10-301/601: Introduction to Machine Learning Lecture 4 – Overfitting

Henry Chai

5/13/25

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

Unsure

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 training error minimization as the splitting criterion, we will always learn the shortest possible decision tree with zero training error.

True

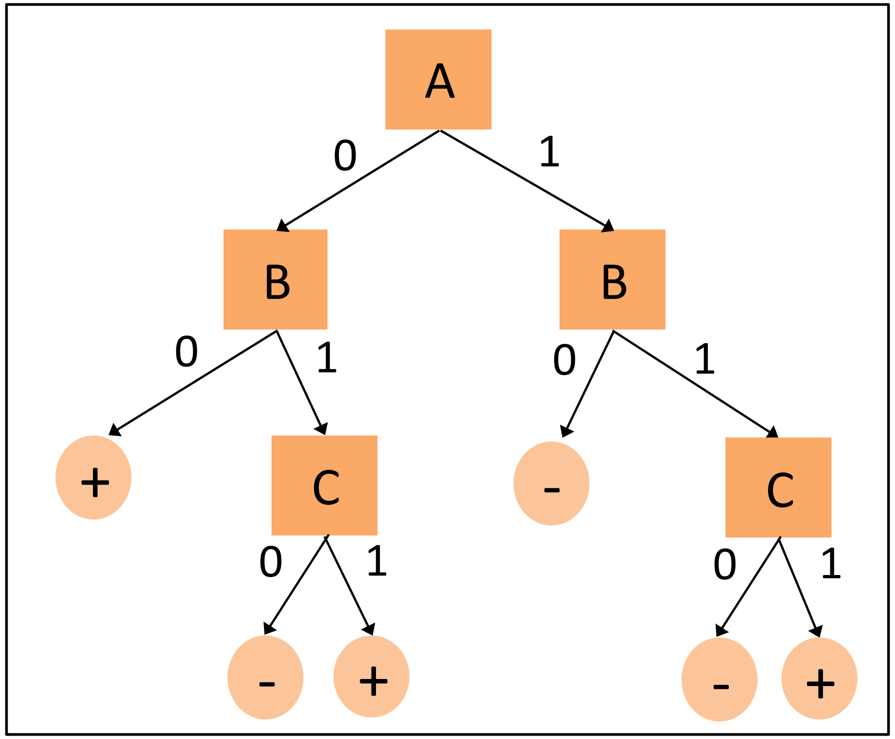
False

Unsure

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

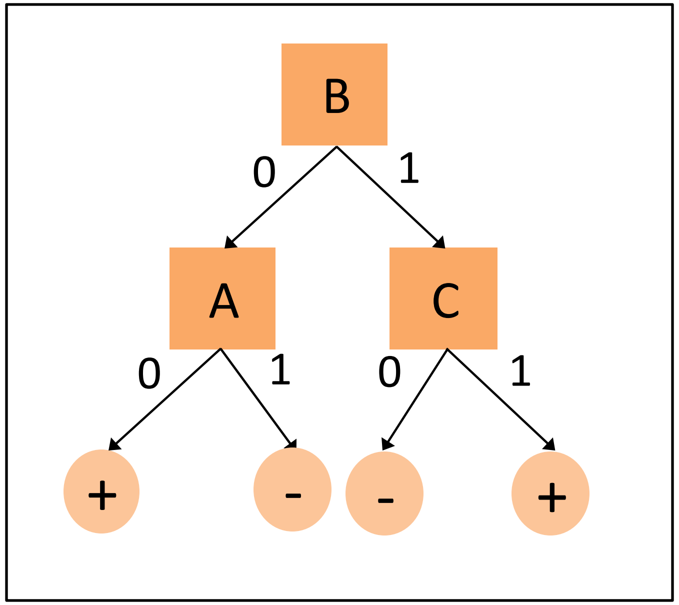
Given this dataset, if you used training error rate as the splitting criterion, you would learn this tree...

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 lowest possible training error with most informative 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

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

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

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

$x < 38.5$

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

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

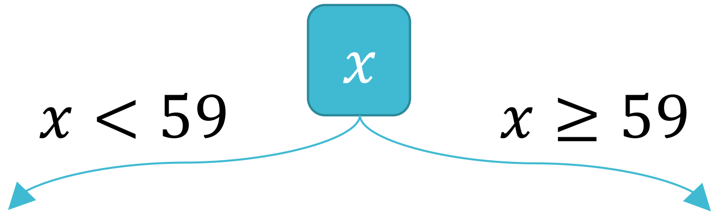
x	y
33	Drive
44	Bus
45	Bus
51	Bus
55	Bus
63	Bike
74	Drive
78	Drive
80	Drive
81	Drive

$x < 44.5$

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

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

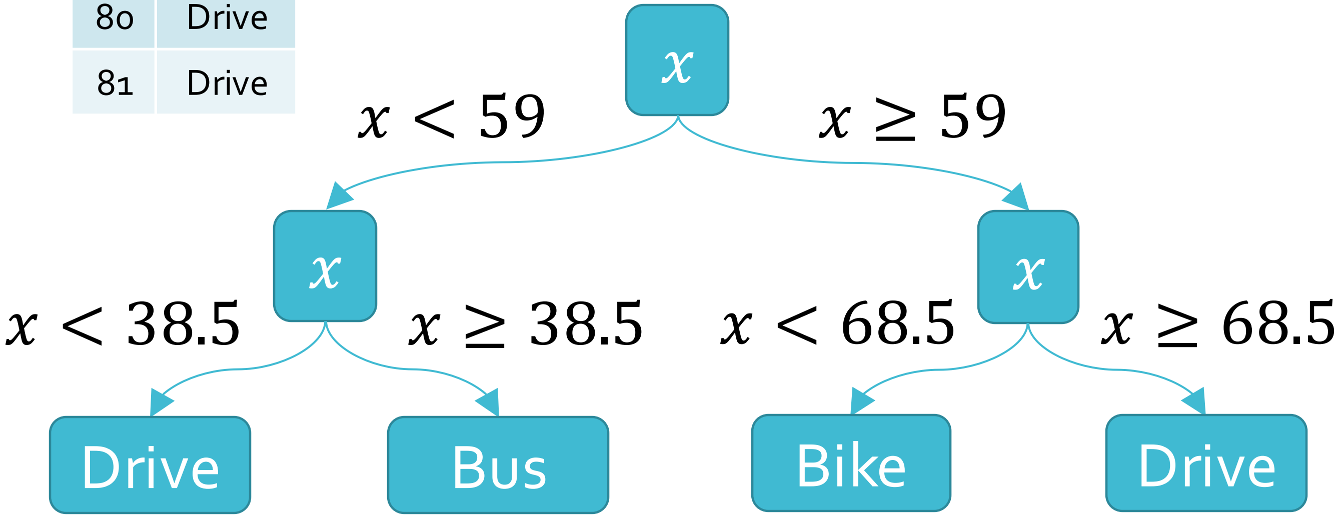
x	y
33	Drive
44	Bus
45	Bus
51	Bus
55	Bus
63	Bike
74	Drive
78	Drive
80	Drive
81	Drive



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

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

x	y
33	Drive
44	Bus
45	Bus
51	Bus
55	Bus
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

