10-701: Introduction to Machine Learning

Lecture 2 – Decision Trees

Hoda Heidari

8/27/2025

* Slides adopted from F24 offering of 10701 by Henry Chai.

Notation

- Feature space, $\mathcal{X}$
- Label space, $\mathcal{Y}$
- (Unknown) Target function, $c^*: \mathcal{X} \rightarrow \mathcal{Y}$
- Training dataset: $\mathcal{D}= \{<\boldsymbol{x}^{(1)}, y^{(1)}> ,..., <\boldsymbol{x}^{(N)}, y^{(N)}> \}$
- Data point:
  $<\boldsymbol{x}^{(i)}, y^{(i)}> = <x_1^{(i)}, x_2^{(i)}, ..., x_D^{(i)}, y= c^*(\boldsymbol{x})>$
- Classifier, h : $\mathcal{X} \rightarrow \mathcal{Y}$
- Goal: find a classifier, h, that best approximates c^*

Notation

- Loss function, $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathrm{R}$
  - Defines how "bad" predictions, $\hat{y} = h(x)$ are
  - compared to the true labels, $y = c^*(\boldsymbol{x})$
- Common choices
  - Binary or 0-1 loss (for classification):
    $$\ell(y, \hat{y}) = \mathbf{1}[y \neq \hat{y}]$$
  - Squared loss (for regression):
    $$\ell(y, \hat{y}) = (y - \hat{y})^2$$
- Error rate:
  $$Err(h, D) = \frac{1}{N} \sum_{i=1} \ell\left(y^{(i)}, \hat{y}^{(i)}\right)$$

A Typical (Supervised) Machine Learning Routine

- Step 1 – **training**
  - Input: a labelled training dataset
  - Output: a classifier
- Step 2 – **testing**
  - Inputs: a classifier, a test dataset
  - Output: predictions for each test data point
- Step 3 – **evaluation**
  - Inputs: predictions from step 2, test dataset labels
  - Output: some measure of how good the predictions are; usually (but not always) error rate

Sample Classifiers

- **Majority vote classifier:** always predict the most common label in the dataset
- **Memorizer:** if the input feature vector exists in the **training** dataset, predict its corresponding label; otherwise, predict the majority vote
- **Decision stump** using a specific feature.

Recall: Our second 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

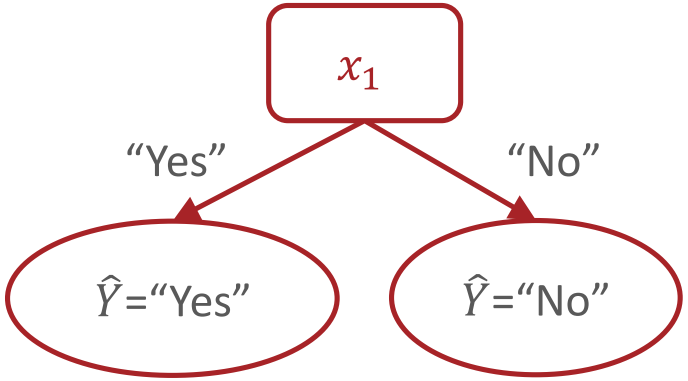
- Decision stump on x_1:

$$h(\boldsymbol{x}') = h(x_1', \dots, x_D') = \begin{cases} \text{“Yes” if } x_1' = \text{“Yes”} \\ \text{“No” otherwise} \end{cases}$$

Recall: Our second 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 Stumps: Questions

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

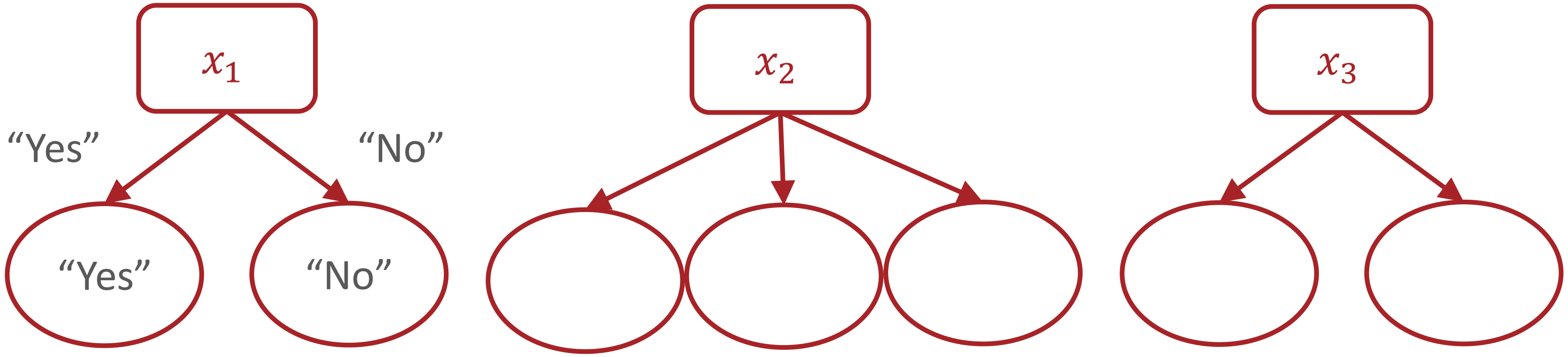
2. Why stop at just one feature?

