- Decision Tree.
  - classification and Regression 둘 다 가능!
  - 예측 결과물에 대해 쉽게 설명할 수 있어 자주 사용하는 모델이다.
    (The output is a set of rules)
  - 설명 변수의 형태 상관없이 사용 가능.
    (Able to handle both numeric, cate)

- Decision Tree (classification)
  - CART Algorithm
    → classification & Regression Tree

≈ CHAID, C4.5, C5.0

- Recursive Partitioning
  → 반복적으로 분할한다.
  (Impurity를 기준으로)

- Pruning the Tree.
  → Full Tree 에서 Merge (Post-Pruning)
  사전조건으로 Cut. (Pre-Pruning)

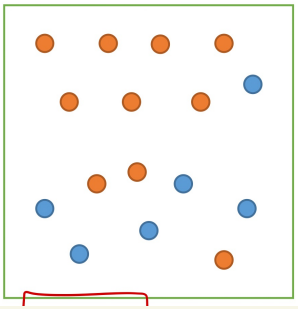
- Recursive Partitioning

- Impurity를 기준으로 분할한다.

① Gain Index.

→ CART에서 사용하는 measure

→ $I(A) = 1 - \sum_{k=1}^{m} P_k^2$



$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

$$= 1 - \left(\frac{6}{16}\right)^2 - \left(\frac{10}{16}\right)^2$$

$$\approx 0.47$$

⇒ Impurity가 높은편.

● 개수 : 10, ● 개수 : 6

→ Max : $1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = \frac{1}{2} = 0.5$

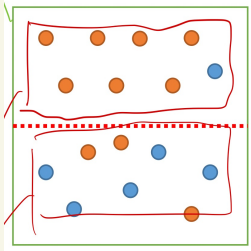
→ Min : $1 - 1^2 - 0^2 = 0$

→ 즉, 작을수록 좋은 분할.

Gini Index를 사용하여 가장 좋은 분할 방법 찾기.

$$\rightarrow I(A) = \sum_{i=1}^{d} \left(R_i \left(1 - \sum_{k=1}^{m} p_{ik}^2 \right) \right)$$

⇒ Gini Index의 가중합.



↳ 분할된 데이터의 비율.

$$I(A) = \frac{8}{16} \times 0.5 \times \left(1 - \left(\frac{7}{8} \right)^2 - \left(\frac{1}{8} \right)^2 \right) + \frac{8}{16} \times 0.5 \times \left(1 - \left(\frac{3}{8} \right)^2 - \left(\frac{5}{8} \right)^2 \right)$$

$$= 0.34$$

→ Information Gain

$= 0.47 - 0.34 = 0.14$

⇒ split에 대한 효과 해석 가능.

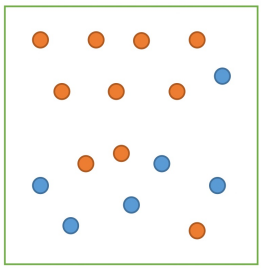
⇒ split point에 대해 Information Gain

을 비교해 가장 큰 값을 가지는
split point로 분할한다!

② Deviance.

→ 데이터 Tree에서 사용하는 것.

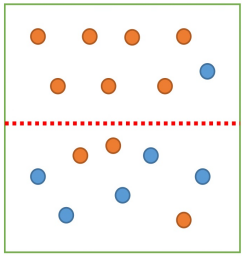
$$D_2 = -2 \sum_{k=1}^{m} n_{ik} \log(P_{ik})$$



$$D_i = -2 \times \left(10 \times log\left(\frac{10}{16}\right) + 6 \times log\left(\frac{6}{16}\right)\right)$$

$$= 21.17$$

→ Min : 0 (same class)



$$D_1 = -2 \times \left(7 \times log\left(\frac{7}{8}\right) + 1 \times log\left(\frac{1}{8}\right)\right) = 6.03$$

$$D_2 = -2 \times \left(3 \times log\left(\frac{3}{8}\right) + 5 \times log\left(\frac{5}{8}\right)\right) = 10.59$$

