Motivation, Syllabus, and Introductions

Machine Learning in Production

Slack

- We use Slack for this course, including during lectures
- See signup link on Canvas (you can do this right now!)
- Setup the ability to read/post to Slack during lecture.
- Sometime in the next day: introduce yourself on #social

Waitlist situation

We *should* be able to accommodate everyone.

Consider expressing ability to move to alternative section time/locations.

Please email the course admins for your number, not us, to inquire about WL status/registration movement.

When you join: post an introduction on Slack (#social).

If necessary, we will post a policy for missed work/extensions, but hopefully things will clear quickly enough that it won't matter.

Learning Goals

- Understand how ML components are parts of larger systems
- Illustrate the challenges in engineering an ML-enabled system beyond accuracy
- Explain the role of specifications and their lack in machine learning and the relationship to deductive and inductive reasoning
- Summarize the respective goals and challenges of software engineers vs data scientists
- Explain the concept and relevance of “T-shaped people”
- Understand the basic mechanics of this course.

Case Study: Music Generation

Context: Music Generation Research

Lam, Max WY, et al. "Efficient neural music generation." Advances in Neural Information Processing Systems 36 (2024).

Recent progress in music generation has been remarkably advanced by the state-of-the-art MusicLM, which comprises a hierarchy of three LMs, respectively, for semantic, coarse acoustic, and fine acoustic modelings. Yet, sampling with the MusicLM requires processing through these LMs one by one to obtain the fine-grained acoustic tokens, making it computationally expensive and prohibitive for a real-time generation. Efficient music generation with a quality on par with MusicLM remains a significant challenge.In this paper, we present MeLoDy (M for music; L for LM; D for diffusion), an LM-guided diffusion model that generates music audios of state-of-the-art quality meanwhile reducing 95.7% to 99.6% forward passes in MusicLM, respectively, for sampling 10s to 30s music. MeLoDy inherits the highest-level LM from MusicLM for semantic modeling, and applies a novel dual-path diffusion (DPD) model and an audio VAE-GAN to efficiently decode the conditioning semantic tokens into waveform. DPD is proposed to simultaneously model the coarse and fine acoustics by incorporating the semantic information into segments of latents effectively via cross-attention at each denoising step. Our experimental results suggest the superiority of MeLoDy, not only in its practical advantages on sampling speed and infinitely continuable generation, but also in its state-of-the-art musicality, audio quality, and text correlation. Our samples are available at https://Efficient-MeLoDy.github.io/.

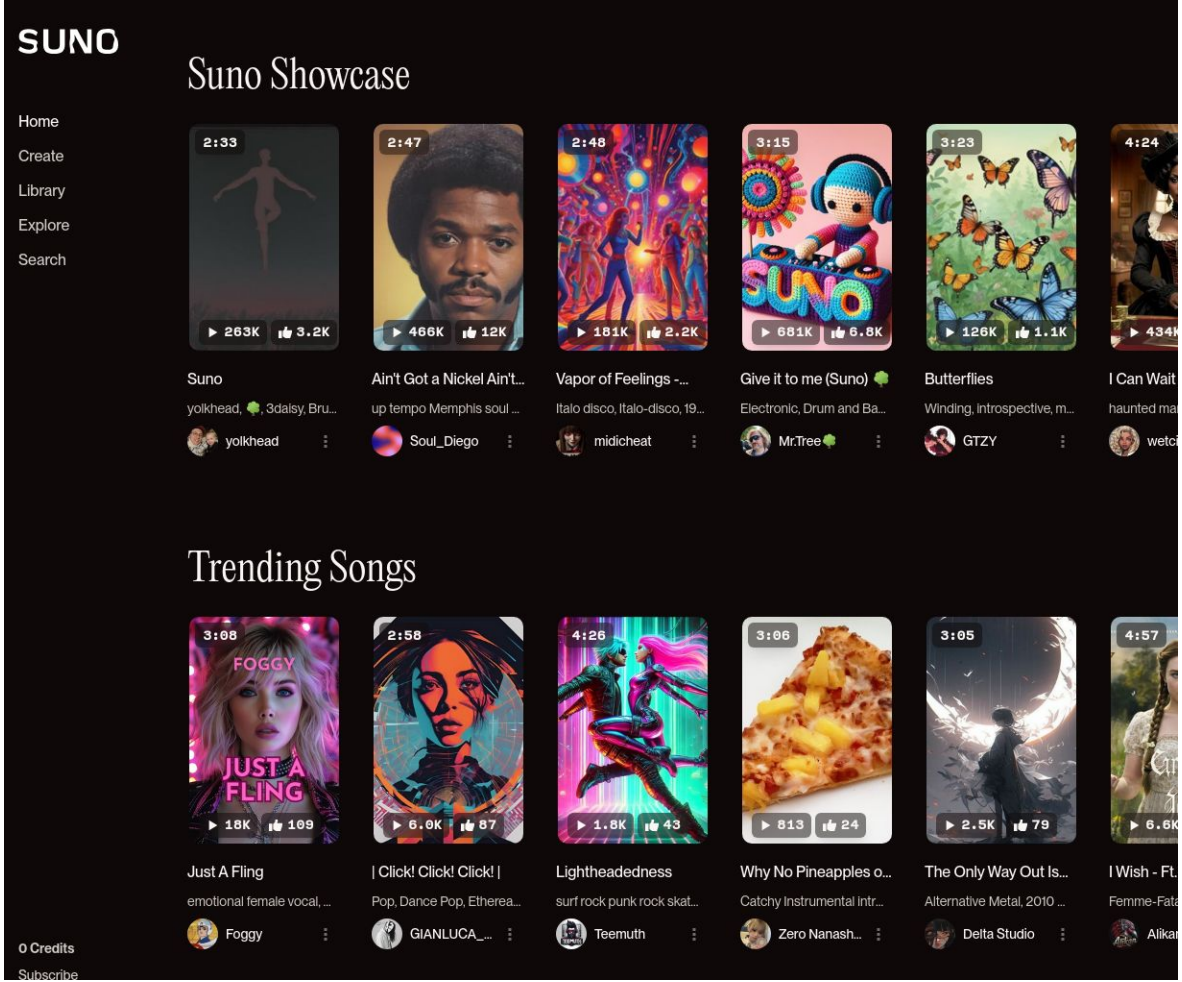
The Startup Idea

You just completed a research thesis as part of your Master's/PhD degree about making deep learning for generative AI more energy efficient

Your recent project was using music generation as a case study. You showed 30% energy improvements with similar quality of generated music on benchmark prompts.

Two friends are excited about the application of music generation.

Idea: Let's commercialize the idea and sell it to end users or music producers.


