

# *Ensemble* 모형 소개

# 1. Random Forest

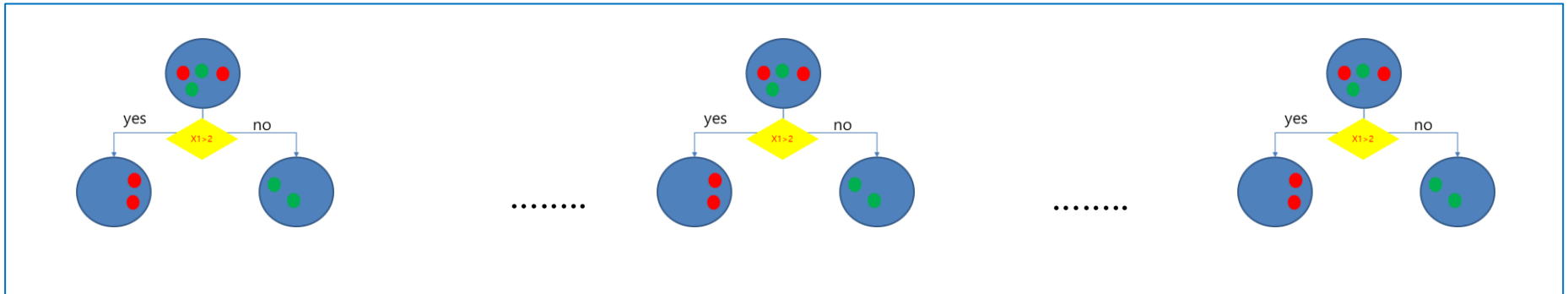
---

- Ensemble 기법: 여러 분류 모형의 결과를 결합하는 기법
- Random Forest: 앙상블 학습 방법의 일종으로, 훈련 과정에서 구성한 다수의 Decision Tree로부터 Voting을 통해 결과 예측
- Bagging: 주어진 데이터에서 랜덤하게 여러 개의 같은 크기의 부분집합을 생성
- Out of Bag과 Voting: Out of Bag(OOB)는 Bagging에서 제외되는 데이터들을 의미하며, Voting은 Random Forest내 여러 Decision Tree의 결과 중 다수의 결과를 선택하는 방법

# 1. Random Forest

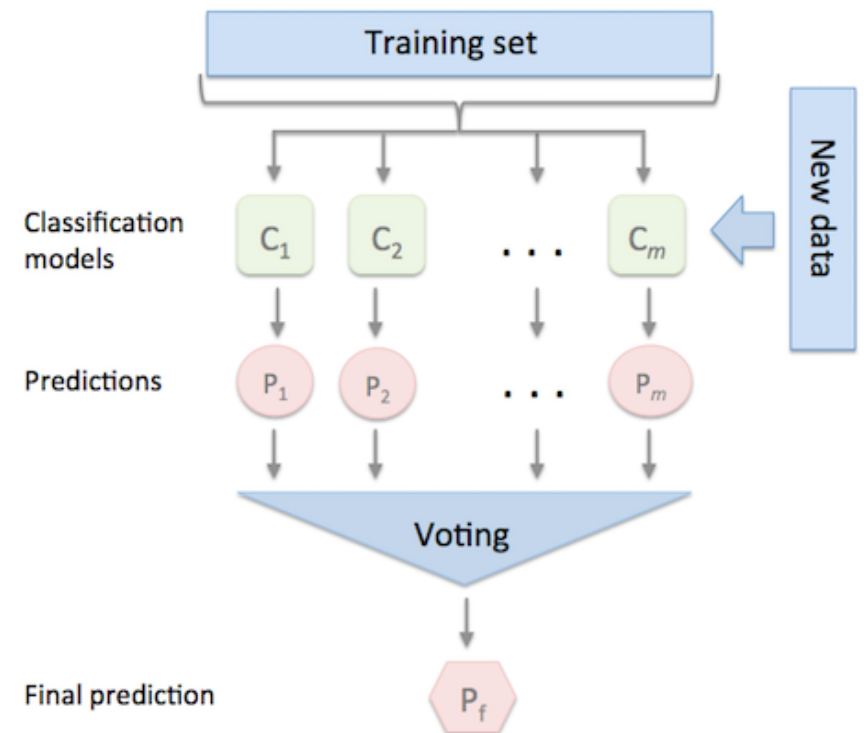
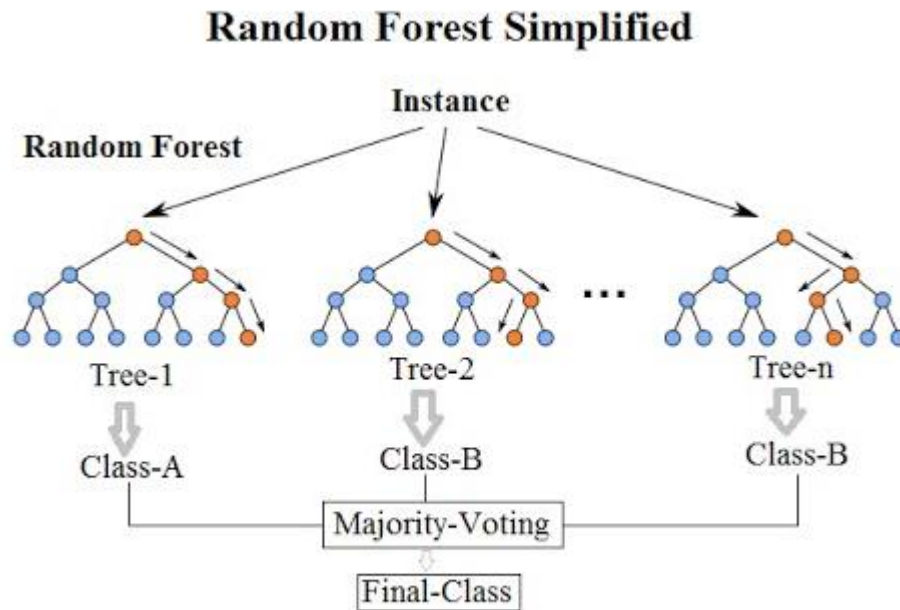
## Random Forest

- Breiman의 " bagging " 과 변수 랜덤 선택 아이디어 기반
- 처음에는 random decision forests로 시작하여 발전
- 데이터의 다양한 경우를 반영할 수 있도록 보완
- 다양한 경우에 대한 Decision Tree를 통해 성능과 안정성을 제고



# 1. Random Forest

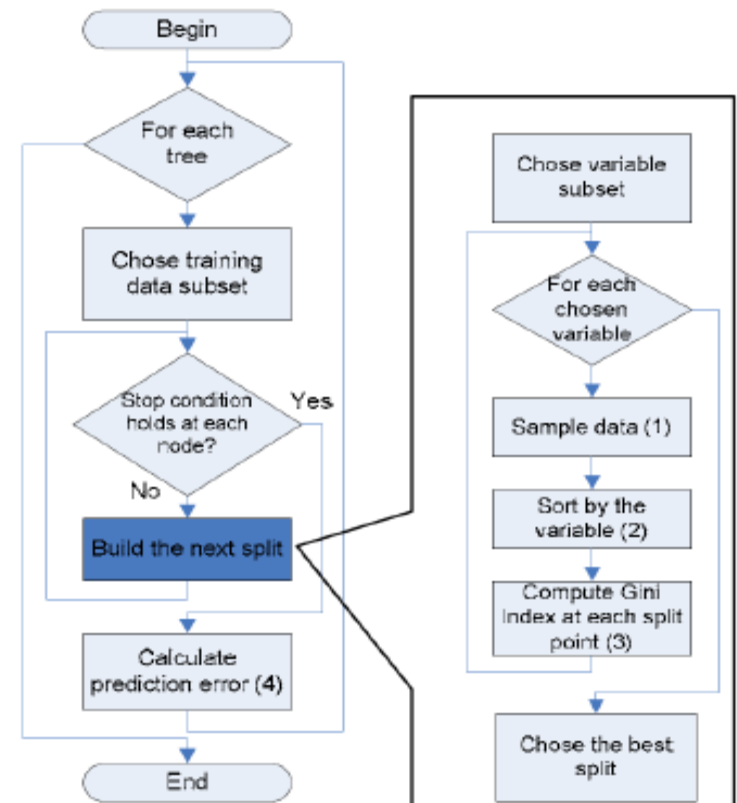
- **Random forest (or random forests)**
  - Ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees



# 1. Random Forest

- Algorithm

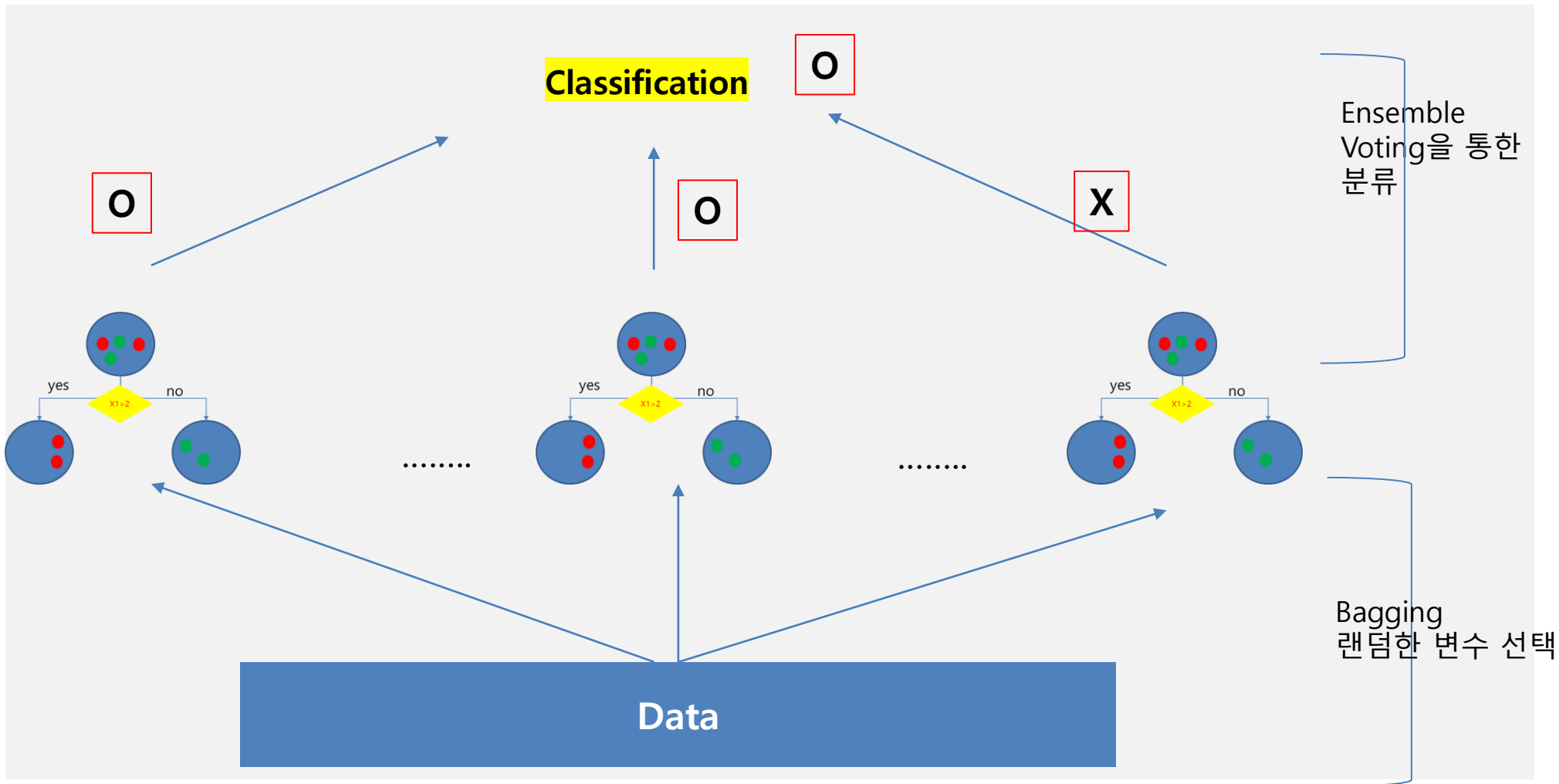
- ①  $N$ : # of training cases /  $M$ : 분류기의 변수
- ②  $M$ 개 중  $m$ 개의 변수가 Tree의 각 노드에서 분류에 사용
- ③  $N$ 개의 training case 중에서 각 tree에 사용되는  $n$ 개의 case를 선택 (예: bootstrap sample). 선택되지 않은 Case는 error 추정에 사용
- ④ 각 tree의 각 노드에서,  $m$ 개의 변수를 무작위 선택하여 분류에 사용. 이후  $m$ 개의 변수로 가장 분류를 잘하도록 계산
- ⑤ 각 Tree **fully grown and not pruned**



# 1. Random Forest

## *Random Forest*

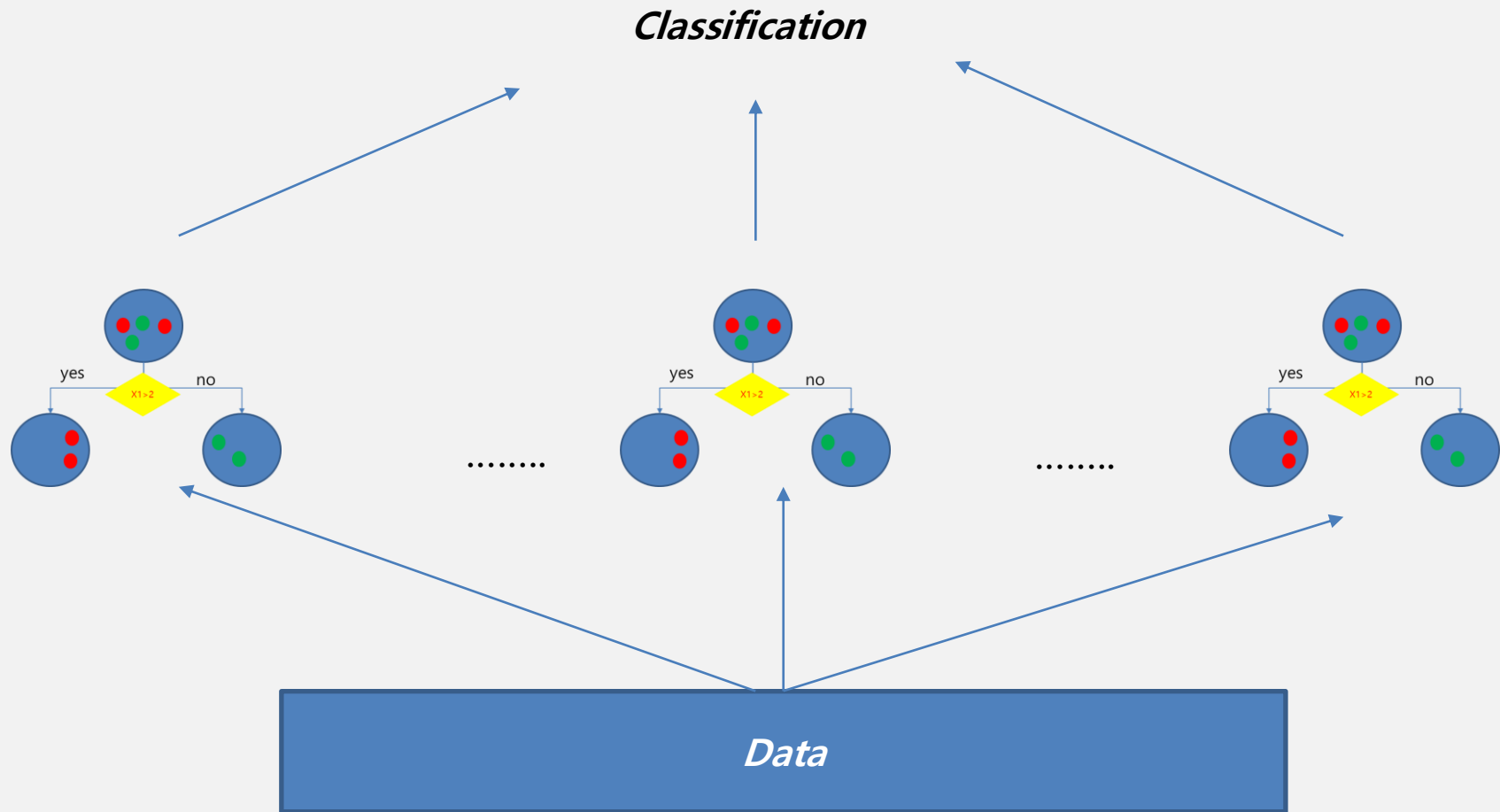
- 데이터의 다양한 경우를 반영할 수 있도록 보완
- 안정성을 제고



# 1. Random Forest

## Random Forest

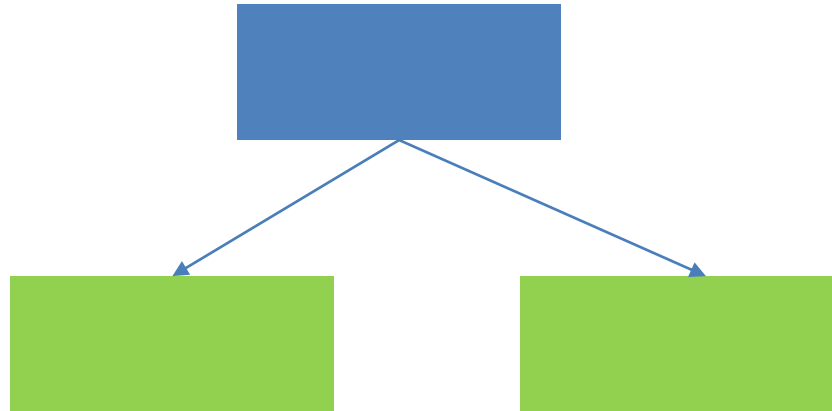
- 몇 개의 Decision Tree를 만들 것인지?
- 몇 개의 X변수를 Random하게 선택할 것인지?



