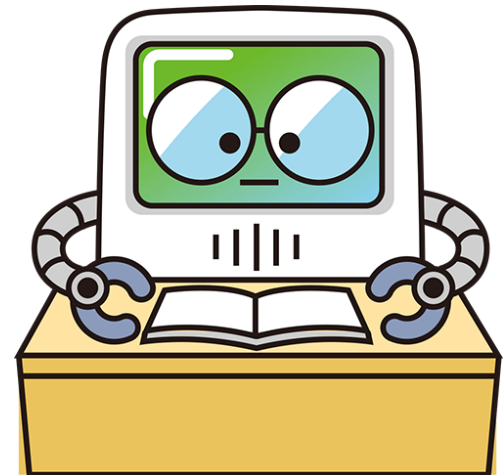


누구나 쉽게 배우는

# 인공지능 스타트



# 08

## 스스로 학습하는 머신러닝

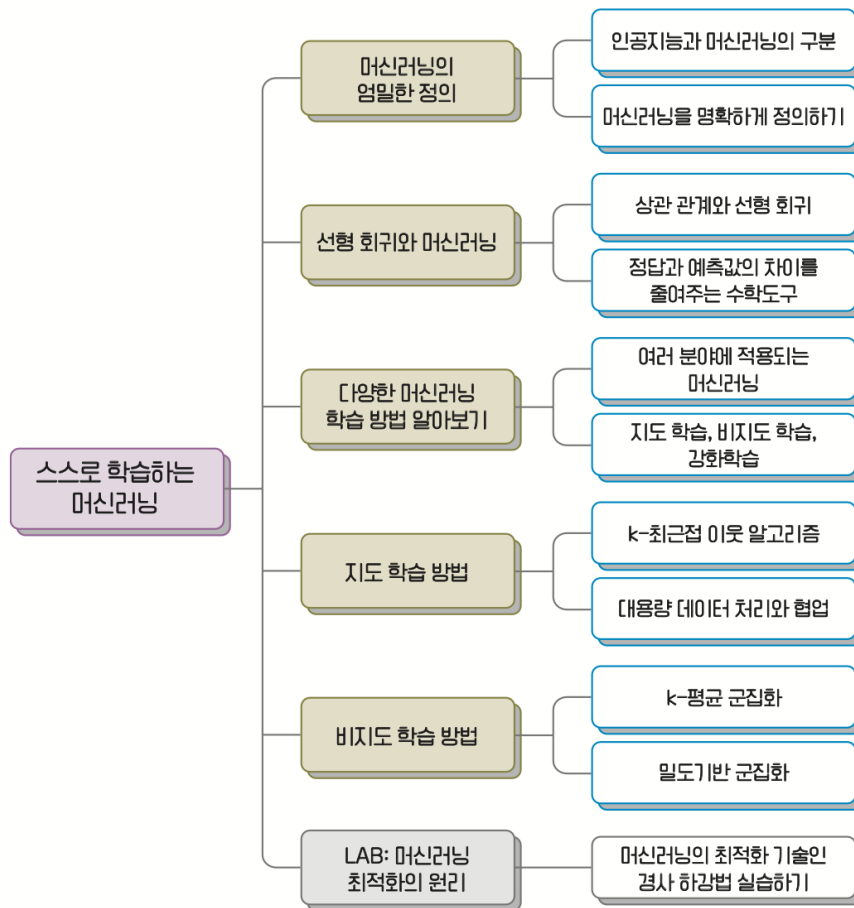
### 학습목차

- 01 머신러닝의 엄밀한 정의
- 02 선형 회귀를 통해 살펴보는 머신러닝
- 03 다양한 머신러닝 학습 방법을 알아보자.
- 04 다양한 지도 학습 방법을 알아보자.
- 05 비지도 학습 방법을 알아보자.
- LAB** 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

요약  
연습 문제  
심화 토의 문제

“우리 시대의 가장 중요한 범용 기술은 인공지능, 특히 머신러닝이다.”

MIT 슬론경영대학원, 에릭 브린올프슨 교수



### 학습목표

- 머신러닝을 이해하고 머신러닝 알고리즘이 어떻게 동작하는지 알아보자.
- 명시적 프로그래밍과 머신러닝의 차이를 이해하자.
- 선형 회귀를 통해 머신러닝의 학습 방법을 이해하자.
- 오차의 크기를 정량화하는 방법에 대해 알아보자.
- 머신러닝의 학습 방법에 따른 각각의 특성에 대해 알아보자.
- 비지도 학습의 대표적인 예로 군집화 기법에 대해 알아보자.

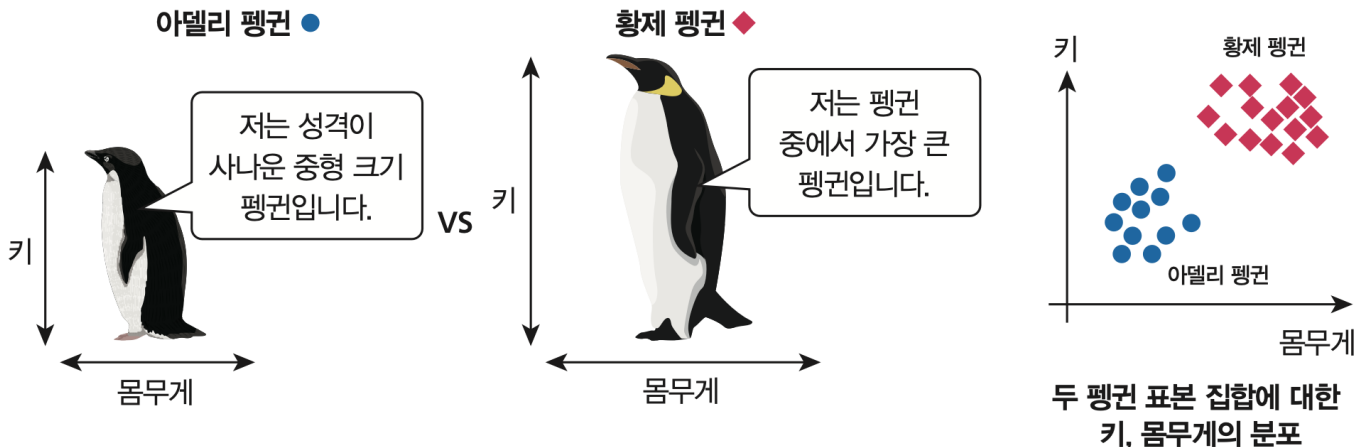
## 04 다양한 지도 학습 방법을 알아보자

### ■ k-최근접 이웃 알고리즘

- k-최근접 이웃 알고리즘은 **k-최근접 이웃** *k-Nearest Neighbor* 의 영문 약자를 따서 **k-NN 알고리즘**이라고도 하는데, 이 알고리즘은 특징 공간에 분포하는 데이터에 대하여 k개의 가장 가까운 이웃을 살펴보고 **다수결 방식으로 데이터의 레이블을 할당하는 분류 방식**이다.

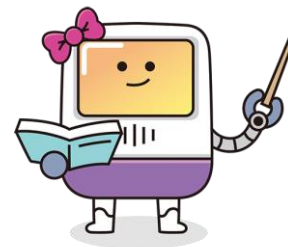
### ■ 예

- 중형 펭귄인 **아델리 펭귄**과 대형 펭귄인 **황제 펭귄**에 대하여 이를 분류하는 방법을 알아보자.
- 아델리 펭귄은 키가 평균 70cm가량에 몸무게가 6kg 정도이며, 반면 황제 펭귄은 키가 평균 120cm이며 몸무게는 30kg 정도이다.
- 여러 마리의 펭귄에 대하여 키와 몸무게의 표본을 모아서 2차원 공간에 표시한다면 오른쪽의 그림과 같이 분포하는 것을 볼 수 있을 것이다.



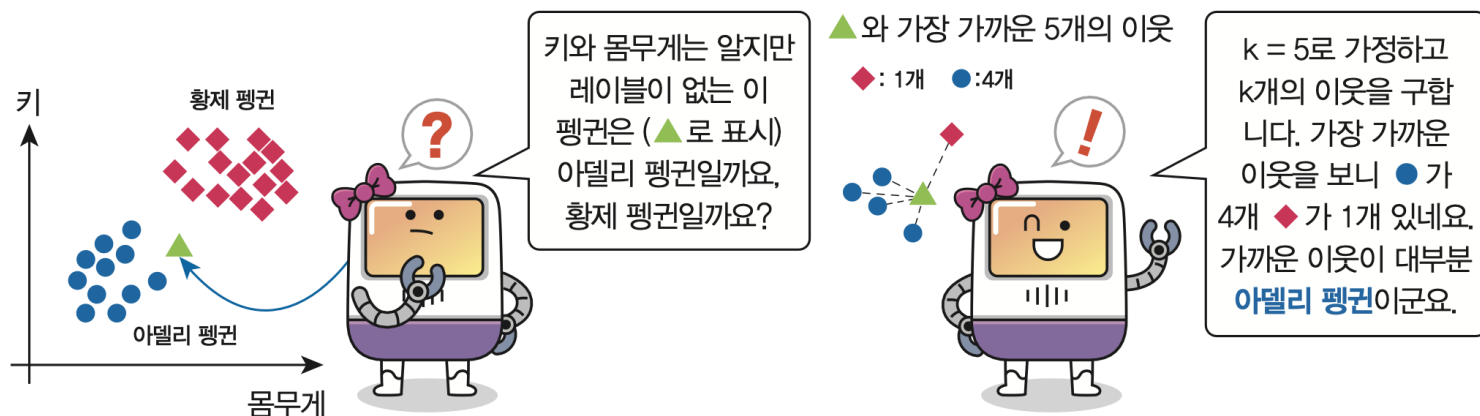
## 04 다양한 지도 학습 방법을 알아보자

- 아델리 펭귄의 표본은 파란색 동그라미로 표시하였으며, 황제 펭귄은 빨간색 다이아몬드로 표시하였다.
- 이 표본 공간에서 키와 몸무게는 속성 feature 이 되며, 황제 펭귄, 아델리 펭귄과 같은 정보는 레이블 label 이 된다.
- 이제 키와 몸무게라는 속성만을 가지고 있는 펭귄이 있는데 레이블이 없는 경우를 가정해 보자. 그리고 이 펭귄을 초록색 세모로 특징 공간에 표시해 보자.