SUNO
Home
Create
Library
Explore
Search
0 Credits
Subscribe
Suno Showcase
2:33
263K 3.2K
Suno
yolkhead, 🌳, 3dalsy, Bru...
yolkhead
2:47
466K 12K
Ain't Got a Nickel Ain't...
up tempo Memphis soul ...
Soul_Diego
2:48
181K 2.2K
Vapor of Feelings -...
Italo disco, Italo-disco, 19...
midicheat
3:15
681K 6.8K
Give it to me (Suno) 🌳
Electronic, Drum and Ba...
Mr.Tree🌳
3:23
126K 1.1K
Butterflies
Winding, introspective, m...
GTZY
4:24
I Can Wait
Trending Songs
3:08
FOGGY
JUST A FLING
18K 109
Just A Fling
emotional female vocal, ...
Foggy
2:58
6.0K 87
| Click! Click! Click! |
Pop, Dance Pop, Etherea...
GIANLUCA_...
4:26
1.8K 43
Lightheadedness
surf rock punk rock skat...
Teemuth
3:06
813 24
Why No Pineapples o...
Catchy Instrumental Intr...
Zero Nanash...
3:05
2.5K 79
The Only Way Out Is...
Alternative Metal, 2010 ...
Delta Studio
4:57
I Wish - Ft.

Breakout: Likely challenges in building commercial product?

As a group, think about challenges that the team will likely face when turning their research into a product:

- One machine-learning challenge
- One engineering challenge in building the product
- One challenge from operating and updating the product
- One team or management challenge
- One business challenge
- One safety or ethics challenge

Post answer to #lecture on Slack and tag all group members (skip if nobody in group has slack set up yet)

Examples for discussion

- What does correctness or accuracy really mean? What accuracy do customers care about?
- How can we see how well we are doing in practice? How much feedback are customers going to give us before they leave?
- Can we estimate how good our generated music is? How are we doing for different customers or different genres?
- How to present results to the customers?
- When customers complain about poor output, how to prioritize and what to do?
- What are unacceptable mistakes and how can they be avoided? Is there a safety risk?
- Can we cope with an influx of customers?
- Will responding to the same prompt twice produce the same result? Does it matter?
- How can we debug and fix problems? How quickly?

Examples for discussion 2

- With more customers, generation is taking longer and longer – what can we do?
- Generation sometimes crashes. What to do?
- How do we achieve high availability?
- How can we see that everything is going fine and page somebody if it is not?
- We improve our model but somehow system behavior degrades… Why?
- Tensorflow update; does our infrastructure still work?
- Once somewhat successful, how to handle large amounts of data per day?
- Buy more machines or move to the cloud?
- Models are continuously improved. When to deploy? Can we roll back?
- Can we offer live music generation as an app? As a web service?
- Can we get better the longer a user interacts with us? Will this benefit the next song they generate as well?

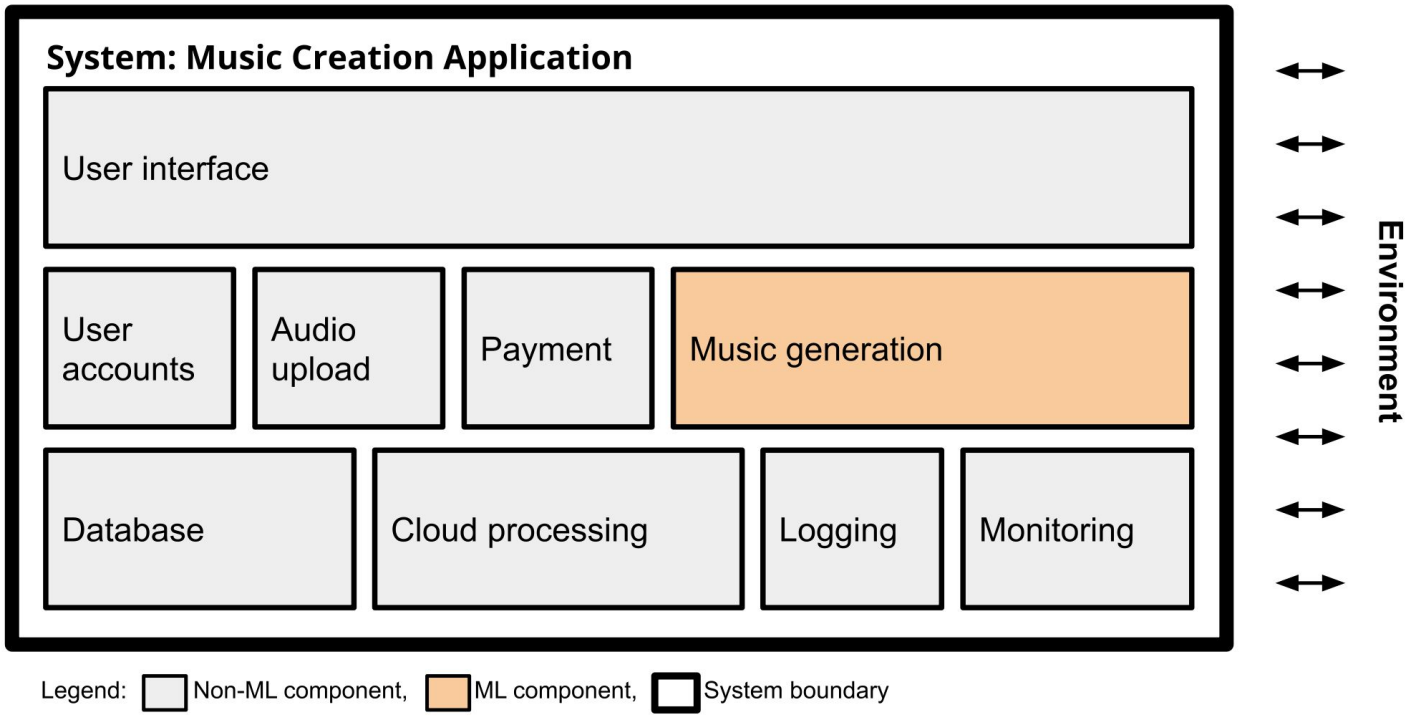
Examples for discussion 3

- How many genres can be supported? Do we have the server capacity?
- How specific should genres be?
- How to make it easy to support new types of music?
- Can we generate different types of vocals? Handle different languages or accents?
- Can and should we learn from customer data?
- How can we debug problems files we are not allowed to see?
- Any chance we might leak private data?
- Can competitors or bad actors attack our system?

