FROM MODELS TO PRODUCTION SYSTEMS (SYSTEMS THINKING)

Christian Kaestner

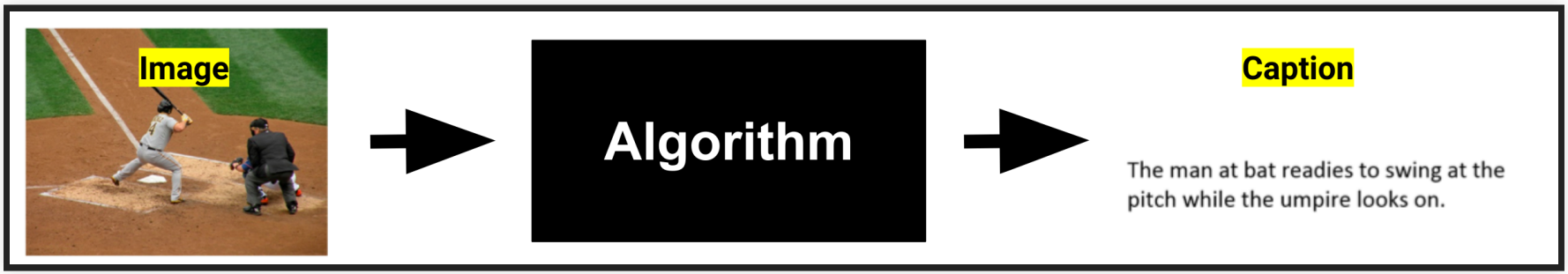
- Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018), Chapters 5, 7, and 8.

LEARNING GOALS

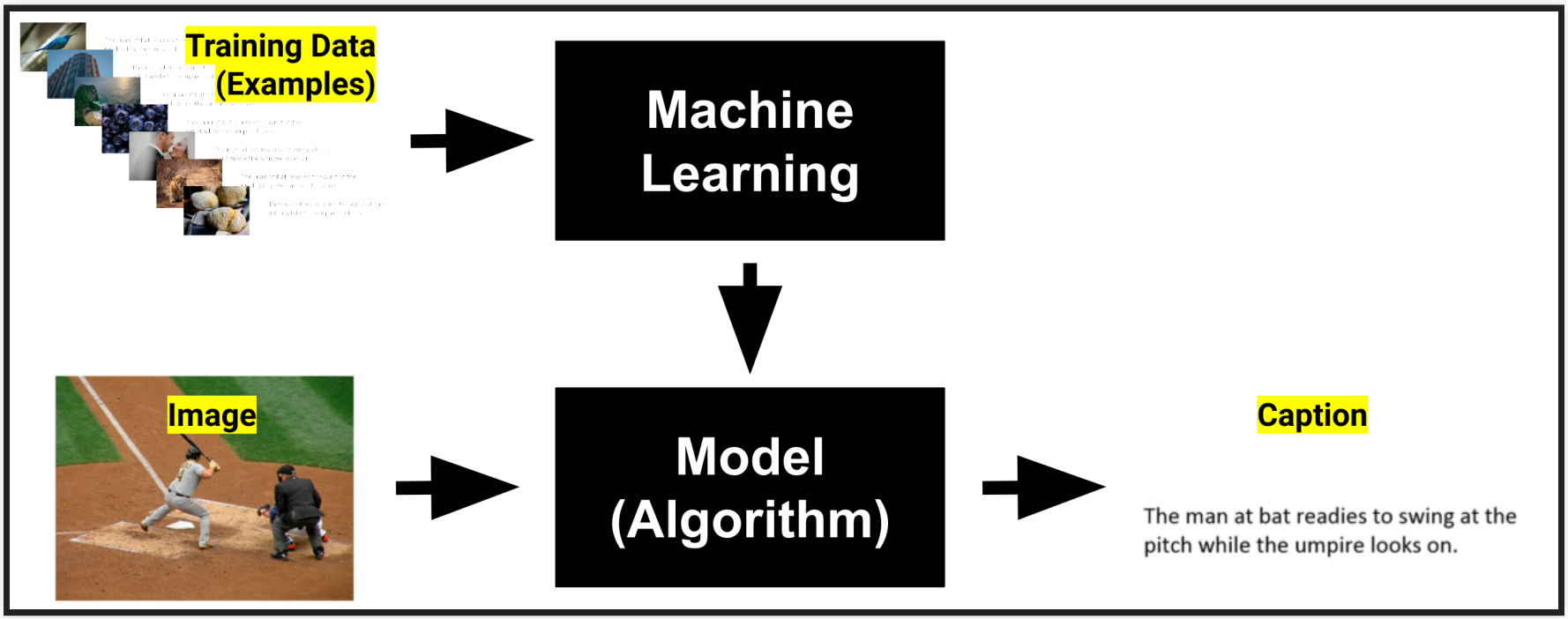
- Explain the consequences of the shift from deductive to inductive reasoning for abstraction and composition
- Explain how machine learning fits into the larger picture of building and maintaining production systems
- Explain the modularity implications of having machine-learning components without specifications
- Describe the typical components relating to AI in an AI-enabled system and typical design decisions to be made

ML MODELS AS PART OF A SYSTEM

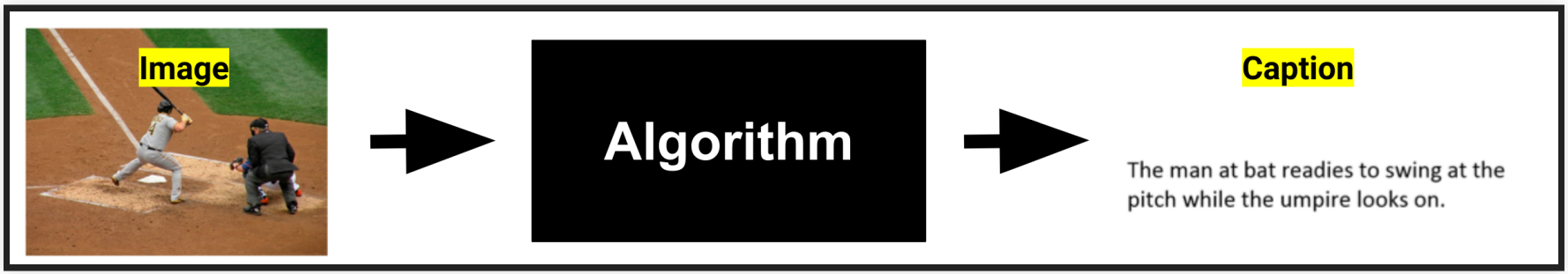
EXAMPLE: IMAGE CAPTIONING PROBLEM



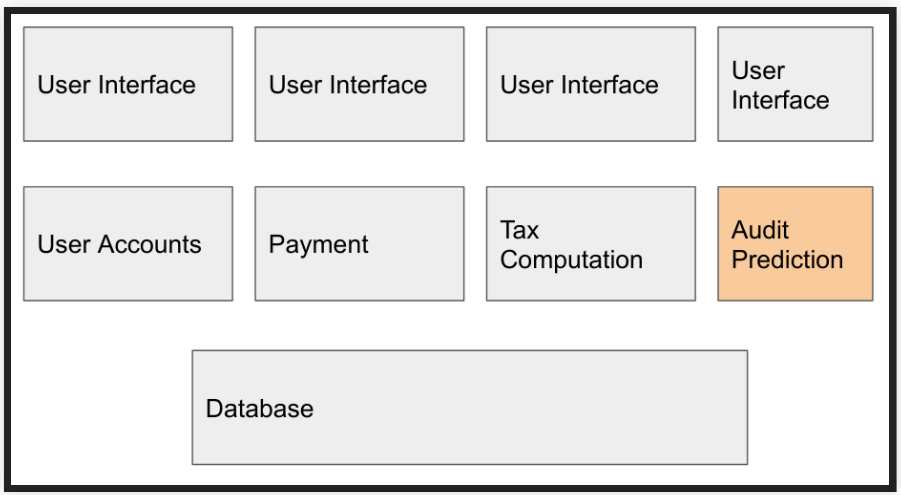
EXAMPLE: IMAGE CAPTIONING PROBLEM

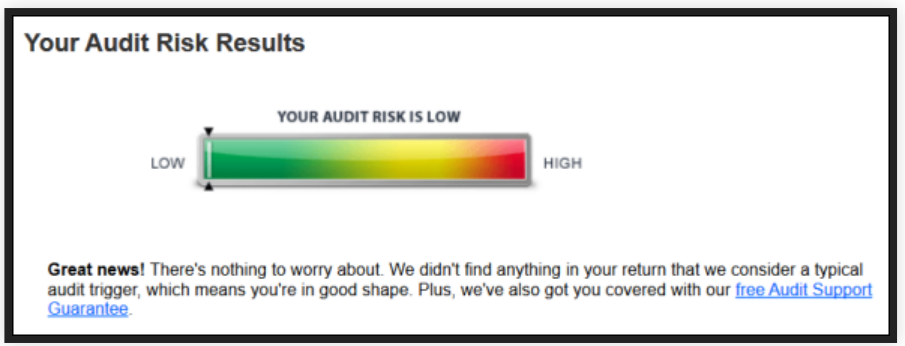


WHY DO WE CARE ABOUT IMAGE CAPTIONING?



