

SOFTWARE ENGINEERING FOR ML-ENABLED SYSTEMS

Christian Kästner

Carnegie Mellon University

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

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

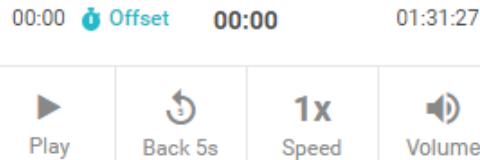
SE FOR ML-ENABLED SYSTEMS != CODING ML FRAMEWORKS



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 (AI-ML-BASED, ML-INFUSED) SYSTEMS



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?

SE FOR ML-ENABLED (AI-ML-BASED, ML-INFUSED) SYSTEMS

AutoSave Off 02-te... - Save... Christian Kaestner

File Home Insert **Design** Transitions Animations Slide Show Review View Help Tell me

Themes

Variants Customize Design Ideas

Design Ideas

Measuring Progress?

- "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."

Measuring Progress?

15-313 Software Engineering 5

46

47 Measuring Progress?

48

49

50

51

52

53

47 15-313 Software Engineering 5

15-313 Software Engineering

isi Institute for SOFTWARE RESEARCH

54



55



56



Slide 47 of 74

Tap to add notes

Notes



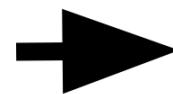
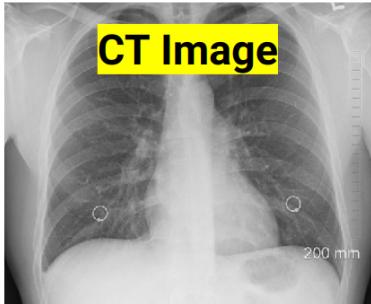
15-313 Software Engineering

6

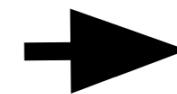
+ 29%



SE FOR ML-ENABLED (AI-ML-BASED, ML-INFUSED) SYSTEMS



Model
(Algorithm)



Cancer?

no cancer

Tryton - Administrator - GNU SOLIDARIO HOSPITAL [Euro]

File User Options Favorites Help

screen

- Addresses
- Categories
- Product
- Financial
- Currency
- Inventory & Stock
- Purchase
- Calendar
- Health
- Patients
- Institutions
- Appointments
- Prescriptions
- Demographics
- Laboratory
- Imaging
- Hospitalizations
- Surgeries
- Pediatrics
- Archives
- Nursing
- Health Services
- Reporting
- Configuration

Patients Obstetric Hist ...

Patients

New Save Switch Reload Previous Next Attachment(0) Action Relate Report E-Mail Print

Main Info

Betz, Ana Female Age: 29y 3m 20d

Critical Information

Personal history of allergy to penicillin
Insulin-dependent diabetes mellitus

Severe allergic reactions to β-lactams



General Info Socioeconomics Medication Diseases Surgeries Genetics Lifestyle QB/GYN

General Screening

Fertile: Pregnant: Menarche age: 12 Menopausal: Menopause age:

OB summary

Pregnancies: 1 Premature: 0 Abortions: 0 Stillbirths: 0

Menstrual History

Date	LMP	Length	frequency	volume	Regular	Dysmenorrhea	Reviewed	Institution
01/24/2015	01/20/2015		5 eumenorrhea	normal	<input type="checkbox"/>	<input type="checkbox"/>	Cordara, Cameron	GNU SOLIDARIO HOSPITAL

tryton://health.gnusolidario.org:8000/health28rc1/model/gnuhealth.patient/1;views=%5B223%2C+224%5D



Data
Scientists



Software
Engineers

and domain experts + lawyers + operators + security experts + regulators + ...

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
- 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

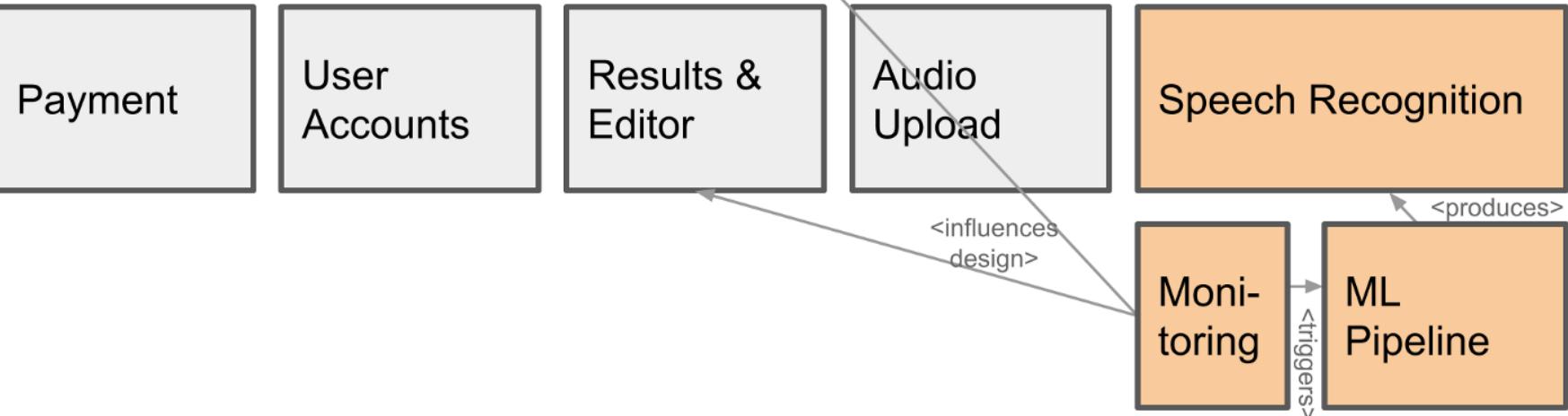
Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed 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?

User Interface



Database, Hadoop, Kafka

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

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) {  
    ...  
}
```

MISSING SPECIFICATIONS

from deductive to inductive reasoning, from specs to examples

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

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

