

Decision Tree & Ensemble

23.02.02 / 8기 장준혁

CONTENTS

01. Decision Tree

- Overview
- Entropy

02. CART

- Gini Impurity & IG
- Binary Tree
- 가지치기
- Regression Tree

03. Ensemble

- Overview
- Voting
- Algorithm

04. Bagging

- Bootstrap Sampling
- Random Forest

05. Boosting

- Boosting
- GBM
- XGBoost
- LightGBM & CatBoost

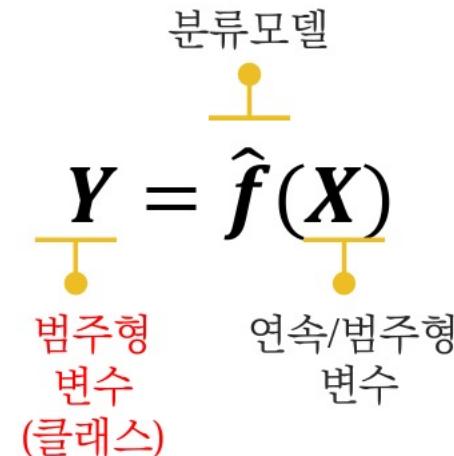
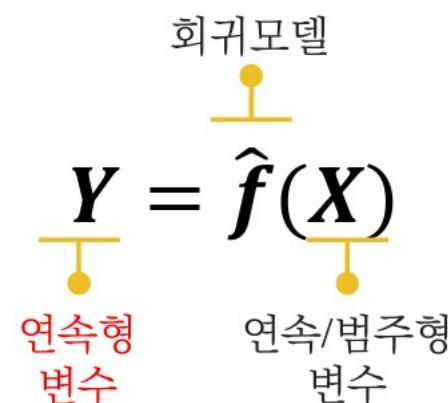
06. SUMMARY

- Summary
- Reference

Decision Tree

Decision Tree

- 지도학습의 종류 : 회귀(Regression)/분류(Classification)
- 데이터의 형태에 따라 어떤 방법을 사용할지 결정해야 함
- Decision Tree는 기본적으로 Classification Task 를 위한 방법론 → Regression Tree로 확장 가능



회귀(Regression)

Linear Regression

분류(Classification)

Logistic Regression

SVM

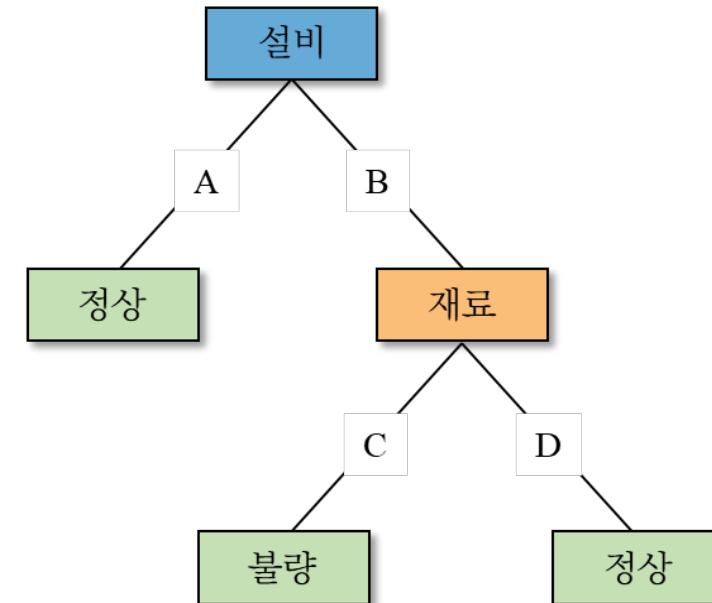


1. Decision Tree

Overview

Decision Tree란?

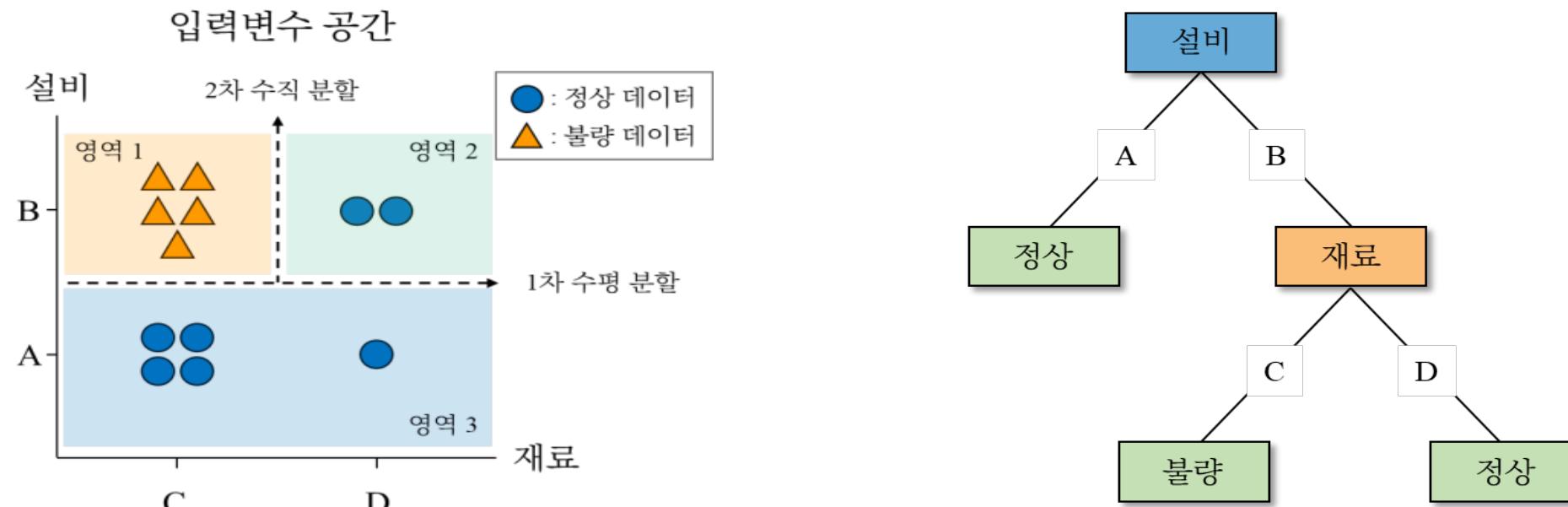
- 의사결정나무는 모델의 의사결정 규칙(Decision Rule)을 나무 형태로 표현하는 모델
 - 나무의 경로는 하나의 의사결정 규칙을 의미하며, 입력변수 데이터가 들어오면 규칙에 따라 범주형 출력변수를 예측
- 일련의 필터 과정 또는 스무고개



1. Decision Tree

Overview

의사결정나무는 분할된 영역에 동일한 클래스 데이터가 최대한 많이 존재하도록
각축으로 영역을 분할하여 생성됨



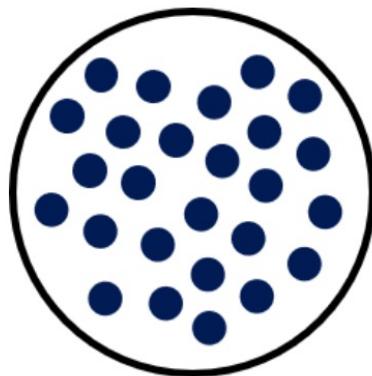
"변별력이 좋은 질문을 정하기" → 불순도에 대한 정의가 필요

1. Decision Tree

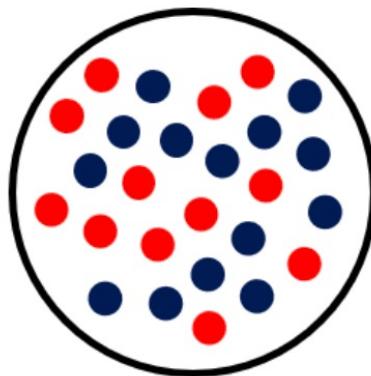
Entropy

불순도 (Impurity)

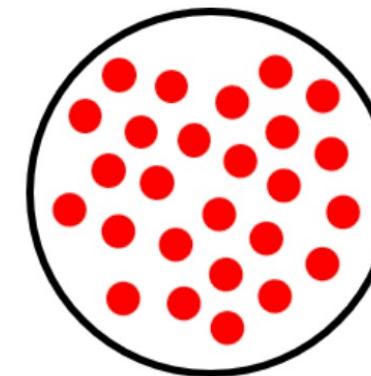
- 불순물이 포함된 정도 = 해당 범주 안에 서로 다른 데이터가 얼마나 섞여 있는지
- 결정 트리는 불순도를 최소화하는 방향으로 학습을 진행



불순도 = 0



불순도 > 0



불순도 = 0

1. Decision Tree

Entropy

Entropy

- 사건 X_i 가 발생할 확률 : $probability = p(x)$
- 사건 X_i 가 갖는 정보량(Information, 놀람의 정도) : $information = I(x) = \log_2 \frac{1}{p(x)}$

$$Entropy = H(S) = \sum_{i=1}^c p_i \log_2 \frac{1}{p_i} = - \sum_{i=1}^c p_i \log_2 p_i$$

- 정보이득(Information Gain) : 분기 전후 엔트로피의 차이값. IG가 가장 큰 지표를 선택해야 함
- ID3, C4.5, C5.0 알고리즘에서 엔트로피를 불순도 지표로 사용

2. CART(Classification and Regression Tree)

Gini impurity & IG

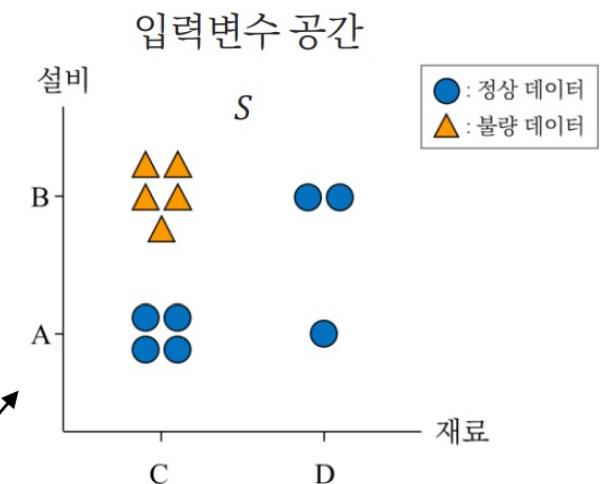
지니 불순도(Gini Impurity)

- CART에서는 클래스 동질성을 측정하는 지표로 지니 불순도(Gini Impurity)를 사용함
- 현 상태에서 더 데이터를 나눠야 하는가에 대한 지표로 작용
- 잘못 분류될 확률을 최소화하기 위한 기준

$$Gini(S) = 1 - p_+^2 - p_-^2$$

- S : 분할된 영역 내에 존재하는 데이터 집합(Set)
- P+ : '정상' 데이터의 비율
- P- : '불량' 데이터의 비율

$$Gini(S) = 1 - \left(\frac{7}{12}\right)^2 - \left(\frac{5}{12}\right)^2 = 0.486$$

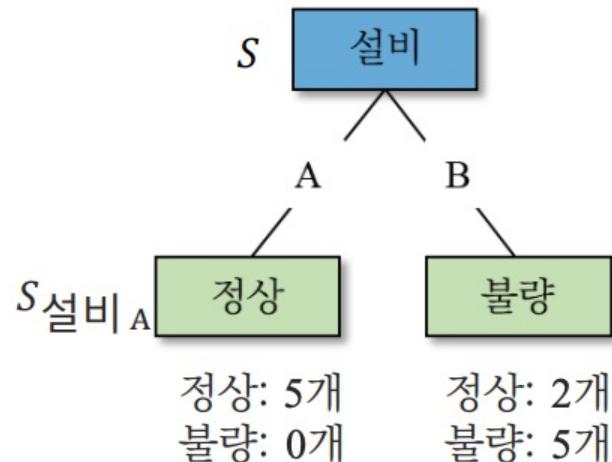


2. CART(Classification and Regression Tree)

Gini impurity & IG

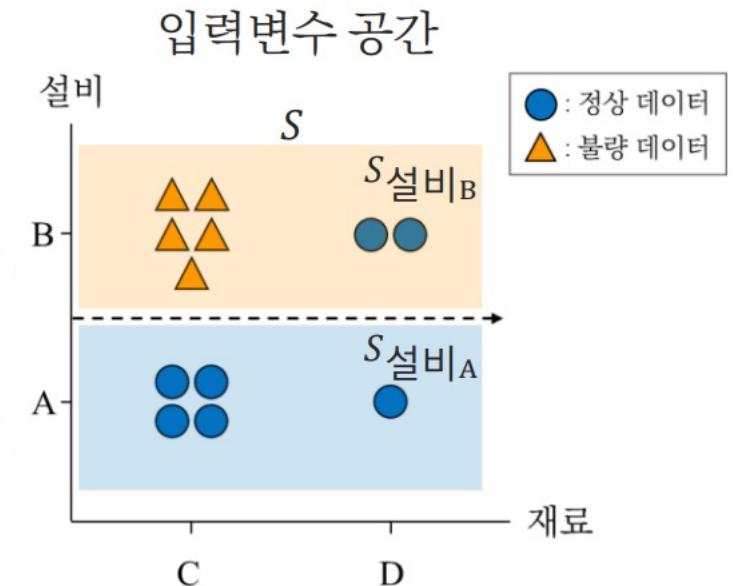
지니 불순도(Gini Impurity)

$$Gini(S) = 1 - p_+^2 - p_-^2$$



$$Gini(S_{설비_B}) = 1 - \left(\frac{5}{7}\right)^2 - \left(\frac{2}{7}\right)^2 = 0.408$$

$$Gini(S_{설비_A}) = 1 - \left(\frac{5}{5}\right)^2 - \left(\frac{0}{5}\right)^2 = 0$$