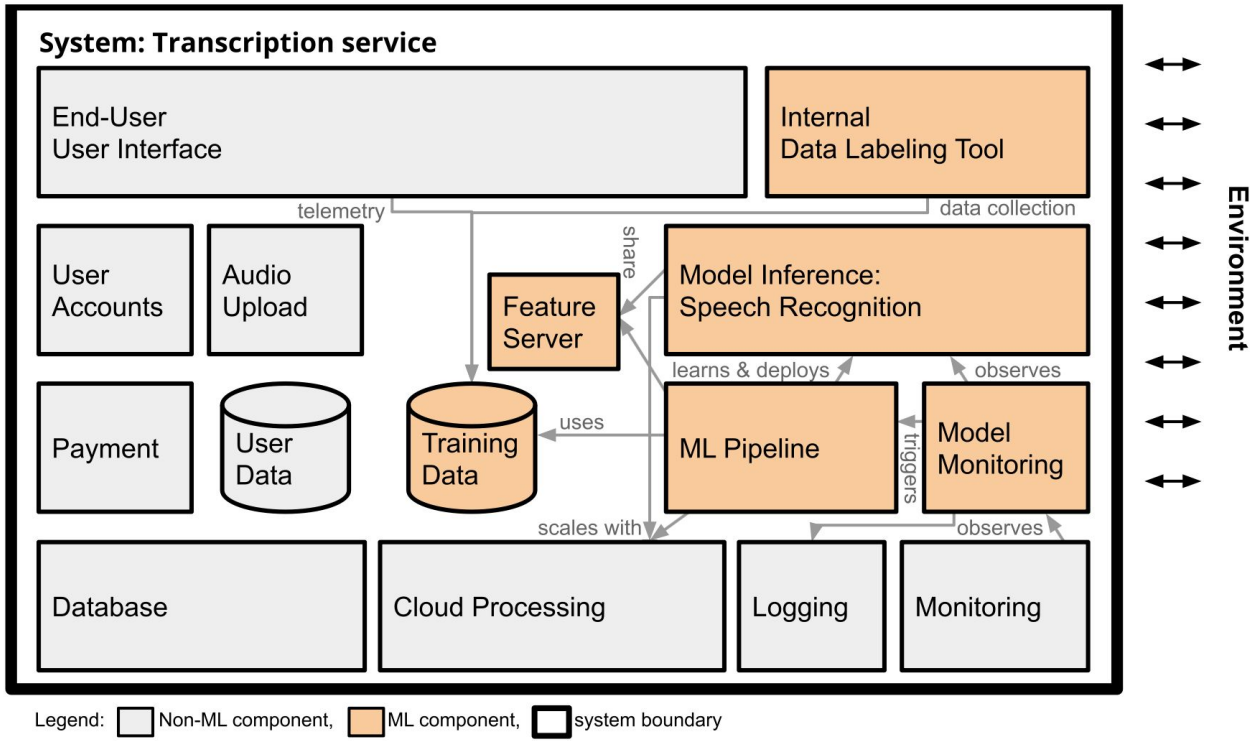
What qualities are important for a good commercial music generator?

What non-ML components are needed?

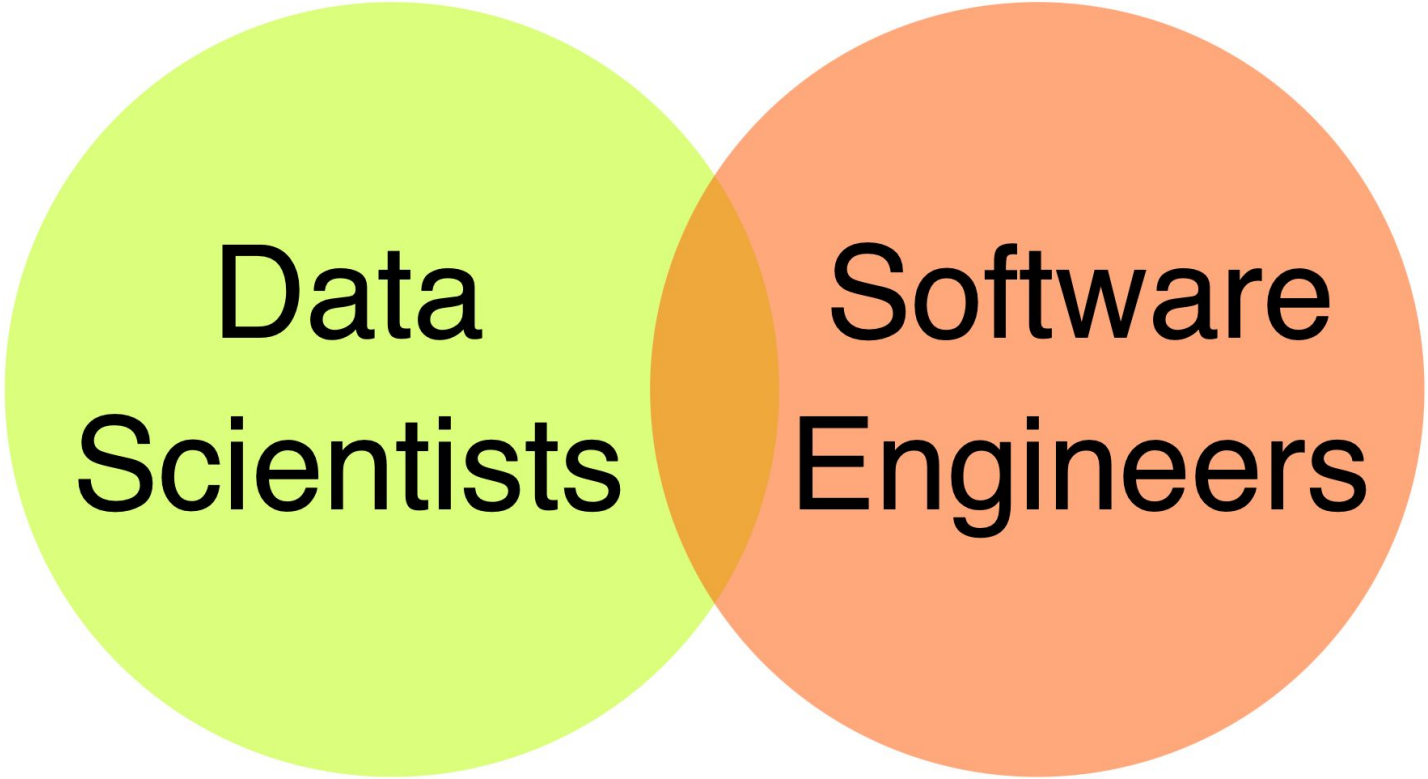
ML in a Production System



ML in a Production System



Development Teams



and Data engineers + Domain specialists + Operators + Business team + Project managers + Designers, UI Experts + Safety, security specialists + Lawyers + Social Scientists + …

