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

TRADITIONAL VS AI-BASED SOFTWARE SYSTEMS

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

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

CONTRACTS IN ML?

"DeepVisotron™,℠,® detects 1000 object categories with less than 1% errors."

Q. Is this the same kind of contract as in software?

"Two big challenges in machine learning", ICML 2015, Leon Bottou, Facebook

(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!

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?

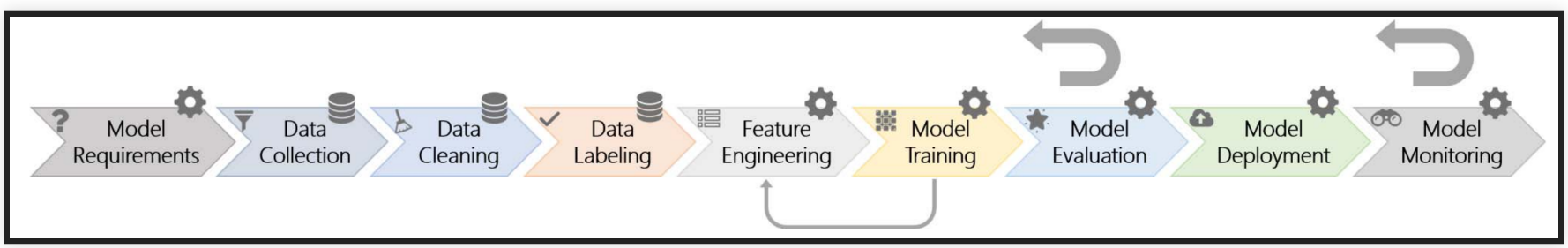
For design, implementation, testing, deployment, operations?

AI-ENABLED SYSTEMS

WHOLE SYSTEM PERSPECTIVE

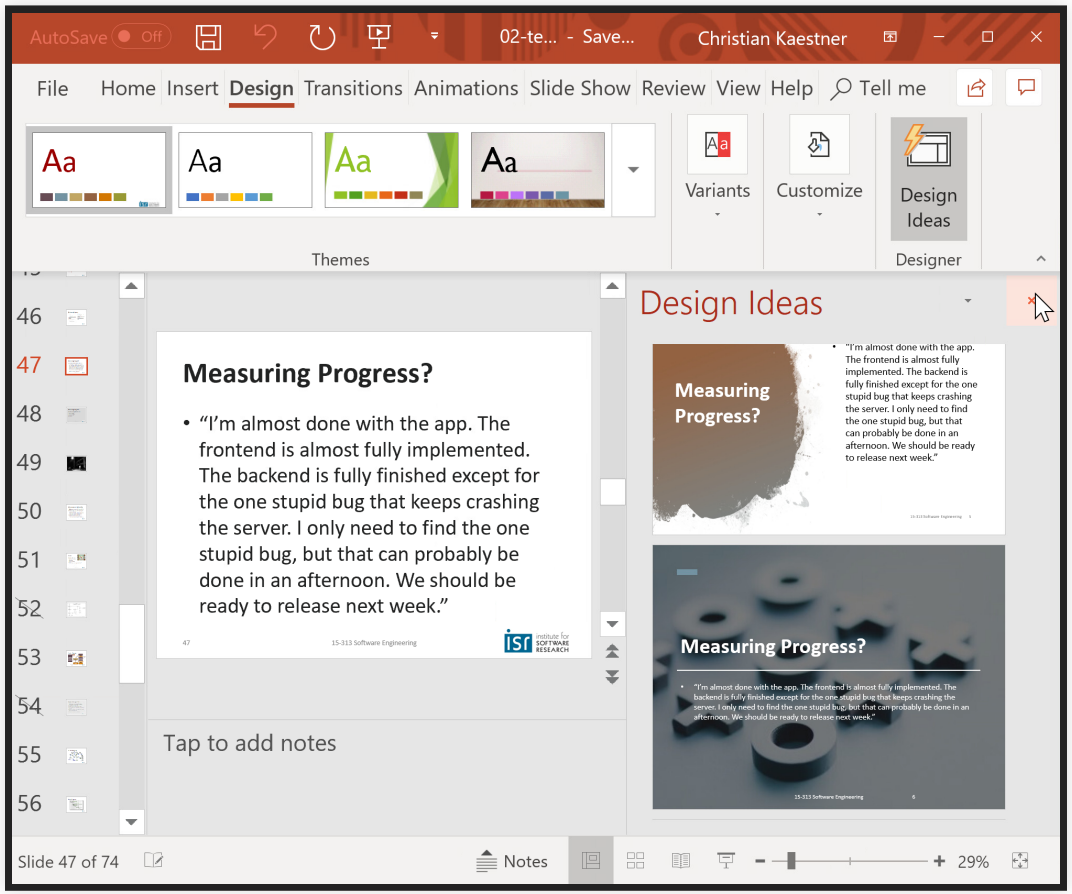
- A model is just one component of a larger system
- Also pipeline to build the model
- Also infrastructure to deploy, update, and serve the model
- Integrating the model with the rest of the system functionality
- User interaction design, dealing with mistakes
- Interaction with other stakeholders,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