```
/**  
 *  
 *  
 */  
Boolean hasCancer(byte[][] image);
```

All models are approximations. Assumptions, whether implied or clearly stated, are never exactly true. All models are wrong, but some models are useful. So the question you need to ask is not "Is the model true?" (it never is) but "Is the model good enough for this particular application?"

-- George Box

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

NON-ML EXAMPLE: NEWTON'S LAWS OF MOTION

2nd law: "the rate of change of momentum of a body over time is directly proportional to the force applied, and

occurs in the same direction as the applied force" $\mathbf{F} = \frac{dp}{dt}$

"Newton's laws were verified by experiment and observation for over 200 years, and they are excellent approximations at the scales and speeds of everyday life."

Do not generalize for very small scales, very high speeds, or in very strong gravitational fields. Do not explain semiconductor, GPS errors, superconductivity, ... Those require general relativity and quantum field theory.

Further readings: https://en.wikipedia.org/wiki/Newton%27s_laws_of_motion

"Since all models are wrong the scientist must be alert to what is importantly wrong. It is inappropriate to be concerned about mice when there are tigers abroad." --
George Box, 1976

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

ENVIRONMENT IS IMPORTANT

(feedback loops, data drift, safety concerns)

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

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 [2:56]

Flat Earth Clues Introduction - Mark Sargent [12:36]

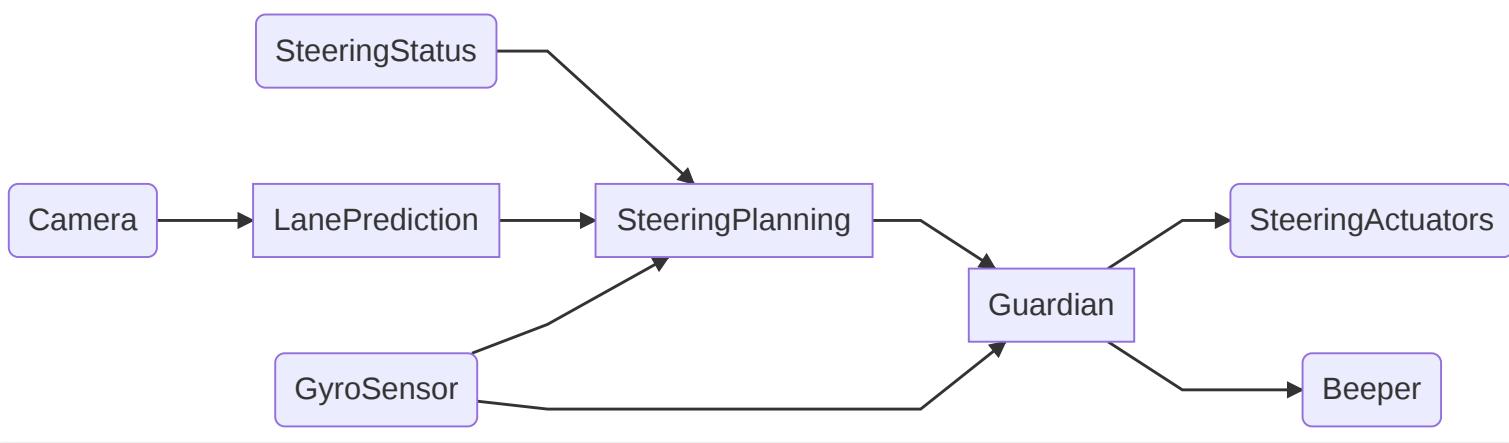
FLAT EARTH Clues Part 1 - Empty Theatre - Mark Sargent [7:20]

FLAT EARTH Clues Part 2 - Byrd Wall - Mark Sargent [14:50]

FLAT EARTH Clues Part 3 - Map Makers - Mark Sargent [6:45]

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

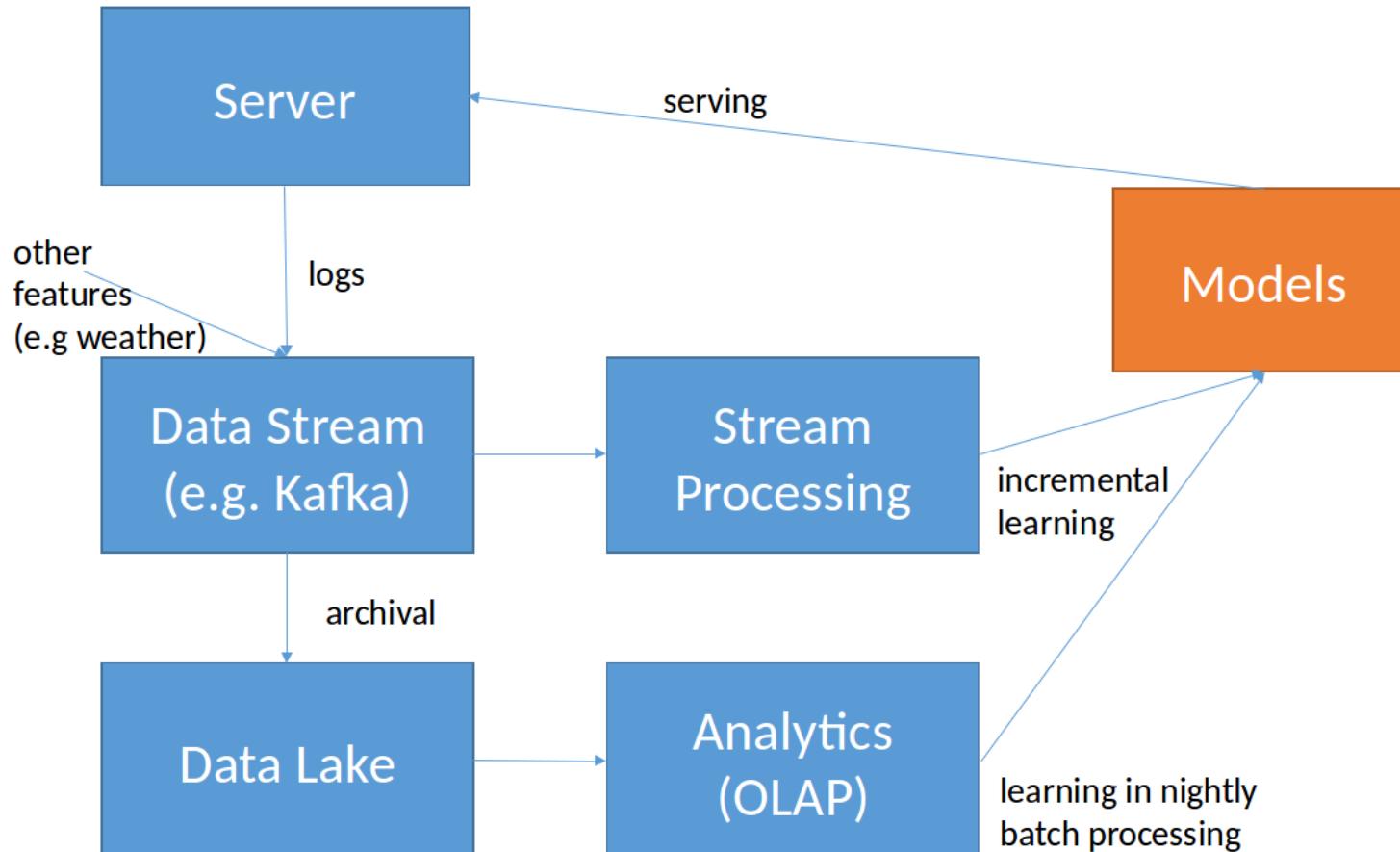
@ddowza not really sorry

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,  
                         ...);
```

```
/**  
 *  
 *  
 */  
Boolean hasCancer(byte[][][] image);
```

SOFTWARE ENGINEERING:

vague specs very common

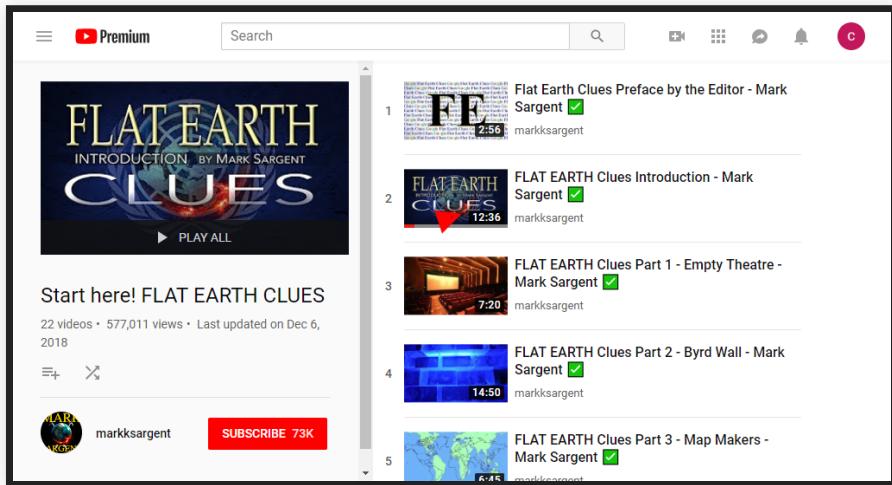
agile methods

safe systems from
unreliable components

("ML is requirements
engineering")