## 2. Adaboost

---

- Adaboost는 Ensemble 기법의 Boosting을 DT에 적용
- Stump로 부터 학습을 시작
  - Stump: 단순한 형태의 Tree, Weak learner



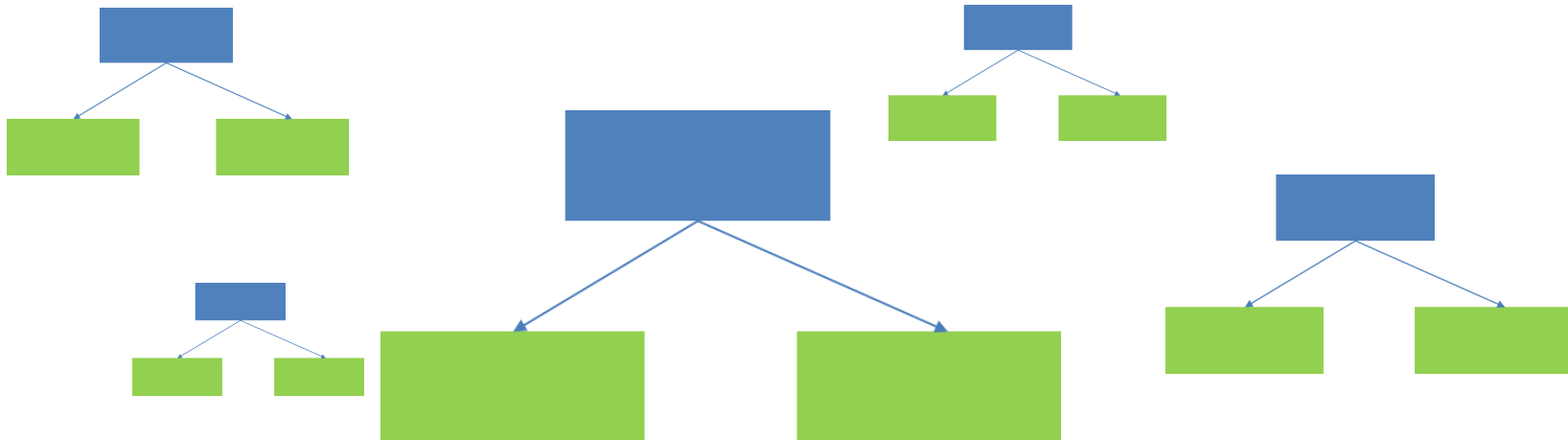


## 2. Adaboost

- Forest of stumps를 활용
  - Random Forest: 모든 tree는 같은 weight를 받음
  - Adaboost: Stump마다 중요도의 차이가 존재
- Random Forest에서는 Tree가 같은 중요도를 지님



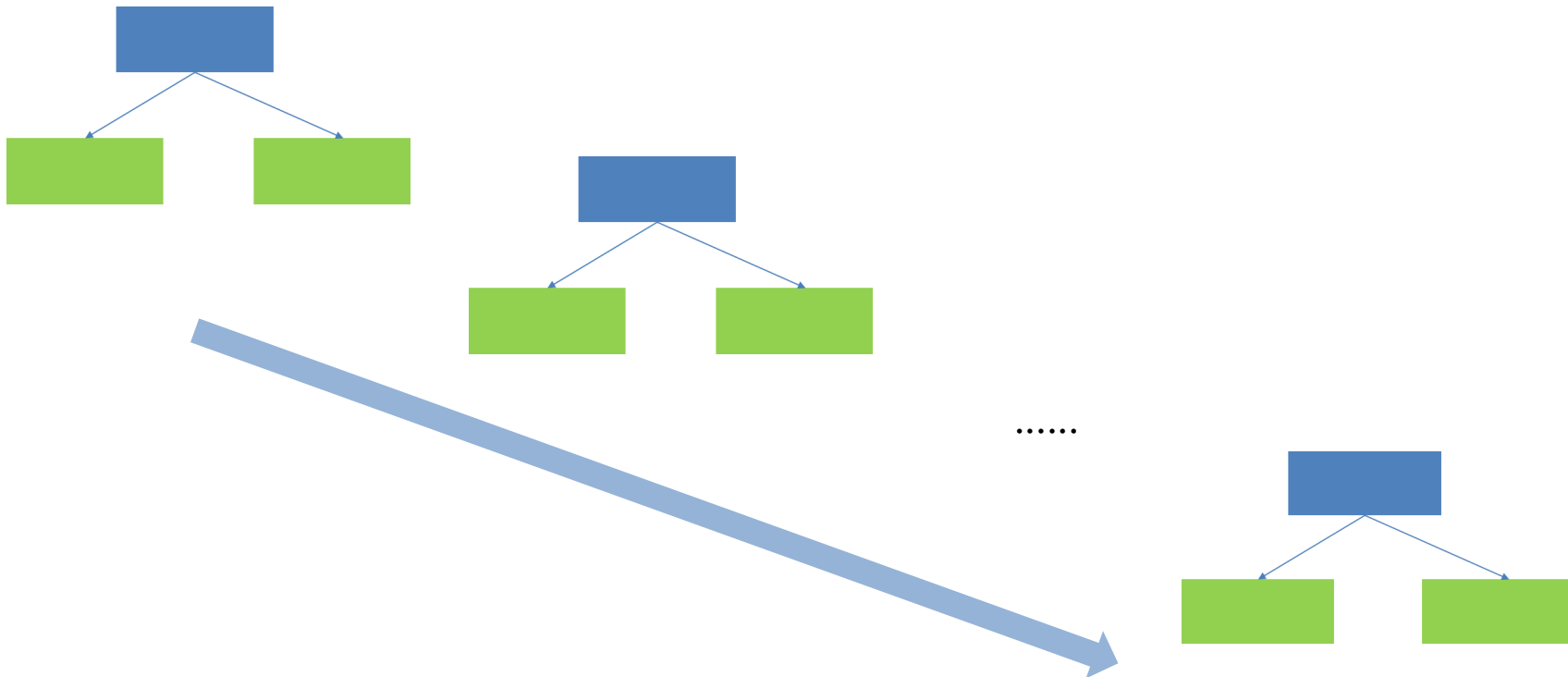
- Adaboost에서의 stump의 중요도: Amount of say로 표현, 클 수록 결과에 큰 영향을 미침



