FSDL 2022

ML products & orgs

Josh Tobin

September 26, 2022

Building ML Products

Building any product is hard...

- Hiring great people
- Managing and developing those people
- Managing your team's output and making sure your vectors are aligned
- Making good long-term technical choices & managing technical debt
- Managing expectations from leadership
- Defining and communicating requirements with stakeholders

... And ML adds complexity

- ML talent is expensive and scarce
- ML teams have a diverse set of roles
- Projects have unclear timelines and high uncertainty
- The field is moving fast and ML is the "high-interest credit card of technical debt"
- Leadership often doesn't understand AI
- ML products fail in ways that are hard for laypeople to understand

Module overview

Roles	ML-related roles and the skills they require
Hiring	How to hire ML engineers. How to get hired.
Orgs	How ML teams are organized and how they fit into the broader organization
Managing	How to manage a ML team & ML product
Design	Design considerations for ML products

Module overview

Roles	ML-related roles and the skills they require
Hiring	How to hire ML engineers. How to get hired.
Orgs	How ML teams are organized and how they fit into the broader organization
Managing	How to manage a ML team & ML product
Design	Design considerations for ML products

Most common ML roles

- ML product manager
- MLOps / ML Platform
- ML engineer
- ML researcher / ML scientist
- Data scientist

Most common ML roles

- ML product manager
- MLOps / ML Platform / ML Infra
- ML engineer
- ML researcher / ML scientist
- Data scientist

What's the difference?

Breakdown of job function by role

Role	Job Function	Work product	Commonly used tools
ML product manager	Work with ML team, business, users, data owners to prioritize & execute projects	Design docs, wireframes, work plans	Jira, etc

Breakdown of job function by role

Role	Job Function	Work product	Commonly used tools
ML product manager	Work with ML team, business, users, data owners to prioritize & execute projects	Design docs, wireframes, work plans	Jira, etc
MLOps / ML platform	Build the infrastructure to make models easier to deploy, more scalable, etc	ML infrastructure	AWS, Kafka, ML tooling vendors, etc.

Breakdown of job function by role

Role	Job Function	Work product	Commonly used tools
ML product manager	Work with ML team, business, users, data owners to prioritize & execute projects	Design docs, wireframes, work plans	Jira, etc
MLOps / ML platform	Build the infrastructure to make models easier to deploy, more scalable, etc	ML infrastructure	AWS, Kafka, ML tooling vendors, etc.
ML engineer	Train, deploy, & maintain prediction models	Prediction system running on real data in production	Tensorflow, Docker

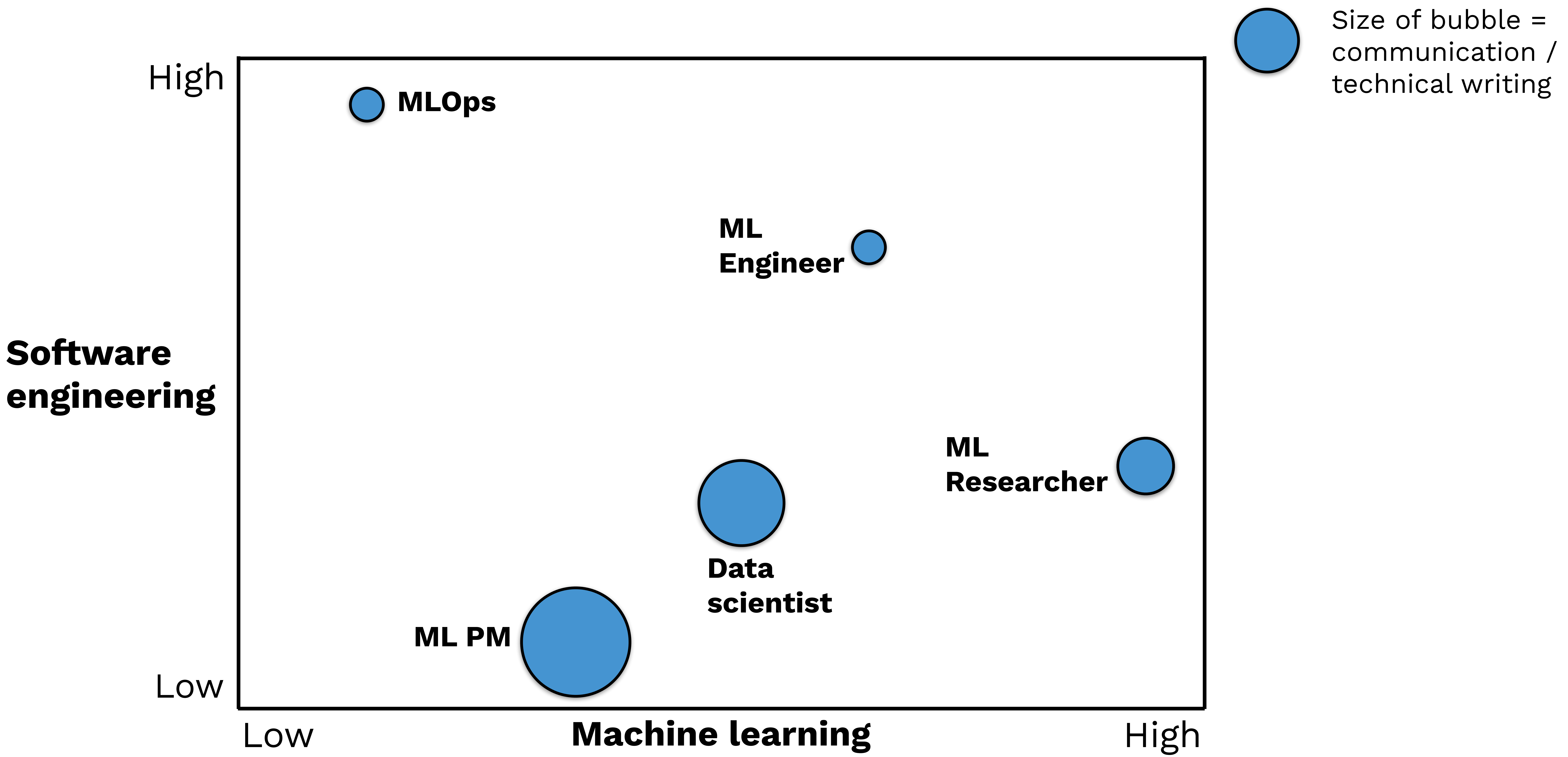
Breakdown of job function by role

Role	Job Function	Work product	Commonly used tools
ML product manager	Work with ML team, business, users, data owners to prioritize & execute projects	Design docs, wireframes, work plans	Jira, etc
MLOps / ML platform	Build the infrastructure to make models easier to deploy, more scalable, etc	ML infrastructure	AWS, Kafka, ML tooling vendors, etc.
ML engineer	Train, deploy, & maintain prediction models	Prediction system running on real data in production	Tensorflow, Docker
ML researcher	Train prediction models (often forward looking or not production-critical)	Prediction model & report describing it	Tensorflow, pytorch, Jupyter

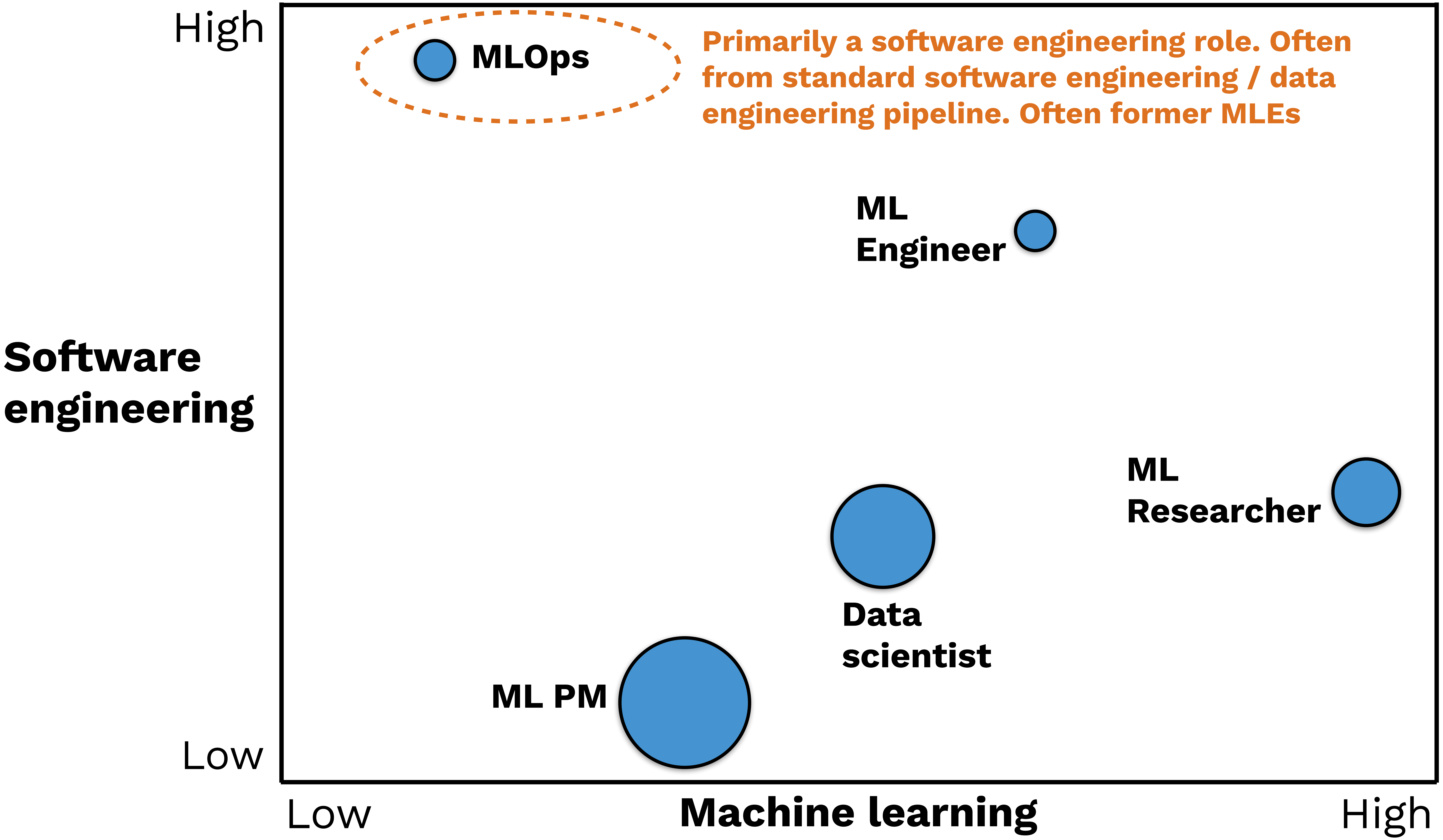
Breakdown of job function by role

Role	Job Function	Work product	Commonly used tools
ML product manager	Work with ML team, business, users, data owners to prioritize & execute projects	Design docs, wireframes, work plans	Jira, etc
MLOps / ML platform	Build the infrastructure to make models easier to deploy, more scalable, etc	ML infrastructure	AWS, Kafka, ML tooling vendors, etc.
ML engineer	Train, deploy, & maintain prediction models	Prediction system running on real data in production	Tensorflow, Docker
ML researcher	Train prediction models (often forward looking or not production-critical)	Prediction model & report describing it	Tensorflow, pytorch, Jupyter
Data scientist	Blanket term used to describe all of the above. In some orgs, means answering business questions using analytics	Prediction model or report	SQL, Excel, Jupyter, Pandas, SKLearn, Tensorflow

What skills are needed for the roles?

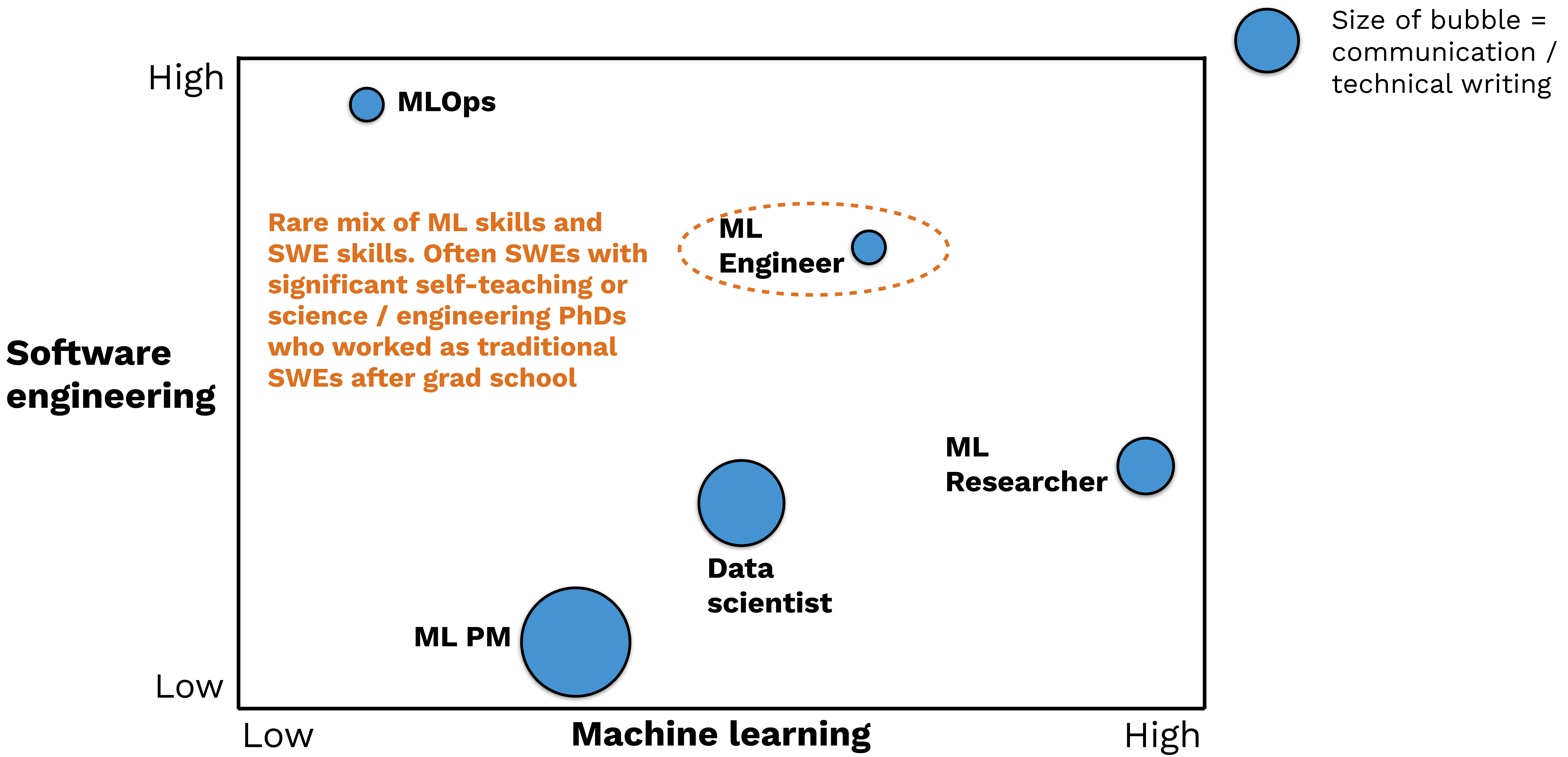


What skills are needed for the roles?

