Machine Learning for Networking

1. Programming in Python / Machine Learning in Python.
2. The Internet / Computer Networking
3. Machine Learning Models, Algorithms

4. Putting it together

What is Machine Learning?

- Some algorithm / program that gets better at performing a task as it gains more data.

- Supervised: Learning to predict outcomes. (data has labels)

$$y = f(x)$$

f is "learned"

y: targets/outcomes; x: features

- Unsupervised: Learning structure from data (data has no labels)
  - Clusters
  - Reducing dimensions

Linear Algebra

$$\beta = [\beta_1, \beta_2, \cdots \beta_n]$$

$$x = [x_1, x_2, \cdots, x_n]$$

$$y = mx + b$$

$$\approx \beta_1 x_1$$

$$\approx \beta_1 x_1 + \beta_2 x_2 + \cdots$$

$$\beta \cdot x = \beta^T x = x \beta^T$$

3×2

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix}$$

Transpose

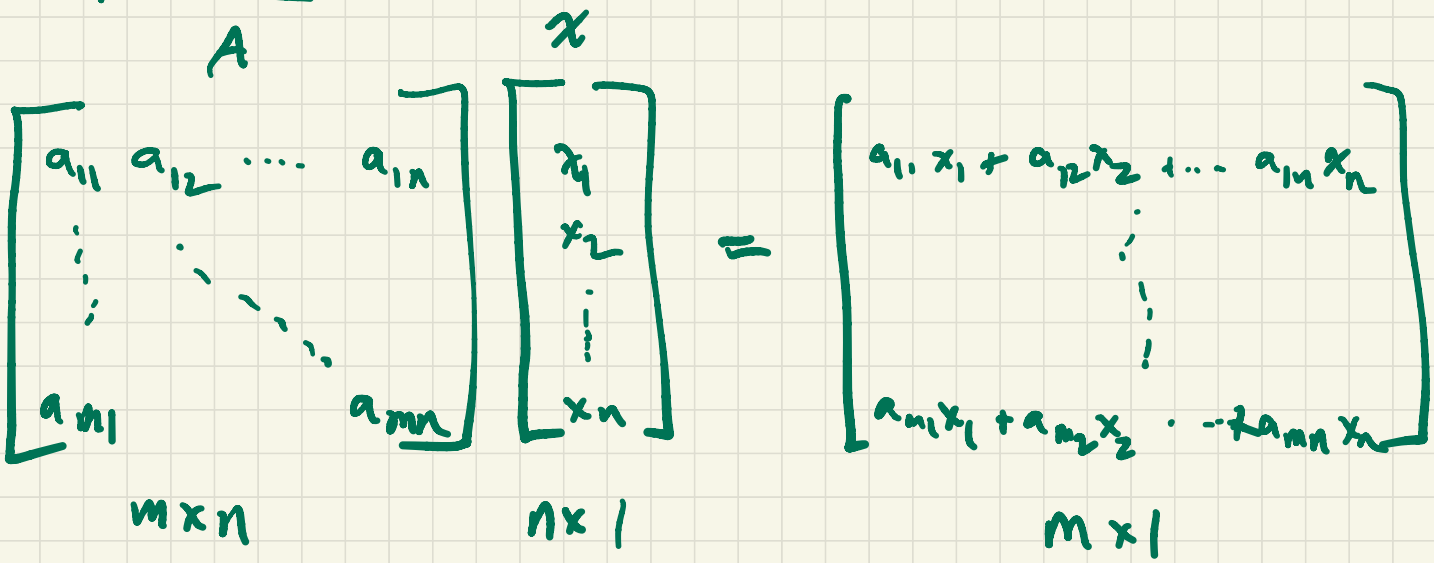
2×3

$$\begin{bmatrix} x_{11} & x_{21} & x_{31} \\ x_{12} & x_{22} & x_{32} \end{bmatrix}$$

Dot Product

$$[x_1 \cdots x_n] \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} \cdots = \beta_1 x_1 + \beta_2 x_2 + \cdots$$

Matrix Multiplication



$$AB \longrightarrow M$$

$m \times n \quad n \times p \qquad m \times p.$

How would you use machine learning in computer networking?

1. Security
   - Detect attacks (e.g., denial of service, spam)
   - Detect unusual behavior ("anomalies")
   - Detect or classify devices, applications, OSes, ...
2. Performance
   - Infer application performance (e.g., streaming video)

① Basic Machine Learning Model:

- Supervised vs unsupervised
- Basic Model: Linear Regression.

② The Internet: Data about the Internet

- How does data get from A to B?
  - Packets ("chunks")  "envelope"
  - Bytes (0s & 1s) → 8 bits.
- What is in a packet? ("Header") — WIRESHARK
  - IP addresses, ports, protocols.

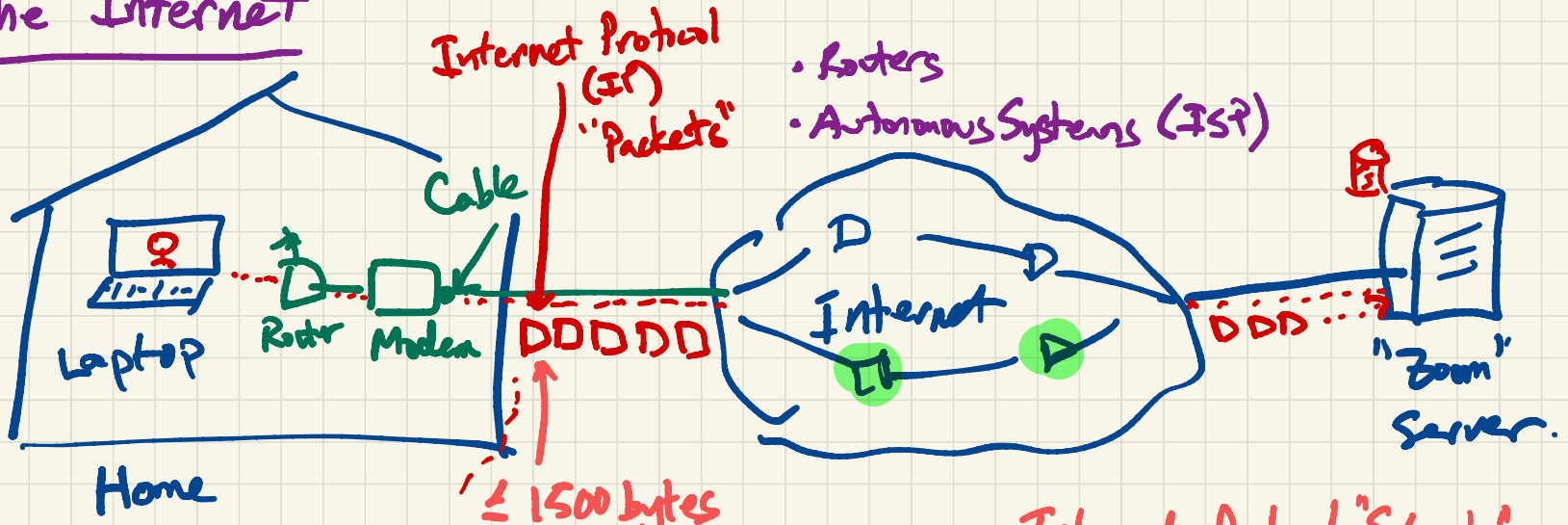
- To come: Names → IP addresses ("DNS")

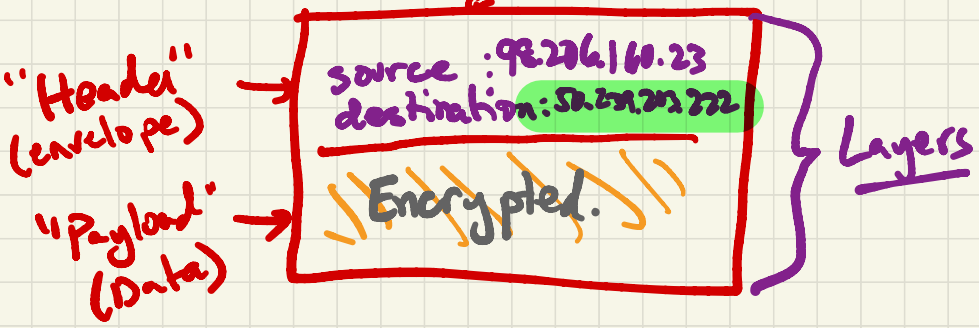
google.com → 172.217.9.46

AGENDA

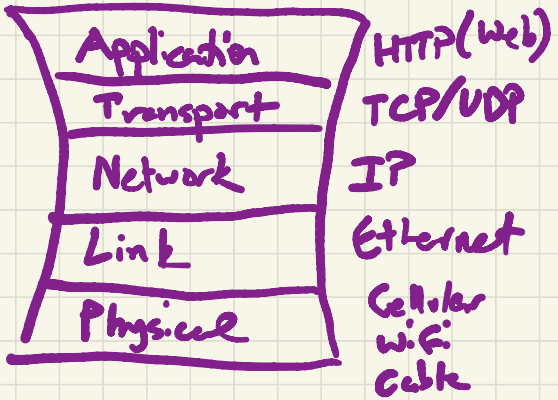
1. Internet Background / Basics.
   - Internet Protocol.
   - Packets
   - Domain Name System (DNS)
2. Live Demonstation
   - Network Measurement / Packet Capture (Wireshark)
   - Web page download.
3. Analyzing Network Traffic in Python.

