SOON CHUN HYANG UNIVERSITY

Machine Learning

-Decision Tree, Random Forest, Boosting-

SCH Univ.
Dept. of AI and Bigdata
Kim JinSeong

Contents

1. Decision Tree Model Ⅰ

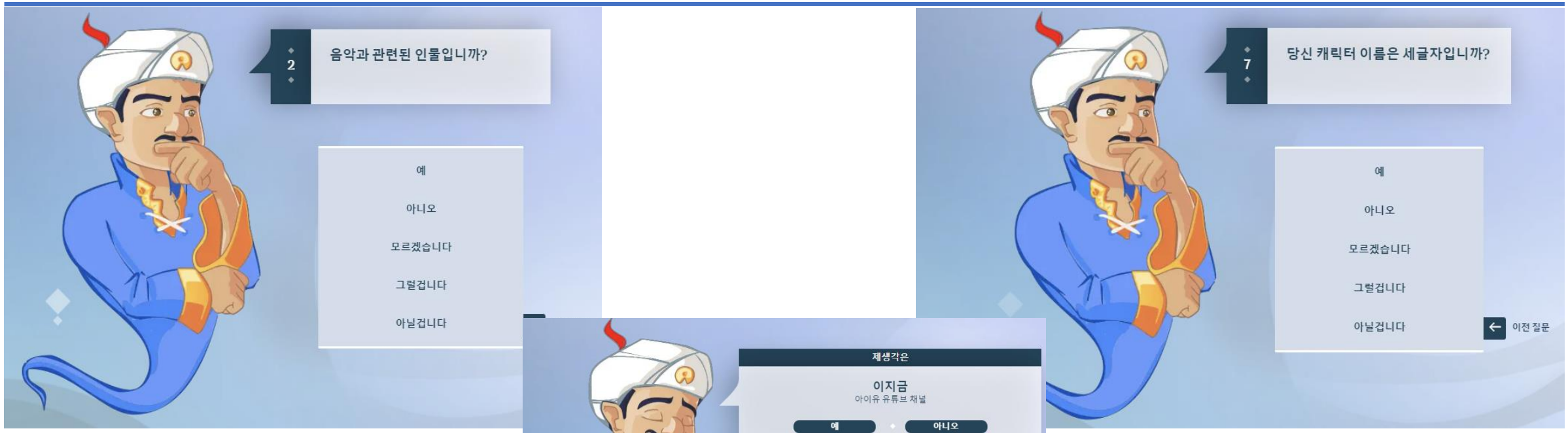
Overview of the Decision Tree Model

- Data에 내제되어 있는 Pattern을 변수의 조합으로 나타내는 예측/분류 모델을 나무의 형태로 만드는 것

Overview of the Decision Tree Model

- Data에 내제되어 있는 Pattern을 변수의 조합으로 나타내는 예측/분류 모델을 나무의 형태로 만드는 것

Overview of the Decision Tree Model



Overview of the Decision Tree Model

Overview of the Decision Tree Model

Data ⇨ Algorithm ⇨ Model(Output)

Input				Output
X_1	X_2	⋯	X_p	Y

Overview of the Decision Tree Model

Data ⇨ Algorithm ⇨ Model(Output)

Input				Output
X_1	X_2	⋯	X_p	Y

Data를 2개 or 그 이상의 부분집합으로 분할

Overview of the Decision Tree Model

Data ⇨ Algorithm ⇨ Model(Output)

Input				Output
X_1	X_2	⋯	X_p	Y

Data를 2개 or 그 이상의 부분집합으로 분할

⬇

Data가 균일해지도록 분할

Overview of the Decision Tree Model

Data ⇨ Algorithm ⇨ Model(Output)

Input				Output
X_1	X_2	⋯	X_p	Y

Data를 2개 or 그 이상의 부분집합으로 분할

↓

Data가 균일해지도록 분할

- 예측
- 분류

Overview of the Decision Tree Model

Data ⇨ Algorithm ⇨ Model(Output)

Input				Output
X_1	X_2	⋯	X_p	Y

Data를 2개 or 그 이상의 부분집합으로 분할

↓

Data가 균일해지도록 분할

- 예측: 비슷한 수치를 가진 관측치끼리 모음
- 회귀

Overview of the Decision Tree Model

Data ⇨ Algorithm ⇨ Model(Output)

Input				Output
X_1	X_2	⋯	X_p	Y

Data를 2개 or 그 이상의 부분집합으로 분할

↓

Data가 균일해지도록 분할

- 예측: 비슷한 수치를 가진 관측치끼리 모음
- 회귀 : 비슷한 범주를 가진 관측치끼리 모음

Overview of the Decision Tree Model

Data ⇨ Algorithm ⇨ Model(Output)

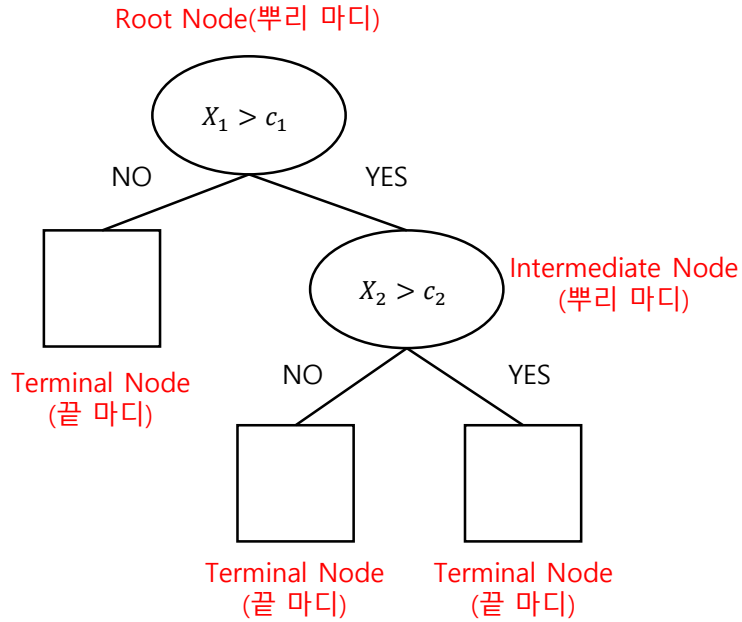
Input				Output
X_1	X_2	⋯	X_p	Y

Data를 2개 or 그 이상의 부분집합으로 분할

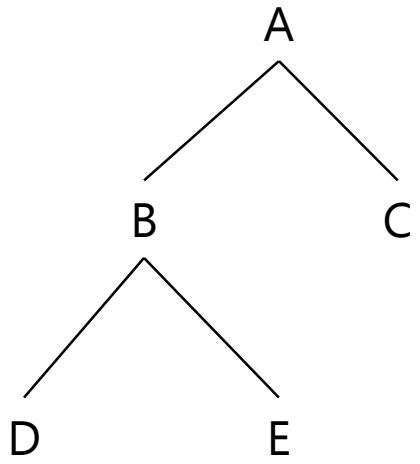
↓

Data가 균일해지도록 분할

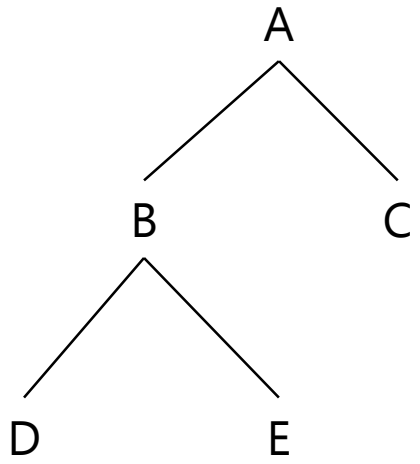
- 예측: 비슷한 수치를 가진 관측치끼리 모음
- 회귀 : 비슷한 범주를 가진 관측치끼리 모음

