDY



Search bar: trees



Today



Fri, Oct 25





"ZOOM ADDING CAPACITY"



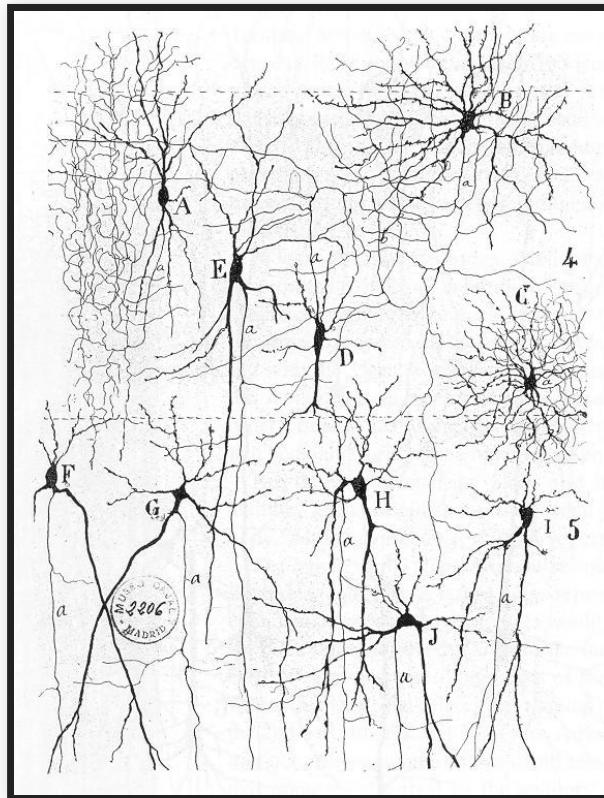
KINDS OF DATA

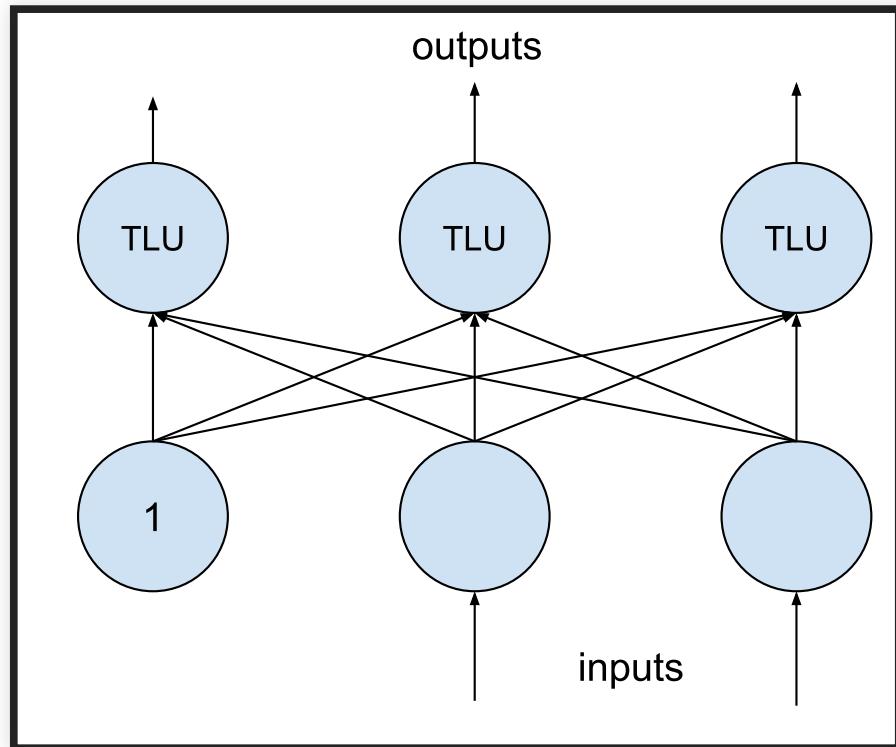
- Training data
- Input data
- Telemetry data
- (Models)

all potentially with huge total volumes and high throughput

need strategies for storage and processing

EXCURSION: DEEP LEARNING & SCALE





$$o_1 = \phi(b_1 + w_{1,1}x_1 + w_{1,2}x_2)$$

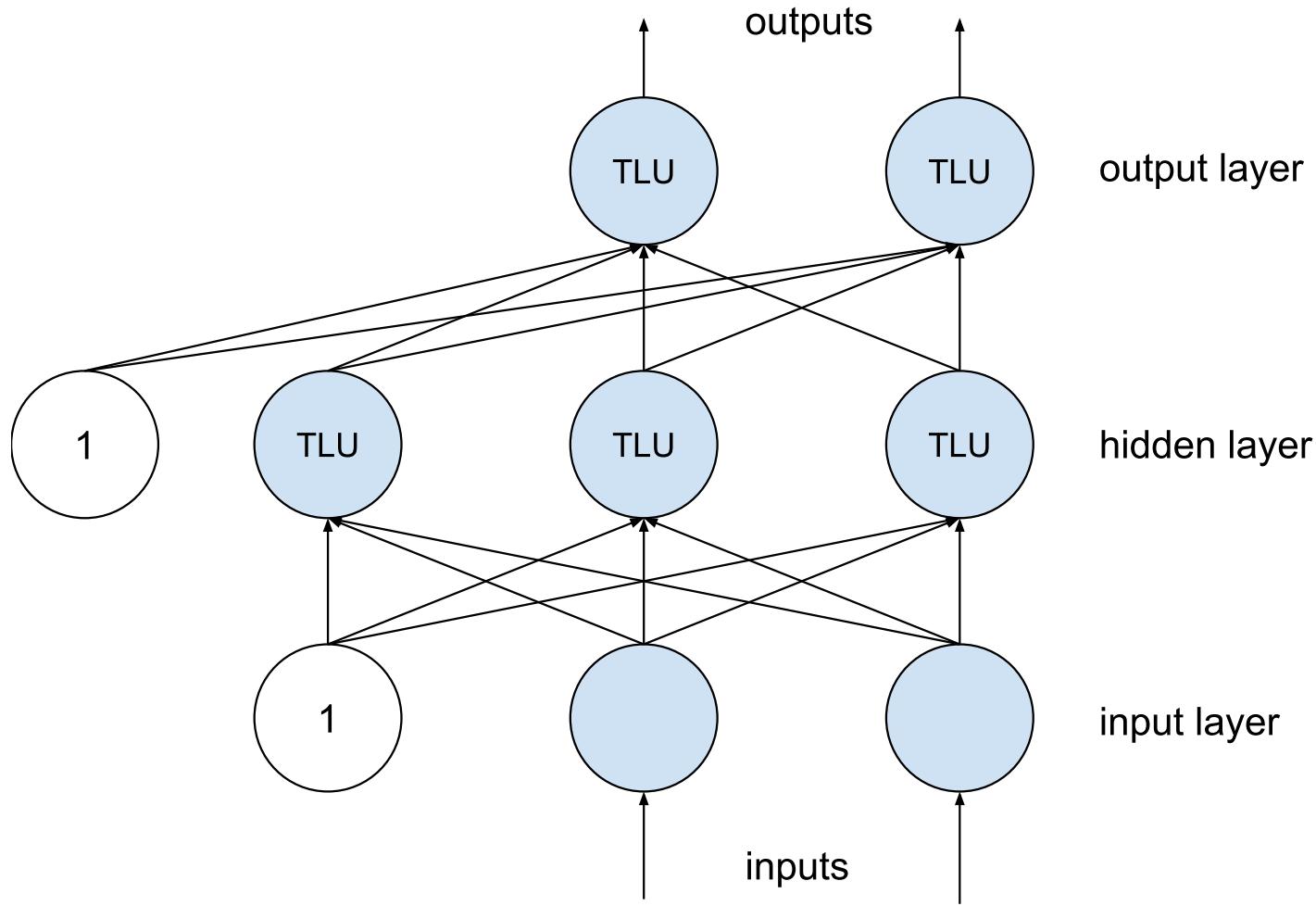
$$o_2 = \phi(b_2 + w_{2,1}x_1 + w_{2,2}x_2)$$

$$o_3 = \phi(b_3 + w_{3,1}x_1 + w_{3,2}x_2)$$

$$f_{\mathbf{W}, \mathbf{b}}(\mathbf{X}) = \phi(\mathbf{W} \cdot \mathbf{X} + \mathbf{b})$$

(**W** and **b** are parameters of the model)

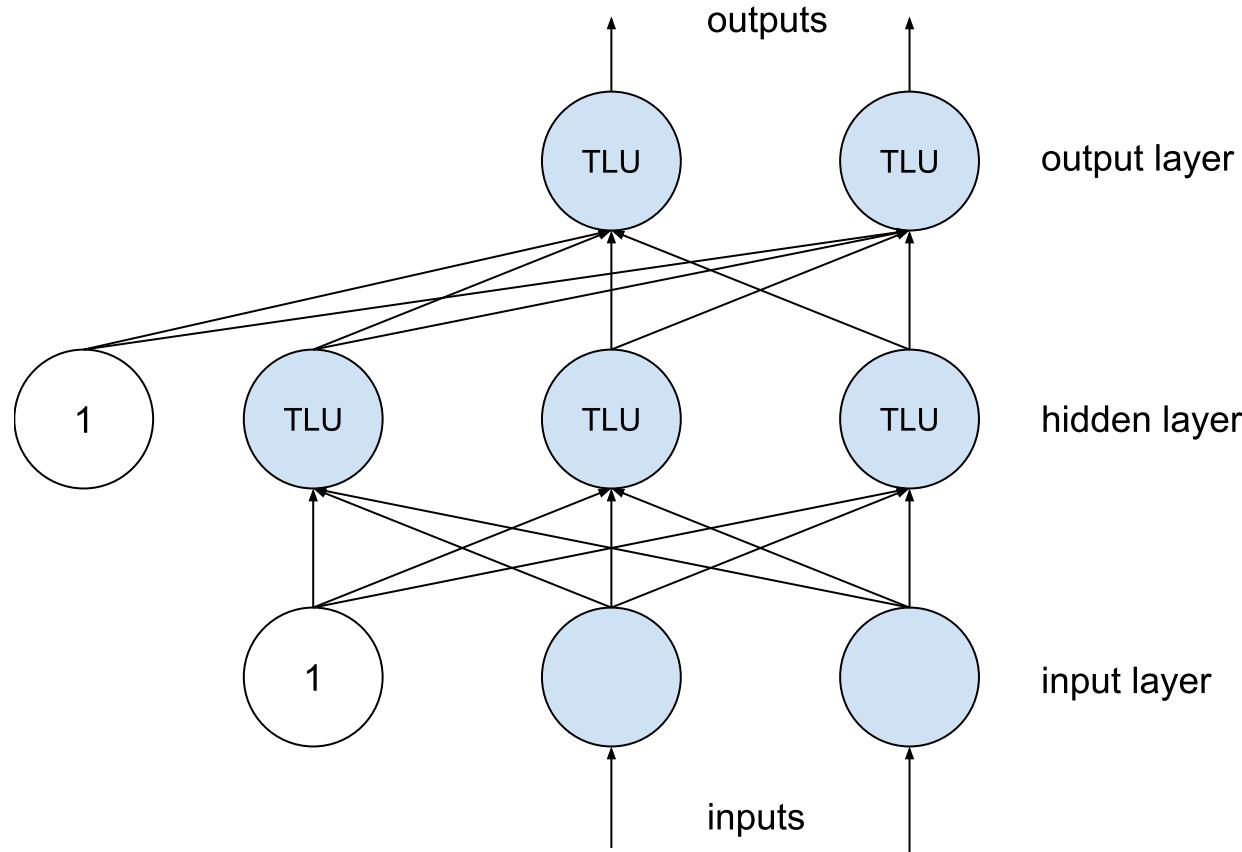
MULTIPLE LAYERS



Speaker notes

Layers are fully connected here, but layers may have different numbers of neurons

$$f_{\mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_o, \mathbf{b}_o}(\mathbf{X}) = \phi(\mathbf{W}_o \cdot \phi(\mathbf{W}_h \cdot \mathbf{X} + \mathbf{b}_h) + \mathbf{b}_o)$$



(matrix multiplications interleaved with step function)

EXAMPLE SCENARIO

- MNIST Fashion dataset of 70k 28x28 grayscale pixel images, 10 output classes
- $28 \times 28 = 784$ inputs in input layers (each 0..255)
- Example model with 3 layers, 300, 100, and 10 neurons

```
model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=[28, 28]),
    # 784*300+300 = 235500 parameters
    keras.layers.Dense(300, activation="relu"),
    # 300*100+100 = 30100 parameters
    keras.layers.Dense(100, activation="relu"),
    # 100*10+10 = 1010 parameters
    keras.layers.Dense(10, activation="softmax")
])
```

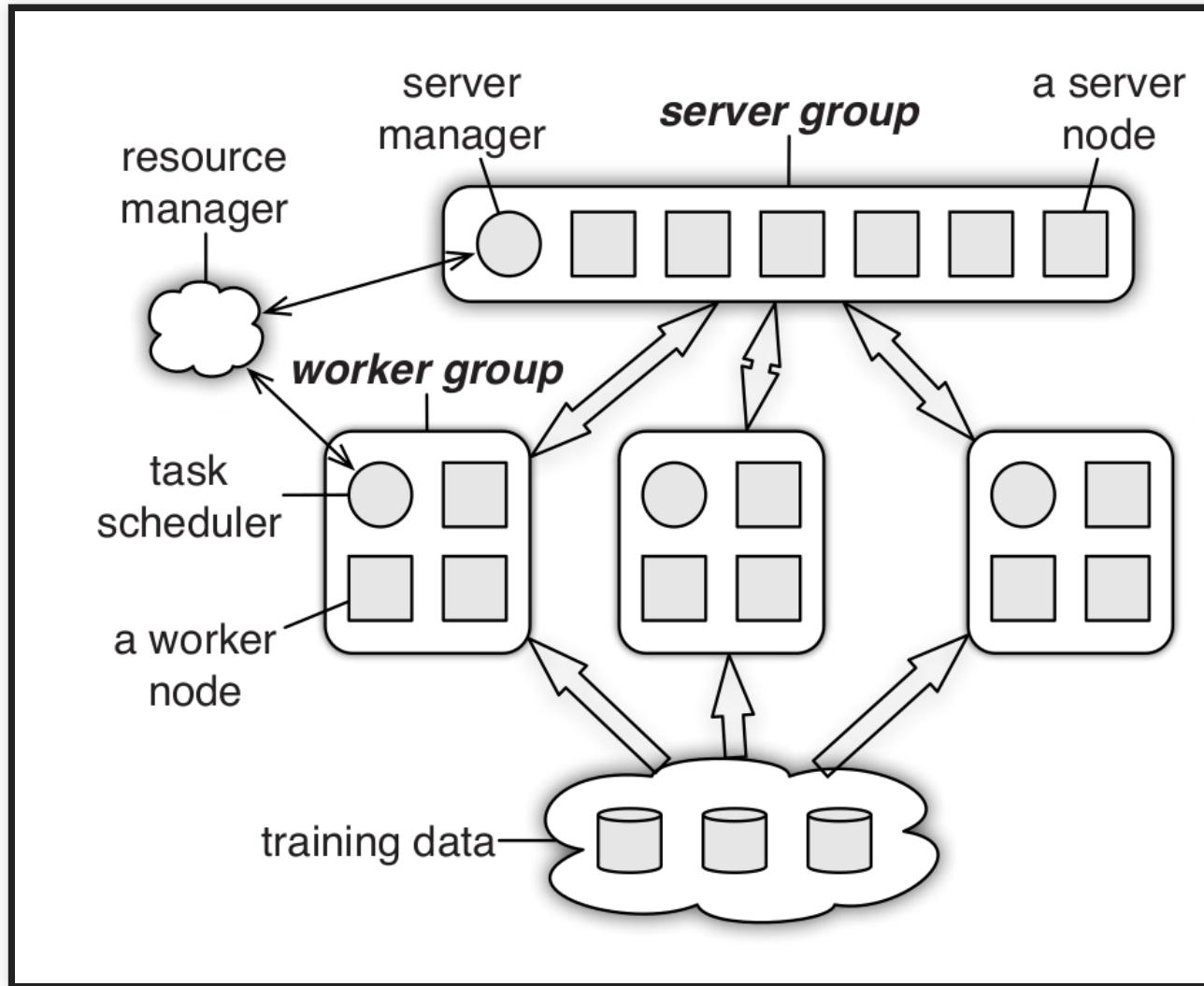
Total of 266,610 parameters in this small example! (Assuming float types, that's 1 MB)

COST & ENERGY CONSUMPTION

Model	Hardware	Hours	CO2	Cloud cost in USD
Transformer	P100x8	84	192	289–981
ELMo	P100x3	336	262	433–1472
BERT	V100x64	79	1438	3751–13K
NAS	P100x8	274,120	626,155	943K–3.2M
GPT-2	TPUv3x32	168	—	13K–43K

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. "[Energy and Policy Considerations for Deep Learning in NLP](#)." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3645–3650. 2019.

PARAMETER SERVER ARCHITECTURE



DOCUMENT DATA MODELS

```
{  
  "id": 1,  
  "name": "Christian",  
  "email": "kaestner@cs.",  
  "dpt": [  
    {"name": "ISR", "address": "..."}  
  ],  
  "other": { ... }  
}
```

```
db.getCollection('users').find({ "name": "Christian" })
```

LOG FILES, UNSTRUCTURED DATA

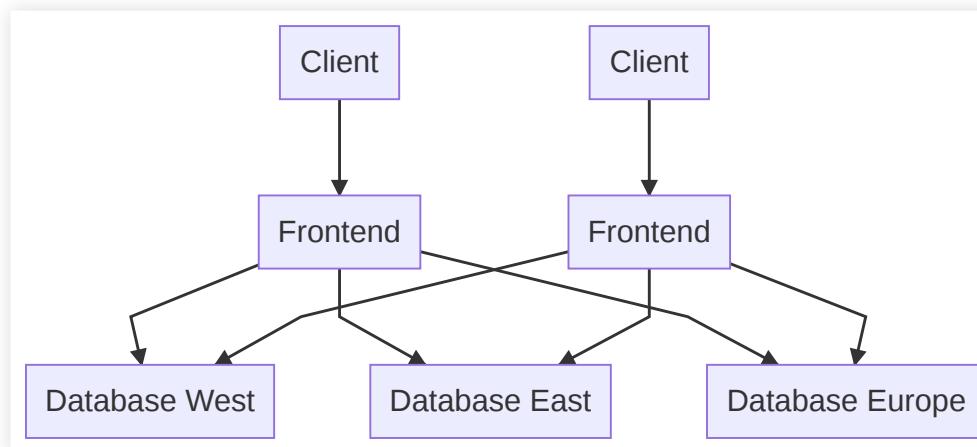
```
2020-06-25T13:44:14, 601844, GET /data/m/goyas+ghosts+2006/17.mpg
2020-06-25T13:44:14, 935791, GET /data/m/the+big+circus+1959/68.mp
2020-06-25T13:44:14, 557605, GET /data/m/elvis+meets+nixon+1997/17
2020-06-25T13:44:14, 140291, GET /data/m/the+house+of+the+spirits+
2020-06-25T13:44:14, 425781, GET /data/m/the+theory+of+everything+
2020-06-25T13:44:14, 773178, GET /data/m/toy+story+2+1999/59.mpg
2020-06-25T13:44:14, 901758, GET /data/m/ignition+2002/14.mpg
2020-06-25T13:44:14, 911008, GET /data/m/toy+story+3+2010/46.mpg
```

PARTITIONING

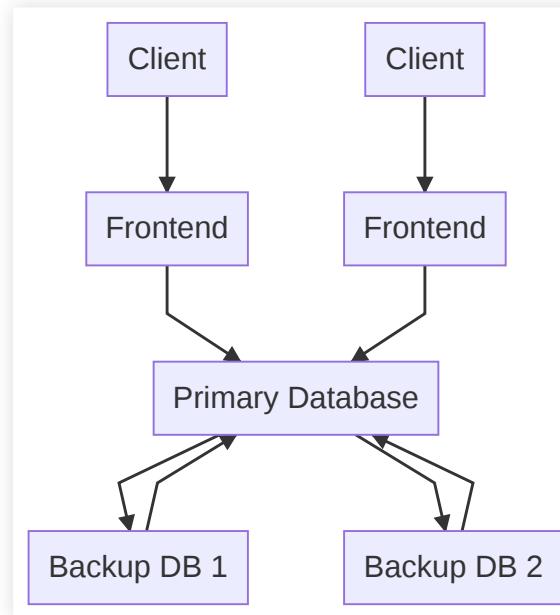
Divide data:

- Horizontal partitioning: Different rows in different tables; e.g., movies by decade, hashing often used
- Vertical partitioning: Different columns in different tables; e.g., movie title vs. all actors

Tradeoffs?



