CMPUT 267: Machine Learning I · Fall 2026

Course Introduction

Xi Ye

2026-09-01

CMPUT 267: Machine Learning I

Course Information

Instructor
Xi Ye

Lectures
Tuesdays & Thursdays
12:30–1:50 p.m.

Location
CCIS 1-440

Today

1. **Important** course policies
2. Course outline

Lectures

ⓘ Lectures are held **in person**.

⚠ Lectures are not recorded.

- One or two lectures (tentatively Oct. 6, Oct. 8) may move to Zoom while I am away at conferences.

Course Materials

The course is based on the **course notes**, which were written specifically for this course.

The lectures will cover the course notes. Additional useful materials can be found on the course website.

Slides and the whiteboard

- We use slides during lectures to highlight the main ideas; slides will be posted after class.
- I may occasionally use the whiteboard; all whiteboard content will be included in the posted slides or course notes.

Course Notes

The course notes were created and are maintained by

Martha White

Associate Professor
Department of Computing Science
University of Alberta

Volodymyr (Vlad) Tkachuk

PhD Student
Department of Computing Science
University of Alberta

Prerequisites

- Raise your hand if you have learned (or will be learning this semester) multivariable derivatives
- Raise your hand if you have learned (or will be learning this semester) matrix multiplication
- Raise your hand if you have learned (or will be learning this semester) probability theory and Bayes rule

Course Team

Teaching Assistants

- Jiajing (Jessica) Chen
- Aditi Vidyarthi
- Kaining Yang
- Quang Hieu Pham
- Connor Mitchell
- Gábor Mihucz
- Ammaar Mohammed
- Songtao Wang
- Thuy Duong Nguyen
- Ho Leong (Mike) Luo

Check the course website for the TA office-hour schedule and locations.

Coursework & Grading

10%

Assignments

30%

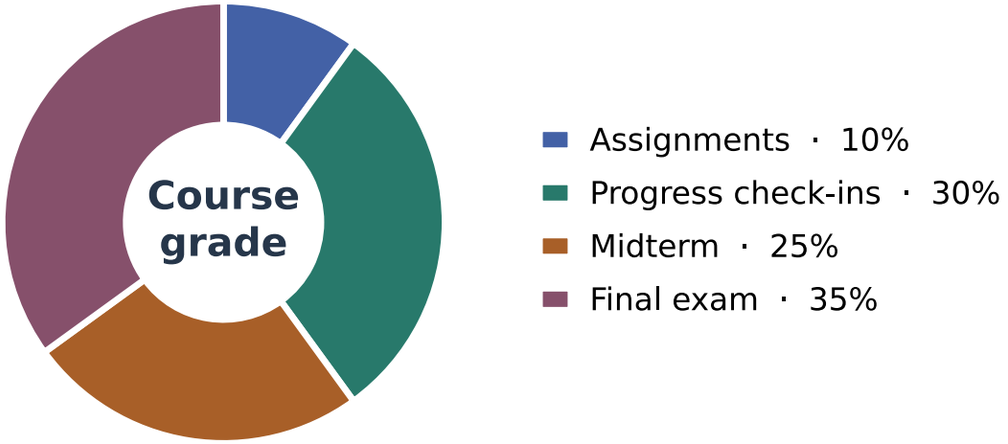
Progress check-ins

25%

Midterm exam

35%

Final exam



Assignments (10%)

- Assignments include both written and coding work.
- Assignments are graded on a **completion basis**.

⚠ We **do not** accept late assignments.

If you can do the assignments, you are likely to succeed on progress check-ins and exams.

Progress Check-Ins (30%)

Two **40-minute, in-class** progress check-ins, worth **15% each**.

Check-In 1

Week 4: Sept. 22 or 24
If missed: 15% moves to the **midterm**.

Check-In 2

Week 10: Nov. 3 or 5
If missed: 15% moves to the **final**.

> ⚠ There are **no make-up progress check-ins**.

The separate **course-policy quiz on Canvas** is due before Week 3 and is required to receive credit for the progress check-ins.

Exams & Final Grades

Midterm (25%)

- Held in class (**tentative date**: Oct. 13)
- If you are granted an excused absence from the midterm, its weight will be transferred to the final exam.

Final (35%)

- Date to be announced

Join Campuswire

Use Campuswire for **all** course communication.

The course team will **not** monitor or respond to course-related email.

