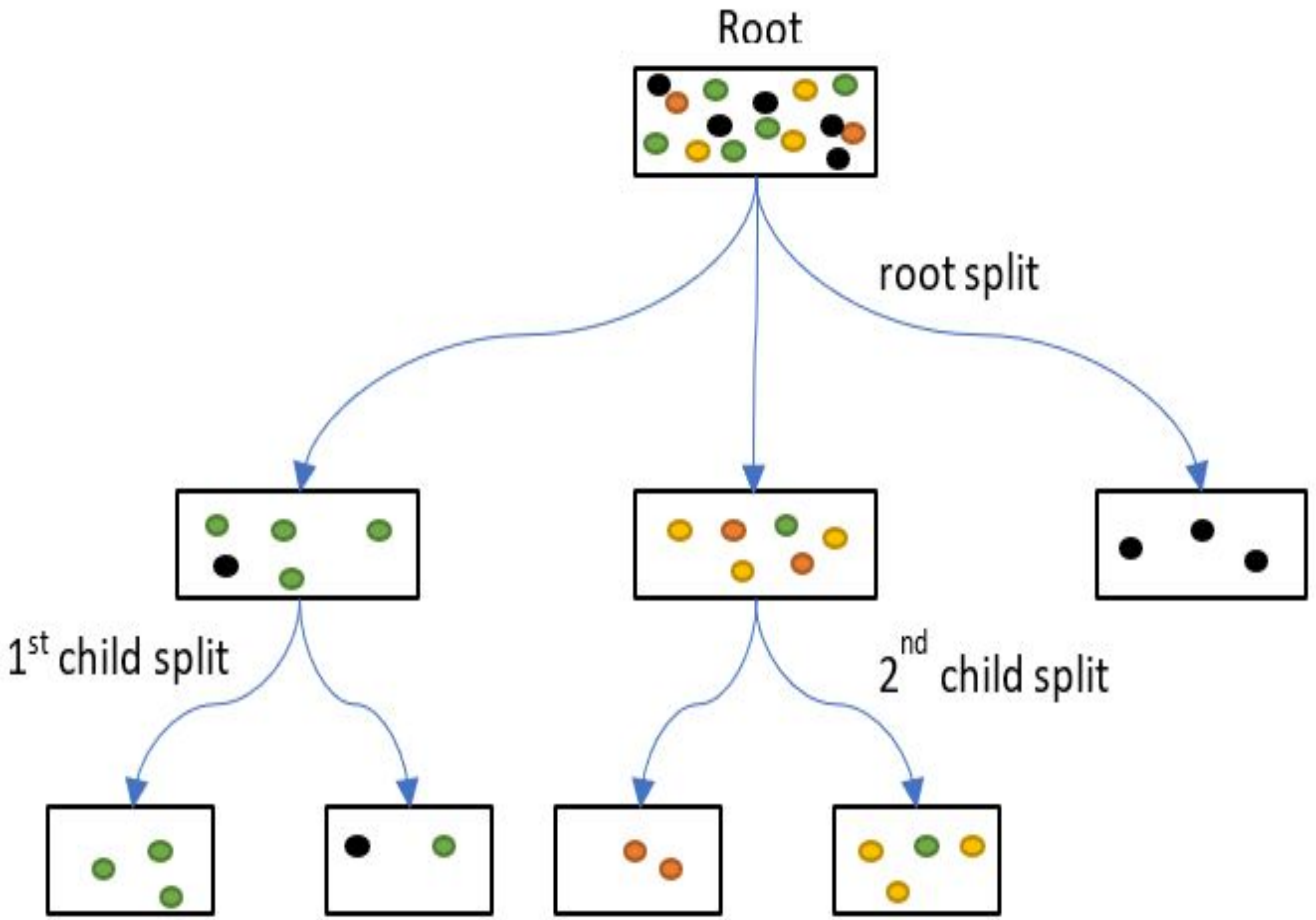
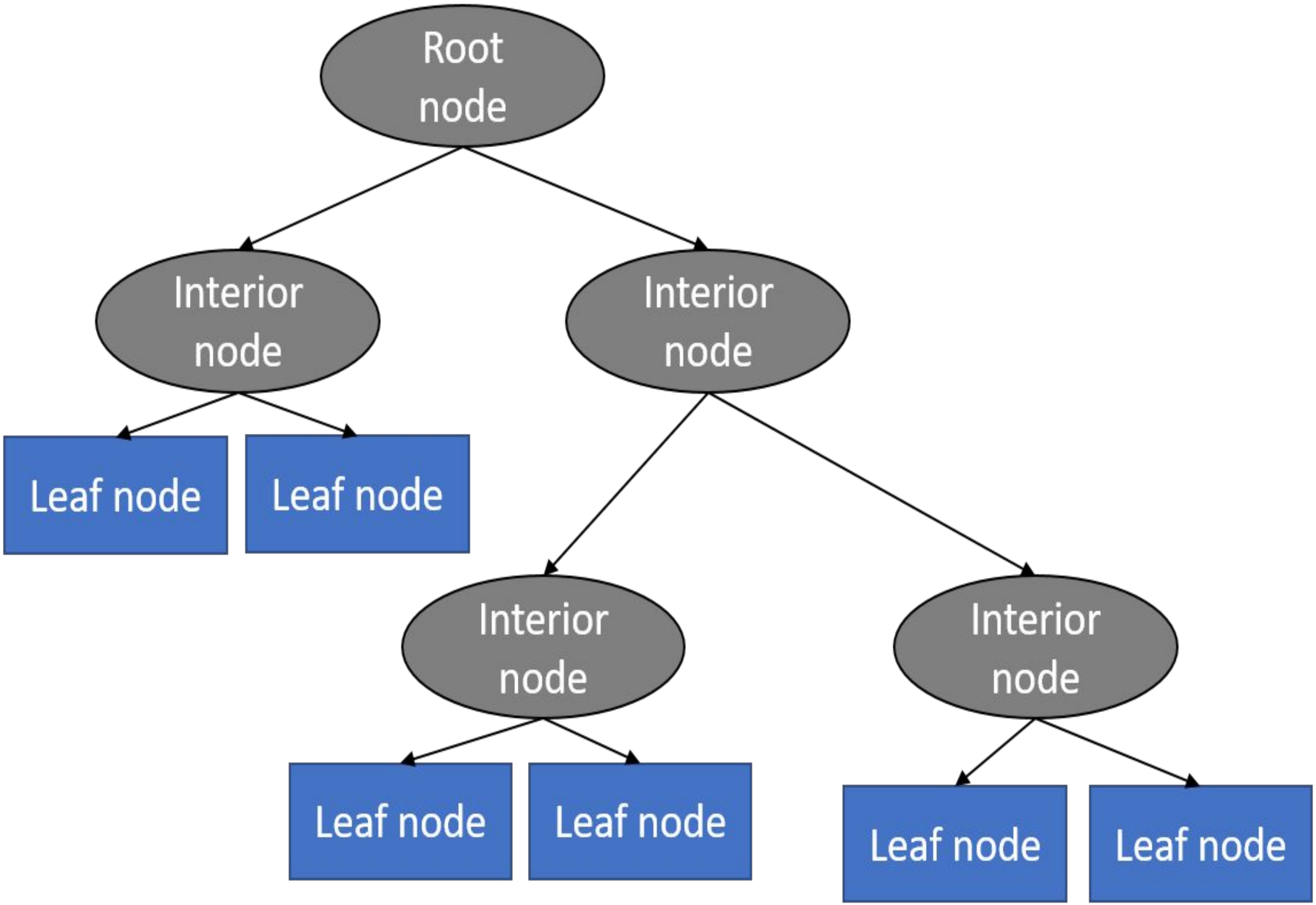
Decision Tree Algorithm

