

캐글(Kaggle)과 함께하는 기계 학습 입문

타이타닉 생존자 예측부터 보스턴 집값 예측까지

나동빈(dongbinna@postech.ac.kr)

Pohang University of Science and Technology

1강) 기계 학습이란?

강의 수강 전에 알아보기

- 본 강의는 파이썬(python) 기초 문법을 이해하고 있는 사람에게 적절한 강의입니다.
- 기계 학습을 처음 공부하는 입문자가 들으면 좋은 강의입니다.
- 한 번에 모든 내용을 다 이해하고 넘어가기보다는, 여러 번 반복 학습 하면서 익히는 게 좋습니다.

강의 수강 전에 알아보기

- 본 강의에서는 파이썬(python) 문법이 사용됩니다.
- 파이썬을 이용한 기본적인 프로그래밍이 가능해야 합니다.
 - 사칙연산
 - 반복문(for 문법)
 - 함수 사용 방법
 - 클래스 사용 방법

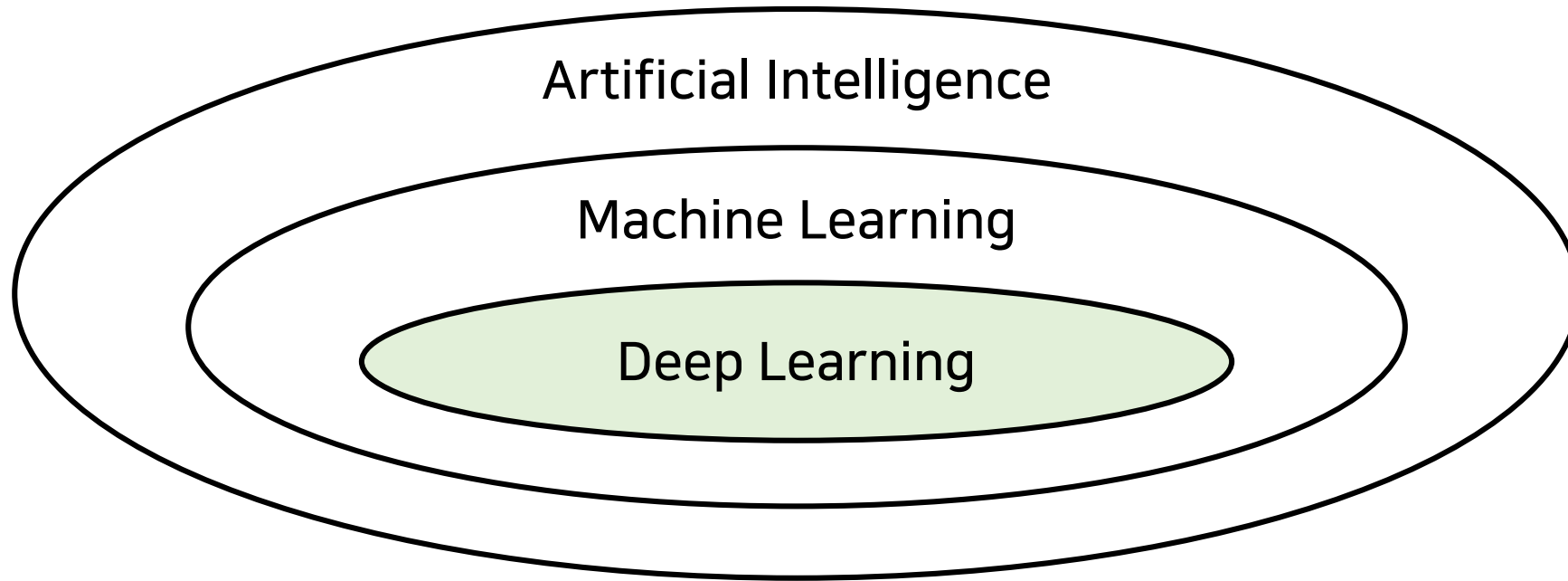


앞으로 배워 볼 내용

- 기계 학습을 배우기 위해 자주 사용되는 기초 학습용 데이터 세트가 있습니다.
- 가장 먼저, 아래의 기본적인 두 가지를 다루게 될 예정입니다.
 1. 타이타닉 생존자 예측
 2. 보스턴 집값 예측

인공지능이란?

- 인공지능이란 기계를 통해 인공적으로 구현된 지능을 의미합니다.
 - 기계 학습: 데이터를 반복적으로 학습하여 데이터에 잠재된 특징을 발견합니다.
 - 딥러닝: 깊은 인공 신경망을 활용하여 더 높은 정확도를 얻습니다.

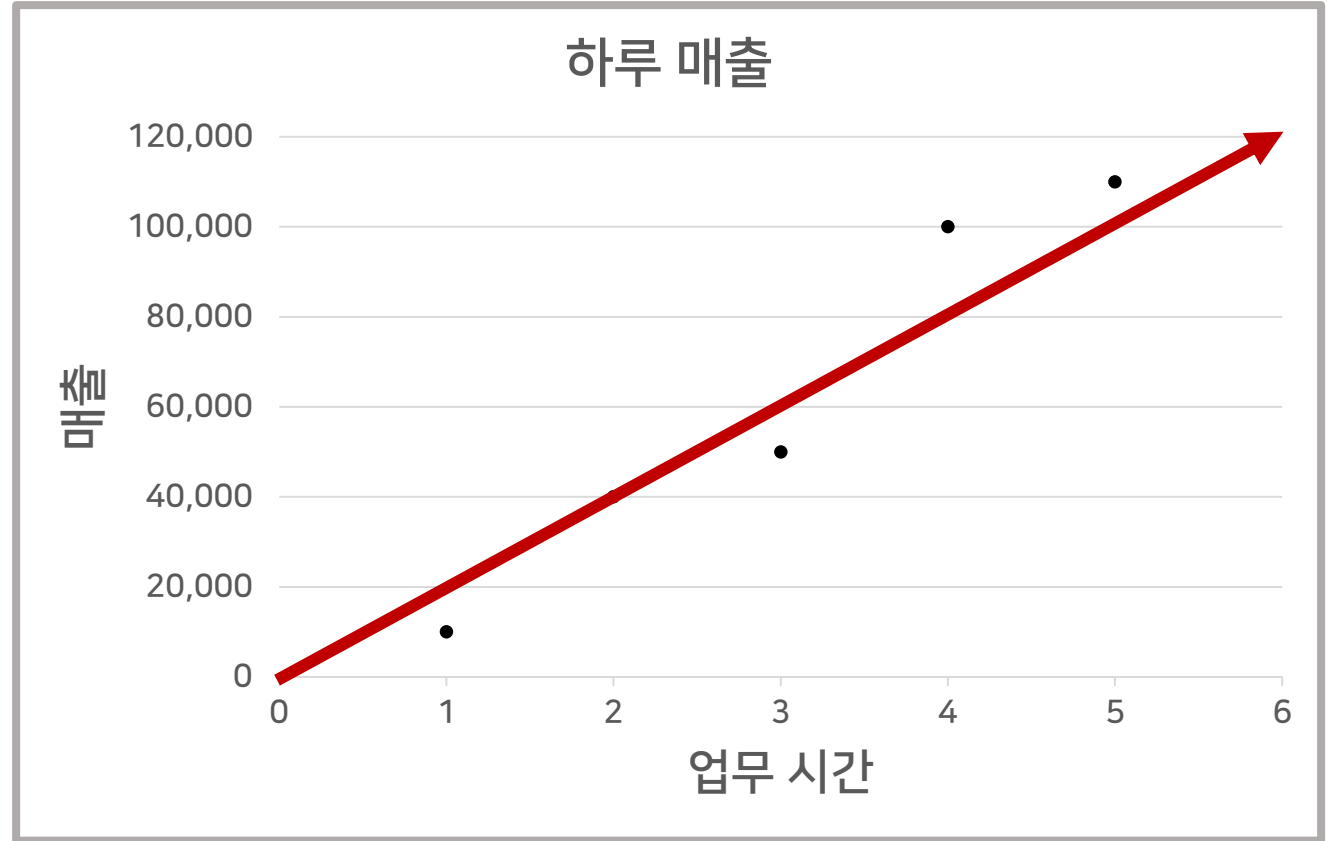


인공지능의 학습 유형 1) 지도 학습(Supervised Learning)

- 지도 학습은 명시적인 정답을 제공하면서 학습시키는 유형입니다.
 - 회귀(Regression)
 - 특정한 데이터가 주어졌을 때 결과를 연속적인 값으로 예측합니다.
 - 예시: "영어 공부를 7시간 했다면, 몇 점이 나올까요?"
 - 분류(Classification)
 - 종류에 따라서 데이터를 분류합니다.
 - 예시: "이 이미지는 고양이인가요? 강아지인가요?"

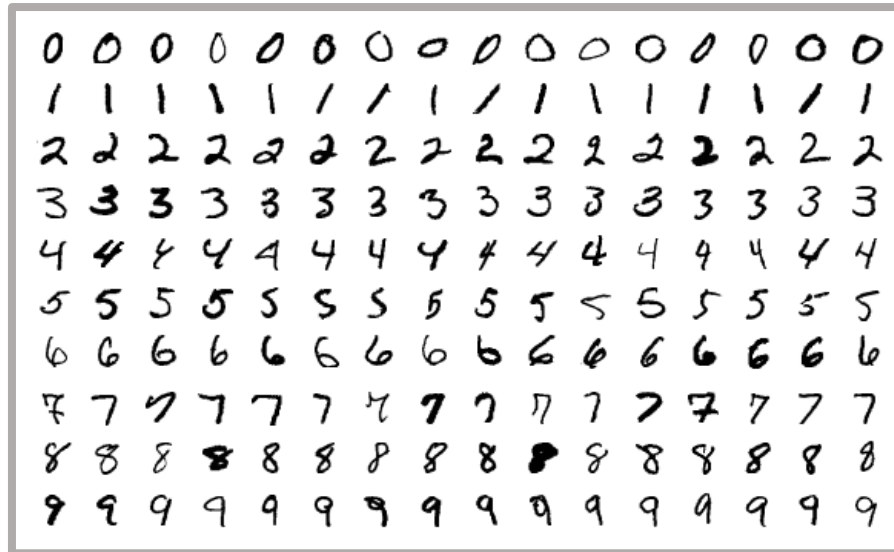
인공지능의 학습 유형 1) 지도 학습(Supervised Learning) – 회귀(Regression)

- 회귀 문제: 입력 데이터가 주어졌을 때, 실수 형태의 데이터를 출력합니다.
 - 학습 시간에 따른 영어 점수 예측
 - 거리에 따른 이동 시간 예측
 - 업무 시간에 따른 매출 예측

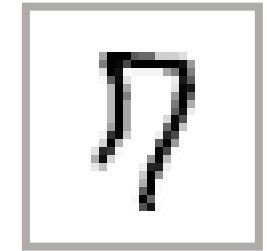


인공지능의 학습 유형 1) 지도 학습(Supervised Learning) – 분류(Classification)

- 분류 문제: 하나의 데이터가 주어졌을 때, 적절한 클래스로 분류를 수행합니다.
 - 손 글씨 분류
 - 강아지/고양이 분류
 - 배경 분류



학습 데이터셋



테스트 이미지



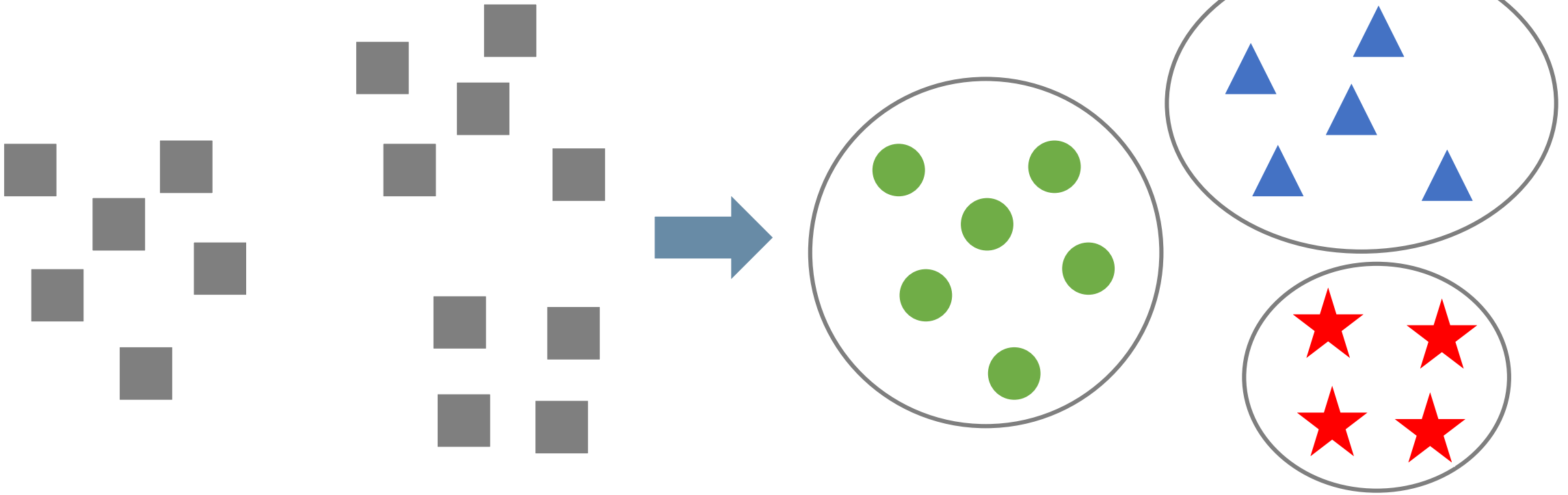
이 이미지는 7입니다.

인공지능의 학습 유형 2) 비지도 학습(Unsupervised Learning)

- 비지도 학습은 명시적인 정답을 제공하지 않으면서 학습시키는 유형입니다.
 - 클러스터링(Clustering)
 - 데이터를 특정한 기준으로 묶습니다.
 - 예시: "사용자들을 3가지 집단으로 나누고 싶어요."
 - 차원 축소(Dimensionality Reduction)
 - 차원을 줄여 데이터 내 유의미한 특징을 추출합니다.
 - 예시: "이 이미지들을 2차원 공간에 투영시켜서 시각화 할 수 있을까요?"

