Intro and problem definition

10-701 Introduction to Machine Learning

Geoff Gordon and Pradeep Ravikumar

Logistics

old link! …/~10701-f26/

- https://www.cs.cmu.edu/~10701-f25/
  - This whole website is required reading
- Highlights:
  - Tech: Piazza, Gradescope, Canvas
  - Grades: homeworks (written and coding parts), homework quizzes (after due date, check understanding), participation (small questions, 48 hrs to complete), midterm & final exams, class project
  - Prereqs: probability, coding, calculus, reasoning/proofs
  - Grace days for homeworks: no need to ask us about small illnesses, trips, etc.
    - applied greedily, limit on total lateness
    - separate from major problems (e.g., longer-lasting illness)

Collaboration

- Is encouraged and can support learning!
- But write up solutions ***on your own***
  - don’t discuss full concrete solutions, just strategies and ideas
  - maybe best not to take detailed notes during discussions, but if you have notes, put them away, wait 10 minutes before writing anything
  - ensures solution isn’t coming from short-term memory
- Make sure to ***protect your own work***
- And ***disclose all collaborators and sources***
- We do check, and we regularly find problems — see syllabus about academic integrity violations

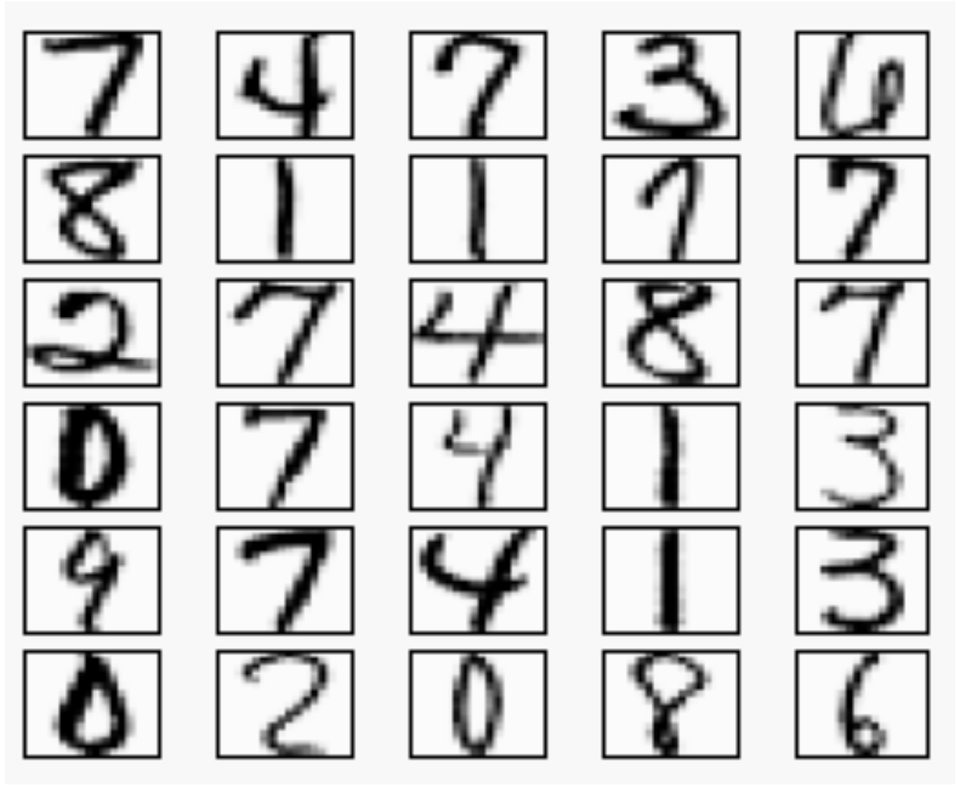
GenAI

- You are encouraged to use GenAI to improve learning!
  - e.g., find literature, generate practice problems, discuss strategy — let us know if you find an interesting and helpful study practice!
- All work that you submit must be your own
- Treat GenAI as if it were another student:
  - interactions should support learning
  - do not discuss complete solutions
  - best not to take written notes during discussion
  - all notes should be closed while you are writing solutions
  - this includes turning off large-scale code completion: you will have trouble with homework quizzes if you yourself don’t learn what the libraries do (and later you will have trouble with understanding and debugging code)

What is ML?