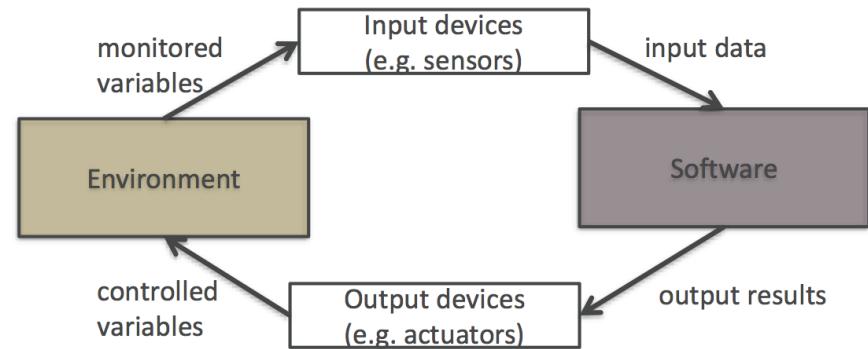
ML: ENVIRONMENT IS IMPORTANT

(feedback loops, data drift)



SOFTWARE ENGINEERING:

the world and the machine

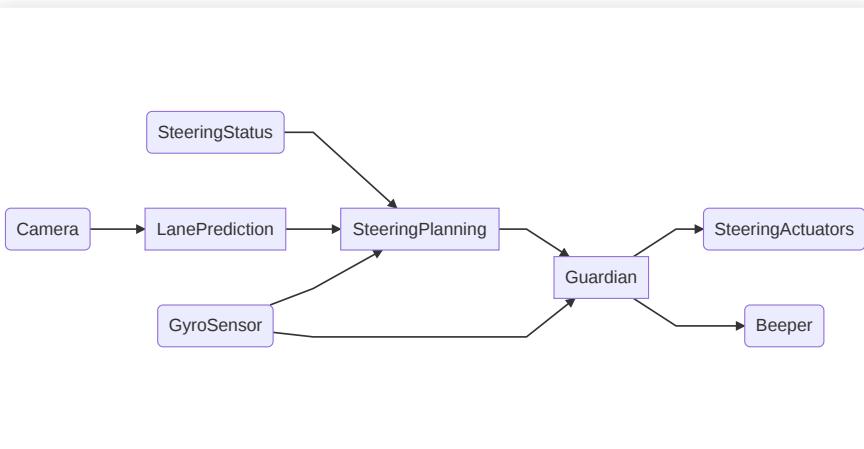


(Jackson ICSE 95)

SOFTWARE ENGINEERING:

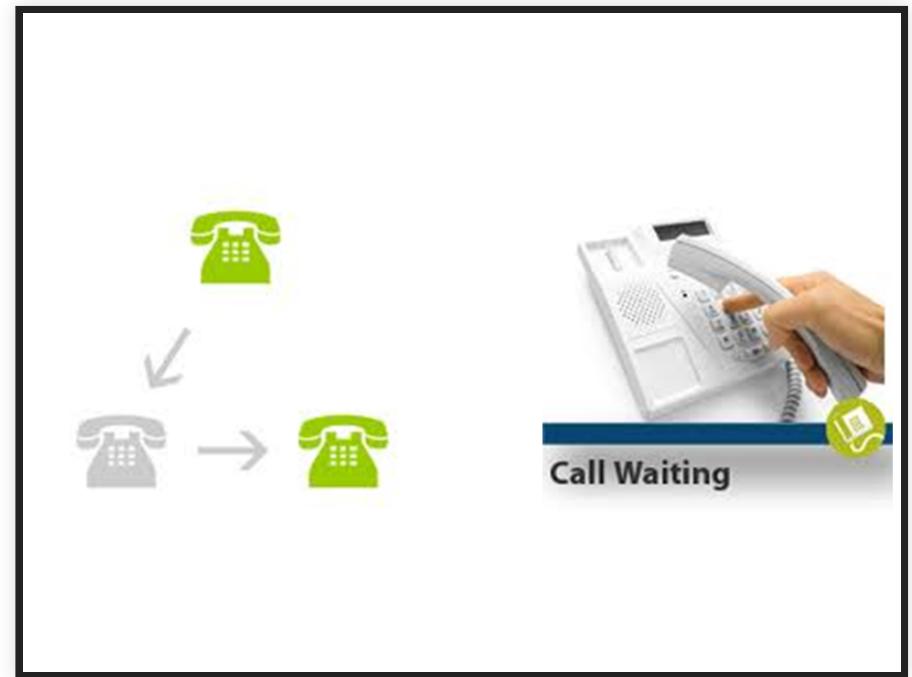
ML: NONMONOTONIC EFFECTS

multiple models in most systems



feature interactions

system testing



ML: TESTING IN PRODUCTION



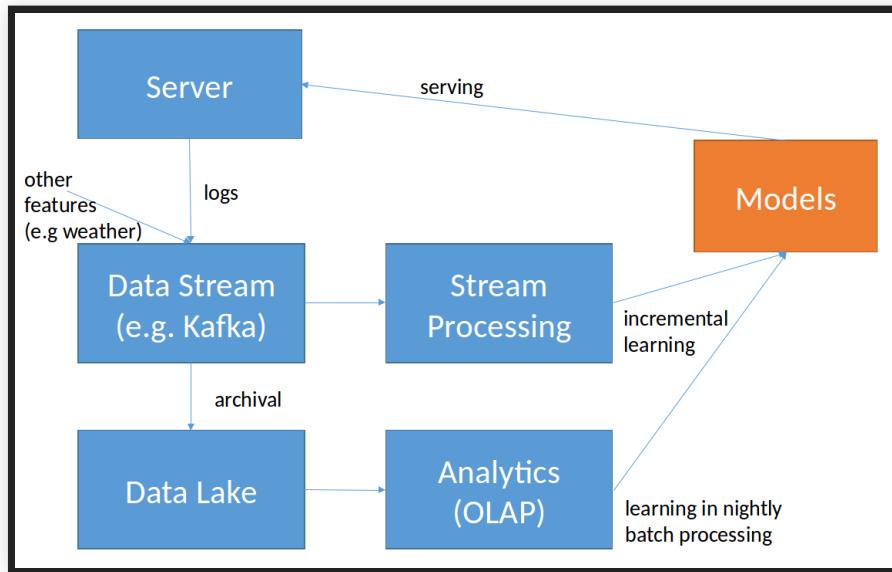
SOFTWARE ENGINEERING:

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*

MY VIEW

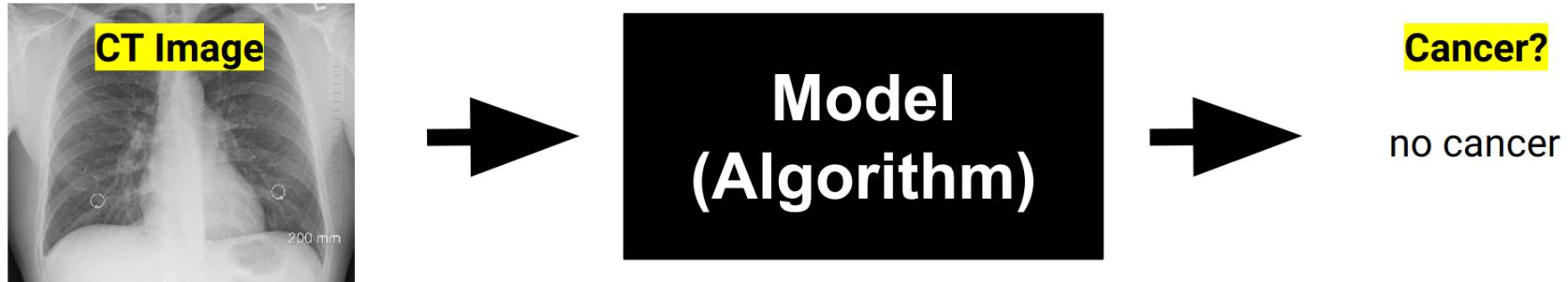
While developers of simple traditional systems may get away with poor practices, most developers of ML-enabled systems will not.

QUALITY ASSURANCE FOR ML-ENABLED SYSTEMS

*Illustrating software engineering and systems concerns by
diving into one problem*

TRADITIONAL FOCUS: MODEL ACCURACY

- Train and evaluate model on fixed labeled data set
- Compare prediction with labels



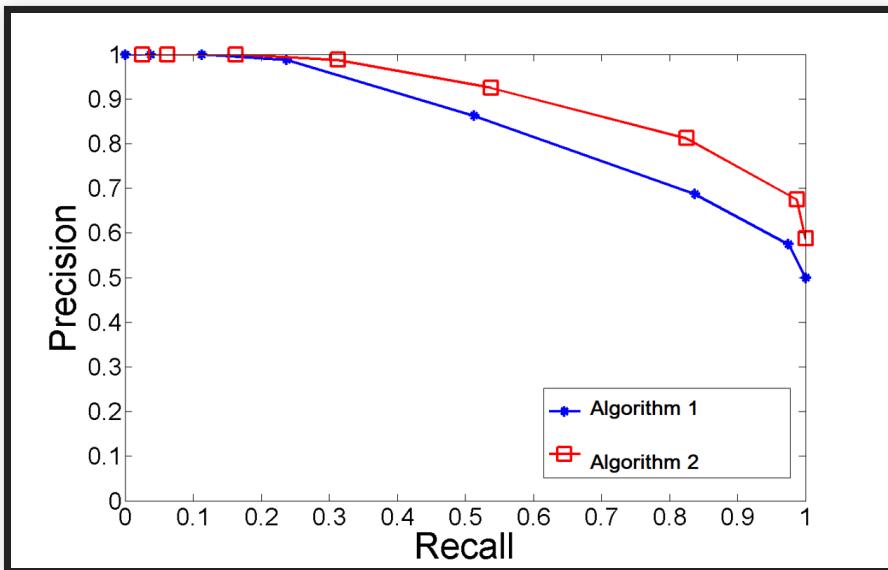
TRADITIONAL FOCUS: MODEL ACCURACY

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

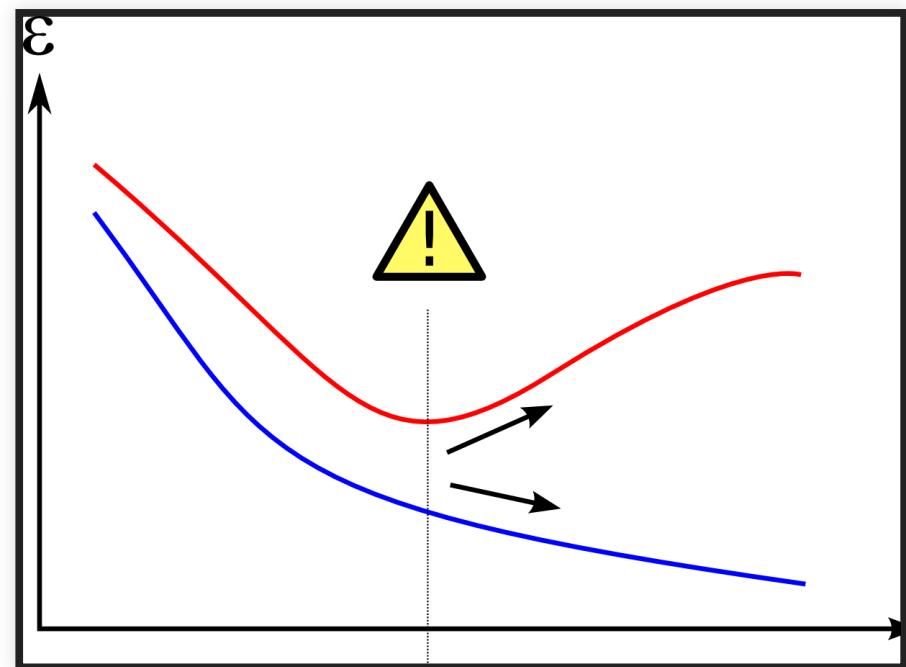
Accuary, Recall, Precision, F1-Score

MORE TRADITIONAL MODEL QUALITY DISCUSSIONS

Many model quality metrics (recall,
MAPE, ROC, log loss, top-k, ...)



Generalization/overfitting (train/test/eval split, crossvalidation)



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

NOT ALL MISTAKES ARE EQUAL

- False positives vs false negatives (e.g., cancer detection)
- Fairness across subpopulations
- Generalization beyond one device and one hospital?
- Learn from black-box testing:
 - Equivalence classes
 - Boundary conditions
 - Critical test cases ("call mom")
 - Combinatorial testing
 - Fuzzing

AUTOMATING MODEL EVALUATION

- Continuous integration, automated measurement, tracking of results
- Data and model versioning, provenance



← 2017-08-19-06-29-22-855-UTC

[SUMMARY](#)[DEPLOY](#)[RETRAIN](#)[PERFORMANCE](#) [MODEL VIS](#) [FEATURES](#)

Test Data Performance

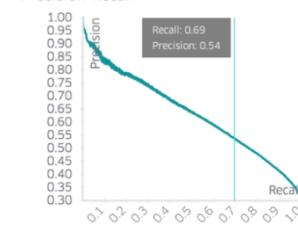


performance

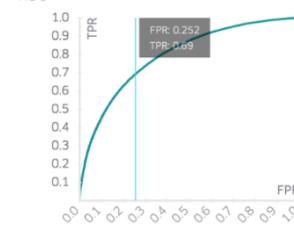
0.7936

auc

Precision-Recall



ROC



Confusion Matrix

Positive label: true

Predicted

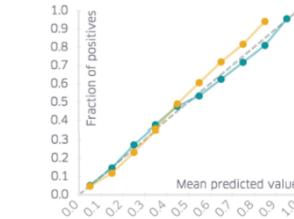
Actual	Predicted	
	YES	NO
YES	TP 0.21 17604 Samples	FN 0.093 7891 Samples
NO	FP 0.18 15005 Samples	TN 0.52 44549 Samples

