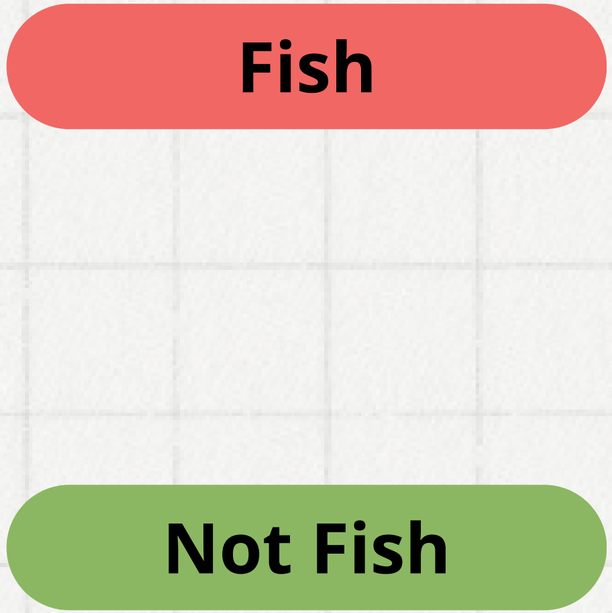
Decision Trees

AI / ML Workshop
NCIT College

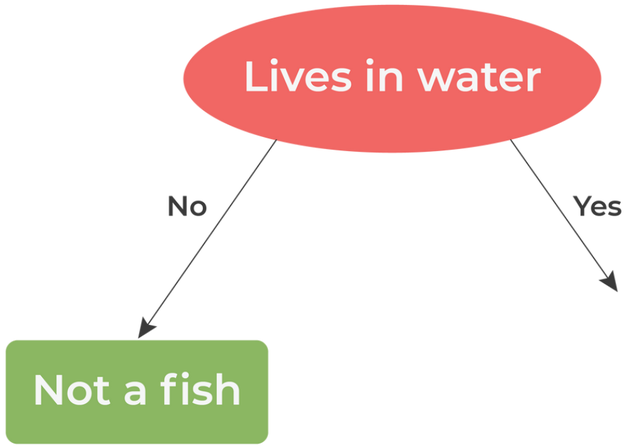
Intuition

- Tree of Decisions (**i.e. Every Node represents some decisions)**
- Divide and Conquer Approach
- Supervised algorithm for both classification and regression

