Fast campus

2018 Machine Learning with R

Naïve Bayesian Classifier
Logistic Regression

강 필 성

고려대학교 산업경영공학부

pilsung_kang@korea.ac.kr

목차

분류 문제 예시

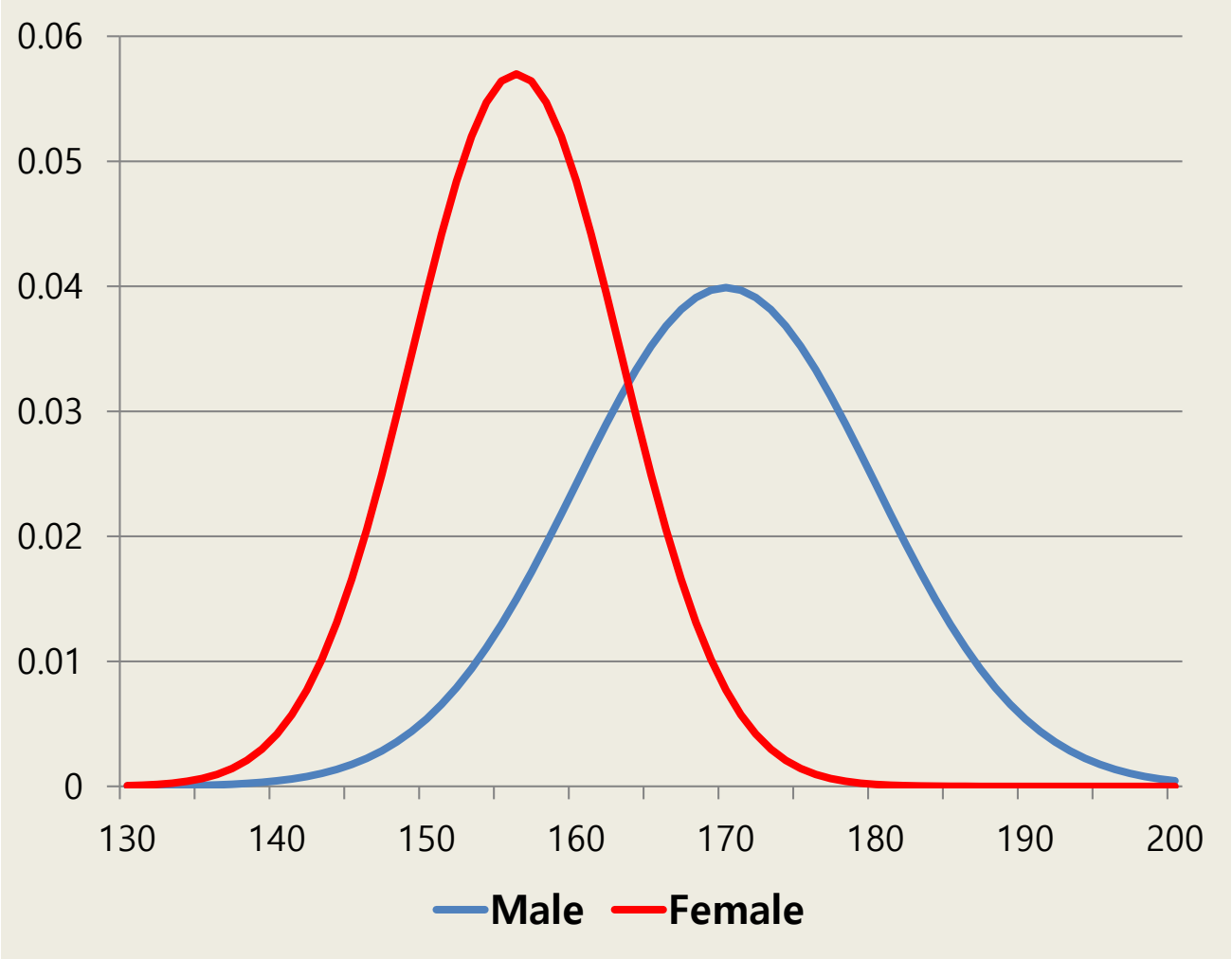
Men

Vs.

Women

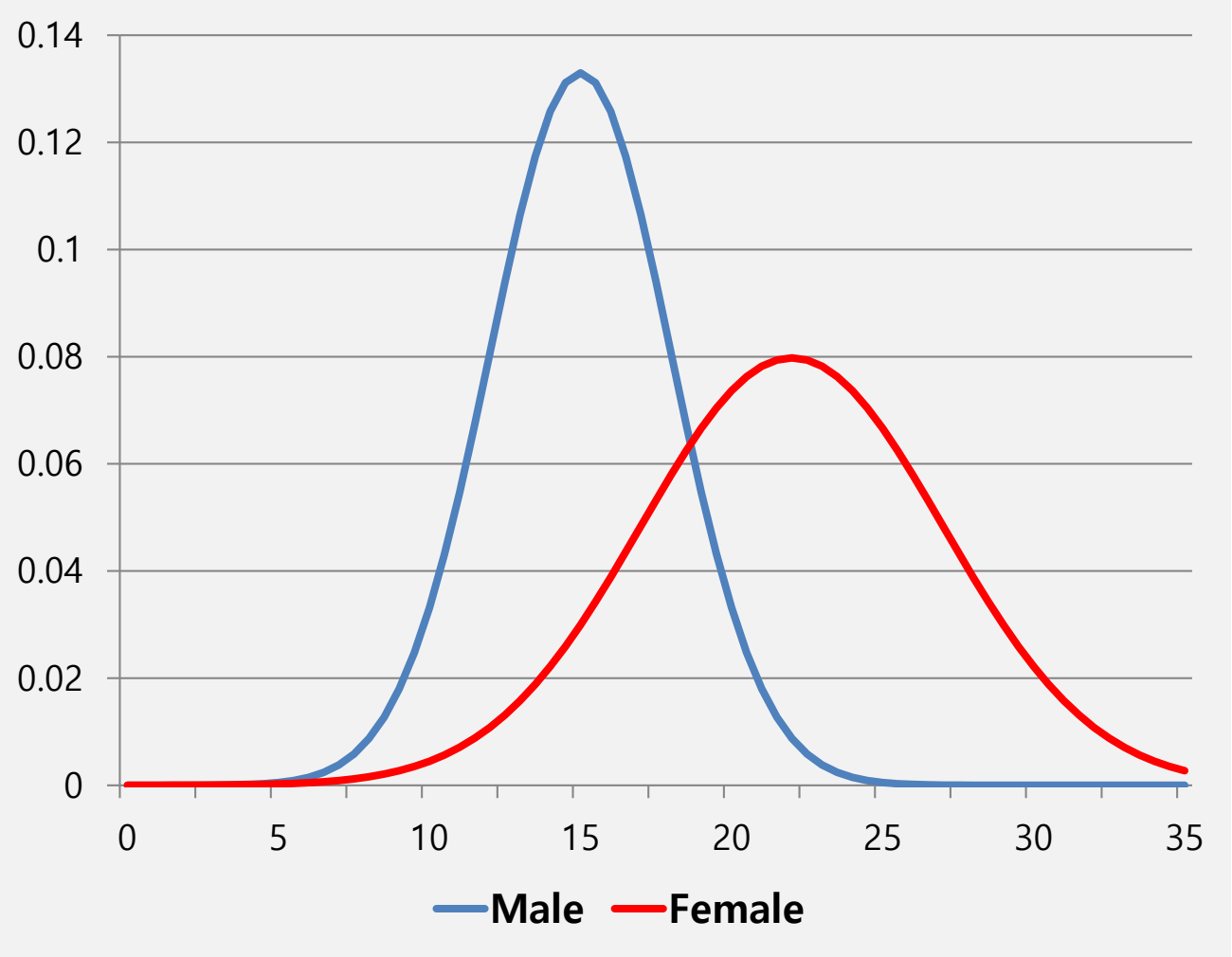
Naïve Bayesian Classification: Concept

❖ 남자와 여자의 키에 대한 사전 분포를 미리 알고 있다면…



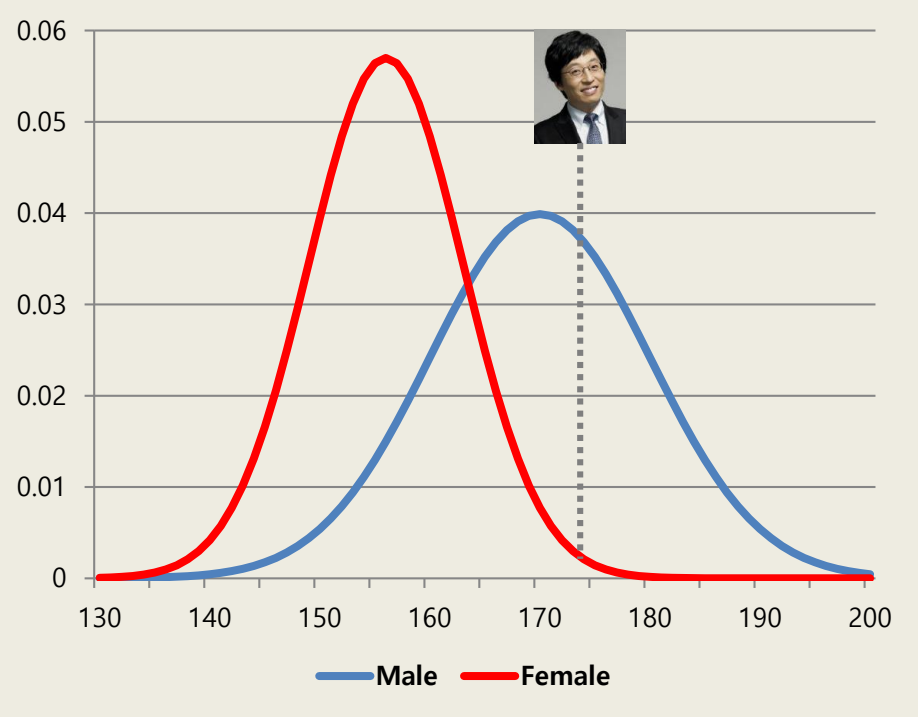
Naïve Bayesian Classification: Concept

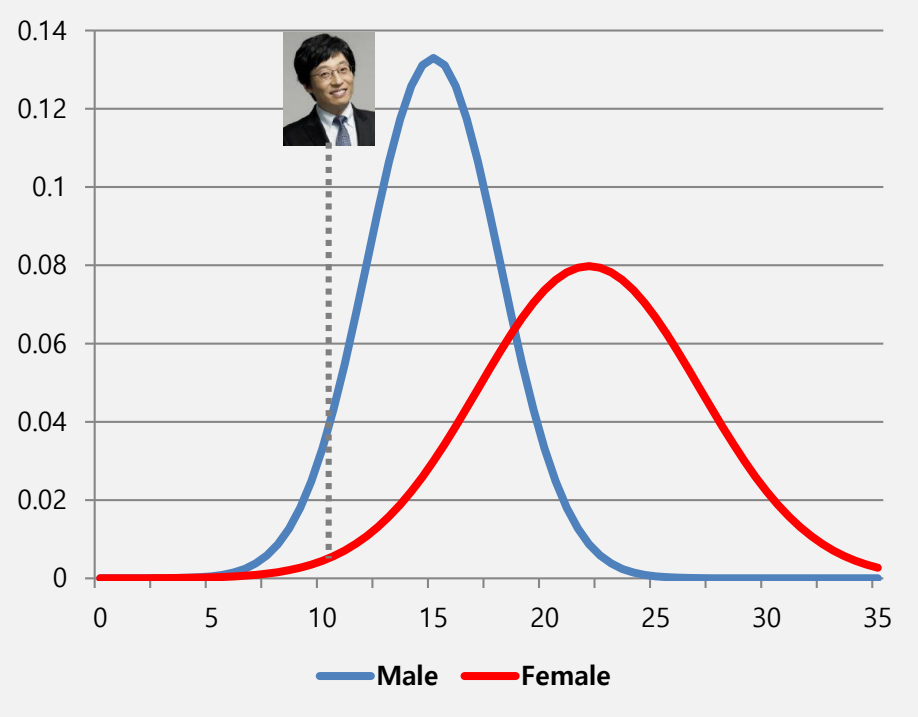
❖ 남자와 여자의 체지방률에 대한 사전 분포를 미리 알고 있다면…



Naïve Bayesian Classification: Concept

❖ 유재석은 남자일까 여자일까?

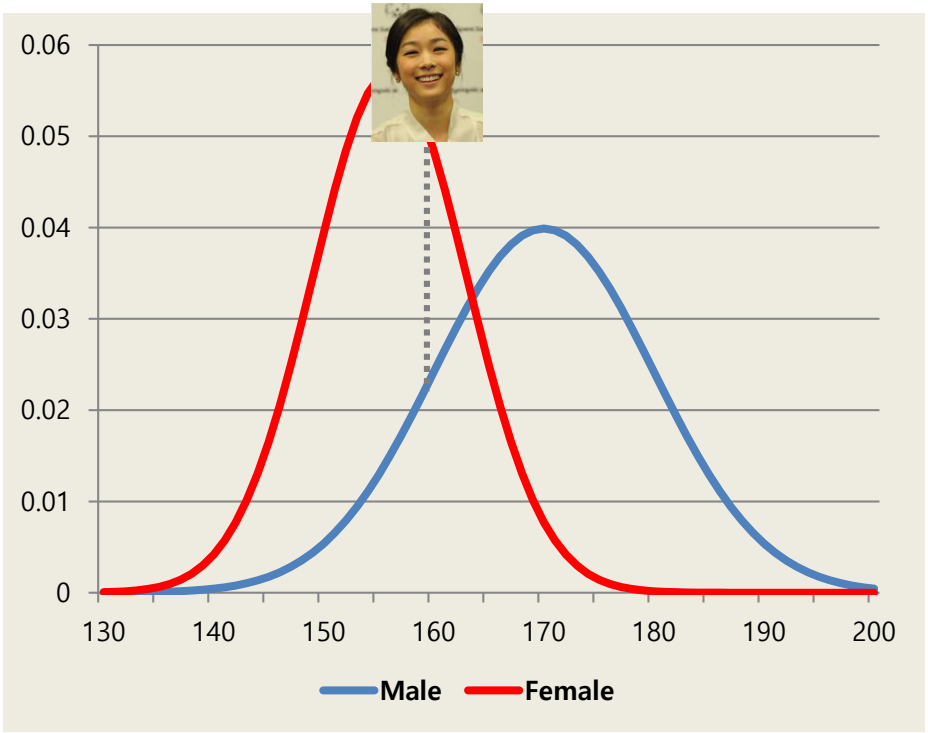


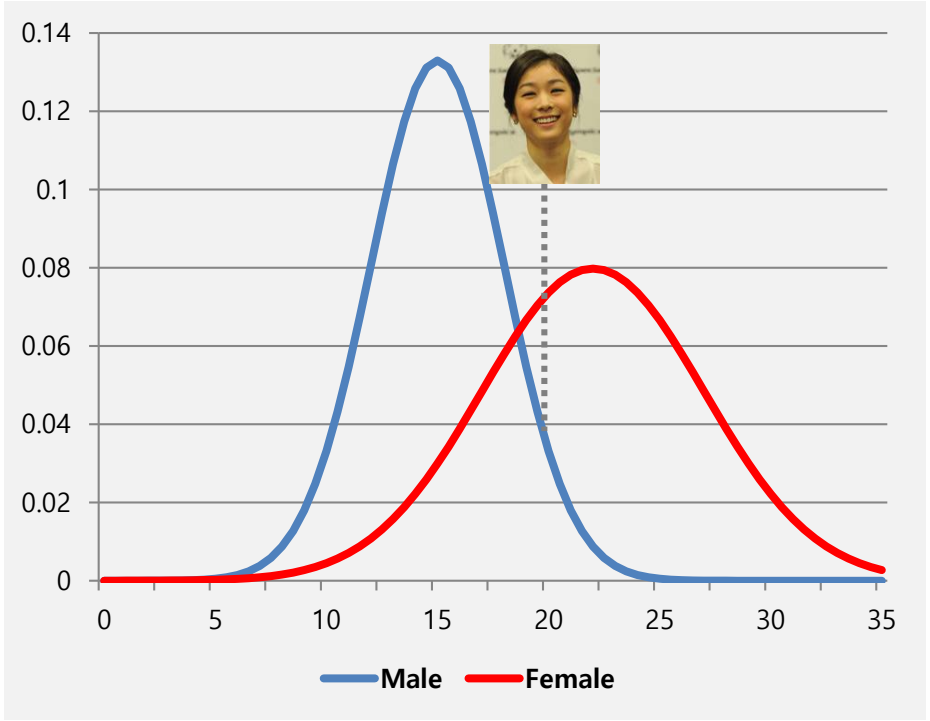


➔ 키로 보나 체지방률로 보나 남자일 가능성이 높음

Naïve Bayesian Classification: Concept

❖ 김연아는 남자일까 여자일까?

