Warm-up as you login

First

1) Rename to append house #

2) Open and log into Piazza

3) Download lecture slides from ~i060l website

Search internet to find:

1. Machine learning classification dataset
   - Discrete/unordered output
2. Machine learning regression dataset
   - Continuous output

Are the input values discrete/continuous?

Are the input values a single value or multiple values?

What assumptions might we make with these datasets?

Announcements

Help us help you

- Student survey (see Piazza)
- Name pronunciation (via Canvas)

Assignments:

- HW1
  - Out today
  - Due Thu, 9/10, 11:59 pm (all times will be Pittsburgh time)

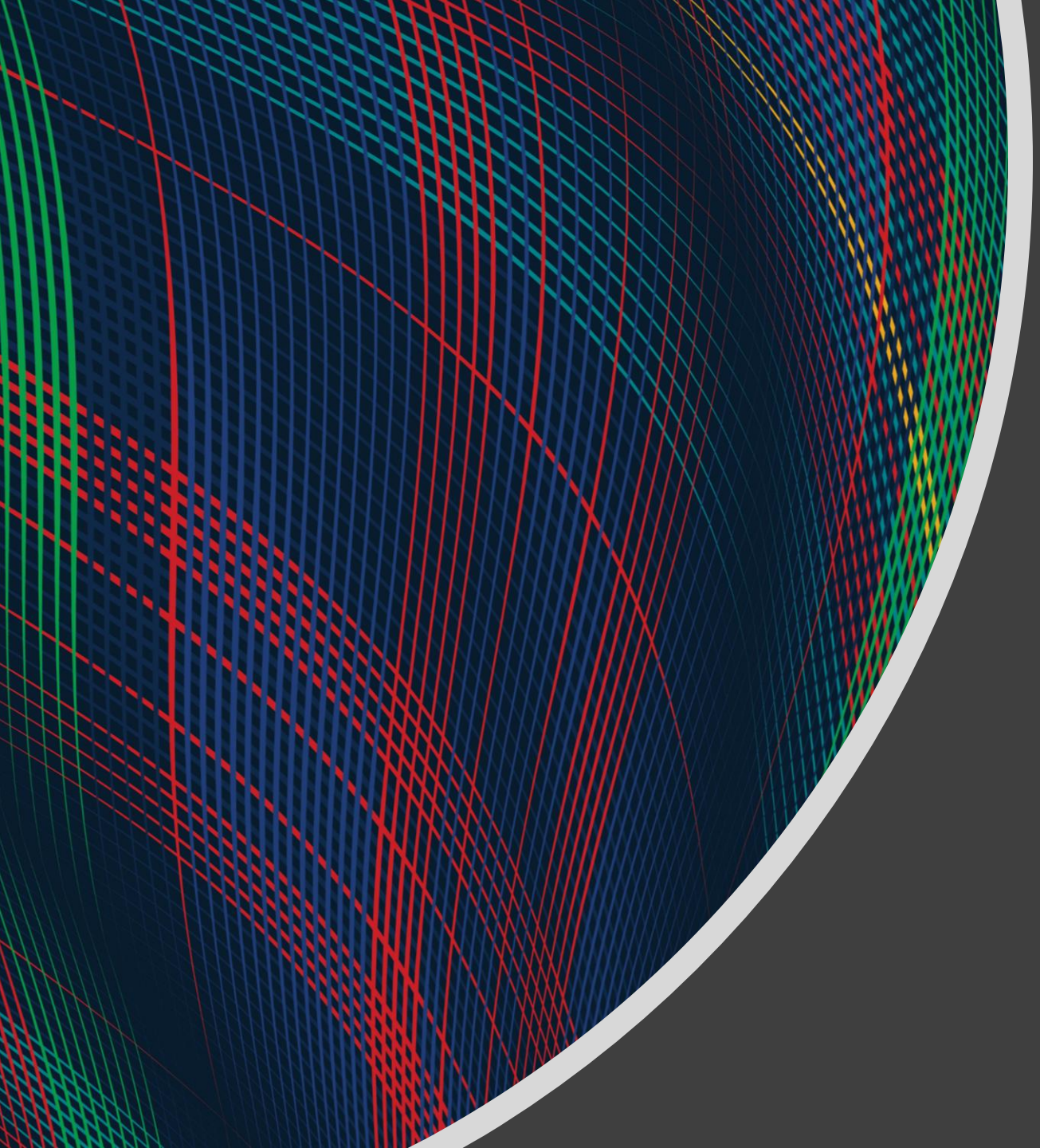
No class Monday: Labor Day

Q&A

Q: In Lecture 1, why did we use the term **experience** instead of just **data**?

