2020 D&A

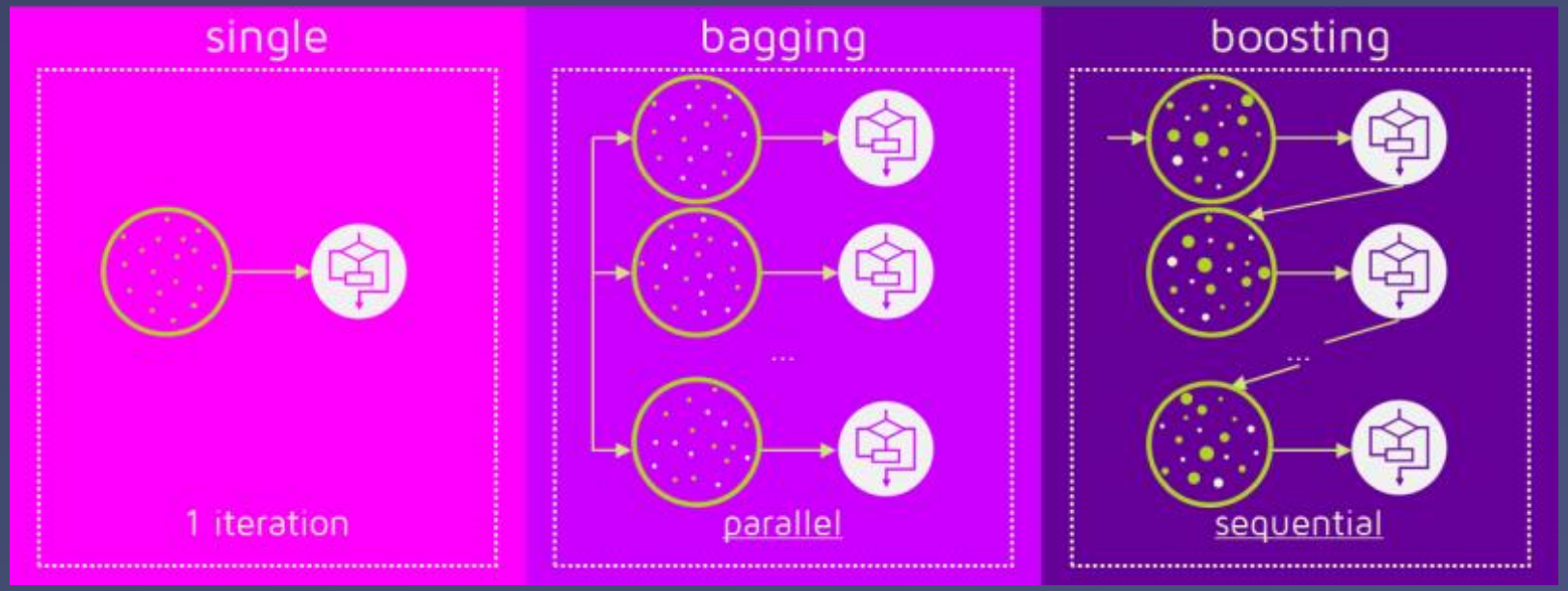
MachineLearning SESSION

Ensemble(Boosting)

D&A

- Boosting

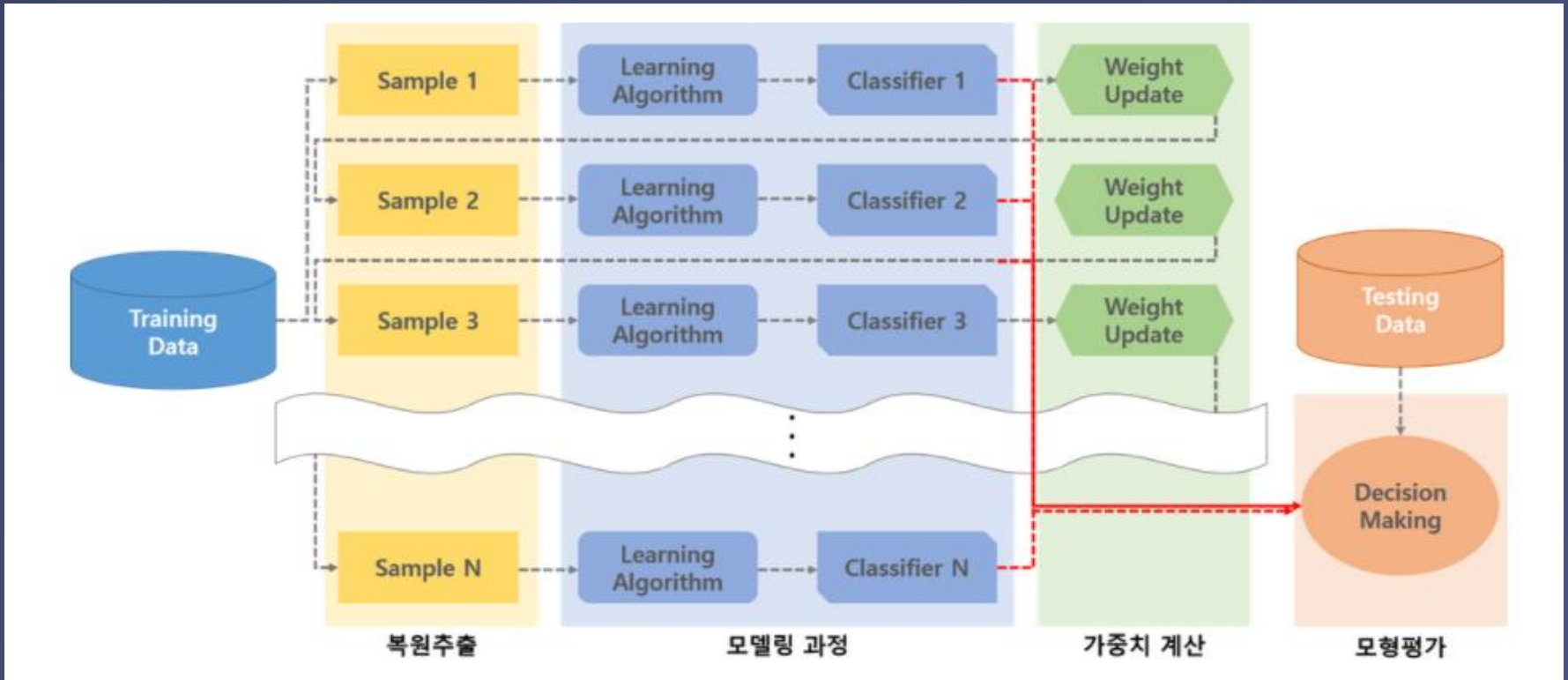
1. Bagging : 일반적인 모형을 만드는데 초점, 분산을 줄여 과대적합(Overfitting)을 막아줌
2. Boosting : 맞추기 어려운 문제를 맞추는 데 초점, 틀린 문제에 가중치 부과



