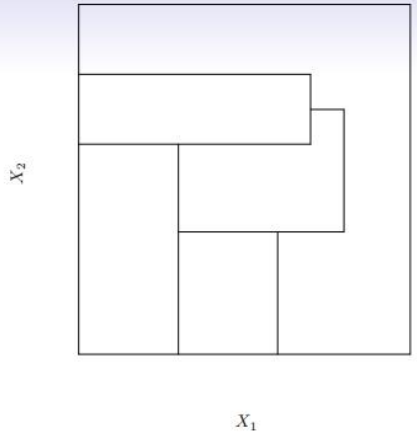
데이터 분석(Python / R)

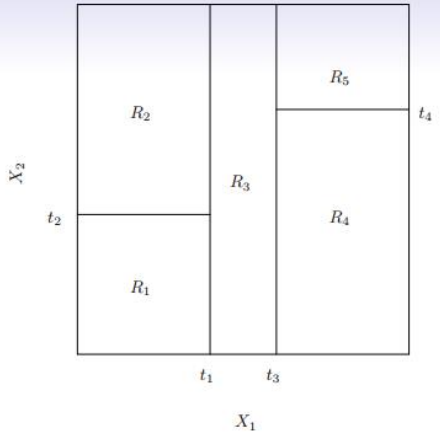
Supervised Learning(지도학습)

