데이터마이닝팀

4팀
성준혁
노정아
이상혁
진재언
한준호

INDEX

1. 데이터마이닝 소개

2. 모델링

3. 모델링 전략

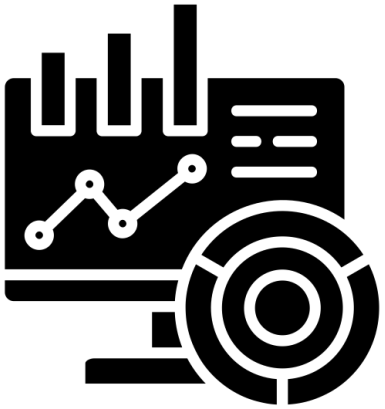
1

데이터마이닝 소개

1 데이터마이닝 소개

정의

Data(데이터)



Mining(채굴)

방대한 데이터로부터 **유용한 정보**를 추출해내는 과정

Exploration, Pattern Identification, Deployment의 과정으로 이루어짐

1 데이터마이닝 소개

정의

Exploration

데이터를 분석에 용이한 형태로 전처리하는 것

Pattern Identification

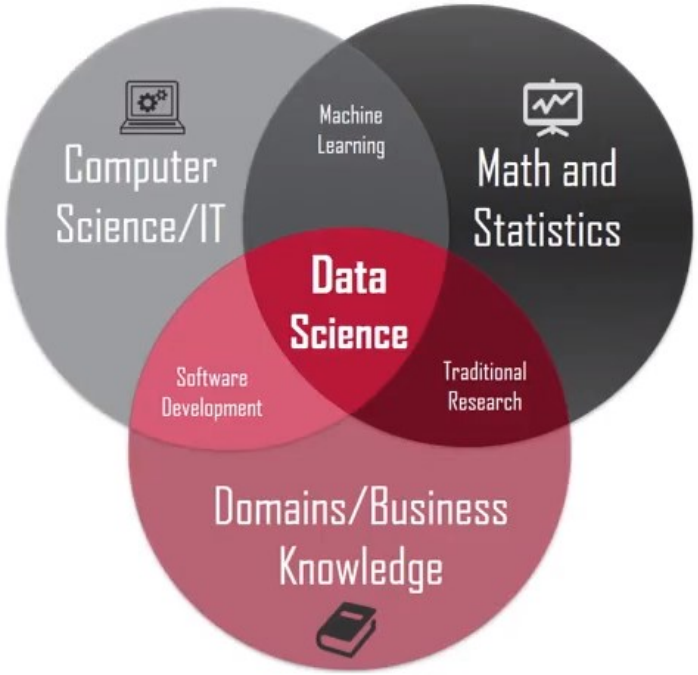
회귀나 분류 등의 방법으로 패턴을 찾는 것

Deployment

찾아낸 패턴을 활용하여 목적에 맞는 결과물을 도출

1 데이터마이닝 소개

학제적 위치

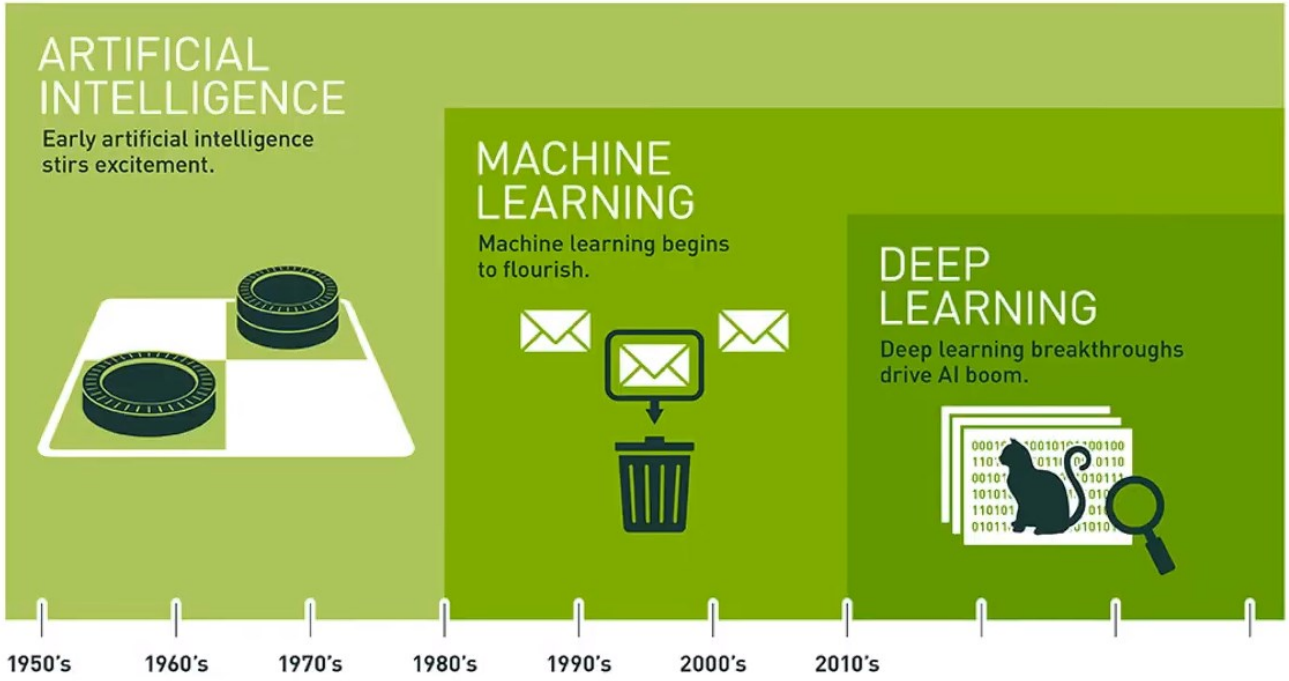


데이터마이닝은 통계학, 컴퓨터 과학, 머신러닝 등

여러 학문을 넘나드는 **간학문적 성격**을 띰

1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝



데이터마이닝의 **간학문적** 특성상,

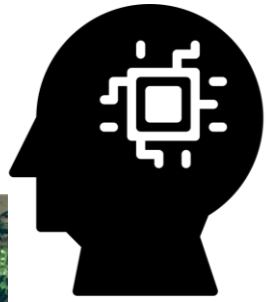
인공지능, 머신러닝, 딥러닝 등의 개념이 혼동될 수 있음

1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝

인공지능

사람의 일을 기계가 대신하는 모든 자동화 과정을 일컫는 것으로
머신러닝과 딥러닝을 포괄하는 개념



단, 모든 AI가 학습을 하는 것은 아님

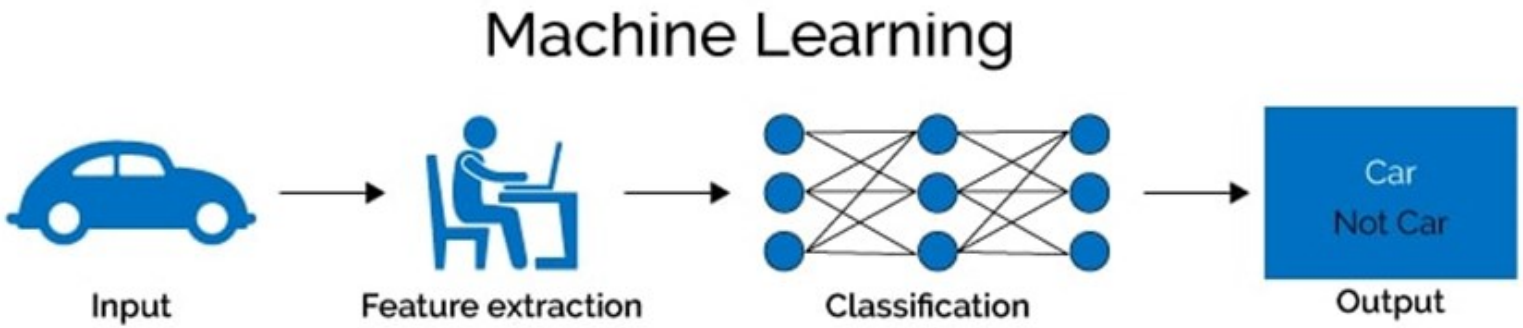
Ruled-based AI처럼
인간이 명시적으로 rule을
프로그래밍한 경우도 있음
(Ex. IBM의 체스 AI '딥블루')

1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝

머신러닝(ML)

명시적인 프로그래밍 없이 기계가 **스스로 패턴을 학습**하는 방법



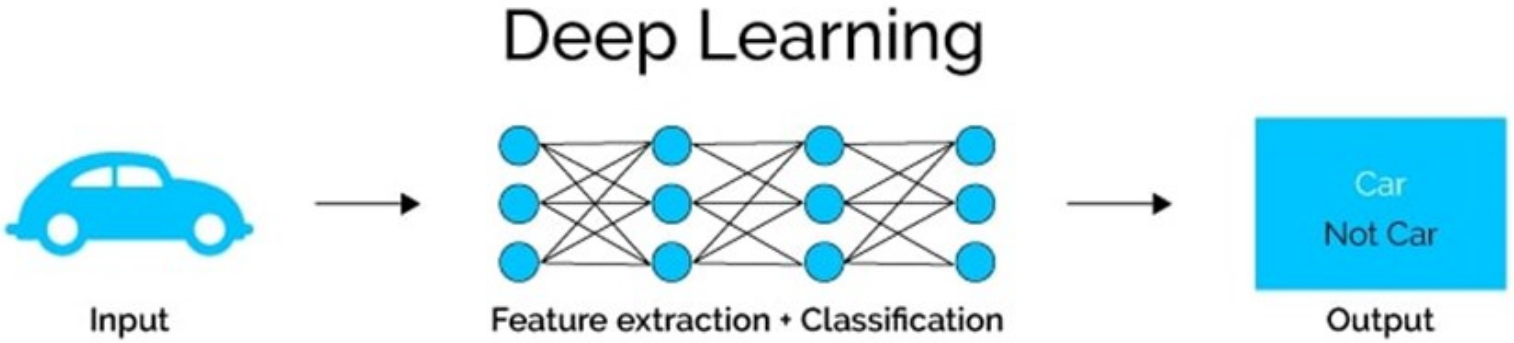
1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝

딥러닝(DL)

머신러닝 알고리즘 중 **인공신경망을 기반**으로 고도화된 모델

주로 자연어나 이미지 데이터 같은 비정형 데이터에서 잘 작동함!



1 데이터마이닝 소개

인공지능 vs 머신러닝 vs 딥러닝

딥러닝(DL)

데이터마이닝은 어디에 해당할까?

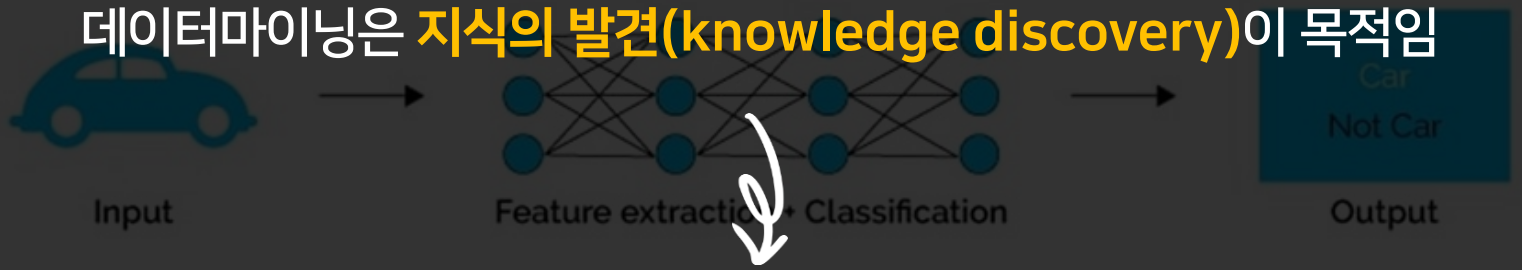
머신러닝 알고리즘 중 인공신경망을 기반으로 한 고도화된 모델

주로 [illegible] 동함!

방법론적으로는 **머신러닝**과 동일한 개념임

그러나 머신러닝은 **예측(prediction)**이 목적인 반면,

데이터마이닝은 **지식의 발견(knowledge discovery)**이 목적임



데이터마이닝은 지식의 발견을 위해 머신러닝을 사용함!

CRISP-DM 방법론

CRISP-DM 방법론

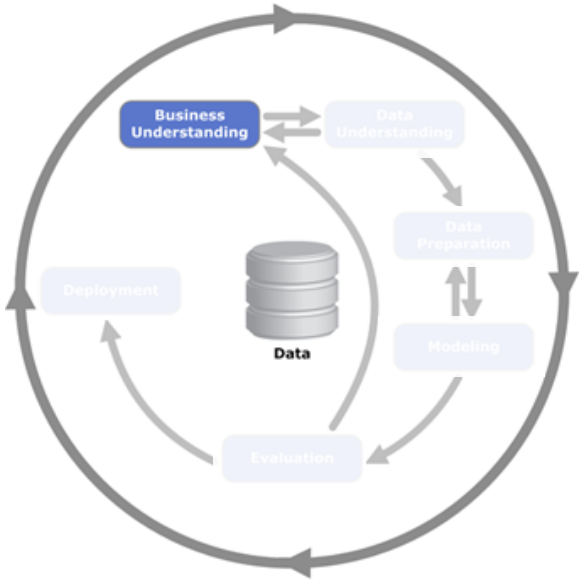
Cross-Industry Standard Process for Data Mining

데이터마이닝 프로젝트를 **6단계로 계획**하도록 하는 방법론

CRISP-DM 방법론

1단계 : 비즈니스 문제 이해

분석을 수행하고자 하는 **과제의 목적**과 **요구사항**을 이해
이를 활용하여 초기 프로젝트의 계획을 수립하는 단계



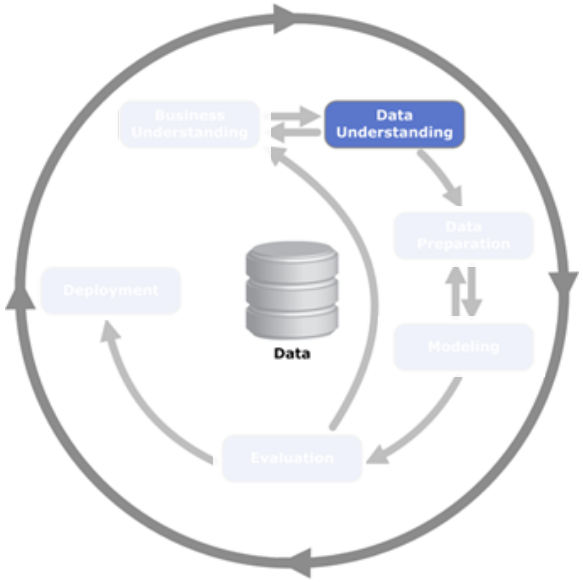
- 비즈니스 관점 이해
- 상황조사 및 영향 평가
- 프로젝트 목표 및 계획 수립

CRISP-DM 방법론

2단계 : 데이터 이해

탐색적 데이터 분석(EDA)을 진행하며 데이터를 뜯어봄

분석을 위해 데이터를 수집하고 이를 이해하는 단계



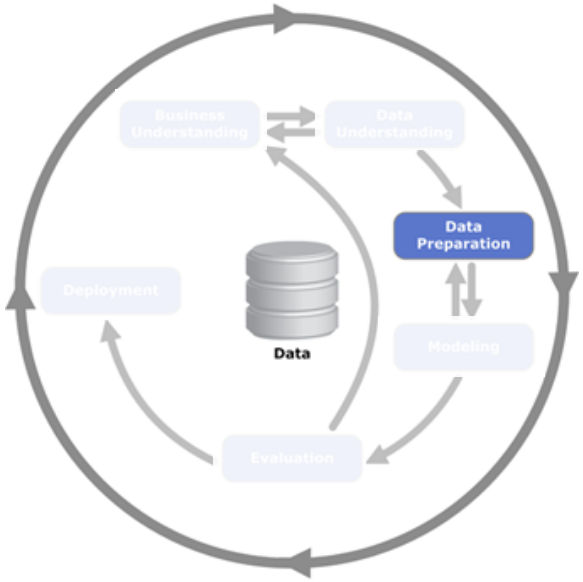
- 데이터 확보
- 데이터 특성 확인 및 품질 검사
- 목표 부합 데이터 추출

CRISP-DM 방법론

3단계 : 데이터 준비

결측치, 이상치 등을 처리해서 분석 목적에 맞게 **데이터를 다듬어야 함**

수집한 데이터를 분석에 용이한 형태로 전처리 하는 단계



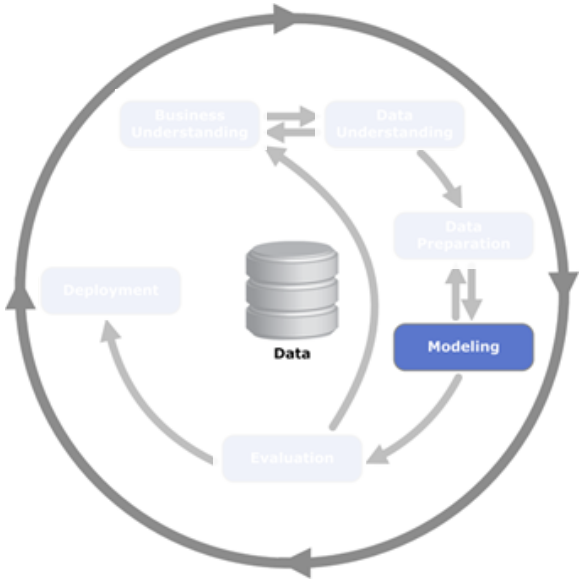
- 데이터 정제 및 품질 확보
- 분석 대상 데이터 구조화
- 데이터 통합

