CSE 417T: Introduction to Machine Learning

Lecture 15: Decision Trees

Henry Chai

10/23/18

Categorical Features

- Up until now, all feature spaces we have considered are of the form $\mathcal{X}^d$ i.e. real-valued features

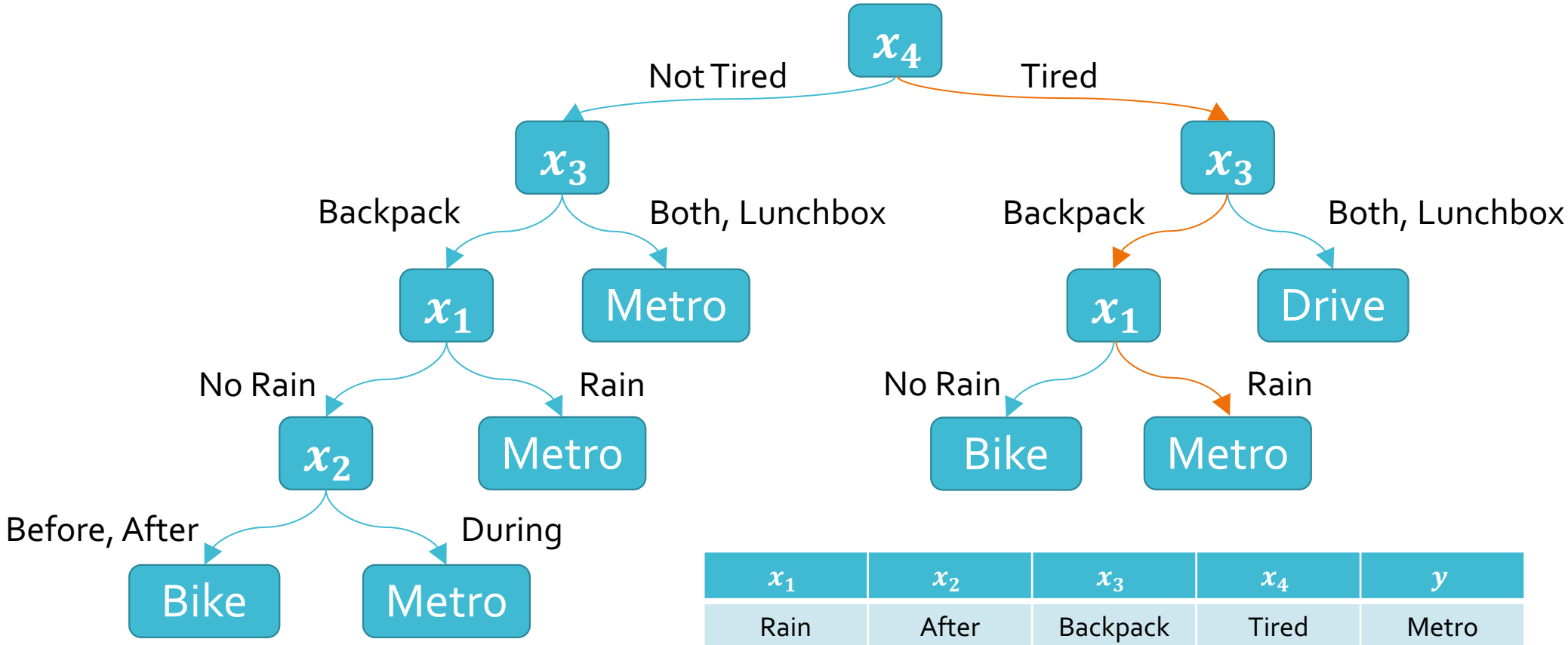
- Frequently, features in the real world are categorical or binary e.g. employment status, profession, race

How should I get to work?

