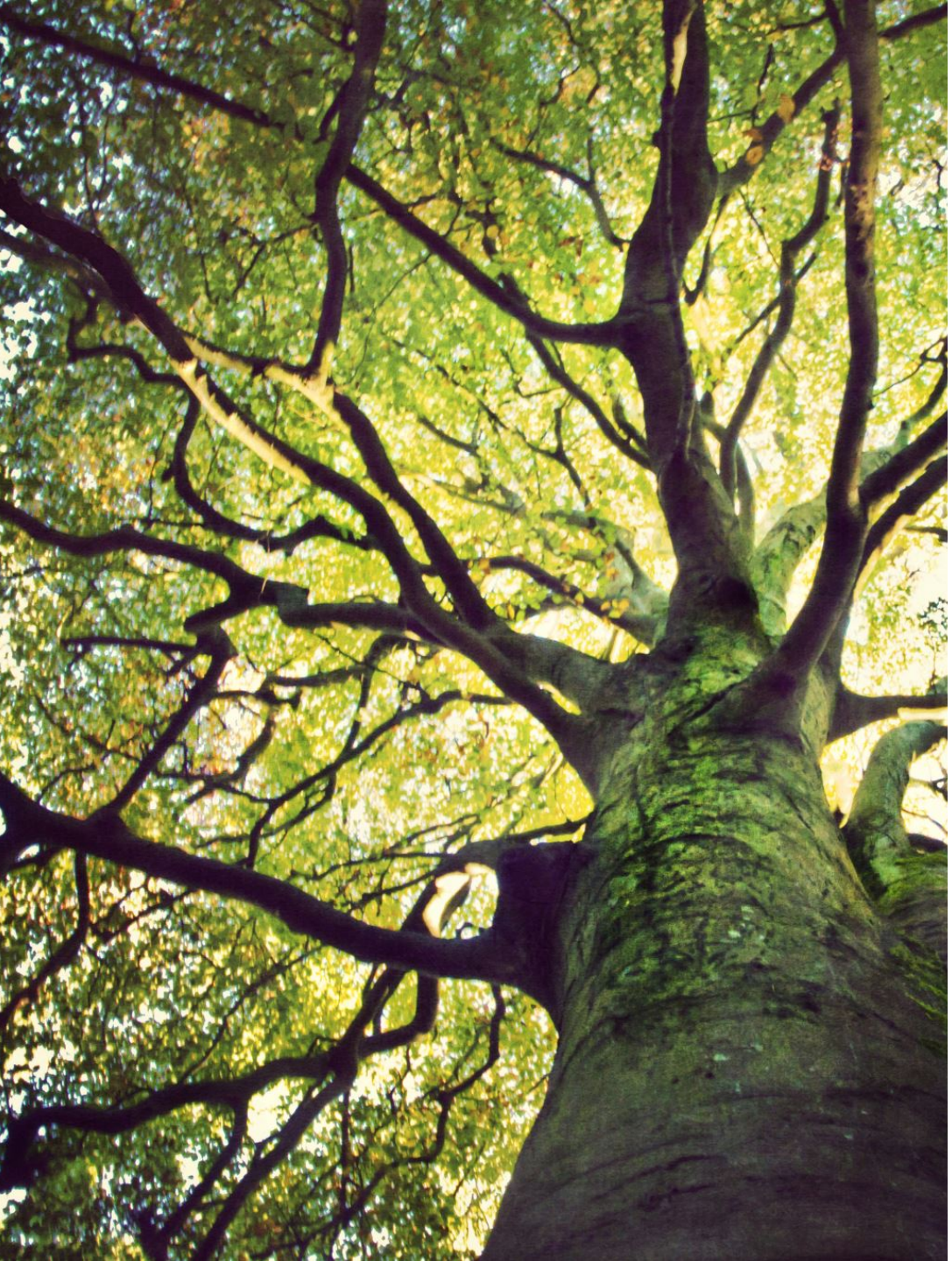

Decision Trees

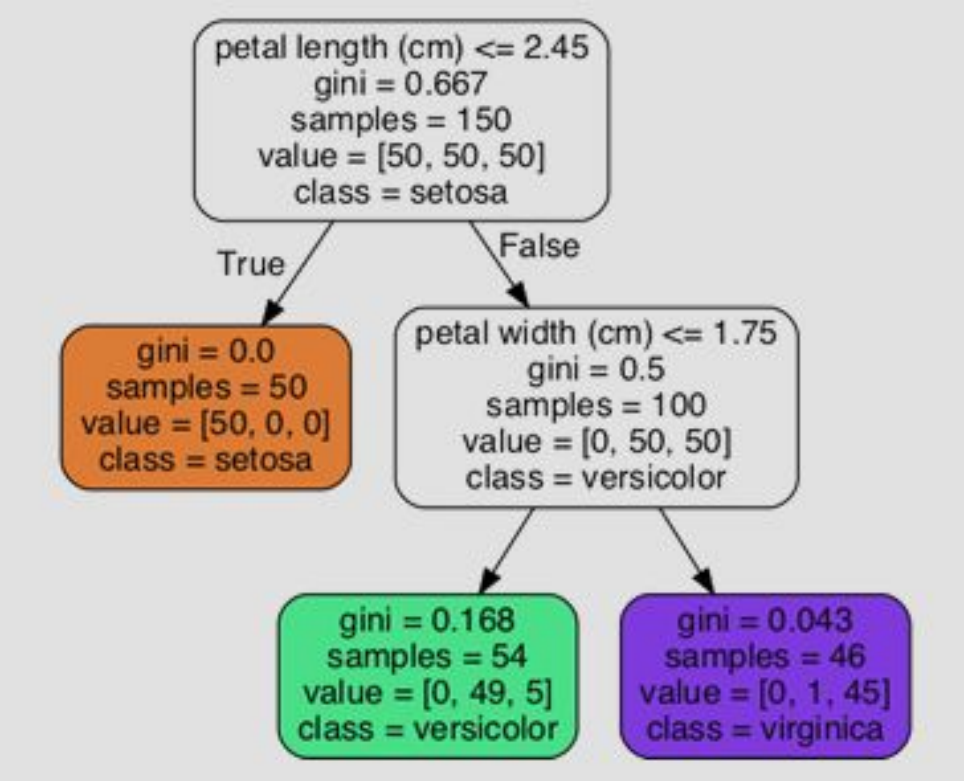


Introduction to Decision Trees

- **What is a Decision Tree?**
  - A flowchart-like structure
  - Used for both classification and regression
  - Mimics human decision-making
- **Key Terms:**
  - Each **internal node** represents a test on an attribute.
  - Each **branch** represents an outcome of the test.
  - Each **leaf node** represents a class label or decision.

Training and Visualizing a Decision Tree

- https://drive.google.com/file/d/1IRHHx6kV-NpytBlxGwovEVLRVYRSPg5-/view?usp=sharing



Making Predictions

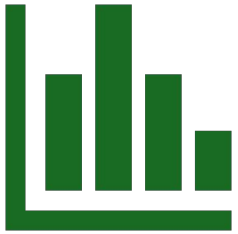
- How a decision tree makes predictions:
  - Start at root
  - Compare feature with threshold
  - Move to corresponding child node
  - Repeat until leaf node is reached
- **White box models:** Decision Trees are intuitive, and their decisions are easy to interpret.

Example: *AllElectronics* Customer Database

RID	age	income	student	credit_rating	buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Constructing the Tree

Goal: Choose the best attribute to split the data.



Methods:

Information Gain
Gini Index
Gain Ratio

Information Gain

- Based on **entropy** and **information theory**.
- **Entropy Formula**:

$$Entropy(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

- **Entropy for Attribute:**

$$Entropy_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Entropy(D_j)$$

- **Information Gain**:

$$Gain(A) = Entropy(D) - Entropy_A(D)$$

Example: Entropy Calculation

-
- For the *AllElectronics* dataset:

$$Entropy(D) = -\left(\frac{9}{14}\right)\log^2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\log^2\left(\frac{5}{14}\right) \approx 0.940$$

Example: Information Gain for *age*

- Values: youth, middle_aged, senior

$$Gain(age) = Entropy(D) - \left[\left(\frac{5}{14}\right) \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \left(\frac{4}{14}\right) \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) + \left(\frac{5}{14}\right) \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)\right] \approx 0.940 - 0.694 \approx 0.246$$

- Similarly
  - $Gain(income) = 0.029$
  - $Gain(student) = 0.151$
  - $Gain(credit\ rating) = 0.048$
- Because *age* has the highest information gain among the attributes, it is selected as the splitting attribute.