REPLICATION STRATEGIES: LEADERS AND FOLLOWERS



BATCH PROCESSING

- Analyzing TB of data, typically distributed storage
- Filtering, sorting, aggregating
- Producing reports, models, ...

```
cat /var/log/nginx/access.log |  
awk '{print $7}' |  
sort |  
uniq -c |  
sort -r -n |  
head -n 5
```

DISTRIBUTED BATCH PROCESSING

- Process data locally at storage
- Aggregate results as needed
- Separate plumbing from job logic

MapReduce as common framework

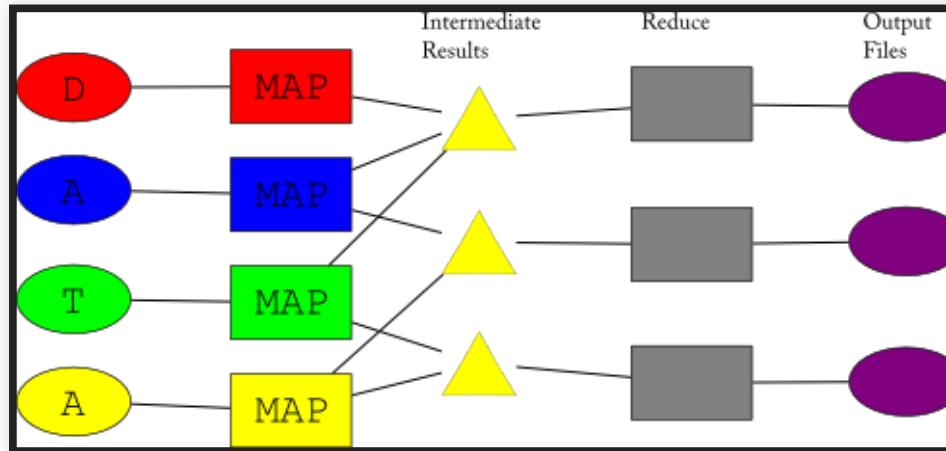


Image Source: Ville Tuulos (CC BY-SA 3.0)

KEY DESIGN PRINCIPLE: DATA LOCALITY

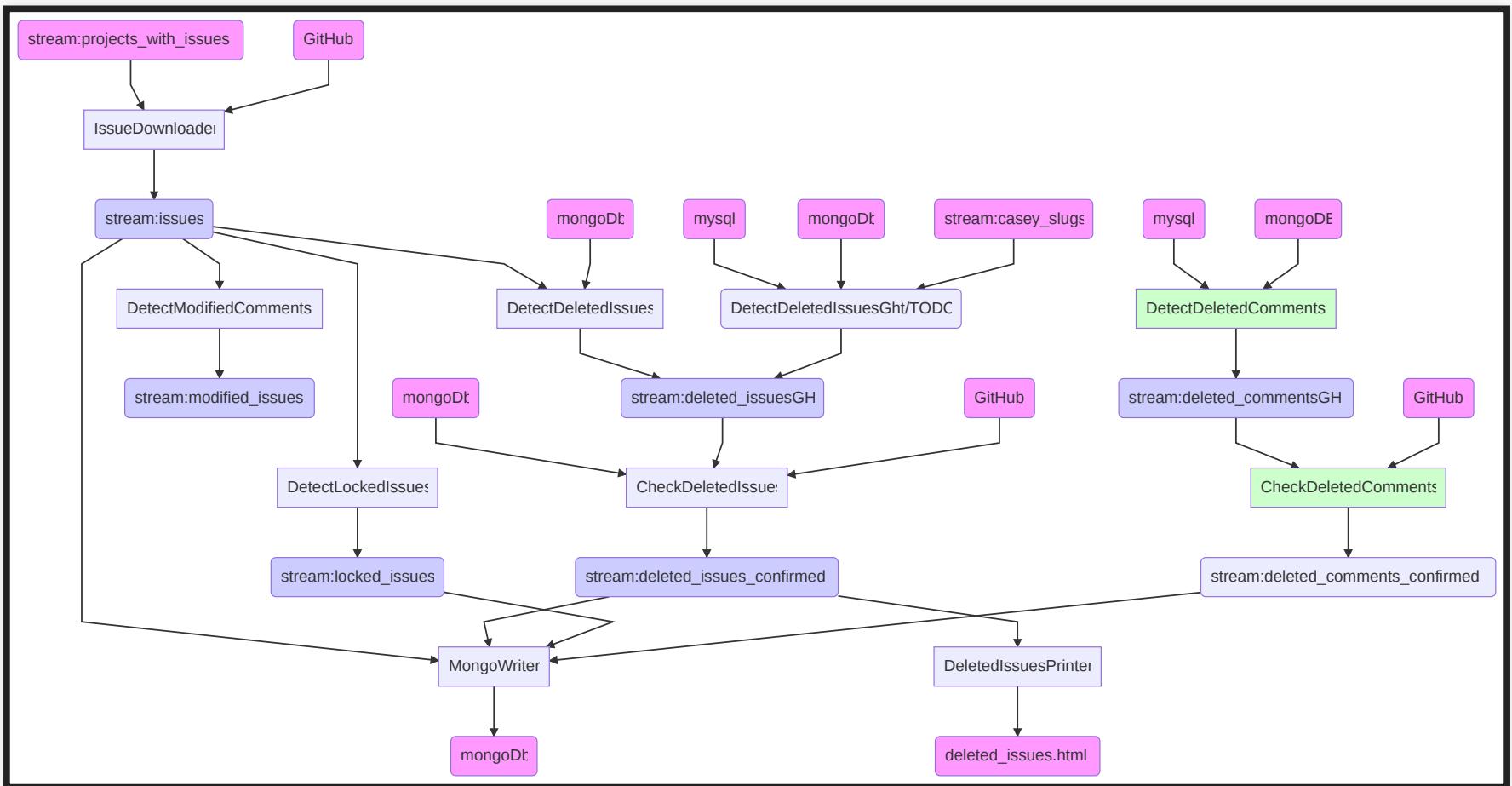
*Moving Computation is Cheaper than Moving Data --
Hadoop Documentation*

- Data often large and distributed, code small
- Avoid transferring large amounts of data
- Perform computation where data is stored (distributed)
- Transfer only results as needed

- "The map reduce way"

STREAM PROCESSING

Like shell programs: Read from stream, produce output in other stream. Loose coupling

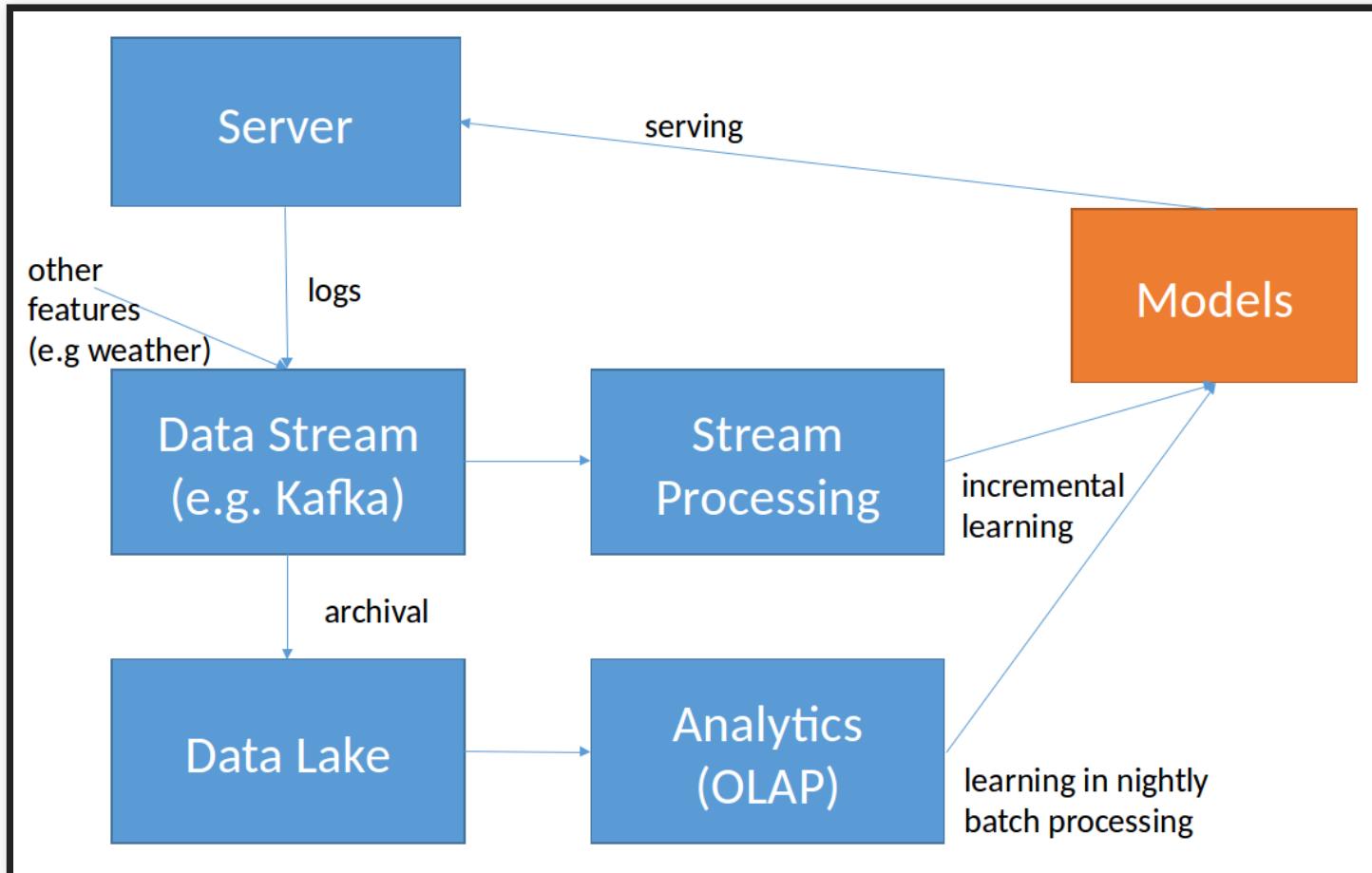


EVENT SOURCING

- Append only databases
- Record edit events, never mutate data
- Compute current state from all past events, can reconstruct old state
- For efficiency, take state snapshots
- Similar to traditional database logs

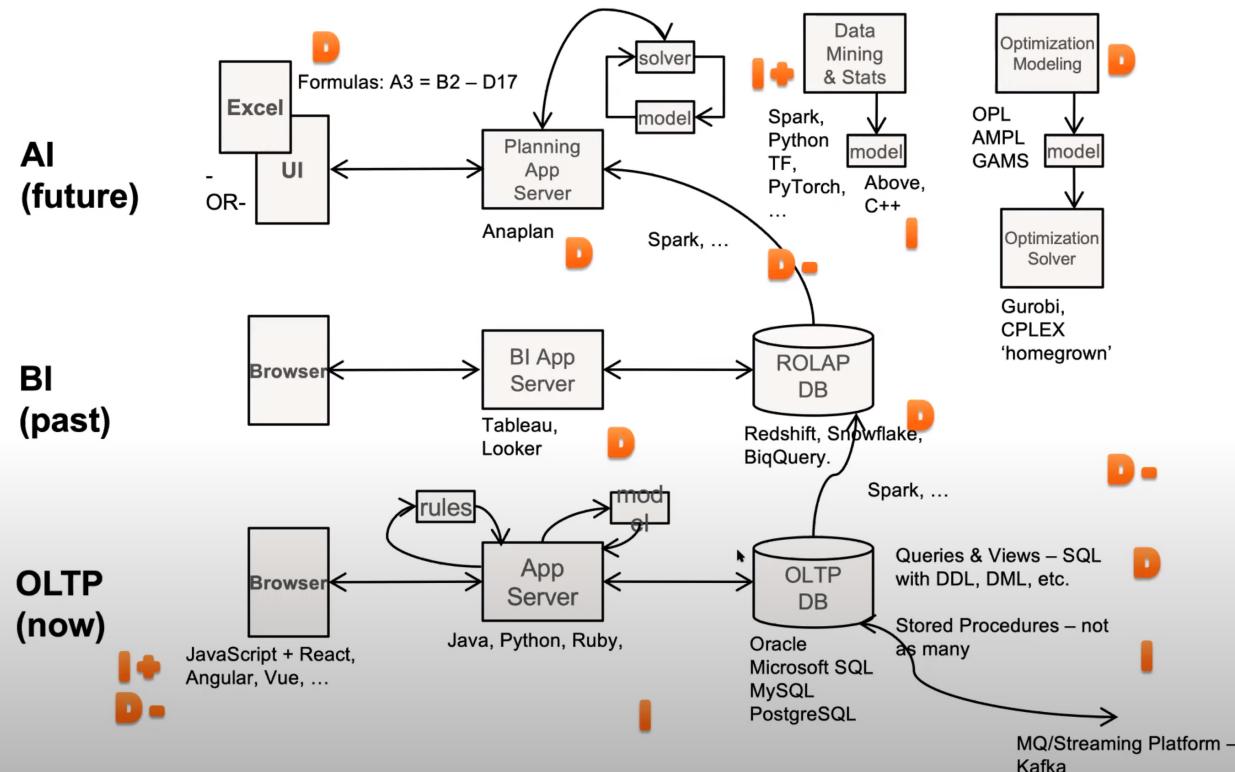
```
createUser(id=5, name="Christian", dpt="SCS")
updateUser(id=5, dpt="ISR")
deleteUser(id=5)
```

LAMBDA ARCHITECTURE AND MACHINE LEARNING



- Learn accurate model in batch job
- Learn incremental model in stream processor

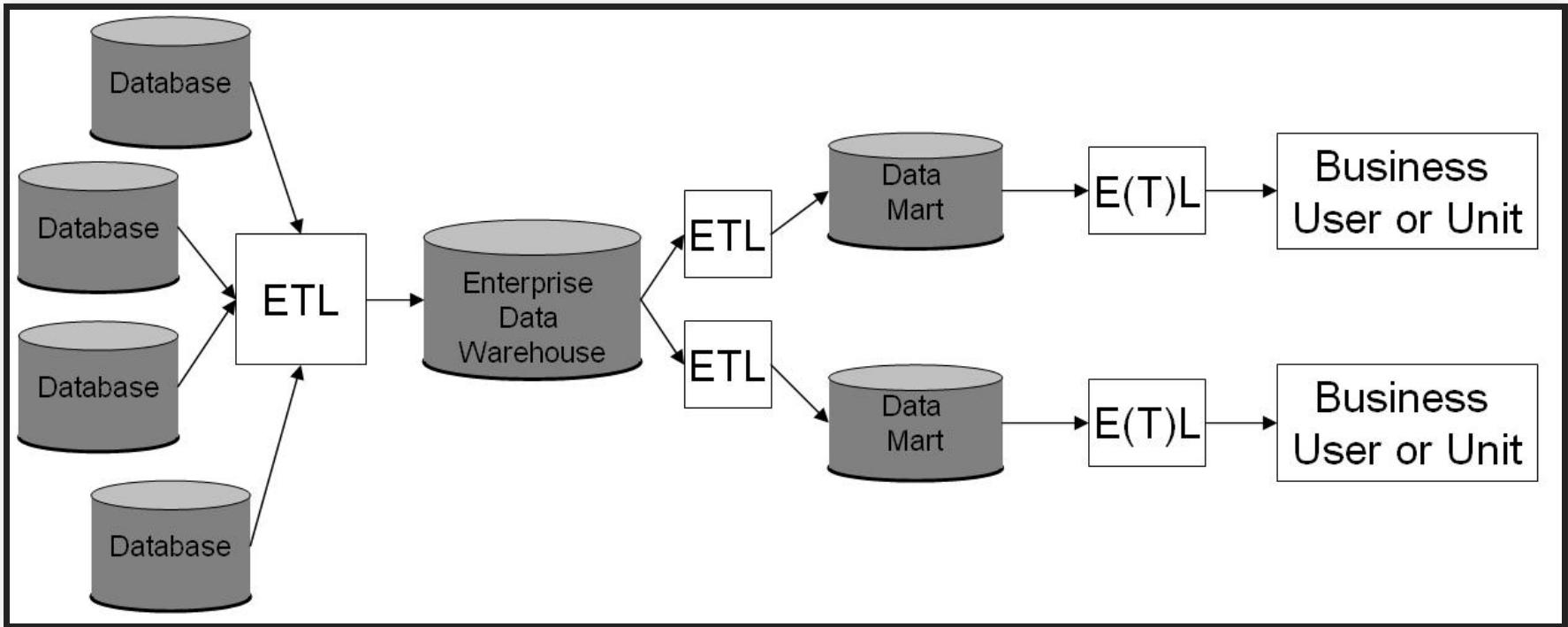
Enterprise Tech Stack – Now isn't much different



Molham Aref "Business Systems with Machine Learning"

DATA WAREHOUSING (OLAP)

- Large denormalized databases with materialized views for large scale reporting queries
- e.g. sales database, queries for sales trends by region
- Read-only except for batch updates: Data from OLTP systems loaded periodically, e.g. over night



PERFORMANCE MONITORING OF DISTRIBUTED SYSTEMS



Source: <https://blog.appdynamics.com/tag/fiserv/>

INTRO TO ETHICS AND FAIRNESS

Eunsuk Kang

Required reading: R. Caplan, J. Donovan, L. Hanson, J. Matthews. "Algorithmic Accountability: A Primer", Data & Society (2018).

LEARNING GOALS

- Review the importance of ethical considerations in designing AI-enabled systems
- Recall basic strategies to reason about ethical challenges
- Diagnose potential ethical issues in a given system
- Understand the types of harm that can be caused by ML
- Understand the sources of bias in ML

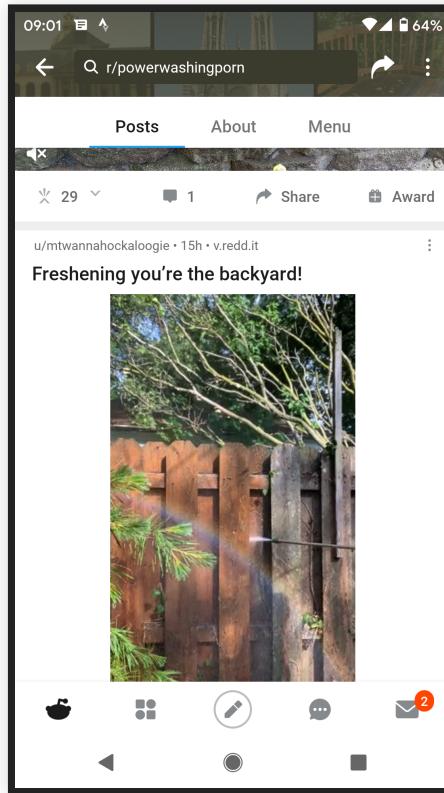
A close-up portrait of Martin Shkreli, a man with dark hair and a beard, wearing a suit and tie, looking slightly to the side with a neutral expression.

In September 2015, Shkreli received widespread criticism when Turing obtained the manufacturing license for the antiparasitic drug Daraprim and raised its price by a factor of 56 (from USD 13.5 to 750 per pill), leading him to be referred to by the media as "the most hated man in America" and "Pharma Bro".

-- [Wikipedia](#)

"I could have raised it higher and made more profits for our shareholders. Which is my primary duty." -- Martin Shkreli

OPTIMIZING FOR ORGANIZATIONAL OBJECTIVE



- How do we maximize the user engagement?
 - Infinite scroll: Encourage non-stop, continual use
 - Personal recommendations: Suggest news feed to increase engagement
 - Push notifications: Notify disengaged users to return to the app

DISINFORMATION & POLARIZATION

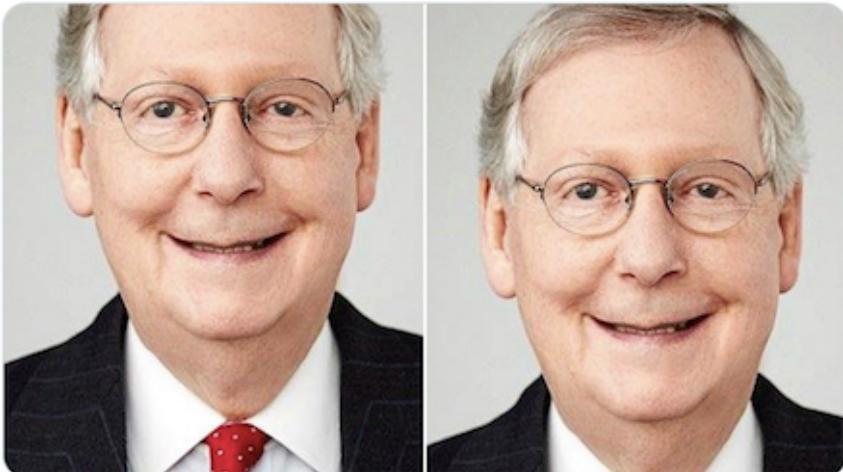


DISCRIMINATION

 Tony "Abolish (Pol)ICE" Arcieri 🇺🇸
@bascule

Trying a horrible experiment...

Which will the Twitter algorithm pick: Mitch McConnell or Barack Obama?



6:05 PM · Sep 19, 2020 · Twitter Web App

64K Retweets 16.5K Quote Tweets 198.3K Likes

<https://twitter.com/bascule/status/1307440596668182528>

CHALLENGES

- Misalignment between organizational goals & societal values
 - Financial incentives often dominate other goals ("grow or die")
- Insufficient amount of regulations
 - Little legal consequences for causing negative impact (with some exceptions)
 - Poor understanding of socio-technical systems by policy makers
- Engineering challenges, both at system- & ML-level
 - Difficult to clearly define or measure ethical values
 - Difficult to predict possible usage contexts
 - Difficult to predict impact of feedback loops
 - Difficult to prevent malicious actors from abusing the system
 - Difficult to interpret output of ML and make ethical decisions
 - ...

These problems have existed before, but they are being rapidly exacerbated by the widespread use of ML

LEGALLY PROTECTED CLASSES (US)

- Race (Civil Rights Act of 1964)
- Color (Civil Rights Act of 1964)
- Sex (Equal Pay Act of 1963; Civil Rights Act of 1964)
- Religion (Civil Rights Act of 1964)
- National origin (Civil Rights Act of 1964)
- Citizenship (Immigration Reform and Control Act)
- Age (Age Discrimination in Employment Act of 1967)
- Pregnancy (Pregnancy Discrimination Act)
- Familial status (Civil Rights Act of 1968)
- Disability status (Rehabilitation Act of 1973; Americans with Disabilities Act of 1990)
- Veteran status (Vietnam Era Veterans' Readjustment Assistance Act of 1974; Uniformed Services Employment and Reemployment Rights Act)
- Genetic information (Genetic Information Nondiscrimination Act)

Barocas, Solon and Moritz Hardt. "[Fairness in machine learning](#)." NIPS Tutorial 1 (2017).

Equality



The assumption is that **everyone benefits from the same supports**. This is equal treatment.

Equity



Everyone gets the supports they need (this is the concept of "affirmative action"), thus producing equity.

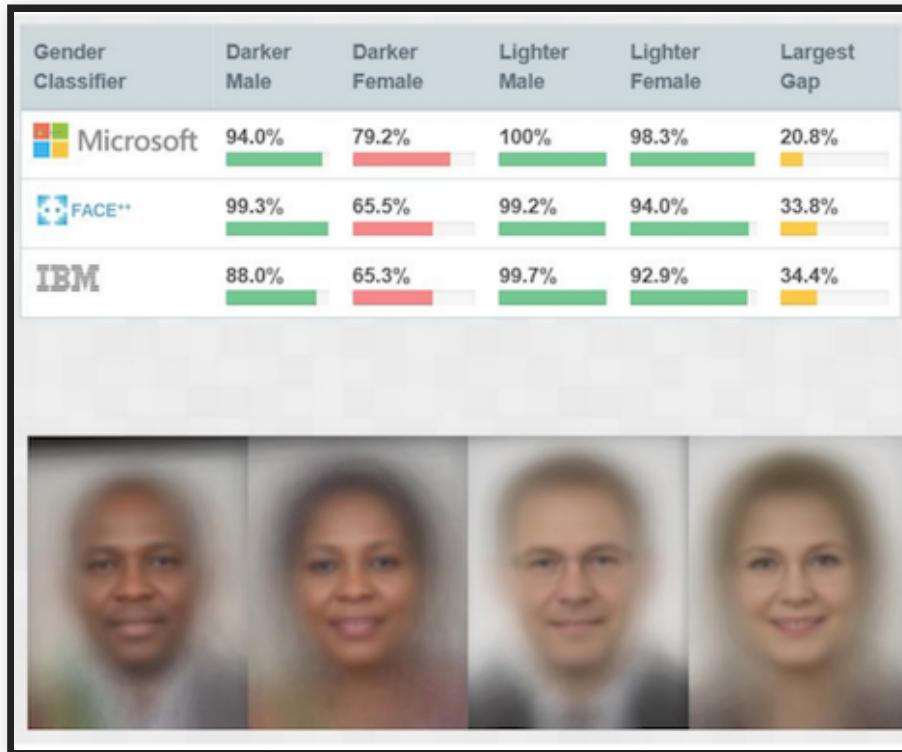
Justice



All 3 can see the game without supports or accommodations because **the cause(s) of the inequity was addressed**.
The systemic barrier has been removed.