0.4907

error

calibration

reliability



The reliability diagram shows how reliable (or "well-calibrated") the model's probability estimates are when evaluated on the test data. For example, A well calibrated (binary) model should classify the samples such that among the samples to which it gives a probability close to 0.8 of belonging to the positive class, approximately 80% of those samples actually belong to the positive class. [More Info](#)

- A Perfectly Calibrated Model
- This Model (Before Calibration)
- This Model (After Calibration)

data

BEYOND ACCURACY: QUALITY CONCERNS FOR ML-ENABLED SYSTEMS

- Learning time, cost and scalability
- Update cost, incremental learning
- Inference cost
- Size of models learned
- Amount of training data needed
- Fairness
- Robustness
- Safety, security, privacy
- Explainability, reproducibility
- Time to market
- Overall operating cost (cost per prediction)

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.

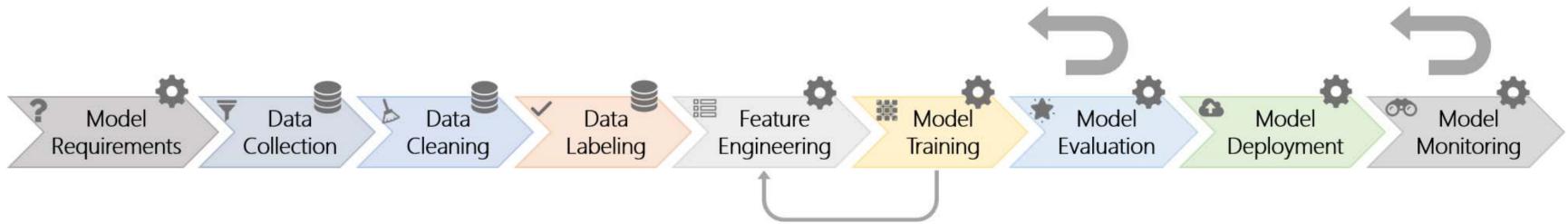
Speaker 5 ► 08:38

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?

INFRASTRUCTURE QUALITY

THINK OF PIPELINES, NOT MODELS, NOT NOTEBOOKS



Many steps: Data collection, data cleaning, labeling, feature engineering, training, evaluation, deployment, monitoring

Automate each step -- test each step

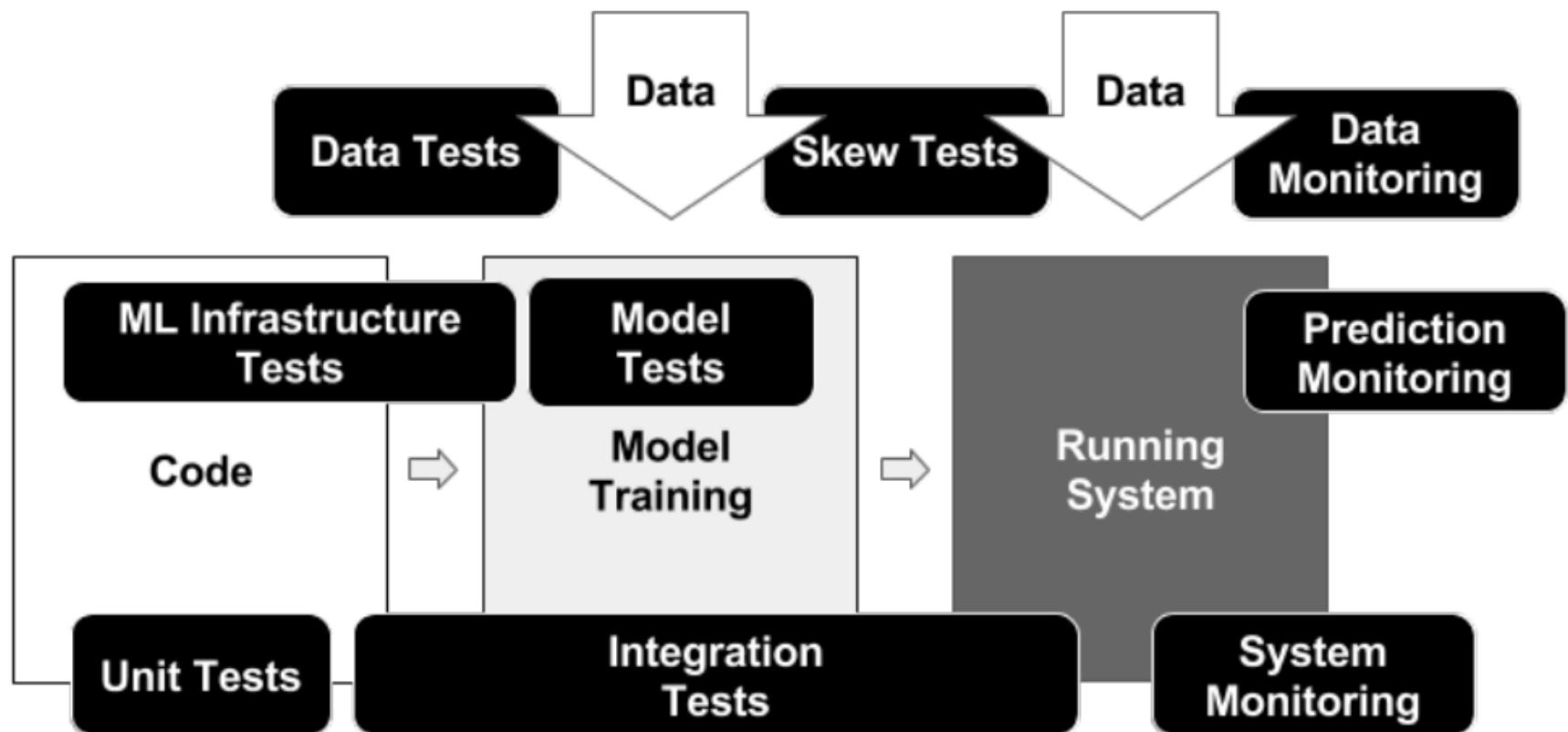
Graphic: Amershi, Saleema, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. "[Software engineering for machine learning: A case study.](#)" In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pp. 291-300. IEEE, 2019.

POSSIBLE MISTAKES IN ML PIPELINES

Danger of "silent" mistakes in many phases:

- Dropped data after format changes
- Failure to push updated model into production
- Incorrect feature extraction
- Use of stale dataset, wrong data source
- Data source no longer available (e.g web API)
- Telemetry server overloaded
- Negative feedback (telemtr.) no longer sent from app
- Use of old model learning code, stale hyperparameter
- Data format changes between ML pipeline steps
- ...

QUALITY ASSURANCE FOR THE ENTIRE PIPELINE



Source: 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)

PIPELINE TESTING

- Unit tests (e.g., data cleaning)
- End to end pipeline tests
- Testing with stubs, test error handling (e.g., test model redeployment after dropped connection)
- Test monitoring infrastructure (e.g., "fire drills")

A screenshot of a transcription software interface. At the top, there's a header with the project 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 OF THE ENTIRE SYSTEM

ML models are "just" one component

LIVING WITH MISTAKES

The smart toaster may occasionally burn my toast, but it should not burn down my kitchen.