Join Campuswire

Scan the QR code

or visit

campuswire.com/p/GA2FC0B85

Passcode

See canvas

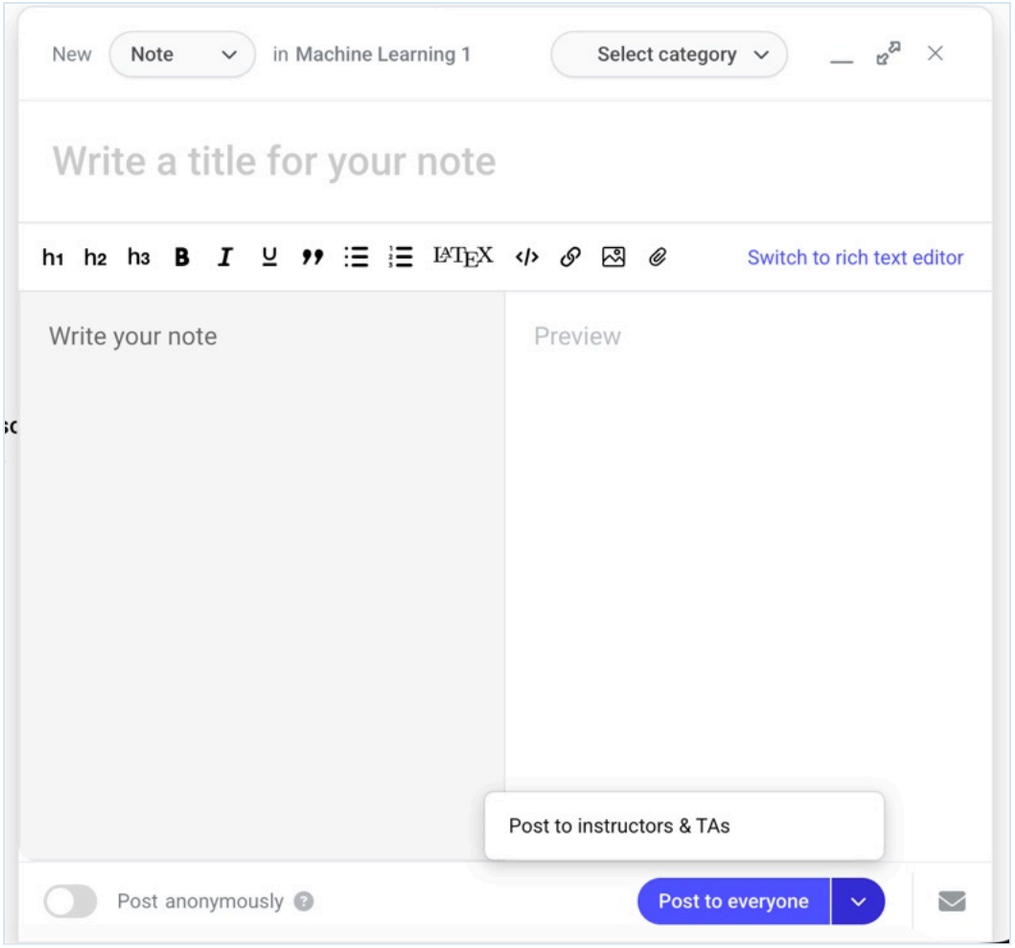
Asking Questions

For questions that may help others and do not reveal assignment solutions:
→ Post a **Q&A on Campuswire**. You may post anonymously.

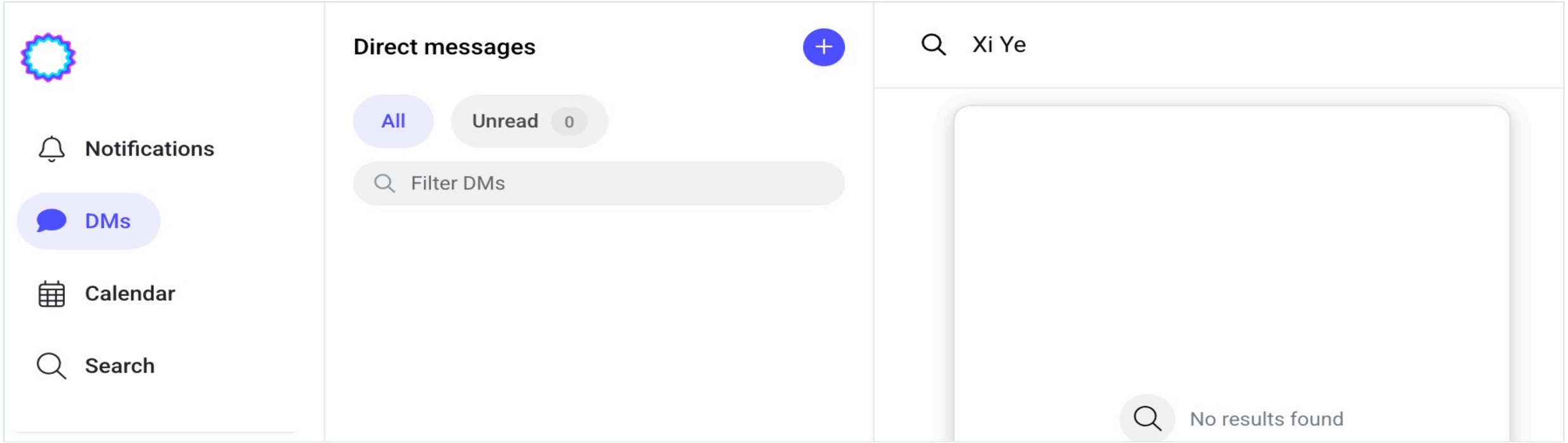
For questions specific to you:
→ Post a **private Q&A on Campuswire** visible only to the TAs and instructor.

For course-related issues (e.g., a missed exam):
→ Direct-message the instructor on **Campuswire.**

Post Only to Instructors & TAs



Direct-Message the Instructor



Remember

Use Campuswire for **all** course communication.

The course team will **not** monitor or respond to course-related email.

AI Policy

- You may use AI assistants to understand concepts in this course. Treat AI as another tool—like web search—that can supplement your understanding of course content.