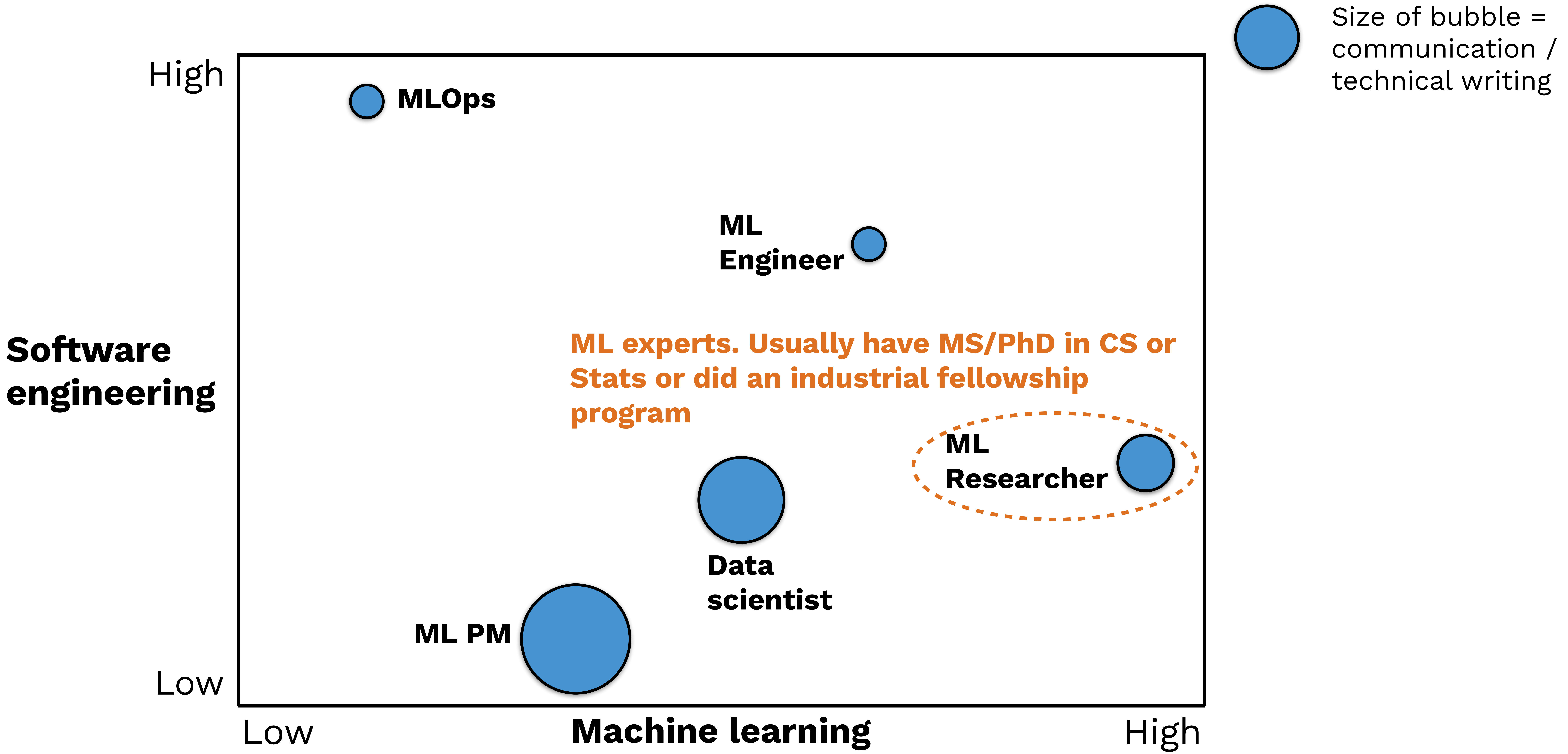


Size of bubble = communication / technical writing

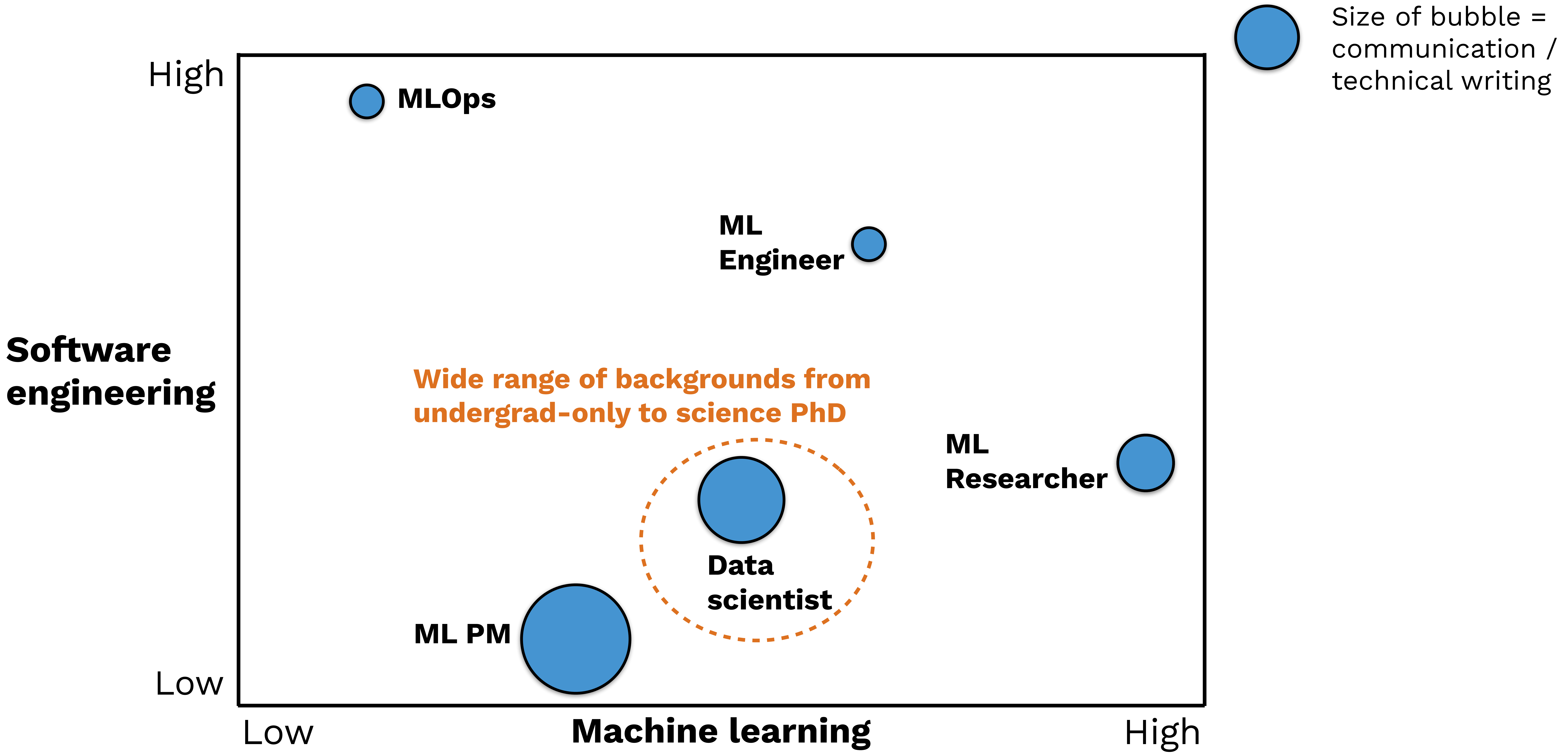
What skills are needed for the roles?



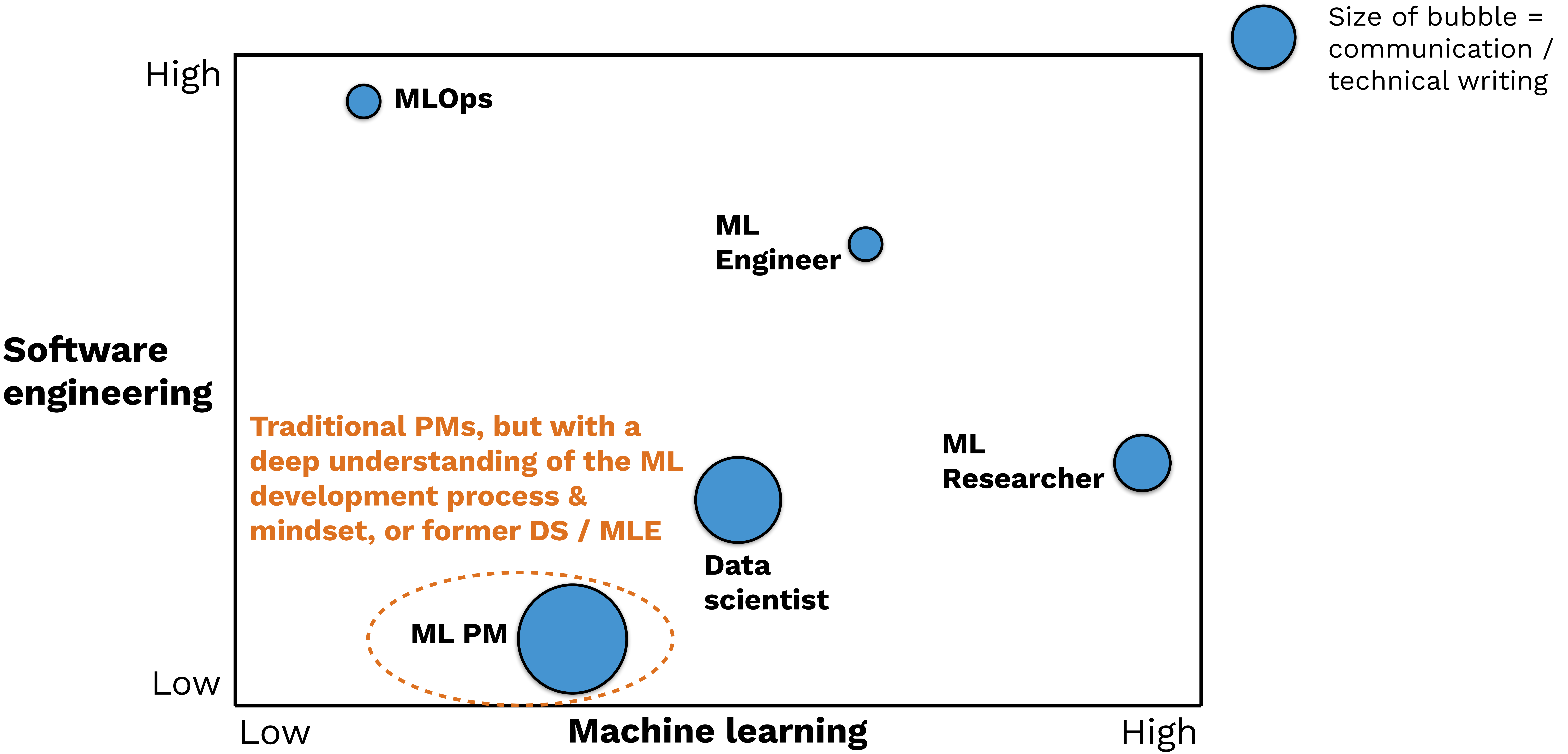
What skills are needed for the roles?



What skills are needed for the roles?



What skills are needed for the roles?



The two kinds of ML engineers

- Task MLEs
  - Responsible for maintaining specific ML pipeline(s)
- Platform MLEs
  - Help task MLEs automate tedious parts of their jobs
  - (ML platform / MLOps in our parlance)

https://www.shreya-shankar.com/phd-year-one/

Module overview

Roles	• ML-related roles and the skills they require
Hiring	• How to hire ML engineers. How to get hired.
Orgs	• How ML teams are organized and how they fit into the broader organization
Managing	• How to manage a ML team & ML product
Design	• Design considerations for ML products

Hiring for ML - outline

- The AI Talent Gap
- Sourcing
- Interviewing
- Finding a job

Hiring for ML - outline

- **The AI Talent Gap**
- Sourcing
- Interviewing
- Finding a job

Four years into FSDL, the AI talent gap persists

The AI talent gap in 2022

NEWSLETTERS · EYE ON A.I.

Here’s one way to deal with the A.I. talent shortage

BY **JONATHAN VANIAN**

April 26, 2022 at 1:56 PM PDT

INNOVATION

Steps Enterprises Can Take To Build A Dependable AI Talent Pool

Sameer Maskey Forbes Councils Member

Forbes Technology Council

COUNCIL POST | Membership (Fee-Based)

Feb 17, 2022, 10:30am EST

The AI talent gap in 2018

Fierce competition for AI talent

“Everyone agrees that the competition to hire people who know how to build artificial intelligence systems is intense. It’s turned once-staid academic conferences into frenzied meet markets for corporate recruiters and driven the salaries of the top researchers to seven-figures.”

(Bloomberg)

Hiring for ML - outline

- The AI Talent Gap
- **Sourcing**
- Interviewing
- Finding a job

Most common ML roles

- ML product manager
- DevOps
- Data engineer
- ML engineer
- ML researcher / ML scientist
- Data scientist

Most common ML roles

- ML product manager
- ML platform / MLOps
- ML engineer
- ML researcher / ML scientist
- Data scientist

Experience working with production ML systems is key

Most common ML roles

- ML product manager
- ML platform / MLOps
- ML engineer
- ML researcher / ML scientist
- Data scientist

Our focus

How to hire MLEs - the wrong way

- Job Description (Unicorn Machine Learning Engineer)
  - Duties
    - Keep up with the state of the art
    - Implement models from scratch
    - Deep understanding of mathematics & ability to come up with new models
    - Build tooling & infrastructure for the ML team
    - Build data pipelines for the ML team
    - Deploy & monitor models into production
  - Requirements
    - PhD
    - At least 4 years tensorflow experience
    - At least 4 years as a software engineer
    - Publications in top ML conference
    - Experience building large-scale distributed systems

How to hire MLEs - the right way

- Hire for software engineering skills, interest in ML, and desire to learn. Train to do ML.
- Go more junior. Most undergrad computer science students graduate with ML experience.
- Be more specific about what you need. Not every ML engineer needs to do DevOps.

How to hire MLRs

- Look for quality of publications, not quantity (e.g., originality of ideas, quality of execution)
- Look for researchers with an eye for working on important problems (many researchers focus on trendy problems without considering why they matter)
- Look for researchers with experience outside of academia
- Consider hiring talented people from adjacent fields (physics, statistics, math)
- Consider hiring people without PhDs (e.g., talented undergraduate / masters students, graduates of Google/Facebook/OpenAI fellowship programs, dedicated self-studiers)

How to find MLE/MLR candidates

- Standard sources: LinkedIn, recruiters, on-campus recruiting, etc
- Monitor arXiv and top conferences and flag first authors of papers you like
- Look for good reimplementations of papers you like
- Attend ML research conferences (NeurIPS, ICLR, ICML)

How to attract MLR / MLE candidates

What do machine learning practitioners want?

- Work with cutting edge tools & techniques
- Build skills / knowledge in an exciting field
- Work with excellent people
- Work on interesting datasets
- Do work that matters

How to make your company stand out?

- Work on research-oriented projects. Publicize them. Invest in tooling for your team & empower employees to try new tools.
- Build team culture around learning (reading groups, learning days, professional development budget, conference budget)
- Hire high-profile people. Help your best people build their profile through publishing blogs & papers.
- Sell the uniqueness of your dataset in recruiting materials.
- Sell the mission of your company and potential impact of machine learning on that mission. Work on projects that have a tangible impact today.

Hiring for ML - outline

- The AI Talent Gap
- Sourcing
- **Interviewing**
- Finding a job

What to test in an ML interview?

- Hire for strengths
- Meet a minimum bar for everything else

What to test in an ML interview?

- Validate your hypotheses of candidate's strengths
  - Researchers: make sure they can think creatively about new ML problems, probe how thoughtful they were about previous projects
  - Engineers: make sure they are great generalist SWEs
- Make sure candidates meet a minimum bar on weaker areas
  - Researchers: test SWE knowledge and ability to write good code
  - SWEs: test ML knowledge

What happens in a ML interview?

- Much less well-defined than software engineering interviews
- Common types of assessments:
  - Background & culture fit
  - Whiteboard coding (similar to SWE interviews)
  - Pair coding (similar to SWE interviews)
  - Pair debugging (often ML-specific code)
  - Math puzzles (e.g., involving linear algebra)
  - Take-home ML project
  - Applied ML (e.g., explain how you'd solve this problem with ML)
  - Previous ML projects (e.g., probing on what you tried, why things did / didn't work)
  - ML theory (e.g., bias-variance tradeoff, overfitting, underfitting, understanding of specific algorithms)

Introduction to Machine Learning Interviews Book

https://huyenchip.com/ml-interviews-book/

Hiring for ML - outline

- The AI Talent Gap
- Sourcing
- Interviewing
- **Finding a job**

Where to look for a ML job?

- Standard sources: LinkedIn, recruiters, on-campus recruiting, etc
- ML research conferences (NeurIPS, ICLR, ICML)
- Apply directly (remember, there’s a talent gap!)

How to stand out for ML roles?

More impressive

- Exhibit interest in ML (e.g., conference attendance, online courses taken)
- Build software engineering skills (e.g., work at a well-known software company)
- Show you have broad knowledge of ML (e.g., write blog posts synthesizing a research area)
- Demonstrate ability to get ML projects done (e.g., create side projects, re-implement papers)
- Prove you can think creatively in ML (e.g., win Kaggle competitions, publish papers)

Module overview

Roles
- ML-related roles and the skills they require

Hiring
- How to hire ML engineers. How to get hired.

Orgs
- How ML teams are organized and how they fit into the broader organization

Managing
- How to manage a ML team & ML product

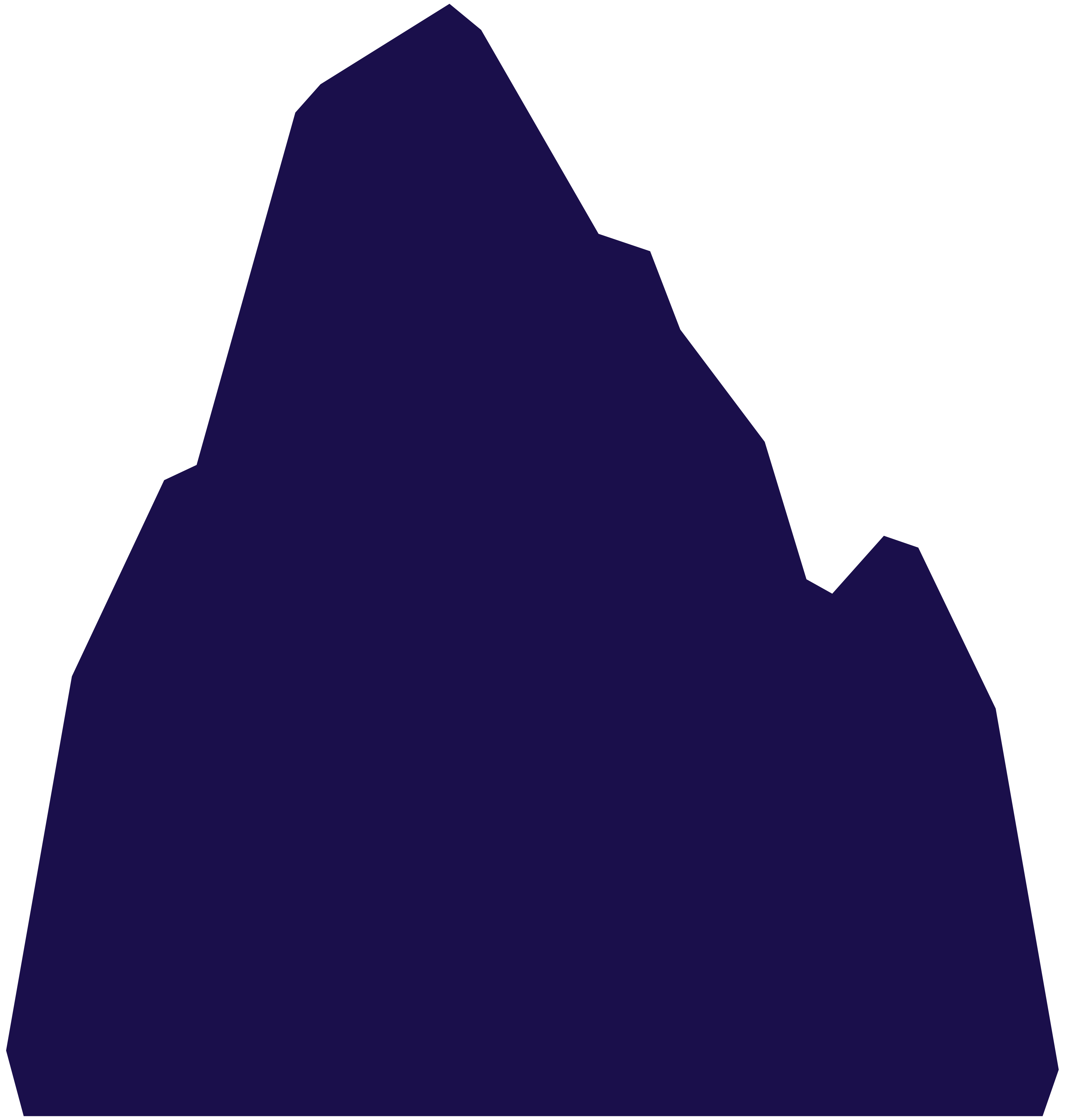
Design
- Design considerations for ML products

ML org structures - lessons learned

- No consensus yet on the right way to structure a ML team
- This lecture: taxonomy of best practices for different organizational maturity levels