Gain Ratio

- introduced to reduce bias toward multi-valued attributes.
- The attribute with the maximum gain ratio is selected as the splitting attribute.
- Split Information:

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

Computation of Gain Ratio for the Attribute *income*

- $Splitinfo_{income}(D)$

$$= -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

- $Gain(income) = 0.029$

- $GainRatio(income) = \frac{0.029}{1.557} = 0.019$

Gini Index

- **Gini Impurity**:

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

- p_i is the probability that a tuple in D belongs to class C_i and is estimated by $\frac{|C_{i,D}|}{D}$.
- The sum is computed over m classes.
- **Splitting Criterion**: Choose attribute with smallest Gini index after split.

Induction of a Decision Tree Using The Gini Index

- Gini index to compute the impurity of D:

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2$$

- Split on *age*
  - $youth$: $5\ (yes = 2, no = 3) \rightarrow\ G = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$
  - $middle - aged$: $4\ (yes = 4, no = 0) \rightarrow\ G = 0$
  - $senior$: $5\ (yes = 3, no = 2) \rightarrow\ G = 0.48$
- Weighted:
  - $\frac{5}{14} \cdot 0.48 + \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0.48 = \frac{10}{14} \cdot 0.48 \approx 0.3429$

Continue…

- Split on *student*
  - no: 7 $(yes = 3, no = 4) \rightarrow\ G = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = \frac{24}{49} \approx 0.4898$
  - yes: 7 $(yes = 6, no = 1) \rightarrow\ G = 1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2 = \frac{12}{49} \approx 0.2449$
- Weighted:
  - $\frac{7}{14} \cdot 0.4898 + \frac{7}{14} \cdot 0.2449 \approx 0.3673$
- Similarly,
  - Split on *income* = 0.4405
  - Split on *credit_rating* = 0.4286
- Best feature to split node is ***age***.

Estimating Class Probabilities

-

- Decision trees can estimate probabilities
- **Equation:**

$$p_k = \frac{number\ of\ instances\ of\ class\ k\ in\ the\ leaf}{total\ instances\ in\ the\ leaf}$$

The CART Training Algorithm

- CART: Classification and Regression Trees
- How it works:
  - Splits training set into two subsets
  - Uses feature k and t_k (e.g., “petal length ≤ 2.45 cm”)
  - Minimizes cost function (e.g., Gini impurity)
- **Equation:**

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

- Where $\begin{cases} G_{left}\ and\ G_{right} \text{ measures the impurity of the left/right subset} \\ m_{left}\ and\ m_{right} \text{ is the number of instances in the left/right subset} \end{cases}$

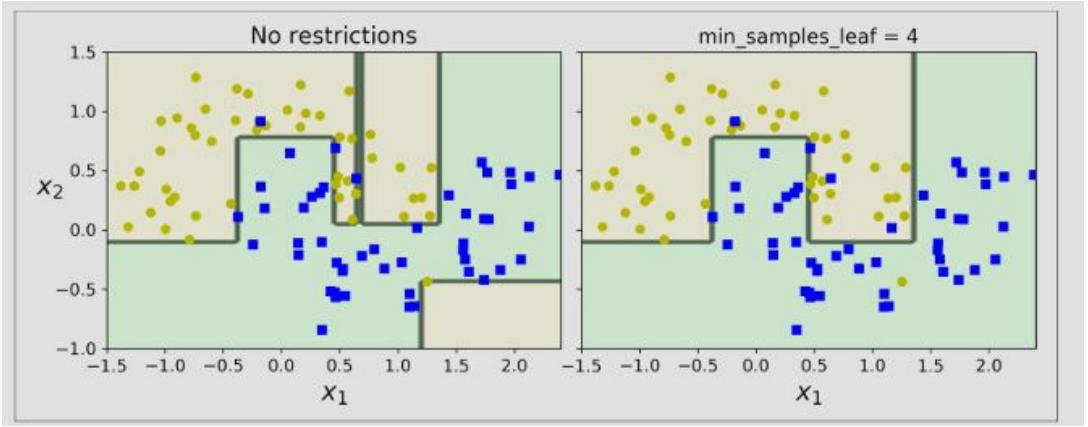
Computational Complexity

-

- **Prediction:** $O(\log_2(m))$
- **Training:** $O(n \times m\log_2(m))$
- Not suitable for very large datasets without tuning

Regularization Hyperparameters

- To avoid overfitting:
  - $\boldsymbol{max\_depth}$: Reducing max_depth will reduce the risk of overfitting.
  - $\boldsymbol{min\_samples\_split}$: The minimum number of samples a node must have before it can be split.
  - $\boldsymbol{min\_samples\_leaf}$: The minimum number of samples a leaf node must have.
  - $\boldsymbol{max\_leaf\_nodes}$: The maximum number of leaf nodes
  - $\boldsymbol{max\_features}$: The maximum number of features that are evaluated for splitting at each node.

