CS1138

Machine Learning

Lecture : Decision Trees

(Credits: Sebastian Raschka and Kilian Weinberger)

Arpan Gupta

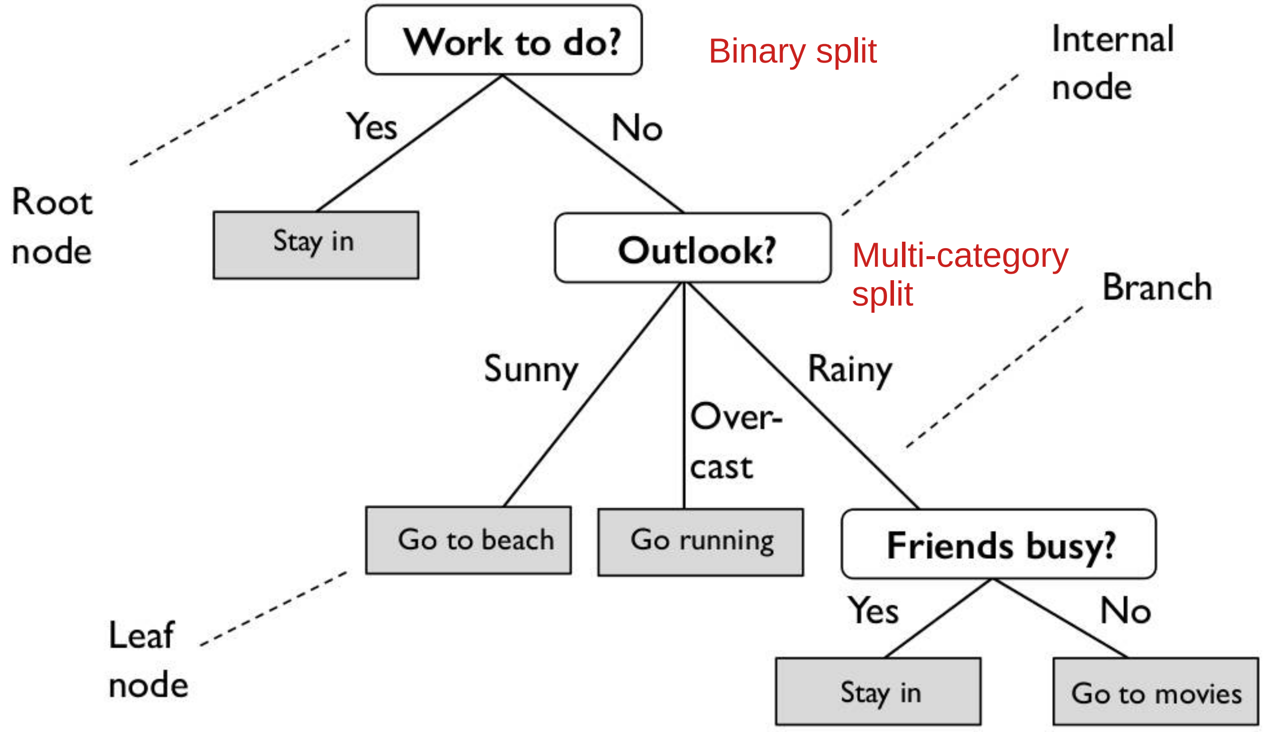
Overview

- Intro to Decision Trees
- Types of DT (CART, ID3)
- Splitting Criteria
- Information Gain, Gini Impurity and Entropy
- ID3 Example

Intro to DTs

- It is a hierarchical set of rules explaining the way in which a large dataset can be divided into smaller data partitions. Each time a split takes place, the components of the resulting partitions become increasingly similar to one another with regard to the target.
- They are interpretable models.
- The decision/predictions can be explained with a set sequence of rules. (Explainable AI)

Decision Tree Terminology

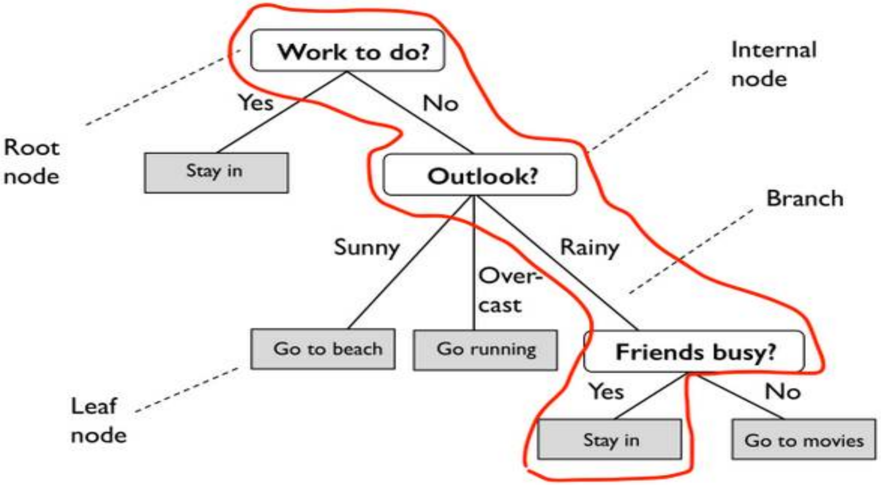


Binary split

Multi-category split

We can also convert Decision Trees with categorical splits into trees having only Binary splits.

Decision Trees as Rulesets



IF Work to do = No ∩ Outlook = Rainy

∩ Friends busy = Yes

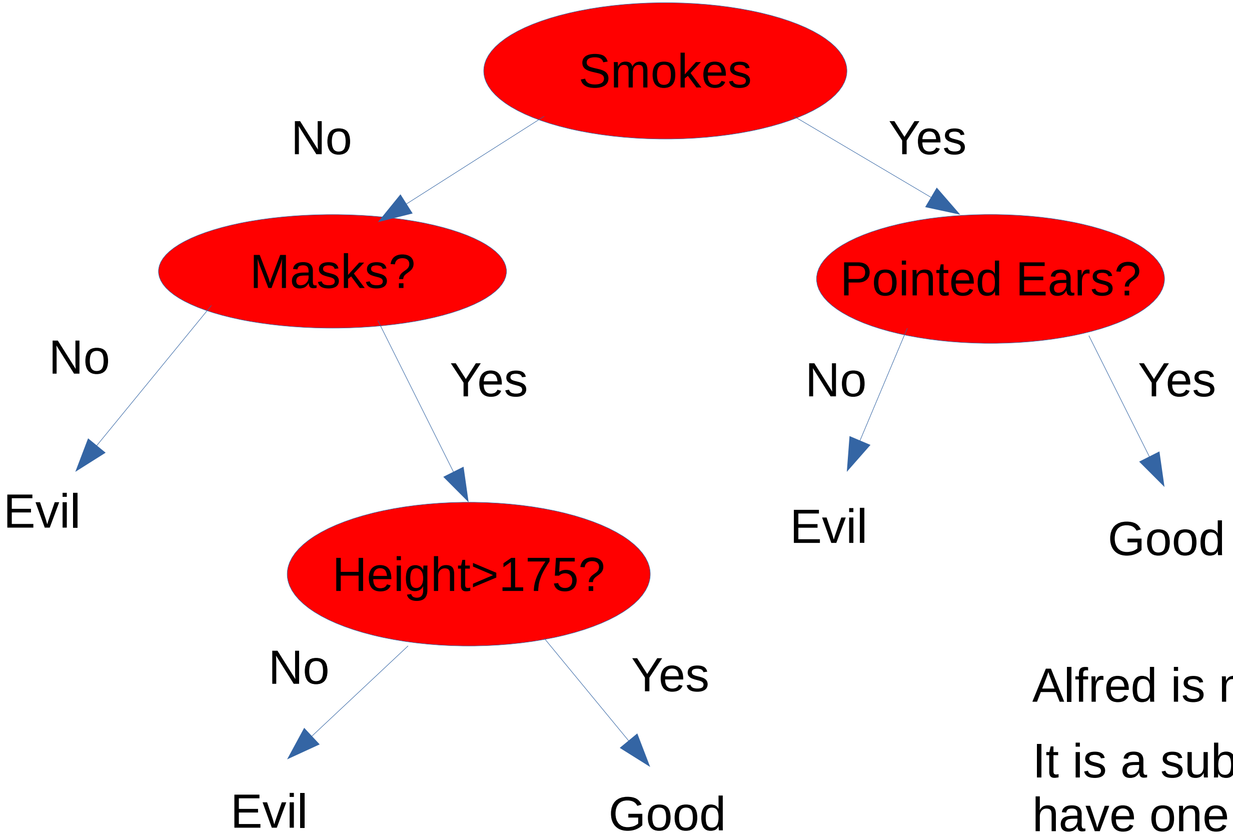
U Work to do = Yes

THEN Stay inside

Example: Is your friend good or evil?

	Mask	**Cape**	**Tie**	**Pointy Ears**	**Smokes**	**Height**	**Class**
Batman	Y	Y	N	Y	N	180	Good
Robin	Y	Y	N	N	N	176	Good
Alfred	N	N	Y	N	N	185	Good
Penguin	N	N	Y	N	Y	140	Evil
Catwoman	Y	N	N	Y	N	170	Evil
Joker	N	N	N	N	N	179	Evil
Batgirl	**Y**	**Y**	**N**	**Y**	**N**	**165**	**?**
Riddler	**Y**	**N**	**N**	**N**	**Y**	**182**	**?**
Your Friend	**N**	**Y**	**Y**	**Y**	**Y**	**181**	**?**

Example: A sample Tree



Is this a good tree?

Alfred is misclassified.

It is a sub-optimal tree. We may have one more split w.r.t. **Tie,** to get a consistent tree.

