



IT와 비즈니스혁신

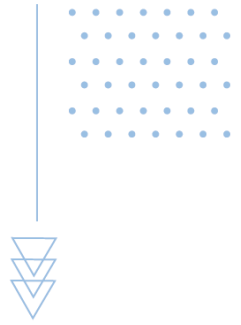
W7. 마이닝 기법 I: 분류 (의사결정나무)



Contents

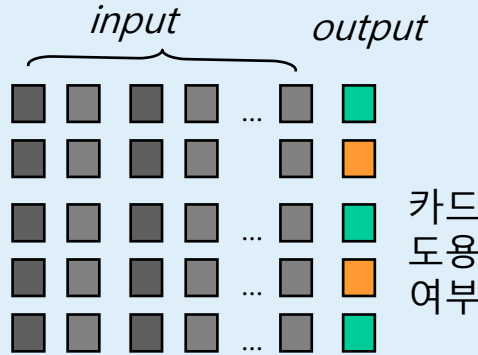
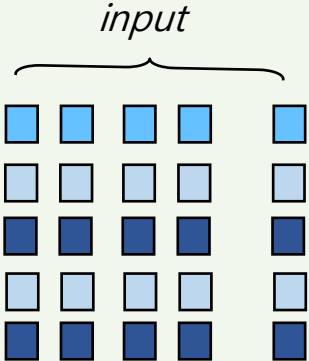
- I. 의사결정나무 개요
- II. 의사결정나무 기법원리
- III. 정리

* 출처: V.Kotu & B.Deshpande. 권영옥 외 공역, 데이터 과학 - RapidMiner를 활용한 데이터 마이닝, 한빛미디어, 2016
Gordon S. Linoff & Michael J. A. Berry, 김종우, 김선태 역, 경영을 위한 데이터마이닝- 마케팅과 CRM 활용을 중심으로, 한경사, 2018
F.Provost & T.Fawcett, 강권학 역, 비즈니스를 위한 데이터 과학 : 빅데이터를 바라보는 데이터 마이닝과 분석적 사고, 한빛미디어, 2014
A.Mueller & S.Guido, 박해선 역, 파이썬 라이브러리를 활용한 머신러닝, 한빛미디어, 2019



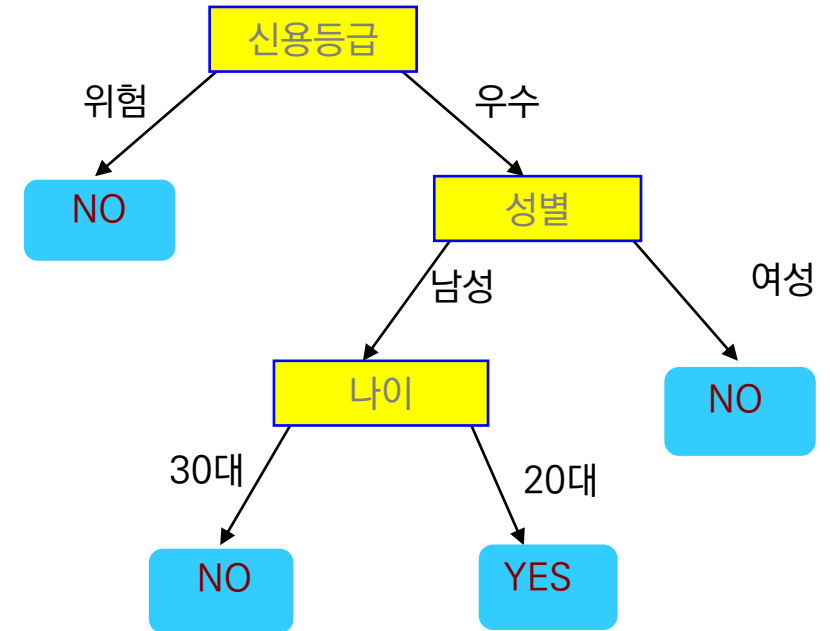
1. 의사결정나무 개요

데이터 마이닝은 크게 **출력 변수**의 존재 여부에 따라 **지도학습과 자율학습**으로 나눌 수 있음

	지도학습 (supervised learning)	자율학습 (unsupervised learning)
의미	<ul style="list-style-type: none"> 입력 데이터와 정답(Label)을 제공 받아 이를 통해 입력(독립)과 출력 (Label, 종속,타겟) 으로 매칭할 수 있는 규칙 생성 <p>예. 카드번호, 성별, 나이,거래 내역 등 →  카드도용 여부</p>	<ul style="list-style-type: none"> 외부에서 정답(Label)이 주어지지 않음 입력 데이터에서 패턴을 찾아내는 작업 <p>예. 군집화: 주어진 데이터를 3 개의 그룹으로 나눔 </p>
특징	출력 변수가 존재함	출력 변수가 존재하지 않음
분석 기법	<ul style="list-style-type: none"> 의사결정나무, 회귀분석, 인공신경망, 판별분석 등 	<ul style="list-style-type: none"> 군집분석, 연관성 분석 등

■ 분류와 예측에 효과적인 지도학습 기법

- 의사결정 규칙을 나무 구조로 도식화하여 분류 또는 예측을 지원하는 지도학습 기법
- 분류 또는 예측의 규칙을 이해하기 쉬운 문장으로 설명 가능
 - 예. **신용등급이 우수이면서 성별이 여성인 경우 비구매**
- 입력 변수들을 반복적으로 분할함으로써 규칙을 생성
 - 분석에 중요한 변수들(예. **신용등급, 성별**)이 규칙에 이용됨
- 데이터 준비 과정이 용이
 - 정규화와 같은 데이터 변환 과정이 불필요
 - 수치형과 범주형 변수 모두 사용 가능
 - 선형 뿐 아니라 비선형 관계 분석도 가능
- 수치형 값의 예측도 가능하나 (회귀 나무, regression tree)
 - 예측력이 높지 않아 주로 분류 업무에 이용

