10-301/601: Introduction to Machine Learning Lecture 2 – Decision Trees: Model Definition

Henry Chai

5/16/23

Front Matter

- Announcements:
  - PA0 released 5/15, due 5/18 at 11:59 PM
    - You must complete all assignments using LaTeX; see this Piazza post for details and a few LaTeX tutorials
  - General advice for the summer:
    - Start HWs early!
    - Go to office hours!
      - MWThF (every weekday except Tuesday) from 5 – 6 PM in NSH 3002
- Recommended Readings:
  - Daumé III, Chapter 1: Decision Trees

Recall: Our first Machine Learning Classifier

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the **training** dataset

labels

data points

Heart Disease?
No
No
Yes
Yes
Yes

- This classifier completely ignores the features…

Recall: Our first Machine Learning Classifier

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the **training** dataset

labels

data points

Heart Disease?	Predictions
No	Yes
No	Yes
Yes	Yes
Yes	Yes
Yes	Yes

- The training error rate is $2/5$

Our second Machine Learning Classifier

- A **classifier** is a function that takes feature values as input and outputs a label
- Memorizer: if a set of features exists in the **training** dataset, predict its corresponding label; otherwise, predict the majority vote

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 second Machine Learning Classifier

- A **classifier** is a function that takes feature values as input and outputs a label
- Memorizer: if a set of features exists in the **training** dataset, predict its corresponding label; otherwise, predict the majority vote

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?	Predictions
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

- The training error rate is 0!

When survey is active, respond at **pollev.com/301601polls**

Lecture 2 Polls

0 done

0 underway

Is the memorizer learning?

Yes

No

Our second Machine Learning Classifier

- A **classifier** is a function that takes feature values as input and outputs a label
- Memorizer: if a set of features exists in the **training** dataset, predict its corresponding label; otherwise, predict the majority vote
- The memorizer (typically) does not **generalize** well, i.e., it does not perform well on unseen data points
- In some sense, good generalization, i.e., the ability to make accurate predictions given a small training dataset, is the whole point of machine learning!

Notation

- Feature space, $\mathcal{X}$
- Label space, $\mathcal{Y}$
- (Unknown) Target function, $c^*: \mathcal{X} \rightarrow \mathcal{Y}$
- Training dataset:

$$\mathcal{D} = \{(\boldsymbol{x}^{(1)}, c^*(\boldsymbol{x}^{(1)}) = y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}) \dots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$$

- Data point:

$$(\boldsymbol{x}^{(n)}, y^{(n)}) = \left(x_1^{(n)}, x_2^{(n)}, \dots, x_D^{(n)}, y^{(n)}\right)$$

- Classifier, $h : \mathcal{X} \rightarrow \mathcal{Y}$
- Goal: find a classifier, h, that best approximates c^*

Evaluation

- Loss function, $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
  - Defines how "bad" predictions, $\hat{y} = h(\boldsymbol{x})$, are compared to the true labels, $y = c^*(\boldsymbol{x})$
  - Common choices

1. Squared loss (for regression): $\ell(y, \hat{y}) = (y - \hat{y})^2$
2. Binary or 0-1 loss (for classification):

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

Evaluation

- Loss function, $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
  - Defines how "bad" predictions, $\hat{y} = h(\boldsymbol{x})$, are compared to the true labels, $y = c^*(\boldsymbol{x})$
  - Common choices

1. Squared loss (for regression): $\ell(y, \hat{y}) = (y - \hat{y})^2$
2. Binary or 0-1 loss (for classification):

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

- Error rate:

$$\text{err}(h, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} \ell\left(h\left(x^{(n)}\right), y^{(n)}\right)$$

$$= \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\left(h\left(x^{(n)}\right) \neq y^{(n)}\right)$$

Notation: Example

- Memorizer: if a set of features exists in the **training** dataset, predict its corresponding label; otherwise, predict the majority vote

	x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?	$\hat{y}$ Predictions
	Yes	Low	Normal	No	No
$\boldsymbol{x}^{(2)}$	No	Medium	Normal	No	No
	No	Low	Abnormal	Yes	Yes
	Yes	Medium	Normal	Yes	Yes
	Yes	High	Abnormal	Yes	Yes

- $N = 5$ and $D = 3$
- $\boldsymbol{x}^{(2)} = \left(x_1^{(2)} = \text{"No"}, x_2^{(2)} = \text{"Medium"}, x_3^{(2)} = \text{"Normal"}\right)$