- Output: mode of transportation
  - $y \in \mathcal{Y} = \{\text{Bike}, \text{Drive}, \text{Metro}\}$
- Inputs: 4 categorial features
  - Is it raining? $x_1 \in \mathcal{X}_1 = \{\text{Rain}, \text{ No Rain}\}$
  - When am I leaving (relative to rush hour)?
    $x_2 \in \mathcal{X}_2 = \{\text{Before}, \text{ During}, \text{ After}\}$
  - What am I bringing?
    $x_3 \in \mathcal{X}_3 = \{\text{Backpack}, \text{ Lunchbox}, \text{ Both}\}$
  - Am I tired? $x_4 \in \mathcal{X}_4 = \{\text{Tired}, \text{ Not Tired}\}$

Data

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



x_1	x_2	x_3	x_4	y
Rain	After	Backpack	Tired	Metro

Decision Tree: Example

Entropy

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

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

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

Entropy

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

  $\text{Entropy}(S) = -1\log_2(1) = 0$

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

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

- If S is evenly split between c different values, then

  $\text{Entropy}(S) = c\left(-\frac{1}{c}\log_2\left(\frac{1}{c}\right)\right) = -\log_2\left(\frac{1}{c}\right) = \log_2(c)$

Information Gain

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

- $\mathrm{IG}(x_i, y) = \mathrm{Entropy}(y) - \sum_{v \in V(x_i)} f_v \left(\mathrm{Entropy}\left(y_{x_i=v}\right)\right)$

where x_i is a feature

y is the collection of all labels

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

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

$y_{x_i=v}$ is the collection of labels where $x_i = v$

Information Gain

- Let x_i be a binary feature and suppose that half of all inputs have $x_i = +1$ and $x_i = -1$

- Let y be a binary label and suppose that half of all inputs have $y = +1$ and $y = -1$

x_i	y
+1	+1
+1	+1
-1	-1
-1	-1

- $\mathrm{IG}(x_i, y) = \mathrm{Entropy}(y) - \sum_{v \in \{-1,+1\}} f_v \left(\mathrm{Entropy}\left(y_{x_i=v}\right)\right)$

Information Gain

- Let x_i be a binary feature and suppose that half of all inputs have $x_i = +1$ and $x_i = -1$

- Let y be a binary label and suppose that half of all inputs have $y = +1$ and $y = -1$

x_i	y
+1	+1
+1	+1
-1	-1
-1	-1

- $\mathrm{IG}(x_i, y) = 1 - \frac{1}{2}\mathrm{Entropy}(y_{x_i=-1}) - \frac{1}{2}\mathrm{Entropy}(y_{x_i=+1})$

Information Gain

- Let x_i be a binary feature and suppose that half of all inputs have $x_i = +1$ and $x_i = -1$

- Let y be a binary label and suppose that half of all inputs have $y = +1$ and $y = -1$

x_i	y
+1	+1
+1	+1
-1	-1
-1	-1

- $\mathrm{IG}(x_i, y) = 1 - \frac{1}{2}(0) - \frac{1}{2}(0) = 1$

Information Gain

- Let x_i be a binary feature and suppose that half of all inputs have $x_i = +1$ and $x_i = -1$

- Let y be a binary label and suppose that half of all inputs have $y = +1$ and $y = -1$

x_i	y
+1	+1
+1	-1
-1	+1
-1	-1

- $\mathrm{IG}(x_i, y) = \mathrm{Entropy}(y) - \sum_{v \in \{-1,+1\}} f_v \left(\mathrm{Entropy}\left(y_{x_i=v}\right)\right)$

Information Gain

- Let x_i be a binary feature and suppose that half of all inputs have $x_i = +1$ and $x_i = -1$

- Let y be a binary label and suppose that half of all inputs have $y = +1$ and $y = -1$

x_i	y
+1	+1
+1	-1
-1	+1
-1	-1

- $\mathrm{IG}(x_i, y) = 1 - \frac{1}{2}\mathrm{Entropy}(y_{x_i=-1}) - \frac{1}{2}\mathrm{Entropy}(y_{x_i=+1})$

Information Gain

- Let x_i be a binary feature and suppose that half of all inputs have $x_i = +1$ and $x_i = -1$

- Let y be a binary label and suppose that half of all inputs have $y = +1$ and $y = -1$

x_i	y
+1	+1
+1	-1
-1	+1
-1	-1

- $\mathrm{IG}(x_i, y) = 1 - \frac{1}{2}(1) - \frac{1}{2}(1) = 0$

$$\text{Entropy}(y) = -\frac{3}{16}\log_2\left(\frac{3}{16}\right) - \frac{6}{16}\log_2\left(\frac{6}{16}\right) - \frac{7}{16}\log_2\left(\frac{7}{16}\right)$$

$$\text{Entropy}(y) \approx 1.5052$$

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

$$\mathrm{IG}(x_1, y) = \mathrm{Entropy}(y)$$

$$-\frac{6}{16}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right)-\frac{1}{2}\log_2\left(\frac{1}{2}\right)\right)$$

