DATA SCIENCE 102: MEASURES OF IMPURITY

MEASURING IMPURITY

- How does the decision tree *split*?
- The decision tree splits by two different popular measures:
  - **Gini Impurity** - Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset
  - **Entropy** - Information Gain; higher the information gain, the better the feature is at homogenous data after the split
- The main focus of these measures is about asking **“how do you split first?”**

GINI IMPURITY

GINI IMPURITY

- **Gini Impurity** is the probability of *incorrectly* classifying a randomly chosen element in the dataset if it were randomly labeled *according to the class distribution* in the dataset. It's calculated as

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

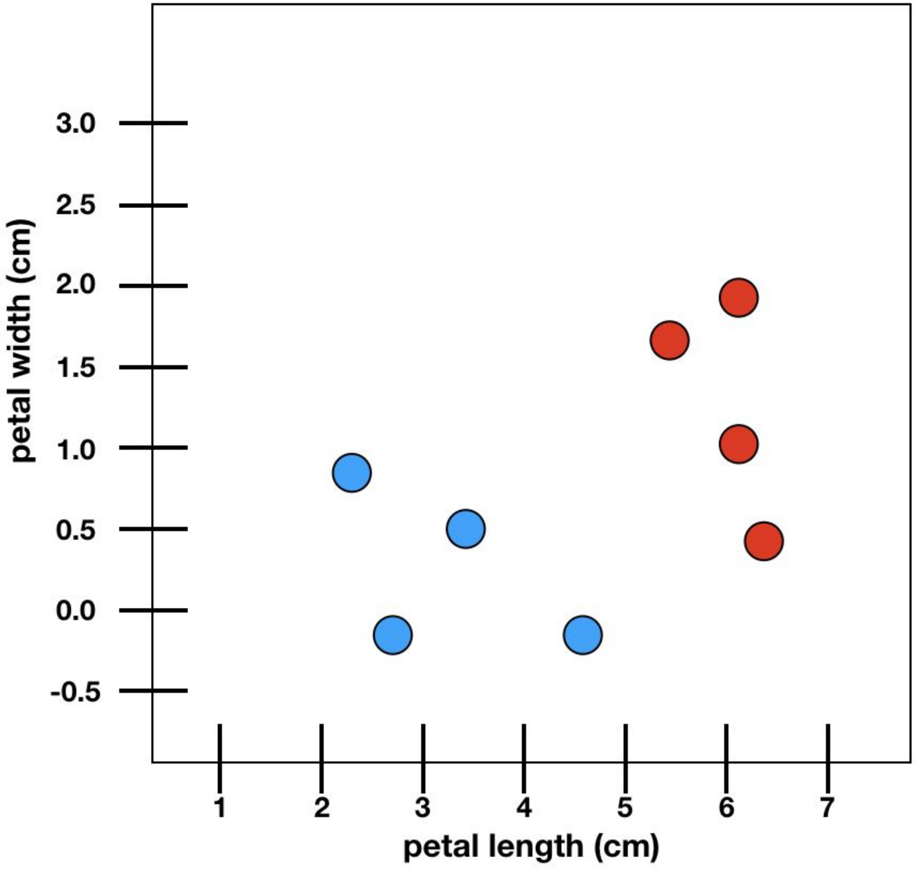
where C is the number of classes and p(i)$p(i)$ is the probability of randomly picking an element of class i

- When training a decision tree, the best split is chosen by **maximizing** the Gini Gain, which is calculated by subtracting **the weighted impurities of the branches from the original impurity**

GINI IMPURITY - EXAMPLE

- Given the following dataset:
- Before the split there is a Gini Impurity of 0.5

Petal Width	Petal Length	Target
1.6	5.5	Red
1	6	Red
0.45	6.4	Red
2	6	Red
0.9	2.5	Blue
-0.2	2.7	Blue
-0.2	4.8	Blue
0.8	2.3	Blue



GINI IMPURITY - EXAMPLE

- Using Gini Impurity, we can quantify which number is the best split
- For example this split:

G (left part) = 0

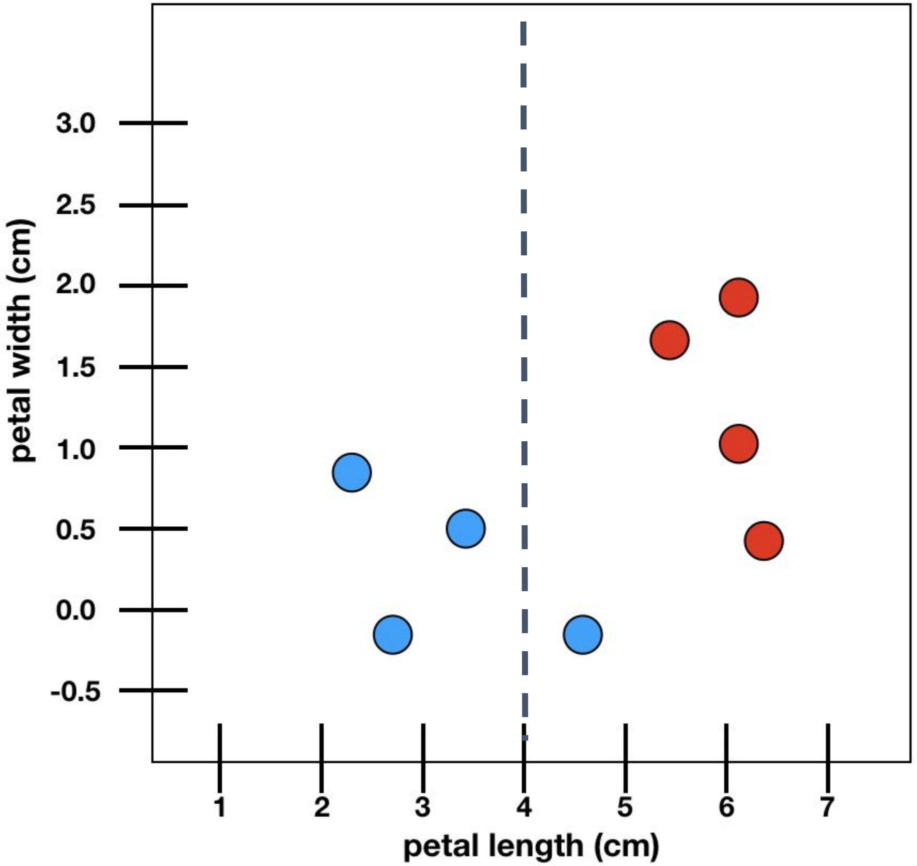
G (right part) = $\frac{1}{5} * (1 - \frac{1}{5}) + \frac{4}{5} * (1 - \frac{4}{5})$

