Machine Learning

| Entropy

Prof. Dr. Mostafa Elhosseini

https://youtube.com/ElhosseiniAcademy

Agenda

- Entropy
- Recommendation Application
- Gini Vs. Entropy

Entropy

- **By default**, the **DecisionTreeClassifier** class uses the **Gini** impurity measure, but you can **select the entropy** impurity measure instead by setting the **criterion** hyperparameter to "entropy".

- The concept of entropy originated in **thermodynamics** as a **measure of molecular disorder**: entropy approaches zero when molecules are still and well ordered.

Entropy

- Entropy later spread to a wide variety of domains, including in Shannon’s information theory, where it measures the average information content of a message
- Entropy is zero when all messages are identical.

- Plays a crucial role in **building decision trees**, particularly in determining the **best feature to split the data**.

    — helps quantify **how well a feature separates the data into distinct classes.**

Entropy

- In a set with m elements and n classes, with a_i elements belonging to the $i-th$ class, the entropy is

$$\text{Entropy} = -p_1 \log_2(p_1) - p_2 \log_2(p_2) - \cdots - p_n \log_2(p_n)$$

$$\text{Entropy} = -\sum_{i=1}^{n} p_i \log_2 p_i$$

- Where $p_i = \frac{a_i}{m}$

Entropy

Calculation Example: A dataset with two classes: Yes and No.

- Before Split: Total Instances: 10 Yes: 6 No: 4
- $p_{yes} = \frac{6}{10} = 0.6$ $p_{no} = \frac{4}{10} = 0.4$
- Entropy $= -(0.6 \log_2 0.6) - 0.4 \log_2 0.4 \sim 0.971$

Entropy

After Split on Feature X:

- Subset 1: Yes: 4 No: 0

  Entropy=0 (pure)

- Subset 2: Yes: 2 No: 4

  $p_{yes} = 0.333$ $p_{no} = 0.667$

  Entropy ~ 0.918

Entropy

- Weighted Entropy: $\left(\frac{4}{10} \times 0\right) + \left(\frac{6}{10} \times 0.918\right) \sim 0.551$

Information Gain:

- $Entropy_{before} - \ weighted\ entropy_{after}$ = 0.971–0.551=0.420

Entropy

- Classifier 1 (by **platform**):
  - Left leaf (iPhone): {A, C, C}
  - Right leaf (Android): {A, A, B}
- Classifier 2 (by **age**):
  - Left leaf (young): {A, A, A}
  - Right leaf (adult): {B, C, C}

Platform	Age	App
iPhone	Young	Atom Count
iPhone	Adult	Check Mate Mate
Android	Adult	Beehive Finder
iPhone	Adult	Check Mate Mate
Android	Young	Atom Count
Android	Young	Atom Count