The Internet

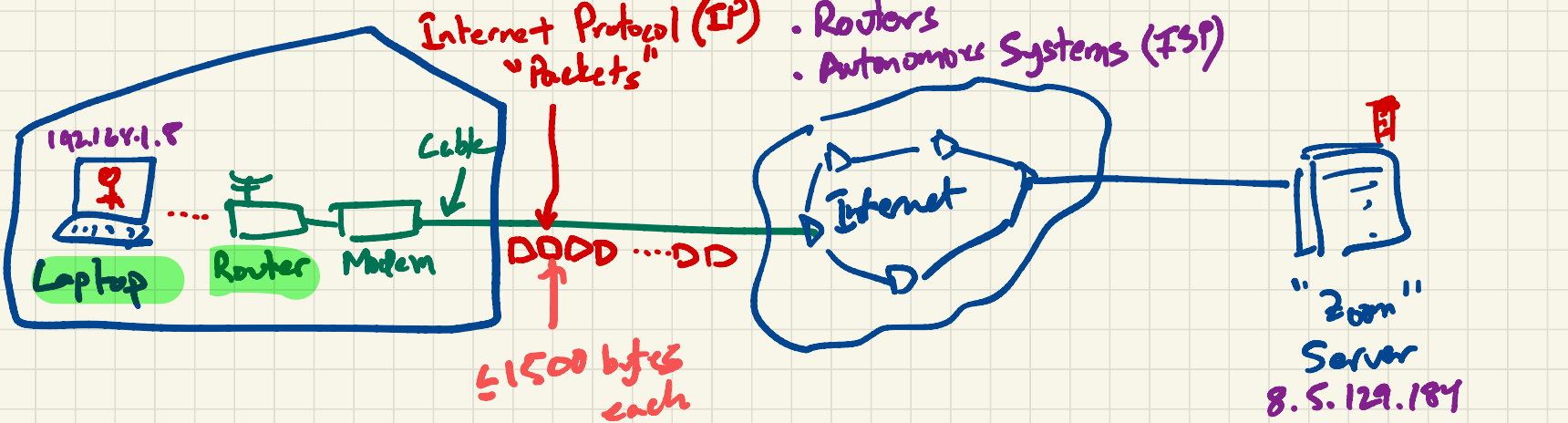


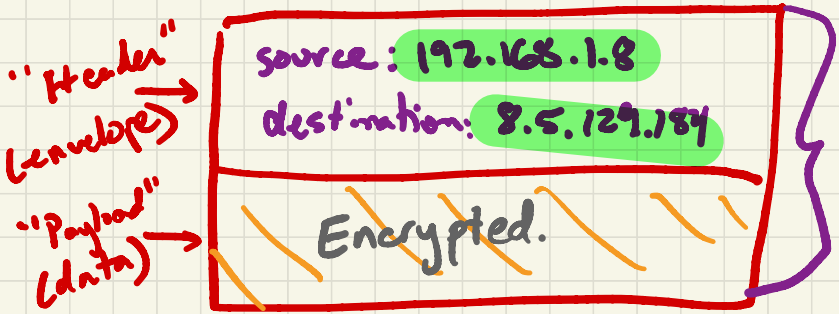


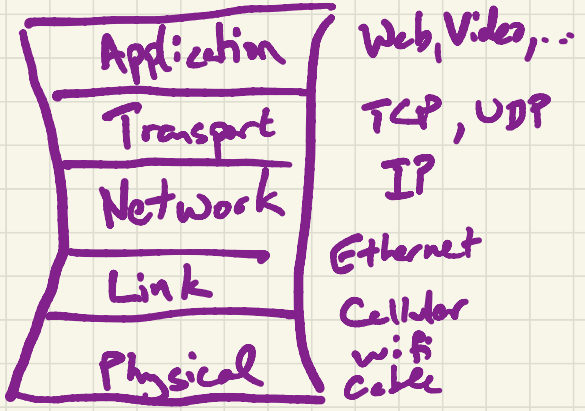
Internet Protocol "Stack"



The Internet





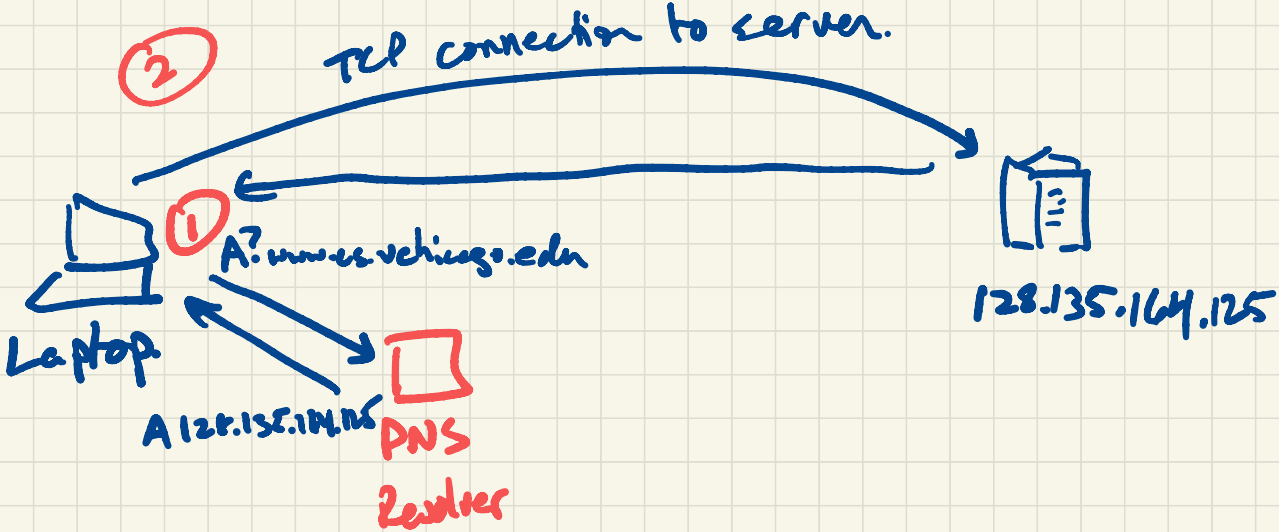


Domain Name System (DNS)

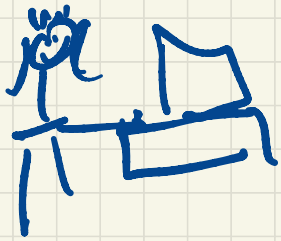
1. DNS Lookup
2. Connect to server

Name: www.cs.uchicago.edu

Answer: 128.135.164.125



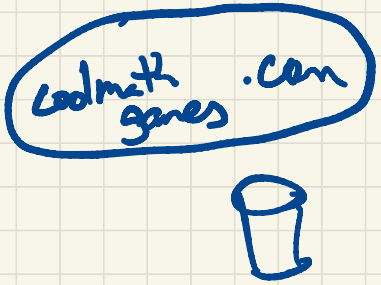
Domain Name System (DNS)

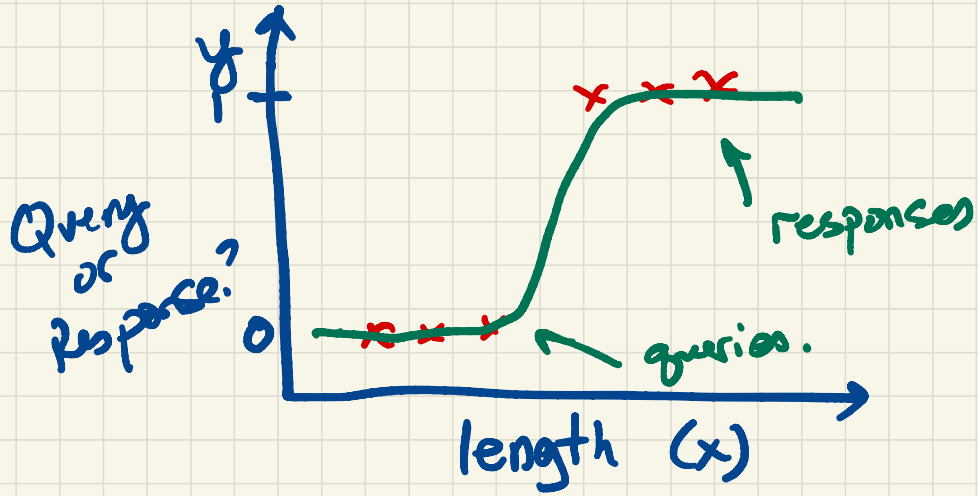


coolmathgames.com (81 bytes) ??

IP address?

151.101.186.49. (138 bytes)