## 2. Adaboost

---

- Forest of stumps
  - 첫 stump는 다음 stump에 영향, 순차적으로 다음 stump에 영향을 주는 방식



## 2. Adaboost

- Example

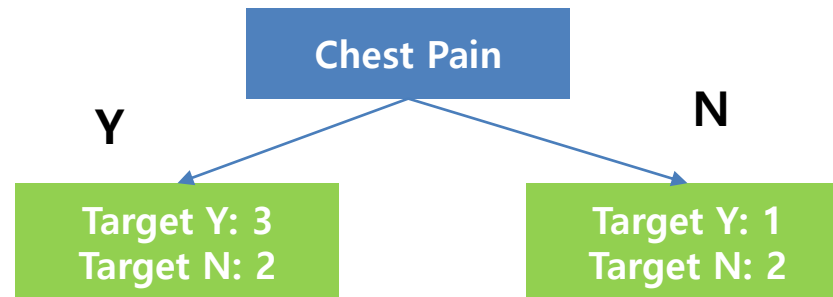
Target

Sample Weight의 합은 1

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

## 2. Adaboost

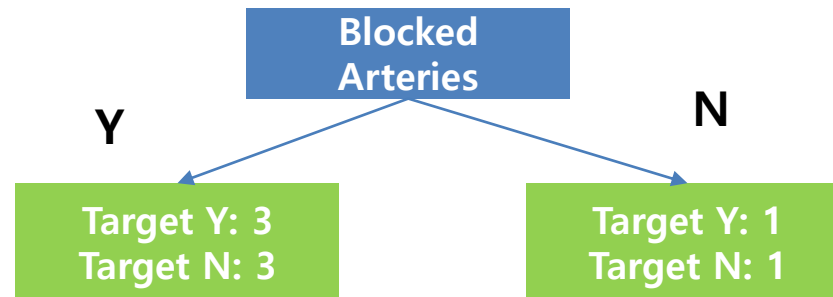
- 각 변수별 Target과의 관계



Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

## 2. Adaboost

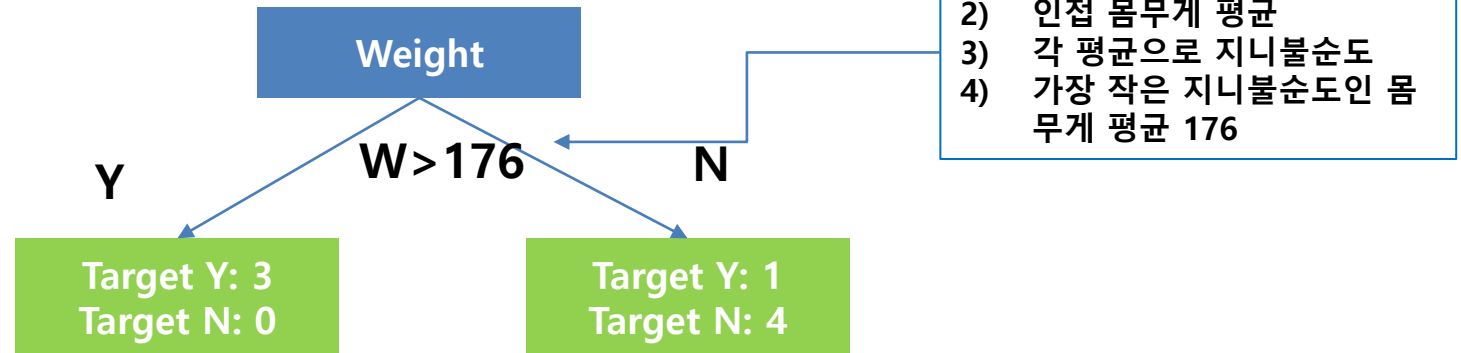
- 각 변수별 Target과의 관계



Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

## 2. Adaboost

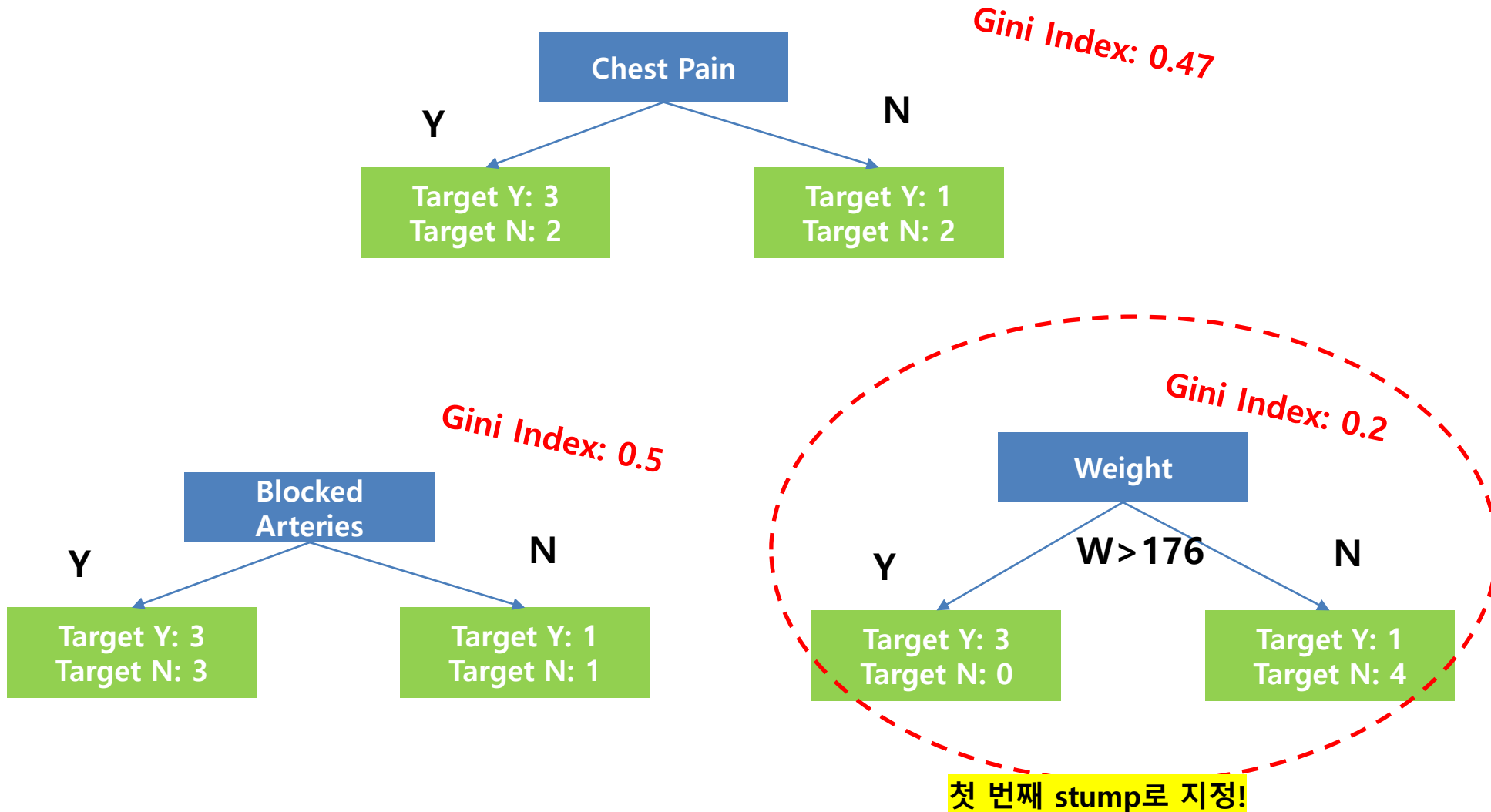
- 각 변수별 Target과의 관계



Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

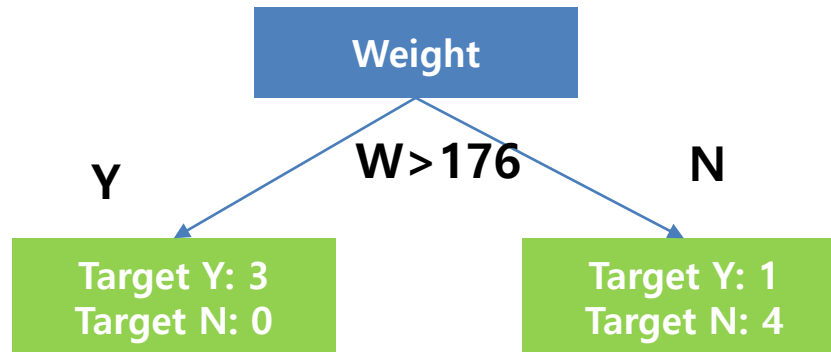
## 2. Adaboost

- 각 stump별 지니 계수



## 2. Adaboost

- 첫 stump의 Total Error



Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

첫 stump에서  
틀린 Case

Total Error = 1/8  
(0~1사이 값)

Amount of Say 계산

$$\text{Amount of Say} = \frac{1}{2} \log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

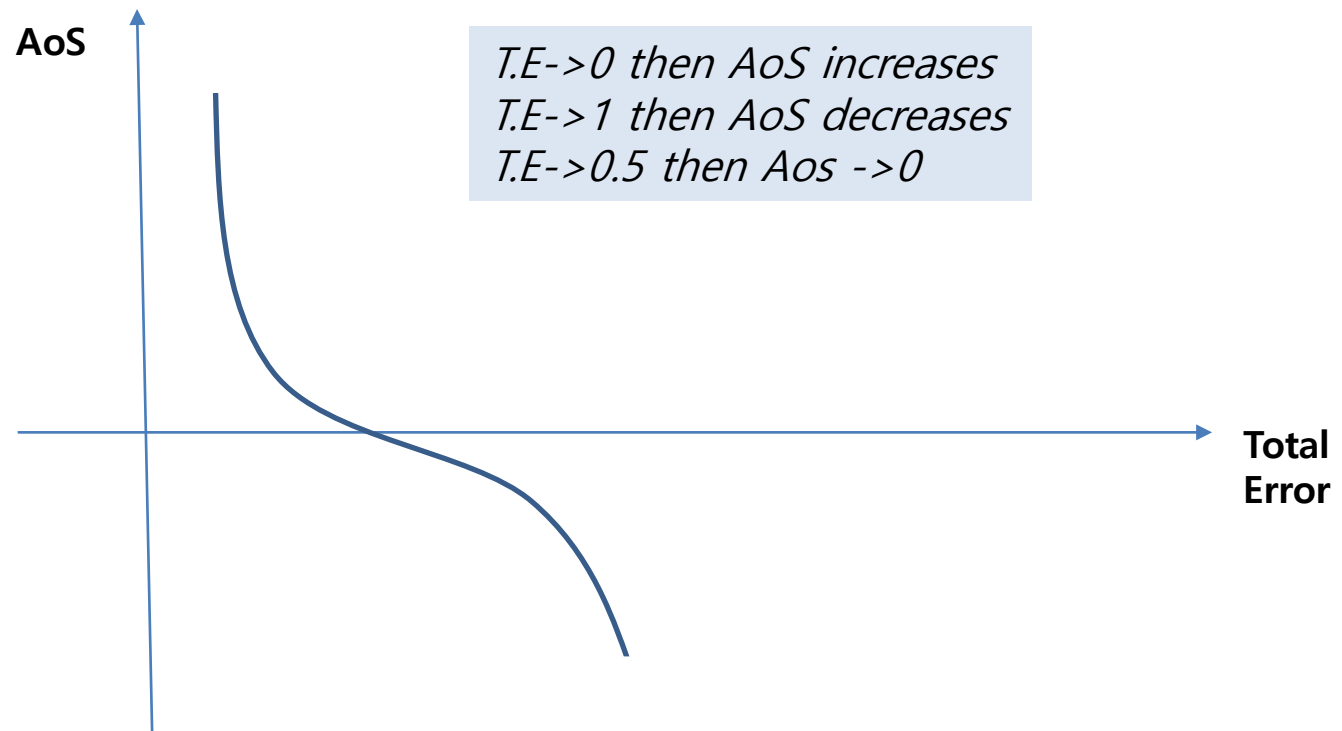
Amount of Say = 0.97



## 2. Adaboost

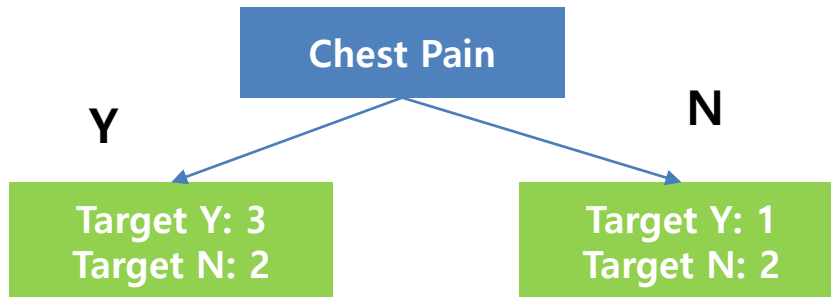
- 첫 stump의 Total Error

$$\text{Amount of Say} = \frac{1}{2} \log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$