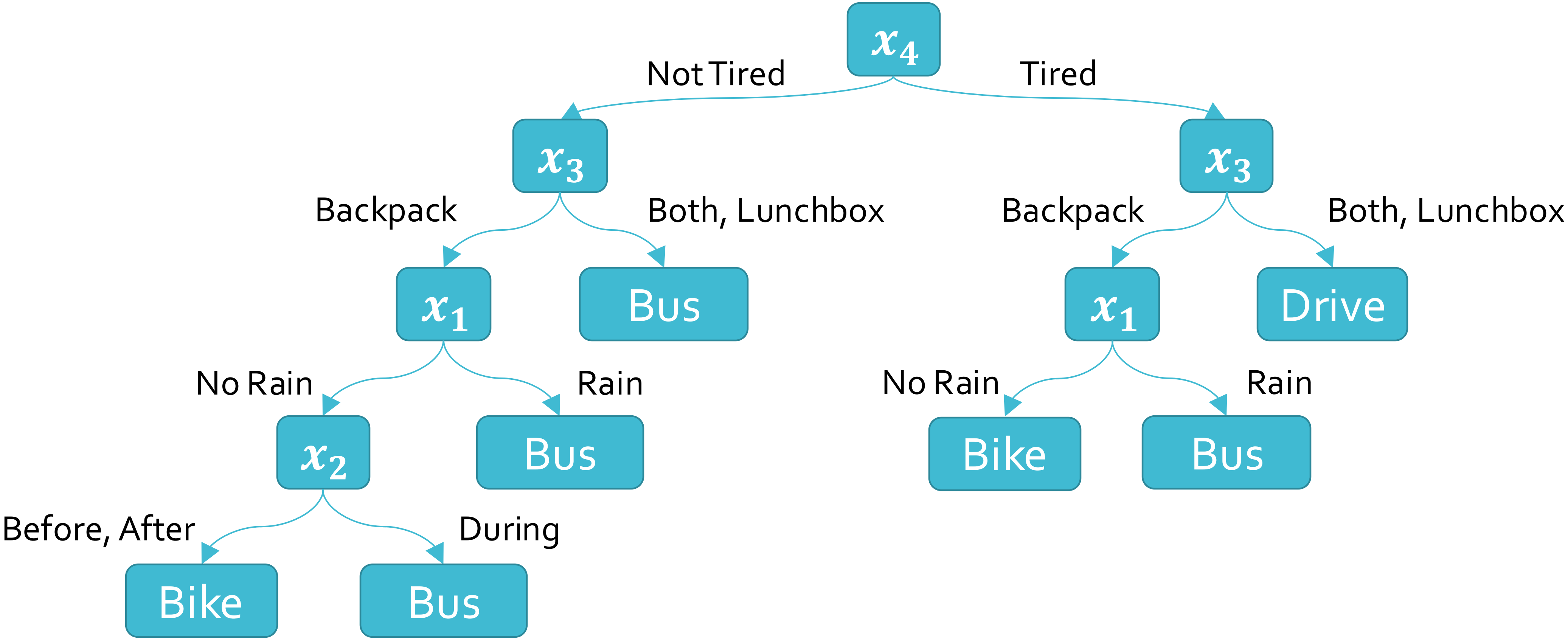
- Overfitting occurs when the classifier (or model)…
  - is too complex
  - fits noise or “outliers” in the training dataset as opposed to the actual pattern of interest
  - doesn’t have enough inductive bias pushing it to generalize
- Underfitting occurs when the classifier (or model)…
  - is too simple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

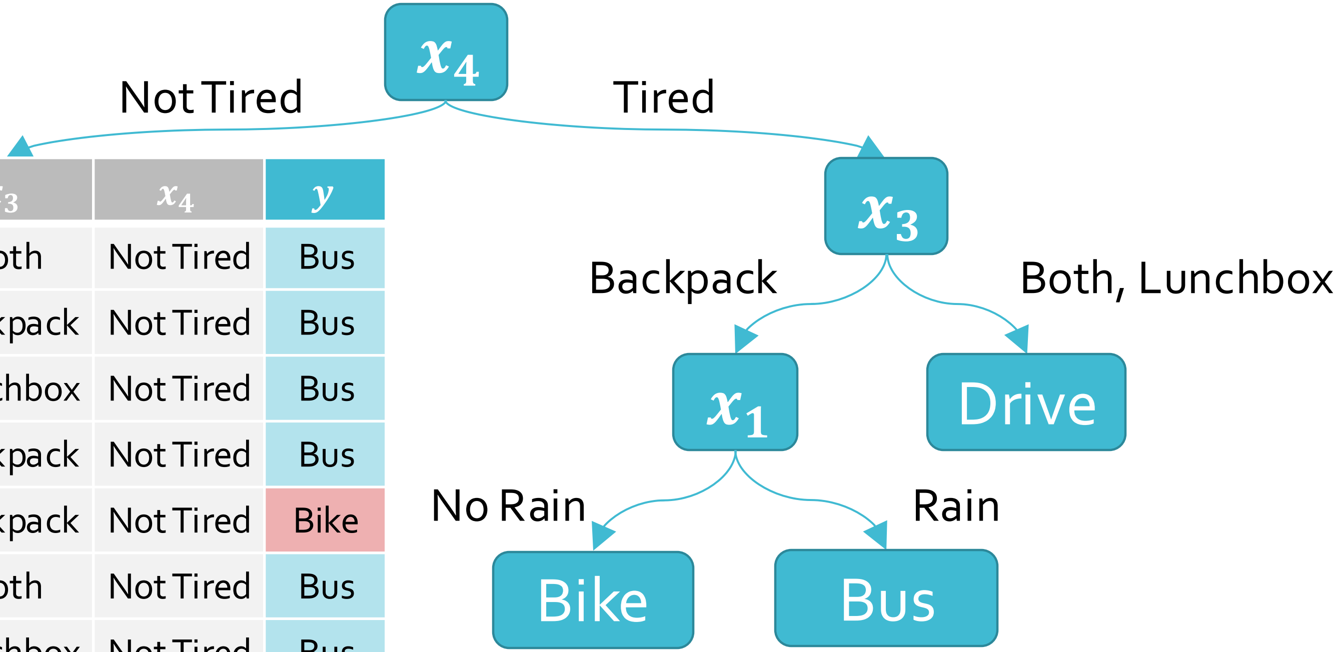
Different Kinds of Error

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

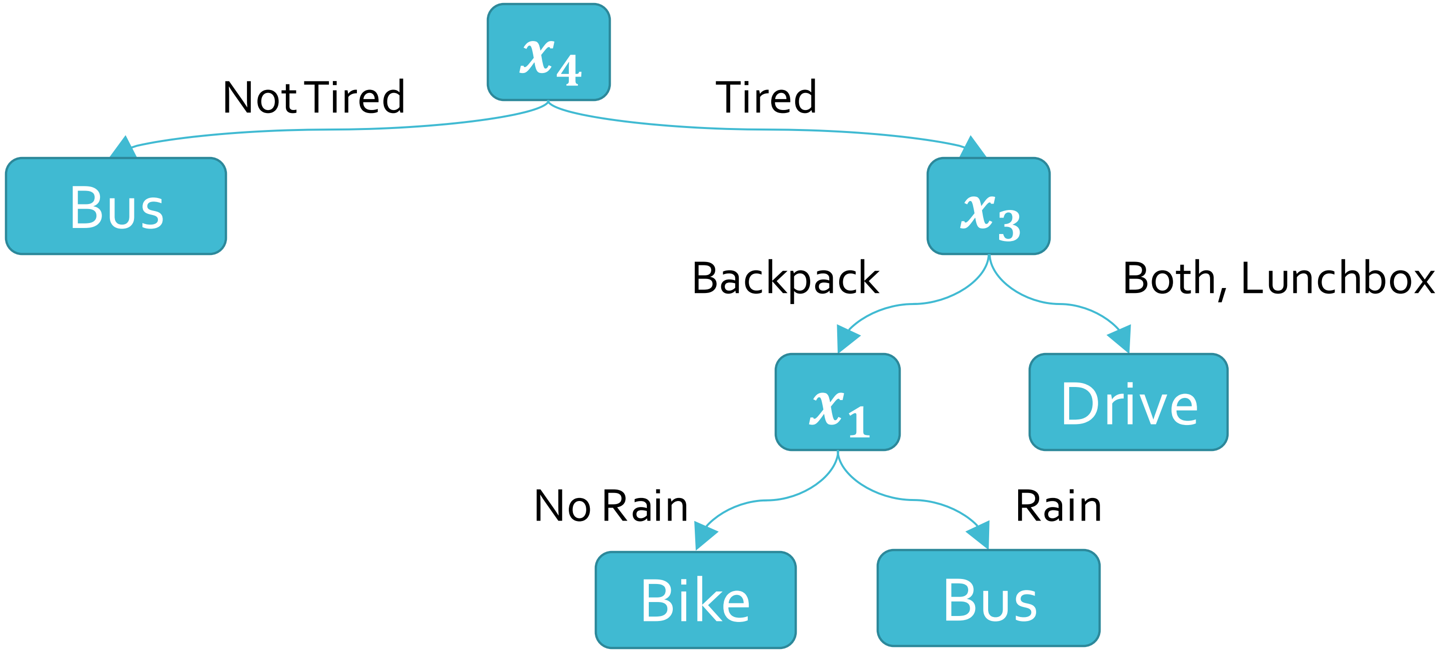
  = the error rate of h on all possible examples