HOME

1. How does my computer connect to the Internet?

2. What other "things" are connected to the Internet in my house?

3. How do so many devices in my house "share" the same Internet connection?

4. What places in your house have good Wifi performance/connectivity? Bad?

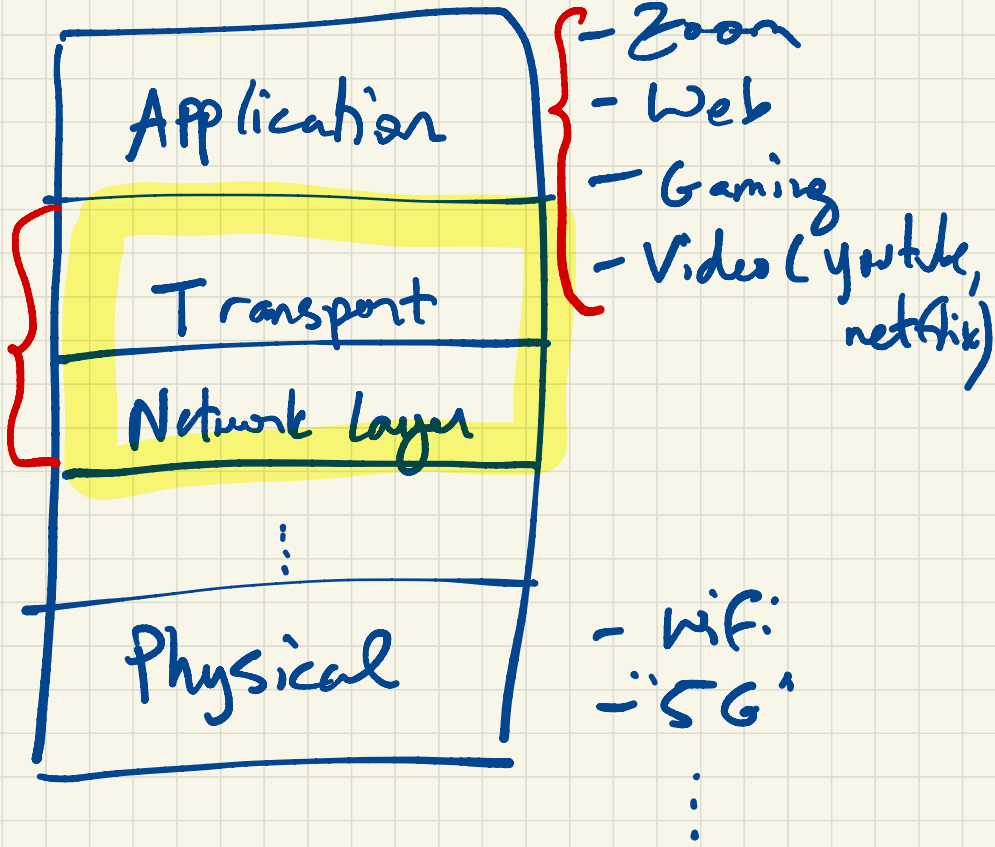
Applications

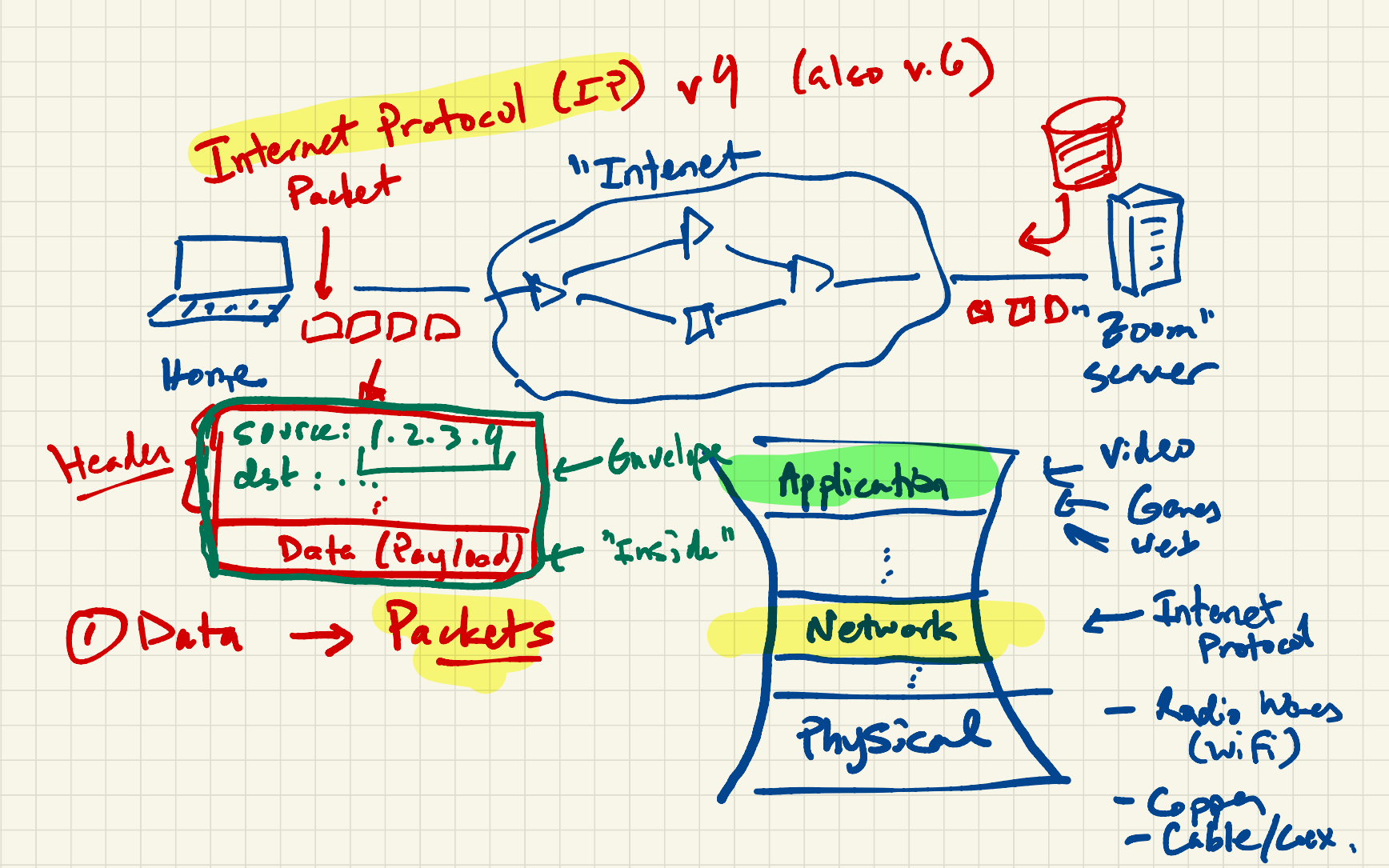
1. What happens when I...
   - download a web page?
   - send an email?
   - stream video?

TCP/IP

"plumbing"

1. RELIABLE
2. IN ORDER



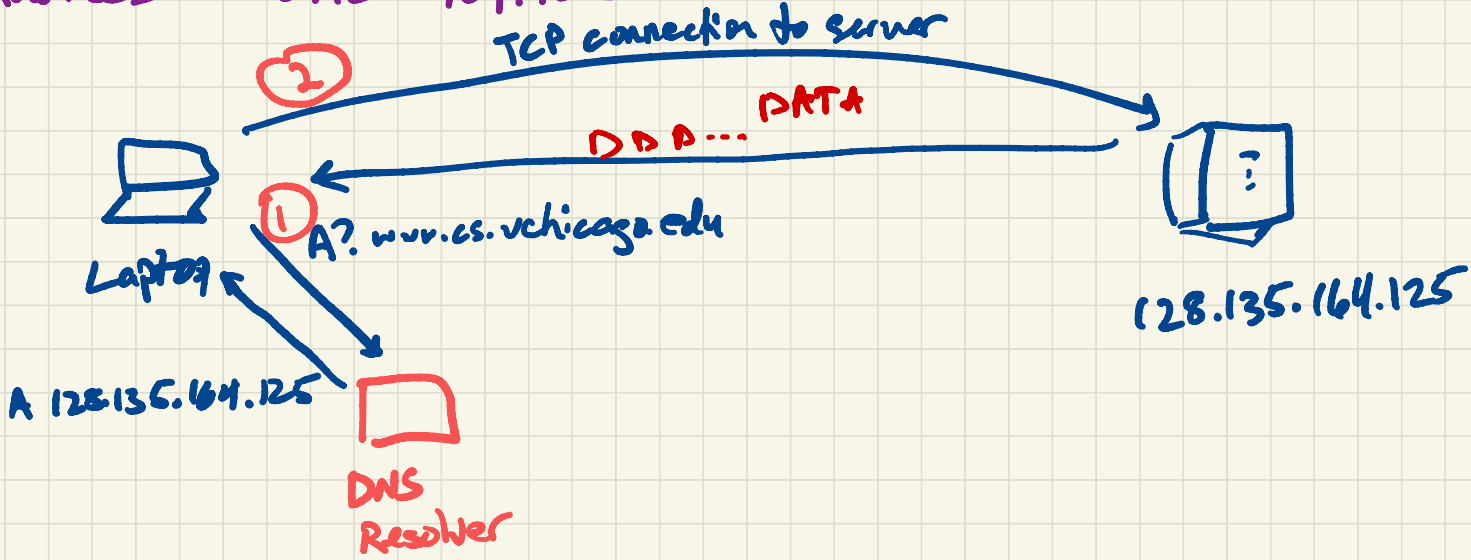

Internet Protocol (IP) v4 (also v.6)
Packet
"Internet
Home
"Zoom" server
Header
Source: 1.2.3.4
dst: ...
Data (Payload)
Envelope
"Inside"
① Data → Packets
Application
Network
Physical
Video
Games
Web
Internet Protocol
- Radio Waves (Wi-Fi)
- Copper
- Cable/Coax.

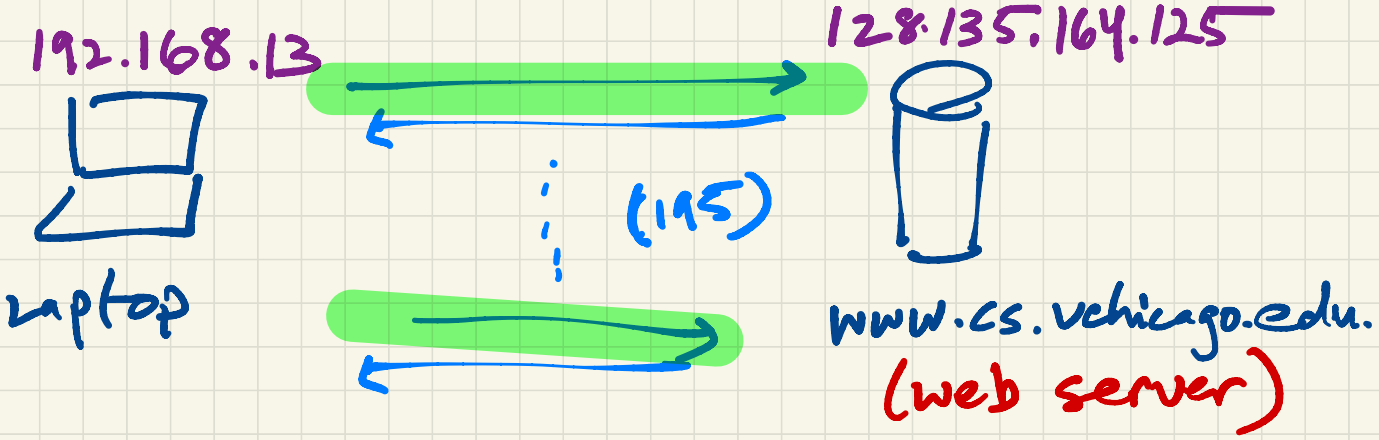
Domain Name System (DNS)

① DNS Lookup
② Connect to server (TCP/IP connection)

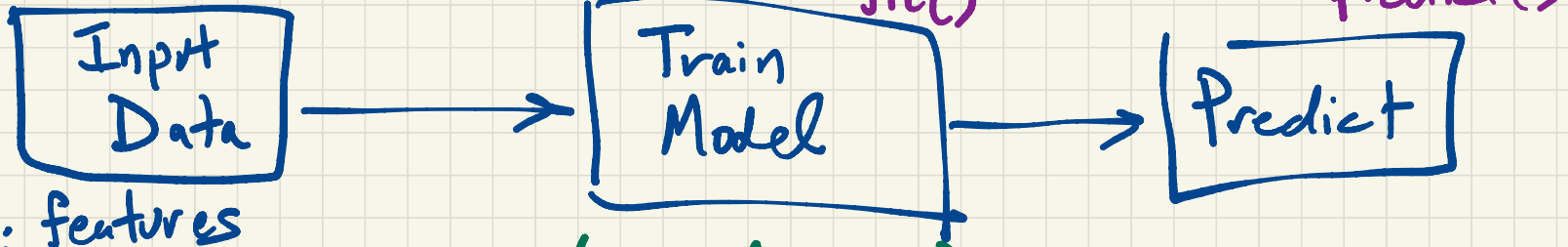
Name: www.cs.uchicago.edu

Address: 128.135.164.125




192.168.13
laptop
128.135.164.125
(195)
www.cs.uchicago.edu.
(web server)

