



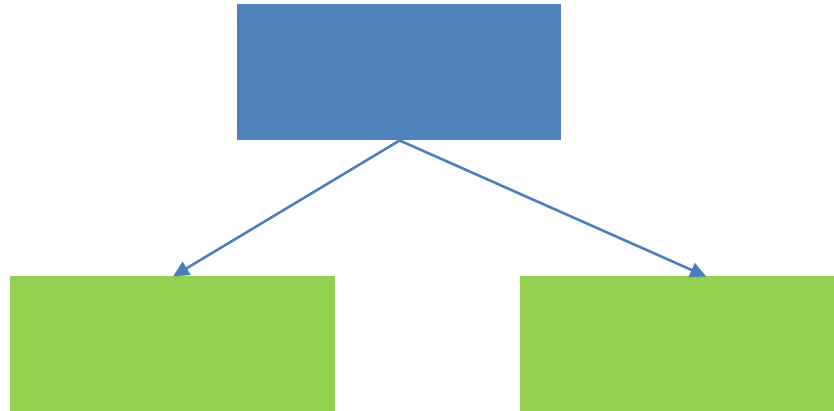
ML Regression

IV. Boosting 기반 지도학습

1. **Adaboost**
2. **Gradient Boost**

1. Adaboost

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

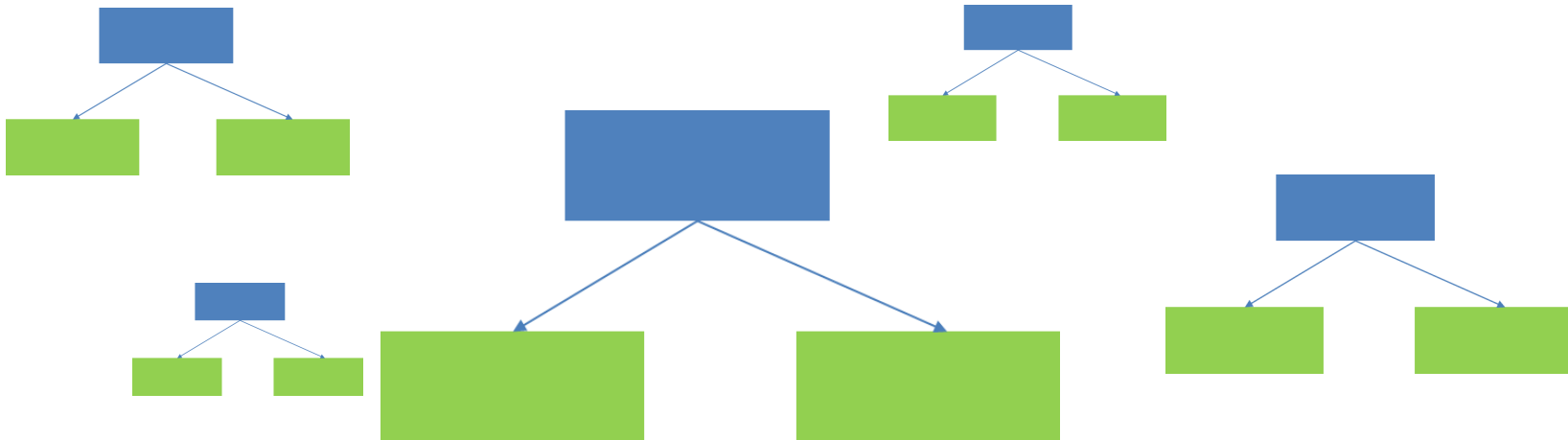


1. Adaboost

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

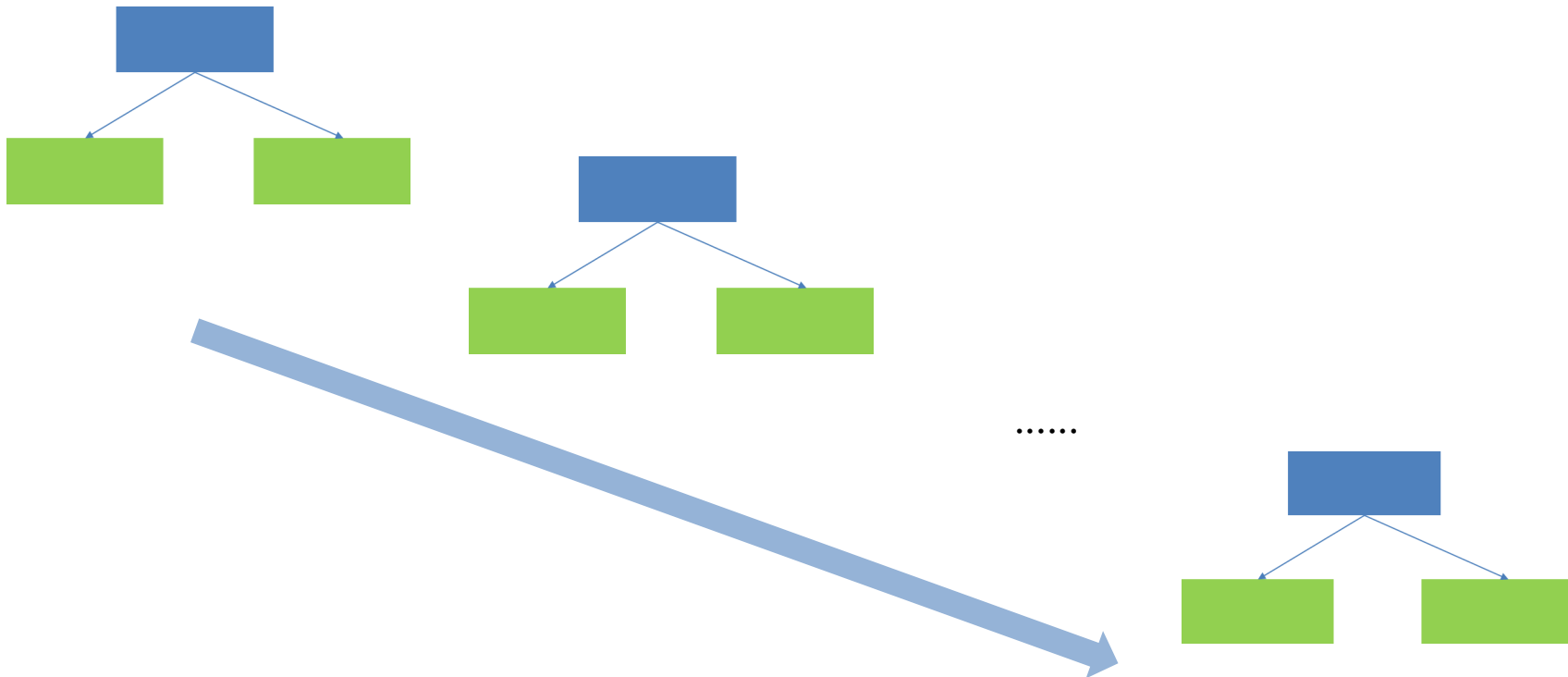


