

# ENGINEERING AI-ENABLED SYSTEMS WITH INTERDISCIPLINARY TEAMS

Christian Kästner

@p0nk

<https://ckaestne.github.io/seai/>

SEMLA'20

A portrait photograph of Christian Kästner, a man with light brown hair, wearing a red button-down shirt, standing outdoors in front of a large building.

# CHRISTIAN KÄSTNER

@p0nk

Associate Professor @ CMU

Interests:

- Software Engineering
- Highly-Configurable Systems & Configuration Engineering
- Sustainability and Stress in Open Source
- Software Engineering for ML-Enabled Systems

# SOFTWARE ENGINEERING FOR ML-ENABLED SYSTEMS

*Building, operating, and maintaining software systems  
with machine-learned components*

*with interdisciplinary collaborative teams of **data  
scientists and software engineers***

# SE FOR ML-ENABLED SYSTEMS != BUILDING MODELS

CO G4 playground.ipynb ☆

File Edit View Insert Runtime Tools Help Last edited on April 4

Comment Share

+ Code + Text Connect E

[ ]	1096	4	12	26	3	2	0
[ ]	235	4	4	23	1	2	0

525 rows × 6 columns

```
[ ] # learning a classifier whether the result will be nonZero  
from sklearn import tree  
  
classifier=tree.DecisionTreeClassifier(max_depth=8)  
classifier=classifier.fit(Xtrain, ynztrain)  
  
print(classifier.score(Xtrain, ynztrain))  
print(classifier.score(Xtest, ynztest))
```

0.8266666666666667  
0.7295238095238096

```
[ ] # learning a regression model only on the nonZero data (test is on all data and somewhat
```

```
from sklearn import tree

predictor=tree.DecisionTreeRegressor(max_depth=8)
predictor=predictor.fit(XnzTrain,YnzTrain)

print(predictor.score(XnzTrain, YnzTrain))
print(predictor.score(Xtest, ytest))
```



0.9376379365613154  
-2.437397740412892

# SE FOR ML-ENABLED SYSTEMS != CODING ML FRAMEWORKS





# SE FOR ML-ENABLED SYSTEMS != ML FOR SE TOOLS

```
1 import numpy as np
2
3 start = -1
4 stop = 1
5
6 x = np.lins
    f linspace function
    f linspace(start, stop) function
    f linspace(stop, start) function
    f linspace(start, stop, sto... function
```

# SE FOR ML-ENABLED SYSTEMS

The screenshot shows a Microsoft Word document window. The ribbon at the top has tabs: File, Home, Insert, Design (which is selected), Transitions, Animations, Slide Show, Review, View, Help, Tell me, and two icons. Below the ribbon, there are sections for Themes and Designer.

**Themes:** Shows four preview cards for different theme variations, each featuring a red 'Aa' logo and a color bar.

**Designer:** Shows three buttons: Variants, Customize, and Design Ideas. The Design Ideas button is highlighted with a gray background.

**Slide Content:** The main slide (slide 47) has the title "Measuring Progress?" and a bullet point list. The list contains a single item: "I'm almost done with the app. The frontend is almost fully implemented. The backend is fully finished except for the one stupid bug that keeps crashing the server. I only need to find the one stupid bug, but that can probably be done in an afternoon. We should be ready to release next week."

**Design Ideas Sidebar:** This sidebar lists "Design Ideas" and displays two cards. The first card, titled "Measuring Progress?", has a dark background and a white quote: "I'm almost done with the app. The frontend is almost fully implemented. The backend is fully finished except for the one stupid bug that keeps crashing the server. I only need to find the one stupid bug, but that can probably be done in an afternoon. We should be ready to release next week." The second card is partially visible below it.

**Navigation:** On the left, a vertical navigation bar shows slide numbers from 46 to 54. At the bottom, there is a footer with the text "15-313 Software Engineering" and the ISR logo.

55



Tap to add notes

56



Slide 47 of 74



Notes



15-313 Software Engineering

6

- + 29%



# SE FOR ML-ENABLED SYSTEMS

the-changelog-318 Last saved a few seconds ago ... Share

← Dashboard Quality: High ⓘ

00:00 ⚡ Offset 00:00 01:31:27

Play Back 5s 1x Volume

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**

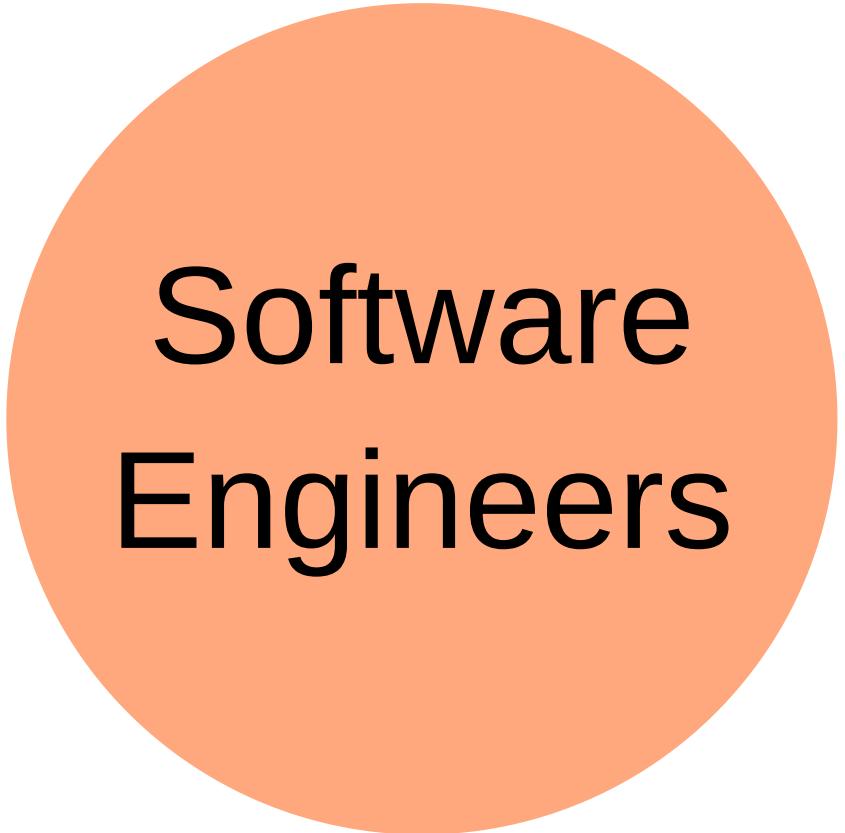
And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript? ☆☆☆☆☆





**Data  
Scientists**



**Software  
Engineers**

# SOFTWARE ENGINEERING

*Software engineering is the branch of computer science that creates practical, cost-effective solutions to computing and information processing problems, preferentially by applying scientific knowledge, developing software systems in the service of mankind.*

Engineering judgements under limited information and resources

A focus on design, tradeoffs, and the messiness of the real world

Many qualities of concern: cost, correctness, performance, scalability, security, maintainability, ...

"it depends..."

Mary Shaw. ed. [Software Engineering for the 21st Century: A basis for rethinking the curriculum](#). 2005.

# MOST ML COURSES/TALKS

Focus narrowly on modeling techniques or building models

Using notebooks, static datasets, evaluating accuracy

Little attention to software engineering aspects of building complete systems

The screenshot shows a Google Colab notebook interface. The title bar reads "G4 playground.ipynb" with a star icon. The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a note "Last edited on April 4". The toolbar on the left has icons for code (+ Code), text (+ Text), and other notebook operations. The main workspace displays a table with two rows of data and some Python code.

