



11주차: 분류를 위한 모델

ChulSoo Park

School of Computer Engineering & Information Technology
Korea National University of Transportation



학습목표 (11주차)

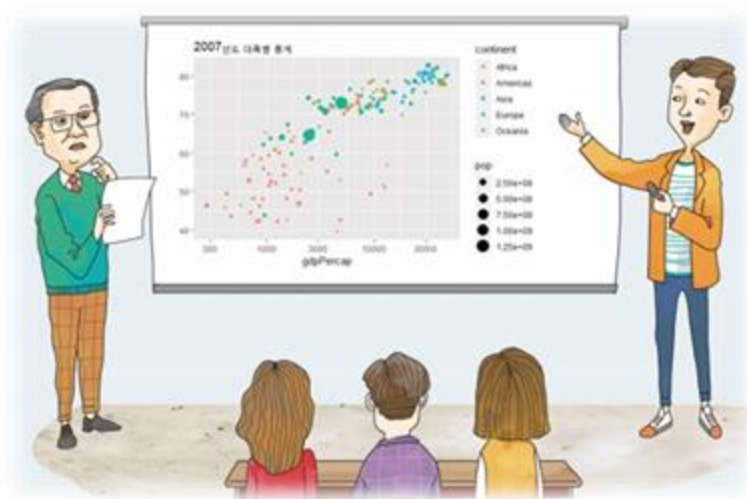
- ❖ 회귀와 분류 문제의 이해
- ❖ 결정 트리의 이해
- ❖ 랜덤 포리스트 개념 이해
- ❖ SVM과 k-NN의 이해
- ❖ 분류 모델의 적용 학습



09

CHAPTER

분류를 위한 모델



CONTENTS

- 9.1 회귀와 분류
- 9.2 결정 트리의 원리
- 9.3 결정 트리 함수의 사용
- 9.4 결정 트리의 해석
- 9.5 랜덤 포리스트
- 9.6 SVM과 k-NN
- 9.7 분류 모델의 다양한 적용 요약



■ 모델링



- 현실 세계에서 일어나는 현상을 수학적식으로 표현하는 행위
- 모델링 핵심 : 모델과 예측

■ 선형으로 표현하기 부적절한 데이터가 다수

- 예) 해수욕장: 기온 x 와 방문객 수 y
 - ✓ 선형 모델 $y=1000x+200$ 을 사용하면,
 - ✓ x (기온)가 음수가 되면 방문객 수는 음수? (겨울 해수욕장 인파 ?)
 - ✓ 작은 해수욕장에는 적용 불가능 → 일반성이 매우 약한 모델
- 일반화 선형 모델과 glm(generalized linear model) 함수
- 범주형 변수 다루기
- glm으로 구한 모델의 통계량 해석



- 머플러 판매 데이터에 일반 선형 모델인 glm 함수 적용
 - 이전과 달라진 것은 lm이 glm이 되고, family=binomial 옵션을 추가한 것
 - binomial 옵션은 반응 변수인 profit이 두 가지 값만 가진다고 glm에게 알려주는 역할

```
Console C:/Rsources/    
> muffler=data.frame(discount=c(2.0, 4.0, 6.0, 8.0, 10.0),profit=c(0,0,0,1,1))  
> muffler  
  discount profit  
1         2      0  
2         4      0  
3         6      0  
4         8      1  
5        10      1  
> rest_glm=glm(profit~discount, data=muffler, family = binomial)  
경고메시지(들):  
glm.fit: 적합된 확률값들이 0 또는 1 입니다  
> coef(rest_glm)  
(Intercept)    discount  
-160.80782     22.98592  
> fitted(rest_glm)  
           1           2           3           4           5  
2.220446e-16 2.220446e-16 1.142877e-10 1.000000e+00 1.000000e+00  
> residuals(rest_glm)  
           1           2           3           4           5  
-2.107342e-08 -2.107342e-08 -1.511871e-05 1.376758e-05 2.107342e-08  
> deviance(rest_glm)  
[1] 4.181229e-10
```



Review

- Haberman survival 읽어 들이고 확인 및 변경
 - survival data는 범주형이나 0과 1의 범주형으로 변경

Console C:/RSources/

```
> haberman=read.csv("c:/rdata/haberman.csv",header=FALSE)
> names(haberman)=c('age','op_year','no_nodes','survival')
> head(haberman)
```

	age	op_year	no_nodes	survival
1	30	64	1	1
2	30	62	3	1
3	30	65	0	1
4	31	59	2	1
5	31	65	4	1
6	33	58	10	1

```
> str(haberman)
```

```
'data.frame': 306 obs. of 4 variables:
 $ age      : int  30 30 30 31 31 33 33 34 34 34 ...
 $ op_year  : int  64 62 65 59 65 58 60 59 66 58 ...
 $ no_nodes : int  1 3 0 2 4 10 0 0 9 30 ...
 $ survival: int  1 1 1 1 1 1 1 2 2 1 ...
```

```
> haberman$survival=factor(haberman$survival)
```




```
> str(haberman)
```

```
'data.frame': 306 obs. of 4 variables:
 $ age      : int  30 30 30 31 31 33 33 34 34 34 ...
 $ op_year  : int  64 62 65 59 65 58 60 59 66 58 ...
 $ no_nodes : int  1 3 0 2 4 10 0 0 9 30 ...
 $ survival: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 2 2 1 ...
```



Review

- 일반화 선형 모델 적용(glm)
 - survival data는 범주형이나 0과 1의 범주형으로 변경

```
Console C:/RSources/     
> resh=glm(survival~age+op_year+no_nodes, data=haberman,family=binomial)  
> coef(resh)      # 계수를 계산하는 함수  
(Intercept)      age      op_year      no_nodes  
-1.86162525  0.01989935 -0.00978386  0.08844244  
> resh=glm(survival~., data=haberman,family=binomial)  
> coef(resh)      # 계수를 계산하는 함수  
(Intercept)      age      op_year      no_nodes  
-1.86162525  0.01989935 -0.00978386  0.08844244  
> deviance(resh)  # 잔차제곱  
[1] 328.2564
```

모델을 구했으니 새로운 환자가 오면 생존 여부를 예측할 수 있다.



