DMQA open seminar

Introduction to boosting

2021. 07. 02

Data Mining & Quality Analytics Lab.

발표자: 고은지

ejkoh21@korea.ac.kr

KOREA UNIVERSITY Data Mining Quality Analytics

Contents

Introduction

❖ 발표자 소개

- 고은지
- 고려대학교 산업경영공학과
- Data Mining & Quality Analytics Lab. (김성범 교수님)
- M.S student (2021.03 ~)

✓ 관심 연구 분야

- Machine Learning and Deep Learning
- Deep Learning for signal processing and time series analysis
- Machine Learning for diagnosing liver cancer and hepatitis with health insurance data

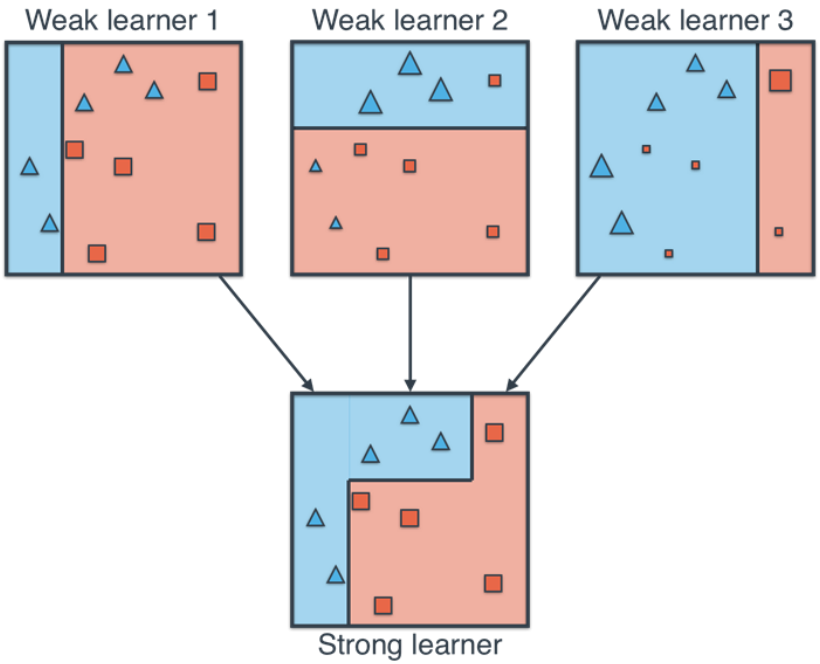
Introduction

- Introduction to boosting

Introduction

❖ Introduction to boosting

- 매우 단순한 learning 모델을 여러 개 사용하여 성능이 매우 좋은 모델 구축
- 단순한 learning 모델: 무작위 선택보다 성능이 약간 우수한 weak learner

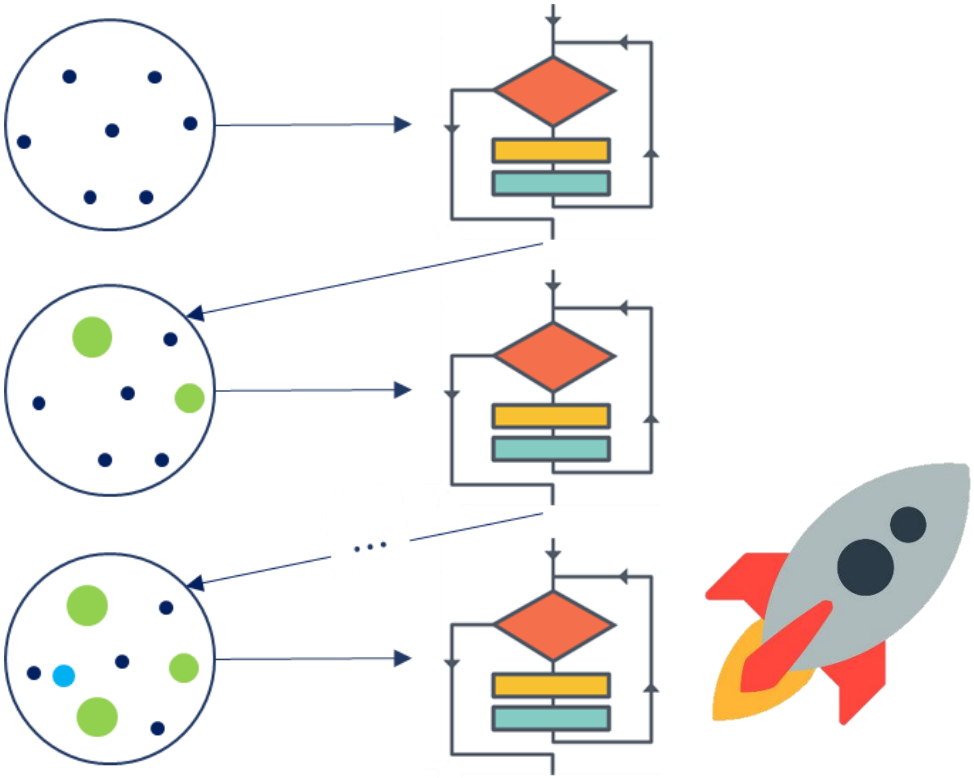


https://mathchi.medium.com/weak-learners-strong-learners-for-machine-learning-e73e32f86ebd

Introduction

❖ Introduction to boosting

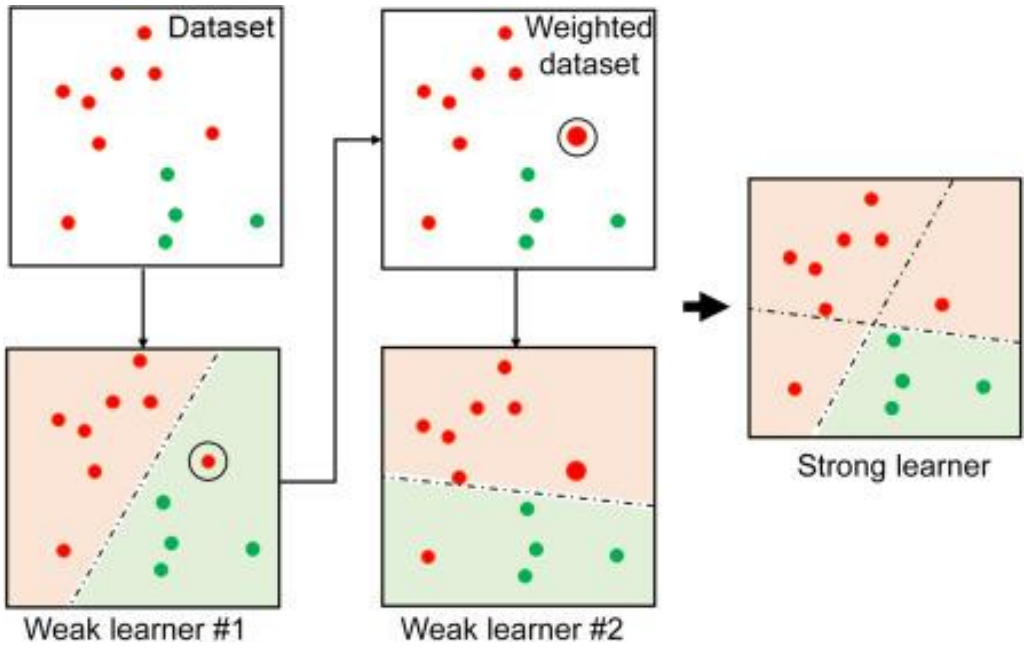
- 무작위 선택보다 약간 우수한 weak learner를 여러 개 결합하는 앙상블 방식
- 모델 구축 시 순서를 고려
- 각 단계의 weak learner는 이전 단계 weak learner의 단점을 보완