HARMS OF ALLOCATION

- Withhold opportunities or resources
- Poor quality of service, degraded user experience for certain groups



Other examples?

HARMS OF REPRESENTATION

- Reinforce stereotypes, subordination along the lines of identity

Ads by Google

[Latanya Sweeney, Arrested?](#)

1) Enter Name and State. 2) Access Full Background Checks Instantly.

www.instantcheckmate.com/

[Latanya Sweeney](#)

Public Records Found For: Latanya Sweeney. View Now.

www.publicrecords.com/

[La Tanya](#)

Search for La Tanya Look Up Fast Results now!

www.ask.com/La+Tanya

Other examples?

Latanya Sweeney. [Discrimination in Online Ad Delivery](#), SSRN (2013).

CASE STUDY: COLLEGE ADMISSION



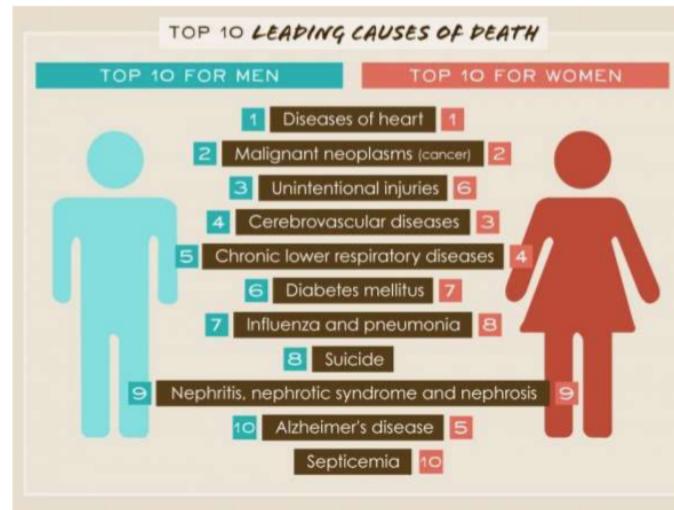
- Objective: Decide "Is this student likely to succeed"?
- Possible harms: Allocation of resources? Quality of service? Stereotyping? Denigration? Over-/Under-representation?

NOT ALL DISCRIMINATION IS HARMFUL



FEDERAL TRADE COMMISSION

Mortgage discrimination is against the law.



- Loan lending: Gender discrimination is illegal.
- Medical diagnosis: Gender-specific diagnosis may be desirable.
- Discrimination is a **domain-specific** concept!

Other examples?

WHERE DOES THE BIAS COME FROM?

The image displays two side-by-side screenshots of the Google Translate interface, illustrating gender bias in machine translation.

Top Screenshot (English to Turkish):

- Input (English):** "He is a nurse
She is a doctor"
- Output (Turkish):** "O bir hemşire
O bir doktor"
- Notes:** The input text is highlighted in blue, while the output text is black. This visual cue, combined with the gendered nature of the translated output, suggests that the model associates "he" with "hemşire" and "she" with "doktor".

Bottom Screenshot (Turkish to English):

- Input (Turkish):** "O bir hemşire
O bir doktor"
- Output (English):** "She is a nurse
He is a doctor" (with a checkmark icon)
- Notes:** The input text is black, while the output text is highlighted in blue. This visual cue, combined with the gendered nature of the translated output, suggests that the model associates "she" with "nurse" and "he" with "doctor".

Caliskan et al., *Semantics derived automatically from language corpora contain human-like biases*, Science (2017).

HISTORICAL BIAS

Data reflects past biases, not intended outcomes

The screenshot shows a search results page for the query "ceo". The interface includes a logo of a cartoon duck, a search bar with the term "ceo", and a magnifying glass icon. Below the search bar are navigation links for "All", "Images" (which is underlined), "Videos", "News", "Maps", and "Meanings". On the right, there is a "Settings" dropdown menu. Further down, there are filters for "All Regions", "Safe Search: Moderate", "All Sizes", "All Types", "All Layouts", and "All Colors". The main content area displays five search results, each featuring a portrait of a man in a suit and a brief summary:

- Cronos CEO: \$1.8 billion from Big Tob...**
marketwatch.com
- Marriott CEO talks...**
bizjournals.com
- Goldman Sachs may claw back milli...**
nypost.com
- Coolest thing about Tesla's C**
businessinsider.com

Below these results, there are five smaller thumbnail images showing more portraits of men in suits.



1000 × 1000

Croatian Doctor To...
croatiaweek.com



999 × 666

Lufthansa CEO Says Brit...
skift.com



1000 × 750

'The ideal match': Lululemon...
business.financialpost.com



750 × 999

Fairview names St...
bizjournals.com



CEO pay: Top 10 highest...
usatoday.com

TAINTED EXAMPLES

Samples or labels reflect human bias

TECH \ AMAZON \ ARTIFICIAL INTELLIGENCE

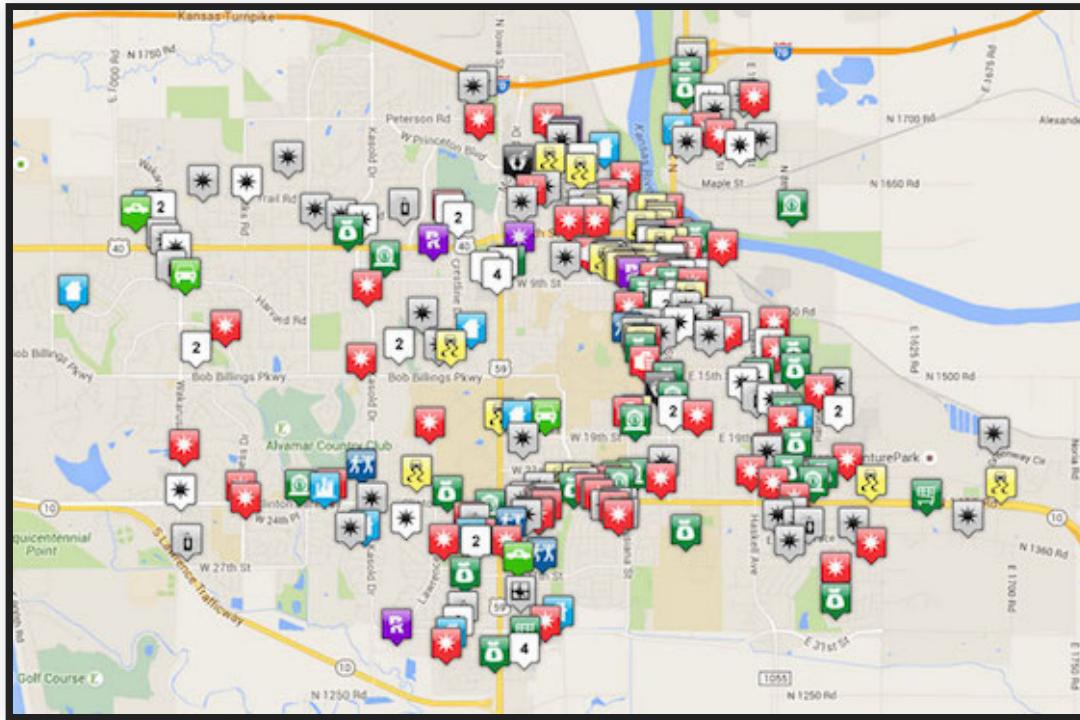
Amazon reportedly scraps internal AI recruiting tool that was biased against women

The secret program penalized applications that contained the word “women’s”

By James Vincent | Oct 10, 2018, 7:09am EDT

SKEWED SAMPLE

Crime prediction for policing strategy



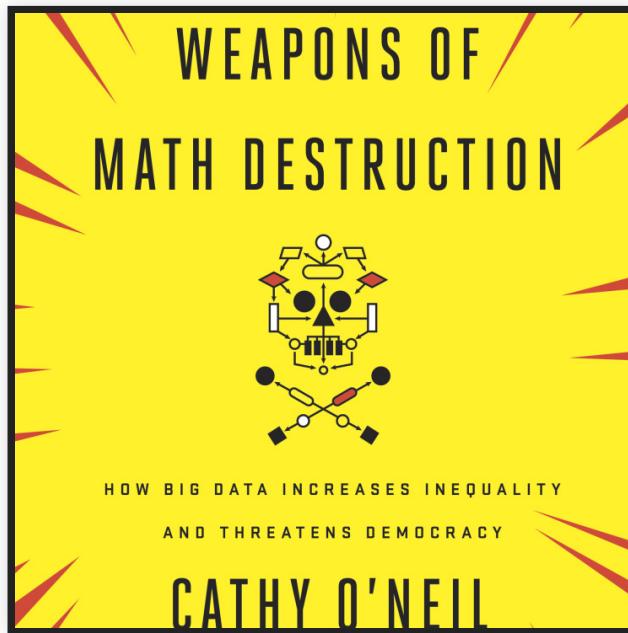
SAMPLE SIZE DISPARITY

Less training data available for certain subpopulations



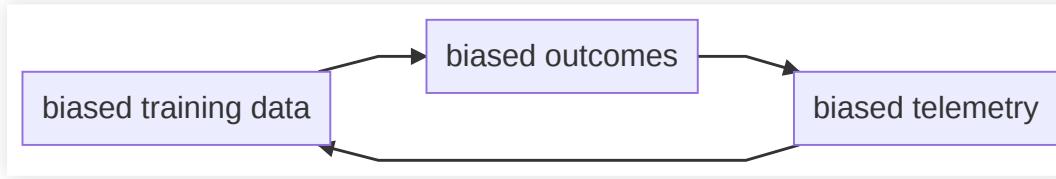
Example: "Shirley Card" used for color calibration

MASSIVE POTENTIAL DAMAGE



O'Neil, Cathy. [Weapons of math destruction: How big data increases inequality and threatens democracy](#). Broadway Books, 2016.

FEEDBACK LOOPS



"Big Data processes codify the past. They do not invent the future. Doing that requires moral imagination, and that's something only humans can provide. "-- Cathy O'Neil in [Weapons of Math Destruction](#)

FAIRNESS: DEFINITIONS AND MEASUREMENTS

Eunsuk Kang

Required reading: Holstein, Kenneth, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach.
["Improving fairness in machine learning systems: What do industry practitioners need?"](#) In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1-16. 2019.

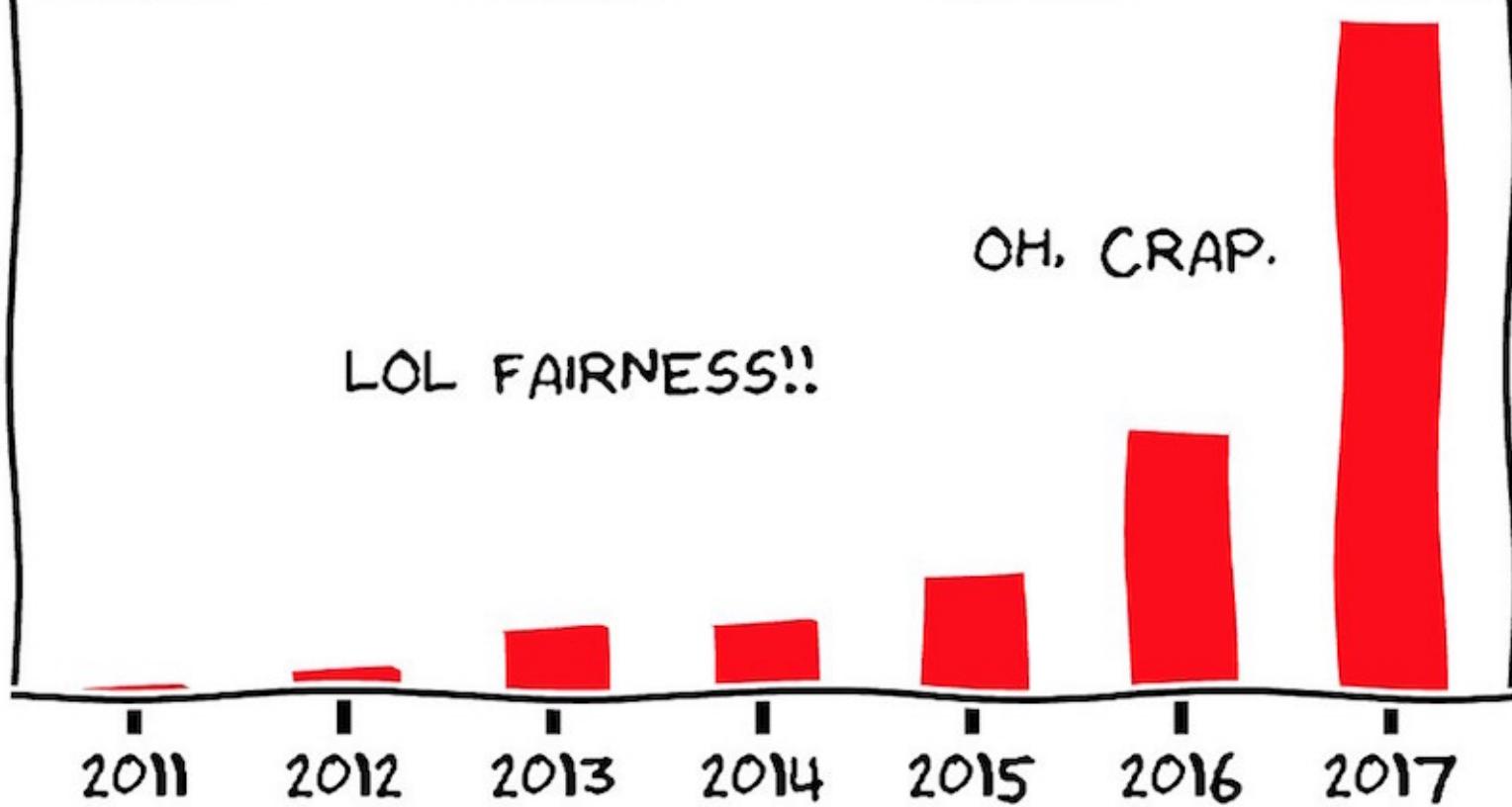
LEARNING GOALS

- Understand different definitions of fairness
- Discuss methods for measuring fairness

FAIRNESS IS STILL AN ACTIVELY STUDIED & DISPUTED CONCEPT!

BRIEF HISTORY OF FAIRNESS IN ML

PAPERS



Source: Mortiz Hardt, <https://fairmlclass.github.io/>

ANTI-CLASSIFICATION



- Ignore/eliminate sensitive attributes from dataset
- Limitations
 - Sensitive attributes may be correlated with other features
 - Some ML tasks need sensitive attributes (e.g., medical diagnosis)

TESTING ANTI-CLASSIFICATION

Straightforward invariant for classifier f and protected attribute p :

$$\forall x. f(x[p \leftarrow 0]) = f(x[p \leftarrow 1])$$

(does not account for correlated attributes)

Test with random input data (see prior lecture on [Automated Random Testing](#)) or
on any test data

Any single inconsistency shows that the protected attribute was used. Can also
report percentage of inconsistencies.

See for example: Galhotra, Sainyam, Yuriy Brun, and Alexandra Meliou. "[Fairness testing: testing software for discrimination](#)." In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, pp. 498-510. 2017.

INDEPENDENCE

(aka *statistical parity, demographic parity, disparate impact, group fairness*)

$$P[R = 1|A = 0] = P[R = 1|A = 1] \text{ or } R \perp A$$

- Acceptance rate (i.e., percentage of positive predictions) must be the same across all groups
- Prediction must be independent of the sensitive attribute
- Example:
 - The predicted rate of recidivism is the same across all races
 - Chance of promotion the same across all genders

EXERCISE: CANCER DIAGNOSIS

Overall Results

True positives (TPs): 16	False positives (FPs): 21
False negatives (FNs): 9	True negatives (TNs): 954

Male Patient Results

True positives (TPs): 3	False positives (FPs): 16
False negatives (FNs): 7	True negatives (TNs): 474

Female Patient Results

True positives (TPs): 13	False positives (FPs): 5
False negatives (FNs): 2	True negatives (TNs): 480

- 1000 data samples (500 male & 500 female patients)
- What's the overall recall & precision?
- Does the model achieve *independence*

CALIBRATION TO ACHIEVE INDEPENDENCE

Select different thresholds for different groups to achieve prediction parity:

$$P[R > t_0 | A = 0] = P[R > t_1 | A = 1]$$

Lowers bar for some groups -- equity, not equality

SEPARATION

$$P[Y' = 1 \mid Y = 0, A = a] = P[Y' = 1 \mid Y = 0, A = b]$$

$$P[Y' = 0 \mid Y = 1, A = a] = P[Y' = 0 \mid Y = 1, A = b]$$

- Also called *equalized odds*
- $Y' \perp A|Y$
 - Prediction must be independent of the sensitive attribute *conditional* on the target variable

Equality



The assumption is that **everyone benefits from the same supports**. This is equal treatment.

Equity



Everyone gets the supports they need (this is the concept of "affirmative action"), thus producing equity.

Justice



All 3 can see the game without supports or accommodations because **the cause(s) of the inequity was addressed**.
The systemic barrier has been removed.



REVIEW OF CRITERIA SO FAR:

Recidivism scenario: Should a person be detained?

- Anti-classification: ?
- Independence: ?
- Separation: ?

CAN WE ACHIEVE FAIRNESS DURING THE LEARNING PROCESS?

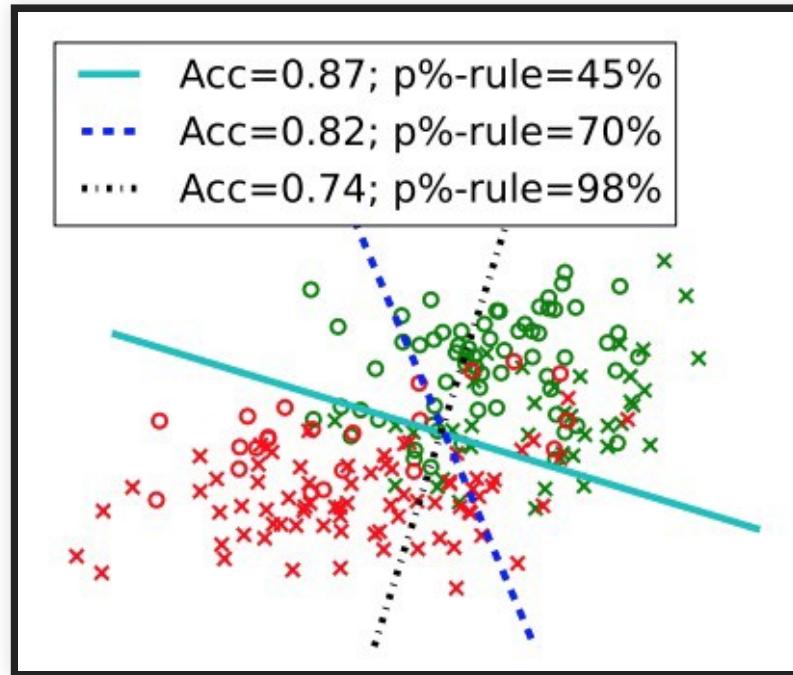
- Data acquisition:
 - Collect additional data if performance is poor on some groups
- Pre-processing:
 - Clean the dataset to reduce correlation between the feature set and sensitive attributes
- Training time constraint
 - ML is a constraint optimization problem (i.e., minimize errors)
 - Impose additional parity constraint into ML

optimization process (as part of the loss function)

- Post-processing
 - Adjust thresholds to achieve a desired fairness metric
- (Still active area of research! Many new techniques published each year)

Training Well-Generalizing Classifiers for Fairness Metrics and Other Data-Dependent Constraints, Cotter et al., (2018).

TRADE-OFFS: ACCURACY VS FAIRNESS



- Fairness constraints possible models
- Fairness constraints often lower accuracy for some group

Fairness Constraints: Mechanisms for Fair Classification, Zafar et al., AISTATS (2017).