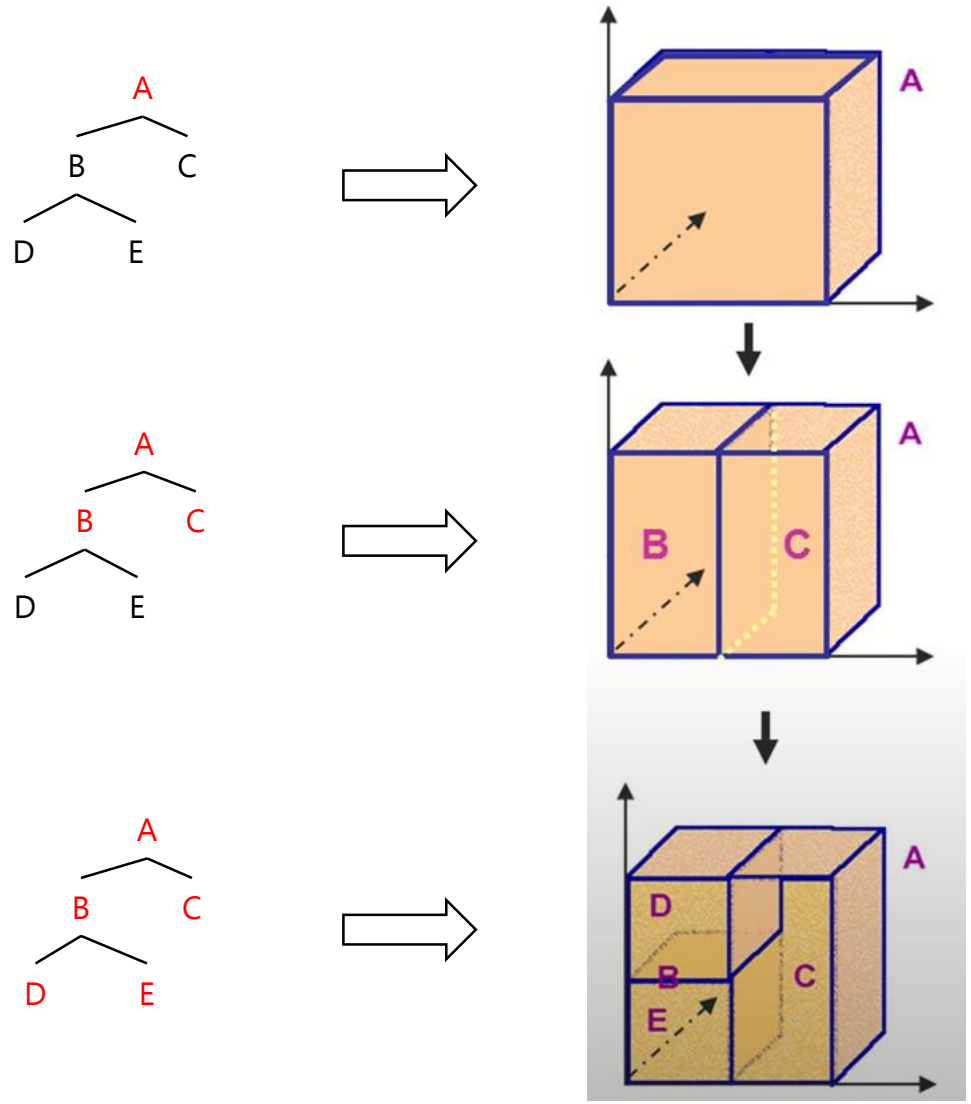


Binary Division

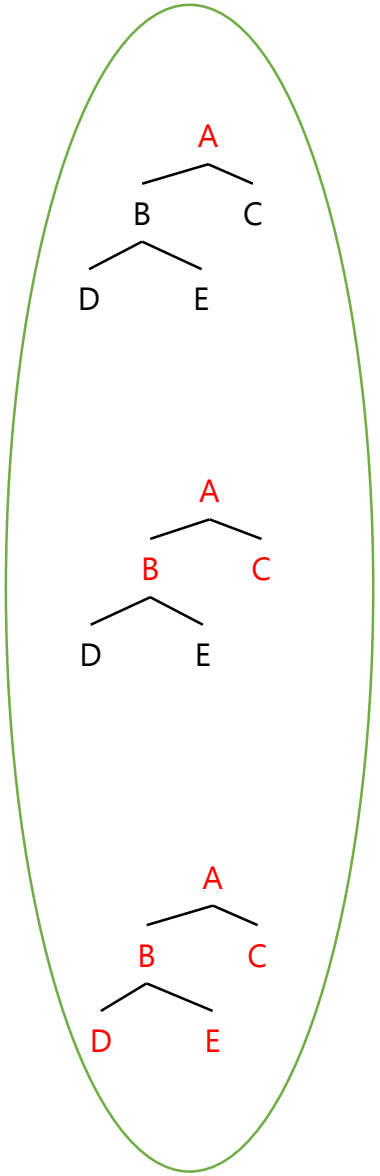


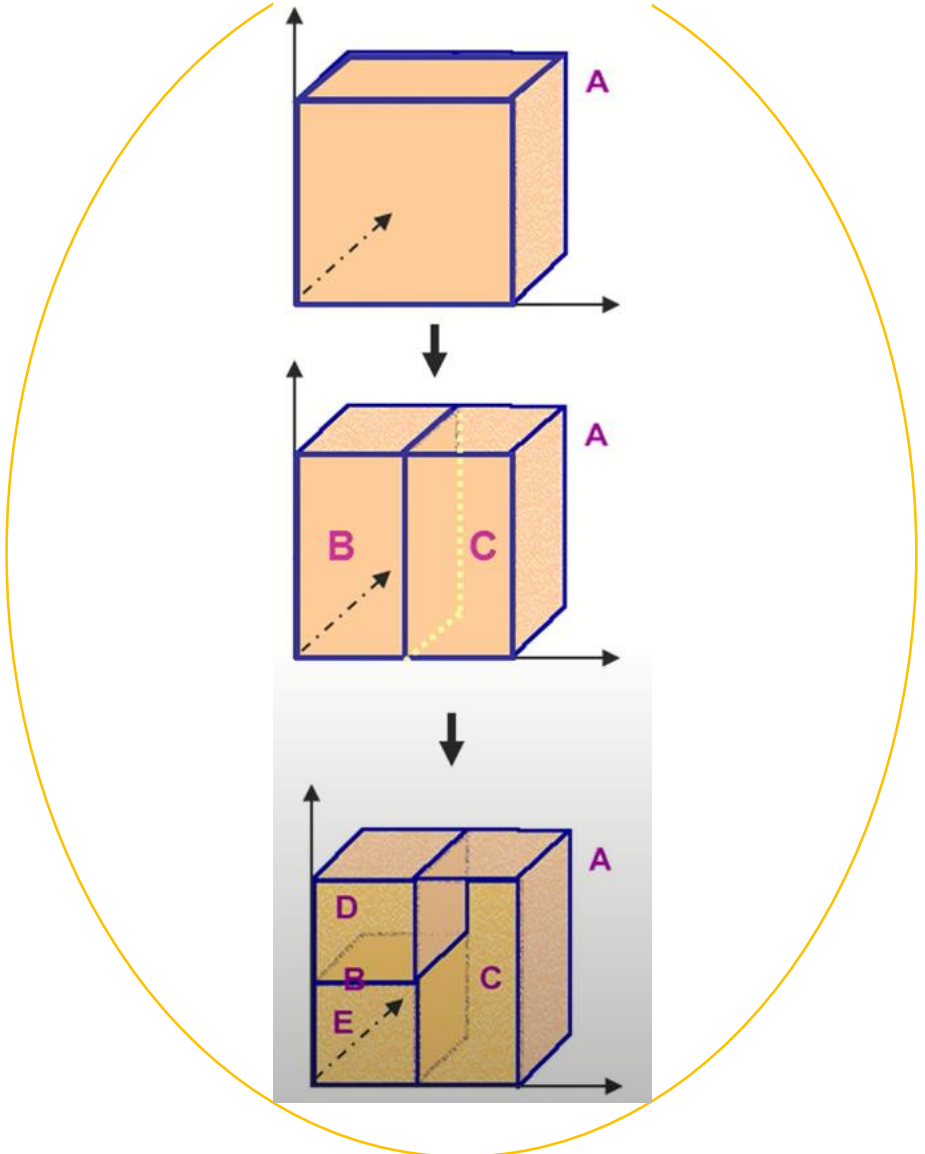
Binary Division





Binary Division

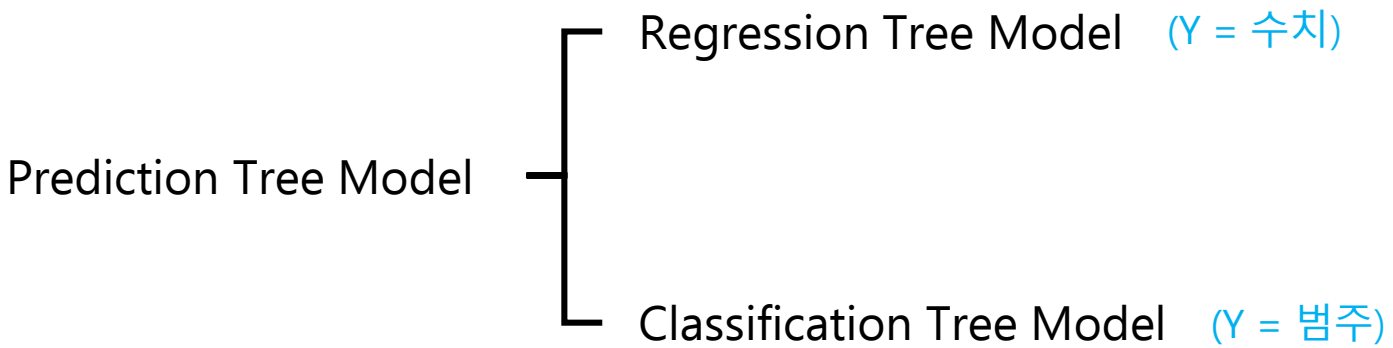




- 고차원도 표현 가능
- 더 많이 쓰이는 표현

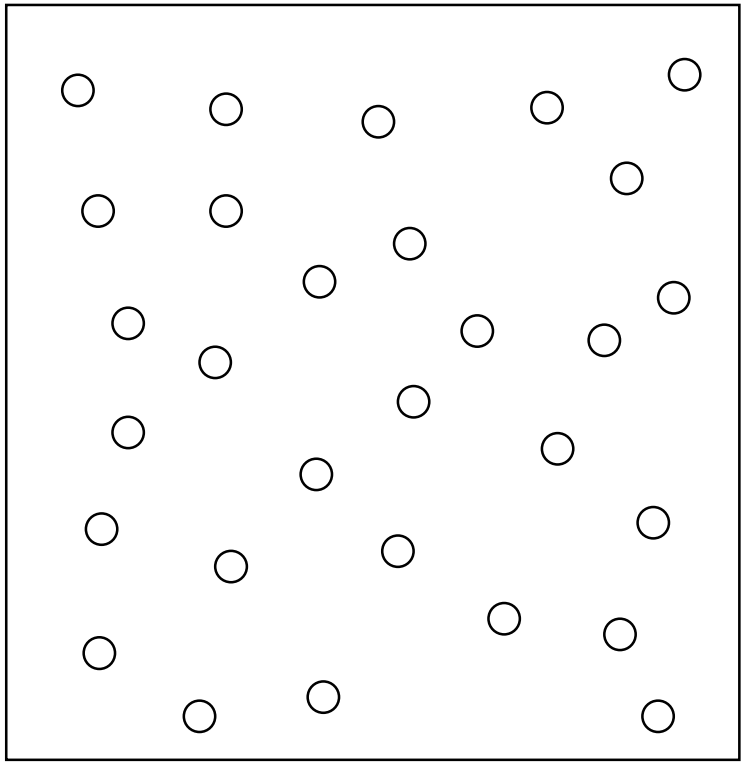
- 고차원으로 갈수록 표현이 힘듦

Prediction Tree Model



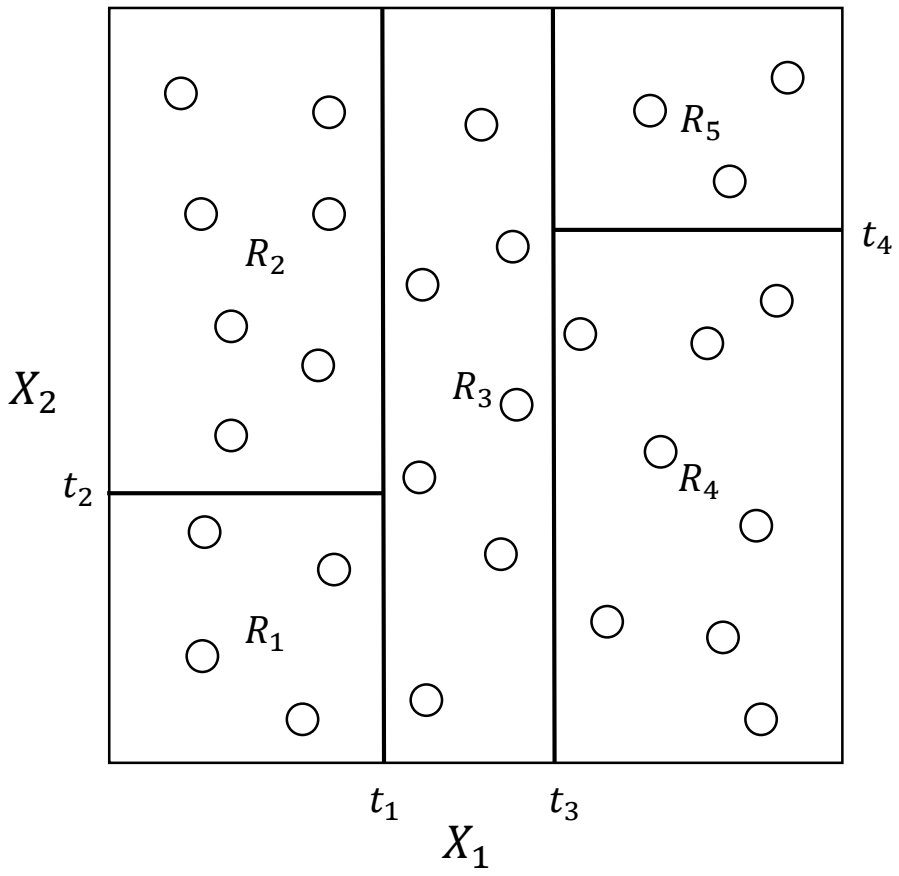
Regression Tree Model

Y = 수치



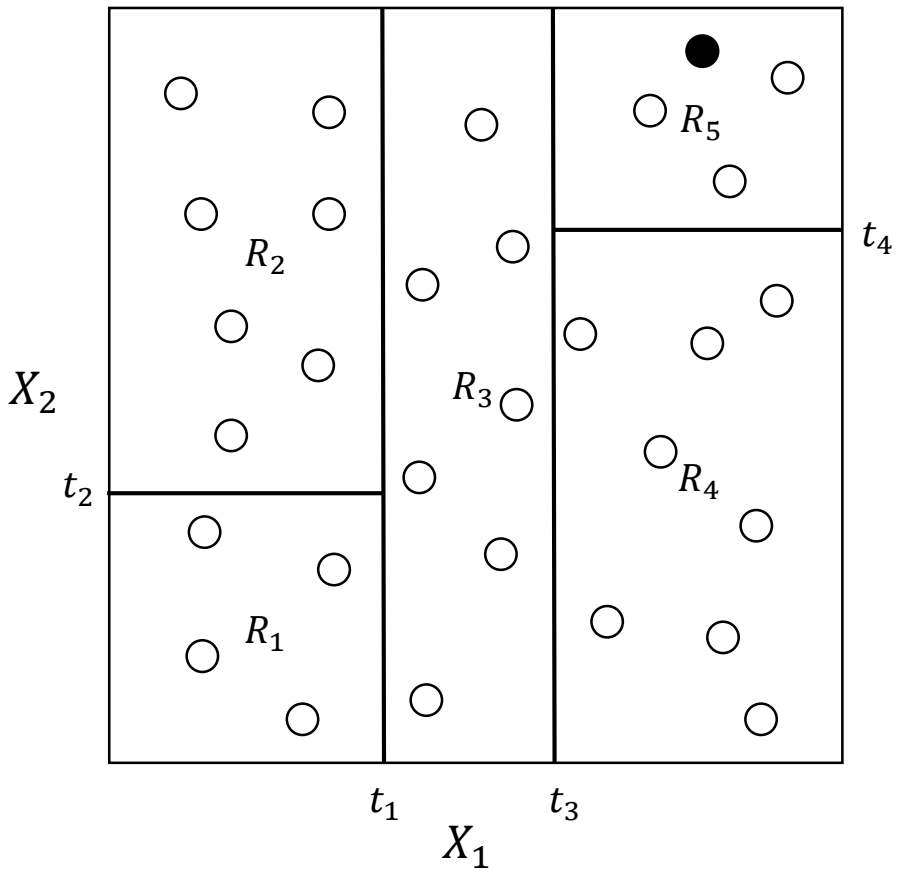
Regression Tree Model

Y = 수치

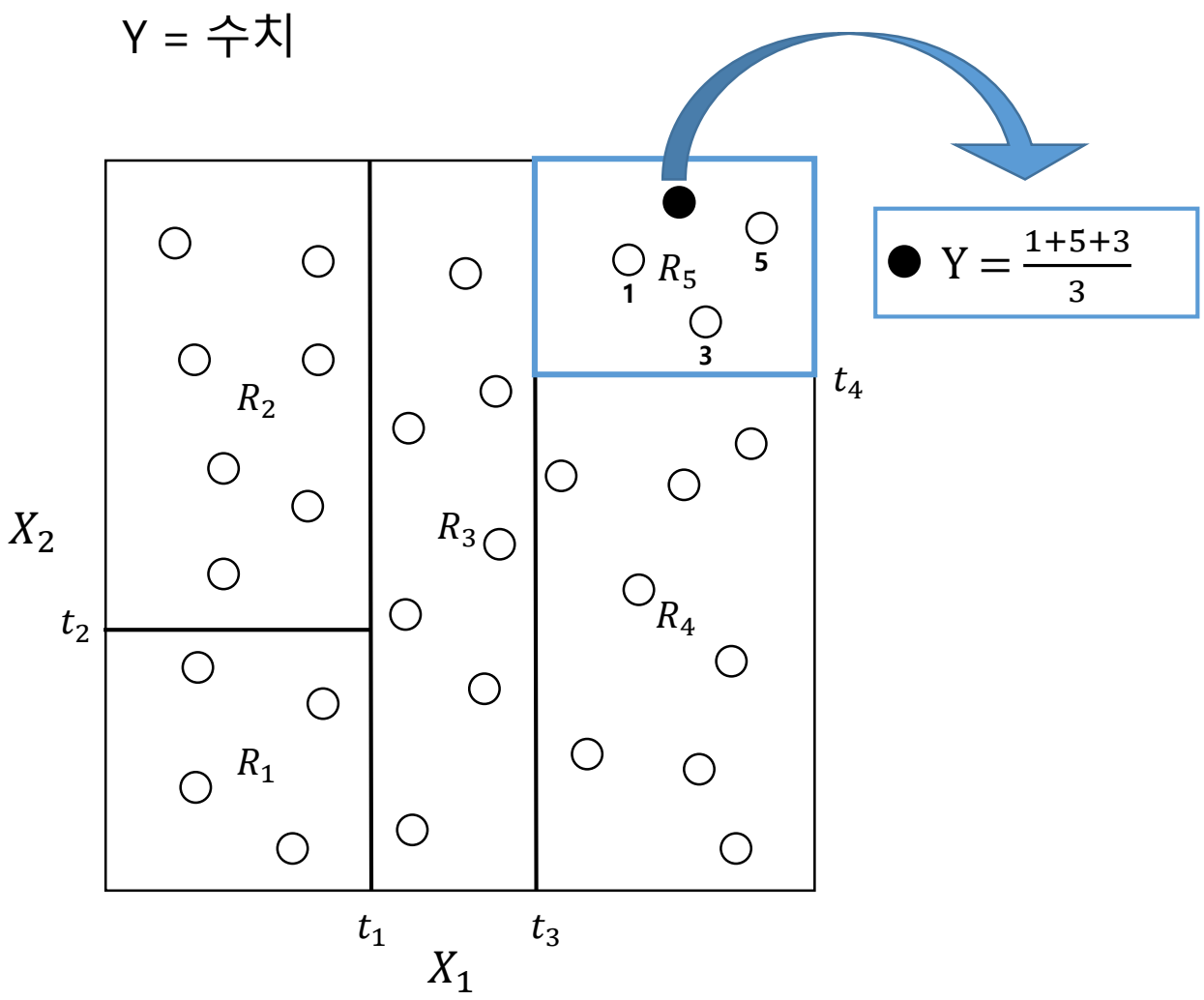


Regression Tree Model

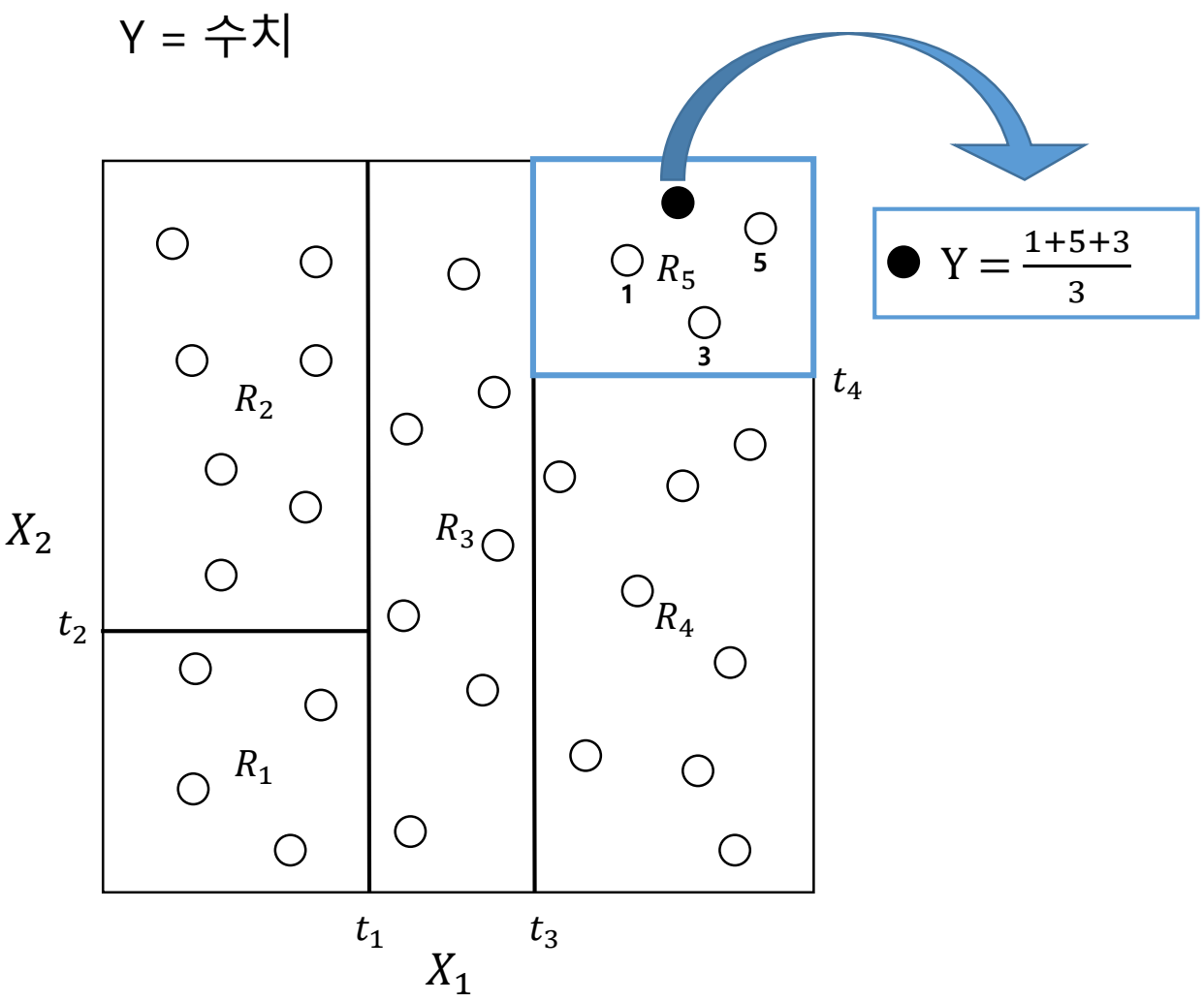
Y = 수치

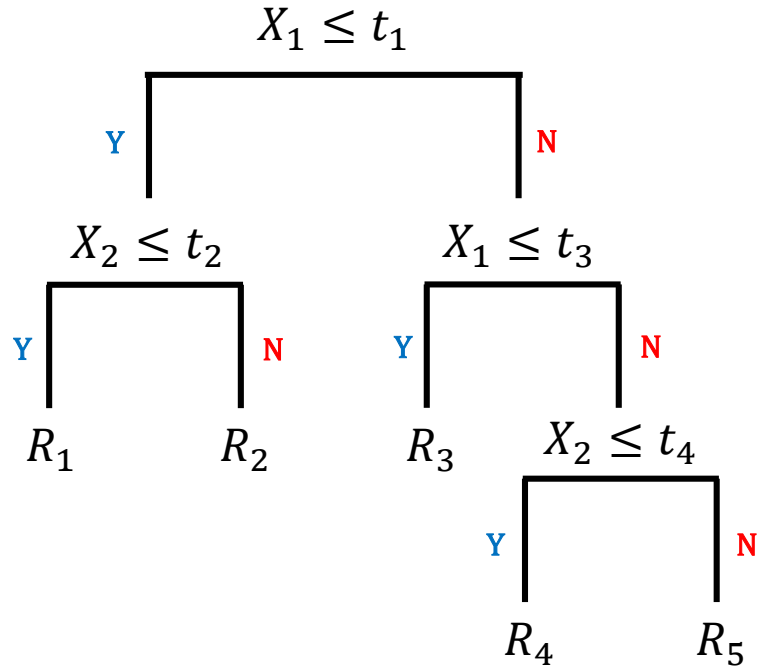


Regression Tree Model

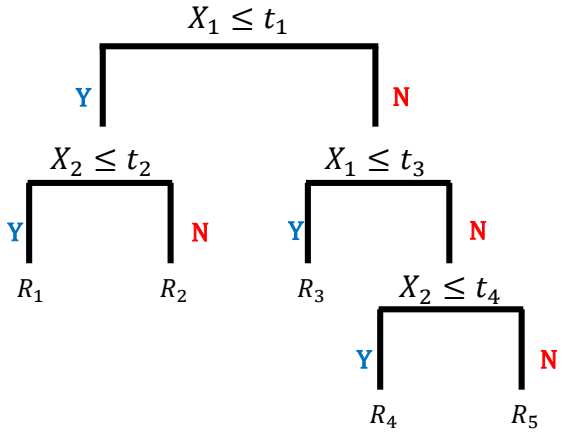


Regression Tree Model



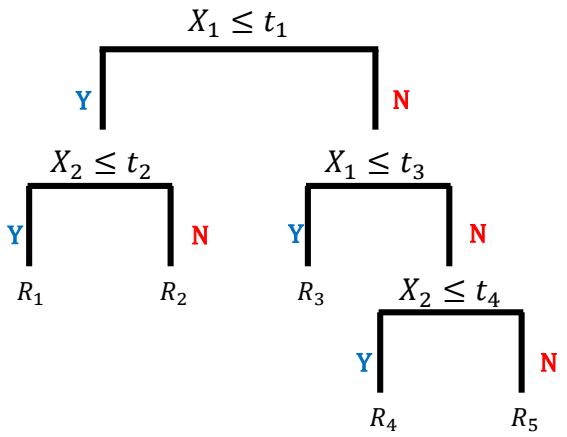


Regression Tree Model



함수로 표현?

Regression Tree Model

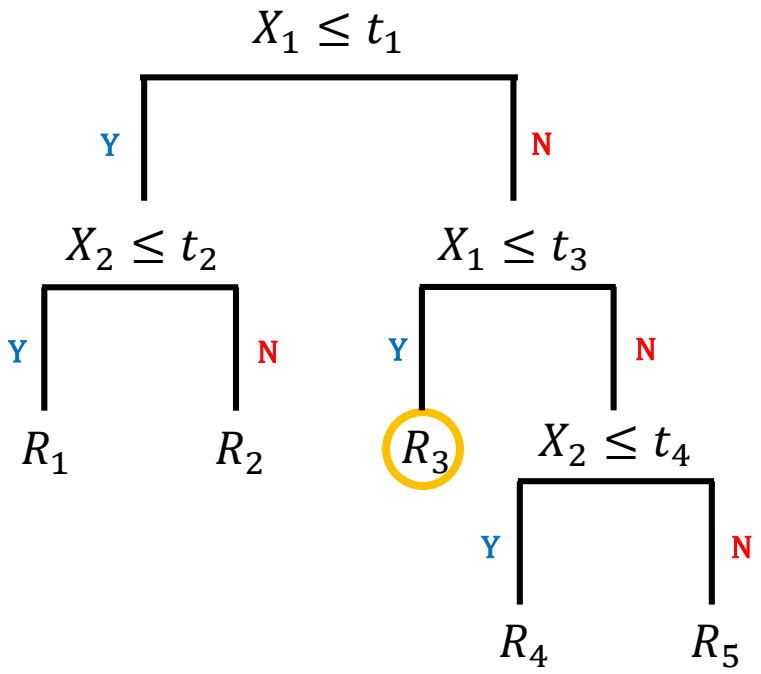


함수로 표현?

- C_m: Regression Tree Model로부터 예측한 R_m의 예측값
- I{}: Indicator function
  - { } is False : 0
  - { } is True : 1

$$\hat{f}(x) = \sum_{m=1}^{5} c_m I\{(x_1, x_2) \in R_m\}$$
$$= c_1 I\{(x_1, x_2) \in R_1\} + c_2 I\{(x_1, x_2) \in R_2\} + c_3 I\{(x_1, x_2) \in R_3\}$$
$$+ c_4 I\{(x_1, x_2) \in R_4\} + c_5 I\{(x_1, x_2) \in R_5\}$$

Regression Tree Model



$$\hat{f}(x) = \sum_{m=1}^{5} c_m I\{(x_1, x_2) \in R_m\}$$
$$= c_1 I\{(x_1, x_2) \in R_1\} + c_2 I\{(x_1, x_2) \in R_2\} + c_3 I\{(x_1, x_2) \in R_3\}$$
$$+ c_4 I\{(x_1, x_2) \in R_4\} + c_5 I\{(x_1, x_2) \in R_5\}$$

$\Rightarrow$ $R_1 \text{x } 0 + R_2 \text{x } 0 + R_3 \text{x } 1 + R_4 \text{x } 0 + R_5 \text{x } 0 = R_3$

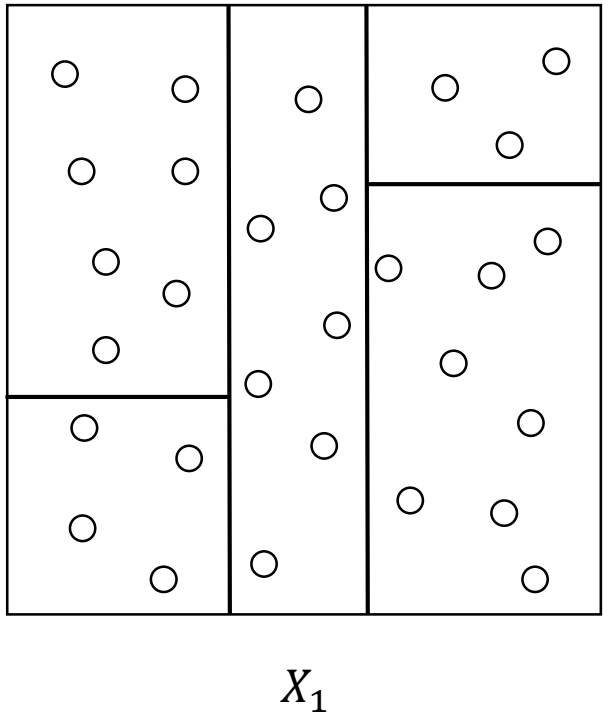
Regression Tree Model

표현 방법

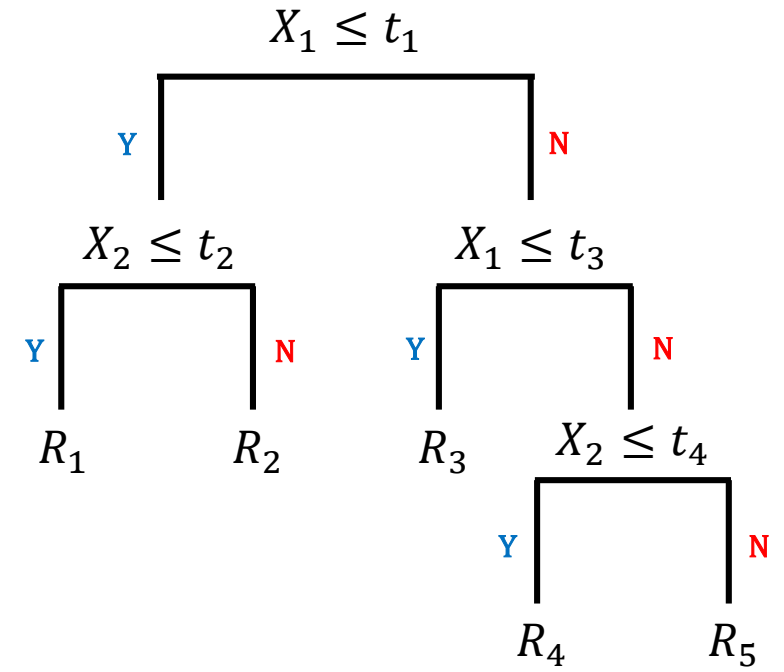
1 $$\hat{f}(x) = \sum_{m=1}^{5} c_m I\{(x_1, x_2) \in R_m\}$$

$$= c_1 I\{(x_1, x_2) \in R_1\} + c_2 I\{(x_1, x_2) \in R_2\} + c_3 I\{(x_1, x_2) \in R_3\} + c_4 I\{(x_1, x_2) \in R_4\} + c_5 I\{(x_1, x_2) \in R_5\}$$

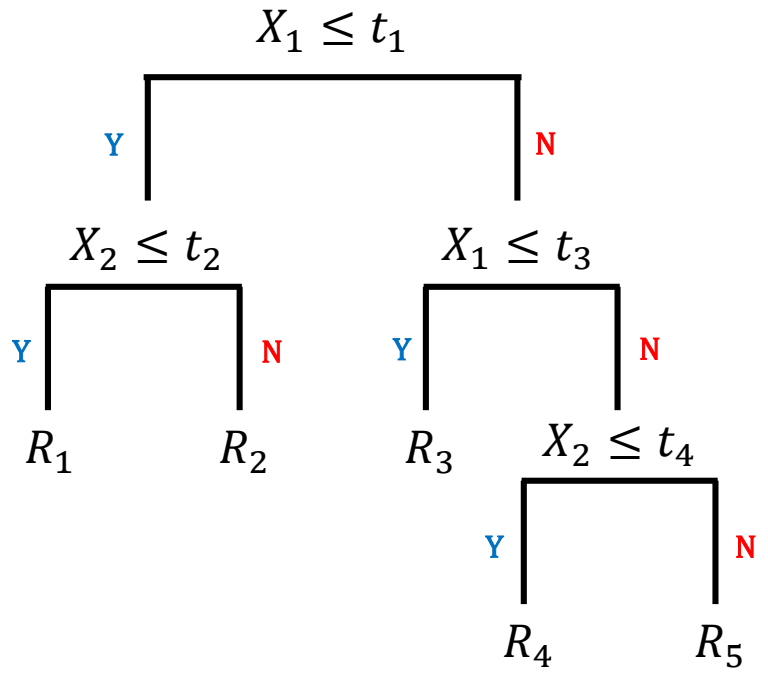
2



3



Regression Tree Model



Q. 어떻게 분할해야 잘 분할할 수 있을까?

Regression Tree Modeling Process

- Data를 M개로 분할: $R_1, R_2, \cdots, R_m$

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

- 최상의 분할은 다음 비용함수(cost function)을 최소로 할 때 얻어짐

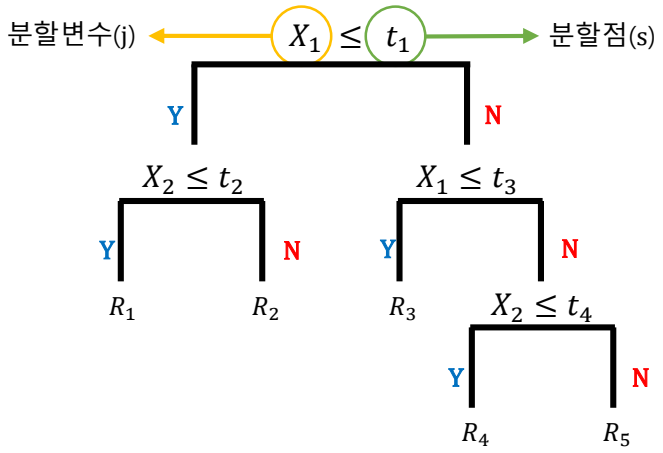
$$\min_{C_m} \sum_{i=1}^{N} (y_i - f(x_i))^2$$

$$= \min_{C_m} \sum_{i=1}^{N} (y_i - \sum_{m=1}^{M} c_m I(x \in R_m))^2$$

- 각 분할에 속해 있는 y값들의 평균으로 예측했을 때 오류가 최소 $\therefore \hat{c}_m = avg(y_i | x_i \in R_m)$

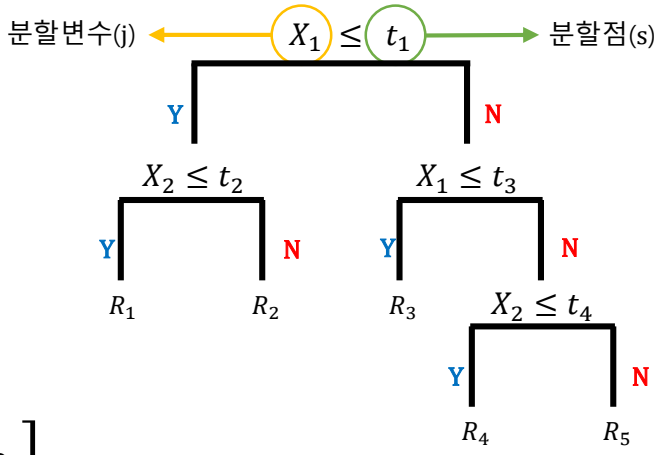
Regression Tree Modeling Process

- 분할변수(j)와 분할점(s)은 어떻게 결정할까?