Our second Machine Learning Classifier: Pseudocode

```
def train(D_train):
    store D_train
def majority_vote(D_train):
    return mode(y^(1), y^(2), ..., y^(N))

def predict(x'):
    if ∃ x^(i) ∈ D_train s.t. x^(i) = x'
        return y^(i)
    else
        return majority_vote(D_train)
```

Our third Machine Learning Classifier

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

- Decision stump: based on a single feature, x_d, predict the most common label in the training dataset among all data points that have the same value for x_d

Our third Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

- Decision stump on x_1:

$$h(\boldsymbol{x}') = h(x'_1, \ldots, x'_D) = \begin{cases} ??? & \boxed{\text{if } x'_1 = \text{“Yes”}} \\ ??? & \text{otherwise} \end{cases}$$

Our third Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

- Decision stump on x_1:

$$h(\boldsymbol{x}') = h(x_1', \dots, x_D') = \begin{cases} \text{“Yes” if } x_1' = \text{“Yes”} \\ ??? \; \boxed{\text{otherwise}} \end{cases}$$

Our third Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

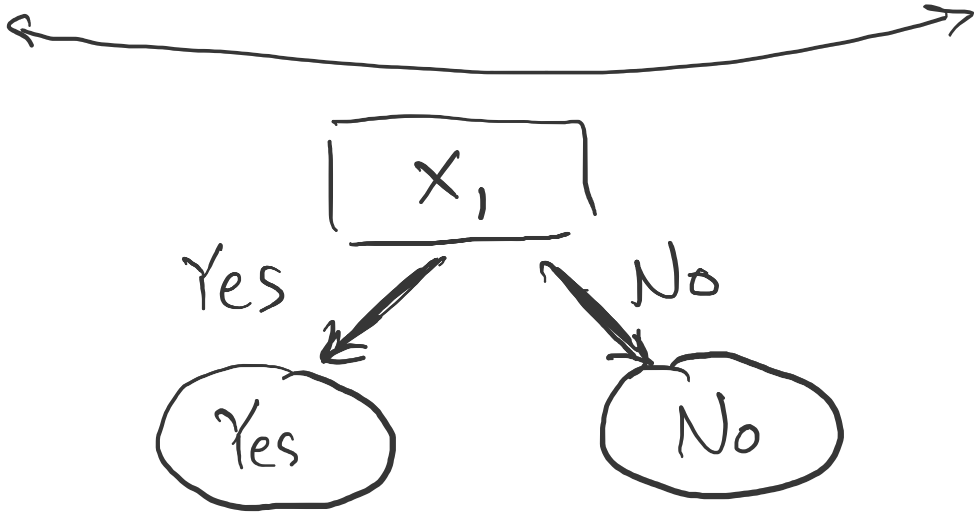
- Decision stump on x_1:

$$h(\boldsymbol{x}') = h(x'_1, \dots, x'_D) = \begin{cases} \text{“Yes” if } x'_1 = \text{“Yes”} \\ \text{“No” otherwise} \end{cases}$$

Our third Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?	$\hat{y}$ Predictions
Yes	Low	Normal	No	Yes
No	Medium	Normal	No	No
No	Low	Abnormal	Yes	No
Yes	Medium	Normal	Yes	Yes
Yes	High	Abnormal	Yes	Yes



Decision Stumps: Pseudocode

def train(D_{train}):

1. Pick a feature x_d to split on
2. for all v in $V(x_d)$:

   $D_v = \{(x^{(i)}, y^{(i)}) \in D_{train} \mid x_d^{(i)} = v\}$
3. for all v in $V(x_d)$:

   compute a majority vote over D_v

   $\hat{y}_v$ = majority_vote(D_v)

def predict(x'):

  for v in $V(x_d)$:

    if $x'_d = v$:

      return $\hat{y}_v$

Decision Stumps: Questions

1. How can we pick which feature to split on?

Which feature do you think we should split on for this data set?