FAIRNESS: BEYOND MODEL

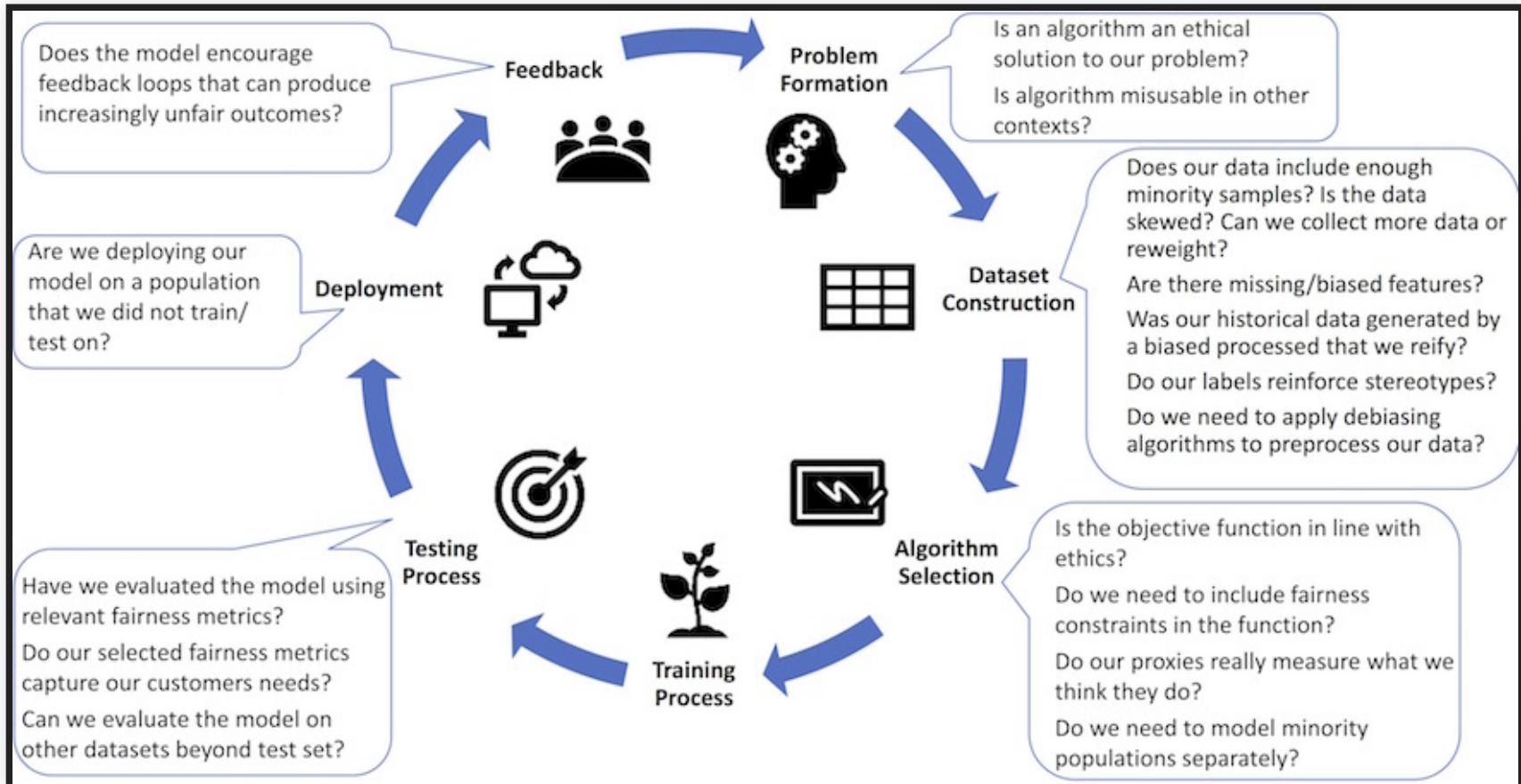
Eunsuk Kang

Required reading: Os Keyes, Jevan Hutson, Meredith Durbin. [A Mulching Proposal: Analysing and Improving an Algorithmic System for Turning the Elderly into High-Nutrient Slurry](#). CHI Extended Abstracts, 2019.

LEARNING GOALS

- Consider achieving fairness in AI-based systems as an activity throughout the entire development cycle
- Understand the role of requirements engineering in selecting ML fairness criteria
- Understand the process of constructing datasets for fairness
- Consider the potential impact of feedback loops on AI-based systems and need for continuous monitoring

FAIRNESS MUST BE CONSIDERED THROUGHOUT THE ML LIFECYCLE!



Fairness-aware Machine Learning, Bennett et al., WSDM Tutorial (2019).

PRACTITIONER CHALLENGES

- Fairness is a system-level property
 - consider goals, user interaction design, data collection, monitoring, model interaction (properties of a single model may not matter much)
- Fairness-aware data collection, fairness testing for training data
- Identifying blind spots
 - Proactive vs reactive
 - Team bias and (domain-specific) checklists
- Fairness auditing processes and tools
- Diagnosis and debugging (outlier or systemic problem? causes?)
- Guiding interventions (adjust goals? more data? side effects? chasing mistakes? redesign?)
- Assessing human bias of humans in the loop

Holstein, Kenneth, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. "[Improving fairness in machine learning systems: What do industry practitioners need?](#)" In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1-16. 2019.

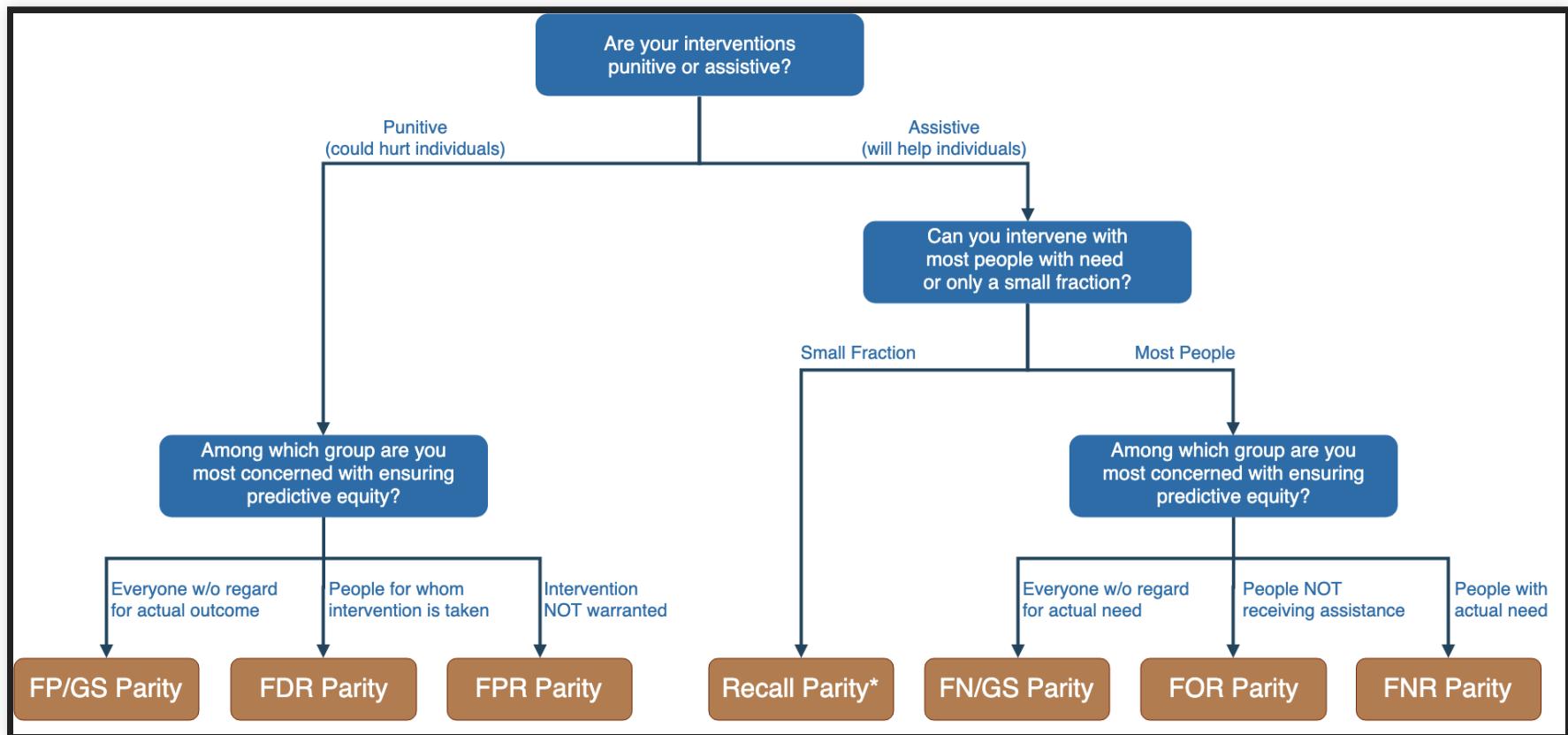
REQUIREMENTS FOR FAIR ML SYSTEMS

- Identify requirements (REQ) over the environment
 - What types of harm can be caused by biased decisions?
 - Who are stakeholders? Which population groups can be harmed?
 - Are we trying to achieve equality vs. equity?
 - What are legal requirements to consider?
- Define the interface between the environment & machine (ML)
 - What data will be sensed/measured by AI? Potential biases?
 - What types of decisions will the system make? Punitive or assistive?
- Identify the environmental assumptions (ENV)
 - Adversarial? Misuse? Unfair (dis-)advantages?
 - Population distributions?

TYPE OF DECISION & POSSIBLE HARM

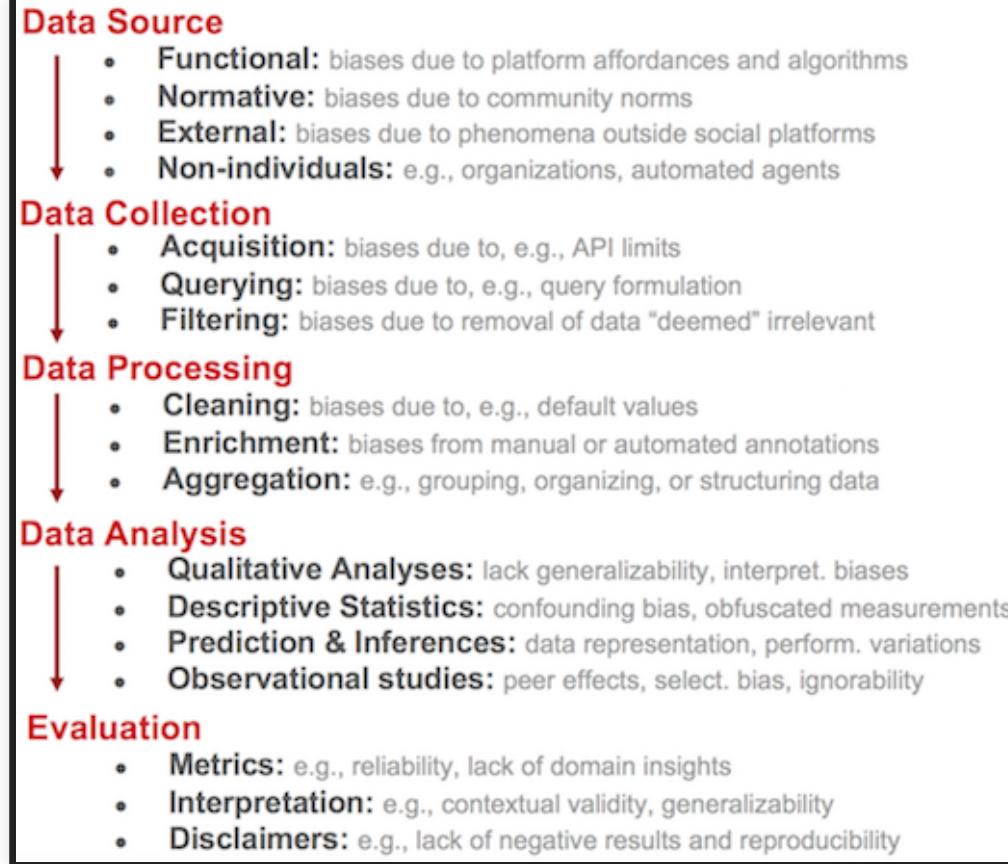
- If decision is *punitive* in nature:
 - e.g. decide whom to deny bail based on risk of recidivism
 - Harm is caused when a protected group is given an unwarranted penalty
 - Heuristic: Use a fairness metric (separation) based on **false positive rate**
- If decision is *assistive* in nature:
 - e.g., decide who should receive a loan or a food subsidy
 - Harm is caused when a group in need is incorrectly denied assistance
 - Heuristic: Use a fairness metric based on **false negative rate**

FAIRNESS TREE



For details on other types of fairness metrics, see:
<https://textbook.coleridgeinitiative.org/chap-bias.html>

DATA BIAS



- A **systematic distortion** in data that compromises its use for a task
- Bias can be introduced at any stage of the data pipeline!

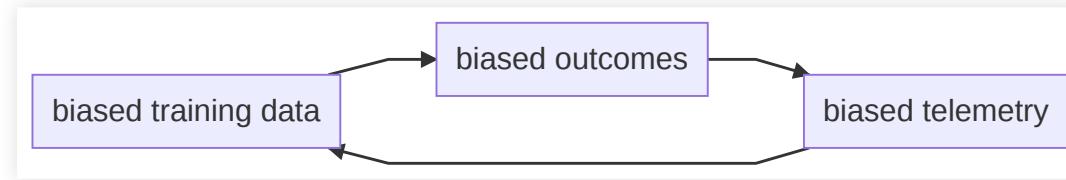
DATA SHEETS

Demographic Characteristic	Value
Percentage of female subjects	22.5%
Percentage of male subjects	77.5%
Percentage of White subjects	83.5%
Percentage of Black subjects	8.47%
Percentage of Asian subjects	8.03%
Percentage of people between 0-20 years old	1.57%
Percentage of people between 21-40 years old	31.63%
Percentage of people between 41-60 years old	45.58%
Percentage of people over 61 years old	21.2%

- A process for documenting datasets
- Based on common practice in the electronics industry, medicine
- Purpose, provenance, creation, composition, distribution: Does the dataset relate to people? Does the dataset identify any subpopulations?

Datasheets for Dataset, Gebru et al., (2019).

MONITORING AND AUDITING: FEEDBACK LOOPS



*"Big Data processes codify the past. They do not invent the future. Doing that requires moral imagination, and that's something only humans can provide. "-- Cathy O'Neil in
[Weapons of Math Destruction](#)*

FAIRNESS CHECKLIST

Envision

Consider doing the following items in moments like:

- Envisioning meetings
- Pre-mortem screenings
- Product greenlighting meetings

1.1 Envision system and scrutinize system vision

1.1.a Envision system and its role in society, considering:

- System purpose, including key objectives and intended uses or applications
 - Consider whether the system should exist and, if so, whether the system should use AI
 - Sensitive, premature, dual, or adversarial uses or applications
 - Consider whether the system will impact human rights
 - Consider whether these uses or applications should be prohibited
 - Expected deployment contexts (e.g., geographic regions, time periods)
 - Expected stakeholders (e.g., people who will make decisions about system adoption, people who will use the system, people who will be directly or indirectly affected by the system, society), including demographic groups (e.g., by race, gender, age, disability status, skin tone, and their intersections)
 - Expected benefits for each stakeholder group, including demographic groups
 - Relevant regulations, standards, guidelines, policies, etc.
- 1.1.b Scrutinize resulting system vision for potential fairness-related harms to stakeholder groups, considering:
- Types of harm (e.g., allocation, quality of service, stereotyping, denigration, over- or underrepresentation)

HOMEWORK 6: FAIRNESS

(credit scoring + recommendation, model + system)

INTERPRETABILITY AND EXPLAINABILITY

Christian Kaestner

Required reading: □ Data Skeptic Podcast Episode “[Black Boxes are not Required](#)” with Cynthia Rudin (32min) or □ Rudin, Cynthia. "[Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.](#)" Nature Machine Intelligence 1, no. 5 (2019): 206-215.

Recommended supplementary reading: □ Christoph Molnar. "[Interpretable Machine Learning: A Guide for Making Black Box Models Explainable](#)." 2019

LEARNING GOALS

- Understand the importance of and use cases for interpretability
- Explain the tradeoffs between inherently interpretable models and post-hoc explanations
- Measure interpretability of a model
- Select and apply techniques to debug/provide explanations for data, models and model predictions
- Eventuate when to use interpretable models rather than ex-post explanations

DETECTING ANOMALOUS COMMITS

The screenshot shows a GitHub commit page for a pull request. The commit message is titled "v8: don't busy loop in cpu profiler thread". It describes a change to reduce CPU overhead by replacing `sched_yield()` with `nanosleep()`. The commit notes that before this, the thread would effectively be a busy loop. It includes links to a PR and an issue, and a review by Trevor Norris.

Below the commit message, it shows the author as "bnoordhuis" from 2014-11-27, with a parent commit "fe20196" and a commit hash "6ebd85e10535dfa9181842fe73834e51d4d3e6c". A "Show Details" button is present.

At the bottom of the commit page, there is a note: "Use 'Show details' button to show commit details."

A blue header bar labeled "ADDITIONAL INFORMATION FOR THIS COMMIT" contains a bulleted list of anomalies:

- Changes were committed at 6am UTC -- **bnoordhuis rarely** commits around that time. (fewer than 0.7% of all commits by bnoordhuis are around that time)
- .gyp files were changed -- such files are **rarely** changed in this repository. (fewer than 2% of all file types changed)
- .cc and .gyp files were changed in the same commit -- this combination of files is **rarely changed together**. (in fewer than 2% of all commits)
- .cc and .gyp files were changed in the same commit -- this combination of files is **rarely changed together** by **bnoordhuis**. (in fewer than 3% of all commits by bnoordhuis)
- .gyp files were changed -- such files are **rarely** changed by **bnoordhuis**. (fewer than 3% of all file types changed by bnoordhuis)

Goyal, Raman, Gabriel Ferreira, Christian Kästner, and James Herbsleb.
"Identifying unusual commits on GitHub." Journal of Software: Evolution and Process 30, no. 1 (2018): e1893.

IS THIS RECIDIVISM MODEL FAIR?

```
IF age between 18-20 and sex is male THEN predict arrest  
ELSE  
IF age between 21-23 and 2-3 prior offenses THEN predict arrest  
ELSE  
IF more than three priors THEN predict arrest  
ELSE predict no arrest
```

Rudin, Cynthia. "[Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.](#)" Nature Machine Intelligence 1, no. 5 (2019): 206-215.

WHAT FACTORS GO INTO PREDICTING STROKE RISK?

1. <i>Congestive Heart Failure</i>	1 point	...
2. <i>Hypertension</i>	1 point	+
3. <i>Age ≥ 75</i>	1 point	+
4. <i>Diabetes Mellitus</i>	1 point	+
5. <i>Prior Stroke or Transient Ischemic Attack</i>	2 points	+
ADD POINTS FROM ROWS 1–5	SCORE	= ...

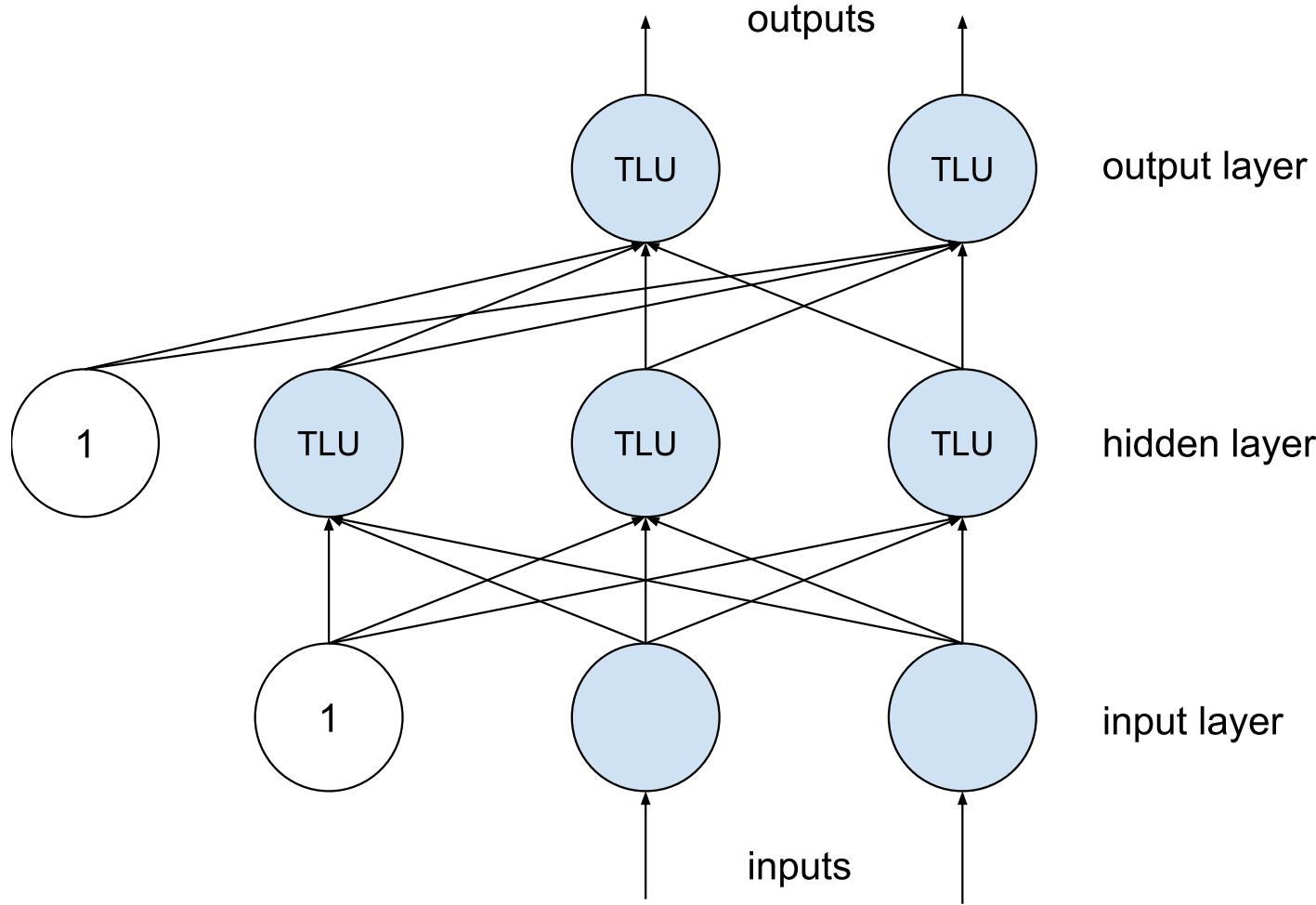
SCORE	0	1	2	3	4	5	6
STROKE RISK	1.9%	2.8%	4.0%	5.9%	8.5%	12.5%	18.2%

Rudin, Cynthia, and Berk Ustun. "Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice." *Interfaces* 48, no. 5 (2018): 449-466.

IS THERE AN ACTUAL PROBLEM? HOW TO FIND OUT?

Tweet

WHAT'S HAPPENING HERE?



LEGAL REQUIREMENTS

The European Union General Data Protection Regulation extends the automated decision-making rights in the 1995 Data Protection Directive to provide a legally disputed form of a right to an explanation: "[the data subject should have] the right ... to obtain an explanation of the decision reached"

US Equal Credit Opportunity Act requires to notify applicants of action taken with specific reasons: "The statement of reasons for adverse action required by paragraph (a)(2)(i) of this section must be specific and indicate the principal reason(s) for the adverse action."

See also https://en.wikipedia.org/wiki/Right_to_explanation

DEBUGGING

- Why did the system make a wrong prediction in this case?
- What does it actually learn?
- What kind of data would make it better?
- How reliable/robust is it?
- How much does the second model rely on the outputs of the first?
- Understanding edge cases

CURIOSITY, LEARNING, DISCOVERY, SCIENCE

- What drove our past hiring decisions? Who gets promoted around here?
- What factors influence cancer risk? Recidivism?
- What influences demand for bike rentals?
- Which organizations are successful at raising donations and why?

