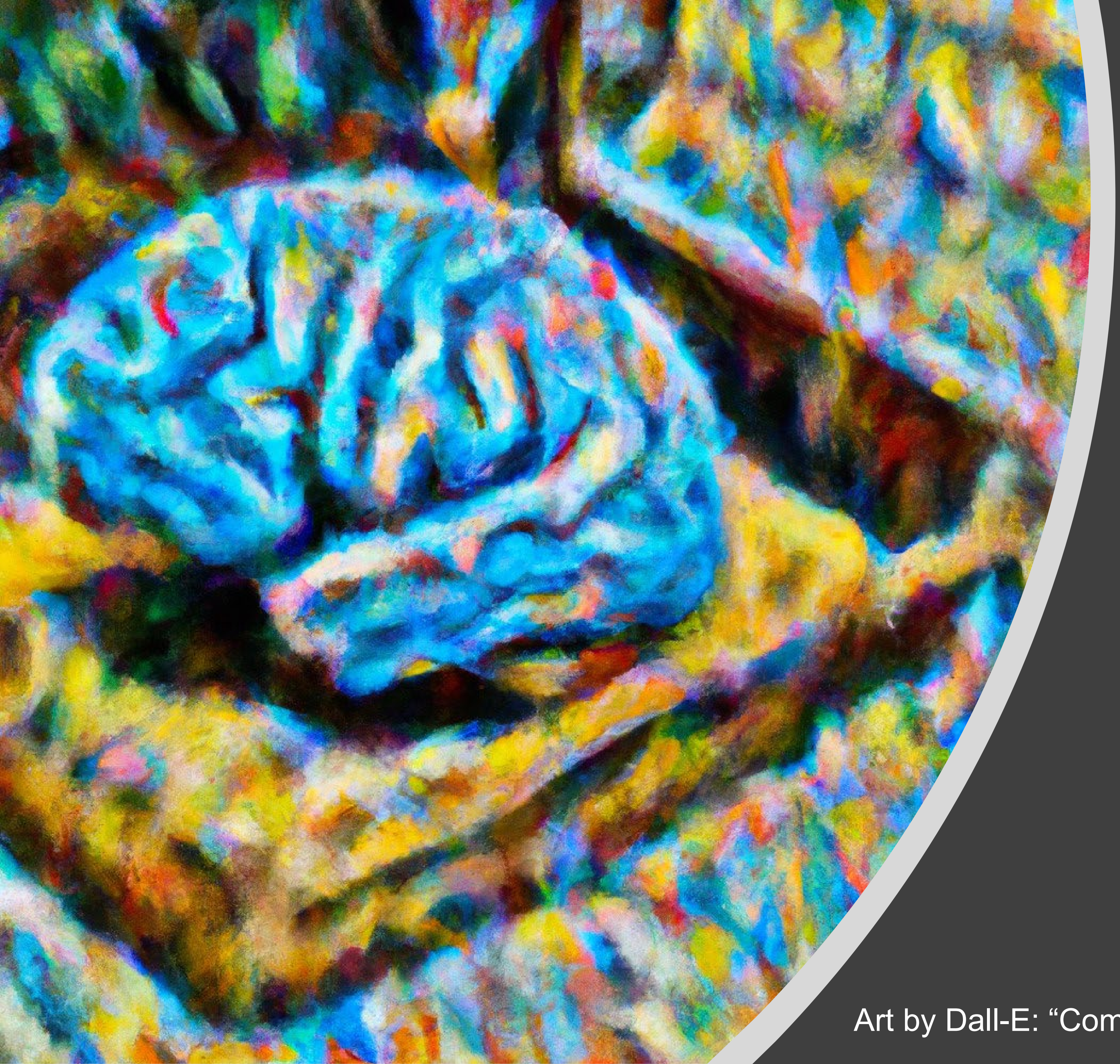


CS441 Applied Machine Learning

Instructor: Derek Hoiem

Art by Dall-E: “Computer brain gathering knowledge, impressionist”

Today’s Class

- A little about me
- Intro to Applied Machine Learning
- Course outline and logistics

About me

Raised in “upstate” NY

A little about me

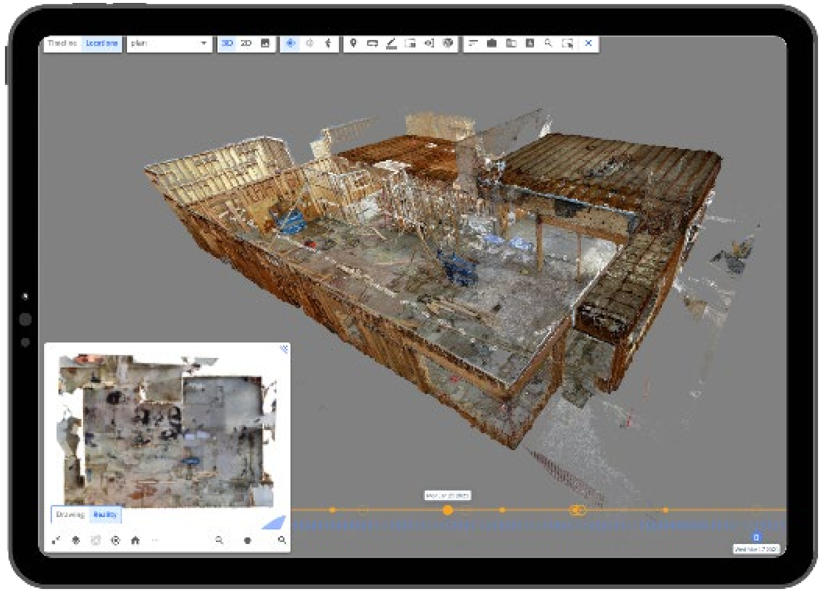
Undergrad at SUNY Buffalo
(EE, CE)

