DACON 스터디
5주차

Ensemble learning

이제윤

20220802

CONTENTS

- Ensemble Learing이란
- Ensemble Learning의 종류
  - Voting
  - Stacking
  - Bagging
  - Boosting
- 과제

Ensemble Learing이란

- 앙상블 학습(Ensemble Learning)은 **여러 개의 분류기를 생성하고, 그 예측을 결합함으로써 더 정확한 예측을 도출하는 기법**을 말합니다. 강력한 하나의 모델을 사용하는 대신 약한 모델 여러 개를 조합하여 더 정확한 예측에 도움을 주는 방식입니다.

- 보통 Classification에서 자주 사용됨.

Ensemble Learning의 종류

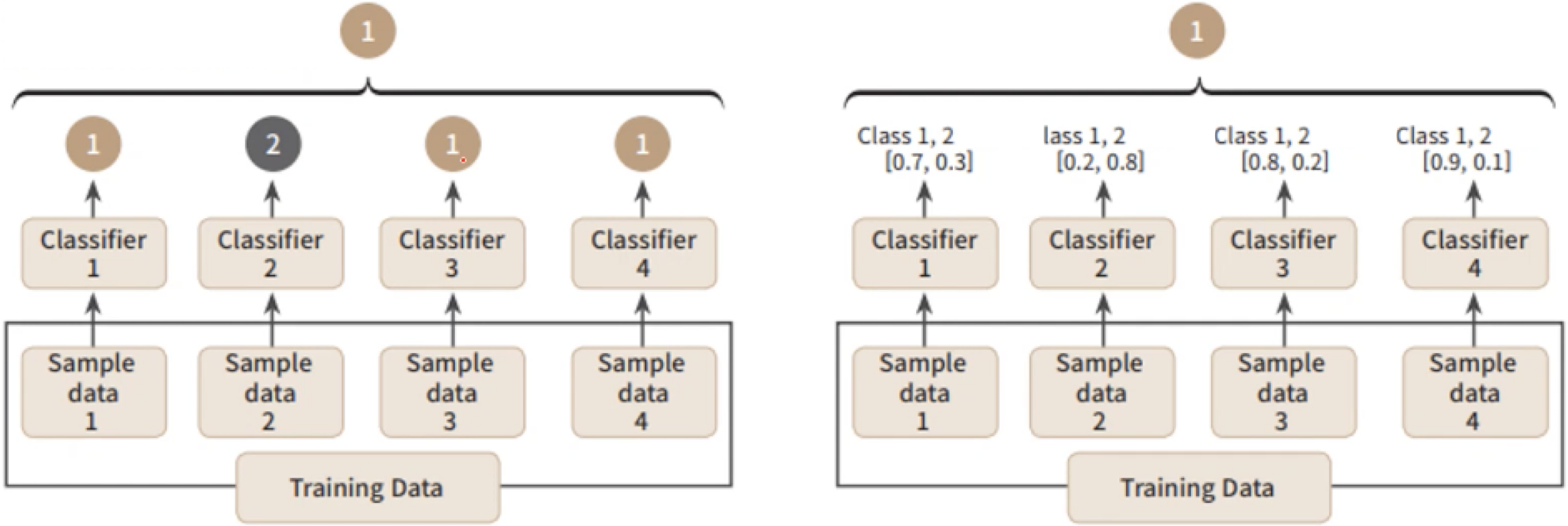
VOTING BASE(병렬적 과정)

1. Voting
2. Stacking
3. Bagging

BOOSTING BASE(순차적 과정)

1. Boosting

1. Voting



<Hard voting vs Soft voting>

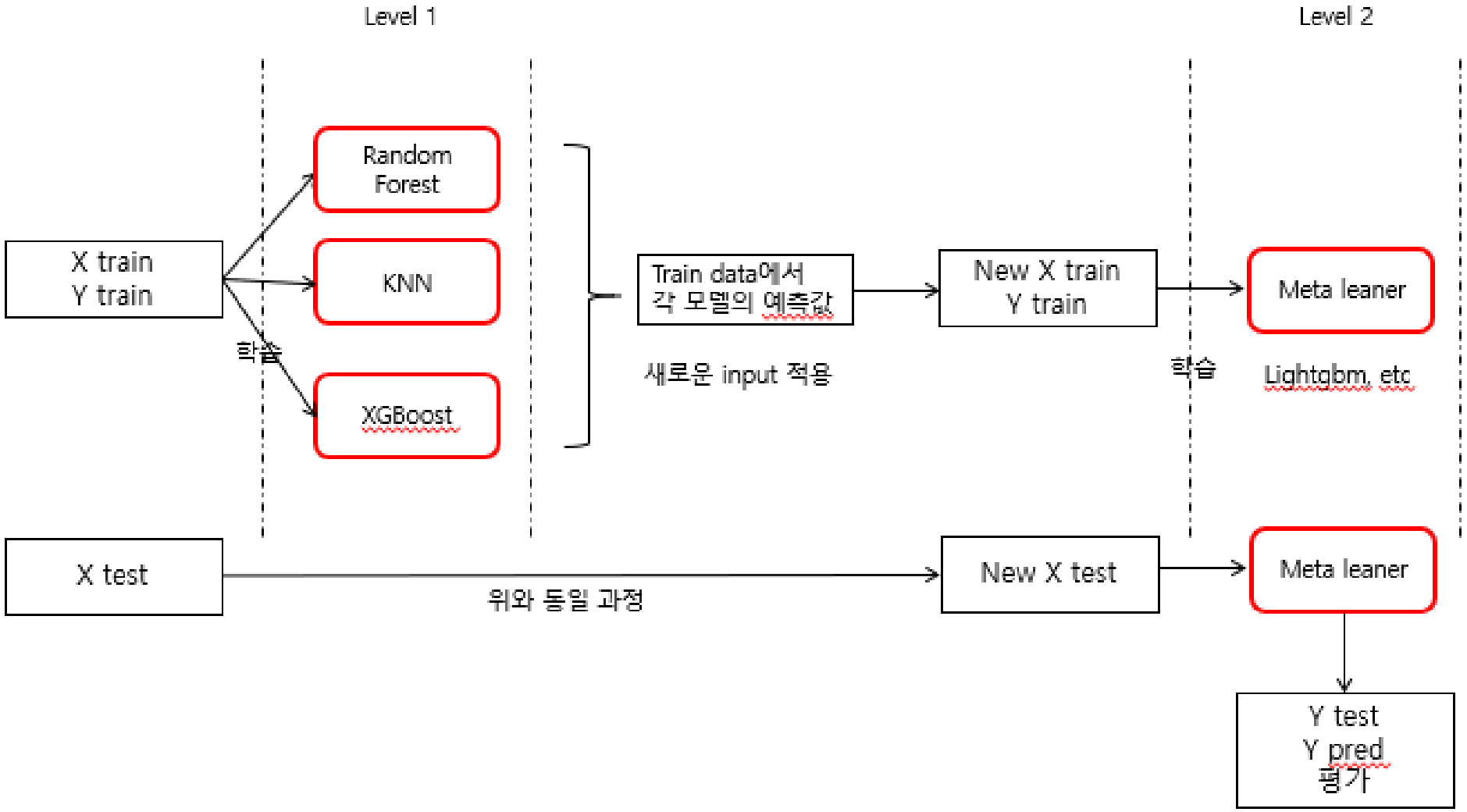
2. Stacking

- 스태킹은 meta-level learning이라고도 불림
- 스태킹을 가장 간단하게 정리하자면 예측값으로 실제값을 다시 예측하는 기법
- 일반 앙상블 알고리즘과 다르게 2단계로 학습을 진행