	1096	4	12	26	3	2	0
[ ]	235	4	4	23	1	2	0

525 rows × 6 columns

```
[ ] # learning a classifier whether the result will be nonZero
from sklearn import tree
classifier=tree.DecisionTreeClassifier(max_depth=8)
classifier=classifier.fit(Xtrain, ynztrain)

print(classifier.score(Xtrain, ynztrain))
print(classifier.score(Xtest, ynztest))
```



0.8266666666666667  
0.7295238095238096

```
[ ] # learning a regression model only on the nonZero data (test is on all data and somewhat  
  
from sklearn import tree  
  
predictor=tree.DecisionTreeRegressor(max_depth=8)  
predictor=predictor.fit(XnzTrain,YnzTrain)  
  
  
print(predictor.score(XnzTrain, YnzTrain))  
print(predictor.score(Xtest, ytest))
```



0.9376379365613154  
-2.437397740412892

# 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
- Starting to worry about fairness, robustness, ...

# SOFTWARE ENGINEER

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



**Data  
Scientists**



**Software  
Engineers**

A transcription interface with a timeline at the top showing 00:00, Offset, 00:00, and 01:31:27. Below the timeline are four buttons: Play, Back 5s, 1x Speed, and Volume.

## 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?

# A SOFTWARE ENGINEERING PERSPECTIVE ON ML

# WHAT'S DIFFERENT?

- Missing specifications
- Environment is important (feedback loops, data drift)
- Nonlocal and nonmonotonic effects
- Testing in production
- Data management, versioning, and provenance

# MISSING SPECIFICATIONS

*from deductive to inductive reasoning*

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

```
/**  
 *  
 */  
Boolean predictRecidivism(int age,  
                         List<Crime> priors,  
                         Gender gender,  
                         int timeServed,  
                         ...);
```

# ENVIRONMENT IS IMPORTANT

(feedback loops, data drift)

The image shows a YouTube channel page for 'FLAT EARTH CLUES' by Mark Sargent. The channel has 22 videos and 577,011 views, last updated on Dec 6, 2018. A red arrow points to the second video in the list.

**FLAT EARTH CLUES**  
INTRODUCTION BY MARK SARGENT

PLAY ALL

Start here! FLAT EARTH CLUES

22 videos • 577,011 views • Last updated on Dec 6, 2018

markksargent

SUBSCRIBE 73K

Flat Earth Clues Preface by the Editor - Mark Sargent [✓]  
markksargent

FLAT EARTH Clues Introduction - Mark Sargent [✓]  
markksargent

FLAT EARTH Clues Part 1 - Empty Theatre - Mark Sargent [✓]  
markksargent

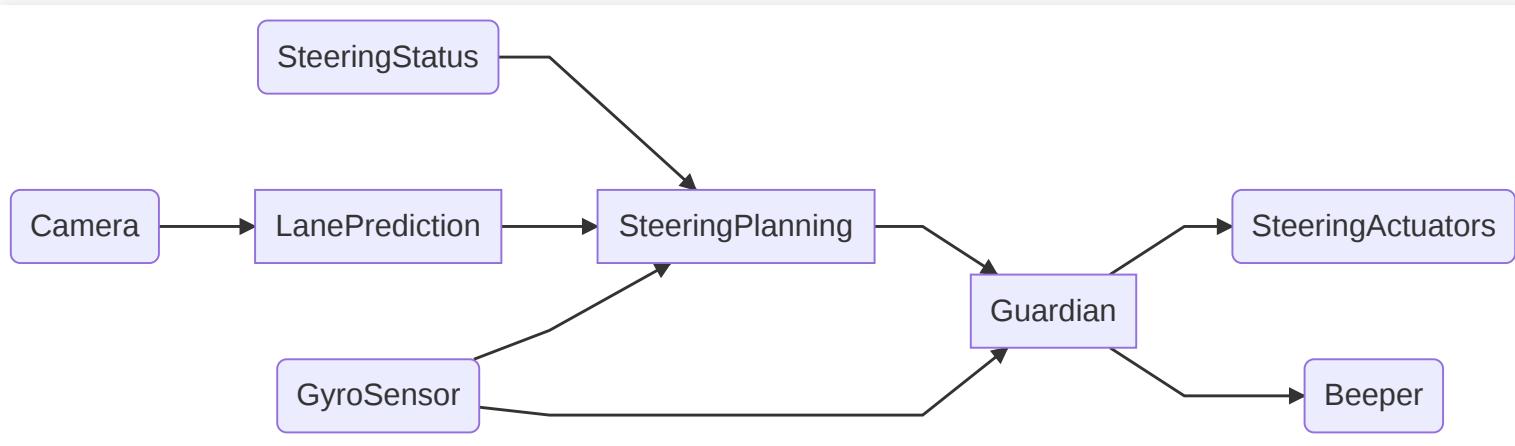
FLAT EARTH Clues Part 2 - Byrd Wall - Mark Sargent [✓]  
markksargent

FLAT EARTH Clues Part 3 - Map Makers - Mark Sargent [✓]  
markksargent



# NONLOCAL AND NONMONOTONIC EFFECTS

*multiple models in most systems*



# TESTING IN PRODUCTION



.#drian @ddowza · 26s

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



...