PhD in Robotics at Carnegie
Mellon

Professor in Computer Science
at University of Illinois

Reconstruct: vision for construction

Co-founder and Chief Science
Officer of Reconstruct



https://vimeo.com/242479887

https://www.reconstructinc.com/

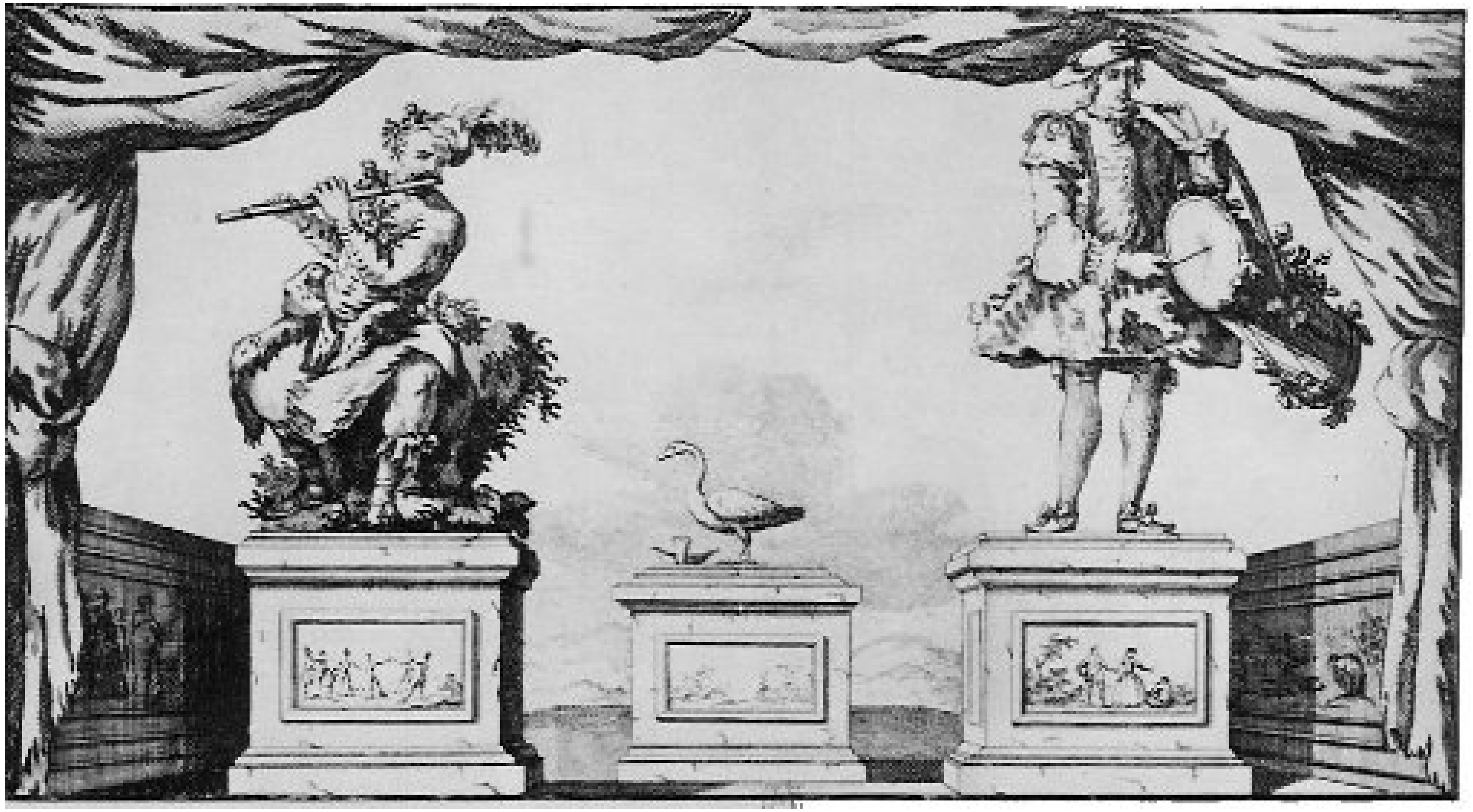
I ❤ ML

Machine learning and the quest for intelligent machines

Rabbi Loew's Golem (16th century)

By Mikoláš Aleš https://commons.wikimedia.org/w/index.php?curid=2210897

Vaucanson’s Automata (1738)