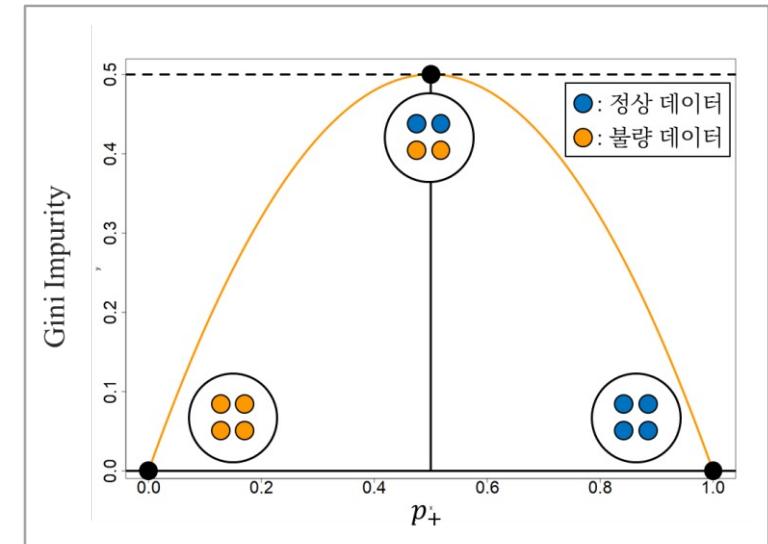
2. CART(Classification and Regression Tree)

Gini impurity & IG

지니 불순도(Gini Impurity)

- 출력변수의 클래스가 C개 존재할 때 지니 불순도 $Gini(S) = 1 - \sum_{i=1}^c p_i^2$
- 두 개의 데이터를 랜덤하게 골랐을 때 두 데이터의 클래스가 서로 다른 확률을 의미

$$\begin{aligned}
 Gini(S) &= \sum_{i=1}^c \sum_{i' \neq i} p_i p_{i'} = \sum_{i=1}^c p_i \sum_{i' \neq i} p_{i'} = \sum_{i=1}^c p_i (1 - p_i) \\
 &= 1 - \sum_{i=1}^c p_i^2
 \end{aligned}$$



- CART 알고리즘에서는 모든 조합에 대해 Gini Impurity를 계산한 후, 가장 낮은 지표를 찾아 분기

2. CART(Classification and Regression Tree)

Gini impurity & IG

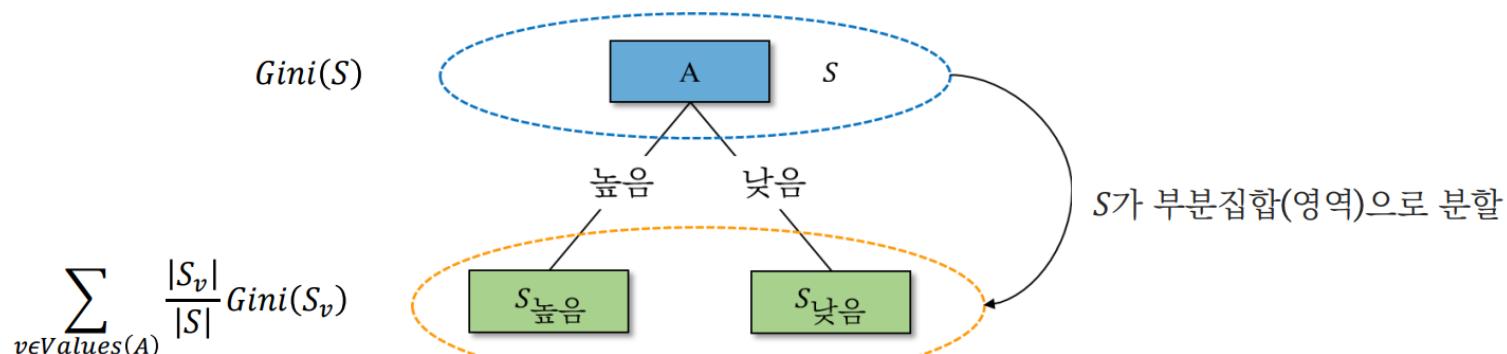
정보 이득(Information Gain)

- Information_Gain(S, A)는 영역 S 의 데이터를 입력변수 A 로 분할하는 경우의 혼잡도 감소량
- 어떤 질문을 기준으로 나눠야 하는가에 대한 지표로서 작용

$$\text{Information_Gain}(S, A) = Gini(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Gini(S_v)$$

입력변수 A 로 데이터를 분류하기 전의 혼잡도

입력변수 A 로 데이터를 분류한 후의 가중 평균 혼잡도



2. CART(Classification and Regression Tree)

Gini impurity & IG

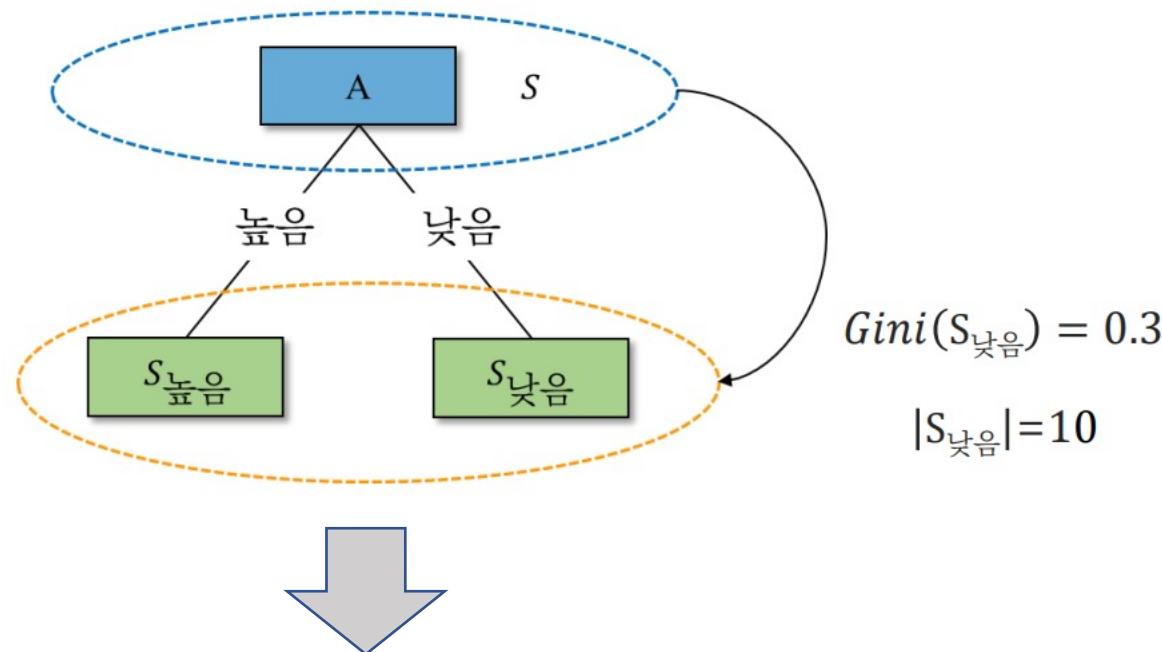
정보 이득(Information Gain)

$$Gini(S) = 0.5$$

$$|S| = 100$$

$$Gini(S_{\text{높음}}) = 0.3$$

$$|S_{\text{높음}}| = 90$$



$$\begin{aligned}
 & Gini(S) - \frac{90}{100} Gini(S_{\text{높음}}) - \frac{10}{100} Gini(S_{\text{낮음}}) \\
 &= 0.5 - 0.9 \times 0.3 - 0.1 \times 0.3 = 0.5 - 0.27 - 0.03 \\
 &= 0.2
 \end{aligned}$$

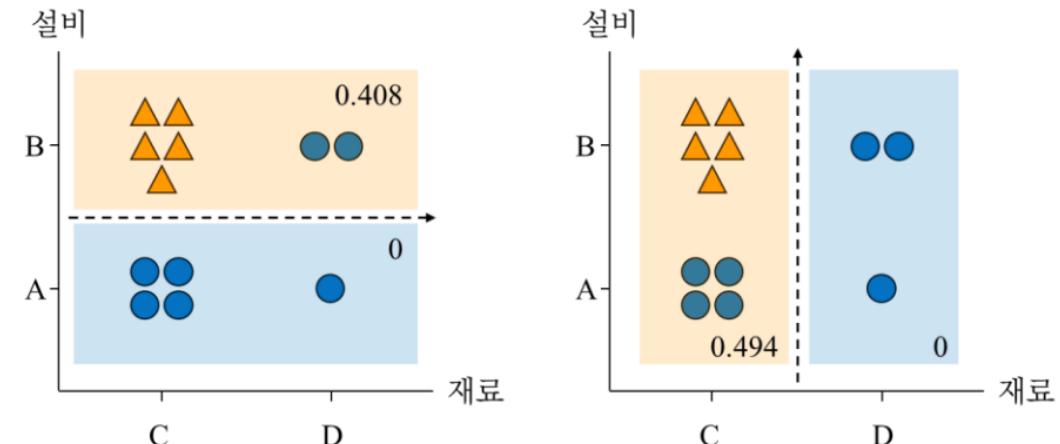
2. CART(Classification and Regression Tree)

Gini impurity & IG

정보 이득(Information Gain)

- 결정 트리 알고리즘은 정보 이득을 최대화하는 방향으로 학습을 결정
- 즉, '설비' 변수로 분할했을 때 감소되는 혼잡도의 양

이 더 크므로(정보이득이 최대) 뿌리 노드에서 '설비' 변수 사용



$$\text{Information_Gain}(S, \text{설비}) = 0.486 - \left(\frac{5}{12} \times 0 + \frac{7}{12} \times 0.408 \right) = 0.248$$

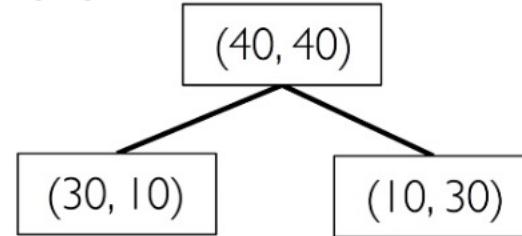
$$\text{Information_Gain}(S, \text{재료}) = 0.486 - \left(\frac{3}{12} \times 0 + \frac{9}{12} \times 0.494 \right) = 0.116$$