CRISP-DM 방법론

4단계 : 분석 및 모델링

적절한 알고리즘을 바탕으로 **모델을 학습**시킴

파라미터 등을 고려하여 본격적으로 모델링을 수행하는 단계



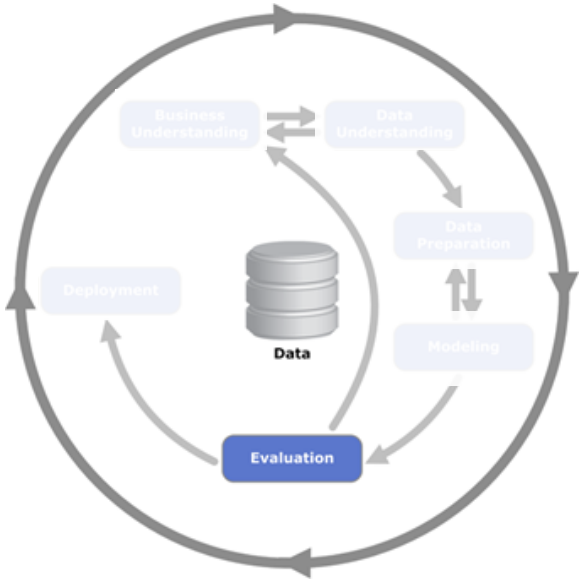
- 모델링 기법 선택
- 유효성 검사
- 모델링 순위 지정

CRISP-DM 방법론

5단계 : 평가

이전 단계에서 진행한 모델의 **성능을 평가**하는 단계

분류는 misclassification rate, F beta-score, 회귀는 RMSE, MAE 등을 사용



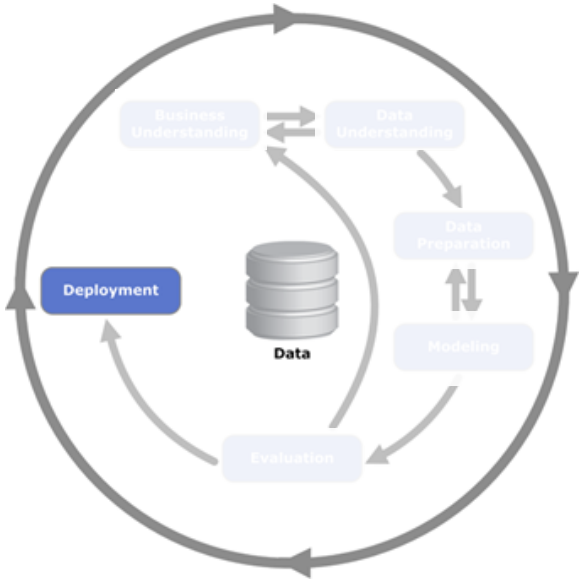
- 모델링 성능 평가
- 주요 항목 리뷰
- 다음 단계 결정

CRISP-DM 방법론

6단계 : 전개

분석 결과를 통해 **업무를 수행**하고 **구체적인 결과물**을 구현함

분석한 내용을 바탕으로 유의미한 결론을 이끌어내는 단계



- 전개 전략 수집
- 유지보수 전략 수립
- 최종보고서 작성 및 검토

2

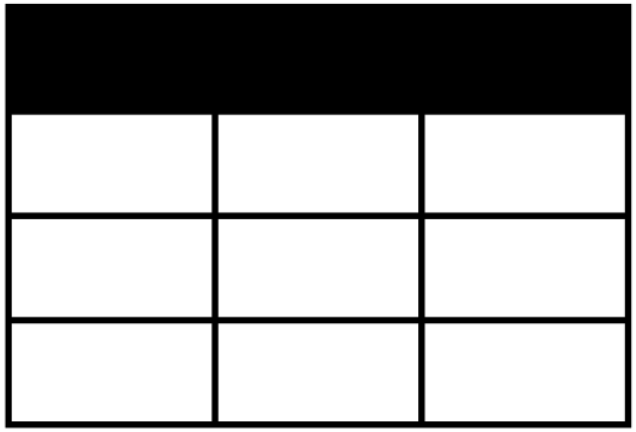
모델링

2 모델링

데이터 정의 - 정형/비정형

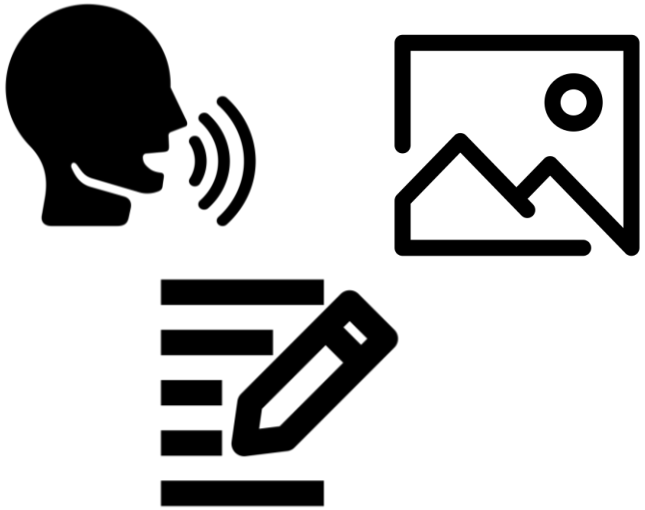
정형 데이터

미리 **정해진 구조**에 따라 저장된
데이터로, 표에 숫자가 담긴 형태임



비정형 데이터

구조화되지 않은 형태로
이미지, 음성, 텍스트 등의
정보로 이루어짐



데이터 정의 – 종속변수/독립변수

관측값은 행(row), 변수는 열(column)로 나타냄

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

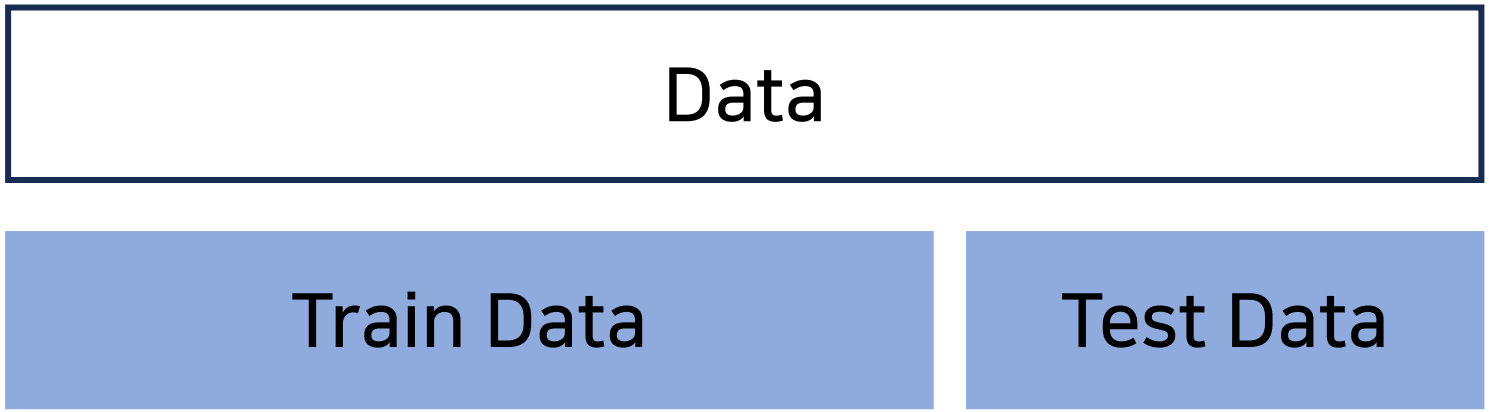
독립변수

종속변수

예측의 대상이 되는 변수(ex. Sales Price)를 **종속변수(y)**라고 부름

종속변수를 예측하는데 쓰이는 다른 변수를 **독립변수(x)**라 부름

데이터 정의 – 훈련/테스트



Train Data

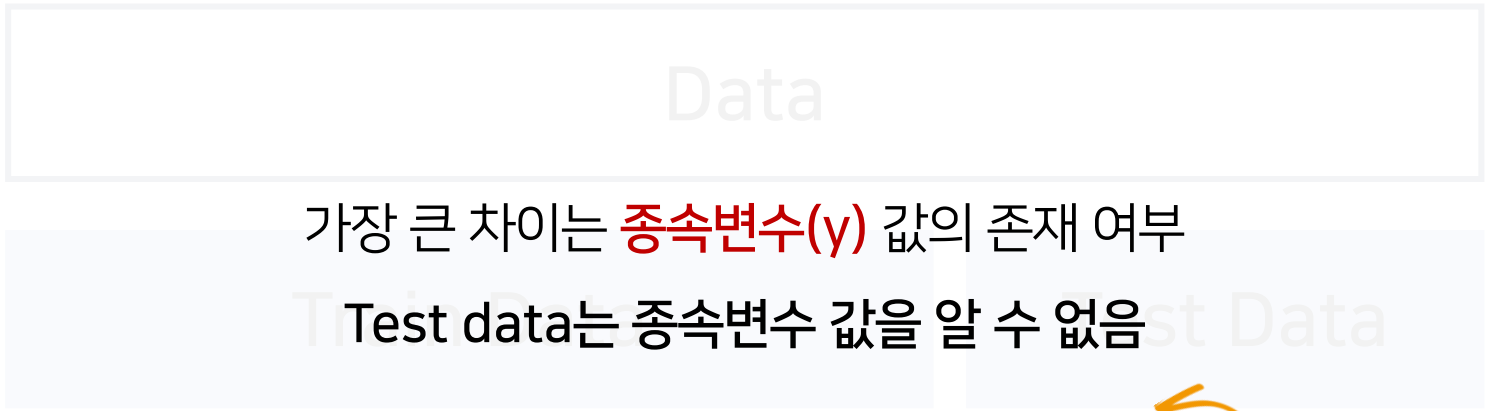
Train data는 모델을
학습시키기 위한 데이터

Test Data

Test data는 학습된 모델을
평가하기 위한 데이터

2 모델링

데이터 정의 – 훈련/테스트



가장 큰 차이는 **종속변수(y)** 값의 존재 여부

Test data는 종속변수 값을 알 수 없음

Train Data

Train data는 모델을
학습시키기 위한 데이터

VS

Test Data

Test data는 학습된 모델을
평가하기 위한 데이터

데이터 정의 – 훈련/테스트

Data

즉, 모델이 Unseen data에서도

잘 작동하는지 평가하기 위해 사용

Train Data

Train data는 모델을

학습시키기 위한 데이터

Test Data

Test data는 학습된 모델을

평가하기 위한 데이터

2 모델링

데이터 정의 – 훈련/테스트

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

[Train]

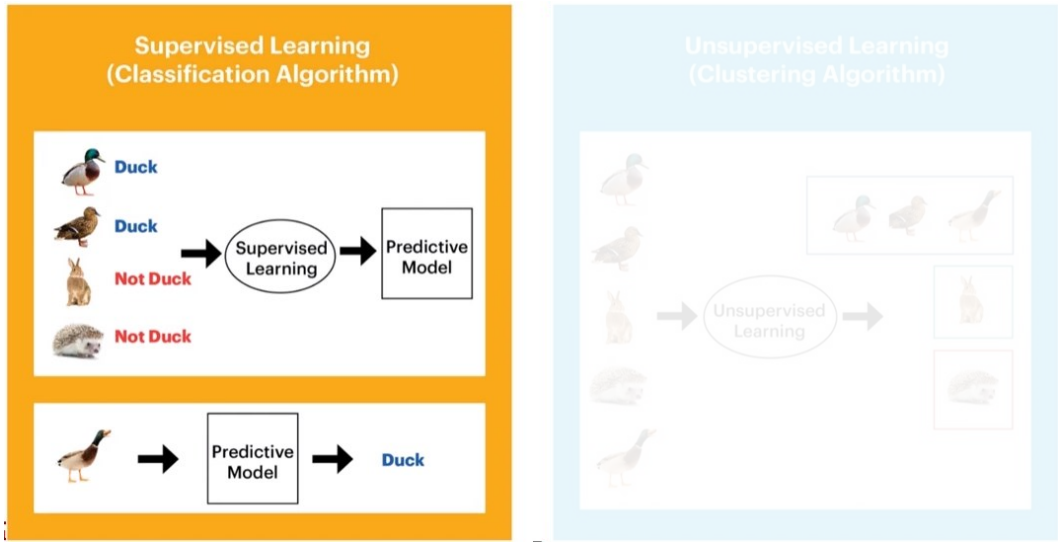
Bedrooms	Sq.feet	Neighborhood	Sales Price
3	2000	Hipstertown	???

[Test]

Train data를 학습한 모델로 Test data의 Sales Price를 예측

2 모델링

모델링(머신러닝)의 종류 - ① 정답 유무에 따라

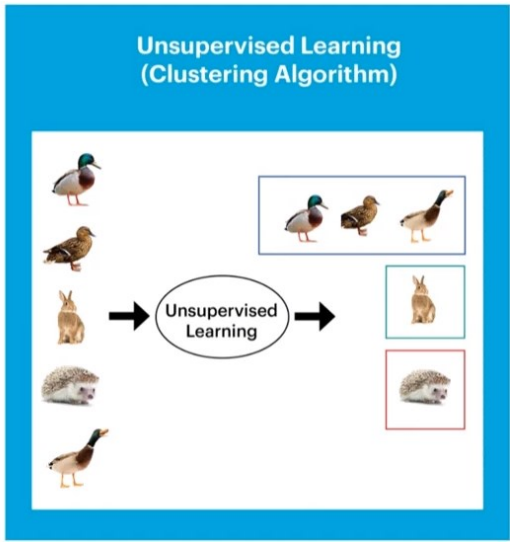


지도학습

입력변수(X)와 출력변수(Y)가 정해져 있어,

둘 사이의 관계를 규명하는 것에 초점을 두는 방법

모델링(머신러닝)의 종류 - ① 정답 유무에 따라

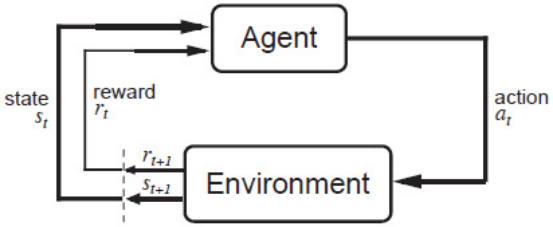


비지도학습

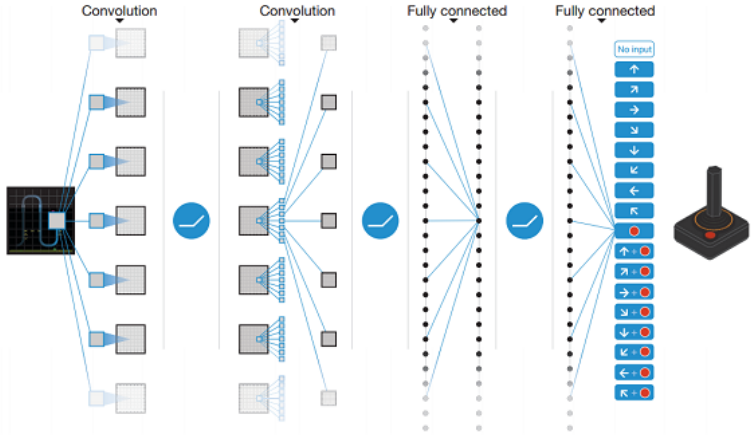
출력변수(Y)가 없는 데이터의 특성이나 구조를
파악하는 것에 초점을 두는 방법

2 모델링

모델링(머신러닝)의 종류 - ① 정답 유무에 따라



AlphaGo



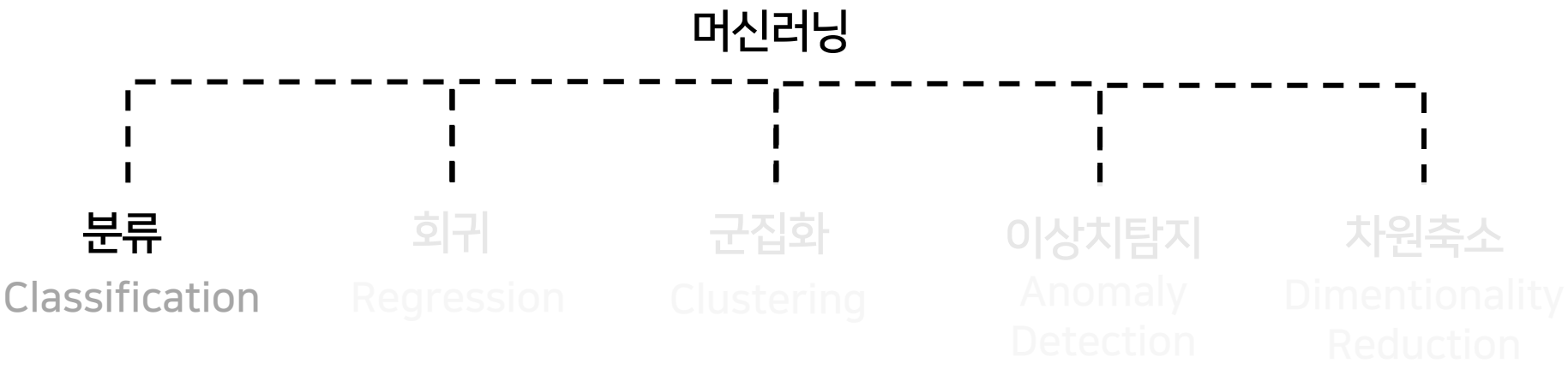
강화학습

보상을 최대화하고 페널티를 최소화하는 방향으로 학습하는 방법

알파고를 만드는 데에 쓰인 것으로 유명함

본 클린업에서는 다루지 않을 예정!