- Adaboost에서의 stump의 중요도: Amount of say로 표현, 클 수록 결과에 큰 영향을 미침



1. Adaboost

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



2. 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

| | | | 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 |

2. Gradient Boost

- **Gradient Boost, Step 1**

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

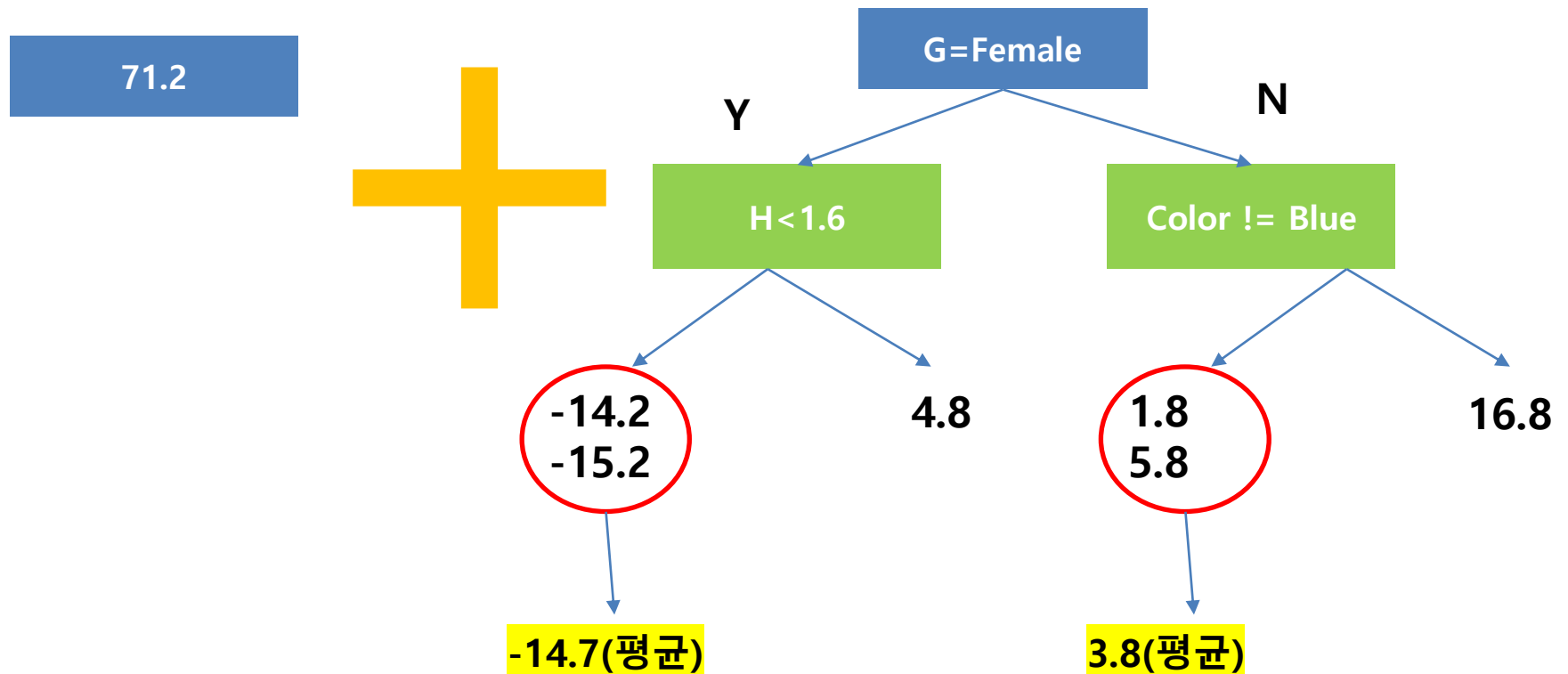
같은 X변수들로
Residual에 대한 Tree

| 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 |

2. Gradient Boost

- Gradient Boost, Step 1

- Leaf + **1st Tree**



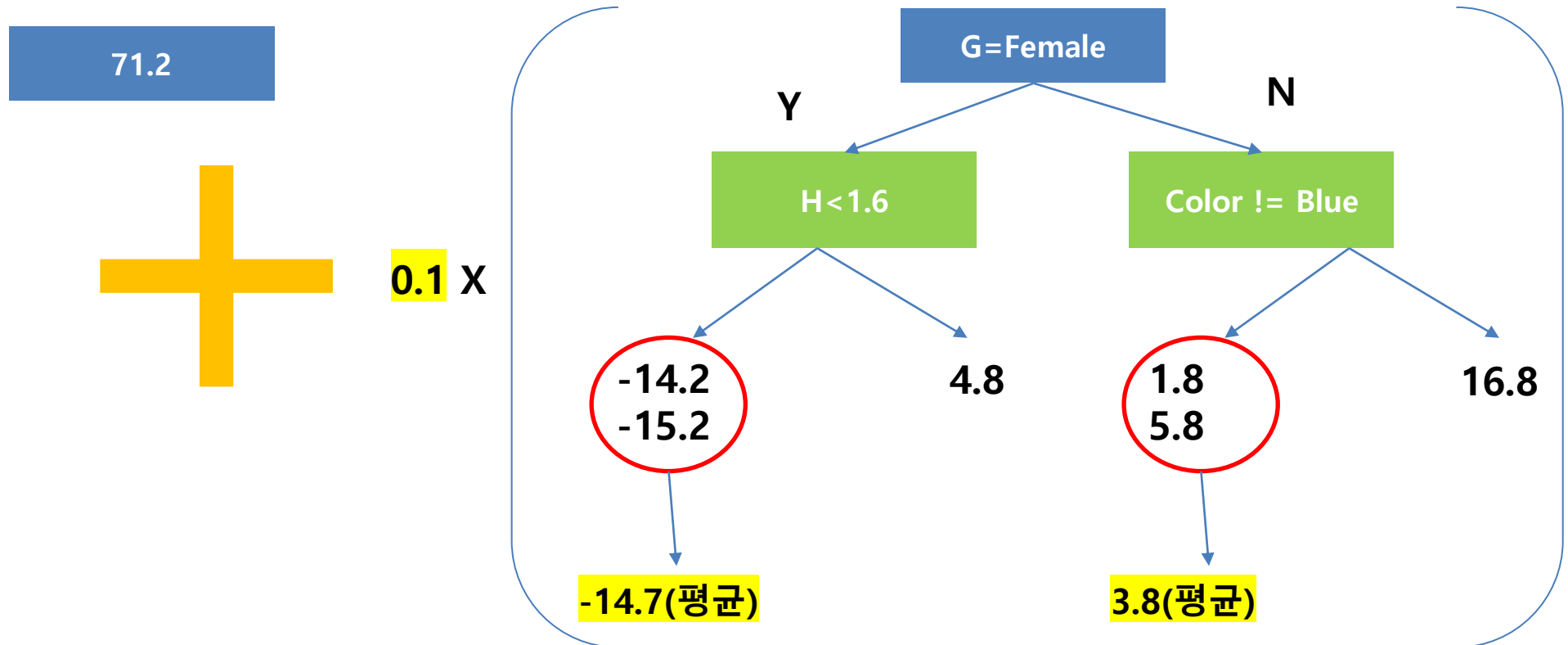
- Male, Blue인 경우 예측 예시:**

