



R을 활용한 머신 러닝 기초

알고리즘을 학습하기 전에 알아야할 기초개념

김영석 최지은



CONTENT

1. 머신러닝 개요

- 1-1. 머신러닝 용어
- 1-2. 연속형 / 범주형
종속변수
- 1-3. 데이터 분석 단계

2. Pre-Processing

- 2-1. 범주형 변수 처리
- 2-2. 이상치, 결측치, 정규화

3. EDA / 시각화

- 3-1. 기본차트
- 3-2. 분포차트
- 3-3. 차트조절

4. 데이터 분할

- 4-1. 학습/검증/평가 데이터
- 4-2. 교차검증

5. 파라미터 튜닝

- 5-1. 파라미터란?
- 5-2. Random Search
- 5-3. Grid Search

6. 모델 평가

- 6-1. 연속형 종속변수
- 6-2. 이진형 종속변수
- 6-3. 과적합

01 머신러닝이란?

1. 인공지능 (Artificial Intelligence)

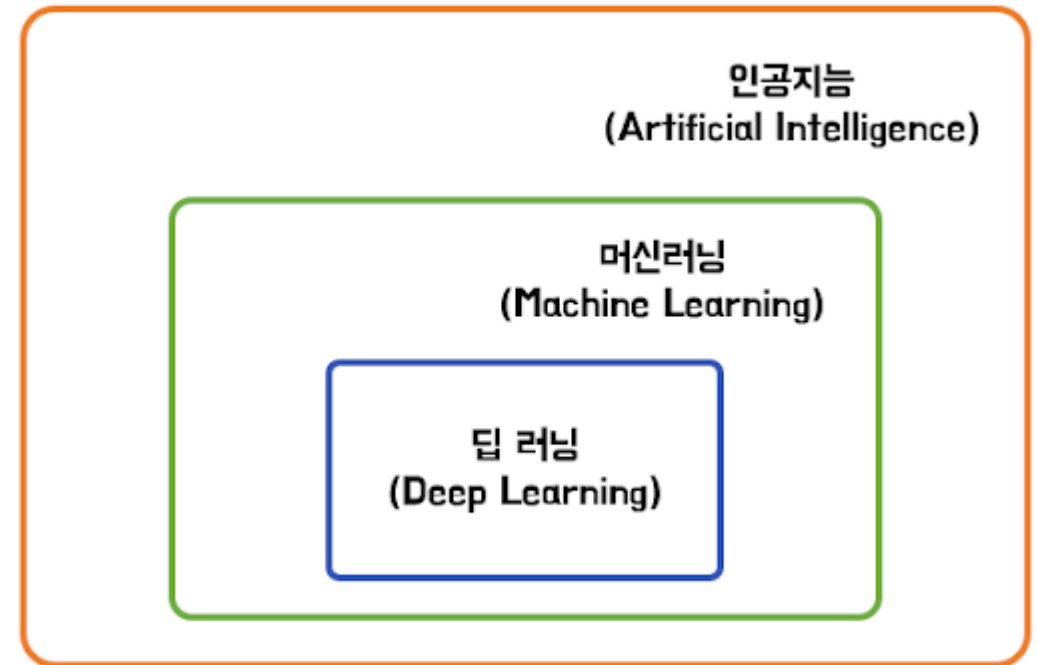
- 인간의 학습능력, 추론능력 등을 컴퓨터를 통해 구현하는 포괄적인 개념
- 알파고, 사진을 보고 사물을 판단하는 소프트웨어, 영화 AI에 나오는 로봇 등등

2. 머신러닝 (Machine Learning)

- 데이터를 이용하여 데이터 특성과 패턴을 학습하여, 그 결과를 바탕으로 미지의 데이터에 대한 그것의 미래 결과(값, 분포)를 예측

3. 딥러닝

- 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 기계학습 알고리즘의 집합
- 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야



01 머신러닝 이란?



머신러닝을 통한 개와 고양이 분류



딥러닝을 통한 개와 고양이 분류

1. 데이터의 특징을 사람이 추출하지 않음
2. 주로 인공신경망 구조를 사용하여 학습

01

머신러닝 개요

1-1. 머신러닝 용어

1-2. 연속형 / 범주형
종속변수

1-3. 데이터 분석 단계

01 머신러닝 용어

I. Target and feature

Predictor variables (예측 변수)

Input variables (입력 변수)

Independent variables (독립 변수)

Target variables (타겟 변수)

Output variables (출력 변수)

Dependent variables (종속 변수)

id	X_1	X_2	...	X_p	Y
1	x_{11}	x_{12}	...	$x_{1,p}$	y_1
2	x_{21}	x_{22}	...	$x_{2,p}$	y_2
...
n	$x_{n,1}$	$x_{n,2}$...	$x_{n,p}$	y_n

01 머신러닝 용어

II. Index(id)

id	X_1	X_2	...	X_p	Y
1	x_{11}	x_{12}	...	$x_{1,p}$	y_1
2	x_{21}	x_{22}	...	$x_{2,p}$	y_2
...
n	$x_{n,1}$	$x_{n,2}$...	$x_{n,p}$	y_n

Point (포인트)
 Sample (샘플)
 Instance (인스턴스)
 Record (레코드)
 Observation (관측치)
 Vector (벡터)

01 머신러닝 용어

III . Loss(Cost) function and Model Parameter

- 손실 함수 (Loss function), 비용 함수 (Cost function), 오차 함수 (Error function), 목적 함수 (Objective function) 다양한 이름을 가진다.

$$Loss = f(y, \hat{y}) = f(y, g(x, \theta))$$

Diagram illustrating the components of the loss function equation:

- f is labeled as the **Loss function**.
- y is labeled as the **Target**.
- \hat{y} is labeled as the **Projection**.
- g is labeled as the **Machine model**.
- θ is labeled as the **Model Parameter**.

01 머신러닝 개요

머신러닝 용어

III . Loss(Cost) function and Model Parameter

- Loss function의 종류

Mean squared error

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^n e_t^2$$

Root mean squared error

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$$

Mean absolute error

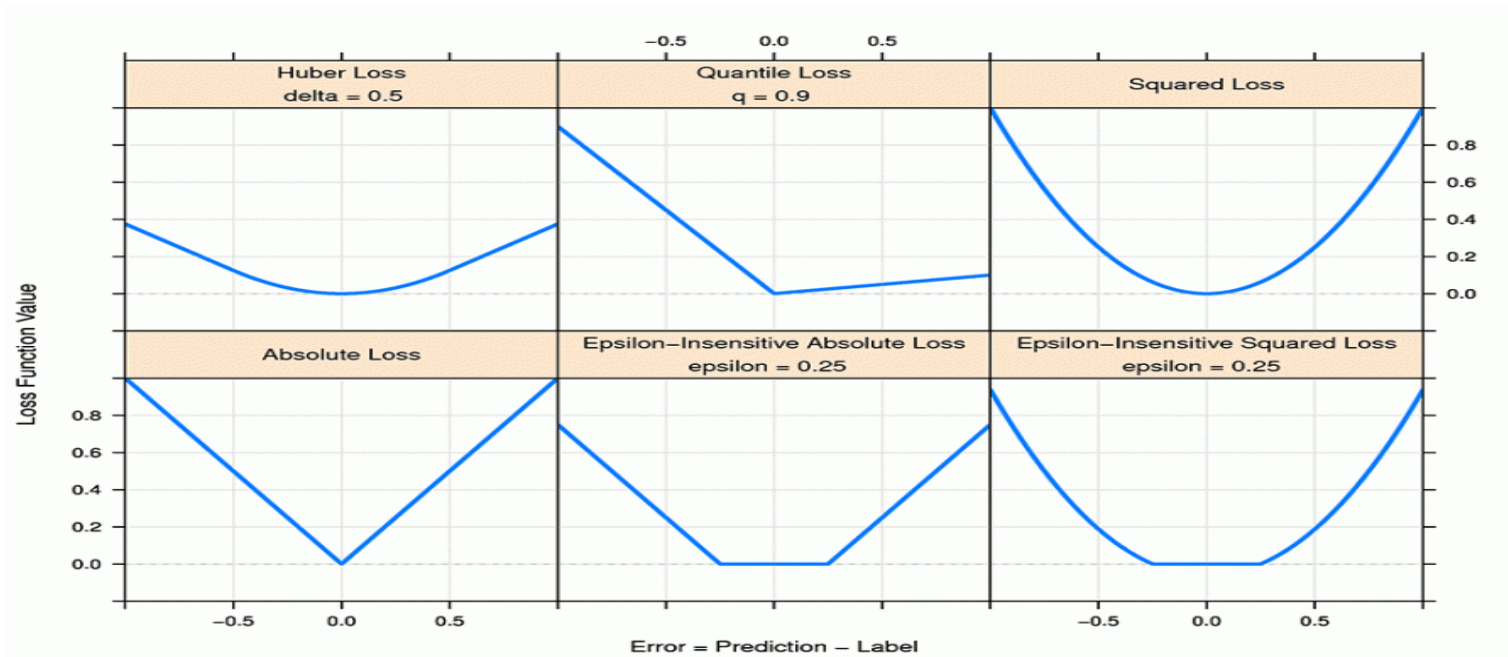
$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |e_t|$$

$$e_t = Y_t - \hat{Y}_t$$

01 머신러닝 용어

III . Loss(Cost) function and Model Parameter

- Loss function의 종류 (다양하게 있습니다..!)



01 머신러닝 용어

III . Loss(Cost) function and Model Parameter

- 머신러닝의 학습 목표는 최적의 모형 파라미터를 찾는 것이고, 최적의 모형 파라미터는 손실 함수를 최소화 하는 파라미터.
- 파라미터 : 모형의 구성 요소이며 데이터로부터 학습 되는 것
- Ex) 회귀모델에서 파라미터들 : intercept(beta0) 와 coefficients

$$\theta_{best} = \arg \min_{\theta} Loss = \arg \min_{\theta} f(y, g(x, \theta))$$

머신러닝 개요 02 연속형/범주형 종속변수

IV. Regression and Classification

- 회귀 모형(Regression model)이란 ?

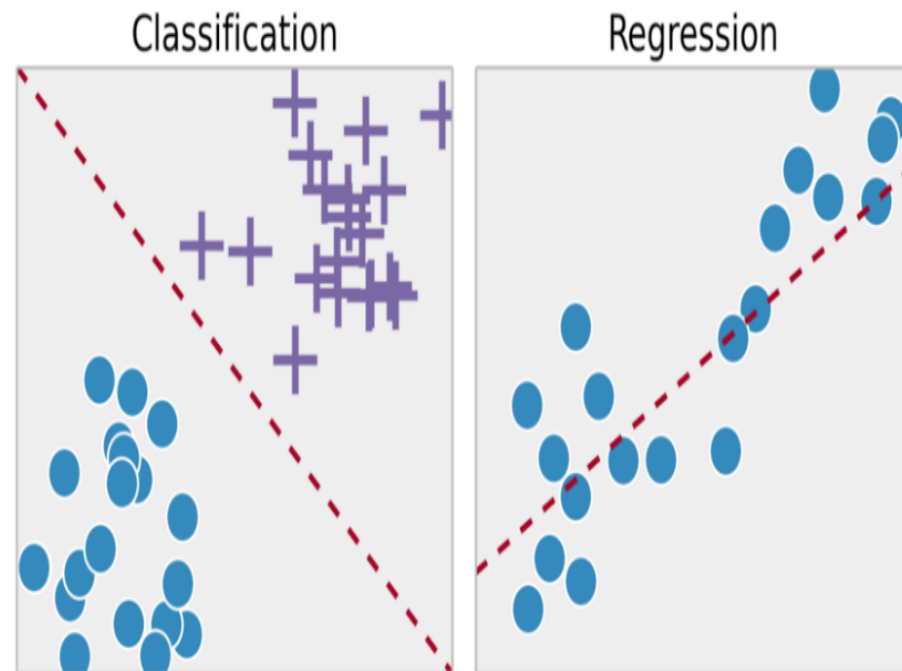
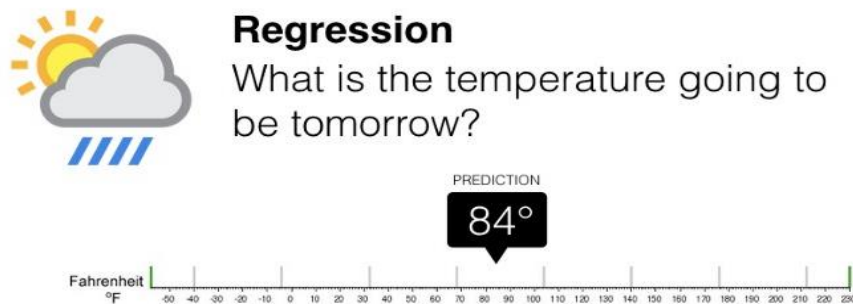
연속형 타겟 변수 (continuous target variable) 과 여러 입력 변수들 (input variables)의 관계를 만드는 모델

- 분류 모형(Classification model)이란 ?

범주형 타겟 변수 (categorical target variable) 과 여러 입력 변수들 (input variables)의 관계를 만드는 모델

머신러닝 개요 02 연속형/범주형 종속변수

IV. Regression and Classification



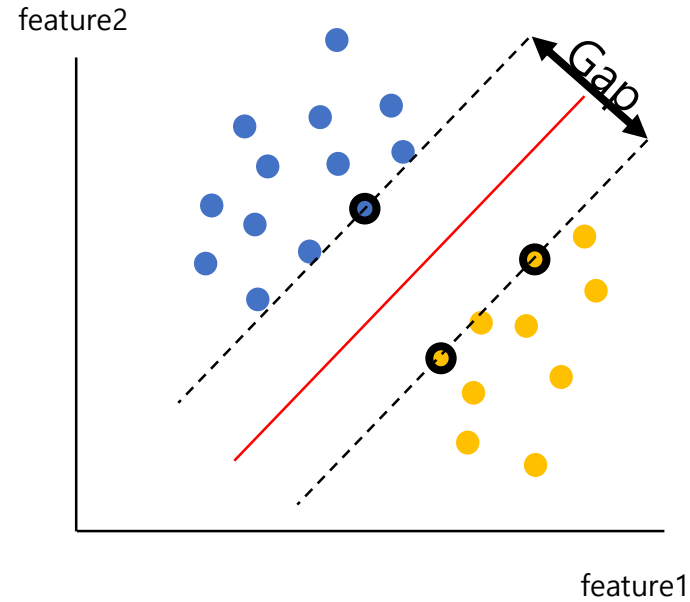
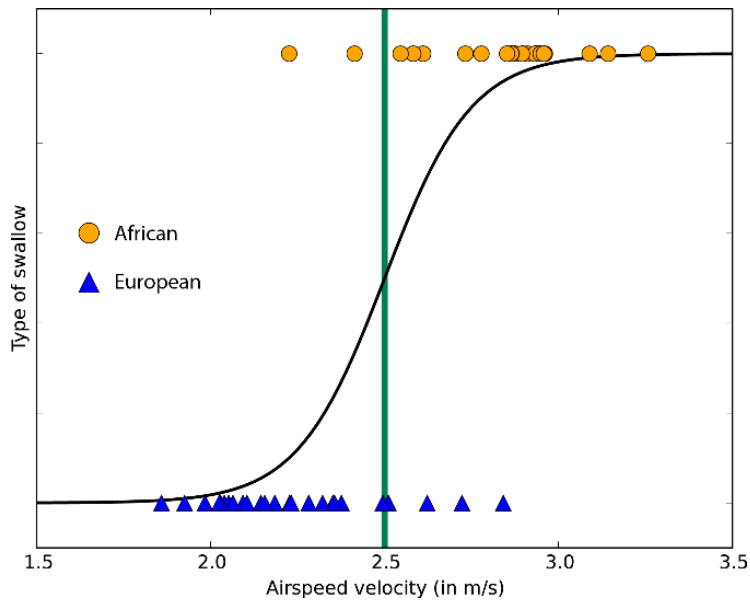
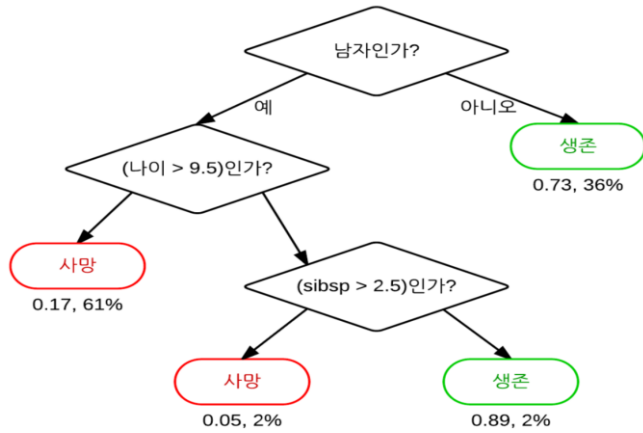
머신러닝 개요 02 연속형/범주형 종속변수

V. Supervised (지도학습) and Unsupervised (비지도학습)

- 지도학습과 비지도학습의 차이점은 **Target의 존재 여부**이다.
- 지도 학습은 정답(Target)을 알려주며 학습시키는 것.
- 비지도 학습은 정답을 따로 알려주지 않고(label이 없다), 비슷한 데이터들을 군집화 하는 것.