모델링(머신러닝)의 종류 - ② 학습 목적에 따라

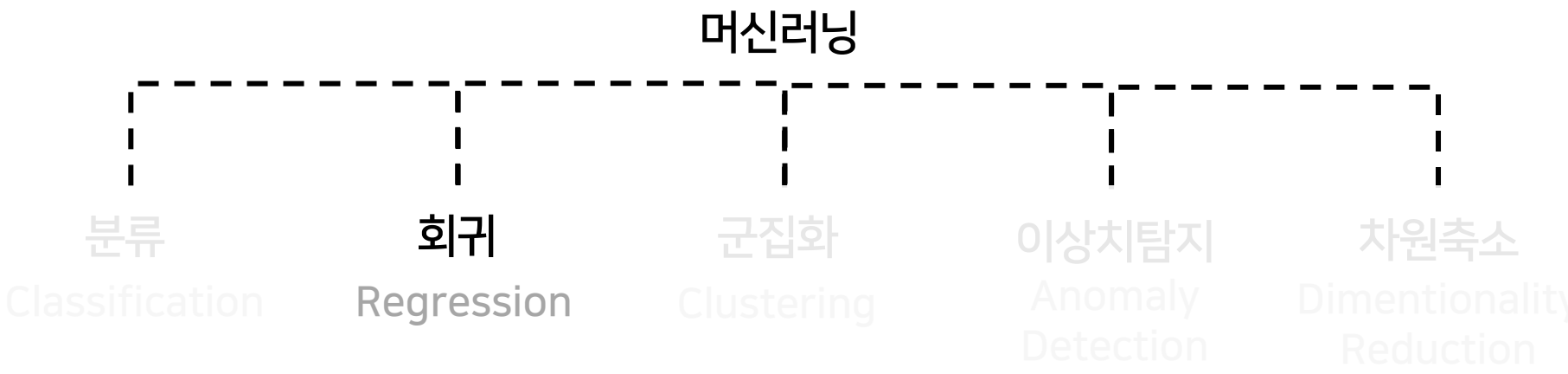


분류

종속변수가 어떤 카테고리에 해당하는지 예측하는 것이 목적으로,

범주형(categorical) 변수를 예측하는 **지도학습 방법론**임

모델링(머신러닝)의 종류 - ② 학습 목적에 따라

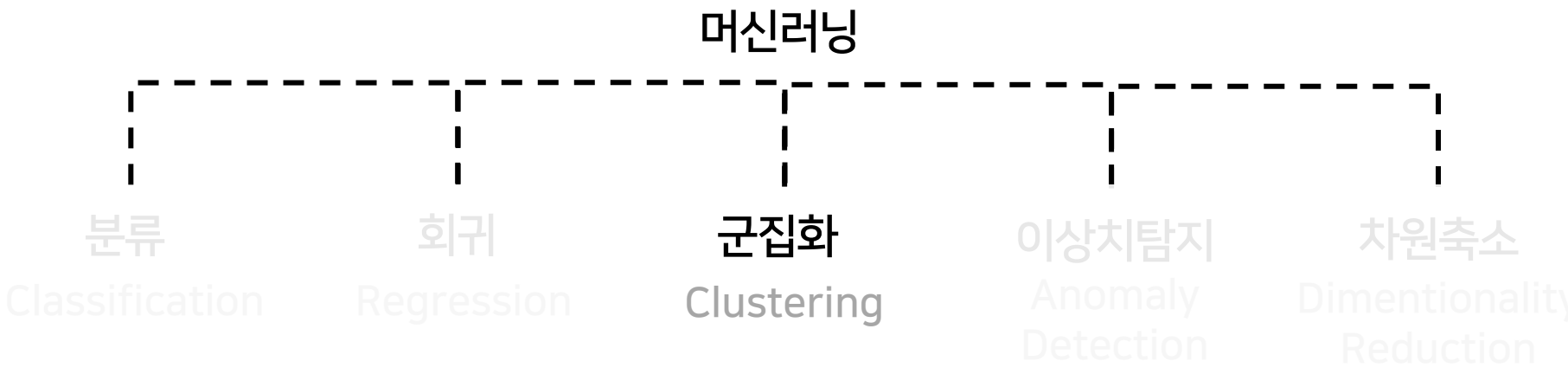


회귀

종속변수가 어떤 값을 갖는지 예측하는 것이 목적으로,

연속형(continuous) 변수를 예측하는 **지도학습 방법론**임

모델링(머신러닝)의 종류 - ② 학습 목적에 따라

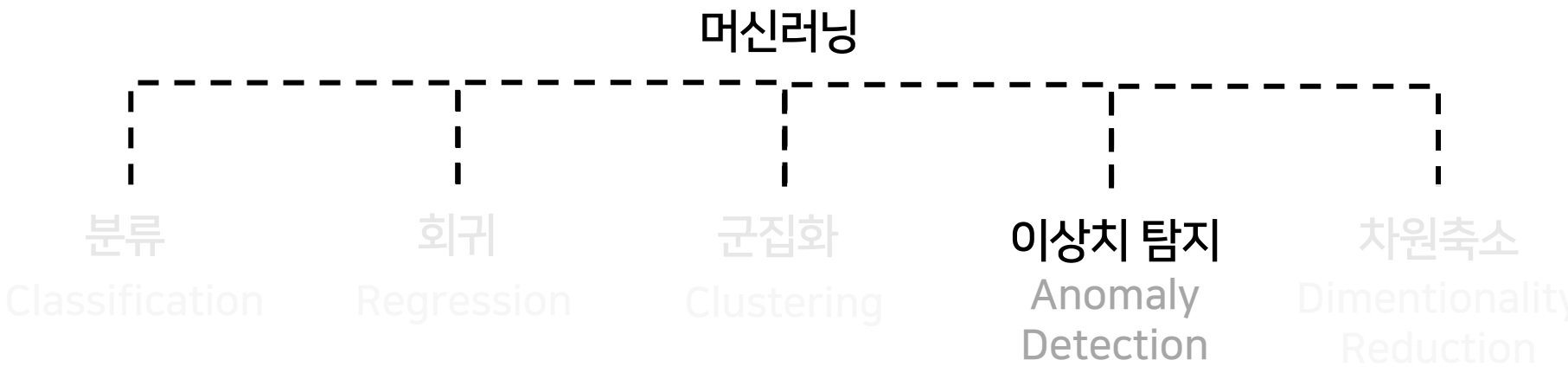


군집화

유사한 개체들의 집단을 판별하는 것이 목적으로,

정답이 없는 **비지도학습 방법론**임

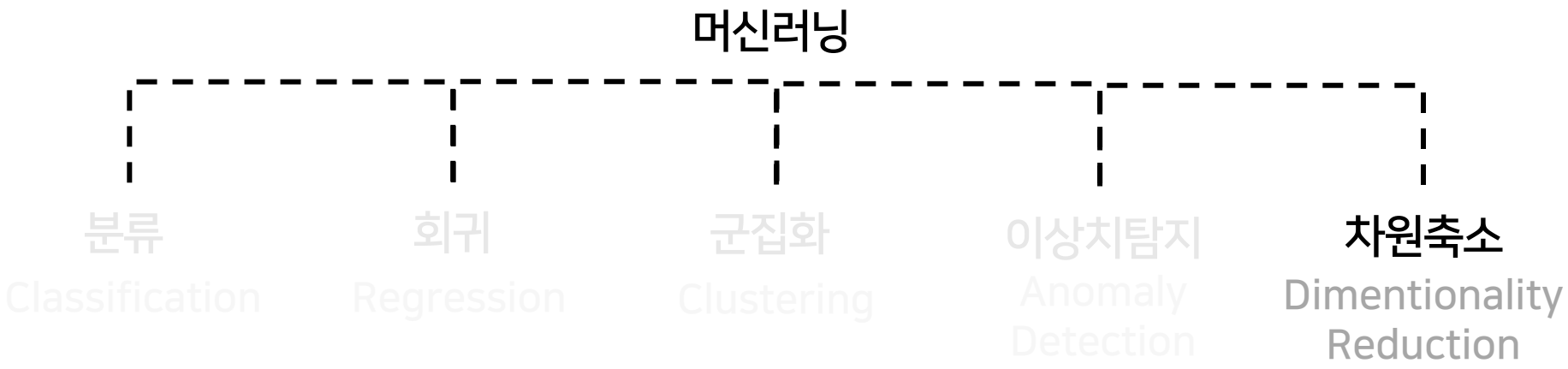
모델링(머신러닝)의 종류 - ② 학습 목적에 따라



이상치 탐지

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

모델링(머신러닝)의 종류 - ② 학습 목적에 따라



차원축소

고차원 데이터를 저차원으로 간소화하는 것이 목적으로, **비지도학습 방법론**임

지도학습의 기본 원리

지도학습

$$Y = f(X) + \epsilon$$

X: 독립변수 (Bedrooms, Sq.feet, Neighborhood…)

Y: 종속변수

f: 독립변수와 종속변수의 관계를 설명하는 함수

ϵ: 랜덤한 오차 (현실 세계의 변칙적 요인들)

지도학습의 기본 원리

지도학습

$$Y = f(X) + \epsilon$$

실제 Y값에 근접한 추정치를 찾기 위해

함수 f를 추정해 나가는 과정

2 모델링

지도학습의 기본 원리

지도학습

함수 f는 왜 추정하는가?

$$Y = f(X) + \epsilon$$

예측 (Prediction)

X값을 통해 Y값을 예측하기 위해

예측 오차(Reducible error)를 줄이는 것

= 더 좋은 모델을 선택하는 것

추론 (Inference)

X와 Y의 관계를 파악하기 위해

변수를 선택하거나 실용적인 해석을 얻는 것

= 정확한 형태를 알아야 함 (양/음, 선형/비선형…)

실제 Y값에 근접한 추정치를 찾기 위해

함수 f를 추정해나가는 과정

f 라는 함수를 추정하면, 이 2가지 goal을 모두 달성할 수 있음

2 모델링

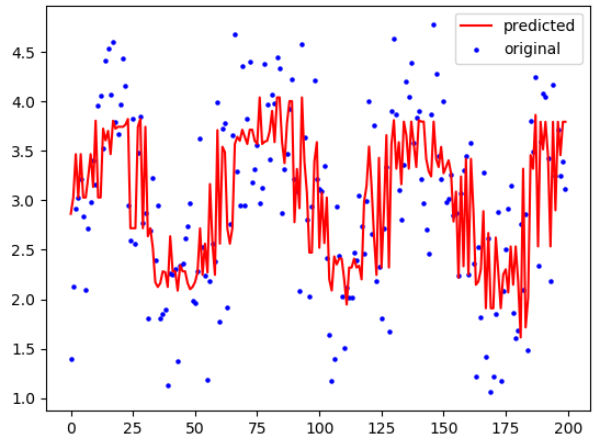
f를 추정하는 방법

$$f(X) = w_0 + w_1X_1 + w_2X_2$$

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

모수적 방법

f를 특정한 형태로 가정하는 방법



비모수적 방법

f의 형태를 가정하지 않는 방법

2 모델링

f를 추정하는 방법

$$f(X) = w_0 + w_1X_1 + w_2X_2$$

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

모수적 방법

f를 특정한 형태로 가정하는 방법

1) f를 추정하는 문제가
parameter를 추정하는 문제로 단순화됨

2) 모델링을 하고 나면
train data가 필요하지 않음

→ if 선형회귀라면, train data로 회귀계수(β) 학습 후
그 파라미터만으로 새 데이터에 대한 예측이 가능함

F의 형태를 가정하지 않는 방법

2 모델링

f를 추정하는 방법

$$f(X) = w_0 + w_1X_1 + w_2X_2$$

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

모수적 방법

f를 특정한 형태로 가정하는 방법

하지만 특정한 형태로 가정하는 만큼,

형태가 실제 f와 다르면 성능이 좋지 않을 수 있음

→ 더 flexible한 모델 선택해 해결할 수 있지만,

과적합을 초래할 수 있는 문제 존재

① 데이터(관측치 개수)가 적거나,

② 사전지식이 어느정도 있을 때 사용

2 모델링

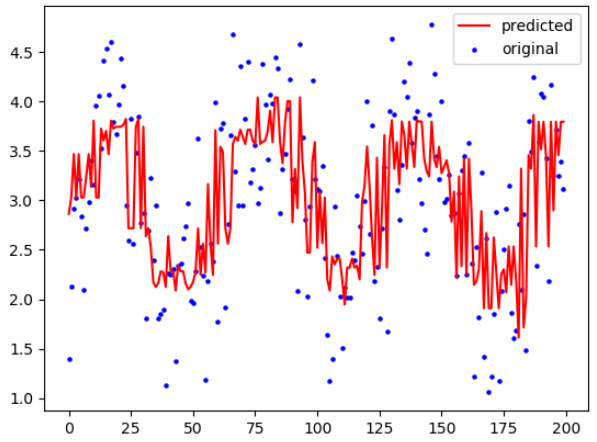
f를 추정하는 방법

데이터에만 의존해서 f를 추정함

Ex) 가까움을 기준으로 Y를 예측하는 KNN 모델

함수에 대한 복잡한 가정이 없어,
더 다양한 범위의 f 형태에 적합 가능

→ 모수적 방법이 특정한 형태 가정으로 가졌던 단점을 지니지 않음!



비모수적 방법

f의 형태를 가정하지 않는 방법

2 모델링

f를 추정하는 방법

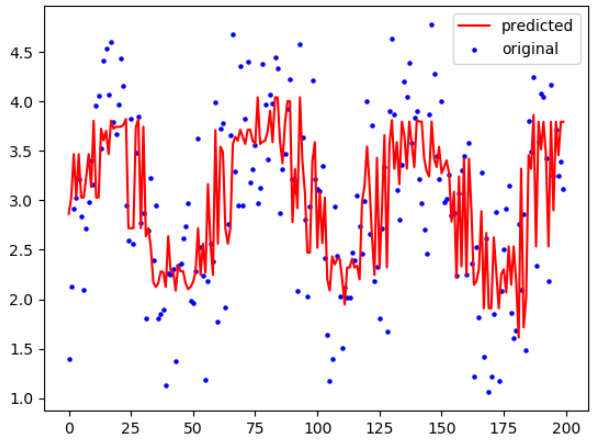
하지만 데이터에 의존해서 f를 추정하므로,

데이터가 충분히 많지 않으면 추정이 어려움

→ 적은 데이터에만 의존 시 과적합 위험이 커짐

① 데이터(관측치 개수)가 많거나,

② 사전지식이 없을 때 사용



비모수적 방법

f의 형태를 가정하지 않는 방법

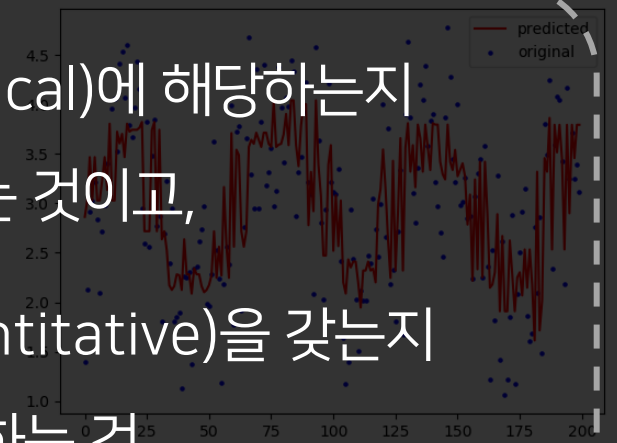
2 모델링

f를 추정하는 방법

f 를 추정하는 방법은 학습 목적에 따라서도 나눌 수 있음

분류는 Y가 **어떤 범주**(categorical)에 해당하는지
예측하는 f를 추정하는 것이고,

회귀는 Y가 **어떤 수치적 값**(quantitative)을 갖는지
예측하는 f를 추정하는 것



① 데이터(관측치 개수)가 많거나,
② 사전지식이 없을 때 사용

비모수적 방법

f의 형태를 가정하지 않는 방법

각각의 방법론에 대한 자세한 내용은
앞으로의 클린업 + 범주팀&회귀팀 클린업 참고!

모델 평가

어떻게 해야 f를 '잘' 추정할 수 있을까?

즉, '어떤 모델이 좋은 모델인가?'라는 평가 기준에 대한 질문과 같음

f를 추정하는 첫번째 목적은 X를 통해 Y를 예측하는 것

→ 모델을 통해 예측한 추정치($\hat{y}$)와 실제값(y)의 차이가 적을수록

예측을 잘 했다고 판단 가능

모델 평가

어떻게 해야 f를 '잘' 추정할 수 있을까?

즉, '어떤 모델이 좋은 모델인가?'라는 평가 기준에 대한 질문과 같음

f를 추정하는 첫번째 목적은 X를 통해 Y를 예측하는 것

→ 모델을 통해 예측한 **추정치($\hat{y}$)와 실제값(y)의 차이가 적을수록**

예측을 잘 했다고 판단 가능

모델 평가

MSE (Mean Squared Error)

$$E[(y - \hat{y})]^2$$

단순히 추정치와 실제값의 차이를 성능으로 정의할 시,

음수 값의 존재로 정확한 성능을 파악할 수 없음

오차의 제곱을 활용한 MSE 값을 사용

MSE를 최소화하는 것 = 모델의 성능을 높이는 것

모델 평가

MSE (Mean Squared Error)

$$E[(y - \hat{y})]^2$$

단순히 추정치와 실제값의 차이를 성능으로 정의할 시,

음수 값의 존재로 정확한 성능을 파악할 수 없음

오차의 제곱을 활용한 MSE 값을 사용

MSE를 최소화하는 것 = 모델의 성능을 높이는 것

모델 평가

MSE (Mean Squared Error)

$$E[(y - \hat{y})]^2$$

단순히 추정치와 실제값의 차이를 성능으로 정의할 시,

음수 값의 존재로 정확한 성능을 파악할 수 없음

MSE를 줄일 실마리를 찾기 위해,

↓

MSE 수식을 분해해보자!

오차의 제곱을 활용한 MSE 값을 사용

MSE를 최소화하는 것 = 모델의 성능을 높이는 것

모델 평가

MSE Decomposition

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

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

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

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

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

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

모델 평가

MSE Decomposition

2단계 1단계

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

Reducible error(축소 가능 오차)

실제 f는 우리가 컨트롤할 수 없지만,

$\hat{f}$는 더 좋은 모델을 찾음으로써 줄일 수 있음

→ 더 좋은 모델을 찾음으로써 줄일 수 있는 error term

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

모델 평가

MSE Decomposition

2단계 1단계

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

Irreducible error(축소 불가능 오차)

$Var(\varepsilon)$는 데이터 자체가 갖고 있는 오차로,
데이터 수집 단계에서 어쩔 수 없이 발생함

→ 우리가 더 이상 줄일 수 없는 error term

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

모델 평가

MSE Decomposition

2단계 1단계

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

∴ 현실적으로 Reducible error를 줄이기 위해 노력

더 분해해보자!

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

모델 평가

MSE Decomposition

2단계 1단계

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

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

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

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

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

$$\vdots \qquad \vdots$$

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

모델 평가

MSE Decomposition

2단계 1단계

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

Bias(편향)

= 추정치의 기댓값과 실제값의 차이

추정한 모델이 실제 모델을 얼마나 잘 설명하는지 의미

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

$$\vdots \qquad\qquad \vdots$$

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

모델 평가

MSE Decomposition

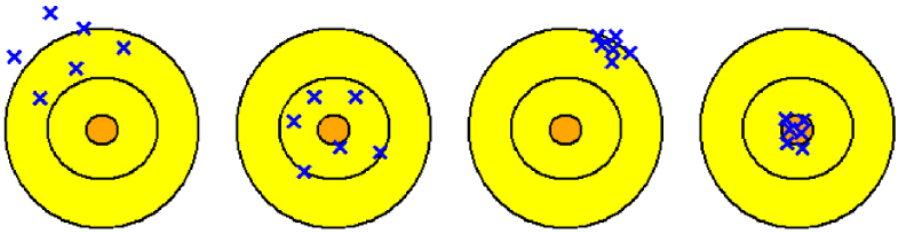
2단계 1단계

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

Bias(편향)

편향 大 : 모델링 반복 수행 아무리 해도 정답을 맞출 가능성이 낮음

편향 小 : 모델링 반복 수행 시 평균적으로 잘 맞춰내기 때문에 작을수록 좋음



Bias	High	Low	High	Low
Variance	High	High	Low	Low

모델 평가

MSE Decomposition

2단계 1단계

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

Variance(분산)

= 추정치의 개별값과 추정치의 기댓값의 차이를 제곱합한 것

추정한 모델에 다른 데이터셋 적합 시, 추정치가 얼마나 달라지는지 의미

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

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

모델 평가

MSE Decomposition

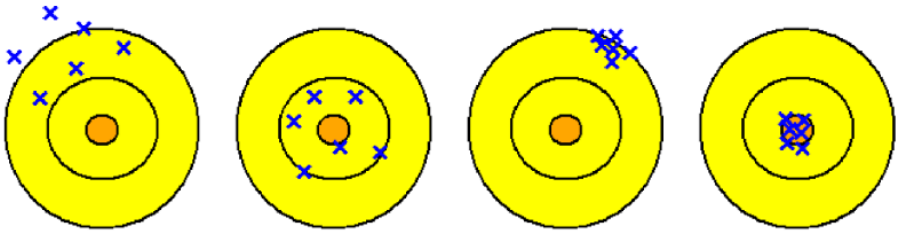
2단계 1단계

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

Variance(분산)