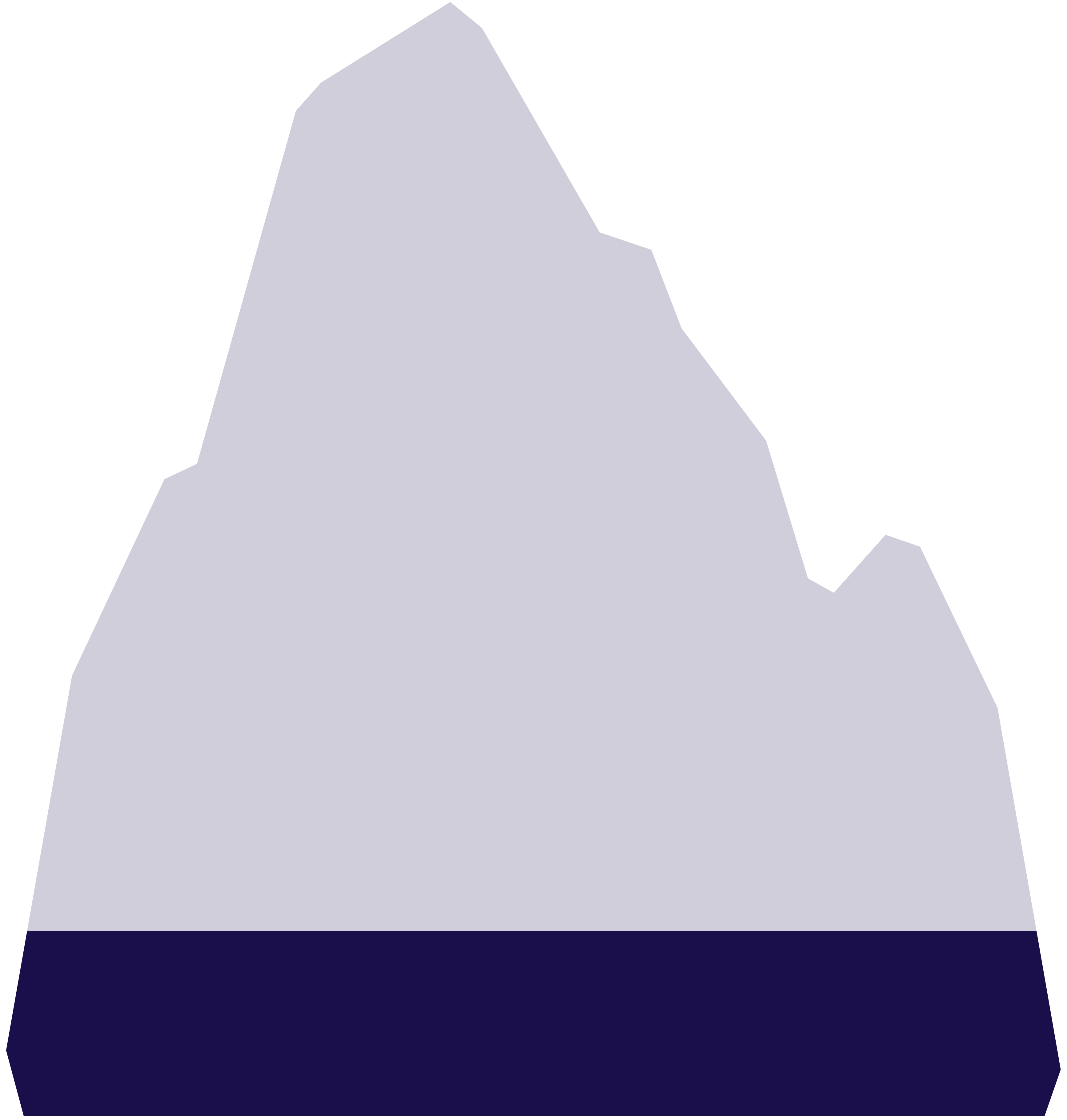
ML organization archetypes

The ML Organization Mountain



ML organization archetypes

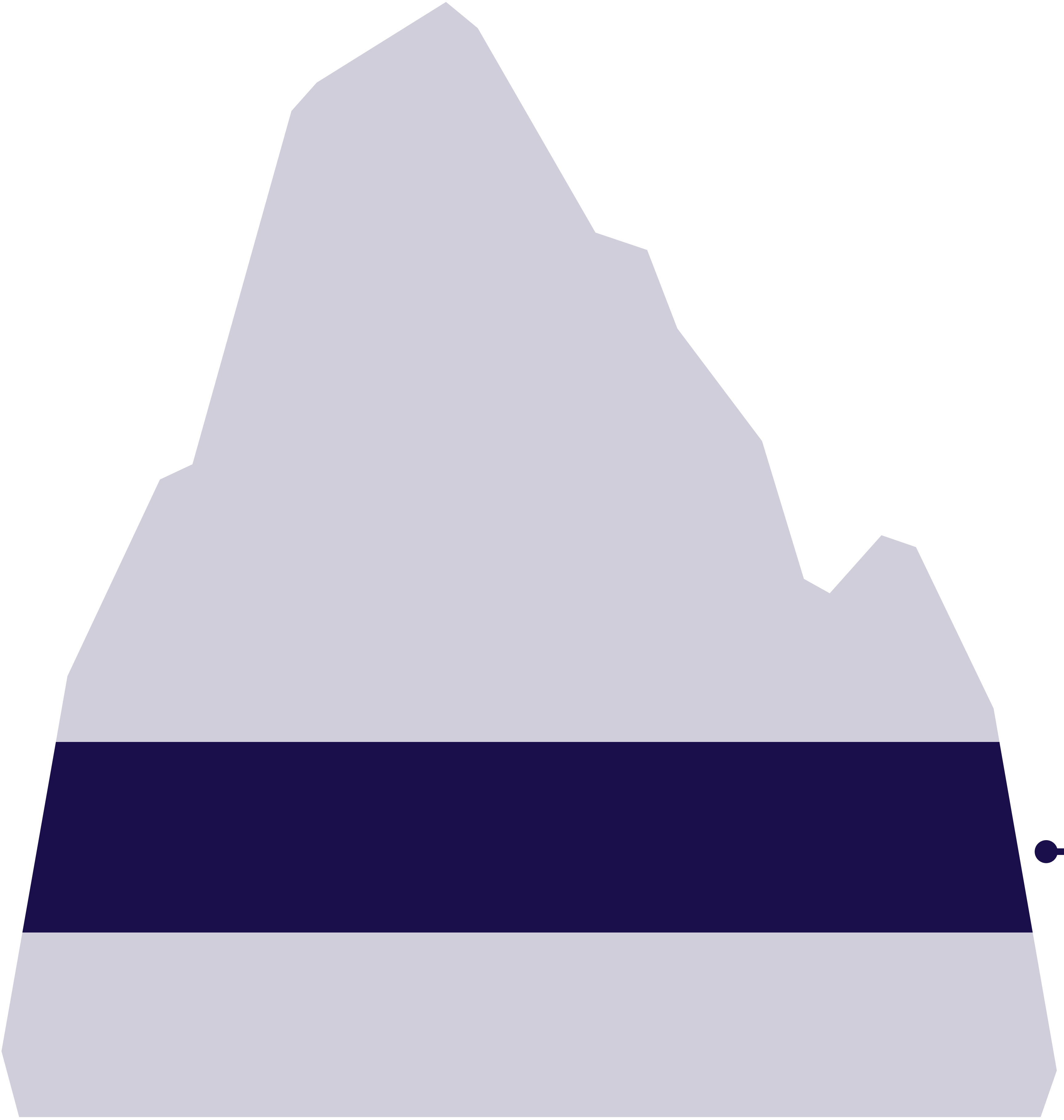
The ML Organization Mountain



Nascent / Ad-Hoc ML	
What it looks like	• No one is doing ML, or ML is done on an ad-hoc basis • Little ML expertise in-house
Example organizations	• Most small-medium businesses • Less technology-forward large companies (education, logistics, etc)
Advantages	• Often low-hanging fruit for ML
Dis-advantages	• Little support for ML projects, difficult to hire and retain good talent

ML organization archetypes

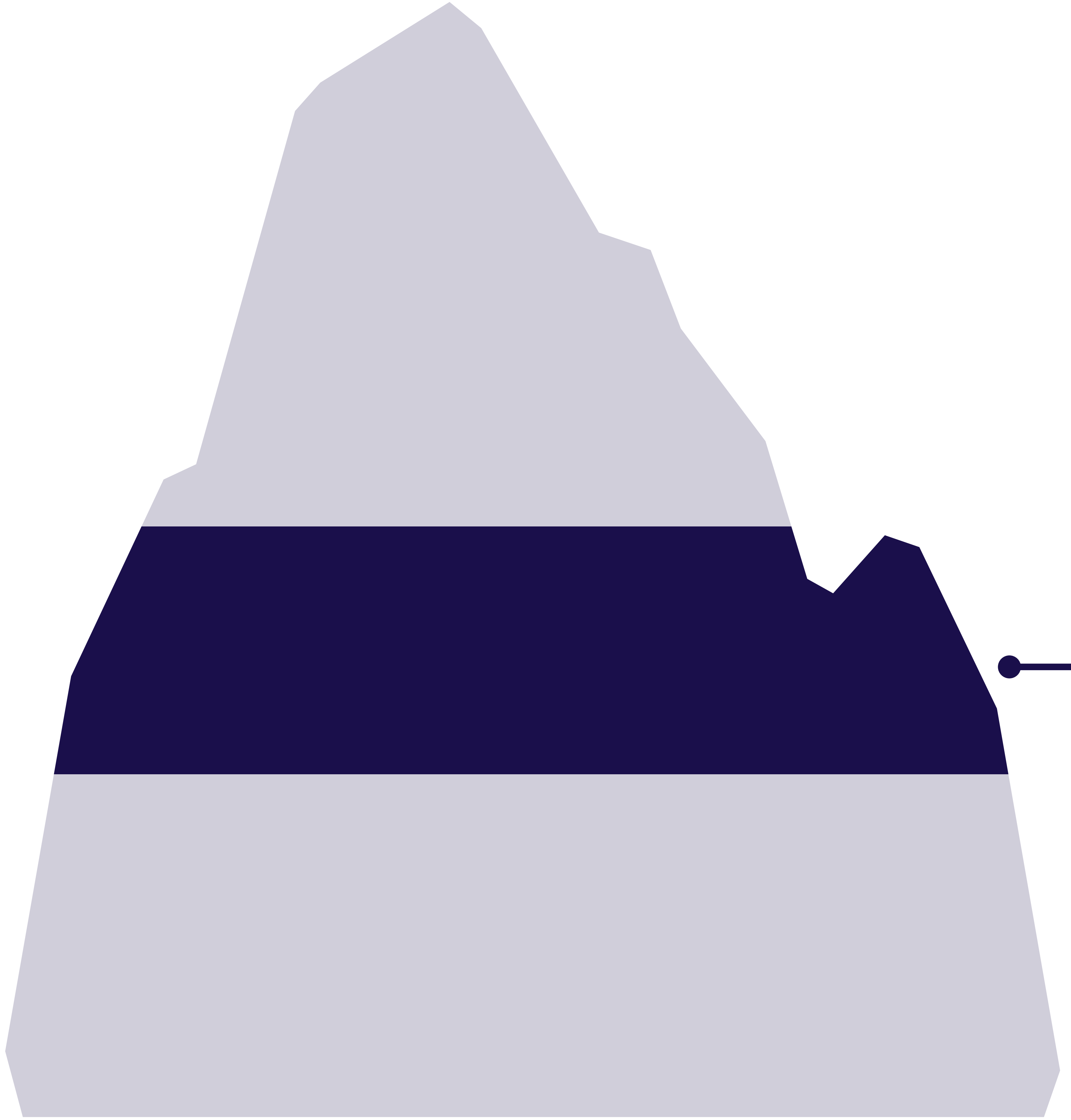
The ML Organization Mountain



ML R&D	
What it looks like	• ML efforts are centered in the R&D arm of the organization • Often hire researchers / PhDs & write papers
Example organizations	• Larger Oil & gas, manufacturing, telecom companies
Advantages	• Often can hire experienced researchers • Can work on long-term business priorities & big wins
Dis-advantages	• Difficult to get data • Rarely translates into actual business value, so usually the amount of investment remains small

ML organization archetypes

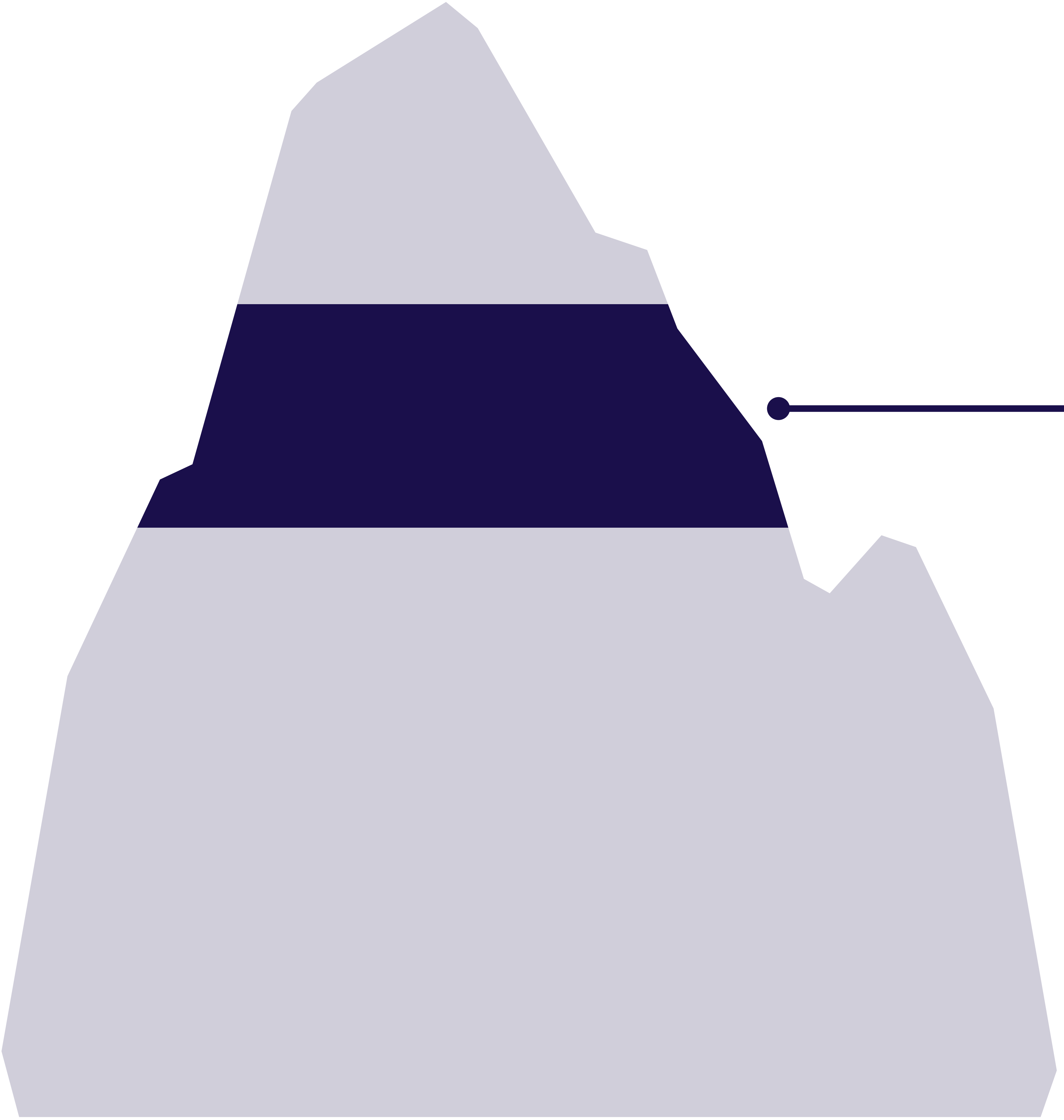
The ML Organization Mountain



ML embedded into business / product teams	
What it looks like	• Certain product teams or business units have ML expertise along-side their software or analytics talent • ML reports up to the team's engineering lead or tech lead
Example organizations	• Software / technology companies • Financial services companies
Advantages	• ML improvements are likely to lead to business value • Tight feedback cycle between idea and product improvement
Dis-advantages	• Hard to hire and develop top talent • Access to resources (data / compute) can lag • ML project cycles conflict with engineering mgmt • Long-term projects can be hard to justify

ML organization archetypes

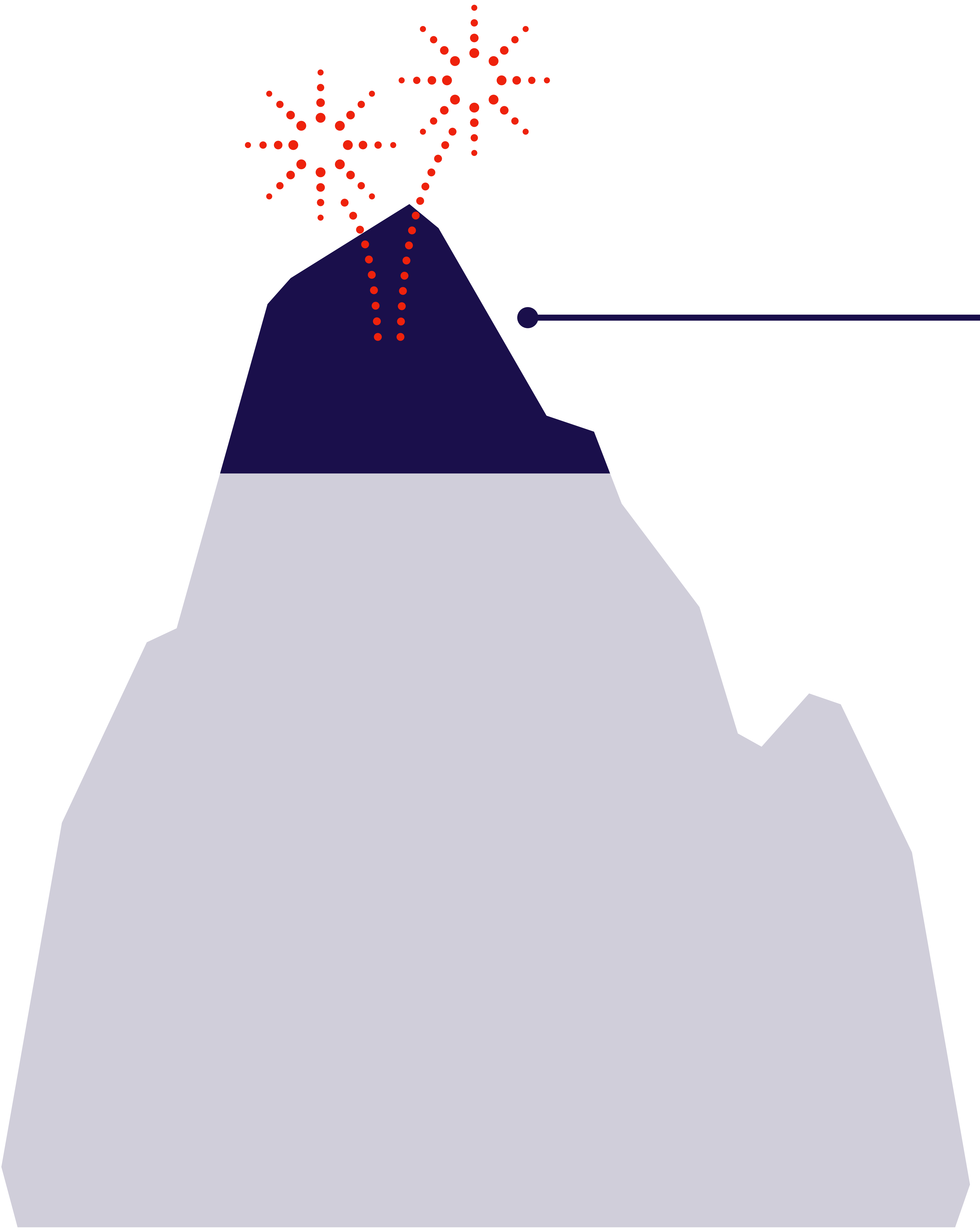
The ML Organization Mountain



Independent ML Function

What it looks like	• ML division reporting to senior leadership (often CEO) • ML PMs work with MLRs, MLEs, and customers to build ML into products • Teams sometimes publish long-term research
Example organizations	• Large financial services companies
Advantages	• Talent density allows to hire & train top practitioners • Senior leaders can marshal data / compute resources • Can invest in tooling, practices, and culture around ML development
Dis-advantages	• Model handoffs to lines of business can be challenging - users need to buy-in and be educated on model use • Feedback cycles can be slow

ML organization archetypes



ML-First Organizations	
What it looks like	• CEO buy-in • ML division working on challenging, long-term projects • ML expertise in every line of business focusing on quick wins and working with central ML division
Example organizations	• Large tech companies • ML-focused startups
Advantages	• Best data access: data thinking permeates the org • Recruiting: ML team works on hardest problems • Easiest deployment: product teams understand ML
Dis-advantages	• Hard to implement • Challenging & expensive to recruit enough talent • Culturally difficult to embed ML thinking everywhere

ML team structures - design choices

	Key questions
Software engineering vs research	• To what extent is the ML team responsible for building or integrating with software? • How important are SWE skills on the team?
Data ownership	• How much control does the ML team have over data collection, warehousing, labeling, and pipelining?
Model ownership	• Is the ML team responsible for deploying models into production? • Who maintains deployed models?