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

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

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

$$\mathrm{IG}(x_1, y) \approx 1.5052$$

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

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

$$\approx 0.1482$$

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

$\text{IG}(x, y)$	
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	Metro
Rain	During	Both	Tired	Drive
Rain	After	Backpack	Not Tired	Metro
Rain	After	Backpack	Tired	Metro
Rain	After	Lunchbox	Tired	Drive
No Rain	Before	Backpack	Tired	Bike
No Rain	Before	Lunchbox	Not Tired	Metro
No Rain	Before	Lunchbox	Tired	Drive
No Rain	During	Backpack	Not Tired	Metro
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	Metro
No Rain	After	Both	Tired	Drive
No Rain	After	Lunchbox	Not Tired	Metro

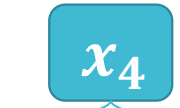
ID3 Learning Algorithm

- Initialize the tree as a single leaf that contains all labels
- While ∃ an impure leaf (not all labels are the same)
  - Pick an arbitrary impure leaf
  - Find the feature, x^*, with the largest information gain relative to the labels in that leaf
  - Create a child (or split) for each unique value of x^*
  - Assign each label in the original leaf to one of its children depending on its corresponding x^* value
    - The original leaf is no longer a leaf
    - All of its children are new leaves

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

x_1	x_2	x_3	x_4	y
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