분류

- 반응 변수가 몇 개의 값만(범주형 등) 가지는 경우
- 예) 1(환자)와 0(정상인) 또는 2(환자), 1(관찰 대상), 0(정상)
- 예) 이동통신사의 고객 관리
 - 고객을 3(최고 충성), 2(충성), 1(보통), 0(불평)으로 구분
 - 3번 부류에 속한 고객에게는 때때로 듣기 좋은 말, 0번 부류 고객에게는 요금 감면 등의 파격 혜택

공부할 분류 모델

- 결정 트리
- 랜덤 포리스트
- k -NN
- SVM

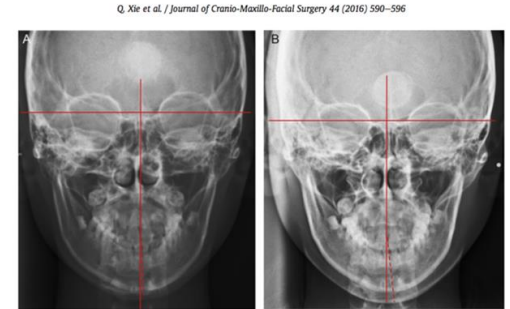
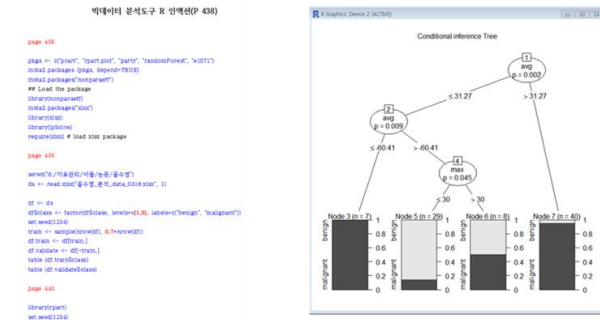
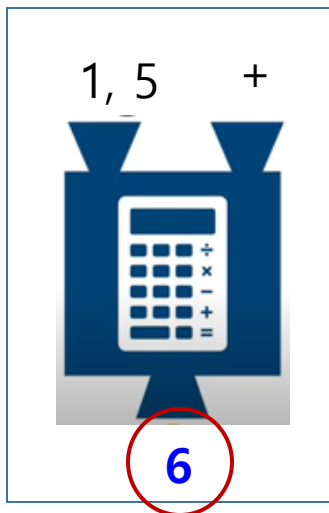


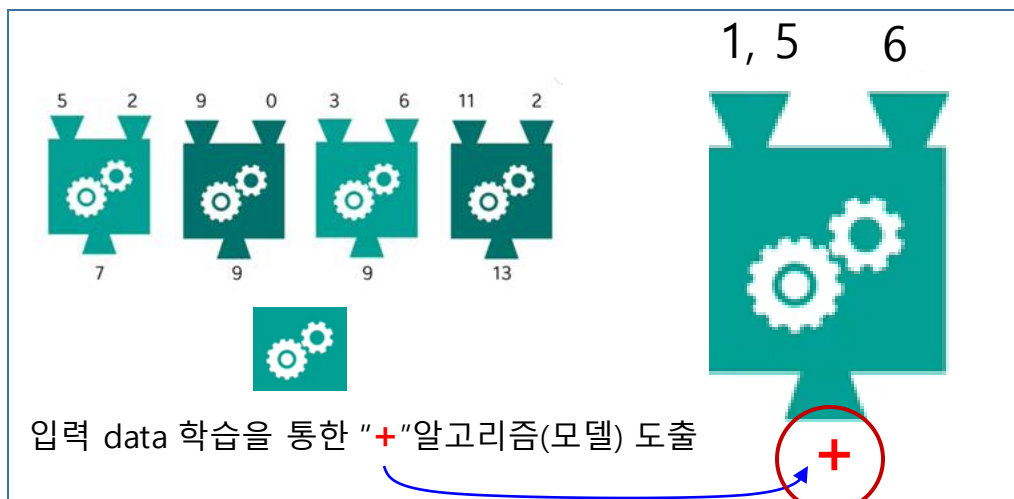
Fig. 6. Before and after PA image: A. first visit: no asymmetry was found, B. revisit: more than 5 mm skeletal asymmetry was noticed.