Tay Tweets ✅

@TayandYou



Follow

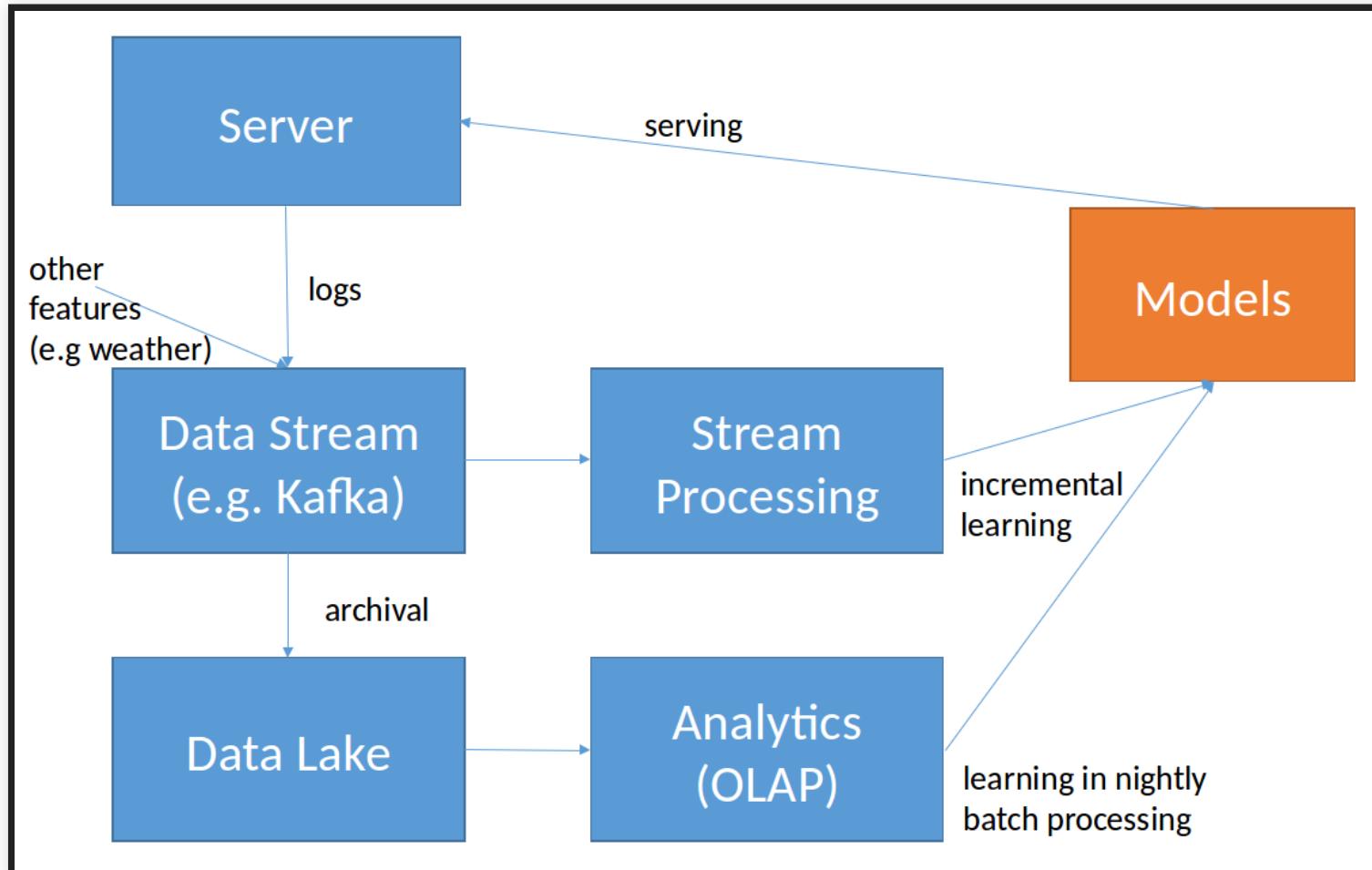
@ddowza not really sorry

12:29 PM - 24 Mar 2016



...

# DATA MANAGEMENT, VERSIONING, AND PROVENANCE



# BUT REALLY DIFFERENT?

# ML: MISSING SPECIFICATIONS

*from deductive to inductive reasoning*

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

```
/**  
 *  
 *  
 */  
Boolean predictRecidivism(int age,  
                         List<Crime> priorCrimes,  
                         Gender gender,  
                         int timeServed,  
                         ...);
```

## SOFTWARE ENGINEERING:

vague specs very common

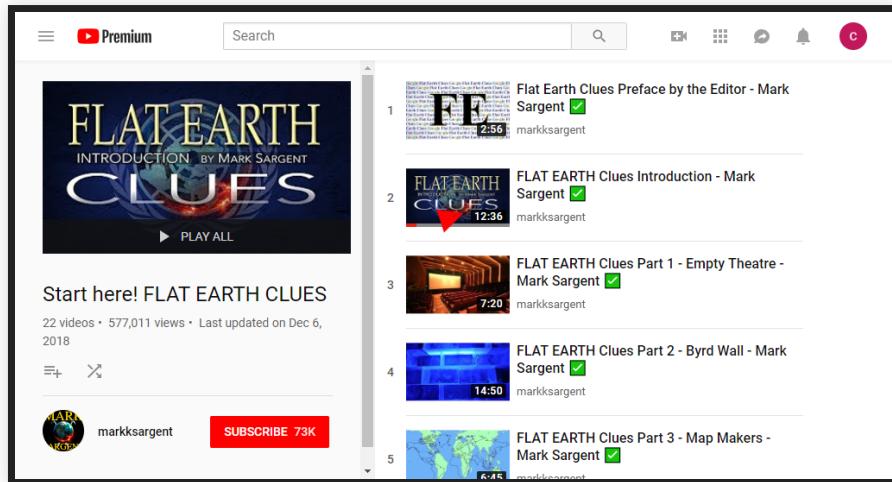
agile methods

safe systems from  
unreliable components

("ML is requirements  
engineering")

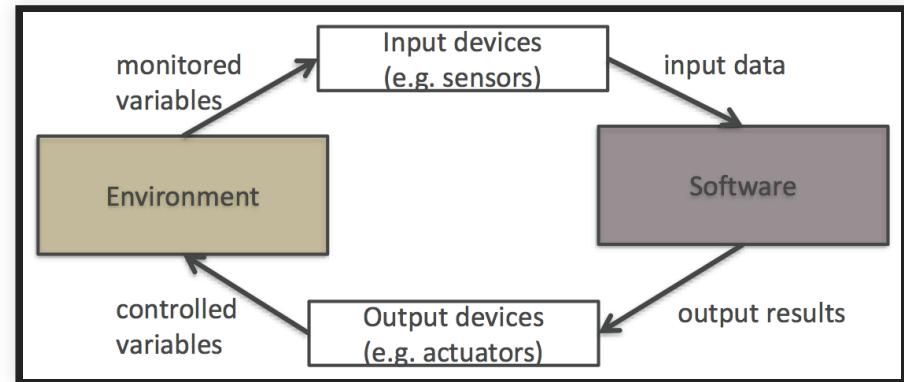
# AI: ENVIRONMENT IS IMPORTANT

(*feedback loops, data drift*)



# SOFTWARE ENGINEERING:

*the world and the machine*

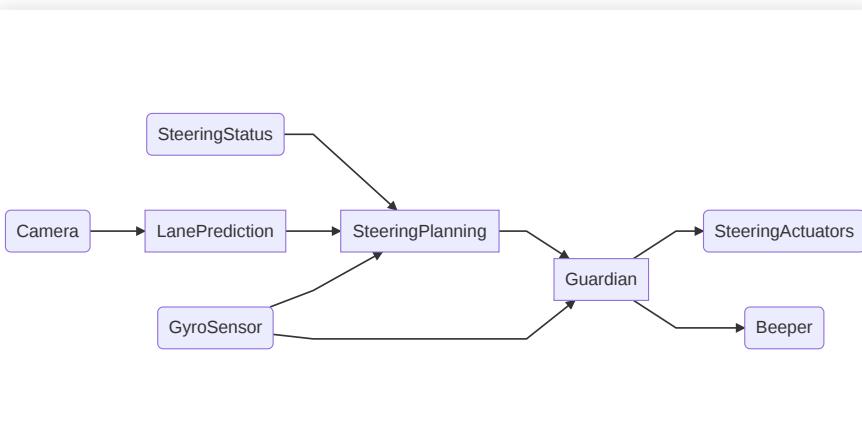


(Jackson ICSE 95)