Decision Tree: Example



Not Tired

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

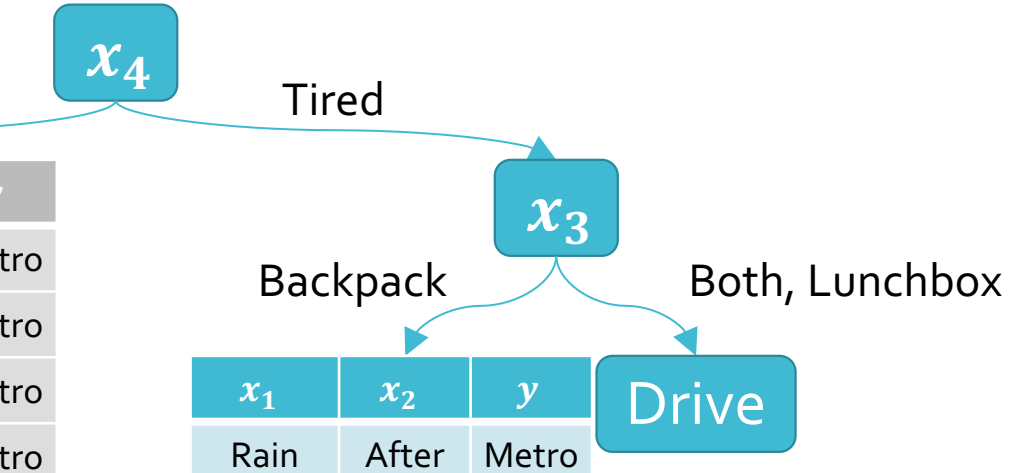
Tired

x_1	x_2	x_3	x_4	y
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

$$\mathrm{IG}\left(x_1, y_{x_4=\mathrm{T}}\right) \approx 0.3244$$

$$\mathrm{IG}\left(x_2, y_{x_4=\mathrm{T}}\right) \approx 0.2516$$

$$\mathrm{IG}\left(x_3, y_{x_4=\mathrm{T}}\right) \approx 0.9183$$



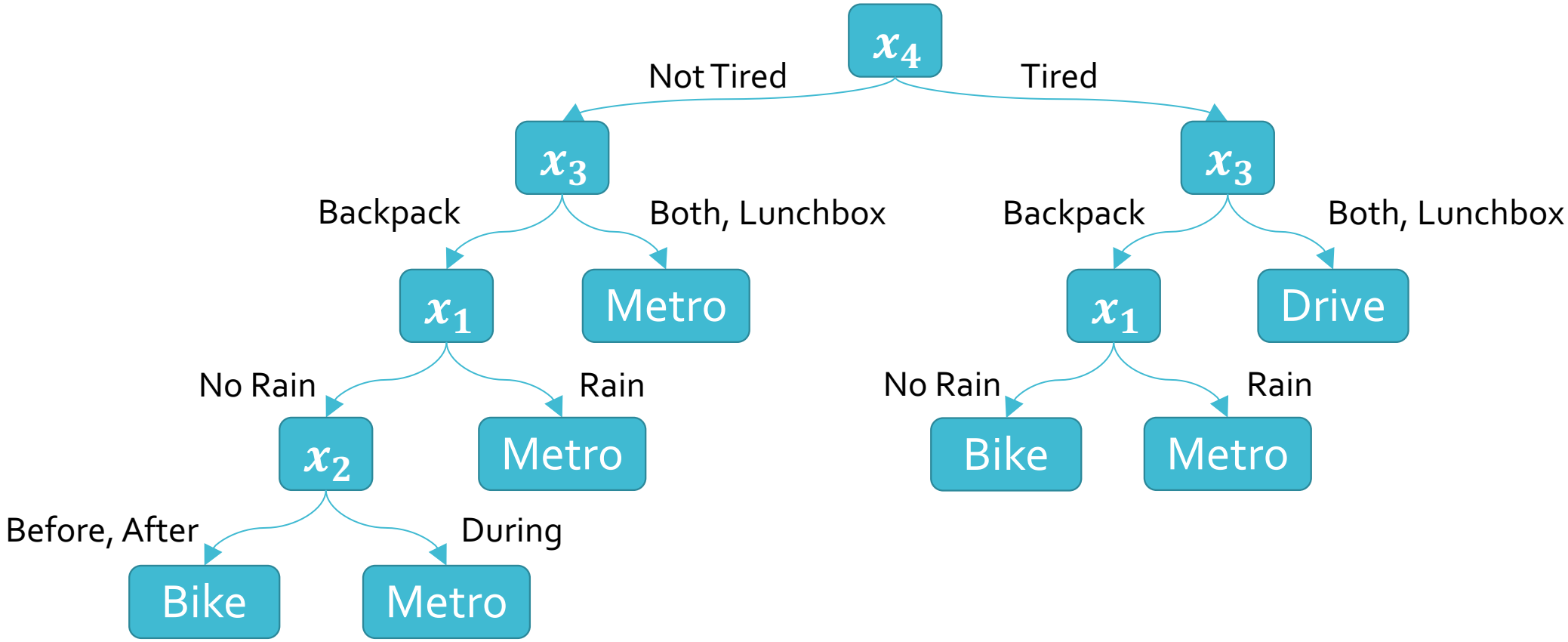
x_1	x_2	x_3	x_4	y
Rain	During	Both	Not Tired	Metro
Rain	After	Backpack	Not Tired	Metro
No Rain	Before	Lunchbox	Not Tired	Metro
No Rain	During	Backpack	Not Tired	Metro
No Rain	After	Backpack	Not Tired	Bike
No Rain	After	Both	Not Tired	Metro
No Rain	After	Lunchbox	Not Tired	Metro

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

$$\mathrm{IG}\left(x_1, y_{x_4=\mathrm{T}}\right) \approx 0.3244$$

$$\mathrm{IG}\left(x_2, y_{x_4=\mathrm{T}}\right) \approx 0.2516$$

$$\mathbf{IG}\left(\boldsymbol{x_3}, \boldsymbol{y_{x_4=T}}\right) \approx \mathbf{0.9183}$$


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

Decision Tree / ID3 Pros

- Intuitive / explainable
- Can handle categorical and real-valued features
- Automatically performs feature selection
- The ID3 algorithm has a preference for shorter trees (simpler hypotheses)

Real-Valued Features: Example

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

$x \geq 38.5$

Real-Valued Features: Example

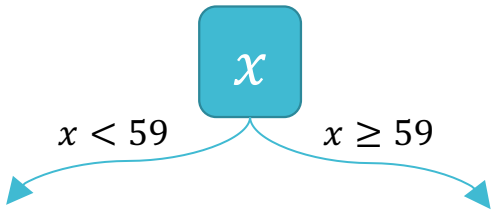
x	y
74	Drive
55	Metro
63	Bike
33	Drive
80	Drive
81	Drive
44	Metro
45	Metro
78	Drive
51	Metro

