Decision Tree

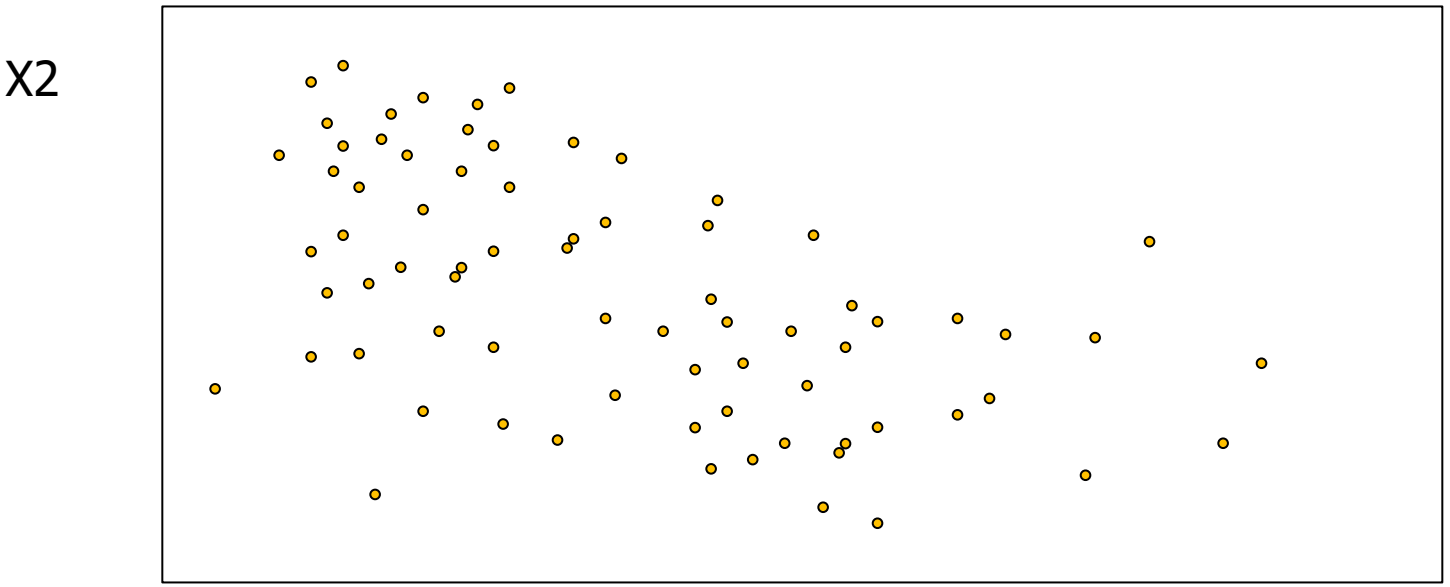
Sang Yup Lee

Decision Tree

- 기본 원리
  - DT는 dataset에 있는 관측치들을 독립변수 (feature)의 값에 따라서 종속변수의 값이 유사한 여러 개의 그룹으로 분리하고, 각 그룹에 속한 관측치들의 종속변수 값을 동일한 값으로 예측하는 알고리즘
    - 회귀문제와 분류문제 모두 적용 가능
    - 회귀문제에 적용되는 DT: Decision Tree Regressor
    - 분류문제에 적용되는 DT: Decision Tree Classifier
  - DT는 독립변수의 값을 이용하여 관측치들을 서로 다른 그룹으로 분리하기 위해서 Tree 형태의 분리 과정을 사용