MACHINE LEARNING



X: features

y: target

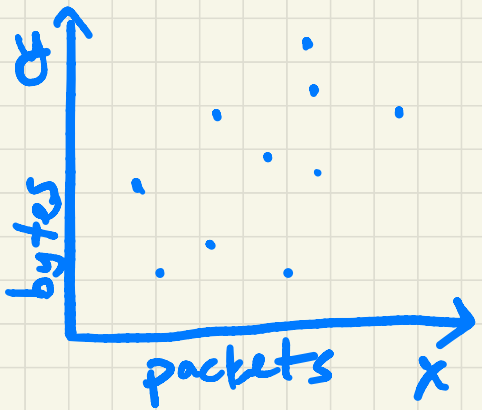
(e.g., Linear Regression)

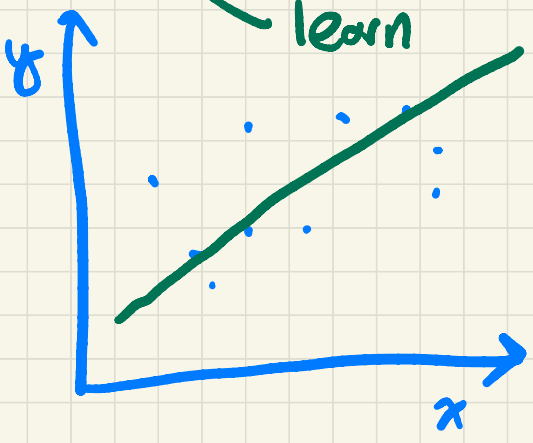
$y = f(x)$

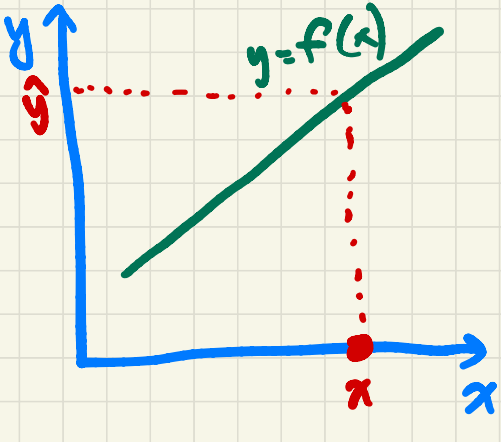
learn

$\hat{y} = f(x)$

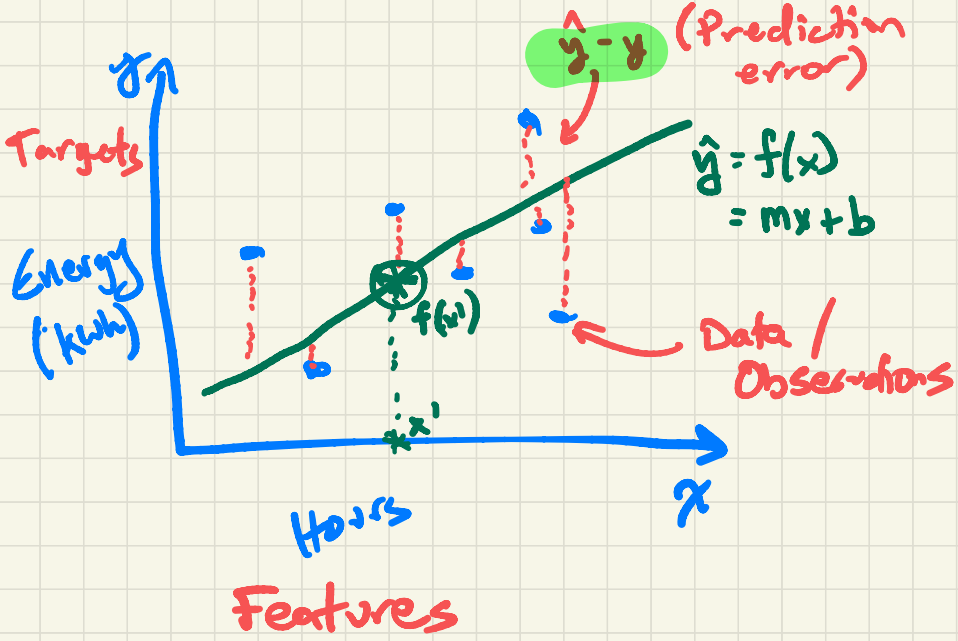
prediction.







Linear Regression



must learn

$$\{y_1, \dots y_n\} = f(\{x_1, \dots, x_n\})$$

Minimize Error

$$RSS = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

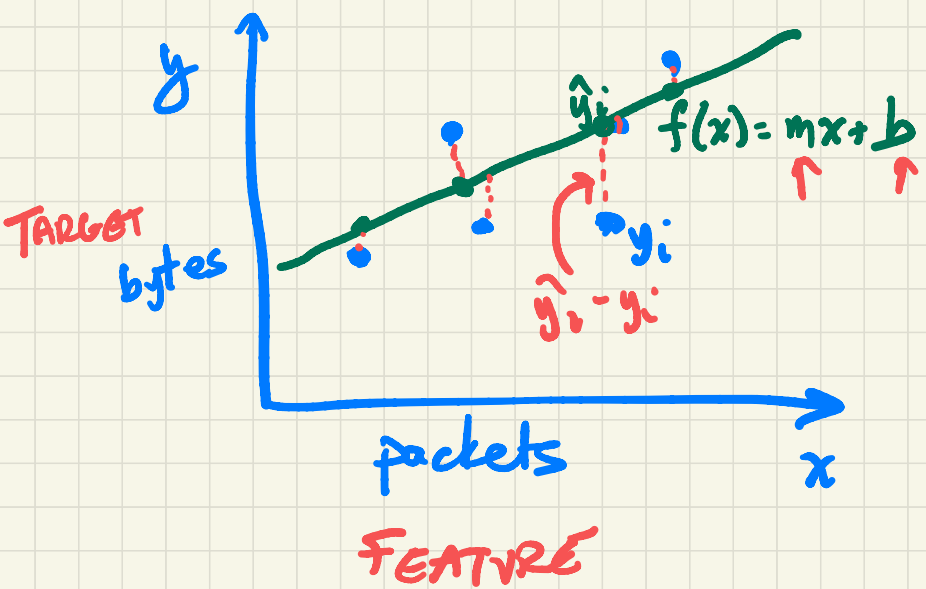
$\hat{y}_i$: prediction; y_i: actual value

$$\hat{y} = f(x) = mx + b$$

m: slope; b: intercept

Note: Can solve in closed form. (Linear Algebra)

Linear Regression (1 feature)



$\{y_1, \ldots, y_n\} = f(\{x_1, \ldots x_n\})$

target — features

must learn

$\hat{y}_i = f(x_i)$ — linear

Goal: Minimize Error

$$RSS = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

Note: Closed form solution.

$$\hat{y} = f(x) = mx + b$$

Linear Regression Optimization

$$\hat{y} = m\vec{x} + b$$

$$\hat{Y} = \hat{\beta}_0 + \sum_{i=1}^{n} x_i \hat{\beta}_i$$

$\hat{\beta}_0$ ↑ b ; $\hat{\beta}_i$ ↑ m ; multiple x variable

$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$

$$\hat{Y} = X^T \hat{\beta} \quad \text{(linear equations)}$$

Error

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} (y_i - x_i^T \beta)^2$$

$$= (y - X\beta)^T (y - X\beta)$$

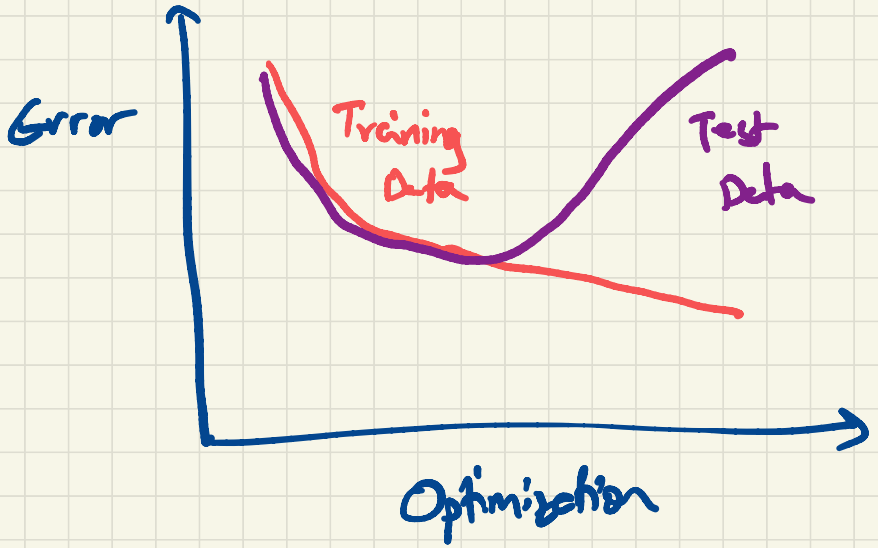
Minimize:

$$X^T(y - X\beta) = 0$$

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Solving a set of linear equations.