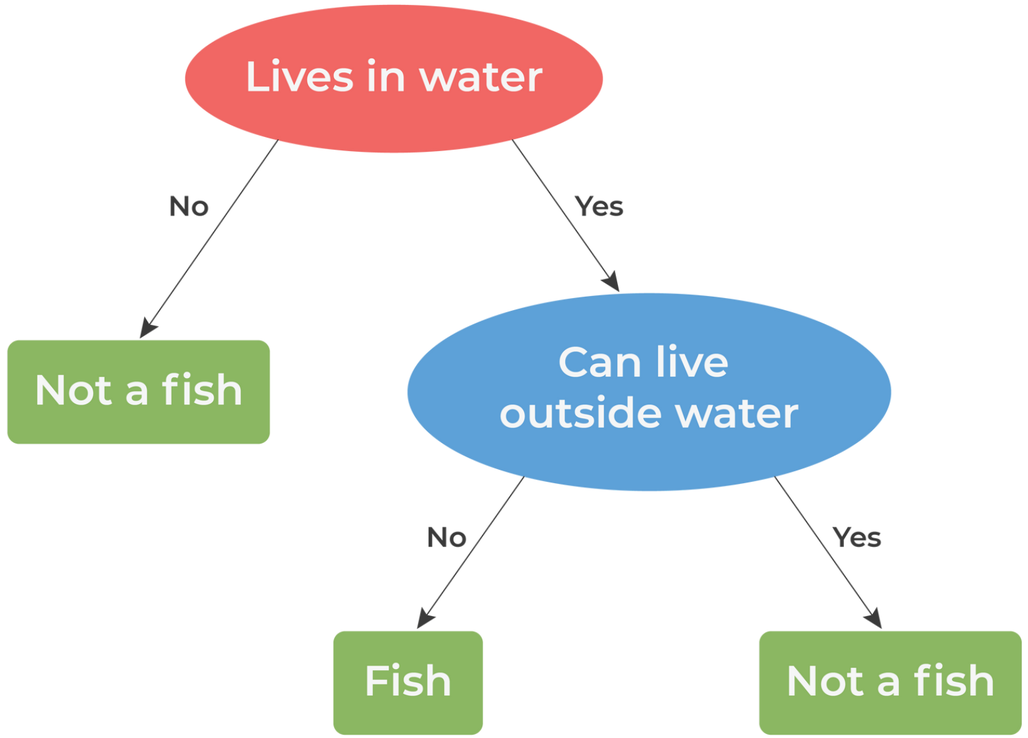



Lives in water

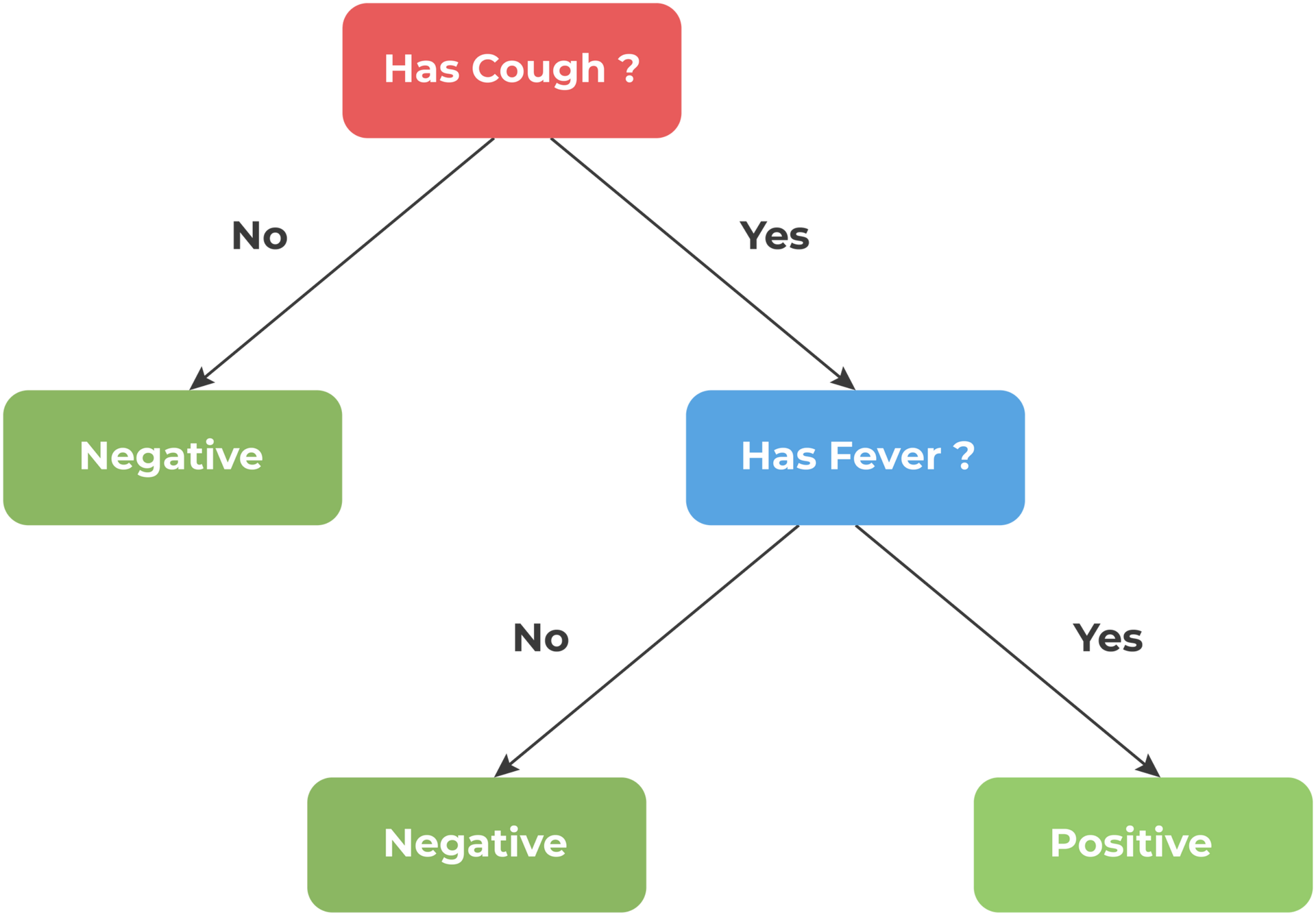
a


Lives in water
No
Yes
Not a fish

b


Lives in water
No
Yes
Not a fish
Can live
outside water
No
Yes
Fish
Not a fish

c


Has Cough ?
No
Yes
Negative
Has Fever ?
No
Yes
Negative
Positive

How Do we Split?