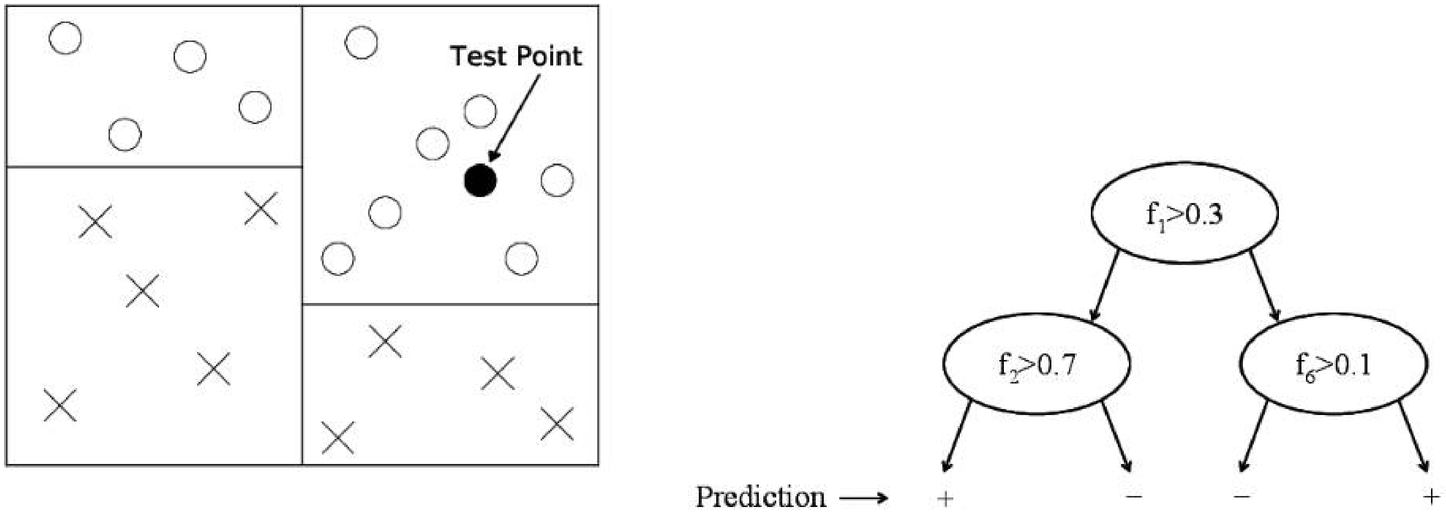
Note

- We can keep splitting until we get everything right.
- We do not want to split on every point.
  - This will overfit on the training set.
  - For eg. A 1M size dataset, will have around 1M leaf nodes. Therefore, overfitting occurs.

Example: Find the smallest tree that gets all training points correct.

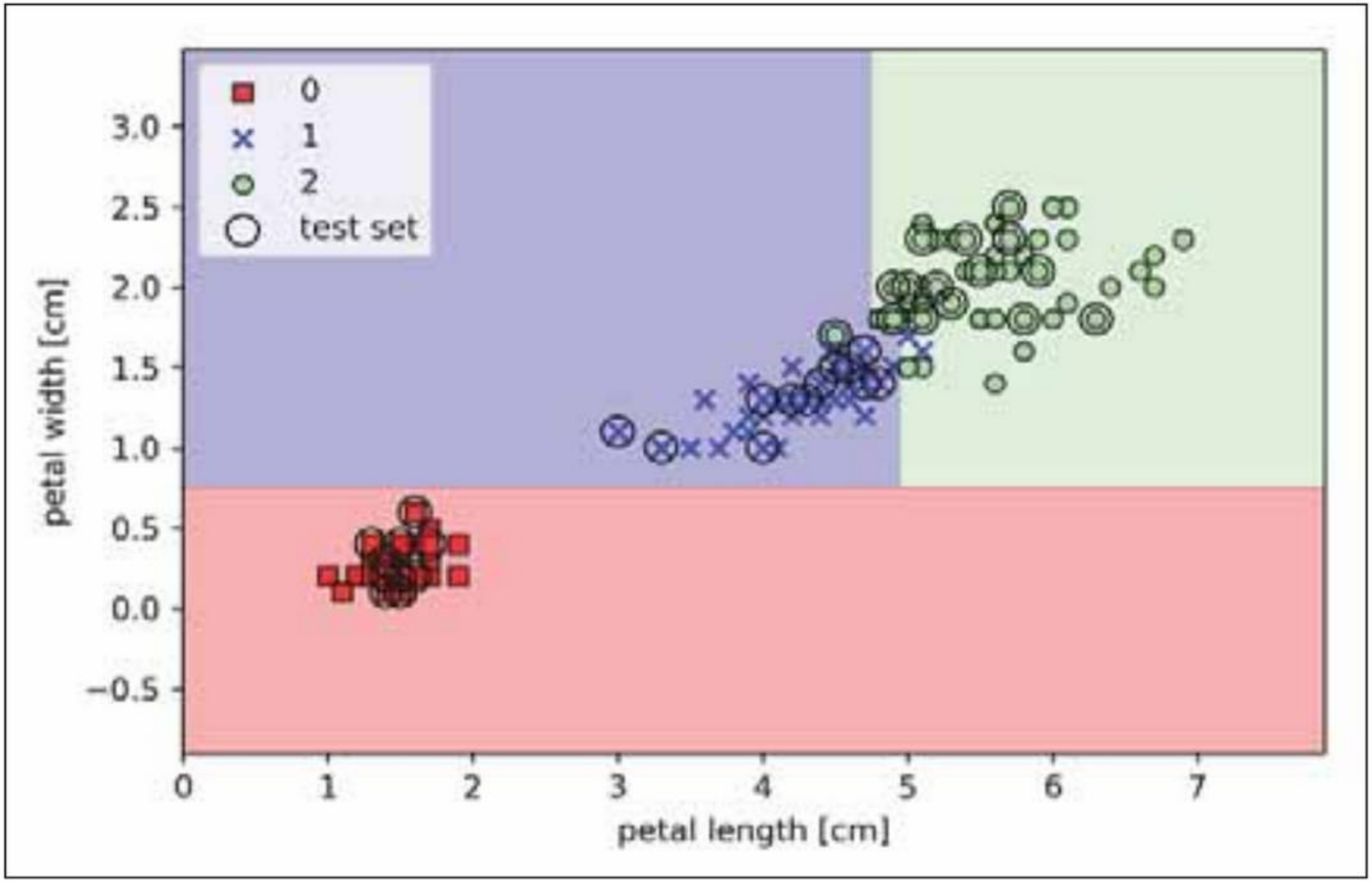
- Split by **Cape?** And then split by **Height>180?**
- **Construct the above tree.**

Visualizing decision boundaries



Binary decision tree. Only labels are stored.

Visualizing decision boundaries



Note

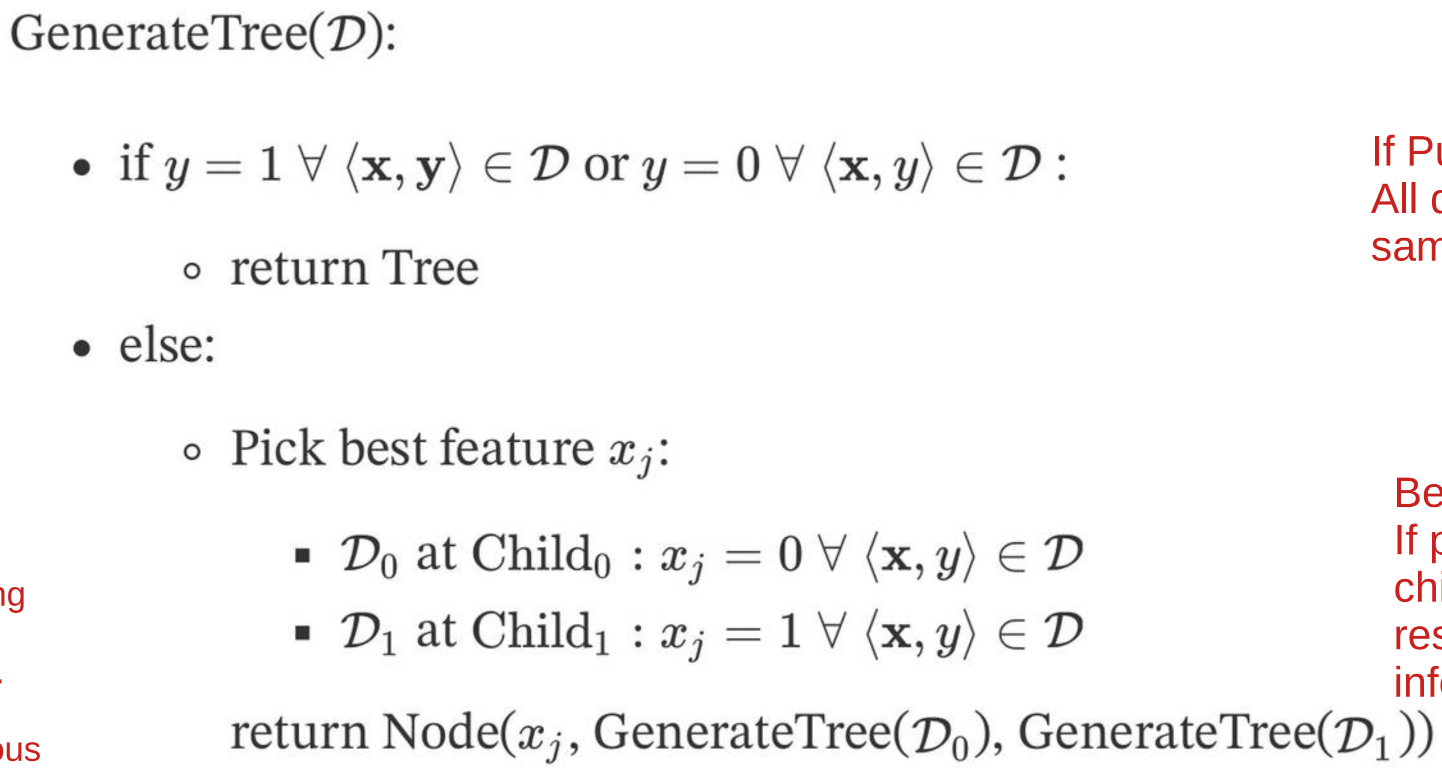
- When do DTs have high bias and when do they have high variance? (as a function of depth of the tree)
- **Answer:**
- Low Depth means high bias (Small tree)
- High Depth mean high variance (large tree)

- Finding the smallest tree that gets all the training set correct, is an NP-hard problem, i.e., as the dataset gets larger, the amount of extra computation we have to do to accomodate the extra data grows exponentially.