  - In machine learning, this is the quantity that we care about but, in most cases, it is unknowable.
- Overfitting occurs when $err(h) > err(h, \mathcal{D}_{train})$
  - $err(h) - err(h, \mathcal{D}_{train})$ 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

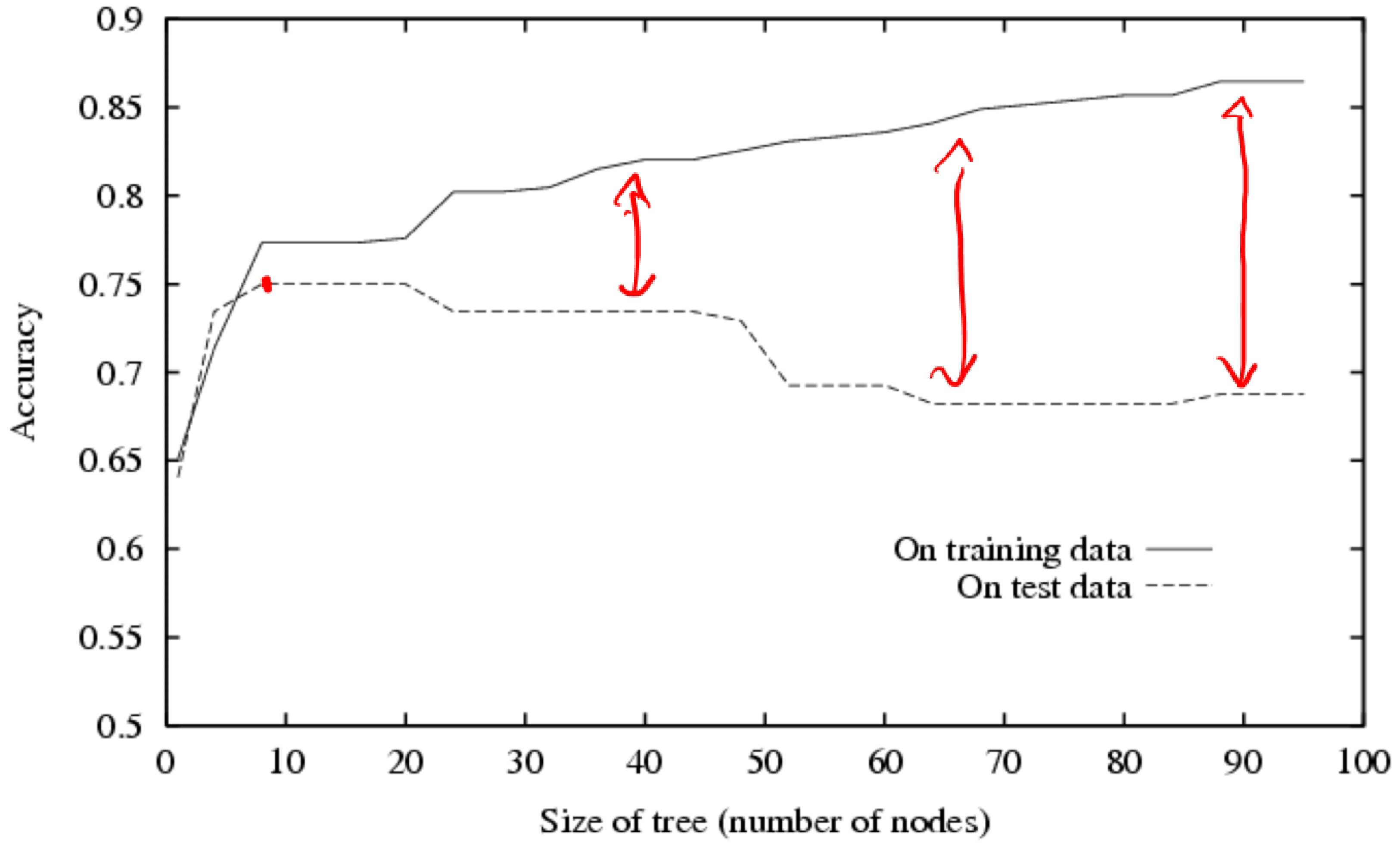


Figure courtesy of Tom Mitchell

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

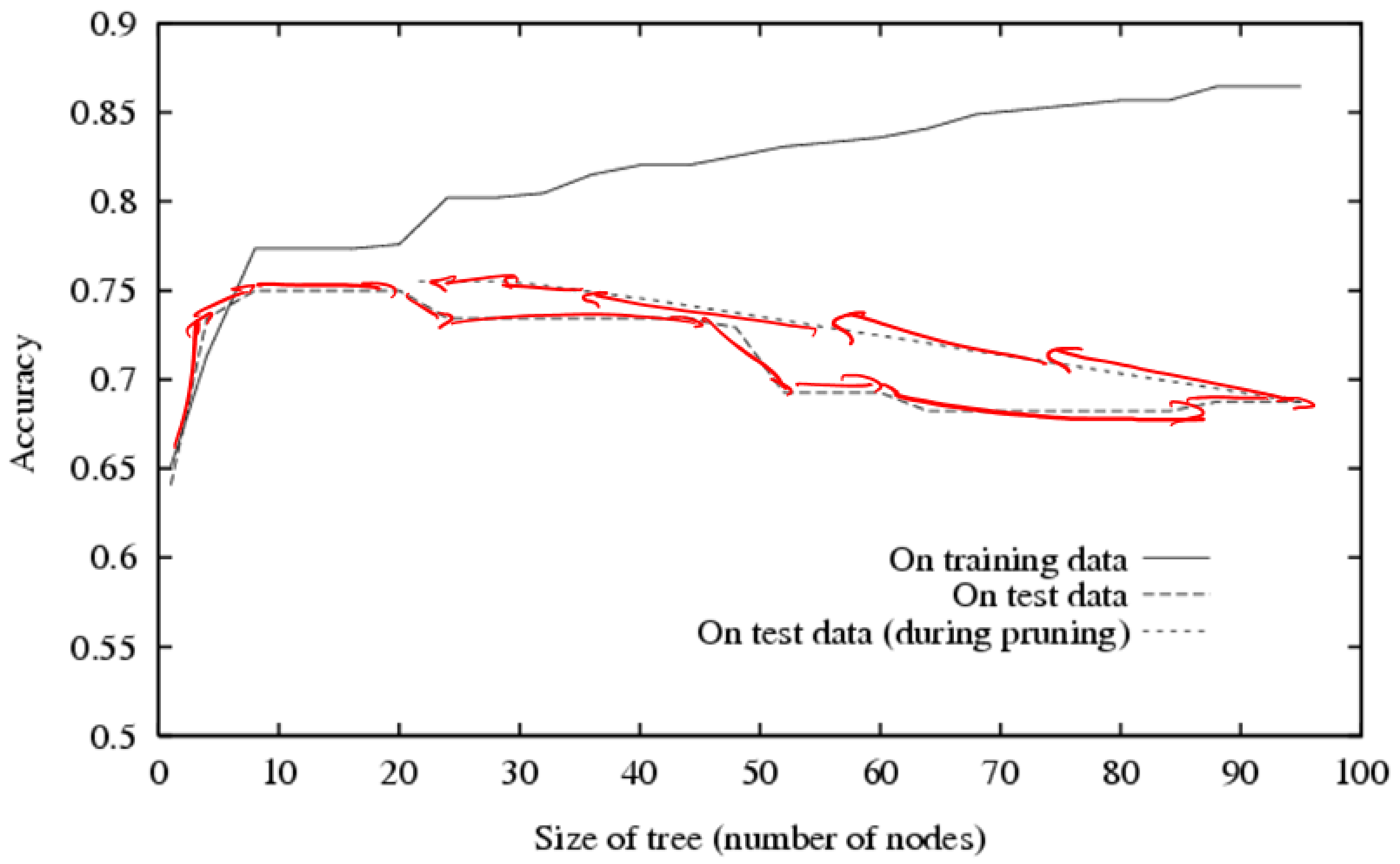
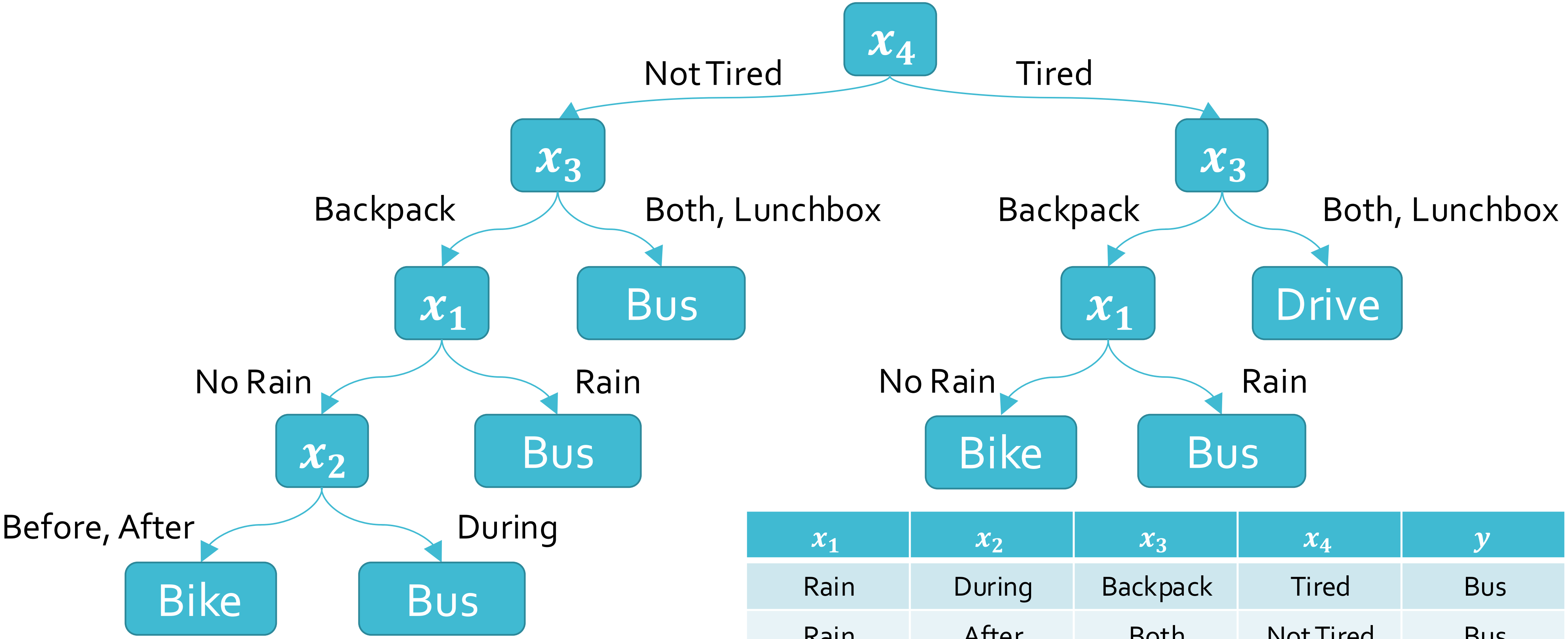
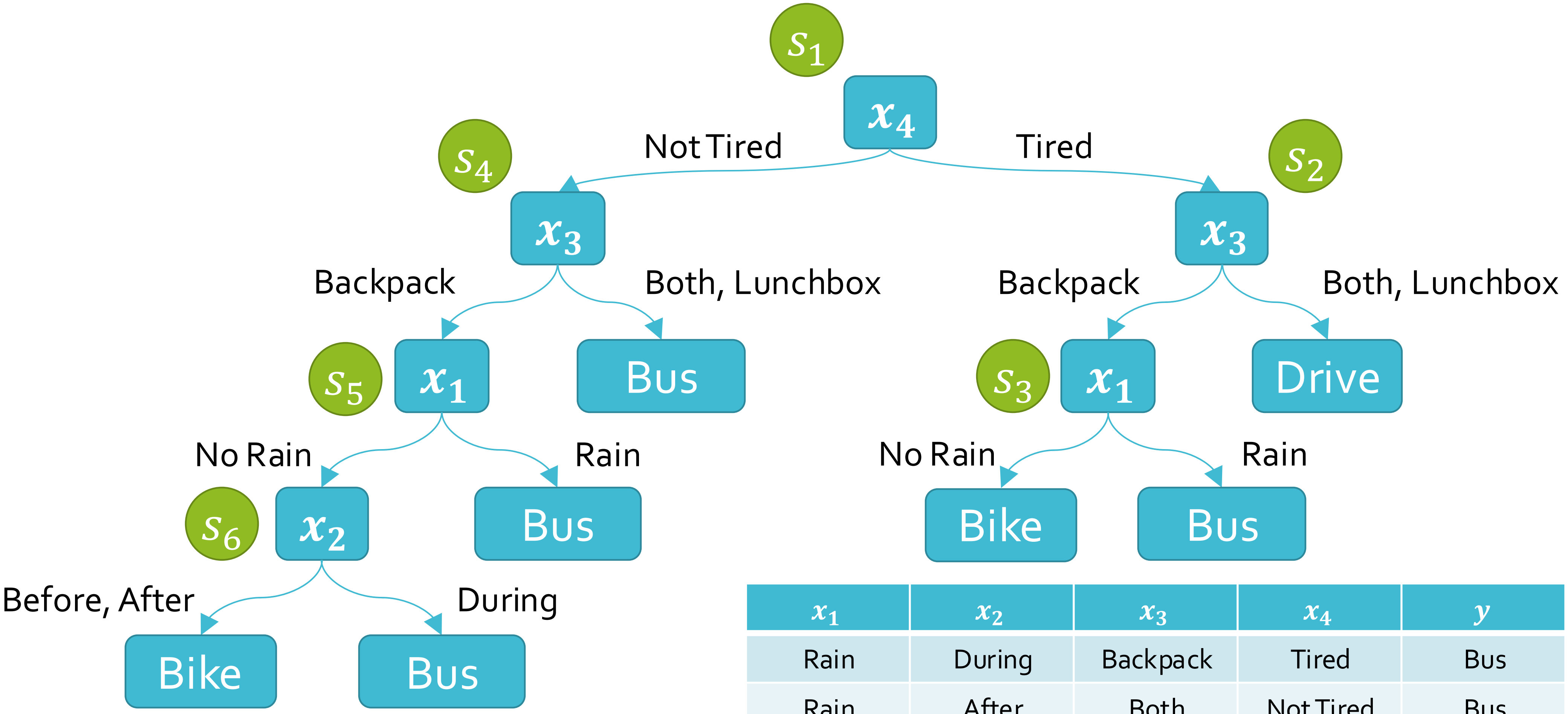