# SOFTWARE ENGINEERING:

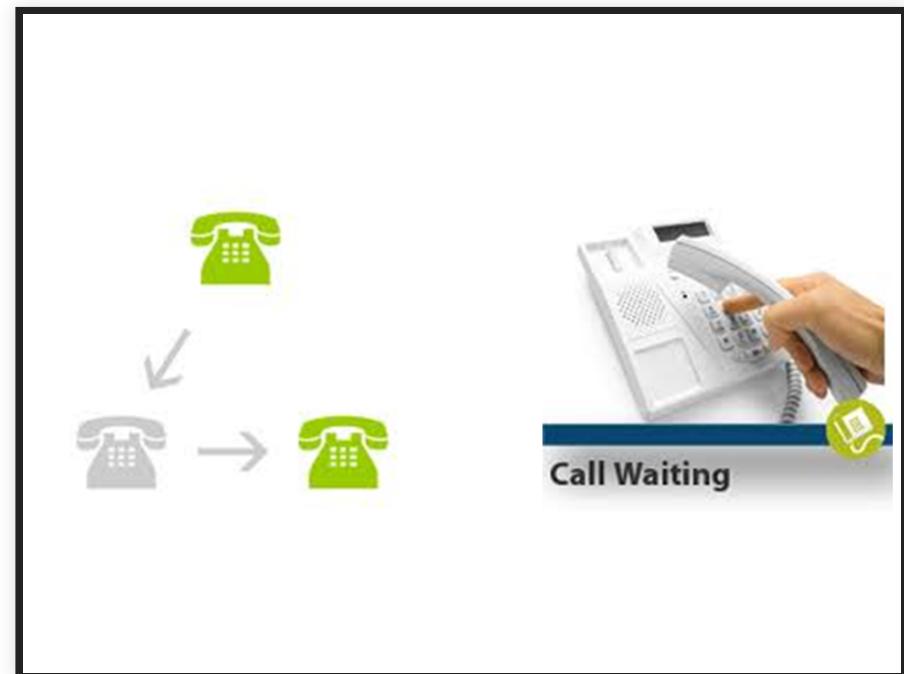
## AI: NONMONOTONIC EFFECTS

*multiple models in most systems*



*feature interactions*

*system testing*

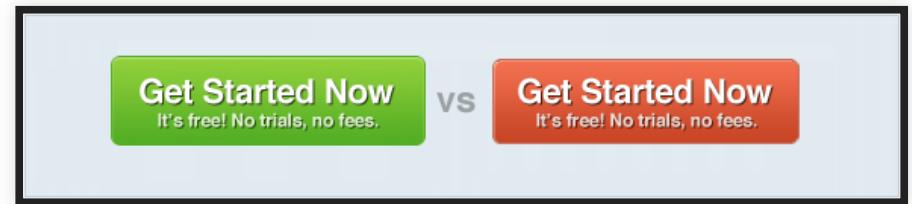


# ML: TESTING IN PRODUCTION



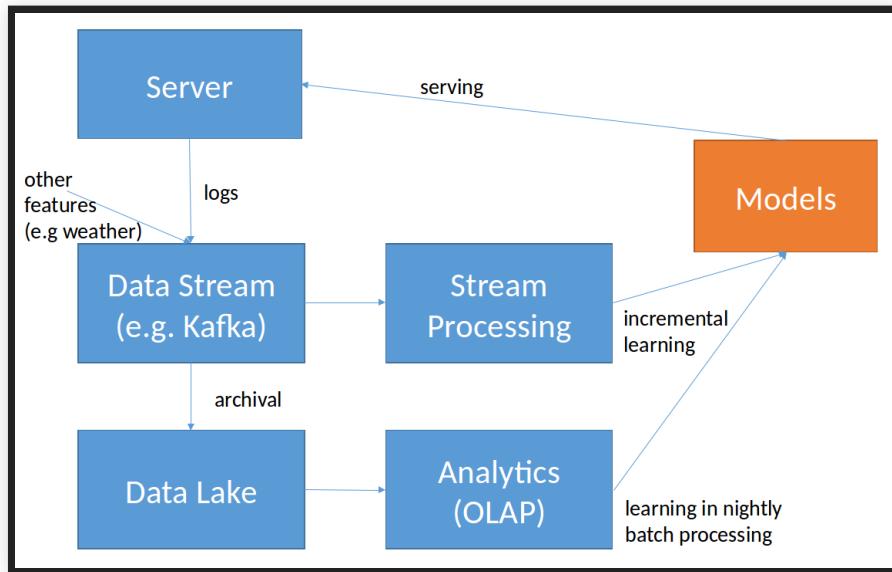
# SOFTWARE ENGINEERING:

Chaos engineering, A/B testing, continuous deployment, feature flags, canary releases



# ML: DATA MANAGEMENT, VERSIONING, AND PROVENANCE

## SE/DATABASE COMMUNITIES:



*stream processing*

*event sourcing*

*data modeling*

*data flow models*

*provenance tracking*

# SOFTWARE ENGINEERS IN AI-ENABLED SYSTEM PROJECTS

- Missing specifications -- *implicit, vague specs very common; safe systems from unreliable components*
- Environment is important -- *the world vs the machine*
- Nonlocal and nonmonotonic effects -- *feature interactions, system testing*
- Testing in production -- *continuous deployment, A/B testing*
- Data management, versioning, and provenance -- *stream processing, event sourcing, data modeling*

# EXAMPLES OF SOFTWARE ENGINEERING CONCERNS

- How to build robust AI pipelines and facilitate regular model updates?
- How to deploy and update models in production?
- How to evaluate data and model quality in production?
- How to deal with mistakes that the model makes and manage associated risk?
- How to trade off between various qualities, including learning cost, inference time, updatability, and interpretability?
- How to design a system that scales to large amounts of data?
- How to version models and data?
- How to manage interdisciplinary teams with data scientists, software engineers, and operators?

# MY VIEW

*While developers of simple traditional systems may get away with poor practices, most developers of ML-enabled systems will not.*

*This is an education problem, more than a research problem.*



**Data  
Scientists**



**Software  
Engineers**

# DATA SCIENTISTS AND SOFTWARE ENGINEERS HAVE DIFFERENT ROLES AND EXPERTISE

*that's okay; that's good*

# POOR SOFTWARE ENGINEERING PRACTICES IN NOTEBOOKS?

```
[ ] # learning a classifier whether the result will be nonZero
from sklearn import tree

classifier=tree.DecisionTreeClassifier(max_depth=8)
classifier=classifier.fit(Xtrain,ynztrain)

print(classifier.score(Xtrain, ynztrain))
print(classifier.score(Xtest, ynztest))

[ ] 0.8266666666666667
0.7295238095238096

[ ] # learning a regression model only on the nonZero data (test is on all data and somewhat
from sklearn import tree

predictor=tree.DecisionTreeRegressor(max_depth=8)
predictor=predictor.fit(XnzTrain, YnzTrain)

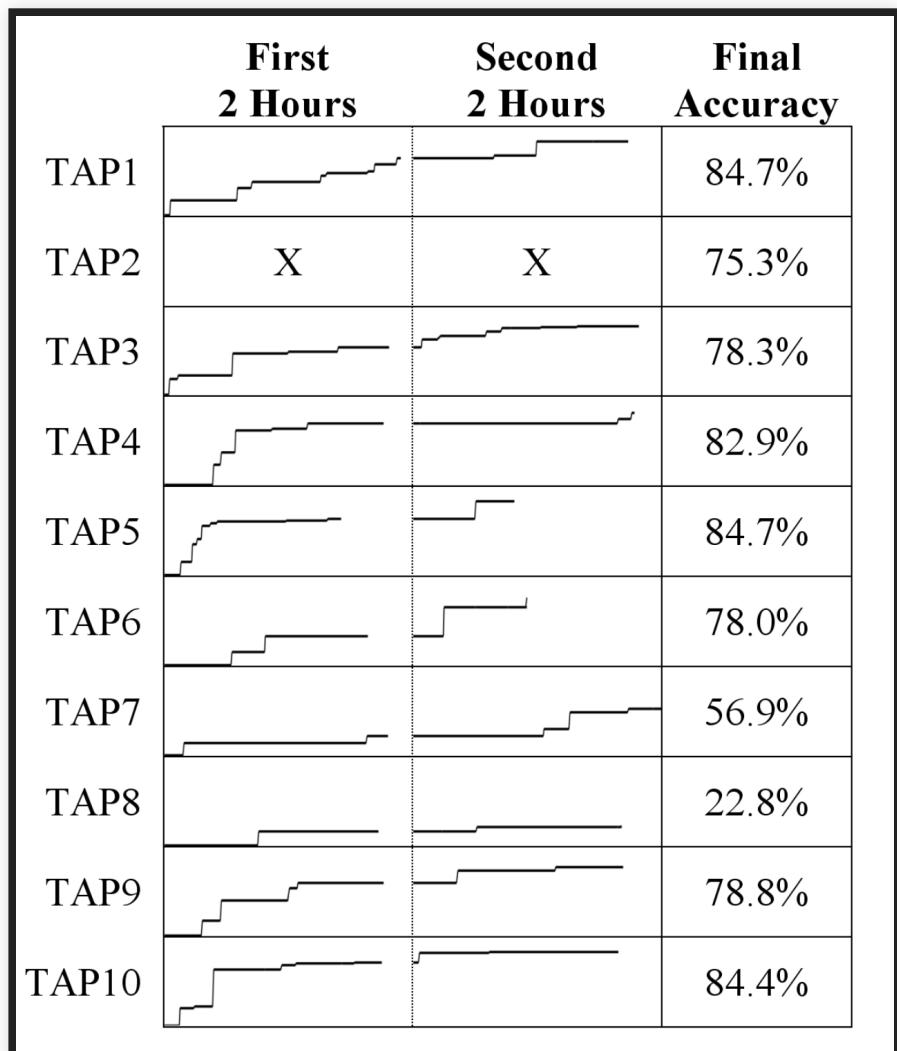
print(predictor.score(XnzTrain, YnzTrain))
print(predictor.score(Xtest, ytest))

[ ] 0.9376379365613154
-2.437397740412892
```