## 2. Adaboost

- AoS 계산 예: Chest Pain (실제 stump는 아님)



Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

Amount of Say = 0.42

$$\text{Amount of Say} = \frac{1}{2} \log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

Amount of Say 계산

Total Error = 3/8  
(0~1사이 값)

*Chest Pain  
stump에서  
틀린 Case*

## 2. Adaboost

- 두 번째 Stump 계산

- 첫 Stump가 잘못 분류한 Sample의 Weight를 높여줌
  - 이후 원래 데이터에서 샘플링을 통해 새롭게 데이터 구성
  - Weight를 활용한 샘플링
- 첫 Stump에서 오분류된 Sample이 높은 Weight으로 더 많이 Sampling됨
  - 다음 Stump에서 이전 단계에 오분류된 Obs.에 집중

$$\text{New Sample Weight} = \text{Sample Weight} \times e^{\text{amount of say}}$$

AoS가 크면 Weight도 증가

- 첫 stump에서의 4번째 행

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

$$\text{New Sample Weight} = \frac{1}{8} \times e^{0.97} = 0.125 \times 2.64 = 0.33$$

## 2. Adaboost

- 두 번째 Stump 계산

- 첫 Stump에서 정분류된 Sample은 낮은 Weight으로 덜 Sampling
- 다음 Stump에서 이전 단계에 정분류된 Obs.는 덜 고려함

$$\text{New Sample Weight} = \text{Sample Weight} \times e^{-\text{amount of say}}$$

AoS가 크면 Weight는 감소

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Y	Y	205	Y	1/8
N	Y	180	Y	1/8
Y	N	210	Y	1/8
Y	Y	167	Y	1/8
N	Y	156	N	1/8
N	Y	125	N	1/8
Y	N	168	N	1/8
Y	Y	172	N	1/8

$$\text{New Sample Weight} = \frac{1}{8} \times e^{-0.97} = 0.125 \times 0.38 = 0.05$$

- Weight의 합이 1이 되도록 정규화

## 2. Adaboost

- 두 번째 Stump를 위한 Sampling

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight	Sampling을 위한 값
Y	Y	205	Y	1/8	0.07	0~0.07
N	Y	180	Y	1/8	0.07	0.07~0.14
Y	N	210	Y	1/8	0.07	0.14~0.21
Y	Y	167	Y	1/8	0.49	0.21~0.7
N	Y	156	N	1/8	0.07	0.7~0.77
N	Y	125	N	1/8	0.07	0.77~0.84
Y	N	168	N	1/8	0.07	0.84~0.91
Y	Y	172	N	1/8	0.07	0.91~1

- 0~1사이 난수 생성
- 해당 난수 값이 속하는 행을 선택
- 중복해서 선택 가능
- Weight가 높은 행이 Sampling될 확률이 높음

## 2. Adaboost

- 다음 Stump를 위한 Sampling 및 학습 반복

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight	New Weight	Sampling 을 위한 값
Y	Y	205	Y	1/8	0.07	0~0.07
N	Y	180	Y	1/8	0.07	0.07~0.14
Y	N	210	Y	1/8	0.07	0.14~0.21
Y	Y	167	Y	1/8	0.49	0.21~0.7
N	Y	156	N	1/8	0.07	0.7~0.77
N	Y	125	N	1/8	0.07	0.77~0.84
Y	N	168	N	1/8	0.07	0.84~0.91
Y	Y	172	N	1/8	0.07	0.91~0.98

- 다음 계산을 위해 가중치 초기화

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
N	Y	156	N	1/8
Y	Y	167	Y	1/8
N	Y	125	N	1/8
Y	Y	167	Y	1/8
Y	Y	167	Y	1/8
Y	Y	172	N	1/8
Y	Y	205	Y	1/8
Y	Y	167	Y	1/8

## 2. Adaboost

- Adaboost의 예측 방식

- 주어진 X값에 대해 Forest of Stump 내의 Stump에 적용

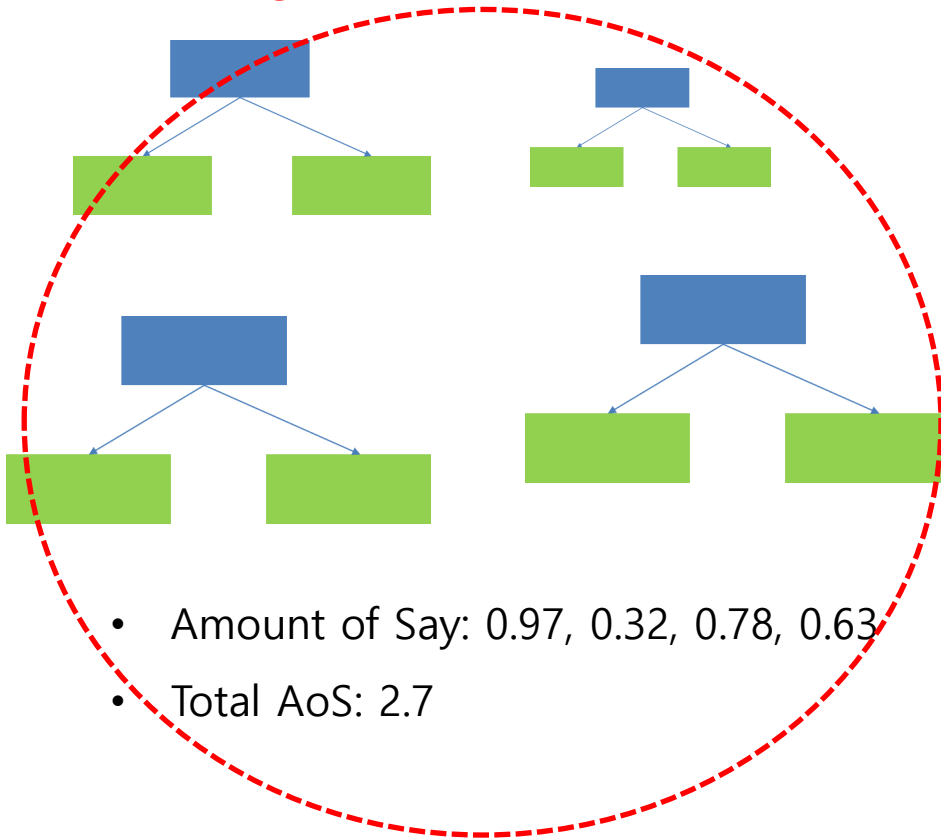


## 2. Adaboost

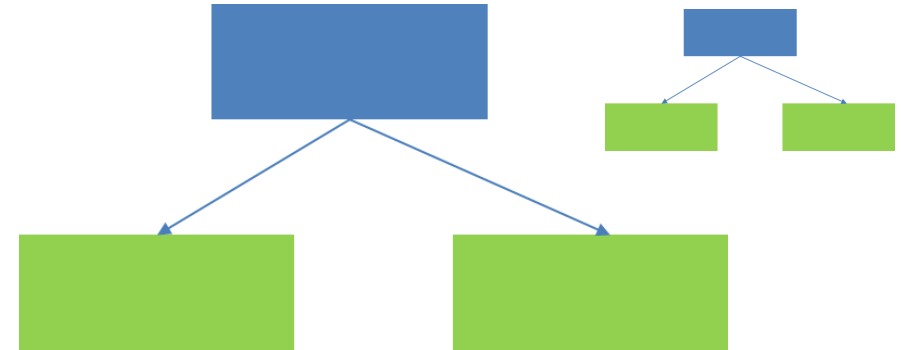
- Adaboost의 예측 방식

- 주어진 X값에 대해 Forest of Stump 내의 Stump에 적용

Target: Y로 예측



Target: N로 예측





### 3. Gradient Boost

---

- **Adaboost VS Gradient Boost**

- Adaboost: 여러 Stump의 순차적 계산
- GB: leaf로 부터 시작
  - Leaf: Target에 대한 초기 추정값(예: 평균,  $\log(\text{odds ratio})$  등)
  - Stump가 아닌 Tree를 생성: 각 tree는 leaf가 8~32개 크기 수준으로 생성

X1	X2	X3	Target
Y	12	Blue	O
Y	87	Green	O
N	44	Blue	X
Y	19	Red	X
N	32	Green	O
N	14	Blue	O

### 3. Gradient Boost

---

- **Gradient Boost for Classification, Step 1**

- Leaf의 계산

- X 범주 2개 대비 O범주는 4개, Odds = 4/2, leaf는  $\log(\text{odds}) = 0.7$

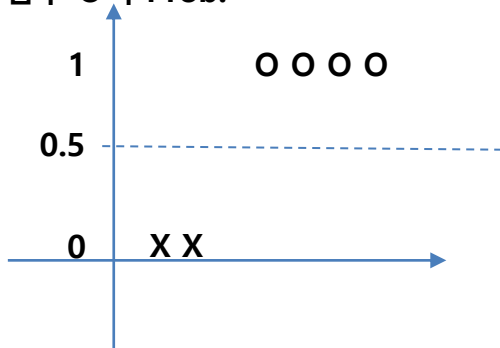
X1	X2	X3	Target
Y	12	Blue	O
Y	87	Green	O
N	44	Blue	X
Y	19	Red	X
N	32	Green	O
N	14	Blue	O

### 3. Gradient Boost

- **Gradient Boost for Classification, Step 1**

- Leaf의 계산: X 범주 2개 대비 O 범주는 4개, Odds = 4/2, leaf는  $\log(\text{odds}) = 0.7$
- Leaf를 통한 O 범주의 확률?
  - $\text{Exponential}(\log(\text{odds})) / (1 + \text{exponential}(\log(\text{odds}))) = 0.7$
  - 이 값이 기준인 0.5와 비교하여 O, X 분류
- Residual을 계산: 예를 들어 O는 확률 1이고, leaf 는 0.7이어서 Residual은 0.3

범주 O의 Prob.