2. CART(Classification and Regression Tree)

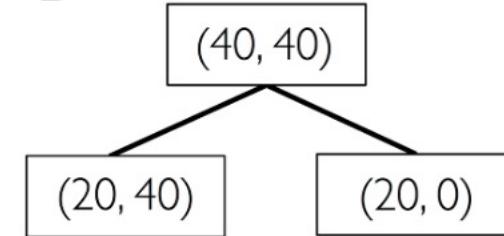
Gini impurity & IG

정보 이득(Information Gain) – Entropy & Gini Impurity 계산

A



B

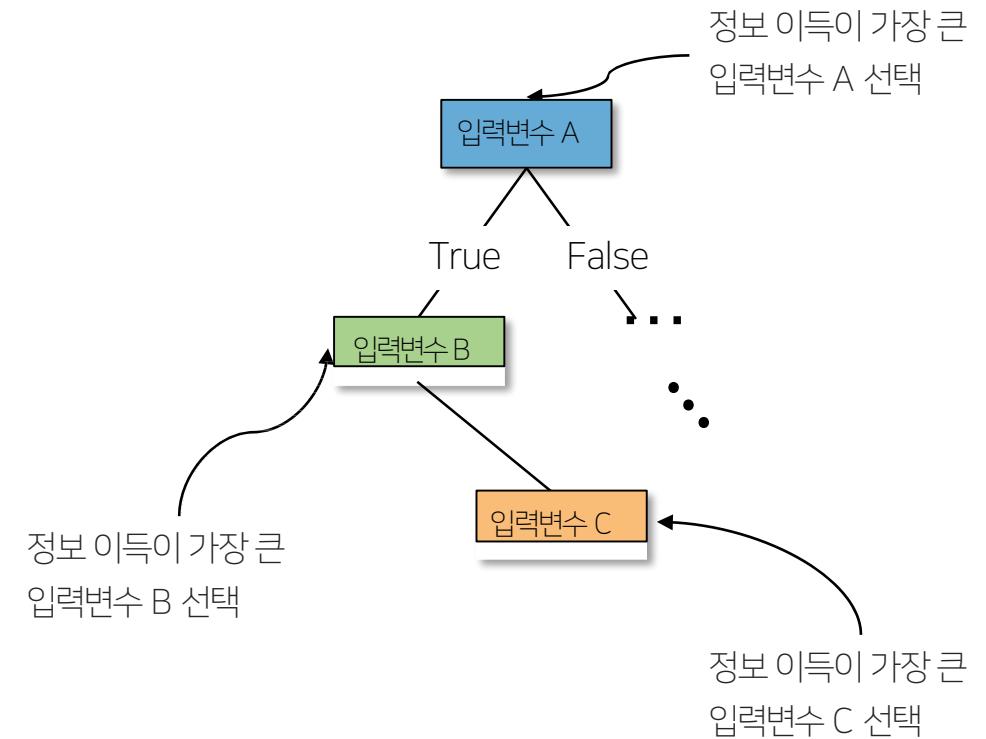


2. CART(Classification and Regression Tree)

Binary Tree

나무 구조 생성 과정

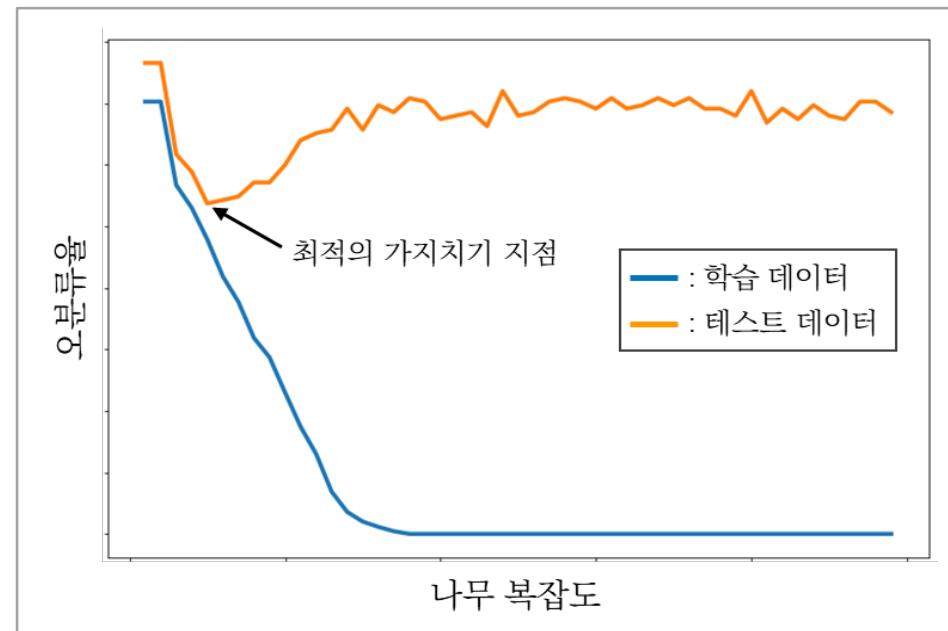
- CART는 가지 분기 시, 이진 분할(Binary Split) 수행함
- 입력변수들 중에서 IG가 가장 큰 입력변수 선택하여 입력 변수 공간을 분할
- 분할된 영역마다 다시 IG가 가장 큰 입력변수를 선택하여 입력변수 공간을 분할
- 분류 정확도가 최대화 될 때까지 재귀적으로 반복



2. CART(Classification and Regression Tree)

가지치기

- 나무에 가지가 너무 많으면(복잡한 모델) Overfitting 문제가 발생함
- 이러한 문제를 막기 위해 의사결정나무는 끝 노드의 일부를 제거하는 가지치기 작업을 수행함

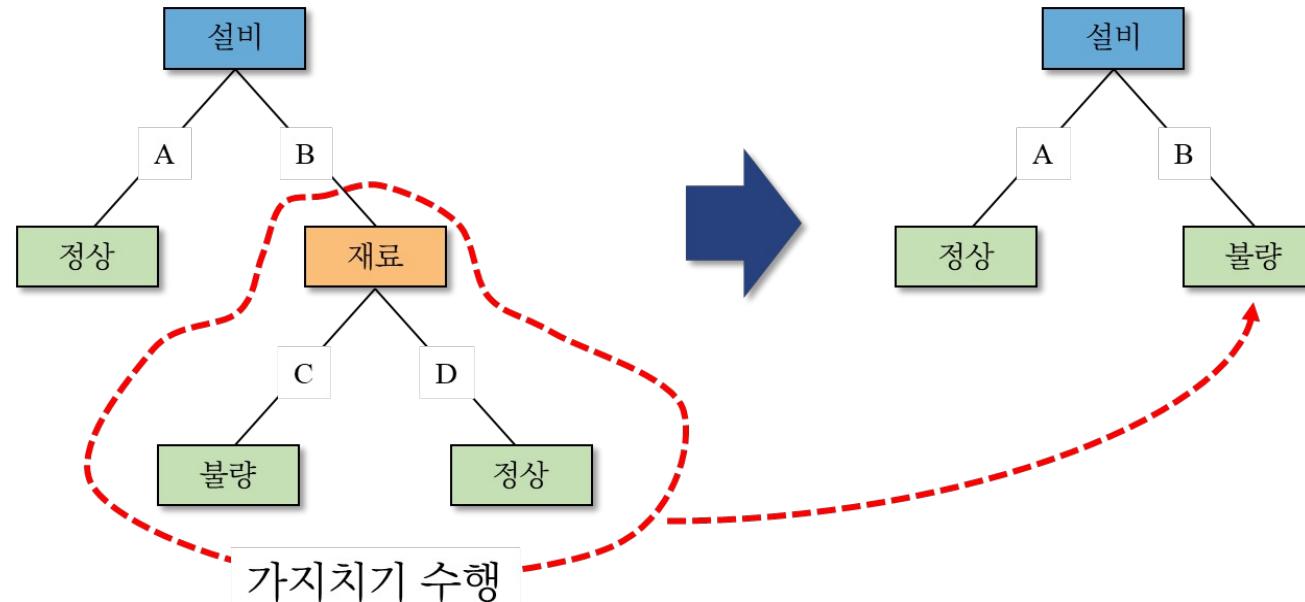


2. CART(Classification and Regression Tree)

가지치기

가지치기 종류

- 사전 가지치기 : 나무가 완성되기 전에 특정 조건을 만족하면 알고리즘 중단(max_depth, min_sample_split 등)
- 사후 가지치기 : 나무가 완성된 후 하단 노드부터 유의미하지 않은 Subtree를 끝 노드로 변환



2. CART(Classification and Regression Tree)

가지치기

사후 가지치기

- 비용-복잡도 가지치기(Cost-Complexity Pruning)을 수행하며 아래 함수를 최소화하는 방향

$$CC(T) = Err(T) + \alpha \cdot |T|$$

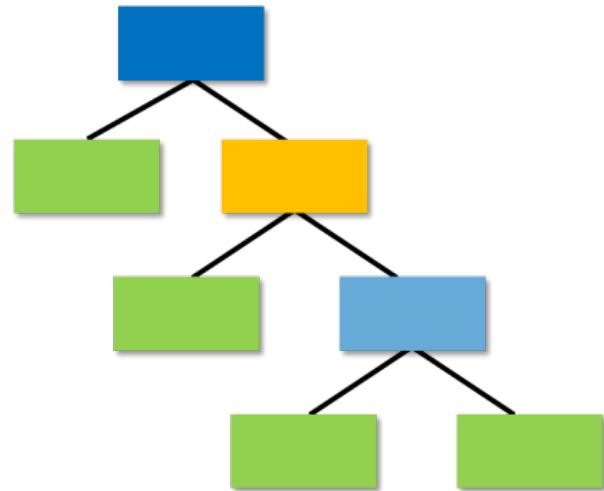
- $CC(T)$: 나무 모델에 대한 비용-복잡도 지표
- $Err(T)$: 검증 데이터에 대한 오분류율
- $|T|$: 끝 노드의 개수 (나무의 복잡도를 나타냄)
- $\alpha (\alpha > 0)$: 함수에서 나무 복잡도의 비중을 나타내는 하이퍼 파라미터로, 값이 커질수록 간결한 모델이 됨

2. CART(Classification and Regression Tree)

가지치기

사후 가지치기

Tree 1



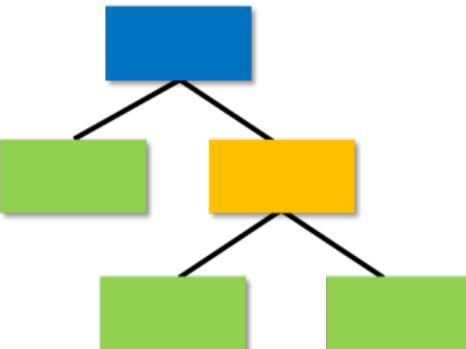
$$Err(T): 0.15$$

$$|T|: 4$$

$$\textcircled{1} \quad \alpha = 0 \quad CC(T) = 0.15 + 0 \times 4 = 0.15$$

$$\textcircled{2} \quad \alpha = 0.1 \quad CC(T) = 0.15 + 0.1 \times 4 = 0.55$$

Tree 2



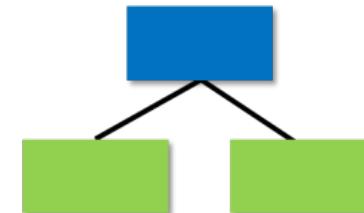
$$Err(T): 0.13$$

$$|T|: 3$$

$$CC(T) = 0.13$$

$$CC(T) = 0.43$$

Tree 3



$$Err(T): 0.18$$

$$|T|: 2$$

$$CC(T) = 0.18$$

$$CC(T) = 0.38$$

2. CART(Classification and Regression Tree)

Regression Tree

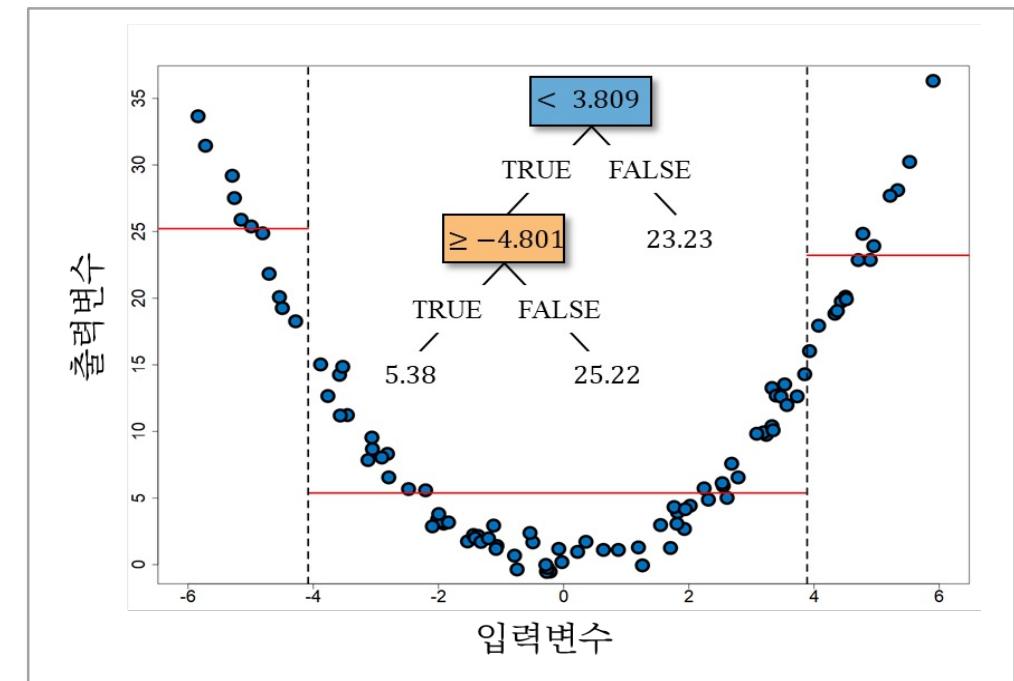
회귀 나무

- 분기 지표를 선택할 때 사용하는 index를 불순도가 아닌 실제값과 예측값의 오차를 사용(=분산)

$$\text{Information gain}(S, A) = Gini(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Gini(S_v)$$



$$\text{Variance difference}(S, A) = \sum_{y_i \in S} (y_i - \bar{y}_S)^2 - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \sum_{y_i \in S_v} (y_i - \bar{y}_{S_v})^2$$