## 04 다양한 지도 학습 방법을 알아보자

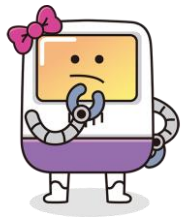
- 예시에서 k-최근접 이웃 알고리즘의 k값을 5로 가정해 보자. 이 알고리즘은 분류를 위해서 초록색 세모(▲)와 가장 가까운 5개의 이웃을 구한다.
- 가장 가까운 이웃을 살펴보니 파란색 동그라미가 4개, 빨간색 다이아몬드가 1개로 나타난다.
- 다수결에 의해 새로운 데이터의 레이블을 아델리 펭귄으로 표시하면 된다.
- 이 알고리즘은 학습 데이터 안에 존재하는 노이즈의 영향을 크게 받지 않으며, 학습 데이터의 수가 많을 때는 꽤 효과적인 분류 성능을 보여준다.



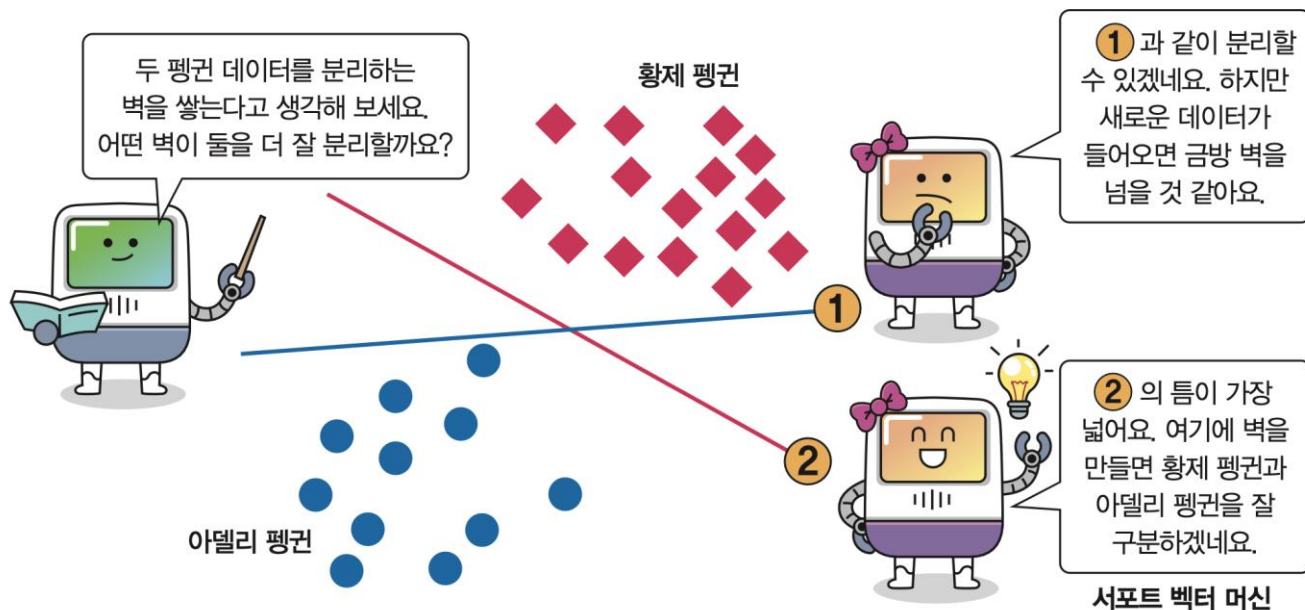
## 04 다양한 지도 학습 방법을 알아보자

### ■ 서포트 벡터 머신

- **서포트 벡터 머신** support vector machine 은 흔히 **SVM**이라고 부르는데, **딥러닝** deep learning 이 오늘날과 같이 인기를 얻기 전까지는 머신러닝 분야에서 가장 인기가 높은 데이터 분류 방법으로 사용되었다.
- 이전에 살펴본 황제 펭귄, 아델리 펭귄의 두 그룹 데이터가 있을 경우 이 펭귄 데이터를 분리하는 벽을 세운다고 가정해 보자. 파란색 선과 빨간색 선으로 분리될 수 있는데 이러한 방식의 **경계를 만들어서 데이터를 분리하는 분류 방법**이 바로 서포트 벡터 머신이다.



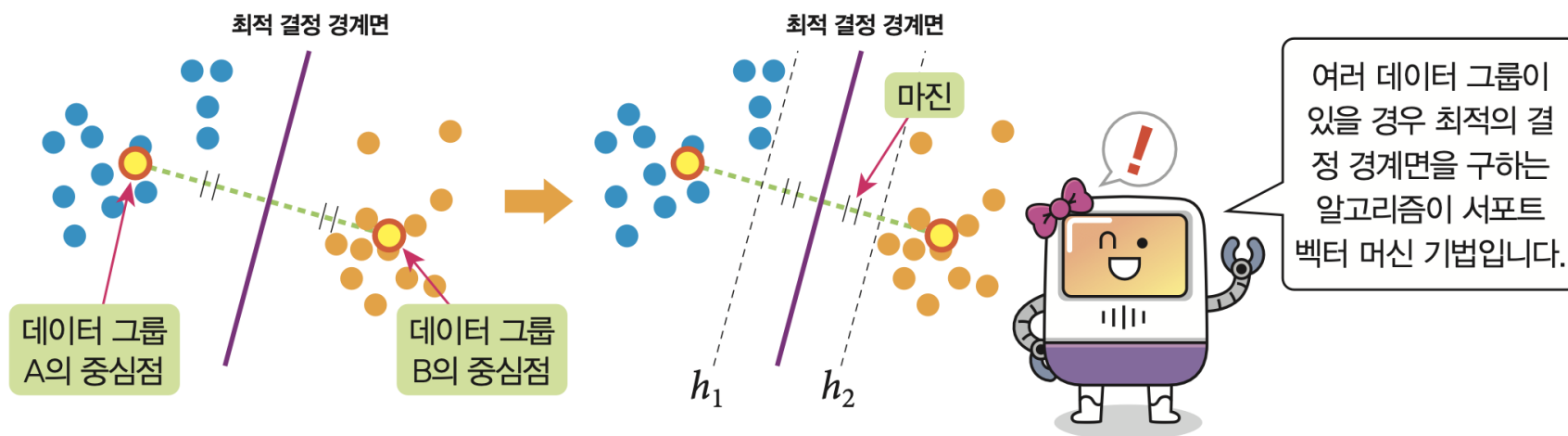
## 04 다양한 지도 학습 방법을 알아보자



- 그림을 살펴보면 ①과 ②의 두 직선은 모두 황제 펭귄과 아델리 펭귄 두 데이터 집합을 분리하는 벽으로 사용할 수 있다.
- 하지만 ①과 같은 벽은 두 데이터 집합 사이의 거리가 너무 좁아서 새로운 데이터가 들어올 경우 두 데이터를 구분하는 데 적합하지 않을 것이다.
- 반면 ②의 직선은 두 데이터를 구분하는 가장 넓은 틈을 가로지르기 때문에 두 데이터를 구분하는 데 적합할 것으로 보인다.
- 서포트 벡터 머신은 두 데이터 그룹을 나누는 평면 중에서 폭이 가장 넓은 것을 찾는 방법이다.

## 04 다양한 지도 학습 방법을 알아보자

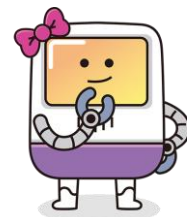
- 분리 평면은 **결정 경계**라고도 한다.
- 결정 경계면은 2차원에서는 직선이지만 그 이상의 차원에서는 가시화할 수 없는 평면으로 **초평면 hyperplane**이라고 부른다.
- 그리고 결정 경계와 서포트 벡터 사이의 거리를 **마진 margin**이라고 부른다.
- 이 결정 경계면을 구하는 방법을 자세히 살펴보면 그림과 같이 나타낼 수 있다.





## 04 다양한 지도 학습 방법을 알아보자

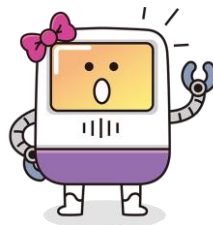
- 그림을 살펴보면 데이터 그룹 A와 B가 학습용으로 주어져 있다.
- 이때, 학습 알고리즘은 왼쪽과 같이 각 그룹의 중심점을 구한 후, 그 가운데를 지나는 최적의 결정 경계면을 구하여 두 데이터 그룹을 나누는 방법을 학습하게 된다.
- 그림과 같이 직선으로 나눌 수 있다면 선형 분류 모델을 적용하고, 직선으로 나눌 수 없는 경우 비선형 분류 모델을 적용한다.