Note

- Decision Trees are terrible ML algorithms.
- However, they have such clear bias-variance problems, that it is very easy to address those problems.
- If we address the variance with **“bagging”**, and address the bias with **“boosting”**, then they become very good.
- Search engines are just boosted DTs.
- Random Forests are bagged DTs and are very good.

Decision Tree in Pseudocode



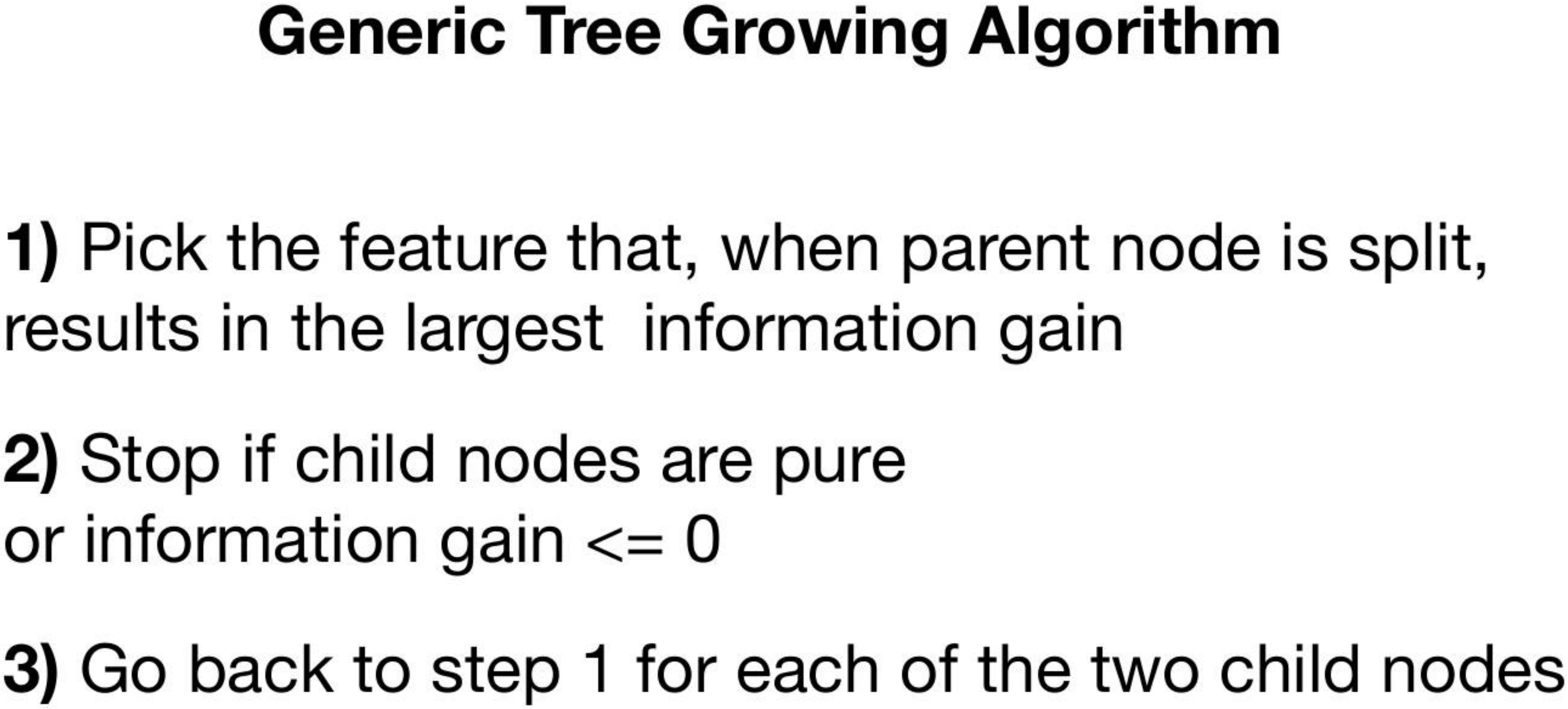
```
GenerateTree($\mathcal{D}$):
    • if $y = 1 \; \forall \; \langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{D}$ or $y = 0 \; \forall \; \langle \mathbf{x}, y \rangle \in \mathcal{D}$ :
        ◦ return Tree
    • else:
        ◦ Pick best feature $x_j$:
            ▪ $\mathcal{D}_0$ at Child$_0$ : $x_j = 0 \; \forall \; \langle \mathbf{x}, y \rangle \in \mathcal{D}$
            ▪ $\mathcal{D}_1$ at Child$_1$ : $x_j = 1 \; \forall \; \langle \mathbf{x}, y \rangle \in \mathcal{D}$
        return Node($x_j$, GenerateTree($\mathcal{D}_0$), GenerateTree($\mathcal{D}_1$))
```

If Pure node is reached?
All data points have
same label.

Best is defined later.
If parent node is split into
child nodes then it
results in the largest
information gain.

Assuming
binary
variable.
Can be
continuous
value.

How to decide the feature for splitting?



Generic Tree Growing Algorithm

1) Pick the feature that, when parent node is split, results in the largest information gain

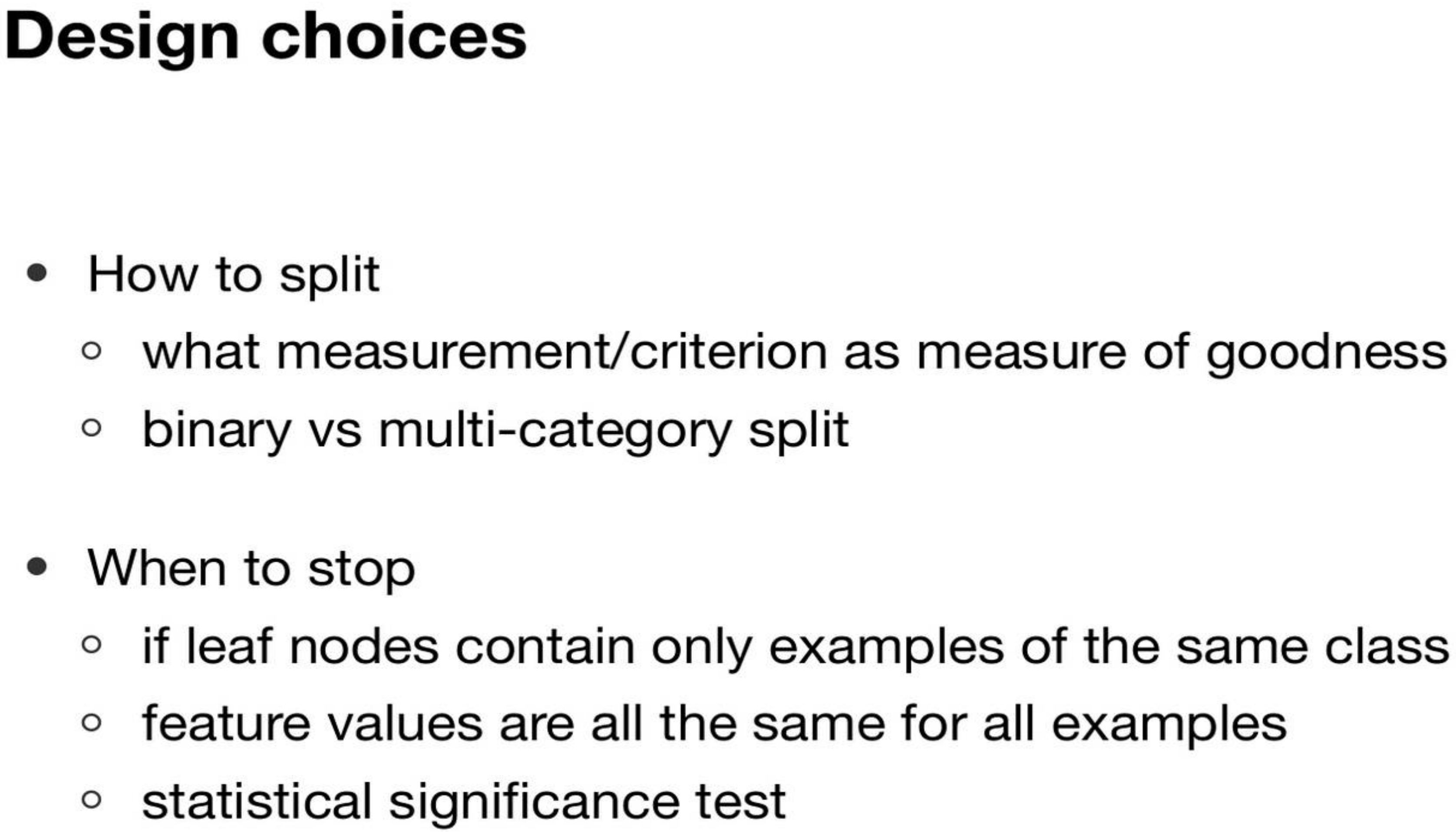
2) Stop if child nodes are pure or information gain <= 0

3) Go back to step 1 for each of the two child nodes

There are also methods for pruning.