Impurity Metrics

Gini

$$\text{Gini} = 1 - \sum_{i=1}^{n} P_i^2$$

- Probability of Misclassifying a Data point

Entropy

It is a measure of disorder or randomness in data.

$$\text{Entropy} = -\sum_{i=1}^{c} P_i \log P_i$$

Here,

c = Total number of labels/classes.

p_i = Probability of an item belonging to class i.

Information Gain

Information gain is the decrease in entropy after making a split.

$$\text{Information Gain}(G) = \text{Entropy}(L) - \text{Entropy}(L, A)$$

where,

$\text{Entropy}(L)$ = Entropy of parent node

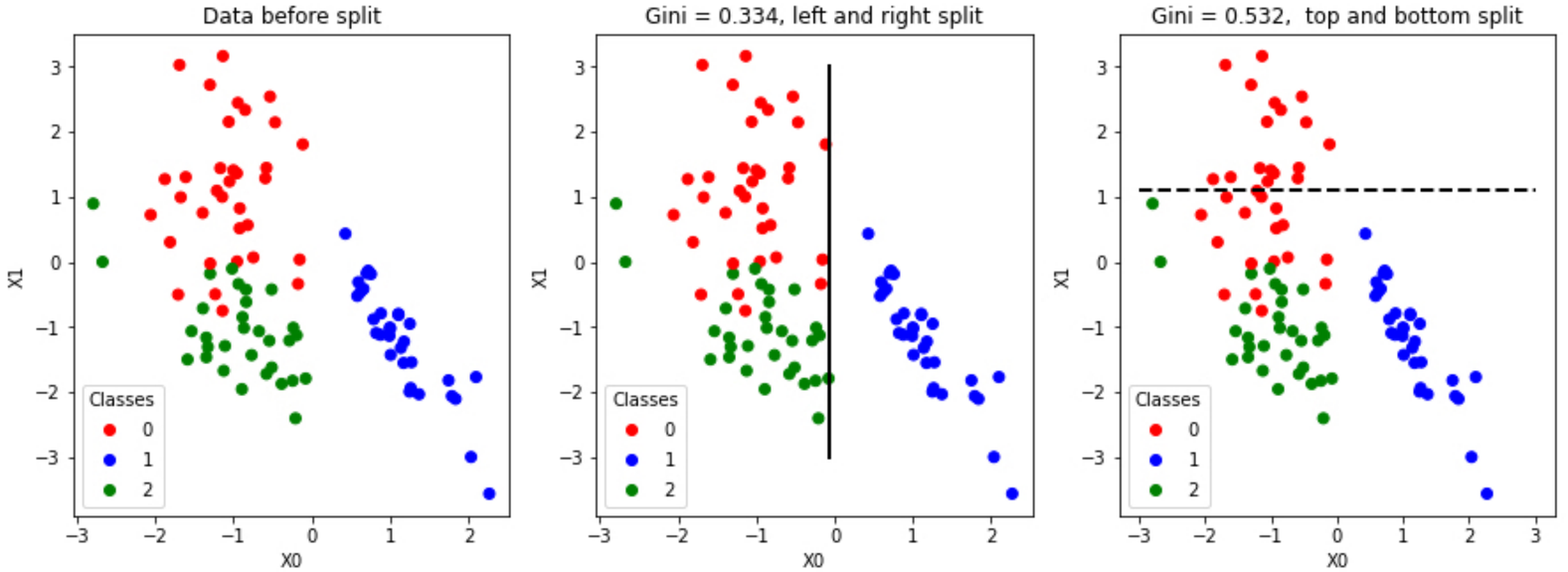
$\text{Entropy}(L, A)$ = Entropy of child nodes after splitting parent node using attribute A

