

고려대학교  
빅데이터 연구회

# KU-BIG

---

머신러닝 개론스터디

학술부 : 유현우 최홍석 정희정 박정진 박건  
빈



# 목 차

- I 머신러닝
- II 전처리
- III 모델링
- IV Resampling(Validation)
- V 모델 평가

# PART.I 머신러닝

1

머신러닝의  
개념

2

머신러닝의  
종류

## 1. 머신러닝의 개념

### 머신러닝이란?

학습데이터



스스로 학습

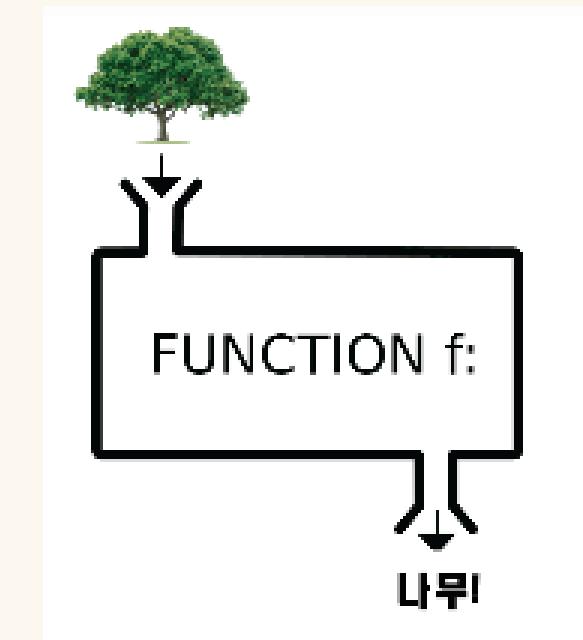
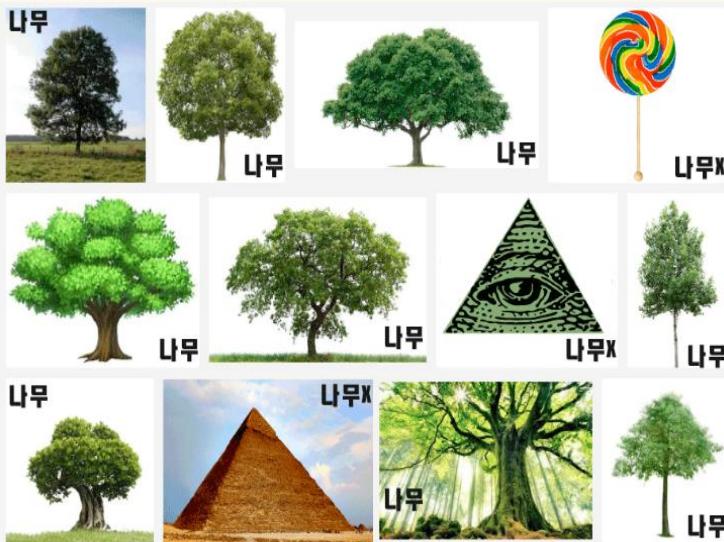


모델 완성

기계가 학습을 진행하면서 자동적으로 패턴  
이나 연관성을 발견

# 1. 머신러닝의 개념

- 수많은 나무와 나무가 아닌 데이터 학습
- 머신러닝을 통해 만들어진 모델 -> 나무인지 아닌지 구별



기계가 스스로 나무인지 아닌지 구별하는  
모델 완성

## 2. 머신러닝의 종류

지도학습

$Y$  (output 변수) 가 존재한다.

비지도학습

$Y$  (output 변수) 가 존재하지 않는다.

VS



강화학습

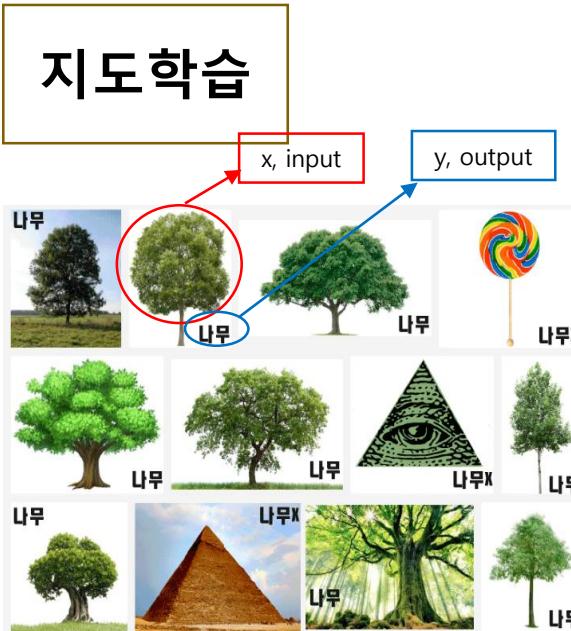
## 2. 머신러닝의 종류

### 지도학습

학습데이터 = x 변수(input) , y 변수(output) 모두를 지칭한다.

- y (output 변수) = Target, Label 이라 부르는 변수가 존재한다.
- 학습데이터를 통해 모델생성 -> 새로운 데이터의 y 변수 예측이 목표!

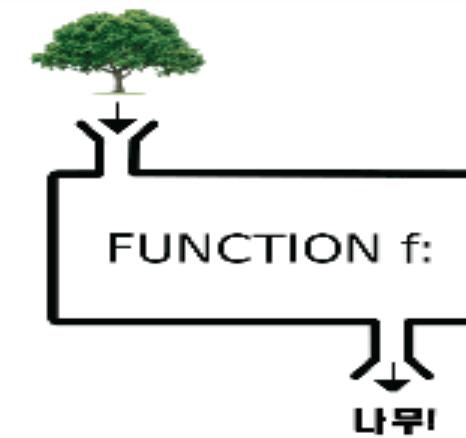
## 2. 머신러닝의 종류



나무 사진 (x 변수, input) 과  
나무인지 아닌지 알려주는 y변  
수, output 를 통해 모델 생성

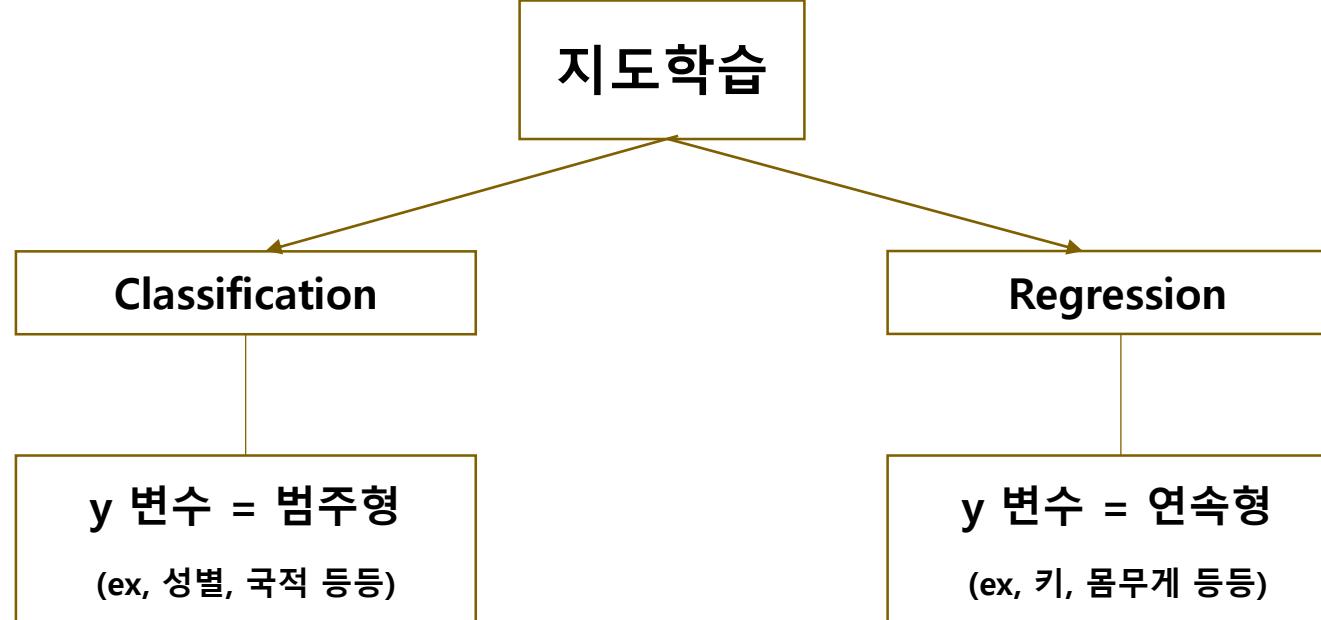


새로운 데이터의 **x** 변수  
(**=input**) 만 주어졌을 때



'나무'라는 y 변수(output)  
을 도출해 내자!

## 2. 머신러닝의 종류



## 2. 머신러닝의 종류

### 지도학습 (Classification)

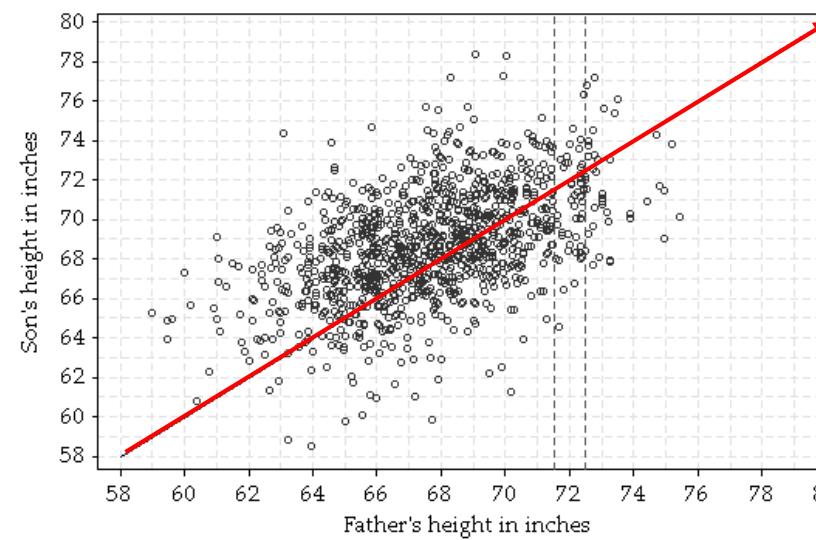
얼굴을 보고 성별 구별



## 2. 머신러닝의 종류

### 지도학습 (Regression)

부모와 자식 간의 키 관계



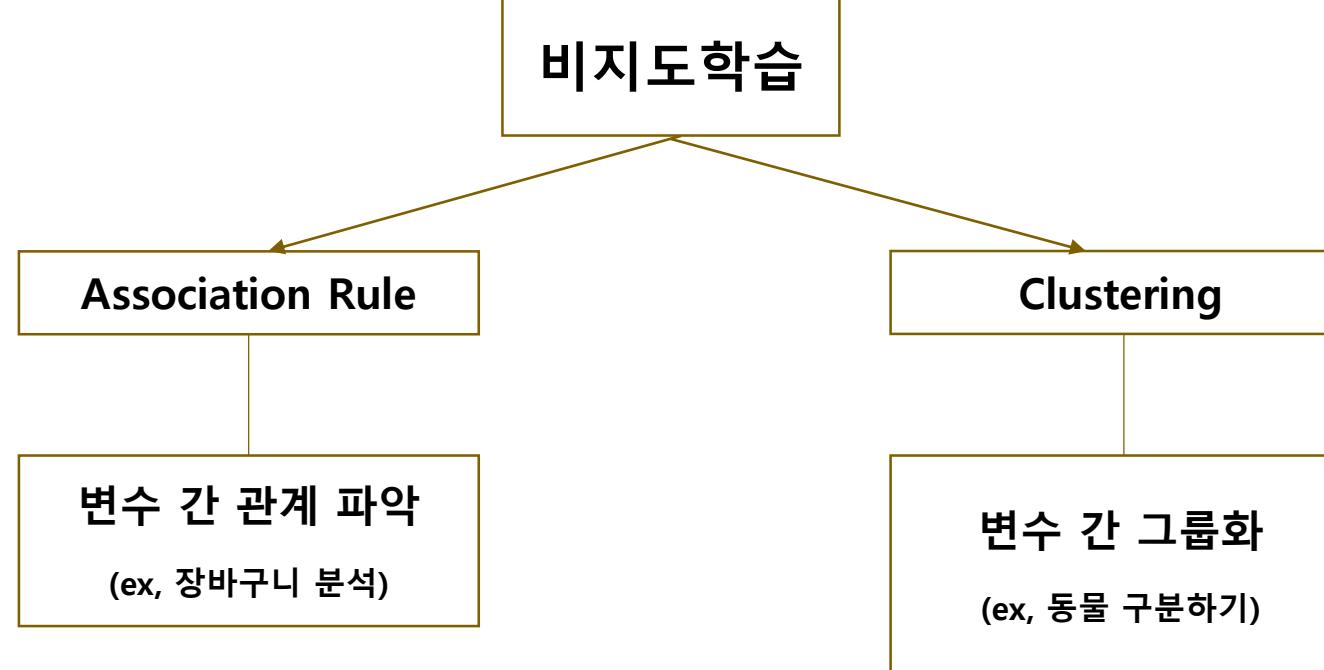
## 2. 머신러닝의 종류

### 비지도학습

학습데이터 = x 변수(input) 만 지칭한다. -비지도학습에는 y변수(output) 이 존재하지 않는다

- y (output 변수) = Target, Label 이라 부르는 변수가 존재하지 않는다.
- 비슷한 학습데이터끼리 그룹화 -> 각 그룹 별 특성을 임의지정 (해석이 자유롭다!)

## 2. 머신러닝의 종류



## 2. 머신러닝의 종류

### 비지도학습 (Association Rule)

#### 장바구니 분석

구매자	구매 리스트
홍길동	맥주, 오징어, 치즈
고길동	맥주, 오징어
이순신	맥주, 오징어, 사이다, 콜라
강감찬	사이다, 오징어, 라면
정약용	맥주, 치즈, 라면



맥주를 사면 오징어, 치즈를 살 가능성이 높겠구나!

## 2. 머신러닝의 종류

### 비지도학습 (Clustering)

비슷한 동물끼리 그룹화



## 2. 머신러닝의 종류

### 강화학습

- $y$  (output 변수) = Target, Label 이라 부르는 변수가 존재하지 않는다.
- 비슷한 학습데이터끼리 그룹화 -> 각 그룹 별 특성을 임의지정 (해석이 자유롭다!)