Data scientist

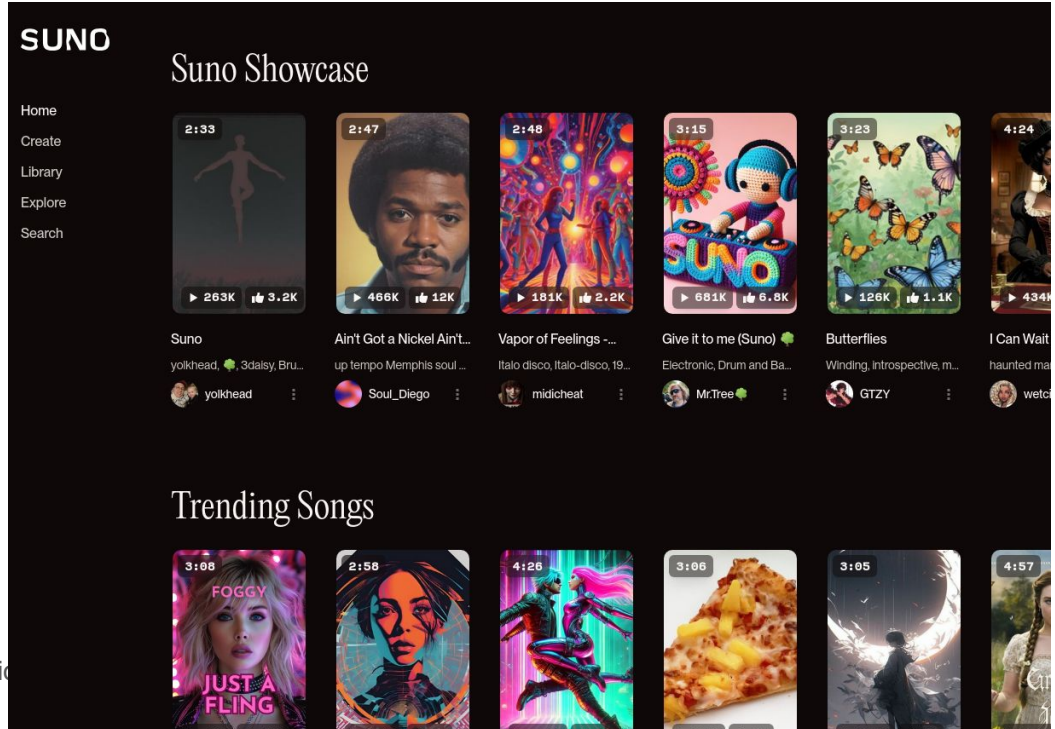
- Often fixed dataset for training and evaluation (e.g., PBS interviews)
- Focused on accuracy
- Prototyping, often Jupyter notebooks or similar
- Expert in modeling techniques and feature engineering
- Model size, updateability, implementation stability typically does not matter

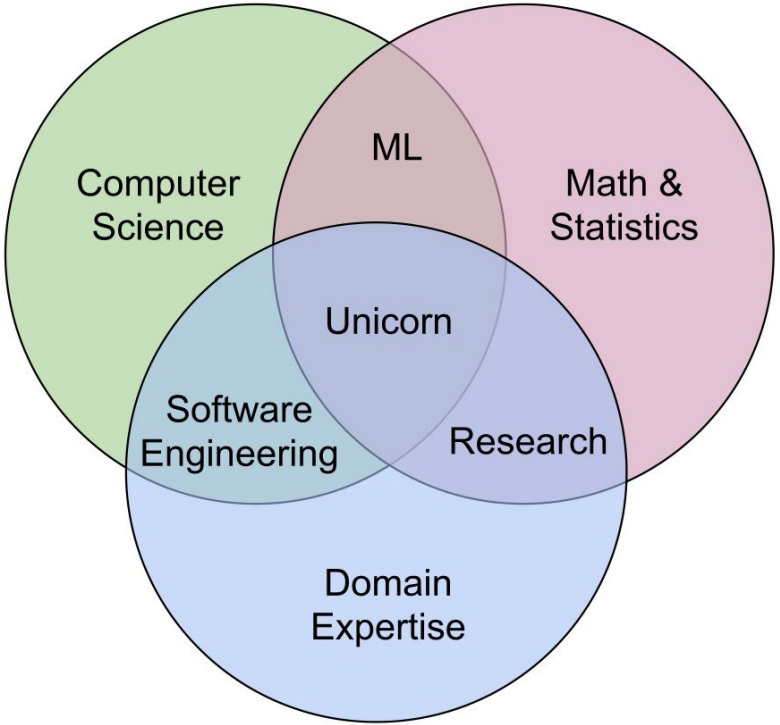
Software engineer

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

Likely collaboration challenges?

What might Software Engineers and Data Scientists focus on?





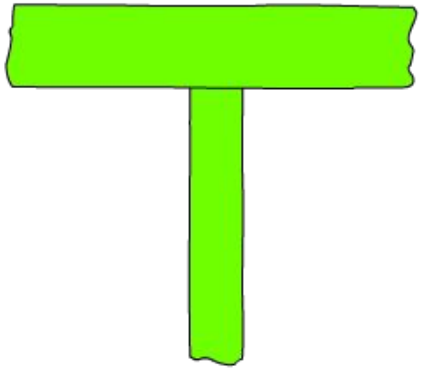
By Steven Geringer, via Ryan Orban. Bridging the Gap Between Data Science & Engineer: Building High-Performance Teams. 2016

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

T-Shaped People

Broad-range generalist + Deep expertise

Example:

- Basic skills of software engineering, business, distributed computing, and communication
- Deep skills in deep neural networks (technique) and medical systems (domain)

Figure: Jason Yip. Why T-shaped people?. 2018

Latest Buzzword: π-Shaped People

Syllabus and Class Structure

17-445/17-645/17-745/11-695, Spring 2025, 12 units

Monday/Wednesdays 2:00-3:20pm

Recitation Fridays 9:30am, 11am, 2pm

Communication

- Email us or ping us on Slack (invite link on Canvas)
- All announcements through Slack #announcements
- Weekly office hours, starting next week, schedule on Canvas
- Post questions on Slack
  - Please use #general or #assignments and post publicly if possible; your classmates will benefit from your Q&A!
- All course materials (slides, assignments, old midterms) available on GitHub and course website: https://mlip-cmu.github.io/s2025/
- Pull requests encouraged!