Figure courtesy of Tom Mitchell



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

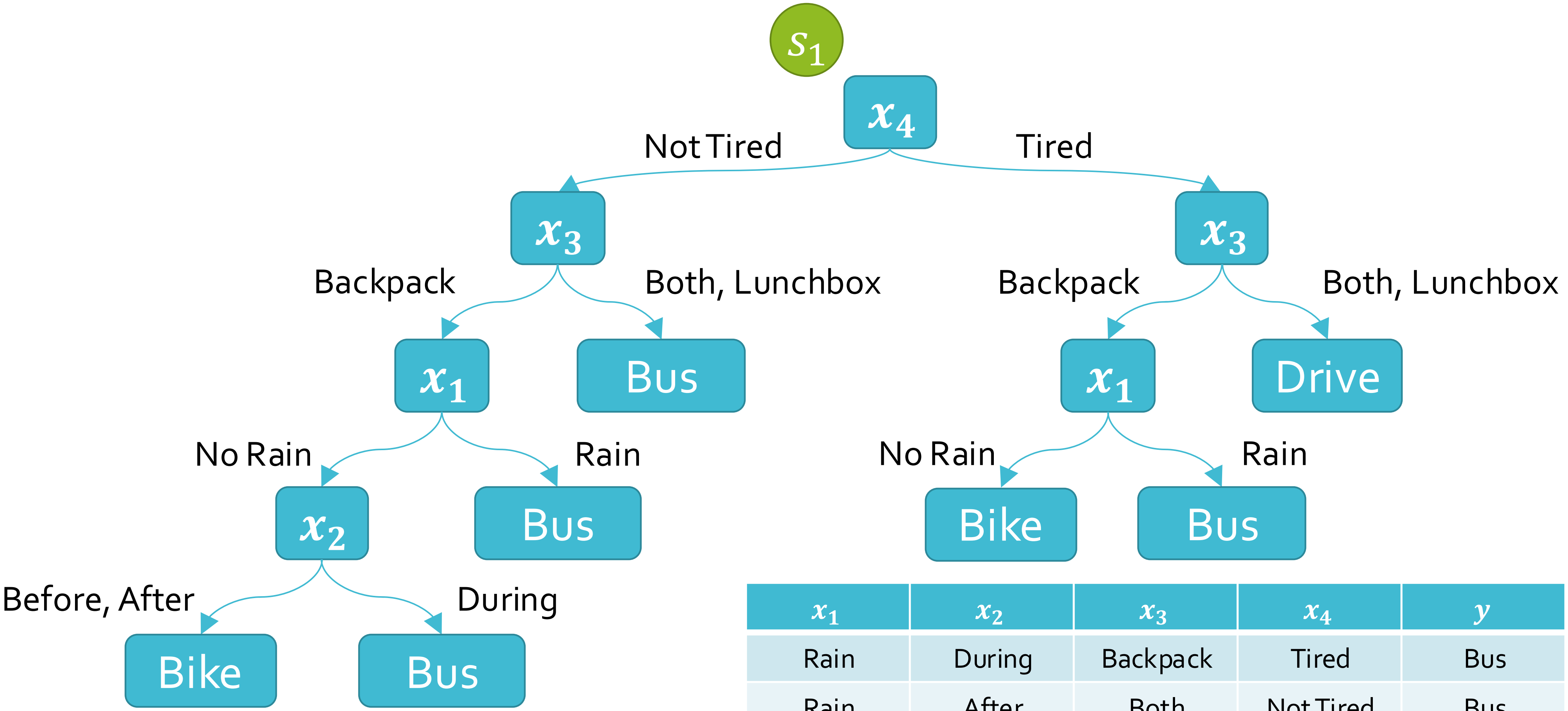
x_1	x_2	x_3	x_4	y
Rain	During	Backpack	Tired	Bus
Rain	After	Both	Not Tired	Bus
No Rain	Before	Backpack	Not Tired	Bus
No Rain	During	Lunchbox	Tired	Drive
No Rain	After	Lunchbox	Tired	Drive