Adaboost

❖ Adaboost (adaptive boosting)

- 이전 단계 weak learner의 단점을 보완하는 새로운 weak learner를 순차적으로 구축
- 매 단계에서 모든 관측치의 Weight를 업데이트 하는 방식으로 학습

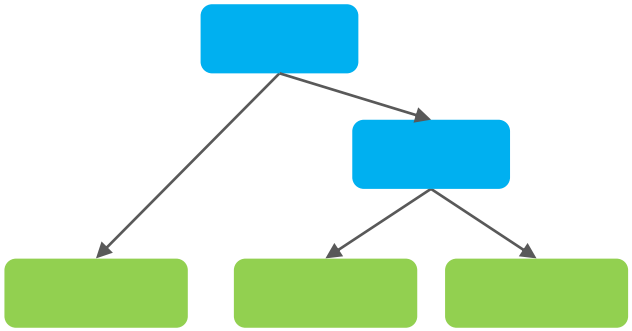


https://www.sciencedirect.com/topics/engineering/adaboost

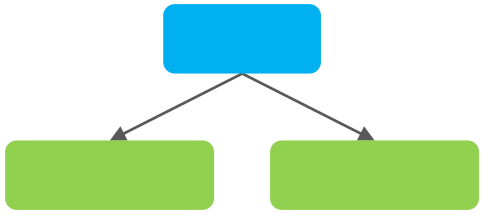
Adaboost

❖ Adaboost 기본 구조

- Weak learner: 하나의 node와 두 개의 leaf로 구성된 stump
- Random forest의 tree와 달리 하나의 stump는 하나의 변수만 사용



A tree in Random Forest



Stump

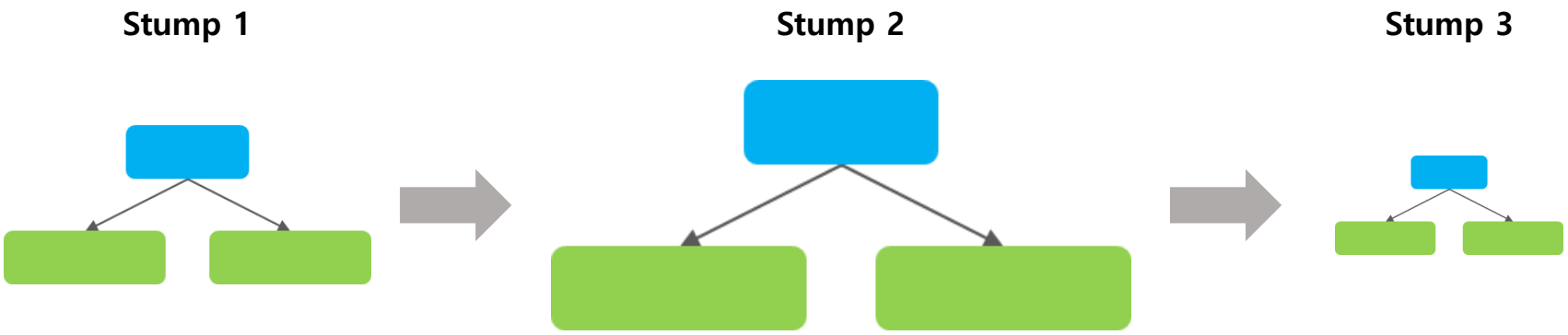
Gender	Age	Weight (kg)	Love Candy
Male	27	88	Yes
Male	44	68	No
Male	58	76	No

Gender	Age	Weight (kg)	Love Candy
Male	27	88	Yes
Male	44	68	No
Male	58	76	No

Adaboost

❖ Adaboost 기본 구조

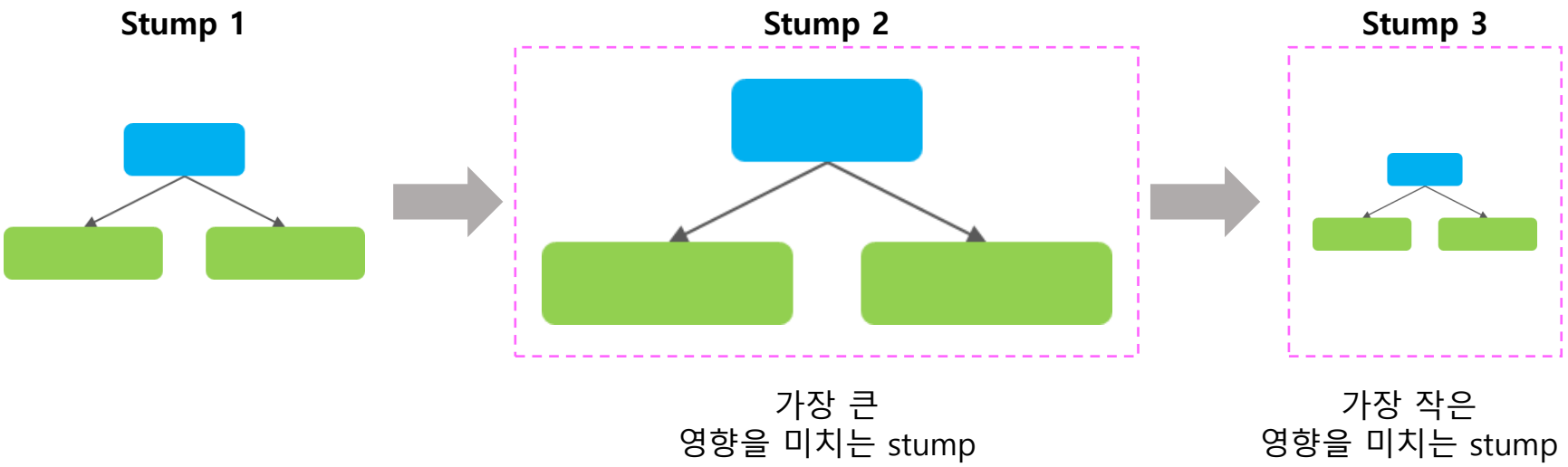
- Stump가 잘 분류/예측하지 못하는 관측치는 이후 생성될 stump에서 크게 고려함
- 순차적으로 구축된 stump가 최종적인 분류 및 예측 결과에 미치는 영향은 모두 다름



Adaboost

❖ Adaboost 기본 구조

- Stump가 잘 분류/예측하지 못하는 관측치는 이후 생성될 stump에서 크게 고려함
- 순차적으로 구축된 stump가 최종적인 분류 및 예측 결과에 미치는 영향은 모두 다름



Adaboost

❖ Selecting stump

- 초기 weight는 모든 관측치에 동일한 값 부여
- Sample Weight는 stump 구축에 사용된 각 관측치의 영향력을 의미

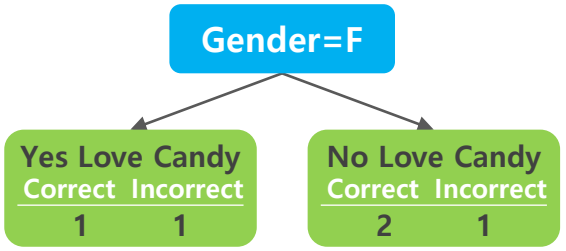
Gender	Age	Weight (kg)	Love Candy	Sample Weight
Male	27	88	Yes	1/5
Male	44	68	No	1/5
Male	58	76	No	1/5
Female	15	35	Yes	1/5
Female	25	54	No	1/5

Adaboost

❖ Selecting stump

- 초기 weight는 모든 관측치에 동일한 값 부여
- Sample Weight는 stump 구축에 사용된 각 관측치의 영향력을 의미

Gender	Age	Weight (kg)	Love Candy	Sample Weight
Male	27	88	Yes	1/5
Male	44	68	No	1/5
Male	58	76	No	1/5
Female	15	35	Yes	1/5
Female	25	54	No	1/5