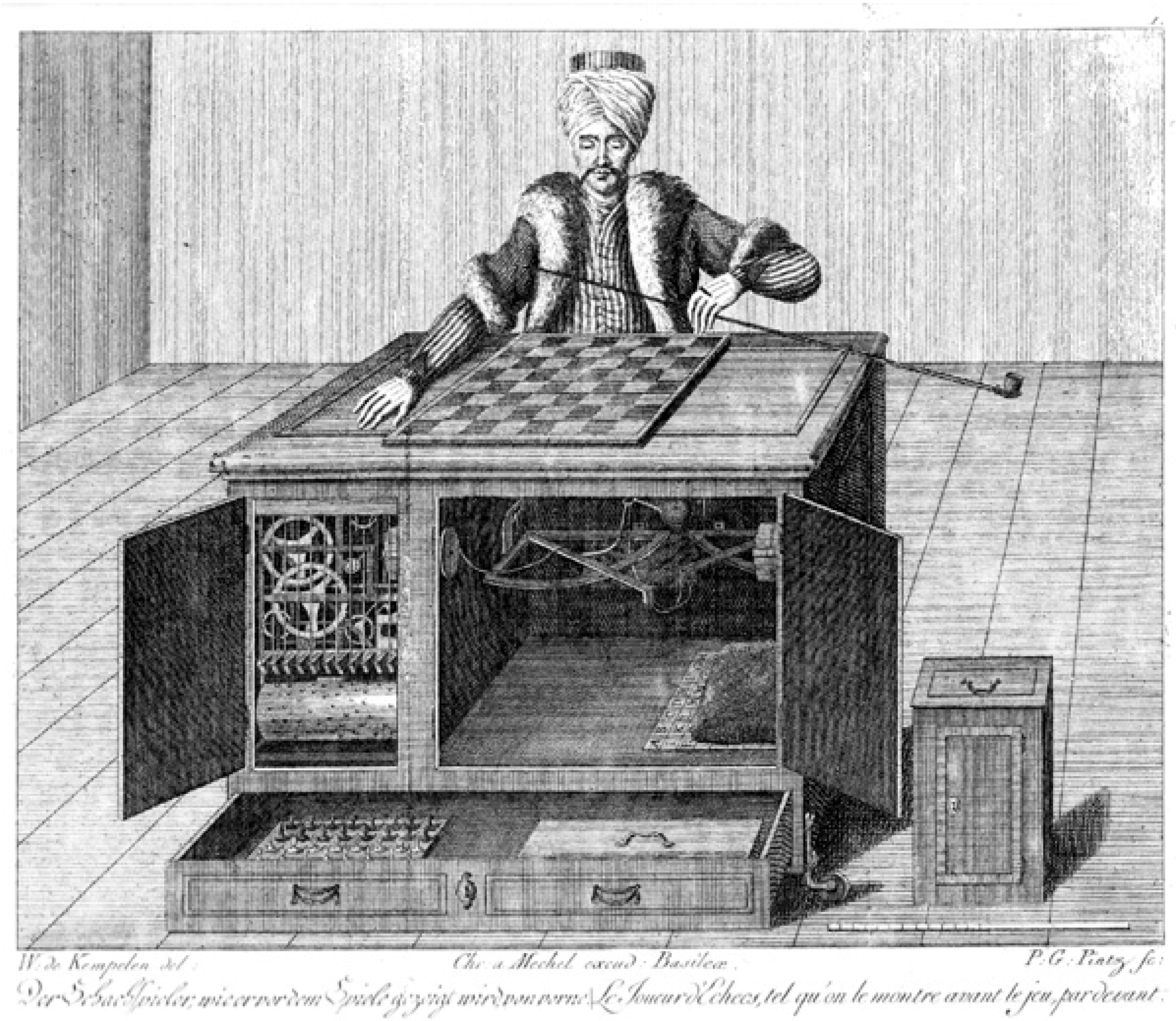


Flute player Defecating Duck Tambourine Player

http://www.francoisjunod.com/automates/eightennth/vaucanson_uk.htm

https://time.graphics/event/1736967

Mechanical Turk (1770)



Engraving of Mechanical Turk ([src](src))

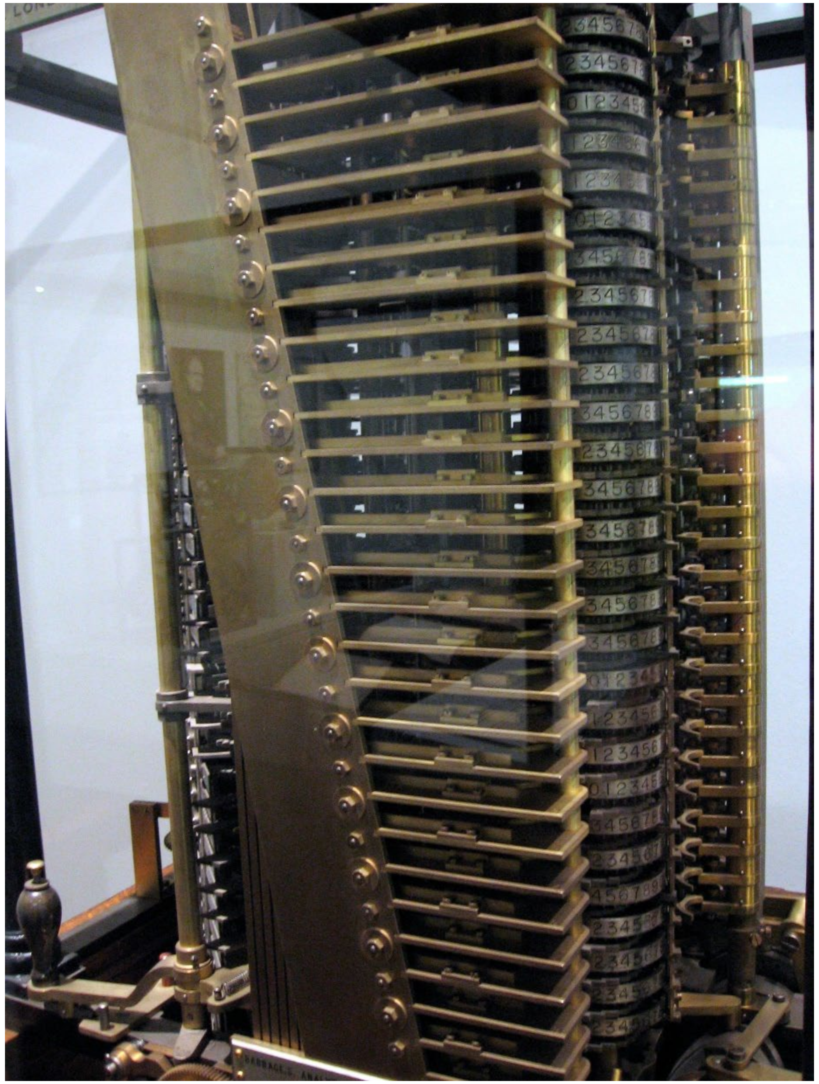
Secretly operated by a chess master, the mechanical turk was exhibited as an automatic chess playing machine.

Opponents included Benjamin Franklin and Napoleon Bonaparte.

Babbage's analytical engine (1837)

Not ever completely built

Ada Lovelace described a way to calculate Bernoulli numbers using the machine in 1843 (first computer program)



Analytical Engine Mill, built 1910 ([src](src))

Some AI milestones

- Alan Turing proposes the Imitation Game (1950)
- The Logic Theorist theorem prover (1955) – Newell and Simon
- The Perceptron - Rosenblatt (1957) (expected to lead to AGI)
- Eliza Chat Bot (1966) – Weizenbaum
- Deep Blue defeats Kasparov in Chess (1997)
- AlphaGo defeats Lee Sedol in Go (2016)
- ChatGPT released to public (2022)

Course goal: Know thyself

Our identities are largely characterized by what we learn.

