



**문화 A0007**

# 데이터사이언스입문

**김 태 완**

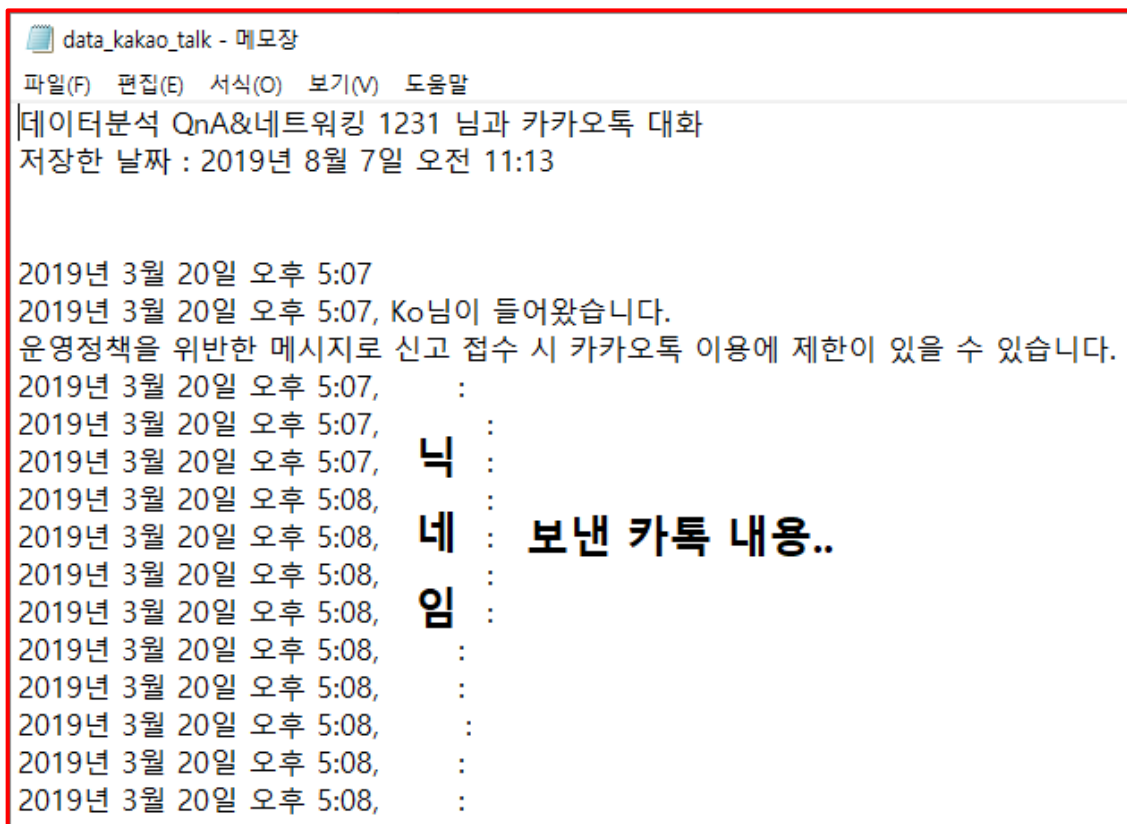
**[kimtwan21@dongduk.ac.kr](mailto:kimtwan21@dongduk.ac.kr)**

# 텍스트마이닝

- 텍스트 마이닝

- 비정형 데이터 : 텍스트

- 비정형 텍스트 데이터로부터 유용한 정보를 추출하는 기술
    - 웹페이지, 블로그, 이메일 등 전자문서로 된 텍스트로부터 유용한 정보를 추출하여 분석하기 위한 도구
    - 데이터마이닝, 자연어처리, 정보 검색등의 이론 및 실무까지 다양한 분야가 융합되어 있는 영역



# 텍스트마이닝

---

- 텍스트 마이닝
  - 주요 기술 분야
    - **문서 분류 (Document Classification)**
      - 도서관에서는 수 많은 도서의 관리를 위해 사서가 각 도서의 내용을 일일이 파악해 정해진 분류체계에 따라 수작업으로 분류 했으나, 시간이 흐를수록 불가능한 수준에 도달
      - 조직 내부에 분산되어 있는 수 많은 정보가 상호 복잡하게 연계되어 있고, 이질적 목적과 형태를 지닌 지식 콘텐츠의 자동 분류 기술 구현은 최근까지도 매우 어려운 과제로 인식
    - **문서 군집 (Document Clustering)**
      - 각 지식 콘텐츠의 특성을 파악해 그 내용 혹은 형태가 유사하거나 상호 관련성이 높은 콘텐츠들을 군집시켜 주는 기술
      - 대상 문서의 언어학적 분석을 통해 차별화된 중요 특성들을 추출해 내고, 이를 다른 문서의 특성들과의 비교(유사도 계산)하여 그 유사도가 높은 문서들을 상호 묶어주는 방식으로 구현

# 텍스트마이닝

---

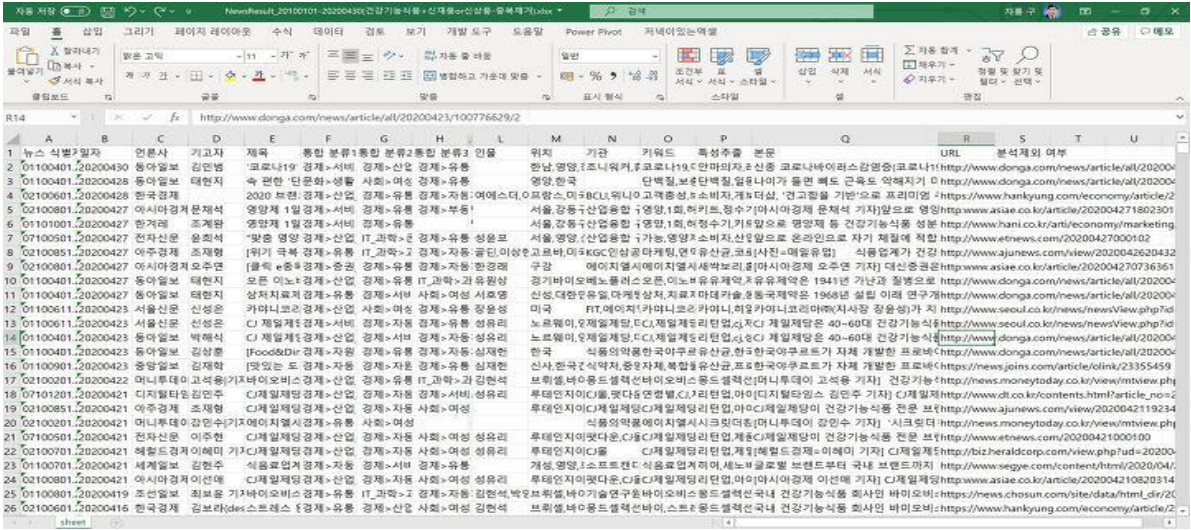
- 텍스트 마이닝
  - 주요 기술 분야
    - **정보 추출 (Information Extraction)**
      - 텍스트 문서내에서 중요한 의미를 가지는 정보들을 자동으로 추출해 주는 기술
      - 문서에서 중요 키워드, 핵심 개념, 특정 사건, 인명, 지명, 날짜, 상황 및 조건, 결론 등의 다양한 정형 정보를 추출하여 활용
      - 키워드와 같은 기본적인 정보는 자동 분류, 군집 등에 직접적으로 활용되는 중요 요소
    - **문서 요약 (Document Summarization)**
      - 문서가 담고 있는 핵심 의미를 유지하면서 그 복잡도와 길이를 효과적으로 줄여주어 각 사용자가 짧고 간단한 요약 문장을 파악함 으로서 빠르게 정보를 이해하고 활용 할 수 있도록 돕기 위한 기술
      - 특성 추출 및 정보 추출 기술에 기반하고 있으며, 텍스트 전체에서 그 문서를 대표 할만한 문장을 추출하여 재구성하는 추출 요약 방식과 추출한 중요 정보들을 활용하여 문장을 생성해내는 생성 요약 방식으로 분류 가능

데이터 마이닝 vs 텍스트마이닝

데이터 마이닝

이름	학번	점수
김태완	20221234	90
박아무개	20221451	92
...	...	...

텍스트 마이닝



	데이터 마이닝	텍스트 마이닝
대상	수치 또는 범주화 데이터	텍스트 데이터
목적	미래 상황 결과의 예측	적합한 정보를 획득 후 의미를 정제하고 범주화
방법	기계 학습	기계학습, 인덱싱, 언어처리, ontology 등
성숙도	1994년 이후 광범위하게 구현	2000년 이후 광범위한 구현 시작

## 텍스트 분석 절차



# 텍스트 데이터 수집

---

- 텍스트 데이터 수집 방법
  - Scraping
    - 스크래핑(혹은 웹 스크래핑)은 인터넷에서 존재하는 데이터를 컴퓨터 프로그램을 통하여 자동화된 방법으로 웹에서 데이터를 수집하는 모든 작업
    - 크롤링은 많은 사람들이 스크래핑과 혼용하여 사용하고 있지만, 사실은 그 의미가 상이
  - Crawling
    - 크롤링은 데이터를 수집하고 분류하는 것을 의미하며, 주로 인터넷 상의 웹페이지(html, 문서 등)를 수집해서 분류하고 저장하는 것을 뜻함
    - 크롤링은 데이터의 수집 보다는 여러 웹페이지를 돌아다닌다는 뜻이 강하며, 데이터가 어디에 저장되어 있는지 위치에 대한 분류 작업이 크롤링의 주요 목적
- 참고 : Parsing
  - 프로그램 언어를 문법에 맞게 분석해 내는 것
  - 어떤 웹페이지의 데이터를 사용자가 원하는 형식, 즉 일정한 패턴으로 추출해 어떠한 정보를 만들어 내는 것
  - 어느 위치에 저장된 데이터에 접근을 했다면, 이 데이터를 원하는 형태로 가공하는 작업이 주요 목적

## 텍스트 데이터 수집 – crawling

---

- 크롤링 방법
  - 연구자가 직접 코드 작성 (Python, R 등)
  - 데이터를 수집하고자 하는 사이트 혹은 회사에서 제공하는 API 사용
    - API (application programming interface)
      - 크롤링의 경우에는 사용자에게 데이터를 쉽게 제공하기 위한 툴
      - 빠른 시간에 정제된 형태의 데이터를 수집 가능
      - 데이터를 소유하고 있는 기업에서 제공하는 툴이기 때문에 데이터 사용에 따른 법적·윤리적 문제가 거의 없지만 최소한의 프로그래밍은 필요



## 텍스트 데이터 수집 – crawling

- 파이썬을 이용한 간단한 크롤링 방법 예시
  - 정보를 가져오고자 하는 url 정의
  - url 정보로 requests로 정보 요청

```
import numpy as np
import pandas as pd
import requests
from bs4 import BeautifulSoup

# url정의
url = 'https://naver.com'

# requests로 url에 정보요청
response = requests.get(url)

# 정보를 html 변환
html = BeautifulSoup(response.text, "html.parser")

html.select('span')[0:10]
```



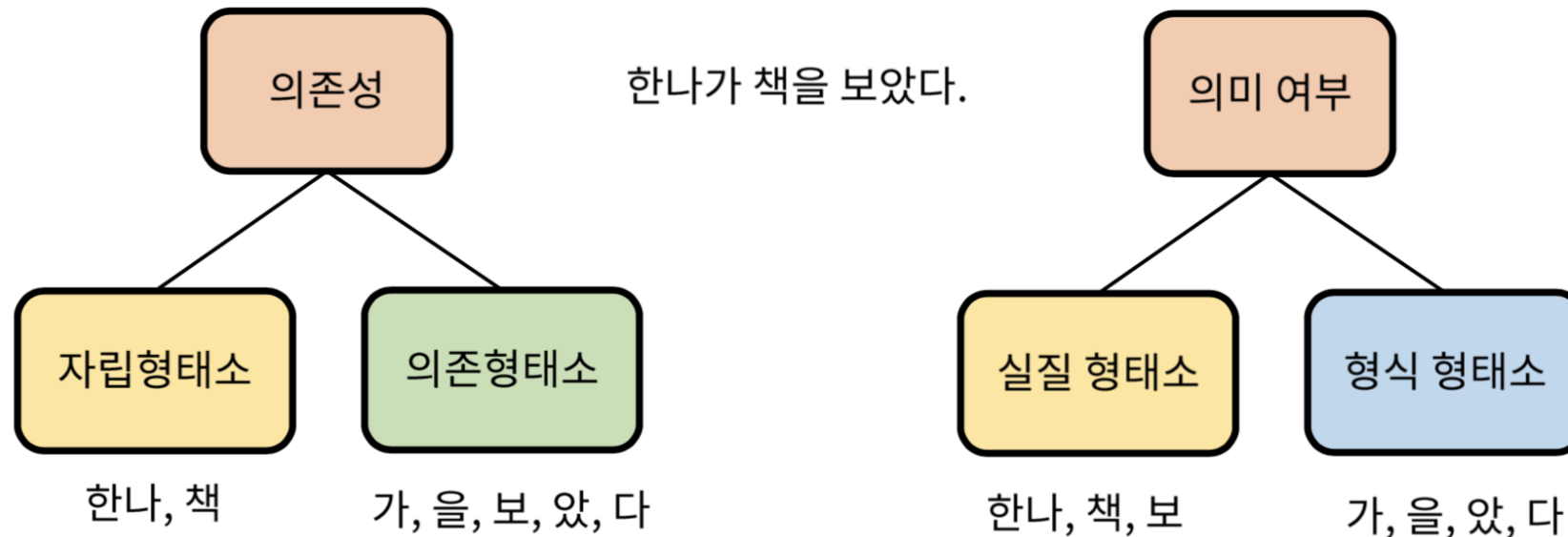
```
PS C:\Users\enoug\Desktop> & "C:/Program Files/Python310/python.exe" c:/Users/enoug/Desktop/python/tmp.py
[<span>뉴스스탠드 바로가기</span>, <span>주제별캐스트 바로가기</span>, <span>타임스퀘어 바로가기</span>, <span>쇼핑캐스
트 바로가기</span>, <span>로그인 바로가기</span>, <span class="blind">NAVER whale</span>, <span class="_1syGnXOL _3di88
A4c" data-clk="dropbanner1b" style="padding-right: 20px; color: black; padding-left: 20px"><span>"돈 있는데 왜 사지를
못해" 직구 치트키, </span><strong style="color: #0da888">홈페이지 번역</strong></span>, <span>"돈 있는데 왜 사지를 못해
" 직구 치트키, </span>, <span style="background-color: #0da888">다운로드</span>, <span class="blind">네이버</span>]
```

# 텍스트 데이터 분석 방법

- 텍스트 분석하기 위해 텍스트 분리하는 방법

- 형태소 분석

- 주어진 텍스트를 단어와 문법적 특성에 맞추어 명사, 동사, 꾸밈어, 조사 등의 형태소로 분리
    - 형태소 (Morphology)
      - 의미가 있는 최소 단위로서 더 이상 분리가 불가능한 가장 작은 의미 요소
      - 즉, 일반적으로 문법적, 관계적인 뜻을 나타내는 단어 또는 단어의 부분



# 텍스트 데이터 분석 방법

---

- 텍스트 분석하기 위해 텍스트 분리하는 방법
  - 텍스트 전처리 (text pre-processing)
    - 텍스트 분석을 위해 문장 분리, 불필요한 문장 성분을 제거하는 과정
  - 텍스트 전처리 과정
    - 토큰화(tokenization): 문서를 토큰(token)이라 불리는 단위로 나누는 작업
      - 예시 : “텍스트 분석을 위해서 파이썬을 이용합니다.”
        - ['텍스트', '분석', '을', '위해', '서', '파이썬', '을', '이용', '합니다', '.']
    - 정제(cleaning): 불필요한 단어 또는 문자를 제거
    - 정규화(normalization): 같은 의미이면서 표현이 다른 단어를 통합
      - 규칙기반 통합
        - 예시: “US”, “USA”, “United States”, ...
      - 대, 소문자 통합
        - 예시: “Automobile” = “automobile”
      - 불필요 단어 제거: 출현빈도가 작은 단어 또는 길이가 짧은 단어
        - 예시 : a, am, the, ...

# 텍스트 데이터 분석 방법

- 텍스트 분석하기 위해 텍스트 분리하는 방법
  - 품사 태깅 (POS tagging : Part-of-speech tagging)
    - 문장에서 각각의 단어를 해당하는 품사로 레이블링 하는 작업
    - 하나의 단어가 여러 품사를 갖을 수 있기 때문에, 품사의 모호성 (혹은 중의성)을 제거하는 과정
    - 문장에 사용된 형태소들의 품사를 파악하고 문장의 구조도 파악할 수 있음
    - 예시 : “나는 도서관에 간다.”



