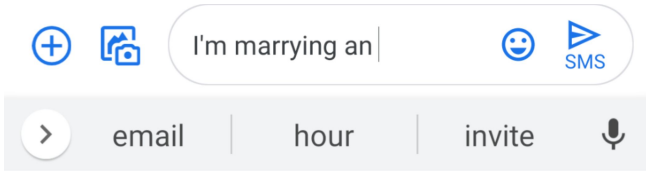
Machine Learning Systems Design

Lecture 2: Designing an ML system

How well does your phone know you?

1. Open any conversation on your phone
2. Type “I’m marrying an” in the response
3. Type in Zoom chat your phone’s first suggestion



Stanford University

Logistics

- Note posted online
- OHs start next week (Zoom links on Canvas)
  - Wed 8.30am - 9am (Chip)
  - Thu 8pm - 8.30pm (Karan)
  - Sun 1pm - 1.30pm (Xi / Michael) - not this Sun!
- No class next Monday
- Assignment 1 out this weekend
- Final project instruction out this weekend

Zoom etiquettes

- Write questions into Zoom chat
  - Feel free to reply to each other — TAs will also reply
- I will stop occasionally for Q&A
  - TAs will re-share some of the questions with me
- After each lecture, a random question will get a random reward

Agenda

1. Goals of ML systems design
2. Different types of ML systems
3. Iterative process
4. Project scoping
5. Breakout exercise
6. When to use ML and when not to use ML
7. Four phases of ML adoption

1. Goals of ML systems design

What’s machine learning systems design?

The process of defining the **interface, algorithms, data**, **infrastructure**, and **hardware** for a machine learning system to satisfy **specified requirements**.

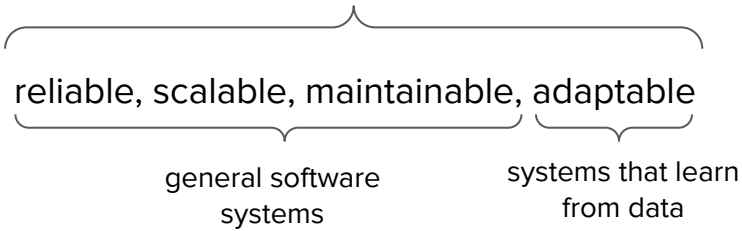
What’s machine learning systems design?

The process of defining the **interface, algorithms, data**, **infrastructure**, and **hardware** for a machine learning system to satisfy **specified requirements**.

reliable, scalable, maintainable, adaptable

What’s machine learning systems design?

The process of defining the **interface, algorithms, data**, **infrastructure**, and **hardware** for a machine learning system to satisfy **specified requirements**.



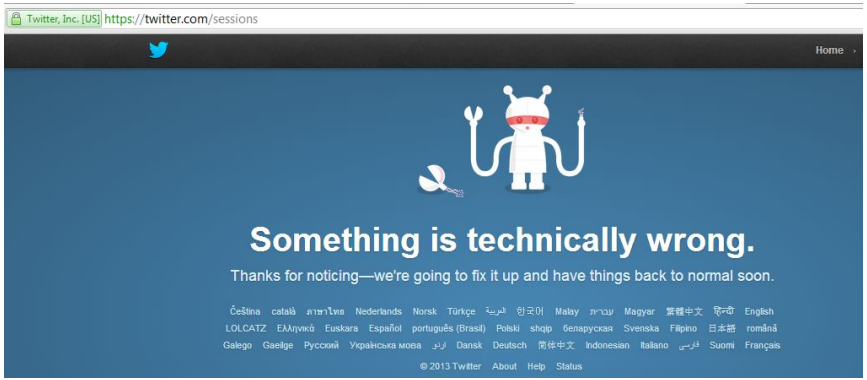
Reliability

The system should continue to perform the **correct function** at the **desired level of performance** even in the face of adversity (hardware or software faults, and even human error).

What does “correct” mean for ML systems when there are no ground truth labels?

ML systems fail silently

Normal software fails



ML systems fails

Scalability

As the system grows (in data volume, traffic volume, or complexity), there should be reasonable ways of dealing with that growth.

We'll focus on systems at scale in this course!

Scalability

As the system grows (in data volume, traffic volume, or complexity), there should be reasonable ways of dealing with that growth.

Autoscaling: the number of machines can go up or down depending on usage

Scalability: cautionary tale

Amazon's one hour of downtime on Prime Day may have cost it up to $100 million in lost sales

Sean Wolfe
Jul 19, 2018, 10:53 AM

> "If their auto-scaling was working, things would have scaled automatically and they wouldn't have had this level of outage," Caesar said. "There was probably an implementation or configuration error in their automatic scaling systems."

Internal documents show how Amazon scrambled to fix Prime Day glitches (Eugene Kim, CNBC 2018)

Maintainability

Over time, many people (ML engineers, DevOps, **subject matter experts**) will work on the system, and they should all be able to work on it productively.

The importance of SMEs in ML systems

- **Subject matter experts** (doctors, lawyers, bankers, farmers, stylists, etc.) are not only users but also developers of ML systems.
- Domain expertise is needed for:
  - problem formulation
  - data labeling
  - feature engineering
  - error analysis
  - model evaluation
  - reranking predictions

Maintainability: cross-team collaboration

- How to help engineers and SMEs communicate effectively?