FALL DETECTION DEVICES

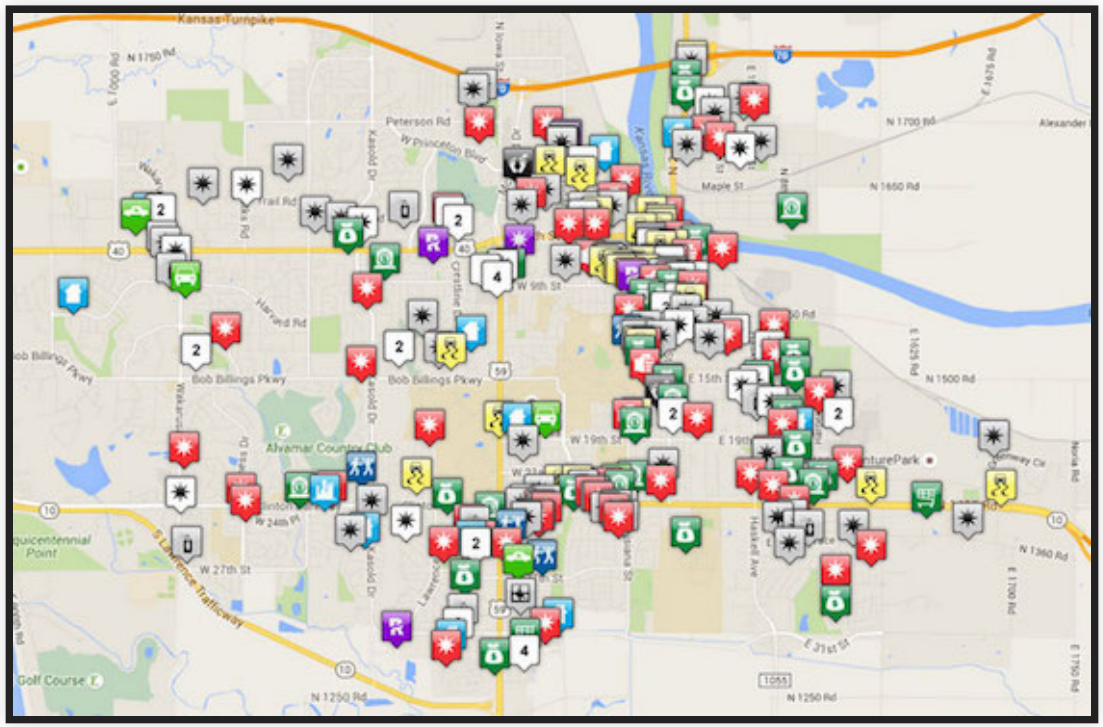
(various devices explored, including smart watches, hearing aids, and wall and floor sensors)

Read more: How fall detection is moving beyond the pendant, MobiHealthNews, 2019

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.