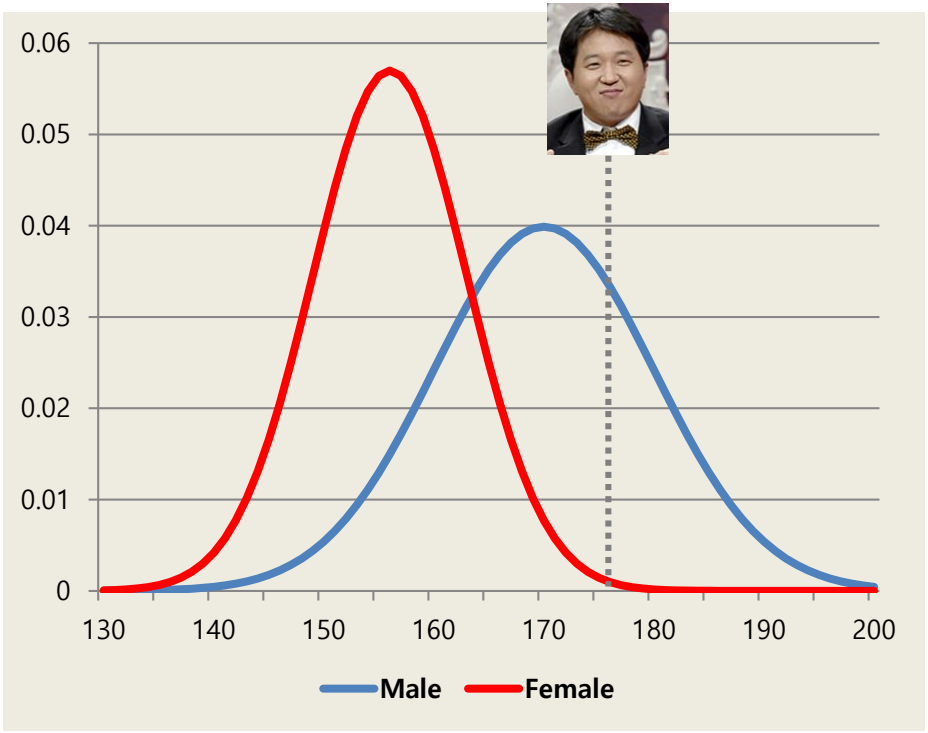


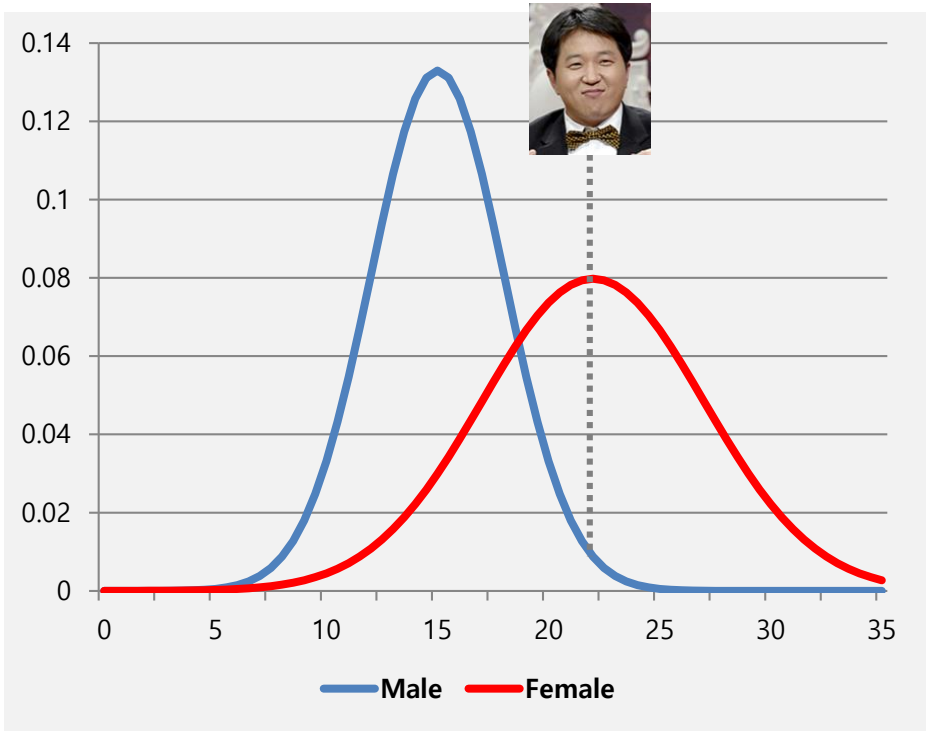


➔ 키로 보나 체지방률로 보나 여자일 확률이 높음

Naïve Bayesian Classification: Concept

❖ 그렇다면 정형돈은?





➔ 키로 보면 남자인데 체지방률로 보면 여자… 어느 범주로 분류를 해야 하지???

Naïve Bayesian Classification: Theory

❖ 베이즈 규칙(Baye's Rule)

Posterior probability (사후확률)

Likelihood (우도)

Prior probability (사전확률)

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B|A) \times P(A)}{P(B)}$$

Evidence (증거)

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{\frac{P(B,A)}{P(A)} \times P(A)}{P(B)}$$

Naïve Bayesian Classification: Theory

❖ 나이브 베이즈 분류기(Naïve Bayesian Classifier)

- 베이즈 규칙 적용
- 각 변수들은 서로 통계적으로 독립이라고 가정

$$P(C_i \mid x_1, x_2, ..., x_d) = \frac{P(x_1, x_2, ..., x_d \mid C_i)P(C_i)}{P(x)}$$

Baye's Rule

$$= \frac{\left(P(x_1 \mid C_i) \cdot P(x_2 \mid C_i) \cdot ... \cdot P(x_n \mid C_i)\right)P(C_i)}{P(x)}$$

Naïve: Variables are statistically independent!

Naïve Bayesian Classification: Decision Rule

- 각 범주에 대한 사후 확률 계산

$$P(C_1 \mid x_1, x_2, ..., x_d) = \frac{\left(P(x_1 \mid C_1) \cdot P(x_2 \mid C_1) \cdot ... \cdot P(x_n \mid C_1)\right) P(C_1)}{P(x)}$$

$$P(C_2 \mid x_1, x_2, ..., x_d) = \frac{\left(P(x_1 \mid C_2) \cdot P(x_2 \mid C_2) \cdot ... \cdot P(x_n \mid C_2)\right) P(C_2)}{P(x)}$$

- 사후 확률값이 높은 범주로 분류

Naïve Bayesian Classification: Concept

- 우리가 여기서 구하고자 하는 것은 다음의 두 확률
  - P(정형돈 Height, 정형돈 BFP | Male)*P(Male) vs.
  - P(정형돈 Height, 정형돈 BFP | Female)*P(Female)
- 만일 두 속성인 height 와 BFP가 통계적으로 독립이라는 가정을 할 수 있고 남자와 여자의 비율이 같다면
  - P(정형돈 Height, 정형돈 BFP | Male) *P(Male) =

    P(정형돈 Height | Male)* P(정형돈 BFP | Male) *P(Male) = 0.035*0.01*= 0.000175
  - P(정형돈 Height, 정형돈 BFP | Female) P(Female) =

    P(정형돈 Height | Female)* P(정형돈 BFP | Female)* P(Female) = 0.001*0.08*0.5 = 0.00004
- 0.000175 > 0.00004 이므로 정형돈은 남자로 분류!

Naïve Bayesian Classification: Procedure

1

학습 데이터 준비

- 설명변수를 정의하고 필요한 학습 데이터 수집
  - ✓ 학습 데이터 총 개체 수: 200 (남성 100명, 여성 100명)
  - ✓ 설명변수: 키(Height), 체지방률(BFS)

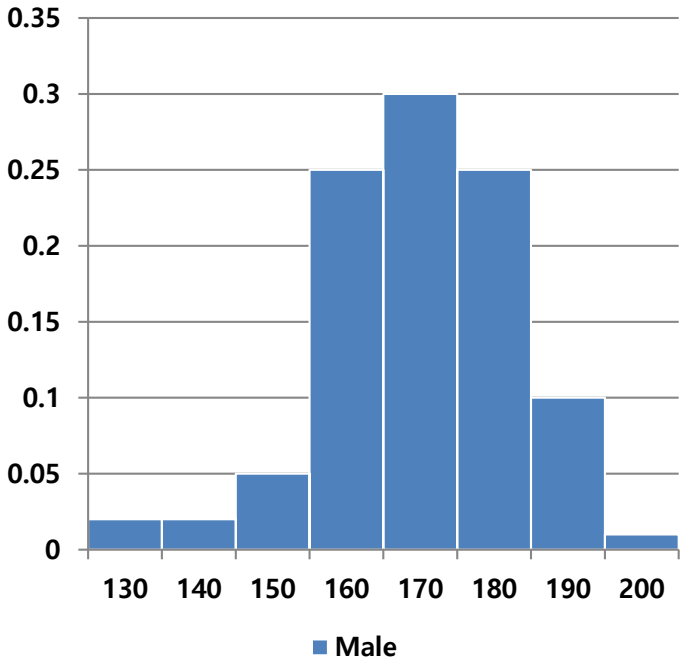
Record	Height	BFS	Class
1	187	15	M
2	165	25	F
3	174	14	M
4	156	29	F
…	…	…	…
N	168	12	M

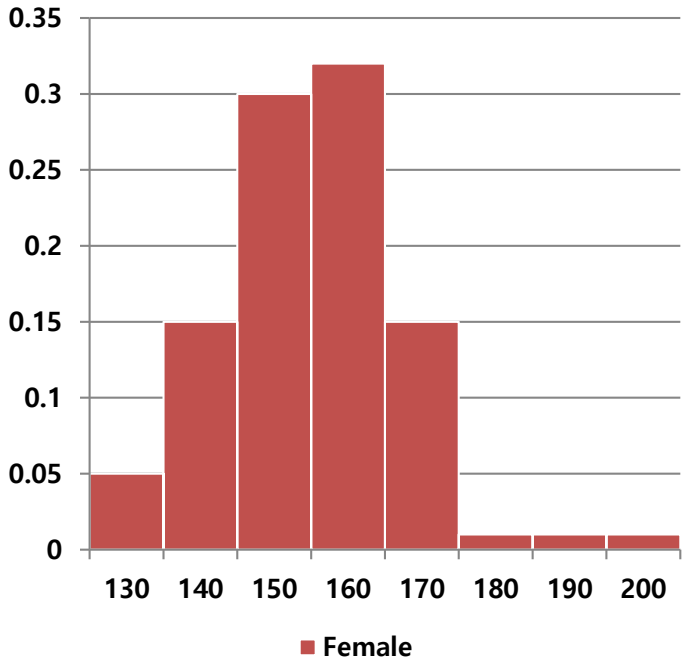
Naïve Bayesian Classification: Procedure

2

범주-변수별 확률분포 추정

- 각 범주의 모든 변수에 대해 확률분포 추정: 히스토그램 사용
- 키에 대한 히스토그램



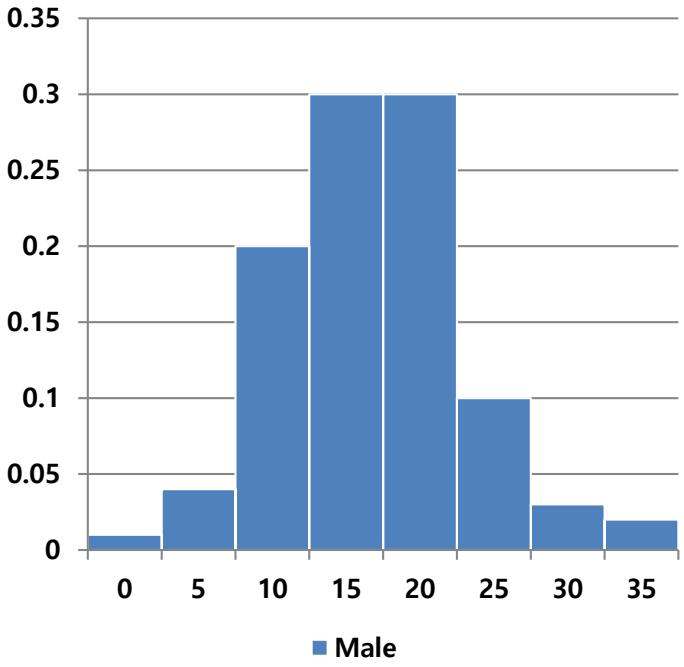


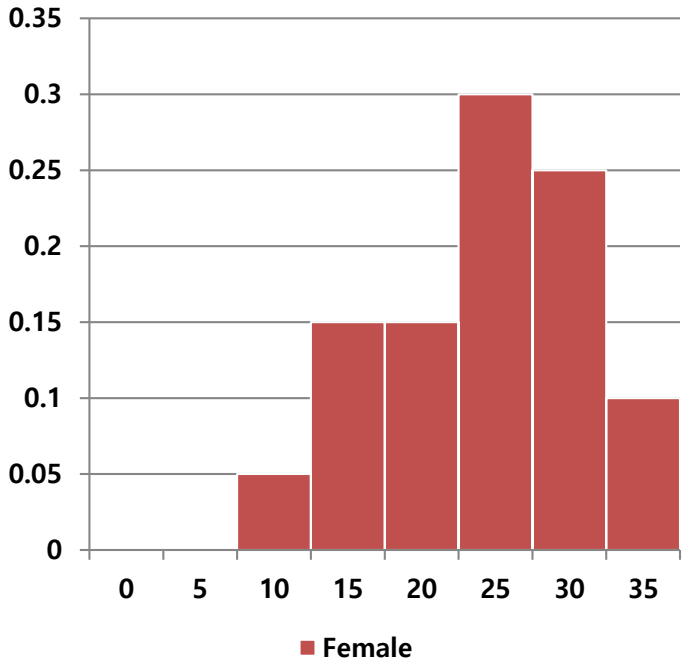
Naïve Bayesian Classification: Procedure

2

범주-변수별 확률분포 추정

- 각 범주의 모든 변수에 대해 확률분포 추정: 히스토그램 사용
- 체지방률에 대한 히스토그램



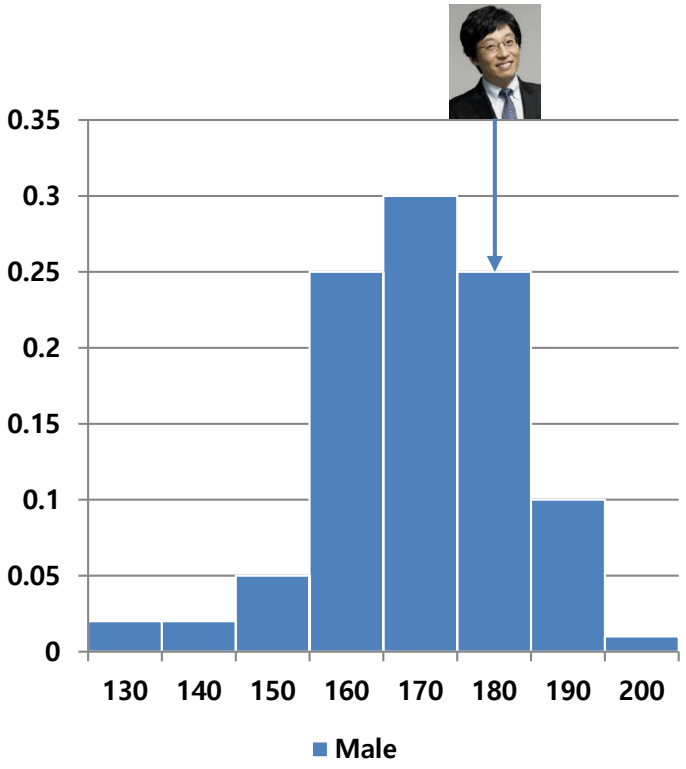


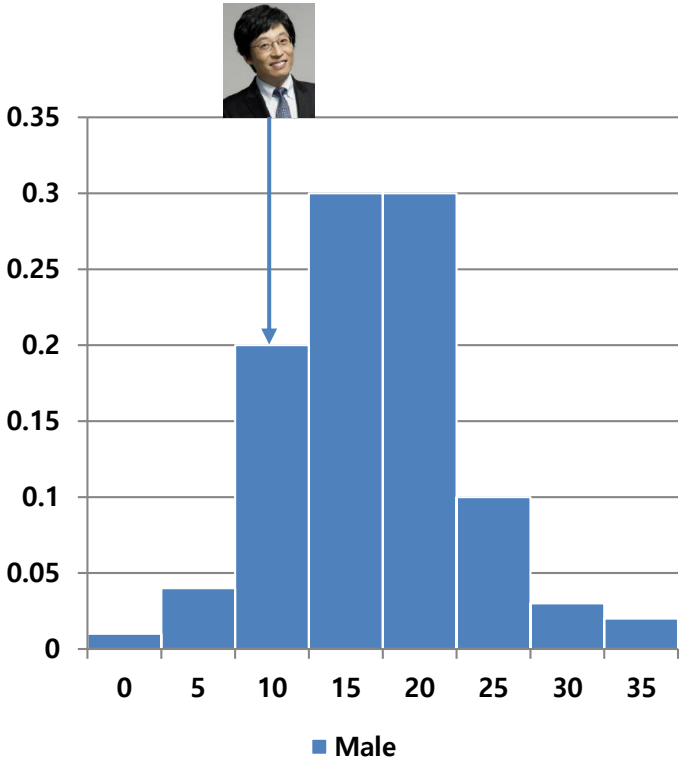
Naïve Bayesian Classification: Procedure

