

Introduction to Machine Learning

세종대학교 AI로봇학과 류승형

강사 소개

- 주요연구분야
 - 에너지 및 산업인공지능 / 시계열 예측 / 이상탐지 / 증강지능
- 경력
 - 2020~2023 / 한국원자력연구원 인공지능응용연구실
- 연락처
 - Office : 02-3408-3366
 - E-mail : shryu@sejong.ac.kr
 - Homepage : ideallab.oopy.io
- 류승형 교수 / IDEAL lab.
- 세종대학교 AI로봇학과

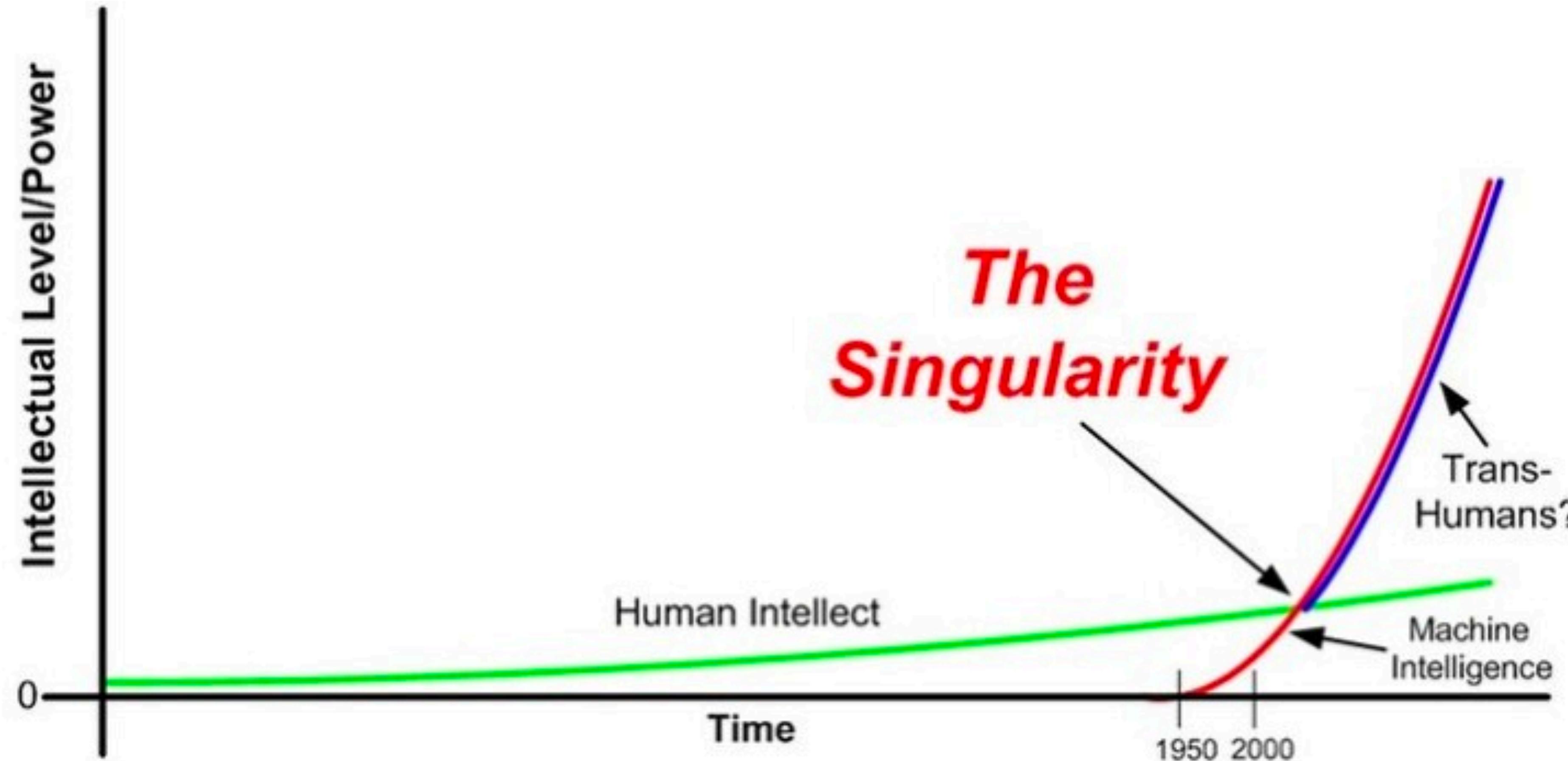


목차

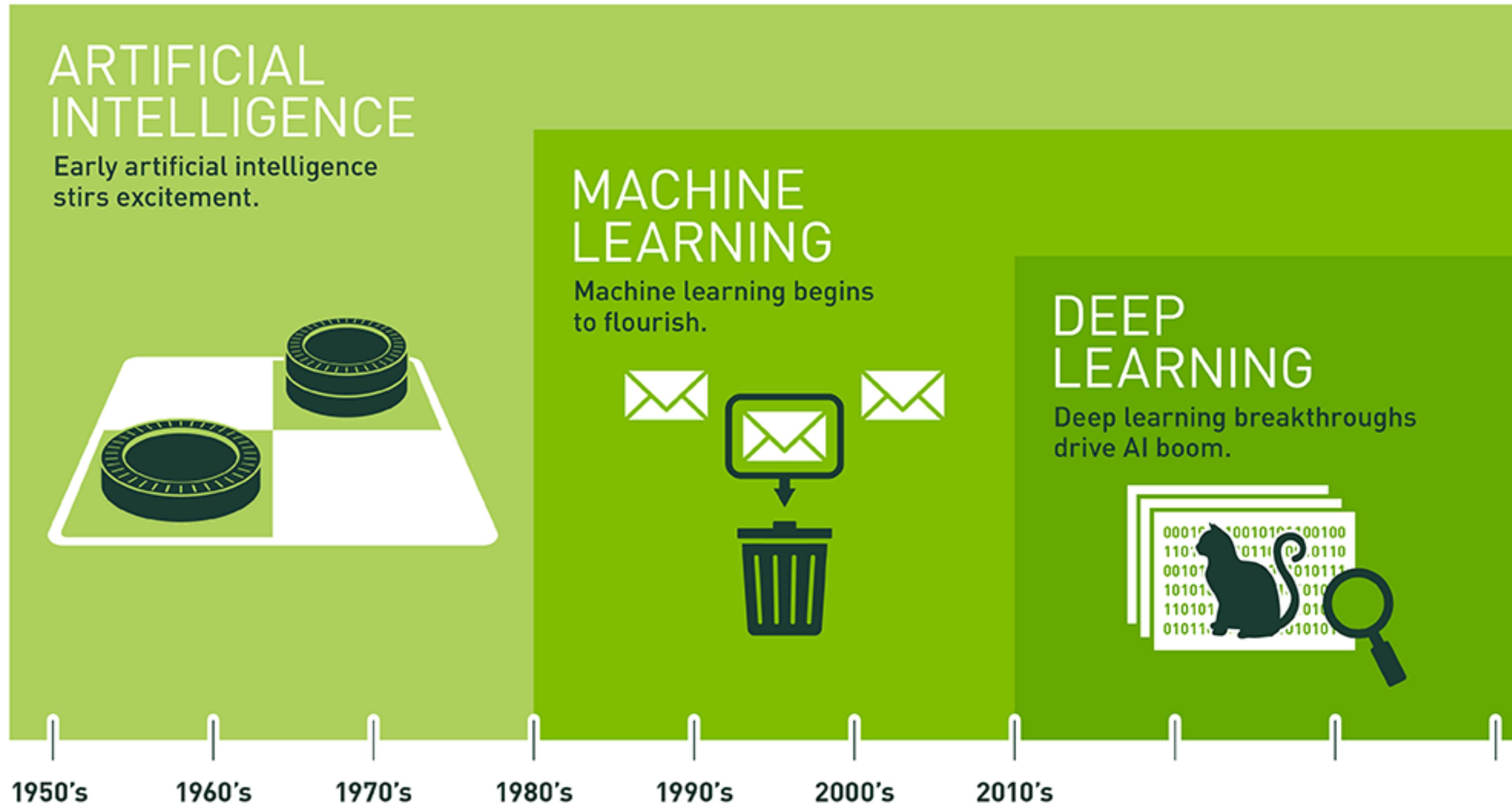
- Chapter 1. Machine learning (ML)
- Chapter 2. Process in ML model development
- Chapter 3. Data, task and model

Chapter 1. Machine Learning

The Singularity



Machine learning



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

1958 in the New York Times

- A new device learns by doing
- The embryo of an electronic computer that expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.
- "걸을 수 있고, 말하고, 볼 수 있으며, 쓸 수 있고, 본인을 복제하며, 자신의 존재를 인식할 수 있는 전자 컴퓨터의 초기 모델입니다."

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's \$2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human be-

ings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls
The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

1958 New York Times...

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

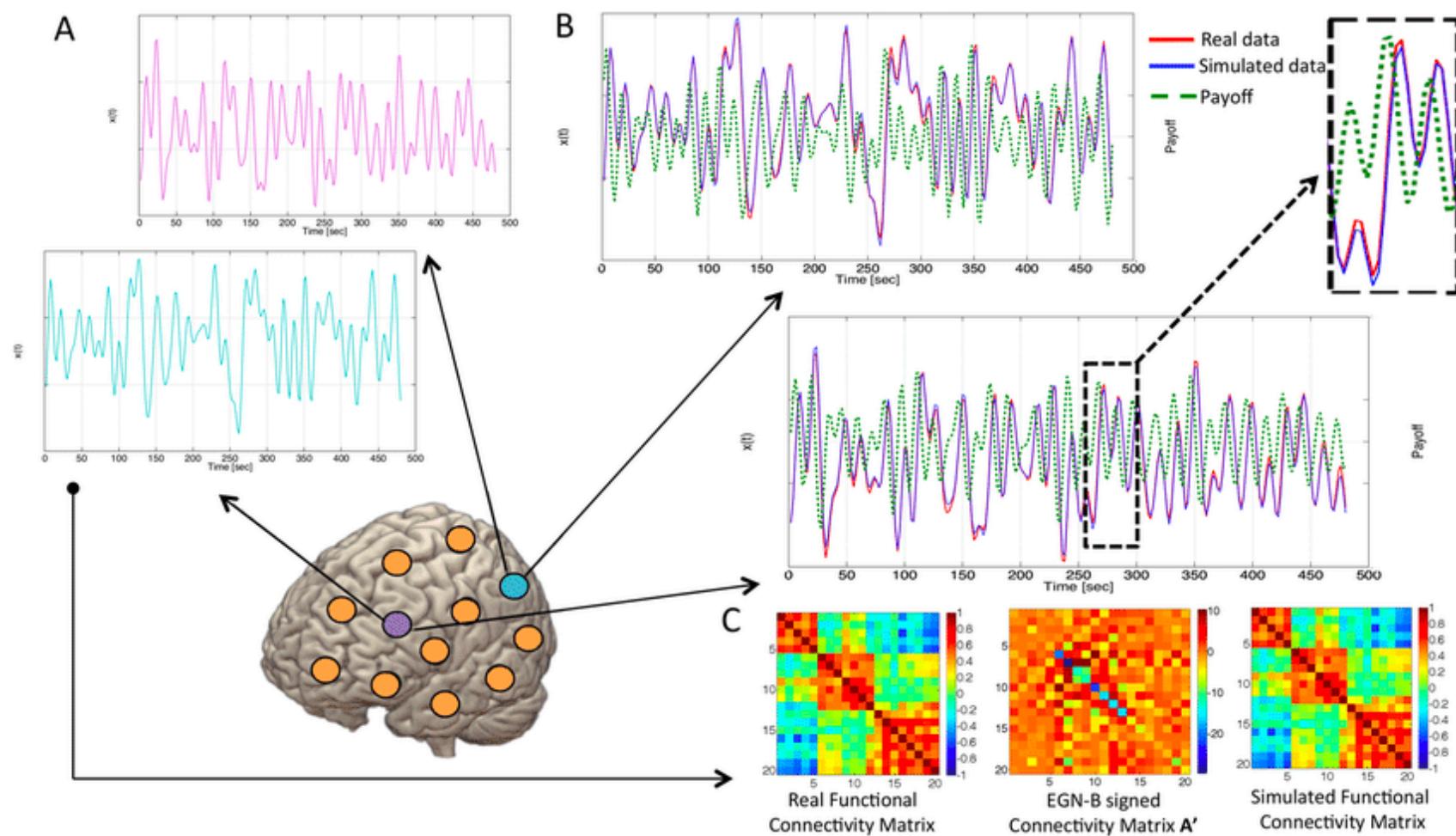
In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

인공 지능을 위한 첫 단추는?

- 인간의 지능은 무엇일까?
- 인간의 두뇌 안에서 일어나는 전기적 신호?
- 신호의 의미를 전부 이해할 수 있을까?



사회

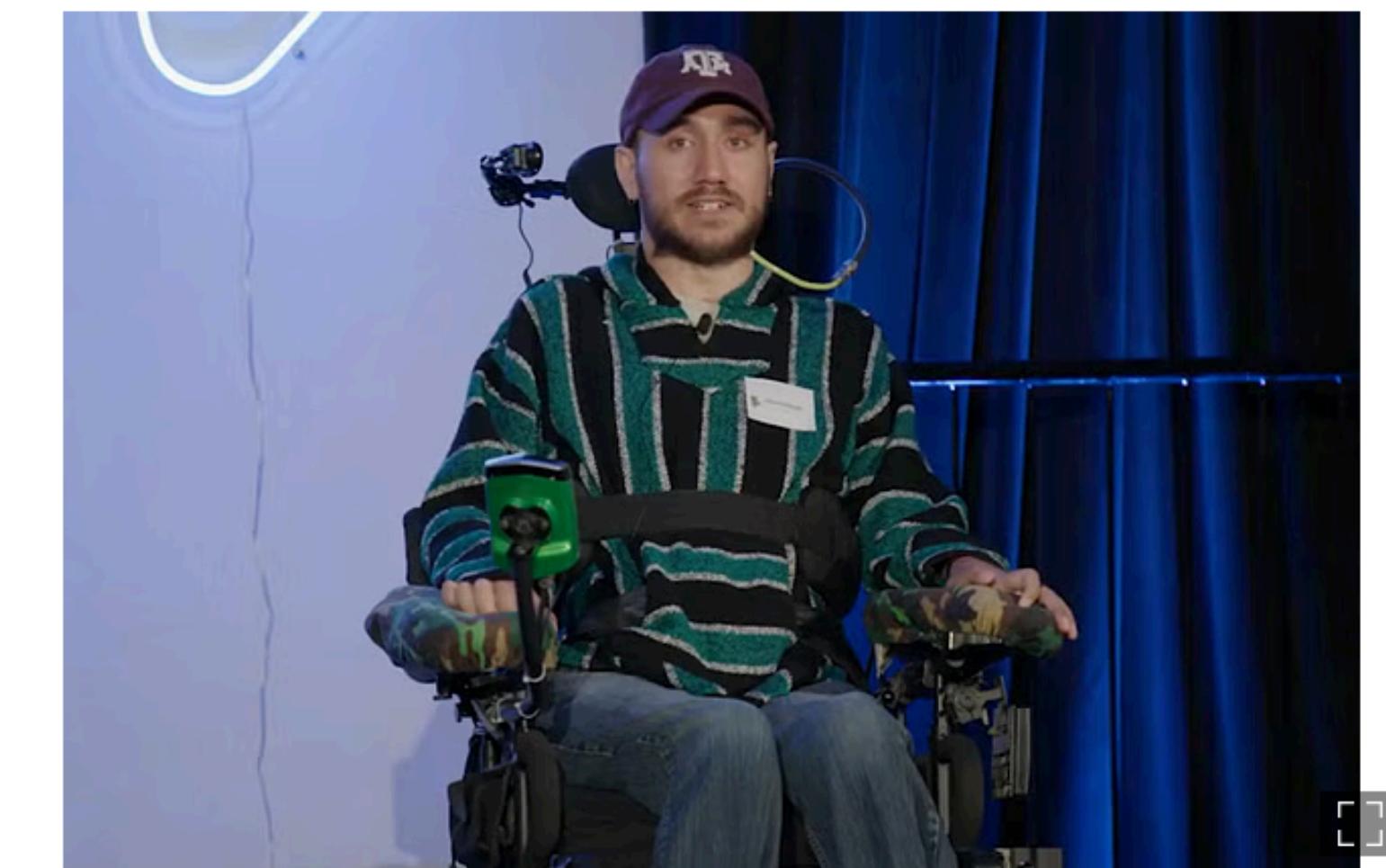
“초능력 가진 것 같아, 인생 바뀌었다”...세계 최초로 뇌에 뉴럴링크 이식한 남자

이덕주 기자 mrdjlee@mk.co.kr

입력 : 2024-05-17 13:44:52 수정 : 2024-05-17 14:58:24

¶ 가 □ ☰ ☱

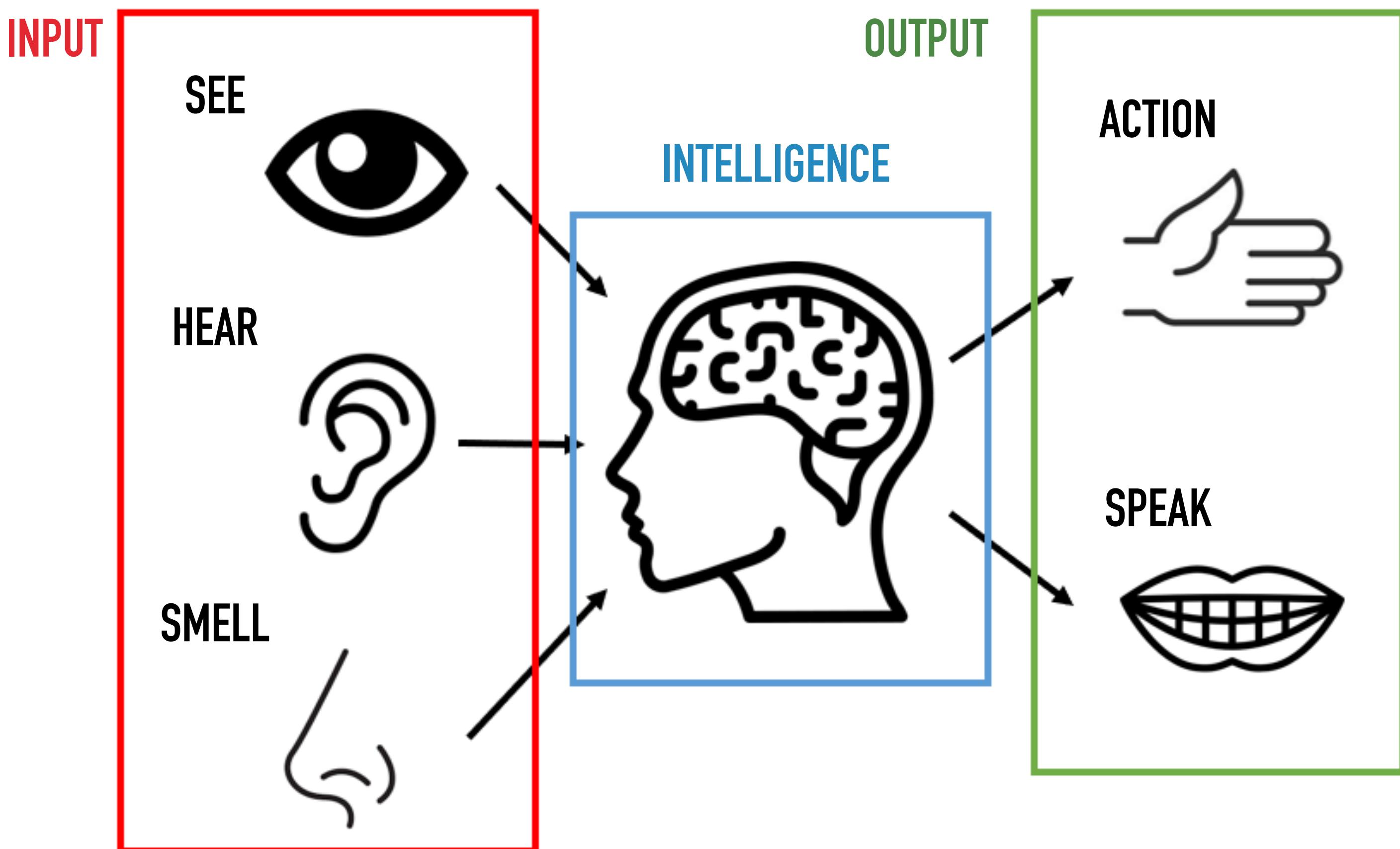
대학생때 다이빙 사고로 전신마비
올해 1월 뇌에 뉴럴링크 이식
“일론 머스크가 수술 후 찾아와”
생각하는 대로 컴퓨터 작동가능



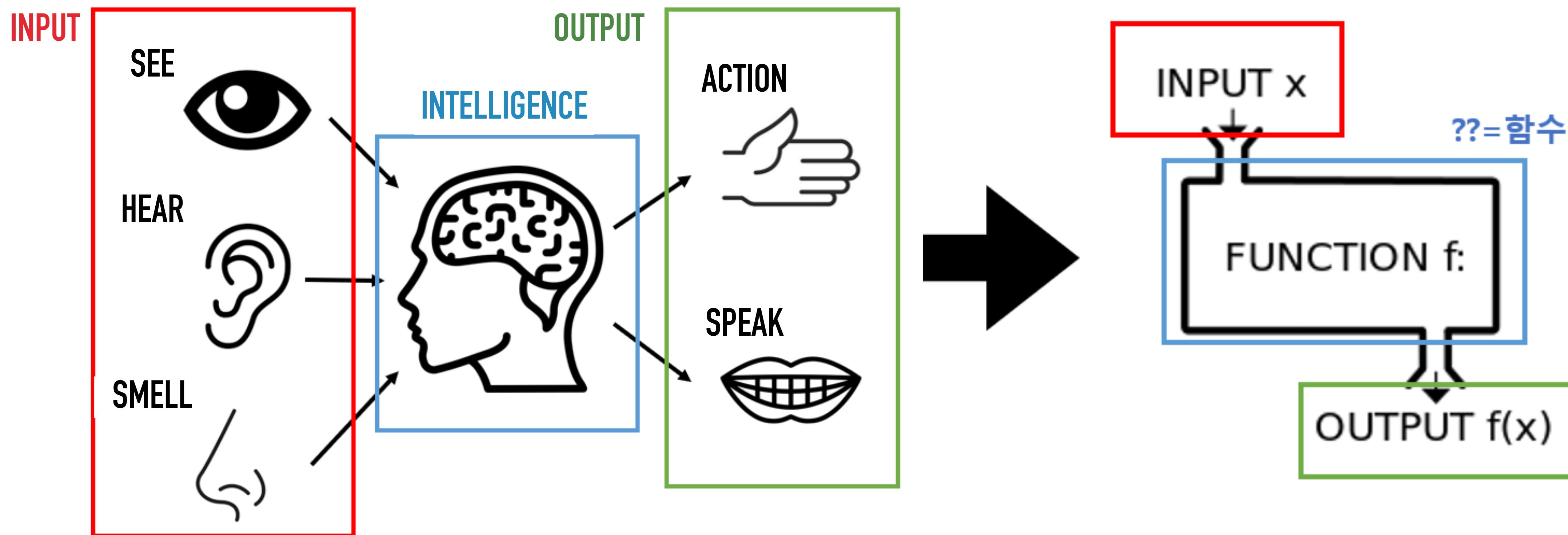
최초의 뉴럴링크 칩 이식자인 놀런 아르보가 뉴럴링크를 방문해 직원들에게 이야기하고 있다. <사진=뉴럴링크>

인간의 사고 과정

- 감각기관을 통한 인지
→ Inputs
- 지성을 통한 사고
- 사고의 결과를 바탕으로 행동
→ Outputs



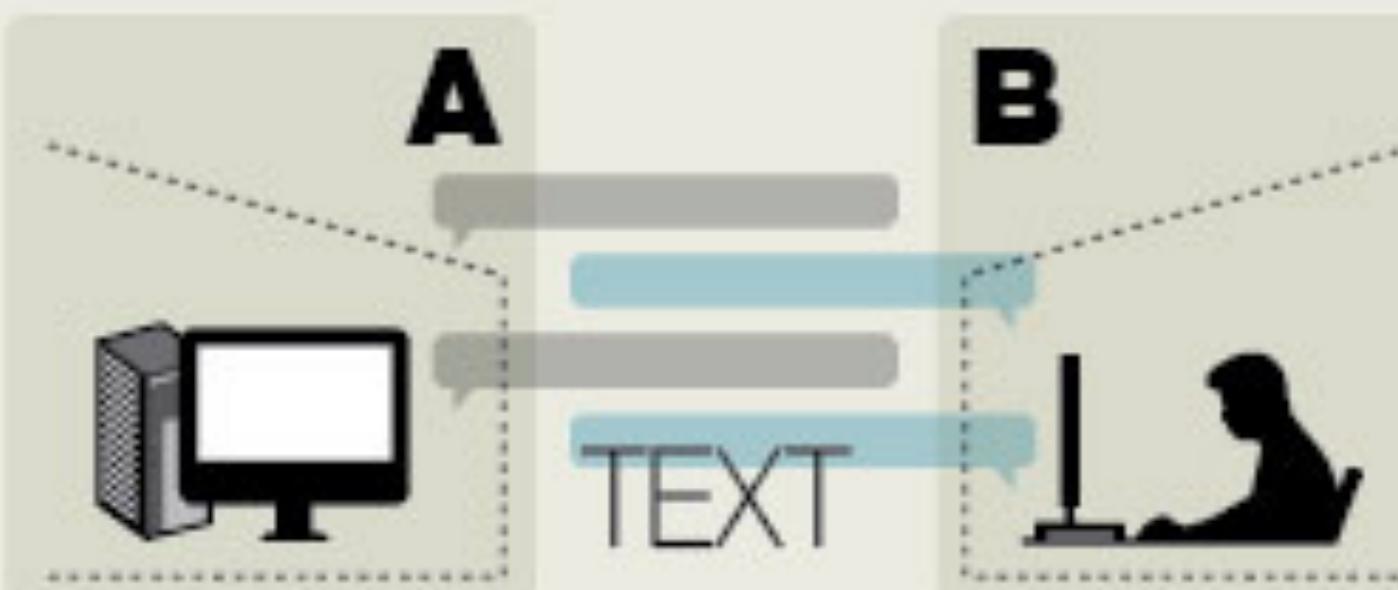
인간의 사고 과정 = 하나의 함수



어떻게 인간의 지능을 모방한 함수를 만들 수 있을까?