= 0.32

Weighting each branch: Left part + Right part

= $(\frac{3}{8} * 0) + (\frac{5}{8} * 0.32) = 0.2$

Total Impurity removed = 0.5 - 0.2 = 0.3

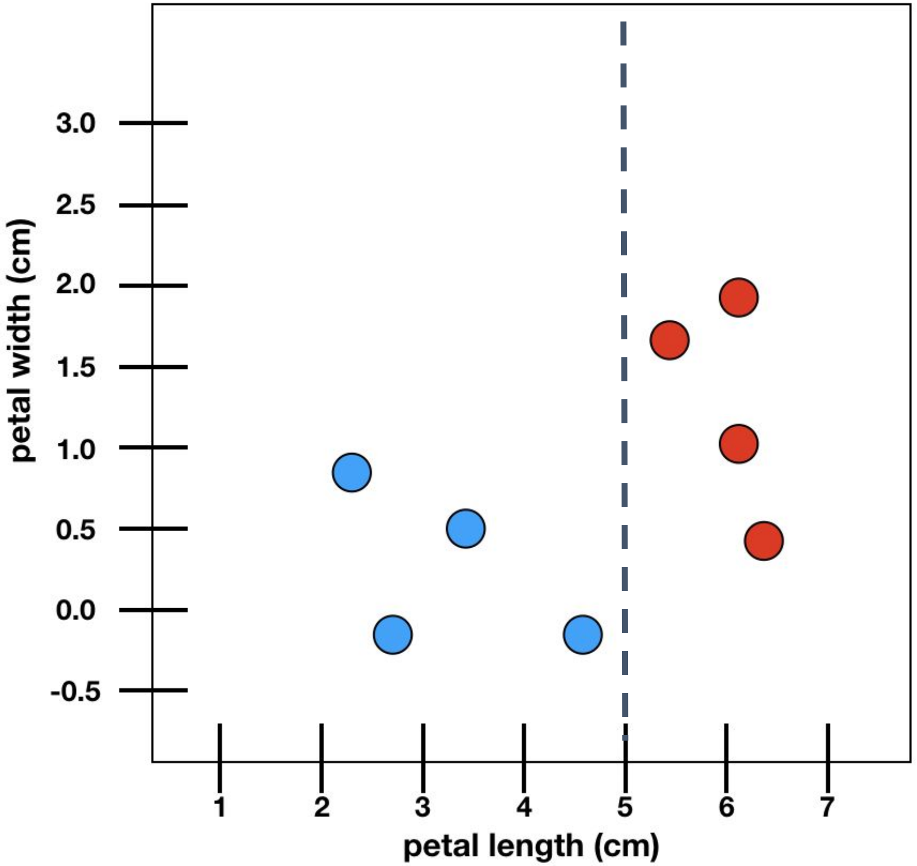


GINI IMPURITY - EXAMPLE

- To get the best split, the amount of impurity removed should be **maximised**
- For example below:

G (left part) = 0

G (right part) = 0



Weighting each branch: Left part + Right part

= (½ * 0) + (½ * 0) = 0

Total Impurity removed = 0.5 - 0 = 0.5

ENTROPY

- Information
- Entropy
- Entropy in Decision Trees

INFORMATION

- Imagine if you rolled a dice, how many true/false question would you need to ask to find out the result?
- For example

result = 2

$6 \div 2 = 3$	is the answer greater than 3? # Q1
$3 \div 2 = 1.5$	is the answer greater than 1.5? # Q2
$1.5 \div 2 = 0.75$	is the answer greater than 1.5? # Q3

- At least 2 questions, at most 3. However for a n-sided dice, how many questions would you need?

INFORMATION

- The number of questions is known as information (I), with unit bits

$$I(\mathrm{p}) = \log_2 \frac{1}{\mathrm{p}}$$

- If we wanted to know the result of a dice roll, we would need, on average, $\log_2 6 = 2.584$ questions to be asked
- It's called Information because you spend it to reduce uncertainty

ENTROPY

- Entropy H is defined as the average information needed to describe all possible results $\mathbf{p}$ for an event X

$$H(X) = \sum p_i \mathrm{I}(p_i) = \sum p_i \log_2 \frac{1}{p_i} = -\sum p_i \log_2 p_i$$

- In a way, entropy measures the total uncertainty in a system / result

ENTROPY

- Example: Coin Flip
- Outcomes: Heads (h) and Tails (t)
- Probabilities: $p_h = \frac{1}{2}$, $p_t = \frac{1}{2}$

$$H(p_h, p_t) = -(p_h \log_2 p_h) - (p_t \log_2 p_t)$$

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

$$= 1$$

ENTROPY

- Example: Coin Flip
- Outcomes: Heads (h) and Tails (t)
- Probabilities: $p_h = \frac{1}{2}$, $p_t = \frac{1}{2}$

$$H(p_h, p_t) = -(p_h \log_2 p_h) - (p_t \log_2 p_t)$$

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

$$= 1$$

ENTROPY

- Example: Balls from a box
- Outcomes: Red (r) and Blue (b)
- Probabilities: $p_r = \frac{2}{6}$, $p_b = \frac{4}{6}$