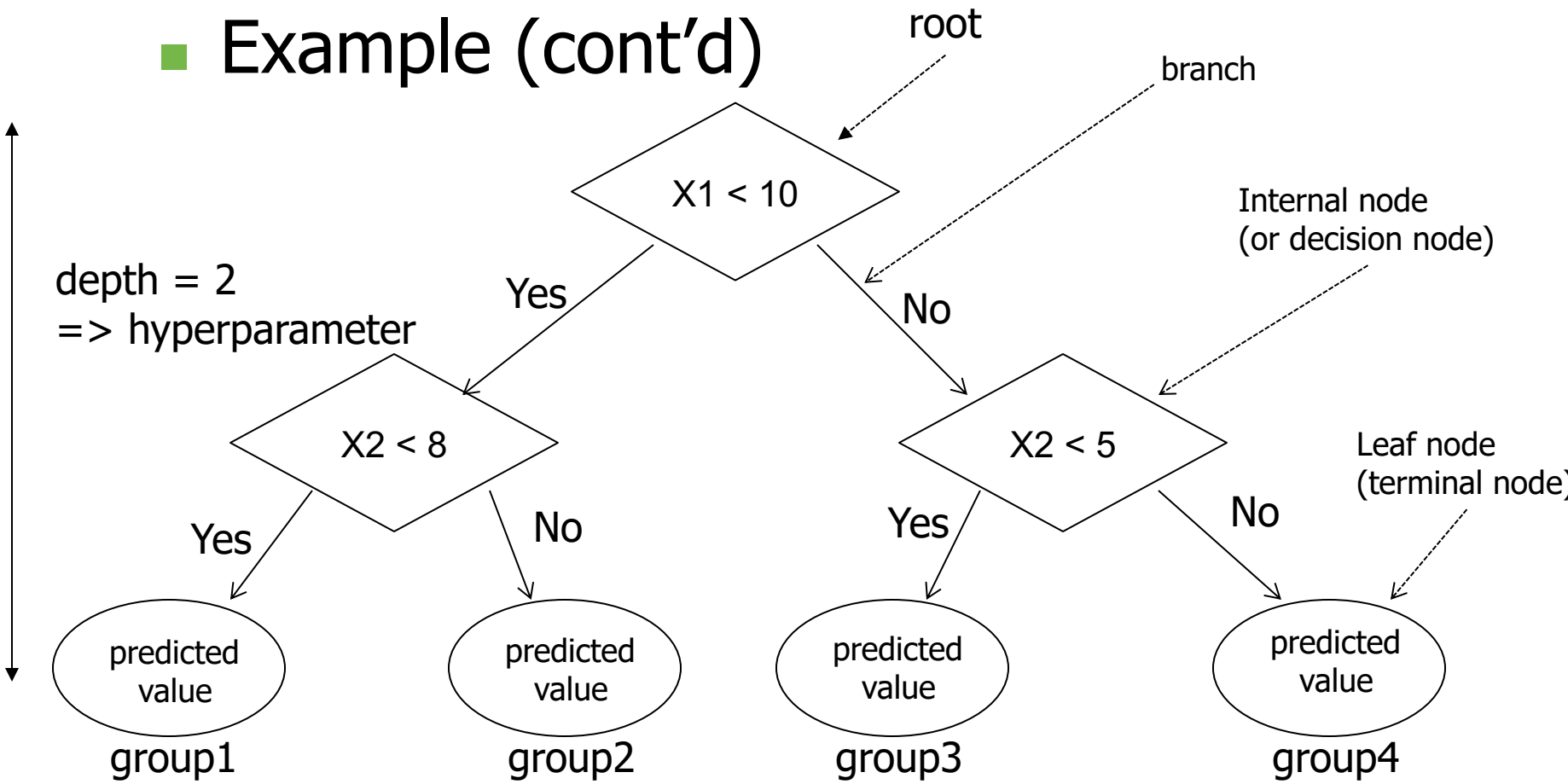
Decision Tree

- Example with two IVs (X1, X2)



