

# 머신 러닝 기본기

• 기계가 학습한다는 건, 프로그램이 특정 작업(T)을 하는 데 있어서 경험(E)을 통해 작업 의 성능(P)을 향상시키는 것

## • 빅데이터

- 。 엄청나게 많은 양의 데이터를 다루는 분야
- 。 데이터 보관/처리법
- 。 데이터 분석 방법들

#### • 인공 지능

。 프로그램이 인간처럼 생각/행동하게 하는 학문

#### • 딥 러닝

ㅇ 머신 러닝 기법 중 하나

#### • 머신 러닝

- 。 지도 학습
  - "답"이 있고 이 답을 맞추는 게 학습의 목적
  - 분류 vs 회귀
  - 학습 데이터의 답을 정해줘야 함.
- 。 비지도 학습
  - "답"이 없고 이 답을 맞추는 게 학습의 목적

#### • k-최근접 알고리즘

- k-Nearest Neighbors Algorithm
- 많은 경험 → 성능 향상 → 머신 러닝
- Python numpy 문법
  - 。 선형대수학

머신 러닝 기본기 1

```
import numpy as np
# numpy 에서 행렬 선언
A = np.array([
   [1,-1,2],
   [3,2,2],
   [4,1,2],
   [7, 5, 6]
])
Α
array([[ 1, -1, 2],
      [ 3, 2, 2],
      [ 4, 1, 2],
      [7, 5, 6]])
# 랜덤값을 가지는 행렬
C = np.random.rand(3,5)
array([[0.36914614, 0.50373658, 0.56483108, 0.26833052, 0.51586141],
       \hbox{\tt [0.02823758, 0.78994553, 0.68682965, 0.4076506, 0.98156854],}
       [0.12719764, 0.27983164, 0.03522864, 0.22322727, 0.74053391]])
# 영행렬 선언
D = np.zeros((2,4))
array([[0., 0., 0., 0.],
      [0., 0., 0., 0.]])
# 행렬에서 특정 위치 접근
A[0][2] #1행 3열
# 행렬 덧셈
A+B
# 행렬 곱셈(단, 원소끼리의 곱셈)
A*B
# 행렬 곱(일반적인 곱셈)
np.dot(A,B)
A @ B
# Transpose
A.T
# 단위 행렬 선언 (3*3)
I = np.identity(3)
# 역행렬 (행렬 A)
np.linalg.pinv(A)
```

머신 러닝 기본기 2

### 。 미분

- 함수의 순간 변화율을 구하는 방법
- 편미분
  - 함수를 변수 하나에 대해서만 미분
- 고차원에서 미분
  - :  $f(x,y) = x^2+2y^2$
  - → 편미분 진행

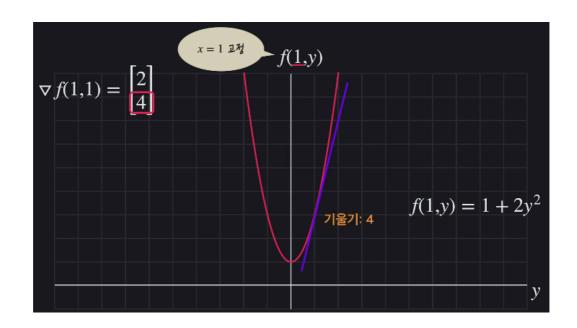
$$\frac{\partial}{\partial x} f(x, y) = 2x$$

$$\Rightarrow \qquad \nabla f(x, y) = \begin{bmatrix} 2x \\ 4y \end{bmatrix}$$

$$\frac{\partial}{\partial y} f(x, y) = 4y$$

$$\nabla f(1, 1) = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$abla f(1,1) = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$
 $f(x,y) = x^2 + 2y^2$ 
 $\downarrow x = 1$  의 교체
 $f(x,y) = x^2 + 2y^2$ 
 $\downarrow x = 1$  교체
 $f(1,y) = 1 + 2y^2$ 



머신 러닝 기본기 4