A: Because our concern isn't just the data itself, but also where the data comes from (e.g. an agent interacting with the world vs. knowledge from a book).

As well, the word *experience* better aligns with the notion of what humans require in order to learn.



Introduction to Machine Learning

Decision Trees

Instructor: Pat Virtue

Plan

Today

- Problem formulation (notation)
- Algorithm 0: Memorization
- Algorithm 1: Majority vote
- Algorithm 2: Decision Stump

HW1

~~Monday~~ Wednesday

- Decision trees

Well-Posed Learning Problems

Three components *<T,P,E>*:

1. Task, *T*
2. Performance measure, *P*
3. Experience, *E*

Definition of learning:

A computer program **learns** if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.

Definition from (Mitchell, 1997)

Problem Formulation

Experience $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ $x \in \mathcal{X}$ $y \in \mathcal{Y}$ } Train Data / Test Data

output, class label, class, label

input, features, attributes

Regression: $y \in \mathbb{R}$

Classification: y discrete (and not ordered) set

Hypothesis

function: $\hat{y} = h(x)$ $h: \mathcal{X} \rightarrow \mathcal{Y}$

Hypothesis space $h \in \mathcal{H}$ $\hat{y} = \underline{m}x + \underline{b}$

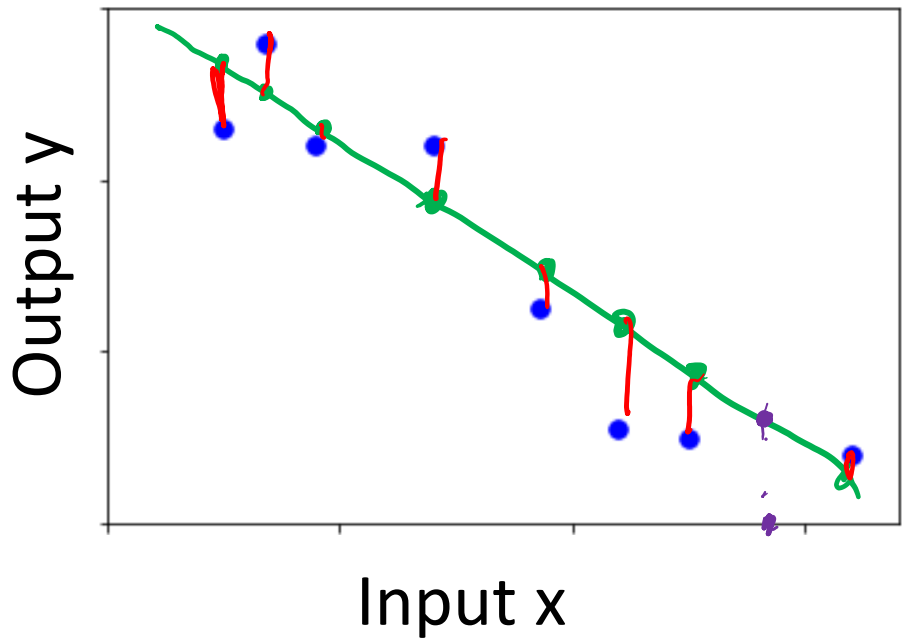
Performance measure

Error

Loss

Objective function

Problem Formulation



Regression

$x \in \mathbb{R}$