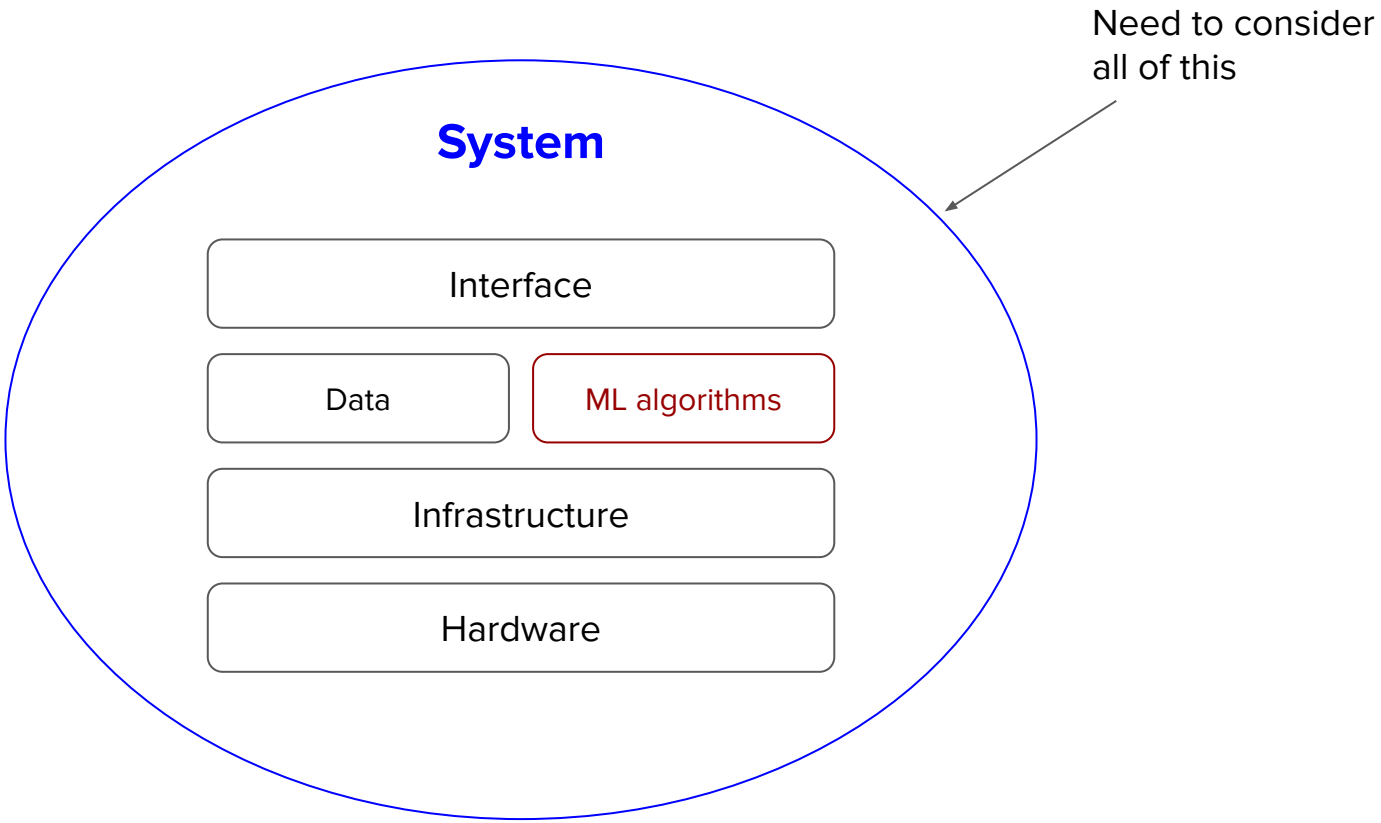
New hot keywords: no-code / low-code ML

Adaptability

To adapt to **changing data distributions** and **business requirements**, the system should have some capacity for both **discovering aspects for performance improvement** and **allowing updates without service interruption**.

Linked to maintainability

We'll cover more about this later!


Need to consider
all of this
System
Interface
Data
ML algorithms
Infrastructure
Hardware

2. Different types of ML systems

⚠️⚠️ The dangers of categorical thinking ⚠️⚠️

- Seemingly different ways of doing things might be fundamentally similar
- Choices don’t have to be mutually exclusive
- Choices can evolve over time

The Dangers of Categorical Thinking

We’re hardwired to sort information into buckets—and that can hamper our ability to make good decisions. by Bart de Langhe and Philip Fernbach

Batch prediction vs. online prediction

Two options for predictions offered by major cloud providers
(valid when using your own data centers too)

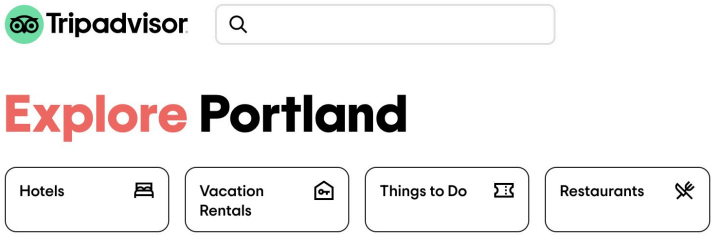
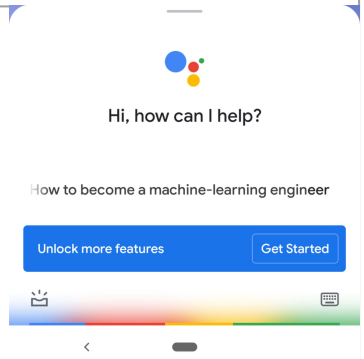
AI Platform Prediction provides two ways to get predictions from trained models: *online prediction* (sometimes called HTTP prediction), and *batch prediction*. In both cases, you pass input data to a cloud-hosted machine-learning model and get inferences for each data instance.

Batch prediction vs. online prediction

- Batch prediction:
  - Generate predictions periodically
  - Predictions are stored somewhere (e.g. SQL tables, CSV files)
  - Retrieve them as needed
  - Allow more complex models
- Online prediction:
  - Generate predictions as requests arrive
  - Predictions are returned as responses

⚠️ Misnomer ⚠️

- Both can do one or more samples (batch) at a time
- If you do compute on the cloud, then both are technically “online” - over the Internet

	Batch prediction	Online prediction
Frequency	Periodical (e.g. every 4 hours)	As soon as requests come
Useful for	Processing accumulated data when you don't need immediate results (e.g. recommendation systems)	When predictions are needed as soon as data sample is generated (e.g. fraud detection)
Optimized	High throughput	Low latency
Input space	Finite: need to know how many predictions to generate	Can be infinite
Examples	• TripAdvisor hotel ranking • Netflix recommendations 	• Google Assistant speech recognition • Twitter feed 

Hybrid: batch & online prediction

- Online prediction is default, but common queries are precomputed and stored
- DOORDASH
  - Restaurant recommendations use batch predictions
  - Within each restaurant, item recommendations use online predictions
- NETFLIX
  - Title recommendations use batch predictions
  - Row orders use online predictions

Edge computing vs. cloud computing

	Cloud computing	Edge computing
Computations	Done on cloud (servers)	Done on edge devices (browsers, phones, tablets, laptops, smart watches, activity watchers, cars, etc.)
Requirements	**Network connections**: availability and speed for data transfer	**Hardware**: memory, compute power, energy for doing computations
Examples	• Most queries to Alexa, Siri, Google Assistant • Google Translate for rare language pairs (e.g. English - Yiddish)	• Wake words for Alexa, Siri, Google Assistant • Google Translate for popular language pairs (e.g. English - Spanish) • Predictive text • Unlocking with fingerprints, faces

Benefits of edge computing

- Can work without (Internet) connections or with unreliable connections
  - Many companies have strict no-Internet policy
  - **Caveat**: devices are capable of doing computations but apps need external information
    - e.g. ETA needs external real-time traffic information to work well
- Don't have to worry about network latency
  - Network latency might be a bigger problem than inference latency
  - Many use cases are impossible with network latency
    - e.g. predictive texting
- Fewer concerns about privacy
  - Don't have to send user data over networks (which can be intercepted)
  - Cloud database breaches can affect many people
  - Easier to comply with regulations (e.g. GDPR)
  - **Caveat**: edge computing might make it easier to steal user data by just taking the device
- Cheaper
  - The more computations we can push to the edge, the less we have to pay for servers

Many companies invest heavily in better chips

Musk Boasts Tesla Has 'Best Chip in the World'

⚠️ unreliable narrator ⚠️

The CEO's newest big prediction: that Tesla will have self-driving cars on the road next year.

Bloomberg
APR 23, 2019

More on edge devices later!