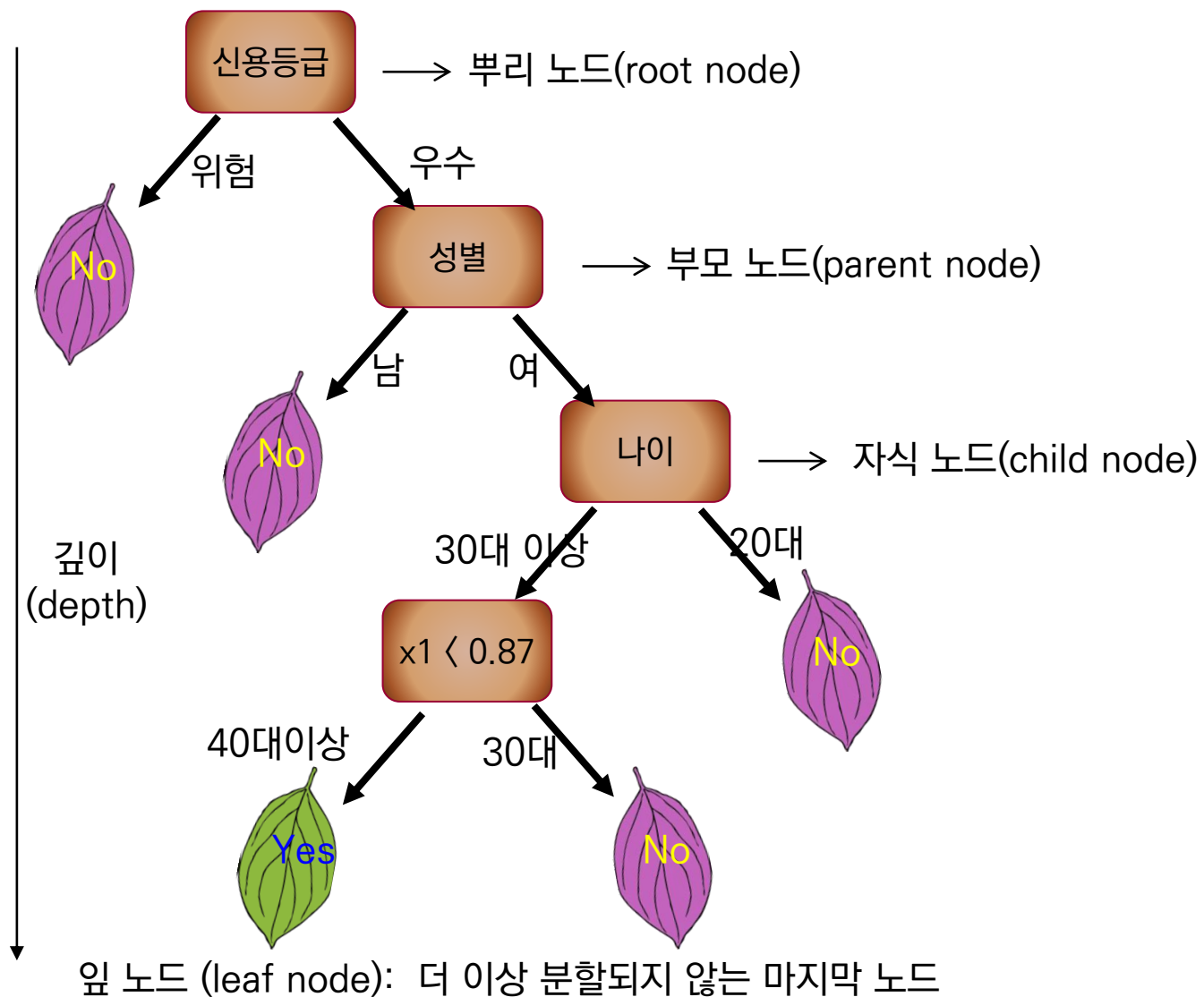


Model: Decision Tree

■ 노드(node)로 구성되는 나무 구조

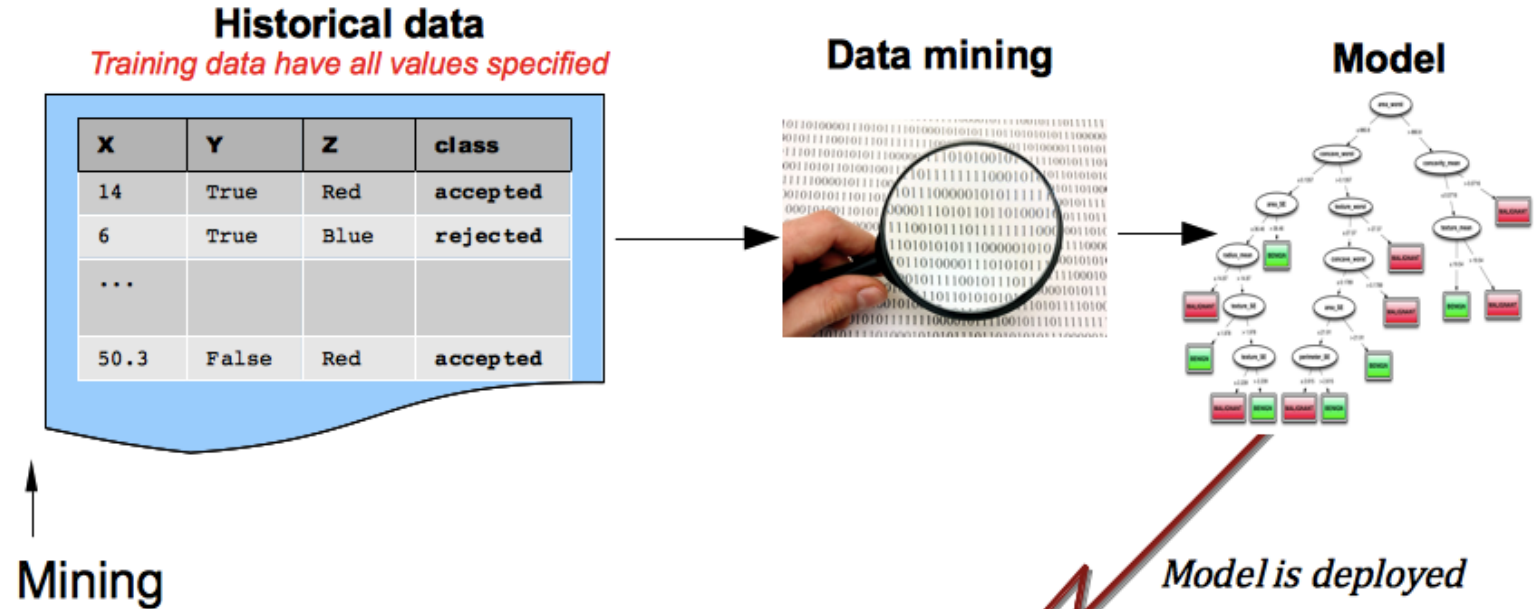
- 노드: 데이터의 전체 혹은 일부 집합
- 가지(branch): 하나의 노드에서 잎노드까지 연결된 일련의 노드들을 의미하며, 노드의 개수를 깊이(depth)라고함
- 자식 노드 수에 따라 Binary, Ternary Decision Tree
- 가지 분할 (splitting): 특정 규칙에 따라 노드를 나누는 것으로 분할을 통해 나무를 생성

*나무의 깊이: 4



마이닝 단계:

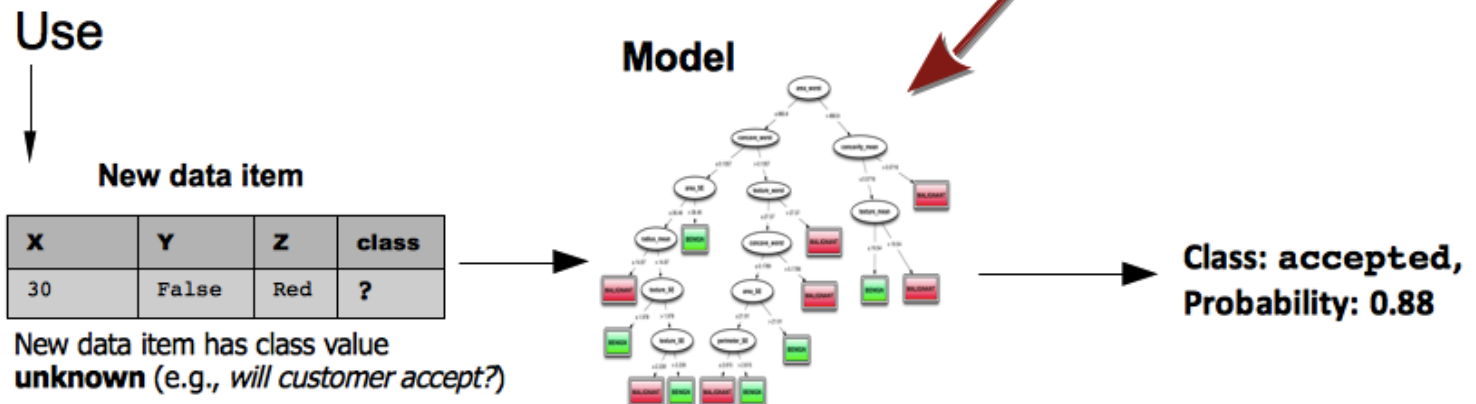
축적된 과거 데이터를 (학습용 데이터, training data)를 이용하여 최적의 의사결정나무 모델 생성



모델 사용 단계:

신규 고객 데이터 (시험용 데이터, test data)를 모델에 적용하여 분류 수행

예. 88%의 확률로 accepted로 분류



기존 고객 데이터를 이용하여 의사결정나무 모델 생성

타겟/종속변수/Label

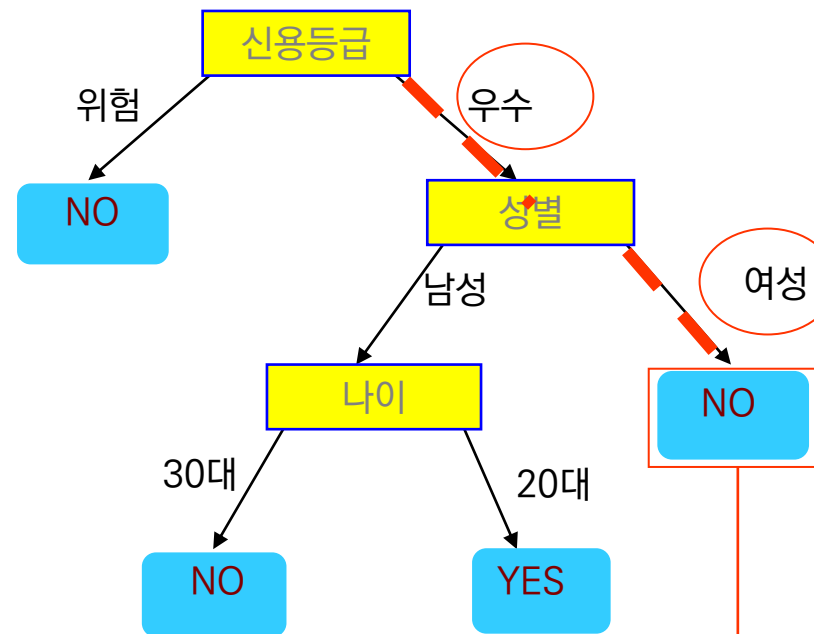
고객id	신용등급	성별	나이	신상품 구매
1	위험	남	31	No
2	우수	여	35	No
3	우수	남	37	No
4	위험	여	40	No
5	우수	남	23	Yes
6	우수	여	24	No
7	위험	남	38	No
8	우수	남	22	Yes
9	우수	여	23	No
10	우수	남	26	Yes

학습용 데이터(training data)

고객id	신용등급	성별	나이	신상품 구매
11	우수	여	33	?

시험용 데이터(test data)