$y \in \mathbb{R}$

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

$\hat{y}^{(i)} = h(x^{(i)})$

$3x+2$

$4x-5$

$-32x+7$

Error $y - \hat{y}$

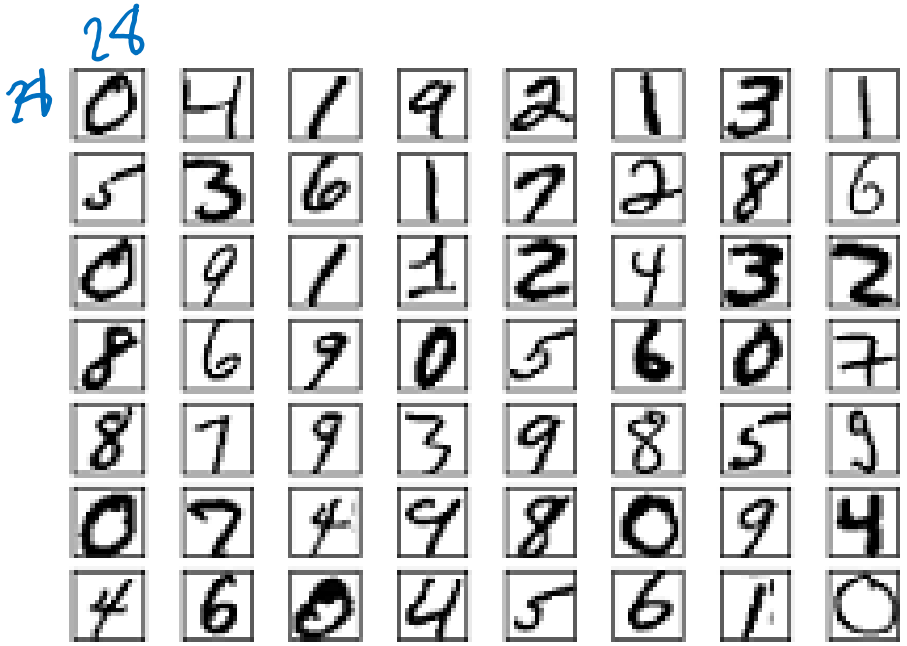
$$\sum_{i=1}^{N} \left(y^{(i)} - \hat{y}^{(i)}\right) \rightarrow \sum \left|y^{(i)} - \hat{y}^{(i)}\right| \rightarrow \sum \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

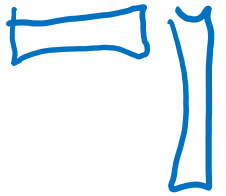
sum sq error

$\frac{1}{N}\sum (y^i - \hat{y}^i)^2$

mean sq error

Problem Formulation





Classification

$y \in \{0, 1, \ldots 9\}$

$\vec{x} \in \mathbb{R}^{784}$ $\quad$ $\vec{x} \in \{0,1\}^{784}$

$\hat{y} = h(\vec{x}) = \vec{m}^T \vec{x} + b$

Error

$$\text{Loss}(y, \hat{y}) = \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & \text{otherwise} \end{cases}$$

$$= \mathbb{1}(y \neq \hat{y})$$

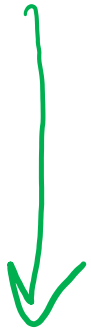
Error Rate

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y^{(i)} \neq \hat{y}^{(i)})$$

Problem Formulation

Medical Prediction

Outcome	Fetal Position	Fetal Distress	Previous C-sec
Natural	Vertex	N	N
C-section	Breech	N	N
Natural	Vertex	Y	Y
C-section	Vertex	N	Y
Natural	Abnormal	N	N



Problem Formulation

Medical Prediction

Y	X_1	X_2	X_3
Outcome	**Fetal Position**	**Fetal Distress**	**Previous C-sec**
Natural	Vertex	N	N
C-section	Breech	N	N
Natural	Vertex	Y	Y
C-section	Vertex	N	Y
Natural	Abnormal	N	N