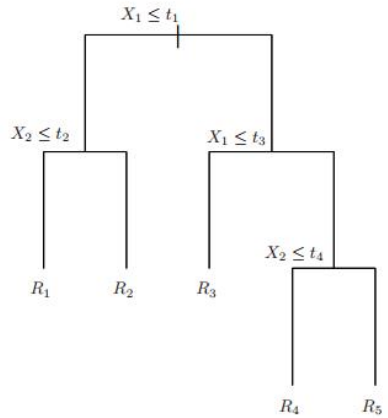
삼성전자공과대학교 3학년 3학기

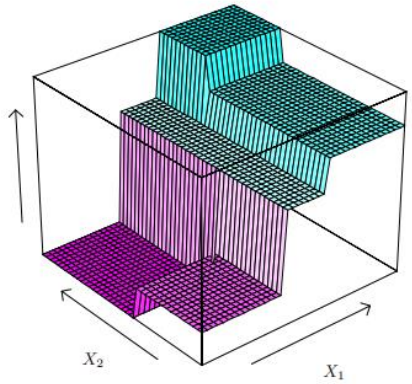
Decision Tree

❑ Tree-based model





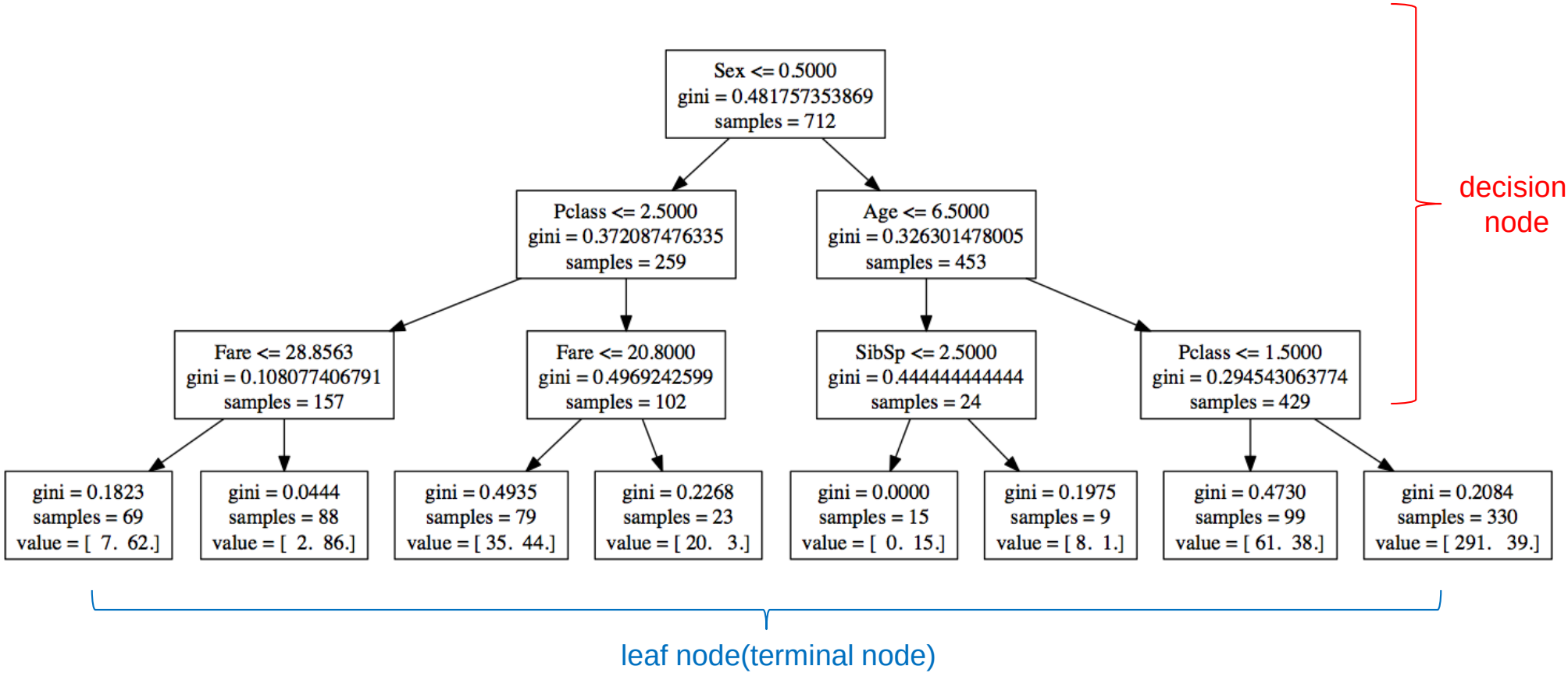




❑ Tree-based model

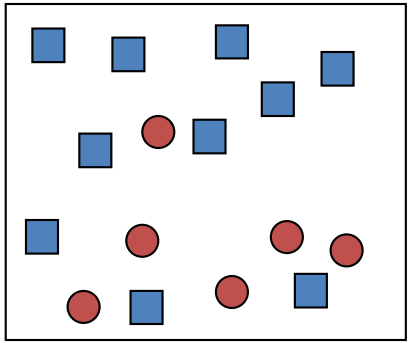
- ❖ 개별 변수의 영역을 반복적으로 분할함으로써 전체 영역에서의 규칙을 생성하는 지도학습 기법
- ❖ If-else-then 형식으로 표현되는 rules을 생성함으로써, 결과에 대한 예측과 함께 그 이유를 설명할 수 있다.
- ❖ Classification, regression 모두 가능
- ❖ 입력 변수들 중 데이터를 가장 잘 분류하는 변수를 택하여 분기가 계속 일어 남
- ❖ 모델에 대한 해석력이 좋다
  - ▪ 분기되는 변수는 중요한 변수
  - ▪ 분류 규칙 추출 가능
- ❖ 변수 선택이 모델 자체에서 이루어짐
- ❖ Continuous, discrete, category 변수 모두 사용 가능

❑ Tree-based model



❑ Tree-based model

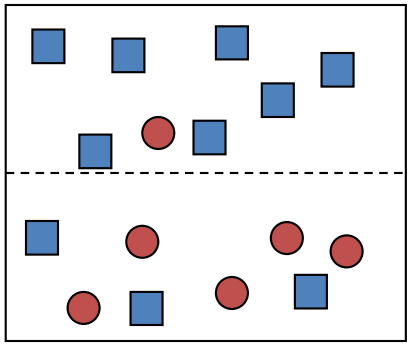
❖ Gini index



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

$$I(A) = 1 - \left(\frac{6}{16}\right)^2 - \left(\frac{10}{16}\right)^2 = 0.47$$

- $I(A) = 0$: 모든 데이터가 같은 클래스
- $I(A) =$ 최대값 : 모든 클래스의 데이터가 동일한 숫자



$$I(A) = \sum_{i=1}^{d} \left(R_i \left(1 - \sum_{k=1}^{n} p_k^2 \right) \right)$$