3

각 변수에 대한 조건부 확률 추정

- P(Height = 178 | Male) = 0.25, P(BFS = 11 | Male) = 0.2



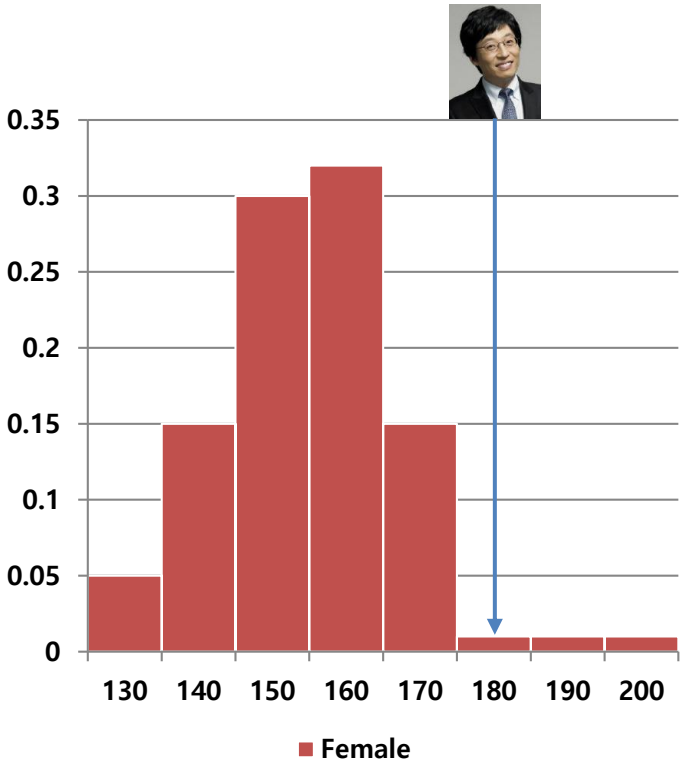


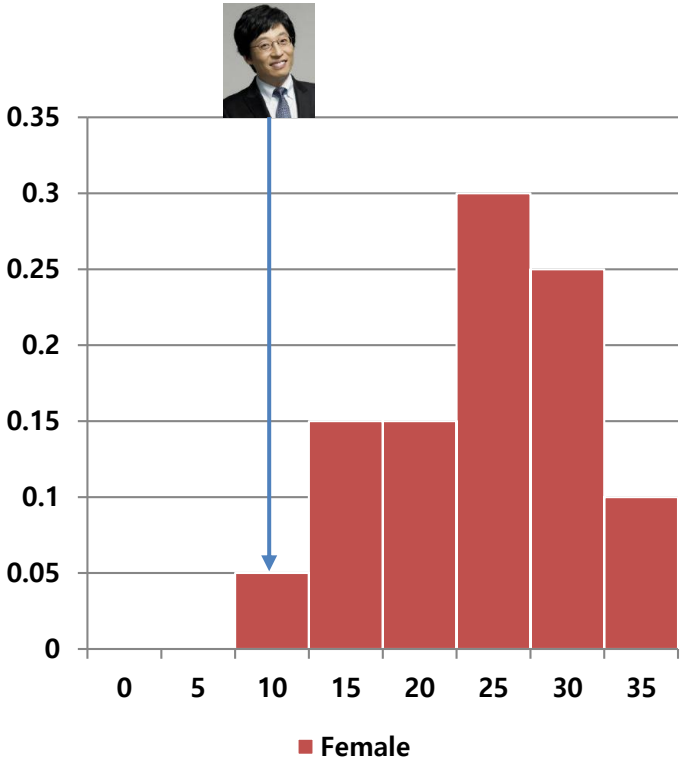
Naïve Bayesian Classification: Procedure

3

각 변수에 대한 조건부 확률 추정

- P(Height = 178 | Female) = 0.01, P(BFS = 11 | Female) = 0.05





Naïve Bayesian Classification: Procedure

각 범주에 속할 사후 확률(Posterior Probability) 추정

- 각 범주에 대한 사후 확률
  - ✓ P(Height = 178, BFS = 11 | Male) * P(Male)

    = P(Height = 178 | Male)* P(BFS = 11 | Male) * P(Male)

    = 0.25*0.2*0.5 = 0.025
  - ✓ P(Height = 178, BFS = 11 | Female) * P(Female)

    = P(Height = 178 | Female)* P(BFS = 11 | Female) * P(Female)

    = 0.01*0.05*0.5 = 0.00025

4

Naïve Bayesian Classification: Procedure

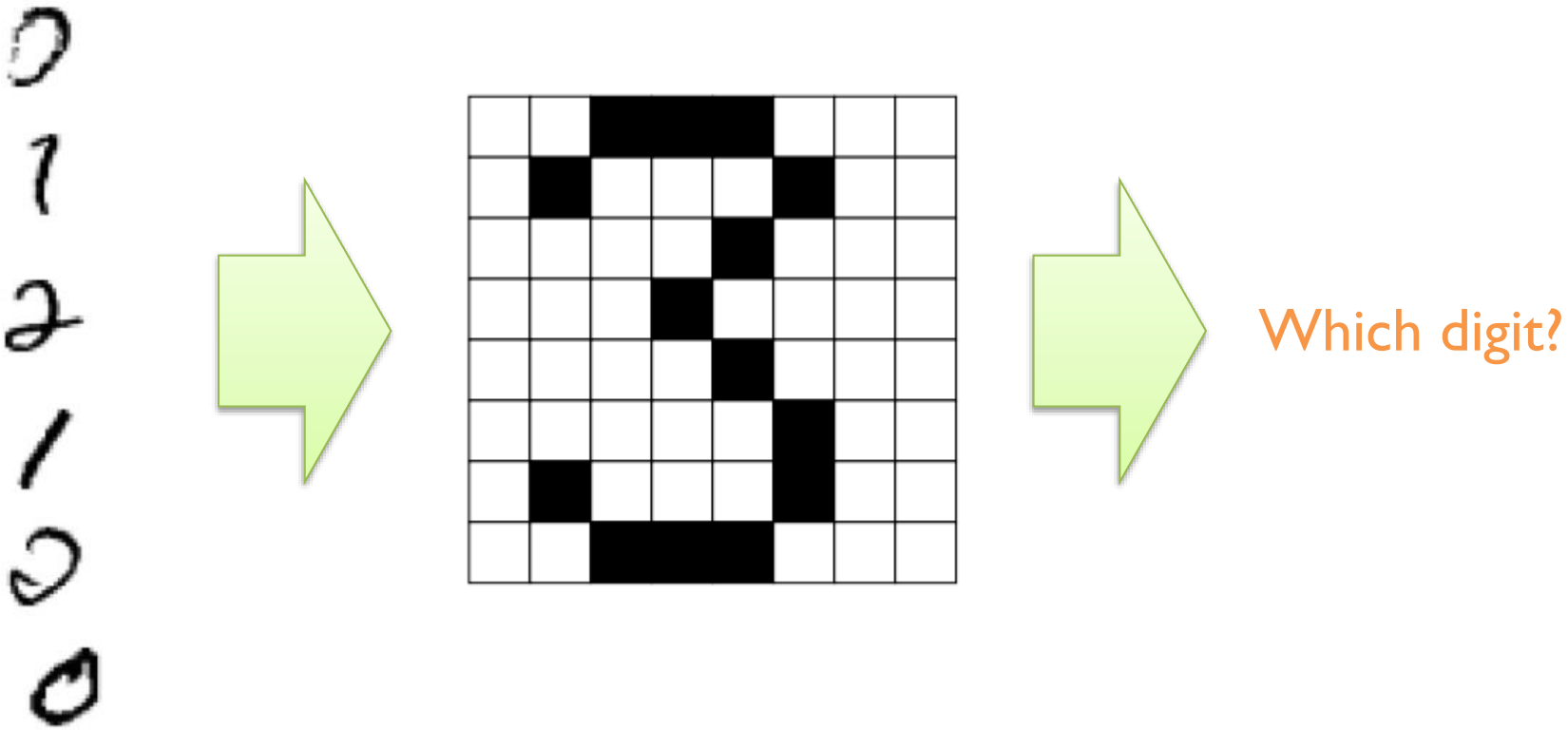
최종 범주 예측

- P(Height=178, BFS=11 | Male) * P(Male) > P(Height=178, BFS=11 | Female) P(Female) ➔ 남성으로 분류
- 만일 학습 데이터가 400명의 남성과 100명의 여성으로 구성되어 있다면?
  - ✓ 각 범주의 사전확률 고려: P(Male) & P(Female)
  - ✓ P(Height=178, BFS=11 | Male)*P(Male) = 0.05*0.8 = 0.04
  - ✓ P(Height=178, BFS=11 | Female)*P(Female) = 0.0005*0.2 = 0.0001

5

Naïve Bayesian Classifier Example

- 손으로 쓴 숫자를 판별하는 문제
  - 입력 변수: 픽셀 정보
  - 범주: 0부터 9까지 10개 범주



Naïve Bayesian Classifier Example

❖ 변수(속성 정의)

- 각 격자의 위치인 <i, j>에 대해 값을 부여
- 단순히 1/0으로 사용할 수도 있고, 어두운 정도를 연속형 숫자로 표현할 수도 있음
- 각 이미지는 아래와 같이 벡터 형태로 변환

$$\rightarrow \langle F_{0,0}=0 \;\; F_{0,1}=0 \;\; F_{0,2}=1 \;\; F_{0,3}=1 \;\; F_{0,4}=0 \;\; \ldots F_{15,15}=0 \rangle$$

❖ 나이브 베이지안 분류기

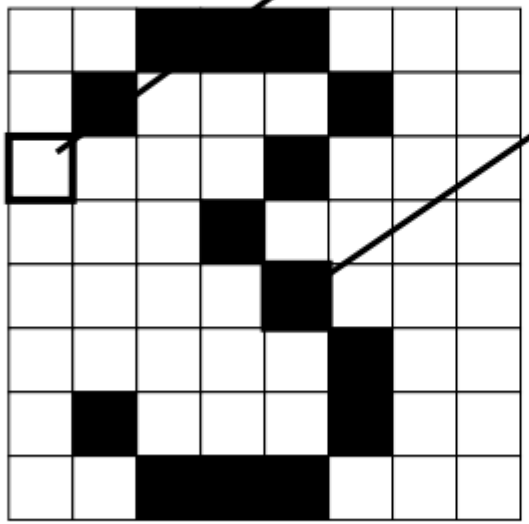
$$P(Y|F_{0,0}\ldots F_{15,15}) \propto P(Y)\prod_{i,j} P(F_{i,j}|Y)$$

Naïve Bayesian Classifier Example

❖ 추정해야 되는 값은 무엇인가?

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

Naïve Bayesian Classifier Example

❖ 학습 절차

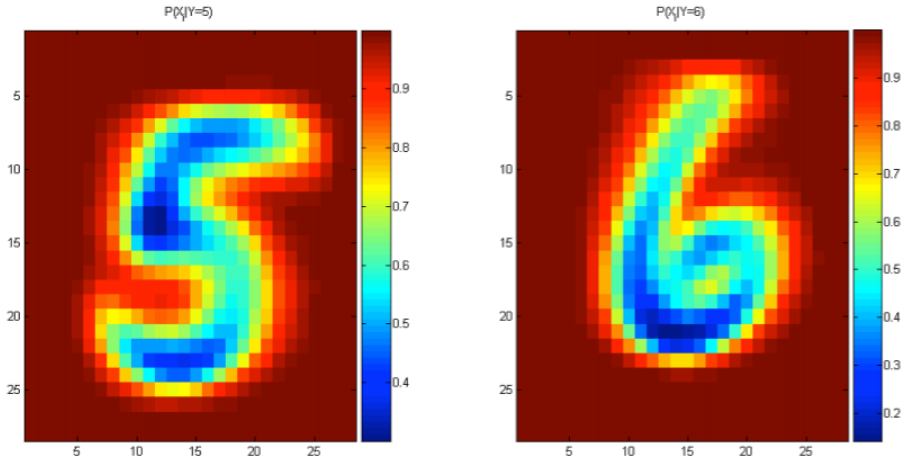
- 각 격자에 대해 범주의 비율을 구함
  - Prior:

$$P(Y = y) = \frac{Count(Y = y)}{\sum_{y'} Count(Y = y')}$$

  - Observation distribution:

$$P(X_i = x | Y = y) = \frac{Count(X_i = x, Y = y)}{\sum_{x'} Count(X_i = x', Y = y)}$$

❖ 학습된 예제

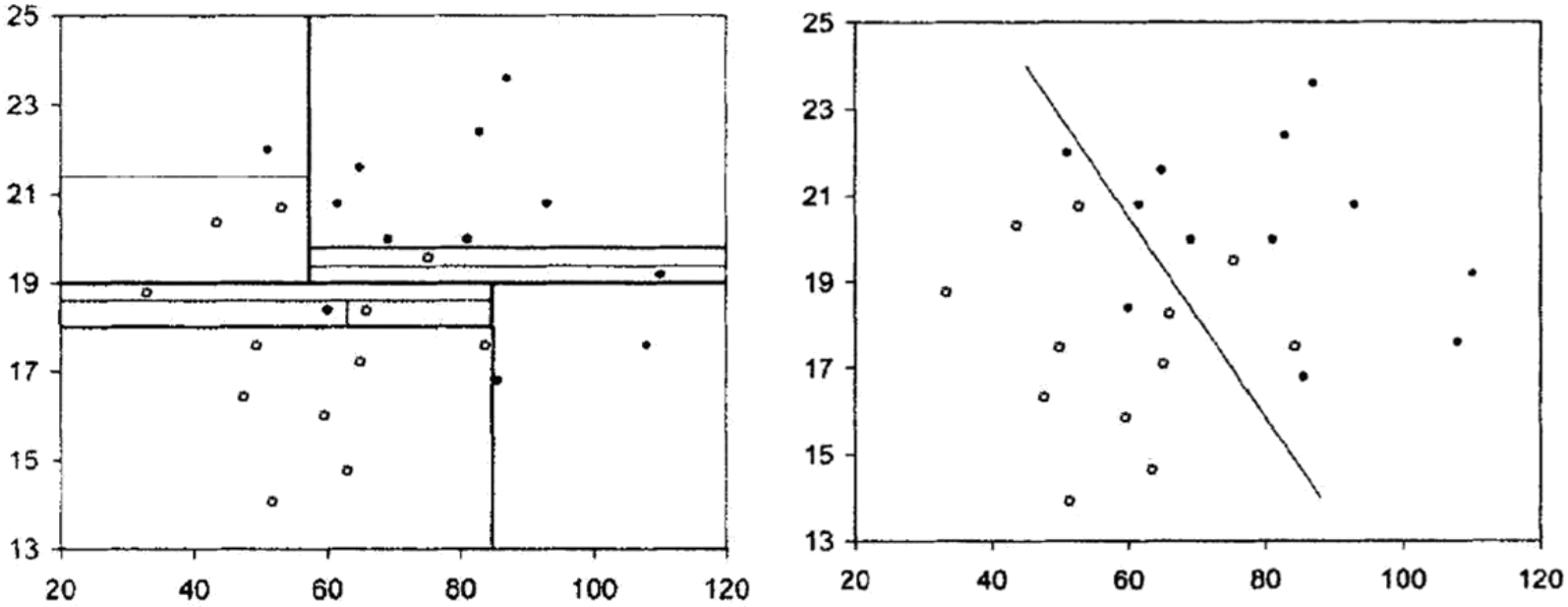


목차

분류 알고리즘 경계면

- 분류를 수행하기 위한 알고리즘은 여러 가지가 존재
  - 동일한 결과를 얻기 위한 다양한 길이 존재하기 때문

"Separate the riding mower buyers(●) from non-buyers(○)"



다중선형회귀분석

❖ 목적

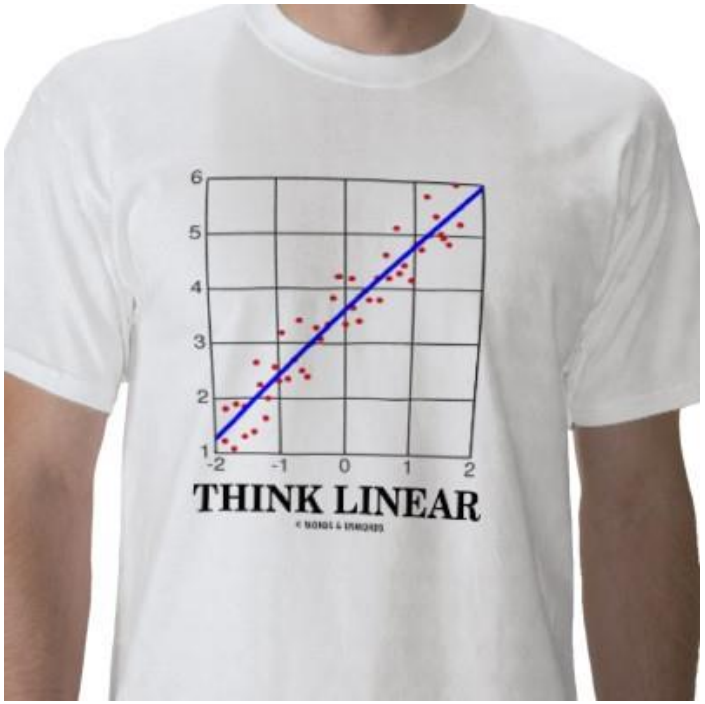
- 수치형 설명변수 X와 종속변수 Y간의 관계를 선형으로 가정하고 이를 가장 잘 표현할 수 있는 회귀 계수를 추정