Bias - Variance Tradeoff



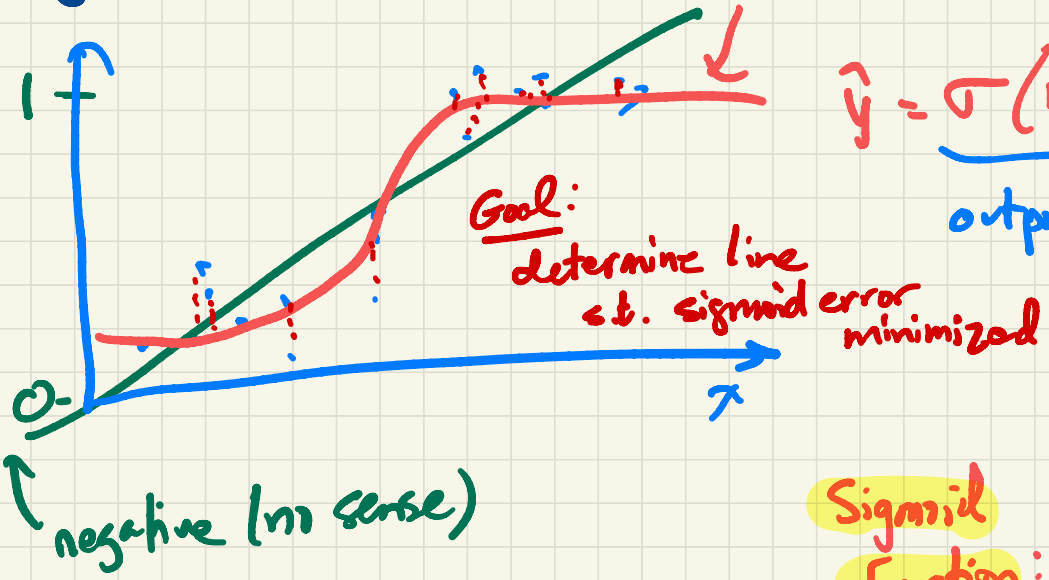
Logistic Regression

- Linear model.
- For all values of x, $y \in [0, 1]$
- **Reason: Want to predict binary value**

(e.g., Spam/ham, malware, attack, ...) → yes/no

sigmoid

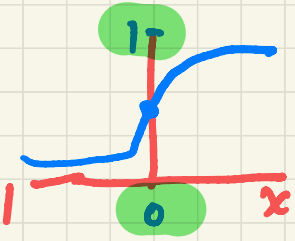
$y = f(\sigma(x))$



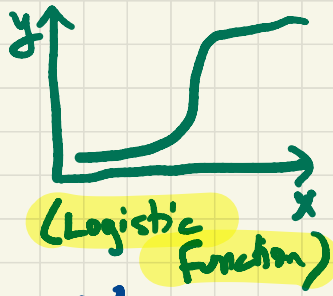
Goal: determine line s.t. sigmoid error minimized

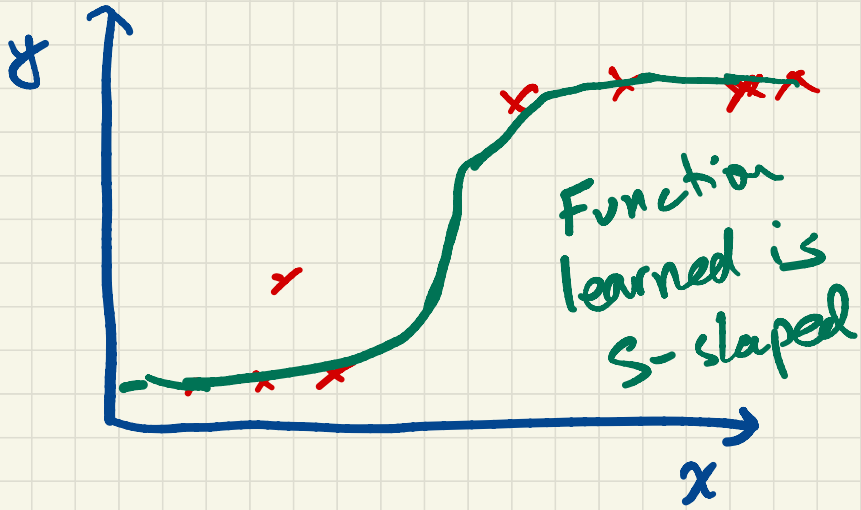
$\hat{y} = \underbrace{\sigma(mx + b)}_{\text{output is b/w 0/1.}}$

Sigmoid Function: $\sigma(x) = \frac{1}{1+e^{-x}}$



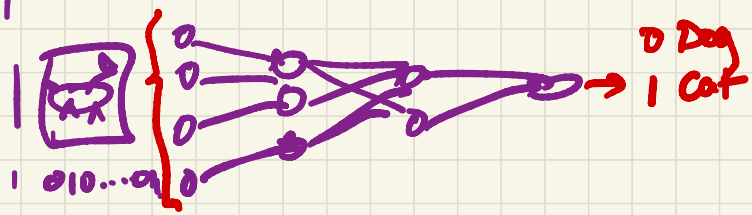
Logistic Regression





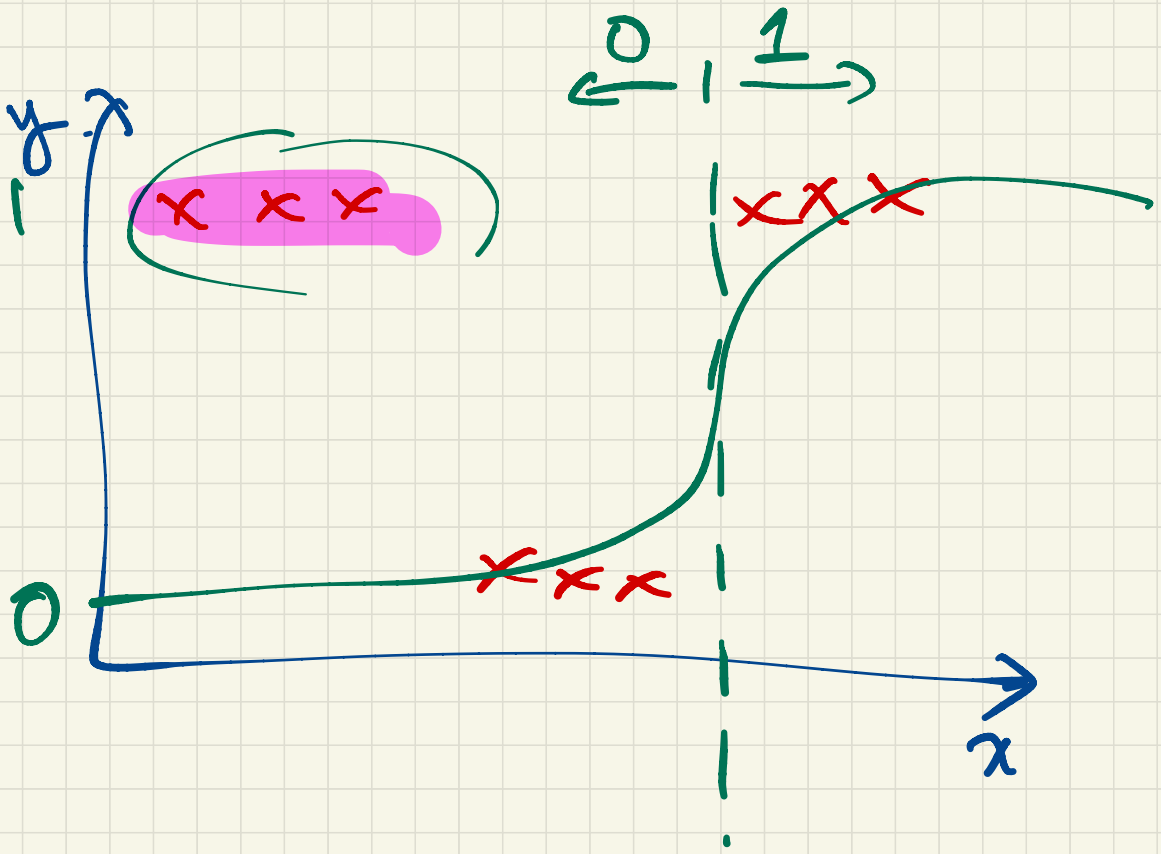
Function learned is S-shaped

("Deep Learning")

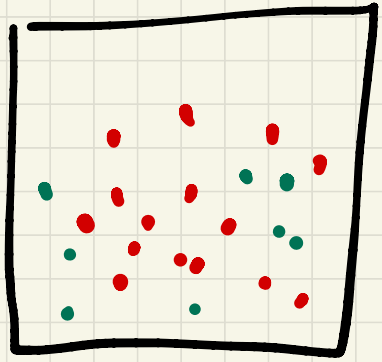


Examples: → Yes (1) or No (0)?

1. Spam email or ham/beer?
2. Is my laptop infected?
3. Is the Internet under attack?


0
1
y
0
x

Basic Probability



100 balls total
75 red balls
25 green balls

$P(\text{red ball}) = \frac{75}{100} = 0.75$

$P(\text{green ball}) = \frac{25}{100} = 0.25$

$P(\text{red ball}, \text{green ball}) = ?$

① Put red ball back?

$P(\text{red}) \cdot P(\text{green}) = \frac{75}{100} \cdot \frac{25}{100} = \frac{3}{16} \approx 0.1875$

② Keep red ball out?

$P(\text{red}) \cdot P(\text{green}|\text{red}) = \frac{75}{100} \cdot \frac{25}{99} \approx 0.19$

① Independence

$P(A, B) = P(A) \cdot P(B)$

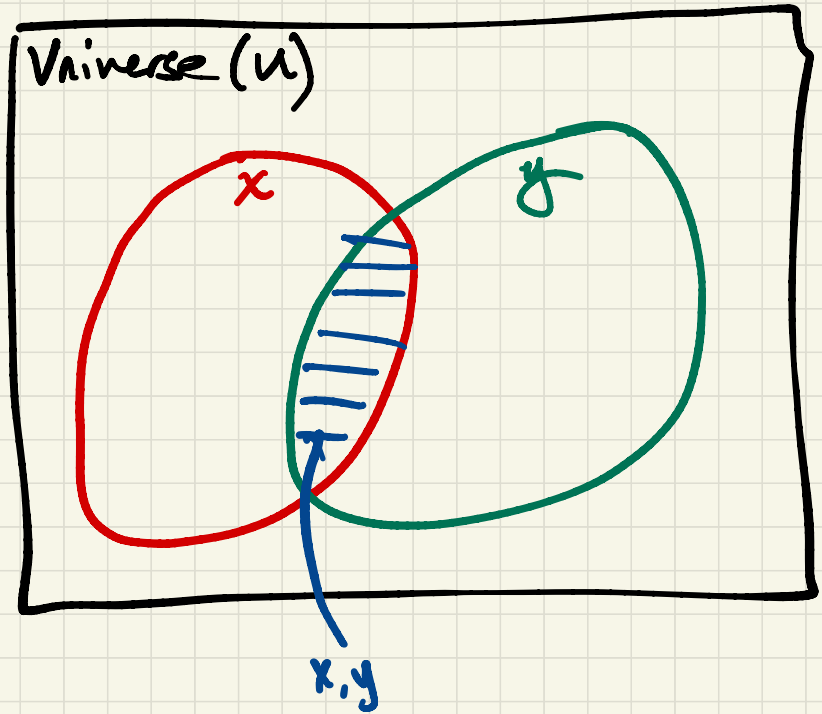
② Conditional Probability

$P(A|B)$

$P(B, A) = P(B|A) \cdot P(A)$

Bayes' Rule

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$$



$$P(x) = \frac{|x|}{|u|} \qquad P(y|x) = \frac{|xy|}{|x|} = \frac{\frac{|xy|}{|u|}}{\frac{|x|}{|u|}} = \frac{P(xy)}{P(x)}$$

$$P(y) = \frac{|y|}{|u|}$$

$$P(x,y) = \frac{|xy|}{|u|}$$

$$P(x|y) = \frac{P(xy)}{P(y)}$$

$$P(y|x) \cdot P(x) = P(x|y) \cdot P(y)$$

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$$

predict — observe

Naïve Bayes Classifier

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$$

Predict $\hat{y}$, having observed x.