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

x_1

x_2

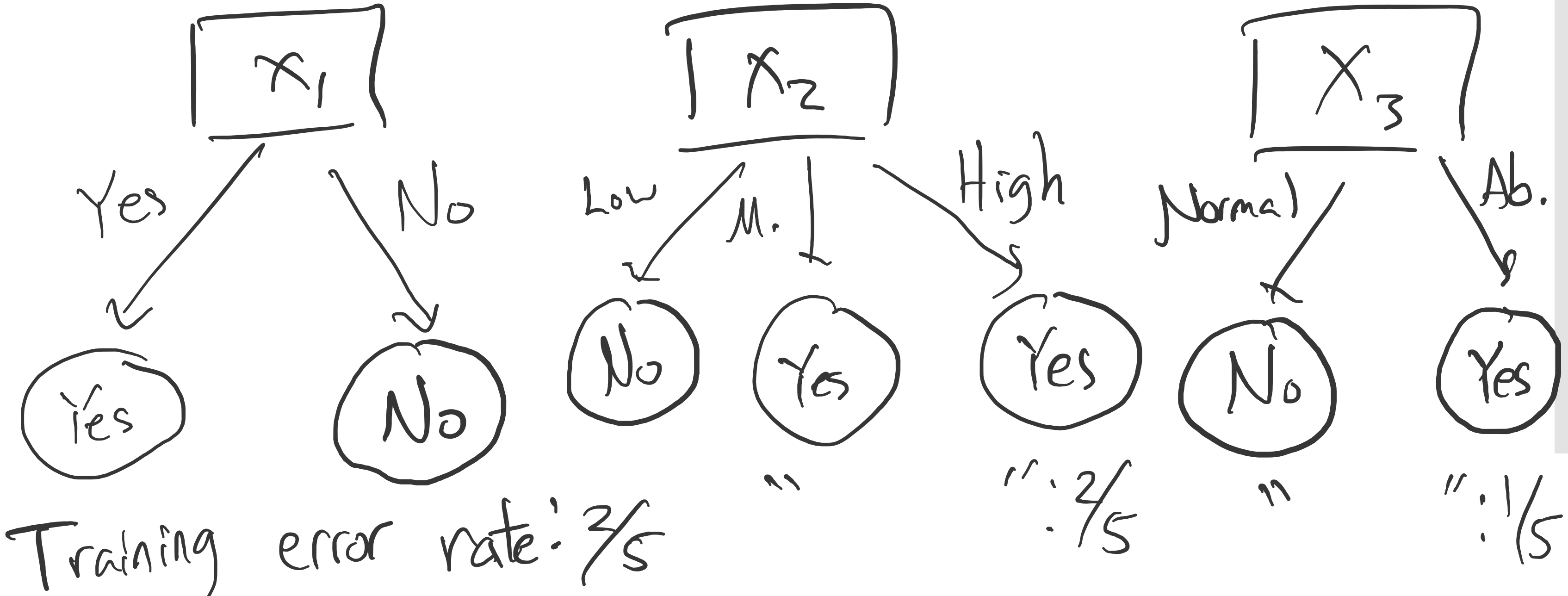
x_3

Splitting Criterion

- A **splitting criterion** is a function that measures how good or useful splitting on a particular feature is *for a specified dataset*
- Insight: use the feature that optimizes the splitting criterion for our decision stump.

Training error rate as a Splitting Criterion

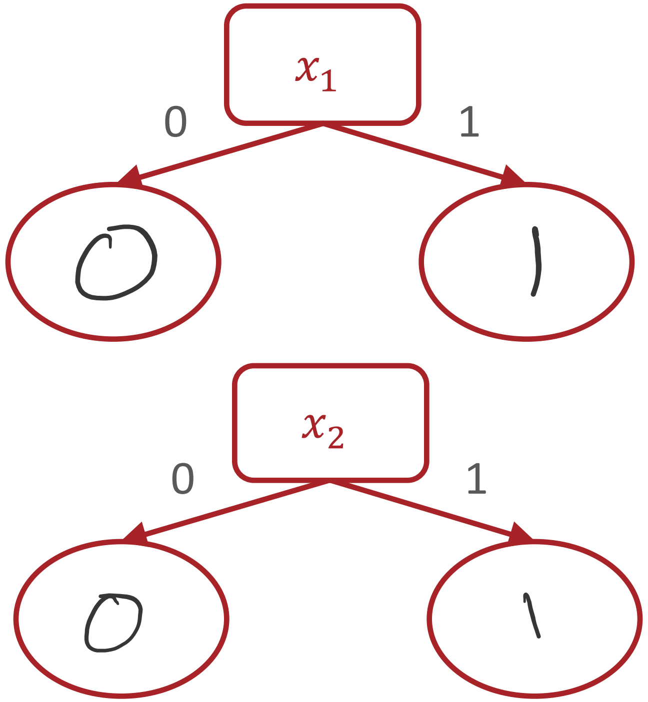
x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes



Training error rate as a Splitting Criterion?

x_1	x_2	y
1	0	0
1	0	0
1	0	1
1	0	1
1	1	1
1	1	1
1	1	1
1	1	1

- Which feature would you split on using training error rate as the splitting criterion?