Class with software engineering flavor

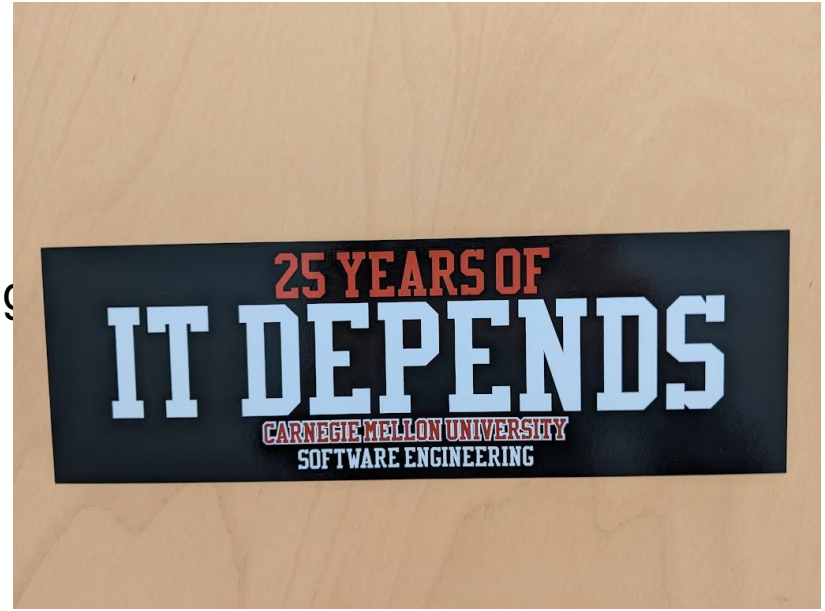
Focusing on engineering judgment

Arguments, tradeoffs, and justification, rather than single correct answer

Practical engagement, building systems, testing automation

Strong teamwork component

Both text-based and code-based homework assignments



Prerequisites

Some machine learning experience required

- Basic understanding of data science process, incl. data cleaning, feature engineering, using ML libraries
- High level understanding of machine-learning approaches
  - Supervised learning
  - Regression, decision trees, neural networks
  - Accuracy, recall, precision, ROC curve
- Ideally, some experience with notebooks, sklearn, or other frameworks

Basic programming and command-line skills will be needed

No further software-engineering knowledge required

- Teamwork experience in product team is useful but not required
- No required exposure to requirements, software testing, software design, continuous integration, containers, process management, etc.
  - If you are familiar with these, there will be some redundancy – sorry!

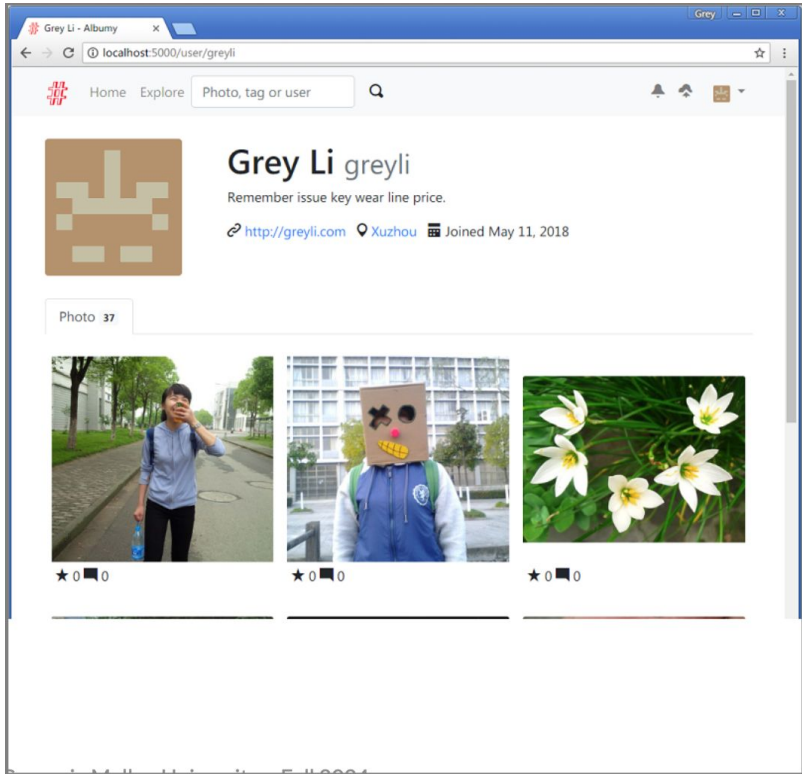
First Homework Assignment I1

“Coding warmup assignment”

Out now, due Monday Jan 27

Enhances simple web application with ML-based features: Image search and automated captioning

Open-ended coding assignment, change existing code, learn new APIs, solve dependency issue



Active lecture

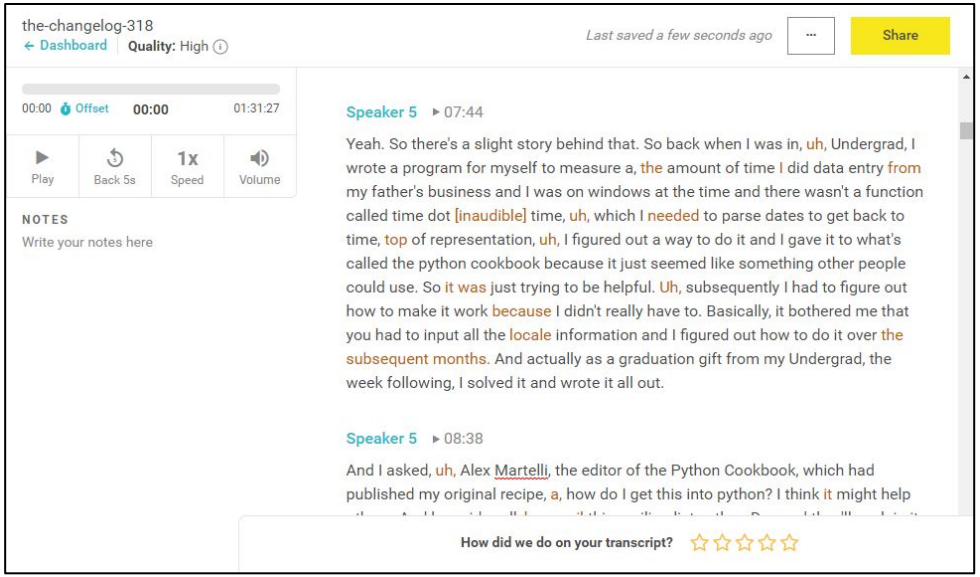
Case study driven

Discussions highly encouraged

Regular in-class activities, breakouts

Contribute your own experience!

Discussions over definitions



Recordings and Attendance

Try to attend lecture – discussions are important to learning

Participation is part of your grade

We make a *best effort* to record lecture; they will be linked from Canvas under “zoom.”

- Sometimes, the recording will be bad or absent. :shrug:
- We do *not* provide synchronous remote attendance or participation.

Watching the lecture post facto does not allow you to make up the participation.