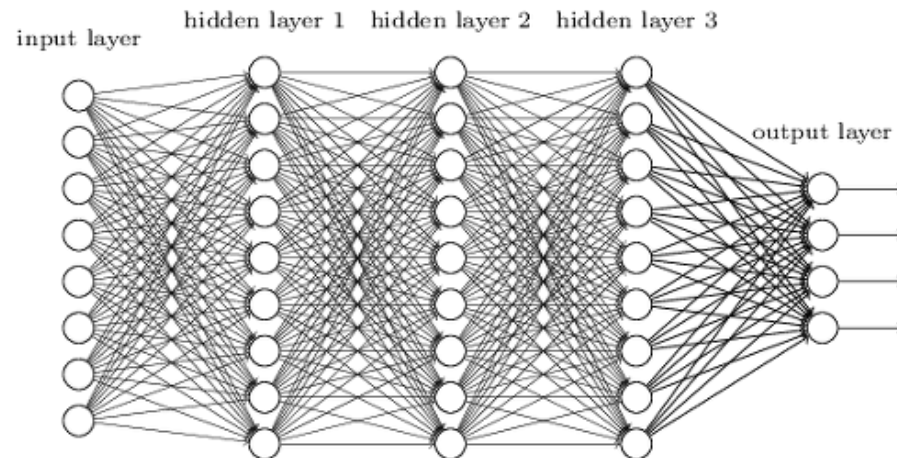
*군집화 : 비슷한 객체끼리 한 그룹으로, 다른 객체는 다른 그룹으로 묶는 행위

머신러닝 개요 02 연속형/범주형 종속변수

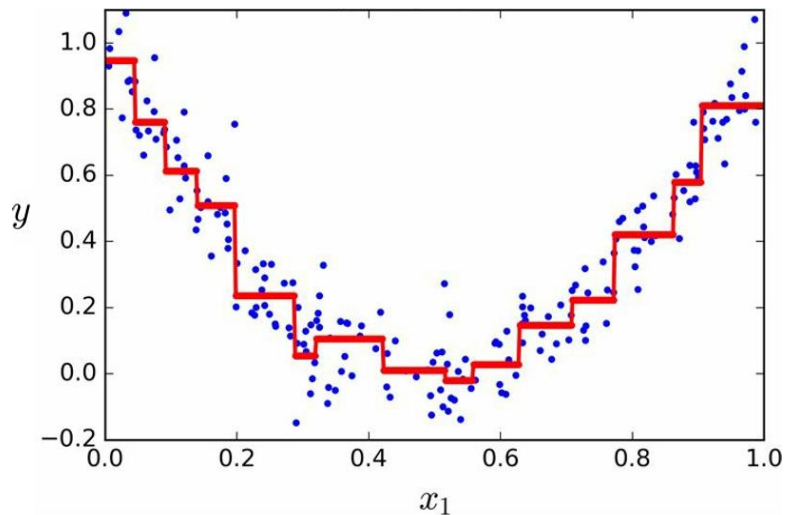
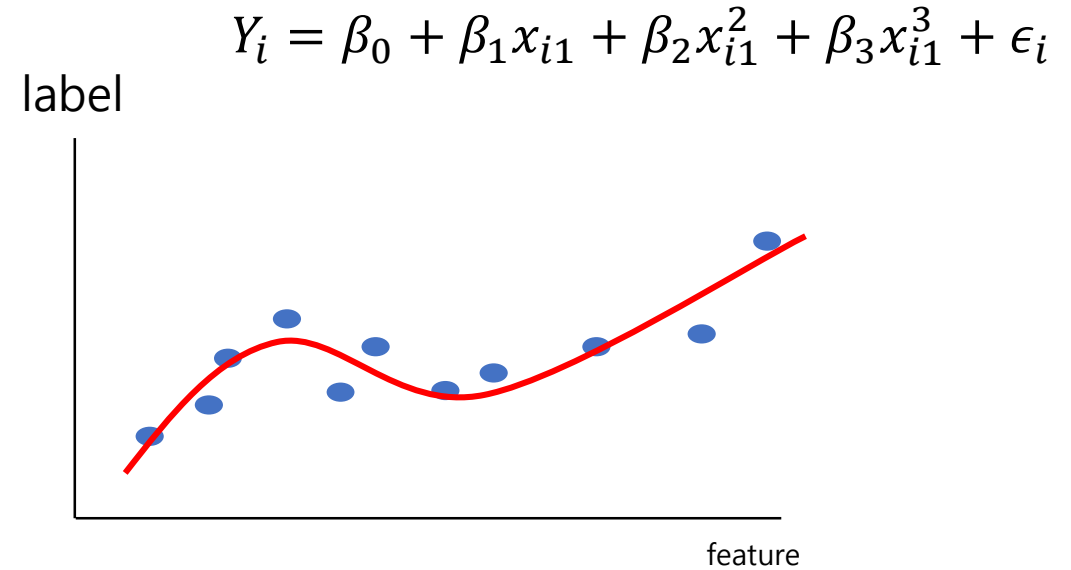
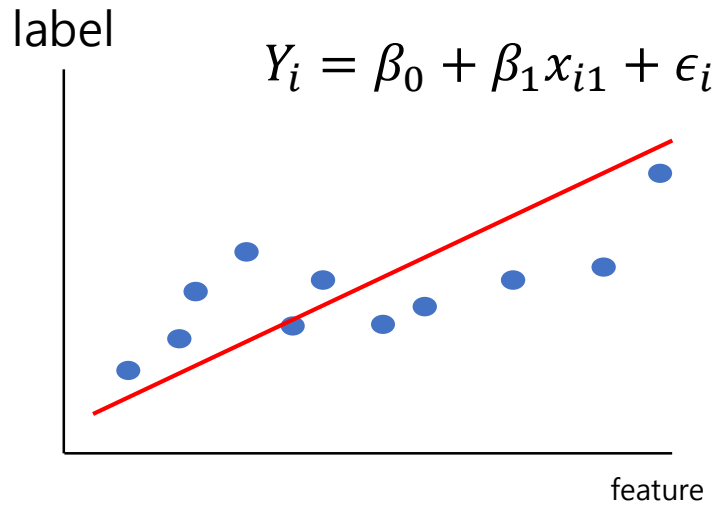


지도학습 - 분류

- 의사결정 나무
- 서포트 벡터 머신
- 로지스틱 회귀
- 인공 신경망



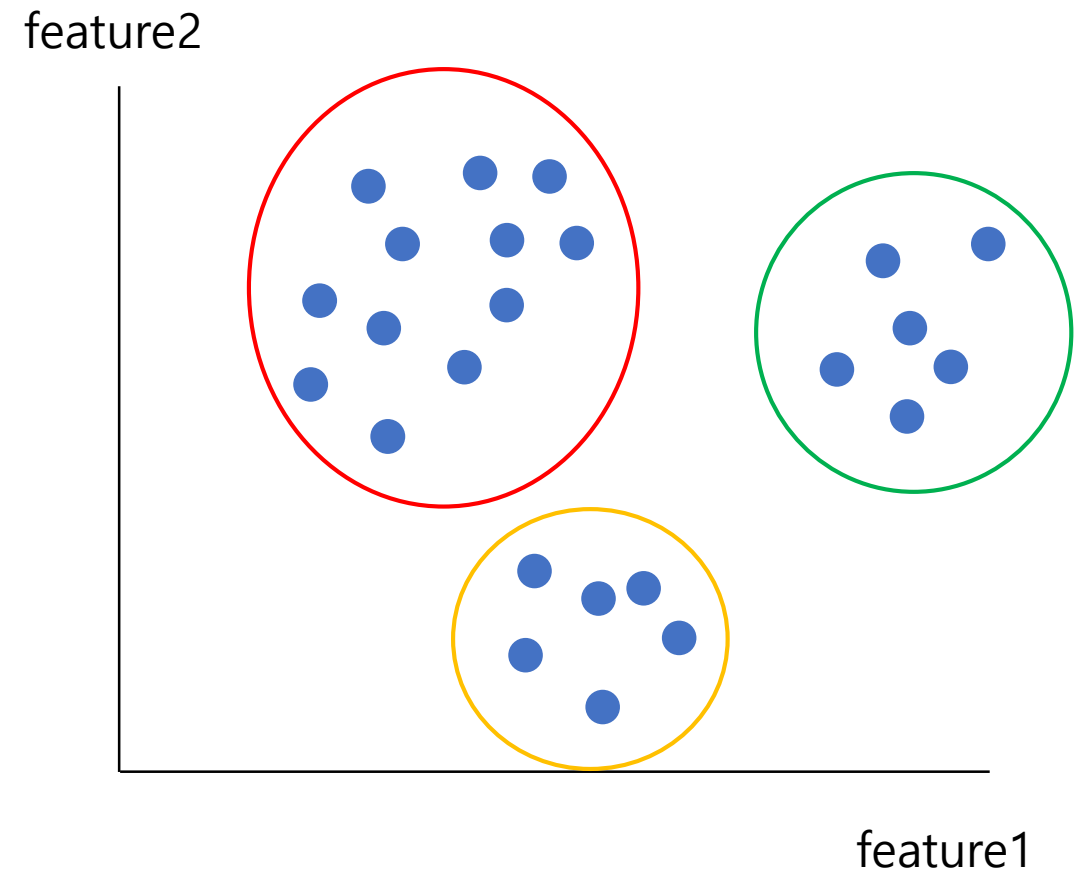
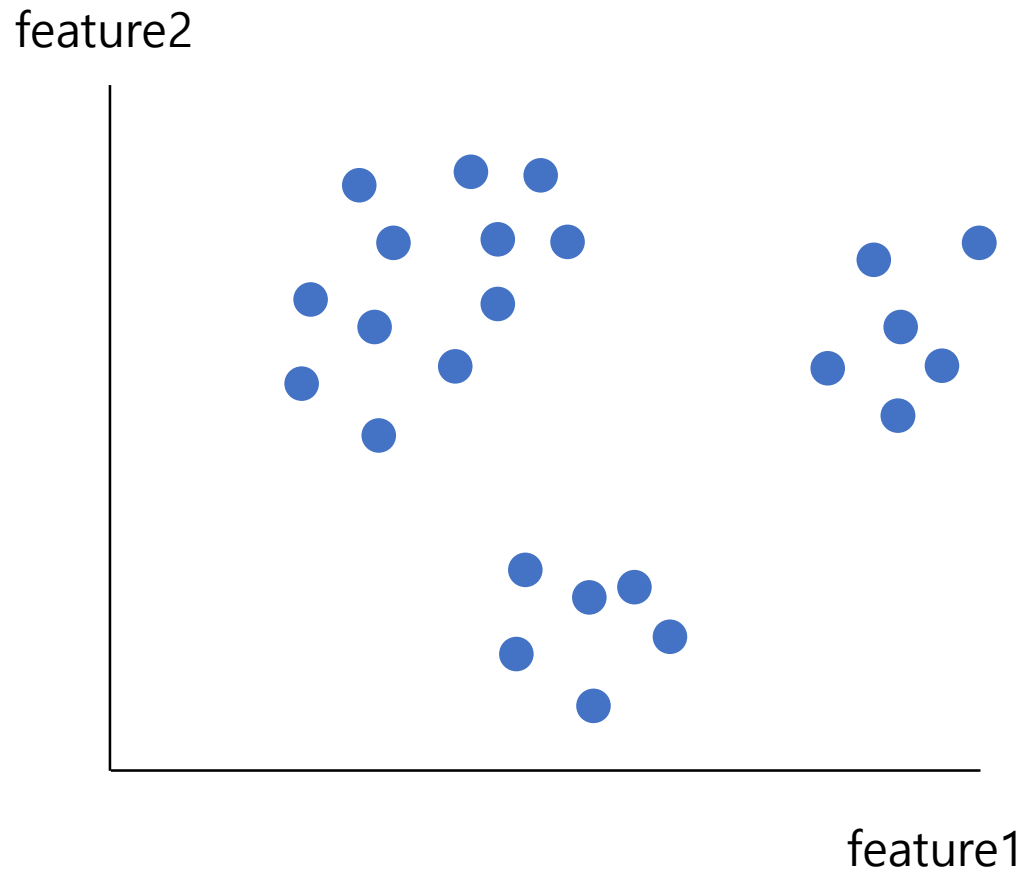
머신러닝 개요 02 연속형/범주형 종속변수



지도학습 - 회귀

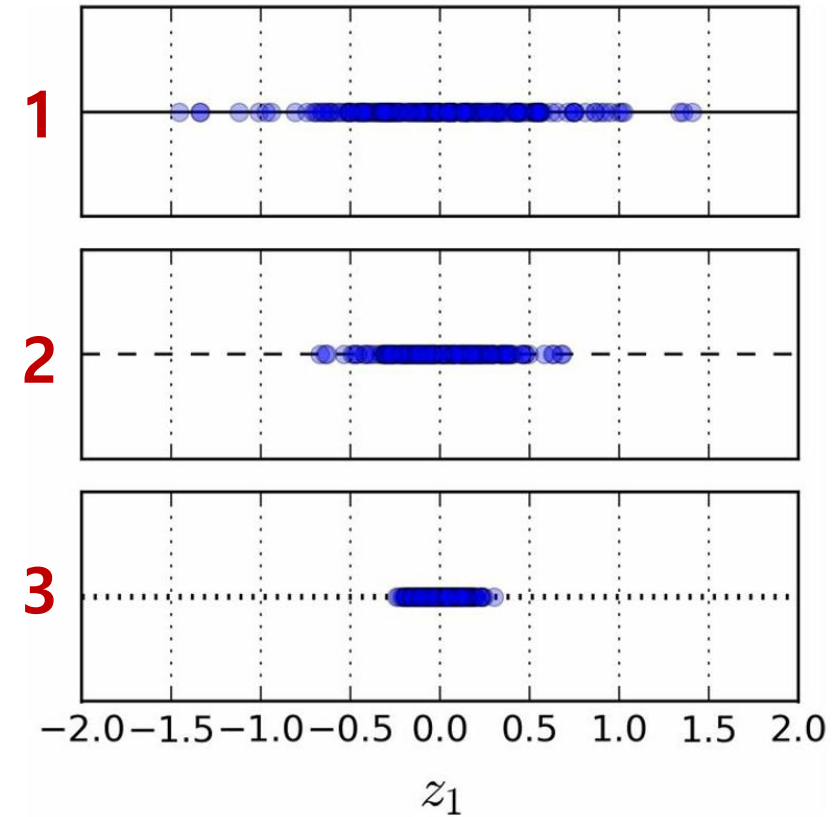
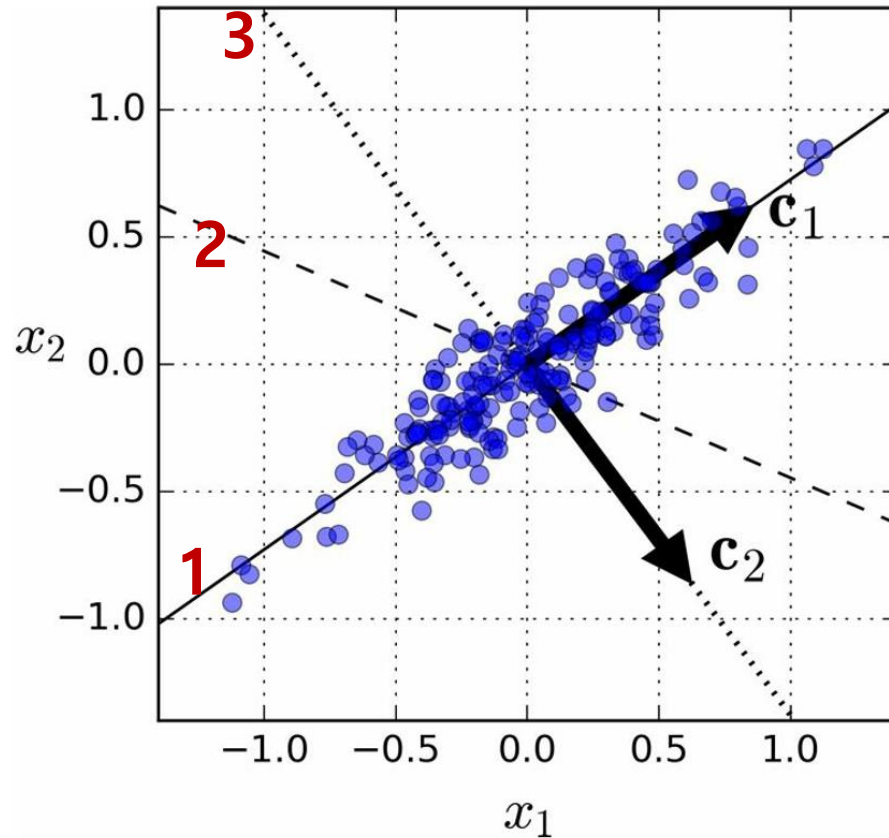
- 선형 회귀
- 다항 회귀
- 결정 나무
- 신경망