## 04 다양한 지도 학습 방법을 알아보자

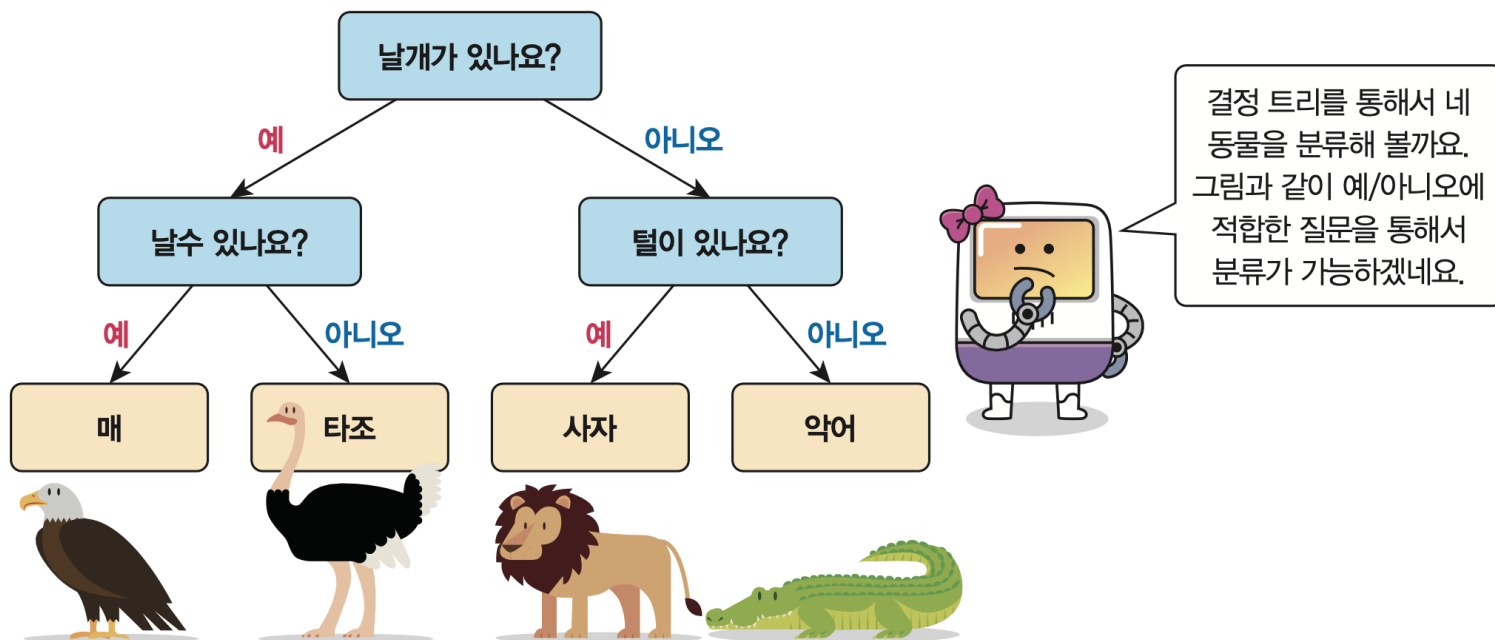
### ■ 결정 트리

- 머신러닝 알고리즘인 의사결정 트리 decision tree 는 분류와 회귀 작업, 다중 출력까지 할 수 있는 재주가 많은 알고리즘이다.
- 결정 트리는 스무고개를 하듯 예/아니오 질문을 이어가며 학습을 한다.



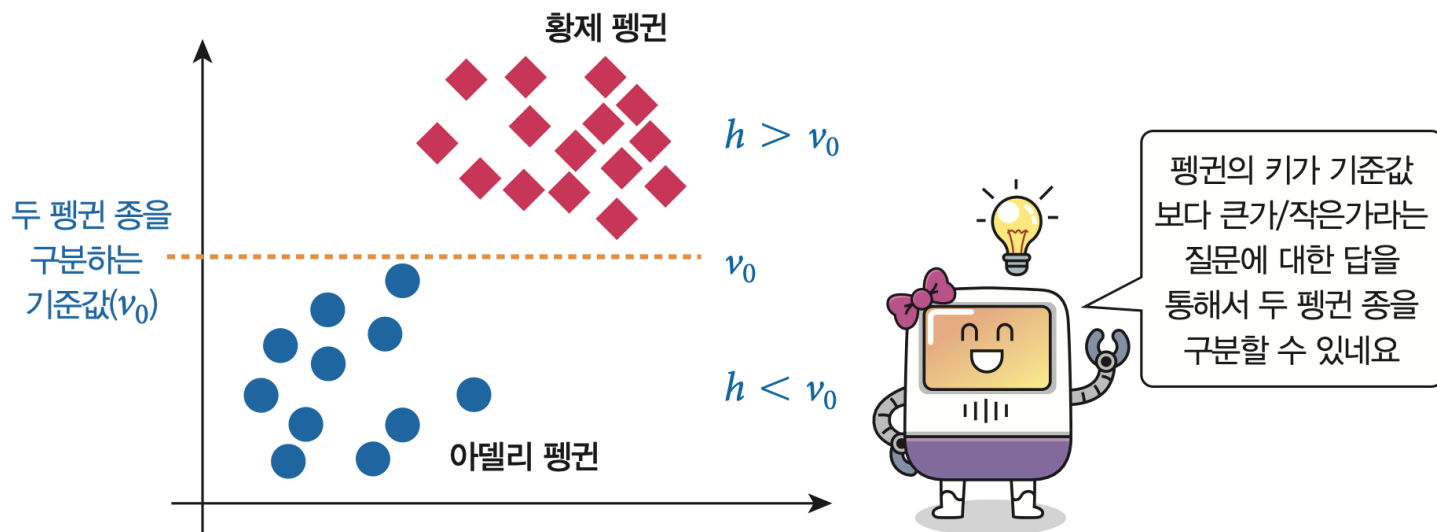
## 04 다양한 지도 학습 방법을 알아보자

- 예를 들어, 매, 타조, 사자, 악어를 구분하는 알고리즘을 가정해 보자.
- 알고리즘은 예/아니오로 대답할 수 있는 질문으로만 구성된다고 가정하자.
- 매와 타조는 날개가 있고, 사자와 악어는 날개가 없다. 따라서 '날개가 있나요?'라는 질문을 통해 [매, 타조] / [사자, 악어]의 두 그룹으로 나눌 수 있다.
- 매와 타조는 '날 수 있나요?'라는 질문으로 나눌 수 있고, 사자와 악어는 '털이 있나요?'라는 질문으로 나눌 수 있다.
- 이 결정 트리를 도식화하면 다음 그림과 같을 것이다.



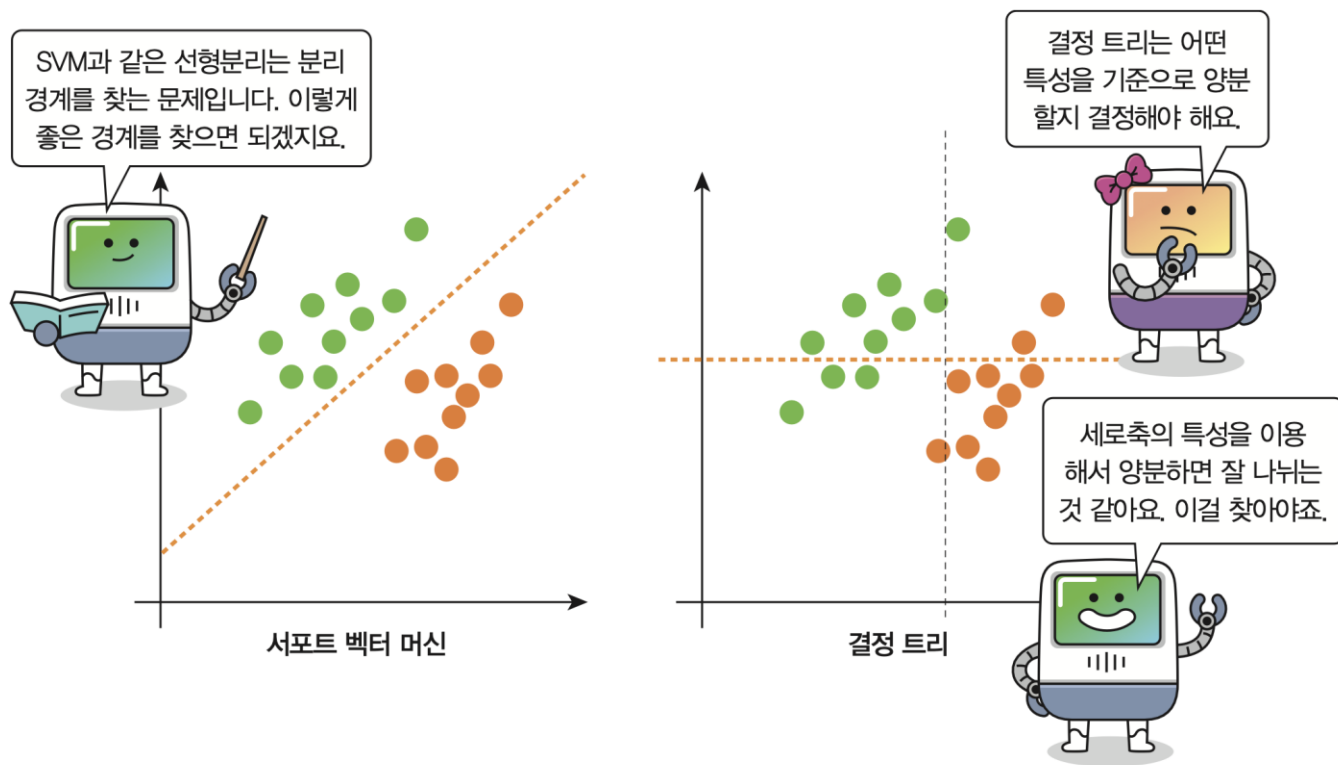
## 04 다양한 지도 학습 방법을 알아보자

- 이것은 데이터들을 트리 구조의 최초 시작점인 **루트 root** 에서 시작하여 차례로 **중간 노드**들을 거쳐 **단말 노드**에 배정하는 기능을 수행한다.
- 아델리 펭귄과 황제 펭귄의 두 펭귄 종의 몸통 높이(키)에 따라 그림과 같이 분포할 때, 이 펭귄 종을 구분할 수 있는 몸통 높이( $h$ )의 기준값  $v_0$ 가 존재한다고 가정하자.
- 이 경우 몸통 높이를 기준으로  $h > v_0$  일 경우 황제 펭귄으로,  $h < v_0$ 의 경우를 아델리 펭귄으로 분류할 수 있을 것이다.
- 이렇게만 본다면 결정 트리와 SVM과 같은 선형분리 문제는 동일한 문제인 것처럼 보인다.