Splitting Criterion

- A **splitting criterion** is a function that measures how good or useful splitting on a particular feature is *for a specified dataset*
- Idea: 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: 2/5

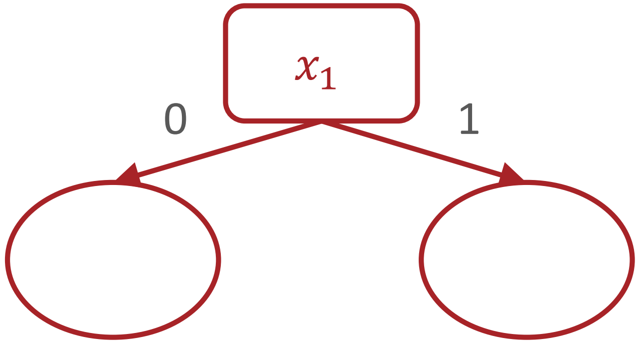
Training error rate:

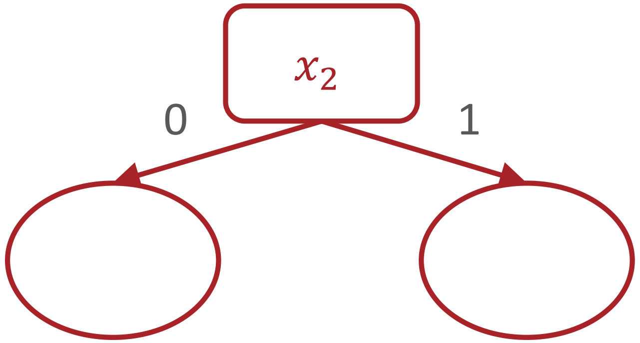
Training error rate:

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?





Splitting Criterion

- A **splitting criterion** is a function that measures how good or useful splitting on a particular feature is *for a specified dataset*
- Idea: 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*
- Idea: 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 of a (discrete) random variable X that takes on values in $\mathcal{X}$:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2\big(p(x)\big)$$

Entropy is a measure of randomness, uncertainty, disorder.

Example: biased vs. fair coin

Entropy

- Entropy of a collection of values S:

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

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

S_v is the collection of elements in S with value v

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

$$H(S) = -1 \log_2(1) = 0$$

Entropy

- Entropy of a collection of values S:

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

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

S_v is the collection of elements in S with value v

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

$$H(S) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = -\log_2\left(\frac{1}{2}\right) = 1$$

Mutual Information

- Mutual information between two random variables X and Y describes how much clarity about the value of one variable is gained by observing the other

$$I(Y;X) = H(Y) - H(Y|X)$$

Where $H(Y|X) = \sum_x p(x)\, H(Y|X = x)$

$$= -\sum_x p(x) \sum_y \frac{p(x,y)}{p(x)} \log_2 \left(\frac{p(x,y)}{p(x)}\right)$$

$$= -\sum_{x,y} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)}\right)$$

Mutual Information

- Mutual information can be used to compute how much information or clarity a particular feature provides about the label

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

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

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

$$= 1 - \frac{1}{2} H\left(Y_{x_d=0}\right) - \frac{1}{2} H\left(Y_{x_d=1}\right)$$

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

Mutual Information: Example

x_d	y
1	1
0	1
1	0
0	0

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

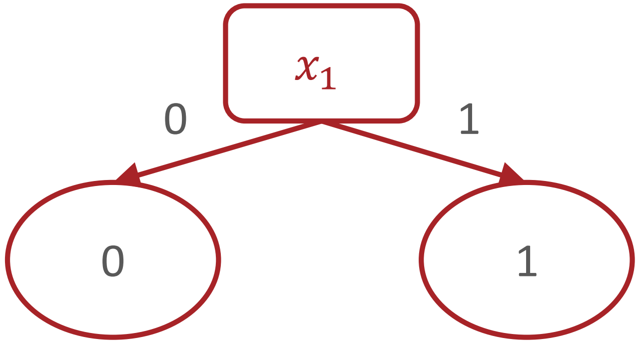
$$= 1 - \frac{1}{2} H\left(Y_{x_d=0}\right) - \frac{1}{2} H\left(Y_{x_d=1}\right)$$