인공지능 판별기준 튜링 테스트 개요

- ① 컴퓨터 화면을 통해 문자로만 대화



기계(실험대상)

사람(대조 샘플)

→ ② A,B 모두 사람이라고 주장 ←



심사위원(제3자)

- ③ 어느 쪽이 사람인지 구분할 수 없을 경우,
A·B 둘다 인간 수준의 사고능력을 가진 것으로 판정

연합뉴스

이재윤 기자 / 20140609

@yonhap_graphics(트위터)

YONHAPNEWS

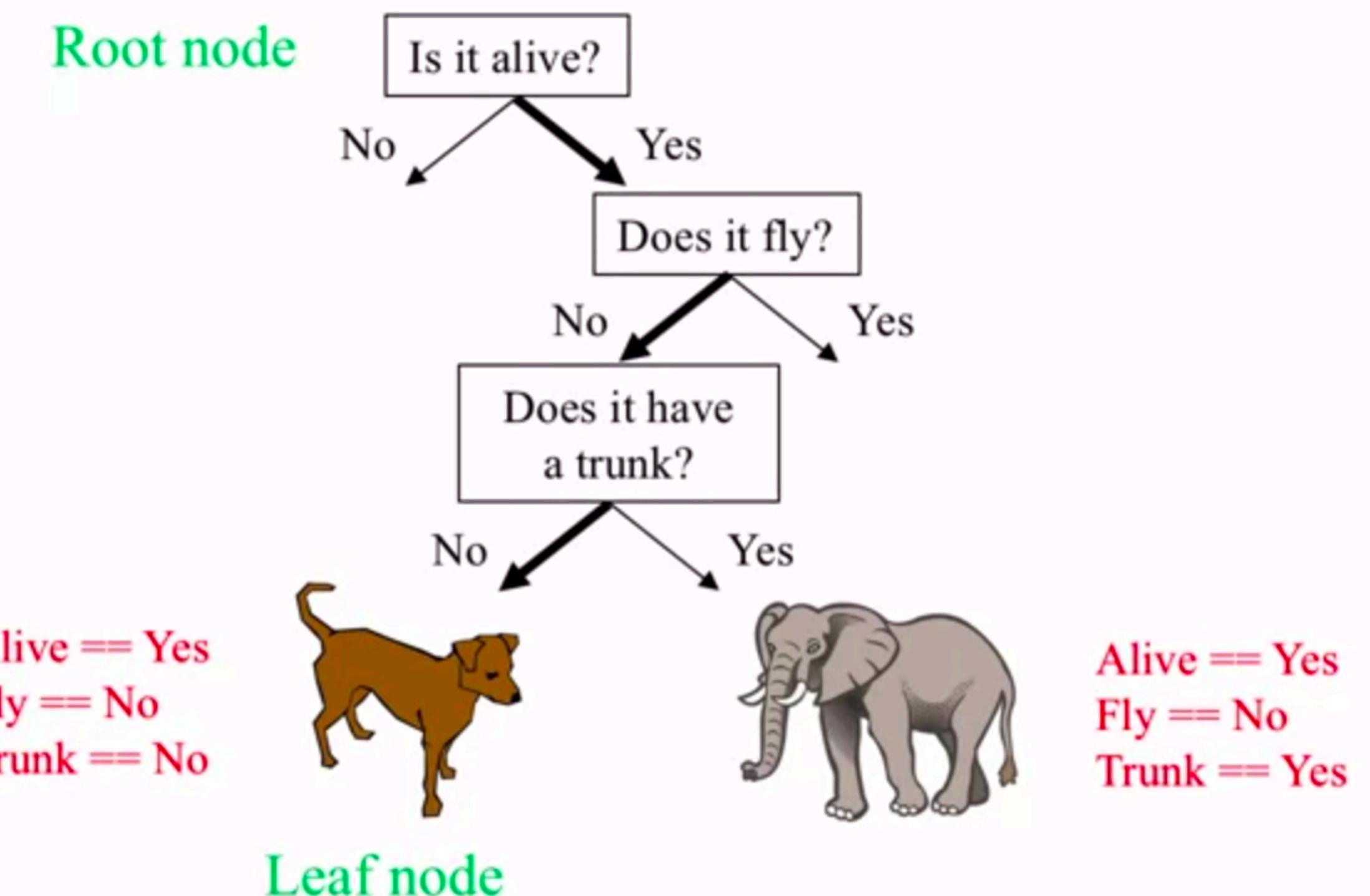
어떻게 인간의 지능을 모방한 함수를 만들 수 있을까?

- 중요한 것은 입력에 대한 출력 → 지능/사고의 결과물로서 출력을 얻어야 함.
- 가장 쉽게 생각할 수 있는 방법은?
 - If, then, else.
 - 조건에 따라 정해진 행동을 수행

Rule-based approach

- 규칙 기반 모델 또는 전문가 시스템
- 직관적임, 하지만 ...?
 - 새로운 입력 또는 질문에 대한 반응은?
 - 단순한 일이 아닌 복잡한 문제는?
- 룰을 결정하는 것이 쉽지 않다...

Decision Tree Example



다른 방법을 생각해 봅시다.

Q. 사람은 어떻게 특정 분야의 전문가가 될 수 있을까요?

다른 방법을 생각해 봅시다.

Q. 사람은 어떻게 특정 분야의 전문가가 될 수 있을까요?

A. 공부, 학습을 통해서 전문가가 됩니다.

기계(프로그램)를 학습시키자!

Q. 그렇다면 기계는 무엇을 배워야 하나요?

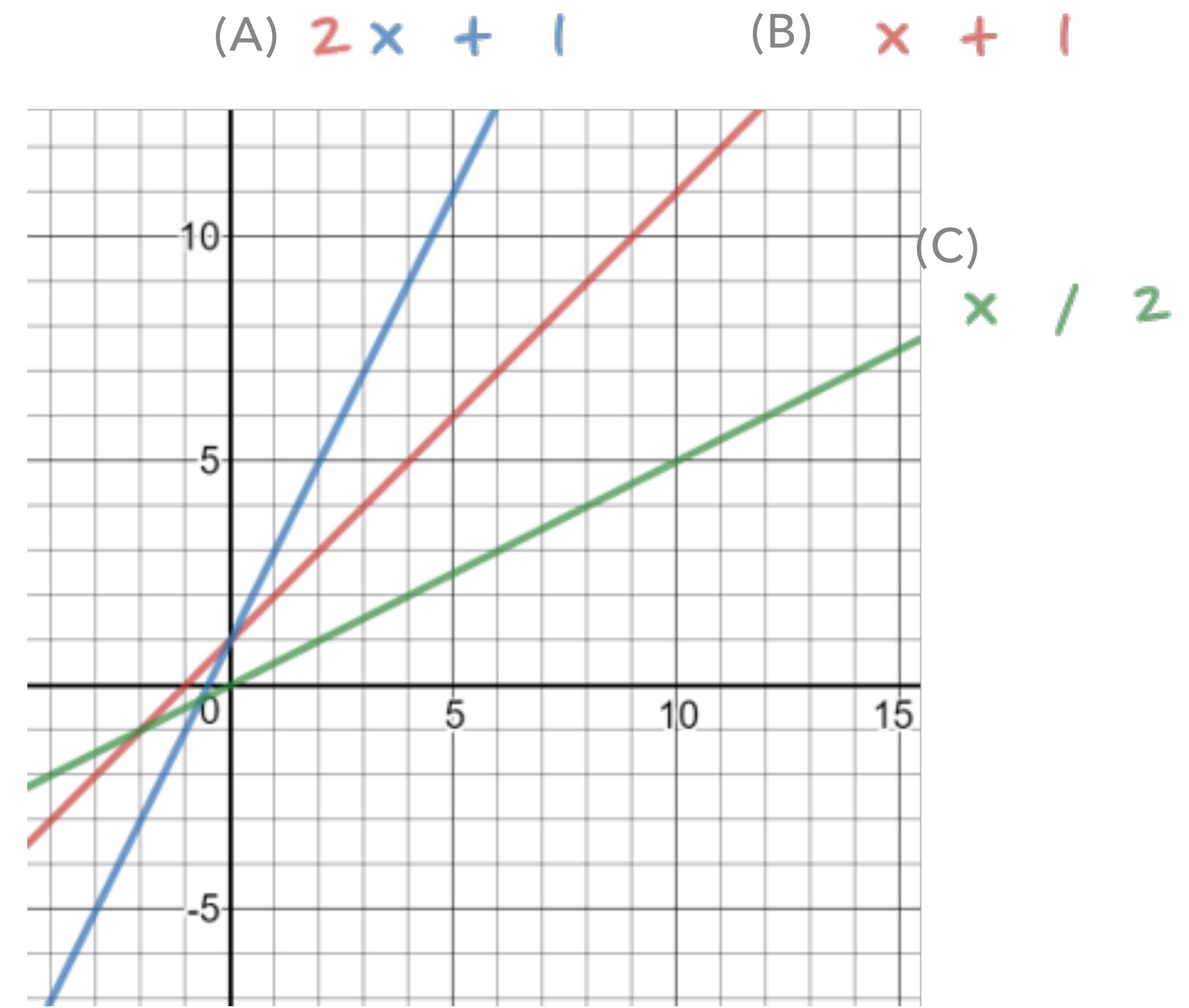
기계(프로그램)를 학습시키자!

Q. 그렇다면 기계는 무엇을 배워야 하나요?

A. 인간 지능의 출력과 동일/유사한 결과를 출력하는 함수

함수를 학습한다는 것의 의미

- $y = ax + b$
 - $a = 2, b = 1$
 - $a = 1, b = 1$
 - $a = 1/2, b = 0$
- $y = ax^2 + b\sin(x) + c$
- $y = f(x; w)$



Example of simple linear graph

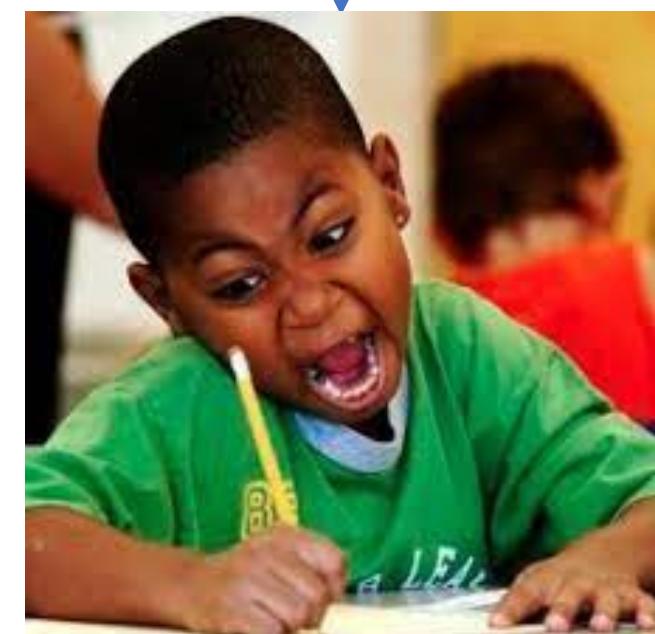
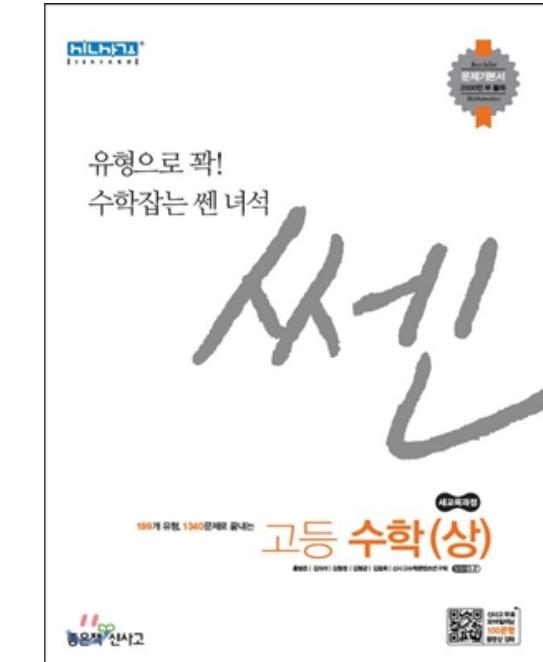
함수를 학습한다는 것의 의미는?

- 함수의 형태를 결정하는 w (웨이트, 파라미터, 계수)를 학습을 통해 찾는 것.
- 어떤 w 가 필요한가요?
 - 주어진 태스크에 대해 적합한 결과를 내는 w .
- 기계 학습 (Machine learning, ML)에서는...
 - ML 모델이 w 를 학습할 수 있도록 판을 깔아줌 → 이후에는 반복을 통해 태스크에 대한 성능 향상
 - ML 모델에 따라서 학습이 되는 w 의 형태는 달라질 수 있음.
 - 결정트리의 경우 분할 규칙을 학습, 인공신경망에서는 레이어의 가중치 등

인간 학습 프로세스

- 정답이 있는 문제지
- 문제풀기
- 채점
- 복습
- 위 과정을 반복한다.

Workbook

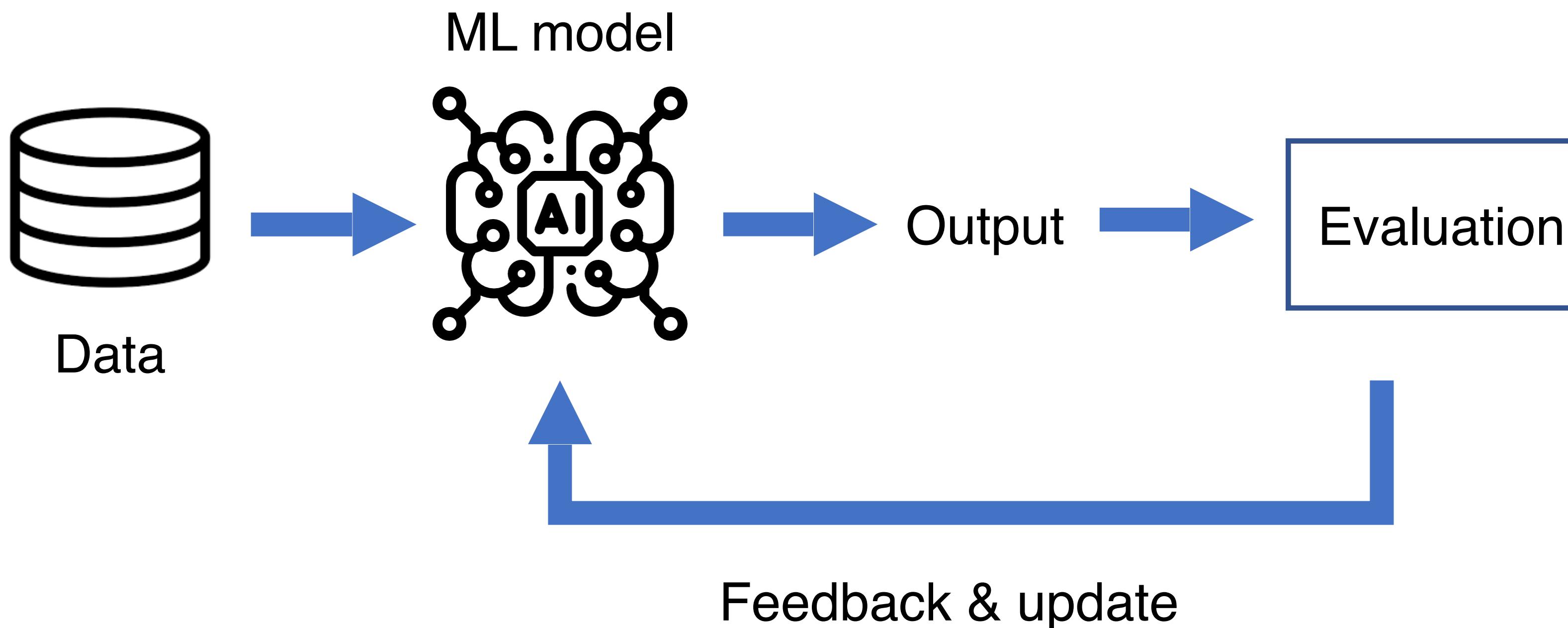


Result

Evaluation

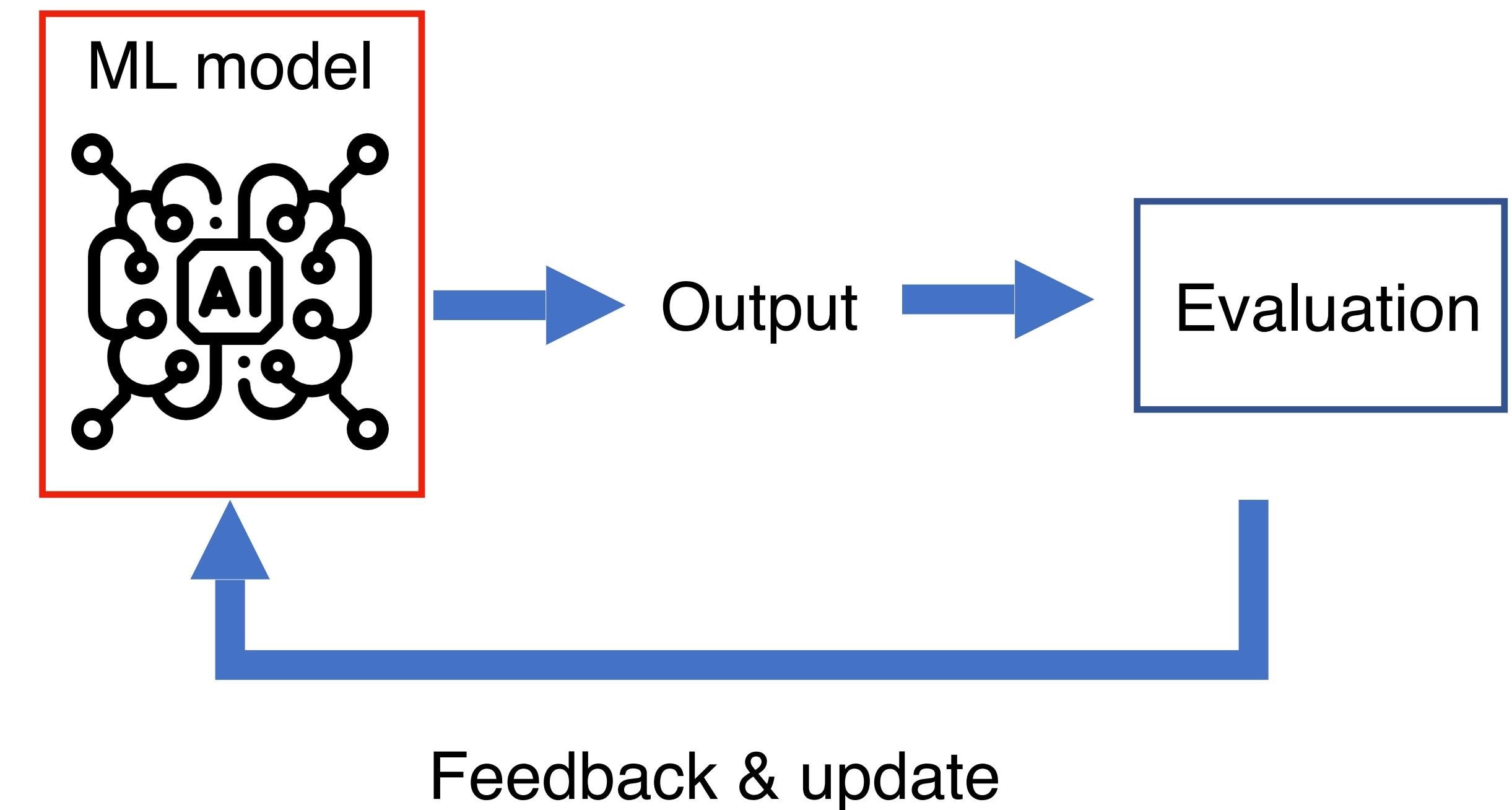
Review

기계 학습 프로세스



어떤 모델을 써야 할까?

- $y = ax + b$ or $y = a \exp(x)$
- 이보다 더 복잡한 형태의 수식?
- 아니면 다른 알고리즘?



사람은 어떤가요?

- 인간 학습 프레임워크에서는 인간의 두뇌가 ML모델의 역할을 수행함.

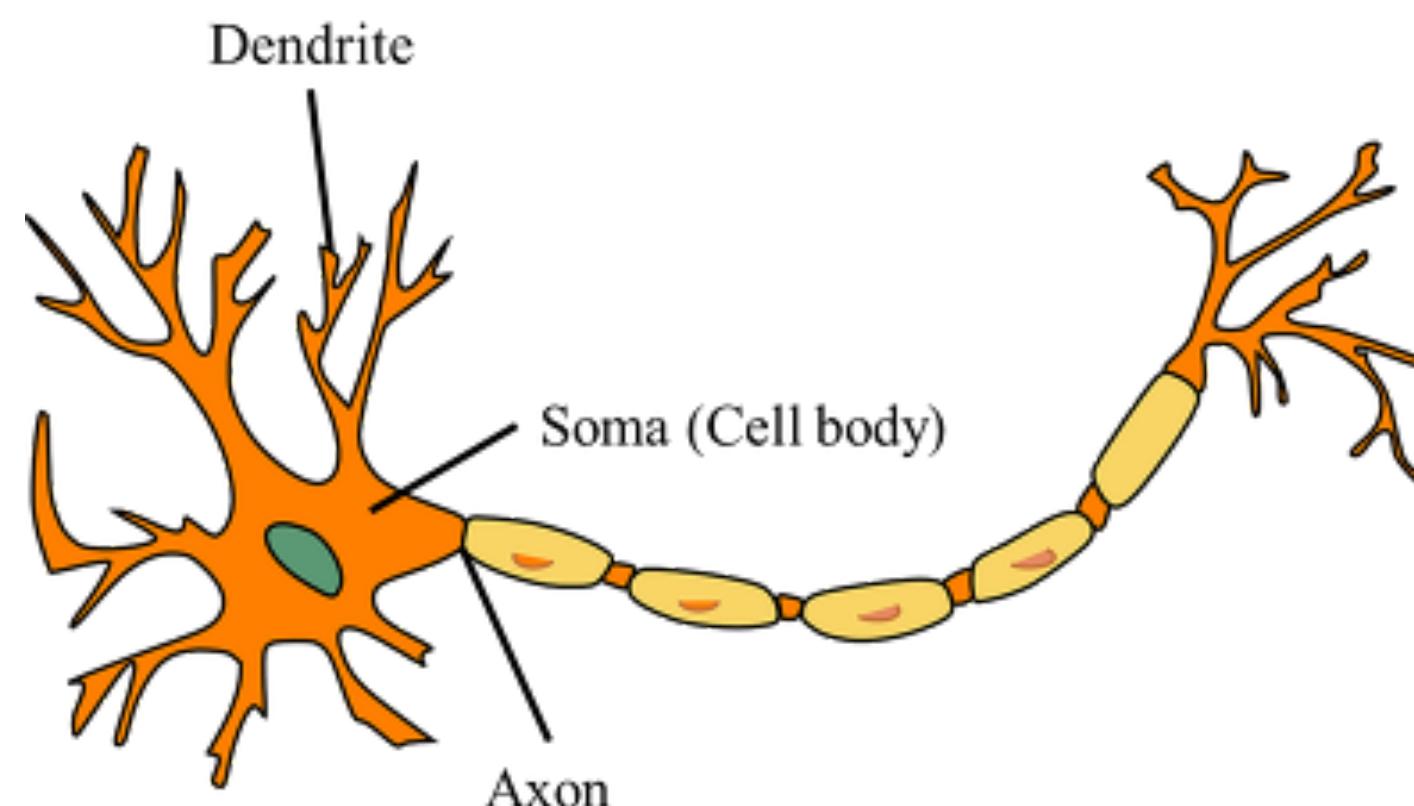


인간의 두뇌는 뉴런들의 거대 네트워크

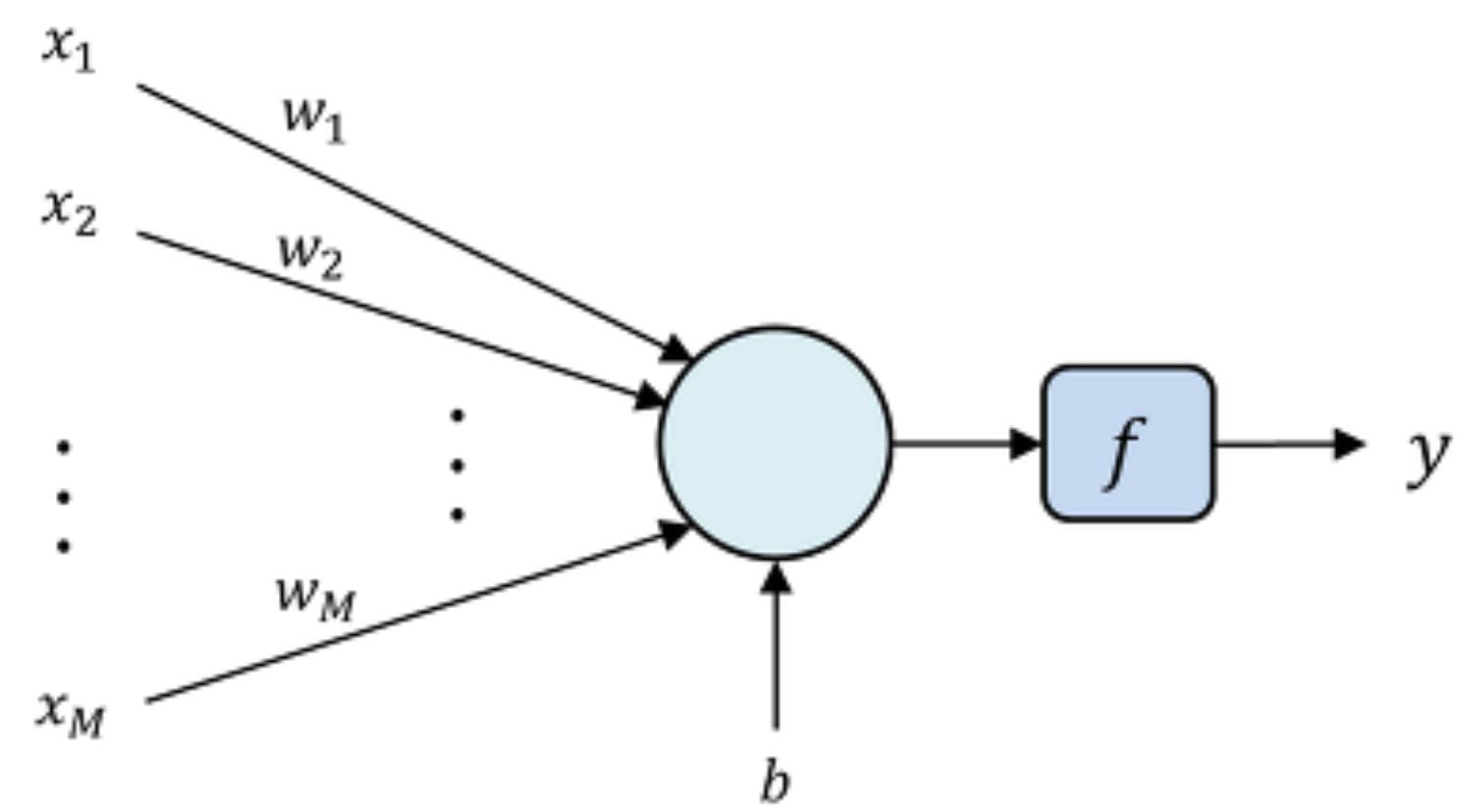
- 약 천억 (100 billion)개의 뉴런
- 약 100조 (100 trilloin)개의 시냅스
- 시냅스: 뉴런과 뉴런 사이의 연결