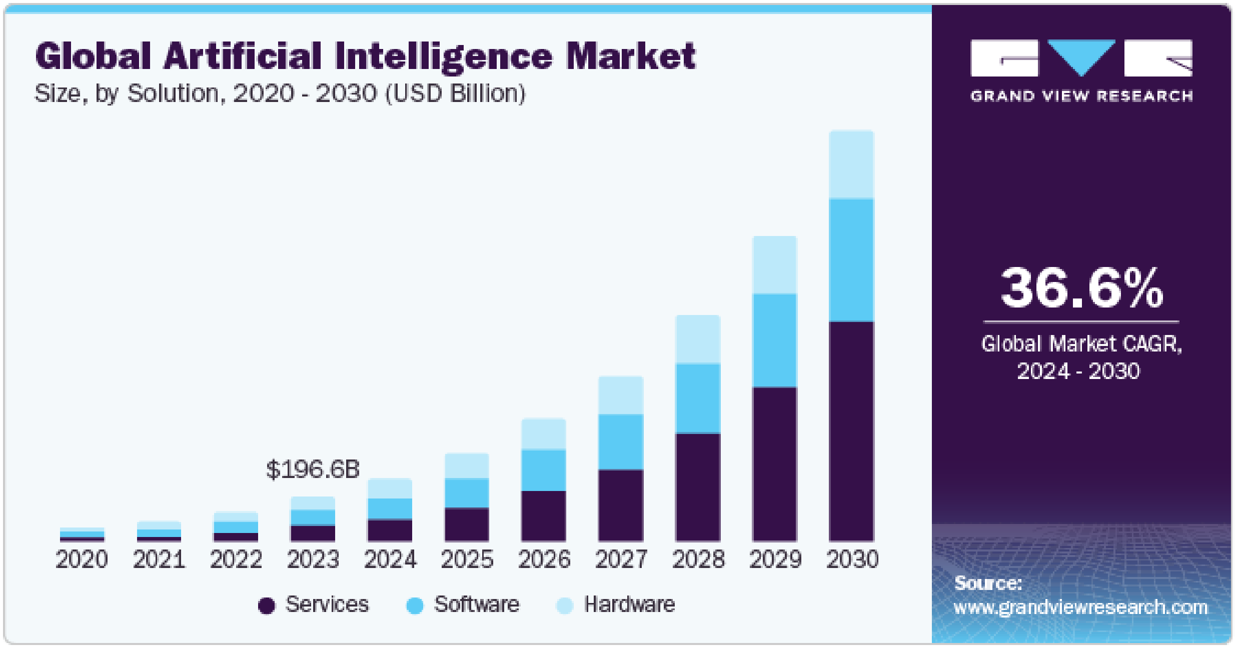
Making machines that learn can help understand our own learning.

Machine learning is also useful

- Unimate (1961) – first industrial robot
- Expert Systems (80’s) - first commercially successful AI for things like medical diagnosis
- USPS fully automated optical reader (1982)
- Tesla autonomous driving, Alexa, Siri, Copilot, …

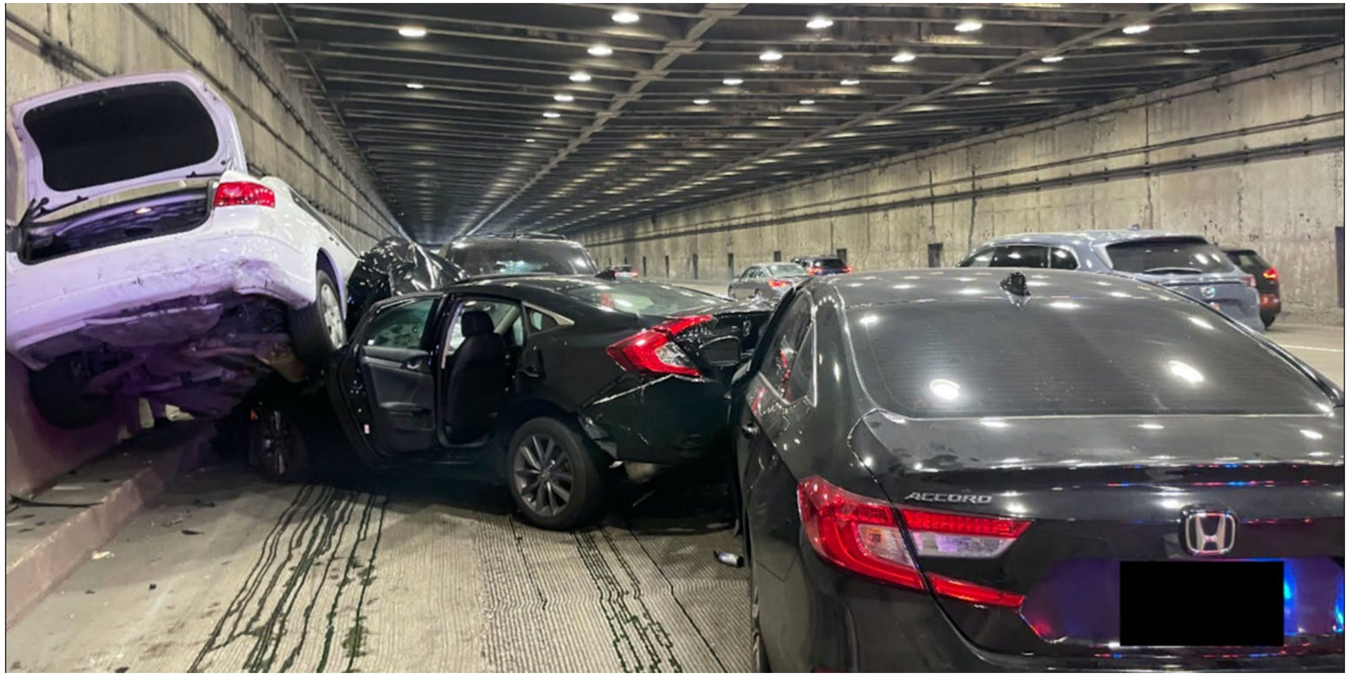
Course goal: Learn how to solve problems with ML