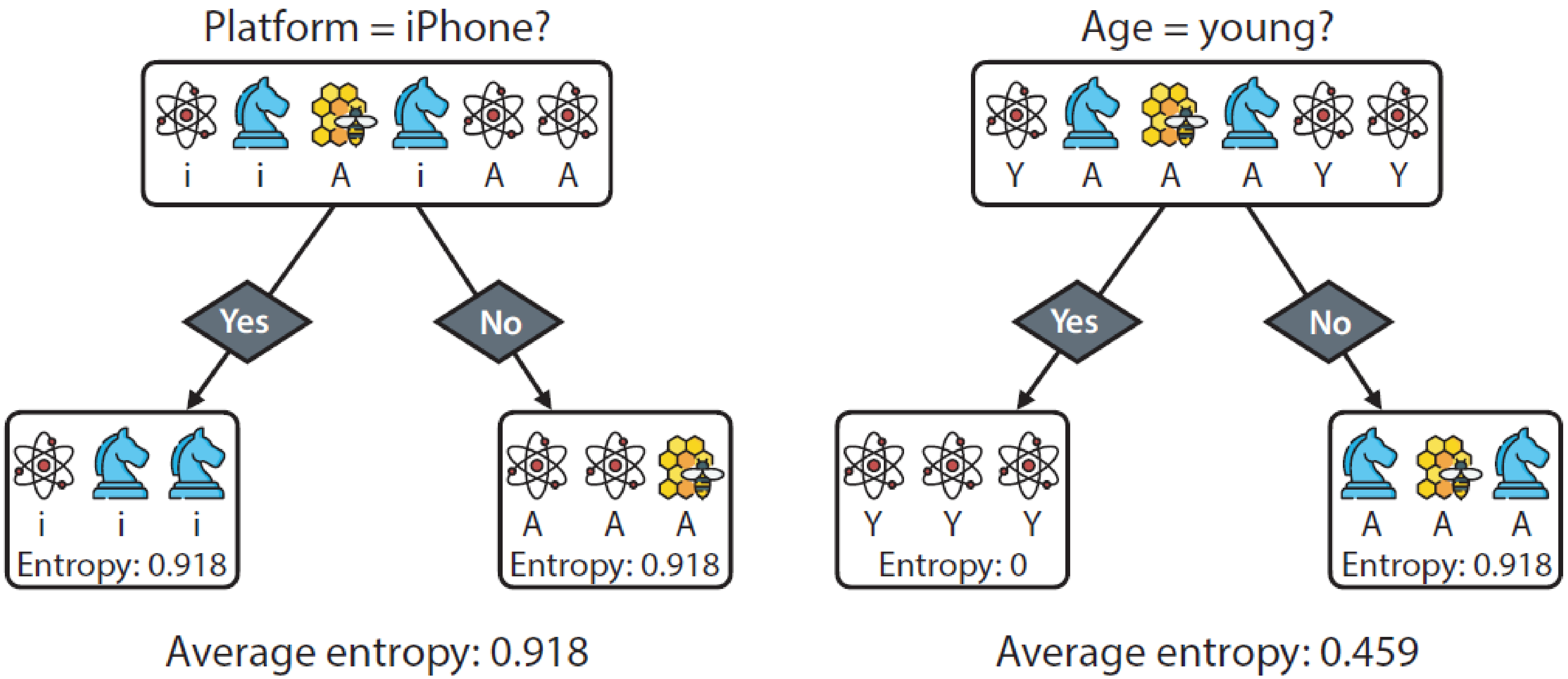
Entropy

- Classifier 1 (by platform):
  - Left leaf (iPhone): {A, C, C}
  - Entropy $= -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$
  - Right leaf (Android): {A, A, B}
  - Entropy $= -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$
- Average $= \frac{1}{2}(0.918 + 0.918) = 0.918$

- Classifier 2 (by **age**):
  - Left leaf (young): {A, A, A}
  - Entropy $= 0$
  - Right leaf (adult): {B, C, C}
  - Entropy $= -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$
- Average $= \frac{1}{2}(0 + 0.918) = \mathbf{0.459}$

Entropy

- Splitting the dataset by age gives us two smaller datasets with a lower average entropy. Therefore, we again choose to split the dataset by age



Weighted Average Entropy

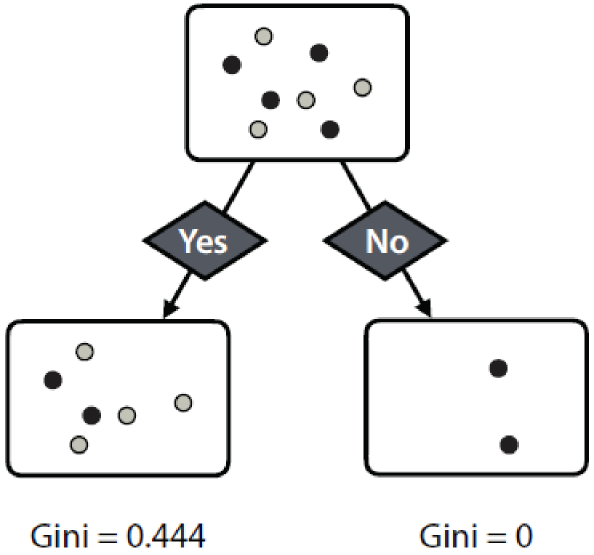
- When constructing decision trees, splits often result in subsets of varying sizes. To accurately assess the quality of these splits using metrics like the Gini Index or Entropy, it's crucial to **account for the different sizes of each subset**
- Splitting a dataset into subsets of different sizes means that larger subsets have a more significant impact on the overall impurity.