## 텍스트 데이터 분석 방법

---

- 텍스트 분석하기 위해 텍스트 분리하는 방법
  - 키워드 추출
    - 가용어의 이해 : 불용어가 아닌 단어들
    - 불용어의 이해 : 단어 성분 중에서 문서의 정보(의미)를 표현하지 못하는 단어
      - 한국어 : 조사 ('는', '을', ...)
    - 키워드의 개념 : 가용어의 중심이 되는 단어
  - 키워드 선정 방법
    - 일반적으로 분석하고자 하는 목적 및 데이터의 종류에 영향을 받지만,
    - 보통 문서 내에서 발생 빈도가 높은 단어들을 키워드로 선정
    - 불용어 처리
      - 불용어가 저장된 데이터 베이스를 참조하여 키워드에서 제외
        - 형태소 분석 결과를 불용어 사전에서 검색하여 일치하는 내용이 등장하면 삭제

## 텍스트 데이터 분석 방법

- 텍스트 분석하기 위해 텍스트 분리하는 방법
  - 키워드 추출

나는 햄버거를 먹었다.

형태소 분석

불용어 처리

불용어 사전

가용어 리스트 : [나, 햄버거, 먹다]

키워드 추출

햄버거 0.6  
먹다 0.4

텍스트 마이닝

- 텍스트 분류
- 텍스트 군집
- 텍스트 요약
- 특성 추출

결과 문서

## Bag of Words (BOW)

---

- Bag of Words (BOW)
  - 문장, 문단 또는 전체 문서들과 같은 텍스트를 단어의 집합으로 표현하는 방법
  - 문법이나 단어들의 순서는 고려하지 않음
  - 다음 두 문장의 bow는 동일
    - 철수는 영희보다 공부를 잘한다.
    - 영희는 철수보다 공부를 잘한다.
- 예시: awesome thank you



## Bag of Words (BOW)

---

- great thank you



- not bad not good

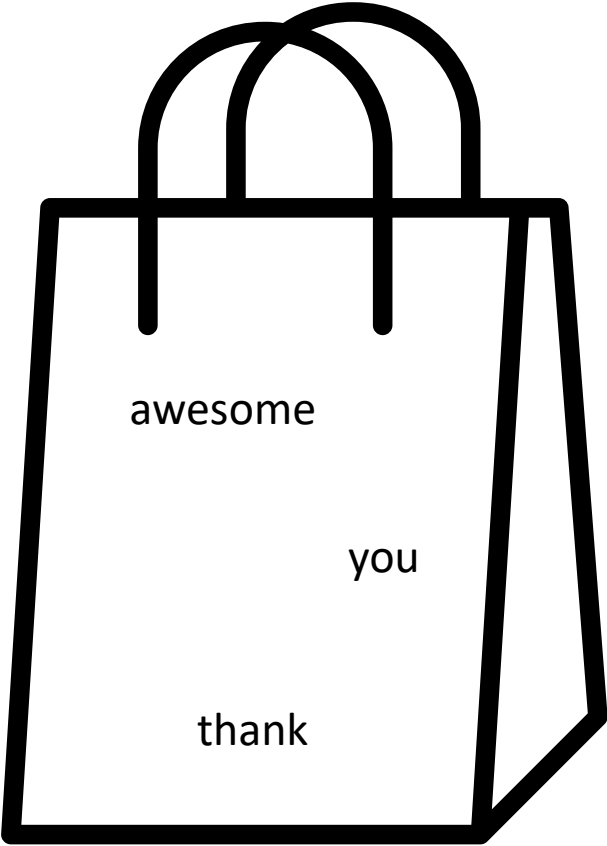




# Bag of Words (BOW)

awesome	thank	you	great	not	bad	good
1	1	1	0	0	0	0

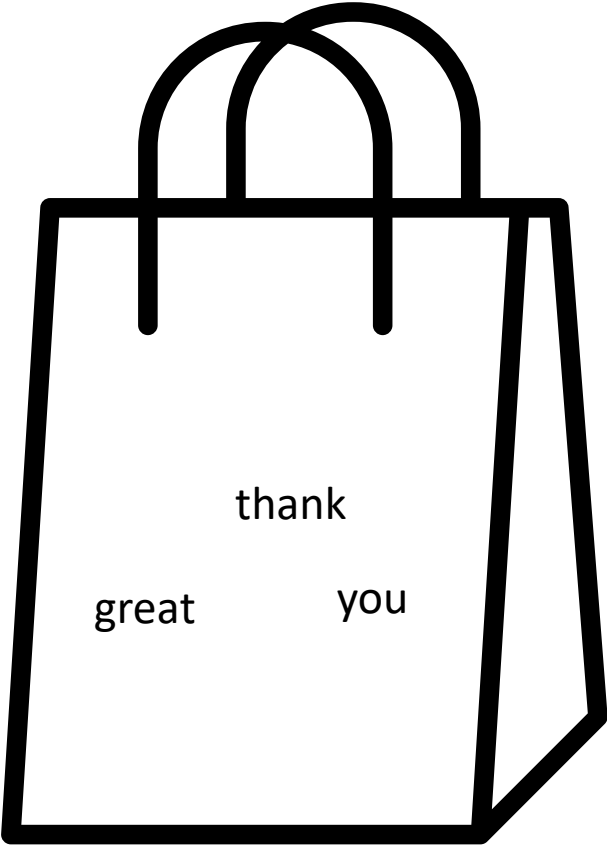
[1, 1, 1, 0, 0, 0, 0]



# Bag of Words (BOW)

awesome	thank	you	great	not	bad	good
0	1	1	1	0	0	0

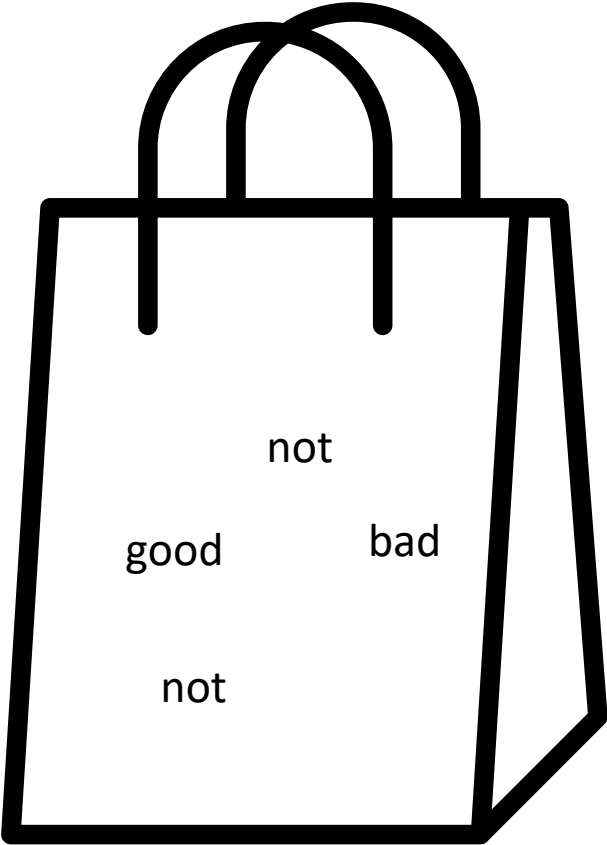
[0, 1, 1, 1, 0, 0, 0]



# Bag of Words (BOW)

awesome	thank	you	great	not	bad	good
0	0	0	0	2	1	1

[0, 0, 0, 0, 2, 1, 1]



## Bag of Words (BOW)

---

- awesome thank you **[1, 1, 1, 0, 0, 0, 0]**
- great thank you **[0, 1, 1, 1, 0, 0, 0]**
- not bad not good **[0, 0, 0, 0, 2, 1, 1]**

- awesome thank you

[1, 1, 1, 0, 0, 0, 0]

x x x x x x x

- great thank you

[0, 1, 1, 1, 0, 0, 0]

1 + 1 = 2

- great thank you

[0, 1, 1, 1, 0, 0, 0]

x x x x x x x

- not bad not good

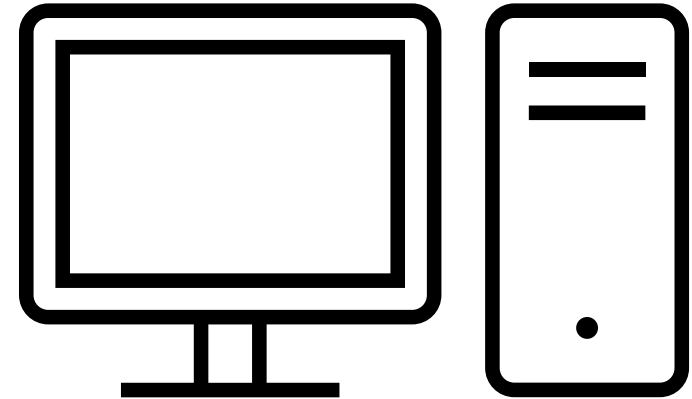
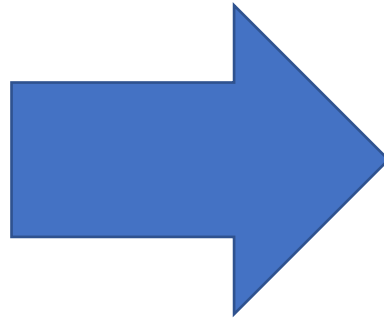
[0, 0, 0, 0, 2, 1, 1]

0

• awesome thank you

• great thank you

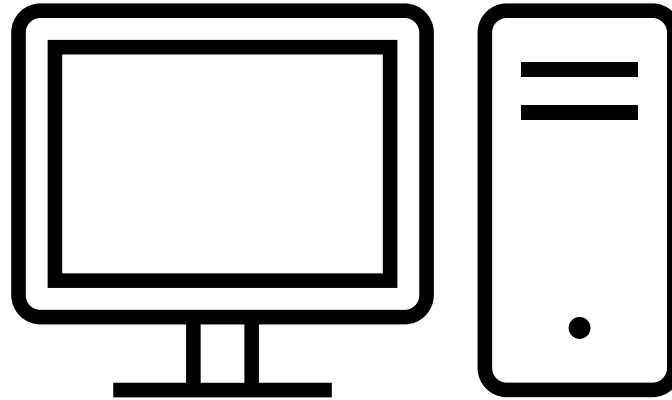
• not bad not good



- **[1, 1, 1, 0, 0, 0, 0]**

- **[0, 1, 1, 1, 0, 0, 0]**

- **[0, 0, 0, 0, 2, 1, 1]**



**HAPPY**

**HAPPY**

**UNHAPPY**

## Bag of Words (BOW)

- Bag of Words (BOW)의 한계
  - Sparsity
    - 만약 우리가 모든 영단어를 BOW에 이용할 경우 vector 안의 많은 부분이 0으로 채워질 것

• **awesome thank you**      **[1, 1, 1, 0, 0, ..., 0, 0, 0]**

- 불용어의 영향을 많이 받음

- The man like the girl

the	man	girl	like	love
2	1	1	1	0

- the man love the girl

the	man	girl	like	love
2	1	1	0	1

- the the the the the

the	man	girl	like	love
5	0	0	0	0



## Bag of Words (BOW)

---

- Bag of Words (BOW)의 한계
  - 단어 순서 무시 : 문장의 뜻을 무시함
    - 철수는 영희보다 공부를 잘한다. vs 영희는 철수보다 공부를 잘한다.
    - Home run vs Run home
- Out of vocabulary
  - 오타 : good moning, tank yo, ...
  - 처음 본 단어 이해 못함: asap, goood, ...

# 텍스트 데이터 분석 방법

---

- 텍스트 데이터 분석 방법

- 주제어 (키워드) 빈도 분석

- 텍스트 자료에 포함된 특정 단어들의 빈도에 따라 주요 단어를 추출
    - 주제어 빈도 분석은 특정 문서 집단 내에서 자주 언급되는 주제어를 추출하고 이들이 언급되는 빈도에 따라 중요도를 분석하는 방법
    - 특정 단어가 수집된 총 문서에서 얼마나 자주 등장하는지를 나타내는 '단어 빈도(Term Frequency: TF)'로 단순하게 결정 가능

- 용어 - 문서 행렬 (Term-Document Matrix : TDM)

- 각 문서 안에 등장하는 용어들의 출현 횟수를 행렬의 형태로 표현한 것
    - 일반적으로 행은 용어, 열은 문서를 표시하며, 행과 열은 바꿀 수 있음
    - 각 셀은 각 용어가 해당 문서에 등장하는 빈도수 또는 출현여부, TF-IDF 등으로 표현됨
    - TDM을 통해 각 용어와 문서 간의 관계를 알 수 있음

# 텍스트 데이터 분석 방법

---

- 텍스트 데이터 분석 방법

- 주제어 (키워드) 중요도 분석

- 단순하게는, TF 값이 큰 단어일수록 중요도가 높다고 판단 가능
    - TF 값이 큰 단어는 모든 문서에서 자주 등장하는, 즉 그 단어가 흔하게 등장한다는 것을 의미하는 '문서빈도(Document Frequency: DF)' 값이 큰 단어일 수 있음
    - 따라서 중요도 높은 키워드 도출을 위해 TF-IDF라는 값을 사용

- TF-IDF

- 단어의 중요도를 측정하는 방법으로 TF와 DF의 역수 값 (inverse DF)의 곱으로 표현
  - 모든 문서에서 자주 출현하는 상투어 (불용어)를 걸러 내기 위함
  - 단순한 단어의 빈도 처리가 아닌 단어의 출현 확률을 기준으로 출현 빈도를 재가공

---

$$\text{TF-IDF} = \text{TF} * \log(\text{N} / \text{DF})$$

---

TF: 문서 내 특정 단어의 빈도수  
N: 분석 대상 문서 통합  
DF: 특정 단어를 포함하는 문서 빈도 수

---

# 텍스트 데이터 분석 방법

- TF-IDF 예시 (N = 4)

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	1	0	1	1	22	0
문서2	0	0	0	1	1	0	1	18	0
문서3	0	1	1	0	2	0	0	24	0
문서4	1	0	0	0	0	0	0	34	1

- 과일이 : 문서 4에서의 TF = 1, IDF =  $\log(4/1)$
- 바나나 : 문서 3에서의 TF = 2, IDF =  $\log(4/2)$
- 저는 : 불용어
  - 문서 1에서의 TF = 22, IDF =  $\log(4/98)$
  - 문서 2에서의 TF = 18, IDF =  $\log(4/98)$
  - 문서 3에서의 TF = 24, IDF =  $\log(4/98)$
  - 문서 4에서의 TF = 34, IDF =  $\log(4/98)$

---

$$\text{TF-IDF} = \text{TF} * \log(\text{N}/\text{DF})$$

---

TF: 문서 내 특정 단어의 빈도수  
N: 분석 대상 문서 통합  
DF: 특정 단어를 포함하는 문서 빈도 수

---

# 텍스트 데이터 분석 방법

---

- 텍스트 데이터 분석 방법

- 텍스트마이닝

- TF와 TF-IDF 분석을 발전시킨 것으로, 텍스트 형태로 이루어진 비정형 데이터들을 자연어 처리 방식을 이용하여 정보를 추출하거나 연계성을 파악하는 기법
    - 웹 문서에서 특정 주제어와 매칭되는 단어를 찾아 수를 부여하는 인덱싱(indexing) 검색 기법에서 발전
    - 점차 특정 주제어나 문맥(context)을 기반으로 데이터의 숨은 의미를 탐색하는 데 활용
    - 텍스트 마이닝이 두드러지는 분야는 뉴스 기사 분석으로, 이를 활용하면 텍스트의 문맥에 따라 쟁점을 파악하고 텍스트 간 연계성을 분석할 수 있다는 장점이 있음
  - '연관어 분석(association keyword analysis)' 혹은 '의미망분석(semantic network analysis)'
    - 관심 주제어를 포함한 대상 문서에서 함께 언급된 주제어를 추출하여 관심 주제어와 어떠한 토픽들이 연결되는 지 분석
    - 예를 들어, 하나의 뉴스 기사에서 동시 출현한 용어의 쌍을 추출하고 전체 문서집합에서 주제어의 쌍별 발생 빈도와 연결 관계를 분석하면, 언론 기사 상의 주요 관심 토픽과 그 연계성의 변화 추적 가능

# 텍스트 마이닝과 자연어처리

---

- 자연어 처리 (Natural Language Processing; NLP)
  - 자연어: 인간이 일상에서 사용하는 언어
    - 정보전달의 수단이며, 인간 고유의 능력
    - 인공어에 대응 되는 개념으로 인공어는 특정 목적을 위해 인위적으로 만든 언어로 자연언어에 비해 엄격한 구문을 가짐 (형식언어, 에스페란토어, 프로그래밍 언어)
  - 자연어처리는 기계가 자연어를 이해하고 해석하여 처리할 수 있도록 하는 일
  - 사람의 말을 이해한다는 것 = 시간 순으로 이어지는 사람의 말에서 기억할 것과 기억하지 않을 것을 구분
  - 인간은 무의식적으로 수행하지만, 컴퓨터는 조금만 달라도 못 알아듣는 한계를 극복해야 함
    - 예시 : 전화해 = 연락 부탁해
  - 자연어 처리는 기계가 인간의 언어를 해석하는데 중점이 두어져 있다면  
텍스트 분석은 텍스트에서 의미 있는 정보를 추출하여 인사이트를 얻는데 더 중점
  - 하지만 머신러닝이 보편화되면서 자연어 처리와 텍스트 분석을 구분하는 것이 큰 의미가 없어짐