# PART. II 전처리

1

데이터 셋  
확인

2

결측값  
처리

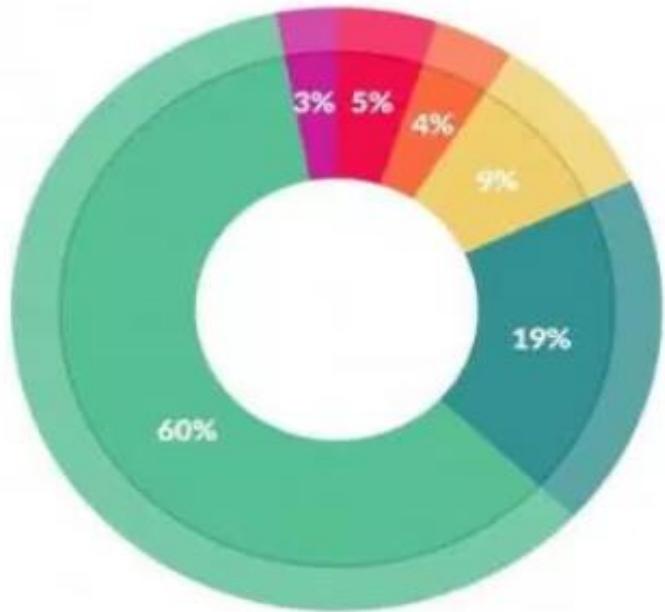
3

이상치  
처리

4

Feature  
Engineering

# 데이터 전처리란?

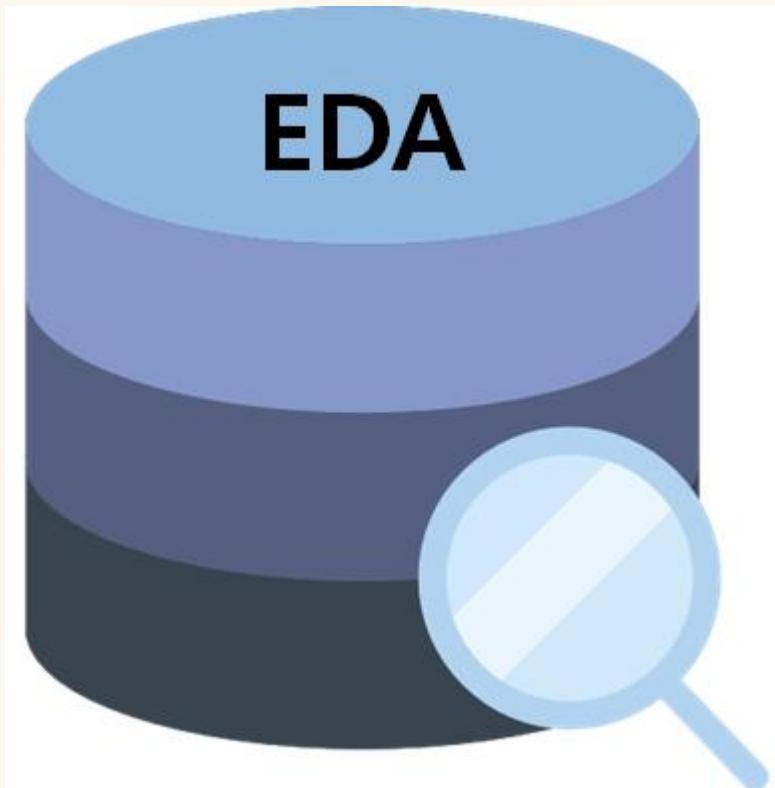


What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60% (highlighted)
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

사실상 데이터 분석 과정의 60%를 차지

## 1. 데이터셋 확인



EDA(탐색적 자료분석)란?

- 가설 검정 등의 이론이 아니라 데이터를 있는 그대로 보여주는 과정에 중점을 둠.
- CDA와는 대비되는 개념
- 저항성, 잔차의 해석, 자료의 재표현, 자료의 현시성

## 1. 데이터셋 확인

### (a) 저항성

: 이상치, 결측치, 이상오류의 영향을 받지 않는 Tool 사용  
Ex) 평균이 아닌 중위수

### (b) 잔차의 해석

: 각 값들이 주된 흐름에서 얼마나 벗어나 있는지 탐색

### (c) 자료의 재표현

: 자료의 여러가지 성질을 나타낼 수 있는 다양한 형태로 표현  
Ex) 자료의 값들에 Log를 취해보기

### (d) 자료의 현시성

: 시각화, 그래프  
Ex) Boxplot 등...

# 1. 데이터셋 확인

○ Company.Maker.if.known. (Factor w/ 416 levels "A. Morin", "Acalli",...)
○ Specific.Bean.Origin.or.Bar.Name (Factor w/ 1039 levels "\\"heirloom\\", Arriba Nacion...
○ REF (int)
○ Review.Date (chr)
○ Cocoa.Percent (num)
○ Company.Location (Factor w/ 60 levels "Amsterdam", "Argentina",...)
○ Rating (num)
○ Bean.Type (Factor w/ 42 levels "", "Amazon", "Amazon mix",...)
○ Broad.Bean.Origin (Factor w/ 101 levels "", "Africa, Caribbean, C. Am.",...)

## 1. 변수 확인

- 독립/종속 변수의 정의, 각 변수의 유형, 변수의 타입을 확인
- 기본적인 부분이지만, 변수의 Type에 따라 모델에서 다른 결과를 도출

# 1. 데이터셋 확인

## 2. RAW 데이터 확인

### 1) 단변량 분석(연속형)

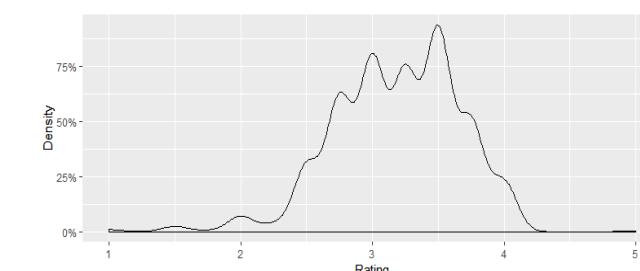
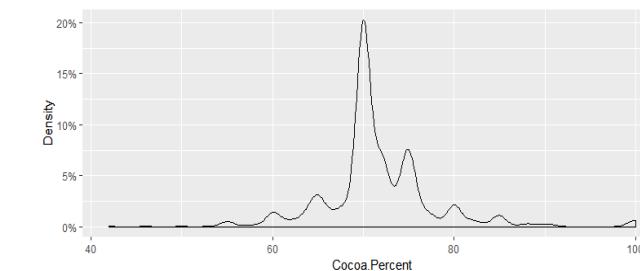
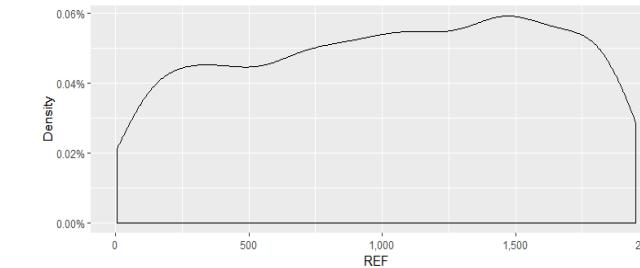
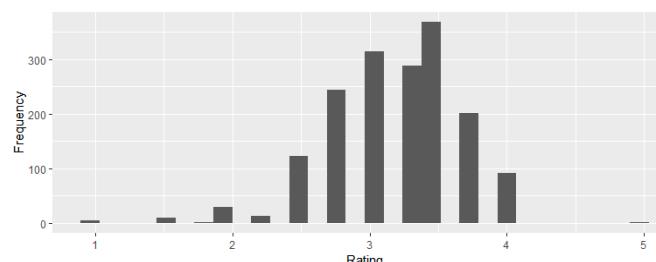
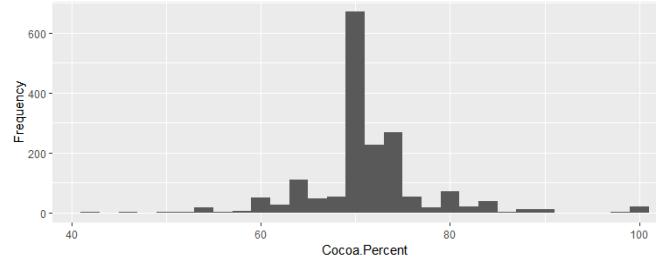
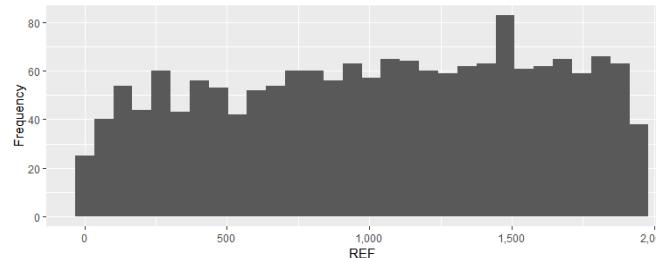
- 군집이 존재

- 집중도 높은 구간

- 대칭성 여부

- 자료의 범위 및 산포

- Y와의 관계

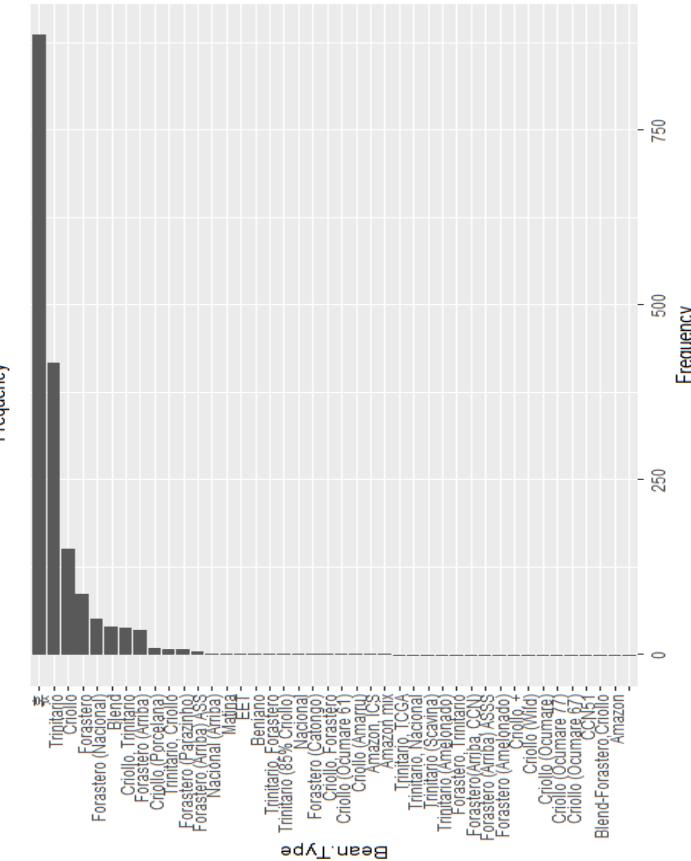
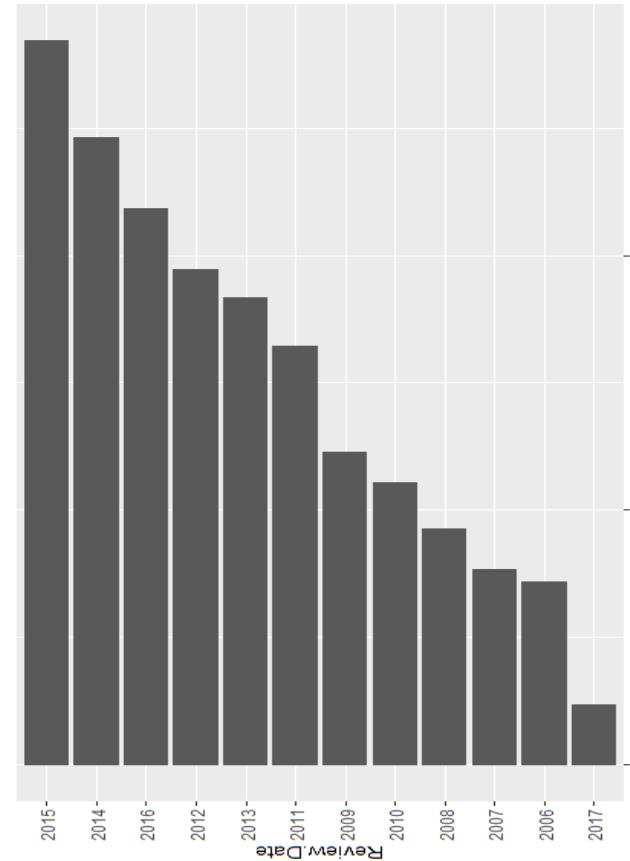


# 1. 데이터셋 확인

## 2. RAW 데이터 확인

1) 단변량 분석(범주형)

- 자료의 분포



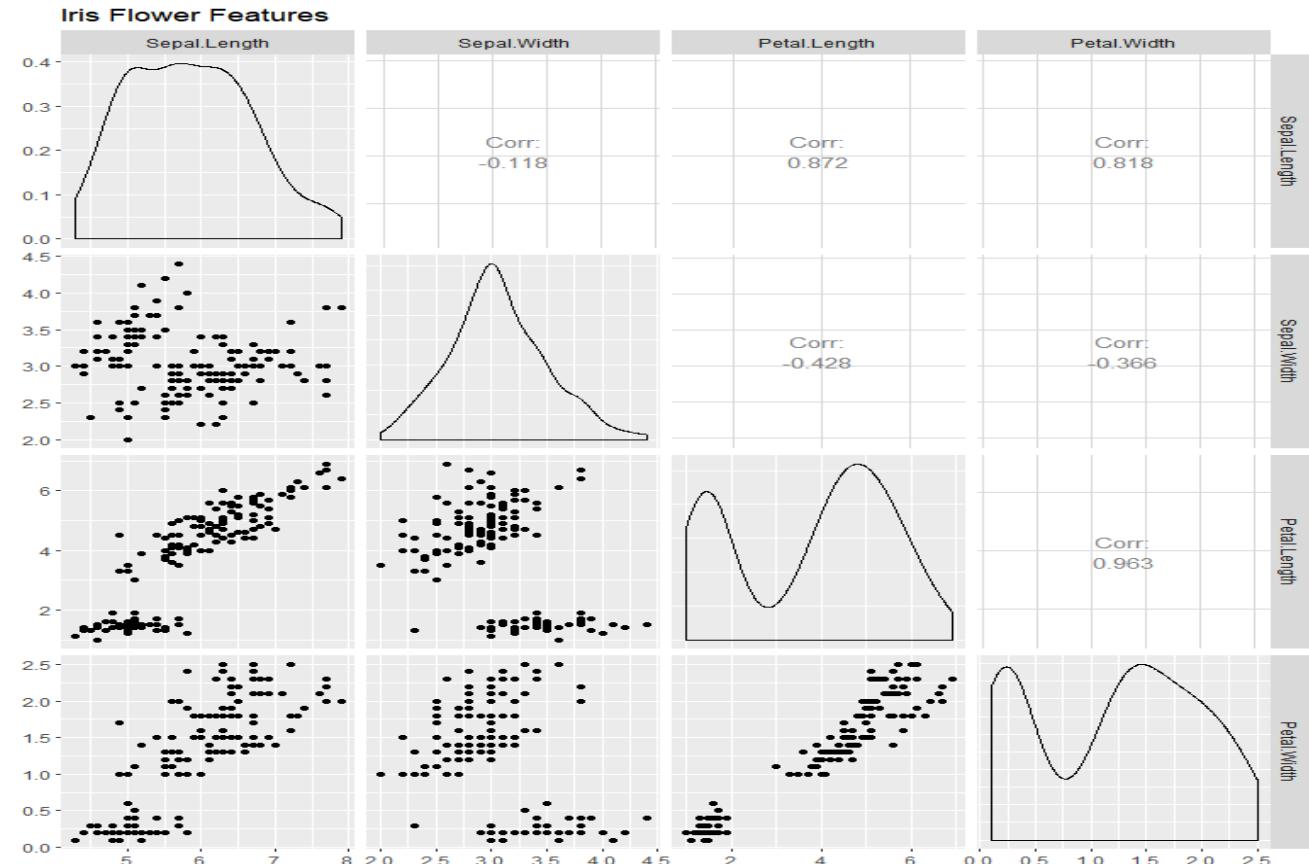
# 1. 데이터셋 확인

## 2. RAW 데이터 확인

2) 이변량(연속형)