Regression Tree Modeling Process

- 분할변수(j)와 분할점(s)은 어떻게 결정할까?



$$R_1(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} \leq s\}$$

$$R_2(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} > s\}$$

$$\underset{j,s}{\operatorname{argmin}} \left[\min_{\mathrm{c}_1} \sum_{x \in R_1(j,s)} (y_\mathrm{i} - \mathrm{c}_1)^2 + \min_{\mathrm{c}_2} \sum_{x \in R_2(j,s)} (y_\mathrm{i} - \mathrm{c}_2)^2 \right]$$

$$= \underset{j,s}{\operatorname{argmin}} \left[\sum_{x \in R_1(j,s)} (y_\mathrm{i} - \hat{c}_1)^2 + \sum_{x \in R_2(j,s)} (y_\mathrm{i} - \hat{c}_2)^2 \right]$$

$$\hat{c}_1 = avg(y_\mathrm{i} | x_\mathrm{i} \in R_1(\mathrm{j,s})) \text{ and } \hat{c}_2 = avg(y_\mathrm{i} | x_\mathrm{i} \in R_2(\mathrm{j,s}))$$

Regression Tree Modeling Process

- 분할변수(j)와 분할점(s)은 어떻게 결정할까?

$$R_1(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} \leq s\}$$

$$R_2(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} > s\}$$

$$\underset{j,s}{\operatorname{argmin}} \left[\min_{\mathrm{c}_1} \sum_{x \in R_1(j,s)} (y_\mathrm{i} - \mathrm{c}_1)^2 + \min_{\mathrm{c}_2} \sum_{x \in R_2(j,s)} (y_\mathrm{i} - \mathrm{c}_2)^2 \right]$$

$$= \underset{j,s}{\operatorname{argmin}} \left[\sum_{x \in R_1(j,s)} (y_\mathrm{i} - \hat{c}_1)^2 + \sum_{x \in R_2(j,s)} (y_\mathrm{i} - \hat{c}_2)^2 \right]$$

$$\hat{c}_1 = avg(y_\mathrm{i} | x_\mathrm{i} \in R_1(\mathrm{j},\mathrm{s})) \text{ and } \hat{c}_2 = avg(y_\mathrm{i} | x_\mathrm{i} \in R_2(\mathrm{j},\mathrm{s}))$$

Example

j	X_1	X_2	X_3
s	2	5	1
	3	6	7

$X_1 > 2$
$X_1 > 3$
$X_2 > 5$
$X_2 > 6$
$X_3 > 1$
$X_4 > 7$

Regression Tree Modeling Process

- 분할변수(j)와 분할점(s)은 어떻게 결정할까?

$$R_1(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} \leq s\}$$

$$R_2(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} > s\}$$

$$\underset{j,s}{\operatorname{argmin}} \left[\min_{\mathrm{c}_1} \sum_{x \in R_1(j,s)} (y_\mathrm{i} - \mathrm{c}_1)^2 + \min_{\mathrm{c}_2} \sum_{x \in R_2(j,s)} (y_\mathrm{i} - \mathrm{c}_2)^2 \right]$$

$$= \underset{j,s}{\operatorname{argmin}} \left[\sum_{x \in R_1(j,s)} (y_\mathrm{i} - \hat{c}_1)^2 + \sum_{x \in R_2(j,s)} (y_\mathrm{i} - \hat{c}_2)^2 \right]$$

$$\hat{c}_1 = avg(y_\mathrm{i} | x_\mathrm{i} \in R_1(\mathrm{j},\mathrm{s})) \text{ and } \hat{c}_2 = avg(y_\mathrm{i} | x_\mathrm{i} \in R_2(\mathrm{j},\mathrm{s}))$$

Example

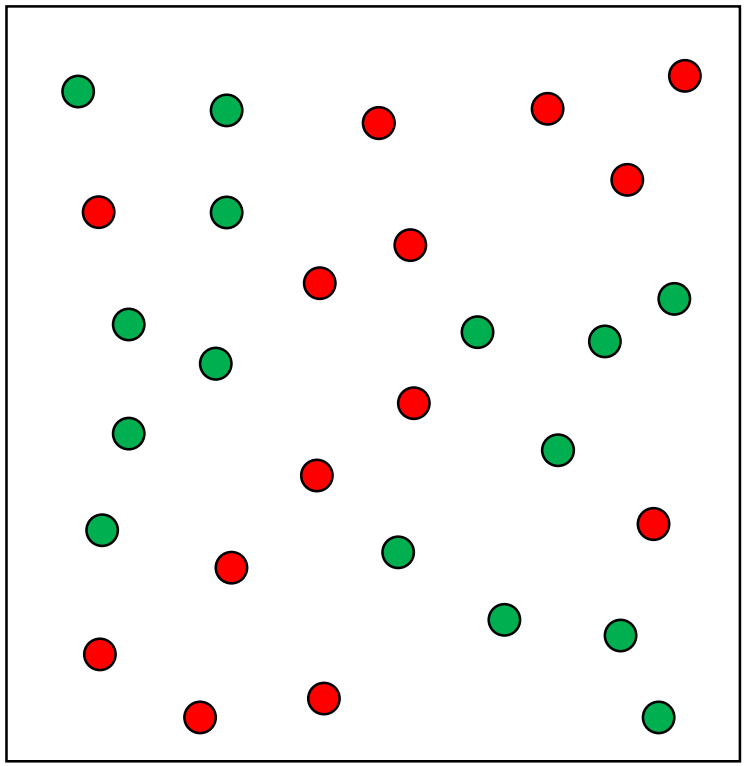
j	X_1	X_2	X_3
s	2	5	1
	3	6	7

$X_1 > 2$
$X_1 > 3$
$X_2 > 5$ → $\mathrm{j}=X_2, \mathrm{s}=5$
$X_2 > 6$
$X_3 > 1$
$X_4 > 7$

2. Decision Tree Model Ⅱ

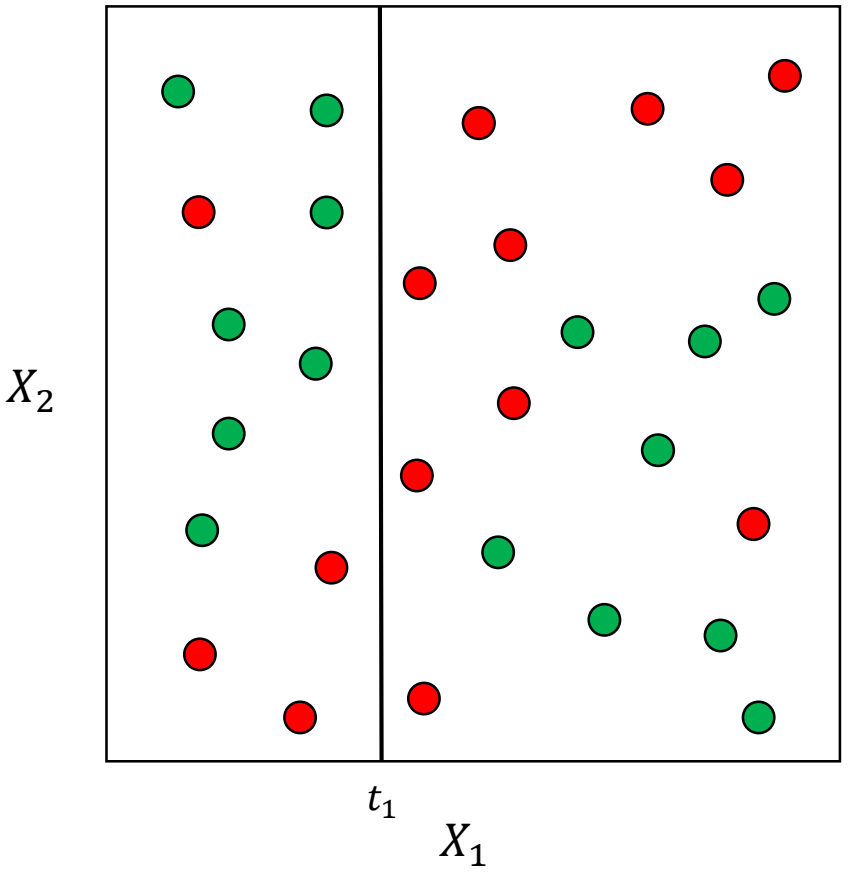
Classification Tree Model

Y = 범주



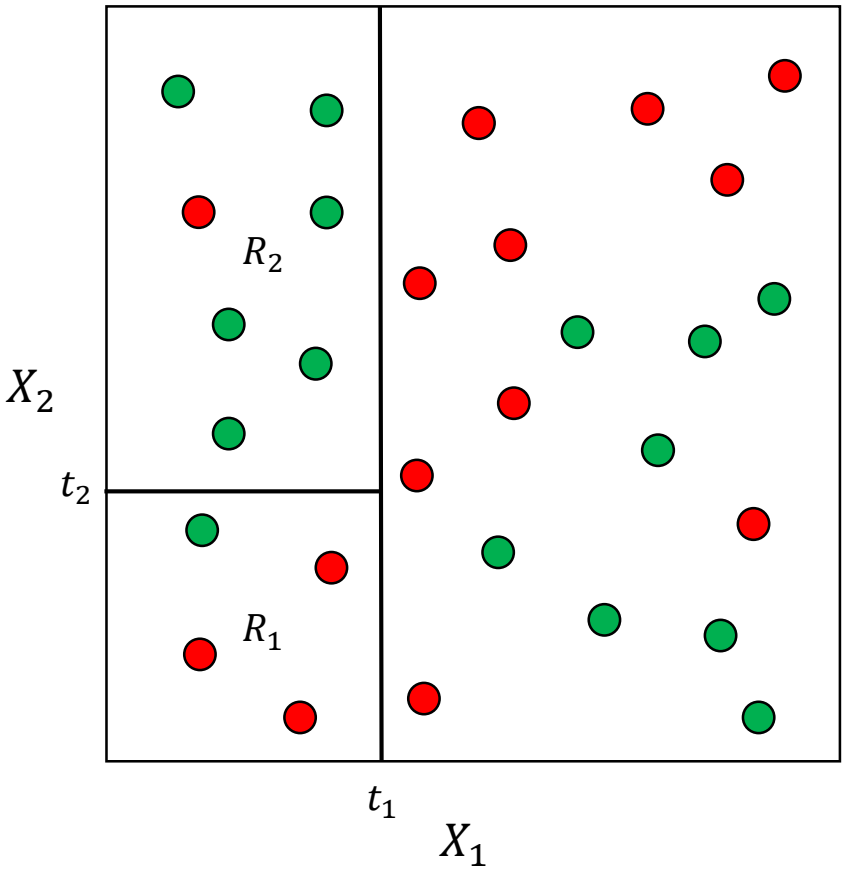
Classification Tree Model

Y = 범주



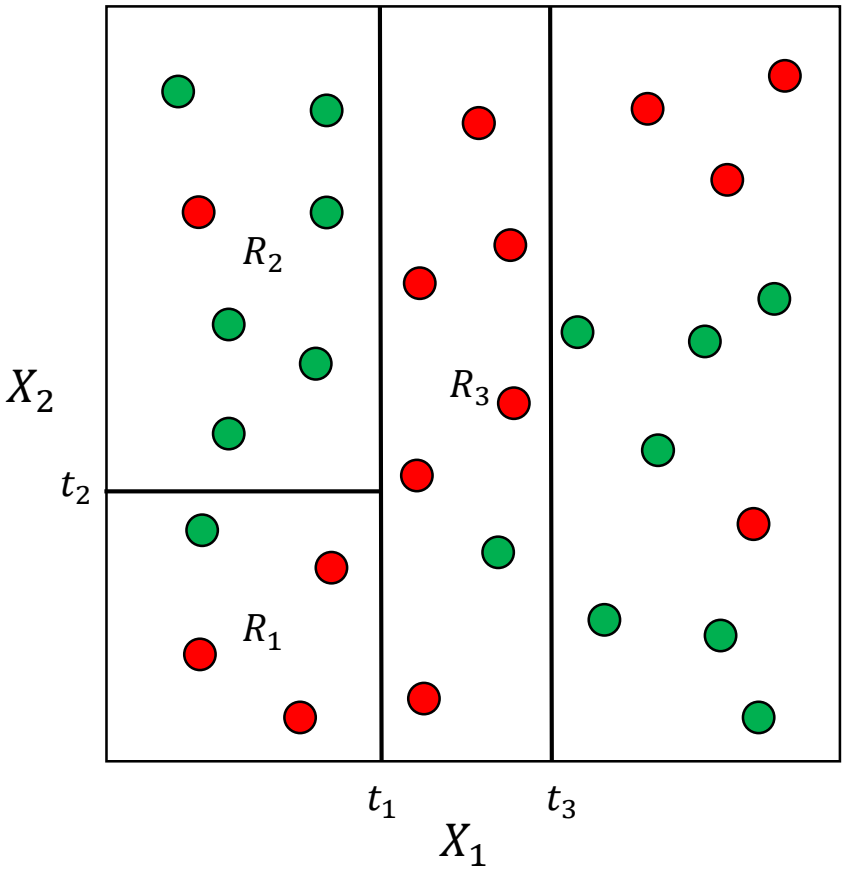
Classification Tree Model

Y = 범주



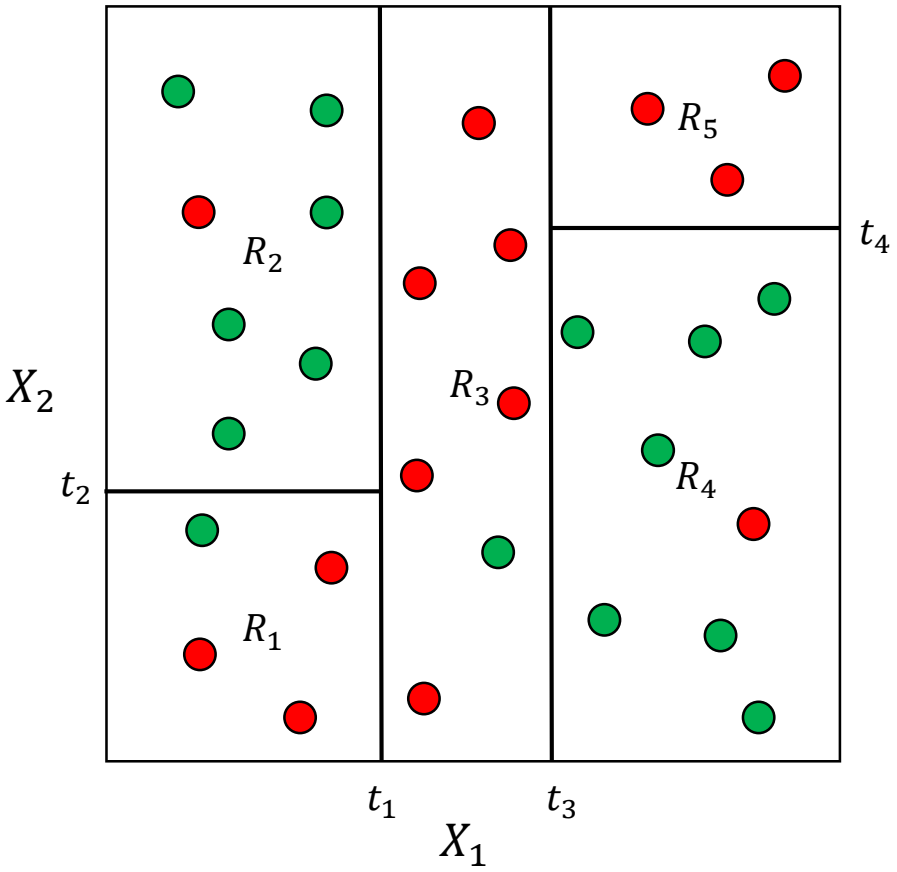
Classification Tree Model

Y = 범주



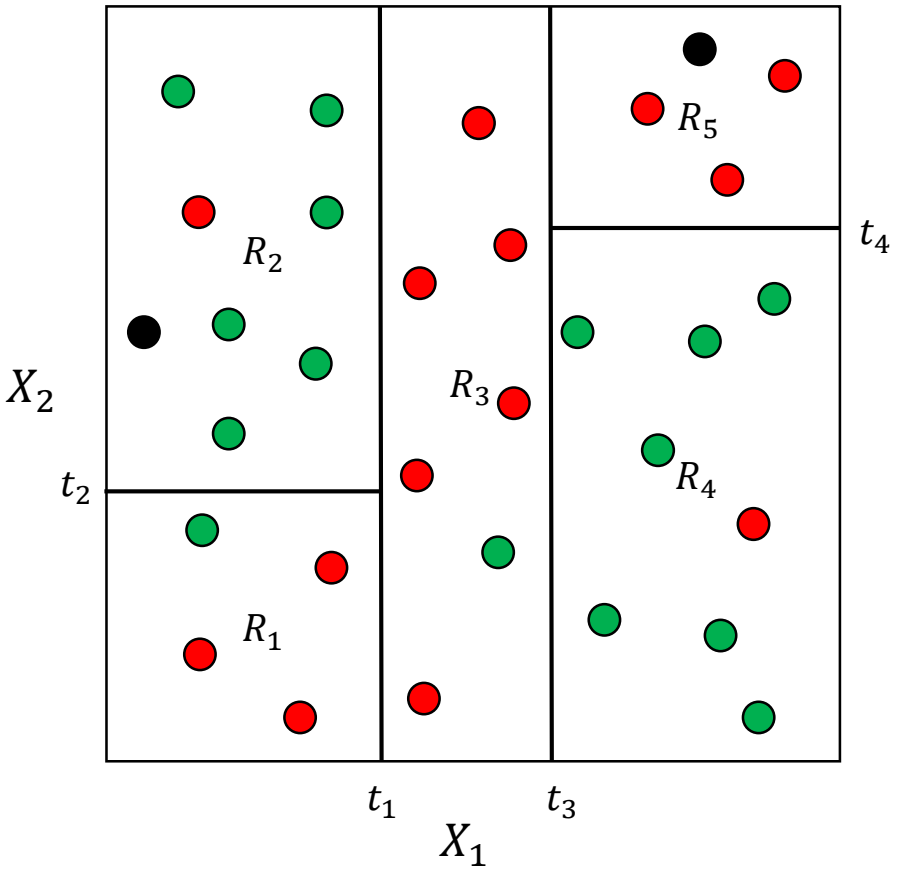
Classification Tree Model

Y = 범주



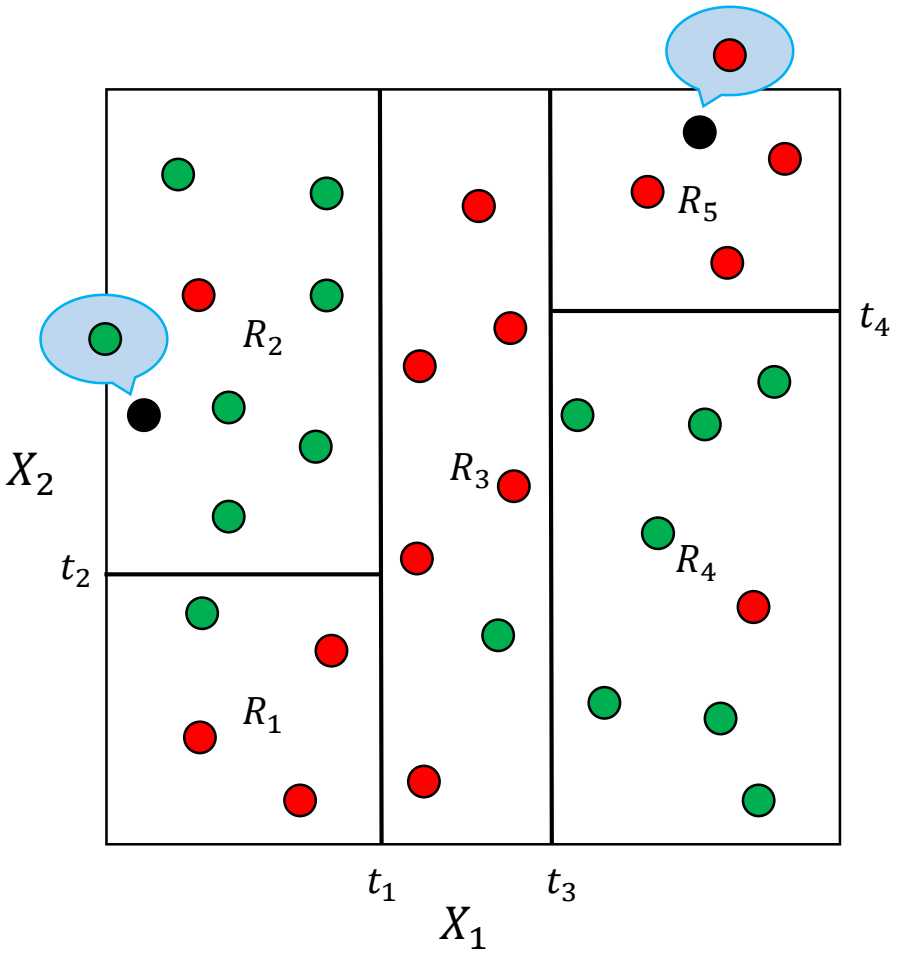
Classification Tree Model

Y = 범주



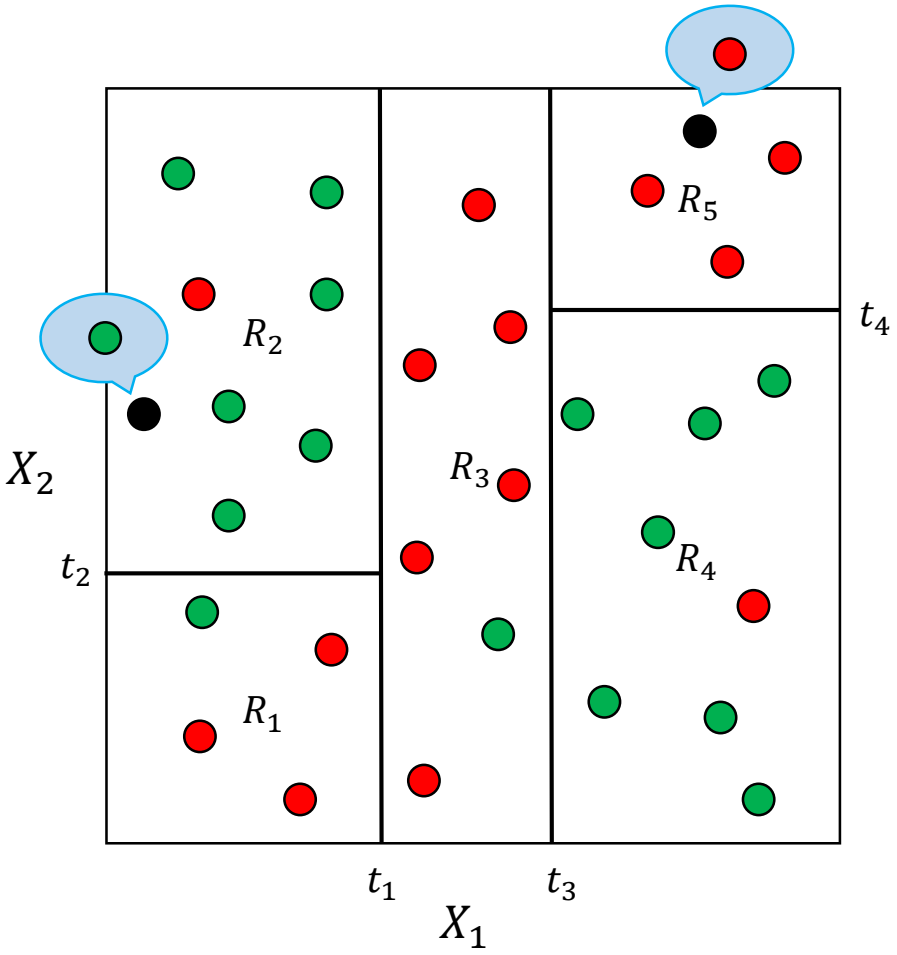
Classification Tree Model

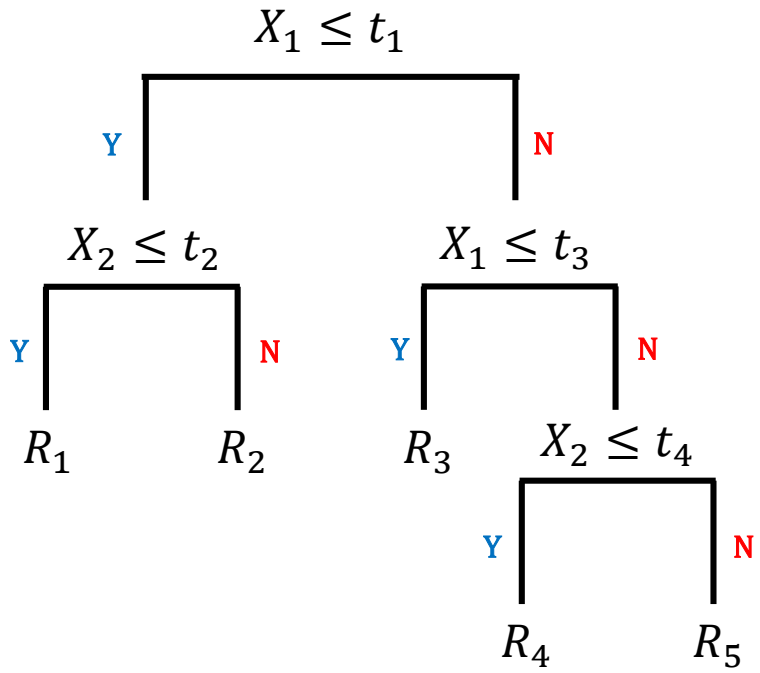
Y = 범주



Classification Tree Model

Y = 범주





Classification Tree Model

- 각 관측치마다 반응변수 값 y_i=1,2,…,K 즉, K개의 클래스 존재
- R_m: 끝 노드 m에 해당하며 관측치 개수를 가지고 있음
- $\hat{p}_{mk}$: 끝 노드 m에서 k클래스에 속해 있는 관측치의 비율

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

- 끝 노드 m으로 분류된 관측치는 k(m) 클래스로 분류

$$\mathrm{k(m)} = \underset{\mathrm{k}}{\mathrm{argmax}}\, \hat{p}_{mk}$$

Classification Tree Model

- 각 관측치마다 반응변수 값 y_i=1,2,…,K 즉, K개의 클래스 존재
- R_m: 끝 노드 m에 해당하며 관측치 개수를 가지고 있음
- $\hat{p}_{mk}$: 끝 노드 m에서 k클래스에 속해 있는 관측치의 비율