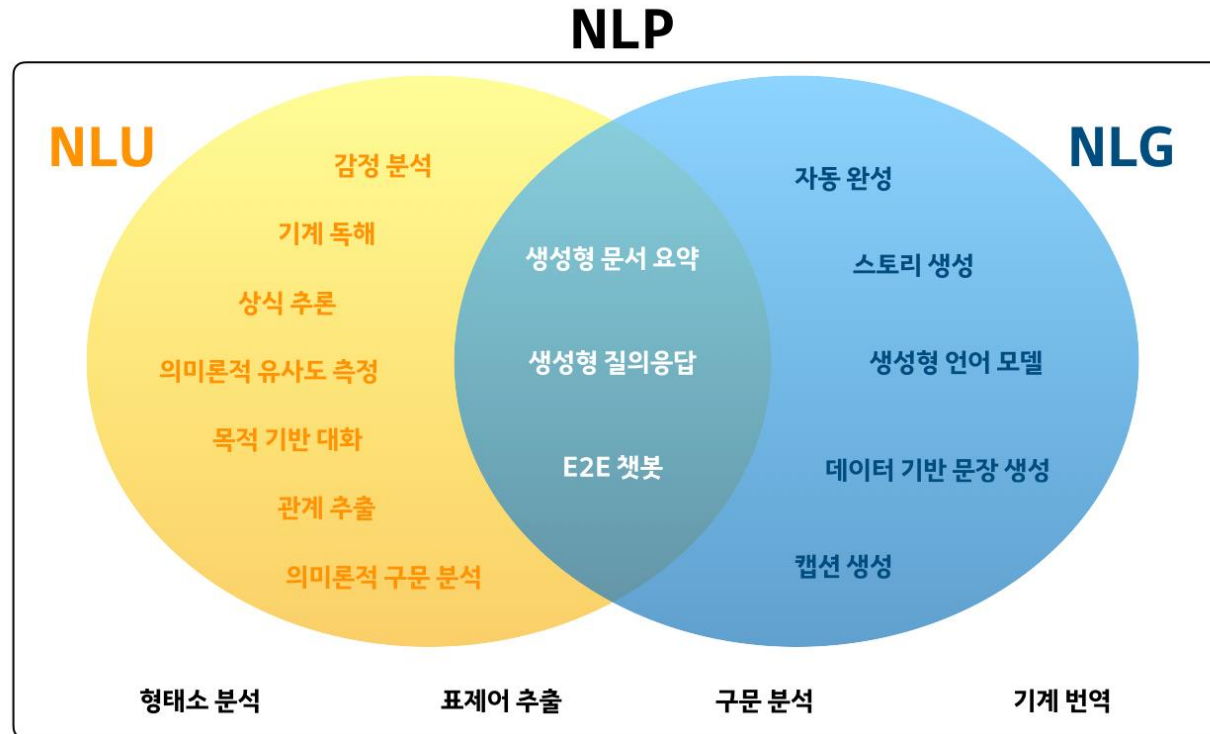
# 자연어처리 (NLP)

---

- 자연어처리 응용분야
  - 인간의 언어가 사용되는 실세계의 모든 영역
  - 정보검색, 질의응답 시스템
    - Google, Naver, Siri, IBM Watson, SK NUGU, KT GIGA GENIE, ...
  - 기계 번역, 자동 통역
    - Papago, google 번역기, ETRI 지니톡, ...
  - 문서작성, 문서요약, 문서 분류, 철자 오류 검색 및 수정, 문법오류 검사 및 수정 등
- IBM Watson
  - [http://www.youtube.com/watch?v=RepnuF8I\\_I0](http://www.youtube.com/watch?v=RepnuF8I_I0)
  - Watson은 자연어 형식으로 된 질문들에 답할 수 있는 인공지능 컴퓨터 시스템
  - 2월 14일부터 16일까지 세 개의 제퍼디 에피소드의 방송에서 왓슨은 금액기준 사상 최대 우승자 브레드 러터, 가장 긴 챔피언십(74번 연속 승리)의 기록 보유자 켄 제닝스와 대결
  - 켄 제닝스와 브레드 러터가 각각 300,000 달러와 200,000 달러를 받는 사이 왓슨은 100만 달러 획득

# 자연어처리 (NLP)

- 자연어 처리
  - 자연어 이해 (Natural Language Understanding)
    - 자연어 표현을 기계가 이해할 수 있는 다른 표현으로 변환
    - 단어나 문장의 형태를 기계가 인식하도록 하는 것이 아닌 의미를 인식
  - 자연어 생성 (Natural Language Generation)
    - 듣고 이해만 하는 과정에서 더 나아가 축적되어 있는 단어들을 조합해서 사용자가 이해할 수 있는 문장으로 출력





# 기계 학습 (Machine Learning)

---

- 기계학습(machine learning)
  - **경험**을 통해서 나중에 유사하거나 같은 **task**를 더 **효율적**으로 할 수 있도록 시스템의 구조나 파라미터를 변경
  - 컴퓨터가 데이터로부터 특정 문제해결을 위한 지식을 자동으로 추출해서 사용할 수 있게 하는 기술

## 경험

필기문자 이미지, 글자

사진, 얼굴영역

이메일, 스팸여부

풍경 사진

바둑 대국

## task

문자 판독(인식)

사진에서 얼굴영역 식별

스팸 이메일 판단

유사한 풍경 사진 식별

바둑두는 방법

## 효율적 (성능)

정확도

정확도

정확도

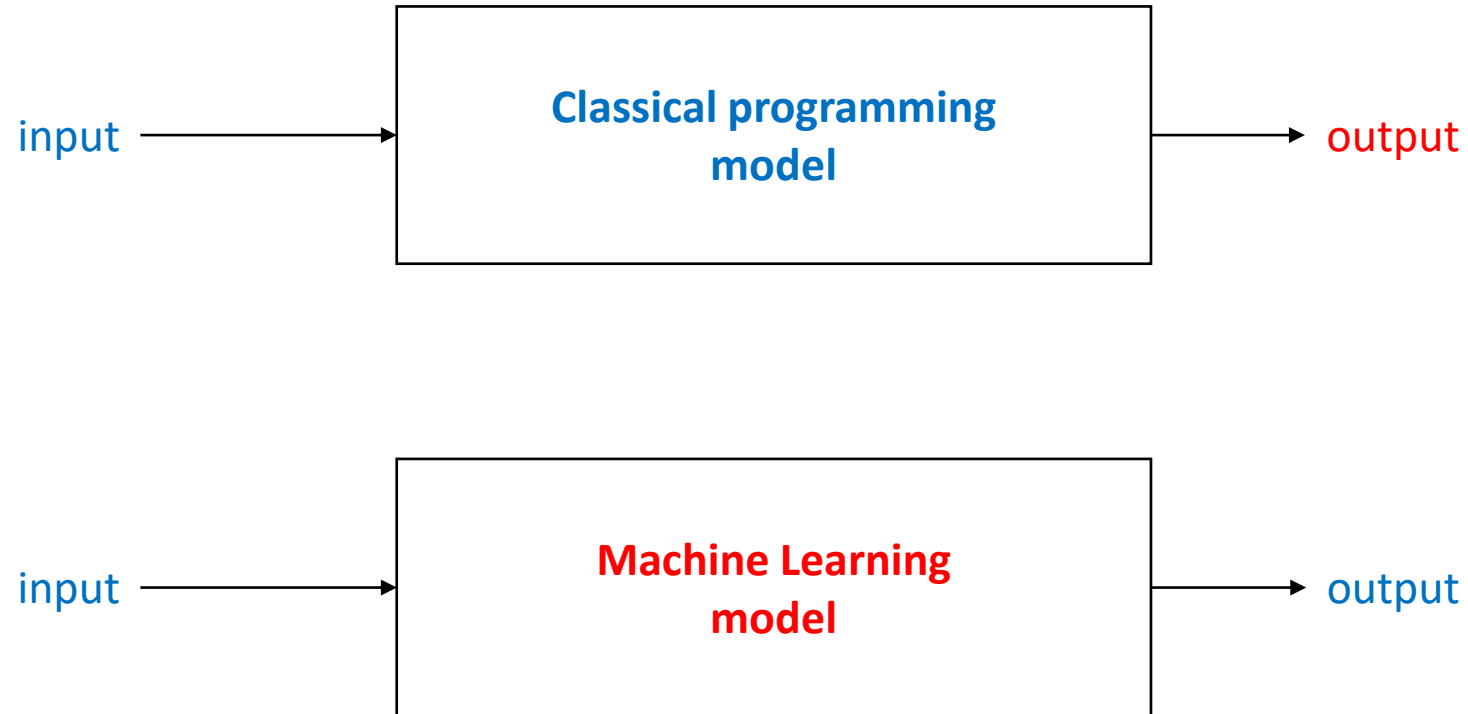
유사도

승률

# 기계 학습 (Machine Learning)

---

- 일반 프로그래밍 방식과 기계 학습
  - 일반 프로그래밍 방식 : 데이터와 룰을 이용해 프로그래밍하여 정해진 답(결과값)을 도출
  - 기계 학습 : 데이터와 결과(기대)값으로 최적의 룰을 도출



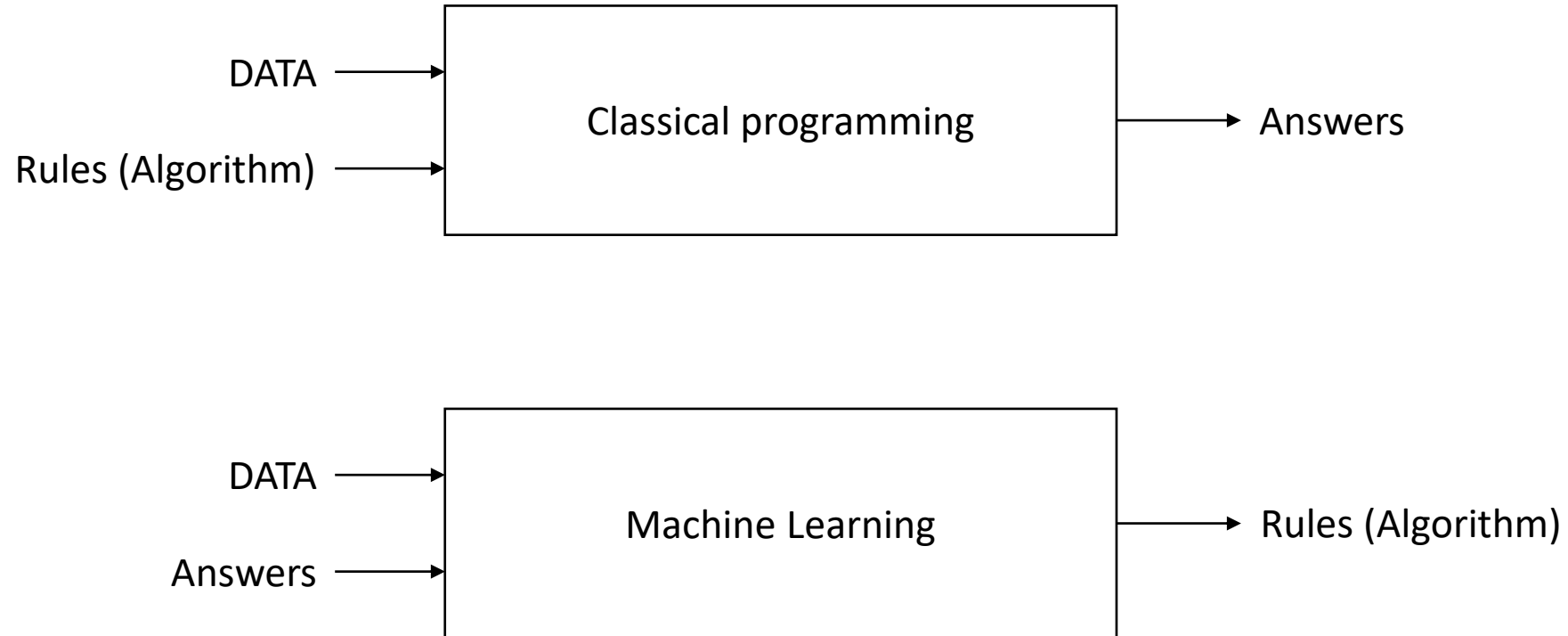
**GIVEN**

**WANTED**

## 기계 학습 (Machine Learning)

---

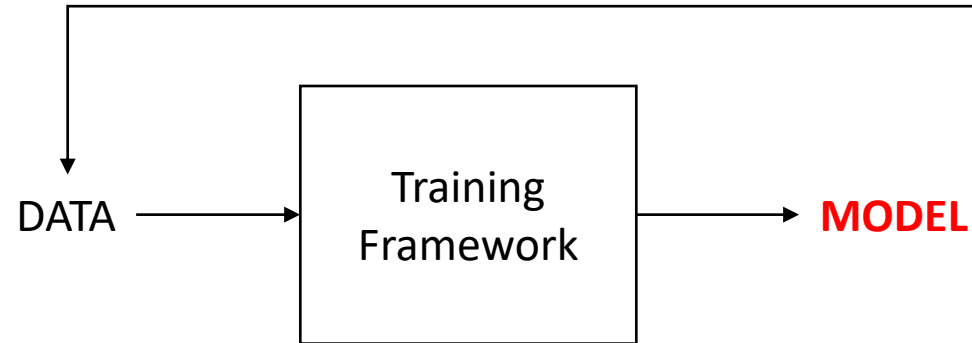
- 일반 프로그래밍 방식과 기계 학습
  - 일반 프로그래밍 방식 : 데이터와 룰을 이용해 프로그래밍하여 정해진 답(결과값)을 도출
  - 기계 학습 : 데이터와 결과(기대)값으로 최적의 룰을 도출



# 기계 학습 (Machine Learning)

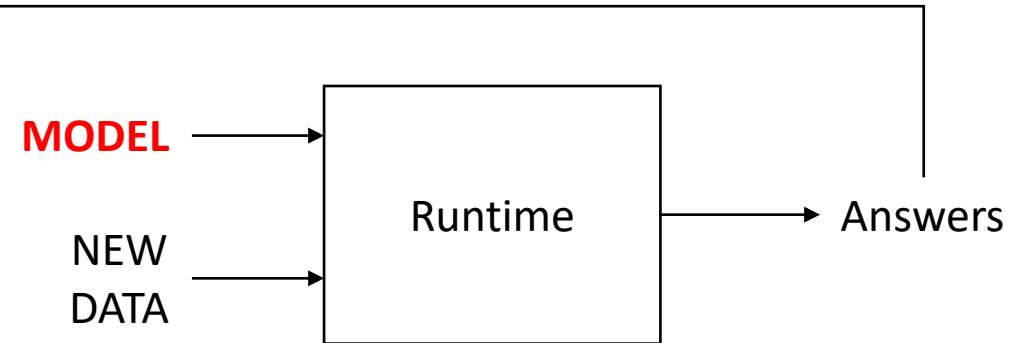
- 머신러닝 모델개발 과 모델을 통한 결과값 추론하기

## Machine Learning



Data Engineer  
Data Scientist / AI Developer  
Domain Expert

## Inferencing



Application Developer  
Infrastructure Expert  
Operations

# 기계 학습 예시

- Play Tennis 문제
  - 어떤 사람이 테니스를 치는 날의 기상 상황을 조사한 데이터
  - 학습 데이터 (Training Data)

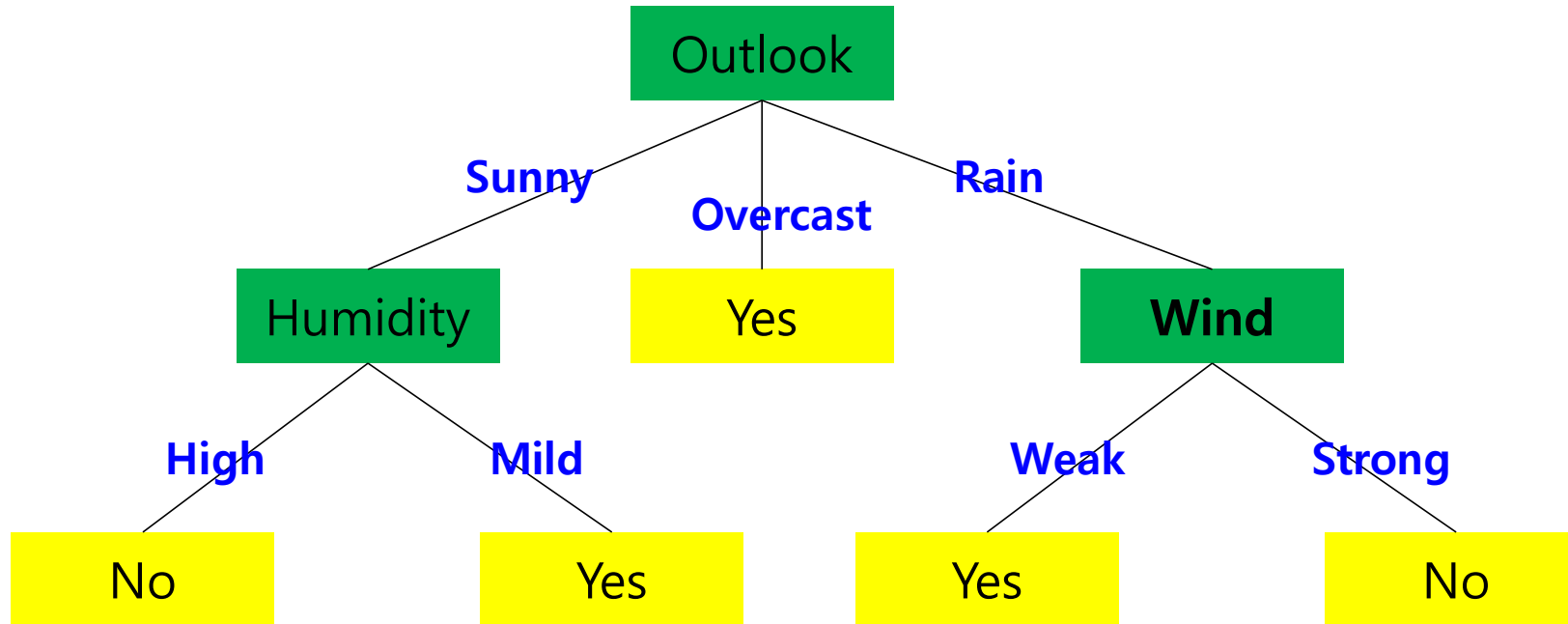
Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

(출처: Machine Learning, Tom Mitchell, 1995)

- 테니스를 치는 날은 ?
- '흐리고 적당한 온도에 습도는 높고 바람이 센 날' 테니스를 칠까?