-Scatter plot

-Correlation 분석



# 1. 데이터셋 확인

## 2. RAW 데이터 확인

### 2) 이변량(범주형)

범주형 X 범주형

- 누적막대그래프
- 100%기준 누적 막대 그래프
- Chi-Square분석  
(두 변수가 독립적인지 여부)

범주형 X 연속형

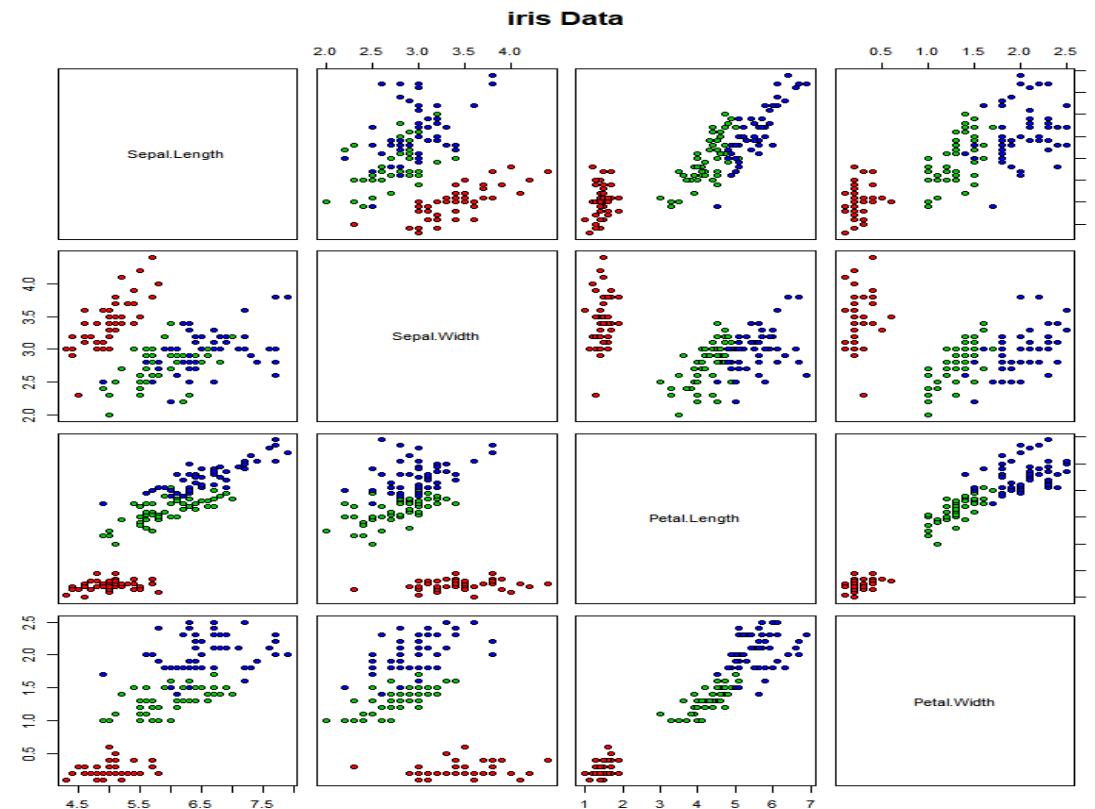
- 누적막대그래프
- 범주 별 Histogram
- 범주의 종류에 따라
  - 2개: T-test/Z-test
  - 3개 이상: ANOVA  
(집단 별 평균 차가 유의한지 여부)

# 1. 데이터셋 확인

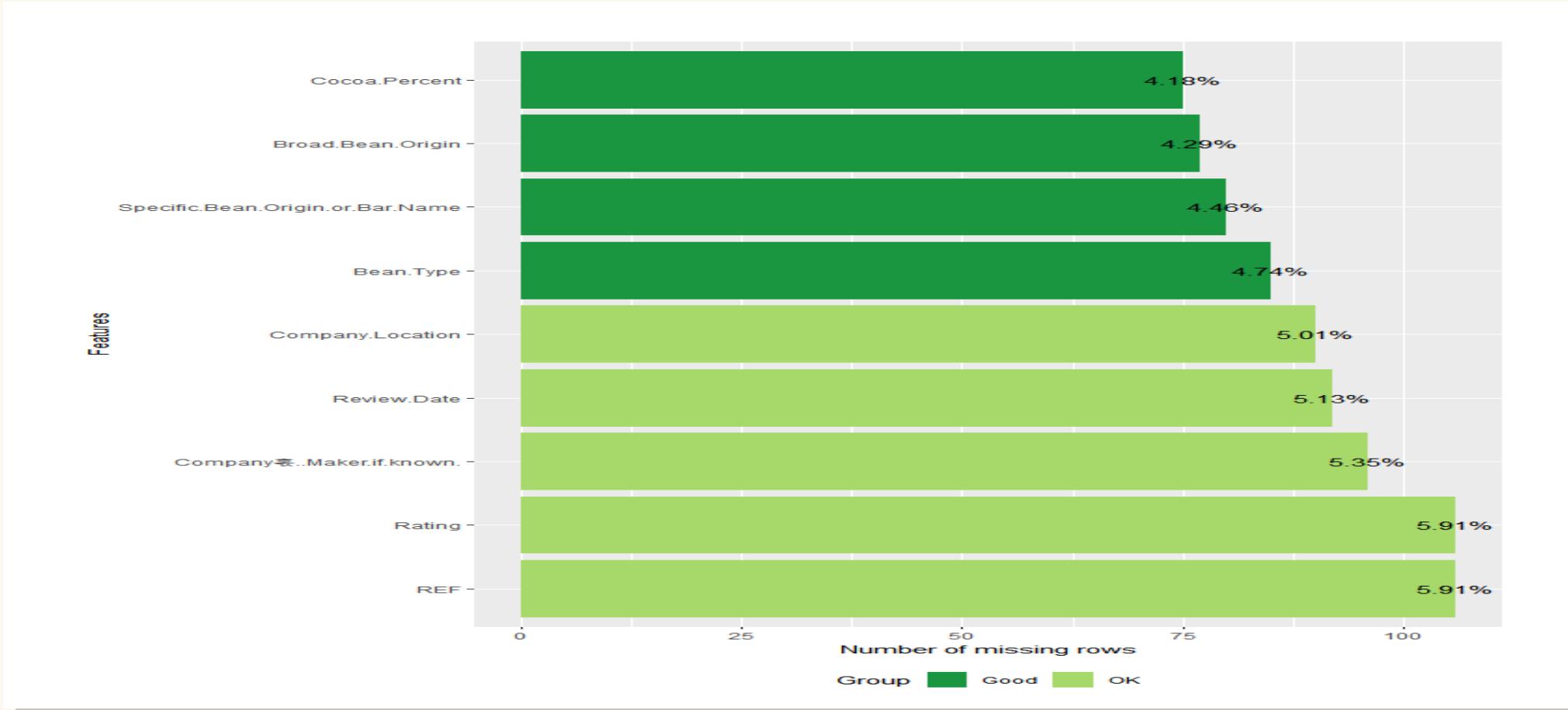
## 2. RAW 데이터 확인

### 3) 셋 이상의 변수

변수 중에 범주형 변수가 있다면 해당 범주 별로 연속형 변수의 관계를 확인



## 2. 결측값 처리



## 2. 결측값 처리

### 1. 삭제

- 결측값이 존재하는 모든 행 삭제(전체 삭제)
- 결측값을 다수 포함하는 변수 삭제(부분 삭제)

하지만 삭제는 결측값이 무작위 발생인 경우

### 2. 다른 값으로 대체

- 평균
- 최빈값
- 중간값

## 2. 결측값 처리

### 3. 예측값 삽입

- 결측값이 없는 관측치를 Training data로 사용해서 결측값을 예측

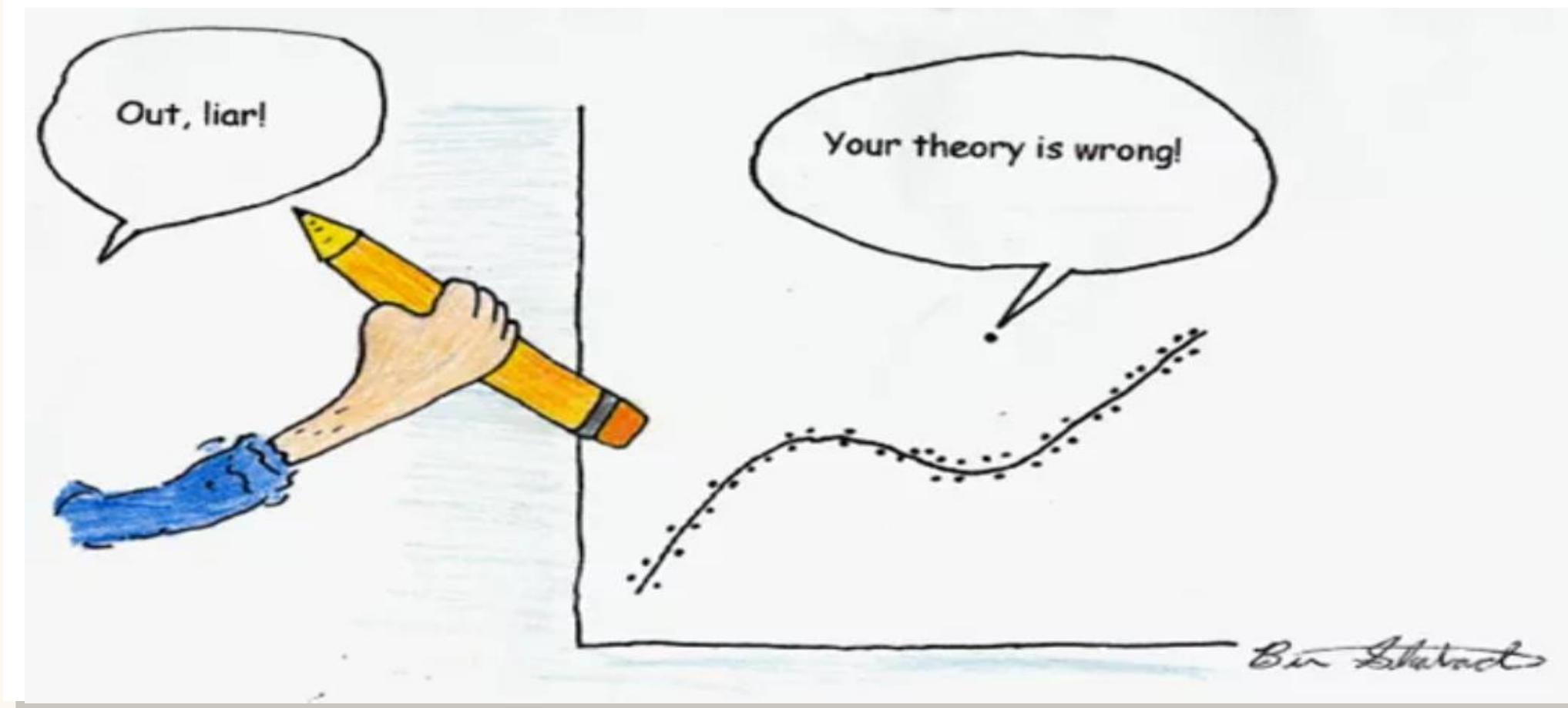
Regression or Logistic regression 사용

- 다중 대입법

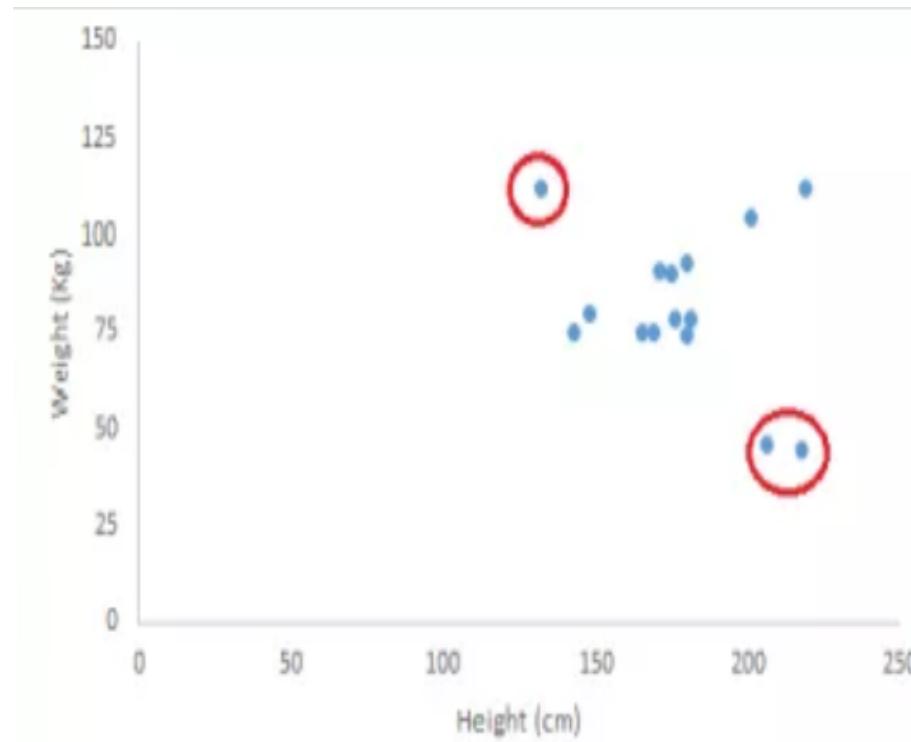
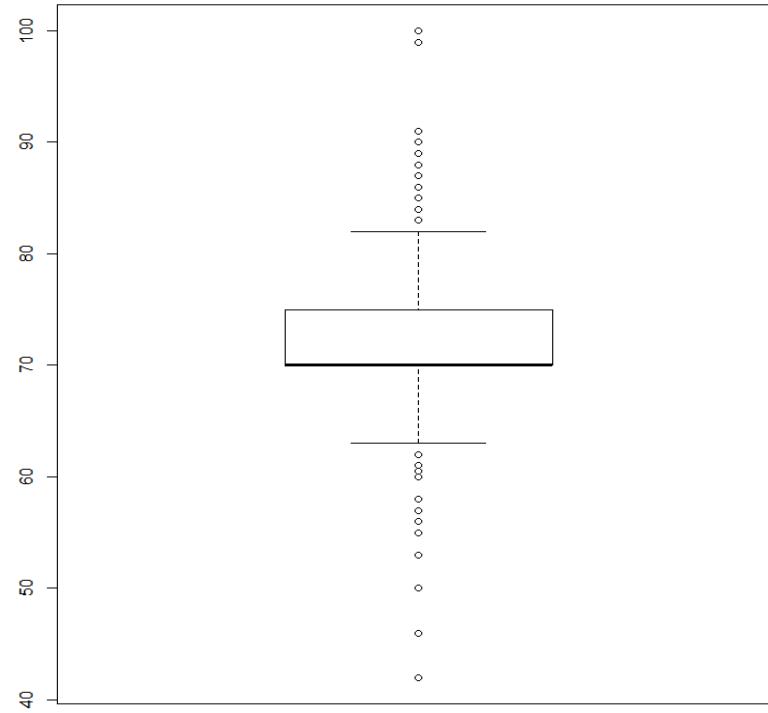
Mice package

But, 결측값이 다양한 변수에서 발생하면 예측력이 낮아질 수 있다.