ML team structures - design choices

	ML R&D
Software engineering vs research	• Research prioritized over SWE skills • Researcher-SWE collaboration lacking
Data ownership	• ML team has no control over data • ML team typically will not have data engineering component
Model ownership	• Models are rarely deployed into production

ML team structures - design choices

	ML R&D	Embedded ML
Software engineering vs research	• Research prioritized over SWE skills • Researcher-SWE collaboration lacking	• SWE skills prioritized over research skills • Often, all researchers need strong SWE as everyone expected to deploy
Data ownership	• ML team has no control over data • ML team typically will not have data engineering component	• ML team generally does not own data production / mgmt • Work with data engineers to build pipelines
Model ownership	• Models are rarely deployed into production	• ML engineers own the models that they deploy into production

ML team structures - design choices

	ML R&D	Embedded ML	ML Function
Software engineering vs research	• Research prioritized over SWE skills • Researcher-SWE collaboration lacking	• SWE skills prioritized over research skills • Often, all researchers need strong SWE as everyone expected to deploy	• Each team has a strong mix of SWE and research skills • SWE and researchers work closely together within team
Data ownership	• ML team has no control over data • ML team typically will not have data engineering component	• ML team generally does not own data production / mgmt • Work with data engineers to build pipelines	• ML team has a voice in data governance discussions • ML team has strong internal data engineering function
Model ownership	• Models are rarely deployed into production	• ML engineers own the models that they deploy into production	• ML team hands off models to user, but is responsible for maintaining them

ML team structures - design choices

	ML R&D	Embedded ML	ML Function	ML First
Software engineering vs research	• Research prioritized over SWE skills • Researcher-SWE collaboration lacking	• SWE skills prioritized over research skills • Often, all researchers need strong SWE as everyone expected to deploy	• Each team has a strong mix of SWE and research skills • SWE and researchers work closely together within team	• Different teams are more or less research oriented • Research teams collaborate closely with SWE teams
Data ownership	• ML team has no control over data • ML team typically will not have data engineering component	• ML team generally does not own data production / mgmt • Work with data engineers to build pipelines	• ML team has a voice in data governance discussions • ML team has strong internal data engineering function	• ML team often owns company-wide data infrastructure
Model ownership	• Models are rarely deployed into production	• ML engineers own the models that they deploy into production	• ML team hands off models to user, but is responsible for maintaining them	• ML team hands off models to user, who operates and maintains them

ML team structures - design choices

	ML R&D	Embedded ML	ML Function	ML First
Software engineering vs research	• Research prioritized over SWE skills • Researcher-SWE collaboration lacking	• SWE skills prioritized over research skills • Often, all researchers need strong SWE as everyone expected to deploy	• Each team has a strong mix of SWE and research skills • SWE and researchers work closely together within team	• Different teams are more or less research oriented • Research teams collaborate closely with SWE teams
Data ownership	• ML team has no control over data • ML team typically will not have data engineering component	• ML team generally does not own data production / mgmt • Work with data engineers to build pipelines	• ML team has a voice in data governance discussions • ML team has strong internal data engineering function	• ML team often owns company-wide data infrastructure
Model ownership	• Models are rarely deployed into production	• ML engineers own the models that they deploy into production	• ML team hands off models to user, but is responsible for maintaining them	• ML team hands off models to user, who operates and maintains them

Module overview

Roles
- ML-related roles and the skills they require

Hiring
- How to hire ML engineers. How to get hired.

Orgs
- How ML teams are organized and how they fit into the broader organization

Managing
- How to manage a ML team & ML product

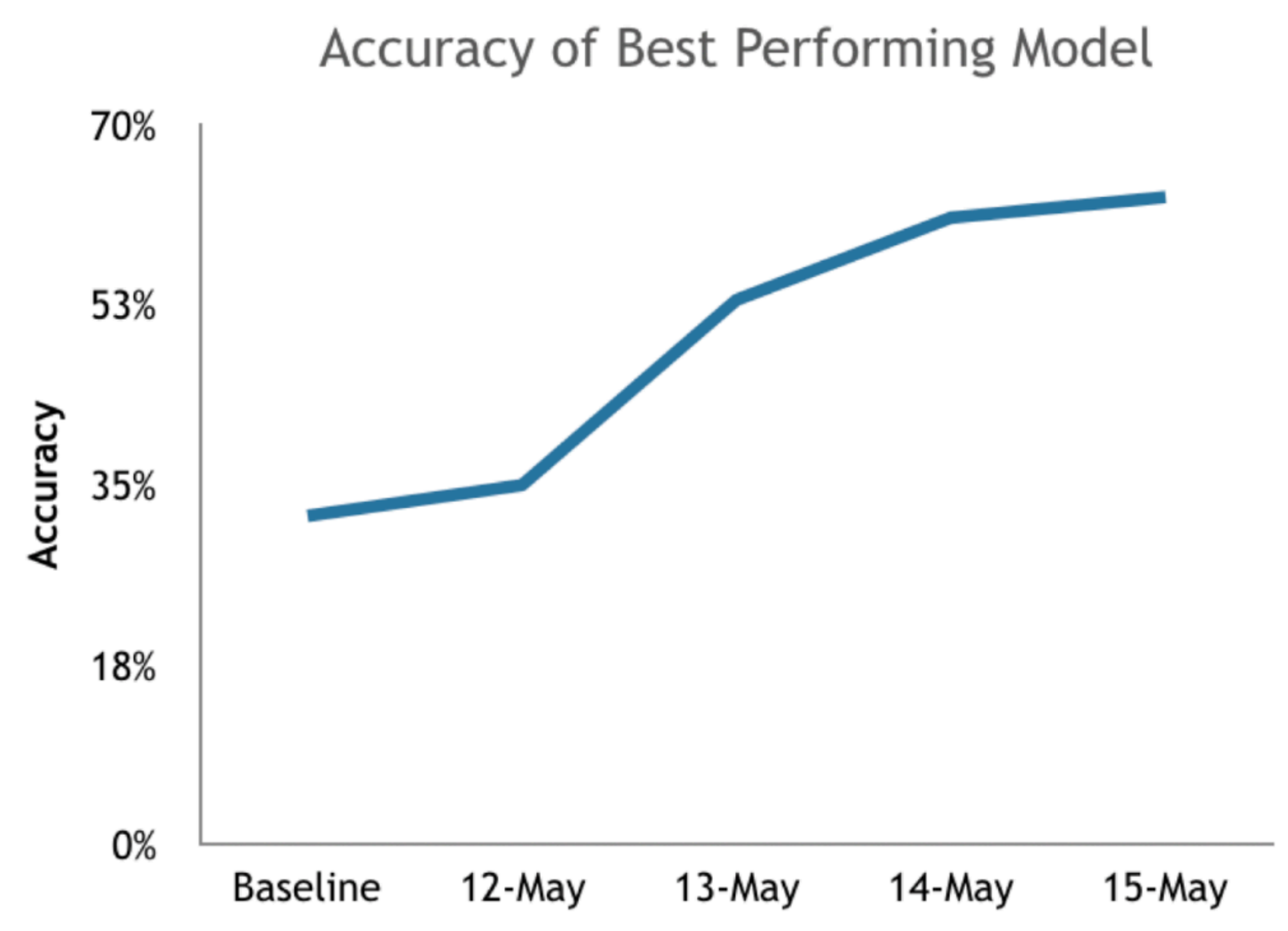
Design
- Design considerations for ML products

Product management for ML is challenging

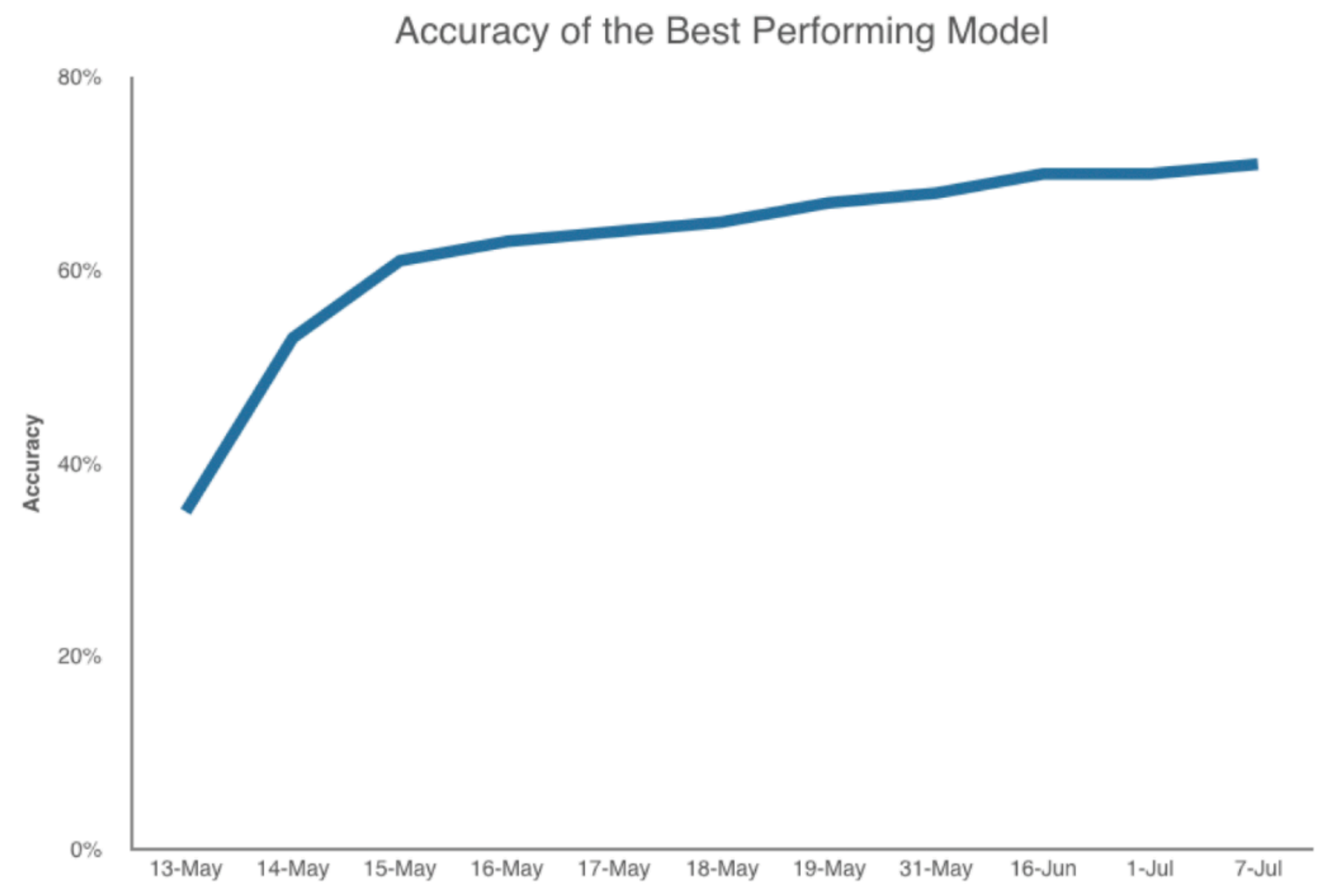
- It's hard to tell in advance how hard or easy something is

Product management for ML is challenging

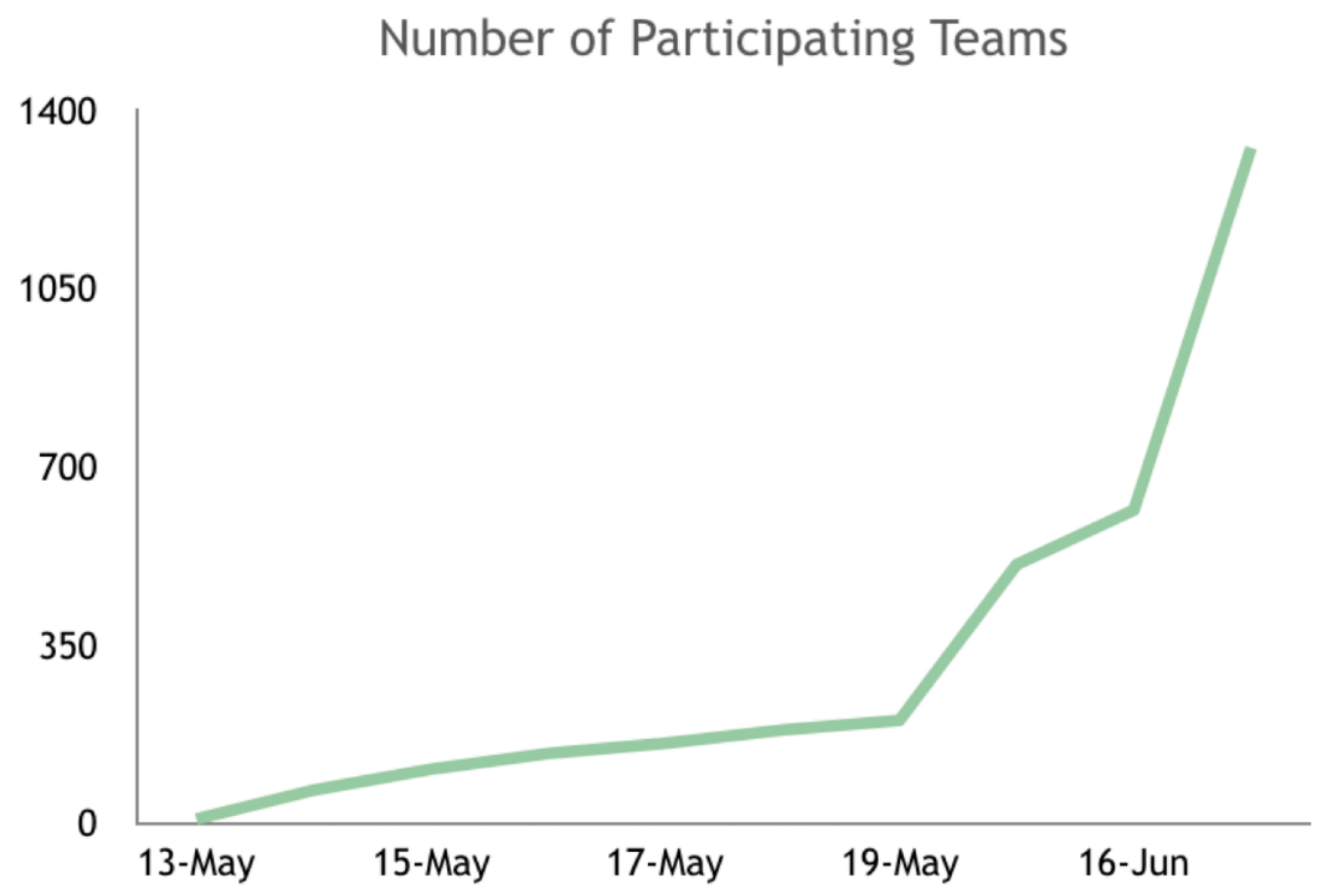
Accuracy improvement in first week



Accuracy improvement in three months



Effort



It’s hard to tell in advance how easy or hard something is

https://medium.com/@l2k/why-are-machine-learning-projects-so-hard-to-manage-8e9b9cf49641

Product management for ML is challenging

- It’s hard to tell in advance how hard or easy something is
- ML progress is nonlinear
  - Very common for projects to stall for weeks or longer
  - In early stages, difficult to plan project because unclear what will work
  - As a result, estimating project timelines is extremely difficult
  - I.e., production ML is still somewhere between “research” and “engineering”

Product management for ML is challenging

- It's hard to tell in advance how hard or easy something is
- ML progress is nonlinear
- There are cultural gaps between research and engineering
  - Different values, backgrounds, goals, norms
  - In toxic cultures, the two sides often don't value one another

Product management for ML is challenging

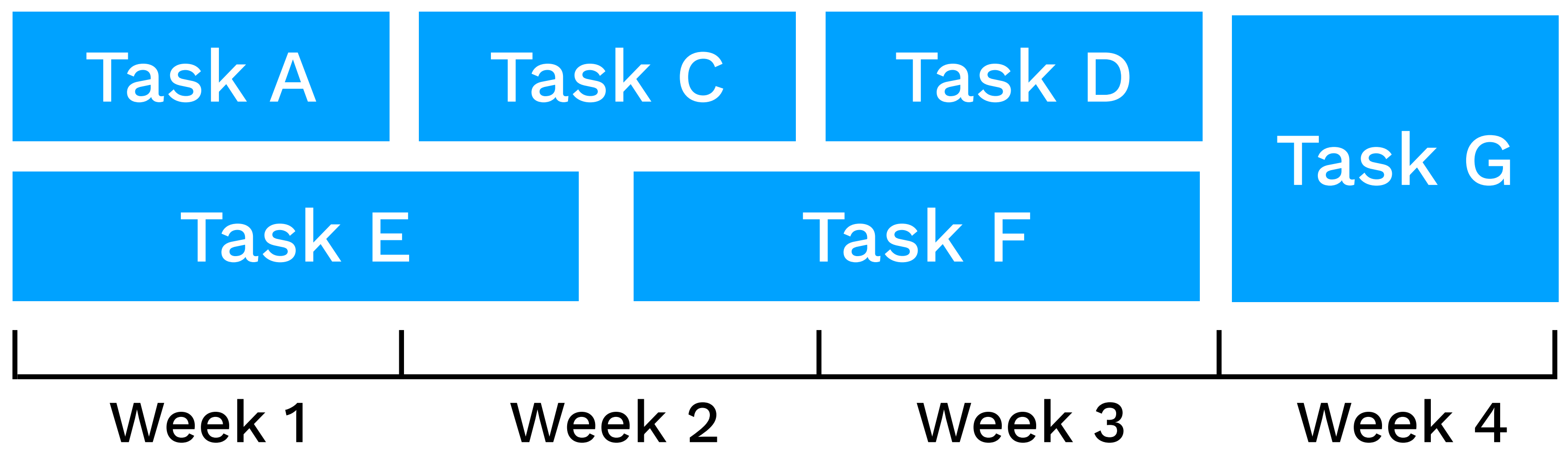
- It’s hard to tell in advance how hard or easy something is
- ML progress is nonlinear
- There are cultural gaps between research and engineering
- Leaders often don’t understand it

How to manage ML projects better

- Do ML Project planning probabilistically

How to manage ML teams better

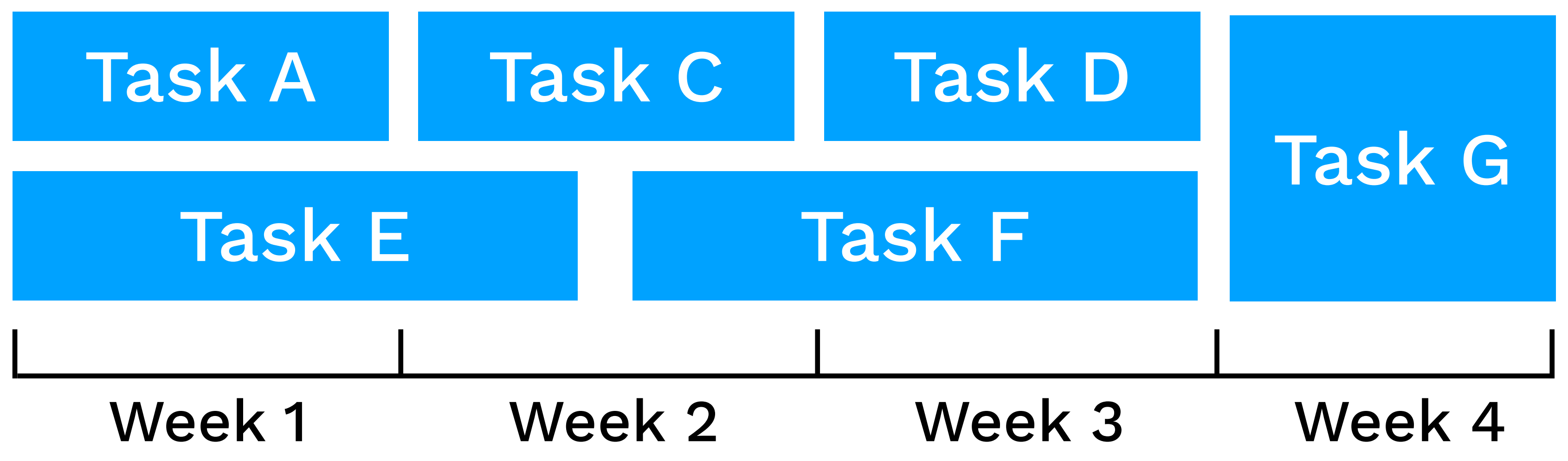
- Do ML project planning probabilistically
  - From:



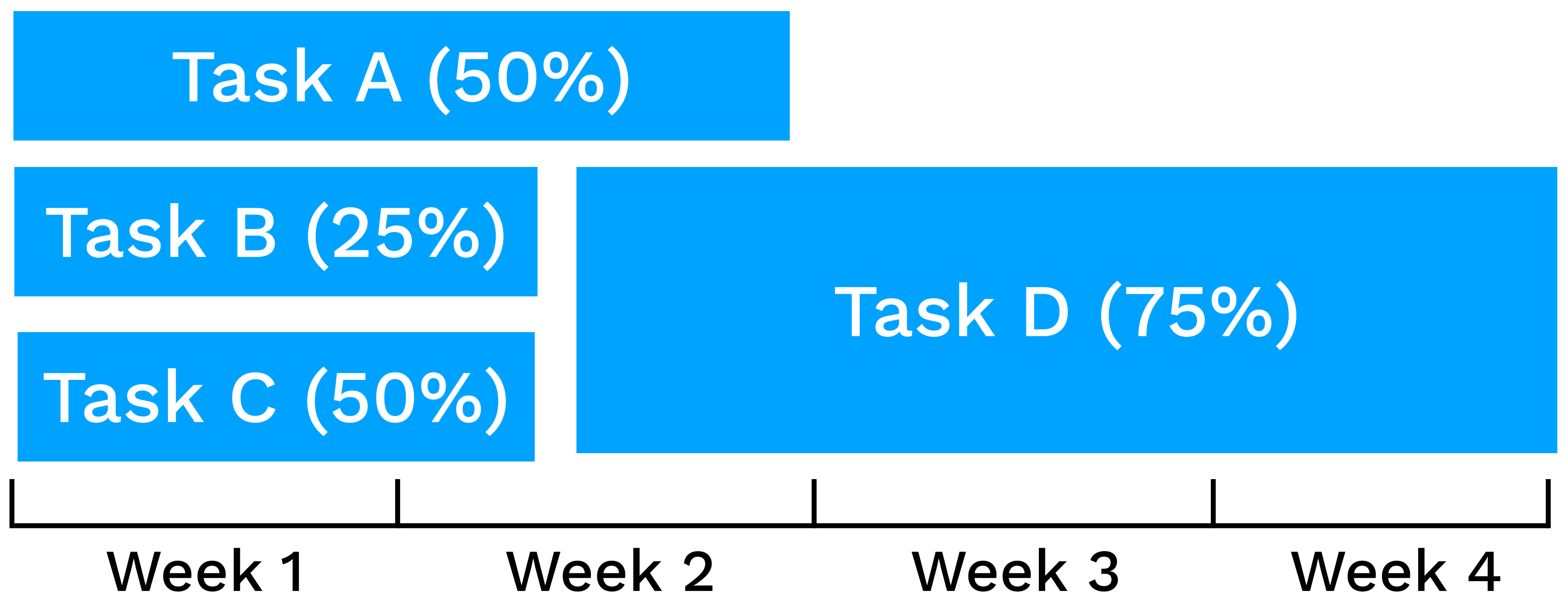
  - To:

How to manage ML teams better

- Do ML project planning probabilistically
  - From:

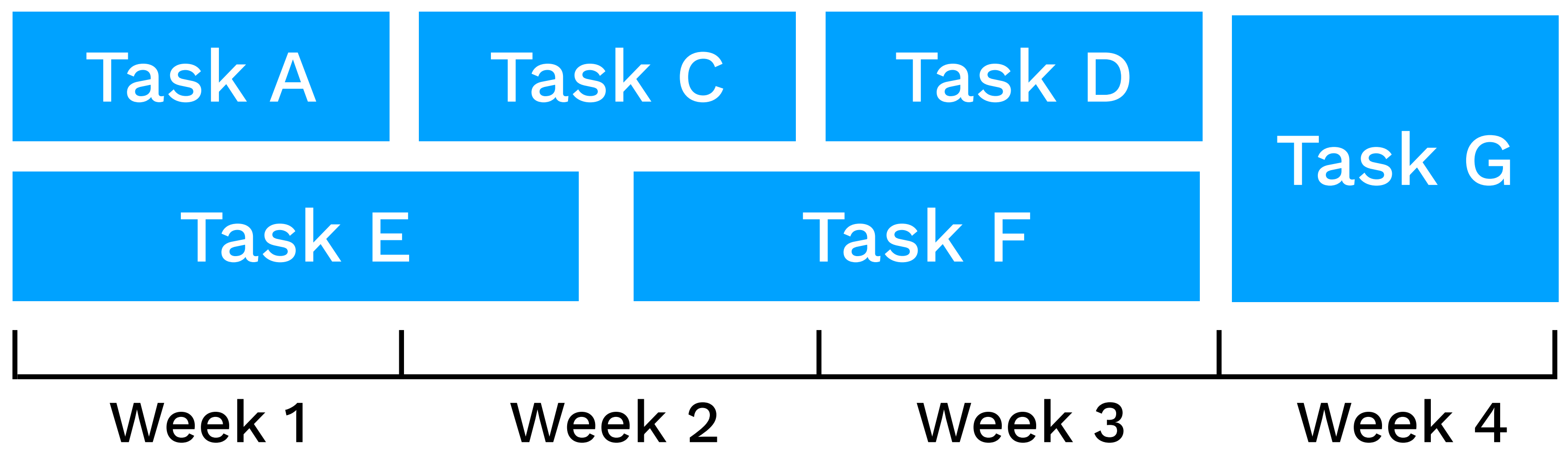


  - To:

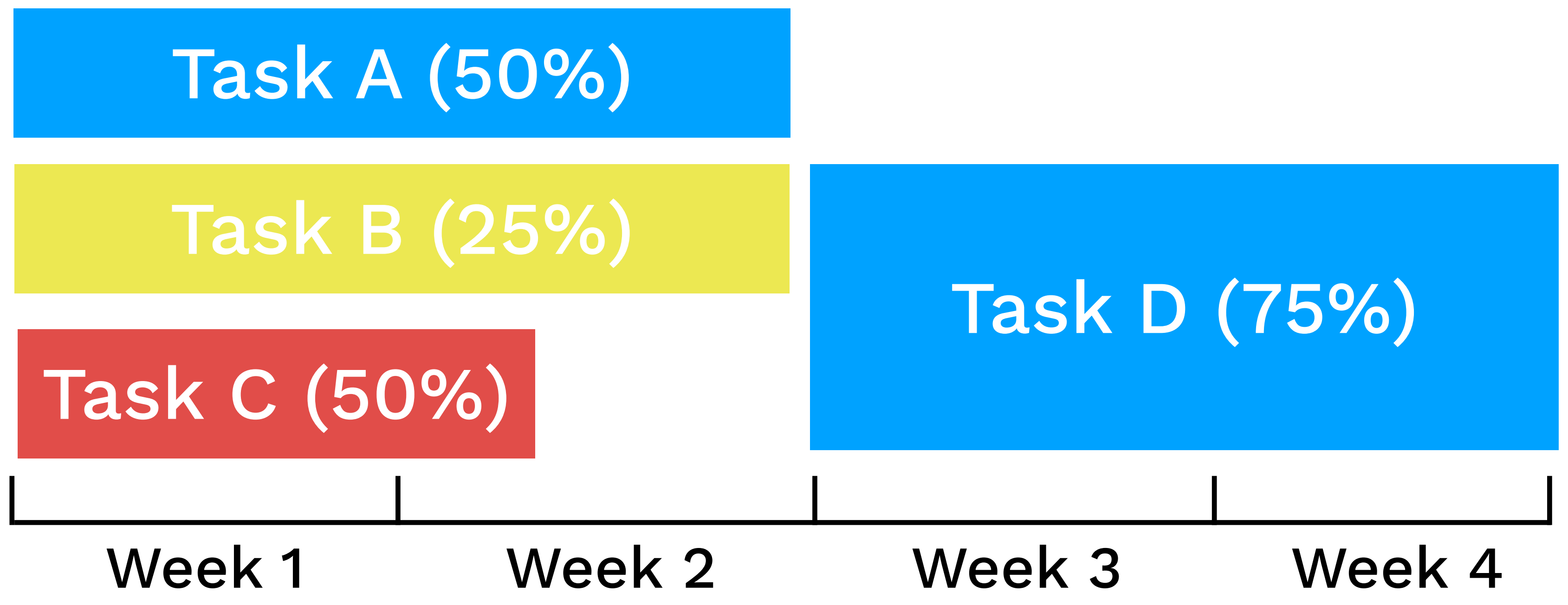


How to manage ML teams better

- Do ML project planning probabilistically
  - From:

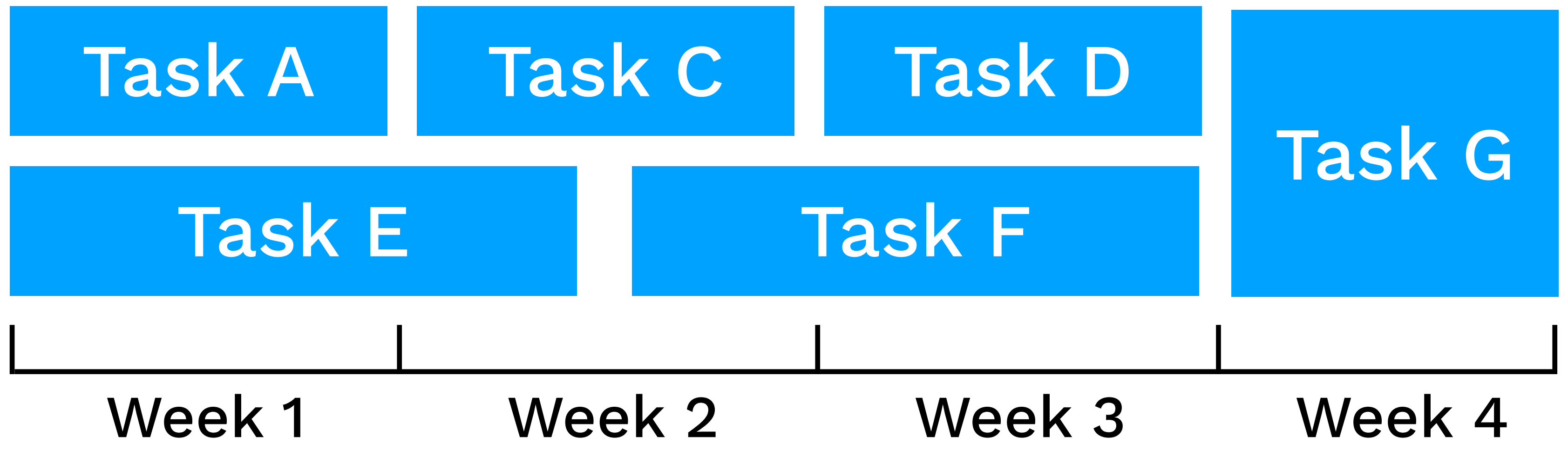


  - To:

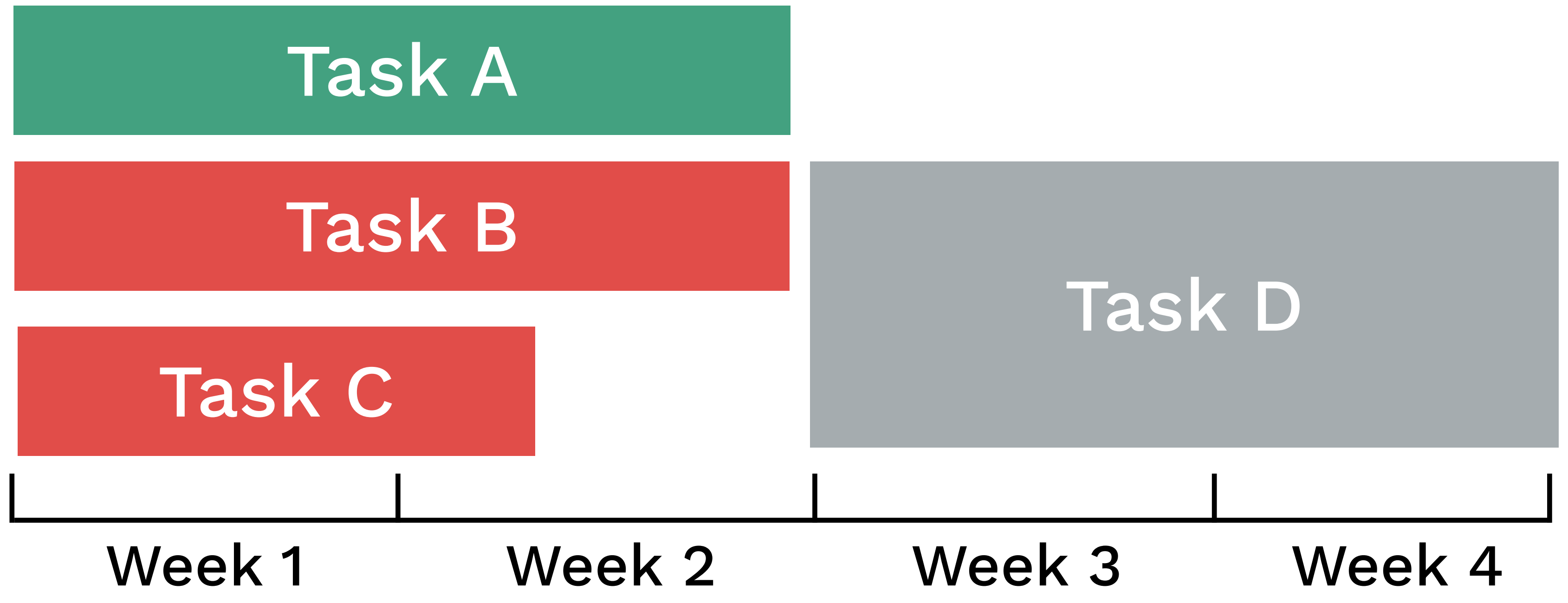


How to manage ML teams better

- Do ML project planning probabilistically
  - From:

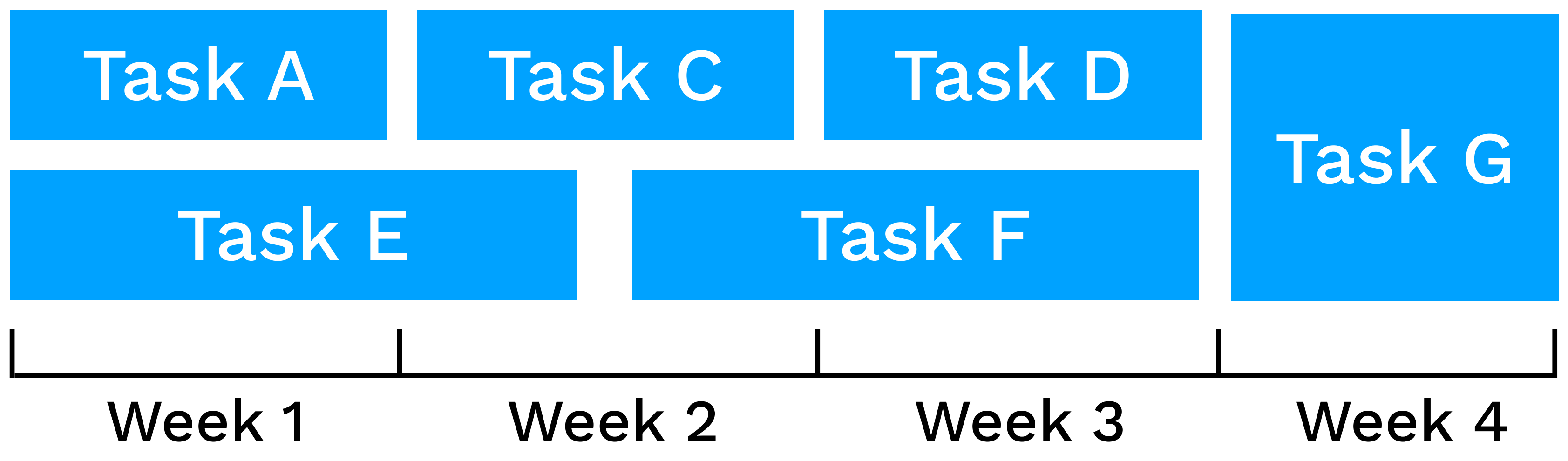


  - To:

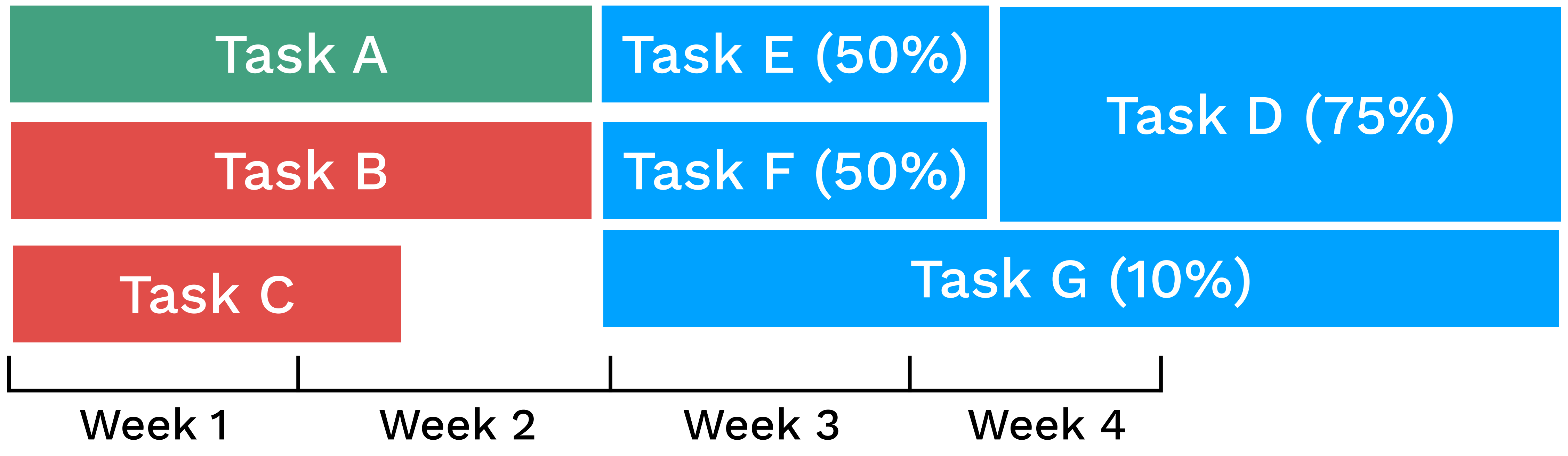


How to manage ML teams better

- Do ML project planning probabilistically
  - From:



  - To:



How to manage ML teams better

- Do ML Project planning probabilistically
- Attempt a portfolio of approaches
- Measure progress based on inputs, not results
- Have researchers and engineers work together
- Get end-to-end pipelines together quickly to demonstrate quick wins
- Educate leadership on ML

What doesn’t the rest of your org understand about ML?

