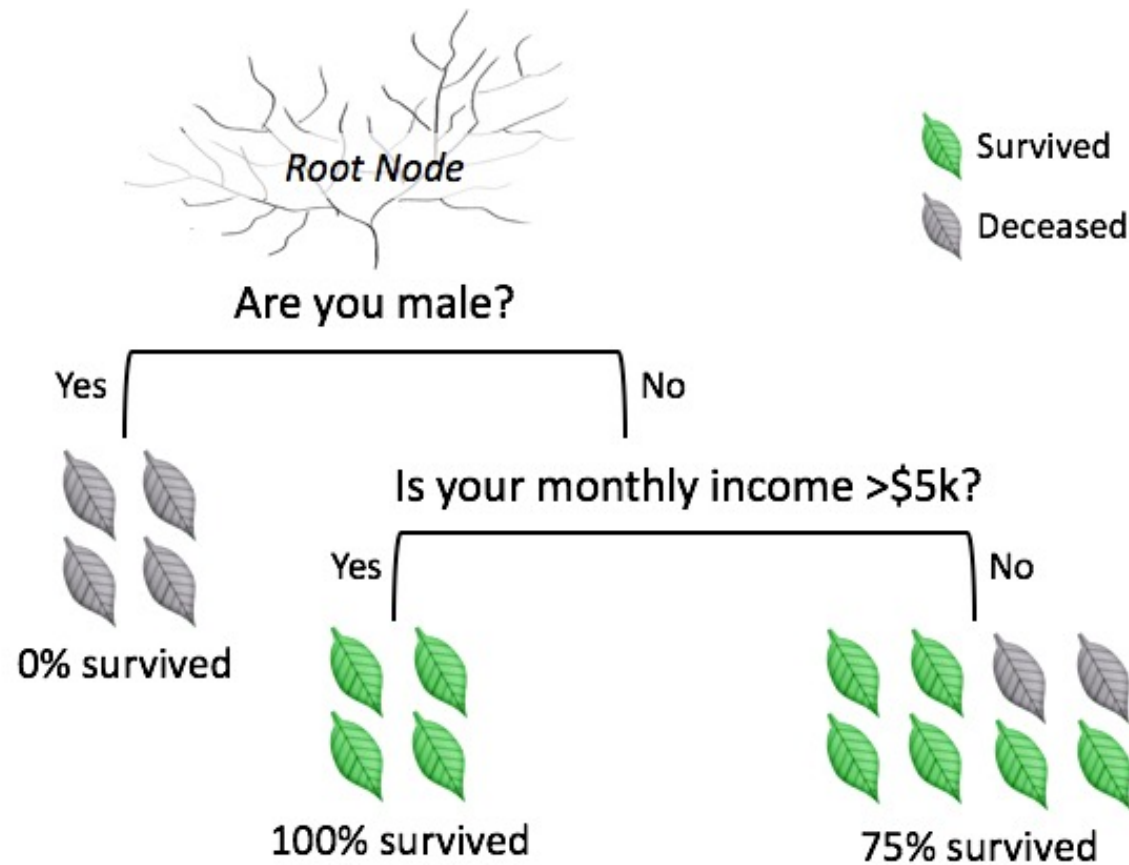


# Decision Tree

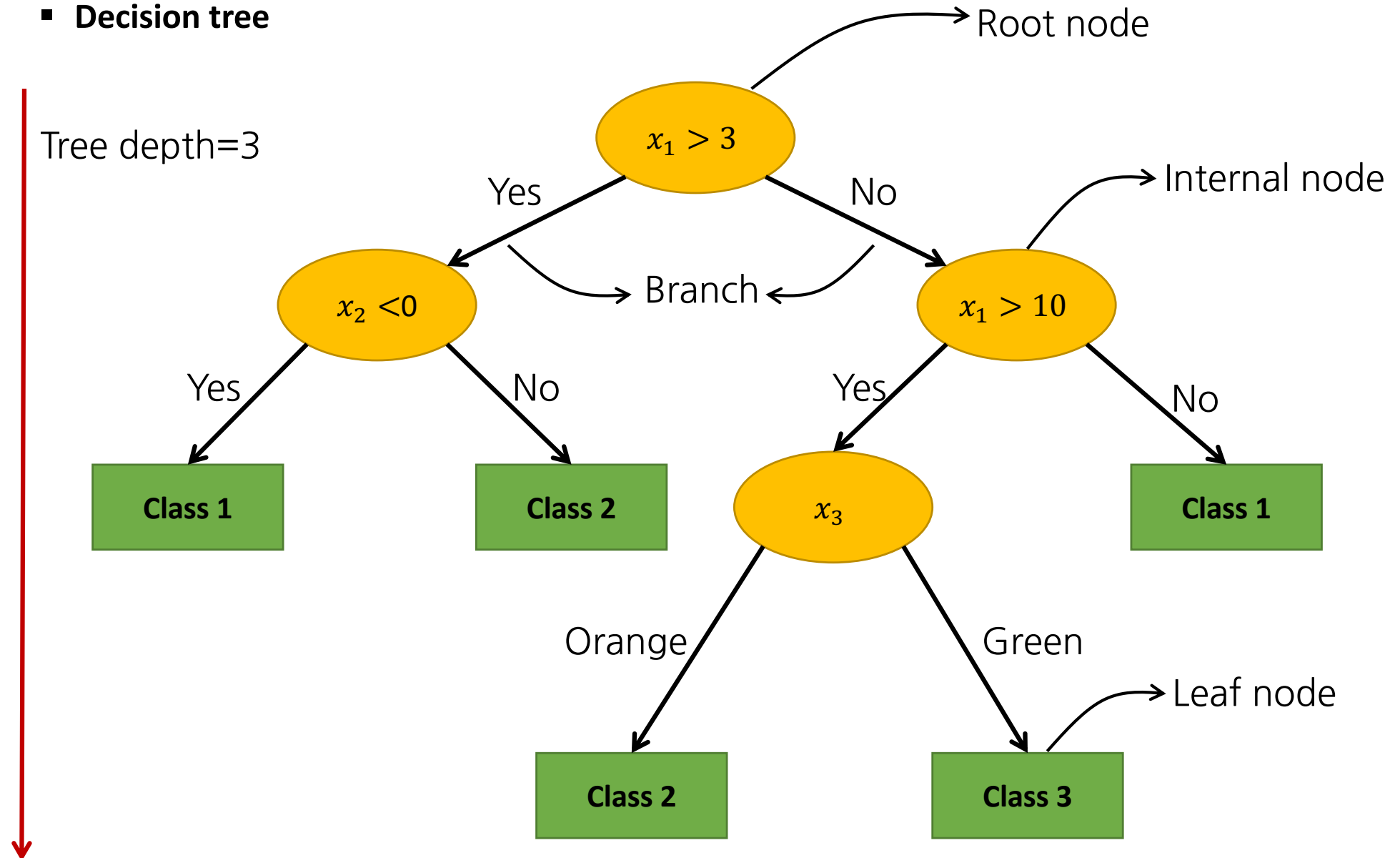


# Decision Tree



# Decision Tree

- Decision tree



# Decision Tree

---

- **Each root node and internal node represent a specific input variable**
  - Root and internal node tests each attribute
    - ✓  $x_1 > 1$
    - ✓  $x_3$  is orange
- **Each branch corresponds to the result of the test of node**
  - Yes vs. No
  - Values of attribute
    - ✓ Orange vs. Green
    - ✓ Long vs. Short
- **Each leaf node assigns a class**