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

2. Gradient Boost

- Gradient Boost, Step 2

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

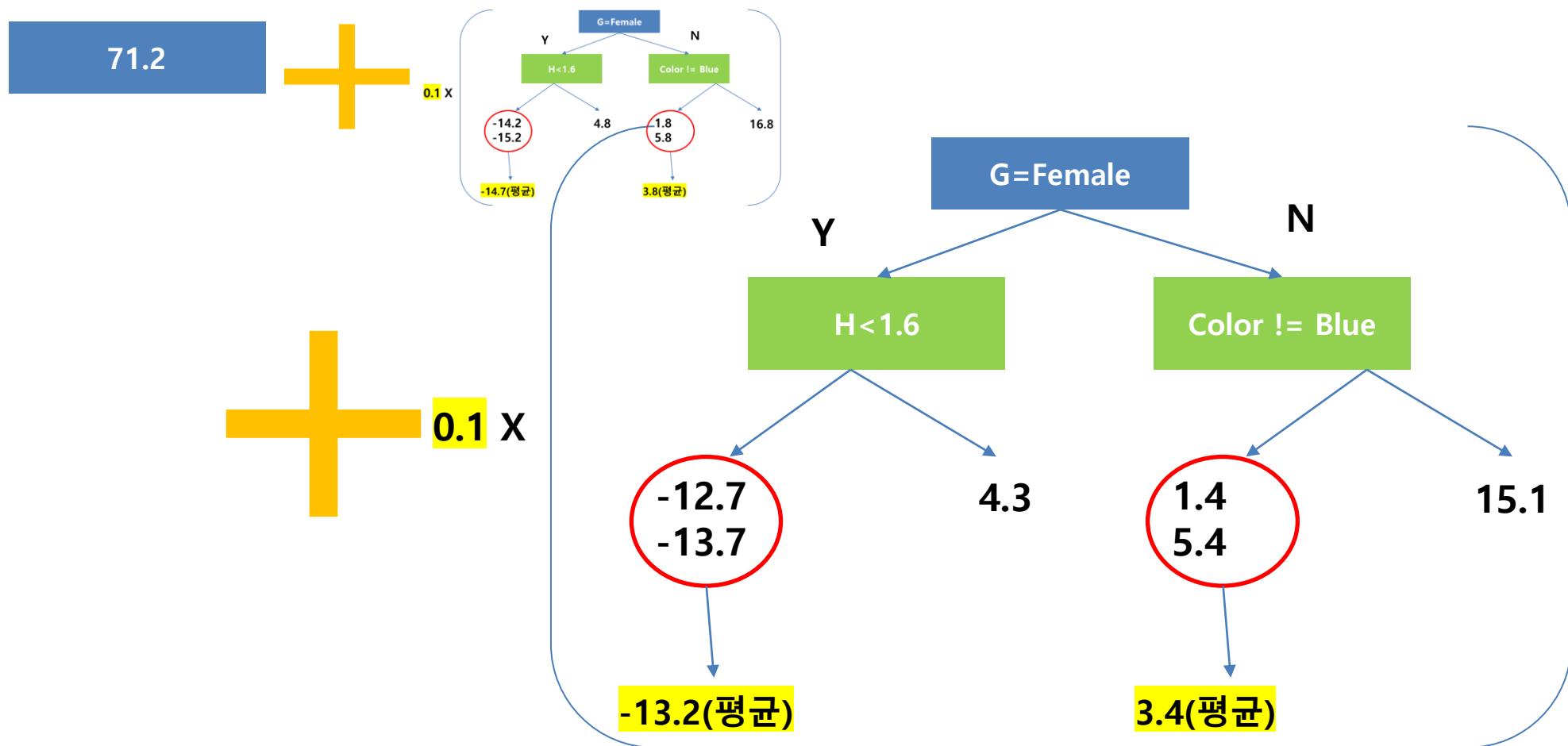


- Male, Blue인 경우 예측 예시:** $71.2 + 0.1 \times 16.8 = 72.9$
 - 실제값에 가까워지지만, 그 정도가 조절됨 (Gradient의 개념)
 - Variance를 낮게 유지할 수 있음

2. Gradient Boost

- Gradient Boost, Step 3

- Learning Rate: 0~1사이, 이 예에서는 0.1 사용



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

2. 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

2. 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
 - 1st tree: leaf 8개
 - 2nd tree: leaf 32개
 - 3rd tree: leaf 16개
 - ...