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

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

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

s_1

Bus

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

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

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

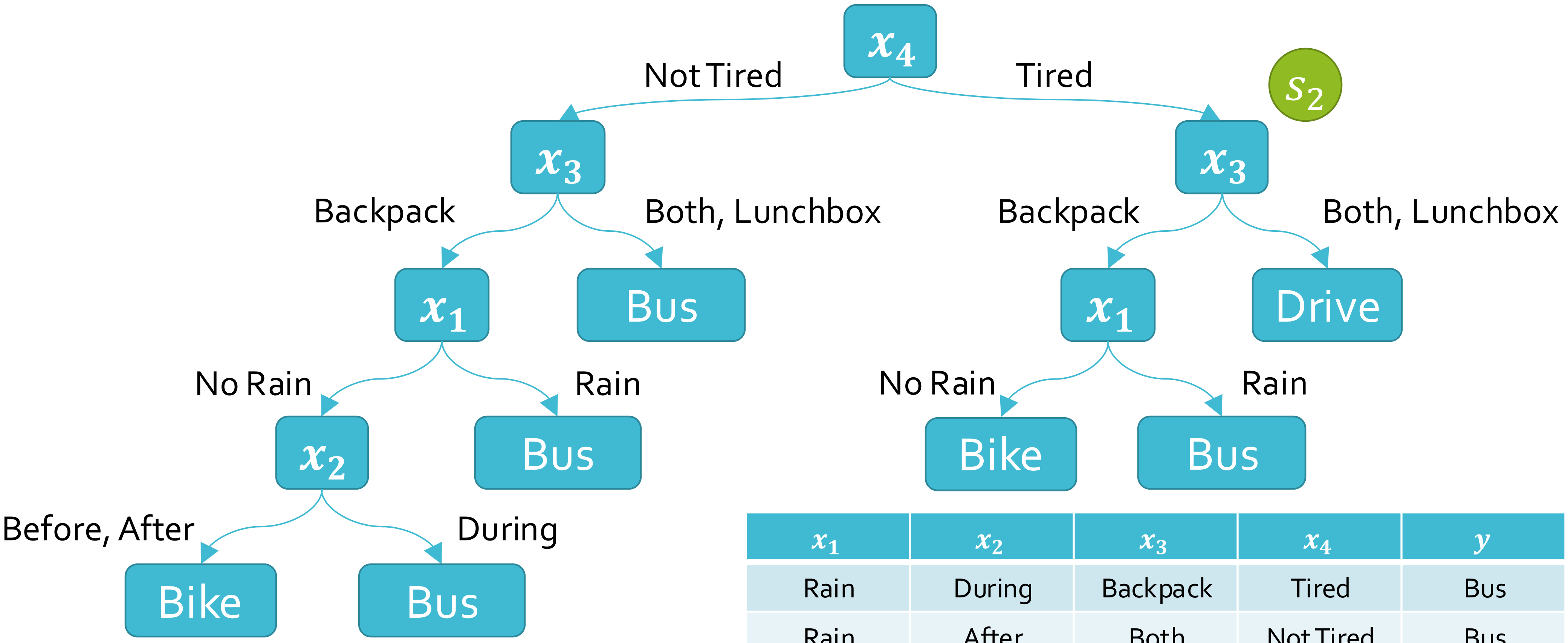
s_1

Bus

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

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

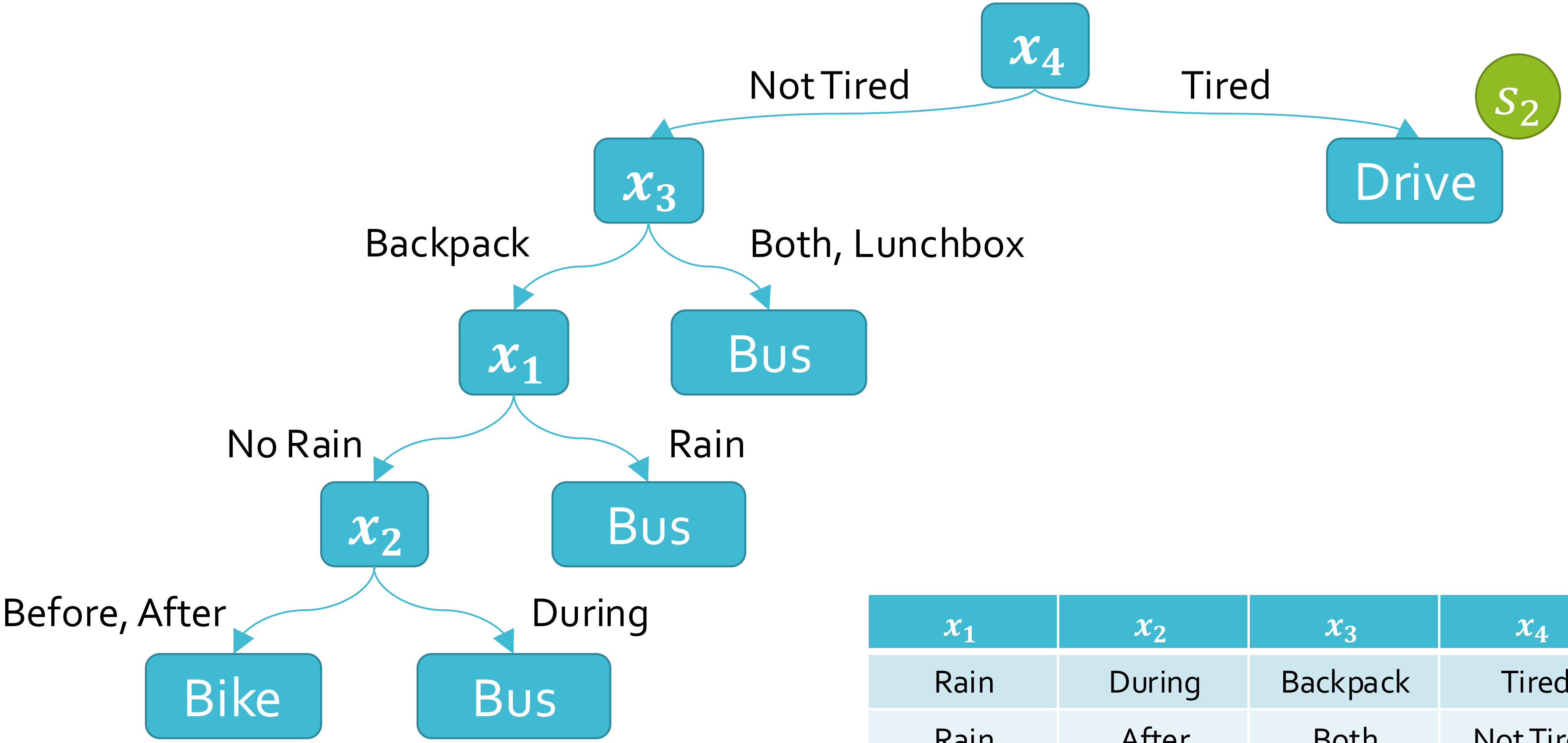
$$err(h - s_1, \mathcal{D}_{val}) = 0.4$$