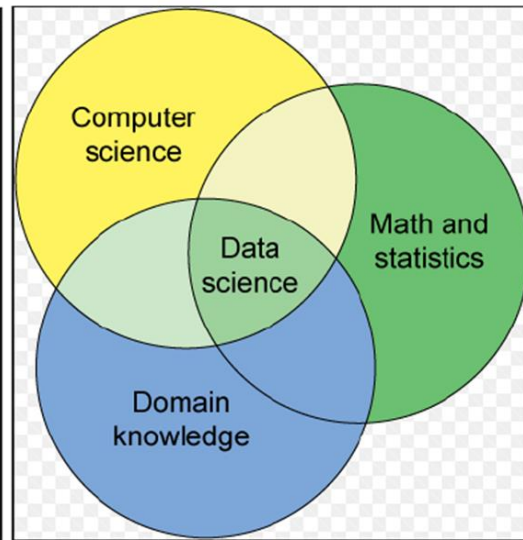
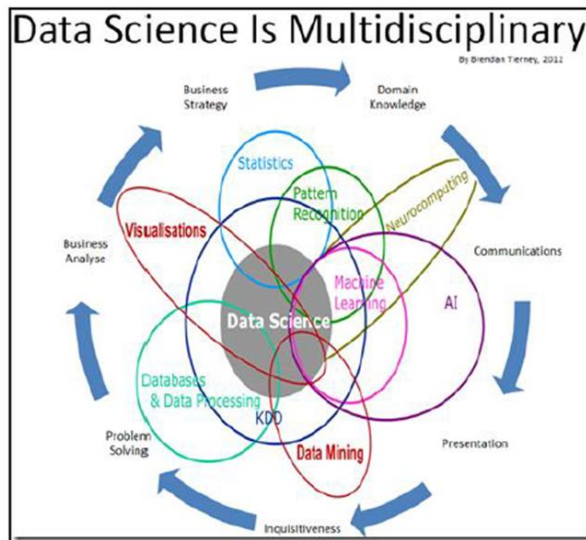
기존 컴퓨터 사이언스



인공지능(AI)

