

정의 및 접근법

여러 학문과 밀접히 맞닿아있어!

Data Mining

통계학

Pattern
Recognition

Machine
Learning

AI



Databases

주요한 인사이트 채굴이 목표!

프로세스: CRISP-DM

1.

비즈니스 문제 이해

- 비즈니스 상황의 배경지식 쌓기
- 데이터 마이닝 과정의 성공 여부 기준 세우기

2.

데이터 이해
(EDA)

- 시각화를 통해 데이터 직관적 이해 달성
- 변수의 의미, 변수 간의 관계 파악
- 이상치, 결측치 유무 파악

3.

데이터 준비

- 데이터 전처리 과정
- 모델의 성능 개선에 주요한 역할

프로세스: CRISP-DM

4.

데이터 분석과
모델링

- 머신러닝 / 딥러닝 기법 적용
- 추천, 예측, 해석 등

5.

분석 모델의
평가

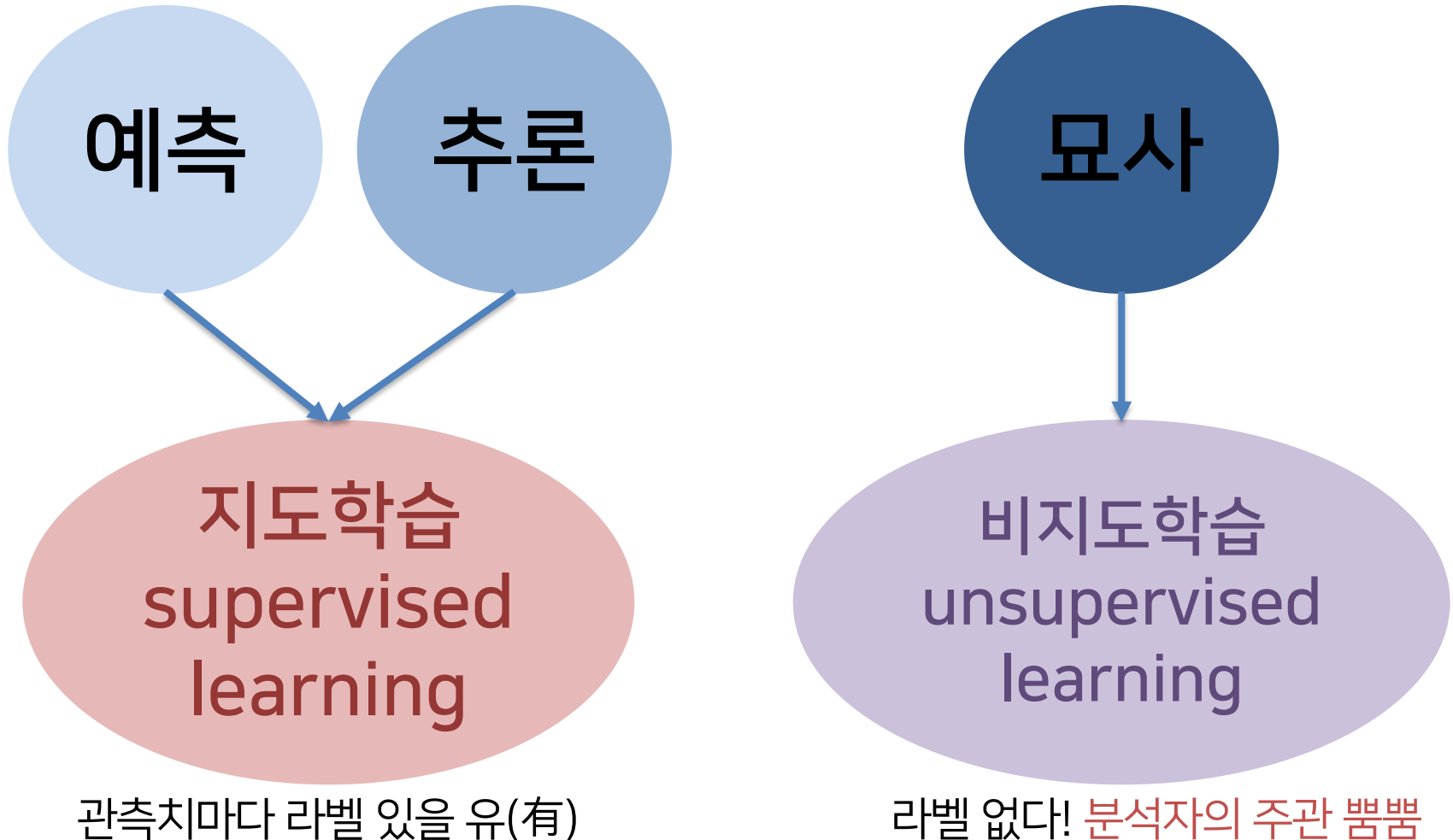
- '모델링이 잘 되었는지' 평가
- 범주형 데이터 (misclassification rate)
- 연속형 데이터 (RMSE)

6.