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

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

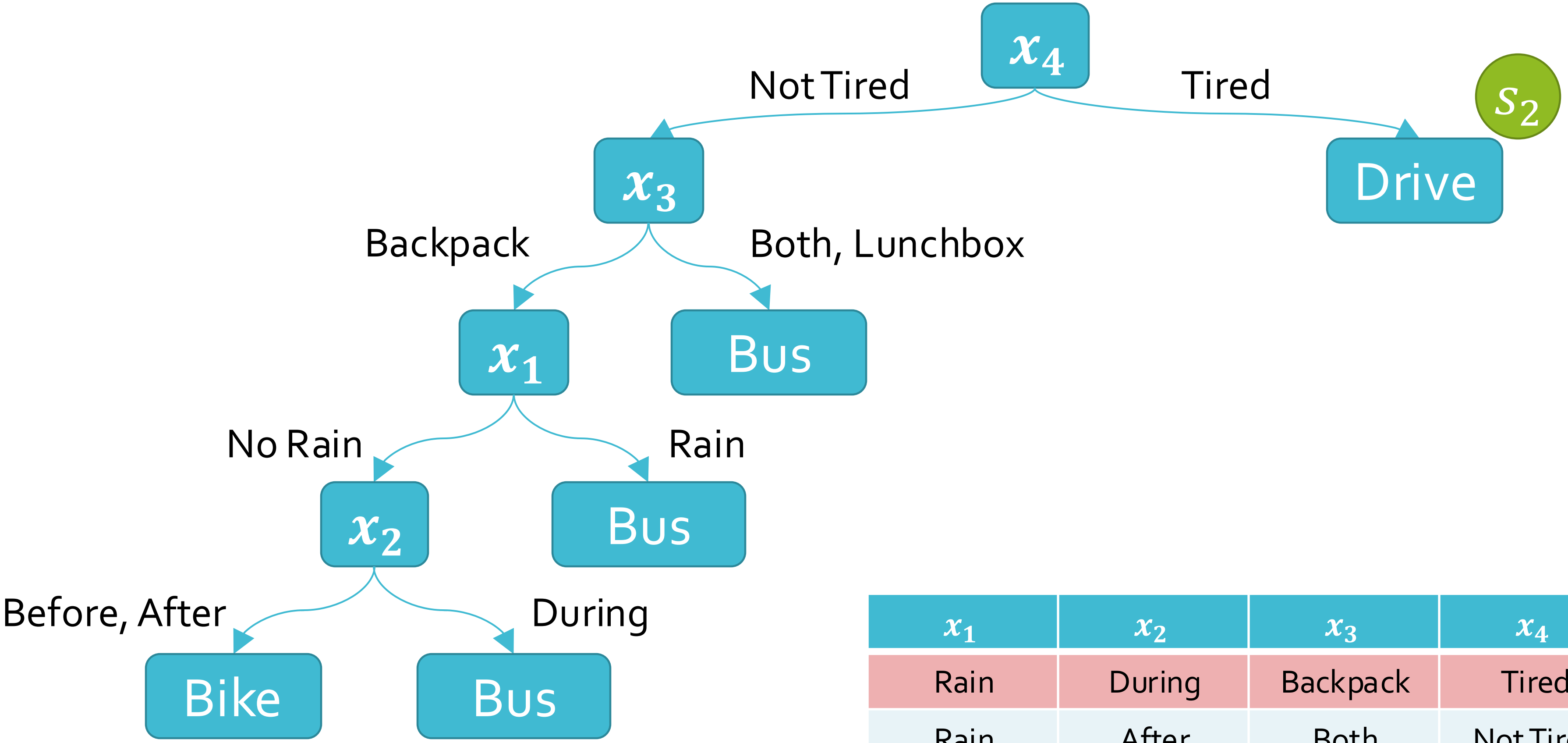
$err(h - s_2, \mathcal{D}_{val})$



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

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

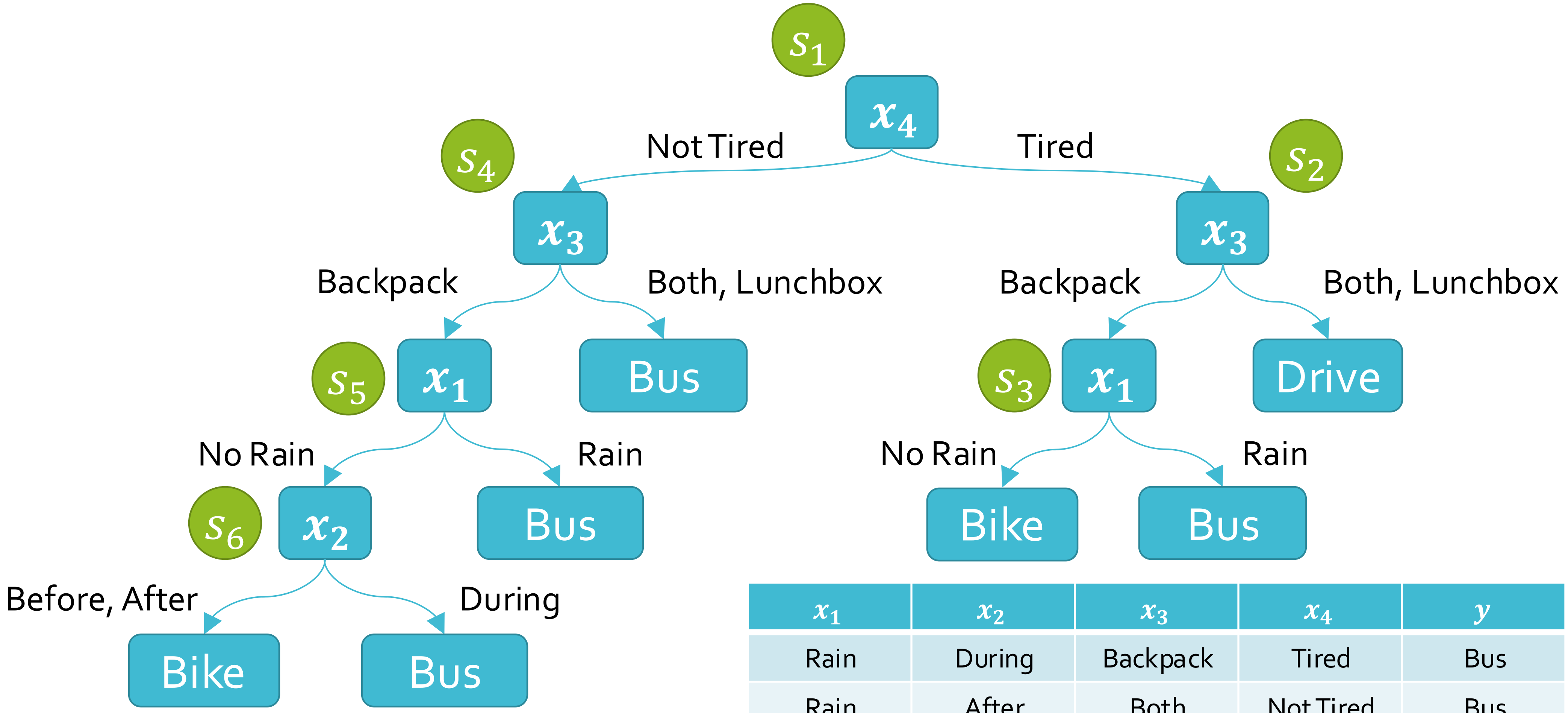
$$err(h - s_2, \mathcal{D}_{val})$$



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

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

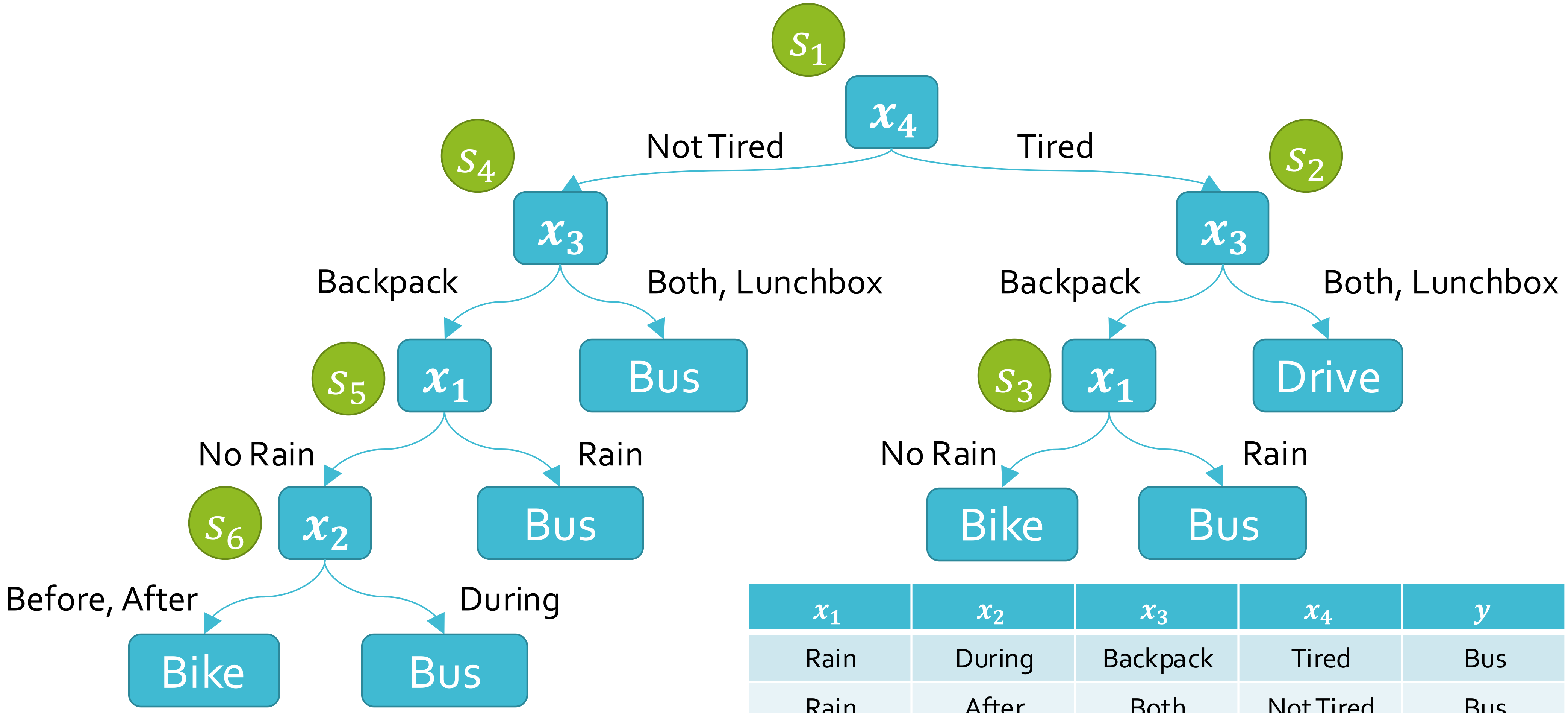
$$err(h - s_2, \mathcal{D}_{val}) = 0.4$$



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

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

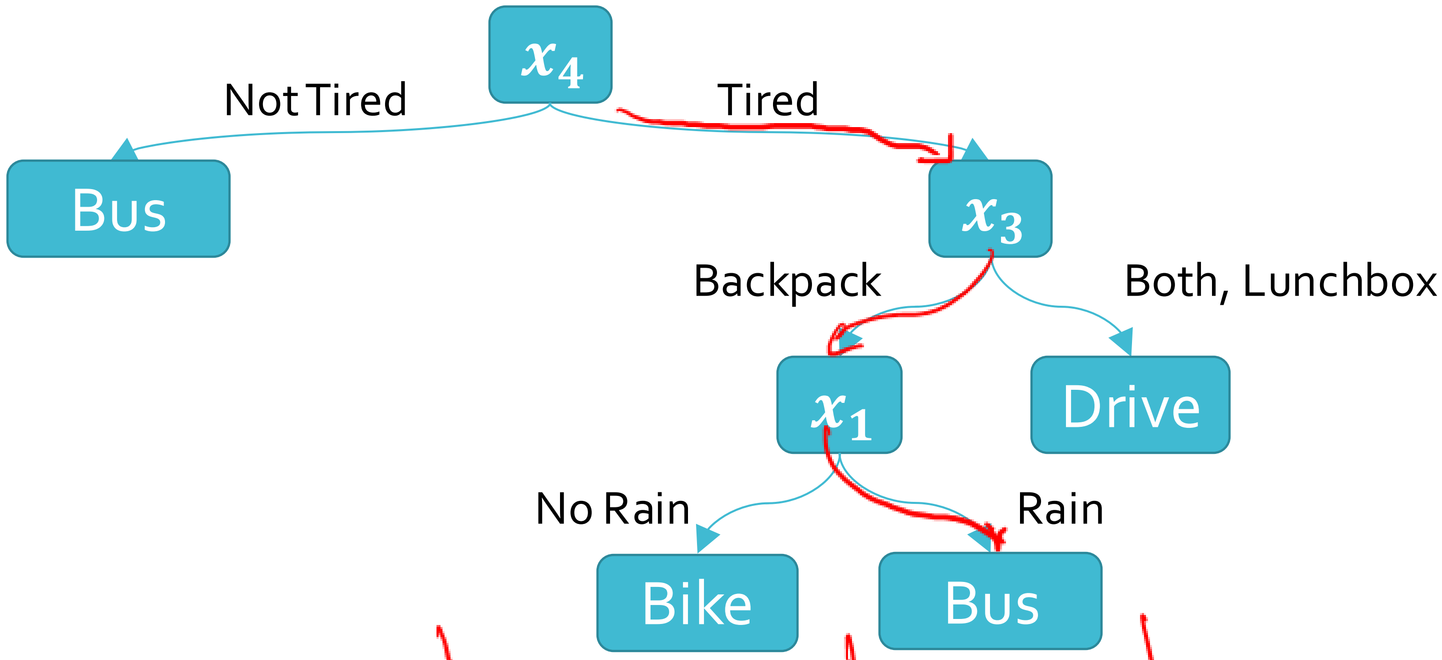
s	s_1	s_2	s_3	s_4	s_5	s_6
$err(h - s, \mathcal{D}_{val})$	0.4	0.4	0.4	0	0	0.2