**In each node, how to choose attribute?  
how to split branches?**

---

# Decision Tree Algorithm

---

**How to determine which one is the most effective?**



**Need some criteria for measure of effectiveness**

---

# Splitting Criteria

---

- **Categorical target - Classification**

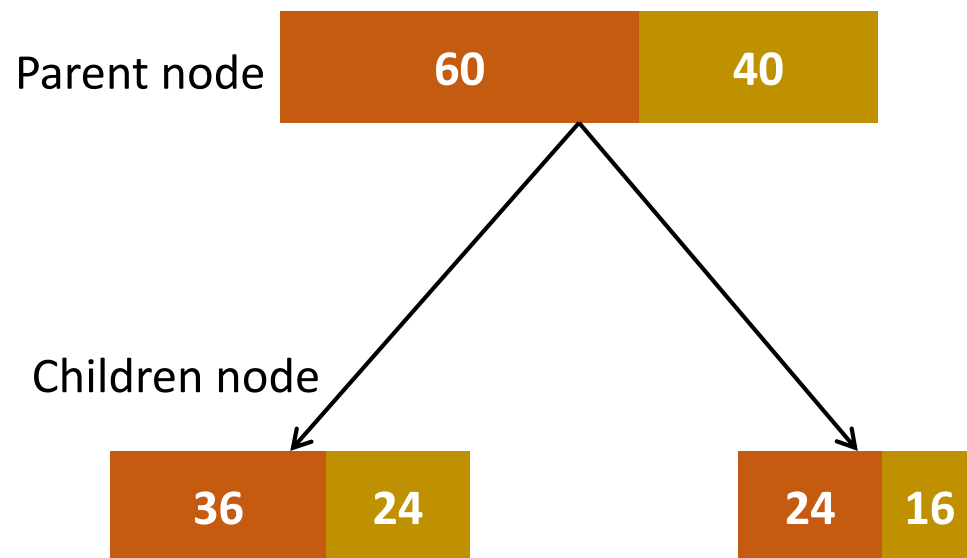
- Entropy
- Gini impurity

- **Continuous target – Regression**

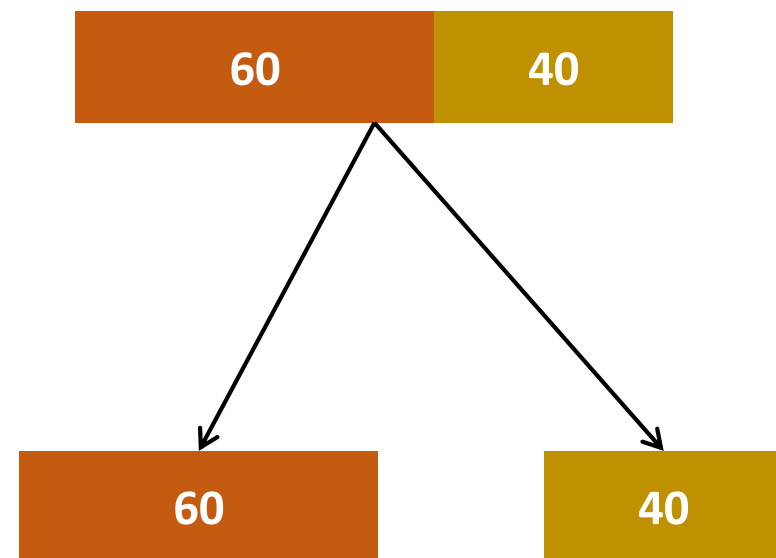
- MSE
- Friedman MSE
- MAE

# Splitting Criteria: Impurity

- Select each split of a node so that in each of the child nodes are less impure than that in the parent node



**No Improvement on purity**



**Perfect Split**

- Entropy and Gini impurity are measures of impurity
  - Split a node toward decreasing impurity → Maximize reduction in impurity

# Entropy

- **Expected value of the information**

- The entropy quantifies how “informative” or “surprising” the entire random variable, averaged on all its possible outcomes

- **Entropy  $H$  of event  $X$**

$$H(X) = E[I(X)] = E[-\ln P(X)]$$

Information content of  $X$

↑                      ↑

Expectation value      Probability function

- **For discrete random variable  $X$**

$$H(X) = \sum_i P(x_i) I(x_i) = - \sum_i P(x_i) \log_b P(x_i)$$

- $X$  with possible values of  $\{x_1, x_2, \dots, x_n\}$
- Commonly  $b$  is 2 (10,  $e$  are also used)



# Example: Entropy

- **When you flip one coin( $X$ )**

- Possible output of  $X$ : H or T
- $P(H) = 0.5, P(T) = 0.5$

$$\begin{aligned} H(X) &= -P(H) \log_2 P(H) - P(T) \log_2 P(T) \\ &= -0.5 \log_2 0.5 - 0.5 \log_2 0.5 \\ &= 1 \end{aligned}$$

- When you flip two coins  $X$

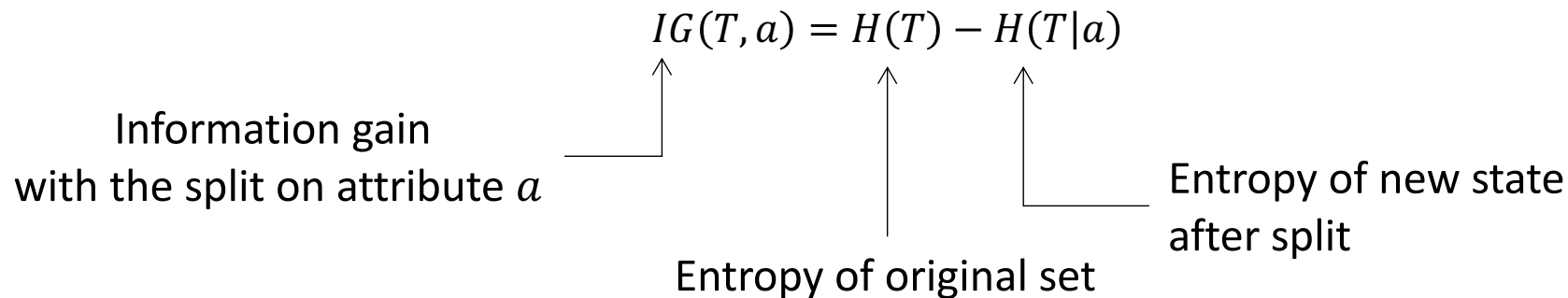
- ✓ Possible output of  $X$ :  $(H, H), (H, T), (T, H), (T, T)$
- ✓  $P(H, H) = P(H, T) = P(T, H) = P(T, T) = 0.25$

$$\begin{aligned} H(X) &= -P(H, H) \log_2 P(H, H) - P(H, T) \log_2 P(H, T) - P(T, H) \log_2 P(T, H) - P(T, T) \log_2 P(T, T) \\ &= -4 \times 0.25 \log_2 0.25 \\ &= 2 \end{aligned}$$

# How to Define Effectiveness of Split

- If split is effective, information gain is large

- Information gain = reduction of uncertainty

$$IG(T, a) = H(T) - H(T|a)$$


Information gain  
with the split on attribute  $a$

Entropy of original set

Entropy of new state  
after split

- Entropy of new state after split = normalized sum of entropy of split sets

$$H(T|a) = \sum_{i=1}^n \frac{|T'_i|}{|T|} \cdot H(T'_i)$$

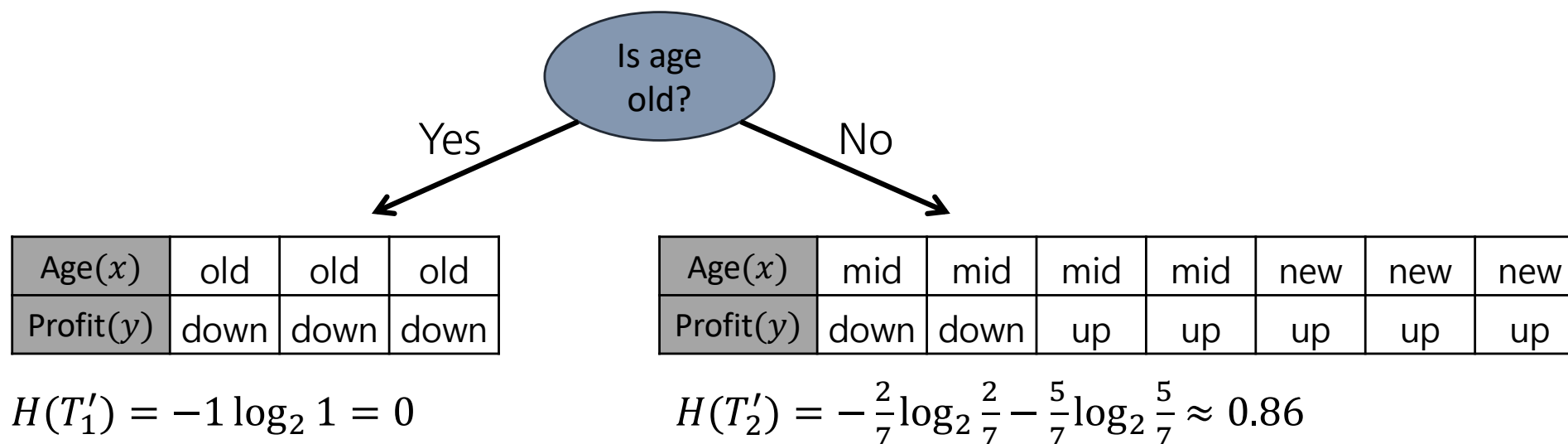
✓  $T$  is split to  $T'_1, T'_2, \dots, T'_n$

# Example: Calculate Information Gain through Entropy

- The node is split by age to predict profit of company

Age(x)	old	old	old	mid	mid	mid	mid	new	new	new
Profit(y)	down	down	down	down	down	up	up	up	up	up

$$\begin{aligned} H(T) &= -P(\text{down}) \log_2 P(\text{down}) - P(\text{up}) \log_2 P(\text{up}) \\ &= -2 \times 0.5 \log_2 0.5 = 1 \end{aligned}$$