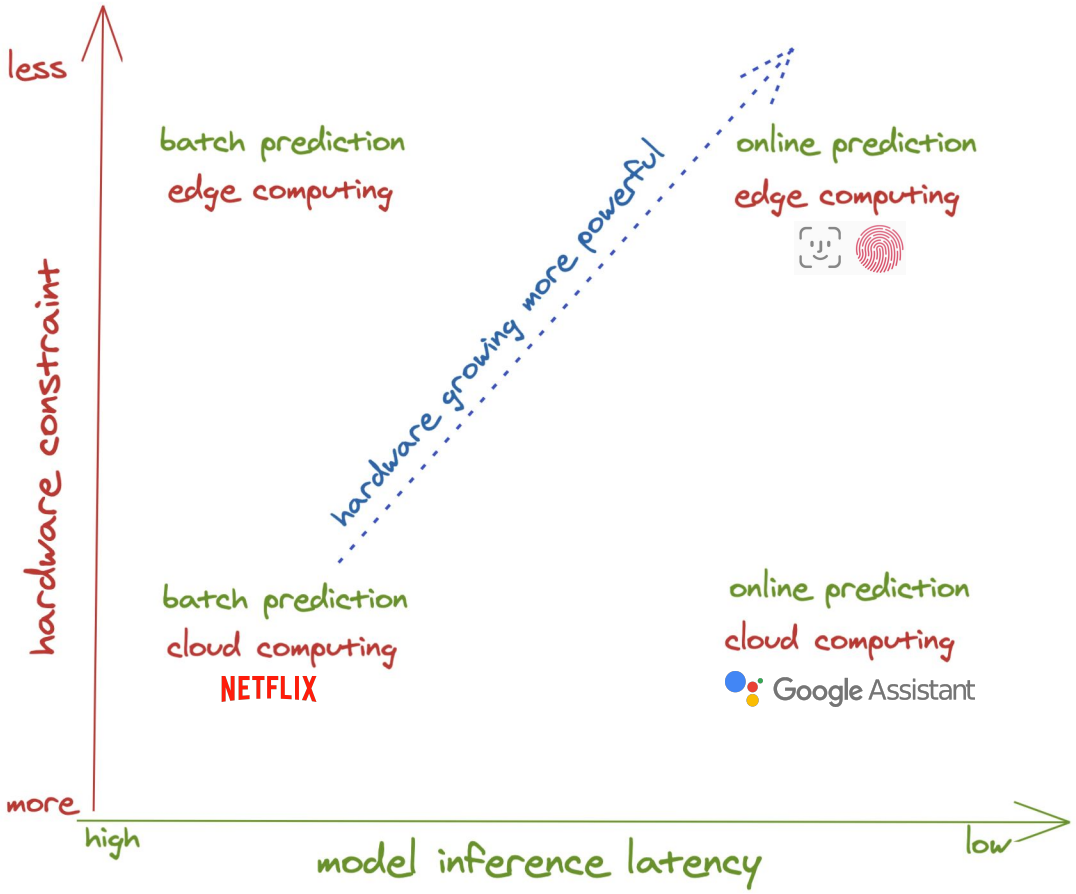
Apr 14, 2020 - Technology

Scoop: Google readies its own chip for future Pixels, Chromebooks

Hybrid

- Common predictions are precomputed and stored on device
- Local data centers: e.g. each warehouse has its own server rack
- Predictions are generated on cloud and cached on device

Future of ML: online and on-device



Offline learning vs. online learning

Harder & less common

	Offline learning	Online learning
Iteration cycle	Periodical (months)	Continual (minutes) ≠ continuous
Batch size	batch (thousands -> millions of samples) GPT-3 125M params: batch size 0.5M GPT-3 175B params: batch size 3.2M	microbatch (hundreds of samples)
Data usage	Each sample seen multiple times (epochs)	Each sample seen at most once
Evaluation	Mostly offline evaluation	Offline evaluation as sanity check Mostly relying on online evaluation (A/B testing)
Examples	Most applications	TikTok recommendation system, Twitter hashtag trending

Online learning vs. offline learning

- Both can be done together to create more stable systems
- **If the infrastructure is set up right**, there's no fundamental difference between online learning and offline learning, just a hyperparam to tune.

More on online learning later!

3. Iterative process

ML in production: expectation

1. Collect data
2. Train model
3. Deploy model
4. 

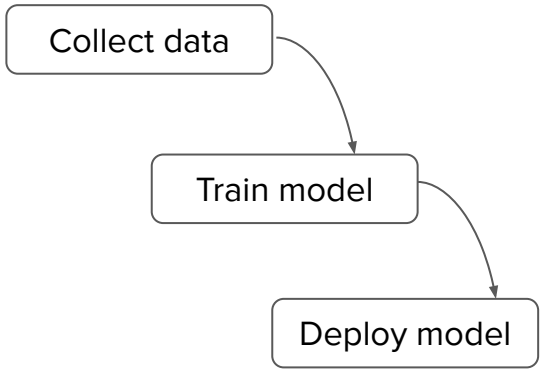
ML in production: reality

1. Choose a metric to optimize
2. Collect data
3. Train model
4. Realize many labels are wrong -> relabel data
5. Train model
6. Model performs poorly on one class -> collect more data for that class
7. Train model
8. Model performs poorly on most recent data -> collect more recent data
9. Train model
10. Deploy model
11. Dream about $$$
12. Wake up at 2am to complaints that model biases against one group -> revert to older version
13. Get more data, train more, do more testing
14. Deploy model
15. Pray
16. Model performs well but revenue decreasing
17. Cry
18. Choose a different metric
19. Start over

Step 15 and 17 are essential

ML in production: expectation

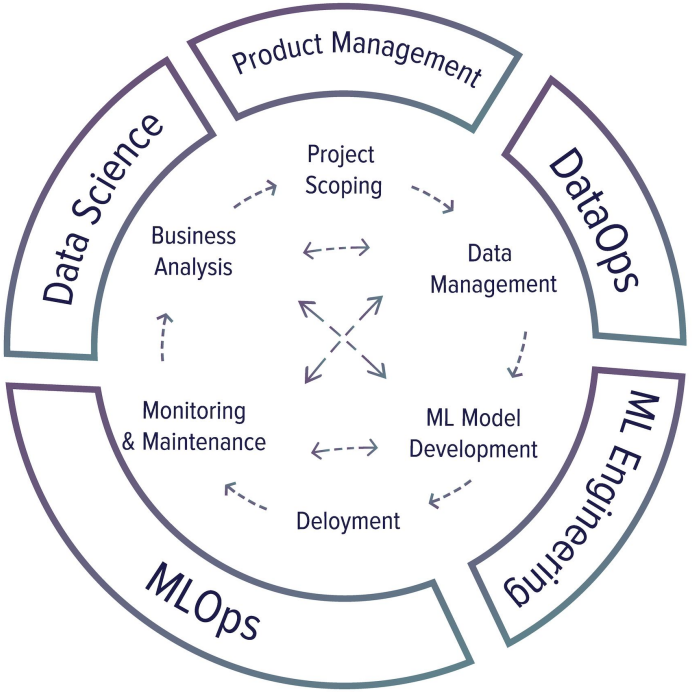
1. Collect data
2. Train model
3. Deploy model
4. 