### 3. 이상값 처리



### 3. 이상값 처리



### 3. 이상값 처리

이상치 검출.

하지만 시각적으로 Outlier을 확인하는 데에는 한계가 있다.

키가 2m면 이상치일 것이다. 하지만 몸무게까지 120kg이라면 합리적일 수 있다.

1. 내면 스튜던트화 잔차 확인
2. Leverage
3. Cooks'D

### 3. 이상값 처리

#### 처리 방법

1. 삭제
2. 상한선 하한선으로 제한
3. 케이스를 분리하여 분석
  - 이상치를 포함한 모델과 포함하지 않은 모델 만들고
  - 각각의 모델에 대한 해석

## 4. Feature Engineering

- Feature Engineering은 머신러닝 알고리즘을 작동하기 위해 데이터에 대한 도메인 지식을 활용하여 특징(Feature)를 만들어내는 과정
- 다른 정의를 살펴보면, 머신러닝 모델을 위한 데이터 테이블의 컬럼(특징)을 생성하거나 선택하는 작업
- 간단히 정리하면, 모델의 성능을 높이기 위해 모델에 입력할 데이터를 만들기 위해 주어진 초기 데이터로부터 특징을 가공하고 생성하는 전체 과정

## 4. Feature Engineering

### ▶ 상관계수

		상관계수				
		만족도평균	불만족평균	슬픔평균	분노평균	기쁨평균
만족도평균	Pearson 상관계수	1	.710**	.073	-.218**	-.324**
	유의확률(양쪽)		.000	.241	.000	.000
	N	259	259	259	259	259
불만족평균	Pearson 상관계수	.710**	1	.035	-.198**	-.207**
	유의확률(양쪽)	.000		.570	.001	.001
	N	259	259	259	259	259
슬픔평균	Pearson 상관계수	.073	.035	1	.209**	.223**
	유의확률(양쪽)	.241	.570		.001	.000
	N	259	259	259	259	259
분노평균	Pearson 상관계수	-.218**	-.198**	.209**	1	.601**
	유의확률(양쪽)	.000	.001	.001		.000
	N	259	259	259	259	259
기쁨평균	Pearson 상관계수	-.324**	-.207**	.223**	.601**	1
	유의확률(양쪽)	.000	.001	.000	.000	
	N	259	259	259	259	259

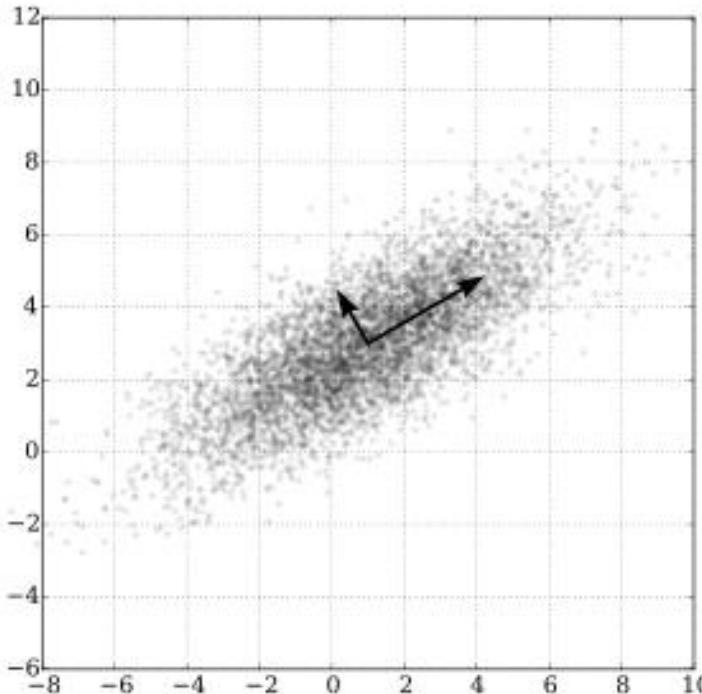
\*\*. 상관계수는 0.01 수준(양쪽)에서 유의합니다.

### - 상관계수 분석(Correlation Analysis)

변수 간 상관관계가 크면 다중 공전성 존재  
이는 Overfitting 등의 문제점 초래

VIF(Variance Inflation Factor)값을 이용하거나  
Ridge Regression 등을 통해 유의한 변수만을  
선택.

## 4. Feature Engineering



- PCA(주성분 분석)

변수간 상관관계가 클 경우 고차원의 데이터를 저차원으로 축소

변수를 선택한다기보다는 기존 변수들을 이용하여  
새로운 특징을 더 잘 나타내는 새로운 변수들 생  
성

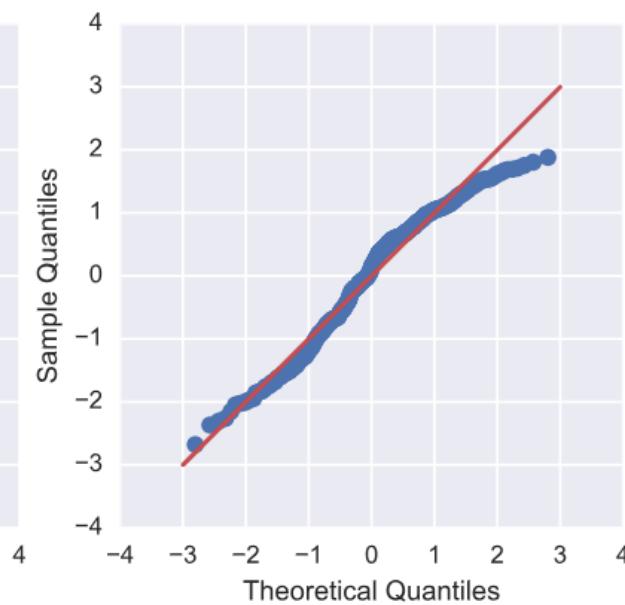
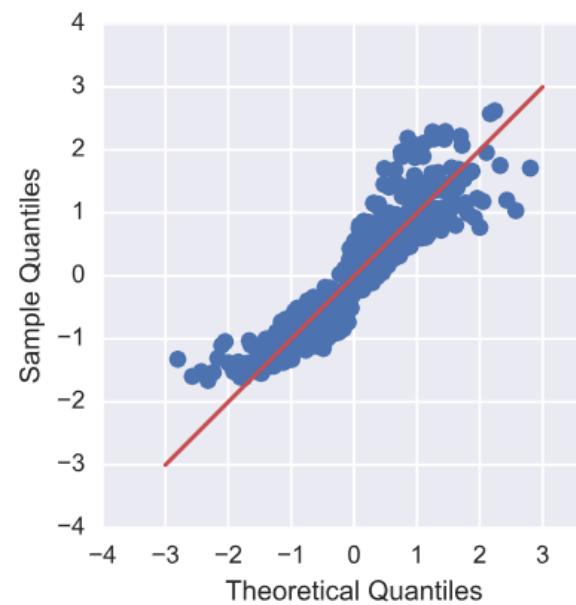
변수 해석의 문제가 있음.

## 4. Feature Engineering

### Scaling

#### 1) Cox box Muller transformation

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln y & \text{if } \lambda = 0, \end{cases}$$



## 4. Feature Engineering

Scaling

2) Min-Max

$$(x - \min(x)) / (\max(x) - \min(x))$$

3) Log

$\log(x, 2)$  or  $\log(x, 10)$  ...

4) 표준화

$$x - \text{mean}(x) / (\text{sd}(x))$$

## 4. Feature Engineering

### Binning

연속형 변수를 범주형 변수로 만드는 방법  
특별한 원칙이 있는 것은 아니다. (성능이 좋으면 된다)  
앞서 단변량 분석에서 Histogram을 보고 파악할 수도 있다.

2	2	0 00
5	3	1 058
15	10	2 1333458889
22	7	3 0355789
(11)	11	4 11133456678
32	17	5 1112223334445688
15	6	6 147779
9	5	7 33478
4	2	8 29
2	2	9 09

65v

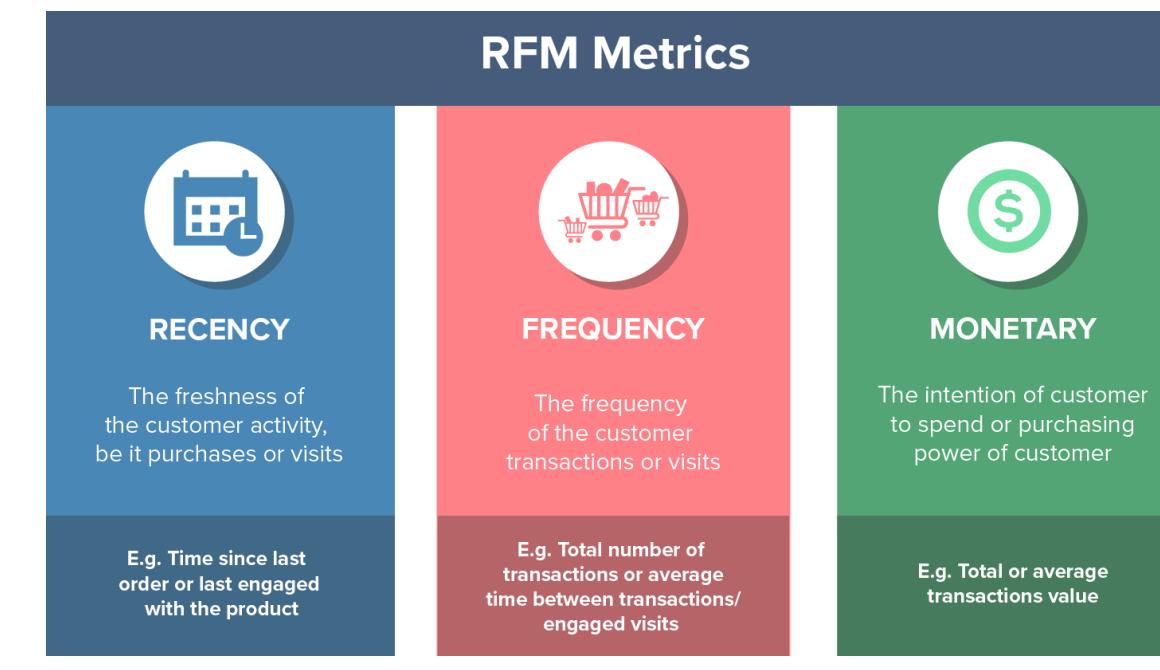
2	2	0* 00
0	.	0. 0
1	*	1* 58
2	.	2. 13334
3	*	2. 58889
3	*	3* 03
3	.	3. 55789
4	*	4* 111334
4	.	4. 56678
5	*	5* 111222333444
5	.	5. 56688
6	*	6* 14
6	.	6. 7779
7	*	7* 334
7	.	7. 78
8	*	8* 2
8	.	8. 9
9	*	9* 0
9	.	9. 9

65v

## 4. Feature Engineering

### 파생변수

변수 하나에서 다른 변수를 만들어 내는 것



Ex) RFM

거래의 최근성+빈도+규모를 이용,  
고객 분석 변수 생성.

# PART.III 모델링

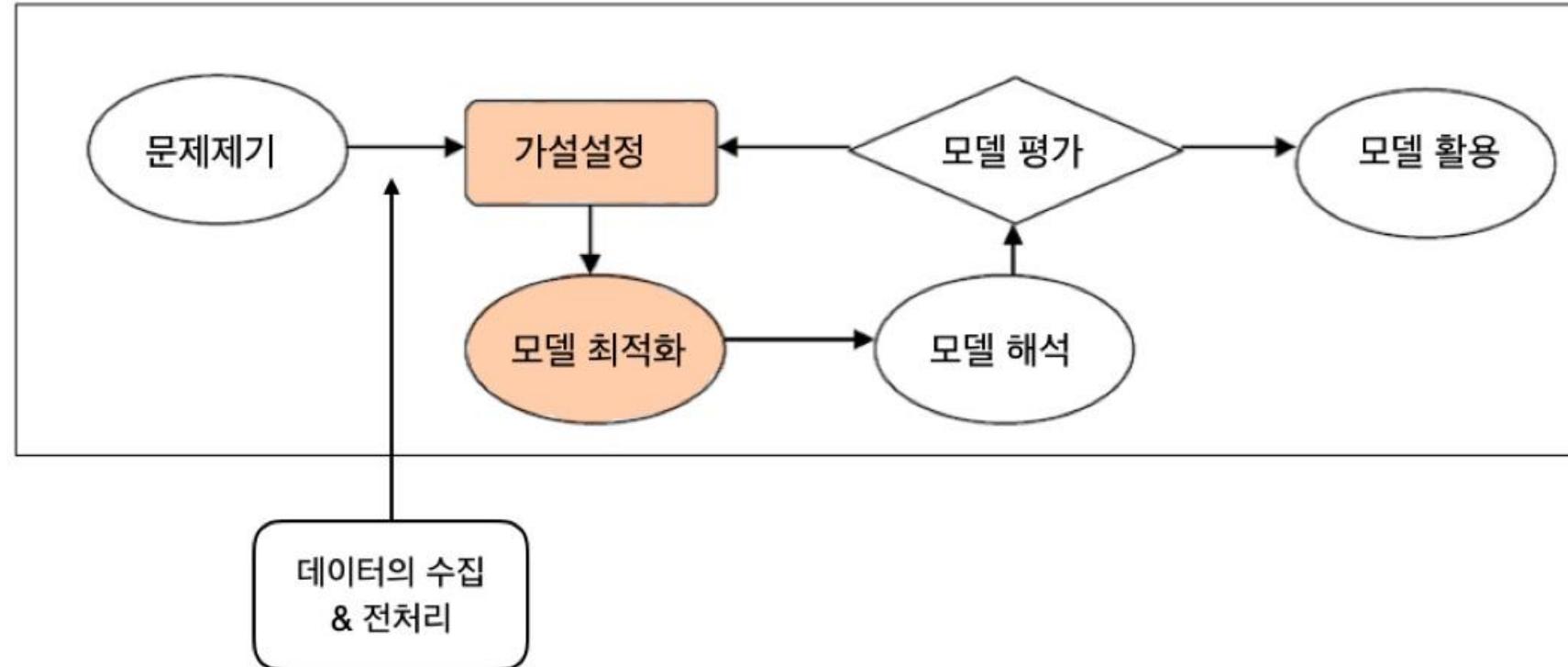
1

**Cost Function  
&  
Optimization**

2

**Gradient Decent  
Algorithm**

# 1. Cost Function & Optimization



# 1. Cost Function & Optimization

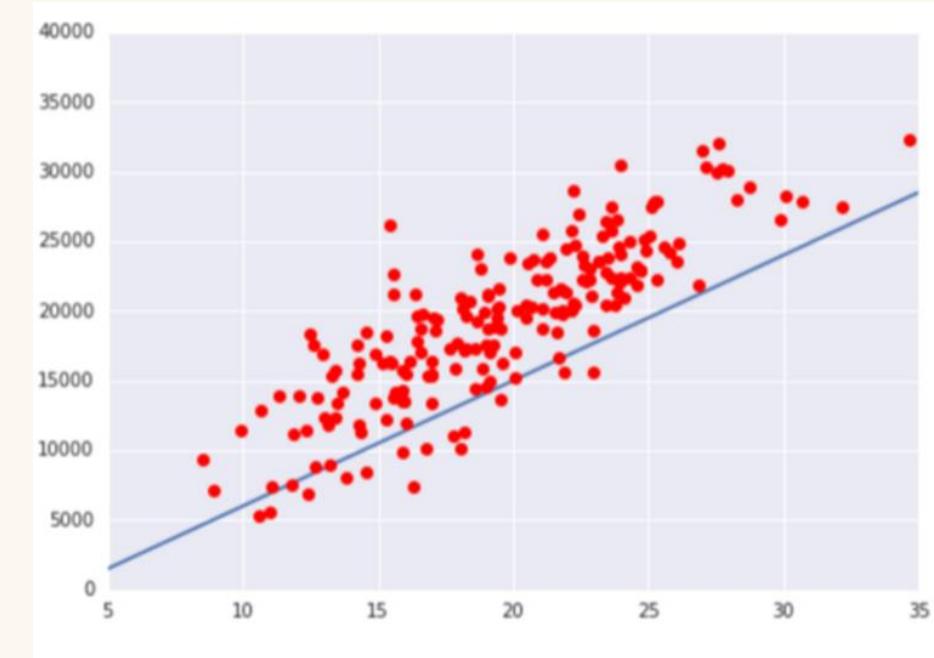
## Ex) Linear Regression

[가설설정]