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [x_1, x_2, x_3]^T$$

$x_1 \in \{Vertex, Breech, Abn\}$
$x_2 \in \{Y, N\}$
$x_3 \in \{Y, N\}$

$y \in \{Csection, Natural\}$

$\hat{y} = h(\boldsymbol{x})$

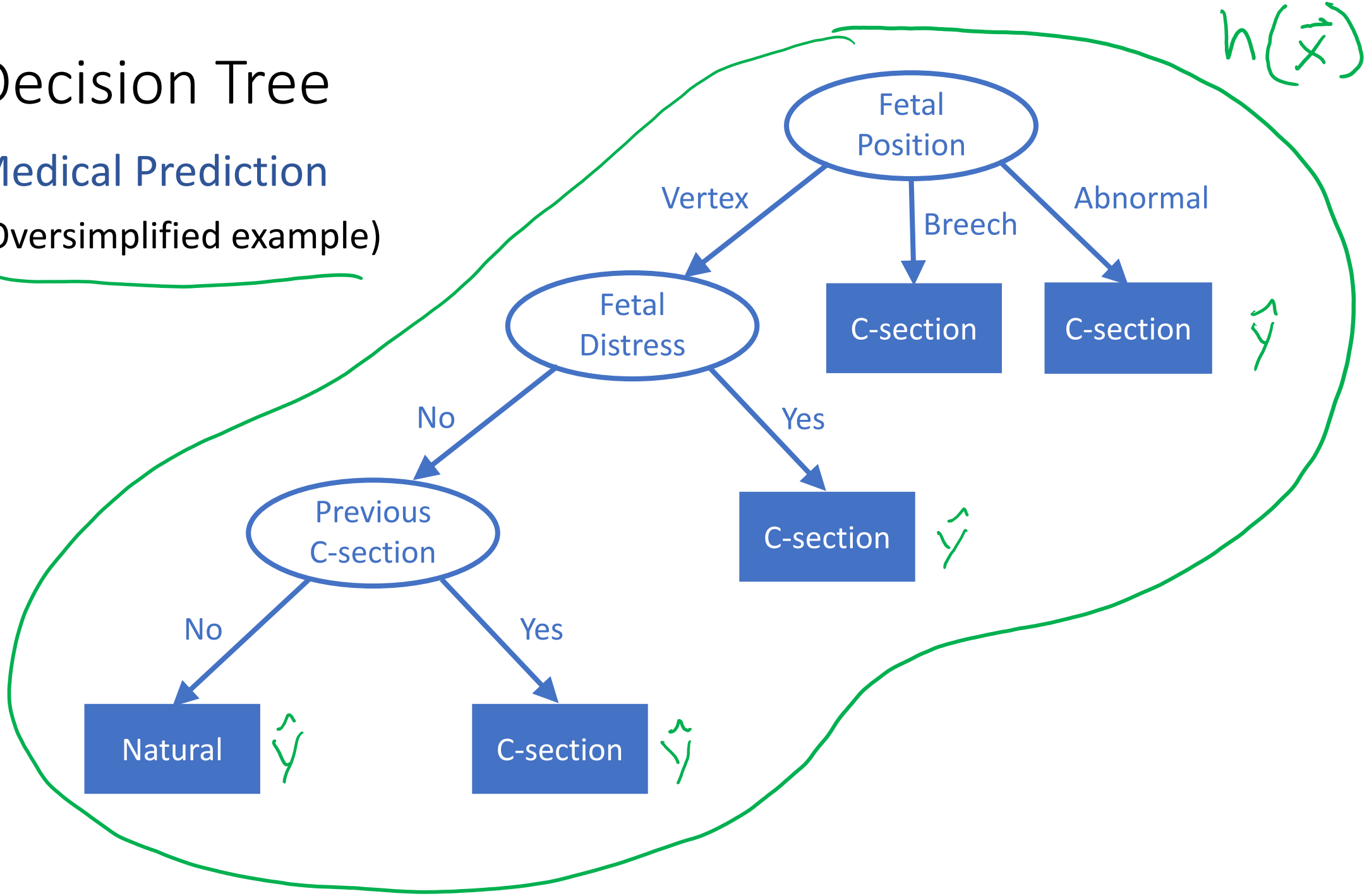
$$Loss(y, \hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & 0.w \end{cases} \longrightarrow Loss(y, \hat{y}) =$$

	$\hat{y}=0$	$\hat{y}=1$
$y=0$	0	100
$y=1$	10	0

Decision Tree

Medical Prediction

(Oversimplified example)



How could we implement training and prediction?