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots + \beta_d x_d + \epsilon$$

unexplained

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

coefficients



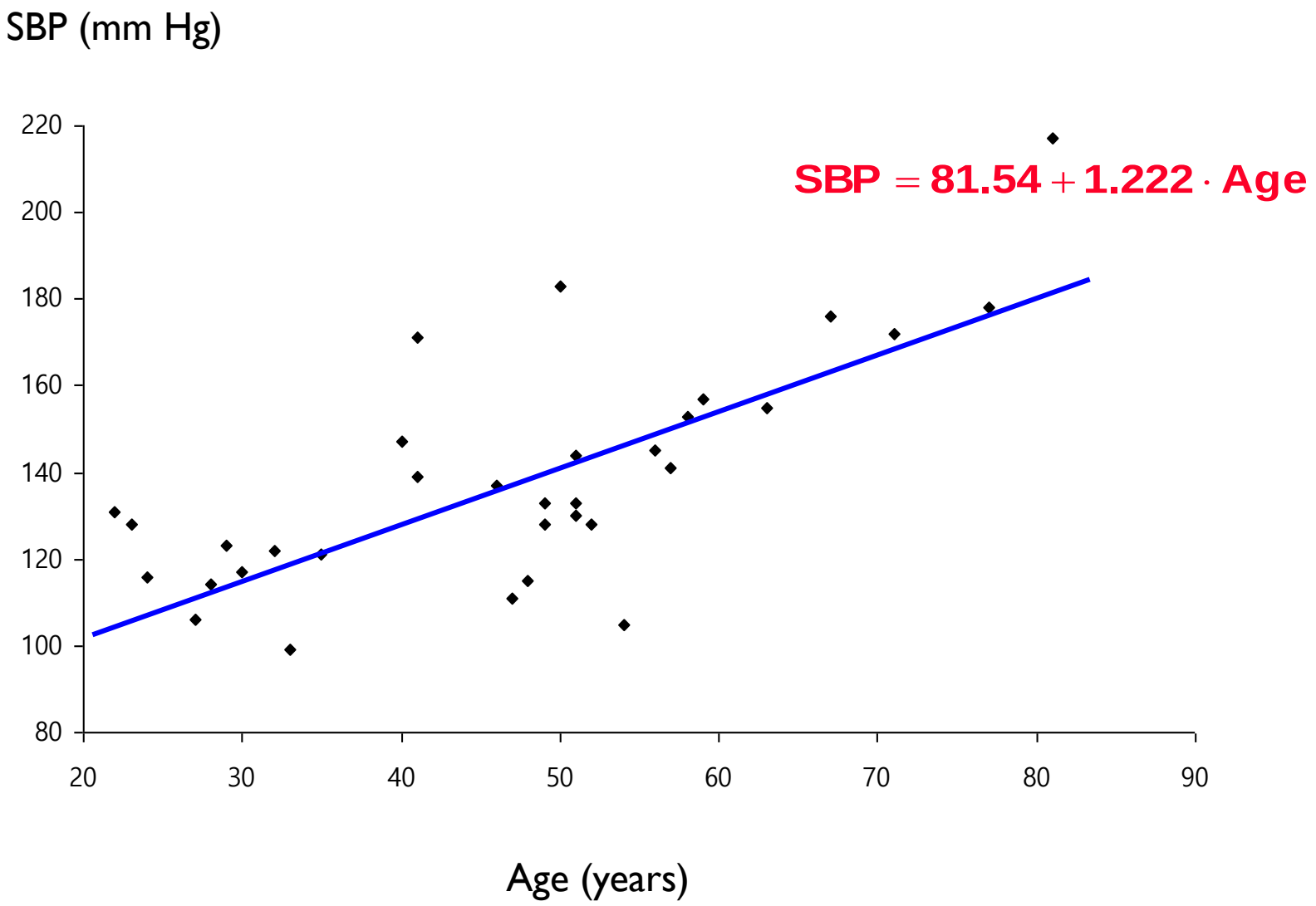
다중선형회귀분석

❖ 예시 I

- 33명의 성인 여성에 대한 나이와 혈압 사이의 관계

Age	SBP	Age	SBP	Age	SBP
22	131	41	139	52	128
23	128	41	171	54	105
24	116	46	137	56	145
27	106	47	111	57	141
28	114	48	115	58	153
29	123	49	133	59	157
30	117	49	128	63	155
32	122	50	183	67	176
33	99	51	130	71	172
35	121	51	133	77	178
40	147	51	144	81	217

다중선형회귀분석

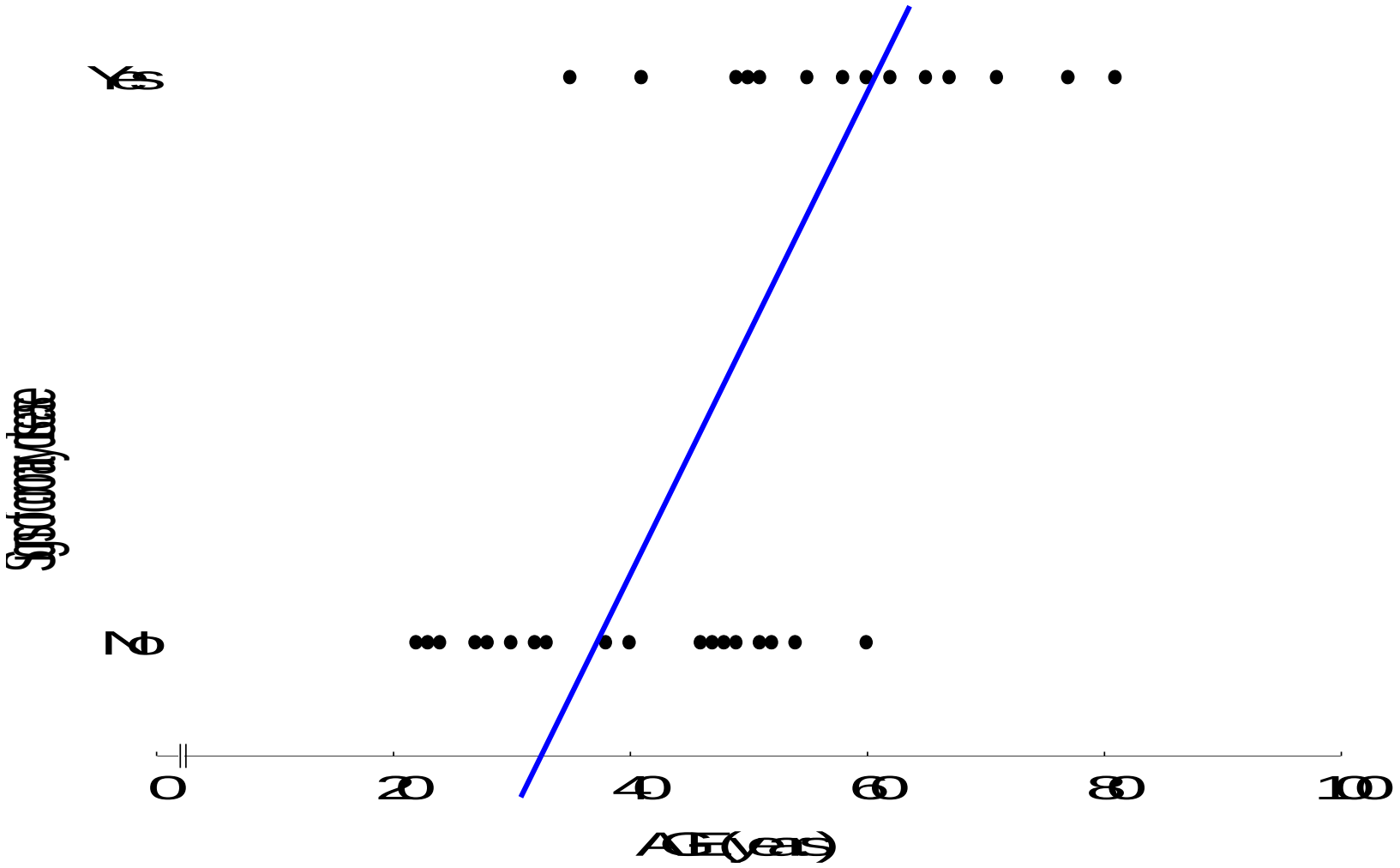

SBP (mm Hg)
220
200
180
160
140
120
100
80
20
30
40
50
60
70
80
90
Age (years)
SBP = 81.54 + 1.222 · Age

만약에...

- 예시 2
  - 연속형 변수가 아닌 이진형(Binary) 변수인 Cancer Diagnosis를 사용한다면?

Age	CD	Age	CD	Age	CD
22	0	40	0	54	0
23	0	41	1	55	1
24	0	46	0	58	1
27	0	47	0	60	1
28	0	48	0	60	0
30	0	49	1	62	1
30	0	49	0	65	1
32	0	50	1	67	1
33	0	51	0	71	1
35	1	51	1	77	1
38	0	52	0	81	1

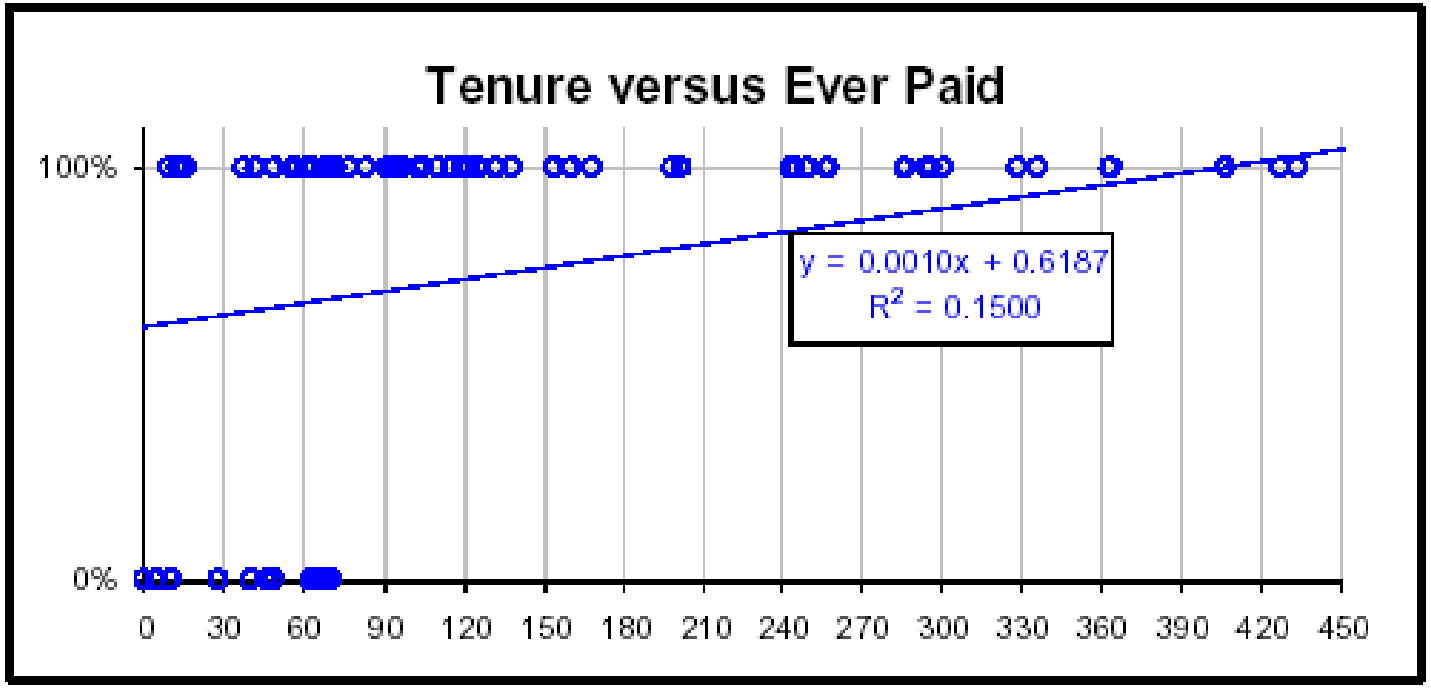
만약에…



분류 문제의 경우

- **확률값을 선형회귀분석의 종속 변수로 사용하는 것이 타당한가?**
  - 선형회귀분석의 우변의 범위에 대한 제한이 없기 때문에 종속변수(좌변) 역시 범위의 제한을 받지 않음

$$P(Y=1) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$



로지스틱 회귀분석

❖ 목적

- 이진형(0/1)의 형태를 갖는 종속변수(분류문제)에 대해 회귀식의 형태로 모형을 추정하는 것

❖ 속성

- 종속변수 Y 자체를 그대로 사용하는 것이 아니라 Y에 대한 로짓 함수(logit function)를 회귀식의 종속변수로 사용
- 로짓함수는 설명변수의 선형결합으로 표현될 수 있음
- 로짓함수의 값은 종속변수에 대한 성공 확률로 역산될 수 있으며, 이는 따라서 분류 문제에 적용할 수 있음

로지스틱 회귀분석

- 2010 World Cup Betting Odds

9 : 2

9 : 2

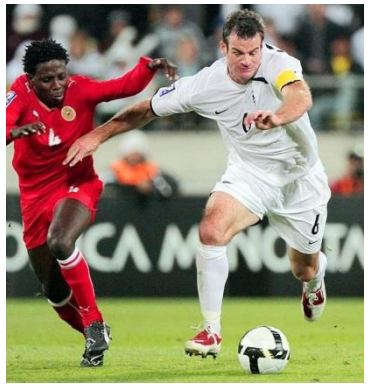
6 : 1

9 : 1

200 : 1

250 : 1

500 : 1



1000 : 1

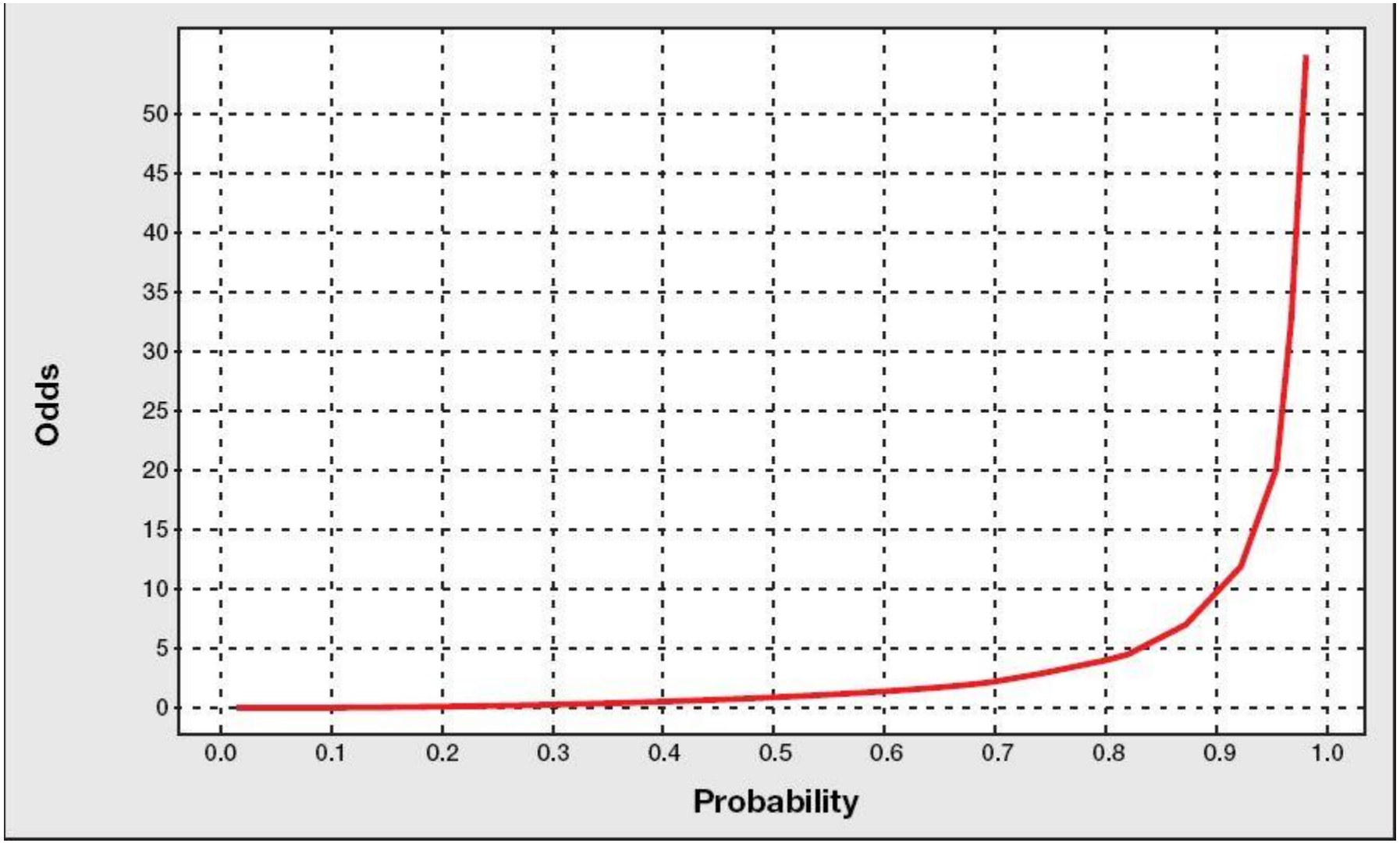
로지스틱 회귀분석: Odds

- Odds
  - p = probability of belonging to class 1 (success).

$$Odds = \frac{p}{1-p}$$

- 이전 예시에 대해
  - 스페인의 우승 odds는 2/9이므로 스페인의 우승 확률은 2/11임
  - 대한민국의 우승 odds는 1/250 이므로 대한민국의 우승확률은 1/251 ≒ 0.00398 (0.398%)임
  - 1,000년을 살면 대한민국이 월드컵에서 한 번 우승하는 모습을 목격할 수 있음