Waterfall model

ML in production: reality

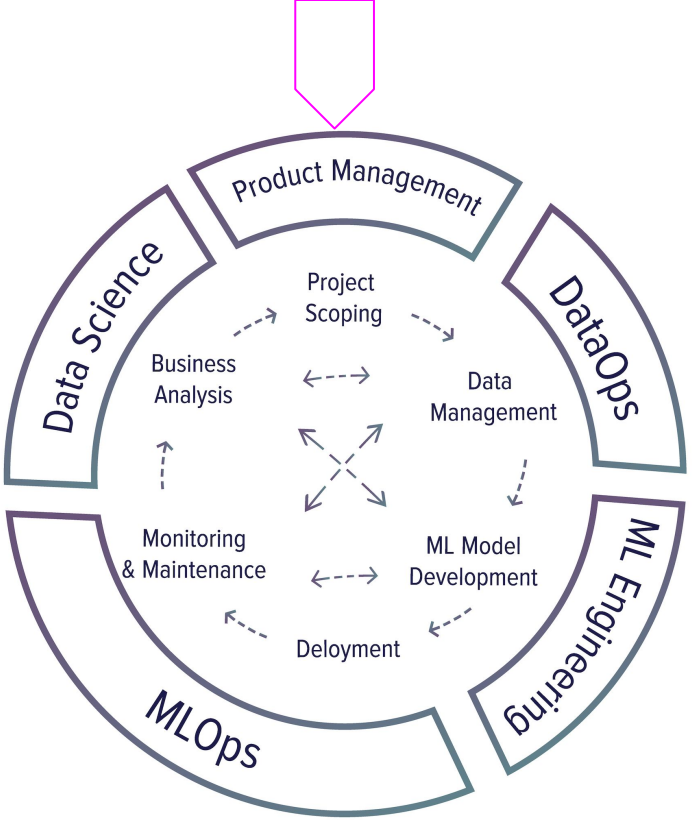
1. Choose a metric to optimize
2. Collect data
3. Train model
4. Realize many labels are wrong -> relabel data
5. Train model
6. Model performs poorly on one class -> collect more data for
7. Train model
8. Model performs poorly on most recent data -> collect more
9. Train model
10. Deploy model
11. Dream about $$$
12. Wake up at 2am to complaints that model biases against or
13. Get more data, train more, do more testing
14. Deploy model
15. Pray
16. Model performs well but revenue decreasing
17. Cry
18. Choose a different metric
19. Start over



Iterative development

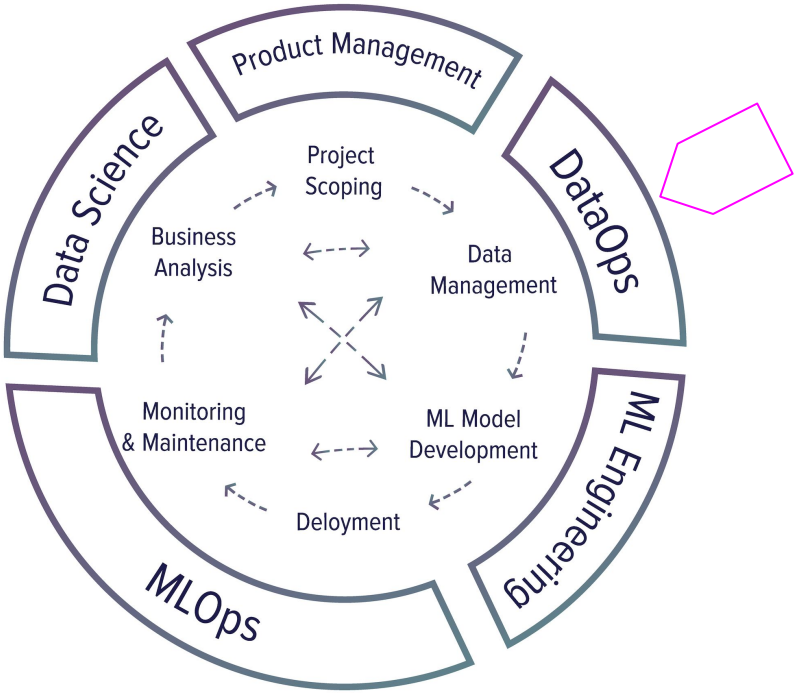
Project scoping

- Goals & objectives
- Constraints
- Evaluation



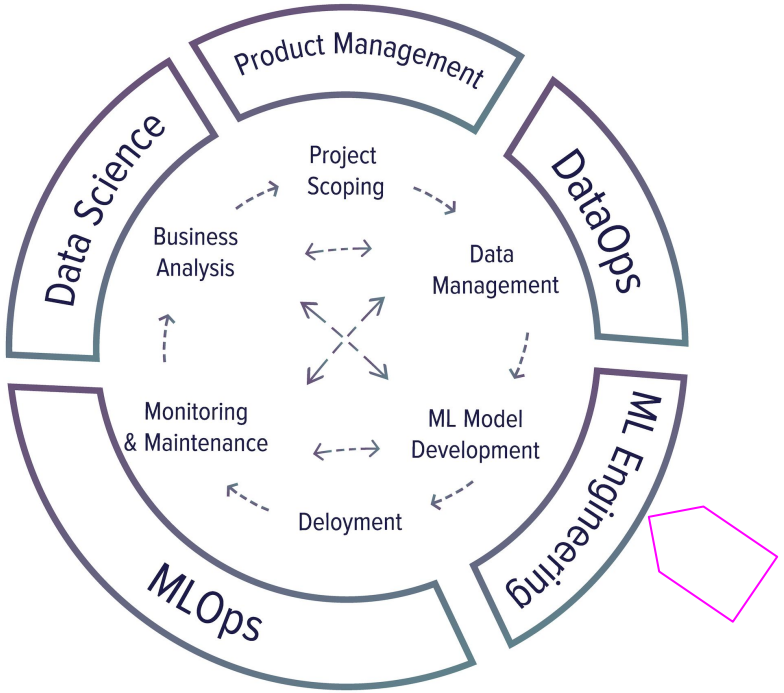
Data management

- Data sources
- Data format
- Processing
- Storage
- Data consumer
- Data controller



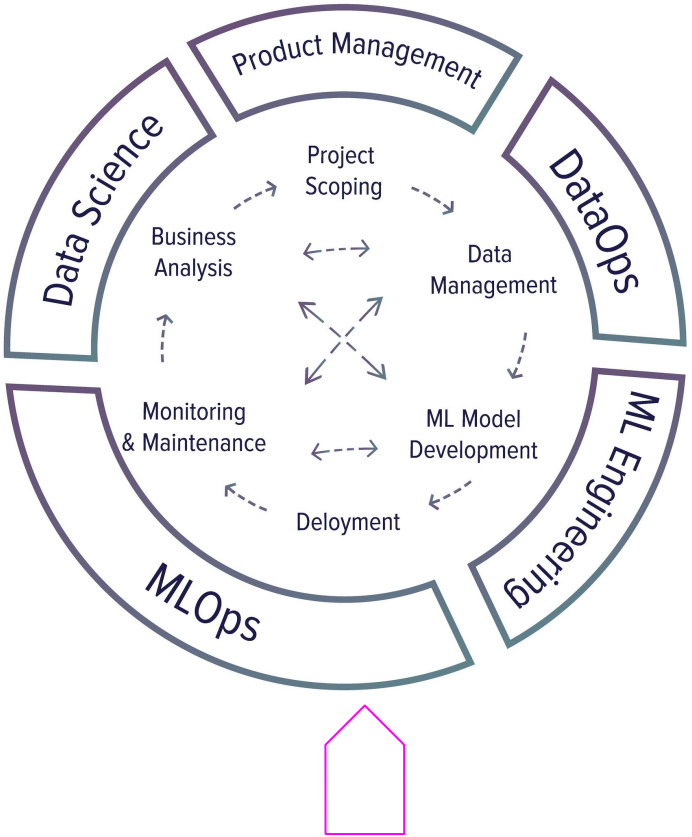
Model development

- Dataset creation
- Feature engineering
- Model training
- Offline model evaluation