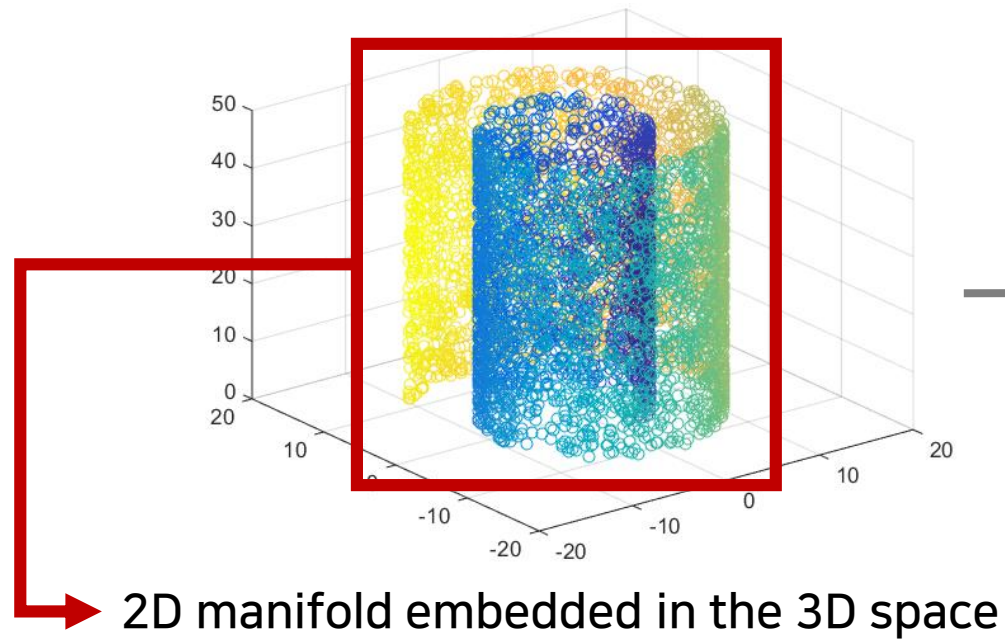
인공지능의 학습 유형 2) 비지도 학습(Unsupervised Learning) - 클러스터링(Clustering)

- 클러스터링(Clustering)
 - 다수의 데이터가 있을 때, 이를 K 개의 클러스터로 그룹화할 수 있습니다.
 - 예시) 비슷한 유형의 사용자끼리 그룹화하기

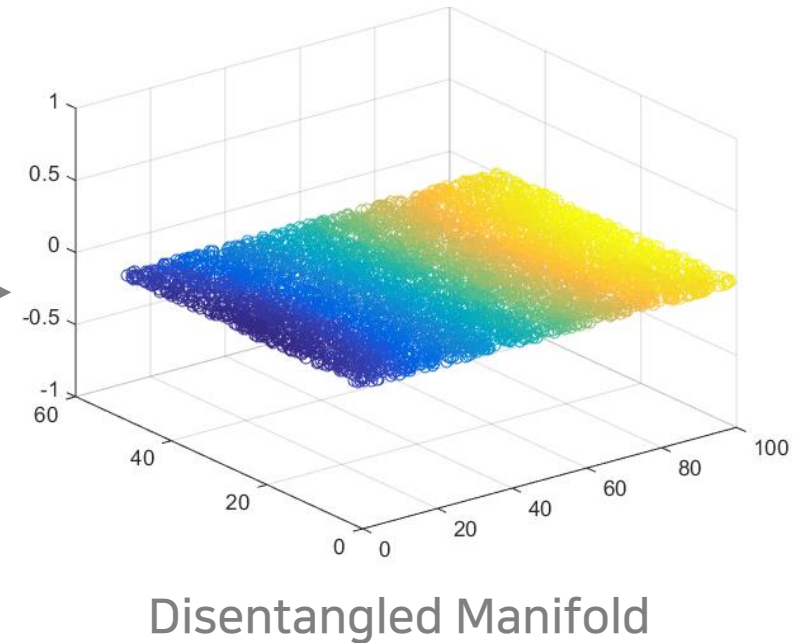


인공지능의 학습 유형 2) 비지도 학습(Unsupervised Learning) – 차원 축소

- 차원 축소(Dimension Reduction)
 - 고차원 데이터의 차원을 축소하여 새로운 차원의 데이터를 생성합니다.
 - 예시: 데이터 시각화, 데이터 압축을 통한 복잡도 개선

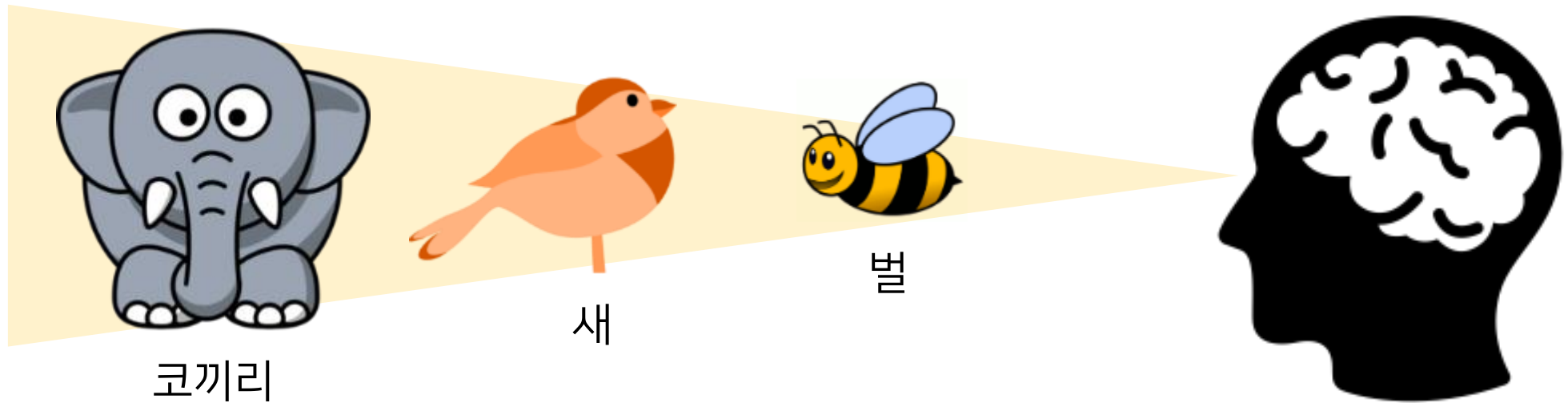


well-reduced



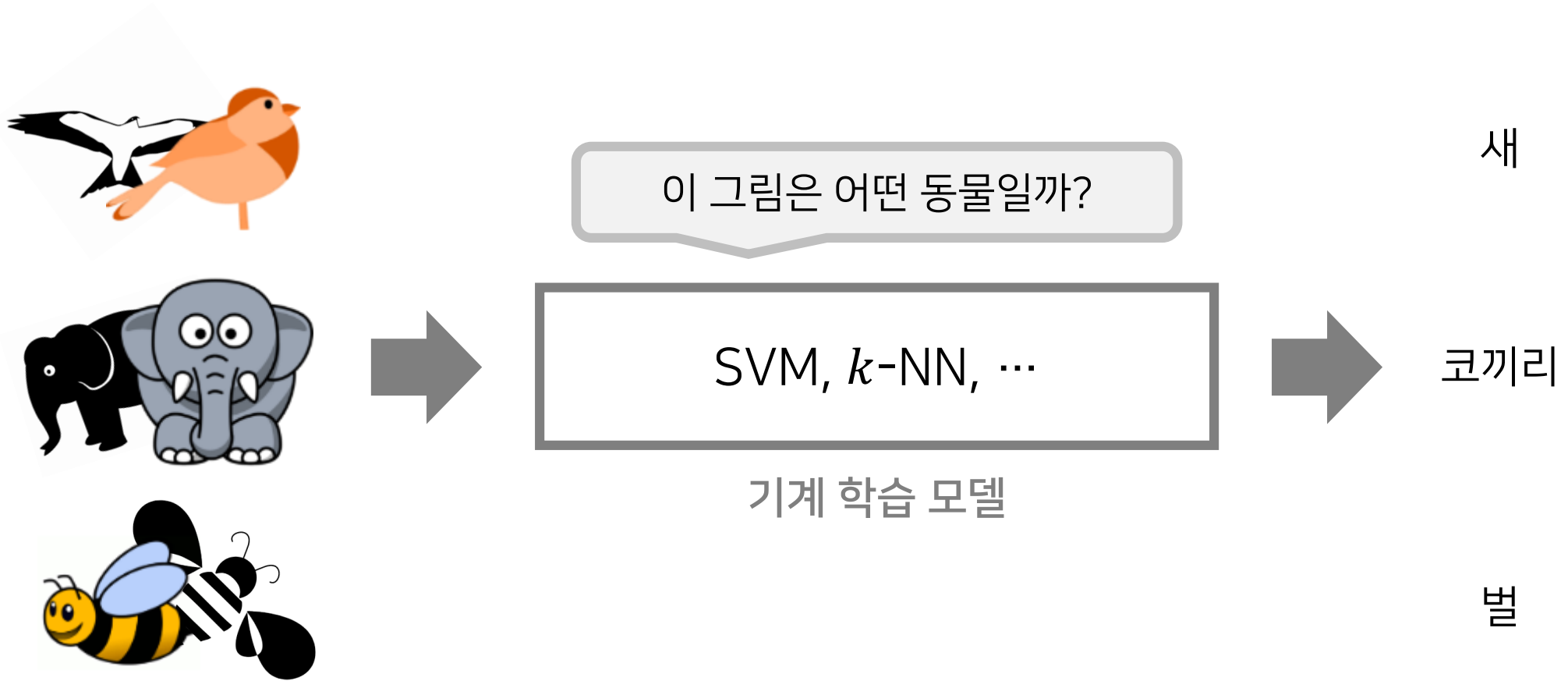
인간을 컴퓨터와 비교하자면?

- 인간은 어려서부터 수많은 입력을 받아서, 적절한 출력을 내보내도록 학습이 이루어집니다.
 - 입력: 시각, 청각, 촉각, ...
 - 출력: 목소리, 몸짓, ...



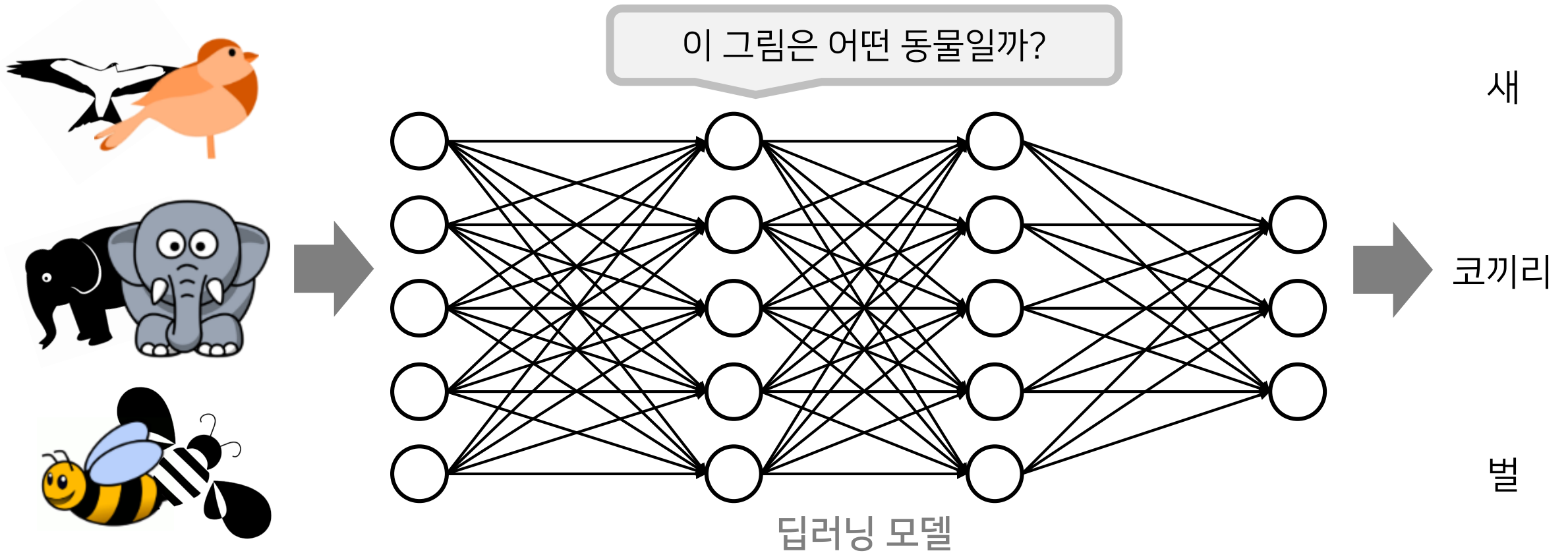
기계 학습

- 기계 학습: 데이터를 반복적으로 학습하여 데이터에 잠재된 특징을 학습합니다.



딥러닝

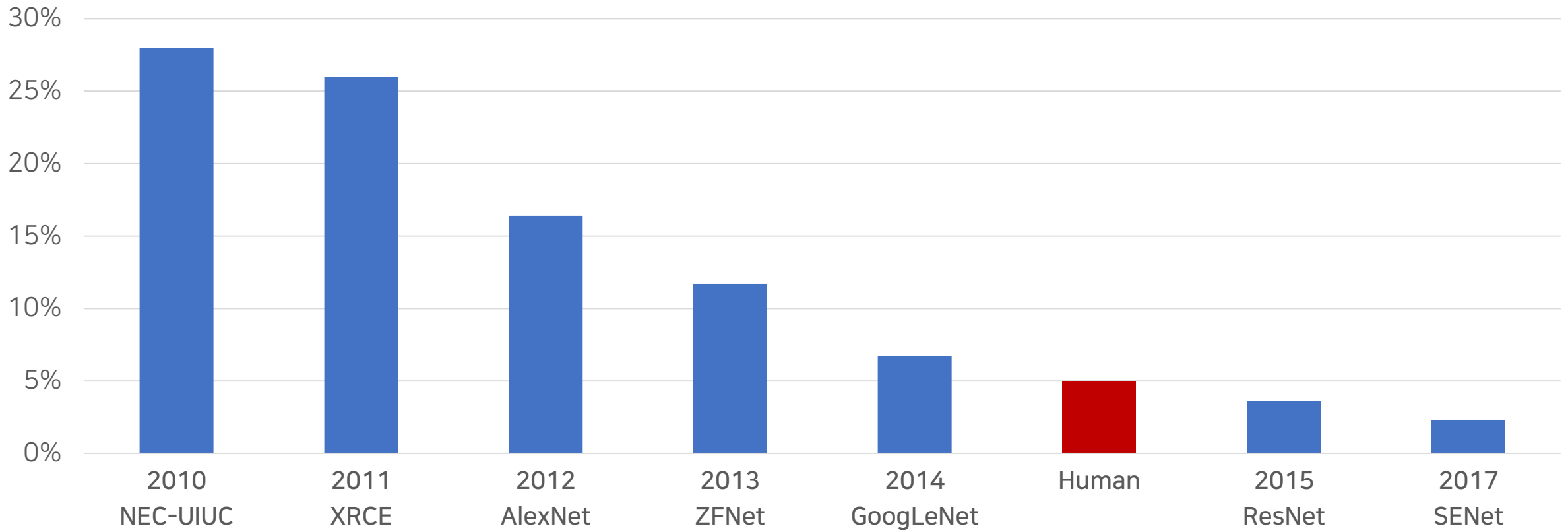
- 딥러닝: 깊은 인공 신경망을 활용하여 더 높은 정확도를 얻는 기계 학습 유형입니다.



딥러닝 기술의 발전

- 딥러닝은 다양한 분야에서 크게 활약하고 있으며, 특히 이미지 분류에서 강력한 성능을 보입니다.