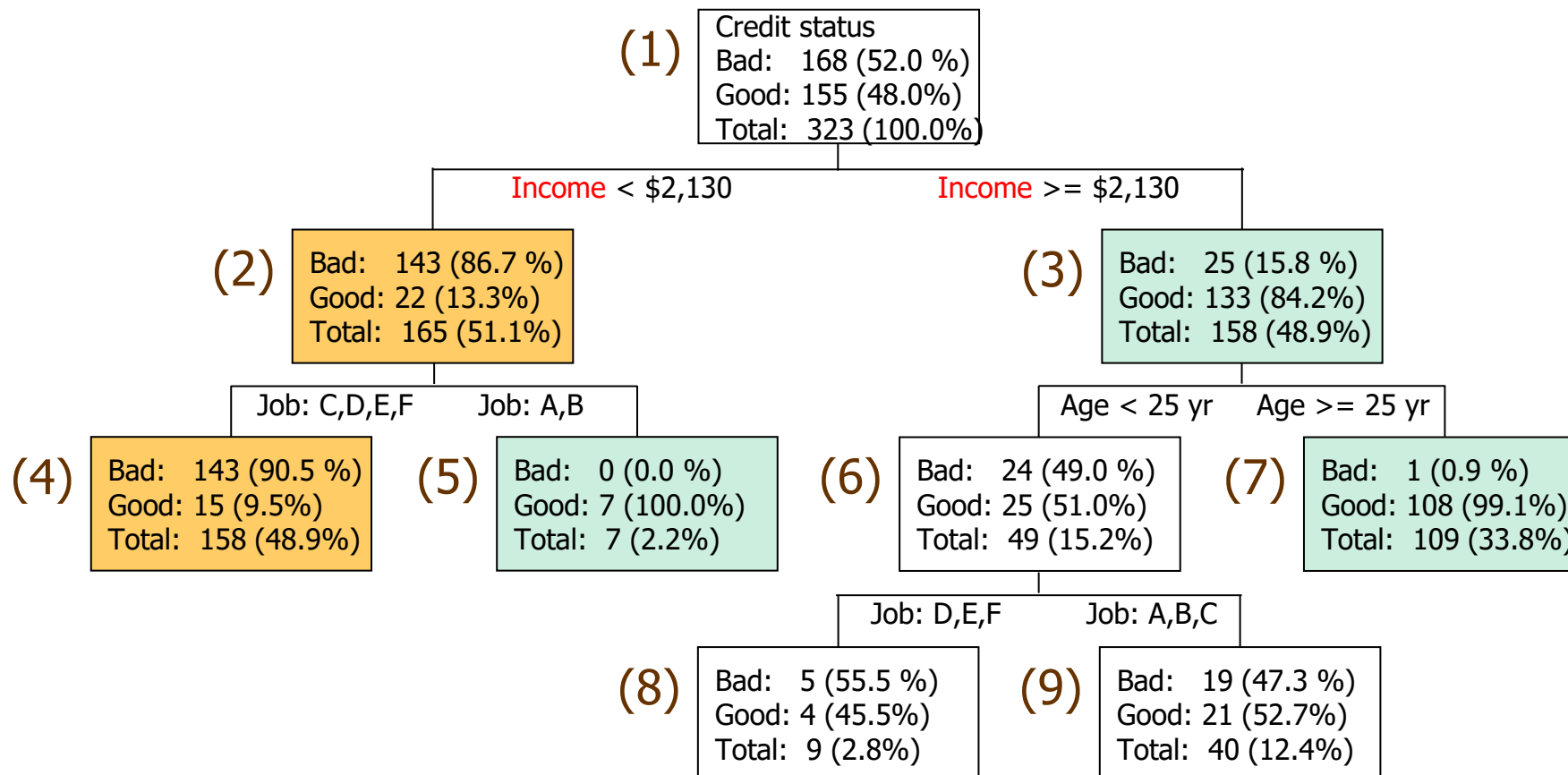
모델을 이용하여 신규 고객의 신상품 구매 예측

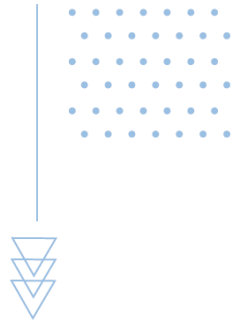


Model: Decision Tree

소득수준(Income), 직업 (Job), 나이(Age)에 따라 신용등급(Good, Bad) 예측

*확률 50% 이상인 등급으로 예측

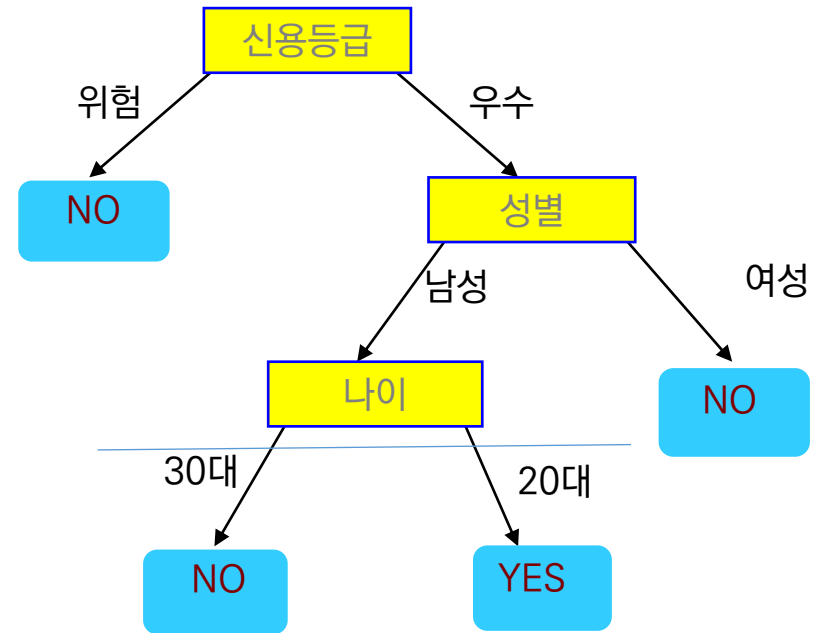




II. 의사결정나무 기법원리

의사결정나무를 만들기 위한 구성요소

- 1) 분할 규칙 (splitting rule): 노드를 나누는 규칙
예. 신용등급, 성별, 나이 등의 기준으로 분할
- 2) 정지 규칙 (stopping rule): 분할을 언제 그만둘 것인지를 결정하는 규칙
예. 신용등급이 위험인 경우에는 더 이상 분할되지 않음
- 3) 가지치기 (pruning): 나무의 크기가 클 때 가지를 제거하여 나무를 축소시키는 방법
예. 모델의 정확도가 떨어진다면 분할된 가지를 삭제



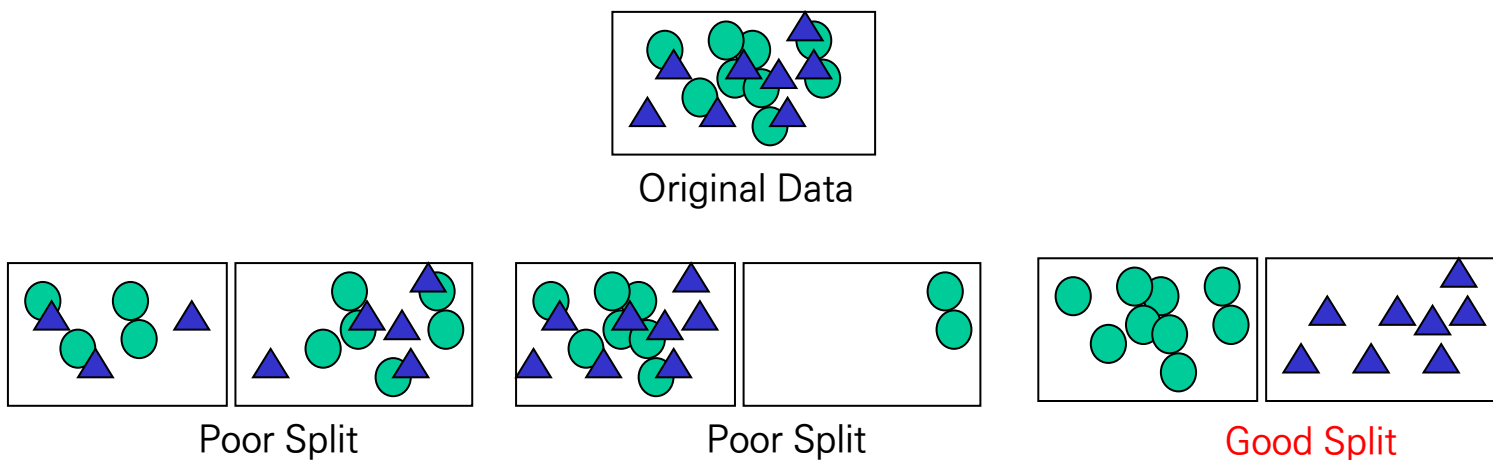
Model: Decision Tree

가장 분할을 통한 나무의 성장

- 각 노드에서 최적의 분할 규칙을 찾아서 나무를 성장시킴
- 타겟변수 측면에서 부모 노드보다 동질성 (homogeneity) 또는 순수도(purity)가 높은 자식 노드들이 되도록, 데이터를 반복적으로 더 작은 집단으로 분할