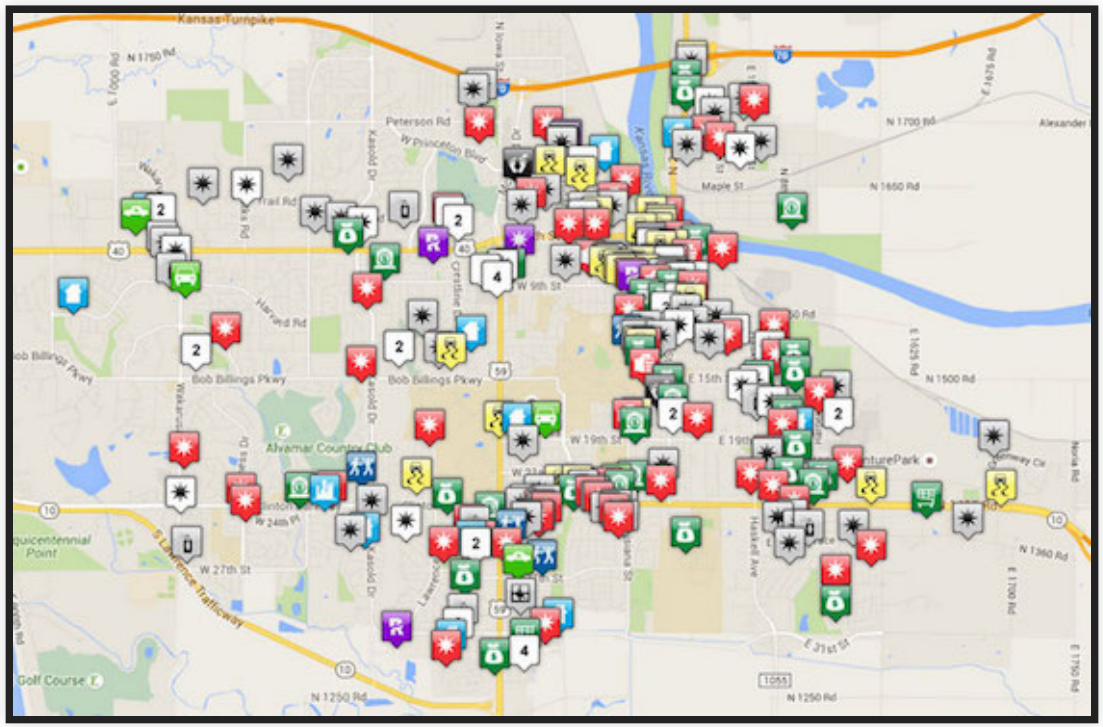
CRIME PREDICTION



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

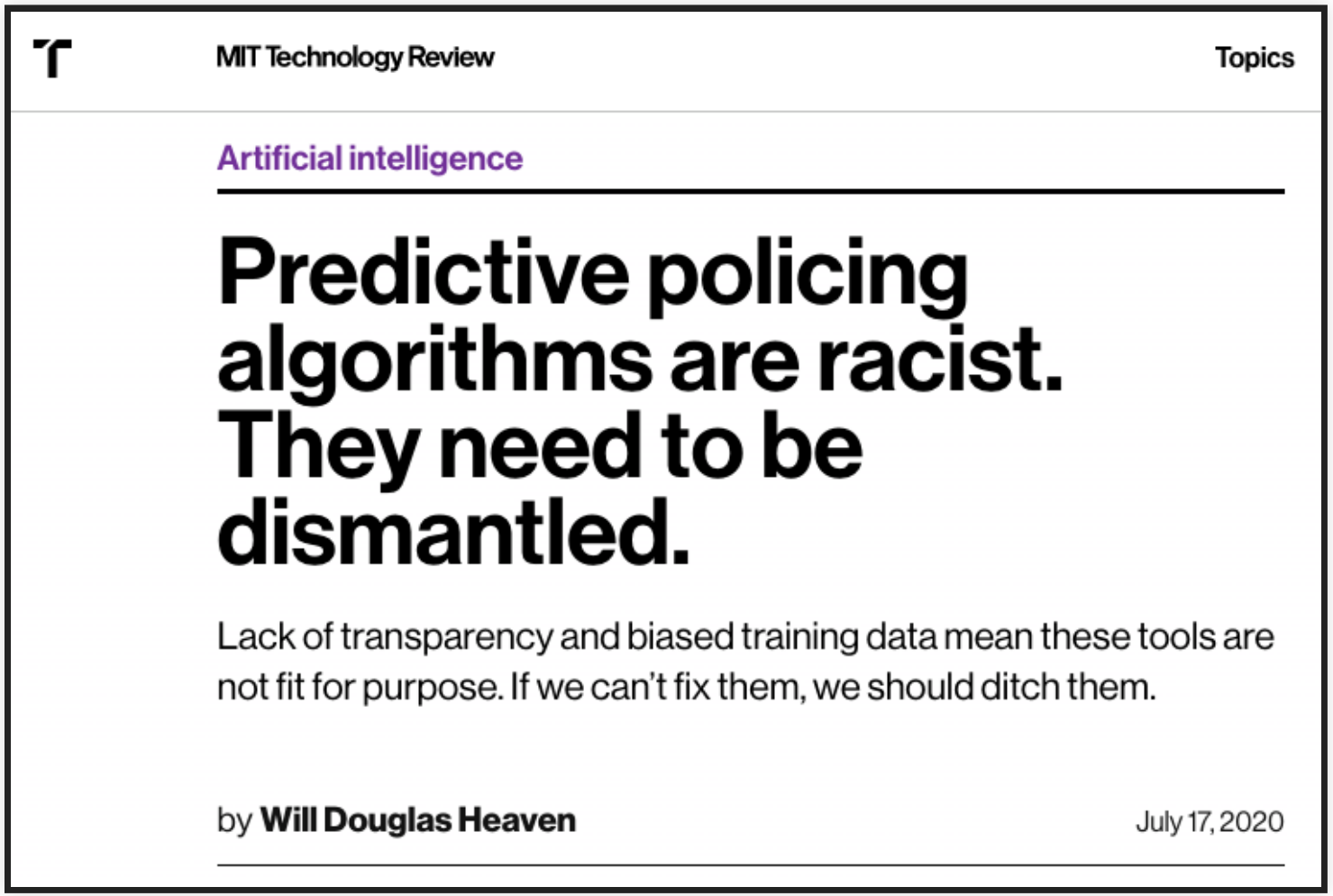
Q. What could possibly go wrong?

FEEDBACK LOOP



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

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

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

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
- ...

COMPONENTS OF AN AI-ENABLED SYSTEM

(Using Hulten's Terminology)

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

ELEMENTS OF AN INTELLIGENT SYSTEM

ELEMENTS OF AN INTELLIGENT SYSTEM

- Meaningful objective: Goals, requirements, business case

ELEMENTS OF AN INTELLIGENT SYSTEM

- Meaningful objective: Goals, requirements, business case
- Intelligent experience: User interactions -- presenting model predictions to user; eliciting & collecting feedback (telemetry)