로지스틱 회귀분석: Odds



로지스틱 회귀분석: Log Odds

❖ Odds의 한계

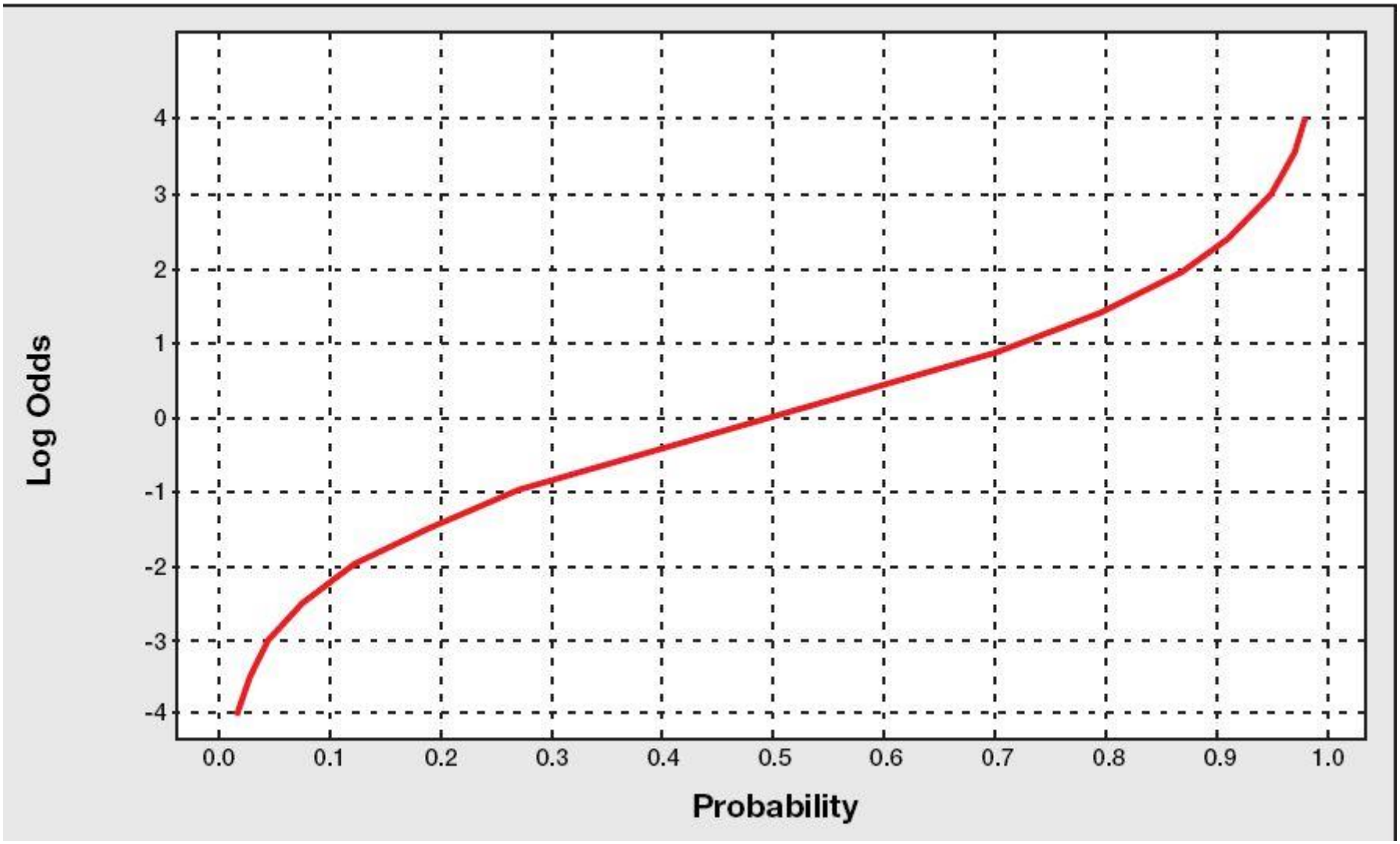
- 여전히 범위에 대한 제약이 존재함: 0 < odds < ∞
- 비대칭성(Asymmetric)

❖ Odds에 로그를 취하자

$$\log(Odds) = \log\left(\frac{p}{1-p}\right)$$

- 드디어 범위에 대한 제약이 없어짐: - ∞ < log(odds) < ∞
- 대칭성 확보
- 성공확률 p가 작으면 음수값을 갖고, 성공확률 p가 크면 양수값을 가짐

로지스틱 회귀분석: Log Odds



로지스틱 회귀분석: Equation

❖ 로지스틱 회귀분석 식

- Log Odds를 이용한 회귀분석 식

$$log(Odds) = log\left(\frac{p}{1-p}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

- 양변에 로그를 취하면

$$\frac{p}{1-p} = e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d}$$

- 성공확률에 대한 식으로 표현

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$

로지스틱 회귀분석: Equation

❖ 로지스틱 회귀분석 식

Logistic Regression 선형식

$$\ln\left(\frac{p}{1-p}\right)$$: logit (odds에 자연로그를 취한 상태)

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i$$

- 로지스틱 회귀 모형은 종속변수가 이분형일 때 선형회귀모형의 제약을 극복하기 위해 확률에 대한 로짓 변환을 고려하여 분석

$$P = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i)}$$

- 위의 모형식에서 추정된 회귀계수로부터 사후확률에 대한 추정식을 계산

로지스틱 회귀분석: 학습 (Optional)

❖ 최대 우도 추정법: Maximum likelihood estimation (MLE)

- Expectation function:

$$P(x_i, y_i \mid \beta) = \begin{cases} \sigma(\mathrm{x}, \beta) & \text{if } y = 1 \\ 1 - \sigma(\mathrm{x}, \beta) & \text{if } y = 0 \end{cases}$$

$$= \sigma(\mathrm{x}, \beta)^{y} \left(1 - \sigma(\mathrm{x}, \beta)\right)^{1-y}$$

- Likelihood and log-likelihood of the training data **X**:

$$L(\mathrm{X}, y, \beta) = \prod_{i=1}^{R} \sigma(\mathrm{x_i}, \beta)^{y_i} \left(1 - \sigma(\mathrm{x_i}, \beta)\right)^{1-y_i}$$

$$\ln L(\mathrm{X}, y, \beta) = \sum_{i=1}^{R} y_i \ln\left(\sigma(\mathrm{x_i}, \beta)\right) + (1 - y_i) \ln\left(1 - \sigma(\mathrm{x_i}, \beta)\right)$$

- 우도함수와 로그-우도함수는 회귀계수 β에 대해 비선형이므로 선형회귀분석과 같이 명시적인 해가 존재하지 않음
  - ✓ Conjugate gradient 등의 최적화 알고리즘을 차용하여 해를 구함

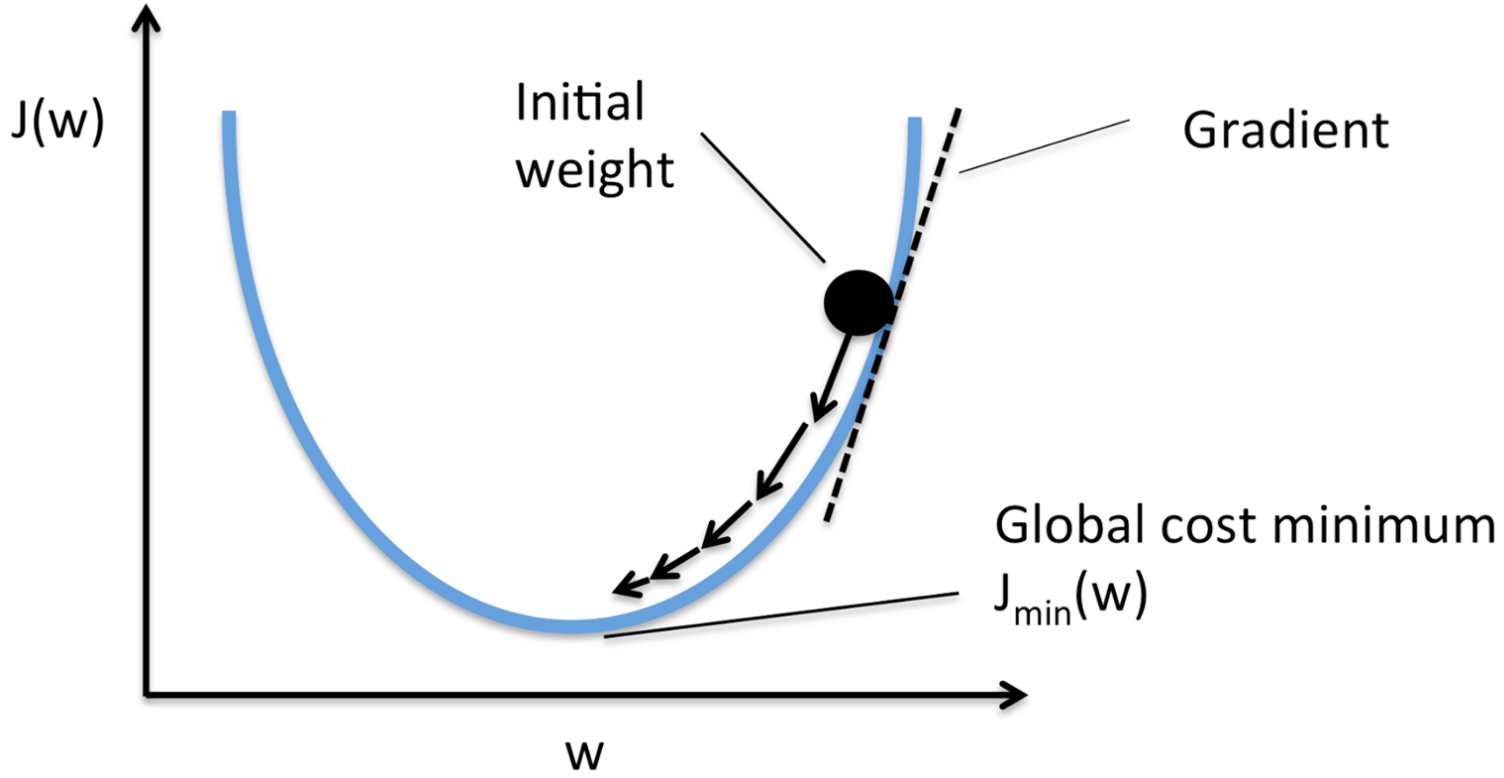
기울기 하강: Gradient Descent

- 눈을 가린 채로 산에서 가장 낮은 곳을 찾아가기

기울기 하강: Gradient Descent

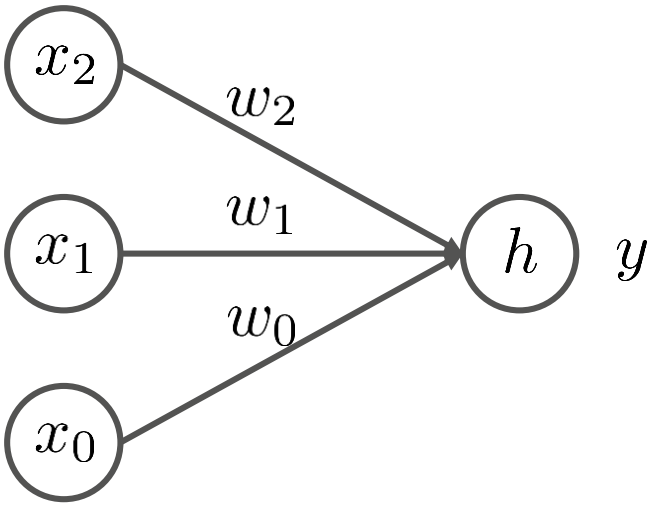
❖ 기울기 하강: Gradient descent algorithm

- 파란색 선: 가중치 w의 변화에 따른 목적함수 값의 변화
- 검은색 점: 현재 해의 위치
- 화살표: 목적함수를 최적화하기 위해 가중치 w가 이동해야 하는 방향



기울기 하강: Gradient Descent

- A Simple Example (Logistic Regression with two input variables)



$$h = \sum_{i=0}^{2} w_i x_i$$

$$y = \frac{1}{1 + exp(-h)}$$

- Let's define the squared loss function $L = \frac{1}{2}(t - y)^2$
- How to find the gradient w.r.t. w or x?

기울기 하강: Gradient Descent

- Use chain rule

$$\frac{\partial L}{\partial y} = y - t$$

$$\frac{\partial y}{\partial h} = \frac{exp(-h)}{(1+exp(-h))^2} = \frac{1}{1+exp(-h)} \cdot \frac{exp(-h)}{1+exp(-h)} = y(1-y)$$

$$\frac{\partial h}{\partial w_i} = x_i$$

- Gradients for w and x

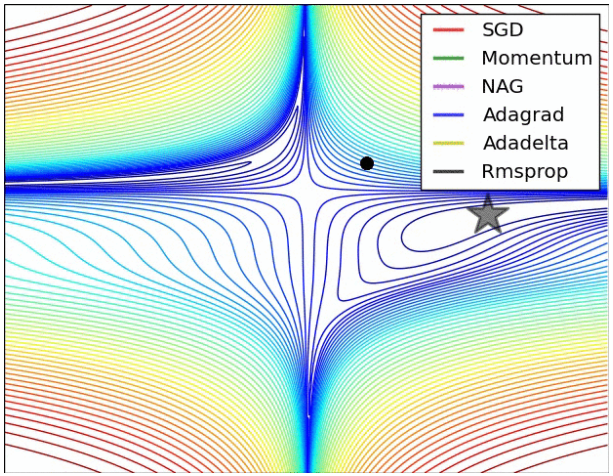
$$\frac{L}{\partial w_i} = \frac{L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial w_i} = (y-t) \cdot y(1-y) \cdot x_i$$

- Update w

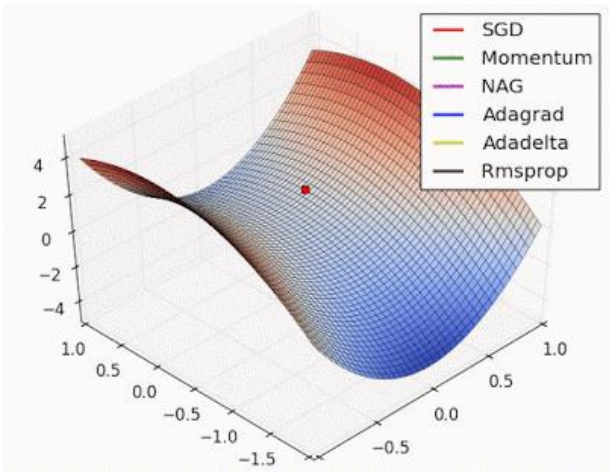
$$w_{new} = w_{old} - \alpha \times \frac{L}{\partial w_i} = w_{old} - \alpha \times (y-t) \cdot y(1-y) \cdot x_i$$

기울기 하강: Gradient Descent

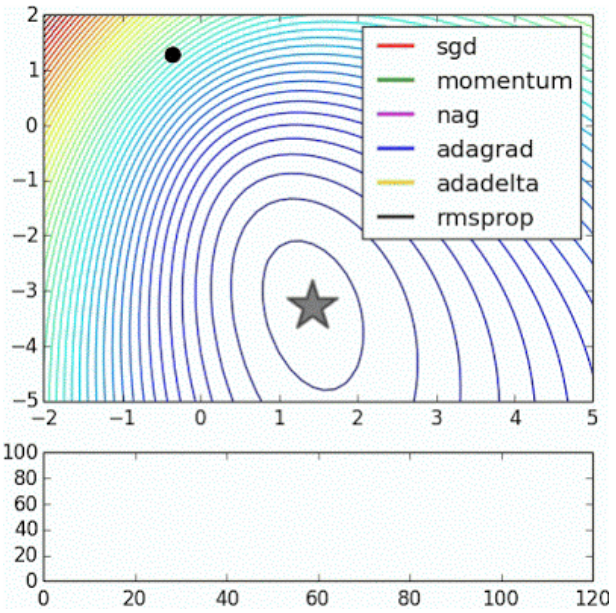
❖ Gradient Descent 의 수렴(Convergence)



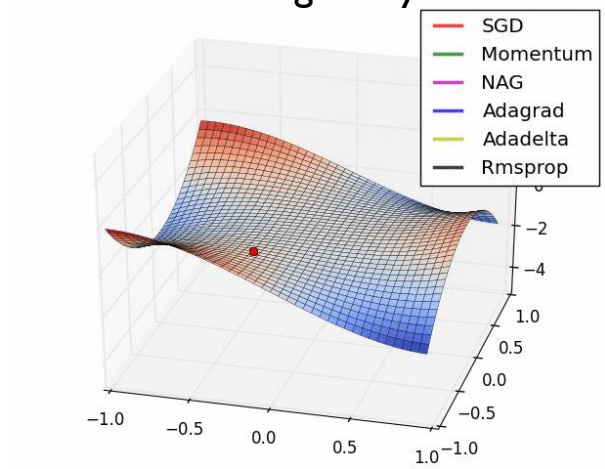
Beale's function



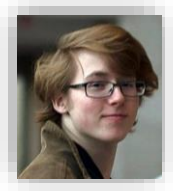
Long valley



Noisy moons



Saddle point



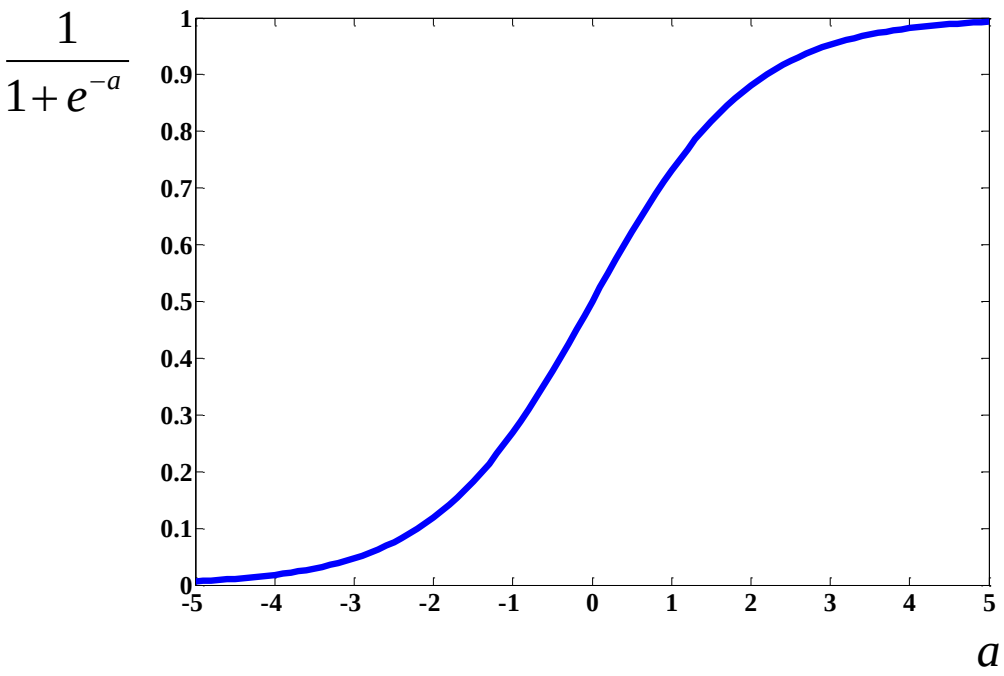
http://www.denizyuret.com/2015/03/alec-radfords-animations-for.html