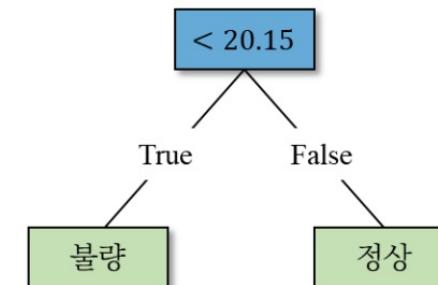
2. CART(Classification and Regression Tree)

Decision/Regression Tree

연속형 입력변수

- 연속형 입력변수의 경우 데이터를 구간(Interval)로 분할하여 범주형 변수로 전환
- CART에서는 구간을 이분하는 단일 경계값 방법 사용(Binary Discretization)
- 우선 입력변수 값을 기반으로 데이터 정렬 후 출력변수 또는 입력변수 값이 바뀌는 지점(평균을 경계값으로 사용)마다의 정보 이득을 계산하여 값이 가장 높은 경계값 선택

온도	18.2	18.3	18.5	18.6	18.7	18.8	20.1	20.2	20.3	20.5	20.7
공정 결과	정상	불량	불량	불량	불량	정상	불량	정상	정상	정상	정상
<hr/>											
	18.25	18.75	19.45	20.15							
	<	≥	<	≥	<	≥	<	≥			
정상	1	5	1	5	2	4	2	4			
불량	0	8	5	3	6	2	8	0			
Gini	0	0.473	0.278	0.469	0.375	0.444	0.32	0			
Info Gain	0.05	0.103	0.085				0.261				



2. CART

회귀(Regression)

Linear Regression

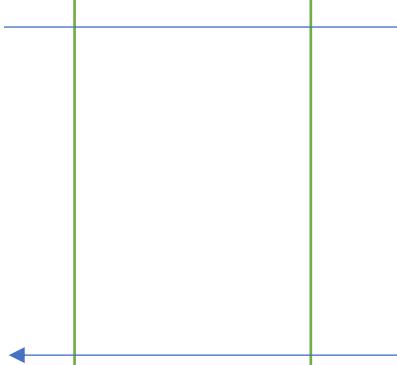
Regression Tree

분류(Classification)

Logistic Regression

SVM

Decision Tree



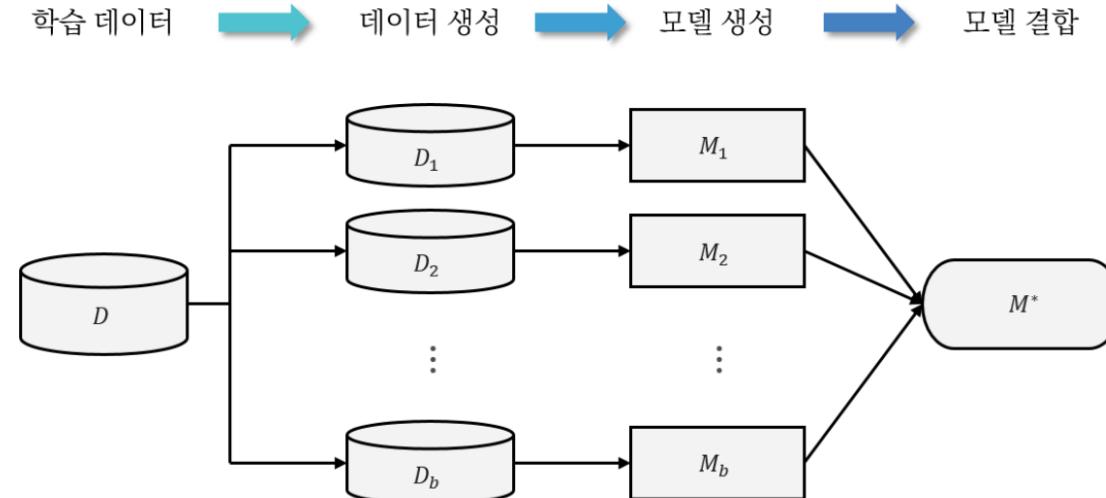
Ensemble

3. Ensemble

Overview

양상분 기법이란

- 다수의 데이터 집합에 대해서 각각의 머신 러닝 모델을 학습하고, 학습된 모델의 예측을 결합하여 최종 예측 수행
- 10명의 전문가로부터 예측 → 10명의 예측 결합 → 전문가 1명보다 더 정확한 예측
- 여러 개의 Weak Learner들이 모여 투표(Voting)을 통해 더 강력한 Stronger Learner를 구성
- Decision tree, SVM, Deep learning 등 모든 종류의 학습 모델이 사용 될 수 있음

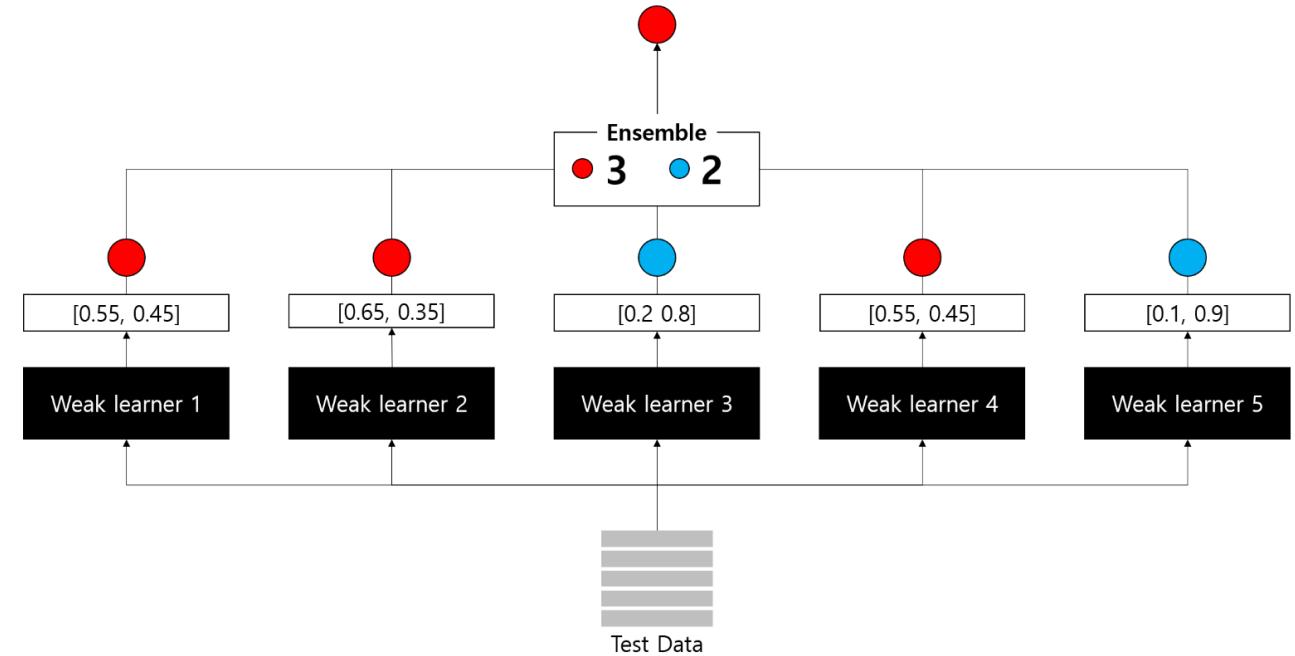


3. Ensemble

Voting

Hard Voting

- 각 Weak Learner들의 예측 결과값을 바탕으로 다수결 투표 하는 방식
- 동일한 데이터에 대해 각 분류기는 클래스별 예측 확률을 제시 → 최종 예측값 계산(다수결 투표)
- 5개 분류기 중 빨간 공으로 예측한 분류기 3개
→ 빨간 공(최종 예측값)



3. Ensemble

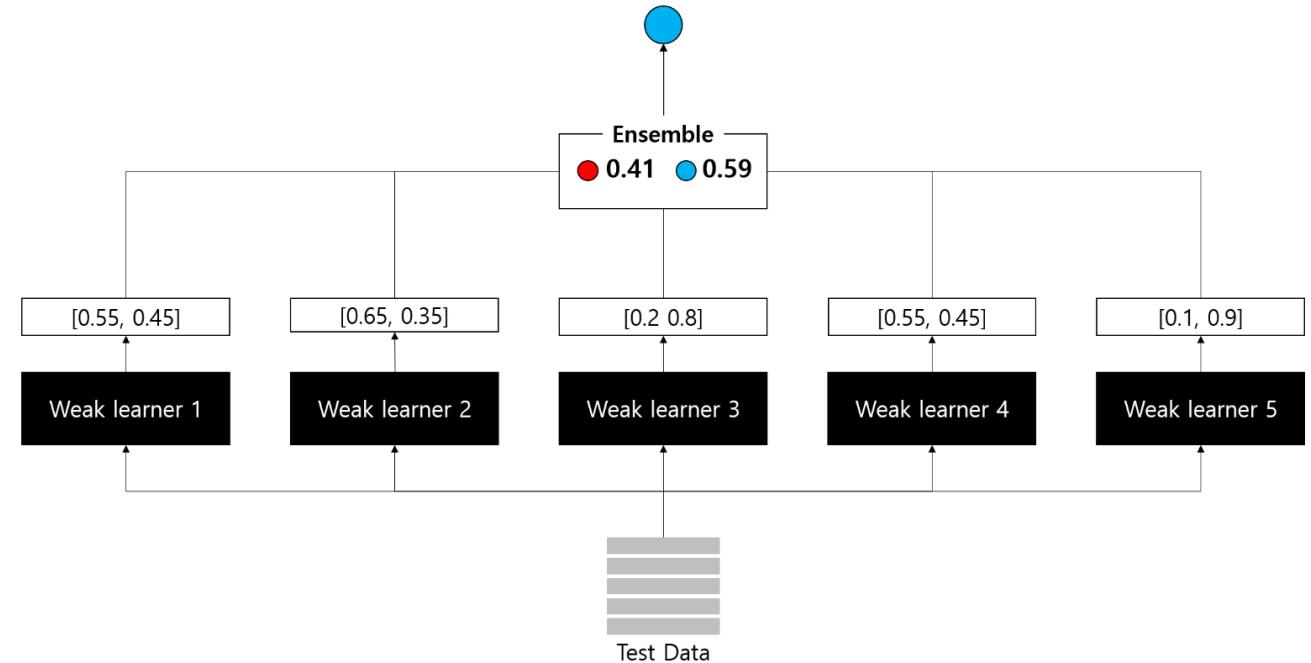
Voting

Soft Voting - Average

- Weak learner 개별의 예측값은 중요하지 않음
- 예측 확률값을 단순 평균내어 확률이 더 높은 클래

스를 최종 예측값으로 결정

- 파란 공으로 예측한 두개 분류기(3, 5)가 높은 확률로 파란 공으로 예측 → 파란 공(최종 예측값)