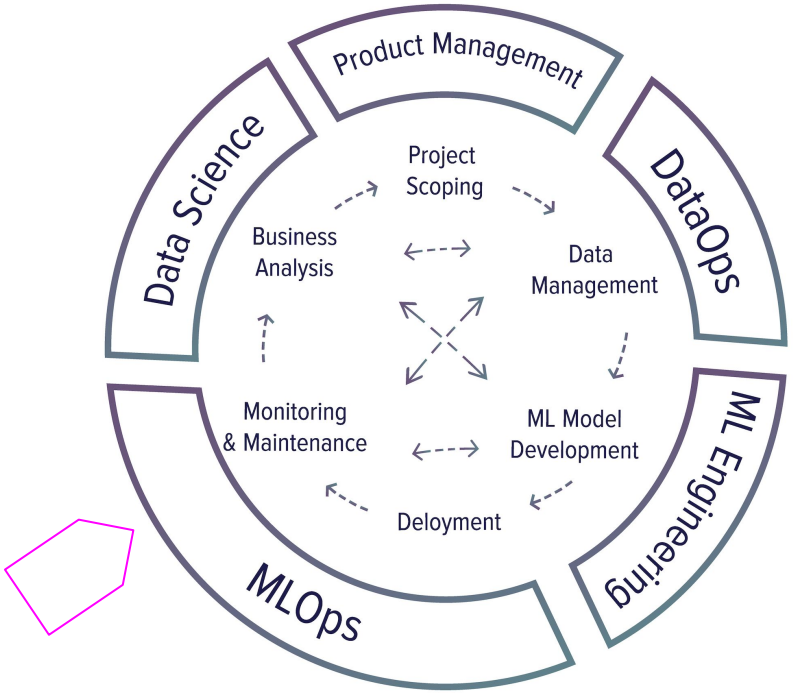
Deployment

- Deploying and serving
- Release strategies
- Online model evaluation



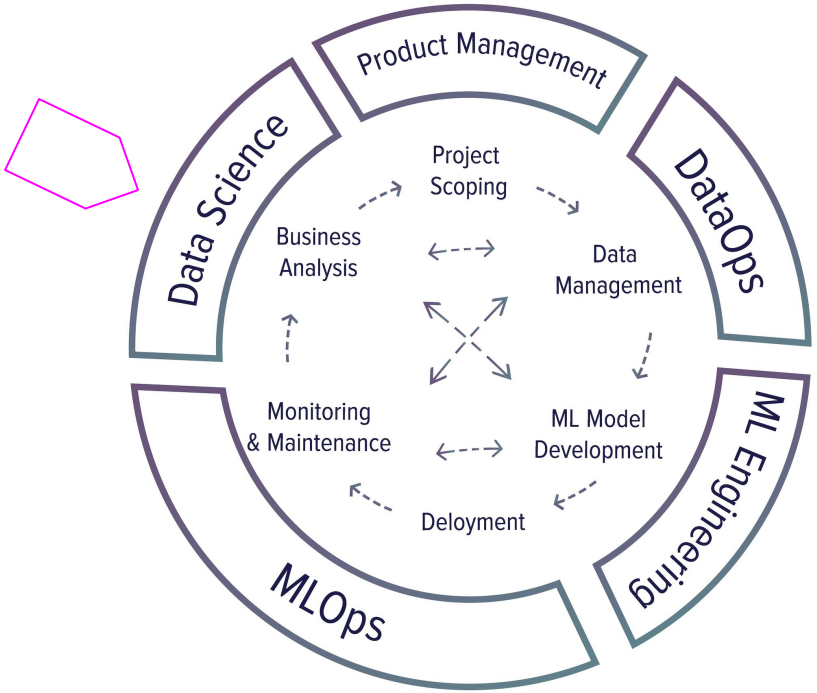
Monitoring & maintenance

- Model performance & data monitoring
- Model retraining
- Model updates



Business analysis

- User experience
- Tying model performance to business performance



4. Project scoping

Goals

“The social responsibility of a business is to increase its profits.”

Milton Friedman (The New York Times, 1970)

Goals

The ultimate goal of an ML project or any project within a business is, therefore, to increase profits directly or indirectly.

- Directly: increasing sales (ads, conversion rates), cutting costs
- Indirectly: increasing customer satisfaction, increasing time spent on a website

Non-profits are exceptions: lots of exciting applications of AI for social good

- environment (climate change, deforestation, flood risk, etc.)
- public health
- education (intelligent tutoring system, personalized learning)

ML to business performance can be confusing

An ML model that gives customers more personalized solutions can either:

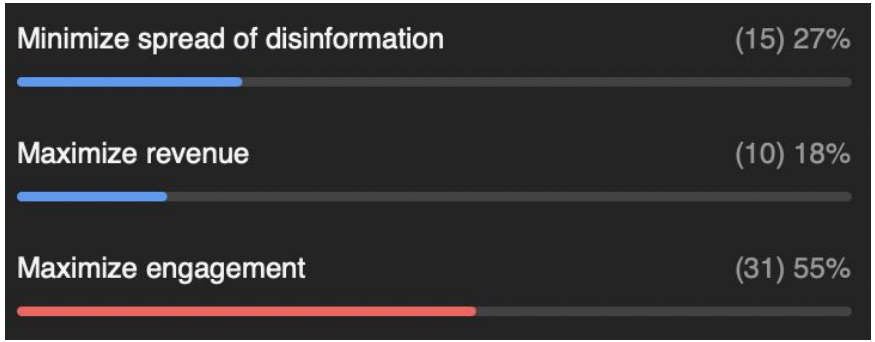
- make them happier which makes them spend more money
- solve their problems faster which makes them spend less money

Example

Possible goals when building a ranking system for newsfeed?

1. minimize the spread of misinformation
2. maximize revenue from sponsored content
3. maximize engagement

Zoom poll: which goal would you choose?



Side note: ethics of maximizing engagement

> Several current and former YouTube employees, who would speak only on the condition of anonymity because they had signed confidentiality agreements, said company leaders were obsessed with increasing engagement during those years. The executives, the people said, rarely considered whether the company's algorithms were fueling the spread of extreme and hateful political content.

Employee raises at Facebook depend on engagement, and newly leaked private Zuckerberg recordings show the Groups algorithm prioritizes enragement.

In data terms, anti-vaxx groups and QAnon hysteria are going to get far better engagement than your average drag queen or 'Vote Yes on Proposition Z' groups. Moreover, the group recommendations tool prioritizes the angriest and most out-to-lunch groups, because those tend to get more clicks when they appear in the recommended field.