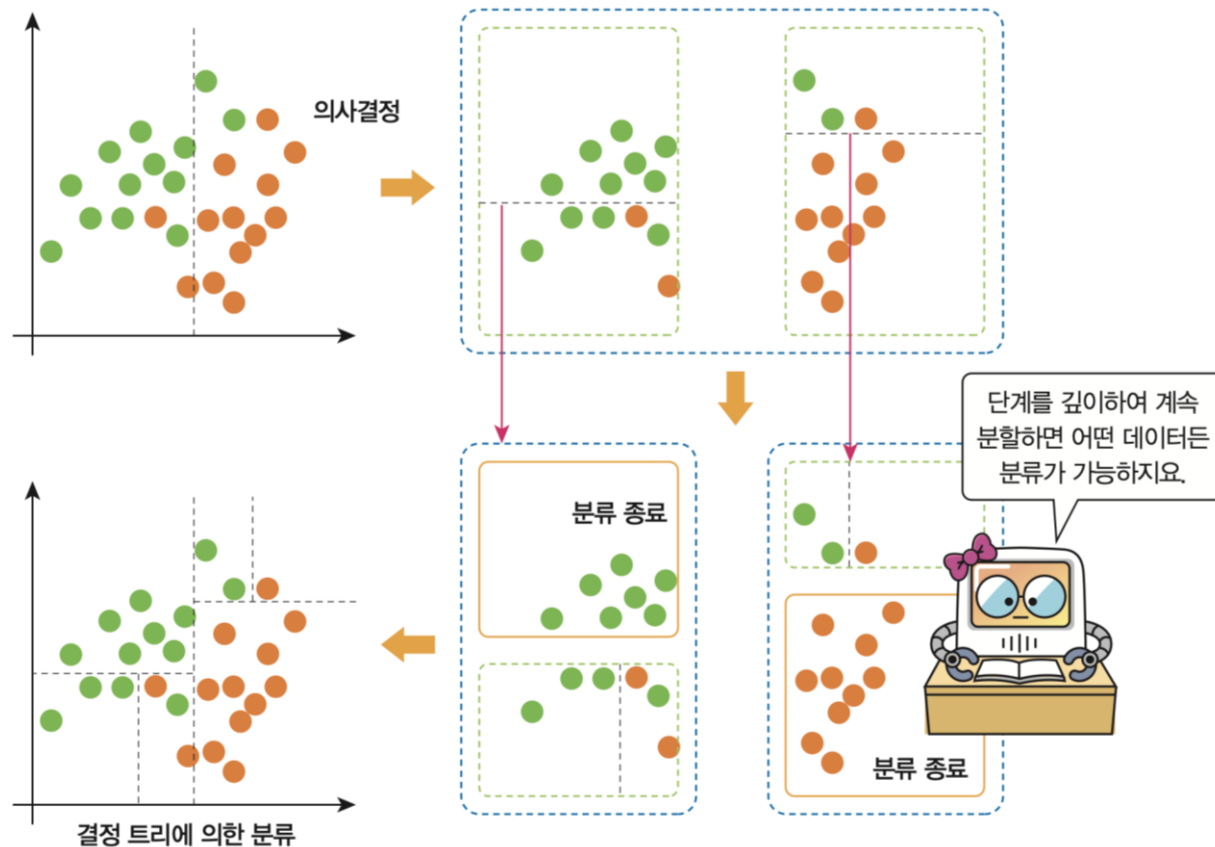
## 04 다양한 지도 학습 방법을 알아보자

- 선형분리는 아래 그림과 같이 특성 공간 내에서 자유롭게 선택할 수 있지만, 결정 트리는 하나의 특성을 기준으로 데이터를 양분해야 한다.
- 하지만 현실 세계의 데이터는 좀 더 복잡한 경우가 많아서 단순히 하나의 특성으로 깔끔하게 데이터를 양분할 수 없는 경우가 많다.
- 그래서 나누어진 각 영역에 대해 다시 다른 특성을 기준으로 양분하는 일을 해 나간다.



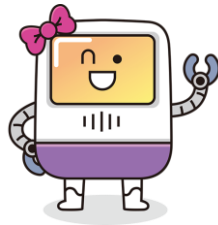
## 04 다양한 지도 학습 방법을 알아보자

- 이 단계를 계속 반복하면 어떤 데이터든지 분리할 수 있는 트리 구조를 만들 수 있을 것이다.
- 하지만 이 단계를 무한정 깊게 하는 것보다는 **가능한 얇은 단계에서 데이터가 분리될 수 있도록 하는 것이 좋다.**
- 가능한 얇은 단계에서 데이터를 잘 분리할 수 있는 방법은 데이터 분리를 가장 잘하는 좋은 특성을 찾아, 해당 특성 내에서 어떤 값을 기준으로 분리를 할 것인지를 결정하는 것이다.



## 05 비지도 학습 방법을 알아보자

- 비지도 학습의 대표적인 예가 군집화이다.
- 군집화는 영어로 **클러스터링** *clustering* 이라고 하는데 데이터가 주어졌을 때 비슷한 특징을 가진 데이터를 군집으로 분류하는 것을 말한다.
- 군집화 알고리즘 중에서 가장 단순한 방법 중의 하나인 **k-평균 군집화** *k-means clustering* 알고리즘에 대해 알아보자.



## 05 비지도 학습 방법을 알아보자

### ■ k-평균 군집화

- 예를 들어 으뜸 게임사는 자사의 핵심 게임인 '모두의 으뜸 게임'을 출시하고 이 게임을 하는 고객들의 성향을 분석하고자 한다.
- 으뜸 게임사는 게임을 하는 사용자를 다음과 같이 시간과 레벨에 따라 배치한 후 두 가지 성향의 사용자로 분리하려고 하는 것이다.

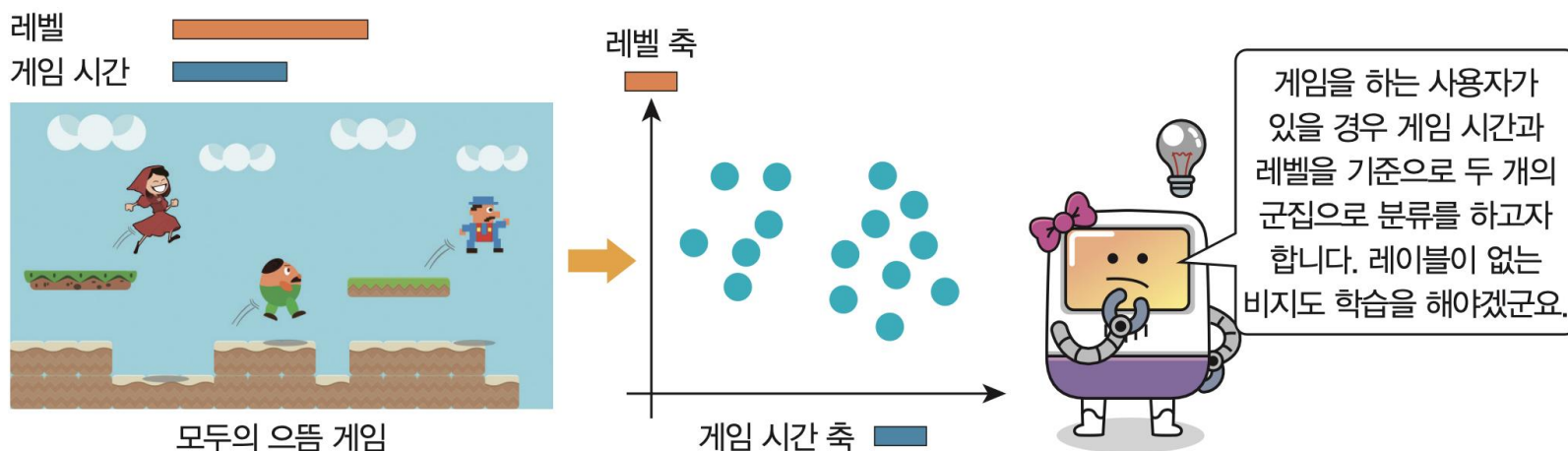
A 그룹(군집): “게임을 오래하며, 레벨이 높은 사용자.”

B 그룹(군집): “게임을 짧게하며, 레벨이 낮은 사용자.”