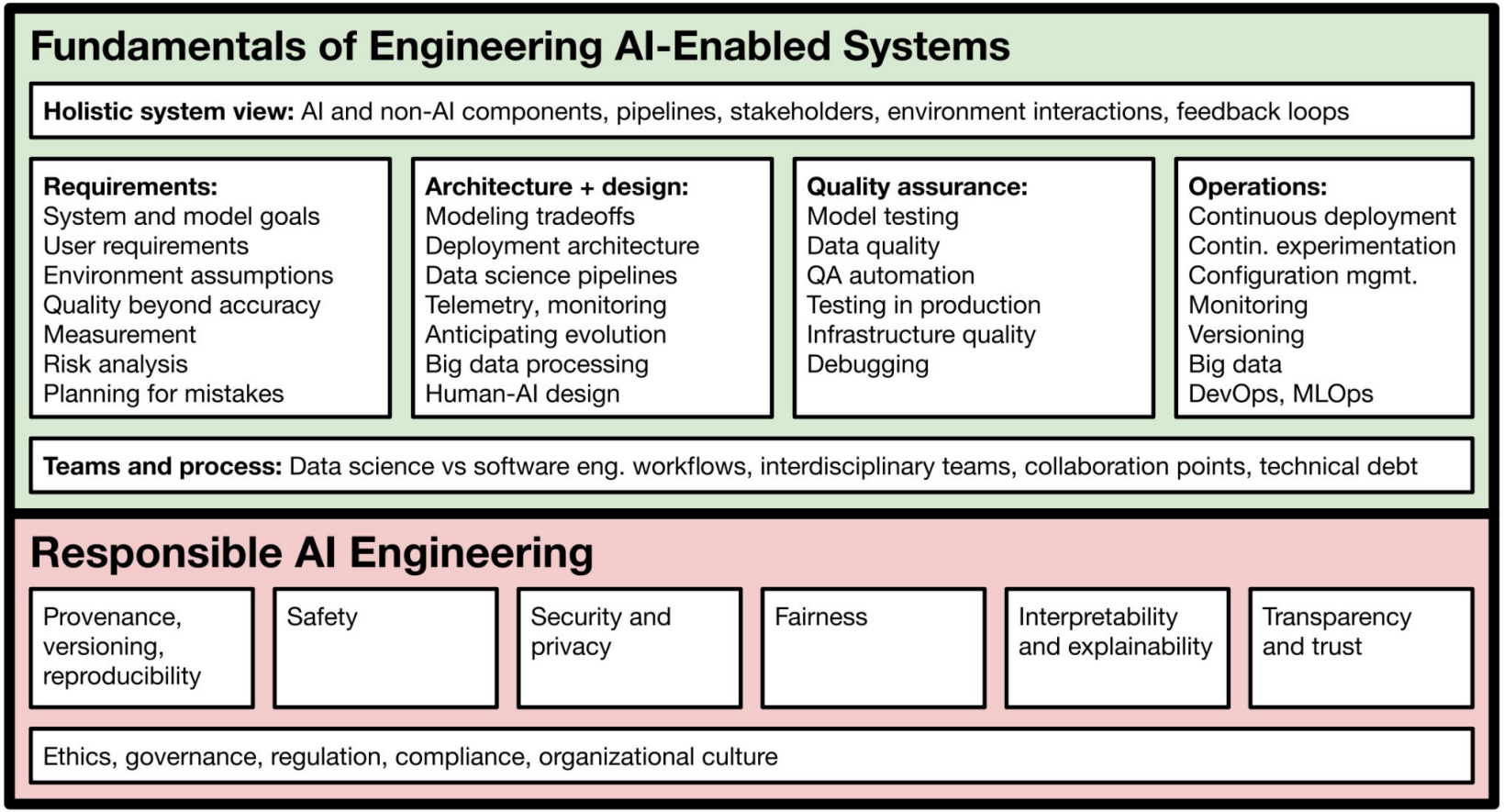
Contact us for accommodations (illness, interview travel, unforeseen events), or have your advisor reach out. We try to be flexible.

Participation

Participation != Attendance

Grading:

- 100%: Participates actively at least once in most lectures by (1) asking or responding to questions or (2) contributing to breakout discussions
- 90%: Participates actively at least once in two thirds of the lectures
- 75% Participates actively at least once in over half of the lectures
- 50%: Participates actively at least once in one quarter of the lectures
- 20%: Participates actively at least once in at least 3 lectures


Fundamentals of Engineering AI-Enabled Systems
Holistic system view: AI and non-AI components, pipelines, stakeholders, environment interactions, feedback loops
Requirements:
System and model goals
User requirements
Environment assumptions
Quality beyond accuracy
Measurement
Risk analysis
Planning for mistakes
Architecture + design:
Modeling tradeoffs
Deployment architecture
Data science pipelines
Telemetry, monitoring
Anticipating evolution
Big data processing
Human-AI design
Quality assurance:
Model testing
Data quality
QA automation
Testing in production
Infrastructure quality
Debugging
Operations:
Continuous deployment
Contin. experimentation
Configuration mgmt.
Monitoring
Versioning
Big data
DevOps, MLOps
Teams and process: Data science vs software eng. workflows, interdisciplinary teams, collaboration points, technical debt
Responsible AI Engineering
Provenance, versioning, reproducibility
Safety
Security and privacy
Fairness
Interpretability and explainability
Transparency and trust
Ethics, governance, regulation, compliance, organizational culture

Reading Assignments & Quizzes

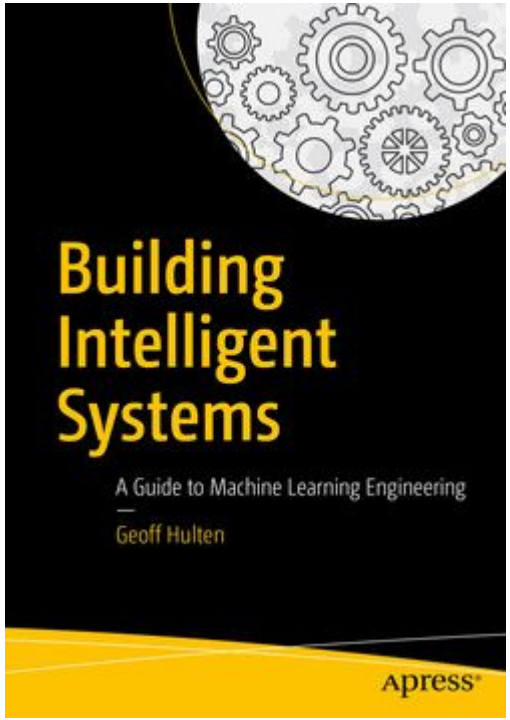
Building Intelligent Systems by Geoff Hulten

https://www.buildingintelligentsystems.com

Most chapters assigned at some point in the semester

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

Electronic version in the library



Reading Quizzes

Short essay questions on readings, due before start of lecture (Canvas quiz)

Planned for: about 30-45 min for reading, 15 min for discussing and answering quiz

Book for the Class

“Machine Learning in Production: From Models to Products”

Mostly similar coverage to lecture

Not required, use as a supplementary reading

Published online: https://mlip-cmu.github.io/book/ (and as printed MIT Press book in April)

Assignments

Most assignments available on GitHub now

Series of 4 small to medium sized **individual assignments**

- Engage with practical challenges
- Reason about tradeoffs and justify your decisions
- Integrate models and explanations into end-user products
- Written reports, a little modeling, some coding

Large team project with 4 milestones:

- Build and deploy a prediction (movie recommendation) service
- Testing in production, monitoring
- Final presentation

Usually due Monday night; see schedule

Research in this Course

We are conducting academic research in this course.