- Key concepts and methodologies for learning from data
- Algorithms and their strengths and limitations
- Domain-specific representations
- Ability to select the right tools for the job



Course goal: Understand real-life application and social implications of machine learning

- Recommending systems
- Surveillance
- Robots
- Smart assistants
- Text generation
- Autonomous cars
- Social media bots

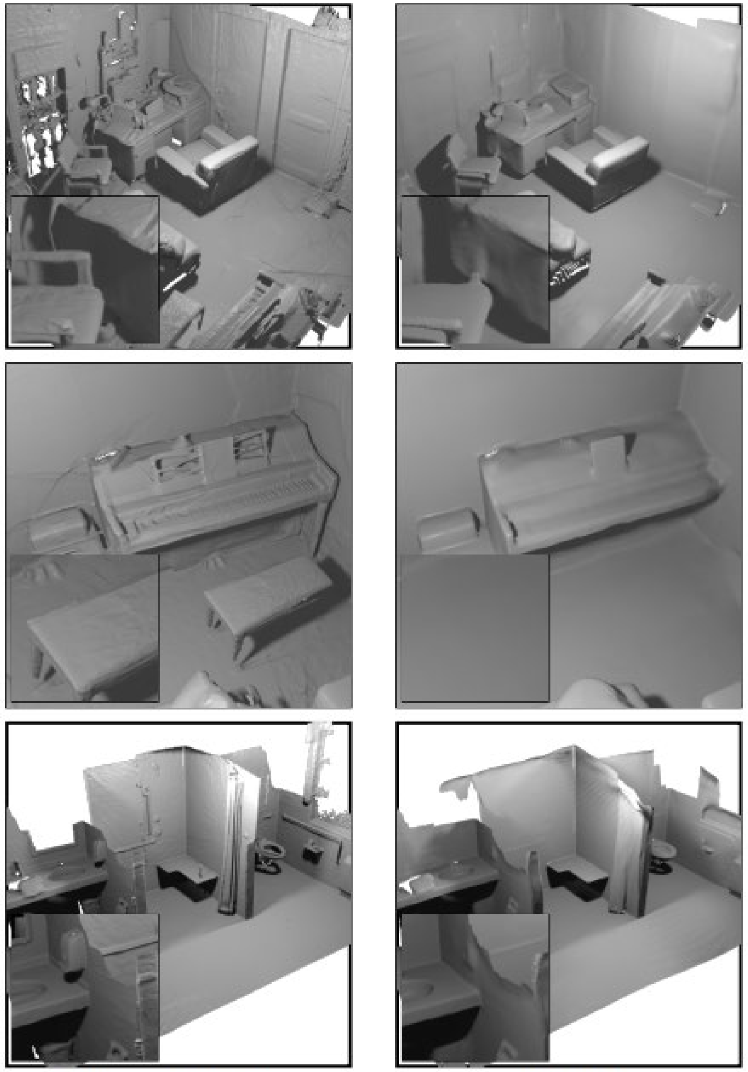


Tesla accident

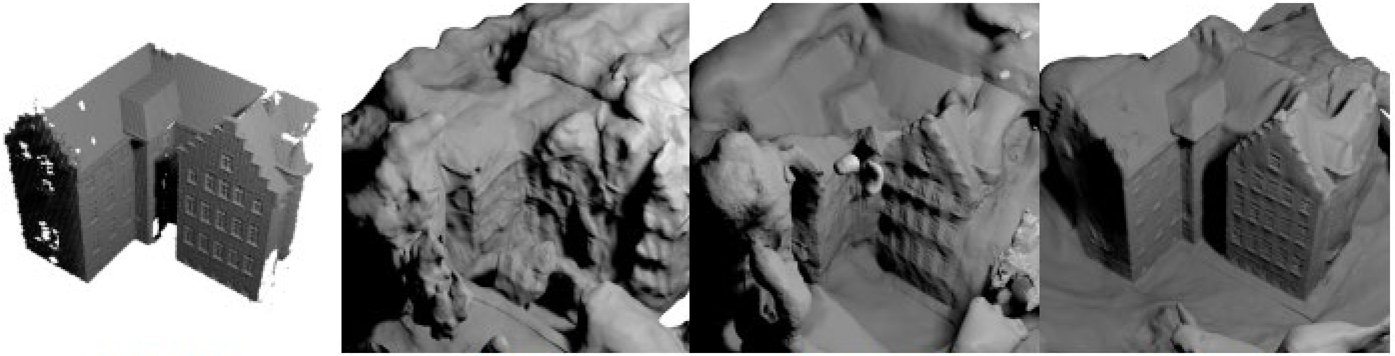
My early research: Learning to interpret geometry