데이터를 설명할 수 있는

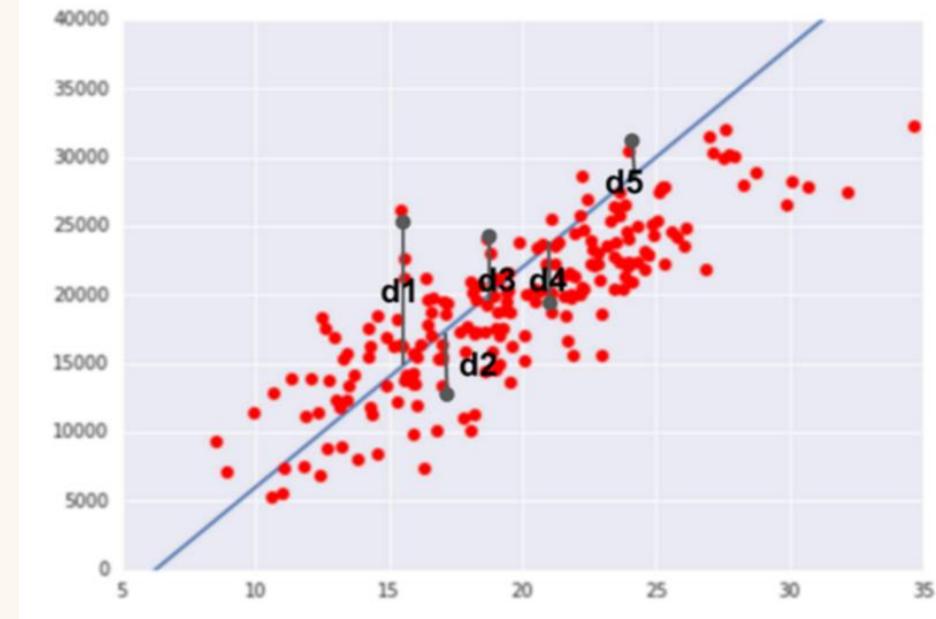
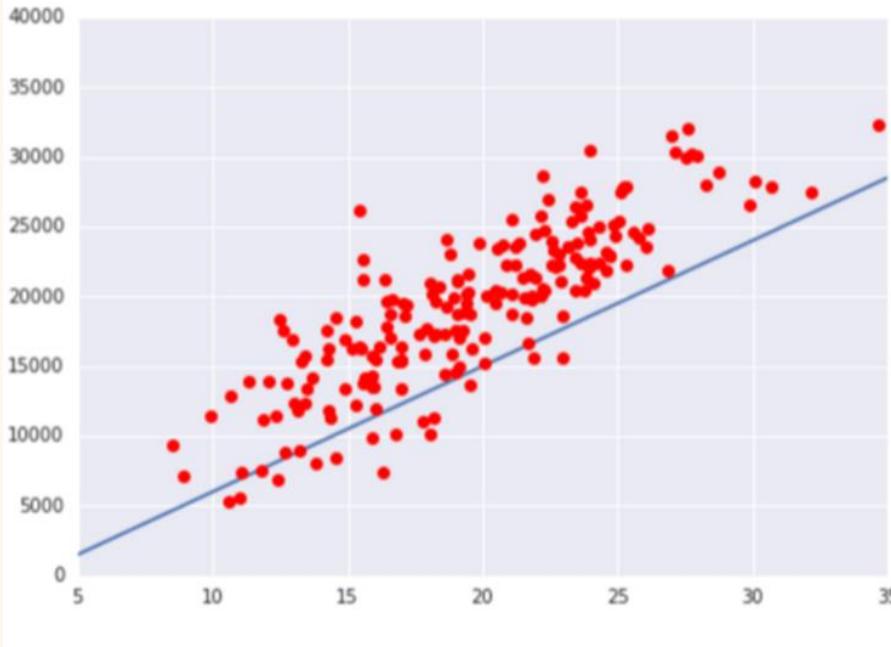
“직선”이 존재한다

$$\rightarrow H(x) = Wx + b$$



# 1. Cost Function & Optimization

## Ex) Linear Regression – Cost Function



$$\text{Error} = H(x^{(i)}) - y^{(i)} \quad \text{Cost} = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2$$

# 1. Cost Function & Optimization

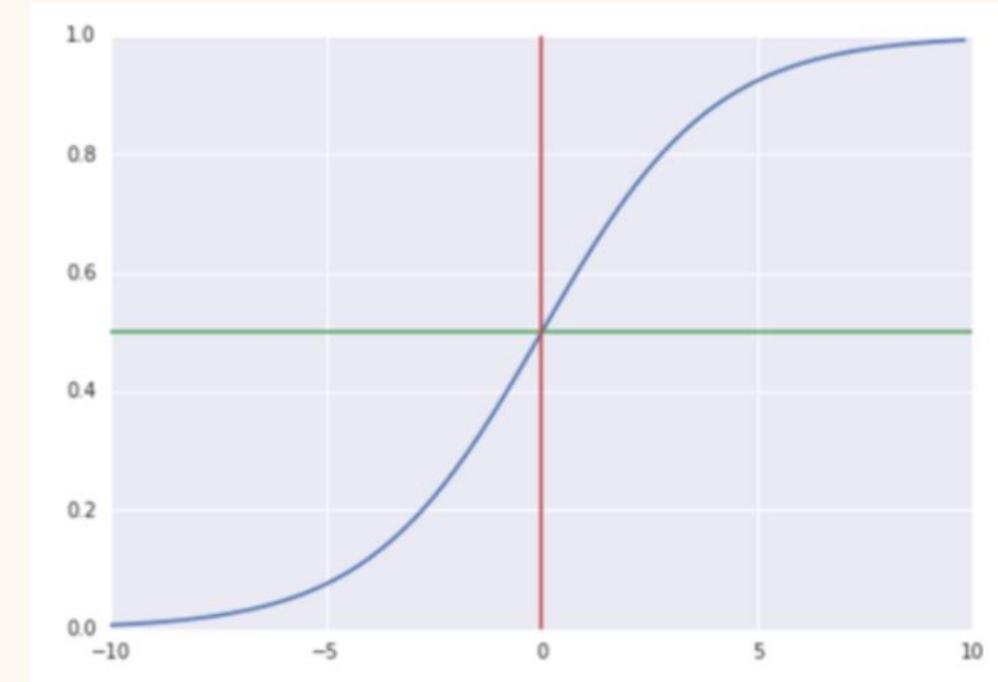
## Ex) Logistic Regression

[가설함수]

$$H(x) = \frac{1}{1 + e^{-x}} \\ = \text{sigmoid}(Wx + b)$$

$Wx + b < 0 \circ]$  면  $H(x) = 0$

$Wx + b > 0 \circ]$  면  $H(x) = 1$



# 1. Cost Function & Optimization

## Ex) Logistic Regression – Cost Function

### Linear Regression

$$H(\alpha) = w\alpha + b$$

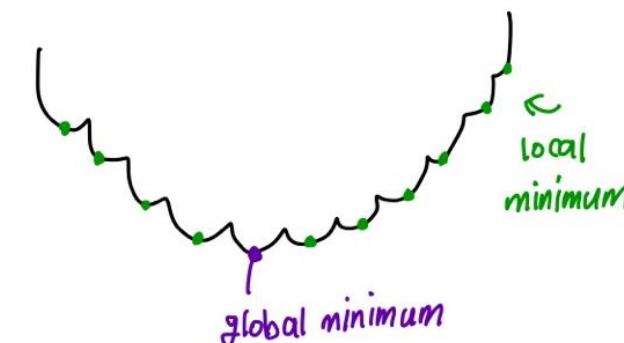


### Logistic Regression

$$\begin{aligned} H(\alpha) &= \text{sigmoid} (w^T x) \\ &= \frac{1}{1 + e^{-w^T x}} \quad (\text{odds 사이의 확률}) \end{aligned}$$

$$\text{if } \text{cost} = \frac{1}{m} \sum (H(\alpha) - y)^2$$

then



# 1. Cost Function & Optimization

## Ex) Logistic Regression – Cost Function

### Linear Regression

$$H(d) = Wd + b$$



### Logistic Regression

$$H(d) = \frac{1}{1 + e^{-(W^T X)}}$$



# 1. Cost Function & Optimization

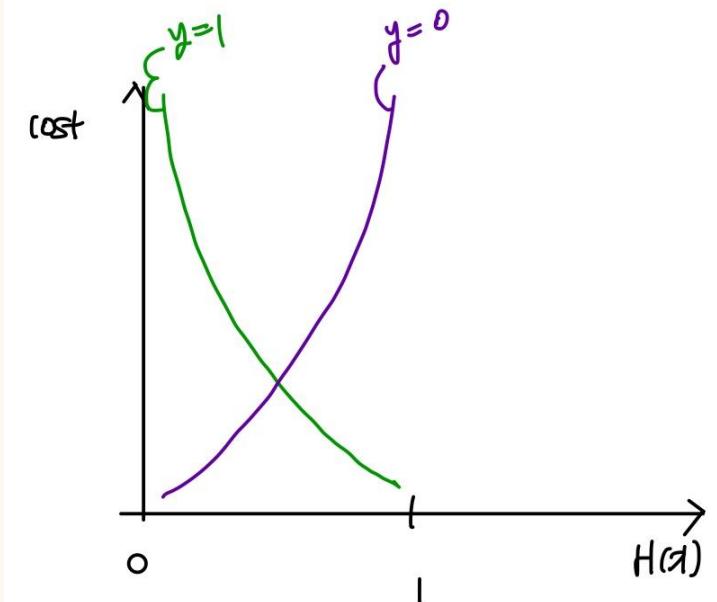
## Ex) Logistic Regression – Cost Function

$$\text{cost}(W) = \frac{1}{m} \sum c(H(x), y)$$

$$c(H(x), y) = \begin{cases} -\log(H(x)) & : y = 1 \\ -\log(1 - H(x)) & : y = 0 \end{cases}$$

$$C(H(x), y) = -y \log(H(x)) - (1 - y) \log(1 - H(x))$$

Cross Entropy



# 1. Cost Function & Optimization

## Cf) Optimization

어떠한 제약조건이 있고, 이 조건 하에서 목적함수를 만족시키는 '최적해(변수의 값)'을 찾는 것.

### [Objective Function(목적함수), Linear Regression]

$$\text{Minimize } cost = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2 , \quad H(x) = Wx + b$$

### [Subject to(제약조건)]

$W, b$ 가 실수

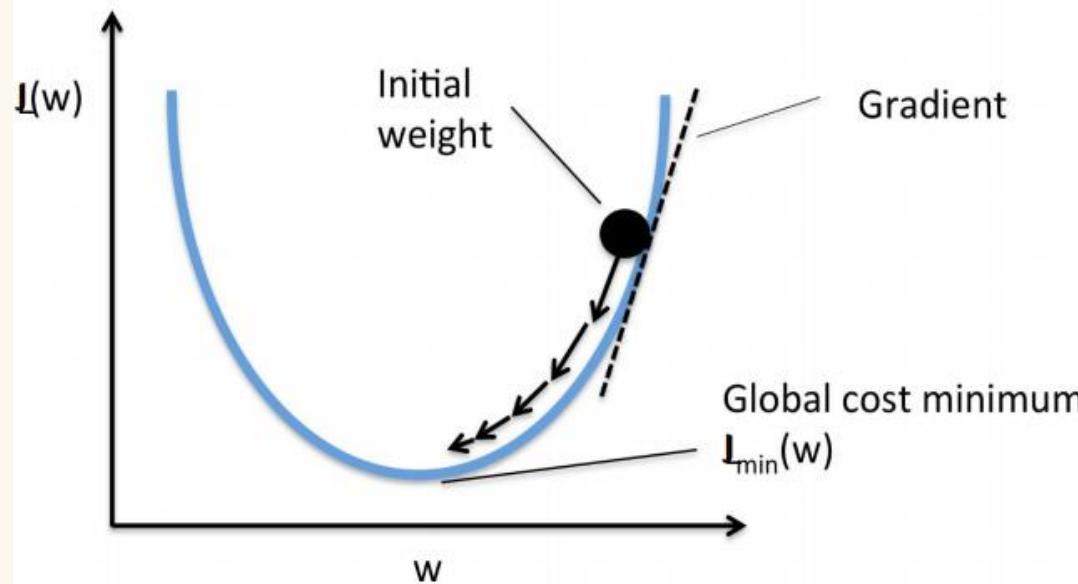


→ 이를 만족시키는  $W$ 와  $b$  (최적해, Optimal Solution) 를 찾고자 하는 것

Gradient  
Decent  
Algorithm

## 2. Optimization Algorithm: Gradient Decent Algorithm

[모델 최적화, Linear Regression]



$$W := W - \alpha \frac{\partial}{\partial W} \text{cost}(W)$$

$\alpha = \text{learning rate.}$

$$W := W - \alpha \frac{1}{2m} \sum_{i=1}^m 2(Wx^{(i)} - y^{(i)})x^{(i)}$$

## 2. Optimization Algorithm: Gradient Decent Algorithm

[모델 최적화, Logistic Regression]

$$C : (H(x), y) = -y \ln(H(x)) - (1 - y) \ln(1 - H(x))$$

$$W := W - \alpha \frac{\partial}{\partial W} cost, \quad \alpha = \text{learning rate.}$$