training error rate: 2/8

Splitting Criterion

- A **splitting criterion** is a function that measures how good or useful splitting on a particular feature is *for a specified dataset*
- Insight: use the feature that optimizes the splitting criterion for our decision stump.
- Potential splitting criteria:
  - Training error rate (minimize)
  - Gini impurity (minimize) → CART algorithm
  - Mutual information (maximize) → ID3 algorithm

Splitting Criterion

- A **splitting criterion** is a function that measures how good or useful splitting on a particular feature is *for a specified dataset*
- Insight: use the feature that optimizes the splitting criterion for our decision stump.
- Potential splitting criteria:
  - Training error rate (minimize)
  - Gini impurity (minimize) → CART algorithm
  - Mutual information (maximize) → ID3 algorithm

Entropy

- Entropy describes the purity or uniformity of a collection of values: the lower the entropy, the more pure

$$H(S) = -\sum_{v \in V(S)} \frac{|S_v|}{|S|} \log_2\left(\frac{|S_v|}{|S|}\right)$$

where S is a collection of values,

$V(S)$ is the set of unique values in S

S_v is the collection of elements in S with value v

- If all the elements in S are the same, then

$$H(S) = -\frac{N}{N}\log_2\left(\frac{N}{N}\right) = -1 \log_2 1 = 0$$

Entropy

- Entropy describes the purity or uniformity of a collection of values: the lower the entropy, the more pure

$$H(S) = -\sum_{v \in V(S)} \frac{|S_v|}{|S|} \log_2\left(\frac{|S_v|}{|S|}\right)$$

where S is a collection of values,

$V(S)$ is the set of unique values in S

S_v is the collection of elements in S with value v

- If S is split fifty-fifty between two values, then

$$H(S) = -\frac{N}{2N}\log_2\frac{N}{2N} - \frac{N}{2N}\log_2\frac{N}{2N}$$

$$= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = \frac{1}{2} + \frac{1}{2} = 1$$

Mutual Information

- Mutual information describes how much information or clarity a particular feature provides about the label

$$I(x_d; Y) = H(Y) - \sum_{v \in V(x_d)} (f_v)\left(H\left(Y_{x_d=v}\right)\right)$$

where x_d is a feature

Y is the collection of all labels

$V(x_d)$ is the set of unique values of x_d

f_v is the fraction of inputs where $x_d = v$

$Y_{x_d=v}$ is the collection of labels where $x_d = v$

Mutual Information: Example

x_d	y
1	1
1	1
0	0
0	0

$$I(x_d; y) = H(Y) - \sum_{v \in V(x_d)} (f_v) H(Y_{x_d = v})$$

$$= 1 - \frac{2}{4} \underbrace{H(Y_{x_d=1})} - \frac{2}{4} \underbrace{H(Y_{x_d=0})}$$

$$= 1 - \frac{2}{4}(0) - \frac{2}{4}(0) = 1$$

Mutual Information: Example

x_d	y
1	1
0	1
1	0
0	0

$$I(x_d ; Y) = H(Y) - \sum_{v \in V(x_d)} (f_v) H(Y_{x_d = v})$$

$$= 1 - \frac{2}{4} \underbrace{H(Y_{x_d=1})} - \frac{2}{4} H(Y_{x_d=0})$$

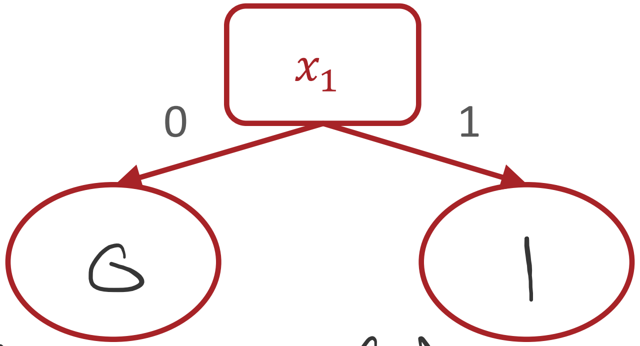
$$= 1 - \frac{2}{4}(1) - \frac{2}{4}(1) = 1 - \frac{1}{2} - \frac{1}{2} = 0$$