Hoiem et al. 2005

Neural Radiance Fields: use deep networks to model 3D scenes



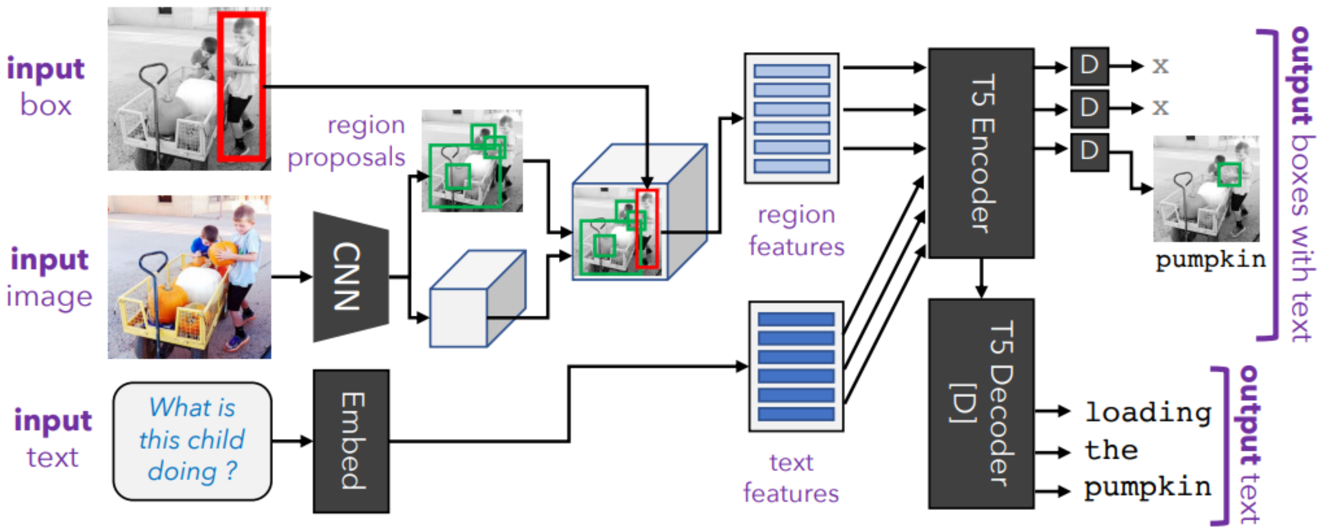
Ground Truth MLP

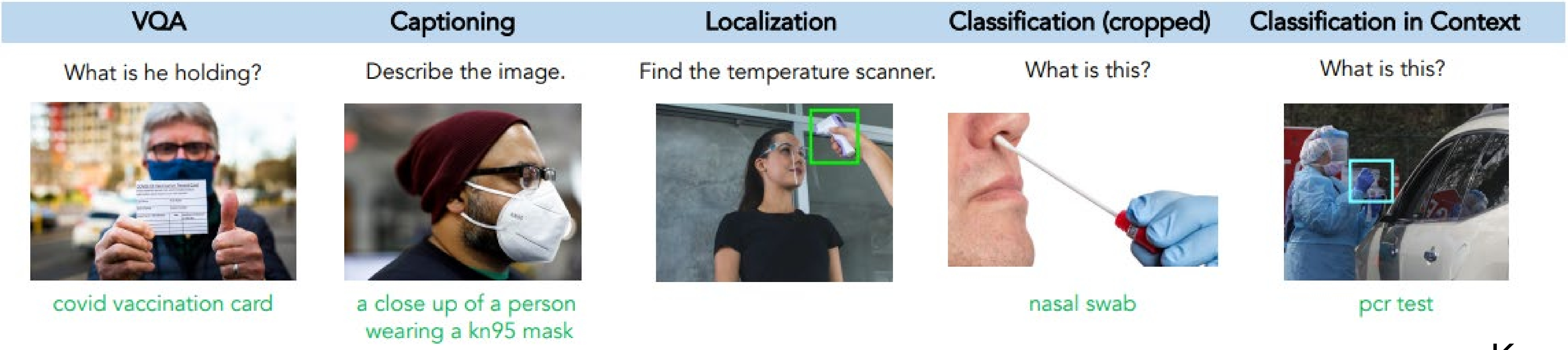


Ground Truth MLP [40] Multi-Res Grid [40] QFF-3D (Ours)

Lee et al. 2022

General Purpose Learners





Kamath et al. 2022

Other examples of my research that use machine learning

- Vision
  - Object detection
  - Image classification
  - Photo album organization
  - Image retrieval
  - Describing objects
  - 3D scene modeling
  - 3D object modeling
  - Robot navigation
  - Shadow detection and removal
  - Generating animations

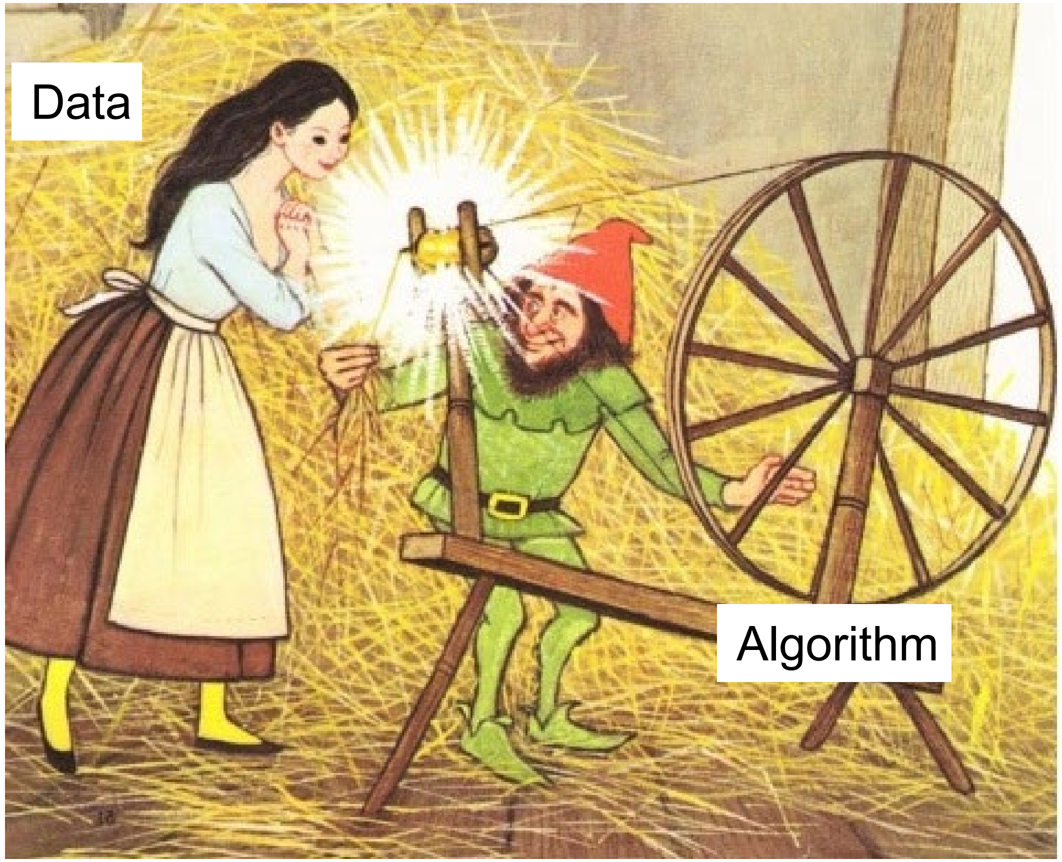
- Vision and Language
  - Visual question answering
  - Phrase grounding
  - Video analysis
  - General purpose vision-language

- Audio
  - Sound detection
  - Music identification

What is machine learning?