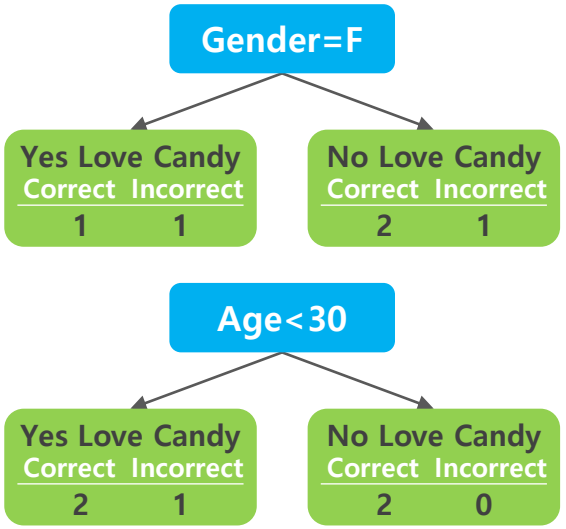


Adaboost

❖ Selecting stump

- 초기 weight는 모든 관측치에 동일한 값 부여
- Sample Weight는 stump 구축에 사용된 각 관측치의 영향력을 의미

Gender	Age	Weight (kg)	Love Candy	Sample Weight
Male	27	88	Yes	1/5
Male	44	68	No	1/5
Male	58	76	No	1/5
Female	15	35	Yes	1/5
Female	25	54	No	1/5

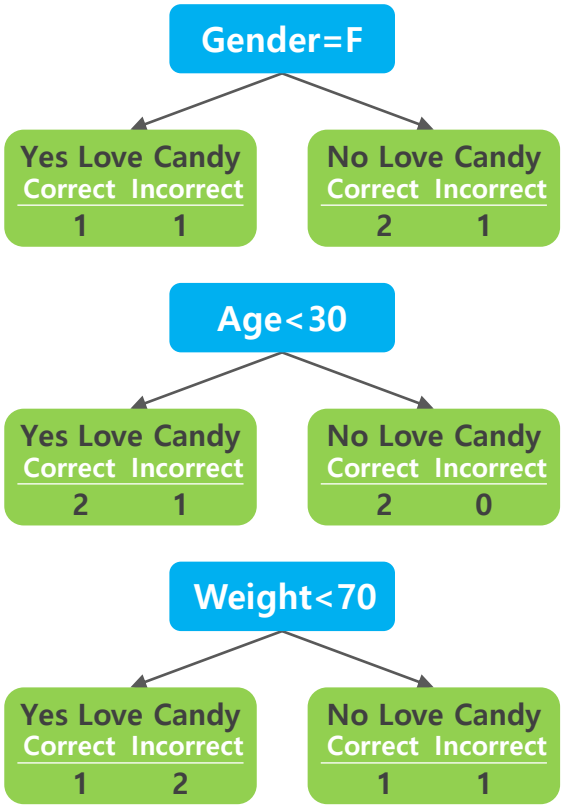


Adaboost

❖ Selecting stump

- 초기 weight는 모든 관측치에 동일한 값 부여
- Sample Weight는 stump 구축에 사용된 각 관측치의 영향력을 의미

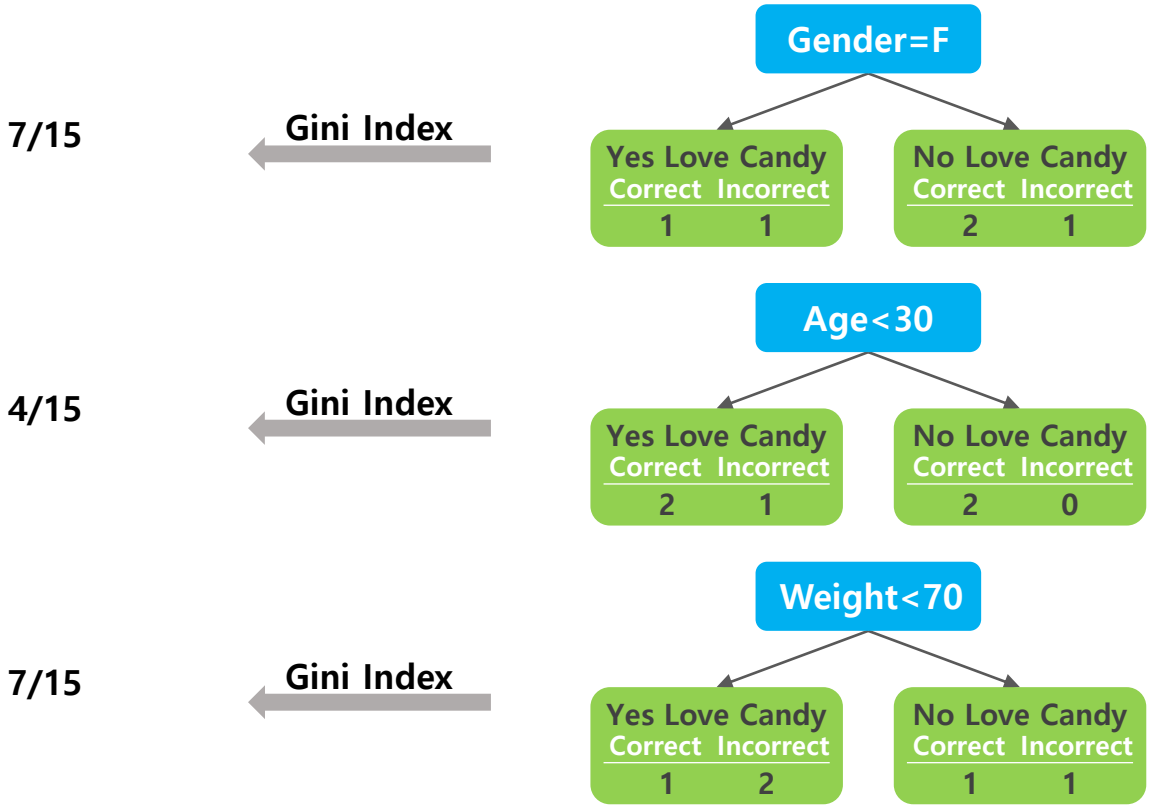
Gender	Age	Weight (kg)	Love Candy	Sample Weight
Male	27	88	Yes	1/5
Male	44	68	No	1/5
Male	58	76	No	1/5
Female	15	35	Yes	1/5
Female	25	54	No	1/5



