3강. 타겟마케팅

서울대학교
통계학과
김용대

목차

1. 타겟마케팅 데이터 예제

buytest dataset

- **개요** : 소매업 회사가 잠재 고객들에게 광고 인쇄물 (DM, Direct Mail)을 발송하고자 함.

- **입력변수** : 지난 24개월간 60$ 이상의 상품을 구입한 고객 약 10,000명에 대해 인구 속성, 거래 회수, 거래 속성 등을 입력변수로 구성.

- **반응변수** : 이들 고객에게 DM을 발송하여 3개월 간의 캠페인을 진행한 후, DM에 대한 응답여부를 조사하여 목표변수를 생성.

- **분석 목표** : 전체 고객 중 메일에 응답할 가능성이 높은 고객의 특성 분석

buytest dataset

	ID	RESPOND	AGE	INCOME	SEX	MARRIED	FICO	OWNHOME	LOC	CLIMATE	BUY6	BUY12	BUY18	VALUE24	ORGSRC	DISCBUY	RETURN24	COA6	C1	C2	C3	C4	C5	C6	C7	PURCHTOT
1	001371057	0	71	67	M	1	719	0	A	10	1	1	1	318	O	1	0	0	0	0	0	0	0	0	0	0
2	002093270	0	53	72	M	1	751	0	A	10	0	0	0	83	R	0	0	0	0	0	0	0	0	0	0	0
3	002783726	0	53	70	F	1	725	0	A	10	1	1	1	265	D	0	0	0	0	0	0	0	0	0	0	0
4	010800860	0	45	56	F	0	684	0	A	10	0	0	1	448	O	1	0	0	0	0	0	0	0	0	0	0
5	014577797	0	32	66	F	0	651	0	A	10	0	0	0	161	R	0	0	0	0	0	0	0	0	0	0	0
6	015884859	0	35	48	F	0	691	1	A	10	0	0	0	250	C	0	1	0	0	0	0	0	0	0	0	0
7	017131376	0	43	49	F	0	694	1	A	10	0	0	0	194	R	0	1	0	0	0	0	0	0	0	0	0
8	018674857	0	39	64	M	0	659	0	A	10	0	0	0	446	D	0	0	0	0	0	0	0	0	0	0	0
9	019417226	0	66	65	M	0	692	0	A	10	0	0	0	214	O	1	0	0	0	0	0	0	0	0	0	0
10	021786286	0	NA	NA		NA	707	NA	A	10	0	0	0	198	O	0	0	0	0	0	0	0	0	0	0	0
11	026897464	0	52	58	M	1	705	1	A	10	0	1	2	216	C	0	0	0	0	0	0	0	0	0	0	0
12	028908796	0	29	40	F	0	693	0	A	10	0	0	0	118	C	0	0	0	0	0	0	0	0	0	0	0

<buytest dataset 개요>

buytest dataset - 변수명

- ID : 고객 번호
- RESPOND : DM에 대한 반응 여부
- AGE : 나이(년)
- INCOME : 연수입 (단위 : 천달러)
- SEX : F : 여자, M : 남자
- MARRIED : 1: 결혼, 0 : 미혼
- FICO : 신용점수
- OWNHOME : 자가 주택 소유 여부 (1: 소유)
- LOC : 거주지 (A–H)
- CLIMATE : 거주지의 기온 (10, 20, 30 ℃)
- BUY6,12,18 : 최근 6,12,18개월 간의 구입 횟수
- VALUE24 : 지난 24개월 간의 구입총액
- ORGSRC : 고객 분류
- DISCBUY : 할인고객 여부 (1: 할인고객)
- RETURN24 : 지난 24개월 간 상품의 반품 여부
- COA6 : 6개월 간의 주소변경 여부 (1: 주소변경)
- C1~C7 : DM에 의한 품목별 구입액
- PURCHTOT : DM에 의한 구입 총액

buytest dataset - 분석

- buytest dataset을 이용한 타겟 마케팅 고객층 분석

  - RESPOND 변수를 목표변수로 설정하여 로지스틱 회귀 모형 설정.

    -> DM에 반응할 것이라고 기대되는 고객들을 대상으로만 메일을 발송할 수 있다!

  - PURCHTOT 변수를 목표변수로 설정하여 선형 회귀 모형 설정.

    -> DM에 반응한 고객 중에서도 구매 액수가 특별히 큰 고객의 특성을 찾을 수 있다!

  - C1~C7 변수를 목표변수로 설정하여 선형 회귀 모형 설정.

    -> 각 품목별로 수요층을 조사함으로써 고객의 특성에 맞는 DM을 구성할 수 있다!

2. 로지스틱 회귀모형

로지스틱 회귀분석

- **로지스틱 회귀모형**
  - 출력변수가 범주형 변수인 경우 (분류문제)에 사용하는 대표적인 회귀모형
  - 범주가 2가지인 경우를 고려하자.
    - $Y = 1$: 출력변수가 첫 번째 범주에 속할 경우.
    - $Y = 0$: 출력변수가 두 번째 범주에 속할 경우.
  - 목적 : 입력변수와 범주형인 출력변수간의 관계를 잘 표현할 수 있는 모형 구축

로지스틱 회귀분석

- **모형**
  - $P(Y = 1 \mid X) = F(X^T\beta)$
  - $F(x)$는 연속이고 증가하며 0과 1사이에서 값을 갖는 함수

- **여러 가지** $F(x)$
  - 로지스틱 모형: $F(x) = \exp(x)/(1 + \exp(x))$
  - 곰배르츠 모형: $F(x) = \exp(-\exp(x))$
  - 프로빗 모형: $F(x)$가 표준정규분포의 분포함수(distribution function)

로지스틱 회귀분석

- 이중 로지스틱 모형이 계산의 편이성으로 가장 널리 사용됨.

- 로지스틱 회귀모형
  - $P(Y=1|X=x)=\frac{\exp(x^T\beta)}{1+\exp(x^T\beta)}, \ \left(\beta=(\beta_1,\beta_2,\dots,\beta_p)^T\right)$
  - i.e. $\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right)=x^T\beta$

로지스틱 회귀분석

- **모수의 추정 (최대 우도 추정)**
  - 모수 : β
  - 자료 : $(y_1, \mathrm{x}_1), \ldots, (y_n, \mathrm{x}_n)$
  - 최대 우도 추정량(Maximum likelihood estimator) $\hat{\beta}$
    - 우도 함수를 최대로 하는 모수값
  - 우도함수:
    - $L(\beta) = \prod_{i=1}^{n} F(\mathrm{x}^T\beta)^{y_i} \times (1 - F(\mathrm{x}^T\beta))^{1-y_i}$ where $F(x) = \frac{exp(x)}{1+exp(x)}$
    - 우도 함수의 최대화는 수치적 방법(numerical method)를 사용하여 구한다.
      - 예 : Newton-raphson방법

로지스틱 회귀분석

- **예측 및 모형의 해석**
  - 예측
    - $\hat{P}(Y=1|X=x)=\frac{\exp(\widehat{\beta_0}+\widehat{\beta_1}\times x)}{1+\exp(\widehat{\beta_0}+\widehat{\beta_1}\times x)}$
    - $\hat{P}(Y=1|X=x)>0.5$이면 1로 예측
    - $\hat{P}(Y=1|X=x)<0.5$이면 0으로 예측
  - 해석
    - $\beta_1>0: x$가 증가하면 $P(Y=1|X=x)$가 증가한다.
    - $\beta_1<0: x$가 증가하면 $P(Y=1|X=x)$가 감소한다.

로지스틱 회귀분석

- **회귀계수와 오즈비**
  - 오즈(odds)

$$P(Y = 1|x)/P(Y = 0|x)$$

  - 오즈비 (odds ratio)

$$\frac{P(Y = 1|x + 1)P(Y = 0|x)}{P(Y = 0|x + 1)P(Y = 1|x)}$$

  - 성질

$$\text{오즈비} = \exp(\beta)$$

로지스틱 회귀분석

- **오즈비의 의미**
  - X가 한 단위 증가 할 때 y=1일 확률과 y=0일 확률의 비가 증가하는 양
  - 예
    - x는 소득이고 y는 어떤 상품에 대한 구입여부 (1=구입, 0=미구입)
    - b=3.72
    - 소득이 한 단위 증가하면 물품을 구매하지 않을 확률에 대한 구매할 확률의 비(오즈비)가 exp(3.72)=42배 증가함을 의미한다.

로지스틱 회귀분석

- **예제**
  - 회사채의 신용등급 (안정, 위험)과 여러 가지 재무자료들 사이의 관계를 규명한다.
  - 입력변수: 여러 재무자료

    x1: 자산대비 부채현황 지표

    x2 : 현금회전율

    x3: 종업원 수 (50인 이하 = 0, 50-100인 = 1, 100인 이상 = 2)
  - 출력변수: 회사채 신용등급: 안정 = 1, 위험 = 0

로지스틱 회귀분석

- **예제: 가변수 생성**
  - x3변수가 질적변수 이므로 가변수가 필요하다.
  - x3를 위한 가변수 z1과 z2

	z1	z2
50인 이하	0	0
50-100인	1	0
100인 이상	0	1

로지스틱 회귀분석

- **예제: 회귀계수 추정**

  회귀계수의 추정 결과는 다음과 같다.

입력변수	회귀계수	표준오차	Chi-square	유의확률
Intercept	12.2216	3.5942	11.5623	0.000
X1	-1.0727	0.3933	7.4389	0.006
X2	11.5243	4.1313	7.7814	0.005
X3-1	0.6981	0.3571	3.9404	0.047
X3-2	2.6141	0.7915	10.9078	0.001

로지스틱 회귀분석

- **예제: 결과해석**
  - X1의 회귀계수의 부호가 음수인데, 이는 부채비율이 높을 수록 회사채의 신용이 낮음을 의미한다.
  - X2의 회귀계수의 부호가 양수인데, 이는 현금회전율이 높을 수록 회사채의 신용이 높아짐을 의미한다.
  - X3-1과 x3-2는 가변수 z1과 z2의 회귀계수의 추정치를 나타낸다.

로지스틱 회귀분석

- **예제: 결과해석 (이어서)**
  - X3-1의 회귀계수로부터 50-100인 규모의 회사와 50인 미만의 회사의 회사채 신용등급의 오즈비를 구할 수 있다. 이는 exp(0.6981)=2.00으로 50-100인 규모의 회사의 회사채가 50인 미만의 회사채에 비하여 약 2배정도 신용이 좋다고 볼 수 있다.
  - 마찬가지로, X3-2의 회귀계수로부터 100인 이상 규모의 회사와 50인 미만의 회사의 회사채 신용등급의 오즈비를 구할 수 있다. 이는 exp(2.6141)=13.65으로 100인 이상 규모의 회사의 회사채가 50인 미만의 회사채에 비하여 약 13.65배 정도 신용이 좋다고 볼 수 있다.

불균형 자료 분석방법

- 많은 분류문제에서 모집단에서 두 그룹의 크기가 현저히 다른 경우가 종종 발생한다.
- 예제
  - 부도예측
  - FDS (Fraud Detection System)
  - 이탈방지
  - 암진단
  - ….

불균형 자료 분석방법

- 모집단이 불균형이 된 경우, 임의추출법으로 자료를 구성하면, 작은 그룹의 자료의 수가 매우 작을 수 있어서 분석에 많은 문제가 생김 (예: 파워가 너무 작다).
- 이런 경우에는 임의추출법을 사용하지 않고 흔히 **case-control sampling**을 사용.
- Case-control sampling은 역학에서 주로 사용되는 방법임.

불균형 자료 분석방법

- **Case-control sampling**
  - 예: 흡연이 암에 미치는 영향에 대한 연구
    - 일반적 sampling : 흡연자 100명과 비흡연자 100명을 추출하여 어느 쪽이 암 발생 확률이 높은지를 조사한다.
    - 문제점 : 암의 발생확률이 매우 낮아서 200명의 자료 중 아무도 암에 걸리지 않을 확률이 매우 높다.
    - 해결책 : 암환자 100명과 건강한 사람 100명을 추출하여 흡연 유무를 조사한다.

불균형 자료 분석방법

- **Case-control sampling**
  - 암환자와 건강한 사람의 분포, 부도회사와 건전한 회사와의 분포
    - 두 그룹 분류문제의 경우 첫 번째 그룹에서 n1명의 자료를 임의추출법을 사용하여 추출하고 두 번째 그룹에서 n2명의 자료를 임의 추출한다.
    - 그리고, 이 두 자료를 모두 사용하여 분석한다.

불균형 자료 분석방법

- **Case-control sampling**
  - Case-control sampling으로 추출된 자료는 원자료와 그 분포가 매우 상이하여 분석 결과의 해석에 매우 조심하여야 한다.
  - 두 그룹 분류 문제를 생각하자.
    - P(x)를 원자료에서 입력변수 x가 주어진 경우의 자료가 두 번째 그룹에 속할 확률이라 하자. (1-P(x)는 입력변수 x가 주어진 경우의 자료가 첫 번째 그룹에 속할 확률이다)
    - Q(x)는 사후추출법으로 추출된 자료에서 입력변수 x가 주어진 경우의 자료가 두 번째 그룹에 속할 확률이라 하자. (1-Q(x)는 입력변수 x가 주어진 경우의 자료가 첫 번째 그룹에 속할 확률이다)

불균형 자료 분석방법

- **Case-control sampling**
  - 일반적으로 P(x)와 Q(x)는 매우 다르다.
  - Case-control sampling으로 추출된 자료를 이용하면 확률 Q(x)를 추정하게 된다.
  - 하지만, 관심사는 원자료에서의 확률 P(x)이다.
  - 원자료에서 두 그룹의 분포를 아는 경우에는 이 정보를 이용하여 Q(x)로 부터 P(x)를 구할 수 있다.
  - 많은 분류문제에서는 P(x)의 정확한 값보다는 두 개의 주어진 입력변수 x1과 x2에 대하여 P(x1)과 P(x2)의 대소를 비교하는 것을 목적으로 한다.

불균형 자료 분석방법

- **Case-control sampling**
  - 예: DM (Direct Mail) 발송
    - 한 회사에서 전체 고객 10만 명 중 구매확률이 높을 것으로 예상되는 10000명의 고객에게 DM을 발송하려고 한다.
    - 이 경우, 알아야 할 사항은 각 고객의 상품구매확률이 아니라, 전체 고객 중 구매력이 큰 10000명이 고객이 누군가 하는 것이다.
    - 즉, P(x)의 값은 알려지지 않아도, 전체 고객 10만 명에 대하여 P(x)가 상위 10000명인 고객은 알 수 있으면 된다.

불균형 자료 분석방법

- **Case-control sampling**
  - 원자료에서 첫 번째 그룹에 속하는 자료의 수를 N1, 두 번째 그룹에 속하는 자료의 수를 N2라 하자.
  - 추출된 자료에서 첫 번째 그룹에 속하는 자료의 수를 n1, 두 번째 그룹에 속하는 자료의 수를 n2라 하자.
  - 그러면, 다음의 관계가 성립된다.

$$\frac{P(x)}{1-P(x)} = \frac{Q(x)}{1-Q(x)} \times \frac{N2 \times n1}{N1 \times n2}$$

불균형 자료 분석방법

- **Case-control sampling**
  - 원자료의 정보가 없는 경우에도, Q(x)를 이용하여 P(x)의 대소를 알 수 있다.
  - Q(x)와 P(x)사이에 다음의 관계가 성립된다.
  - 주어진 두 개의 입력변수 x1과 x2에 대하여 만약 Q(x1)<Q(x2)이면 P(x1)<P(x2)를 만족한다.
  - 앞의 DM 예제에서, P(x)가 큰 상위 10000명의 선택은 Q(x)가 큰 상위 10000명과 같다.
  - Q(x)와 같이 원자료의 P(x)와 대소가 같은 값을 스코어(score)라고 부른다.

불균형 자료 분석방법

- **로지스틱 회귀모형과 Case-control sampling**
  - 집단의 p(y=1)에 대한 사전정보를 p1이라 하자.
  - 정리
    - 자료에서의 입력변수와 출력변수와 관계가

$$Q(x) = \exp(a + bx) / (1 + \exp(a + bx))$$

라 하자. 모집단의 사전정보를 고려하면 모집단에서의 입력변수와 출력변수와 관계는

$$\underline{P(x) = \exp(a^* + bx) / (1 + \exp(a^* + bx))}$$

가 된다. 이때, $a^* = a + \log(p1/(1-p1)) - \log(n1/n0)$. 여기서 n0와 n1은 자료에서 0인 그룹과 1인 그룹의 자료의 수이다.

3. 모형선택

모형선택

- **모형의 선택 (변수 선택)**
  - 일반적으로 많은 입력 변수 중에서 출력 변수에 영향을 미치는 변수는 그리 많지 않다.
  - 목표:
    - 많은 입력 변수 중에서 출력 변수에 영향을 미치는 소수의 입력 변수를 찾아내는 것.

모형선택

- **모형 선택이 필요한 이유**
  - 많은 입력 변수를 관리하는데 필요한 노력과 비용 절약
  - 적절한 모형의 복잡도(변수의 개수) 유지로 예측 오차 감소 가능

❖ 유의한 입력변수를 모형에 넣지 않으면, 중요한 정보를 놓칠 수 있으며, 또한 그 결과가 편향된다. 따라서 변수 선택에 있어서는 너무 많은 변수를 사용해서 생기는 문제를 피하고, 동시에 중요한 변수를 놓쳐서 생기는 정보의 손실을 최소로 해야 한다. 즉, 적당한 수의 변수를 선택해야 한다.

모형선택

- **학습자료와 예측자료**

학 습 자 료

모형을 구축하기
위하여 사용한 자료
(표 본)

예 측 자 료

구축된 모형을
이용하여 예측을
하고자 하는 자료
(모 집 단)

모형선택

- **학습자료와 예측자료의 예제**

A 은행의 신용 평가

학습자료
현재 A은행의 고객자료로써 각 고객의 신용등급(출력)이 알려짐

예측자료
미래의 가망 고객자료로써 신용등급(출력)이 알려지지 않음

목 적
학습자료를 이용하여 분류함수를 구축한 후, 예측자료의 알려지지 않은 출력값(신용등급)을 예측

주 의
예측자료는 분석 당시에는 관측되지 않은, 즉 사용될 수 없는 자료

모형선택

- 학습에러와 예측에러

학 습 에 러

학습자료로부터
구한 에러

예 측 에 러

예측자료로부터
구한 에러

기계학습은 일반화에 관심을 둔다. 따라서, 학습에러보다는 예측에러에 더 많은 관심을 둔다. 즉, 기계학습의 목적은 예측에러를 최소화하는 모형의 구축에 있다.

모형선택

- **모형의 복잡도**
  - 모형복잡도 예
    - 입력변수를 많이 사용할수록 모형이 복잡해진다 :

      모형 $y = a + b_1 x_1 + b_2 x_2$가 모형 $y = a + b_1 x_1$보다 복잡하다.
    - 입력변수와 출력변수의 관계를 나타내는 식이 비선형적 일수록 모형이 복잡하다 :

      모형 $y = 1/(1 + \exp(a + bx))$가 모형 $y = a + bx$보다 복잡하다.