INTERPRETABILITY DEFINITIONS

Interpretability is the degree to which a human can understand the cause of a decision

Interpretability is the degree to which a human can consistently predict the model's result.

(No mathematical definition)

GOOD EXPLANATIONS ARE CONTRASTIVE

Counterfactuals. *Why this, rather than a different prediction?*

Your loan application has been declined. If your savings account had had more than \$100 your loan application would be accepted.

Partial explanations often sufficient in practice if contrastive

INHERENTLY INTERPRETABLE MODELS: SPARSE LINEAR MODELS

$$f(x) = \alpha + \beta_1 x_1 + \dots + \beta_n x_n$$

Truthful explanations, easy to understand for humans

Easy to derive contrastive explanation and feature importance

Requires feature selection/regularization to minimize to few important features
(e.g. Lasso); possibly restricting possible parameter values

1. <i>Congestive Heart Failure</i>	1 point	...					
2. <i>Hypertension</i>	1 point	+					
3. <i>Age ≥ 75</i>	1 point	+					
4. <i>Diabetes Mellitus</i>	1 point	+					
5. <i>Prior Stroke or Transient Ischemic Attack</i>	2 points	+					
ADD POINTS FROM ROWS 1–5		SCORE					
		= ...					
SCORE	0	1	2	3	4	5	6
STROKE RISK	1.9%	2.8%	4.0%	5.9%	8.5%	12.5%	18.2%

INHERENTLY INTERPRETABLE MODELS: DECISION TREES

Easy to interpret up to a size

Possible to derive counterfactuals and feature importance

Unstable with small changes to training data

```
IF age between 18-20 and sex is male THEN predict arrest  
ELSE IF age between 21-23 and 2-3 prior offenses THEN predict ar  
ELSE IF more than three priors THEN predict arrest  
ELSE predict no arrest
```

POST-HOC EXPLANATIONS OF BLACK-BOX MODELS

(large research field, many approaches, much recent research)

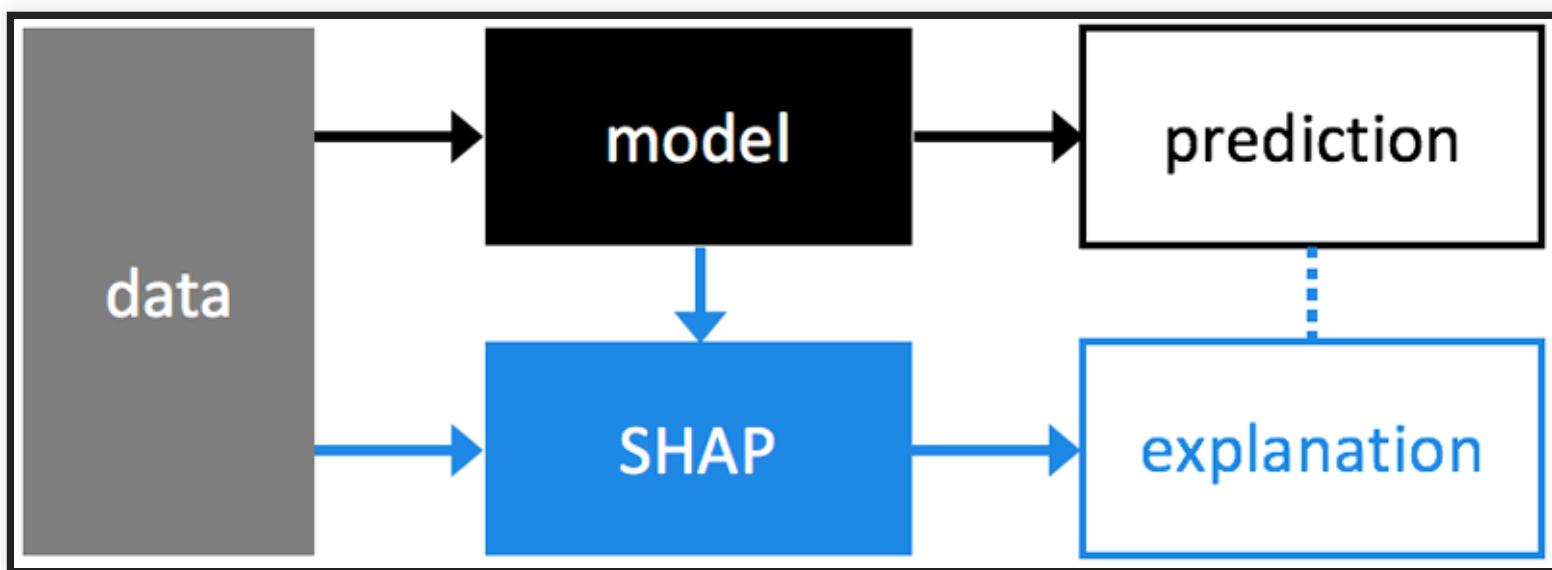


Figure: Lundberg, Scott M., and Su-In Lee. [A unified approach to interpreting model predictions](#). Advances in Neural Information Processing Systems. 2017.

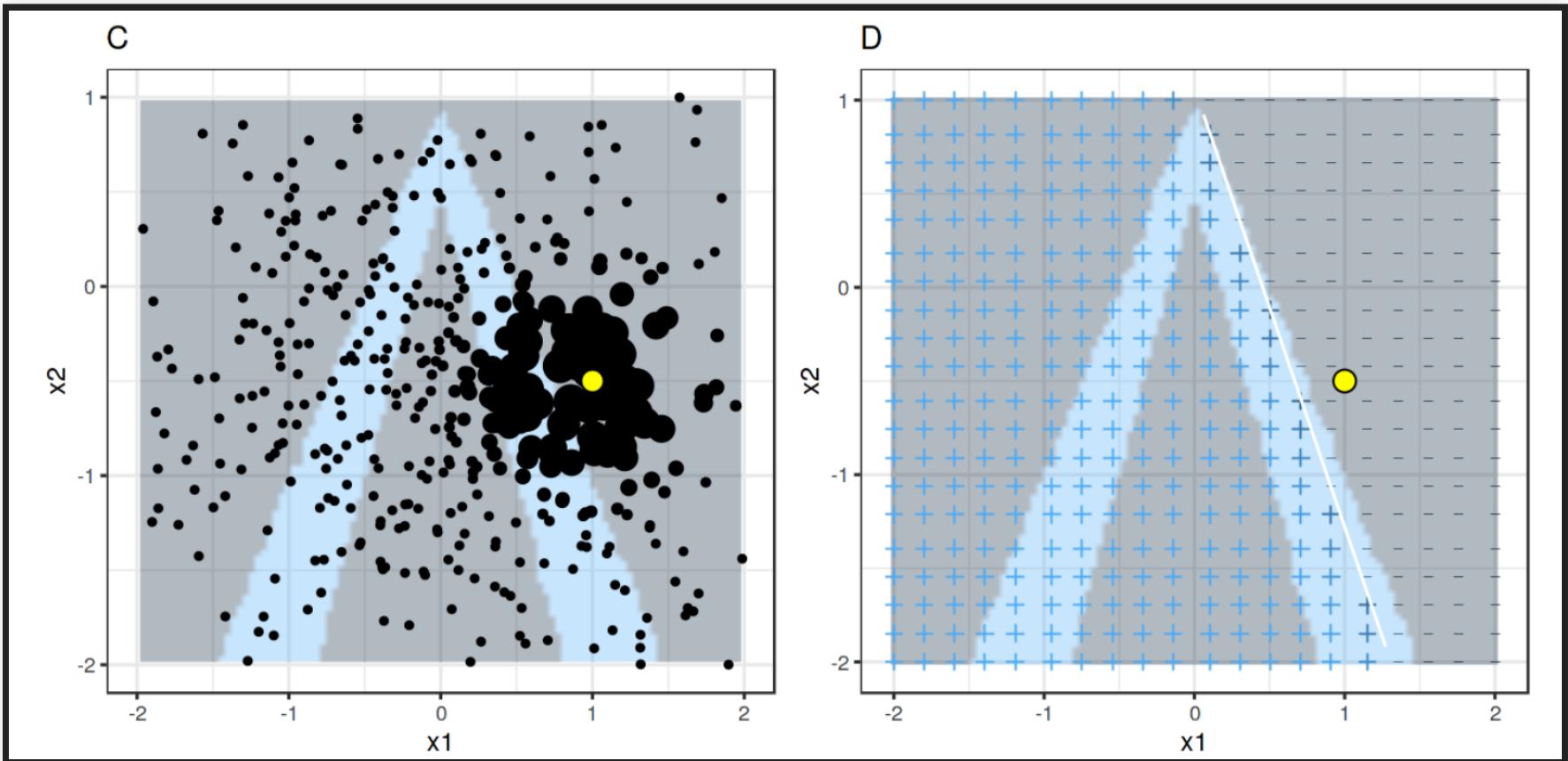
GLOBAL SURROGATES

1. Select dataset X (previous training set or new dataset from same distribution)
2. Collect model predictions for every value ($y_i = f(x_i)$)
3. Train inherently interpretable model g on (X,Y)
4. Interpret surrogate model g

Can measure how well g fits f with common model quality measures, typically R^2

Advantages? Disadvantages?

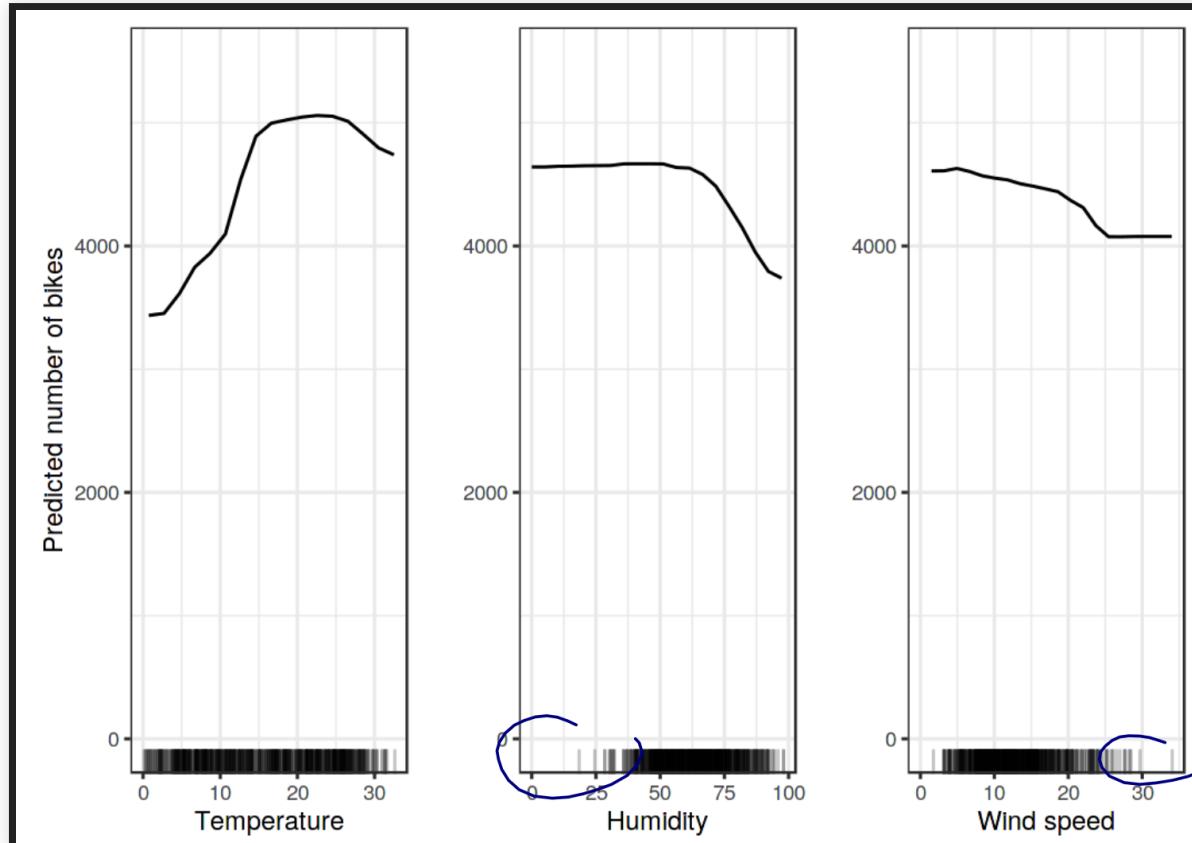
LIME EXAMPLE



Source: Christoph Molnar. "[Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.](#)"
2019

PARTIAL DEPENDENCE PLOT EXAMPLE

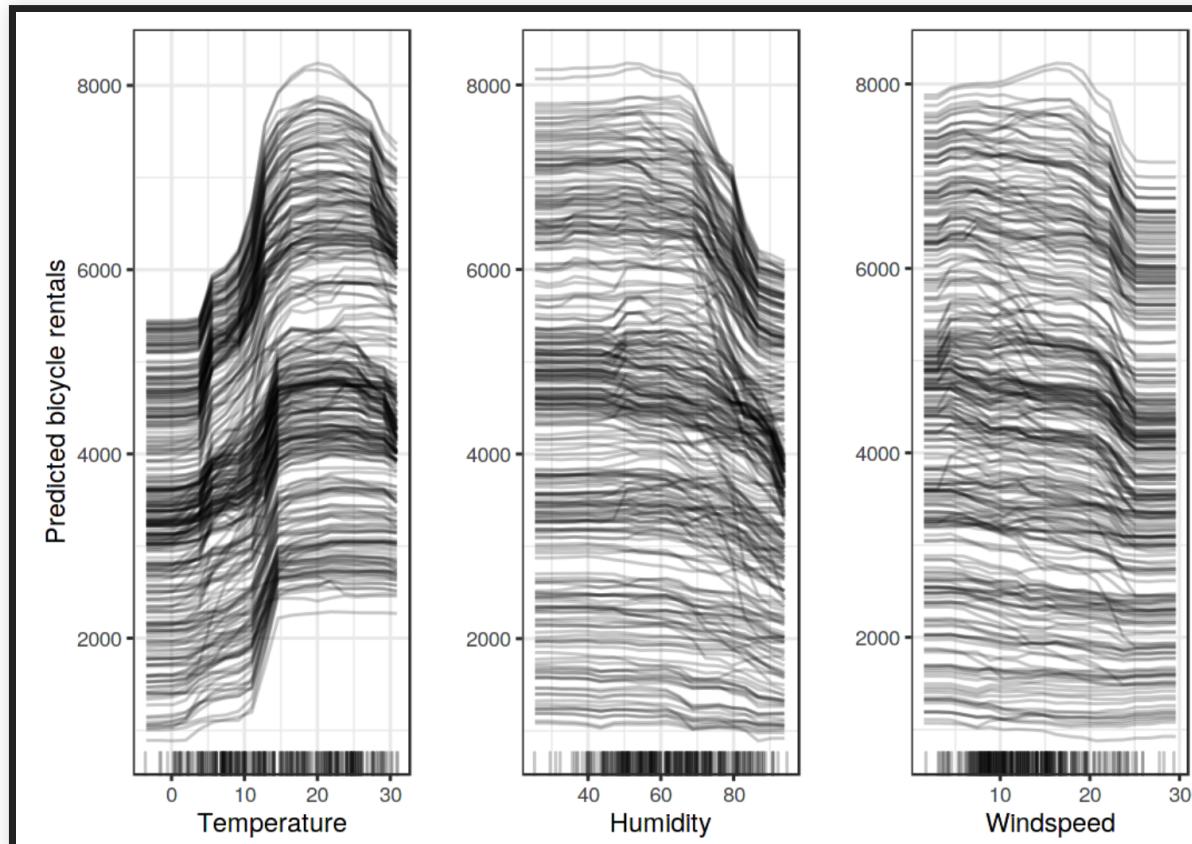
Bike rental in DC



Source: Christoph Molnar. "[Interpretable Machine Learning](#)." 2019

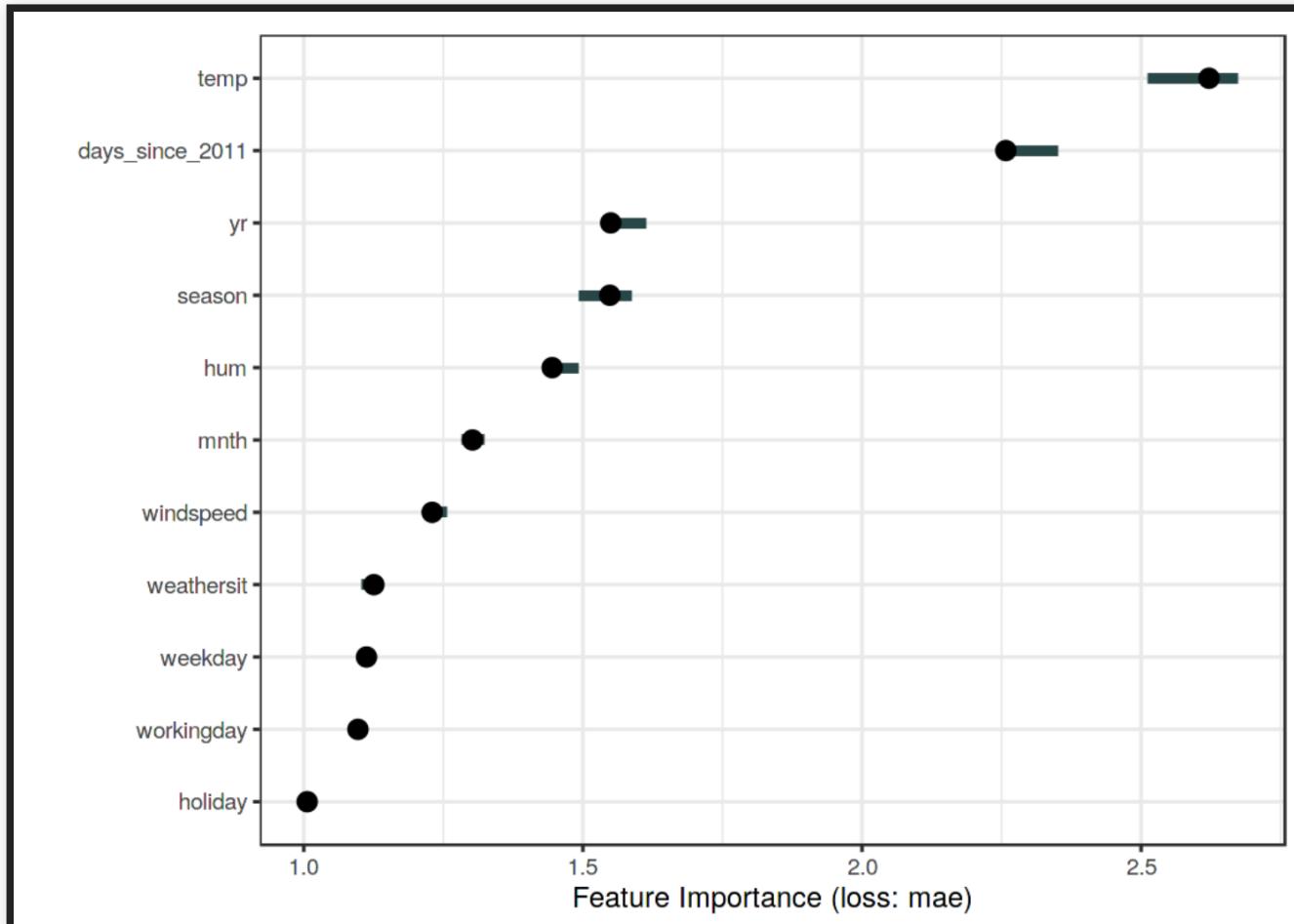
INDIVIDUAL CONDITIONAL EXPECTATION (ICE)

Similar to PDP, but not averaged; may provide insights into interactions



Source: Christoph Molnar. "[Interpretable Machine Learning](#)." 2019

FEATURE IMPORTANCE EXAMPLE



Source: Christoph Molnar. "[Interpretable Machine Learning](#)." 2019

EXAMPLE: ANCHORS

	If	Predict
adult	No capital gain or loss, never married	$\leq 50K$
	Country is US, married, work hours > 45	$> 50K$
rcdvv	No priors, no prison violations and crime not against property	Not rearrested
	Male, black, 1 to 5 priors, not married, and crime not against property	Re-arrested
lending	FICO score ≤ 649	Bad Loan
	$649 \leq$ FICO score ≤ 699 and $\$5,400 \leq$ loan amount $\leq \$10,000$	Good Loan

Source: Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "[Anchors: High-precision model-agnostic explanations](#)." In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

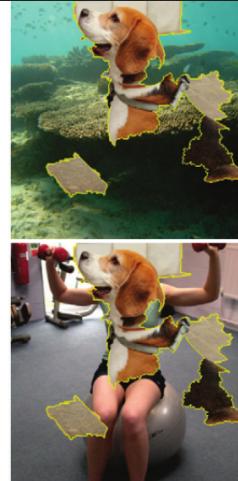
EXAMPLE: ANCHORS



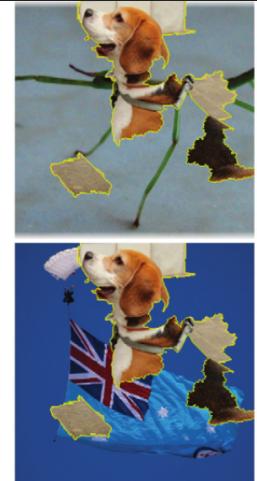
(a) Original image



(b) Anchor for “beagle”



(c) Images where Inception predicts $P(\text{beagle}) > 90\%$



Source: Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "[Anchors: High-precision model-agnostic explanations](#)." In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

COUNTERFACTUAL EXPLANATIONS

if X had not occurred, Y would not have happened

Your loan application has been declined. If your savings account had had more than \$100 your loan application would be accepted.

-> Smallest change to feature values that result in given output

MULTIPLE COUNTERFACTUALS

Often long or multiple explanations

Your loan application has been declined. If your savings account ...

Your loan application has been declined. If you lived in

...

Report all or select "best" (e.g. shortest, most actionable, likely values)

(Rashomon effect)



GAMING/ATTACKING THE MODEL WITH EXPLANATIONS?

Does providing an explanation allow customers to 'hack' the system?

- Loan applications?
- Apple FaceID?
- Recidivism?
- Auto grading?
- Cancer diagnosis?
- Spam detection?



GAMING THE MODEL WITH EXPLANATIONS?