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

$$\hat{p}_{10} = \frac{2}{6},\ \hat{p}_{11} = \frac{3}{6},\ \hat{p}_{12} = \frac{1}{6}$$

- 끝 노드 m으로 분류된 관측치는 k(m) 클래스로 분류

$$k(m) = \underset{k}{\operatorname{argmax}}\, \hat{p}_{mk}$$

$$= \operatorname{argmax}(0.334, 0.5, 0.1667)$$
$$= 1$$

Classification Tree Model

- 식으로 표현

$$\hat{f}(x) = \sum_{m=1}^{5} k(m) I\{(x_1, x_2) \in R_m\}$$

$$= \mathrm{k}(1) I\{(x_1, x_2) \in R_1\} + \mathrm{k}(2) I\{(x_1, x_2) \in R_2\} + \mathrm{k}(3) I\{(x_1, x_2) \in R_3\}$$

$$+ \mathrm{k}(4) I\{(x_1, x_2) \in R_4\} + \mathrm{k}(5) I\{(x_1, x_2) \in R_5\}$$

Classification Tree Model

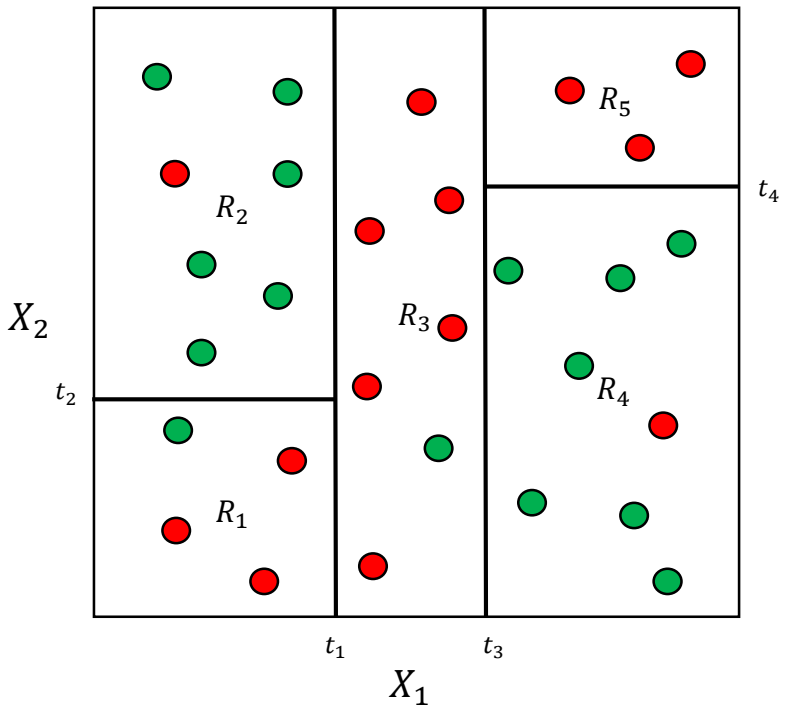
표현 방법

1 $$\hat{f}(x) = \sum_{m=1}^{5} k(m) I\{(x_1, x_2) \in R_m\}$$

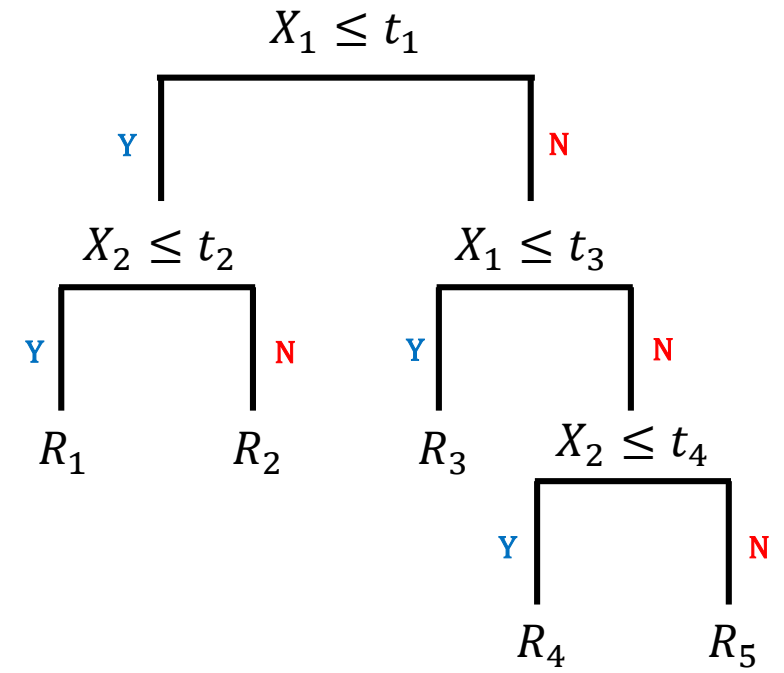
$$= \mathrm{k}(1) I\{(x_1, x_2) \in R_1\} + \mathrm{k}(2) I\{(x_1, x_2) \in R_2\} + \mathrm{k}(3) I\{(x_1, x_2) \in R_3\}$$

$$+ \mathrm{k}(4) I\{(x_1, x_2) \in R_4\} + \mathrm{k}(5) I\{(x_1, x_2) \in R_5\}$$

2



3



Classification Tree Model

- 분류 모델에서의 비용함수

일반적인 비용함수 → $\sum(y - f(x))^2$

But, 분류에서는 y값과 f(x)가 수치 값으로 나타나지 않음 !!

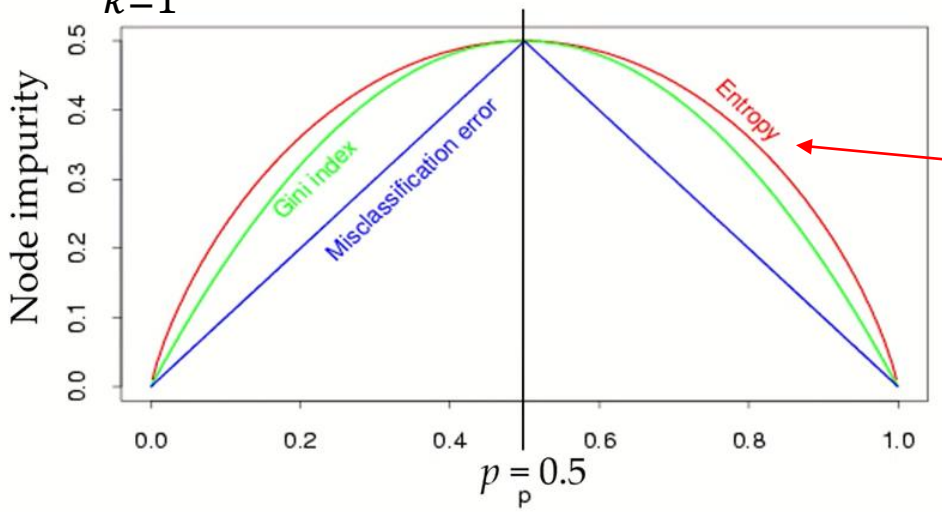
Classification Tree Model

- 분류 모델에서의 비용함수 (불순도 측정)

Misclassification rate: $\frac{1}{N_m}\sum_{x_i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{(mk)m}$

Gini Index: $\sum_{k \neq k'} \hat{p}_{mk}\,\hat{p}_{mk'} = \sum_{k=1} \hat{p}_{mk}(1 - \hat{p}_{mk})$

Cross-entropy: $-\sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk})$



P=0.5일 때 Cross-Entropy = 1
비교를 위해 Scaling 한 것

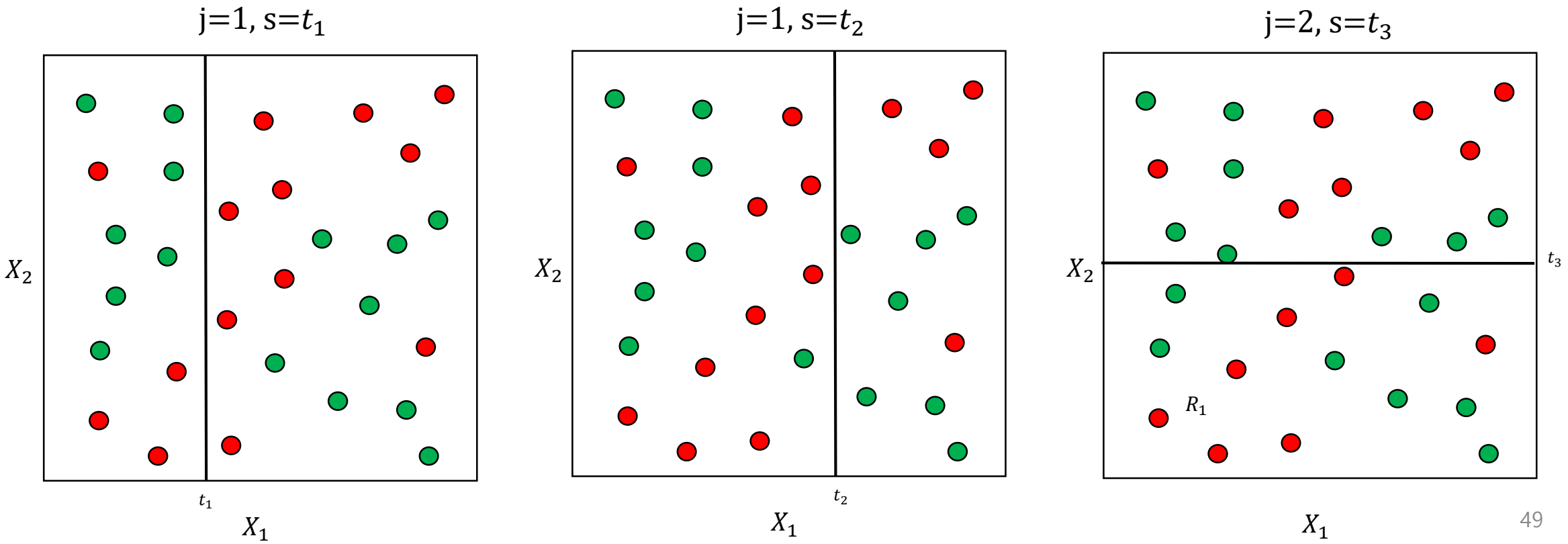
Classification Tree Modeling Process

- 분할변수(j)와 분할점(s)은 어떻게 결정할까?

$$R_1(\mathrm{j}, s) = \{\mathrm{x} \mid x_\mathrm{j} \le s\}$$

$$R_2(\mathrm{j}, s) = \{\mathrm{x} \mid x_\mathrm{j} > s\}$$

불순도 낮은 값 찾기!!



Classification Tree Modeling Process

- 분할변수(j)와 분할점(s)은 어떻게 결정할까?

$$R_1(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} \leq s\}$$

$$R_2(\mathrm{j}, \mathrm{s}) = \{\mathrm{x} \mid x_\mathrm{j} > s\}$$

모든 경우 다 수행 -> Cost Function(불순도)을 최소로 하는 j,s 구하기!!!

- Misclassification rate를 비용함수로 사용했을 때,

$$arg\min_{j,s}\left[\frac{1}{N_{R_1(j,s)}}\sum_{x_i \in R_1(j,s)} I(y_i \neq k(m)) + \frac{1}{N_{R_2(j,s)}}\sum_{x_i \in R_2(j,s)} I(y_i \neq k(m))\right]$$

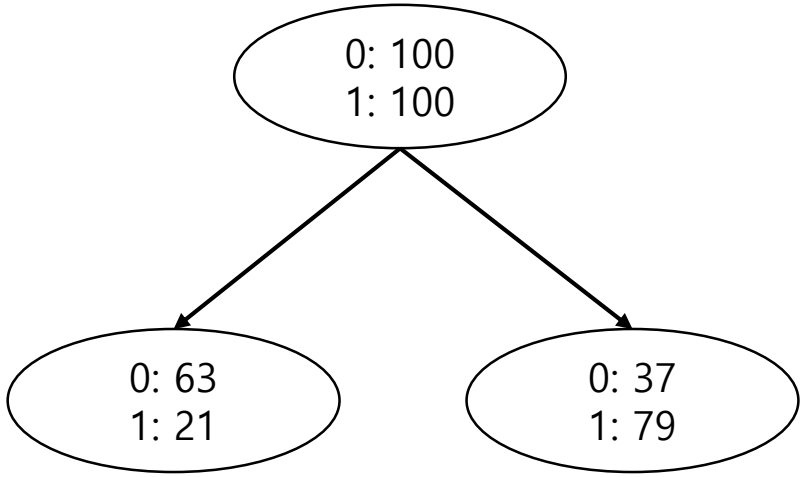
$$\mathrm{k(m)} = \underset{\mathrm{k}}{\mathrm{argmax}}\, \hat{\mathrm{p}}_{\mathrm{mk}} \qquad \hat{p}_{mk} = \frac{1}{N_m}\sum_{x_i \in R_m} I(y_i = k)$$

Classification Tree Modeling Process

- 분할법칙
  - 분할변수와 분할기준은 목표변수의 분포를 가장 잘 구별해주는 쪽으로 정함
  - 목표변수의 분포를 잘 구별해주는 측도로 순수도(purity) 또는 불순도(impurity)를 정의
  - 예를 들어 클래스 0과 클래스 1의 비율이 45%와 55%인 노드는 각 클래스의 비율이 90%와 10%인 마디에 비하여 순수도가 낮다(또는 불순도가 높다)라고 해석
  - 각 노드에서 분할변수의 분할점의 설정은 불순도의 감소가 최대가 되도록 선택

Classification Tree Modeling Process

- 오분류율 (misclassification rate)



- L 노드의 오분류율 = 21/(63+21) = 21/84 = 0.25
- R 노드의 오분류율 = 37/(37+79) = 37/116 = 0.32
- 총 오분류율 = ((21/84) x (84/200)) + ((37/116) x (116/200)) = 0.29

Example (Gini Index, Entropy)

예제: 어떤 노드의 구성이 다음과 같을 때, Gini와 entropy 지수를 계산하여라

Gini 지수 계산

$$\varphi(g) = \sum_j P_j(g)(1 - P_j(g))$$
$$= \left(\frac{6}{7} \times \frac{1}{7}\right) + \left(\frac{1}{7} \times \frac{6}{7}\right)$$
$$= 0.2449$$

Entropy 지수 계산

$$\varphi(g) = -\sum_j P_j(g) log\, P_j(g)$$
$$= -\frac{6}{7}\log\frac{6}{7} - \frac{1}{7}\log\frac{1}{7}$$
$$= 0.1781$$

Information Gain – Cross Entropy

- $Entropy(S) = -\sum_{i=1} p_i \log_2 p_i$

  c is the number of class, p_i is the proportion of S belong to class i

  - *For example, suppose S is a collection of 14 examples [9+, 5-]*

    $$Entropy[9+, 5-] = -\sum_{i=1}^{2} p_i \log_2 p_i = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.94$$

- ***Information gain(IG)****: the expected reduction in entropy caused by partitioning the data according to this variable* (특정 변수를 사용했을 때 *entropy* 감소량)
  - *IG(S, A): Information gain of a variable A.* (변수 *A*를 사용했을 때 *entropy* 감소량)

    $$IG(S, A) = Entropy(S) - \sum_{v \in value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

  - *value(A): the set of all possible values for a variable A*
  - S_v*: the subset of S for which variable A has value v*

Example (Information Gain)

Outlook	Temperature	Humidity	Wind	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

$$Values(wind) = False, True$$

$$S = [9+, 5-]$$ ← +: Play(Yes) / -:Play(No)

Wind(False)
+: Play(Yes) / -:Play(No) → $S_{False} \leftarrow [6+, 2-]$

$S_{True} \leftarrow [3+, 3-]$ ← Wind(True)
+: Play(Yes) / -:Play(No)

$$IG(S, Wind) = Entropy(S) - \sum_{v \in value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= Entropy(s) - \frac{8}{14} Entropy(S_{False}) - \frac{6}{14} Entropy(S_{True})$$

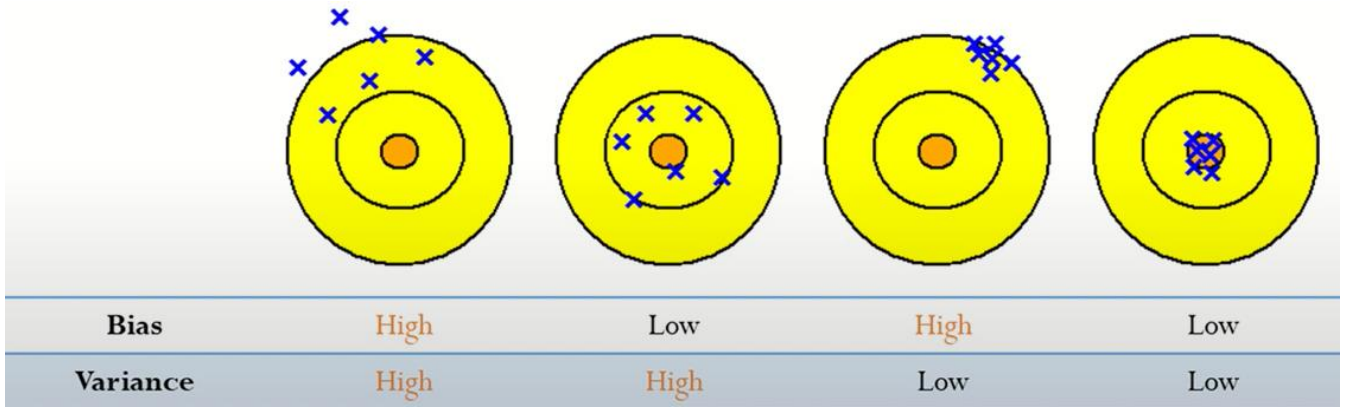
$$= 0.048$$

Similarly, IG(S, Humidity) = 0.151

Humidity provides greater information gain than Wind

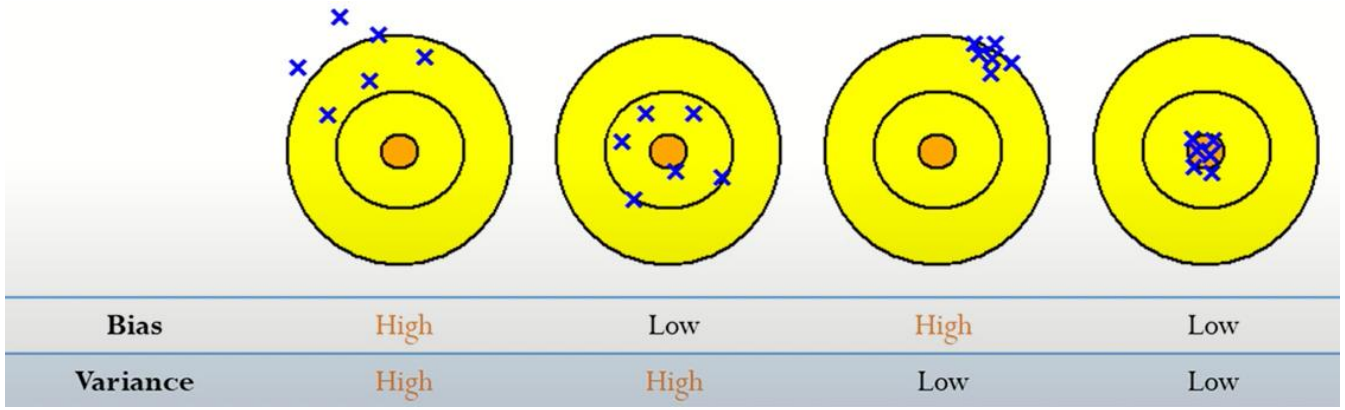
Disadvantages of Individual Tree Model

- 계층적 구조로 인해 중간에 에러가 발생하면 다음 단계로 에러가 계속 전파
- 학습 데이터의 미세한 변동에도 최종 결과 크게 영향
- 적은 개수의 노이즈에도 크게 영향
- 나무의 최종노드 개수를 늘리면 과적합 위험 (Low Bias, Large Variance)



Disadvantages of Individual Tree Model

- 계층적 구조로 인해 중간에 에러가 발생하면 다음 단계로 에러가 계속 전파
- 학습 데이터의 미세한 변동에도 최종 결과 크게 영향
- 적은 개수의 노이즈에도 크게 영향
- 나무의 최종노드 개수를 늘리면 과적합 위험 (Low Bias, Large Variance)

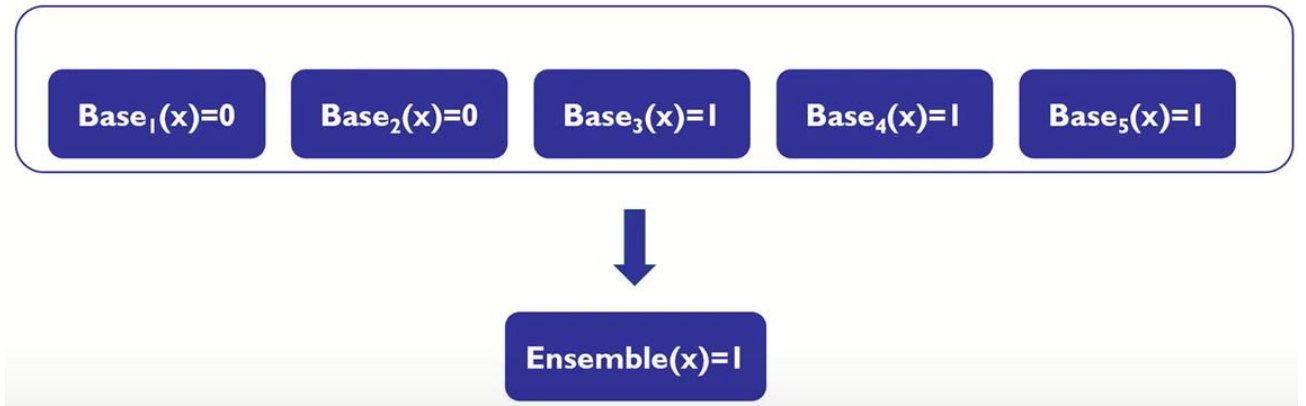


- **해결방안 → 랜덤 포레스트 (Random Forest)**

3. Random Forest

Background of Random Forest - Ensemble

- 여러 Base 모델들의 예측을 다수결 법칙 or 평균을 이용해 통합 -> 예측 정확성 향상시키는 방법



- 조건 (Ensemble Model 성능 > Base 모델 성능)
  - Base 모델들이 서로 독립적
  - Base 모델들이 무작위 예측을 수행하는 모델보다 성능이 좋은 경우

-> Base 모델의 오류율이 0.5보다 작음

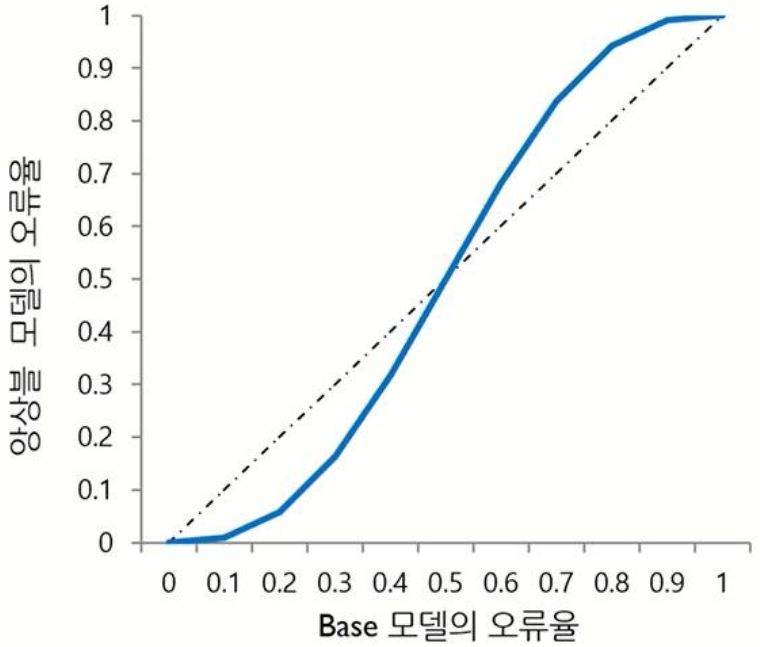
Background of Random Forest - Ensemble

- 앙상블 모델의 오류율

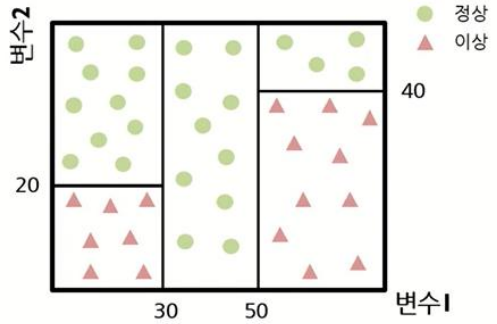
$$e_{ensemble} = \sum_{i=[N/2]}^{N} \binom{N}{i} e^i (1-e)^{N-i}$$

e: error rate of Base Model
N: number of Base Model

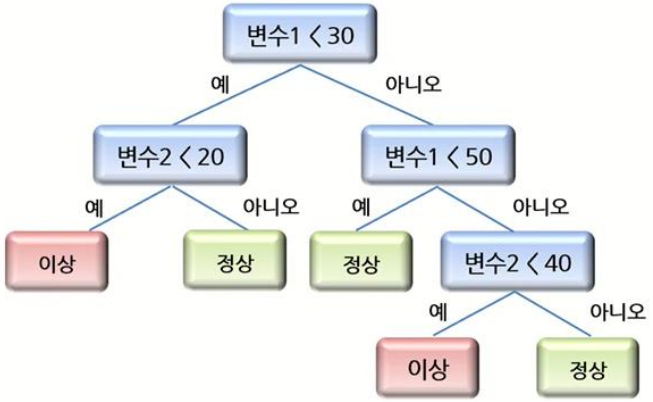
  - 예를 들어, 5개의 binary classifier를 base model로 가지는 앙상블 모델의 오류율은 아래 그림과 같이 나타남. Base Model의 성능이 무작위 모델보다는 좋아야 함.