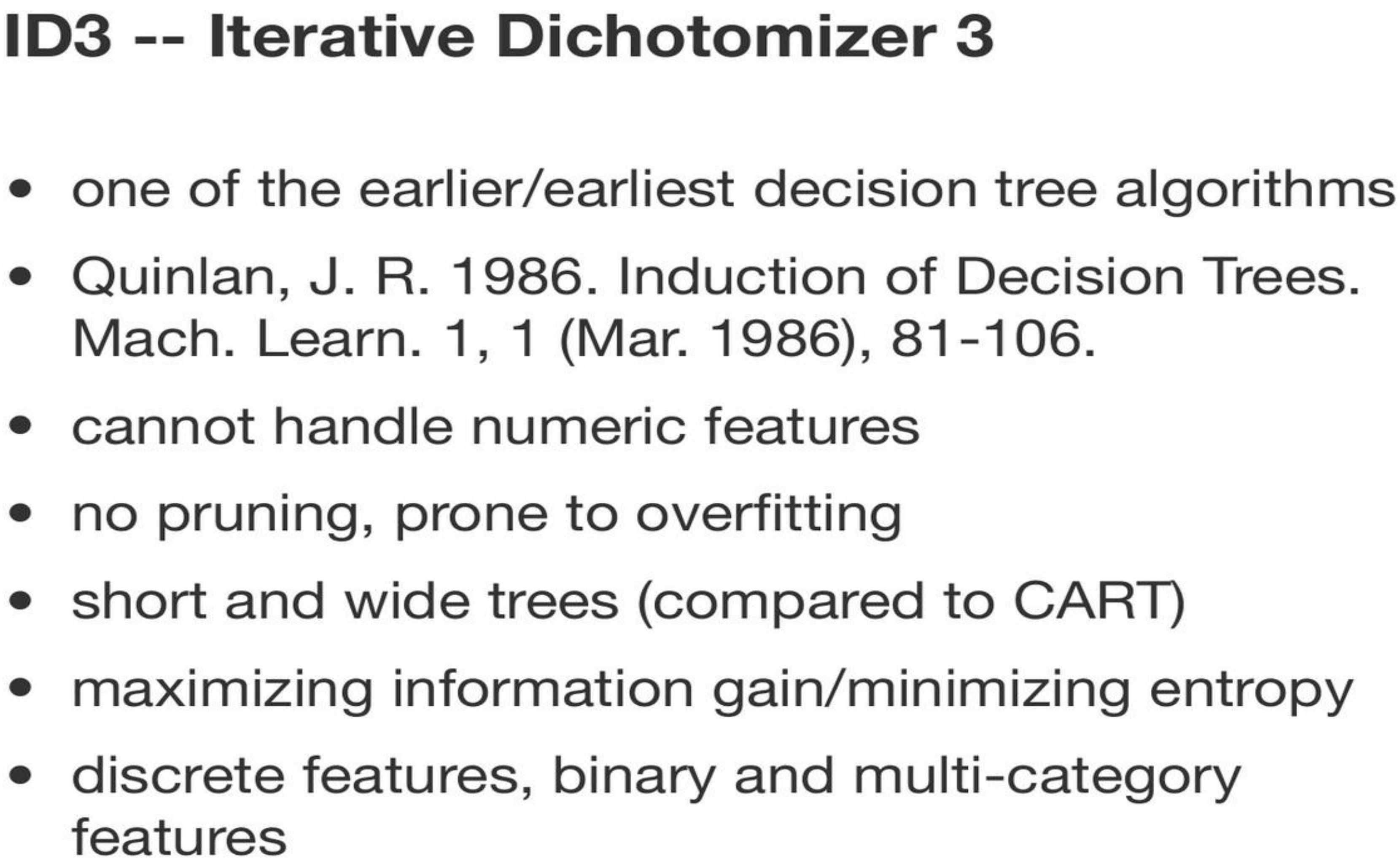
Homework

- Analyze the time complexity for constructing a decision tree with only binary splits.
- Analyze the time complexity for querying a decision tree with only binary splits.



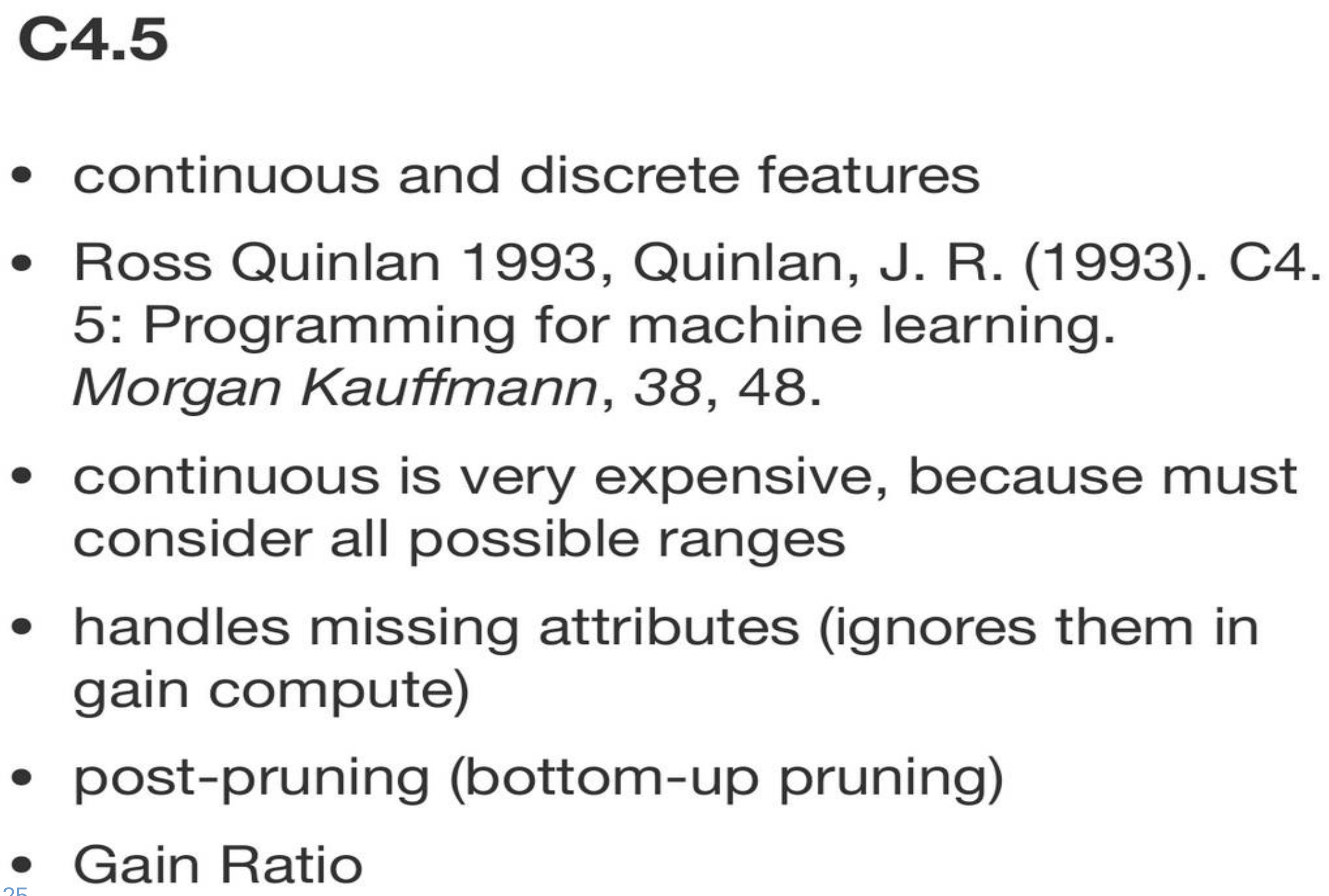
Design choices

- How to split
  - what measurement/criterion as measure of goodness
  - binary vs multi-category split
- When to stop
  - if leaf nodes contain only examples of the same class
  - feature values are all the same for all examples
  - statistical significance test



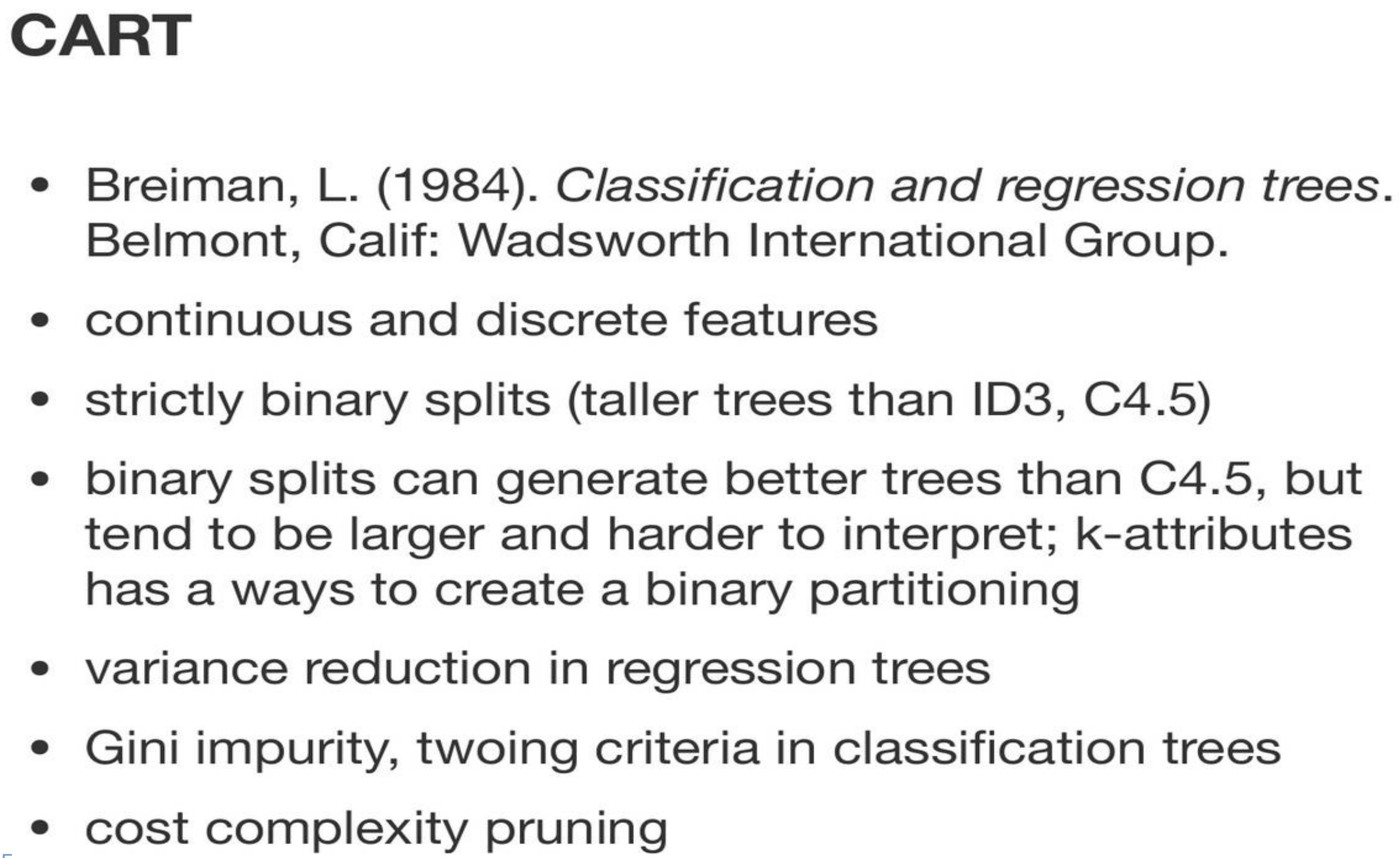
ID3 -- Iterative Dichotomizer 3

- one of the earlier/earliest decision tree algorithms
- Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106.
- cannot handle numeric features
- no pruning, prone to overfitting
- short and wide trees (compared to CART)
- maximizing information gain/minimizing entropy
- discrete features, binary and multi-category features