머신러닝 개요 02 연속형/범주형 종속변수 » 비지도 학습 - 군집



02 머신러닝 개요 연속형/범주형 종속변수

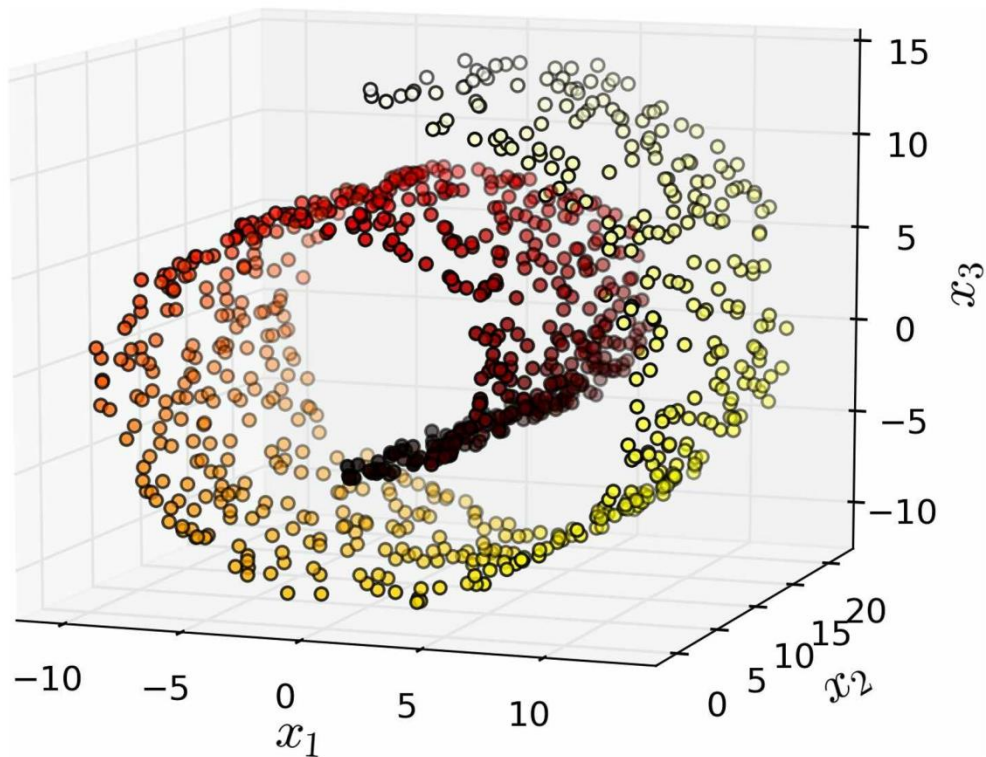
» 비지도 학습 - 차원축소



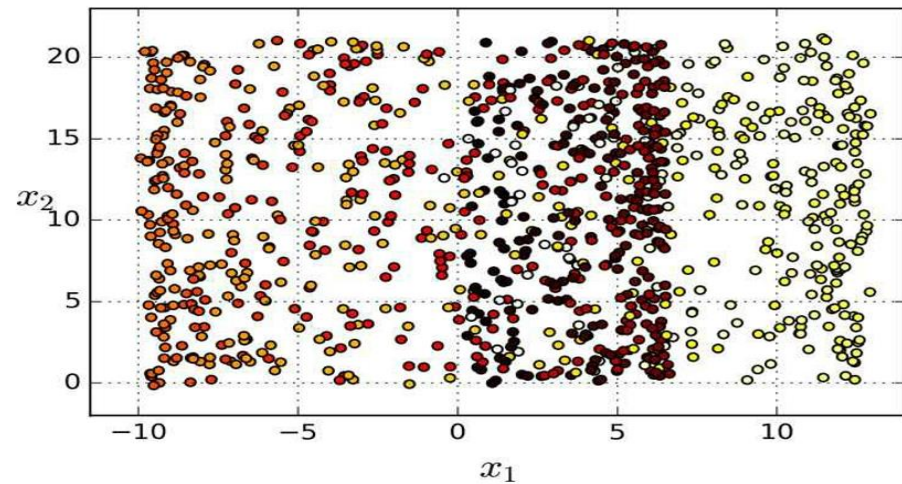
02 머신러닝 개요

연속형/범주형 종속변수

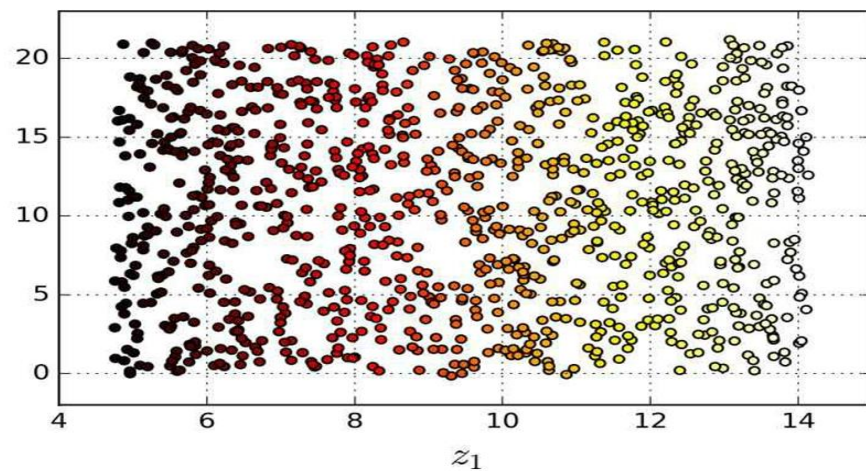
» 비지도 학습 - 차원 축소



투영










































































































































펼치기

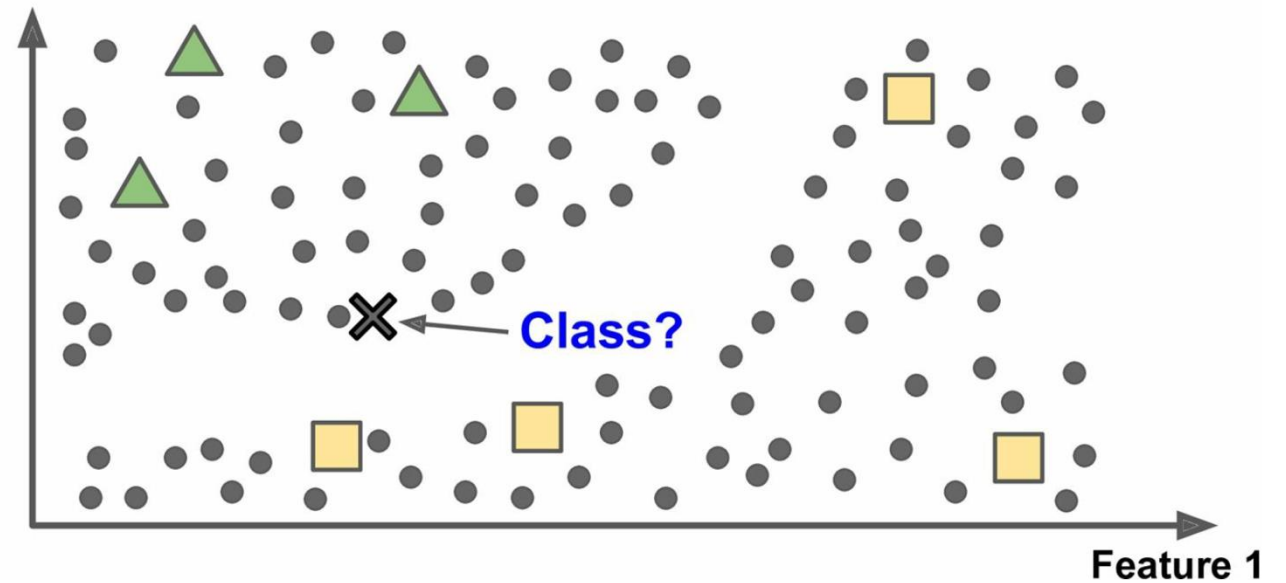


02 머신러닝 개요 연속형/범주형 종속변수

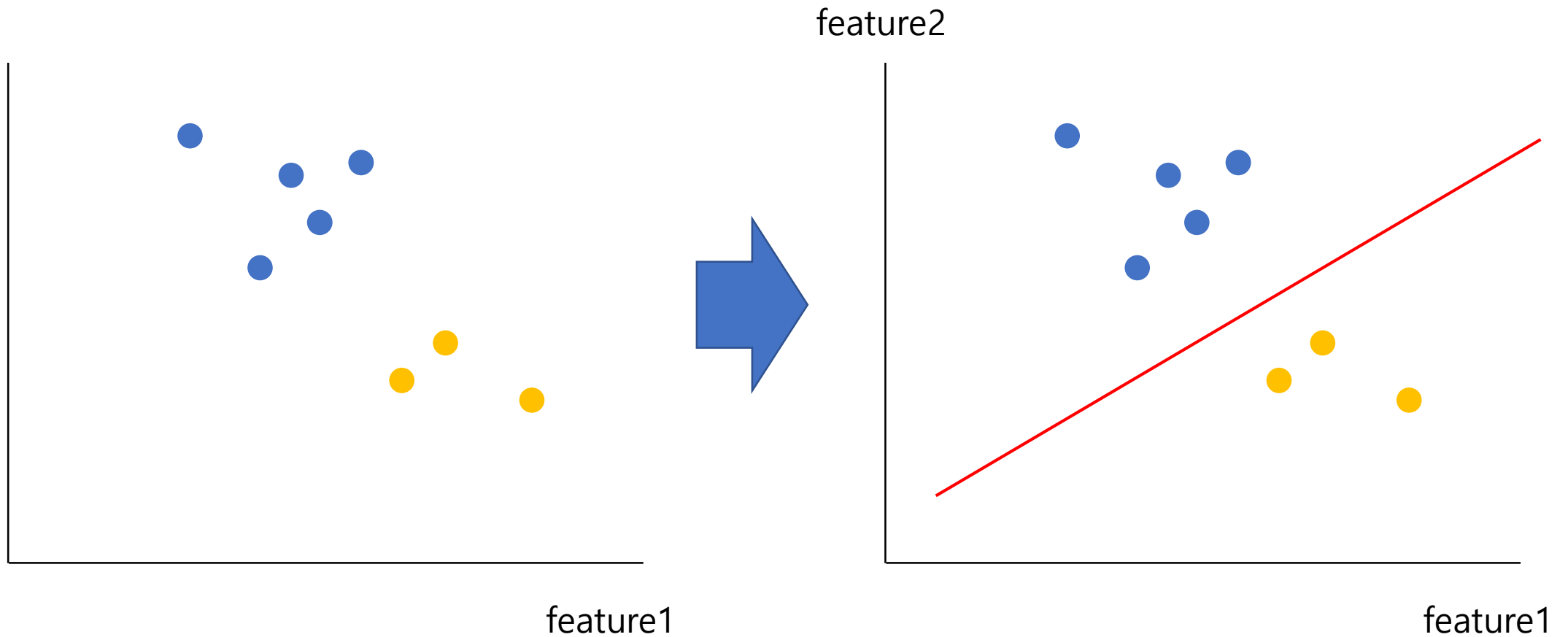
» 준지도학습 - 레이블이 일부만 있는 데이터 구조

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_p	Y			
1												...		
2												...		
3												...		
4												...		
5												...		
6												...		
7												...		
8												...		
9												...		
10												...		
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮	
m												...		

Feature 2

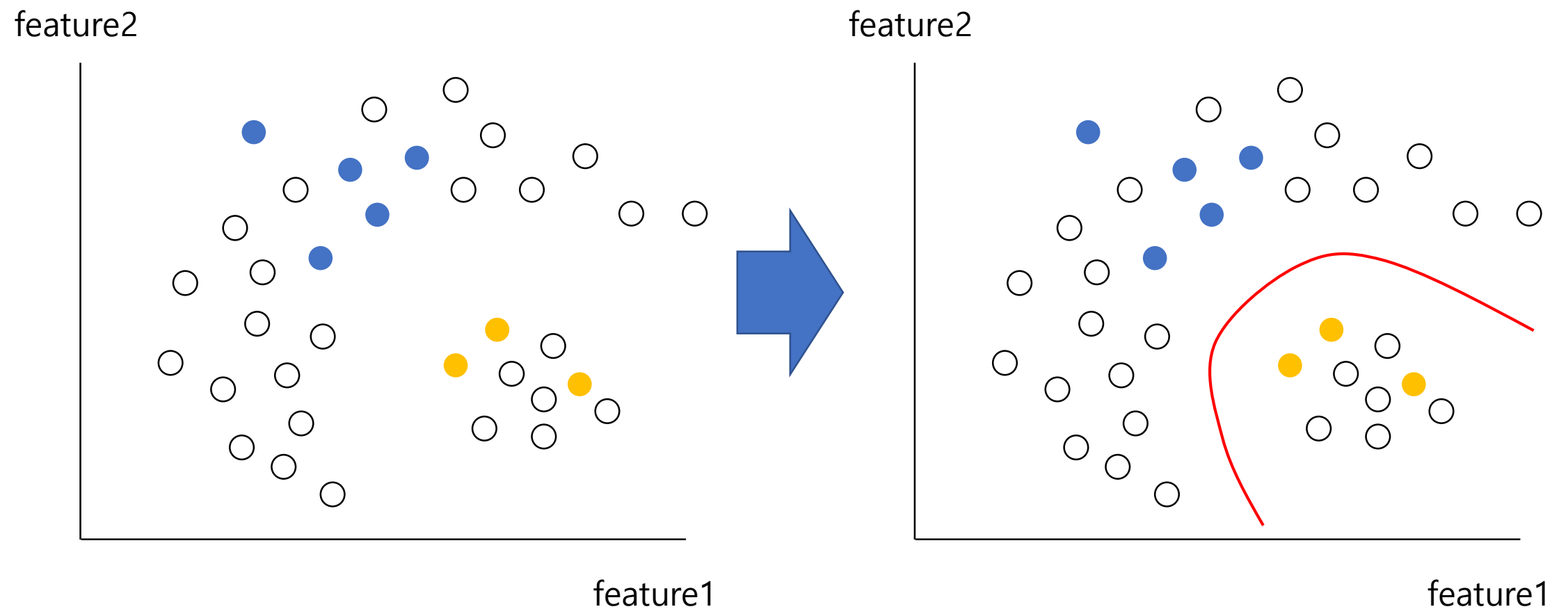


머신러닝 개요 02 연속형/범주형 종속변수



머신러닝 개요

02 연속형/범주형 종속변수

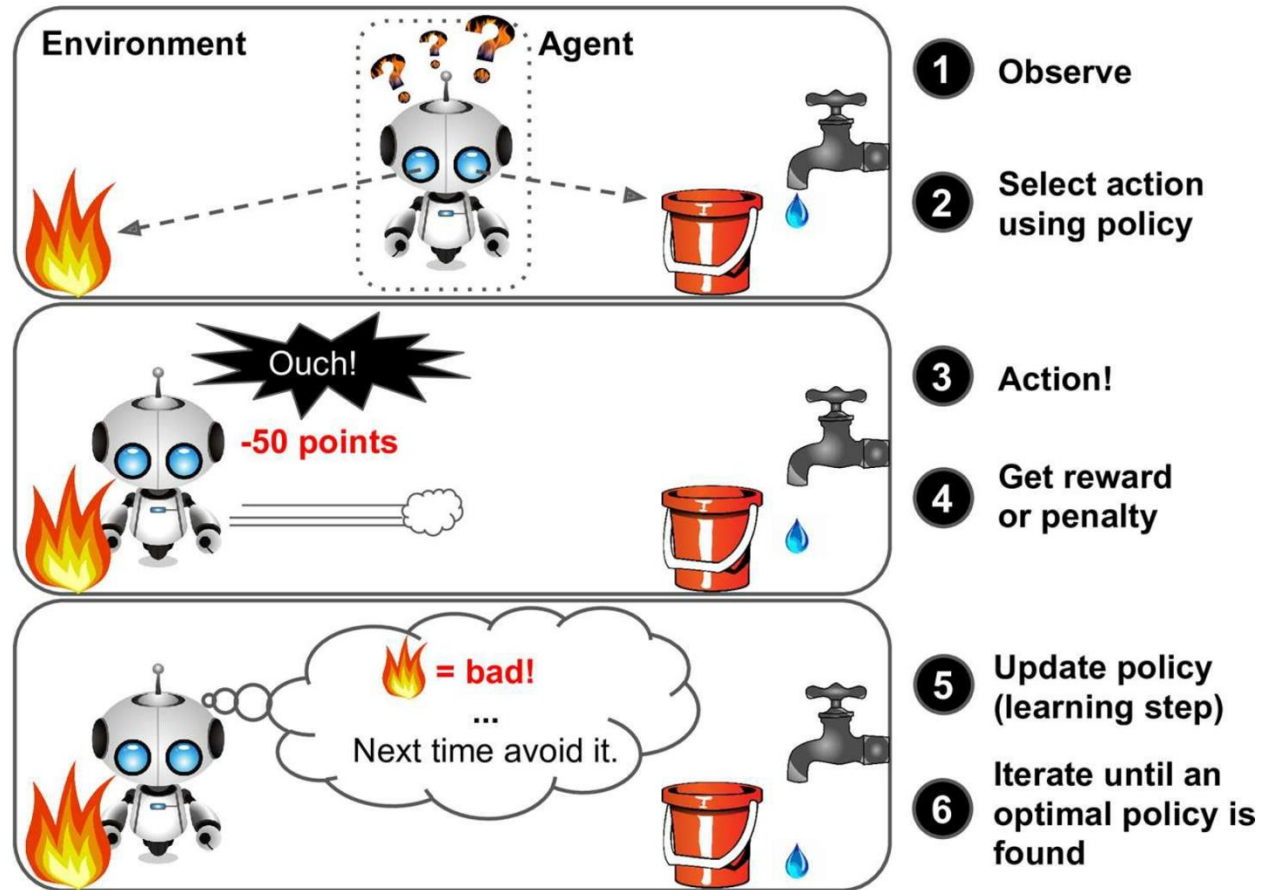


머신러닝 개요

02 연속형/범주형 종속변수

» 강화 학습

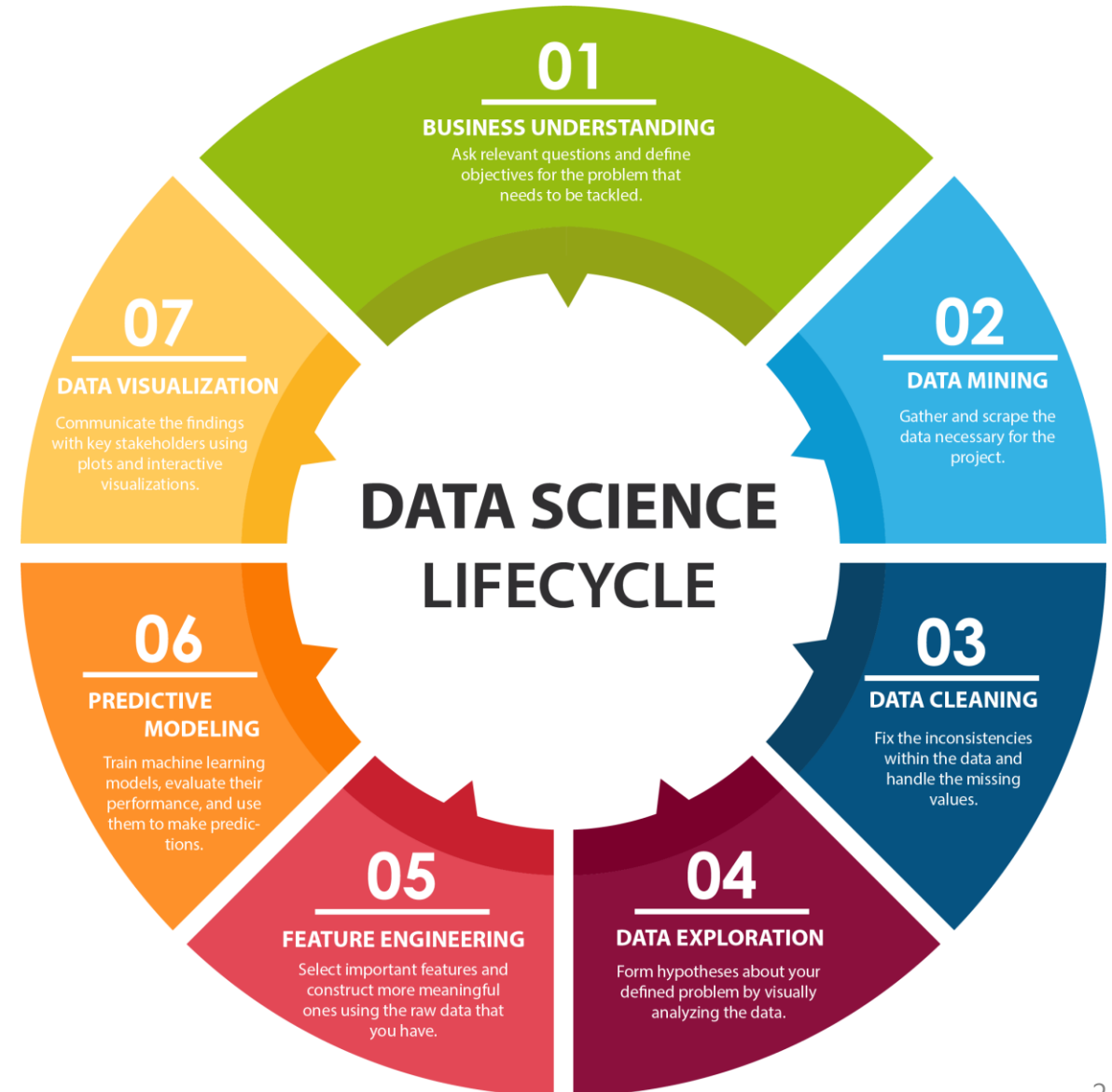
- 매우 다른 종류의 알고리즘
- 환경을 관찰해서 행동을 실행하고 그 결과로 보상을 받으며, 시간이 지나면서 가장 큰 보상을 얻기 위한 전략을 스스로 학습