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

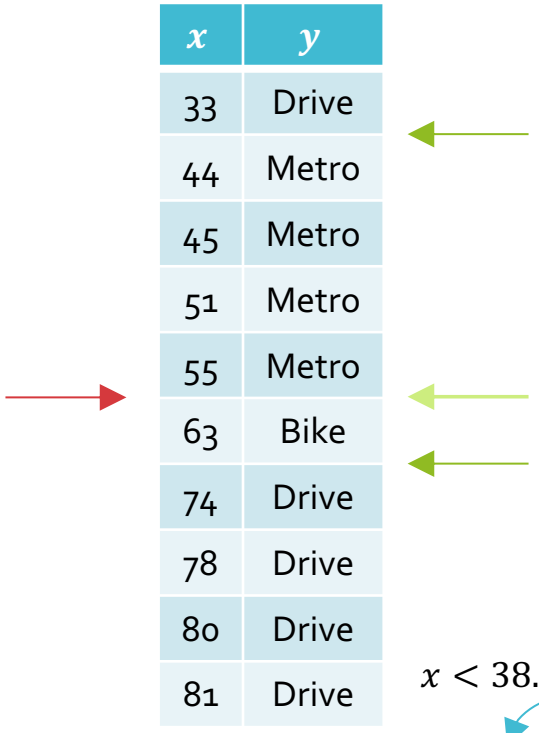
x	y
74	Drive
55	Metro
63	Bike
33	Drive
80	Drive
81	Drive
44	Metro
45	Metro
78	Drive
51	Metro