Speaker notes

A smart toaster may occasionally burn the toast, but it should never burn down the kitchen. The latter can be achieved without relying on perfect accuracy of a smart component, just stop it when it's overheating.

Plan for mistakes: User interaction, undo, safeguards

MODEL ACCURACY VS SYSTEM GOALS

- System goals are supported by AI components, e.g.,
 - maximizing sales
 - minimizing loss
 - maximizing community growth
 - retaining customers
 - maximizing engagement time
- A better model will support system goals better
 - more accurate
 - faster answers
 - fewer bad mistakes
 - more explainable
 - easier to evolve

A screenshot of a transcription software interface. At the top, there's a header with the project 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?

TESTING IN PRODUCTION

Production data = ultimate unseen data

Focus on system goals, not model accuracy

Monitoring performance over time, canary releases

Finding and debugging common mistakes

Experimentation with A/B tests

HOW ARE WE DOING IN PRODUCTION?

the-changelog-318

[← Dashboard](#) | Quality: High

Last saved a few seconds ago

00:00 00:00 01:31:27

Play Back 5s **1x** Speed 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?

HOW ARE WE DOING IN PRODUCTION?

The screenshot shows a Microsoft Word document window. The title bar includes 'AutoSave Off', a ribbon icon, a refresh icon, a power icon, a dropdown arrow, the file name '02-te... - Save...', the author 'Christian Kaestner', and standard window control buttons. The ribbon tabs are File, Home, Insert, Design (selected), Transitions, Animations, Slide Show, Review, View, Help, Tell me, and two icons.

The main content area displays a slide with the following elements:

- Section Header:** Measuring Progress?
- List 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."
- Page Number:** 47
- Page Footer:** 15-313 Software Engineering
- ISRI Logo:** institute for SOFTWARE RESEARCH

To the right of the slide, a 'Design Ideas' sidebar is open, showing a preview of the slide with a red gradient overlay and the text 'Measuring Progress?'. Below this, there is a list of design ideas:

- "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."

55



Tap to add notes

56



Slide 47 of 74



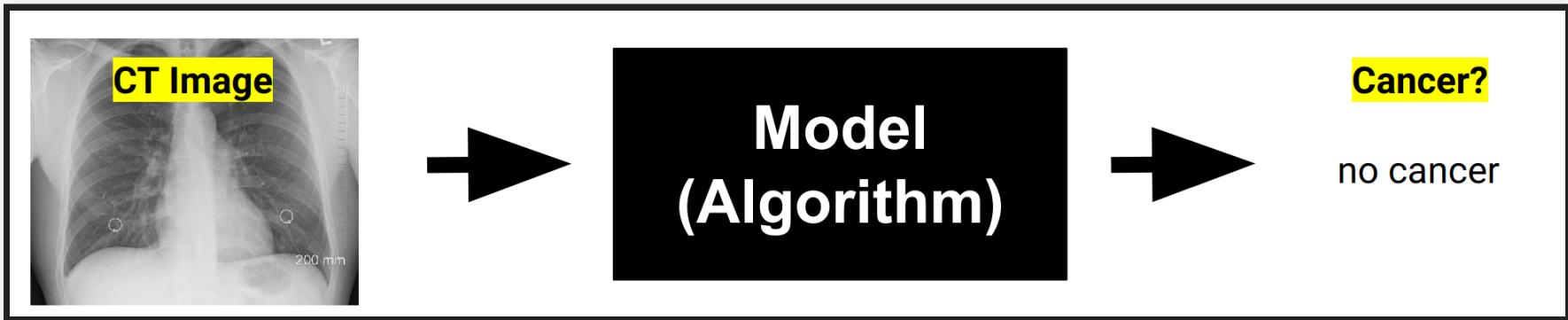
Notes



15-313 Software Engineering

6

HOW ARE WE DOING IN PRODUCTION?



KEY DESIGN CHALLENGE: TELEMETRY

- Identify mistakes in production (“what would have been the right prediction?”)
- Many challenges:
 - 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?

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



The logo for Amazon Mechanical Turk. It features the word "amazon" in its signature black font, with a black curved arrow underneath the "o" and "z". Below this, the words "mechanical turk" are written in a smaller, orange, sans-serif font.

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, a message 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 checked options: "nonstop", "1 stop", and "2+ stops". Below that is a "Times" section 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

Create a price alert

Take-off Dallas

Arrival San Francisco

Speaker notes

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

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?

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

MONITORING MODEL QUALITY IN PRODUCTION

- Watch for jumps after releases
 - roll back after negative jump
- Watch for slow degradation
 - Stale models, data drift, feedback loops, adversaries
- Debug common or important problems
 - Mistakes uniform across populations?
 - Challenging problems -> refine training, add regression tests