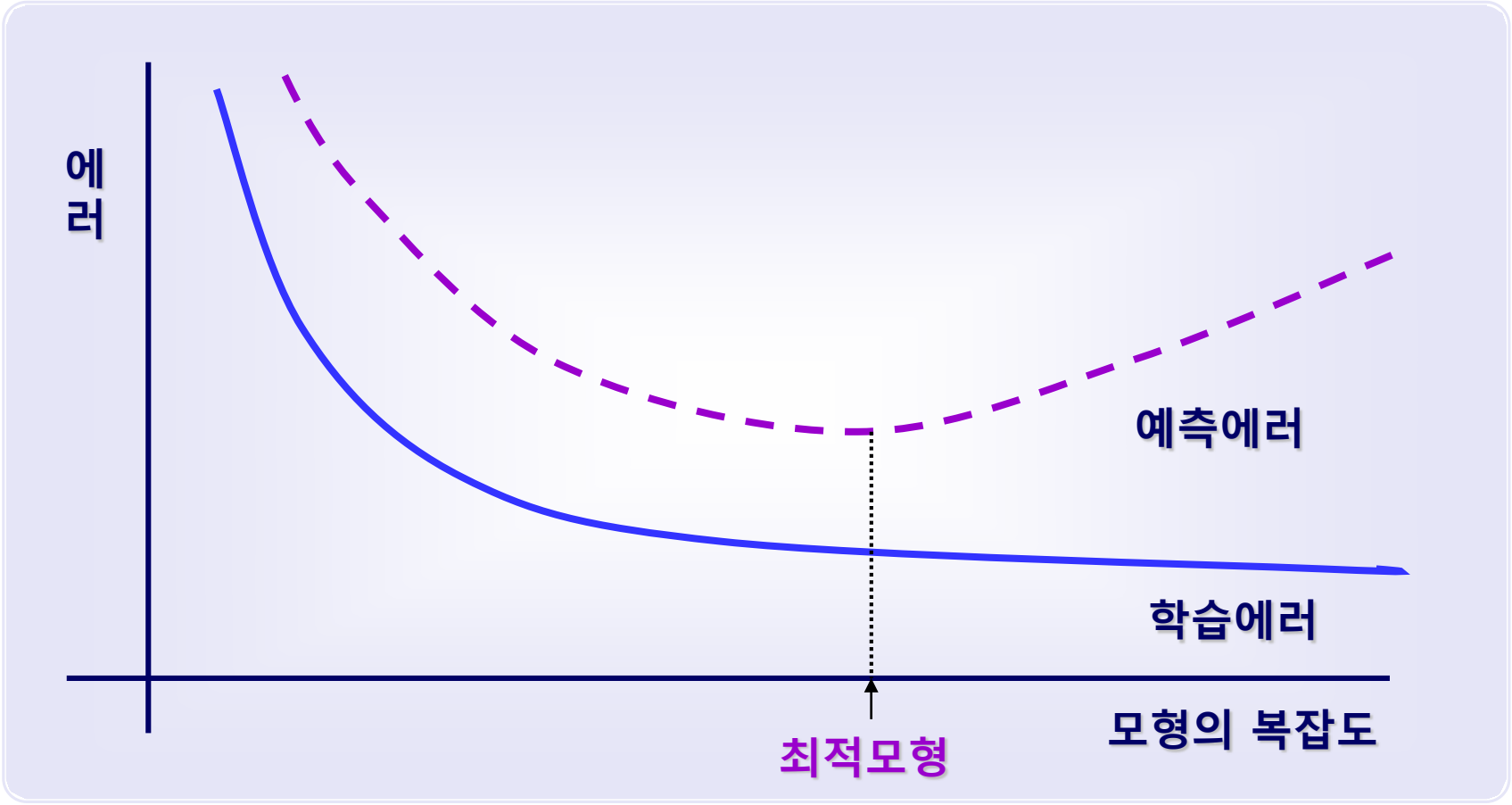
모형선택

- **과적합**
  - 정의

    매우 복잡한 모형을 사용하여 학습에러를 매우 작게 한 경우 예측에러가 매우 커질 수 있는데, 이러한 현상을 과적합이라한다.
  - 학습에러는 사용되어진 자료를 통하여 구할 수 있으나, 예측에러는 미래의 자료에 대한 에러로써 실제로 구할 수 없다.
  - 따라서 기계학습에서 학습에러를 너무 작게 하는 것이 항상 좋은 것은 아니다.

모형선택

- 모형의 복잡도와 과적합



모형선택

- **다변량 모형선택 방법의 문제점**
  - 각 입력변수와 출력변수와의 상관계수를 구하고, 이 상관계수의 절대값이 큰 순서대로 변수를 선택한다.
  - 적어도 다음의 두 가지 문제점 때문에 이러한 단순한 방법은 올바른 변수를 선택하지 못한다.
    - 다중공선성
    - Simpson's paradox

모형선택

- **다중공선성 예제**
  - 예제: 월소득(X1)과 가구원수(X2)가 월저축액(Y)에 미치는 영향 단변량 회귀분석 결과

	비표준화계수		유의확률
	B	표준오차	
월소득	**.205**	**.035**	**.000**

	비표준화계수		유의확률
	B	표준오차	
가구인원수	**1.625**	**1.182**	**.206**

모형선택

- **다중공선성 예제**
  - 예제: 월소득(X1)과 가구원수(X2)가 월저축액(Y)에 미치는 영향 다변량 회귀분석 결과

	비표준화계수		유의확률
	B	표준오차	
월소득	**.301**	**.029**	**.000**
가구인원수	**-2.091**	**.470**	**.003**

모형선택

- **Simpson's paradox**

Berkeley Admission Data

	합격	불합격	지원자(계)
남자	1400(52%)	1291(48%)	2691(100%)
여자	772(42%)	1063(58%)	1835(100%)
전체	2172(48%)	2354(52%)	4526(100%)

✓ 합격, 불합격 변수에 관심을 두고 전체 결과만 놓고 보면, 남자의 합격률이 여자의 합격률보다 높다.

⇒ 성차별 주장이 제기.

모형선택

- **Simpson's paradox**

	남		여	
분야	지원자	합격률	지원자	합격률
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	60%	341	70%

- **분야별로 보면 남자의 합격률보다 오히려 여자의 합격률이 더 높다.**

  성차별 주장은 잘못되었다.

모형선택

- **Simpson's paradox**
  - 1951년 미국 통계학자 E.H.Simpson이 내세운 이론으로, 분할표 분석에 있어서 전체 분석결과와 세부 분석의 결과가 모순되는 현상 (동일하지 않은 가중치 적용으로).
  - 어느 한 변수에 관심을 둘 때, 이에 영향을 줄 수 있는 제어변수를 모두 고려해야 할 필요가 있음을 강조.
  - 전체분석에 모든 것을 의존하는 것이 잘못된 자료해석을 초래할 수 있음을 제시.

모형선택

- **모형을 만드는 방법**
  - 모든 가능한 회귀 모형(All possible method)
  - 전진 선택법(Forward selection)
  - 후진 소거법(Backward elimination)
  - 단계적 방법(Stepwise method)

모형선택

- **AIC (Akaike Information Criteria)**
  - 모형의 예측성능을 측정하는 통계량
  - AIC :

$$\log(\text{잔차제곱합}) + 2k$$

    - k : 선택된 변수의 개수
    - 주어진 모형 중에 AIC를 최소로 하는 모형을 선택한다.
- **BIC (Bayesian Information Criteria)**

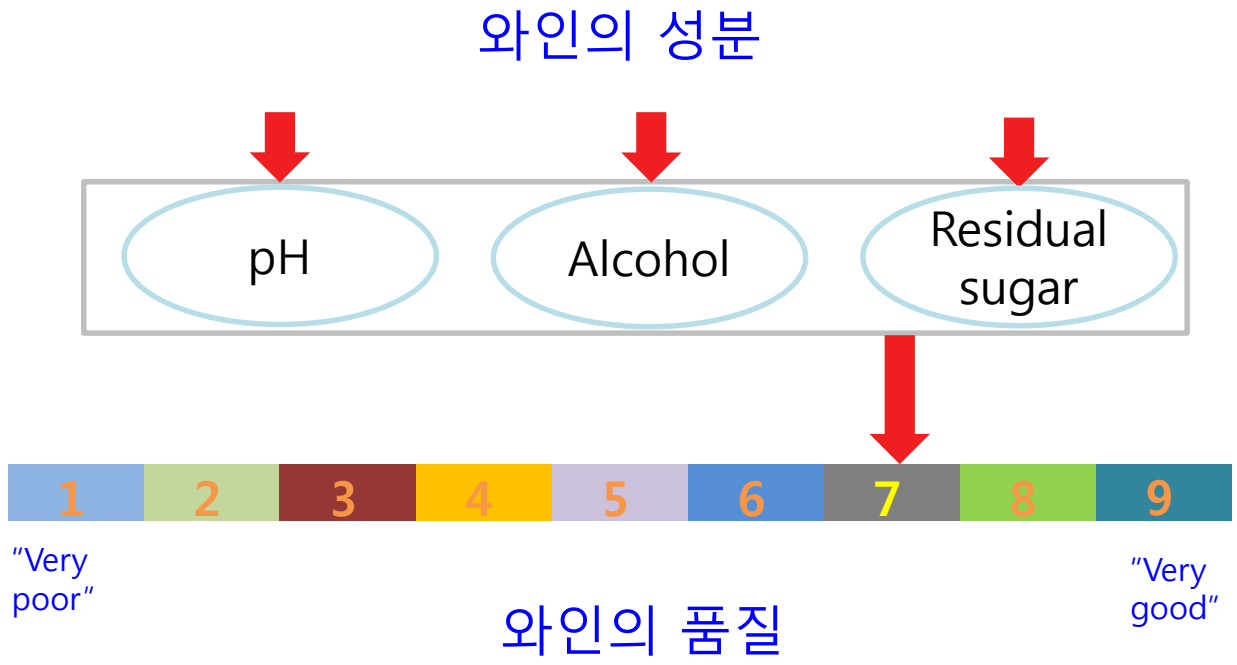
$$\log(\text{잔차제곱합}) + k\log n$$

모형선택

- **예제**
  - Red wine data
    - 1599개 red wine대상
    - 물리 화학적 계측법으로 얻어진 값
      - 산도, 알콜량, 당도 등등 11개 항목
    - 관능검사로 얻어진 값
      - 사람의 감각으로 종합적인 평가
      - Quality가 여기에 해당
      - Quality : 0~10

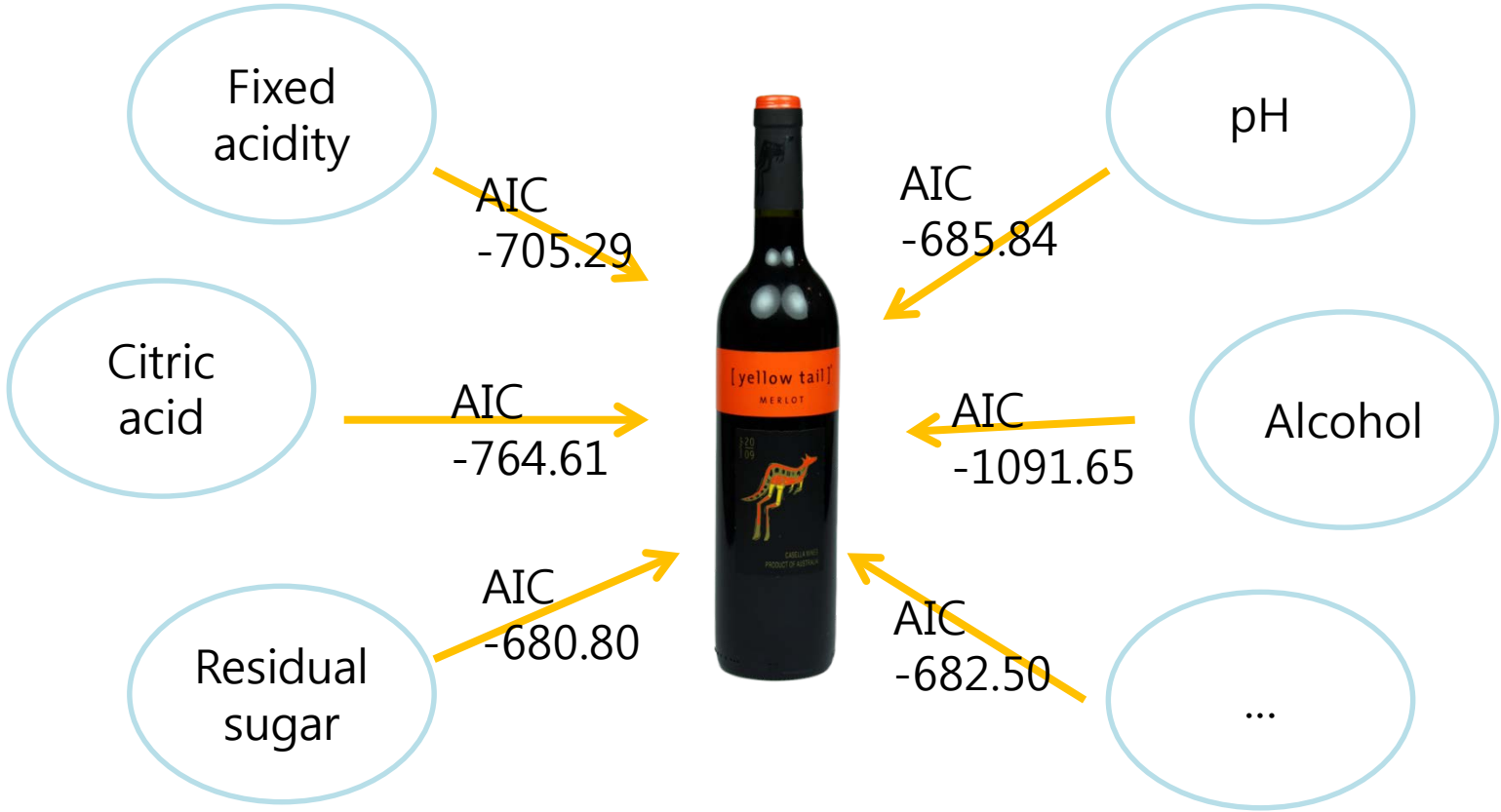
모형선택

- **예제**
  - Wine의 quality를 물리화학적으로 계측한 변수로 모델링.
  - AIC를 이용한 전진선택법으로 변수를 선택해보자.