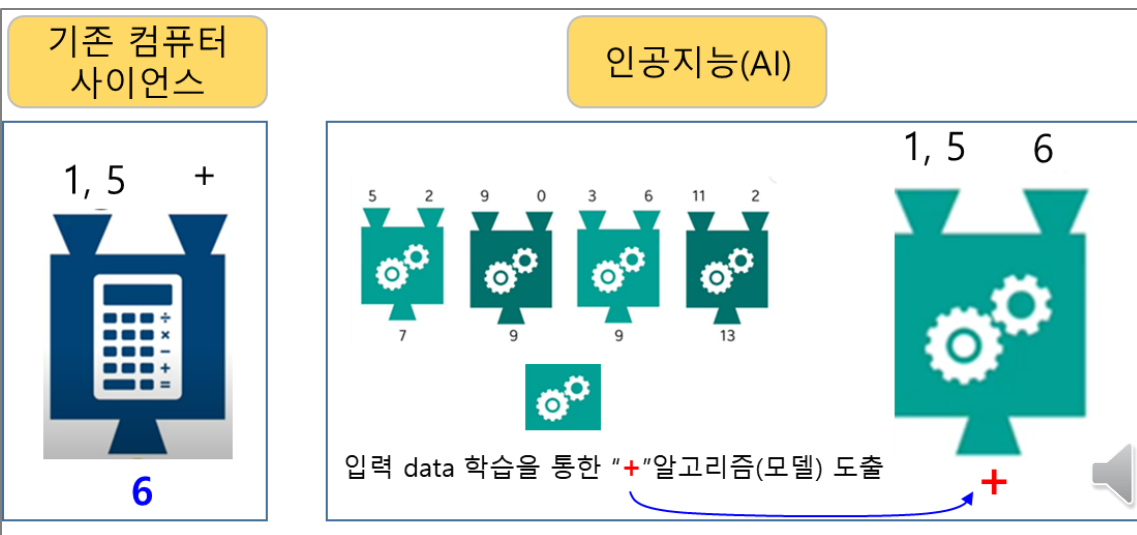


■ 데이터 사이언스

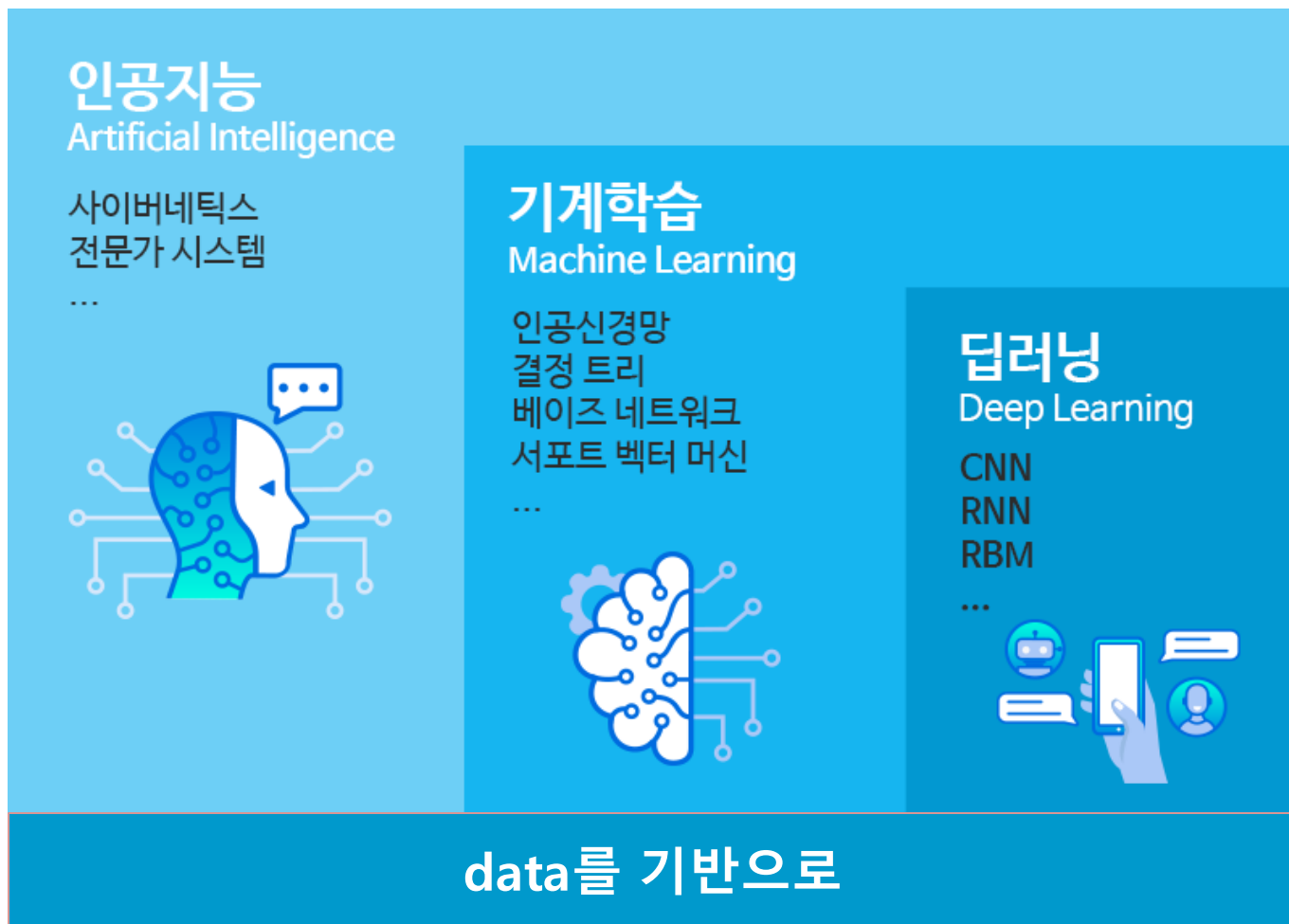


출처 : www.oralytics.com

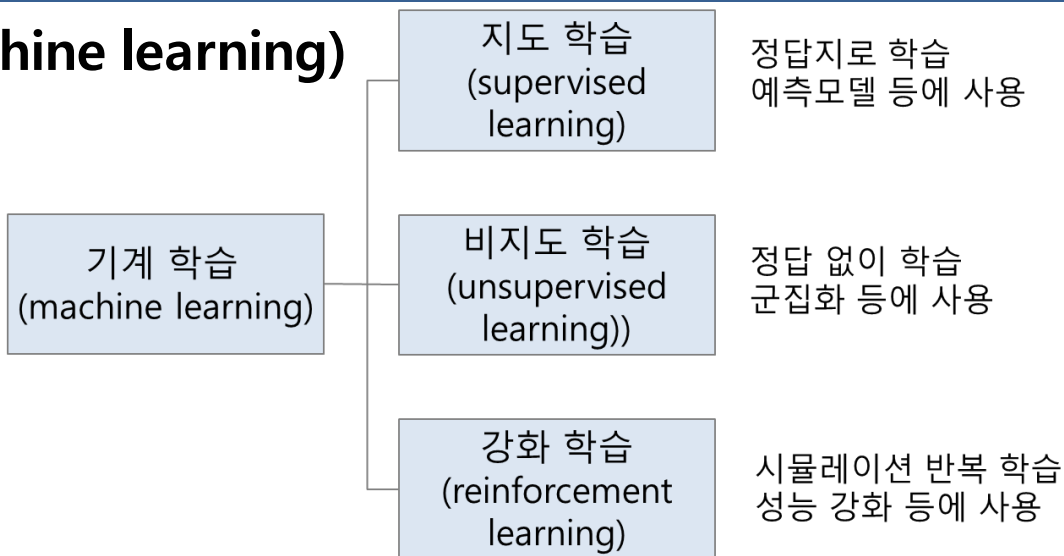
■ 인공지능(AI)



■ 인공지능(AI)