2. Gradient Boost

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

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

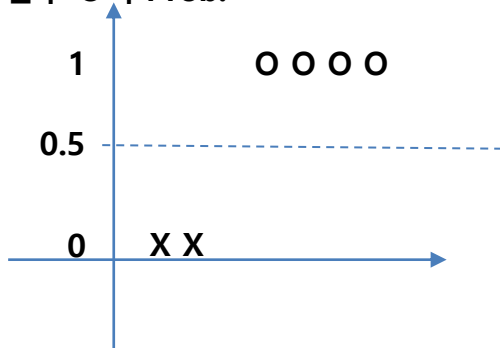
| X1 | X2 | X3 | Target |
|----|----|-------|--------|
| Y | 12 | Blue | O |
| Y | 87 | Green | O |
| N | 44 | Blue | X |
| Y | 19 | Red | X |
| N | 32 | Green | O |
| N | 14 | Blue | O |

2. 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

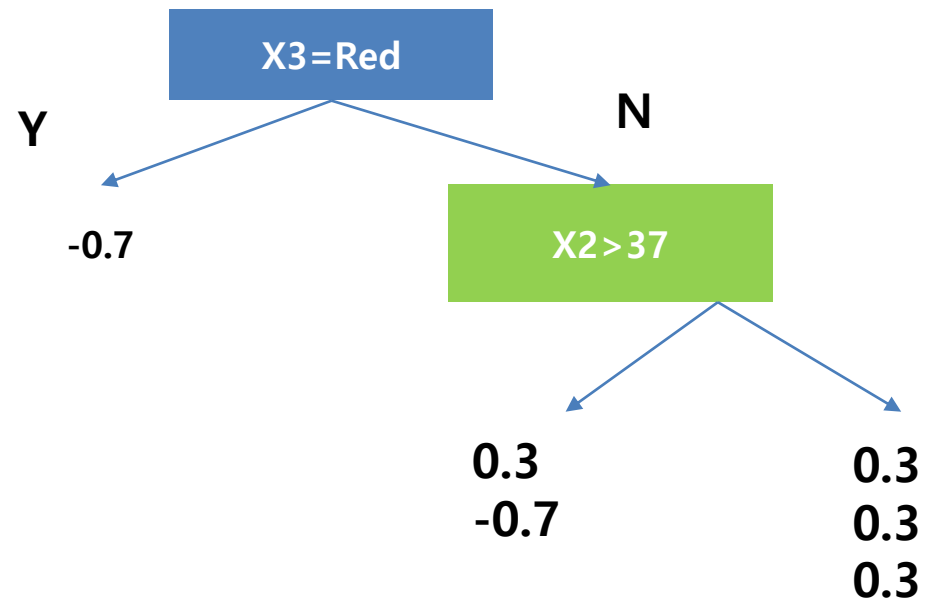
| X1 | X2 | X3 | Target | Residual |
|----|----|-------|--------|----------|
| Y | 12 | Blue | O | 0.3 |
| Y | 87 | Green | O | 0.3 |
| N | 44 | Blue | X | -0.7 |
| Y | 19 | Red | X | -0.7 |
| N | 32 | Green | O | 0.3 |
| N | 14 | Blue | O | 0.3 |