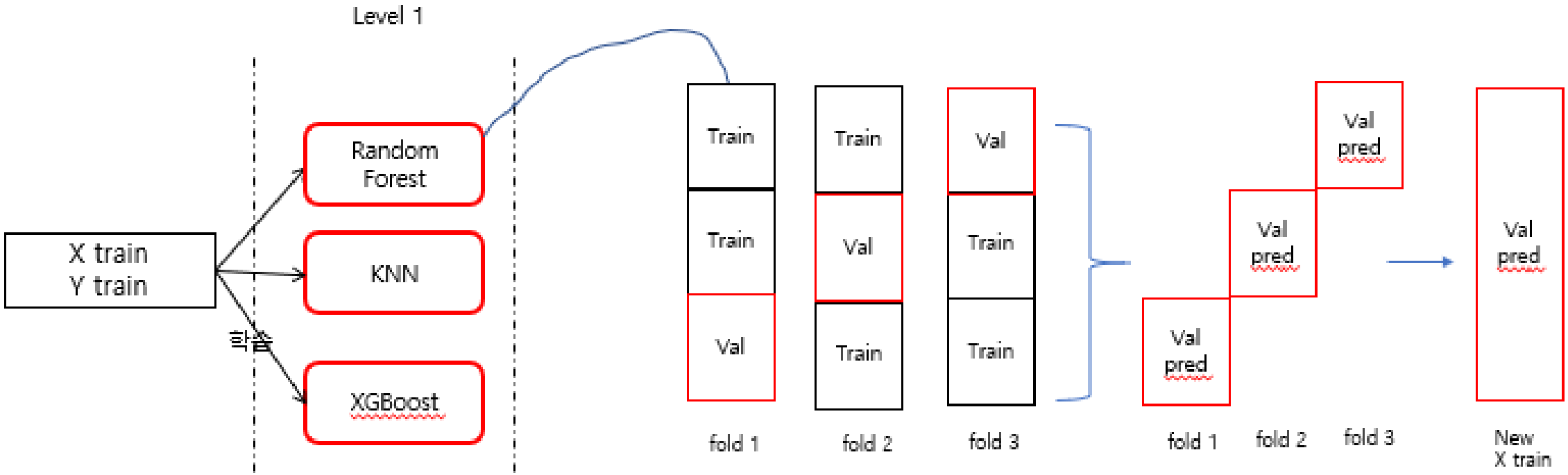
- Classification 문제에서만 사용
- 과적합 문제가 있어서 CV 거의 필수적으로 사용
- 실용적으로 많이 쓰지는 않음.

2. Stacking

[바람돌이/머신러닝] 앙상블(Ensemble Learning)(3) - 스태킹(Stacking) 이론 : 네이버 블로그



2. Stacking (with CV)



3. Bagging

- Decision Tree를 전체 Train에 대해 사용한다면… Overfitting 우려
- Train set을 일부 추출하게 되면 그 우려를 줄일 수 있음.
- 그 추출 방법 중에 Booststrap 방법을 사용하고
- 결과들을 aggregating 하면 Bagging

3. Bagging

배깅(Bagging)은 Bootstrap Aggregating의 약자로, 보팅(Voting)과는 달리 동일한 알고리즘으로 여러 분류기를 만들어 보팅으로 최종 결정하는 알고리즘

**배깅은 다음과 같은 방식으로 진행이 됩니다.

(1) 동일한 알고리즘을 사용하는 일정 수의 분류기 생성

(2)각각의 분류기는 **부트스트래핑(Bootstrapping)**방식으로 생성된 샘플데이터를 학습

(3)최종적으로 모든 분류기가 보팅을 통해 예측 결정

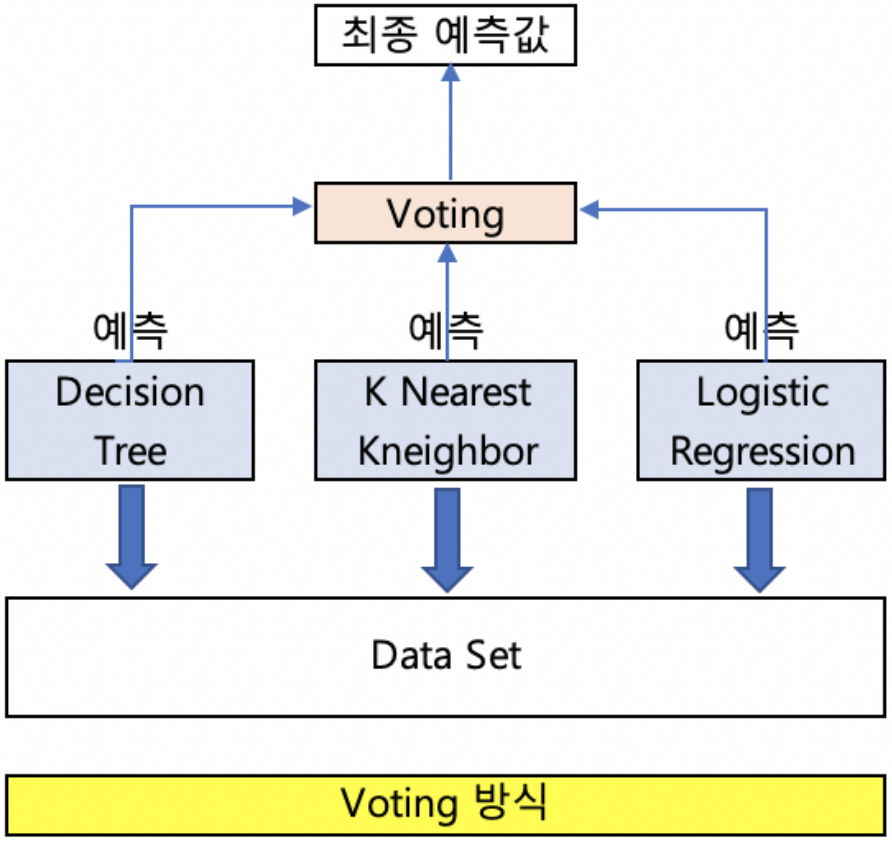
※ 부트스트래핑 샘플링은 전체 데이터에서 일부 데이터의 중첩을 허용하는 방식

Booststrapping이란..

EX)
9개의 Train set 중
9개씩 랜덤하게
복원추출로 뽑기
cf) crossvalidation

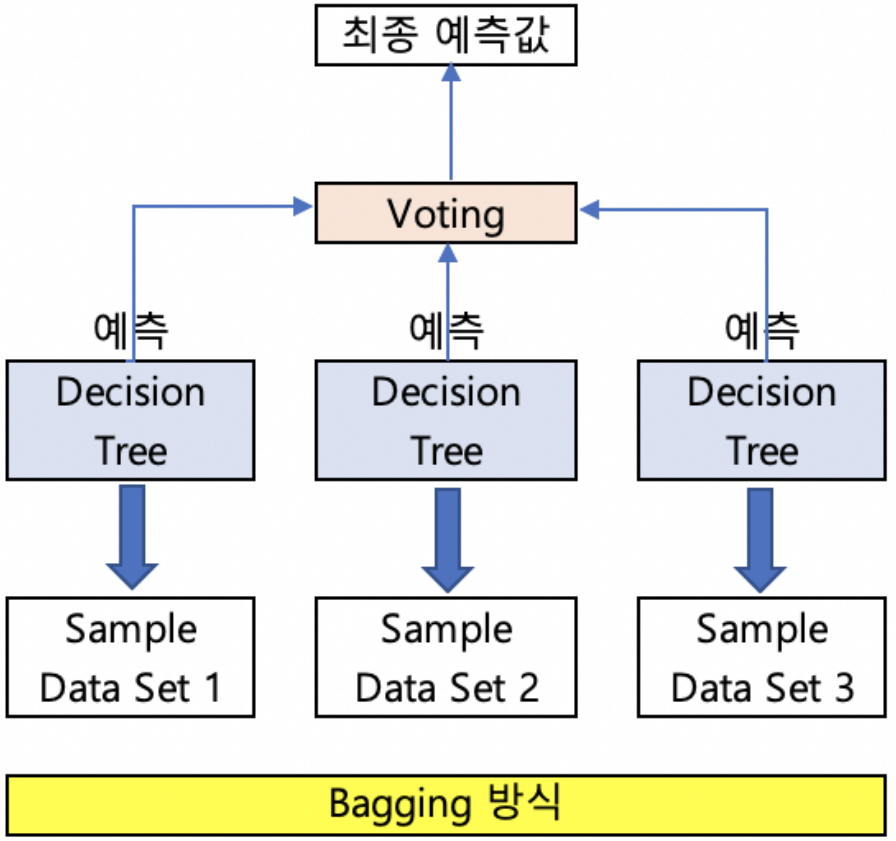
=> Voting

3. Bagging



Voting

- Generally combine **different** classifiers
- Classifiers use identical dataset