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

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

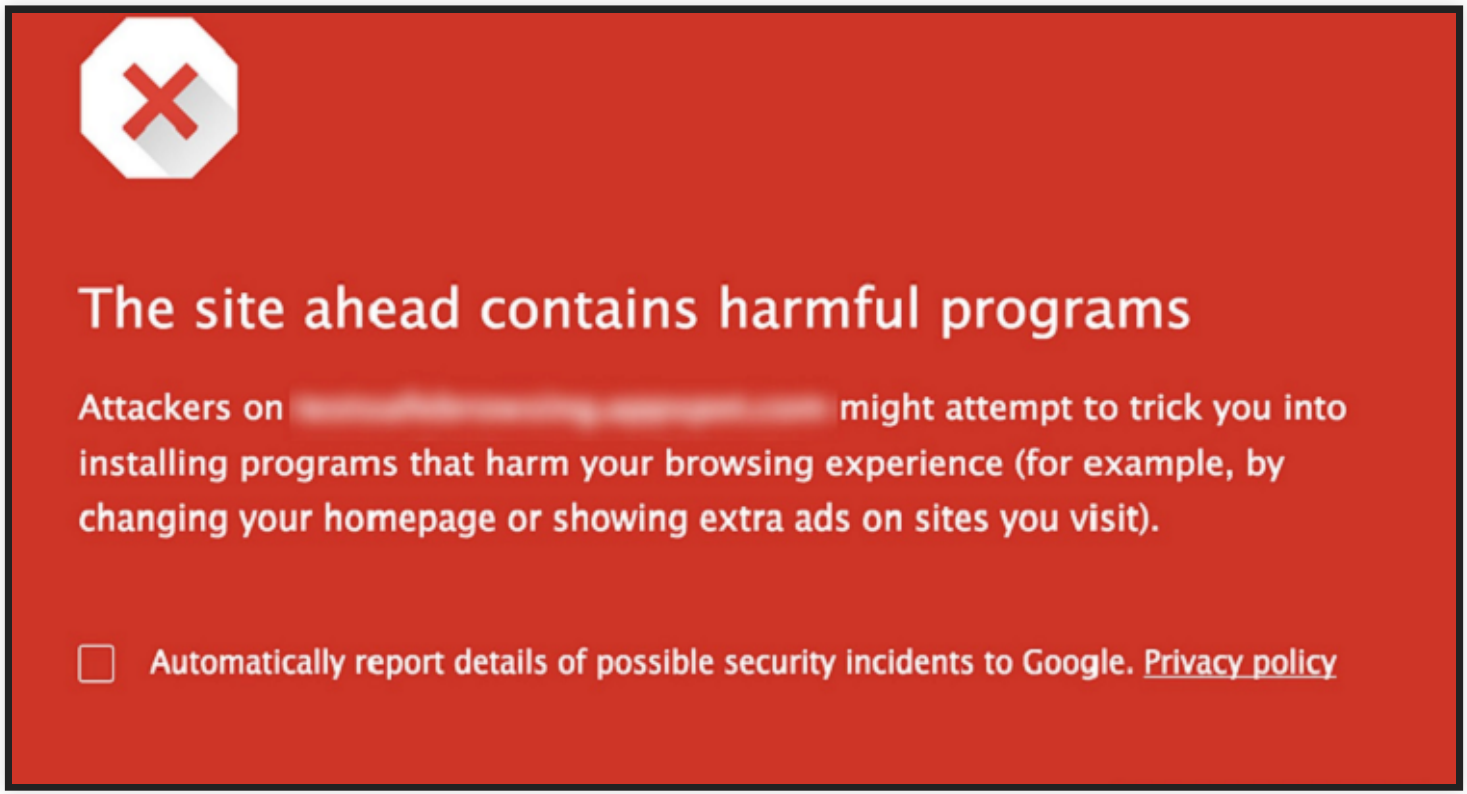
ELEMENTS OF AN INTELLIGENT SYSTEM

- 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

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 outcomes:
    - Achieving objectives
    - Protection from mistakes
    - Collecting data for training

ANOTHER EXAMPLE: SAFE BROWSING FEATURE



MEANINGFUL OBJECTIVES

DEFINING MEANINGFUL OBJECTIVES

- What is the system trying to achieve?
- **System** objective, not model qualities

DEFINING MEANINGFUL OBJECTIVES

- What is the system trying to achieve?
- **System** objective, not model qualities
- Properties of meaningful objectives
    - Measurable: Enables tracking & objective comparison
    - Achievable: Possible to achieve in time-to-market
    - Communicable: Transparent & comprehensible to stakeholders

DEFINING MEANINGFUL OBJECTIVES

- What is the system trying to achieve?
- **System** objective, not model qualities
- Properties of meaningful objectives
    - Measurable: Enables tracking & objective comparison
    - Achievable: Possible to achieve in time-to-market
    - Communicable: Transparent & comprehensible to stakeholders
- Q. What are the objectives of a safe browsing feature?

DEFINING MEANINGFUL OBJECTIVES

- What is the system trying to achieve?
- **System** objective, not model qualities
- Properties of meaningful objectives
    - Measurable: Enables tracking & objective comparison
    - Achievable: Possible to achieve in time-to-market
    - Communicable: Transparent & comprehensible to stakeholders
- Q. What are the objectives of a safe browsing feature?
    - "Prevent users from being hacked"