Adaboost

❖ Selecting stump

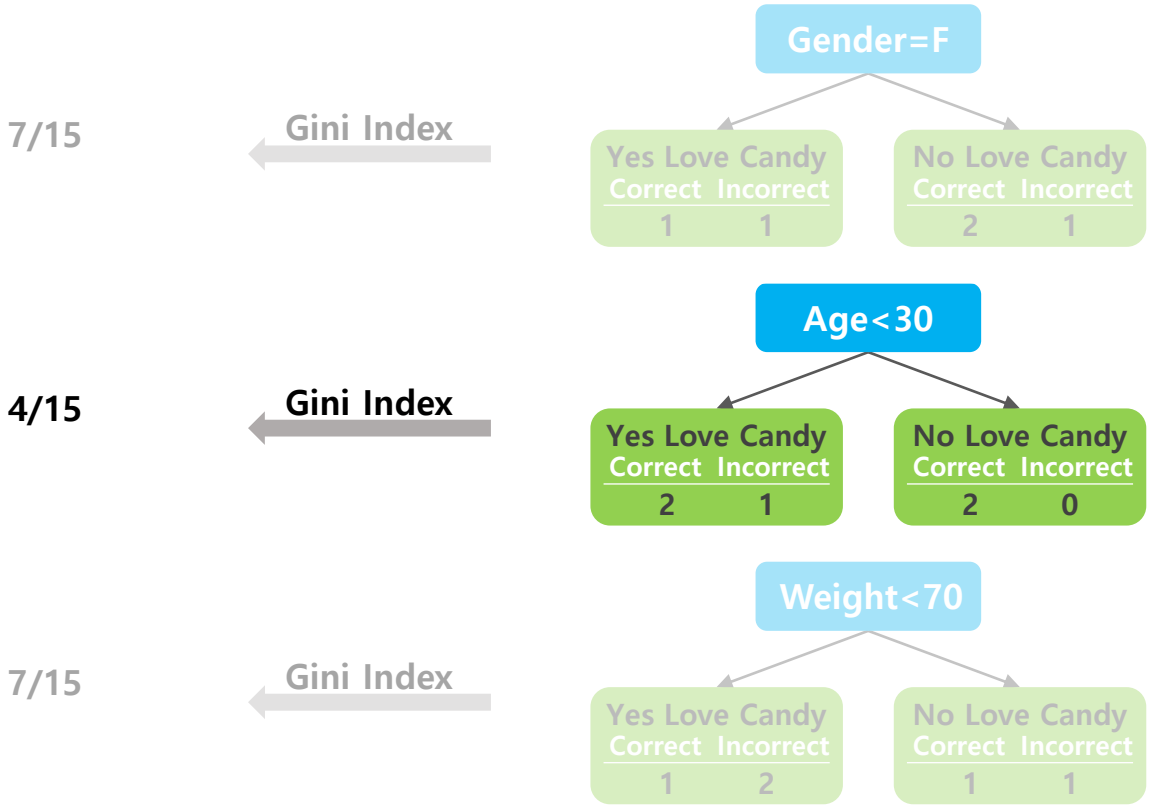
- Gini Index가 가장 작은 stump를 해당 단계의 weak learner로 사용



Adaboost

❖ Selecting stump

- Gini Index가 가장 작은 stump를 해당 단계의 weak learner로 사용

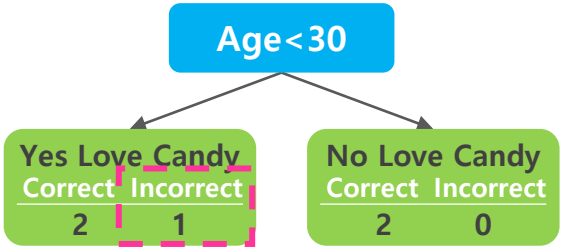


Adaboost

❖ How much say this stump

- 최종적인 분류 및 예측 결과에 대한 해당 stump의 영향력
- $\text{Amount of say} = \frac{1}{2}\log(\frac{1-Total\ Error}{Total\ Error})$

Gender	Age	Weight (kg)	Love Candy	Sample Weight
Male	27	88	Yes	1/5
Male	44	68	No	1/5
Male	58	76	No	1/5
Female	15	35	Yes	1/5
Female	25	54	No	1/5

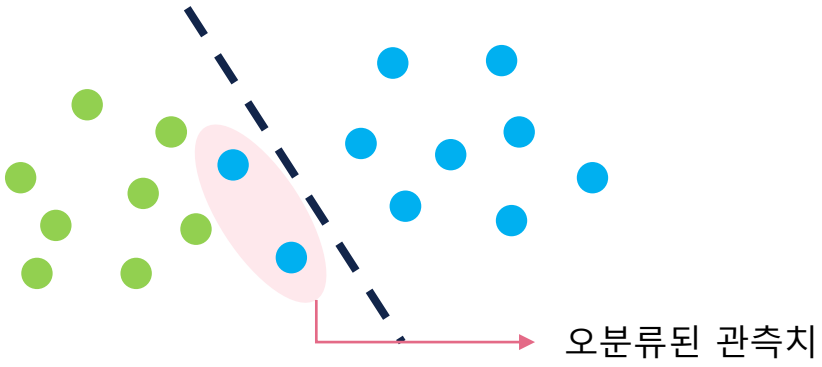


$$\text{Amount of say} = \frac{1}{2}\log\left(\frac{1-1/5}{1/5}\right) = 0.69$$

Adaboost

❖ Update sample weight

- Stump의 분류 결과를 반영하여 sample weight 업데이트
- Sample weight를 업데이트하여 이후 생성되는 stump가 오분류된 관측치에 집중하게 함



	정분류된 관측치	오분류된 관측치
Weight update 목적	Sample weight 감소	Sample weight 증가
New sample weight	$Sample\ Weight \times e^{-Amount\ of\ say}$	$Sample\ Weight \times e^{Amount\ of\ say}$

Adaboost

❖ Update sample weight

- Stump의 분류 결과를 반영하여 sample weight 업데이트
- Sample weight를 업데이트하여 이후 생성되는 stump가 오분류된 관측치에 집중하게 함

	오분류된 관측치	정분류된 관측치
New sample weight	$Sample\ Weight \times e^{Amount\ of\ say}$	$Sample\ Weight \times e^{-Amount\ of\ say}$

Gender	Age	Weight (kg)	Love Candy	Sample Weight	Sample Weight
Male	27	88	Yes	1/5	
Male	44	68	No	1/5	
Male	58	76	No	1/5	
Female	15	35	Yes	1/5	
Female	25	54	No	1/5	

Adaboost

❖ Update sample weight

- Stump의 분류 결과를 반영하여 sample weight 업데이트
- Sample weight를 업데이트하여 이후 생성되는 stump가 오분류된 관측치에 집중하게 함

	오분류된 관측치	정분류된 관측치
New sample weight	$Sample\ Weight \times e^{Amount\ of\ say}$	$Sample\ Weight \times e^{-Amount\ of\ say}$

Gender	Age	Weight (kg)	Love Candy	Sample Weight	Sample Weight
Male	27	88	Yes	1/5	
Male	44	68	No	1/5	
Male	58	76	No	1/5	
Female	15	35	Yes	1/5	
Female	25	54	No	1/5	0.4

$$\frac{1}{5} \times e^{0.69} = 0.4$$

Adaboost

❖ Update sample weight

- Stump의 분류 결과를 반영하여 sample weight 업데이트
- Sample weight를 업데이트하여 이후 생성되는 stump가 오분류된 관측치에 집중하게 함

	오분류된 관측치	정분류된 관측치
New sample weight	$Sample\ Weight \times e^{Amount\ of\ say}$	$Sample\ Weight \times e^{-Amount\ of\ say}$

Gender	Age	Weight (kg)	Love Candy	Sample Weight	Sample Weight
Male	27	88	Yes	1/5	0.1
Male	44	68	No	1/5	0.1
Male	58	76	No	1/5	0.1
Female	15	35	Yes	1/5	0.1
Female	25	54	No	1/5	0.4

$$\frac{1}{5} \times e^{-0.69} = 0.1$$

Adaboost

❖ Update sample weight

- 업데이트 된 Sample weight를 사용하여 다음 단계의 stump를 구축하기 위한 dataset 생성
- 생성된 dataset은 새로 생성될 stump가 직전 stump에서 오분류된 관측치에 더욱 집중하게 함

Gender	Age	Weight (kg)	Love Candy	Sample Weight
Male	27	88	Yes	0.1
Male	44	68	No	0.1
Male	58	76	No	0.1
Female	15	35	Yes	0.1
Female	25	54	No	0.4

기존 dataset

Gender	Age	Weight (kg)	Love Candy	Sample Weight
Female	15	35	Yes	0.1
Female	25	54	No	0.4
Male	58	76	No	0.1
Male	27	88	Yes	0.1
Female	25	54	No	0.4