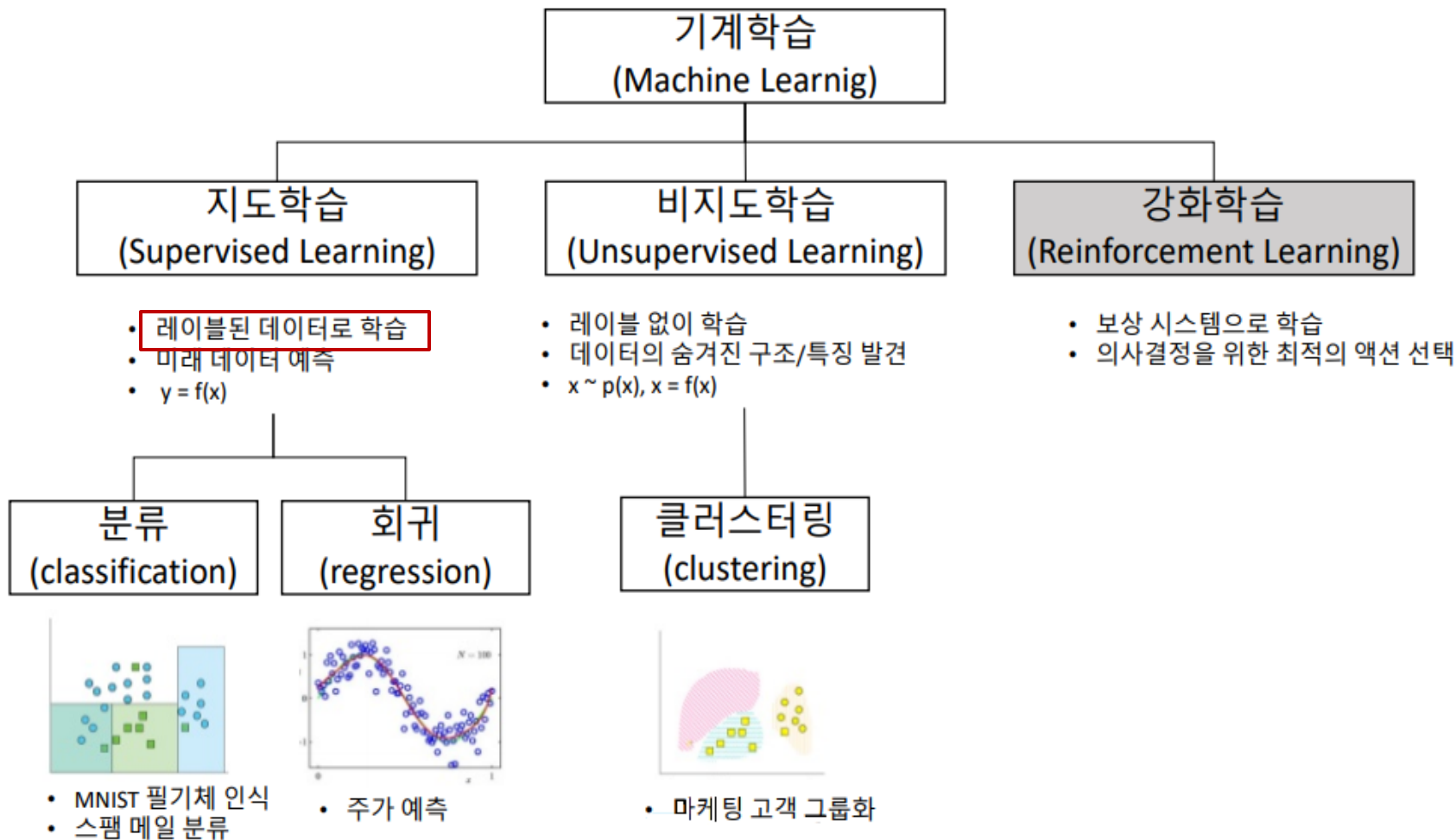
머신러닝(machine learning)



구분	기계학습 유형		대표 알고리즘
비지도 학습 (unsupervised learning)	서술형 (descriptive)	클러스터링	k-means
		연관 분석	패턴 분석
지도 학습 (supervised learning)	예측형 (prediction)	분류 예측	k-NN, 베이리어스, 의사결정 트리
		수치 예측	선형 회귀 분석, 회귀 트리, SVM



■ 지도학습(Supervised Learning)



■ 분류(의사결정트리) 사례 / 레이블

Q. Xie et al. / Journal of Cranio-Maxillo-Facial Surgery 44 (2016) 590–596

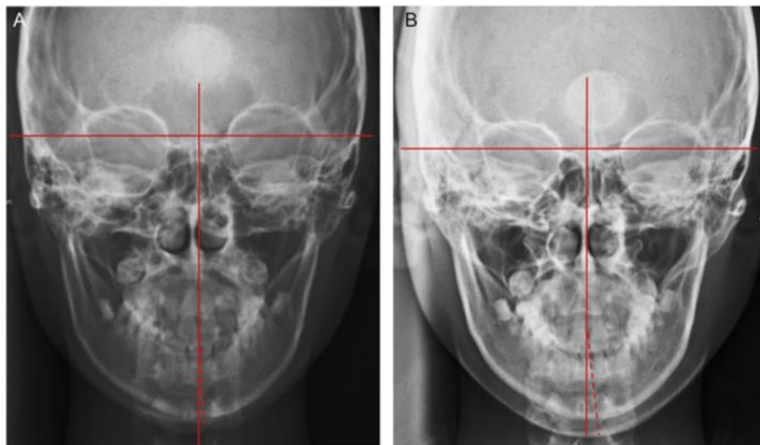


Fig. 6. Before and after PA image: A. first visit: no asymmetry was found, B. revisit: more than 5 mm skeletal asymmetry was noticed.