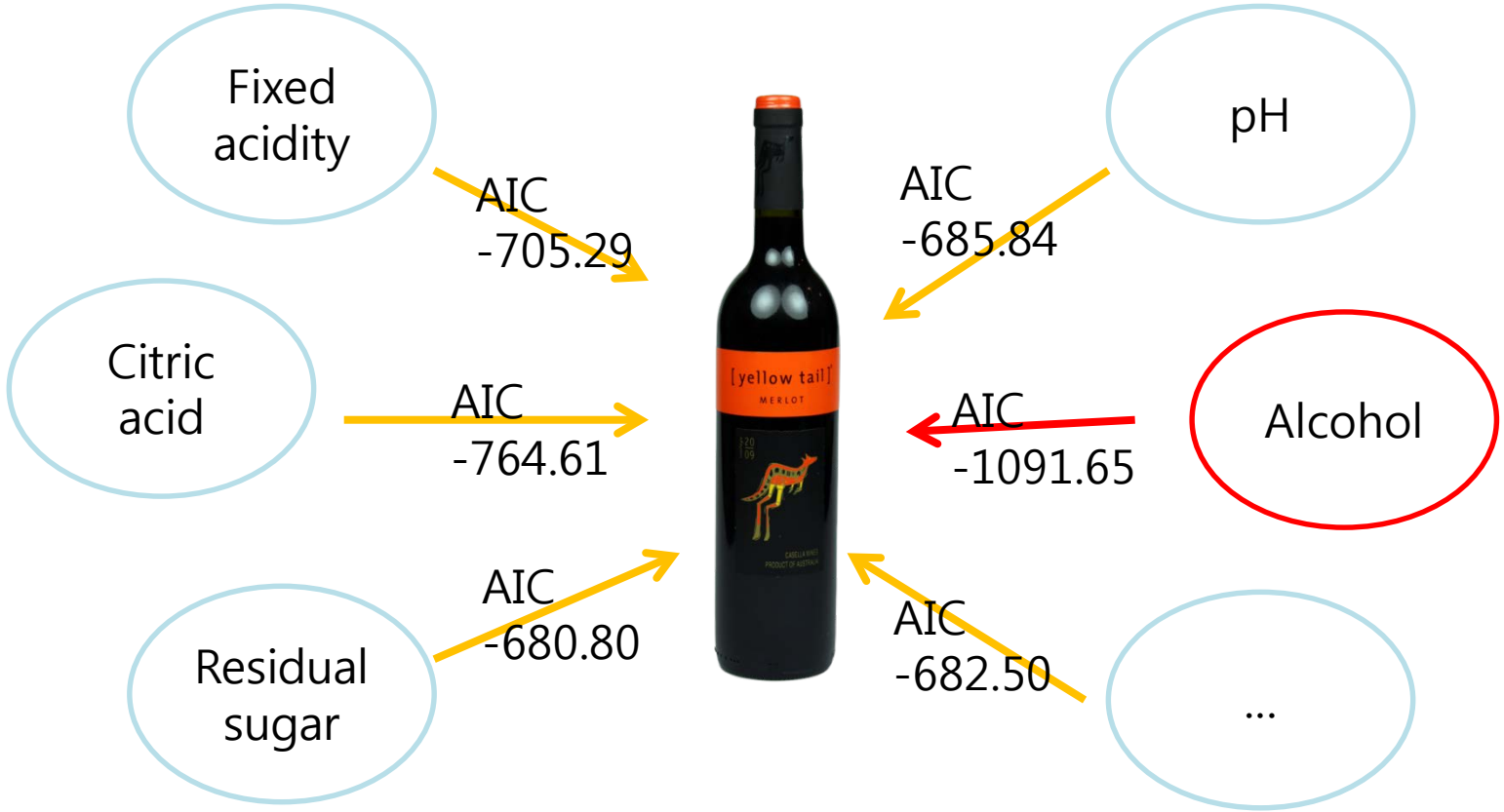
모형선택

- **예제- 1번째 변수선택**



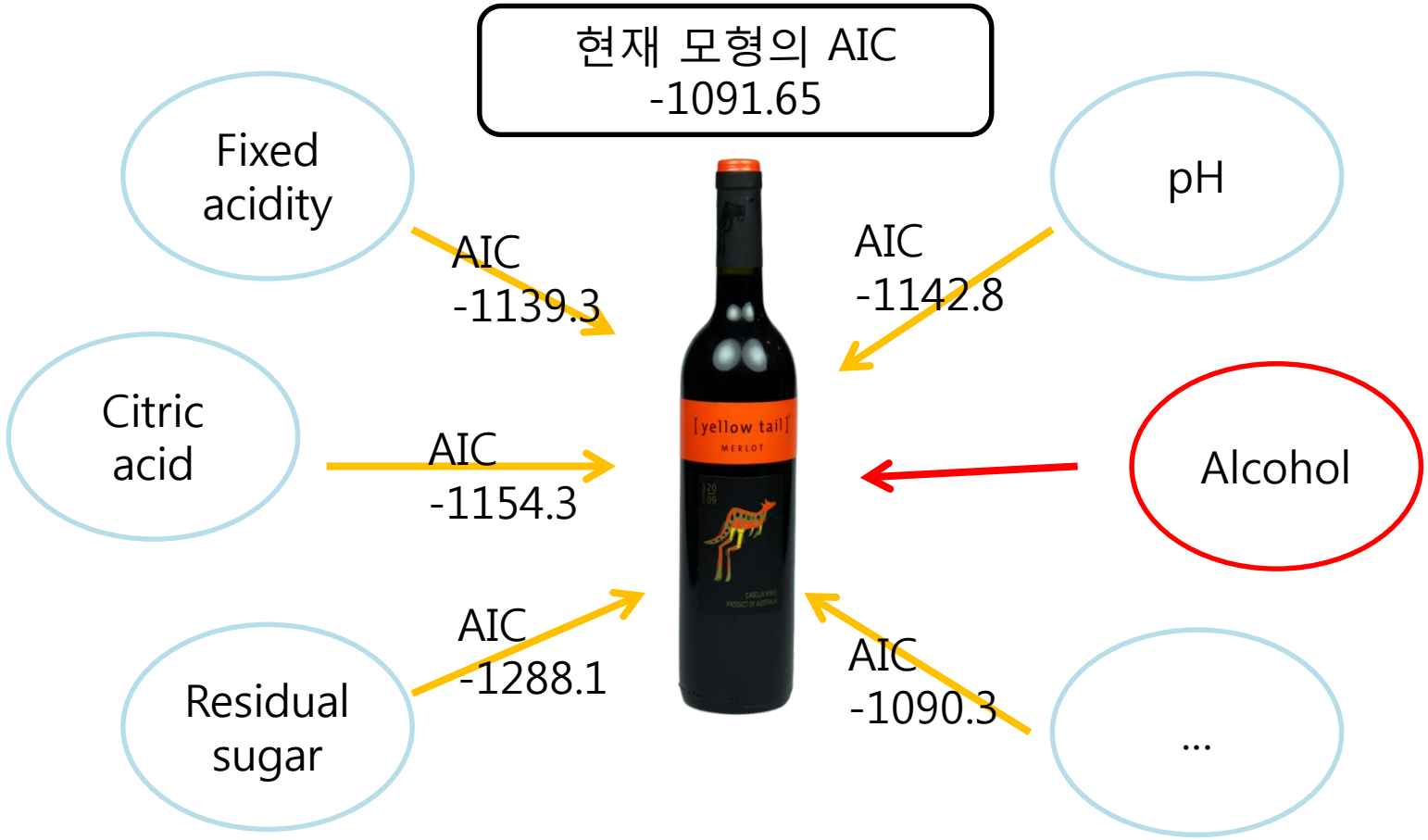
모형선택

- **예제- 1번째 변수선택**



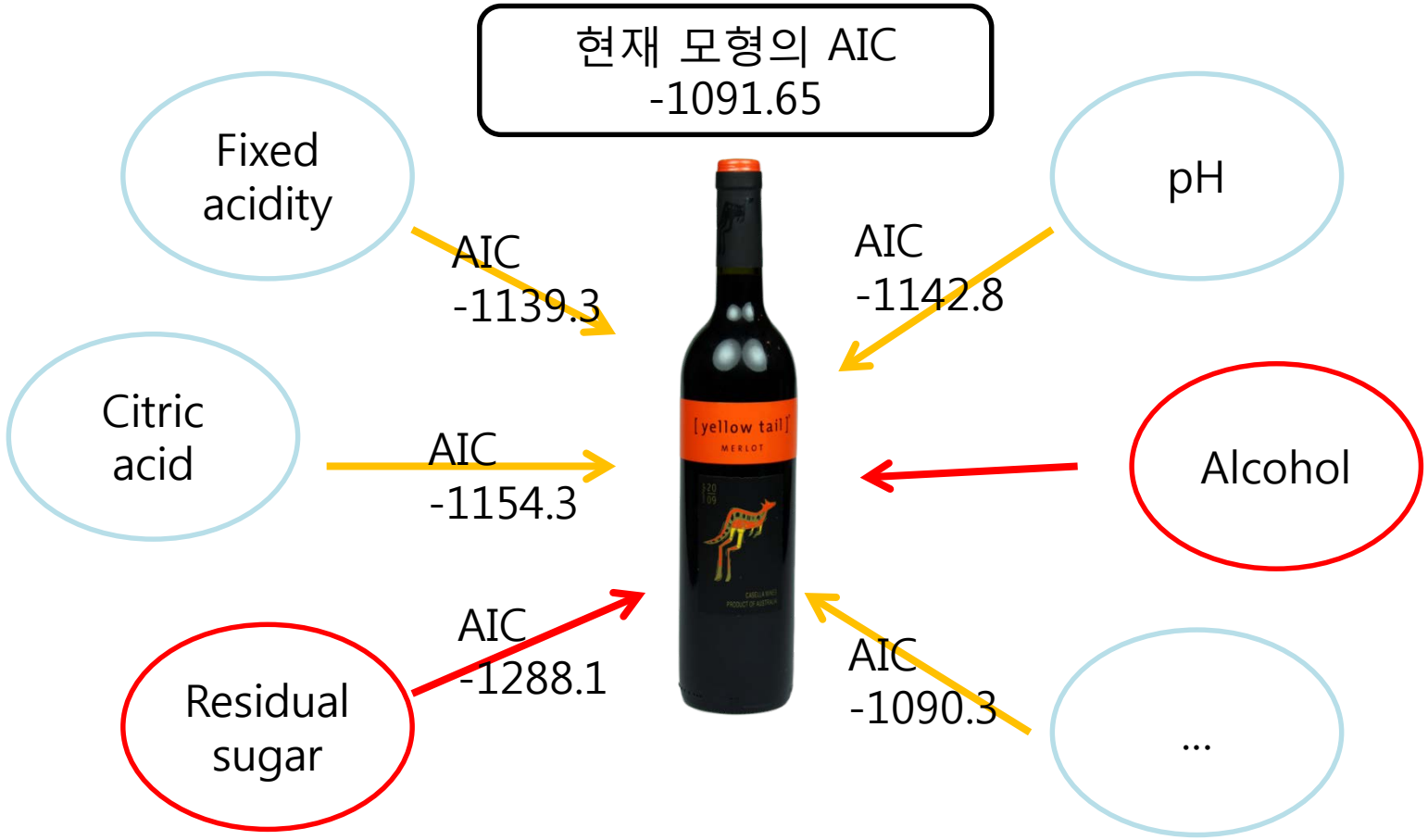
모형선택

- **예제- 2번째 변수선택**



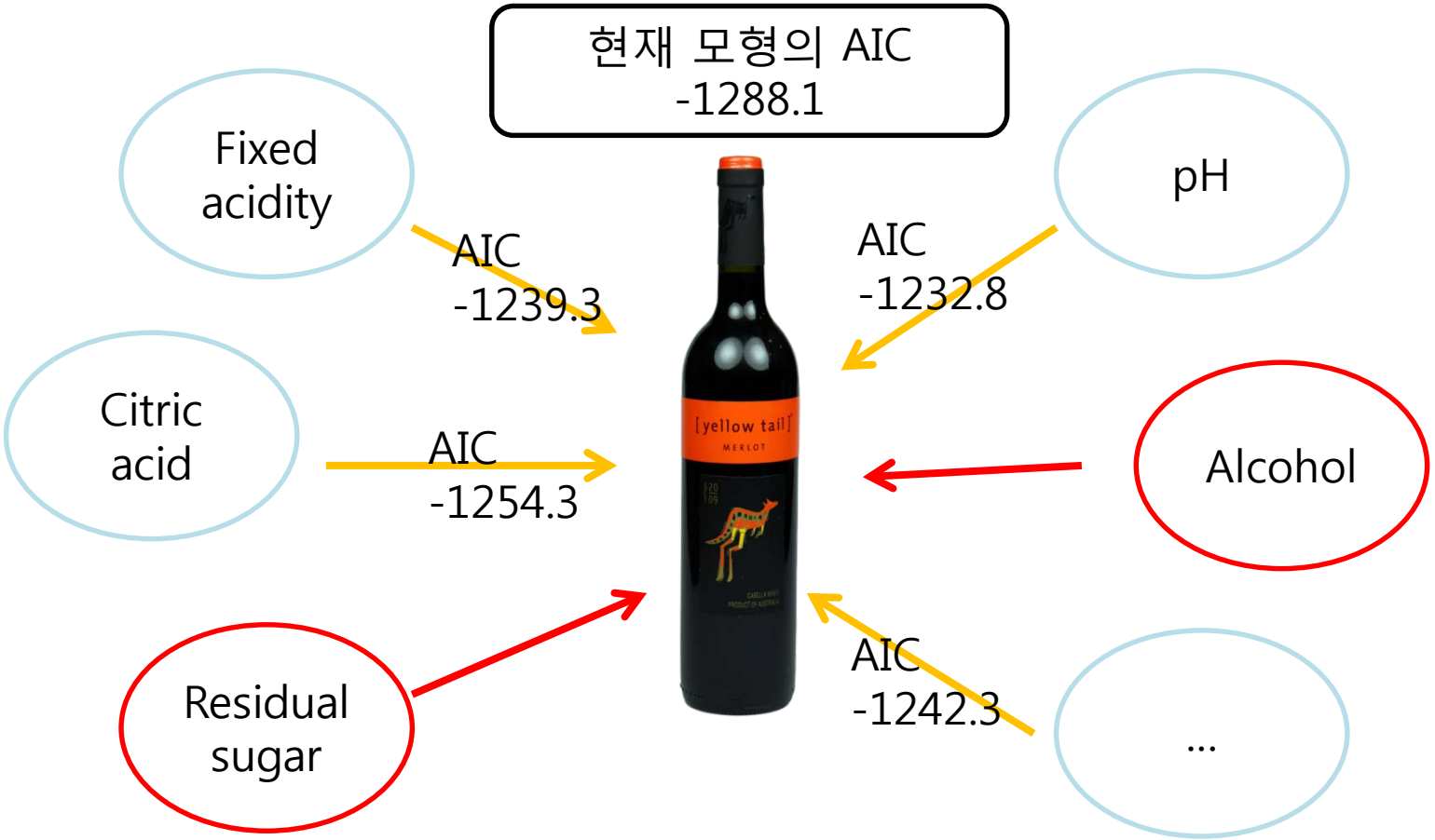
모형선택

- **예제- 2번째 변수선택**



모형선택

- **예제- 3번째 변수선택**



모형선택

- **예제- 최종모형**

현재 모형의 AIC
-1288.1

Alcohol

Residual sugar

모형선택

- **전진선택법의 장단점**
  - 방법의 이해가 쉽다.
  - 변수의 개수가 많은 경우에도 사용할 수 있다.
  - 최적의 모형을 선택하지 못할 수도 있다.
  - 변수 값의 작은 변동에도 결과가 크게 달라진다.
  - 한 변수가 선택된 경우 상관관계가 큰 다른 변수는 선택되지 못한다.
  - “Usually overly greedy”

4. 모형평가

모형의 평가

- 하나의 자료를 분석 할 때, 여러 가지 통계모형들을 상정하여 분석하는 것이 바람직하다.
- 그리고, 이 여러 가지 모형 중 자료를 최적으로 설명하는 모형을 선택한다.
- 최적의 모형을 선택하기 위하여는 여러 모형을 비교/평가 해야 하고, 특정한 모형이 다른 모형에 비하여 우수하다는 사실을 입증해야 한다.
- 모형을 평가하는 방법
  - 예측력: 얼마나 예측을 잘 하는가?
  - 해석력: 입력변수와 출력변수와의 관계를 잘 설명하는가?
  - 효율성: 얼마나 적은 수의 입력변수로 모형을 구축했는가?
  - 안정성: 모집단 내 다른 자료에 적용하는 경우 같은 결과를 주는가?

모형의 평가 (계속)

- 이 중, 가장 중요한 고려 사항은 예측력이다. 아무리 안정적이고 효율적이며 해석이 쉬워도, 실제 문제에 적용했을 경우 빗나간 결과만을 양산하는 경우, 아무 의미가 없다.
- 따라서, 모형의 평가란, 예측(prediction)을 위해 만든 모형이 임의의 모형(random model)보다 예측력이 우수한지, 고려된 다른 모형들 중 어느 모형이 가장 우수한 예측력을 보유하고 있는지를 비교, 분석하는 과정이다.
- 모형을 선택하는 기준
  - 회귀모형: Mallow's Cp, Adjusted R^2, AIC,BIC
  - 분류모형: 예측자료 오분류율, AIC, BIC, ROC, Lift
- 분류모형에서의 모형선택방법을 주로 다룬다.
- 특별히 case-control sampling자료에 대한 모형평가 방법을 배운다.

모형의 평가

- **예제 자료**
  - 데이터세트 이름: HMEQ
  - 데이터 설명: 한 은행의 대출부서가 담보대출 심사과정에서 얻은 대출자의 신상에 관련된 변수와 대출과 관련된 여러 변수들로 구성
  - 분석 목적: 대출자의 신상자료로부터, 이 대출자의 신용을 평가 한 후 이를 토대로 대출 승인을 결정
  - 변수 설명
    - 출력변수: 대출상환여부 (bad, 0 = 상환,1 = 미상환)
    - 입력변수: 총대출액, 저당액, 재산액, 대출사유 (빚 정리, 주택개량), 직업, 직장 근무년수, 대출금 대 수입의 비율, 신용거래 중 불량사유 보고회수, 체납회수, 최장대출기간, 최근 신용거래 요청회수, 신용거래회수

모형의 평가

- **사후확률의 계산**
  - 로지스틱 모형을 이용하여 출력변수의 사후확률 $P(Y = 1|x_1, \ldots, x_p)$을 계산
  - 즉, 사후확률은 입력변수 $(x_1, \ldots, x_p)$가 주어졌을 때, 목표변수 Y가 1이 될 확률이다.
  - 출력변수는 상환인 경우에 0, 미상환인 경우에 1로 주어졌기 때문에, 사후확률은 주어진 입력변수에서 대출금을 상환하지 않을 확률이다.
  - 사전확률과 사후확률
    - 사전확률: 입력변수를 고려하지 않은 경우의 확률 $P(Y = 1)$ (이 경우 자료의 분포와 모집단의 분포가 같아야 함)
    - 사후확률: 입력변수를 고려한 확률 $P(Y = 1|x)$

모형의 평가

- 사후확률의 계산의 결과

〈표 4-2〉
사후확률

(A) 목표변수

ID	MORTDUE	LOAN	VALUE	JOB	YOJ	DEROG	BAD
1	70167	12900	90812	Office	25.0	0	0
2	56957	12900	73181	Other	29.0	1	0
3	61031	12900	71550	Other	28.0	0	0
4	15374	13000	59309	Other	6.0	2	0
5	72931	13000	96803	Other	6.0	0	0
6	39386	13000	53149	Other	4.0	0	0
7	53951	13000	70095	Other	28.0	0	0
8	59000	13000	83187	Other	.	0	0
9	94941	13000	120000	ProfExe	0.1	0	1
10	47000	13000	60000	Self	1.0	0	1
11	177000	13000	200000	Mgr	3.0	0	1
12	111000	13000	137813	Mgr	6.0	2	1
13	60719	13000	73677	Other	11.0	0	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

→

(B)

사후확률
0.039
0.433
0.115
0.531
0.039
0.039
0.115
0.302
0.716
0.415
0.741
0.964
0.039
.
.
.
.
.

분류 기준값

- **분류 기준값이란?**
  - 사후확률을 구한 후, 이를 이용하여 자료를 분류(목표변수의 범주를 결정)를 하기 위하여 정하는 값
  - 예: 사후확률이 0.8 이상이면 자료를 1그룹에 할당하고 사후확률이 0.8 미만이면 자료를 0 그룹에 할당한다,

분류 기준값

- **분류 기준값의 예**
  - 첫 번째 자료 (ID=1)
    - 사후확률은 0.039.
    - 즉, 이 고객이 대출금을 미상환할 확률은 3.9%, 따라서, 대출금을 상환할 확률은 96.1%이다.
    - 상환할 사후확률이 매우 높으므로, 이 고객을 “상환”범주로 분류한다.

분류 기준값

- **분류 기준값의 예 (계속)**
  - 두 번째 자료 (ID=2)
    - 사후확률은 0.433.
    - 즉, 이 고객이 대출금을 미상환할 확률은 43.3%, 따라서, 대출금을 상환할 확률은 56.7%이다.
    - 상환할 사후확률이 상환하지 않을 사후확률보다 크다. 따라서, 이 고객을 상환으로 분류한다.
    - 그러나, 상환과 미상환의 사후확률이 큰 차이를 보이지 않으므로 분류에 대한 확실한 판단을 내리기가 어렵다.

분류 기준값

- **분류 기준값의 예 (계속)**
  - 일반적으로 J개의 범주가 있는 경우 분류 기준값은 1/J. 예제의 경우 J=2, 따라서 50%가 분류 기준값. 하지만, 분류 기준값은 사전확률과 손실함수 등 여러 가지를 고려하여 결정.

