데이터마이닝팀

4팀
오주원
이동기
김형석
이경미
최종혁

INDEX

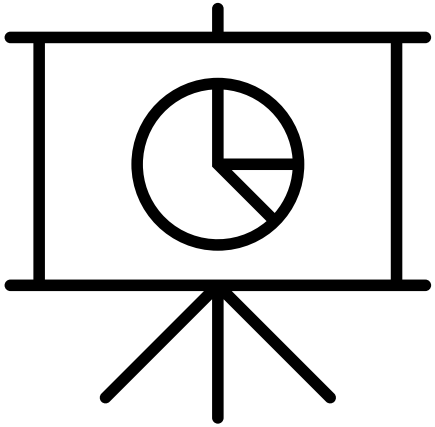
1

데이터마이닝 소개

1 데이터마이닝 소개

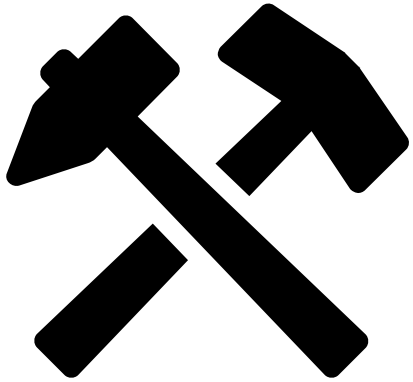
정의

Data (데이터)



\+

Mining (채굴)



데이터로 쌓여있는 산에서 **유용한 데이터**를 **추출**하는 과정

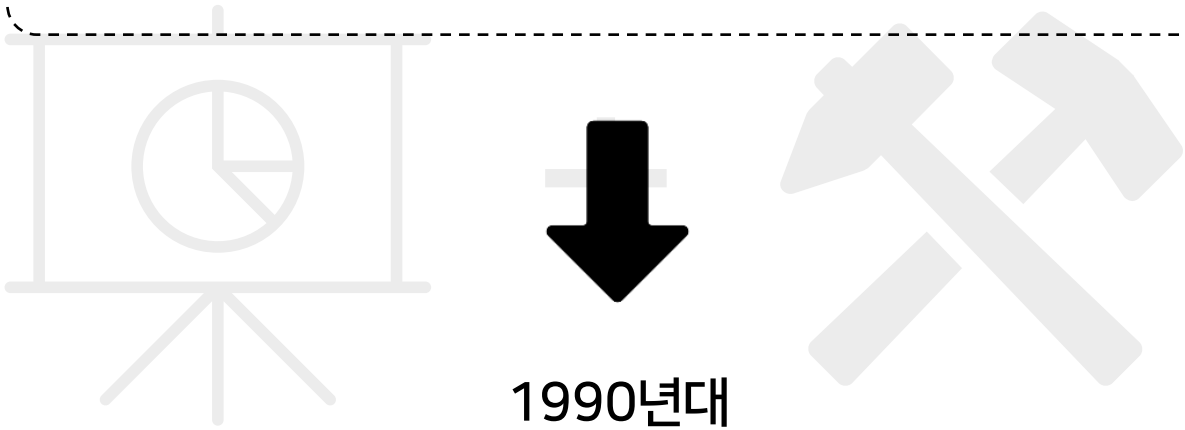
1 데이터마이닝 소개

정의

1930년대

Data (데이터) Mining (채굴)

데이터베이스의 **효율적인 저장**



1990년대

데이터를 통한 **지식 발견**(Knowledge Discovery)

1 데이터마이닝 소개

정의

Exploration

데이터를 분석에 용이한 형태로 전처리

Pattern Identification

분류나 회귀 등의 방법으로 데이터로부터 패턴 발견

Deployment

찾아낸 패턴을 활용하여 목적에 맞는 결과물 도출

1 데이터마이닝 소개

정의

Exploration

데이터를 분석에 용이한 형태로 전처리

Pattern Identification

대량의 데이터로부터 **유용한 정보와 패턴을 추출**해내는 일련의 과정

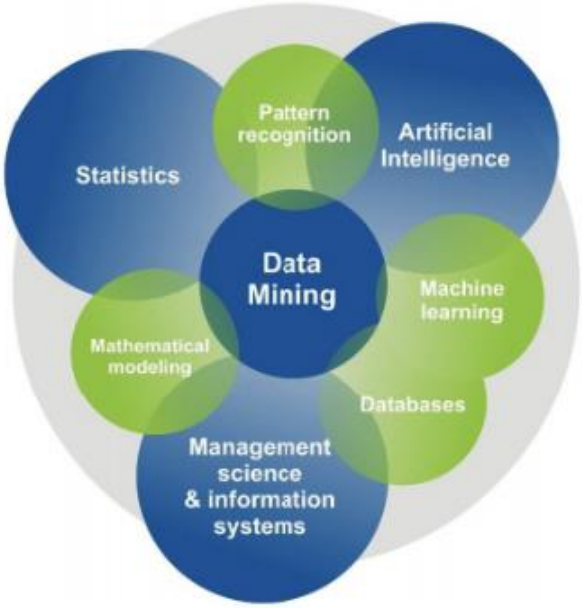
분류나 회귀 등의 방법으로 데이터로부터 패턴 발견

Deployment

찾아낸 패턴을 활용하여 목적에 맞는 결과물 도출

1 데이터마이닝 소개

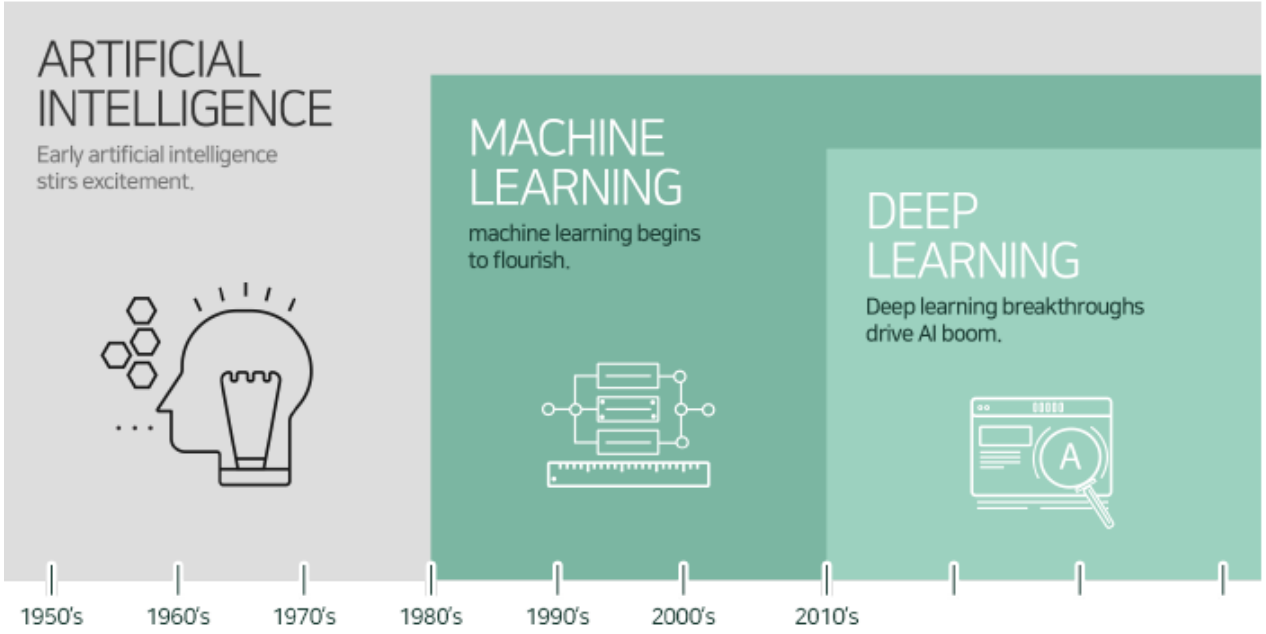
학제적 위치



패턴인식, 통계학, 머신러닝, 인공지능 등과 같은

여러 학문을 넘나드는 **간학문적이고 융합적인** 분야

인공지능 vs 머신러닝 vs 딥러닝



인공지능, 머신러닝, 딥러닝의 개념을 구분할 필요가 있음

1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝

인공지능

사람의 일을 기계가 대신하는 모든 자동화 과정을 일컫는 개념

머신러닝과 딥러닝을 모두 포괄함

머신러닝

명시적인 프로그래밍 없이 기계가 스스로 패턴을 학습하는 방법

컴퓨터가 알아서 데이터를 학습해 결과 출력

딥러닝

머신러닝 알고리즘 중 인공신경망을 기반으로 , 사람의 신경망과 유사한 학습 체계를 구축해 목적 달성을 위한 과정을 수행하는 방법

1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝

인공지능

사람의 일을 기계가 대신하는 모든 자동화 과정을 일컫는 개념

머신러닝과 딥러닝을 모두 포괄함

머신러닝

명시적인 프로그래밍 없이 **기계가 스스로 패턴을 학습**하는 방법

컴퓨터가 알아서 데이터를 학습해 결과 출력

딥러닝

머신러닝 알고리즘 중 인공신경망을 기반으로 , 사람의 신경망과 유사한 학습 체계를

구축해 목적 달성을 위한 과정을 수행하는 방법

1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝

인공지능

사람의 일을 기계가 대신하는 모든 자동화 과정을 일컫는 개념

머신러닝과 딥러닝을 모두 포괄함

머신러닝

명시적인 프로그래밍 없이 기계가 스스로 패턴을 학습하는 방법

컴퓨터가 알아서 데이터를 학습해 결과 출력

딥러닝

머신러닝 알고리즘 중 **인공신경망을 기반**으로,

사람의 신경망과 유사한 학습 체계를 구축해 학습하는 방법

1 데이터마이닝 소개

딥러닝 < 머신러닝 < 인공지능 순서로 범위 확장

데이터마이닝은 머신러닝과 과정은 비슷하지만, 머신러닝은 예측에 집중하는
개념이라면, 데이터마이닝은 **지식 발견**에 초점

인사이트 발견을 위한
데이터마이닝 과정에서
통계학의 역할이 아주 중요

CRISP-DM 방법론

CRISP-DM 방법론

Cross Industry Process for Data Mining

데이터 마이닝의 일반적인 접근 방식을 설명하는 **데이터 분석 프로세스 모델**

6단계로 구성

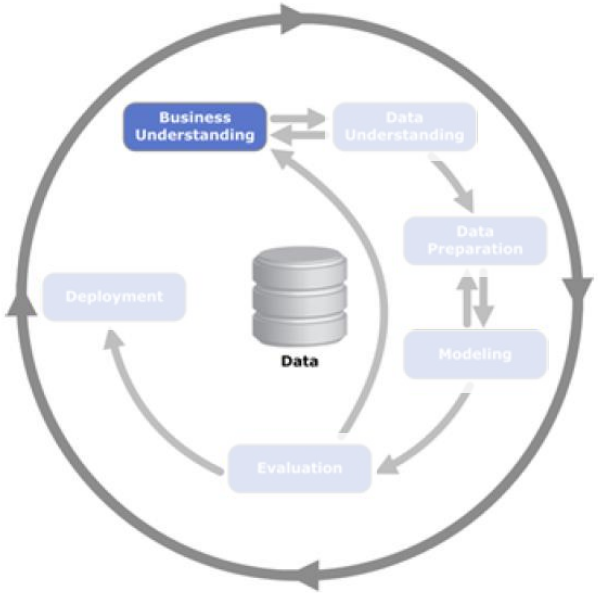
+

단계별 피드백

CRISP-DM 방법론

1단계 : 문제 이해

과제의 **목적**과 **요구사항**을 이해하고 도메인 지식을 활용하여
초기 프로젝트 **계획을 수립**하는 단계



주요 Task

1. 업무 목적 파악
2. 상황 파악
3. 목표 설정
4. 프로젝트 계획 수립

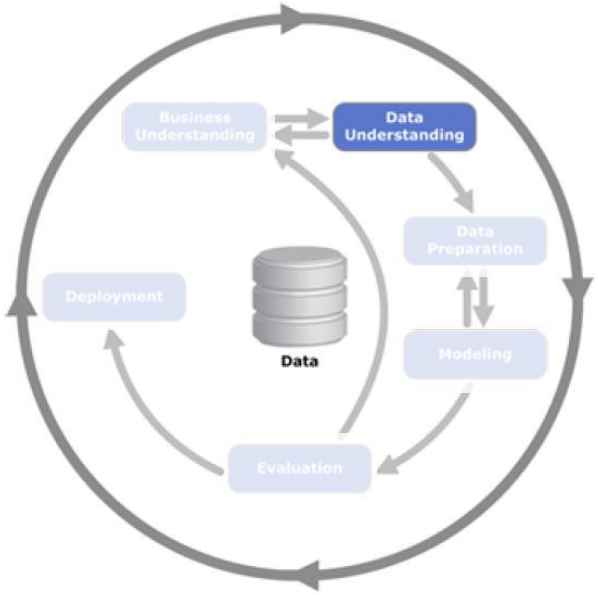
CRISP-DM 방법론

2단계 : 데이터 이해

탐색적 데이터 분석(EDA)를 수행하는 단계

데이터의 통계량, 특징, 관계 등을 하나하나 뜯어보는 단계

이상치와 결측치 확인은 필수



주요 Task

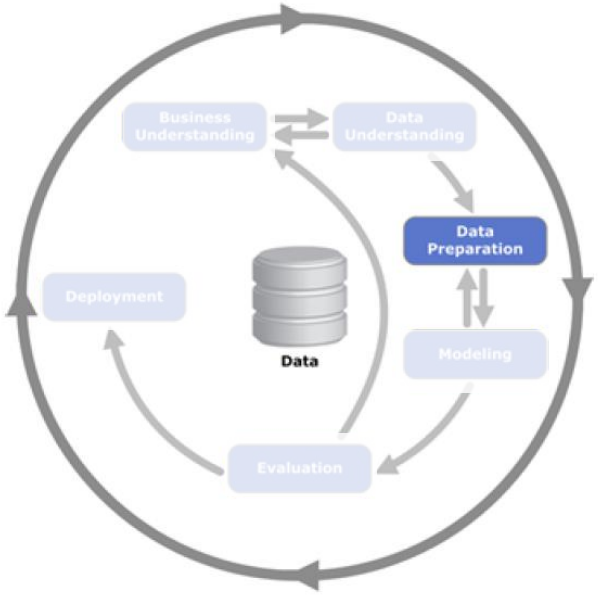
1. 데이터 수집
2. 데이터 기술 분석
3. 데이터 탐색
4. 데이터 품질 확인

CRISP-DM 방법론

3단계 : 데이터 준비

현실 세계의 데이터를 clean하게 만드는 단계

분석 목적에 잘 다듬어주는 것이 중요



주요 Task

1. 분석용 데이터셋 선택
2. 데이터 정제
3. 분석용 데이터셋 편성
4. 데이터 통합 & 포맷팅

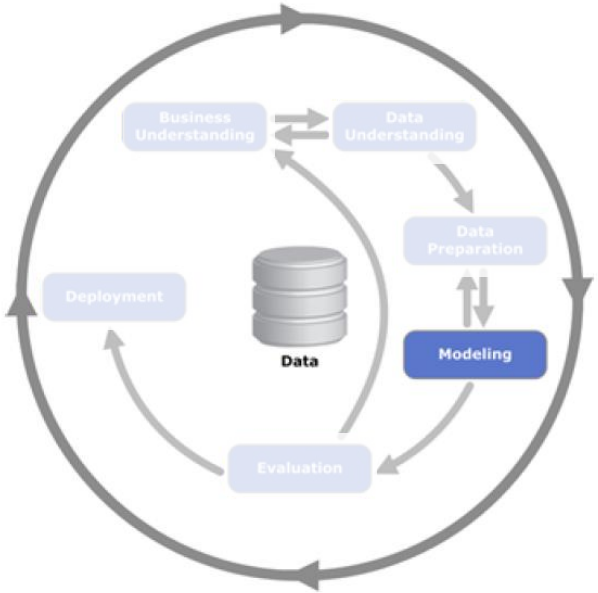
CRISP-DM 방법론

4단계 : 분석 및 모델링

모델링 수행 단계

다양한 알고리즘 선택과 파라미터 최적화

머신러닝과 딥러닝의 여러 통계적 모델 활용



주요 Task

1. 모델링 기법 선택
2. 테스트 계획 설계
3. 모델 작성
4. 모델 평가

CRISP-DM 방법론

5단계 : 평가

모델링의 **성능 평가 단계**

과제의 목적에 알맞은 평가지표를 사용하여 모델 성능 평가

분류의 경우 misclassification rate, F1 score, 회귀의 경우 RMSE, MAE 등이 있음

주요 Task

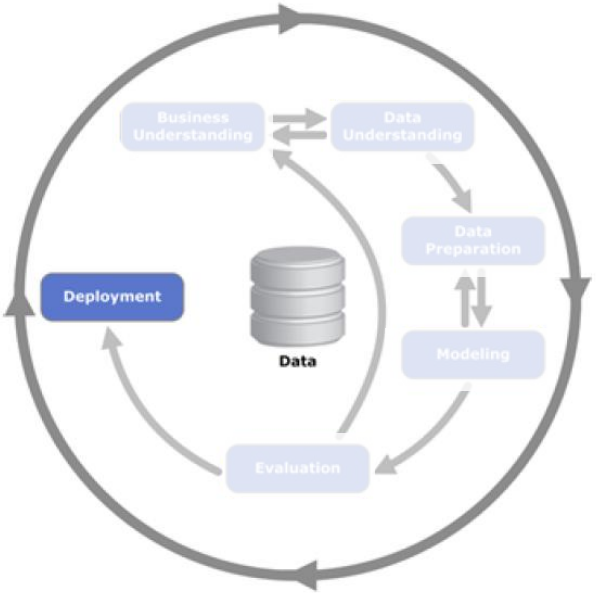
1. 분석 결과 평가
2. 모델링 과정 평가
3. 모델 적용성 평가

CRISP-DM 방법론

6단계 : 전개

현실 적용 단계

실제 현업에 적용하고 유의미한 결론을 이끌어내는 단계



주요 Task

1. 전개 계획 수립
2. 유지보수 계획 수립
3. 프로젝트 리뷰

2

모델링

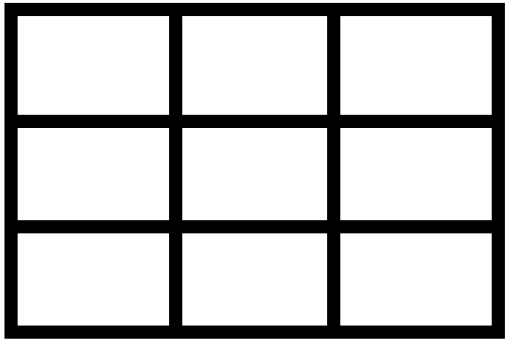
2 모델링

데이터 정의

정형 데이터

미리 정해진 구조에 따라 저장,

테이블 형식 데이터

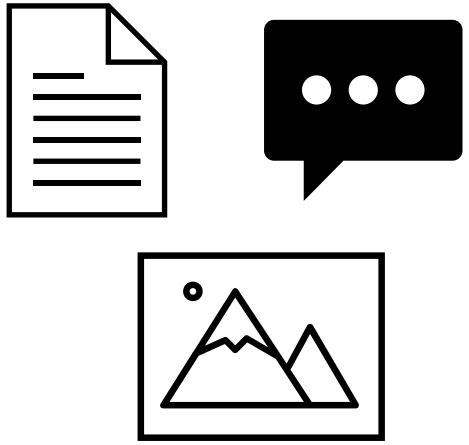


비정형 데이터

구조화되지 않은 형태로 저장,

테이블 형식으로 나타낼 수 없는

데이터



2 모델링

데이터 정의

Sales Price	Neighborhood	Sq. feet	Bedrooms
$ 250,000	Normaltown	2000	3
$ 300,000	Hipsterton	800	2
$ 150,000	Normaltown	850	2
$ 78,000	Normaltown	550	1
$ 150,000	Skid Row	2000	4