- Code that gets better at some ***task*** by taking advantage of some kind of ***experience***
  - Task is formalized with a performance ***metric***
  - Experience is formalized with ***data***

For example



→ 7

- Task: read handwritten ZIP code on mail
- Experience: mail that has been correctly routed
- Becomes:
  - data: images of individual digits w/ corresponding labels
  - metric: per-digit accuracy (%)

Another example

- Learning whether to approve a loan or line of credit
  - Task:

    Decide whether to approve a loan
  - Experience:

    Records: applications w/ decisions
  - Data:

    Features we extract from applications
  - Metric:

    % accuracy

No single answer

- Often the same/similar ML problem can be formulated several ways
  - previous slide's metric: accuracy of approval decision (classification)
  - could also predict a numerical credit score (regression)
  - or distribution of total amount repaid (conditional density estimation)
  - or each loan payment over time (sequence learning)
  - …

No single answer

- Often the same/similar ML problem can be formulated several ways
  - prev… sion
    (clas…
  - coul… ssion)
  - or di… density
    estir…
  - or ea… ng)
  - …

Output type	**Name**
boolean	Binary classification
categorical	Multiclass classification
real	Regression
ordering	Ranking
multiple related	Structured prediction (e.g., graphical models)
multiple over time	Sequence learning
policy / control	Reinforcement learning
…	…

Self-driving car

What is our experience? How do we turn it into data?

prior driving behavior → GPS, speed, lidar, camera, IMU, external temp, weather, steering wheel, brake, gas

processed versions of ↑

What is our task? How do we measure performance?

Safe driving, fast progress, gas/elec consumption

few interventions, robustness
- ↳ weather
- ↳ road type

→ # pedestrians hit

following laws / few tickets

impact on non-users

few protests

this one is a freebie: no penalty if you didn't bring a device

Participation question

- Define your own learning problem (any one you find interesting) in terms of T, E, D, P

https://forms.gle/5oqJA7VrzmAh9S5C8

Your well-posed ML problems

Task

Face unlock
Disease diagnosis from scans
Play Super Mario
Recipe planning

Experience

Successful unlocks
CT scans and diagnoses
Pro speedruns
Meals made by human cooks