머신러닝 개요

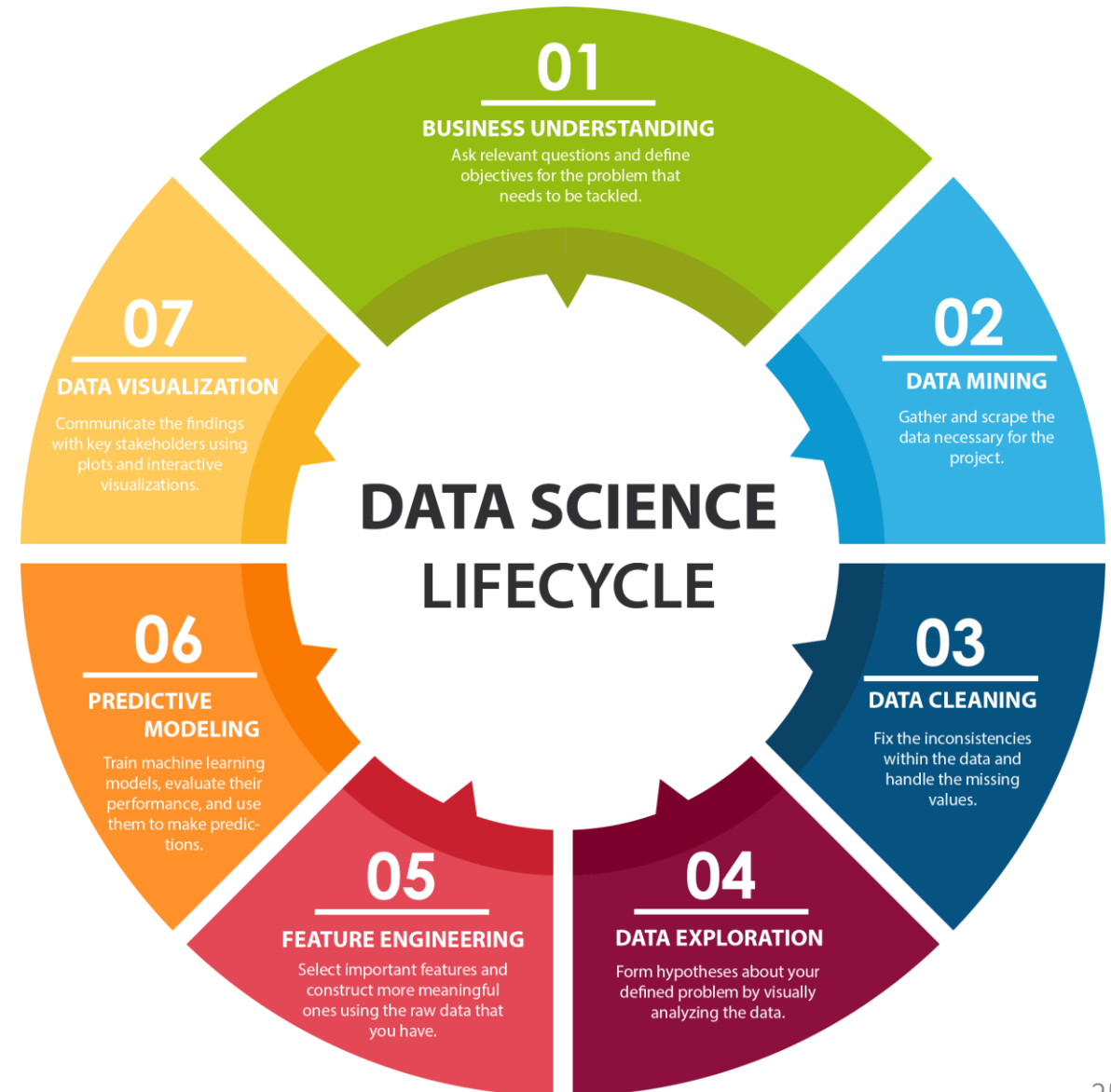
03 데이터 분석 단계

1. Business Understanding
 - 분석목적 및 방향 파악
 - 예) 1. 이번 주 주식 예측
2. 항공기 지연 요인 파악
2. Data Mining
 - 데이터 수집 (크롤링 등을 활용)
3. Data Cleaning
 - 이상치, 결측치 등 처리
4. Data Exploration
 - 시각화 등을 활용한 데이터 탐색



머신러닝 개요 03 데이터 분석 단계

5. Feature Engineering
 - 모델에 넣을 변수 선택
 - 파생변수, 외부변수 생성 등
6. Predictive Modeling
 - 모델 생성 및 평가
7. Data Visualization
 - 분석 결과 시각화 및 보고서 작성



02

Pre-Processing

2-1. 범주형 변수 처리

2-2. 이상치, 결측치, 정규화

01 Pre-Processing 범주형 변수 처리

- 범주형 값은 여러개의 다른 상태를 나타내는 값이다.
- 범주형 값을 'A', 'B', 'C'라는 문자로 표현하는 경우도 있고 '1', '2', '3'과 같이 숫자로 표현하는 경우도 있지만 이 경우는 'A'라는 글자대신 '1'이라는 글자를 이용한 것 뿐이지 **숫자로서의 의미는 없다.**
- 즉, '2'라는 값이 '1'이라는 값보다 2배 더 크다는 뜻이 아니다.
- 모델링을 할 때는 숫자가 아닌 독립변수 값을 쓸 수 없기 때문에 어떤 방식으로든 범주형 독립변수의 값을 사용할 수 있는 방법을 찾아야 한다. 범주형 독립변수를 처리하는 가장 일반적인 방법은 **더미변수(dummy variable)로 변환**하는 것이다.

과일종류	문자표현	숫자표현
사과	A	1
바나나	B	2
오렌지	O	3

01 Pre-Processing 범주형 변수 처리

》》 더미변수 (Dummy Variable)

- 0 또는 1만으로 표현되는 값으로 어떤 특징이 존재하는가 존재하지 않는가를 표시하는 독립변수

방법

1. 원핫인코딩(One-Hot-Encoding)방식

- 더미변수의 값을 원핫인코딩(one-hot-encoding)방식으로 지정

2. 축소랭크(Reduced-Rank)방식

- 특정한 하나의 범주값을 기준값(reference)으로 하고 기준값에 대응하는 더미변수의 가중치는 항상 1으로 놓는다.

01 Pre-Processing 범주형 변수 처리

I. 원핫인코딩(One-Hot-Encoding)

- 해당 하는 값에 대해서 맞으면 1 아니면 0
- 회귀분석에서는 독립변수 간 선형독립 즉, full-rank 만족을 위해 cell-means model(intercept가 없는 모델)이 생성

과일종류 \ 변수	intercept	사과	바나나	오렌지
사과	존재x	1	0	0
바나나	존재x	0	1	0
오렌지	존재x	0	0	1
바나나	존재x	0	1	0

01 Pre-Processing 범주형 변수 처리

II. 축소랭크(Reduced-Rank)

기준값(reference)를 제외하고 해당하는 값에 대하여 맞으면 1 아니면 0

오른쪽 표는 사과를 기준값으로 한 회귀모형의 예

회귀분석에서는 이를 **reference coding** 이라고함.

과일종류 \ 변수	intercept	바나나	오렌지
사과	1	0	0
바나나	1	1	0
오렌지	1	0	1
바나나	1	1	0

01 Pre-Processing 범주형 변수 처리

결론적으로, 범주형 변수가 모델에 들어갈 경우 K개 혹은 K-1개의 변수가 생성된다.

관측치가 정말 많다면 상관없을 수 있지만, 변수가 많아지면 모델이 잘 학습되지 않을 수 있다.

단, 해석적인 측면에서는 직관적이고 용이할 수 있다.

Pre-Processing 02 이상치, 결측치, 정규화

? 결측치란

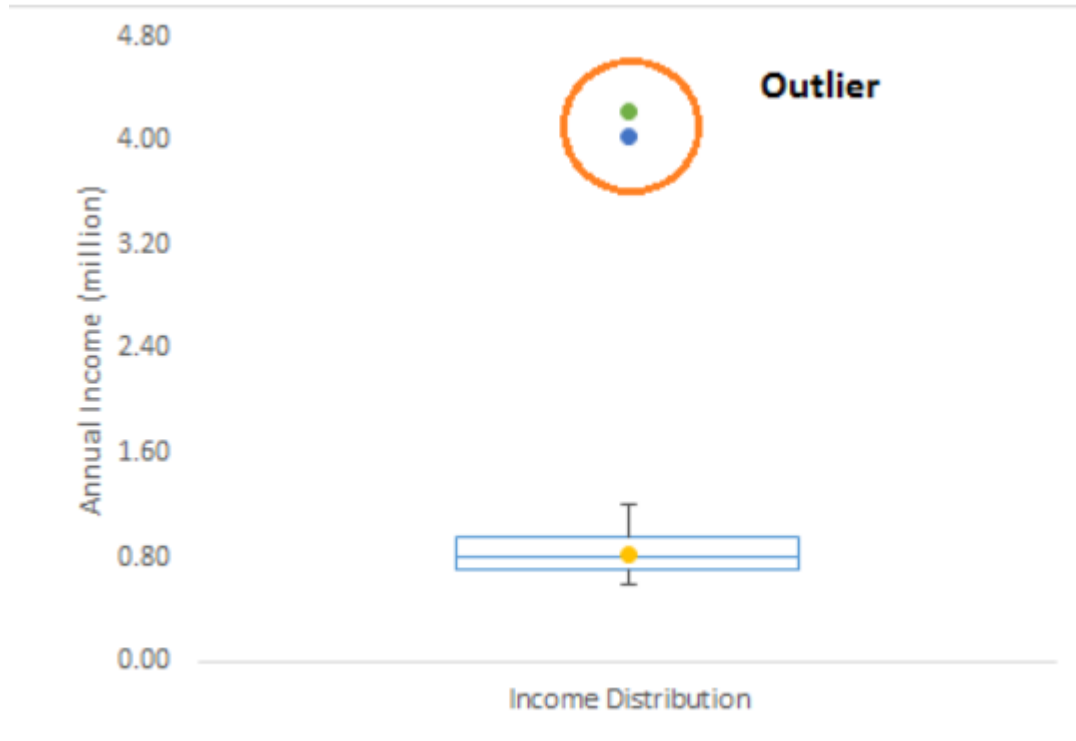
누락된 값, 비어 있는 값을 의미하며 NA 혹은 999 등 약속된 값으로 주로 표기
실제 데이터는 데이터 수집과정에서 발생한 오류 등으로 인해 결측치가 포함되어있는 경우가 많음.
결측치를 정제하는 과정을 거쳐야만 함수 사용에 문제가 발생하지 않고 분석 결과에 왜곡이 없음.

? 이상치란

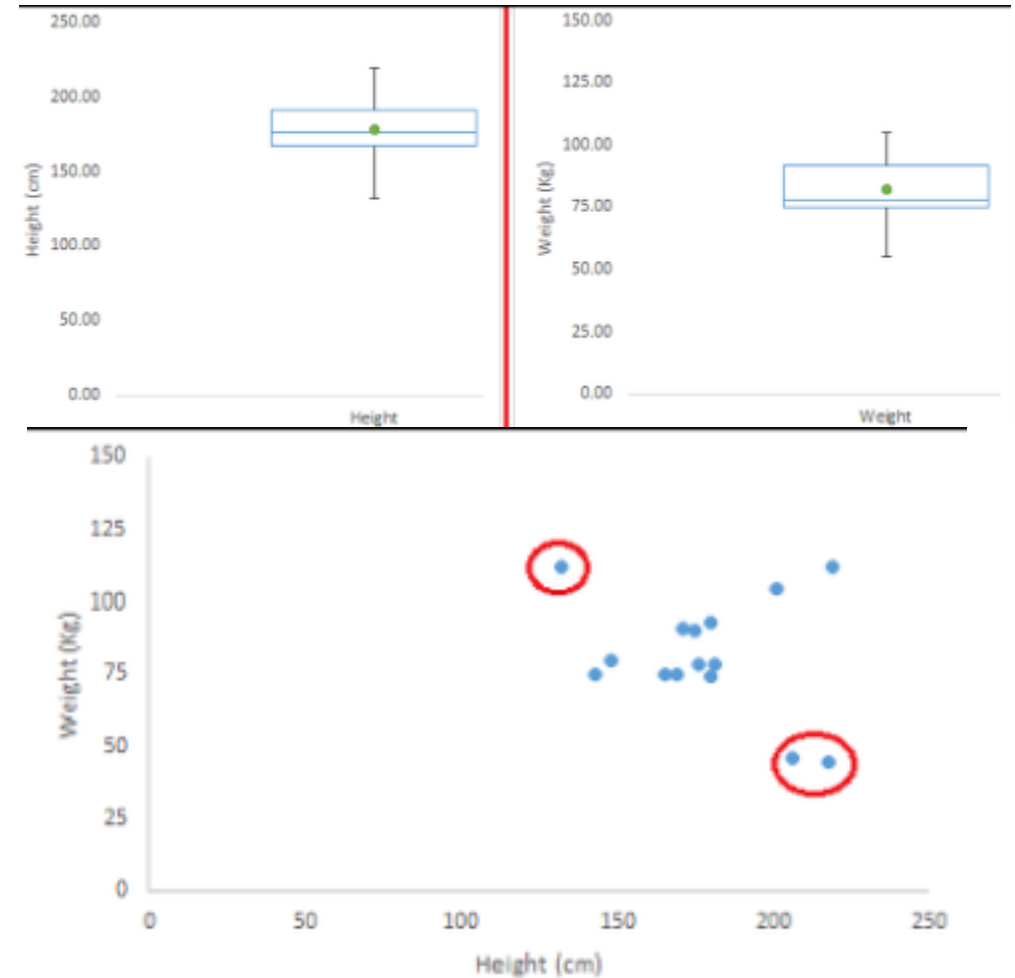
샘플의 전체적인 패턴에서 벗어나게 관측되는 값.
결측치와 마찬가지로 데이터 수집과정에서 발생할 수 있고 측정과정이나 샘플링 과정에서 발생할 수 있음.
분포적인 관점 혹은 다양한 시각화를 통해서 확인 가능

02 이상치, 결측치, 정규화

Pre-Processing



Univariate Outlier



Multivariate Outlier

Pre-Processing 02 이상치, 결측치, 정규화

? 결측치 처리방법

속 편하게 행 삭제

중심 경향 값 넣기 (평균, 중앙값, 최빈값 등) -> 무난하게 채울 수 있는 방법이나 분산이 줄어들음

Regression Imputation -> 각 관측치의 특성을 고려하여 회귀모형을 이용하여 결측치 대체

MICE (Multivariate Imputation via Chained Equations) -> 다양한 모델을 활용한 결측치 대체

? 이상치 처리방법

속 편하게 행 삭제

- $1.5 \times IQR \sim 1.5 \times IQR$ 이용하여 제거, 이 때 1.5 라는 수치는 조정 가능

범주화

Log transformation

Pre-Processing

02 이상치, 결측치, 정규화

» 정규화

- Regularization(정규화)는 머신러닝에서 훈련 데이터로 훈련시킨 모델을 훈련 데이터 뿐만 아니라 다른 데이터에서도 잘 들어 맞을 수 있도록 해 주는 것
- 모델의 복잡도에 제약/페널티를 부여하는 것, 오버피팅을 방지하는 것을 목표로 한다.
- **L2, L1 cost**가 대표적이며, Cost를 최소화 시키는 것을 머신러닝 학습 목표로 한다.

• ex) 선형회귀, 로지스틱 회귀

• L1 cost : $Loss + \lambda \sum_{i=1}^p |\beta_i|$ 경계면이 날카로워 변수 선택 효과가 있다. (Sparse modeling)

• L2 cost : $Loss + \lambda \sum_{i=1}^p |\beta_i|^2$ 경계면이 날카롭지 않아 최소점을 찾기 쉽다.

03

EDA / 시각화

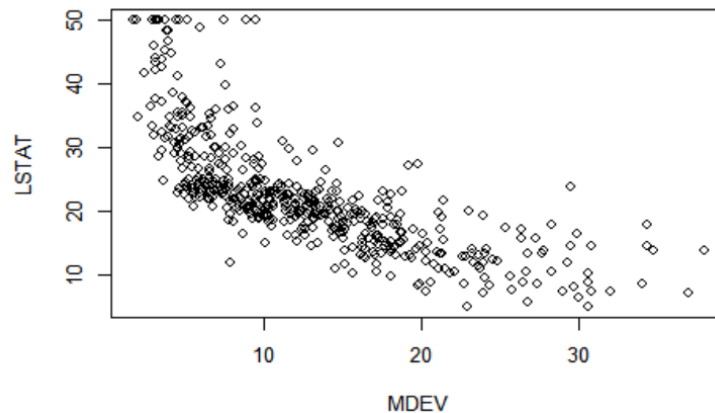
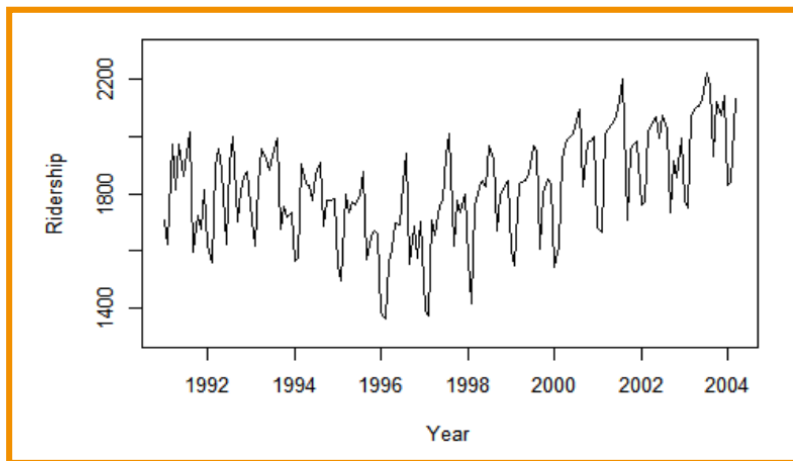
3-1. 기본차트

3-2. 분포차트

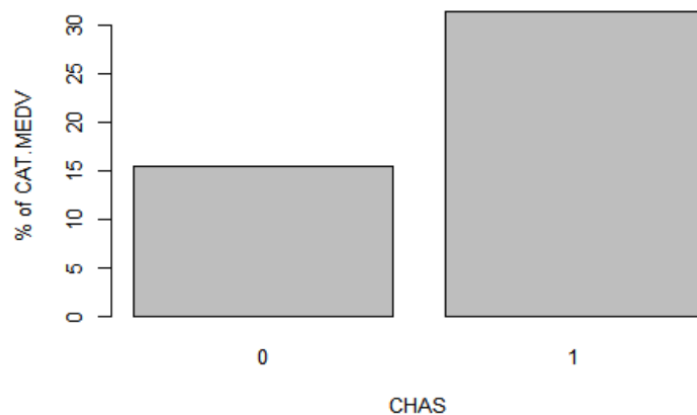
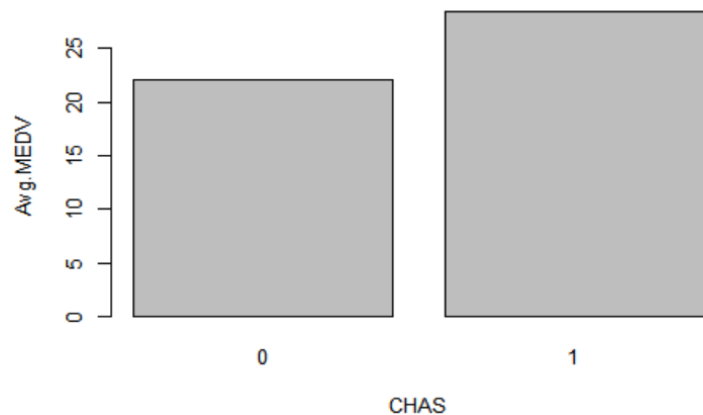
3-3. 차트조절