로지스틱 회귀분석: 학습

❖ 성공 확률

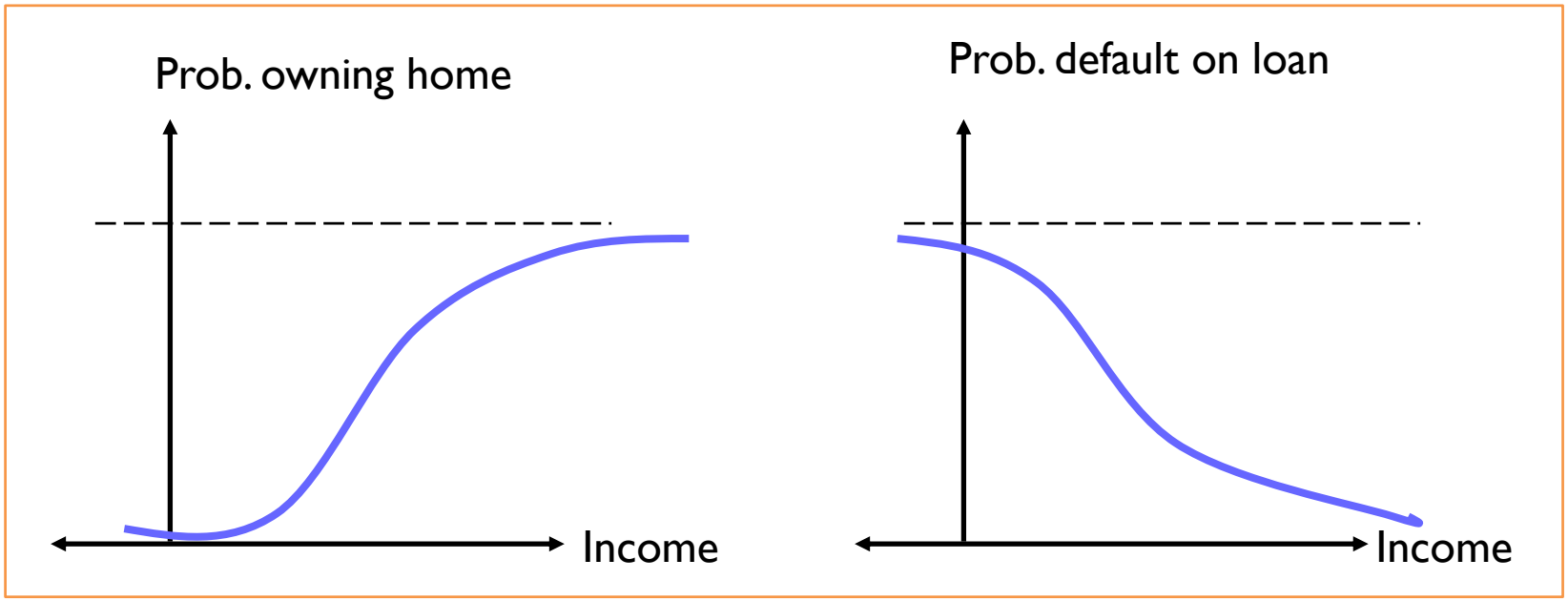
- 회귀계수가 추정되고 나면 주어진 설명변수집합에 대한 성공확률을 다음과 같이 계산할 수 있음

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$



로지스틱 함수의 의미

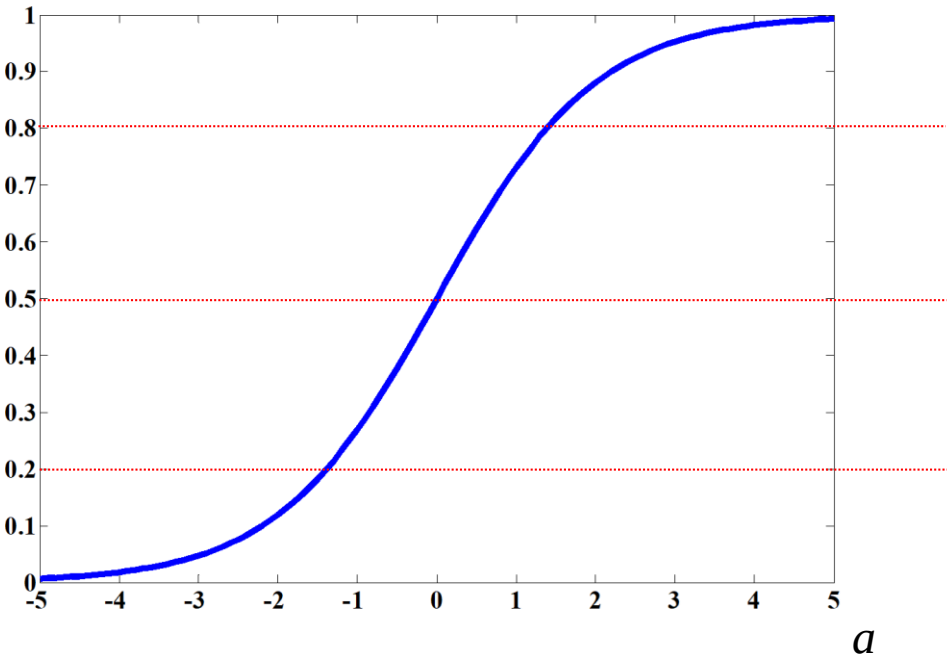
- 실제 상황에서는
  - 특정 변수에 대한 확률 값은 선형이 아닌 S-커브 형태를 따르는 경우가 많음



로지스틱 회귀분석: Cut-off

❖ 이진분류를 위한 cut-off 설정

$$\frac{1}{1+e^{-a}}$$



성공 범주의 비중이 높을 때 (제조업 데이터에서는 거의 발생하지 않는 case)

가장 일반적인 cut-off

성공 범주의 비중이 낮을 때 (예: 불량 예측)

- 일반적으로 0.50가 주로 사용됨
- 사전확률을 고려한 cut-off나 검증데이터의 정확도를 최대화하는 cut-off 등이 사용될 수도 있음

로지스틱 회귀분석: Interpretation

❖ 로지스틱 회귀분석 회귀계수의 의미

- 선형 회귀분석 회귀식

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

✓ 선형 회귀분석에서의 회귀계수는 해당 변수가 1 증가함에 따른 종속변수의 변화량

- 로지스틱 회귀분석 회귀식

$$log(Odds) = log\left(\frac{p}{1-p}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$

✓ 로지스틱 회귀분석에서의 회귀계수는 해당 변수가 1 증가함에 따른 로그 승산의 변화량

로지스틱 회귀분석: Interpretation

❖ 승산 비율: Odds Ratio

- 로지스틱 회귀분석에서 나머지 변수는 모두 고정시킨 상태에서 한 변수를 1만큼 증가시켰을 때 변화하는 Odds의 비율
- Odds ratio:

$$\frac{odds(x_1+1,\cdots,x_d)}{odds(x_1,\cdots,x_d)} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1(x_1+1)+\hat{\beta}_2 x_2+\cdots+\hat{\beta}_d x_d}}{e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\hat{\beta}_2 x_2+\cdots+\hat{\beta}_d x_d}} = e^{\hat{\beta}_1}$$

- x_1이 1 증가하게 되면 성공에 대한 승산 비율이 e^{β_1} 만큼 변화함
  - ✓ 회귀 계수가 양수 → 변수가 증가하면 성공 확률이 증가 (성공범주와 양의 상관관계)
  - ✓ 회귀 계수가 음수 → 변수가 증가하면 성공 확률이 감소 (성공범주와 음의 상관관계)

$$\frac{p}{1-p} = e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\hat{\beta}_2 x_2\cdots+\hat{\beta}_d x_d}$$

로지스틱 회귀분석: Interpretation

❖ 로지스틱 회귀분석 결과 및 해석

- 로지스틱 회귀분석을 수행하고 나면 선형 회귀분석과 유사하게 다음과 같은 표를 결과로 얻을 수 있음

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

로지스틱 회귀분석: Interpretation

❖ 로지스틱 회귀분석 결과 및 해석

- 회귀계수: Coefficient
  - ✓로지스틱 회귀분석에서 각 변수에 대응하는 베타값임
  - ✓선형회귀분석에서는 해당 변수가 1단위 증가할 때 종속변수의 변화량을 의미하나, 로지스틱 회귀분석에서는 해당 변수가 1단위 증가할 때 로그승산비의 변화량을 의미
  - ✓양수이면 성공확률과 양의 상관관계, 음수이면 성공 확률과 음의 상관관계

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

로지스틱 회귀분석: Interpretation

❖ 로지스틱 회귀분석 결과 및 해석

- 유의확률: p-value
  - ✓로지스틱 회귀분석에서 해당 변수가 통계적으로 유의미한지 여부를 알려주는 지표
  - ✓0에 가까울수록 모델링에 중요한 변수이며, 1에 가까울수록 유의미하지 않은 변수임
  - ✓특정 유의수준(α)을 설정하여 해당 값 미만의 변수만을 사용하여 다시 로지스틱 회귀분석을 구축하는 것도 가능함 (주로 $\alpha = 0.05$ 사용)

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

로지스틱 회귀분석: Interpretation

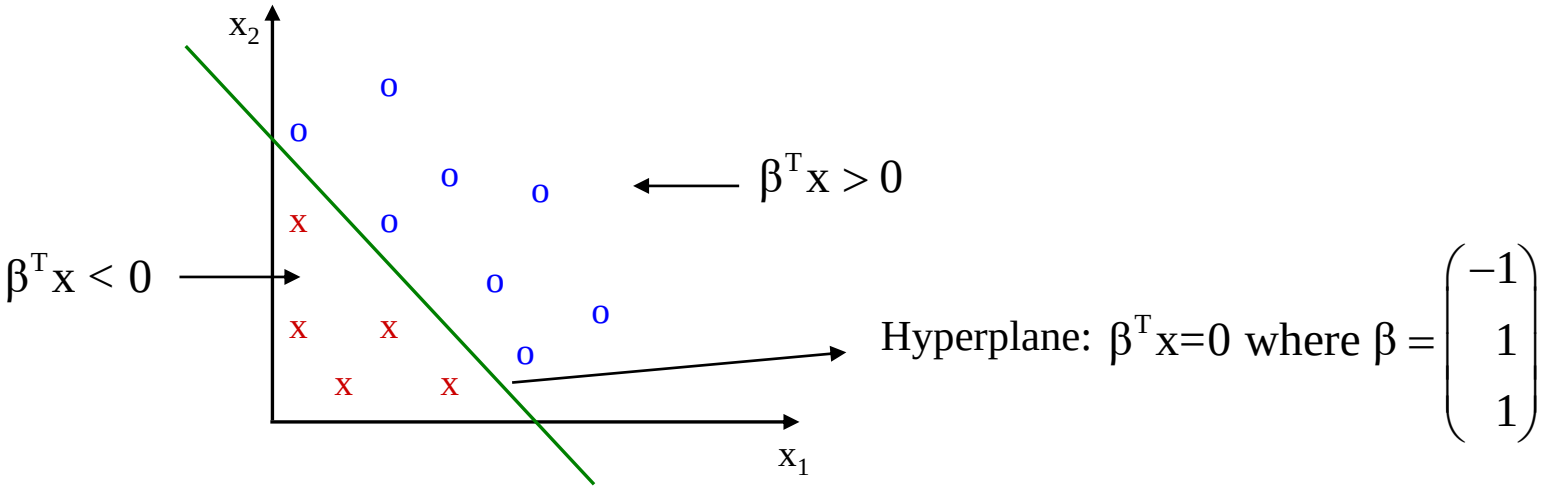
❖ 로지스틱 회귀분석 결과 및 해석

- 승산 비율: Odds Ratio
  - ✓나머지 변수는 모두 고정시킨 상태에서 한 변수를 1만큼 증가시켰을 때 변화하는 Odds의 비율

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

로지스틱 회귀분석: Interpretation

- Geometric interpretation
  - 로지스틱 회귀분석은 d차원의 데이터를 구분하는 (d-1)차원의 초평면을 찾는 것으로 이해할 수 있음



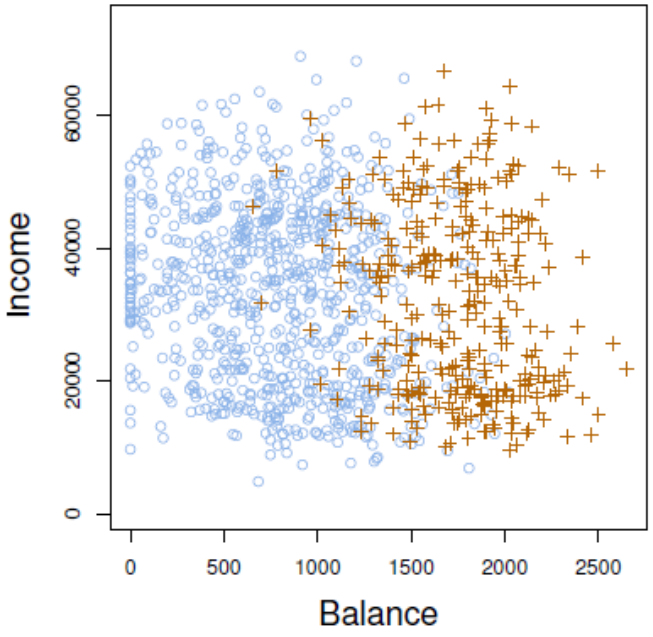
Classifier

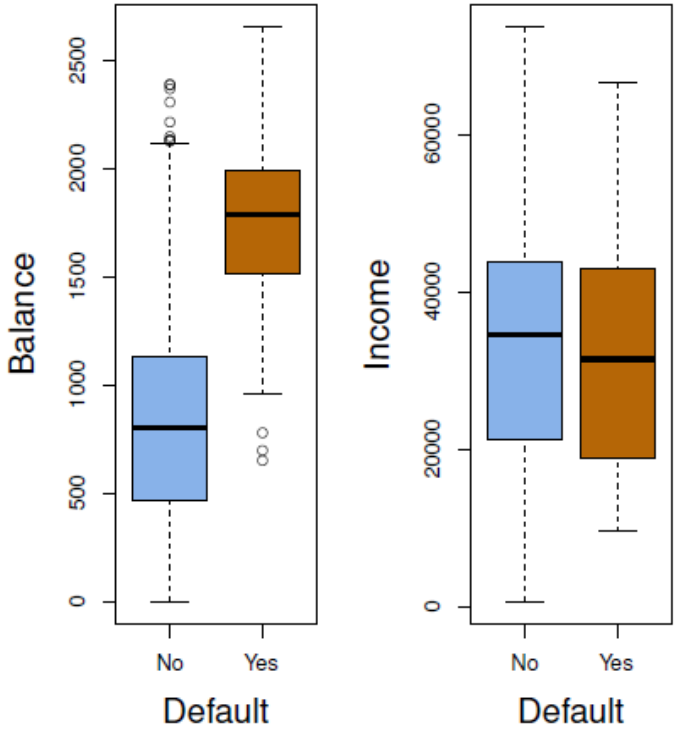
$$y = \frac{1}{\left(1+\exp(-\beta^T x)\right)}$$

$$\begin{pmatrix} y \to 1 & if & \beta^T x \to & \infty \\ y = \frac{1}{2} & if & \beta^T x = & 0 \\ y \to 0 & if & \beta^T x \to & -\infty \end{pmatrix}$$

로지스틱 회귀분석: 예시

❖ 신용카드 연체 예측





$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

로지스틱 회귀분석: 예시

❖ Credit Card Default: single variable

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-10.6513	0.3612	-29.5	< 0.0001
`balance`	0.0055	0.0002	24.9	< 0.0001

What is our estimated probability of `default` for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

With a balance of \$2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

로지스틱 회귀분석: 예시

❖ Credit Card Default: multiple variables

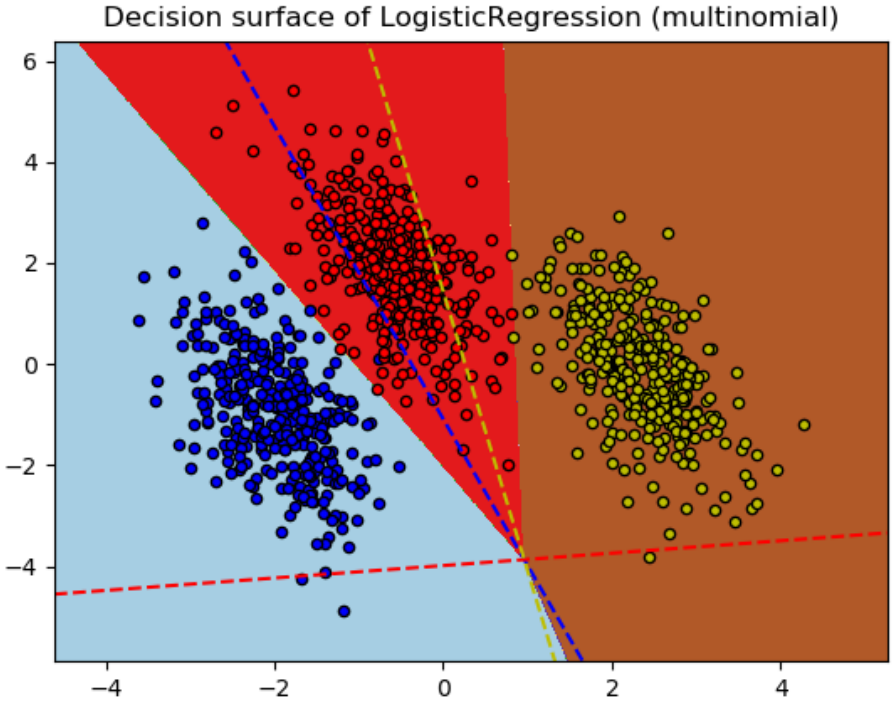
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