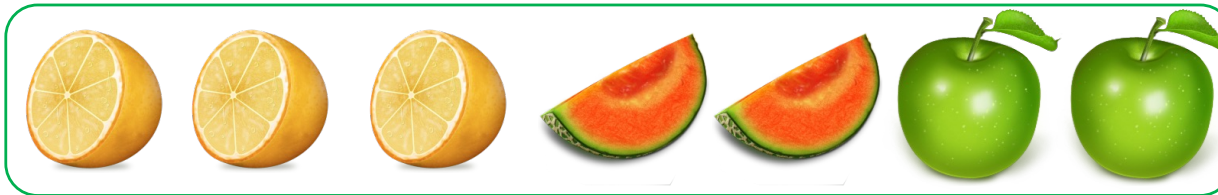
$$IG = H(\text{before}) - H(\text{after}) = 1 - 0.86 \times \frac{7}{10} = 1 - 0.602 = 0.398$$

# Gini Impurity

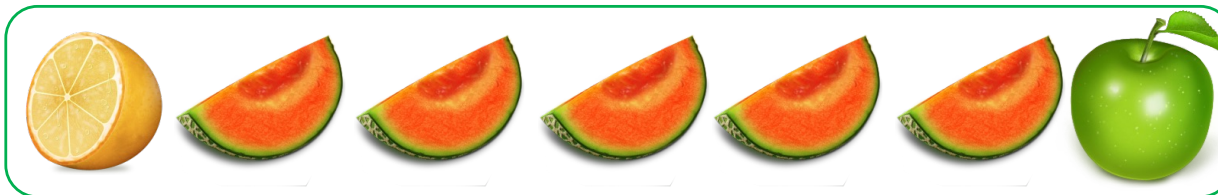
- **Gini impurity: measure of impurity**

$$G(T) = \sum_{i \neq j} P(i|T)P(j|T) = 1 - \sum_j P(j|T)^2 = 1 - \sum_j \left( \frac{n_j(T)}{n(T)} \right)^2$$

- $P(j|t)$  is the probability of output  $j$  in node  $T$
- $n(t)$  is the total number of samples in node  $T$
- $n_j(t)$  is the number of samples with output  $j$  in node  $T$



$$G = 1 - \left( \frac{3}{7} \right)^2 - \left( \frac{2}{7} \right)^2 - \left( \frac{2}{7} \right)^2 = \frac{32}{49}$$



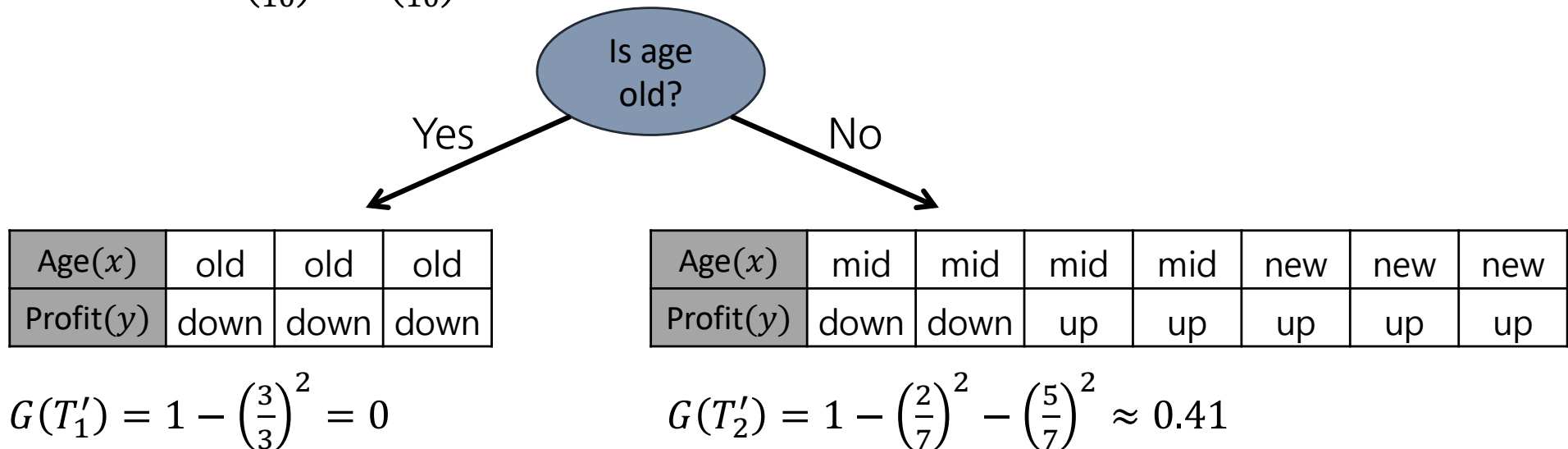
$$G = 1 - \left( \frac{1}{7} \right)^2 - \left( \frac{5}{7} \right)^2 - \left( \frac{1}{7} \right)^2 = \frac{22}{49}$$

## Example: Calculate Information Gain through Gini Impurity

- The node is split by age to predict profit of company

Age(x)	old	old	old	mid	mid	mid	mid	new	new	new
Profit(y)	down	down	down	down	down	up	up	up	up	up

$$\begin{aligned} G(T) &= 1 - P^2(down) - P^2(up) \\ &= 1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2 = 0.5 \end{aligned}$$



$$IG = G(\text{before}) - G(\text{after}) = 0.5 - 0.41 \times \frac{7}{10} = 0.5 - 0.287 = 0.213$$

# Question

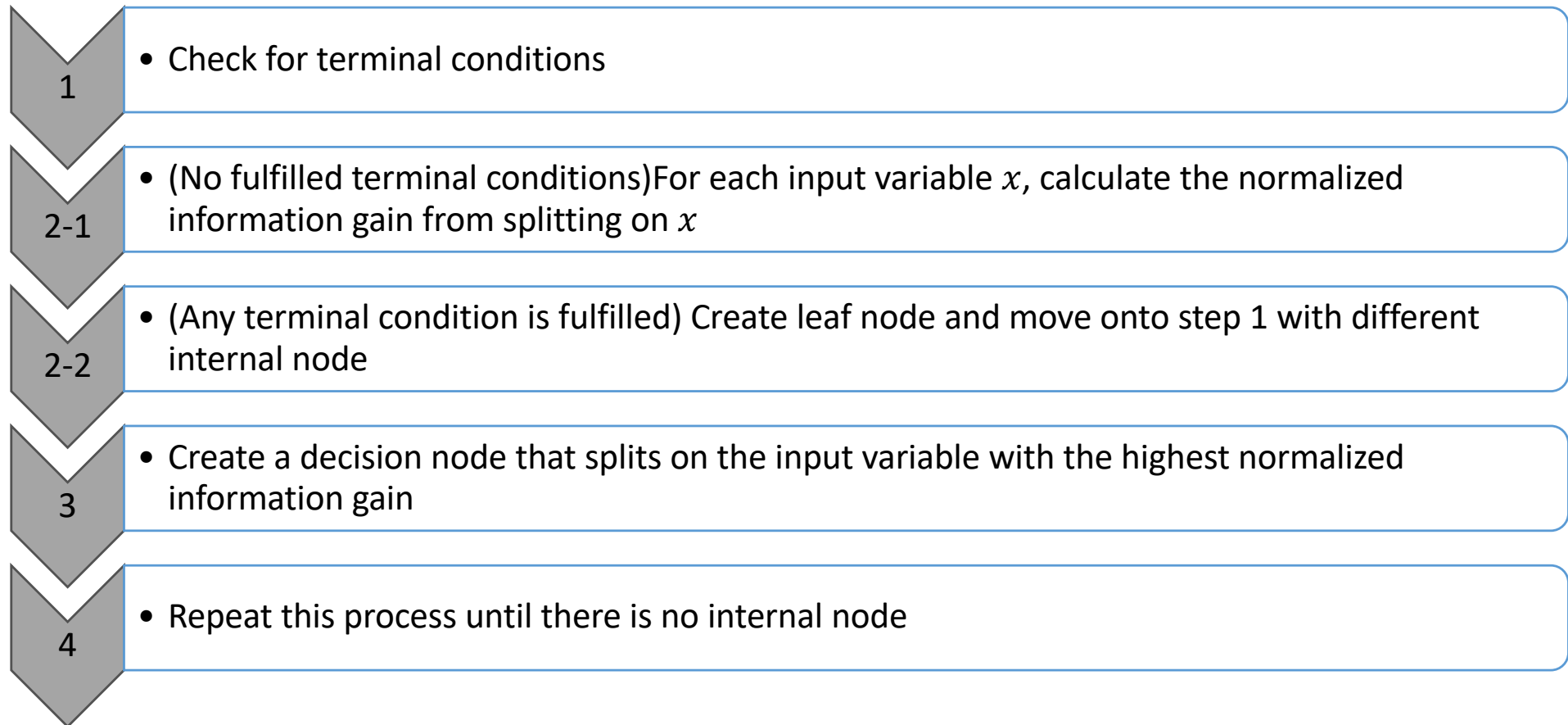
- Predict profit of companies based on age of company, type of company, and competition status

Age	Competition	Type	Profit
old	yes	S/W	down
old	no	S/W	down
old	no	H/W	down
mid	yes	S/W	down
mid	yes	H/W	down
mid	no	H/W	up
mid	no	S/W	up
new	yes	S/W	up
new	no	H/W	up
new	no	S/W	up

- 1) How much information gain based on Entropy is obtained with the splitting on competition?
- 2) How much information gain based on Gini impurity is obtained with the splitting on type of company?