Facebook Employee Raises Powered by 'Really Dangerous' Algorithm That Favors Angry Posts (SFist, 2019)
The Making of a YouTube Radical (NYT, 2019)

Objectives

Goals	Objectives
General purpose of a project	Specific steps on how to realize that purpose

Example: ranking system for newsfeed

Goals	**Objectives**
General purpose of a project	Specific steps on how to realize that purpose
Maximize users' engagement	1. Filter out spam 2. Filter out NSFW content 3. Rank posts by engagement: how likely users will click on it

Example: ranking system for wholesome newsfeed

Goals	Objectives
General purpose of a project	Specific steps on how to realize that purpose
Maximize users' engagement while **minimizing the spread of extreme views** and **misinformation**	1. Filter out spam 2. Filter out NSFW content 3. Filter out misinformation 4. Rank posts by quality 5. Rank posts by engagement: how likely users will click on it

Example: ranking system for wholesome newsfeed

Goals	**Objectives**
General purpose of a project	Specific steps on how to realize that purpose
Maximize users' engagement while **minimizing the spread of extreme views** and **misinformation**	1. Filter out spam 2. Filter out NSFW content 3. Filter out misinformation 4. Rank posts by quality 5. Rank posts by how likely users will click on it

How to combine these two objectives?

Multiple objective optimization (MOO)

- Rank posts by quality
  - Predict posts' quality
  - Minimize **quality_loss:** difference between predicted quality and true quality

- Rank posts by how likely users will click on it
  - Predict posts' engagement
  - Minimize **engagement_loss**: difference between predicted clicks and true clicks

MOO: one model with combined loss

- Rank posts by quality
  - Predict posts' quality
  - Minimize **quality_loss:** difference between predicted quality and true quality

- Rank posts by how likely users will click on it
  - Predict posts' engagement
  - Minimize **engagement_loss**: difference between predicted clicks and true clicks

$$\text{loss} = \alpha\ \text{quality\_loss} + \beta\ \text{engagement\_loss}$$

Train one model to minimize this combined loss
Tune α and β to meet your need

Side note 1: check out Pareto optimization if you want to learn about how to choose α and β

MOO: one model with combined loss

- Rank posts by quality
  - Predict posts' quality
  - Minimize **quality_loss:** difference between predicted quality and true quality

- Rank posts by how likely users will click on it
  - Predict posts' engagement
  - Minimize **engagement_loss**: difference between predicted clicks and true clicks

$$\text{loss} = \alpha \text{ quality\_loss} + \beta \text{ engagement\_loss}$$

Train one model to minimize this combined loss

Side note 2: this is quite common, e.g. style transfer

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

A Neural Algorithm of Artistic Style (Gatys et al, 2017)

MOO: one model with combined loss

- Rank posts by quality
  - Predict posts' quality
  - Minimize **quality_loss:** difference between predicted quality and true quality

- Rank posts by how likely users will click on it
  - Predict posts' engagement
  - Minimize **engagement_loss**: difference between predicted clicks and true clicks

$$\text{loss} = \alpha\ \text{quality\_loss} + \beta\ \text{engagement\_loss}$$

Train one model to minimize this combined loss

⚠️ Every time you want to tweak α and β, you have to retrain your model! ⚠️

MOO: combine different models

- Rank posts by quality
  - Predict posts' quality
  - Minimize **quality_loss:** difference between predicted quality and true quality

- Rank posts by how likely users will click on it
  - Predict posts' engagement
  - Minimize **engagement_loss**: difference between predicted clicks and true clicks

M_q: optimizes **quality_loss**
M_e: optimizes **engagement_loss**

Rank posts by α **M_q(post)** + β **M_e(post)**

Now you can tweak α and β without retraining models

Decouple different objectives

- Easier for training:
  - Optimizing for one objective is easier than optimizing for multiple objectives
- Easier to tweak your system:
  - E.g. α % model optimized for quality + β % model optimized for engagement
- Easier for maintenance:
  - Different objectives might need different maintenance schedules
    - Spamming techniques evolve much faster than the way post quality is perceived
    - Spam filtering systems need updates more frequently than quality ranking systems

Constraints: time & budget

- Time
  - Rule of thumb: 20% time to get initial working system, 80% on iterative development
- Budget
  - Data, resources, talent

Possible time/budget tradeoffs:

- Use more/more powerful machines to get things done faster
- Hire more people to label more data/run more experiments
- Buy existing solutions

Constraints: performance

- Baselines
  - Existing solutions, simple solutions, human experts, competitors solutions, etc.

Constraints: performance

- Baselines
- Usefulness threshold
  - Self-driving needs human-level performance. Predictive texting doesn’t.

Constraints: performance

- Baselines
- Usefulness threshold
- False negatives vs. false positives
  - Covid screening: no false negative (patients with covid shouldn’t be classified as no covid)
  - Fingerprint unlocking: no false positive (unauthorized people shouldn’t be given access)

Constraints: performance

- Baselines
- Usefulness threshold
- False negatives vs. false positives
- Interpretability
  - Does it need to be interpretable? If yes, to whom?

Constraints: performance

- Baselines
- Usefulness threshold
- False negatives vs. false positives
- Interpretability
- Confidence measurement (how confident it is about a prediction)
  - Does it need confidence measurement?
  - Is there a confidence threshold? What to do with predictions below that threshold—discard it, loop in humans, or ask for more information from users?

Constraints: performance

- Baselines
  - Existing solutions, simple solutions, human experts, competitors solutions, etc.
- Usefulness threshold
  - Self-driving needs human-level performance. Predictive texting doesn't.
- False negatives vs. false positives
  - Covid screening: no false negative (patients with covid shouldn't be classified as no covid)
  - Fingerprint unlocking: no false positive (unauthorized people shouldn't be given access)
- Interpretability
  - Does it need to be interpretable? If yes, to whom?
- Confidence measurement (how confident it is about a prediction)
  - Does it need confidence measurement?
  - Is there a confidence threshold? What to do with predictions below that threshold—discard it, loop in humans, or ask for more information from users?

Constraints: privacy

- Privacy requirements for annotation, storage, third-party solutions, cloud services, regulations
  - Can data be shipped outside organizations for annotation?
  - Can the system be connected to the Internet?
  - How long can you keep users data?

Constraints: technical constraints

- Competitors
- Legacy systems

Evaluation

- How to evaluate models during training?
- How to evaluate models in serving, when there’s no hand-labeled ground truths?

5. Breakout exercise

10 mins - group of 4

Each question on Quora often gets many different answers. How would you build a system to rank the answers?