- Do not let AI replace your own understanding and thinking:
  - Do not have AI solve an assignment.
  - Do not submit AI-generated writing or code.
  - Do not treat AI output as automatically correct.

⚠ AI can help you study, but it cannot replace your work.

Scope of the Course

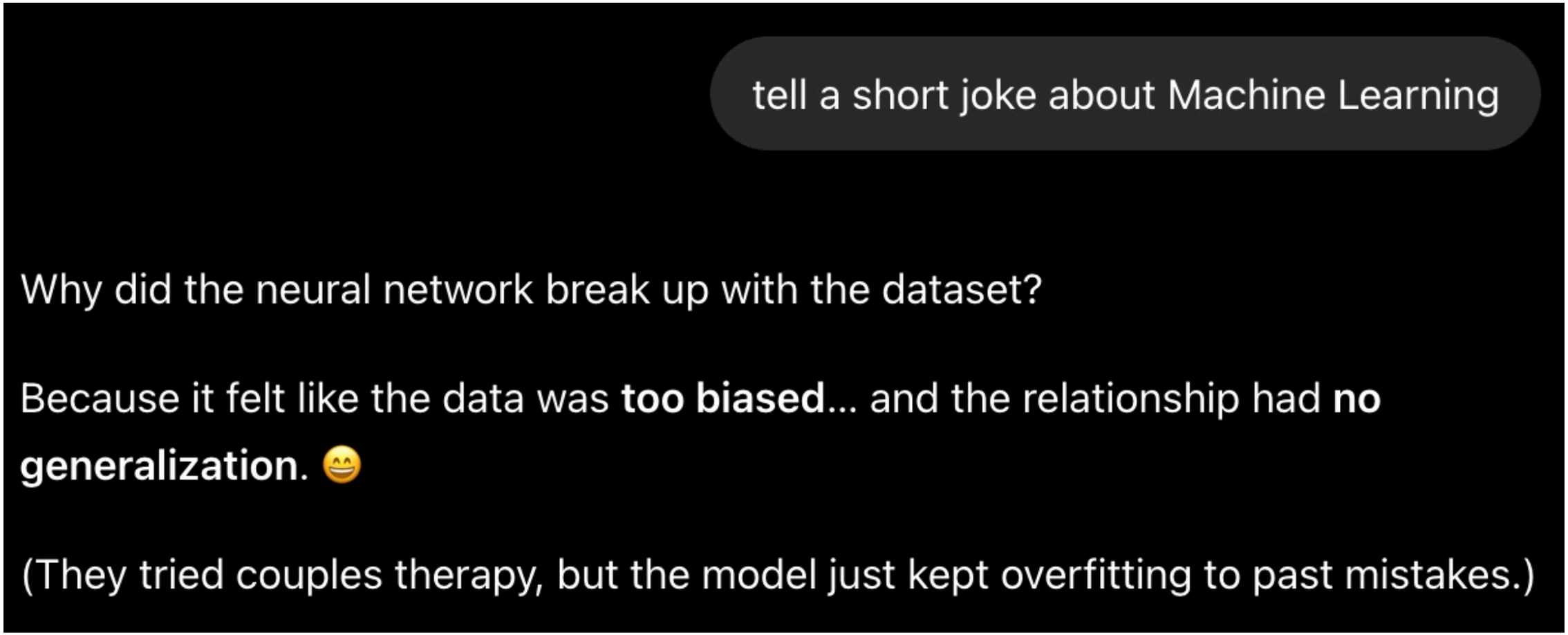
⚠ **Disclaimer:** This course is, to a large extent, about the mathematical background needed for machine learning.

- We cover foundational mathematics and concepts: probability, linear algebra, linear regression, classification, optimization, and model evaluation.
- We give a very light introduction to deep learning and language models.
- We also look briefly at the backbone of **ML systems engineering**: how automatic differentiation works. *(New this semester.)*

Introduction to Machine Learning

What is Machine Learning?