- Where it can (and can’t!) be used
- How to measure if it’s working
- That it’s probabilistic: user-visible failures are inevitable
- That it needs to be managed differently than other software projects

Resources for educating execs

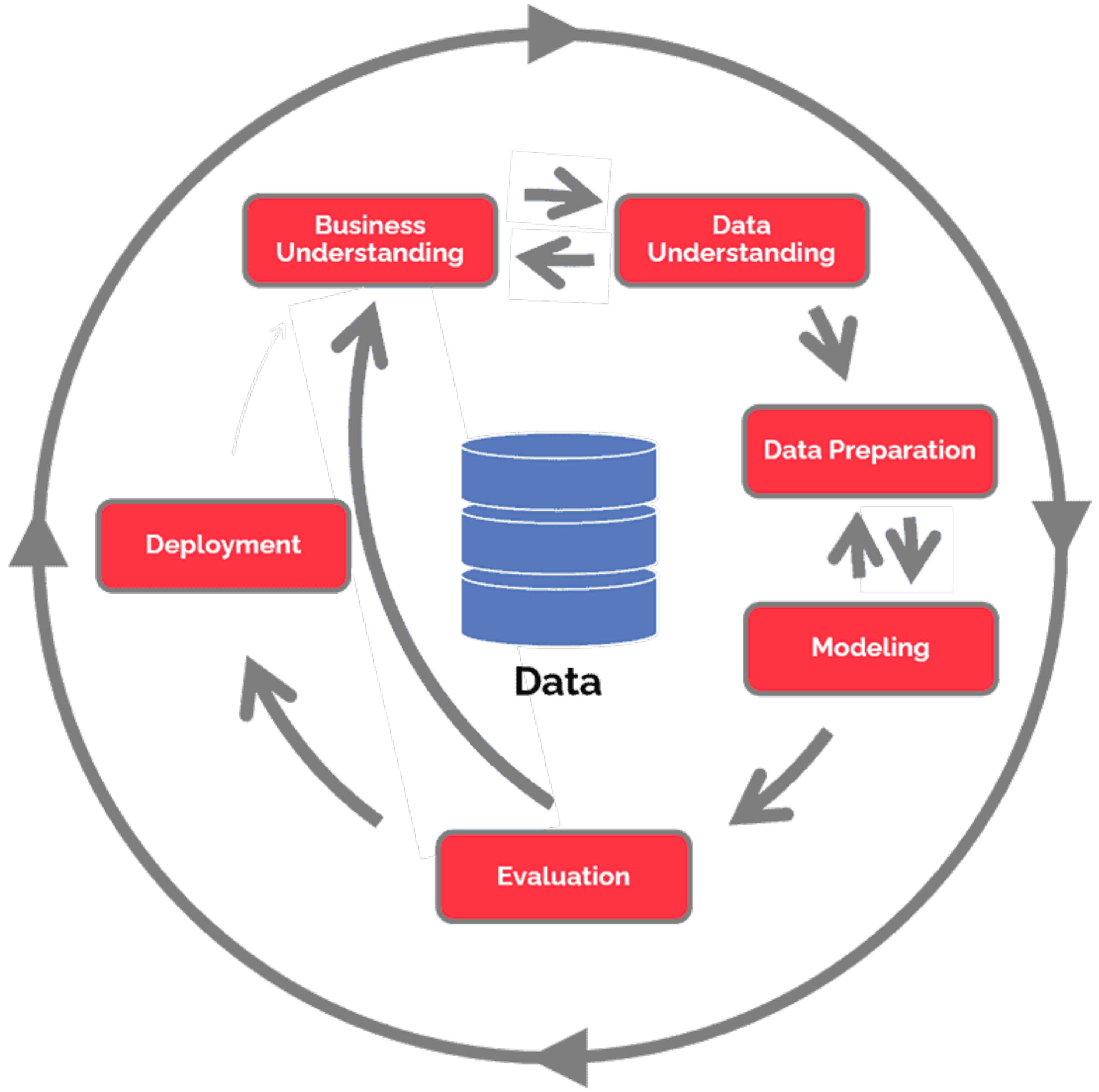
- Pieter Abbeel's AI Strategy class: https://emeritus-executive.berkeley.edu/artificial-intelligence/
- Google's People + AI Guidebook: https://pair.withgoogle.com/guidebook

Two types of ML PMs

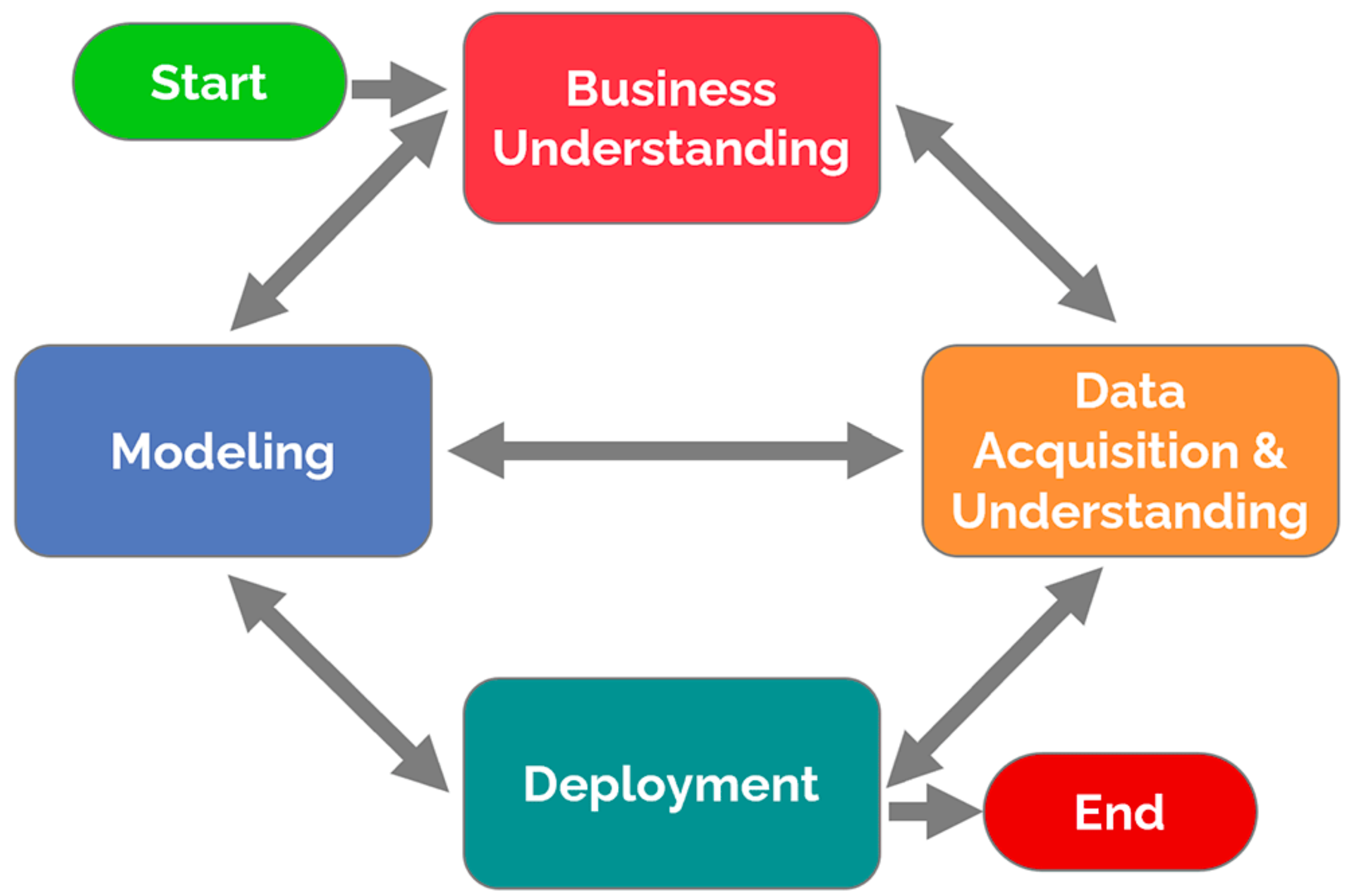
- Task ML PMs
  - Responsible for a specific ML-powered product / product feature
  - Specialized knowledge of ML
  - E.g., PM for trust and safety, PM for recommendations, etc
- Platform ML PMs
  - Responsible for managing workflow & communicating priorities for a centralized ML team
  - Broad knowledge of ML — where it can and can't be applied
  - Responsible for spreading ML knowledge & culture throughout the org
  - Helping stakeholders get "bought in" to trusting the model

Emerging ML Project management methodologies

CRoss Industry Standard Process for Data Mining (CRISP-DM)



Team Data Science Process (TDSP)



https://www.datascience-pm.com/

Emerging ML Project management methodologies

- What are they?
  - Structured, data-science native approaches to project management
- Why use them?
  - Provide standardization for project stages, roles, and artifacts
  - Used at Microsoft (TDSP) and IBM / others (CRISP-DM)
- Which one?
  - TDSP is more structured, CRISP-DM is more high-level
  - If you need an actual project management framework, try TDSP
- When to use them?
  - Reasonable to use if you have a large-scale coordination problem
  - Otherwise, skip it — they are more focused on “traditional” DS and will slow you down

Module overview

Roles	ML-related roles and the skills they require
Hiring	How to hire ML engineers. How to get hired.
Orgs	How ML teams are organized and how they fit into the broader organization
Managing	How to manage a ML team & ML product
Design	**Design considerations for ML products**

What users think when they hear "AI powered-product"

Understands the world

Knows me better than I do

Superhuman - does it like I do, but better

Generalizes to new problems

Learns from its mistakes

What they actually get

Weird little guys
(fail in strange,
unexpected ways)

Get distracted easily
(don't generalize
outside of a narrow
domain)

No learning without
treats
(feedback / rewards)

May misbehave if
left unattended

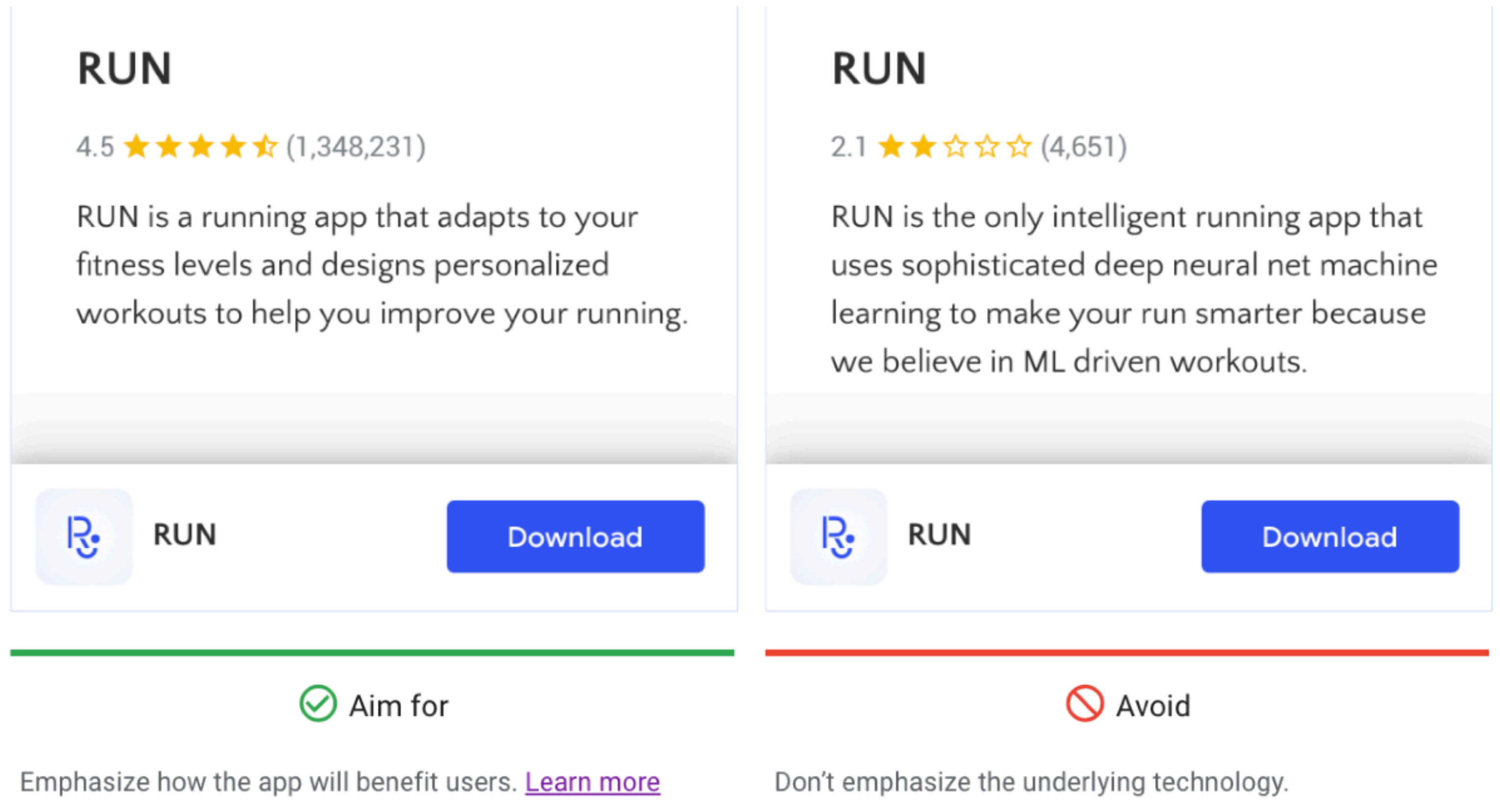
Can't teach an old one new tricks
(hard to adapt knowledge to new
tasks / context)

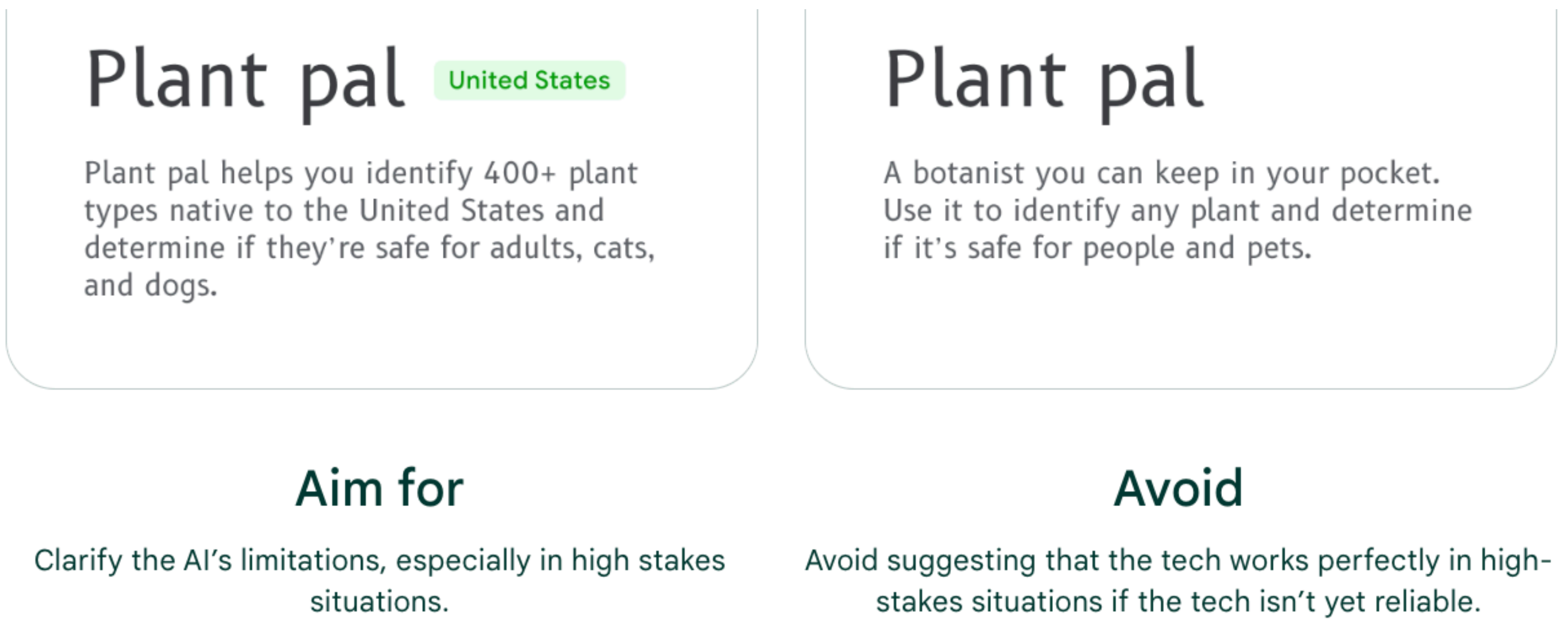
Good ML product design bridges users expectations vs reality

- Help users understand the benefits & limitations of the model
- Handle failures gracefully: don’t over-rely on automation and fall back to human-in-the-loop
- Build feedback loops to improve the system

Explaining the benefits & limitations of the system

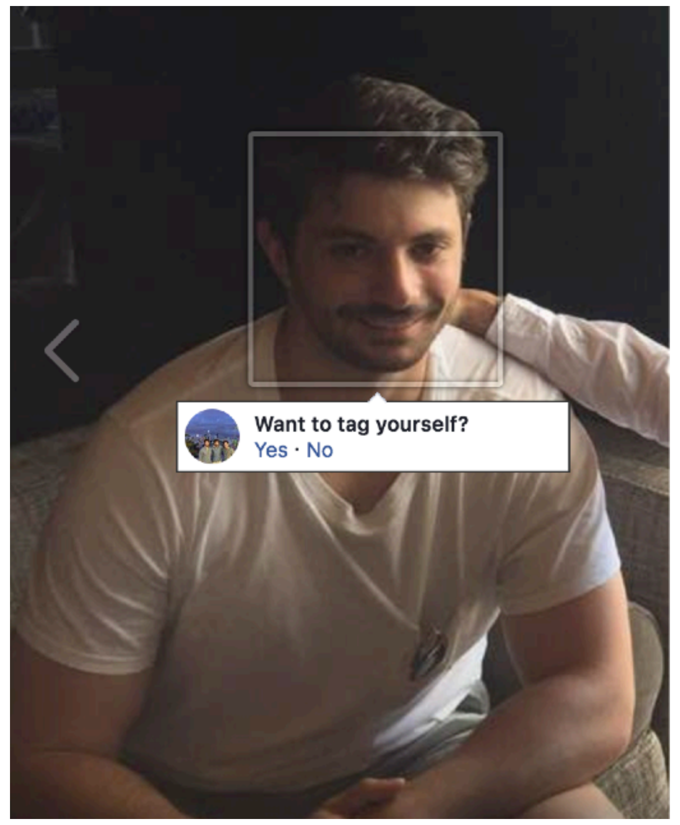
- Focus on benefit for users, not that it's "AI-powered"
- If you make the system feel "human-like" (unconstrained input, human-like responses), expect users to treat it as human-like
- Explain the model's limitations, and consider baking in its intended use as "guardrails" for its acceptable inputs