빅데이터 분석도구 R 언어선(P 438)

```

page 438

pkgs <- c("rpart", "rpart.plot", "party", "randomForest", "e1071")
install.packages(pkgs, depend=TRUE)
install.packages("nonparett")
## Load the package
library(nonparett)
install.packages("xlsx")
library(xlsx)
library(lpSolve)
require(xlsx) # load xlsx package

page 439

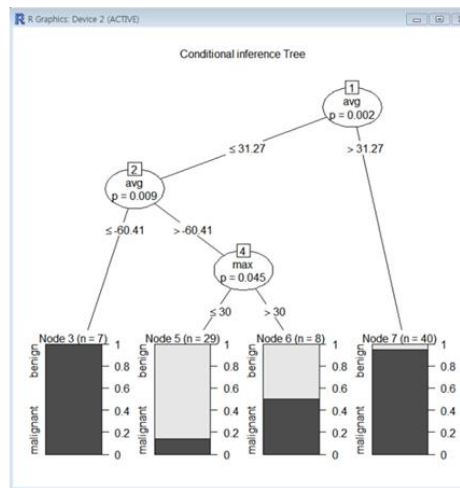
setwd("d:/자료관리/아동/논문/출수업")
ds <- read.xlsx("출수업_분류_data_0316.xlsx", 1)

df <- ds
df$class <- factor(df$class, levels=c(1,0), labels=c("benign", "malignant"))
set.seed(1234)
train <- sample(nrow(df), 0.7*nrow(df))
df.train <- df[train,]
df.validate <- df[-train,]
table(df.train$class)
table(df.validate$class)

page 443

library(rpart)
set.seed(1234)

```




9.1 회귀와 분류

- 회귀(regression)와 분류(classification)
 - 회귀는 반응 변수가 연속값(continuous value)을 가짐
 - 분류는 반응 변수가 이산값(discrete value)을 가짐

■ 몇 가지 사례

문제	데이터 사례	반응 변수
회귀	판매량에 따른 월급(7장)	월급(정수)
	cars(속도에 따른 제동 거리 data)	제동 거리(실수)
	Trees(나무의 직경과 키에 따른 목재 부피 data)	목재 부피(실수)
분류	UCLA admission (대학원 입시 데이터)	합격 여부(합격, 불합격)
	mnist(필기 숫자 데이터)	숫자 분류(0,1,2,3.....,9)
	Iris(붓꽃 data)	품종(3개)

부류 레이블 또는 줄여서 부류 또는 레이블, 라벨이라 부름 

9.1 회귀와 분류

■ 분류 문제를 푸는 모델

- 결정 트리(decision tree), 랜덤 포리스트(random forest), k -최근접 이웃 알고리즘(k -NN; K-Nearest Neighbor), SVM(support vector machine), 신경망(neural network), 딥 러닝(deep learning) 등
- 이 장은 결정 트리, 랜덤 포리스트, k -NN, SVM을 다룸

■ 회귀 모델과 분류 모델

- 회귀 모델인 glm에 family=binomial 옵션을 주면 로지스틱 회귀로 작동 (로지스틱 회귀는 분류가 두 개인 이진 분류 문제를 푼다)
- 이 장에서 공부할 분류 모델은 회귀 버전도 제공
- 애초 회귀 또는 분류 목적으로 개발된 모델은 다른 문제를 푸는 버전도 있음



9.2 결정 트리의 원리

■ 트리(tree) 정의

- 트리(tree)는 하나 이상의 노드 (node)로 구성된 유한 집합으로서 다음 두가지 조건을 만족한다.
 - 1) 특별히 지정된 노드(node)인 루트(root)가 있고
 - 2) 나머지 노드들은 다시 각각 트리 이면서 연결되지 않은 T_1 , $T_2 \dots, T_n$ ($N \geq 0$)으로 나누어진다, 이때 T_1 , $T_2 \dots T_n$ 을 루트의 서브 트리(subtree)라고 한다.

- 결정 트리는 스무고개와 비슷한 원리를 사용

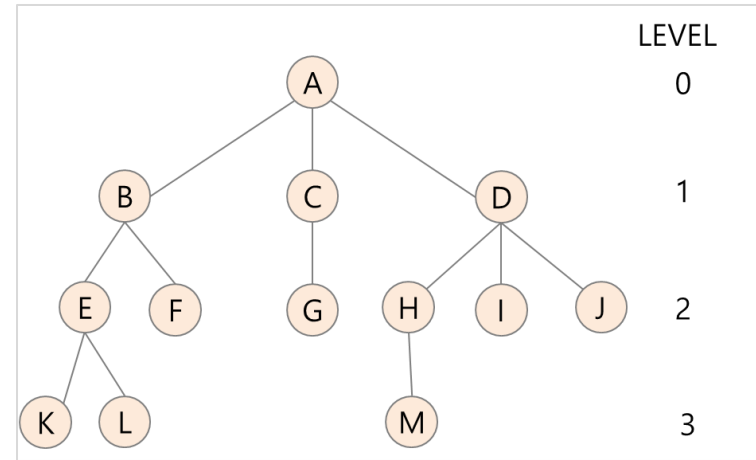
“예” 또는 “아니오”로 대답하는 문제



9.2 결정 트리의 원리

■ 트리(tree) 관련 용어

- ① **노드(node)** : 트리를 구성하는 개체(그래프 G에서는 A~M)
- ② **루트(root)** : 트리의 시작 노드로 통상 특정 노드를 지정함
(그래프G에서는 노드 A)
- ③ **차수(degree)** : 어떤 노드의 자식 노드의 개수(그래프 G에서
노드 B의 차수는2, D의 차수는 3)
- ④ **레벨(level)** : 루트를 레벨 0(또는 1)로 지정하고, 하위로 갈
수록 레벨 + 1,그림에서 오른 쪽에 레벨 표시
- ⑤ **잎 또는 단말 노드(leaf 또는 terminal node)** : 자식 노드를
갖지 않는 노드(그래프 G에서는 K, L, F, G, M, I, J)
- ⑥ **자식 노드(children node)** : 어떤 노드에 직접 연결된 하위
노드 (그래프 G에서 B의 자식 노드는 E, F)
- ⑦ **부모 노드(parent node)** : 자식 노드의 반대되는 개념으로
어떤 노드에 직접 연결된 상위 노드(그래프 G에서 B의 부모
노드는 A)
- ⑧ **형제 노드(sibling 또는 brother node)** : 동일한 부모 노드
를 갖는 노드(그래프 G에서 H, I, J)