같은 X변수들로  
Residual에 대한 Tree

X1	X2	X3	Target	Residual
Y	12	Blue	O	0.3
Y	87	Green	O	0.3
N	44	Blue	X	-0.7
Y	19	Red	X	-0.7
N	32	Green	O	0.3
N	14	Blue	O	0.3

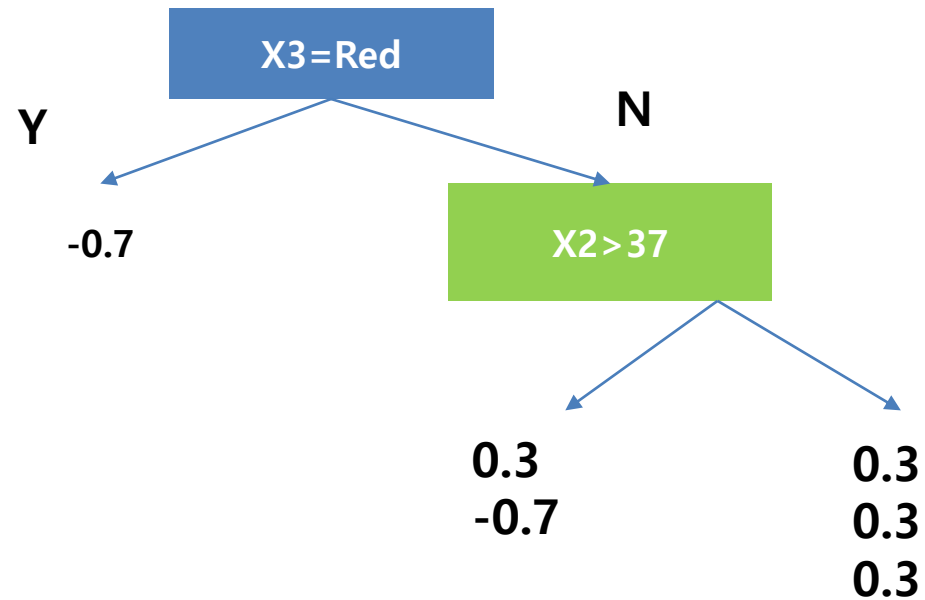
### 3. Gradient Boost

---

- Gradient Boost, Step 1

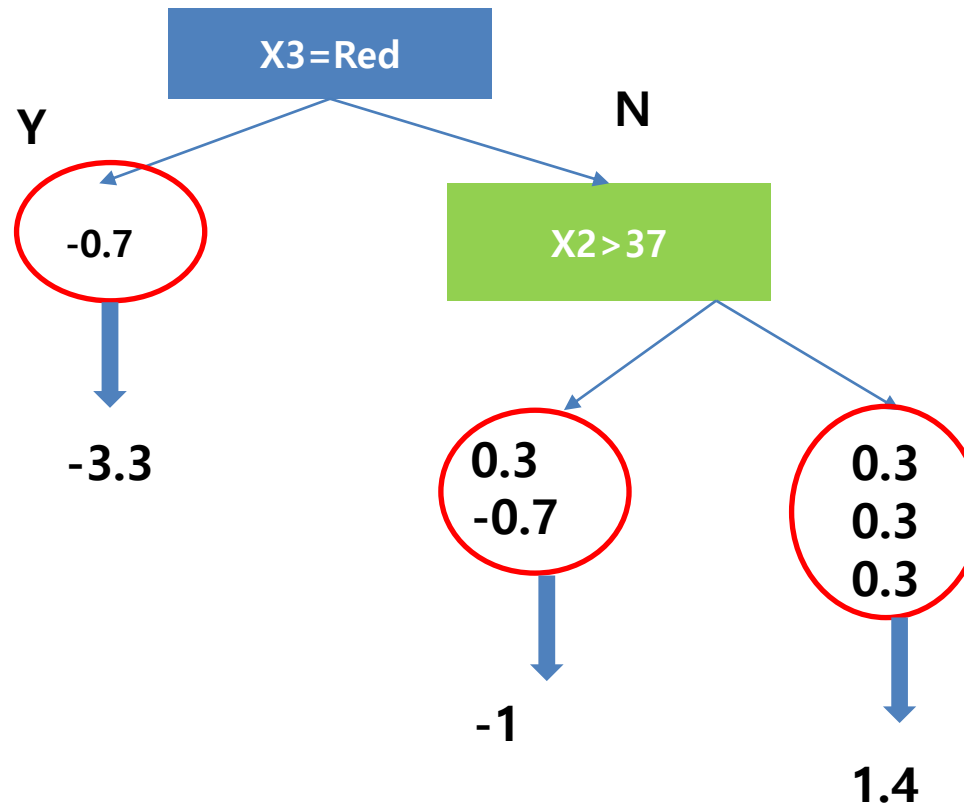
- 1st Tree**

- leaf의 수를 8~32로 제한하며 그 범위내에서 tree 생성



### 3. Gradient Boost

- Gradient Boost, Step 1
  - **1st Tree**



$$\frac{\sum \text{Residuals}}{\sum (\text{Previous Prob} \times (1 - \text{Previous Prob}))}$$

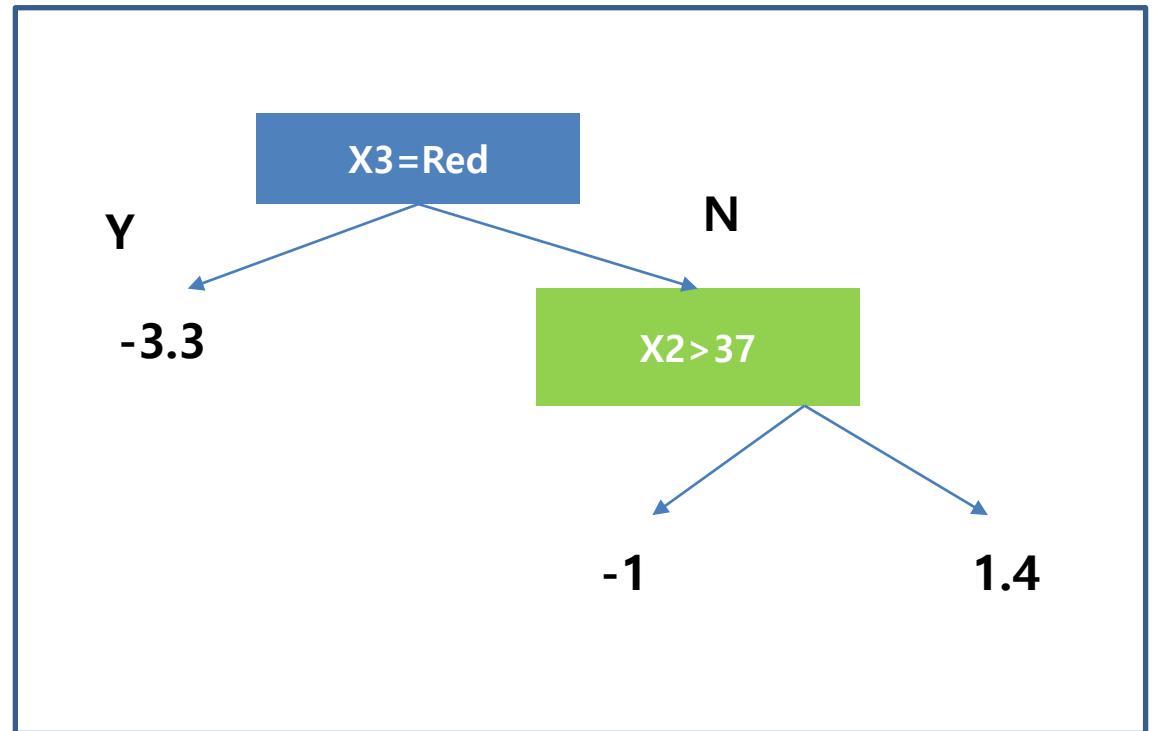
### 3. Gradient Boost

- Gradient Boost, Step 2

- Leaf 의 initial prediction에 tree에 학습을 반영하여 계산
- Leaf + 1st Tree**

A diagram illustrating the addition of a new tree's output to the initial prediction. It consists of a blue box containing the number 0.7, followed by a large yellow plus sign, and then a yellow box containing the number 0.8 followed by an 'X'.