# 기계 학습 예시

- Play Tennis 문제



Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Sunny	Hot	Mild	Weak	?
Rain	Hot	High	Weak	?

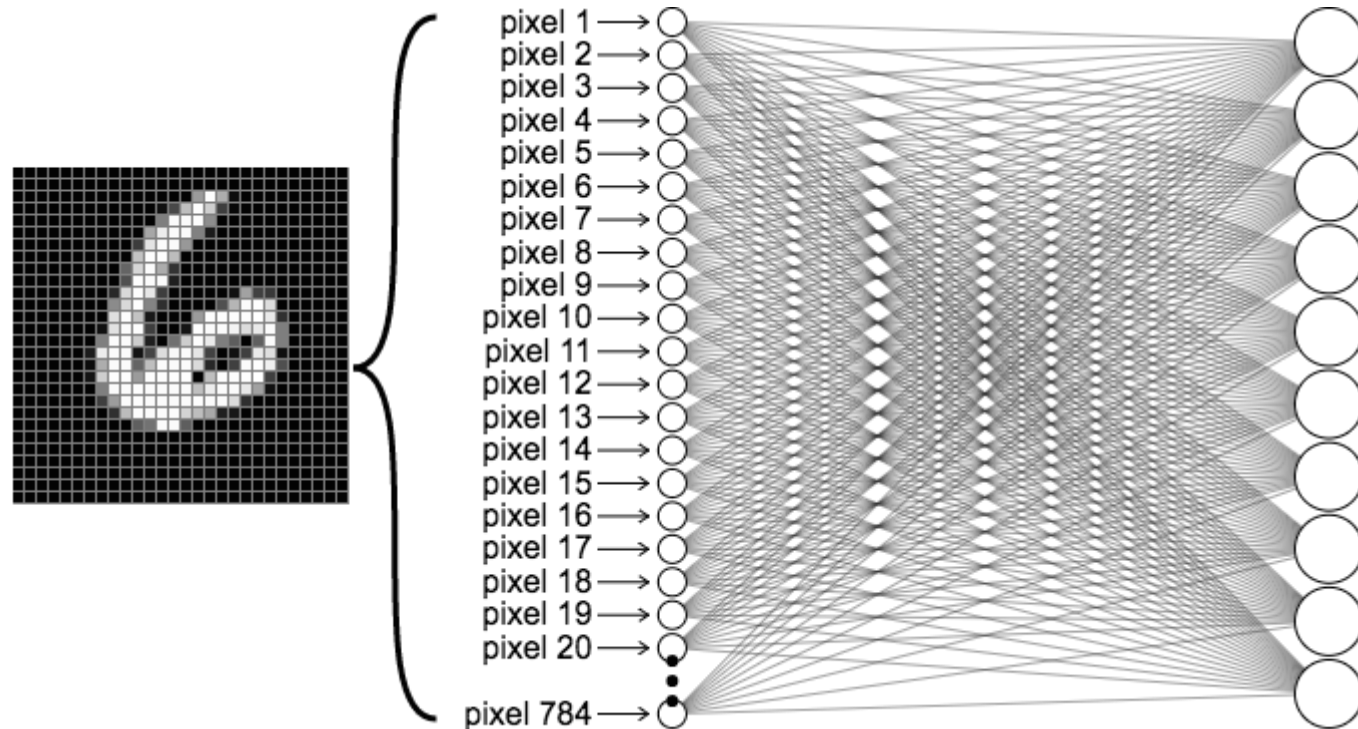
## 기계 학습 예시

- 필기문자 인식
  - 일반 프로그래밍 방식 : 직접 만든 규칙이나 휴리스틱(heuristics)
    - 복잡하고 낮은 성능
  - 기계학습 방법 : 자동으로 분류 규칙이나 프로그램 생성
    - 괄목할 만한 성능



# 기계 학습 예시

- 필기문자 인식
  - MNIST (Modified National Institute of Standards and Technology) database
    - 인공신경망 실습의 print(Hello World) 라고 할 수 있는, 전형적인 예제에 사용되는 데이터 셋
    - Training 55000장의 이미지, Validation 5000장의 이미지, Test 용으로 10000장의 이미지
    - 각 이미지는 가로 28픽셀, 세로 28픽셀의 정사각형 모양이며, 서로 다른 손 글씨 이미지
    - 컴퓨터에 저장되는 이미지는 픽셀(pixel) 단위로 표현되며 (검은 부분 0, 흰 부분 255)







# 기계 학습의 종류

- 지도학습 / 비지도학습 / 준지도학습 / 강화학습

- 지도학습 (supervised learning)

- 입력(문제)-출력(답)의 데이터들로 부터 새로운 입력에 대한 출력을 결정할 수 있는 패턴 추출

- 비지도학습 (unsupervised learning)

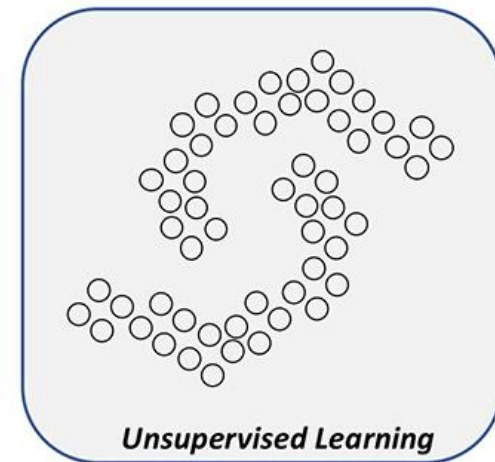
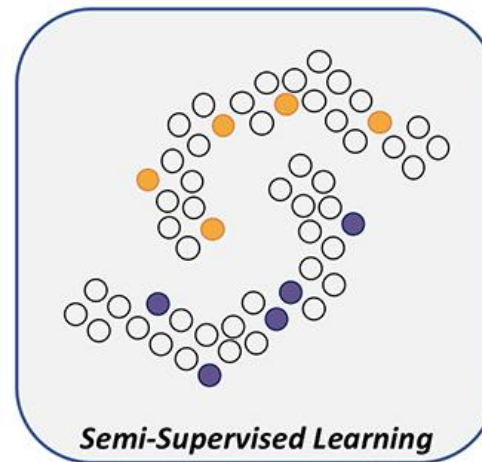
- 출력에 대한 정보가 없는 데이터로 부터 필요한 패턴 추출

- 준지도학습 (semi-supervised learning)

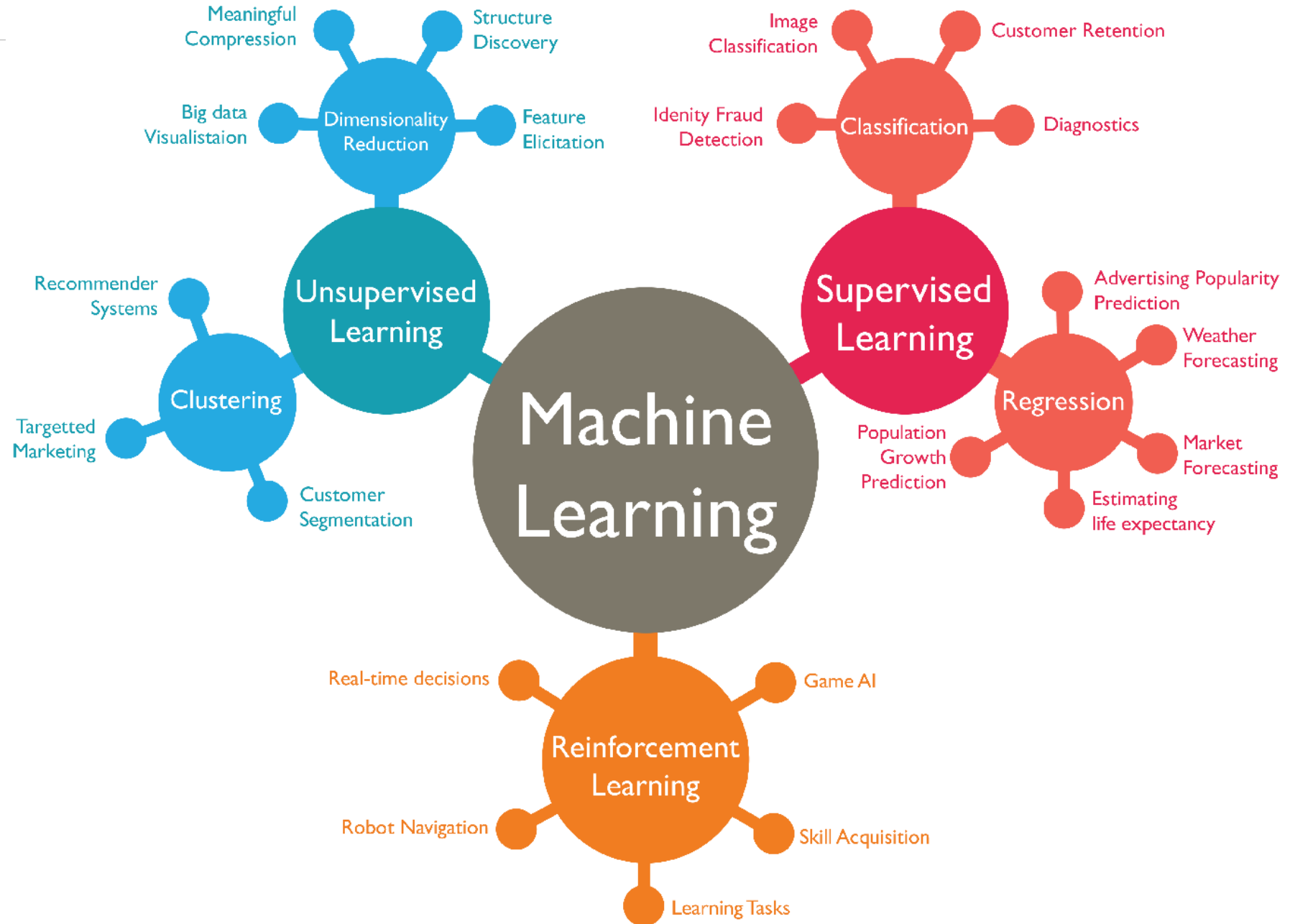
- 일부 학습 데이터만 출력 값이 주어진 상태에서 일반화한 패턴 추출

- 강화학습 (reinforcement learning)

- 출력에 대한 정확한 정보를 제공하지는 않지만, 평가정보(reward)는 주어지는 문제에 대해 각 상태에서의 행동(action)을 결정



# 기계 학습의 종류



## 기계 학습의 종류

---

- 지도학습 (Supervised Learning)
  - 지도학습은 주어진 입력 으로부터 출력 값을 예측하고자 할 때 사용
  - 입력과 데이터를 사용해 모델을 학습 시킨 후 새로운 입력 데이터에 대해 정확한 출력을 예측하는 것이 목표
  - 데이터 구축을 위해 많은 시간과 노력 (돈)이 필요하지만 높은 성능을 기대할 수 있음
- 분류 (classification) 와 회귀 (regression)
  - 분류는 범주형의 입력 데이터를 미리 정의된 여러 개의 클래스 중 하나로 예측
  - 분류는 클래스의 개수가 2개인 이진 분류 (binary classification)와 3개 이상인 다중 분류 (multi-class classification) 로 나눌 수 있음
  - 회귀는 연속적인 데이터 (숫자)를 예측하는 것으로 어떤 사람의 나이, 농작물의 수확량, 주식 가격 등 출력 값이 연속성을 가짐

## 기계 학습의 종류

- 지도학습 (Supervised Learning) – classification

기본 분류기 (Base classifier)	앙상블 분류기 (Ensemble classifier)
의사 결정 트리 (Decision tree)	Bagging, Random Forest
규칙 기반 (Rule-base method)	Boosting, AdaBoost, ...
인접 이웃 (Nearest-neighbor)	Stacking
신경망 (Neural Networks)	
심화신경망 (Deep Learning)	
베이지안 (Naiive Bayes)	
지지도 벡터 머신 (Support Vector Machine)	

## 지도 학습 - 앙상블 분류기

---

- 앙상블 학습

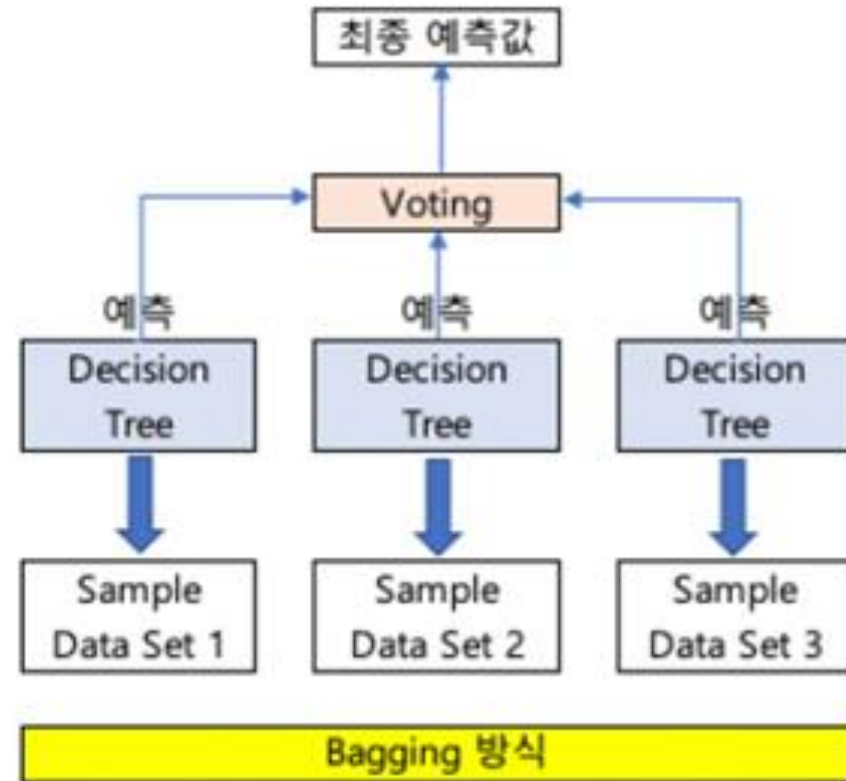
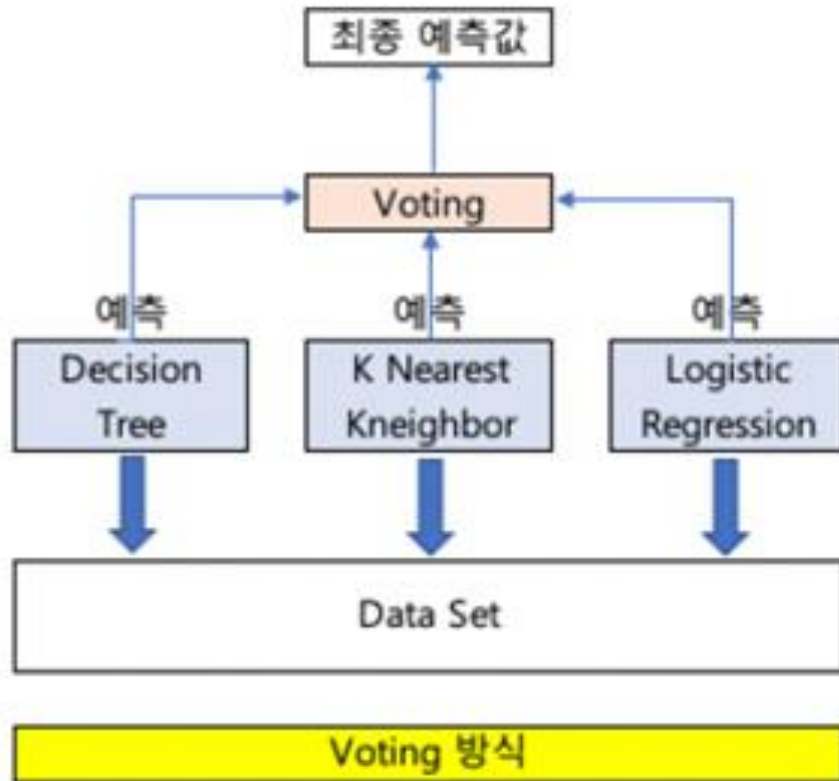
- 여러 개의 분류기(Classifier)를 생성하고 그 예측을 결합함으로써 보다 정확한 최종 예측을 도출하는 기법
- 쉽고 편하면서도 강력한 성능을 보유하고 있는 것이 바로 앙상블 학습의 특징