# Decision Tree Algorithm: C4.5

## ■ Procedure of C4.5

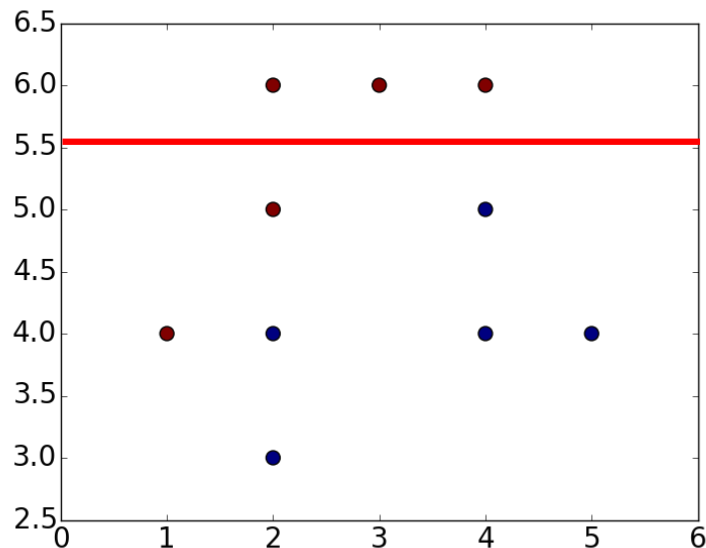


### Terminal conditions

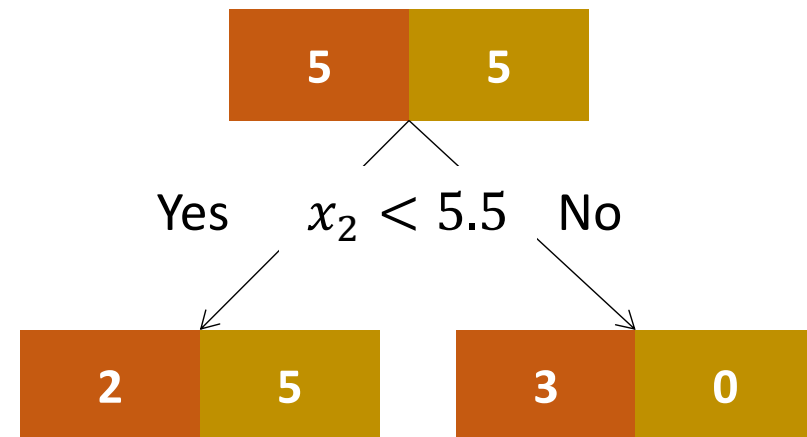
1. All the samples in the list belong to the same class
2. None of the features provide any information gain

# Simple Example for Tree

Class	$x_1$	$x_2$
1	1	4
1	2	6
1	2	5
0	2	4
0	2	3
1	3	6
1	4	6
0	4	5
0	4	4
0	5	4



$$G = 1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2 = 0.5$$



$$G = 1 - \left(\frac{2}{7}\right)^2 - \left(\frac{5}{7}\right)^2 \approx 0.4083$$

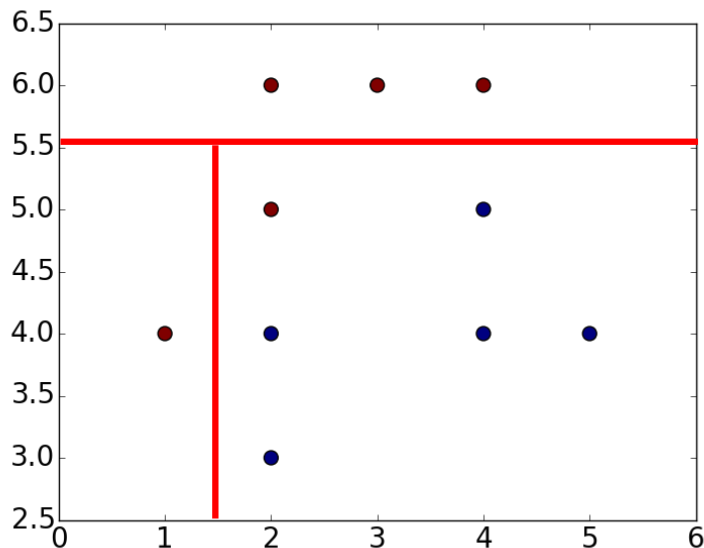
$$G = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

$$IG = 0.5 - 0.408 \times \frac{7}{10} - 0 \times \frac{3}{10} = 0.2144$$

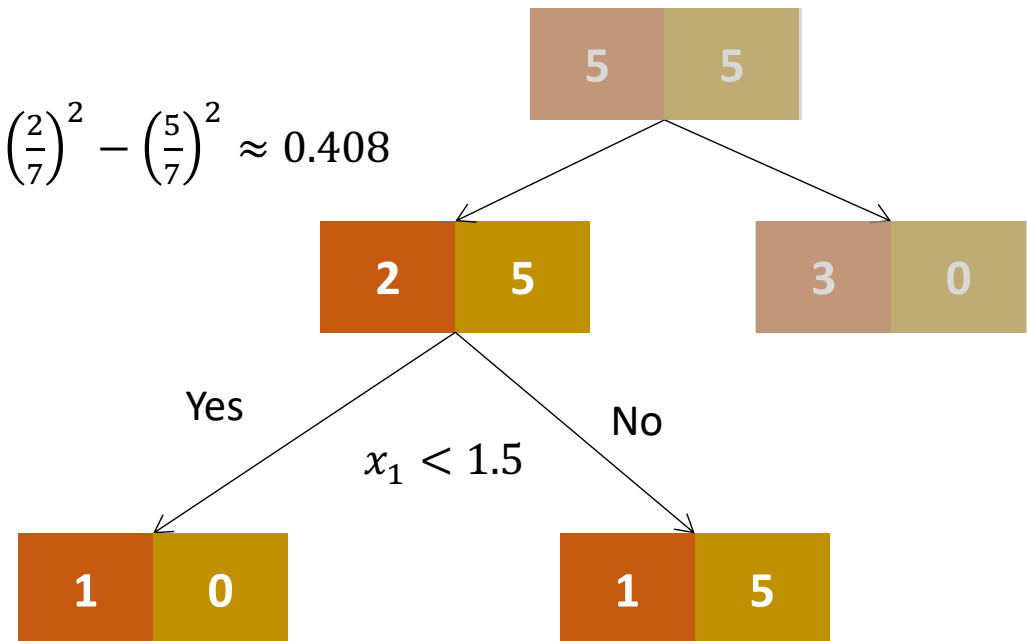


# Simple Example for Tree

Class	$x_1$	$x_2$
1	1	4
1	2	6
1	2	5
0	2	4
0	2	3
1	3	6
1	4	6
0	4	5
0	4	4
0	5	4



$$G = 1 - \left(\frac{2}{7}\right)^2 - \left(\frac{5}{7}\right)^2 \approx 0.408$$