분류 기준값

- 분류 기준값의 선정과 그 결과

			분류기준값	
OBS	BAD	범주 1에 대한 사후확률	0.5	0.25
1	0	0.039	0	0
2	0	0.433	0	1
3	0	0.115	0	0
4	0	0.531	1	1
5	0	0.039	0	0
6	0	0.039	0	0
7	0	0.115	0	0
8	0	0.302	0	1
9	1	0.716	1	1
10	1	0.415	0	1
11	1	0.741	1	1
12	1	0.964	1	1
13	0	0.039	0	0

오분류표

- **오분류표**
  - 목표변수의 실제 범주와 모형에 의해 예측된 범주 사이의 관계를 나타내는 표

오분류표

원래 목표변수	예측된 변수의 범주 1	2	3	4	
1	9	4	0	1	14
2	6	21	2	1	30
3	15	3	71	7	96
4	0	2	0	8	10
	30	30	73	17	**150**

오분류표

- 오분류율 = 1-정분류율 = 1-(9+21+71+8)/150 = 41/150
- 대부분의 오분류가 목표변수 범주 3을 잘못 분류에 기인

오분류표

		예측된 변수의 범주				
		1	2	3	4	
원래 목표변수	1	9	4	0	1	14
	2	6	21	2	1	30
	3	15	3	71	7	96
	4	0	2	0	8	10
		30	30	73	17	**150**

오분류표

- **오분류율에 대한 다양한 추정치**
  - 범주가 2개인 경우의 오분류표의 구성

예측된 변수

원래 목표변수		0	1	
	0	실제 0 예측 0	실제 0 예측 1	실제 0
	1	실제 1 예측 0	실제 1 예측 1	실제 1
		예측 0	예측 1	

민감도와 특이도

- **오분류률은 Case-Control Sampling에서는 모집단의 오분률이 아님.**
- **Case-Control Sampling에서 다양한 모형성능 측정치**
  - 민감도(sensitivity)=(실제1,예측1)의 빈도/실제 1의 빈도
  - 특이도(specificity)=(실제0,예측0)의 빈도/실제 0의 빈도

(*) 정분류율=(실제0,예측0)빈도+(실제1,예측1)의 빈도/전체빈도

(*) 오분류율=(실제0,예측1)빈도+(실제1,예측0)의 빈도/전체빈도

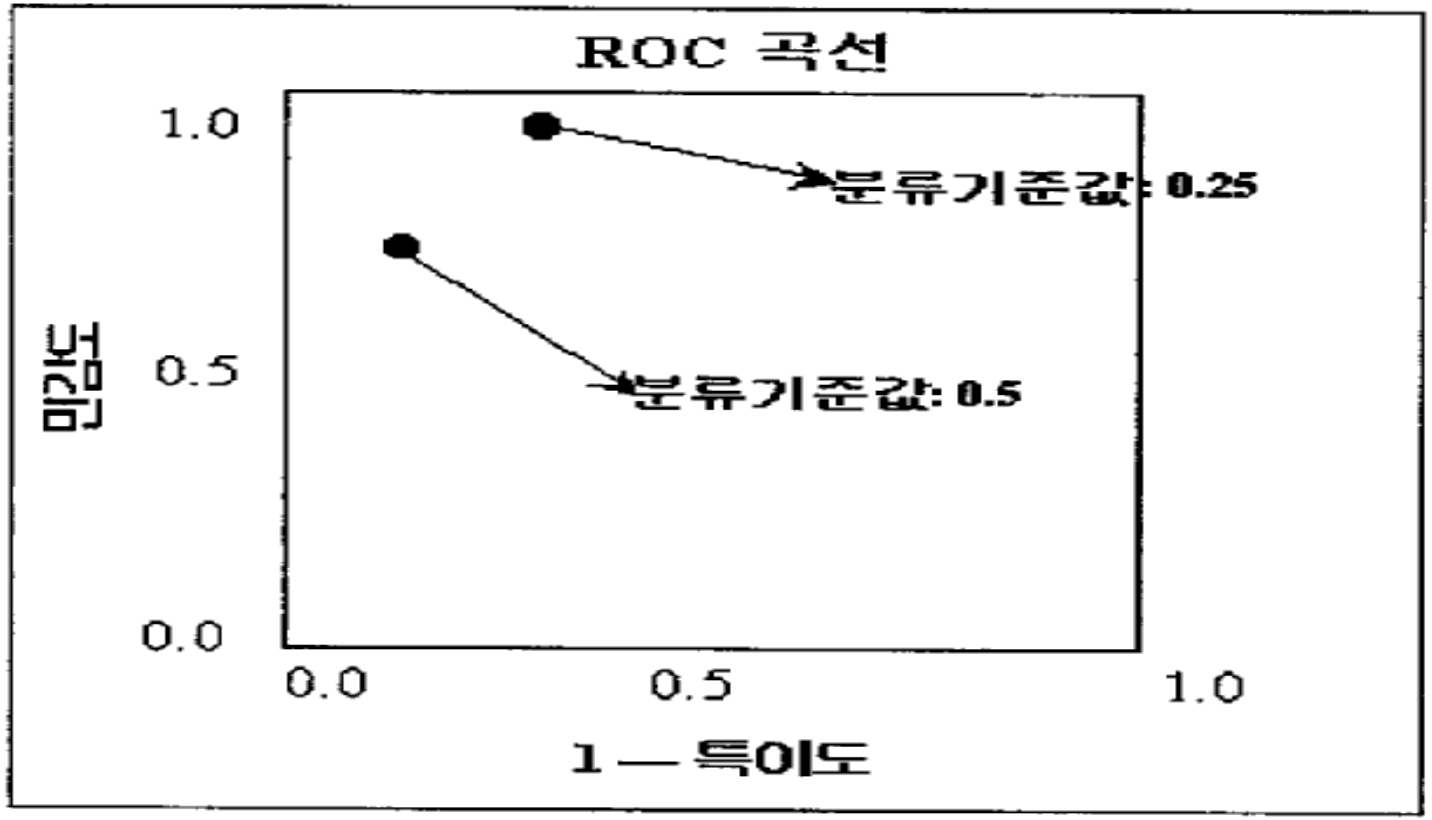
- 민감도는 범주 1에서의 정분류율이고, 특이도는 범주 0에서의 정분류율이다.
- 민감도와 특이도는 case-control sampling 자료에 대해서도 대응되는 모집단의 값을 잘 추정한다.

ROC 곡선

- **ROC 곡선**
  - 구축한 모형의 성능을 민감도와 특이도에 의해 판단.
  - 신호감지이론(Signal Decision Theory)으로부터 출발: 신호와 잡음의 분리를 목적으로 함.
  - 분류 기준값이 바뀌면서 특이도와 민감도의 값이 바뀐다.
  - 여러 개의 분류 기준값에서 구하여진 민감도 특이도 값의 쌍들을 x 축에는 1-특이도, y축에는 민감도를 지정하고 그린 곡선.
  - 민감도와 특이도는 반비례하고, 따라서, ROC곡선은 증가한다.

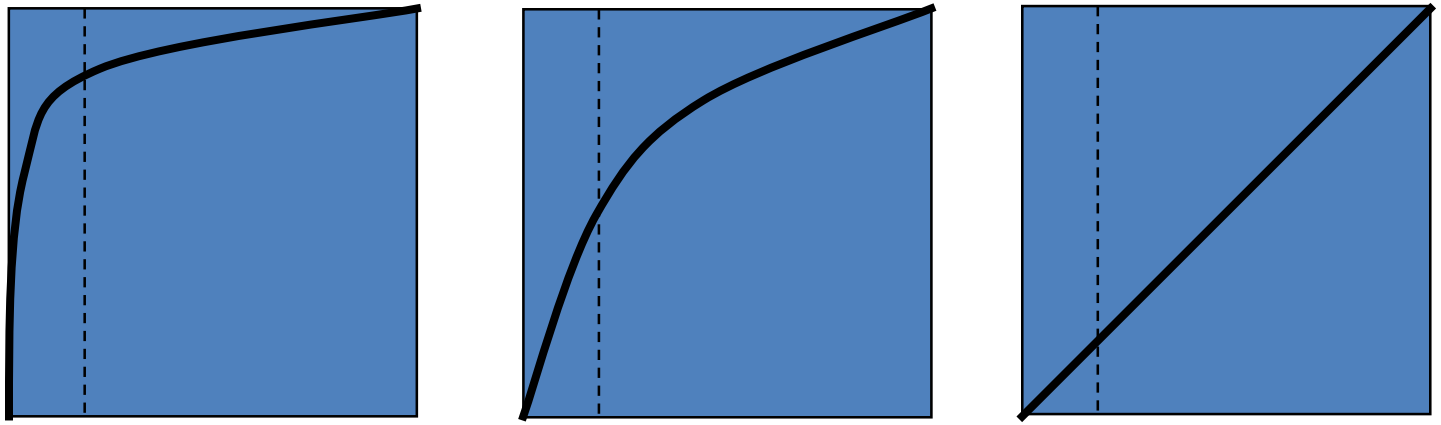
ROC 곡선

- ROC 곡선의 예



ROC 곡선

- **ROC 곡선의 비교**
  - 3가지 ROC 곡선
    - (a), (b), (c)의 세로점선은 동일한 '1-특이도'를 나타내고 있다.



(a) (b) (c)

ROC 곡선

- **ROC 곡선의 비교 (계속)**
  - 같은 1-특이도에서 그림 (a)의 민감도가 가장 높다.
  - 즉, 범주 0의 오분류률(1-특이도)이 같은 경우에, 그림 (a)가 범주 1의 정분류율 (민감도가) 가장 크다.
  - 그림 (c)는 모형구축의 효과가 전혀 없다. 즉, 모든 분류 기준값에서 민감도와 1-특이도가 같다. 범주0과 범주의 1의 자료의 수가 같을 때, 민감도와 1-특이도가 같다는 것은 오분류표에서 정분류 자료의 개수(대각원소의 개수)가 분류 기준값에 상관없이 항상 일정하다는 것을 의미한다.
  - ROC곡선의 면적은 AUC (Area Under the Curve)라고 하고, 이 값이 큰 모형이 좋은 모형으로 판단한다.

리프트 그래프

- **리프트 그래프**
  - 적합된 모형을 통해 각 개체에 대한 사후확률의 순서만을 이용하여 모형을 평가하는 방법.
  - 리프트 그래프 생성과정.
    1. 적합된 모형을 통해 모든 개체에 대해 사후확률 계산
    2. 사후확률의 크기에 따라 데이터를 내림차순으로 정렬.
    3. 데이터를 균일하게 K개의 그룹으로 나눈 뒤, 각 그룹에서 목표범주의 빈도 계산
    4. 각 그룹에 대해 반응률 (목표범주 1의 비율), 리프트 (반응률 / 기준선 반응률), 목표변수 1이 각 그룹에 얼마나 분포하는지 나타내는 %Captured을 계산.

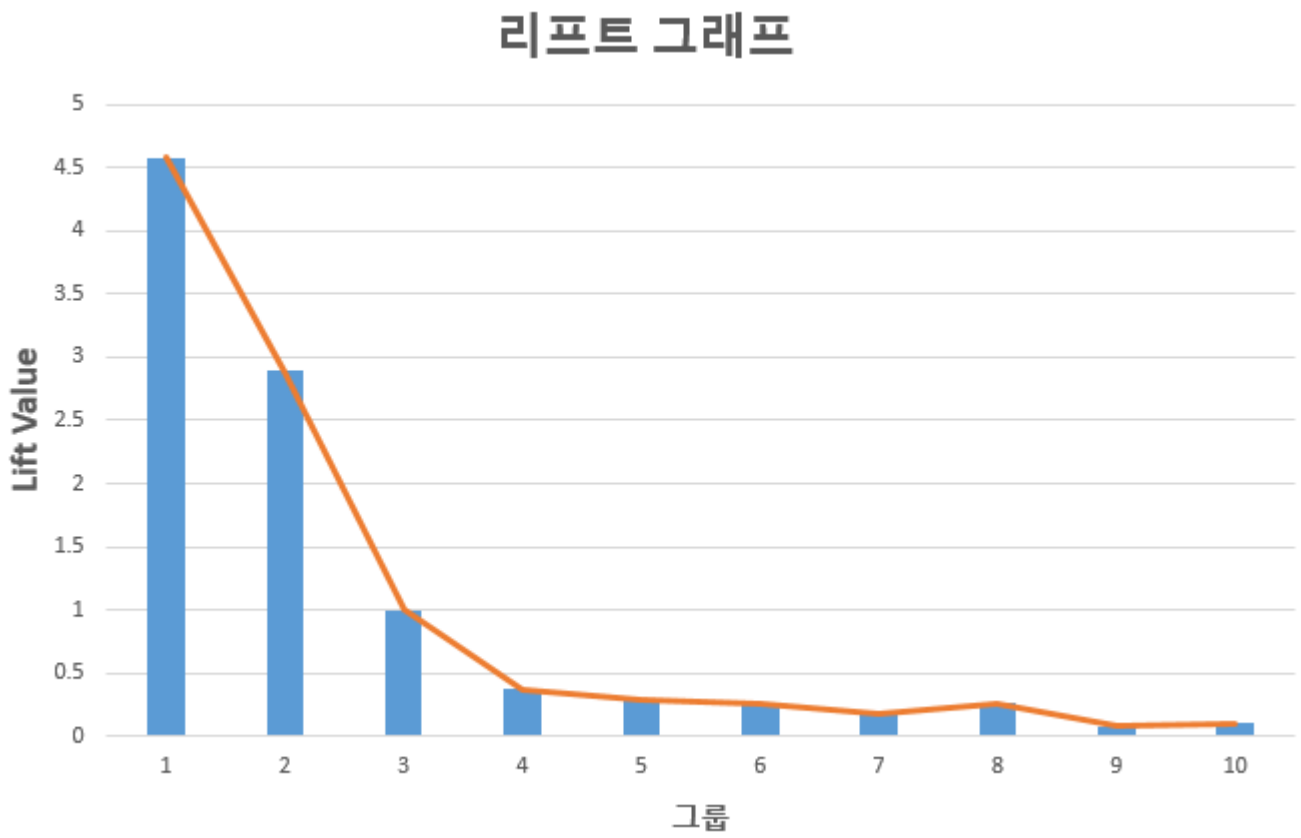
리프트 그래프

- **리프트 그래프의 예 (광고의 반응 여부)**
  - N = 2000이고 범주 1과 0의 개수가 각각 381개와 1619개라고 할 때,
    기준선 반응률은 381/2000 = 19.0%이다. 여기서 10개의 그룹으로 나누어서 구한 리프트 테이블은 아래와 같다.

	빈도			반응률	
그룹	합계	Y=1	Y=0	%Response	Lift
1	200	174	26	174/200=87.0	87.0/19=4.57
2	200	110	90	110/200=55.0	55.0/19=2.89
3	200	38	162	38/200=19.0	19.0/19=1
4	200	14	186	14/200=7.0	7.0/19=0.36
5	200	11	189	11/200=5.5	5.5/19=0.28
6	200	10	190	10/200=5.0	5.0/19=0.26
7	200	7	193	7/200=3.5	3.5/19=0.18
8	200	10	190	10/200=5.0	5.0/19=0.26
9	200	3	197	3/200=1.5	1.5/19=0.07
10	200	4	196	4/200=2.0	2.0/19=0.10
전체	Base Line %Response = 381/2000 = 19.0%				

리프트 그래프

- 리프트 그래프의 예 (계속)



리프트 그래프

- **리프트 그래프**
  - 각 등급은 사후 확률에 따라 매겨진 순위이므로, 좋은 예측 모형이라면 상위 등급에서는 더 높은 반응률 (또는 리프트), 하위 등급에서는 더 낮은 반응률을 보여야 한다.
  - 등급 (그룹)에 관계 없이 반응률 (리프트)에 별 차이가 없는 모형은 성능이 좋지 않음을 나타낸다.

5. 기계학습 알고리즘 1:
고차원 선형/로지스틱 모형

5-1. 고차원 모형 소개

고차원 선형 회귀모형

- 선형회귀모형에서 설명변수의 수 p가 데이터의 수 n보다 큰 경우
- 다중 공선성 (Multicollinearity)
  - 몇 개의 설명 변수들은 굉장히 높은 상관 관계를 가지고 있을 수 있다.
  - 선형 회귀 분석에서 최소 제곱 추정량이 유일하지 않을 수 있다.
- 과대적합 (overfitting)
  - 실제 모델보다 더 많은 설명 변수를 사용해 모델을 적합하게 될 수 있다.
  - 과대 적합은 좋지 않은 예측력을 유발한다.

고차원 선형 회귀모형

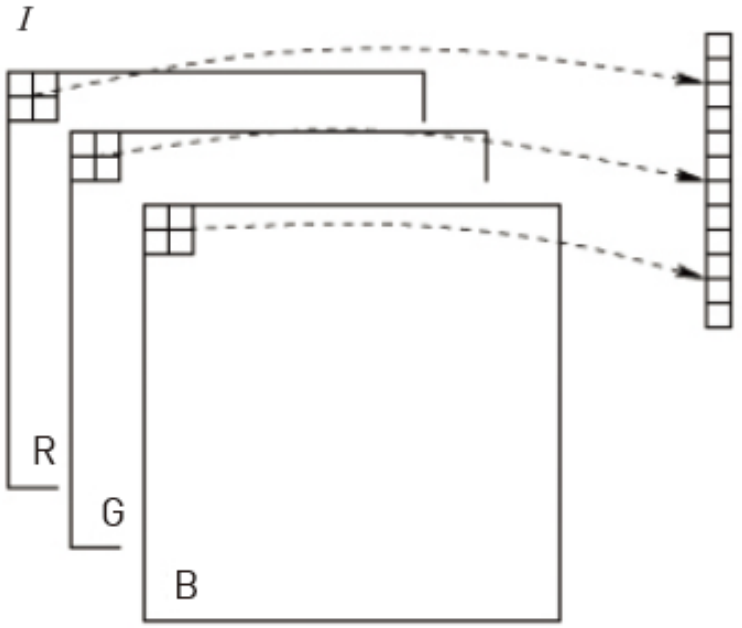
- **고차원 선형 회귀모형의 예**
  - 이미지 분석

original image

noisy image

고차원 선형 회귀모형

- 이미지 분석 (계속)
  - 잡음이 포함된 이미지의 모형화



If 3-channel 640480 image, $p = 3 \times 640 \times 480$

고차원 선형 회귀모형

– 이미지 분석 (계속)

• 잡음이 포함된 이미지의 모형화

1) 모형 :

$$y = \beta + \epsilon$$

- y는 관측된 픽셀들의 정보

- β는 잡음을 제거한 이미지의 픽셀정보

2) 이미지 분석의 목적

: y를 관측한 후 β에 대한 추정하는 것

고차원 선형 회귀모형

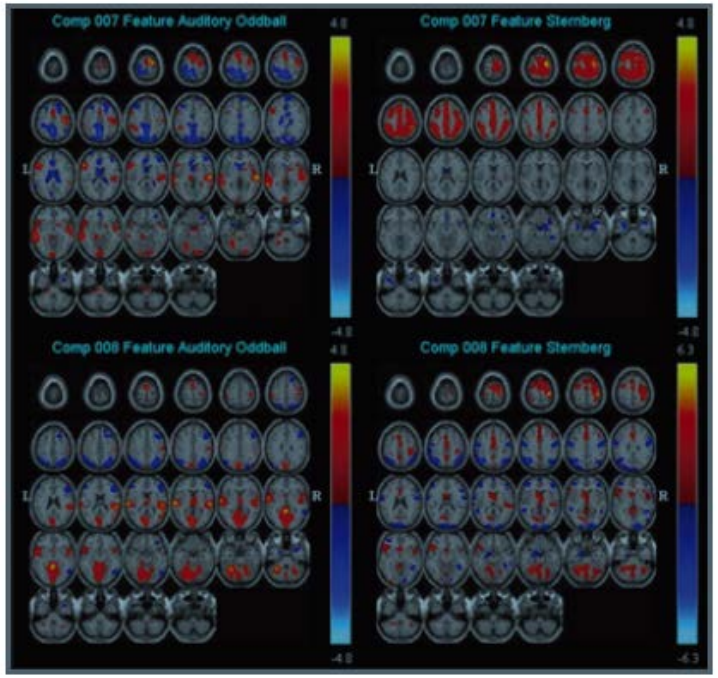
- 이미지 분석 (계속)
  - 벌점화 방법으로 이미지 잡음제거를 한 결과.