- Boosting

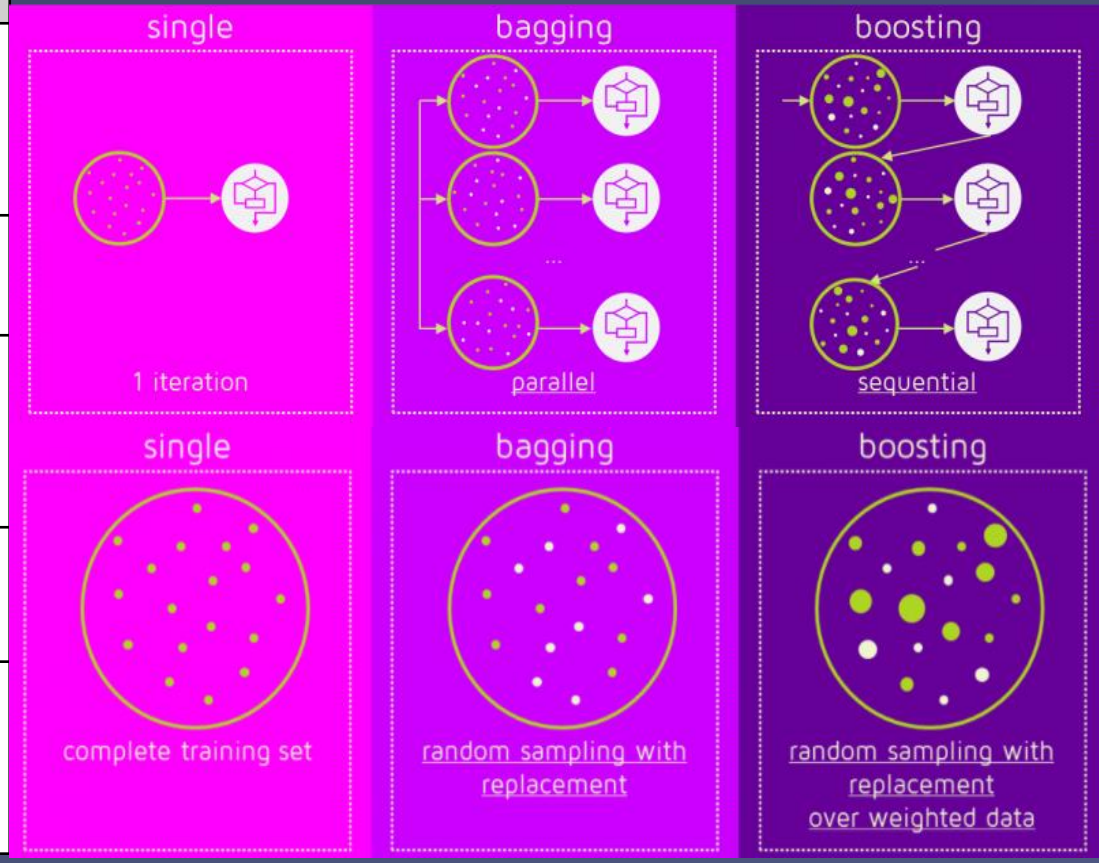
Boosting 이란 ?

- 개별 모델들의 앙상블 기법 중 하나
- 약한 분류기 (weak classifier) 를 결합하여 강한 분류기를 만드는 방법
- 재표본 과정에서 분류에 오류가 있는 경우, 오류에 가중치를 주어 표본 추출



- Bagging vs Boosting

비교	Bagging	Boosting
특징	**병렬 앙상블 모델** (각 모델은 서로 독립적)	**연속 앙상블** (이전 모델의 오류를 고려)
목적	Variance 감소	Bias 감소
적합한 상황	**복잡한 모델** (High Variance, Low bias)	Low variance, High bias 모델
대표 알고리즘	Random Forest	AdaBoost, Gradient Boosting
표본추출	Random Sampling	Random Sampling with weight on Error