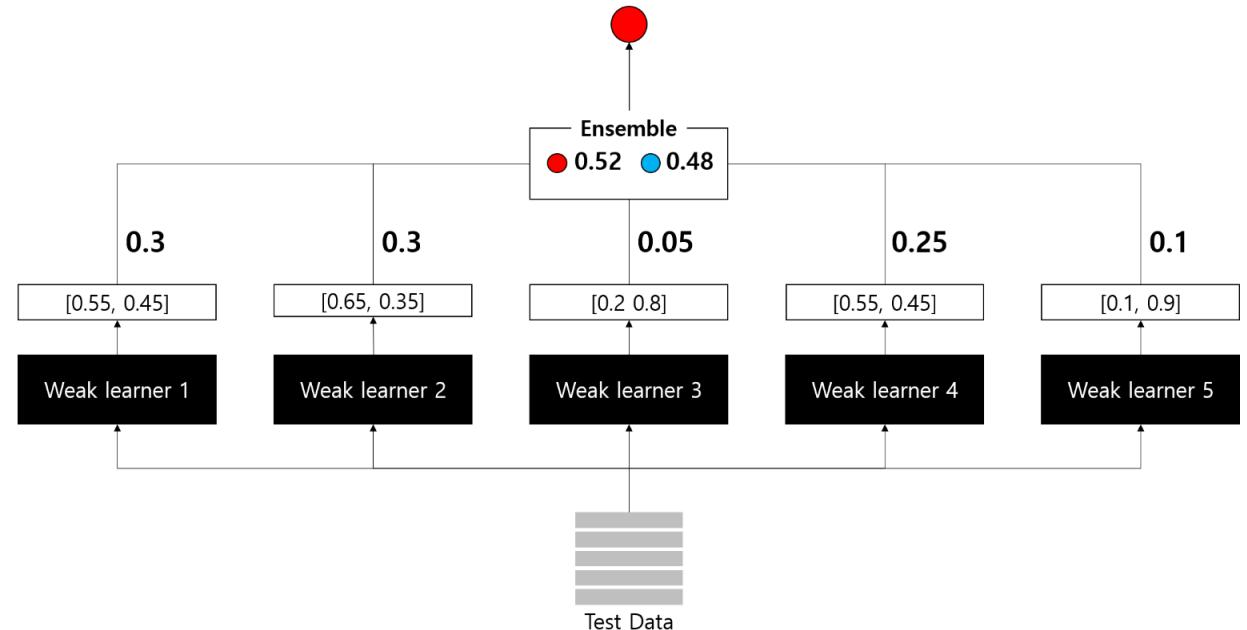


3. Ensemble

Voting

Soft Voting – Weighted Sum

- Weak learner들에 대한 신뢰도가 다를 경우, 가중치를 부여하여 가중치 합 사용
- 각 분류기의 예측확률값에 상이한 가중치 부여
→ 빨간 공(최종 예측값)



3. Ensemble

Algorithm

Bagging

- Bootstrap Aggregating의 약자. **부트스트랩(Bootstrap)**을 이용
- **부트스트랩(복원추출)** : 주어진 데이터셋에서 Random Sampling 하여 새로운 데이터셋을 만들어냄
- **부트스트랩**을 통해 만들어진 여러 데이터셋을 바탕으로 Weak learner를 훈련시킨 뒤, Voting

Boosting

- 반복적으로 모델을 업데이트 → 이전 iteration의 결과에 따라 데이터셋 샘플에 대한 가중치 부여
- 반복할 때마다 각 샘플의 중요도에 따라 다른 분류기가 만들어짐
- 최종적으로 모든 iteration에서 생성된 모델의 결과를 voting

Stacking

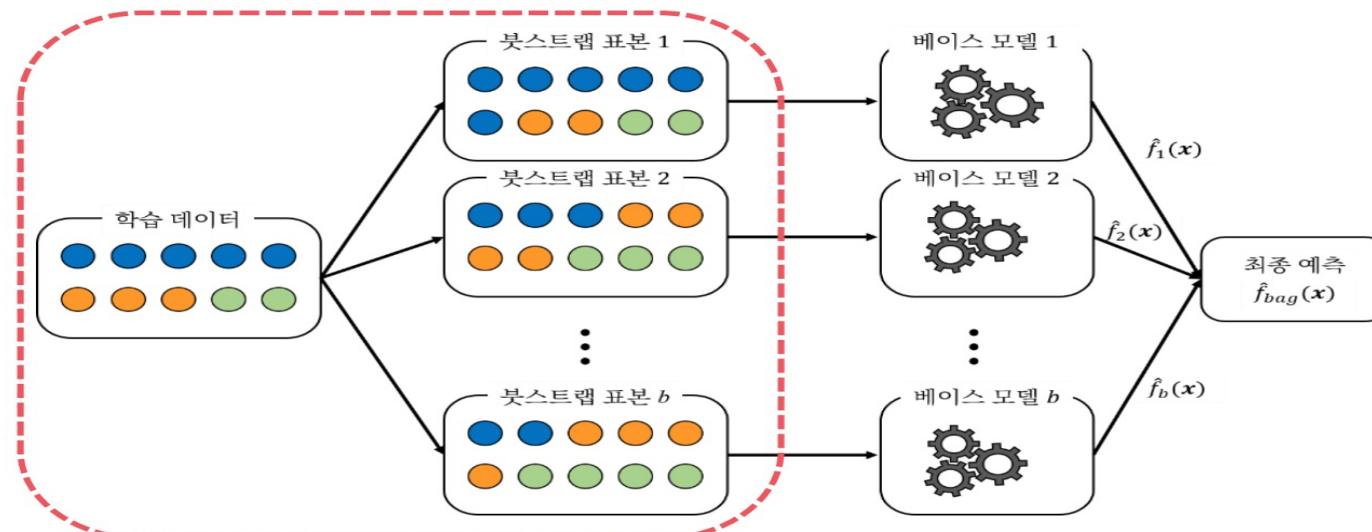
- Weak learner들의 예측 결과를 바탕으로 Meta learner로 학습시켜 최종 예측값 결정
- Meta learner 또한 학습이 필요하며 사용되는 데이터는 training data에 대한 각 Weak learner들의 예측 확률값의 모음

4. Bagging

Bootstrap

부트스트랩 샘플링

- 학습 데이터로부터 원하는 크기의 샘플을 **복원추출하는 방법**
- 표본의 크기가 학습 데이터의 수와 같다면 각 표본은 평균적으로 학습 데이터의 약 63.2 %만 사용
- 따라서 **부트스트랩** 표본 내에서도 동일 데이터가 중복되어 관측 및 표본 간에도 중복 데이터가 존재

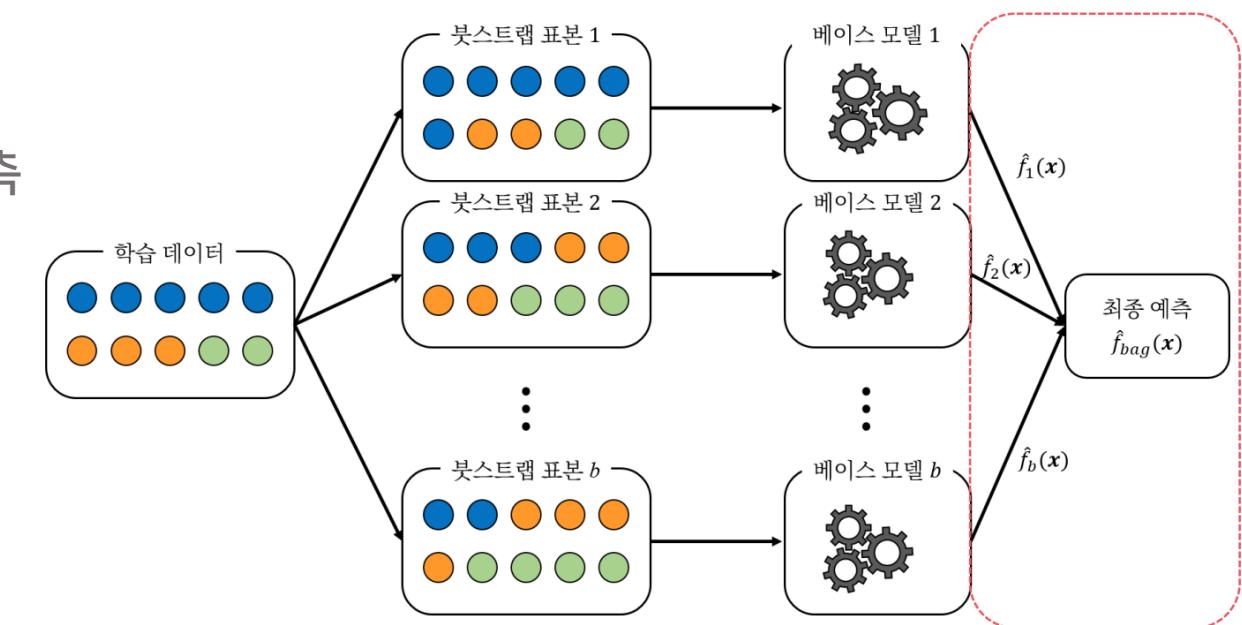


4. Bagging

Bootstrap

모델 결합

- 출력변수가 범주형인 경우(분류 문제)
 - 각 모델의 예측값들을 다수결 투표하여 최종 예측
- 출력변수가 연속형인 경우(회귀 문제)
 - 각 모델의 예측값들을 평균 내어 최종 예측

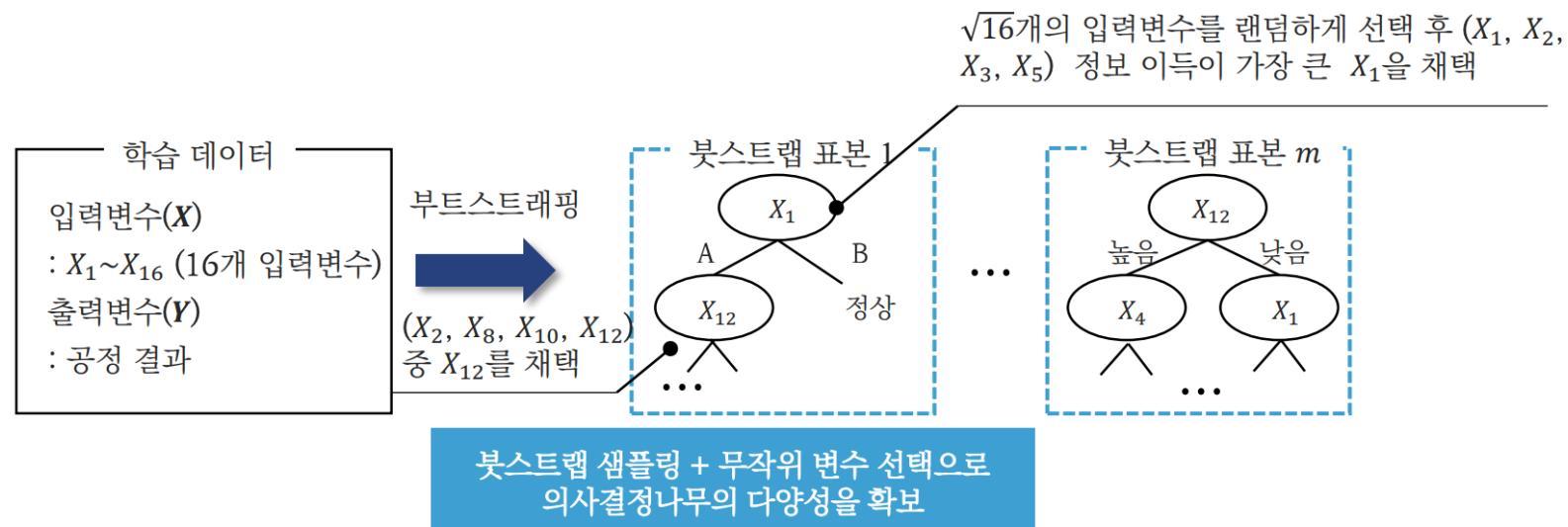


4. Bagging

Random Forest

Random Forest란

- 부스트랩 샘플링을 기반으로 여러 개의 의사결정나무를 생성하여 다수결 또는 평균에 따라 출력변수 예측
- 무작위 변수 선택 기법 사용 : 각 노드마다 입력변수의 일부를 랜덤으로 선택하고, 선택된 변수 중 불순도 감소량이 가장 큰 입력변수를 선택
- Random Forests는 변수의 중요도를 산출하여 제공