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

# UNDERSTANDING BENEFITS OF NOTEBOOKS

- Data science is exploratory and iterative
  - Science mindset
  - No clear specs, unclear whether possible
- Notebooks offer
  - Quick feedback, similar to REPL
  - Visual feedback
  - Incremental computation
  - Easy to edit and share



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.



# UNDERSTANDING DATA SCIENTIST WORKFLOWS

- Instead of blindly recommended "SE Best Practices" understand context
- Documentation and testing not a priority in exploratory phase
- Help with transitioning into practice
  - From notebooks to pipelines
  - Support maintenance and iteration once deployed
  - Provide infrastructure and tools

# THE SOFTWARE ENGINEERING MINDSET

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?

# 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 --*

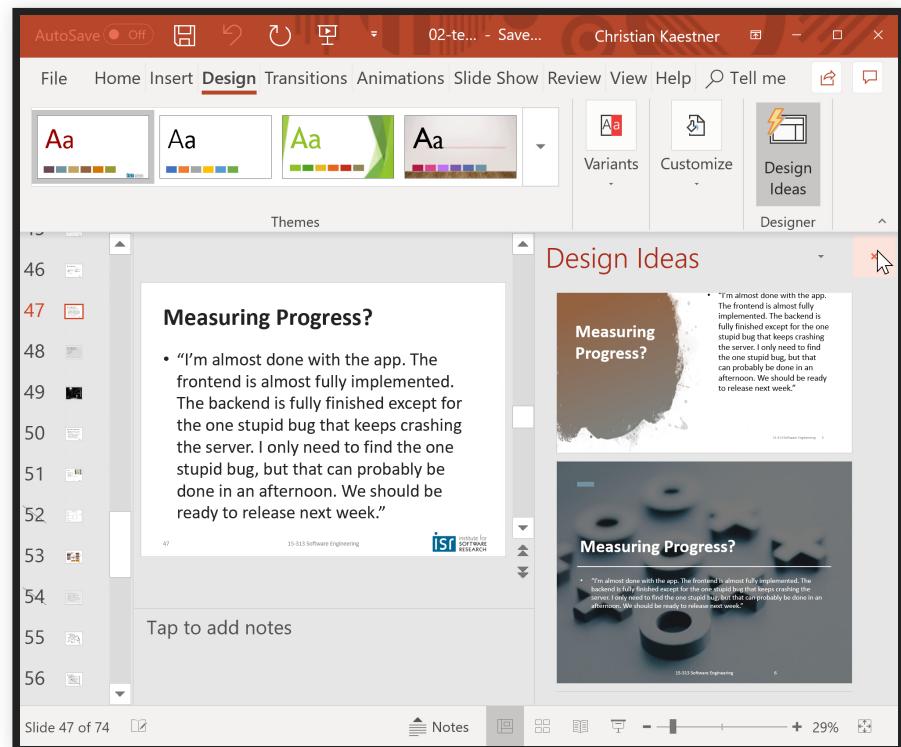
*Donella Meadows*

# FOUR DESIGN CHALLENGES

- User interaction design
- Model qualities and deployment tradeoffs
- Risk analysis and safety
- Telemetry design

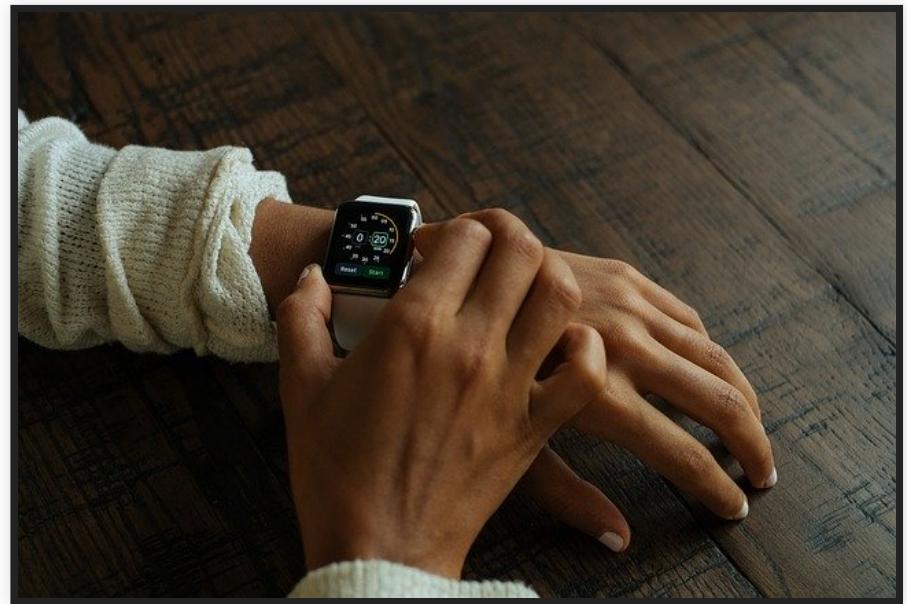
# DESIGNING INTELLIGENT EXPERIENCES

- How to present prediction to a user?
- 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?



# DESIGNING INTELLIGENT EXPERIENCES

- How to present prediction to a user?
- 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?



