10-301/601: Introduction to Machine Learning Lecture 3 – Decision Trees: Learning

Henry Chai

5/17/23

Front Matter

- Announcements:
  - PA0 released 5/15, due 5/18 (tomorrow!) at 11:59 PM
    - You must complete all assignments using LaTeX; see this Piazza post for details and a few LaTeX tutorials
  - PA1 released 5/18 (tomorrow!)
  - Recitation tomorrow will cover
    - Programming tips to help you with PA1
    - Practice problems for Quiz 1 on 5/23
    - Recitations are optional but **they will not be recorded**; solutions will be made available afterwards
- Recommended Readings:
  - Daumé III, Chapter 1: Decision Trees

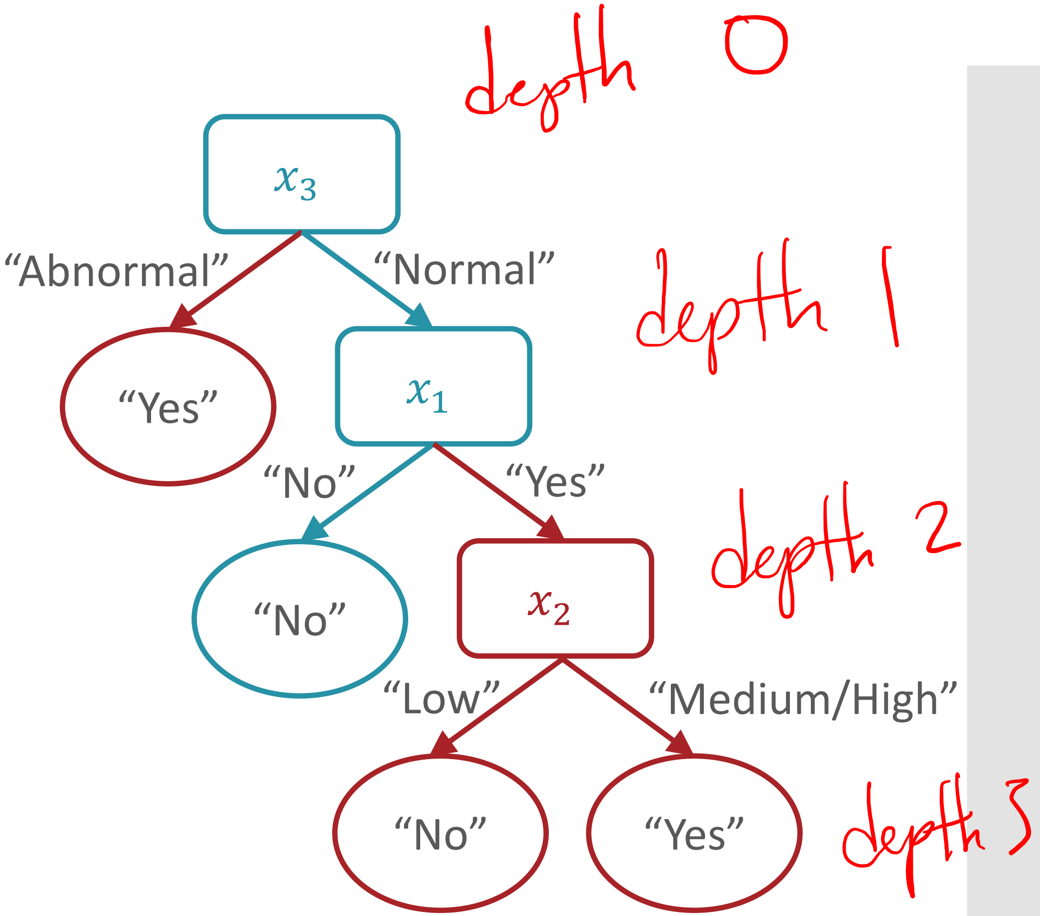
Recall: Decision Stumps Questions

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

2. Why stop at just one feature?

From Decision Stump to Decision Tree

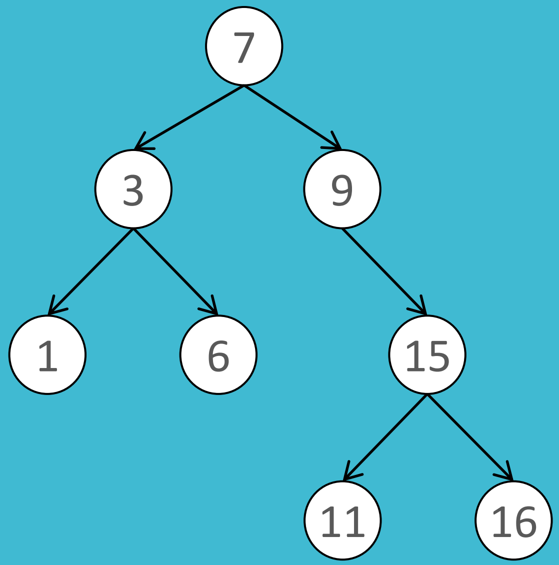
x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes
No	High	Normal	No



Decision Tree Prediction: Pseudocode

```
def predict(x'):
  - walk from the root node to a leaf
  while(true)
      if current_node is a split:
          check the associated feature
          x_d and go down the branch
          according to x'_d
      else:
          return the label stored at
          this leaf node
```

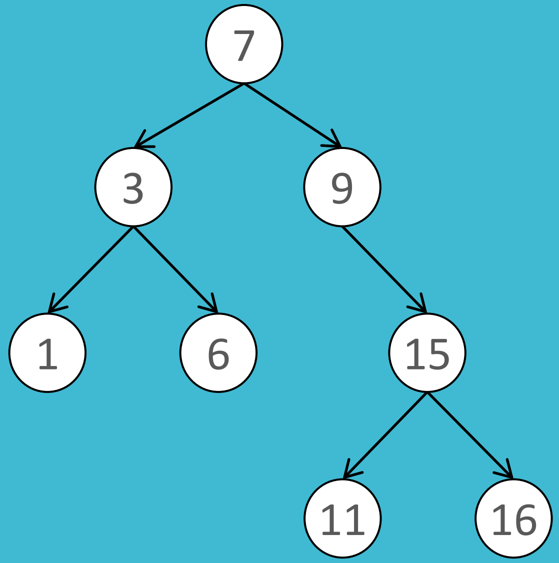
Background: Recursion



- A **binary search tree** (BST) consists of nodes, where each node:
  - has a value, v
  - up to 2 children, a left descendant and a right descendant
  - all its left descendants have values less than v and its right descendants have values greater than v
- We like BSTs because they permit search in O(log(n)) time, assuming n nodes in the tree

```
def contains_iterative(node, key):
    current_node = node
    while(true):
        if key < current_node.value & current_node.left != null
            current_node = current_node.left
        elif key > current_node.value & current_node.right != null
            current_node = current_node.left
        else
            break => return key == current_node.value
```

Background: Recursion



- A **binary search tree** (BST) consists of nodes, where each node:
  - has a value, v
  - up to 2 children, a left descendant and a right descendant
  - all its left descendants have values less than v and its right descendants have values greater than v
- We like BSTs because they permit search in O(log(n)) time, assuming n nodes in the tree

```
def contains_recursive(node, key):
    if key < node.value & node.left != null
        contains_recursive(node.left, key)
    elif key > node.value & node.right != null
        contains_recursive(node.right, key)
    else
        return key == node.value
```

Decision Tree Learning: Pseudocode

```
def train(𝒟):
    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 for x_d:
            D_v = {(x^(i), y^(i)) ∈ D′ | x_d^(i) = v}
            q.children(v) = tree_recurse(D_v)
  ★ return q
```