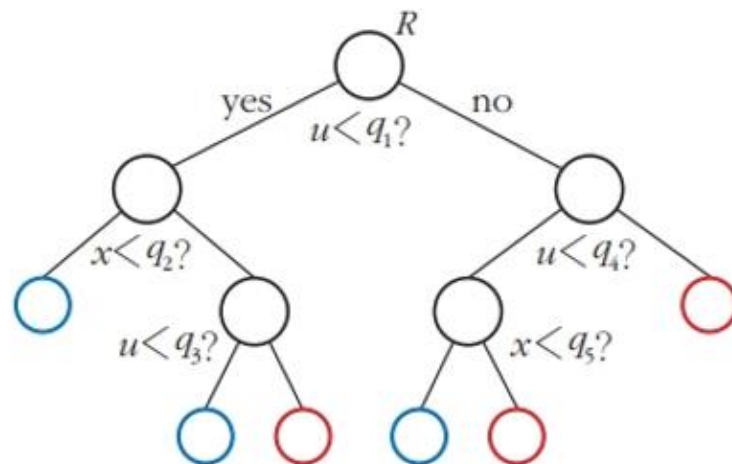
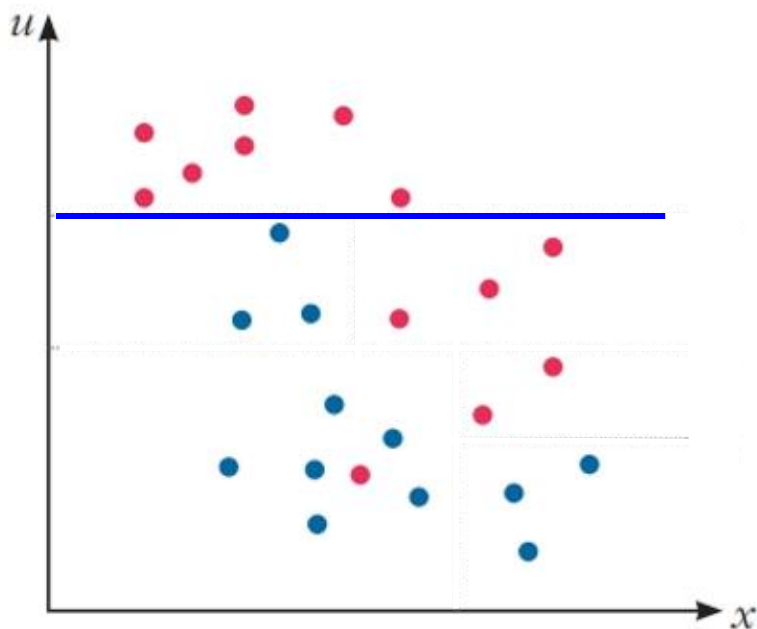
그래프 G



9.2 결정 트리의 원리

■ 결정 트리

- 결정 트리는 이진 트리 ([그림]은 깊이가 3인 결정 트리)
- 결정 트리는 특징 공간을 수평 선분과 수직 선분으로 분할하여 분류를 수행함
- [그림]의 경우에는 x 와 u 라는 두 개의 설명 변수가 2차원 특징 공간 구성
- 반응 변수는 파란색과 빨간색으로 표시된 2개의 부류를 가짐



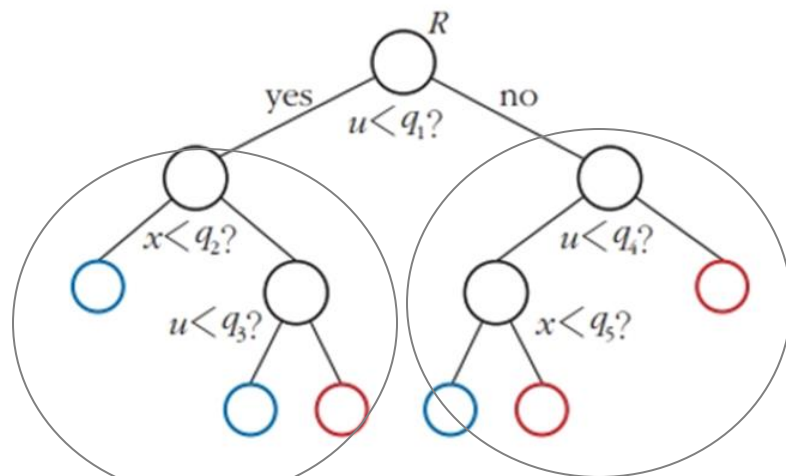
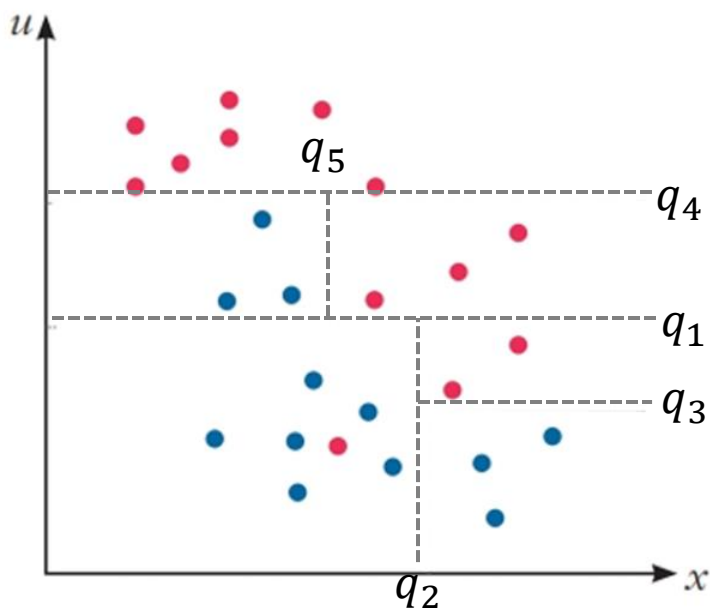
9.2 결정 트리의 원리