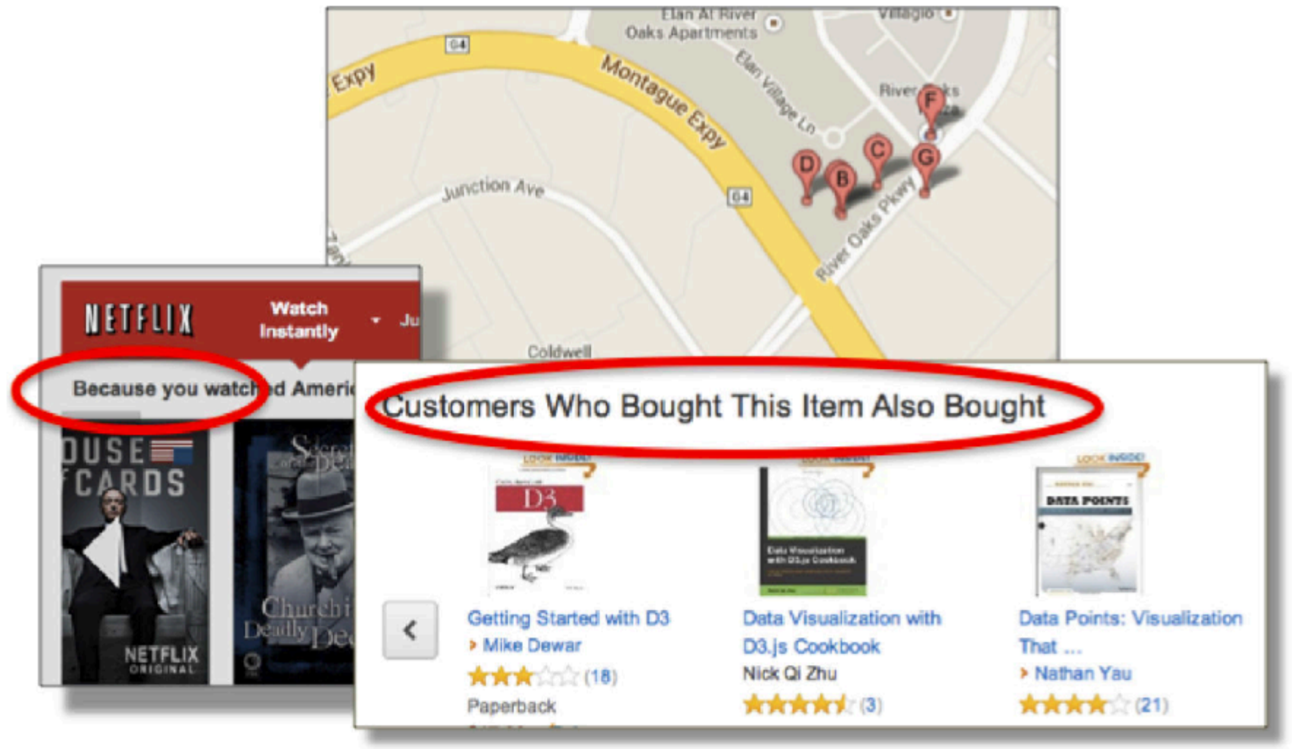


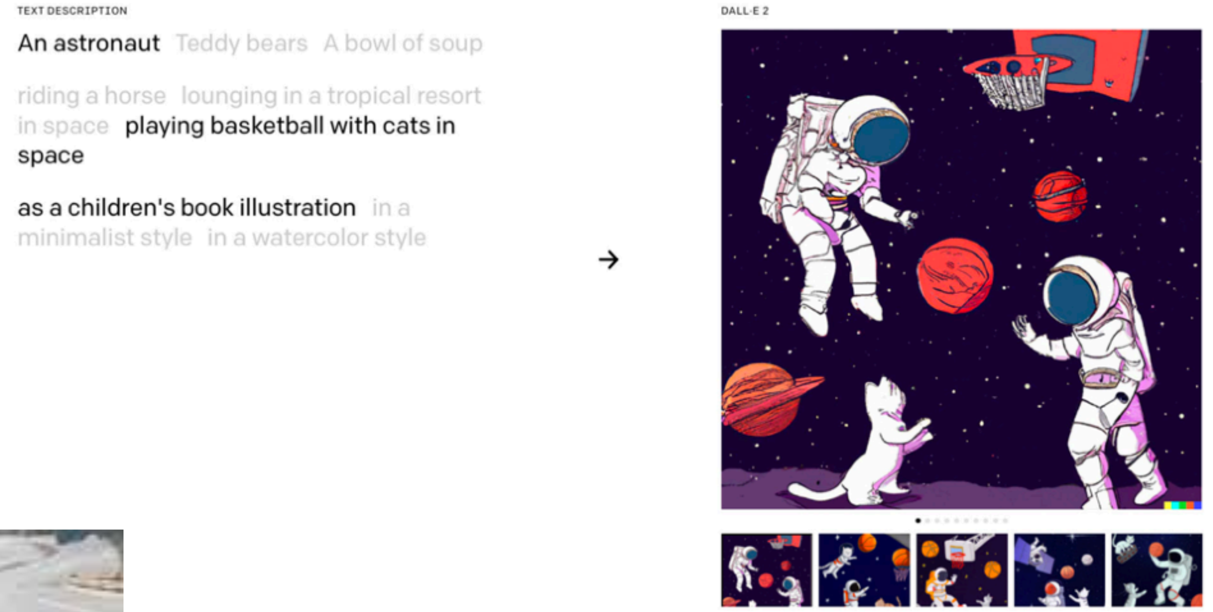


Handle failures gracefully via human-in-the-loop

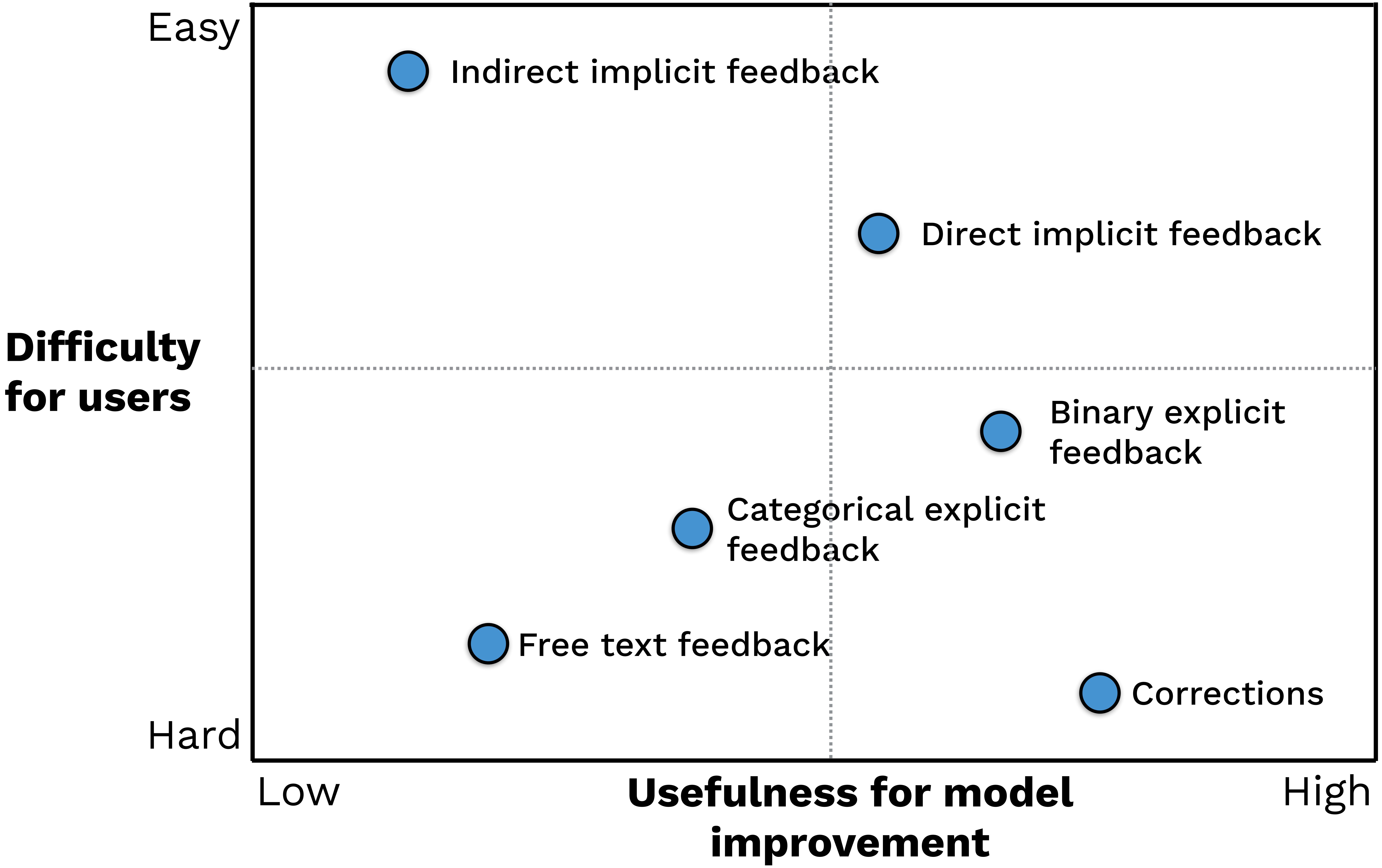
- Automation is great, but failed automation is worse than no automation at all
  - Consider low-friction ways to let users “confirm” the model’s predictions
  - Present multiple options when possible
- Mitigate the cost of bad predictions
  - Consider choosing the response based on model confidence
  - Let users “take the wheel” if things are going wrong



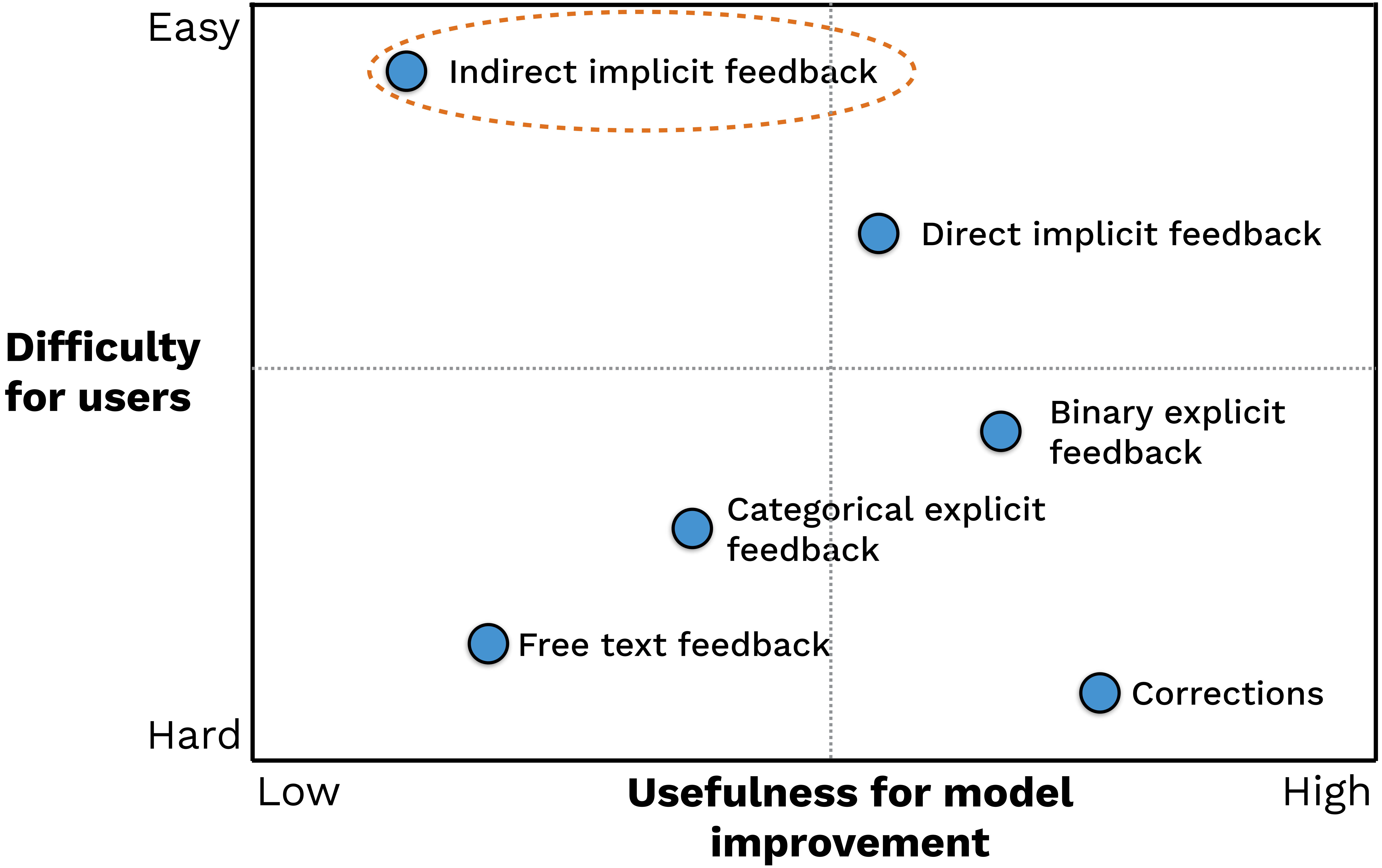




Types of user feedback

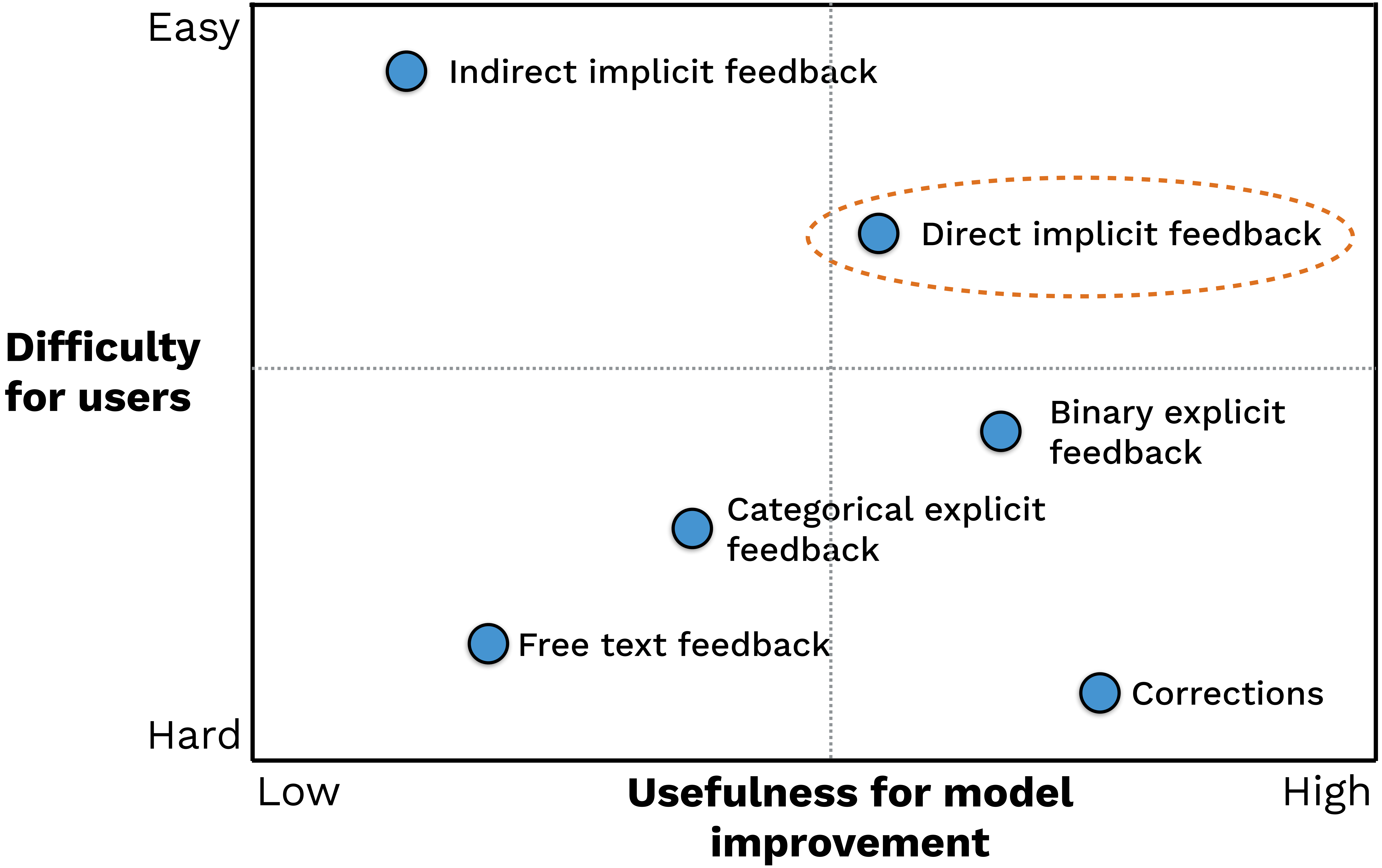


Types of user feedback



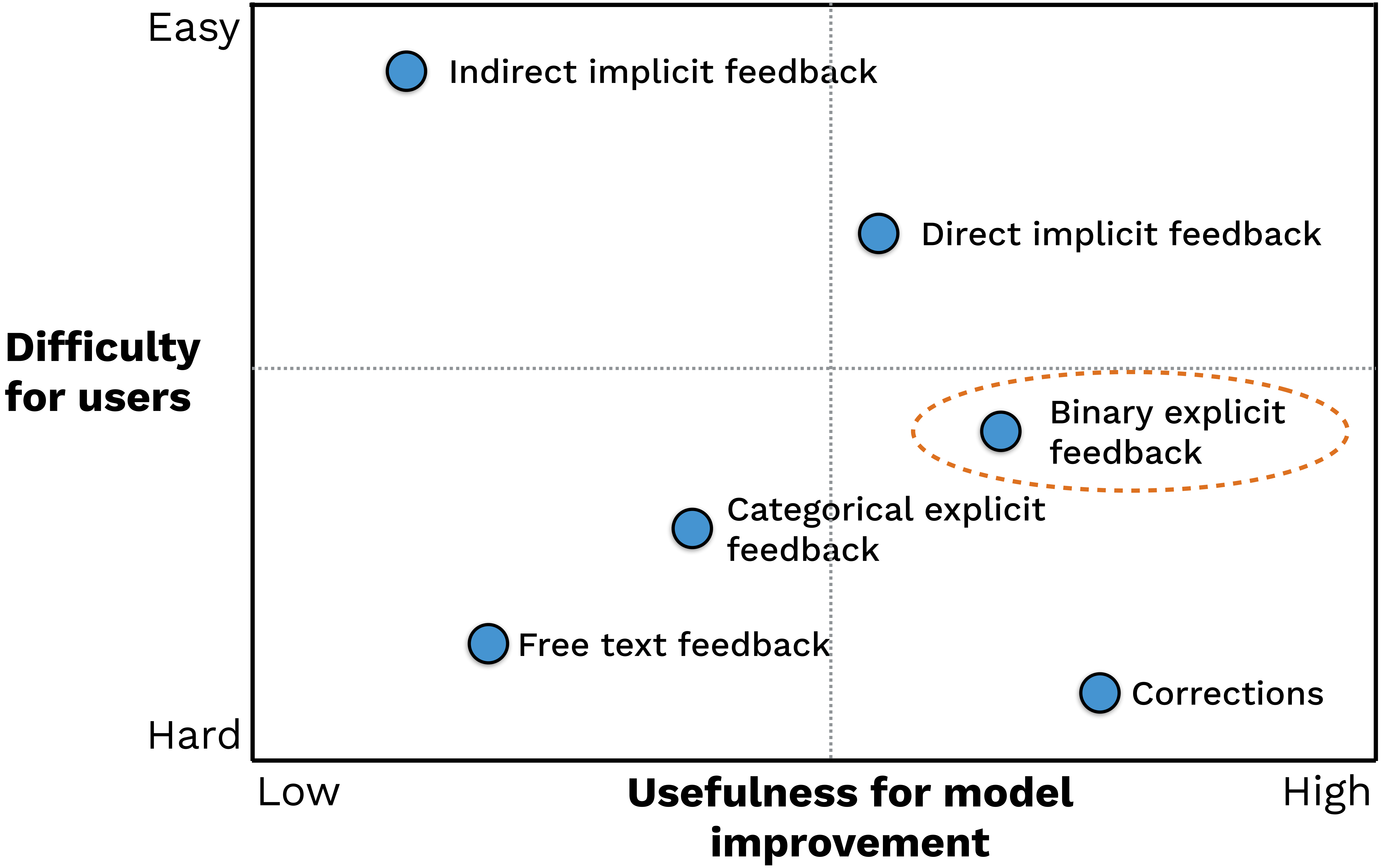
- Signals about user behavior that are a proxy for model performance
- Super easy to collect
- Useful because they usually correspond to important "outcomes", but hard to tell if the model is the cause
- E.g., "did the user churn"