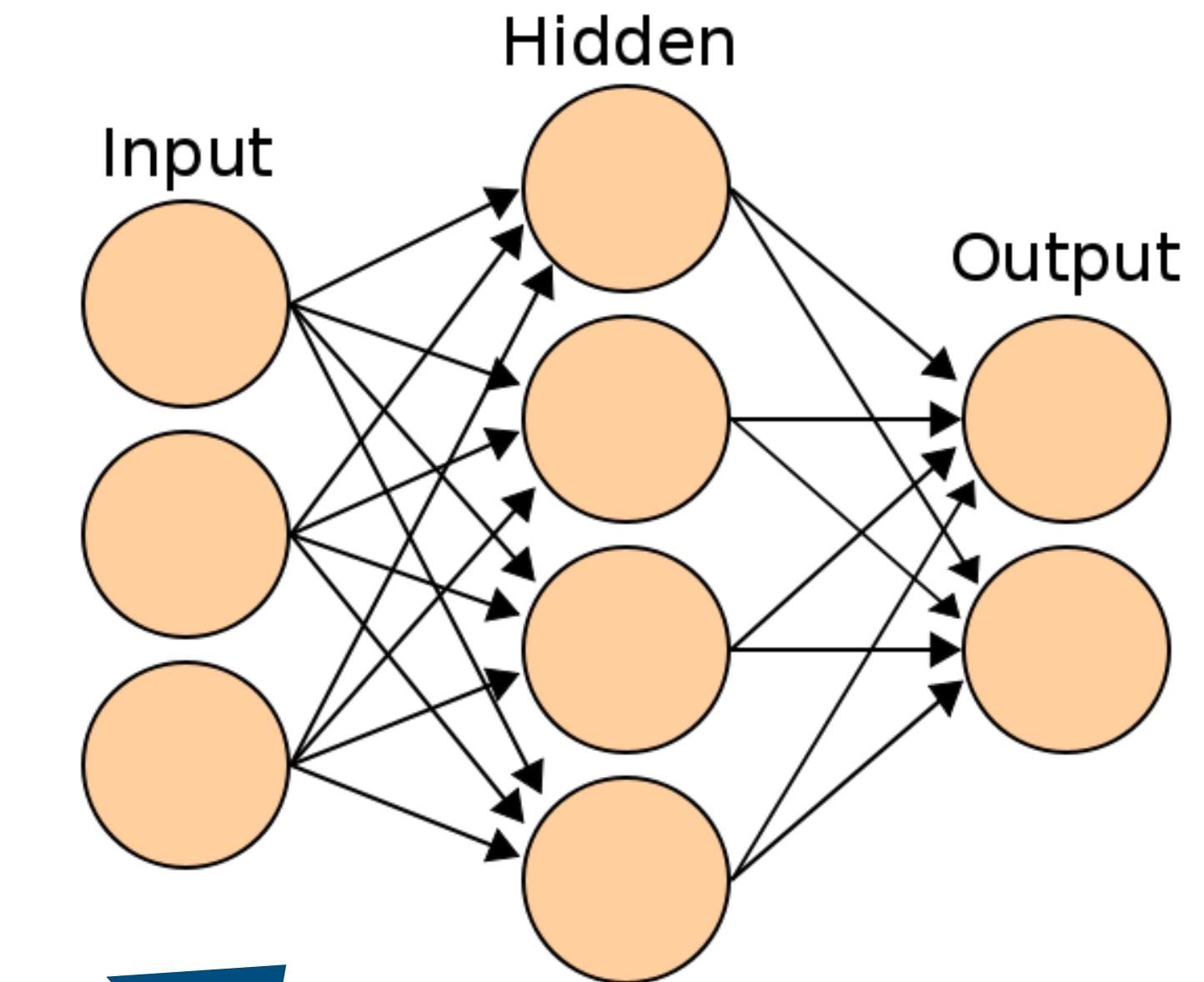
인간학습 → 기계학습, 인간두뇌 → 인공신경망



뉴런

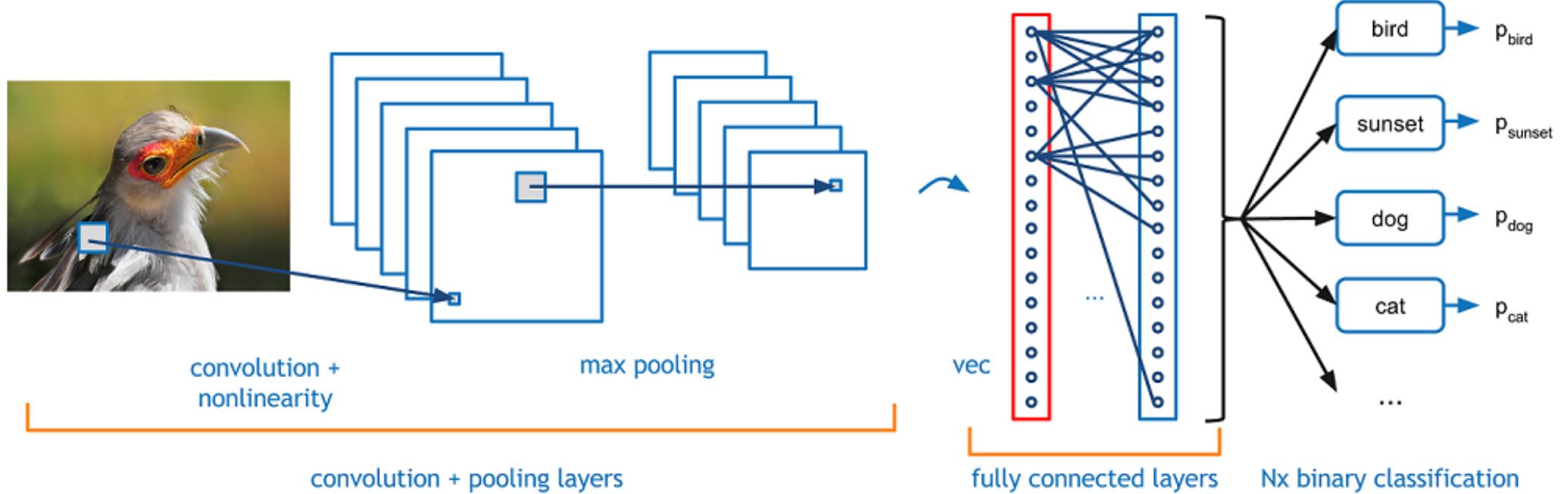


인공 뉴런

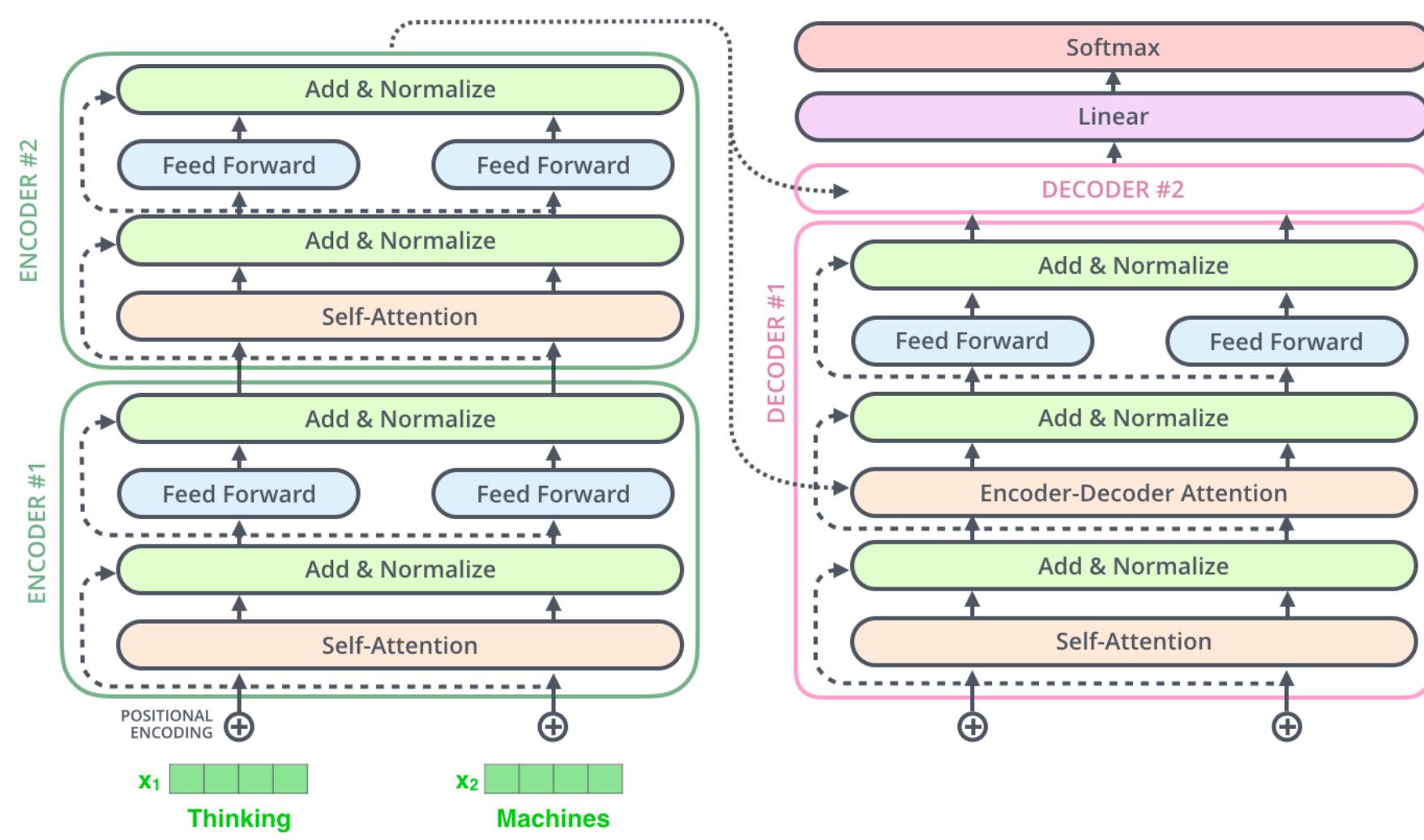


인공 신경망

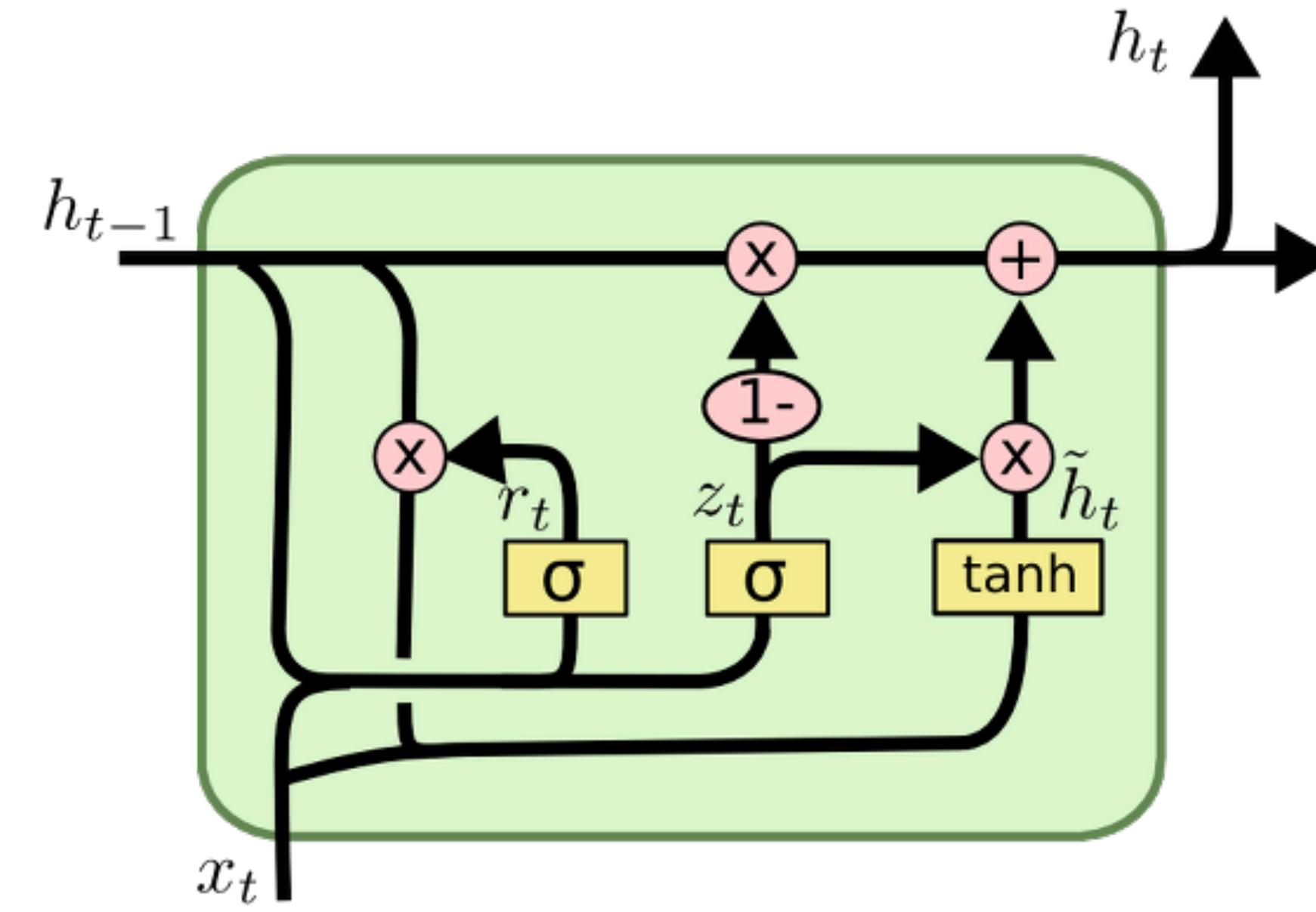
다양한 딥러닝 모델 구조



Convolutional neural networks



Transformer

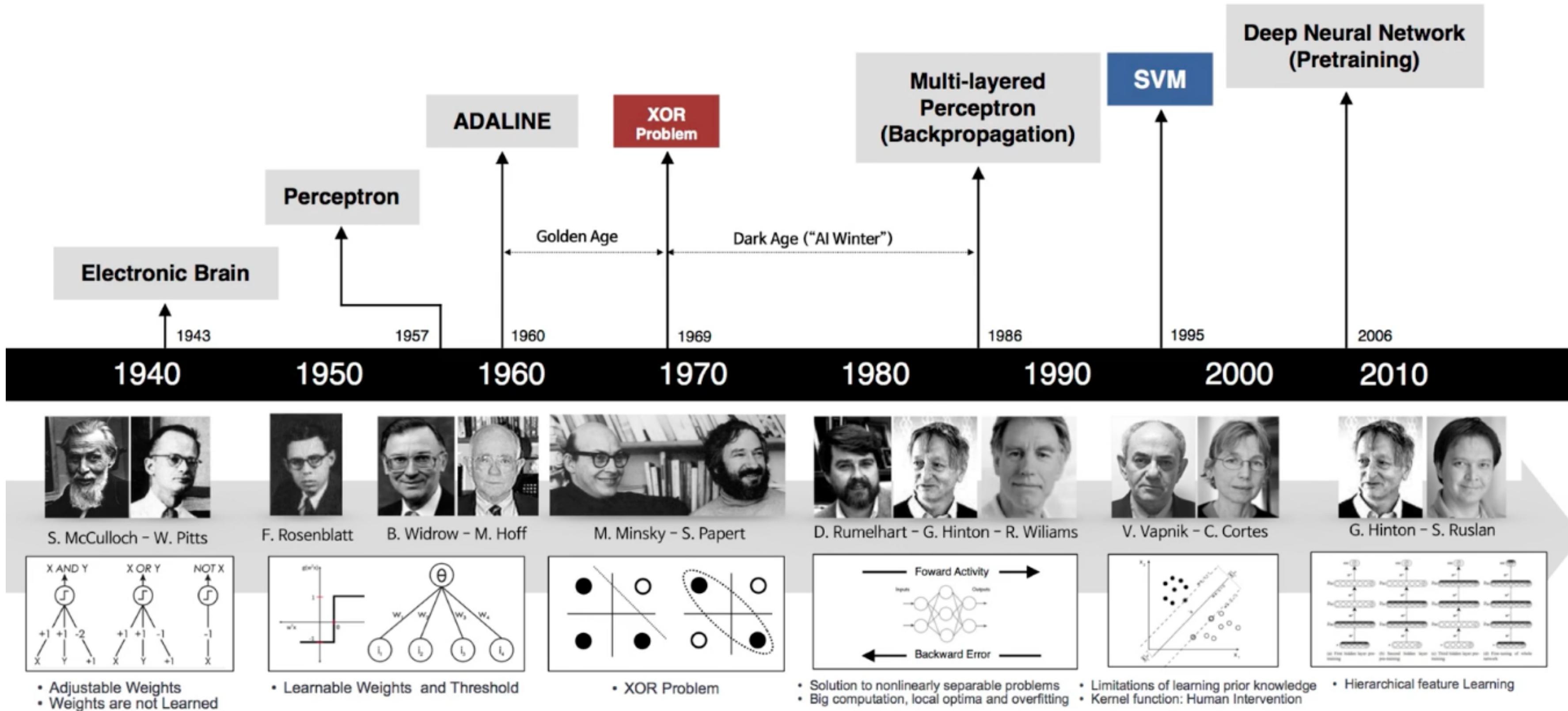


Long short-term memory

딥러닝이란

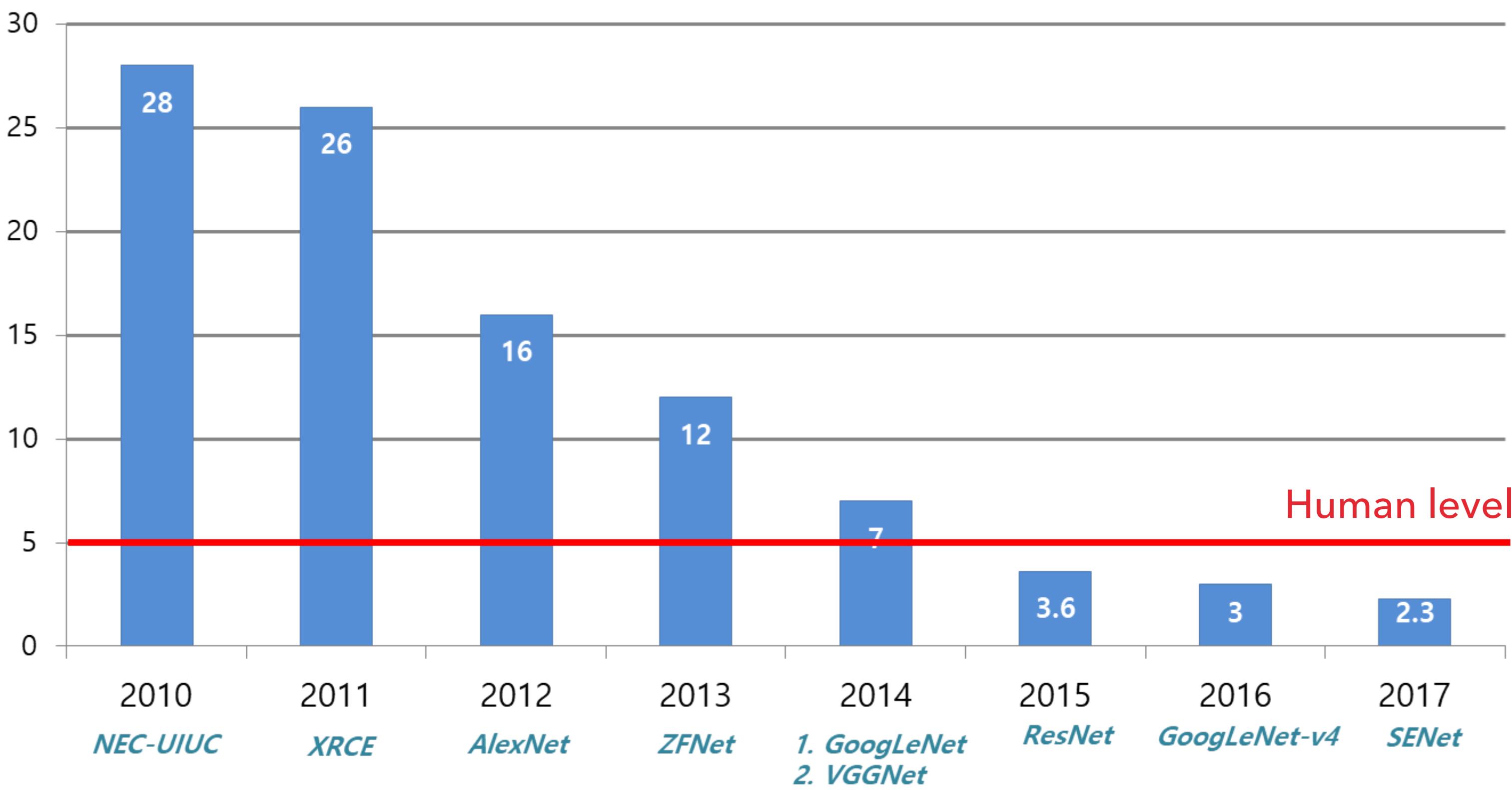
심층신경망 + 빅데이터 + 기계학습 프로세스

생각 보다 긴 딥러닝의 역사



인간을 넘어

Classification error of winning algorithms



바둑을 넘어

Configuration and strength ^[21]			
Versions	Playing hardware ^[22]	Elo rating	Matches
AlphaGo Fan	176 GPUs, ^[2] distributed	3,144 ^[1]	5:0 against Fan Hui
AlphaGo Lee	48 TPUs, ^[2] distributed	3,739 ^[1]	4:1 against Lee Sedol
AlphaGo Master	4 TPUs, ^[2] single machine	4,858 ^[1]	60:0 against professional players; Future of Go Summit
AlphaGo Zero (40 days)	4 TPUs, ^[2] single machine	5,185 ^[1]	100:0 against AlphaGo Lee 89:11 against AlphaGo Master
AlphaZero (34 hours)	4 TPUs, single machine ^[7]	4,430 (est.) ^[7]	60:40 against a 3-day AlphaGo Zero