This research will involve analyzing student work of assignment **after the end of the semester.**

You will not be asked to do anything above and beyond the normal learning activities and assignments that are part of this course. All data will be analyzed in de-identified form and presented in the aggregate, without any personal identifiers.

You are free not to participate in this research, and your participation will have no influence on your grade for this course or your academic career at CMU. If you do not wish to participate, please send an email to Nadia Nahar (nadian@andrew.cmu.edu); instructors will not know who opts out before assigning final grades.

See syllabus for details.

17-745 PhD Research Project

Research project instead of individual assignments I3 and I4

Design your own research project and write a report

- A case study, empirical study, literature survey, etc.

Very open ended: Align with own research interests and existing projects

See the project requirements and talk to us

First hard milestone: initial description due Feb 27

Labs

Please attend the lab for which you are registered.

Introducing various tools, e.g., fastAPI (serving), Kafka (stream processing), Jenkins (continuous integration), MLflow (experiment tracking), Docker & Kubernetis (containers), Prometheus & Grafana (monitoring), CHAP (explainability)...

Hands on exercises, bring a laptop

Often introducing tools useful for assignments

About 1h of work, graded pass/fail, low stakes, show work to TA

First lab **this Friday**: Calling, securing, and creating APIs

Lab grading and collaboration

We recommend that you start each lab before the recitation, but can be completed during

Graded pass/fail by TA on the spot, can retry

Relaxed collaboration policy: Can work with others before and during recitation, but have to present/explain solution to TA individually

(Think of recitations as mandatory office hours)

Grading

- 35% individual assignment
- 30% group project with final presentation
- 15% two in-class exams
- 5% participation
- 5% reading quizzes
- 10% labs
- No final exam (final presentations will take place in that time slot)

Expected grade cutoffs in syllabus (>82% B, >94 A-, >96% A, >99% A+)

Grading Philosophy

Specification grading, based in adult learning theory

Giving you choices in what to work on or how to prioritize your work

We are making every effort to be clear about expectations (specifications), will clarify if you have questions

Assignments broken down into expectations with point values, each graded **pass/fail**

Opportunities to resubmit work until last day of class

[Example]

Token System for Flexibility

8 individual tokens per student:

- Submit individual assignment 1 day late for 1 token (after running out of tokens, 15% penalty per late day)
- Redo individual assignment for 3 token
- Resubmit or submit reading quiz late for 1 token
- Redo or complete a lab late for 1 token (show in office hours)
- Remaining tokens count toward participation

8 team tokens per team:

- Submit milestone 1 day late for 1 token (no late submissions accepted when out of tokens)
- Redo milestone for 3 token

How to use tokens

- No need to tell us if you plan to submit very late. We will assign 0 and you can resubmit
- Instructions and Google form for resubmission on Canvas (pages)
- We will automatically use remaining tokens toward participation at the end
- Remaining individual tokens reflected on Canvas, for remaining team tokens, ask your team mentor.

Group project

- Instructor-assigned teams
- Teams stay together for project throughout semester.
- There will be a Catme Team survey before Y (3pt)
- Some advice in lectures; we'll help with debugging team issues
- TA assigned to each team as mentor; mandatory debriefing with mentor and peer grading on all milestones (based on citizenship on team)
- Bonus points for social interaction in project teams

Academic honesty

- See web page
- In a nutshell: do not copy from other students, do not lie, do not share or publicly release your solutions
- In group work, be honest about contributions of team members, do not cover for others
- Collaboration okay on labs, but not quizzes, individual assignments or exams
- If you feel overwhelmed or stressed, please come and talk to use (see syllabus for other support opportunities)

Thoughts on Generative AI for Homework?

GPT4, ChatGPT, Copilot, …? Reading quizzes, homework submissions, …?

Our Position on Generative AI for Homework

This is a course on responsible building of ML products. This includes questions of how to build generative AI tools responsibly and discussing what use is ethical.

Feel free to use them and explore whether they are useful. Welcome to share insights/feedback.

Warning: Be aware of hallucinations. Requires understanding to check answers. We test them; they often generate bad/wrong answers for reading quizzes.

You are responsible for the correctness of what you submit!

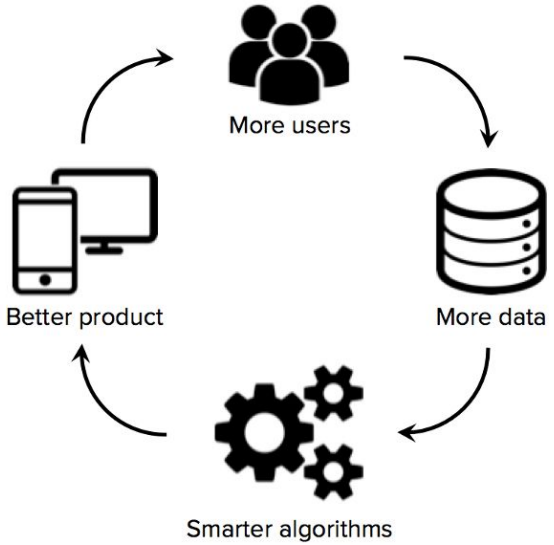
What makes software with ML challenging?

- Traditional SE: specify what to do & how to test
- MLs are usually black boxes (maybe justified, for their complexity…)
- Even if you put specs in, e.g., LLM prompts, unclear if they will follow

```
/**
  Return the text spoken within the audio file
  ????
*/
String transcribe(File audioFile);
```

Data Focused and Scalable

- MLs get the “specs” from data; Larger the better
- *Deductive reasoning* (applying logic rules) to *Inductive Reasoning* (generalizing from observation)
- Cause scalability issues