분석 결과의
적용

- 실제 비즈니스 상황에 적용

| 학습 목적에 따른 분류



지도학습 (supervised learning)

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$

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

실제 수학적식, 그러니까 모델의 형태 알 수 없어!

➡ **“추정”** 하자!

우리의 모델이 실제와 최대한 유사하도록.

편향-분산 트레이드 오프(Bias-Variance Tradeoff)

MSE

=

Irreducible
Error

+

Reducible
Error



표본 추출로 인해 발생하는 Irreducible error (δ^2) 보다는,
Reducible error (편향+분산)를 최소화 하는 모델 설계가 목표!

* **Bias**는 편향, 즉 f 와 \hat{f} 의 차이!

* **$Var(\hat{f})$** 은 모델의 분산으로, 매번 다른 표본 추출마다 얼마나 다양한 형태를 나타내는 정도!

편향-분산 트레이드 오프(Bias-Variance Tradeoff)



Overfitting [과적합 문제]

과적합 발생 위험 증가!

우리에게 주어진 데이터에 대해서만
완벽히 설명하는 모델 설계

↓ 새로운 데이터가 들어온다면?

새로운 데이터에 대한 설명력 확보하지 못함!

모델의 '재사용성'은 이렇게 날라가고...!

Overfitting [과적합 문제]

따라서,

과적합 발생 위험 증가!
우리가 설계한 **모델의 성능**을 평가할 때

조금 더 객관적일 필요가 있다!

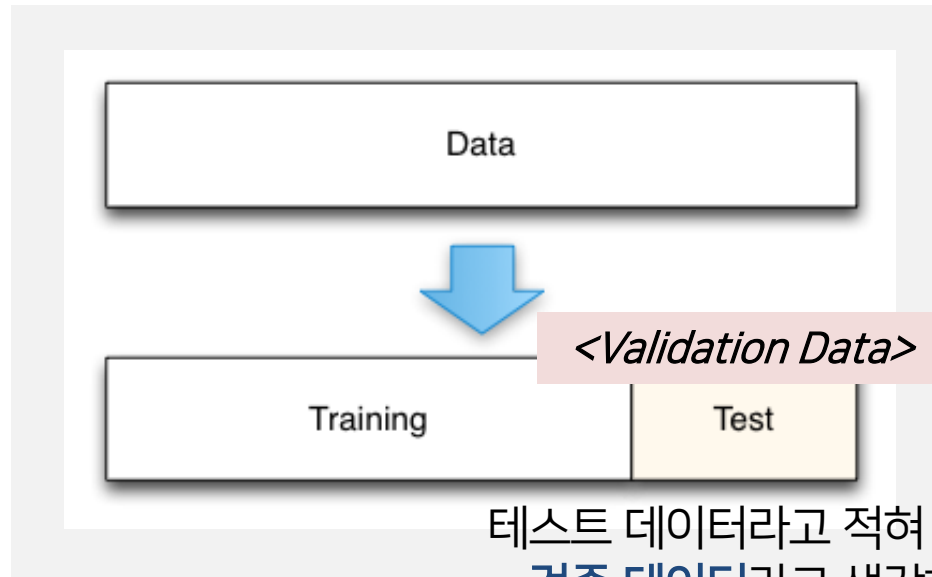
우리에게 주어진 데이터에 대해서는
완벽히 설명하는 모델 설계

✓ 새로운 데이터 (**검증 데이터**)에 대해서
새로운 데이터가 들어온다면?

새로운 데이터에 대한 모델의 성능은?
어떻게 반응할지 궁금하다!

모델의 '재사용성'은 이렇게 날라가고...!

Train-test split [검증데이터 분할]



테스트 데이터라고 적혀 있지만
검증 데이터라고 생각하자!