ImageNet Top 5 Error Rate



딥러닝과 기계 학습

- 간단한 테이블 형식의(tabular) 데이터 세트에서는 전통적인 기계 학습 모델로도 충분히 우수한 성능을 보일 때가 많습니다.
 - 가장 먼저, 간단한 모델을 이용하여 타이타닉 생존자 예측 모델을 학습해 봅시다.

2강) 좋은 모델과 나쁜 모델

모델(Model)이란?

- 흔히 기계학습에서는 모델(model)이라는 표현을 많이 사용합니다.
 - 프로그래밍에서는 하나의 알고리즘은 함수로 구현될 수 있습니다.
 - **기계학습 모델**도 입출력을 가지는 일종의 알고리즘이자 함수라고 보면 됩니다.

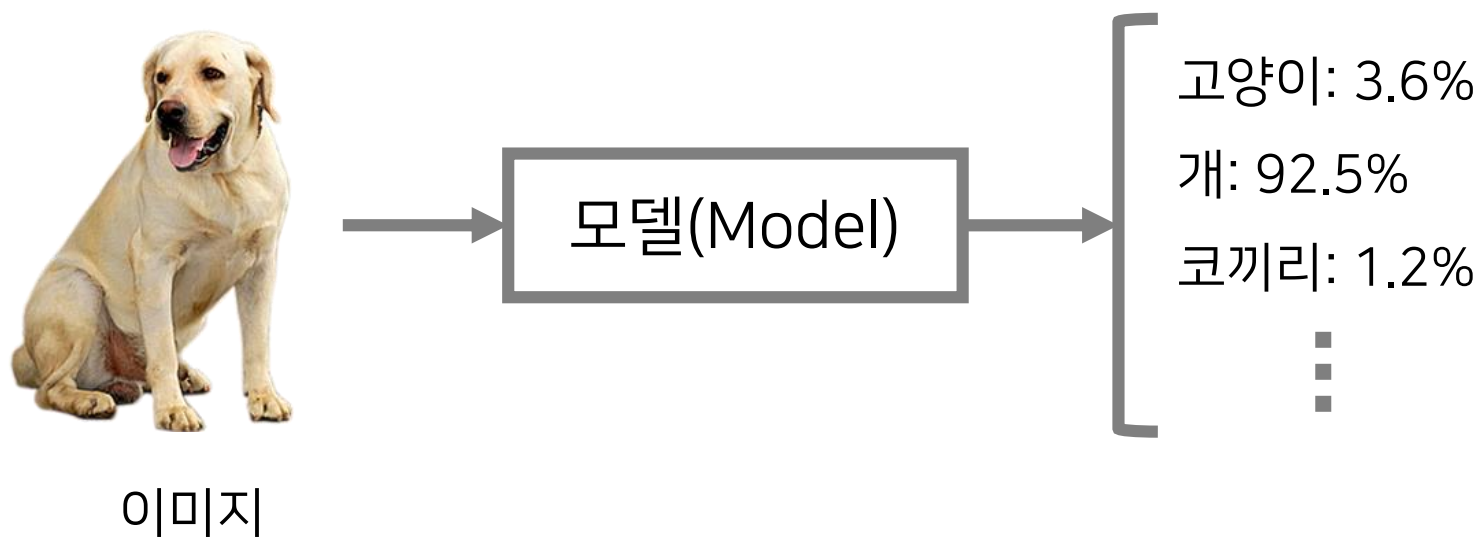
모델(Model)이란?

- 학습된 모델은 입력을 받아 출력을 수행하는 방식으로 동작합니다.
- 예시) 스팸 메시지 분류 모델



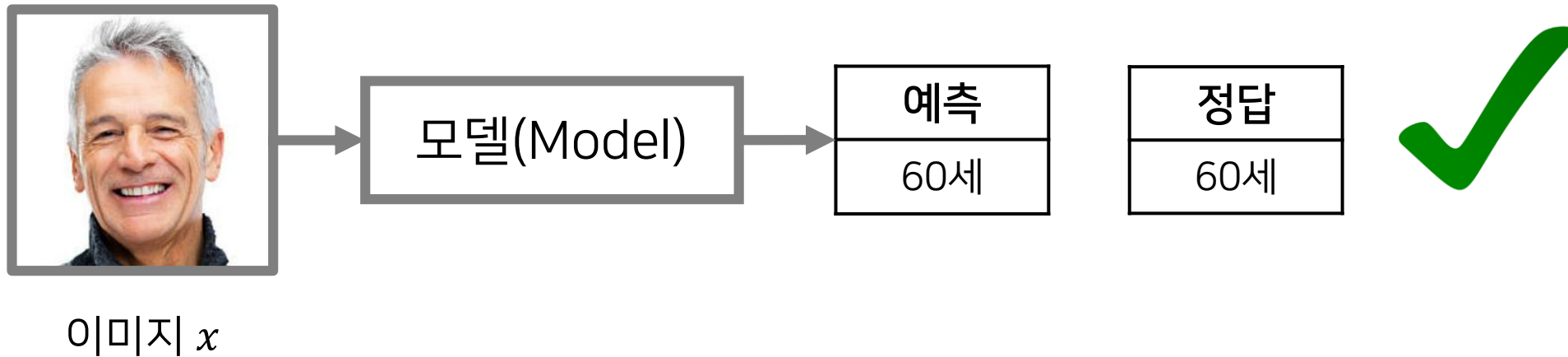
모델(Model)이란?

- 학습된 모델은 입력을 받아 출력을 수행하는 방식으로 동작합니다.
- 예시) 이미지 분류 모델



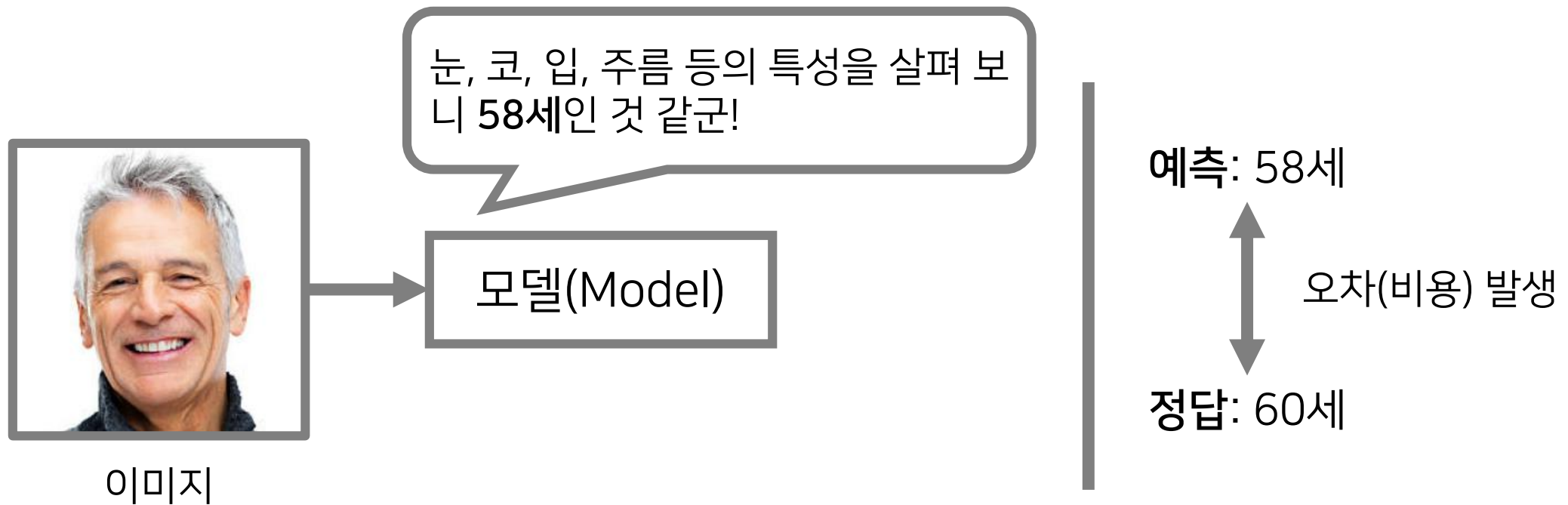
좋은 기계학습 모델이란?

- 그렇다면, 좋은 모델이란 무엇일까요?
 - 정답에 가까운 예측을 하는 모델이 좋은 모델입니다.
- 좋은 모델의 예시는 다음과 같습니다.



비용(Cost)

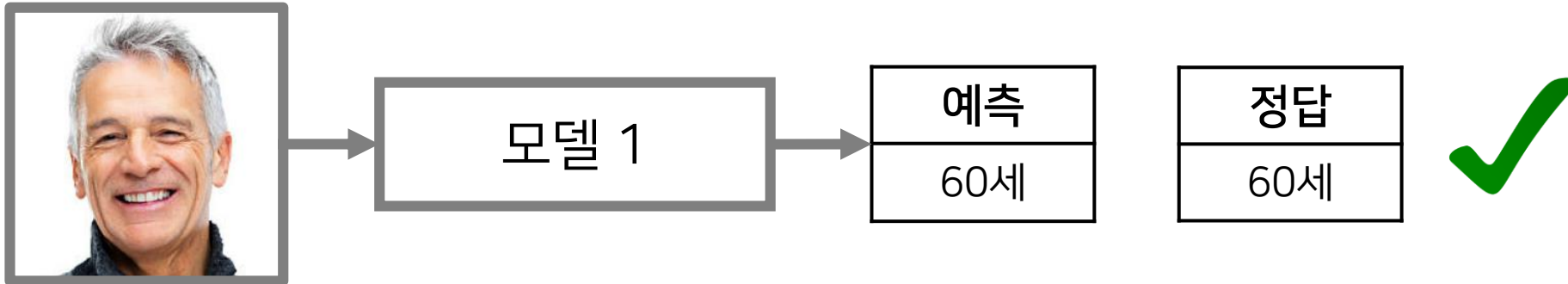
- 수학적으로 보았을 때, **비용(cost)**이 적은 모델이 좋은 모델입니다.
- **비용**: 모델이 ① 예측한 결과와 ② 실제 결과(정답)가 얼마나 다른지 측정된 값



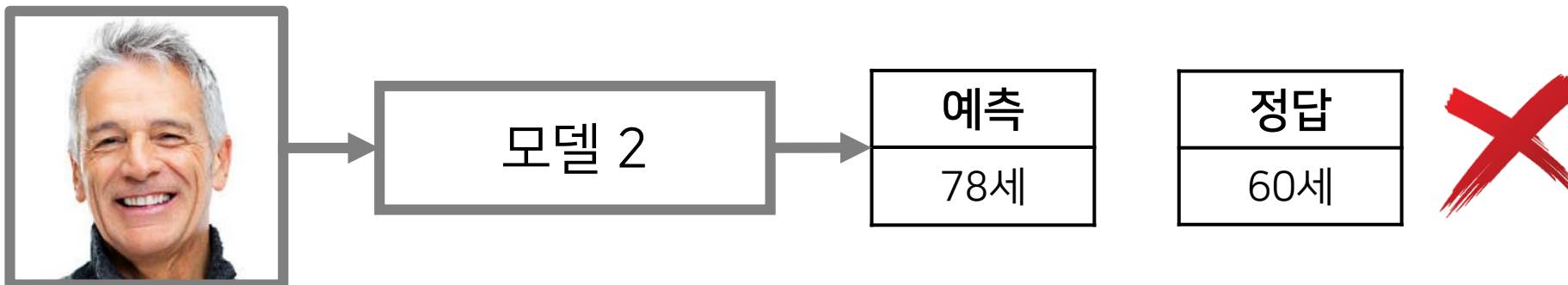
- 기계 학습은 이러한 **비용**을 줄이는 방향으로 학습을 진행합니다.

좋은 모델 vs. 나쁜 모델

- 좋은 모델의 예시

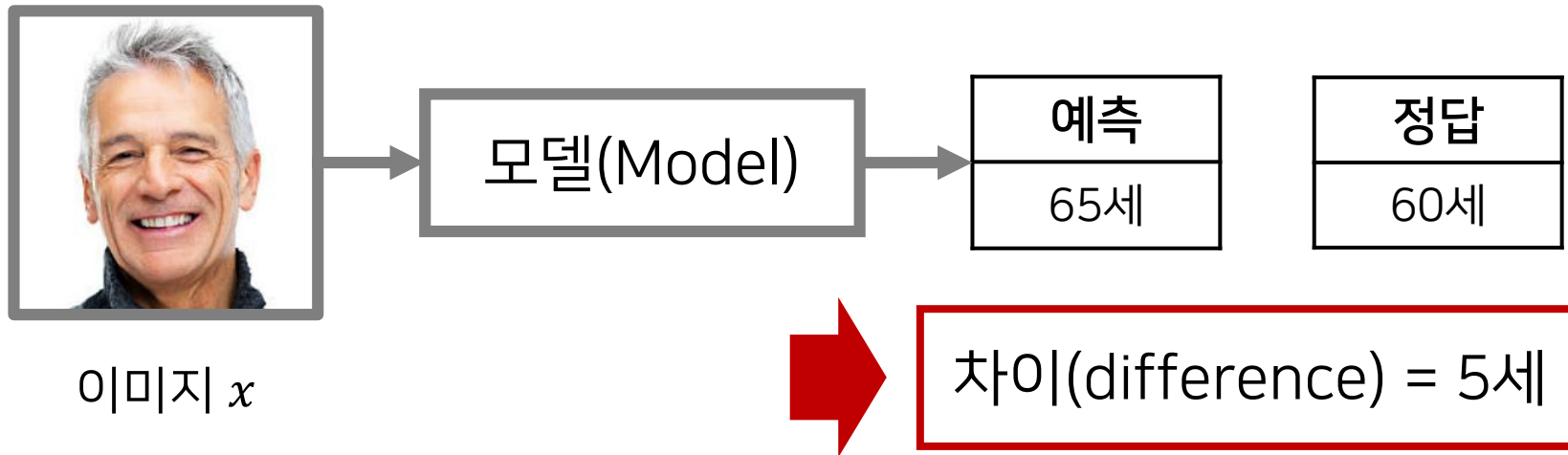


- 나쁜 모델의 예시



기준(Criterion)

- 좋은 모델의 여부를 판단하기 위해서는 "적절한 기준(criterion) = 비용(cost)"이 필요합니다.
- 나이 예측 모델을 생각해 봅시다.
 - 사람의 얼굴 이미지가 들어오면, 예측된(predicted) 나이가 출력됩니다.
- **실제 정답 나이와 예측 나이의 차이**를 비용으로 설정하면 어떨까요?
 - 차이가 크면 모델의 비용이 커지는 것이므로, 이는 합당합니다.



기준(Criterion) – 절대 오차 (Absolute Error)

- 절대 오차(absolute error, AE)를 비용으로 사용할 수 있습니다.
 - 예측 값에서 정답 값을 뺀 뒤에, 절댓값을 취합니다.



- A 모델의 경우 예측한 나이와 실제 나이가 1살만 차이가 난다.
 - A 모델이 더 우수한 모델임을 알 수 있다.

기준(Criterion) 예시 - 평균 절대 오차 (Mean Absolute Error)

- 우리에게 N 개의 학습 데이터가 있다고 가정합니다.
- N 개의 데이터에 대하여 정답 y 와 예측 \hat{y} 의 차이(difference)의 **평균**을 구할 수 있습니다.
- 이를 **평균 절대 오차(MAE)**라고 합니다.