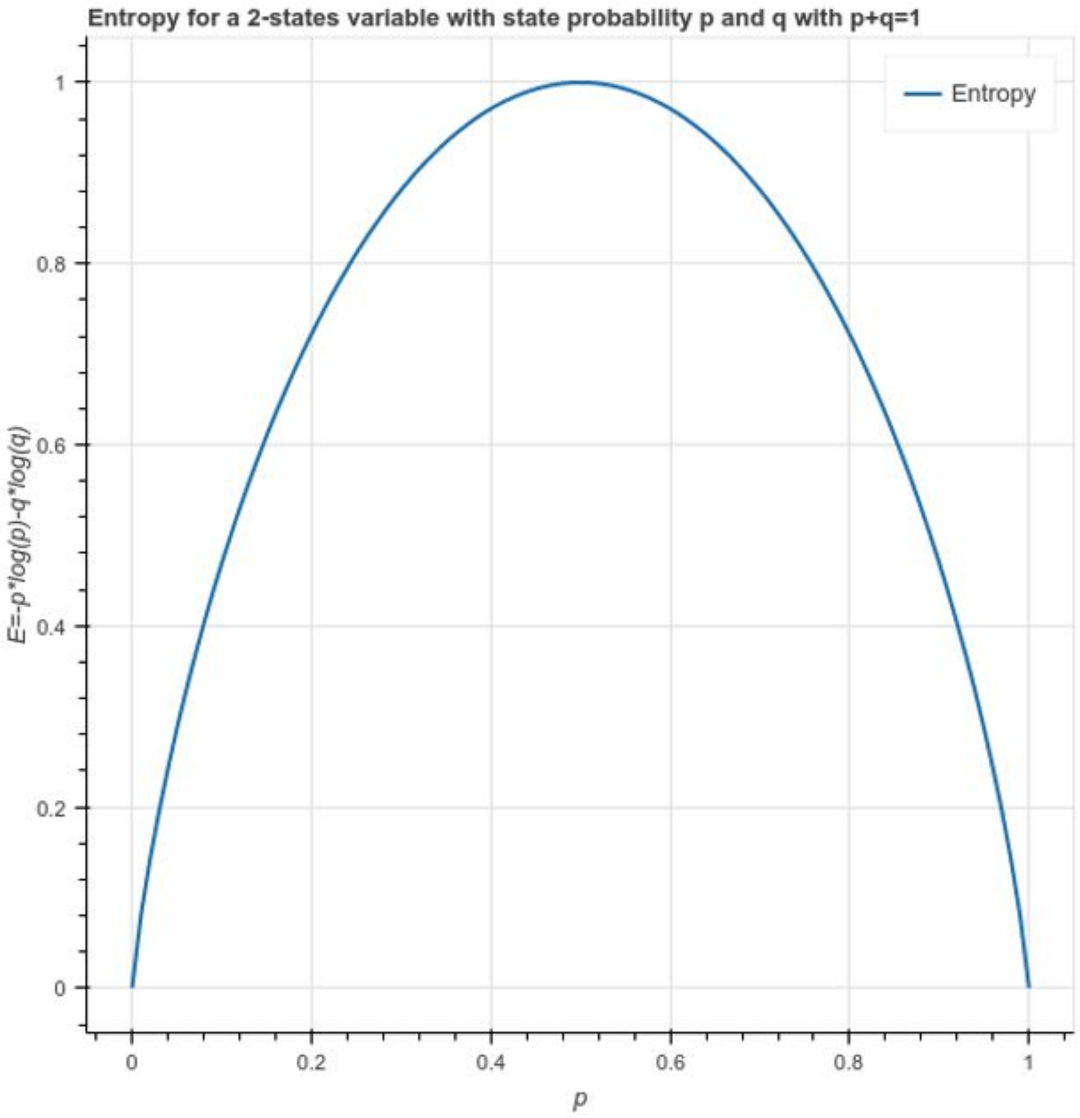
$$H(p_r, p_b) = -(p_r \log_2 p_r) - (p_b \log_2 p_b)$$

$$= -\left(\frac{2}{6}\log_2 \frac{2}{6}\right) - \left(\frac{4}{6}\log_2 \frac{4}{6}\right)$$

$$= 0.918$$

ENTROPY

- Entropy H is defined as the average information needed to describe all possible results $\mathbf{p}$ for an event X
- Ranges from 0 being certain (most pure) to increasing uncertainty ($\log_2 m$)