“

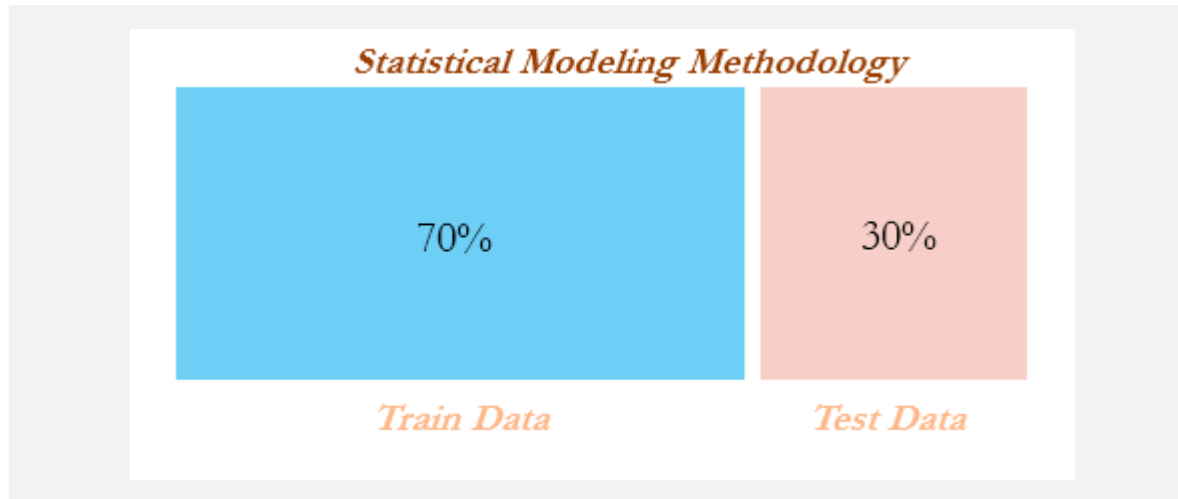
학습 데이터의 일부를

”

검증 데이터로 삼아 모델의 성능을 평가해보자!

Train-test split [검증데이터 분할]

“What is the most proper ratio?”



학습 데이터 : 검증 데이터를
7:3 혹은 8:2 비율로 두면
오버피팅 상황과 언더피팅 상황을 적절히 피해!

K-fold CV [K-fold 교차검증]

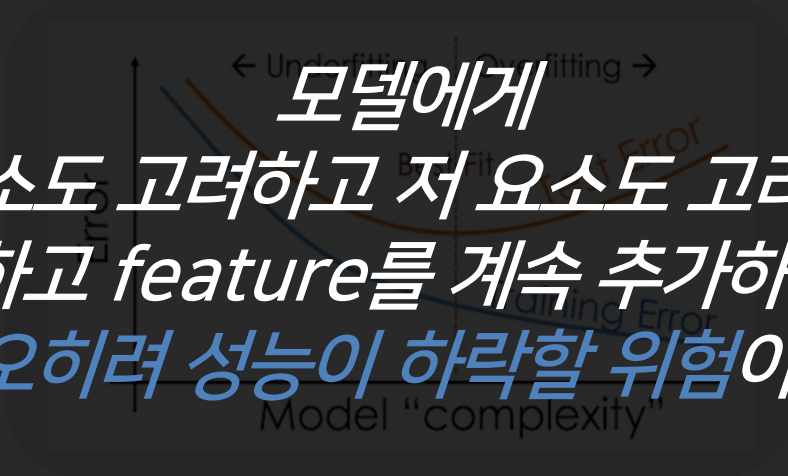
주의

우리나라에서 만들어서 앞에 K 붙은 거 아님

Estimation 1	Test	Train	Train	Train	Train
Estimation 2	Train	Test	Train	Train	Train
Estimation 3	Train	Train	Test	Train	Train
Estimation 4	Train	Train	Train	Test	Train
Estimation 5	Train	Train	Train	Train	Test

전체 데이터를 **k개의 그룹(fold)으로** 나눈 후
 한 개의 데이터셋을 검증 데이터셋으로,
 나머지 k-1개의 데이터셋을 학습 데이터셋으로 사용

차원의 저주 [Curse of Dimensionality]



모델에게
'이 요소도 고려하고 저 요소도 고려하렴~'
하고 *feature*를 계속 추가하면
오히려 성능이 하락할 위험이!!

→ **오버피팅 문제 야기**가 너무 많아!"

모델의 복잡도가 높아짐에 따라
검증 데이터 *error*는 다시 증가하는 모습을 보이는데...

자세한 내용은 회귀/선대/딥팀 교안 참고하시고~!

차원의 저주 [Curse of Dimensionality]

따라서, 너무 적지도 많지도 않은 **적절한 변수 개수를 설정**해야 하는데...

몇 가지 방법 소개해드릴름...!

1. Feature Selection

EX) Forward Selection,
Stepwise Selection

2. Feature Extraction

EX) Principal Component
Analysis(PCA, 주성분 분석)

3. Early Stopping

이건 변수 선택의 문제라기보다
사용자 지정 accuracy 지점에 도달했을 때 모델의 학습 과정을 멈추는 것!