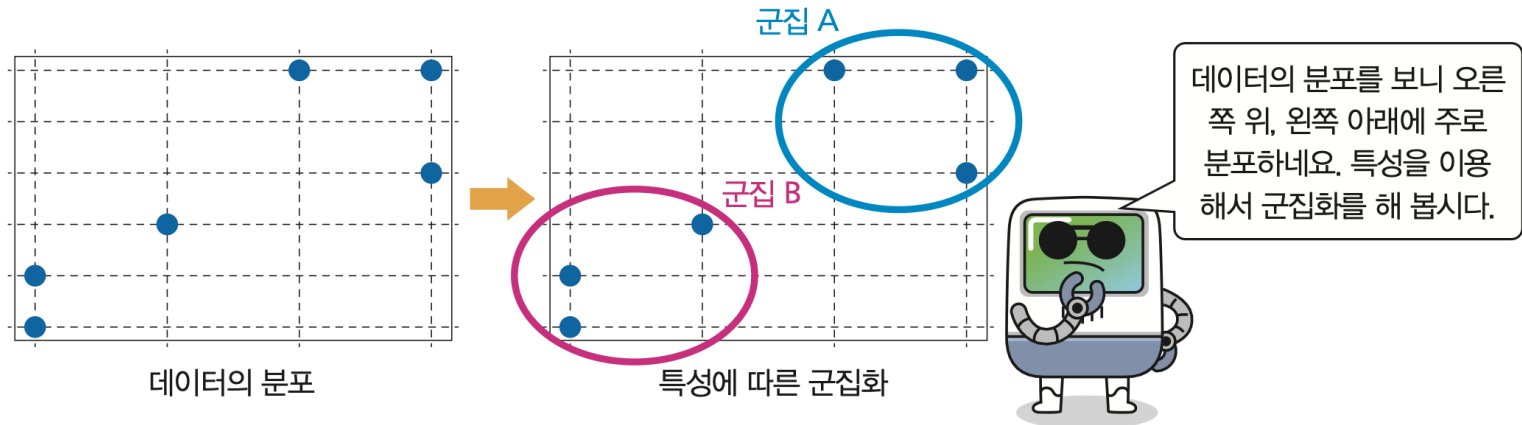
## 05 비지도 학습 방법을 알아보자

- 게임 사용자의 전체 **게임 시간**과 **레벨**을 특성 공간에 배치하였으나 아무런 레이블이 없는 상태이다.
- 따라서 이를 군집으로 나누는 것은 대표적인 비지도 학습에 해당한다.



## 05 비지도 학습 방법을 알아보자

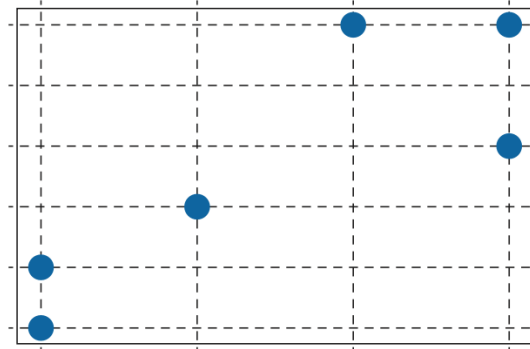
- 데이터의 분포를 보니 오른쪽 위(군집 A), 왼쪽 아래(군집 B)에 주로 분포하는 것을 알 수 있다.



- 우리가 사용할 k-평균 알고리즘은 원리가 단순하고 직관적이며 성능이 좋은 군집화 알고리즘이다.

## 05 비지도 학습 방법을 알아보자

2 차원 특징 공간에 데이터가 분포

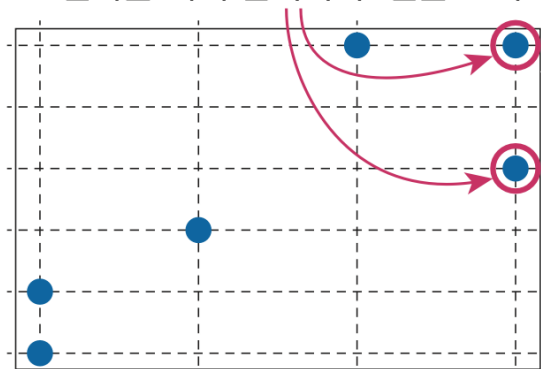


(a)

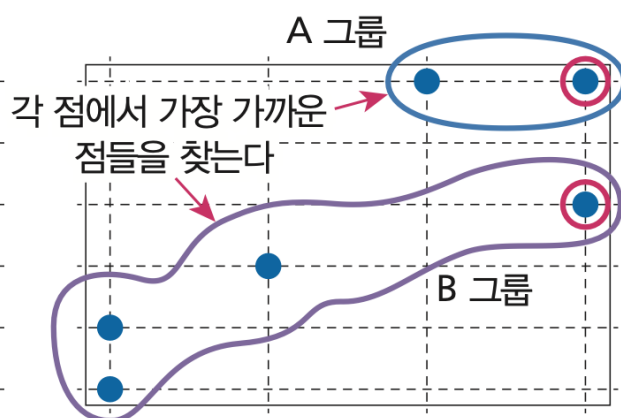
- 우선 다음 그림 (a)와 같이 2차원 공간상에 분포하는 6개의 점들이 있고  $k$  값이 2인 경우를 가정하여,  $k$ -평균 알고리즘이 수행되는 과정을 살펴보도록 하자.

## 05 비지도 학습 방법을 알아보자

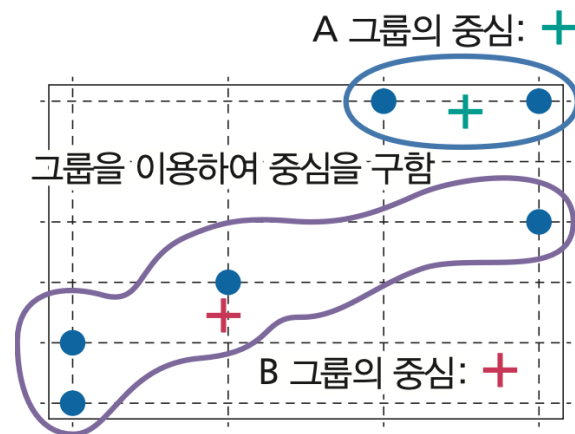
군집화를 위해 임의의 두 점을 선택



(b)



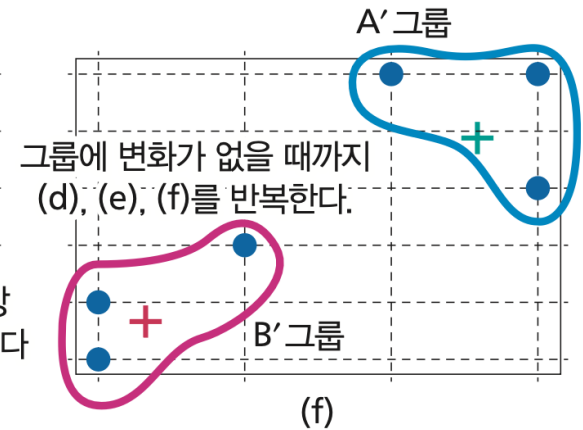
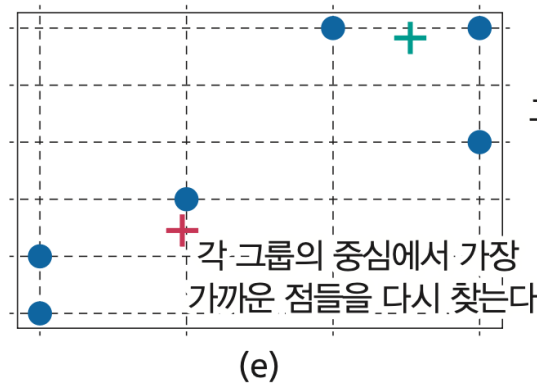
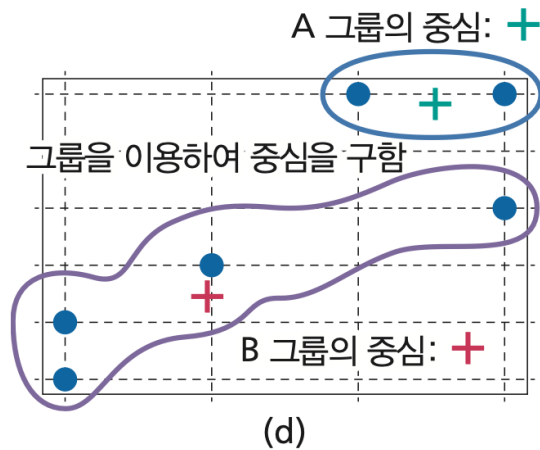
(c)



(d)

- $k$ 가 2인 경우 (b)에 표시된 것과 같이 6개의 점들 중에서 빨간색 동그라미로 표시된 임의의 두 점을 선택하는 것으로 알고리즘이 시작된다.
- 이 두 점과 가장 가까운 점을 그룹(군집)으로 묶을 경우 그림 (c)와 같이 A 그룹, B 그룹으로 묶이게 된다.
- 이때 A 그룹에는 2개의 점이 있으며, B 그룹에는 4개의 점이 있는데 이 점들의 중심점을 구하면 그림 (d)와 같이 표시할 수 있다.