종속 변수

예측의 대상이 되는 변수

(Sales Price)

독립 변수

종속 변수를

예측하는데 쓰이는 변수들

(나머지 변수들)

2 모델링

데이터 정의

Sales Price	Neighborhood	Sq. feet	Bedrooms
$ 250,000	Normaltown	2000	3
$ 300,000	Hipsterton	800	2
$ 150,000	Normaltown	850	2
$ 78,000	Normaltown	550	1
$ 150,000	Skid Row	2000	4

[Train data]

Sales Price	Neighborhood	Sq. feet	Bedrooms
???	Hipstertown	2000	3

[Test data]

Train data

모델을
학습시키기 위한 데이터

독립변수와 종속변수가 모두 존재

Test data

학습된 모델을
평가하기 위한 데이터

종속변수가 존재하지 않음, 예측

2 모델링

데이터 정의

Sales Price	Neighborhood	Sq. feet	Bedrooms
$ 250,000	Normaltown	2000	3
$ 300,000	Hipsterton	800	2
$ 150,000	Normaltown	850	2
$ 78,000	Normaltown	550	1
$ 150,000	Skid Row	2000	4

[Train data]

Sales Price	Neighborhood	Sq. feet	Bedrooms
???	Hipstertown	2000	3

[Test data]

Train data

모델을

학습시키기 위한 데이터

독립변수와 종속변수가 모두 존재

Test data

학습된 모델을

평가하기 위한 데이터

종속변수가 존재하지 않음, 예측

⋮

즉, 모델이 unseen data
에서도 잘 작동할지 평가하기 위해 활용

2 모델링

데이터 정의

Train/ Test 분할 시, 몇 가지 규칙에 유의해야 함

Train data의 개수가 Test data의 개수보다 **많아야 함**

Train data와 Test data가 동일한 비율의 클래스 분포를 가져야 함

Train data와 Test data에 중복되는 데이터가 없어야 함

2 모델링

데이터 정의

Train/ Test 분할 시, 몇 가지 규칙에 유의해야 함

Train data의 개수가 Test data의 개수보다 **많아야** 함

Train data와 Test data가 **동일한 비율의 클래스 분포**를 가져야 함

Train data와 Test data에 중복되는 데이터가 없어야 함

2 모델링

데이터 정의

Train/ Test 분할 시, 몇 가지 규칙에 유의해야 함

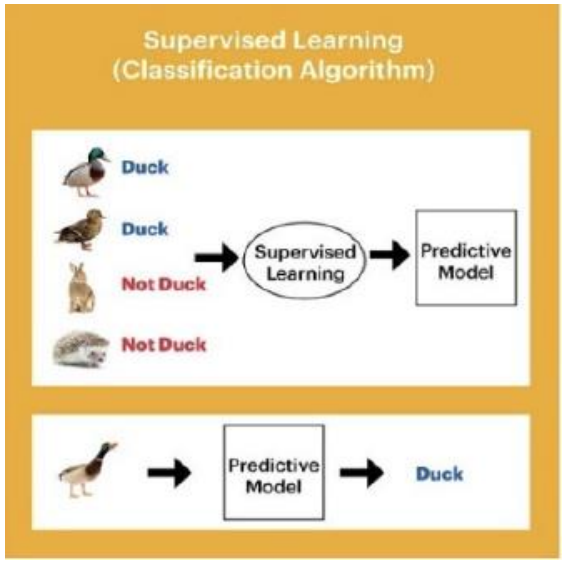
Train data의 개수가 Test data의 개수보다 **많아야 함**

Train data와 Test data가 동일한 비율의 클래스 분포를 가져야 함

Train data와 Test data에 **중복되는 데이터가 없어야 함**

2 모델링

모델링(머신러닝)의 종류 - ① 정답 유무에 따라



지도학습

입력값과 출력값을 제공하여

학습시키는 방법

둘 사이의 관계를 규명하는 것에 초점

➡ 정답을 알려주고 학습

비지도학습

종속변수(y) 없이

입력값들을 구분하도록 학습하는 방법

데이터의 구조를 묘사하고

관계를 해석하는 데 초점

➡ 명시적인 정답 없이 학습

2 모델링

모델링(머신러닝)의 종류 - ① 정답 유무에 따라

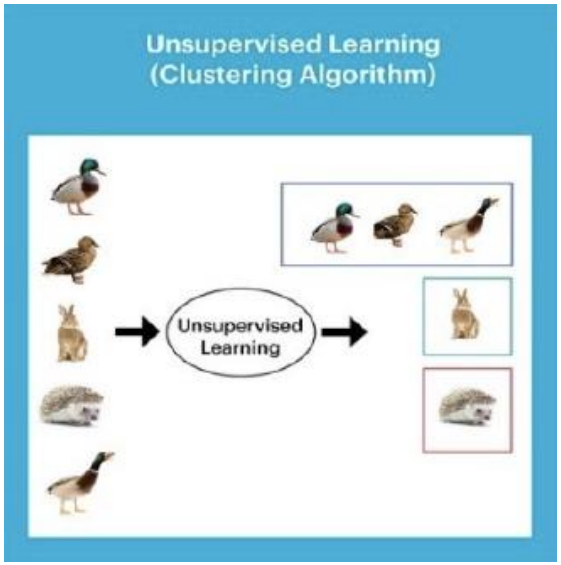
지도학습

입력값과 출력값을 제공하여

학습시키는 방법

둘 사이의 관계를 규명하는 것에 초점

➡ 정답을 알려주고 학습



비지도학습

종속변수(y) 없이

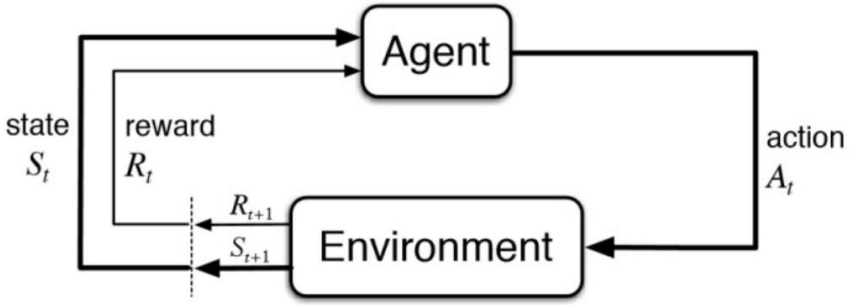
입력값들을 구분하도록 학습하는 방법

데이터의 구조를 묘사하고

관계를 해석하는 데 초점

➡ 명시적인 정답 없이 학습

모델링(머신러닝)의 종류 - ① 정답 유무에 따라



AlphaGo

강화학습

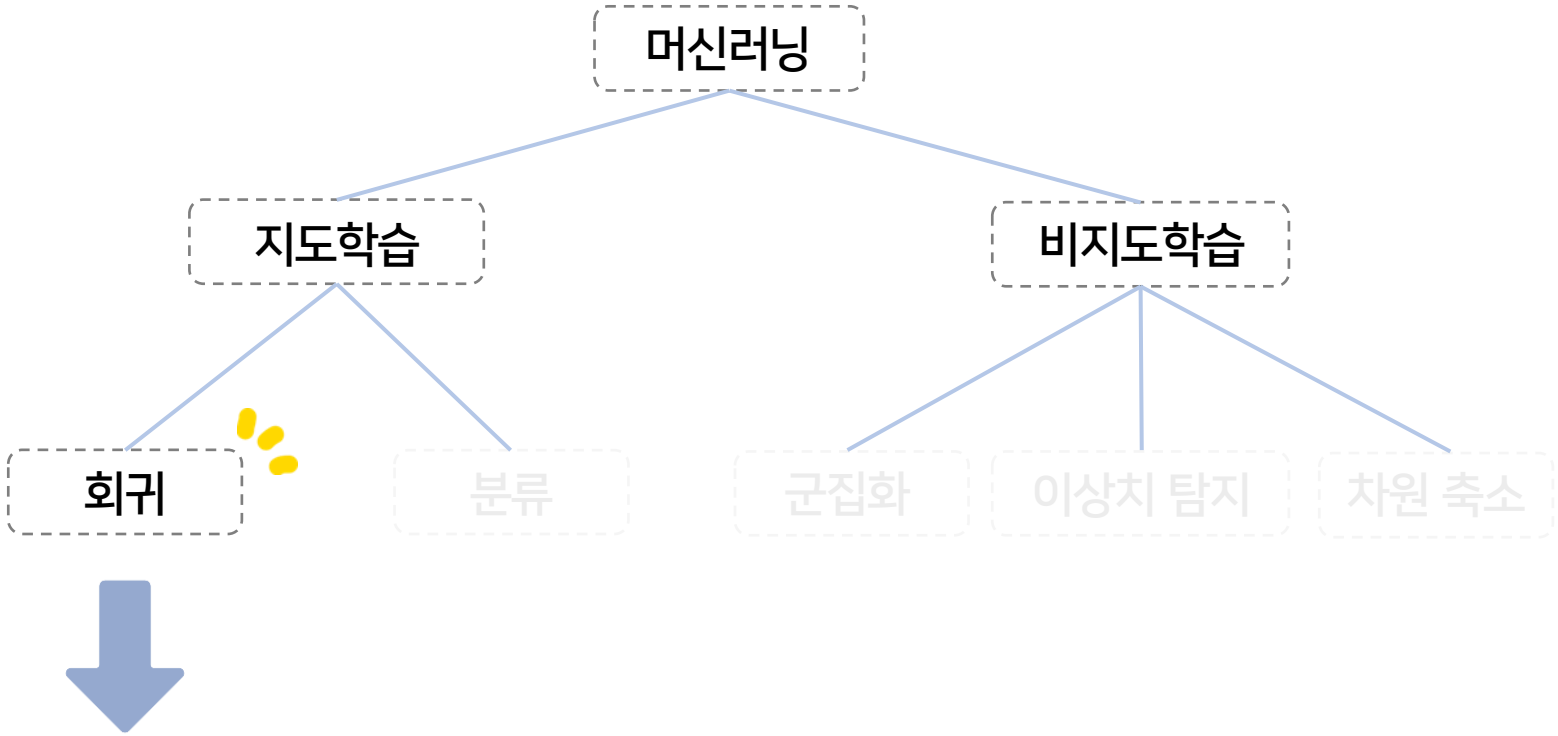
학습 결과에 대한 보상을 최대화하고

페널티를 최소화하는 방향으로 학습하는 방법

본 클린업에서는 다루지 않을 예정!