MACHINE LEARNING AS (SMALL) COMPONENT IN A SYSTEM

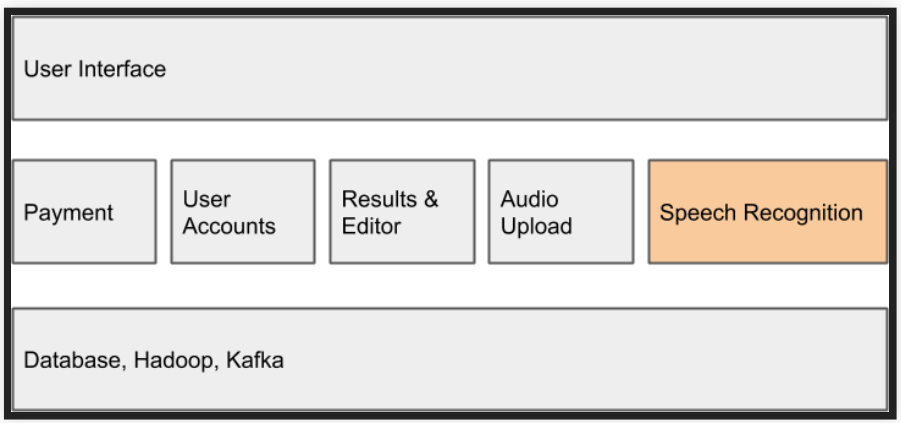


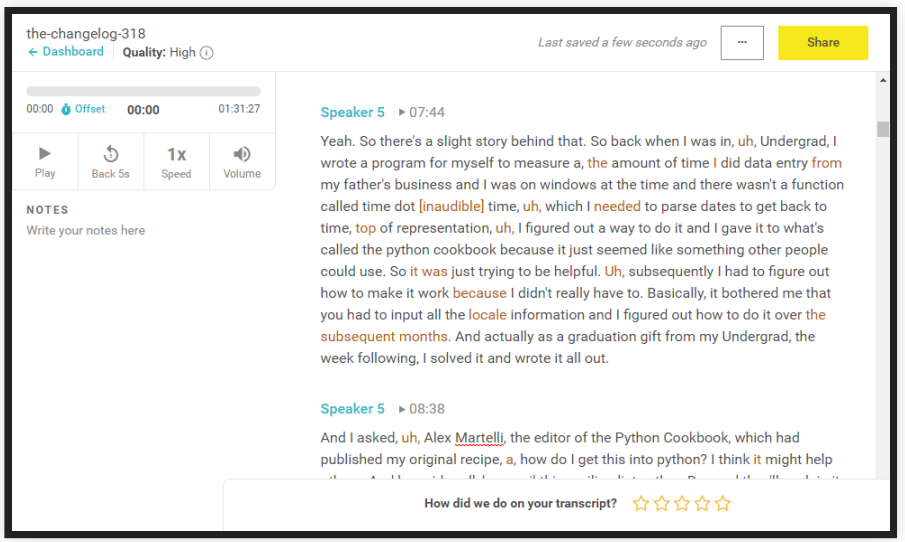


Speaker notes

Traditional non-ML tax software, with an added ML component for audit risk estimation

MACHINE LEARNING AS (CORE) COMPONENT IN A SYSTEM





Speaker notes

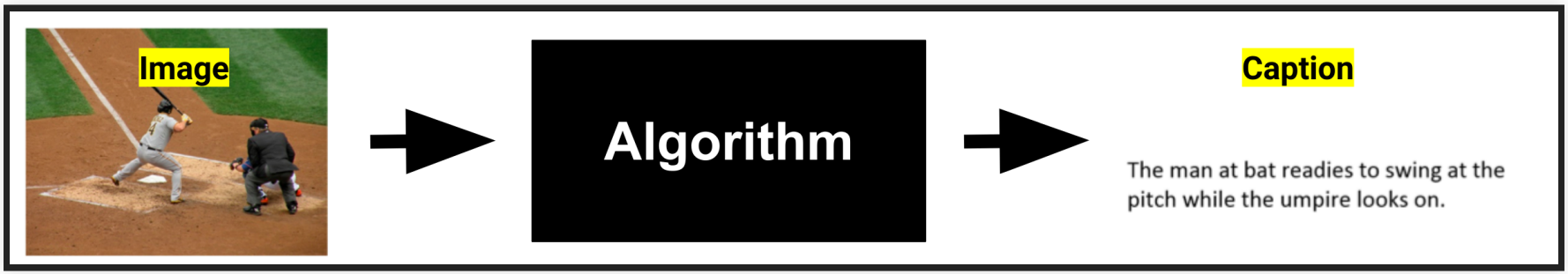
Transcription service, where interface is all built around an ML component

MANY MORE EXAMPLES:

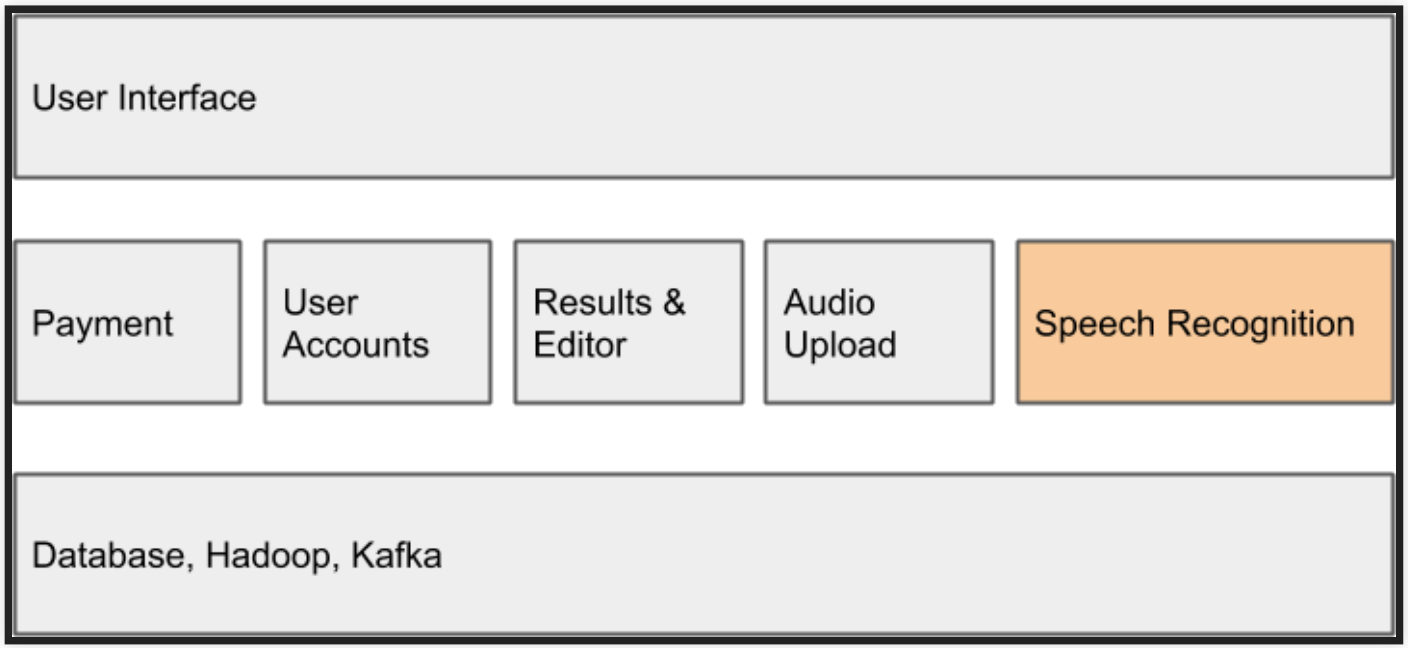
- Product recommendations on Amazon
- Surge price calculation for Uber
- Inventory planning in Walmart
- Search for new oil fields by Shell
- Adaptive cruise control in a car
- Smart app suggestion in Android
- Fashion trends prediction with social media data
- Suggesting whom to talk to in a presidential campain
- Tracking and predicting infections in a pandemic
- Adaptively reacting to network issues by a cell phone provider
- Matching players in a computer game by skill
- ...

- Some for end users, some for employees, some for expert users
- Big and small components of a larger system

MODEL VS SYSTEM GOAL?



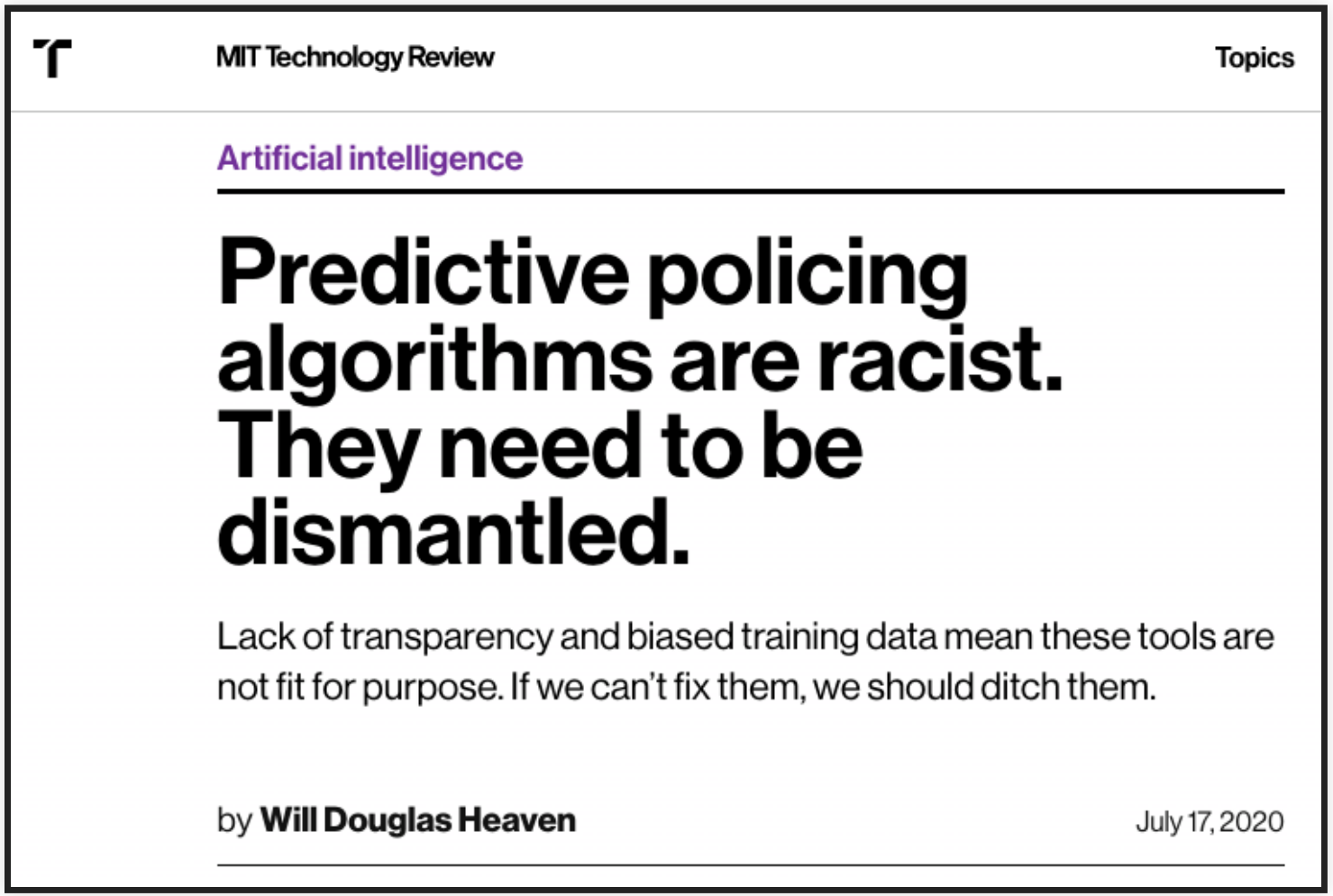
MODEL VS SYSTEM GOAL?



MORE ACCURATE PREDICTIONS MAY NOT BE THAT IMPORTANT

- "Good enough" may be good enough
- Prediction critical for system success or just an gimmick?
- Better predictions may come at excessive costs
    - need way more data, much longer training times
    - privacy concerns
- Better user interface ("experience") may mitigate many problems
    - e.g. explain decisions to users
- Use only high-confidence predictions?