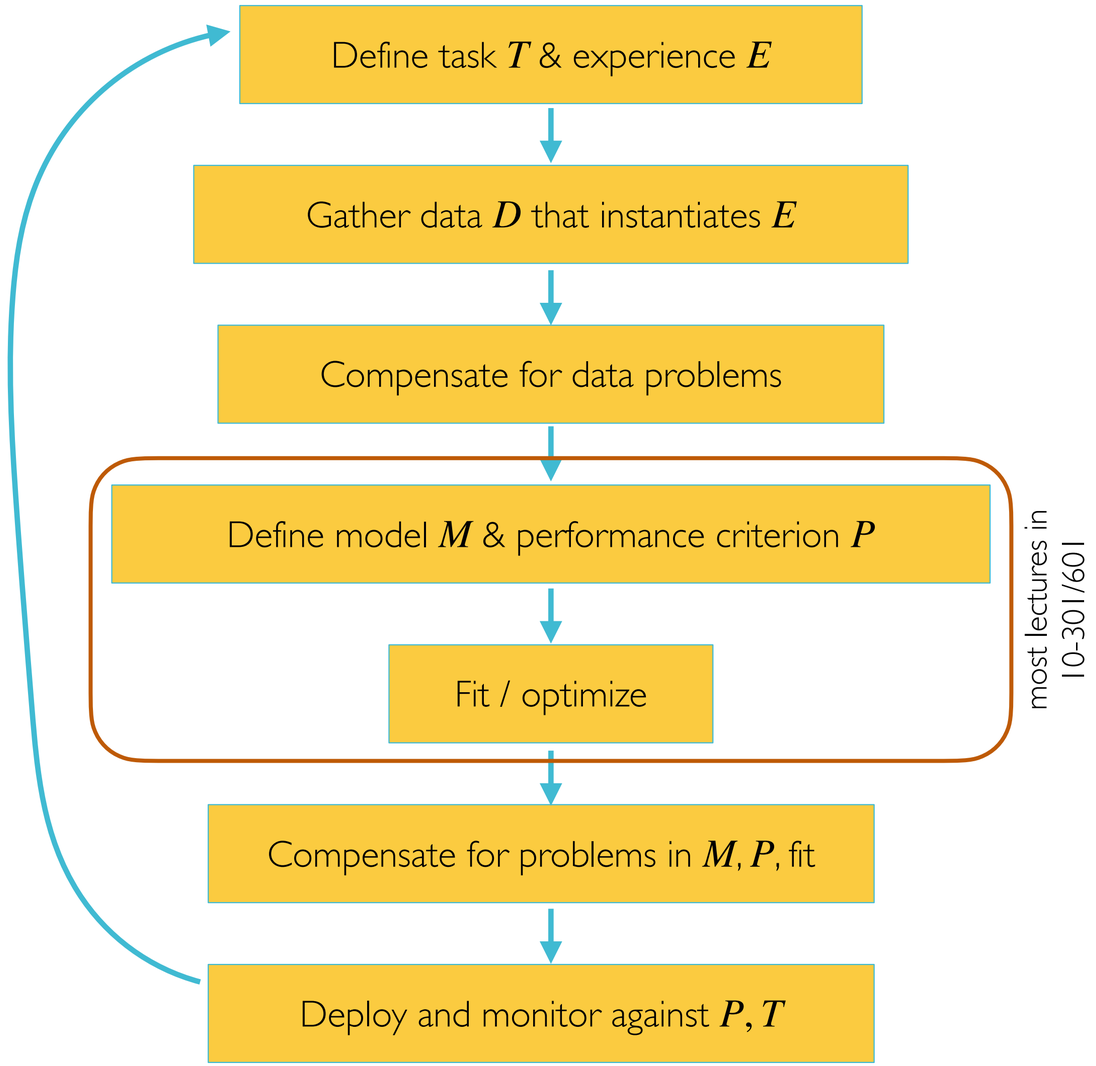
Performance metric

Error rate
Error rate / cost of mistreatment
Time of run, coins collected, …
Nutrition, taste, cost, dietary restrictions, do I have ingredients

Data

Image/3d scan of face, label of who
Pixels of scan, patient demographic and biometric features
Video, audio, and controller input from speedrun
Refrigerator contents -> human menus

ML pipeline



what do we want to accomplish?

define input and output representations

distribution shift, admixture (outliers), mislabeling, …

how do we represent our hypothesis? how do we measure success?

hyperparameters, regularization, …

model mismatch, distribution shift, overfitting, bad local optima, safety, laws, business rules …

we never get it right the first time — learner does what we tell it, not what we should have told it!

A brief history of machine learning

- Early Foundations (1940s–1960s)
  - 1957: Rosenblatt develops the perceptron, an early neural network for classification
- Symbolic AI & the First AI Winter (1970s–1980s)
  - Limitations of perceptrons (Minsky & Papert, 1969) and lack of computing power lead to skepticism and reduced funding
- Statistical & Algorithmic Advances (1980s–1990s)
  - 1986: Rumelhart, Hinton & Williams popularize backpropagation, enabling multi-layer neural networks to learn
  - 1980s–90s: Emergence of support vector machines (SVMs), decision trees, boosting (AdaBoost), Bayesian methods, Reinforcement learning

A brief history of machine learning

- Second AI Winter, Rise of Data (1990s–2000s)
  - Disillusionment with difficulty of training deep networks and problems with scaling RL, eventually overcome by…
  - Explosion of digital data + faster computing power
  - Kernel methods, ensemble methods come into their own
- Deep Learning Revolution (2010s)
  - 2012: AlexNet (Krizhevsky, Sutskever, Hinton) wins ImageNet competition w/ GPU + deep CNNs, igniting deep learning boom
  - Reinforcement learning breakthroughs (e.g., AlphaGo in 2016)
- Foundation Models & Generative AI (2020s–present)
  - Rise of transformers (Vaswani et al., 2017) revolutionizes NLP (BERT, GPT)
  - Emergence of foundation models trained on massive datasets for general-purpose use
  - Policy, ethics, and responsible AI practices gain prominence due to societal impacts