# PART.IV 모델평가

1

Bias, Variance

2

Under/Overfitting

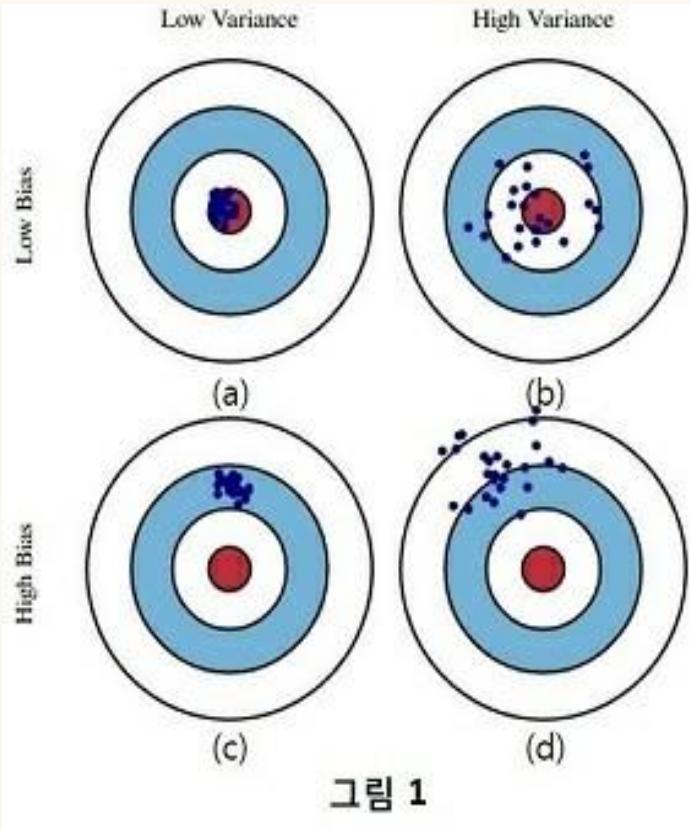
3

Validation

4

Evaluation Metric

# 1. Bias-Variance

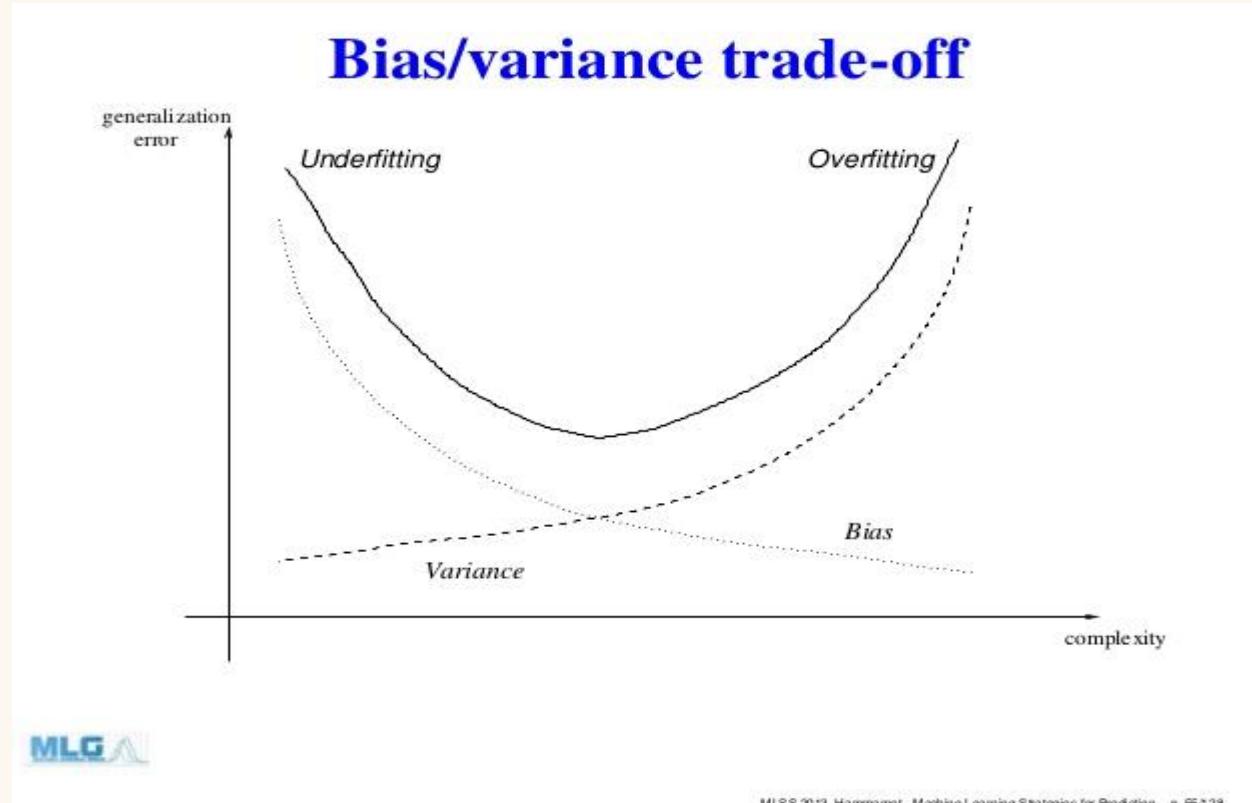


- Bias : 참값과 예측값의 차이

- Variance : 예측값의 산포도

작은 Bias와 Variance를 가진 예측값이 좋다

# 1. Bias-Variance Tradeoff

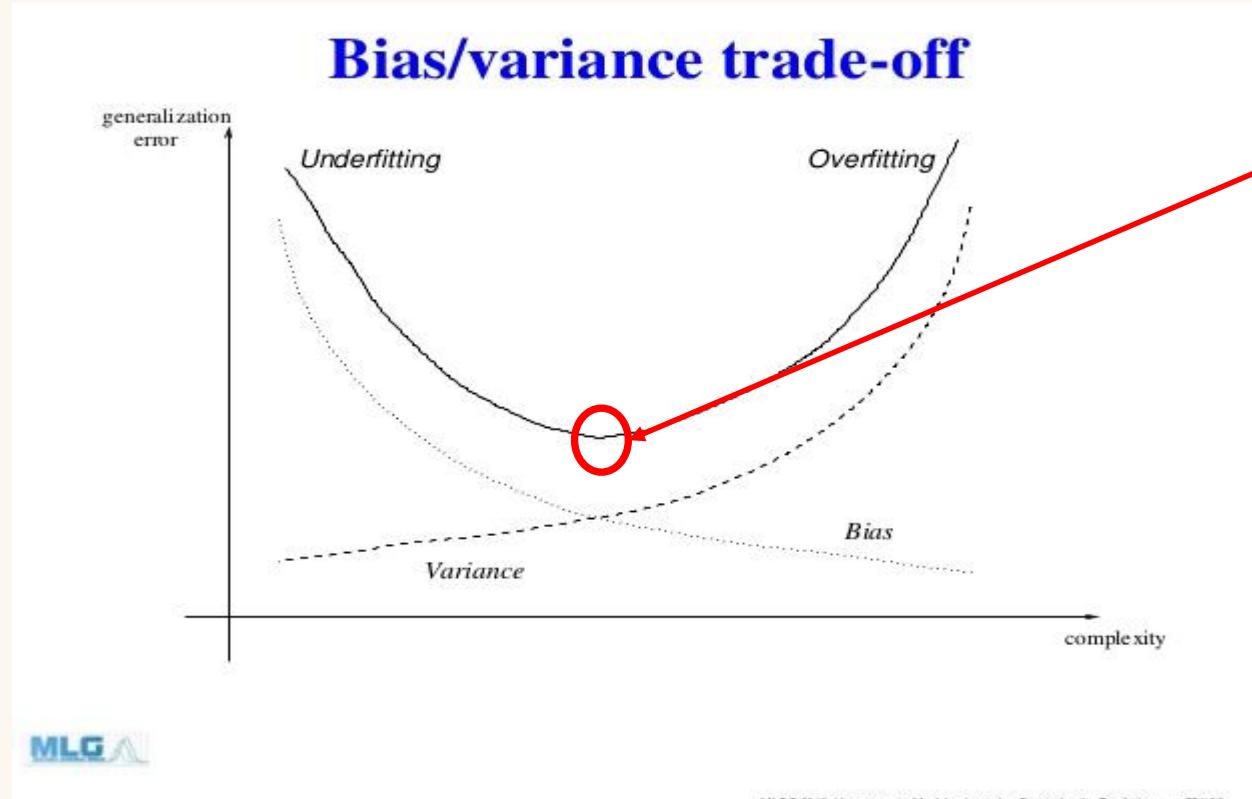


- Bias : 모델이 복잡해질수록 감소

- Variance : 모델이 복잡해질수록 증가

Optimal Point를 찾아야함!

# 1. Bias-Variance Tradeoff

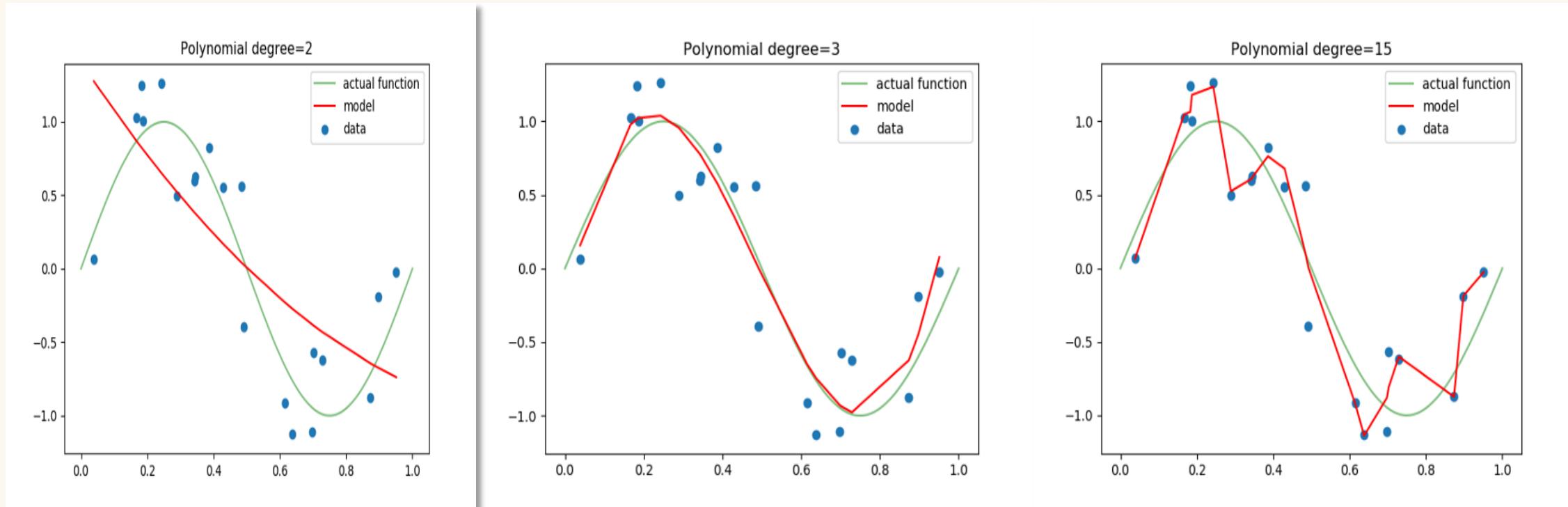


일반화 오류가 최소화되는 지점

Optimal point에 근접한 모델을 산출

Optimal model을 찾는 출발점

## 2. Underfitting / Overfitting



<Underfitting>

<Desirable fit>

<Overfitting>

## 2. Desirable fit : somewhere between under/overfitting

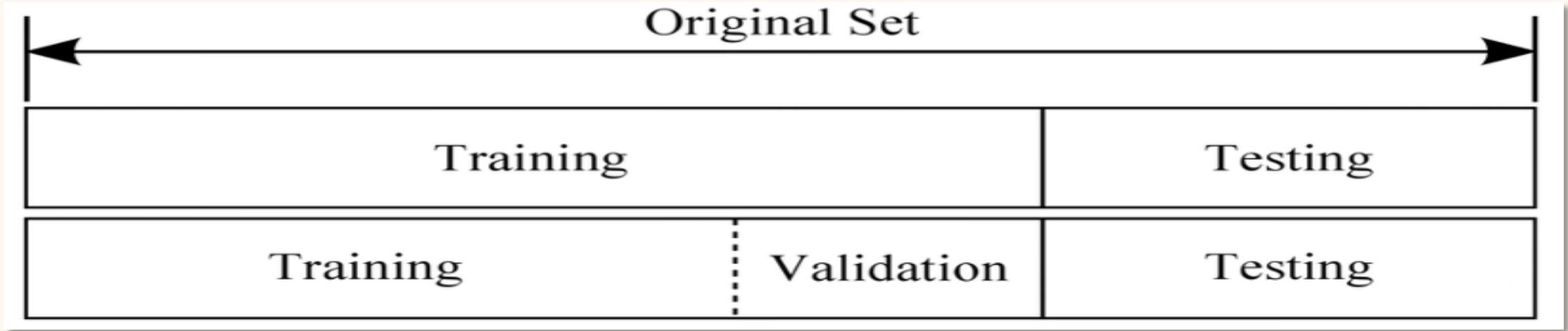


모델은 일반화 가능하도록 **최대한 간결**해야함

모델은 주어진 데이터를 **충분히 설명할만큼 복잡**해야함

어떻게 모델의 적합을 조율할 수 있을까?

### 3. Cross - Validation



Train Set 과 Test Set의 **완전한 분리**가 핵심!

충분한 data를 수집했다면, 추가적으로 Validation set을 활용하여 중간점검을 해 볼 수 있음

### 3. Cross - Validation



#### Underfitting 1

Train Performance : poor  
Test Performance : poor

#### Overfitting

Train Performance : good  
Test Performance : poor

#### Underfitting 2

Train Performance : poor  
Test Performance : good

#### Desirable Fit

Train Performance : good  
Test Performance : good

### 3. Cross - Validation



어떻게 Model Performance에  
접근할 수 있을까?

## 4. Evaluation Metric



어떻게 Model을 정량적으로  
평가할 수 있을까?