Background of Random Forest - Ensemble



〈데이터 분포〉

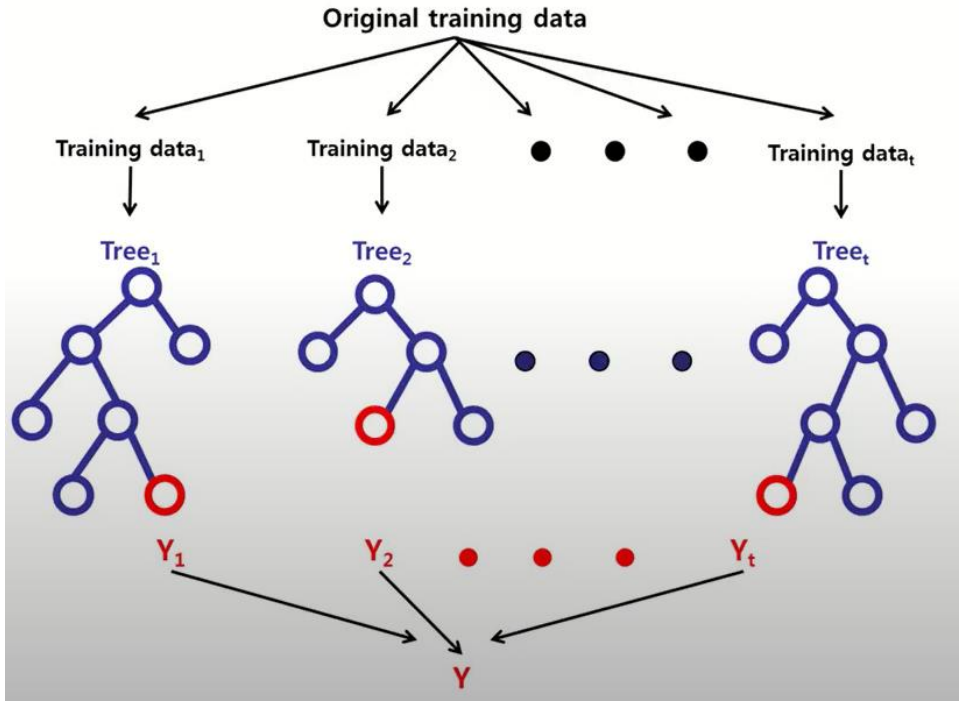


〈Decision tree 구성〉

- 의사결정나무모델은 앙상블 모델의 base 모델로써 활용도가 높음 (Maclin et al., 1999)
- Low computational complexity: 데이터의 크기가 방대한 경우에도 모델을 빨리 구축할 수 있음
- Nonparametric: 데이터 분포에 대한 전제가 필요하지 않음

Overview of the Random Forest

- 다수의 의사결정나무 모델에 의한 예측을 종합하는 앙상블 방법
- 일반적으로 하나의 의사결정나무 모델보다 높은 예측 정확성을 보여줌
- 관측치 수에 비해 변수의 수가 많은 고차원 데이터에서 중요 변수 선택 기법으로 널리 활용됨



1. Bootstrap 기법을 이용하여 다수의 training data 생성
2. 생성된 training data로 decision tree 모델 구축 (무작위 변수를 사용하여)
3. 예측 종합

Overview of the Random Forest

- **핵심 아이디어: Diversity, Random 확보**

1. 여러 개의 Training data를 생성하여 각 데이터마다 개별 의사결정나무 모델을 구축
   → Bagging

2. 의사결정나무 모델을 구축 시 변수를 무작위로 선택
   → Random subspace

Overview of the Random Forest

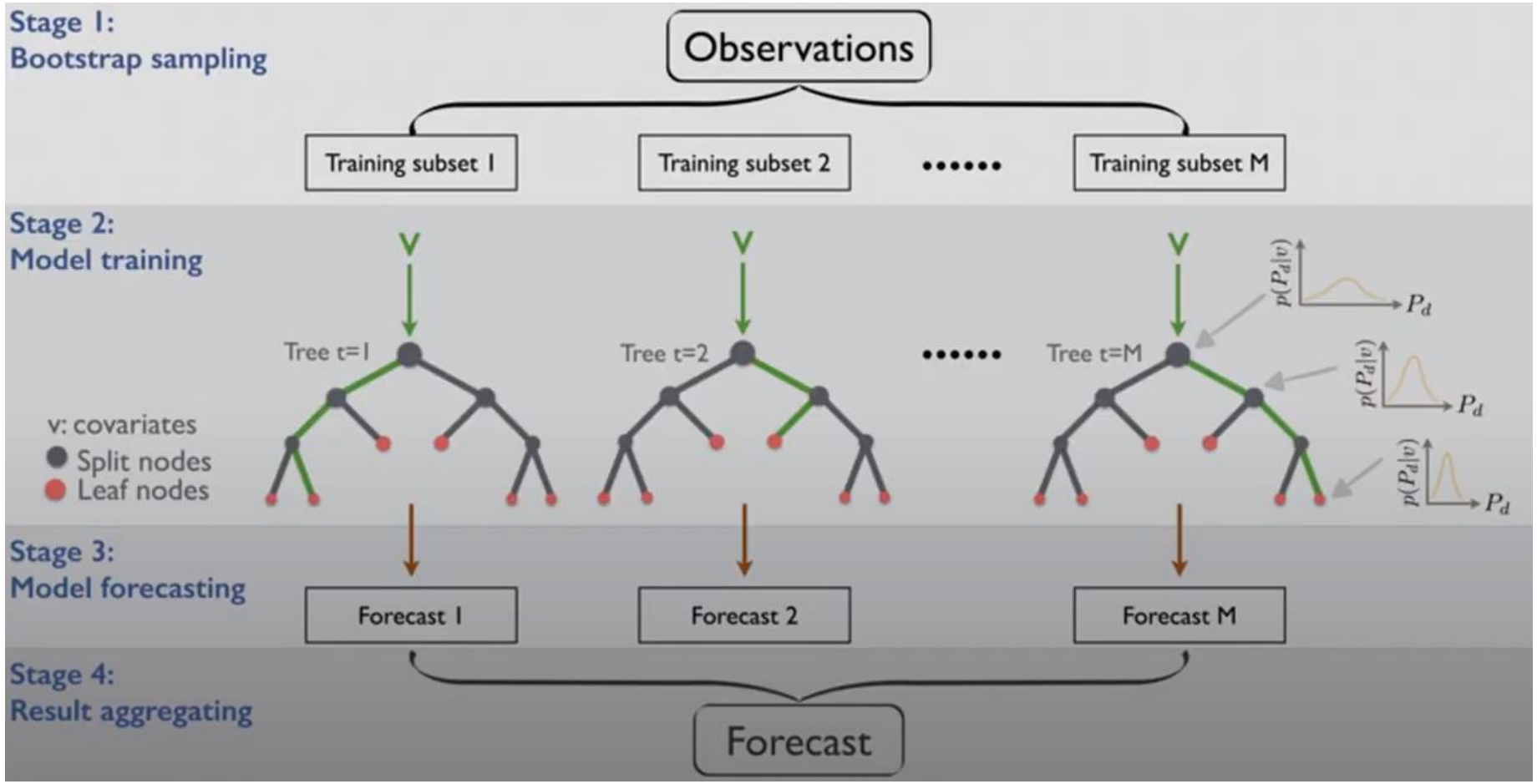
- **핵심 아이디어: Diversity, Random 확보**

1. 여러 개의 Training data를 생성하여 각 데이터마다 개별 의사결정나무 모델을 구축
   → Bagging

2. 의사결정나무 모델을 구축 시 변수를 무작위로 선택
   → Random subspace

Bagging (Bootstrap Aggregating)

Bagging (Bootstrap Aggregating): 각각의 bootstrap 샘플로부터 생성된 모델을 합침



Bagging (Bootstrap Aggregating)

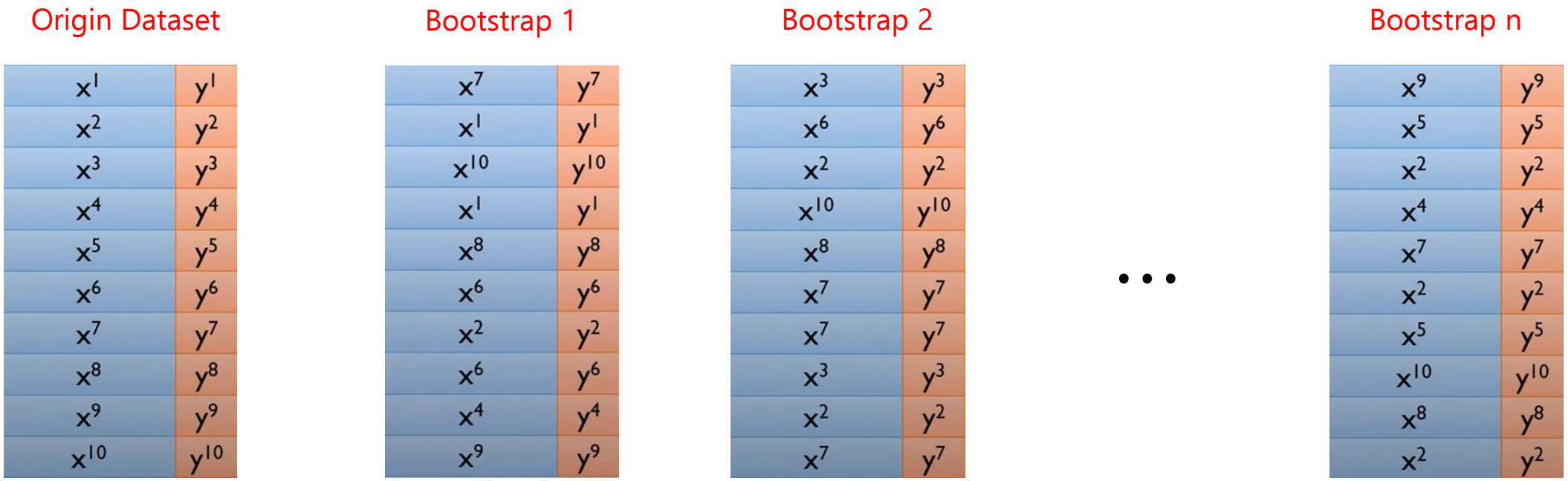
- Bootstrapping
  - 각 모델은 서로 다른 학습 Dataset을 이용
  - 각 Dataset은 복원 추출(sampling with replacement)을 통해 원래 데이터의 수만큼의 갖도록 샘플링
  - 개별 Dataset을 Bootstrap set이라고 부름

Origin Dataset | Bootstrap 1 | Bootstrap 2 | Bootstrap n

x^1	y^1
x^2	y^2
x^3	y^3
x^4	y^4
x^5	y^5
x^6	y^6
x^7	y^7
x^8	y^8
x^9	y^9
x^{10}	y^{10}

Bagging (Bootstrap Aggregating)

- Bootstrapping
  - 각 모델은 서로 다른 학습 Dataset을 이용
  - 각 Dataset은 복원 추출(sampling with replacement)을 통해 원래 데이터의 수만큼의 갖도록 샘플링
  - 개별 Dataset을 Bootstrap set이라고 부름



Bagging (Bootstrap Aggregating)

- Bootstrapping
  - 이론적으로 한 개체가 하나의 Bootstrap에 한번도 선택되지 않을 확률

$$p = \left(1 - \frac{1}{N}\right)^N \rightarrow \lim_{N \to \infty} \left(1 - \frac{1}{N}\right)^N = e^{-1} = 0.368$$

Bagging (Bootstrap Aggregating)

- Result Aggregating
  - For classification problem
    - Majority voting

$$Ensemble(\hat{y}) = \underset{i}{arg\max}\left(\sum_{j=1}^{n} I(\hat{y}_j = i), i \in \{0,1\}\right)$$

Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

Bagging (Bootstrap Aggregating)

- Result Aggregating
  - For classification problem
    - Majority voting

$$Ensemble(\hat{y}) = \underset{i}{arg\max}\left(\sum_{j=1}^{n} I(\hat{y}_j = i), i \in \{0,1\}\right)$$

Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

$$\sum_{j=1}^{n} I(\hat{y}_j = 0) = 4$$

$$\sum_{j=1}^{n} I(\hat{y}_j = 1) = 6$$

$$Ensemble(\hat{y}) = 1$$

Bagging (Bootstrap Aggregating)

- Result Aggregating
  - For classification problem
    - Weighted voting (weight = training accuracy of individual models)

$$Ensemble(\hat{y}) = \underset{i}{arg\max}\left(\frac{\sum_{j=1}^{n}(TrainAcc_j)\cdot I(\hat{y}_j = i)}{\sum_{j=1}^{n}(TrainAcc_j)}\right), i \in \{0,1\}$$

Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

$$\frac{\sum_{j=1}^{n}(TrainAcc_j)\cdot I(\hat{y}_j = 0)}{\sum_{j=1}^{n}(TrainAcc_j)} = 0.424$$

$$\frac{\sum_{j=1}^{n}(TrainAcc_j)\cdot I(\hat{y}_j = 1)}{\sum_{j=1}^{n}(TrainAcc_j)} = 0.576$$

$$Ensemble(\hat{y}) = 1$$

Bagging (Bootstrap Aggregating)

- Result Aggregating
  - For classification problem
    - Weighted voting (weight = predicted probability for each class)

$$Ensemble(\hat{y}) = \underset{i}{argmax}\left(\frac{1}{n}\sum_{j=1}^{n} P(y=i), i \in \{0,1\}\right)$$

Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

1 - P(y=1)

$$\frac{1}{10}\sum_{j=1}^{10} P(y=0) = 0.371$$

$$\frac{1}{10}\sum_{j=1}^{10} P(y=1) = 0.629$$

$$Ensemble(\hat{y}) = 1$$

Overview of the Random Forest

- **핵심 아이디어: Diversity, Random 확보**

1. 여러 개의 Training data를 생성하여 각 데이터마다 개별 의사결정나무 모델을 구축
   → Bagging

2. 의사결정나무 모델을 구축 시 변수를 무작위로 선택
   → Random subspace

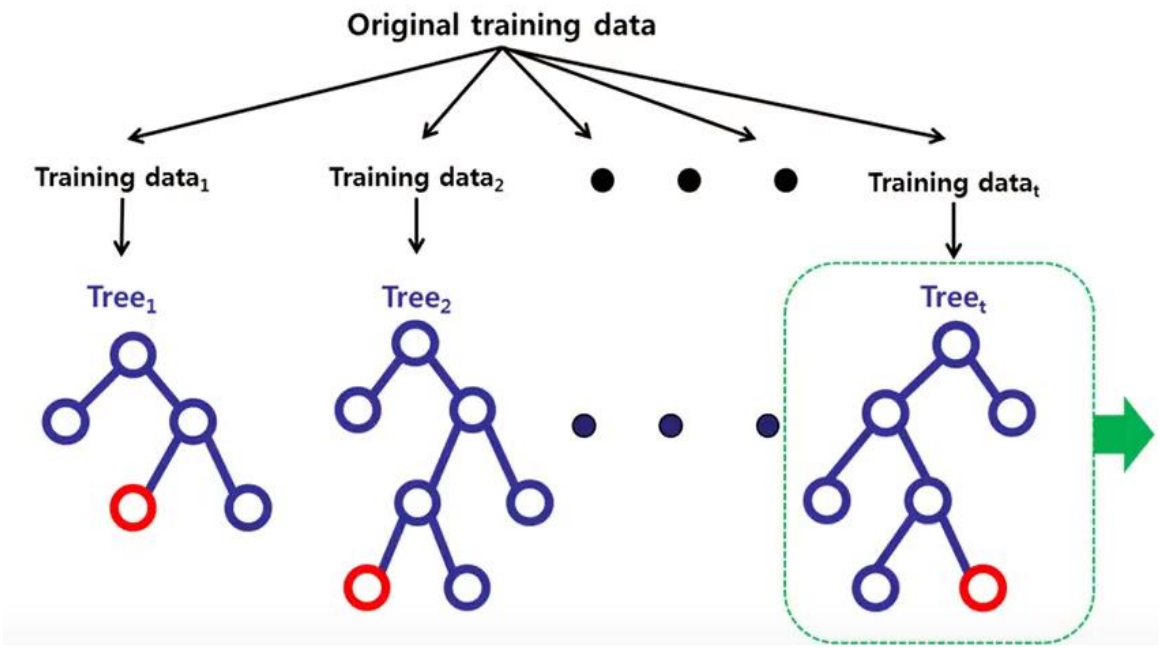
Overview of the Random Forest

- **핵심 아이디어: Diversity, Random 확보**

1. 여러 개의 Training data를 생성하여 각 데이터마다 개별 의사결정나무 모델을 구축
   → Bagging

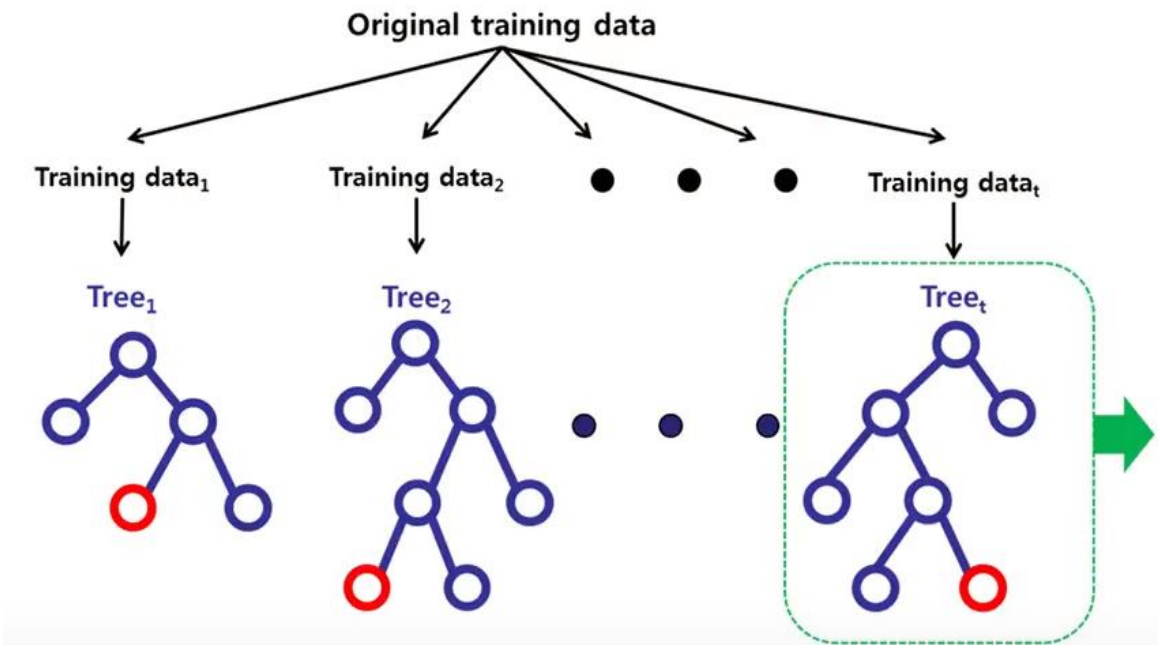
2. 의사결정나무 모델을 구축 시 변수를 무작위로 선택
   → Random subspace