Decision Tree: Pseudocode

```
def train(𝒟):
    root = tree_recurse(D)

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

Decision Tree: Example – How is Henry getting to work?

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

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:

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

$$H(Y) = -\frac{3}{16}\log_2\left(\frac{3}{16}\right) - \frac{6}{16}\log_2\left(\frac{6}{16}\right) - \frac{7}{16}\log_2\left(\frac{7}{16}\right) \approx 1.5052$$

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

$I(x_1, Y) =$

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

$I(x_1, Y) \approx 1.5052$

$-\frac{6}{16}\left(-\frac{3}{6}\log_2\left(\frac{3}{6}\right) - \frac{3}{6}\log_2\left(\frac{3}{6}\right)\right)$

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

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

$I(x_1, Y) \approx 1.5052$

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

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

$I(x_1, Y) \approx 1.5052$

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

$-\frac{10}{16}\left(-\frac{3}{10}\log_2\left(\frac{3}{10}\right) - \frac{3}{10}\log_2\left(\frac{3}{10}\right) - \frac{4}{10}\log_2\left(\frac{4}{10}\right)\right)$

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

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~~ 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

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	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

$$H\left(Y_{x_4=\text{Tired}}\right) =$$

x_4

Not Tired | 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

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

$$I\left(x_1, Y_{x_4=\text{Tired}}\right) = H\left(Y_{x_4=\text{Tired}}\right) - \frac{4}{9} H\left(Y_{x_4=\text{Tired}, x_1=\text{Rain}}\right) - \frac{5}{9} H\left(Y_{x_4=\text{Tired}, x_1=\text{No Rain}}\right)$$

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

$$I\left(x_1, Y_{x_4=\text{Tired}}\right) \approx 1.2244 - \frac{4}{9}(0.8113) - \frac{5}{9}(0.9710) \approx 0.3244$$

x_4

Not Tired | 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

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

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

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

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

x_4

Not Tired

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_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_3

Backpack

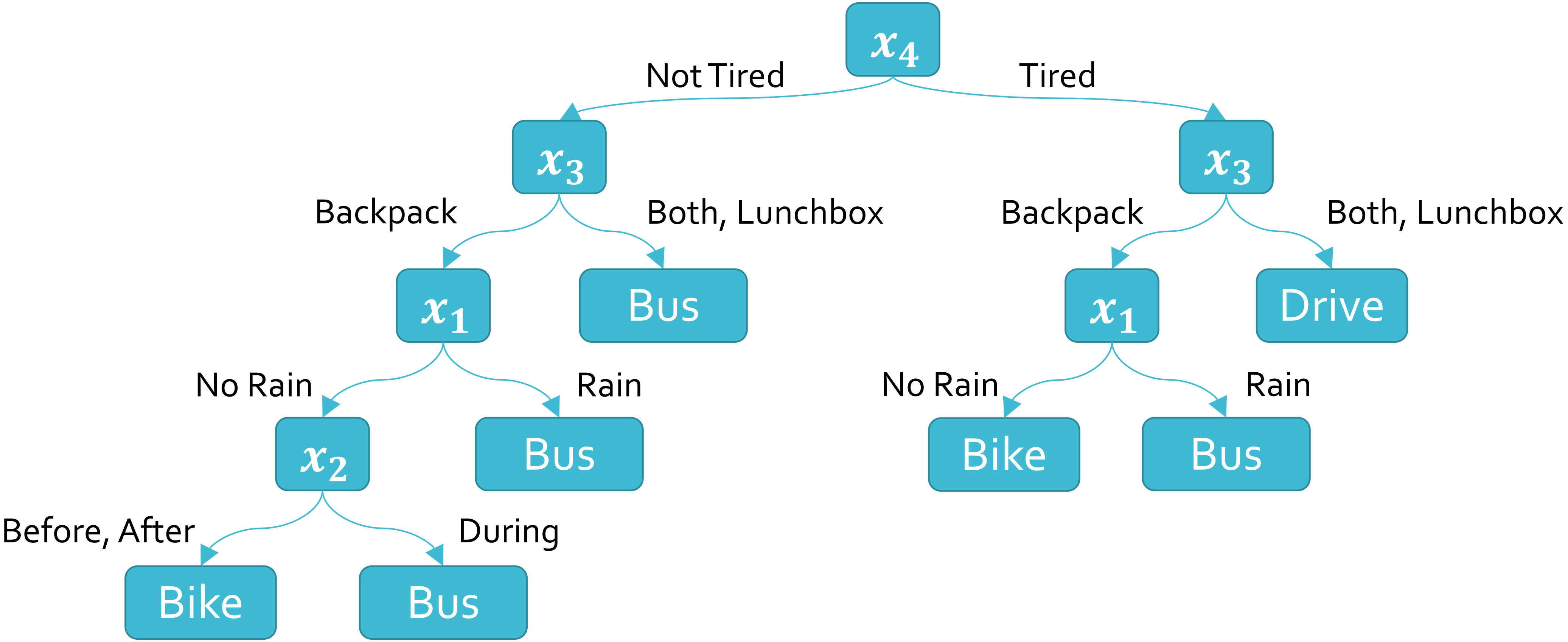
x_1	x_2	y
Rain	After	Bus
No Rain	Before	Bike
No Rain	After	Bike

Both, Lunchbox → Drive

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

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

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