01 EDA / 시각화 기본차트

» 막대차트, 선그래프, 산점도

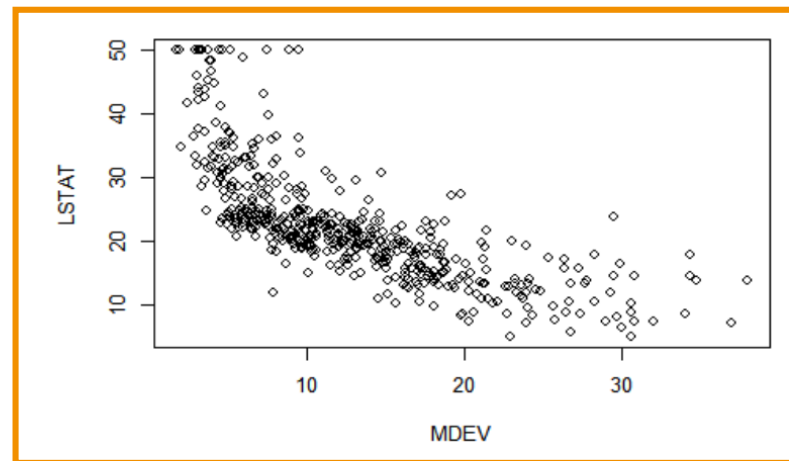
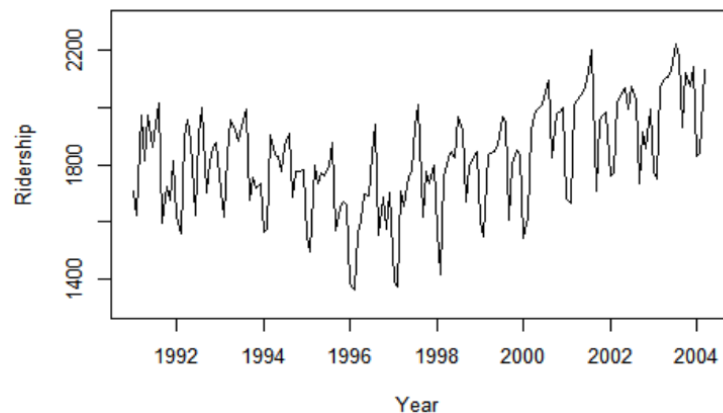


- 선그래프
 - 시계열 데이터
 - X와 Y 모두 연속형 변수
- ex) 앰트랙의 월간 승객수



01 EDA / 시각화 기본차트

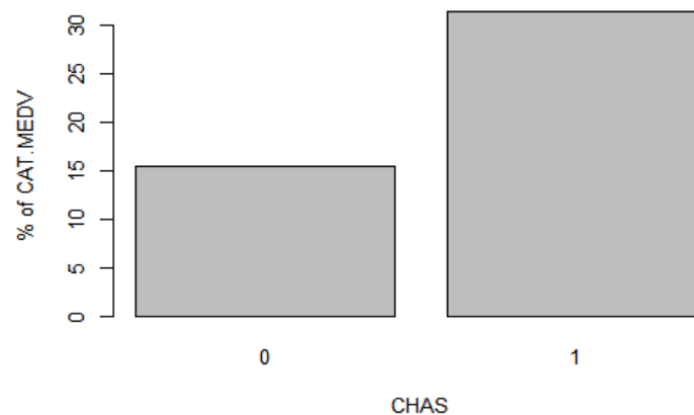
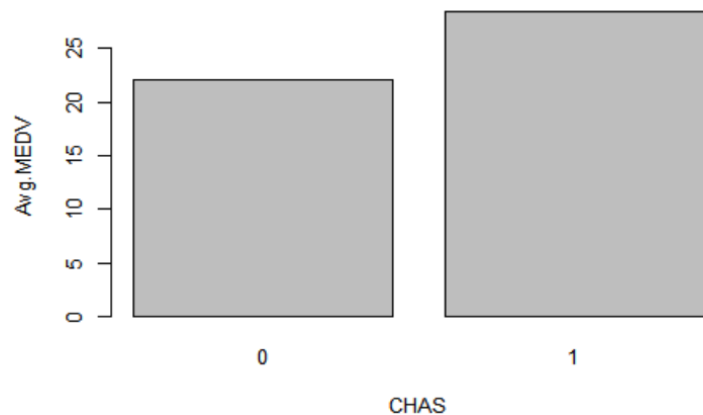
>> 막대차트, 선그래프, 산점도



• 산점도

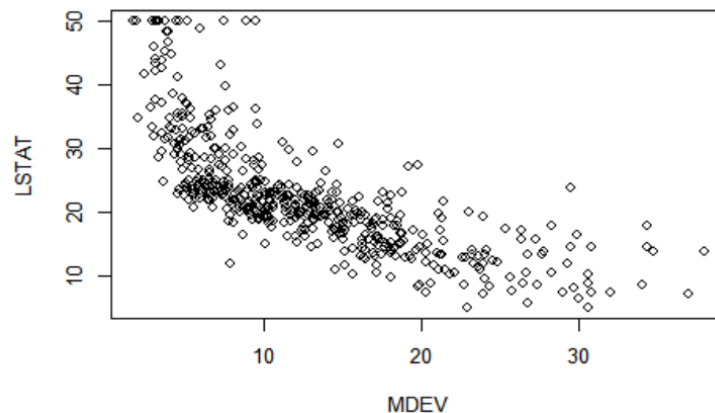
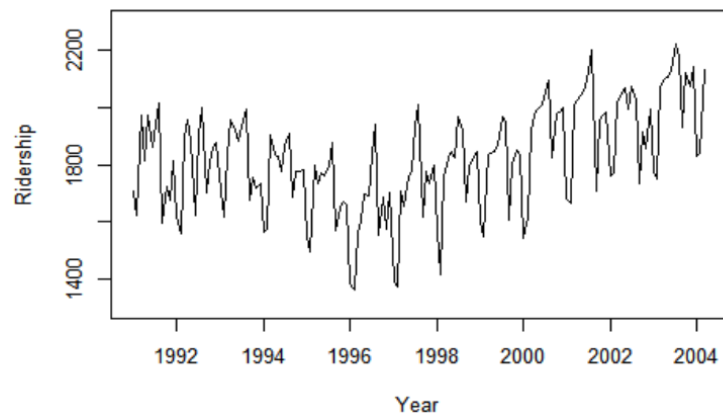
- 수치형 데이터
- 두 변수간의 관계 파악

ex) X: 주택가격의 중앙값
Y: 저소득층 비율



01 EDA / 시각화 기본차트

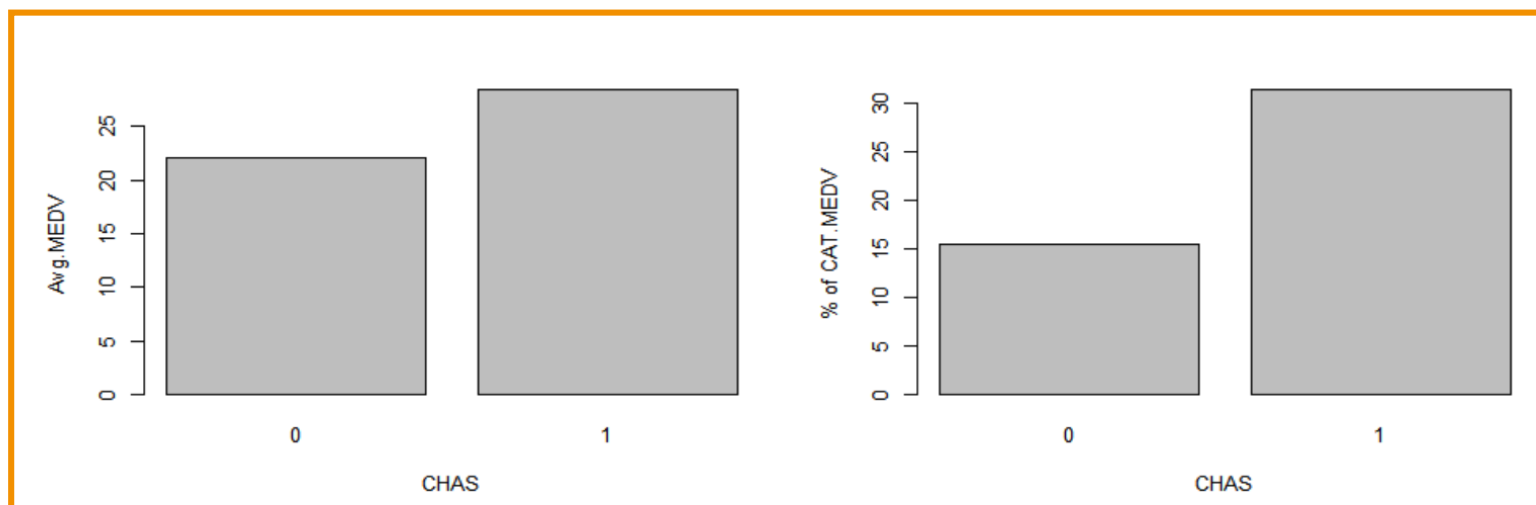
» 막대차트, 선그래프, 산점도



• 막대차트

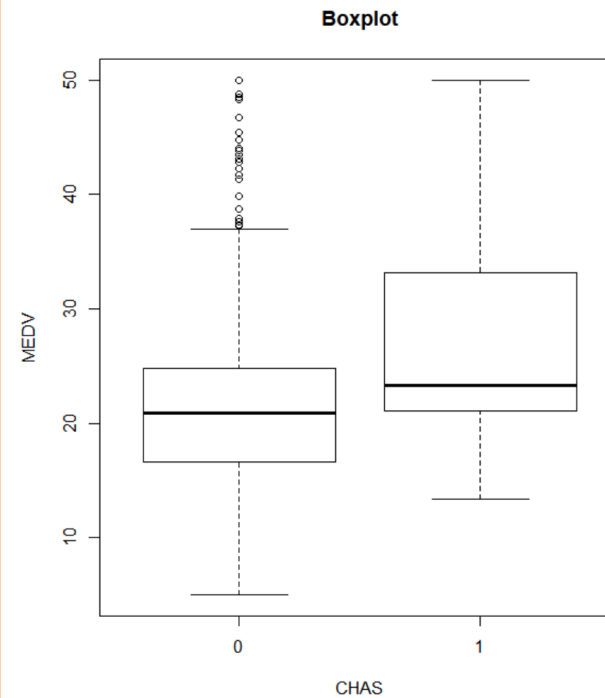
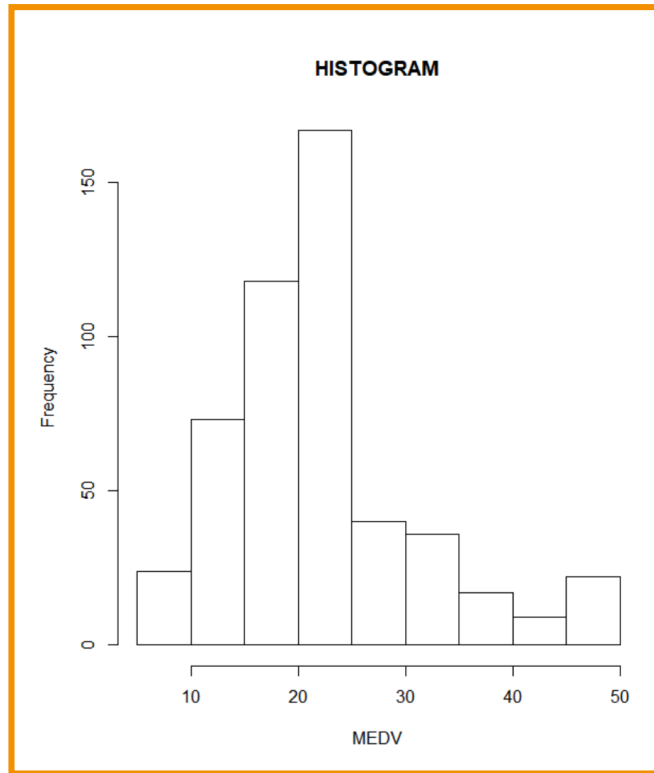
- 그룹별로 비교할 때 유용
- 각 막대는 다른 집단 의미

ex) X : 찰스강 인접 여부
Y : 주택가격의 평균



EDA / 시각화 02 분포차트

» 박스플롯 & 히스토그램



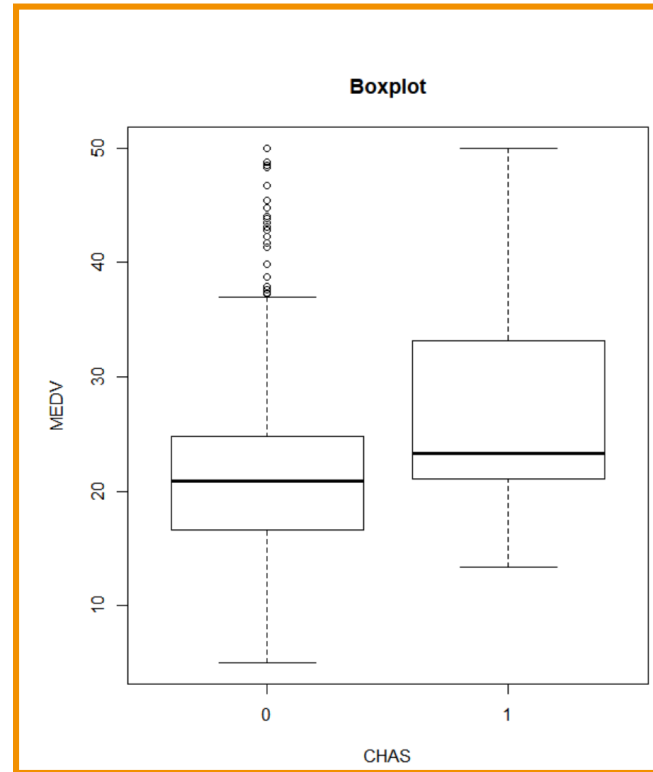
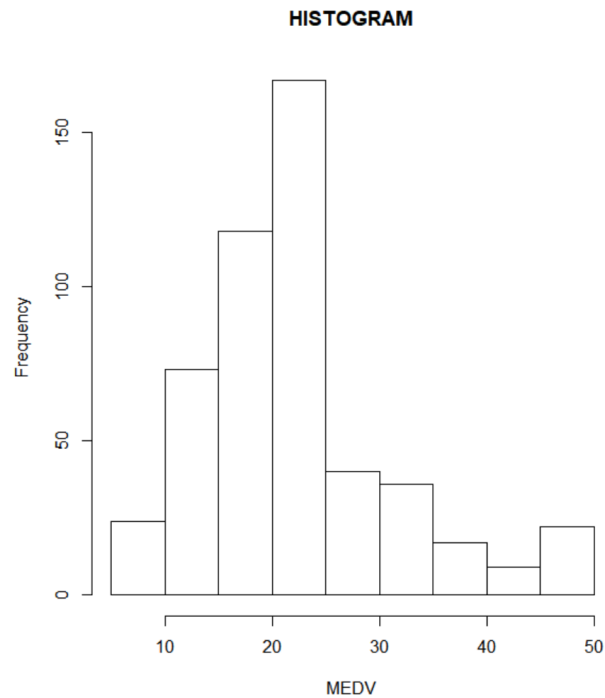
• 히스토그램

- 모든 x 값의 출현 빈도
이상치/변수선택에 반영
- 편향 되어있을 경우 반드시
변환과정 거치기

ex) X : 주택가격의 중앙값

02 EDA / 시각화 분포차트

» 박스플롯 & 히스토그램



• 박스플롯

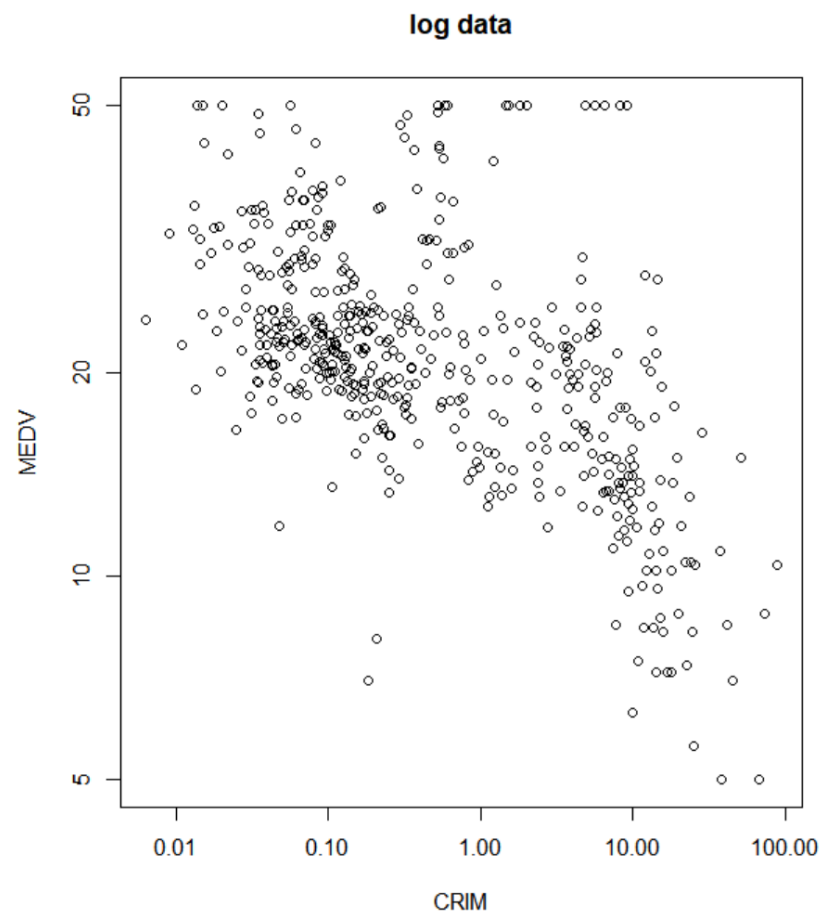
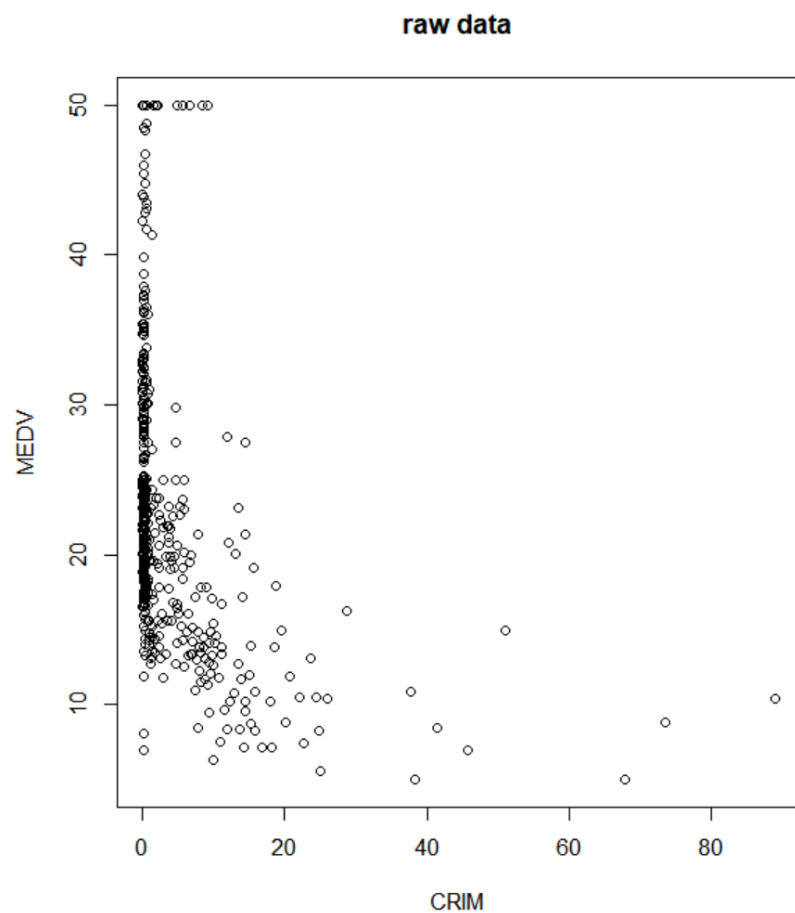
- X는 범주형, Y는 연속형 변수
- X에 대한 Y의 분포를 쉽게 파악가능