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

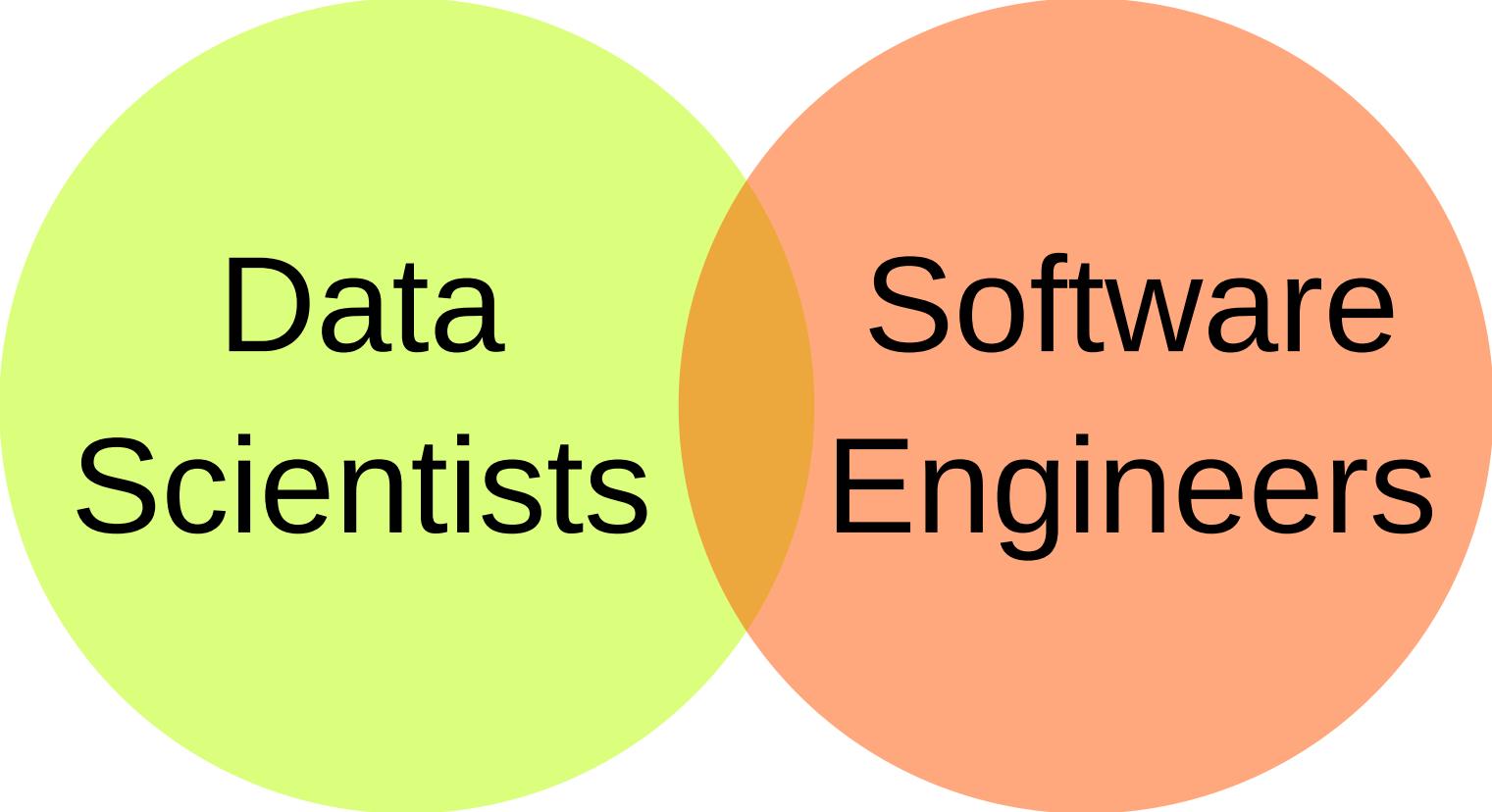




Data
Scientists



Software
Engineers

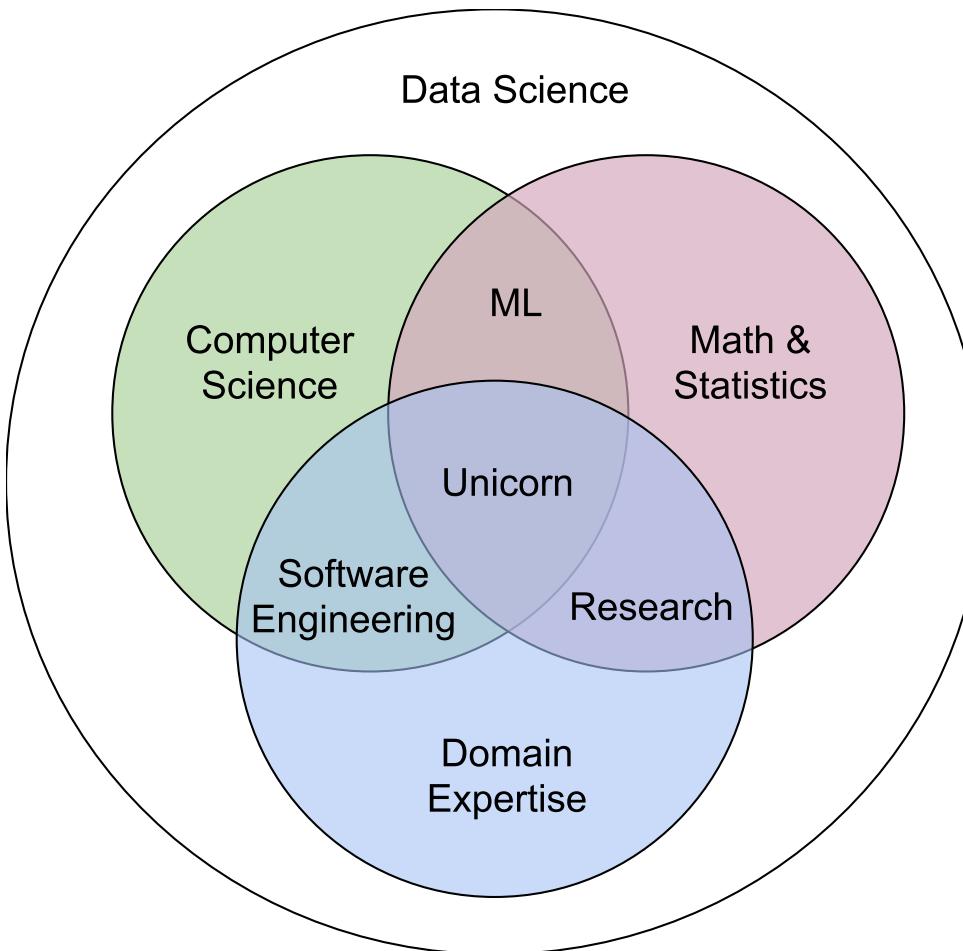


A Venn diagram consisting of two overlapping circles. The left circle is light green and contains the text "Data Scientists". The right circle is light orange and contains the text "Software Engineers". The two circles overlap in the center, representing the intersection of the two fields.

Data
Scientists

Software
Engineers





By Steven Geringer, via Ryan Orban. [Bridging the Gap Between Data Science & Engineer: Building High-Performance Teams](#). 2016

T-SHAPED PEOPLE

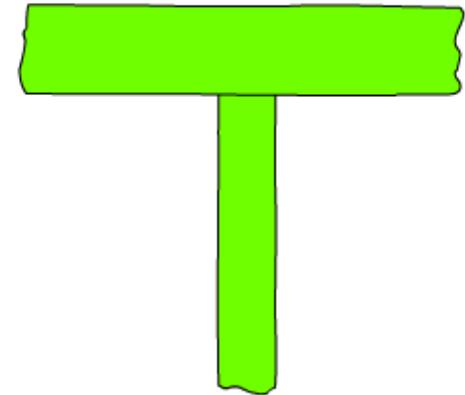
Broad-range generalist + Deep expertise



'I-shaped'
Expert at one thing



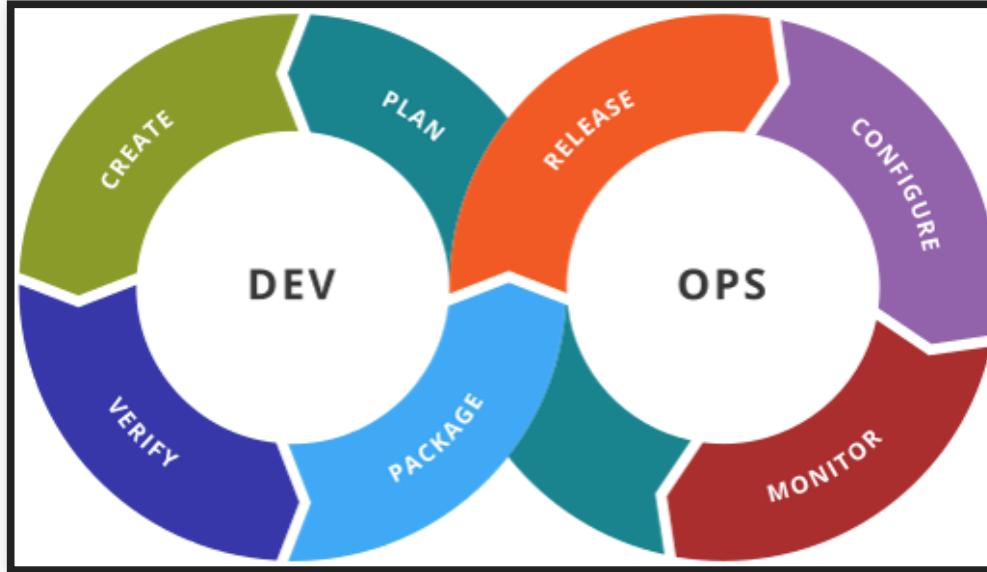
Generalist
Capable in a lot of things
but not expert in any