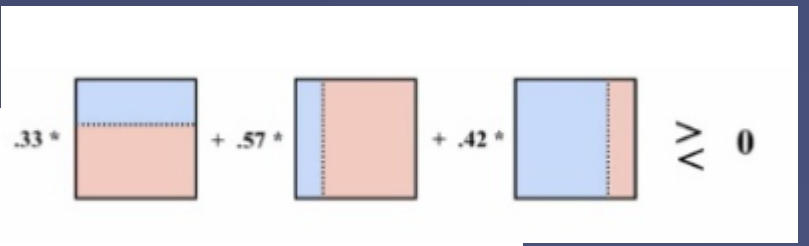


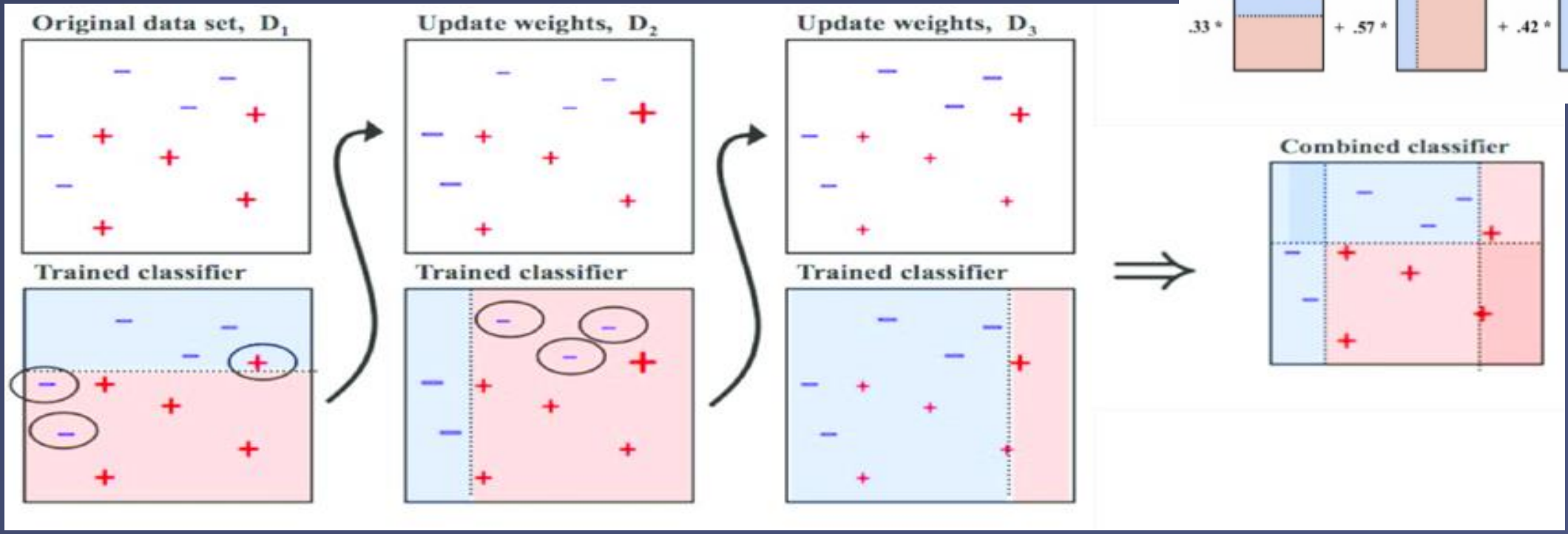
- Adaptive Boosting
  - First classifier 에서 잘못 예측한 데이터에 가중치를 부여
  - Second classifier는 잘못 예측한 데이터를 분류하는데 더 집중
  - Third classifier는 First, Second 가 잘못 예측한 데이터를 분류하는데 집중

- *Cost Function :* 가중치를 반영하여 계산

$$J(\theta) = \sum_i w_i J_i(\theta, x^{(i)})$$

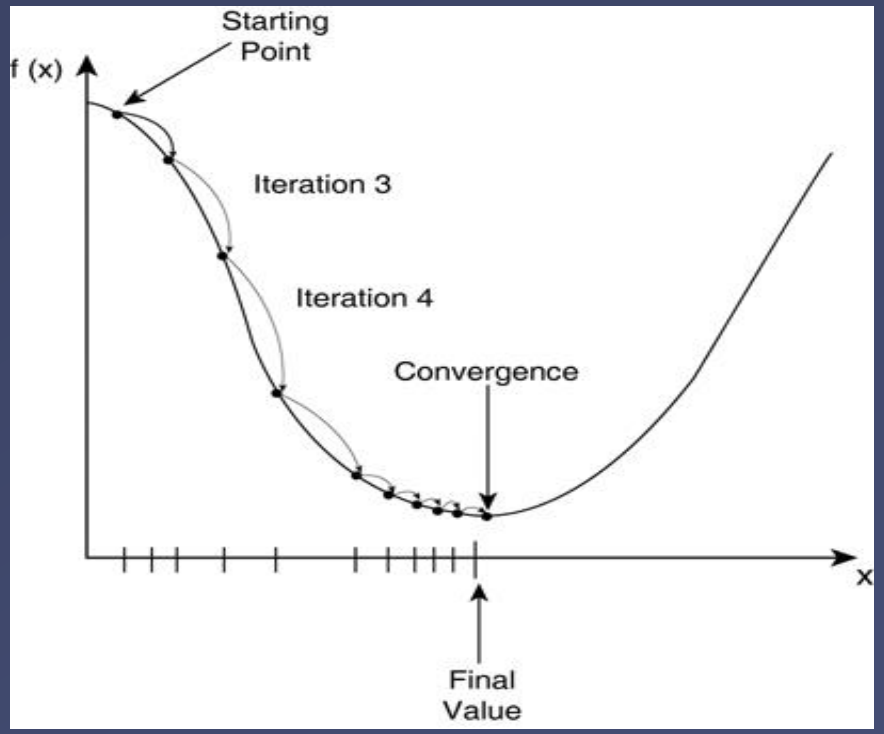
- 3개의 모델별로 계산된 가중치를 합산하여 최종 모델 생성





• Gradient Boosting

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \eta \nabla f(\mathbf{x}_{t-1})$$