분산 大 : 노이즈가 바뀜에 따라 개별 추정값들이 많이 바뀜

분산 小 : 노이즈가 바뀌어도 개별 추정값들이 잘 바뀌지 않기 때문에 작을수록 좋음



Bias	High	Low	High	Low
Variance	High	High	Low	Low

모델 평가

MSE Decomposition

MSE 분해 결과

모델의 Bias와 Variance를 줄이는 방향으로

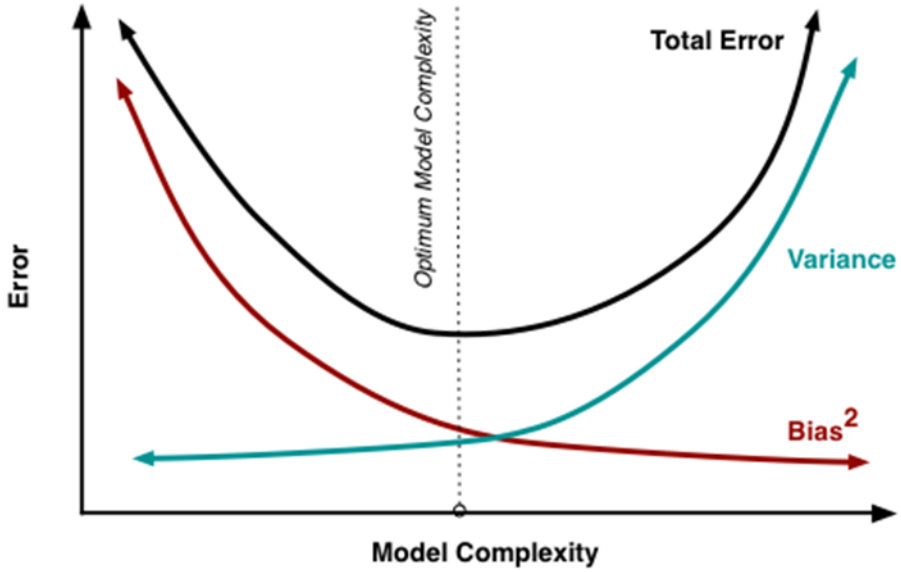
학습해야 한다는 것을 알게 되었지만

⋮

현실적으로는 2가지를 동시에 줄이기 어려운 딜레마 상황이 발생함

Variance-Bias Trade off

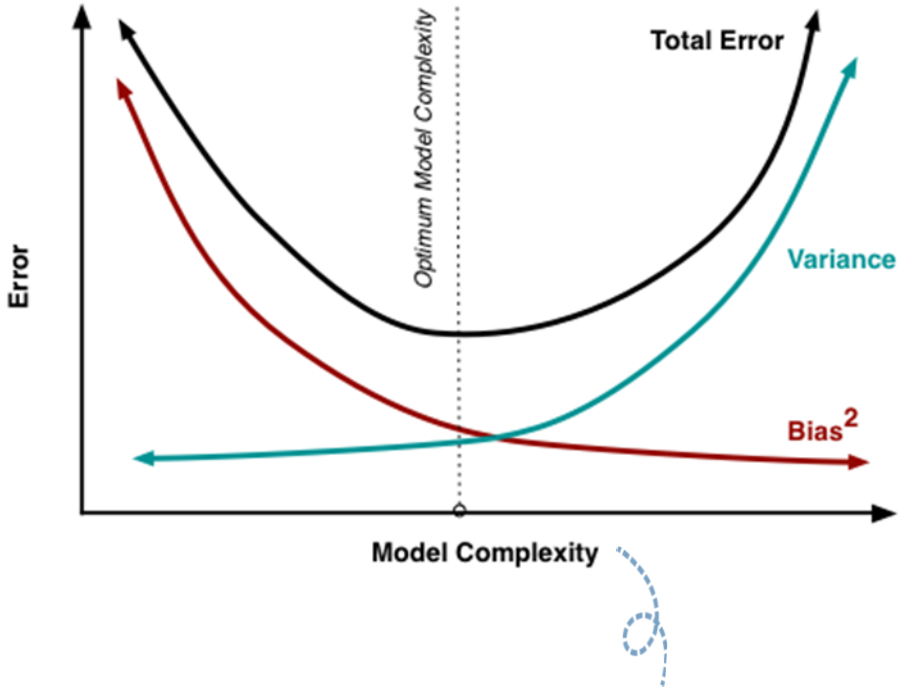
한쪽을 줄이면 다른 한쪽은 커지는 Trade-off 관계



모델의 Bias와 Variance가 서로 **반대 방향으로** 움직임

Variance-Bias Trade off

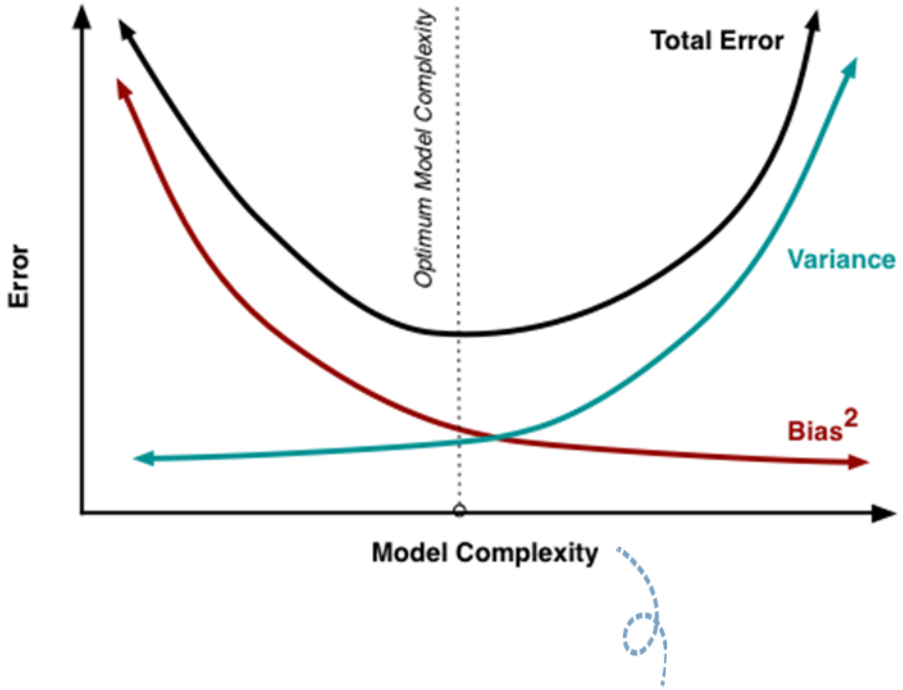
한쪽을 줄이면 다른 한쪽은 커지는 Trade-off 관계



모델이 복잡할수록 Bias가 감소하지만 Variance는 증가하고,
이에 따라 전체 오차도 일정 부분을 지나면 다시 증가하는 추세를 보임

Variance-Bias Trade off

한쪽을 줄이면 다른 한쪽은 커지는 Trade-off 관계



즉, 우리가 줄일 수 있는 오차인 Bias와 Variance는

모델 복잡도의 영향을 많이 받음!

Variance-Bias Trade off

단순한 모델

예측값과 실제값의 차이가 큼 = Bias 大

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

지나치게 단순하면 좋은 모델이 될 수 없음 = Underfitting

Variance-Bias Trade off

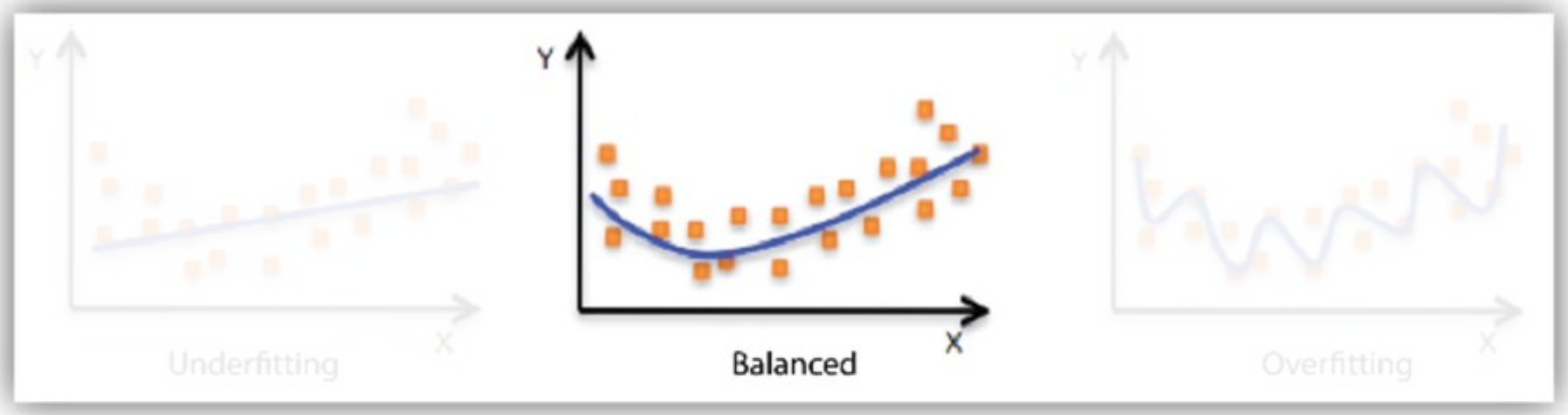
복잡한 모델

예측값과 실제값의 차이가 작음 = Bias 小

데이터 포인트가 하나만 추가되어도 그에 맞게 매우 달라짐 = Variance 大

지나치게 복잡하면 좋은 모델이 될 수 없음 = Overfitting

Variance-Bias Trade off

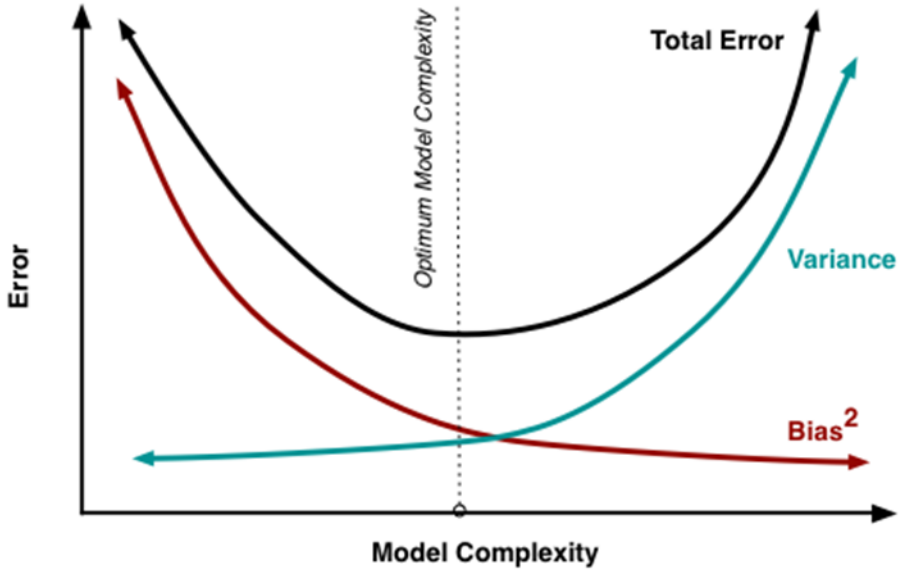


적절한 모델

따라서 너무 단순하지도, 너무 복잡하지도 않은 모델을 찾는 것이 핵심

즉, Bias와 Variance 모두 적절히 작아 그 합이 최소가 되는 지점을 찾아야 함

Variance-Bias Trade off



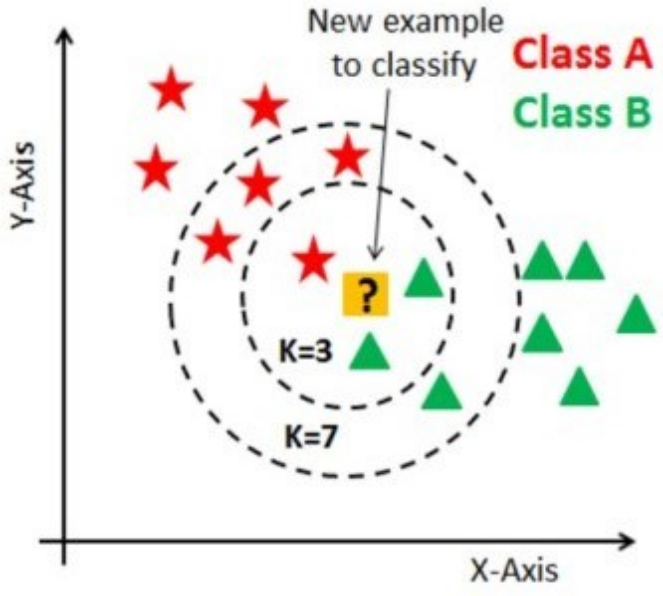
점선 = optimal point

즉, MSE가 최소가 되는 point

KNN(K-Nearest Neighbor)

KNN-Classifier

대표적인 **비모수적**인 모델로 **K개의 가까운** 이웃데이터들 중 다수결로 예측



가까움의 정도는

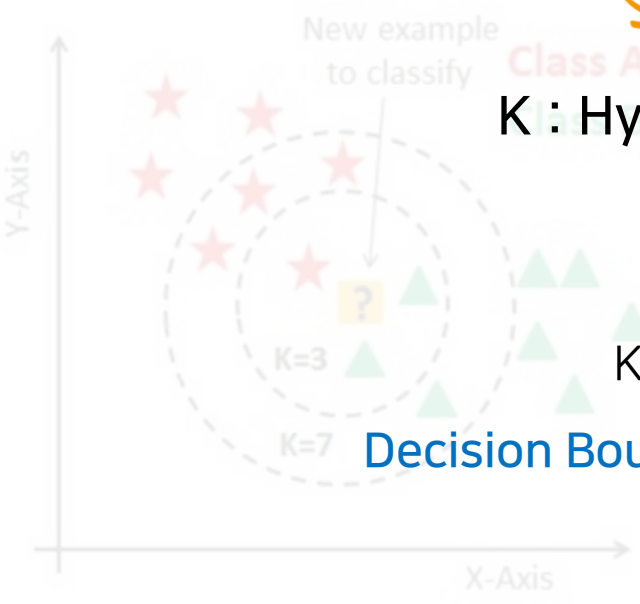
유클리드 거리를 측정하여 판단!

2 모델링

KNN(K-Nearest Neighbor)

KNN-Classifier

대표적인 **비모수적**인 모델로 K개의 **가까운** 이웃데이터들 중 다수결로 예측



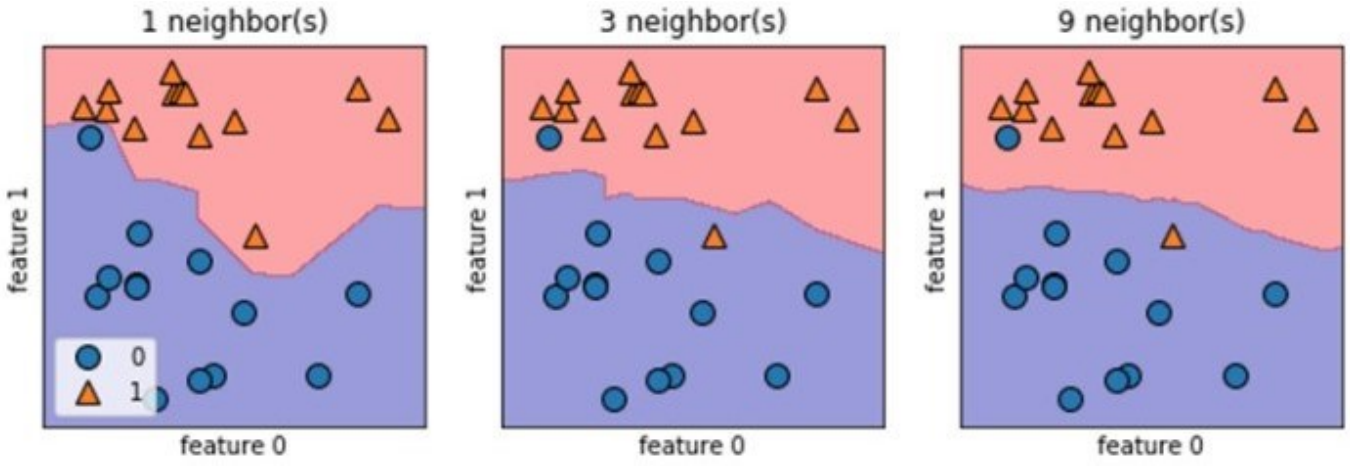
K : Hyperparameter

K의 값에 따라

Decision Boundary(분류 경계면) **변화**

2 모델링

KNN(K-Nearest Neighbor)



K = 1

자기 자신만을 이웃으로 참조하기 때문에

Decision Boundary가 **매우 복잡함**

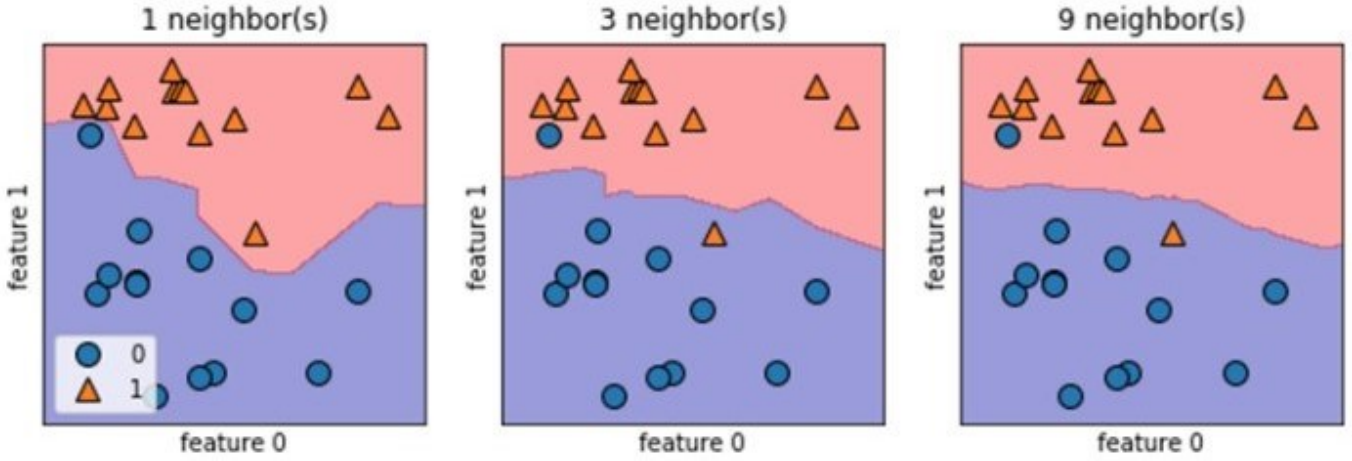
K = 9

더 많은 개수의 이웃을 참조하기 때문에

Decision Boundary가 **간단함**

2 모델링

KNN(K-Nearest Neighbor)



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

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

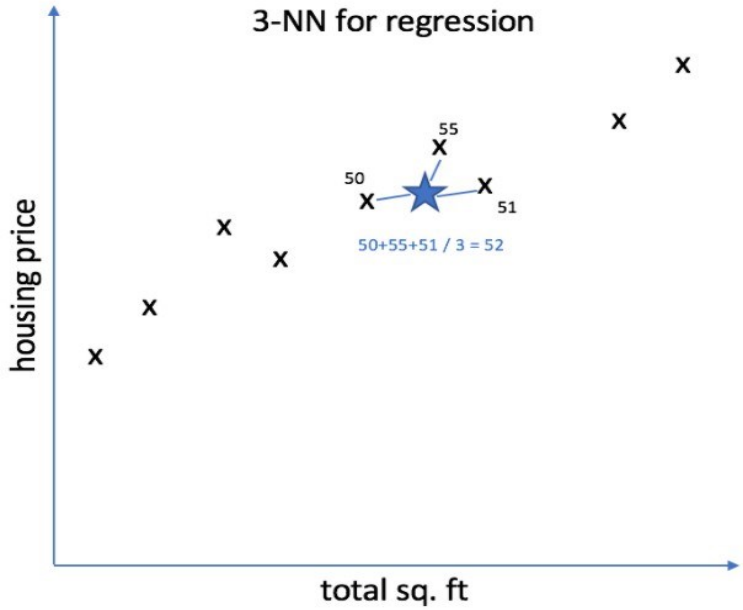
KNN(K-Nearest Neighbor)