불순도(impurity)를 측정하는 방법

- 범주형 변수: 지니계수, 엔트로피, 정보이득, 카이제곱 통계량
- 연속형 변수: 분산의 감소량, 분산분석의 F 통계량



좋은 분할은 모든 자식노드의 순수도를 증가시킨다

분할 규칙 ① 지니계수 (Gini Index)

- 코라도 지니(Gini): 이탈리아의 통계학자이자 경제학자
- 소득의 불균형을 나타내는 지수
- 인구 다양성을 조사하는 생물학자들과 환경 공학자들이 자주 사용
- 같은 모집단에서 무작위로 선택된 두 항목들이 같은 클래스에 있을 확률
- 1에서 클래스의 비율의 제곱의 합을 뺀 값

$$G = 1 - \sum_k p_k^2$$

- 0 (불순도 최소, 순수)에서 0.5 (불순도 최대)의 값을 가짐
- 클래스의 비율이 10%/90%, 50%/50%로 나뉜 경우 지니계수 비교

$$1 - (0.1*0.1 + 0.9*0.9) = 1 - 0.82 = 0.18$$

$$1 - (0.5*0.5 + 0.5*0.5) = 1 - 0.5 = 0.5$$

*값이 작을수록 순수도가 증가

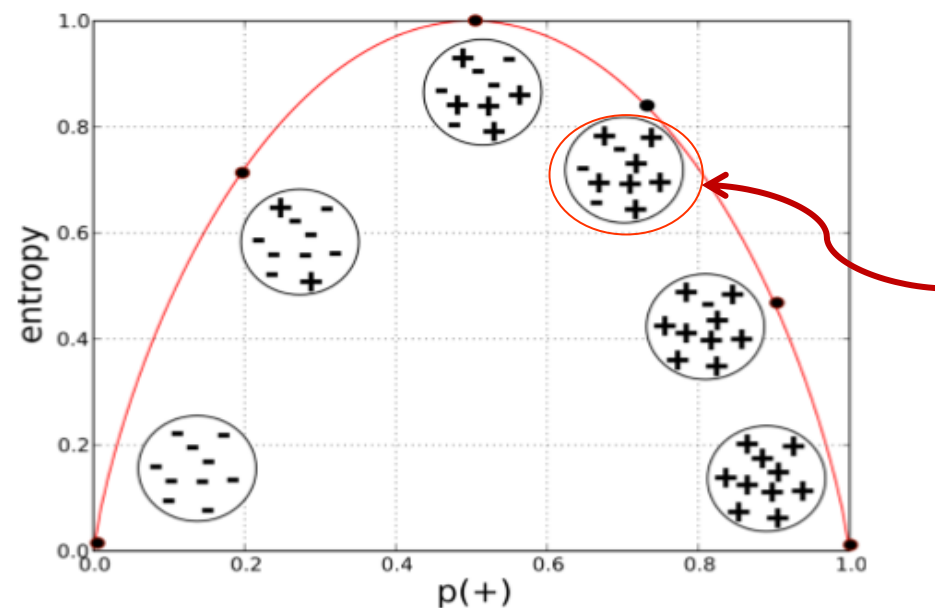
분할 규칙 ② 엔트로피 (Entropy)

- 무질서의 정도를 측정하는 척도
- 특정 의사결정나무 노드의 엔트로피
 - 노드에 포함된 모든 클래스에 대하여, 특정 클래스의 비율을 구하고 이 값과 이 값에 밑이 2인 로그를 취한 값을 곱한 값들의 합
 - 양수를 만들기 위해서 -1을 곱함

$$Entropy(H) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \dots = -\sum_k p_k \log_2 (p_k)$$

- E=0: 무질서 최소, 같은 항목으로만 구성 (순수)
- E=1: 무질서 최대, 각 항목이 동일하게 구성

*값이 작을수록 순수도가 증가



예. 고객 10명 중, 7명이 모기지 상환을 정상적으로 하고 3명이 상환하지 않은 경우

$$P(\text{정상상환}) = 7/10 = 0.7$$

$$P(\text{미상환}) = 3/10 = 0.3$$

$$Entropy = -0.7 \log_2 0.7 - 0.3 \log_2 0.3 = 0.88$$

분할 규칙 ② 엔트로피 (Entropy)

- 정보이득(Information Gain, IG)

IG = 분할 전의 엔트로피 - 분할 후의 엔트로피

$$= \text{Entropy}(\text{부모}) - [p(\text{자식1}) \times \text{Entropy}(\text{자식1}) + p(\text{자식2}) \times \text{Entropy}(\text{자식2}) + \dots]$$

- 추가된 정보(속성)에 따라 엔트로피 “변화” 를 의미 함
- 정보 증가량 값이 클수록 분류에 좋은 속성임

- 정보이득비율(Gain Ratio)

- 정보이득의 변형으로, 데이터가 많은 클래스를 선호하게 되는 편향성(bias)을 줄인 일반적으로 가장 좋은 옵션
- 분할하기 전에 가지들의 수를 고려함으로써 정보이득의 문제점을 해결
- 고유 정보량을 고려하여 정보이득을 수정함

분할 규칙 ② 엔트로피 (Entropy)

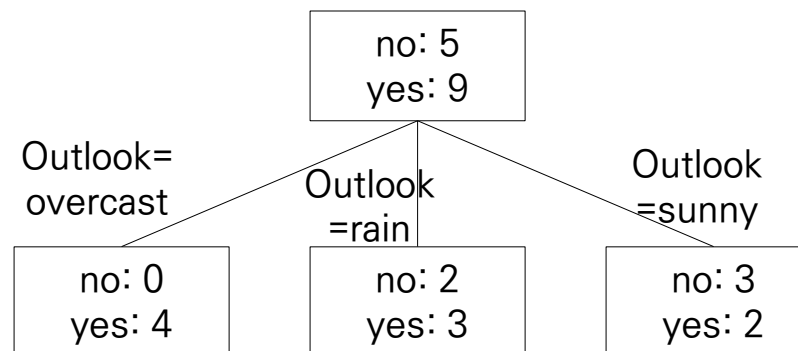
- 예. 골프경기 문제에서 Outlook (날씨 전망) 속성의 정보이득 계산

golf_train (학습용 데이터)

Outlook	Temperature	Humidity	Wind	Play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	78	false	yes
rain	70	96	false	yes
rain	68	80	false	yes
rain	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rain	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rain	71	80	true	no

분할 전

분할 후



분할 전 엔트로피

$$H_{no\ split} = -(5/14) \times \log_2(5/14) - (9/14) \times \log_2(9/14) = 0.940$$

