

# 인공지능

## 02. 머신 러닝

Dept. of Digital Contents



# 1. 머신러닝이란?

## ○ 기계학습(machine learning)

- 정의 : 데이터로 학습을 수행해 인식 성능을 최대화하는 접근 방법
- 1959년 아서 사무엘(Arthur Samuel)
  - IBM 701에서 최초의 체커 프로그램(게임) 제작
  - 탐색 기법을 이용하여 컴퓨터가 문제를 해결할 수 있도록 함
- 전통적인 프로그래밍과 머신러닝의 차이점



# 1. 머신러닝이란?

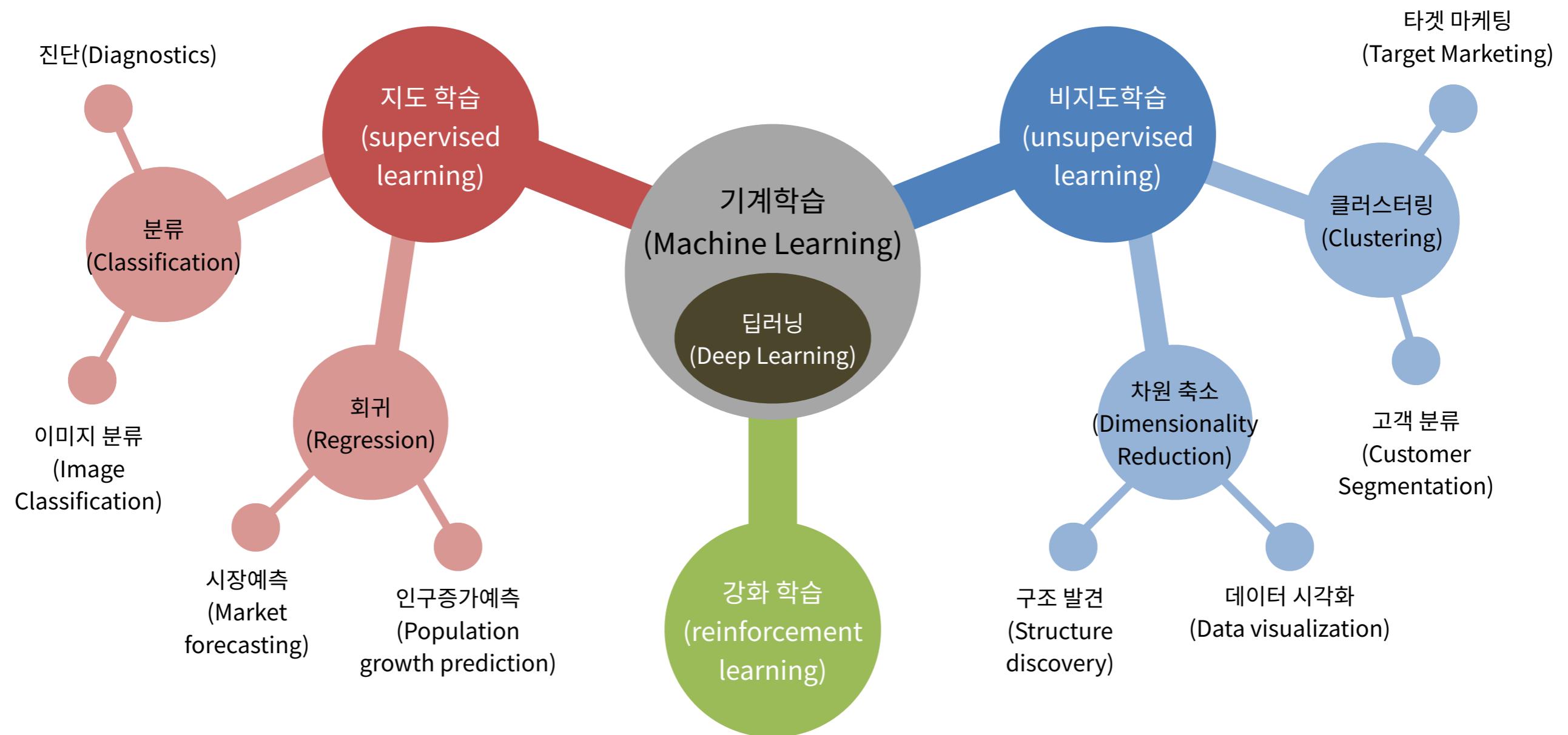
---

## ○ 머신러닝의 종류

- 지도 학습(Supervised Learning)
  - 주어진 데이터(샘플)와 정답(레이블)을 제공 받음
- 비지도 학습(Unsupervised Learning)
  - 정답(레이블)이 주어지지 않고 스스로 패턴을 발견하는 학습
- 강화 학습(Reinforcement Learning)
  - 보상과 처벌의 형태로 학습 데이터가 주어짐