Mutual Information as a Splitting Criterion

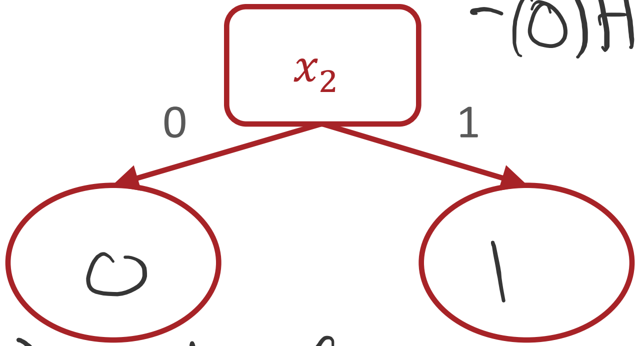
x_1	x_2	y
1	0	0
1	0	0
1	0	1
1	0	1
1	1	1
1	1	1
1	1	1
1	1	1

$X_2 = 1$

- Which feature would you split on using mutual information as the splitting criterion?



$$I(x_1;Y) = H(Y) - (1)H(Y_{x_1=1}) - (0)H(Y_{x_1=0}) = 0$$

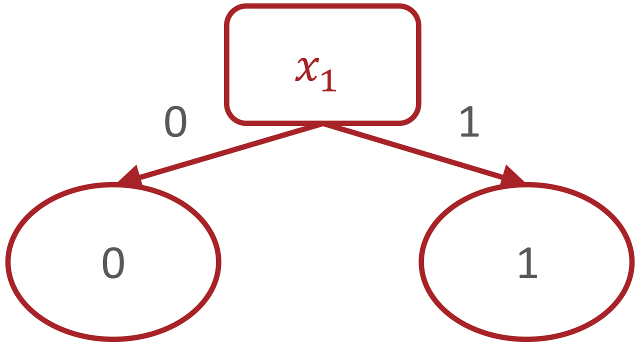


$$I(x_2;Y) = H(Y) - \frac{4}{8}H(Y_{x_2=1}) - \frac{4}{8}H(Y_{x_2=0}) > 0$$

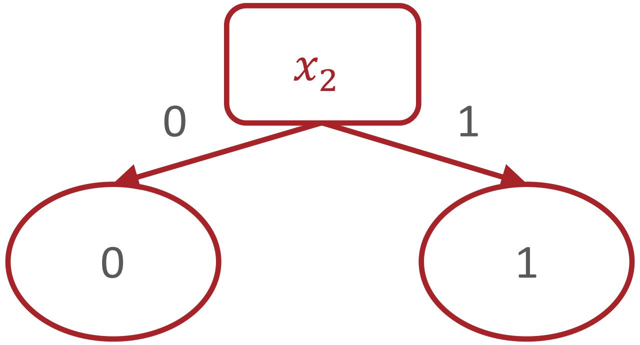
Mutual Information as a Splitting Criterion

x_1	x_2	y
1	0	0
1	0	0
1	0	1
1	0	1
1	1	1
1	1	1
1	1	1
1	1	1

- Which feature would you split on using mutual information as the splitting criterion?



Mutual Information: 0



Mutual Information: $-\frac{2}{8}\log_2\frac{2}{8}-\frac{6}{8}\log_2\frac{6}{8}-\frac{1}{2}(1)-\frac{1}{2}(0)\approx 0.31$

Decision Stumps: Questions

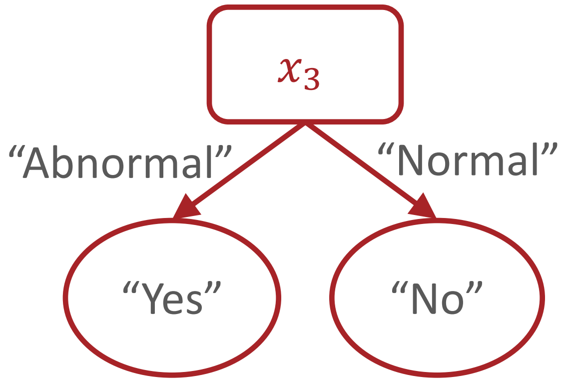
1. How can we pick which feature to split on?

   Maximize the mutual information.