Some aspects to consider:

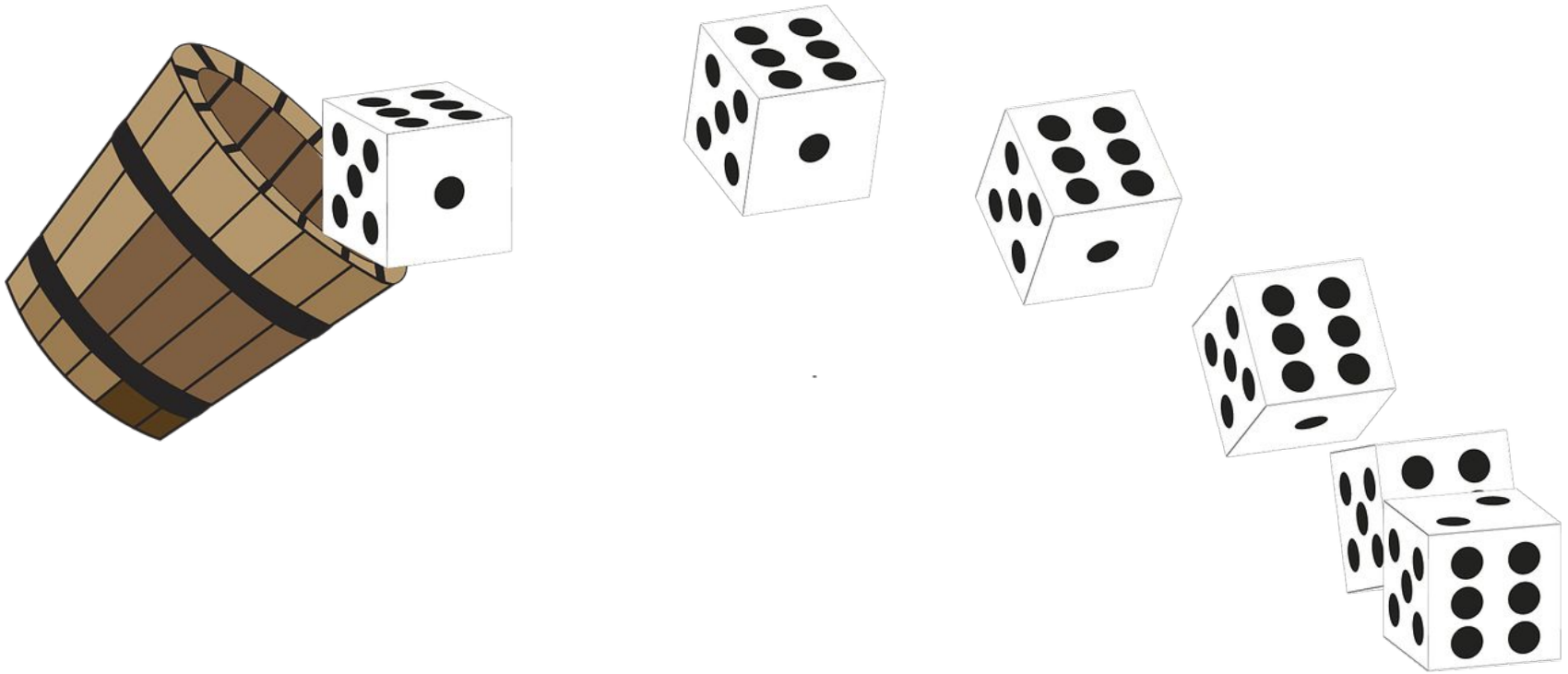
1. Goals and objectives?
2. Constraints
3. What should labels be?
4. What features to include?
5. How to evaluate models, during training & in production?
6. How to collect data?

OK to google
Don’t forget to introduce yourself to your teammates!

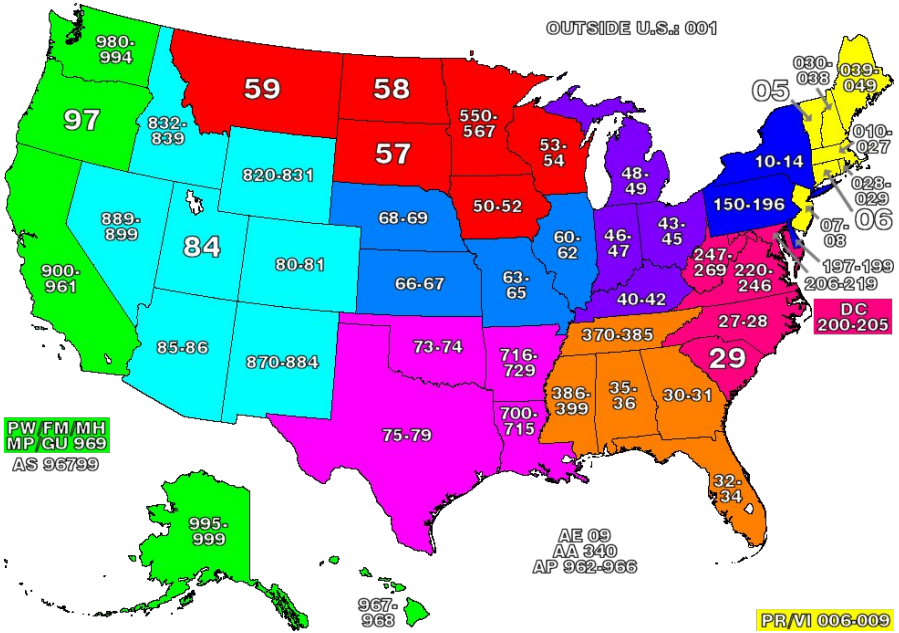
6. When to use ML

Machine learning is an approach to learn complex patterns from existing data and use these patterns to make predictions on unseen data.