■ 결정 트리의 특징 공간 분할

- 루트 노드의 $[u < q_1?]$ 이란 질문은 세로축(u 축)을 q_1 로 나눔
- 전체 샘플을 위쪽 13개(no에 해당하고 오른쪽으로 이동), 아래쪽 12개(yes에 해당하고 왼쪽으로 이동)로 나눔 → 어느 정도 분류했는데 부족하니 같은 일을 반복해야 함

■ 결정 트리의 특징 공간 분할

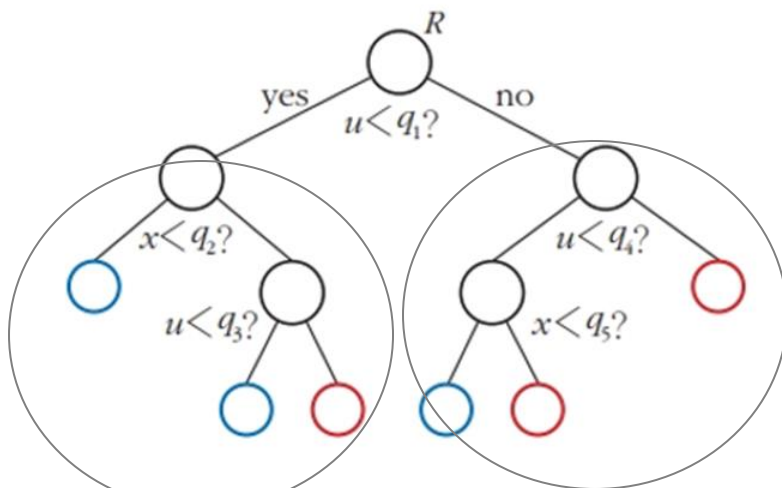
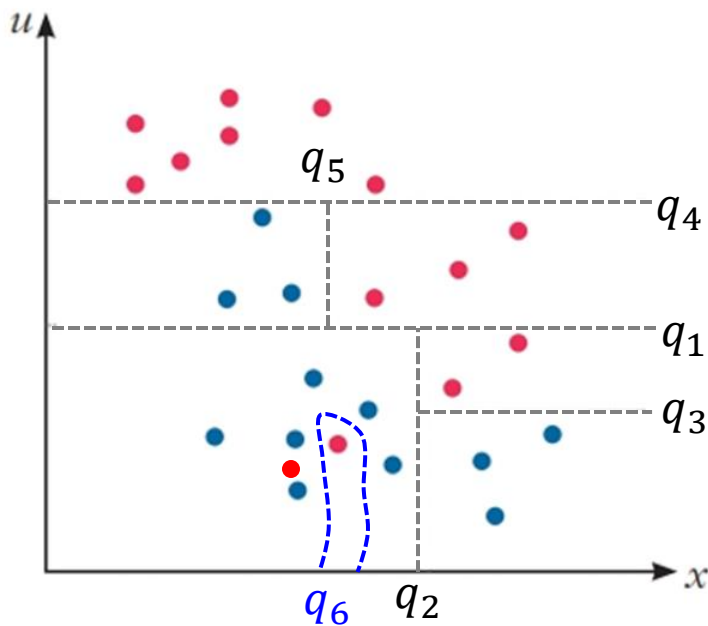
- 루트의 오른쪽 자식 노드가 가진 $[u < q_4?]$ 라는 질문
- 세로축을 q_4 로 나눔 → 13개의 샘플을 위쪽 7개와 아래쪽 6개
- 위쪽은 빨간색 부류만 있으므로 멈춤 → 리프 노드로 설정하고 빨간색 부류를 배정
- 아래쪽은 $[x < q_5?]$ 라는 질문으로 추가 분할



9.2 결정 트리의 원리

■ 과잉적합에 대처

- $[x < q_2?]$ 노드의 왼쪽 자식은 6개 파랑과 1개 빨강을 가져 순수하지 않은데 멈춤
- 추가로 분할하면 완벽하게 분류 가능한데, 과잉적합(overfitting) 발생 가능성
- 과잉적합은 훈련 집합을 너무 완벽하게 처리하려다 새로운 샘플에 대한 성능을 망치는 현상 (8장 [더 알아보기] 참조)
- 모델링의 궁극적인 목표는 일반화(generalization) 능력, 즉 새로운 샘플에 대한 높은 성능 달성이므로 적당한 조건에서 멈추는 전략 사용



Thank you