Classical and modern models

- Huge amount of change over time in the types of problems and types of models that people focus on
- But, some ideas and techniques span the whole history
  - how to set up problems, useful mathematical ideas, relationship to optimization, even just overall ML mindset
- One of our goals this course: expose these common ideas so that you can develop new models & techniques and be part of whatever the next revolution is
  - simple example: linear models vs. last layer of deep net

Techniques

- Some of the common techniques:
  - linear algebra
  - differential calculus
  - optimization (e.g., SGD)
  - hypothesis classes, complexity, regularization
  - tensor manipulation
  - function spaces
  - …

Driving (some of the) differences: how much data, compute

- Data:
  - deep nets can get very high performance with big data, can lose badly to classical methods like feature engineering + logistic regression if we have small data
  - kernel methods are intermediate (may need more data than simple classical models, but don't seem to be SOTA on huge data)
  - small data still happens
    - e.g., if expensive to collect
    - e.g., long tail

Driving (some of the) differences: how much data, compute

- Compute:
  - kernel methods can be prohibitively expensive on big data (there are approximations that can trade off accuracy)
  - deep nets are driving a huge boom in construction of data centers (and corresponding electricity use)
  - small compute still happens (e.g., phones, sensors)
- Extreme compute limitation: models that humans can run in their heads (e.g., MMSE for evaluating cognitive impairment)
  - for interpretability, trust, unplugged use

Related areas

- Data science: Extracting knowledge/insights from noisy, unstructured data
- Statistics: Understanding how to draw verifiable conclusions from observable evidence
- Artificial intelligence: Creating machines that can mimic human behavior/cognition
- Computational neuroscience: Understanding how learning (and many other things) are implemented in the brain
- …

Our first ML task

- Learning to diagnose heart disease (T)
  - as a **(supervised) binary classification task** (E)

features attributes regressors

labels target

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

data points / instances

$\leftarrow D$

define: instances, attributes, ground truth

Our first ML task

- Learning to diagnose heart disease (T)
  - as a **(supervised) binary classification task** (E)

our goal: a ***classifier*** h : features → labels

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

data points

$\leftarrow D$

e.g. want h(No, Medium, Normal) = No

Our first ML task

- Learning to diagnose heart disease (T)
  - as a **(<u>supervised</u>) binary classification task** (E)

our goal: a ***classifier*** h : features → labels

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

$\leftarrow D$

Our first ML task

- Learning to diagnose heart disease (T)
  - as a **(supervised) <u>binary classification</u> task** (E)

our goal: a ***classifier*** h : features → labels

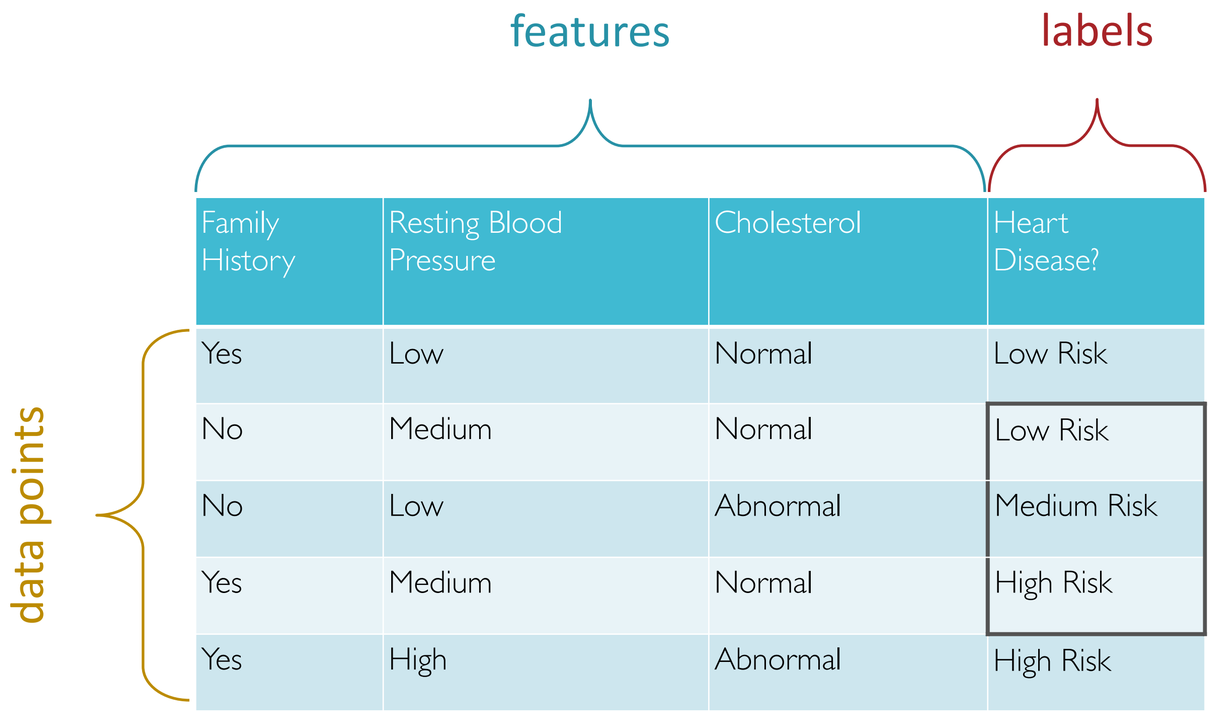
data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