https://en.wikipedia.org/wiki/AlphaGo_Zero

ELO rating / 2023

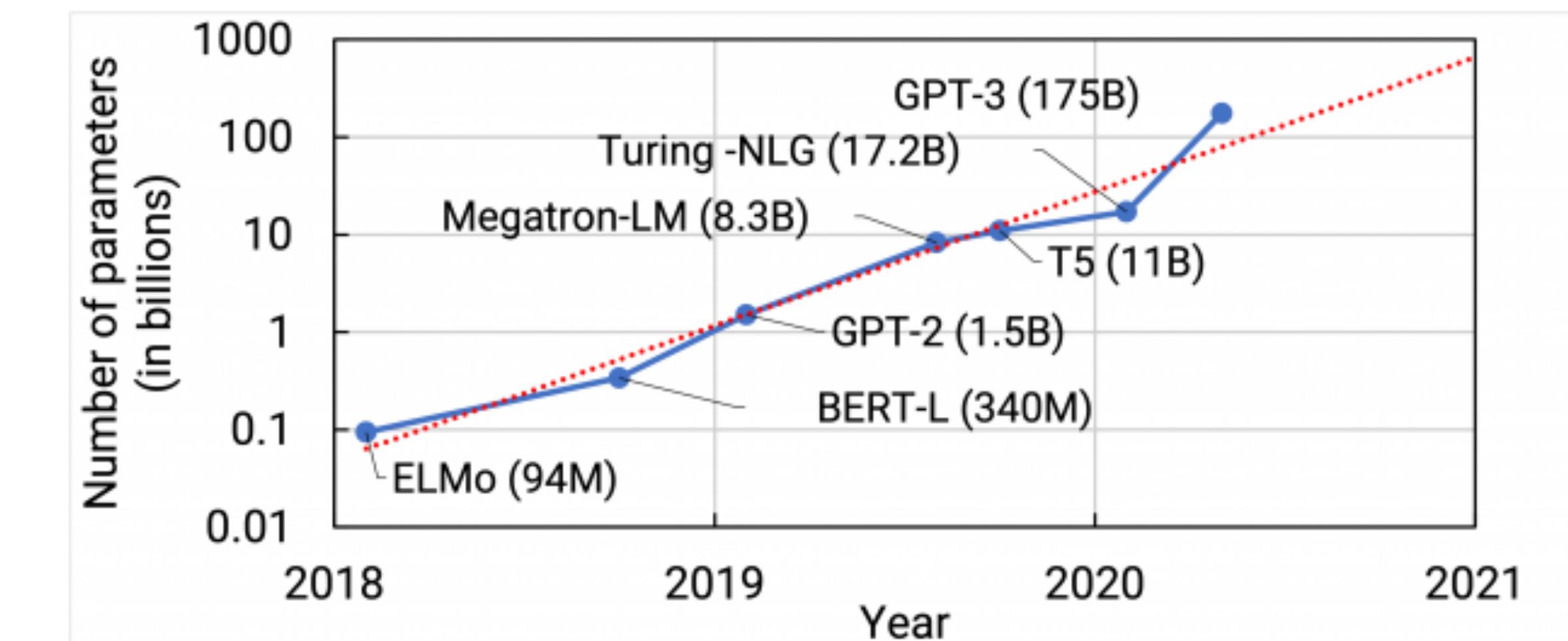
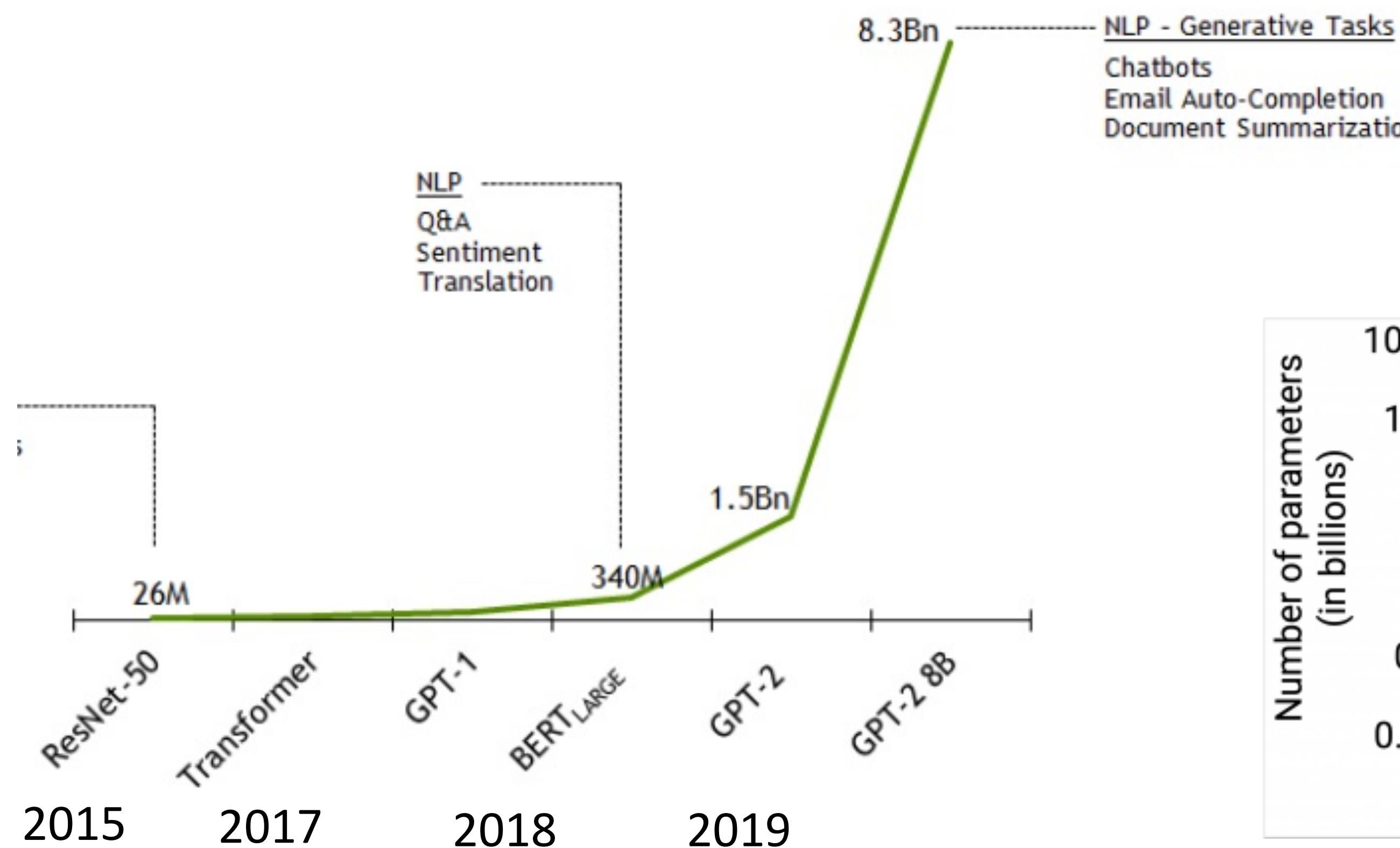
Rank	Name	♂♀	Flag	Elo
1	Shin Jinseo	♂		3826
2	Ke Jie	♂		3716
3	Park Junghwan	♂		3704

<https://www.goratings.org/en/>

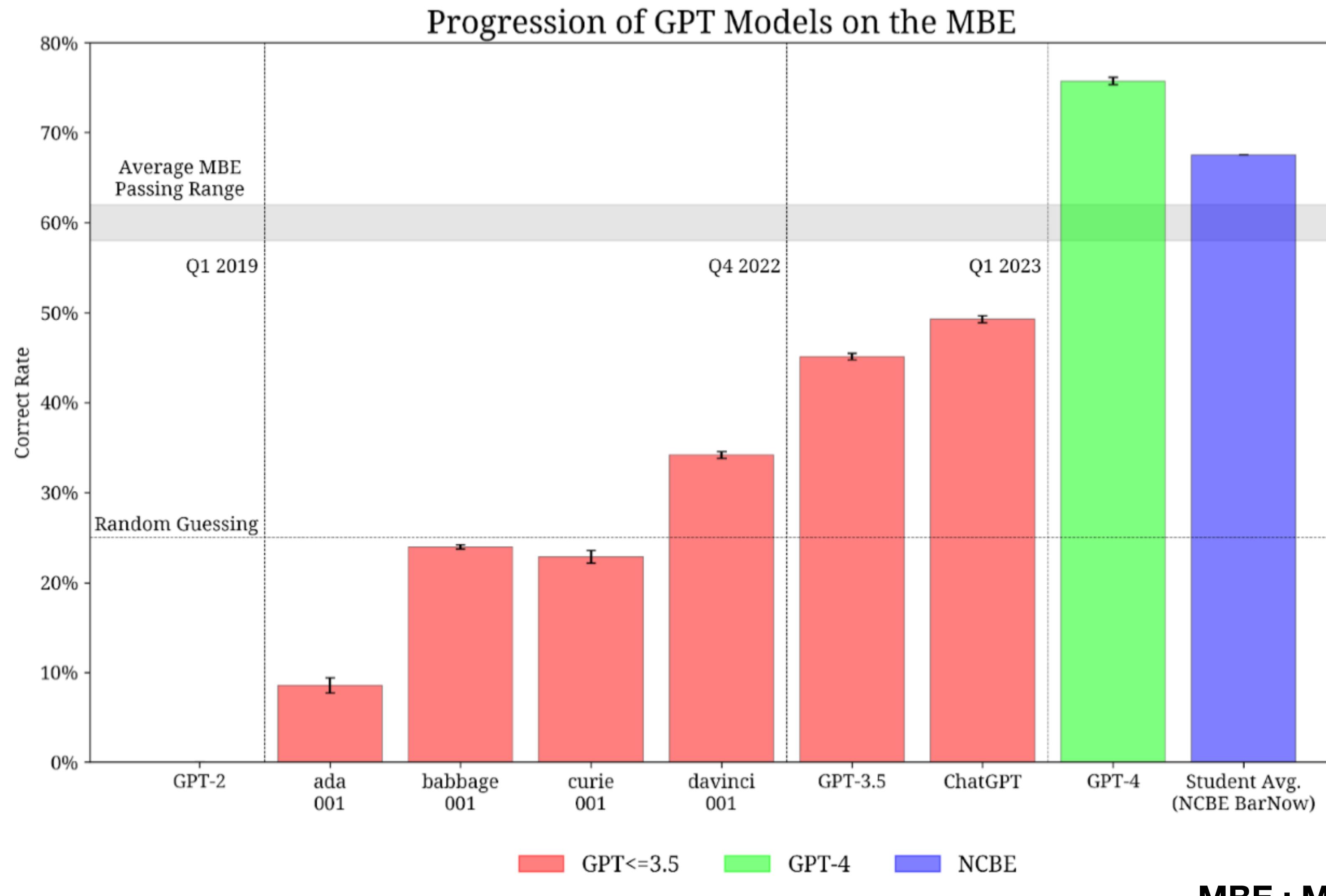
ELO rating / 2024

Rank	Name	♂♀	Flag	Elo
1	Shin Jinseo	♂		3854
2	Park Junghwan	♂		3697
3	Ding Hao	♂		3670

점점 더 커지는 신경망 모델



변호사 시험도?



최근 모델들

- GPT-3 : 1750억개
- PaLM : 540B
- GPT-4 : 1조개 이상?
- Llama 3.1 : 8B, 70B, 405B(4050억)

인공지능의 가능성 - 예술



AI(인공지능) 프로그램 미드저니가 생성한 '스페이스 오페라 극장'. 글을 입력하면 거기에 맞는 이미지를 만들어주는 이 프로그램으로 만든 작품이 미국 콜로라도주의 한 미술대회에서 디지털 아트 부문 1위를 차지하면서 예술 범위를 둘러싼 논쟁이 확산하고 있다. [사진 위키미디어]

인공지능의 가능성 - 예술

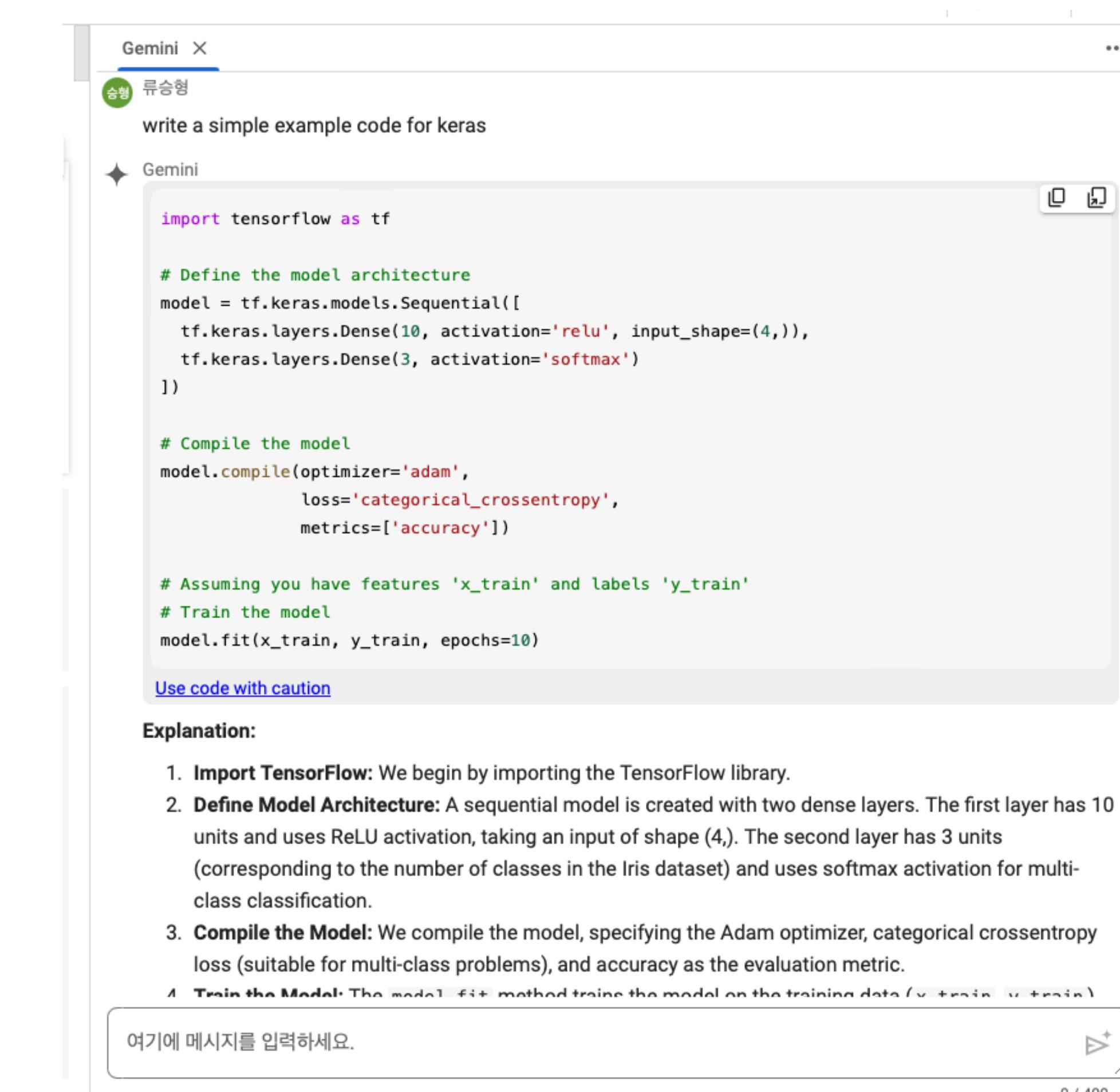


'1839 어워드'의 AI 부문 동상을 차지한 '플라밍곤(FLAMINGONE)'

인공지능의 가능성 - 코딩



The image shows the GitHub Copilot landing page. At the top, there are two tabs: "Open Source" and "Product". Below them, the main title is "Introducing GitHub Copilot: your AI pair programmer". A sub-section below the title says, "Today, we're launching a technical preview of GitHub Copilot, a new AI pair programmer that helps you write better code." A "Technical preview" button is visible. The central part of the page features the text "Your AI pair programmer" in large, bold, white font. Below this, there's a code editor window showing a snippet of JavaScript for fetching a NASA picture of the day. The GitHub Copilot logo is at the bottom left.



The image shows a screenshot of the Gemini AI interface. At the top, it says "Gemini X" and "Luis" (류승형). Below that, the user has typed "write a simple example code for keras". The AI has generated the following Python code:

```
import tensorflow as tf

# Define the model architecture
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(10, activation='relu', input_shape=(4,)),
    tf.keras.layers.Dense(3, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Assuming you have features 'x_train' and labels 'y_train'
# Train the model
model.fit(x_train, y_train, epochs=10)
```

Below the code, there's a link "Use code with caution". An "Explanation:" section follows, containing four numbered steps:

1. Import TensorFlow: We begin by importing the TensorFlow library.
2. Define Model Architecture: A sequential model is created with two dense layers. The first layer has 10 units and uses ReLU activation, taking an input of shape (4,). The second layer has 3 units (corresponding to the number of classes in the Iris dataset) and uses softmax activation for multi-class classification.
3. Compile the Model: We compile the model, specifying the Adam optimizer, categorical crossentropy loss (suitable for multi-class problems), and accuracy as the evaluation metric.
4. Train the Model: The `model.fit` method trains the model on the training data (`x_train`, `y_train`)

At the bottom, there's a text input field with the placeholder "여기에 메시지를 입력하세요." and a send button with a right arrow icon.

인공지능의 가능성 - 코딩

Tab

Cursor includes a powerful autocomplete that predicts your next edit. Once enabled, it is always on and will suggest edits to your code across multiple lines, taking into account your recent changes.

Code Generation

Cursor sees your recent changes, so it can predict what you want to do next.

```
You, 9 seconds ago | 1 author (You) | 0 implementations
7 pub struct EncodedMessageQueue {
6     pub queue: Vec<EncodedMessage, ClientFilter>,
5
4     sender: Arc<Sender<Vec<EncodedMessage, ClientFilter>>>,
3     receiver: Arc<Receiver<Vec<EncodedMessage, ClientFilter>>>,
2 }
1
6 impl EncodedMessageQueue {      You, 8 seconds ago * Uncommitted changes
5     pub fn new() -> Self {
4         let (sender, receiver) = crossbeam_channel::unbounded();
3         Self {
2             queue: vec![],
1             sender: Arc::new(sender),
             receiver: Arc::new(receiver),
         }
     }
1 }
```

Multi-Line Edits

Cursor can suggest multiple edits at once, saving you time.

```
.is_see_through(true) BlockBuilder
.light_reduce(true) BlockBuilder
.transparent_standalone(true) BlockBuilder
.build(),
Block::new(name: "Orange Concrete Block").id(5006).build(),
Block::new(name: "Blue Concrete Block").id(5007).build(),
Block::new(name: "Red Concrete Block").id(5008).build(),
Block::new(name: "White Concrete Block").id(5009).build(),
Block::new(name: "Ivory").id(5012).build(),
make_stairs(name: "Oak Stairs", id: 5013),
make_stairs(name: "Ivory Stairs", id: 5014),
make_stairs(name: "Stone Stairs", id: 5015),
```

ChatGPT

RS

오토ML에 대해 공부하고 싶어. 무엇부터 해야할까? 기초적인 내용을 알려줘



오토ML(AutoML)은 기계 학습 모델의 선택, 튜닝, 전처리 등을 자동화하여 개발자나 데이터 과학자들이 더 쉽게 기계 학습 모델을 개발할 수 있도록 돕는 기술입니다. 또한, 오토ML은 기존의 기계 학습 프로세스에 필요한 전문 지식을 덜 요구하기 때문에 더 많은 사람들이 데이터 과학을 활용할 수 있게 합니다.



오토ML에 대해 배우기 위해서는 다음과 같은 기본적인 내용들을 이해해야 합니다:

- 기계 학습 기본 지식:** 오토ML을 제대로 이해하려면 기계 학습에 대한 기본적인 이해가 필요합니다. 이에는 지도 학습, 비지도 학습, 강화 학습 등의 기본적인 학습 알고리즘과 특성 엔지니어링, 모델 선택 및 튜닝 등의 기본적인 개념이 포함됩니다.
- 데이터 전처리 및 특성 엔지니어링:** 이 단계에서는 데이터를 모델링하기 적합한 형태로 변환하는 방법에 대해 배웁니다. 또한, 원시 데이터에서 유용한 정보를 추출하거나 변환하는 특성 엔지니어링 기법에 대해서도 배웁니다.
- 모델 선택 및 하이퍼파라미터 튜닝:** 오토ML의 핵심은 모델 선택 및 하이퍼파라미터 튜닝을 자동화하는 것입니다. 이를 위해, 다양한 기계 학습 알고리즘과 이들의 하이퍼파라미터에 대해 이해하는 것이 필요합니다. 또한, 그리드 탐색, 랜덤 탐색, 베이지안 최적화 등의 튜닝 기법에 대해서도 배웁니다.
- 모델 평가 및 검증:** 모델이 얼마나 잘 훈련된 모델이 새로운 데이터에 얼마나

↳ Regenerate response

생성형 딥러닝 모델의 시대

- 딥러닝 모델이 글을 이해하고 그림을 그리는 시대.
 - 거대 모델 + 거대 데이터
- 여전히 기본적 머신러닝 모델이나 오토ML에 대한 공부를 해야할까?

인공지능 결과물의 활용

- 결과물에 대한 책임을 누가 질 것인가?

글로벌

챗GPT가 알려준 '가짜 판례' 제출한 美 변호사, 법원서 징계 위기

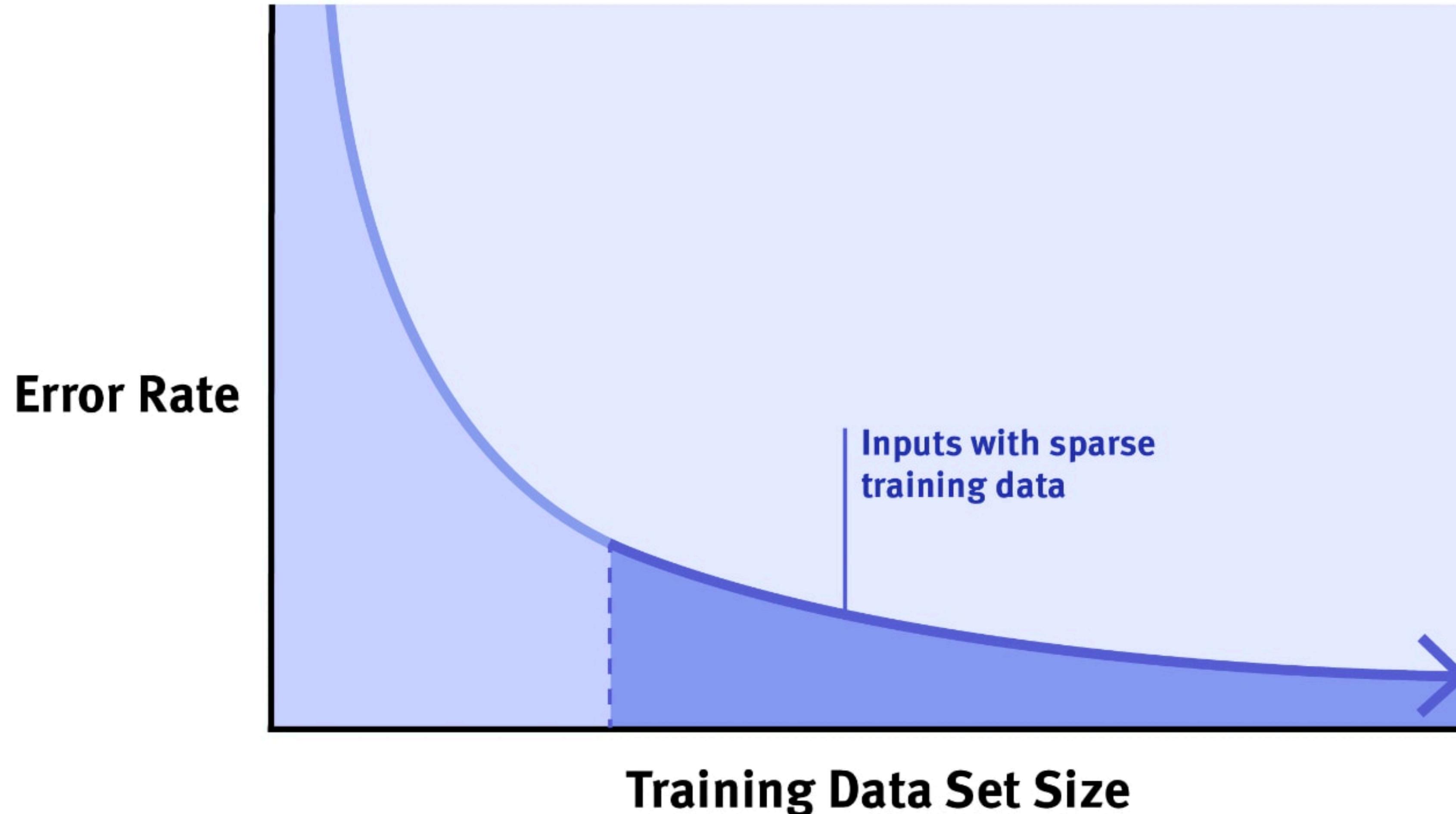
홍윤지 기자 | 2023-05-29 14:13



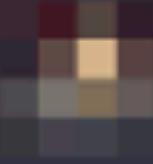
美 법원, 청문회 거쳐 변호사 윤리법 위반 여부 판단할 듯

<https://www.lawtimes.co.kr/news/187960>

데이터 기반 모델의 롱테일 문제



Hallucination - 2023

 조선왕조실록에 기록 된 세종대왕의 맥북프로 던짐 사건에 대해 알려줘

 세종대왕의 맥북프로 던짐 사건은 역사서적인 조선왕조실록에 기록된 일화로, 15세기 조선시대 세종대왕이 새로 개발한 훈민정음(한글)의 초고를 작성하던 중, 문서 작성 중단에 대한 담당자에게 분노하여 맥북프로와 함께 그를 방으로 던진 사건입니다.

세종대왕은 훈민정음 개발을 위해 열심히 노력하던 중, 한글 문자를 이해하지 못하는 당시의 관료들이 문서 작성을 돋기 위해 임명한 기획관 중 한 명인 최환(崔煥)에게 작성 중단 사실을 들게 되었습니다. 세종대왕은 이에 분노하여 최환을 부르고, 맥북프로를 함께 들고 그를 방으로 불러들이며 화를 내었습니다. 이후, 세종대왕은 훈민정음의 초고 작성을 다른 담당자에게 넘겨주게 되었습니다.

이 사건은 조선시대의 통치자로서 세종대왕이 어떻게 민족문화를 발전시키기 위해 노력했는지를 보여주는 일화로 유명합니다. 또한, 세종대왕의 열정과 업무처리에 대한 철저한 태도를 보여주는 사례로도 언급되고 있습니다.

한국일보

Hallucination - 2024

조선왕조실록에 기록된 맥북프로던짐사건에 대해 이야기해줘



"맥북프로던짐사건"은 조선왕조실록에 기록된 사건이 아니며, 현대의 디지털 기기인 맥북 프로와는 전혀 관련이 없습니다. 조선왕조실록은 조선 왕조의 역사와 관련된 기록으로, 1392년부터 1910년까지 약 500여 년간의 역사를 다룹니다.

이와 관련된 내용을 찾을 수는 없지만, 대신 조선왕조실록에 기록된 실제 사건 중 흥미로운 예를 들어볼 수 있습니다. 예를 들어, 세조 때의 사육신 사건, 연산군의 폭정, 임진왜란 등 다양한 사건들이 기록되어 있습니다. 조선왕조실록은 매우 상세하고 방대한 역사 기록으로, 조선 시대의 정치, 사회, 문화, 경제 등 다양한 측면을 담고 있습니다.

혹시 다른 조선왕조실록의 실제 사건에 대해 궁금한 점이 있다면, 말씀해 주세요.



Hallucination - 2024

조선왕조실록에 기록된 막복불호 투척 사건에 대해 이야기해줘



"막복불호 투척 사건"은 조선왕조실록에 기록된 실제 사건 중 하나입니다. 이 사건은 조선 세조(재위 1455-1468) 때 발생한 일로, 당시의 정치적 긴장과 권력 다툼의 일면을 보여줍니다.

사건 배경

세조는 단종을 폐위하고 왕위에 오른 후, 자신의 정권을 안정시키기 위해 많은 반대 세력들을 제거했습니다. 이 과정에서 권력 다툼과 암투가 끊이지 않았습니다. 막복불호 투척 사건은 이러한 정치적 불안정 속에서 벌어진 일입니다.

사건의 전개

이 사건은 세조의 측근이었던 한명회와 관련이 있습니다. 한명회는 세조의 즉위를 도운 공신으로, 당시 조선의 실세 중 한 명이었습니다. 하지만 한명회의 권력 남용과 전횡에 대해 불만을 가진 사람들이 많았습니다.