Types of user feedback



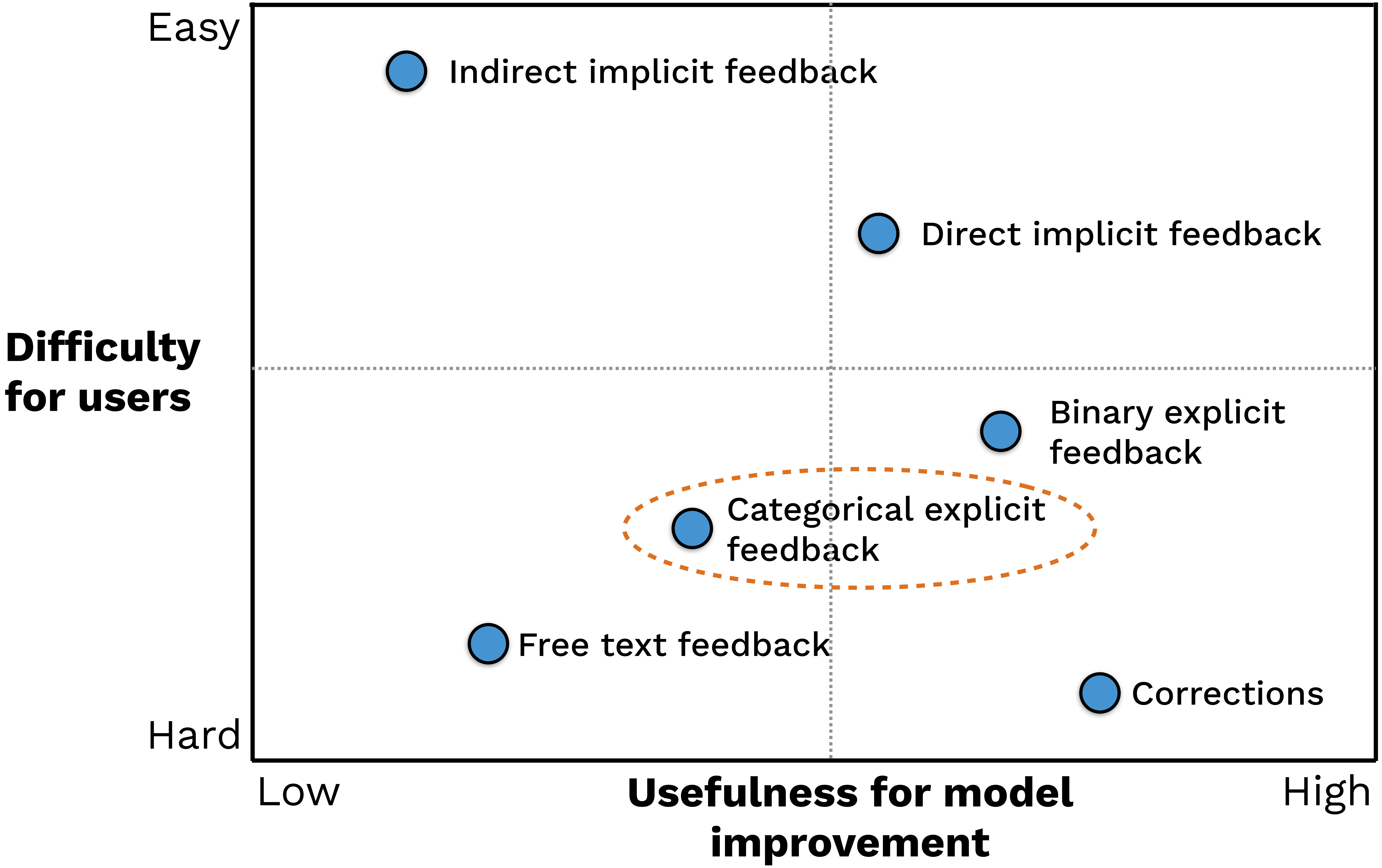
- Signals from the product that directly measure how useful the prediction is
- Often of the form "did the user take the next step"
- Sometimes require redesigning the product
- Can train on them directly
- E.g., "did the user click the recommendation", "did the user send the suggested email", "did the user copy our suggestion"

Types of user feedback



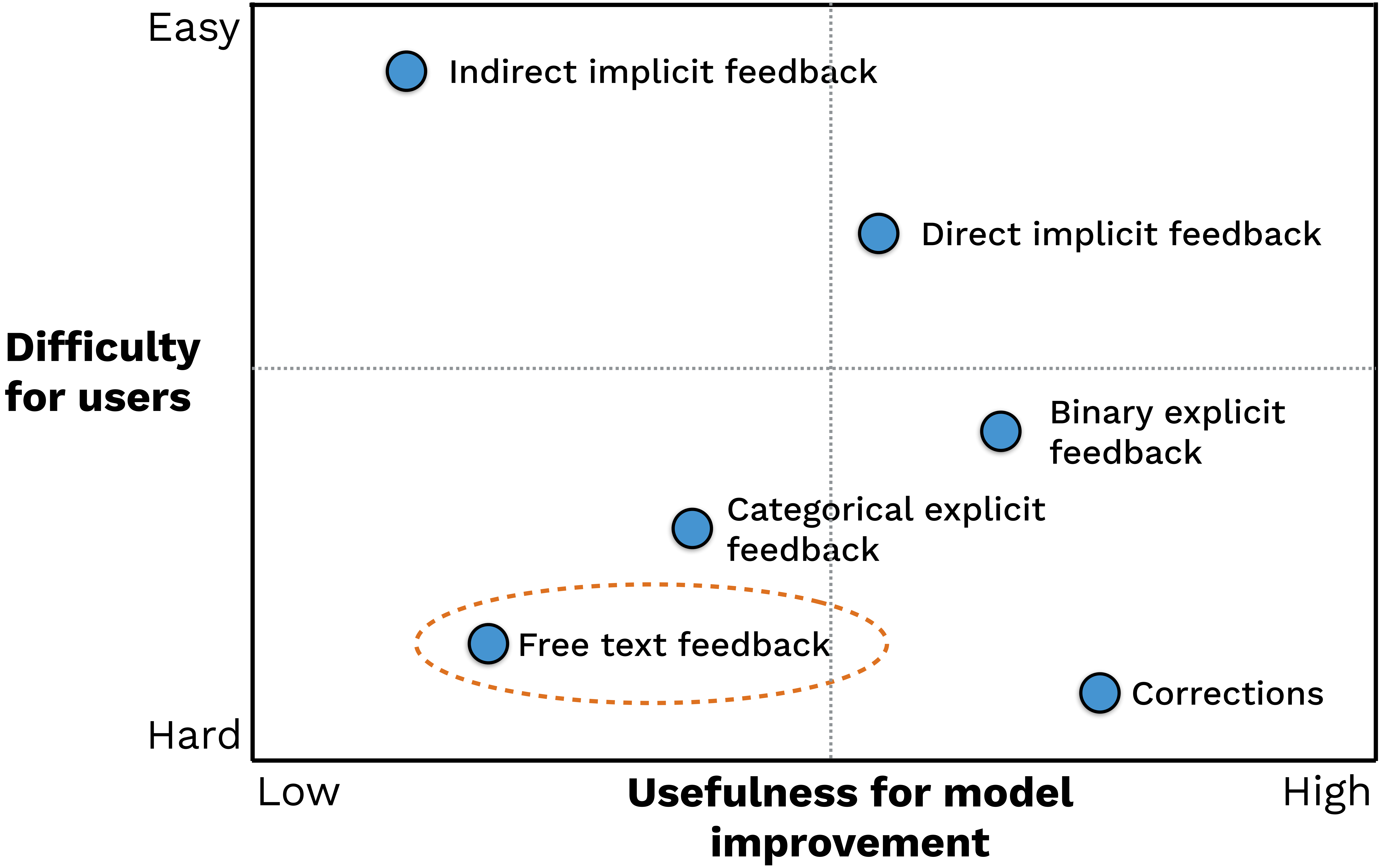
- Thumbs up / thumbs down button or this vs that option
- Easy for users, decent training signal
- See e.g., “Fine-Tuning Language Models from Human Preferences” (Ziegler et al — OpenAI 2019)

Types of user feedback



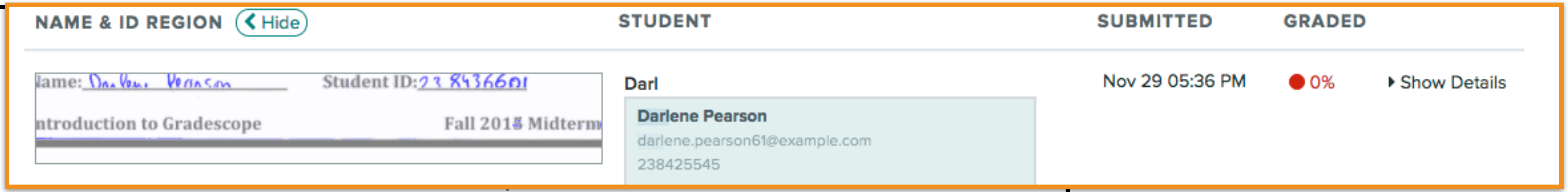
- Allow users to help you "categorize" bad predictions
- E.g., "incorrect", "irrelevant", "offensive", "not useful", etc
- Can be a "second step" after binary feedback
- Difficult to train on it, but useful for debugging

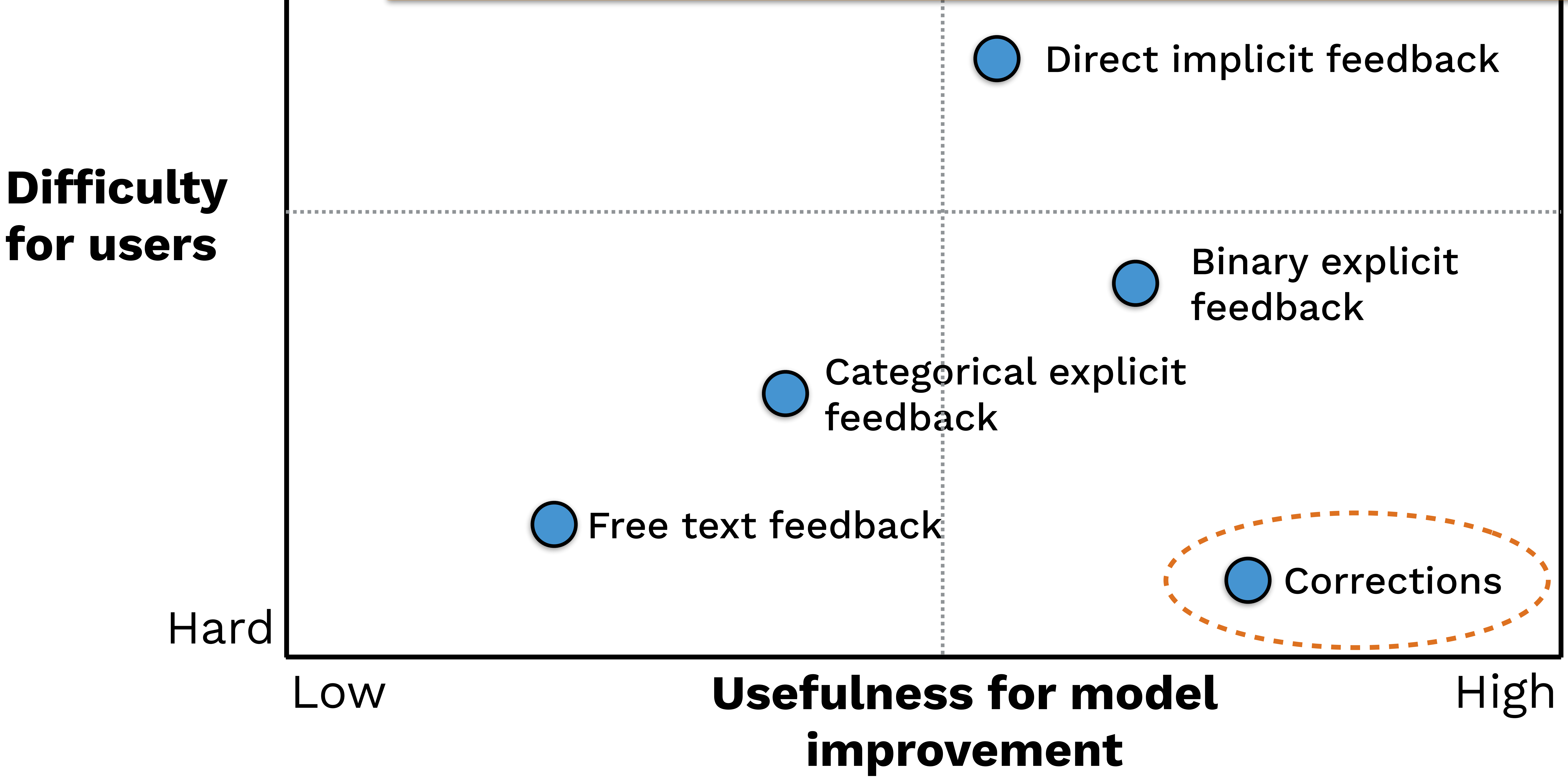
Types of user feedback



- Free text box for feedback on predictions
- E.g., support tickets / support requests
- Hard for users, difficult to use for model developers
- Yet, useful, because they tend to highlight “highest friction” predictions

Types of user feedback

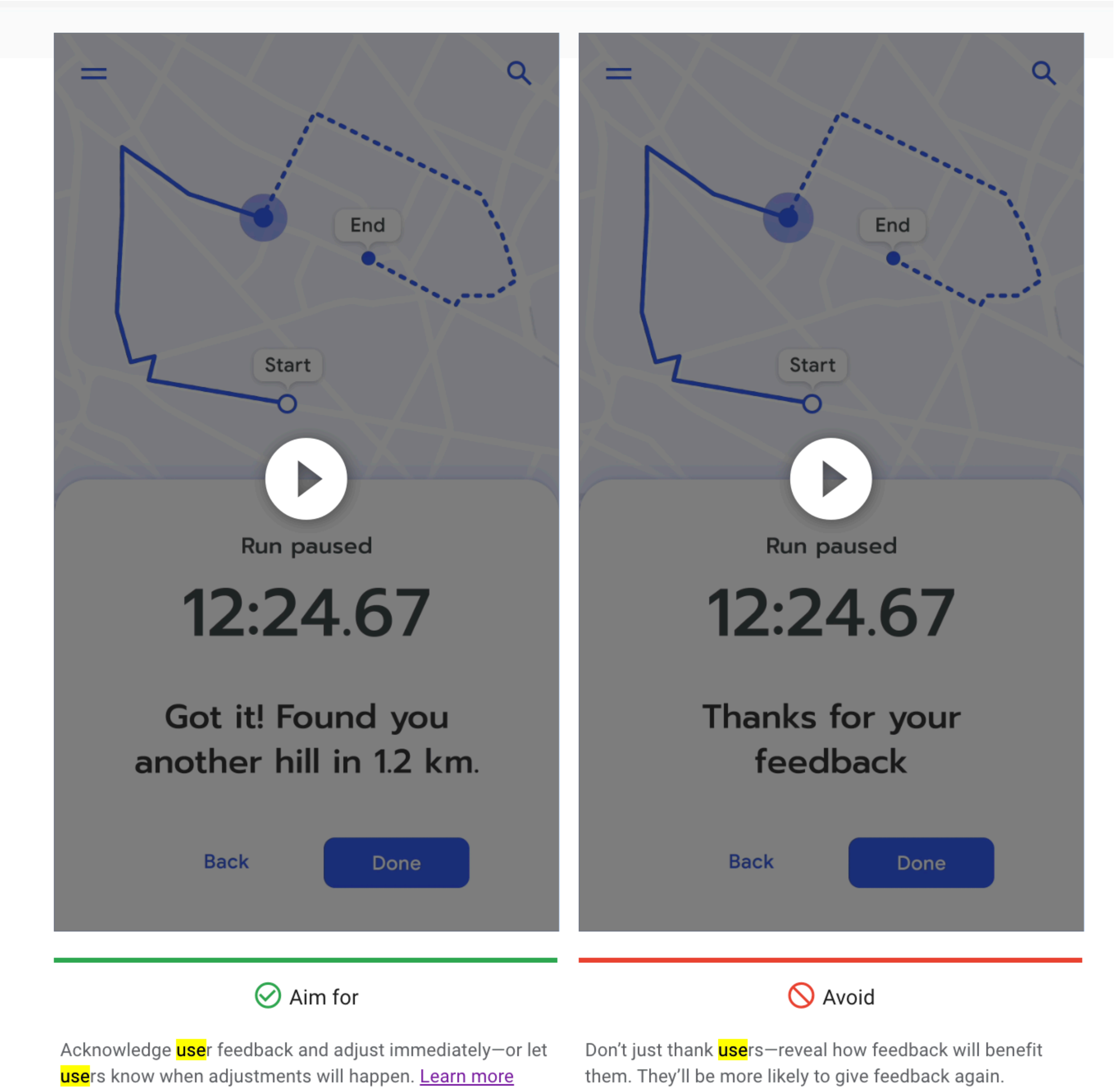




- If you can get users to label stuff for you... must be nice :)
- Often the case where the thing predicted is useful downstream *in the same product*

Encouraging users to give you feedback

- Gather feedback as part of an existing user workflow
- Make it explicit how the feedback will make the user experience better (the more explicit and shorter time to impact the better)
- Try to avoid relying on users altruism and desire to help you



Great ML product experiences are designed from scratch

- ML is not superhuman intelligence in silicon
- Your product experience needs to help users understand that
- It also needs to help users interact safely with an imperfect system via human-in-the-loop and guardrails
- Great ML products are powered by feedback loops — how can your users help make the product better?

Learn more about ML product design

- Google's People + AI Guidebook (https://pair.withgoogle.com/guidebook)
- Guidelines for Human-AI Interaction (Amershi et al 2019, https://dl.acm.org/doi/abs/10.1145/3290605.3300233)
- Agency plus automation: Designing artificial intelligence into interactive systems (Jeffrey Heer 2018, http://idl.cs.washington.edu/files/2019-AgencyPlusAutomation-PNAS.pdf)
- Designing Collaborative AI (Ben Reinhardt, https://medium.com/@Ben_Reinhardt/designing-collaborative-ai-5c1e8dbc8810)

Conclusion

Roles	• Lots of different skills involved in production ML, so there's an opportunity for many to contribute
Hiring	• Talent is scarce, so be specific about what is must-have. It can be hard to break in as an outsider - use projects to build awareness.
Orgs	• ML teams are becoming more standalone, hence more interdisciplinary
Managing	• Managing ML teams is hard. There's no silver bullet, but shifting toward probabilistic planning can help
Design	• Today's ML systems are limited. Make sure users understand that, and that can help you mitigate those limitations