2 모델링

모델링(머신러닝)의 종류 - ② 학습 목적에 따라

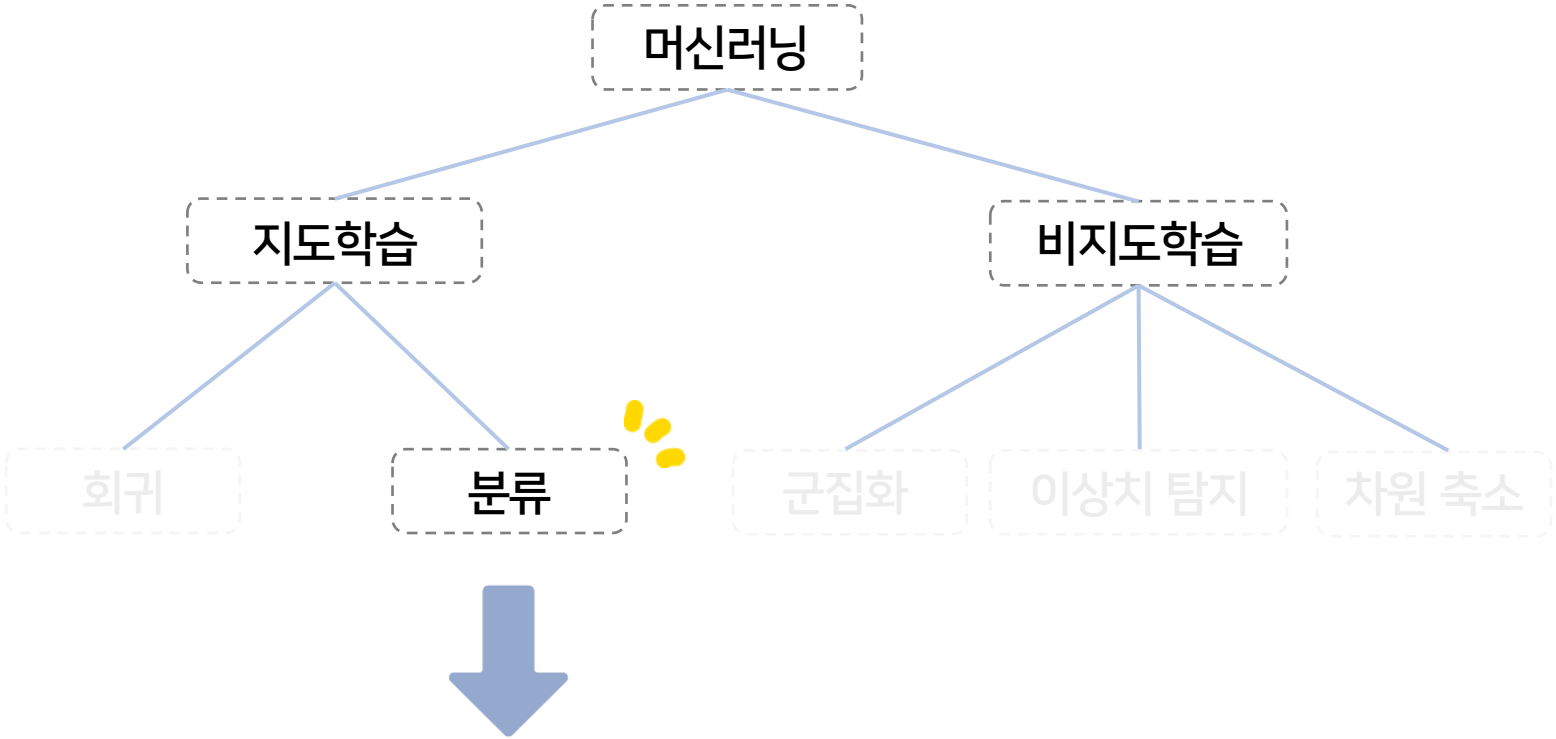


종속변수가 어떤 값을 갖는지 예측하는 것이 목적으로,

연속형 변수를 예측하는 **지도학습 방법론**

2 모델링

모델링(머신러닝)의 종류 - ② 학습 목적에 따라

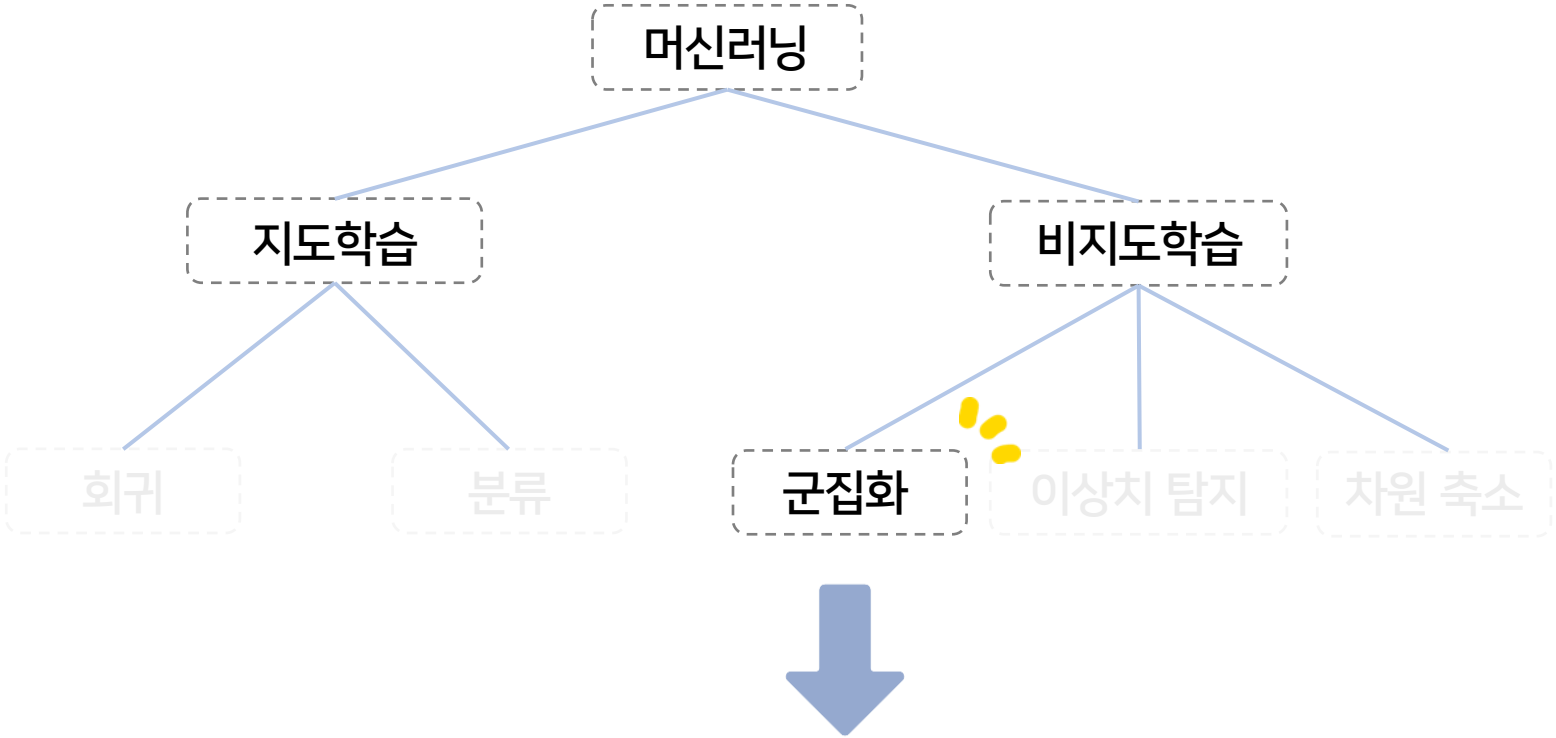


종속변수가 어떤 카테고리에 해당하는지 예측하는 것이 목적으로,

범주형 변수를 예측하는 **지도학습 방법론**

2 모델링

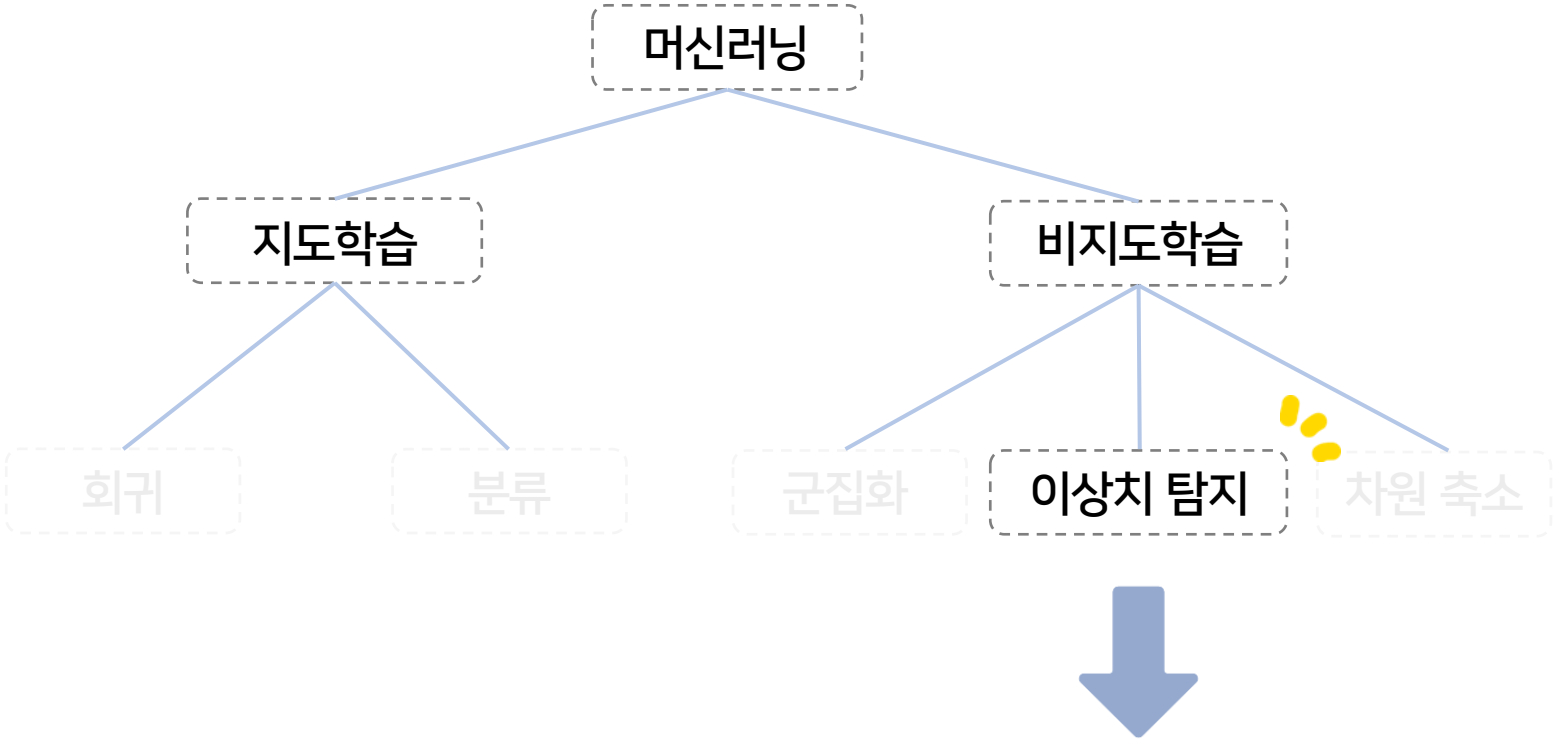
모델링(머신러닝)의 종류 - ② 학습 목적에 따라



유사한 개체들의 집단을 판별하는 것이 목적으로,

정답이 없는 **비지도학습 방법론**

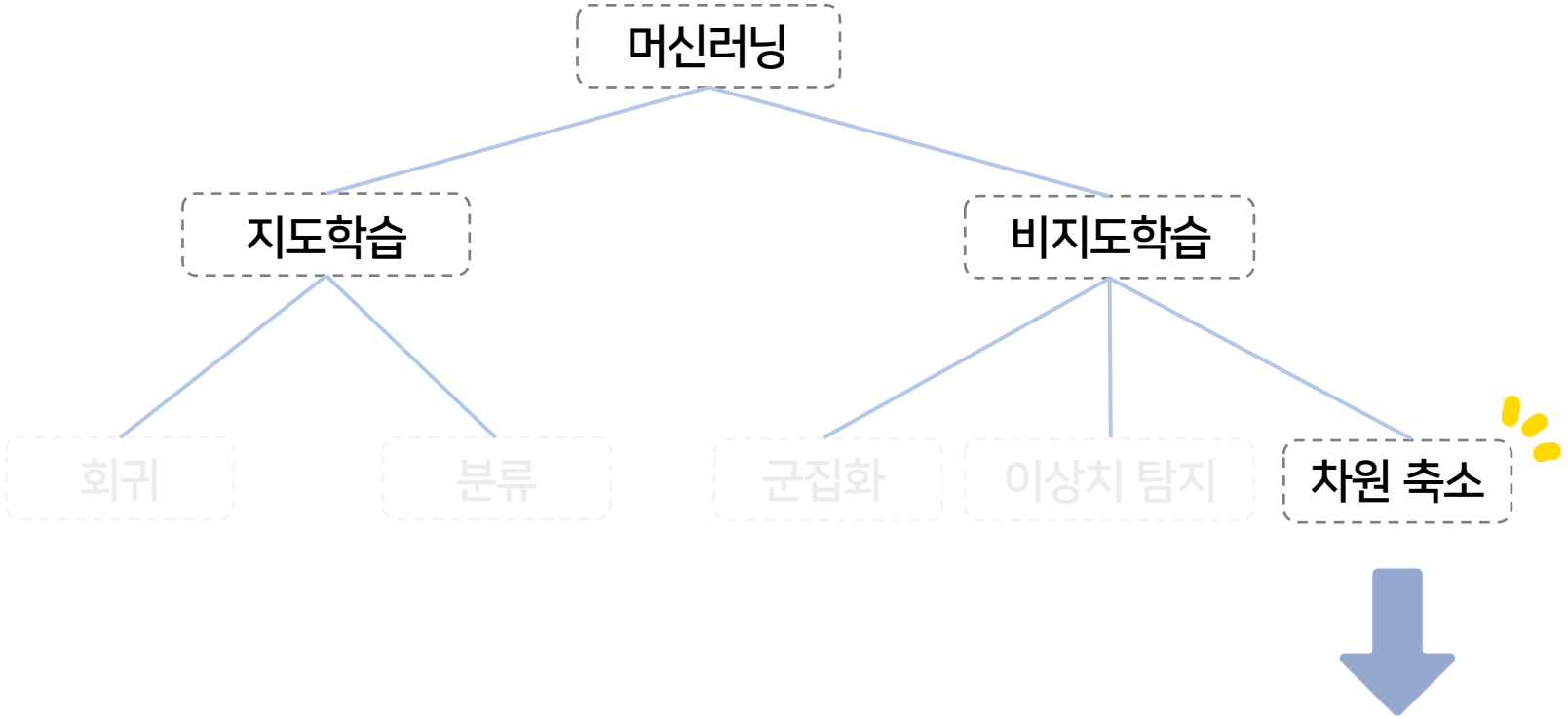
모델링(머신러닝)의 종류 - ② 학습 목적에 따라



대부분이 정상 데이터인 상황에서 매우 낮은 확률로 발생하는
이상 데이터를 찾아내는 것이 목적으로, **비지도학습 방법론**

2 모델링

모델링(머신러닝)의 종류 - ② 학습 목적에 따라



고차원 데이터를 저차원으로 간소화하는 것이 목적으로,

비지도학습 방법론

2 모델링

지도학습의 기본 원리

$$y = f(\mathrm{x}) + \varepsilon,\ \ \varepsilon \sim N(0, \sigma^2)$$

X: 독립변수

Y: 종속변수

f : X와 Y의 관계를 설명하는 함수

ε : 랜덤한 오차

지도학습의 기본 원리

$$y = f(\mathrm{x}) + \varepsilon,\ \ \varepsilon \sim N(0, \sigma^2)$$

지도학습은 실제 Y값에 근접한 추정치를 찾기 위해 함수 f를 추정해나가는 과정

1 모델링

지도학습의 기본 원리

f의 추정 목적

예측

X값을 통해 Y값을
예측하기 위해

< 목표 >

예측 오차 (Reducible Error)
최소화

추론

X와 Y의 관계를
파악하기 위해

< 목표 >

변수 선택 or 실용적인 해석

지도학습은 실제 Y값에 근접한 추정치를 찾기 위해 함수 f를 추정해나가는 과정

2 모델링

f를 추정하는 방법

$$f(X) = w_0 + w_1X_1 + w_2X_2 \qquad f(X) = \frac{1}{1 + e^{-(w_0 + w_1X_1 + w_2X_2)}}$$

$$f(X) = \sum_{m=1}^{n} k(m) I\{(x_1, x_2) \in R_m\}$$

$$f(X) = \frac{1}{1 + exp\left(-\left(w_0 + w_1\left(\frac{1}{1 + e^{-(w_{01} + w_{11}X_1 + w_{21}X_2)}}\right)\right) + w_2\left(\frac{1}{1 + e^{-(w_{02} + w_{12}X_1 + w_{22}X_2)}}\right)\right)}$$

모수적 방법

f 를 특정한 함수 또는
분포로 이루어졌다 가정,
f 의 파라미터들을
집중적으로 추정하는 방법

비모수적 방법

f 의 형태를 가정 X
데이터에만 의존하여 f를 추정,
ex) KNN 모델

2 모델링

f를 추정하는 방법

모수적 방법

f를 추정하는 문제가

parameter를 추정하는 문제로 단순화됨

모델링 이후에는 train data 불필요

ex) 회귀계수(β)를 학습하고 나면 새로운 데이터에 대해 예측 가능

다만, 가정한 형태가 실제 f와 맞지 않으면 성능이 좋지 않을 수 있음

이를 해결하기 위해 모델의 유연성을 증가시킬수록 **과적합** 위험 증가

2 모델링

f를 추정하는 방법

모수적 방법이 유리한 상황

1. 표본의 개수가 n ≥ 30으로 충분히 큰 경우 (중심 극한 정리)

2. 10 <= n <30이면서 정규성 검정에서 정규분포임이 확인되는 경우

3. 데이터에 대한 사전지식이 어느 정도 있는 경우

2 모델링

f를 추정하는 방법

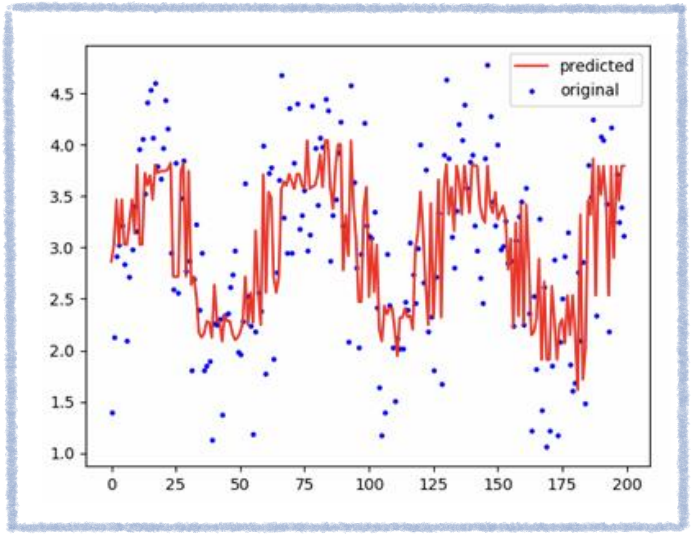
$$f(X) = w_0 + w_1X_1 + w_2X_2 \qquad f(X) = \frac{1}{1+e^{-(w_0+w_1X_1+w_2X_2)}}$$

$$f(X) = \sum_{m=1}^{n} k(m) I\{(x_1, x_2) \in R_m\}$$

$$f(X) = \frac{1}{1+exp\left(-\left(w_0 + w_1\left(\frac{1}{1+e^{-(w_{01}+w_{11}X_1+w_{21}X_2)}}\right) + w_2\left(\frac{1}{1+e^{-(w_{02}+w_{12}X_1+w_{22}X_2)}}\right)\right)\right)}$$

모수적 방법

f 를 특정한 함수 또는
분포로 이루어졌다 가정,
f 의 파라미터들을
집중적으로 추정하는 방법



비모수적 방법

f 의 형태를 가정 X

데이터에만 의존하여 f를 추정,

ex) KNN 모델

f를 추정하는 방법

비모수적 방법

함수에 대한 복잡한 가정이 없으니,

더 다양한 범위의 f 형태에 적합 가능

순위 척도 및 숫자로 이루어진 모든 경우에 대해 적용 가능

하지만, 데이터에 의존해서 f를 추정하기 때문에 적은 데이터의 경우 과적합 위험

f를 추정하는 방법

비모수적 방법이 유리한 상황

1. 모집단이 정규성을 띄지 않는다는 증명이 있거나 정규성을 가정하기 힘든 경우

2. 표본 집단의 크기가 작은 경우 (n < 10)

3. 데이터에 대한 사전지식이 없는 경우

모델 평가

어떤 모델이 **좋은 모델**인가?

이는 앞서 봤듯이 f **를 잘 추정하는 것**과 같음

f 를 추정하는 첫 번째 목적은 X를 통해 Y를 예측하는 것 (**예측 정확도**)

모델을 통해 예측한 **추정치(y)와 실제값($\hat{y}$)의 차이가 적을 수록** 좋은 예측이라고 판단 가능

모델 평가

$$MSE = E\left[(y - \hat{f}(x))^2\right]$$

단순히 $y - \hat{y}$ 를 성능으로 정의한다면,

음수 값의 존재로 정확한 성능 파악 불가

따라서, 오차의 제곱을 활용한 MSE(Mean Squared Error) 사용

MSE를 최소화하는 방향으로 모델링을 진행해야 함

MSE 분해를 통해 MSE를 간단화 해보자!

모델 평가

$$MSE = E\left[(y - \hat{f}(x))^2\right]$$

단순히 $y - \hat{y}$ 를 성능으로 정의한다면,

음수 값의 존재로 정확한 성능 파악 불가

따라서, 오차의 제곱을 활용한 MSE(Mean Squared Error) 사용

MSE를 최소화하는 방향으로 모델링을 진행해야 함

MSE 분해를 통해 MSE를 간단화 해보자!

모델 평가

MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

$$= E\left[\,(f(x)+\varepsilon-\hat{f}(x))^2\right]$$

$$= E\left[(f(x)-\hat{f}(x))^2+\varepsilon^2+2\left(f(x)-\hat{f}(x)\right)\varepsilon\right]$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + E[\varepsilon^2] + 2 * E\left[f(x)-\hat{f}(x)\right] * E[\varepsilon]$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + E[\varepsilon^2]$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + \sigma^2$$

모델 평가

MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

Reducible Error(축소 가능 오차)

$\hat{f}$ 는 우리가 컨트롤할 수 있는 오류

예측 수행 시 더 좋은 모델을 찾음으로써 감소

→ 감소시킬 수 있는 Error Term

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + \sigma^2$$

모델 평가

MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

$$= E\left[(f(x)+\varepsilon-\hat{f}(x))^2\right]$$

$$= E\left[(f(x)-\hat{f}(x))^2 + \right.$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + E[\varepsilon^2] + 2\times E[f(x)-\hat{f}(x)]\times E[\varepsilon]$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + E[\varepsilon^2]$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + \sigma^2$$

Irreducible Error(축소 불가능 오차)

X들로는 Y를 완전히 결정할 수 없음

생각 못한 변수, 관측이 불가능한 변수 등

→ 우리가 줄일 수 없는 Error Term

모델 평가

MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

우리는 결국 **Reducible Error를 줄여야 함**

→ Reducible Error을 더 분해해보자!