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

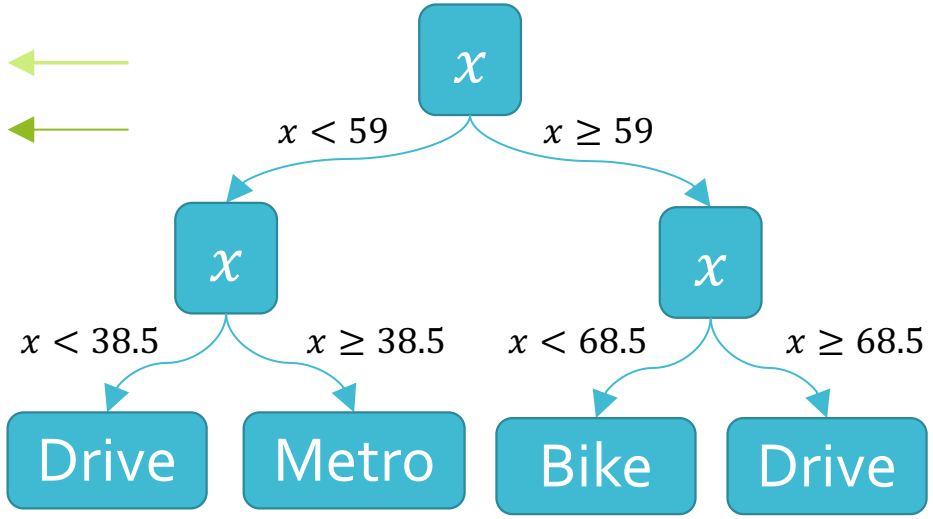


Real-Valued Features: Example

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

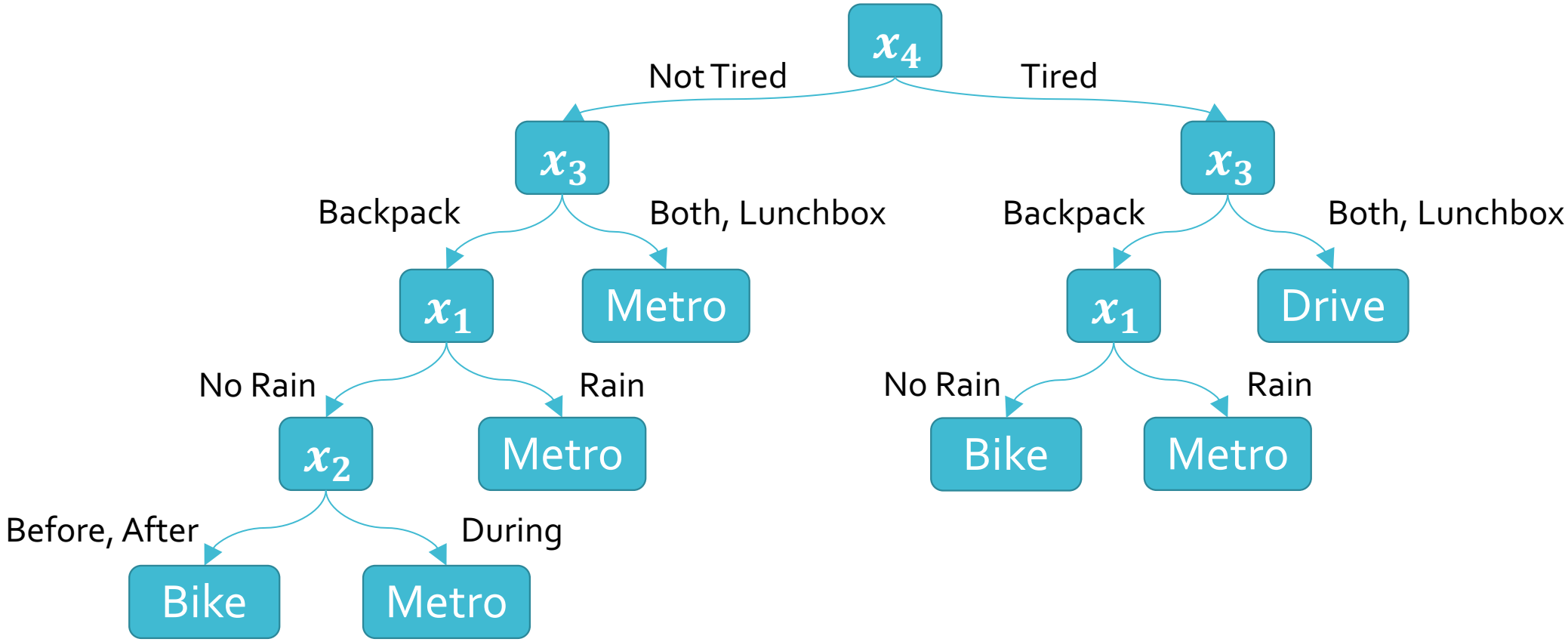


x	y
33	Drive
44	Metro
45	Metro
51	Metro
55	Metro
63	Bike
74	Drive
78	Drive
80	Drive
81	Drive