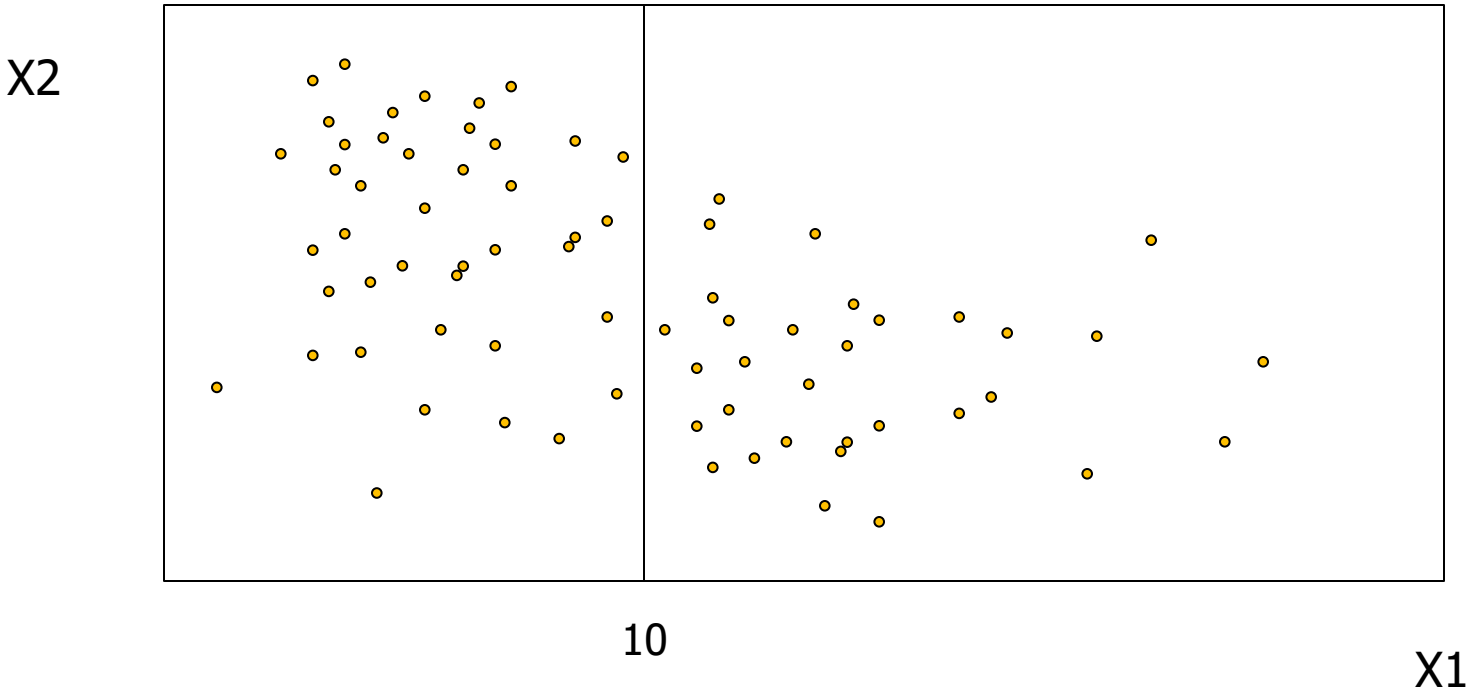
Decision Tree

- Example (cont'd)