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] + \sigma^2$$

모델 평가

MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] = E\left[\left(f(x)-\bar{f}(x)+\bar{f}(x)-\hat{f}(x)\right)^2\right]$$

$$= E\left[\left(f(x)-\bar{f}(x)\right)^2 + \left(\hat{f}(x)-\bar{f}(x)\right)^2 + 2*\left(f(x)-\bar{f}(x)\right)*\left(\hat{f}(x)-\bar{f}(x)\right)\right]$$

$$= E\left[\left(f(x)-\bar{f}(x)\right)^2\right] + E\left[\left(\hat{f}(x)-\bar{f}(x)\right)^2\right] + 2*E\left[f(x)-\bar{f}(x)\right]*E\left[\left(\hat{f}(x)-\bar{f}(x)\right)\right]$$

$$= E\left[\left(f(x)-\bar{f}(x)\right)^2\right] + E\left[\left(\hat{f}(x)-\bar{f}(x)\right)^2\right]$$

$Bias\left(\hat{f}(x)\right)^2$ $\quad$ $Var\left(\hat{f}(x)\right)$

모델 평가

MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

$$= E\left[\left(f(x)-\hat{f}(x)\right)^2\right] = E\left[\left(f(x)-\bar{f}(x)+\bar{f}(x)-\hat{f}(x)\right)^2\right]$$

Bias(편향)

= 추정치의 기댓값 $\bar{f}(x)$ 과 실제값 $f(x)$ 의 차이

추정한 모델이 실제 모델을 얼마나 잘 설명하는지 의미

$$= E\left[\left(f(x)-\bar{f}(x)\right)^2\right] - E\left[\left(\hat{f}(x)-\bar{f}(x)\right)^2\right] + 2 * E\left[f(x)-\bar{f}(x)\right] * E\left[\left(\hat{f}(x)-\bar{f}(x)\right)\right]$$

$$= E\left[\left(f(x)-\bar{f}(x)\right)^2\right] + E\left[\left(\hat{f}(x)-\bar{f}(x)\right)^2\right]$$

$$Bias\left(\hat{f}(x)\right)^2 \qquad Var\left(\hat{f}(x)\right)$$

모델 평가

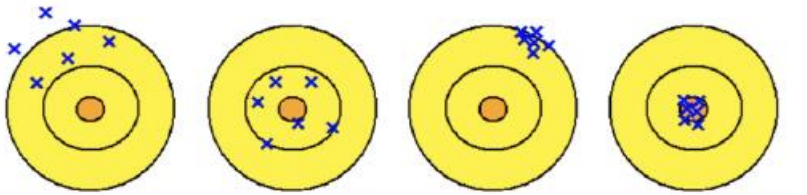
MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

Bias(편향)

편향 大 : 모델링 반복 수행 해도 정답 가능성 낮음

편향 小 : 모델링 반복 수행시 평균적으로 잘 맞춰냄



Bias	High	Low	High	Low
Variance	High	High	Low	Low

$$Bias\left(\hat{f}(x)\right)^2$$

$$Var\left(\hat{f}(x)\right)$$

모델 평가

MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

Variance(분산)

= 추정치의 개별값 $\hat{f}(x)$ 와 추정치의 기댓값 $\bar{f}(x)$ 의 차이를 제곱합

추정한 모델에 다른 데이터셋을 적합했을 때, **추정치가 얼마나 달라지는지 의미**

$$= E\left[\left(f(x)-\bar{f}(x)\right)^2\right] + E\left[\left(\hat{f}(x)-\bar{f}(x)\right)^2\right]$$

$$Bias\left(\hat{f}(x)\right)^2 \qquad Var\left(\hat{f}(x)\right)$$

모델 평가

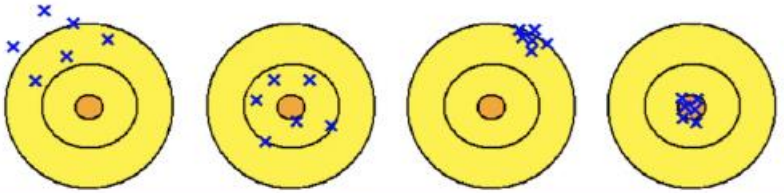
MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

Variance(분산)

분산 大 : 노이즈가 바뀜에 따라 개별 추정값들이 많이 바뀜

분산 小 : 노이즈가 바뀌어도 개별 추정값들이 잘 바뀌지 않음



Bias	High	Low	High	Low
Variance	High	High	Low	Low

$Bias\left(\hat{f}(x)\right)^2$

$Var\left(\hat{f}(x)\right)$

모델 평가

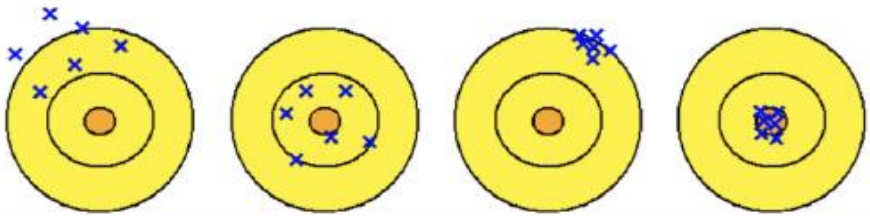
MSE Decomposition

$$E\left[(y-\hat{f}(x))^2\right] = Bias(f(x))^2 + Var\left(f(x)\right) + \sigma^2$$

정리하자면!

Bias : 모델의 **평균 정확도가 얼마나 많이 변하는지**를 보여줌

Variance : 특정 입력 데이터에 대해 **알고리즘이 얼마나 민감한지**를 보여줌



Bias	High	Low	High	Low
Variance	High	High	Low	Low

모델 평가

MSE Decomposition

$$E[(y-\hat{f}(x))^2] = Bias(f(x))^2 + Var(f(x)) + \sigma^2$$

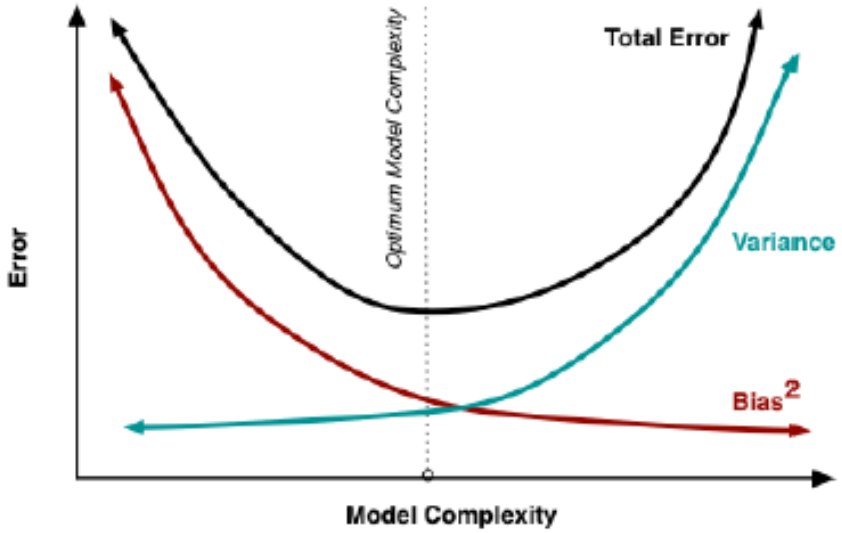
MSE 분해 결과,

모델의 Bias와 Variance를 줄이는 방향으로

학습을 진행해야 한다는 걸 알게 됨

그러나 **2가지를 동시에 줄이는 것이** 과연 가능할까?

Bias-Variance Trade off

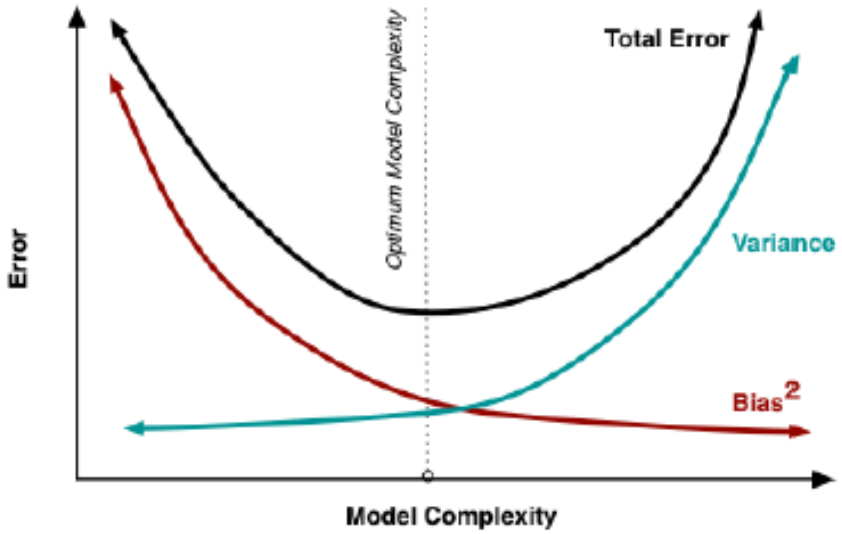


Bias-Variance Trade off

위 그림처럼, 현실적으로 2가지를 동시에 줄이기는 어려움

→ Bias와 Variance는 서로 **반대 방향**으로 움직임

Bias-Variance Trade off



Bias-Variance Trade off

즉, **모델이 복잡할수록** 예측값이 불안정해지고 Variance 증가

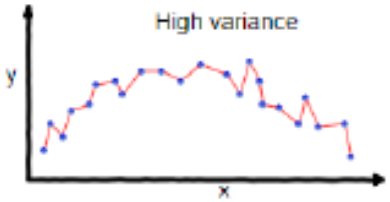
반대로 **모델이 단순할수록** 예측값과 실제값 사이의 차이가 커지고 Bias 증가

Bias-Variance Trade off

요약하자면,

Bias와 Variance는 **모델 복잡도의 영향**을 많이 받음!

Bias-Variance Trade off



overfitting

underfitting

Good balance

복잡한 모델

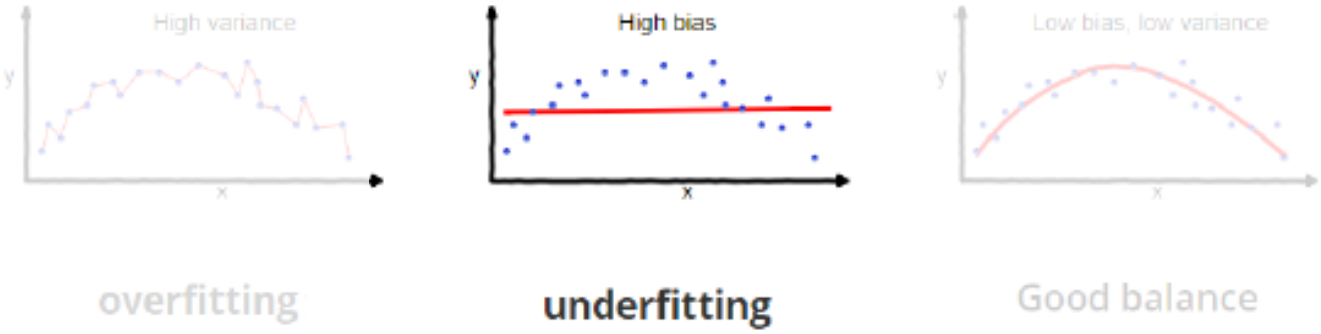
예측값과 실제값의 차이가 작음 = Bias ↓

새로운 데이터가 하나만 추가되어도 매우 달라짐 = Variance ↑

지나치게 복잡하면 좋은 모델이 될 수 없음 = Overfitting

2 모델링

Bias-Variance Trade off



단순한 모델

예측값과 실제값의 차이가 큼 = Bias ↑

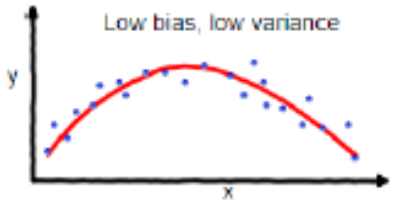
데이터 포인트가 하나 더 생겨도 약간의 기울기 변화만 생김 = Variance ↓ (=Robust)

지나치게 단순하면 좋은 모델이 될 수 없음 = Underfitting

Bias-Variance Trade off

overfitting

underfitting



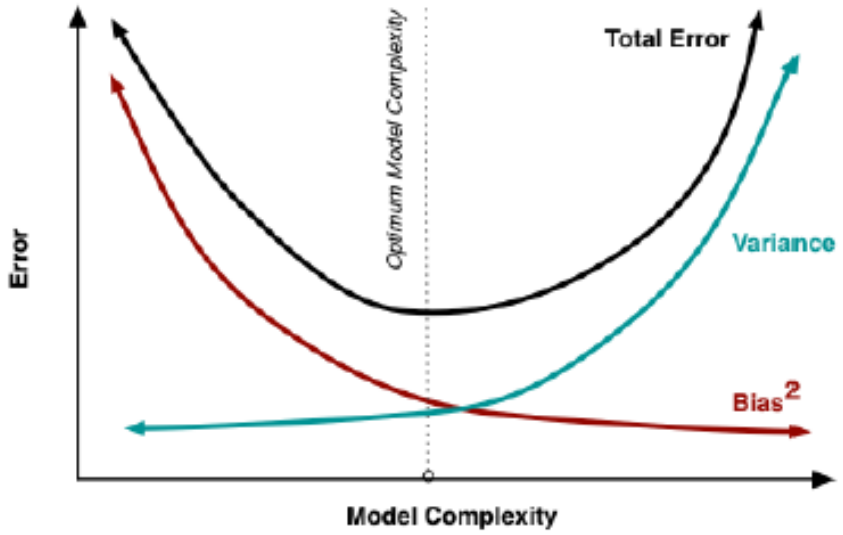
Good balance

적절한 모델

너무 복잡하지도, 너무 단순하지도 않음

Bias와 Variance가 모두 적절히 작아 그 합이 최소

Bias-Variance Trade off



Bias-Variance Trade off

Bias – Variance Trade off 그래프의 점선이 optimal point

= MSE가 최소가 되는 Point

2 모델링

KNN (K-Nearest Neighbor)

KNN-Classifier

비모수적인 모델로 K개의 가까운 이웃데이터들 중 **다수결로 클래스를 예측**

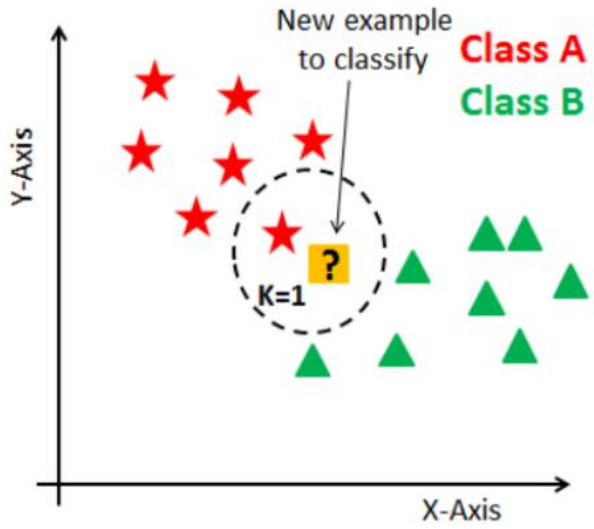
KNN-Regressor

비모수적인 모델로 K개 이웃들의 **Y값 평균을 활용하여 종속변수 값을 예측**

KNN (K-Nearest Neighbor)

KNN-Classifier

비모수적인 모델로 K개의 가까운 이웃데이터들 중 **다수결로 클래스를 예측**



유클리드 거리를 측정하여
가까운 정도를 판단

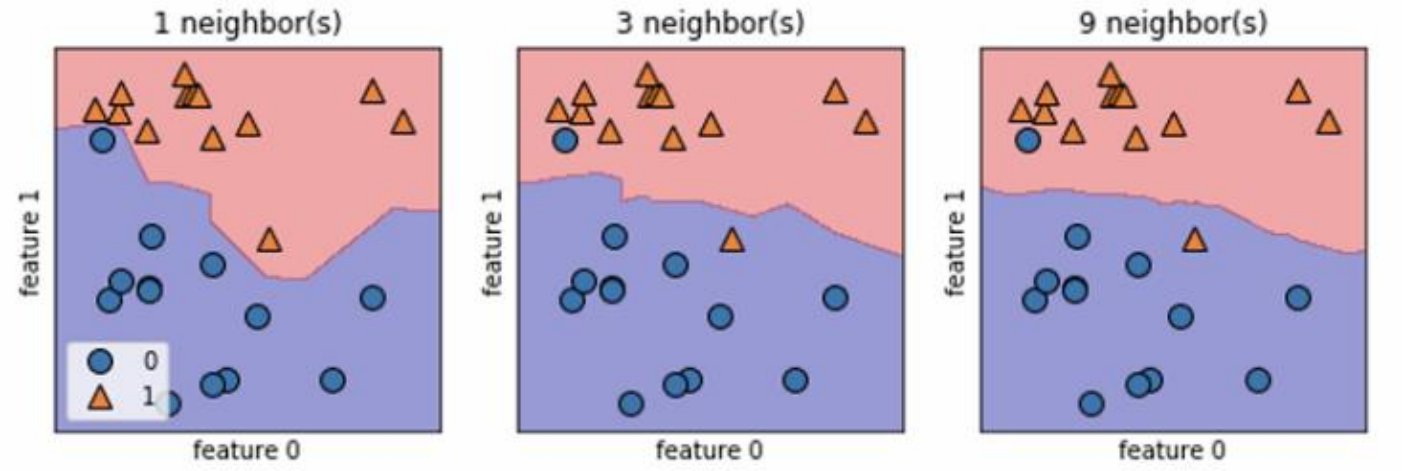
KNN (K-Nearest Neighbor)

KNN-Classifier

비모수적인 모델로 **K개**의 가까운 이웃데이터들 중 **다수결로 클래스를 예측**

Hyperparameter로서

Decision Boundary 의 복잡함 정도를 결정



KNN (K-Nearest Neighbor)