Random subspace

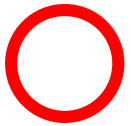


원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수																

Random subspace



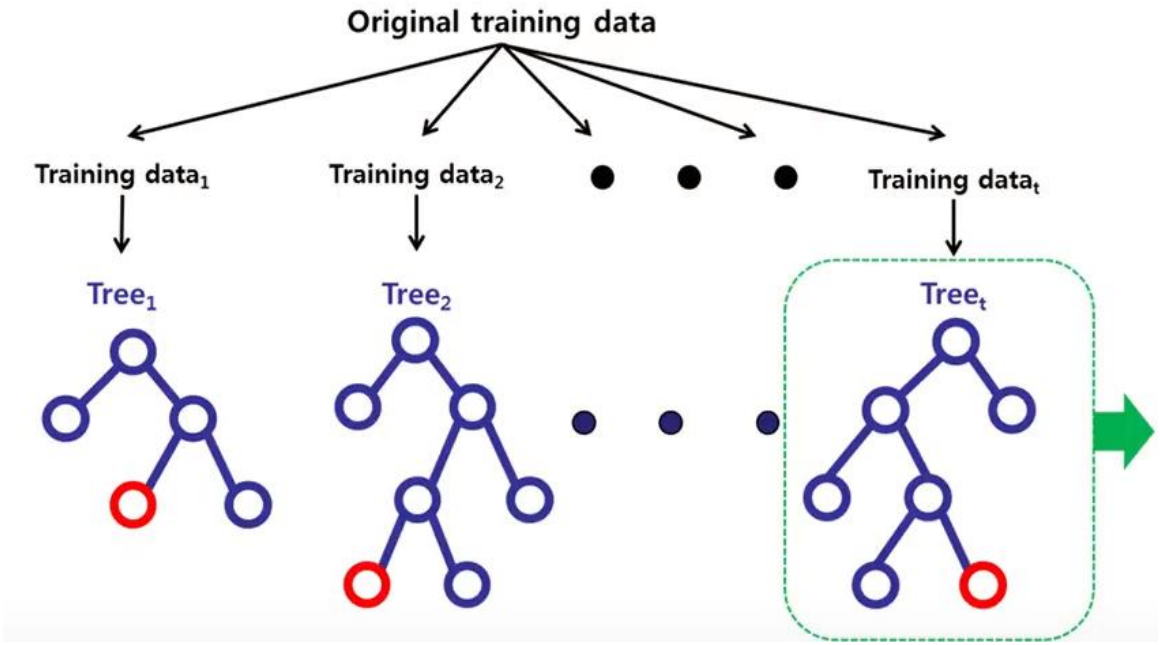
Tree

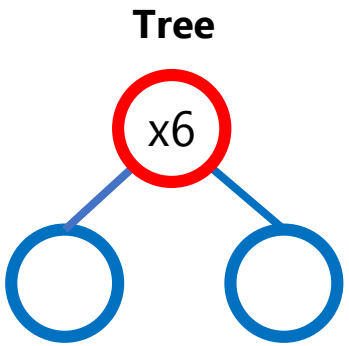


원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수			x3		x5	x6				x10						

1. 원래 변수들 중 모델 구축에 쓰일 입력 변수를 무작위로 선택

Random subspace

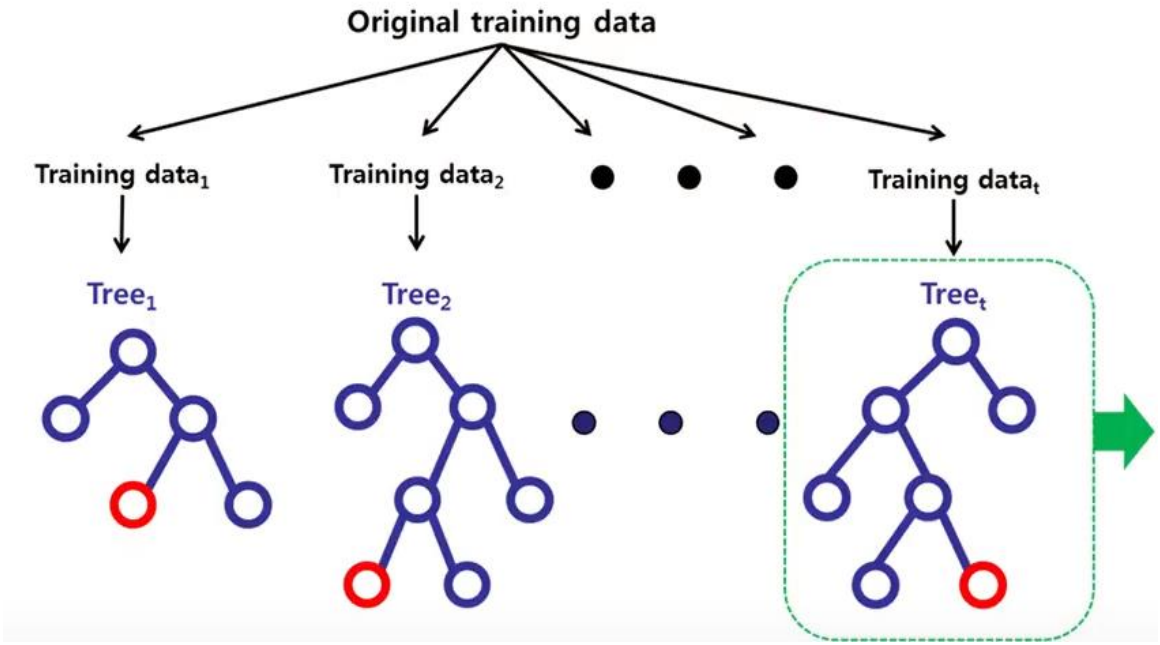


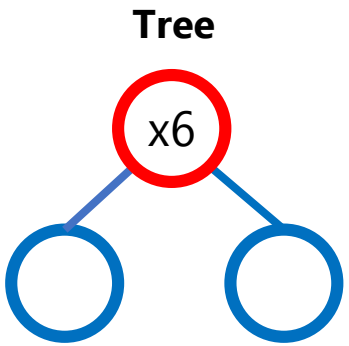


원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수			x3		x5	x6				x10						

2. 선택된 입력 변수 중 분할될 변수를 선택

Random subspace

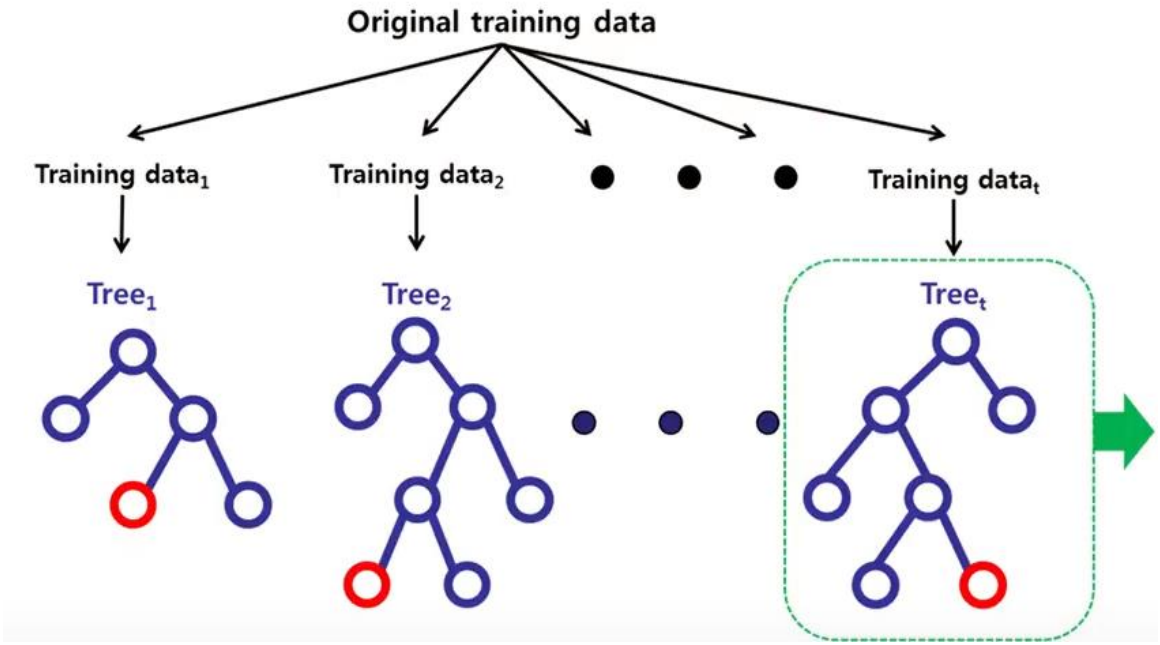


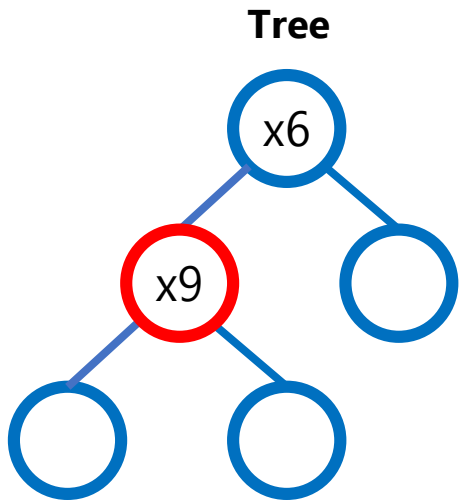


원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수			x3		x5	x6				x10						

2. 선택된 입력 변수 중 분할될 변수를 선택 -> **일반 tree의 분할점과 분할 변수를 찾는 과정과 같음**

Random subspace



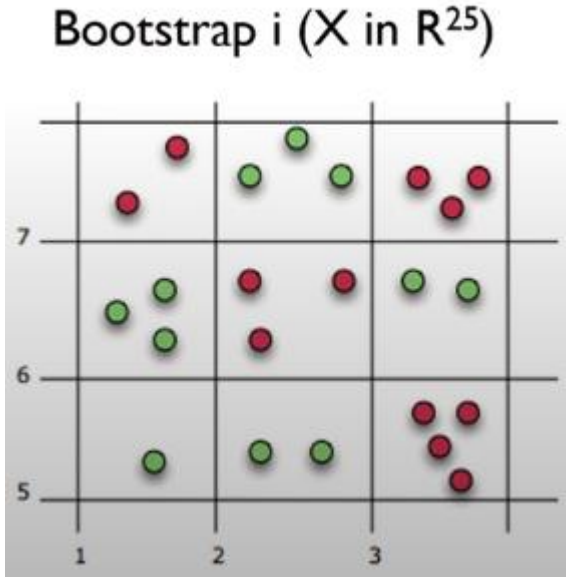


원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수	x1				x5				x9	x10						

3. 이러한 과정을 full-grown tree가 될 때까지 반복

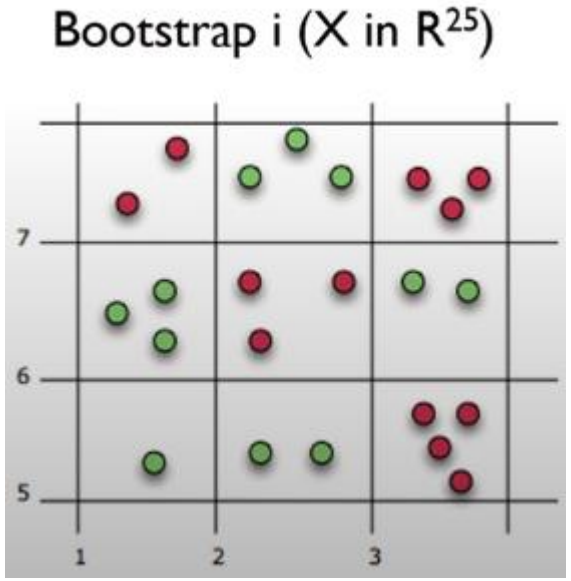
Random subspace

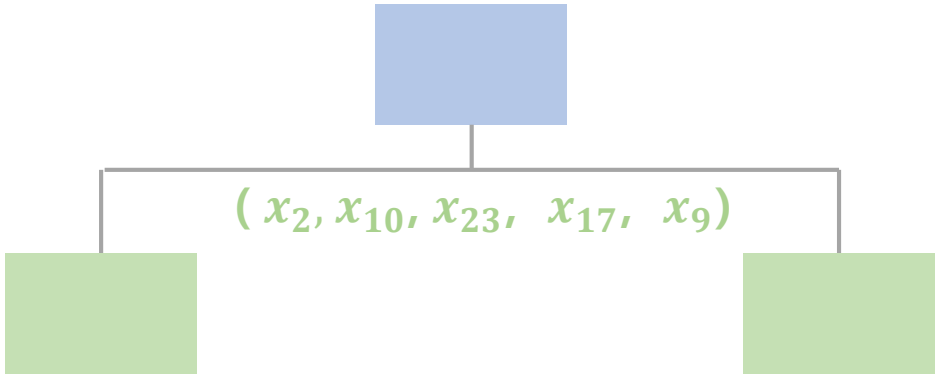
- **Random subspace**
  - 의사결정 나무의 분기점을 탐색할 때, 원래 변수의 수보다 적은 수의 변수를 임의로 선택하여 해당 변수들만을 고려 대상으로 함
  - 예시



Random subspace

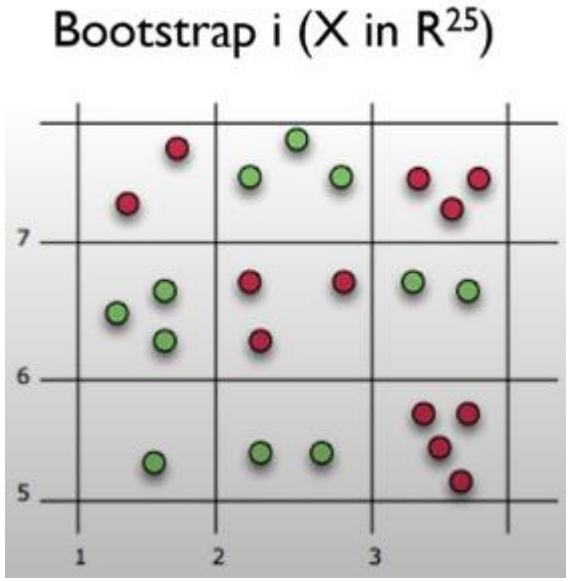
- **Random subspace**
  - 의사결정 나무의 분기점을 탐색할 때, 원래 변수의 수보다 적은 수의 변수를 임의로 선택하여 해당 변수들만을 고려 대상으로 함
  - 예시

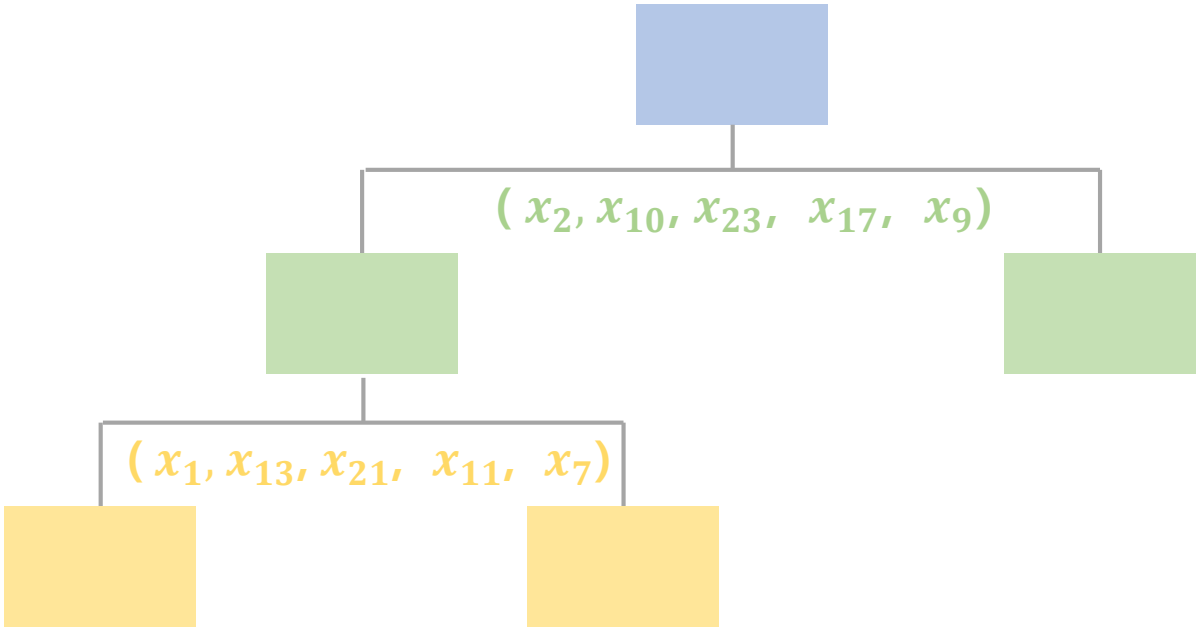




Random subspace

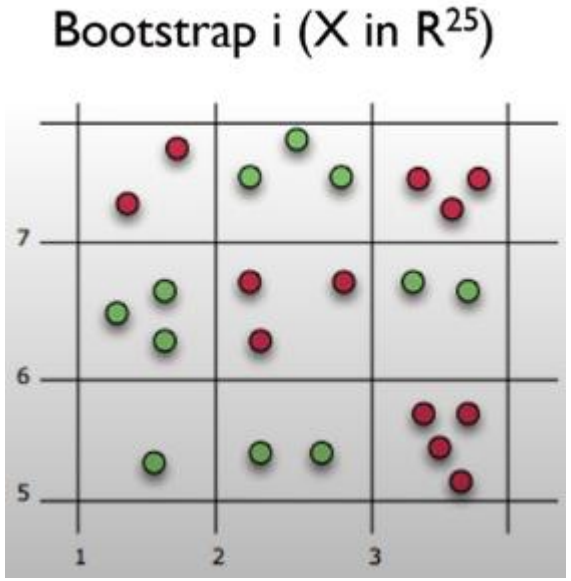
- **Random subspace**
  - 의사결정 나무의 분기점을 탐색할 때, 원래 변수의 수보다 적은 수의 변수를 임의로 선택하여 해당 변수들만을 고려 대상으로 함
  - 예시

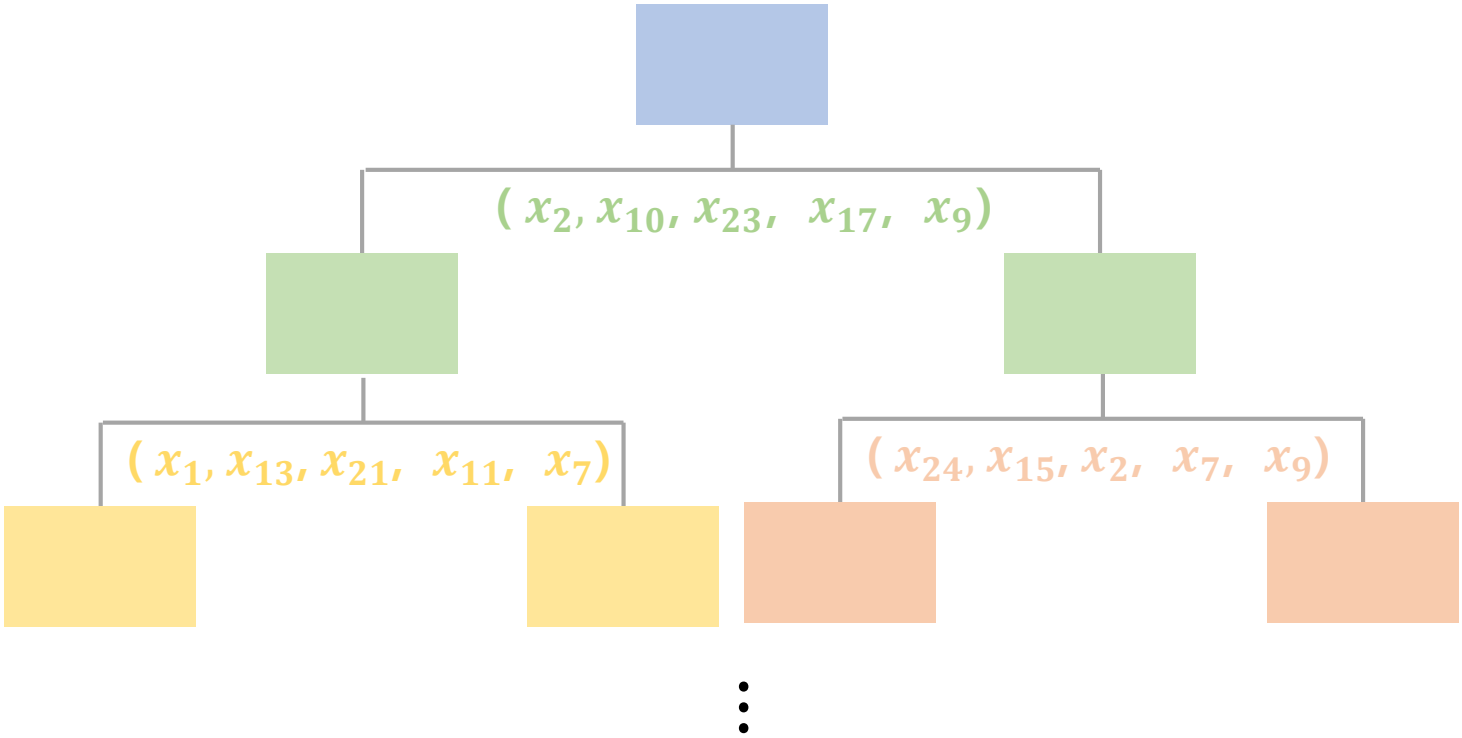




Random subspace

- **Random subspace**
  - 의사결정 나무의 분기점을 탐색할 때, 원래 변수의 수보다 적은 수의 변수를 임의로 선택하여 해당 변수들만을 고려 대상으로 함
  - 예시





Overview of the Random Forest

- **핵심 아이디어: Diversity, Random 확보**

1. 여러 개의 Training data를 생성하여 각 데이터마다 개별 의사결정나무 모델을 구축
   → Bagging → Diversity 확보

2. 의사결정나무 모델을 구축 시 변수를 무작위로 선택
   → Random subspace → Random 확보

Random Forest – Generalization Error

- 각각의 개별 tree는 과적합 될 수도 있음
- Random Forest는 tree 수가 충분히 많을 때 Strong Law of Large Numbers에 의해 과적합 되지 않고 그 에러는 limiting value에 수렴됨

$$Generalization\ Error \leq \frac{\bar{p}(1-s^2)}{s^2} \quad \rightarrow$$

Generalization Error의 Upper bound
(작으면 작을수록 좋음)

$\bar{p}$: Decision Tree 사이의 평균 상관관계 <= Decision Tree 간의 상관관계는 낮을수록(독립적) 좋음
s: 올바르게 예측한 tree와 잘못 예측한 tree 수 차이의 평균 <= s가 클수록 좋음 (정확도 좋음)

- 개별 tree의 정확도가 높을수록 s 증가
- Bagging과 Random Subspace 기법은 각 모델들의 독립성, 일반화, 무작위성을 최대화 시켜 모델간의 상관관계 p를 감소시킴
- 개별 Tree의 정확도, 독립성이 높을수록 Random Forest의 성능이 높아짐

Random Forest – Selection of important variable

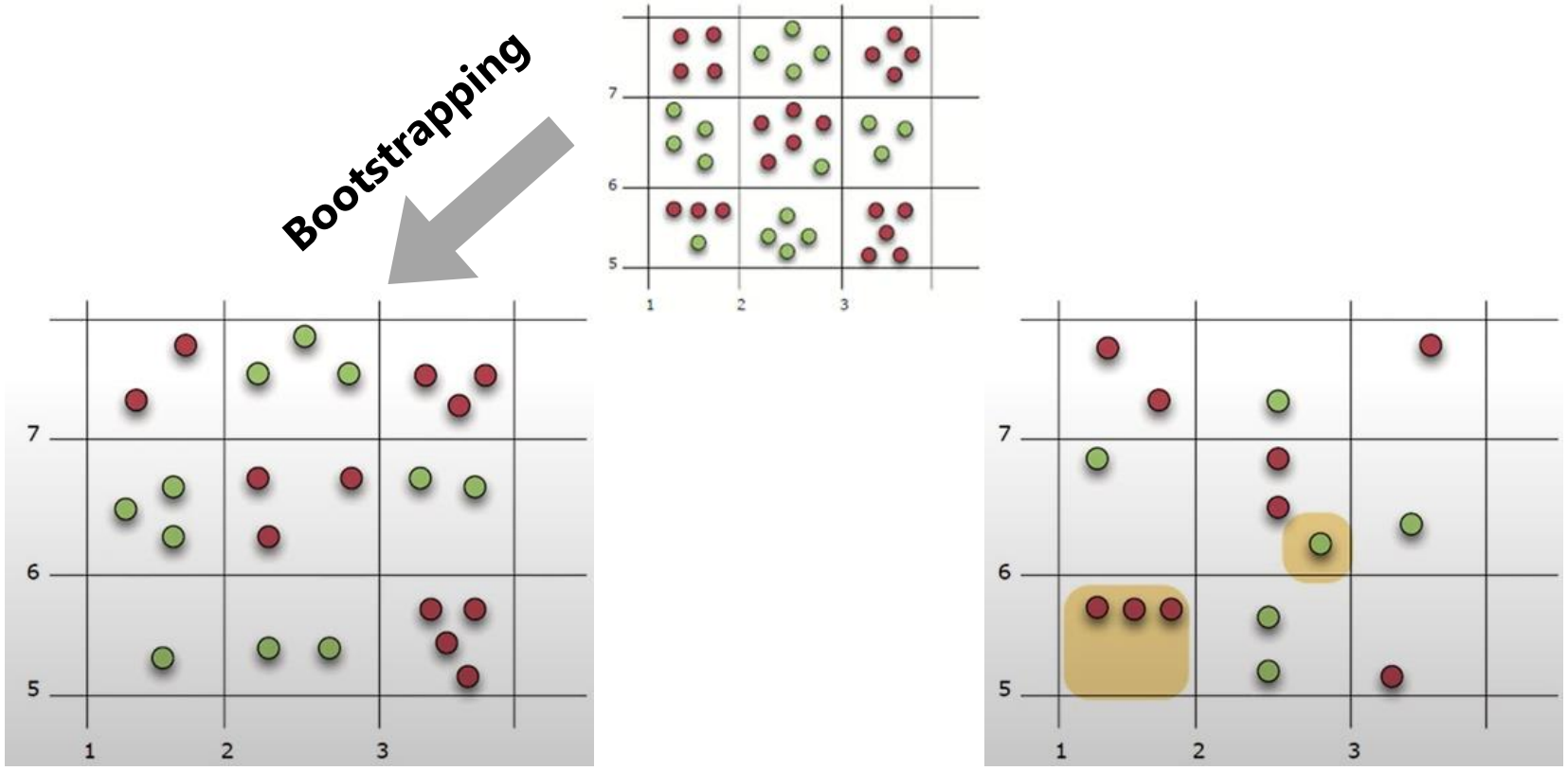
- **변수의 중요도**

  - Random Forest는 선형 회귀모델/로지스틱 회귀모델과는 달리 개별 변수가 통계적으로 얼마나 유의한지에 대한 정보를 제공하지 않음 (알려진 확률분포를 가정하지 않음 i.e., 비모수적 모델)

  - 대신 Random Forest는 다음과 같은 간접적인 방식으로 변수의 중요도를 결정
    - 1단계: 원래 Dataset에 대해 Out of Bag(OOB) Error를 구함
    - 2단계: 특정 변수의 값을 임의로 뒤섞은 Dataset에 대해 OOB Error를 구함
    - 3단계: 개별 변수의 중요도는 2단계와 1단계 OOB Error 차이의 평균과 분산을 고려하여 결정