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

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

Untitled survey

0 done

0 underway

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

True or False: if we use mutual information maximization as the splitting criterion, we will always learn the shortest possible decision tree with zero training error.

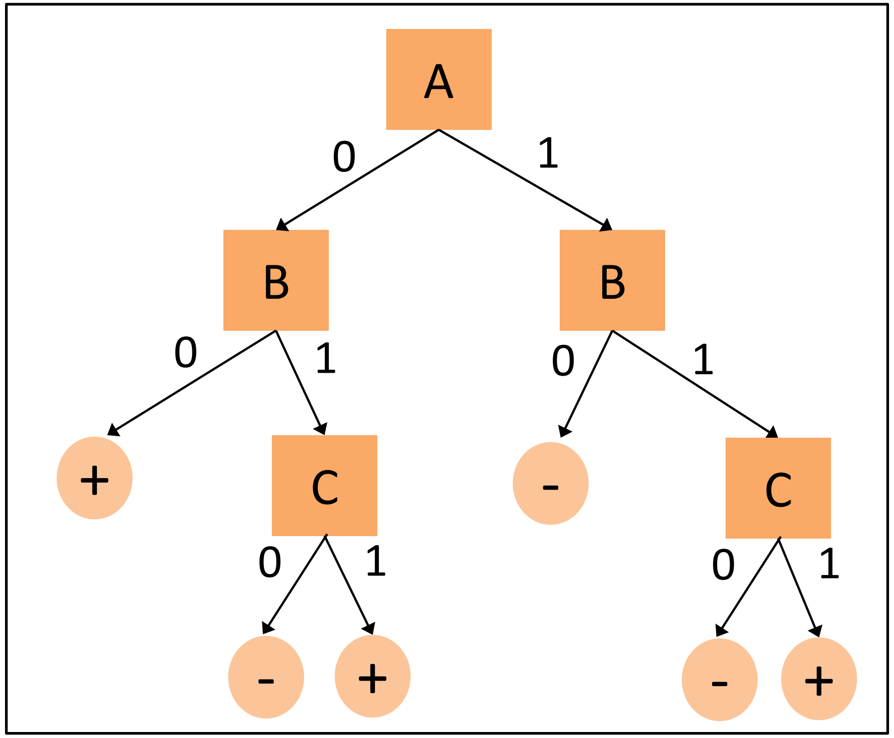
- True
- False

True or False: if we use training error minimization as the splitting criterion, we will always learn the shortest possible decision tree with zero training error.

- True
- False

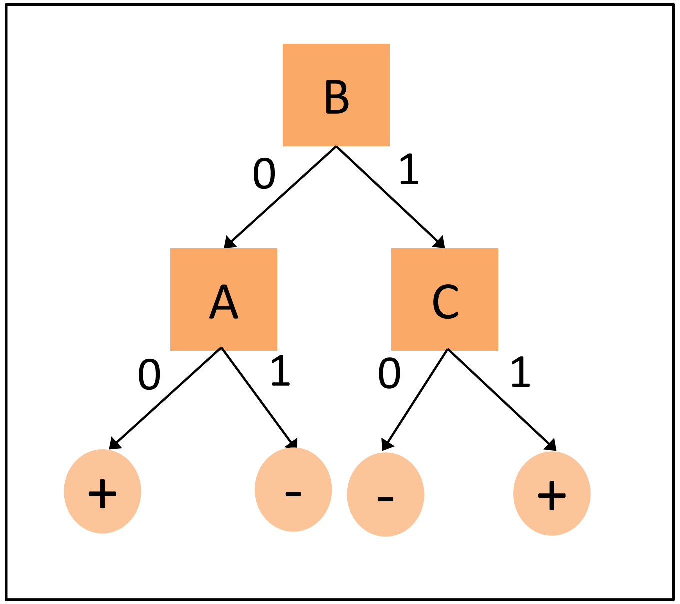
Given this dataset, if you used training error rate as the splitting criterion, you would learn this tree...

A	B	C	y
0	0	0	+
0	0	1	+
0	1	0	−
0	1	1	+
1	0	0	−
1	0	1	−
1	1	0	−
1	1	1	+



... but there actually exists a shorter decision tree with zero training error!

A	B	C	y
0	0	0	$+$
0	0	1	$+$
0	1	0	$-$
0	1	1	$+$
1	0	0	$-$
1	0	1	$-$
1	1	0	$-$
1	1	1	$+$



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 shortest tree that achieves zero (low) training error with high splitting criterion (mutual information) features at the top
- Occam's razor: try to find the "simplest" (e.g., smallest decision tree) classifier that explains the training dataset

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

Real-Valued Features: Example - x = Outside Temperature (°F)

x	y