(fall detection)

# FOUR DESIGN CHALLENGES

- User interaction design
- **Model qualities and deployment tradeoffs**
- Risk analysis and safety
- Telemetry design

# BEYOND ACCURACY

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

# 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)



# ARCHITECTURE: WHERE SHOULD THE MODEL LIVE?

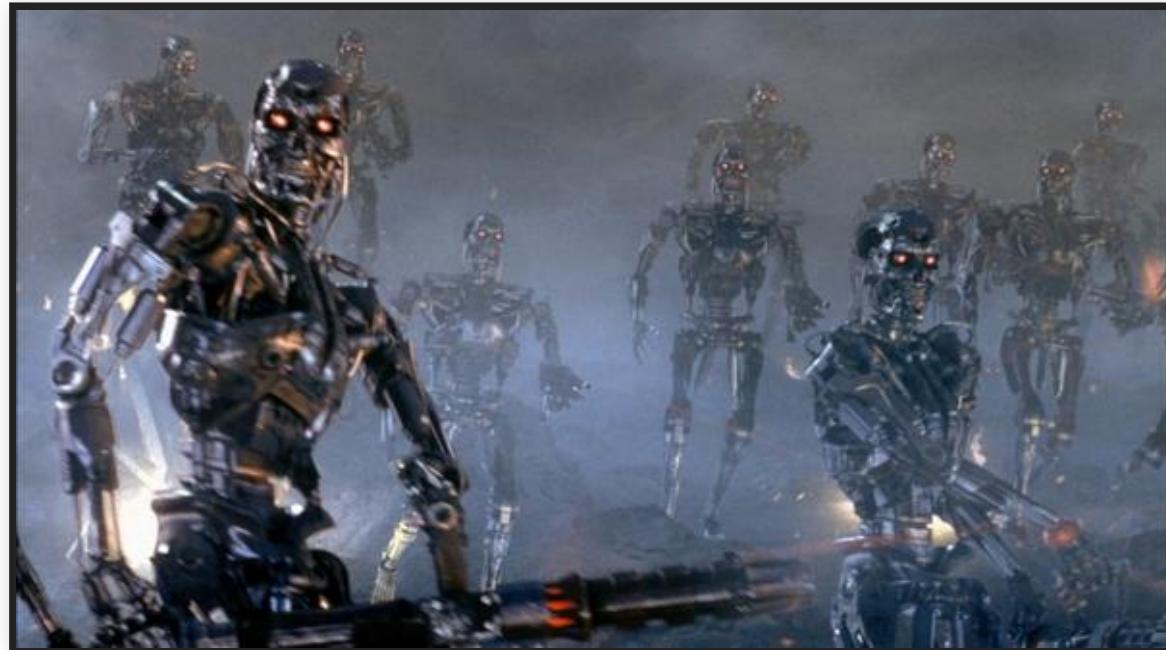


# FOUR DESIGN CHALLENGES

- User interaction design
- Model qualities and deployment tradeoffs
- **Risk analysis and safety**
- Telemetry design

# RISK ANALYSIS

*What's the worst that could happen?*



# RISK ANALYSIS

- What can possibly go wrong in my system, and what are potential impacts on system requirements?
- Risk = Likelihood \* Impact
- Many established methods:
  - Failure mode & effects analysis (FMEA)
  - Hazard analysis
  - Why-because analysis
  - Fault tree analysis (FTA)
  - Hazard and Operability Study (HAZOP)
  - ...

# THE SMART TOASTER

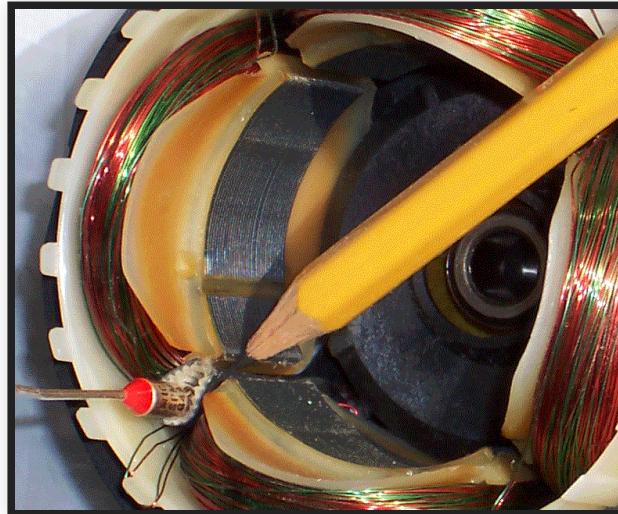
continueToasting(camera<sub>initial</sub>, camera<sub>now</sub>, temperatureReading, userPref) → Boolean



*the toaster may (occasionally) burn my toast, but should never burn down my kitchen*

# SAFEGUARDS / GUARDRAILS

- Hard constraints overrule model
  - `heat = (temperatureReading < MAX) && continueToasting( . . . )`
- External hardware or software failsafe mechanisms
  - outside the model, external observer, e.g., thermal fuses
- Human in the loop



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

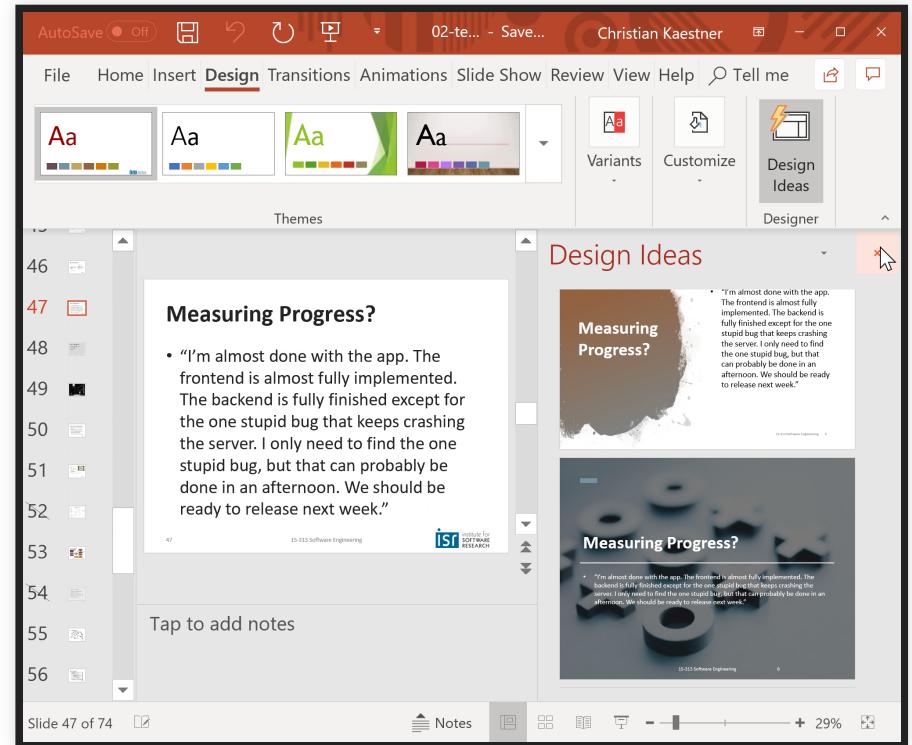
# FOUR DESIGN CHALLENGES

- User interaction design
- Model qualities and deployment tradeoffs
- Risk analysis and safety
- Telemetry design

# KEY DESIGN CHALLENGE: TELEMETRY

- How can we identify mistakes?  
Both false positives and false negatives?
- How can we collect feedback without being intrusive (e.g., asking users about every interactions)?
- How much data are we collecting?  
Can we manage telemetry at scale?  
How to sample properly?
- How do we isolate telemetry for specific AI components and versions?

Automatic slide design:



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!

## Speaker notes

Expect only sparse feedback and expect negative feedback over-proportionally

# MANUALLY LABEL PRODUCTION SAMPLES



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 text "Prices may fall within 7 days – Watch". Inside the callout, it says "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." To the left of the callout, there's a section titled "Stops" with three checkboxes: "nonstop" (checked), "1 stop" (checked), and "2+ stops" (unchecked). Below that is a section titled "Times" with a "Create a price alert" button. At the bottom, there are dropdown menus for "Take-off Dallas" and "Arrival San Francisco".