K ↓

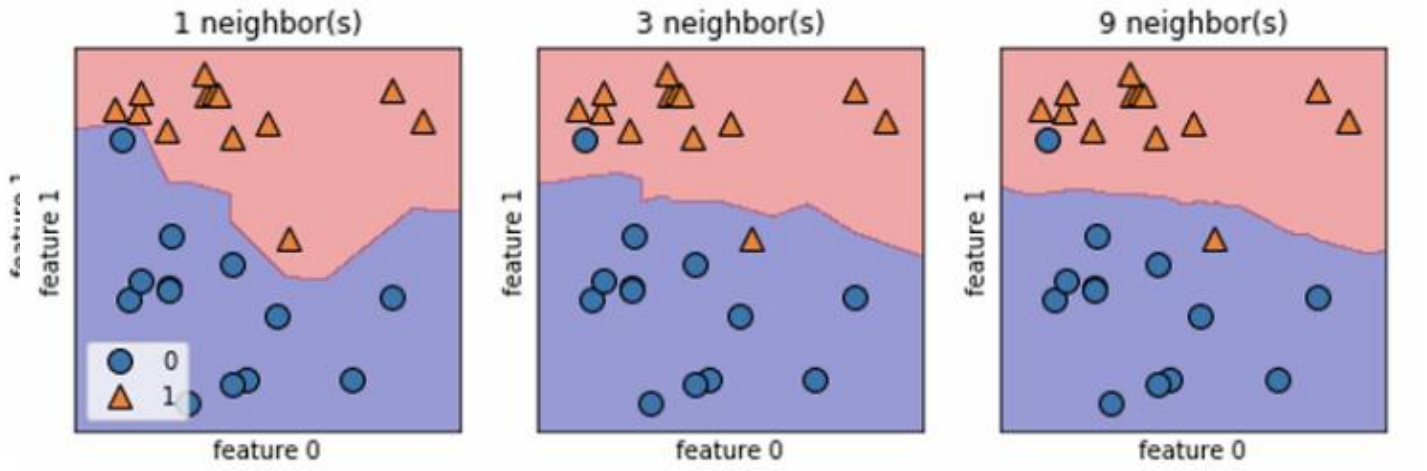
Decision Boundary 가

복잡해짐

K ↑

Decision Boundary 가

단순해짐



2 모델링

KNN (K-Nearest Neighbor)

KNN-Regressor

비모수적인 모델로 K개 이웃들의 **Y값 평균을 활용하여 종속변수 값을 예측**

$$\widehat{Y_0} = \frac{1}{K} \sum_{x_i \epsilon N_K(x_0)} Y_i$$

(x_i, y_i) : *training data*

$N_k(x_o)$: *neighborhood of* x_0

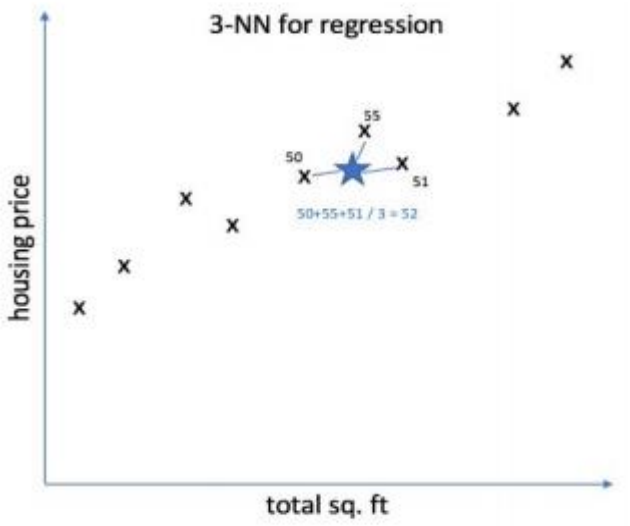
비슷한 X값을 가진 데이터끼리는 **비슷한 Y**값을 가질 것이라는 Idea

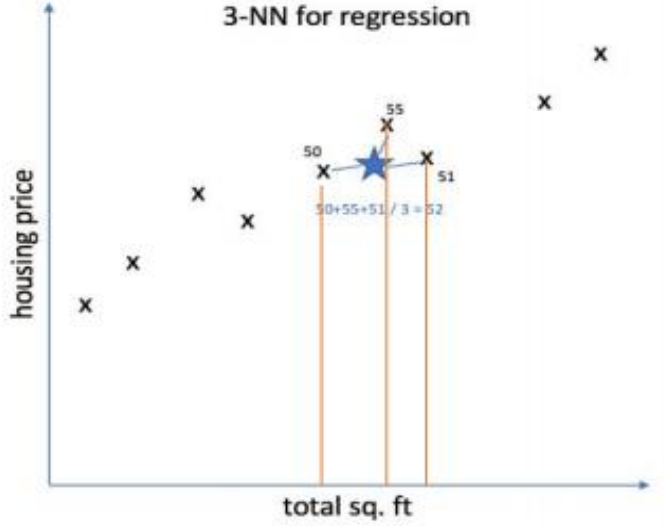
KNN (K-Nearest Neighbor)

KNN-Regressor

비모수적인 모델로 K개 이웃들의 **Y값 평균을 활용하여 종속변수 값을 예측**

X값을 기준으로 가까운 점을 찾아서 Y값 평균을 계산





KNN (K-Nearest Neighbor)

K ↓

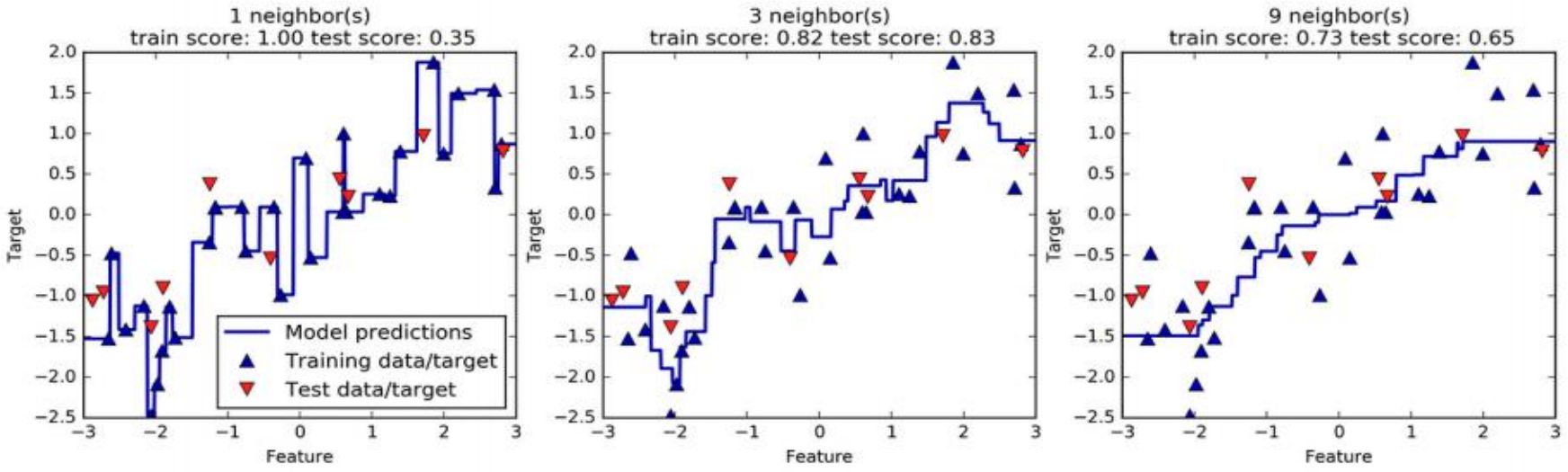
Decision Boundary 가

복잡해짐

K ↑

Decision Boundary 가

단순해짐



KNN (K-Nearest Neighbor)

K ↓ K 증감에 따른 Bias & Variance 변화 K ↑

Decision Boundary 가 복잡해짐 Decision Boundary 가 단순해짐

K 증가 -> Model Complexity 감소 -> High Bias & Low Variance

K 감소 -> Model Complexity 증가 -> Low Bias & High Variance

3

모델링 전략

Hyperparameter vs Parameter

Hyperparameter

모델 학습 시 **사용자가 직접 지정**해야 하는 변수

Parameter

데이터로부터 학습되어 결정됨 모델 내부에서 결정되는 변수

Hyperparameter vs Parameter

라면을 끓이는 데 고려해야 할 요인

(물 온도, 물의 양 ...)

Hyperparameter

연구자들이 개발한

라면 제품의 재료성분

Parameter

3 모델링 전략

Hyperparameter vs Parameter

모델의 성능을 높이기 위해서는

Hyperparameter Optimization을 통해

적절한 Hyperparameter를 찾아서

MSE의 Reducible error부분을 줄여야 함

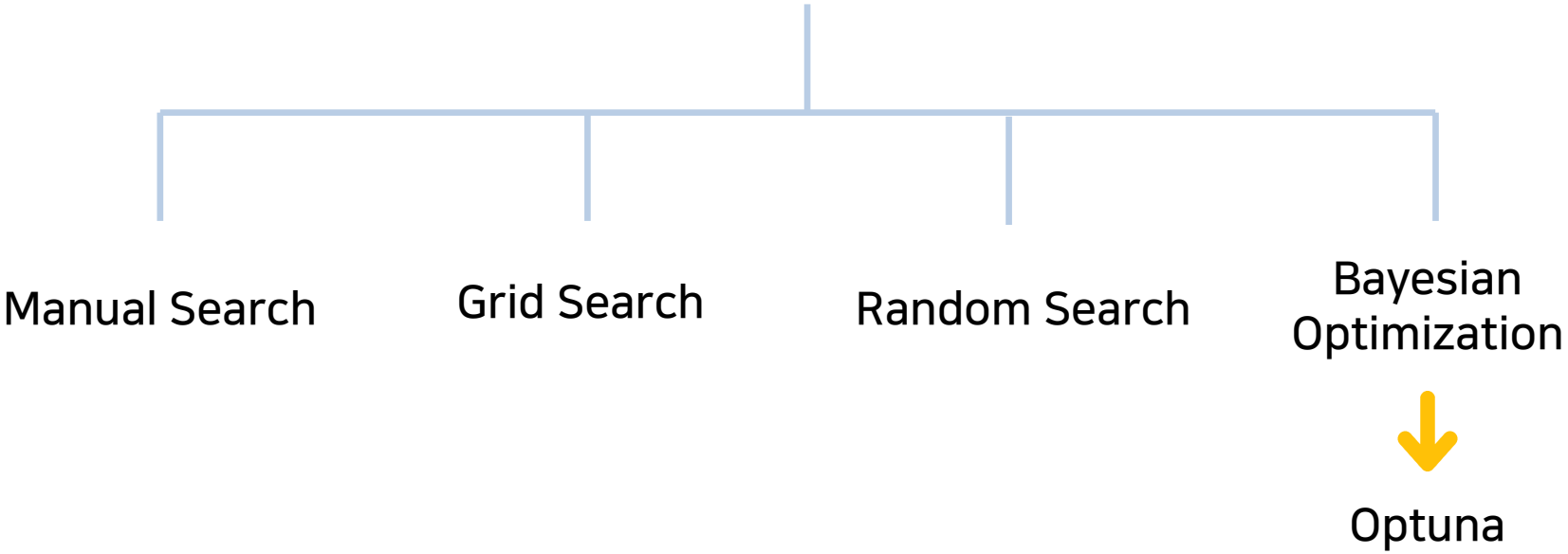
Parameter

Hyperparameter

Hyperparameter Optimization

Hyperparameter Optimization

Hyperparameter 조합 중 **성능을 가장 좋게 만드는** 조합을 찾아가는 과정



Manual Search

Manual Search

사용자가 **수동으로 성능을 비교**해가며 Hyperparameter를 튜닝하는 방법

사용자의 **직관 및 경험에 의존** 할 수밖에 없음

Manual Search

Manual Search

사용자가 **수동으로 성능을 비교**해가며 Hyperparameter를 튜닝하는 방법

시간이 오래 걸리며 **효율성**을 보장하기 어렵고,

고차원의 Hyperparameter를 탐색할 때

직관만을 이용하기에는 한계가 존재

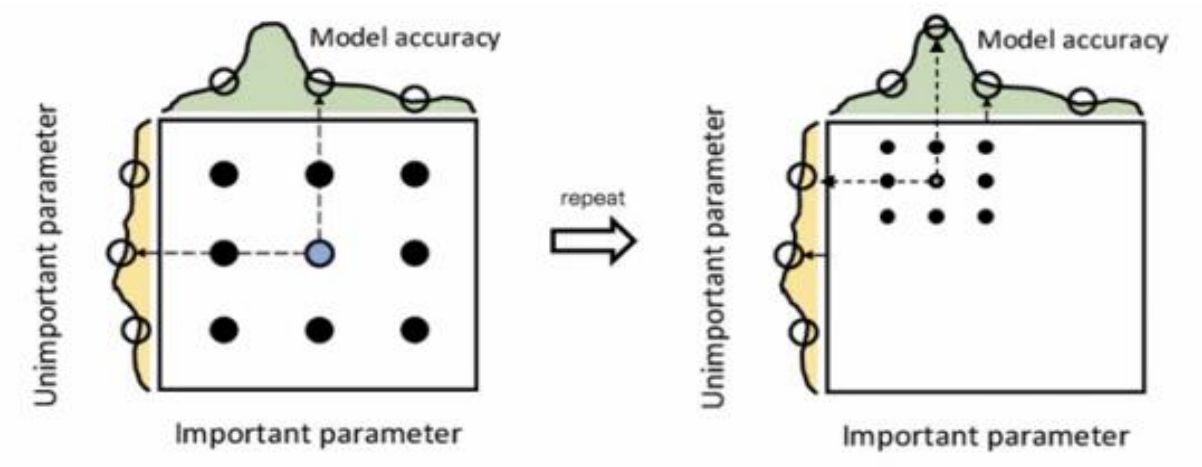
Grid Search

Grid Search

탐색 범위 내의 모든 Hyperparameter 조합들을

격자 방식으로 비교하여 탐색하는 방법

각 Trial을 **병렬화 하기 용이**하고 구현이 쉬움



Grid Search

Grid Search

탐색 범위 내의 모든 Hyperparameter 조합들을
격자 방식으로 비교하여 탐색하는 방법

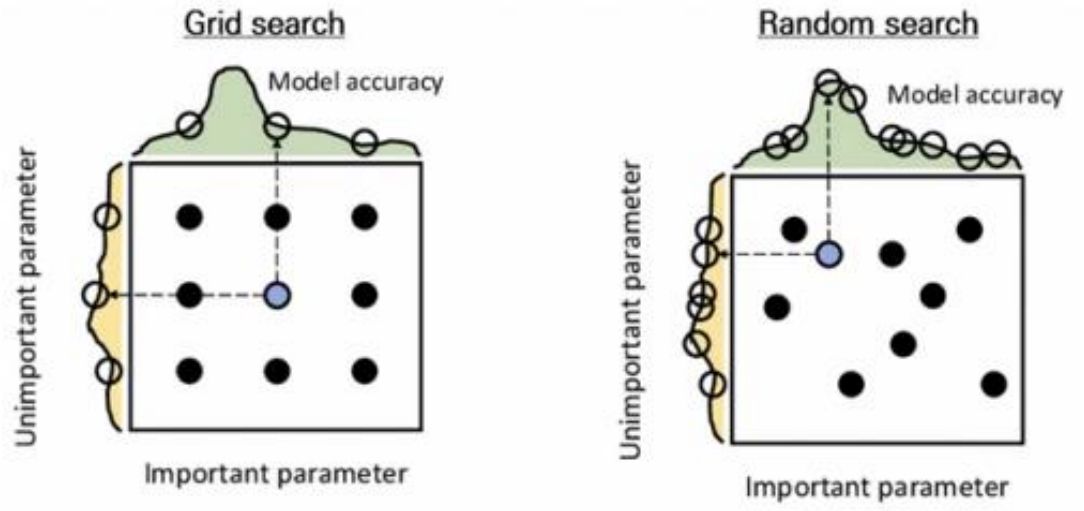
모든 조합을 탐색해야 하므로
고차원의 Hyperparameter에는 사용이 어렵고
탐색 범위 역시 **수동**으로 좁혀나가야 함

Random Search

Random Search

구간 내 랜덤 샘플링을 통해 후보 Hyperparameter 값들을 선택하는 방법

범위 내의 임의의 값으로 조합을 시도해볼 수 있음



Random Search

정해진 간격 사이에서 **확률적 탐색**이 가능해

예상치 못한 Hyperparameter 조합 시도 가능

Grid Search 에 비해 **시간이 적게** 걸리며 **높은 성능**을 보임

3 모델링 전략

Random Search

Grid Search 및 Random Search 의 한계

성능이 좋지 않은 조합들도 그냥 탐색해 **계산 자원을 낭비**

각각의 Trial이 독립적이어서 **사전 지식을 전혀 반영하지 못함**

Grid Search 에 비해 **시간이 적게** 걸리며 **높은 성능**을 보임

이러한 한계점들을 해결해줄 수 있는

Bayesian Optimization 활용 필요

3 모델링 전략

Bayesian Optimization

Bayesian Optimization

베이지안 확률에 기반해 **목적함수 $f(x)$ 를 최대로 하는 해**를 찾는 방법

사전 정보를 이용하여 사후 모델을 개선시켜 나감

사전 정보를 얻으려면?