- **보팅(Voting), 배깅(Bagging), 부스팅(Boosting)**

- 보팅과 배깅은 여러 개의 분류기가 투표를 통해 최종 예측 결과를 결정하는 방식
- 보팅 : 일반적으로 서로 다른 알고리즘을 가진 분류기를 결합
- 배깅 : 각각의 분류기가 모두 같은 유형의 알고리즘 기반이지만, 데이터 샘플링을 서로 다르게 가져가면서 학습을 수행해 보팅을 수행. 대표적인 배깅 방식에는 랜덤 포레스트 (random forest)가 있음
- 부스팅 : 여러 개의 분류기가 순차적으로 학습을 수행하되, 앞에서 학습한 분류기가 예측이 틀린 데이터에 대해서는 올바르게 예측할 수 있도록 다음 분류기에게는 가중치(weight)를 부여하면서 학습과 예측을 진행

## 지도 학습 - 앙상블 분류기

- 보팅(Voting)과 배깅(Bagging)



# 지도 학습 - 앙상블 분류기

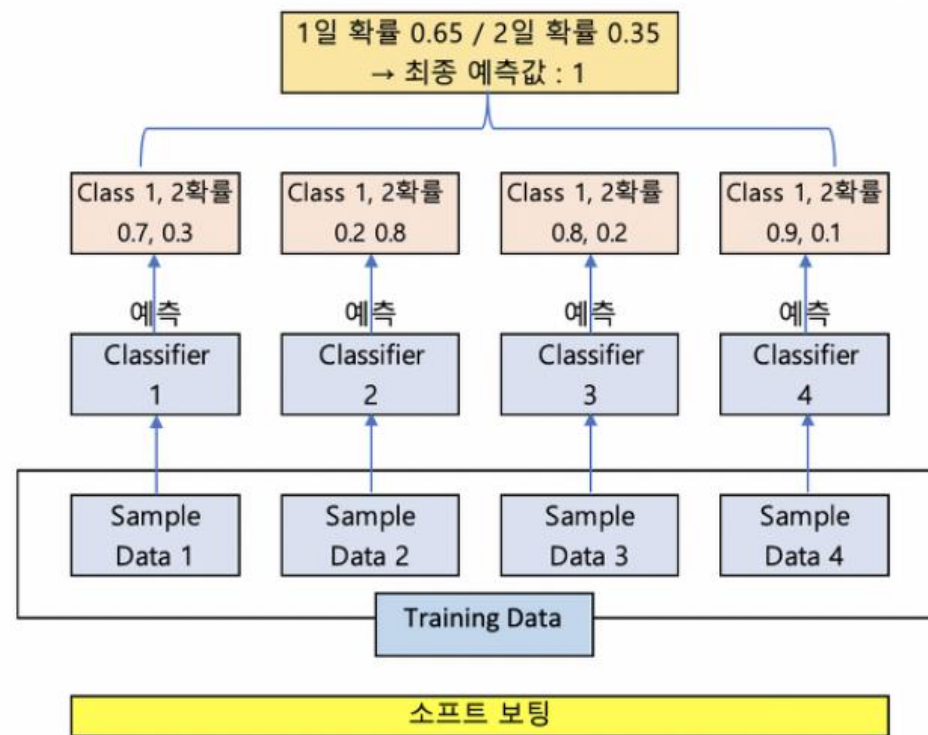
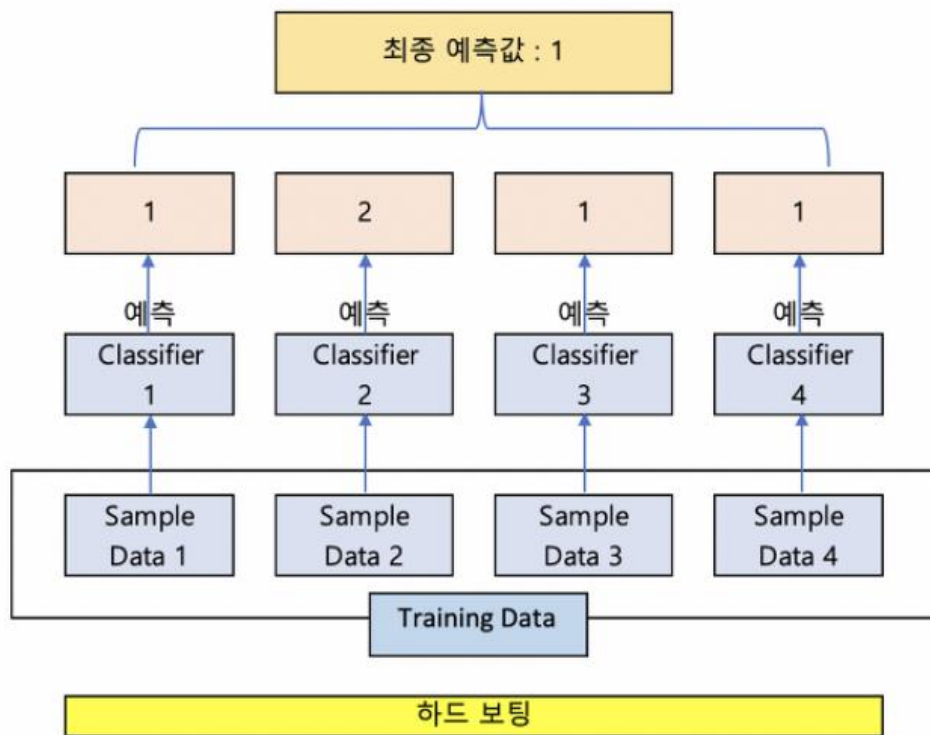
- 보팅(Voting)

- 하드 보팅 (Hard Voting)과 소프트 보팅 (Soft Voting)

- 하드 보팅을 이용한 분류(Classification)는 다수결 원칙과 비슷

- 소프트 보팅은 분류기들의 레이블 값 결정 확률을 모두 더하고 이를 평균해서

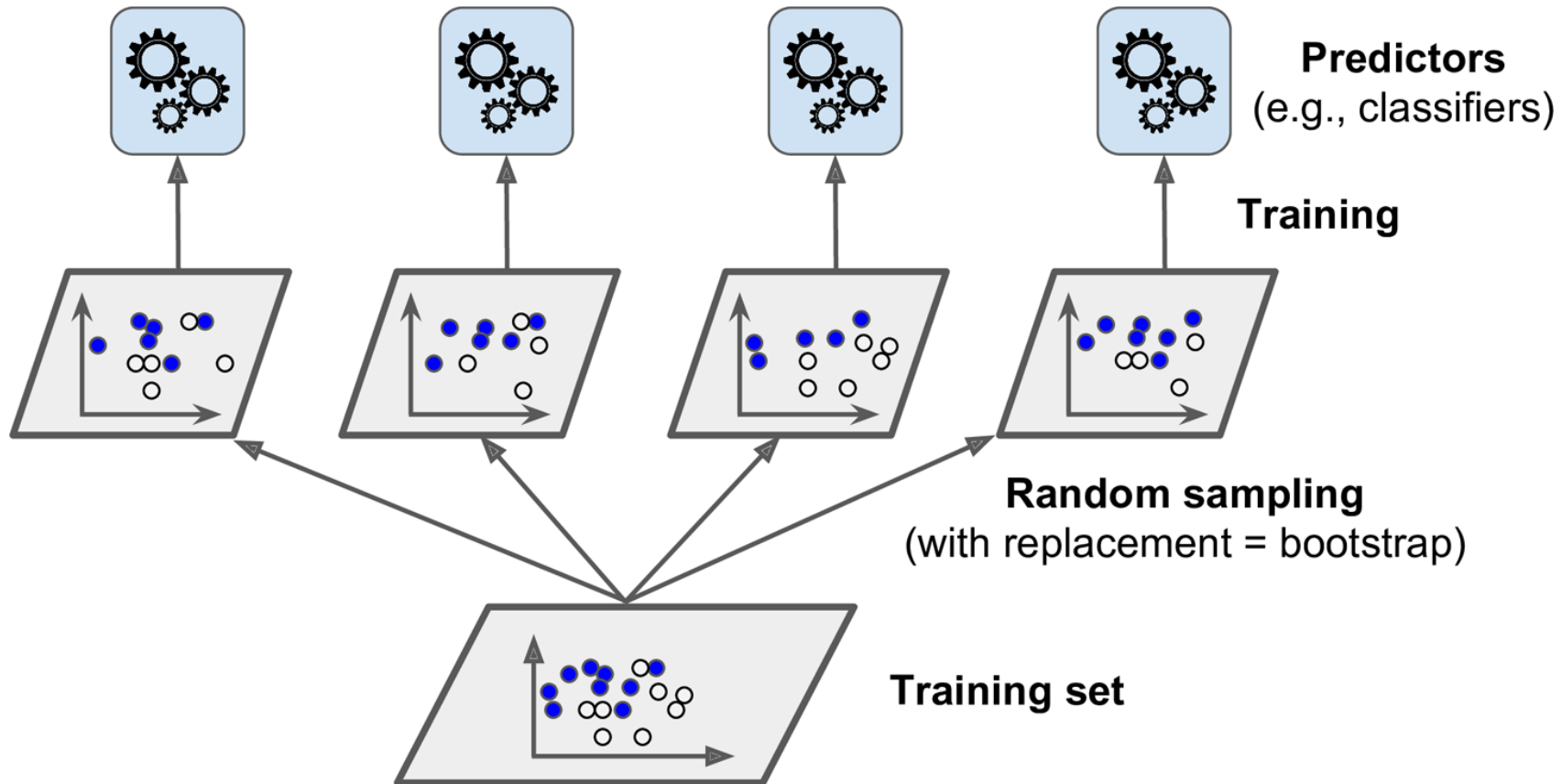
이들 중 확률이 가장 높은 레이블 값을 최종 보팅. 일반적으로 소프트 보팅이 보팅 방법으로 적용





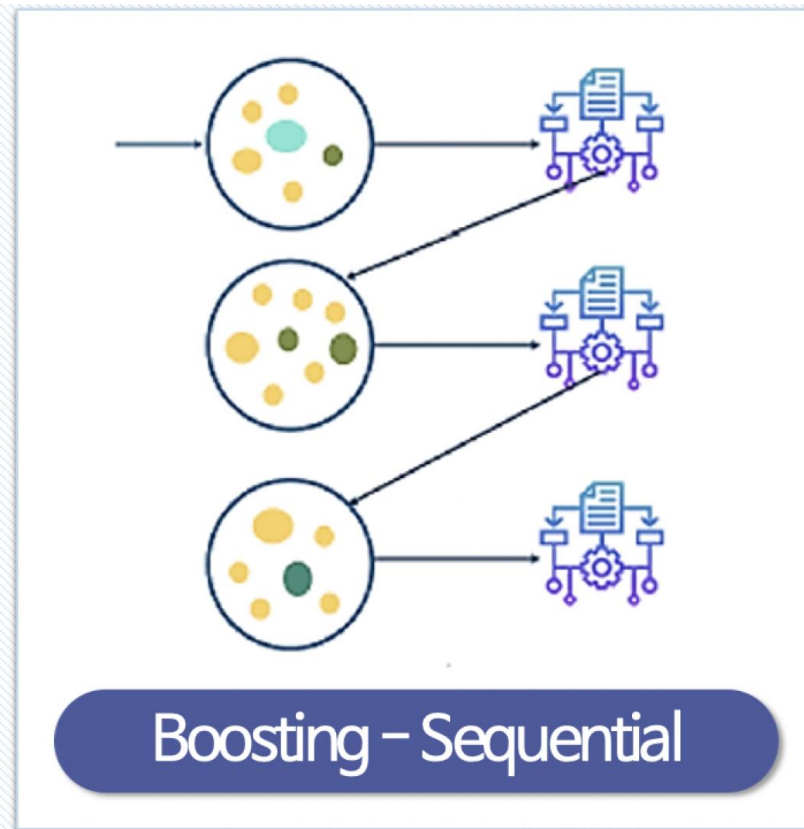
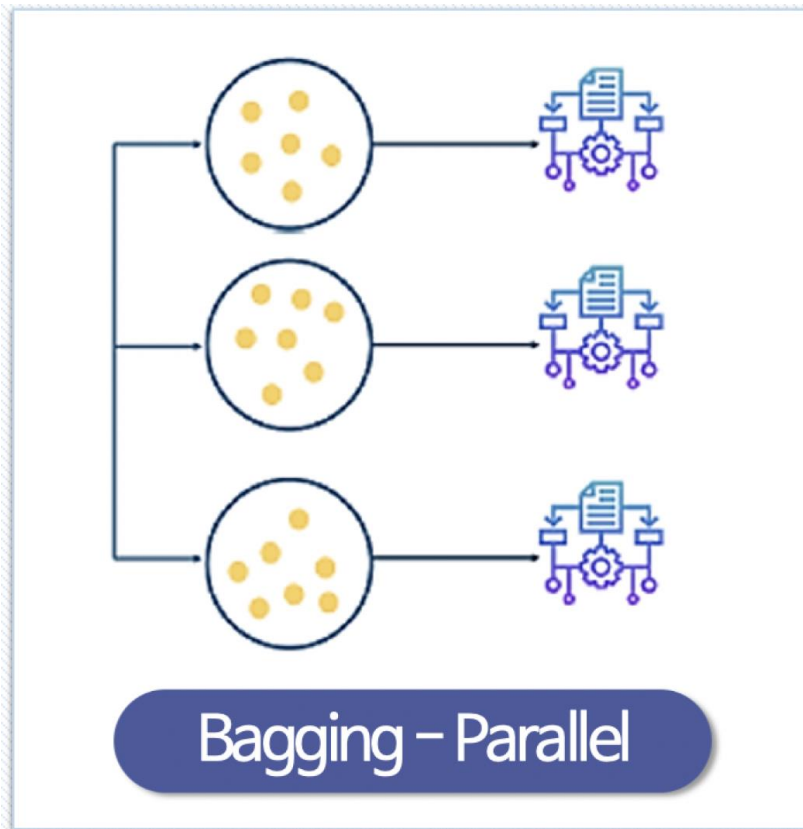
## 지도 학습 - 앙상블 분류기

- 배깅(Bagging)
  - 같은 모델에서 여러 개의 분류기를 만들어서 보팅으로 최종 결정하는 알고리즘
  - 부트스트래핑(Bootstrapping) 분할 방식 : 각 분류기에 할당되는 데이터는 원본 데이터를 샘플링해서 추출