ENTROPY IN DECISION TREES - INFORMATION GAIN

- For splitting, we need to consider the information gained from splitting where $EH(A)$ is the weighted average of all entropies of each branch

$$EH(A) = \sum \frac{n_i}{N} H(a_i, b_i)$$

$$I(A) = H(a,b) - EH(A)$$

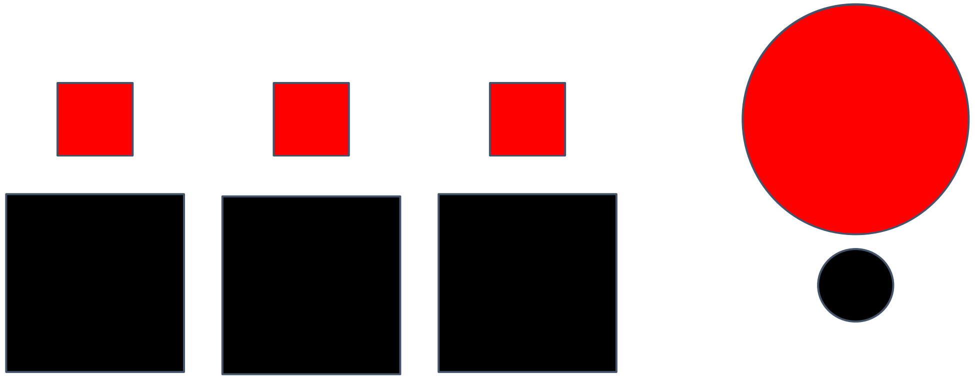
Entropy before splitting

Entropy after splitting

ENTROPY IN DECISION TREES

Split by shape or size?

Shape	Size	Target
Circle	Large	Red
Circle	Small	Black
Square	Small	Red
Square	Small	Red
Square	Large	Black
Square	Large	Black
Square	Large	Black
Square	Small	Red



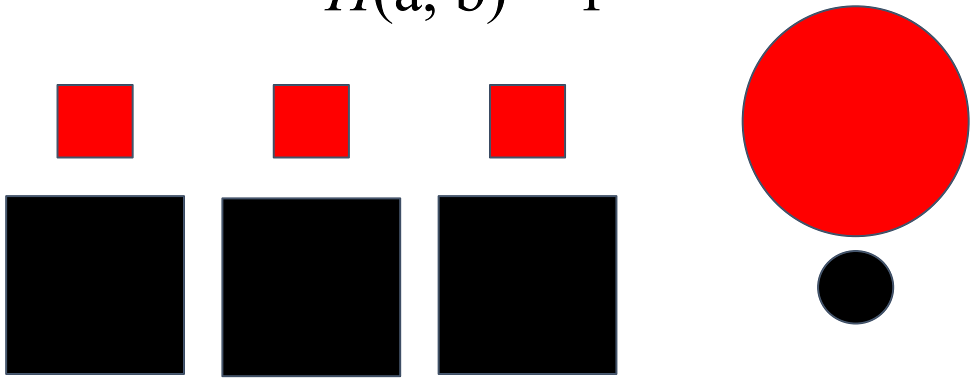
ENTROPY IN DECISION TREES

- **Before** splitting, what is the entropy $H(\mathrm{a}, \mathrm{b})$?
- Let $\mathrm{a} = \mathrm{p}_{\mathrm{red}}$, $\mathrm{b} = \mathrm{p}_{\mathrm{black}}$
- Similar to a coin flip,