- Gradient Descent 와 Boosting의 합성어, Boosting에 Gradient Descent를 접목시킨 머신러닝 알고리즘
- 약한 분류기들을 단계적으로 부스팅하는 과정에서 이전 모델의 오류를 손실함수로 나타내고 이 손실함수를 최소화하는 방법으로 Gradient Descent를 사용하는 알고리즘
- 즉 y값을 예측하는 약한 분류기 모델과 실제 값의 잔차를 통해 예측 성능을 올리고자 함

- Gradient Boosting

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$

   (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute

$$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$

   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

- Gradient Boosting

Loss Function : $\frac{1}{2}(Observed - Predicted)^2$

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

- 최소화 하는 γ 값을 찾기 위하여 미분을 이용 !

$$\frac{\partial}{\partial\, Predicted}\frac{1}{2}(Observed - Predicted)^2$$

$$= -(Observed - Predicted)$$

즉, 밑의 식을 만족하는 $Predicted$ 값이 $f_o(x)$!

-(88 - $Predicted$) + -(76 - $Predicted$) + -(56 -$Predicted$) = 0

$f_o(x)$ = 73.3

• Gradient Boosting

2. For $m = 1$ to M:

   (a) For $i = 1, 2, \ldots, N$ compute

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$

Loss Function : $\frac{1}{2}(Observed - Predicted)^2$

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{df(x_i)}\right]_{f=f_{m-1}} \quad f_{m-1} = f_0 = 73.3$$

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	-17.3

$$r_{1,1} = -\left[\frac{\partial \frac{1}{2}(88 - f(x_i))^2}{df(x_i)}\right]_{f=f_{m-1}}$$

$$r_{1,1} = (88 - f(x_i)) = 14.7$$

$$r_{2,1} = 2.7$$

$$r_{3,1} = -17.3$$

D&A

- Gradient Boosting

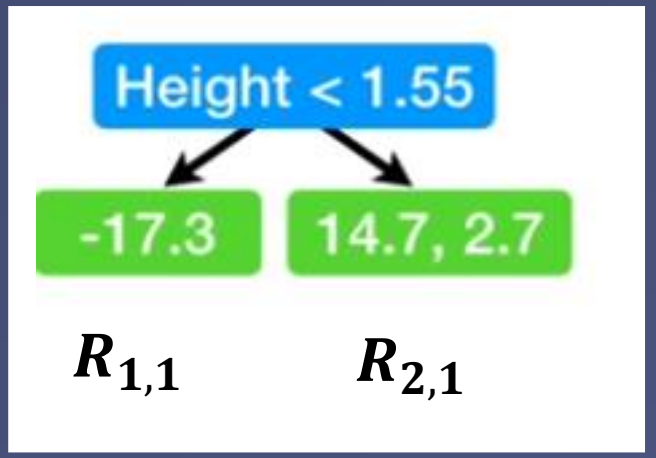
(b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	−17.3

- Gradient Boosting

(b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	−17.3



• Gradient Boosting

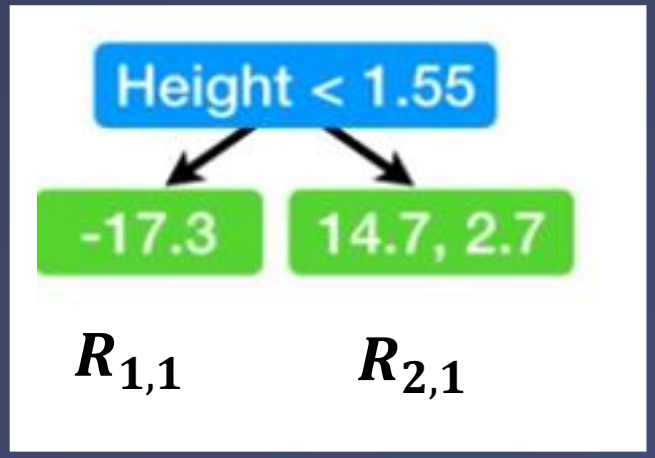
- Step 1 과 동일하게 Loss Function을 최소화 하는 γ 값을 찾아야 하지만 다른점은 이전 Prediction을 고려한다는 것이다.
- 또한 $x_i \in R_{JM}$ 를 통해 포함되는 샘플들만 고려하여 Summation 진행

(c) For $j = 1, 2, \ldots, J_m$ compute

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$

Step 1 :

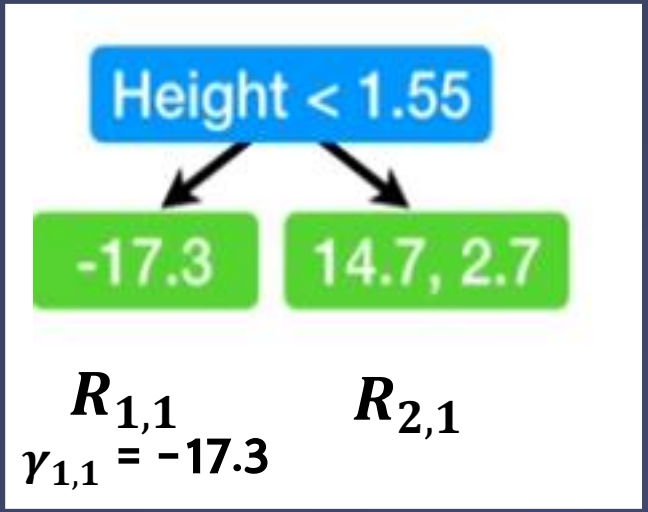
$$f_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma).$$



$$\gamma_{1,1} = \underset{\gamma}{arg\min} \frac{1}{2}(y_3 - (F_{m-1}(x_3) + \gamma))^2$$

$$\gamma_{2,1} = \underset{\gamma}{arg\min} \frac{1}{2}\{(y_1 - (F_{m-1}(x_1) + \gamma))^2 + (y_2 - (F_{m-1}(x_2) + \gamma))^2\}$$