분할 후 엔트로피

$$H_{Outlook:overcast} = -(0/4) \times \log_2(0/4) - (4/4) \times \log_2(4/4) = 0.0$$

$$H_{Outlook:rain} = -(2/5) \times \log_2(2/5) - (3/5) \times \log_2(3/5) = 0.971$$

$$H_{Outlook:sunny} = -(3/5) \times \log_2(3/5) - (2/5) \times \log_2(2/5) = 0.971$$

$$H_{Outlook} = P_{Outlook:overcast} \times H_{Outlook:overcast} + P_{Outlook:rain} \times H_{Outlook:rain} + P_{Outlook:sunny} \times H_{Outlook:sunny} = (4/14) \times (0) + (5/14) \times 0.971 + (5/14) \times 0.971 = 0.693$$

$$IG_{outlook} = H_{nosplit} - H_{outlook} = 0.940 - 0.693 = 0.247$$

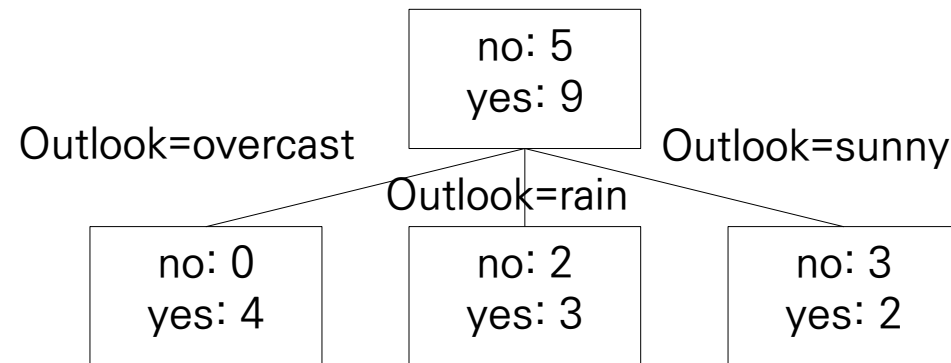
$$IG_{temper} = 0.029$$

$$IG_{humidity} = 0.102$$

$$IG_{wind} = 0.048$$

분할 규칙 ③ 카이제곱 통계량

- 1900년 영국의 통계학자 칼 피어슨(Karl Pearson)이 개발
- 빈도에 대한 기대값과 실제값의 표준화된 차이의 제곱들의 합으로 정의
- 기대값과 실제값의 차이가 클수록 변수와 클래스가 독립이라는 귀무가설을 기각 (변수가 클래스에 영향을 미침)



$$\chi^2 = (\text{카이 제곱 통계량}) = \sum_i \sum_j \frac{(E_{ij} - O_{ij})^2}{E_{ij}}$$

$$\begin{aligned} \chi^2 &= \frac{(1.429 - 0)^2}{1.429} + \frac{(2.571 - 4)^2}{2.571} \\ &+ \frac{(1.786 - 2)^2}{1.786} + \frac{(3.214 - 3)^2}{3.214} \\ &+ \frac{(1.786 - 3)^2}{1.786} + \frac{(3.214 - 2)^2}{3.214} = 3.547 \end{aligned}$$

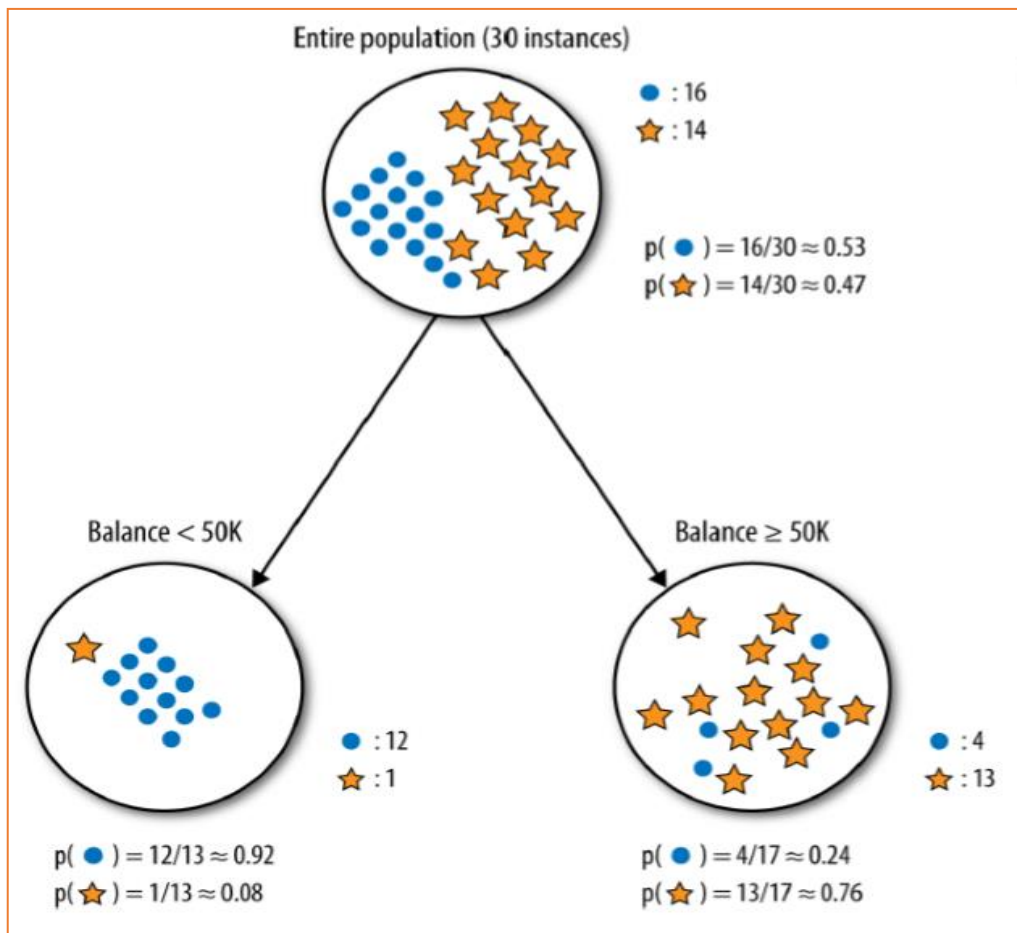
*값이 클수록 순수도가 증가

기대도수 (E _{ij})			
	no	yes	total
overcast	1.429*	2.571	4
rain	1.786	3.214	5
sunny	1.786	3.214	5
total	5	9	14

* $14 \times (4/14) \times (5/14) = 1.429$

실제도수 (O _{ij})			
	no	yes	total
overcast	0	4	4
rain	2	3	5
sunny	3	2	5
total	5	9	14

■ (연습문제) 아래 분할의 정보이득값을 구하시오.