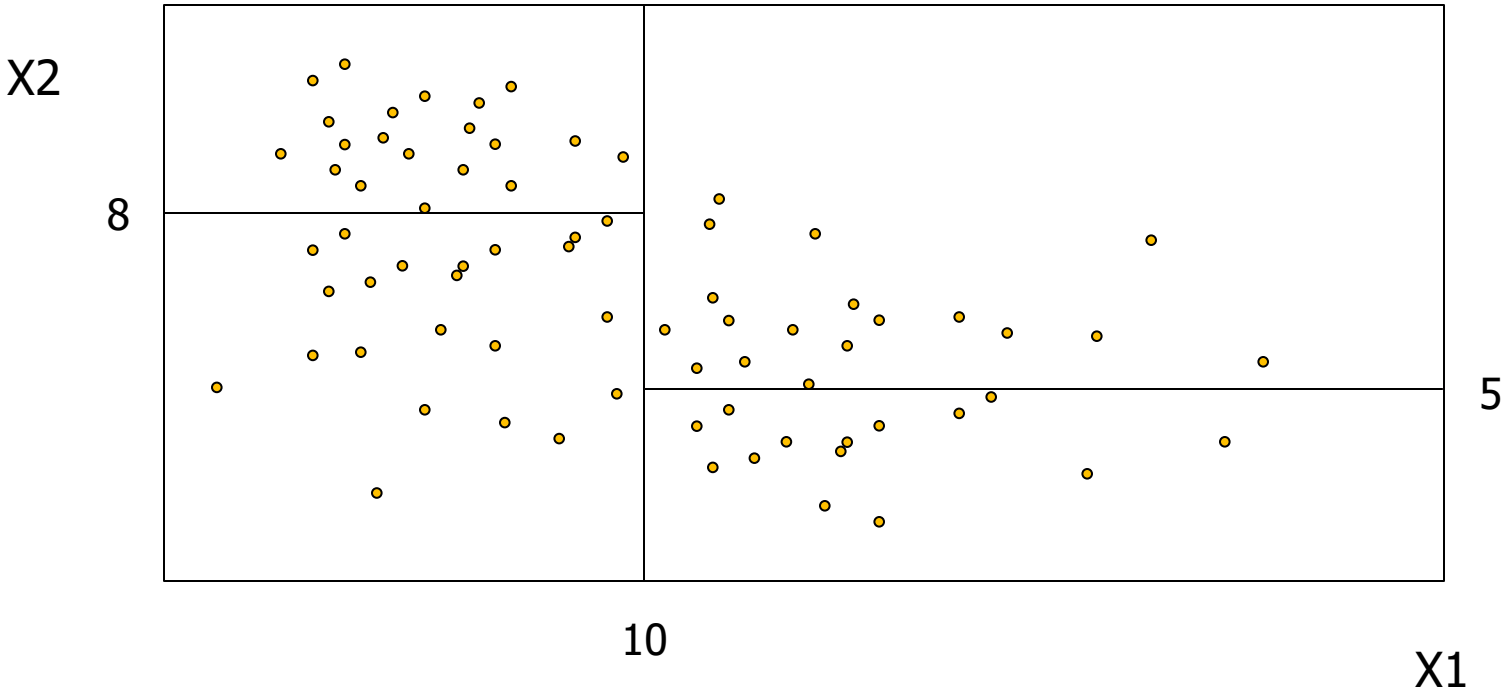
Decision Tree

- Example (cont'd) – step 1



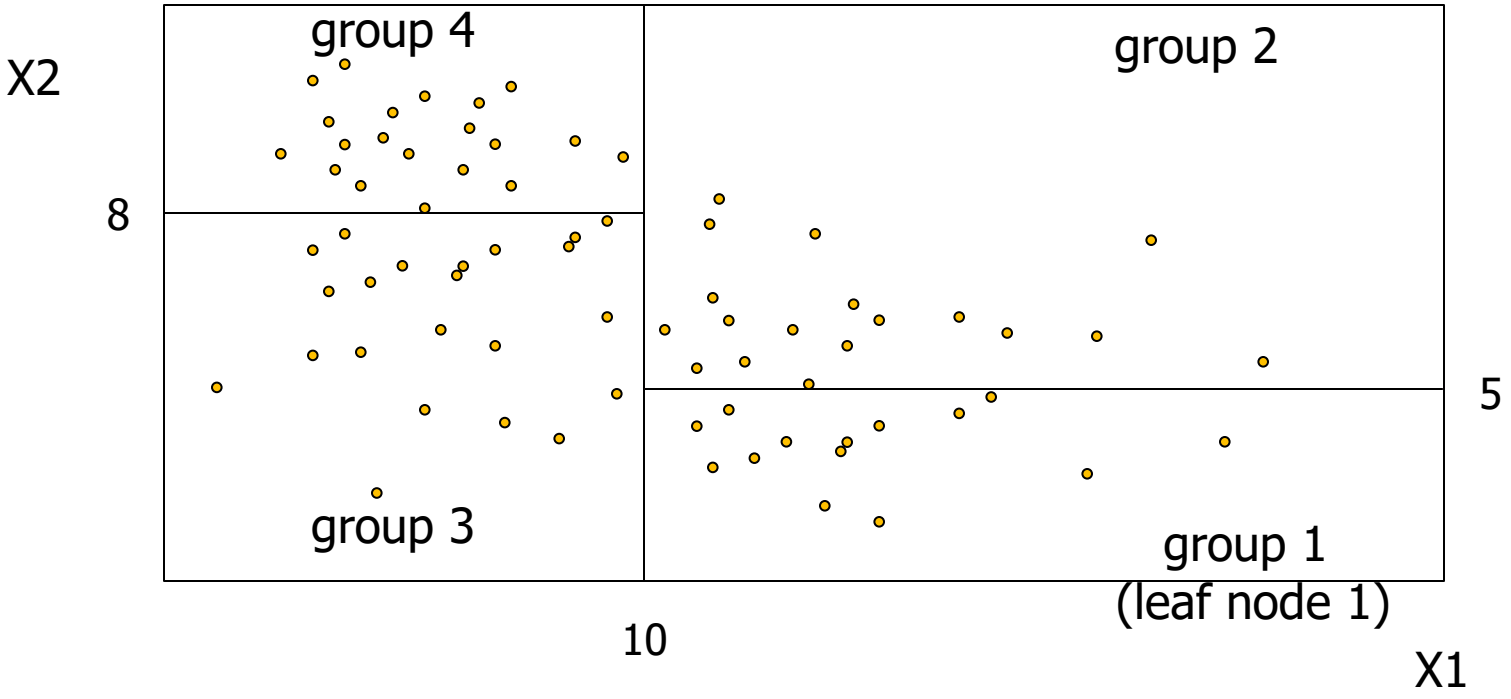
Decision Tree

- Example (cont'd) – step 2



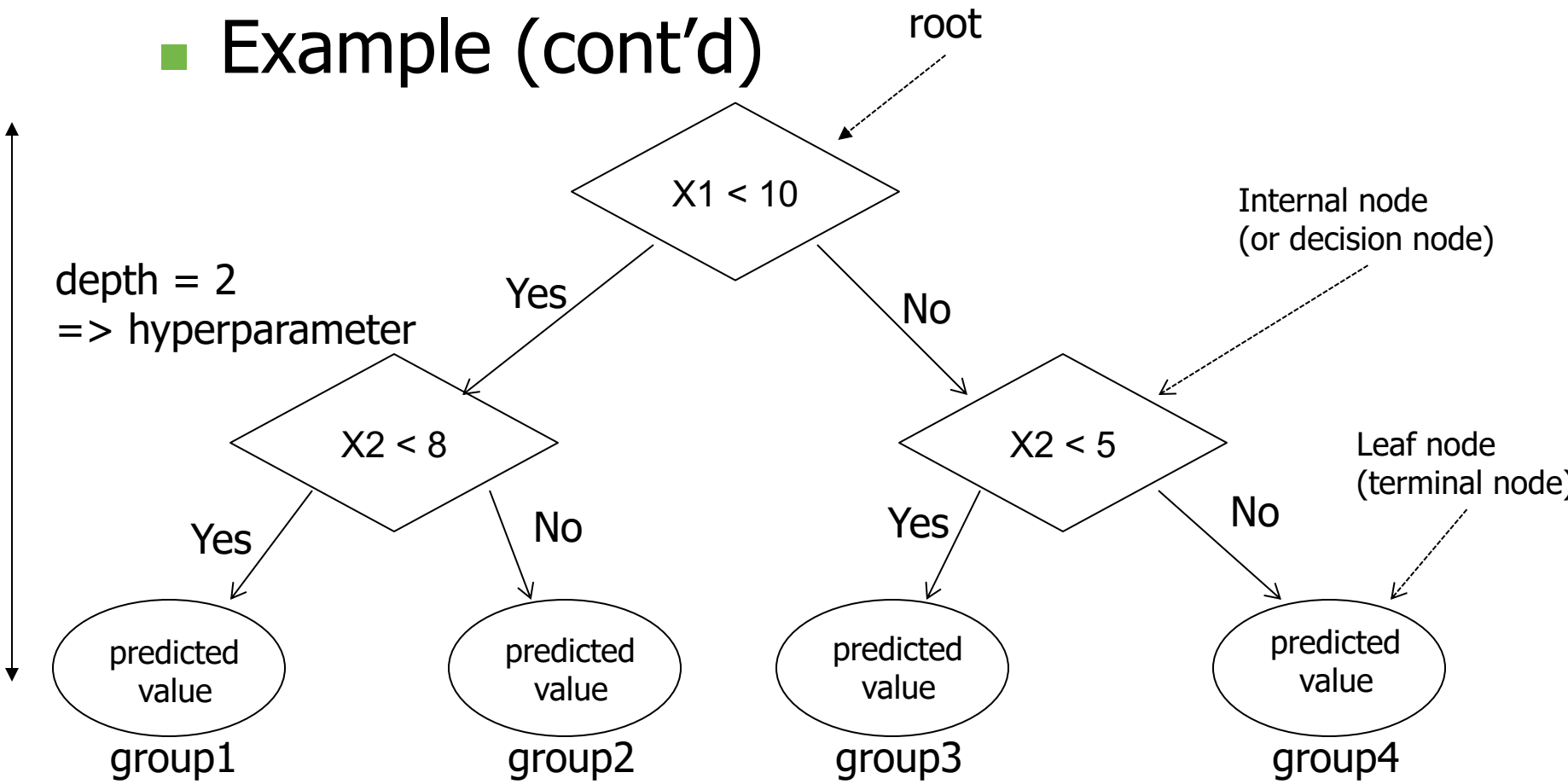
Decision Tree

- Example (cont'd)



Decision Tree

- Example (cont'd)



Decision Tree

- 종속변수 값의 예측
  - 각 그룹에 속한 관측치들은 동일한 예측치를 갖는다.
  - Group k에 속한 관측치들의 종속변수 값 예측치