## 4. Evaluation Metric

### Classification

Accuracy  
Precision-Recall  
ROC-AUC

### Regression

MSE  
RMSE  
R2 Score  
Adjusted R2

## 4. Evaluation Metric - Classification

<Confusion Matrix>

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	$a/(a+b)$
	Negative	c	d	Negative Predictive Value	$d/(c+d)$
		Sensitivity	Specificity	$\text{Accuracy} = (a+d)/(a+b+c+d)$	
		$a/(a+c)$	$d/(b+d)$		

모델이 예측한 값과 참값을 table의 형식으로 작성한 행렬

Classification model에서의 Evaluation metric은 이에 기반

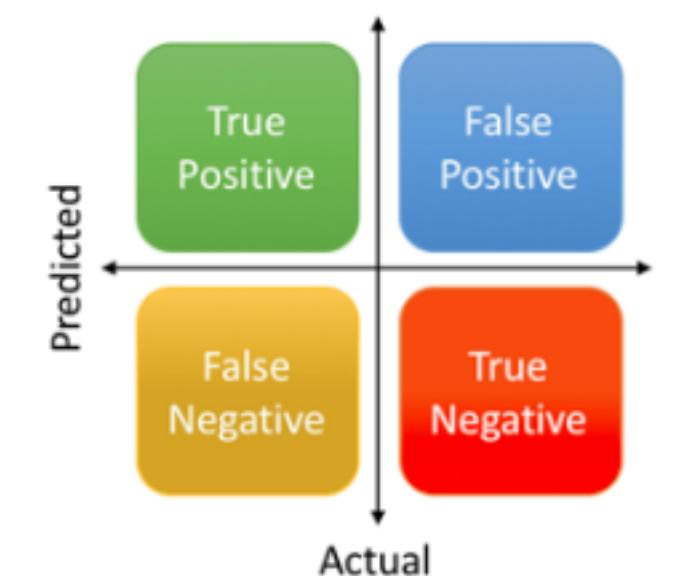
## 4. Evaluation Metric - Classification

<Accuracy, Precision, Recall>

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total}}$$



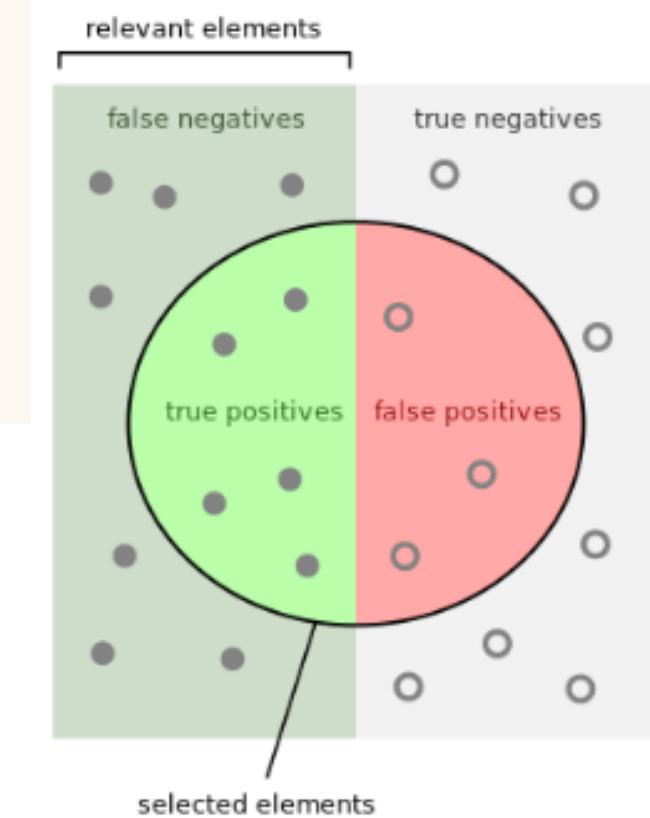
## 4. Evaluation Metric - Classification

<Accuracy, Precision, Recall>

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total}}$$



How many selected items are relevant?

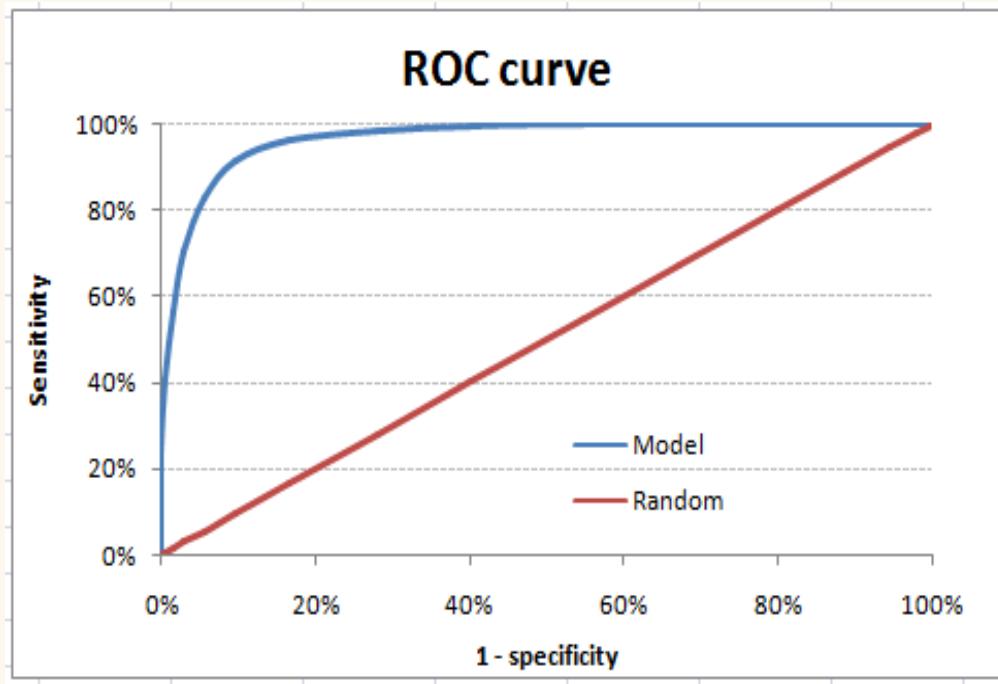
$$\text{Precision} = \frac{\text{green}}{\text{red} + \text{green}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{green}}{\text{green} + \text{light grey}}$$

## 4. Evaluation Metric - Classification

<ROC curve & AUC>



$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

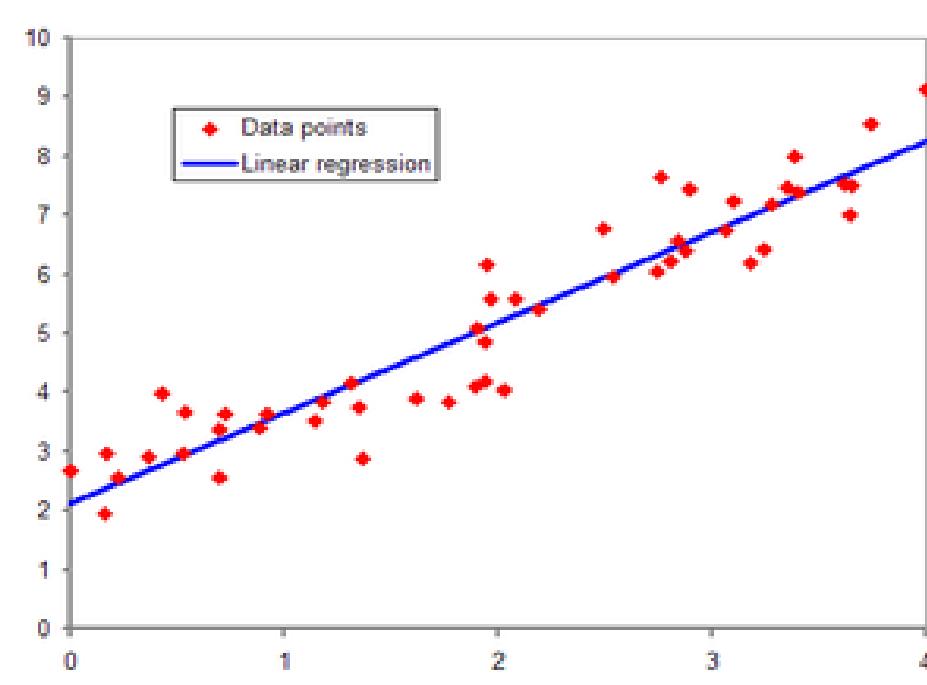
- ROC curve :  
가능한 Cutoff에서의 Sensitivity와  
(1-Specificity)의 그래프

- Sensitivity  
= Recall, True Positive Rate

- AUC :  
ROC curve 아래의 영역 넓이  
모델의 performance로 해석

## 4. Evaluation Metric – Regression

<Mean Squared Error(MSE)>



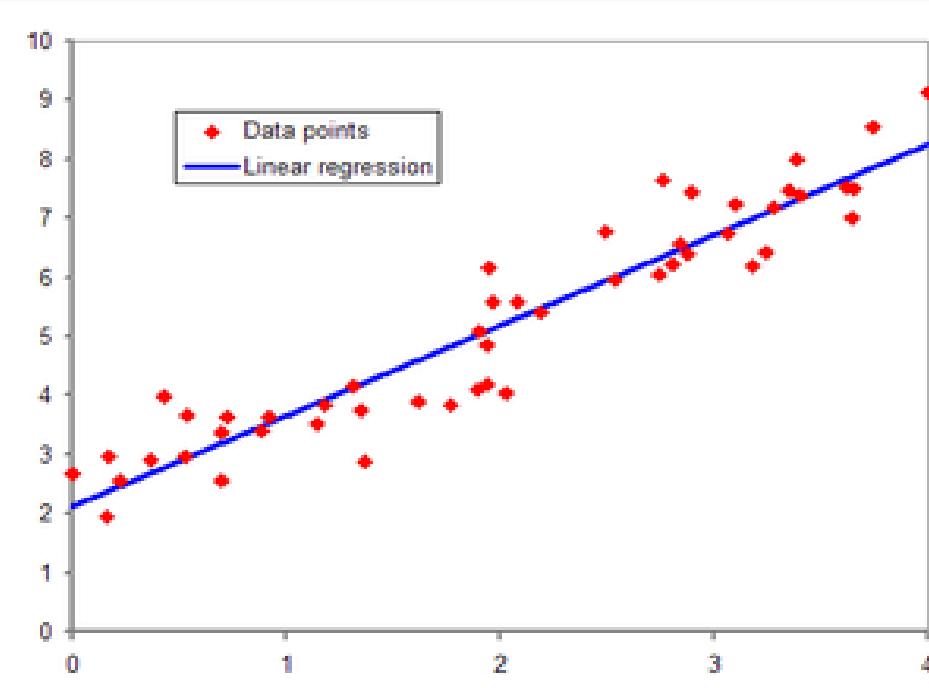
$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2$$

where  $N$  is the number of data points,  
 $f_i$  the value returned by the model and  
 $y_i$  the actual value for data point  $i$ .

모델의 성능을 평가하는 정량적인 지표로 활용

## 4. Evaluation Metric – Regression

<Root Mean Squared Error(RMSE)>



$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predicted_i - Actual_i)^2}{N}}$$

모델의 성능을 평가하는 정량적인 지표로 활용

## 4. Evaluation Metric – Regression

<R2 Score>

### ○ R2 SCORE

#### BAD MODEL

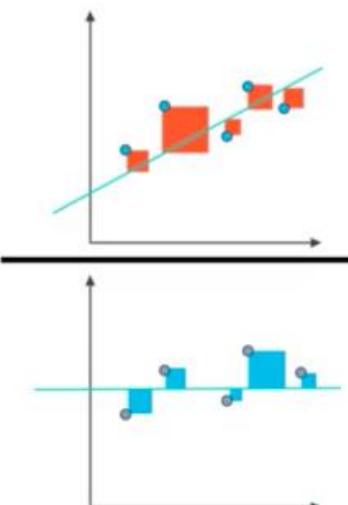
The errors should be similar.  
R2 score should be close to 0.

#### GOOD MODEL

The mean squared error for the linear regression model should be a lot smaller than the mean squared error for the simple model.

R2 score should be close to 1.

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$



$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

모델의 성능을 평가하는 정량적인 지표로 활용

## 4. Evaluation Metric – Regression

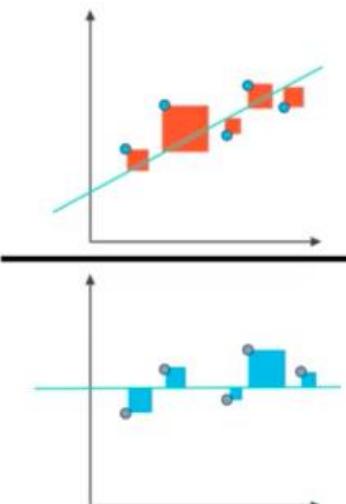
<Adjusted R2 Score>

○ R2 SCORE

**BAD MODEL**  
The errors should be similar.  
R2 score should be close to 0.

**GOOD MODEL**  
The mean squared error for the linear regression model should be a lot smaller than the mean squared error for the simple model.  
R2 score should be close to 1.

$$R^2 = 1 -$$



$$R^2 = 1 - \frac{SS_{residuals}}{SS_{total}}$$

$$\text{Adjusted } R^2 = 1 - \frac{\frac{SS_{residuals}}{(n - K)}}{\frac{SS_{total}}{(n - 1)}}$$

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

모델의 성능을 평가하는 정량적인 지표로 활용

# PART.V Validation

1

Validation 이란

2

Validation set  
approach

3

LOOCV  
(Leave one-out cv)

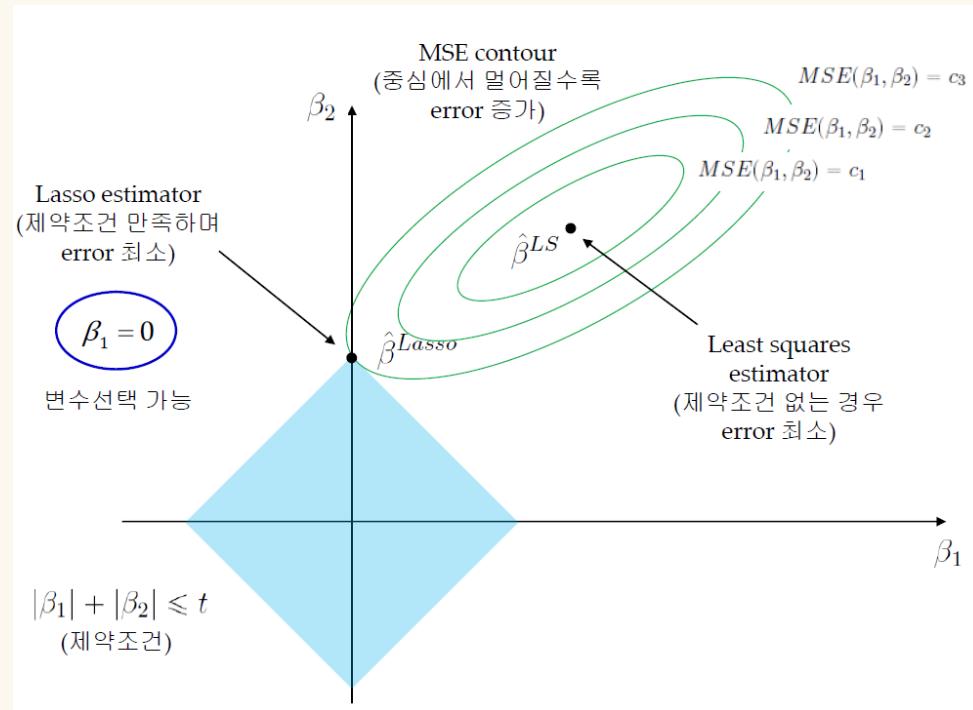
4

K-fold cv

# 1. Validation

참고자료: LASSO(Least Absolute shrinkage and selection)

$$\hat{\beta}^{Lasso} = \min(\beta) \{ (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda \|\beta\|_1 \}$$



- L1 norm을 제한하는 기법

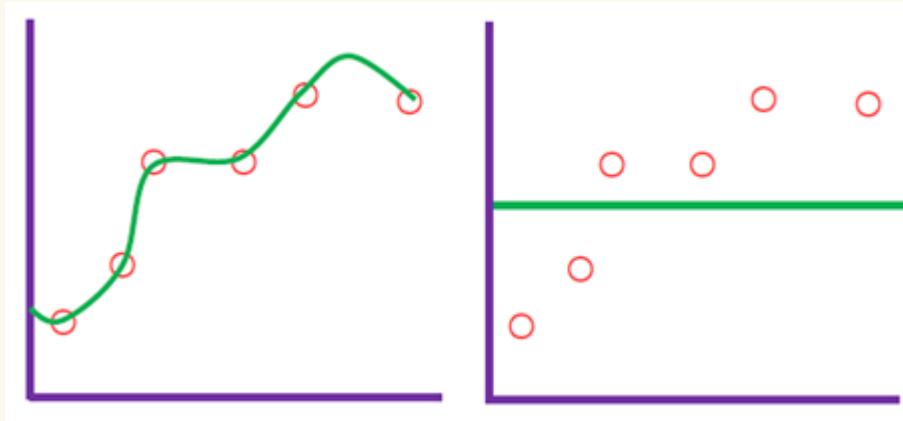
$$\begin{aligned} L_1 &= \left( \sum_i^n |x_i| \right) \\ &= |x_1| + |x_2| + |x_3| + \dots + |x_n| \end{aligned}$$