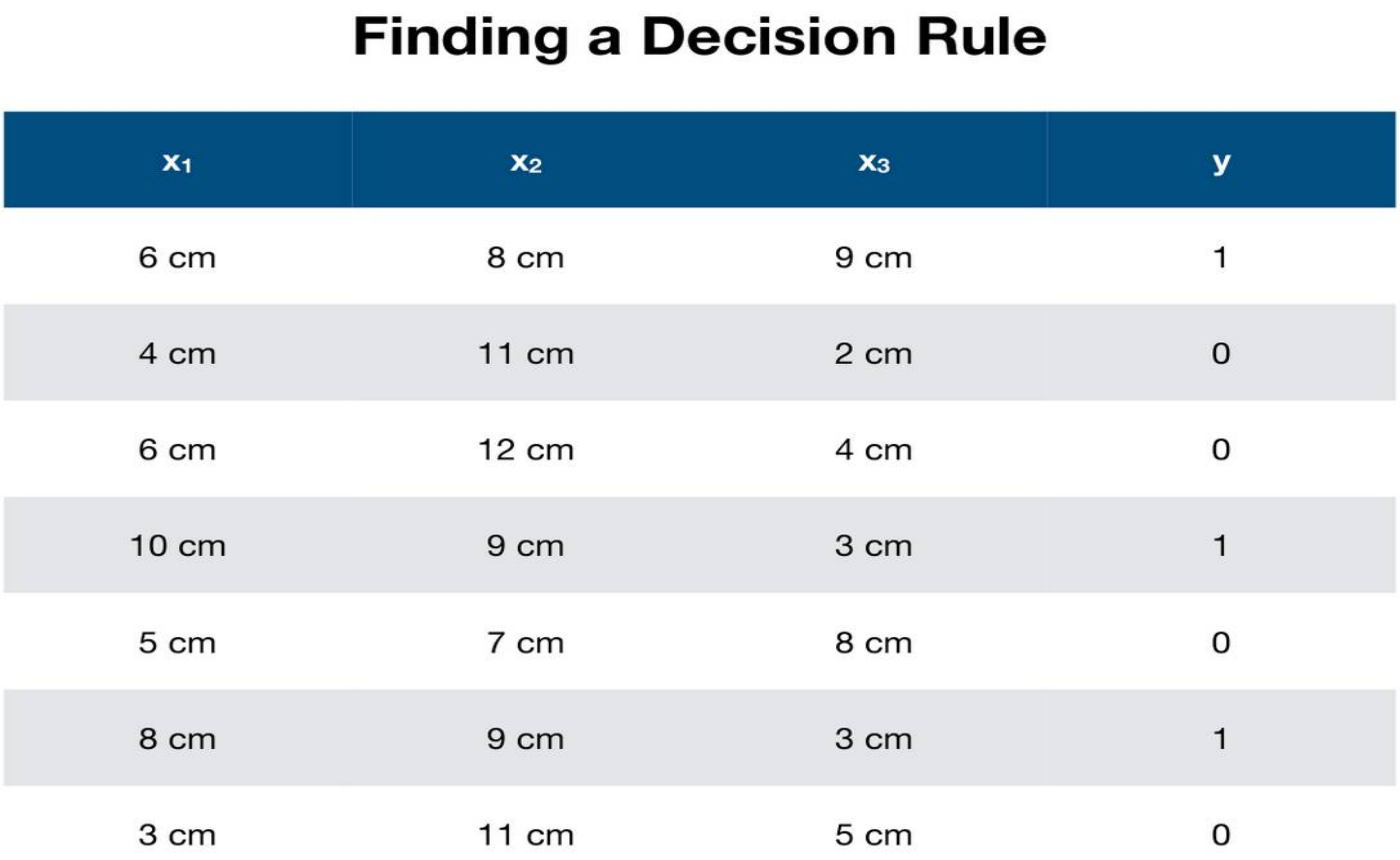
C4.5

- continuous and discrete features
- Ross Quinlan 1993, Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, *38*, 48.
- continuous is very expensive, because must consider all possible ranges
- handles missing attributes (ignores them in gain compute)
- post-pruning (bottom-up pruning)
- Gain Ratio



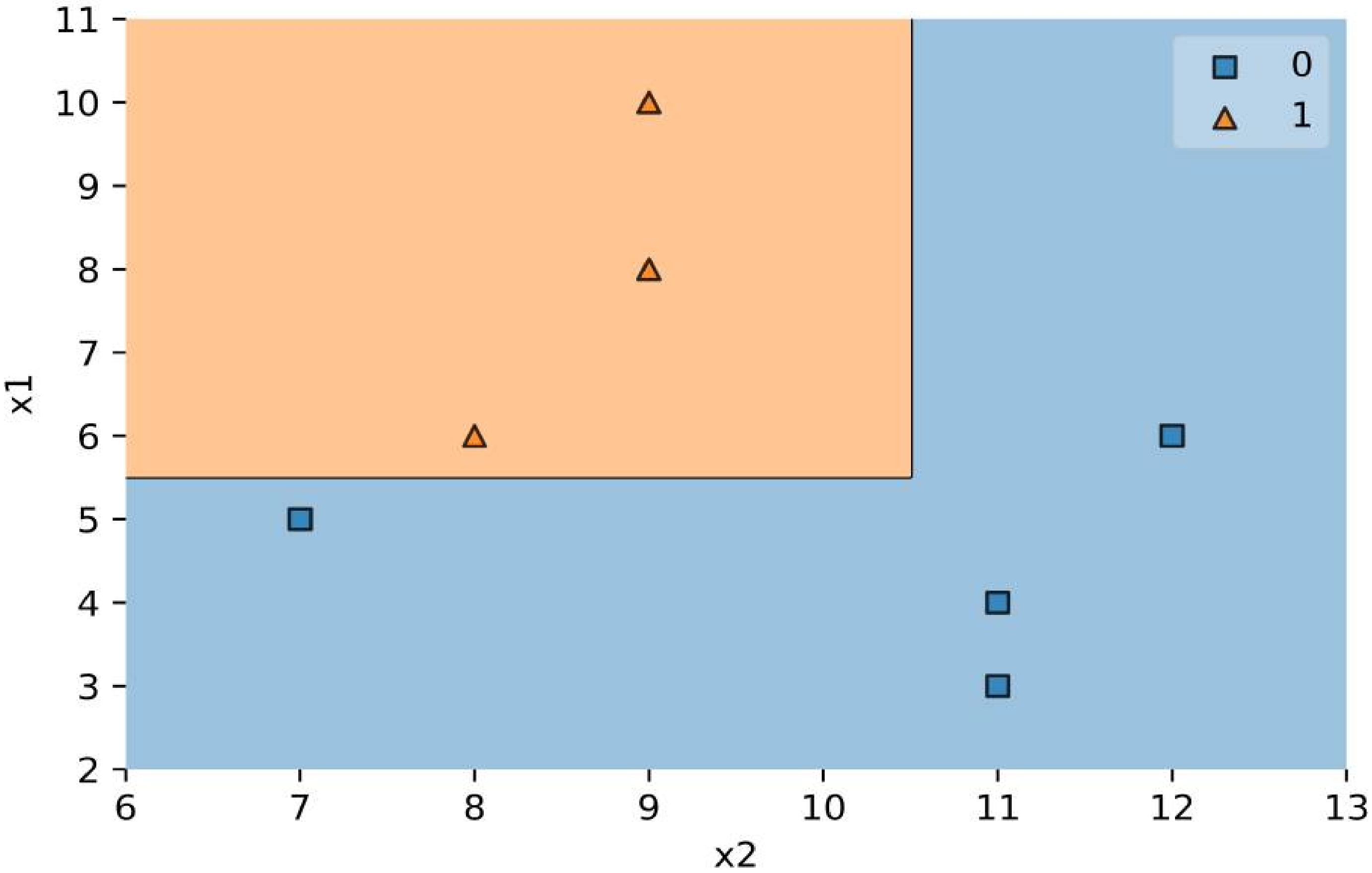
CART

- Breiman, L. (1984). *Classification and regression trees*. Belmont, Calif: Wadsworth International Group.
- continuous and discrete features
- strictly binary splits (taller trees than ID3, C4.5)
- binary splits can generate better trees than C4.5, but tend to be larger and harder to interpret; k-attributes has a ways to create a binary partitioning
- variance reduction in regression trees
- Gini impurity, twoing criteria in classification trees
- cost complexity pruning