Bagging

- Generally use **identical** classifiers
- Classifiers use different datasets

RandomForest

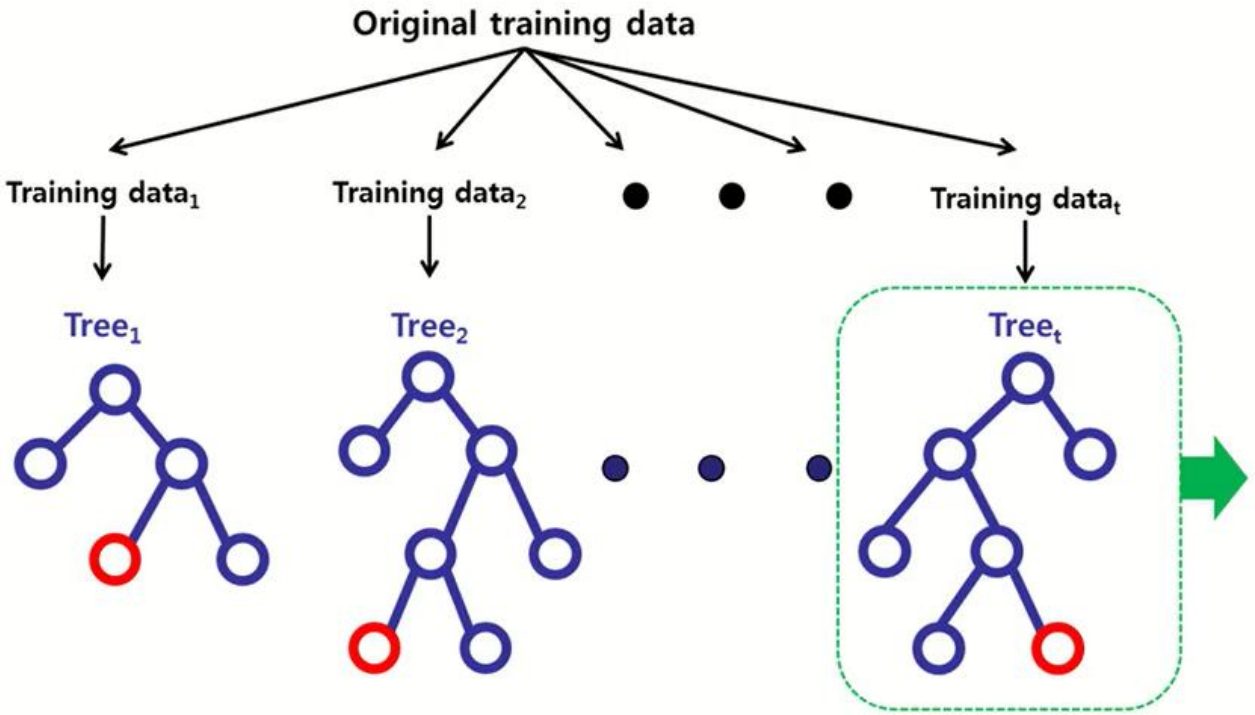
[핵심 머신러닝] Boosting - YouTube

- 핵심 아이디어: Diversity, Random 확보

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

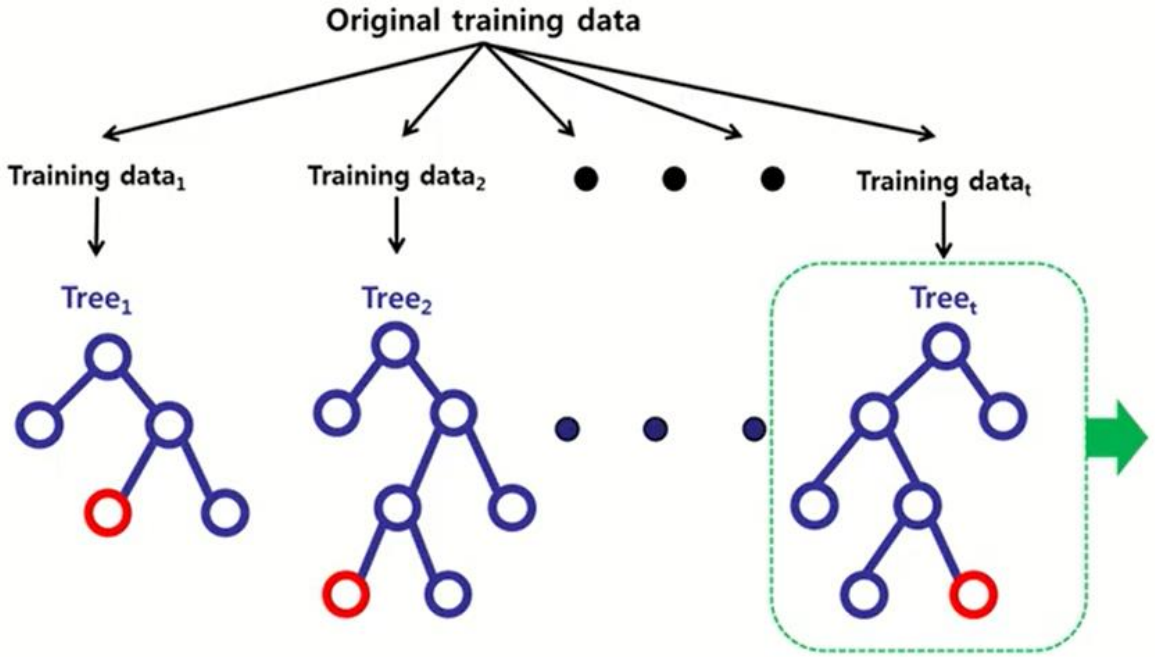
2. 의사결정나무모델 구축 시 변수 무작위로 선택
→ Random subspace