다항 로지스틱 회귀분석

- 지금까지의 로지스틱 회귀분석은 이범주 분류(Binary classification)를 풀기 위한 방식임
  - Q) 범주가 3개 이상인 다범주 분류에는 로지스틱 회귀분석을 어떻게 적용할 수 있을까?



http://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic_multinomial.html

다항 로지스틱 회귀분석

❖ 다항 로지스틱 회귀분석

- 기준(Baseline)이 되는 범주를 설정하고 이 범주 대비 다른 범주가 발생할 로그 승산을 회귀식으로 추정
- 예시) 범주가 3개인 분류 문제의 경우 아래 두 개의 회귀식에 대한 회귀 계수를 추정
  - ✓범주 3대비 범주 1의 발생 확률에 대한 로지스틱 회귀분석

$$log\left(\frac{p(y=1)}{p(y=3)}\right) = \hat{\beta_{10}} + \hat{\beta_{11}}x_1 + \hat{\beta_{12}}x_2 \cdots + \hat{\beta_{1d}}x_d = \boldsymbol{\beta}_{1.}^T\mathbf{x}$$

  - ✓범주 3 대비 범주 2의 발생 확률에 대한 로지스틱 회귀분석

$$log\left(\frac{p(y=2)}{p(y=3)}\right) = \hat{\beta_{20}} + \hat{\beta_{21}}x_1 + \hat{\beta_{22}}x_2 \cdots + \hat{\beta_{2d}}x_d = \boldsymbol{\beta}_{2.}^T\mathbf{x}$$

다항 로지스틱 회귀분석

❖ 다항 로지스틱 회귀분석

- 왜 범주는 3개인데 2개의 모형만 학습하는가? (일반화하면 K개의 범주가 있을 때, (K-1)개의 모형만 학습하는 이유는?
  - ✓각 범주에 속할 확률의 합은 항상 1이므로 나머지 K번째 범주에 대한 확률은 자동으로 산출됨

$$\frac{p(y=1)}{p(y=3)} = e^{\boldsymbol{\beta}_{1\cdot}^T \mathbf{x}} \qquad \frac{p(y=2)}{p(y=3)} = e^{\boldsymbol{\beta}_{2\cdot}^T \mathbf{x}}$$

$$p(y=1) + p(y=2) + p(y=3) = 1$$

$$p(y=3) \times e^{\boldsymbol{\beta}_{1\cdot}^T \mathbf{x}} + p(y=3) \times e^{\boldsymbol{\beta}_{2\cdot}^T \mathbf{x}} + p(y=3) = 1$$

$$p(y=3) = \frac{1}{1 + e^{\boldsymbol{\beta}_{1\cdot}^T \mathbf{x}} + e^{\boldsymbol{\beta}_{2\cdot}^T \mathbf{x}}}$$

다항 로지스틱 회귀분석

❖ 다항 로지스틱 회귀분석에서의 회귀계수 분석

- 개별 모형에 대해서 회귀 계수와 이에 대한 유의확률을 산출할 수 있음
  - ✓ Total phenols, Flavanoids, Monflavanoid penols, Hue, OD280~ 변수는 1 vs. 3, 2 vs. 3에서 모두 유의미한 변수로 나타남
  - ✓ Ash., Proanthocyanins 변수는 범주 1과 3을 구분할 때는 유의미하지 않으나 2와 3을 구분할 때 매우 유의미함

	1 vs 3		2 vs 3	
	Coefficient	p-value	Coefficient	p-value
(Intercept)	-223.7894	0.0000	340.9326	0.0000
Alcohol.2	19.6193	0.7880	-35.2596	0.6828
Malic.acid.	1.0581	0.9228	-0.3022	0.9899
Ash.	14.6800	0.3881	-204.7437	0.0000
Alcalinity.of.ash.	-20.3881	0.8815	-2.2832	0.9864
Magnesium.	2.0553	0.9975	2.1132	0.9974
Total.phenols.	-169.4205	0.0000	-40.3325	0.0000
Flavanoids.	193.7935	0.0000	16.2013	0.0188
Nonflavanoid.phenols	93.5409	0.0000	214.1837	0.0000
Proanthocyanins.	15.5178	0.1453	115.3184	0.0000
Color.intensity.	-16.6775	0.4212	-11.5066	0.7671
Hue	-50.0008	0.0000	352.7617	0.0000
OD280.OD315.of.diluted.wines.	75.2435	0.0000	84.2914	0.0000
Proline.	-0.0120	1.0000	-0.2899	0.9999

목차

Naïve Bayes & Logistic Regression: R Exercise

❖ Personal Loan Prediction

- 은행 고객의 인구통계학적 정보 및 은행상품 이용정보를 바탕으로 미래에 개인신용대출 상품을 이용할 고객 예측

Data Description:

ID	Customer ID
Age	Customer's Age in completed years
Experience	#years of professional experience
Income	Annual income of the customer ($000)
ZIPCode	Home Address ZIP code.
Family	Family size (dependents) of the customer
CCAvg	Avg. Spending on Credit Cards per month ($000)
Education	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
Mortgage	Value of house mortgage if any. ($000)
Personal Loan	Did this customer accept the personal loan offered in the last campaign?
Securities Account	Does the customer have a Securities account with the bank?
CD Account	Does the customer have a Certificate of Deposit (CD) account with the bank?
Online	Does the customer use internet banking facilities?
CreditCard	Does the customer use a credit card issued by UniversalBank?

R 실습: Naïve Bayesian Classification

❖ 분류모델 성능평가 함수 작성

```
# Performance Evaluation Function ---------------------------------------
perf_eval2 <- function(cm){
          # True positive rate: TPR
          TPR = cm[2,2]/sum(cm[2,])
          # True negative rate: TNR
          TNR = cm[1,1]/sum(cm[1,])
          # Simple Accuracy
          ACC = (cm[1,1]+cm[2,2])/sum(cm)
          # Balanced Correction Rate
          BCR = sqrt(TPR*TNR)
          return(c(TPR, TNR, ACC, BCR))
}
```

- 함수명: perf_eval2
  - ✓이범주 분류 성능 평가 함수
  - ✓함수 실행에 필요한 인자: confusion matrix (cm)
  - ✓함수 결과물: True positive rate, True negative rate, Simple accuracy, Balanced correction rate

R 실습: Naïve Bayesian Classification

❖ 분류모델 성능평가 함수 작성

```
perf_eval3 <- function(cm){
# Simple accuracy
ACC <- sum(diag(cm))/sum(cm)

# ACC for each class
A1 <- cm[1,1]/sum(cm[1,])
A2 <- cm[2,2]/sum(cm[2,])
A3 <- cm[3,3]/sum(cm[3,])
BCR <- (A1*A2*A3)^(1/3)

return(c(ACC, BCR))
}
```

- 함수명: perf_eval3
  - ✓3범주 분류 성능 평가 함수
  - ✓함수 실행에 필요한 인자: confusion matrix (cm)
  - ✓함수 결과물: Simple Accuracy, Balanced correction rate (BCR)

R 실습: Naïve Bayesian Classification

❖ 필요 패키지 설치

```
# Naive Bayesian Classifier ------------------------------------------------
# e1071 package install
install.packages("e1071", dependencies = TRUE)

# Call the e1071 package
library(e1071)
```

- Naïve Bayesian Classification: “e1071” 패키지에서 제공
- Logistic regression: R base에서 기본으로 제공하므로 별도의 패키지 필요 없음

R 실습: Naïve Bayesian Classification

❖ 데이터 불러오기 및 전처리

```
ploan <- read.csv("Personal Loan.csv")
input_idx <- c(2,3,4,6,7,8,9,11,12,13,14)
target_idx <- 10

ploan_input <- ploan[,input_idx]
ploan_target <- as.factor(ploan[,target_idx])
ploan_data <- data.frame(ploan_input, ploan_target)

# Split the data into the training/validation sets
set.seed(12345)
trn_idx <- sample(1:dim(ploan_data)[1], round(0.7*dim(ploan_data)[1]))
ploan_trn <- ploan_data[trn_idx,]
ploan_tst <- ploan_data[-trn_idx,]
```

- ID변수(1열), zip code(5열) 제거
- Naïve Bayesian classifier는 종속변수의 형태로 factor형을 요구함
- 70%의 학습 데이터와 30%의 테스트 데이터를 무작위로 선택

R 실습: Naïve Bayesian Classification

❖ Naïve Bayesian Classifier 학습

```
# Training the Naive Bayesian Classifier
nb_model <- naiveBayes(ploan_target ~ ., data = ploan_trn)
nb_model$apriori
nb_model$tables
```

- naiveBayes( ): Naïve Bayesian classifier 학습 함수
  - ✓첫 번째 인자: Formula
  - ✓두 번째 인자: 학습 데이터
  - ✓이 패키지에서는 모든 변수를 정규분포로 가정하고 각 범주-변수 조합별 평균 및 표준편차를 산출 → 별도의 파라미터가 존재하지 않음
- nb_model$apriori: 각 범주별 학습 데이터 수
- nb_model$tables: 각 범주-변수별 평균 및 표준편차가 저장된 테이블

R 실습: Naïve Bayesian Classification

❖ Naïve Bayesian Classifier 학습

```
# Training the Naive Bayesian Classifier
nb_model <- naiveBayes(ploan_target ~ ., data = ploan_trn)
nb_model$apriori
nb_model$tables
```

```
> nb_model$apriori
Y
   0    1
1573  177
```

→ prior distribution

```
> nb_model$tables
$Age
   Age
Y       [,1]     [,2]
  0 45.24984 11.49341
  1 44.72881 11.61039

$Experience
   Experience
Y       [,1]     [,2]
  0 20.03942 11.48734
  1 19.48023 11.73286

$Income
   Income
Y        [,1]     [,2]
  0  67.03242 41.28474
  1 143.28814 34.74226
```

→ [,1]: mean
[,2]: standard deviation

R 실습: Naïve Bayesian Classification

❖ Naïve Bayesian Classifier 성능 평가

```
# Predict the new input data based on Naive Bayesian Classifier
posterior = predict(nb_model, ploan_tst, type = "raw")
nb_prey = predict(nb_model, ploan_tst, type ="class")
```

- predict( ): 학습된 모형을 새로운 데이터에 적용하여 예측을 수행하는 함수
  - ✓첫 번째 인자: 학습된 모형
  - ✓두 번째 인자: 새로운 데이터
  - ✓세 번째 인자 (type): "raw": 각 범주에 속할 확률을 반환, "class" 예측된 범주를 반환

```
> posterior
                  0           1
 [1,] 9.999055e-01 9.449567e-05
 [2,] 9.998128e-01 1.871716e-04
 [3,] 9.998873e-01 1.127369e-04
 [4,] 9.998346e-01 1.654241e-04
 [5,] 9.996633e-01 3.367041e-04
 [6,] 7.584541e-01 2.415459e-01
 [7,] 9.421491e-01 5.785089e-02
 [8,] 9.996655e-01 3.345240e-04
 [9,] 9.996813e-01 3.186617e-04
[10,] 9.871787e-01 1.282129e-02
```

```
> nb_prey
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [17] 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 [33] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0
 [49] 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0
 [65] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 [81] 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
 [97] 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
```

R 실습: Naïve Bayesian Classification

❖ Naïve Bayesian Classifier 성능 평가

```
# Generate a confusion matrix
cfmatrix <- table(ploan_val$ploan_target, nb_prey)

# Evaluate the performance
perf_mat <- matrix(0, 4, 3)
perf_mat[,1] <- perf_eval(cfmatrix)
perf_mat
```

- table( ): 교차빈도표를 작성해주는 함수, 정답 범주와 모델에 의해 예측된 범주를 사용하여 confusion matrix 생성 가능

```
> cfmatrix
   nb_prey
      0   1
  0 615  56
  1  32  47
```

```
> perf_mat
           [,1] [,2] [,3]
[1,] 0.5949367    0    0
[2,] 0.9165425    0    0
[3,] 0.8826667    0    0
[4,] 0.7384340    0    0
```

- Naïve Bayesian Classifier는 대출 이용 고객에 대한 정확도가 **59.4%**로 낮은 편이며 단순 정확도는 **88.27%**, 균형정확도는 **0.7384**를 나타냄

R 실습: Logistic Regression

❖ 데이터 전처리: 정규화 및 데이터 분할

```
# Logistic Regression ------------------------------------------------------
# Conduct the normalization
ploan_input <- scale(ploan_input, center = TRUE, scale = TRUE)
ploan_target <- as.numeric(ploan_target)-1
ploan_data <- data.frame(ploan_input, ploan_target)

ploan_trn <- ploan_data[trn_idx,]
ploan_tst <- ploan_data[-trn_idx,]
```

- Gradient descent를 사용하는 알고리즘들은 정확한 수렴을 위하여 데이터 정규화가 필요 (scale 함수 사용)
- Naïve Bayesian classifier와는 달리 로지스틱 회귀분석은 분류문제임에도 불구하고 종속변수의 속성이 수치형(numeric)이어야 함
- (주의) factor를 바로 수치형으로 변환하면 예상과는 다른 결과가 나타날 가능성이 높음

R 실습: Logistic Regression

❖ 모든 변수를 사용하여 로지스틱 회귀분석 학습

```
# Train the Logistic Regression Model with all variables
full_lr <- glm(ploan_target ~ ., family=binomial, ploan_trn)
summary(full_lr)
```

- glm( ): R에서 기본 제공하는 generalized linear model 함수이며 로지스틱 회귀분석을 포함한 다양한 형태의 모형 학습 가능
  - ✓첫 번째 인자: Formula
  - ✓두 번째 인자: "family = binomial"로 설정해야 로지스틱 회귀분석이 학습됨
  - ✓세 번째 인자: 학습 데이터

R 실습: Logistic Regression

❖ 모든 변수를 사용하여 로지스틱 회귀분석 학습

```
# Train the Logistic Regression Model with all variables
full_lr <- glm(ploan_target ~ ., family=binomial, ploan_trn)
summary(full_lr)
```

- 신뢰수준 95%에서 유의미한 변수: Income, Family, CCAvg, Education, Securities.Account, CD.Account, CreditCard

```
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.21016    0.22999 -18.306  < 2e-16 ***
Age                -0.05479    1.06837  -0.051  0.95910
Experience          0.23514    1.06214   0.221  0.82480
Income              2.07961    0.17125  12.144  < 2e-16 ***
Family              0.80944    0.13411   6.036 1.58e-09 ***
CCAvg               0.30738    0.10800   2.846  0.00442 **
Education           1.13270    0.14325   7.907 2.63e-15 ***
Mortgage            0.07188    0.08685   0.828  0.40790
Securities.Account -0.44039    0.15266  -2.885  0.00392 **
CD.Account          0.94355    0.12160   7.760 8.52e-15 ***
Online             -0.13209    0.12191  -1.083  0.27859
CreditCard         -0.61753    0.15835  -3.900 9.63e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

R 실습: Logistic Regression

❖ 후방 소거법을 이용한 변수 선택

```
# Train the Logistic Regression Model with selected variables
reduced_lr <- step(full_lr, direction = "backward")
summary(reduced_lr)
```

- 총 11개의 변수 중 3개의 변수(Age, Mortgage, Online)가 제거됨
- Experience를 제외하고 신뢰수준 99%에서 모두 통계적으로 유의한 변수임

```
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          -4.1925     0.2279 -18.394  < 2e-16 ***
Experience            0.1838     0.1211   1.518 0.128956
Income                2.1047     0.1700  12.379  < 2e-16 ***
Family                0.8192     0.1337   6.128 8.92e-10 ***
CCAvg                 0.2990     0.1071   2.793 0.005229 **
Education             1.1224     0.1397   8.034 9.46e-16 ***
Securities.Account   -0.4336     0.1518  -2.856 0.004293 **
CD.Account            0.9123     0.1170   7.795 6.42e-15 ***
CreditCard           -0.6037     0.1571  -3.844 0.000121 ***
```

R 실습: Logistic Regression

❖ 모든 변수를 사용한 모형의 검증

```
# Evaluate the logistic regression performance on the validation data
# Case 1: full model
full_response <- predict(full_lr, type = "response", newdata = ploan_tst)
full_target <- ploan_tst$ploan_target
full_predicted <- rep(0, length(full_target))
full_predicted[which(full_response >= 0.5)] <- 1

cm_full <- table(full_target, full_predicted)
perf_mat[,2] <- perf_eval(cm_full)
```

- predict( )함수의 type = “response”로 설정하면 positive(1) 범주에 속할 확률을 반환함
- 본 실습에서는 로지스틱 회귀모형의 cut-off를 0.5로 설정하였으나 상황에 따라 이를 다르게 설정할 수 있음

R 실습: Logistic Regression

❖ 모든 변수를 사용한 모형의 검증