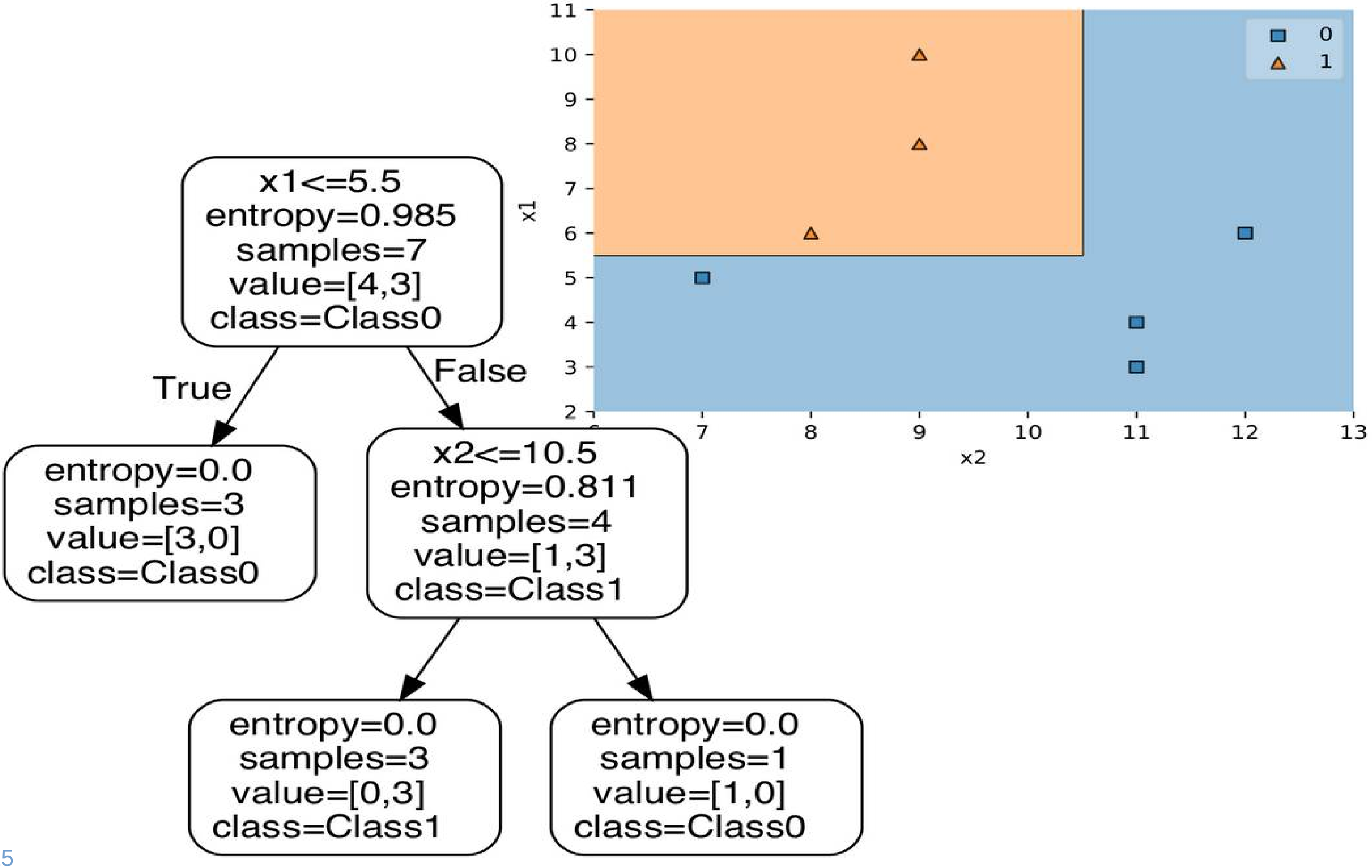


Finding a Decision Rule

x_1	x_2	x_3	y
6 cm	8 cm	9 cm	1
4 cm	11 cm	2 cm	0
6 cm	12 cm	4 cm	0
10 cm	9 cm	3 cm	1
5 cm	7 cm	8 cm	0
8 cm	9 cm	3 cm	1
3 cm	11 cm	5 cm	0



11
10
9
8
7
6
5
4
3
2
x1
6
7
8
9
10
11
12
13
x2
■ 0
▲ 1



x1<=5.5
entropy=0.985
samples=7
value=[4,3]
class=Class0

True

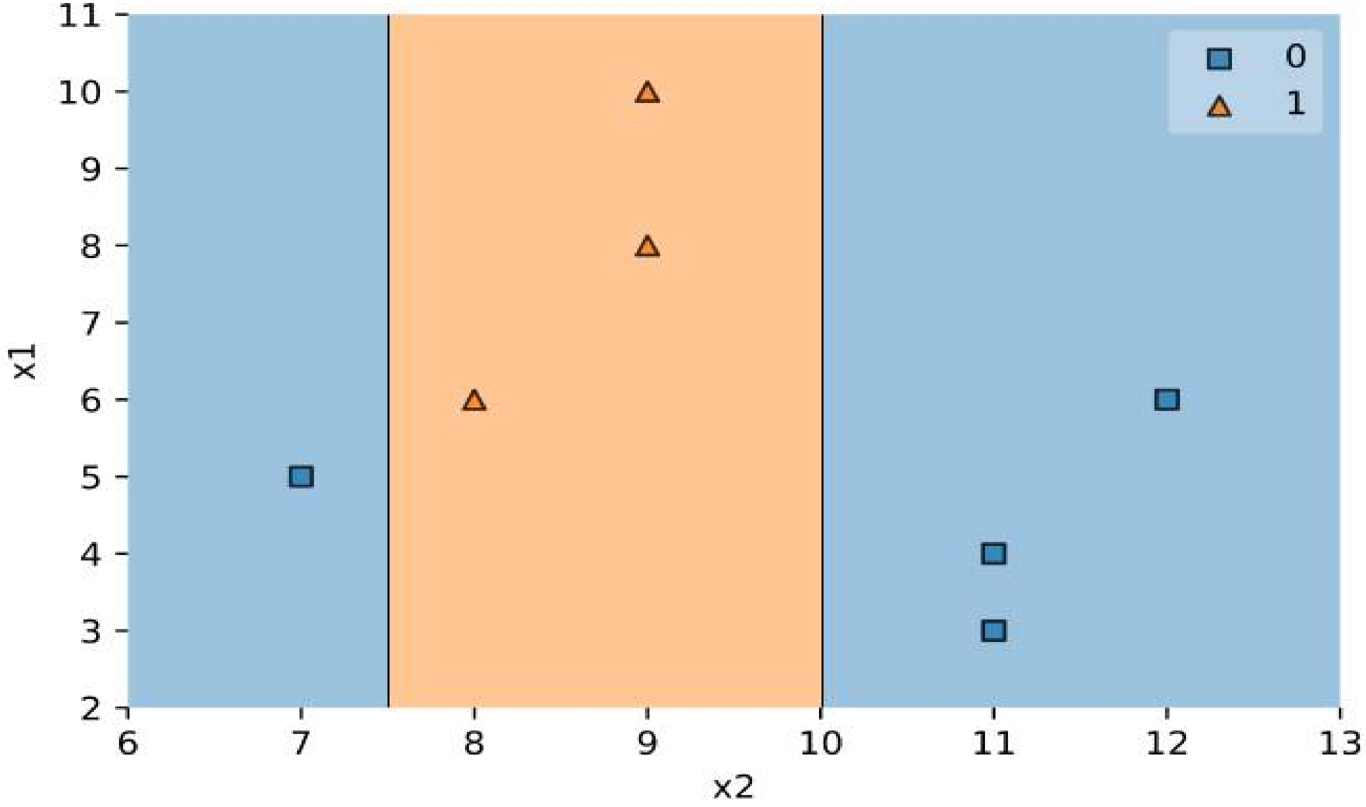
entropy=0.0
samples=3
value=[3,0]
class=Class0