Weighted Average Entropy

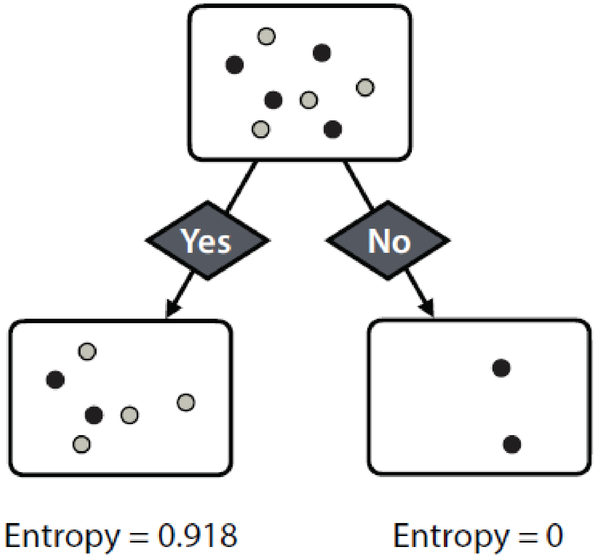
- Dataset Split:
  - Total Data Points: 8
  - Subset 1: 6 data points → Weight = 6/8 = 0.75
  - Subset 2: 2 data points → Weight = 2/8 = 0.25
- Weighted Impurity:
  - Subset 1 Gini: 0.40 → Weighted = 0.75 * 0.40 = 0.30
  - Subset 2 Gini: 0.60 → Weighted = 0.25 * 0.60 = 0.15
  - Total Weighted Gini: 0.30 + 0.15 = 0.45

Weighted Average Entropy

- A split of a dataset of size eight into two datasets of sizes six and two



Weighted average Gini = $0.444 \cdot \frac{6}{8} + 0 \cdot \frac{2}{8} = 0.333$



Weighted average entropy = $0.918 \cdot \frac{6}{8} + 0 \cdot \frac{2}{8} = 0.689$

Practical Example

- Building a decision tree to classify whether to play tennis based on weather conditions.

- Dataset Split:
  - Before Split:
    - Yes: 9
    - No: 5
    - $$\text{Entropy} = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) \approx 0.940$$
  - After Split on "Outlook":
    - Sunny:
      - Yes: 2
      - No: 3
      - $$\text{Entropy} = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) \approx 0.971$$

Practical Example

- Building a decision tree to classify whether to play tennis based on weather conditions.

- **Overcast**:
  - **Yes**: 4
  - **No**: 0
  - $$\text{Entropy} = 0$$
- **Rain**:
  - **Yes**: 3
  - **No**: 2
  - $$\text{Entropy} = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) \approx 0.971$$

- **Weighted Entropy**:

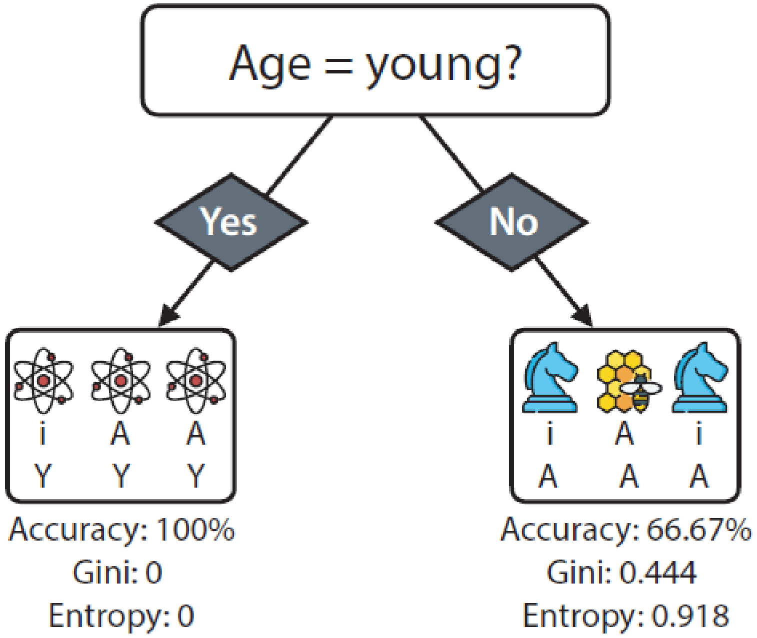
$$\left(\frac{5}{14} \times 0.971\right) + \left(\frac{4}{14} \times 0\right) + \left(\frac{5}{14} \times 0.971\right) \approx 0.694$$