$\leftarrow D$

Our first ML task

- Learning to diagnose heart disease (T)
  - as a **(supervised)** **<u>classification</u> task** (E)



Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	Low Risk
No	Medium	Normal	Low Risk
No	Low	Abnormal	Medium Risk
Yes	Medium	Normal	High Risk
Yes	High	Abnormal	High Risk

$\leftarrow D$

Our first ML task

- Learning to diagnose heart disease (T)
  - as a **(supervised) <u>regression</u> task** (E)

features

labels

data points

Family History	Resting Blood Pressure	Cholesterol	Medical cost from heart disease?
Yes	Low	Normal	$0k
No	Medium	Normal	$0k
No	Low	Abnormal	$10k
Yes	Medium	Normal	$17k
Yes	High	Abnormal	$23k

$\leftarrow D$

Our first ML method

- Majority vote classifier
  - for a **(supervised) classification task**

features

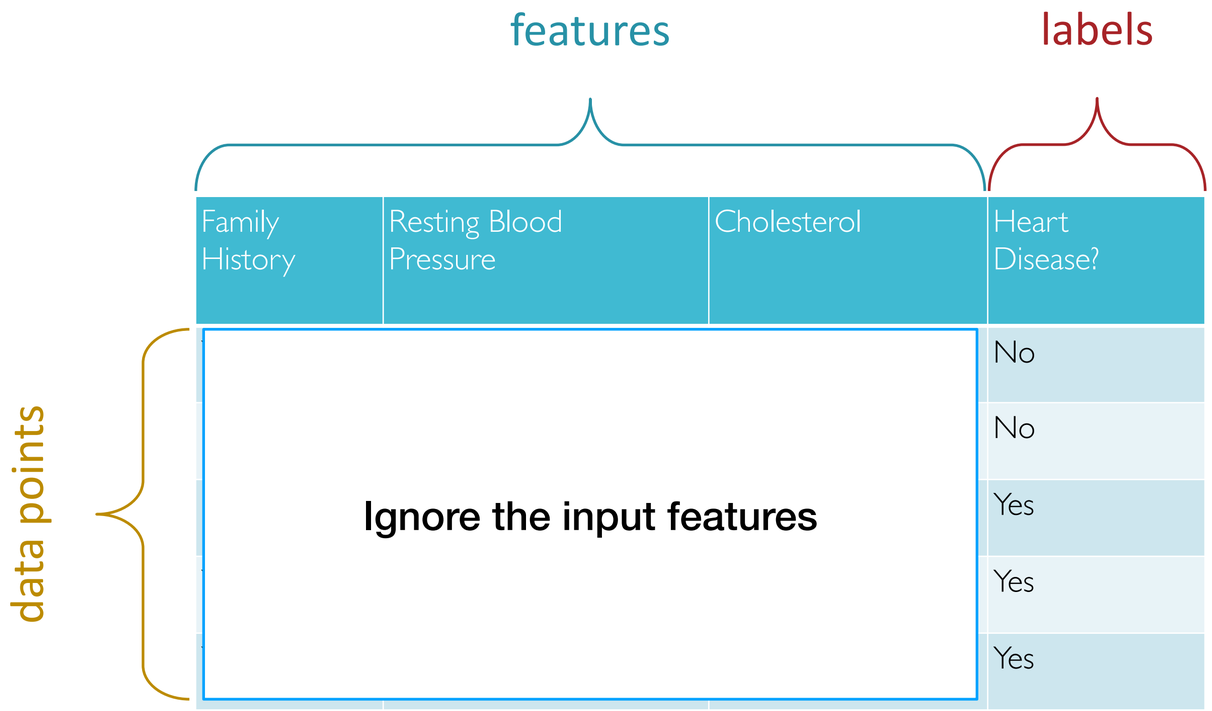
labels

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

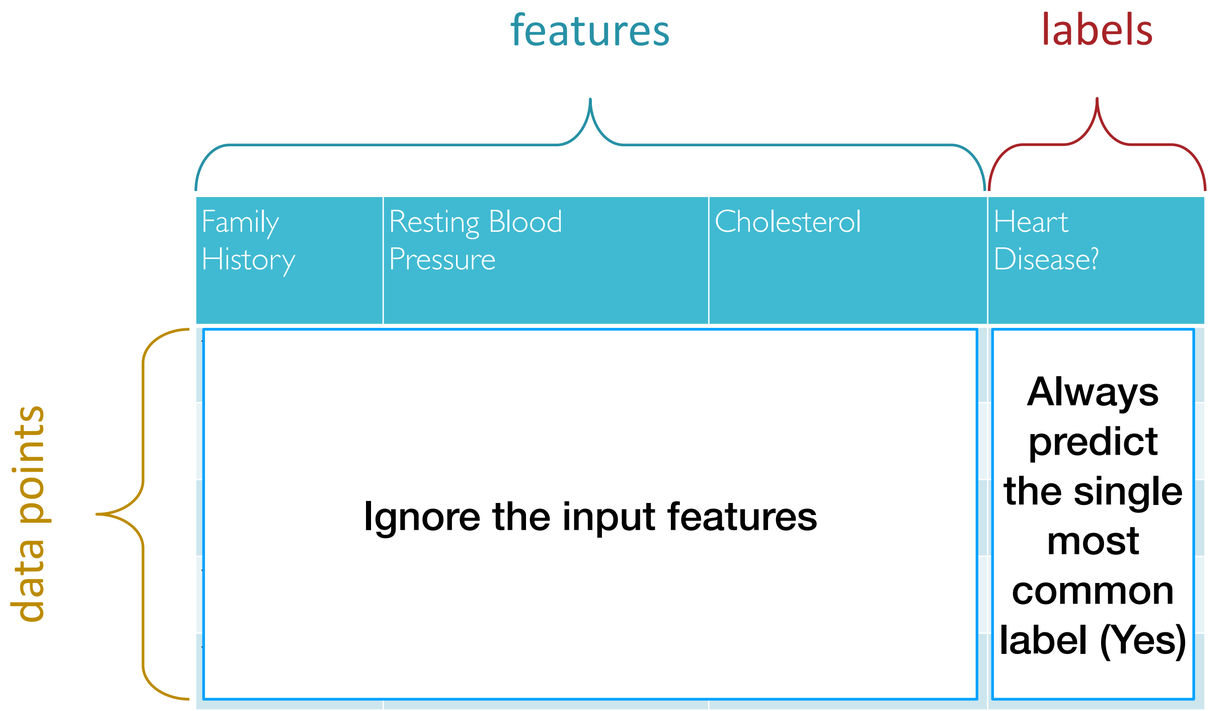
Our first ML method

- Majority vote classifier
  - for a **(supervised) classification task**



Our first ML method

- Majority vote classifier
  - for a **(supervised) classification task**



Our first ML method

- Majority vote classifier
  - for a **(supervised) classification task**

ground truth
true

features | labels | predictions

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?	Model output
Yes	Low	Normal	No	Yes
No	Medium	Normal	No	Yes
No	Low	Abnormal	Yes	Yes
Yes	Medium	Normal	Yes	Yes
Yes	High	Abnormal	Yes	Yes

$$\text{error rate} = \frac{1}{N}\sum_{(P)} \mathbb{I}(\text{true} \neq \text{predicted}) = \frac{2}{5}$$

Is this good?