RandomForest



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

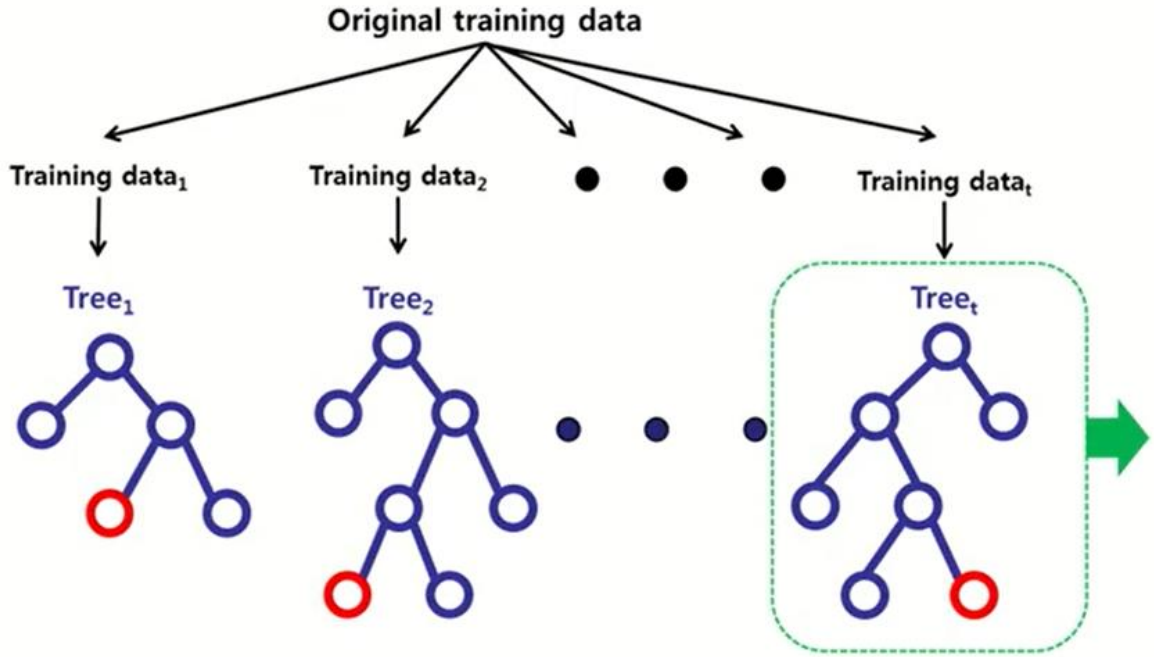
RandomForest

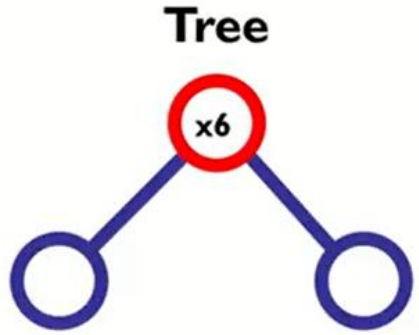


원래 변수	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
입력 변수			x3		x5	x6				x10						

I. 원래 변수들 중에서 모델 구축에 쓰일 입력 변수를 무작위로 선택

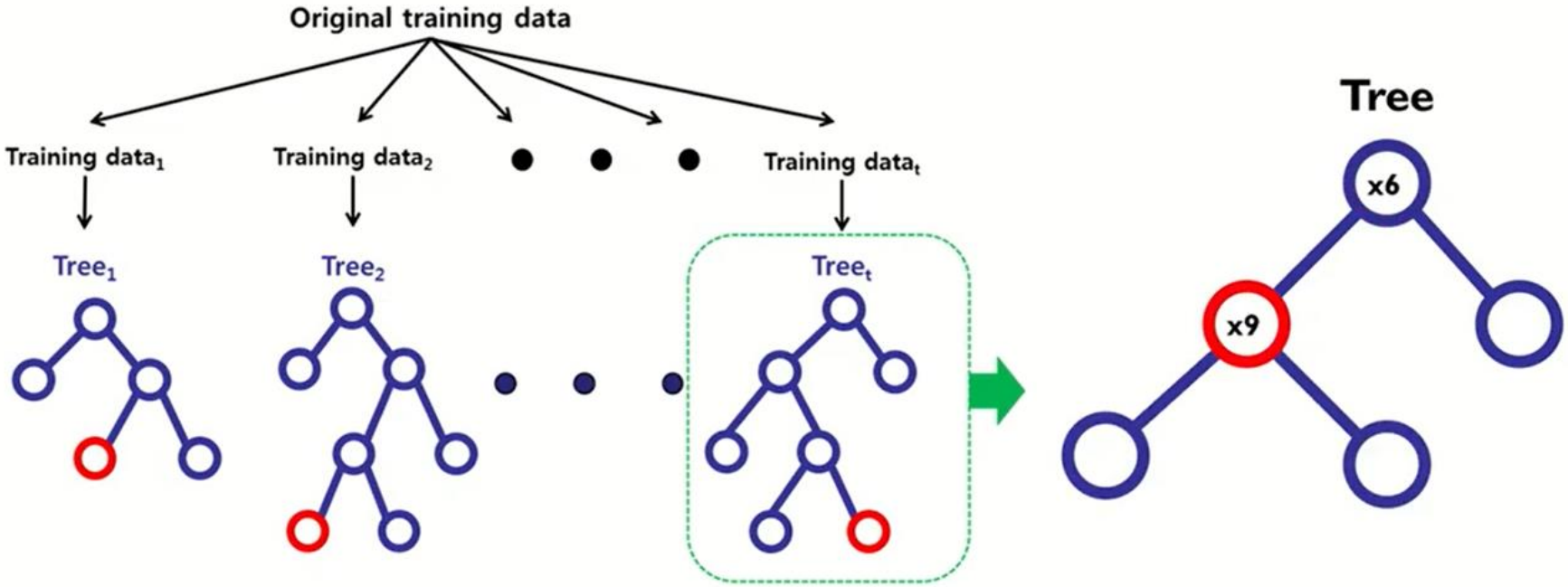
RandomForest





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 중에 분할될 변수를 선택



Original Features	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
Input Features	x1				x5				x9	x10						