- Create predictive models or useful insights from raw data
  - Alexa speech recognition
  - Amazon product recommendations
  - Tesla autopilot
  - GPT-3 text generation
  - Image generation
  - Data visualization

- How? Select a model and solve for the parameters that optimize some objective in training data



ML spins raw data into gold!

The whole machine learning problem

Example: voice recognition in Alexa

1. Data preparation
   a. Collect and curate data
   b. Annotate the data (for supervised problems)
   c. Split your data into train, validation, and test sets

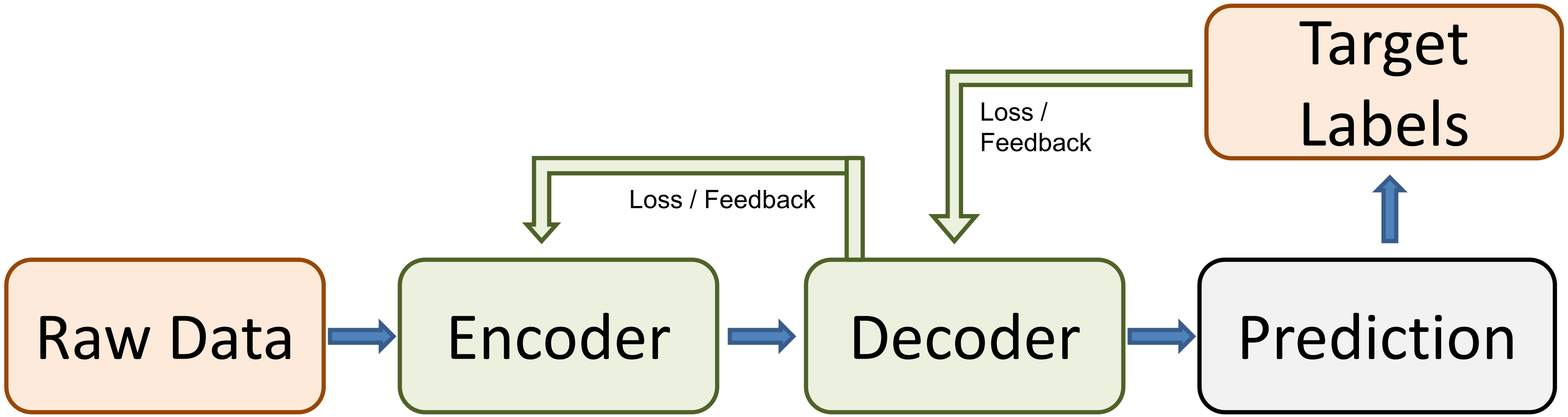
2. Algorithm and model development
   a. Design methods to extract features from the data
   b. Design a machine learning model and identify key parameters and loss
   c. Train, select parameters, and evaluate your designs using the validation set

Our focus, but it's important to understand all of it

3. Final evaluation using the test set

4. Integrate into your application

Algorithm and model development



Text
Images
Audio
Discrete/continuous values
Structured/unstructured
Few/many examples
Few/many features
Clean/noisy labels

Manual feature design
Deep networks
Trees
Clustering
Feature selection
Kernels
Density estimation

Linear regressor
SVN
Logistic regressor
Nearest Neighbor
Probabilistic model

Clusters
Low dimensional embedding
Category
Pixel labels
Generated text, image, audio
Positions
Yes/No

Course outline

Prof: Derek Hoiem dhoiem@illinois.edu

TAs

- Aakriti (aa117)
- Deema Alnuhait (deemaa2)
- Saharsh Barve (ssbarve2)
- Ashutosh Sharma (sharma96)

Website: https://courses.engr.illinois.edu/cs441/fa2024/ ⇦ **Bookmark this page!!**

Topics

- Fundamentals of learning
  - How to build classifiers and regressors based on provided features
  - Working with data, instance-based methods, linear models, probabilistic methods, trees
- Deep representation learning
  - How to learn effective representations
  - Optimization, MLPs, CNNs, transformers, vision, language, foundational models
- Applications
  - Ethics and impact, bias/fairness, building applications, RL, audio and time series

Date	Topic
Aug 27 (Tues)	Introduction
	Fundamentals of Learning
Aug 29 (Thurs)	K-NN Classification, Data Representation
Sep 3 (Tues)	K-NN Regression, Generalization
Sep 5 (Thurs)	Search and Clustering
Sep 9 (Mon)	Probability/Background Review (9pm)
Sep 10 (Tues)	Dimensionality reduction: PCA, embeddings
Sep 12 (Thurs)	Linear regression, regularization, bias-variance trade-off
Sep 16 (Mon)	*HW 1 (Instance-based Methods) due*
Sep 17 (Tues)	Linear classifiers: logistic regression, SVM
Sep 19 (Thurs)	Naïve Bayes Classifier
Sep 24 (Tues)	EM and Latent Variables
Sep 26 (Thurs)	Density estimation: MoG, Hists, KDE
Sep 30 (Mon)	*HW 2 (PCA and Linear Models) due*
Oct 1 (Tues)	Outliers and Robust Estimation
Oct 3-6	*Exam 1 at CBTF (Optional review on Oct 3)*
Oct 8 (Tues)	Decision Trees
Oct 10 (Thurs)	Ensembles and Random Forests
	Deep Learning
Oct 14 (Mon)	*HW 3 (PDFs and Outliers)*
Oct 15 (Tues)	Stochastic Gradient Descent
Oct 17 (Thurs)	MLPs and Backprop
Oct 22 (Tues)	CNNs and Keys to Deep Learning
Oct 24 (Thurs)	Deep Learning Optimization and Computer Vision
Oct 28 (Mon)	*HW 4 (Trees and MLPs) due*
Oct 29 (Tues)	Words and Attention
Oct 31 (Thurs)	Transformers in Language and Vision
Nov 5 (Tues)	Foundation Models: CLIP and GPT-3
Nov 7-10	*Exam 2 at CBTF (Optional review on Nov 7)*
	Applications
Nov 12 (Tues)	Ethics and Impact of AI
Nov 14 (Thurs)	Bias in AI, Fair ML
Nov 18 (Mon)	*HW 5 (Deep Learning and Applications) due*
Nov 19 (Tues)	Building and Deploying ML
Nov 21 (Thurs)	Audio and 1D Signals
Nov 23-Dec 1	Fall Break
Dec 3 (Tues)	Reinforcement Learning
Dec 5 (Thurs)	*Review, summary, looking forward*
Dec 6-10	*Exam 3 at CBTF*
Dec 15 (Sun)	*Final Project due (cannot be late)*