# 1. 머신러닝이란?

## ○ 머신러닝의 종류



# 1. 머신러닝이란?

---

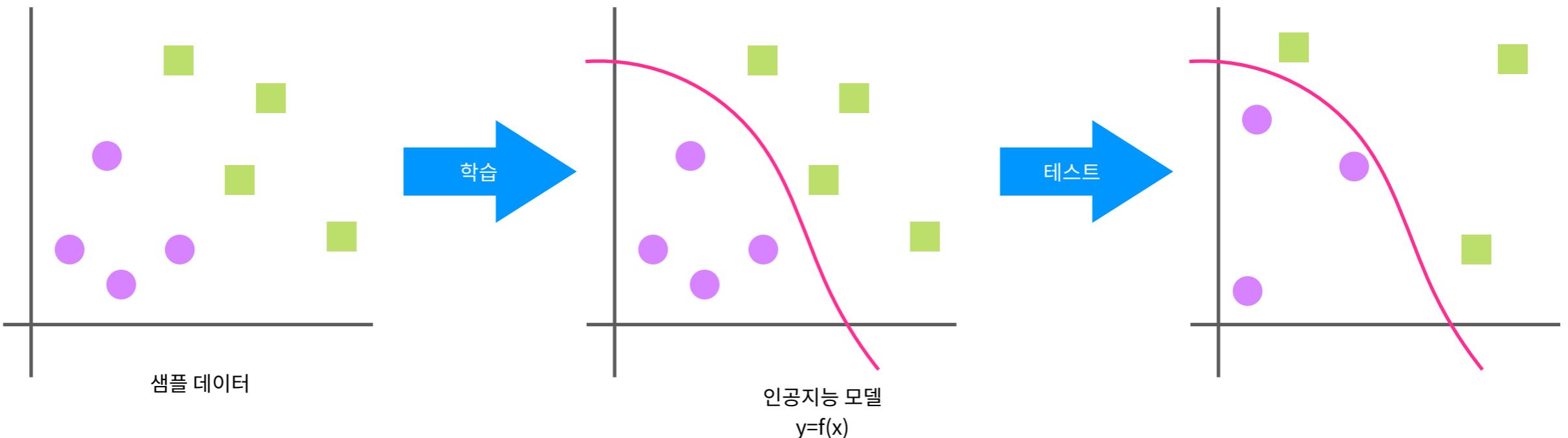
## ○ 머신러닝의 용어들

- 특징(Feature)
  - 학습 모델에게 공급하는 입력
  - 예) 사과 등급 분류 : 색상, 크기, 모양 등
- 함수 근사와 머신러닝
  - 머신 러닝 = 입력(X)를 출력(Y)에 매핑시키는 함수  $y=f(x)$ 를 찾아내는 것
- 레이블(Label)
  - 머신러닝으로 예측하는 결과(출력값)
- 샘플(Sample)
  - 머신러닝으로 학습하기 위해 주어지는 입력 예제(지도학습에서 정답 레이블이 존재)
  - 예측(prediction) : 학습된 모델이 입력 예제가 아닌 데이터 입력으로 결과를 만들어 내는 것