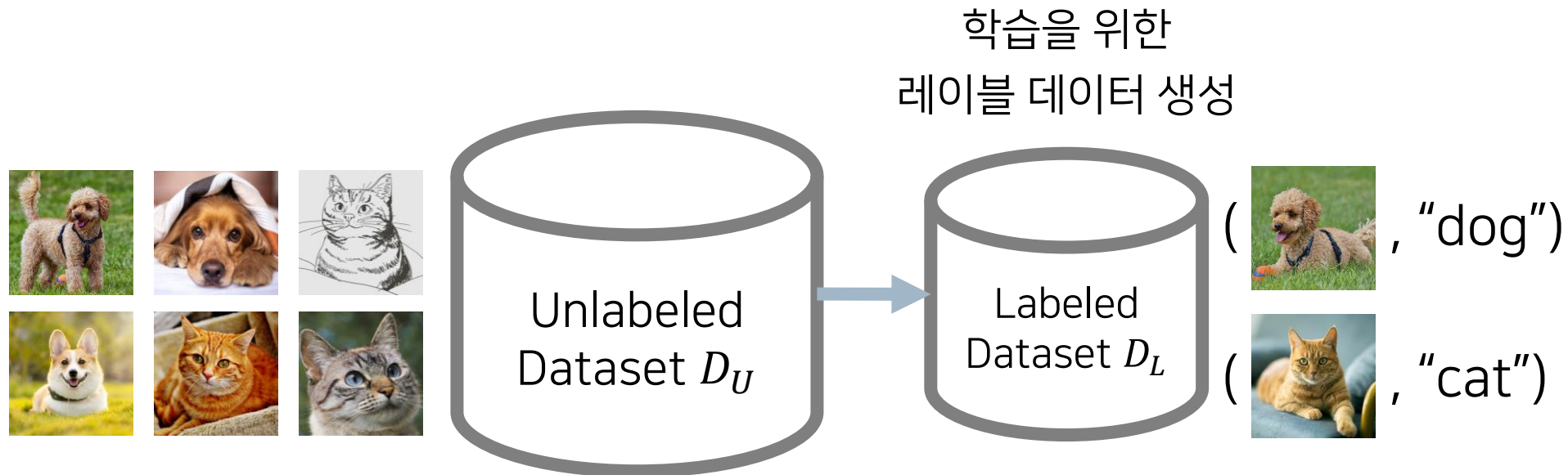
$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$$

- 우리는 이러한 **비용(cost)**을 줄이는 방향으로 기계학습 모델을 학습합니다.

3강) 데이터와 데이터 세트

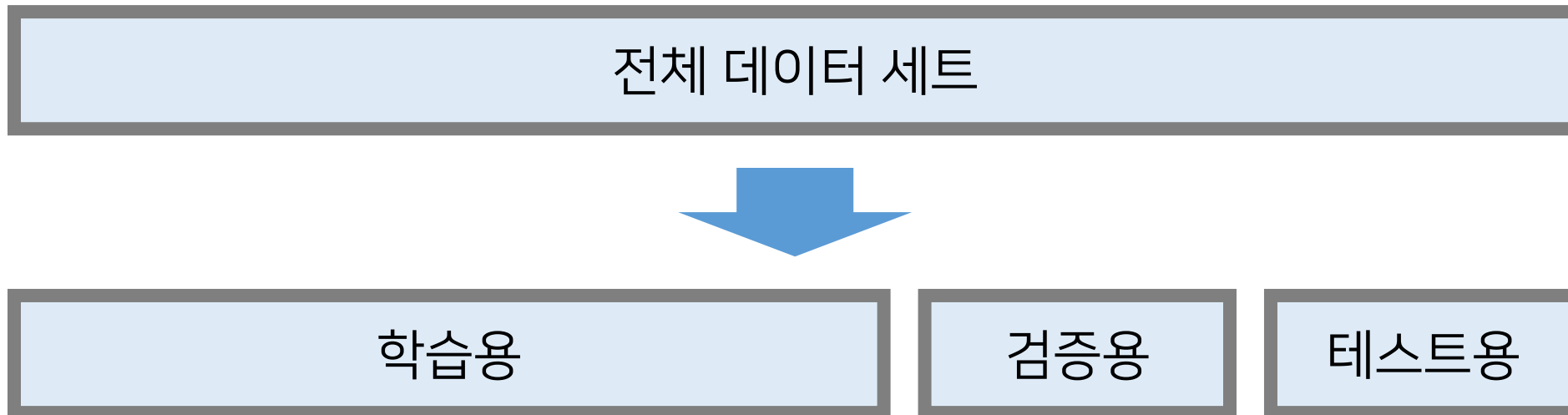
데이터 세트(Dataset)

- 기계학습에서 가장 중요한 준비물 중 하나는 바로 데이터 세트입니다.
- 예를 들어 이미지 분류 모델에서는 (이미지, 레이블) 형태의 데이터 세트가 필요합니다.
- 통상적으로 인공지능 분야에서는 학습을 위해 수천 개에서 수만 개의 데이터 세트가 필요합니다.



데이터 세트의 분할

- 좋은 기계 학습 모델을 만들기 위하여, 가지고 있는 전체 데이터 세트 일반적으로 세 가지로 나눕니다.
- 기계 학습 모델을 학습할 때, 전체 데이터 세트를 적절히 구분하여 사용합니다.
- 전체 데이터 세트를 ① 학습 목적, ② 검증 목적, ③ 테스트 목적으로 구분합니다.



데이터 세트의 분할

- 학습 데이터 세트(training dataset): 실질적으로 학습할 때 사용하는 데이터 세트
- 검증 데이터 세트(validation dataset): 학습된 모델을 검증하기 위해 사용하는 데이터 세트 일반적으로 모델을 배포할 때는 검증 정확도가 가장 높은 모델을 사용합니다.
- 테스트 데이터 세트(test dataset): 학습된 모델을 최종적으로 평가하기 위해 사용합니다.

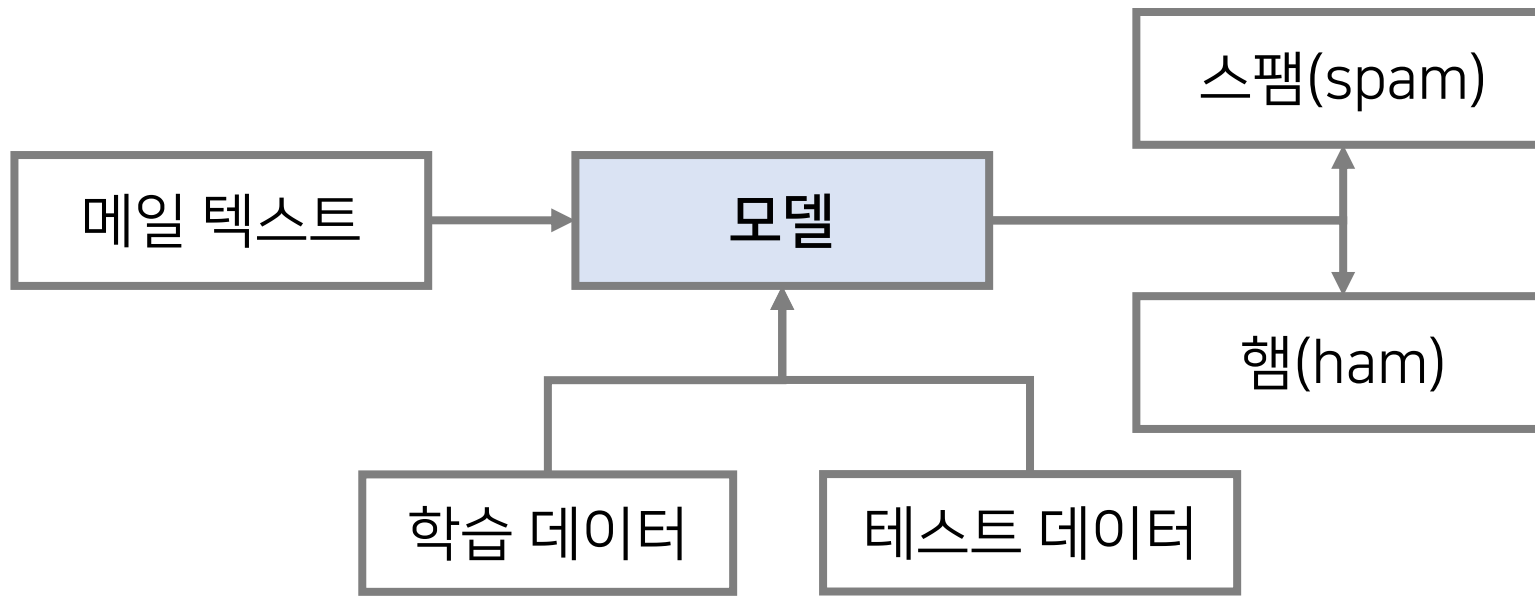
학습용

검증용

테스트용

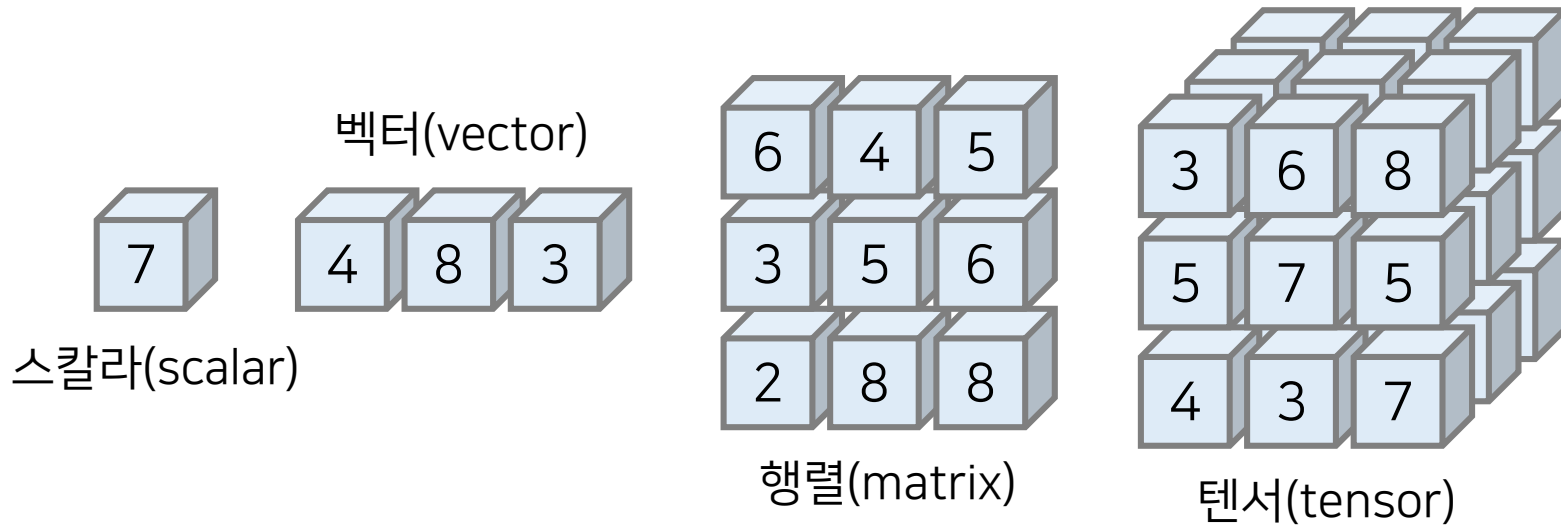
데이터 세트의 분할 - 텍스트 분류 예시

- 텍스트 분류는 주어진 텍스트를 특정한 클래스(class)로 분류하는 작업을 의미합니다.
 - 학습 데이터: 학습 과정에서 모델을 훈련하기 위해 사용하는 데이터
 - 학습 데이터를 ① 실제 학습 목적, ② 학습 중인 모델 검증 목적으로 나누곤 합니다.
 - 테스트 데이터: 학습된 이후에 모델의 성능을 테스트하기 위해 사용하는 데이터



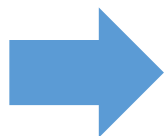
데이터의 단위

- 기계 학습에서 쓰이는 데이터들은, 컴퓨터의 배열(Python의 리스트)로 이해할 수 있습니다.
 - 스칼라(scalar): 하나의 변수(0차원의 점)
 - 벡터(vector): 1차원 배열 → 참고로 수열(sequence)을 나타낼 때도 1차원 배열을 사용합니다.
 - 행렬(matrix): 2차원 배열
 - 텐서(tensor): 3차원 이상의 배열



1차원 배열과 벡터

- 대학교 이전 교육과정에서는 일반적으로 변수가 하나인 경우를 다룹니다.
- 하지만 현실세계에서는 변수가 여러 개인 경우가 많습니다.
 - 예시) ① 나이, ② 공부한 시간, ③ 성별이 입력으로 들어왔을 때 → 점수를 예측하는 문제
- 기계 학습에서 이미지나 텍스트는 많은 정보(변수)를 담고 있으므로, 종종 벡터로 표현됩니다.
- 컴퓨터에서는 벡터를 표현할 때 1차원 배열을 사용합니다.

$$\begin{bmatrix} 4.3 \\ 5.5 \\ 7.2 \\ 2.4 \\ 5.0 \end{bmatrix}$$


0	1	2	3	4
4.3	5.5	7.2	2.4	5.0

4강) 데이터 전처리란?

데이터 전처리(Data Pre-processing)

- 데이터 전처리는 기계 학습/딥러닝 분야에서 매우 중요합니다.
- 데이터가 제대로 처리되어 있지 않으면, 모델 학습 자체가 어려운 경우가 많습니다.

데이터 전처리 - 결측 값(Missing Value)

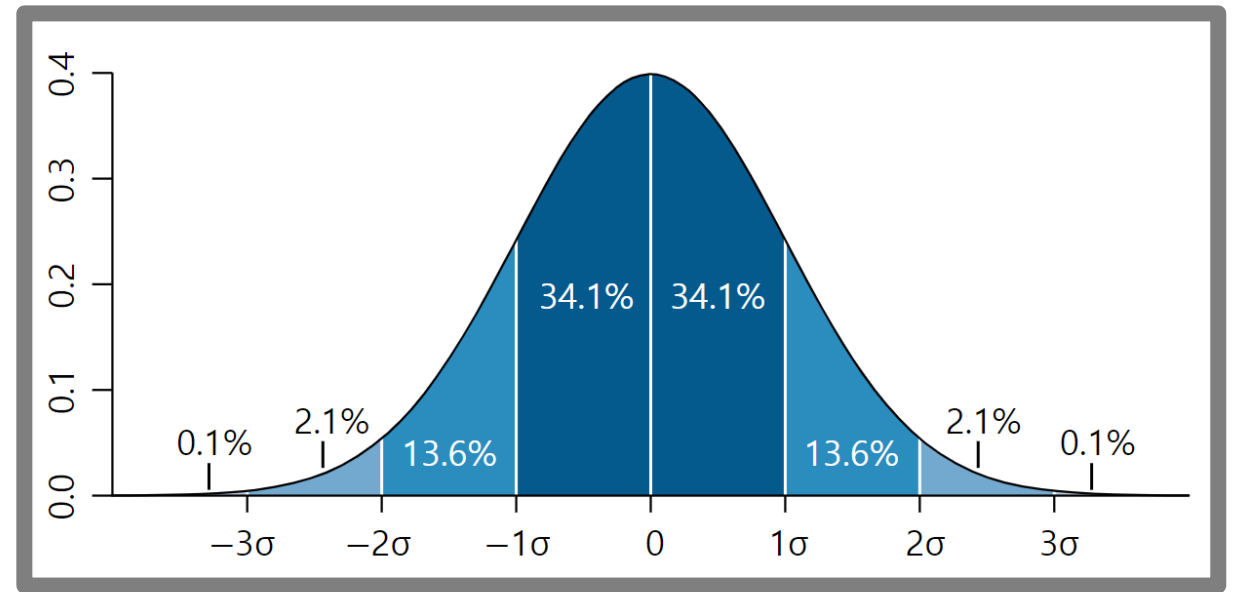
- 수능 성적으로, 미래 소득 분위기를 예측하는 모델을 만든다고 해봅시다.
- 어떤 학생의 경우, 국어와 수학 시험까지 참여했을 수도 있습니다.
 - 이처럼 영어 데이터가 없다면 어떻게 할까요?