    - 회귀문제
      - 평균값 사용
      - $\hat{y}_k = \frac{1}{m_k}\sum_{i \in Group_k} y_i$, where m_k = # of points in group k
    - 분류문제
      - Mode 값 (최빈값) 사용
      - 즉, 해당 그룹에서 가장 많은 관측치가 갖는 종속변수 값을 예측치로 사용

Decision Tree

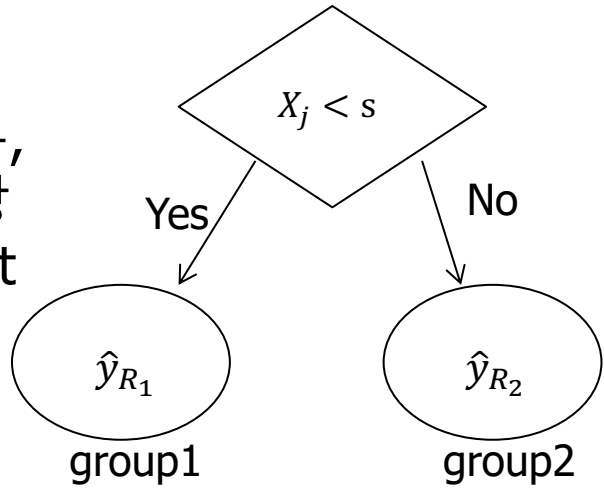
- 종속변수 값의 예측 (cont'd)
  - 특정 그룹에 대해서
    - 회귀문제
      - 10개의 관측치들의 종속변수 값 => (2, 3, 4, 4, 3, 6, 4, 10, 2, 12)
      - 종속변수의 예측치는 (2+ 3+ 4+ 4+ 3+ 6+ 4+ 10+ 2+ 12)/10 = 5
    - 분류문제
      - 종속변수가 취할 수 있는 값 => 0, 1, 2
      - 10개의 관측치들의 종속변수 값 => (2, 0, 1, 1, 1, 2, 1, 1, 0, 1)
      - 종속변수의 예측치 => 최빈값

Decision Tree

- How to split data into two groups?
  - 즉, 각 decision node에서 어떠한 변수의 어떠한 값(cutpoint value)으로 데이터를 split 할 것인가?
    - split되었을 때 발생하는 에러 정도를 최소화하게끔 split
    - 회귀문제와 분류문제에서의 에러 정도를 계산하는 방법이 상이