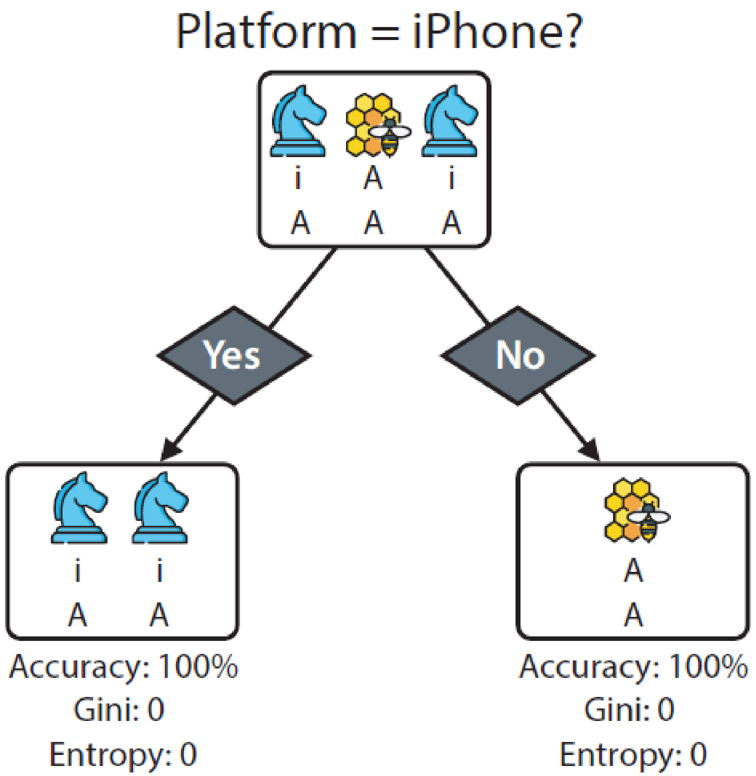
- **Information Gain**:

$$0.940 - 0.694 = 0.246$$

Interpretation: Splitting on "Outlook" provides an information gain of **0.246**, indicating that this feature effectively reduces uncertainty in the dataset.

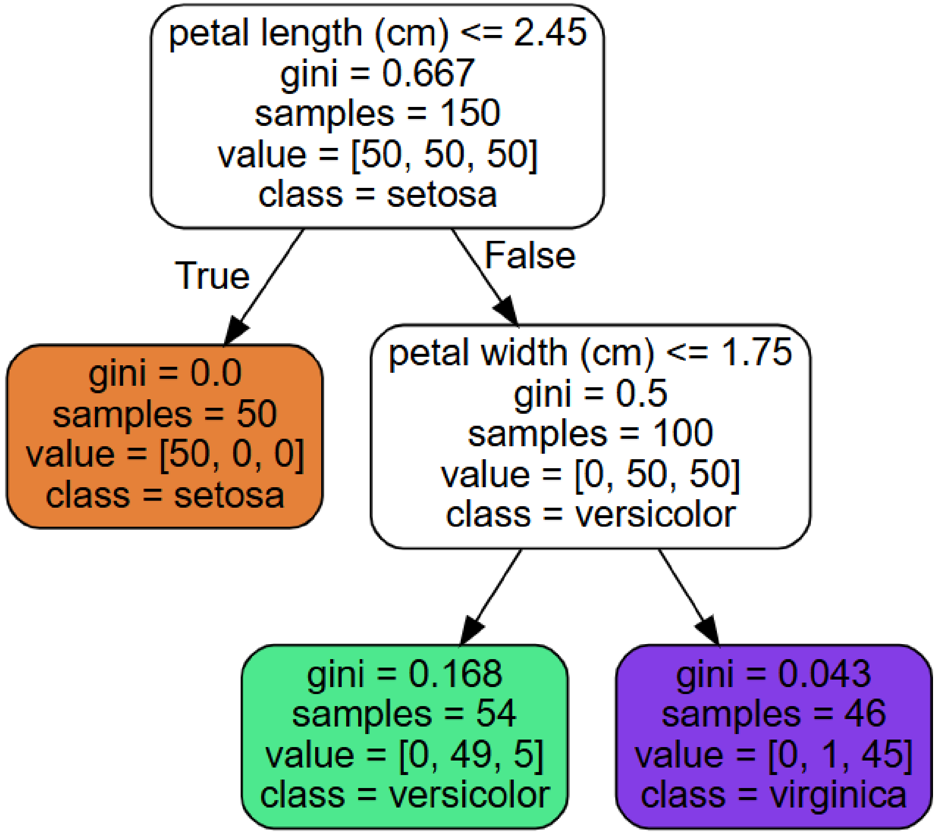
Second step to build the model: Iterating





Gini Impurity or Entropy?

- depth-2 left node
- Entropy $= -\left(\frac{49}{54}\right)\log_2\left(\frac{49}{54}\right) - \left(\frac{5}{54}\right)\log_2\left(\frac{5}{54}\right) = 0.445$



Comparison with Other Metrics

- Entropy (Information Gain):
  - Both Gini Impurity and Entropy measure node purity.
  - Gini tends to be faster to compute and often produces similar results to Entropy.
  - Entropy provides a more nuanced measure of impurity but is computationally more intensive.
- Choosing Between Them:
  - Gini is preferred in algorithms like CART (Classification and Regression Trees).
  - Entropy is used in algorithms like ID3 and C4.5.