	국어	수학	영어
학생 1	97	100	X
학생 2	83	95	92
학생 3	95	92	88

- 처리 예시 1) 다른 성적의 평균으로 대체합니다.
- 처리 예시 2) 결측 값(missing value)이 포함된 데이터(row)는 제거합니다.

데이터 전처리 - 이상치(Outlier)

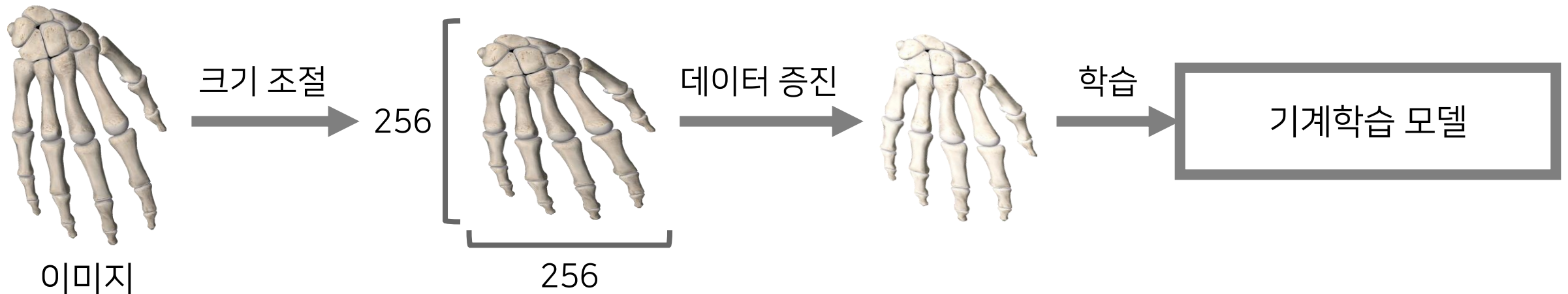
- 다른 데이터와 다르게 과하게 작거나 큰 데이터를 제거할 수 있습니다.
- 데이터를 표준 정규 분포(평균: 0, 표준 편차: 1)로 변경할 수 있습니다.
- $[-2, 2]$ 에 해당하는 값만 취할 수 있습니다.
 - $P(Z \leq 2)$ 는 약 98.17%에 해당합니다.



- 따라서 $[-2, 2]$ 범위에 포함되는 비율은 약 96%에 해당합니다.
- 즉, 이상치(outlier)로 판단되는 4%의 데이터는 사용하지 않을 수 있습니다.

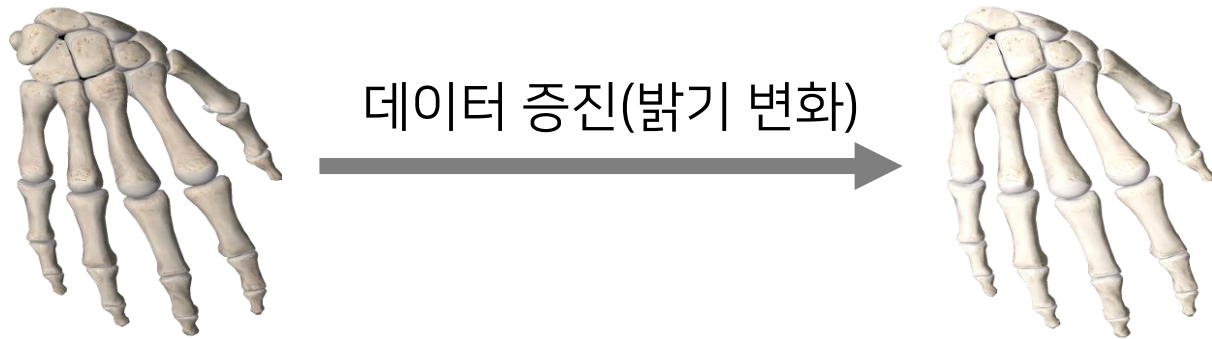
데이터 전처리 - 모델에 입력으로 넣기

- 기계 학습 모델에 입력으로 넣기 전에도 전처리(pre-processing)가 필수적으로 사용됩니다.
 - 이미지, 텍스트 등 데이터의 종류와 상관 없이 사용됩니다.
- 일반적으로 기계학습 모델은 동일한 크기 및 정규화된 이미지를 입력으로 받습니다.
 - ① 크기 변경(resize) → ② 데이터 증진(augmentation) → ③ 정규화(normalization)



데이터 전처리 - 증진(Augmentation)

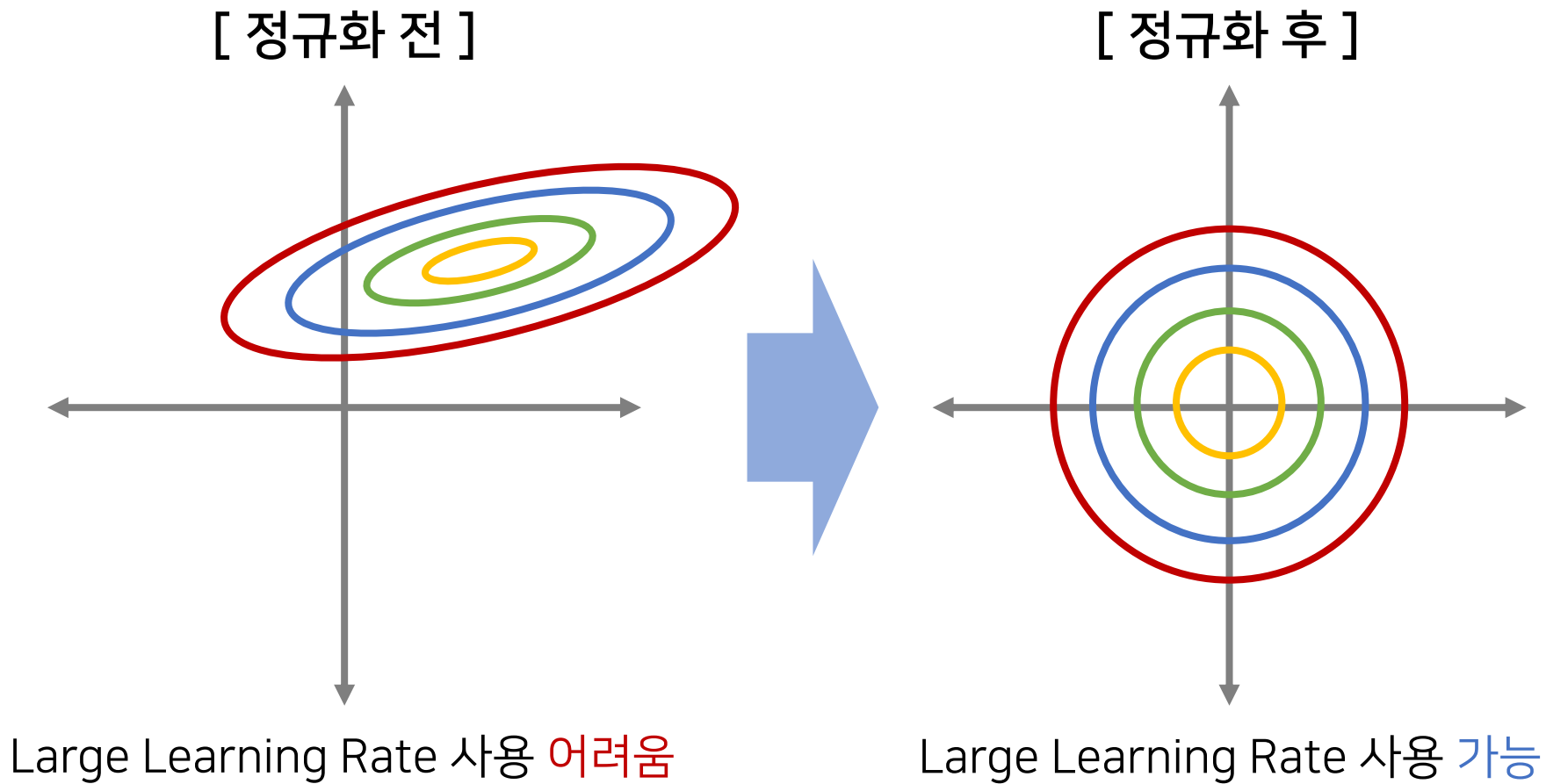
- 데이터의 개수가 적으면 모델의 학습 결과가 좋지 않으므로, 데이터 증진을 사용합니다.
- 데이터 증진(data augmentation): 이미지에 변형을 가해 새로운 이미지를 생성하는 기법
- 모델에 입력으로 넣기 전에 데이터를 증진하여, 데이터가 많은 것과 유사한 효과를 낼 수 있습니다.
 - 예를 들어, 입력 이미지의 밝기를 조절할 수 있습니다.



일반적으로 학습할 데이터는 한정적입니다. 그래서 데이터를 변형(transformation)하여 다수의 데이터를 생성할 수 있습니다.

데이터 전처리 - 입력 데이터 정규화(Normalization)

- 입력 데이터 정규화: 각 차원의 데이터의 평균을 0으로 만들고, 동일 범위 내의 값을 갖게 만듭니다.

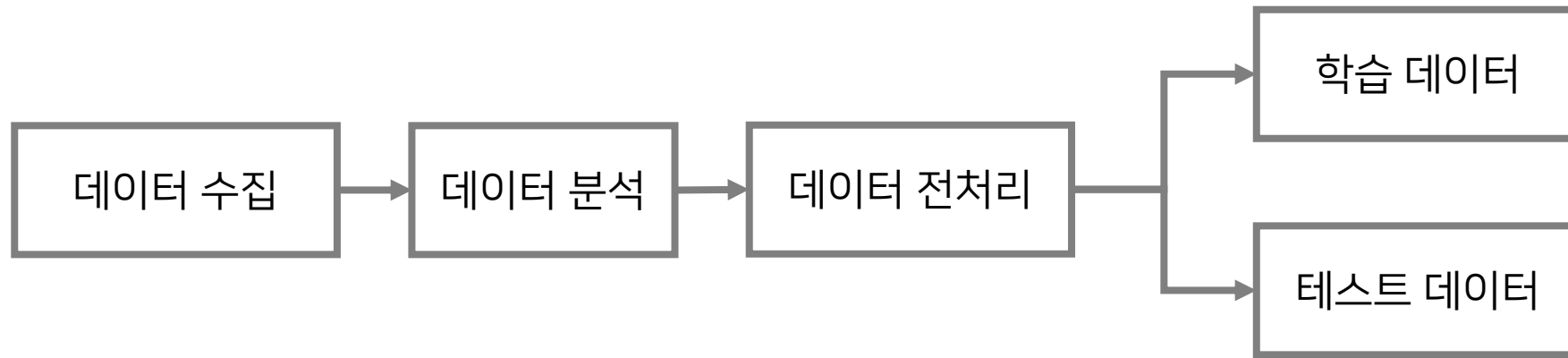


5강) 모델 학습 과정과 과대 적합(Overfitting)

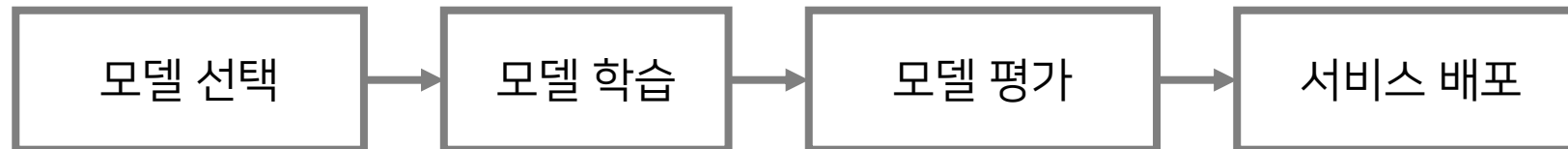
기계 학습의 수행 과정

- 기계 학습을 수행하는 일반적인 과정은 다음과 같습니다.

1. 데이터 처리하기



2. 모델 처리하기

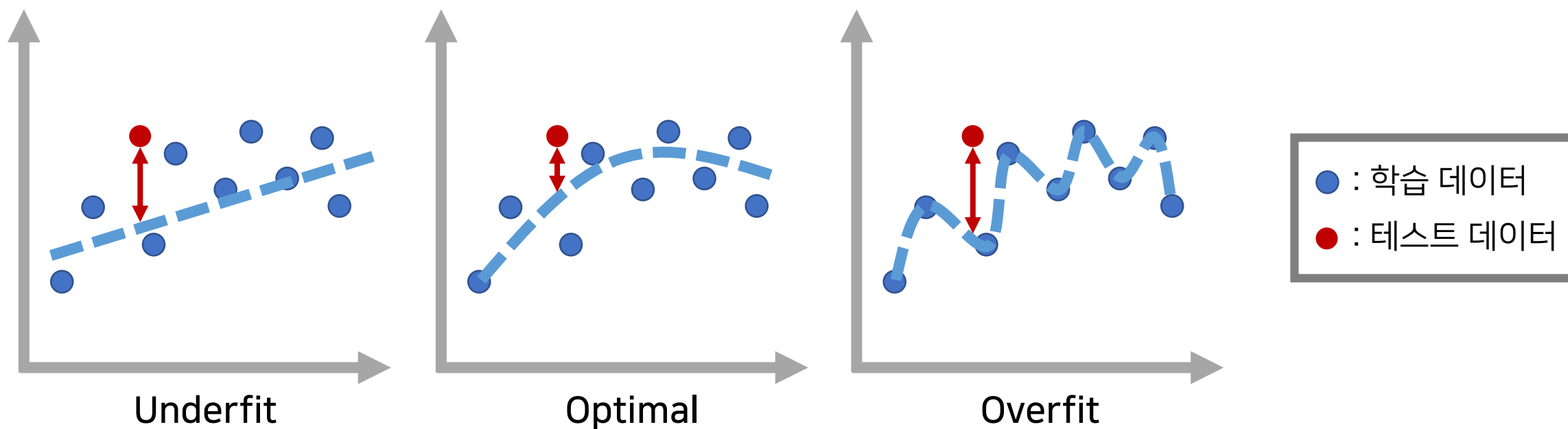


과대 적합(Overfitting)

- 기계 학습 모델을 학습 데이터에 대해서만 단순히 좋은 정확도를 보이도록 학습하면 어떻게 될까요?
 - 과대 적합(overfitting)이 발생할 수 있습니다.
 - 학습 데이터 세트에 **대해서만** 맞추어져(fitted) 좋은 성능을 보일 수 있습니다.
 - 실제로 중요한 테스트 데이터 세트에 대한 정확도가 낮다면 무용지물입니다.