R_i : 클래스의 비율

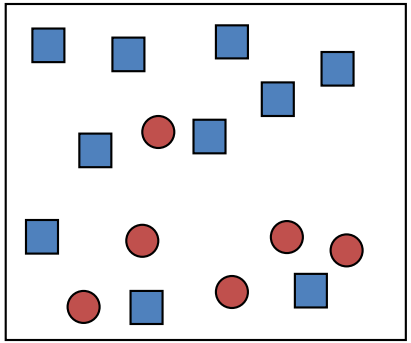
$$I(A) = 0.5 * \left(1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2\right) + 0.5 * \left(1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2\right)$$

$= 0.34$

- $information\ gain : 0.47 - 0.34$

❑ Tree-based model

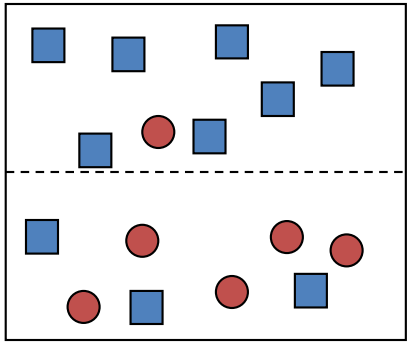
❖ Entropy



$$I(A) = -\sum_{k=1}^{n} p_k log_2(p_k)$$

$$I(A) = 0.95$$

- $0 \leq I(A) \leq log\ (n)$



$$I(A) = \sum_{i=1}^{d} \left(R_i \left(-\sum_{k=1}^{n} p_k log_2(p_k) \right) \right)$$

$$I(A) = 0.75$$

R_i : 클래스의 비율

- $information\ gain$: $0.95 - 0.75$

❑ Tree-based model

❖ Basic algorithm

i. 전체 데이터를 포함하는 root node 생성

ii. 만일 샘플들이 모두 같은 클래스이면 node는 leaf가 되고 해당 클래스로 label 부여

iii. 그렇지 않으면 information gain이 높은 속성 선택

iv. 선택된 속성으로 branch를 만들고 하위 node 생성

v. 각 노드에 대하여 ii 부터 반복

➢ 정지 조건 : 해당 node에 속하는 데이터들이 모두 같은 클래스를 가지거나 상위 node에서 모든 속성을 사용

❖ Information gain : 특정 속성을 기준으로 데이터를 구분할 때 감소되는 entropy의 양

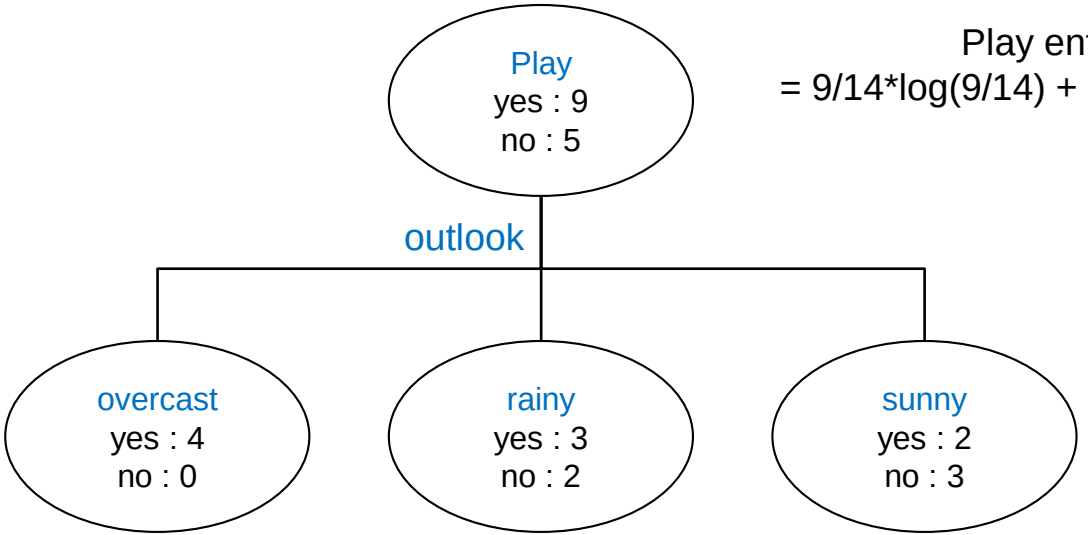
$$Gain(S,A)=Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (\ S_v=\{s \in S | A(s)=v\} \ )$$