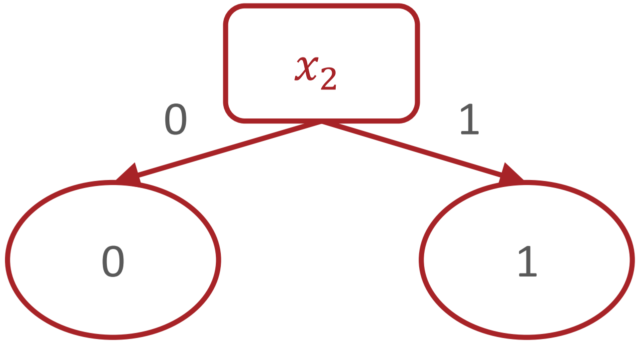
$$= 1 - \frac{1}{2}(1) - \frac{1}{2}(1) = 0$$

Poll 1:

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?





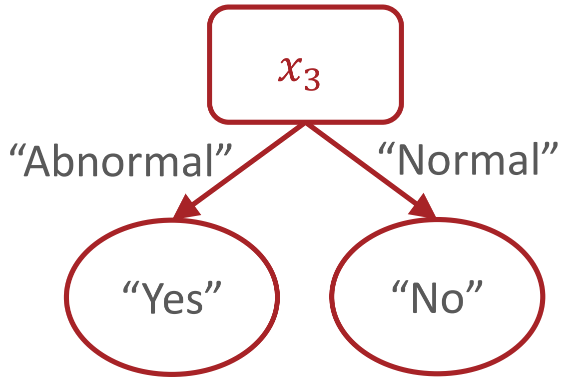
Decision Stumps: Questions

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

2. Why stop at just one feature?

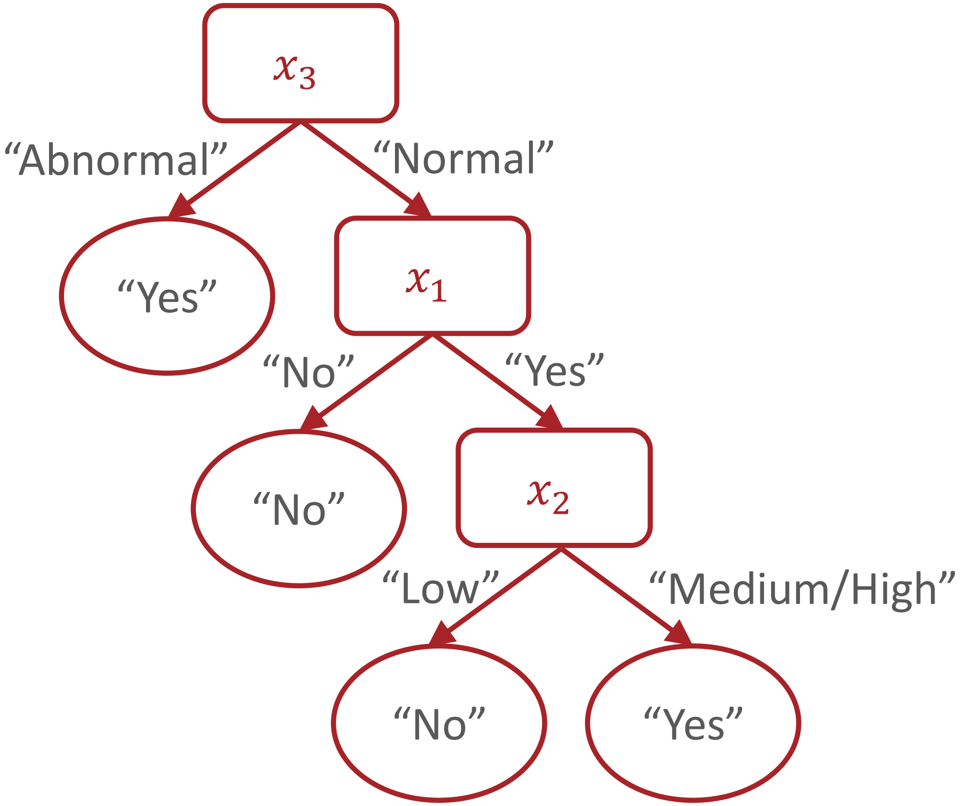
From Decision Stump …

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