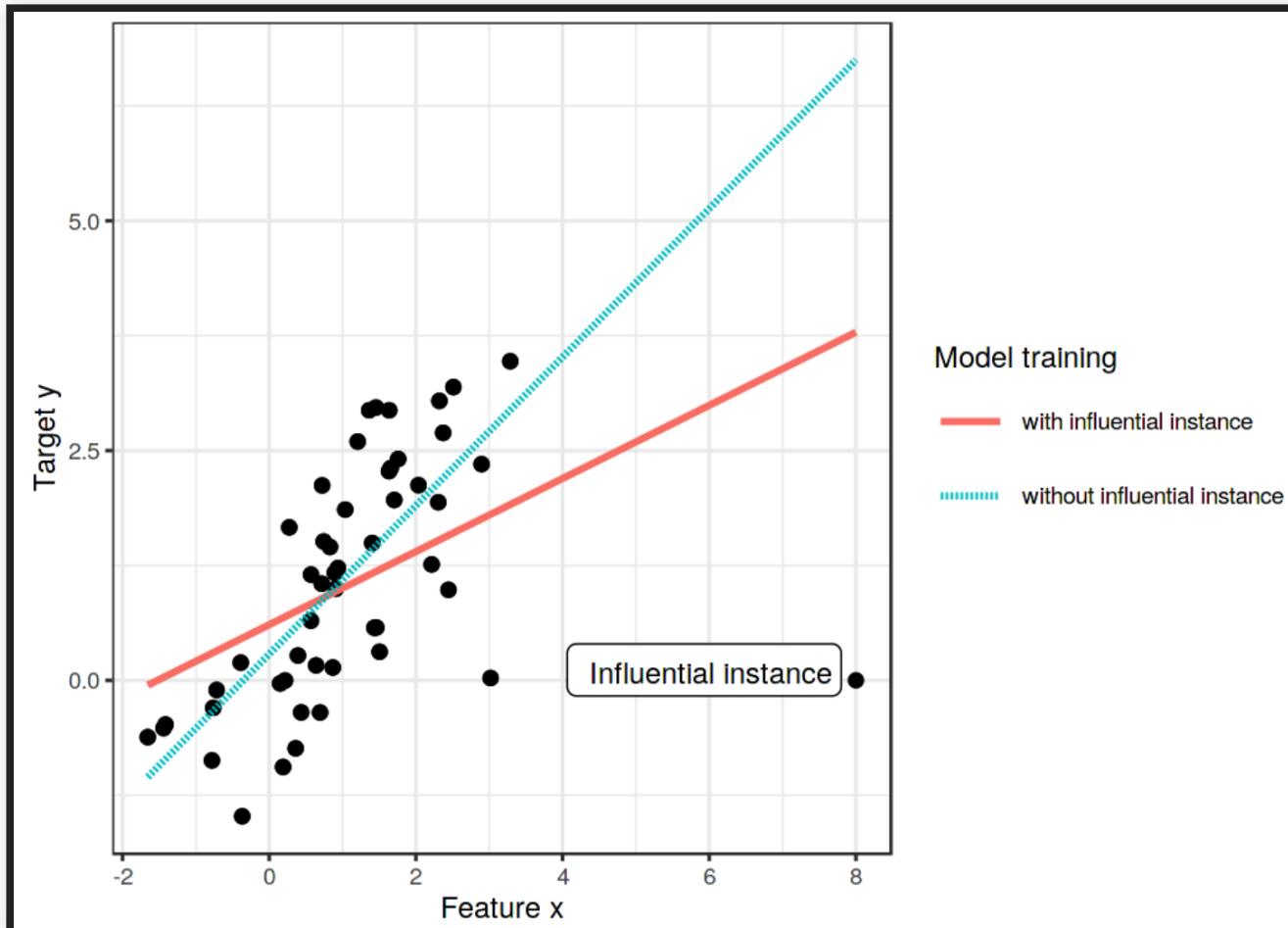


EXAMPLE: PROTOTYPES AND CRITICISMS



Source: Christoph Molnar. "[Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.](#)"
2019

EXAMPLE: INFLUENTIAL INSTANCE



Source: Christoph Molnar. "[Interpretable Machine Learning](#)." 2019

**"STOP EXPLAINING BLACK
BOX MACHINE LEARNING
MODELS FOR HIGH STAKES
DECISIONS AND USE
INTERPRETABLE MODELS
INSTEAD."**

Cynthia Rudin (32min) or [Cynthia Rudin](#), Cynthia. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." *Nature Machine Intelligence* 1, no. 5 (2019): 206-215.

Microsoft AI principles

We put our responsible AI principles into practice through the Office of Responsible AI (ORA) and the AI, Ethics, and Effects in Engineering and Research (Aether) Committee. The Aether Committee advises our leadership on the challenges and opportunities presented by AI innovations. ORA sets our rules and governance processes, working closely with teams across the company to enable the effort.

[Learn more about our approach >](#)

Fairness

AI systems should treat all people fairly

[▷ Play video on fairness](#)

Reliability & Safety

AI systems should perform reliably and safely

[▷ Play video on reliability](#)

Privacy & Security

AI systems should be secure and respect privacy

[▷ Play video on privacy](#)

Inclusiveness

AI systems should empower everyone and engage people

[▷ Play video on inclusiveness](#)

Transparency

AI systems should be understandable

[▷ Play video on transparency](#)

Accountability

People should be accountable for AI systems

[▷ Play video on accountability](#)

4,576 views | Mar 1, 2020, 01:00am EST

This Is The Year Of AI Regulations



Kathleen Walch Contributor

COGNITIVE WORLD Contributor Group ⓘ

AI

-
- f The world of artificial intelligence is constantly evolving, and certainly so is the legal and regulatory environment

ALGORITHMIC TRANSPARENCY

Guest lecture by Motahhare Eslami

VERSIONING, PROVENANCE, AND REPRODUCABILITY

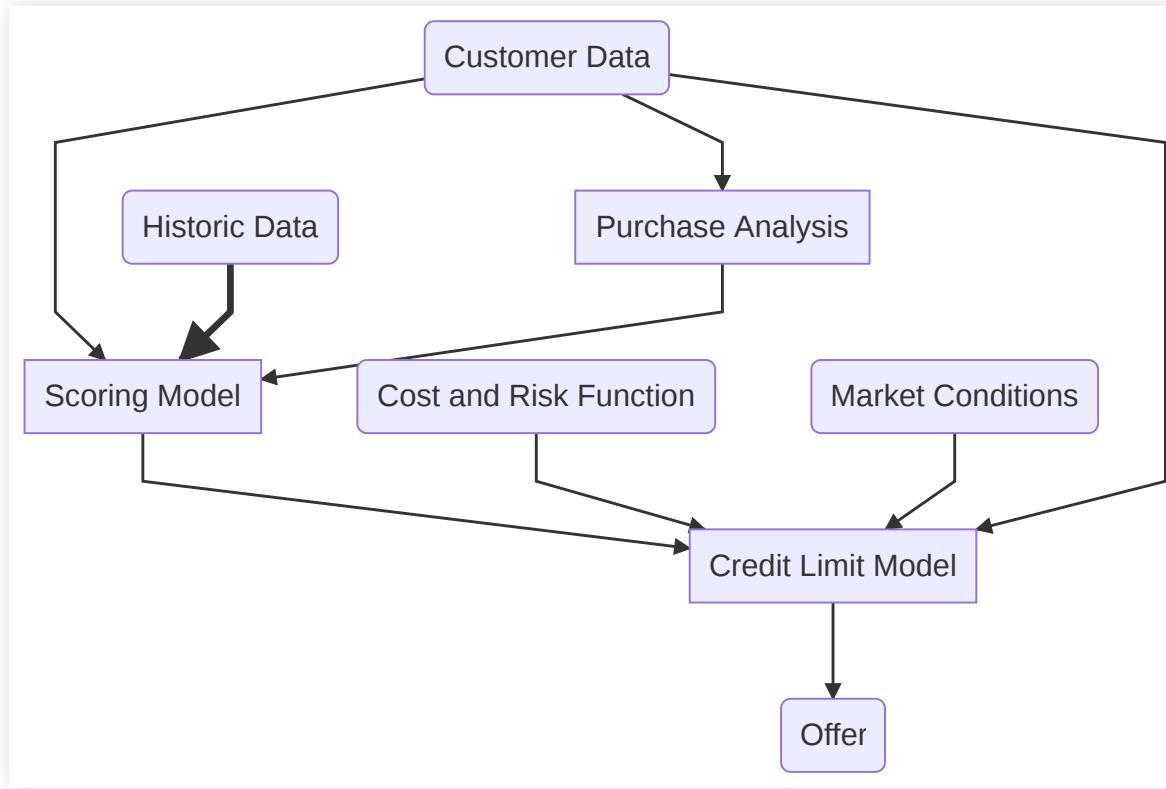
Christian Kaestner

Required reading: □ Halevy, Alon, Flip Korn, Natalya F. Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. [Goods: Organizing google's datasets](#). In Proceedings of the 2016 International Conference

LEARNING GOALS

- Judge the importance of data provenance, reproducibility and explainability for a given system
- Create documentation for data dependencies and provenance in a given system
- Propose versioning strategies for data and models
- Design and test systems for reproducibility

Tweet



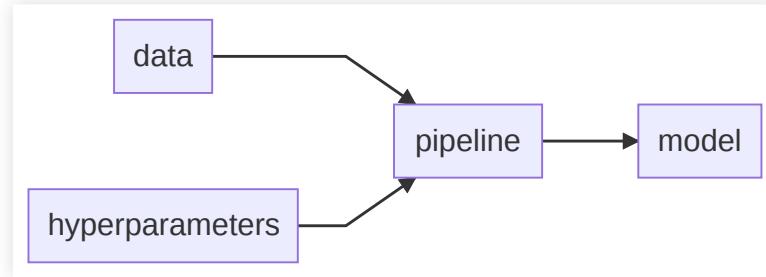
DATA PROVENANCE

- Track origin of all data
 - Collected where?
 - Modified by whom, when, why?
 - Extracted from what other data or model or algorithm?
- ML models often based on data driven from many sources through many steps, including other models

VERSIONING DATASETS

- Store copies of entire datasets (like Git)
- Store deltas between datasets (like Mercurial)
- Offsets in append-only database (like Kafka offset)
- History of individual database records (e.g. S3 bucket versions)
 - some databases specifically track provenance (who has changed what entry when and how)
 - specialized data science tools eg [Hangar](#) for tensor data
- Version pipeline to recreate derived datasets ("views", different formats)
 - e.g. version data before or after cleaning?
- Often in cloud storage, distributed
- Checksums often used to uniquely identify versions
- Version also metadata

VERSIONING PIPELINES



PROJECT M3: MONITORING AND CONTINUOUS DEPLOYMENT

(containization, monitoring, canary releases, provenance)

SECURITY

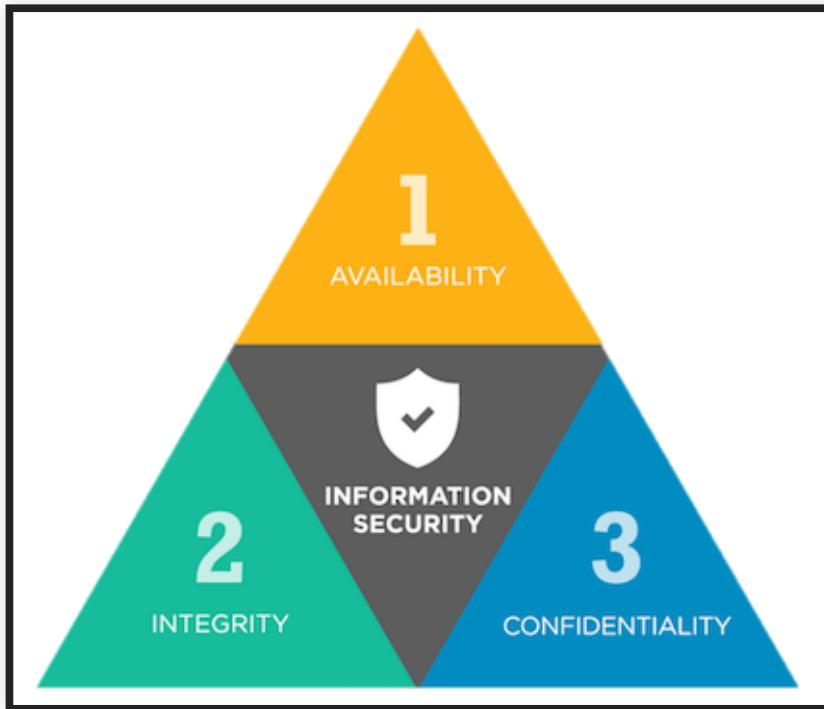
Eunsuk Kang

Required reading: *Building Intelligent Systems: A Guide to Machine Learning Engineering*, G. Hulten (2018), Chapter 25: Adversaries and Abuse. *The Top 10 Risks of Machine Learning Security*, G. McGraw et al., IEEE Computer (2020).

LEARNING GOALS

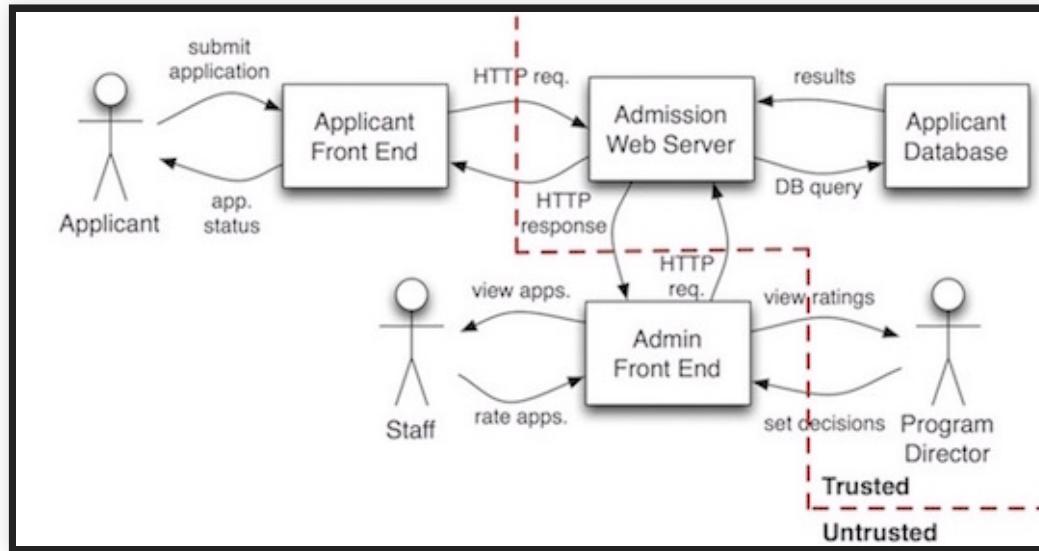
- Explain key concerns in security (in general and with regard to ML models)
- Analyze a system with regard to attacker goals, attack surface, attacker capabilities
- Describe common attacks against ML models, including poisoning and evasion attacks
- Understand design opportunities to address security threats at the system level
- Identify security requirements with threat modeling
- Apply key design principles for secure system design

SECURITY REQUIREMENTS



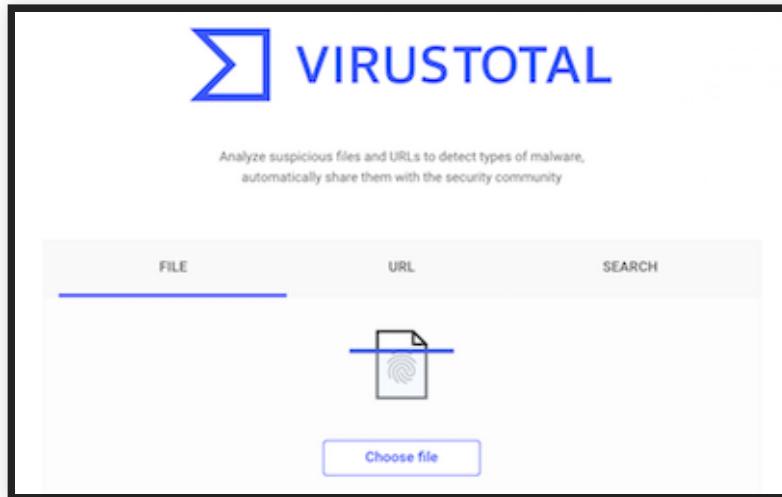
- "CIA triad" of information security
- **Confidentiality:** Sensitive data must be accessed by authorized users only
- **Integrity:** Sensitive data must be modifiable by authorized users only
- **Availability:** Critical services must be available when needed by clients

ATTACKER CAPABILITY



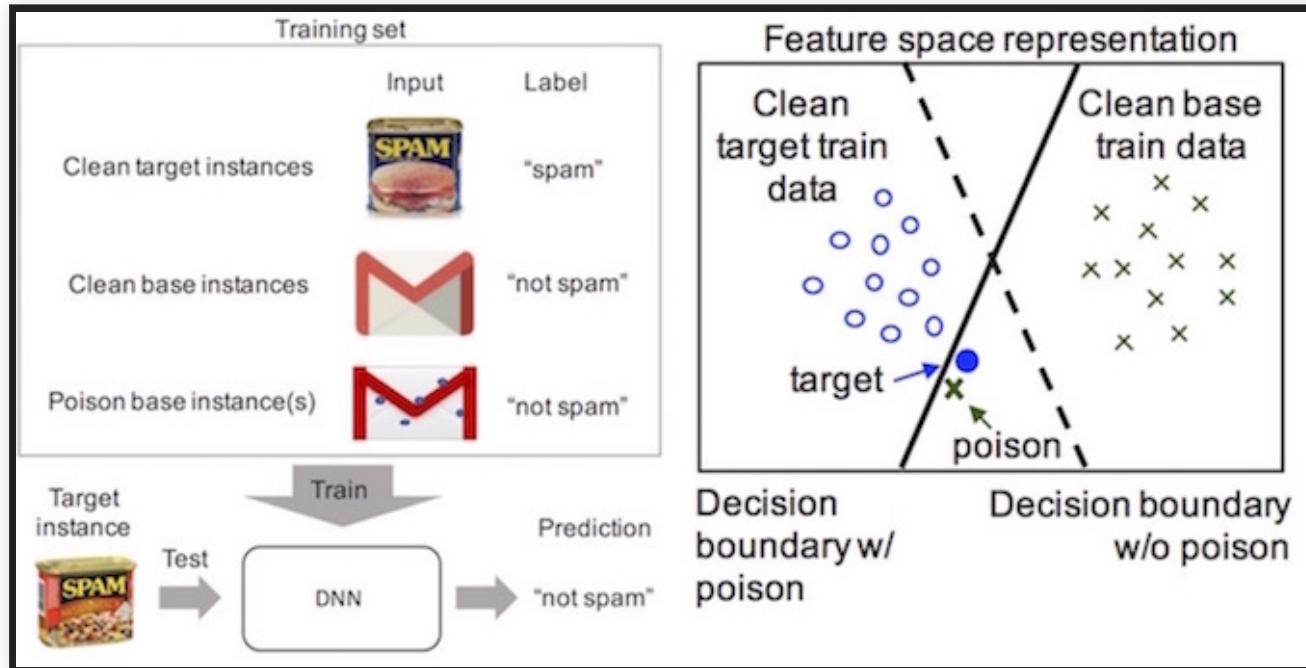
- What are the attacker's actions?
 - Depends on system boundary & its exposed interfaces
 - Use an architecture diagram to identify attack surface & actions
- Example: College admission
 - Physical: Break into building & access server
 - Cyber: Send malicious HTTP requests for SQL injection, DoS attack
 - Social: Send phishing e-mail, bribe an insider for access

POISONING ATTACK: AVAILABILITY



- Availability: Inject mislabeled training data to damage model quality
 - 3% poisoning => 11% decrease in accuracy (Steinhardt, 2017)
- Attacker must have some access to the training set
 - e.g., models trained on public data set (e.g., ImageNet)
- Example: Anti-virus (AV) scanner
 - Online platform for submission of potentially malicious code
 - Some AV company (allegedly) poisoned competitor's model

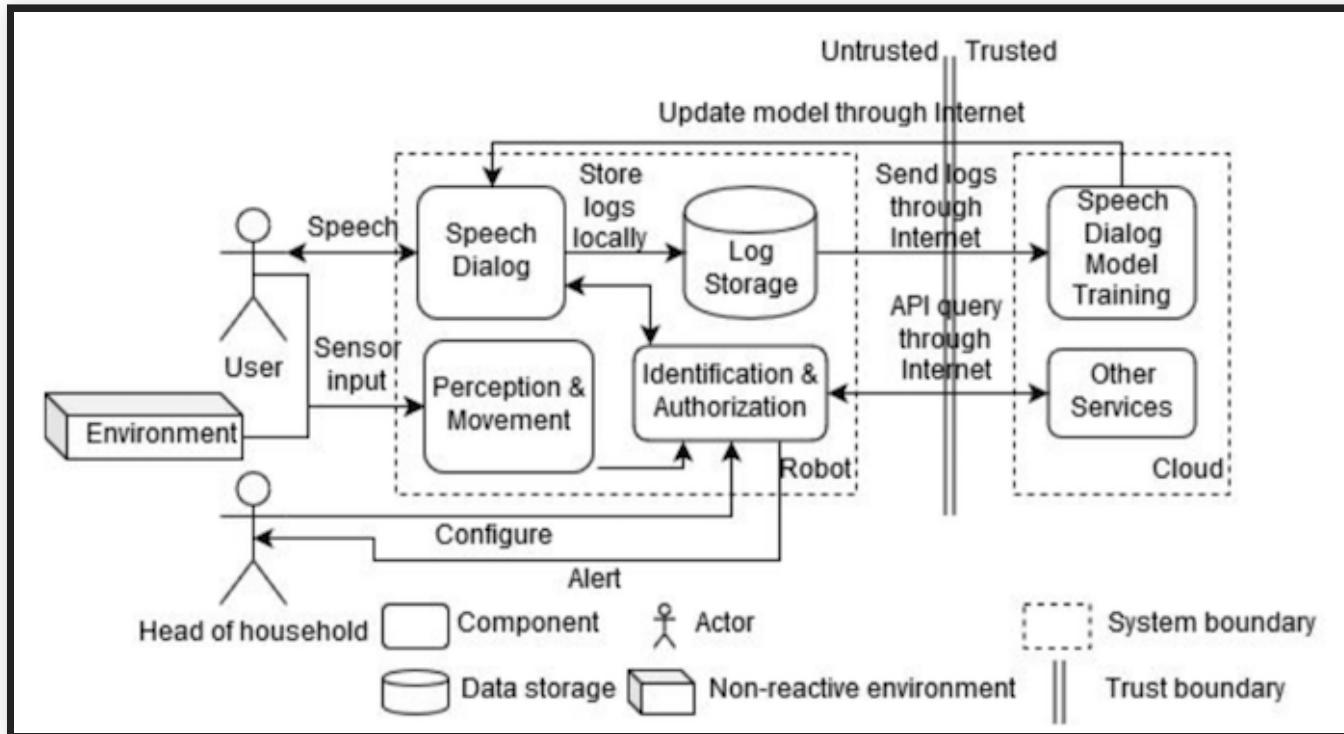
POISONING ATTACK: INTEGRITY



- Insert training data with seemingly correct labels
- More targeted than availability attacks
 - Cause misclassification from one specific class to another

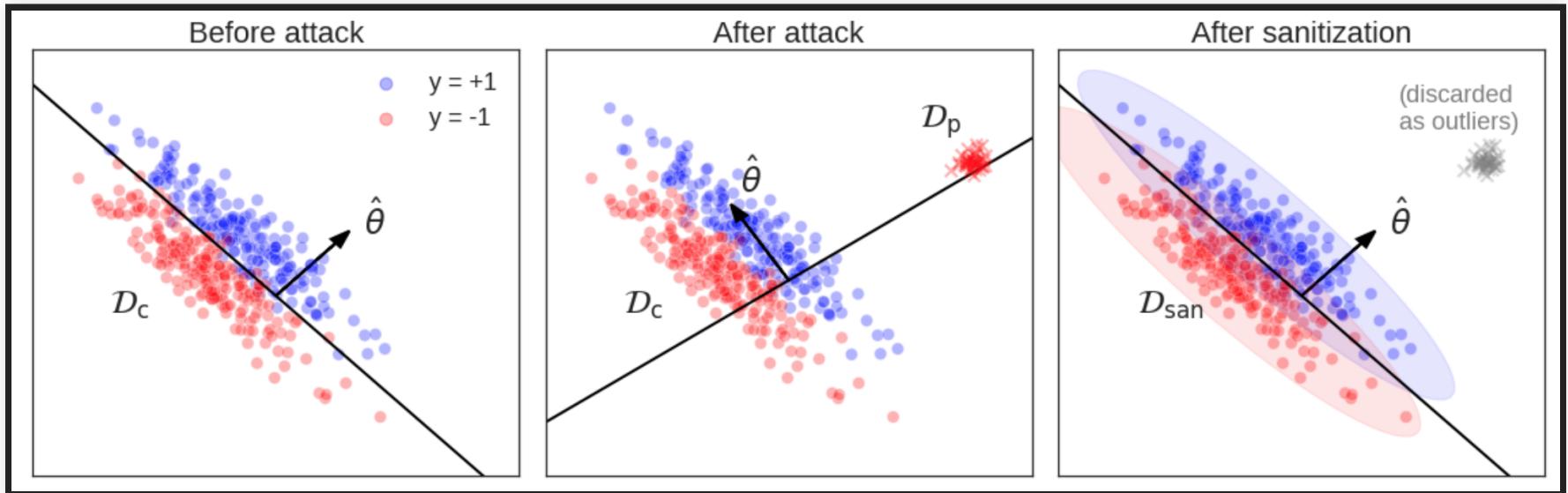
Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks, Shafahi et al. (2018)

EXAMPLE: HOME ASSISTANT ROBOT



- What are the security requirements?
- What are possible poisoning attacks?
- What does the attacker need to know/access?