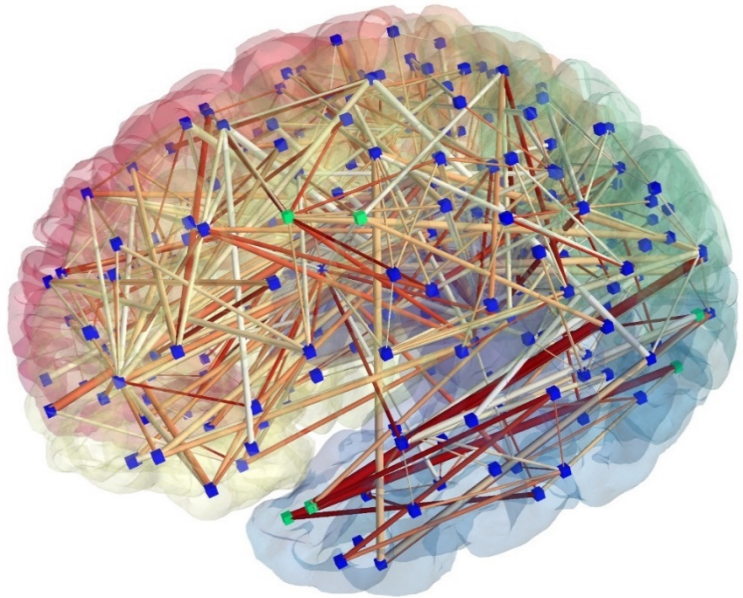
lasso denoising

fusion denoising

고차원 선형 회귀모형

- **고차원 선형 회귀모형의 예**
  - 브레인 네트워크 분석
    - 뇌의 활동을 그래프모형으로 표현한 것





고차원 선형 회귀모형

– 브레인 네트워크 분석

- 브레인 네트워크 그래프 모형

1. 노드 : 뇌의 각 지역

   - x_i : i번째 지역의 혈류량

   - $x = (x_1, \dots, x_p)$: p개 지역에서 혈류량

2. 두 개의 지역이 뇌의 행동에 연관되어 있느냐?

   - 편상관계수로 이해

3. 모형 (편상관성 추정)

   - $X_k = \sum_{i \neq k} \beta_{ki} X_i + \epsilon_k$

   - k번째와 i번째 지역의 편상관계수가 0 $\Leftrightarrow \beta_{ki} = 0$

   - 벌점화 방법을 이용하여 회귀계수 추정

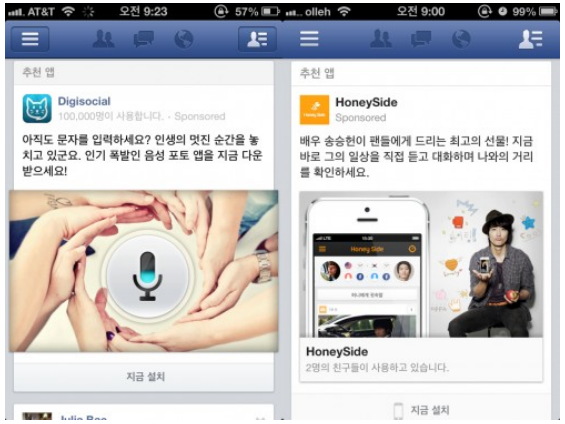
고차원 선형 회귀모형

- **고차원 선형 회귀모형의 예**
  - SNS을 이용한 광고추천

Text Mining

ID	nova	galaxy	heat	h'wood	film	role	diet	fur
A	10	5	3					
B	5	10						
C				10	8	7		
D				9	10	5		
E							10	10
F							9	10
G	5		7			9		
H		6	10	2	8			
I				7	5		1	3

고차원회귀



고차원 선형 회귀모형

- SNS을 이용한 광고추천
  - 출력변수는 특정 광고의 선호도
  - 입력변수는 사용한 단어의 빈도
  - 입력변수의 수는 사용 가능한 단어의 수 (거의 무한대)
  - 입력변수의 대부분이 0이다 (자료가 sparse하다!)
  - 예상: 광고선호도에 영향을 주는 단어는 아주 많지 않을 것이다

5-2. 능형회귀 (RIDGE REGRESSION)

벌점화 방법

- 벌점화 방법 소개
  - 주어진 벌점함수 $J_\lambda(\cdot)$에 대해서 다음의 벌점화 잔차제곱합을 최소로 하는 회귀계수를 추정하는 방법

$$\text{argmin}_\beta \sum (y_i - x_i'\beta)^2 + \sum J_\lambda(|\beta_k|)$$

여기서 λ는 양의 실수로 조율 모수이며, 벌점함수가 추정량에 미치는 영향을 조절

능형 회귀

- **능형회귀**
  - 선형회귀분석에서 최소 제곱 추정량은

$$\hat{\beta} = argmin_{\beta} \sum_{i=1}^{n} (y_i - \mathrm{x}_i^T \beta)^2 = (X^T X)^{-1} X^T y$$

  - 설명변수의 숫자가 표본의 숫자보다 크게 되면 (즉, $p > n$) $X^T X$의 역행렬이 존재하지 않는다.
    - 최소 제곱 추정량이 유일하지 않다.
    - 능형회귀!

능형 회귀

- **능형회귀의 정의**
  - 능형 추정량은 벌점함수 $J_\lambda(|\beta|) = \lambda\beta^2$을 사용하여 구한 벌점화 최소추정량

$$\text{argmin}_\beta\left(\sum(y_i - x_i'\beta)^2 + \lambda\sum\beta_k^2\right)$$
$$= (X^TX + \lambda I)^{-1}X^Ty$$

  - 능형 회귀에서의 벌점화 함수는 제곱 형태를 띠고 있으므로 l_2-벌점화 함수라고도 한다.

능형 회귀

- **축소 추정량**
  - 입력변수들끼리 독립인 경우,
    능형회귀 추정량은 **a*최소제곱추정량** 으로 주어지고, 여기서 a는 1보다 작다.
  - 즉, 능형회귀추정량을 최소제고추정량을 0으로 축소시키는 추정량이다.
  - 이러한 추정량을 축소추정량 (shrinkage estimator)라 부른다.

능형 회귀

- **축소 추정량**
  - 1차원 정규모집단에서 평균 벡터를 추정할 때 표본평균이 가장 좋은 추정량이다.
  - 2차원 정규모집단에서 평균 벡터를 추정할 때 표본평균이 가장 좋은 추정량이다.
  - 3차원 이상의 정규모집단에서 평균 벡터를 추정할 때 표본평균이 가장 좋은 추정량이 아니다.
  - 표본평균을 축소함으로써 더 좋은 추정량을 얻을 수 있다.

능형 회귀

Term	LS	Selection	Ridge
Intercept	2.480	2.495	2.467
x_1	0.680	0.740	0.389
x_2	0.305	0.367	0.238
x_3	-0.141		-0.029
x_4	0.210		0.159
x_5	0.305		0.217
x_6	-0.288		0.026
x_7	-0.021		0.042
x_8	0.267		0.123
Test Error	0.586	0.574	0.540

- 회귀 계수 추정량의 비교 및 시험 오차 비교.

능형 회귀

- **능형회귀의 한계**
  - 모든 회귀 계수들의 추정값이 0이 아니므로 설명 변수가 많을 경우 과대적합 및 해석력의 문제가 생길 수 있다.
  - 변수 선택의 기능까지 있는 벌점화 함수가 필요!

→ **LASSO (Least Absolute Shrinkage and Selection Operator)!**

5-3. 라쏘회귀 (LASSO REGRESSION)

라쏘 회귀

- 라쏘 벌점함수 : $J_\lambda(|\beta|) = \lambda|\beta|$
- 라쏘 추정량
  - 라쏘 벌점함수를 이용하여 구한 벌점화 최소제곱 추정량

$$\hat{\beta} = argmin_\beta \sum_{i=1}^{n}(y_i - \mathrm{x}_i^T\beta)^2 + \lambda \sum_{j=1}^{p}|\beta_j|$$

  - 라쏘 추정방법을 사용하면 몇 개의 회귀 계수들이 정확히 0으로 추정된다. (Model is sparse!)

라쏘 회귀

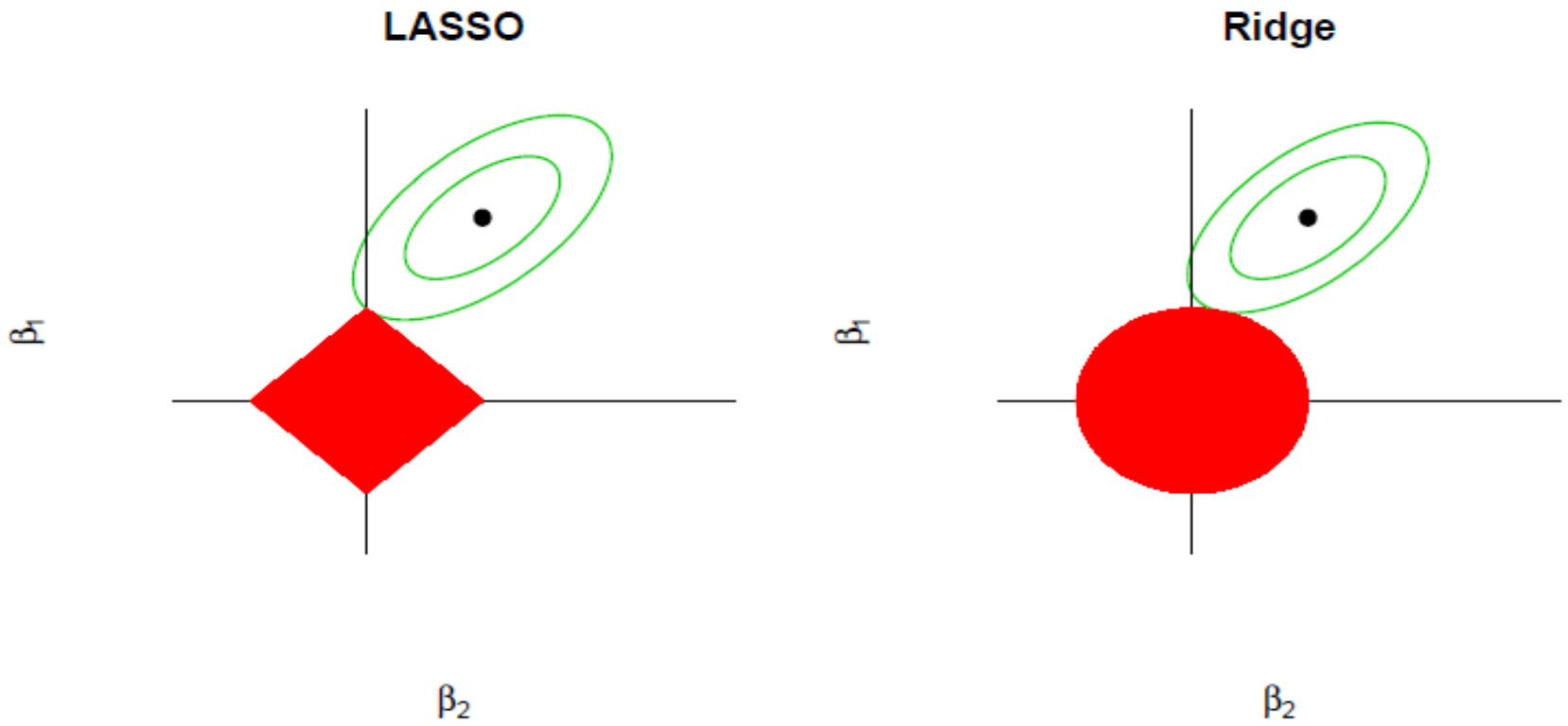
- **왜 라쏘는 sparse할까?**
  - Equivalent optimization problem
    - 능형회귀

      Minimize $\sum_{i=1}^{n}(y_i - \mathrm{x}_i^T\beta)^2$ subject to $\sum\beta_k^2$<s for some s>0.
    - 라쏘회귀

      Minimize $\sum_{i=1}^{n}(y_i - \mathrm{x}_i^T\beta)^2$ subject to $\sum_{j=1}^{p}|\beta_j|$<s for some s>0.
  - 여기서 s와 λ 사이에는 일대일 관계가 있음.

라쏘 회귀

- 왜 라쏘는 sparse할까?



라쏘 회귀

- 회귀 계수 추정량 비교

여러 가지 방법을 사용해 만든 추정 계수들

Term	LS	Ridge	Lasso
Intercept	2.480	2.467	2.477
x_1	0.680	0.389	0.545
x_2	0.305	0.238	0.237
x_3	-0.141	-0.029	
x_4	0.210	0.159	0.098
x_5	0.305	0.217	0.165
x_6	-0.288	0.026	
x_7	-0.021	0.042	
x_8	0.267	0.123	0.059
Test Error	0.586	0.540	0.491

→ 변수 선택 기능!

5-4. 조율 모수 선택

조율 모수 선택

- **예측자료 이용**
  - 데이터가 큰 경우에 사용
  - 데이터를 학습자료와 예측자료로 분할
  - 학습자료를 이용하여 다양한 조율 모수에서 모형 구축
  - 각 모형의 예측에러를 예측자료를 이용하여 추정
  - 예측에러가 가장 작은 조율 모수 선택

조율 모수 선택

- **교차 확인법**
  - 데이터를 몇 개의 조각을 나눈 후 예측 오차를 구하는 방법

1. 주어진 데이터 D를 k개의 조각 $D_1, \ldots, D_k$으로 나눔 (서로 배반)
2. 주어진 λ에 대해 k번 반복 수행
   ① 주어진 j에 대해서 $\cup_{i \neq j} D_i$만을 이용하여 능형 또는 라쏘 추정량 구함
   ② D_j데이터에서 예측오차 구함
3. k개의 오차의 평균을 λ에 대한 교차확인오차라 하고 이것이 가장 작은 λ선택

6. 기계학습 방법론 2: 앙상블

의사결정나무
배깅
랜덤포레스트
부스팅

7. 타겟 마케팅을 위한 자료분석

R 실습

Buytest 데이터 변수 설명

- 설명변수
  - ID, AGE, INCOME, SEX, MARRIED (1: 결혼, 0: 미혼), FICO (신용점수),
    OWNHOME (자가 주택 소유 여부, 1: 소유), LOC (거주지, A-H),
    BUY6, 12, 18 (최근 6, 12, 18개월 간의 구입 횟수), VALUE24 (지난 24개월 간의 구입 총액),
    ORGSRC (고객 분류), DISCBUY (할인 고객 여부, 1: 할인 고객),
    RETURN24 (지난 24개월 간 상품 반품 여부), COA6 (6개월 간의 주소변경 여부, 1: 주소변경)
- 반응변수
  - RESPOND (DM에 대한 반응 여부)

R 실습

- Buytest 데이터에서 결측치가 있는 개체를 제거한 뒤, 학습자료 (70%), 예측자료 (30%)으로 분할.

```
> set.seed(123)
> train_ind = sample(1:nrow(buydata), size = floor(nrow(buydata)*0.7),
+                    replace = F)
> train = as.data.frame(buydata[train_ind,])
> test = as.data.frame(buydata[-train_ind,])
> X_train = buydata[train_ind, -1]
> y_train = buydata[train_ind, 1]
> X_test = buydata[-train_ind, -1]
> y_test = buydata[-train_ind, 1]
>
> dim(X_train)
[1] 6454   26
> dim(X_test)
[1] 2767   26
```

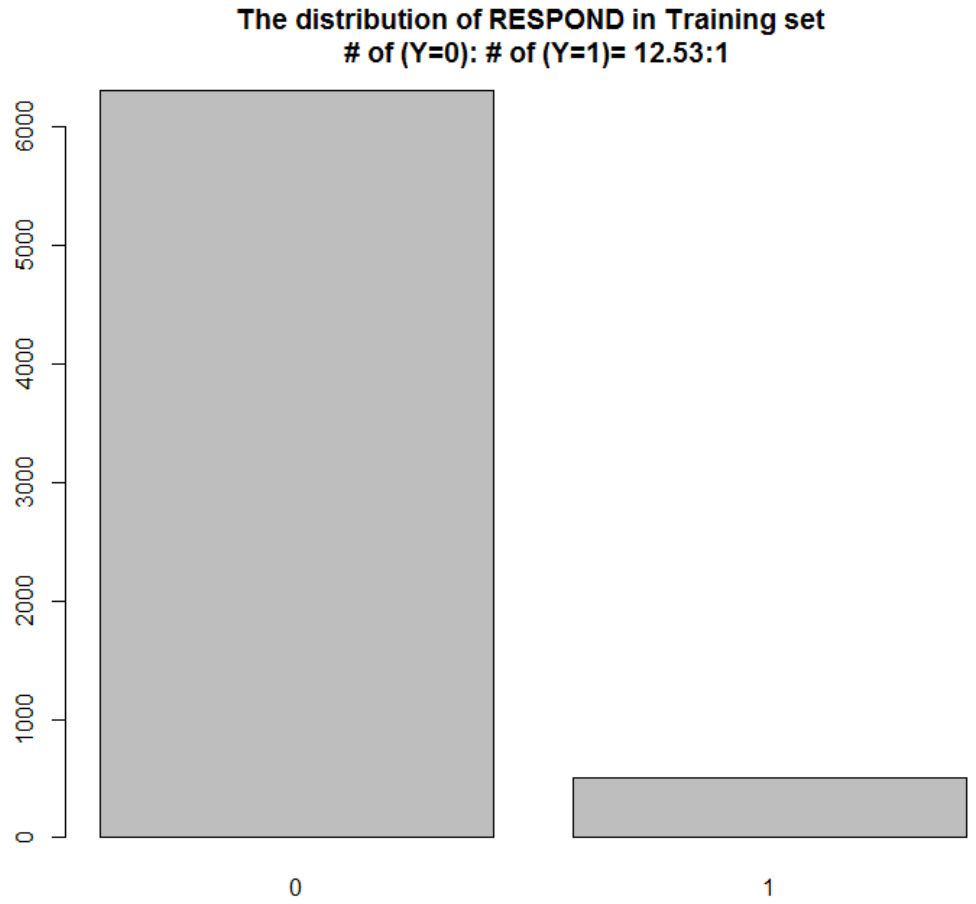
✓ 범주형 변수 (LOC, ORGSRC)에 대해 가변수 생성

✓ 여기서 C1~C7, PURCHTOT 변수는 DM에 의한 품목별, 그리고 총 구입 가격이므로 입력변수로 사용하지 않음.