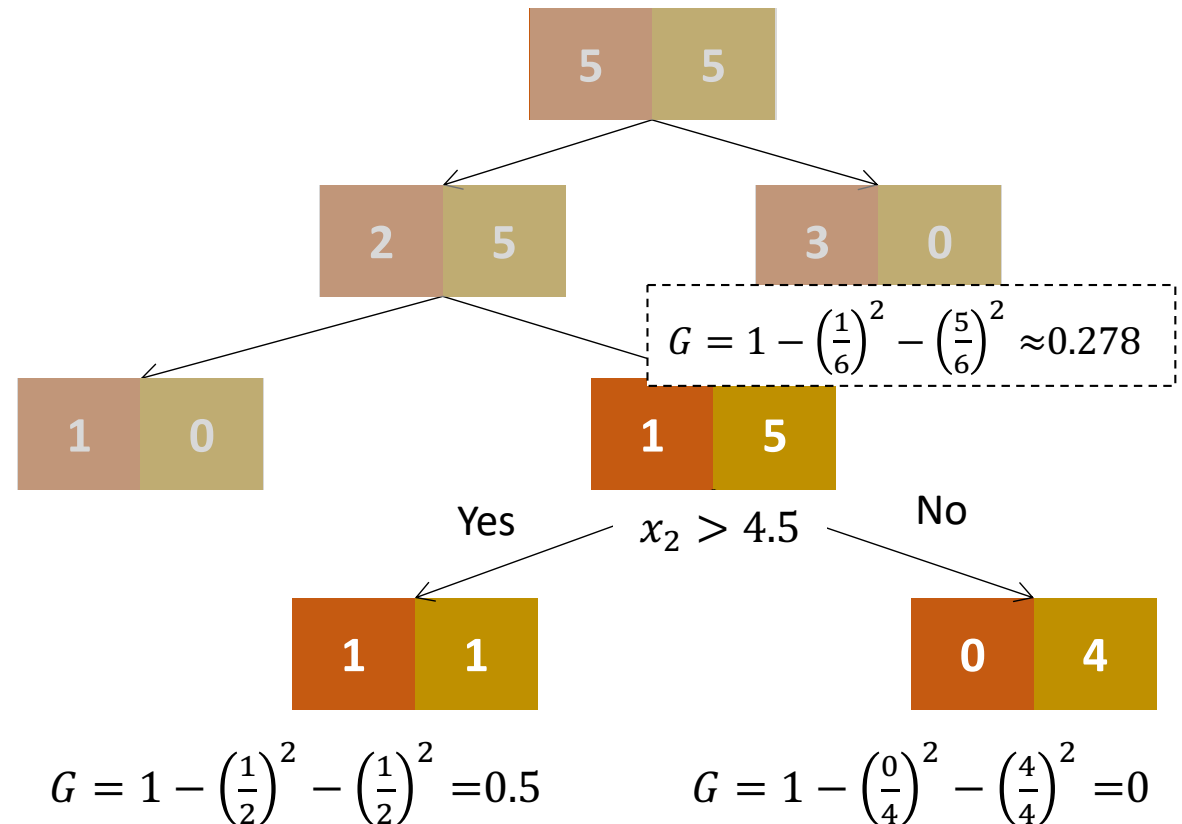
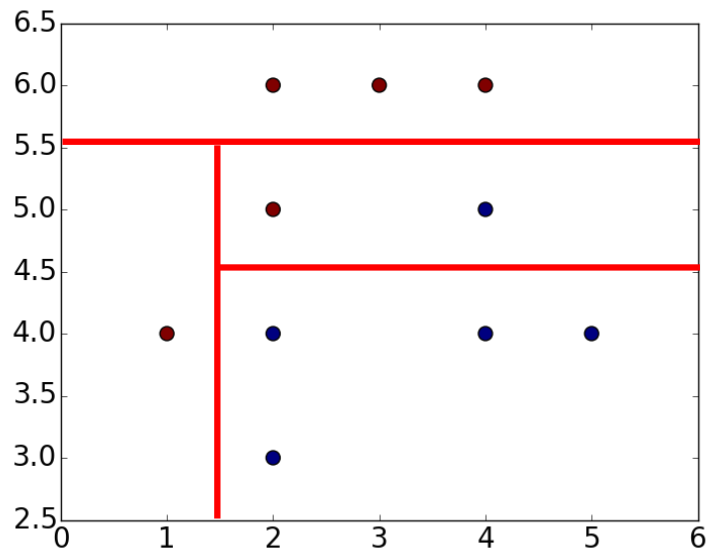
$$G = 1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

$$G = 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 \approx 0.278$$

$$IG = \left(0.408 - 0 \times \frac{1}{7} - 0.278 \times \frac{6}{7}\right) \times \frac{7}{10} \approx 0.11$$

# Simple Example for Tree

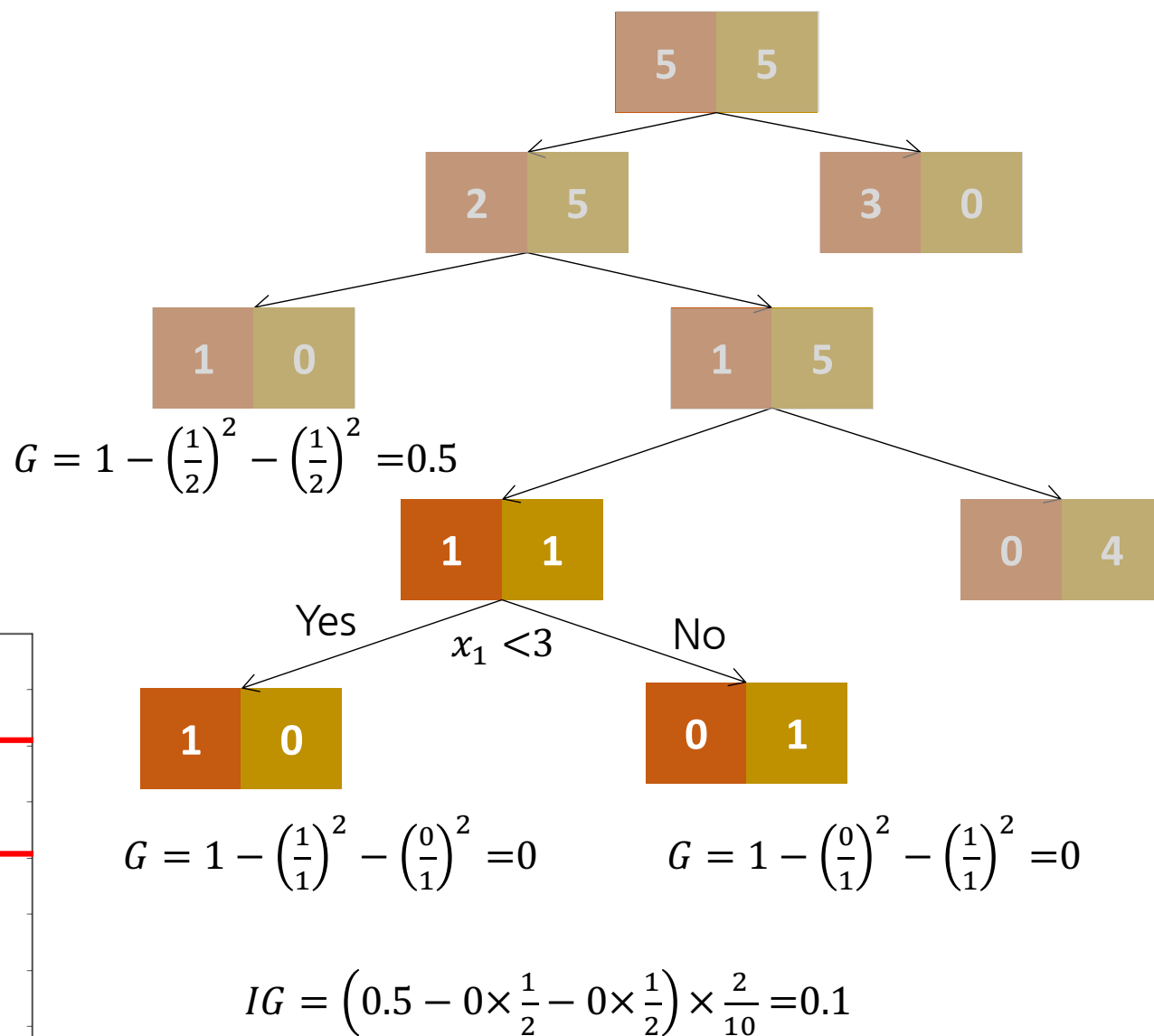
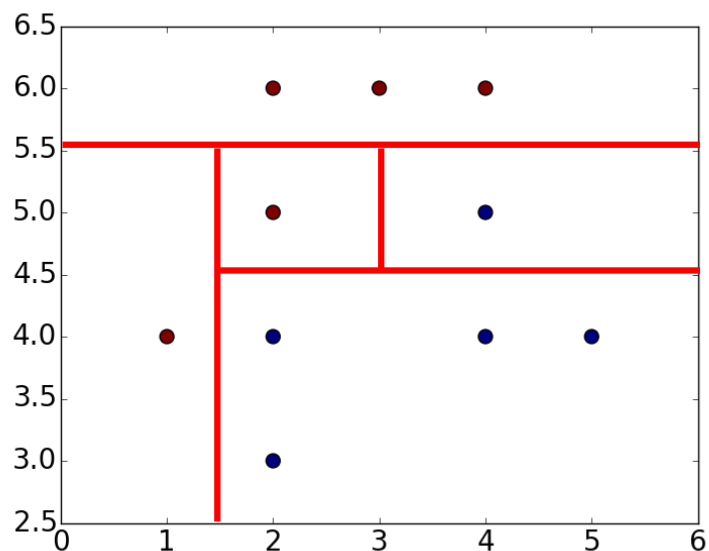
Class	$x_1$	$x_2$
1	1	4
1	2	6
1	2	5
0	2	4
0	2	3
1	3	6
1	4	6
0	4	5
0	4	4
0	5	4