어느 날 한명회의 집 앞에서 '막복불호(莫僕不呼)'라고 쓰인 글씨가 적힌 종이를 던진 사건이 발생했습니다. 이 글귀는 "나의 종이 아니면 부르지 않는다"라는 의미로, 한명회의 전횡을 비판하는 내용이었습니다. 이 사건은 단순한 반항이 아니라, 당시에 한명회의 권력 남용에 대한 사회적 불만이 얼마나 커는지를 보여주는 사건이었습니다.

Hallucination - 2024

막복불호의 뜻이 뭐야

"막복불호(莫僕不呼)"의 뜻은 "나의 종이 아니면 부르지 않는다"입니다. 여기서 각 한자의 의미는 다음과 같습니다:

- 莫 (막): "없다" 또는 "하지 말라"
- 僕 (복): "종"
- 不 (불): "아니다"
- 呼 (호): "부르다"

이를 직역하면 "종이 아니면 부르지 않는다"라는 의미로 해석할 수 있습니다. 이 표현은 특정한 사람의 권력 남용이나 독재적인 태도를 비판하는 의미로 사용될 수 있습니다. 따라서 막복불호 투척 사건은 한명회의 권력 남용을 비판하는 사회적 불만을 상징적으로 나타낸 사건입니다.



Hallucination - 2024

세종대왕 막복불호 사건에 대해서 알려줘

❖ Sources

세종 1397 ~ 1450 - 우리역
사넷

contents.history.... · 1

세종 - 위키백과, 우리 모두의
백과사전

ko.wikipedia · 2

세종(조선) - 나무위키

namu · 3



View 5 more

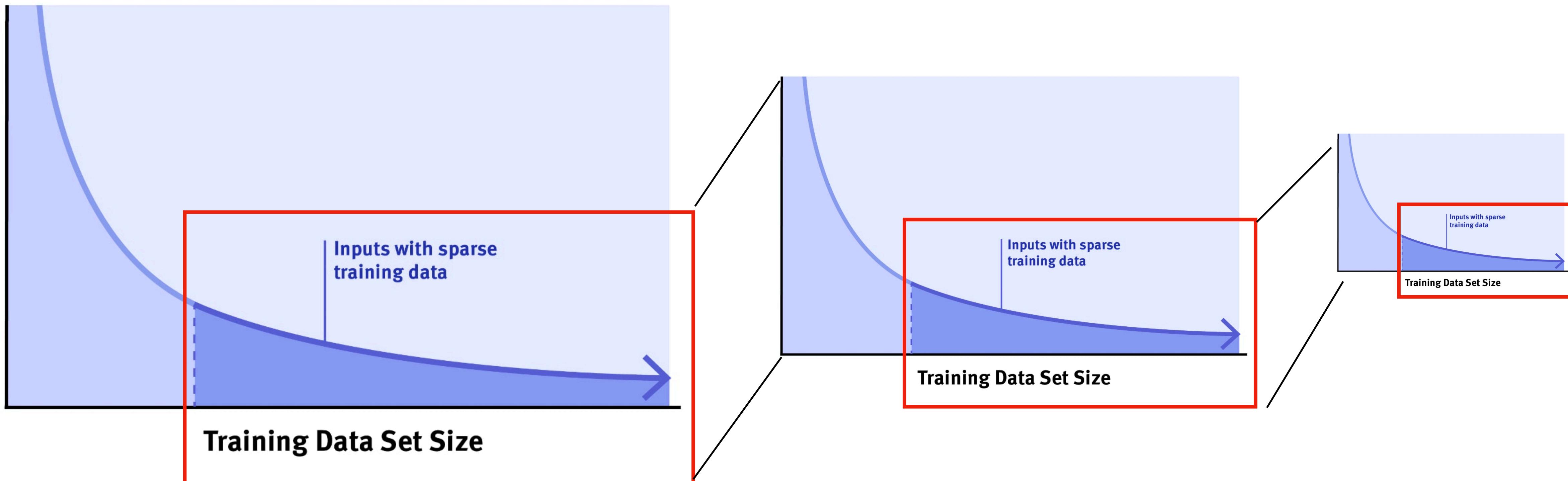
✳️ Perplexity

세종대왕의 막복불호(莫復不好) 사건은 세종이 세자로 책봉되기 전, 충녕대군 시절에 일어난 중요한 사건입니다. 이 사건은 세종의 인품과 정치적 능력을 잘 보여주는 일화로 알려져 있습니다.

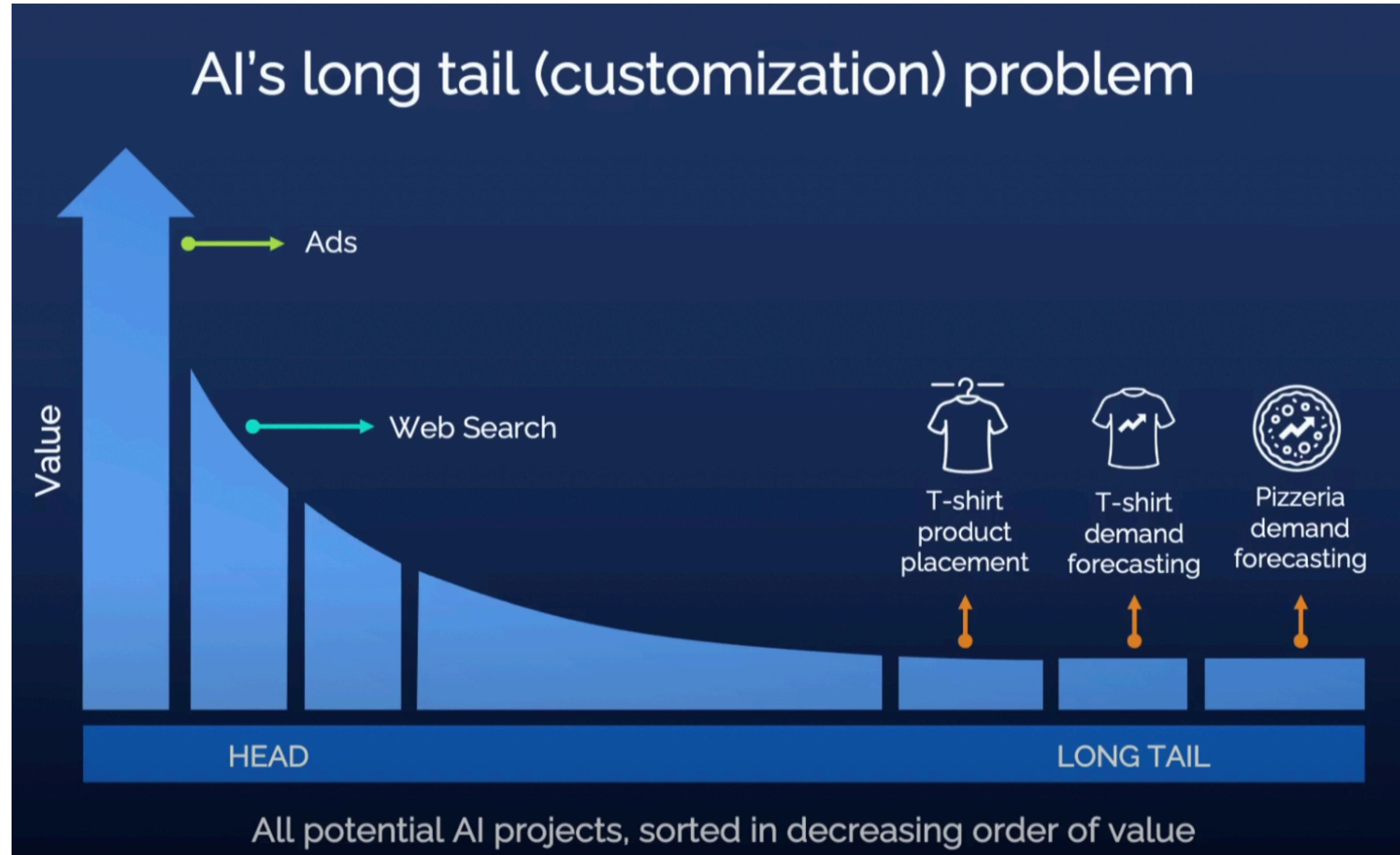
사건의 배경

세종이 세자로 책봉되기 전, 그의 형인 양녕대군이 세자의 자리에 있었습니다. 그러나 양녕대군은 학업에 소홀했고 자기 절제 능력이 부족하여 여러 가지 문제를 일으켰습니다 ¹. 이로 인해 태종은 양녕대군의 세자 자격에 대해 고민하게 되었습니다.

데이터 기반 모델의 롱테일 문제



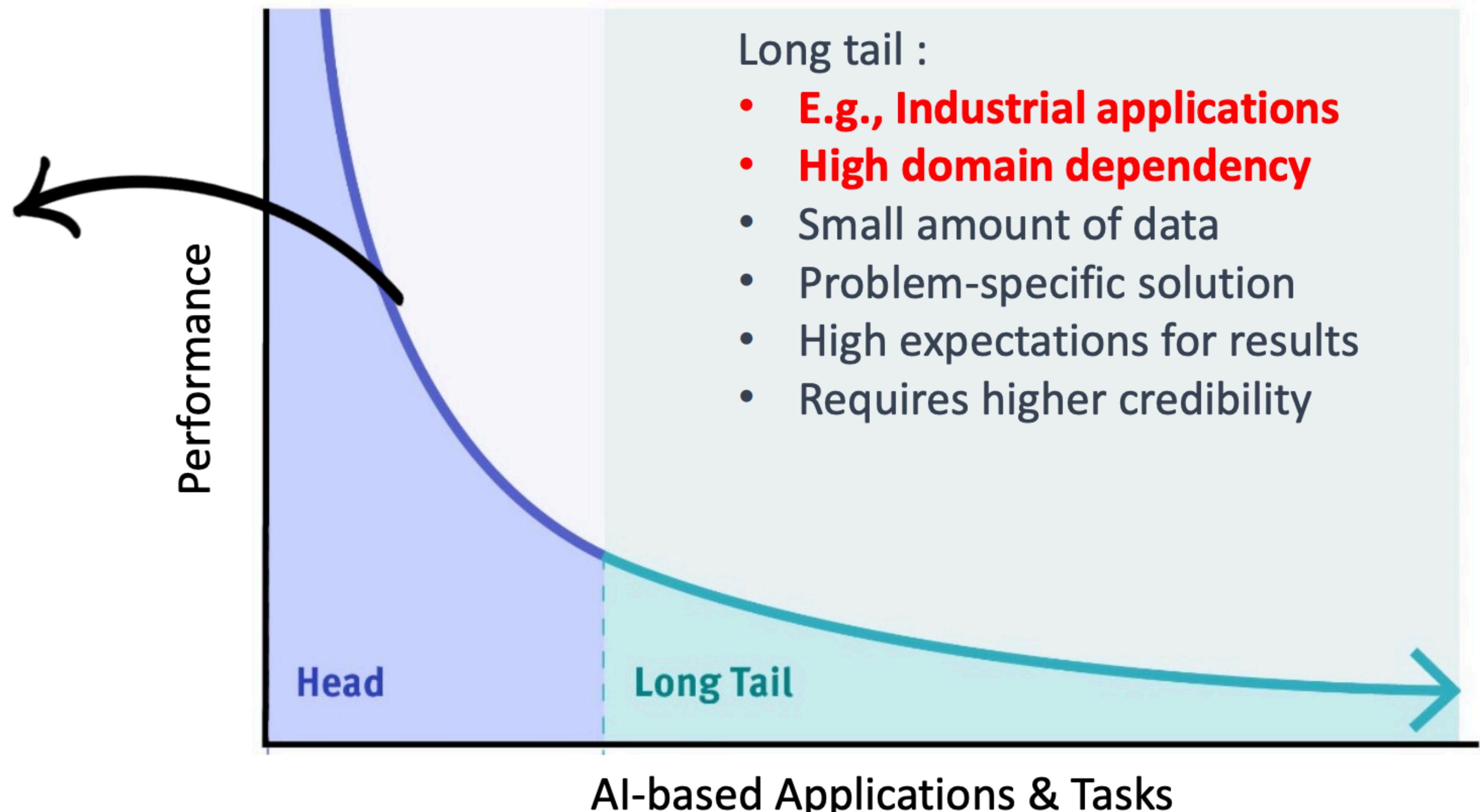
태스크 단위에서의 롱테일 문제



상대적으로 데이터가 적은 부분에서는?

Head :

- E.g., Computer vision, NLP
- General task
- Plenty of training data
- Low domain dependency
- Tolerant to plausible result



다양한 머신러닝 모델들

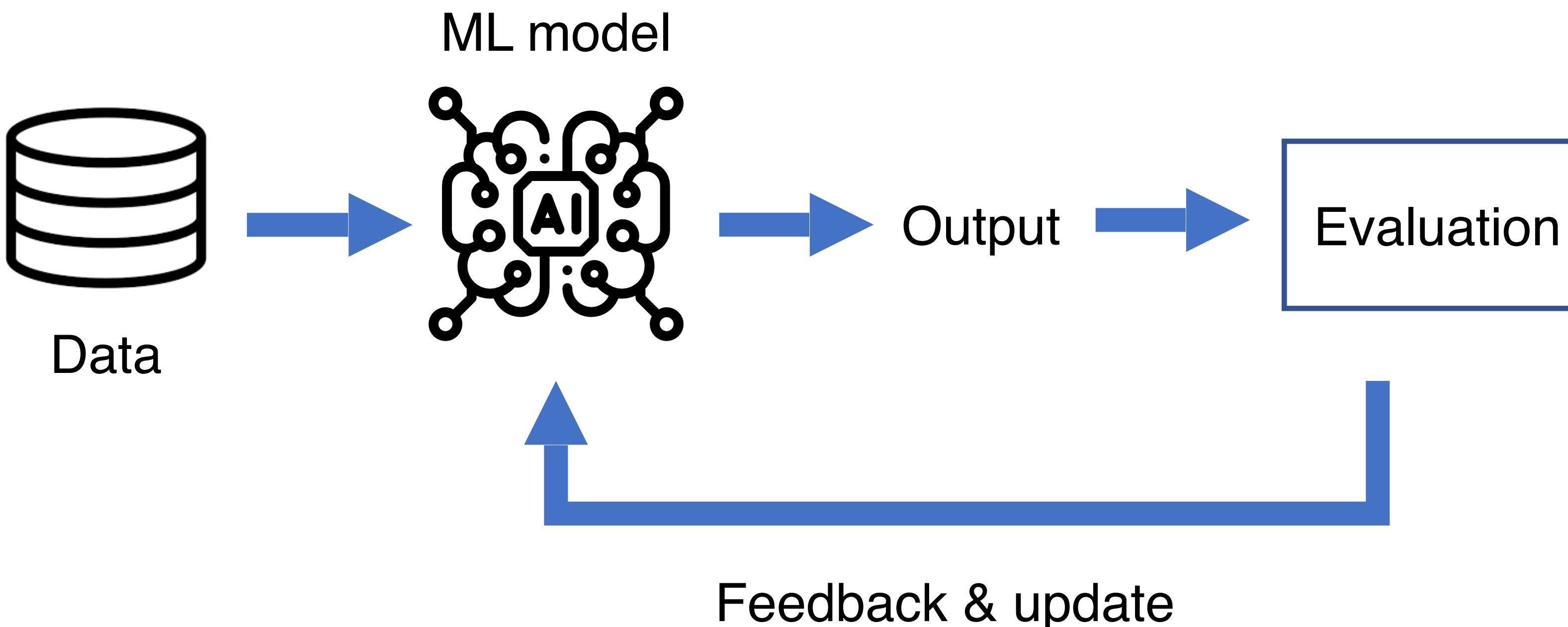
- 선형 회귀 (Linear Regression)
- 로지스틱 회귀 (Logistic Regression)
- 의사결정 트리 (Decision Tree)
- 랜덤 포레스트 (Random Forest)
- 그래디언트 부스팅 트리 (**Gradient Boosting Tree**)
 - XGBoost
 - LightGBM
 - CatBoost
- 서포트 벡터 머신 (Support Vector Machine, SVM)
- 나이브 베이즈 (Naive Bayes)
- K-최근접 이웃 (K-Nearest Neighbors, KNN)
- 신경망 (**Neural Networks**)
 - 다층 퍼셉트론 (Multi-Layer Perceptron, MLP)
 - 컨볼루션 신경망 (Convolutional Neural Network, CNN)
 - 순환 신경망 (Recurrent Neural Network, RNN)
 - 트랜스포머 (Transformer)

모델의 학습은?

- 머신러닝 모델에 따른 학습 알고리즘을 통해 학습 수행.
- 선형 회귀: 최소제곱법, 경사하강법
- K-평균 군집화: Expectation-Maximization
- 의사결정 트리: 정보 이득, 지니 불순도
- 신경망: 경사하강법 + 오류역전파법(Back propagation)

지금까지....

- 머신러닝 모델의 학습 프로세스
- 그러나 모델 개발 과정에서는 학습 외에도 다양한 과정들이 요구됨



필요한 과정들

- 어떤 문제를 풀 것인지?
- 가용한 데이터는 무엇인지?
- 데이터에 대한 변환이 필요한지?
- 어떤 머신러닝 모델을 사용할 것인지?
- 머신러닝 모델의 구조는 어떻게 설정할 것인지? 등등

AutoML

- AutoML (Automated Machine Learning) : 자동화된 기계 학습
- AutoML은 기계 학습 모델을 구축하고 최적화하는 전체 프로세스를 자동화하기 위한 기술과 방법을 의미.

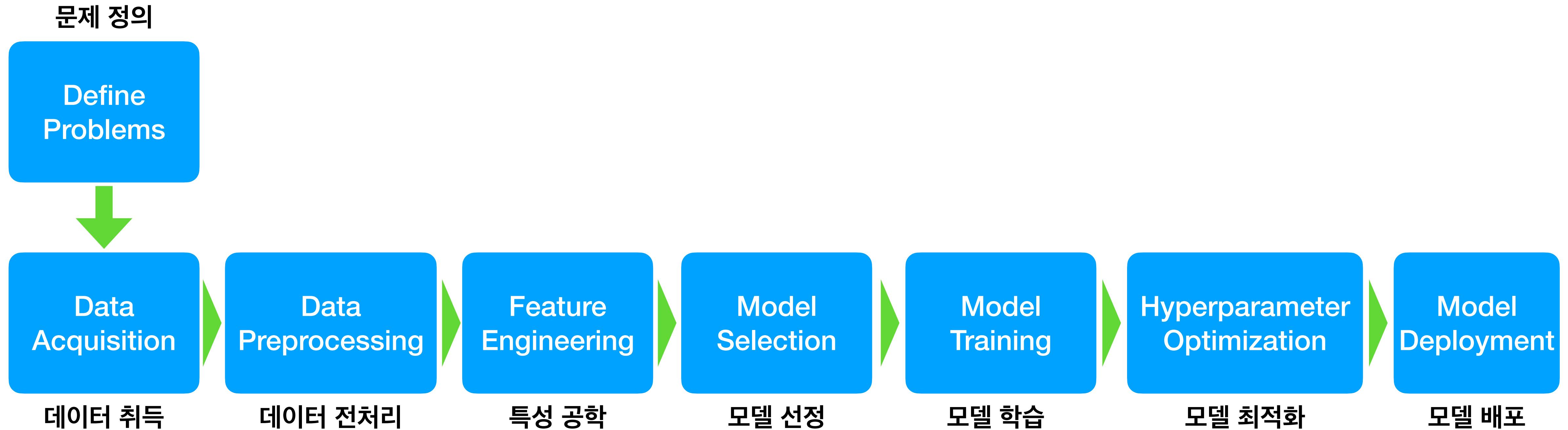
Next

- 기계학습 모델 개발의 전체 파이프라인에 대해 이해해야 함.

Chapter 2.

ML model development process

머신 러닝 모델 개발 프로세스

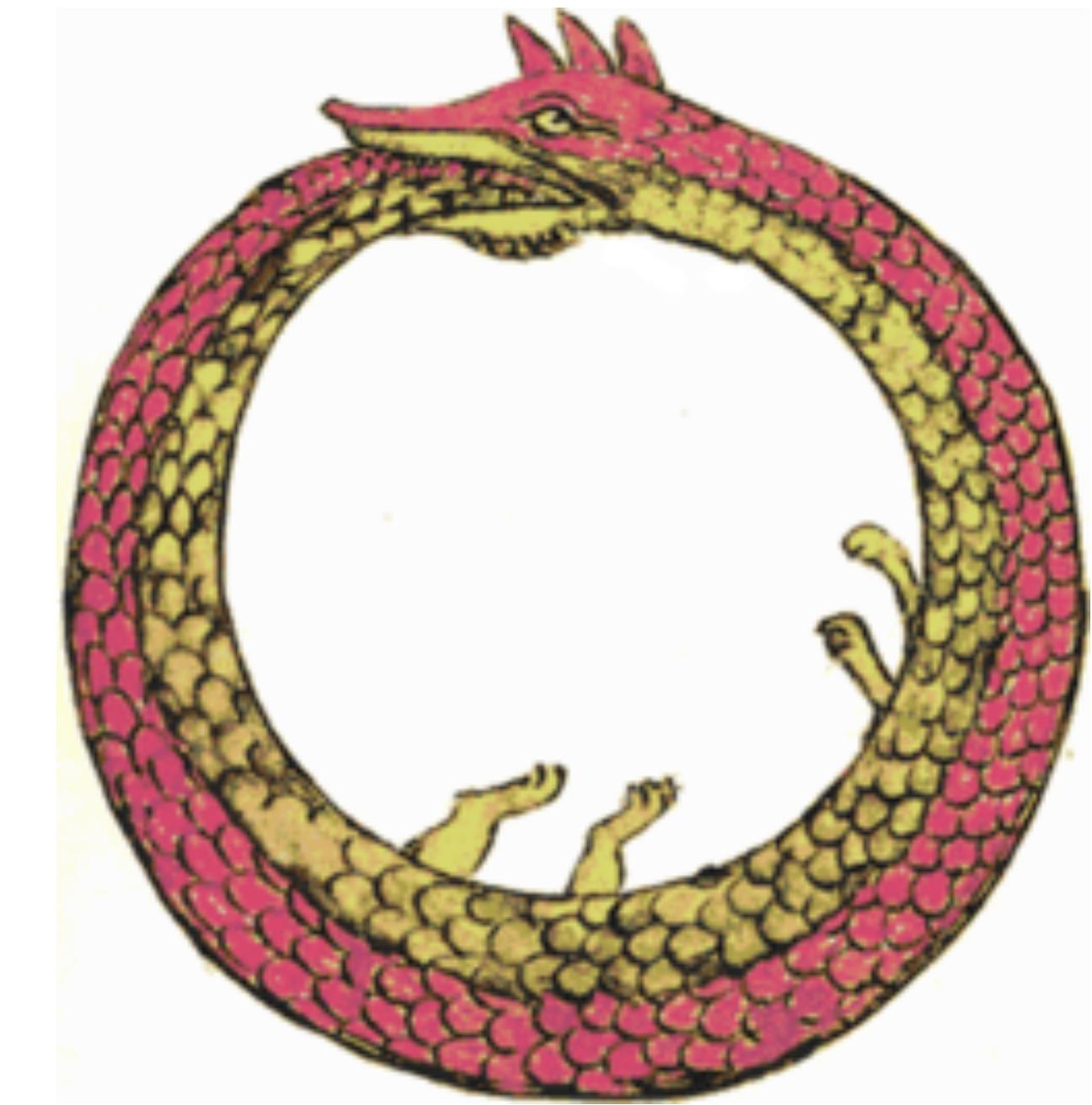


문제 정의 및 모델링

- 어떤 문제를 해결하고 싶은지?
 - 이미지를 통해 불량품을 탐지하고 싶다.
 - 구매 이력을 바탕으로 상품을 추천해주고 싶다.
- 풀고자 하는 문제의 유형은?
 - 대부분 Regression, Classification, Clustering
 - Regression : 연속적인 값을 예측하는 문제 (e.g., 주택 가격, 기온 예측 등)
 - Classification : 주어진 클래스 중 하나로 예측하는 문제 (e.g., 사물 분류 등)
 - Clustering : 데이터를 몇 개의 그룹 또는 클러스터로 나누는 문제 (e.g., 고객 그룹)

