Decision Trees

What Are Decision Trees?

- A **supervised learning** method for classification
- Uses **tree-structured sequences of tests**
- Each internal node tests an attribute
- Each leaf predicts a class
- Learned using **top-down, recursive splitting**

Why Decision Trees?

- Simple, interpretable models
- Can handle:
  - Nominal attributes
  - Numeric attributes
  - Missing values
- Fast prediction
- Can be converted into rules

Basic Divide-and-Conquer Process

1. Select the **best attribute** to split the data
2. Split dataset into subsets based on attribute values
3. Recurse on each subset
4. Stop when:
   - All instances have same class
   - No attributes remain
   - No further information gained

Entropy

- Entropy $H(X)$ of a random variable

$$H(X) = -\sum_{i=1}^{n} P(X = i) log_2(P(X = i))$$

- Specific conditional entropy $H(X \mid Y = v)$ of X given $Y = v$

$$H(X \mid Y = v) = -\sum_{i=1}^{n} P(X = i \mid Y = v) log_2(P(X = i \mid Y = v))$$

- Conditional entropy $H(X \mid Y)$ of X given Y

$$H(X \mid Y) = -\sum_{v \in values(Y)} P(Y = v) H(X \mid Y = v)$$

Choosing the Best Attribute

- Mutual information of X and Y:

$$I(X, Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$

- Uses **Information Gain** = entropy(parent) - (average entropy(children))
- The attribute with the **highest gain** is selected.

Example: Weather Dataset

- Attributes considered at root:
  - Outlook
  - Temperature
  - Humidity
  - Windy
- Compute **information gain** for each to decide root split.

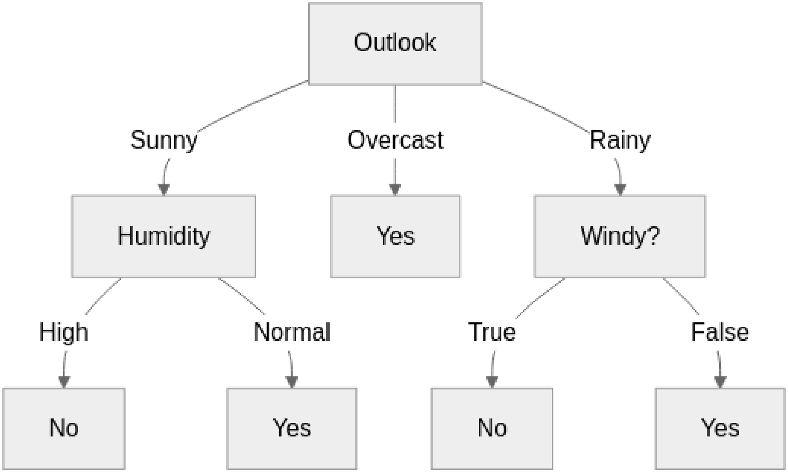
Example Calculation (Simplified)

- For *Outlook*:
- Leaf distributions:
  - sunny: 2 yes, 3 no
  - overcast: 4 yes, 0 no
  - rainy: 3 yes, 2 no
- Weighted entropy:

$$Gain(Outlook) = 0.940 - 0.693 = 0.247$$

- Outlook is the **best attribute**.

Weather Decision Tree



Stopping Criteria

Stop when: - All examples have same class
- No attributes left
- No further information gain
- Identical attribute values but conflicting labels

Example: ID Code Attribute

- Unique value per instance
- InfoGain = maximal
- SplitInfo = extremely large
- GainRatio becomes small -> avoids useless splits

Handling Numeric Attributes

- Decision trees use **threshold-based splits**:

  ```
  Temperature <= 75?
  Humidity > 82.5?
  ```

- Search for threshold with **maximum info gain**.

Handling Missing Values

- During Training:
  - Instances split **fractionally** across branches
- During Testing:
  - Follow all possible branches, weighted by probability
  - Final prediction is **probability-weighted**

Advantages of Decision Trees

- Easy to interpret
- Handles mixed data types
- Minimal preprocessing
- Fast inference
- Easily converted into rule sets

Limitations

- Can overfit (requires pruning)
- Greedy split selection may miss global optimum
- Sensitive to noise and small data changes
- Decision boundaries are axis-aligned

Pruning (Brief Overview)

- Reduces overfitting:
  - **Pre-pruning**: stop early based on thresholds
  - **Post-pruning**: grow full tree -> remove unreliable branches
- Improves generalization.

Summary

- Decision trees use **divide-and-conquer** strategy
- Attribute selection via **information gain**
- Handles numeric, nominal, and missing values
- Produces interpretable, rule-like models