$$0.7 + 0.8 X$$



### 3. Gradient Boost

- **Gradient Boost for Classification, Step 3**

- 각 범주에 대한 발생 확률 계산
  - 1<sup>st</sup> Obs의 업데이트된 log(odds)는 1.8
    - Leaf 0.7 + 1.4(from tree) X 0.8 = 1.8
  - 1<sup>st</sup> Obs의 확률:  $\frac{e^{1.8}}{1+e^{1.8}}$

X1	X2	X3	Target	Residual	Prob.
Y	12	Blue	O	0.3	0.9
Y	87	Green	O	0.3	0.5
N	44	Blue	X	-0.7	0.5
Y	19	Red	X	-0.7	0.1
N	32	Green	O	0.3	0.9
N	14	Blue	O	0.3	0.9

### 3. Gradient Boost

- **Gradient Boost for Classification, Step 3**

- Residual 다시 계산, 다음 tree 생성

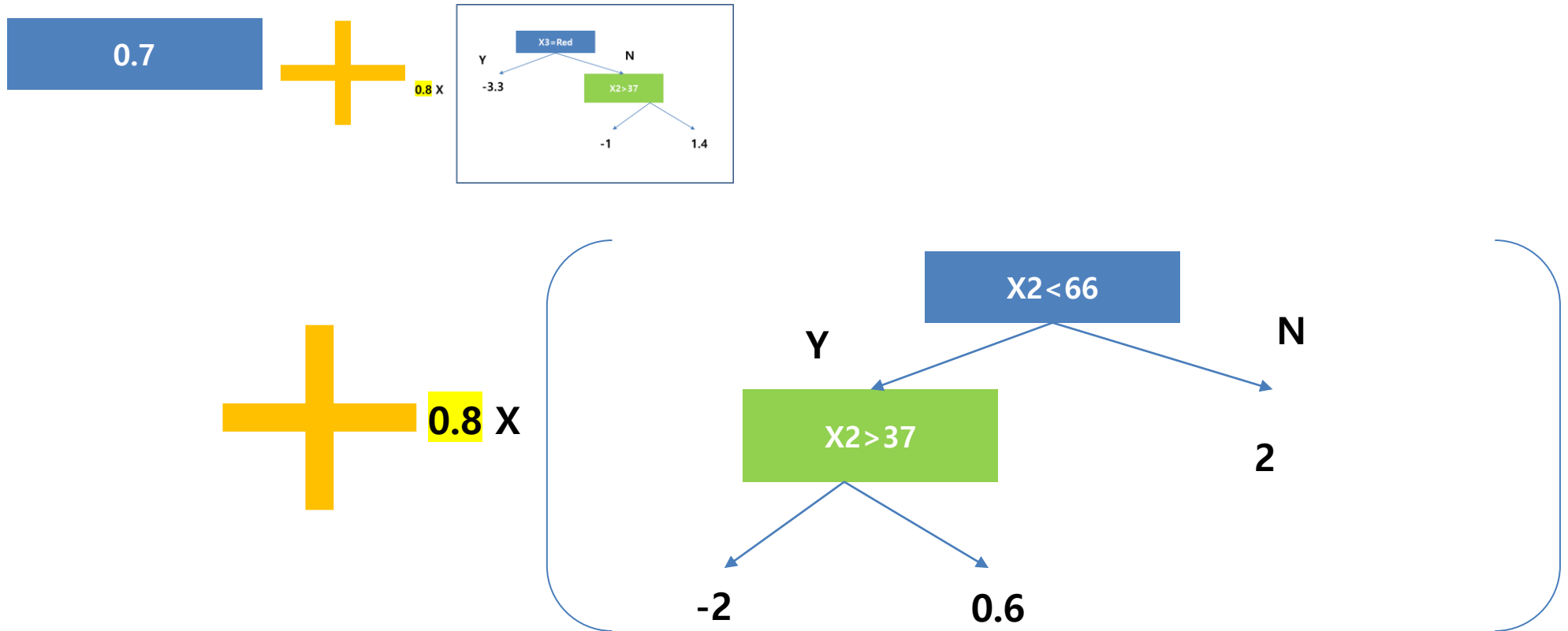
X1	X2	X3	Target	Residual	Prob.	New Residual
Y	12	Blue	O	0.3	0.9	1-0.9
Y	87	Green	O	0.3	0.5	1-0.5
N	44	Blue	X	-0.7	0.5	0-0.5
Y	19	Red	X	-0.7	0.1	0-0.1
N	32	Green	O	0.3	0.9	1-0.9
N	14	Blue	O	0.3	0.9	1-0.9



### 3. Gradient Boost

- Gradient Boost, Step 3

- Learning Rate: 0~1사이, 이 예에서는 0.1 사용



### 3. Gradient Boost

---

- **Gradient Boost for Regression**

- GB: leaf로 부터 시작
  - Leaf: Target에 대한 초기 추정값(예: 평균,  $\log(\text{odds ratio})$  등)
  - Stump가 아닌 Tree를 생성: 각 tree는 leaf가 8~32개 크기 수준으로 생성

Target			
Height	Color	Gender	Weight
1.6	B	M	88
1.6	G	F	76
1.5	B	F	56
1.8	R	M	73
1.5	G	M	77
1.4	B	F	57

### 3. Gradient Boost

- **Gradient Boost, Step 1**

- Leaf의 계산
- Target인 Weight의 평균: 71.2
- Residual을 계산: 실제값과 예측값의 차이(error)

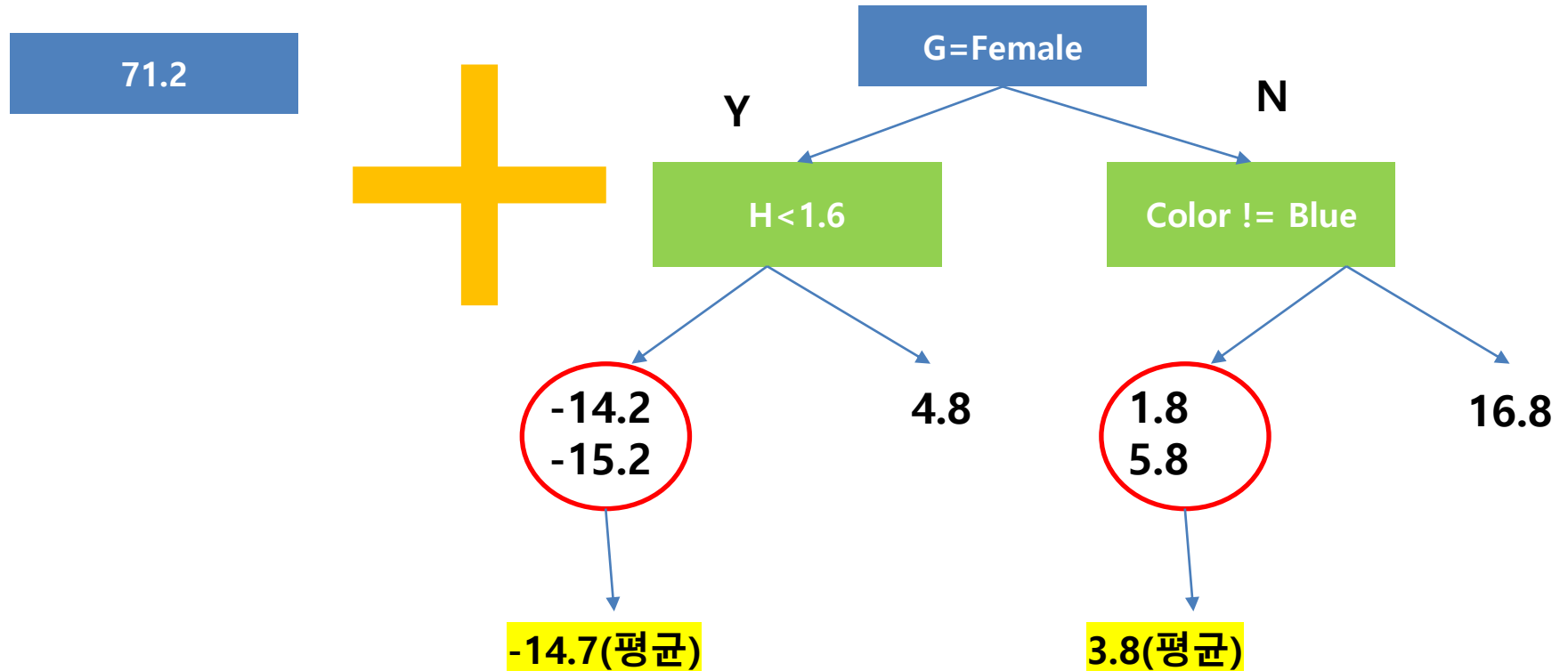
같은 X변수들로  
Residual에 대한 Tree

Height	Color	Gender	Weight	Residual
1.6	B	M	88	16.8
1.6	G	F	76	4.8
1.5	B	F	56	-15.2
1.8	R	M	73	1.8
1.5	G	M	77	5.8
1.4	B	F	57	-14.2