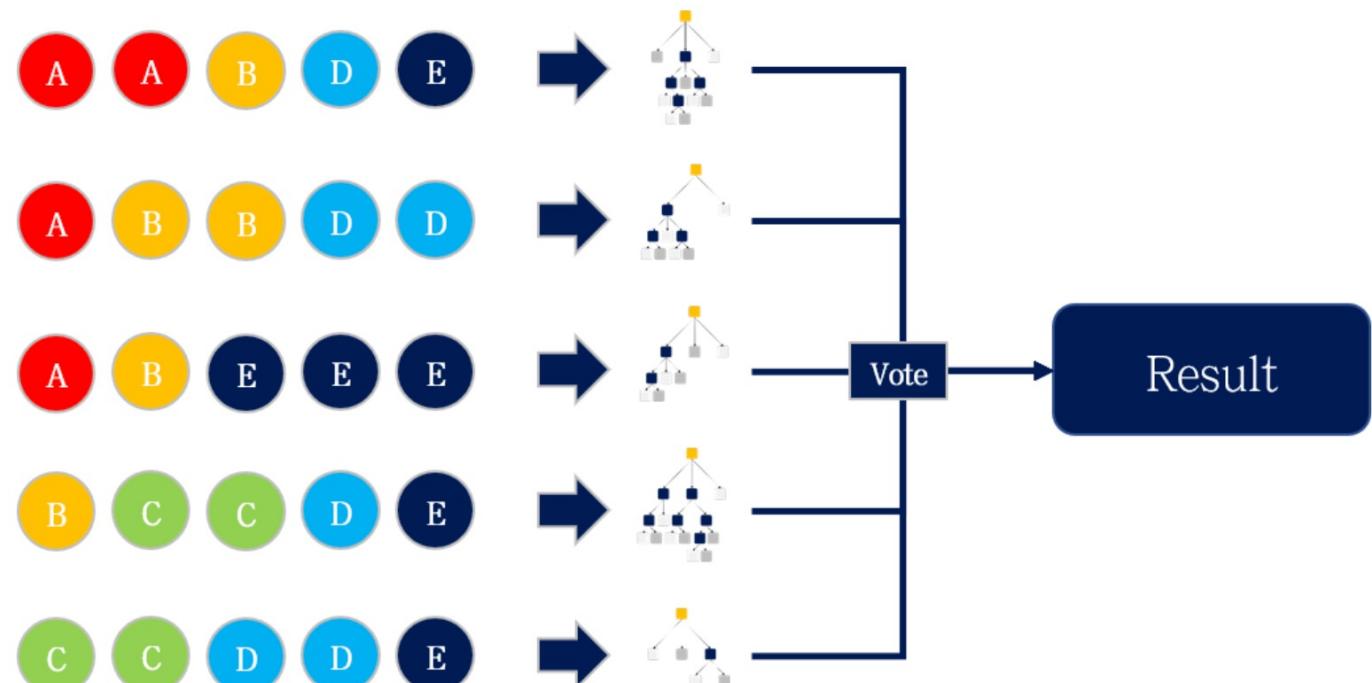


4. Bagging

Random Forest

Random Forest란

1. N개의 랜덤한 부트스트랩 샘플 추출(중복 허용)
2. 각 노드에서 중복을 허용하지 않고 랜덤하게 d개의 특성을 선택
3. IG와 같은 목적 함수를 기준으로 최선의 분할을 만드는 특성을 사용해 노드를 분할
4. 1~3 단계를 k번 반복
5. 다수결 투표 또는 평균에 따라 클래스 레이블 할당

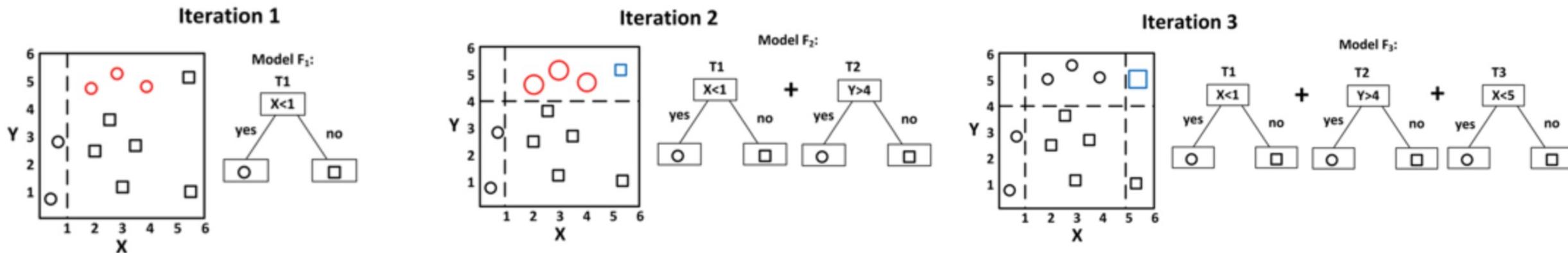


5. Boosting

Boosting

Boosting이란

- 베이스 모델을 각 학습 라운드마다 순차적으로 학습시키고, 이를 결합하여 예측
- 순차적 학습 과정에서 각 라운드의 모델들은 이전 모델들이 잘못 예측한 부분에 대해 중점적으로 학습
- Ada Boost 계열과 Gradient Boosting Machine(GBM) 계열이 많이 사용
- Xgboost, LightGBM, Catboost 알고리즘 등이 좋은 성능 보임



5. Boosting

GBM

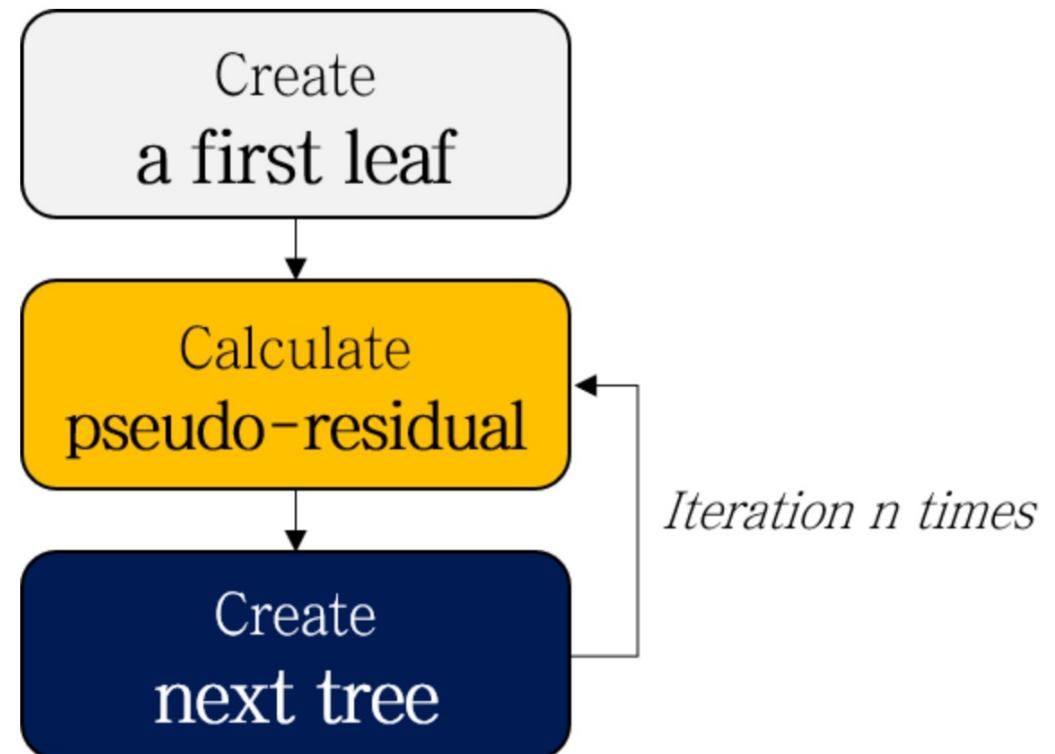
GBM 원리 : 잔차를 지속적으로 학습

- GBM은 m 라운드의 베이스 모델 h_m 이 ($m - 1$)라운드 까지의 베이스 모델을 결합한 $H_m = h_1 + \dots + h_{m-1}$ 의 잔차를 학습하는 알고리즘
- 즉 이전 결합 모델의 예측 오류는 아직까지 학습 못한 특징이라고 가정함

$$\begin{aligned}
 y &= h_0(x) + \text{error 1} \\
 \text{error 1} &= h_1(x) + \text{error 2} \\
 \text{error 2} &= h_2(x) + \text{error 3} \\
 &\vdots \\
 \text{error } (M) &= h_M(x) + \text{error } (M + 1)
 \end{aligned}$$

최종 결합 모델: $H_M = h_0 + h_1 + \dots + \underline{h_M}$

M 은 사용자가 정한 최대 베이스 모델 수



5. Boosting

회귀 문제에서의 학습

- m 라운드의 베이스 모델 h_m 은 m 라운드까지의 예측 오류($y - H_m(x)$)를 최소화하는 방향으로 학습

n 개 데이터에 대한
평균 오류자승

예측한 출력변수

Minimize h_m 학습

데이터 개수

실제 출력변수

$$MSE = \frac{1}{n} \sum_{i=1}^n [y_i - H_{m-1}(x_i)]^2$$

$$MSE = \frac{1}{n} \sum_{i=1}^n [y_i - H_m(x_i)]^2$$

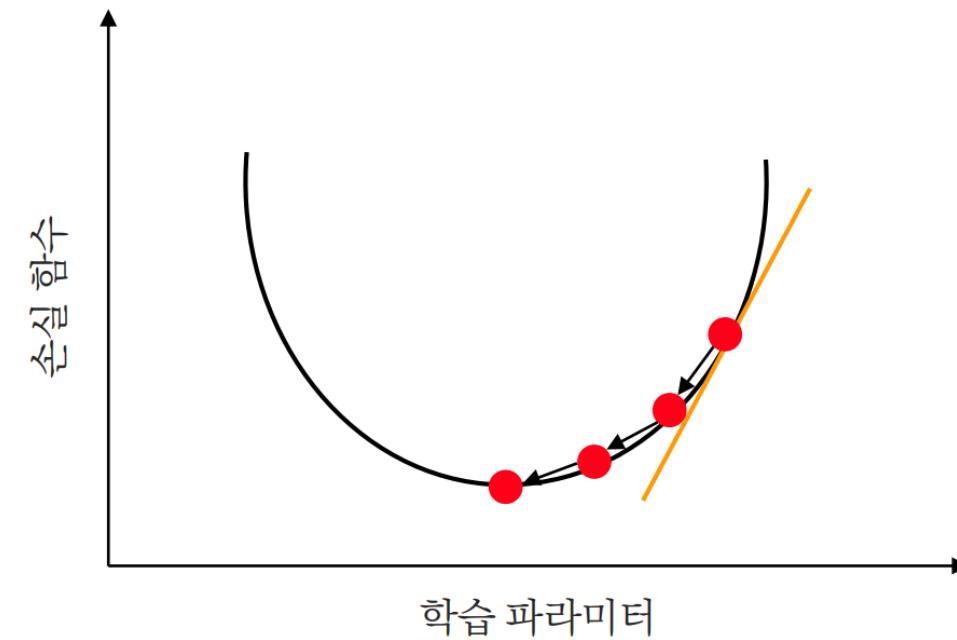
 $H_{m+1}(x) = H_m(x) + h_{m+1} = h_1 + \dots + h_m +$

5. Boosting

GBM

회귀 문제에서의 학습

- 손실 함수를 최소화하기 위해서는 일반적으로 함수의 음의 기울기가 0이 되는 방향으로 베이스 모델의 학습 파라미터를 조절함 (Gradient Descent 방법)
- 음의 기울기 = 잔차
- 잔차를 순차적으로 최소화하는 과정이 손실 함수를 최소화하는 방법 → Boosting 알고리즘을 적용함



5. Boosting

GBM

GBM 회귀 모델 예시

- 각각의 베이스 모델이 학습하는 입력변수와 출력변수는 아래와 같음
- h_0 는 초기값으로 일반적으로 회귀 문제에서는 출력변수의 평균을 사용
- 베이스 모델로는 회귀 트리 알고리즘인 CART를 주로 사용함

X_1	X_p	출력변수
$x_{1,1}$	$x_{1,p}$	y_1
$x_{2,1}$	$x_{2,p}$	y_2
\vdots	\vdots	\vdots
$x_{n-1,1}$	$x_{n-1,p}$	y_{n-1}
$x_{n,1}$	$x_{n,p}$	y_n

h_0 학습

X_1	X_p	출력변수
$x_{1,1}$	$x_{1,p}$	$y_1 - h_0(x_1)$
$x_{2,1}$	$x_{2,p}$	$y_2 - h_0(x_2)$
\vdots	\vdots	\vdots
$x_{n-1,1}$	$x_{n-1,p}$	$y_{n-1} - h_0(x_{n-1})$
$x_{n,1}$	$x_{n,p}$	$y_n - h_0(x_n)$

h_1 학습

X_1	X_p	출력변수
$x_{1,1}$	$x_{1,p}$	$y_1 - h_0(x_1) - v h_1(x_1)$
$x_{2,1}$	$x_{2,p}$	$y_2 - h_0(x_2) - v h_1(x_2)$
\vdots	\vdots	\vdots
$x_{n-1,1}$	$x_{n-1,p}$	$y_{n-1} - h_0(x_{n-1}) - v h_1(x_{n-1})$
$x_{n,1}$	$x_{n,p}$	$y_n - h_0(x_n) - v h_1(x_n)$

h_2 학습

5. Boosting

GBM

GBM 회귀 모델 예시

- 특정 제품의 두께를 예측하는 예제 ($\nu=0.8$, 학습률)