데이터 획득

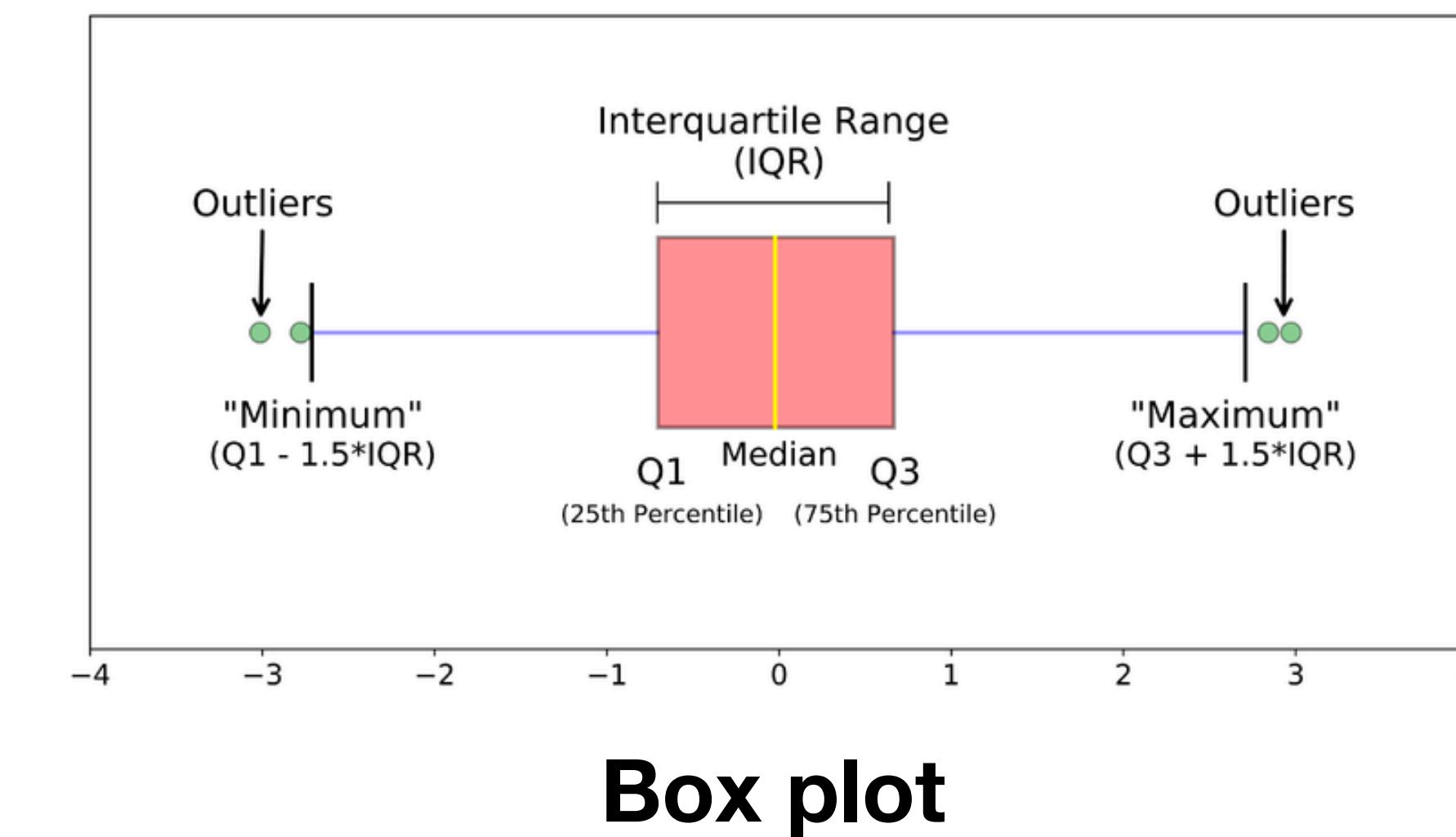
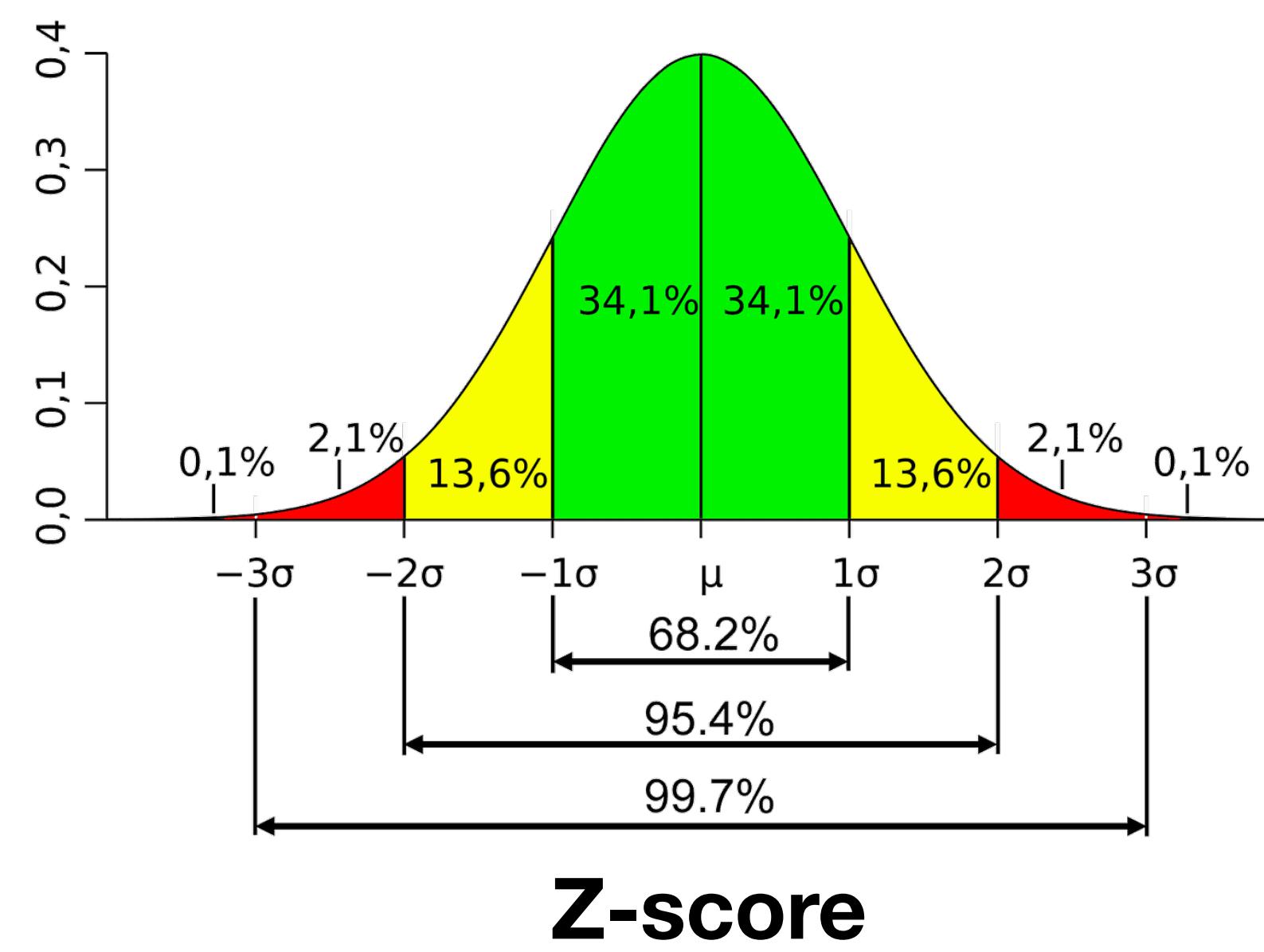
- 데이터 획득
 - 문제정의 과정에서 대략적인 데이터의 유형을 파악 가능
 - 데이터 수집 과정 및 저장소에서 데이터 가져오기
 - 데이터의 형태, 구조 및 특성 파악
 - 데이터가 먼저냐? 문제가 먼저냐?



우로보로스

데이터 전처리 (결측치 및 이상치)

- 결측치 처리: 결측치 확인 및 대체 또는 제거
 - Interpolation, 평균, 중앙값, 최빈값 등 치환, 결측값 구분
- 이상치 처리: 이상치 탐지 및 대체 또는 제거



데이터 전처리 (정규화 및 인코딩)

- 데이터 정규화: 데이터 범위 변환
 - Z-score normalization : 데이터의 평균과 표준편차를 각각 0,1이 되도록 보정 (
 - min-max normalization : 데이터의 최소값과 최대값이 각각 0, 1이 되도록 보정
 - 로그 변환
- 범주형 데이터 인코딩:
 - 레이블 인코딩 : 범주형 레이블에 대한 숫자 레이블 할당 (e.g., ‘개’:1, ‘고양이’ :2)
 - 원 핫(One-hot) 인코딩 : 전체 클래스에 대한 벡터를 생성 후 해당 클래스에 1 할당. (e.g., ‘개’, ‘고양이’, ‘토끼’. → ‘개’ = [1,0,0], ‘고양이’ = [0,1,0], ‘토끼’ = [0,0,1].

특성 공학 (Feature engineering)

- 다항 변환 (Polynomial feature) : 기존 변수들의 다항식 조합 활용, (e.g., x^2 , xy , y^3).
- 그룹 특성 (Group feature) : 연관된 변수 그룹이 존재 시 통계값 활용, (e.g., $t_1, t_2, t_3, \dots, t_k \rightarrow \mu_t, \sigma_t$)
- 구간화 (Binning) : 연속형 변수를 구간을 나누어 범주형 변수로 변환
 - (e.g., 0~200까지 속도값, → stop, slow, fast, very fast 클래스로 구분)
- 변수 분할 (Feature split) : 복합적으로 구성된 변수를 분할하여 활용
 - (e.g., 20220301 → (연도) 2022, (월) 3, (일) 1.
- 푸리에 및 웨이브렛 변환 (시계열 데이터), 변수 추가 (e.g., 날짜데이터 → 공휴일 변수), PCA 등

모델 선정

- 주어진 문제 구성과 데이터에 적합한 기계학습 모델 선정
 - K-means 클러스터링 → 시계열 예측 ?
 - 시계열 데이터 → 그래프 이미지 변환 → 2D CNN ?
 - 가능하지만 정석적인 접근법부터 시작.
- 딥러닝 이후에는...
 - 이미지, 자연어, 시계열 데이터 → 딥러닝 모델들
 - 표 형식의 데이터 → LightGBM, Catboost 등 Gradient boosting 모델들

모델 학습

- 모델 선정 이후에는 모델의 학습이 필요
- 반복적인 학습 과정을 통해 점진적인 성능의 향상
- 학습 과정과 개념은 기계학습 알고리즘 별로 차이가 있음

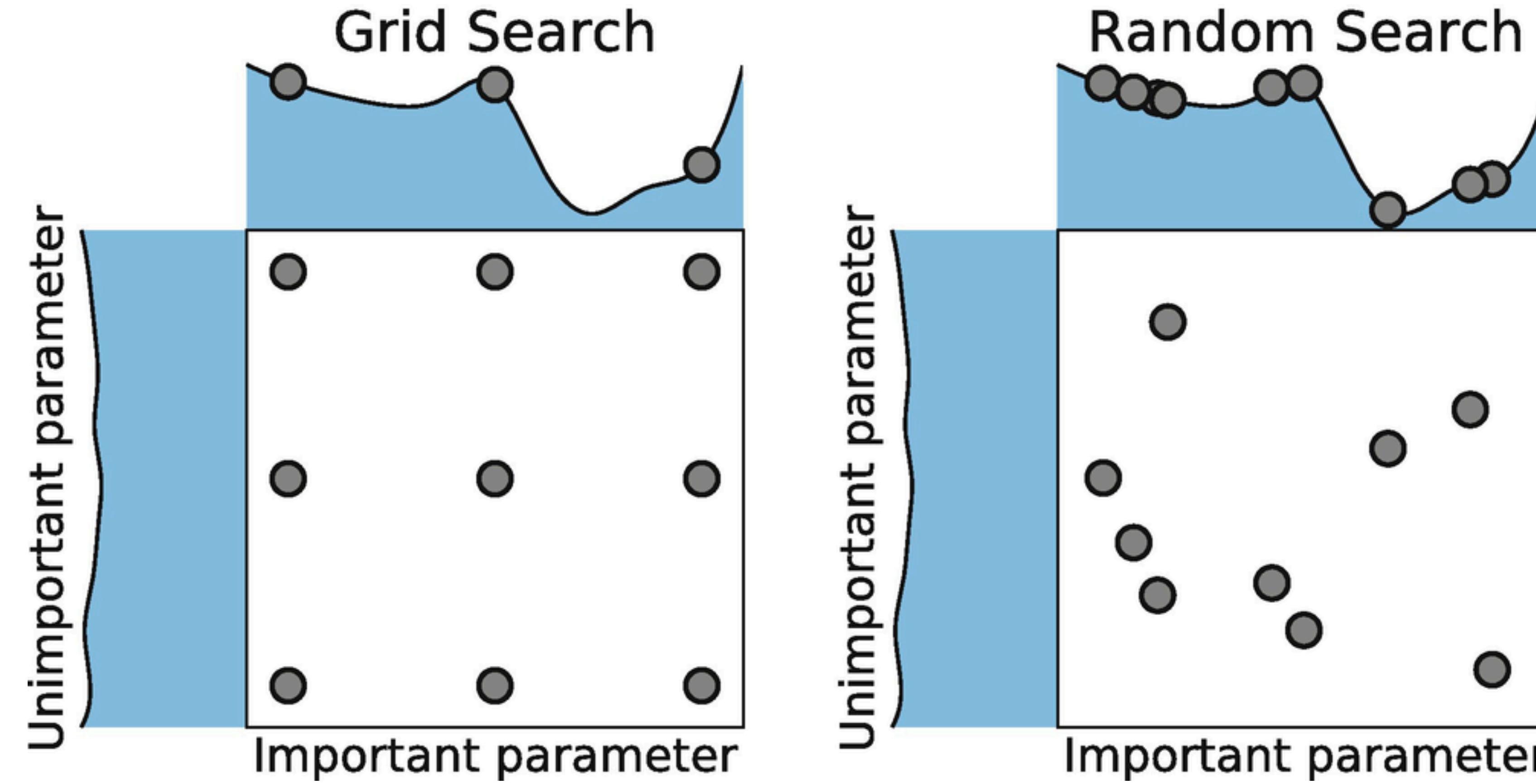
하이퍼 파라미터 튜닝

- 하이퍼 파라미터 :
 - 학습하고자 하는 머신러닝 모델에 대해서 모델 구조, 학습 과정 등과 관련하여 사전에 설정해주는 변수들.
 - 학습을 통해 배우는 파라미터는 아니지만, 모델 학습 결과에 큰 영향을 미침.
- 머신러닝 모델 별 하이퍼 파라미터 예시
 - 신경망 : 뉴런의 수, 레이어의 수, 모델 구조, 활성 함수, learning rate, optimizer, etc.
 - 서포트벡터머신 : 커널의 종류, 마진(margin)
 - 의사결정트리, 랜덤포레스트, lightGBM : 트리의 수, 최대 깊이, 서브샘플링 여부 등

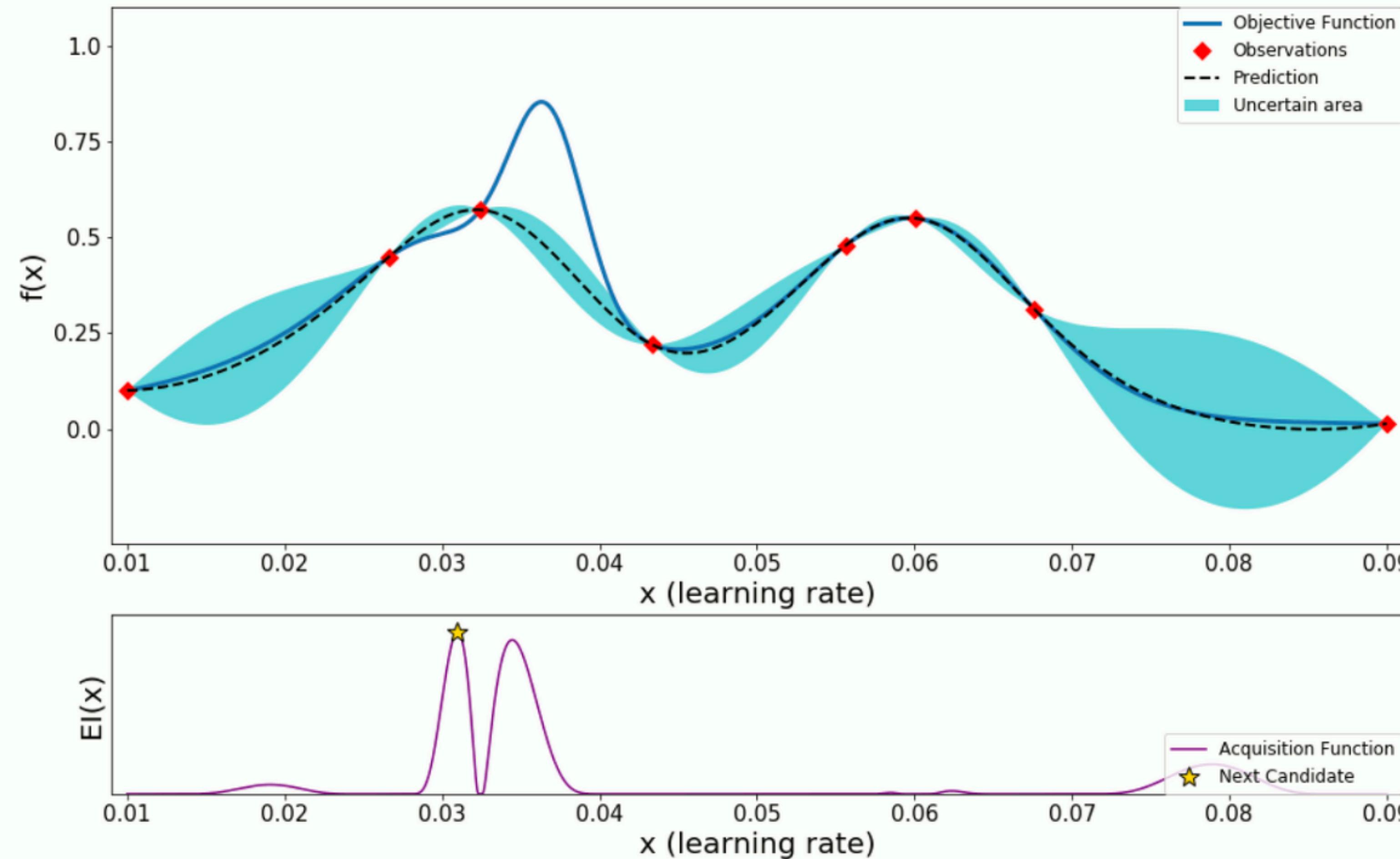
하이퍼 파라미터 튜닝

- 최적의 하이퍼 파라미터를 선정하기 위해서는?
 - 모의고사가 필요함, 즉 밸리데이션 셋에 대한 성능을 바탕으로 하이퍼 파라미터 탐색.
 - 테스트 셋으로 하이퍼 파라미터 선정시 테스트셋에 과적합 될 수 있음.
- 대표적인 방법론
 - Manual search, Grid search, Random search, Bayesian optimization

Grid search & Random search



Bayesian optimization

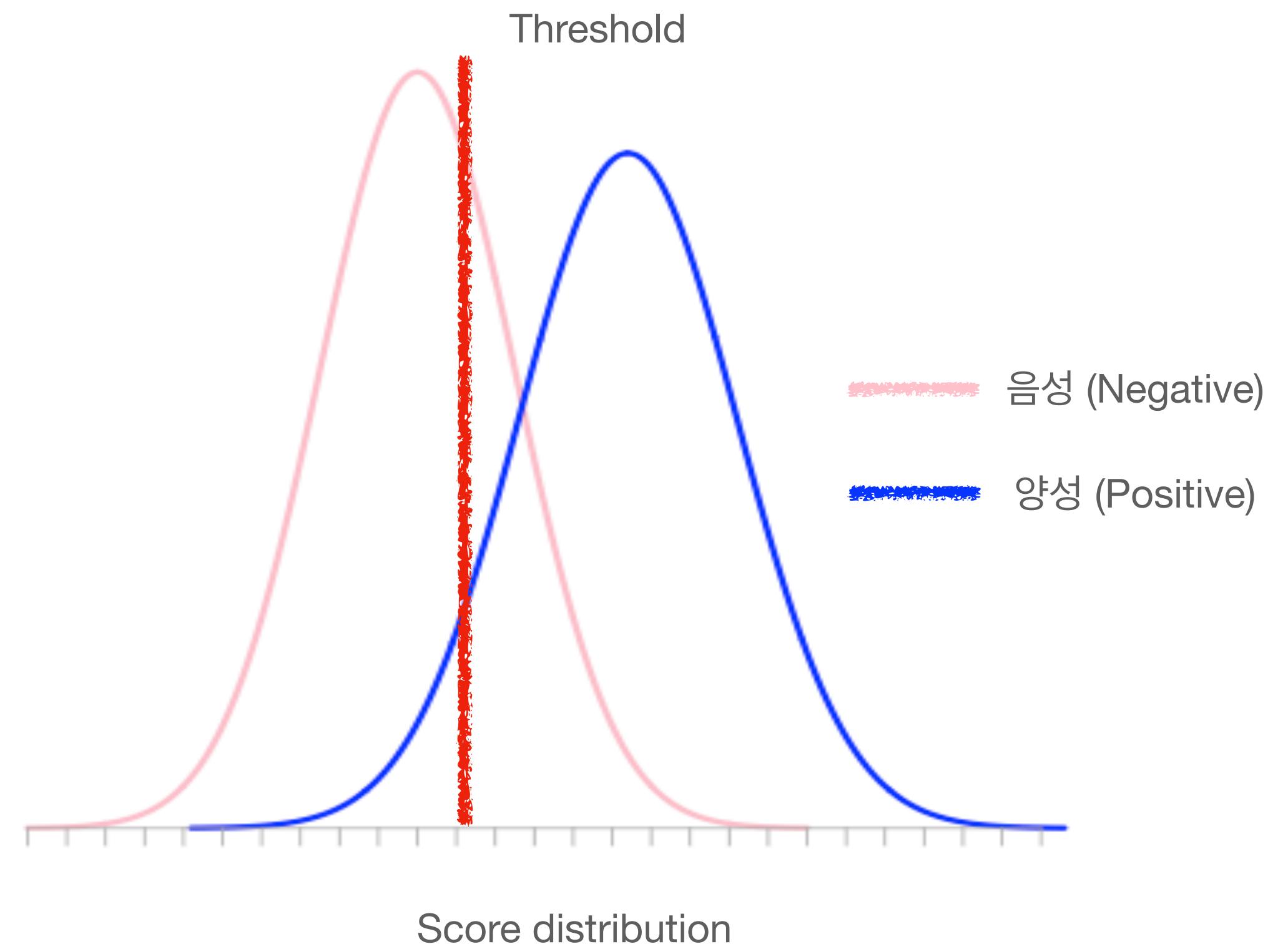


성능 평가

- Classification : Accuracy, F1 score, AUROC
- Regression : Mean absolute error, Mean squared error, R-squared

분류 문제 메트릭 - 기초 요소

- True Positive (TP) : (기준선 위의 파란 영역)
 - 실제 양성을 양성으로 분류.
- True Negative (TN): (기준선 밑의 붉은 영역)
 - 실제 음성을 음성으로 분류.
- False Positive (FP): (기준선 위의 붉은 영역)
 - 실제 음성을 양성으로 분류, False alarm.
- False Negative (FN): (기준선 밑의 파란 영역)
 - 실제 양성을 음성으로 분류, Miss detection.



분류 문제 메트릭 - Confusion matrix

- Recall, True Positive Rate (TPR) :
 - $TP/(TP+FN)$, 양성 예측 / 실제 양성
 - 값이 클수록 좋음.
- Fall-out, False Positive Rate (FPR)
 - $FP/(FP+TN)$, 양성 예측 / 실제 음성
 - 값이 작을수록 좋음
- Precision
 - $TP/(TP+FP)$, 실제 양성 / 양성 예측
 - 값이 클수록 좋음
- Accuracy
 - $(TP+TN)/(TP+TN+FP+FN)$, 맞게 예측 / 전체 데이터
 - 값이 클수록 좋음

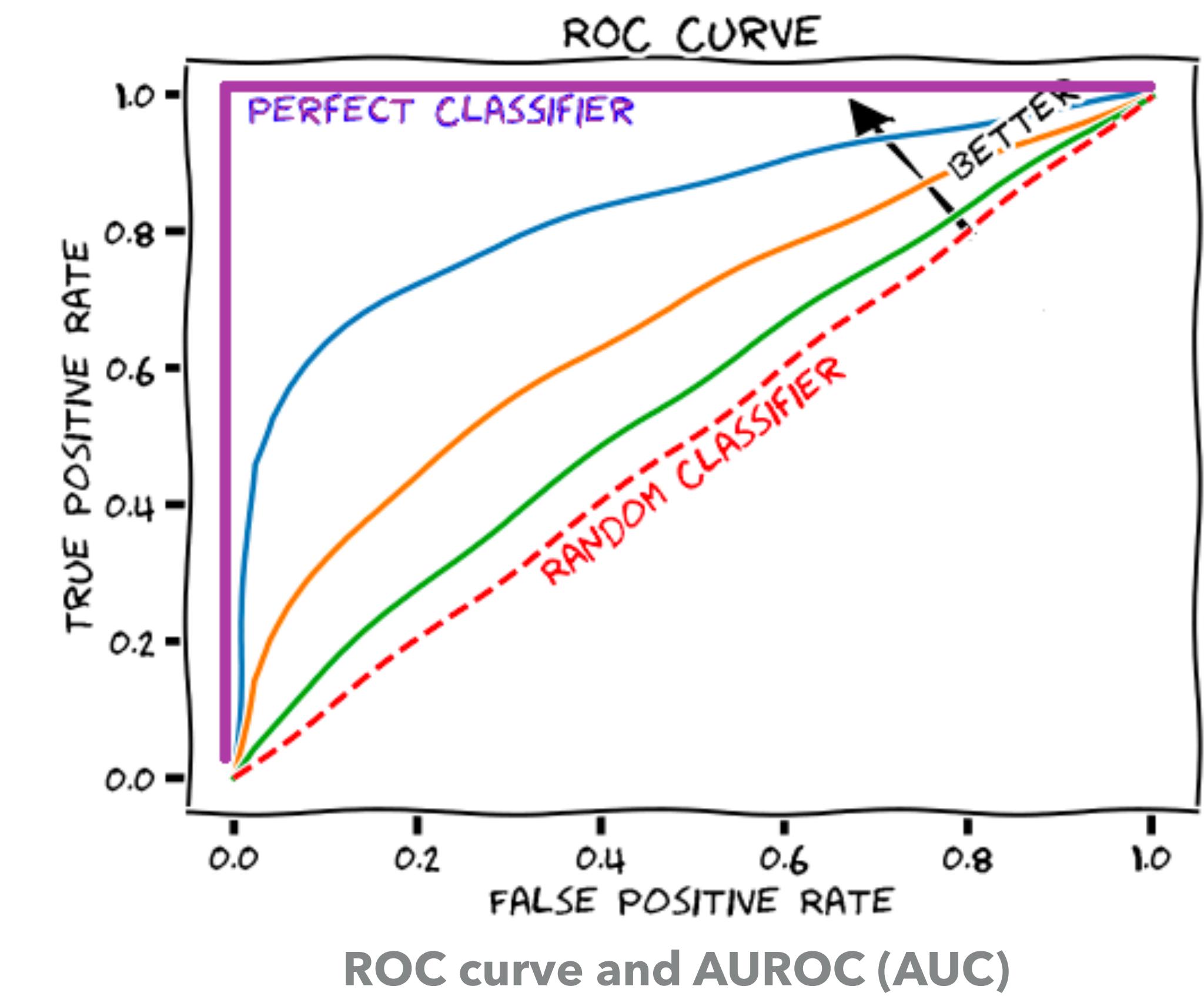
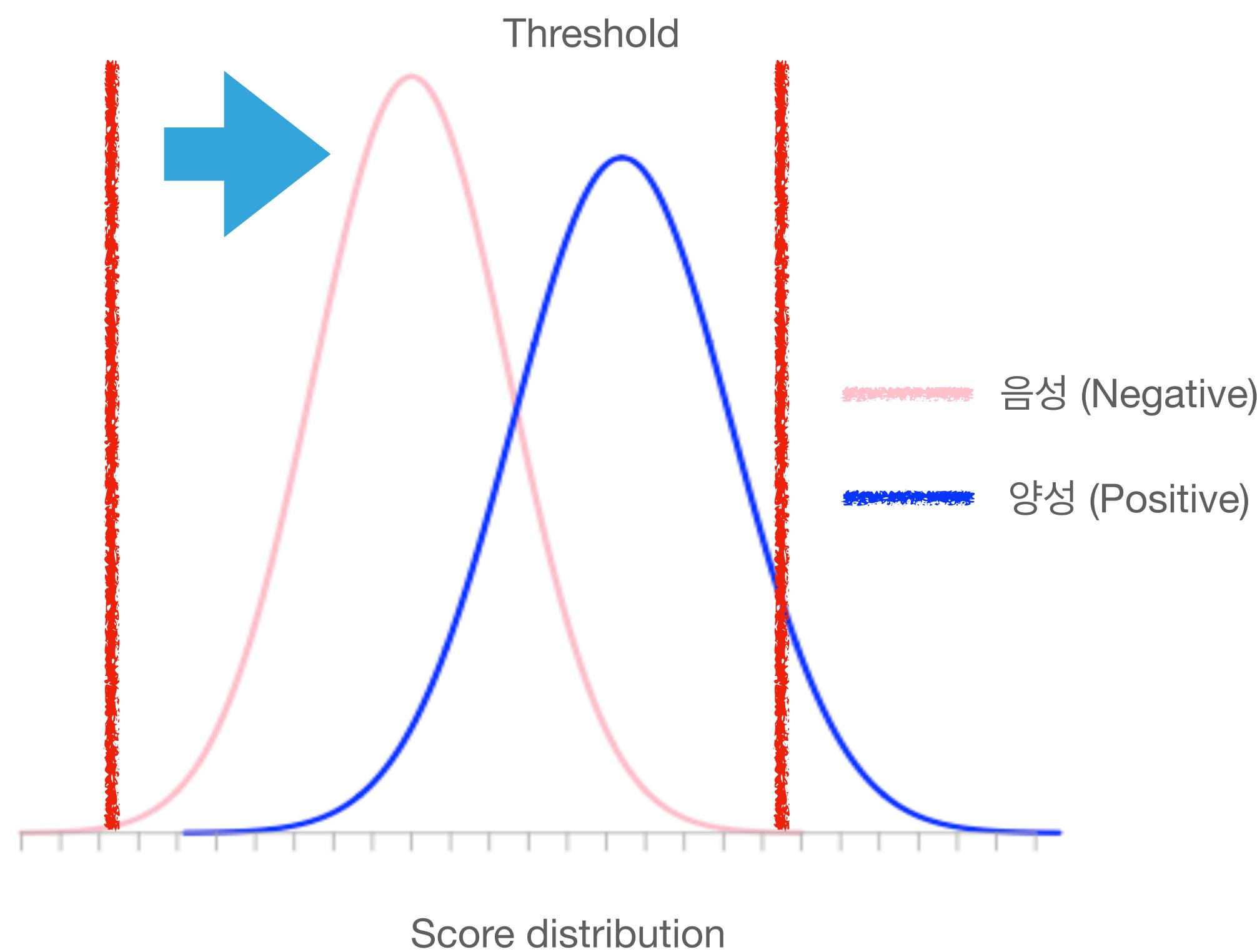
		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP False alarm
	Negative	FN Miss detection	TN
		Confusion Matrix	