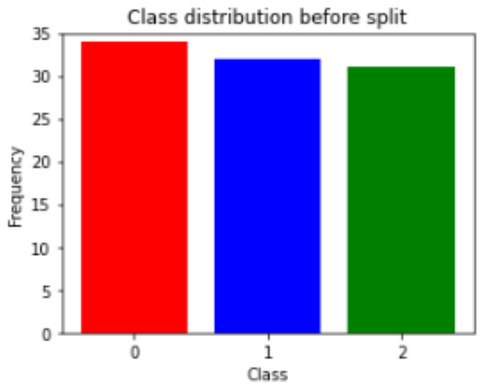
L = Label of the dataset

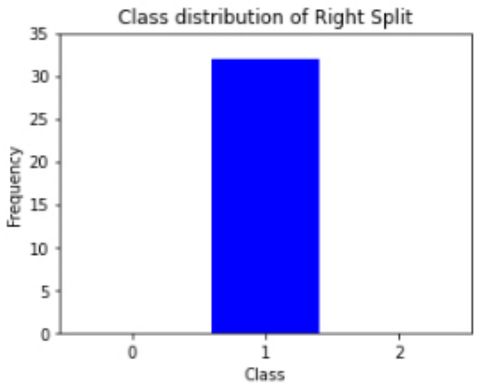
Gain Ratio

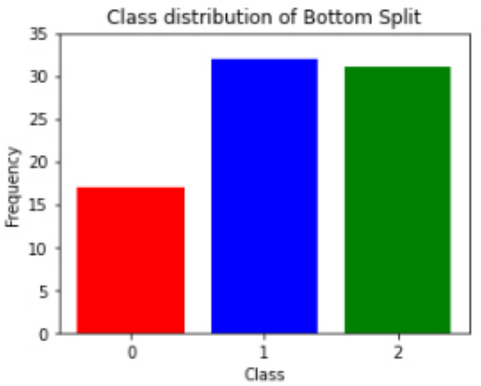
Mostly Used in Decision Tree Problems

$$\text{Gain Ratio}(\text{L, A}) = \frac{Information\ Gain(G)}{Split\ Entropy(L, A)}$$

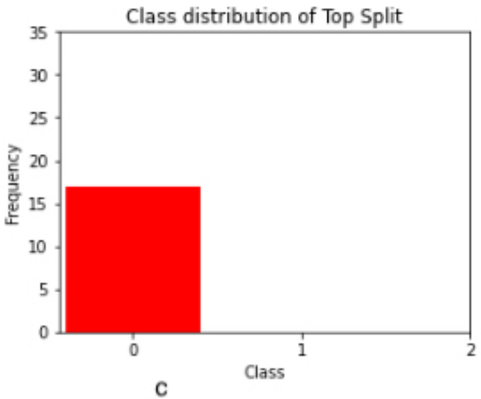

Data before split
Gini = 0.334, left and right split
Gini = 0.532, top and bottom split
X0
X1
Classes
0
1
2