Patterns: there must be patterns to learn



Complex: the patterns are complex

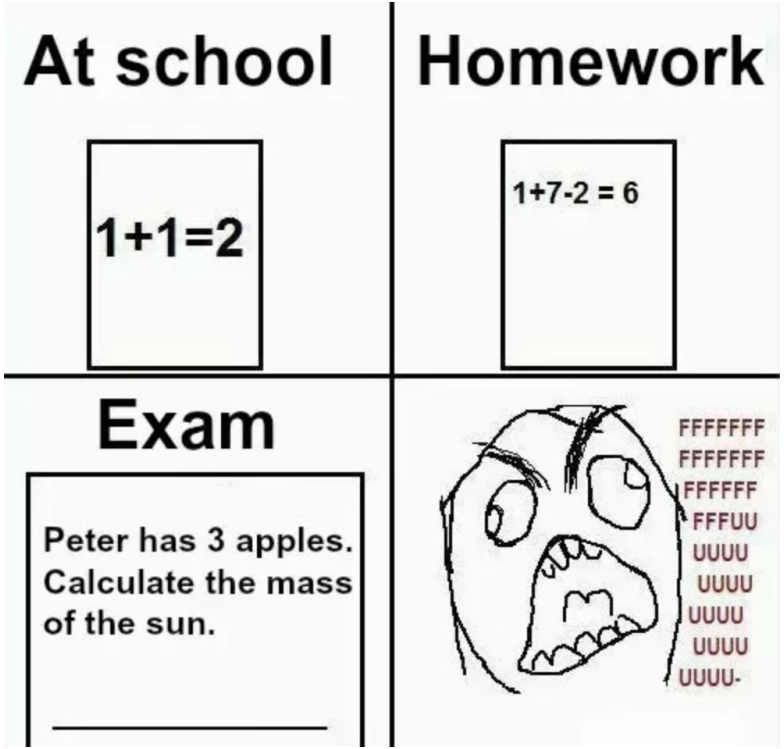


Simple

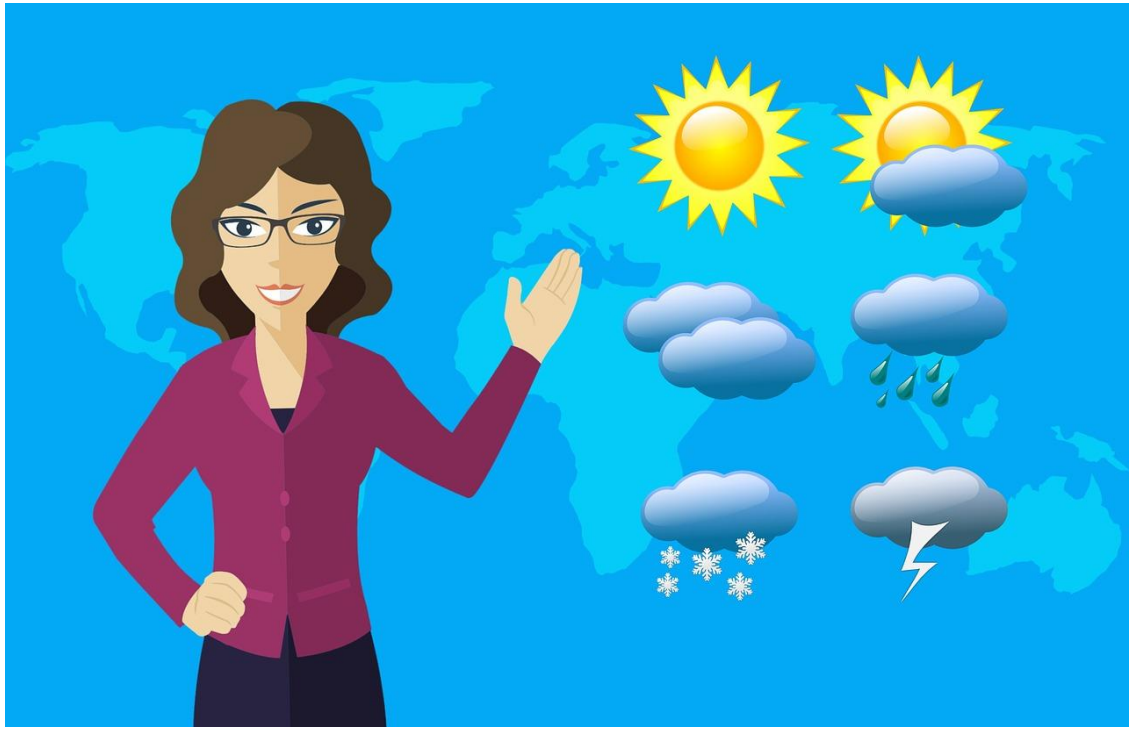
Complex

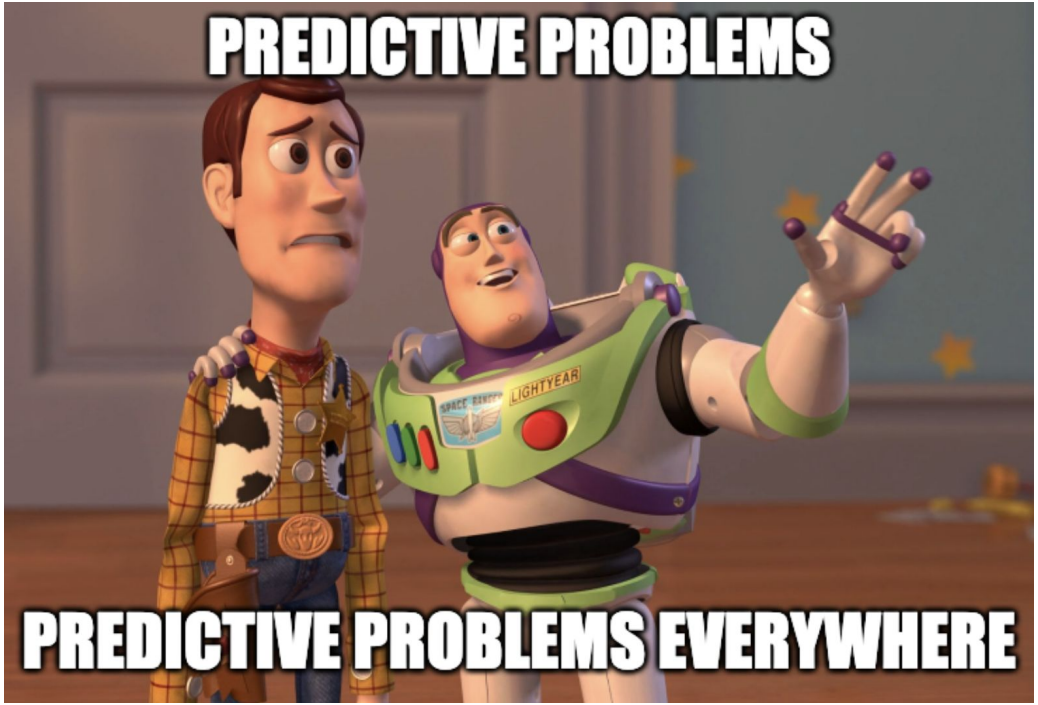
Existing data: it's possible to collect

Unseen data: the patterns generalize



Predictions: it's a predictive problem





A question can be turned into a predictive problem by asking:
What would the answer/solution look like?

When not to use ML

- It’s unethical.
- Simpler solutions do the trick.
- It’s impossible to get the right data.
- One single prediction error can cause devastating consequences.
- Every single decision the system makes must be explainable.
- It’s not cost-effective.

When not to use ML

- It’s unethical.
- Simpler solutions do the trick.
- It’s impossible to get the right data.
- One single prediction error can cause devastating consequences.
- Every single decision the system makes must be explainable.
- It’s not cost-effective.

Caution: it might not be cost-effective now but might be in the future

If ML isn’t the solution, it can be part of it

- Break your problem into smaller problems. ML might be able to solve some of them.
  - If a chatbot can’t answer all customers’ queries, build an ML model to predict whether a query matches one of the frequently asked questions.
  - Human-in-the-loop

7. Four phases of ML adoption

Phase 1: Before ML

“If you think that machine learning will give you a 100% boost, then a heuristic will get you 50% of the way there.”

Martin Zinkevich, Google

https://newsfeed.org/what-mark-zuckerbergs-news-feed-looked-like-in-2006/

Phase 2: Simplest ML models

Start with a simple model that allows visibility into its working to:

- validate hypothesis
- validate pipeline

Phase 3: Optimizing simple models

- Different objective functions
- Feature engineering
- More data
- Ensembling

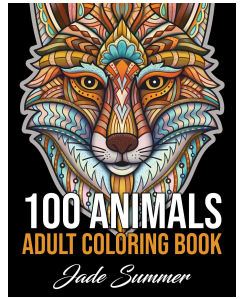
Phase 4: Complex ML models


STAR WARS
THE MANDALORIAN
500
PUZZLE
ROMPECABEZAS


FERRERO
COLLECTION
24


Premium Scalp Massager


SMACKER


100 ANIMALS
ADULT COLORING BOOK
Jade Summer


Bear


COMO SI TE A Llama

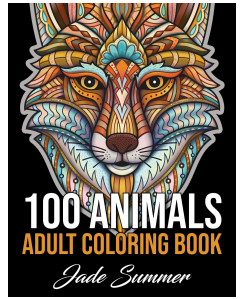

Bob Ross

4

2

8

6



10

John got a mug warmer!

7

1

3

5

9

Machine Learning Systems Design

Next class: Data engineering