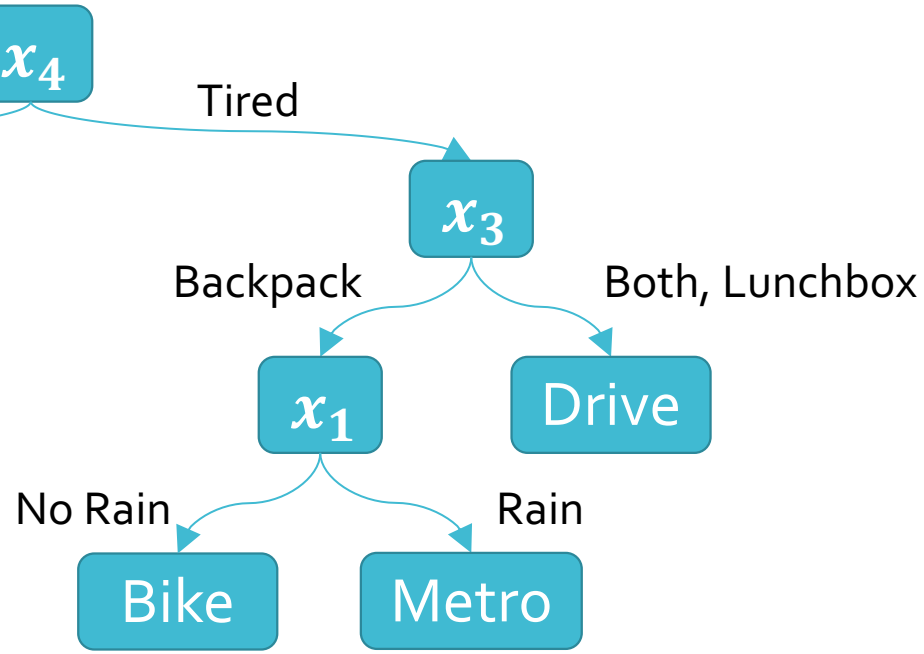


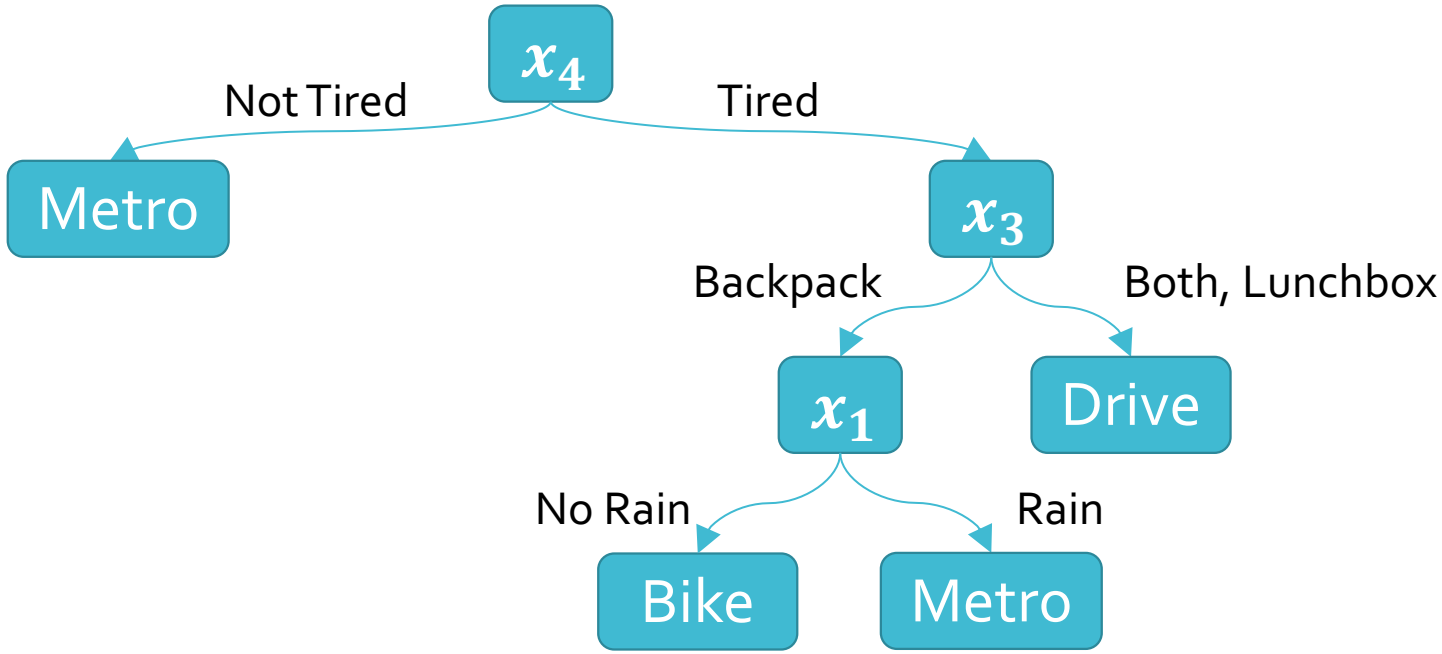
Decision Tree / ID3 Cons

- The ID3 algorithm is greedy (it selects the feature w/ the highest information gain at every step) so no optimality guarantee
- Overfitting!


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

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





This tree only misclassifies one training point

Addressing Overfitting

- Heuristics ("regularization"):
  - Do not split leaves past a fixed depth δ
  - Do not split leaves with fewer than c labels
  - Do not split leaves where the maximal information gain is less than τ
  - Predict the most common label at each leaf
- Pruning ("validation"):
  - Evaluate each split using a validation set
  - Compare the validation error with and without that split (replacing it with the most common label at that point)