새로운 dataset

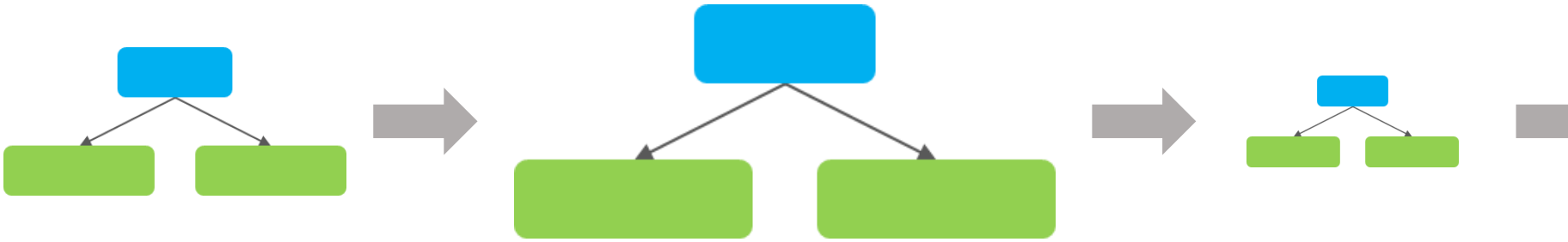
Adaboost

❖ Adaboost (adaptive boost)

- 개별 관측치의 weight와 stump의 영향력을 순차적으로 계산
- 최종적으로 amount of say의 합을 통해 classification

Gender	Age	Weight (kg)	Love Candy	Sample Weight	Sample Weight	Sample Weight
Male	27	88	Yes	1/5		
Male	44	68	No	1/5		
Male	58	76	No	1/5		

...

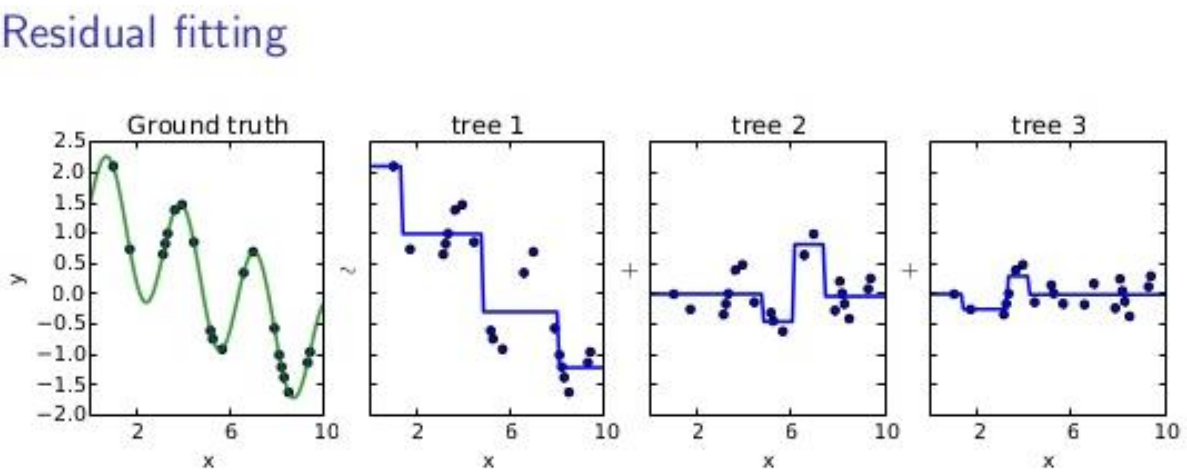


Stump 1 **Stump 2** **Stump 3**

Gradient Boosting Machine

❖ GBM (Gradient Boosting Machine)

- Single leaf로 시작하여, 이후 각 단계에서 이전 tree의 error를 반영한 새로운 tree 구축
- Tree는 이전 단계에서 발생한 residual을 예측하는 방식으로 학습



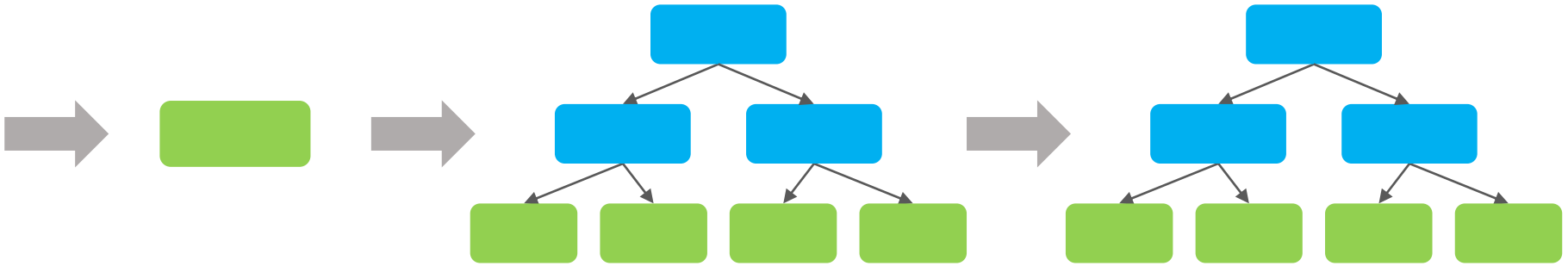
https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d

Gradient Boosting Machine

❖ GBM (Gradient Boosting Machine)

- Single leaf로 시작하여, 이후 각 단계에서 이전 tree의 error를 반영한 새로운 tree 구축
- Tree는 이전 단계에서 발생한 residual을 예측하는 방식으로 학습

Height (m)	Gender	Age	Weight (kg)
1.8	Male	27	88
1.7	Male	44	68
1.7	Male	58	76
1.5	Female	15	35



Single Leaf **Tree 1** **Tree 2**

Gradient Boosting Machine

❖ First step: Single leaf

- 첫번째 단계에서 single leaf 생성
- Regression task인 경우 평균, Classification task인 경우 log(odds) 사용
- Single leaf 값을 사용하여 첫번째 단계의 residual 계산

Height (m)	Gender	Age	Weight (kg)
1.8	Male	27	88
1.7	Male	44	68
1.7	Male	58	76
1.5	Female	15	35
1.6	Female	25	54

Average Weight

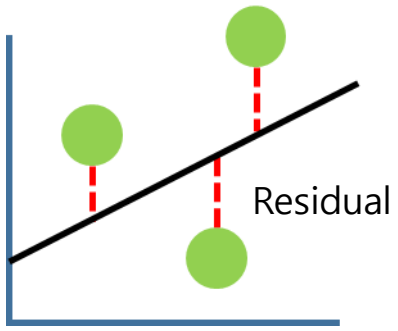
64.2

Gradient Boosting Machine

❖ Calculating Residual

- 첫번째 단계에서 single leaf 생성
- Regression task인 경우 평균, Classification task인 경우 log(odds) 사용
- Single leaf 값을 사용하여 첫번째 단계의 residual 계산

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	
1.7	Male	44	68	
1.7	Male	58	76	
1.5	Female	15	35	
1.6	Female	25	54	



Residual = 실제값 – 예측값

Gradient Boosting Machine

❖ Calculating Residual

- 첫번째 단계에서 single leaf를 생성
- Regression task인 경우 평균, Classification task인 경우 log(odds)를 사용
- Single leaf 값을 사용하여 첫번째 단계의 residual 계산

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	23.8
1.7	Male	44	68	3.8
1.7	Male	58	76	11.8
1.5	Female	15	35	-29.2
1.6	Female	25	54	-10.2