A decision tree is built top-down from a root node

- **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
- **Leaf/ Terminal Node:** Nodes with no children (no further split) is called Leaf or Terminal node.
- **Branch / Sub-Tree:** A sub section of decision tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent nod


Root
root split
1st child split
2nd child split


Root node
Interior node
Interior node
Leaf node
Leaf node
Interior node
Interior node
Leaf node
Leaf node
Leaf node
Leaf node

- Decision Tree algorithm belongs to the family of supervised learning algorithms

- Can be used for solving **regression and classification problems**

- The idea of the Decision Tree is to divide the data into smaller datasets based on a certain feature value until the target variables all fall under one category

- While the human brain decides to pick the "splitting feature" based on the experience, algorithm splits the dataset based on the **maximum *information gain Entropy and by using Ginni Index***

Using Gini Split / Gini Index

- Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.
- The degree of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes.
- A Gini Index of 0.5 denotes equally distributed elements into some classes.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

where p_i is the probability of an object being classified to a particular class.

- While building the decision tree, we would prefer choosing the attribute/feature with the least Gini index as the root node.

- Calculated after each split
- To calculate Gini :
    - sum the square of probability of finding each class after a node
    - subtract this amount from 1
- For this reason, when a subset is pure (i.e. there is only one class in it),
  Gini will be 0, because the probability of finding that class is 1
- And in that case, we say we have reached a And in that case, we say we have reached a ***leaf***
- It means an attribute with lower Gini index should be preferred.