Surrogate Model

⊕

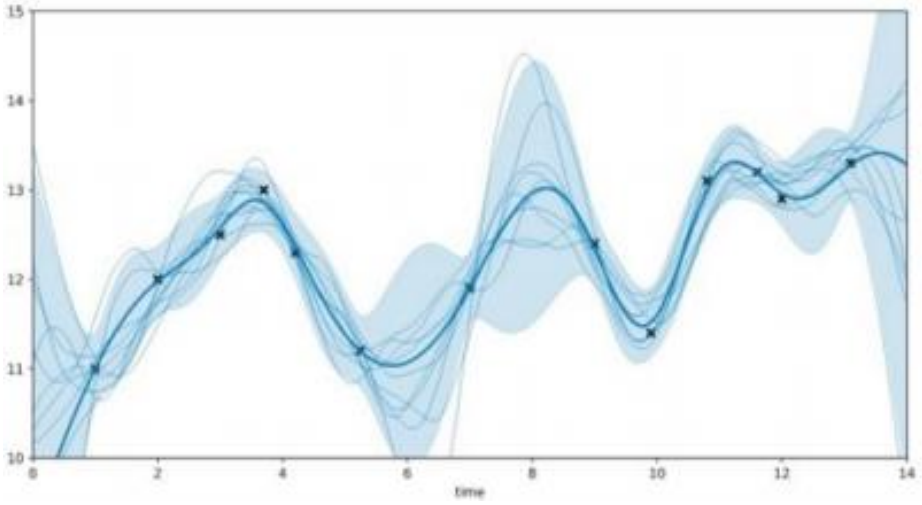
얻은 사전 정보를
다음 탐색에 이용하려면?

Acquisition Function

Bayesian Optimization

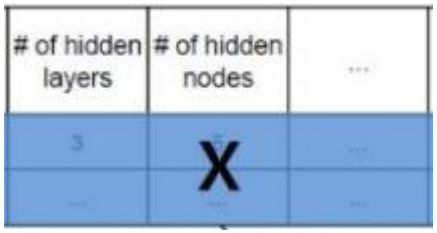
Surrogate Model

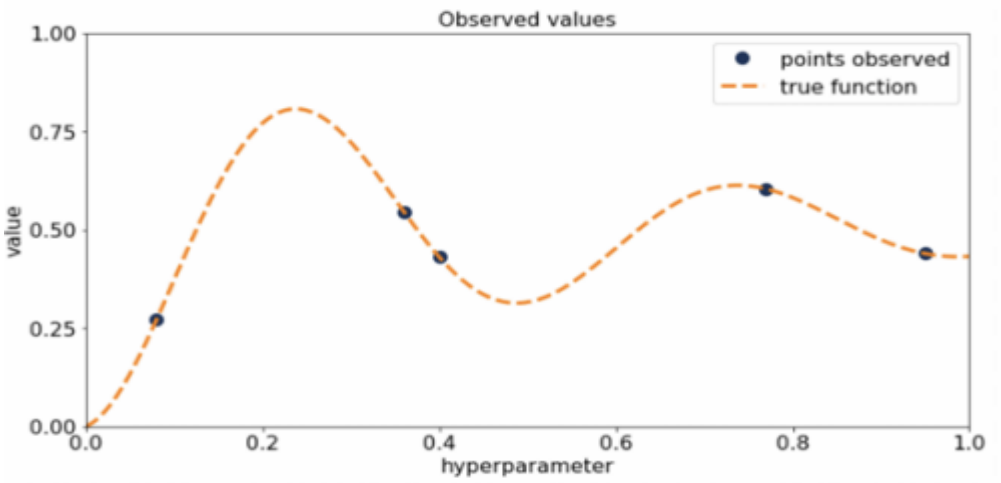
미지의 목적 함수의 형태에 대해 **확률적인 추정**을 수행하는 모델



Bayesian Optimization

Surrogate Model 작동 과정



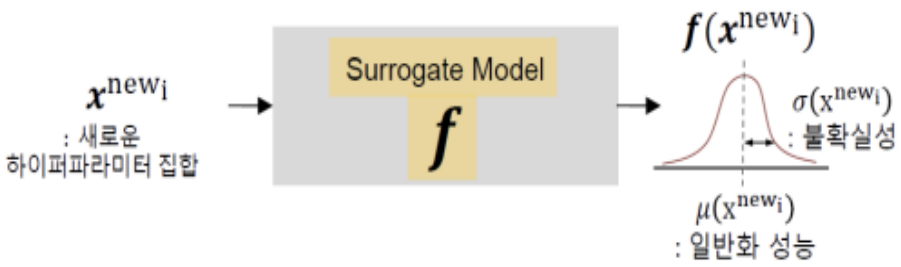


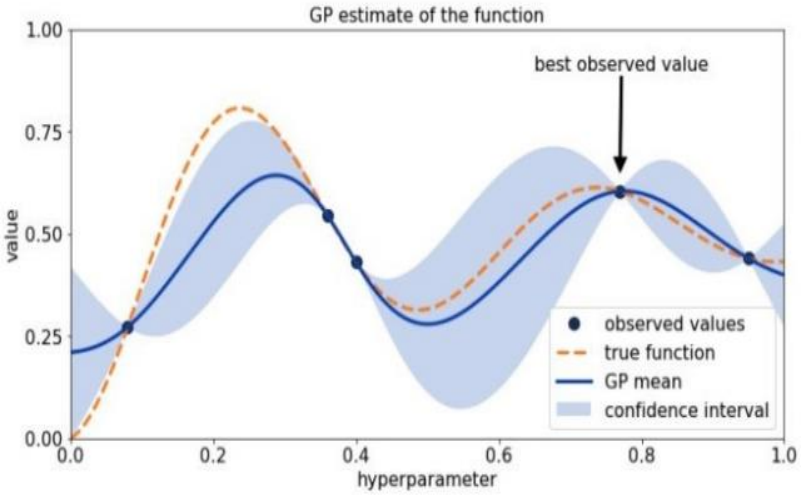
① Hyperparameter들을 랜덤 샘플링 하여 **성능 결과를 관측**

주황색 점선 = 우리가 찾아야 하는 목적함수

Bayesian Optimization

Surrogate Model 작동 과정





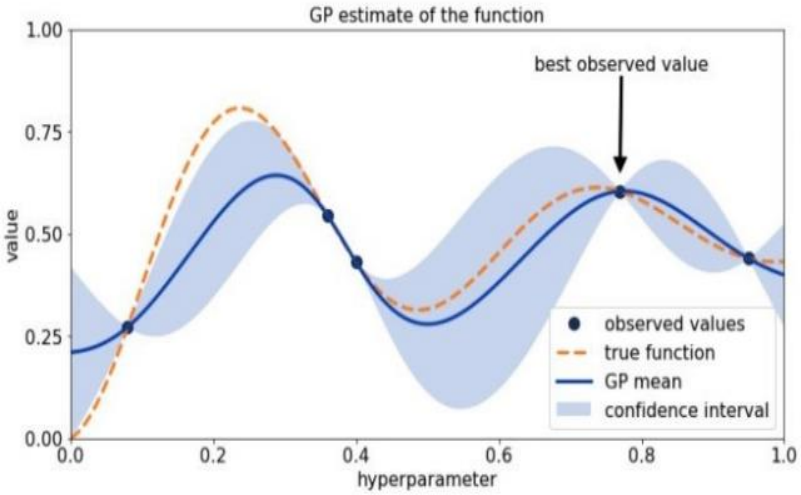
② 관측된 결과를 기반으로 Surrogate Model이 목적함수 f를 추정

Bayesian Optimization

Surrogate Model 작동 과정

파란색 영역은

신뢰 구간으로, 불확실성을 의미



목적함수를 추정하는 데 있어

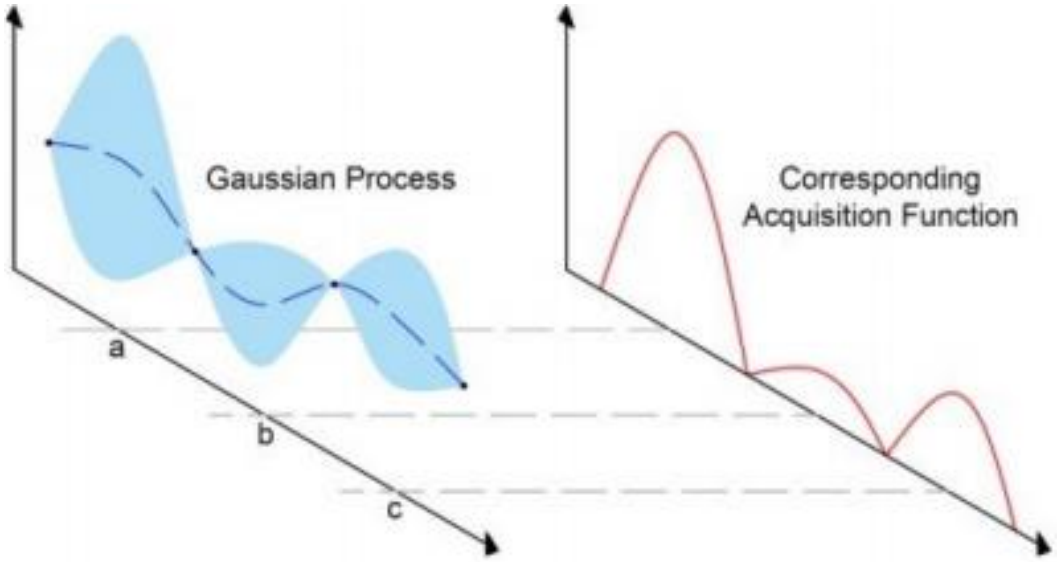
일반적으로 Gaussian Process (GP)를 사용

Bayesian Optimization

Acquisition Function

목적 함수에 대한 현재까지의 확률적 추정 결과를 바탕으로

다음 탐색 후보를 추천하는 함수



3 모델링 전략

Bayesian Optimization

Acquisition Function

목적 함수에 대한 현재까지의 확률적 추정 결과를 바탕으로

다음 탐색 후보를 추천하는 함수

Acquisition Function이 **탐색 후보를 추천해주는 전략**에는

아래와 같이 2가지가 존재

착취 (Exploitation)

최적 값일 가능성이 높은 값을 추천

탐색 (Exploration)

탐색이 불확실한 값을 추천

3 모델링 전략

Bayesian Optimization

Exploitation – Exploration은 서로 Trade-off 관계

따라서 성공적인 최적 입력 값 탐색을 위해서는

Exploitation – Exploration 간의 상대적 강도를

적절히 조절하는 게 중요

Exploitation

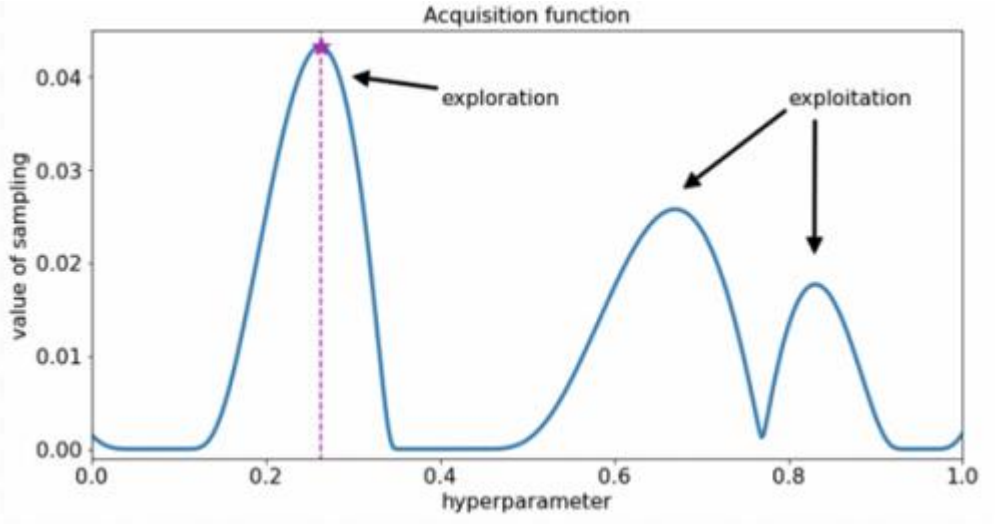
최적 값일 가능성이 높은 값을 추천

Exploration

탐색이 불확실한 값을 추천

Bayesian Optimization

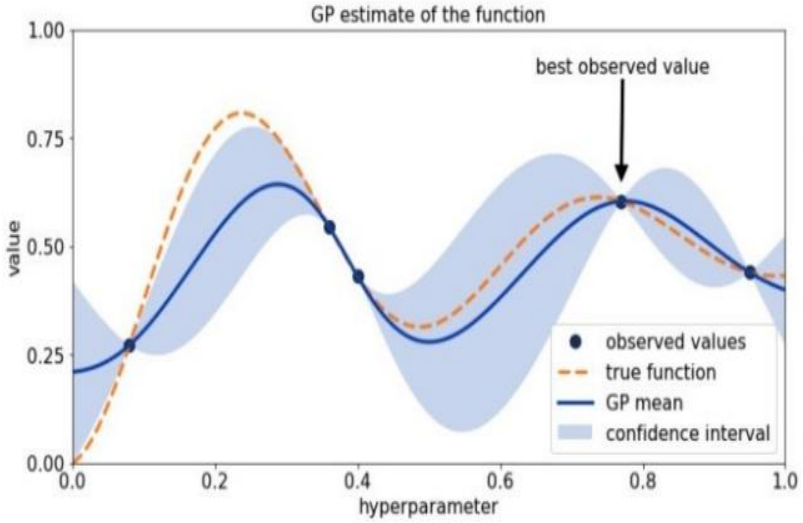
Acquisition Function 작동 과정

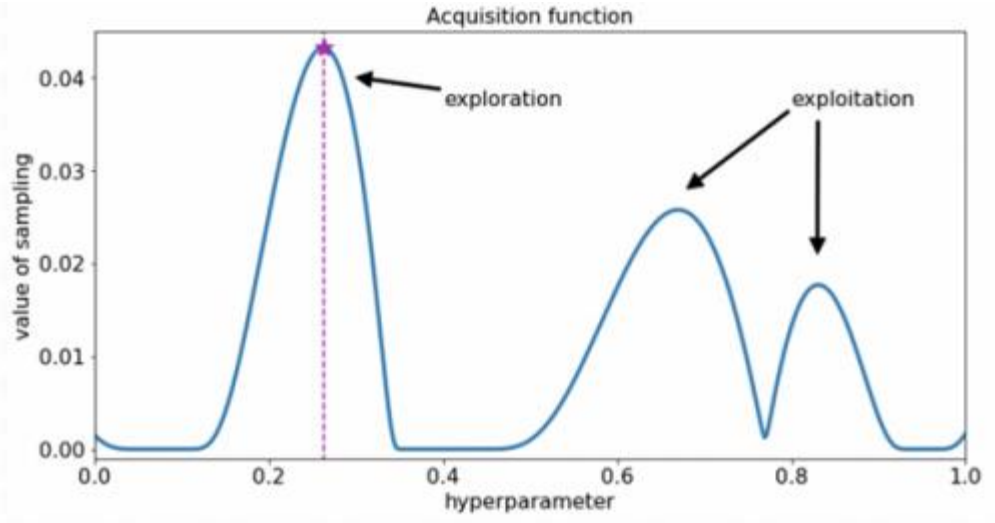


① 추정된 Surrogate Model 기반으로 Acquisition Function 계산

Bayesian Optimization

Acquisition Function 작동 과정



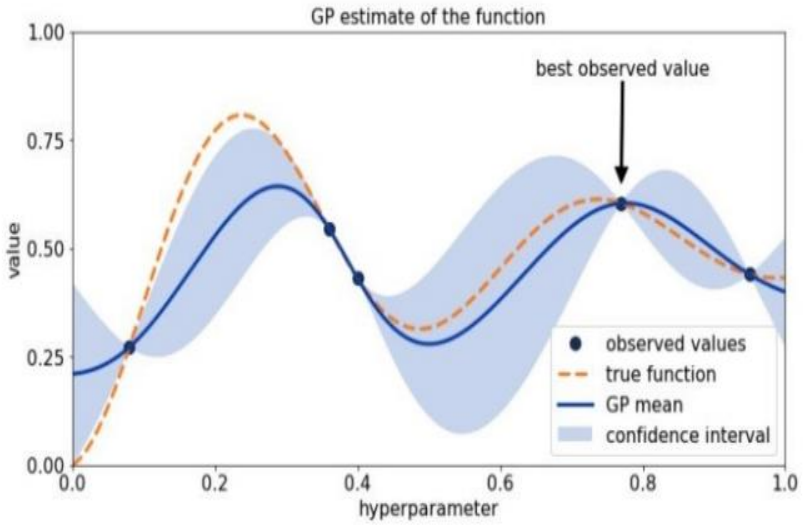


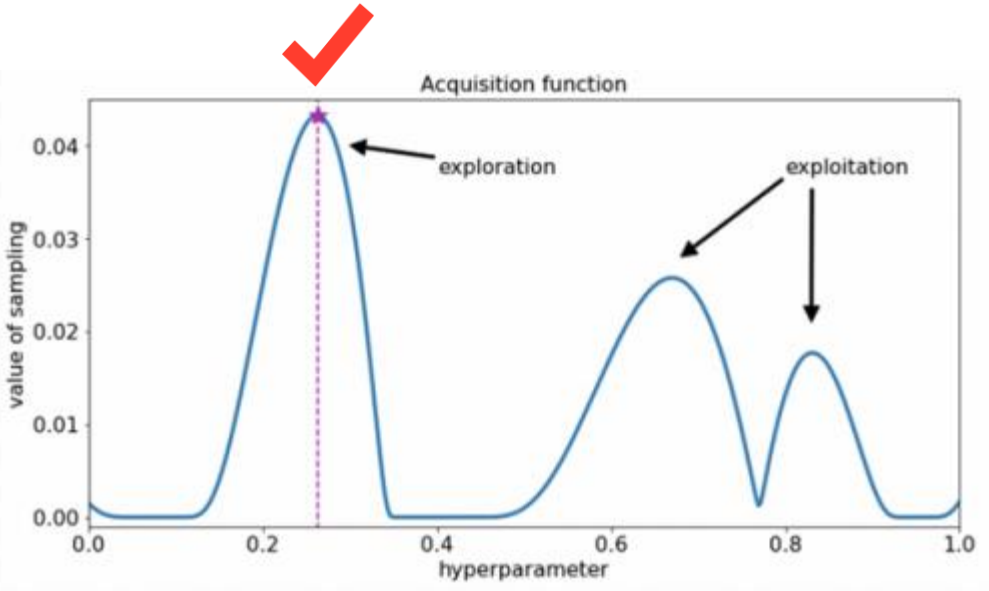
② Acquisition Function에 의해 Hyperparameter가 추천됨