BEYOND SOFTWARE: IMPACT ON OUR SOCIETY



MIT Technology Review Topics

Artificial intelligence

Predictive policing algorithms are racist. They need to be dismantled.

Lack of transparency and biased training data mean these tools are not fit for purpose. If we can't fix them, we should ditch them.

by **Will Douglas Heaven** July 17, 2020

MACHINE LEARNING THAT MATTERS

- 2012 essay lamenting focus on algorithmic improvements and benchmarks in ML
    - focus on standard benchmark sets, not engaging with problem: Iris classification, digit recognition, ...
    - focus on abstract metrics, not measuring real-world impact: accuracy, ROC
    - distant from real-world concerns
    - lack of follow-through, no deployment, no impact
- Failure to *reproduce* and *productionize* paper contributions common
- Ignoring design choices in how to collect data, what problem to solve, how to design human-AI interface, measuring impact, ...

- Should focus on making impact -- requires building systems

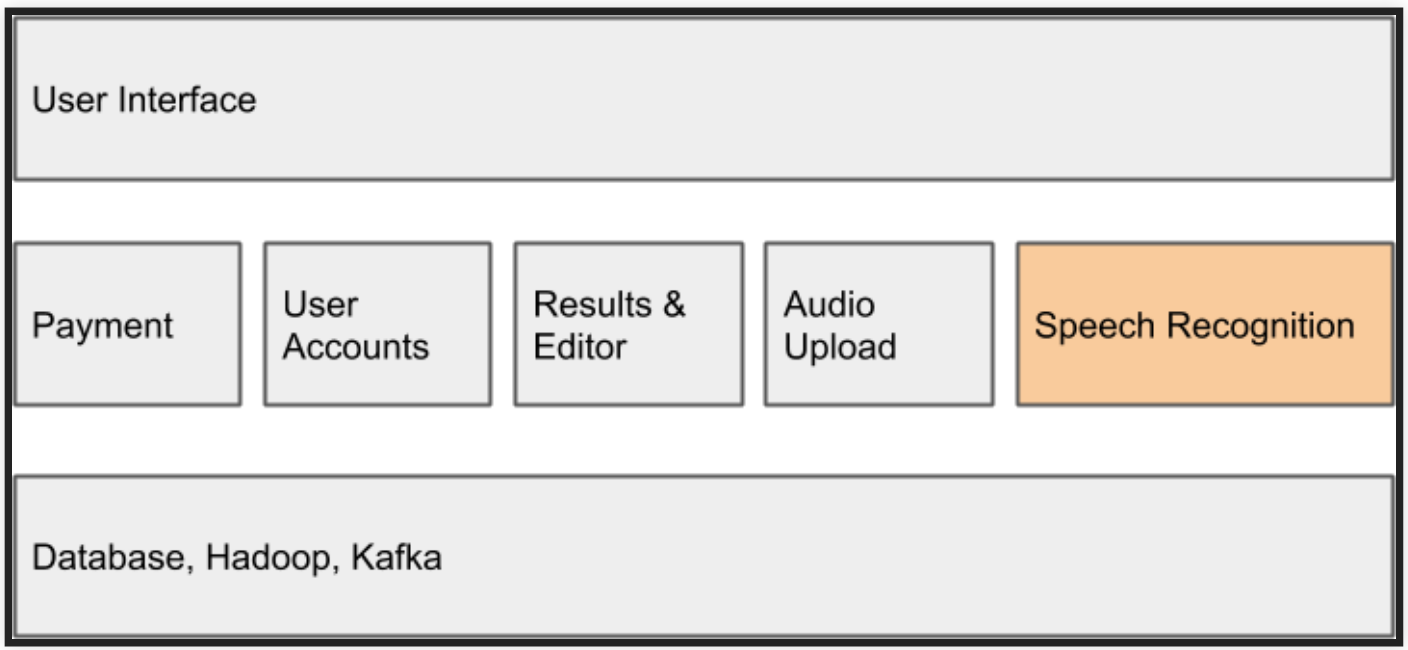
Wagstaff, Kiri. "Machine learning that matters." In Proceedings of the 29 th International Conference on Machine Learning, (2012).

ON TERMINOLOGY

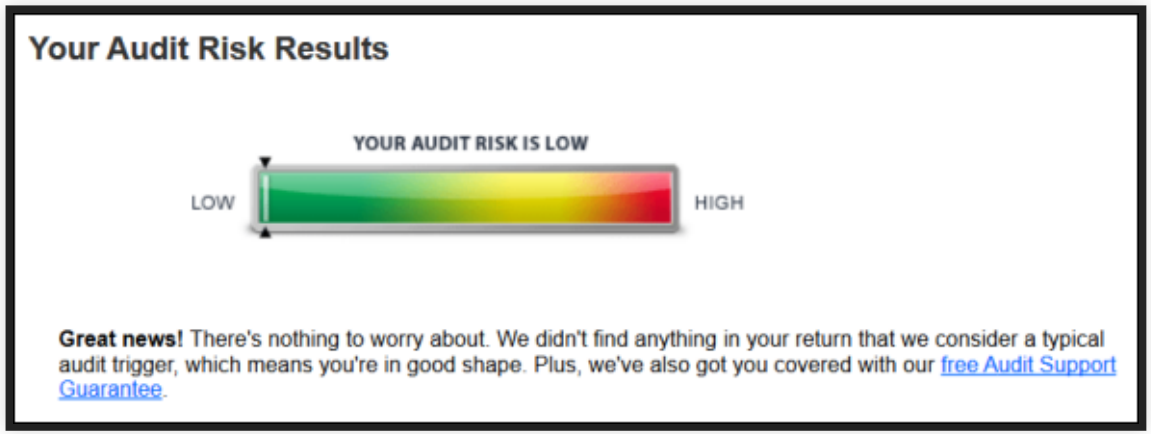
- There is no standard term for referring to building systems with AI components
- "Production ML Systems", "AI-Enabled Systems", "ML-Enabled Systems" or "ML-Infused Systems"; SE4AI, SE4ML
- sometimes "AI Engineering" (but often used with a data-science focus)
- sometimes "ML Systems Engineering" (but often this refers to building distributed and scalable ML learning and data storage platforms)
- "AIOps" ~ using AI to make automated decisions in operations; "DataOps" ~ use of agile methods and automation in business data analytics; "MLOps" ~ technical infrastructure for operating AI-based products and on deploying updates