설비	온도	재료	제품 두께	Error1
A	18	C	15	4
B	20	D	9	-2
A	22	C	14	3
B	19	D	8	-3
B	17	C	9	-2



h_0 는 제품 두께의 평균을 예측하는 회귀 트리 모델

$$\frac{15 + 9 + 14 + 8 + 9}{5} = 11$$

Error1 계산

$$1\text{번 데이터: } 15 - 11 = 4$$

$$2\text{번 데이터: } 9 - 11 = -2$$

$$3\text{번 데이터: } 14 - 11 = 3$$

$$4\text{번 데이터: } 8 - 11 = -3$$

$$5\text{번 데이터: } 9 - 11 = -2$$

5. Boosting

GBM

GBM 회귀 모델 예시

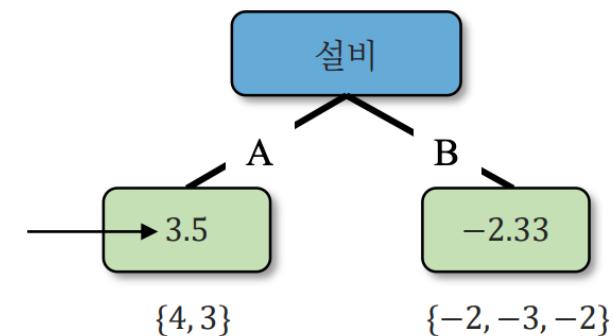
설비	온도	재료	제품 두께	Error1
A	18	C	15	4
B	20	D	9	-2
A	22	C	14	3
B	19	D	8	-3
B	17	C	9	-2

h_0 는 제품 두께의 평균을 예측하는 회귀 트리 모델

$$\frac{15 + 9 + 14 + 8 + 9}{5} = 11$$

h_1 은 Error1을 출력변수로 두고 회귀 트리 모델 생성

$$\text{끝노드 출력값} = \frac{j\text{노드 내 잔차합}}{j\text{ 노드 내 데이터 개수}} \\ = \frac{4+3}{2} = 3.5$$



5. Boosting

GBM

GBM 회귀 모델 예시

설비	온도	재료	제품 두께	Error1	예측값
A	18	C	15	4	13.8
B	20	D	9	-2	9.14
A	22	C	14	3	13.8
B	19	D	8	-3	9.14
B	17	C	9	-2	9.14

h_0 는 제품 두께의 평균을 예측하는 회귀 트리 모델

$$\frac{15 + 9 + 14 + 8 + 9}{5} = 11$$

h_1 은 Error1을 출력변수로 두고 회귀 트리 모델 생성

$H_1(h_0 + \nu h_1)$ 예측

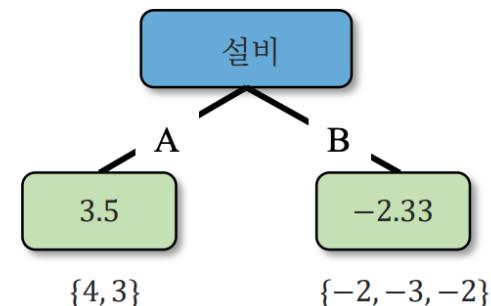
$$1\text{번 데이터: } 11 + 0.8 \times 3.5 = 13.8$$

$$2\text{번 데이터: } 11 + 0.8 \times -2.33 = 9.14$$

$$3\text{번 데이터: } 11 + 0.8 \times 3.5 = 13.8$$

$$4\text{번 데이터: } 11 + 0.8 \times -2.33 = 9.14$$

$$5\text{번 데이터: } 11 + 0.8 \times -2.33 = 9.14$$



5. Boosting

XGBoost

XGBoost 소개

- XGBoost는 GBM과 동일하게 이전 라운드에서의 예측 오류를 다음 라운드의 모델 학습에 반영시킴
 - 그러나 학습을 위한 목적식에 규제항 $\Omega(h)$ 가 추가됨

Minimize m 라운드까지의 예측 오류($y - H_m(x)$) + 트리 규제항 $\Omega(h_m)$

h_m 학습

- 규제항은 트리의 복잡성에 패널티를 부여하는 항임
 - ① : 트리가 커짐에 따른 패널티 / ② : 특정 끝노드에서의 출력값이 커지는 것에 대한 패널티 → 과적합 문제 방지

T : 트리 끝노드 수, γ : 트리 복잡도에 대한 페널티

$$\Omega(h_m) = \frac{1}{\gamma T} + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

베이스 모델
 출력값 페널티
 트리의 j 번째 끝노드에서의 출력값

①
 ②

5. Boosting

XGBoost

XGBoost 소개

- 앞선 목적함수 전개 시 각 노드의 출력값과 Quality Score(QS)를 계산할 수 있음
- QS : Similarity Score, 해당 노드에 속하는 데이터들이 얼마나 유사한지를 계산하는 지표

$$\text{Similarity score} = \frac{\text{sum of residuals}^2}{\text{the number of residuals} + \lambda}$$

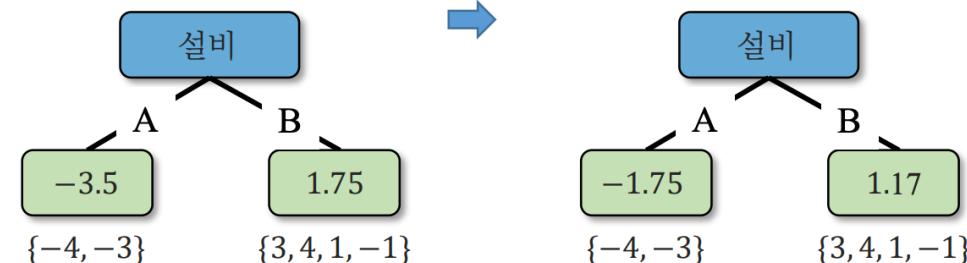
- λ 는 출력값을 수축해주는 하이퍼 파라미터로 λ 가 커질수록 출력값 w 의 절대값이 작아짐

$$(\lambda=0) \text{ 출력값}_A = \frac{-4-3}{2+\lambda} = \frac{-7}{2} = -3.5, \quad (\lambda=2) \text{ 출력값}_A = \frac{-4-3}{2+2} = \frac{-7}{4} = -1.75,$$

$$\text{출력값}_B = \frac{3+4+1-1}{4+\lambda} = 1.75 \quad \text{출력값 축소}$$

$$\text{출력값}_B = \frac{3+4+1-1}{4+2} = 1.17$$

$$\text{출력값 } w_j = \frac{j\text{노드 내 잔차합}}{j\text{ 노드 내 데이터 개수} + \lambda}$$



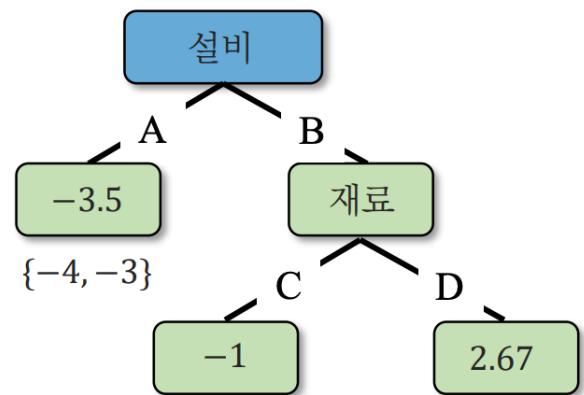
5. Boosting

XGBoost

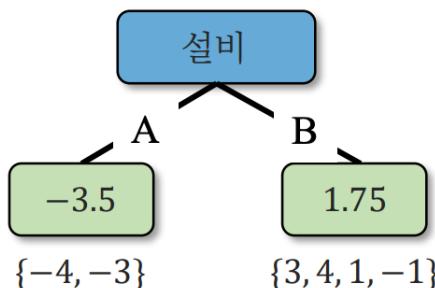
XGBoost 소개

- γ 는 가지치기의 정도를 결정해주는 하이퍼 파라미터로 γ 가 커질수록 더욱 많은 가지치기가 수행됨

($\gamma=5$)



($\gamma=10$)



($\gamma=20$)



모델이 생성되지 않음

5. Boosting

XGBoost

XGBoost 예시

- 특정 제품의 두께를 예측하는 예제($\nu=0.8$, $\gamma = 10$, $\lambda = 2$ 이고 베이스 모델로는 회귀 트리를 사용)

설비	재료	제품 두께	error1
A	C	8	-4
A	D	9	-3
B	D	15	3
B	D	16	4
B	D	13	1
B	C	11	-1

h_0 는 제품 두께의 평균을 예측하는 회귀 트리 모델

$$\frac{8 + 9 + 15 + 16 + 13 + 11}{6} = 12$$

error1 계산

1번 데이터: $8 - 12 = -4$

2번 데이터: $9 - 12 = -3$

3번 데이터: $15 - 12 = 3$

4번 데이터: $16 - 12 = 4$

5번 데이터: $13 - 12 = 1$

6번 데이터: $11 - 12 = -1$

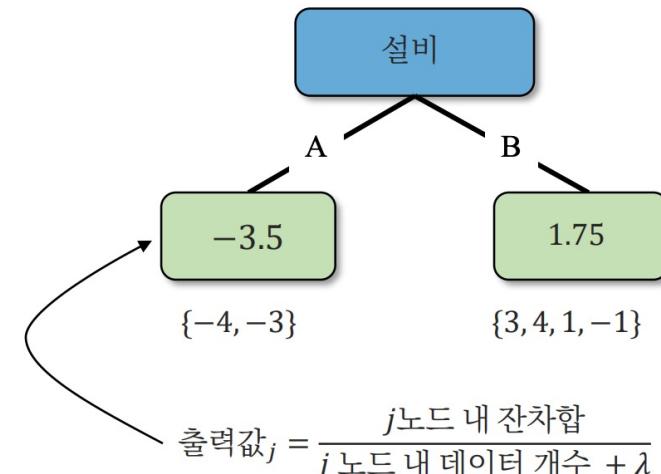
5. Boosting

XGBoost

XGBoost 예시

- $\lambda = 0$ 인 경우, 각 노드의 출력값은 GBM과 동일하지만 λ 가 커질수록 출력값이 수축됨

설비	재료	제작 수량	error1
A	C	8	-4
A	D	9	-3
B	D	15	3
B	D	16	4
B	D	13	1
B	C	11	-1



$$(\lambda=0) \quad \text{출력값}_A = \frac{-4-3}{2+\lambda} = \frac{-7}{2} = -3.5, \quad \text{출력값}_B = \frac{3+4+1-1}{4+\lambda} = 1.75$$

$$(\lambda=2) \quad \text{출력값}_A = \frac{-4-3}{2+2} = \frac{-7}{4} = -1.75, \quad \text{출력값}_B = \frac{3+4+1-1}{4+2} = 1.17$$