74	Drive
55	Metro
63	Bike
33	Drive
80	Drive
81	Drive
44	Metro
45	Metro
78	Drive
51	Metro

→

x	y
33	Drive
44	Metro
45	Metro
51	Metro
55	Metro
63	Bike
74	Drive
78	Drive
80	Drive
81	Drive

← $x < 38.5$ (between rows 33 and 44)

Real-Valued Features: Example - x = Outside Temperature (°F)

x	y
74	Drive
55	Metro
63	Bike
33	Drive
80	Drive
81	Drive
44	Metro
45	Metro
78	Drive
51	Metro

→

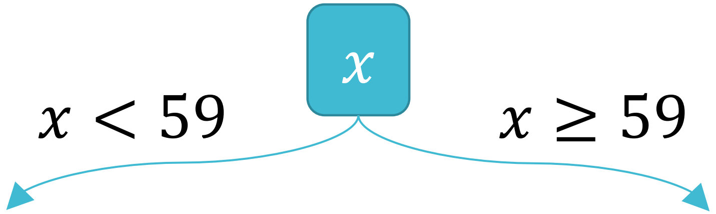
x	y
33	Drive
44	Metro
45	Metro
51	Metro
55	Metro
63	Bike
74	Drive
78	Drive
80	Drive
81	Drive

← $x < 44.5$ (between rows 44 and 45)

Real-Valued Features: Example - x = Outside Temperature (°F)

x	y
74	Drive
55	Metro
63	Bike
33	Drive
80	Drive
81	Drive
44	Metro
45	Metro
78	Drive
51	Metro

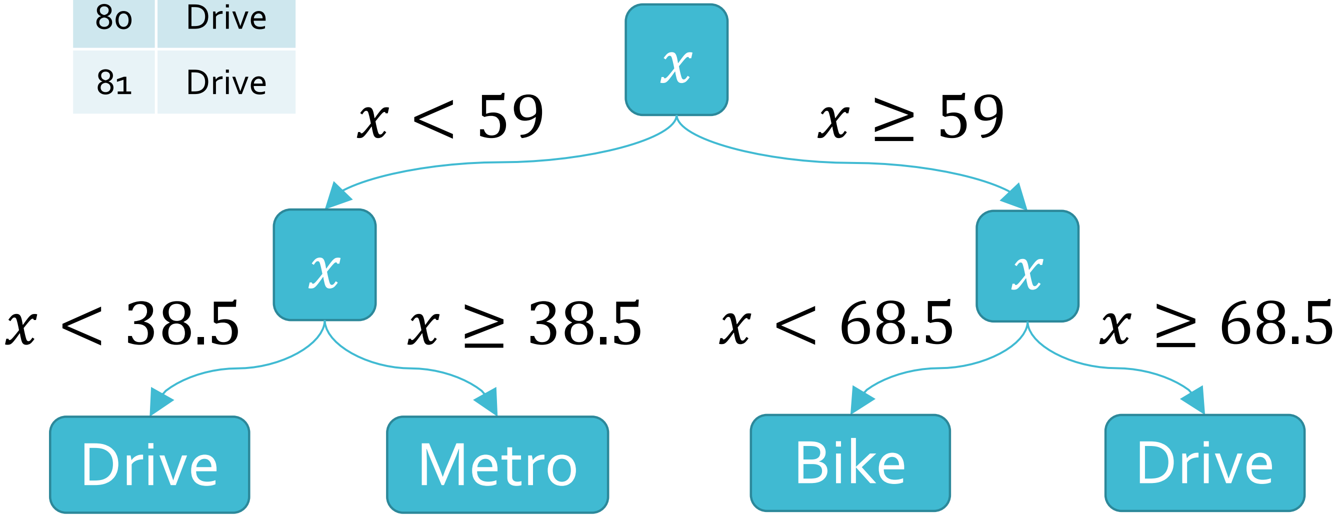
x	y
33	Drive
44	Metro
45	Metro
51	Metro
55	Metro
63	Bike
74	Drive
78	Drive
80	Drive
81	Drive



Real-Valued Features: Example - x = Outside Temperature (°F)

x	y
74	Drive
55	Metro
63	Bike
33	Drive
80	Drive
81	Drive
44	Metro
45	Metro
78	Drive
51	Metro

x	y
33	Drive
44	Metro
45	Metro
51	Metro
55	Metro
63	Bike
74	Drive
78	Drive
80	Drive
81	Drive



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!

Overfitting

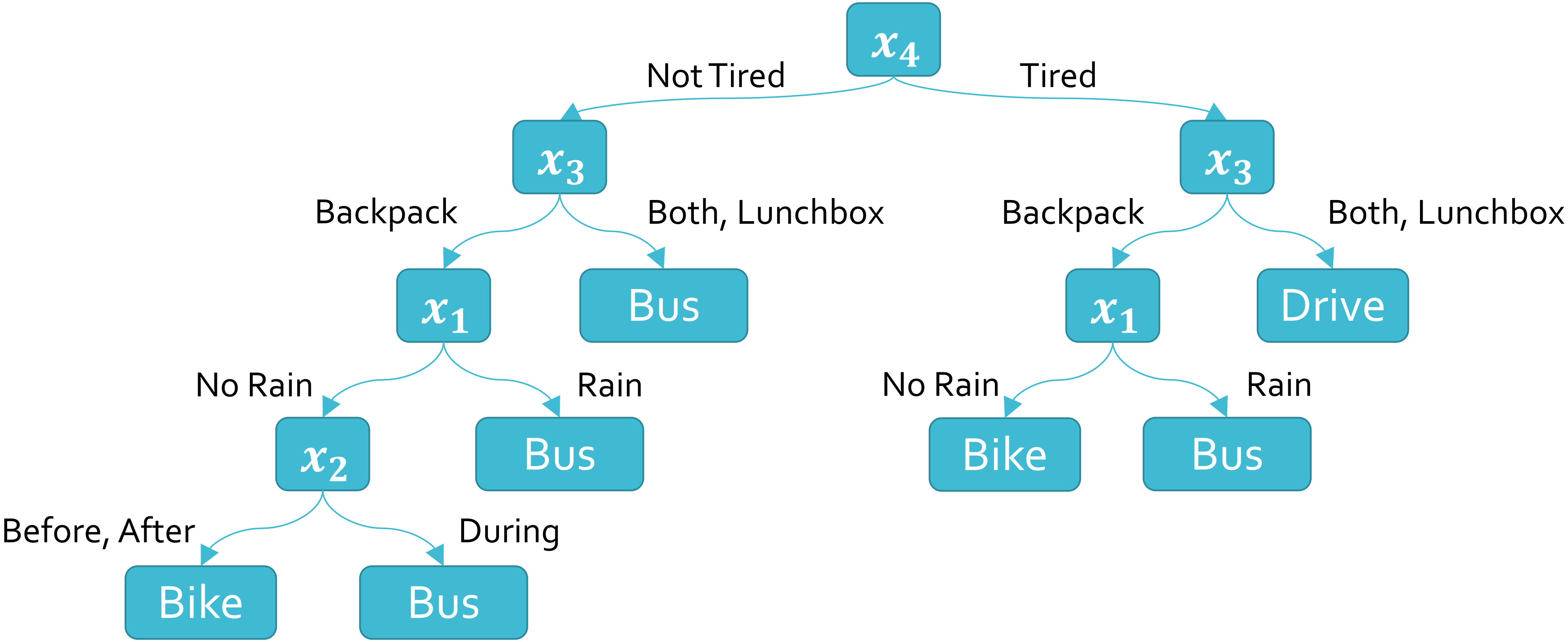
- Overfitting occurs when the classifier (or model)…
  - is too complex
  - fits noise or “outliers” in the training dataset as opposed to the actual pattern of interest
  - doesn’t have enough inductive bias pushing it to generalize