❖ Entropy(무질서도) $H(p) = -\sum_{x \in X} p(x) \log p(x)$

❑ Tree-based model

Outlook	Temperature Numeric	Temperature Nominal	Humidity Numeric	Humidity Nominal	Windy	Play
overcast	83	hot	86	high	FALSE	yes
overcast	64	cool	65	normal	TRUE	yes
overcast	72	mild	90	high	TRUE	yes
overcast	81	hot	75	normal	FALSE	yes
rainy	70	mild	96	high	FALSE	yes
rainy	68	cool	80	normal	FALSE	yes
rainy	65	cool	70	normal	TRUE	no
rainy	75	mild	80	normal	FALSE	yes
rainy	71	mild	91	high	TRUE	no
sunny	85	hot	85	high	FALSE	no
sunny	80	hot	90	high	TRUE	no
sunny	72	mild	95	high	FALSE	no
sunny	69	cool	70	normal	FALSE	yes
sunny	75	mild	70	normal	TRUE	yes

❑ Tree-based model



Play entropy(9,5)
= 9/14*log(9/14) + 5/14*log(5/14) = 0.94

Information gain(outlook)
= 0.94 - (4/14*0 + 5/14*0.97 + 5/14*0.97)
= 0.23

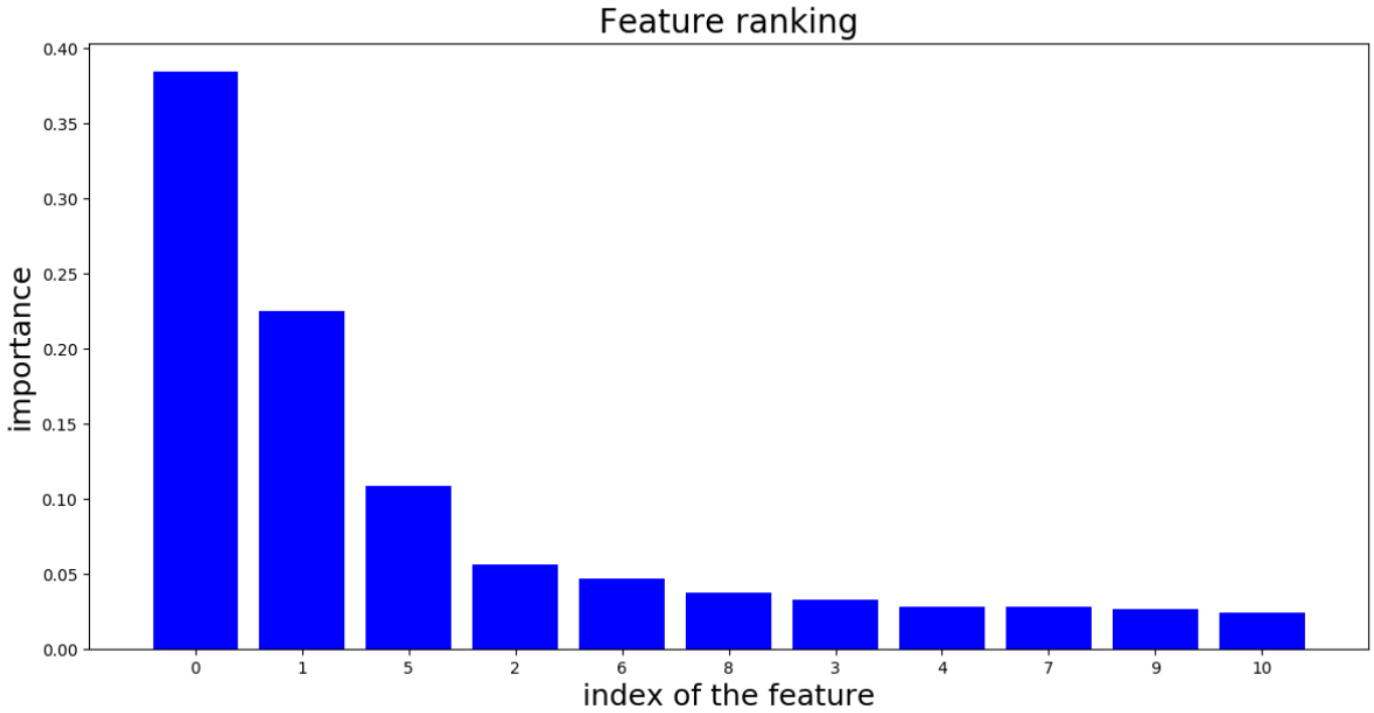
overcast entropy(4,0)
= 4/4*log(4/4) + 0/4*log(0/4) = 0