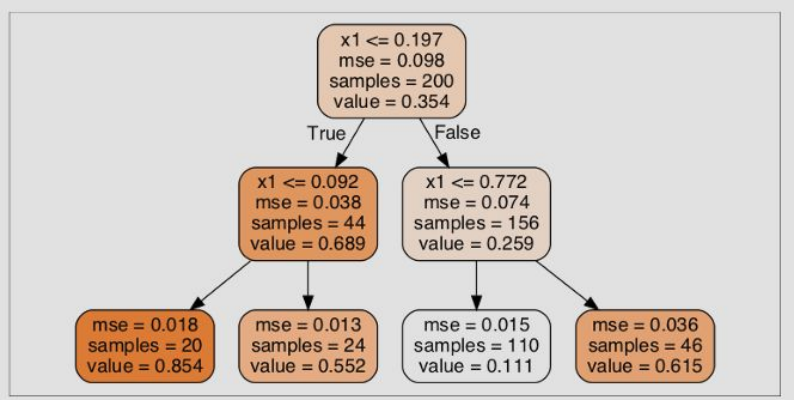


Pruning

- A node whose children are all leaf nodes is considered unnecessary if the purity improvement it provides is not statistically significant.
- **Pruning Methods**:
  - **Pre-pruning:** Stop early during building.
  - **Post-pruning:** Remove branches after tree is built.

Regression with Decision Trees

- CART cost function for regression
- $J(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right}$
- Where $\begin{cases} MSE_{node} = \sum_{i \in node} (\hat{y}_{node} - y^i)^2 \\ \hat{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y^i \end{cases}$



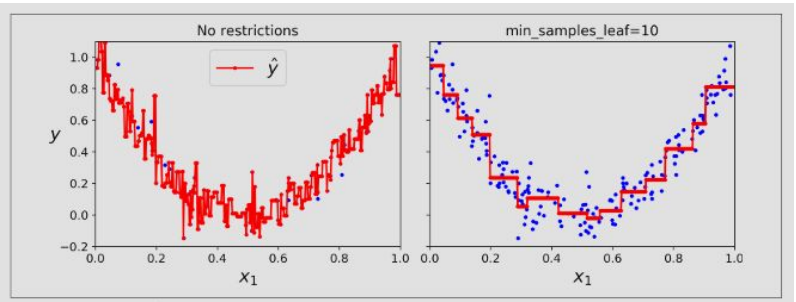


Figure 6-6. Regularizing a Decision Tree regressor

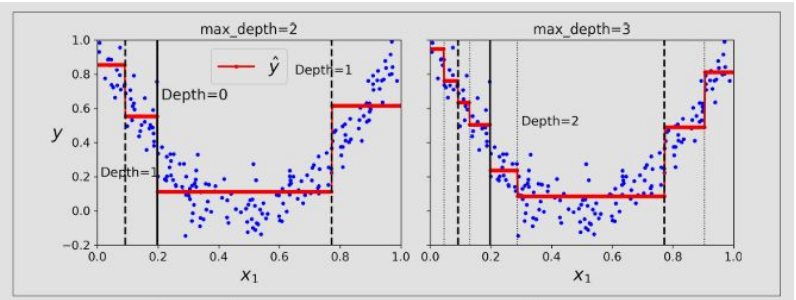


Figure 6-5. Predictions of two Decision Tree regression models

Example: Predicting House Prices Based on Square Footage

Square Feet (x)	Price (y) in $1000s
1000	200
1500	250
2000	300
2500	350
3000	400
3500	450

Step 1: Calculate Possible Splits

- First, we need to find all possible split points between our data points:
- Possible splits: 1250, 1750, 2250, 2750, 3250

Step 2: Calculate MSE for Each Split

Split at 1250:

- Left node (≤1250): [1000]
  - Mean = 200
  - MSE = $(200-200)^2 = 0$
- Right node (>1250): [1500,2000,2500,3000,3500]
  - Mean = (250+300+350+400+450)/5 = 350
  - MSE = $(250-350)^2 + (300-350)^2 + (350-350)^2 + (400-350)^2 + (450-350)^2$
  - MSE = 10000 + 2500 + 0 + 2500 + 10000 = 25000
- Total MSE = 0 + 25000 = 25000