2. Gradient Boost

- Gradient Boost, Step 1

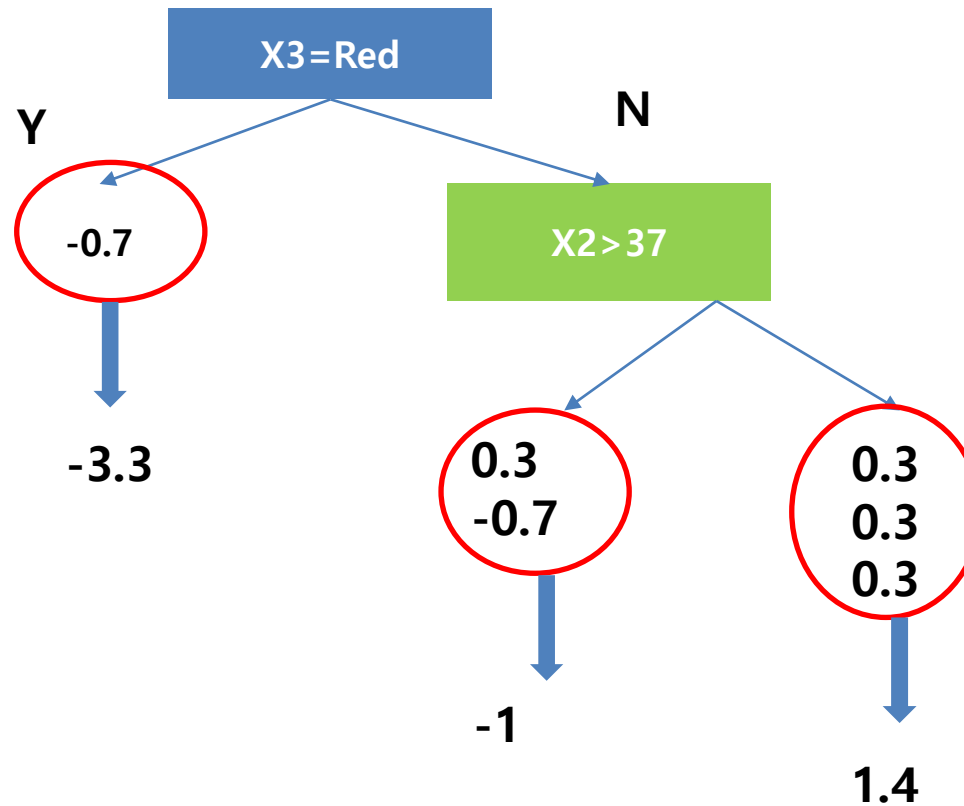
- 1st Tree**

- leaf의 수를 8~32로 제한하며 그 범위내에서 tree 생성



2. Gradient Boost

- Gradient Boost, Step 1
 - 1st Tree



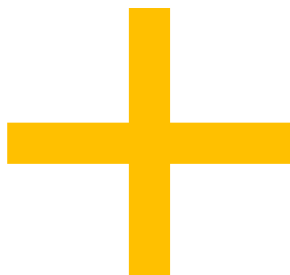
$$\frac{\sum \text{Residuals}}{\sum (\text{Previous Prob} \times (1 - \text{Previous Prob}))}$$