$$D_1 + D_2 = 16.62$$

Information gain

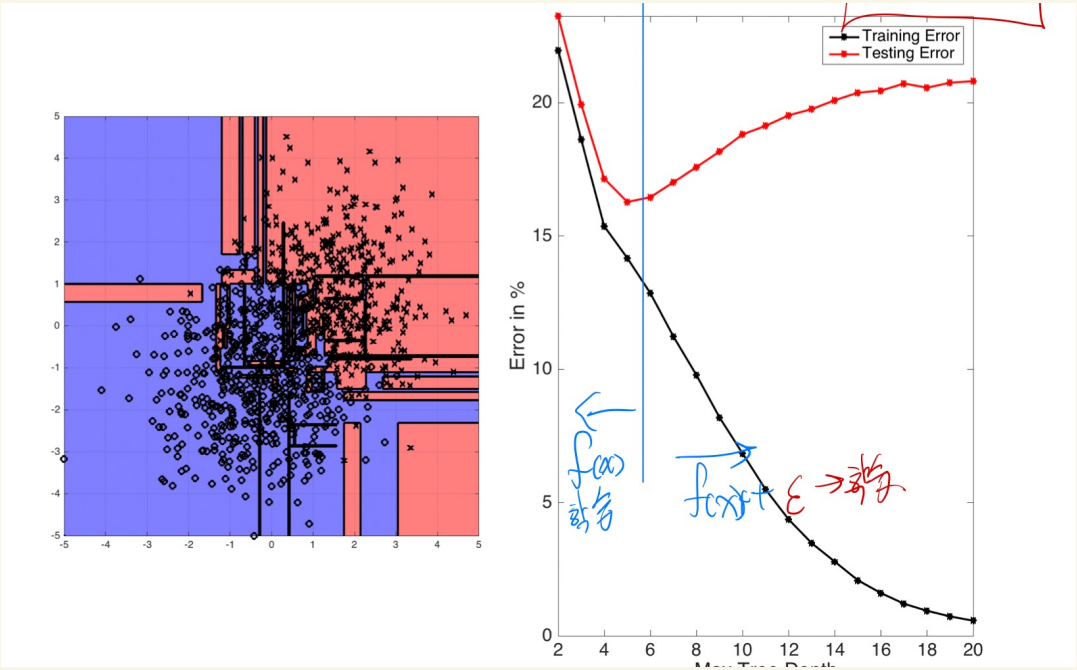
$= 21.17 - 16.62 = 4.55$

⇒ split 함.

• Full Tree 만들기

- 설명변수 : m
- record : n
- 분할 선택하다. m개의 변수에서 sorting 후 분할 포인트를 찾고, Information gain을 구한다음 최대의 값을 가지는 split을 한다.
- 종료조건 : Impurity 가 0이 되는 순간 leaf Node로 설정하여 더 이상 분할 X.

- Full Tree는 100% of Acc를 가지면 데이터의 noise까지 학습한 overfitting 된 모델이다!



- Pruning.

→ Overfitting 문제를 해결하기 위해!

→ Merge.

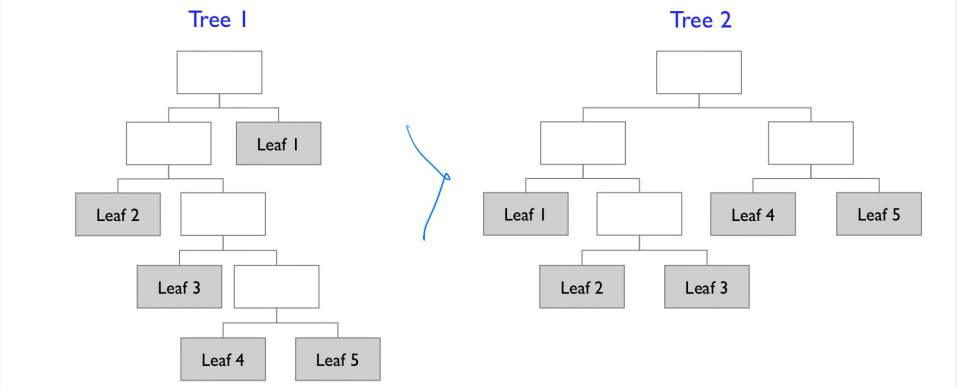
→ Cost complexity를 계산!