5. Boosting

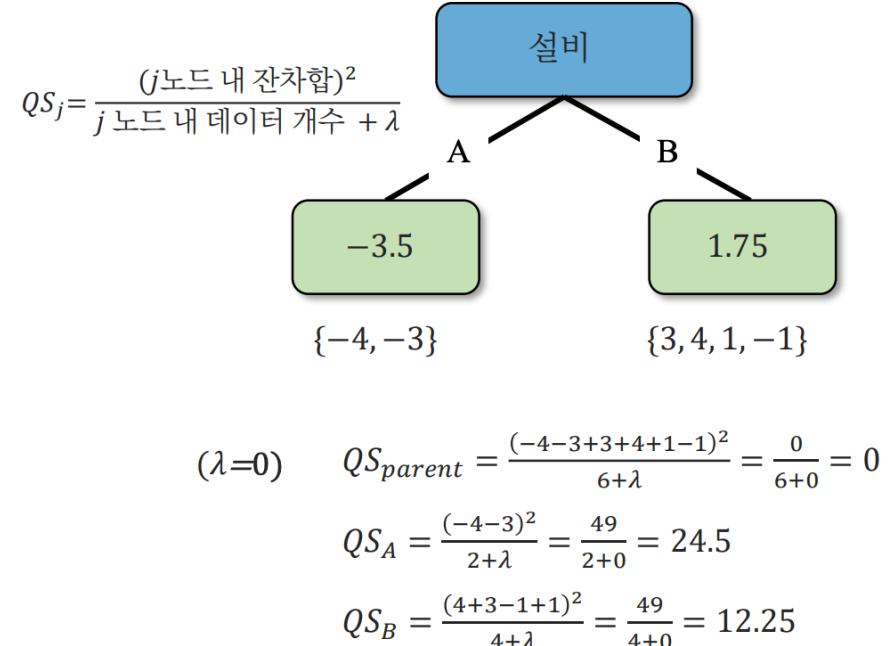
XGBoost

XGBoost 예시

- QS역시 λ 가 커질수록 값이 수축되며 불순도 감소량(정보이득)을 나타내는 Gain 계산 가능

설비	재료	제품 두께	error1
A	C	8	-4
A	D	9	-3
B	D	15	3
B	D	16	4
B	D	13	1
B	C	11	-1

$$\begin{aligned} Gain &= QS_{Left} + QS_{Right} - QS_{parent} \\ &= 24.5 + 12.25 - 0 = 36.75 \end{aligned}$$



5. Boosting

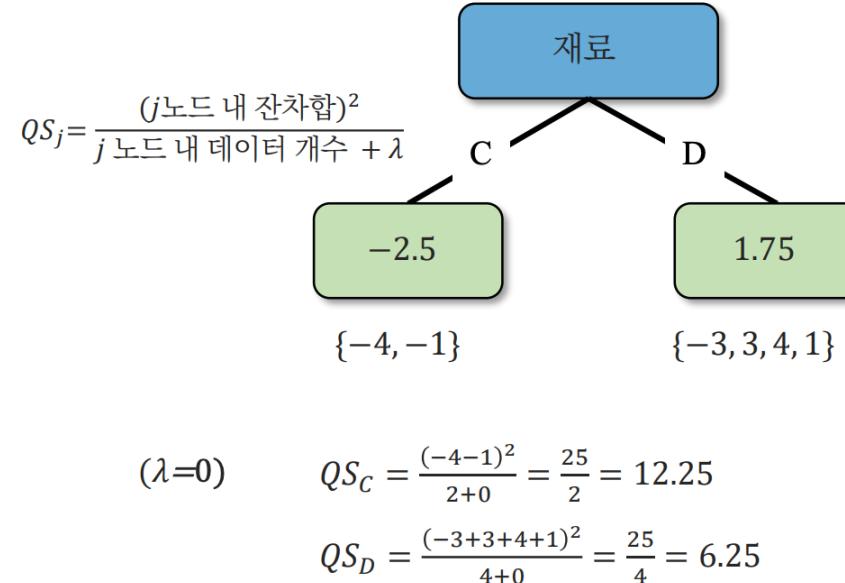
XGBoost

XGBoost 예시

- 설비의 Gain이 더 높으므로, 첫 번째 노드에서의 분할 변수는 설비가 됨

설비	재료	제품 두께	error1
A	C	8	-4
A	D	9	-3
B	D	15	3
B	D	16	4
B	D	13	1
B	C	11	-1

$$\begin{aligned}Gain &= QS_{Left} + QS_{Right} - QS_{parent} \\&= 12.25 + 6.25 - 0 = 18.5\end{aligned}$$

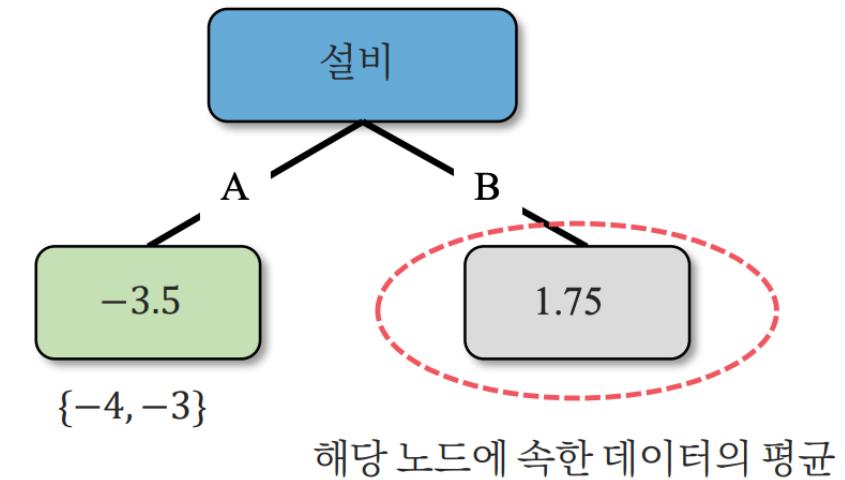
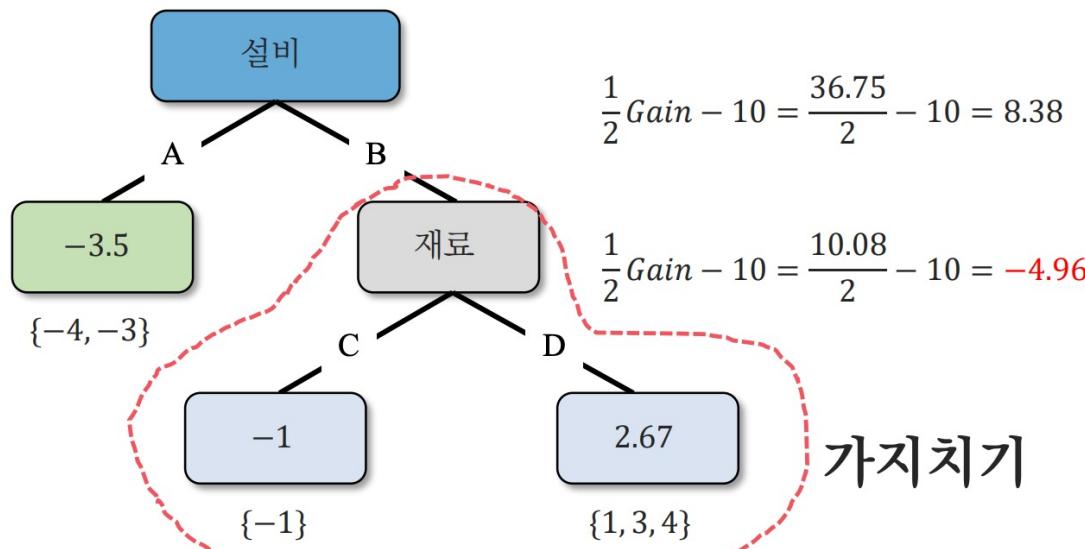


5. Boosting

XGBoost

XGBoost 예시

- 사용자가 지정한 깊이까지 트리를 생성한 후에 가지치기를 수행하며, $\frac{1}{2} Gain - \gamma$ 가 음수인 경우 가지치기



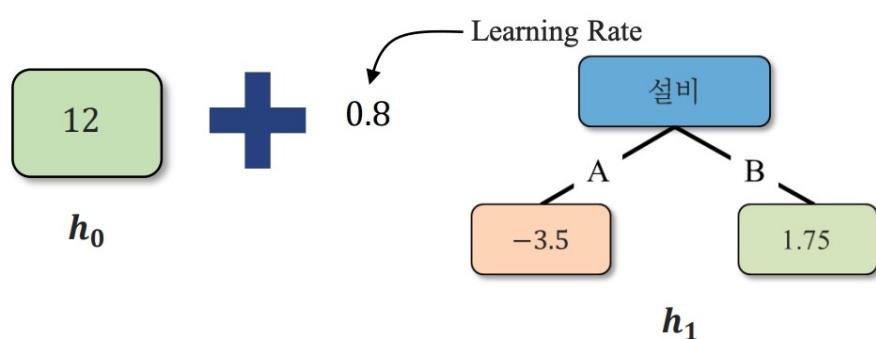
5. Boosting

XGBoost

XGBoost 예시

설비	재료	제품 두께	error1	예측값
A	C	8	-4	9.2
A	D	9	-3	9.2
B	D	15	3	13.4
B	D	16	4	13.4
B	D	13	1	13.4
B	C	11	-1	13.4

최종 모델: $H_1(h_0 + vh_1)$



$H_1(h_0 + vh_1)$ 예측

1번 데이터: $12 + 0.8 \times -3.5 = 9.2$

2번 데이터: $12 + 0.8 \times -3.5 = 9.2$

3번 데이터: $12 + 0.8 \times 1.75 = 13.4$

4번 데이터: $12 + 0.8 \times 1.75 = 13.4$

5번 데이터: $12 + 0.8 \times 1.75 = 13.4$

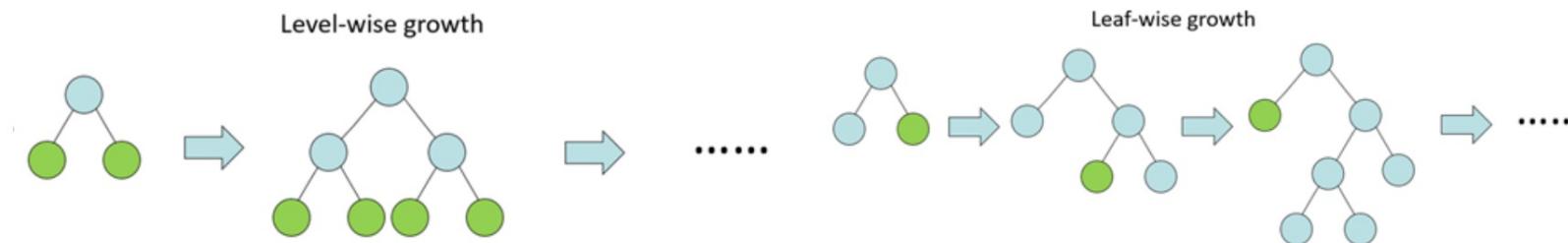
6번 데이터: $12 + 0.8 \times 1.75 = 13.4$

5. Boosting

LightGBM & CatBoost

Light Gradient Boosting Machine(LightGBM)

- XGBoost 모델은 성능은 좋지만 학습시간이 오래 걸림
- LightGBM은 Tree 구조가 수평적으로 확장하는 다른 알고리즘과 달리 수직적으로 확장(level-wise)
- 속도가 빠르고 적은 메모리를 차지



Categorical Boosting(CatBoost)

- 기존의 GBM 모델들은 범주형 입력변수를 처리하는데 문제점 → Cardinality가 너무 클 경우 변수의 수가 급격히 증가하여 계산시간 증가와 과적합 문제 발생
- Ordered Boosting(일부만 잔차계산 한 뒤 뒤의 데이터는 예측한 값 사용), Random Permutation, Order Target Encoding, Categorical Feature Combinations…

6. Summary

Decision Tree

- 의사결정나무는 입력변수 공간을 수직 또는 수평으로 나누는 작업을 반복하며 분류 규칙을 만들어내는 알고리즘
- 좋은 질문을 통한 분류를 위해선 '불순도'에 대한 정의가 필요하다 → Entropy, Gini Impurity
- 결정 트리가 깊어질수록 결정 경계가 복잡해지고 과대적합되기 쉬움 → 양상을 기법

Ensemble

- 집단 지성의 원리
- Random Forest : 부트스트랩 샘플링 기반의 양상을 알고리즘
- GBM : Boosting 기반의 양상을 알고리즘. Gradient Descent + Boosting. 회귀와 분류 둘 다 사용가능. 과적합 문제 있음
- XGBoost : 자체 과적합 규제 기능으로 강한 내구성. 작은 데이터에 대하여 과적합 가능성이 있음

6. Summary

회귀(Regression)

Linear Regression

Regression Tree

Ensemble

Bagging

- RandomForest

Boosting

- Ada-Boost
- GBM
- XGBoost
- LightGBM
- CATBoost

평균

분류(Classification)

Logistic Regression

SVM

Decision Tree

Ensemble

Bagging

- RandomForest

Boosting

- Ada-Boost
- GBM
- XGBoost
- LightGBM
- CATBoost

다수결

- <https://tyami.github.io/>
- <https://dailyheumsi.tistory.com/136>
- <https://nicola-ml.tistory.com/51>
- 김창욱 교수님 머신러닝과 산업 응용 4강, 6강
- 머신러닝 교과서 with 파이썬, 사이킷런, 텐서플로

DATA SCIENCE LAB

발표자 장준혁 010-9113-5314
E-mail: jj980512@gmail.com