sunny entropy(4,0)
= 2/5*log(2/5) + 3/5*log(3/5) = 0.97

rainy entropy(4,0)
= 3/5*log(3/5) + 2/5*log(2/5) = 0.97

Decision Tree

- Fature Importance(특징중요도)
  - Variable Importance(변수중요도)라고도 함
  - Tree Model의 Accuracy(정확도)와 Impurity(불순도) 개선에 기여하는 정도



Decision Tree

Library

- sklearn.tree.DecisionTreeClassifier

```
DecisionTreeClassifier(
    class_weight=None      # label별 가중치 [[None, 'balanced'], [{class_label:weight}]]
  , criterion='gini'       # split 기준 - ['gini', 'entropy']
  , max_depth=None         # tree의 최대 깊이
  , max_features=None      # best split 특징 수 - [[1, 2,...], [0.1, 0.2, ...], [None, 'auto', 'sqrt', 'log2']]
  , max_leaf_nodes=None    # 최대 leaf node 수
  , min_impurity_split=1e-07   # 불순도 결정 임계치
  , min_samples_leaf=1     # Leaf의 최소 Sample 수
  , min_samples_split=2    # split 최소 수
  , min_weight_fraction_leaf=0.0 # 전체 가중 합 중 leaf의 최소 가중치 비율
  , presort=False          # 데이터의 사전 정렬 여부
  , random_state=0         # random number 생성 seed 값
  , splitter='best'        # Node split 선택 전략 - [best, random]
  )
```

Decision Tree

Library

- sklearn.tree.DecisionTreeRegressor

```
DecisionTreeRegressor(
    criterion='mse'        # split 기준 - ['mse', 'friedman_mse', 'mae']
  , max_depth=None      # 최대 깊이
  , max_features=None       # best split 특징 수 - [[1, 2,...], [0.1, 0.2, ...], [None, 'auto', 'sqrt', 'log2']]
  , max_leaf_nodes=None     # 최대 leaf node 수
  , min_impurity_split=1e-07    # 불순도 결정 임계치
  , min_samples_leaf=1      # Leaf의 최소 Sample 수
  , min_samples_split=2       # split 최소 수
  , min_weight_fraction_leaf=0.0  # 전체 가중 합 중 leaf의 최소 가중치 비율
  , presort=False        # 데이터의 사전 정렬 여부
  , random_state=None       # random number 생성 seed 값
  , splitter='best'         # Node split 선택 전략 - [best, random]
  )
```

❑ Tree-based model

❖ 장점

- 사용하기 쉽고 이해하기 쉽다 / 모델에 대한 이해도가 높다
- 변수 선택과 축소가 알고리즘 내에서 이루어 진다
- 통계 모델의 가정이 필요 없다
- 결측치를 정밀하게 다루지 않아도 사용 가능하다

❖ 단점

- 많은 데이터들은 수평 또는 수직으로 나누기 어렵다
- 변수 사이의 상호작용을 알아내기 어렵다