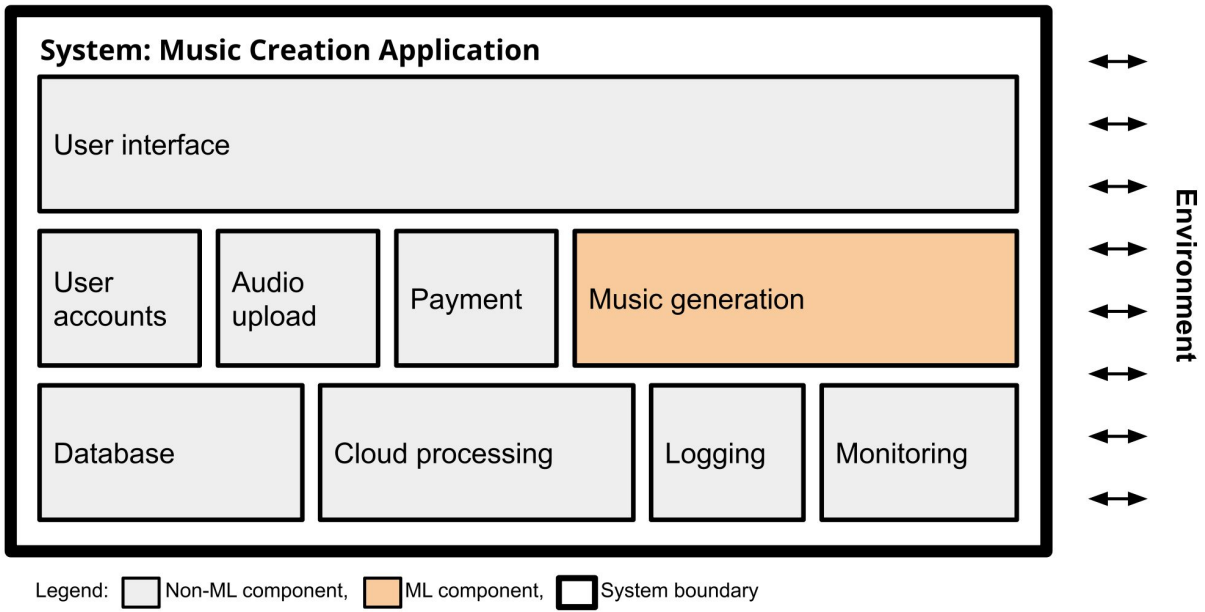
ML Models Make Mistakes

- …Often in unexpected ways
- Hard to foresee and capture because no spec
- What does it mean to be correct? Can only evaluate whether it works well enough (on average) on some test data!

NeuralTalk2: A flock of birds flying in the air
Microsoft Azure: A group of giraffe standing next to a tree
Image: Fred Dunn, https://www.flickr.com/photos/gratapictures - CC-BY-NC

Interaction with the environment

Our system must be able to tolerate some incorrect predictions, and be aware of how it might influence the world…



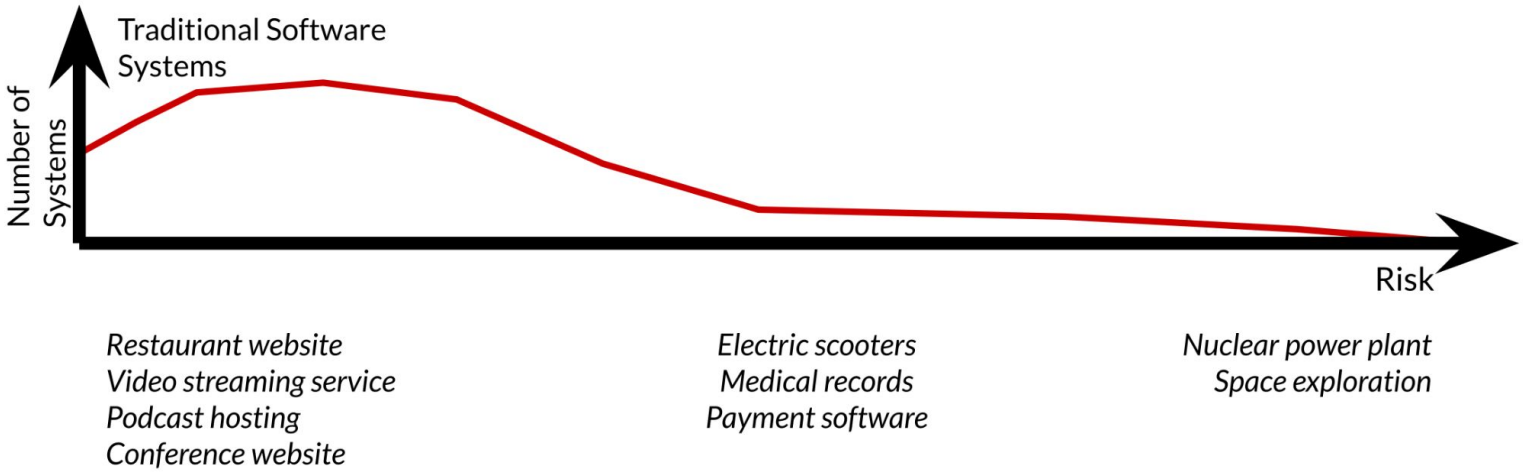
It’s not all new

We routinely build:

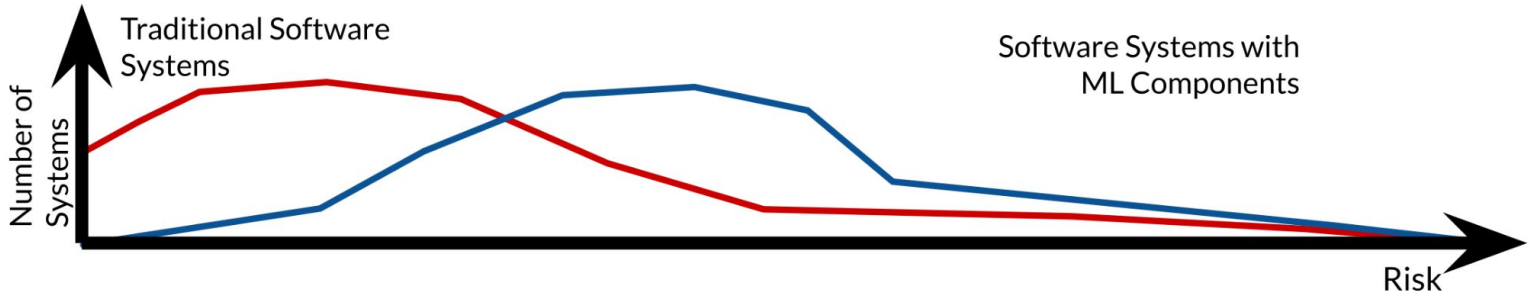
- Safe software with unreliable components
- Non-ML big data systems, cloud systems
- “Good enough” and “fit for purpose” not “correct”
- Cyberphysical systems

ML intensifies our challenges

Complexity


Traditional Software Systems
Number of Systems
Risk
Restaurant website
Video streaming service
Podcast hosting
Conference website
Electric scooters
Medical records
Payment software
Nuclear power plant
Space exploration

Complexity



Game AI
Traffic planning
Photo filters

Photo search
Social media feeds
Sidewalk robot delivery
Smart home devices
Automated transcription

Autonomous vehicles
Predictive policing
Deep Fakes

Introductions

Before the next lecture, introduce yourself in Slack channel #social:

- Your (preferred) name
- In 1-2 sentences, your data science background and goals (e.g., coursework, internships, work experience)
- In 1-2 sentences, your software engineering background, if any, and goals (e.g., coursework, internships, work experience)
- One topic you are particularly interested in learning during this course?
- A hobby or a favorite activity outside school

Summary

Machine learning components are part of larger systems

Data scientists and software engineers have different goals and focuses

- Building systems requires both
- Various qualities are relevant, beyond just accuracy

Machine learning brings new challenges and intensifies old ones