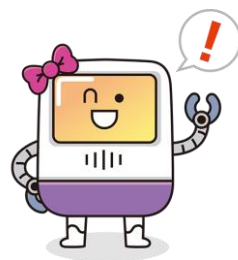
## 05 비지도 학습 방법을 알아보자



- 이제 그림 (e)와 같이 +로 표시된 A 그룹의 중심점과 +로 표시된 B 그룹의 중심점을 이용하여 모든 점과의 거리를 구한 후 다시 그룹을 만들어 A'와 B' 그룹을 만들도록 한다(그림 (f)).
- 이제 이 과정을 (d)단계에서부터 다시 반복하여 이전의 군집 배정과 결과가 같거나, 갱신이 거의 발생하지 않을 경우 알고리즘을 종료한다.
- 아래 예시에서는 (f)단계에서 종료될 것이다.

## 05 비지도 학습 방법을 알아보자

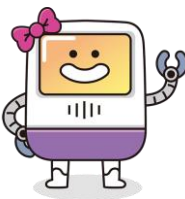
- k-평균 알고리즘은 비교적 단순하며 거리와 중심점을 사용하는 그룹화 방법으로 범주에 대한 사전 정보가 없을 경우 주어진 관측값들 사이의 거리를 측정하여 유사성을 입증하는 방법이다.
- 이 방법은 매우 단순하며 짧은 계산 시간 안에 대용량의 데이터에 대하여 적용할 수 있다는 장점이 있으나, 군집의 수  $k$ 를 사전에 정의해 주어야 한다는 한계가 있다.
- 그리고 데이터들의 분포를 한정된 크기 내의 값으로 미리 전처리해야 한다는 한계도 있다.



## 05 비지도 학습 방법을 알아보자

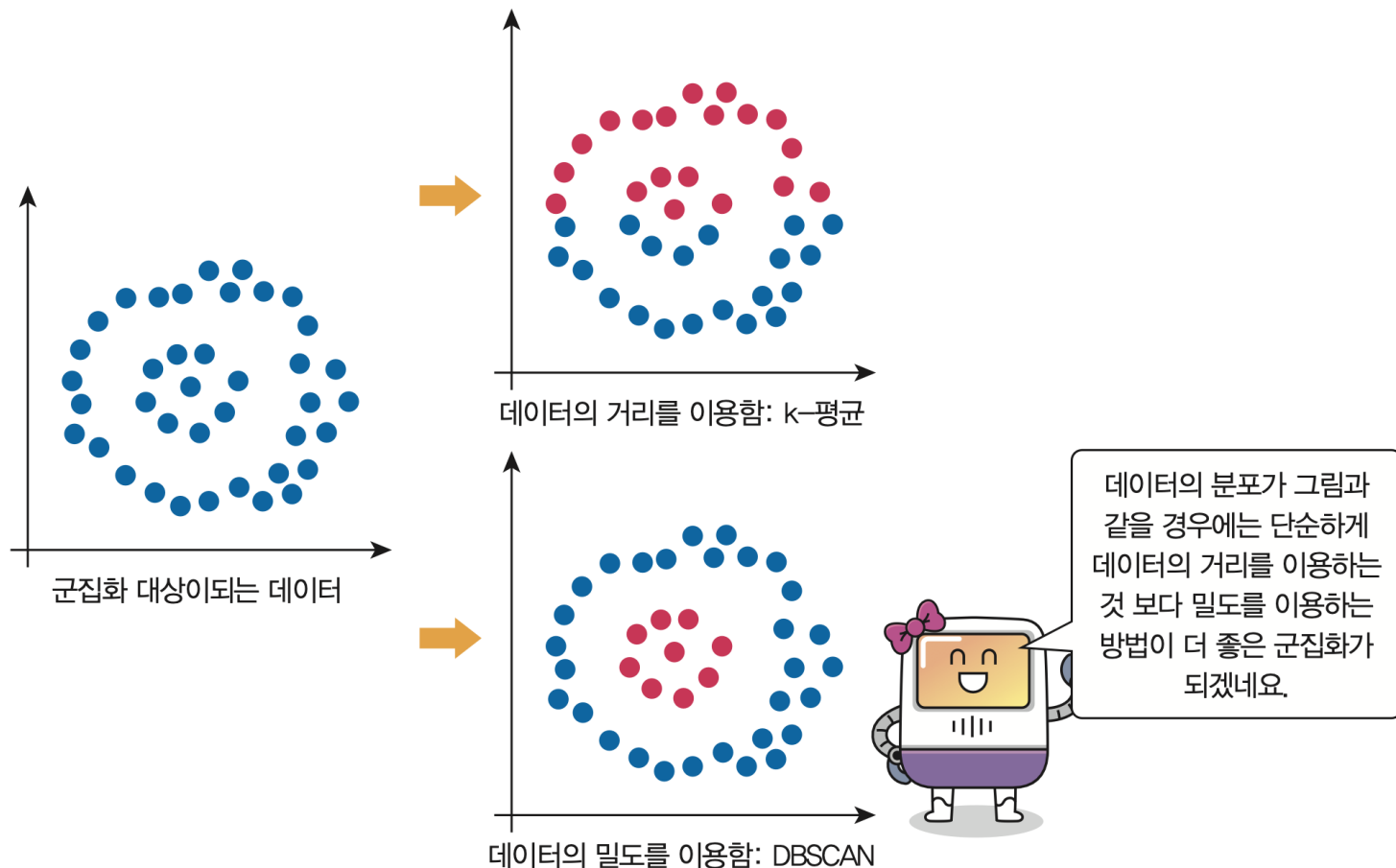
### ■ 밀도 기반 군집화

- **밀도 기반 군집화** Density-based spatial clustering of applications with noise:DBSCAN 은 군집화 중에서 비교적 고급 방법으로 밀도를 기반으로 군집을 나누는 유용한 기술이다.
- 앞서 설명한 k-평균 군집화의 경우 데이터 사이의 거리를 이용하여 군집화를 하는 방법인데, 단순히 거리를 이용하여 클러스터링을 할 경우 원형으로 이루어진 데이터에 대하여 비슷한 특징의 군집을 제대로 인식하지 못하는 문제점이 있다.



## 05 비지도 학습 방법을 알아보자

- 반면 밀도 기반의 군집화는 데이터가 몰려있는 곳의 밀도를 기반으로 군집화를 하는 방법이다.
- 따라서 가운데에 몰려있는 데이터들을 하나의 군집으로 묶고(빨간색 점들), 외부의 데이터를 또 다른 군집으로 묶어서(파란색 점들) 나눌 수 있다.





## LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

- 실습목표 : 이번 실습에서는 머신러닝의 최적화 기법에 대하여 알아볼 것이다. 이 실습에서는 간단한 수식이 등장하는데, 이 수식의 최소값을 컴퓨터가 어떻게 찾아 나가는가에 대하여 알아볼 것이다.
- 이번 장에서 다룬 바와 같이 경사 하강법은 대표적인 최적화 기법이다. 임의의 곡선이나 곡면이 있을 경우 이것의 극소값을 찾기 위해서는 어떻게 해야 할까? 곡선의 방정식을 풀어서 그 해를 구하는 방법도 있겠지만 이 방식은 컴퓨터에서는 적합하지 않다. 그 이유는 곡선이나 곡면의 함수들이 닫힌 형태가 아니거나 그 형태가 너무나 복잡해서 미분계수와 그 해를 구하는 것이 어려울 때가 많기 때문이다. 대신 컴퓨터가 잘하는 반복적인 작업 능력을 활용하는 것이 어떨까? 경사 하강법은 최적해를 얻기 위해서 반복적으로 파라미터를 조절하는 최적화 기법이다. 이 기법을 알아보기 위하여 다음의 웹사이트에 접속하도록 하자.

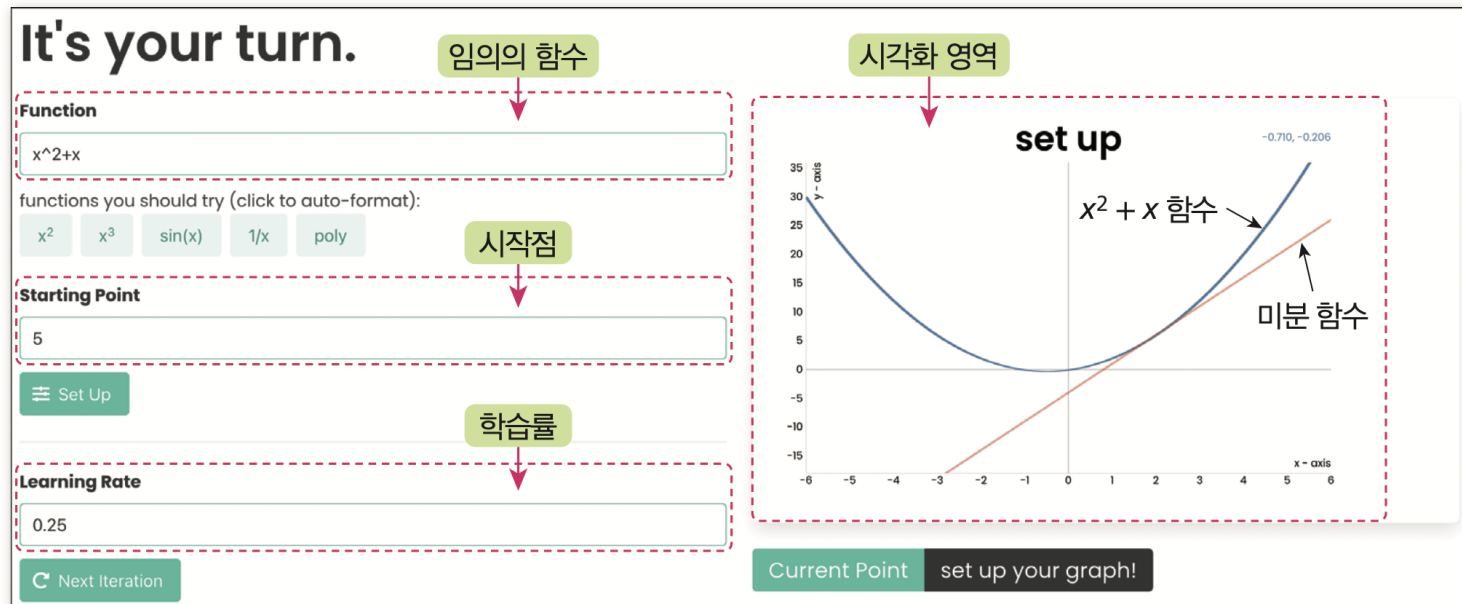
<https://uclaacm.github.io/gradient-descent-visualiser/#playground>

이 사이트는 경사 하강법의 원리를 시각적으로 보여주는 기능을 제공한다.

# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

## ■ 실습하기

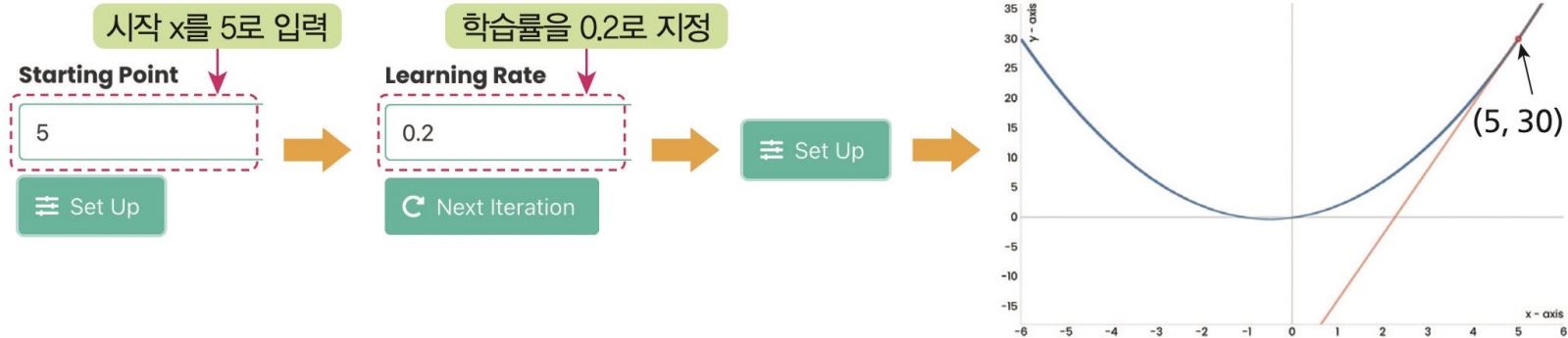
- 단계 1 : 위의 웹사이트에 접속한다. 다음과 같은 초기 화면이 나타날 것이다. 각 영역에 대한 설명이 그림에 나타나 있다.
- 화면의 왼쪽 위의 수식  $x^2 + x$ 는  $x^2 + x$  함수의 제곱 부분을 ^ 기호를 사용하여 표기한 것으로 시각화 영역에 있는 함수를 말한다. 다음으로 **시작점(Starting Point)**은 이 함수의 최소값을 얻기 위하여 처음으로 구한 임의의 해가 있는 위치를 말한다. 그리고 **학습률(Learning Rate)**은 해를 구하기 위한 이동 거리로 볼 수 있다.



# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

## ■ 실습하기

- 단계 2 : 시작점의 위치를 5로, 학습률을 0.2로 지정하도록 하자. 그리고 나서 Set up 버튼을 클릭하면 그림의 오른쪽과 같이 x의 시작 위치값이 5에 오게 되고  $y = x^2 + x$  함수의 출력에 해당하는 y값은 30이 된다( $y$ 값은  $5^2 + 5 = 25 + 5 = 30$ 으로 얻는다).



## LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

### ■ 실습하기

- 단계 3 : 다음으로 Next Iteration 버튼을 클릭해보자.

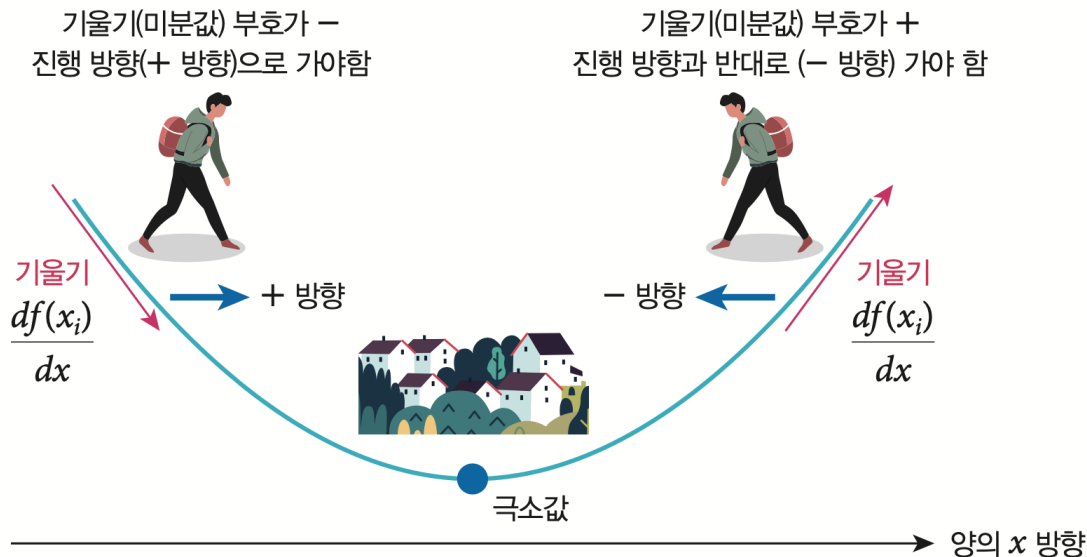
A green rectangular button with rounded corners, featuring a white circular arrow icon on the left and the text "Next Iteration" in white to its right.

↻ Next Iteration

# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

## ■ 실습하기

- 단계 3 : 그러면 경사 하강법을 통해서 새롭게 구한  $x$  값과  $y$  값을 보여준다.  $f(x)$  함수  $x^2 + x$ 의 1차 미분 함수  $f'(x)$ 는  $2x + 1$ 인데, **미분의 의미는 임의의 구간에서의 순간적인 변화량**이라는 것을 되새겨보도록 하자. 이제  $x = 5$ 일 때의 미분값을 구해보면  $2 \times 5 + 1 = 11$ 이 되는 것을 알 수 있다. 미분값의 부호가 양수가 된다는 것은 기울기가 양의 값이 된다는 의미이며, 이 곡선에서의 최소값은 5보다 작은 값이라는 의미이다. 예를 들어 깜깜한 산길을 걷고 있으며 길을 잃었다고 생각해보자. 우리는 한시라도 빨리 산 아래쪽에 있는 마을로 가야한다. 이때, 우리가 향하고 있는 산길이 오르막 방향(+)이라면 어떤 선택을 해야 할까? 아마도 진행 방향과 반대 방향(-방향)으로 가야만 할 것이다. 반대로 우리가 향하고 있는 산길이 내리막 방향(-)이라면 어떻게 해야만 할까? 아마도 진행 방향(+ 방향)으로 계속해서 가야만 할 것이다.



## LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

### ■ 실습하기

- 단계 4 : 다음 화면은 Next Iteration을 클릭한 결과이다. 새로운 위치값  $x_{i+1}$ 은 이전 단계의 위치값  $x_i$ 으로부터 구하는데 다음과 같은 간단한 식이 될 것이다. a의 앞에 - 부호가 붙는 이유는 그림 과 같이 미분의 부호가 +인 경우 진행 방향과 반대인 +로 가야하기 때문이다.

$$x_{i+1} = x_i - \alpha \frac{df(x_i)}{dx}$$

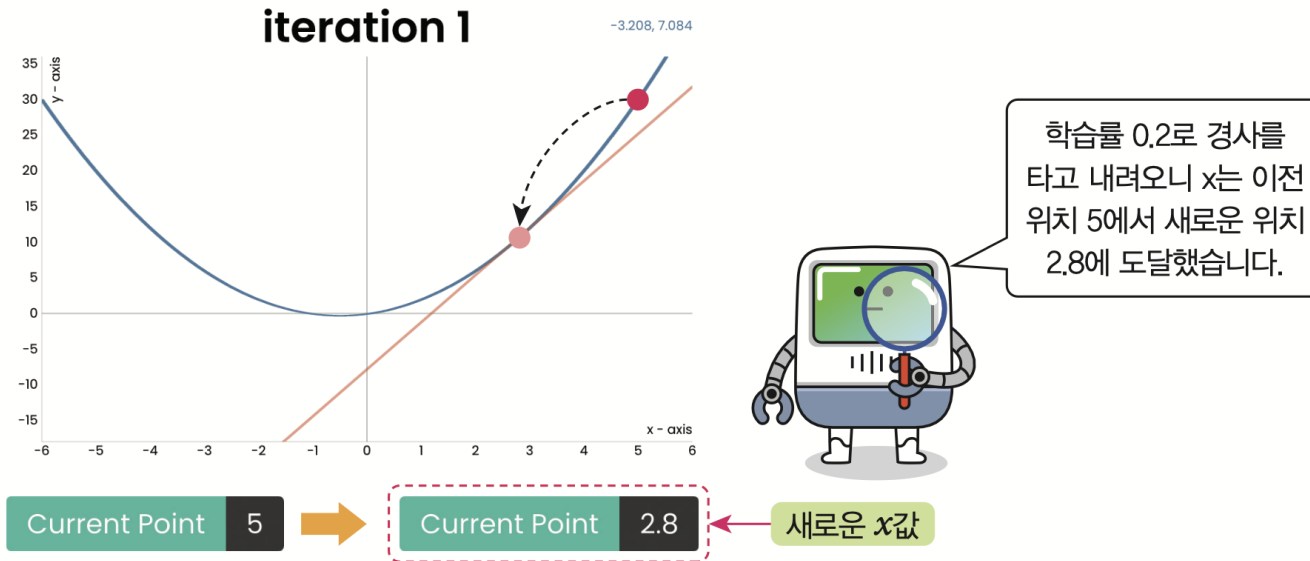
- a는 학습률로 얼마만큼 이동 거리를 조절할 것인가에 대한 파라미터이고, 이 예제에서는 0.2를 사용하였다. 따라서 새로운  $x_{i+1}$ 는 다음과 같이 구할 수 있다.