R 실습

- 학습 자료 형태

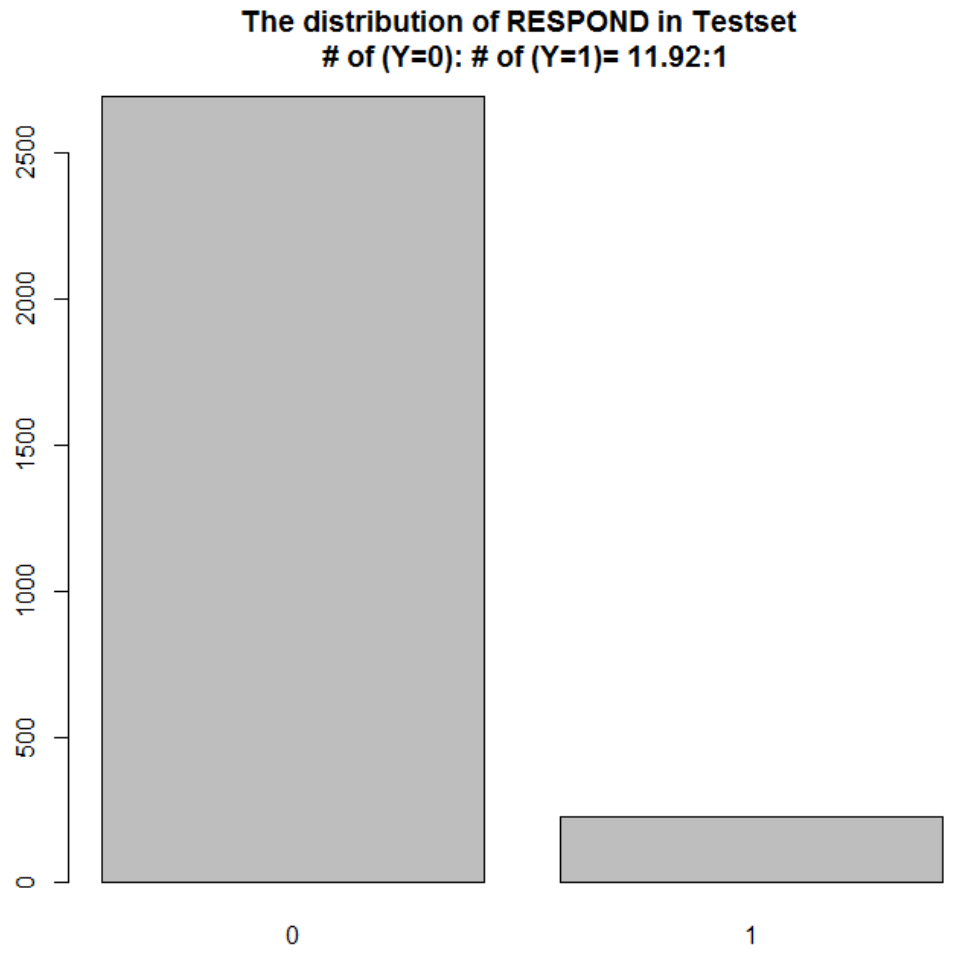
```
> head(X_train)
      AGE INCOME SEXM MARRIED FICO OWNHOME LOCB LOCC LOCD LOCE LOCF
2876   37     51    0       0  665       0    0    1    0    0    0
7887   37     59    1       0  640       1    0    0    0    0    1
4083   44     19    1       1  709       1    0    0    0    1    0
8829   38     54    0       0  637       1    0    0    0    0    0
9398   41     41    1       1  617       0    0    0    0    0    0
461    50     35    1       1  707       0    0    0    0    0    0
      LOCG LOCH BUY6 BUY12 BUY18 VALUE24 ORGSRCD ORGSRCI ORGSRCO
2876     0    0    0     0     1      538       0       0       0
7887     0    0    0     0     0      182       0       0       0
4083     0    0    1     1     1      246       0       0       0
8829     1    0    0     0     0      226       0       0       0
9398     0    1    0     0     2      456       1       0       0
461      0    0    0     0     0      190       0       0       1
      ORGSRCP ORGSRCR ORGSRCU DISCBUY RETURN24 COA6
2876        1       0       0       0        0    0
7887        0       0       1       0        0    0
4083        0       0       0       1        0    0
8829        1       0       0       0        0    0
9398        0       0       0       0        0    0
461         0       0       0       0        0    0
```



R 실습

- 예측 자료 형태

```
> head(X_test)
   AGE INCOME SEXM MARRIED FICO OWNHOME LOCB LOCC LOCD LOCE LOCF
2   53     72    1       1  751       0    0    0    0    0    0
3   53     70    0       1  725       0    0    0    0    0    0
4   45     56    0       0  684       0    0    0    0    0    0
13  48     57    0       0  698       0    0    0    0    0    0
14  67     33    1       0  713       0    0    0    0    0    0
15  44     17    1       1  751       0    0    0    0    0    0
   LOCG LOCH BUY6 BUY12 BUY18 VALUE24 ORGSRCD ORGSRCI ORGSRCO
2     0    0    0     0     0      83       0       0       0
3     0    0    1     1     1     265       1       0       0
4     0    0    0     0     1     448       0       0       1
13    0    0    0     0     0     226       0       0       1
14    0    0    0     0     0     145       0       0       0
15    0    0    1     1     1     494       0       0       0
   ORGSRCP ORGSRCR ORGSRCU DISCBUY RETURN24 COA6
2        0       1       0       0        0    0
3        0       0       0       0        0    0
4        0       0       0       1        0    0
13       0       0       0       0        1    0
14       0       0       1       0        0    0
15       0       0       0       0        0    0
```



R 실습

- DM에 대한 반응여부 (RESPOND)에 대해 로지스틱 회귀모형 적합 및 결과.

```
> logit_model = glm(RESPOND~., data = train, family = 'binomial')
> summary(logit_model)

Call:
glm(formula = RESPOND ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0853  -0.4205  -0.3489  -0.2848   2.8658

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.869e+00  1.196e+00   2.399 0.016431 *
AGE         -3.486e-02  5.641e-03  -6.180 6.40e-10 ***
INCOME      -1.372e-03  3.113e-03  -0.441 0.659448
SEXM         6.579e-03  9.885e-02   0.067 0.946940
MARRIED      5.971e-01  1.127e-01   5.300 1.16e-07 ***
FICO        -5.927e-03  1.649e-03  -3.595 0.000325 ***
OWNHOME     -3.764e-01  1.131e-01  -3.327 0.000879 ***
LOCB        -1.644e-01  2.213e-01  -0.743 0.457336
LOCC         2.494e-01  2.586e-01   0.965 0.334765
LOCD         3.593e-01  2.547e-01   1.411 0.158289
LOCE        -2.325e-01  2.205e-01  -1.054 0.291686
LOCF        -2.102e-01  2.215e-01  -0.949 0.342472
LOCG        -1.776e-01  2.524e-01  -0.704 0.481569
LOCH        -1.852e-01  2.431e-01  -0.762 0.445987
BUY6        -1.462e-01  1.986e-01  -0.736 0.461580
BUY12        4.601e-01  1.871e-01   2.459 0.013950 *
```

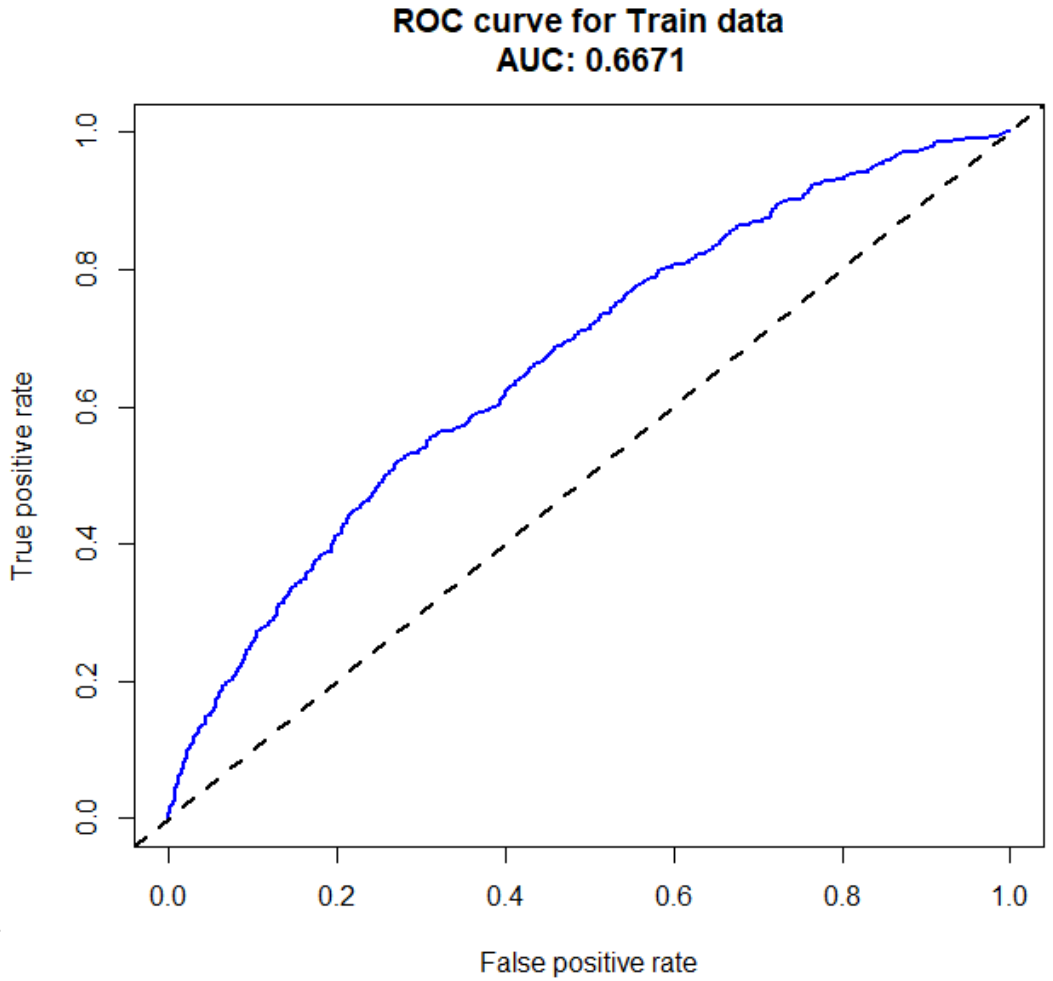
R 실습

- 절단값에 따른 학습자료의 오분류율, 민감도, 특이도.

```
> print(as.data.frame(cutoff_out))
   cutoff error rate sensitivity specificity
1    0.05     0.6244      0.8489      0.3373
2    0.10     0.2135      0.3851      0.8190
3    0.15     0.1147      0.1739      0.9429
4    0.20     0.0863      0.0787      0.9812
5    0.25     0.0800      0.0311      0.9920
6    0.30     0.0767      0.0186      0.9965
7    0.35     0.0756      0.0083      0.9985
8    0.40     0.0750      0.0041      0.9995
9    0.45     0.0747      0.0021      1.0000
10   0.50     0.0748      0.0000      1.0000
11   0.55     0.0748      0.0000      1.0000
12   0.60     0.0748      0.0000      1.0000
13   0.65     0.0748      0.0000      1.0000
14   0.70     0.0748      0.0000      1.0000
15   0.75     0.0748      0.0000      1.0000
16   0.80     0.0748      0.0000      1.0000
17   0.85     0.0748      0.0000      1.0000
18   0.90     0.0748      0.0000      1.0000
19   0.95     0.0748      0.0000      1.0000
```

R 실습

- 로지스틱 회귀모형 적합 및 결과: 학습자료에서의 ROC curve 및 AUC



R 실습

- 로지스틱 회귀모형 + 전진선택 AIC

```
null = glm(RESPOND~1, data = train, family = 'binomial')
full = glm(RESPOND~., data = train, family = 'binomial')
forward = step(null, scope = list(lower = null, upper = full),
                                  data = train, direction = "forward")
```

```
> summary(forward)

Call:
glm(formula = RESPOND ~ BUY18 + AGE + MARRIED + BUY12 + OWNHOME +
    FICO + LOCD + LOCC + ORGSRCD, family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0707  -0.4200  -0.3511  -0.2879   2.8534

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.604953   1.151763   2.262 0.023715 *
BUY18        0.788699   0.103752   7.602 2.92e-14 ***
AGE         -0.035309   0.005615  -6.288 3.22e-10 ***
MARRIED      0.597999   0.112279   5.326 1.00e-07 ***
BUY12       -0.538083   0.146958  -3.661 0.000251 ***
OWNHOME     -0.400939   0.109551  -3.660 0.000252 ***
FICO        -0.005879   0.001642  -3.581 0.000342 ***
LOCD         0.545529   0.175817   3.103 0.001917 **
LOCC         0.433527   0.180889   2.397 0.016546 *
ORGSRCD     -0.210047   0.129680  -1.620 0.105289
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

R 실습

- 로지스틱 회귀모형 + Ridge
  - 여기서 λ은 0보다 큰 조율모수.

```
> ridge.fit = glmnet(X_train, as.factor(y_train), alpha = 0,
+                    family="binomial" )
> ridge.fit$lambda[c(1, 10, 30, 50, 70, 100)]
[1] 26.175438761 11.330722582  1.762693481  0.274218020  0.042659443  0.002617544
> ridge.fit$beta[,c(1, 10, 30, 50, 70, 100)]
26 x 6 sparse Matrix of class "dgCMatrix"
                  s0            s9           s29           s49           s69           s99
AGE      -1.556600e-39 -1.352294e-04 -8.436225e-04 -0.0045947385 -0.0158590484 -0.0298988675
INCOME   -4.868359e-40 -4.212515e-05 -2.568025e-04 -0.0012517708 -0.0032791809 -0.0047845732
SEXM     -1.505248e-39 -1.283891e-04 -7.272097e-04 -0.0023800252 -0.0012267896  0.0005356394
MARRIED   1.347485e-38  1.173292e-03  7.397331e-03  0.0431806531  0.1824278470  0.4010388756
FICO     -2.504627e-40 -2.173753e-05 -1.347891e-04 -0.0007075373 -0.0021136085 -0.0033362463
OWNHOME  -3.683045e-38 -3.192917e-03 -1.968875e-02 -0.1016583307 -0.3091006492 -0.5038841946
LOCB     -1.319271e-38 -1.146848e-03 -7.169857e-03 -0.0392067059 -0.1360059278 -0.3429212597
LOCC      4.351218e-38  3.774385e-03  2.333365e-02  0.1194164657  0.3088154673  0.3137131718
LOCD      2.881456e-38  2.499459e-03  1.545332e-02  0.0787673228  0.1930149866  0.1215334481
LOCE     -1.382887e-38 -1.195488e-03 -7.267204e-03 -0.0348625307 -0.0946311593 -0.2504779251
LOCF     -7.235543e-39 -6.217229e-04 -3.649061e-03 -0.0145641308 -0.0255344641 -0.1242547872
LOCG      9.875142e-39  8.453843e-04  4.855941e-03  0.0167855795  0.0076121801 -0.0729549813
LOCH     -2.054345e-39 -1.873398e-04 -1.457174e-03 -0.0140773592 -0.0717584406 -0.1873351916
BUY6      2.795898e-38  2.382148e-03  1.323047e-02  0.0365711934 -0.0195239630 -0.0459882449
BUY12     2.912916e-38  2.487462e-03  1.404610e-02  0.0435644534 -0.0242258850 -0.3600843364
BUY18     4.622690e-38  3.989714e-03  2.404137e-02  0.1135871980  0.3504977982  0.7857864556
VALUE24   1.034113e-40  8.871536e-06  5.179606e-05  0.0001995164  0.0002029927 -0.0004360299
ORGSRCD  -8.747317e-39 -7.565892e-04 -4.606207e-03 -0.0222160244 -0.0512764844 -0.0179000291
ORGSRCI   2.879934e-39  2.485901e-04  1.510222e-03  0.0077935806  0.0342278349  0.1348531161
```

R 실습

- 로지스틱 회귀모형 + Ridge CV
  - 조율모수 λ에 대해 교차확인법 (k = 10)을 통해 적합시킨 결과.

```
> set.seed(1)
> ridge_cv = cv.glmnet(X_train, y_train, family = 'binomial', alpha = 0, type.measure="auc")
> bestlam = ridge_cv$lambda.min
> ridge = glmnet(X_train, y_train, family = 'binomial', alpha = 0, lambda = bestlam)
> ridge$lambda
[1] 0.005329218
> ridge$beta
26 x 1 sparse Matrix of class "dgCMatrix"
                      s0
AGE       -0.0305310222
INCOME    -0.0013637922
SEXM       0.0072689774
MARRIED    0.5125657599
FICO      -0.0054580277
OWNHOME   -0.3498657642
LOCB      -0.0895459348
LOCC       0.3027882553
LOCD       0.4061275100
LOCE      -0.1554666640
LOCF      -0.1348673412
LOCG      -0.1238465136
LOCH      -0.1268344594
BUY6      -0.1662867578
```

R 실습

- 로지스틱 회귀모형 + lasso
  - 여기서 λ은 0보다 큰 조율모수.

```
> lasso.fit = glmnet(X_train, as.factor(y_train), alpha = 1,
+                    family="binomial")
> lasso.fit$lambda[c(1, 10, 20, 30, 40, 50)]
[1] 0.0261754388 0.0113307226 0.0044690705 0.0017626935 0.0006952426 0.0002742180
> lasso.fit$beta[,c(1, 10, 20, 30, 40, 50)]
26 x 6 sparse Matrix of class "dgCMatrix"
                s0           s9          s19           s29           s39           s49
AGE       .  -0.005299303 -0.019337839 -2.665943e-02 -0.0298494481 -0.0311155123
INCOME    .   .           -0.001298873 -3.401180e-03 -0.0043372655 -0.0046929124
SEXM      .   .            .            .             .             .
MARRIED   .   .            0.190705849  3.368980e-01  0.3973332932  0.4201472825
FICO      .   .           -0.001241910 -2.561771e-03 -0.0031150115 -0.0033405460
OWNHOME   .  -0.166842286 -0.382340381 -4.682495e-01 -0.5082933792 -0.5192000120
LOCB      .   .           -0.039723732 -1.569195e-01 -0.2478038084 -0.3306573108
LOCC      .   .            0.297597418  3.788900e-01  0.3782345245  0.3303140950
LOCD      .   .            0.102308065  1.889710e-01  0.1781485584  0.1265907021
LOCE      .   .            .           -6.957188e-02 -0.1523027519 -0.2335745230
LOCF      .   .            .            .            -0.0211315609 -0.1000974567
LOCG      .   .            .            .             .            -0.0373461969
LOCH      .   .            .           -1.318631e-02 -0.0821089261 -0.1558777597
BUY6      .   .            .            .            -0.0086043261 -0.0173791672
BUY12     .   .            .           -2.757622e-01 -0.3853027967 -0.4257979600
BUY18     .   0.324379835  0.439410185  6.511618e-01  0.7868699463  0.8430223387
VALUE24   .   .            .           -7.410242e-05 -0.0003886507 -0.0005300746
ORGSRCD   .   .            .           -3.311372e-02 -0.0580339035 -0.0296856867
ORGSRCI   .   .            .            .             .             0.0849279869
```

R 실습

- 로지스틱 회귀모형 + lasso CV
  - 조율모수 C에 대해 교차확인법 (k = 10)을 통해 적합시킨 결과.