Decision Tree

- 회귀 문제
  - 각 그룹 (e.g., Group j)의 RSS
    - $\sum_{i \in G_j} (y_i - \hat{y}_{G_j})^2$
    - $\hat{y}_{G_j}$: Group j에 대한 종속변수 예측치
  - 노드를 split해서 두개의 그룹이 발생하는 경우, 각 그룹 RSS 합을 minimize 하는 변수 j와 해당 변수의 값 (s)을 찾아야 함, 이를 기준으로 split
    - 즉, $\min_{j,s}(RSS_1 + RSS_2)$



Decision Tree

- 분류 문제
  - 분류 문제의 경우, 각 그룹에서의 오차 정도를 측정하기 위해 다음 두가지 값을 사용
  - Gini index
    - $G = \sum_{k=1}^{K} \hat{p}_{j,k}(1 - \hat{p}_{j,k})$
      - 종속변수가 취할 수 있는 값 => 1, ..., K
      - $\hat{p}_{j,k}$= Group j에서 class k의 비중 = $\frac{m_k}{mj}$

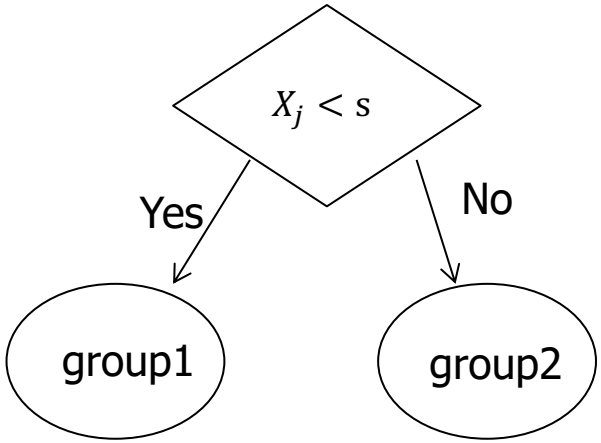
Example) Group 1
(0,0,0, 1,1,1, 2,2,2,2)
$\hat{p}_{1,0}$ =?

Decision Tree

- 분류 문제
  - Entropy
    - $E = -\sum_{k=1}^{K} \hat{p}_{j,k} \log \hat{p}_{j,k}$
      - $\hat{p}_{j,k}$= Group j에서 class k의 비중 = $\frac{m_k}{mj}$
    - Group j에 존재하는 종속변수의 불확실성을 의미
    - 종속변수의 값이 동일할수록 불확실성 감소 => 즉 entropy 값 감소
  - 두 값 모두 impurity (or heterogeneity / uncertainty) 정도를 의미 (즉, 각 그룹에서 동일한 class의 points가 많을 수록 값이 작아짐)
    - 모든 dp가 동일한 값을 갖을 경우 제일 작고
    - 각 값을 갖는 dp의 수가 동일한 경우 제일 크다.

Decision Tree

- 분류 문제
  - 각 노드에서 다음 값을 minimize하는 변수와 해당 변수의 값을 찾아야 함
    - $Gini_1 + Gini_2$ 또는
    - $E_1 + E_2$
  - 각 그룹의 data points 수에 따라 weight를 주기도 함 (즉, weighted average 사용)
    - $\frac{n_1}{n} Gini_1 + \frac{n_2}{n} Gini_2$

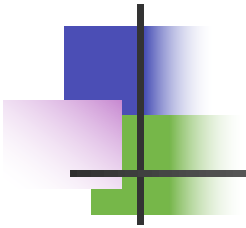


Decision Tree

- 분류문제
  - DT_clf_iris.ipynb
  - petal length and width 만 사용
- In Python, DecisionTreeClassifier 사용
  - https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
  - 주요 hyperparameters
    - criterion: gini or entropy
    - max_depth: maximum depth of the tree
    - min_samples_split: minimum number of data points a decision node must have before it can be split
    - min_samples_leaf: minimum number of data points a leaf node must have
    - max_leaf_nodes: maximum number of leaf nodes

Stopping criterion

Hyperparameter가 많아서, Gridsearch 방법 사용 권고



Exercise

- 회귀문제
  - DT_reg_hitters.ipynb