3. 이러한 과정을 **full-grown tree**가 될 때까지 반복

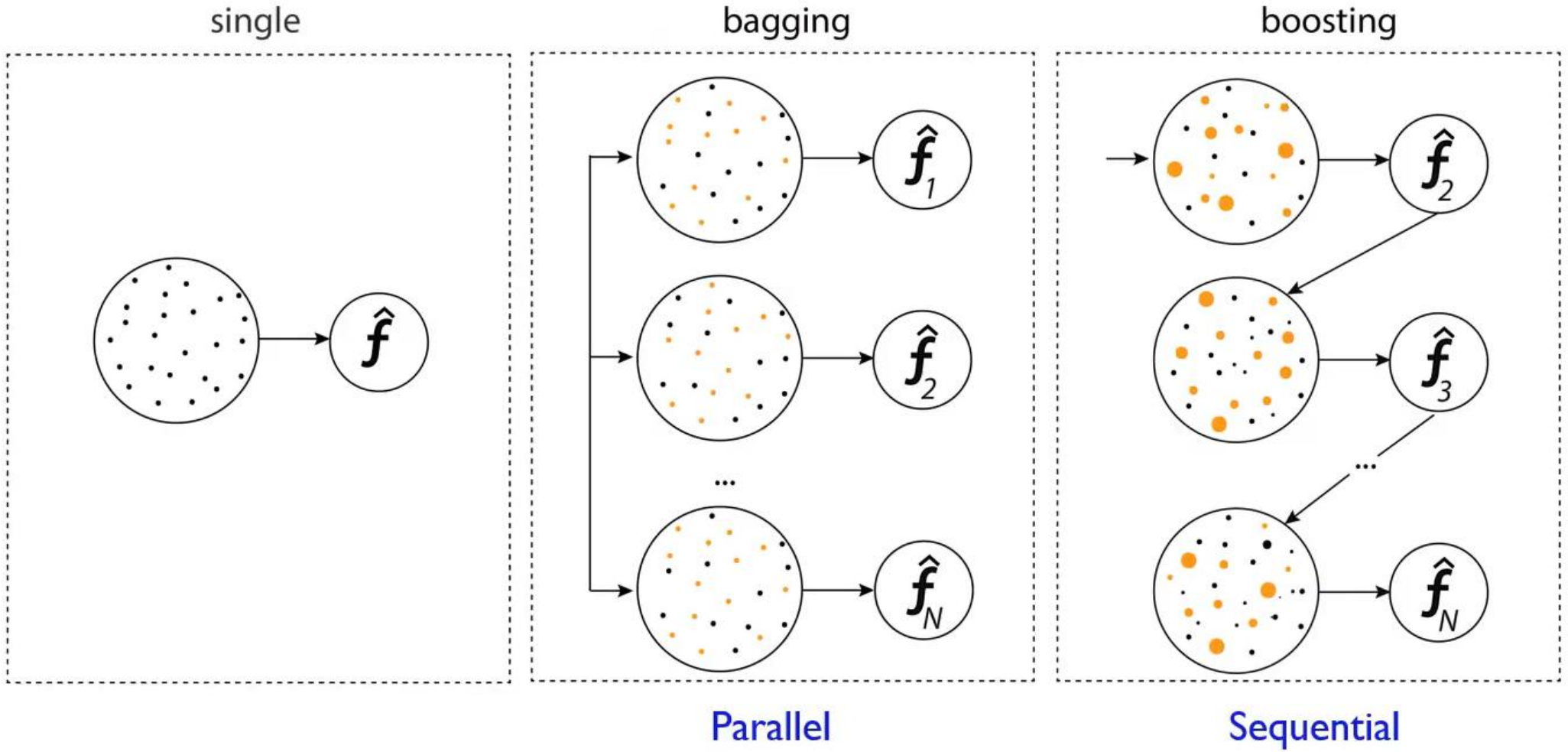
4. Boosting

[핵심 머신러닝] Boosting - YouTube

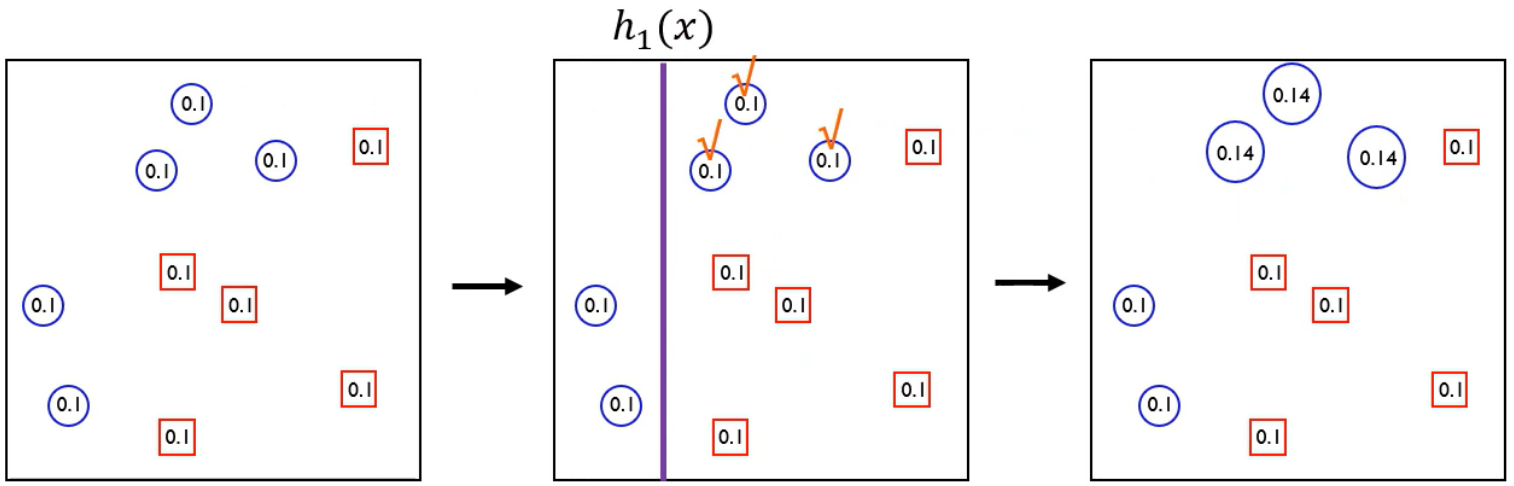
❖ Boosting 아이디어

- 여러 개의 learning 모델을 순차적으로 구축하여 최종적으로 합침 (앙상블)
- 여기서 사용하는 learning 모델은 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

4. Boosting



AdaBoost



$$L_1 = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_1(x))}{\sum_{i=1}^{n} W_i} = \frac{0.1 \times 3}{0.1 \times 10} = 0.3$$ 10개중 3개 오분류

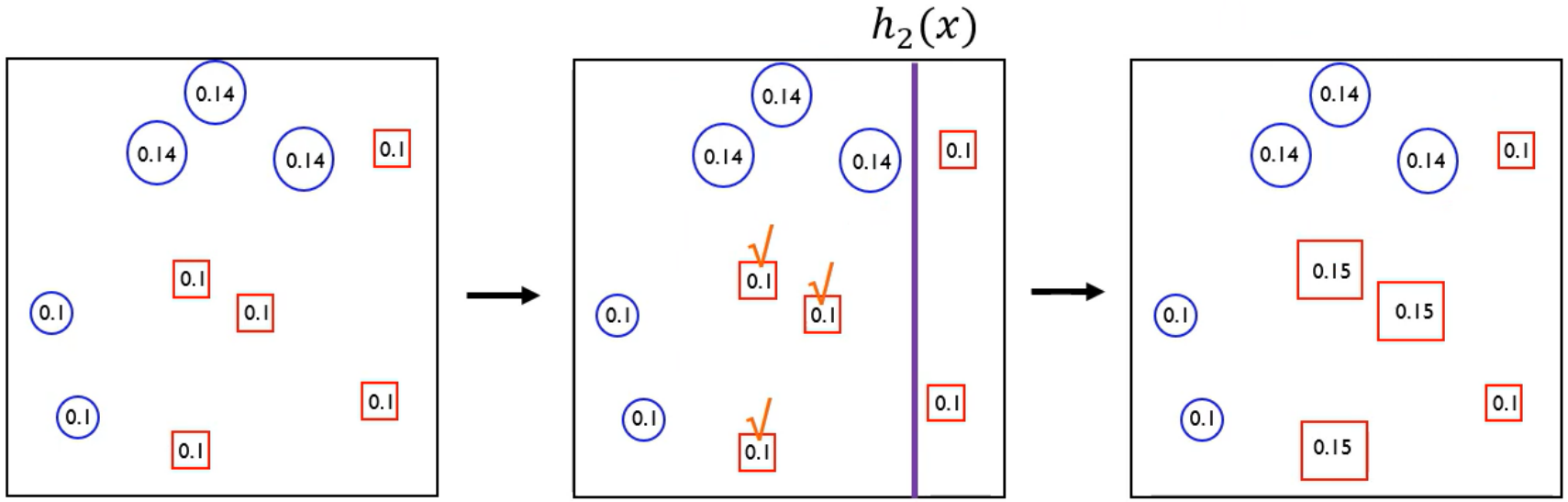
$$\alpha_1 = \log\left(\frac{1 - L_j}{L_j}\right) = \log\left(\frac{1 - 0.3}{0.3}\right) \approx 0.37$$

$W_c = W_i \, e^{\alpha_1 I(y_i \neq h_1(x))} = 0.1e^{0.37 \times 0} = 0.1$ 정분류 관측치 가중치

$W_{nc} = W_i \, e^{\alpha_1 I(y_i \neq h_1(x))} = 0.1e^{0.37 \times 1} = 0.14$ 오분류 √ 관측치 가중치

☺ 도형의 크기(안의 숫자)는 해당 관측치가 다음 단계의 학습데이터셋에 선택될 확률을 의미

AdaBoost



$$L_2 = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_2(x))}{\sum_{i=1}^{n} W_i} = \frac{0.1 \times 3}{0.1 \times 7 + 0.14 \times 3} = 0.27$$