과대 적합(Overfitting)

- 모델을 학습 데이터에 **"적절한 수준으로"** 학습할 필요가 있습니다.
- 과소 적합(under-fitting): 데이터에 과소하게 적합이 된 경우
- 과대 적합(over-fitting): 데이터에 과대하게 적합이 된 경우 (학습 데이터에 대해서만 좋은 성능)
 - 과대 적합이 발생하면, 실제 테스트 데이터에 대해서 좋지 않은 정확도를 보일 수 있습니다.



과대 적합(Overfitting)의 사례

- 일반적으로 학습 데이터로 학습을 하되, 검증 정확도가 가장 높을 때의 모델을 채택합니다.
- 결과적으로 **테스트 정확도**가 높게 나오는 것이 목표입니다.

학습 과정	검증 정확도
1단계	80.2%
2단계	91.9%
3단계	97.1%
4단계	92.4%
...	...



채택! 테스트 정확도: 96.5%

6강) 기계 학습을 위한 기초 라이브러리 소개

사이킷런(Scikit-learn) 라이브러리 소개

- 기계 학습에 **입문하는 사람**도 가볍게 사용할 수 있는 기계 학습 라이브러리입니다.
- 기계 학습을 배우기 위해 자주 사용되는 공부 목적의 데이터 세트를 제공합니다.
 - 예를 들어, 보스턴 집값 데이터 세트를 곧바로 불러와 사용할 수 있습니다.
- 기본적인 기계 학습 알고리즘을 곧바로 가져와 적용할 수 있습니다.

Pandas 라이브러리

- 엑셀(excel)과 같은 기능이 필요할 때 사용하는 라이브러리입니다.
 - 엑셀(Excel)과 유사한 기능을 제공합니다.
- Pandas의 데이터프레임(data-frame)은 일종의 테이블(table)과 같습니다.
- 작은 크기의 데이터 세트는 Pandas로 불러와 곧 바로 처리해 사용할 수 있습니다.
- 테이블 형태의 데이터를 효과적으로 처리하고, 보여줄 수 있도록 도와주는 라이브러리입니다.
- NumPy와 함께 사용되어 연계되는 기능을 제공합니다.

Pandas 라이브러리 - 시리즈(Series)

- 시리즈(series)를 기본적인 데이터로 사용합니다.
 - 엑셀(excel)에서 하나의 컬럼(column)으로 이해할 수 있습니다.
 - 인덱스(index)와 값(value)의 쌍으로 구성됩니다.
 - 인덱스(index)에 따라 데이터를 나열하므로 Python의 사전(dictionary) 데이터 타입에 가깝습니다.

인덱스	meaning
Apple	사과
Banana	바나나
Carrot	당근
Durian	두리안

Pandas 라이브러리 - 데이터프레임(Data-frame)

- 데이터 프레임은 여러 개의 시리즈(series) = 컬럼(column)의 묶음으로 이해할 수 있습니다.
- 하나의 완성된 테이블(table) 형태로 이해할 수 있습니다.

인덱스	meaning	frequency	Importance
Apple	사과	3	5
Banana	바나나	5	2
Carrot	당근	7	1
Durian	두리안	2	1

Matplotlib 라이브러리

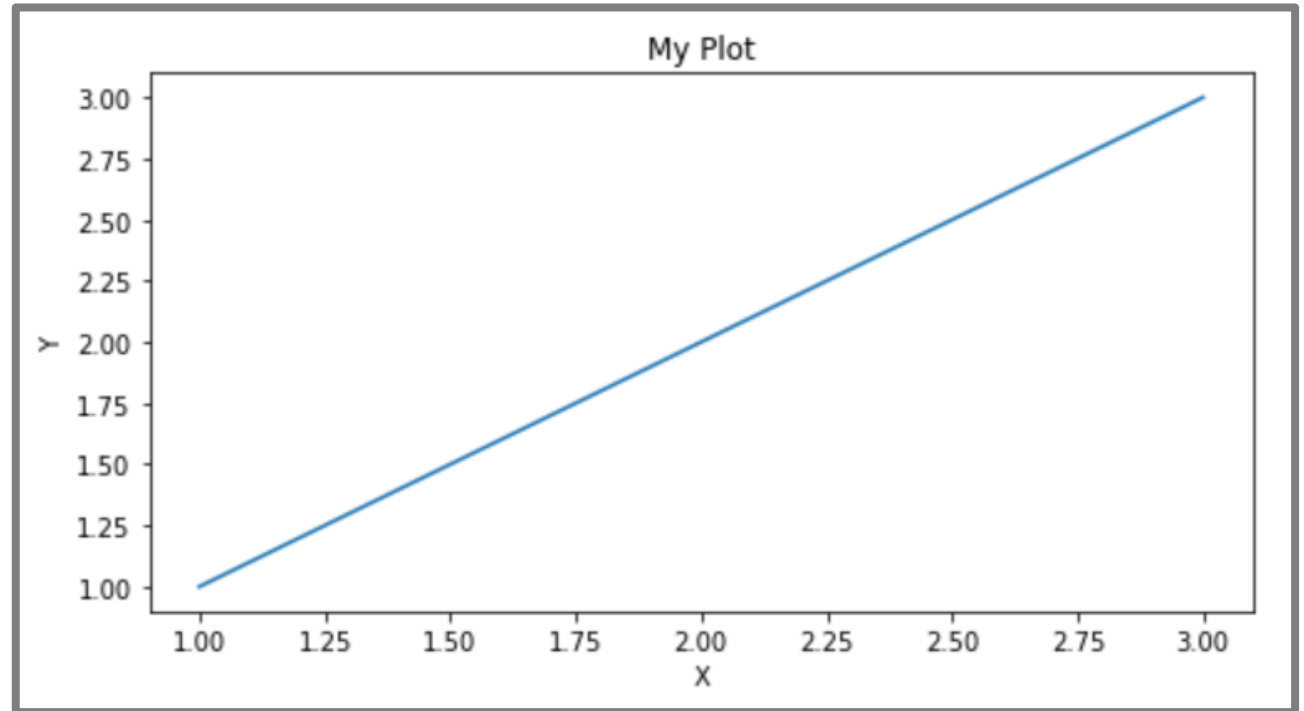
- 다양한 데이터를 시각화 할 수 있도록 도와주는 라이브러리입니다.
- 간단한 데이터 분석에서부터 인공지능 모델의 시각화까지 활용도가 매우 높습니다.

Matplotlib 라이브러리 - 직선 그래프 그리기

- 간단히 직선 그래프를 그릴 수 있습니다.

```
import matplotlib.pyplot as plt

x = [1, 2, 3]
y = [1, 2, 3]
plt.plot(x, y)
plt.title("My Plot")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

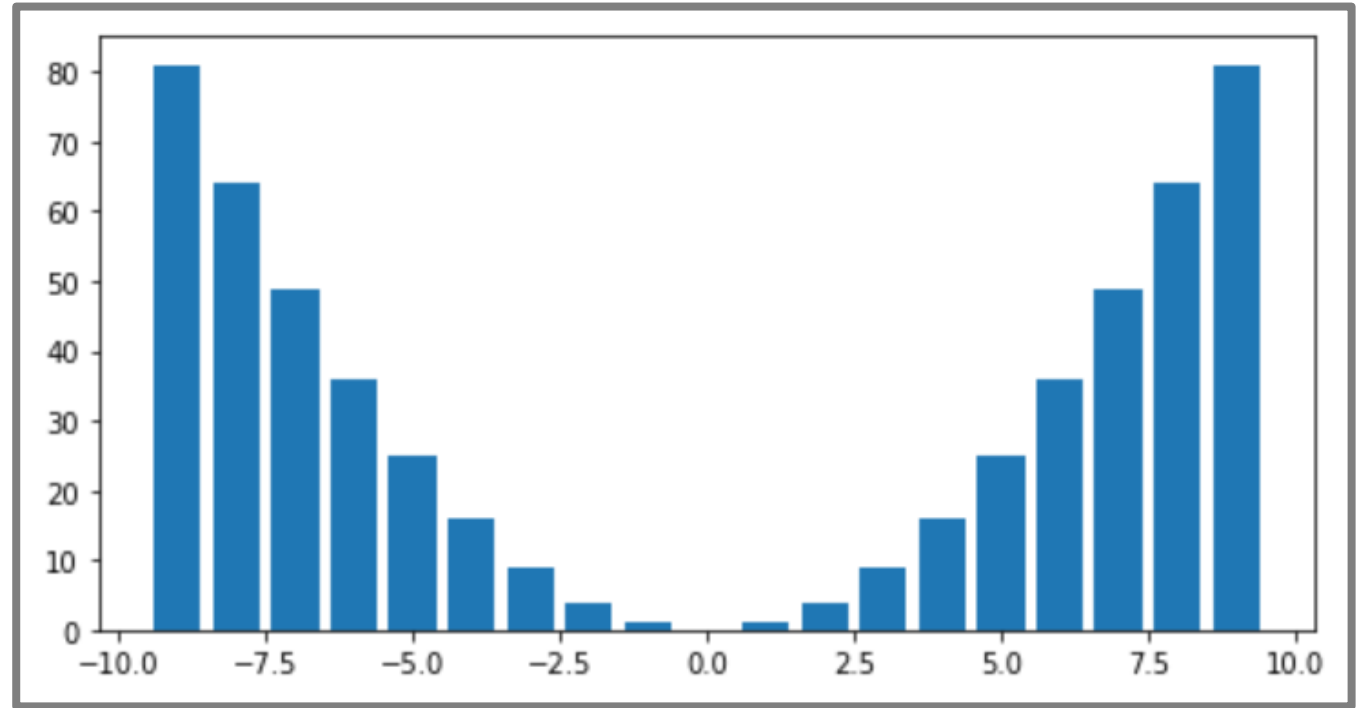


Matplotlib 라이브러리 - 막대 그래프 그리기

- 간단히 막대 그래프를 그릴 수 있습니다.