SYSTEMS THINKING

REPEAT: MACHINE LEARNING AS COMPONENT IN A SYSTEM



THE SYSTEM INTERACTS WITH USERS

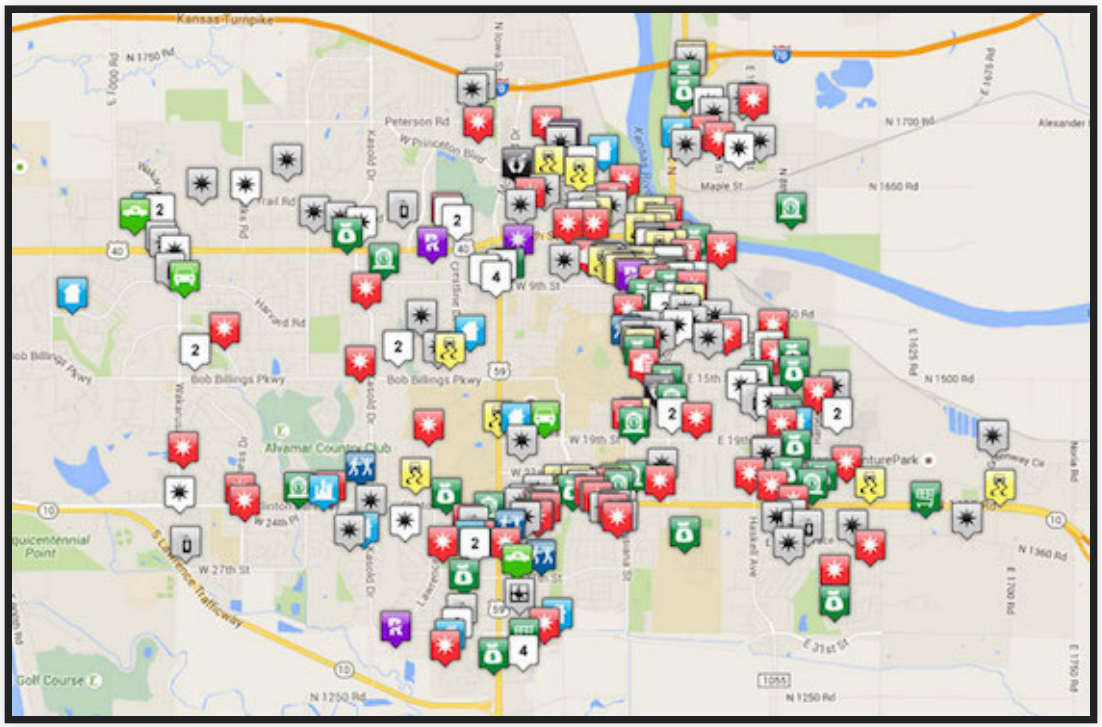


Speaker notes

Audit risk meter from Turbo-Tax

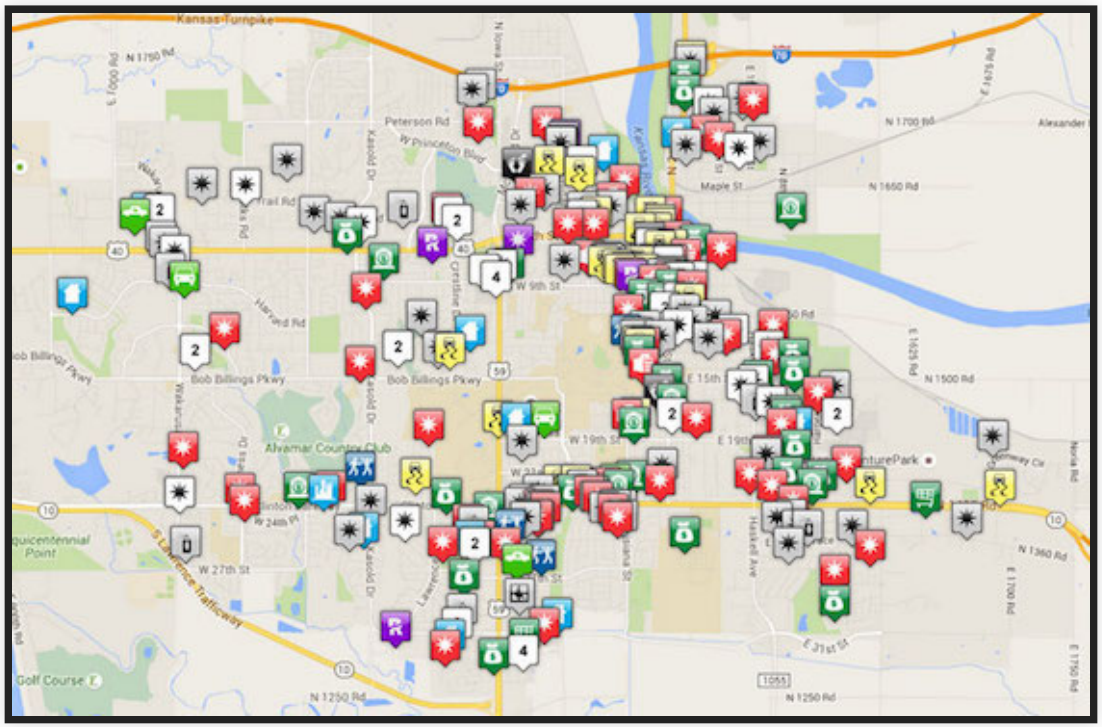
THE SYSTEM INTERACTS WITH THE WORLD

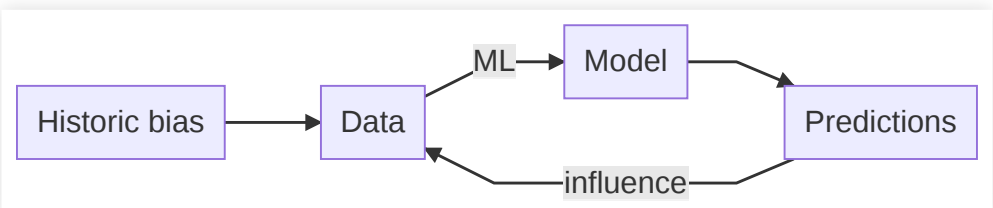
THE SYSTEM INTERACTS WITH THE WORLD



- Model: Use historical data to predict crime rates by neighborhoods
- Used for predictive policing: Decide where to allocate police patrol

SYSTEM <-> WORLD = FEEDBACK LOOPS?





ML PREDICTIONS HAVE CONSEQUENCES

- Assistance, productivity, creativity
- Manipulation, polarization, discrimination
- Feedback loops

SAFETY IS A SYSTEM PROPERTY

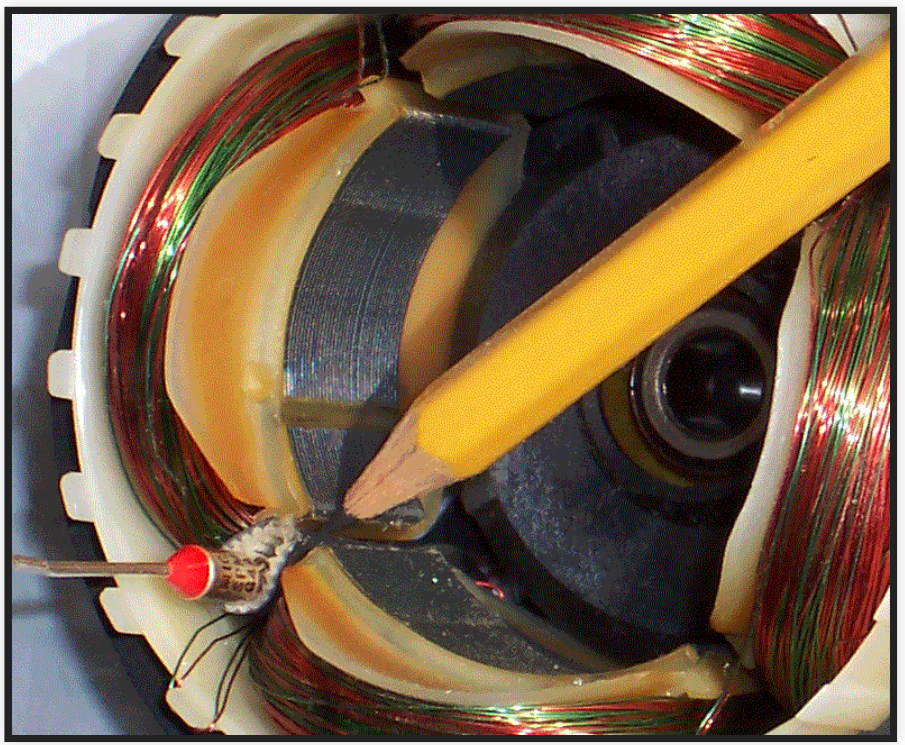
- Code/models are not unsafe, cannot harm people
- Systems can interact with the environment in ways that are unsafe

SAFETY ASSURANCE IN THE MODEL/OUTSIDE THE MODEL

Goal: Ensure smart toaster does not burn the kitchen

SAFETY ASSURANCE IN THE MODEL/OUTSIDE THE MODEL

- In the model
  - Ensure maximum toasting time
  - Use heat sensor and past outputs for prediction
  - Hard to make guarantees
- Outside the model (e.g., "guardrails")
  - Simple code check for max toasting time
  - Non-ML rule to shut down if too hot
  - Hardware solution: thermal fuse



(Image CC BY-SA 4.0, C J Cowie)

MODEL VS SYSTEM PROPERTIES

- Similar to safety, many other qualities should be discussed at model **and** system level
    - Security
    - Privacy
    - Transparency, accountability
    - Maintainability
    - Scalability, energy consumption
    - Impact on system goals
    - ...

THINKING ABOUT SYSTEMS

- Holistic approach, looking at the larger picture, involving all stakeholders
- Looking at relationships and interactions among components and environments
    - Everything is interconnected
    - Combining parts creates something new with emergent behavior
    - Understand dynamics, be aware of feedback loops, actions have effects
- Understand how humans interact with the system

> *A system is a set of inter-related components that work together in a particular environment to perform whatever functions are required to achieve the system's objective -- Donella Meadows*

Leyla Acaroglu. "Tools for Systems Thinkers: The 6 Fundamental Concepts of Systems Thinking." Blogpost 2017

SYSTEM-LEVEL CHALLENGES FOR AI-ENABLED SYSTEMS

- Getting and updating data, concept drift, changing requirements
- Handling massive amounts of data
- Interactions with the real world, feedback loops
- Lack of modularity of AI components, lack of specifications, nonlocal effects
- Deployment and maintenance
- Versioning, debugging and incremental improvement
- Keeping training and operating cost manageable
- Interdisciplinary teams
- Setting system goals, balancing stakeholders and requirements
- ...

DESIGNING INTELLIGENT EXPERIENCES

(Human-AI Interaction)

AI PREDICTIONS SHOULD INFLUENCE THE WORLD

- Smart toaster
- Automated slide design
- Product or music recommendations
- Feed curation in social media or news
- Recidivism prediction
- Health monitoring
- Transcription services
- Image search engine
- Self-driving cars
- Smart home

- Interact with the world through actuators (smart devices) or by influencing people

DESIGNING INTELLIGENT EXPERIENCES

- How to use the output of a model's prediction (for a objective)?
- Design considerations:
    - How to present prediction to a user? Suggestions or automatically take actions?
    - How to effectively influence the user's behavior toward the system's goal?
    - How to minimize the consequences of flawed predictions?
    - How to collect data to continue to learn from users and mistakes?
- Balancing at least three **system-level** outcomes:
    - Achieving objectives
    - Protection from mistakes
    - Collecting data for training

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

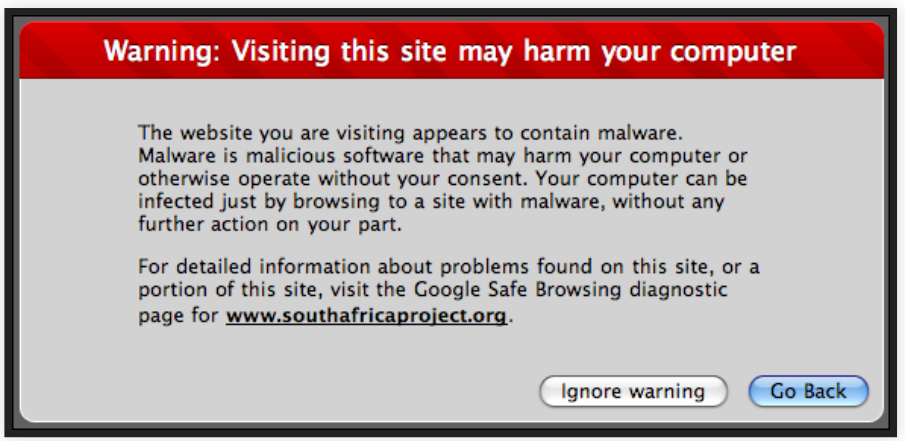
FACTORS TO CONSIDER

- **Forcefulness**: How strongly to encourage taking an action (or even automate it)?
- **Frequency**: How often to interact with the user?
- **Value**: How much does a user (think to) benefit from the prediction?
- **Cost**: What is the damage of a wrong prediction?

BREAKOUT DISCUSSION: EXPERIENCE DESIGN

Fall detection for elderly people:

Safe browsing: Blocking malicious web pages

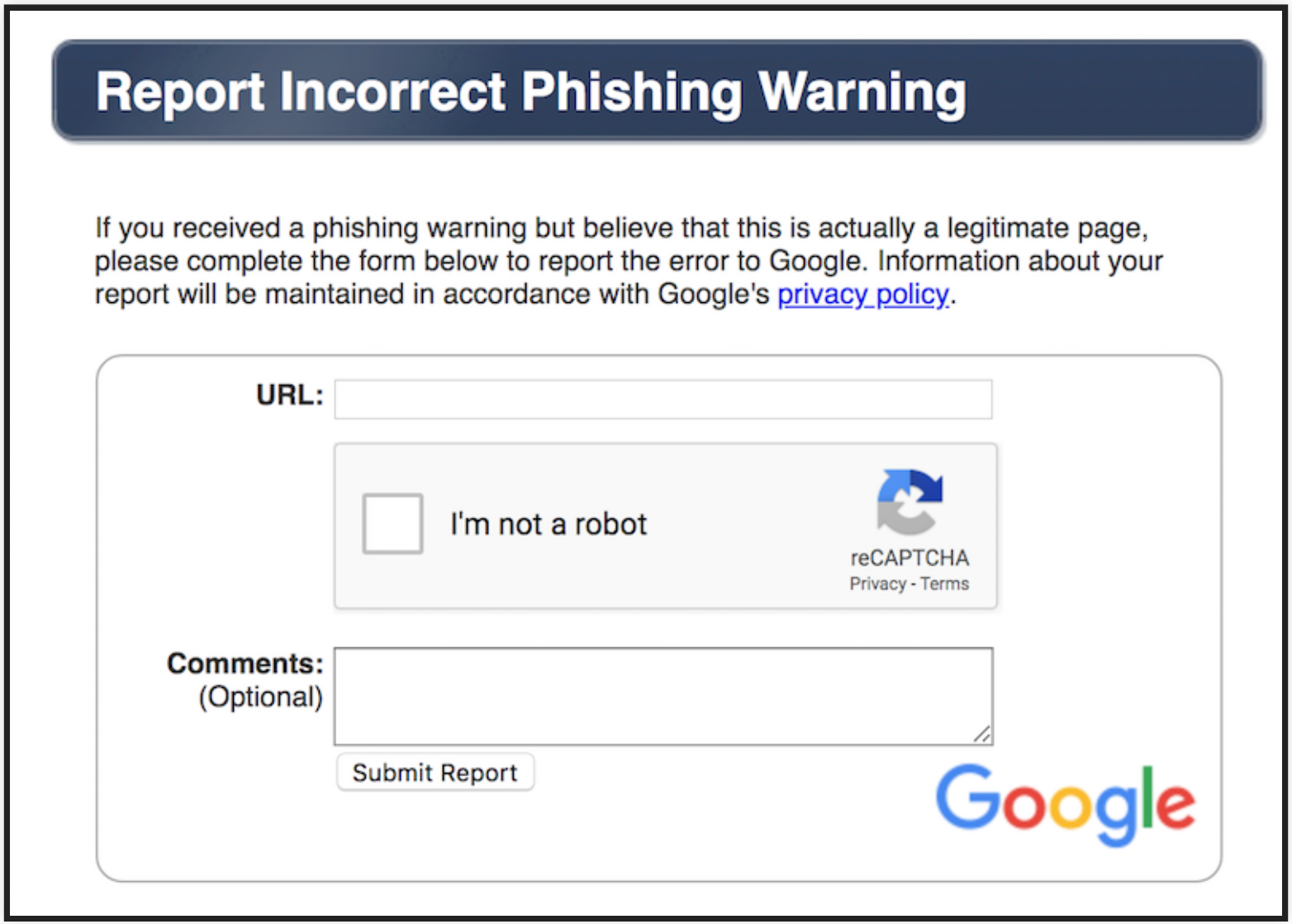


- How do we present the intelligence to the user?
- Consider system goals, forcefulness, frequency, value of correct and cost of wrong predictions

Speaker notes

Devices for older adults to detect falls and alert caretaker or emergency responders automatically or after interaction. Uses various inputs to detect falls. Read more: How fall detection is moving beyond the pendant, MobiHealthNews, 2019

COLLECTING FEEDBACK

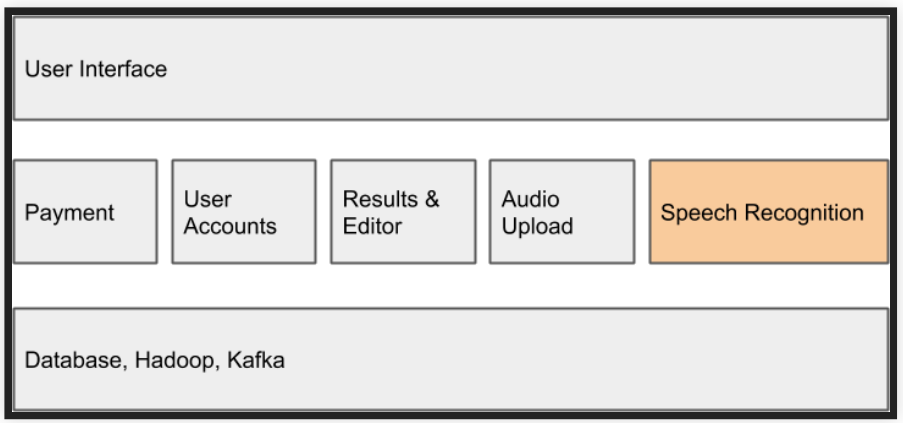


OPERATING PRODUCTION ML SYSTEMS

(deployment, updates)

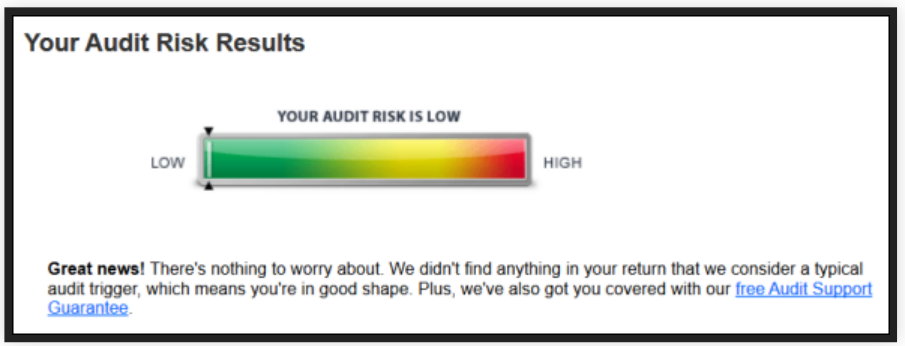
THINGS CHANGE...

- Newer better models released
  - Better model architectures
  - More training data
- Goals and scope change
  - More domains supported
  - Better recognition of dialects
- Model training due to drift
  - New terms (jargon) emerge in domain
  - Increased adoption in region with dialect
- Online experimentation



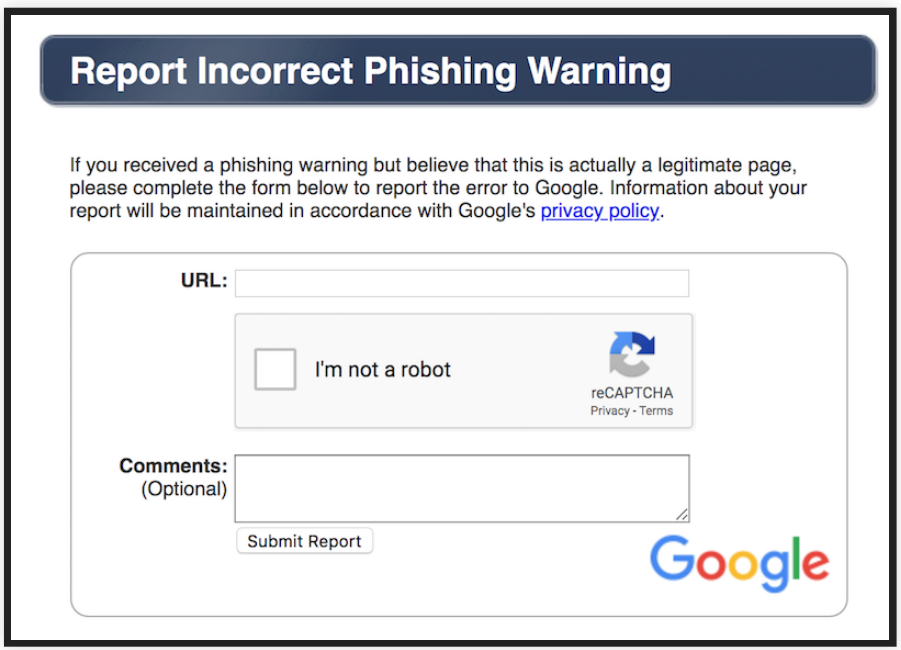
THINGS CHANGE...

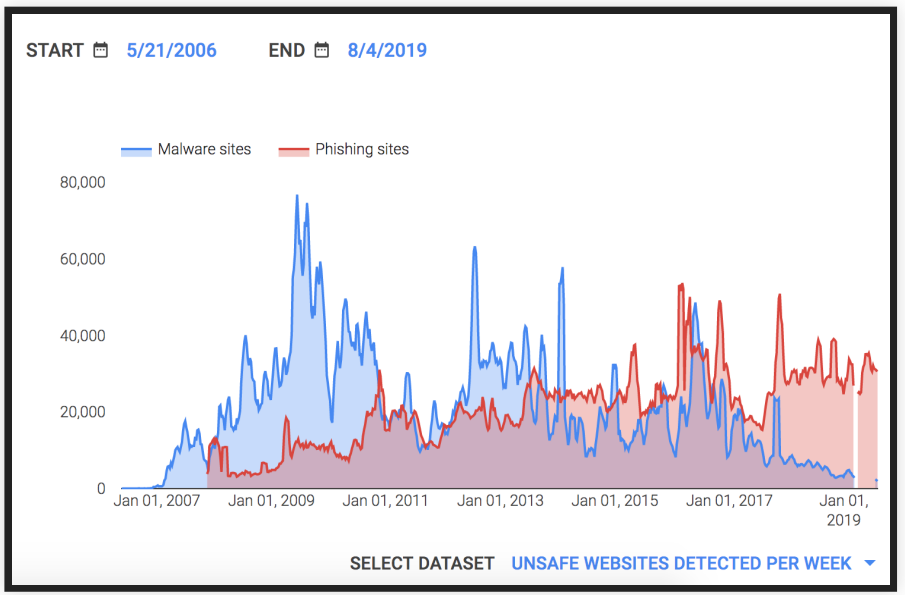
Reasons for change in audit risk prediction?



MONITORING IN PRODUCTION

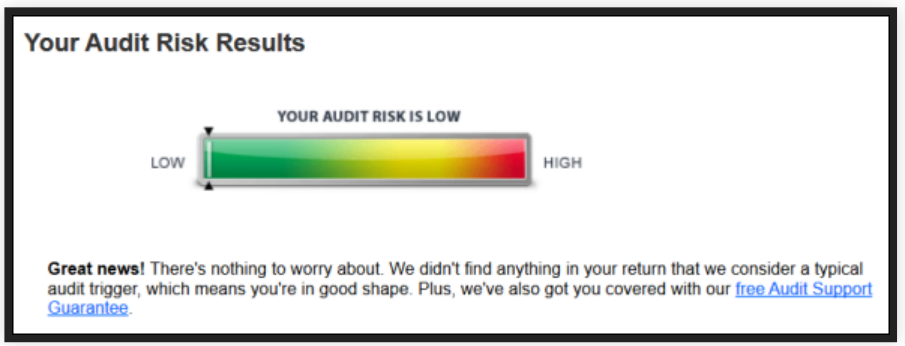
Design for telemetry



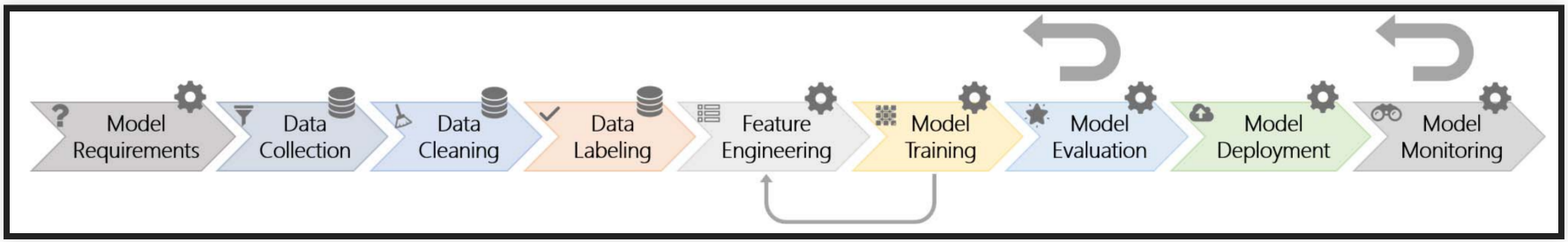


MONITORING IN PRODUCTION

What and how to monitor in audit risk prediction?