- Gradient Boosting



Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	−17.3

$$\gamma_{1,1} = \underset{\gamma}{argmin} \frac{1}{2}(y_3 - (F_{m-1}(x_3) + \gamma))^2$$

$f_o(x)$ = 73.3

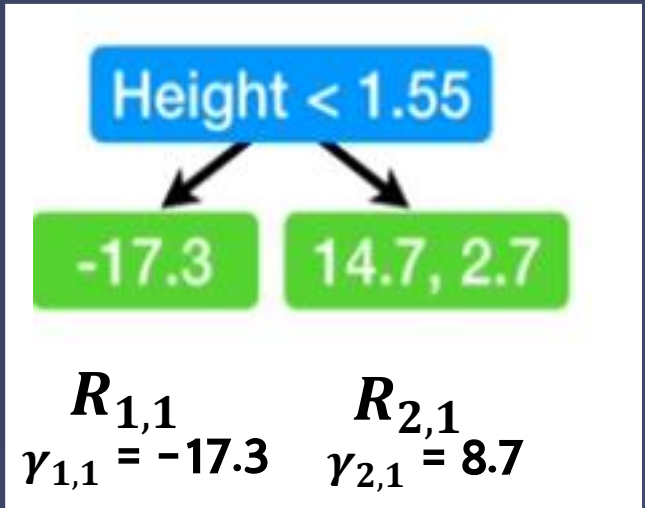
$$= \underset{\gamma}{argmin} \frac{1}{2}(56 - (73.3 + \gamma))^2$$

$$= \underset{\gamma}{argmin} \frac{1}{2}(-17.3 - \gamma)^2$$

$$\frac{d}{d\gamma}\frac{1}{2}(-17.3 - \gamma)^2 \rightarrow 17.3 + \gamma = 0$$

$\gamma_{1,1}$ = −17.3

- Gradient Boosting



$R_{1,1}$ $\gamma_{1,1}$ = −17.3 $R_{2,1}$ $\gamma_{2,1}$ = 8.7

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$
1.6	Blue	Male	88	14.7
1.6	Green	Female	76	2.7
1.5	Blue	Female	56	−17.3

$f_o(x)$ = 73.3

$$\gamma_{2,1} = \underset{\gamma}{argmin}\frac{1}{2}\{(y_1 - (F_{m-1}(x_1) + \gamma))^2 + (y_2 - (F_{m-1}(x_2) + \gamma))^2\}$$
$$= \underset{\gamma}{argmin}\frac{1}{2}\{(88 - (73.3 + \gamma))^2 + (76 - (73.3 + \gamma))^2\}$$
$$= \underset{\gamma}{argmin}\frac{1}{2}\{(14.7 - \gamma)^2 + (2.7 - \gamma)^2\}$$

$$\frac{d}{d\gamma}\frac{1}{2}\{(14.7 - \gamma)^2 + (2.7 - \gamma)\}^2 \rightarrow -14.7 + \gamma + -2.7 + \gamma = 0$$