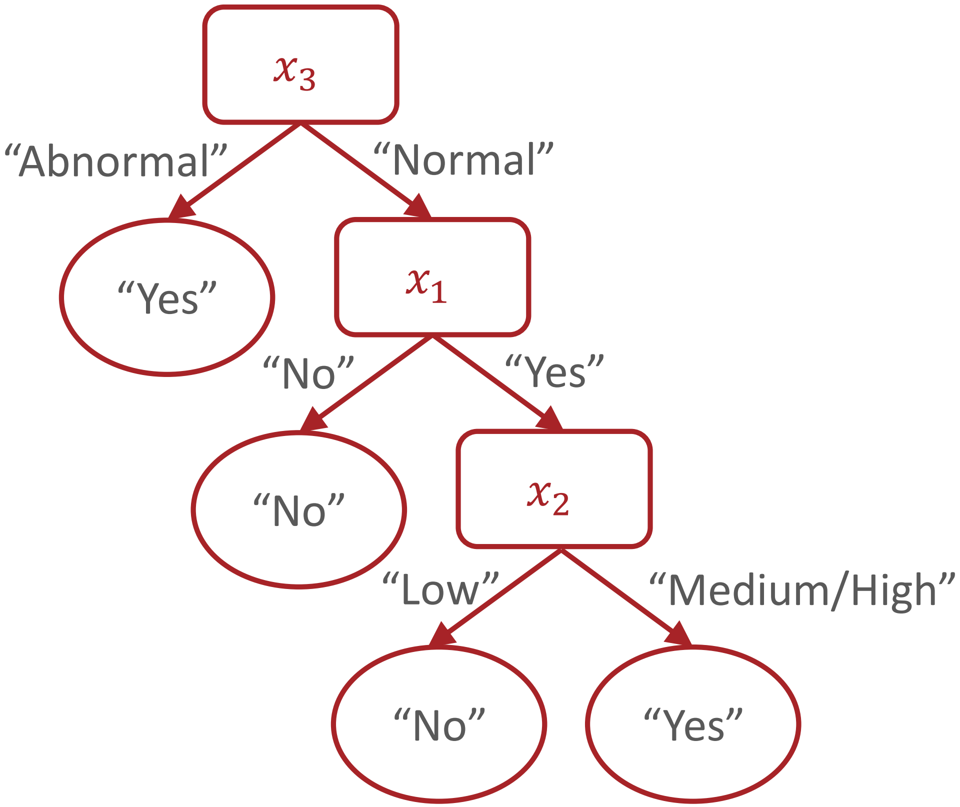
From Decision Stump to Decision Tree

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



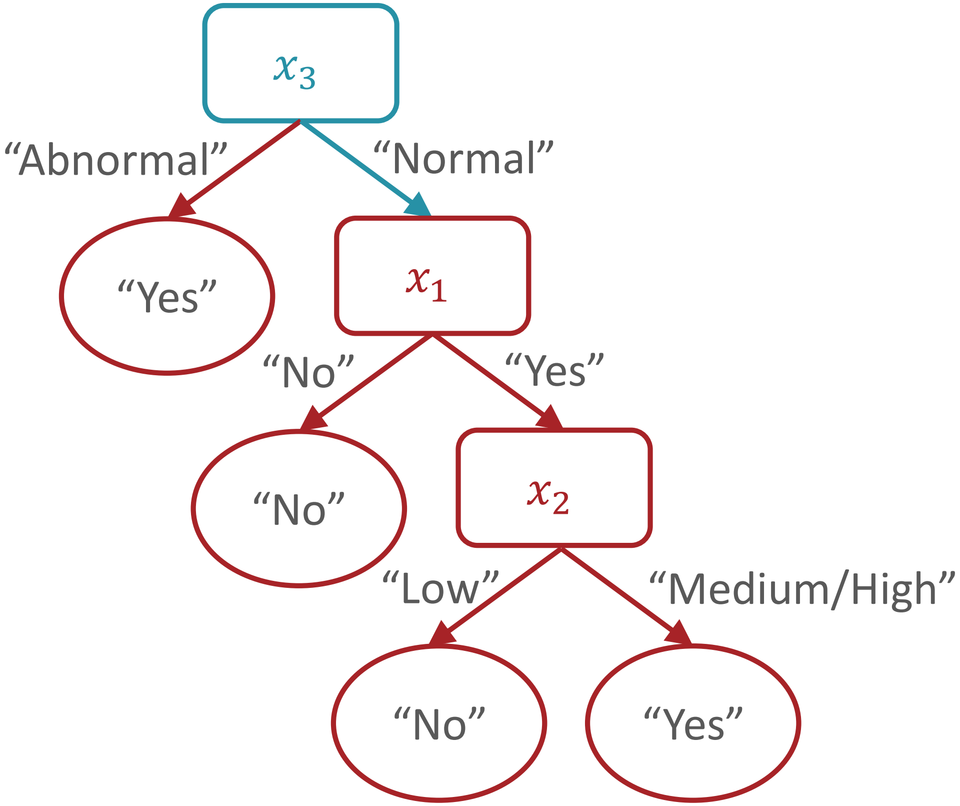
From Decision Stump to Decision Tree

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



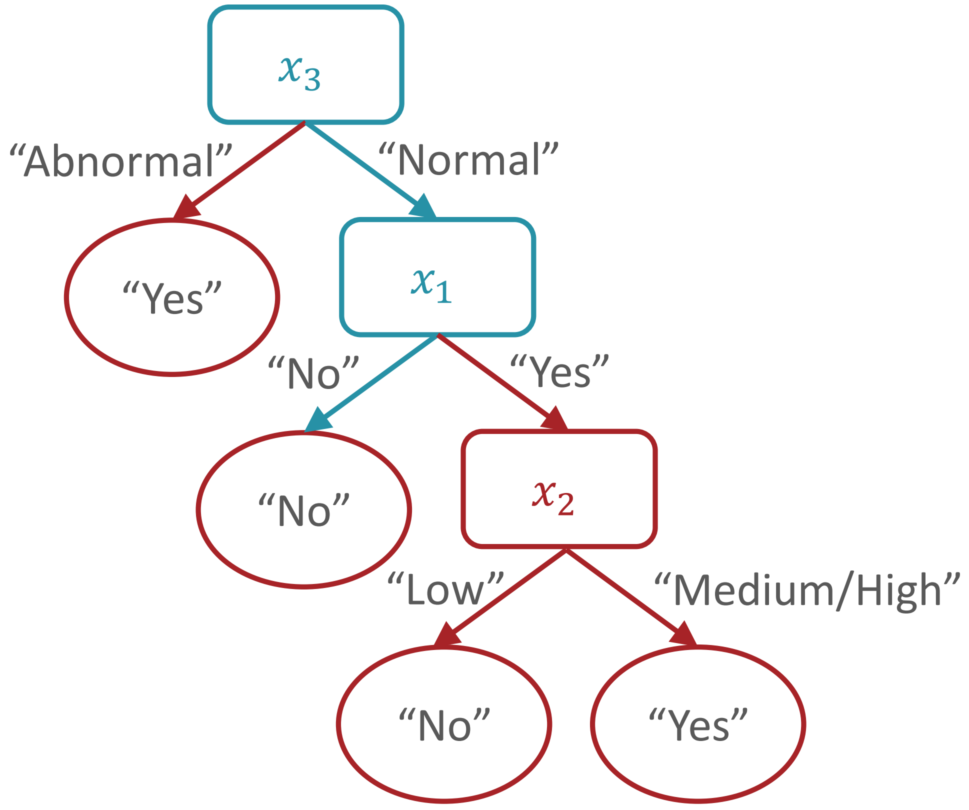
From Decision Stump to Decision Tree

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



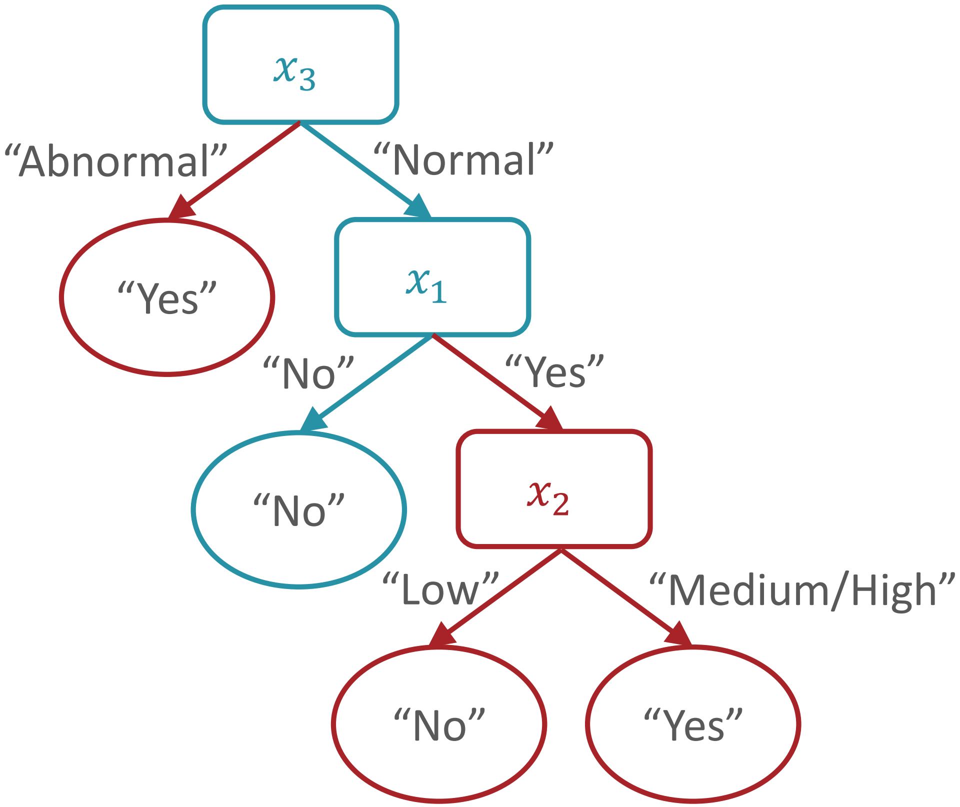
From Decision Stump to Decision Tree

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 Prediction: Pseudocode

```
def predict(x'):
  - walk from root node to a leaf node
    while(true):
       if current node is internal (non-leaf):
            check the associated attribute, x_d
            go down branch according to x'_d
       if current node is a leaf node:
            return label stored at that leaf
```