```
> set.seed(1)
> lasso_cv = cv.glmnet(X_train, y_train, family = 'binomial', alpha = 1, type.measure="auc")
> bestlam = lasso_cv$lambda.min
> lasso = glmnet(X_train, y_train, family = 'binomial', alpha = 1, lambda = bestlam)
> lasso$lambda
[1] 0.001704959
> lasso$beta
26 x 1 sparse Matrix of class "dgCMatrix"
                      s0
AGE       -3.043130e-02
INCOME    -4.723494e-05
SEXM       .
MARRIED    4.981849e-01
FICO      -5.024747e-03
OWNHOME   -3.406842e-01
LOCB       .
LOCC       3.547119e-01
LOCD       4.631173e-01
LOCE      -1.264350e-02
LOCF       .
LOCG       .
LOCH       .
BUY6      -1.186681e-01
```

R 실습

- 로지스틱 회귀모형, 전진선택법, Ridge, Lasso 모형들의 회귀계수 비교.

```
4 x 27 sparse Matrix of class "dgCMatrix"
            (Intercept)      AGE   INCOME     SEXM MARRIED     FICO  OWNHOME     LOCB    LOCC
logit           2.86855 -0.03486 -0.00137 0.00658 0.59709 -0.00593 -0.37642 -0.16444 0.24943
logit+AIC       2.60495 -0.03531  .        .       0.59800 -0.00588 -0.40094  .       0.43353
logit+Ridge     2.31487 -0.03053 -0.00136 0.00727 0.51257 -0.00546 -0.34987 -0.08955 0.30279
logit+Lasso     1.86763 -0.03043 -0.00005 .       0.49818 -0.00502 -0.34068  .       0.35471
                LOCD     LOCE     LOCF     LOCG     LOCH     BUY6    BUY12   BUY18  VALUE24
logit       0.35932 -0.23252 -0.21025 -0.17762 -0.18524 -0.14621 -0.46006 0.81307 -0.00009
logit+AIC   0.54553  .        .        .        .        .       -0.53808 0.78870  .
logit+Ridge 0.40613 -0.15547 -0.13487 -0.12385 -0.12683 -0.16629 -0.31734 0.67016  0.00012
logit+Lasso 0.46312 -0.01264  .        .        .       -0.11867 -0.29728 0.67276  .
            ORGSRCD ORGSRCI ORGSRCO ORGSRCP  ORGSRCR ORGSRCU  DISCBUY RETURN24    COA6
logit       -0.17629 0.30113 0.01723 0.14661 -0.03889 0.02001 -0.07190 -0.26786 0.18283
logit+AIC   -0.21005 .       .       .       .       .       .        .        .
logit+Ridge -0.16343 0.27390 0.01963 0.13483 -0.03541 0.01746 -0.06099 -0.24550 0.18053
logit+Lasso -0.12224 0.06538 .       0.07985  .       .       -0.00988 -0.15139 0.05961
```

R 실습

- 랜덤포레스트.
  - $p = \sqrt{\text{변수의 수}} \approx 5$, 나무수는 100으로 설정.

```
> library(randomForest)
> set.seed(2)
> rf = randomForest(X_train, as.factor(y_train), ntree = 100,
+                   p = floor(sqrt(ncol(X_train))))
>
> rf

Call:
 randomForest(x = X_train, y = as.factor(y_train), ntree = 100,
   proximity = floor(sqrt(ncol(X_train))))
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 5

        OOB estimate of  error rate: 7.47%
Confusion matrix:
     0 1  class.error
0 5967 4 0.0006699045
1  478 5 0.9896480331
```

R 실습

- 랜덤포레스트.
  - 나무의 수를 결정하기 위해 Validation set을 설정하고,

    Validation set에서 AUC가 최대가 되는 나무 수를 선택.

```
> set.seed(1)
> val_ind = sample(1:nrow(X_train), size = floor(nrow(X_train)*0.2))
> val_auc = c();  val_err = c()
> candidates = seq(from = 20, length = 20, by = 20)
> for (i in candidates){
+   rf_train = randomForest(X_train[-val_ind,], as.factor(y_train[-val_ind]),
+                           ntree = i, p = floor(sqrt(ncol(X_train))))
+   pred_prob_val = predict(rf_train, X_train[val_ind,], type = 'prob')[,2]
+   auc = performance(prediction(pred_prob_val , y_train[val_ind]) , "auc")@y.values
+   val_auc = c(val_auc, auc[[1]])
+   val_err = c(val_err, mean(y_train[val_ind] != round(pred_prob_val)))
+ }
> val_auc
 [1] 0.5689664 0.5332899 0.5663613 0.5740168 0.5484538 0.5969748 0.5833866 0.6053109 0.5628109 0.5882605
[11] 0.5916597 0.5891261 0.5893613 0.5969580 0.5879118 0.5812395 0.5863235 0.5664916 0.5944580 0.5882941
> candidates[which.max(val_auc)]
[1] 160
```

R 실습

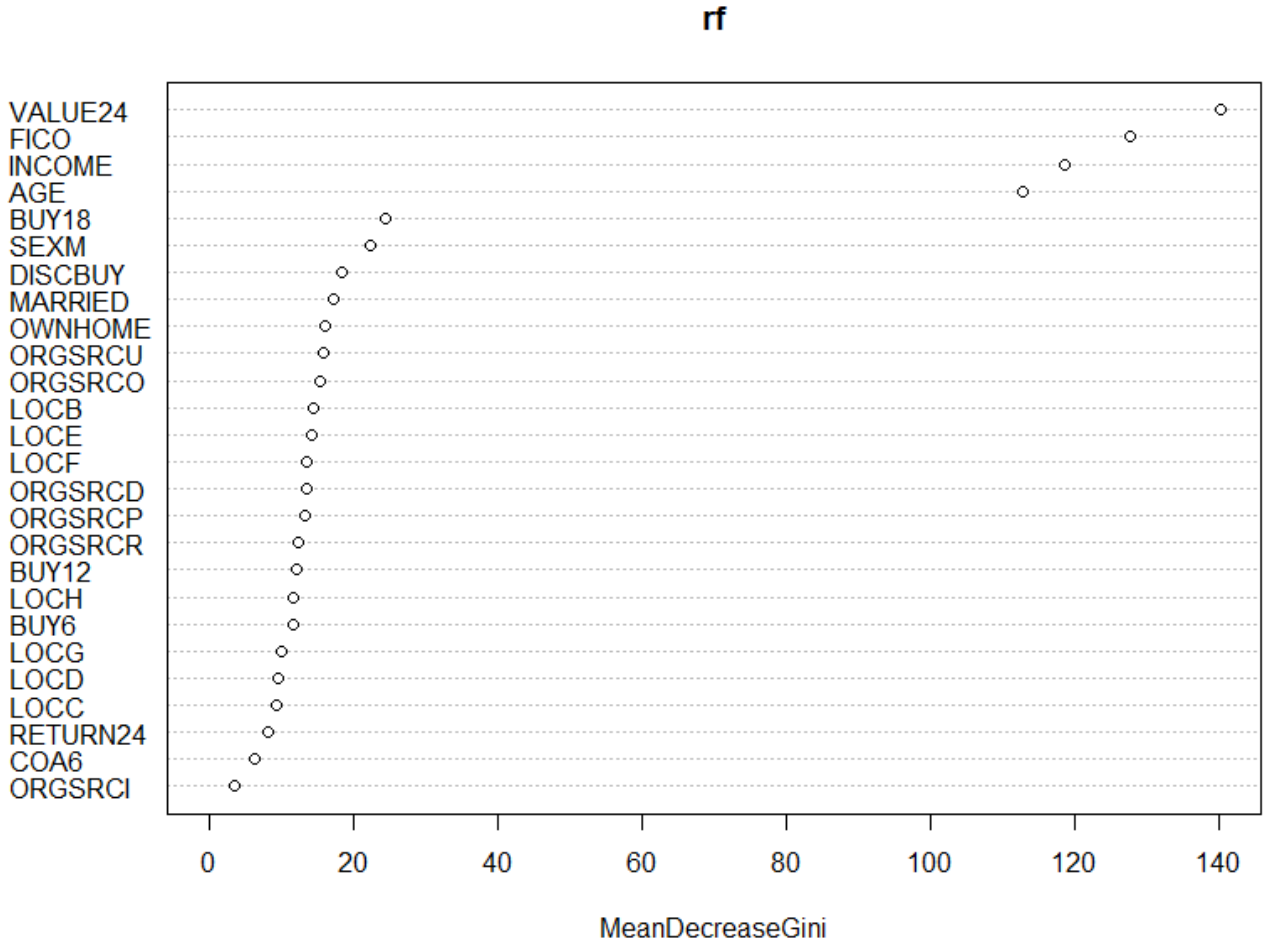
- 랜덤포레스트.
  - 앞의 결과에 의해 나무수를 160으로 설정.

```
> rf = randomForest(X_train, as.factor(y_train), ntree = candidates[which.max(val_auc)],
+                   p = floor(sqrt(ncol(X_train))))
> rf

Call:
 randomForest(x = X_train, y = as.factor(y_train), ntree = candidates[which.max(val_auc)],
proximity = floor(sqrt(ncol(X_train))))
               Type of random forest: classification
                     Number of trees: 160
No. of variables tried at each split: 5

        OOB estimate of  error rate: 7.44%
Confusion matrix:
     0 1  class.error
0 5968 3 0.0005024284
1  477 6 0.9875776398
```

R 실습

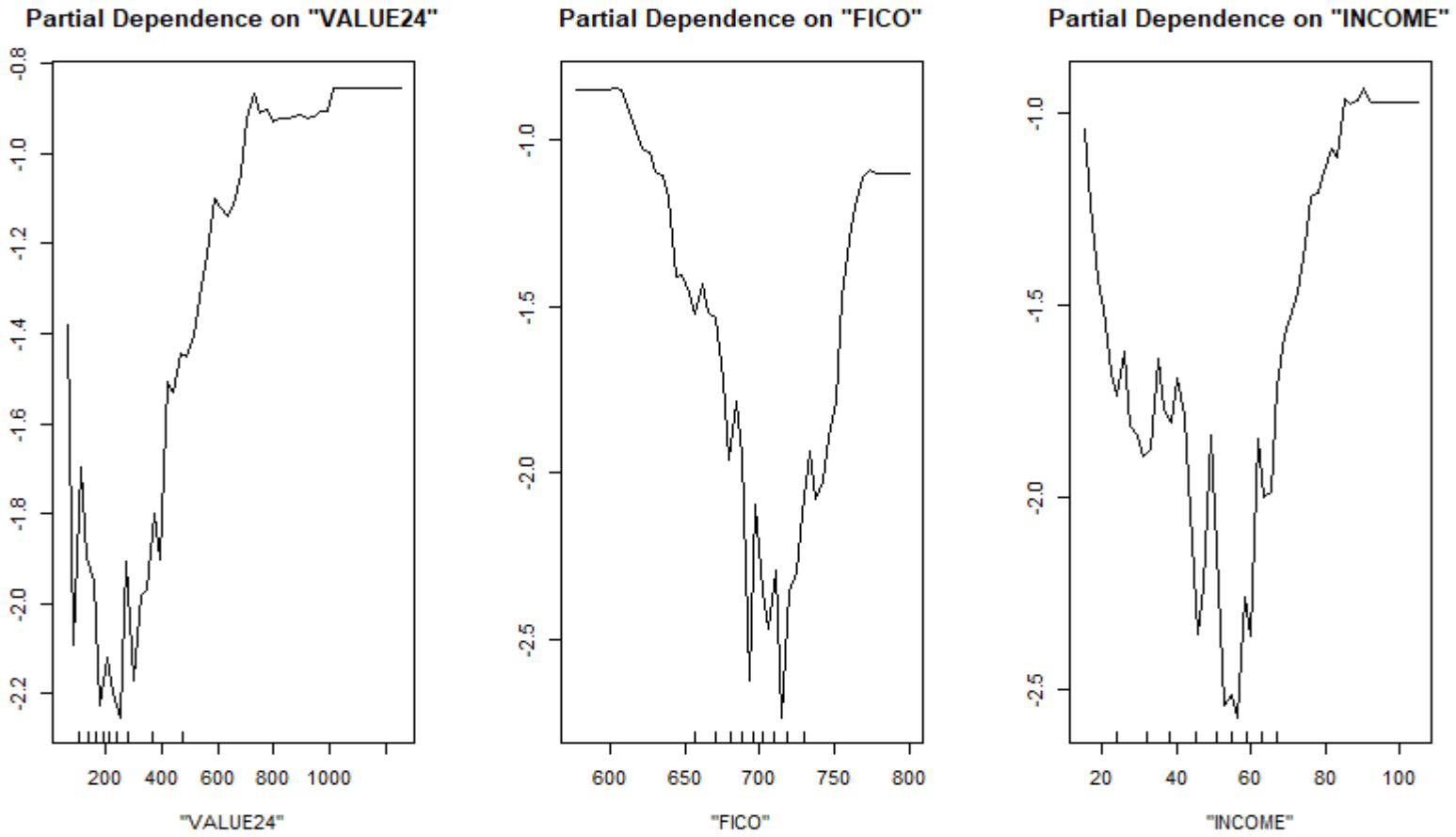
- 랜덤포레스트.
  - 변수의 중요도 (Relative importance) 평가

```
> rf$importance
          MeanDecreaseGini
AGE             112.967459
INCOME          118.708831
SEXM             22.491785
MARRIED          17.350333
FICO            127.857991
OWNHOME          16.215769
LOCB             14.580147
LOCC              9.304816
LOCD              9.615182
LOCE             14.250930
LOCF             13.621949
LOCG             10.161023
LOCH             11.833365
BUY6             11.737501
BUY12            12.168855
BUY18            24.438602
VALUE24         140.325159
ORGSRCD          13.521165
ORGSRCI           3.611188
ORGSRCO          15.534923
ORGSRCP          13.308144
ORGSRCR          12.444368
ORGSRCU          15.810644
```



R 실습

- 랜덤포레스트.
  - Relative importance을 계산하였을 때 상위 3개의 변수에 대한 Partial Dependence Plot.



R 실습

- 부스팅.
  - eta: learning rate, max_depth: 나무의 최대 깊이, nround: 반복 최대 횟수 (최대 나무 수), eval_metric = 오분류율.

```
> library(xgboost)
> set.seed(1)
> xgb = xgboost(data = X_train, label = y_train, max.depth = 1,
+               eta = 0.1, nround = 10, objective = "binary:logistic")
[1]     train-error:0.074837
[2]     train-error:0.074837
[3]     train-error:0.074837
[4]     train-error:0.074837
[5]     train-error:0.074837
[6]     train-error:0.074837
[7]     train-error:0.074837
[8]     train-error:0.074837
[9]     train-error:0.074837
[10]    train-error:0.074837
```

R 실습

- 부스팅.
  - 나무의 최대 깊이를 3으로 설정하고, eval_metric에 auc을 이용한 경우.

```
> xgb = xgboost(data = x_train, label = y_train, max.depth = 3, eval_metric = "auc",
+               eta = 0.1, nround = 10, objective = "binary:logistic")
[1]     train-auc:0.587280
[2]     train-auc:0.599829
[3]     train-auc:0.636172
[4]     train-auc:0.646835
[5]     train-auc:0.664934
[6]     train-auc:0.664098
[7]     train-auc:0.668649
[8]     train-auc:0.669646
[9]     train-auc:0.669538
[10]    train-auc:0.670883
```

R 실습

- 부스팅.
  - 적절한 나무의 최대 수와 최대 깊이를 설정하기 위해, 랜덤포레스트와 마찬가지로 Validation set을 이용.

```
> depth_can = c(1, 2, 3, 4)
> val_auc = matrix(NA, nrow = length(tree_can), ncol = length(depth_can))
> for (j in 1:length(depth_can)){
+   xgb_train = xgboost(data = X_train[-val_ind,], label = y_train[-val_ind], verbose = 0,
+                       eval_metric = "auc", eta = 0.1, objective = "binary:logistic",
+                       max.depth = depth_can[j], nround = max(tree_can))
+   for (i in 1:length(tree_can)){
+     pred_prob_val = predict(xgb_train, X_train[val_ind,], ntreelimit = tree_can[i])
+     val_auc[i,j] = performance(prediction(pred_prob_val , y_train[val_ind]) , "auc")@y.values[[1]]
+   }
+ }
> val_auc
           [,1]      [,2]      [,3]      [,4]
 [1,] 0.5744076 0.5932899 0.6171891 0.6222563
 [2,] 0.6012479 0.6245546 0.6352395 0.6276807
 [3,] 0.6222059 0.6351513 0.6427521 0.6346387
 [4,] 0.6326723 0.6372017 0.6424580 0.6388655
 [5,] 0.6367059 0.6350630 0.6406429 0.6306218
 [6,] 0.6396891 0.6346303 0.6405798 0.6245714
 [7,] 0.6394328 0.6310588 0.6372269 0.6202101
 [8,] 0.6394790 0.6272143 0.6364622 0.6233109
 [9,] 0.6383613 0.6266429 0.6295546 0.6122605
[10,] 0.6393571 0.6244328 0.6225882 0.6122437
```

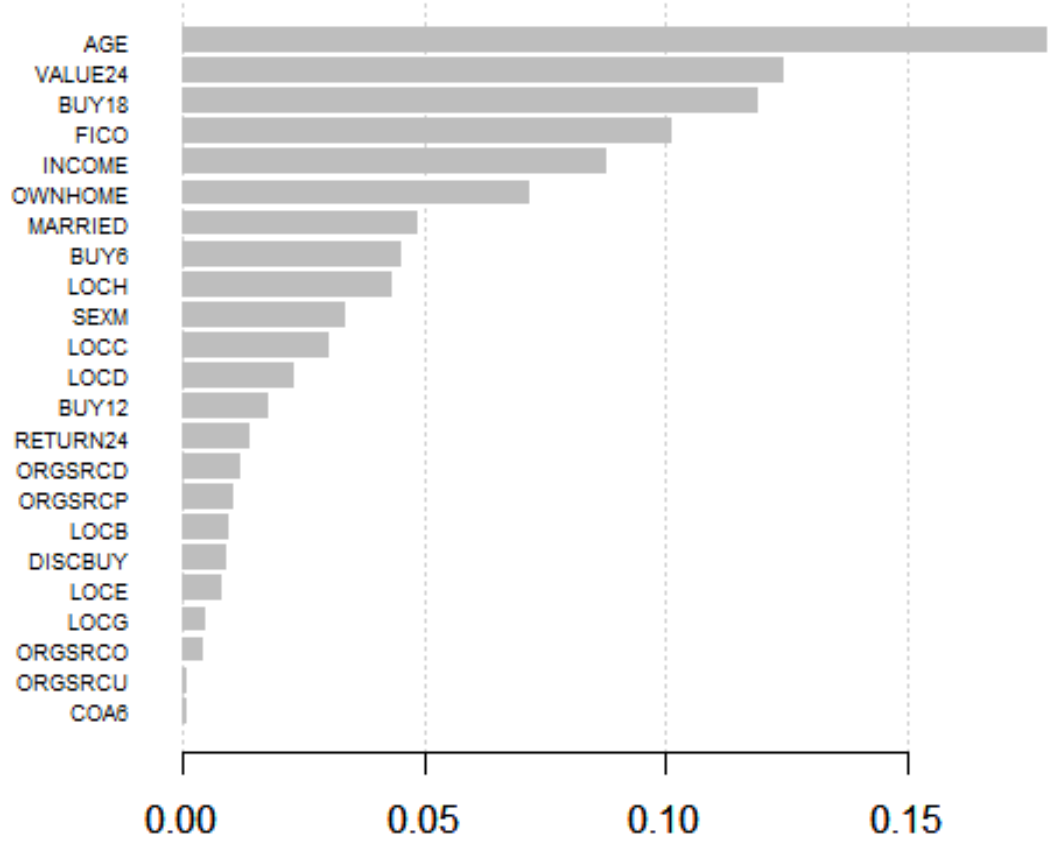
R 실습

- 부스팅.
  - 앞의 결과로 나무의 최대 깊이는 3으로 최대 나무수는 60으로 설정.
  - 변수 중요도 평가