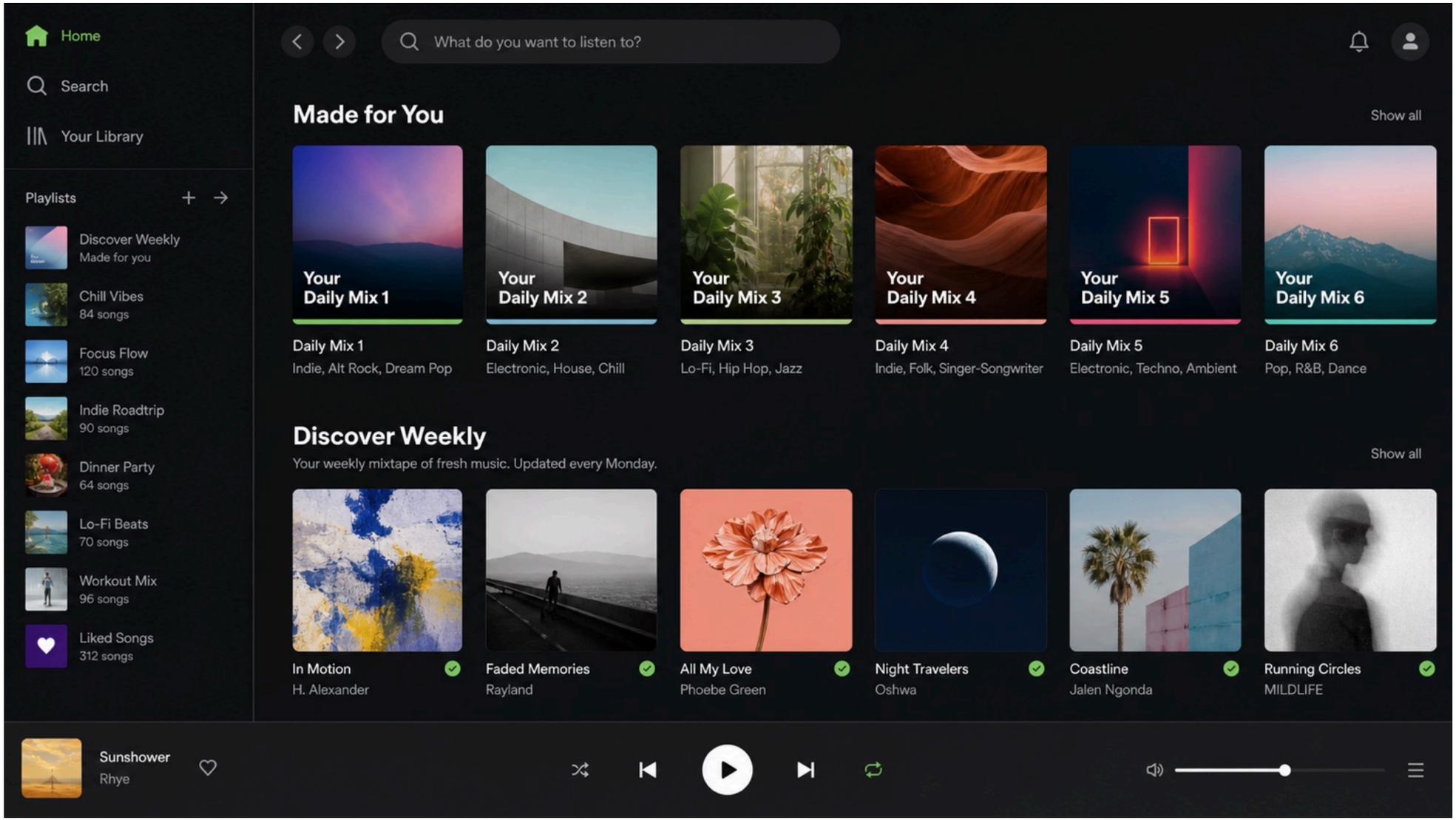
Systems Powered by ML



Systems Powered by ML

Prompt: “An armchair in the shape of an avocado.”

Systems Powered by ML



Informal Definition

A computer program/algorithm that learns from experience (and can be applied on new cases).

A (Slightly) More Formal Definition

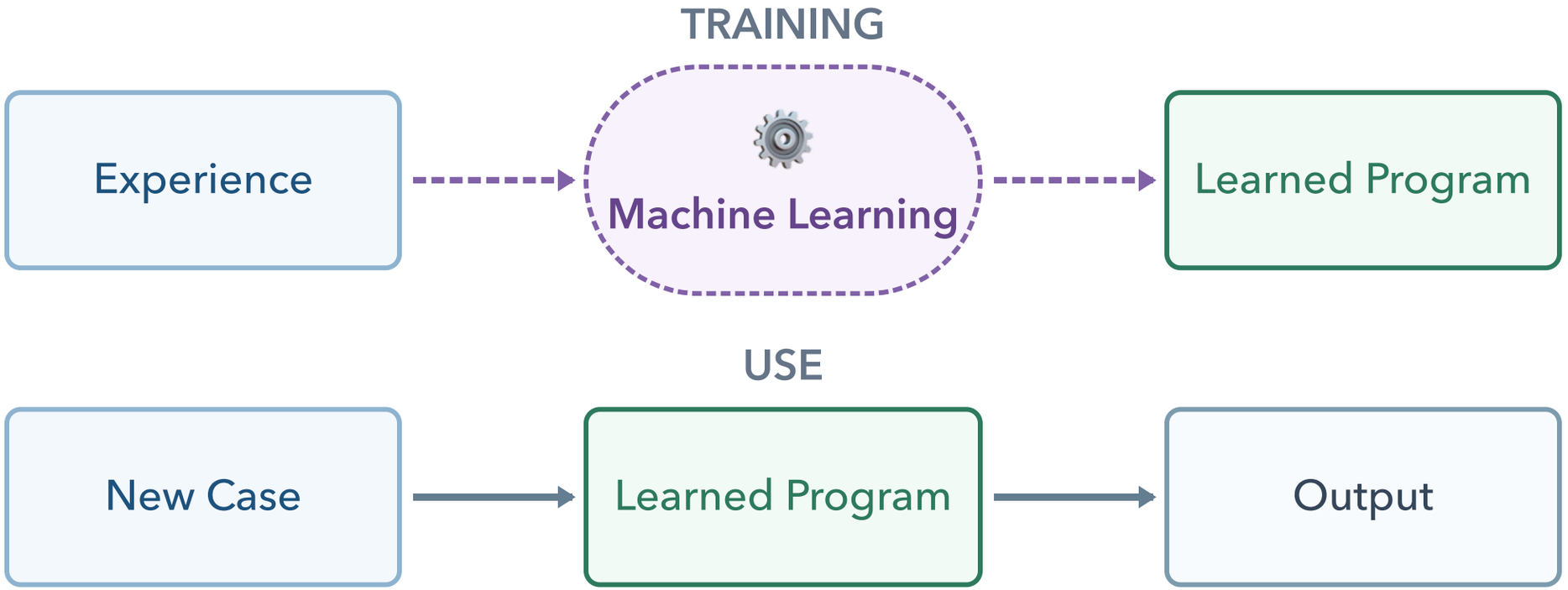
> A computer program learns from **experience** E, with respect to a **task** T and **performance measure** P, if its performance on T, as measured by P, improves with E. (Tom Mitchell, 1998)