Split at 1750:

- Left node (≤1750): [1000,1500]
  - Mean = (200+250)/2 = 225
  - MSE = $(200-225)^2 + (250-225)^2 = 625 + 625 = 1250$
- Right node (>1750): [2000,2500,3000,3500]
  - Mean = (300+350+400+450)/4 = 375
  - MSE = $(300-375)^2 + (350-375)^2 + (400-375)^2 + (450-375)^2$
  - MSE = 5625 + 625 + 625 + 5625 = 12500
- Total MSE = 1250 + 12500 = 13750

Continue…

Split at 2250:

- Left node (≤2250): [1000,1500,2000]
  - Mean = (200+250+300)/3 = 250
  - MSE = $(200-250)^2 + (250-250)^2 + (300-250)^2$ = 2500 + 0 + 2500 = 5000
- Right node (>2250): [2500,3000,3500]
  - Mean = (350+400+450)/3 = 400
  - MSE = $(350-400)^2 + (400-400)^2 + (450-400)^2$ = 2500 + 0 + 2500 = 5000
- Total MSE = 5000 + 5000 = 10000

Split at 2750:

- Left node (≤2750): [1000,1500,2000,2500]
  - Mean = (200+250+300+350)/4 = 275
  - MSE = $(200-275)^2 + (250-275)^2 + (300-275)^2 + (350-275)^2$
  - MSE = 5625 + 625 + 625 + 5625 = 12500
- Right node (>2750): [3000,3500]
  - Mean = (400+450)/2 = 425
  - MSE = $(400-425)^2 + (450-425)^2$ = 625 + 625 = 1250
- Total MSE = 12500 + 1250 = 13750

Continue…

Split at 3250:

- Left node (≤3250): [1000,1500,2000,2500,3000]
  - Mean = (200+250+300+350+400)/5 = 300
  - MSE = $(200-300)^2 + (250-300)^2 + (300-300)^2 + (350-300)^2 + (400-300)^2$
  - MSE = 10000 + 2500 + 0 + 2500 + 10000 = 25000
- Right node (>3250): [3500]
  - Mean = 450
  - MSE = $(450-450)^2 = 0$
- Total MSE = 25000 + 0 = 25000

Step 3: Choose the Best Split

- **Best split is at 2250 with MSE = 10000**

Split Point	Total MSE
1250	25000
1750	13750
2250	**10000**
2750	13750
3250	25000

Instability of Decision Trees

- Sensitive to small variations in data
- Rotations can ruin performance
- **Solution:** Use PCA or ensemble methods (e.g., Random Forests)

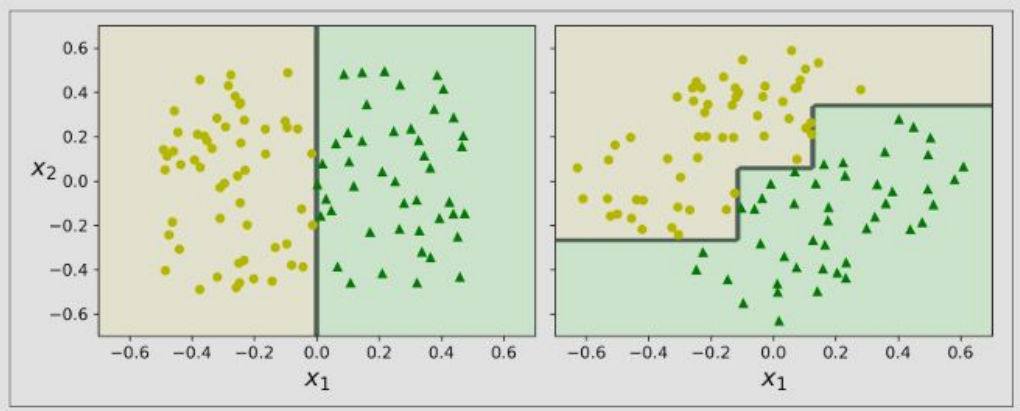


Figure 6-7. Sensitivity to training set rotation

Advantages of Decision Trees

- Easy to understand and interpret
- Require little data preprocessing
- Handle both numerical and categorical data

Disadvantages

- Prone to overfitting.
- Can be unstable with small data changes.
- Biased toward attributes with more levels.