Decision Tree Learning: ID3 Algorithm

1. Start with the entire training dataset, $\mathcal{D}$.
2. For each attribute, x_d, calculate the information gain if the dataset were split using that attribute.
3. Select the attribute with the highest information gain as the splitting attribute for the current node.
4. Create a new node in the decision tree with this attribute.
5. For each possible value of the chosen attribute, create a new branch and a corresponding subset of the data.
6. Recursively apply steps 2-6 to each subset until a stopping criteria is met:
   - All examples in the subset have the same label.
   - There are no more attributes to split on.
   - There are no more examples in the subset.
   - …

Decision Tree Learning: Pseudocode

```
def train(𝒟):
    store root = tree_recurse(𝒟)
def tree_recurse(𝒟′):
    q = new node()
    base case – if (SOME CONDITION):
    recursion – else:
        find best attribute to split on, x_d
        q.split = x_d
        for v in V(x_d), all possible values of x_d:
```

$$\mathcal{D}_v = \left\{\left(x^{(n)}, y^{(n)}\right) \in \mathcal{D} \mid x_d^{(n)} = v\right\}$$

```
            q.children(v) = tree_recurse(𝒟_v)
    return q
```

Decision Tree: Pseudocode

```
def train(𝒟):
    store root = tree_recurse(𝒟)
def tree_recurse(𝒟′):
    q = new node()
    base case – if (𝒟′ is empty OR
       all labels in 𝒟′ are the same OR
       all features in 𝒟′ are identical OR
       some other stopping criterion):
       q.label = majority_vote(𝒟′)


    recursion – else:
    return q
```

Decision Tree: Example (Iteratively)

- How am I getting to work?
- Label: mode of transportation
  - $y \in \mathcal{Y} = \{\text{Bike}, \text{Drive}, \text{Bus}\}$
- Features: 4 categorial features
  - Is it raining? $x_1 \in \{\text{Rain}, \text{No Rain}\}$
  - When am I leaving (relative to rush hour)?
    $x_2 \in \{\text{Before}, \text{During}, \text{After}\}$
  - What am I bringing?
    $x_3 \in \{\text{Backpack}, \text{Lunchbox}, \text{Both}\}$
  - Am I tired? $x_4 \in \{\text{Tired}, \text{Not Tired}\}$

Data

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Not Tired	Bus
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Bus
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Bus
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Bus

Which feature would we split on first using mutual information as the splitting criterion?

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Not Tired	Bus
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Bus
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Bus
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Bus

Recall:

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

$H(Y)$

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Not Tired	Bus
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Bus
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Bus
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Bus

Recall:

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

$$H(Y) = -\frac{3}{16}\log_2\left(\frac{3}{16}\right)$$

$$-\frac{6}{16}\log_2\left(\frac{6}{16}\right)$$

$$= -\frac{7}{16}\log_2\left(\frac{7}{16}\right)$$

$$H(Y) \approx 1.5052$$

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Not Tired	Bus
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Bus
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Bus
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Bus

Recall: $I(x_d; Y) = H(Y)$

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

$$I(x_1, Y) \approx 1.5052$$

$$-\frac{6}{16}(1)$$

$$-\frac{10}{16}\left(-\frac{3}{10}\log_2\left(\frac{3}{10}\right)\right.$$

$$-\frac{3}{\ }\log_2\left(\frac{3}{\ }\right)$$

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Not Tired	Bus
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Bus
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Bus
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Bus

Recall: $I(x_d; Y) = H(Y)$

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

$$I(x_1, Y) \approx 1.5052$$

$$- \frac{6}{16}(1)$$

$$- \frac{10}{16}(1.5710)$$

$$\approx 0.1482$$

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Not Tired	Bus
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Bus
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Bus
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Bus

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

	$I(x_d, Y)$
x_1	0.1482
x_2	0.1302
x_3	0.5358
x_4	0.5576

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Not Tired	Bus
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Bus
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Bus
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Bus

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

$I(x_d, Y)$	
x_1	0.1482
x_2	0.1302
x_3	0.5358
x_4	0.5576

x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Bus
Rain	After	Backpack	Not Tired	Bus
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	During	Backpack	Not Tired	Bus
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Lunchbox	Not Tired	Bus
Rain	Before	Both	Tired	Drive
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Tired	Drive