E · **Experience**

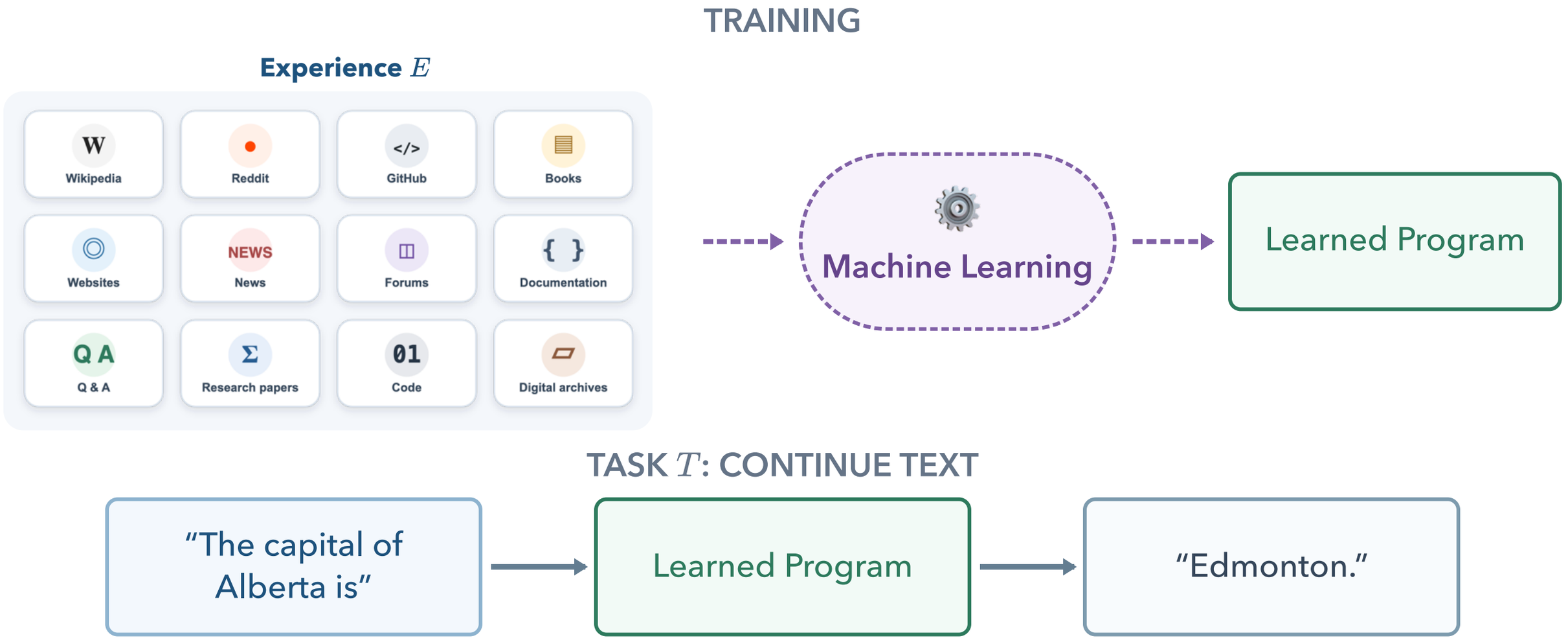
T · **Task**

P · **Performance**

Training and Using a Learned Program



Learning from Text



Learning from Image–Text Pairs

TRAINING

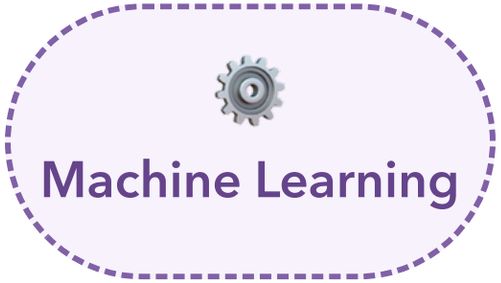
Experience E

"snow-covered mountains"

"an orange cat on a sofa"

"a red bicycle outside a café"

"a bowl of strawberries"



Learned Image Program

TASK T: GENERATE AN IMAGE

"An avocado-shaped armchair"

Learned Image Program

Learning Paradigms

Watch a cooking demonstration with the ingredients, steps, and finished dish shown.

Supervised Learning

Taste many dishes and organize them into groups with similar flavours and styles.

Unsupervised Learning

Cook, receive good or bad feedback or taste it yourself, and change what you do on the next attempt.