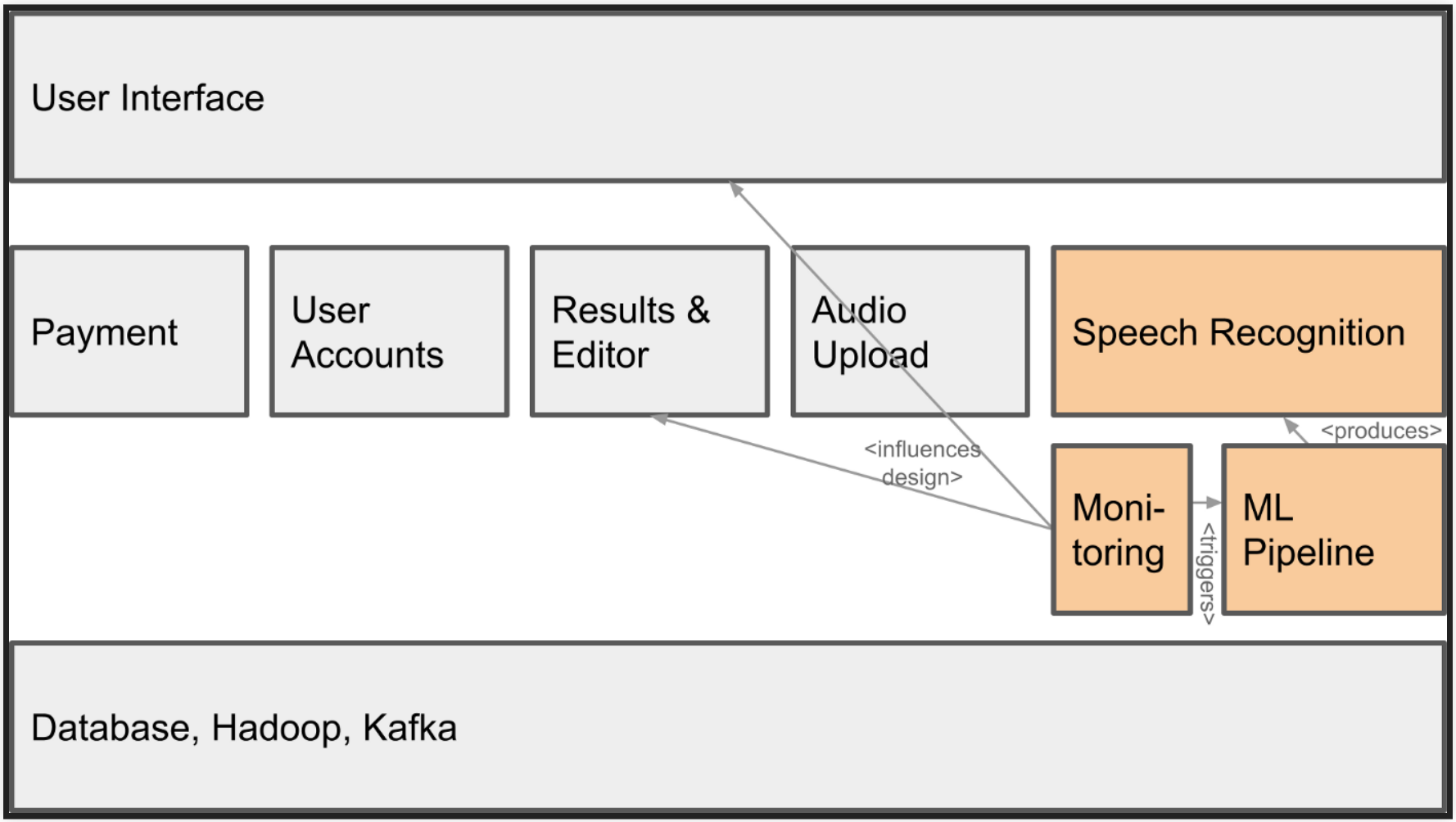


PIPELINE THINKING



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

DESIGN WITH PIPELINE AND MONITORING IN MIND



SHIFTING FROM MODELS TO PIPELINES IS CHALLENGING

Across interviews with enterprise ML teams:

- Data scientists often focus on modeling in local environment, model-centric workflow
- Rarely robust infrastructure, often monolithic and tangled
- Challenges in deploying systems and integration with monitoring, streams etc

- Shifting to pipeline-centric workflow challenging
- Requires writing robust programs, slower, less exploratory
- Standardized, modular infrastructure

- Big conceptual leap, major hurdle to adoption

O'Leary, Katie, and Makoto Uchida. "Common problems with Creating Machine Learning Pipelines from Existing Code." Proc. Third Conference on Machine Learning and Systems (MLSys) (2020).

TRADITIONAL VS AI-BASED SOFTWARE SYSTEMS

(deductive vs inductive reasoning)

COMPLEXITY IN ENGINEERED SYSTEMS

- Automobile: ~30,000 parts; Airplane: ~3,000,000 parts
- MS Office: ~ 40,000,000 LOCs; Debian: ~ 400,000,000 LOCs
- How do we build such complex systems?

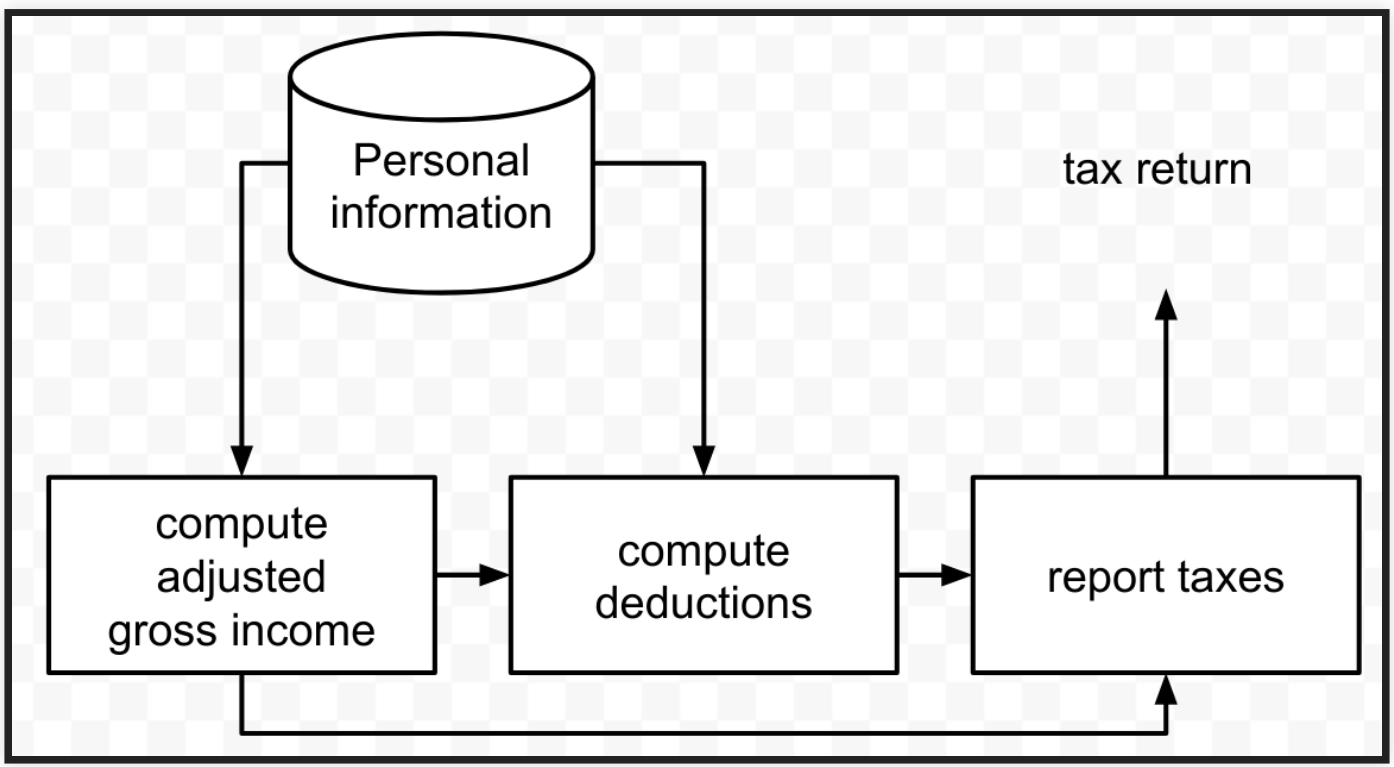
MANAGING COMPLEXITY IN SOFTWARE

- **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

```
/**
 * compute deductions based on provided adjusted
 * gross income and expenses in customer data.
 *
 * see tax code 26 U.S. Code A.1.B, PART VI
 *
 * Adjusted gross income must be positive;
 * returned deductions are not negative.
 */
float computeDeductions(float agi, Expenses expenses) {
  ...
}
```

DIVIDE AND CONQUER

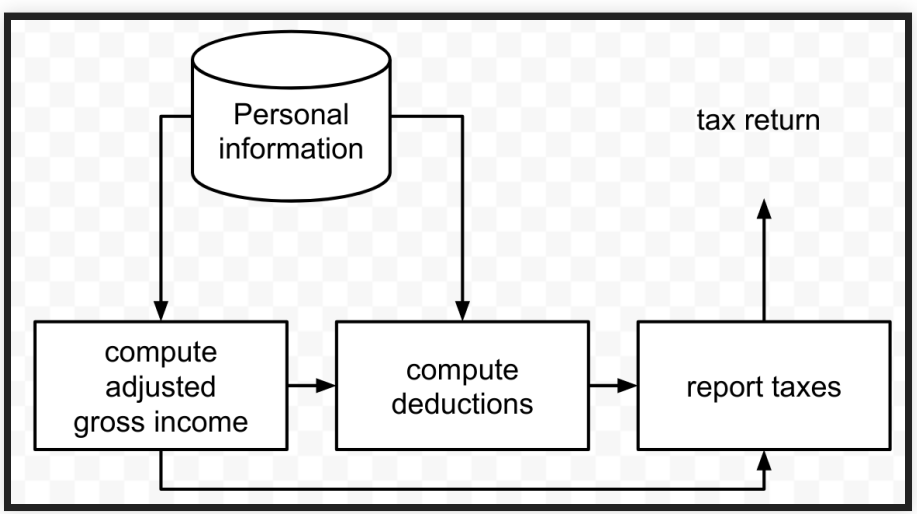
- Human cognitive ability is limited
- Decomposition of software necessary to handle complexity
- Allows division of labor
- Deductive reasoning, using logic