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

Let’s start by calculating the Gini Index for ‘Past Trend’.
P(Past Trend=Positive): 6/10
P(Past Trend=Negative): 4/10

If (Past Trend = Positive & Return = Up), probability = 4/6
If (Past Trend = Positive & Return = Down), probability = 2/6
Gini index = 1 - ((4/6)^2 + (2/6)^2) = 0.45

If (Past Trend = Negative & Return = Up), probability = 0
If (Past Trend = Negative & Return = Down), probability = 4/4
Gini index = 1 - ((0)^2 + (4/4)^2) = 0

Weighted sum of the Gini Indices can be calculated as follows:
Gini Index for Past Trend = (6/10)*0.45 + (4/10)*0 = 0.27

Similarly calclating Gini Index for Open Interest, Trading Volume

Gini Index for Open Interest = (4/10)*0.5 + (6/10)*0.45 = 0.47
Gini Index for Trading Volume = (7/10)*0.49 + (3/10)*0 = 0.34

- we observe that ‘Past Trend’ has the lowest Gini Index and hence it will be chosen as the root node

- calculate the Gini Index for the ‘Positive’ branch of Past Trend as follows

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Positive	Low	High	Up
Positive	High	High	Up
Positive	Low	Low	Down
Positive	Low	Low	Down
Positive	High	High	Up

Calculation of Gini Index of Open Interest for Positive Past Trend

P(Open Interest=High): 2/6
P(Open Interest=Low): 4/6

If (Open Interest = High & Return = Up), probability = 2/2
If (Open Interest = High & Return = Down), probability = 0
Gini index = 1 - (sq(2/2) + sq(0)) = 0

If (Open Interest = Low & Return = Up), probability = 2/4
If (Open Interest = Low & Return = Down), probability = 2/4
Gini index = 1 - (sq(0) + sq(2/4)) = 0.50

Weighted sum of the Gini Indices can be calculated as follows:

Gini Index for Open Interest = (2/6)*0* + *(4/6)*0.50 = **0.33**

Calculation of Gini Index for Trading Volume
Gini Index for Trading Volume = (4/6)*0* + *(2/6)*0 = **0**
We will split the node further using the 'Trading Volume' feature, as it has the minimum Gini index.

Splitting with Information Gain and Entropy

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

- ***'p'*** denotes the probability and E(S) denotes the entropy. Entropy is not preferred due to the 'log' function as it increases the computational complexity.
- Weights probability of class by log(base=2) of the class probability
- Entropy is measured between 0 and 1.

is Information Gain?

Information Gain is used to determine which feature/attribute gives us the maximum information about a class. It is based on the concept of entropy, which is the degree of uncertainty, impurity