Algorithm 0: Memorization algorithm

def train(D)
    store D

def h(x)
    if $\exists\, x^{(i)} \in D$ s.t. $x^{(i)} = x$
        return $y^{(i)}$
    else
        return random $y \in \mathcal{Y}$

Slide credit: CMU MLD Matt Gormley

Piazza Poll 1

Does the memorization algorithm learn?

A. Yes 34%

B. No 61%

C. I have no clue

Slide credit: CMU MLD Matt Gormley

How could we implement training and prediction?

Algorithm 1: Majority vote algorithm

def train(D)

store v = majority_vote(D)

= class y that appears most in D

def h(x)

return ✓

Piazza Poll 2

What does the majority vote algorithm return on this training data?

A. A

B. B

C. C

D. 0

E. 1

F. +

G. -

Dataset:

Output Y, Attributes A, B, C

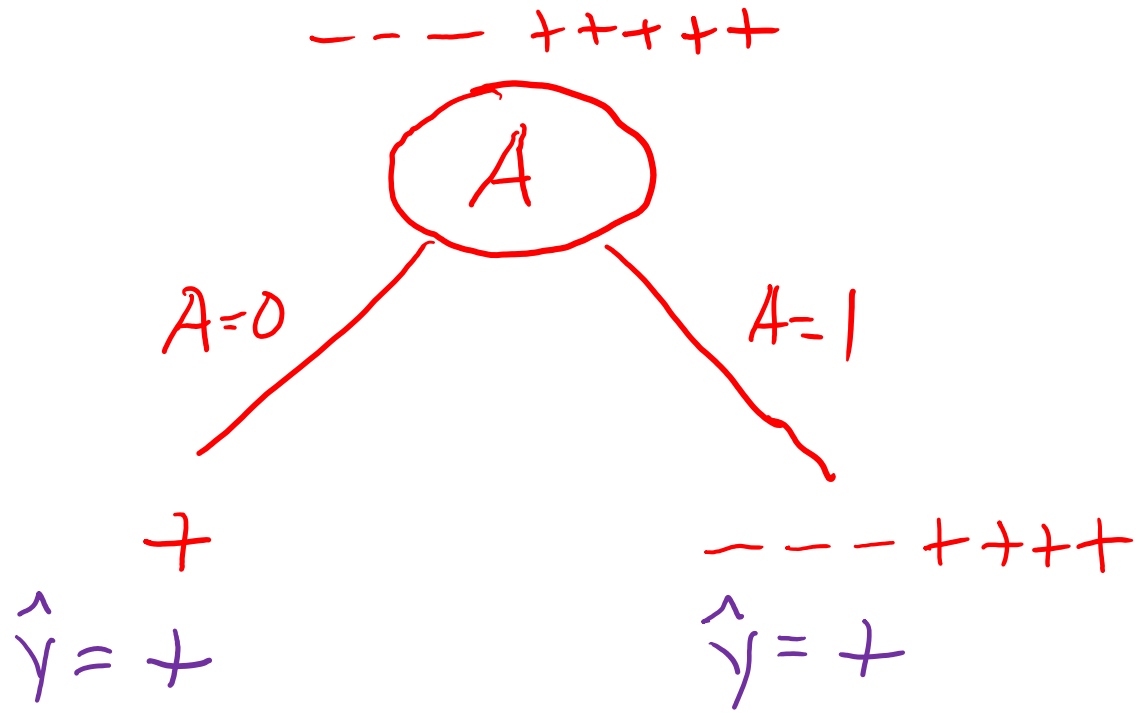
Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Slide credit: CMU MLD Matt Gormley

Decision Stumps

Split data based on a single attribute

Majority vote at leaves



Dataset:

Output Y, Attributes A, B, C

Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Slide credit: CMU MLD Matt Gormley

How could we implement training and prediction?