DEBUGGING AND ASSIGNING BLAME

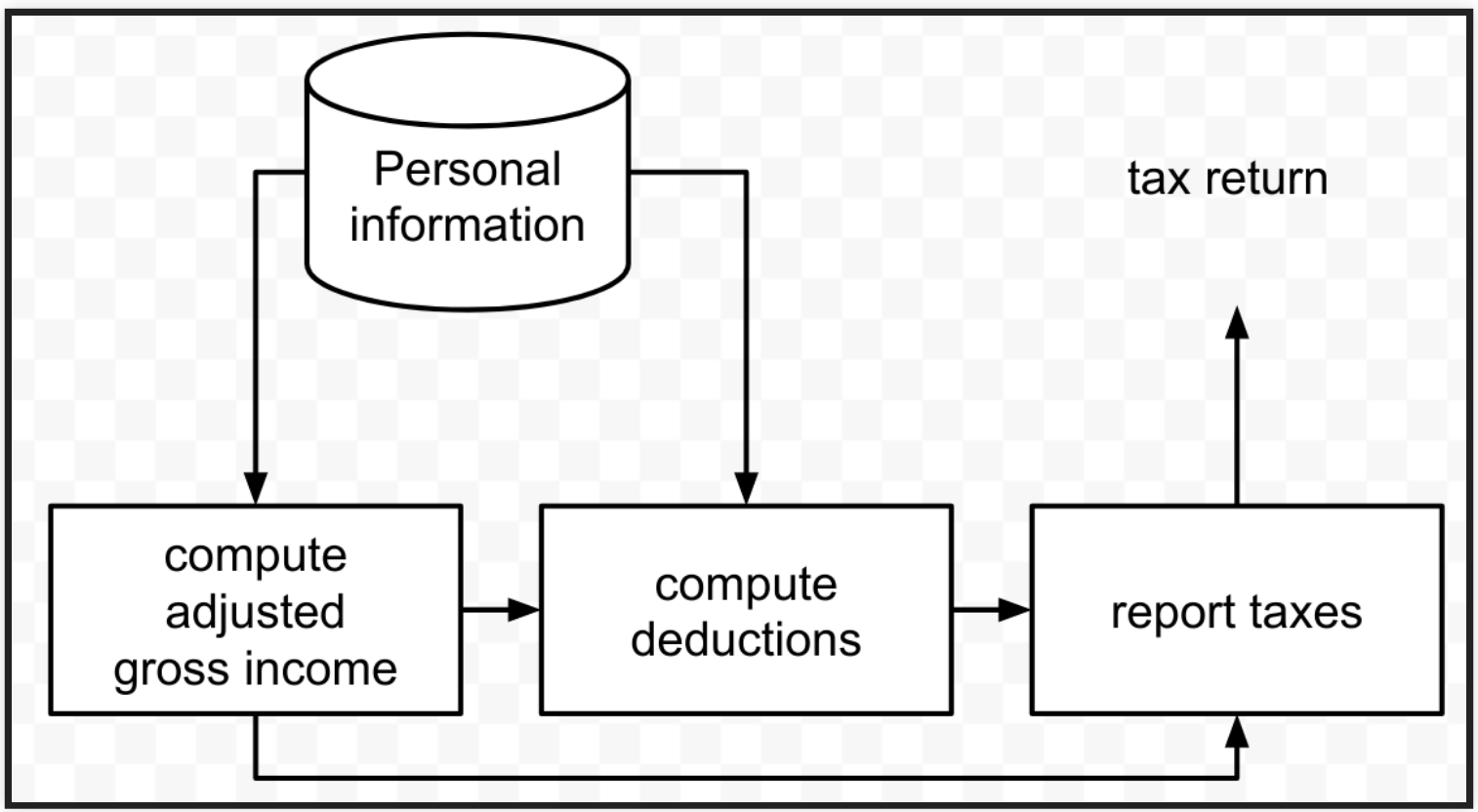
- Each component has own specification
- For each input, specification indicates whether output correct

```
/**
 * compute deductions based on provided adjusted
 * gross income and expenses in customer data.
 *
 * see tax code 26 U.S. Code A.1.B, PART VI
 */
float computeDeductions(float agi, Expenses expenses);
```



STRICT CORRECTNESS ASSUMPTION

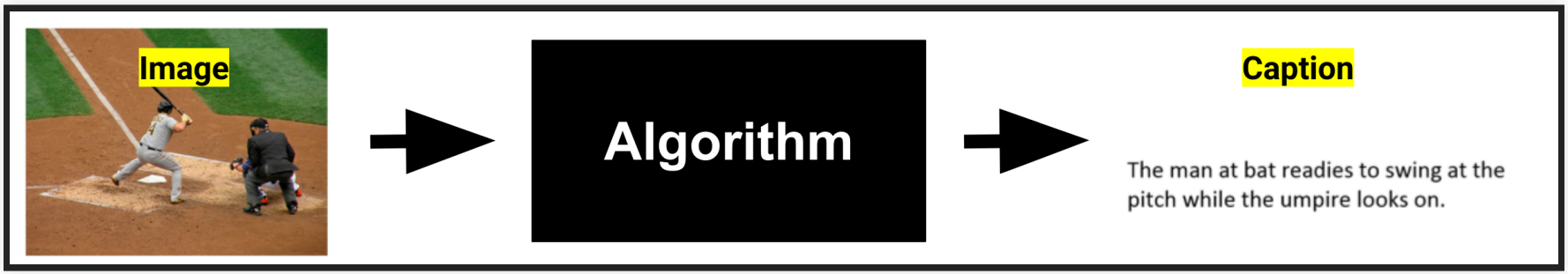
- Specification determines which outputs are correct/wrong
- Not "pretty good", "95% accurate", or "correct for 98% of all users"
- A single wrong result indicates a bug in the system



Speaker notes

A single wrong tax prediction would be a bug. No tolerance of occasional wrong predictions, approximations, nondeterminism.

IMAGE CAPTIONING ALGORITHM

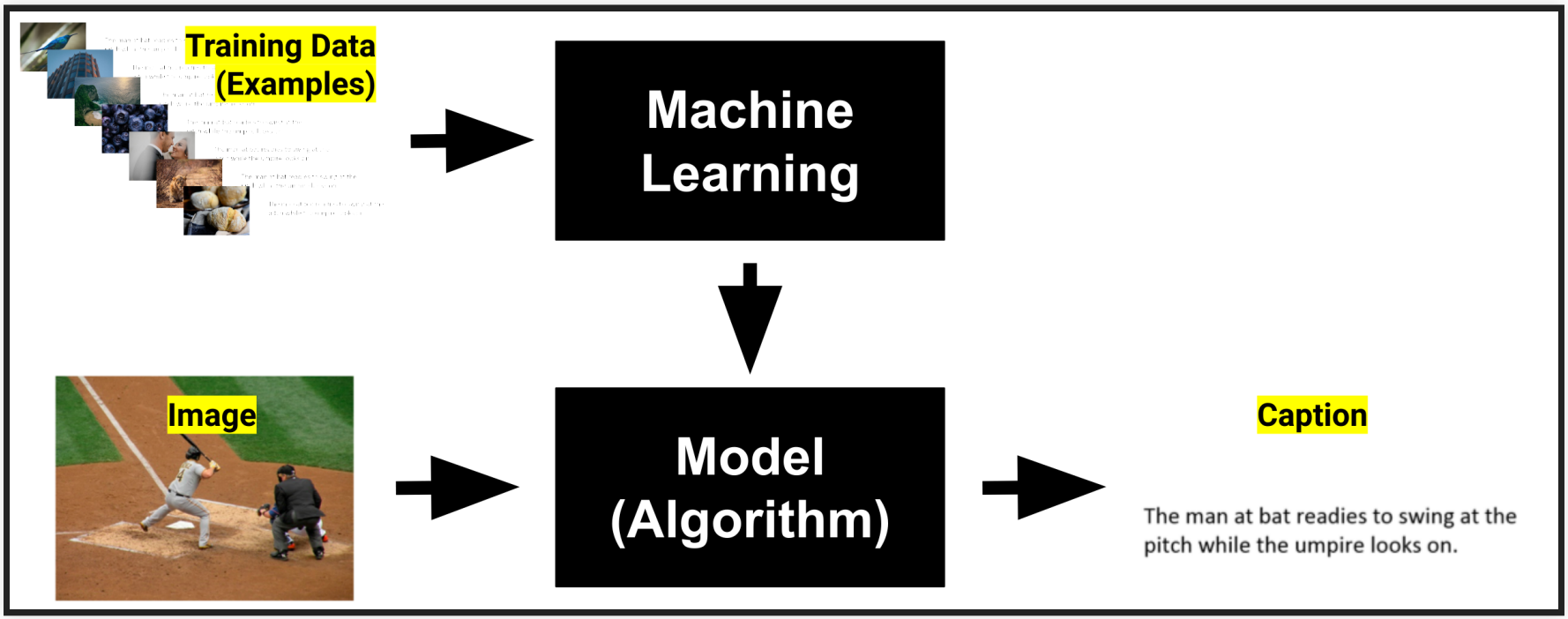



```
/**
  ????
*/
String getCaption(Image img);
```

Speaker notes

We do not know how to program this or specify this. No way of saying whether caption is "correct" for input, but defer to human judgement.

LEARNING IMAGE CAPTIONING ALGORITHM



Learning rules by fitting to examples, no specification, inductive reasoning

Speaker notes

"Rules"/algorithm learned from data. Still no specification. Best fit to given training data.

CORRECTNESS OF MODEL?

All models are wrong, but some are useful. -- George Box

Image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: troubleshooting integrative machine learning systems." In Proc. AAAI. 2017.

Speaker notes

Human judgment needed. Furthermore, a single bad example is not a problem.