## 지도 학습 - 앙상블 분류기

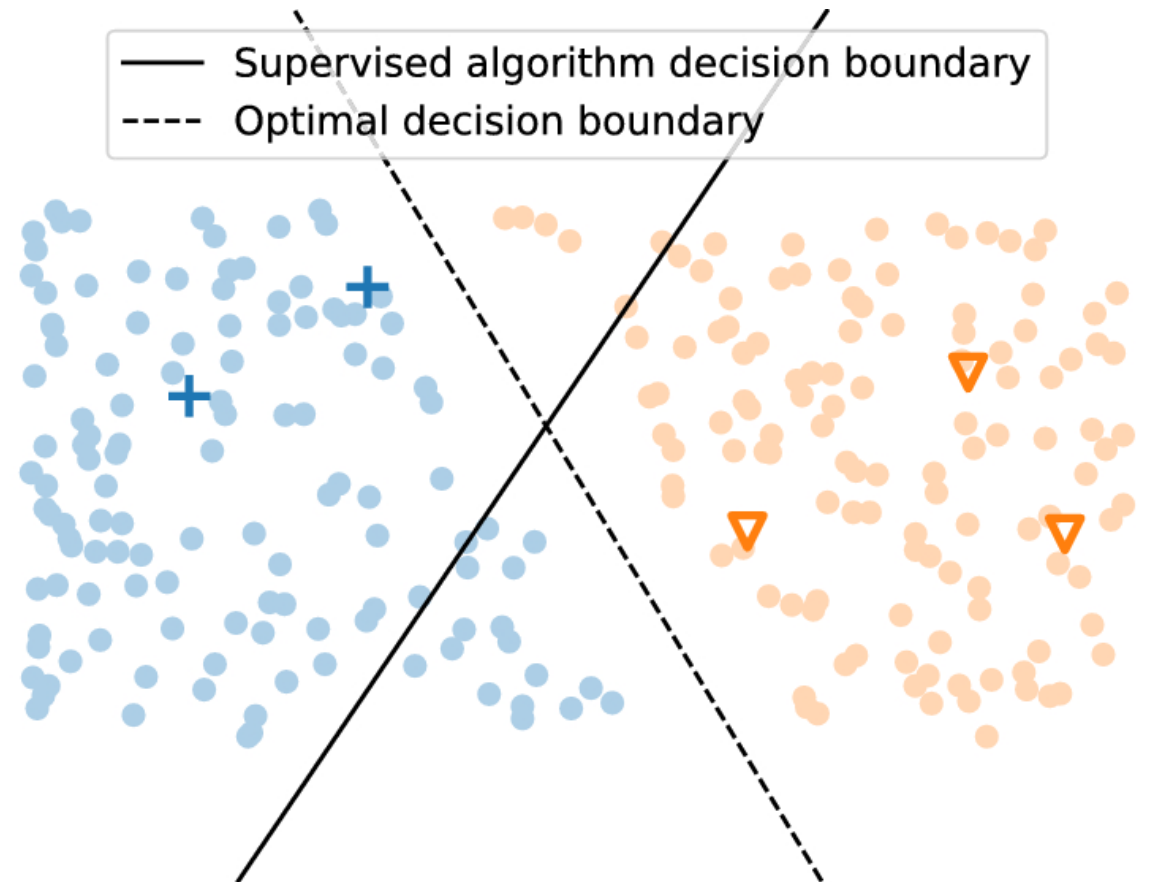
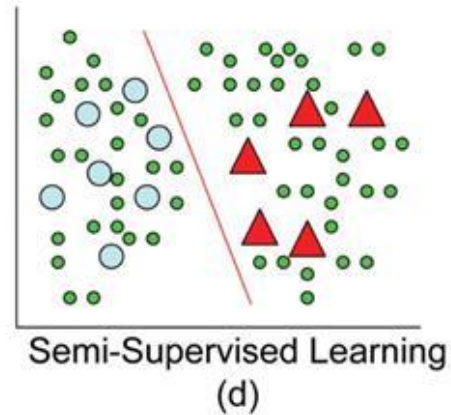
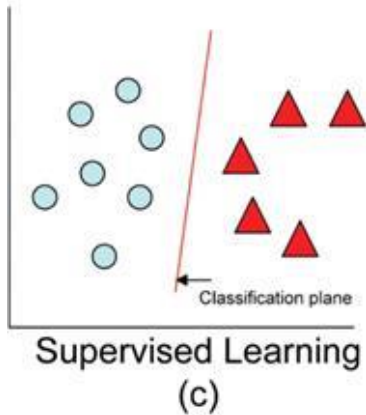
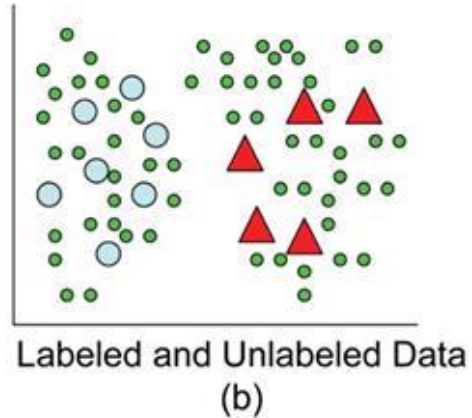
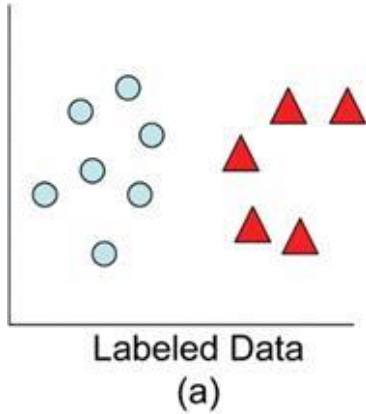
- 부스팅(Boosting)
  - 여러 개의 분류기가 순차적으로 학습을 수행하되, 앞에서 학습한 분류기가 예측이 틀린 데이터에 대해서는 올바르게 예측할 수 있도록 다음 분류기에게는 가중치(weight)를 부여하면서 학습과 예측을 진행
  - 계속해서 분류기에게 가중치를 부스팅하면서 학습을 진행하기에 부스팅 방식이라고 부름
  - 예측 성능이 뛰어나 앙상블 학습을 주도



# 기계 학습의 종류

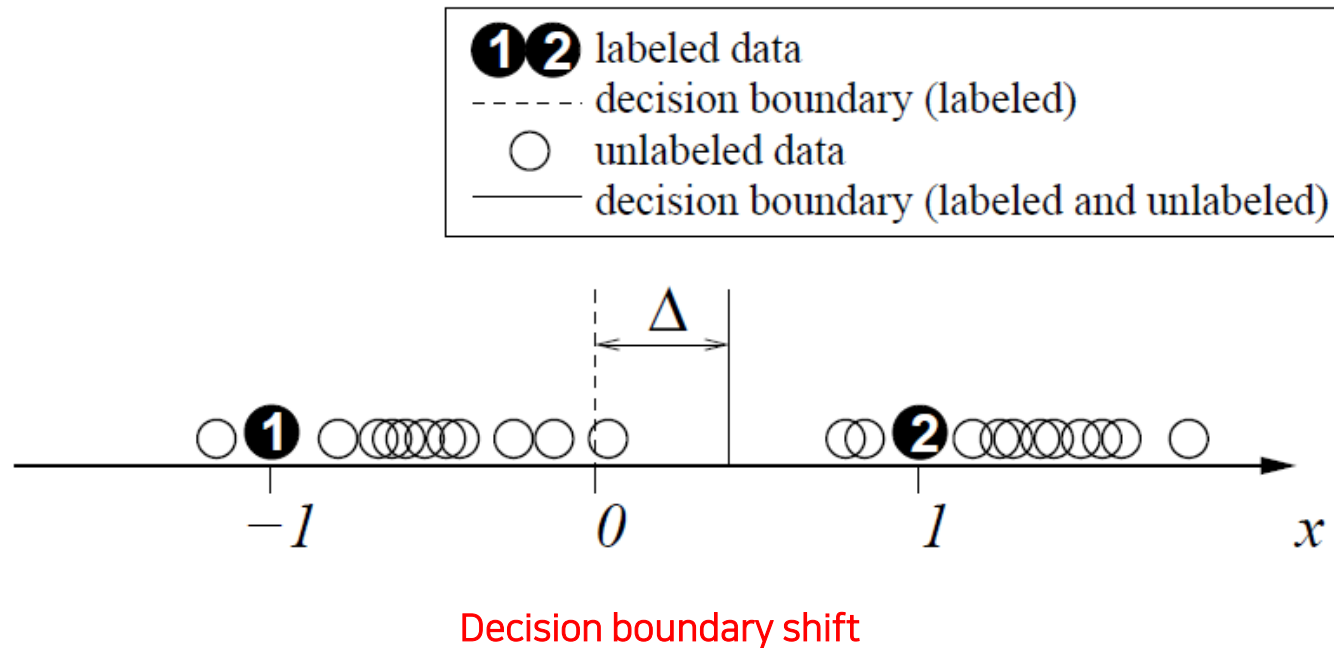
- 준지도학습 (Semi-supervised Learning)

- 우리 주변에는 레이블이 없는 데이터가 대부분이며, 이를 사용해 분류기의 성능을 향상시키는 것을 목표
- 필요성 : 정답 데이터를 수집하는 '데이터 레이블링' 작업에 소요되는 많은 자원과 비용 때문에 등장
- 목적: 모델 성능을 높이기 위해 정답이 있는 데이터와 정답이 없는 데이터를 함께 사용



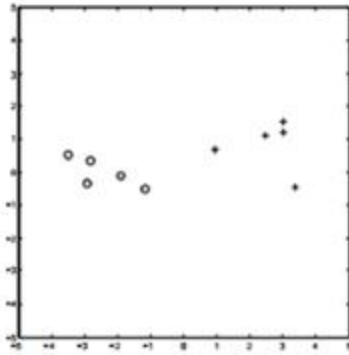
# 기계 학습의 종류

- 준지도 학습 (Semi-supervised Learning)
  - 준지도 학습이 항상 도움이 되는가?
    - Unfortunately, this is not the case, yet. (Ben David et al. (2008) and Singl et al. (2008).)
  - 준지도 학습이 오히려 성능이 저하시킬 수 있음

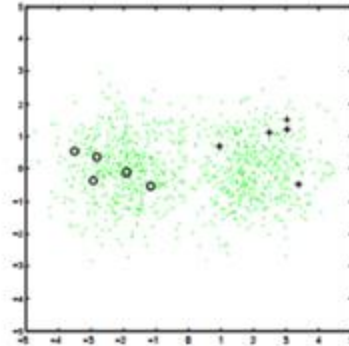


# 기계 학습의 종류

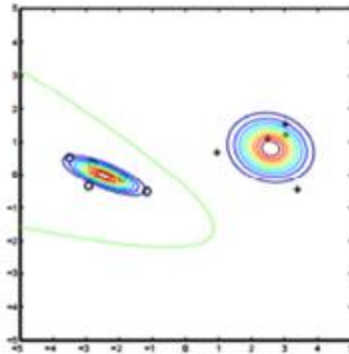
- 준지도 학습 (Semi-supervised Learning)
  - 준지도 학습이 항상 도움이 되는가?
    - Unfortunately, this is not the case, yet. (Ben David et al. (2008) and Singl et al. (2008).)
  - 준지도 학습이 오히려 성능이 저하시킬 수 있음