$$\hat{y} = \underset{y}{\operatorname{argmax}} \left\{ P(y = C \mid x = x_1, x_2, x_3 \cdots, x_n) \right\}$$

(e.g., spam, ham)

need to compute!

$$= \underset{y}{\operatorname{argmax}} \left\{ \frac{P(x = x_1, \cdots, x_n | y) \cdot P(y)}{\cancel{P(x)}} \right\}$$

observed or estimated (easy)

easy

$$= \underset{y}{\operatorname{argmax}} \left\{ P(x_1|y) \cdot P(x_2|y) \cdots P(x_n|y) \cdot P(y) \right\}$$

Example: Spam filtering

TRAINING

SPAM (y=SPAM)

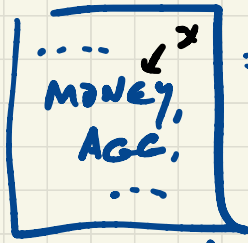
- MONEY, RILEX
- MONEY, PILLS
- PHARMACY, AGE

$P(\text{MONEY} \mid \text{SPAM}) = 2/3$

HAM (y=HAM)

- CLASS
- MACHINE LEARNING
- AGE

$P(\text{MONEY} \mid \text{HAM}) = 0$
$P(\text{AGE} \mid \text{HAM}) = 1/3$



$= \underset{y}{\text{argmax}} \{ P(x_1 \mid y) \cdot P(x_2 \mid y) \cdots P(x_n \mid y) \cdot P(y) \}$

TESTING

① $P(y=\text{SPAM} \mid \text{MONEY, AGE})$

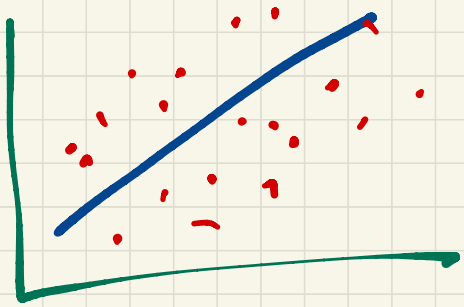
$\approx P(\text{MONEY} \mid \text{SPAM}) \cdot P(\text{AGE} \mid \text{SPAM}) \cdot P(\text{SPAM})$

$\approx 2/3 \cdot 1/3 \cdot 9/10 \approx \boxed{18/90}$

② $P(y=\text{HAM} \mid \text{MONEY, AGE})$

$\propto P(\text{MONEY} \mid \text{HAM}) \cdot P(\text{AGE} \mid \text{HAM}) \cdot P(\text{HAM})$

$\approx 0 \cdot 1/3 \cdot 1/10 = \boxed{0}$



Machine Learning

Supervised Learning (data & predictions have labels)

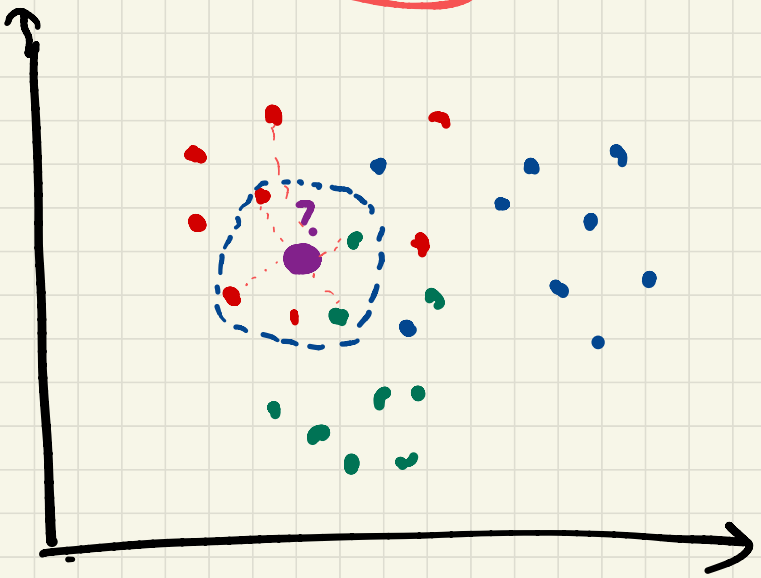
- Linear
  - Linear Regression (continuous)
- Probabilistic
  - Logistic Regression (binary)
  - Naive Bayes
- Non-Parametric
  - Nearest Neighbor ← Today.

Unsupervised Learning (no labels)

- Clustering ← Thursday

Nearest Neighbor (k-NN)

Not known! (Explore)



Intuition:

Point is of type that is the same as other "close" points.

Example:

3-NN
- 3 red
- 2 green

⇓

RED

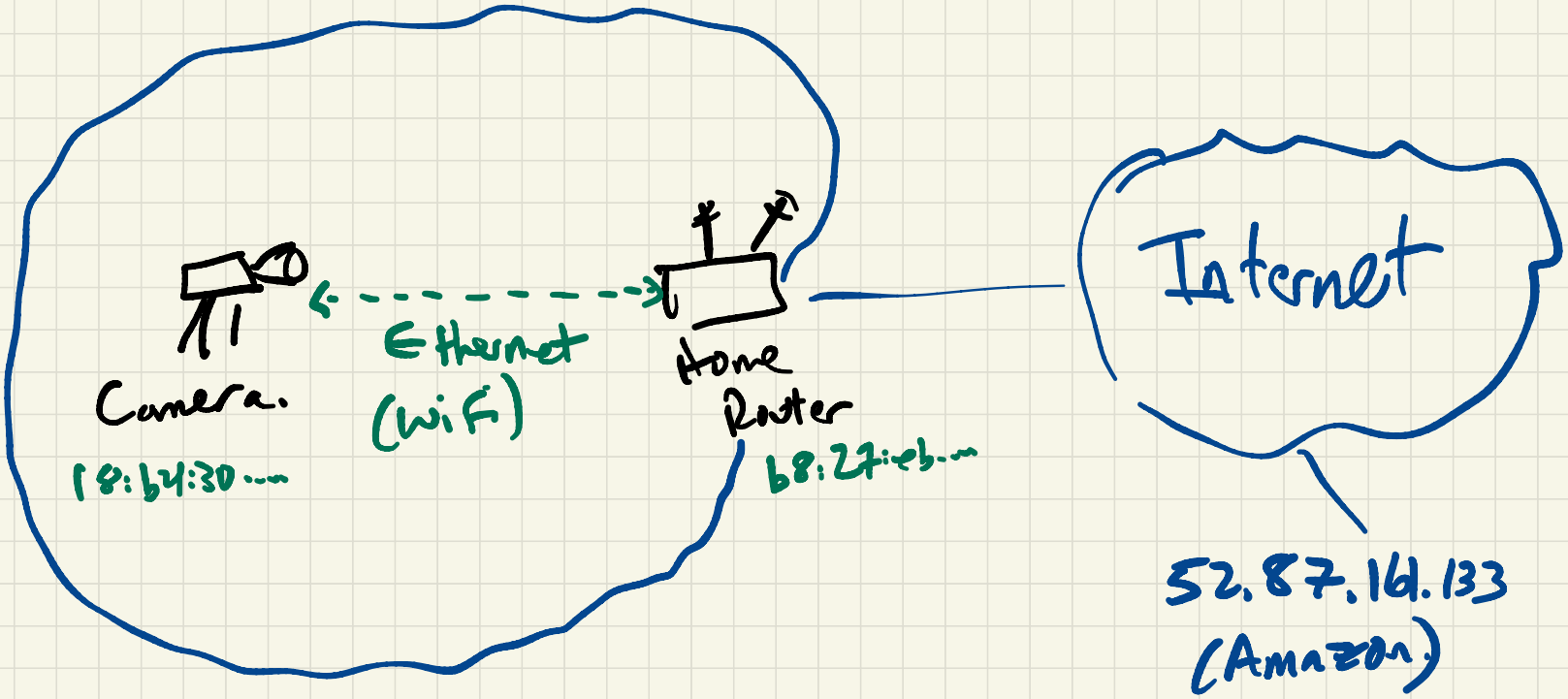
① Other: How many?

② Close?

→ Euclidean

To compute:

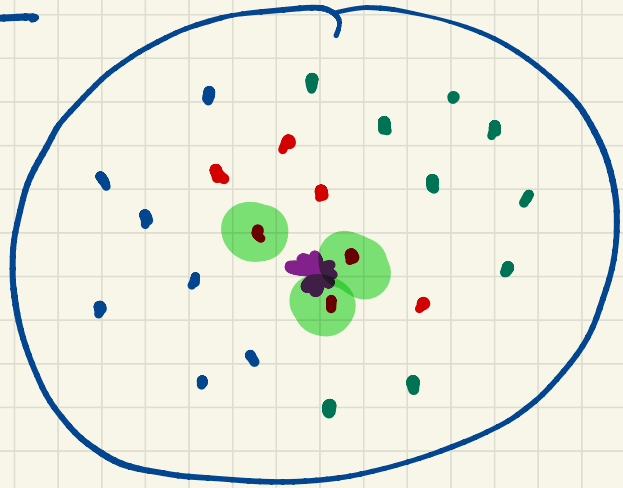
→ Need to store all data points

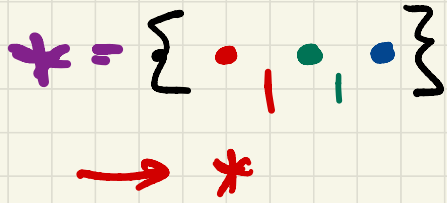

Internet
Camera
18:b4:30...
Ethernet
(WiFi)
Home
Router
b8:27:eb...
52.87.161.133
(Amazon)

Nearest Neighbor

Parameter: k

e.g., k=3





In this example, k closest points all red.