DEFENSE AGAINST POISONING ATTACKS



Stronger Data Poisoning Attacks Break Data Sanitization Defenses, Koh, Steinhardt, and Liang (2018).

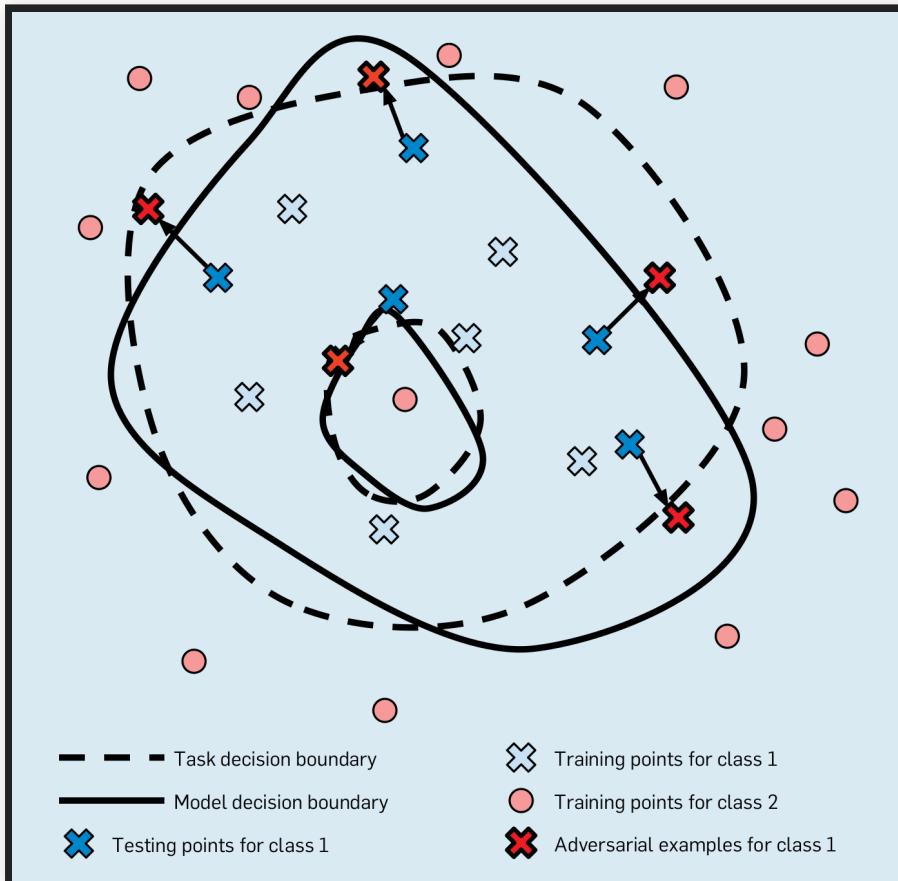
EVASION ATTACKS (ADVERSARIAL EXAMPLES)



- Add noise to an existing sample & cause misclassification
- Attack at inference time
 - Typically assumes knowledge of the model (algorithm, parameters)
 - Recently, shown to be possible even when the attacker only has access to model output ("blackbox" attack)

Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition, Sharif et al. (2016).

TASK DECISION BOUNDARY VS MODEL BOUNDARY



From Goodfellow et al (2018). [Making machine learning robust against adversarial inputs](#). *Communications of the ACM*, 61(7), 56-66.

GENERATING ADVERSARIAL EXAMPLES

- see [counterfactual explanations](#)
- Find similar input with different prediction
 - targeted (specific prediction) vs untargeted (any wrong prediction)
- Many similarity measures (e.g., change one feature vs small changes to many features)
 - $x^* = x + \operatorname{argmin}\{|z| : f(x + z) = t\}$
- Attacks more affective which access to model internals, but also black-box attacks (with many queries to the model) feasible
 - With model internals: follow the model's gradient
 - Without model internals: learn [surrogate model](#)
 - With access to confidence scores: heuristic search (eg. hill climbing)

MODEL INVERSION: CONFIDENTIALITY



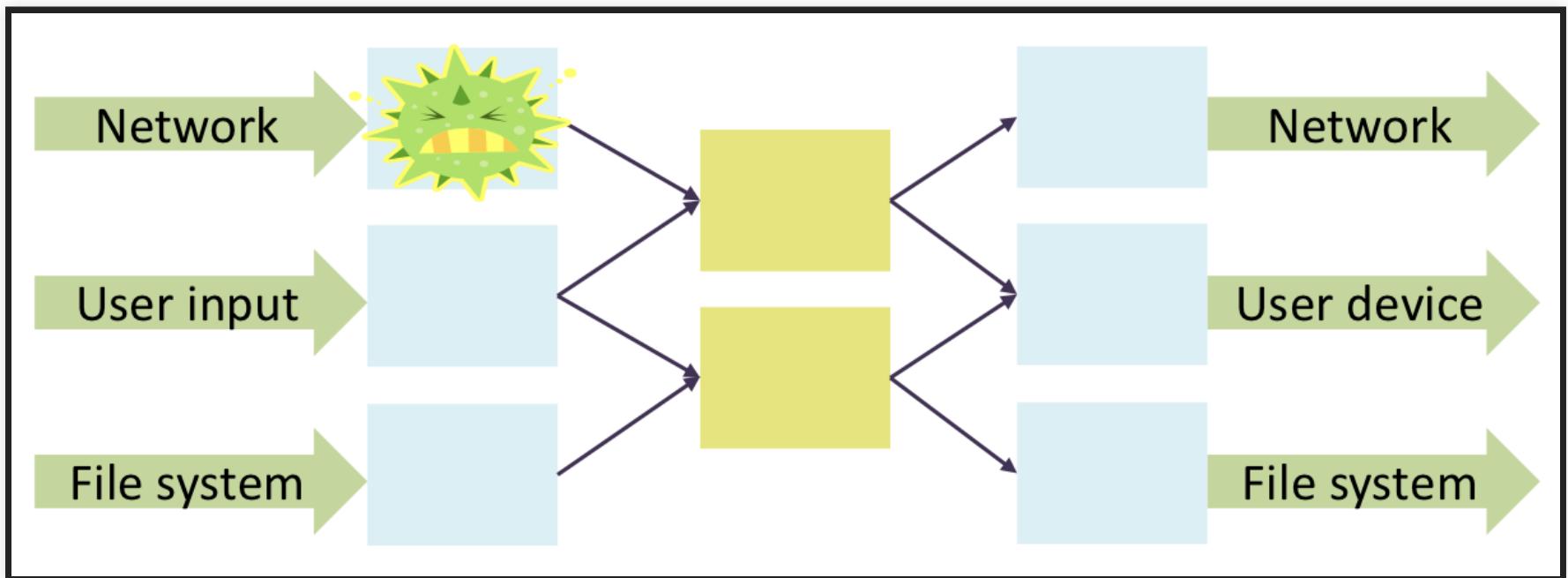
- Given a model output (e.g., name of a person), infer the corresponding, potentially sensitive input (facial image of the person)
- One method: Gradient descent on input space
 - Assumes that the model produces a confidence score for prediction
 - Start with a random input vector & iterate towards input values with higher confidence level

DESIGNING FOR SECURITY: SECURITY MINDSET



- Assume that all components may be compromised at one point or another
- Don't assume users will behave as expected; assume all inputs to the system as potentially malicious
- Aim for risk minimization, not perfect security; reduce the chance of catastrophic failures from attacks

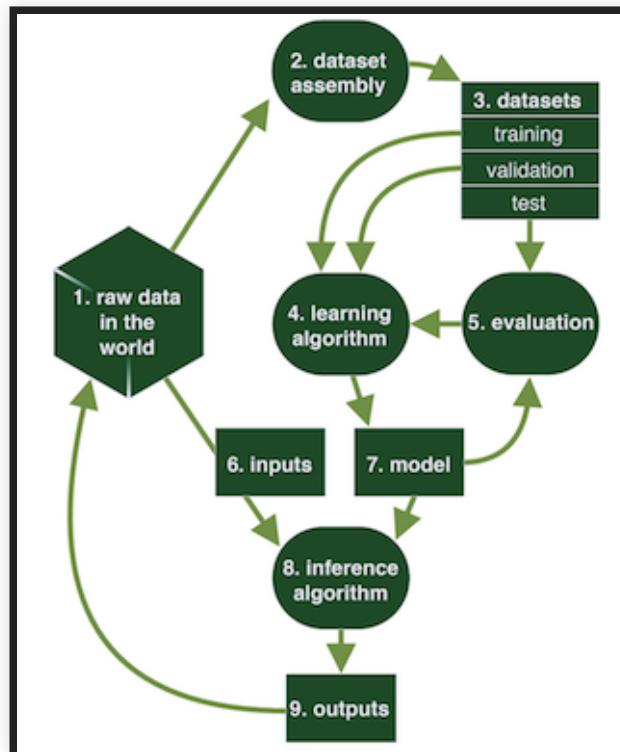
COMPARTMENTALIZED DESIGN



Flaw in one component => Limited impact on the rest of the system!

SECURE DESIGN PRINCIPLES FOR ML

- Principle of least privilege
 - Who has access to training data, model internal, system input & output, etc.,?
 - Does any user/stakeholder have more access than necessary?
 - If so, limit access by using authentication mechanisms



HUMAN AI INTERACTION

Guest Lecture by Haiyi Zhu

SAFETY

Eunsuk Kang

Required Reading: [Practical Solutions for Machine Learning Safety in Autonomous Vehicles](#). S. Mohseni et al.,
SafeAI Workshop@AAAI (2020).

LEARNING GOALS

- Understand safety concerns in traditional and AI-enabled systems
- Apply hazard analysis to identify risks and requirements and understand their limitations
- Discuss ways to design systems to be safe against potential failures
- Suggest safety assurance strategies for a specific project
- Describe the typical processes for safety evaluations and their limitations

SAFETY OF AI-ENABLED SYSTEMS

Tweet

SAFETY OF AI-ENABLED SYSTEMS

Tweet

SAFETY IS A BROAD CONCEPT

- Not just physical harms/injuries to people
- Includes harm to mental health
- Includes polluting the environment, including noise pollution
- Includes harm to society, e.g. poverty, polarization

CASE STUDY: SELF-DRIVING CAR

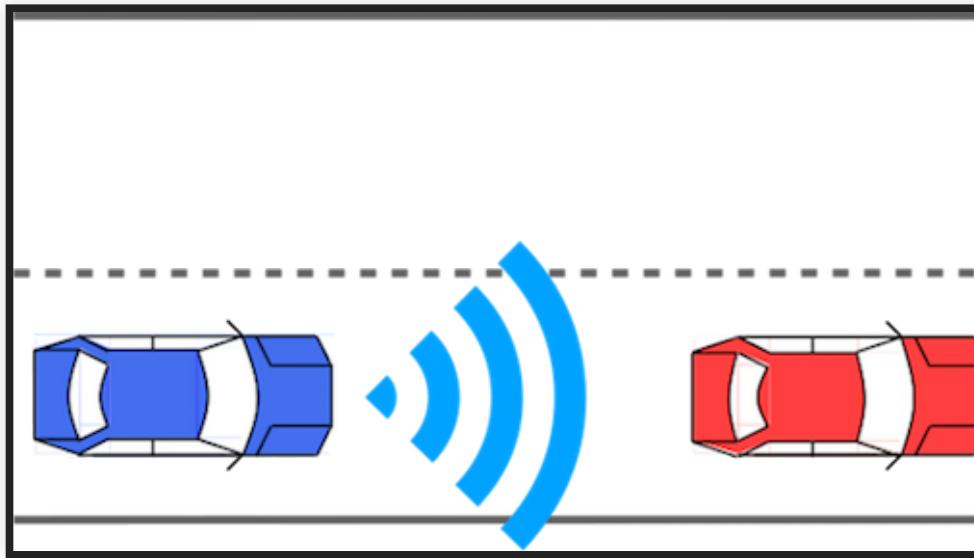


CHALLENGE: EDGE/UNKNOWN CASES



- Gaps in training data; ML will unlikely to cover all unknown cases
- **Why is this a unique problem for AI? What about humans?**

WHAT IS HAZARD ANALYSIS?



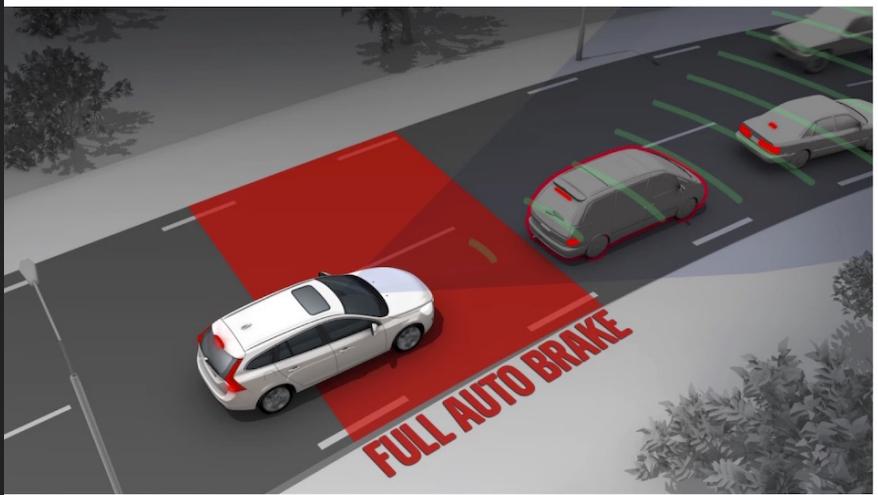
- **Hazard:** A condition or event that may result in undesirable outcome
 - e.g., "Ego vehicle is in risk of a collision with another vehicle."
- **Safety requirement:** Intended to eliminate or reduce one or more hazards
 - "Ego vehicle must always maintain some minimum safe distance to the leading vehicle."
- **Hazard analysis:** Methods for identifying hazards & potential root causes

FAILURE MODE AND EFFECTS ANALYSIS (FMEA)

	Function	Potential Failure Mode	Potential Effect(s) of Failure	SEV i	Potential Cause(s) of Failure	OCC i	Current Design Controls (Prevention)	Current Design Controls (Detection)	DET i	RPN i	Recommended Action(s)
1	Provide required levels of radiation	Radiation level too high for the required intervention	Over radiation of the patients.		Technician did not set the radiation at the right level.			Current algorithm resets to normal levels after imaging each patient.			Modify software to alert technician to unusually high radiation levels before activating.
2		Radiation at lower level than required	Patient fails to receive enough radiation.		Software does not respond to hardware mechanical setting.			Failure detection included in software			Include visual / audio alarm in the code when lack of response.
3											Improve recovery protocol.
4	Protect patients from unexpected high radiation	Higher radiation than required	Radiation burns		sneak paths in software			Shut the system if radiation level does not match the inputs.			Perform traceability matrix.

- A **forward search** technique to identify potential hazards
- For each function, (1) enumerate possible *failure modes* (2) possible safety impact (*effects*) and (3) mitigation strategies.
- Widely used in aeronautics, automotive, healthcare, food services, semiconductor processing, and (to some extent) software

HAZOP EXAMPLE: EMERGENCY BRAKING (EB)



The diagram shows a silver car on a road. A red diagonal band from the front of the car extends to the right, labeled "FULL AUTO BRAKE". Behind the car, a green dashed line indicates the path it has traveled. In the background, there are other cars and trees.

Guide Word	Meaning
NO OR NOT	Complete negation of the design intent
MORE	Quantitative increase
LESS	Quantitative decrease
AS WELL AS	Qualitative modification/increase
PART OF	Qualitative modification/decrease
REVERSE	Logical opposite of the design intent
OTHER THAN / INSTEAD	Complete substitution
EARLY	Relative to the clock time
LATE	Relative to the clock time
BEFORE	Relating to order or sequence
AFTER	Relating to order or sequence

- Specification: EB must apply a maximum braking command to the engine.
 - **NO OR NOT:** EB does not generate any braking command.
 - **LESS:** EB applies less than max. braking.
 - **LATE:** EB applies max. braking but after a delay of 2 seconds.
 - **REVERSE:** EB generates an acceleration command instead of braking.
 - **BEFORE:** EB applies max. braking before a possible crash is detected.