$\gamma_{2,1} = \frac{14.7+2.7}{2}$ = 8.7

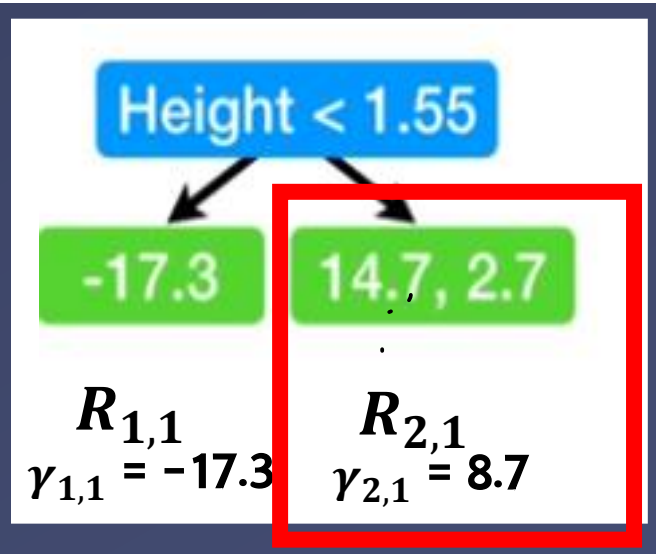
- Gradient Boosting

(d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

$f_0(x)$

$f_1(x)$ = 73.3 + η x

η = Learning rate
= 0.1



Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

$f_o(x)$ = 73.3

$f_1(x_1)$ = 73.3 + 0.1 x 8.7 = 74.2

Weight (kg)	$r_{i,1}$	$r_{i,2}$
88	14.7	13.8
76	2.7	
56	−17.3	

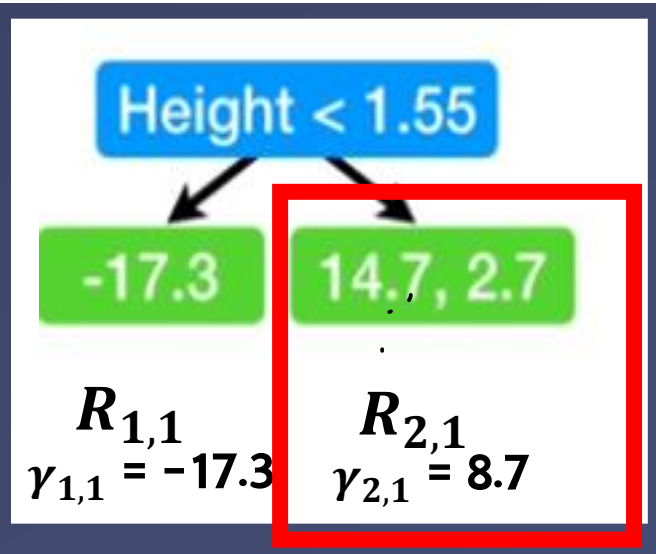
- Gradient Boosting

(d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

$f_0(x)$

$f_1(x)$ = 73.3 + η x

η = Learning rate = 0.1



Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

$f_o(x)$ = 73.3

$f_1(x_2)$ = 73.3 + 0.1 x 8.7 = 74.2

Weight (kg)	$r_{i,1}$	$r_{i,2}$
88	14.7	13.8
76	2.7	1.8
56	−17.3	

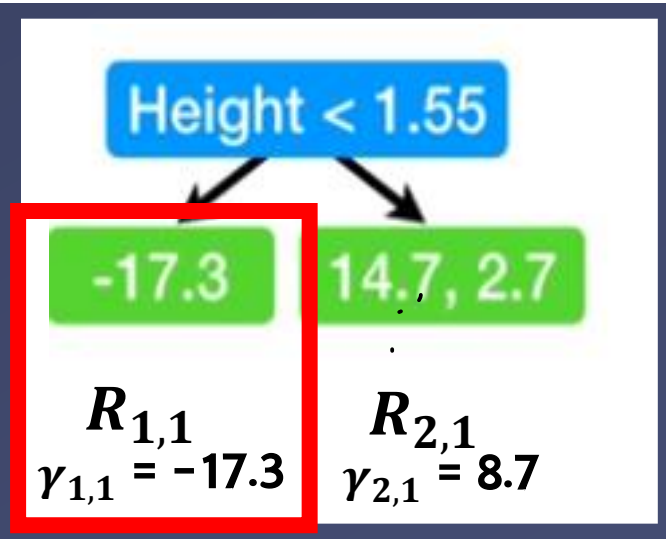
- Gradient Boosting

(d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

$f_0(x)$

$f_1(x)$ = 73.3 + η x

η = Learning rate = 0.1



Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

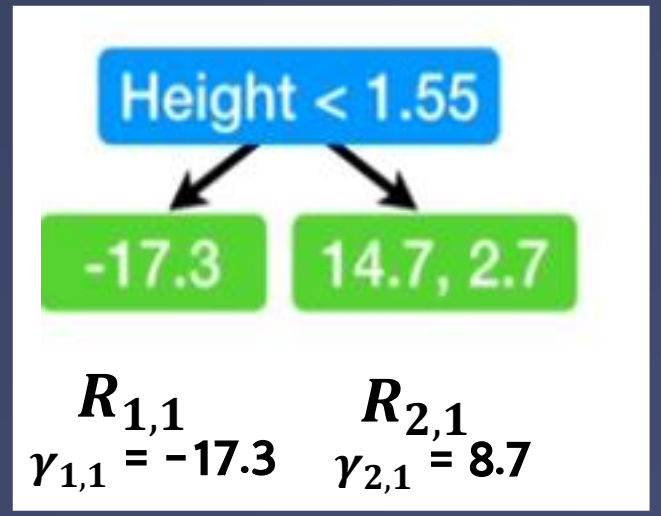
$f_o(x)$ = 73.3

$f_1(x_3)$ = 73.3 + 0.1 x −17.3 = 74.2

Weight (kg)	$r_{i,1}$	$r_{i,2}$
88	14.7	13.8
76	2.7	1.8
56	−17.3	−15.6

- Gradient Boosting

$f_1(x)$



$R_{1,1}$ $\gamma_{1,1} = -17.3$

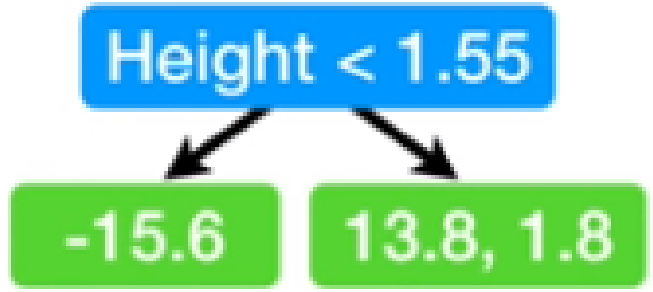
$R_{2,1}$ $\gamma_{2,1} = 8.7$

$f_0(x)$

$f_2(x) = 73.3 + \eta \times$

η = Learning rate = 0.1

$+ \eta \times$



$R_{1,2}$ $\gamma_{1,2} = -15.6$

$R_{2,2}$ $\gamma_{2,2} = 7.8$

Height (m)	Favorite Color	Gender	Weight (kg)	$r_{i,1}$	$r_{i,2}$	$r_{i,3}$
1.6	Blue	Male	88	14.7	13.8	12.8
1.6	Green	Female	76	2.7	1.8	0.8
1.5	Blue	Female	56	-17.3	-15.6	14

• eXtreme Gradient Boosting (XGB)

- Gradient Boosting 의 단점 :

1. 수행시간이 오래걸리고, 하이퍼 파라미터 튜닝 노력이 필요

2. weak classifier의 순차적인 예측 오류(잔차) 보정을 통해 학습을 진행하기 대문에 멀티 CPU 코어 시스템을 사용하더라도 병렬 처리가 지원되지 않아서 대용량 데이터의 경우 학습에 매우 많은 시간이 필요

- XGB의 특징 :

1. 병렬/분산처리 가능하기 때문에 대용량 데이터의 경우 Gradient Boosting에 비해 학습이 빠르다

2. SPLIT 지점을 일부만 보고 결정 가능

3. 모델의 성능과 복잡성을 동시에 고려 (복잡성 ↑ Variance ↑ Overfitting 가능성 증가)