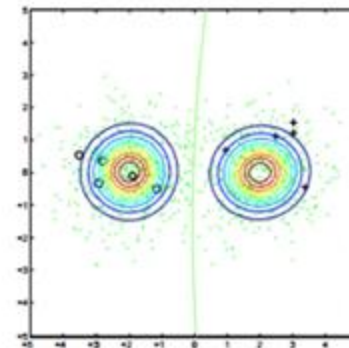
(a) 레이블링된 데이터



(b) 레이블링된 데이터와 레이블링이 안 된 데이터

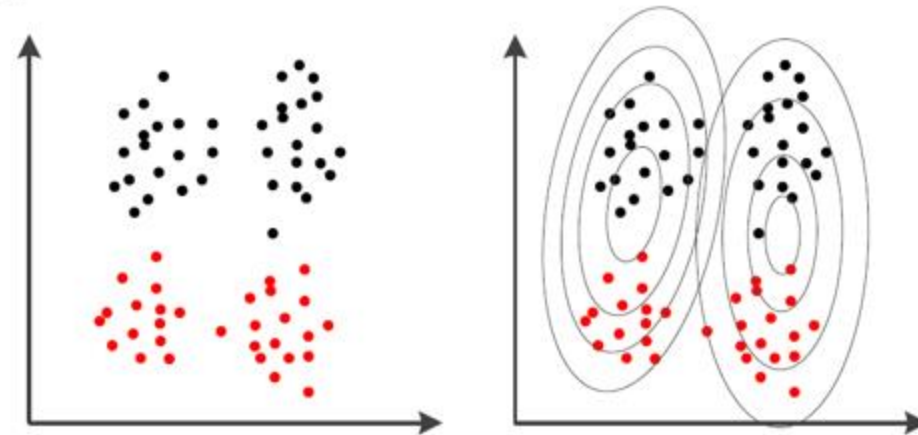


(c) 지도 학습으로 추정된 모델



(d) 준지도 학습으로 추정된 모델

준지도 학습이 도움이 되는 경우

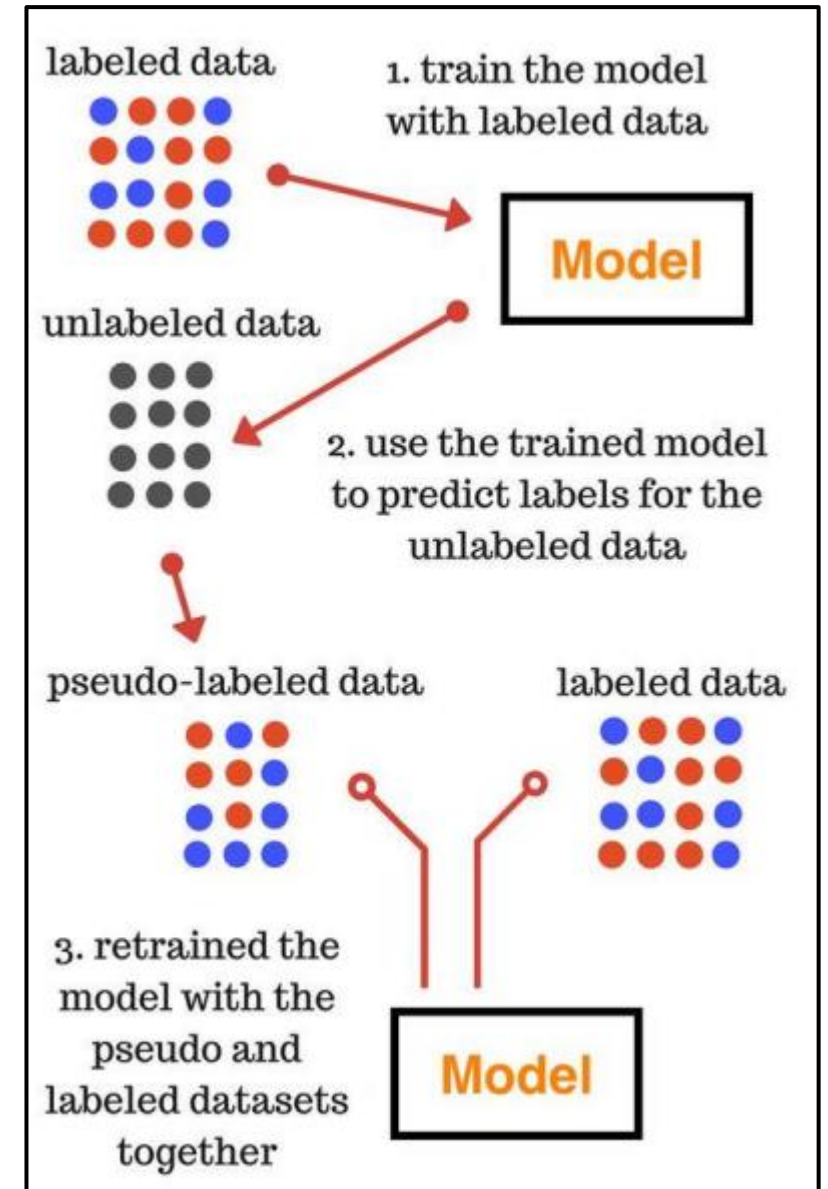


준지도 학습이 도움이 되지 않는 경우

# 준지도학습의 대표적인 방법

## Self-Training

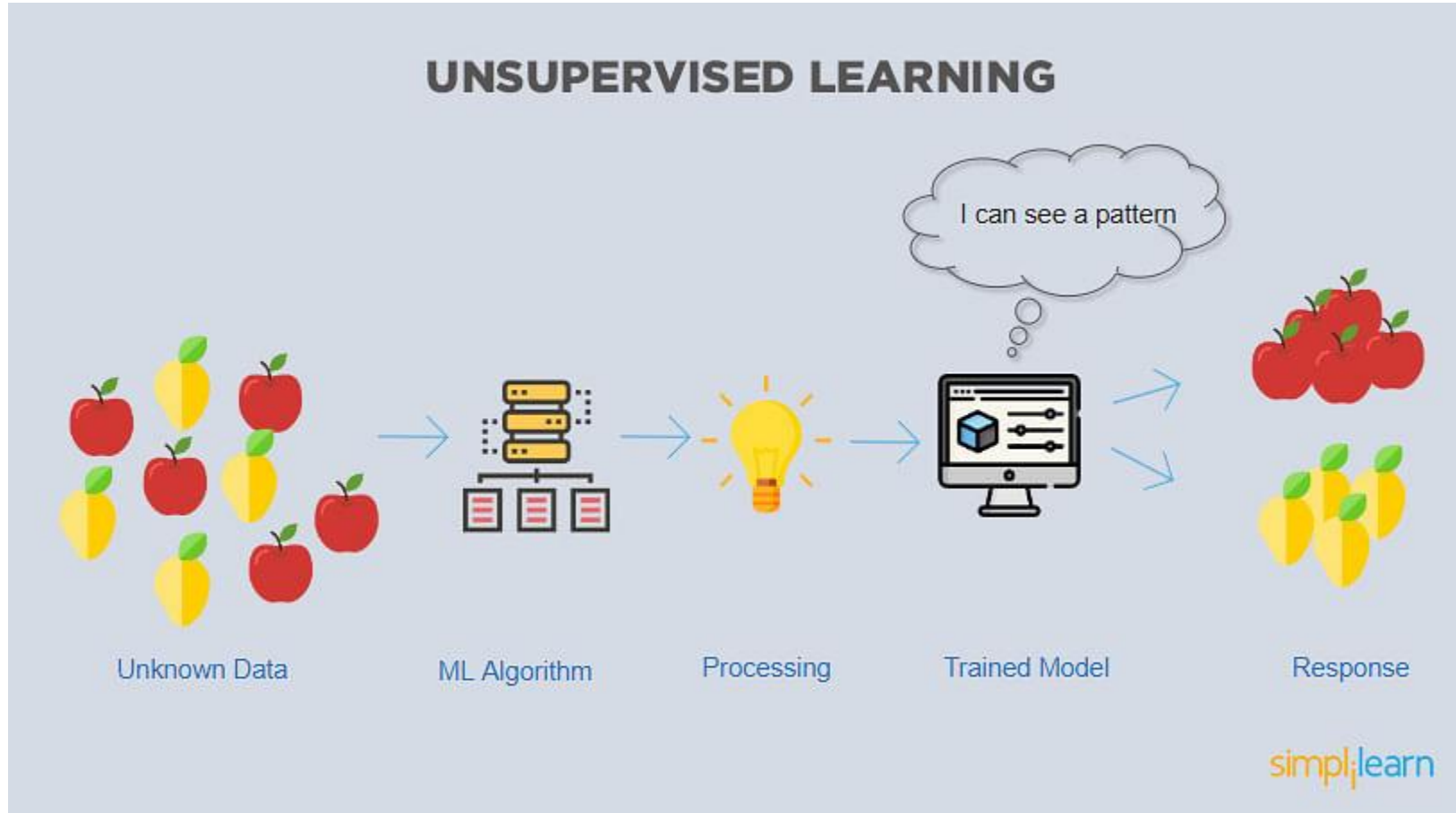
- Assumption in the self-Training
  - One's own high confidence predictions are correct
- Algorithm
  - Train  $f$  from  $(x_l, y_l)$
  - Predict on  $x \in x_u$
  - Add  $\{x, f(x)\}$  to labeled data
  - Repeat
- Variations in self-training
  - Add a few most confident  $\{x, f(x)\}$  to labeled data
  - Add all  $\{x, f(x)\}$  to labeled data
  - Add all  $\{x, f(x)\}$  to labeled data with weights by confidence





# 기계 학습의 종류

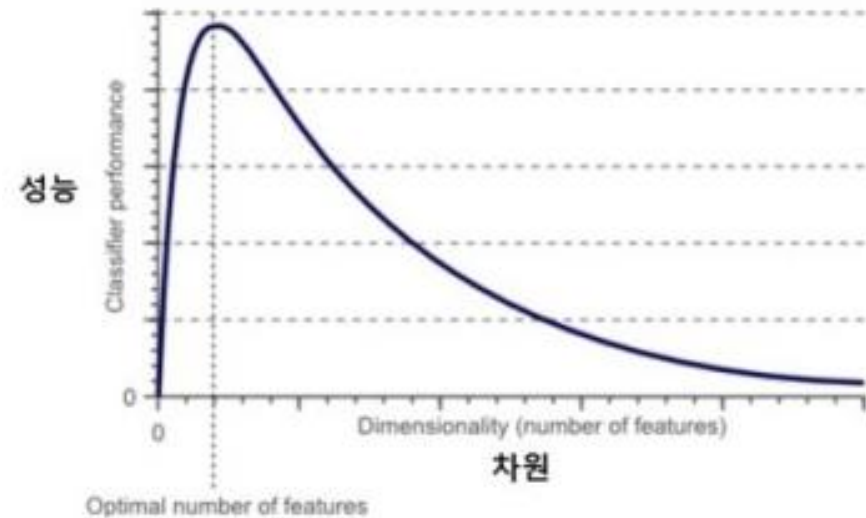
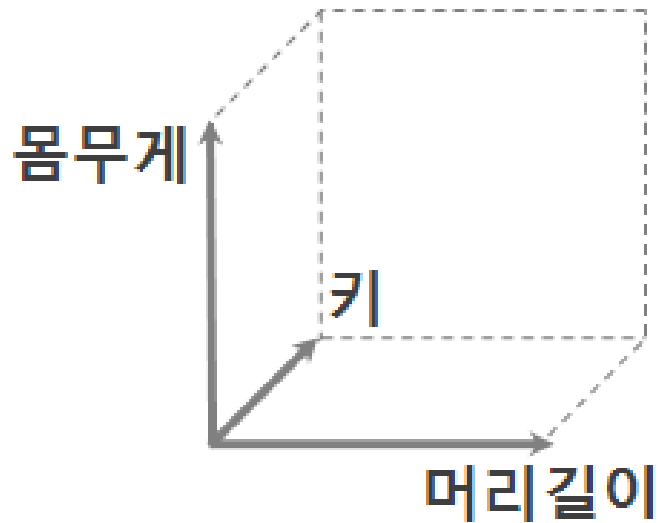
- 비지도학습 (Unsupervised Learning)
  - 지도학습과 달리 training data의 정답(혹은 label)이 없는 데이터가 주어지는 학습방법



# 비지도학습

- 차원 축소 (Dimension Reduction)

- 고차원의 데이터를 정보의 손실을 최소화하면서 저차원으로 변환하는 것
- 목적
  - 2, 3차원으로 변환해 시각화하면 직관적 데이터 분석 가능
  - 차원의 저주(curse of dimensionality) 문제 완화
    - 차원이 늘어남에 따라 필요한 데이터의 양이 급격하게 증가
    - 차원이 늘어나면 해당 공간을 설명하기에 데이터가 부족하기 때문에 과적합 (overfitting) 문제 발생





# 비지도학습

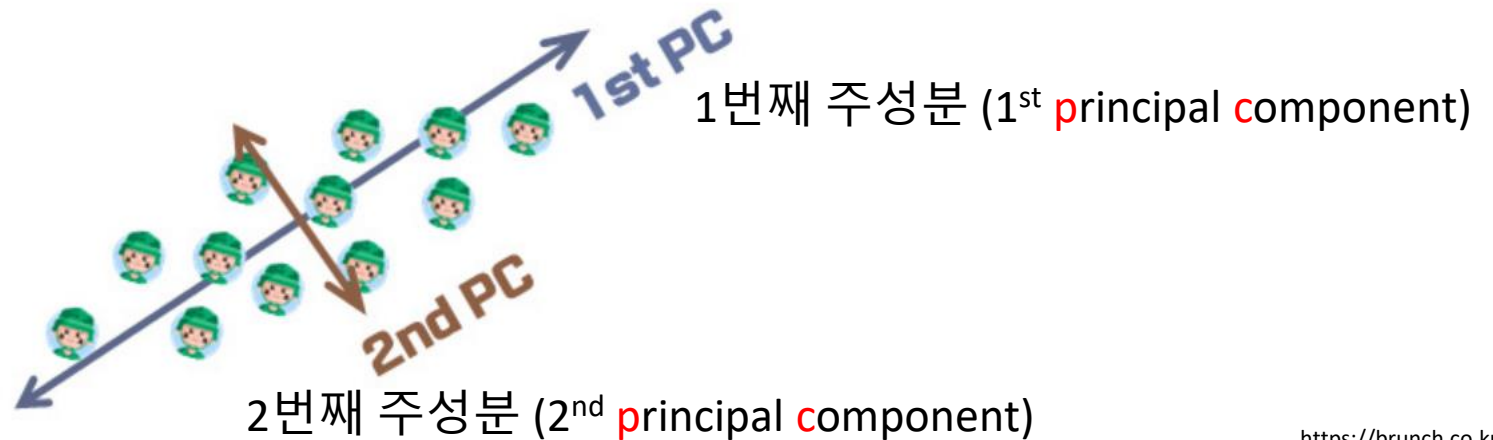
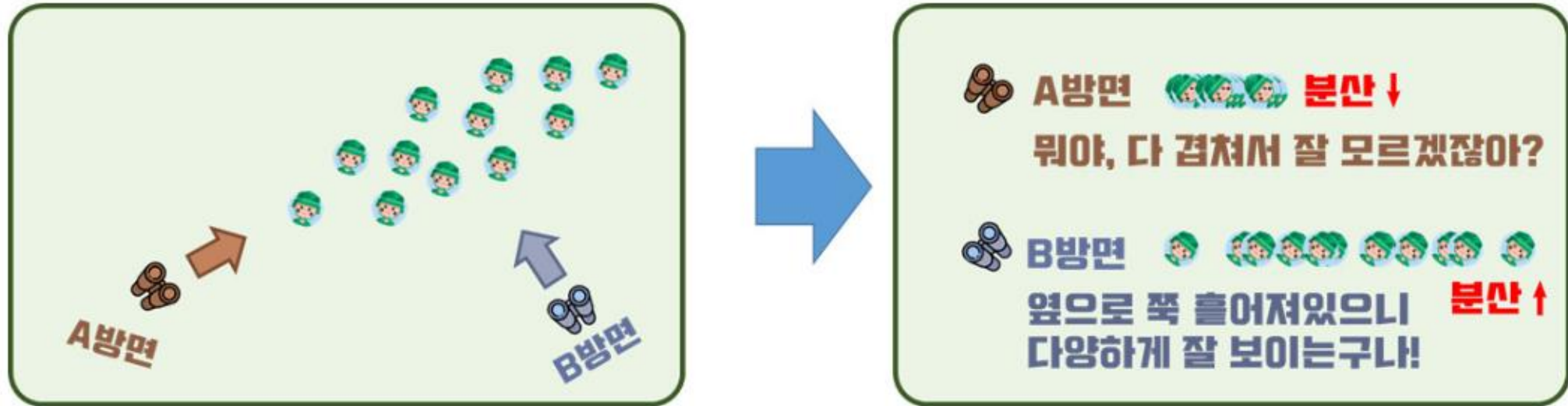
- 차원 축소 (Dimension Reduction)
  - 특성 선택 : 특성의 수를 줄이는 방법
  - 특성 추출 : 기존의 특성들 에서 새로운 특성을 만드는 방법



$$\text{이과} = (\text{수학1} + \text{물리학}) / 2$$

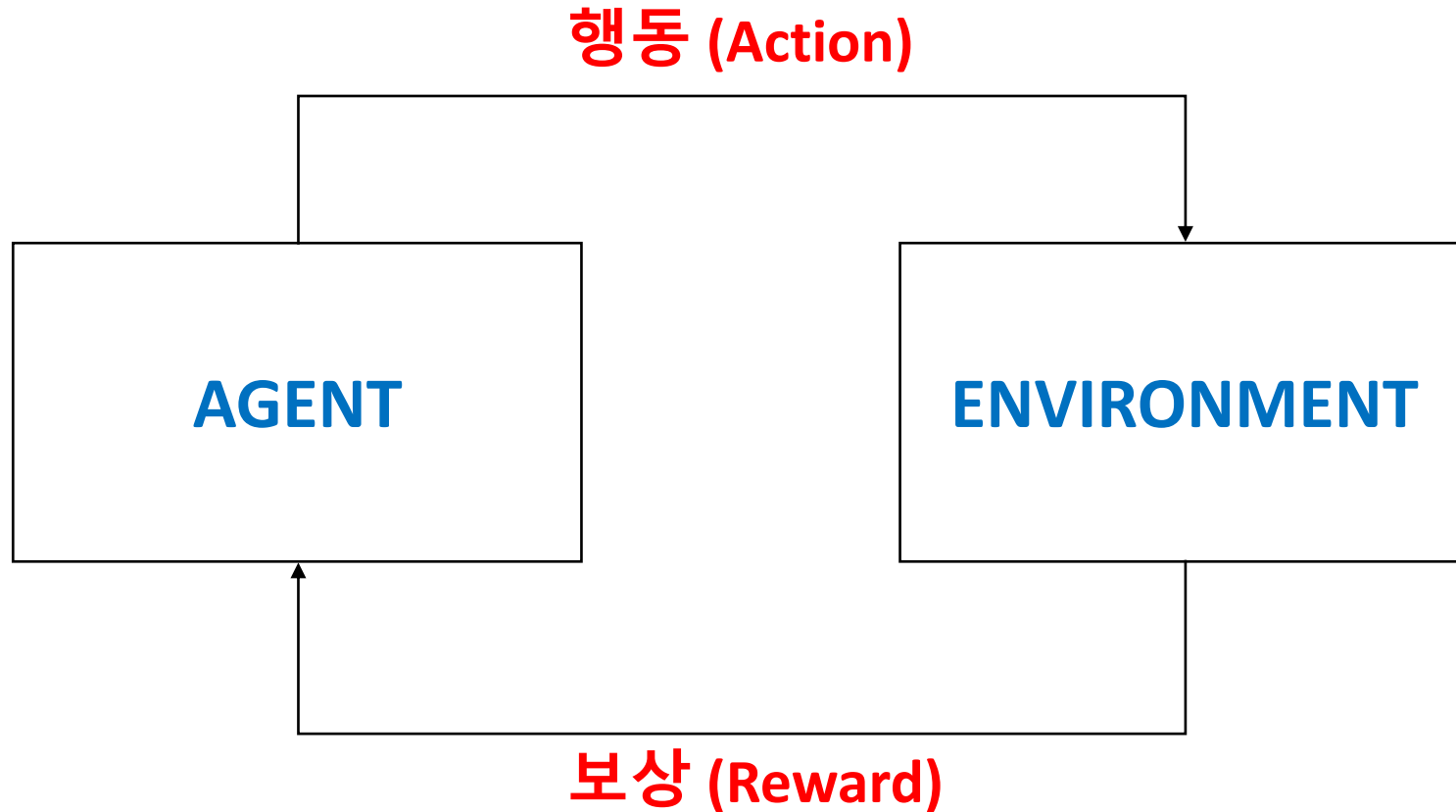
# 비지도학습

- 차원 축소 (Dimension Reduction)
  - 차원을 줄일 때는 자료의 특성을 가장 다양하게 표현할 수 있는 방향을 찾는 것이 중요



## 기계 학습의 종류

- 강화학습 (Reinforcement Learning)
  - 어떠한 환경에서 어떠한 행동을 했을 때 그것이 잘 된 행동인지 잘못된 행동인지를 나중에 판단하고 보상 (또는 벌칙)을 줌으로써 반복을 통해 스스로 학습
  - 에이전트가 환경을 관찰하고, 행동을 실행 → 피드백을 통해 자동으로 학습하고 성능 향상시킴



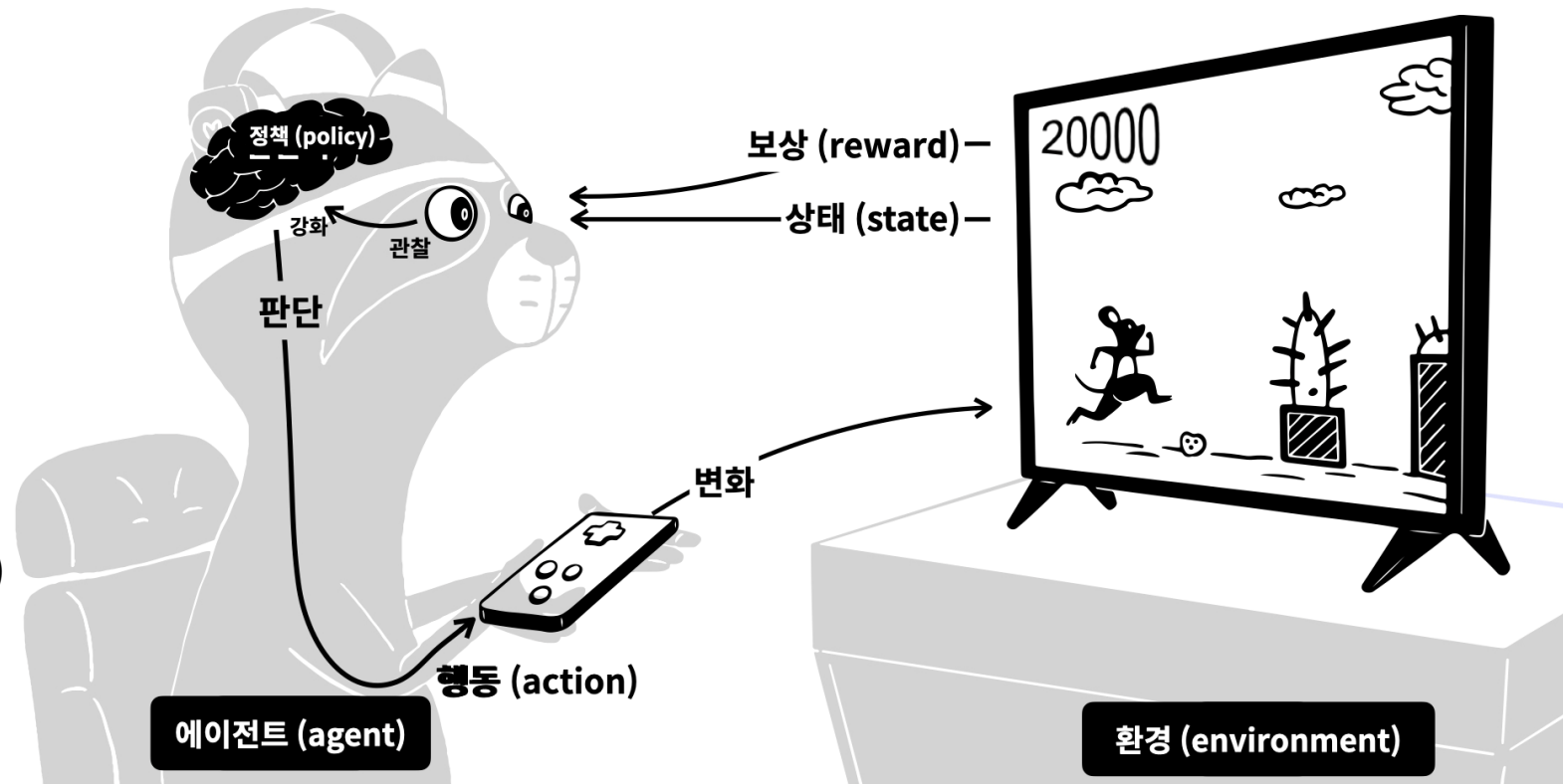
# 기계 학습의 종류

- 강화학습 (Reinforcement Learning)