ex) X : 찰스강 인접 여부
Y : 주택가격의 중앙값

EDA / 시각화

03 차트조절

>> Log 변환



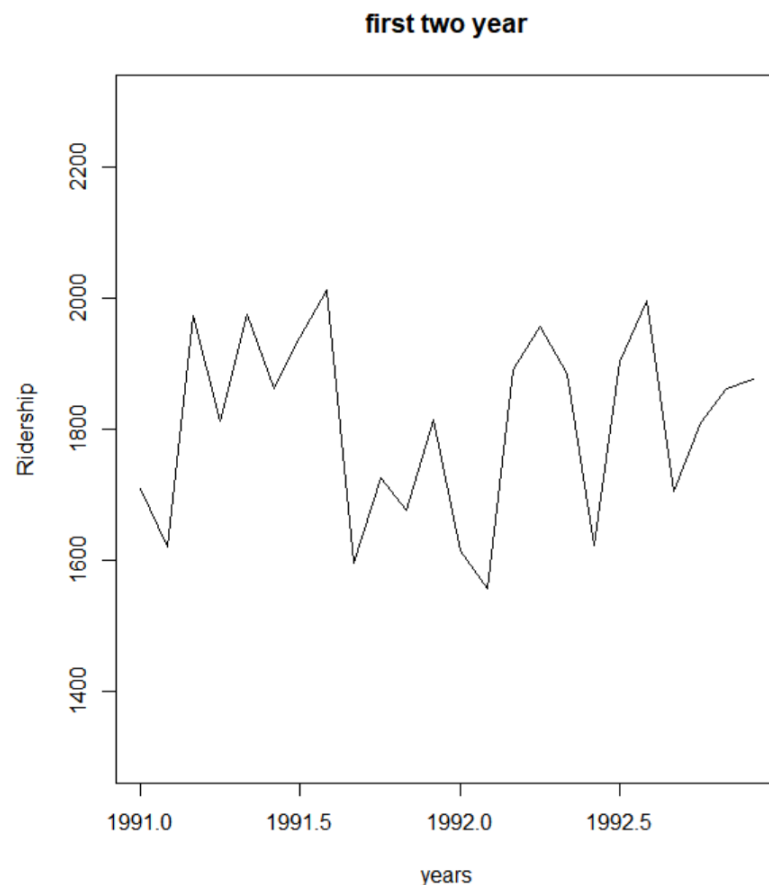
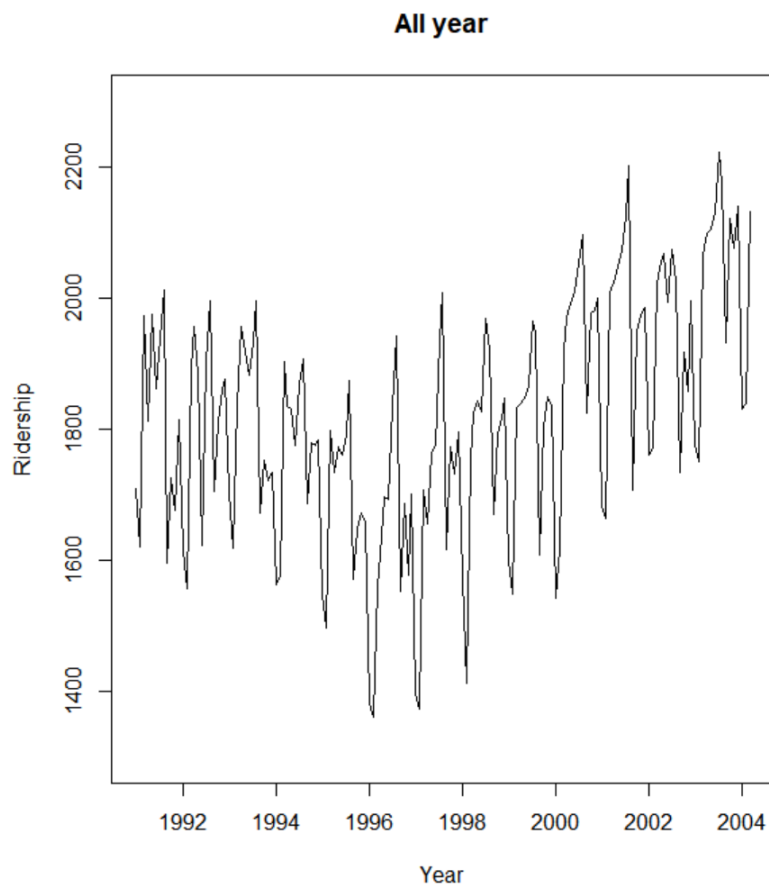
X 변수는 왼쪽으로
Y 변수는 아래쪽으로
치우쳐져 있음

=> log 변환

EDA / 시각화

03 차트조절

>> 확대 축소



시계열 데이터

- 시간의 범위가 긴 경우 시간에 대한 Y의 특징을 파악하기 힘들

=> 특정 시기 확대

04

데이터 분할

4-1. 학습/검증/평가 데이터

4-2. 교차 검증

01 데이터 분할 학습/검증/평가 데이터

» 데이터 정의



학습 데이터

처음 모델을 학습시키기 위해서
필요한 데이터



검증 데이터

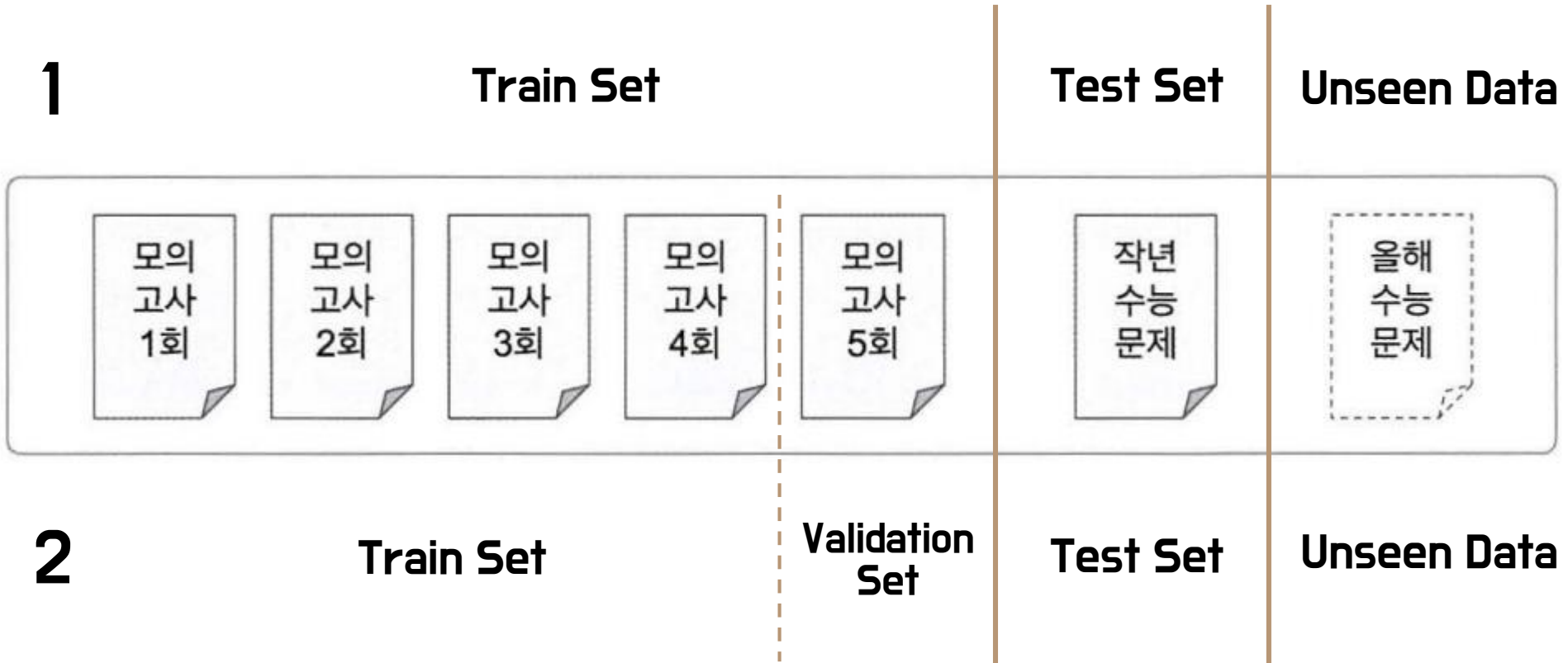
만들어진 모델의 예측력을
평가하는 데이터
예측 결과를 통해 모델 수정



평가 데이터

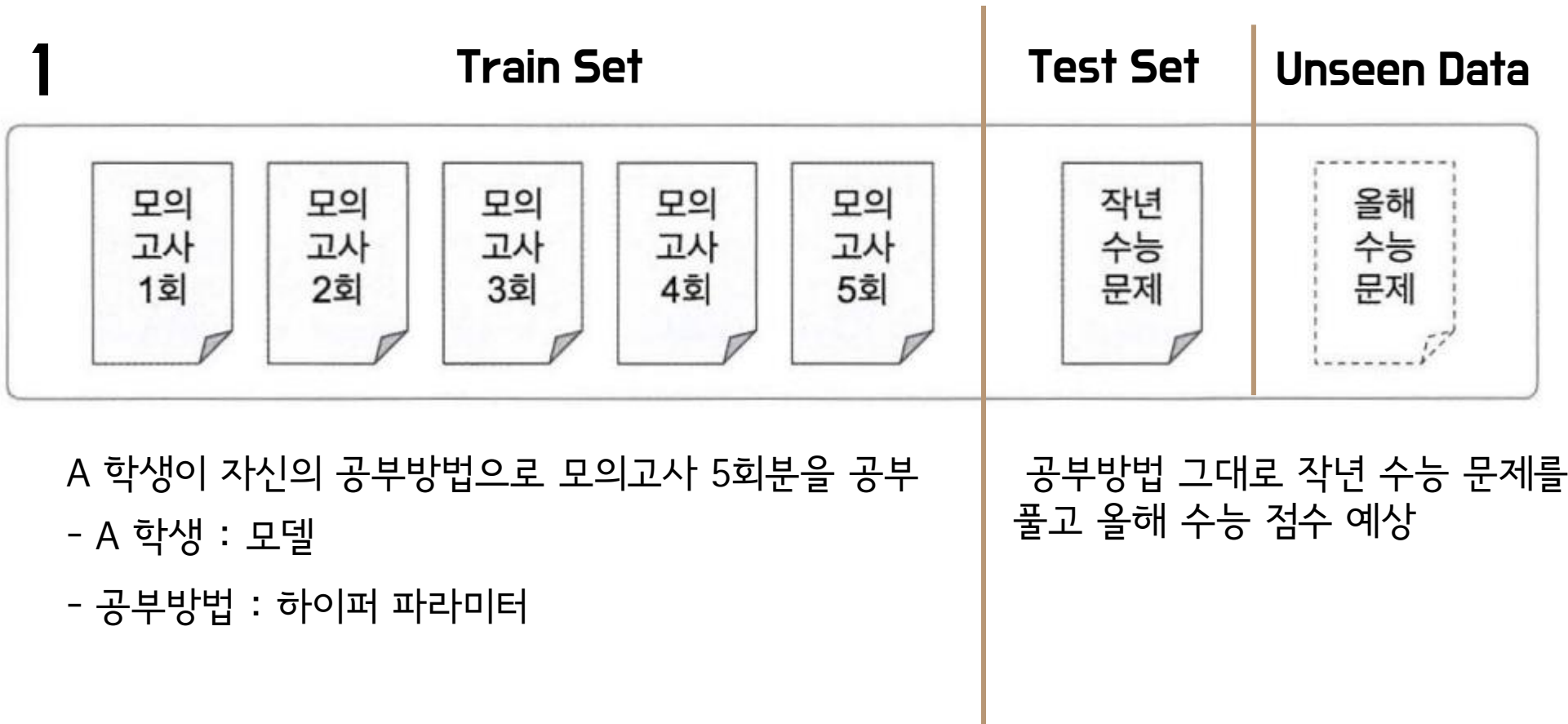
최종 모델의 성능 파악

01 데이터 분할 학습/검증/평가 데이터



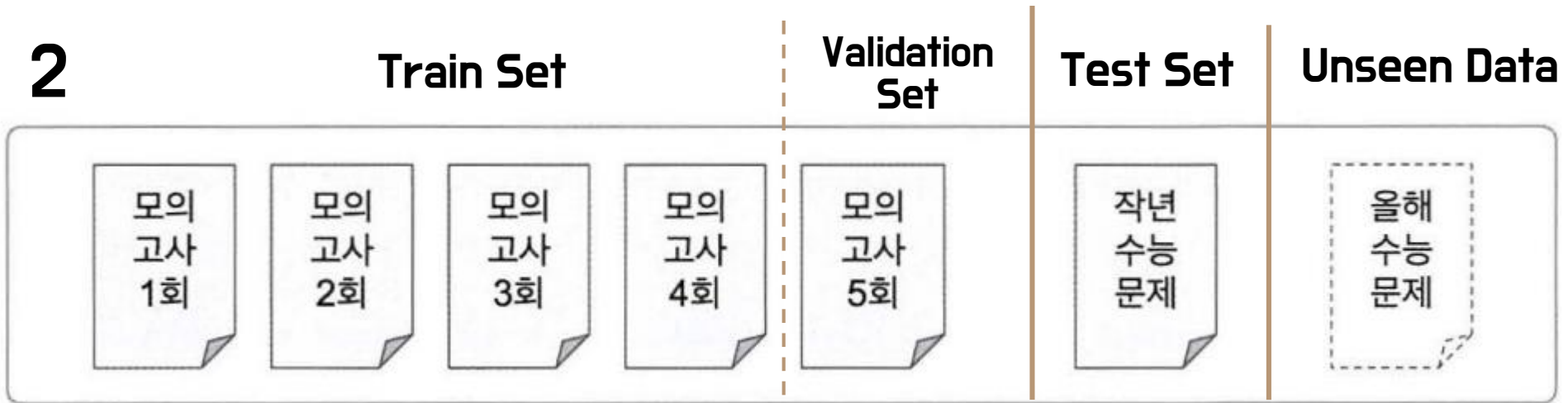
01 데이터 분할

학습/검증/평가 데이터



01 데이터 분할

학습/검증/평가 데이터



A, B, C 학생이 자신의 공부방법으로 모의고사 4회분을 공부

+

각 학생 모두 자신의 공부방법이 맞는지
확인해 보고 싶다!

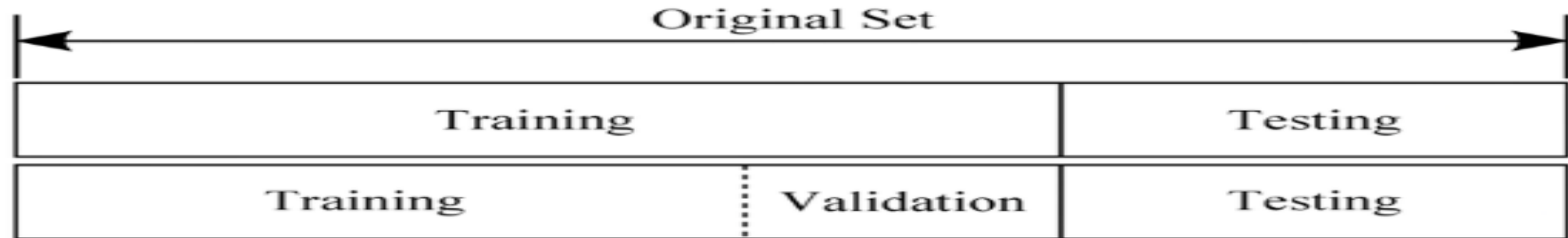


5회차 점수로
가장 성적이
좋은 학생과
공부방법 선정



5회차 점수로 선정된 학생이 같은
공부방법으로 작년 수능 문제 풀기

01 데이터 분할 학습/검증/평가 데이터



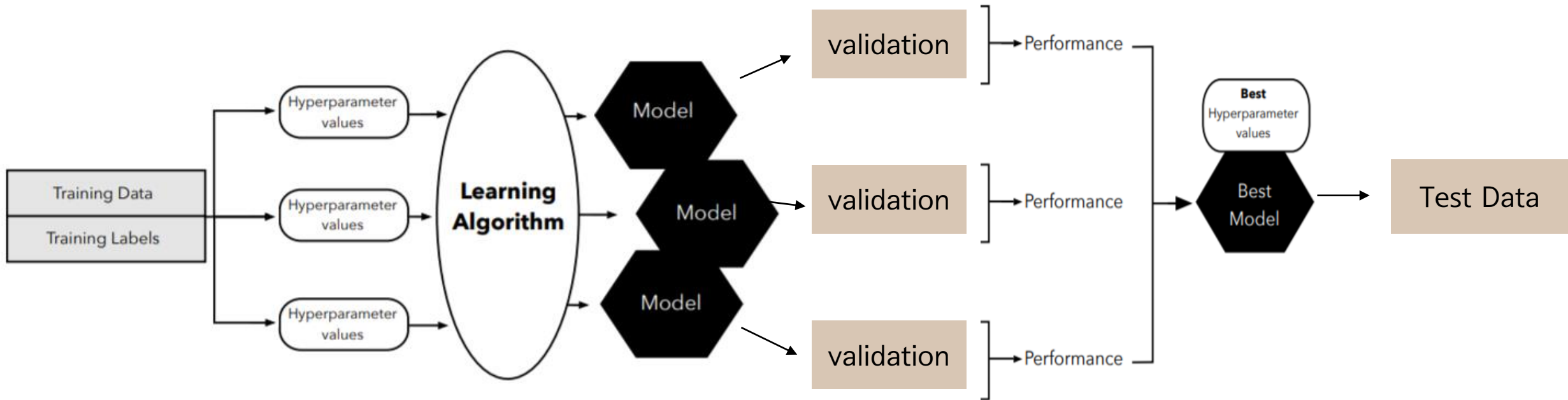
“ 보편적인 데이터 분할 비율

Training 50% / Validation 30% / Test 20%

Training 70% / Validation 20% / Test 10% ”

01 데이터 분할 학습/검증/평가 데이터

» Train & Valid & Test



02 데이터 분할 교차검증

- 정의 : k개로 나누어진 데이터 그룹에 대하여
각 그룹을 한번씩 Train Set으로 지정해주며
k번의 검증결과의 평균을 계산
- 장점 : 샘플링 할 경우의 데이터 편향을 보완
소수의 데이터에서 underfitting 방지
- 단점 : 메모리와 시간관리가 어려움.