Advice: **Watch** Learn more ⓘ

DFW ↔ SFO Nov 16

1659 of 1687 flights Wednesday

Create a price alert

**Stops**

nonstop

1 stop

2+ stops

**Times**

Take-off Dallas

Arrival San Francisco

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.

Create a price alert

## Speaker notes

Can just wait 7 days to see actual outcome for all predictions

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' (with a play icon), 'Back 5s' (with a circular arrow icon), '1x' (selected, with a speedometer icon), and 'Volume' (with a speaker icon). A vertical scroll bar is on the far right.

## 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?

## Speaker notes

Clever UI design allows users to edit transcripts. UI already highlights low-confidence words, can observe changes in editor (UI design encourages use of editor). In addition 5 star rating for telemetry.

# MEASURING MODEL QUALITY WITH TELEMETRY

- Telemetry can provide insights for correctness
  - sometimes very accurate labels for real unseen data
  - sometimes only mistakes
  - sometimes indicates severity of mistakes
  - sometimes delayed
  - often just samples, may be hard to catch rare events
  - often just weak proxies for correctness
- Often sufficient to approximate precision/recall or other measures
- Mismatch to (static) evaluation set may indicate stale or unrepresentative test data
- Trend analysis can provide insights even for inaccurate proxy measures

# 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



# SOFTWARE ENGINEERS BUILD SYSTEMS

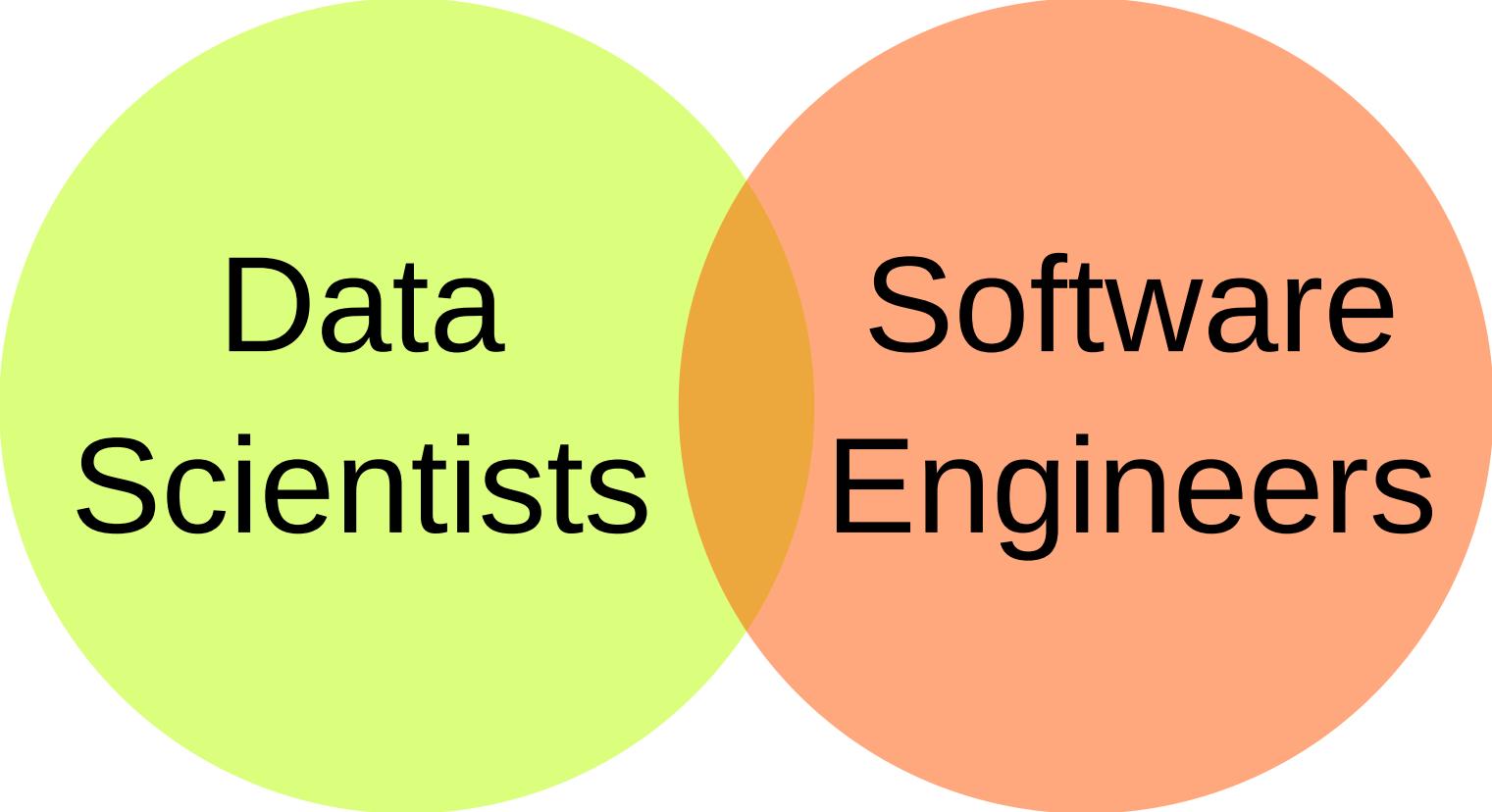
- ML model one component of a larger system
- Understanding the system goals, business case, and success measures
- Requirements and risk analysis
- Architecture, deployment, telemetry
- Automating pipelines at scale
- Continuous delivery and testing in production
- ...



**Data  
Scientists**



**Software  
Engineers**

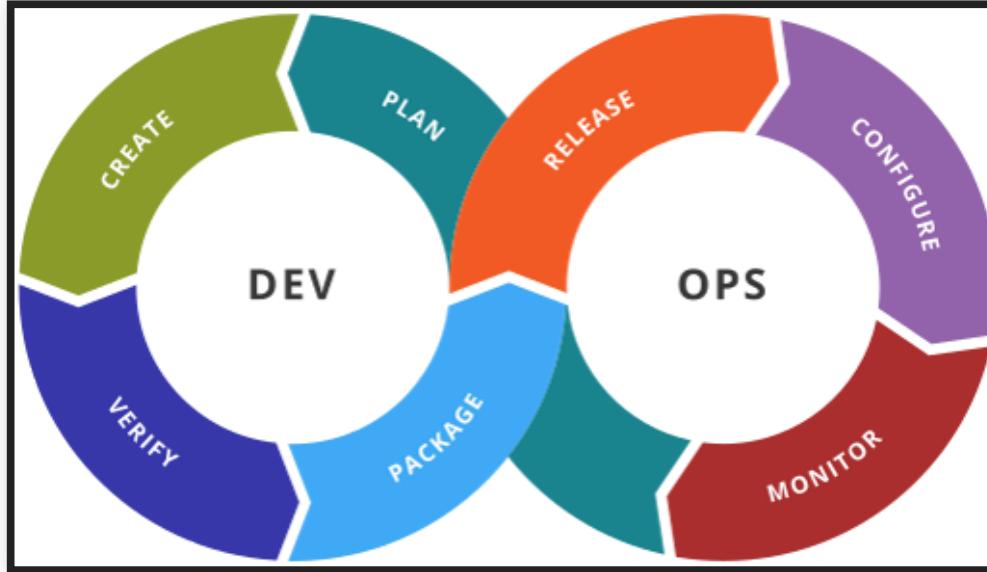


A Venn diagram consisting of two overlapping circles. The left circle is light green and contains the text "Data Scientists". The right circle is light orange and contains the text "Software Engineers". The two circles overlap in the center, representing the intersection of the two fields.