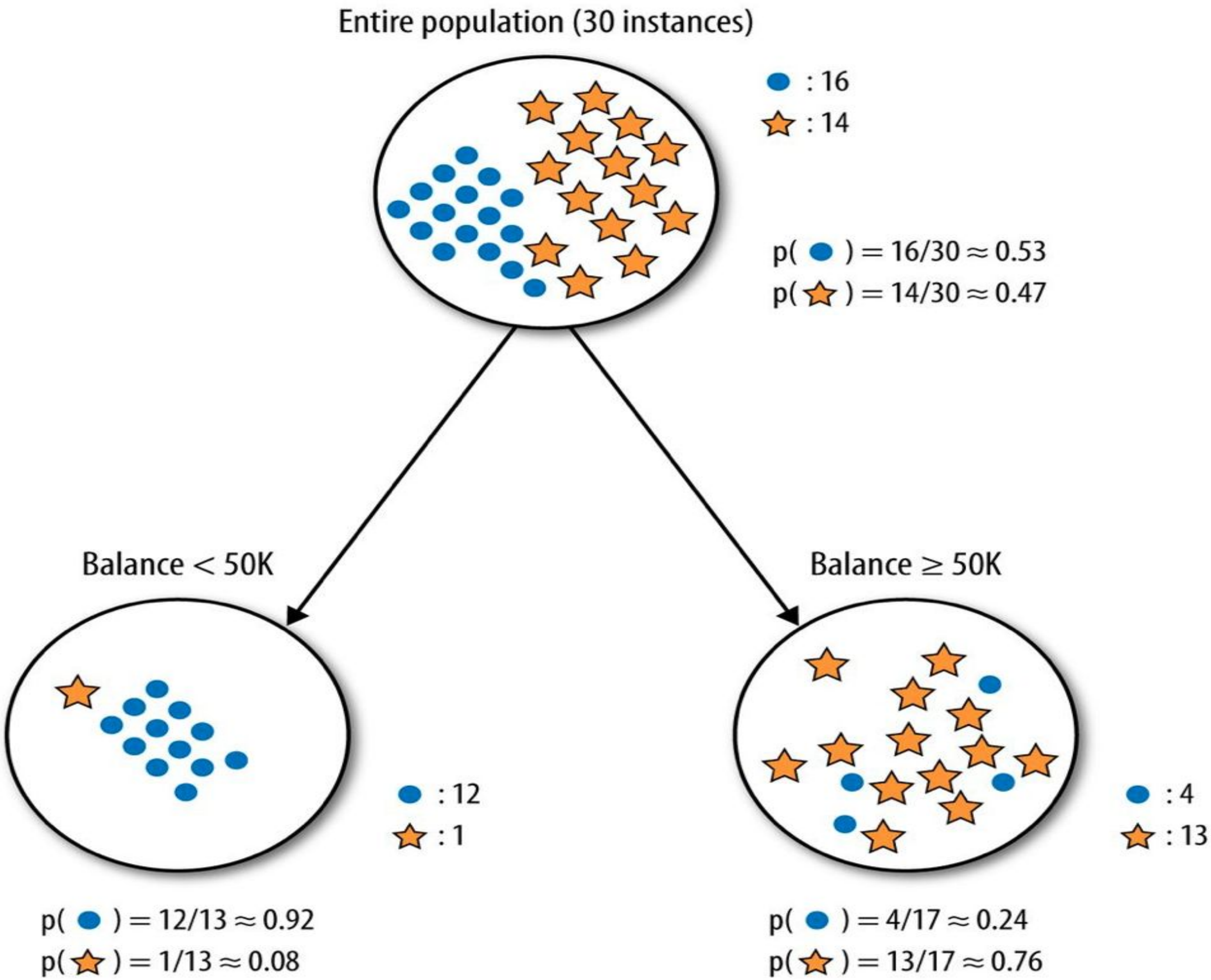
Information Gain = Entropy Before – Entropy after

Feature with the *largest information gain* is chosen for the split in a greedy manner

$$IG(Y, X) = E(Y) - E(Y|X)$$

Information Gain from X on Y

- Information gain tells us how important a given attribute of the feature vectors is



Entire population (30 instances)
: 16
: 14
p( ) = 16/30 ≈ 0.53
p( ) = 14/30 ≈ 0.47
Balance < 50K
: 12
: 1
p( ) = 12/13 ≈ 0.92
p( ) = 1/13 ≈ 0.08
Balance ≥ 50K
: 4
: 13
p( ) = 4/17 ≈ 0.24
p( ) = 13/17 ≈ 0.76

$$E(Parent) = -\frac{16}{30}\log_2\left(\frac{16}{30}\right) - \frac{14}{30}\log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Balance < 50K) = -\frac{12}{13}\log_2\left(\frac{12}{13}\right) - \frac{1}{13}\log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(Balance > 50K) = -\frac{4}{17}\log_2\left(\frac{4}{17}\right) - \frac{13}{17}\log_2\left(\frac{13}{17}\right) \approx 0.79$$