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

$I(x_d, Y)$	
x_1	0.1482
x_2	0.1302
x_3	0.5358
x_4	0.5576

x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Bus
Rain	After	Backpack	Not Tired	Bus
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	During	Backpack	Not Tired	Bus
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Lunchbox	Not Tired	Bus
Rain	Before	Both	Tired	Drive
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Tired	Metro
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Tired	Drive

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

$I(x_d, Y)$	
x_1	0.1482
x_2	0.1302
x_3	0.5358
x_4	0.5576

x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Bus
Rain	After	Backpack	Not Tired	Bus
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	During	Backpack	Not Tired	Bus
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Lunchbox	Not Tired	Bus
Rain	Before	Both	Tired	Drive
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Tired	Drive

Decision Tree: Example

x_4

Not Tired

x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Bus
Rain	After	Backpack	Not Tired	Bus
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	During	Backpack	Not Tired	Bus
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Lunchbox	Not Tired	Bus

Tired

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Tired	Drive

x_4

Not Tired

x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Bus
Rain	After	Backpack	Not Tired	Bus
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	During	Backpack	Not Tired	Bus
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Lunchbox	Not Tired	Bus

Tired

x_1	x_2	x_3	x_4	y
Rain	Before	Both	Tired	Drive
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Tired	Bus
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Both	Tired	Drive
No Rain	After	Backpack	Tired	Bike
No Rain	After	Both	Tired	Drive

x_4

Not Tired

x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Bus
Rain	After	Backpack	Not Tired	Bus
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	During	Backpack	Not Tired	Bus
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Lunchbox	Not Tired	Bus

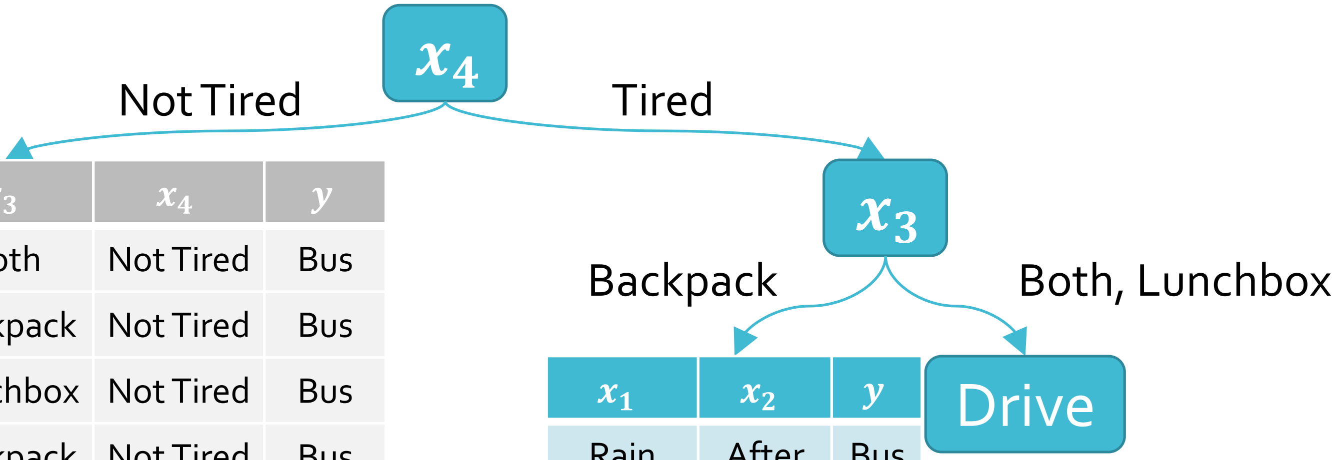
Tired

x_1	x_2	x_3	x_4	y
Rain	After	Backpack	Tired	Bus
No Rain	Before	Backpack	Tired	Bike
No Rain	After	Backpack	Tired	Bike
Rain	Before	Both	Tired	Drive
Rain	During	Both	Tired	Drive
No Rain	During	Both	Tired	Drive
No Rain	After	Both	Tired	Drive
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Lunchbox	Tired	Drive