$$\begin{aligned} x_{i+1} &= x_i - \alpha \frac{df(x_i)}{dx} \\ &= 5 - 0.2 \times 11 \\ &= 2.8 \end{aligned}$$

# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

## ■ 실습하기

- 단계 4 : 위의 전개식에 의해서 경사를 타고 내려와서 새로운 위치에 도달했을 때의  $x$ 값은 5에서 2.8로 변경되고 다음 그림과 같이 표시된다.



# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

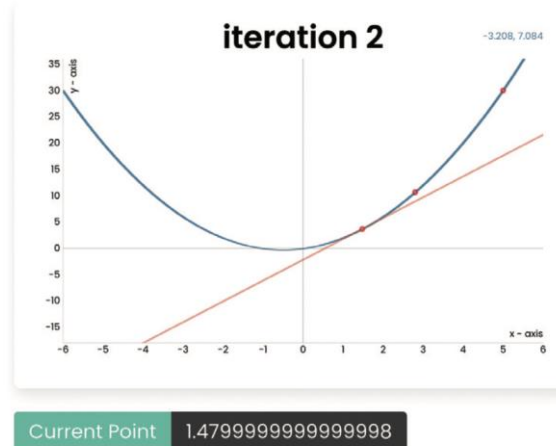
## ■ 실습하기

- 단계 5 :  $x_{i+1}$  이 이제는 2.8이므로  $f'(x)$ 는  $2x + 1$ 에 의해 6.6이 된다. 따라서 다시 한번 Next Iteration으로  $x_{i+2}$ 를 구해보자.

다소 지루하기는 하지만  $x_{i+2}$ 을 다음 식으로 전개해서 풀어보면 1.48이 된다.

$$\begin{aligned}x_{i+2} &= x_{i+1} - \alpha \frac{df(x_{i+1})}{dx} \\&= 2.8 - 0.2 \times 6.6 \\&= 1.48\end{aligned}$$

- 이 결과는 아래와 같이 시각적으로 표시된다. Current Point가 1.4799999999998로 1.48의 근사값이 나오는데 이는 실수 계산에서 발생하는 오차값이다.

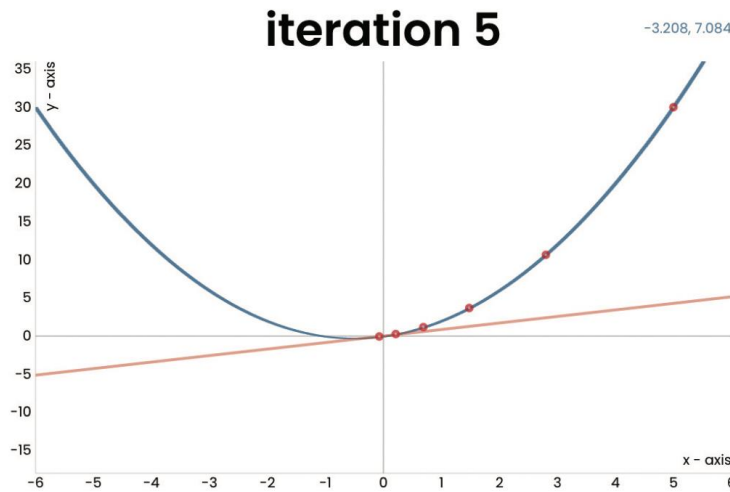




# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

## ■ 실습하기

- 단계 6 : 앞의 과정을 반복하면 다음 그림과 같이 iteration 5단계에서는 최소값에 매우 가깝게 도달한 것을 확인할 수 있다.



- 톰 미첼의 머신러닝 정의를 다시 한번 생각해 보면 이 프로그램은 작업  $\tau$ 와 성능 척도  $p$ 에 대해 경험  $E$ 로부터 학습을 하여 최적값에 도달하므로 경사 하강법은 머신러닝 기법의 일종으로 간주할 수 있다.

# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

## ■ 실습하기

- 손으로 계산하기 : 위의 최적화 과정은 다음과 같은 표를 만들어서 직접 손으로 계산해볼 수 있다. 여러분은 이 수치값을 얻기 위해 계산기를 사용하면 된다. 단계 4까지의 계산을 손으로 직접 적어보도록 하자. 물론 이러한 과정은 컴퓨터 프로그램을 통해서 간단하게 구현할 수도 있다.

단계( <i>i</i> )	$x$	$f(x) = x^2 + x$	$\frac{df(x)}{dx} = 2x + 1$	$\alpha$	$x_{next} = x - \alpha \frac{df(x)}{dx}$
1	5	30	11	0.2	$5 - (0.2 * 11) = 2.8$
2	2.8				
3					
4					

# LAB 08 : 머신러닝 최적화에 숨어있는 수학적 원리를 알아보자

## ■ 실습하기

### 도전 문제



- 이 웹사이트에서 시작 위치를  $-5$ 로, 학습률( $\alpha$ )을  $0.2$ 로 두고 위의 과정을 반복해 보자. 시작 위치가  $-5$ 일 경우 미분값의 부호는 무엇인가?
- 학습을 5번 반복할 때의 과정을 다음의 표를 채우면서 살펴보자.

단계( $i$ )	$x$	$f(x) = x^2 + x$	$\frac{df(x)}{dx} = 2x + 1$	$\alpha$	$x_{next} = x - \alpha \frac{df(x)}{dx}$
1	$-5$	20	$-9$	0.2	$-5 - (0.2 \times -9) = -3.2$
2	$-3.2$				
3					
4					
5					

1. 우리가 다루는 **인공지능** 기술은 인간의 학습능력, 추론능력, 지각능력을 인공적으로 구현하려는 컴퓨터 과학의 세부적인 분야로 정의할 수 있는데, 좀 더 구체적으로 말하면 **컴퓨터를 사용하여 인간의 지능을 구현하려는 기술**이라고 할 수 있다.
2. **머신러닝**은 명시적 프로그래밍 없이 컴퓨터가 **학습을 통해 작업 성능을 높여나가는 기술**을 말한다.
3. 머신러닝은 **데이터를 기반으로 학습하여 작업 성능을 개선하는 알고리즘**으로 여러 장점을 가진다.
4. 두 개의 **변수들이 함께 변화하는 관계**가 **상관관계**이다. 그리고 이 변수들 사이의 상관관계의 정도를 나타내는 수치가 **상관계수**이다.
5. 선형 회귀는 실제 데이터를 바탕으로 새로운 데이터가 입력되었을 경우의 **출력 결과를 예측**할 수 있다는 측면에서 의의가 크다고 할 수 있다.
6. 선형 회귀에서 **오차의 합**을 구하고 그것을 데이터의 개수로 나누어 평균을 구하여 얻은 오차값을 **평균 절대값 오차**라고 한다. 또한 오차 제곱 값의 평균으로 오차를 측정하는 방식으로 얻은 오차값을 **평균 제곱 오차**라고 한다.

7. 데이터의 분포를 가장 잘 설명하는 선형 회귀 직선을 구하는 과정을 **최적화**라고 한다. 그리고 이 최적화 과정에 사용되는 대표적인 기법이 **경사 하강법**이다.
8. 머신러닝의 **지도 학습**에서 컴퓨터는 “교사”에 의해 데이터와 **레이블**을 제공받는다.
9. 머신러닝의 **비지도 학습**은 지도 학습과는 달리 외부에서 레이블을 주지 않고 학습 알고리즘 이 스스로 입력으로부터 특정한 구조를 발견하는 학습이다. 비지도 학습의 대표적인 예가 **군집화**이다.
10. 머신러닝의 **강화 학습**은 **에이전트, 환경, 행동, 보상**과 **상태**라는 학습 데이터가 주어진다.
11. 지도 학습에서 주어진 데이터를 바탕으로 예측하고자 하는 것이 무엇인지에 따라서 크게 **회귀**와 **분류**로 나눈다. 이 중에서 예측하고 싶은 **종속 변수가 숫자일 때 회귀**라는 머신러닝 기법을 사용한다. 반면 **범주형일 경우 분류 기법**을 사용한다.
12. 분류 문제에서 클래스가 두 가지인 경우를 **이진 분류**라고 하며, 둘 이상이 될 경우를 **다중 분류**라고 한다.
13. 지도 학습 방법의 대표적인 예로 **k-최근접 이웃 알고리즘, 서포트 벡터 머신**이 있으며 비지도 학습 방법으로 **k-평균 군집화, 밀도 기반 군집화**가 있다.