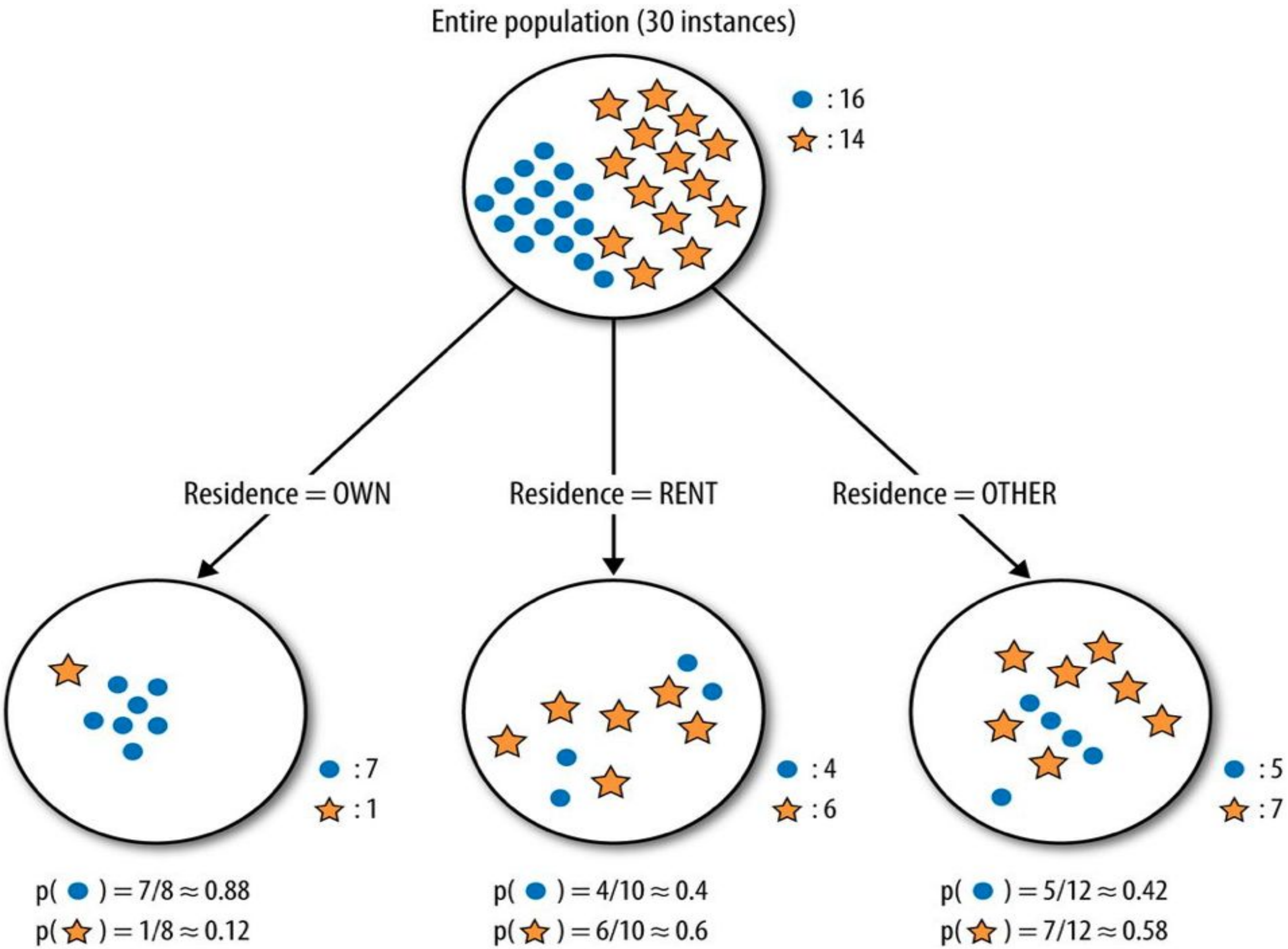
$Weighted\ Average\ of\ entropy\ for\ each\ node:$

$$\begin{aligned} E(Balance) &= \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79 \\ &= 0.62 \end{aligned}$$

$Information\ Gain:$

$$\begin{aligned} IG(Parent, Balance) &= E(Parent) - E(Balance) \\ &= 0.99 - 0.62 \\ &= 0.37 \end{aligned}$$

Residen



Entire population (30 instances)
: 16
: 14
Residence = OWN
Residence = RENT
Residence = OTHER
: 7
: 1
: 4
: 6
: 5
: 7
p( ) = 7/8 ≈ 0.88
p( ) = 1/8 ≈ 0.12
p( ) = 4/10 ≈ 0.4
p( ) = 6/10 ≈ 0.6
p( ) = 5/12 ≈ 0.42
p( ) = 7/12 ≈ 0.58

$$E(\,Residence\,=\,OWN)\;=\;-\;\frac{7}{8}\log_2\!\left(\frac{7}{8}\right)-\;\frac{1}{8}\log_2\!\left(\frac{1}{8}\right)\;\approx 0.54$$

$$E(\,Residence\,=\,RENT\,)\;=\;-\;\frac{4}{10}\log_2\!\left(\frac{4}{10}\right)-\;\frac{6}{10}\log_2\!\left(\frac{6}{10}\right)\;\approx 0.97$$

$$E(\,Residence\,=\,OTHER)\;=\;-\;\frac{5}{12}\log_2\!\left(\frac{5}{12}\right)-\;\frac{7}{12}\log_2\!\left(\frac{7}{12}\right)\;\approx 0.98$$

Weighted Average of entropies for each node:

$$E(\,Residence)\;=\frac{8}{30}\times\,0.54\;+\;\frac{10}{30}\times 0.97\;+\frac{12}{30}\times 0.98 = 0.86$$

Information Gain:

$$\begin{aligned}IG(\,Parent,\;Residence)\;&=E(\,Parent)\;-\;E(\,Residence)\\&=0.99-\;0.86\\&=0.13\end{aligned}$$