Random Forest – Selection of important variable

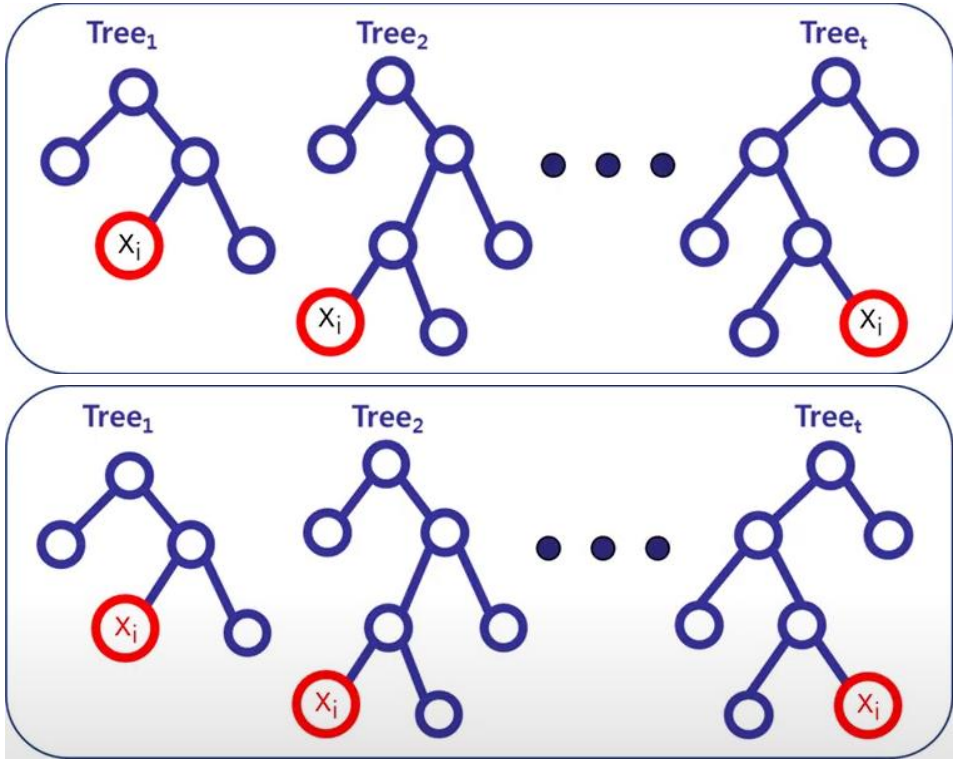
- **Out of Bag(OOB)**
  - Bagging을 사용할 경우 Bootstrap에 포함되지 않는 데이터들을 검증 집합으로 사용



Bootstrap에 포함된 data

Bootstrap에 포함되지 않은 data
(OOB Data)

Random Forest – Selection of important variable



(1) 각 Bootstrap으로부터 생성된 Tree에서 OOB Error 계산 (r_i = 1,2,…,t)

(2) Step (1) Tree에서 특정변수(X_i)의 값을 임의로 뒤섞은 Dataset에 대해 OOB Error 계산 (e_i = 1,2,…,t)

(3) $d_i = e_i - r_i \quad (i = 1,2,\dots,t)$

$$\bar{d} = \frac{1}{t}\sum_{i=1}^{t} d_i\,, \qquad {S_d}^2 = \frac{1}{t-1}\sum_{i=1}^{t}(d_i - \bar{d})^2$$

d_i의 평균 d_i의 분산

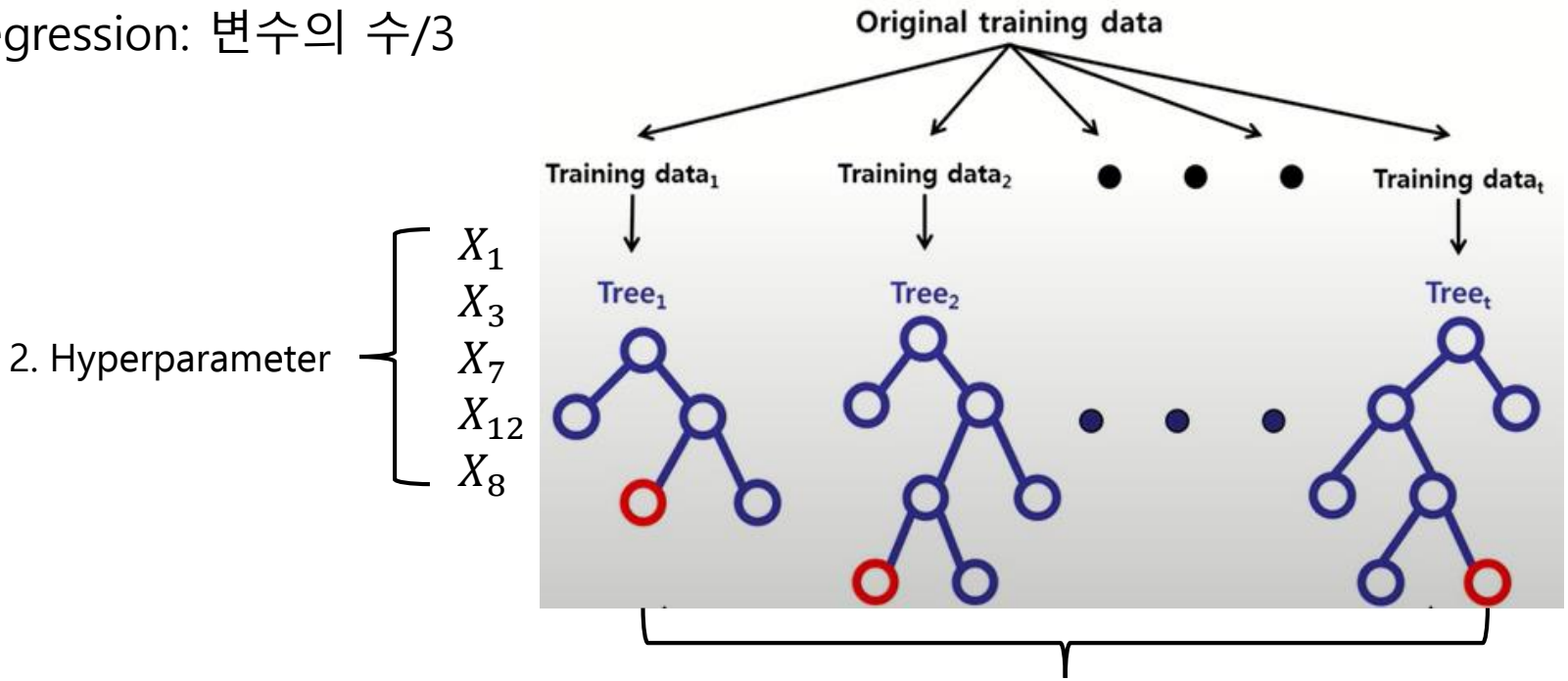
(4) X_i 변수 중요도: $v_i = \frac{\bar{d}}{s_d}$

d_i가 커진다? -> e와 r의 차이가 크다 -> 기존변수에 다른 값이 들어왔을 때 Error가 커진다
i.e., 기존 변수의 값이 다른값으로 대체되면 안된다 -> 변수 중요도가 높다

Random Forest – Hyper parameter

1. Decision Tree의 수
   - String Law of Large numbers를 만족시키기 위해 2,000개 이상의 Decision Tree 필요

2. Decision Tree에서 노드 분할 시 무작위로 선택되는 변수의 수
   - 일반적으로 변수의 수에 따라 다음과 같이 추천됨 (Dias-Uriarte et el., 2006)
   - Classification: $\sqrt{\text{변수의 수}}$
   - Regression: 변수의 수/3



4. Boosting

Overview of the Boosting

- Boosting Idea

  - 여러 개의 learning model을 순차적으로 구축하여 최종적으로 합침 (Ensemble)
  - 여기서 사용하는 learning model은 매우 단순한 모델
    - ✓ 단순한 모델: Model that slightly better than chance
  - 순차적 → 모델 구축에 순서를 고려
  - 각 단계에서 새로운 base learner를 학습하여 이전 단계의 base learner의 단점을 보완
  - 각 단계를 거치면서 모델이 점차 강해짐 → boosting

Types of Boosting Algorithm

- Adaptive Boosting (AdaBoost)
- Gradient Boosting Machines (GBM)
- XGBoost
- Light Gradient Boost Machines (Light GBM)
- CatBoost

Adaptive Boosting (AdaBoost)

❖ Adaptive Boosting (AdaBoost)

- 각 단계에서 새로운 base learner를 학습하여 이전 단계의 base learner의 단점을 보완
- Training error가 큰 관측치의 선택 확률(가중치)를 높이고, Training error가 작은 관측치의 선택 확률을 낮춤
  - ✓ 오분류한 관측치에 보다 집중!
- 앞 단계에서 조정된 확률(가중치)을 기반으로 다음 단계에서 사용될 Training dataset을 구성
- 다시 첫 단계로 감
- 최종 결과물은 각 모델의 성능지표를 가중치로 하여 결합 (Ensemble)

Adaptive Boosting (AdaBoost)

- AdaBoost Algorithm

1. *Set* $W_i = \frac{1}{n}$, *i = 1,2, …, n (impose equal weight initially)*

2. *for j=1 to m (m: number of classifiers)*

   Step1. Find $h_j(x)$ *that minimizes* L_j *(weighted loss function)*

   $$L_j = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_j(x))}{\sum_{i=1}^{n} W_i}$$

   Step2. Define the weight of a classifier: $a_j = log\left(\frac{1-L_j}{L_j}\right)$

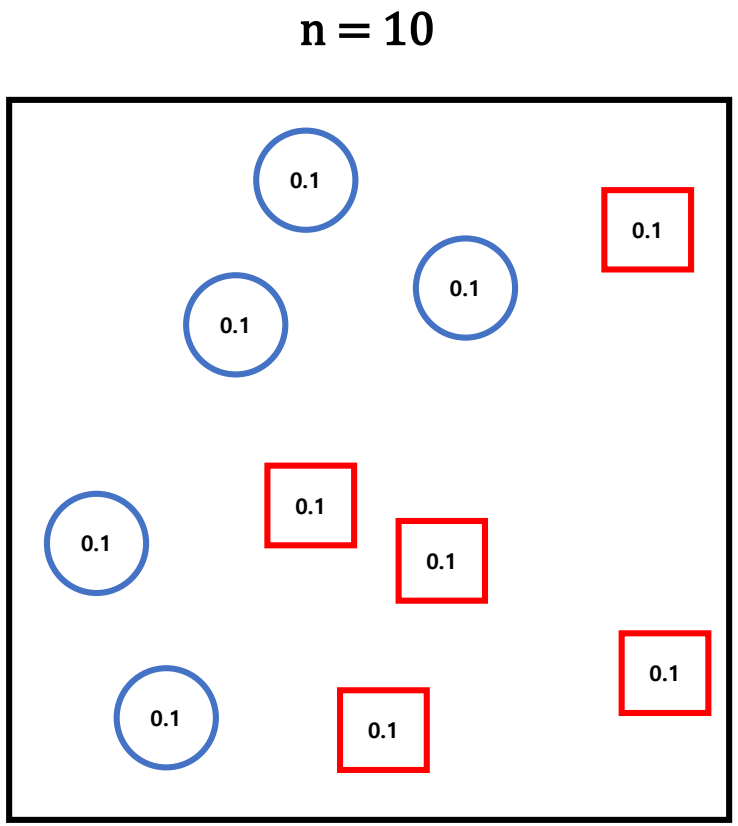
   Step3. Update weight: $W_i \leftarrow W_i e^{a_j I(y_i \neq h_j(x))}$, *i=1,2,…,n*

   End for

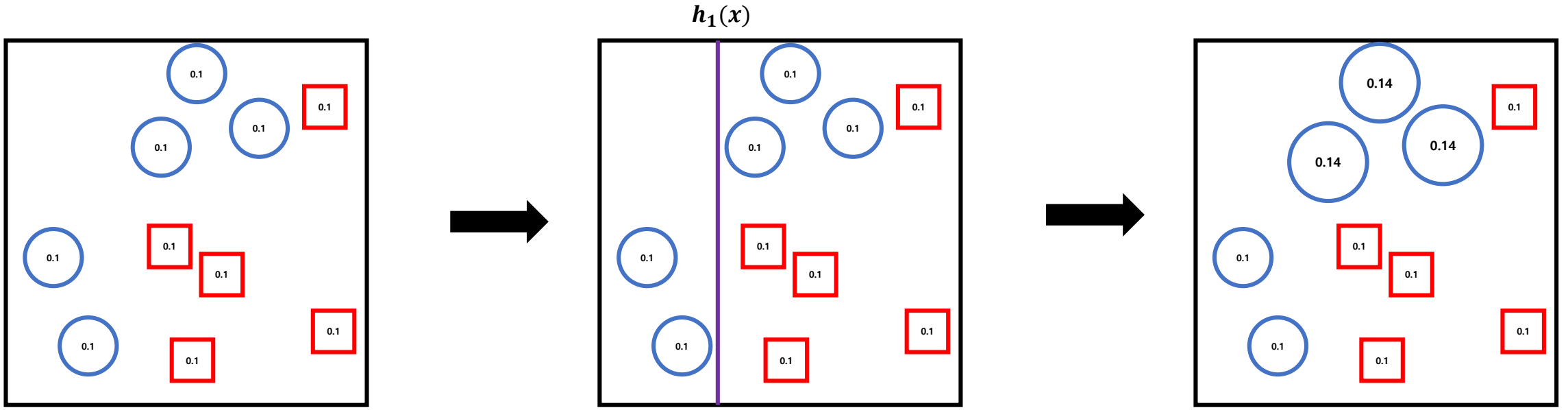
3. *Find boosted model:* $h(x) = sign[\sum_{i=1}^{m} a_j h_j(x)]$

Adaptive Boosting (AdaBoost)

Set $W_i = \frac{1}{n}$, *i=1,2, …, n* (impose equal weight initially)



Adaptive Boosting (AdaBoost)



$$L_1 = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_j(x))}{\sum_{i=1}^{n} W_i} = \frac{0.1 \times 3}{0.1 \times 10} = 0.3$$ ← 10개 중 3개 오분류

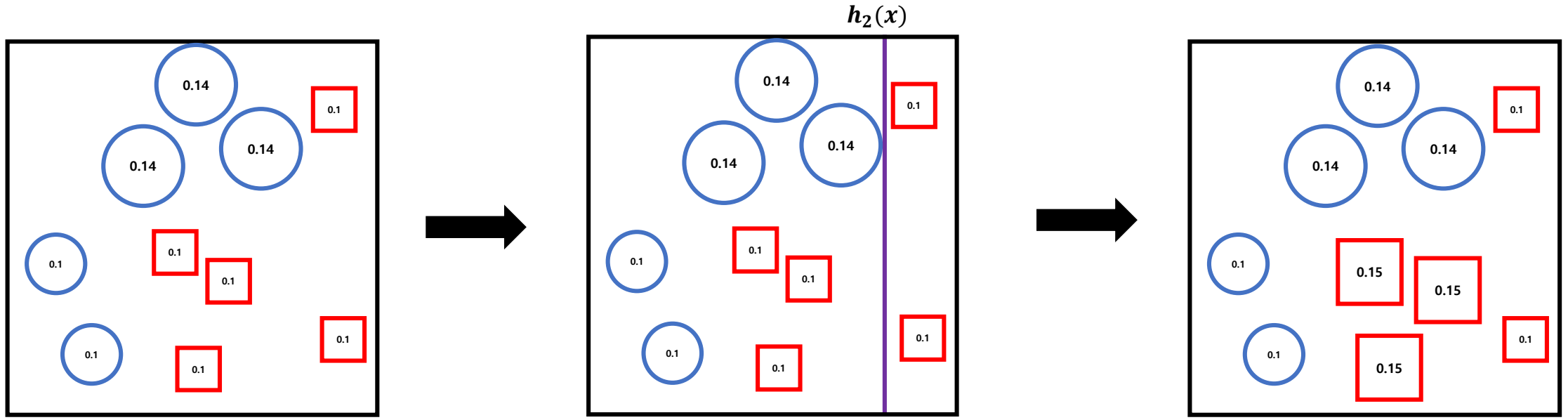
$$a_1 = log\left(\frac{1 - L_j}{L_j}\right) = log\left(\frac{1 - 0.3}{0.3}\right) \approx 0.37$$

$W_c \leftarrow W_i e^{a_1 I(y_i \neq h_1(x))} = 0.1e^{0.37 \times 0} = 0.1$ ← 정분류 관측치 가중치

$W_{nc} \leftarrow W_i e^{a_1 I(y_i \neq h_1(x))} = 0.1e^{0.37 \times 1} = 0.14$ ← 오분류 관측치 가중치

- 도형의 크기(안의 숫자)는 해당 관측치가 다음 단계의 학습 dataset에 선택될 확률을 의미

Adaptive Boosting (AdaBoost)



$$L_2 = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_j(x))}{\sum_{i=1}^{n} W_i} = \frac{0.1 \times 3}{0.1 \times 7 + 0.14 \times 3} = 0.3 \quad \leftarrow 10\text{개 중 } 3\text{개 오분류}$$

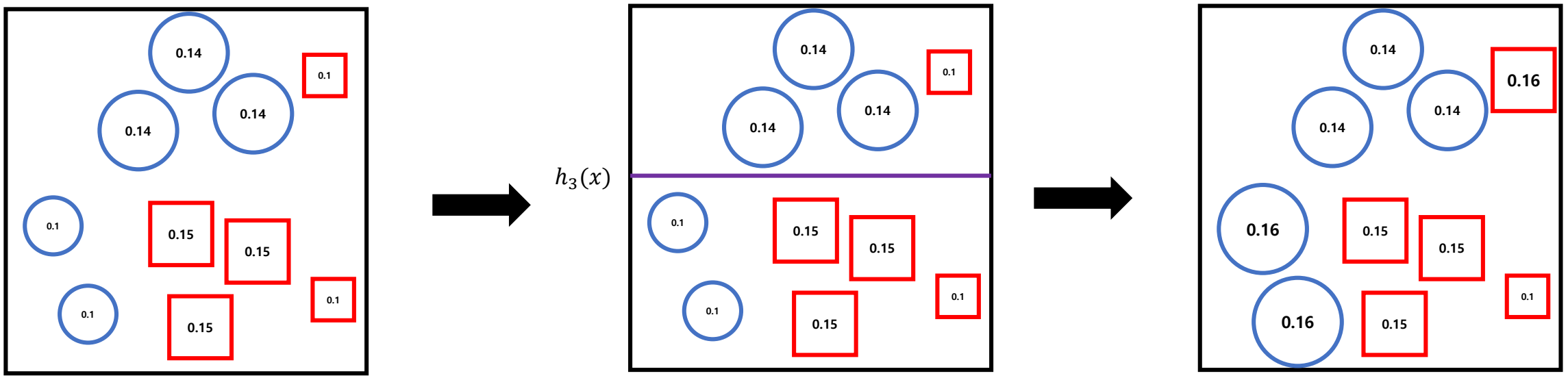
$$a_2 = log\left(\frac{1 - 0.27}{0.27}\right) \approx 0.43$$

$$W_c \leftarrow W_i e^{a_2 I(y_i \neq h_2(x))} = \{0.1 \; or \; 0.14\} e^{0.43 \times 0} = 0.1 \text{ or } 0.14 \quad \leftarrow \text{정분류 관측치 가중치}$$

$$W_{nc} \leftarrow W_i e^{a_2 I(y_i \neq h_2(x))} = 0.1 e^{0.43 \times 1} = 0.15 \quad \leftarrow \text{오분류 관측치 가중치}$$

- 도형의 크기(안의 숫자)는 해당 관측치가 다음 단계의 학습 dataset에 선택될 확률을 의미

Adaptive Boosting (AdaBoost)



$$L_3 = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_j(x))}{\sum_{i=1}^{n} W_i} = \frac{0.1 \times 3}{0.1 \times 7 + 0.14 \times 3 + 0.15 \times 3} = 0.24$$ ← 10개 중 2개 오분류

$$a_3 = log\left(\frac{1 - 0.24}{0.24}\right) \approx 0.5$$

$W_c \leftarrow W_i e^{a_3 I(y_i \neq h_3(x))} = \{0.1 \; or \; 0.14 \; or \; 0.15\} e^{0.5 \times 0} = 0.1 \text{ or } 0.14 \text{ or } 0.15$ ← 정분류 관측치 가중치

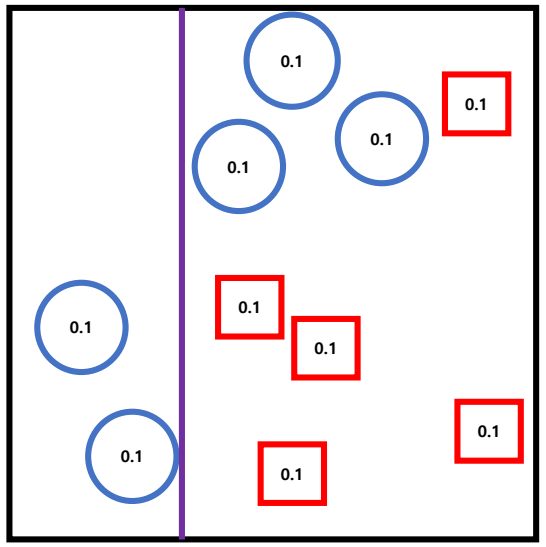
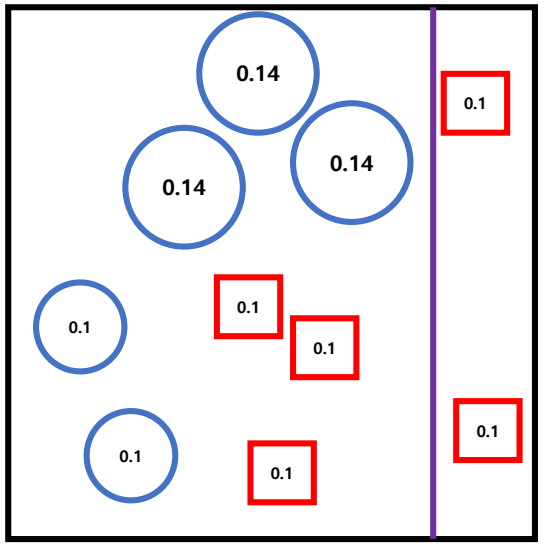
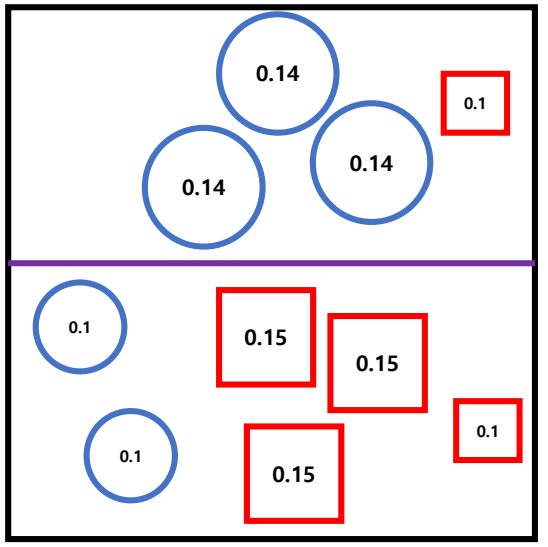
$W_{nc} \leftarrow W_i e^{a_3 I(y_i \neq h_3(x))} = 0.1 e^{0.5 \times 1} = 0.16$ ← 오분류 관측치 가중치

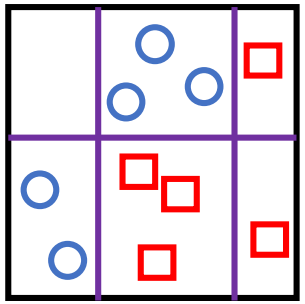
- 도형의 크기(안의 숫자)는 해당 관측치가 다음 단계의 학습 dataset에 선택될 확률을 의미

Adaptive Boosting (AdaBoost)

$$h(x) = sign\left[\sum_{i=1}^{m=3} a_j h_j(x)\right]$$

$h_1(x)$ $h_2(x)$ $h_3(x)$

$h(x) = sign$ [0.37  + 0.43  + 0.5 ]