Average Weight

64.2

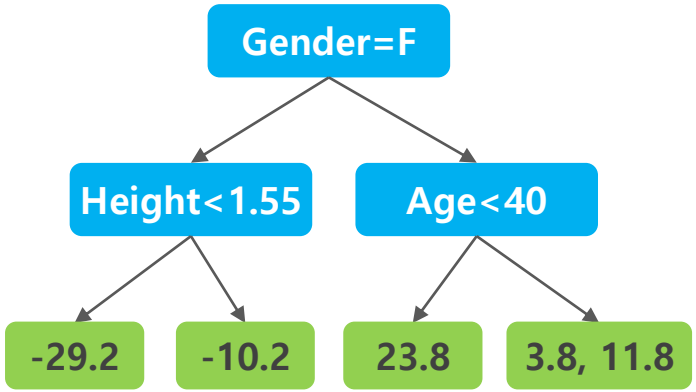
$$76 - 64.2 = 11.8$$

Gradient Boosting Machine

❖ Making Tree

- 순차적으로 size가 고정된 tree 생성
- 이전 단계에서 구한 residual을 사용하여 해당 단계의 tree를 학습

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	23.8
1.7	Male	44	68	3.8
1.7	Male	58	76	11.8
1.5	Female	15	35	-29.2
1.6	Female	25	54	-10.2

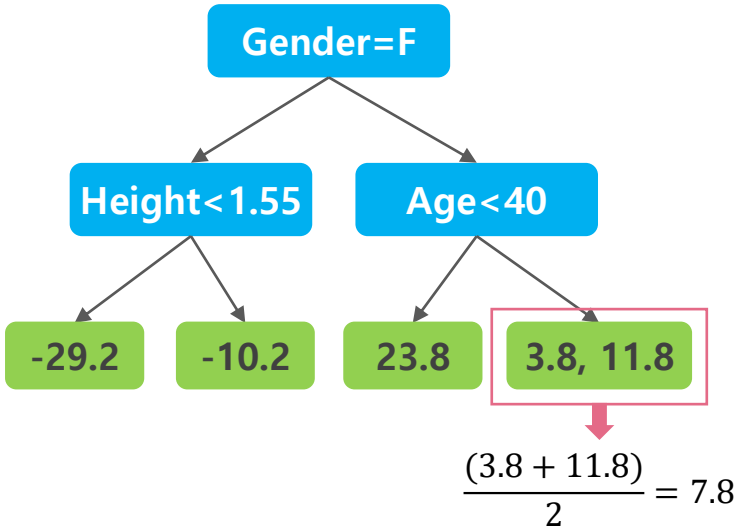


Gradient Boosting Machine

❖ Making Tree

- 순차적으로 size가 고정된 tree 생성
- 이전 단계에서 구한 residual을 사용하여 해당 단계의 tree를 학습

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	23.8
1.7	Male	44	68	3.8
1.7	Male	58	76	11.8
1.5	Female	15	35	-29.2
1.6	Female	25	54	-10.2

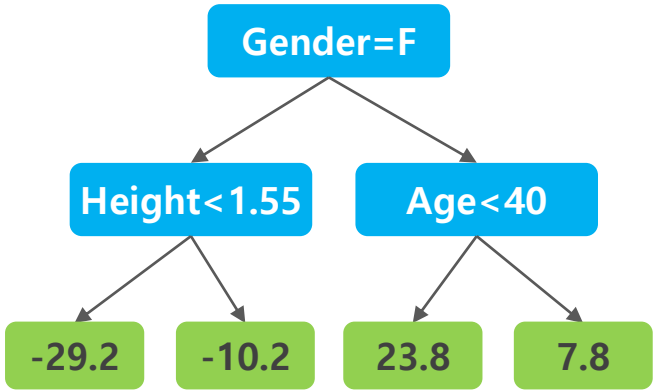


Gradient Boosting Machine

❖ Making Tree

- 순차적으로 size가 고정된 tree 생성
- 이전 단계에서 구한 residual을 사용하여 해당 단계의 tree를 학습

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	23.8
1.7	Male	44	68	3.8
1.7	Male	58	76	11.8
1.5	Female	15	35	-29.2
1.6	Female	25	54	-10.2

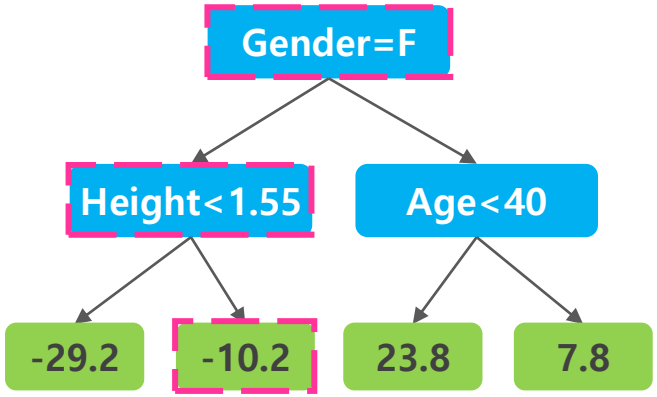


Gradient Boosting Machine

❖ Making Tree

- 순차적으로 size가 고정된 tree 생성
- 이전 단계에서 구한 residual을 사용하여 해당 단계의 tree를 학습

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	23.8
1.7	Male	44	68	3.8
1.7	Male	58	76	11.8
1.5	Female	15	35	-29.2
1.6	Female	25	54	-10.2



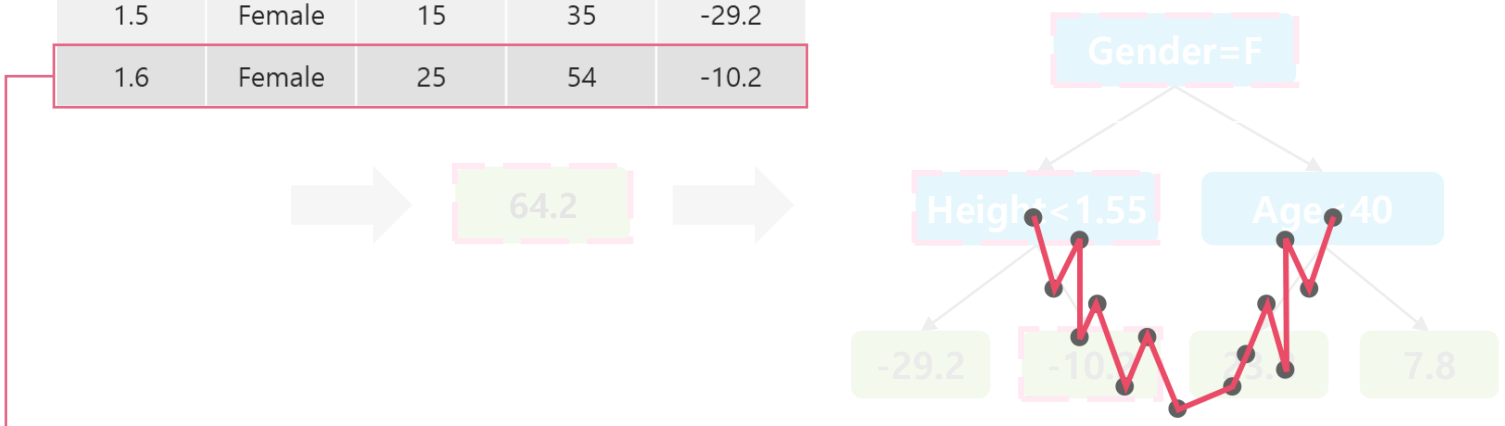
$$Predicted\ Weight = 64.2 + (-10.2) = 54$$