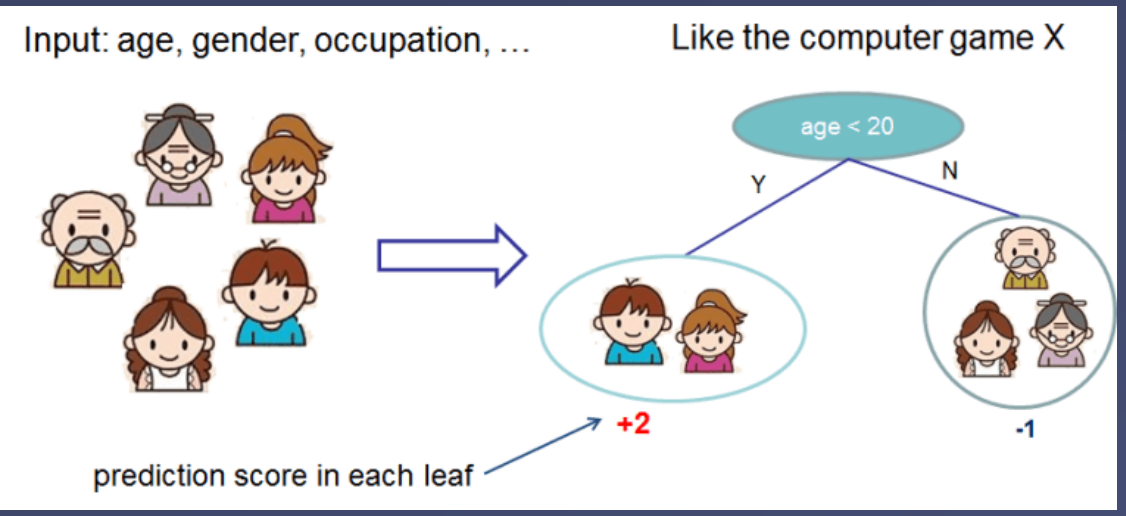
• eXtreme Gradient Boosting (XGB)

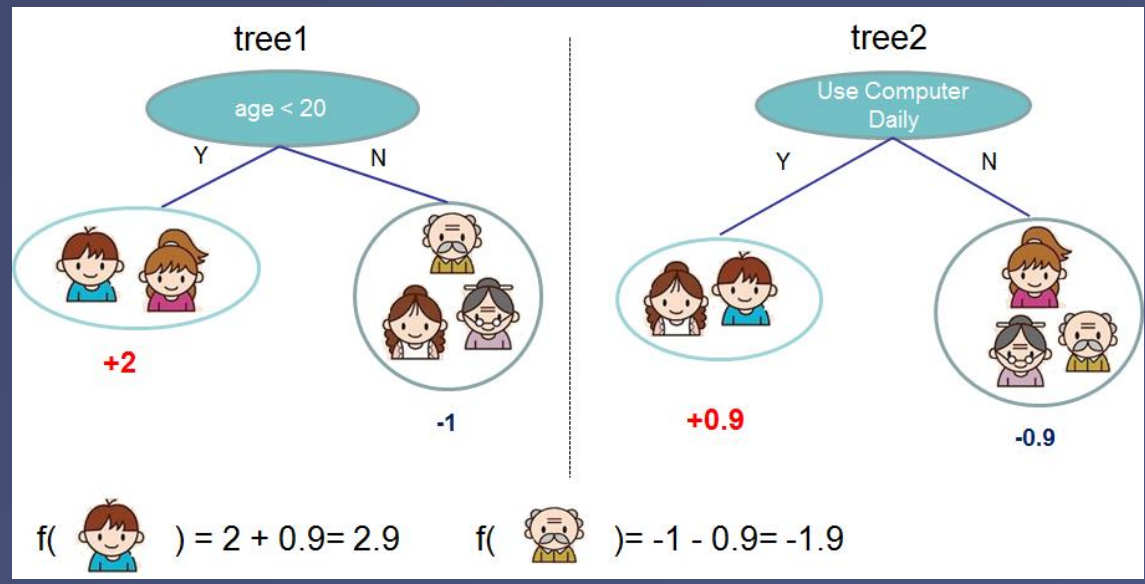
1. 병렬/분산처리 가능

- 각자 할당받은 변수들로 tree들을 생성
- 그 후 모든 tree들을 통해 Ensemble하여 예측함

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

K = the number of trees
f = function in the function space $\mathcal{F}$

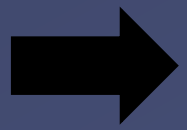




- eXtreme Gradient Boosting (XGB)

2. SPLIT 지점을 일부만 보고 결정 가능

Height (m)	Favorite Color	Gender
1.6	Blue	Male
1.6	Green	Female
1.5	Blue	Female
1.7	Red	Male
1.6	Green	Female
1.4	Red	Female



GBM보다 더 적은 비용으로,
더 Bias가 낮은 결과를 얻을 수 있다.

• eXtreme Gradient Boosting (XGB)

2. SPLIT 지점을 일부만 보고 결정 가능

- Sparsity Awareness 가능 : '0' 인 데이터를 건너 뛰면서 학습이 가능하다 !