» A 학생 '가' 공부방법



05

Hyperparameter 튜닝

5-1. Hyperparameter?

5-2. Grid Search

5-3. Random Search

01 파라미터 튜닝 Hyperparameter?

>> Parameter vs Hyperparameter

Parameter

데이터를 통해 구해지는 값
사용자에 의해서 **조정될 수 없음**
학습된 모델의 일부가 됨.

ex) 회귀분석의 베타 값

Hyperparameter

01 파라미터 튜닝 Hyperparameter?

>> Parameter vs Hyperparameter

Parameter

데이터를 통해 구해지는 값
사용자에 의해서 **조정될 수 없음**
학습된 모델의 일부가 됨.

ex) 회귀분석의 베타 값

Hyperparameter

모델링 시 사용자가 **직접 설정**하는 값
정해진 최적의 값이 없음
- 주로 경험에 의하는 경우가 많음
따라서 **다양한 튜닝 방법**으로 찾음

ex) KNN의 k, 의사결정 나무의 나무개수

01 파라미터 튜닝 Hyperparameter?

>> 하이퍼 파라미터의 영향력

R 내장 데이터 Pima.tr 과 Pima.te로 RandomForest 모델 구축						
나무개수 : 50개, 나무의 변수개수 : 1개				나무개수 : 200개, 나무의 변수개수 : 4개		
	실제					
예측		NO	YES	예측		
	NO	194	50		NO	42
	YES	29	59		YES	67

01 파라미터 튜닝 Hyperparameter?

>> 하이퍼 파라미터의 영향력

R 내장 데이터 Pima.tr 과 Pima.te로 RandomForest 모델 구축

나무개수 : 50개, 나무의 변수개수 : 1개

나무개수 : 200개, 나무의 변수개수 : 4개

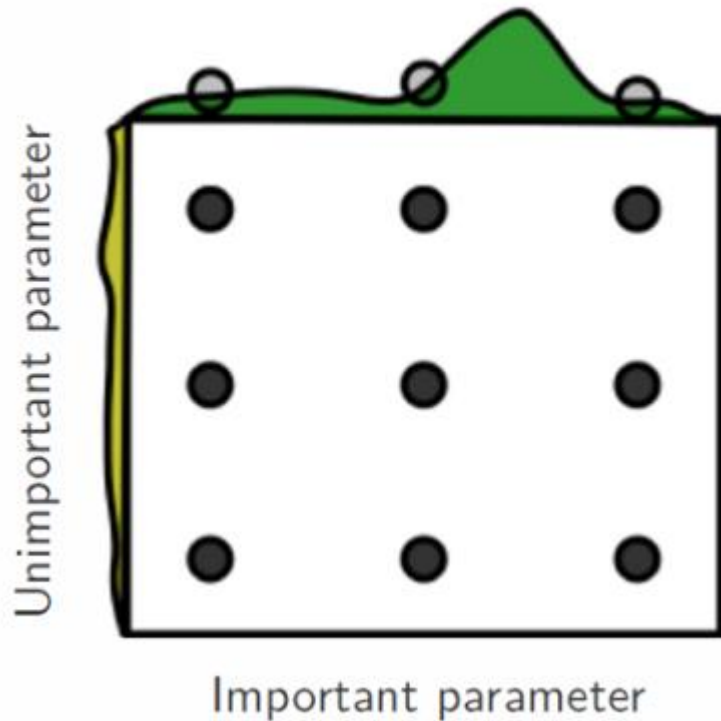
**“데이터에 가장 적합한 파라미터는
어떻게 찾을까?”**

	실제	
예측	NO	YES
NO	194	50
YES	29	59

	실제	
예측	NO	YES
NO	189	42
YES	34	67

02 파라미터 튜닝 Grid Search

Grid Layout



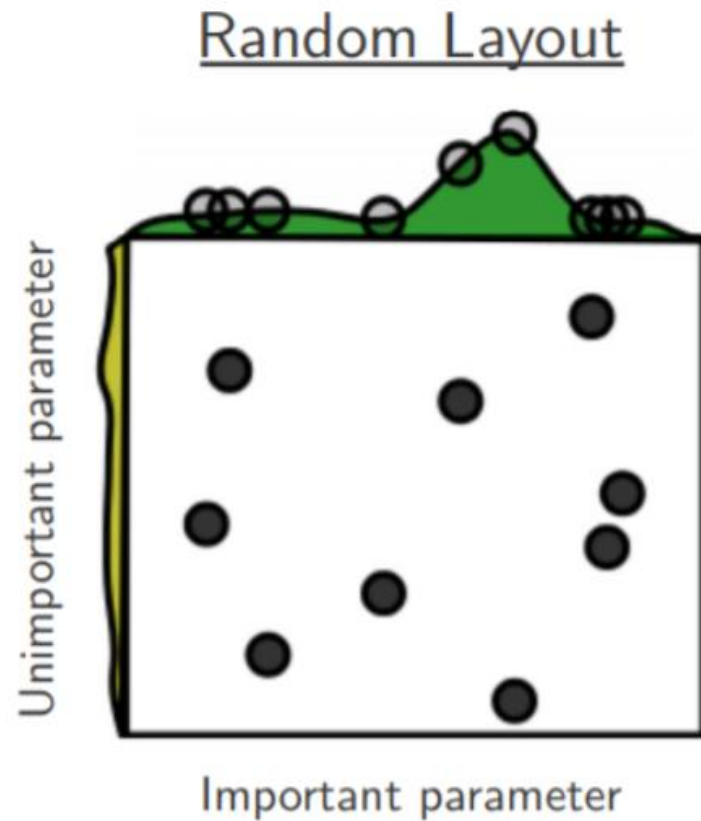
- 하이퍼 파라미터의 **후보군들을 사용자가 지정**
- 사용자가 지정한 조합을 순차적으로 실행한후 성능을 보여줌

변수개수 나무개수	3	5
100	(100,3)	(100,5)
150	(150,3)	(150,5)
200	(200,3)	(200,5)

ex) 6개의 사용자 지정 파라미터로 모델 생성 후 성능 비교

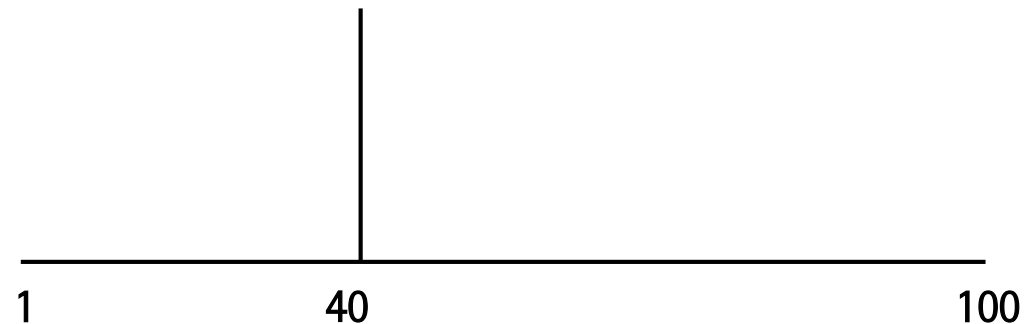
파라미터 튜닝

03 Random Search



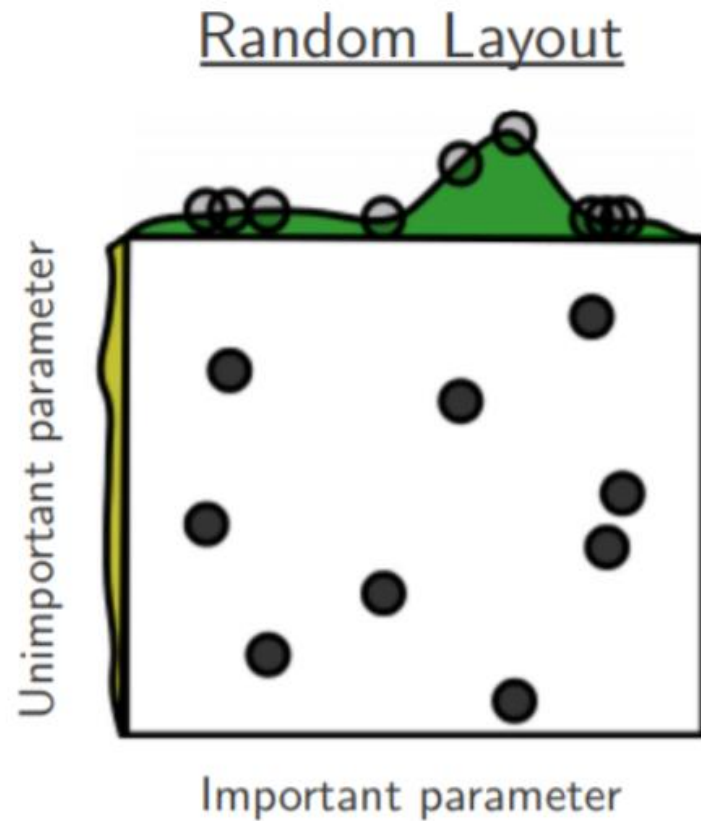
- 하이퍼 파라미터의 **범위만 지정**
- 범위 안에서 무작위로 조합을 만들고, error가 점점 작은 쪽으로 찾아감.
- Grid Search 보다 **시간자원 절약**

ex) `range(1,100)` 이고 무작위로 선정된 것이 40
 - 40의 error 계산 후, 적절한 이웃 (60)선택
 - 40과 60의 error 값 비교하여 적은 쪽으로
 새로운 조합 무작위선정



파라미터 튜닝

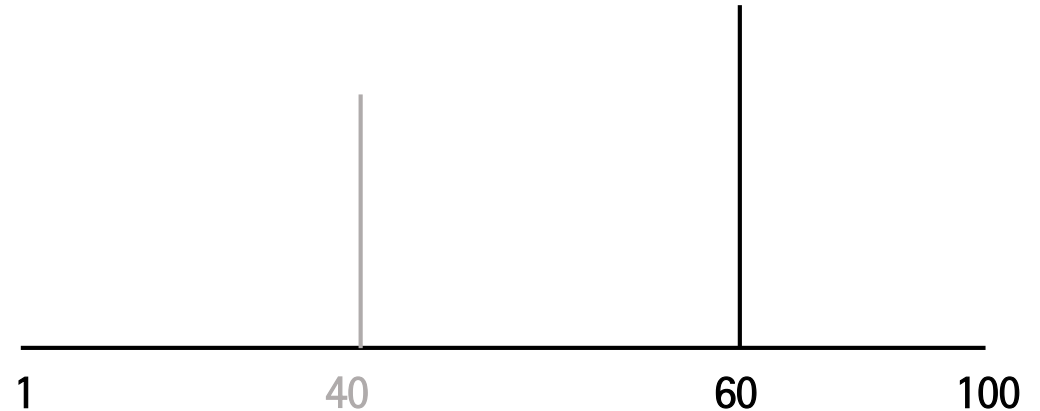
03 Random Search



- 하이퍼 파라미터의 **범위만 지정**
- 범위 안에서 무작위로 조합을 만들고, error가 점점 작은 쪽으로 찾아감.
- Grid Search 보다 **시간자원 절약**

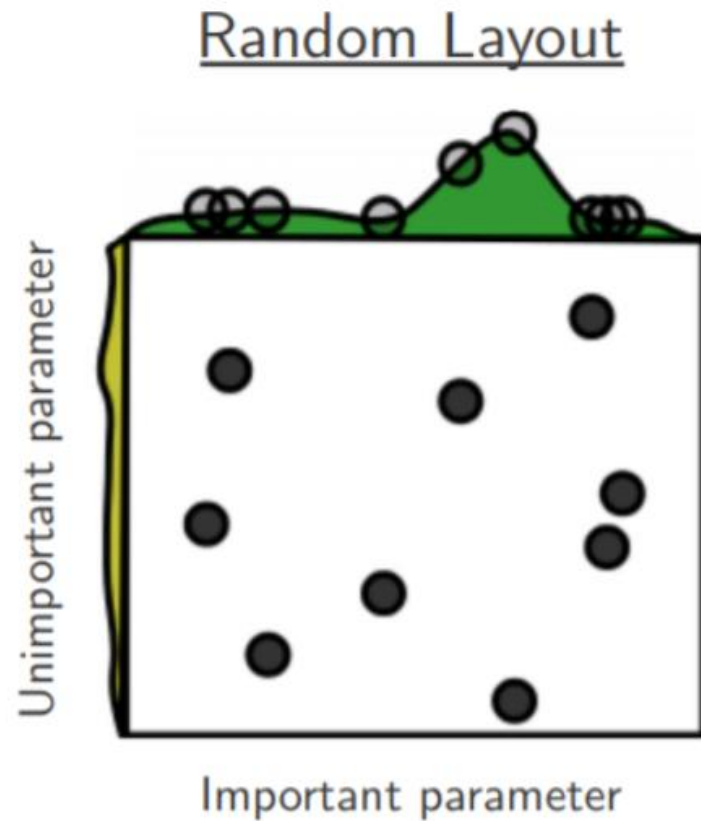
ex) `range(1,100)` 이고 무작위로 선정된 것이 40

- 40의 error 계산 후, 적절한 이웃 (60)선택
- 40과 60의 error 값 비교하여 적은 쪽으로 새로운 조합 무작위선정



파라미터 튜닝

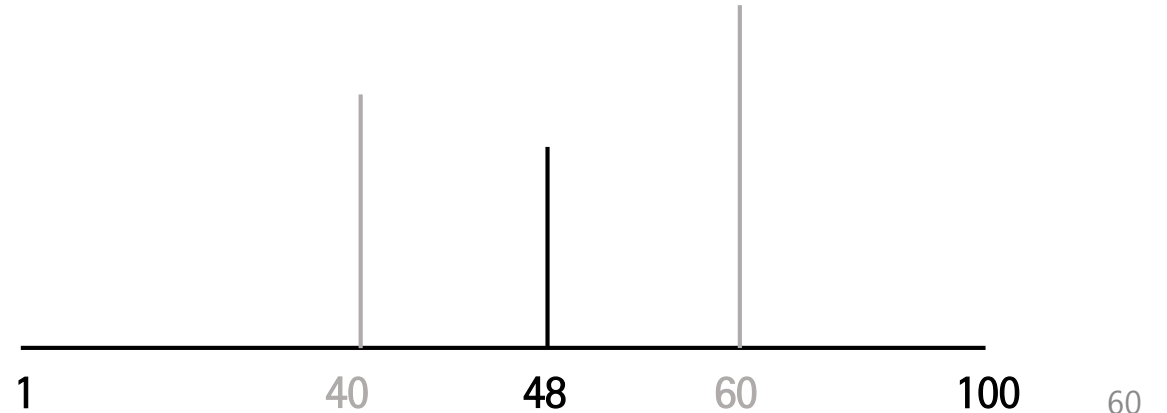
03 Random Search



- 하이퍼 파라미터의 **범위만 지정**
- 범위 안에서 무작위로 조합을 만들고, error가 점점 작은 쪽으로 찾아감.
- Grid Search 보다 **시간자원 절약**

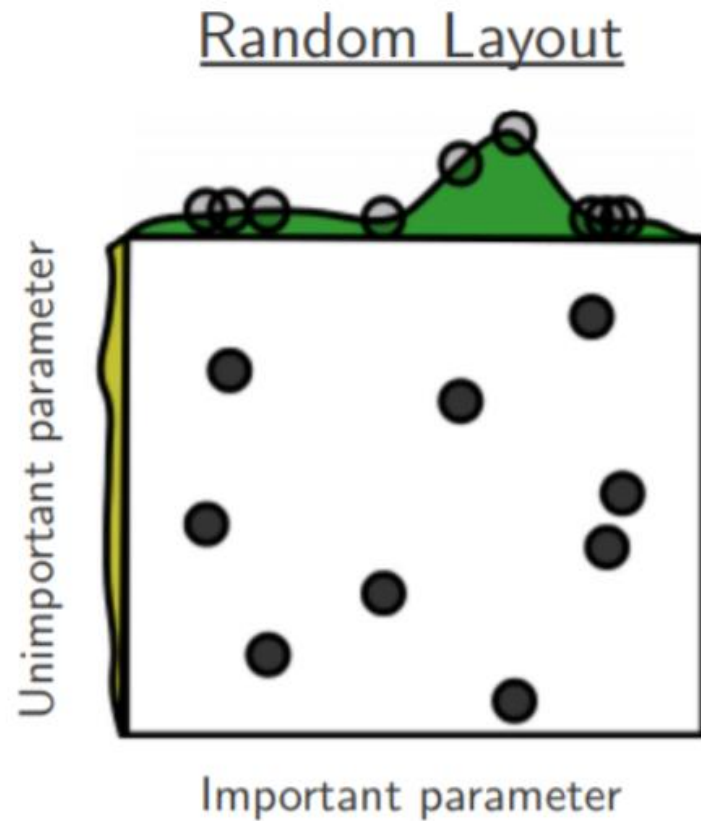
ex) `range(1,100)` 이고 무작위로 선정된 것이 40

- 40의 error 계산 후, 적절한 이웃 (60)선택
- 40과 60의 error 값 비교하여 적은 쪽으로 새로운 조합 무작위선정



파라미터 튜닝

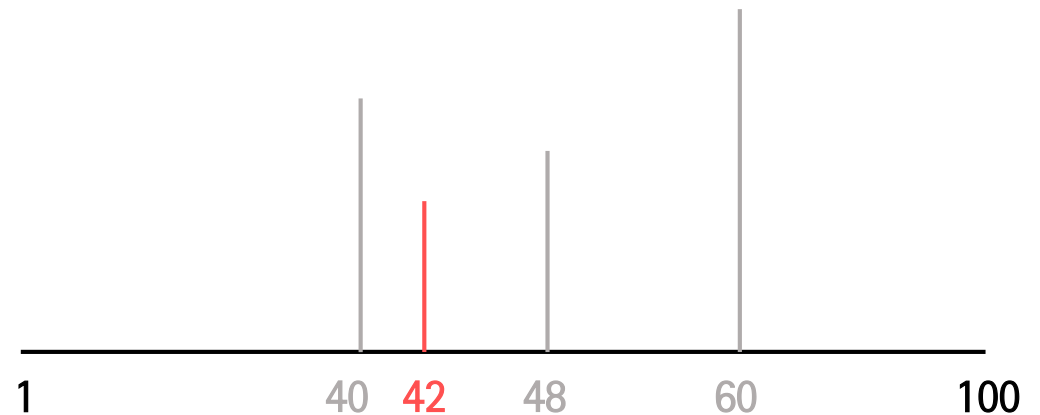
03 Random Search



- 하이퍼 파라미터의 **범위만 지정**
- 범위 안에서 무작위로 조합을 만들고, error가 점점 작은 쪽으로 찾아감.
- Grid Search 보다 **시간자원 절약**

ex) `range(1,100)` 이고 무작위로 선정된 것이 40

- 40의 error 계산 후, 적절한 이웃 (60)선택
- 40과 60의 error 값 비교하여 적은 쪽으로 새로운 조합 무작위선정



06

모델 평가

6-1. 연속형 종속변수

6-2. 이진형 종속변수

6-3. 과적합

01 모델 평가 연속형 종속변수

$$e_i = y_i - \hat{y}_i$$

1. 평균오차 $1/n \sum_{i=1}^n e_i$: 오차에 대한 평균(0으로 수렴할 가능성)
2. MAE(절대평균 오차) $1/n \sum_{i=1}^n |e_i|$: 오차가 음인지 양인지 판단 어려움
3. MPE(평균 백분율 오차) $100 \times 1/n \sum_{i=1}^n e_i/y_i$: 예측이 실제값에서 벗어난 퍼센트
4. MAPE $100 \times 1/n \sum_{i=1}^n |e_i/y_i|$: 예측이 실제값에서 벗어난 퍼센트 + 절대값
5. RMSE $\sqrt{1/n \sum_{i=1}^n e_i^2}$: 오차를 제곱하여 평균한 값의 제곱근,
가장 보편적으로 쓰이는 평가지표