- Underfitting occurs when the classifier (or model)…
  - is too simple
  - can’t capture the actual pattern of interest in the training dataset
  - has too much inductive bias

Different Kinds of Error

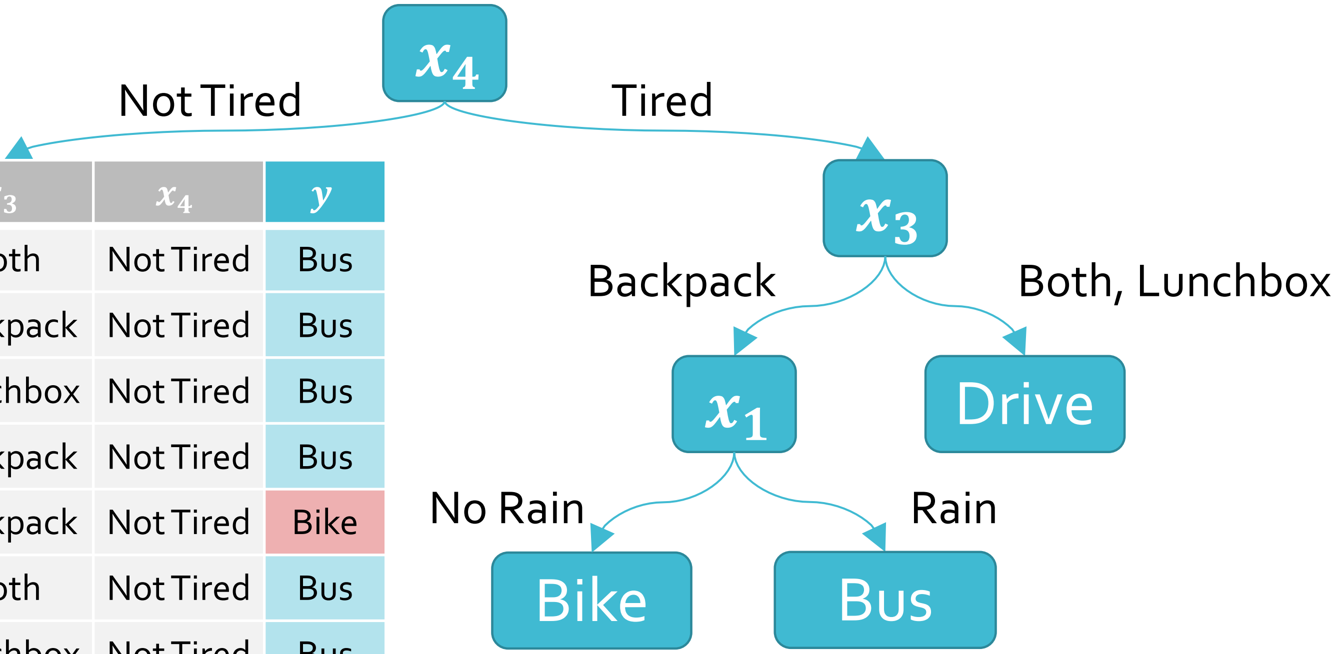
- Training error rate = $err(h, \mathcal{D}_{train})$
- Test error rate = $err(h, \mathcal{D}_{test})$
- True error rate = $err(h)$

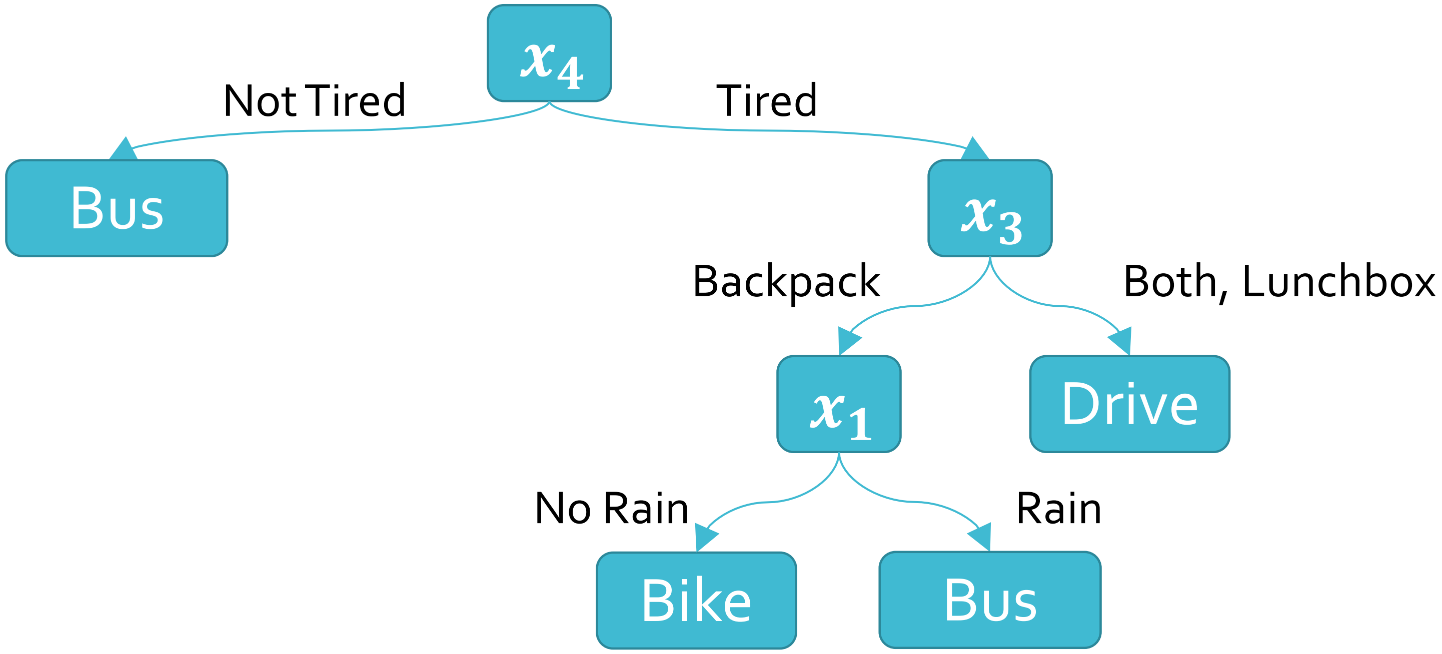
  = the error rate of h on all possible examples

  - In machine learning, this is the quantity that we care about but, in most cases, it is unknowable.
- Overfitting occurs when $err(h) \gg err(h, \mathcal{D}_{train})$
  - $err(h) - err(h, \mathcal{D}_{train})$ can be thought of as a measure of overfitting


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

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





This tree only misclassifies one training data point!

Overfitting in Decision Trees

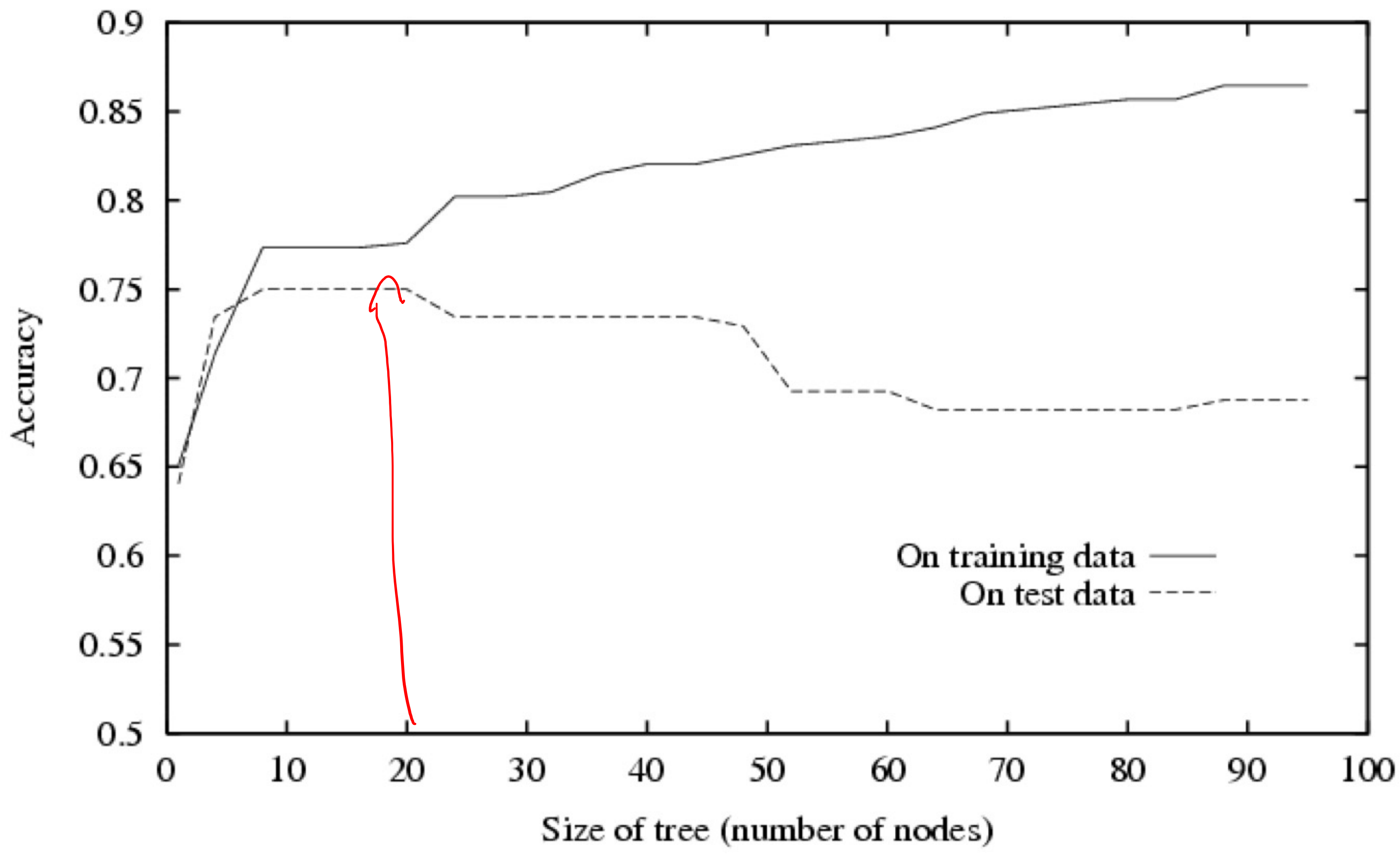


Figure courtesy of Tom Mitchell

Combatting Overfitting in Decision Trees

- Heuristics:
  - Do not split leaves past a fixed depth, δ
  - Do not split leaves with fewer than c data points
  - Do not split leaves where the maximal information gain is less than τ
- Take a majority vote in impure leaves

Combatting Overfitting in Decision Trees

- Pruning:
  1. First, learn a decision tree