- Shrinkage and selection(overfitting 방지)

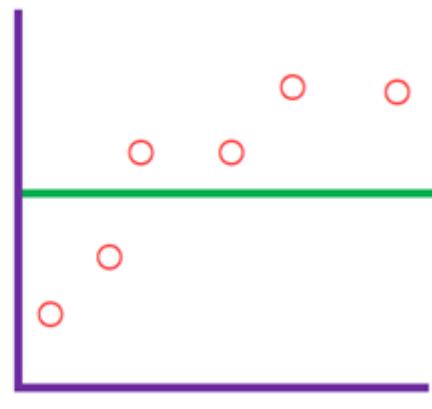
- 적절한  $\lambda$  값 선택이 crucial

# 1. Validation

Too small  $\lambda$



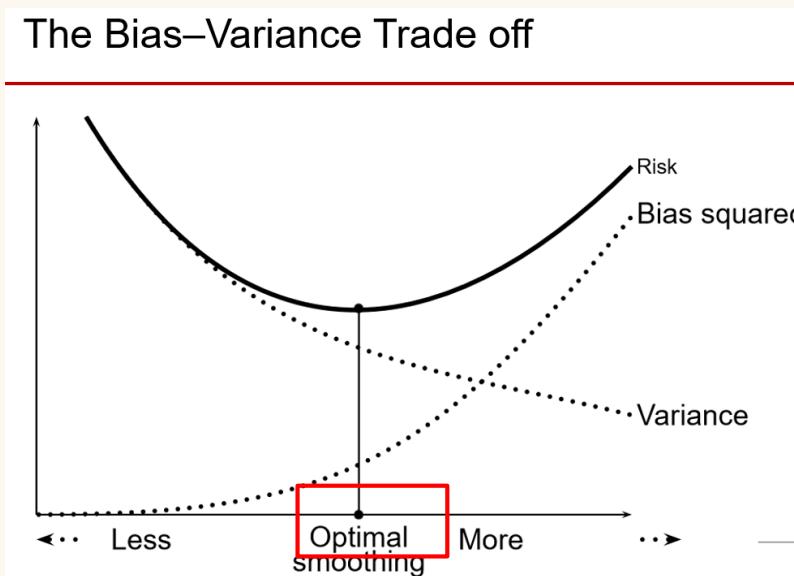
Too large  $\lambda$



- 최적의  $\lambda$  찾는 법? Validation ! (검증) : “검정”과 다름

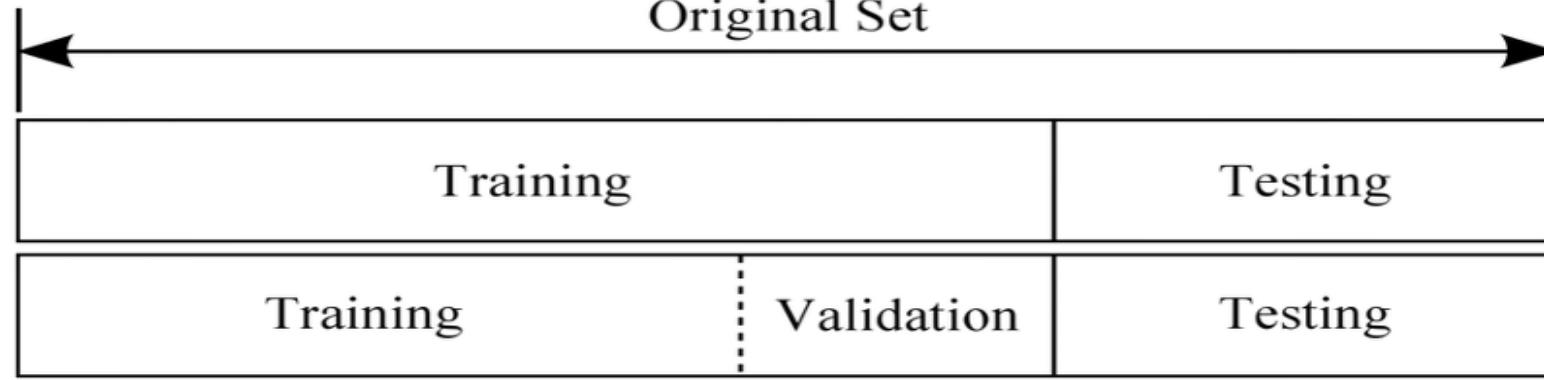
# 1. Validation의 목적

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

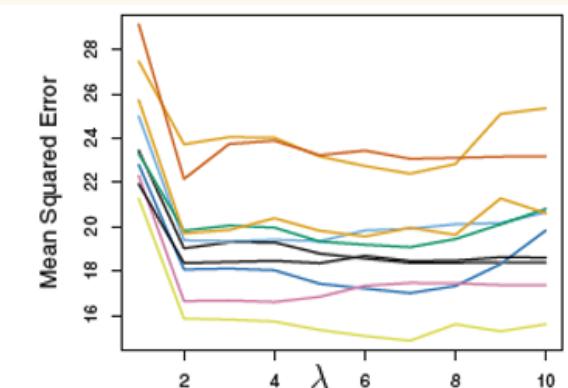
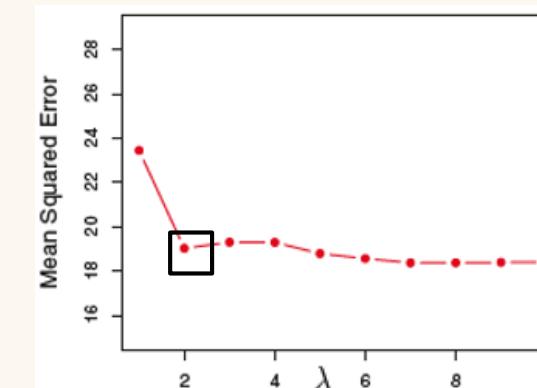


- Test error 추정
- 적절한 Smoothing parameter( $\lambda$ ) 정 할 때
- 1) Validation set 2) LOOCV 3) k-fold cv

## 2. Validation set approach(단일 Validation set)

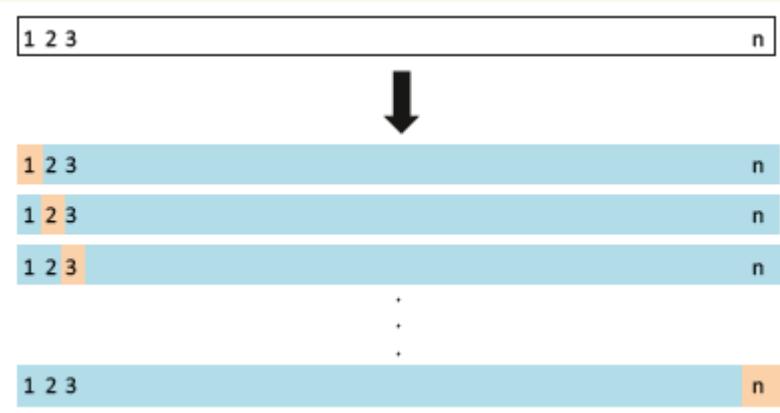


- 70% train set(30% validation set), 30% test set
- 랜덤성에 따라 다양한 MSE 추정량 : 신뢰 ↓
- Test error를 Over-estimate 하는 경향 존재



### 3. LOOCV(Leave-one-out Cross validation)

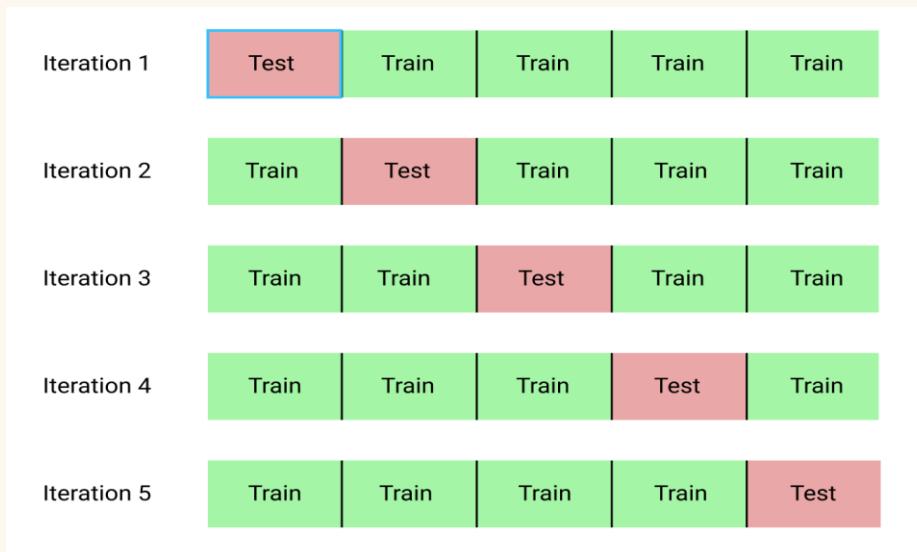
$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n MSE_i.$$



- 훨씬 적은 bias(Validation set approach에 비해)
- 변하지 않는 validation error : 신뢰 ↑

## 4. k-fold cv

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k MSE_i.$$



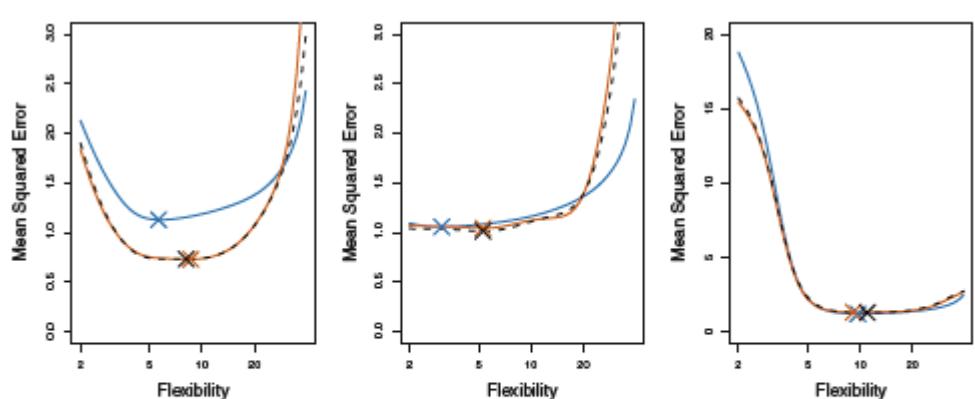
- Validation에 드는 시간 절감

- bias, variance trade-off : 5-fold CV > LOOCV

- EX) 100개면 LOOCV: 99개, 5-fold: 80개

## 4. 10-fold cv V.S LOOCV

파란색 선: True test MSE, 검은색 선: LOOCV estimate for MSE ,  
오렌지색 선 : 10-fold CV estimate for MSE

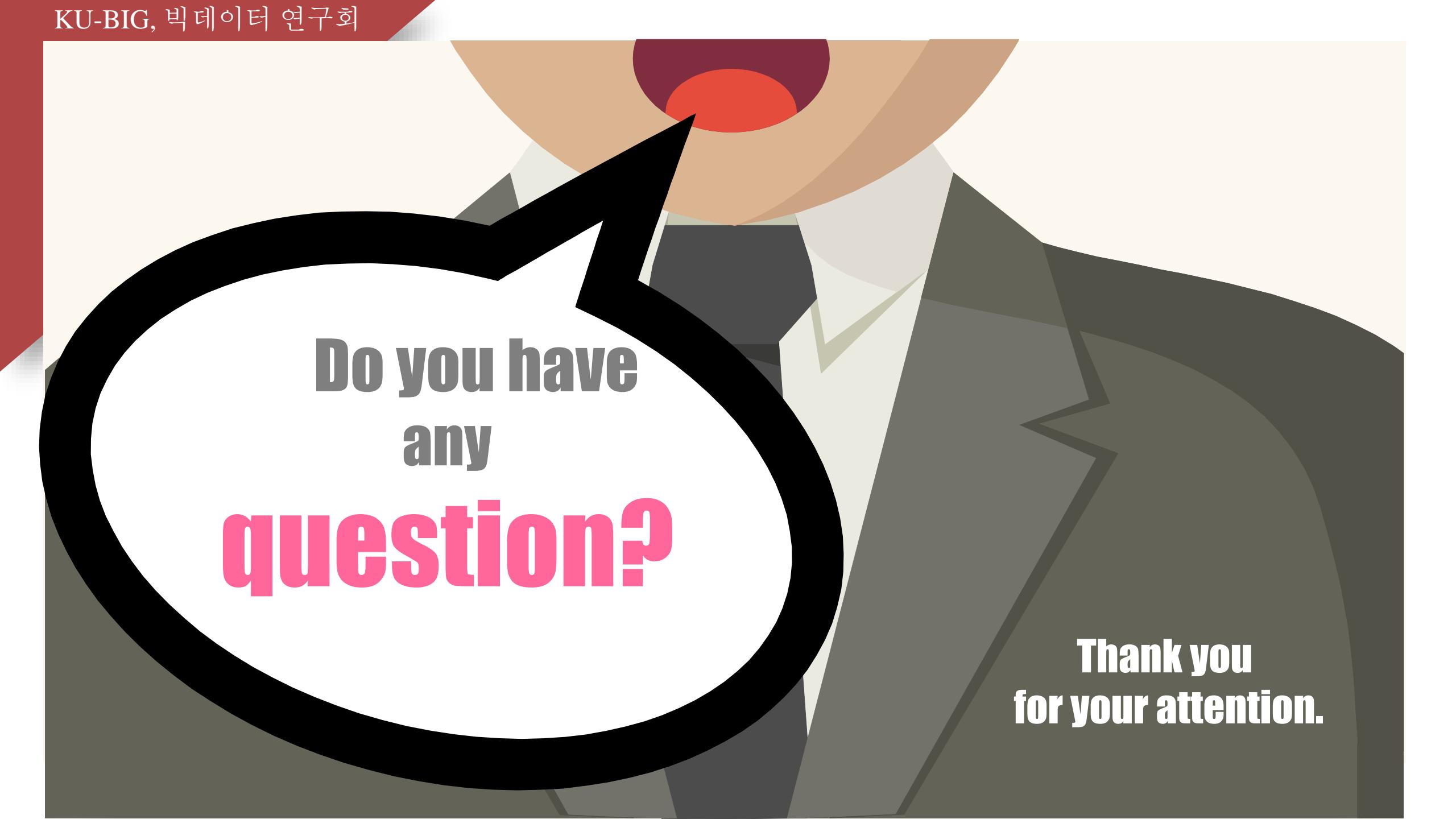


**FIGURE 5.6.** True and estimated test MSE for the simulated data sets in Figures 2.9 (left), 2.10 (center), and 2.11 (right). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.

- 마땅한 test set이 없을 때: Estimated test error
- Smoothing parameter 정할 때: Estimated test error 가 최소가 되는 ‘지점’ ★ (잘 된 cross-validation)
- Classification은 MSE 대신 오분류율 기준

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n Err_i,$$

where  $Err_i = I(y_i \neq \hat{y}_i)$



Do you have  
any  
**question?**

Thank you  
for your attention.