2. Gradient Boost

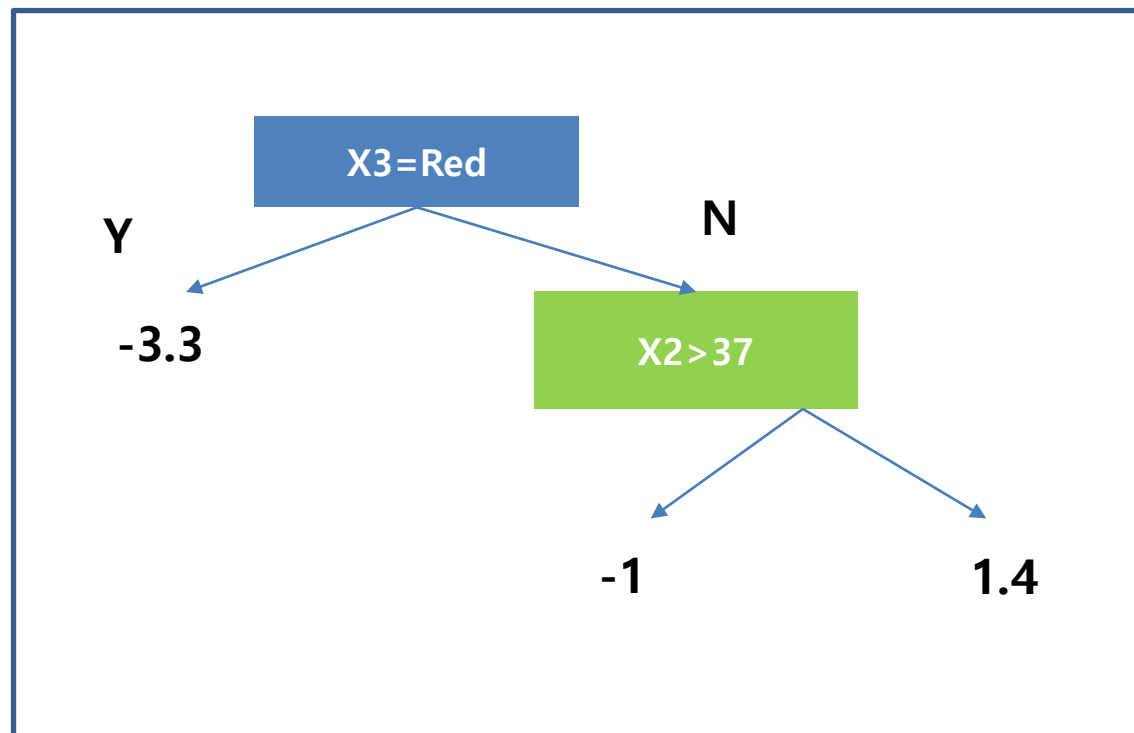
- Gradient Boost, Step 2

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

0.7



0.8 X



2. Gradient Boost

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

- 각 범주에 대한 발생 확률 계산
 - 1st Obs의 업데이트된 $\log(\text{odds})$ 는 1.8
 - Leaf $0.7 + 1.4(\text{from tree}) \times 0.8 = 1.8$
 - 1st 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 |

2. Gradient Boost

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

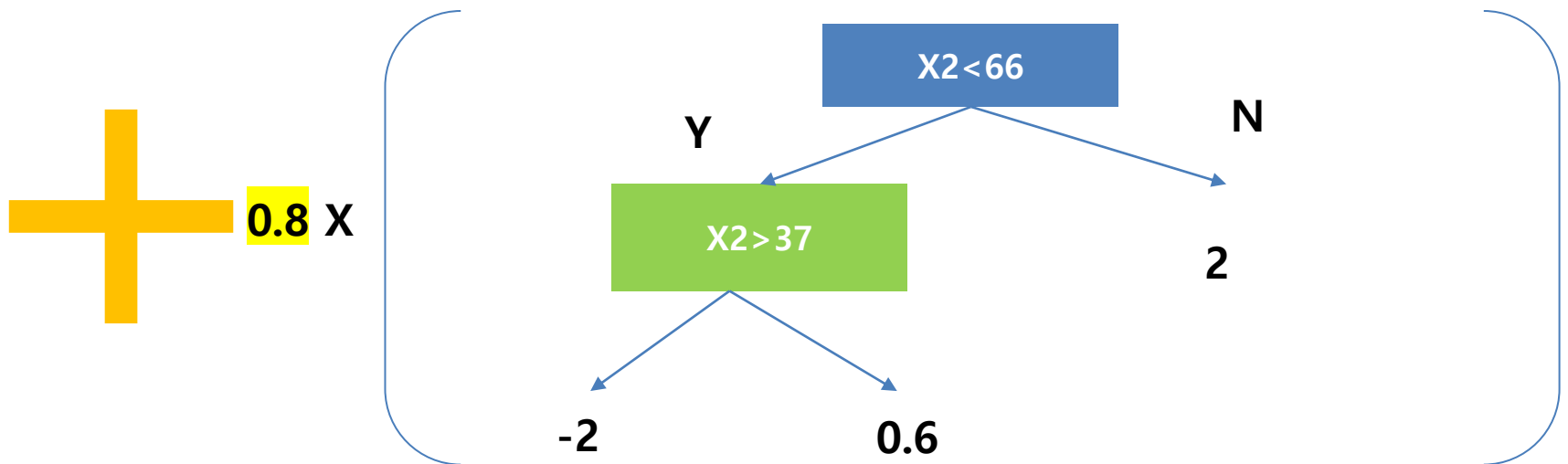
- Residual 다시 계산, 다음 tree 생성

| 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 |

2. Gradient Boost

- Gradient Boost, Step 3

- Learning Rate: 0~1사이, 이 예에서는 0.1 사용



Q&A