  Probabilities: $\mathrm{a} = \frac{1}{2}$, $\mathrm{b} = \frac{1}{2}$

  $H(\mathrm{a}, \mathrm{b}) = 1$

$H(\mathrm{a}, \mathrm{b}) = 1$

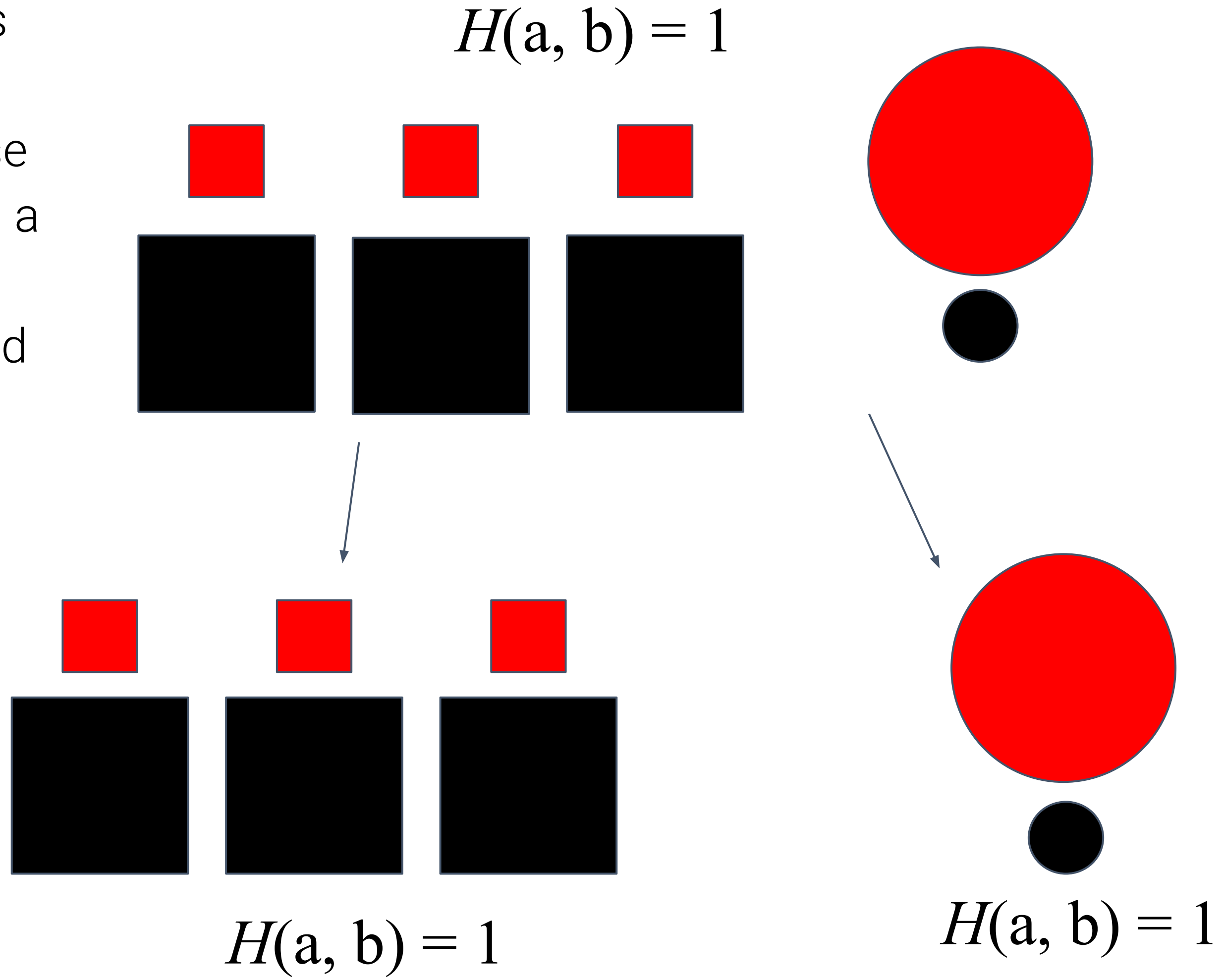


ENTROPY IN DECISION TREES

- Now let's consider splitting by **shape**, what is the entropy $EH(A)$?
- It's the **same** as the previous entropy because each branch has the same chance of getting a red and black color
- Therefore, information gained is 0 and it's bad to split by shape