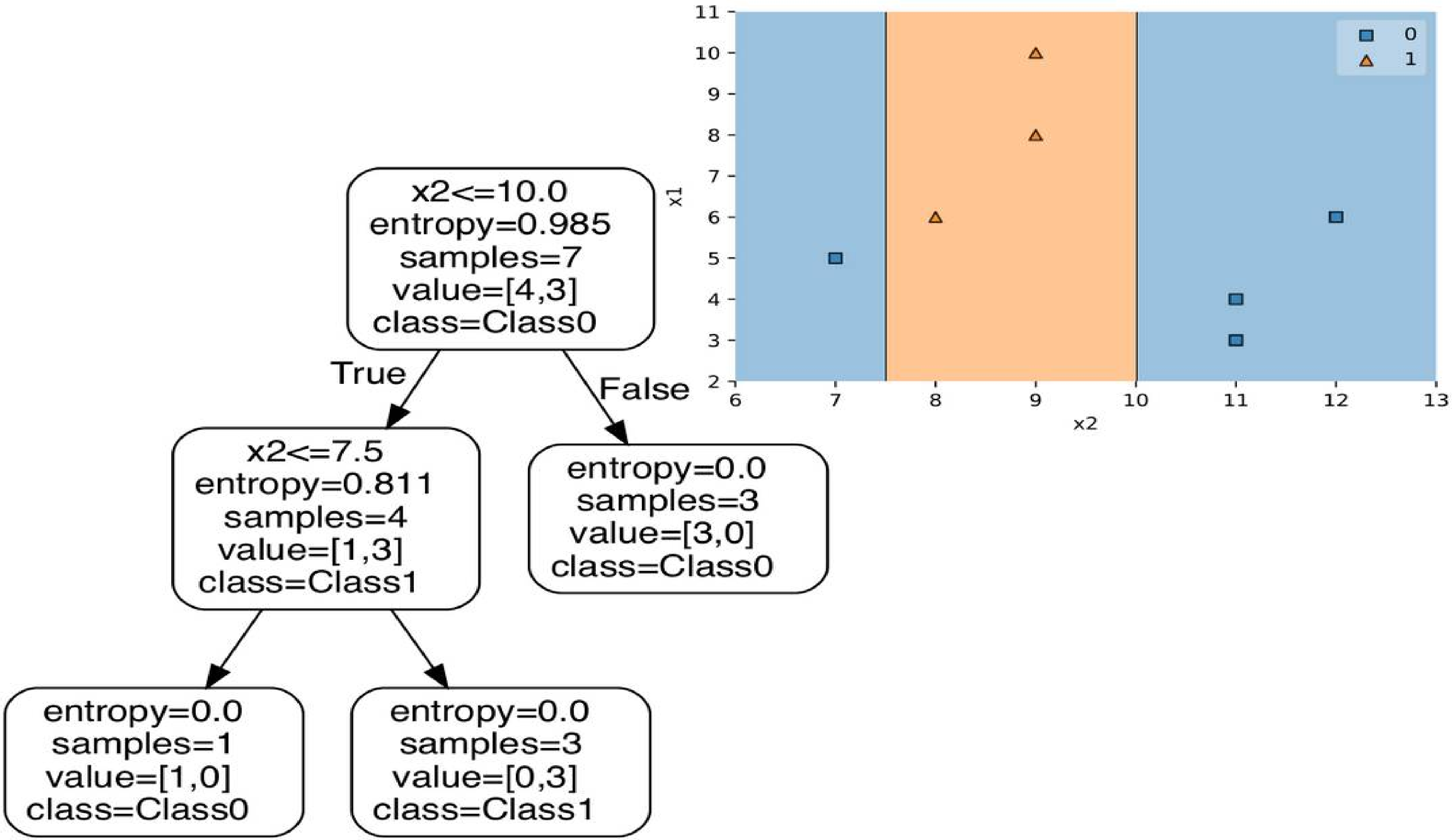
False

x2<=10.5
entropy=0.811
samples=4
value=[1,3]
class=Class1

entropy=0.0
samples=3
value=[0,3]
class=Class1

entropy=0.0
samples=1
value=[1,0]
class=Class0


0
1
x1
11
10
9
8
7
6
5
4
3
2
6
7
8
9
10
11
12
13
x2



x2<=10.0
entropy=0.985
samples=7
value=[4,3]
class=Class0

True

x2<=7.5
entropy=0.811
samples=4
value=[1,3]
class=Class1

entropy=0.0
samples=1
value=[1,0]
class=Class0

entropy=0.0
samples=3
value=[0,3]
class=Class1

False

entropy=0.0
samples=3
value=[3,0]
class=Class0

Information Gain

- The information gain, when attribute x_j is used to split the dataset S, is given as

$$Gain(S, x_j) = H(S) - \sum_{v \in Values(x_j)} \frac{|S_v|}{|S|} H(S_v)$$

- H(S) is an impurity function (Entropy for ID3, gini index for CART)
- An attribute with maximum information gain, is used for splitting.

Gini Impurity

- Let S be a dataset and p_k be the prob of a point belong to a class k. If total number of classes are K, then gini impurity can be written as

$$G(S) = \sum_{k=1}^{K} p_k (1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

- For two classes and p = 0, G=0
- For p = 1, G=0,
- For p = 0.5 , G is maximum = 0.5
- Plot shown in next slides.

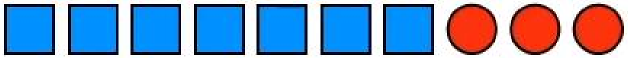
Entropy

- Let S be a dataset and p_k be the prob of a point belong to a class k. If total number of classes are K, then entropy can be written as

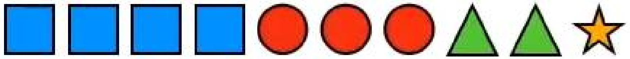
$$H(S) = -\sum_{k=1}^{K} p_k \log_2 p_k$$

Gini Impurity

- Measures the diversity in a dataset.
- Which of the following is more **diverse**?