KNN-Regressor

$$\hat{Y}_0 = \frac{1}{K} \sum_{x_i \in N_K(x_0)} Y_i$$

x_i, y_i : *training data*

N_k x_0 : *neighborhood of* x_o



비슷한 X값을 가진 데이터끼리는 **비슷한 Y값**을 가질 것이라는 아이디어로

K개 이웃들의 Y값 평균을 활용하여 예측하는 **비모수적 방법**

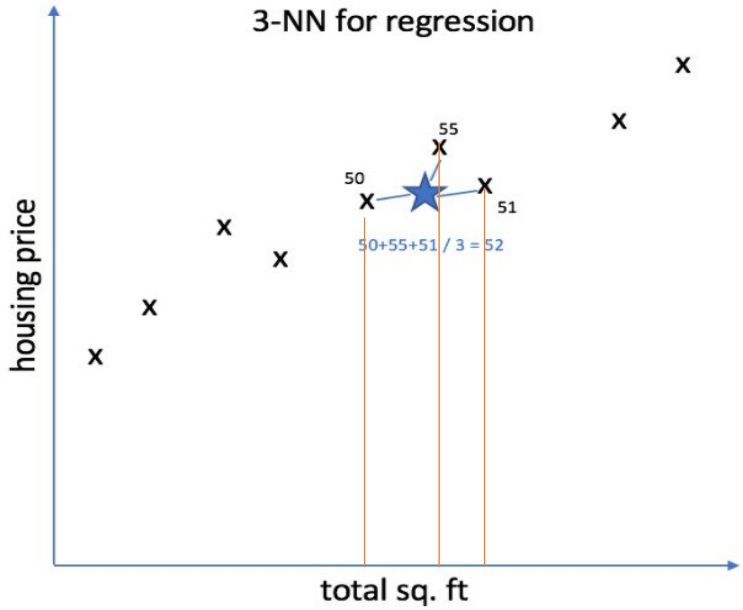
2 모델링

KNN(K-Nearest Neighbor)

$$\hat{Y}_0 = \frac{1}{K} \sum_{x_i \in N_K(x_0)} Y_i$$

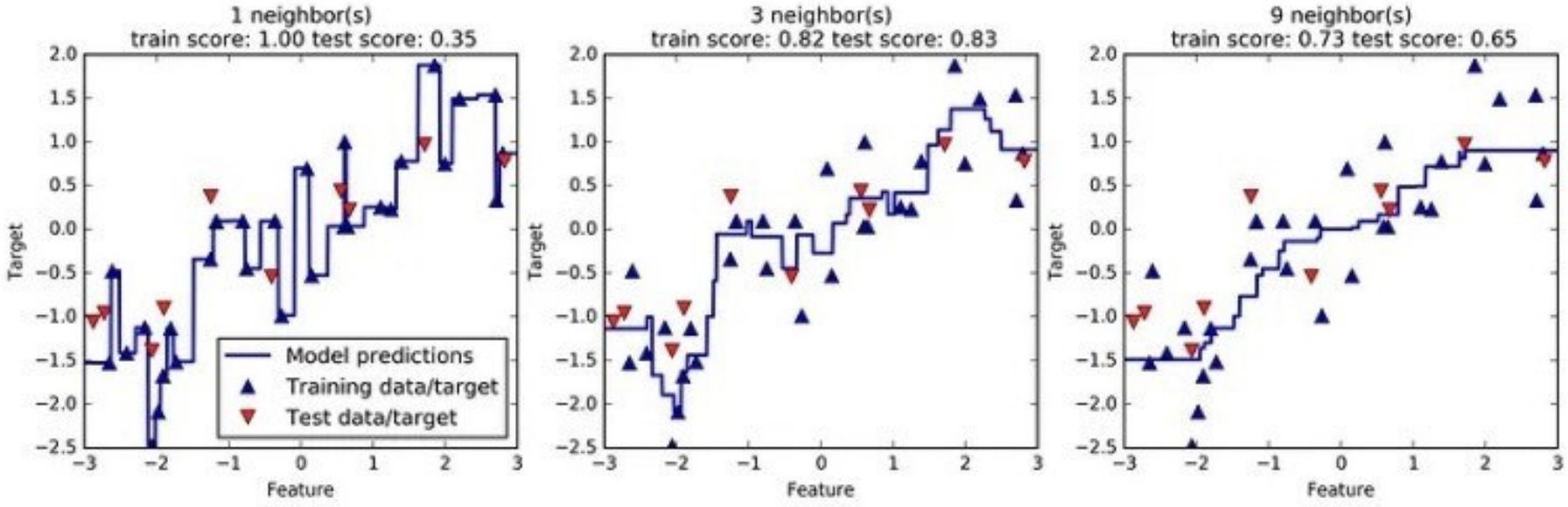
x_i, y_i : *training data*

N_k x_0 : *neighborhood of* x_o



X값을 기준으로 가까운 점들을 찾고, 그 점들의 Y값을 평균 내어 예측값으로 계산

KNN(K-Nearest Neighbor)



K값에 따라 모델 복잡도가 결정되고, Bias와 Variance가 영향을 받음

K값이 작을수록 복잡한 Decision Boundary 형성

3

모델링 전략

3 모델링 전략

Hyperparameter Optimization

파라미터 (Parameter)

데이터로부터 학습되어 결정됨

모델 내부에서 **결정**되는 변수

하이퍼파라미터 (Hyperparameter)

모델 학습 시에 사용자가

직접 지정해야 하는 변수

3 모델링 전략

Hyperparameter Optimization

어떤 방식으로 지정해줄 수 있을까?

Hyperparameter Optimization을 통해 찾을 수 있음!

하이퍼파라미터 (Hyperparameter)

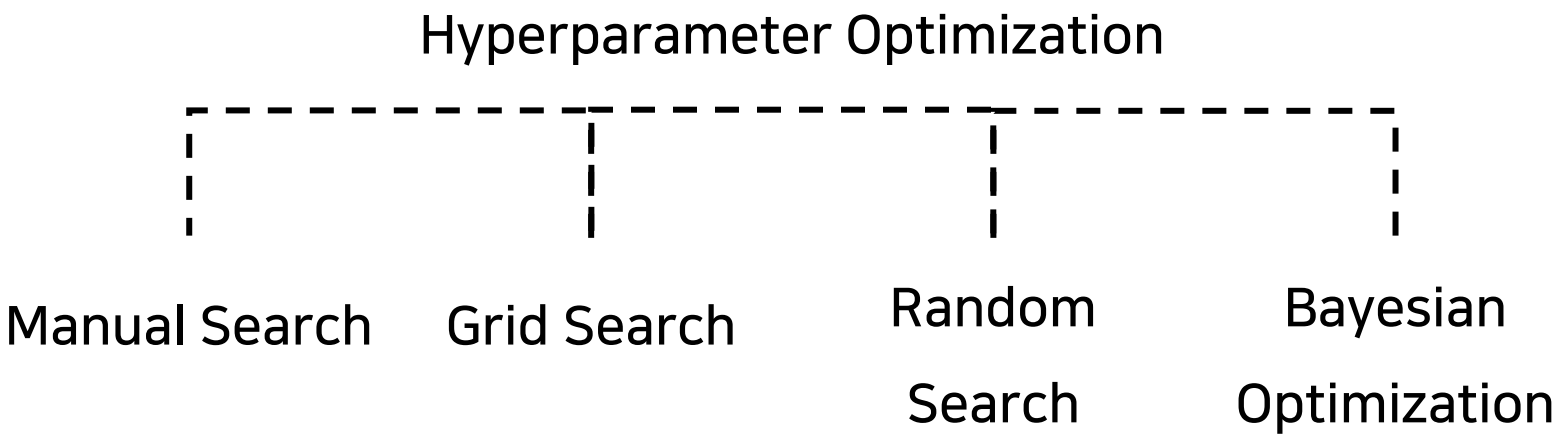
모델 학습 시에 사용자가

직접 지정해야 하는 변수

Hyperparameter Optimization

Hyperparameter Optimization

하이퍼파라미터 조합 중 **가장 성능을 좋게 만드는 조합**을 찾아나가는 과정



Manual Search

Manual Search

사용자가 **수동으로 성능을 비교해가며 튜닝**하는 방법

직관 또는 대중적으로 알려진 노하우 등에 의존

Manual Search

Manual Search

사용자가 **수동으로 성능을 비교해가며 튜닝**하는 방법

직관 또는 대중적으로 알려진 노하우 등에 의존

사용자의 직관과 경험에 의존하기 때문에

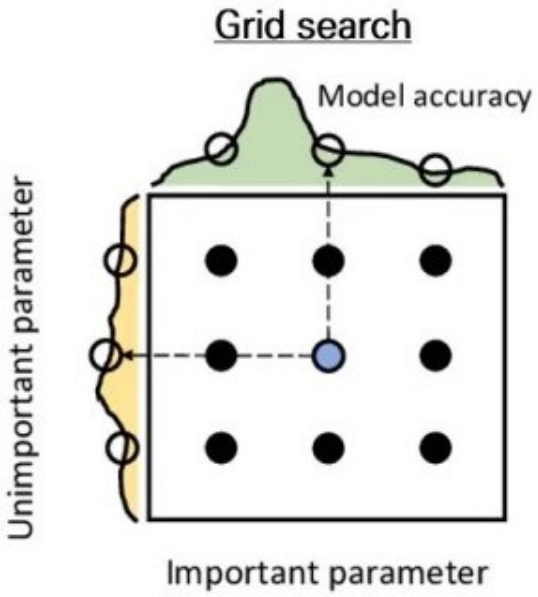
시간이 오래 걸리고 효율도 보장하기 어려움

Grid Search

Grid Search

탐색 대상의 구간 내의 **모든 후보 하이퍼파라미터 값**들을

격자 방식으로 비교해서 탐색하는 방법



Grid Search

Grid Search

탐색 대상의 구간 내의 **모든 후보 하이퍼파라미터 값**들을

격자 방식으로 비교해서 탐색하는 방법

Manual Search보다는 효율적이지만,

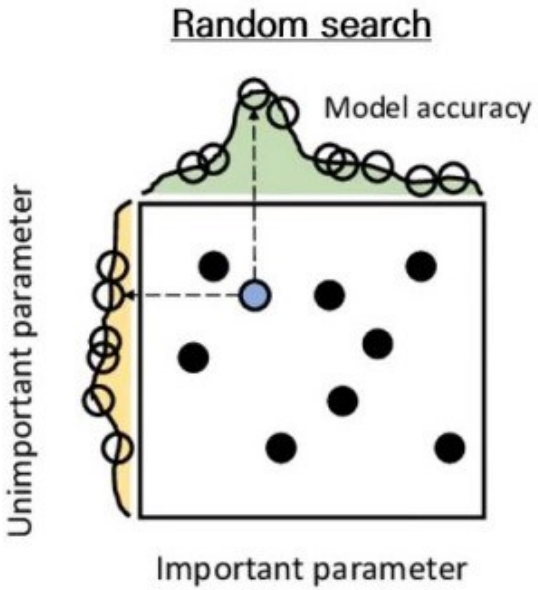
탐색할 파라미터가 많아질수록 탐색 비용이 **기하급수적** (n^k) 으로 늘어남

Random Search

Random Search

구간 내 후보 하이퍼파라미터 값들을 **랜덤 샘플링**하는 방법

범위 내의 **임의의 값으로 조합**을 시도해볼 수 있음



Random Search

Random Search

구간 내 후보 하이퍼파라미터 값들을 **랜덤 샘플링**하는 방법

범위 내의 **임의의 값으로 조합**을 시도해볼 수 있음

Grid Search에 비해 확률적으로

더 빠르게 최적해를 찾을 수 있음

3 모델링 전략

Random Search

Grid Search와 Random Search의 한계

Random Search

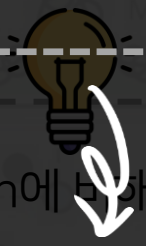
구간 내 후보 하이퍼파라미터 값들을 랜덤 샘플링 하는 방법

범위 내의 임의의 값을 탐색해볼 수 있음

두 방법 모두

매 탐색이 독립적으로 진행되기 때문에

이전에 탐색한 정보를 반영하지 못함



Grid Search에 비해 확률적으로

중요한 파라미터를 더 빠르게 살펴볼 수 있음

사전 지식을 반영하는 **Bayesian optimization**

Bayesian Optimization

Bayesian Optimization

임의의 목적함수 $f(x)$ 를 **최대로 하는 해**를 찾는 방법으로

전체적인 탐색 과정을 체계적으로 수행할 수 있음

베이지안 확률에 기반을 둔 최적화 방법으로,

사전 정보를 바탕으로 **사후 모델**을 **개선**해 나감

3 모델링 전략

Bayesian Optimization

Bayesian Optimization의 핵심 요소

어떻게 **사전 정보**를
얻을 수 있을까?

Surrogate Model
(대리 모델)

획득한 사전 정보를 어떻게
다음 탐색에 활용할까?

Acquisition Function
(획득 함수)

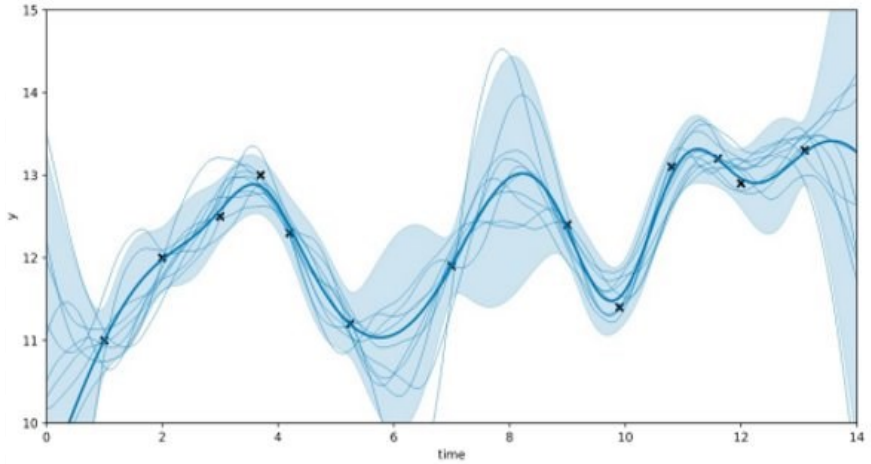
3 모델링 전략

Bayesian Optimization

Bayesian Optimization의 핵심 요소

어떻게 **사전 정보**를
얻을 수 있을까?

Surrogate Model
(대리 모델)

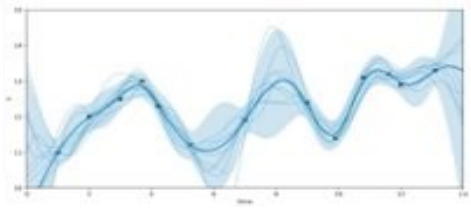


미지의 목적 함수의 형태에 대해
확률적인 추정을 **수행**하는 모델

3 모델링 전략

Bayesian Optimization

Learning rate	# of iterations	# of hidden layers	# of hidden nodes	...	...	Mini batch size	Generalization performance
0.001	1000	3	X	...	...	128	Y
...	...	...	...	...	...	...	200



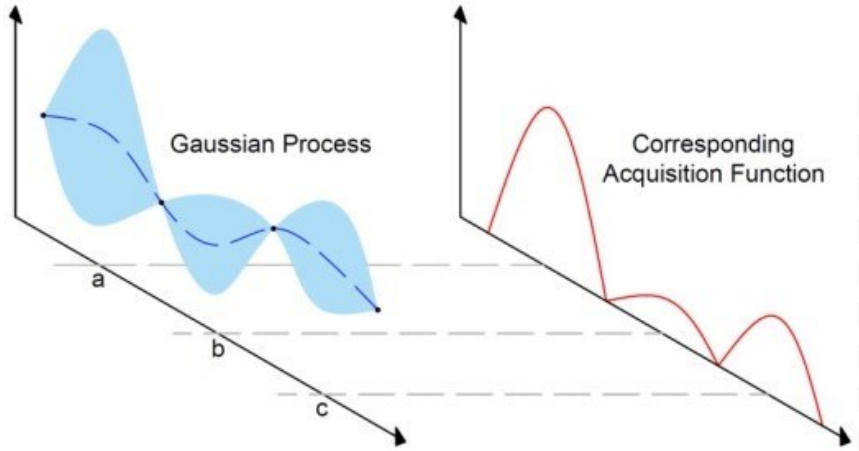
하이퍼파라미터 집합(X)과 성능(Y)의 관계를 모델링함

일반적으로 Gaussian Process를 사용

3 모델링 전략

Bayesian Optimization

Bayesian Optimization의 핵심 요소



목적 함수에 대한

현재까지의 확률적 추정 결과를 바탕으로

다음 번 탐색 후보를 추천하는 함수

획득한 사전 정보를 어떻게

다음 탐색에 활용할까?

Acquisition Function

(획득 함수)

3 모델링 전략

Bayesian Optimization

Acquisition Function은 **두 가지 전략**을 바탕으로 하이퍼파라미터를 추천

Exploitation (착취)

최적값일 가능성이 높은 값을 추천

Exploration (탐험)

탐색이 **불확실한 부분**에 있는 **값**을 추천

대표적인 방법으로 PI(Probability of Improvement)와 EI(Expected Improvement)가 있음

3 모델링 전략

Bayesian Optimization

Acquisition Function은 **두 가지 전략**을 바탕으로 하이퍼파라미터를 추천

Exploitation

최적값일 가능성이 높은 값을 추천

Exploration

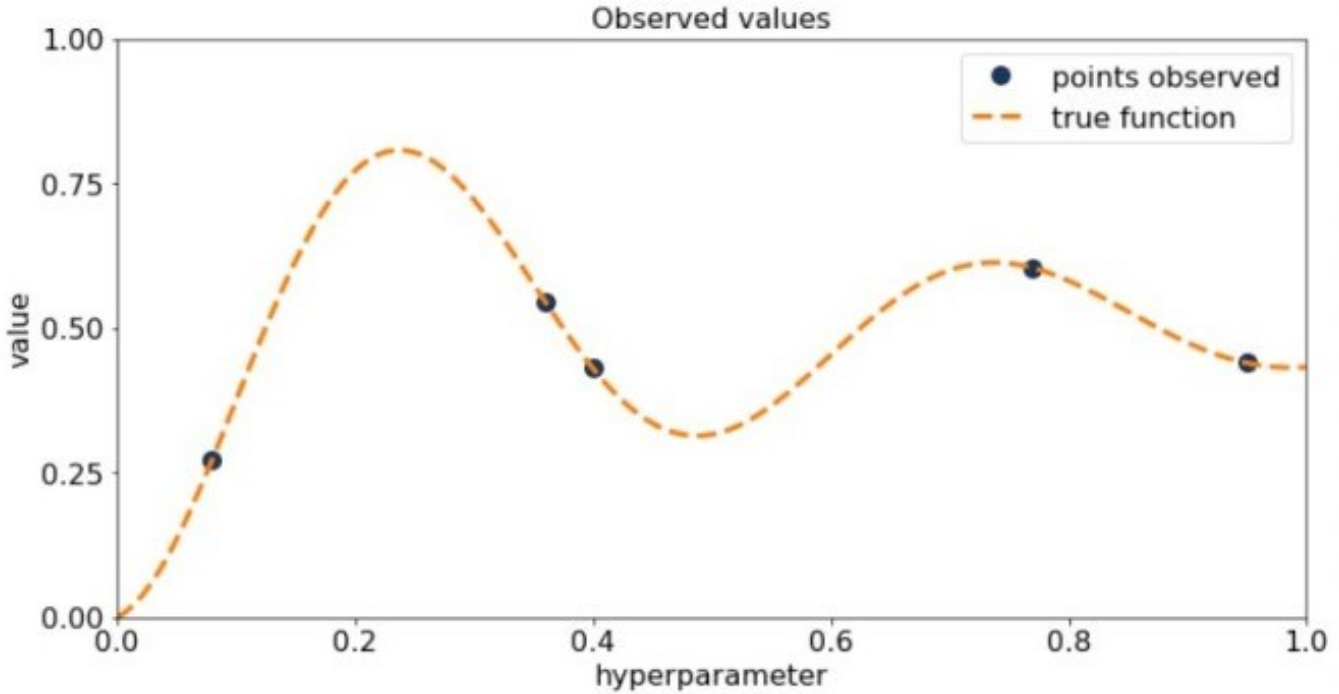
탐색이 불확실한 부분에 있는 값을 추천

이를 바탕으로

Bayesian Optimization의 작동 과정을 알아보자!