Class distribution before split
Frequency
Class


Class distribution of Right Split
Frequency
Class


Class distribution of Bottom Split
Frequency
Class


Class distribution of Left Split
Frequency
Class

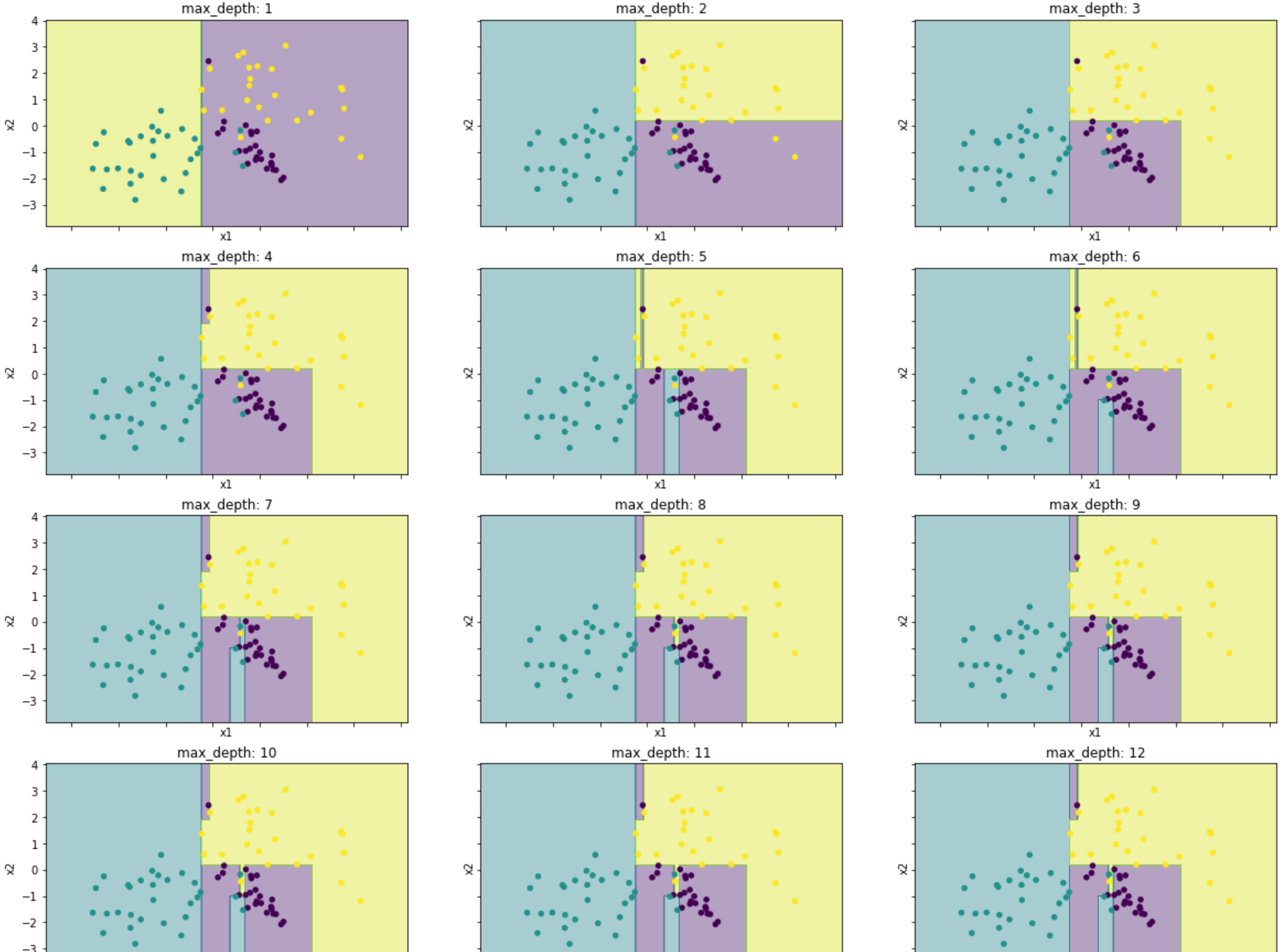

Class distribution of Top Split
Frequency
Class

a b c

Algorithm

- If
  - all records belong to same class. Label as leaf Node
- else
  - Calculate **Impurity metrics** for each attribute i.e **Gini, Entropy, MSE**
  - Select the **best attribute**
  - Split data into subsets
  - Repeat

Should We Worry about Overfitting the Data?


max_depth: 1
max_depth: 2
max_depth: 3
max_depth: 4
max_depth: 5
max_depth: 6
max_depth: 7
max_depth: 8
max_depth: 9
max_depth: 10
max_depth: 11
max_depth: 12
x1
x2