```
> xgb = xgboost(data = X_train, label = y_train, max.depth = 3, eval_metric = "auc",
+               eta = 0.1, nround = 60, objective = "binary:logistic", verbose = 0)
>
> # Relative Importance
> import_mat = xgb.importance(colnames(X_train), model = xgb)
> print(import_mat)
      Feature         Gain        Cover   Frequency
 1:       AGE 0.1792257544 0.2464172535 0.186440678
 2:   VALUE24 0.1245560971 0.0874670036 0.116222760
 3:     BUY18 0.1192225077 0.2035802467 0.082324455
 4:      FICO 0.1013928800 0.1385675059 0.135593220
 5:    INCOME 0.0877522418 0.0471877326 0.101694915
 6:   OWNHOME 0.0717570688 0.0536756611 0.065375303
 7:   MARRIED 0.0486946413 0.0756370314 0.050847458
 8:      BUY6 0.0455008299 0.0050158548 0.029055690
 9:      LOCH 0.0432518738 0.0103688705 0.026634383
10:      SEXM 0.0334937845 0.0029775296 0.026634383
```

R 실습

- 부스팅.
  - 앞의 결과로 나무의 최대 깊이는 3으로 최대 나무수는 60으로 설정.
  - 변수 중요도 평가

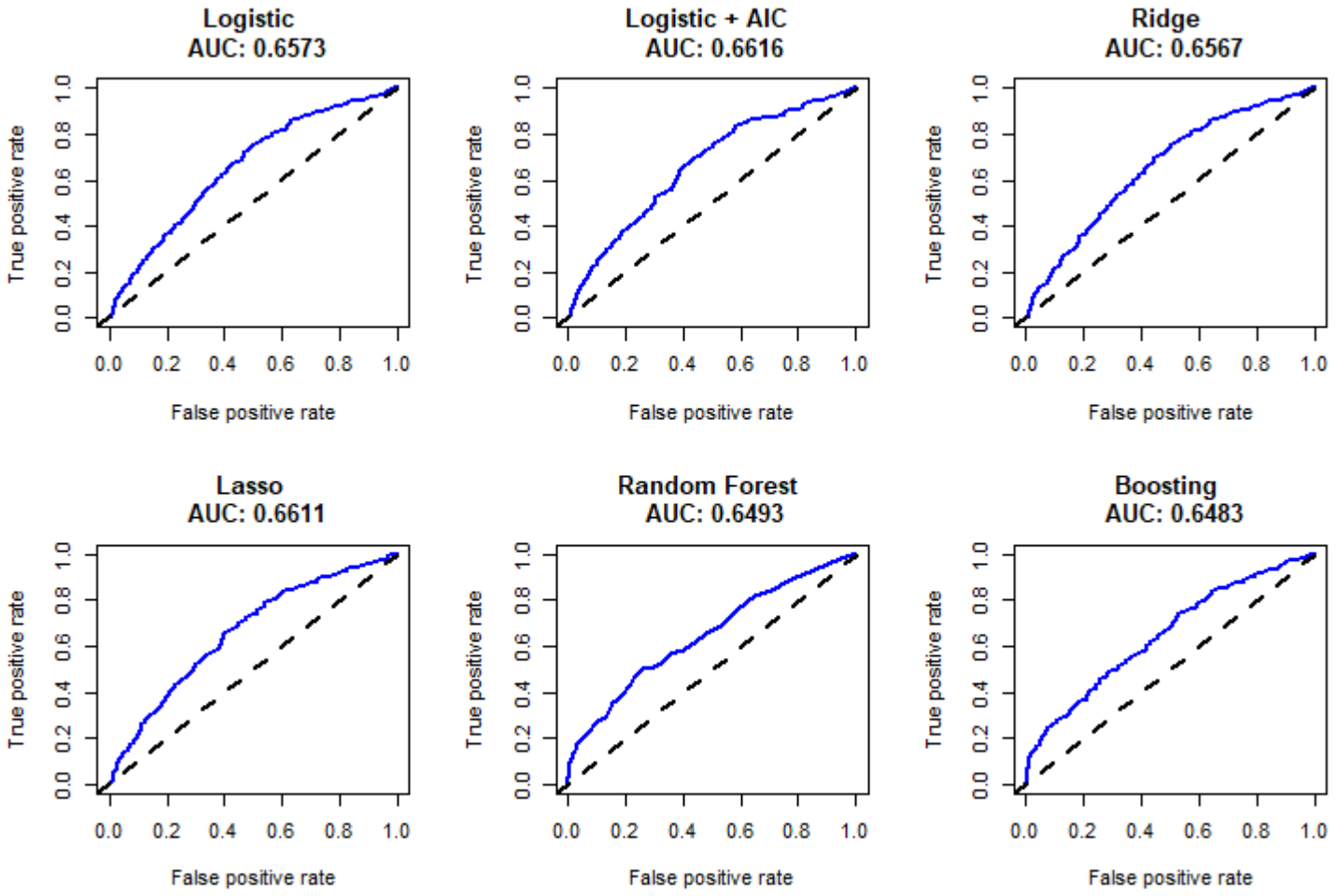


R 실습

- 모형의 비교
  - 로지스틱 회귀분석
  - 로지스틱 회귀분석 + 변수선택 (전진 선택, AIC)
  - 로지스틱 회귀분석 + Ridge CV
  - 로지스틱 회귀분석 + Lasso CV
  - 랜덤포레스트
  - 부스팅
- 모형의 성능 비교: ROC 곡선과 AUC, 리프트 그래프
- 학습 자료를 이용한 절단값의 선택
  - 오분류율을 최소로 하는 절단값
  - 민감도 (Sensitivity)을 0.5 이상으로 하는 절단값

R 실습

- 예측자료에서 적합된 모형을 통해 사후확률을 계산하여 구한 ROC 곡선 .



R 실습

- 학습자료의 오분류율을 최소로 하는 절단값을 선택하여 예측자료에서 예측한 경우.

```
> cutoff_can = seq(0.01, 0.99, by = 0.01)
> cut_sel = c()
> for (i in 1:6){
+   cutoff_out =  t(sapply(cutoff_can,
+                          function(cut) cutoff_res(newx = X_train, response = y_train, cutoff = cut,
+                                                   pred_prob = pred_prob_train[,i])[[1]]))
+   cut_sel[i] = cutoff_out[which.min(cutoff_out[,2]), 1]
+ }
> matrix(t(sapply(1:6, function(i) cutoff_res(newx = X_test, response = y_test,
+                                             cutoff = cut_sel[i], pred_prob = pred_prob_test[,i])[[1]])), nrow = 6,
+        dimnames =list(model_names, c("cutoff","error rate","sensitivity","specificity")))
              cutoff error rate sensitivity specificity
Logistic        0.42     0.0759      0.0048      0.9992
Logistic + AIC  0.44     0.0763      0.0000      0.9992
Ridge           0.37     0.0759      0.0048      0.9992
Lasso           0.39     0.0755      0.0000      1.0000
Random Forest   0.33     0.0777      0.0574      0.9930
Boosting        0.26     0.0773      0.0718      0.9922
```

R 실습

- 민감도를 0.5이상으로 하는 절단값을 선택하여 예측자료에서 예측한 경우.

```
> cut_sel = c()
> for (i in 1:6){
+   cutoff_out =  t(sapply(cutoff_can,
+                          function(cut) cutoff_res(newx = X_train, response = y_train, cutoff = cut,
+                                                   pred_prob = pred_prob_train[,i])[[1]]))
+   cut_sel[i] = cutoff_out[tail(which(cutoff_out[,3] >= 0.5), n = 1), 1]
+ }
> matrix(t(sapply(1:6, function(i) cutoff_res(newx = X_test, response = y_test,
+                                             cutoff = cut_sel[i], pred_prob = pred_prob_test[,i])[[1]])),
+        dimnames =list(model_names, c("cutoff","error rate","sensitivity","specificity")))
              cutoff error rate sensitivity specificity
Logistic         0.08     0.3184      0.5120      0.6955
Logistic + AIC   0.08     0.3162      0.5120      0.6978
Ridge            0.08     0.3209      0.5263      0.6916
Lasso            0.08     0.3231      0.5311      0.6888
Random Forest    0.59     0.0752      0.0048      1.0000
Boosting         0.10     0.2165      0.3493      0.8190
```

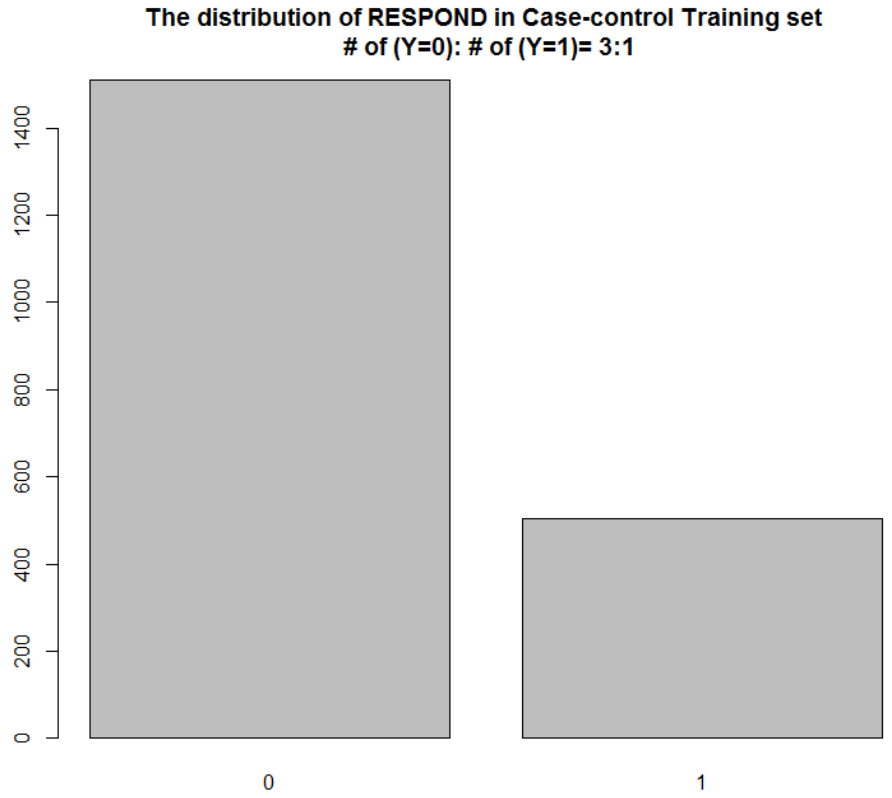
R 실습 (Case-control)

- Case-control sampling
  - 데이터의 반응변수의 분포가 불균형하기 때문에 학습 자료에서 "RESPOND = 0"인 자료의 랜덤 추출을 진행.
  - 이 때 랜덤 추출할 자료의 수는 "RESPOND = 1"인 자료의 수의 3배.
  - 위에서 추출된 자료와 "RESPOND = 1"의 자료를 Case-control 학습 자료라 하고 이 자료를 이용하여 앞의 네 개의 모형을 적합.
- Case-control 이용하여 적합된 모형으로 앞서 진행하였던 모형의 성능 비교.
  - ROC와 AUC, 리프트 그래프 비교.
  - Case-control 학습 자료를 이용하여 절단값을 추정, 예측자료에서의 오분류율, 민감도, 특이도 비교.

R 실습 (Case-control)

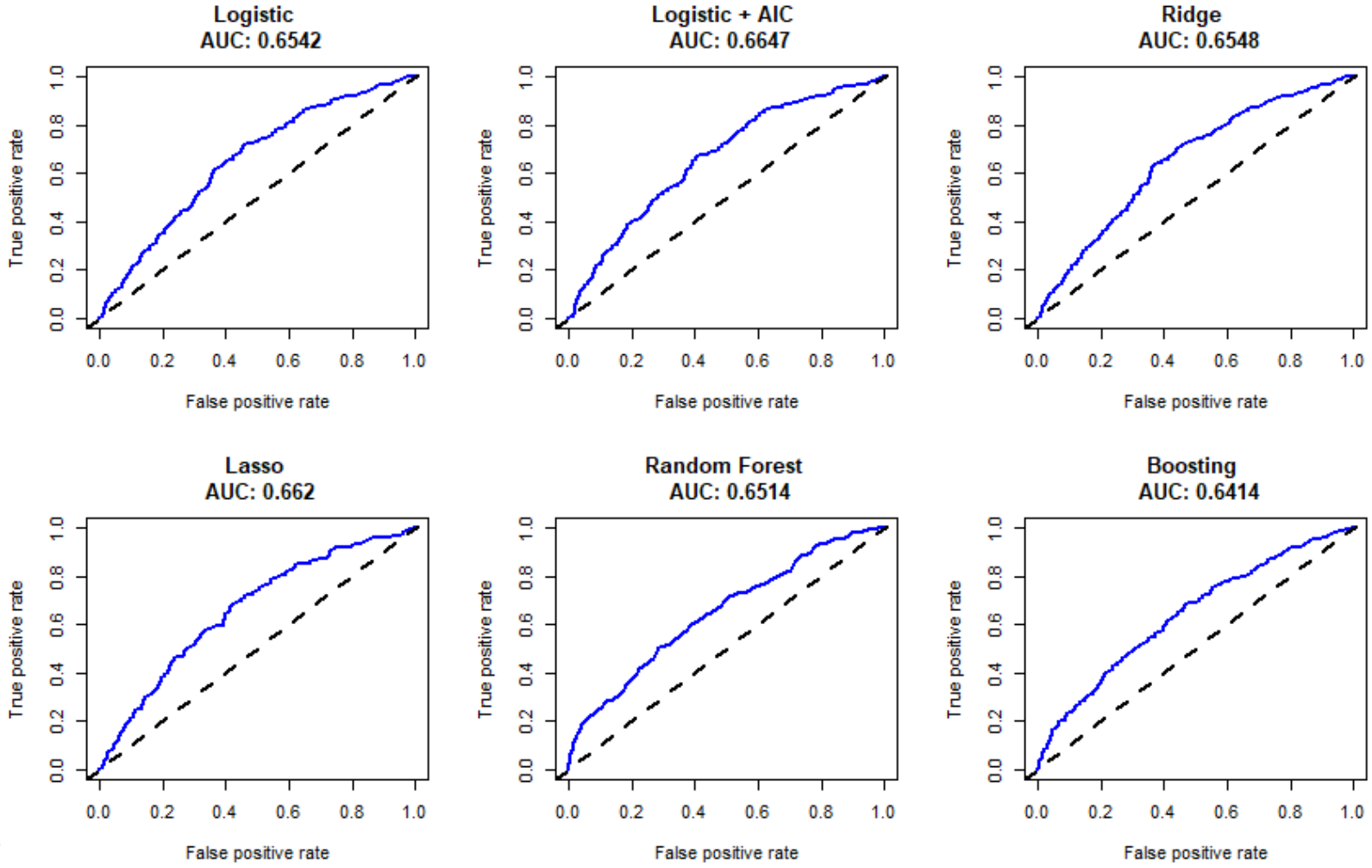
- Case-control sampling 학습 자료의 형태

```
> n1 = sum(y_train == 1)
> set.seed(1)
> cc_ind = sample(1:sum(y_train == 0),
+                 size = 3*n1, replace = F)
> cc_data = rbind(train[y_train == 1,],
+                 train[y_train == 0,][cc_ind,])
> table(cc_data$RESPOND)

   0    1
1509  503
> dim(cc_data)
[1] 2012   27
```



R 실습 (Case-control)

- 예측자료에서 적합된 모형을 통해 사후확률을 계산하여 구한 ROC 곡선.



R 실습 (Case-control)

- 학습자료의 오분류율을 최소로 하는 절단값을 선택하여 예측자료에서 예측한 경우.

	cutoff	error rate	sensitivity	specificity
Logistic	0.45	0.1113	0.1100	0.9523
Logistic + AIC	0.50	0.0922	0.0574	0.9773
Ridge	0.44	0.1077	0.1053	0.9566
Lasso	0.38	0.1146	0.1148	0.9484
Random Forest	0.33	0.2382	0.3876	0.7924
Boosting	0.38	0.1446	0.2297	0.9066

- 민감도를 0.5로 하는 절단값을 선택하여 예측자료에서 예측한 경우.

	cutoff	error rate	sensitivity	specificity
Logistic	0.27	0.3144	0.4833	0.7021
Logistic + AIC	0.27	0.3010	0.5167	0.7138
Ridge	0.27	0.3119	0.4833	0.7048
Lasso	0.27	0.3007	0.5072	0.7150
Random Forest	0.69	0.0755	0.0000	1.0000
Boosting	0.34	0.1966	0.3062	0.8440

R 실습 (Case-control)

- 전체 학습자료와 Case-Control 학습자료를 각각 이용한 경우의 비교 1
  각 학습자료를 이용하여 적합된 모형으로 예측자료에서의 AUC을 구한 결과

	전체 학습자료	Case-Control 학습자료
Logistic	0.6573	0.6542
Logistic + AIC	0.6616	0.6647
Logistic + Ridge CV	0.6567	0.6548
Logistic + Lasso CV	0.6611	0.6620
Random Forest	0.6493	0.6514
Boosting	0.6483	0.6414

R 실습 (Case-control)

- 전체 학습자료와 Case-Control 학습자료를 각각 이용한 경우의 비교 2
  학습자료에서의 민감도를 0.5로 하는 절단값을 구하여 예측자료에 적용한 경우

	전체 학습자료			Case-Control 학습자료		
	오분류율	민감도	특이도	오분류율	민감도	특이도
Logistic	0.3184	0.5120	0.6955	0.3144	0.4833	0.7021
Logistic + AIC	0.3162	0.5120	0.6978	0.3010	0.5167	0.7138
Logistic + Ridge CV	0.3209	0.5263	0.6916	0.3119	0.4833	0.7048
Logistic + Lasso CV	0.3231	0.5311	0.6888	0.3007	0.5072	0.7150
Random Forest	0.0752	0.0048	1.0000	0.0755	0.0000	1.0000
Boosting	0.2165	0.3493	0.8190	0.1966	0.3062	0.8440

Q & A