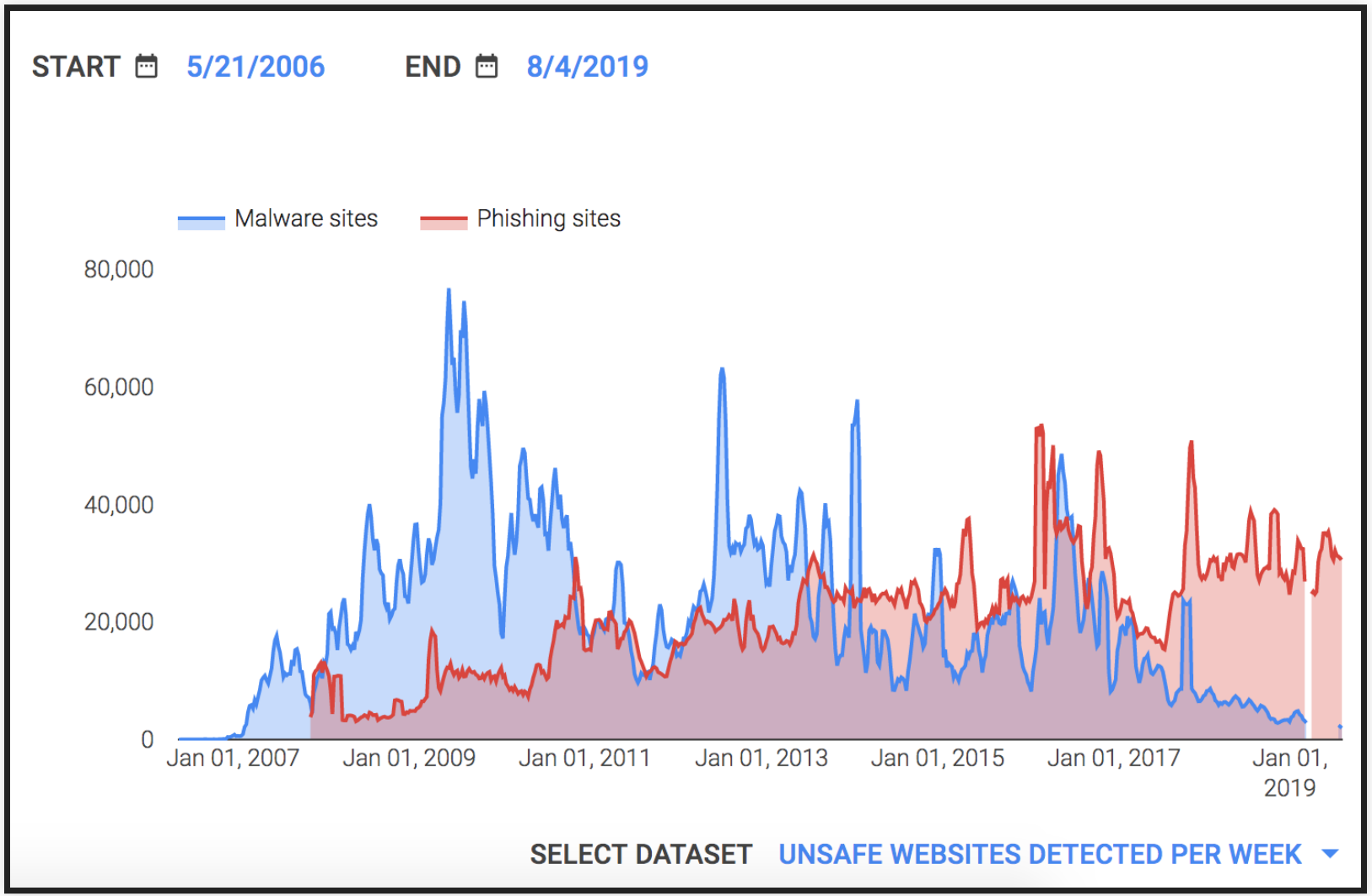
DEFINING MEANINGFUL OBJECTIVES

- What is the system trying to achieve?
- **System** objective, not model qualities
- Properties of meaningful objectives
    - Measurable: Enables tracking & objective comparison
    - Achievable: Possible to achieve in time-to-market
    - Communicable: Transparent & comprehensible to stakeholders
- Q. What are the objectives of a safe browsing feature?
    - "Prevent users from being hacked"
    - "Minimize users' inconvenience"

DEFINING MEANINGFUL OBJECTIVES

- What is the system trying to achieve?
- **System** objective, not model qualities
- Properties of meaningful objectives
    - Measurable: Enables tracking & objective comparison
    - Achievable: Possible to achieve in time-to-market
    - Communicable: Transparent & comprehensible to stakeholders
- Q. What are the objectives of a safe browsing feature?
    - "Prevent users from being hacked"
    - "Minimize users' inconvenience"
    - (Are these good? Can we do better?)

MEASURABLE


START 5/21/2006
END 8/4/2019
Malware sites
Phishing sites
80,000
60,000
40,000
20,000
0
Jan 01, 2007
Jan 01, 2009
Jan 01, 2011
Jan 01, 2013
Jan 01, 2015
Jan 01, 2017
Jan 01, 2019
SELECT DATASET
UNSAFE WEBSITES DETECTED PER WEEK

ACHIEVABLE?