3 모델링 전략

Bayesian Optimization

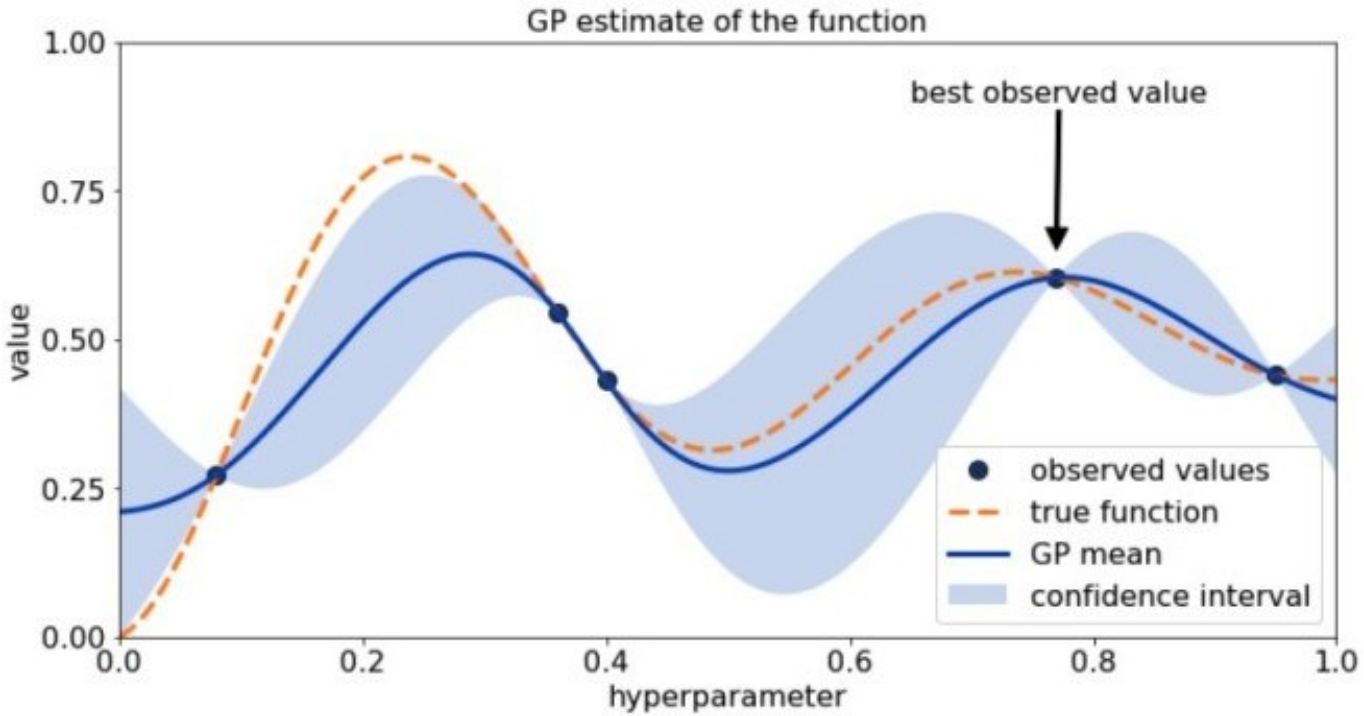


① 일단 **랜덤하게 하이퍼파라미터들을 샘플링**하고 성능 결과 관측

여기서 주황색 점선은 우리가 찾아야 하는 목적함수

3 모델링 전략

Bayesian Optimization

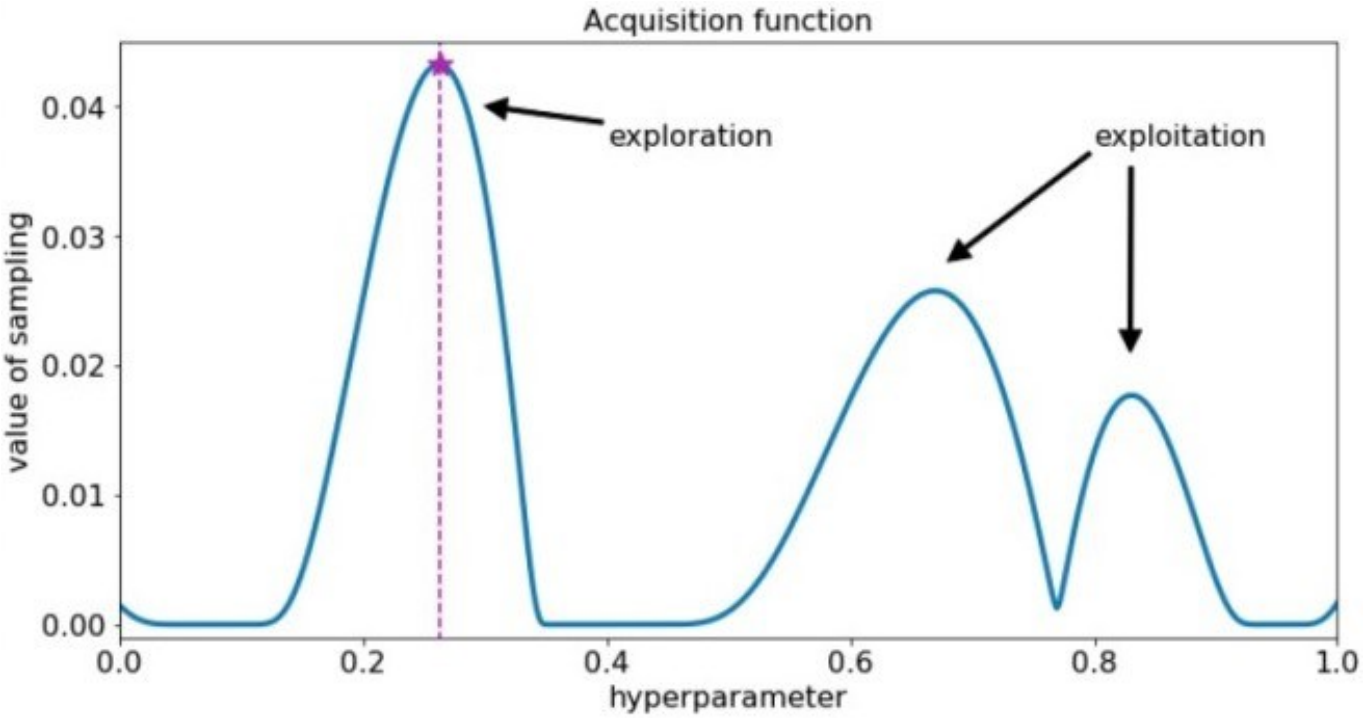


② 관측된 값을 기반으로 Surrogate Model이 **목적함수 f를 추정**

파란색 영역은 신뢰 구간으로, 불확실성 의미

3 모델링 전략

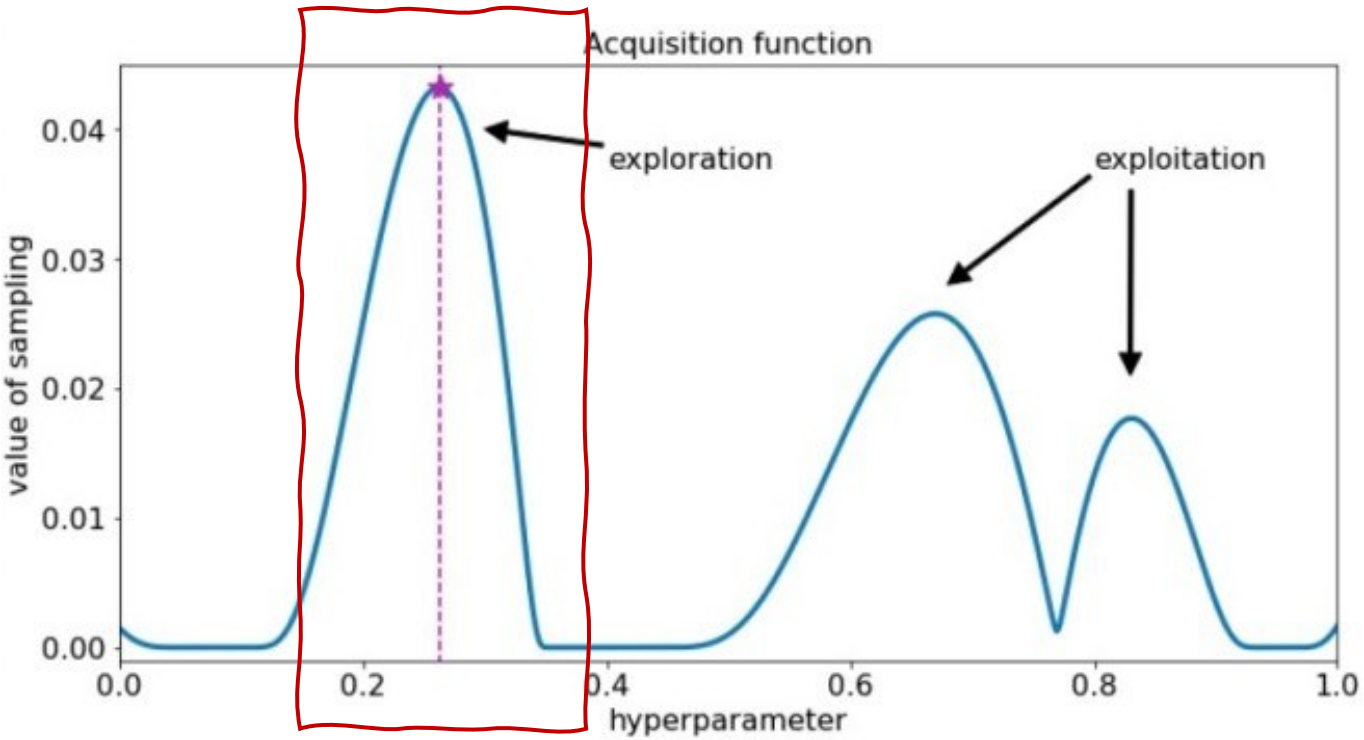
Bayesian Optimization



③ 추정된 Surrogate Model 기반으로 Acquisition Function 계산

3 모델링 전략

Bayesian Optimization

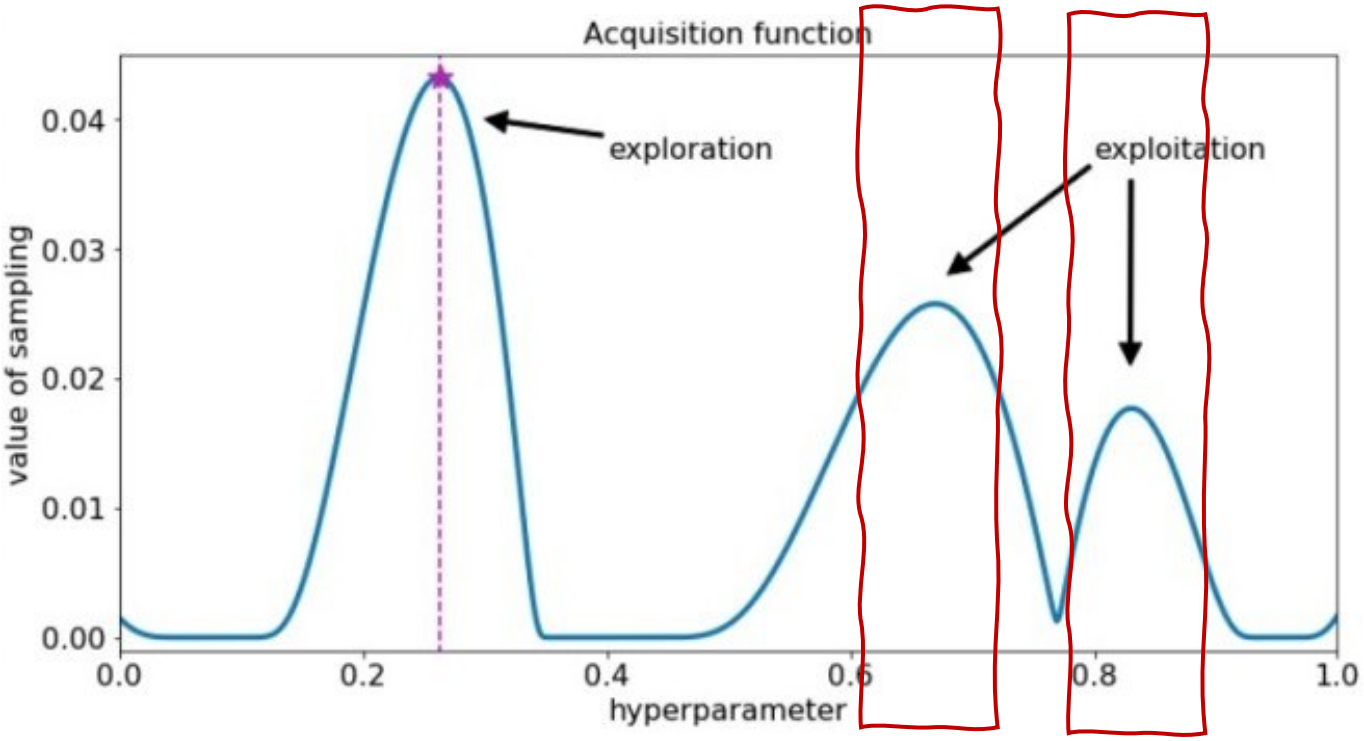


④ Acquisition Function에 의해 하이퍼파라미터가 추천됨

Surrogate **함숫값이 큰 지점**이 다음 입력 값으로 추천(Exploitation)

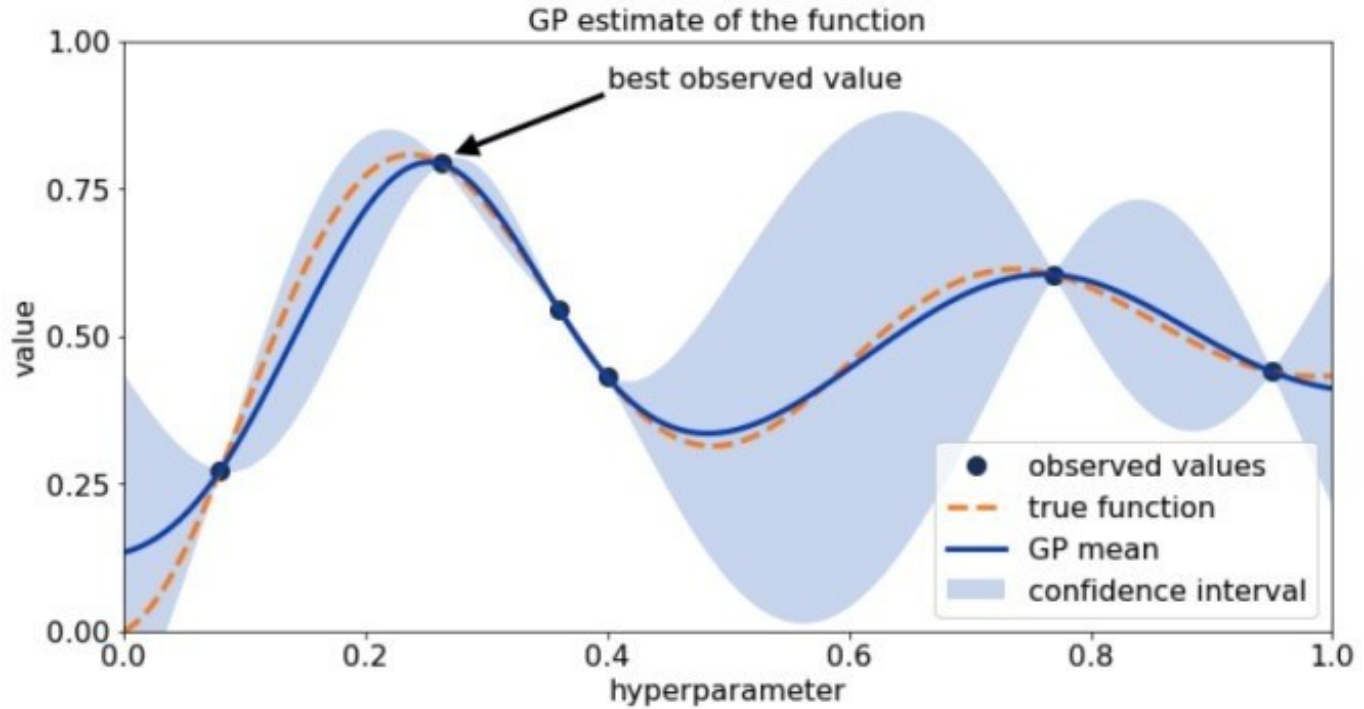
3 모델링 전략

Bayesian Optimization



불확실성이 높은 지점들도 **높은 추천 점수**를 가짐(Exploration)

Bayesian Optimization

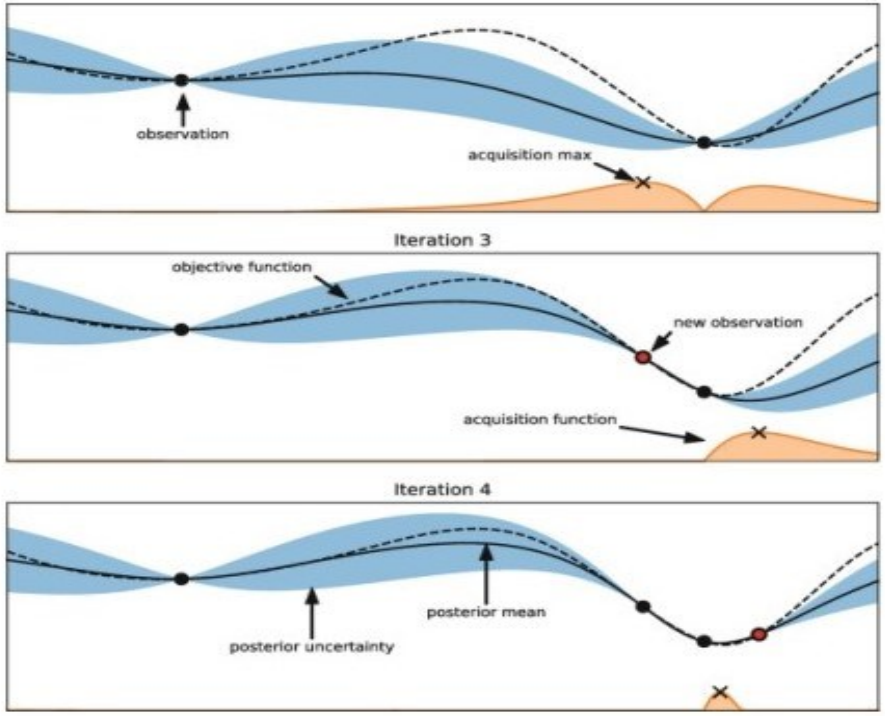


⑤ 추천 받은 하이퍼파라미터로 성능 측정 후,

그에 맞게 Surrogate Model 갱신

3 모델링 전략

Bayesian Optimization



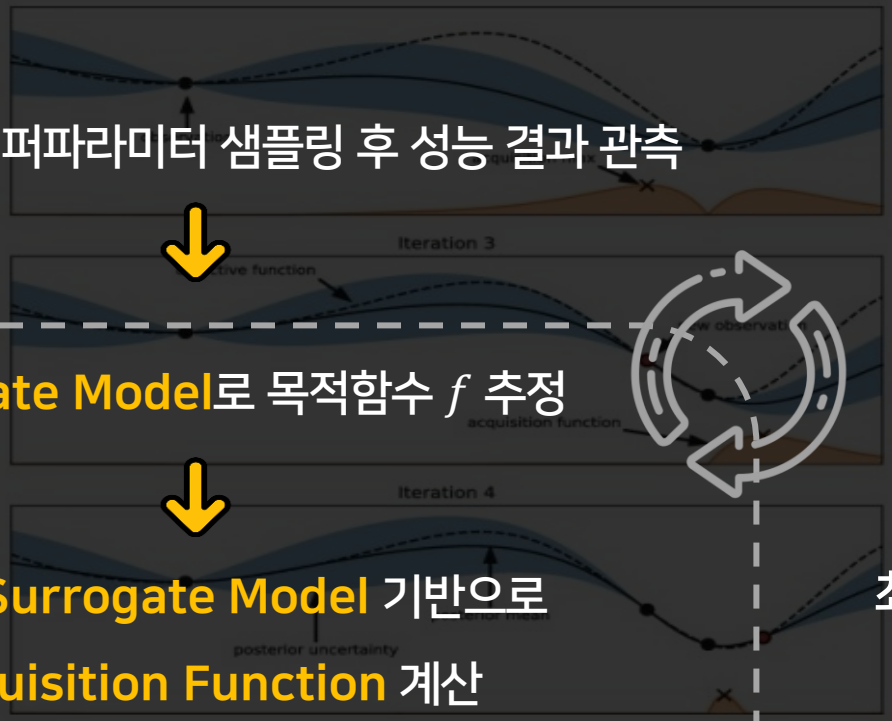
앞선 ② ~ ⑤ 과정을 반복하면서 실제 목적함수에 근사 가능

→ 최적의 하이퍼파라미터 탐색!