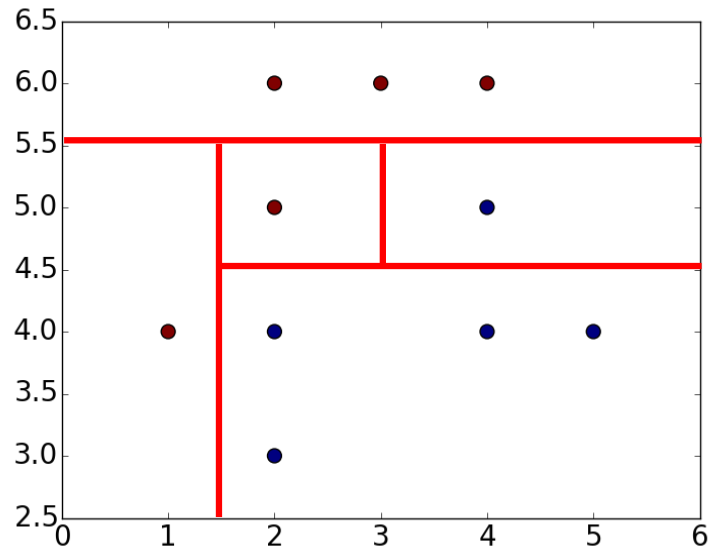
$$IG = \left(0.278 - 0.5 \times \frac{2}{6} - 0 \times \frac{4}{6}\right) \times \frac{6}{10} \approx 0.067$$

# Simple Example for Tree

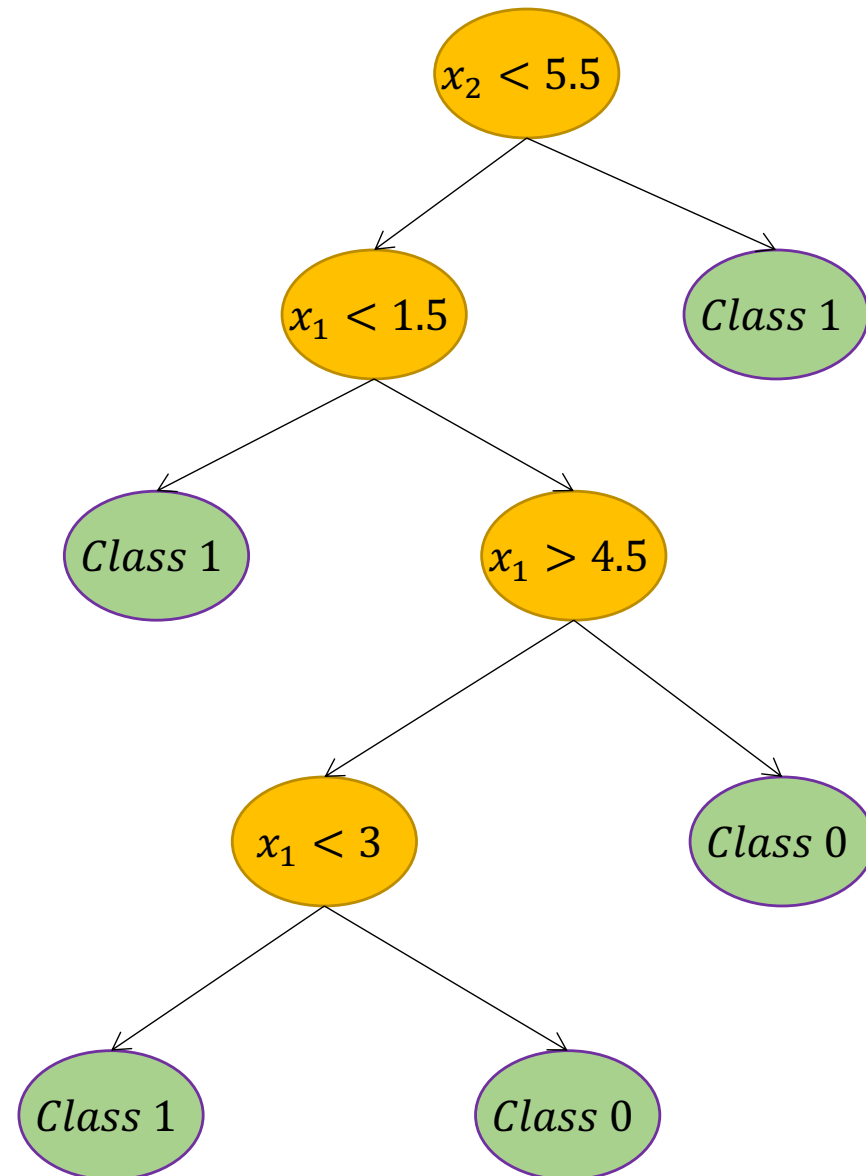
Class	$x_1$	$x_2$
1	1	4
1	2	6
1	2	5
0	2	4
0	2	3
1	3	6
1	4	6
0	4	5
0	4	4
0	5	4



# Simple Example for Tree



Step	Impurity Change	Total impurity
0	0.000	0.500
1	0.214	0.286
2	0.119	0.167
3	0.067	0.100
4	0.100	0.000



# What Does Tree Stop Growing?

---

- **Growing full-size tree can cause overfitting**
  - Low classification accuracy on the test set
- **Introduce pruning step after growing tree**
  - Pruning simplifies the tree by trimming some branches of the fully grown tree
  - Generate several pruned trees and select best tree
- **There are several popular pruning algorithms**
  - Reduced error pruning: Starting at leaf node, each node is replaced with its most popular class and if the prediction accuracy is not affected then the change is kept
  - Cost complexity pruning: Cost complexity pruning removes subtree based on cost complexity measure at each step

# Cost Complexity Pruning

## ▪ Misclassification cost at node $t$

$$r(t) = \min_i \sum_{k=1}^K C(i|k)p(k|t) \rightarrow r(t) = 1 - \max_k p(k|t)$$

- $C(i|k) = \begin{cases} 1, & \text{if } i \neq k \\ 0, & \text{if } i = k \end{cases}$  is the cost of classifying a sample of class  $k$  into class  $i$ .
- $p(k|t)$  is the probability that the class of data point is  $k$  given that it is in node  $t$

## ▪ Misclassification cost at tree $T$

$$R(T) = \sum_{t \in \tilde{T}} r(t)p(t) = \sum_{t \in \tilde{T}} R(t)$$

- $\tilde{T}$  is set of terminal nodes of tree  $T$
- $p(t)$  is probability that data point is in node  $t$
- Set  $R(t) = r(t)p(t)$