  2. Then, evaluate each split using a "validation" dataset by comparing the validation error rate with and without that split
  3. Greedily remove the split that most decreases the validation error rate
     - Break ties in favor of smaller trees
  4. Stop if no split is removed

Pruning Decision Trees

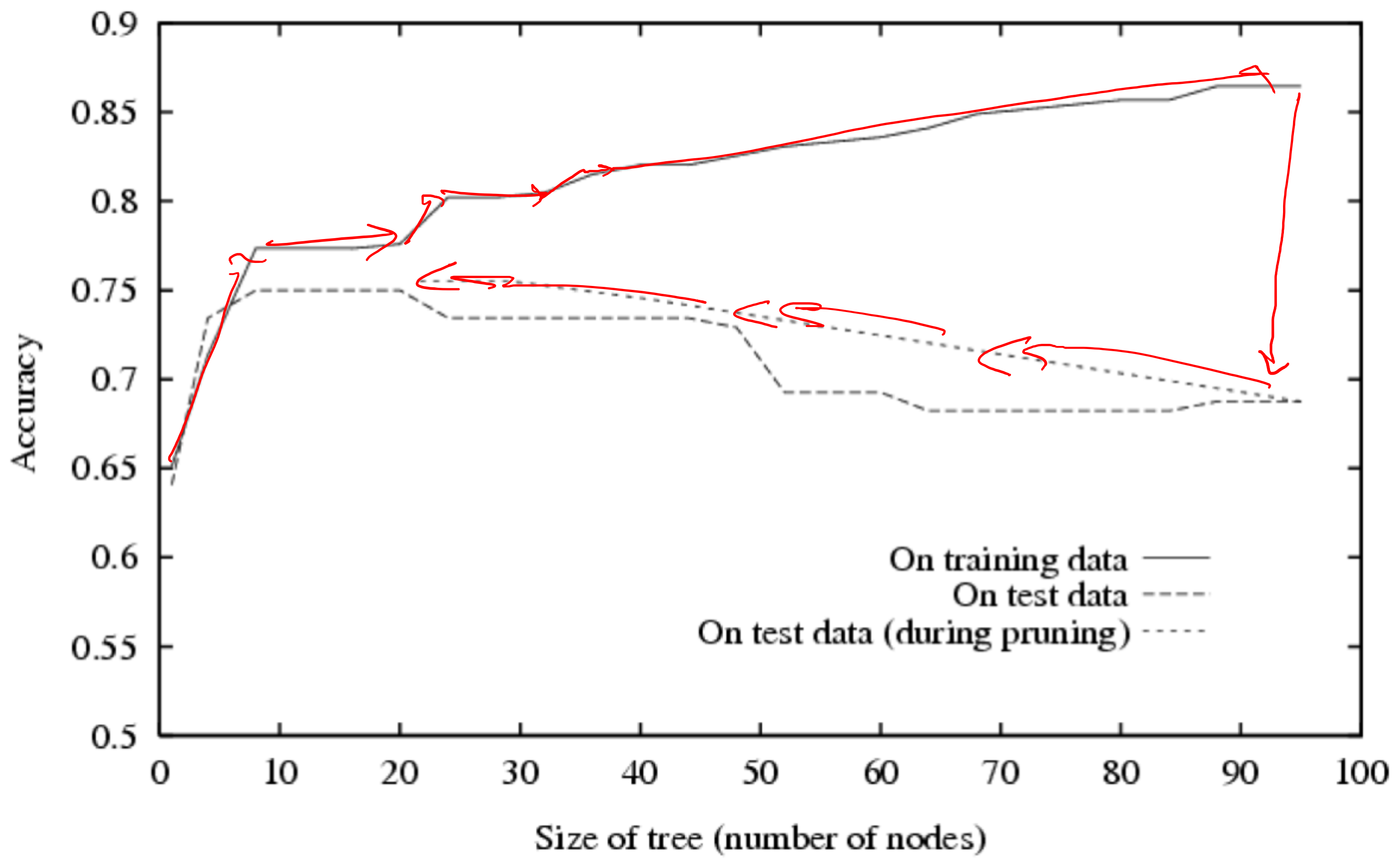
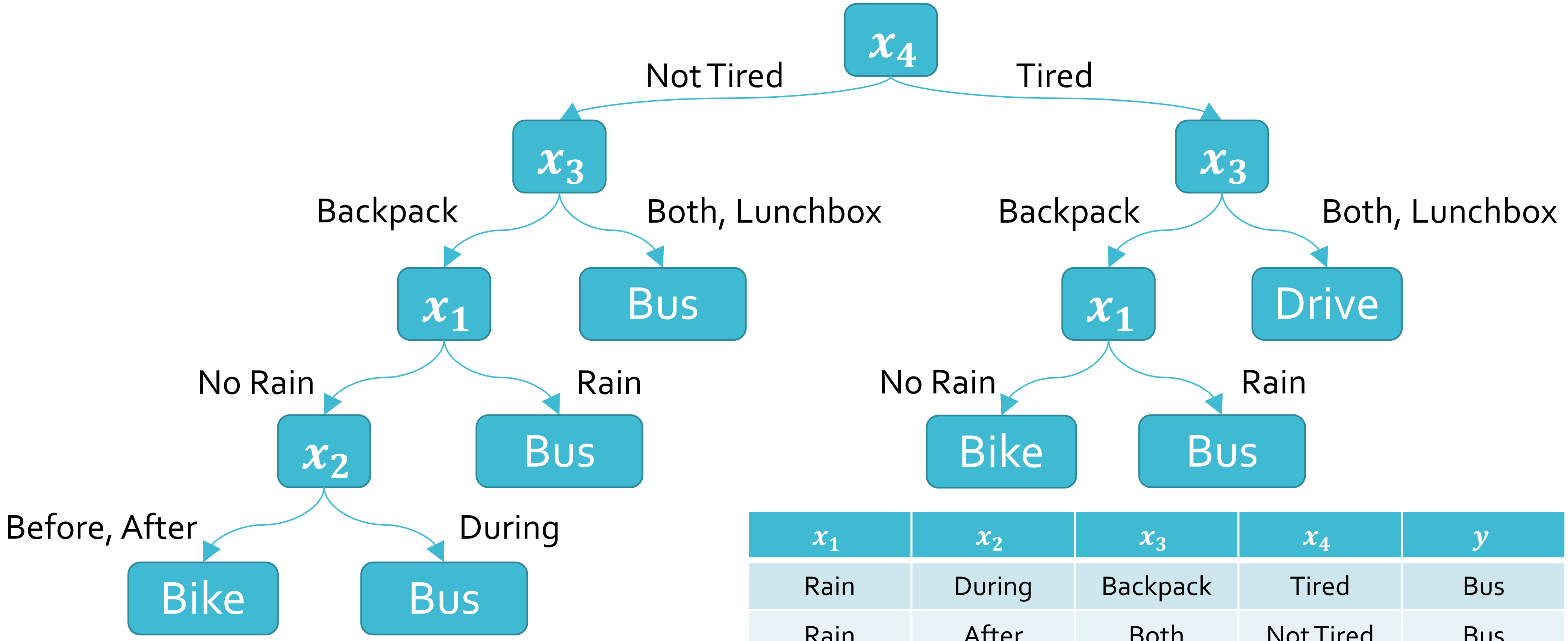
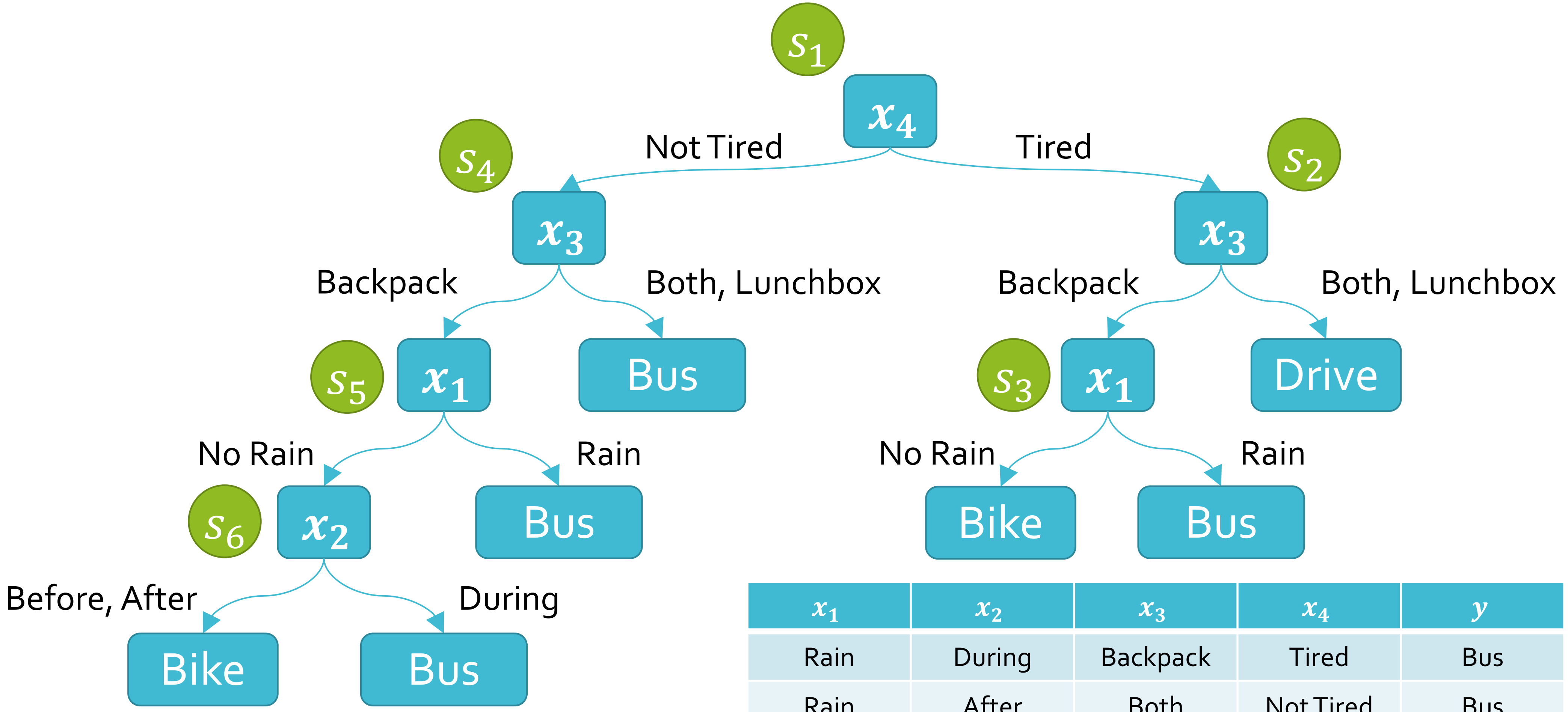


Figure courtesy of Tom Mitchell



$\mathcal{D}_{val} =$

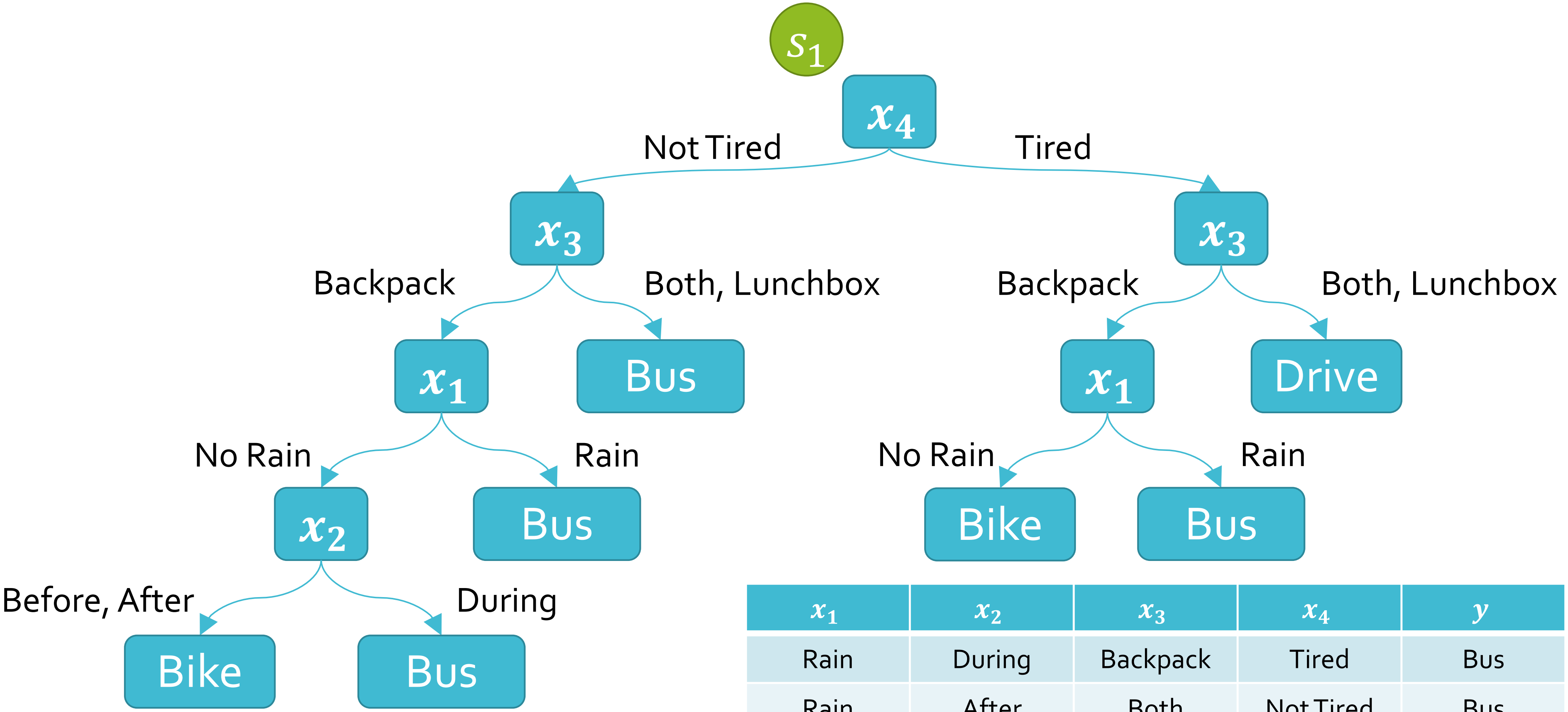
x_1	x_2	x_3	x_4	y
Rain	During	Backpack	Tired	Bus
Rain	After	Both	Not Tired	Bus
No Rain	Before	Backpack	Not Tired	Bus
No Rain	During	Lunchbox	Tired	Drive
No Rain	After	Lunchbox	Tired	Drive



$$\mathcal{D}_{val} =$$

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

$$err(h, \mathcal{D}_{val}) = 0.2$$



$\mathcal{D}_{val} =$

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

$err(h - s_1, \mathcal{D}_{val})$

$\mathcal{D}_{val} =$

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

$err(h - s_1, \mathcal{D}_{val})$

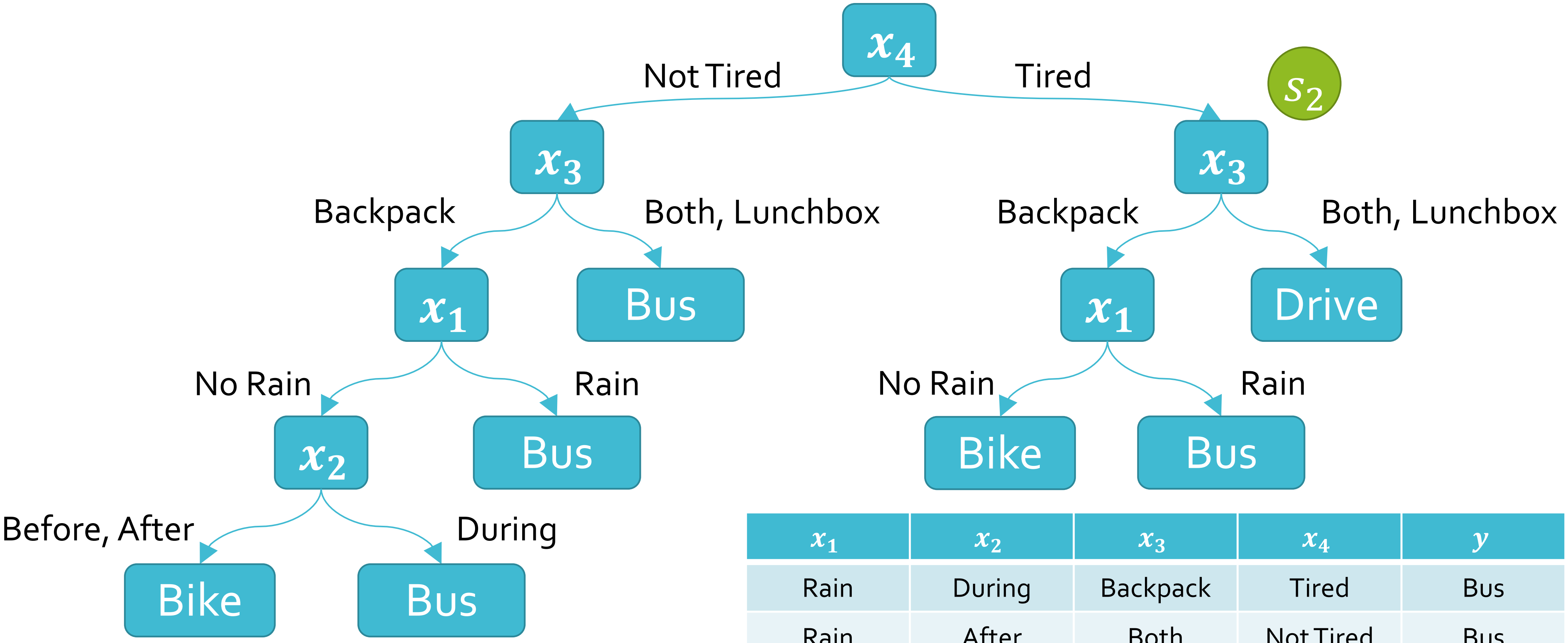
s_1

Bus

$$err(h - s_1, \mathcal{D}_{val}) = 0.4$$

$\mathcal{D}_{val} =$

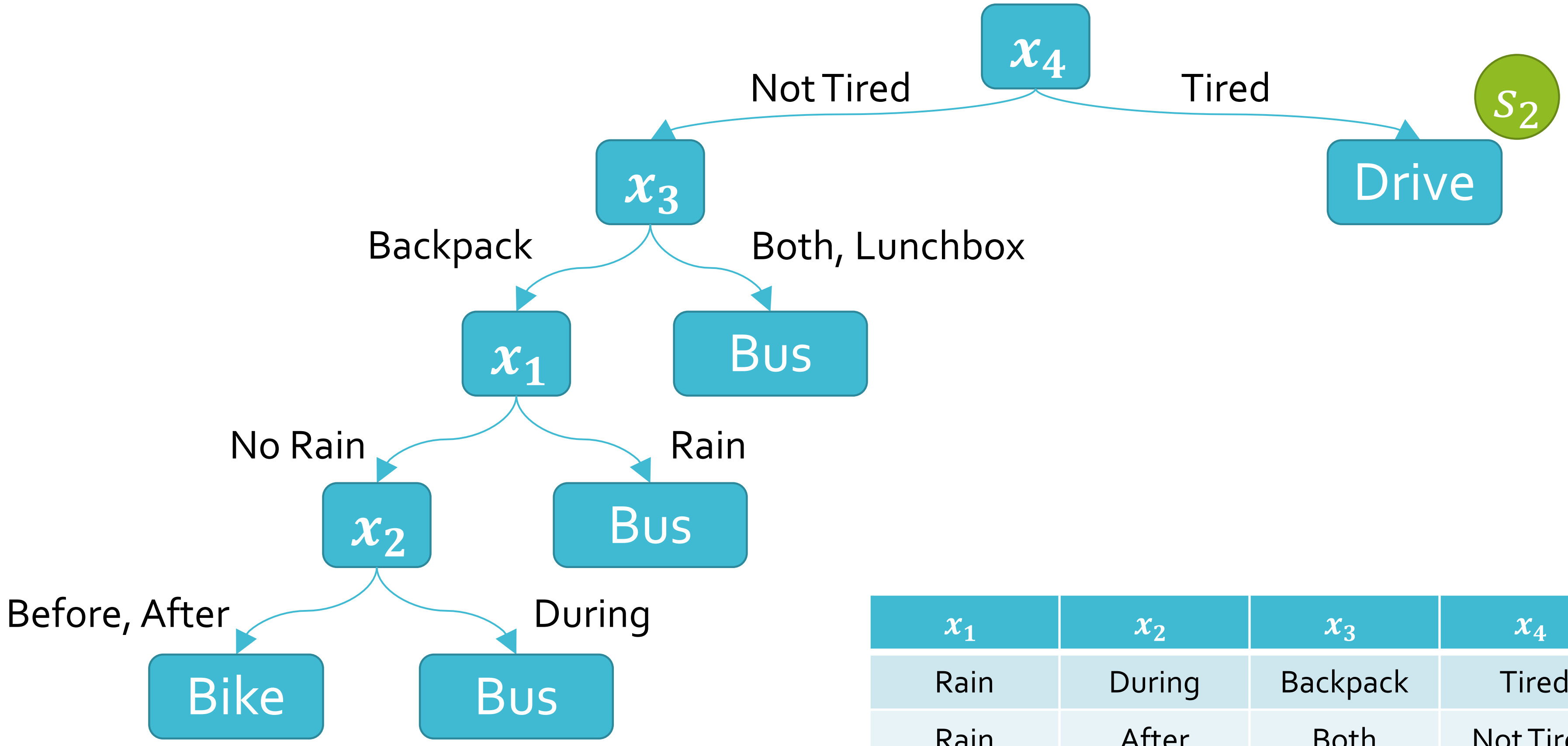
x_1	x_2	x_3	x_4	y
Rain	During	Backpack	Tired	Bus
Rain	After	Both	Not Tired	Bus
No Rain	Before	Backpack	Not Tired	Bus
No Rain	During	Lunchbox	Tired	Drive
No Rain	After	Lunchbox	Tired	Drive



$\mathcal{D}_{val} =$