### 3. Gradient Boost

- Gradient Boost, Step 1

- Leaf + **1st Tree**



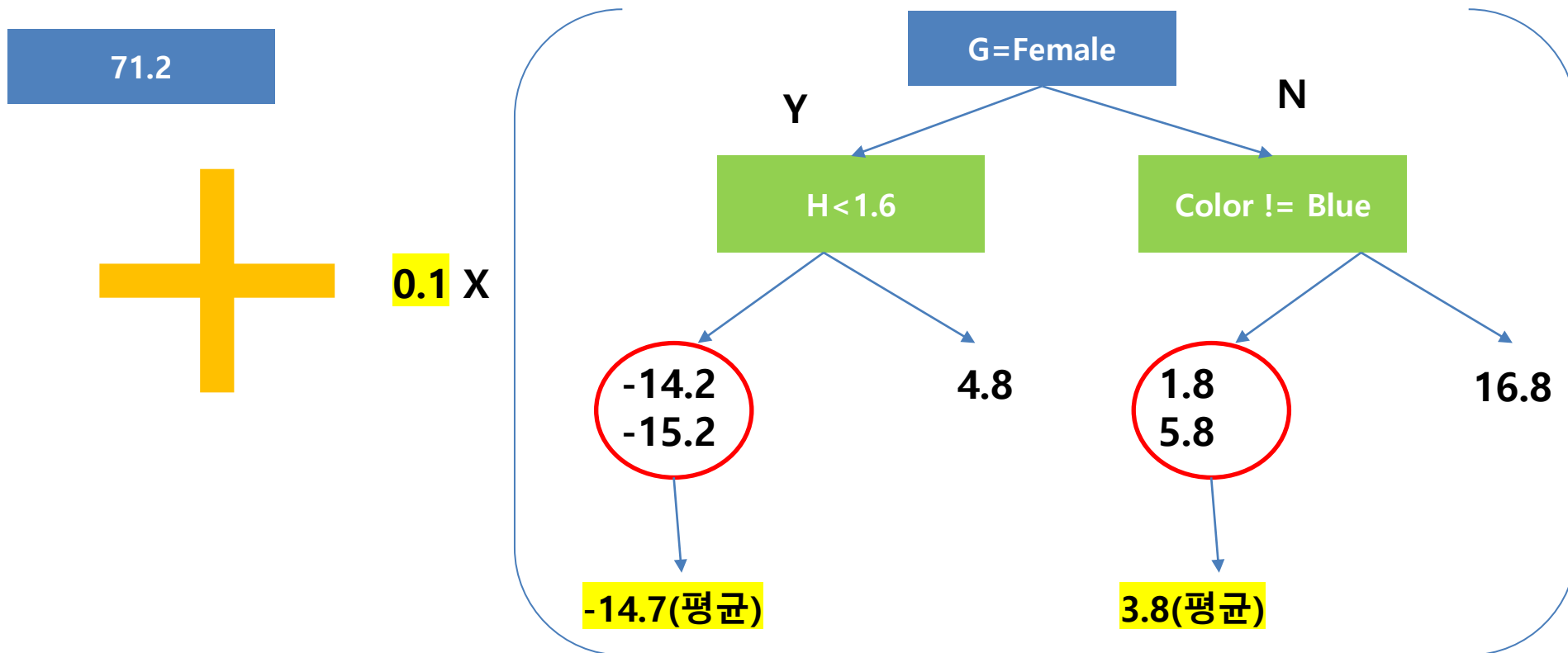
- Male, Blue인 경우 예측 예시:**

- $71.2 + 16.8 = 88$  (관측치와 동일하지만 과적합)
- Bias는 작지만 Variance 큰 상태

### 3. Gradient Boost

- Gradient Boost, Step 2

- 과적합 방지, 학습속도 조절을 위한 학습율 도입
- Learning Rate: 0~1사이, 이 예에서는 0.1 사용

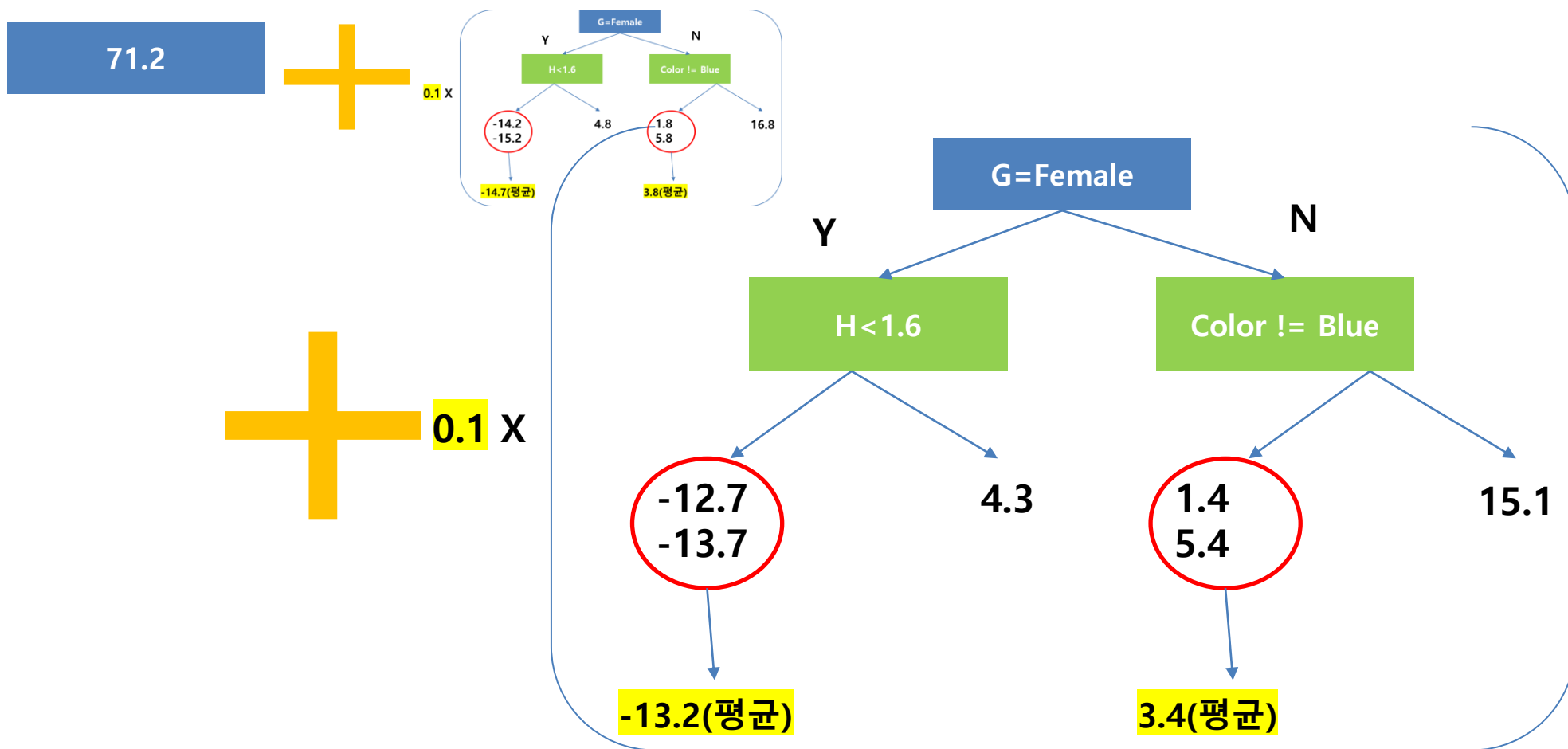


- **Male, Blue인 경우 예측 예시:**  $71.2 + 0.1 \times 16.8 = 72.9$ 
  - 실제값에 가까워지지만, 그 정도가 조절됨 (Gradient의 개념)
  - Variance를 낮게 유지할 수 있음

### 3. Gradient Boost

- Gradient Boost, Step 3

- Learning Rate: 0~1사이, 이 예에서는 0.1 사용



- H=1.6, Male, Blue인 경우 예측 예시:  $71.2 + 0.1 \times 16.8 + 0.1 \times 15.1 = 74.4$

### 3. Gradient Boost

- Gradient Boost, Step 3

- 학습을 반영 예측값을 통한 두 번째 Residual 계산

같은 X변수들로 New  
Residual에 대한 Tree

Height	Color	Gender	Weight	Residual	Residual(new)
1.6	B	M	88	16.8	15.1
1.6	G	F	76	4.8	4.3
1.5	B	F	56	-15.2	-13.7
1.8	R	M	73	1.8	1.4
1.5	G	M	77	5.8	5.4
1.4	B	F	57	-14.2	-12.7



Residual 크기 감소

### 3. Gradient Boost

---

- **Gradient Boost**
  - 위의 과정을 계속 반복
    - 정해진 iteration한도 까지 반복
    - 또는 이전 단계와 이후 단계의 Residual 차이가 없을 때까지 반복
  - 매 iteration에서의 Tree의 leaf는 8~32개 사이에서 생성
  - 매 iteration마다 다르게 생성
    - 1<sup>st</sup> tree: leaf 8개
    - 2<sup>nd</sup> tree: leaf 32개
    - 3<sup>rd</sup> tree: leaf 16개
    - ...



---

# Q&A