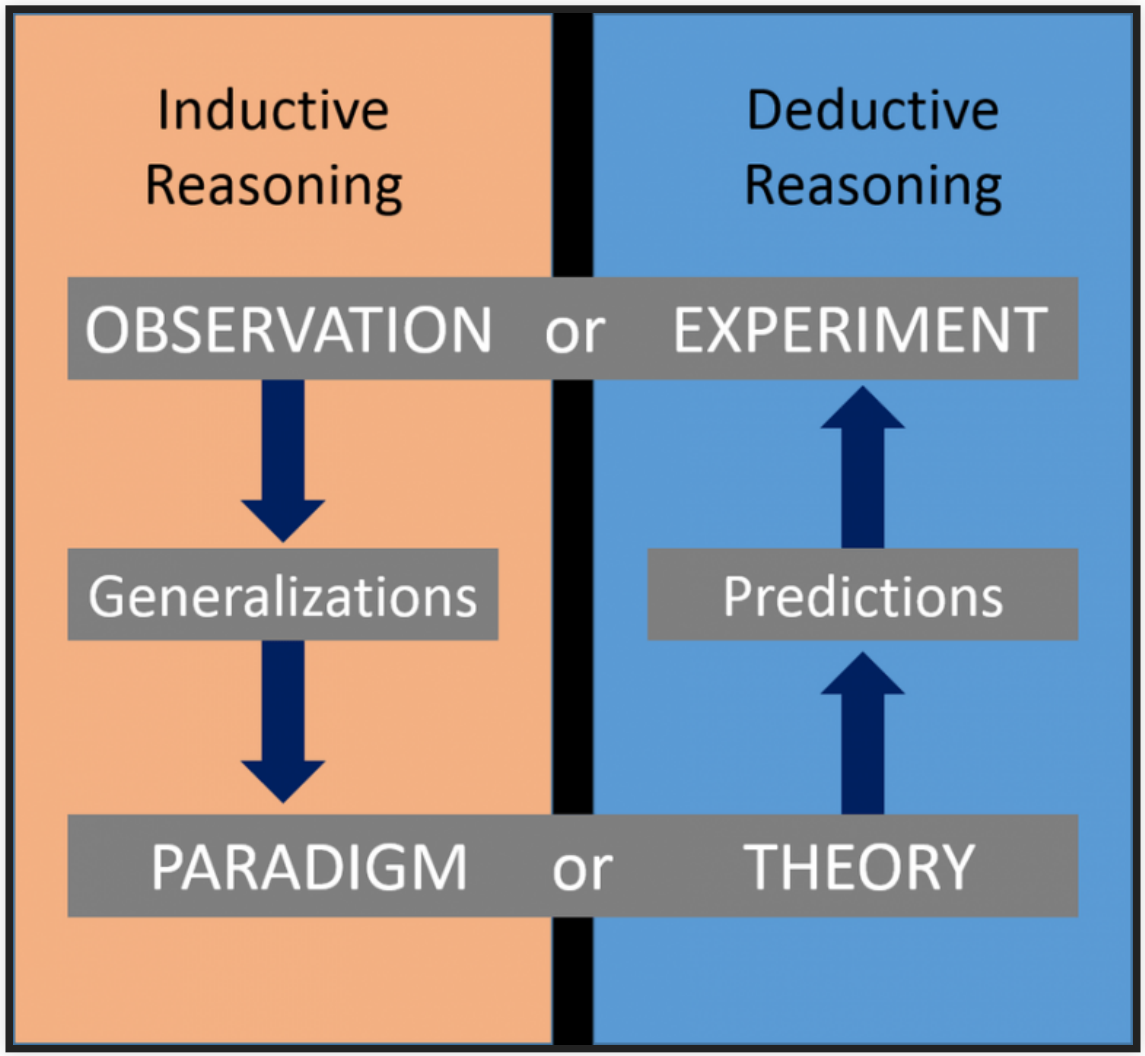
WEAK CORRECTNESS ASSUMPTIONS

- Often no reliable ground truth (e.g. human judgment)
- Accepting that mistakes will happen, hopefully not to frequently; "95% accuracy" may be pretty good
- More confident for data similar to training data

SPECIFICATIONS IN MACHINE LEARNING?

- Usually clear specifications do not exist -- we use machine learning exactly because we do not know the specifications
- Can define correctness for some data, but not general rules; sometimes can only determine correctness after the fact
- Learning for tasks for which we cannot write specifications
    - Too complex
    - Rules unknown
- ML will learn rules/specifications (inductive reasoning), often not in a human-readable form, but are those the right ones?

- Usually goals used instead --> maximize a specific objective



(Daniel Miessler, CC SA 2.0)

DEDUCTIVE REASONING

- Combining logical statements following agreed upon rules to form new statements
- Proving theorems from axioms
- From general to the particular
- *mathy reasoning, eg. proof that π is irrational*

- Formal methods, classic rule-based AI systems, expert systems

INDUCTIVE REASONING

- Constructing axioms from observations
- Strong evidence suggests a rule
- From particular to the general
- *sciency reasoning, eg. finding laws of nature*

- Most modern machine learning systems, statistical learning

CONSEQUENCES FROM LACK OF SPECIFICATIONS

Speaker notes

Breaks many traditional assumptions and foundations for compositional reasoning and divide and conquer

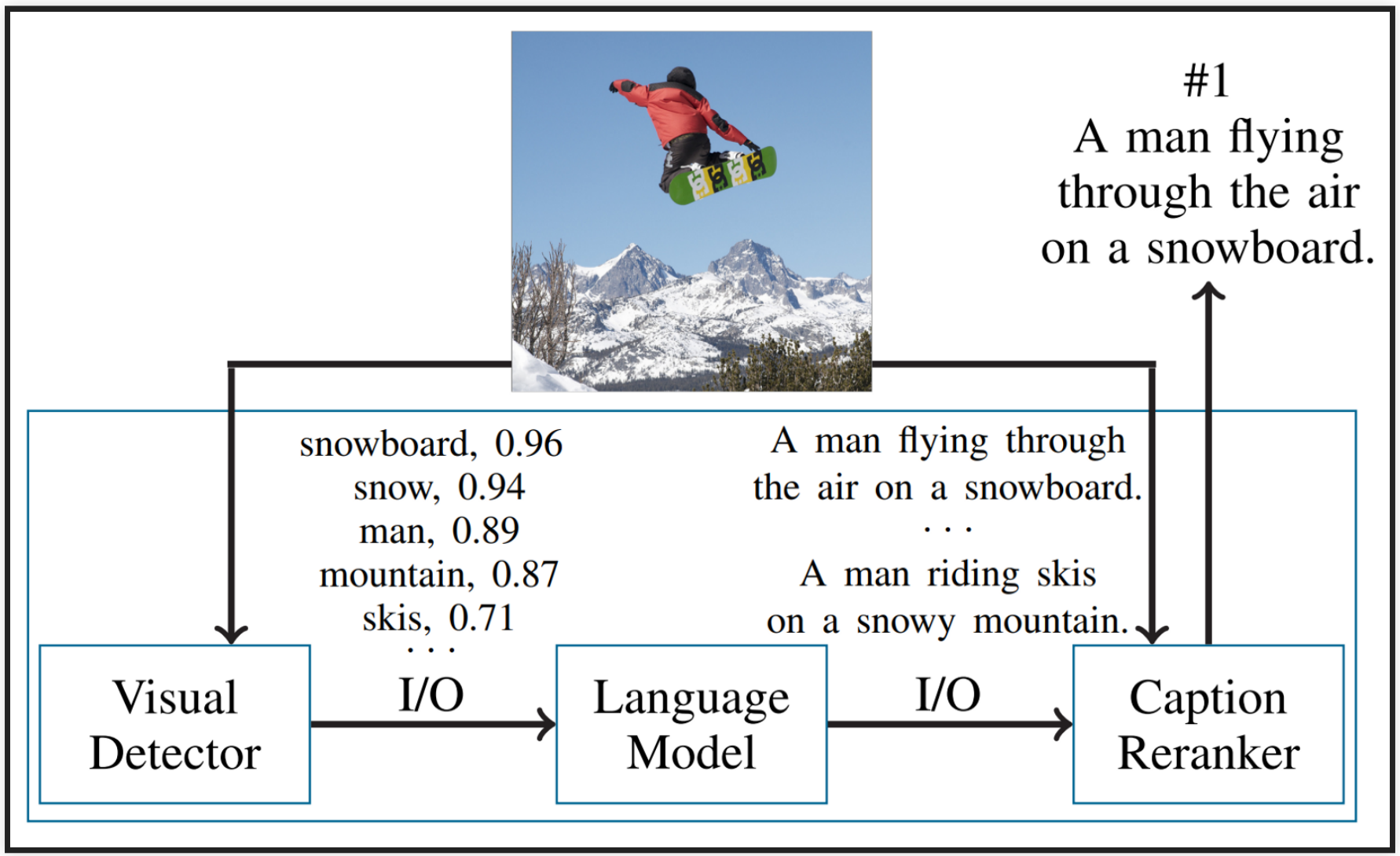
Poorly understood interactions between models: Ideally, develop models separately & compose together. In general, must train & tune together.

DECOMPOSING THE IMAGE CAPTIONING PROBLEM?

Speaker notes

Using insights of how humans reason: Captions contain important objects in the image and their relations. Captions follow typical language/grammatical structure

STATE OF THE ART DECOMPOSITION (IN 2015)



Example and image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: troubleshooting integrative machine learning systems." In Proc. AAAI. 2017.

BLAME ASSIGNMENT?

Visual Detector		Language Model	Caption Reranker
1. teddy	0.92	1. A teddy bear.	1. A blender sitting on top of a cake.
2. on	0.92	2. A stuffed bear.	2. A teddy bear in front of a birthday cake.
3. cake	0.90	...	3. A cake sitting on top of a blender.
4. bear	0.87	108. A blender sitting on top of a cake.	
5. stuffed	0.85		
...			
15. blender	0.57		

Example and image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: troubleshooting integrative machine learning systems." In Proc. AAAI. 2017.

NONMONOTONIC ERRORS

Visual Detector

teddy	0.92
computer	0.91
bear	0.90
wearing	0.87
keyboard	0.84
glasses	0.63

1. A teddy bear sitting on top of a computer.

Fixed Visual Detector

teddy	1.0
bear	1.0
wearing	1.0
keyboard	1.0
glasses	1.0

1. a person wearing glasses and holding a teddy bear sitting on top of a keyboard.

Example and image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: troubleshooting integrative machine learning systems." In Proc. AAAI. 2017.

TAKEAWAY: SHIFT IN DESIGN THINKING?

Breaking traditional decomposition and reasoning strategies...

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?

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

SUMMARY

- ML changes many engineering assumptions; from deductive to inductive reasoning; consequences for composition and abstraction
- Production AI-enabled systems require a *whole system perspective*, beyond just the model
- Engineering pipelines not models
- Large design space for user interface (intelligent experience): forcefulness, frequency, telemetry
- Quality at a *system* level: safety beyond the model, beyond accuracy

RECOMMENDED READINGS

- Wagstaff, Kiri. "Machine learning that matters." In Proceedings of the 29th International Conference on Machine Learning, (2012).
- 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.
- Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: troubleshooting integrative machine learning systems." In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1017-1025. 2017.
- O'Leary, Katie, and Makoto Uchida. "Common problems with Creating Machine Learning Pipelines from Existing Code." Proc. Third Conference on Machine Learning and Systems (MLSys) (2020).
- Blog post: On the process for building software with ML components