3 모델링 전략

Bayesian Optimization

Bayesian Optimization 작동 과정 정리



랜덤하게 하이퍼파라미터 샘플링 후 성능 결과 관측

↓

Surrogate Model로 목적함수 f 추정

↓

추정된 **Surrogate Model** 기반으로

Acquisition Function 계산

↓

추천 받은 하이퍼파라미터로 수행한 성능 측정,

Surrogate Model 갱신

반복 수행하면서

목적함수 f에 근사

⋮

최적의 파라미터를

찾게 됨!

목적함수에 근사 가능

파라미터 탐색

3 모델링 전략

Bayesian Optimization

Bayesian Optimization은 **사전 정보**를 활용하여 더 **효율적으로** **최적화 가능**

그러나, 가우시안 프로세스를 Surrogate Model로 쓸 경우,

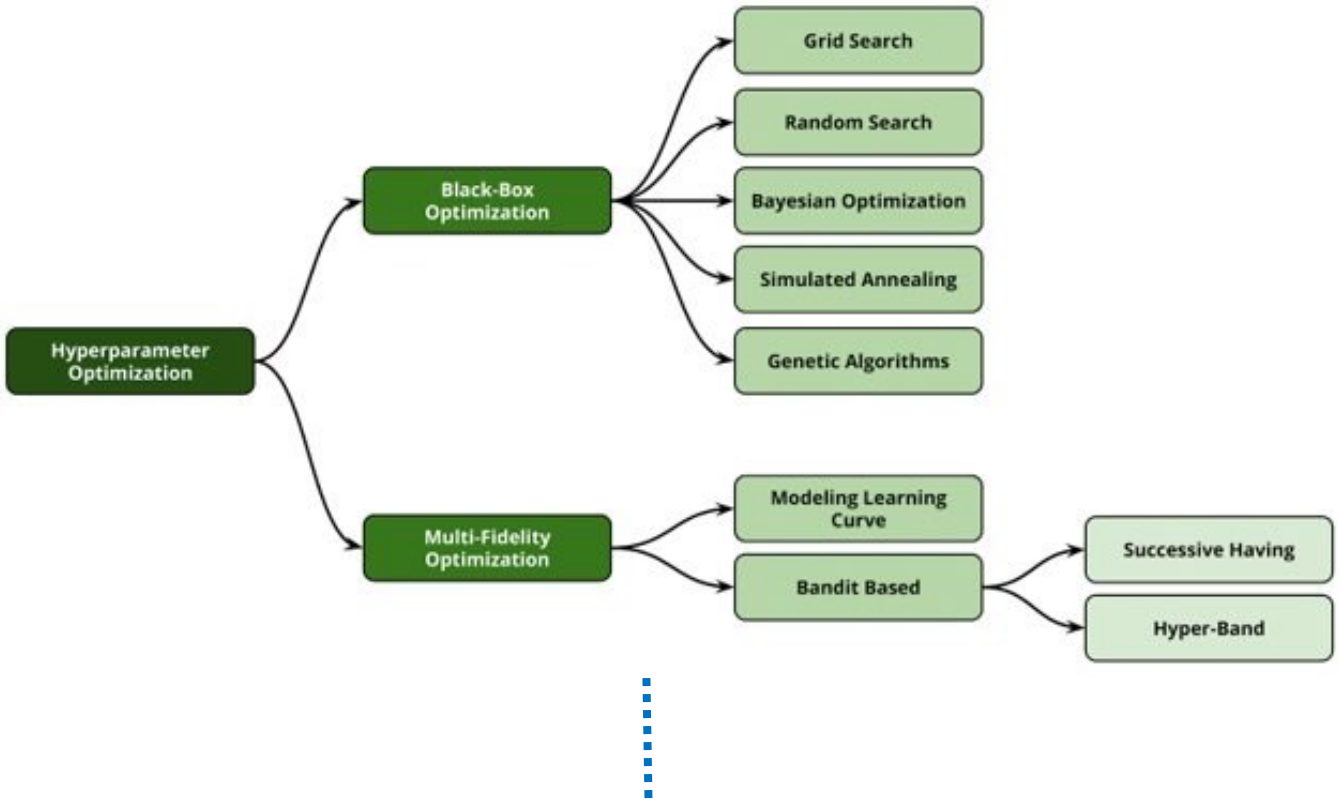
시간복잡도가 높고, 연속형 변수에만 사용할 수 있다는 단점 존재

(자세한 내용은 1학기 알파팀 자료 참고)

이런 단점을 개선하기 위해 **Tree Parzen Estimator(TPE)** 등

다른 Surrogate Model을 사용할 수 있음

Bayesian Optimization



이외에도 다양한 최적화 방법들이 존재하며, 절대적으로 우월한 방법은 없음

모델과 데이터의 특성에 맞게 적절한 방식을 사용해야 함

과적합 문제해결 | 지도학습 Remind

Train data로 f를 추정하여 모델 학습

$\hat{y}$는 오차항 ε을 포함한 y와의 비교를 통해 추정됨

노이즈에 대한 학습이 불가피함

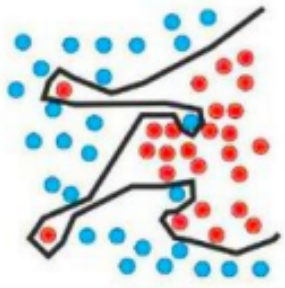
노이즈에 대해 과도하게 학습하게 된다면?

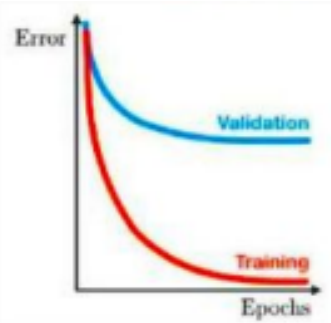
과적합 문제해결 | Overfitting

과대적합(Overfitting)

모델이 Train data는 잘 예측하나,

Test data 예측에는 좋은 성능을 보이지 않는 문제





Train MSE는 작게 나와도 Test MSE가 높을 수 있음

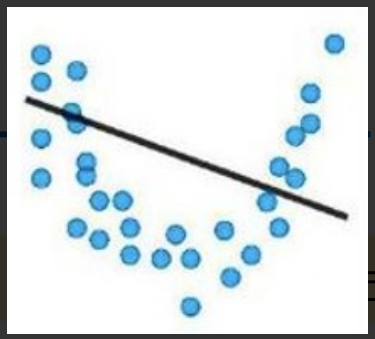
3 모델링 전략

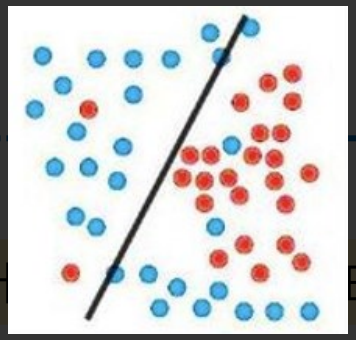
과대적합 (Overfitting)

노이즈를 적게 반영되는

과대적합? **간단한 모델**을 만들면 문제가 해결될까?

모델이 Train data는 잘 예측하나,
모델이 너무 간단하면 오히려 데이터의
Test data 예측에는 좋은 성능을 보이지 않는 문제
패턴을 **충분히 학습하지 못하는 경우**가 발생





과소적합(underfitting)의 문제가 발생

3 모델링 전략

과적합 해결

자주 문제가 되는 **과대적합**을 방지하기 위해
다양한 방법들이 등장함

교차 검증

Train set을 세부적으로
나누어 모델 학습

차원 축소

변수 선택 or 변수 추출

교차 검증 (Cross Validation)

교차 검증

Train data를 다시 Train data와 Validation data로 나누어
모델의 성능을 미리 검증하는 방법

Test data에만 과대적합 되는 것을 방지하며
모델의 성능을 정확하게 판단하여 모델링의 방향을 잡는데 도움을 줌

3 모델링 전략

교차 검증 (Cross Validation)

교차 검증?

Test data vs **Validation data**

Train data를 다시 Train data + Validation data로 나눠서

모델의 성능을 미리 검증

Test data	Validation data
검증이 끝난 모델에 대해	**하이퍼파라미터를 조정**하며
최종 성능만을 평가	모델 성능을 확인
즉, 모델링 과정에는	즉, 모델링 과정에
전혀 관여하지 않음	간접적으로 관여함

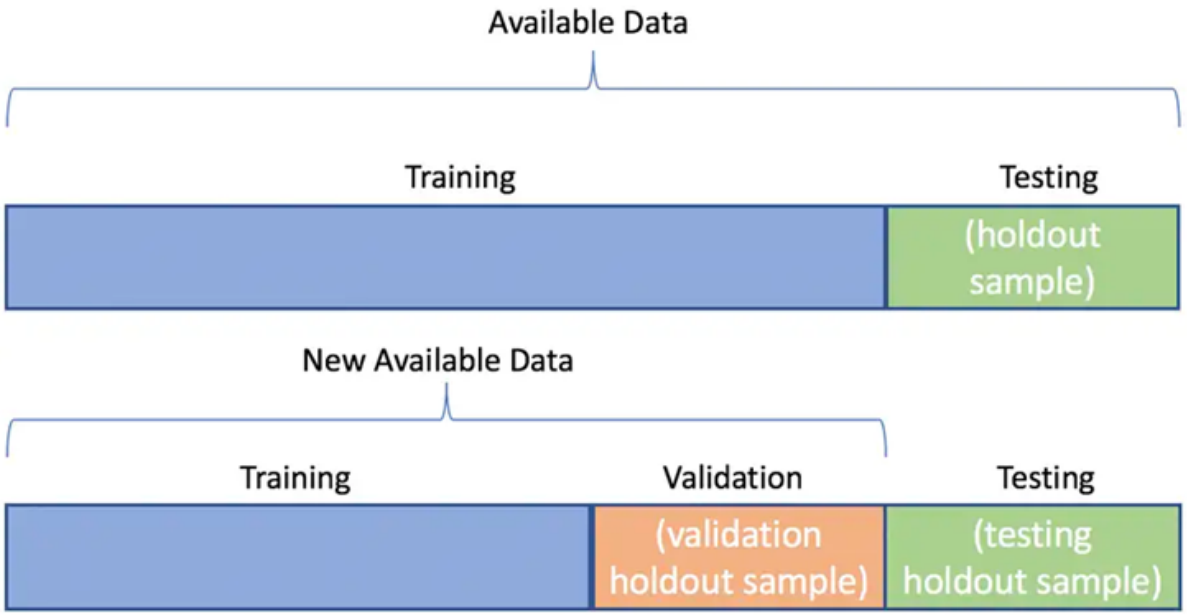
교차 검증의 효과

Validation data가 생김으로서 Test data에 대한

data로 하이퍼파라미터를 조정

교차 검증 (Cross Validation)

Hold-out Validation



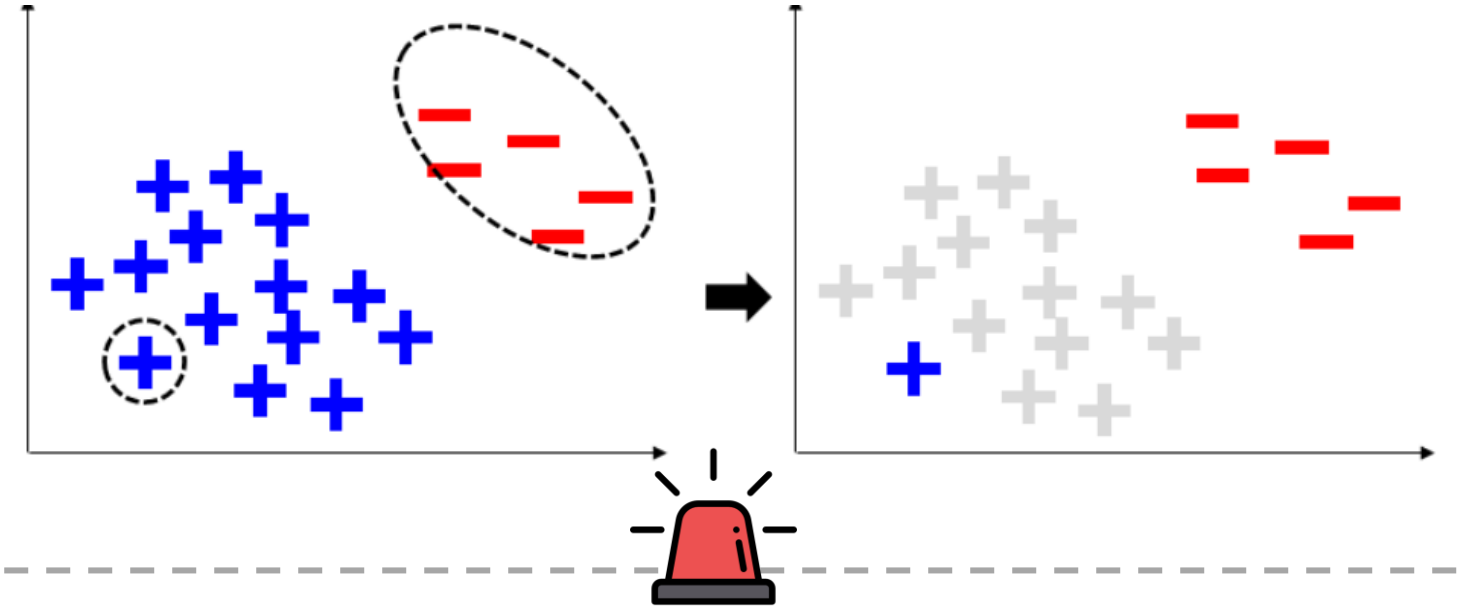
가장 단순한 방법으로, **Train data를 한번 더 나눠서**

Validation data 생성 (주로 7:3 혹은 8:2)

3 모델링 전략

교차 검증 (Cross Validation)

Hold-out Validation



① Validation data가 전체 데이터의 **경향성을 충분히 포함하지 않는다면** 모델이 **왜곡**

② **학습에 참여할 데이터가 감소됨** (작은 데이터셋일 경우 더욱 치명적)

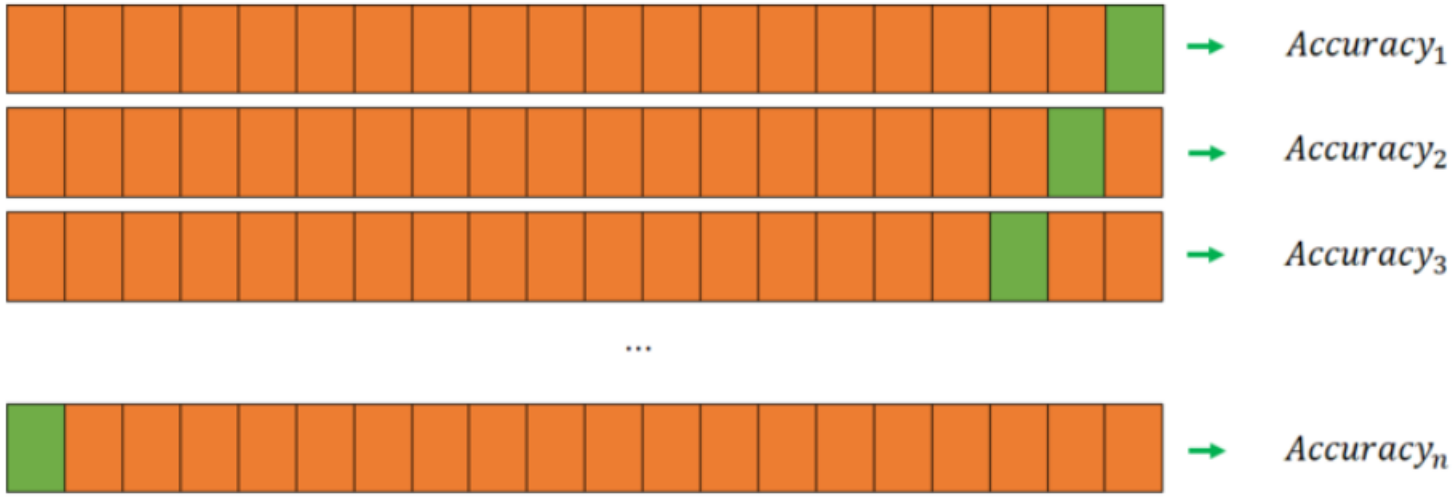
교차 검증 (Cross Validation)

이러한 단점을 보완하기 위해, Validation data를

고정시키지 않고 **데이터의 모든 부분을 교차**하며 사용할 수 있음

교차 검증 (Cross Validation)

LOOCV (Leave-One-Out Cross Validation)



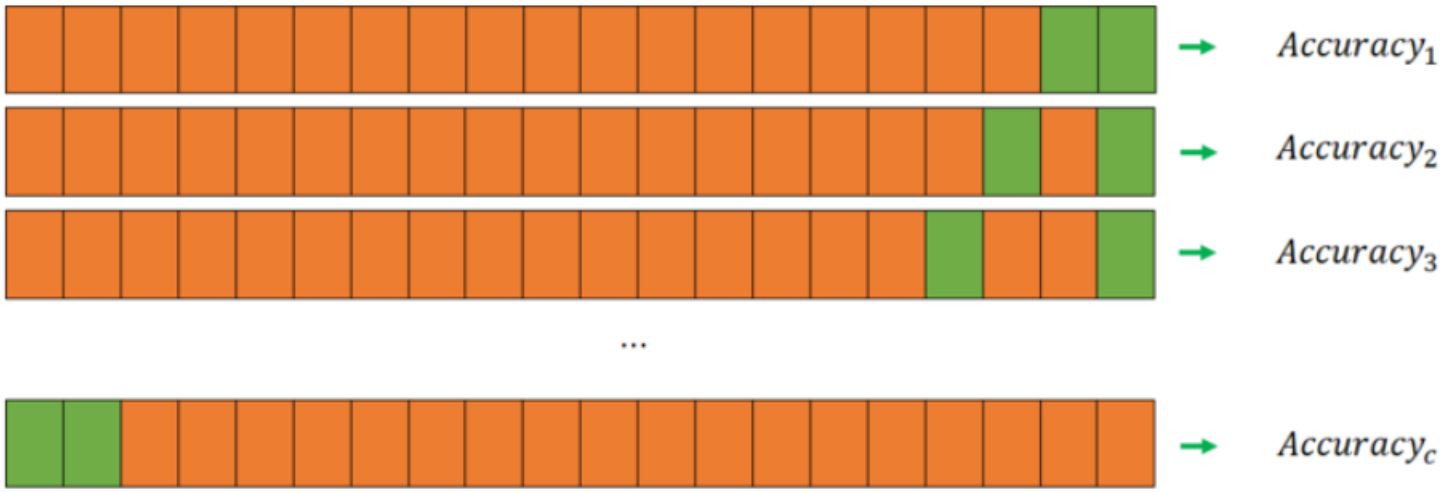
전체 데이터 n개 중 1개만을 validation data로,

나머지 (n-1)개를 train data로 사용

3 모델링 전략

교차 검증 (Cross Validation)

LpOCV (Leave-p-Out Cross Validation)



전체 데이터 **n개 중 p개**만을 validation data로,

나머지 (n-p)개를 train data로 사용

교차 검증 (Cross Validation)

LOOCV와 LpOCV의 단점 : **과도한 연산량**

LOOCV

n개의 validation dataset

↓

n번의 교차 검증

LpOCV

$\binom{n}{p}$개의 validation dataset

↓

$\binom{n}{p}$ 번의 교차 검증

K-Fold CV 사용

3 모델링 전략

교차 검증 (Cross Validation)

K-Fold CV

$$Accuracy = Average(Accuracy_1, \cdots, Accuracy_k)$$

k는 주로 5 ~ 10을 사용!