Reinforcement Learning

ⓘ This course focuses mainly on **supervised learning** (such as regression and classification).

Example Problem Setup

- Problem: estimate the traveling time to CCIS by bus.

8.2 km	8:10 a.m.	snow	heavy	12
Distance	Departure	Weather	Traffic	Stops

→

38 minutes

Input: trip features **Output:** observed travel time

Training Examples

Trip A

- 8.2 km
- 8:10 a.m.
- snow
- heavy traffic

38 min

observed travel time

Trip B

- 8.2 km
- 1:15 p.m.
- clear
- light traffic

24 min

observed travel time

Trip C

- 13.5 km
- 5:05 p.m.
- clear
- heavy traffic

47 min

observed travel time

ⓘ The experience E is a collection of past trips whose outcomes are known.

Rule-Based Programs

```
minutes = 8
minutes += 1.5 × distance

if rush_hour:
    minutes += 10
if snow:
    minutes += 12
if traffic is heavy:
    minutes += 8
if stops > 10:
    minutes += 5
```

A human chooses:

- which conditions matter
- every threshold
- every adjustment
- how conditions interact

This is a program.

Limits of Hand-Written Rules

Weather

light snow, freezing rain, temperature, visibility

Road conditions

construction, collision, lane closure, special event

Interactions

snow may matter differently at rush hour on different routes

More cases → more `if` statements → more exceptions → brittle rules

How to automatically learn these rules from data?

Parameterized Programs

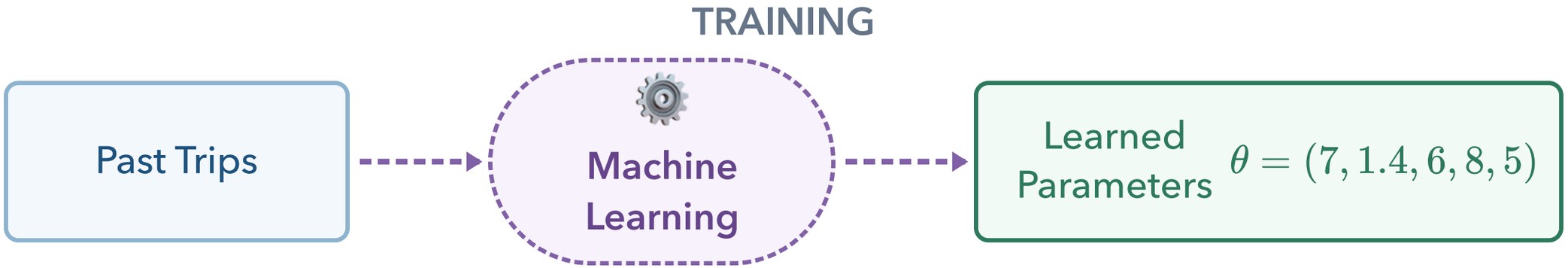
We specify a program family rather than every case:

$$\hat{t} = \theta_0 + \theta_1(\text{distance}) + \theta_2(\text{rush hour}) + \theta_3(\text{snow}) + \theta_4(\text{heavy traffic}).$$

θ_0	θ_1	$\theta_2, \theta_3, \theta_4$
Base Time	Distance Effect	Condition Effects

Humans choose the form and features. The adjustable parameters θ are not yet known.

Parameter Learning



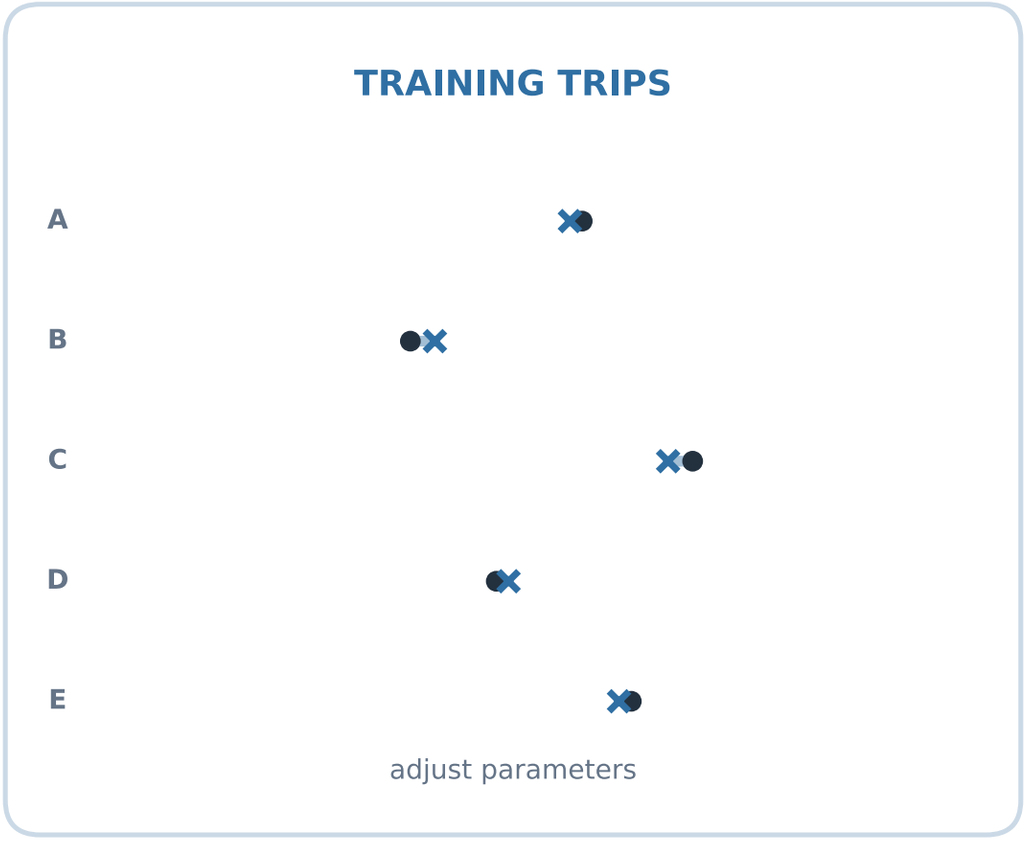
For the snowy 8.2 km morning trip:

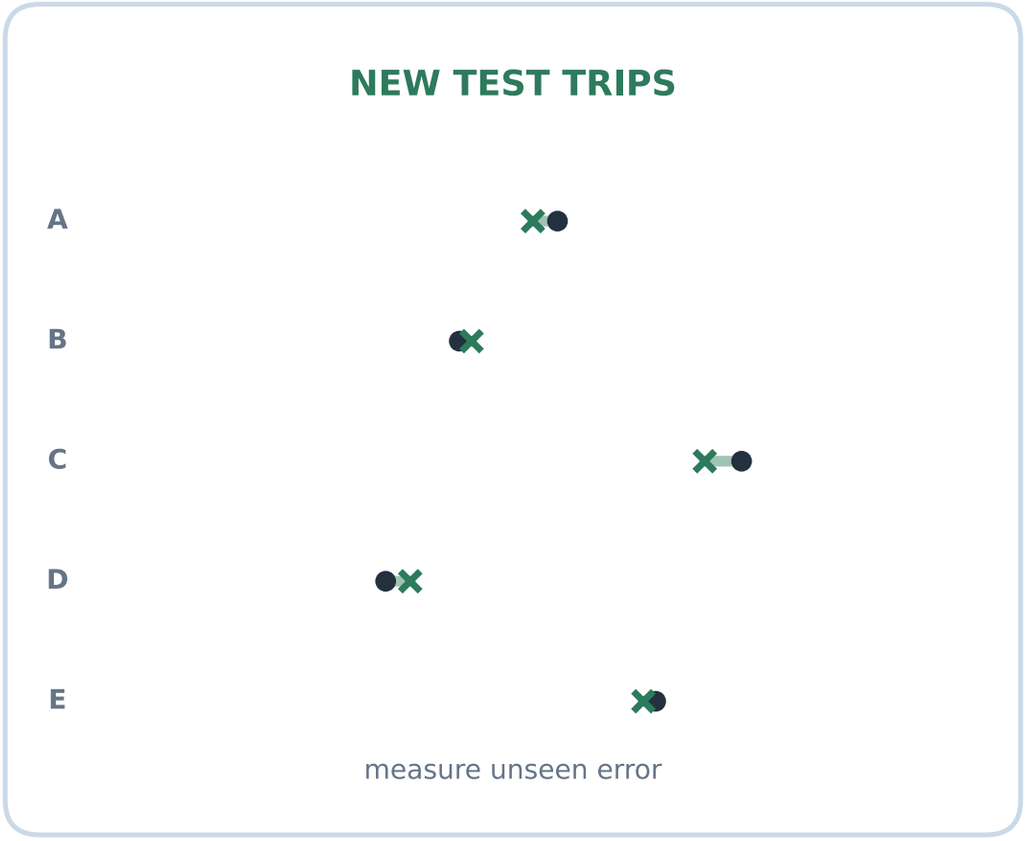
$$7 + 1.4(8.2) + 6 + 8 + 5$$

37.5 minutes

Observed: **38 minutes**

Training and Test Error





Use training error to adjust the parameters.

Use test error to evaluate the learned program.

● observed time × prediction line length = error

A learned program is useful when it has low **test error** on examples not used to choose its parameters.

Experience, Task, and Performance

> A computer program learns from **experience** E, with respect to a **task** T and **performance measure** P, if its performance on T, as measured by P, improves with E. (Tom Mitchell, 1998)