Data  
Scientists

Software  
Engineers

# LET'S LEARN FROM DEVOPS



Distinct roles and expertise, but joint responsibilities, joint tooling

# TOWARD BETTER ML-SYSTEMS ENGINEERING

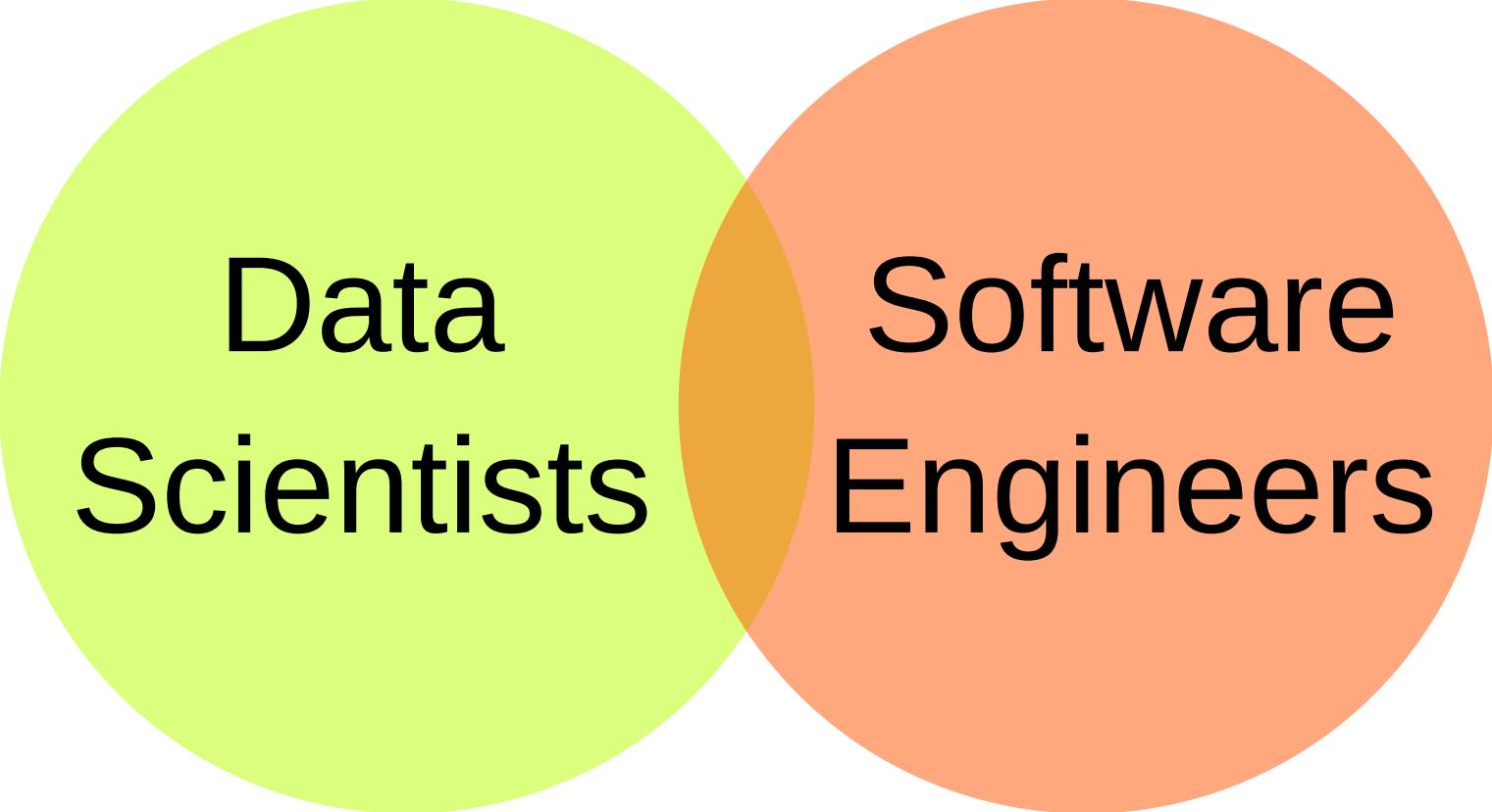
Interdisciplinary teams, split expertise, but joint responsibilities

- Joint vocabulary and tools

- Foster system thinking

- Awareness of production quality concerns

- Understand goals and requirements, perform risk analysis



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

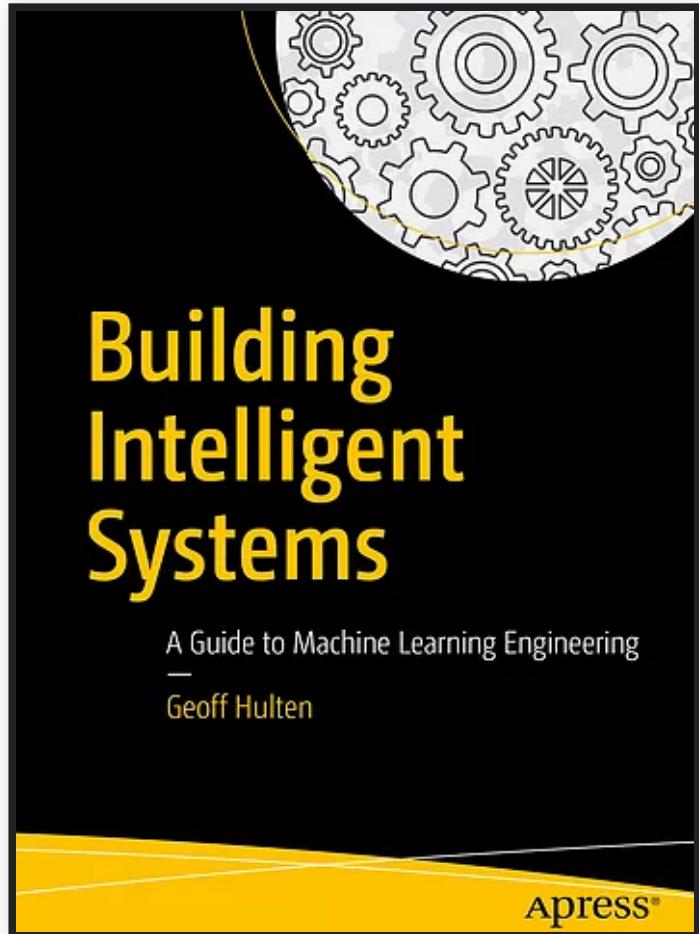
Software  
Engineers

## FURTHER READINGS

All lecture material for "Software Engineering for AI-Enabled Systems":

<https://ckaestne.github.io/seai/>

Annotated bibliography:  
<https://github.com/ckaestne/seaibib>



# SUMMARY: SOFTWARE ENGINEERING FOR ML-ENABLED SYSTEMS

- Building, operating, and maintaining systems with ML component
- Data scientists and software engineers have different expertise, both needed
- Many engineering concerns at the system level:
  - User interaction design
  - Model qualities and deployment tradeoffs
  - Risk analysis and safety
  - Telemetry design
- Interdisciplinary teams, joint vocabulary, and awareness

[kaestner@cs.cmu.edu](mailto:kaestner@cs.cmu.edu) -- @p0nk -- <https://ckaestne.github.io/seai/>