How will you learn?

Getting up to speed

- A solid understanding of linear algebra, probability, calculus, and data structures is assumed, as well as ability to learn Python
- Learn about Jupyter, Numpy, and linear algebra using the provided tutorials
- I will do a recorded session on probability on Sep 9

How will you learn?

Date	Topic	Link	Reading/Notes
Aug 27 (Tues)	Introduction		Jupyter notebook tutorial vid ipynb cc Numpy tutorial vid cc Linear algebra tutorial vid cc
	Fundamentals of Learning		
Aug 29 (Thurs)	K-NN Classification, Data Representation		AML Ch 1.1-1.2
Sep 3 (Tues)	K-NN Regression, Generalization		AML Ch 1.1-1.2

Lectures and Reading

- Lecture attendance is expected
- Reading materials and notes are optional
- Lectures will be recorded, and slides and recordings will be available
- If a technical problem prevents recording, only slides will be available

How will you learn?

Assignments

Conduct experiments in Jupyter notebook, report results, answer questions to test your understanding

- HW 1: Instance-based methods (Sep 16)
  - Retrieval, clustering, KNN classification and regression
- HW 2: Linear Models (Sep 30)
  - PCA and embeddings, linear regression, logistics regression, SVM
- HW 3: Probabilistic Methods (Oct 14)
  - Naïve Bayes, EM, robust estimation
- HW 4: Trees and MLPs (Oct 28)
  - Decision trees, random forests, boosting, multi-layer perceptrons
- HW 5: Deep learning and applications (Nov 18)
  - Linear probe, fine-tuning, investigating a real-world application area
- Final Project (Dec 5): Apply ML, compare algorithms, select parameters on a problem of your choosing

How will you learn?

Getting Help Outside of Class

1. For questions that have short answers, see if it’s been answered on CampusWire; if not, post
   - We try to answer within 24 hours (our schedule to check will be posted)

2. For more in-depth help, or to better understand the material, come to office hours
   - Schedule will be posted on CampusWire

How will you learn?

Exams

- Three 50-minute exams to test conceptual knowledge
- Exams will be held in CBTF (closed book/notes)
- Cumulative

Grading

Experience Points: 500+ points

- 5 homeworks: 100+ points each
- 1 final project: 100+ points
- Participation: 2 points per opportunity (up to ~30 opportunities)
- 3 credit: target is 500 points
- 4 credit: target is 625 points

Exams: 200 pts

- Three 50 minute exams, at CBTF
- Lowest exam score is dropped

Final grade calculation

Course_Grade = (EP + highest_exam + 2nd_highest_exam) / (max(EP, EP_target) + 200)

Late policy

- Up to ten free days total – use them wisely!
- 5 point penalty per day after that
- Project must be submitted within two weeks of due date to receive any points

Earn 2 points now – tell me a little about you

https://tinyurl.com/441-fa24-L1

Covid, masks, sickness

- If you’re well, please come to lectures and office hours. Masks are optional.

- If you’re sick, please stay home. No need to show proof of illness or get permission to miss.

Academic Integrity

These are OK

- Discuss homeworks with classmates (don’t show each other code)
- Use Stack Overflow to learn how to use a Python module
- Use GPT/Co-Pilot/Gemini etc to learn how to use a Python module, or streamline coding
- Get ideas from online (make sure to attribute the source)

Not OK

- Copying or looking at homework-specific code (i.e. so that you claim credit for part of an assignment based on code that you didn’t write)
- Using external resources (code, ideas, data) without acknowledging them

Remember

- Ask if you’re not sure if it’s ok
- You are safe as long as you acknowledge all of your sources of inspiration, code, etc. in your write-up
- If you use an AI tool, acknowledge it

How is this course different from…

- CS 446 ML
  - This course provides a foundation for ML practice, while 446 provides a foundation for ML research
  - This course has less theory, derivations, and optimization, and more on application, representations, and examples
- Online version of CS 441 AML
  - CS 441 online is being updated to be more like this one
  - This course focuses more on concepts and modern usage of ML, homeworks require more independence
- CS 444 Deep Learning for CV, other domain-oriented courses
  - This course is much broader

What is expected of you?

- You have sufficient background in math and CS, and/or will work hard to catch up in the first couple weeks
- You will attend all lectures (preferred), or watch the recordings
- If you get stuck, you'll try your best to figure it out, but then seek help if you can't
- You are willing to spend ~10-12 hours per week on lectures, reading, review and assignments

Feedback is welcome

- I will occasionally solicit feedback through surveys – please respond
- You can always talk to me after class or send me a message on CampusWire
- My goal is to be a force multiplier on how much you can learn with a given amount of effort

What to do next

- **Bookmark the website**
- **Sign up for campuswire** (if not already signed up)
- **Read the syllabus and schedule**
- Unless you consider yourself highly proficient in Python/numpy and linear algebra, **watch/do the tutorials** linked in the web page
- If you are planning to drop the class, please do it right away so that others can register