Gradient Boosting Machine

❖ Making Tree

- 순차적으로 size가 고정된 tree 생성
- 이전 단계에서 구한 residual을 사용하여 해당 단계의 tree를 학습

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	23.8
1.7	Male	44	68	3.8
1.7	Male	58	76	11.8
1.5	Female	15	35	-29.2
1.6	Female	25	54	-10.2



$$Predicted\ Weight = 64.2 + (-10.2) = 54$$

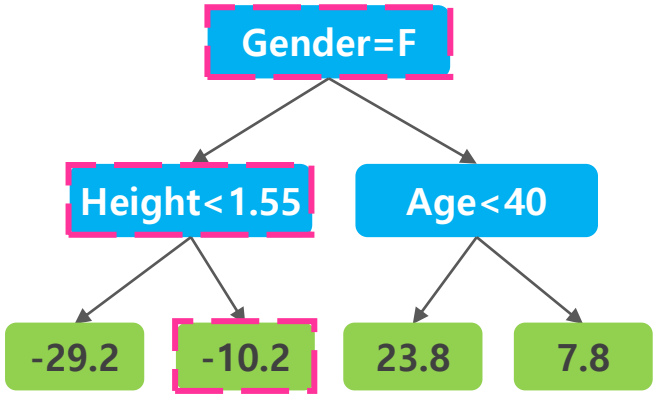
과적합 발생 가능

Gradient Boosting Machine

❖ Learning Rate

- 최종 예측 결과에 대한 해당 모델의 기여도를 scaling하여 과적합 방지 역할
- Learning rate는 0에서 1사이의 값 지정

Height (m)	Gender	Age	Weight (kg)	Residual
1.8	Male	27	88	23.8
1.7	Male	44	68	3.8
1.7	Male	58	76	11.8
1.5	Female	15	35	-29.2
1.6	Female	25	54	-10.2



$$Predicted\ Weight = 64.2 + Learning\ Rate \times (-10.2) = 54$$

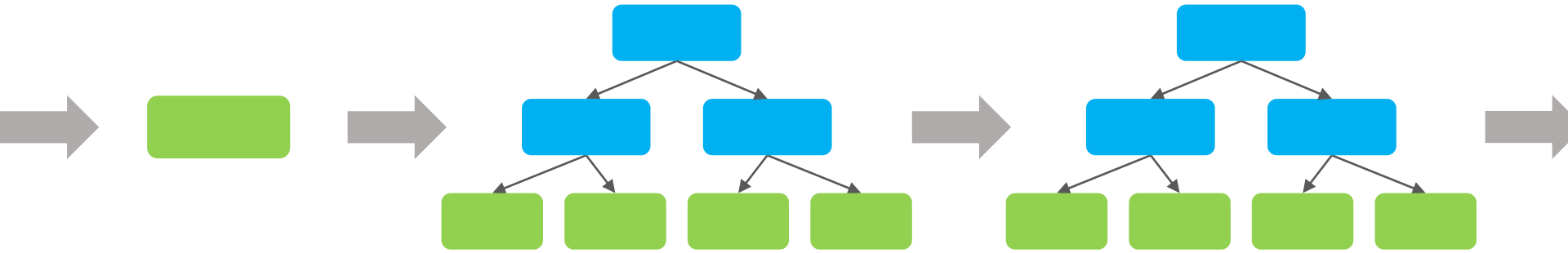
Gradient Boosting Machine

❖ Making Tree

- 새로 생성된 예측 값을 사용한 residual 계산 및 새로운 tree 구축 반복
- Residual이 더 이상 유의미하게 감소하지 않을 때까지 반복 시행

Height (m)	Gender	Age	Weight (kg)	Residual	Residual	Residual
1.8	Male	27	88			
1.7	Male	44	68			
1.7	Male	58	76			
1.5	Female	15	35			
1.6	Female	25	54			

…

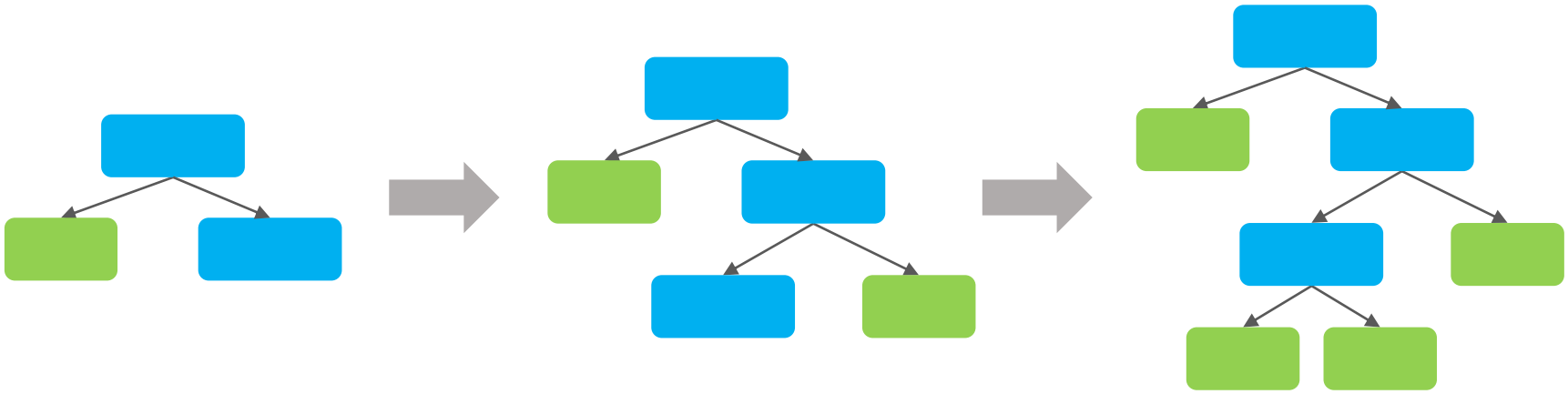


Single Leaf **Tree 1** **Tree 2**

Light Gradient Boosting Machine

❖ Light GBM (Light Gradient Boosting Machine)

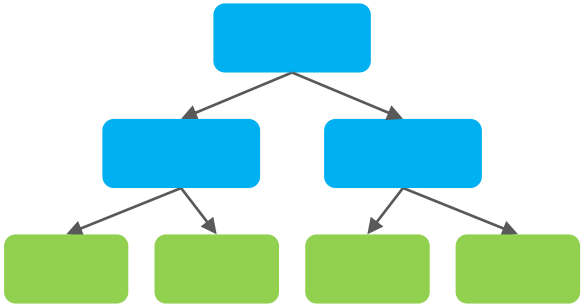
- GBM과 동일하게 순차적으로 tree를 생성하고 결합하는 방식으로 학습
- GBM 기반의 알고리즘이며, 상대적으로 학습 시간 및 메모리 사용량 단축 가능



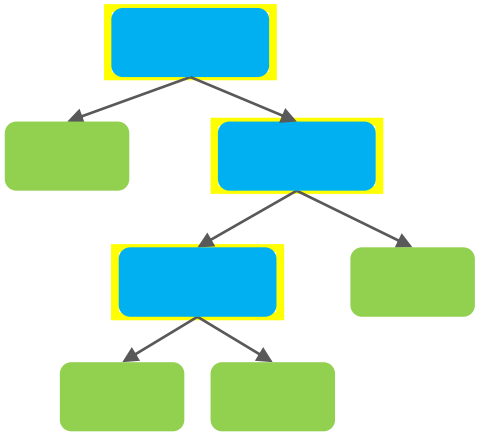
Light Gradient Boosting Machine

❖ Light GBM (Light Gradient Boosting Machine)

- Leaf-wise tree 분할 방식을 사용하여 예측 오류 손실 최소화
  - ➢ Leaf-wise tree 분할 방식: Gradient가 가장 큰 node를 순차적으로 분할