Connect Password Manag
https://www.mcafeesecure.com/get-connect/
Secure Browser
Open financial accounts through a 100% secure browser.
Site Monitoring
Avoid fake sites with monitoring of over 9.000
McAfee SECURE
100% SECURE
LIKE WE HAVEN'T HEARD THAT ONE BEFORE

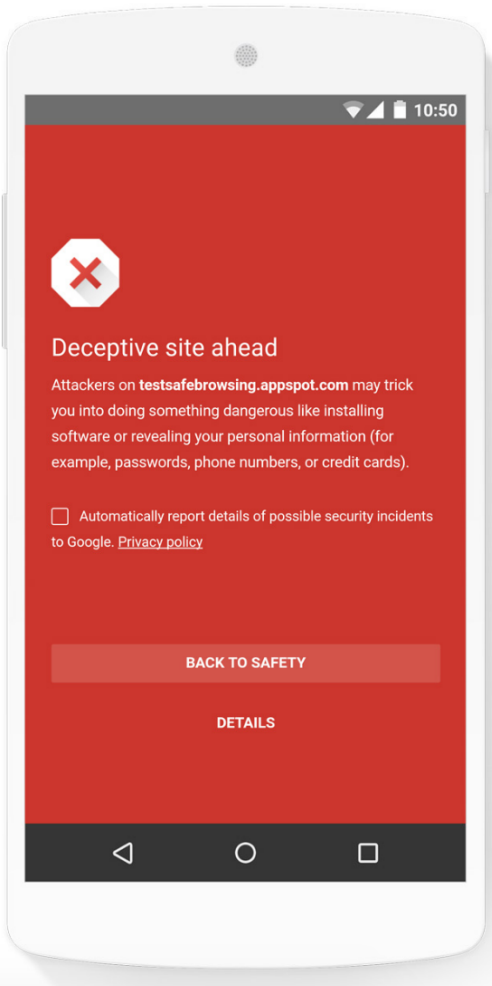
COMMUNICABLE

SAFETY FIRST

Keeping over four billion devices safer.

Google Safe Browsing helps protect over four billion devices every day by showing warnings to users when they attempt to navigate to dangerous sites or download dangerous files. Safe Browsing also notifies webmasters when their websites are compromised by malicious actors and helps them diagnose and resolve the problem so that their visitors stay safer. Safe Browsing protections work across Google products and power safer browsing experiences across the Internet.

Our Transparency Report includes details on the threats that Safe Browsing identifies. The Transparency Report includes our Site Status diagnostic tool that you can use to see whether a site currently contains content that Safe Browsing has determined to be dangerous.



USER INTERACTIONS (INTELLIGENT EXPERIENCES)

PRESENTING INTELLIGENCE

PRESENTING INTELLIGENCE

- Automate: Take action on user's behalf

PRESENTING INTELLIGENCE

- Automate: Take action on user's behalf
- Prompt: Ask the user if an action should be taken

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

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

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

When designing an intelligent experience 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?

PRESENTING INTELLIGENCE: SAFE BROWSING

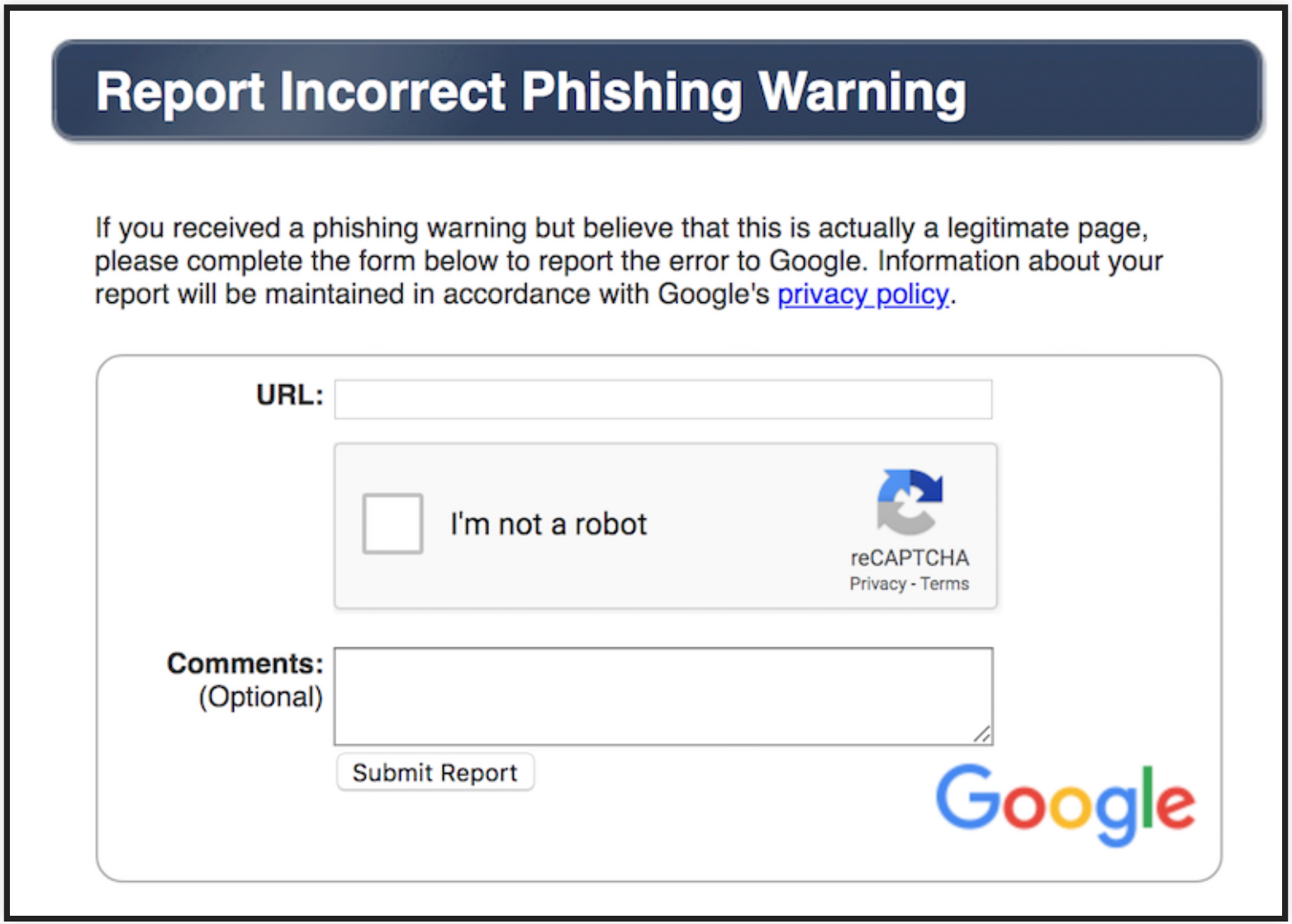
- Compare against more forceful, "automate" option
  - What are trade-offs between them?
  - If model makes a mistake, what kind of damage can it cause? Which one is easier to recover from?