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

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

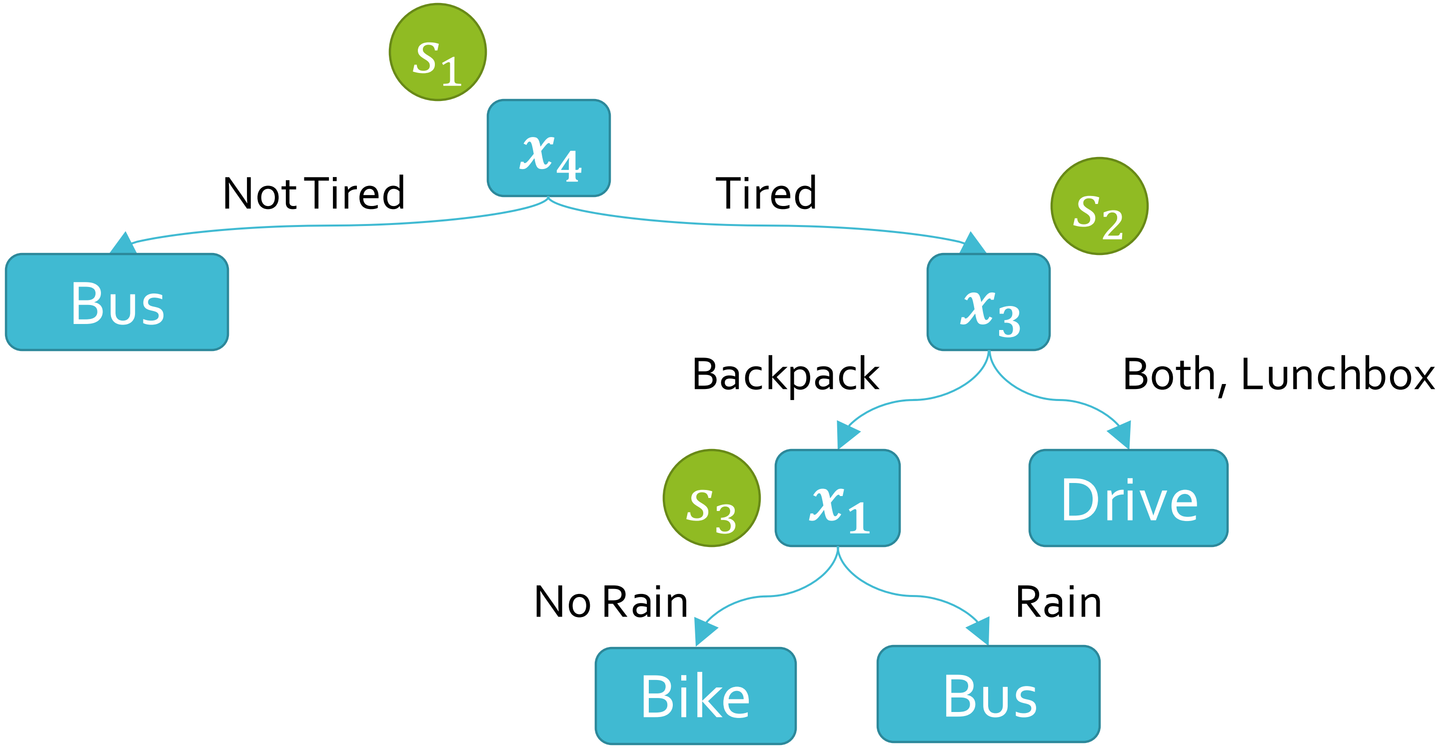
x_1	x_2	x_3	x_4	y
Rain	During	Backpack	Tired	Bus
Rain	After	Both	Not Tired	Bus
No Rain	Before	Backpack	Not Tired	Bus
No Rain	During	Lunchbox	Tired	Drive
No Rain	After	Lunchbox	Tired	Drive



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

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

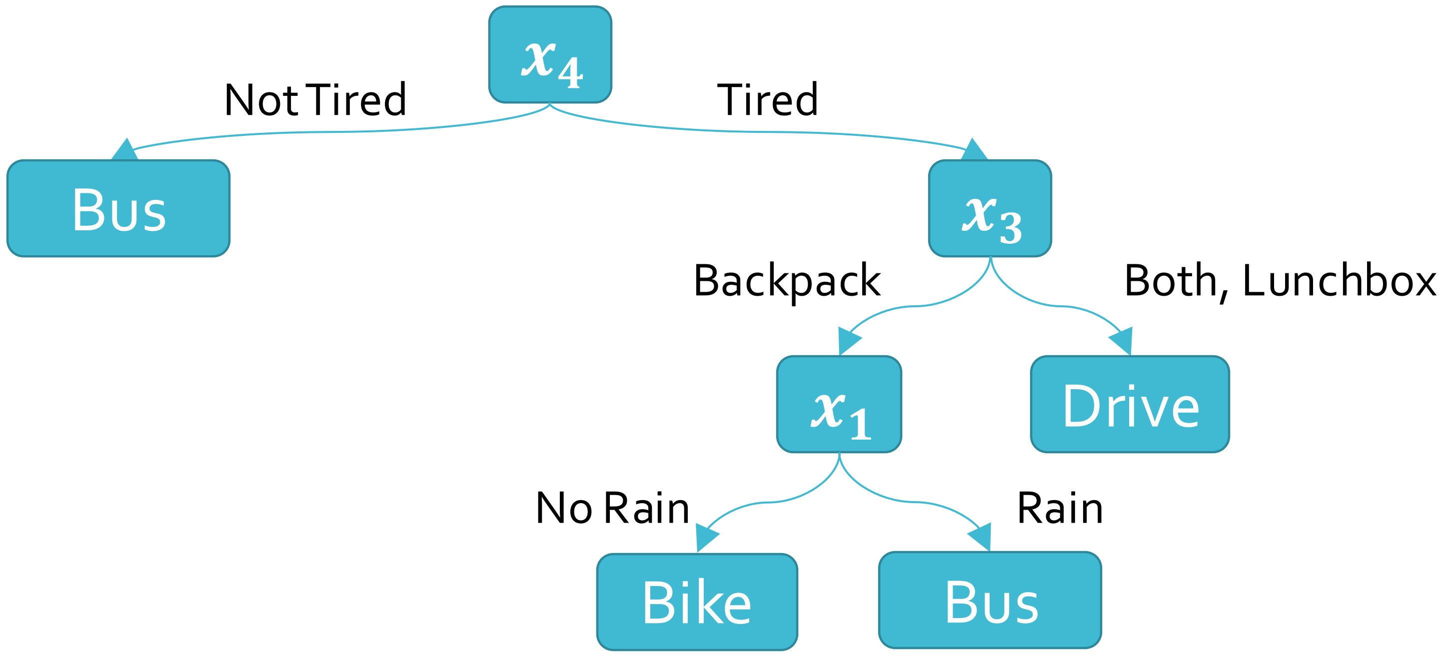
$err(h, \mathcal{D}_{val}) \boxed{= 0}$



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

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

- Inductive bias of decision trees
- Overfitting vs. Underfitting
- How to combat overfitting in decision trees