2. Why stop at just one feature?

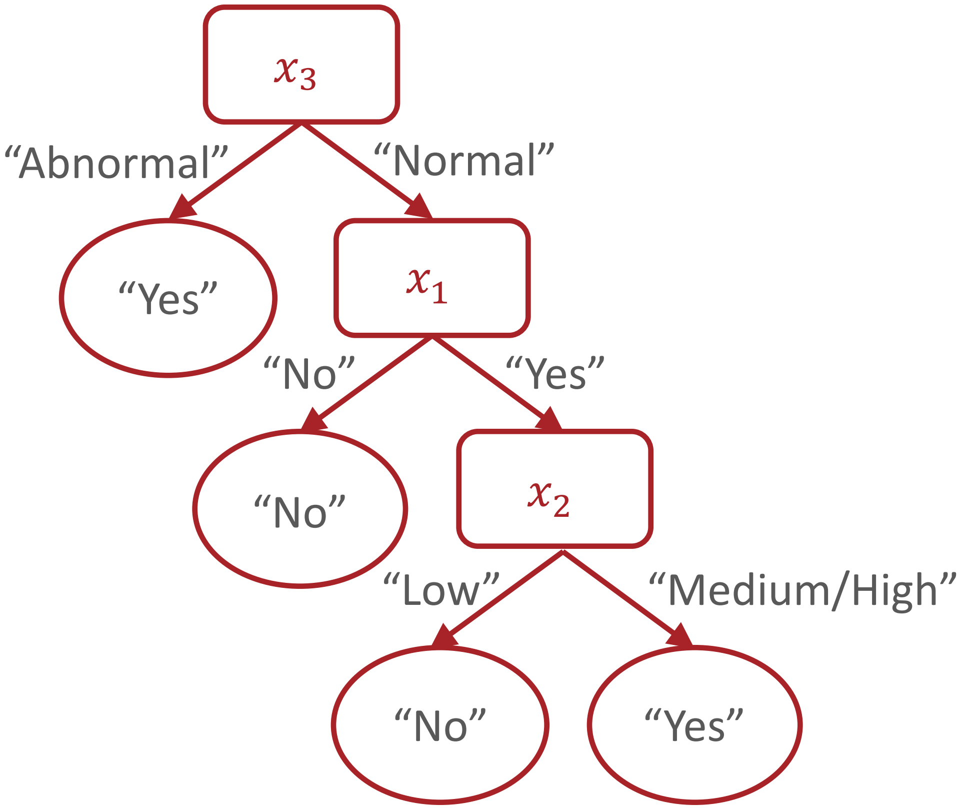
From Decision Stump …

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes



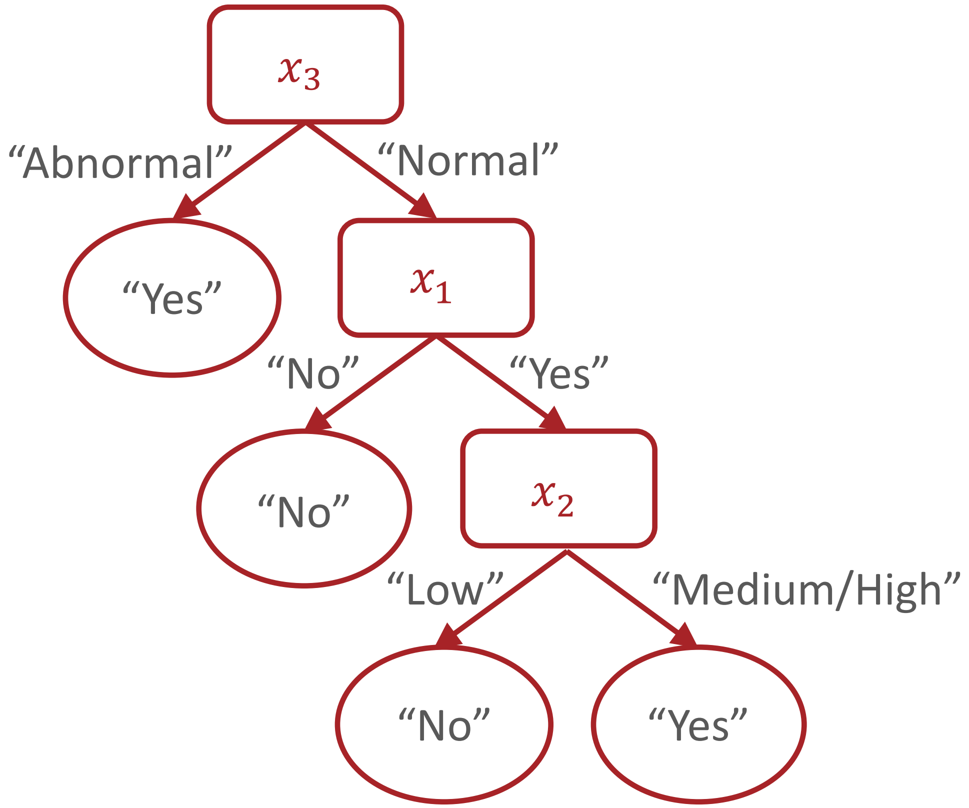
From Decision Stump to Decision Tree

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes



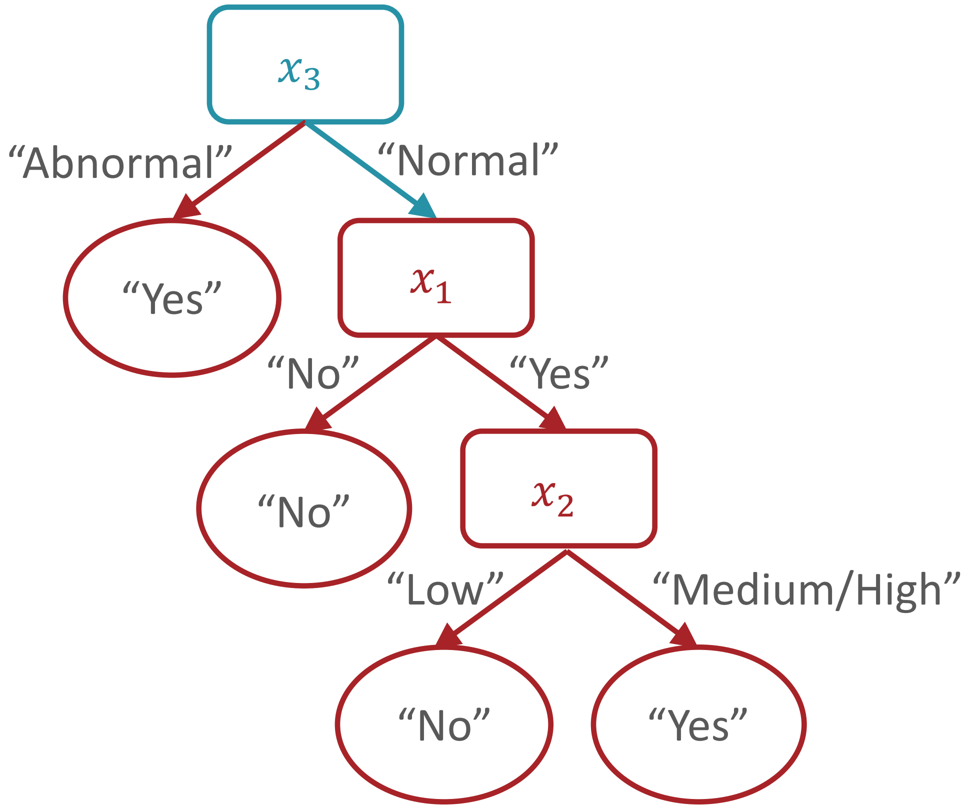
From Decision Stump to Decision Tree

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes
No	High	Normal	No



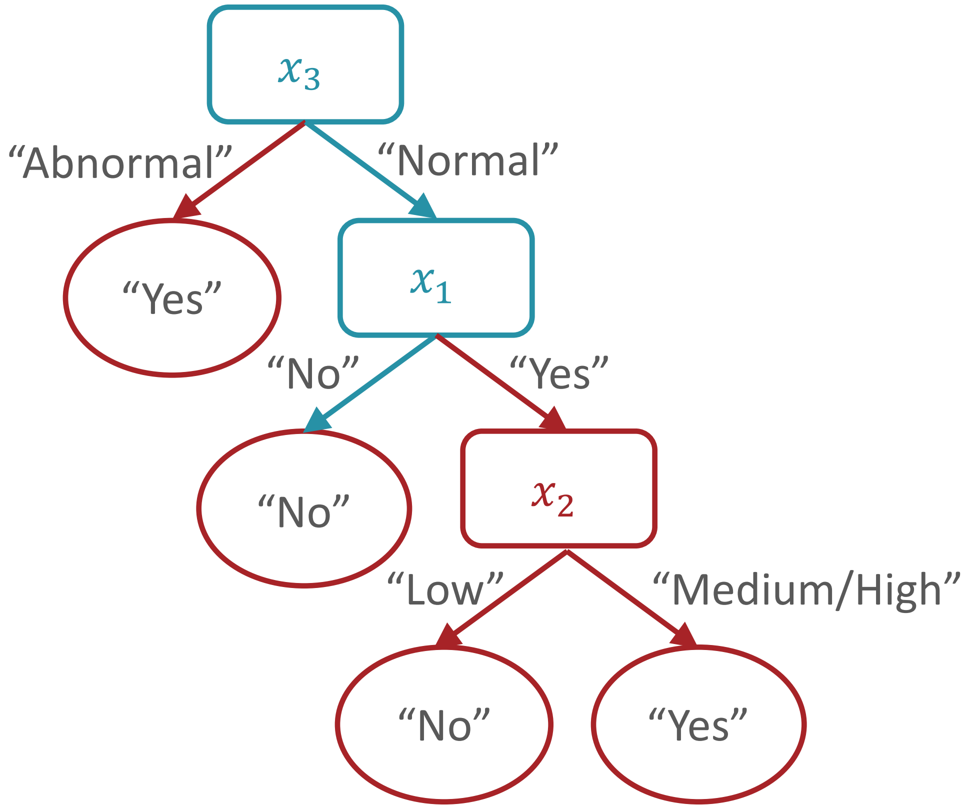
From Decision Stump to Decision Tree

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes
No	High	Normal	No



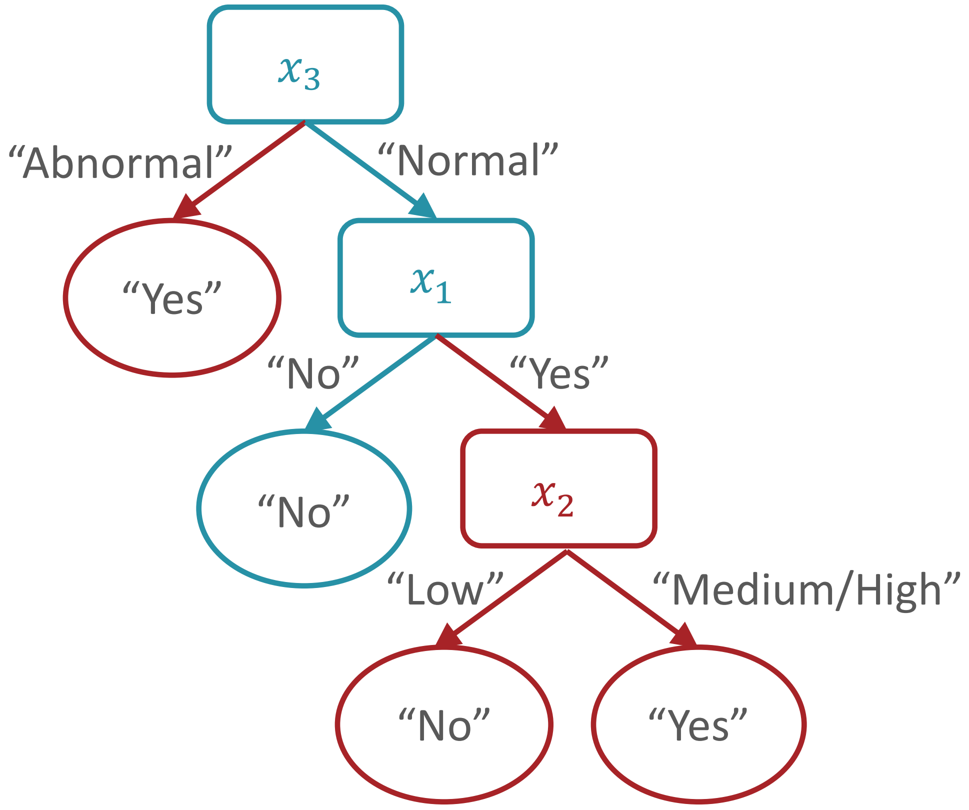
From Decision Stump to Decision Tree

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes
No	High	Normal	No



From Decision Stump to Decision Tree

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes
No	High	Normal	No



Decision Tree: Example

Learned from medical records of 1000 women

Negative examples are C-sections

```
                                  [833+,167-] .83+ .17-
depth 0 → Fetal_Presentation = 1: [822+,116-] .88+ .12-
depth 1 → | Previous_Csection = 0: [767+,81-] .90+ .10-
depth 2 → | | Primiparous = 0: [399+,13-] .97+ .03-
depth 2 → | | Primiparous = 1: [368+,68-] .84+ .16-
        → | | | Fetal_Distress = 0: [334+,47-] .88+ .12-
        → | | | Fetal_Distress = 1: [34+,21-] .62+ .38-
depth 1 → | Previous_Csection = 1: [55+,35-] .61+ .39-
depth 0 → Fetal_Presentation = 2: [3+,29-] .11+ .89-
depth 0 → Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

Figure courtesy of Tom Mitchell

Key Takeaways

- Memorization as a form of learning
- Generalization
- Mutual information as a splitting criterion for decision stumps/trees
- Decision tree prediction algorithm