$=$ 

Adaptive Boosting (AdaBoost)

- AdaBoost Algorithm

1. *Set* $W_i = \frac{1}{n}$, *i = 1,2, …, n (impose equal weight initially)*

2. *for j=1 to m (m: number of classifiers)*

   Step1. Find $h_j(x)$ *that minimizes* L_j *(weighted loss function)*

   $$L_j = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_j(x))}{\sum_{i=1}^{n} W_i}$$

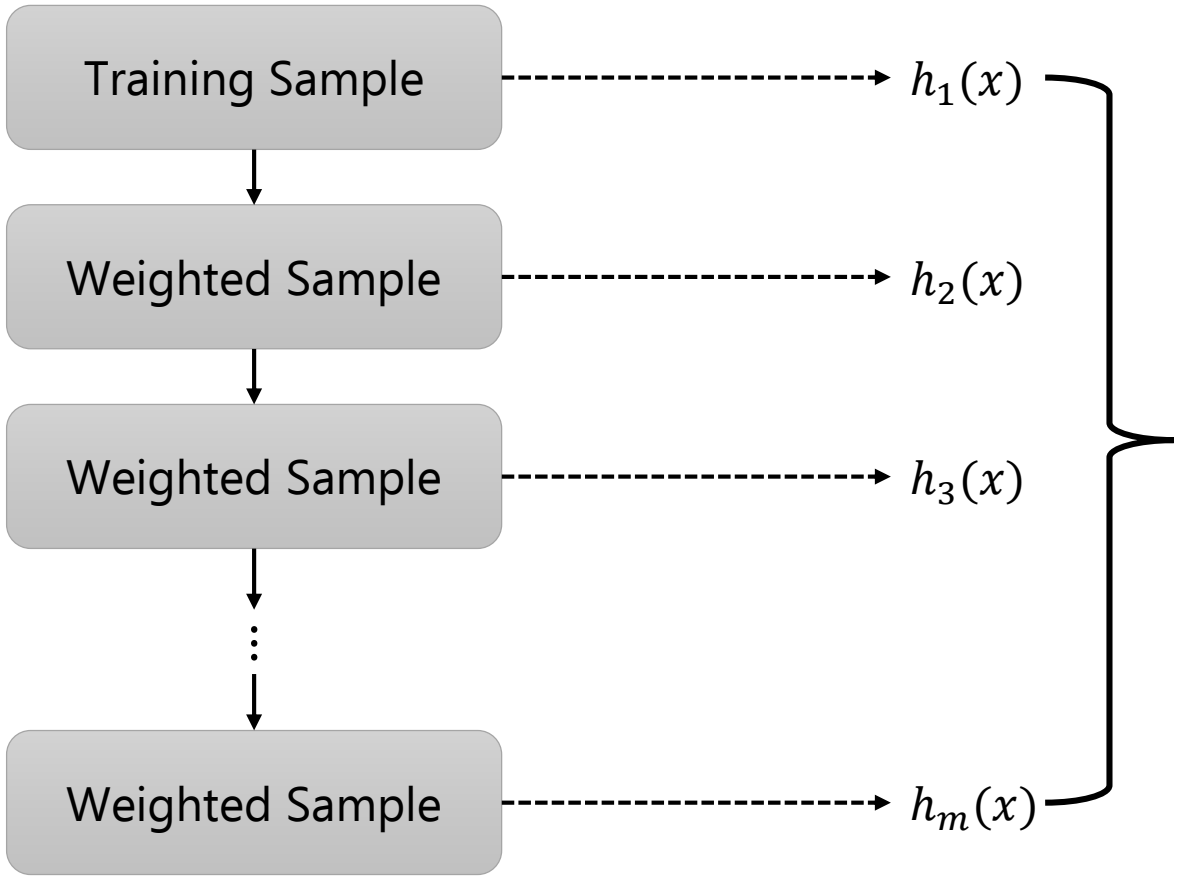
   Step2. Define the weight of a classifier: $a_j = log\left(\frac{1-L_j}{L_j}\right)$

   Step3. Update weight: $W_i \leftarrow W_i e^{a_j I(y_i \neq h_j(x))}$, *i=1,2,…,n*

   End for

3. *Find boosted model:* $h(x) = sign[\sum_{i=1}^{m} a_j h_j(x)]$

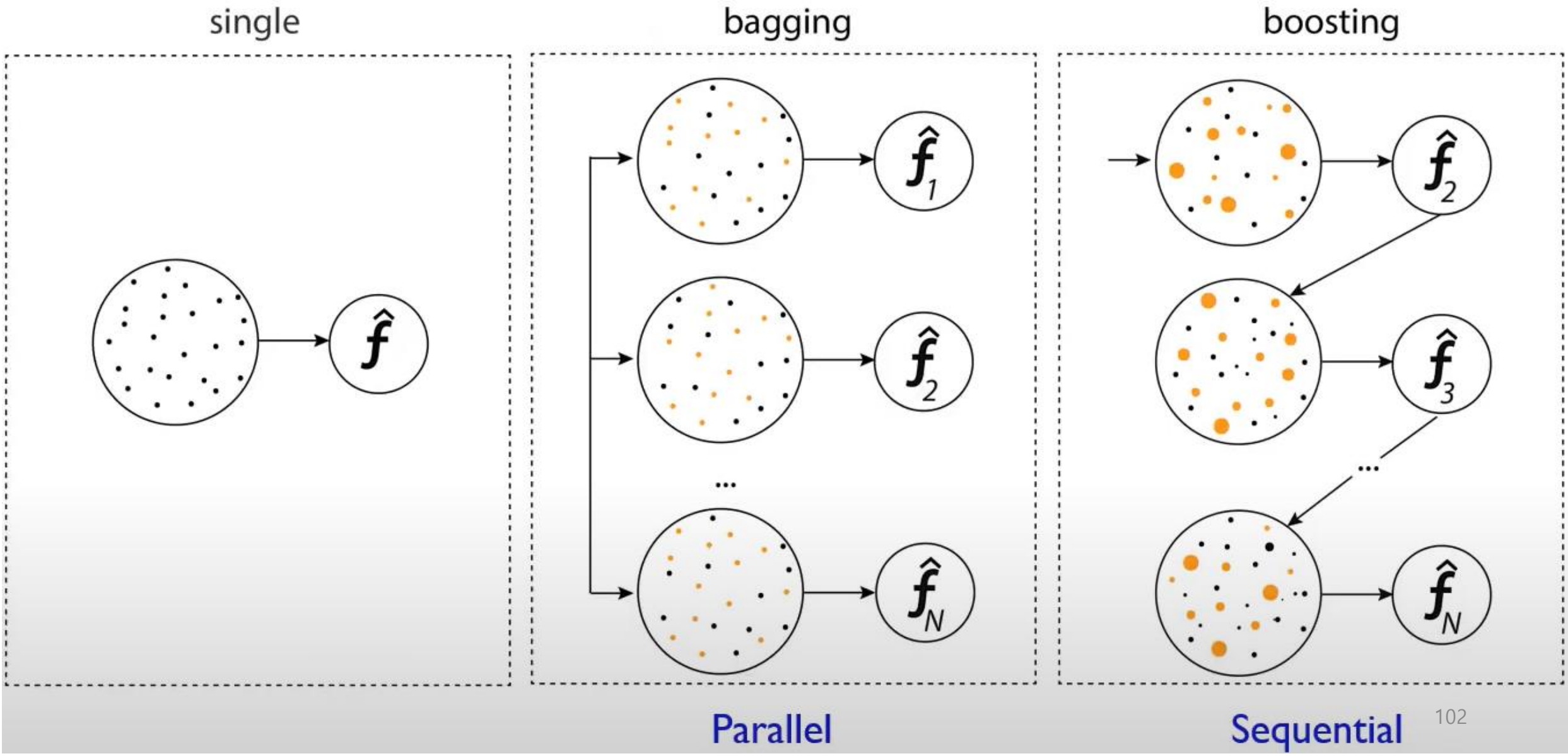
Adaptive Boosting (AdaBoost)



$$h(x) = \text{sign}\left[\sum_{i=1}^{m} a_j h_j(x)\right]$$

$h_1(x)$를 만들고 이를 바탕으로 $h_2(x)$를 만들고, 이를 바탕으로 $h_3(x), \ldots$

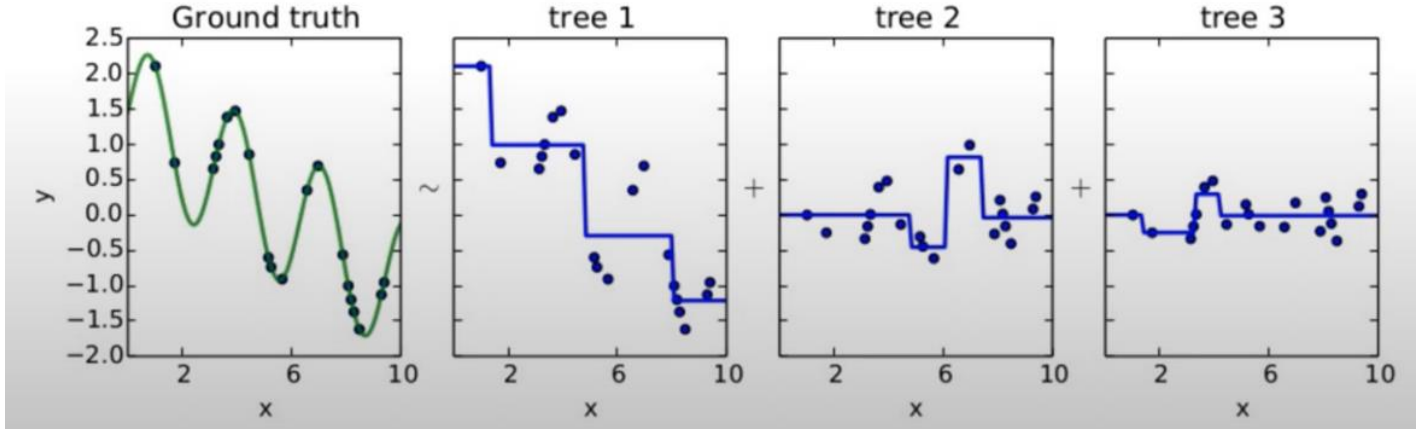
Bagging vs Boosting



Gradient Boosting Machines (GBM)

❖ GBM

- Gradient Boosting = Boosting with Gradient Descent
- 첫 번째 단계에서 모델 tree1통해 Y를 예측 / 두 번째 단계 모델 tree2 통해 Residual 예측
  ⟶ 여기서 발생한 Residual을 모델 tree3로 예측
- 점차 Residual 작아짐
- Gradient Boosting Model = tree1 + tree2 + tree3



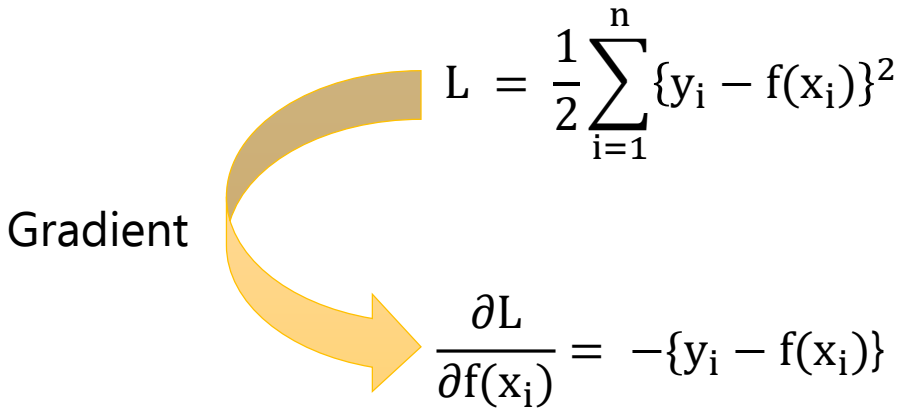
Gradient Boosting Machines (GBM)

- Why Gradient?

$$L = \frac{1}{2}\sum_{i=1}^{n}\{y_i - f(x_i)\}^2$$

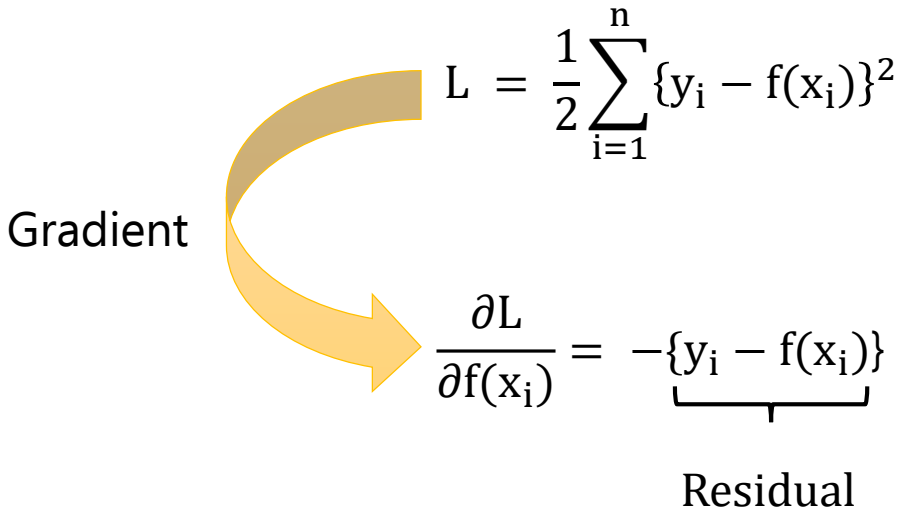
Gradient Boosting Machines (GBM)

- Why Gradient?



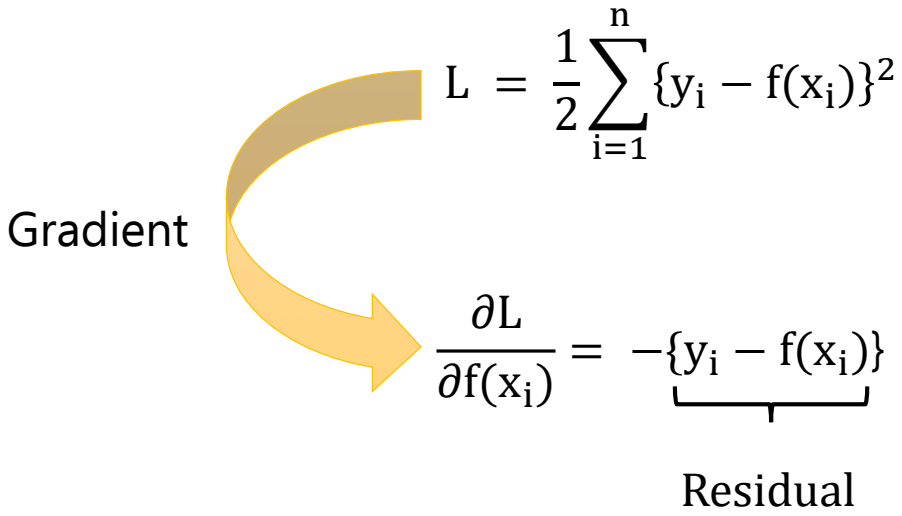
Gradient Boosting Machines (GBM)

- Why Gradient?



Gradient Boosting Machines (GBM)

- Why Gradient?



$$y_i - f(x_i) = -\frac{\partial L}{\partial f(x_i)}$$

Gradient Boosting Machines (GBM)

- Why Gradient?

$$L = \frac{1}{2}\sum_{i=1}^{n}\{y_i - f(x_i)\}^2$$

Gradient

$$\frac{\partial L}{\partial f(x_i)} = -\underbrace{\{y_i - f(x_i)\}}_{\text{Residual}}$$

$$y_i - f(x_i) = -\frac{\partial L}{\partial f(x_i)}$$

Residual = Negative Gradient

Gradient Boosting Machines (GBM)

Original Dataset

x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4
x_5	y_5
x_6	y_6
x_7	y_7
x_8	y_8
x_9	y_9
x_{10}	y_{10}

Modified Dataset #1

x_1	y_1-$f_1(x_1)$
x_2	y_2-$f_1(x_2)$
x_3	y_3-$f_1(x_3)$
x_4	y_4-$f_1(x_4)$
x_5	y_5-$f_1(x_5)$
x_6	y_6-$f_1(x_6)$
x_7	y_7-$f_1(x_7)$
x_8	y_8-$f_1(x_8)$
x_9	y_9-$f_1(x_9)$
x_{10}	y_{10}-$f_1(x_{10})$

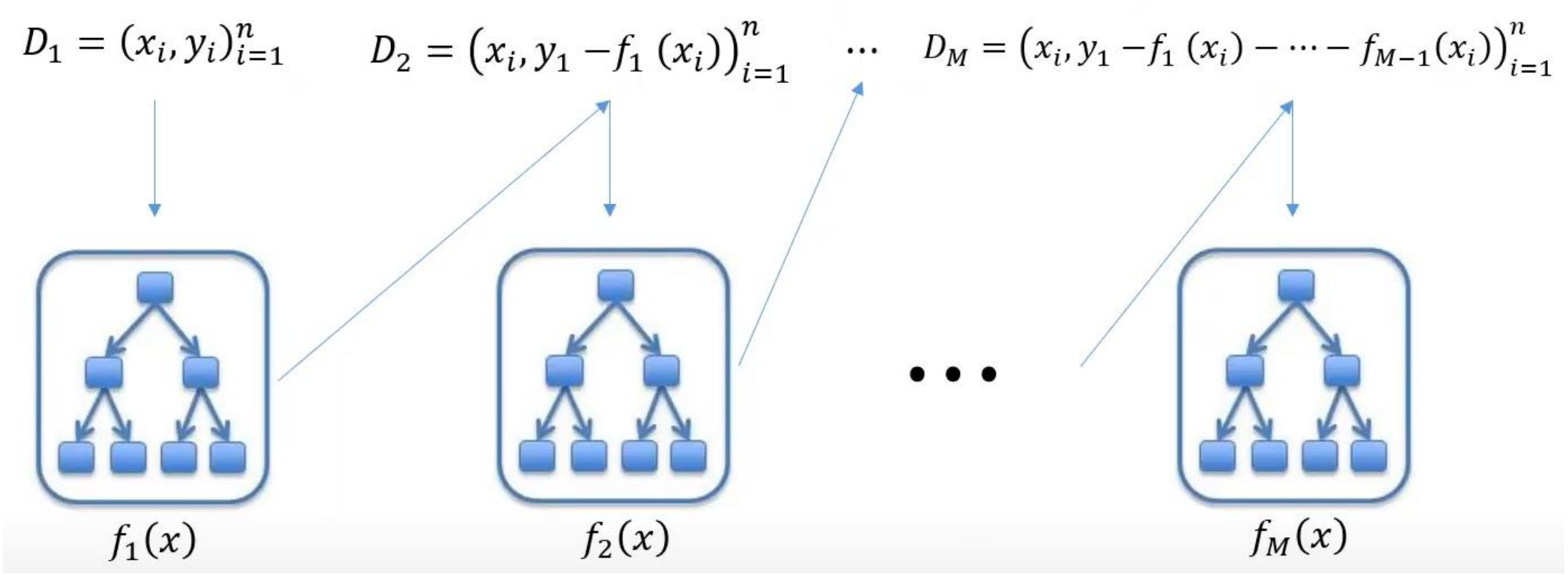
Modified Dataset #2

x_1	y_1-$f_1(x_1)$-$f_2(x_1)$
x_2	y_2-$f_1(x_2)$-$f_2(x_2)$
x_3	y_3-$f_1(x_3)$-$f_2(x_3)$
x_4	y_4-$f_1(x_4)$-$f_2(x_4)$
x_5	y_5-$f_1(x_5)$-$f_2(x_5)$
x_6	y_6-$f_1(x_6)$-$f_2(x_6)$
x_7	y_7-$f_1(x_7)$-$f_2(x_7)$
x_8	y_8-$f_1(x_8)$-$f_2(x_8)$
x_9	y_9-$f_1(x_9)$-$f_2(x_9)$
x_{10}	y_{10}-$f_1(x_{10})$-$f_2(x_{10})$

$y = f_1(x)$ $\quad$ $y - f_1(x) = f_2(x)$ $\quad$ $y - f_1(x) - f_2(x) = f_3(x)$

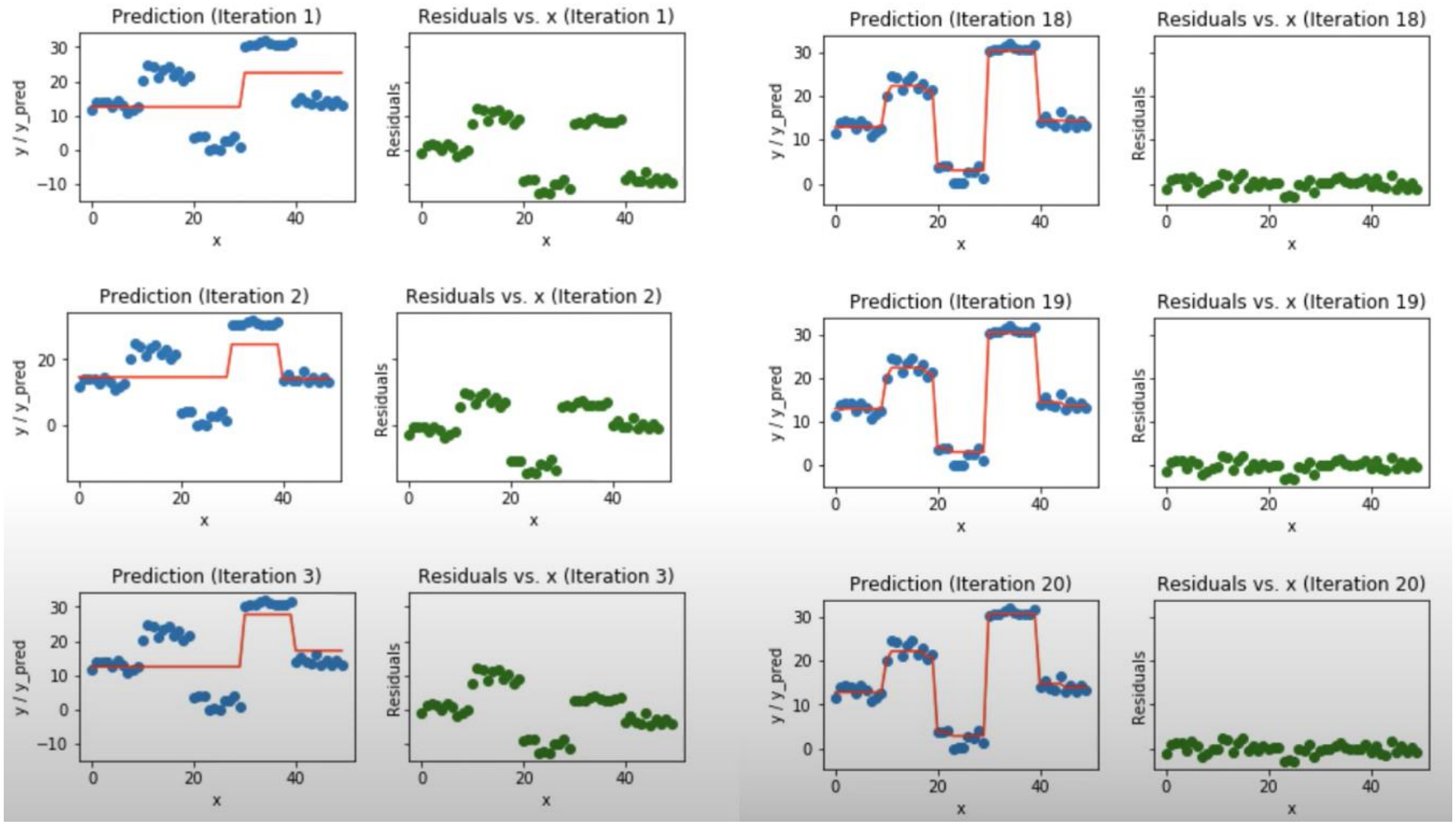
Gradient $y - f_1(x)$ $\quad$ $\{y - f_1(x)\} - f_2(x)$ $\quad$ $\{y - f_1(x) - f_2(x)\} - f_3(x)$

Gradient Boosting Machines (GBM)



$$\hat{f}(x) = f_1(x) + f_2(x) + \cdots + f_M(x)$$

Gradient Boosting Machines (GBM)



Thank you