$$CC(T) = Err(T) + \alpha \times L(T)$$

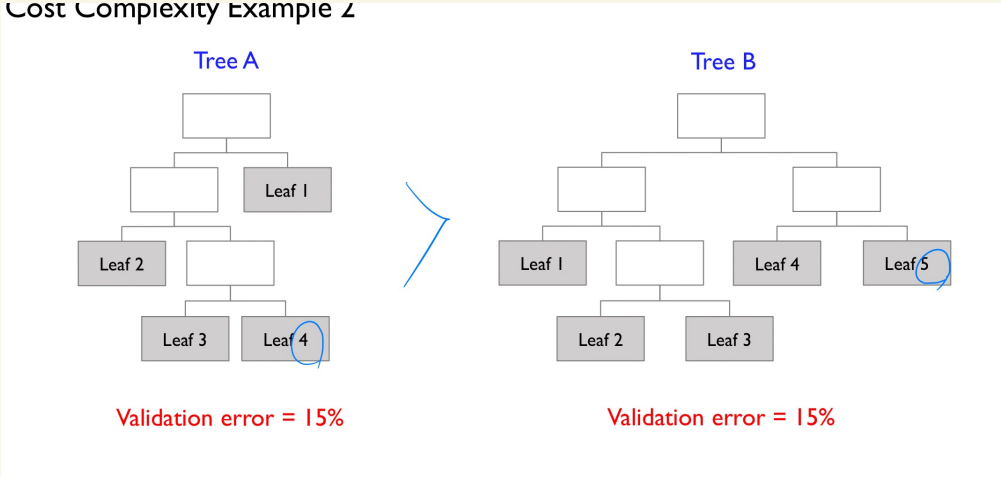
* $Err(T)$
  → proportion of misclassified records in the variables data

*L(T) : Number of leaf Node

Cost Complexity Example 1

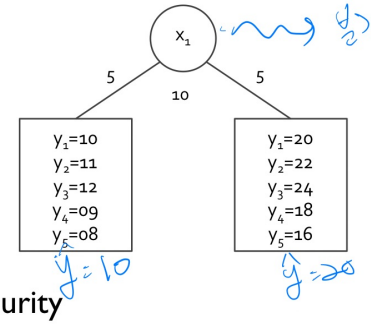


Cost Complexity Example 2



- Regression Tree

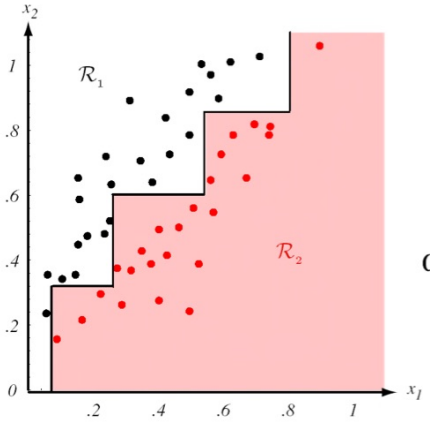
→ output : 영역의 종속변수의 평균!

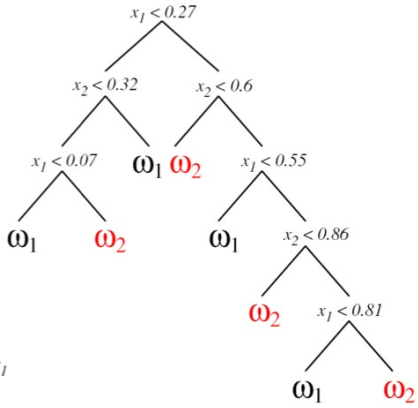


- Predicted value of the left leaf node = 10
- Predicted value of the right leaf node = 20

- Impurity
  - ✓ Sum of squared error (SSE: $\sum_{i=1}^{n}(y_i - \hat{y})^2$)
  - ✓ SSE(Parent) = 300, SSE(Left) = 10, SSE(Right) = 40, Gain = 250

CART Disadvantages.





one variable at a time