E · Experience

Past trips with features and observed travel times

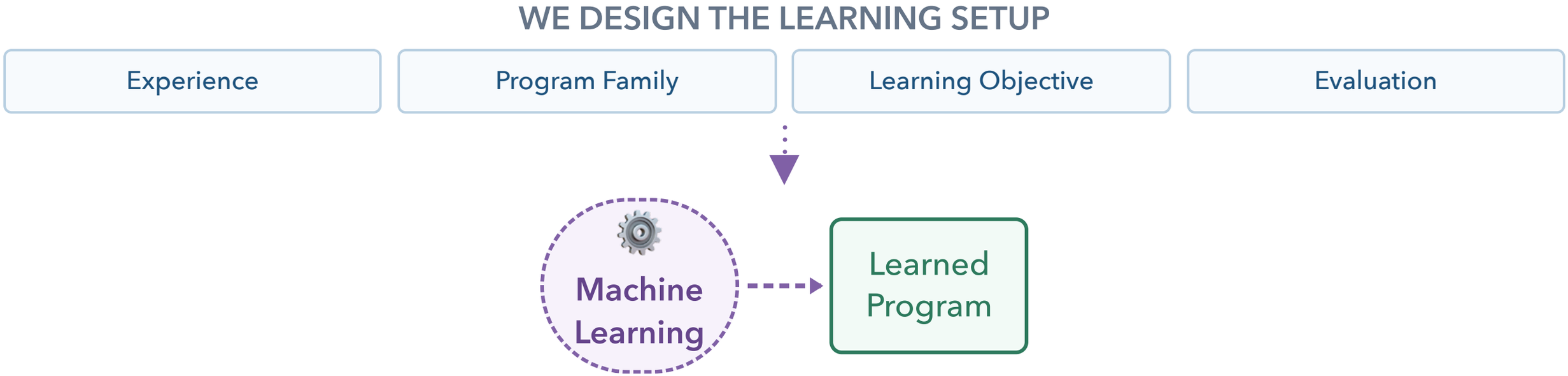
T · Task

Predict the travel time for a new trip

P · Performance

Low mean absolute error on unseen test trips

Our Jobs in Machine Learning



Pace of AI in Math

Pace of AI in Math

ASDiv · ACL 2020

Math Word Problem
A sandwich is priced at $0.75. A cup of pudding is priced at $0.25. Tim bought 2 sandwiches and 4 cups of pudding. How much money should Tim pay?
Solution: 0.75 x 2 + 0.25 x 4 = 2.5

Table 1: A math word problem

One short expression

Pace of AI in Math

GSM8K · 2021

Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?
Solution: Beth bakes 4 2 dozen batches of cookies for a total of 4*2 = <<4*2=8>>8 dozen cookies
There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of 12*8 = <<12*8=96>>96 cookies
She splits the 96 cookies equally amongst 16 people so they each eat 96/16 = <<96/16=6>>6 cookies
Final Answer: 6

Usually 2–8 reasoning steps

Pace of AI in Math

2020	2021	2025	2026
Simple arithmetic	Grade School Level Math	**Olympiad Gold**	Open Research

5 of 6 problems | 35 of 42 points | Officially graded at gold-medal level

Pace of AI in Math

Planar Point Sets with Many Unit Distances

OpenAI

Abstract

For a finite planar set P, let $\nu(P)$ be the number of unordered unit-distance pairs in P, and let $\nu(n)$ be the maximum of $\nu(P)$ over all n-point planar sets. We prove that, for some fixed $\delta > 0$, one has $\nu(n) \geq n^{1+\delta}$ for infinitely many n. This disproves the well-known unit distance conjecture from [Erd46].

The construction passes through an infinite unramified tower of totally real number fields with 3-power Galois groups of growing degree, in which a fixed set of rational primes splits completely. After adjoining i, these fields produce high-dimensional lattices with many elements whose images under every complex embedding have absolute value 1. Golod–Shafarevich theory ensures existence of an infinite such tower, even after a quotient step making the prescribed Frobenius classes trivial. A crucial property of the construction is that all resulting discriminants and class numbers are at most exponential in the extension degree.

1 Main Result

For a finite set $P \subset \mathbb{R}^2$, let $\nu(P) = \#\{\{x, y\} \subset P : |x - y| = 1\}$ and $\nu(n) = \max_{|P|=n} \nu(P)$. The planar unit-distance problem goes back to Erdős [Erd46], who conjectured that there should be an absolute constant C such that, for all sufficiently large n,

$$\nu(n) \leq n^{1+C/\log\log n}. \tag{1}$$

The same 1946 paper also introduced the distinct-distances problem, later resolved up to logarithmic factors by Guth–Katz [GK15]. An elementary upper bound is $\nu(n) = O(n^{3/2})$: the Euclidean unit-distance graph is $K_{2,3}$-free, since two unit circles meet in at most two points, and the Kővári–Sós–Turán theorem applies [KST54]. The best known upper bound $\nu(n) = O(n^{4/3})$ is due to Spencer–Szemerédi–Trotter [SST84]; Székely later gave a short crossing-number proof of this incidence bound [Sze97].

It is useful to compare this with the same question for other norms on $\mathbb{R}^2$, a direction studied systematically by Brass [Bra96]. If the unit ball has a flat segment, one can arrange quadratically many unit distances, so the interesting normed-plane analogues usually impose strict convexity or genericity. Székely's crossing-number argument extends to strictly convex norms, again giving $O(n^{4/3})$, and Valtr constructed a strictly convex norm for which this exponent is attained [Sze97, Val05]. Related work of Eisenbrand–Pach–Rothvoß–Sopher on convexly independent subsets of Minkowski sums gave the corresponding $O(m^{2/3}n^{2/3} + m + n)$ upper bound, with later matching lower bounds by Bilka–Buchin–Fulek–Kiyomi–Okamoto–Tanigawa–Tóth [EPRS08, BBF+10]; Brass–Moser–Pach give a broader survey [BMP05]. For Baire-generic norms the behavior is now known much more sharply. Matoušek proved $O(n \log n \log\log n)$ for

1

An open math problem. A conjecture with no progress made on it for nearly 80 years.

18 pages of proof.

Next lecture

math review & probability