분류 문제 메트릭 - F1 score

- F1-score
 - $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
 - Precision과 recall의 조화평균

분류 문제 메트릭 - AUROC

- Perfect classifier : TPR(Recall)=1, FPR = 0, Precision=1
- 0.5 : Random, 1 = Perfect classifier



회귀 문제 메트릭 - MAE, MAPE, MSE

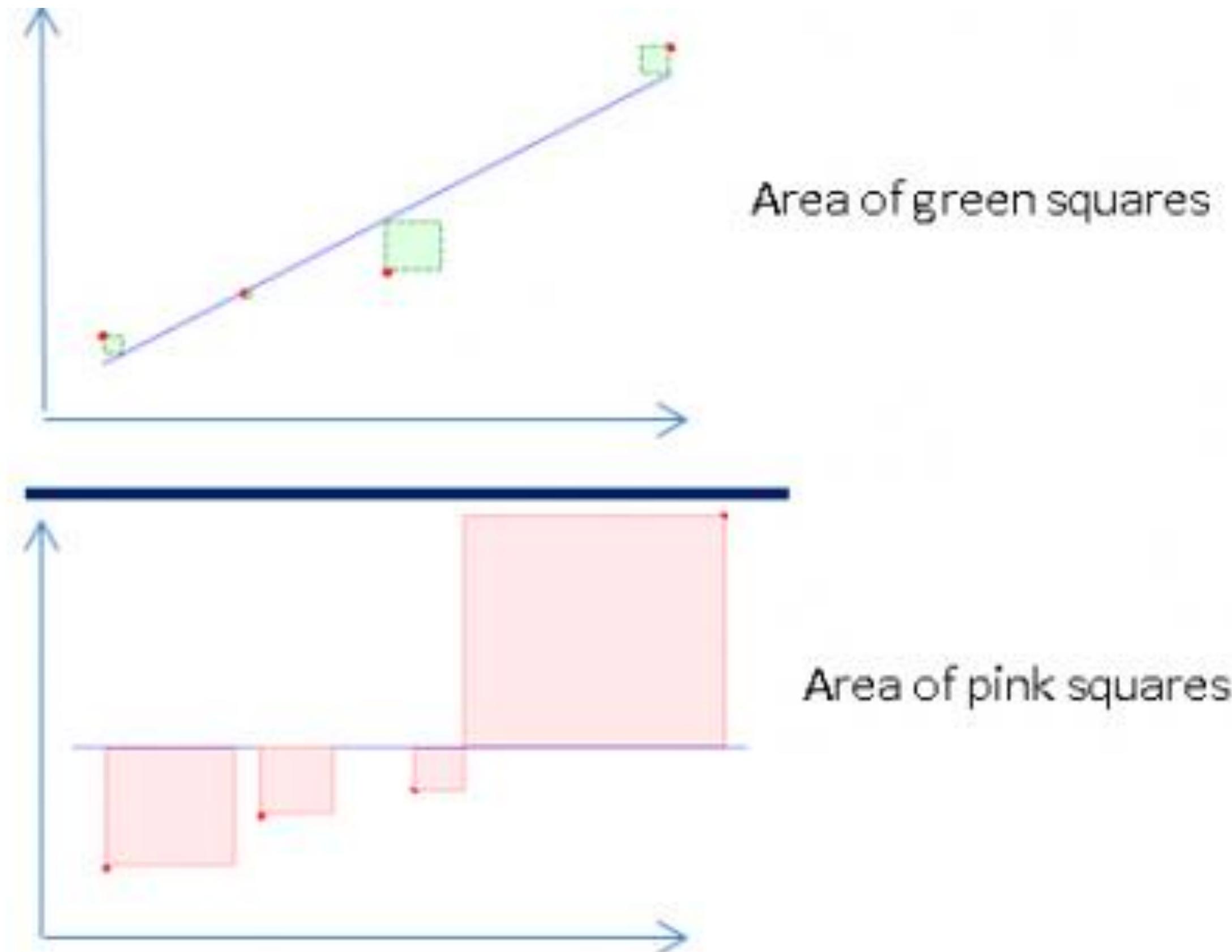
- N : data 샘플 수, y : 실제값, \hat{y} : 예측값
- MAE (mean absolute error) : $1/N \sum |y - \hat{y}|$
- MAPE (mean absolute percentage error) : $100/n \sum \left| \frac{y - \hat{y}}{y} \right|$
- MSE (mean squared error) : $100/n \sum (y - \hat{y})^2$
- 모두 작을 수록 좋음.

회귀 문제 메트릭 - R-squared

- N : data 샘플 수, y : 실제값, \hat{y} : 예측값, \bar{y} : y 값들의 평균
- SST (total sum of squares) : $\sum (y - \bar{y})^2$
- SSE (residual sum of squares): $\sum (y - \hat{y})^2$
- $R_{\text{squared}} = 1 - (\text{SSE}/\text{SST}) \rightarrow 1$ 에 가까울 수록 좋음.
 - 작은 SSE : 예측 오차가 작음.
 - 큰 SST : 예측이 어려움 (variance가 큼)

R-squared

$$R^2 = 1 - \frac{\text{Area of green squares}}{\text{Area of pink squares}}$$



모델 개발의 목적

- 지금까지 기계학습 모델들의 학습 프로세스에 대해 살펴보았음.
- 다양한 머신러닝 모델들이 존재하지만 넓은 관점에 볼 때 하나의 함수로 볼 수 있음.
- 최종적인 목적은 학습된 모델을 실제 서비스에 적용하는 것에 있음.
- 학습된 모델이 좋은 모델인지 어떻게 평가/예상 할 수 있을까?

모델의 일반화 성능

- 온실 속에서 키운 모델을 온실 밖으로 보냈을때 잘 할 수 있을까?
- 모델의 일반화 성능(generalization performance)이란?
 - 이전에 보지 못한 새로운 데이터를 적용시 모델이 갖는 성능
 - a) 실제 서비스에 적용 후 데이터 분석 → 가능하지만 단점이 많음.
 - b) 실제 서비스 적용 전에 일반화 성능을 유추 할 수 있는 방법?

대학 입시 과정을 통한 예시



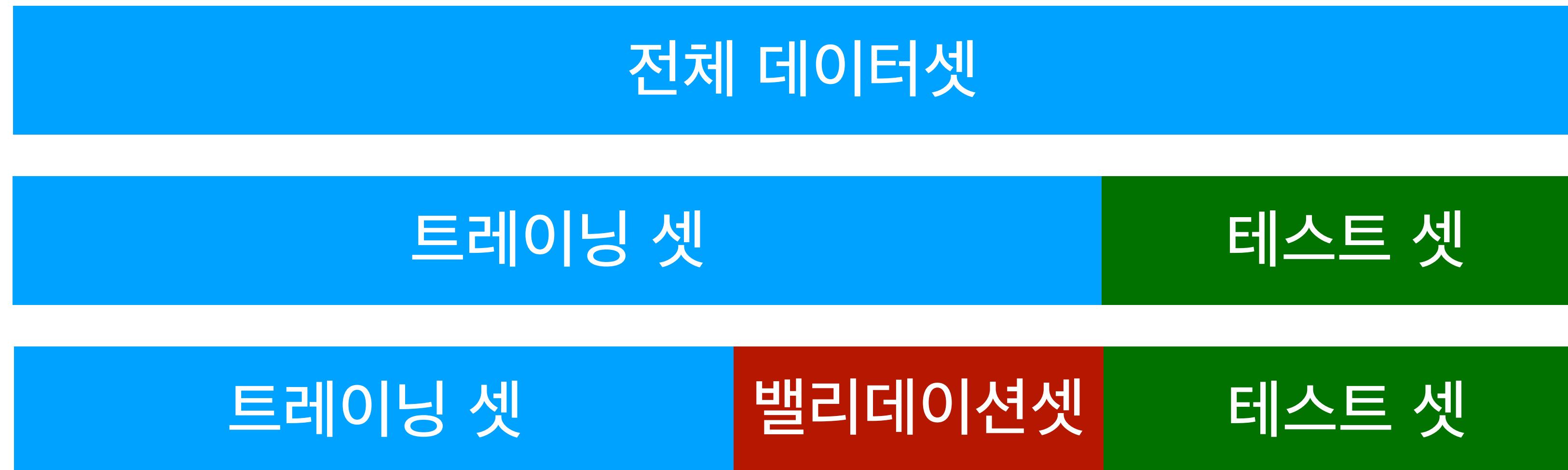
구성 요소들

- 수학능력평가시험 : 대학에서의 교육과정을 얼마나 잘 수학(修學)할 수 있는지 평가하는 것이 시험의 목적
- 수능 공부 : 수학능력평가 시험을 잘 보기위한 학습
- 모의 고사 : 수능 시험을 잘 볼 수 있도록 학습이 잘 진행되고 있는지 체크

대학 입시 과정과 기계학습 모델 학습 과정

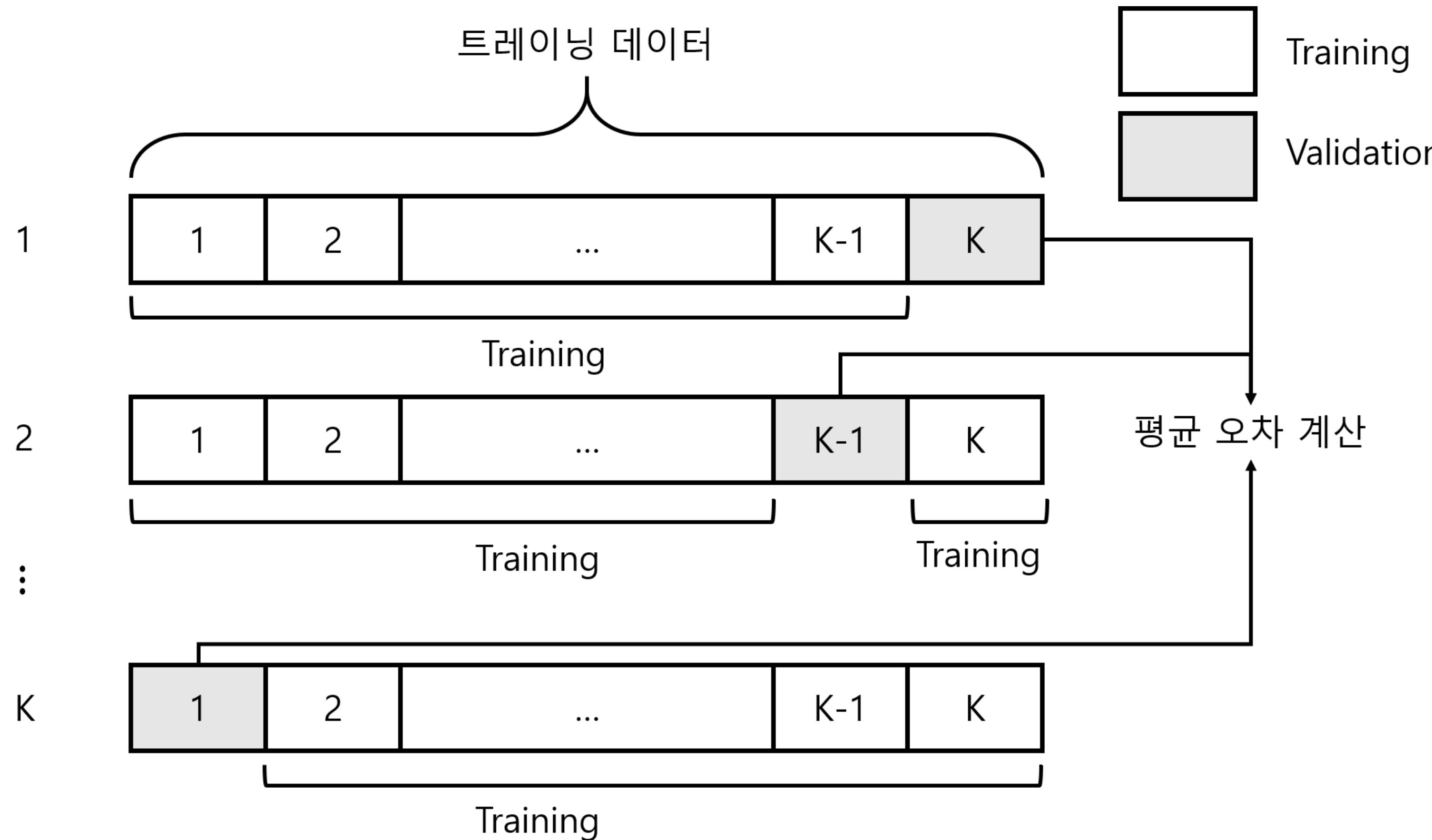
- 사람 : 모델
- 대학에서의 공부 : 실제 서비스 적용 시의 문제
- 수능 : 테스트 셋 (test set)
- 모의고사 : 밸리데이션 셋 (validation set)
- 공부 : 트레이닝 셋 (training set)
- 수능문제가 유출된다면? → Data leakage를 주의해야 함.