Advantages: No training → store data points

Disadvantages: Classification can be expensive ← computation
Must store entire dataset. ← storage

Tree-Based Methods

Idea: Split feature space into simpler regions.

- Uses a sequence of rules
- Sequence of rules summerized in a tree

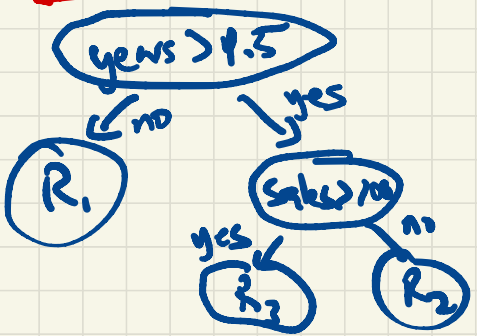
\+ Easy to interpret.

− Single tree not that accurate (solution: Bagging, Boosting)

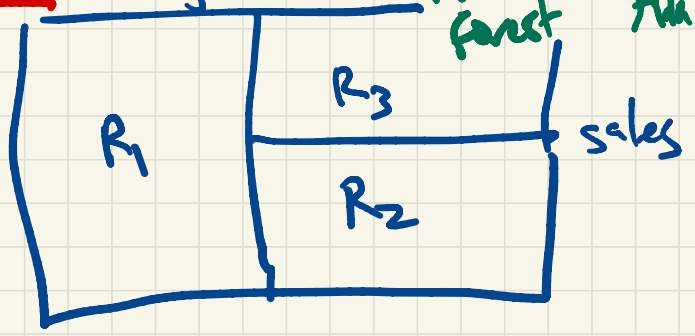
↑ Random Forest ↑ Adaboost

Example

Tree



Map



Regression Tree

- Predict Continuous Values.
- Splits in tree → Reduction in mean squared error

Greedy Approach:

$$R_1(j,s) = \{X | X_j < s\}$$

$$R_2(j,s) = \{X | X_j > s\}$$

Minimize w.r.t S

$$\sum_{i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

y_i: true values; $\hat{y}_{R_1}$, $\hat{y}_{R_2}$: prediction

✓

Classification Trees

- Predict Qualitative/Categorical Outcome.
- Split criteria: Classification Error Rate

  → Fraction of training observations that don't belong to most common class in region.

Gini Index

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$$

Entropy

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

Ensemble Methods

Problem: Decision trees have high variance

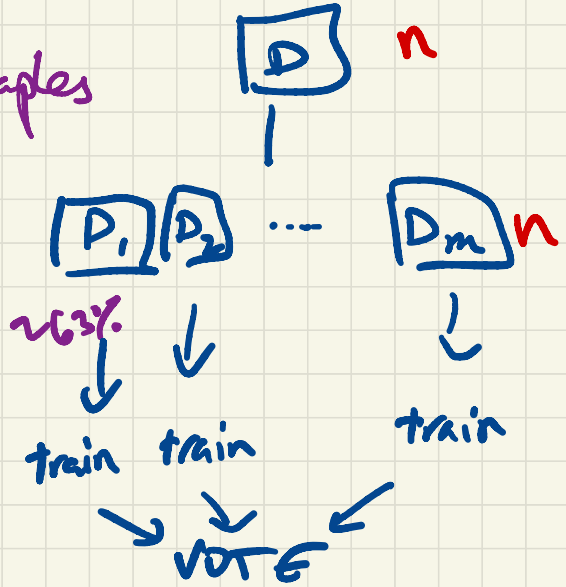
Bagging (Bootstrap Aggregation)

→ Separate Model w/ repeated random samples

→ Average predictions

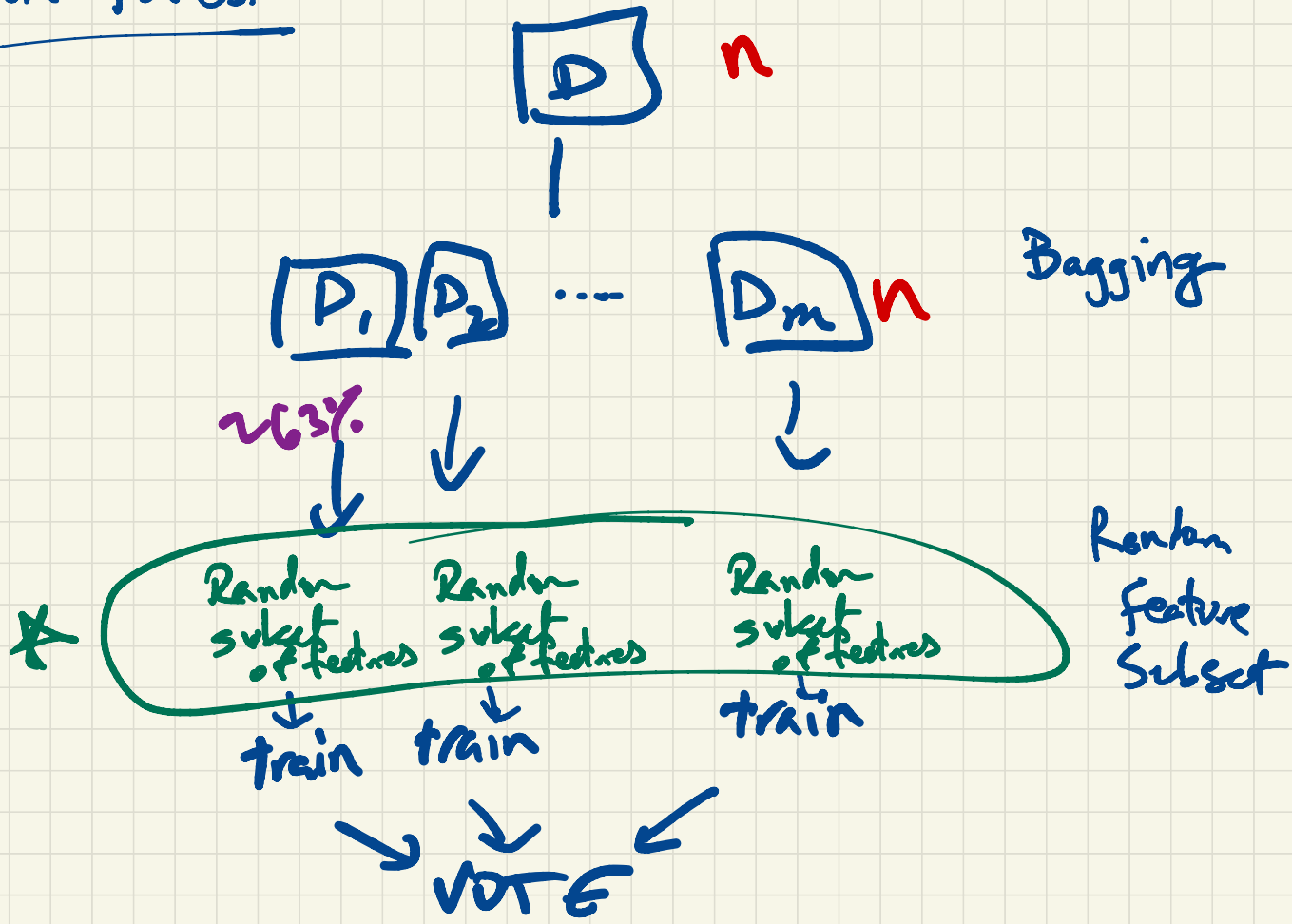
→ training set → n

M samples of size n



✓

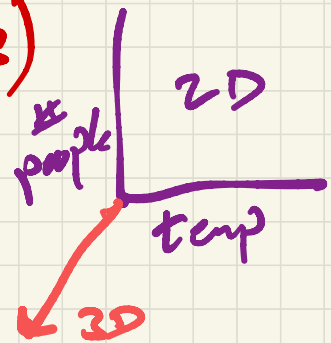
Random Forest



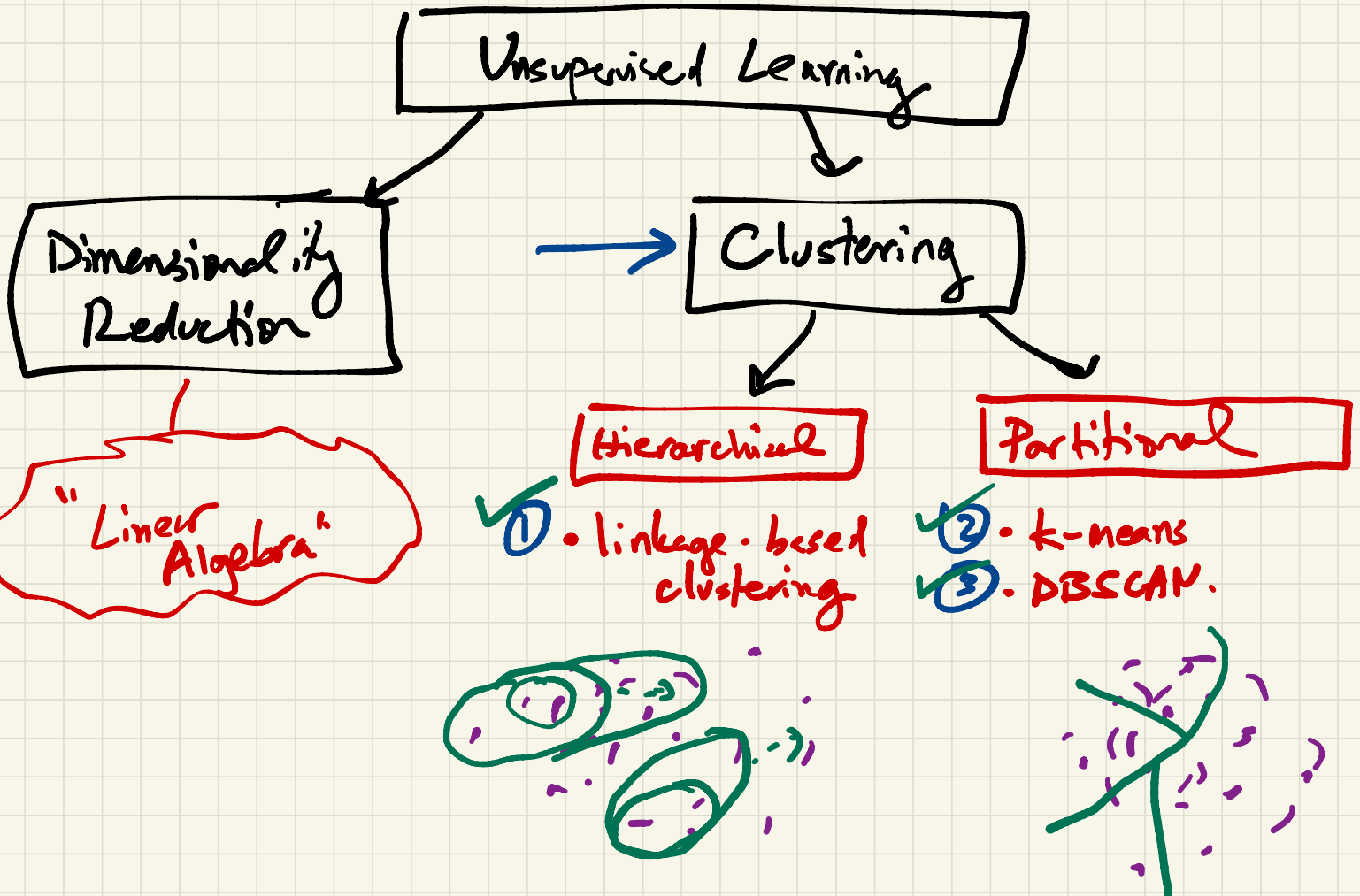
Why Unsupervised Learning?