```
# Evaluate the logistic regression performance on the validation data
# Case 1: full model
full_response <- predict(full_lr, type = "response", newdata = ploan_tst)
full_target <- ploan_val$ploan_target
full_predicted <- rep(0, length(full_target))
full_predicted[which(full_response >= 0.5)] <- 1

cm_full <- table(full_target, full_predicted)
perf_mat[,2] <- perf_eval(cm_full)
```

- 모든 변수를 사용한 로지스틱 회귀분석의 경우 대출 고객에 대한 정확도는 **67%**, 단순 정확도 **96%**, BCR **0.8166**으로 나이브베이즈 분류기보다 향상된 예측 성능을 나타냄

```
> cm_full
           full_predicted
full_target   0   1
          0 667   4
          1  26  53
```

```
> perf_mat
          [,1]      [,2] [,3]
[1,] 0.5949367 0.6708861    0
[2,] 0.9165425 0.9940387    0
[3,] 0.8826667 0.9600000    0
[4,] 0.7384340 0.8166313    0
```

R 실습: Logistic Regression

❖ 후방소거법에 의해 선택된 변수를 사용한 모형의 검증

```
# Case 2: reduced model
reduced_response <- predict(reduced_lr, type = "response", newdata = ploan_tst)
reduced_target <- ploan_val$ploan_target
reduced_predicted <- rep(0, length(reduced_target))
reduced_predicted[which(reduced_response >= 0.5)] <- 1

cm_reduced <- table(reduced_target, reduced_predicted)
perf_mat[,3] <- perf_eval(cm_reduced)
colnames(perf_mat) <- c("Naive Bayes", "LR with all variables", "LR with selected
                         variables")
rownames(perf_mat) <- c("TPR", "TNR", "ACC", "BCR")
perf_mat
```

- 결과물에 대한 빠른 이해를 위해 colnames( ) 및 rownames( ) 함수를 사용하여 열과 행에 이름을 지정

R 실습: Logistic Regression

❖ 후방소거법에 의해 선택된 변수를 사용한 모형의 검증

- 변수선택을 수행한 모형은 전체 변수를 사용한 모형과 비교할 때 대출 이용 고객 2명을 정확히 예측하지 못하는 차이를 보임

```
> cfmatrix
   nb_prey
      0   1
0 615  56
1  32  47
```

```
> cm_full
             full_predicted
full_target   0   1
          0 667   4
          1  26  53
```

```
> cm_reduced
                reduced_predicted
reduced_target   0   1
             0 667   4
             1  28  51
```

- 그 결과, TPR 기준 2.5%p, 단순 정확도 기준 0.03%p, BCR 기준 0.015의 성능 저하를 보임

```
> perf_mat
    Naive Bayes LR with all variables LR with selected variables
TPR   0.5949367            0.6708861                  0.6455696
TNR   0.9165425            0.9940387                  0.9940387
ACC   0.8826667            0.9600000                  0.9573333
BCR   0.7384340            0.8166313                  0.8010750
```

R 실습: 다범주 분류

- Dataset: Wine dataset from UCI machine learning repository
  - 이탈리아의 특정 지역에서 생산된 세 품종의 포도를 이용하여 생산한 178종의 와인이 함유하는 구성 성분에 대한 데이터

UCI

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Wine Data Set

Download: Data Folder, Data Set Description

Abstract: Using chemical analysis determine the origin of wines

Data Set Characteristics:	Multivariate	**Number of Instances:**	178	**Area:**	Physical
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	13	**Date Donated**	1991-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	733901

Source:

Original Owners:

Forina, M. et al, PARVUS -
An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno,
16147 Genoa, Italy.

Donor:

Stefan Aeberhard, email: stefan '@' coral.cs.jcu.edu.au

R 실습: 다범주 분류

- Dataset: Wine dataset from UCI machine learning repository
  - 종속변수(1열): 와인을 생산한 포도의 품종
  - 설명변수(2-14열)
    - ✓ 1) Alcohol
    - ✓ 2) Malic acid
    - ✓ 3) Ash
    - ✓ 4) Alcalinity of ash
    - ✓ 5) Magnesium
    - ✓ 6) Total phenols
    - ✓ 7) Flavanoids
    - ✓ 8) Nonflavanoid phenols
    - ✓ 9) Proanthocyanins
    - ✓ 10)Color intensity
    - ✓ 11)Hue
    - ✓ 12)OD280/OD315 of diluted wines
    - ✓ 13)Proline

R 실습: 다범주 분류

❖ 데이터 불러오기 및 성능 비교 행렬 초기화

```
# Part 2: Multi-class classification
wine <- read.csv("wine.csv")

# Initialize the performance matrix
perf_mat_wine <- matrix(0, 2, 2)
colnames(perf_mat_wine) <- c("Naive Bayes", "Multinomial Logistic Regression")
rownames(perf_mat_wine) <- c("ACC", "BCR")
```

- 와인 데이터셋 불러오기
- 성능 평가는 두 가지 관점에서 수행: 단순 정확도 및 균형 정확도
  - ✓세 개 이상의 범주에서는 Taget Class가 달라지면 F1지표가 달라지므로 성능 평가 지표에서 제외

R 실습: 다범주 분류

❖ Naïve Bayes 학습

```
# Naive Bayes
wine$Class <- as.factor(wine$Class)
# Split the data into the training/validation sets
set.seed(12345)
trn_idx <- sample(1:dim(wine)[1], round(0.7*dim(wine)[1]))
wine_trn <- wine[trn_idx,]
wine_tst <- wine[-trn_idx,]
# Training the Naive Bayesian Classifier
nb_model <- naiveBayes(Class ~ ., data = wine_trn) nb_model$apriori
nb_model$tables
```

- Naïve Bayes 분류기는 범주에 대한 정보를 factor형으로 받으므로 이에 맞게 변수 속성 수정
- 이범주 분류에서와 마찬가지로 각 범주-변수별로 평균 및 표준편차를 계산함

R 실습: 다범주 분류

❖ Naïve Bayes 학습

- 이범주 분류에서와 마찬가지로 각 범주-변수별로 평균 및 표준편차를 계산함
  - ✓학습 데이터의 범주별 객체 수(좌) 및 범주-변수별 평균 및 표준편차(우)

```
> nb_model$apriori
Y
 1  2  3
41 50 34
```

```
> nb_model$tables
$Alcohol.2
   Alcohol.2
Y       [,1]      [,2]
  1 13.77585 0.4179353
  2 12.38420 0.4998738
  3 13.17706 0.5185150

$Malic.acid.
   Malic.acid.
Y       [,1]      [,2]
  1 1.950244 0.6588607
  2 1.952600 1.0300125
  3 3.098824 1.0735519

$Ash.
   Ash.
Y       [,1]      [,2]
  1 2.455854 0.2320342
  2 2.264600 0.2947653
  3 2.445000 0.1931909
```

R 실습: 다범주 분류

❖ 학습된 모델로 테스트 수행

```
# Predict the new input data based on Naive Bayesian Classifier
posterior = predict(nb_model, wine_tst, type = "raw")
options(scipen=10)
posterior[1:10,]
```

- options(scipen=10): 숫자를 출력할 때 scientific notation이 아닌 소수점 자리수로 표현

```
> posterior[1:10,]
             1           2           3
 [1,] 1.0000000 1.022948e-08 1.939191e-31
 [2,] 1.0000000 3.166779e-18 2.823132e-45
 [3,] 0.9331474 6.685265e-02 9.564332e-16
 [4,] 1.0000000 6.021244e-10 6.442364e-27
 [5,] 1.0000000 2.217728e-14 2.704115e-45
 [6,] 1.0000000 2.783036e-08 1.200844e-23
 [7,] 1.0000000 4.248183e-10 2.207826e-29
 [8,] 1.0000000 5.078834e-13 3.977007e-32
 [9,] 0.9999946 5.394226e-06 1.315012e-23
[10,] 0.9999999 5.227292e-08 1.082562e-31
```

R 실습: 다범주 분류

❖ 학습된 모델로 테스트 수행

```
nb_prey = predict(nb_model, wine_tst, type ="class")
# Generate a confusion matrix
cfmatrix <- table(wine_tst$Class, nb_prey)
cfmatrix perf_mat_wine[,1] <- perf_eval3(cfmatrix)
perf_mat_wine
```

- 3 by 3 confusion matrix 생성 및 성능 지표 산출

```
> cfmatrix
   nb_prey
     1  2  3
  1 16  2  0
  2  0 21  0
  3  0  0 14
```

```
> perf_mat_wine
    Naive Bayes Multinomial Logistic Regression
ACC   0.9622642                                0
BCR   0.9614997                                0
```

R 실습: 다범주 분류

❖ 다항 로지스틱 회귀분석 학습

```
# Multinomial logistic regression
install.packages("nnet")
library(nnet)
# Define the baseline class
wine$Class <- relevel(wine$Class, ref = "3")
wine_trn <- wine[trn_idx,]
wine_tst <- wine[-trn_idx,]
# Train multinomial logistic regression
ml_logit <- multinom(Class ~ ., data = wine_trn)
```

- 다항 로지스틱 회귀분석은 nnet 패키지의 multinom 함수 사용
- relevel(arg1, arg2): Baseline 범주를 지정하는 함수
  - ✓arg1: 원래 데이터의 범주 컬럼
  - ✓arg2: Baseline으로 사용할 범주 지정
- multinom(formula, data): formula에 제시된 다항 로지스틱 회귀분석의 회귀계수를 data를 사용하여 학습

R 실습: 다범주 분류

❖ 다항 로지스틱 회귀분석 학습

```
# Check the coefficients
summary(ml_logit)
t(summary(ml_logit)$coefficients)
```

- summary( ) 함수는 각 회귀분석의 회귀계수와 표준편차를 제공

```
> summary(ml_logit)
Call:
multinom(formula = Class ~ ., data = wine_trn)

Coefficients:
  (Intercept) Alcohol.2 Malic.acid.       Ash. Alcalinity.of.ash. Magnesium. Total.phenols. Flavanoids.
1   -223.7894  19.61933   1.0580849   14.68003         -20.388079   2.055332     -169.42046   193.79355
2    340.9326 -35.25956  -0.3022464 -204.74374          -2.283177   2.113213      -40.33249    16.20125
  Nonflavanoid.phenols Proanthocyanins. Color.intensity.       Hue OD280.OD315.of.diluted.wines.    Proline.
1             93.54088         15.51784        -16.67753 -50.00078                       75.24351 -0.01203835
2            214.18367        115.31843        -11.50662 352.76166                       84.29139 -0.28993650

Std. Errors:
  (Intercept) Alcohol.2 Malic.acid.     Ash. Alcalinity.of.ash. Magnesium. Total.phenols. Flavanoids.
1    5.654384  72.95880    10.91696 17.00797           136.7347   652.2961      15.020358   15.803357
2    6.939562  86.27172    23.99149 13.80019           133.4915   651.5538       4.710628    6.894325
  Nonflavanoid.phenols Proanthocyanins. Color.intensity.      Hue OD280.OD315.of.diluted.wines. Proline.
1             2.552353         10.656309         20.73537 5.832211                       17.616282 4109.541
2             4.025819          3.763046         38.85249 5.124116                        8.260529 4109.592
```

R 실습: 다범주 분류

❖ 다항 로지스틱 회귀분석 학습

```
# Check the coefficients
summary(ml_logit)
t(summary(ml_logit)$coefficients)
```

- 모형별 회귀계수만을 출력

```
> t(summary(ml_logit)$coefficients)
                                      1            2
(Intercept)                -223.78940771  340.9325762
Alcohol.2                    19.61933356  -35.2595638
Malic.acid.                   1.05808493   -0.3022464
Ash.                         14.68003170 -204.7437391
Alcalinity.of.ash.          -20.38807892   -2.2831770
Magnesium.                    2.05533182    2.1132125
Total.phenols.             -169.42046037  -40.3324875
Flavanoids.                 193.79354912   16.2012512
Nonflavanoid.phenols         93.54087708  214.1836744
Proanthocyanins.             15.51784015  115.3184288
Color.intensity.            -16.67753036  -11.5066163
Hue                         -50.00078306  352.7616638
OD280.OD315.of.diluted.wines. 75.24350827   84.2913856
Proline.                      -0.01203835   -0.2899365
```

R 실습: 다범주 분류

❖ 다항 로지스틱 회귀분석 학습

```
# Conduct 2-tailed z-test to compute the p-values
z_stats <- summary(ml_logit)$coefficients/summary(ml_logit)$standard.errors
t(z_stats)

p_value <- (1-pnorm(abs(z_stats), 0, 1))*2
options(scipen=10)
t(p_value)
```

- multinom( )함수는 유의확률(p-value)을 산출하지 않으므로 **z-test**를 통해 이를 계산

```
> t(z_stats)
                                 1             2
(Intercept)            -39.578031925993  49.12883214048
Alcohol.2                0.268909763073  -0.40870360627
Malic.acid.              0.096921223161  -0.01259806372
Ash.                     0.863126616468 -14.83629776773
Alcalinity.of.ash.      -0.149106857745  -0.01710353294
Magnesium.               0.003150918424   0.00324334331
Total.phenols.         -11.279389133104  -8.56201835550
Flavanoids.             12.262809308770   2.34994003780
Nonflavanoid.phenols    36.648876364813  53.20250467051
Proanthocyanins.         1.456211559167  30.64497158210
Color.intensity.        -0.804303408791  -0.29616165373
Hue                     -8.573212736721  68.84342297985
OD280.OD315.of.diluted.wines.  4.271247881180  10.20411469128
Proline.                -0.000002929367  -0.00007055116
```

```
> t(p_value)
                                 1             2
(Intercept)            0.00000000000 0.00000000
Alcohol.2              0.78799912377 0.68275719
Malic.acid.            0.92278895451 0.98994847
Ash.                   0.38806785529 0.00000000
Alcalinity.of.ash.     0.88146931469 0.98635402
Magnesium.             0.99748593500 0.99741219
Total.phenols.         0.00000000000 0.00000000
Flavanoids.            0.00000000000 0.01877644
Nonflavanoid.phenols   0.00000000000 0.00000000
Proanthocyanins.       0.14533414508 0.00000000
Color.intensity.       0.42122176949 0.76710663
Hue                    0.00000000000 0.00000000
OD280.OD315.of.diluted.wines.  0.00001943822 0.00000000
Proline.               0.99999766270 0.99994371
```

R 실습: 다범주 분류

❖ 다항 로지스틱 회귀분석 학습

```
cbind(t(summary(ml_logit)$coefficients), t(p_value))
```

- 모형별 회귀계수와 유의확률 출력

```
> cbind(t(summary(ml_logit)$coefficients), t(p_value))
                                      1           2             1           2
(Intercept)                -223.78940771  340.9325762 0.00000000000 0.00000000
Alcohol.2                    19.61933356  -35.2595638 0.78799912377 0.68275719
Malic.acid.                   1.05808493   -0.3022464 0.92278895451 0.98994847
Ash.                         14.68003170 -204.7437391 0.38806785529 0.00000000
Alcalinity.of.ash.          -20.38807892   -2.2831770 0.88146931469 0.98635402
Magnesium.                    2.05533182    2.1132125 0.99748593500 0.99741219
Total.phenols.             -169.42046037  -40.3324875 0.00000000000 0.00000000
Flavanoids.                 193.79354912   16.2012512 0.00000000000 0.01877644
Nonflavanoid.phenols         93.54087708  214.1836744 0.00000000000 0.00000000
Proanthocyanins.             15.51784015  115.3184288 0.14533414508 0.00000000
Color.intensity.            -16.67753036  -11.5066163 0.42122176949 0.76710663
Hue                         -50.00078306  352.7616638 0.00000000000 0.00000000
OD280.OD315.of.diluted.wines.  75.24350827   84.2913856 0.00001943822 0.00000000
Proline.                       -0.01203835   -0.2899365 0.99999766270 0.99994371
```

회귀계수 (1 vs. 3)	회귀계수 (2 vs. 3)	유의확률 (1 vs. 3)	유의확률 (2 vs. 3)

R 실습: 다범주 분류

❖ 다항 로지스틱 회귀분석 학습

```
# Predict the class probability
ml_logit_haty <- predict(ml_logit, type="probs", newdata = wine_tst)
ml_logit_haty[1:10,]
```

- predict( )함수의 type = "probs" 옵션을 사용하면 각 테스트 객체마다 각 범주에 속할 확률을 반환함

```
> ml_logit_haty[1:10,]
                3 1             2
3   6.441185e-91 1  3.279515e-69
4   1.195594e-59 1 8.384095e-146
5   9.660673e-42 1  1.002201e-14
9  1.428128e-103 1  2.389442e-90
11  6.343487e-88 1  4.980457e-82
12  5.478659e-63 1  2.089522e-89
13  6.344532e-62 1  1.802906e-93
16  3.119966e-49 1  8.562155e-93
20 1.684647e-128 1 8.581093e-115
21 5.983873e-132 1  9.329299e-29
```

R 실습: 다범주 분류

❖ 다항 로지스틱 회귀분석 학습

```
# Predict the class label
ml_logit_prey <- predict(ml_logit, newdata = wine_tst)
cfmatrix <- table(wine_tst$Class, ml_logit_prey)
cfmatrix perf_mat_wine[,2] <- perf_eval3(cfmatrix)
perf_mat_wine
```

- predict( )함수의 type = "probs" 옵션을 사용하지 않으면 가장 확률이 높은 범주를 반환함

```
> cfmatrix
   ml_logit_prey
      3  1  2
  3 14  0  0
  1  0 18  0
  2  1  1 19
```

- Naïve Bayes와 다항 로지스틱 회귀분석 모두 0.96 근처의 ACC와 BCR 값을 나타냄

```
> perf_mat_wine
    Naive Bayes Multinomial Logistic Regression
ACC   0.9622642                       0.9622642
BCR   0.9614997                       0.9671892
```

Q & A