```
import matplotlib.pyplot as plt
import numpy as np

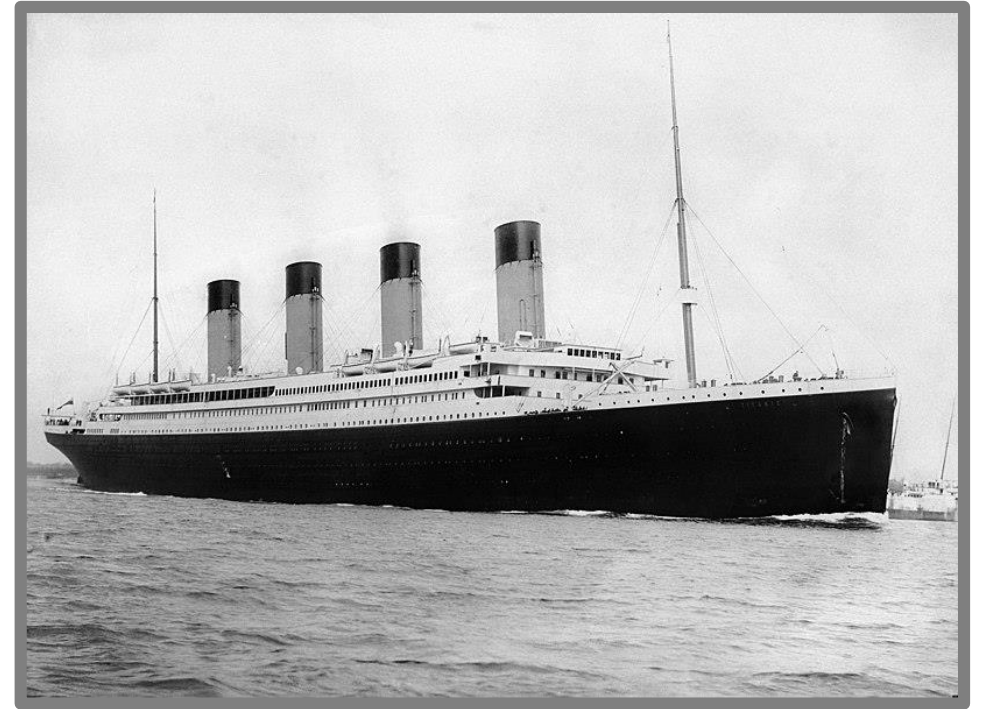
x = np.arange(-9, 10)
plt.bar(x, x ** 2)
plt.show()
```



7강) 타이타닉 생존자 예측 문제 개요

타이타닉 생존자 예측 - 문제 설명

- 1912년, 타이타닉호가 빙산과 충돌하여 침몰했습니다.
- 당시에 구명 보트가 충분하지 않아 2,224명의 승객과 승무원 중에서 1,502명이 사망했습니다.
- 이때, 단순히 운에 따라서 생존 여부가 갈리기 보다는, 특정한 사람이 가지는 특징(feature)이 생존 여부를 결정하는 요인으로 작용했습니다.
 - 예시) 어린이와 노인의 생존율이 높았습니다.



* https://ko.wikipedia.org/wiki/RMS_%ED%83%80%EC%9D%B4%ED%83%80%EB%8B%89

타이타닉 생존자 예측 - 특징(Feature) 분석

- 타이타닉호에 타 있었던 사람들을 구분할 수 있는 **특징**으로는 다음과 같은 것들이 있습니다.
- 이러한 특징에 따라서 생존 여부를 예측할 수 있을까요?

<타이타닉호에 탄 사람들의 특징들>

- ① 나이
- ② 성별
- ③ 경제적 계층 등

...

타이타닉 생존자 예측 - 가설 세우기

- 우리가 기계 학습 모델을 만들 때는 가설을 세운 뒤에, 반영하는 것이 유리합니다.
- 예를 들어 다음과 같은 가설을 세울 수 있습니다.

[가설 단계]

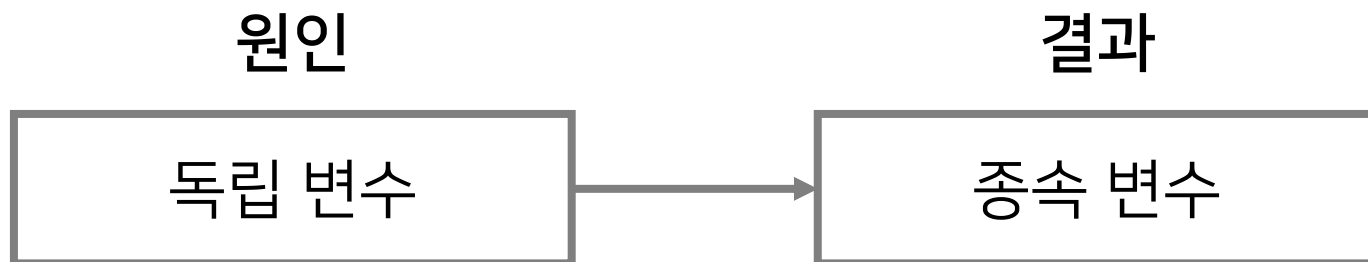
- 젊은 남성에 비해 노인과 여성은 상대적으로 보호받을 것입니다.
- 따라서, 노인과 여성은 생존율이 높을 것입니다.

[확인 단계]

- 우리가 가지고 있는 데이터를 분석하면 이러한 가설이 맞는지 검토할 수 있습니다.

독립 변수와 종속 변수

- 독립 변수와 종속 변수의 개념을 쉽게 설명하자면 다음과 같습니다.
 - 독립 변수: 원인이 되는 변수
 - 종속 변수: 결과가 되는 변수



독립 변수와 종속 변수 예시

- 가설: 노인의 경우 배려를 받아 생존율이 높을 것입니다.
- 이때, 독립 변수와 종속 변수는 다음과 같습니다.
 - 독립 변수: 나이
 - 종속 변수: 생존 여부

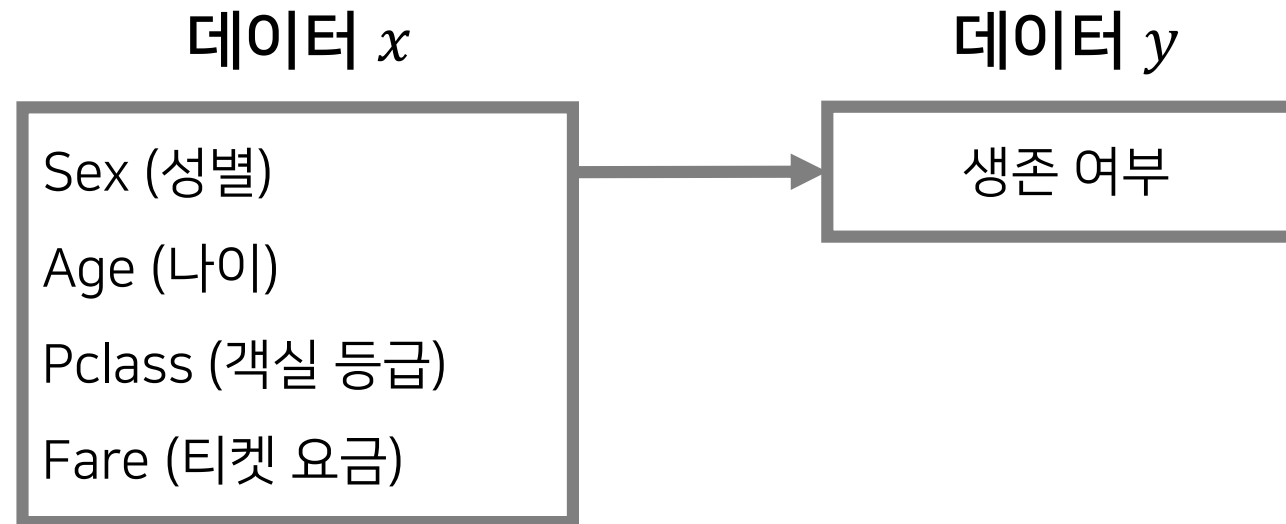
타이타닉 생존자 예측 데이터 세트 - 입력 데이터 특징(Feature)

- 아래 특징에서 생존 여부(survival)을 제외한 나머지 특징은 모두 독립 변수입니다.

번호	특징(Feature)	설명(Description)
1	PassengerId	탑승자 고유 번호
2	Survival	생존 여부 (생존: 1, 사망 0)
3	Pclass	객실 등급(1: 1등급, 2: 2등급, 3: 3등급)
4	Name	이름
5	Sex	성별
6	Age	나이
7	Sibsp	함께 탑승한 형제자매 혹은 배우자의 수
8	Parch	함께 탑승한 부모 혹은 자식의 수
9	Ticket	티켓 번호
10	Fare	티켓 요금
11	Cabin	객실 번호
12	Embarked	탑승장(Cherbourg, Queenstown, Southampton)

타이타닉 생존자 예측 데이터 세트 - 확인해 보기

- 테스트(test) 데이터 세트에는 생존 여부(survived) 컬럼이 존재하지 않습니다.
 - 생존 여부는 우리가 맞추어야 하는 **정답 y 값**이기 때문입니다.
- 데이터 중에서 네 가지만 이용한다고 가정해 봅시다.
 - ① 성별, ② 나이, ③ 객실 등급, ④ 티켓 요금

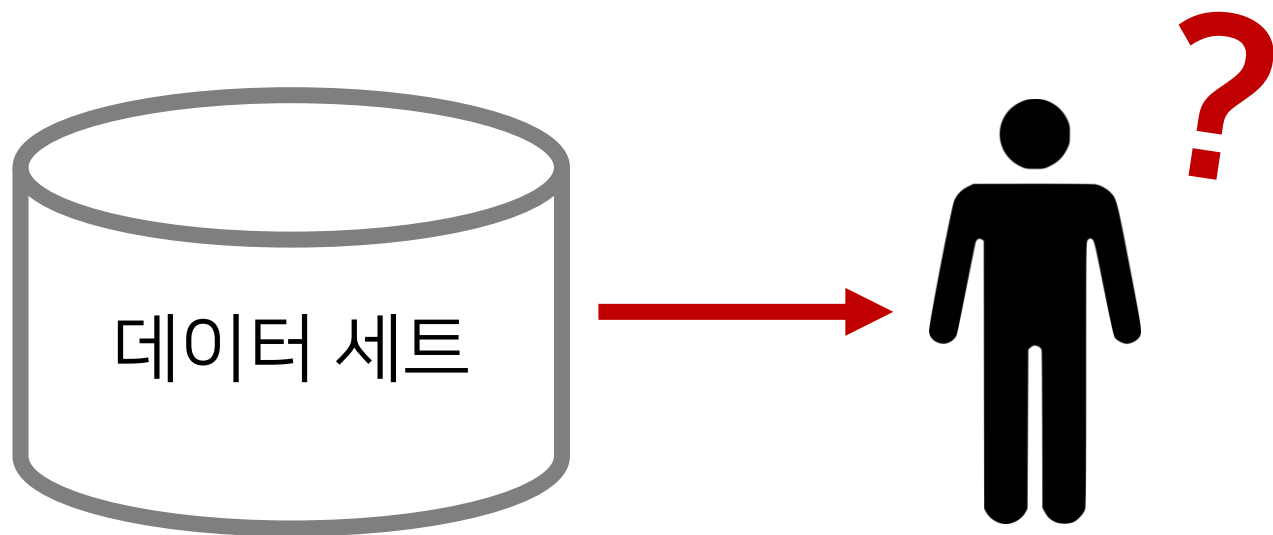


상관 분석(Correlation Analysis)

- 두 변수간의 관계를 수치로 나타낸 것을 상관 계수(correlation coefficient)라고 합니다.
- 일반적으로 상관 분석은 독립 변수 사이에 대하여 수행합니다.
 - 상관 관계 예시: 탑승객의 나이가 많을수록 지불한 티켓 요금이 큰 편입니다.

데이터 분석의 필요성

- 주어진 데이터를 분석하는 단계는, 데이터를 바르게 이해하는 과정으로 필요합니다.
 - 기계 학습 모델을 실제로 학습하기 전에 반드시 선행되어야 하는 작업입니다.



데이터 분석 예시 1) 이상치(Outlier) 분석

- 전반적인 데이터를 확인하여 이상한 값을 필터링한다.
- 예를 들어, 승객 중에서 나이가 130세로 기록된 데이터가 있다고 합시다.
 - 이것은 잘못 기록된 값일까요?
 - 혹은 정말 130세라면, 이것을 학습 데이터로 사용하는 것이 합리적일까요?

데이터 분석 예시 1) 이상치(Outlier) 분석

[예시] 과도하게 큰 값이 있어서, 그것이 전체 데이터의 평균을 올릴 수 있습니다.

- 승객이 5명만 존재하고, 나이가 각각 10, 11, 9, 8, 87세라고 해봅시다.
 - 중간 값(median): 10세
 - 평균 값(average value): 25세
- 하나의 **극단적인 값**으로 인하여 평균 나이는 25세가 됩니다.

데이터 분석 예시 2) 시각화(Visualization)

- 그래프를 이용해 데이터를 시각화하여 쉽게 한 눈에 이해되도록 표현할 수 있습니다.
 - Python 프로그래밍 언어를 사용하면 데이터 시각화를 편리하게 수행할 수 있습니다.
 - 대표적인 라이브러리로 Matplotlib이 사용됩니다.

8강) 타이타닉 생존자 예측 데이터 기초 분석

타이타닉 데이터 분석 과정 1) 데이터를 불러와 바르게 이해하기

- 각 데이터의 특징(feature)에 대한 설명을 제대로 이해할 필요가 있습니다.
- 예를 들어, 타이타닉 호 예시에서는 다음과 같은 속성(property)이 존재합니다.

<타이타닉호에 탄 사람들의 특징들>

- ① 나이(age)
- ② 성별(sex)
- ③ 티켓의 요금(fare)

...

타이타닉 데이터 분석 과정 1) 데이터를 불러와 바르게 이해하기

- 다음과 같은 코드로 학습 데이터를 불러올 수 있습니다.

```
import pandas as pd

train_dataset = pd.read_csv("titanic_train.csv")
train_dataset.head(3)
```

- 다음의 코드를 이용하여 테스트 데이터를 불러올 수 있습니다.

```
test_dataset = pd.read_csv("titanic_test.csv")
test_dataset.head(3)
```

타이타닉 데이터 분석 과정 2) 결측 값 (Missing Value) 확인 및 처리

- 현실의 많은 데이터에는 결측 값이 존재합니다.
- 한 승객의 티켓 요금, 객실 등급은 알지만 나이에 대한 정보가 유실되었다면 어떻게 할까요?
 - 해당 승객의 정보를 학습에서 제외시켜야 할까요?
 - 전체 승객의 평균 나이로 대체하면 될까요?

타이타닉 데이터 분석 과정 2) 결측 값 (Missing Value) 확인 및 처리

- 데이터 중에서 결측 값(missing value)이 많이 존재합니다.
 - 특정한 이유로 유실된 데이터로 이해할 수 있습니다.
 - Cabin (방 호수)의 경우 데이터가 존재하지 않아 NaN으로 표시된 경우가 있습니다.
- 이때 info() 메서드를 이용하여 데이터를 전체적으로 이해할 수 있습니다.
- info() 메서드: 특정한 데이터프레임(data-frame)의 정보를 출력합니다.

<info() 메서드>

- ① 열(column)의 개수
- ② 열(column)의 데이터 타입
- ③ 존재하지 않는 데이터(NULL)의 개수
- ④ 데이터의 크기(메모리 크기)

타이타닉 데이터 분석 과정 2) 결측 값 (Missing Value) 확인 및 처리

- 데이터 확인 결과 나이(age)와 호실 정보(cabin)에 대한 정보가 많이 유실되었습니다.
- [추측] 나이(age)의 경우, 생존 여부와 상당히 큰 연관성이 있을 것입니다.
 - 그렇다면 나이 정보가 없는 탑승객에 대하여 어떻게 처리할 수 있을까요?

타이타닉 데이터 분석 과정 3) 생존자/사망자 데이터 분석

- head() 메서드를 사용하여 생존자 데이터를 일부 출력합니다.

```
train_dataset[train_dataset["Survived"] == 1].head(10)
```

index	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1	C123	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.55	C103	S
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	248706	16.0	NaN	S
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0	NaN	S
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.225	NaN	C

타이타닉 데이터 분석 과정 3) 생존자/사망자 데이터 분석

- head() 메서드를 사용하여 사망자 데이터를 일부 출력합니다.

```
train_dataset[train_dataset["Survived"] == 0].head(10)
```

index	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.075	NaN	S
12	13	0	3	Saunderscock, Mr. William Henry	male	20.0	0	0	A/5. 2151	8.05	NaN	S
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.275	NaN	S
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406	7.8542	NaN	S
16	17	0	3	Rice, Master. Eugene	male	2.0	4	1	382652	29.125	NaN	Q
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31.0	1	0	345763	18.0	NaN	S

타이타닉 데이터 분석 과정 3) 생존자/사망자 데이터 분석

- 생존자/사망자에 따른 표본 10개씩을 보았을 때, 어떤 정보를 확인할 수 있나요?
 1. 여성이 남성보다 상대적으로 많이 생존했습니다.
 2. 1등석에 탑승한 사람이, 3등석에 탑승한 사람보다 많이 생존했습니다.

타이타닉 데이터 분석 과정 4) 성별에 따른 생존자 수 분석

- `value_counts()`는 특정한 컬럼에서 고유 값(unique value)마다 출현 빈도를 계산합니다.
- 결과로는 하나의 **시리즈(series)**가 도출됩니다.
 - 인덱스(index): 고유 값
 - 값(value): 값의 개수
- 성별에 따른 생존자 수를 출력해 봅시다.
- 여성의 생존자 수가 많은 것을 확인할 수 있습니다.

```
survived = train_dataset[train_dataset["Survived"] == 1]["Sex"].value_counts()  
print(survived)
```

타이타닉 데이터 분석 과정 5) 생존자/사망자 통계 함수 만들기

- 생존자 통계를 살펴보기 위하여, 함수를 작성할 수 있습니다.

```
def show(feature):  
    # 특징(feature)에 따른 생존자(survived) 수를 나타내는 컬럼  
    survived = train_dataset[train_dataset["Survived"] == 1][feature].value_counts()  
    # 특징(feature)에 따른 사망자(dead) 수를 나타내는 컬럼  
    dead = train_dataset[train_dataset["Survived"] == 0][feature].value_counts()  
    # 두 컬럼을 묶어서 데이터프레임(dataframe)으로 생성  
    df = pd.DataFrame([survived, dead])  
    df.index = ["Survived", "Dead"]  
    print(df)  
    df.plot(kind="bar", stacked=True)
```

타이타닉 데이터 분석 과정 6) 성별에 따른 생존 확률 확인

- 이제 성별에 따른 생존율을 확인할 수 있습니다.
 - 여성이 남성에 비하여 생존율이 높은 것을 확인할 수 있습니다.
 - 남성의 생존율은 $109/577 = 18.89\%$ 입니다.
 - 여성의 생존율은 $233/314 = 74.20\%$ 입니다.

```
show("Sex")
```

타이타닉 데이터 분석 과정 7) 탑승권의 등급에 따른 생존율 확인

- 탑승권의 등급에 따른 생존율을 확인할 수 있습니다.
 - **좌석의 등급이 높을수록**, 생존율이 높은 것을 확인할 수 있습니다.