1. Might be challenging/costly to get labels
2. May want to solve a problem other than making a prediction

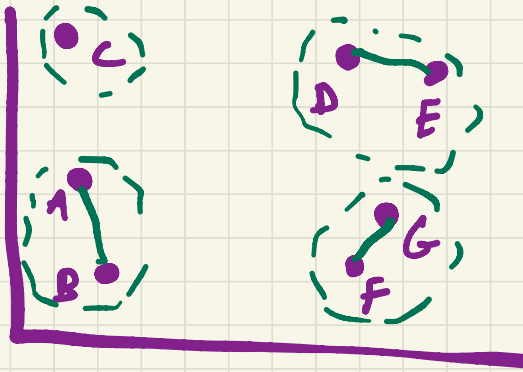
- Visualization (≤ 3 dimensions)



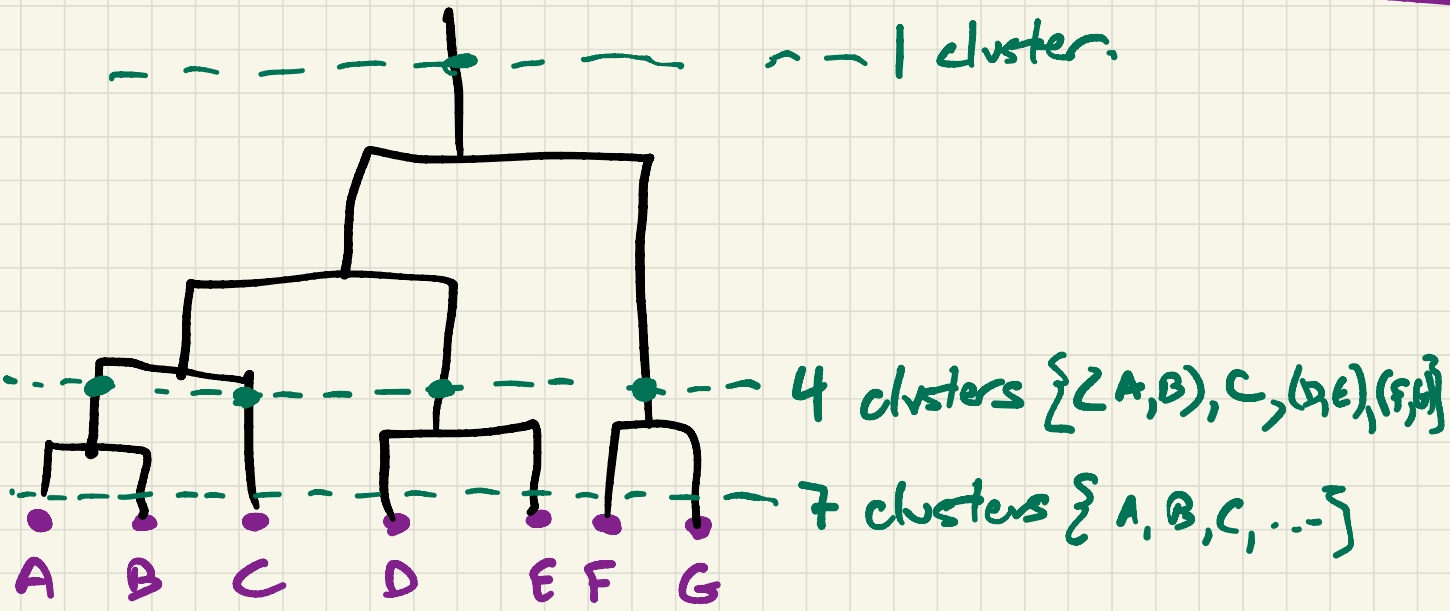
- Understand groups/structure/similarity
- Dimensionality Reduction.


Unsupervised Learning
Dimensionality Reduction
Clustering
"Linear Algebra"
Hierarchical
Partitional
1. linkage based clustering
2. k-means
3. DBSCAN.

Linkage-Based Clustering / Hierarchical



Dendrogram



Not All Data is Hierarchical

Data: People. {gender, nationality}
(M, F) ; (US, Canada, Mexico)

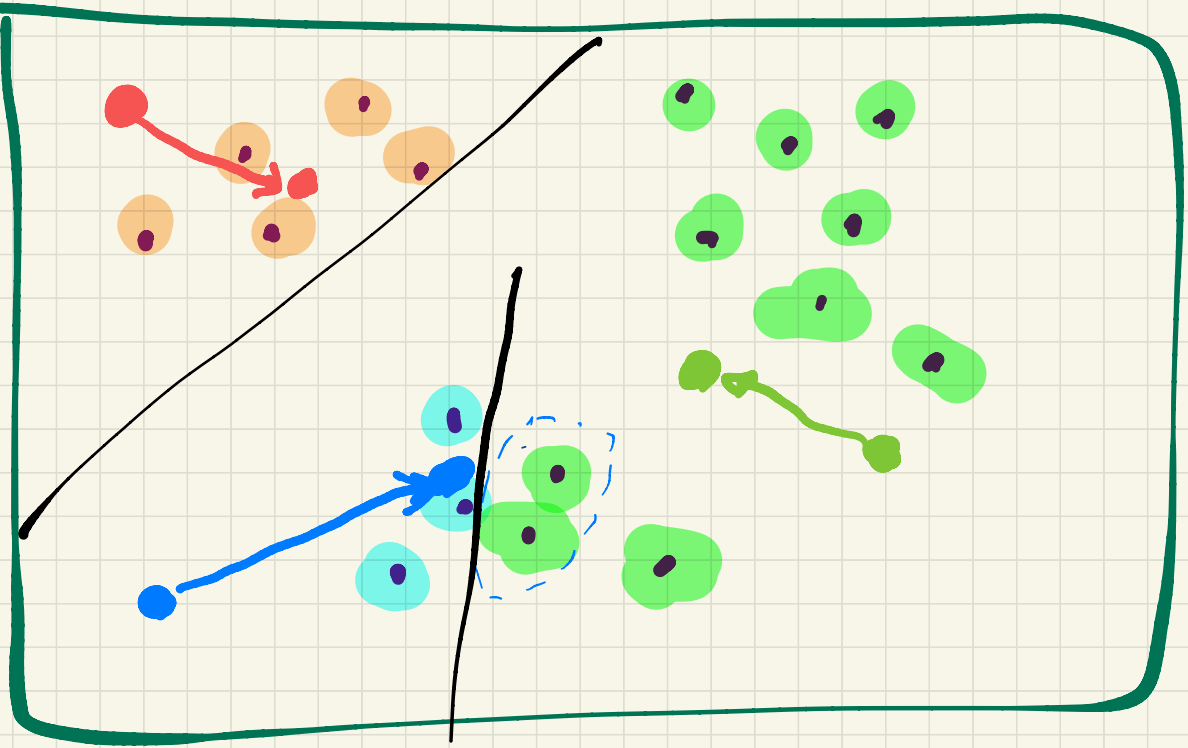
k = 2 → best clustering might be based on gender

k = 3 → best clustering might be based on nationality

k-means clustering

must select!

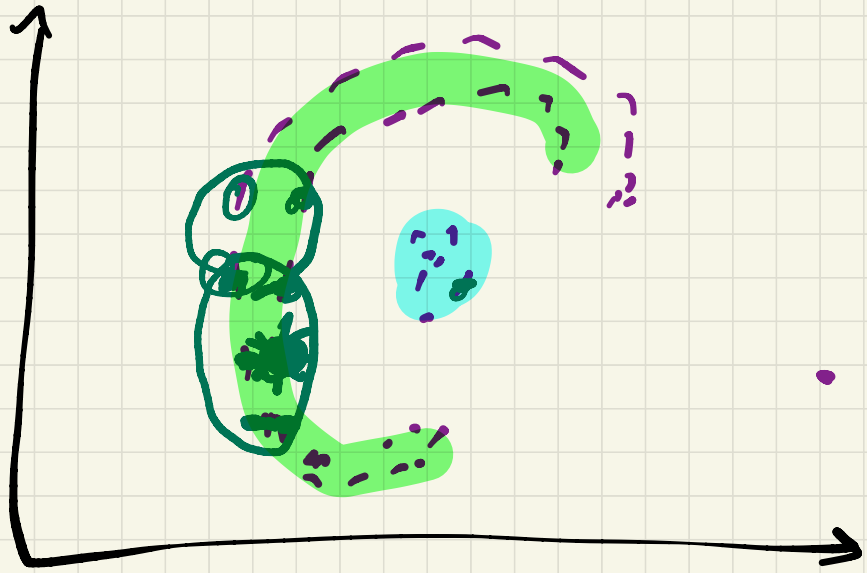
1. e.g., k=3
2. Randomly choose cluster centers
3. Recompute cluster centers
4. Repeat 2 & 3, ...



Density-Based Clustering (e.g., DBSCAN)

Concept: Regions with high density in same cluster.

1. Point is in a cluster if it has a min. # of "neighbors"
2. Repeat for all point's neighbors
3. Pick new random point



k-means failure scenarios