3 모델링 전략

Bayesian Optimization

Acquisition Function 작동 과정





Acquisition 함숫값이 큰 지점이 다음 입력 값으로 추천됨

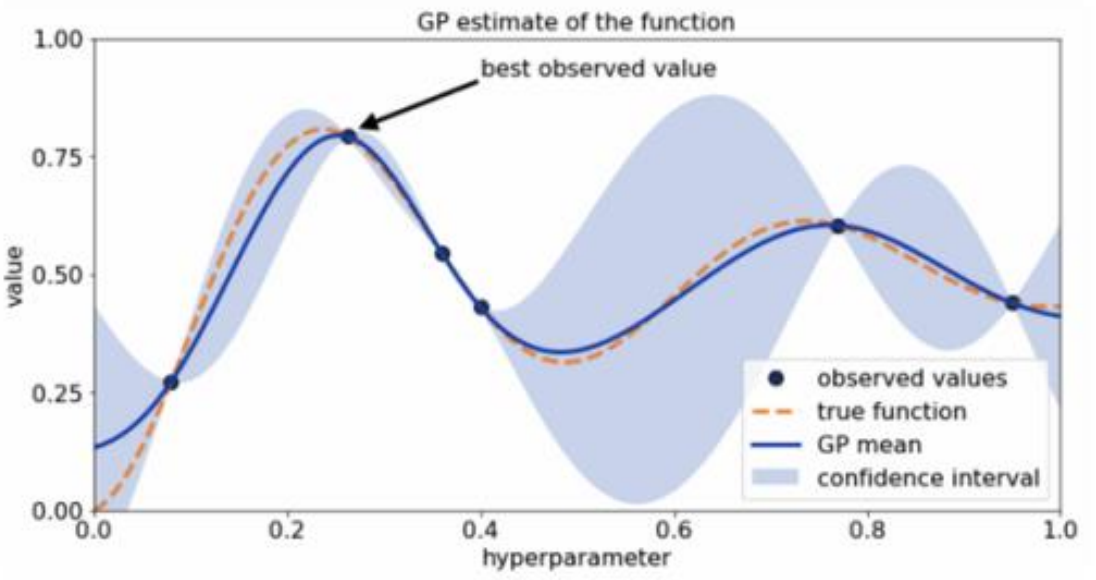
② Acquisition Function에 의해 Hyperparameter가 추천됨

(Exploitation < Exploration)

(자세한 내용은 24-1 데마팀 클린업 1주차 교안 참고)

Bayesian Optimization

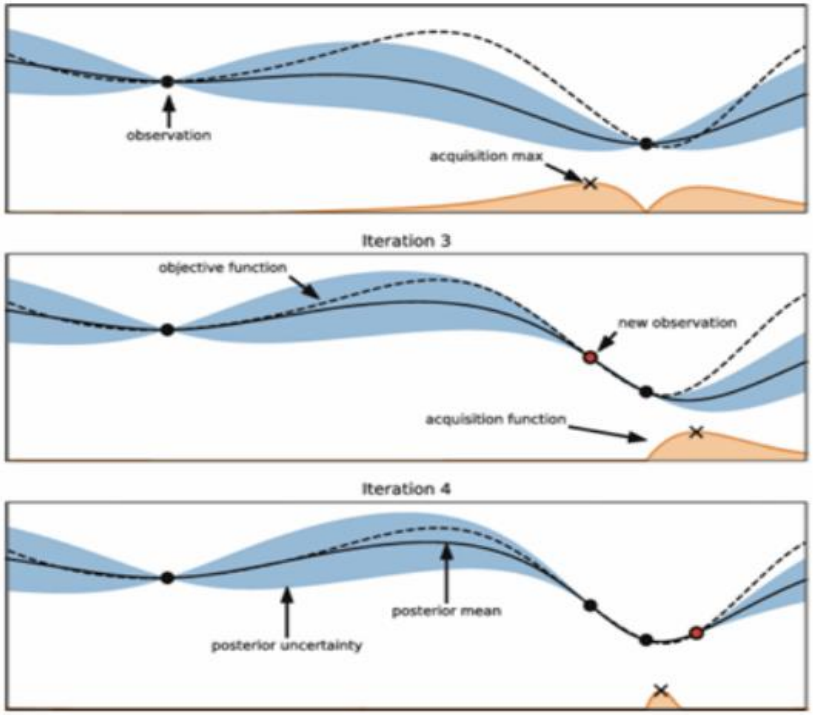
Acquisition Function 작동 과정



③ 추천 받은 Hyperparameter로 성능 측정 후 Surrogate Model 갱신

3 모델링 전략

Bayesian Optimization



Surrogate Model 작동 과정 ②

~

Acquisition Function 작동 과정 ④

반복을 통해 실제 목적함수에 근사 가능

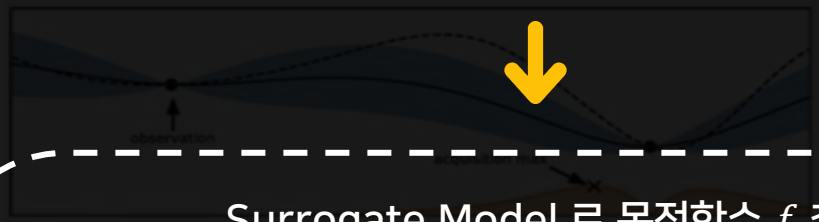
↓

최적의 Hyperparameter Find

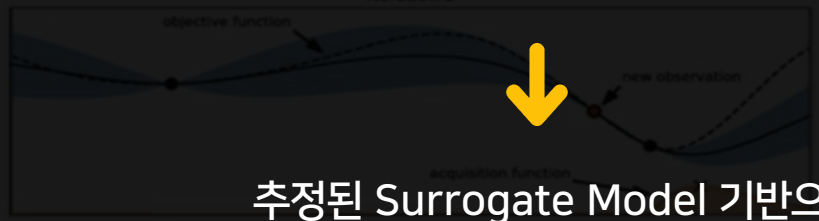
Bayesian Optimization 작동과정 정리

Bayesian Optimization

Hyperparameter 랜덤 샘플링 후 성능 결과 관측

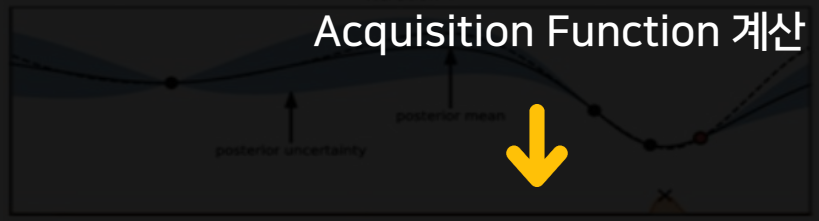


Surrogate Model 로 목적함수 f 추정



추정된 Surrogate Model 기반으로

Acquisition Function 계산



추천 받은 Hyperparameter 로 성능 측정 후

Surrogate Model 갱신

Surrogate Model 작동 과정 ②

Acquisition Function 작동 과정 ④

반복을 통해 실제 목적함수에 근사 가능

$\tilde{f}$ 목적함수 근사

최적의 Hyperparameter Find

최적의

Hyperparameter

Find

Bayesian Optimization

Gaussian Process(GP) 를 Surrogate Model 로 쓸 경우,
시간복잡도가 크고 연속형 변수에만 사용 가능하다는 단점이 존재

이를 위한 해결방안으로 Tree Parzen Estimator(TPE) 등
다른 Surrogate Model 들이 존재

(자세한 내용은 23-1 알파팀 자료 참고)

3 모델링 전략

Optuna

Optuna

베이지안 최적화에 기반해 확률모델을 구축하고
이를 토대로 최적의 Hyperparameter 값을 추천해주는,
파이썬 기반의 **오픈소스 라이브러리**

Optuna 의 장점

- 비교적 사용이 간단하고 속도가 빠름
- 최신 동향의 다양한 최적화 알고리즘이 구축됨
- 병렬 처리 가능
- 간단하고 직관적인 API 제공

3 모델링 전략

Optuna

Optuna

베이지안 최적화에 기반해 확률모델을 구축하고
이를 토대로 최적의 Hyperparameter 값을 추천해주는,
파이썬 기반의 **오픈소스 라이브러리**

Optuna 의 장점

- 비교적 사용이 간단하고 속도가 빠름
- 최신 동향의 다양한 최적화 알고리즘이 구축됨
- 병렬 처리 가능
- 간단하고 직관적인 API 제공

3 모델링 전략

과적합 방지

과대적합 (Overfitting)

모델의 복잡도가 지나치게 높아지면 모델링 과정에서 모델의 학습 데이터를

모조리 정확하게 예측하려고 하는 상황 발생

⋮

Train data에서는 좋은 성능을 보여줄지 몰라도 학습에 참여하지 않은

Test data를 예측할 때에는 좋지 않은 성능을 보일 수 있음

과적합 방지

과소적합 (Underfitting)

모델이 너무 단순한 경우,

데이터의 패턴을 충분히 학습하지 못해 성능이 좋지 않을 수 있는 상황

⋮

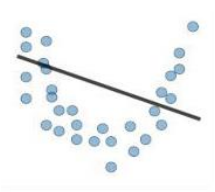
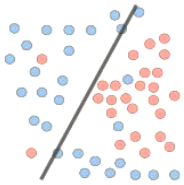
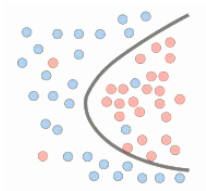
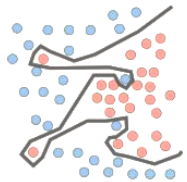
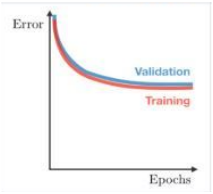
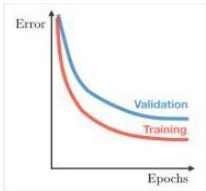
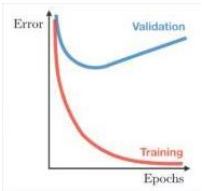
파라미터가 더 많은 복잡한 모델을 선택하거나,

데이터의 개수를 늘리는 방법 등으로 해결 가능!

3 모델링 전략

과적합 방지

클린업에서는
과대적합에 집중!

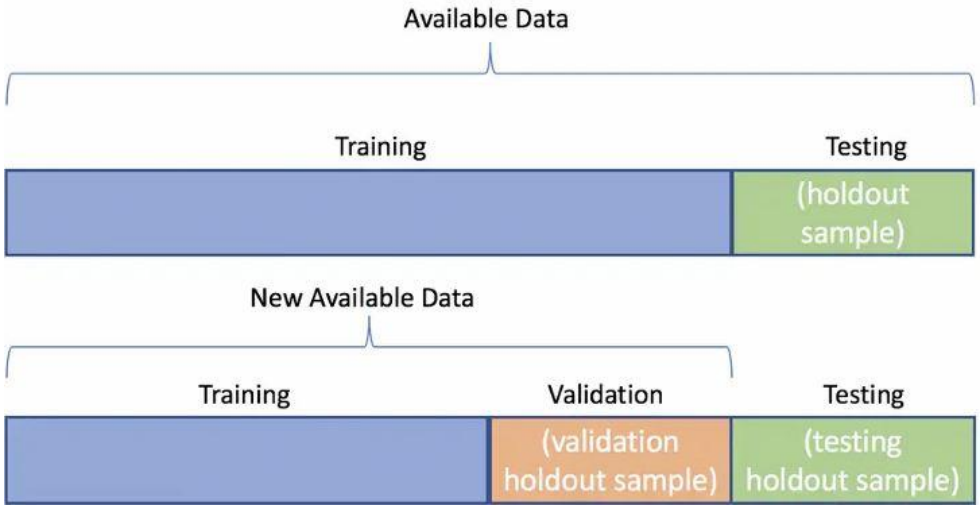
	과소적합	적합	과대적합
회귀적 표현			
분류적 표현			
딥러닝적 표현			

교차 검증

교차 검증 (Cross Validation)

학습에 사용되는 Train data를 다시

Train / Validation data로 나누어 **모델의 성능을 미리 검증**하는 방법



3 모델링 전략

교차 검증

교차 검증 (Cross Validation)

학습에 사용되는 Train data를 다시

Train / Validation data로 나누어 모델의 성능을 미리 검증하는 방법

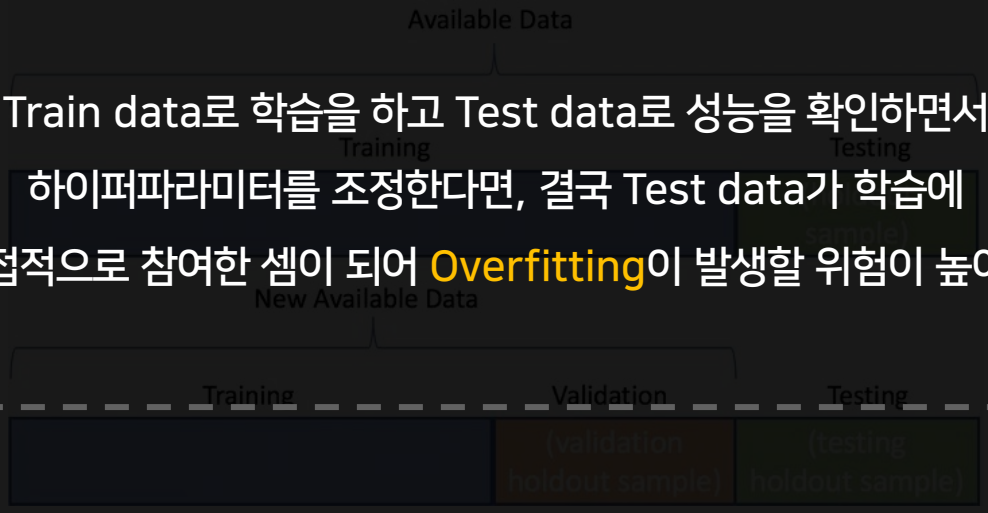
Q : Train data를 Train, Valid로 다시 쪼개는 이유?

A : 모델링 과정에서 하이퍼파라미터를 조정해야 하기 때문!

Train data로 학습을 하고 Test data로 성능을 확인하면서

하이퍼파라미터를 조정한다면, 결국 Test data가 학습에

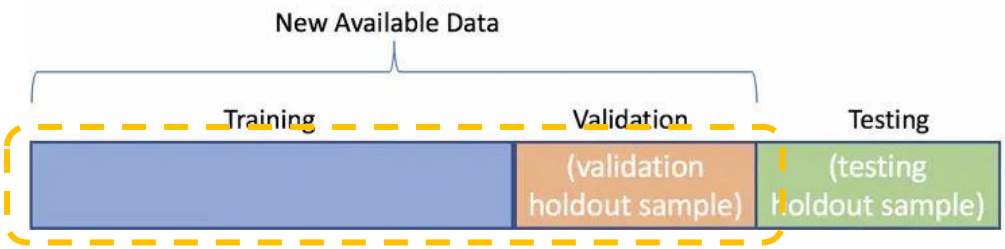
간접적으로 참여한 셈이 되어 Overfitting이 발생할 위험이 높아짐



3 모델링 전략

교차 검증

Validation data를 생성하는
가장 단순한 방법은 Train data를 7:3 혹은 8:2로 쪼개기
(Hold-out Validation)



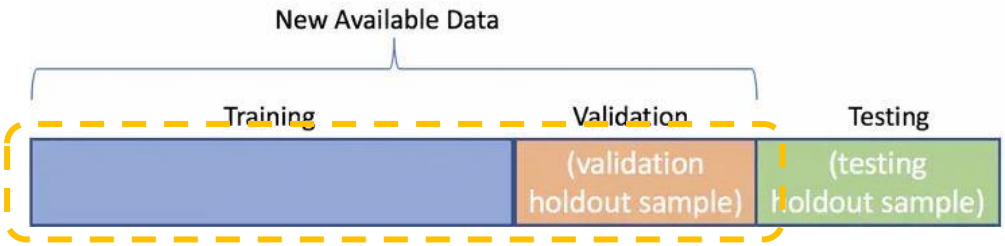
3 모델링 전략

교차 검증

하지만, 그림처럼 단일한 Validation set을 생성하는

방법은 모델의 성능을 검정하는 최적의 방법이라고 할 수 없음

1. Validation set이 전체 데이터의 경향성을 충분히 포함하지 않는다면 모델이 왜곡됨
2. 학습에 참여할 데이터가 줄어듦 (The smaller the worse)



3 모델링 전략

교차 검증

1. Validation set이 전체 데이터의 경향성을 충분히 포함하지 않는다면 모델이 왜곡됨
2. 학습에 참여할 데이터가 줄어듦 (The smaller the worse)

⋮

이러한 문제를 해결하기 위해 Validation set을

특정 부분으로 고정시키지 않고 데이터의 **모든 부분을 번갈아 가며 사용!**

3 모델링 전략

교차 검증

교차 검증(Cross Validation)

학습에 사용되는 Train data를 다시

Train / Validation data로 나누어 **모델의 성능을 미리 검증**하는 방법

교차 검증의 장점

1. 특정 데이터셋에 대한 **과대적합 방지**
2. 더욱 **일반화된 모델 생성** 가능
3. 데이터셋 규모가 적을 시 **과소적합 방지**

교차 검증의 단점

1. 모델 훈련 및 평가 **소요시간 증가**

3 모델링 전략

교차 검증

교차 검증(Cross Validation)

학습에 사용되는 Train data를 다시
Train / Validation data로 나누어 **모델의 성능을 미리 검증**하는 방법

교차 검증의 장점

1. 특정 데이터셋에 대한 **과대적합 방지**
2. 더욱 **일반화된 모델 생성** 가능
3. 데이터셋 규모가 적을 시 **과소적합 방지**

교차 검증의 단점

1. 모델 훈련 및 평가 **소요시간 증가**

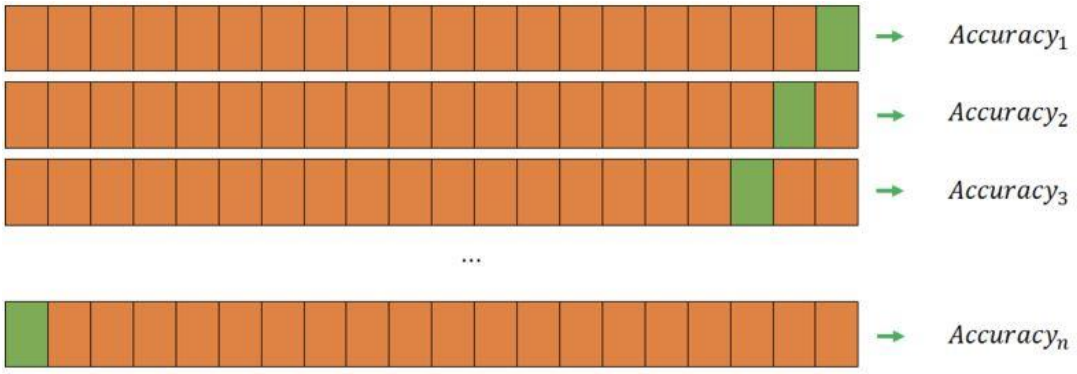
이제 교차 검증의
방법을 알아보자!