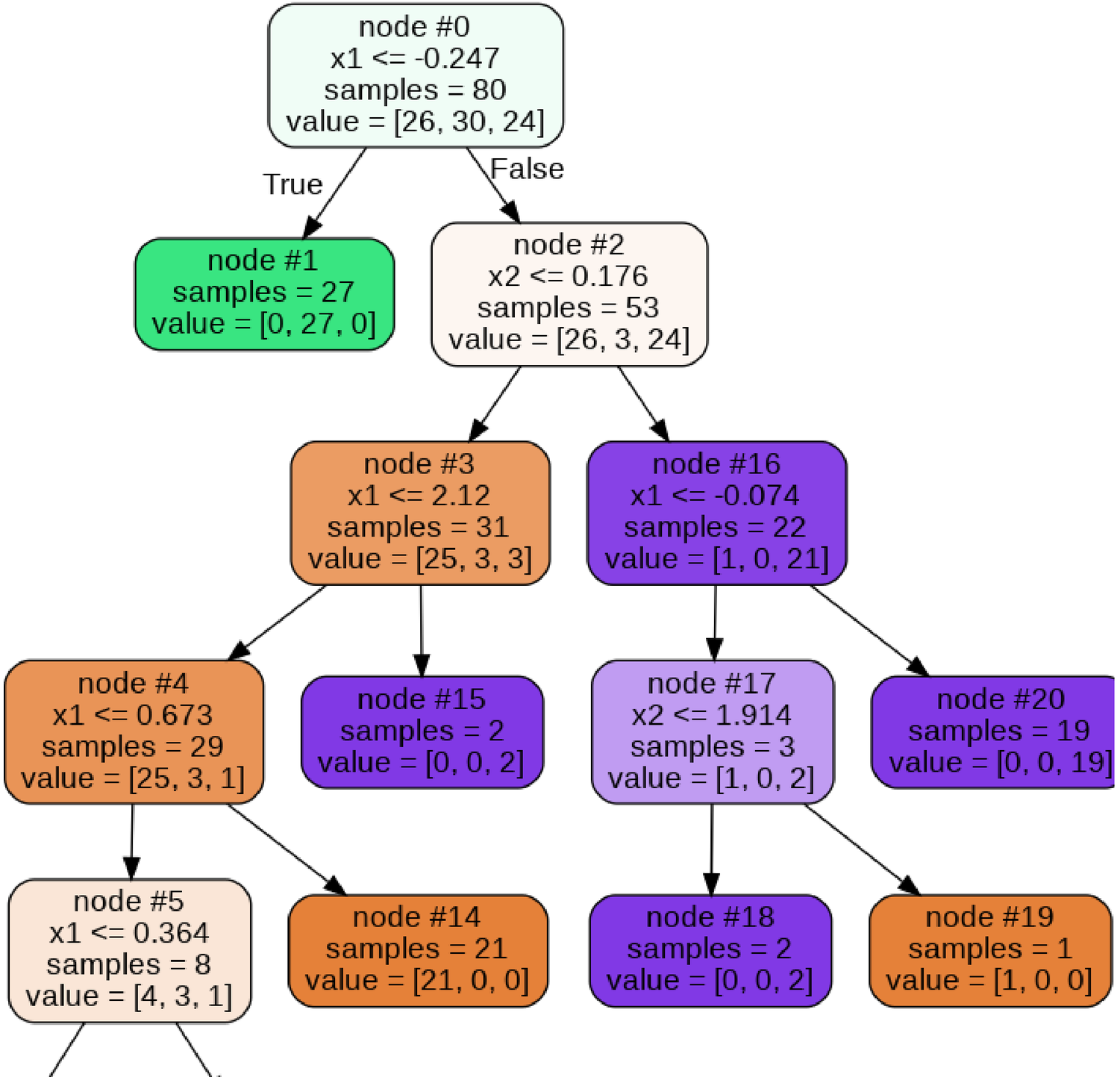
OCCAM'S RAZOR PRINCIPLE

If there exist multiple solutions to a problem, the simplest one is usually the best.

TECHNIQUES TO OVERCOME

Early Stopping

Pruning


node #0
x1 <= -0.247
samples = 80
value = [26, 30, 24]
True
False
node #1
samples = 27
value = [0, 27, 0]
node #2
x2 <= 0.176
samples = 53
value = [26, 3, 24]
node #3
x1 <= 2.12
samples = 31
value = [25, 3, 3]
node #16
x1 <= -0.074
samples = 22
value = [1, 0, 21]
node #4
x1 <= 0.673
samples = 29
value = [25, 3, 1]
node #15
samples = 2
value = [0, 0, 2]
node #17
x2 <= 1.914
samples = 3
value = [1, 0, 2]
node #20
samples = 19
value = [0, 0, 19]
node #5
x1 <= 0.364
samples = 8
value = [4, 3, 1]
node #14
samples = 21
value = [21, 0, 0]
node #18
samples = 2
value = [0, 0, 2]
node #19
samples = 1
value = [1, 0, 0]

Thank you very much!