```
show("Pclass")
```


9강) 캐글(Kaggle) 접속 및 정답 제출 방법

캐글(Kaggle) 개요

- 캐글(Kaggle)이란, 기계 학습 대회 플랫폼입니다.
- 일반적으로 기업/기관에서 데이터 및 과제를 등록하여 상금을 제시합니다.
- 참가자(학생 혹은 과학자)는 해당 과제에서 높은 정확도를 보이는 모델을 만들어 상금을 받습니다.



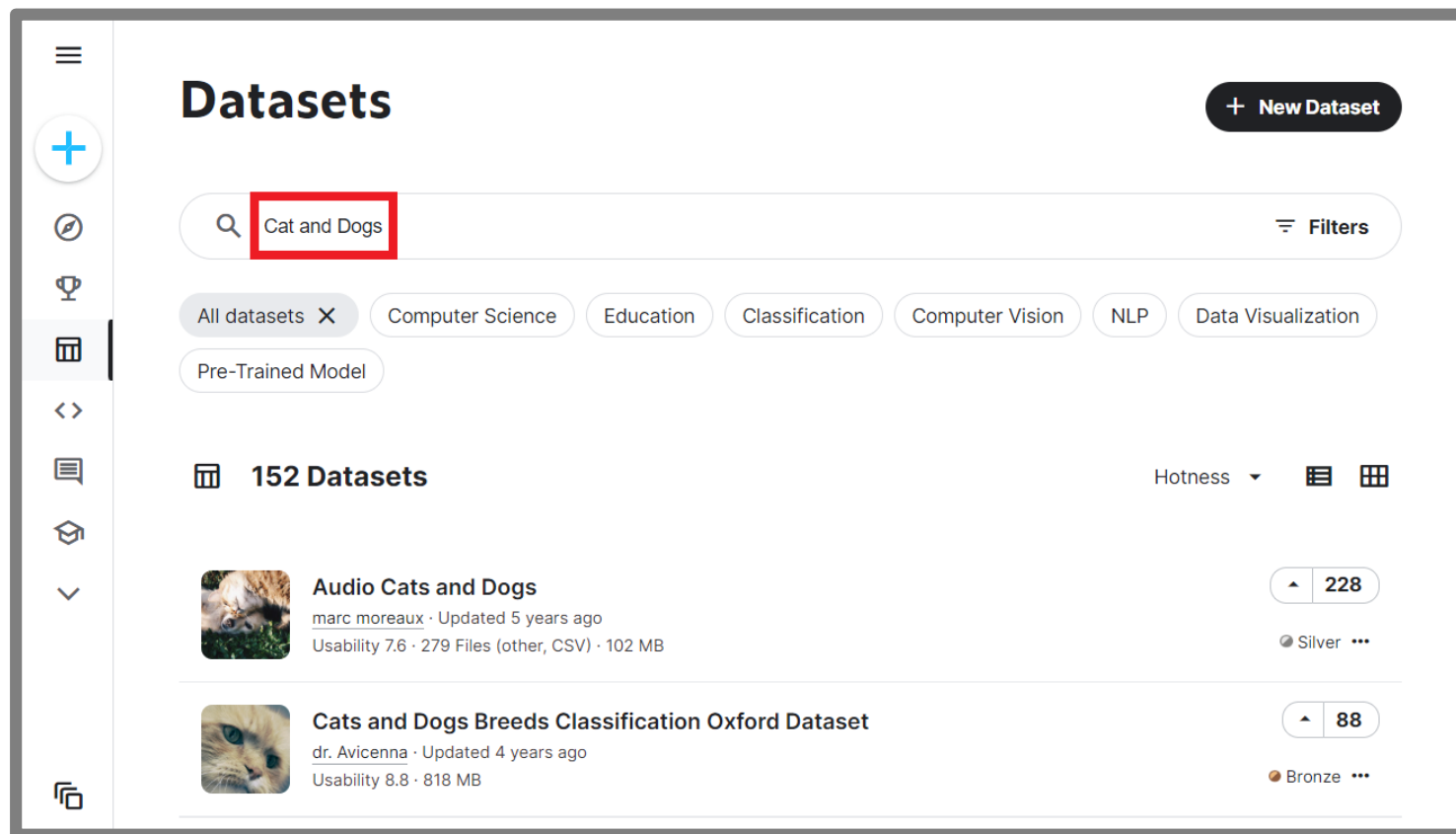
* <https://www.kaggle.com/>

캐글(Kaggle) 이용하는 방법

- 따라서, 캐글은 기계 학습을 공부하는 학생 및 연구자에게 매우 유용한 사이트입니다.
 1. 캐글에 많은 유용한 데이터가 있으므로, 원하는 데이터를 검색해 수집할 수 있습니다.
 2. 캐글에는 다양한 태스크(task)에서 사용할 수 있는 많은 유용한 소스 코드가 있습니다.
 3. 다양한 프레임워크(Scikit-learn, PyTorch, TensorFlow 등)의 코드가 공유되어 있습니다.

캐글(Kaggle)에서 데이터 세트 찾기

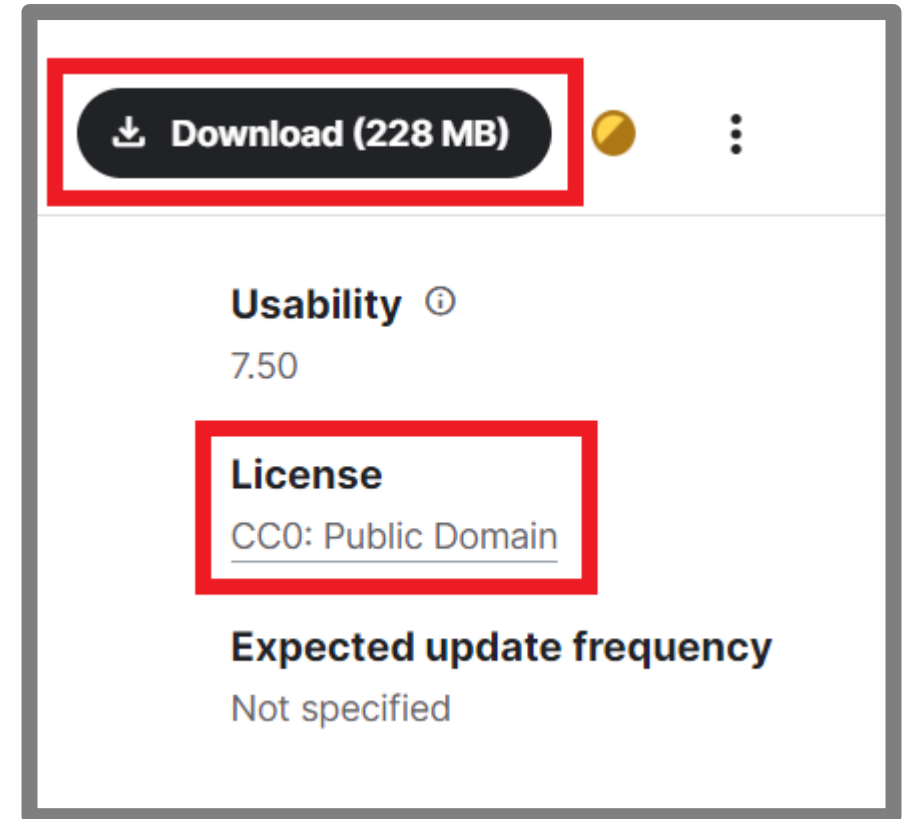
- 캐글에서는 다양한 데이터를 공유할 수 있습니다.
- 예를 들어, 고양이와 강아지 데이터 세트를 찾을 때는 다음과 같이 검색할 수 있습니다.



* <https://www.kaggle.com/datasets>

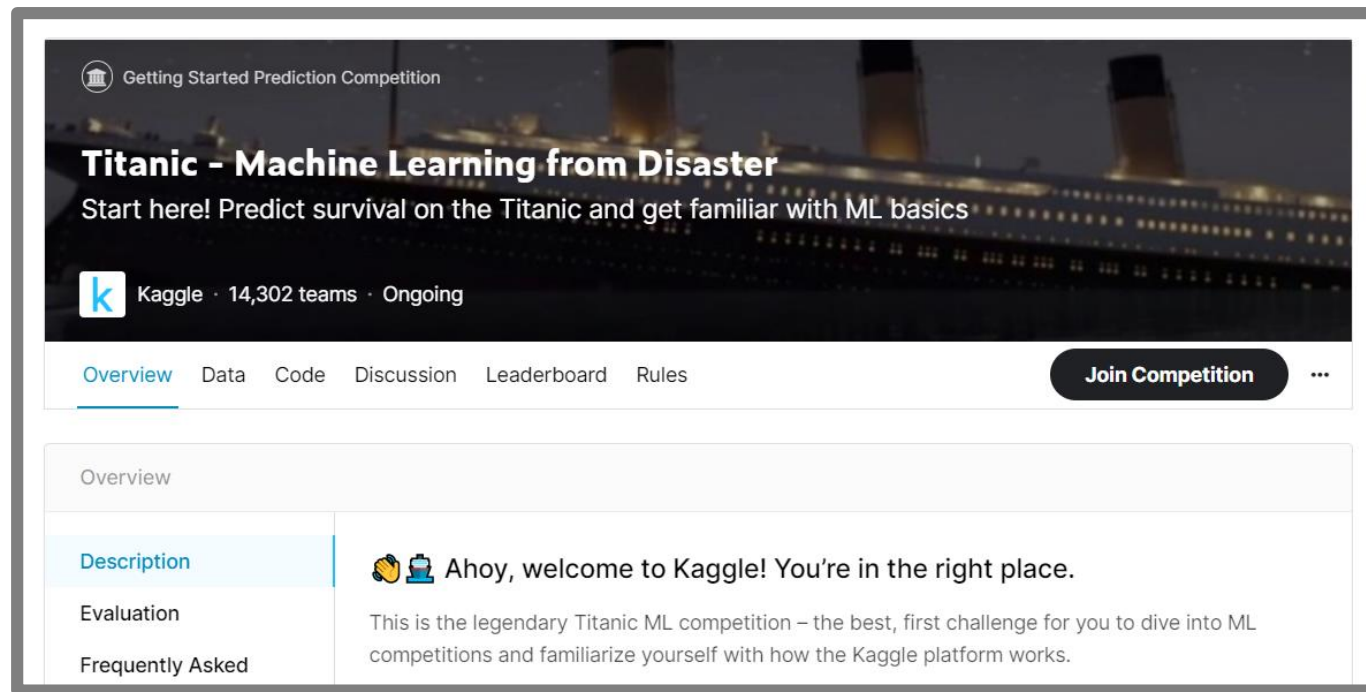
캐글(Kaggle)에서 원하는 데이터 다운로드하기

- 회원가입 이후에는 원하는 데이터 세트를 다운로드할 수 있습니다.
- 데이터 사용시 라이선스(license)를 확인해야 합니다.
- 예시) CC0은 상업적으로 사용 가능한 라이선스입니다.



캐글(Kaggle)에서 타이타닉 문제 확인하기

- 캐글에서 타이타닉 문제와 데이터를 확인할 수 있습니다.
- 가장 중요한 것 중 하나는 제출 형식(submission format)입니다.
- 테스트 데이터가 공개되어 있으며, 정답을 채워서 엑셀 파일(.csv)로 제출합니다.



* <https://www.kaggle.com/competitions/titanic>

제출(submission) 파일 만들기 - 성별에 따른 생존 여부 예측

- 단순히 성별에 따라서 생존율을 예측할 수 있습니다.
 - **성별에 따라서 생존/사망 여부를 기입**하는 코드는 다음과 같습니다.

```
pred = test_dataset["Sex"] == "female"
pred[pred == True] = 1 # 여성인 경우 생존(1)
pred[pred == False] = 0 # 남성인 경우 사망(0)
pred = pred.astype("int32")
print(pred)
```

제출(submission) 파일 만들기 - 성별에 따른 생존 여부 예측

- 각 승객 번호(passenger id)에 따른 생존 여부를 기입해야 합니다.
 - 문제에서 요구하는 제출 형식(submission format)을 정확히 따를 필요가 있습니다.

```
# 제출용 엑셀 파일(.csv) 생성
submission = pd.DataFrame({'PassengerId': test_dataset['PassengerId'], 'Survived': pred})
submission.to_csv('submission.csv', index=False)
print(submission)
```



제출(submission) 파일 만들기

- 생성된 **제출 파일(submission.csv)**은 다음과 같습니다.

	A	B
1	PassengerId	Survived
2	892	0
3	893	1
4	894	0
5	895	0
6	896	1
7	897	0
8	898	1

캐글에 제출하여 순위 확인하기

- 캐글(Kaggle)에 제출하면, 다음과 같은 결과를 얻을 수 있습니다.
 - 더욱 높은 점수를 얻기 위해서는 어떻게 해야 할까요?
 - 단순히 성별(sex) 말고 **다른 유용한 특징(feature)**을 함께 **활용**해 봅시다.

Submission and Description		Public Score ⓘ
	submission.csv Complete · 1s ago	0.76555

10강) 타이타닉 데이터 세트 특징 공학

특징 공학(Feature Engineering) 과정

- 모델 학습을 진행하기 전에, 데이터 전처리 과정은 필수적입니다.
- **특징 공학(feature engineering)**을 진행합니다.
 1. 불필요한 특징(feature) 제거하기
 2. 문자열 형태의 데이터를 수 데이터로 바꾸어 주기
 3. 결측 값(missing value) 처리하기
 4. 이상치(outlier) 처리하기
 5. 각 특징 값의 범위(range)를 유사하게 맞추어 주기

특징 공학 1) 데이터 불러오기

- 특징 공학을 위해 기존 타이타닉 데이터 세트를 다시 불러올 수 있습니다.

```
import pandas as pd

train_dataset = pd.read_csv("titanic_train.csv")
test_dataset = pd.read_csv("titanic_test.csv")
train_dataset.head(10)
```

특징 공학 2) 불필요한 특징(Feature) 제거하기

- 유용한 정보를 얻기 어려운 특징이라면, 특징 제거를 수행합니다.

1. 티켓의 번호(ticket)
2. 방의 번호(cabin)
3. 이름 정보(name)

[유의 사항] 이름 데이터에는 사모님(Mrs)과 같은 호칭이 들어가 있습니다.

- 나이(age) 정보와 연관성이 있지만, 본 과정에서는 편의를 위해 단순 삭제하겠습니다.
- 참고로 학습(train)과 테스트(test) 데이터 세트 모두에서 불필요한 특징을 제거해야 합니다.

특징 공학 2) 불필요한 특징(Feature) 제거하기

- 유용한 정보를 얻기 어려운 특징이라면, 특징 제거를 수행합니다.
 - *inplace*의 값이 *True*면 원본 데이터 프레임의 값이 직접 수정됩니다.

```
train_dataset.drop("Ticket", axis=1, inplace=True)
train_dataset.drop("Cabin", axis=1, inplace=True)
train_dataset.drop("Name", axis=1, inplace=True)
train_dataset.head()
```

- 이후에 다음과 같이 처리할 수 있습니다.

```
test_dataset.drop("Ticket", axis=1, inplace=True)
test_dataset.drop("Cabin", axis=1, inplace=True)
test_dataset.drop("Name", axis=1, inplace=True)
test_dataset.head()
```