Entropy(부모)

$$\begin{aligned}
 &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\
 &= -[0.53 \times (-0.9) + 0.47 \times (-1.1)] \\
 &= 0.99 \text{ (very impure!)}
 \end{aligned}$$

왼쪽자식: $entropy(Balance < 50K)$

$$\begin{aligned}
 &= -[p(\bullet) \log_2 p(\bullet) + p(\star) \log_2 p(\star)] \\
 &= -[0.92 \times (-0.12) + 0.08 \times (-3.7)] \\
 &= 0.39
 \end{aligned}$$

오른쪽자식: $entropy(Balance \geq 50K)$

$$\begin{aligned}
 &= -[p(\bullet) \log_2 p(\bullet) + p(\star) \log_2 p(\star)] \\
 &= -[0.24 \times (-2.1) + 0.76 \times (-0.39)] \\
 &= 0.79
 \end{aligned}$$

$$\begin{aligned}
 IG &= entropy(\text{부모}) - [p(\text{Balance} < 50K) \times entropy(\text{Balance} < 50K) + p(\text{Balance} \geq 50K) \times entropy(\text{Balance} \geq 50K)] \\
 &= 0.99 - [13/30 \times 0.39 + 17/30 \times 0.79] = 0.37
 \end{aligned}$$

*IG값이 클수록 좋은 분할

■ 의사결정나무 알고리즘과 분할 규칙

- CART (Classification and Regression Trees)

- 1984년 L. Breiman, J. Friedman, R. Olshen, C. Stone에 의해서 발표
- 이진 나무를 만든 후 순수도를 증가시킬 수 있도록 계속 분할
- 순수도: 범주형인 경우 **지니지수**, 연속형인 경우 분산 이용

- C5.0

- 오스트레일리아 연구자 J. Ross Quinlan에 의해서 개발
- 초기 버전은 1986년에 개발된 ID3 (Iterative Dichotomiser 3)
- CART와 다르게 범주형 변수에 대한 다중 분할 가능
- 순수도: **엔트로피 지수** 이용

- CHAID (Chi-squared Automatic Interaction Detection)

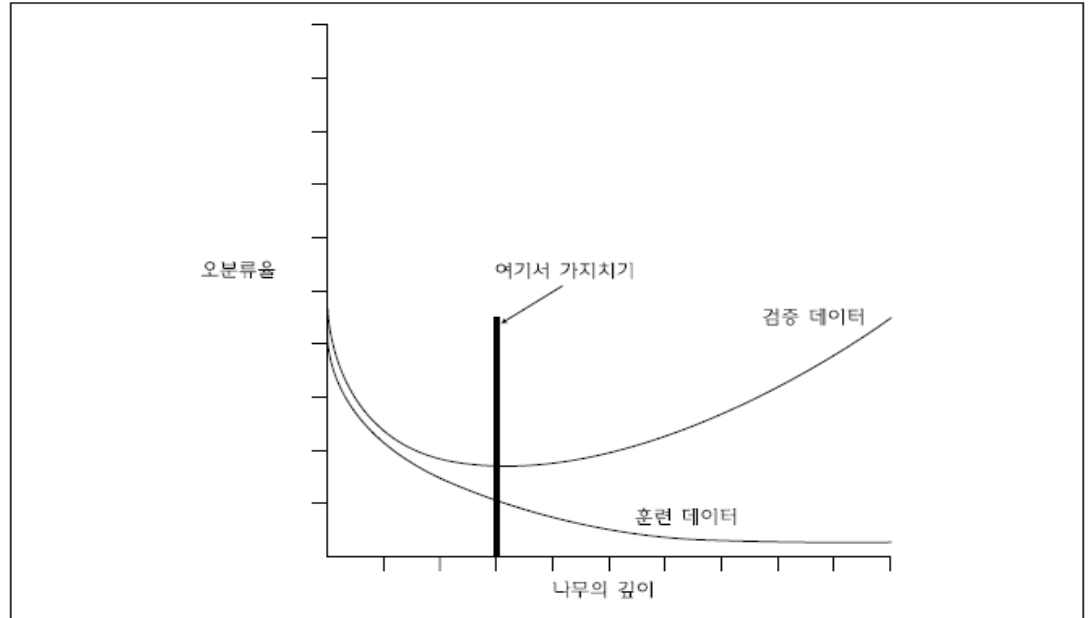
- 1975년 John A. Hartigan에 의해서 발표, 변수들 간의 통계적 관계를 감지
- 고전적인 CHAID 알고리즘에서는 모든 입력변수가 범주형
- 연속형 변수들은 구간화(binning)하거나 대, 중, 소와 같은 순차적인 클래스로 대체
- 순수도: **카이제곱 통계량** 이용

더 이 분할이 일어나지 못하도록 하는 규칙

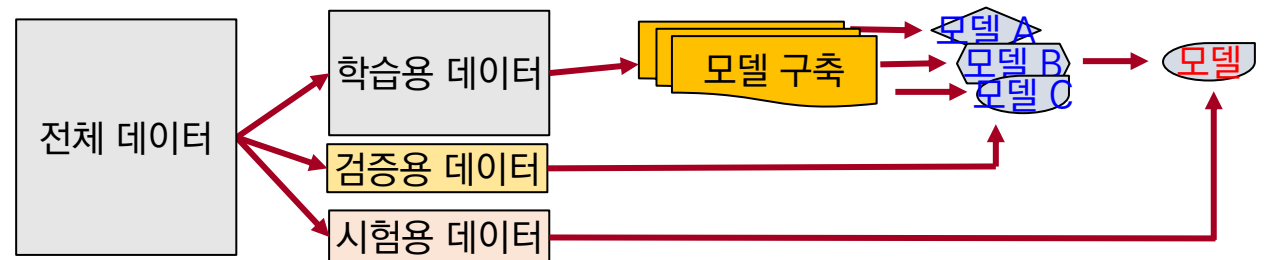
- 실제 데이터에서는 100% 순수한 잎 노드 (마지막 노드)를 얻는 경우는 거의 없으므로 언제 분할을 멈추어야 할지를 결정해야 함
- 현재의 마디에서 더 이상 분할이 일어나지 못하게 하는 규칙
 - **minimal gain**
정보이득(불순도의 감소량)의 최소 임계치를 충족하는 속성이 하나도 없는 경우
 - **maximal depth** (scikit-learn의 DecisionTreeClassifier에서는 max_depth)
나무가 최대 깊이에 도달한 경우
의사결정나무가 커질수록 결과해석이 어려워질 뿐 아니라 과적합의 문제가 생김
 - **minimal size for split** (scikit-learn의 DecisionTreeClassifier에서는 min_samples_split)
분할 후 노드에 속하는 사례수가 특정 수 이하인 경우
과적합을 막기 위한 메커니즘