- 지도학습이 배움을 통해서 학습을 하는 것이라면 강화학습은 경험을 통해서 학습
- 행동의 결과가 자신에게 유리한 것이었다면 상을 받고, 불리한 것이었다면 벌을 받음
- 이 과정을 반복 하다보면 더 많은 보상을 받을 수 있는 더 좋은 답을 찾을 수 있음

- 게임

- 게임 : 환경 (environment)
- 게이머 : 에이전트 (agent)
- 게임화면 : 상태 (state)
- 게이머의 조작 : 행동 (action)
- 상과 벌 : 보상 (reward)
- 게이머의 판단력 : 정책 (policy)





## 배치학습과 온라인학습

---

- 배치 학습 (Batch Learning)
  - 전체 데이터를 모두 사용해 training data를 학습시키는 방법으로, 시간과 자원을 많이 소모
  - 일반적으로 오프라인 환경에서 수행되므로 오프라인 학습 (offline learning)이라고 함
  - 학습은 추론 전에 일어나고, 제품에 모델이 탑재되면 더 이상의 학습 없이 사용만 함
  - 새로운 데이터가 등장하며, 모델을 재학습하고 싶은 경우, 이전 데이터에 새로운 데이터를 포함한 전체 데이터를 학습시키고, 학습된 새로운 모델을 사용
  - 컴퓨팅 자원이 풍부한 경우 사용함

## 배치학습과 온라인학습

---

- 온라인 학습 (Online Learning)
  - 일반적으로 학습이 끝나 제품화가 된 모델에 대하여 미니 배치 (mini-batch)라 부르는 작은 묶음 단위의 데이터를 주입하여 모델을 학습시키는 방법
  - 미니 배치의 크기가 작기 때문에 학습 단계가 빠르고 비용이 적게 들기 때문에 모델은 데이터가 도착하는 대로 즉시 학습을 할 수 있음
  - 점진적으로 학습이 일어나기 때문에 점진적 학습 (incremental learning)이라고도 함
  - 연속적으로 데이터를 받고 빠른 변화에 스스로 적응해야 하거나 자원이 매우 한정된 환경에 적합
  - 새로운 데이터 샘플을 학습하면, 학습이 끝난 데이터는 더 이상 필요하지 않기 때문에 보관하지 않아도 되므로 저장 공간을 많이 아낄 수 있음
  - 실시간 시스템이나 메모리 부족의 경우 사용함
- 문제점
  - 나쁜 데이터가 주입되었을 때, 시스템의 성능이 점진적으로 감소할 수 있음
  - Catastrophic forgetting : 모델이 이전에 알고 있던 내용에 대한 성능이 현저하게 떨어지는 문제가 있음

# 일반화, 과소적합과 과대적합

- 일반화 예시

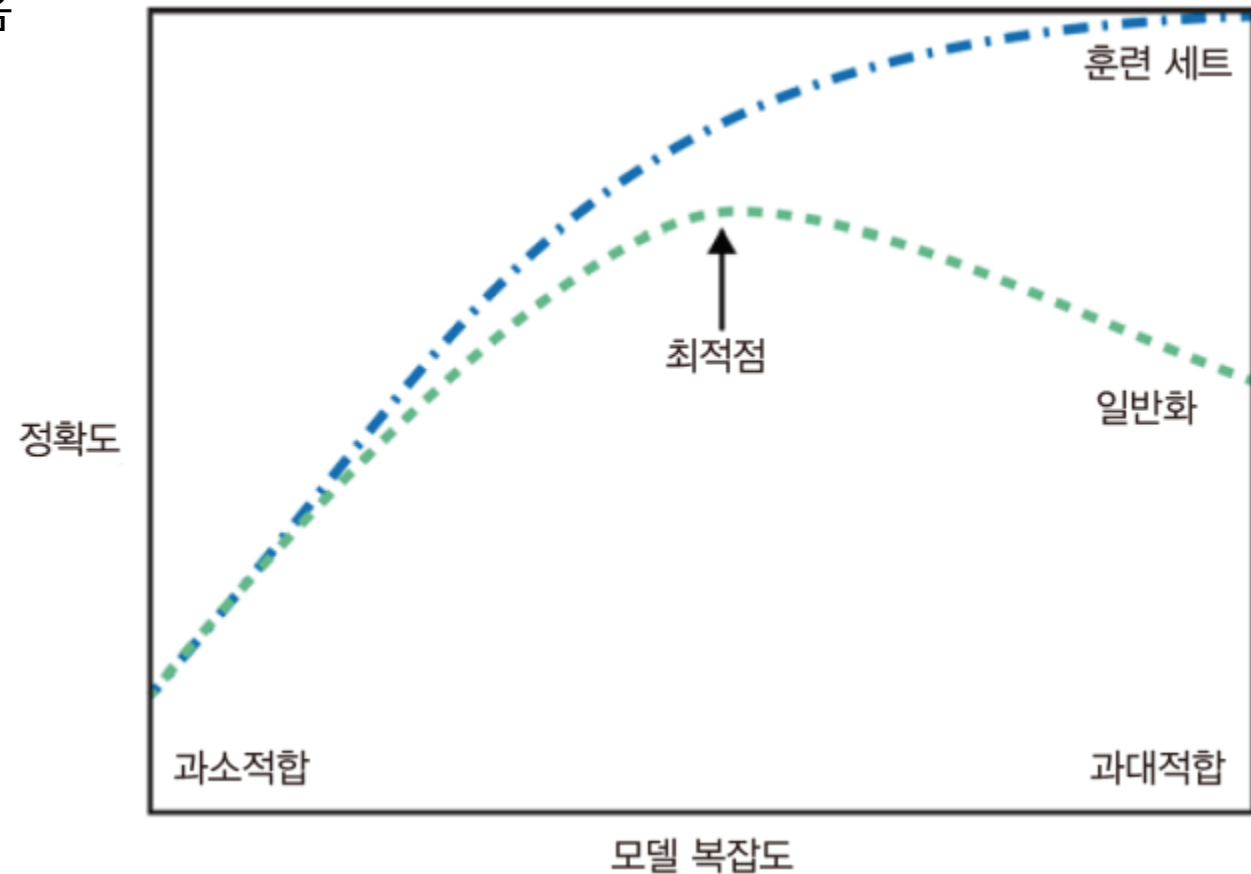
45세 이상이고 자녀가 셋 미만이며 이혼하지 않은 고객은 요트를 살 것이다.

나이	보유차량수	주택보유	자녀수	혼인상태	애완견	보트구매
66	1	yes	2	사별	no	yes
52	2	yes	3	기혼	no	yes
22	0	no	0	기혼	yes	no
25	1	no	1	미혼	no	no
44	0	no	2	이혼	yes	no
39	1	yes	2	기혼	yes	no
26	1	no	2	미혼	no	no
40	3	yes	1	기혼	yes	no
53	2	yes	2	이혼	no	yes
64	2	yes	3	이혼	no	no
58	2	yes	2	기혼	yes	yes
33	1	no	1	미혼	no	no



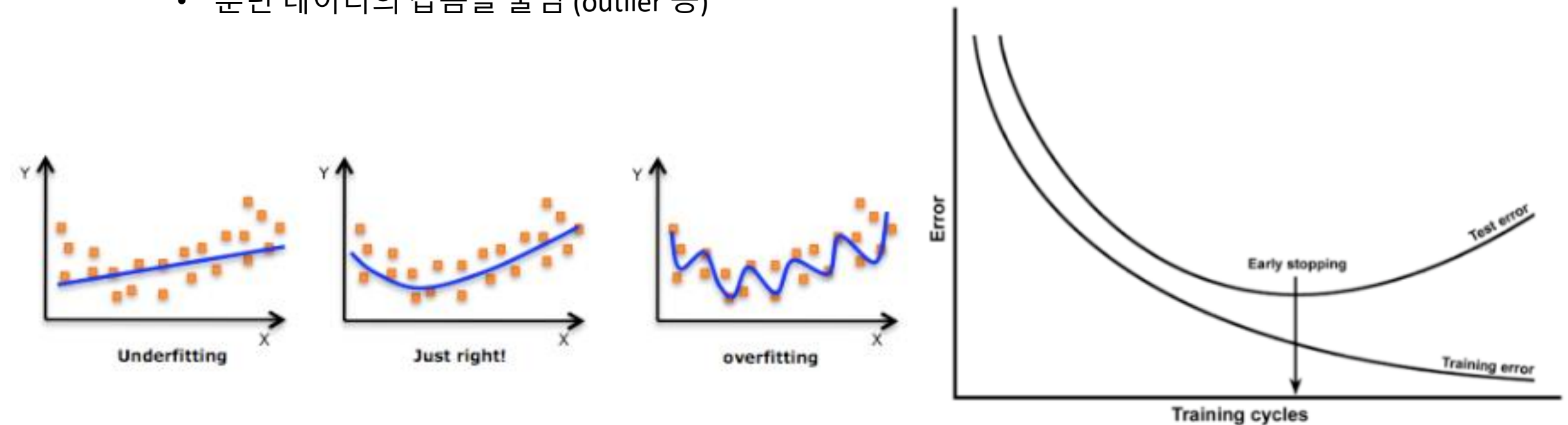
## 일반화, 과소적합과 과대적합

- 우리가 찾으려는 모델은 일반화 성능이 최대가 되는 최적점에 있는 모델
  - 모델의 복잡도는 훈련 데이터셋에 담긴 입력 데이터의 다양성과 관련
  - 요트 판매 예로 돌아가 보면, 고객 데이터를 10,000개 모아봤더니 전부 "45세 이상이고 자녀가 셋 미만이며 이혼하지 않는 고객은 요트를 사려고 한다." 라는 규칙을 만족한다면 12개만 사용할 때보다 더 좋은 규칙이라고 할 수 있음



# 일반화, 과소적합과 과대적합

- 과대적합 (Overfitting)
  - 학습데이터를 과하게 잘 학습한 것을 의미
  - 학습 데이터에 대해서는 오차가 감소하지만, 실제 데이터에 대해서는 오차가 증가하는 지점이 존재
  - 해결 방법
    - Regularization : 파라미터 수가 적은 모델을 선택하거나, 모델에 제약을 가하여 단순화
    - 훈련 데이터를 더 많이 확보
    - 훈련 데이터의 잡음을 줄임 (outlier 등)



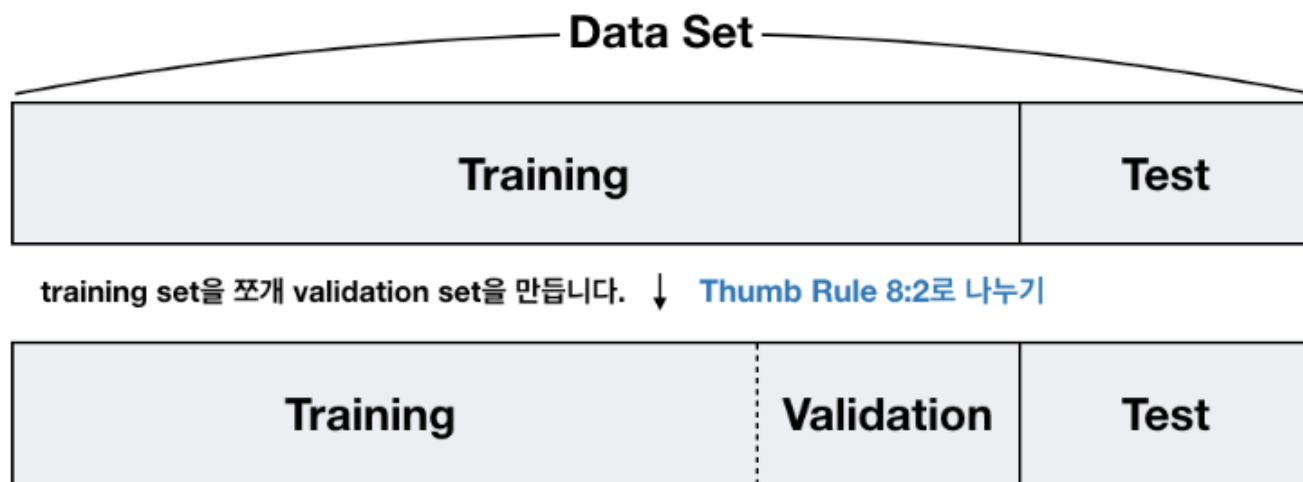
## 일반화, 과소적합과 과대적합

---

- 과소적합 (Underfitting)
  - 과대적합의 반대 개념으로, 모델이 너무 단순해서 데이터의 내재된 구조를 학습하지 못할 때 발생
  - 해결 방법
    - 파라미터가 더 많은 강력한 모델을 선택
    - 학습 알고리즘에 더 좋은 특성을 제공
    - 모델의 제약을 줄임
- 모델 복잡도와 데이터셋 크기의 관계
  - 데이터의 다양성이 클수록 더 복잡한 모델을 사용하면 좋은 성능을 얻을 수 있음
  - 일반적으로 더 큰 데이터셋 (데이터 수, 특징 수)일수록 다양성이 높기 때문에 더 복잡한 모델을 사용 가능
  - 하지만 같은 데이터를 중복하거나 비슷한 데이터를 모으는 것은 다양성 증가에 도움이 되지 않음
  - 데이터를 더 많이 수집하고 적절한 모델을 만들어 사용하면 지도학습을 사용해 높은 성능의 결과 확보 가능




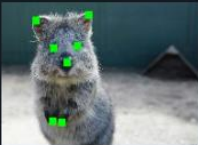
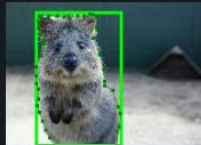









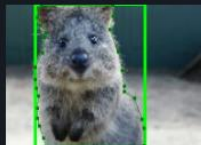


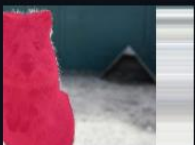
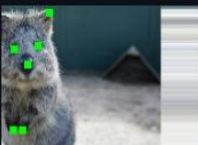
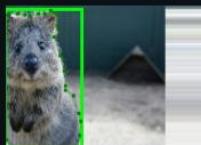





# 데이터 셋

- 러닝 모델의 설계, 학습, 테스트를 위해 확보한 데이터
  - 모델의 일반화 성능을 측정하기 위해 훈련 셋, 테스트 셋으로 구분
    - 훈련 셋으로 모델을 학습하고 테스트 셋으로 모델의 일반화 성능 측정
  - 테스트 셋을 이용해 여러 모델을 평가하면 테스트 셋에 과대적합 가능성 있음
    - 모델 선택을 위해 훈련 셋, 검증 셋, 테스트 셋으로 구분해야 함
  - 검증 셋은 교차검증 (cross-validation) 수행



## 데이터 증강 (Data Augmentation)

- 기계 학습과 인공지능에서 항상 발생하는 문제는 데이터의 부족
- 데이터 증강 (Data Augmentation)은 데이터의 양을 늘리기 위해 원본에 각종 변환을 적용하여 개수를 증강 방법
  - 보통 컴퓨터 비전 (computer vision) 분야에서 많이 사용

	Image	Heatmaps	Seg. Maps	Keypoints	Bounding Boxes, Polygons
Original Input					
Gauss. Noise + Contrast + Sharpen					
Affine					
Crop + Pad					
FlipH + Perspective					



## 데이터 증강 (Data Augmentation)



감사합니다

[kimtwan21@dongduk.ac.kr](mailto:kimtwan21@dongduk.ac.kr)

김 태 완