ROBUSTNESS IN A SAFETY SETTING

- Does the model reliably detect stop signs?
- Also in poor lighting? In fog? With a tilted camera? Sensor noise?
- With stickers taped to the sign? (adversarial attacks)

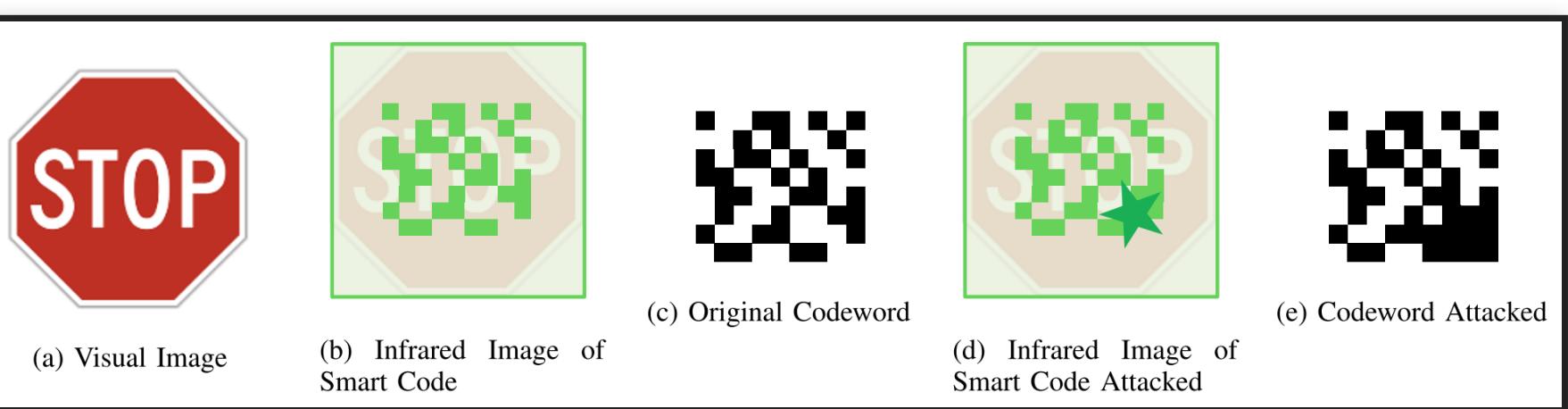
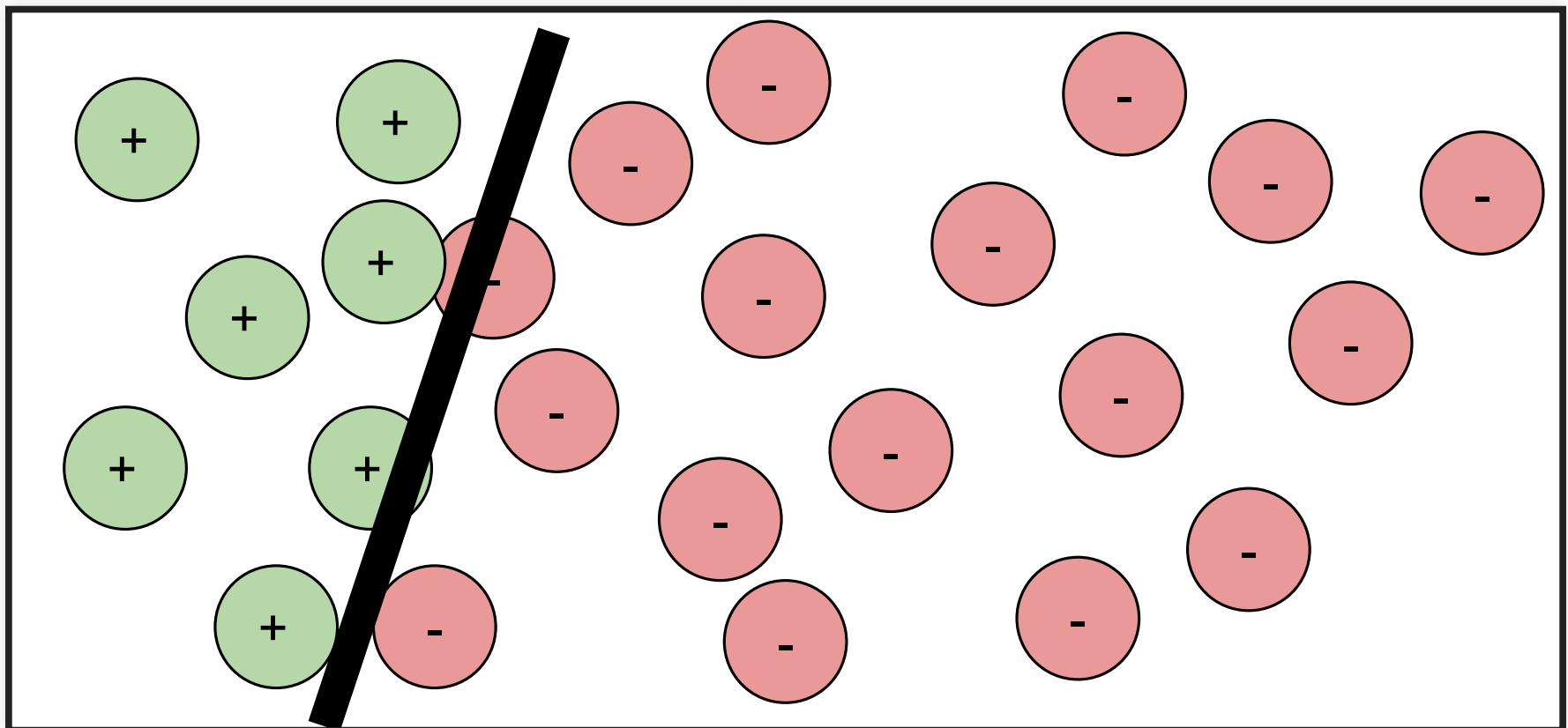


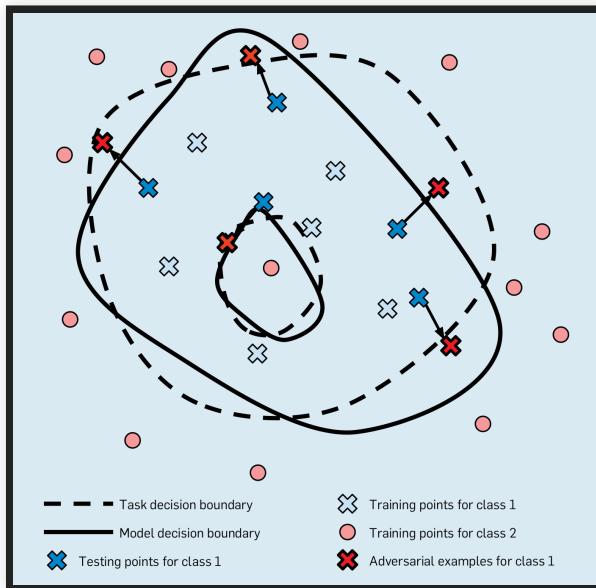
Image: David Silver. [Adversarial Traffic Signs](#). Blog post, 2017

NO MODEL IS FULLY ROBUST

- Every useful model has at least one decision boundary (ideally at the real task decision boundary)
- Predictions near that boundary are not (and should not) be robust



TASK DECISION BOUNDARY VS MODEL BOUNDARY



- Decision boundary: Ground truth; often unknown and not specifiable
- Model boundary: What the model learns; an approximation of decision boundary
- Often, learned & actual decision boundaries do not match!

From Goodfellow et al (2018). [Making machine learning robust against adversarial inputs](#). *Communications of the ACM*, 61(7), 56-66.

SAFETY ASSURANCE WITH ML COMPONENTS

- Consider ML components as unreliable, at most probabilistic guarantees
- Testing, testing, testing (+ simulation)
 - Focus on data quality & robustness
- *Adopt a system-level perspective!*
- Consider safe system design with unreliable components
 - Traditional systems and safety engineering
 - Assurance cases
- Understand the problem and the hazards
 - System level, goals, hazard analysis, world vs machine
 - Specify *end-to-end system behavior* if feasible
- Recent research on adversarial learning and safety in reinforcement learning

SOFTWARE ENGINEERING FOR SAFE SELF-DRIVING

Owen Cheng, Uber ATG

FOSTERING INTERDISCIPLINARY TEAMS

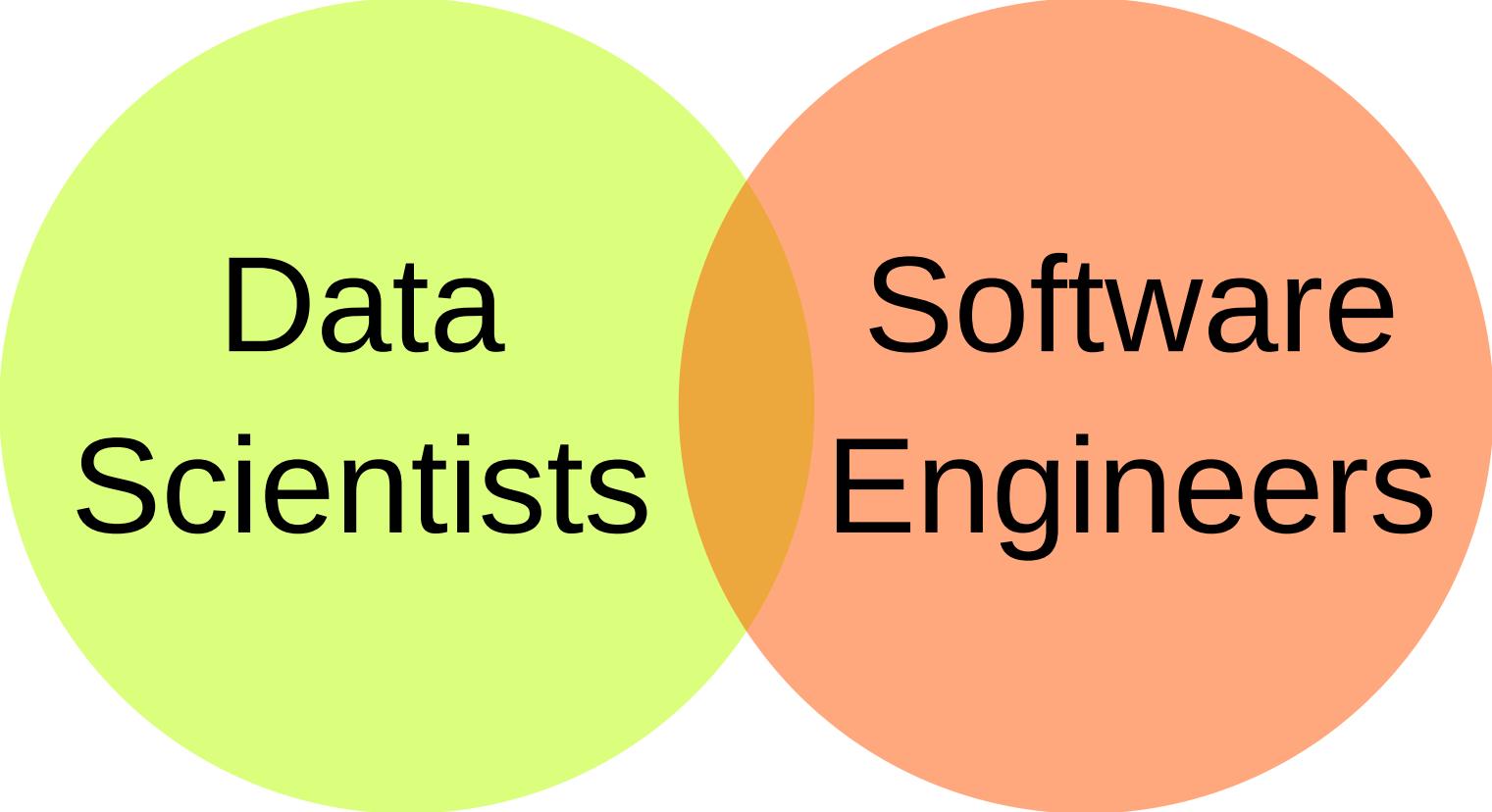
(Process and Team Reflections)

Christian Kaestner

Required reading: Kim, Miryung, Thomas Zimmermann, Robert DeLine, and Andrew Begel. "[Data scientists in software teams: State of the art and challenges.](#)" IEEE Transactions on Software Engineering 44, no. 11 (2017): 1024-1038.

LEARNING GOALS

- Plan development activities in an inclusive fashion for participants in different roles
- Describe agile techniques to address common process and communication issues



A Venn diagram consisting of two overlapping circles. The left circle is light green and contains the text "Data Scientists". The right circle is light orange and contains the text "Software Engineers". The two circles overlap in the center.

Data
Scientists

Software
Engineers



DATA SCIENCE ROLES AT MICROSOFT

- Polymath
- Data evangelist
- Data preparer
- Data shaper
- Data analyzer
- Platform builder
- 50/20% moonlighter
- Insight actors

Kim, Miryung, Thomas Zimmermann, Robert DeLine, and Andrew Begel. "[Data scientists in software teams: State of the art and challenges.](#)" IEEE Transactions on Software Engineering 44, no. 11 (2017): 1024-1038.

OTHER ROLES IN AI SYSTEMS PROJECTS?

- Domain specialists
- Business, management, marketing
- Project management
- Designers, UI experts
- Operations
- Lawyers
- Social scientists, ethics
- ...

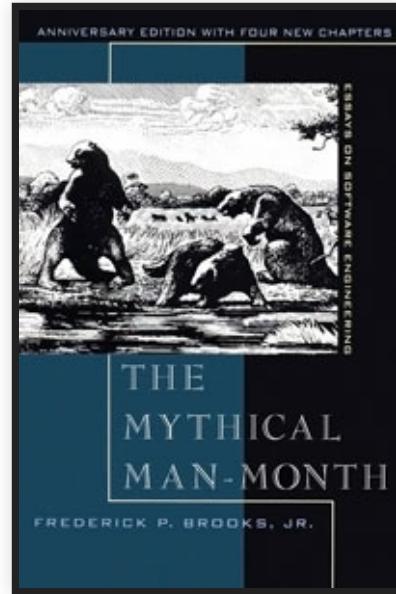
HOW TO STRUCTURE TEAMS?

Mobile game; 50ish developers; distributed teams?



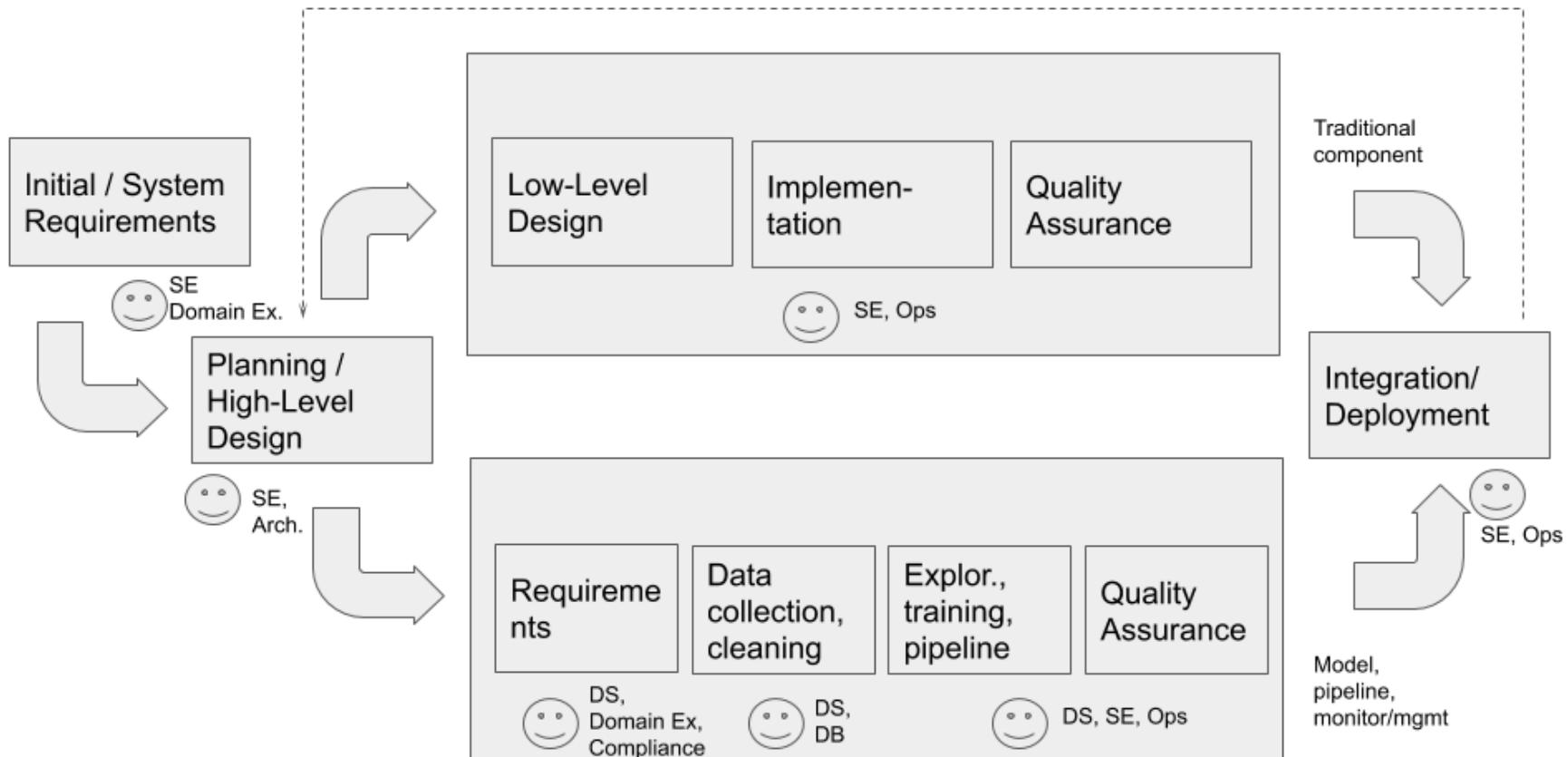
MYTHICAL MAN MONTH

Brooks's law: Adding manpower to a late software project makes it later

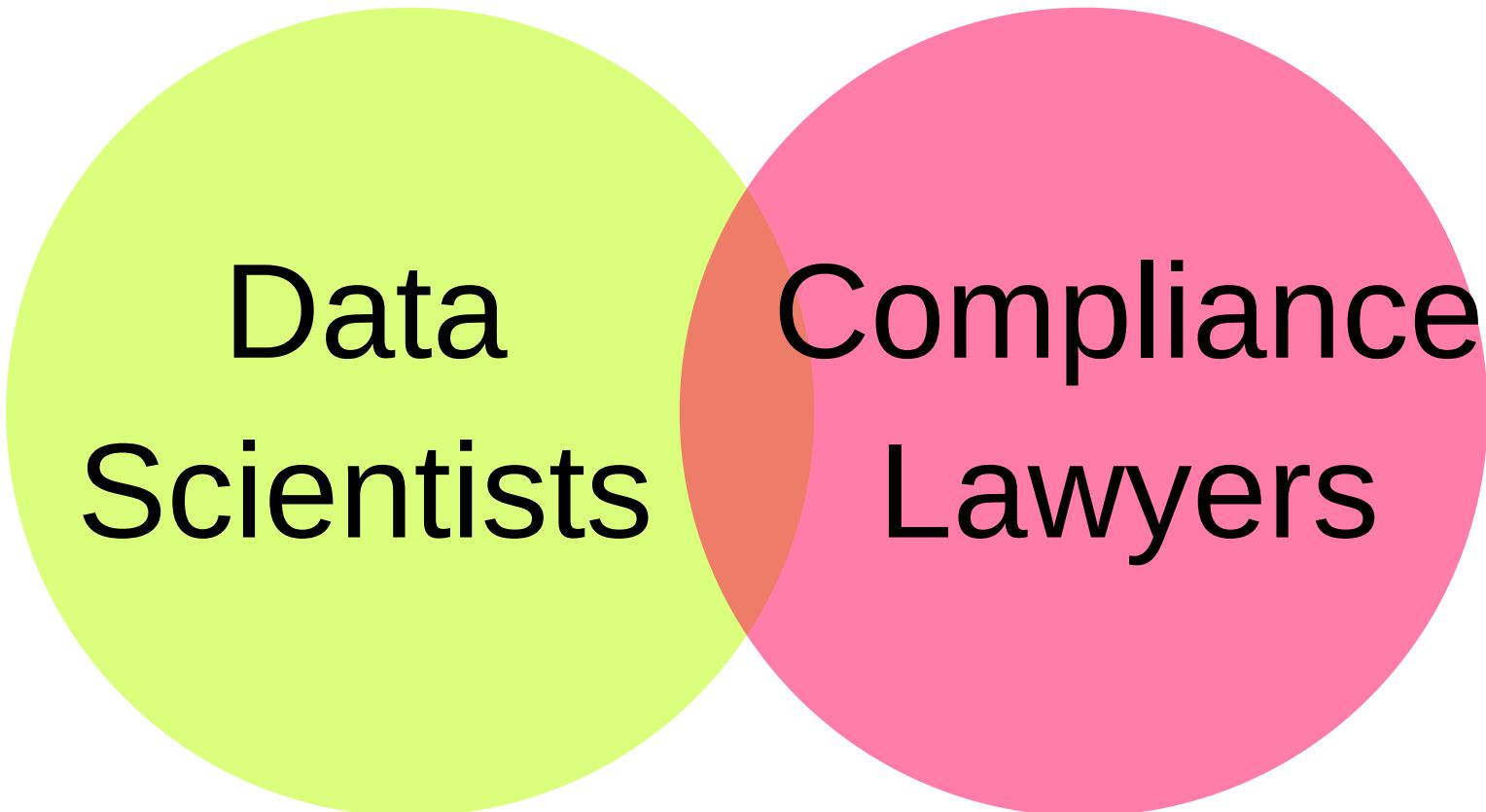


1975, describing experience at IBM developing OS/360

INTERDISCIPLINARY COLLABORATIONS



CONFLICTING GOALS?



T-SHAPED PEOPLE

Broad-range generalist + Deep expertise

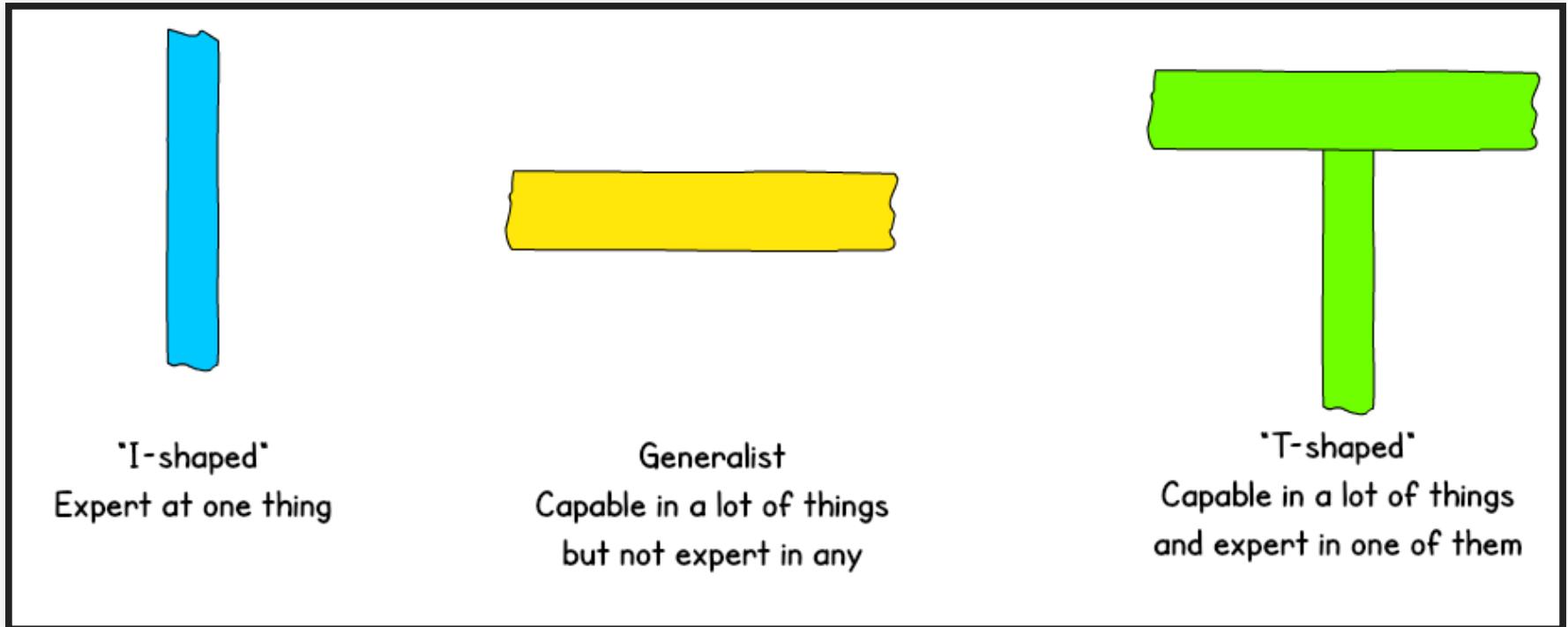


Figure: Jason Yip. [Why T-shaped people?](#). 2018

TEAM ISSUES: GROUPTHINK





TEAM ISSUES: SOCIAL LOAFING



TODAY

Looking back at the
semester

Discussion of future of
SE4AI

Feedback for future
semesters