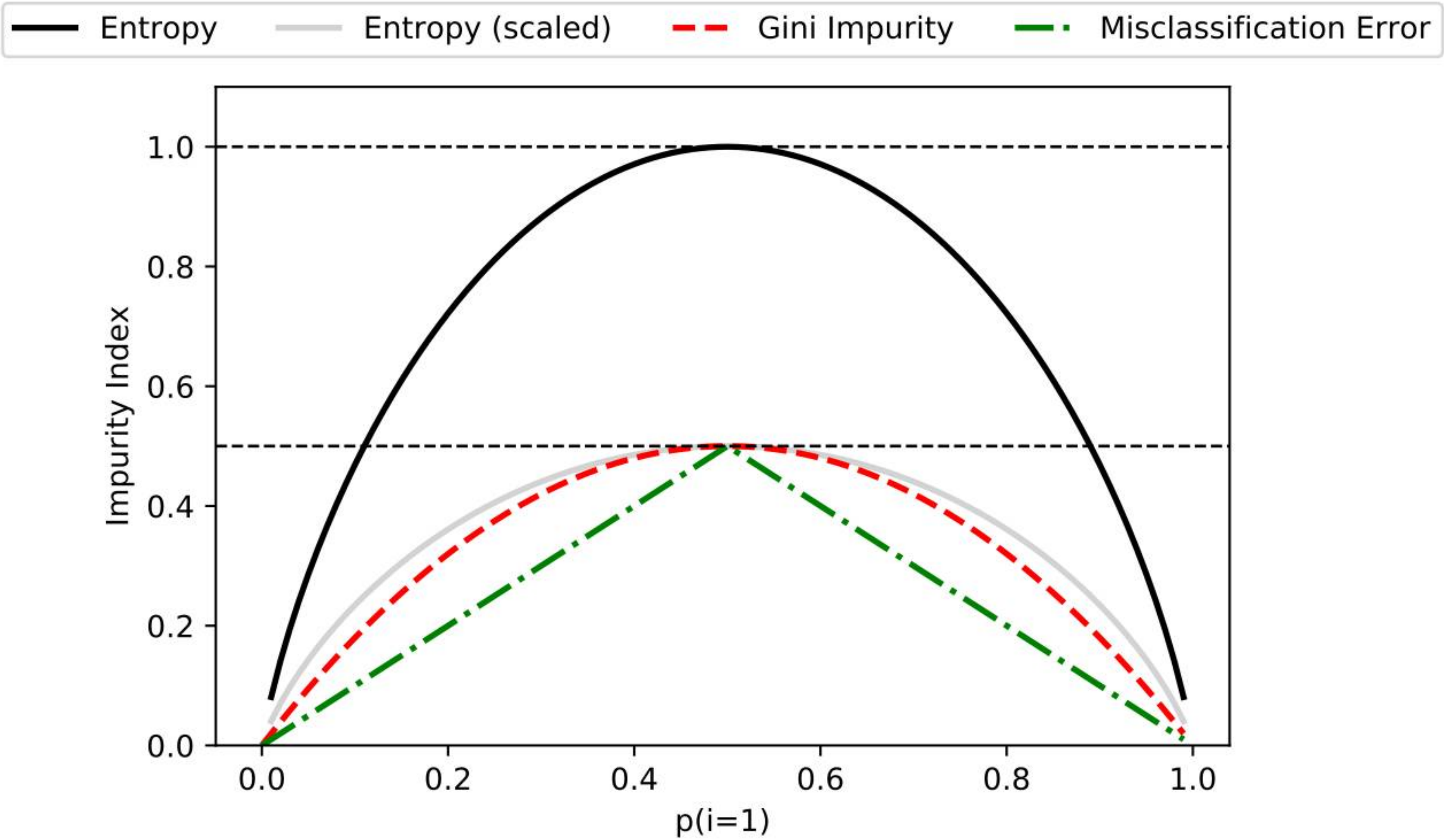


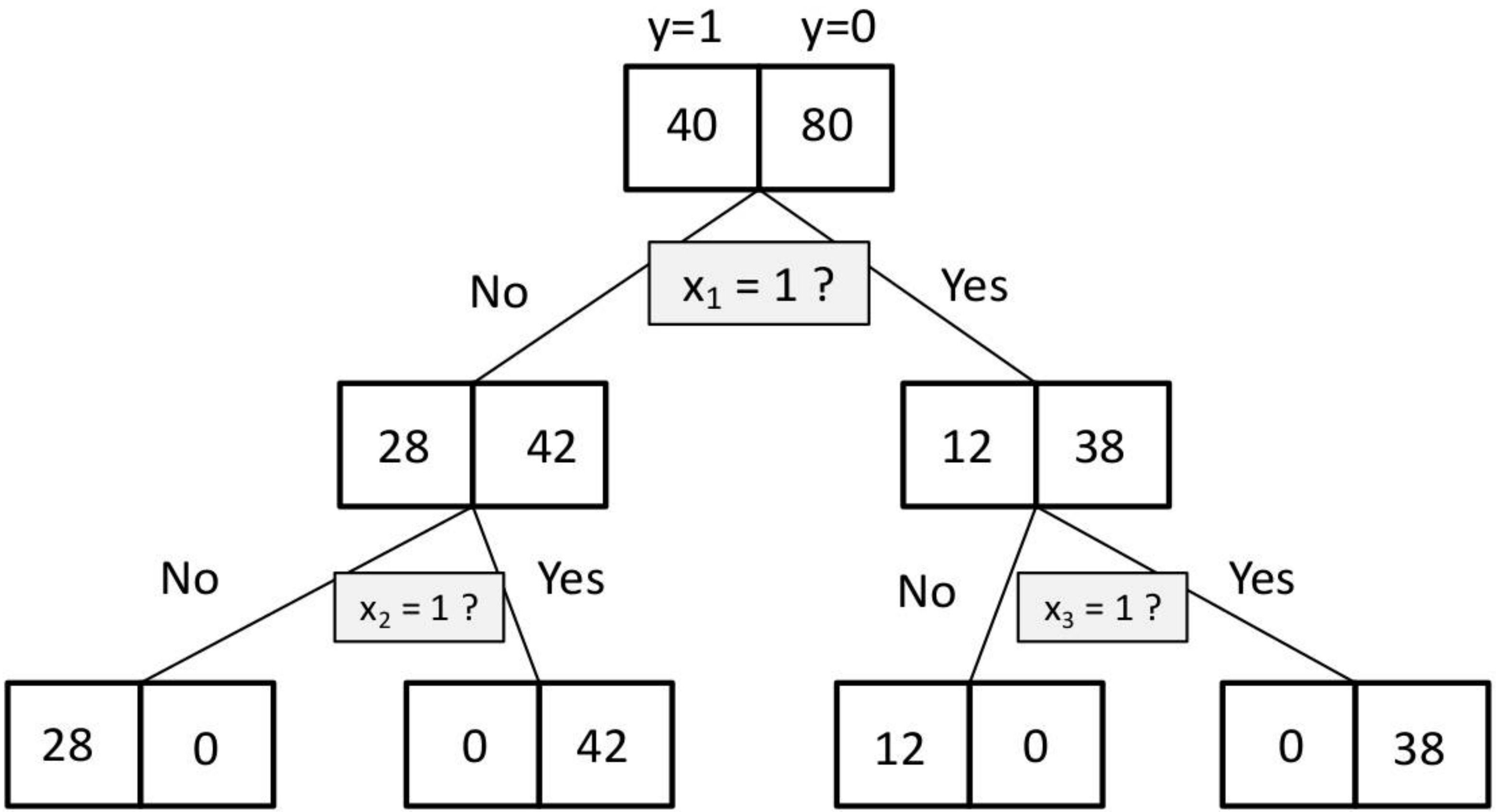
Gini = 0.42

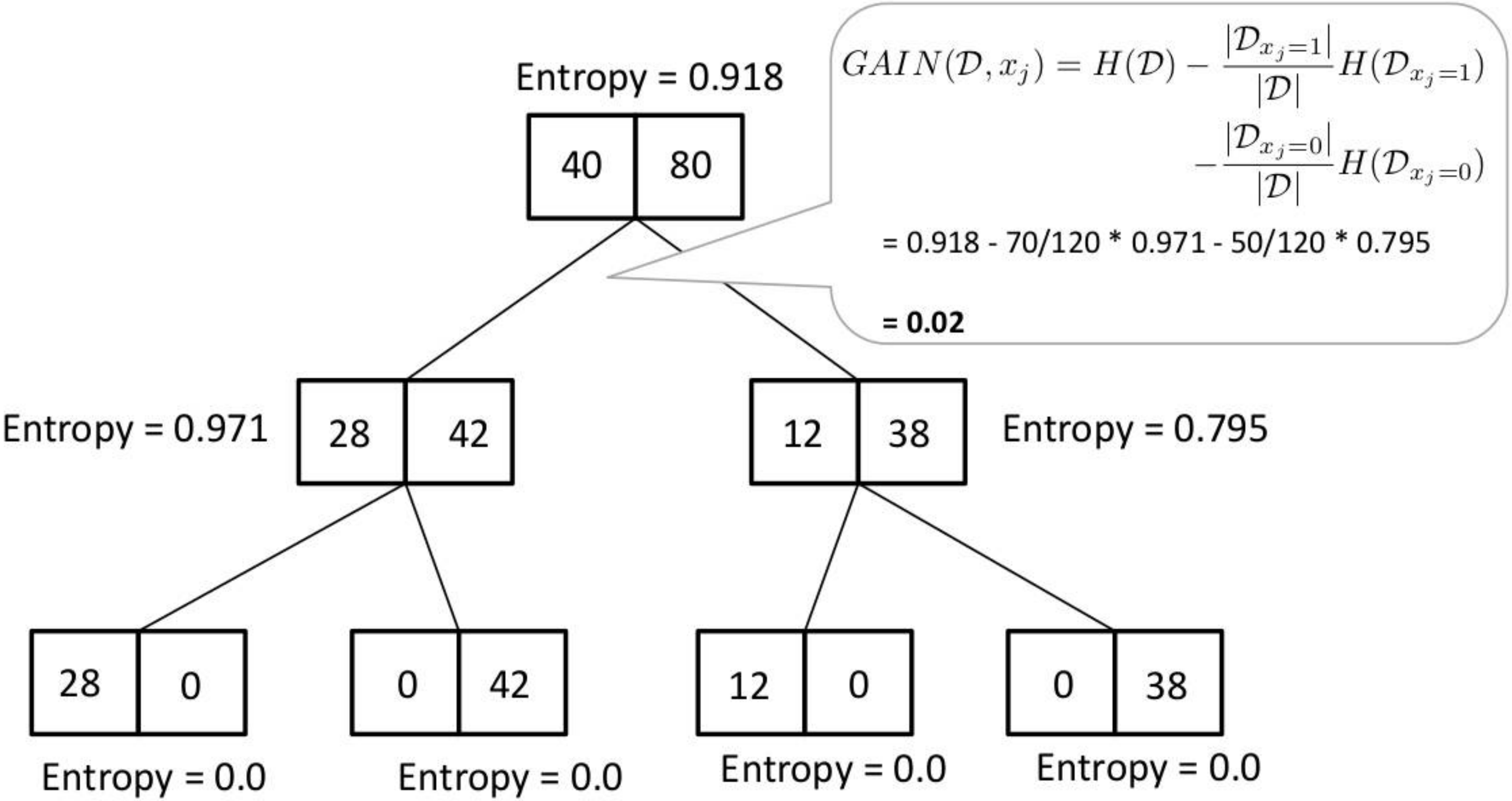


Gini = 0.7

- Right one is more diverse (more elements and more variety in them)
- GI will be higher for the set that is more diverse.


Entropy
Entropy (scaled)
Gini Impurity
Misclassification Error
Impurity Index
0.0
0.2
0.4
0.6
0.8
1.0
p(i=1)
0.0
0.2
0.4
0.6
0.8
1.0


y=1
y=0
40
80
x₁ = 1 ?
No
Yes
28
42
12
38
x₂ = 1 ?
No
Yes
x₃ = 1 ?
No
Yes
28
0
0
42
12
0
0
38

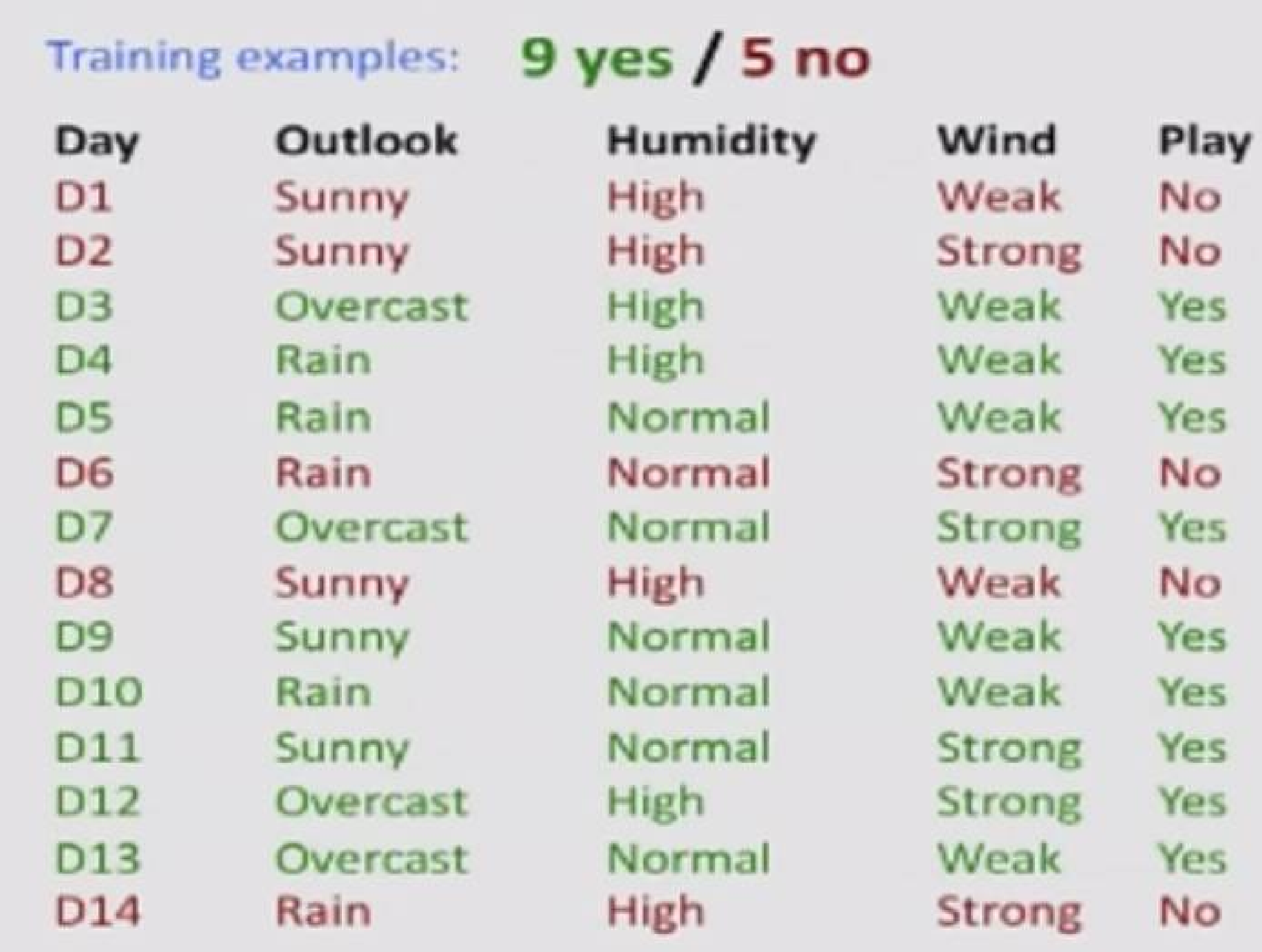


$$GAIN(\mathcal{D}, x_j) = H(\mathcal{D}) - \frac{|\mathcal{D}_{x_j=1}|}{|\mathcal{D}|} H(\mathcal{D}_{x_j=1}) - \frac{|\mathcal{D}_{x_j=0}|}{|\mathcal{D}|} H(\mathcal{D}_{x_j=0})$$

= 0.918 - 70/120 * 0.971 - 50/120 * 0.795

= 0.02

Predict if John will play tennis



Training examples: 9 yes / 5 no

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Source: Victor Lavrenko

Example: Use ID3 for creating the tree

- Done in class

References

- STAT 451: Intro to Machine Learning, Fall 2020 Sebastian Raschka
  - http://stat.wisc.edu/~sraschka/teaching/stat451-fs2020/
- Cornell CS4780 Lecture 29: Kilian Weinberger
- https://www.cs.cornell.edu/courses/cs4780/2021fa/lectures/lecturenote17.html
-

End of Lecture