$$I\left(x_1, Y_{x_4=\text{Tired}}\right) \approx 0.3244$$

$$I\left(x_2, Y_{x_4=\text{Tired}}\right) \approx 0.2516$$

$$I\left(x_3, Y_{x_4=\text{Tired}}\right) \approx \mathbf{0.9183}$$



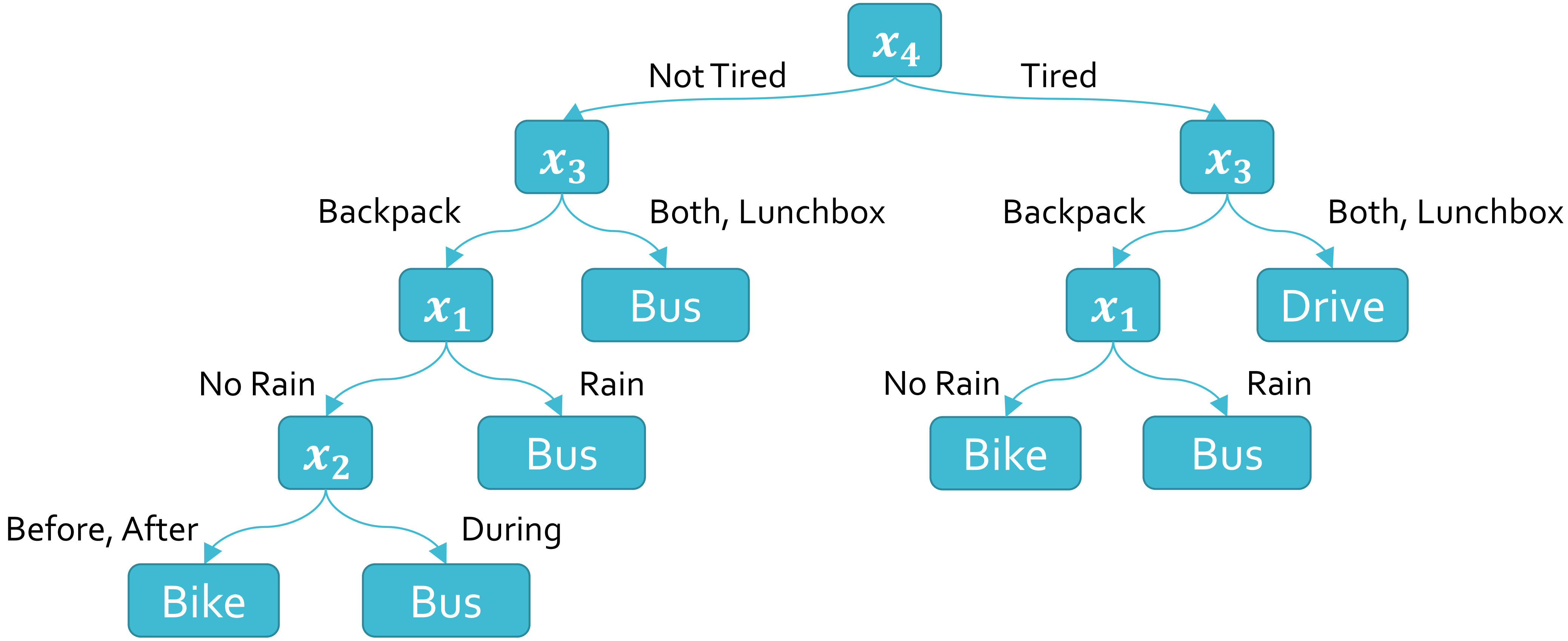
x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Bus
Rain	After	Backpack	Not Tired	Bus
No Rain	Before	Lunchbox	Not Tired	Bus
No Rain	During	Backpack	Not Tired	Bus
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Bus
No Rain	After	Lunchbox	Not Tired	Bus

x_1	x_2	y
Rain	After	Bus
No Rain	Before	Bike
No Rain	After	Bike

$$I\left(x_1, Y_{x_4=\text{Tired}}\right) \approx 0.3244$$

$$I\left(x_2, Y_{x_4=\text{Tired}}\right) \approx 0.2516$$

$$I\left(x_3, Y_{x_4=\text{Tired}}\right) \approx \mathbf{0.9183}$$


x_4
Not Tired
Tired
x_3
x_3
Backpack
Both, Lunchbox
Backpack
Both, Lunchbox
x_1
Bus
x_1
Drive
No Rain
Rain
No Rain
Rain
x_2
Bus
Bike
Bus
Before, After
During
Bike
Bus

Decision Trees: Inductive Bias

- The **inductive bias** of a machine learning algorithm is the principal by which it generalizes to unseen examples
- What is the inductive bias of the ID3 algorithm i.e., decision tree learning with mutual information maximization as the splitting criterion?
    - Try to find the ___________ tree that achieves ___________________________________ with ________________________________ features at the top

Decision Trees: Pros & Cons

- Pros
  - Interpretable
  - Efficient (computational cost and storage)
  - Can be used for classification and regression tasks
  - Compatible with categorical and real-valued features
- Cons

Decision Trees: Pros & Cons

- Pros
  - Interpretable
  - Efficient (computational cost and storage)
  - Can be used for classification and regression tasks
  - Compatible with categorical and real-valued features
- Cons
  - Learned greedily: each split only considers the immediate impact on the splitting criterion
    - Not guaranteed to find the smallest (fewest number of splits) tree that achieves a training error rate of 0.
  - Liable to overfit!