10개중 3개 오분류

$$\alpha_2 = \log\left(\frac{1 - 0.27}{0.27}\right) \approx 0.43$$

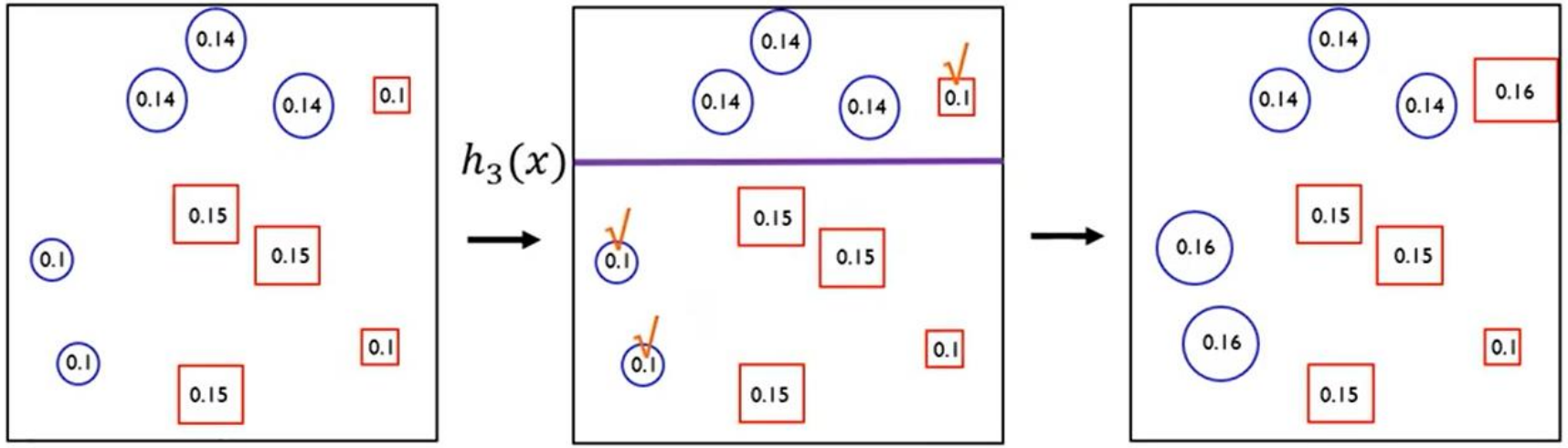
$$W_c = W_i \, e^{\alpha_2 I(y_i \neq h_2(x))} = \{0.1 \; or \; 0.14\} \, e^{0.43 \times 0} = 0.1 \; or \; 0.14$$

정분류 관측치 가중치

$$W_{nc} = W_i \, e^{\alpha_2 I(y_i \neq h_2(x))} = 0.1 e^{0.43 \times 1} = 0.15$$

오분류 √ 관측치 가중치

AdaBoost



$$L_3 = \frac{\sum_{i=1}^{n} W_i \, I(y_i \neq h_2(x))}{\sum_{i=1}^{n} W_i} = \frac{0.1 \times 3}{0.1 \times 4 + 0.14 \times 3 + 0.15 \times 3} = 0.24$$

10개중 3개 오분류

$$\alpha_3 = \log\left(\frac{1 - 0.24}{0.24}\right) \approx 0.5$$

$$W_c = W_i \, e^{\alpha_3 I(y_i \neq h_3(x))} = \{0.1 \; or \; 0.14 \; or \; 0.15\} \, e^{0.5 \times 0} = 0.1 \; or \; 0.14 \; or \; 0.15$$

정분류 관측치 가중치

$$W_{nc} = W_i \, e^{\alpha_3 I(y_i \neq h_3(x))} = 0.1 e^{0.5 \times 1} = 0.16$$

오분류 √ 관측치 가중치

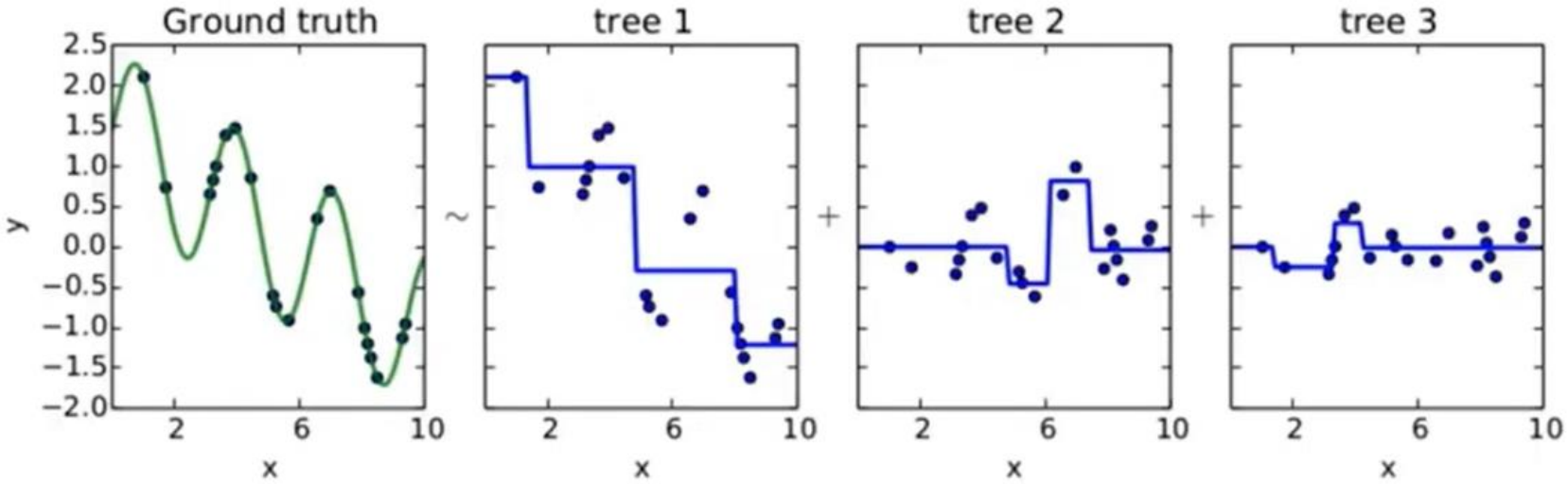
GBM (Gradient Boosting Machine)

Gradient boosting = Boosting with gradient decent

첫번째 단계의 모델 tree1을 통해 Y를 예측하고, Residual을 다시 두번째 단계 모델 tree2를 통해 예측하고, 여기서 발생한 Residual을 모델 tree3로 예측

점차 residual 작아 짐

Gradient boosted model = tree1 + tree2 + tree3



GBM (Gradient Boosting Machine)

❖ Why gradient?