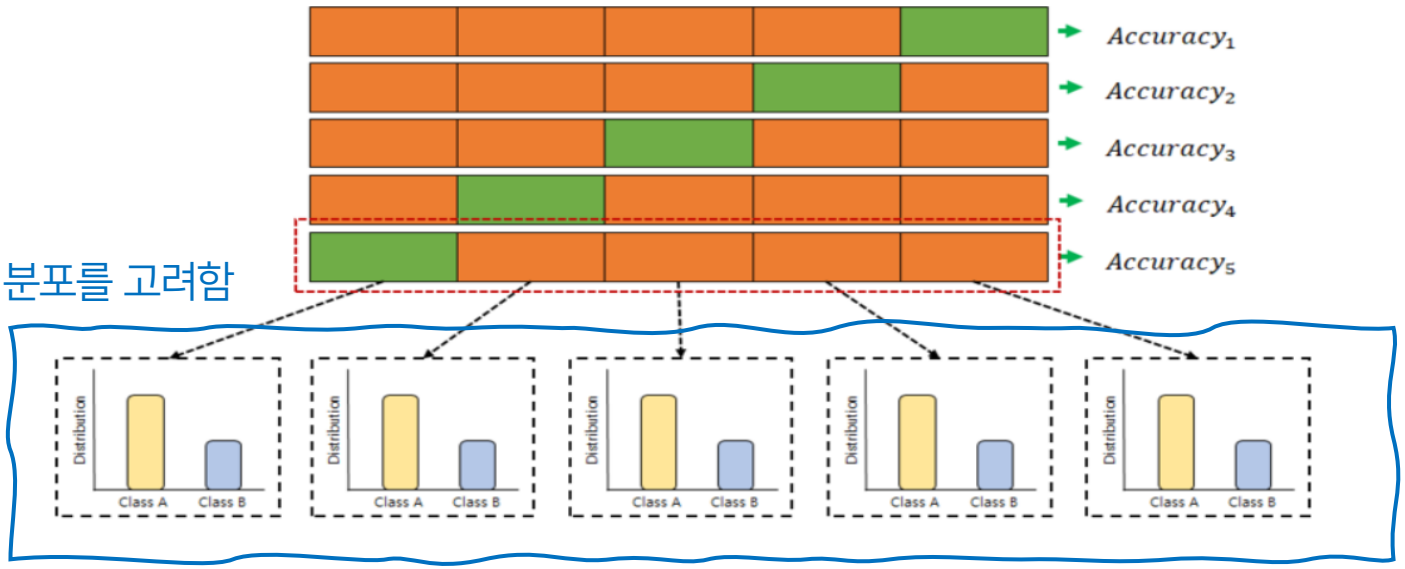
전체 데이터를 **k개의 그룹(fold)**로 나누어

1개의 그룹을 Validation data, **k-1개의 그룹**을 Train data로 사용

3 모델링 전략

교차 검증 (Cross Validation)

Stratified K-Fold CV



전체 label 분포를 고려함

전체 데이터의 **분포를 고려**하여 Fold를 나누는 K-Fold 방법

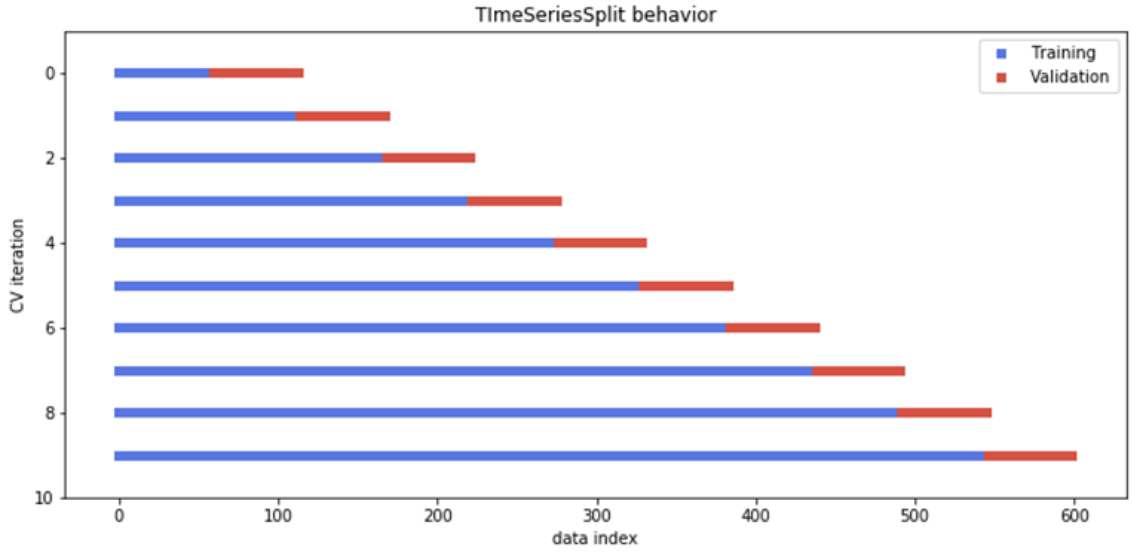
데이터셋이 불균형할 경우 모델이 왜곡되지 않기 위해 사용함

3 과적합 (Overfitting) 문제해결

교차 검증 (Cross Validation)

Time Series CV

(자세한 내용은 시계열팀 클린업 참고)



시계열 데이터는 **전후 데이터 사이의 상관관계**가 존재하므로

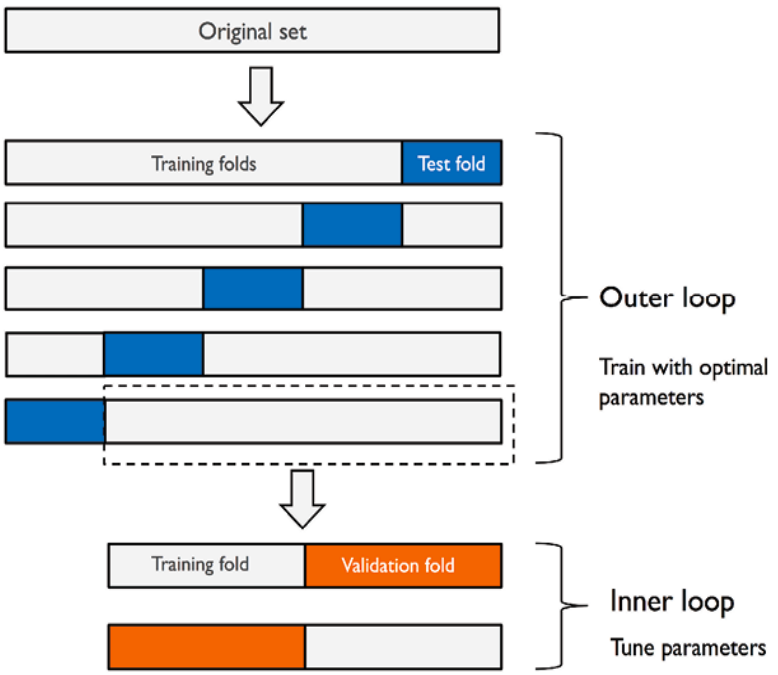
기존 교차 검증 방법 적용 불가

⋮

Train data가 Validation data보다 항상 **앞선** 시간으로 할당!

교차 검증 (Cross Validation)

Nested CV



Outer loop

Test set을 여러 fold로 나누어 구성함

Inner loop

Train set에 대해서 다시 Train fold와 Validation fold로 나누어 검증

⋮

한번의 Test로 결과를 신뢰하기 힘든 경우에 사용!

차원의 저주 (Curse of Dimensionality)

차원의 저주

독립 변수의 개수가 늘어남으로 인해 차원의 수가 늘어나고,

데이터의 특징이 많아져서 **과적합의 원인**이 됨

3 모델링 전략

차원의 저주 (Curse of Dimensionality)

독립 변수의 개수가 많으면 데이터가 **고차원**의 공간을 가짐

관측치 간의 거리가 **기하급수적**으로 멀어져

데이터셋이 전체 공간을 충분히 나타내지 못하게 됨

차원의 저주?

독립 변수의 개수가 늘어남으로 인해 차원의 수가 늘어나고,

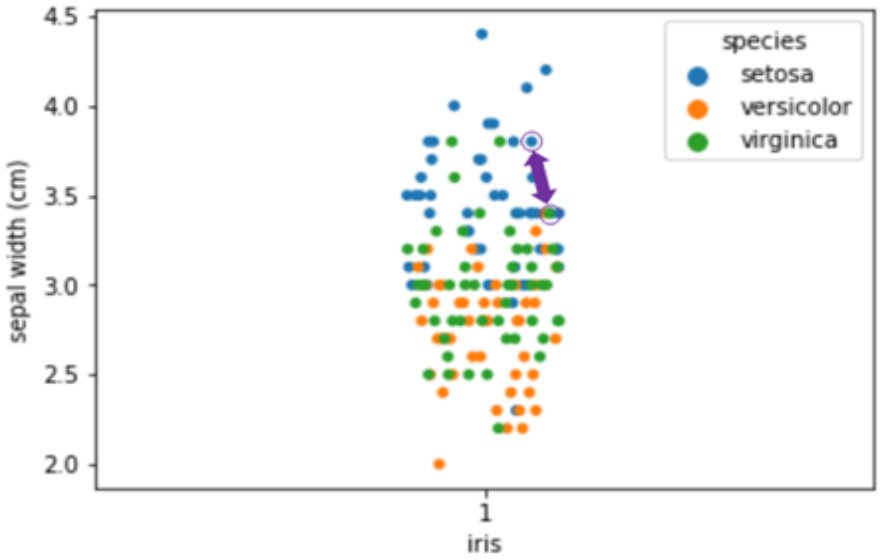
데이터의 특징이 많아져서 과적합이 일어나는 문제

모델이 패턴을 학습하기 어려움 😢

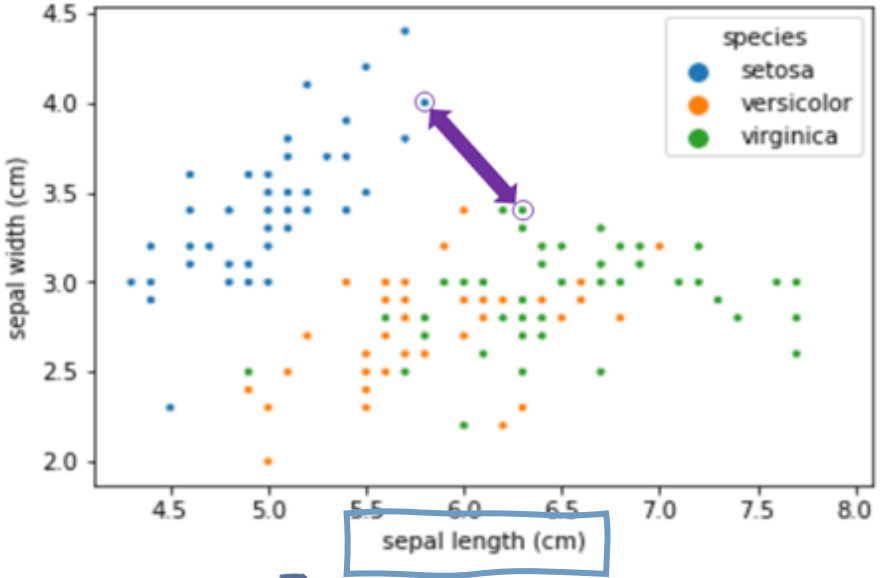
차원의 저주 (Curse of Dimensionality)

Ex) KNN

▼ 변수가 1개인 경우



▼ 변수가 2개인 경우



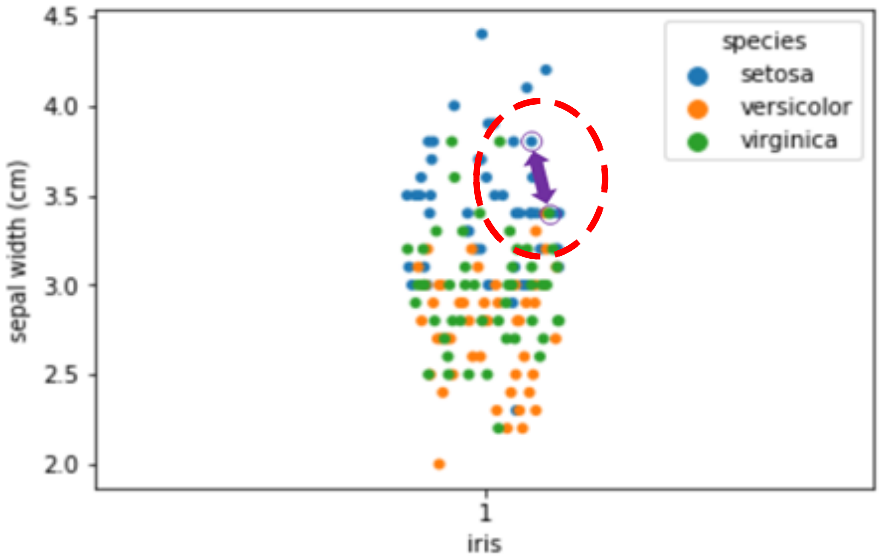
"Sepal Length"라는 독립변수 추가

3 모델링 전략

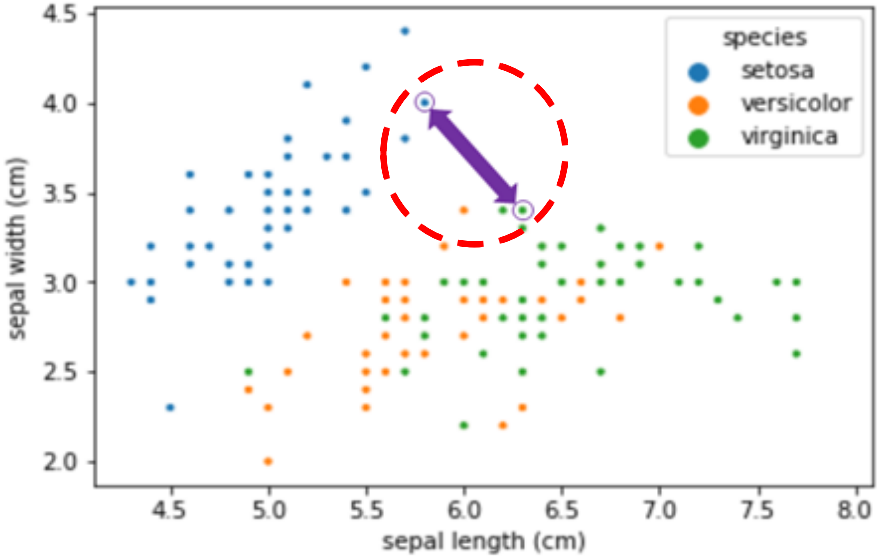
차원의 저주 (Curse of Dimensionality)

Ex) KNN

▼ 변수가 1개인 경우



▼ 변수가 2개인 경우



변수가 늘어남에 따라 **두 점 간 거리가 멀어짐 !**

3 모델링 전략

차원의 저주 (Curse of Dimensionality)

차원의 저주를 어떻게 해결할까?

차원이 무한히 커진다면 어느 포인트끼리 거리를 구해도
차원이 큰 것이 문제였으니 차원을 줄여보자!

패턴을 찾기 어려움

차원 축소

(Dimension Reduction)

3 모델링 전략

차원의 저주 (Curse of Dimensionality)

변수선택법
(Feature Selection)

중요한 변수만 선택, **불필요한 변수들은 제거**

전진선택법

특성을 가장 잘 설명하는

변수들을 **하나씩 더해감**

후진선택법

특성을 잘 설명하지 못하는

변수들을 **하나씩 제거함**

단계적선택법

전진과 후진 거듭

사용하는 평가 지표 :

AIC, BIC, R^2_{adj}, Mallow's Cp

(자세한 내용은 회귀팀 클린업 참고)

3 모델링 전략

차원의 저주 (Curse of Dimensionality)

변수선택법
(Feature Selection)

중요한 변수만 선택, 불필요한 변수들은 제거

전진선택법

특성을 가장 잘 설명하는

변수들을 하나씩 더해감

후진선택법

특성을 잘 설명하지 못하는

변수들을 하나씩 제거함

단계적선택법

전진과 후진 거듭

변수추출법
(Feature Extraction)

고차원의 데이터를

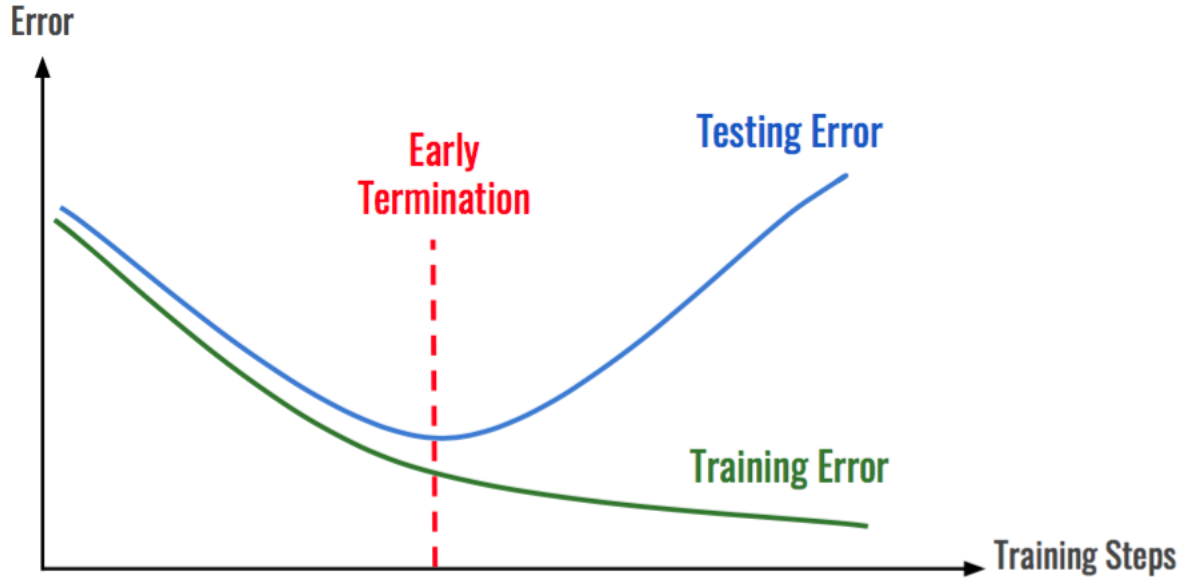
저차원 공간의 데이터로 변환

PCA

(Principal Component Analysis)

(자세한 내용은 선대팀 클린업 참고)

차원의 저주 (Curse of Dimensionality)



조기 종료 (Early Stopping)

학습에 소요되는 **시간에 제한**을 두거나, 모델 **성능이 일정 수준** 이상이 됐을 때

학습을 자동으로 종료하는 조건을 설정

↓

Testing Error가 증가하기 전에 빠르게 학습 종료

다음 주 예고

1. 트리 기반 모델
2. 클러스터링
3. 비선형 모델

감사합니다