# Cost Complexity Pruning

- **Cost complexity measure**

Complexity parameter

$$R_{\alpha}(T) = R(T) + \alpha|T|$$

- $|T|$  is tree complexity = the number of terminal nodes

- For a terminal node  $t$

$$R_{\alpha}(t) = R(t) + \alpha$$

- For a subtree at  $t$

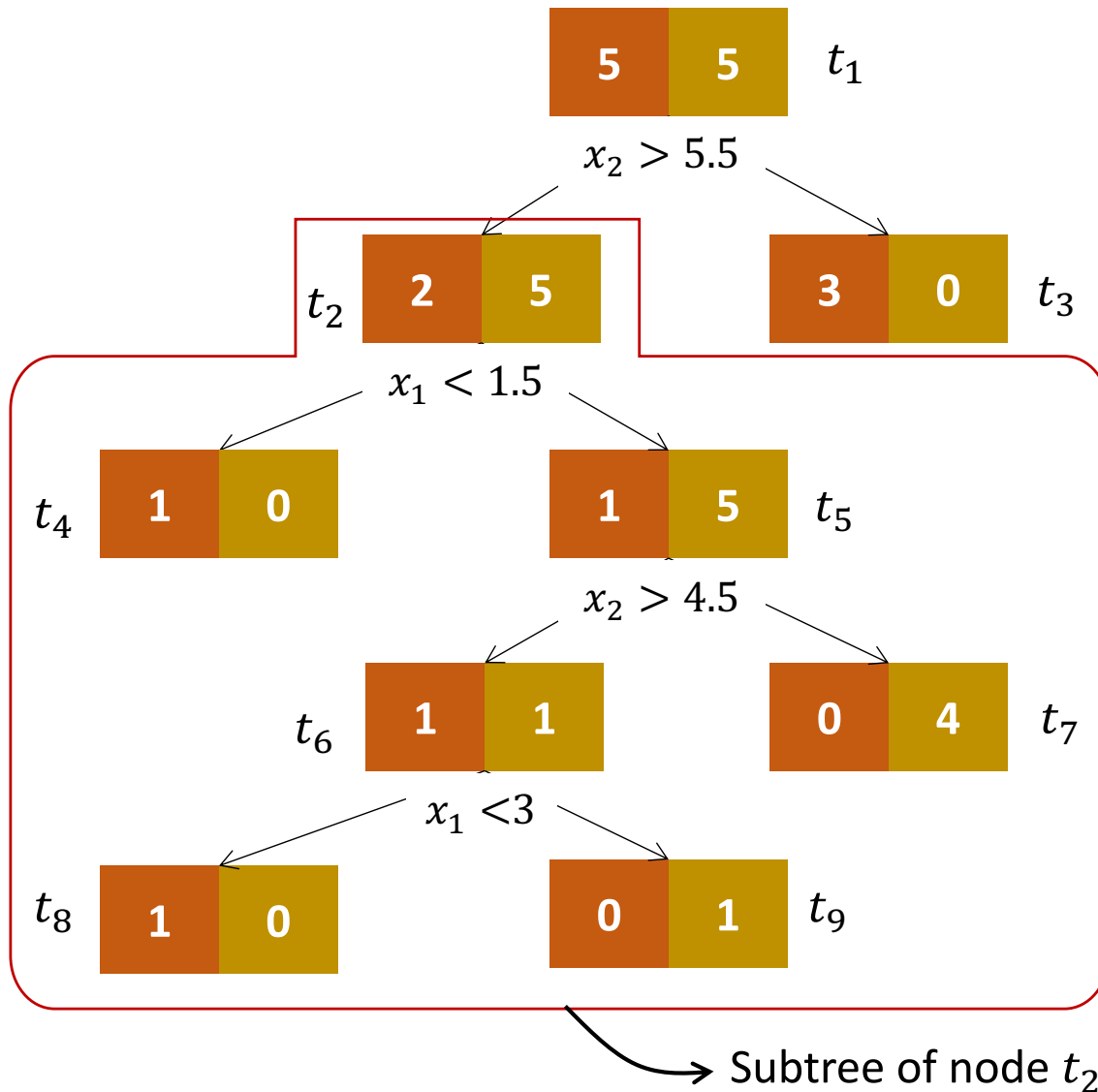
$$R_{\alpha}(T_t) = R(T_t) + \alpha|T_t|$$

- **Cost complexity pruning prunes subtree at  $t$  which minimizes**

$$\alpha(t) = \frac{R(t) - R(T_t)}{|T_t| - 1}$$

- ~~Prunes subtree if  $R_{\alpha}(t) \leq R_{\alpha}(T_t)$~~

# Cost Complexity Pruning



At node  $t_2$

$$R(t_2) = r(t_2)p(t_2)$$

$$= \left(1 - \frac{5}{7}\right) \times \frac{7}{10} = 0.2$$

$$|T_{t_2}| = 4$$

$$R(T_{t_2})$$

$$= (1 - 1) \times \frac{1}{10} + (1 - 1) \times \frac{1}{10}$$

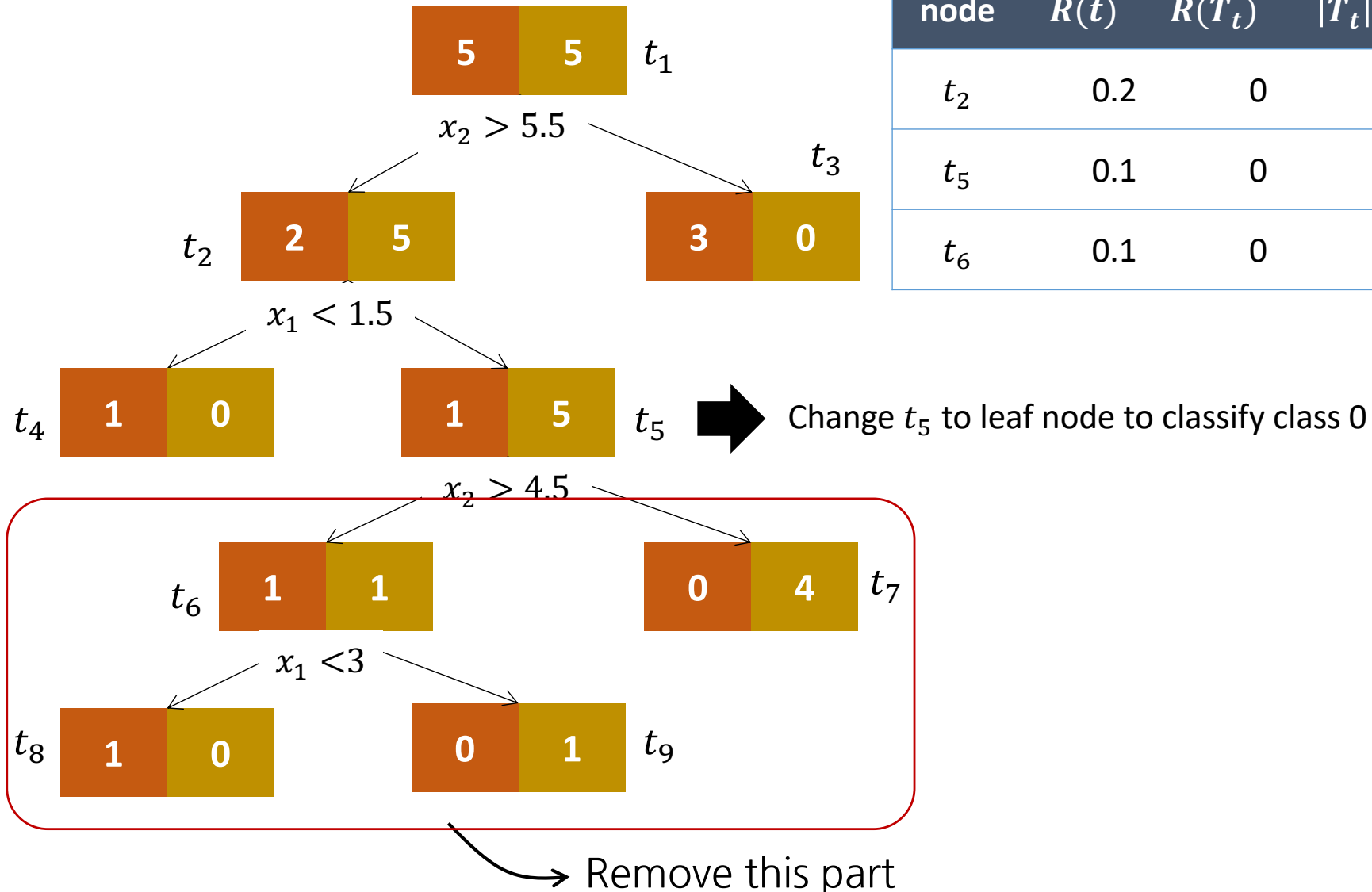
$$+ (1 - 1) \times \frac{1}{10} + (1 - 1) \times \frac{4}{10} = 0$$

$$\alpha(t_2) = \frac{0.2 - 0}{4 - 1} \approx 0.067$$



# Cost Complexity Pruning

node	$R(t)$	$R(T_t)$	$ T_t $	$\alpha(t)$
$t_2$	0.2	0	4	0.067
$t_5$	0.1	0	3	<b>0.050</b>
$t_6$	0.1	0	2	0.100



---

# Pros and Cons of Decision Tree

---

## ■ Pros

- Easy interpretation
- Non-parametric approach
- Inherently non-linear
- Easy to handle categorical variables (also friendly to unnormalized numeric variables)
- Implicitly perform feature selection

## ■ Cons

- Large computing cost
- Lack of linearity of main effects
- Each node only considers single variable
  - ✓ Many algorithms has been proposed to overcome this problem

---

# Regression Tree

---

- **CART**

- Classification And Regression Tree (CART) is one of decision tree algorithms
- Decision tree can be also used for regression analysis

**How?**

# Regression Tree

## ▪ New split rule

- Entropy and Gini impurity are not appropriate split measure for regression analysis
- MSE (Mean squared error)
  - ✓ The split that most decrease the MSE is selected

$$\hat{y}_i = \frac{\sum_{j \in t_i} y_j}{|t_i|}$$

$$R(t_i) = \frac{1}{N_{t_i}} \sum_{j \in t_i} (y_j - \hat{y}_i)^2$$

$$IG = p(t_p)R(t_p) - p(t_r)R(t_r) - p(t_l)R(t_l)$$

- Friedman MSE

$$\hat{y}_i = \frac{\sum_{j \in t_i} y_j}{|t_i|}$$

$$IG = \frac{N_{t_r} N_{t_l}}{N_{t_r} + N_{t_l}} (\hat{y}_{t_r} - \hat{y}_{t_l})^2$$