ID	거주지역
1	서울
2	대전
3	대구
4	부산
5	제주도

<원데이터>

ID	서울	대전	대구	부산	제주도
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

<더미 매트릭스>

• eXtreme Gradient Boosting (XGB)

3. 모델의 성능과 복잡성을 동시에 고려하는 Loss Function을 사용

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training loss　　Complexity of the Trees

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

Number of leaves　　L2 norm of leaf scores

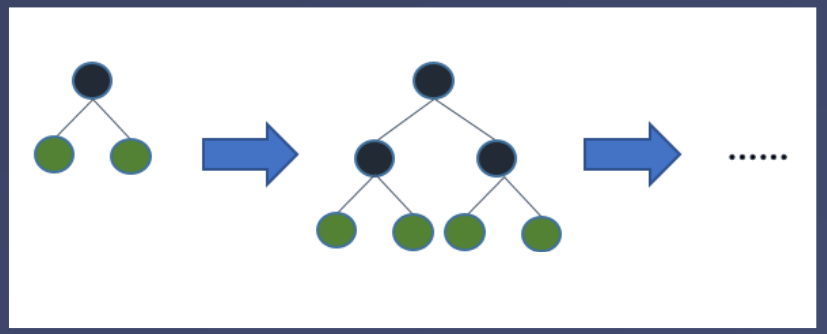
- 기존의 Loss Function에 모델의 복잡성을 고려하여 Tree들 간의 Variance를 L2 norm을 이용하여 규제함
  → Overfitting 방지

- tree모델은 계수가 없기 때문에 w는 마지막 노드의 예측값
  → 개별모델의 예측값을 규제함

• Light Gradient Boosting (LGBM)

1. 기존 Boosting 알고리즘과 차이점

Level – wise tree

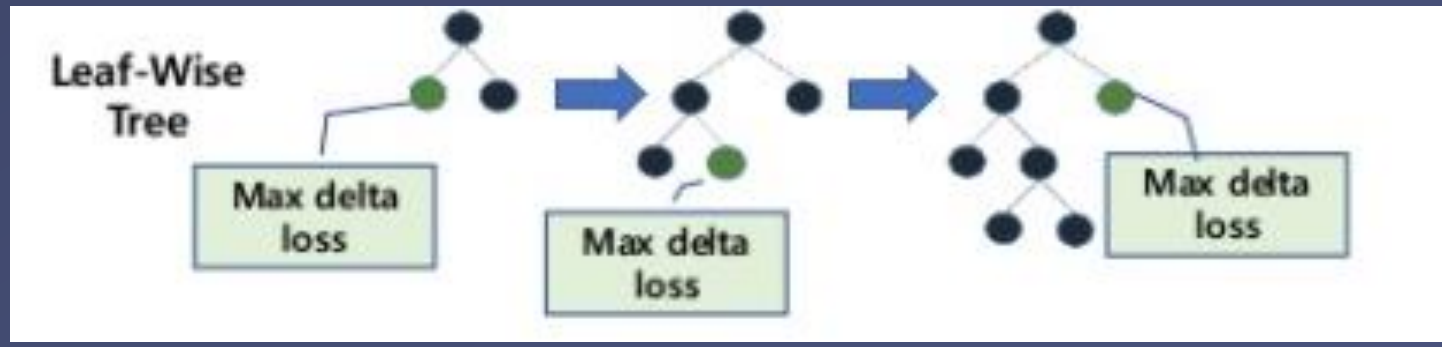


Level – wise tree

사용 모델 : RF, GB

성장 방법 :
Root에서의 거리를 기준으로
수평성장

Leaf – wise tree



Leaf – wise tree

사용 모델 : LGBM, XGB

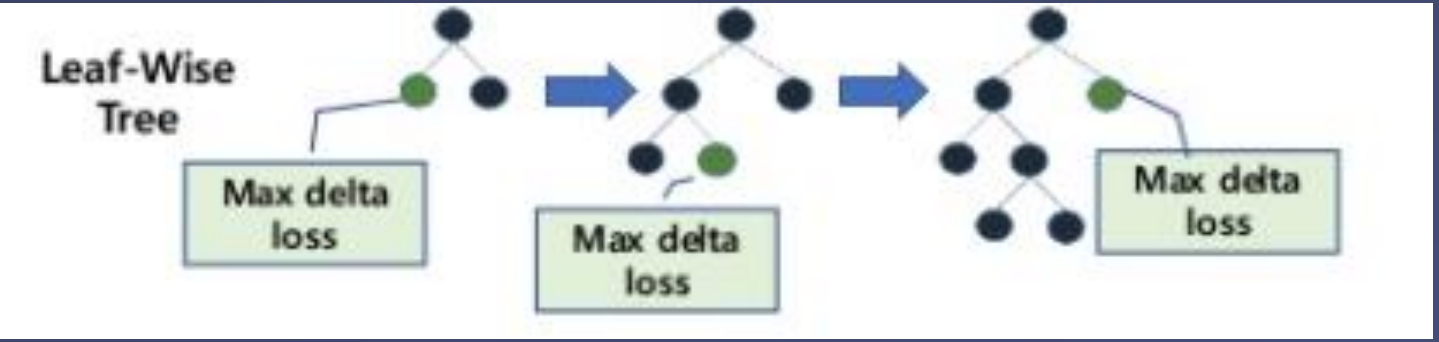
성장 방법 :
가장 Loss의 변화가 큰 노드에서
수직성장

D&A

• Light Gradient Boosting (LGBM)

1. 기존 Boosting 알고리즘과 차이점

Leaf – wise tree



특징

- 속도가 빠르고, 성능이 좋다.
- 저장 공간을 덜 차지한다.
- 병렬적인 학습(동시적)이 가능하다.
- Overfitting에 민감하다.

대용량 data에 적합
(적어도 10,000건 이상)

D&A

- BOOSTING 기반 알고리즘

알고리즘	특징	한계점
Gradient boosting (GBM)	Loss Function의 Gradient를 통해 오답에 가중치 부여	시간소요가 길다.
eXtreme Gradient Boosting (XGB)	**GBM 대비 성능 향상** Kaggle을 통한 성능 검증	시간소요가 길다.
Light GBM (LGBM)	**XGB 대비 자원소모 최소화** (시간소요가 가장 짧음) => 대용량 데이터에 적합	Overfitting 가능성 ↑