'T-shaped'
Capable in a lot of things
and expert in one of them

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

LET'S LEARN FROM DEVOPS



Distinct roles and expertise, but joint responsibilities, joint tooling

TOWARD BETTER ML-SYSTEMS ENGINEERING

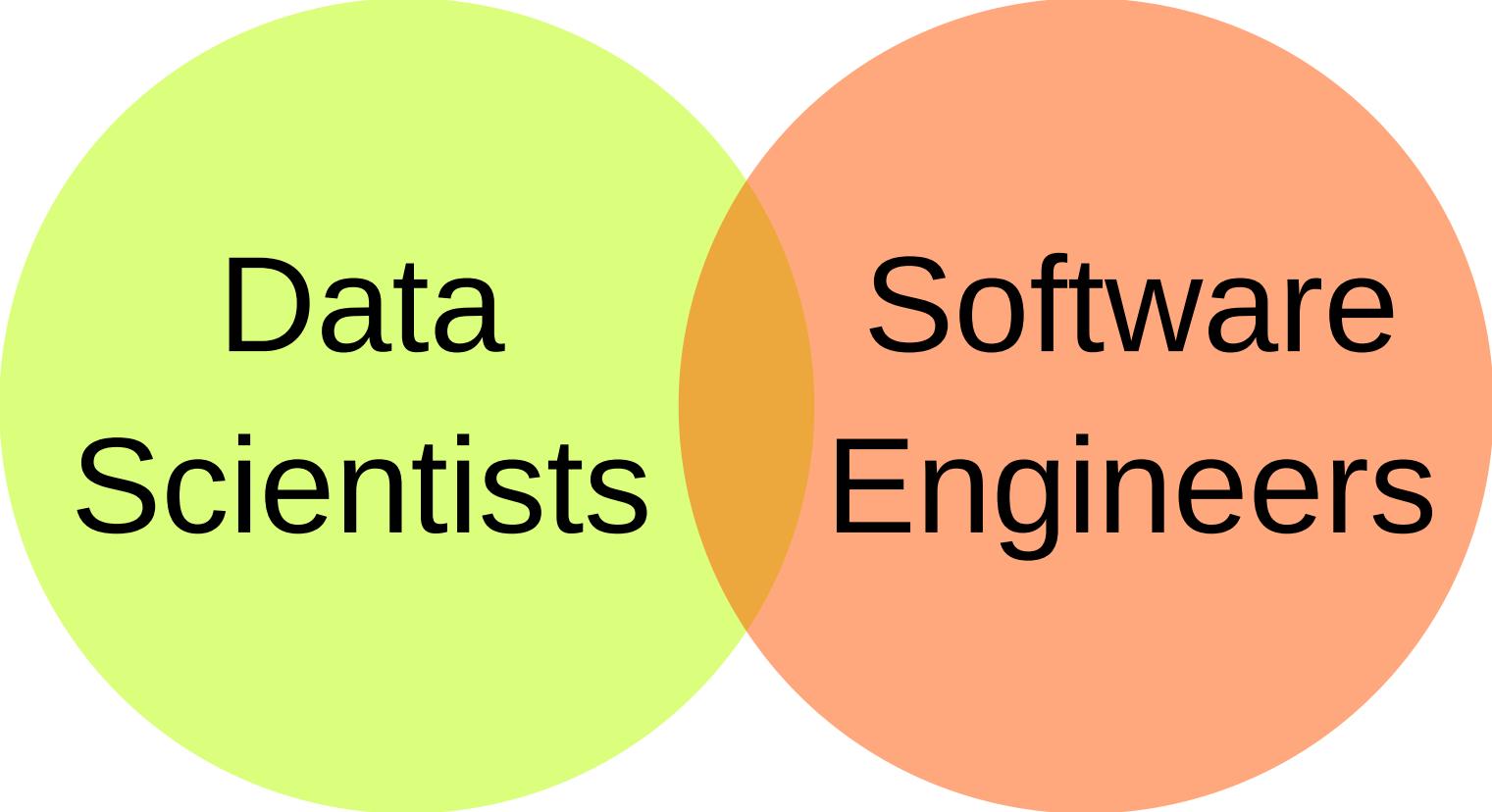
Interdisciplinary teams, split expertise, but joint responsibilities

- Joint vocabulary and tools

- Foster system thinking

- Awareness of production quality concerns

- Perform risk + hazard 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, representing the intersection of the two fields.

Data
Scientists

Software
Engineers

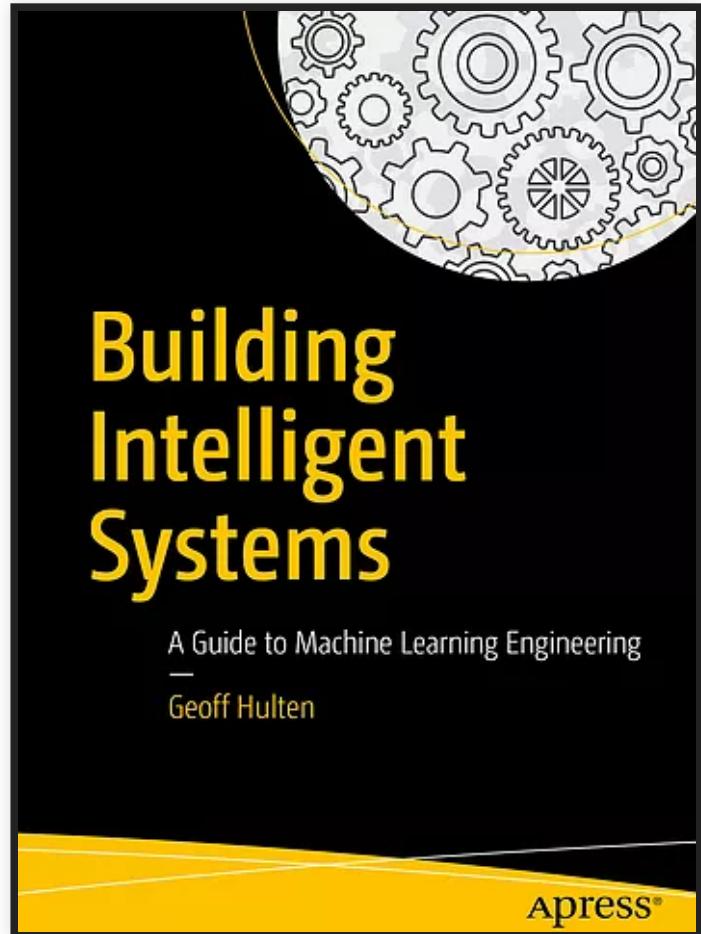
READINGS

All lecture material:

<https://github.com/ckaestne/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
- Quality assurance beyond model accuracy
 - Blackbox testing, test automation
 - Testing the entire ML pipeline
 - Consider whole system
 - Testing in production with telemetry
- Interdisciplinary teams, joint vocabulary, and awareness

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