- Is 40% error high or low?

  50% chance error rate if even π40% use

- Do we care about this error rate?

  No?

Training set vs. testing set

features | labels

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

training data points

learn

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
No	Low	Normal	No
No	Medium	Normal	No
Yes	Medium	Abnormal	Yes

testing data points

hide

training error rate = 2/5 $(\hat{P})$

testing error rate = 2/3 (P)

Geoff Gordon

Training set vs. testing set

Majority vote classifier: Always predict the single most common label *in the training set* (Yes)

features | labels

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

training data points

No	Low	Normal	No
No	Medium	Normal	No
Yes	Medium	Abnormal	Yes

testing data points

training error rate = $(\hat{P})$

testing error rate = (P)

Training set vs. testing set

Majority vote classifier: Always predict the single most common label *in the training set* (Yes)

features | labels | predictions

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?	Model output
Yes	Low	Normal	No	Yes
No	Medium	Normal	No	Yes
No	Low	Abnormal	Yes	Yes
Yes	Medium	Normal	Yes	Yes
Yes	High	Abnormal	Yes	Yes

training data points

No	Low	Normal	No	Yes
No	Medium	Normal	No	Yes
Yes	Medium	Abnormal	Yes	Yes

testing data points

training error rate = $(\hat{P})$

testing error rate = (P)

Geoff Gordon

Our second ML method

Memorizer: if we've seen *exact same* inputs before, predict corresponding output, else majority

features | labels

	Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
training data points	Yes	Low	Normal	No
	No	Medium	Normal	No
	No	Low	Abnormal	Yes
	Yes	Medium	Normal	Yes
	Yes	High	Abnormal	Yes
testing data points	No	Low	Normal	No
	No	Medium	Normal	No
	Yes	Medium	Abnormal	Yes

training error rate =

testing error rate =

Our second ML method

Memorizer: if we've seen *exact same* inputs before, predict corresponding output, else majority

features | labels | predictions

	Family History	Resting Blood Pressure	Cholesterol	Heart Disease?	Model output
training data points	Yes	Low	Normal	No	No
	No	Medium	Normal	No	No
	No	Low	Abnormal	Yes	Yes
	Yes	Medium	Normal	Yes	Yes
	Yes	High	Abnormal	Yes	Yes
testing data points	No	Low	Normal	No	
	No	Medium	Normal	No	
	Yes	Medium	Abnormal	Yes	

training error rate =

testing error rate =

Our second ML method

Memorizer: if we've seen *exact same* inputs before, predict corresponding output, else majority

features | labels | predictions

	Family History	Resting Blood Pressure	Cholesterol	Heart Disease?	Model output
training data points	Yes	Low	Normal	No	No
	No	Medium	Normal	No	No
	No	Low	Abnormal	Yes	Yes
	Yes	Medium	Normal	Yes	Yes
	Yes	High	Abnormal	Yes	Yes
testing data points	No	Low	Normal	No	Yes
	No	Medium	Normal	No	No
	Yes	Medium	Abnormal	Yes	Yes

training error rate = 0%

testing error rate = 33%

Geoff Gordon

Notation

- Data point = $\langle x^{(t)}, y^{(t)} \rangle$
- Feature vector and feature space $x^{(t)} \in \mathcal{X}$
- Label and label space $y^{(t)} \in \mathcal{Y}$
- Hypothesis (classifier) and hypothesis space $h \in \mathcal{H}$
- ***Unknown*** target or true classifier h^*
  - best possible test error, might not even be in $\mathcal{H}$
- Apparent best classifier $\hat{h} \in \mathcal{H}$ given training set (sometimes called ERM, empirical risk minimizer)
  - minimizes training error over $\mathcal{H}$
- ML method: a way of picking a hypothesis given data
  - e.g., majority vote = empirical risk minimization over constant classifiers

Learning goals

- You should be able to
  - Formulate a well-posed learning problem for a real- world task by identifying the task, performance measure, and training experience
  - Describe the supervised learning paradigm in terms of the type of data needed, the form of prediction, and the structure of the output prediction
  - Explain the difference between memorization and generalization