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

Gradient

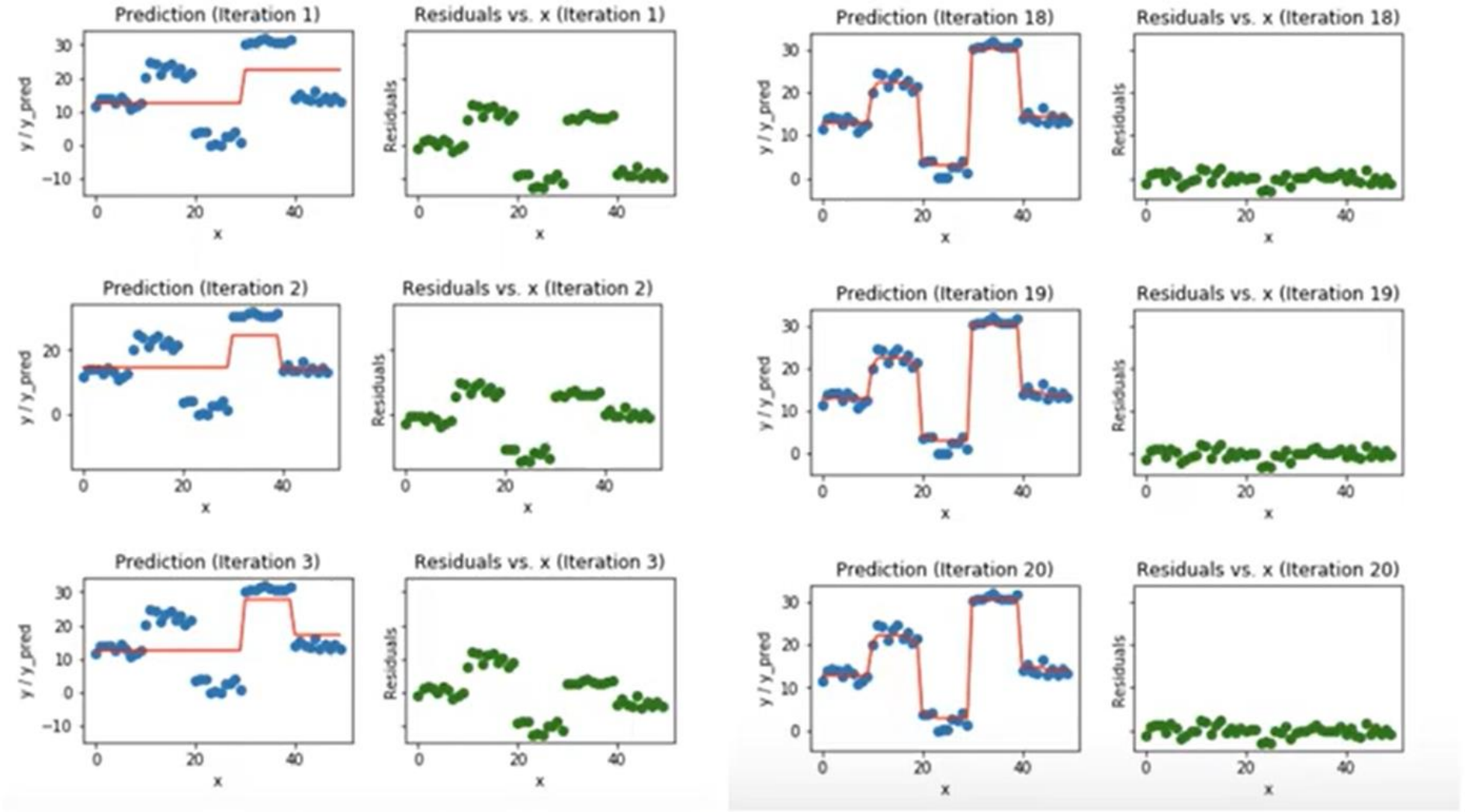
$$\frac{\partial L}{\partial f(x_i)} = -\{y_i - f(x_i)\}$$

Residual

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

Residual = Negative Gradient

GBM (Gradient Boosting Machine)

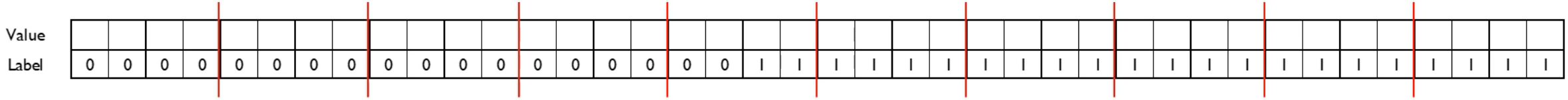


XGBoost

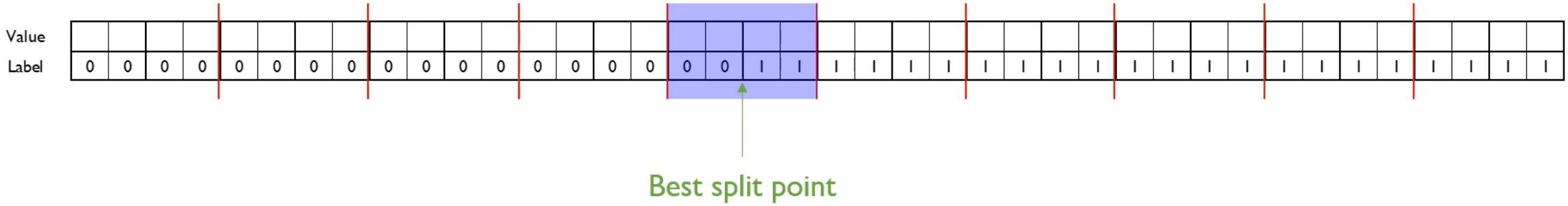
- Extreme Gradient Boost 의 준말.
- 기본적으로 DecisionTree를 Boosting하는 GradientBoost임.
- 그런데 그 속도를 향상시키고 병렬처리를 가능하게 하기 위해 XGBoost를 사용.
- 결측치까지도 고려하여 학습하는 효과가 있음.
- 규제화의 효과도 있어서 Overfitting을 막을 수 있음.

XGBoost

- Split Finding Algorithm
  - ✓ Approximate algorithm: an illustrative example
    - Assume that the value is sorted in an ascending order (left: small, right: large)
    - Divide the dataset into 10 buckets