Hold-out validation

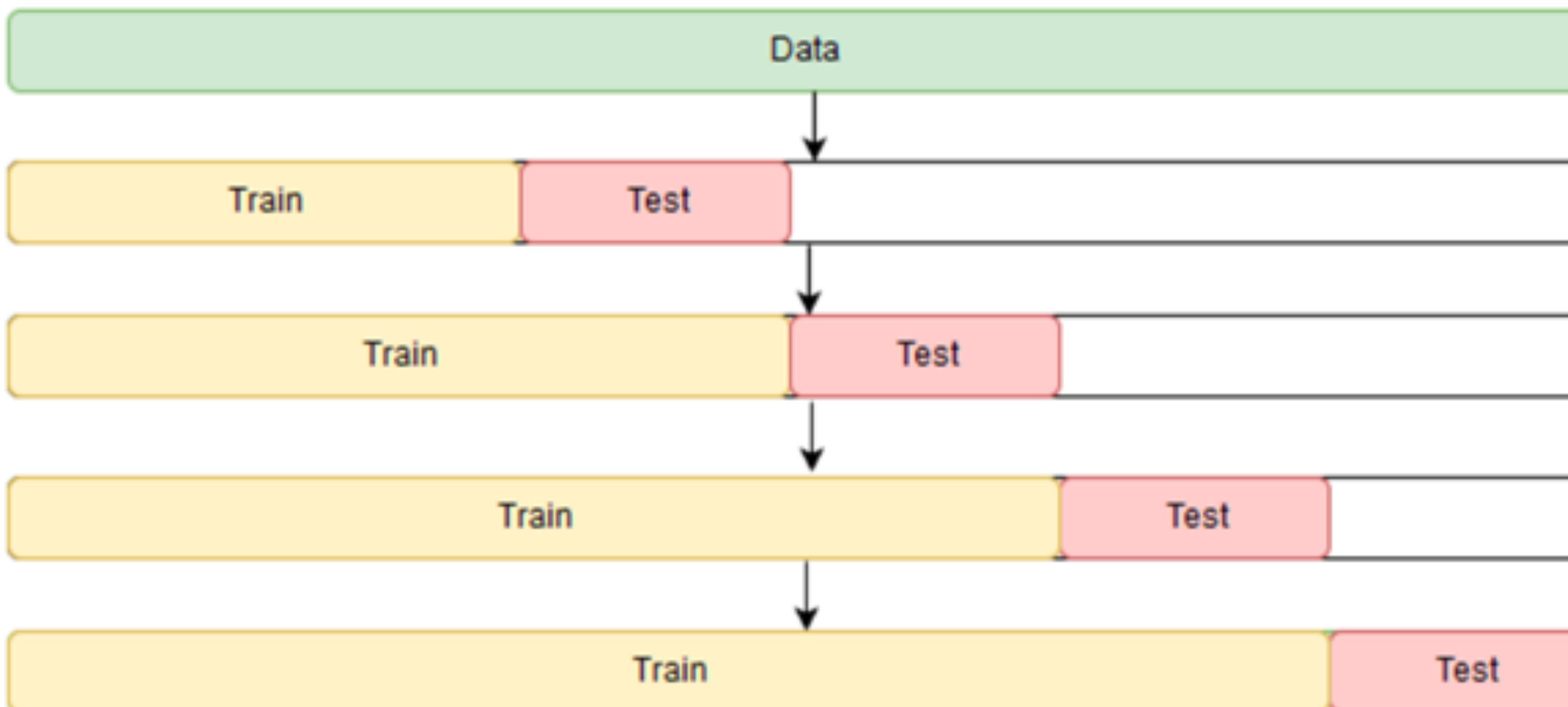


7:3, 6:2:2

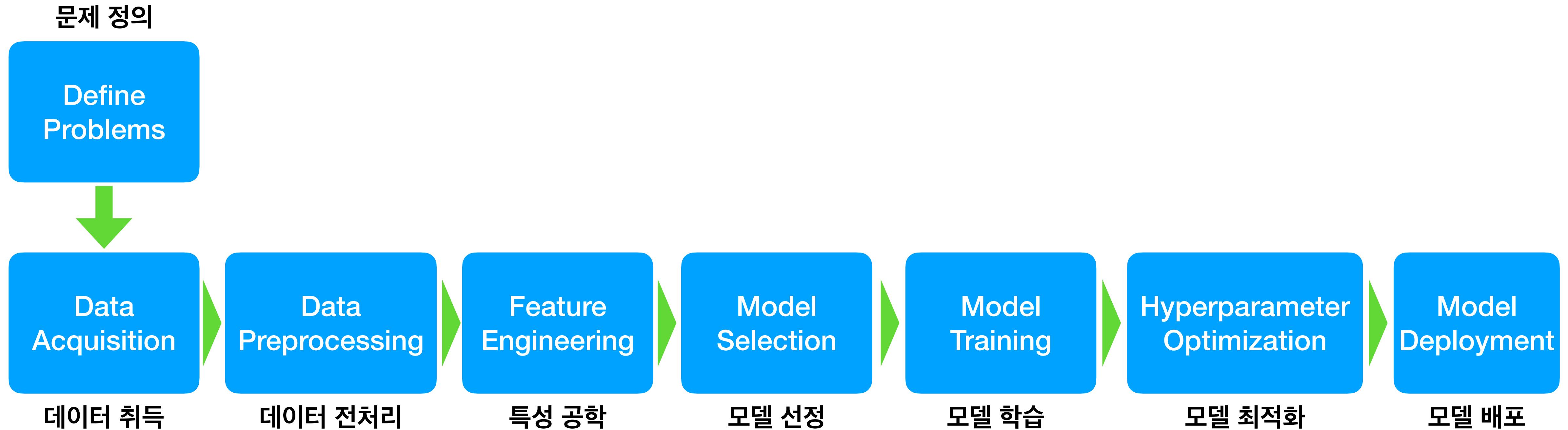
K-fold cross validation



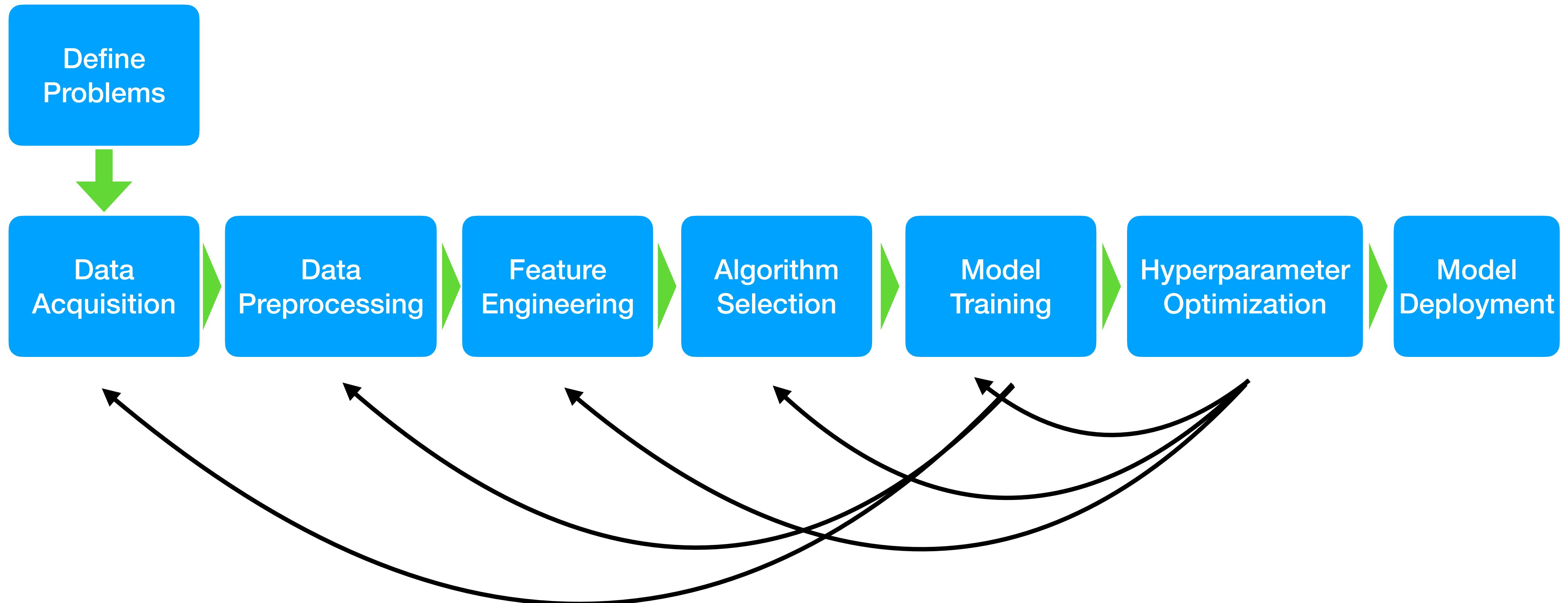
Time series split



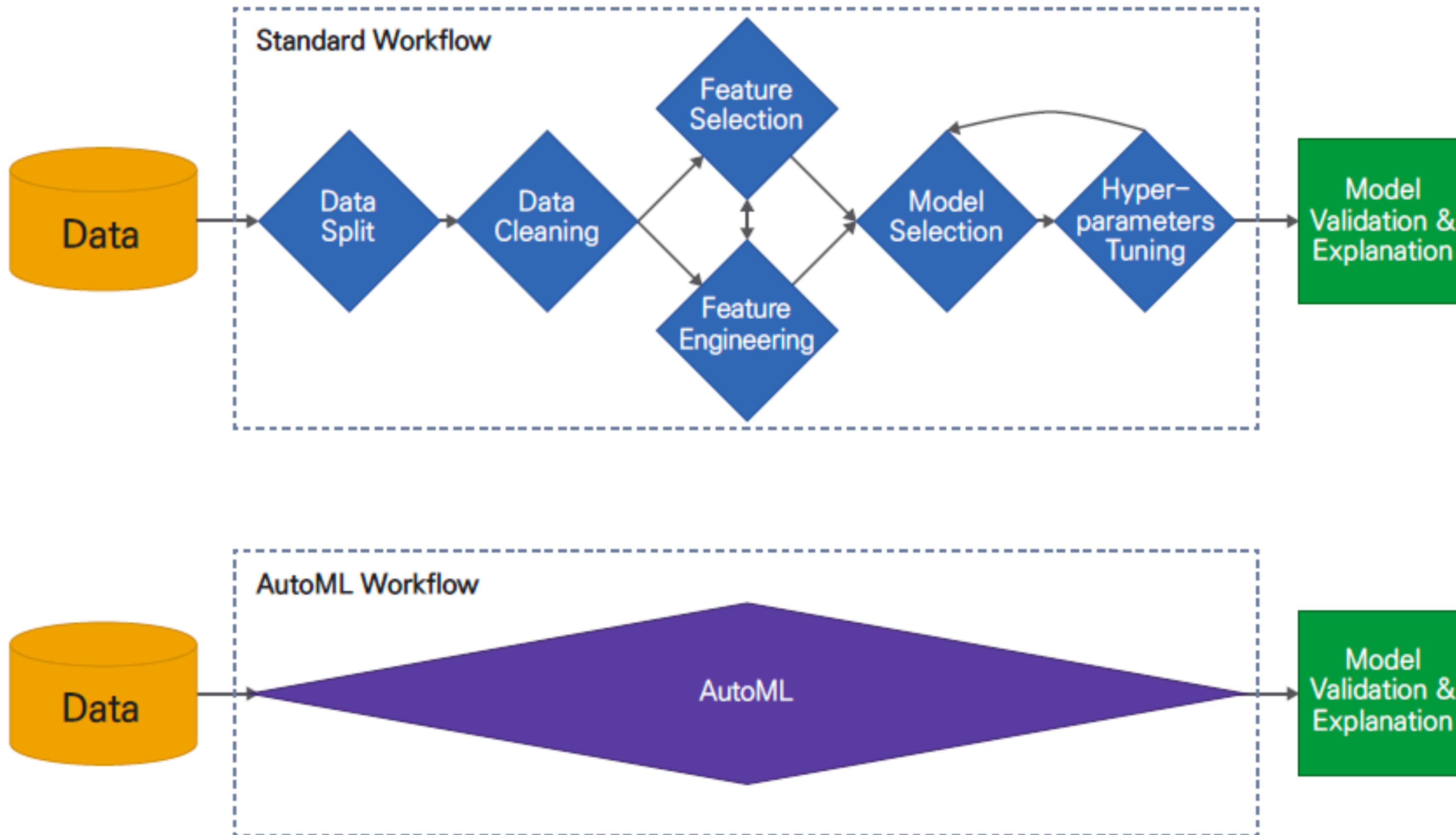
머신 러닝 모델 개발 프로세스



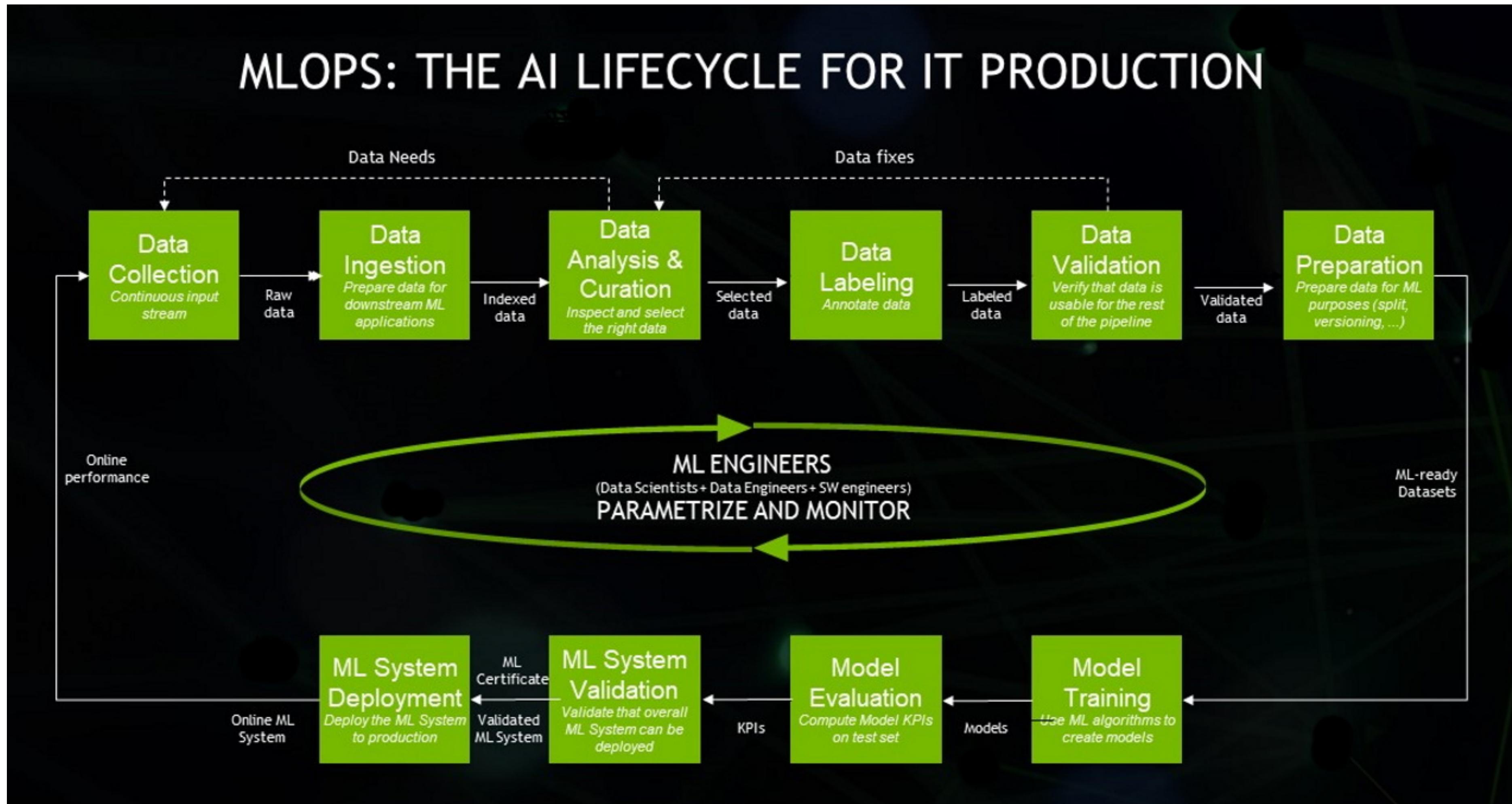
단방향이 아닌 과정...



Automation of Machine Learning



Auto ML & MLOps



Auto ML frameworks



Machine Learning for Humans



Google cloud AutoML



Google cloud AutoML



Chapter 3. Data, Task, Model

1. 데이터란 무엇일까?

데이터, 데이터, 데이터...

- 데이터 중심~, 데이터 기반~, 데이터 학습, 이미지 데이터, 자연어 데이터, 등등등
- 데이터란 무엇일까?
 - a collection of discrete values that convey information.

직관적인 이해를 위해서

- 데이터는 어떤 공간 상의 점 (point)
- 다양한 유형의 데이터
 - 표
 - 이미지
 - 오디오
 - 그래프
- 하지만 모두 어떤 공간상의 점으로 표현할 수 있음.

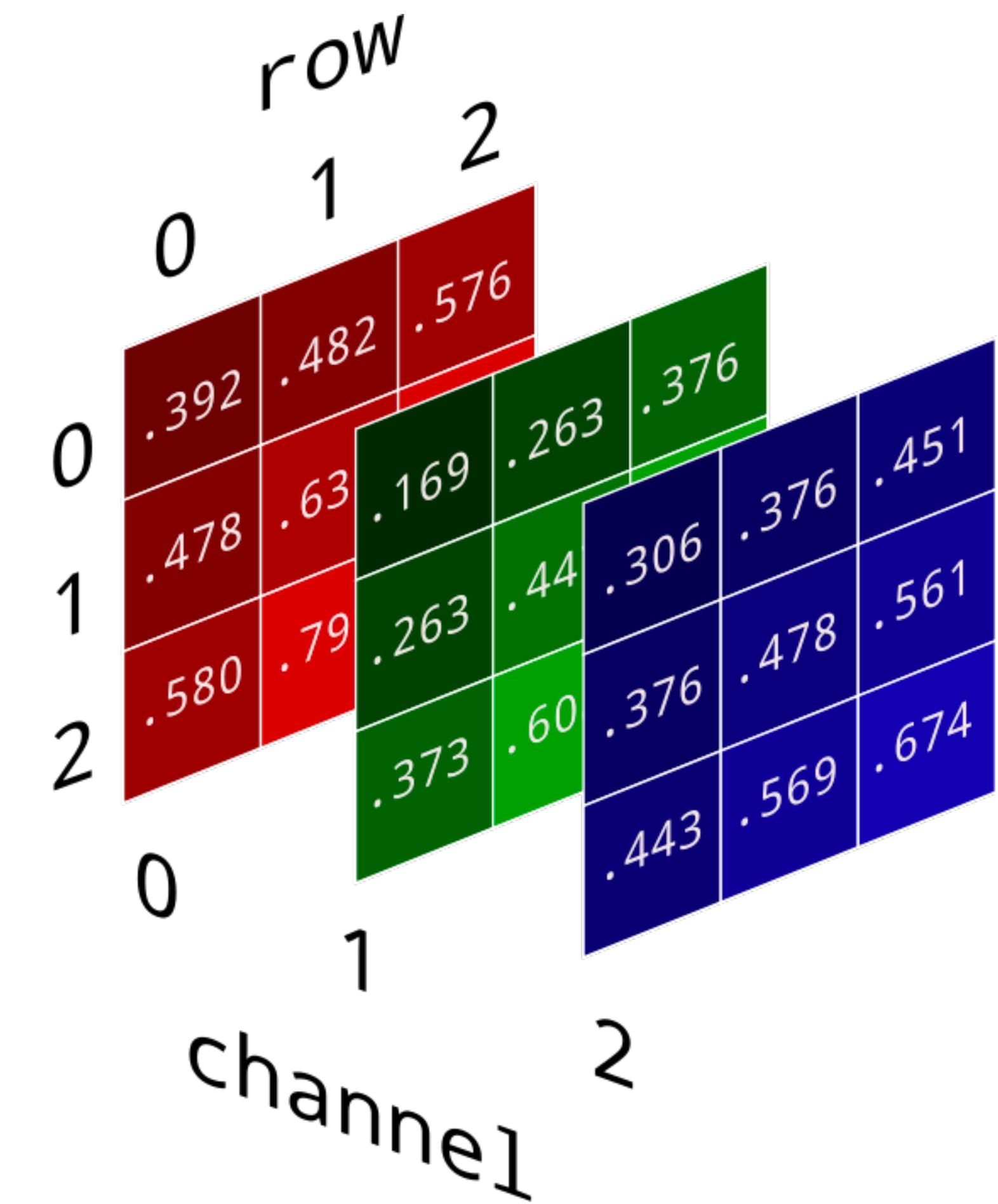
Tabular data

SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5.0	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa

Iris dataset

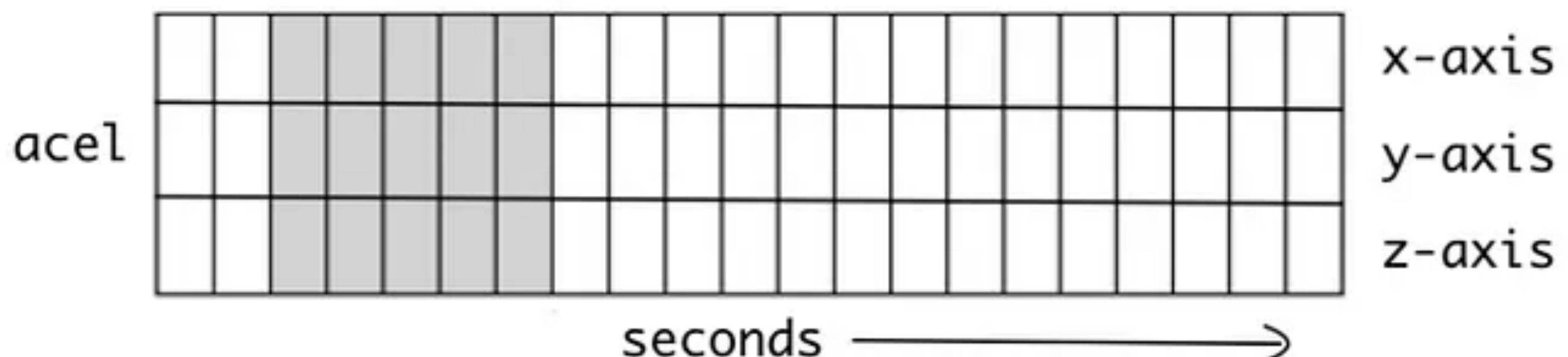
Image data

- 사진을 찍는다.
- 각 지점 (pixel)의 RGB 채널의 값이 기록
- [Height, Width, Channel]
- 3×3 매트릭스가 3개 >> 27차원 공간 상의 점
- 컴퓨터 화면 스크린샷? $1024 \times 768 \times 3$ 차원 공간 상의 점
- 스크린샷이 여러개라면? $[N \times H \times W \times C]$



Time-series data

- 특정 시점간격으로 데이터 수집
- 만약 1분 간격으로 온도와 습도, 압력을 수집한다면?
- 3차원 벡터가 1분 간격으로 생성, 1시간 동안의 데이터는? 60×3 매트릭스
- 시계열 또는 시퀀스 데이터는 형태는 $[N \times L \times D]$



Tabular data의 대표 구성 요소

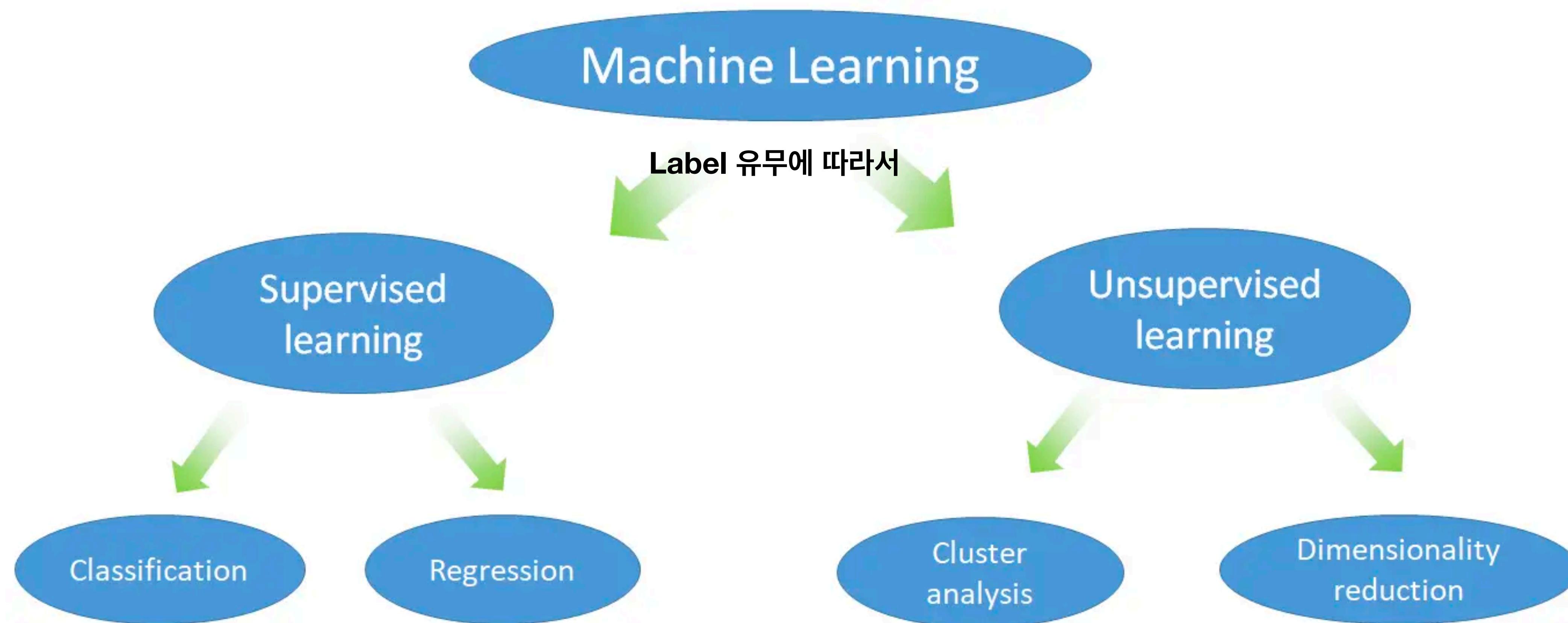
- Row : sample, Column : feature
- Numerical data : 연속적인 수치, ex) 온도, 무게, 가격 등
- Categorical data : 범주형 변수, ex) 성별, 제품 분류, 소속 등
- Datetime data : 날짜와 시간을 나타내는 데이터
- 데이터 별로 전처리가 필요

2. 데이터로 어떤 문제를 풀 것인가?

다양한 현실세계의 문제들

- 일기예보 : 내일의 날씨에 대한 정보를 예측
- 이상탐지 : 객체가 정상인지 비정상인지를 판별
- 고객 세분화 : 고객의 정보를 바탕으로 유사 특성을 갖는 고객 그룹을 생성
- 자율운전 : 영상을 바탕으로 자동차 조작

학습 방식에 따른 분류



Self-supervised learning

Labeled data is limited and often expensive

Using unlabeled data is critical

Self-supervision to the rescue!

Two-step training process:

- pre-train on unlabeled data
- fine-tune on labeled data

Supervised learning



Model

Labeled data

Self-supervised learning



Model

Unlabeled data

Semi-supervised learning



Model

Labeled data



태스크의 분류

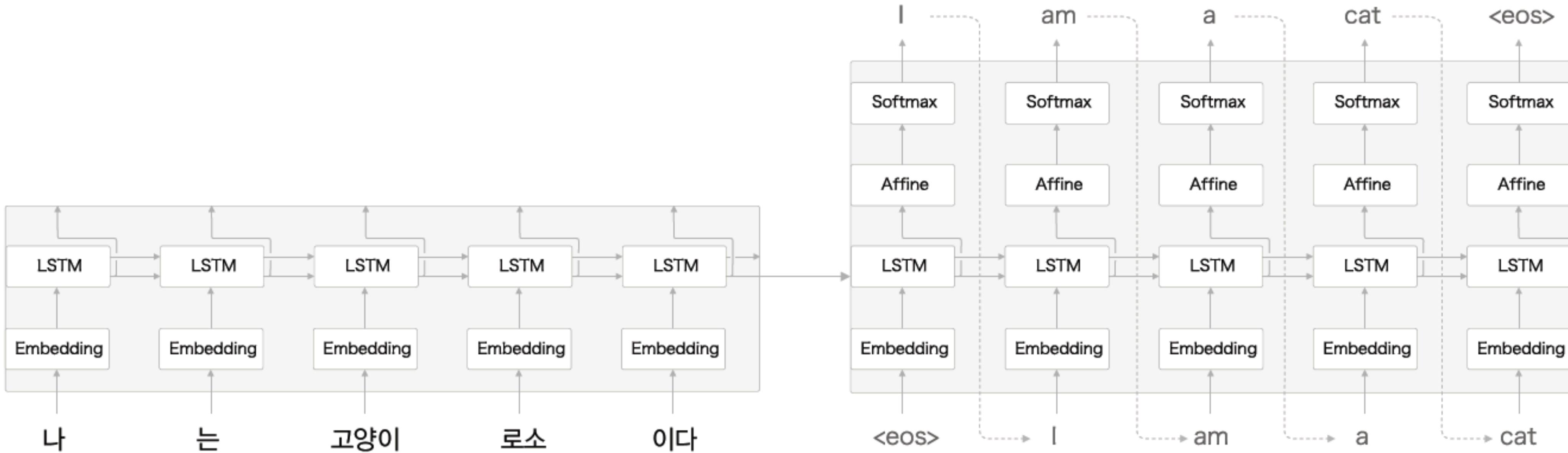
- **분류, Classification** : 출력 결과가 특정 클래스로 분류하는 문제인 경우
- **회귀, Regression** : 출력 결과가 어떤 변수에 대한 값을 예측하는 문제인 경우
- **클러스터링, Clustering** : 데이터를 그룹화 하고싶은 경우
- **차원축소, Dimension reduction** : 고차원의 데이터를 저차원으로 변환

어떻게 문제를 태스크로 모델링할까?

- 기본적인 두 가지 태스크
 - 회귀 : 실수값 출력 \rightarrow MSE 최소화
 - 분류 : 클래스로 분류, $[0, 0, 1, 0]$: 3 번째 클래스에 속할 확률이 100%.
 \rightarrow Cross entropy 최소화
- 다른 문제로의 확장?
 - 위의 두가지 태스크를 수행하는 문제로 모델링

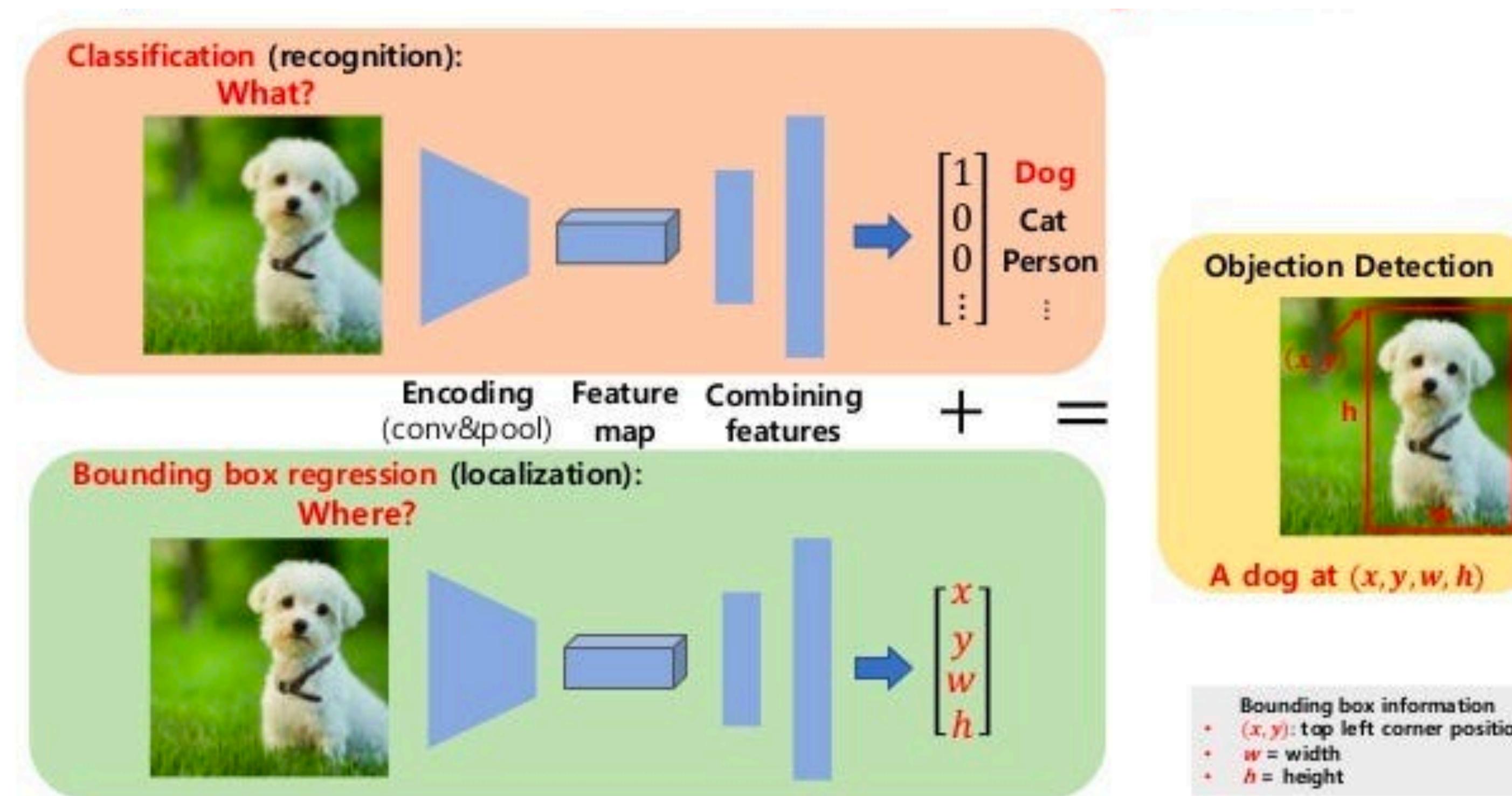
자연어 처리 (번역)

- 문장을 입력으로 받아 순차적으로 다음 단어를 예측하는 **분류 문제**
(각 단어를 클래스로)



컴퓨터 비전 (오브젝트 디텍션)

- 이미지를 입력 받아 객체가 존재하는 바운딩 박스의 좌표를 예측하는 **회귀 문제**
- 바운딩 박스 내의 객체에 대한 분류를 수행하는 **분류 문제**



3. 어떤 모델을 쓸 것인가?

머신러닝 개발은 논술문제

- 문제를 모델링하고 모델을 사용하는 것에는 정답이 없다.
 - 1) 시계열 데이터를 그래프로 표현하고 2D CNN 적용.
 - 2) MNIST 숫자 분류 문제를 Regression 모델로 0~9 사이 값을 직접 예측.
- 모범 답안은 있을 수 있음.
 - 혹은 납득할 수 있는 근거 또는 이유를 설명할 수 있어야 함.
 - 다양한 문제 풀이 경험이 필요.
 - 오토ML이 이러한 근거를 보강 할 수 있음.

다양한 머신러닝 모델들

- 선형 회귀 (Linear Regression)
- 로지스틱 회귀 (Logistic Regression)
- 의사결정 트리 (Decision Tree)
- 랜덤 포레스트 (Random Forest)
- 그래디언트 부스팅 트리 (**Gradient Boosting Tree**)
 - XGBoost
 - LightGBM
 - CatBoost
- 서포트 벡터 머신 (Support Vector Machine, SVM)
- 나이브 베이즈 (Naive Bayes)
- K-최근접 이웃 (K-Nearest Neighbors, KNN)
- 신경망 (Neural Networks)
 - 다층 퍼셉트론 (Multi-Layer Perceptron, MLP)
 - 컨볼루션 신경망 (Convolutional Neural Network, CNN)
 - 순환 신경망 (Recurrent Neural Network, RNN)
 - 트랜스포머 (Transformer)

Tabular data

감사합니다.