FEEDBACK (TELEMETRY)

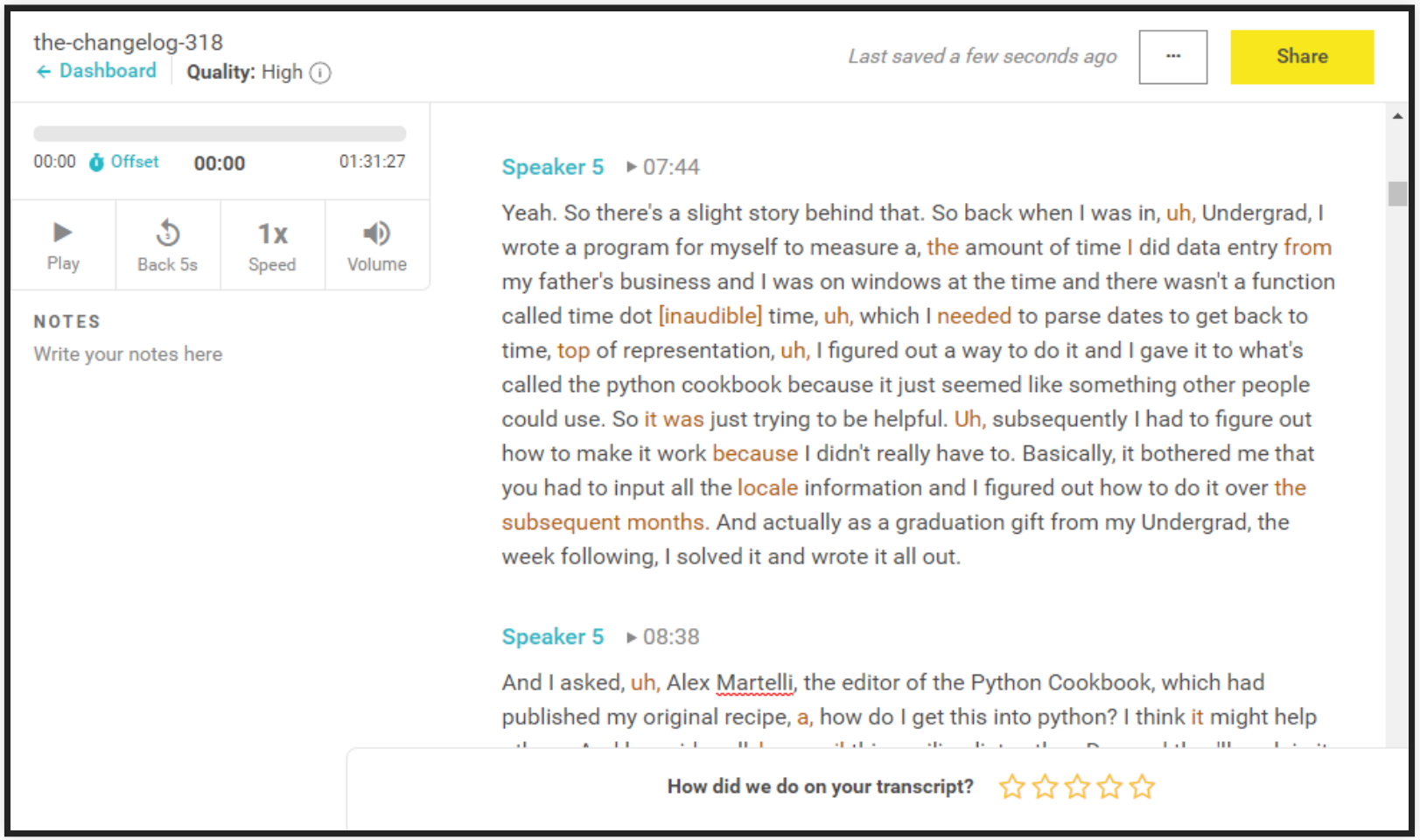
- To design good interactions we need to know how we are doing...
- How many predictions are ignored?
- How many actions are reversed?
- How often does the user ask for extra predictions?
- How much value do users get out of predictions?
- How much are we supporting the system's goals?
- How much cost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

COLLECTING FEEDBACK



OUTLOOK: TELEMETRY DESIGN



More on this later...