- Compute the gradient for each bucket and find the best split



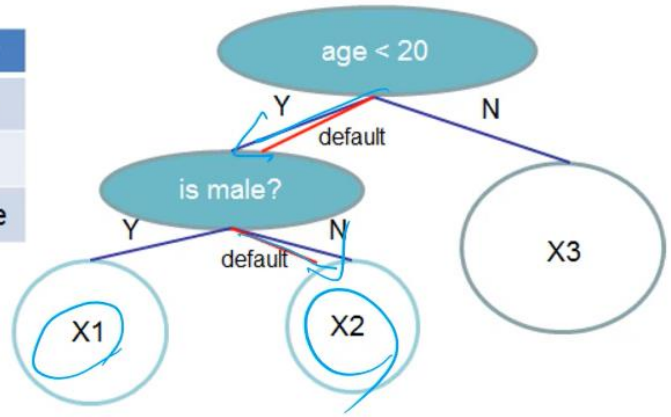
XGBoost

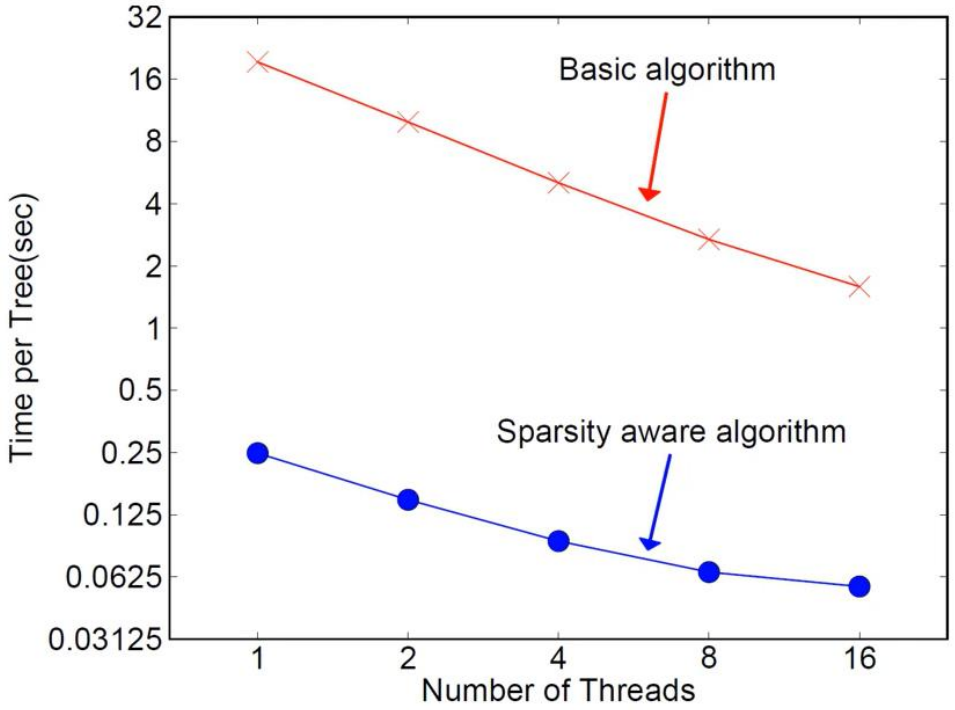
XGBoost

- Sparsity-Aware Split Finding

Data

Example	Age	Gender
X1	?	male
X2	15	?
X3	25	female





XGBoost

XGBoost

- Sparsity-Aware Split Finding

Algorithm 3: Sparsity-aware Split Finding

Input: I, instance set of current node
Input: $I_k = \{i \in I | x_{ik} \neq \text{missing}\}$
Input: d, feature dimension
Also applies to the approximate setting, only collect statistics of non-missing entries into buckets
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i, H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ ***to*** m **do**
  // enumerate missing value goto right
  $G_L \leftarrow 0,\ H_L \leftarrow 0$
  for j *in sorted*$(I_k,$ *ascent order by* $\mathbf{x}_{jk})$ **do**
    $G_L \leftarrow G_L + g_j,\ H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L,\ H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
  // enumerate missing value goto left
  $G_R \leftarrow 0,\ H_R \leftarrow 0$
  for j *in sorted*$(I_k,$ *descent order by* $\mathbf{x}_{jk})$ **do**
    $G_R \leftarrow G_R + g_j,\ H_R \leftarrow H_R + h_j$
    $G_L \leftarrow G - G_R,\ H_L \leftarrow H - H_R$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split and default directions with max gain

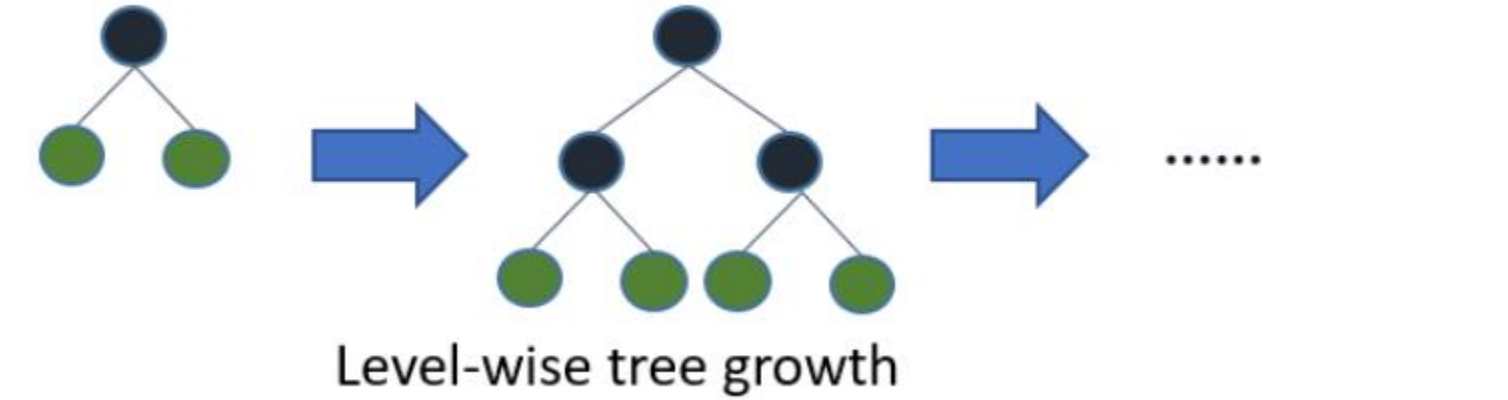
Value	1.3		1.1	0.2		1.9	0.5		1.5	1.8
Class	1	0	1	0	0	1	0	0	1	1

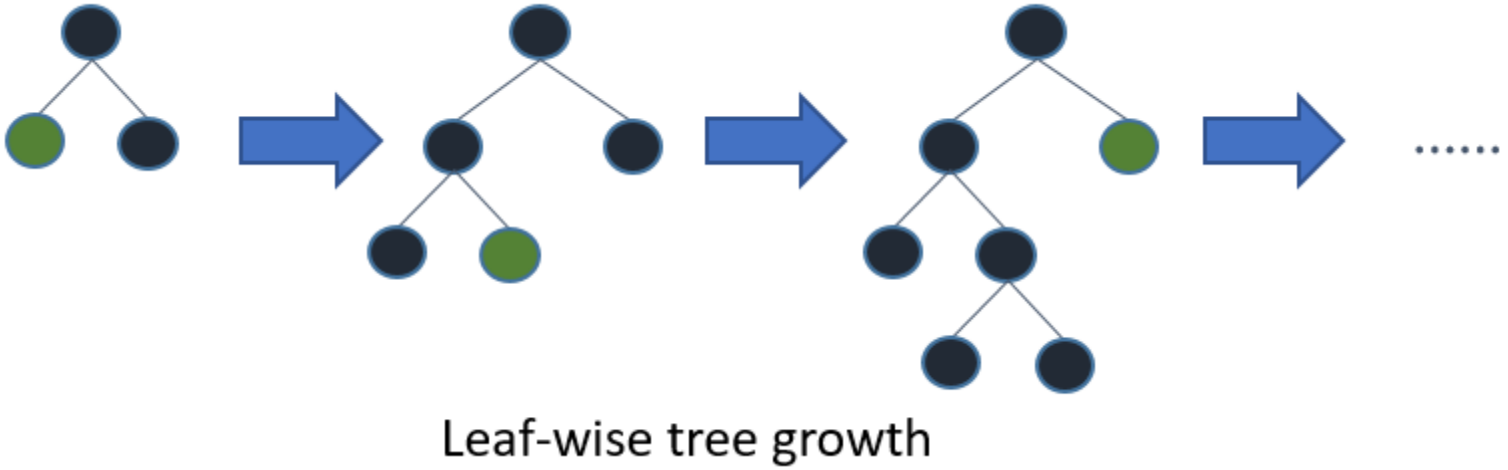
Value	0.2	0.5	0.8	1.1	1.3	1.5	1.9			
Class	0	0	1	1	1	1	1	0	0	0

Value				0.2	0.5	0.8	1.1	1.3	1.5	1.9
Class	0	0	0	0	0	1	1	1	1	1

Best split, default direction = left

LightGBM





4. Boosting

1.	Adaptive Boosting	Ada Boost
2.	Gradient Boosting Machine	GBM
3.	Extreme Gradient Boosting Machine	XG Boost
4.	Light Gradient Boositng Machine	Light GBM
5.	Categorical Boosting Machine	CAT Boost

Well done

과제

마지막 과제이니까 꼭 해보기!

앙상블 Ensemble – YouTube

XGBoost, LightGBM – YouTube

위 두 영상 더보기란에 Colab 링크를 다운로드 받아서 완성하고
두 개의 .ipynb파일 제출하기.
모르는 내용은 각주 및 설명 작성해오기!

Reference

Ensemble Learning : Voting and Bagging

[바람돌이/머신러닝] 앙상블(Ensemble Learning)(3) - 스태킹(Stacking) 이론 : 네이버 블로그

[핵심 머신러닝] Boosting – YouTube

KUBIG 2021-Fall ML Study

<이제윤의 ML + DL reference 북마크 모음>

https://docs.google.com/document/d/1Ssk5SsQIKaRvcNWvaLPU9mxLZ3itQTKw_X3p_PgjZew/edit?usp=sharing