Algorithm 2: Decision stump algorithm

def train(D)

① pick an attribute X_m

② Divide dataset as

$$D^{(0)} = \{(\vec{x}, y) \in D \mid X_m = 0\}$$

$$D^{(1)} = \{(\vec{x}, y) \in D \mid X_m = 1\}$$

③ two votes

$$v^{(0)} = \text{majority}(D^{(0)})$$

$$v^{(1)} = \text{majority}(D^{(1)})$$

def h($\vec{x}$)

if $x_m = 0$

return $v^{(0)}$

if $x_m = 1$

return $v^{(1)}$

Slide credit: CMU MLD Matt Gormley

Piazza Poll 3

Splitting on which attribute {A, B, C} creates a decision stump with the lowest training error?

Dataset:

Output Y, Attributes A, B, C

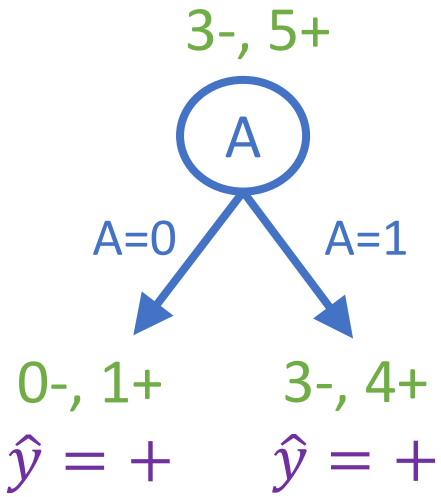
Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Slide credit: CMU MLD Matt Gormley

Piazza Poll 3

Splitting on which attribute {A, B, C} creates a decision stump with the lowest training error?

Answer: B



Error rate: (0 + 3) / 8
= 3/8

Dataset:

Output Y, Attributes A, B, C

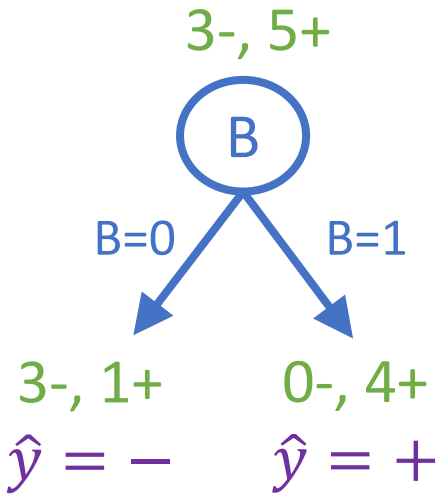
Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Slide credit: CMU MLD Matt Gormley

Piazza Poll 3

Splitting on which attribute {A, B, C} creates a decision stump with the lowest training error?

Answer: B



Error rate: (1 + 0) / 8
= 1/8

Dataset:

Output Y, Attributes A, B, C

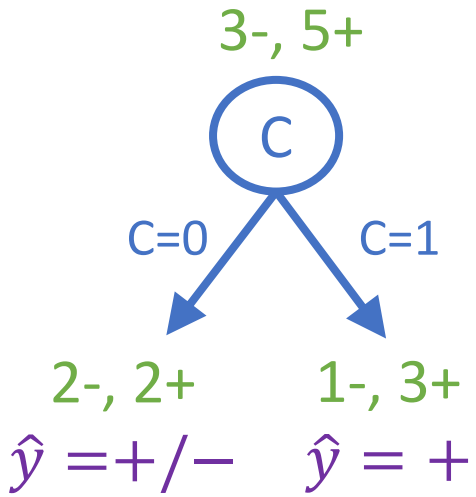
Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Slide credit: CMU MLD Matt Gormley

Piazza Poll 3

Splitting on which attribute {A, B, C} creates a decision stump with the lowest training error?

Answer: B



Error rate: (2 + 1) / 8
= 3/8

Dataset:

Output Y, Attributes A, B, C

Y	A	B	C
-	1	0	0
-	1	0	1
-	1	0	0
+	0	0	1
+	1	1	0
+	1	1	1
+	1	1	0
+	1	1	1

Slide credit: CMU MLD Matt Gormley