01 모델 평가 연속형 종속변수

$$e_i = y_i - \hat{y}_i$$

1. 평균오차 $1/n \sum_{i=1}^n e_i$: 오차에 대한 평균(0으로 수렴할 가능성)
2. MAE(절대평균 오차) $1/n \sum_{i=1}^n |e_i|$: 오차가 음인지 양인지 판단 어려움
3. MPE(평균 백분율 오차) $100 \times 1/n \sum_{i=1}^n e_i/y_i$: 예측이 실제값에서 벗어난 퍼센트
4. MAPE $100 \times 1/n \sum_{i=1}^n |e_i/y_i|$: 예측이 실제값에서 벗어난 퍼센트 + 절대값
5. RMSE $\sqrt{1/n \sum_{i=1}^n e_i^2}$: 오차를 제곱하여 평균한 값의 제곱근,
가장 보편적으로 쓰이는 평가지표

“이상치들의 영향을 많이 받게 됨”

모델 평가 02 이진형 종속변수

>> 정오행렬(Confusion Matrix)

- 학습데이터로 구축한 모델을 이용하여 검증데이터의 Y 예측

예측 \ 실제	0(NO)	1(YES)
0	TN (True Negative)	FN (False Negative)
1	FP (False Positive)	TP (True Positive)

- TN과 TP가 많을수록 정확한 예측
- **정확도** : 전체 중에서 정확하게 분류한 비율

$$Accuracy = 1 - \frac{TN + TP}{TN + TP + FN + FP}$$

모델 평가 02 이진형 종속변수

>> Imbalanced Data

- 정상인 990명과 암환자 10명

예측 \ 실제	0(NO)	1(YES)
0	990	10
1	0	0

- 정확도 : $990/1000 = 99\%$
- 정확도는 높지만 **무의미한 모델**
 - 암일 때 암이라고 진단할 정확도가 0%...

02 이진형 종속변수

>> 정오행렬(Confusion Matrix)

- 학습데이터로 구축한 모델을 이용하여 검증데이터의 Y 예측

예측 \ 실제	0(NO)	1(YES)
0	TN (True Negative)	FN (False Negative)
1	FP (False Positive)	TP (True Positive)

- **정밀도** : 모델이 1이라고 예측 했을 때 실제 1일 확률(신중한 판단)

$$Precision = \frac{TP}{TP + FP}$$

ex) 사망선고를 내릴 때,
그 사망선고는 반드시 정확해야함

02 이진형 종속변수

>> 정오행렬(Confusion Matrix)

- 학습데이터로 구축한 모델을 이용하여 검증데이터의 Y 예측

예측 \ 실제	0(NO)	1(YES)
0	TN (True Negative)	FN (False Negative)
1	FP (False Positive)	TP (True Positive)

- **재현율** : 실제 값이 1일 때,
모델이 1이라고 예측할 확률(놓치지말것)

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

ex) 암 진단할 때,
진짜 암환자는 모두 암으로 진단해야함

모델 평가 02 이진형 종속변수

>> Recall과 Precision의 반비례관계

실제값	0	0	0	0	0	1	1	1	1	1
예측된 1일 확률	0.5	0.5	0.6	0.4	0.4	0.5	0.6	0.6	0.6	0.4

실제 \ 예측	0(NO)	1(YES)
0	4	2
1	1	3

- Precision : 3/4
- Recall : 3/5

모델이 1이라고 예측하면 75%의 확률로 1일 가능성이 있음

ex) 암환자로 진단 내렸을 때,
진짜 암 환자일 확률

확률 0.6 이상을 1이라고 할 때

모델 평가 02 이진형 종속변수

» Recall과 Precision의 반비례관계

실제값	0	0	0	0	0	1	1	1	1	1
예측된 1일 확률	0.5	0.5	0.6	0.4	0.4	0.5	0.6	0.6	0.6	0.4

- Precision : 5/10
- Recall : 5/5

앞보다는 1의 검출력이 좋음
하지만, 그만큼 거짓된 1을 잡아낼 확률
도 높아짐.

ex) 암 환자에게 암 진단을 내릴 확률

실제 \ 예측	0(NO)	1(YES)
0	0	0
1	5	5

확률 0.4 이상을 1이라고 할 때

모델 평가 02 이진형 종속변수

》 F1 Score

- 정상인 150명과 암환자 50명

예측 \ 실제	0(NO)	1(YES)
0	100	10
1	50	50

- F1 Score**: Recall과 Precision의 조화평균
 - 큰 값에 영향을 덜 받기 때문에,
imbalanced Data 평가 지표로 적합

$$2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- precision : $50/100 = 0.5$
- recall : $10/50 = 0.2$
- f1 : $2 * (0.1/0.7) = 2/7$
- accuracy : $150/200 = 0.75$

03 모델 평가 과적합

>> 현재 데이터에 “과”하게 “적합”되다 = “과적합”

- 모델 구축의 목적

- 현재 데이터 정확하게 예측 < 미래 데이터 정확하게 예측
- 현재의 데이터를 매우 정확하게 예측하게 된다면?

ex) 기부금액 예측

(가정: 키와 기부금액은 서로 독립적)

- 현재의 데이터에 사람의 키 추가시 예측 오차 감소
- 하지만, 새롭게 수집될 데이터는 이 경향이 나타나지 않을 것 -> 과적합

- 모델이 고차원일수록 위험성이 커짐

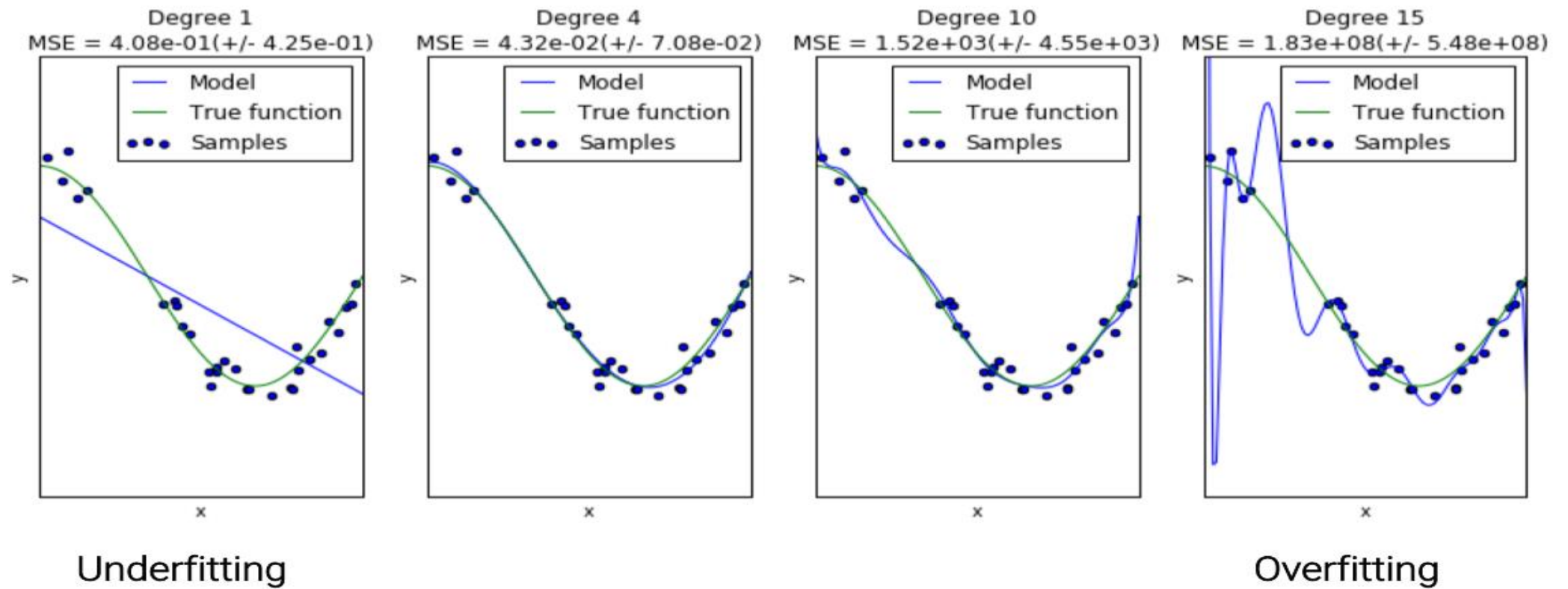
ex) 기부금액 예측

- 소득단위가 100개 단위로 구성
- 100개의 더미 변수들이 생성되어 고차원 모델 구축

03 모델 평가

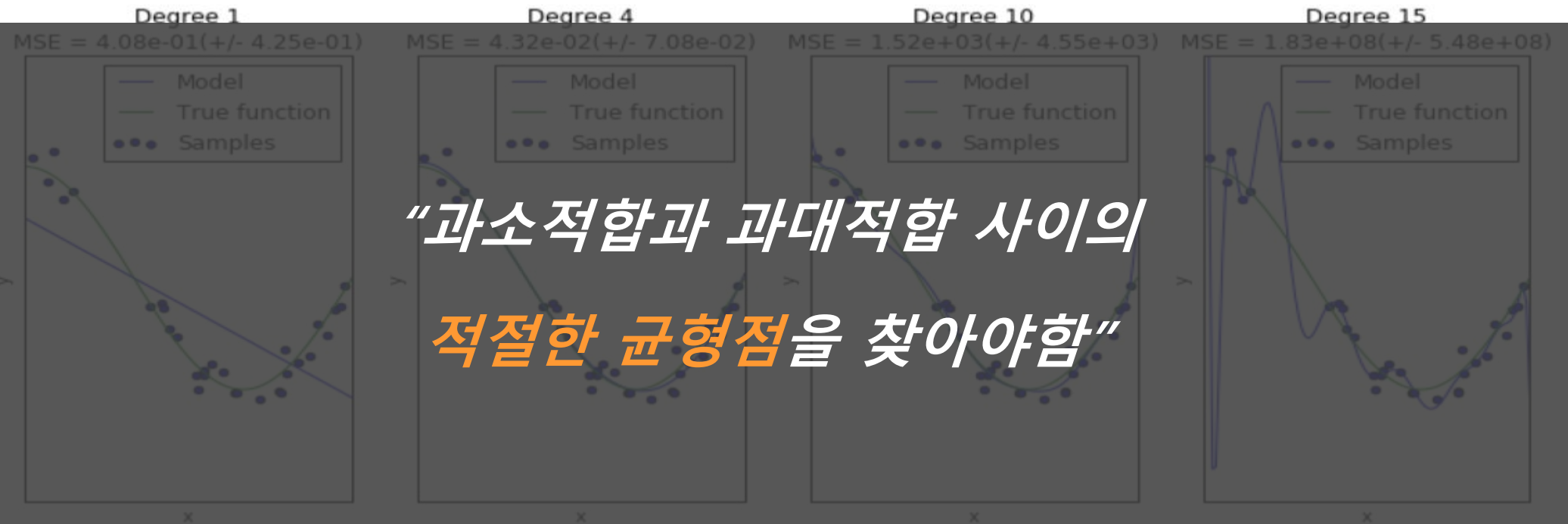
과적합

모델의 차원이 올라감



03 모델 평가 과적합

모델 학습 강도가 올라감

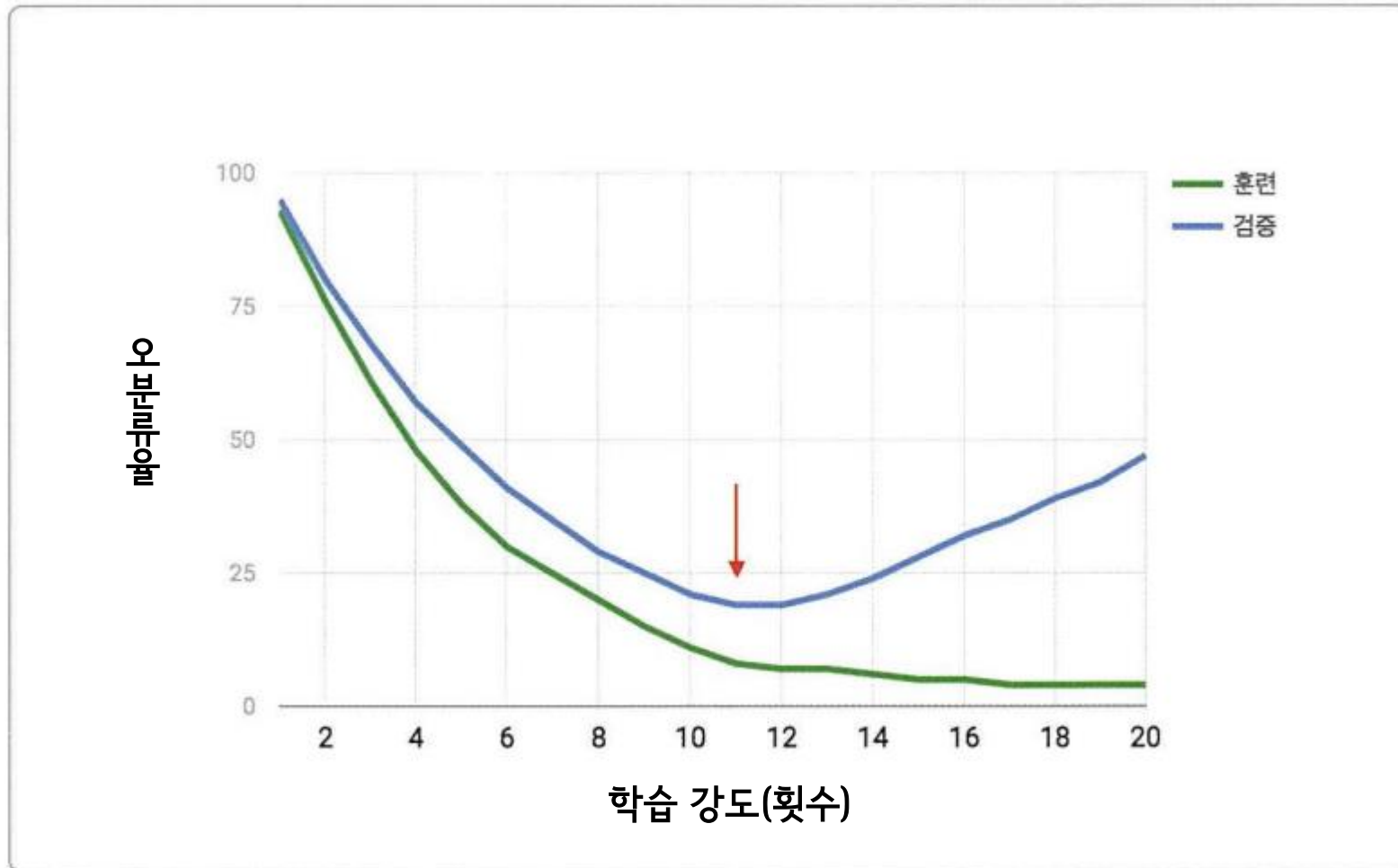


“과소적합과 과대적합 사이의
적절한 균형점을 찾아야함”

Underfitting

Overfitting

03 모델 평가 과적합



Appendix

오늘의 실습

향후 학습 가이드

I. 머신러닝을 공부하는 방법

머신러닝 공부는 코드도 물론 중요하지만, 원리와 이론을 알고 사용해야 의미가 있다고 생각합니다. 코드는 구글링해서 금방 돌릴 수 있어요. 하지만 그 기법이 어떻게 돌아가는지 알고 자신의 데이터에 맞게 하이퍼파라미터나 다양한 값들을 조정해줄 수 있는게 남들보다 앞서나갈 수 있는 경쟁력이 될 것입니다 ☺

물론 코드도 돌려봐야 자기 것이 되겠죠? 이론 공부 후에 코드를 돌려가면서 하이퍼파라미터도 이것저것 해보고 많은 시간을 투자해서 두들겨본다면 머리속에 오래 남을 것이라고 생각합니다!

향후 학습 가이드

II. 앞으로의 발표에서 꼭 넣어야 하는 내용

1. 모형의 학습 과정 알고리즘과 원리를 꼭 설명해주세요.
2. 코드 작성에도 신경을 써주세요. 데이터 예제를 선정한 후, 하이퍼파라미터 튜닝 과정을 꼭 담아주세요.
3. 모형의 특징과 주로 사용 되는 곳에 대한 내용도 조사하여 발표해주세요.

P.S : 이 내용은 [카페](#)에도 게시하였습니다 ☺



비타민의 무궁한 발전을 위하여... 협조 부탁드립니다 ^0^





R을 활용한 머신 러닝 기초

알고리즘을 학습하기 전에 알아야할 기초개념

- 끝 -