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

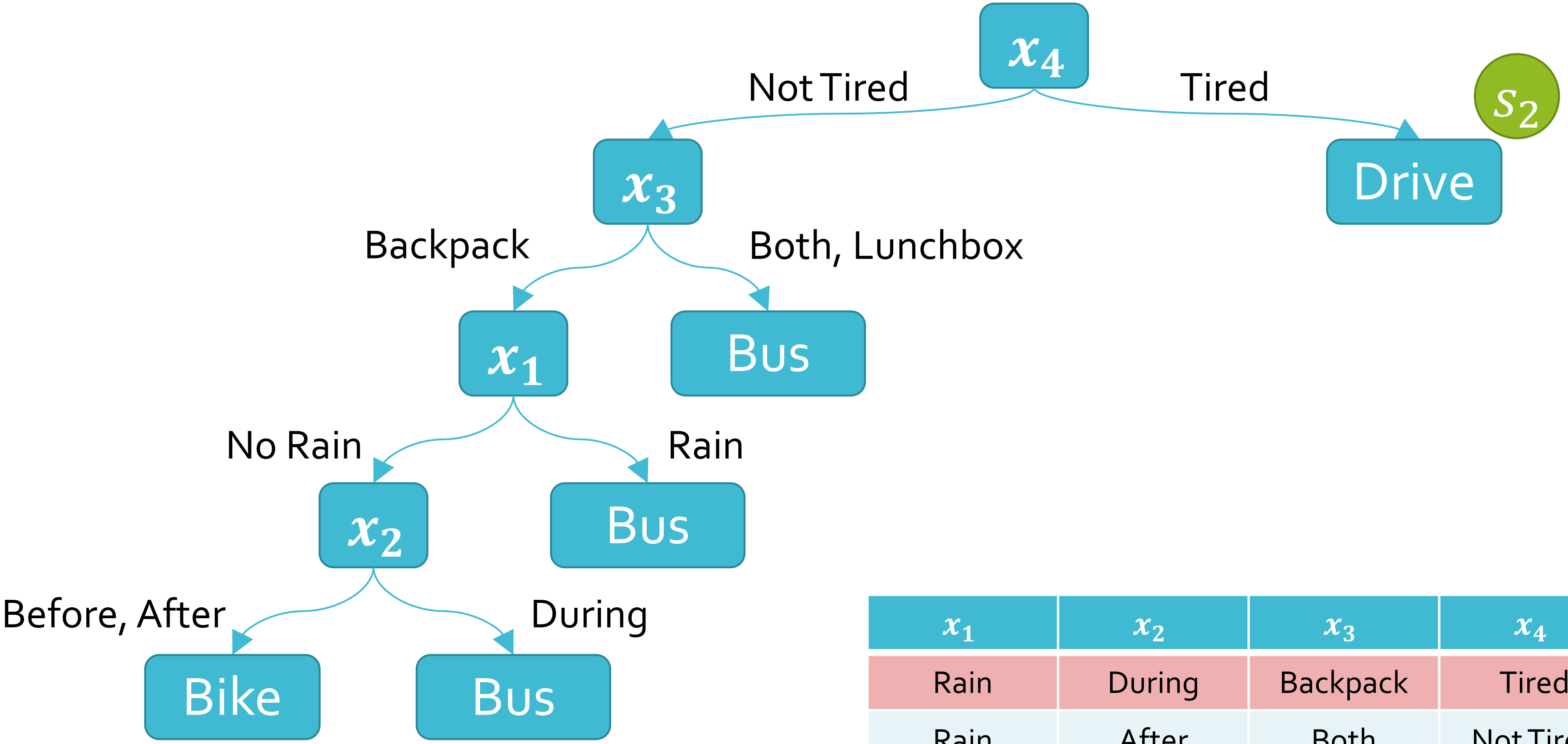
$err(h - s_2, \mathcal{D}_{val})$



$\mathcal{D}_{val} =$

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

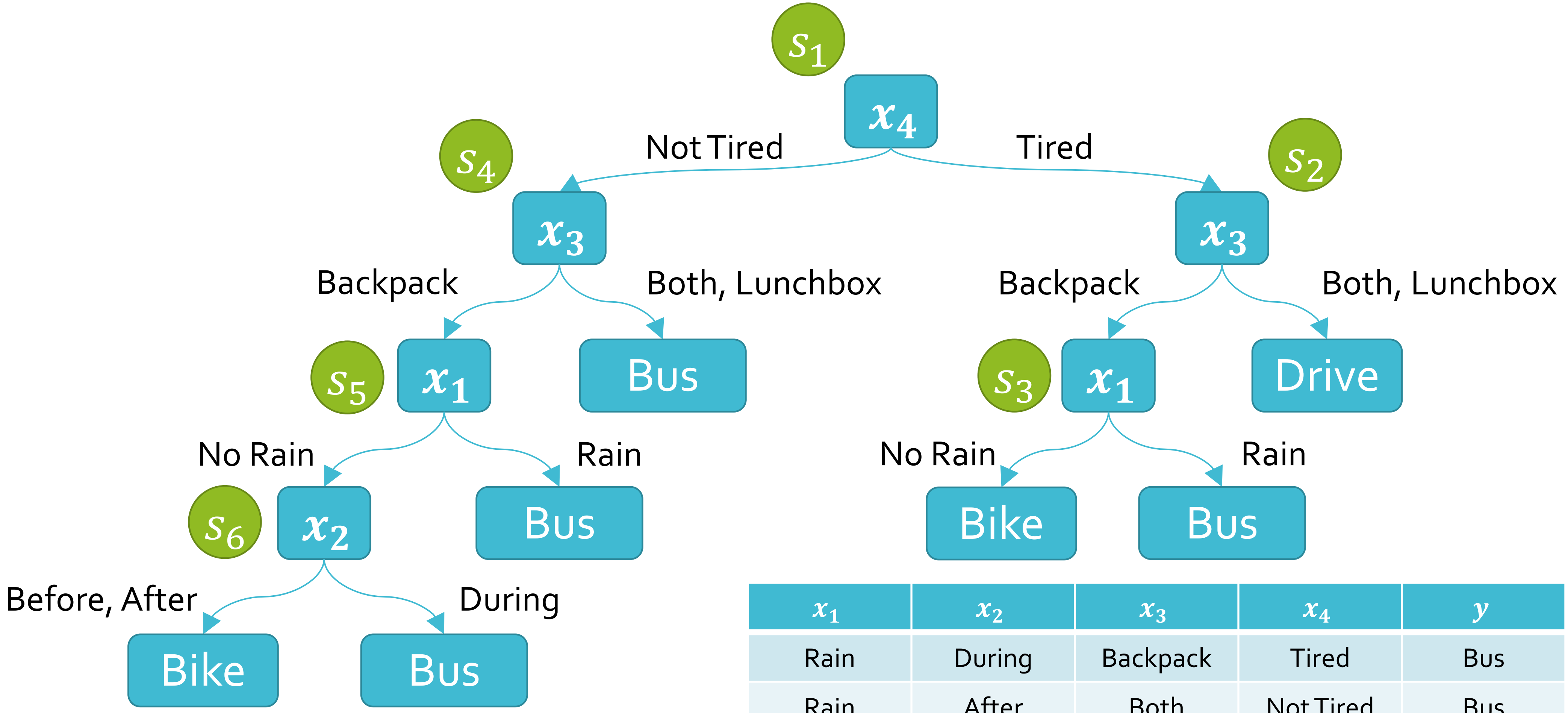
$err(h - s_2, \mathcal{D}_{val})$



$\mathcal{D}_{val} =$

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

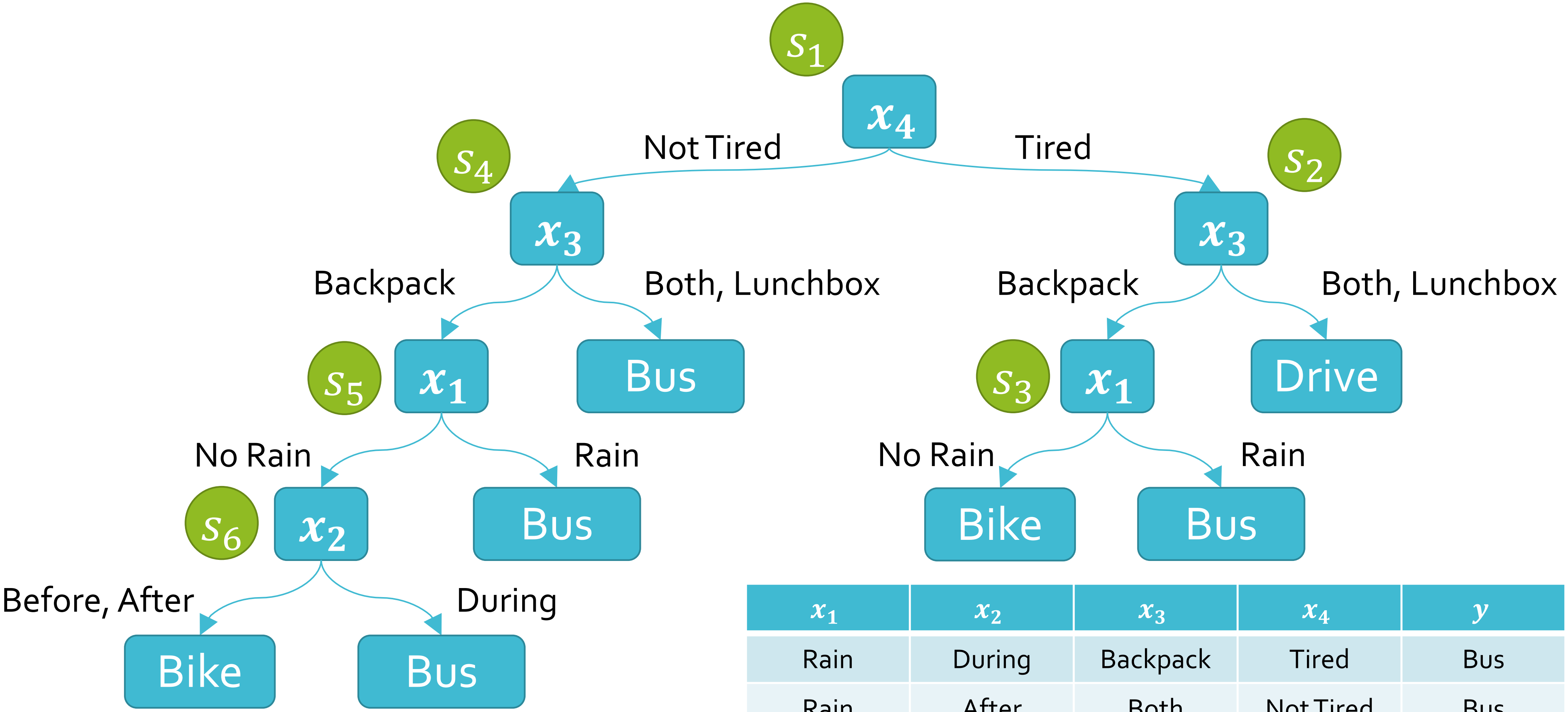
$err(h - s_2, \mathcal{D}_{val}) = 0.4$



$\mathcal{D}_{val} =$

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

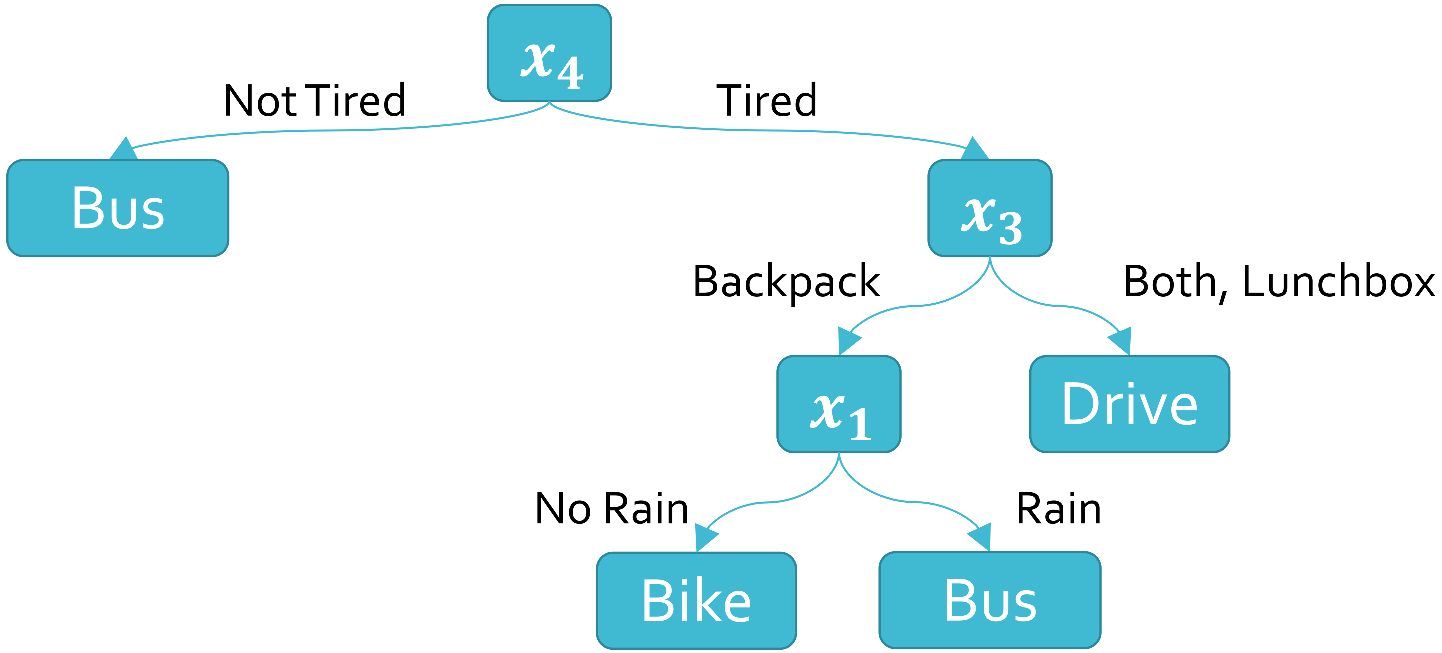
s	s_1	s_2	s_3	s_4	s_5	s_6
$err(h - s, \mathcal{D}_{val})$	0.4	0.4	0.4	0	0	0.2



$\mathcal{D}_{val} =$

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

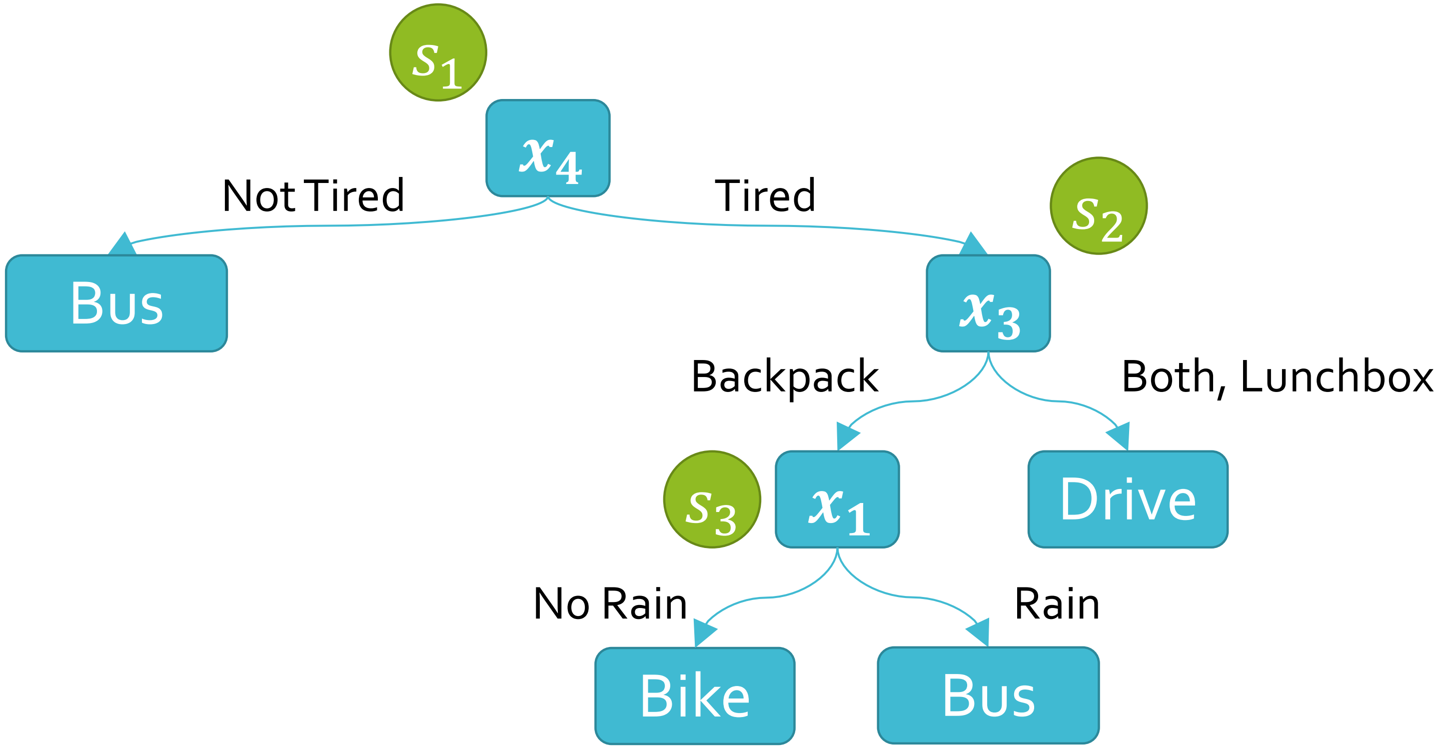
s	s_1	s_2	s_3	s_4	s_5	s_6
$err(h - s, \mathcal{D}_{val})$	0.4	0.4	0.4	0	0	0.2



$\mathcal{D}_{val} =$

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

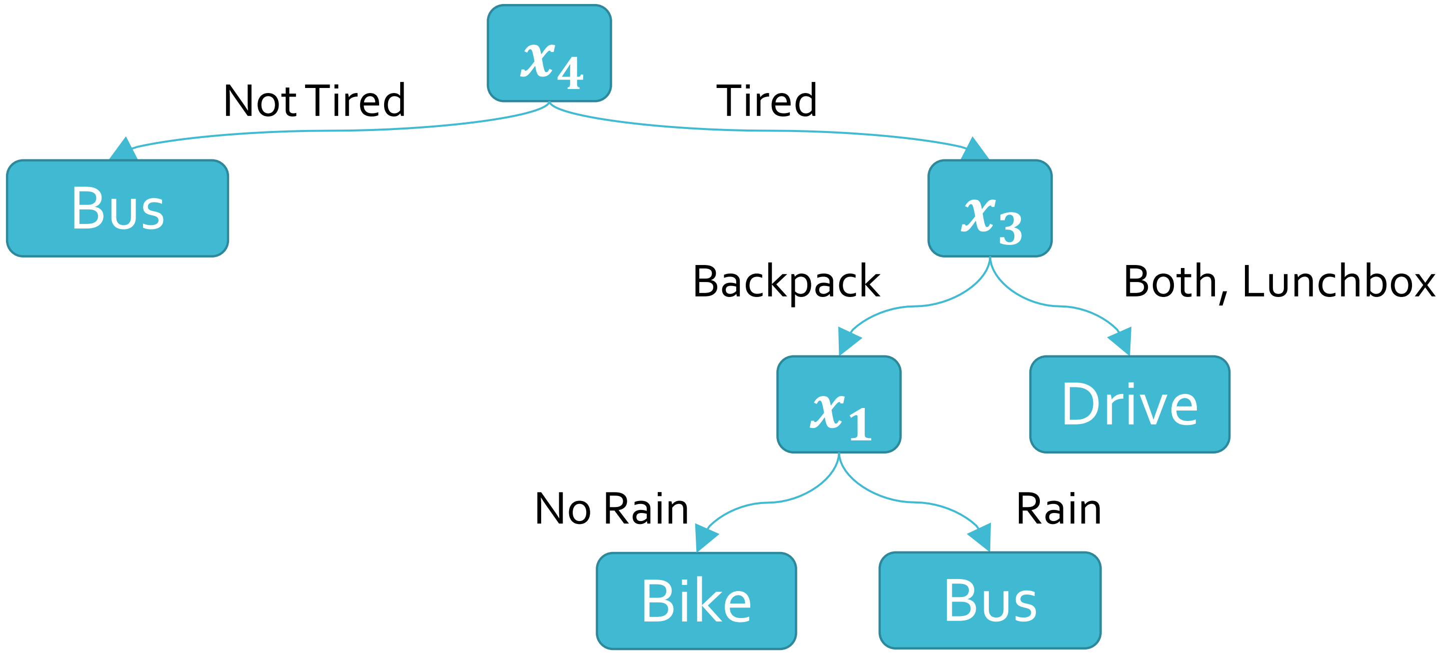
$$err(h, \mathcal{D}_{val}) = 0$$



$\mathcal{D}_{val} =$

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

s	s_1	s_2	s_3
$err(h - s, \mathcal{D}_{val})$	0.4	0.2	0.2



x_4
Not Tired
Tired
Bus
x_3
Backpack
Both, Lunchbox
x_1
Drive
No Rain
Rain
Bike
Bus

Key Takeaways

- Decision tree prediction algorithm
- Decision tree learning algorithm via recursion
- Inductive bias of decision trees
- Overfitting vs. Underfitting
- How to combat overfitting in decision trees