$$H = 1, EH(A) = 1$$

$$I(A) = H(a, b) - EH(A) = 0$$



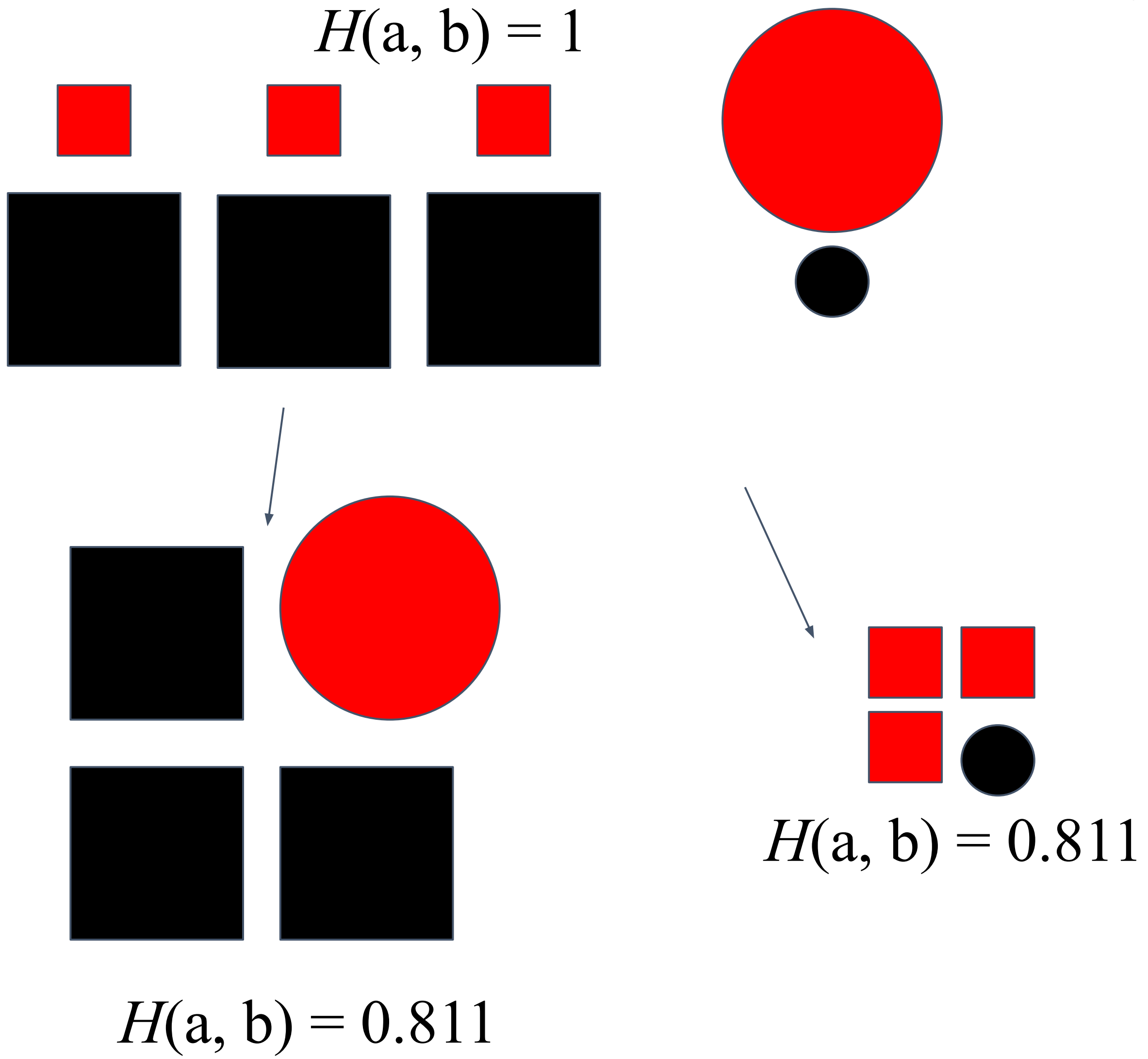
ENTROPY IN DECISION TREES

- Now let's consider splitting by **size**, what is the entropy $EH(A)$?
- Since both branches are now different in its probabilities the entropy for each reduces
- The information gained from this split is 0.189 bits
- Therefore it is better to split by **size**

$$H(a,b) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.811$$

$$EH(A) = \frac{1}{2}(0.811) + \frac{1}{2}(0.811) = 0.811$$

$$I = 1 - 0.811 = 0.189$$



WHICH TO USE

WHICH TO USE?

- There is not much difference in terms of both measures
- They are both good to use
- However, Gini Impurity is preferred because it is simpler to compute without using the log function