■ 가지치기(pruning)가 필요한 경우

- 지나치게 많은 노드를 가지는 (복잡한 모델의) 의사결정 나무는 새로운 데이터에 적용할 때 오차가 매우 클 가능성이 있음
- 성장이 끝난 나무의 가지를 제거하여 적당한 크기의 나무 모델을 최종 모델로 선택하는 것이 정확도 향상에 도움이 됨
- 적당한 크기를 결정하는 방법은 검증용 (validation) 데이터를 사용하여 오분류율을 구하고 이 오분류율이 가장 작은 모델을 선택



- 학습용 (training) 데이터: 모델을 구축하기 위해 사용되는 데이터
- 검증용 (validation) 데이터: 모델이 얼마나 잘 구축되었는지 평가하고 구축된 모델들 중에서 가장 좋은 것을 선택하기 위해 사용되는 데이터
- 시험용 (test) 데이터: 최종 선택된 모델을 새로운 데이터에 적용하여 정확성을 평가하기 위해 사용되는 데이터



과적합 (overfitting) 문제 해결

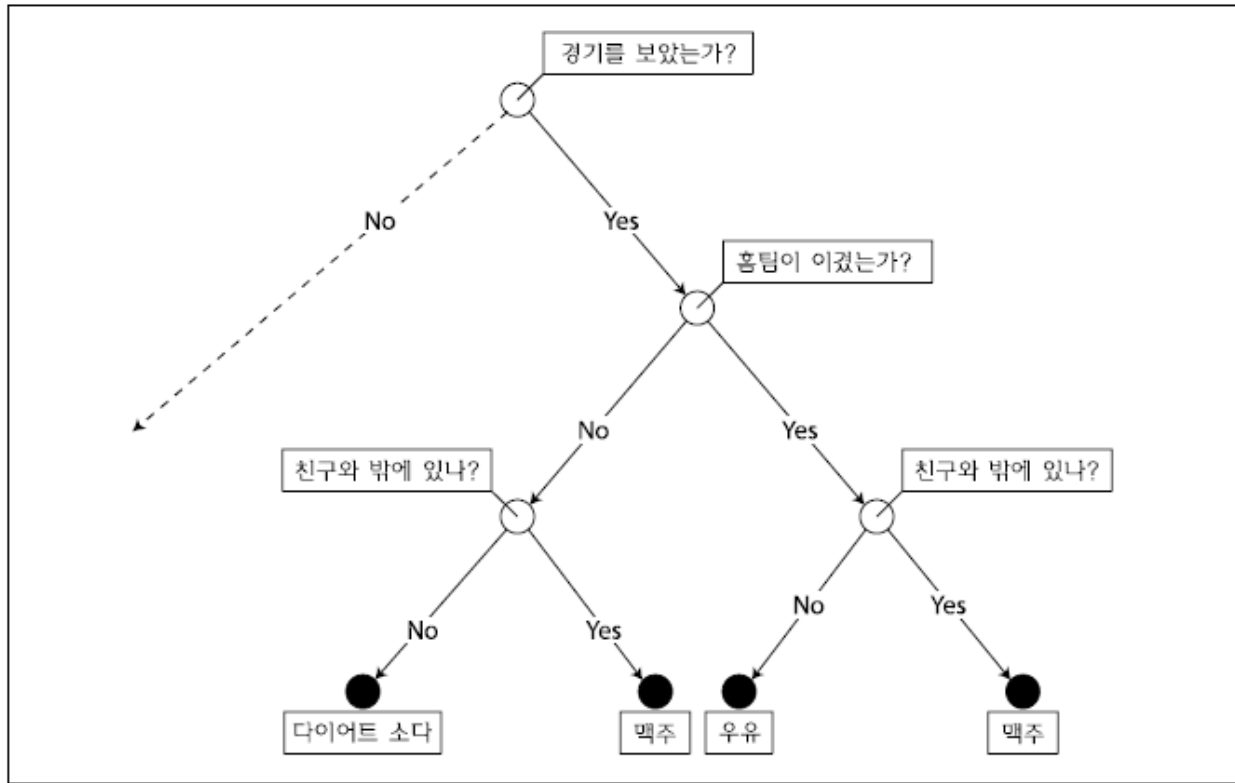
- 과적합이란 모델이 학습용 데이터에 대해서만 성능이 좋고, 새로운 데이터에 적용했을 경우에는 성능이 좋지 않은 경우를 의미
- 과적합을 방지하기 위해 가지치기 등을 통해 의사결정나무 성장을 제한하거나 줄여야 할 필요가 있음

사전, 사후 가지치기

- 사전 가지치기 (pre-pruning) *scikit-learn은 사전 가지치기만 지원
 - 의사결정나무가 성장하기 전에 가지치기를 하거나 성장하는 중간에 가지치기를 하는 방법
- 사후 가지치기(post-pruning)
 - 가지의 개수나 나무의 깊이를 데이터가 허용하는 데까지 제한하지 않고 나무를 생성한 후 나중에 분류 오차율에 영향을 주지 않는 가지를 치는 방법
 - 의사결정나무의 깊이를 최대로 하므로, 속성값들과 클래스들 간의 작지만 잠재적으로 의미 있는 관계 파악 가능
 - 단점은 역 가지치기를 하기 위해 추가적인 계산 필요

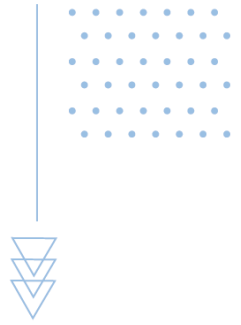
■ 의사결정나무의 결과는 규칙들의 집합

- 중복되는 규칙 제거
- 이해 하기 좋은 부분 집합으로 줄이기



경기를 보았는데, 친구와 밖에 있었던 경우에는 맥주를 마심

- 경기를 보았는데, 홈팀이 이기고 친구와 밖에 있었던 경우는 맥주를 마심
- 경기를 보았는데, 홈팀이 지고 집에 있었던 경우에는 다이어트 소다를 마심
- 경기를 보았는데, 홈팀이 지고 친구와 밖에 있었던 경우는 맥주를 마심
- 경기를 보았는데, 홈팀이 이기고 집에 있었던 경우에는 우유를 마심



III. 정리

□ 의사결정나무의 개요

- 의사결정 규칙을 나무 구조로 도식화하여 분류 또는 예측을 지원하는 지도학습 기법
- 생성된 규칙을 이해하기 쉬움
- 수치형과 범주형 변수를 모두 사용 가능
- 모형의 가정 (선형성 등)이 필요없으며 전처리 과정이 용이
- 나무가 너무 깊은 경우 예측력의 저하 뿐 아니라 해석도 어려움

□ 의사결정나무의 기법 원리

- 의사결정나무는 최적의 분할규칙 (지니계수, 정보이득, 카이제곱 통계량)을 이용하여 순수도를 높이는 방향으로 가지 분할
- 오분류율이 너무 증가하지 않도록 나무의 크기를 조절하기 위해서 분할정지 및 가지치기 필요