# Regression Tree

## ▪ New split rule

- MAE (Mean absolute error)

✓ The split that minimizes the L1-loss using the median of each terminal node is selected

$\hat{y}_i$  = the median of each terminal node

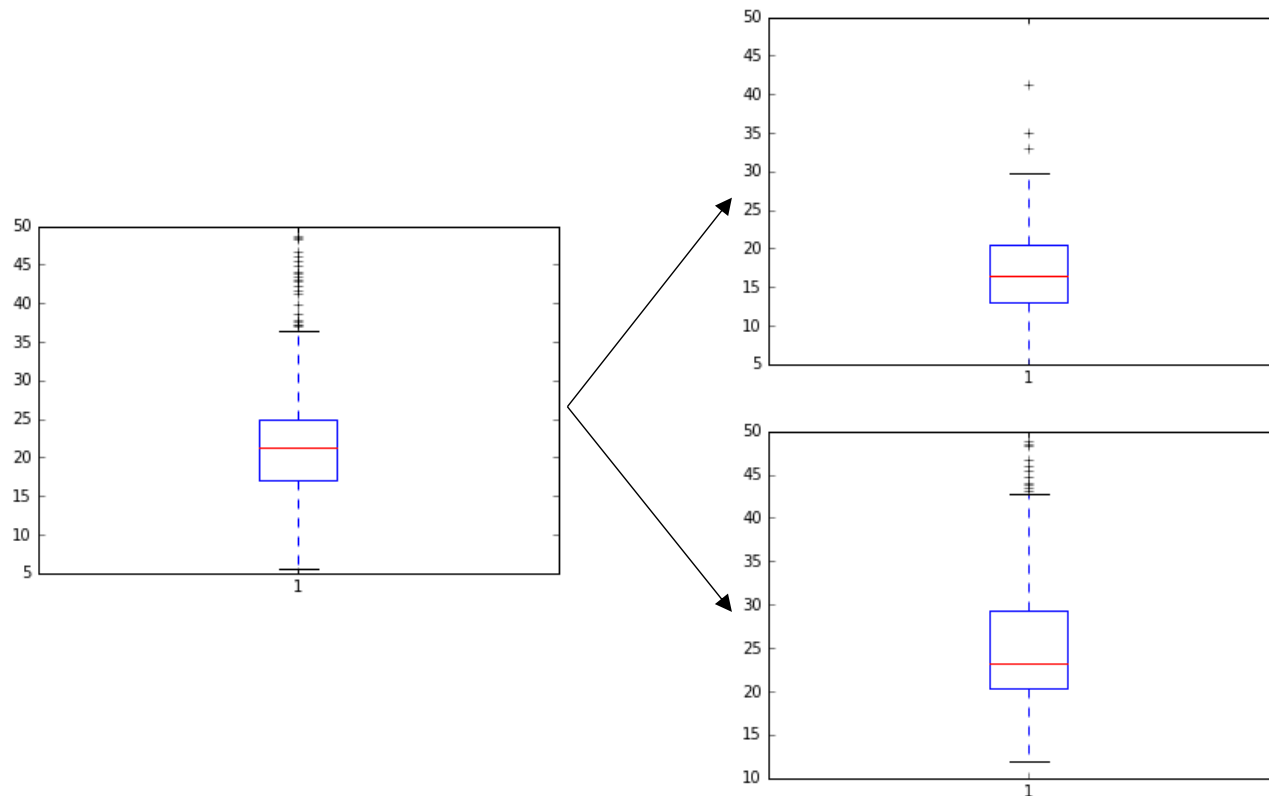
$$R(t_i) = \frac{1}{N_{t_i}} \sum_{j \in t_i} |y_j - \hat{y}_i|$$

$$IG = p(t_p)R(t_p) - p(t_r)R(t_r) - p(t_l)R(t_l)$$

# Regression Tree

## ■ New split rule

- Entropy and Gini impurity are not appropriate split measure for regression analysis
- Use least square deviation (LSD) split rule
  - ✓ After split, mean of output values in child nodes should be significantly different.



# Regression Tree

## ■ New split rule

- Entropy and Gini impurity are not appropriate split measure for regression analysis
- Use least square deviation (LSD) split rule
  - ✓ After split, mean of output values in children nodes should be significantly different

$$\hat{y}_i = \frac{\sum_{j \in t_i} y_j}{|t_i|}$$

$$G(t_i) = \sum_{j \in t_i} (y_j - \hat{y}_i)^2$$

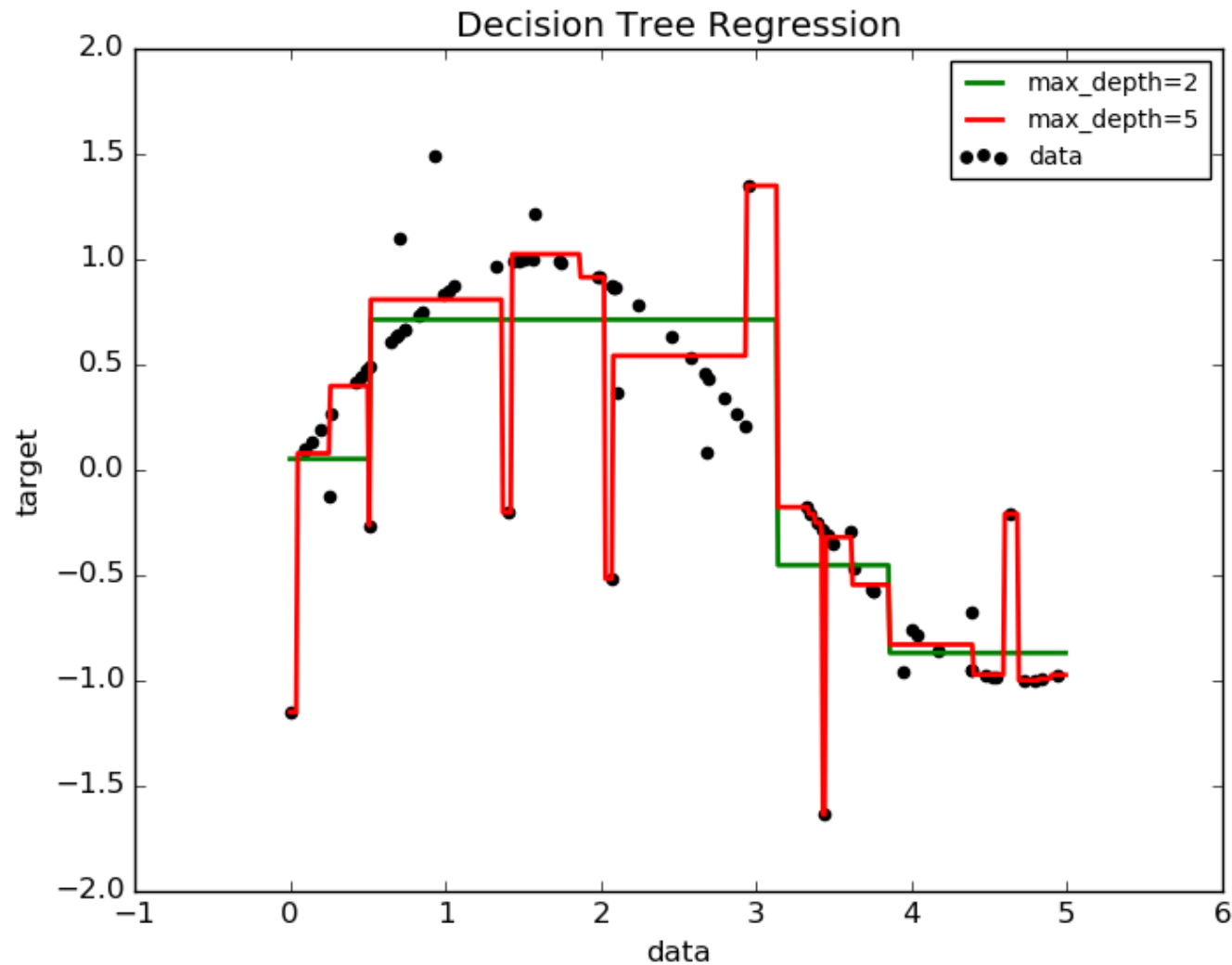
$$IG = G(t_p) - P_r G(t_r) - P_l G(t_l)$$

## ■ Prediction

- Output value is predicted based on output values of training samples at the leaf node
  - ✓ Calculate average

# Regression Tree

- Drawback of regression tree





**Thank you!**