Level-wise tree growth
GBM

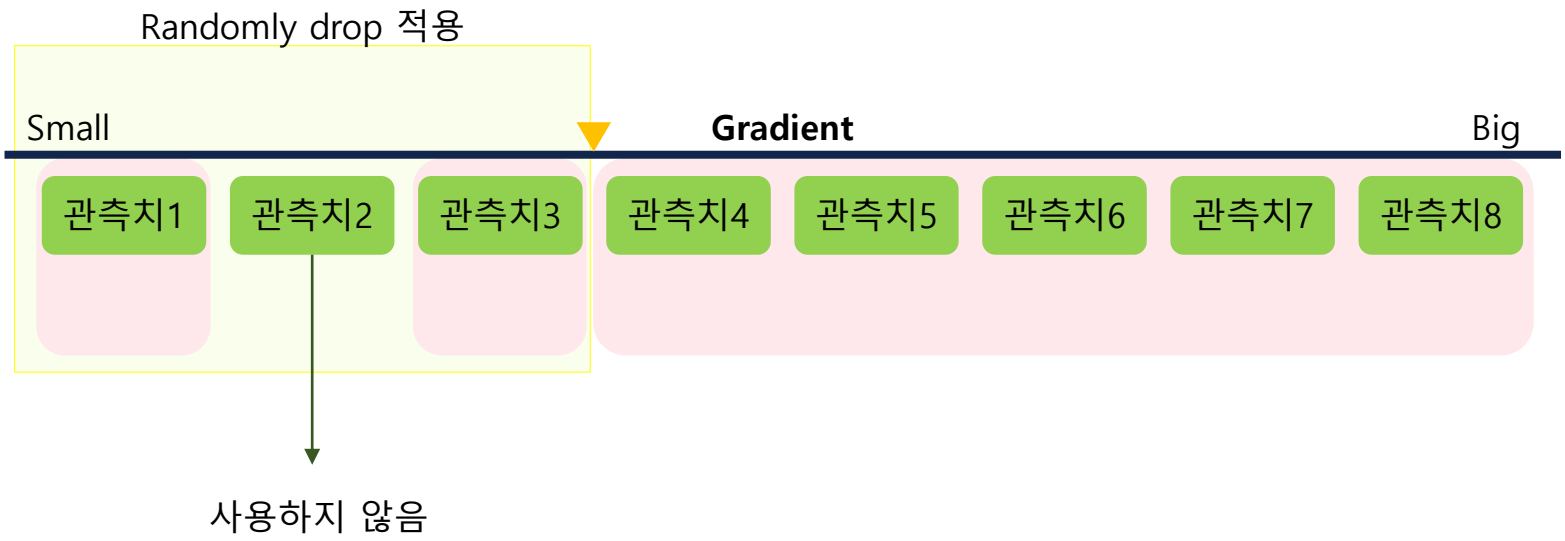


Leaf-wise tree growth
Light GBM

Light Gradient Boosting Machine

❖ Light GBM (Light Gradient Boosting Machine)

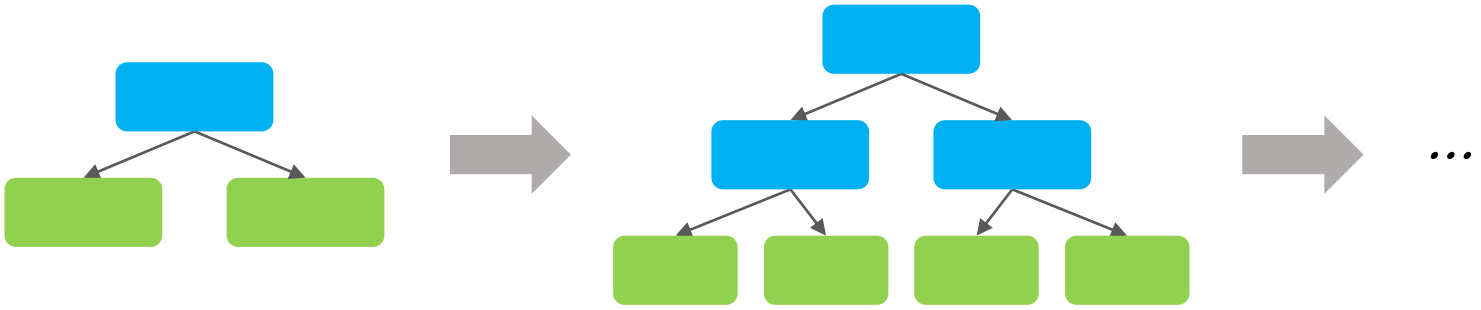
- Randomly drop을 통해 gradient가 작은 관측치 중에서는 일부만 사용
  - ➢ 학습 속도 향상 및 메모리 사용량 단축 효과



Categorical Boosting

❖ CatBoost (Categorical Boosting)

- Gradient descent 방식으로 학습하며, 범주형 변수가 많은 dataset에서 높은 성능
- Level-wise tree 분할 방식: tree의 균형을 유지하는 방식으로 분할



Tree 1 **Tree 2**

Categorical Boosting

❖ CatBoost (Categorical Boosting)

- 범주형 변수 처리 방법 개선을 통한 학습시간 단축
- 매우 많은 범주로 표현되는 변수를 처리하기 위해 목표 통계량(Target Statistic)별로 범주 그룹화
  - ➢ Target Statistic: 범주형 변수를 같은 범주에 속하는 관측치들 target 값의 statistic으로 대체

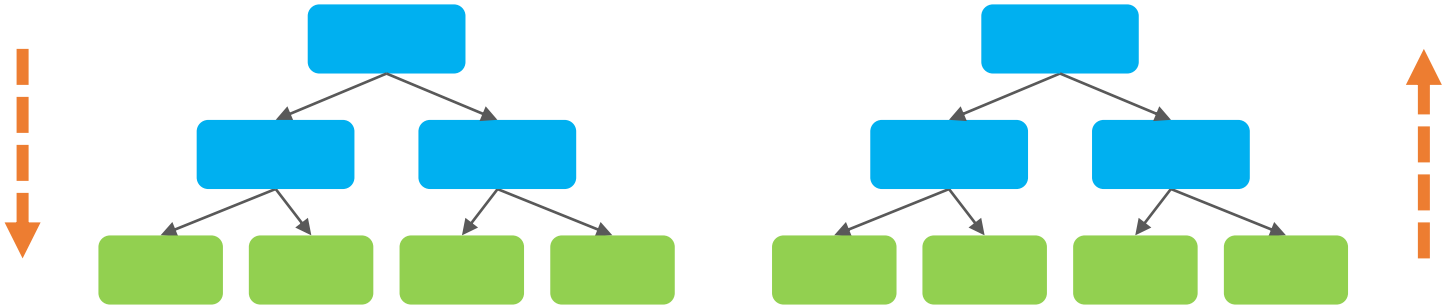
ID	Gender	Love color pink	Target
AAA	Male	Yes	10
BBB	Male	No	13
CCC	Male	Yes	17
DDD	Female	Yes	18
EEE	Female	No	13

매우 많은 범주로 표현되는 변수
(High cardinality feature)

Categorical Boosting

❖ CatBoost (Categorical Boosting)

- Tree structure를 우선 선정하고 leaf를 구하는 기존 방식과 반대로, leaf를 먼저 구한 후 tree structure를 확정하는 ordered boosting 방식



GBM CatBoost

Reference

[1] Shwartz-Ziv, Ravid, and Amitai Armon. "Tabular Data: Deep Learning is Not All You Need." arXiv preprint arXiv:2106.03253 (2021)

[2] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of computer and system sciences 55.1 (1997): 119-139.

[3] Friedman, Jerome H. "Stochastic gradient boosting." Computational statistics & data analysis 38.4 (2002): 367-378.

[4] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017): 3146-3154.

[5] Prokhorenkova, Liudmila, et al. "CatBoost: unbiased boosting with categorical features." arXiv preprint arXiv:1706.09516 (2017).

[6] YouTube lecture: https://www.youtube.com/c/joshstarmer/featured

감사합니다