BREAKOUT DISCUSSION

CASE STUDY: FALL DETECTION

- What are meaningful objectives of the system?
- How do we present the intelligence to the user?
    - How forceful should it be?
    - What are potential costs of mistakes?
- How do we collect data to continue to learn from users and mistakes?

SYSTEM QUALITIES VS MODEL ACCURACY

SYSTEMS HAVE GOALS

... selling stuff, increasing engagement, encouraging responsible behavior

Model predictions support those goals

more in a later lecture

MORE ACCURATE PREDICTIONS MAY NOT BE THAT IMPORTANT

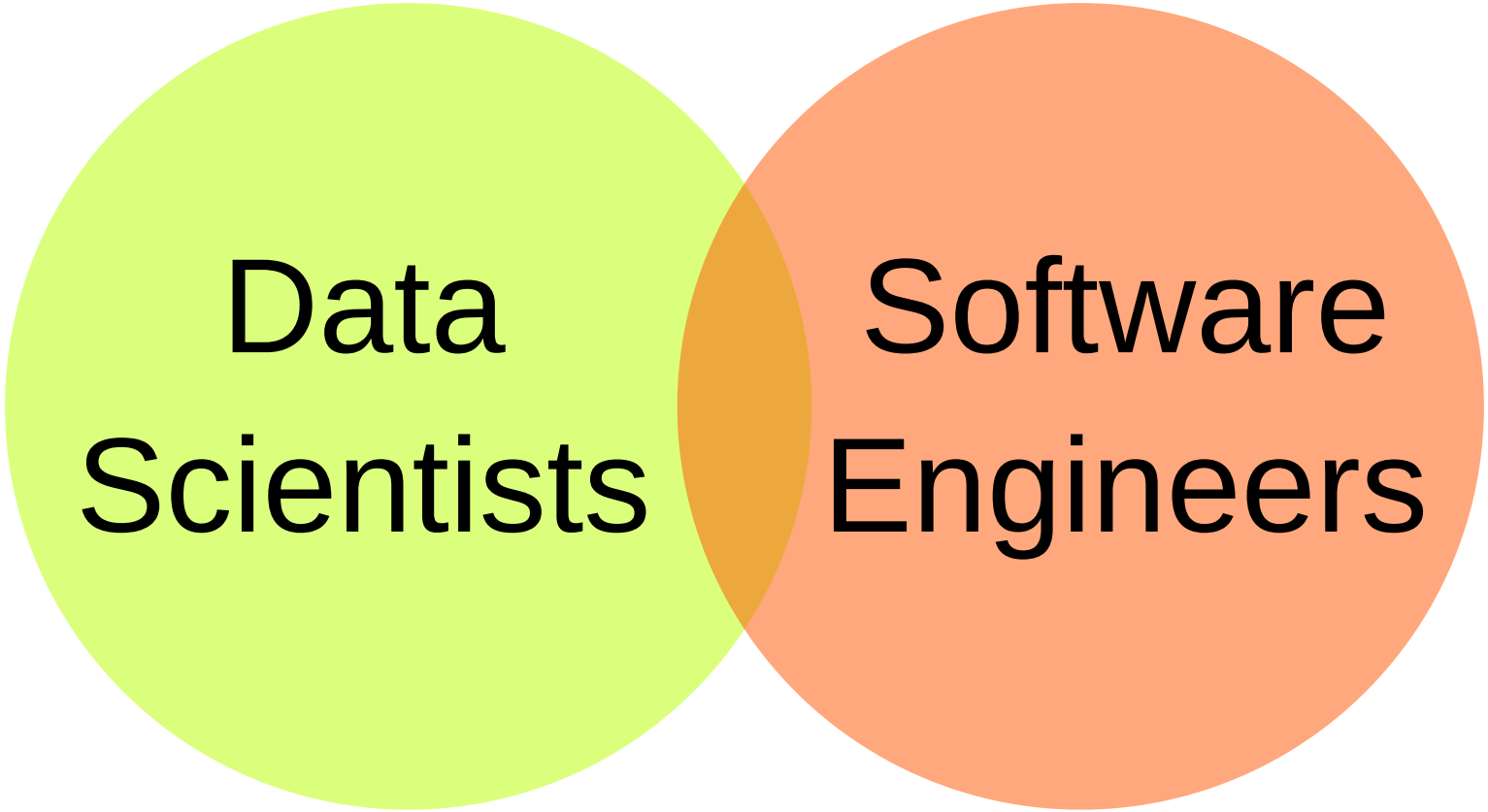
- "Good enough" may be good enough
- Prediction critical for system success or just an gimmick?
- Better predictions may come at excessive costs
    - need way more data, much longer training times
    - privacy concerns
- Better user interface ("experience") may mitigate many problems
    - e.g. explain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

BEYOND MODEL QUALITY

Many other aspects of a model's quality may matter when operating a system

- Learning time, inference time
- Incremental learning
- Explainability
- Model size
- Kinds of mistakes
- Fairness, privacy, security, robustness
- Reproducibility
- Maintainability

(More in a later lecture!)


Data
Scientists
Software
Engineers

SUMMARY

- Production AI-enabled systems require a *whole system perspective*, beyond just the model
- Components: Objectives, user interface, infrastructure, AI component, and operations
- Large design space for user interface (intelligent experience): forcefulness, frequency, telemetry
- Quality at a system level: safety beyond the model, beyond accuracy