교차 검증

LOOCV (Leave-One-Out Cross-Validation)

전체 데이터 **n개 중에서 단 1개만**을 Validation set으로,

나머지 n-1개는 Train set으로 사용하는 방식



bias ↓ variance ↑

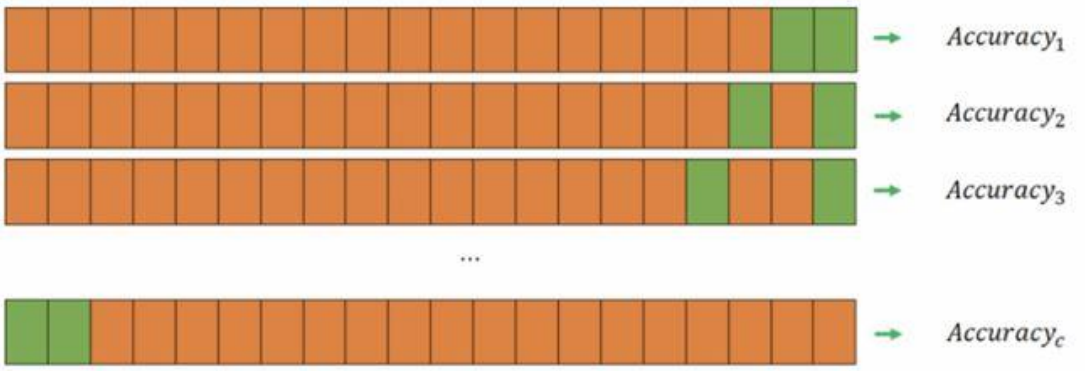
n번의 교차 검증이 실행되므로 **연산량이 많다**는 단점 존재 → 작은 데이터셋 권장

교차 검증

LpOCV (Leave-p-Out Cross-Validation)

LpOCV는 전체 데이터 **n개 중 p개만**을 Validation set으로,

나머지 n-p개는 Train set으로 사용하는 방식



이 방법은 더 정교한 교차검증이 가능하지만

Iteration 횟수가 $\binom{n}{p}$ 번으로 **연산량이 훨씬 많아짐** → 작은 데이터셋 권장

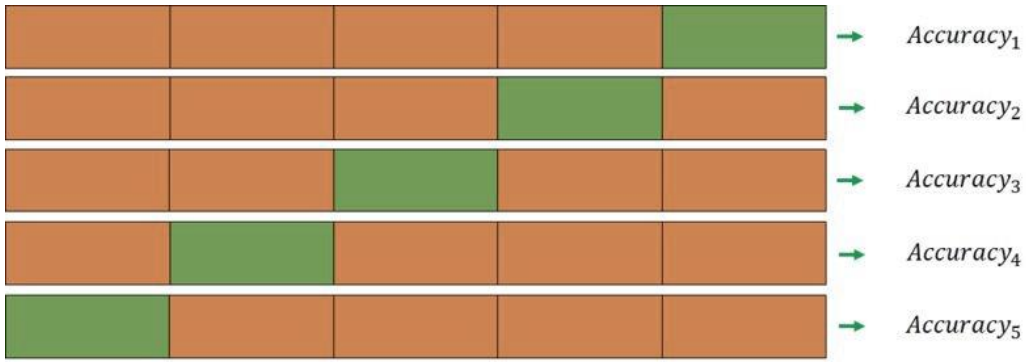
3 모델링 전략

교차 검증

K-Fold CV

전체 데이터 셋을 k개의 그룹(fold)으로 분할하여

1개 그룹은 Validation set, **나머지 k-1개 그룹**은 Train set으로 사용하는 방식



중간 정도의 bias & variance, LOOCV보다 연산량 ↓

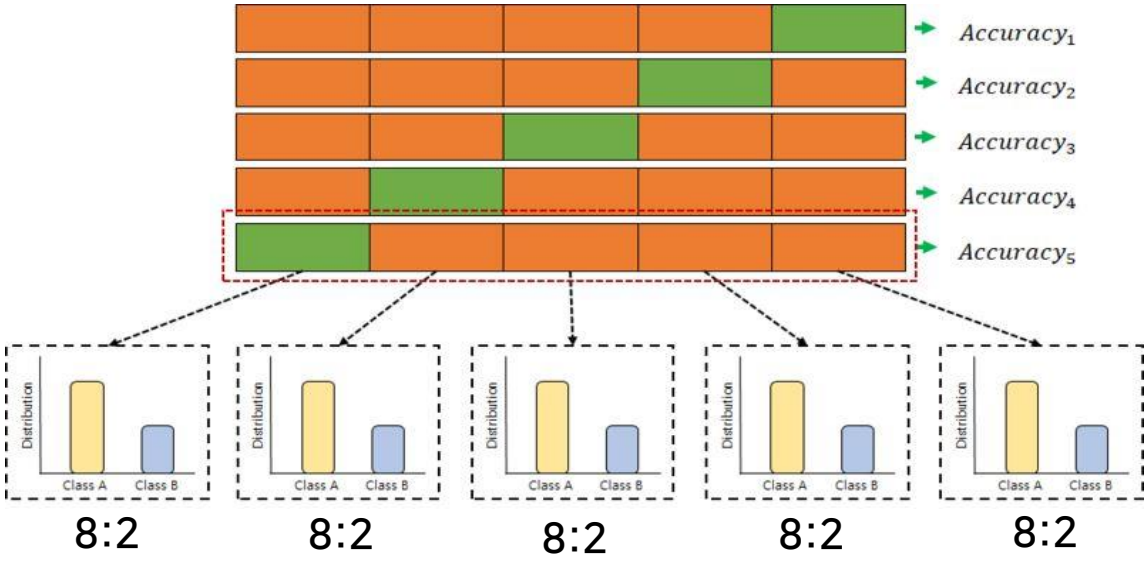
K : 사용자가 임의로 결정하는 값, 경험적으로 5~10 사용

3 모델링 전략

교차 검증

Stratified K-Fold CV

불균형 데이터에서 전체 클래스의 비율까지 고려하여 검증하는 방법



Stratified K-Fold를 사용하여 전체 클래스 비율을 고려하고

데이터를 적절히 분배하여 데이터셋을 구성하는 것이 좋음

3 모델링 전략

교차 검증

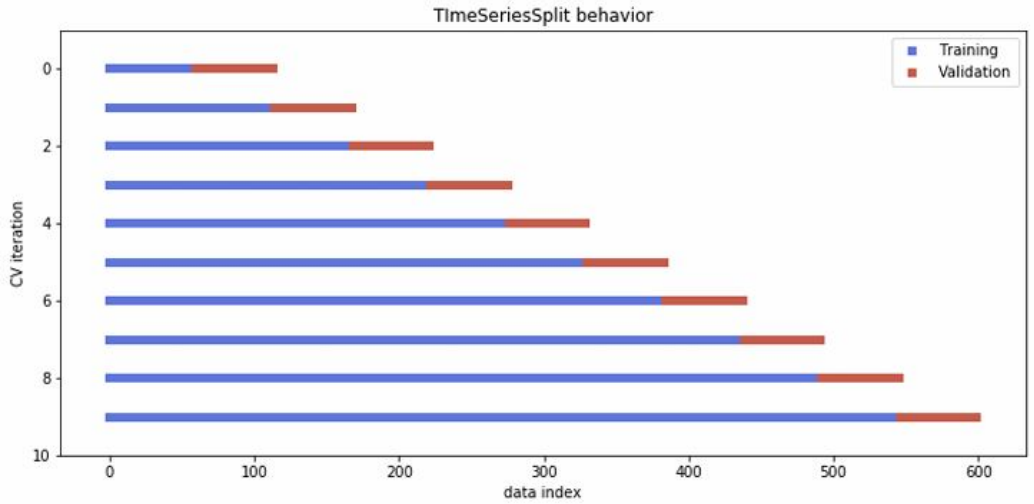
Time Series CV

시계열 데이터는 전후 데이터 사이의

상관관계가 존재하므로 기존 교차 검증 방법 적용 불가

⋮

Train set이 Validation set 보다 **항상 앞선 시간으로 할당!**

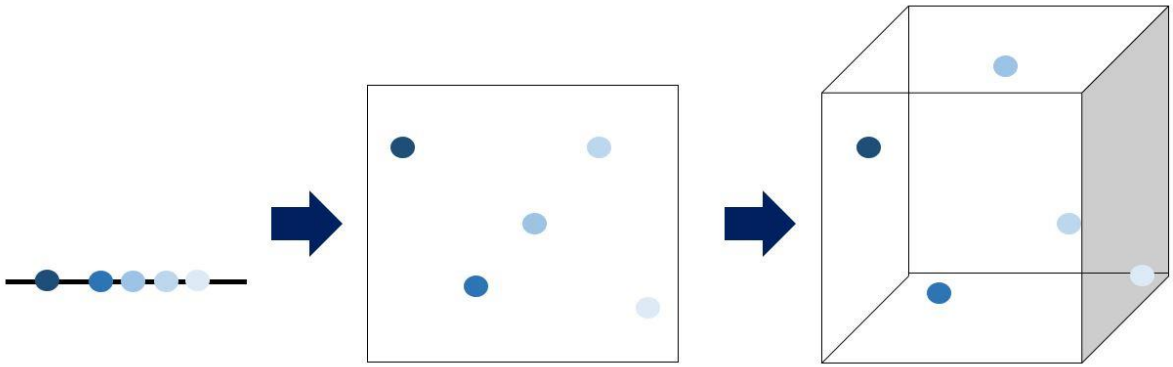


차원의 저주

차원의 저주 (Curse of Dimensionality)

학습 데이터 수가 늘어나면서 **차원의 수**가 늘어나고,

데이터의 특징이 너무 많아져서 **과적합**이 일어나는 문제



Made by: ta-daa

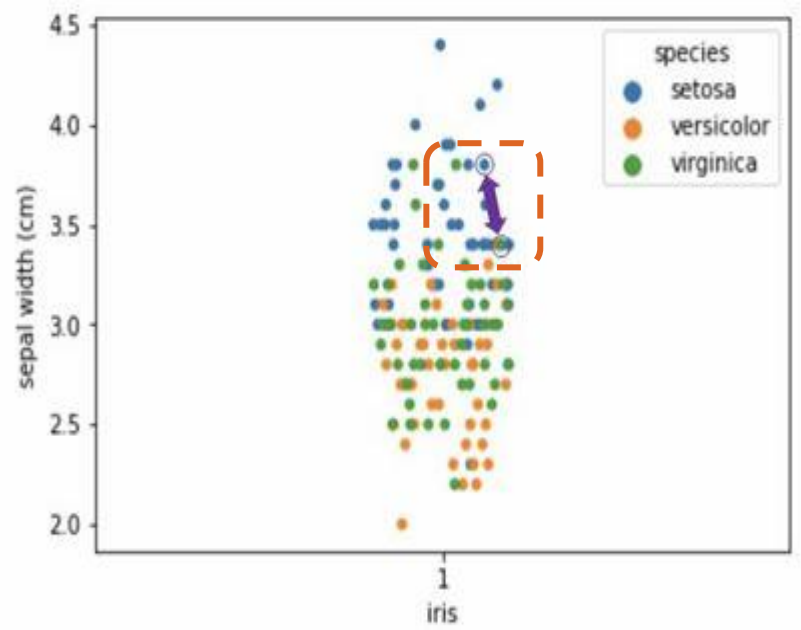
변수가 증가한다고 해서 무조건 차원의 저주가 발생하는 것이 아니라

변수의 개수가 관측치의 개수보다 많아지면 발생

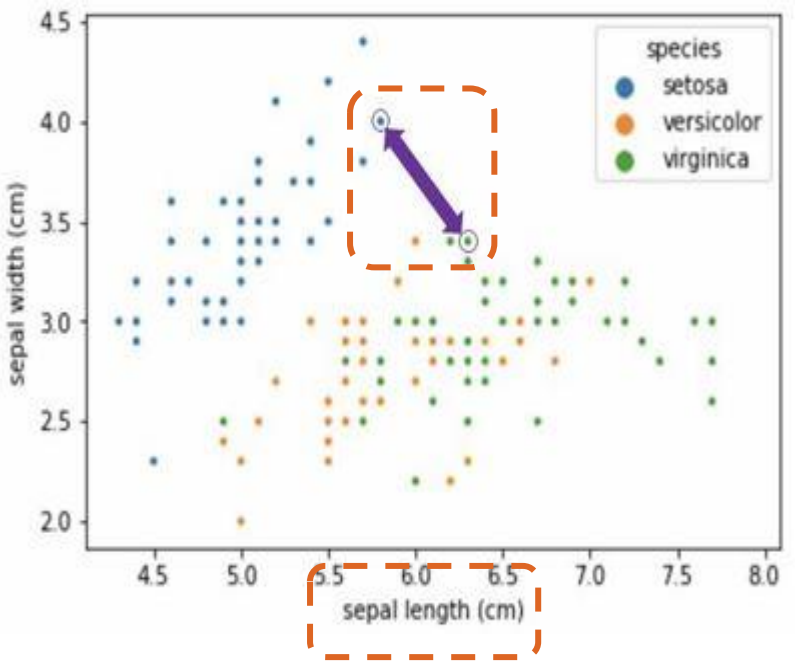
3 모델링 전략

차원의 저주

ex) KNN



변수 개수 1개



+ “Sepal Length” 변수

3 모델링 전략

차원의 저주

변수가 추가됨에 따라

두 점 간 거리가 멀어짐

⋮

차원이 너무 큰 것이 문제 였으니,

반대로 **차원을 축소**해서 문제를 해결할 수 있음!

3 모델링 전략

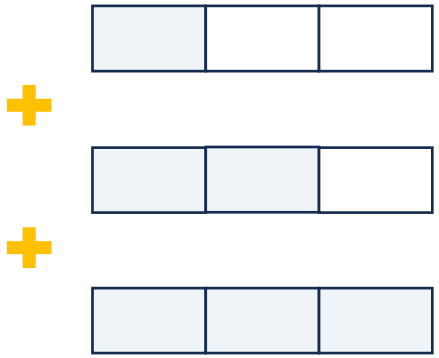
차원의 저주

변수선택법

데이터를 예측하는 데

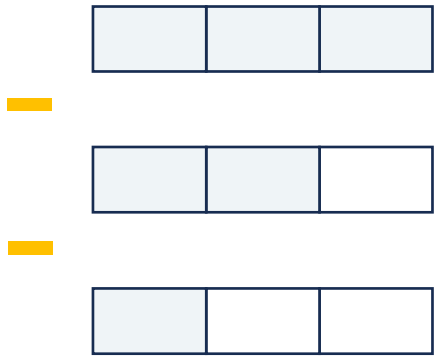
중요한 변수들만 선택하고 쓸모없는 변수들은 버리겠다는 아이디어

전진선택법



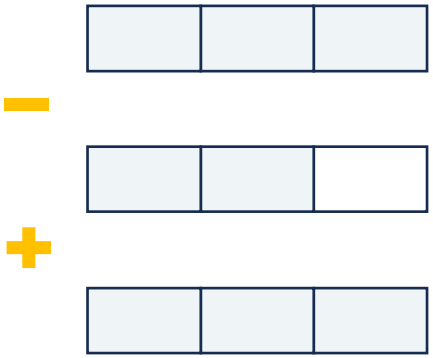
변수를 더해감

후진선택법



변수를 제거함

단계적선택법



전진, 후진 거듭

변수 선택 시 모델의 적절성을 평가하는 지표

$R^2_{adj}, AIC, BIC, Mallow's\ CP$ 등

3 모델링 전략

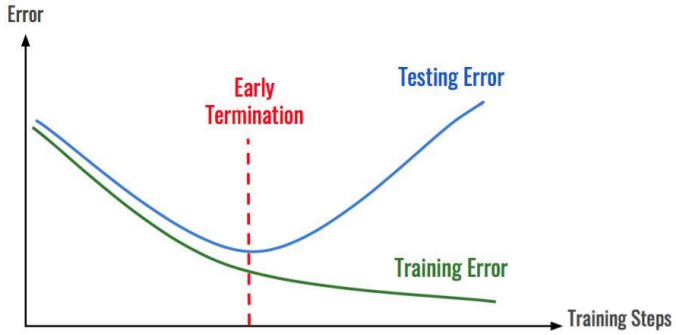
차원의 저주

변수추출법

고차원의 데이터를 저차원 공간의 데이터로 변환하는 것으로
이 과정을 거치고 나면 변수의 종류와 값 자체가 바뀌게 됨 (ex. PCA)

조기 종료 (Early Stopping)

학습에 소요되는 시간에 제한을 두거나
모델 성능이 일정 수준 이상이 되면 자동으로 종료하는 조건을 설정하는 방법



Test error 가 다시 증가하기 전에 학습을
종료하여 과적합을 방지할 수 있음

다음 주 예고

1. 트리 기반 모델
2. 앙상블 기법
3. 비선형 모델

THANK YOU

❤ 르데마핌 파이팅 ❤