# 1. 머신러닝이란?

## ○ 머신러닝의 용어들

- 학습 데이터와 테스트 데이터

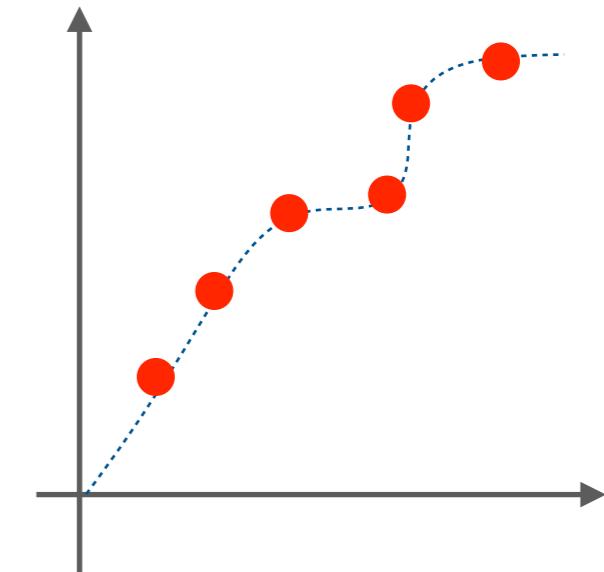
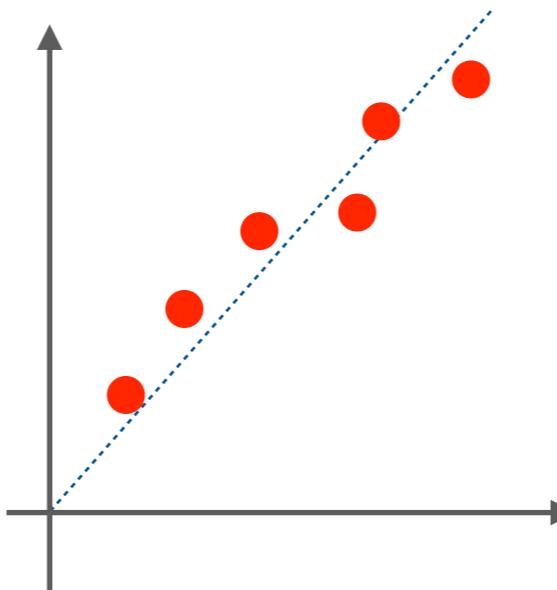
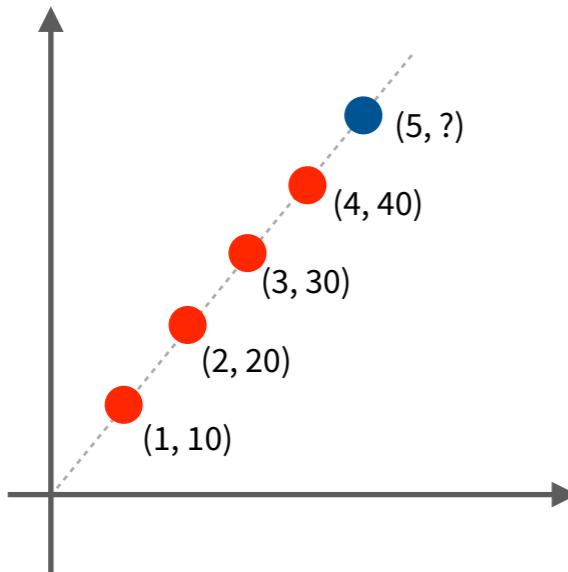


## 2. 지도학습

### ○ 회귀와 분류

#### ■ 회귀(Regression)

- 주어진 입력-출력 쌍을 학습한 후 새로운 입력값이 들어왔을 때 합리적인 출력값을 예측
- $y = f(x)$ 의 매팅 함수를 찾음
- 함수의 종류에 따라 선형(linear) 회귀와 비선형(polynomial) 회귀가 있음

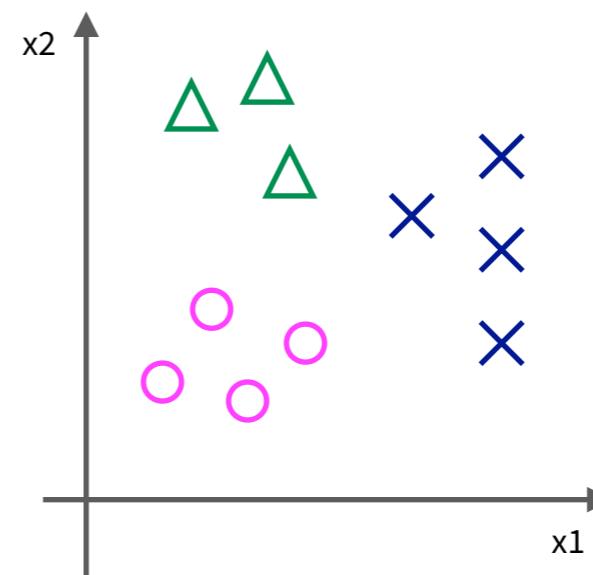
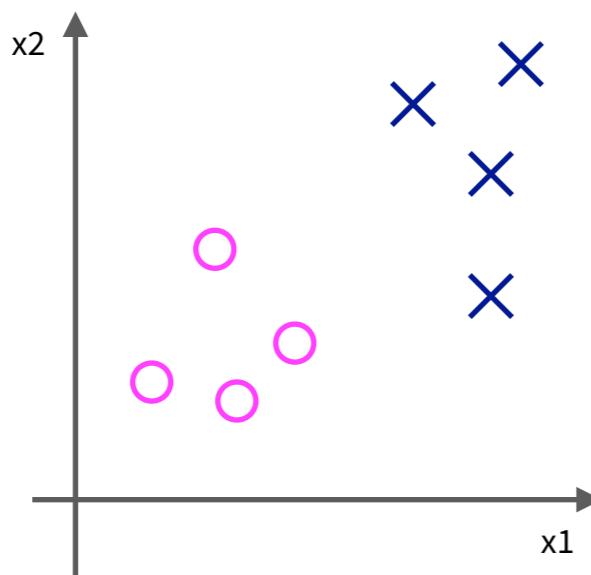


## 2. 지도학습

### ○ 회귀와 분류

#### ■ 분류(Classification)

- 주어진 입력 데이터를 2개 이상의 클래스(부류)로 나누는 것
- 예) 사진을 입력하면 고양이 또는 강아지로 분류하여 각각의 폴더에 저장
- 분류 클래스 수에 따라 이진(binary) 분류와 멀티클래스(Multi-Class) 분류가 있음

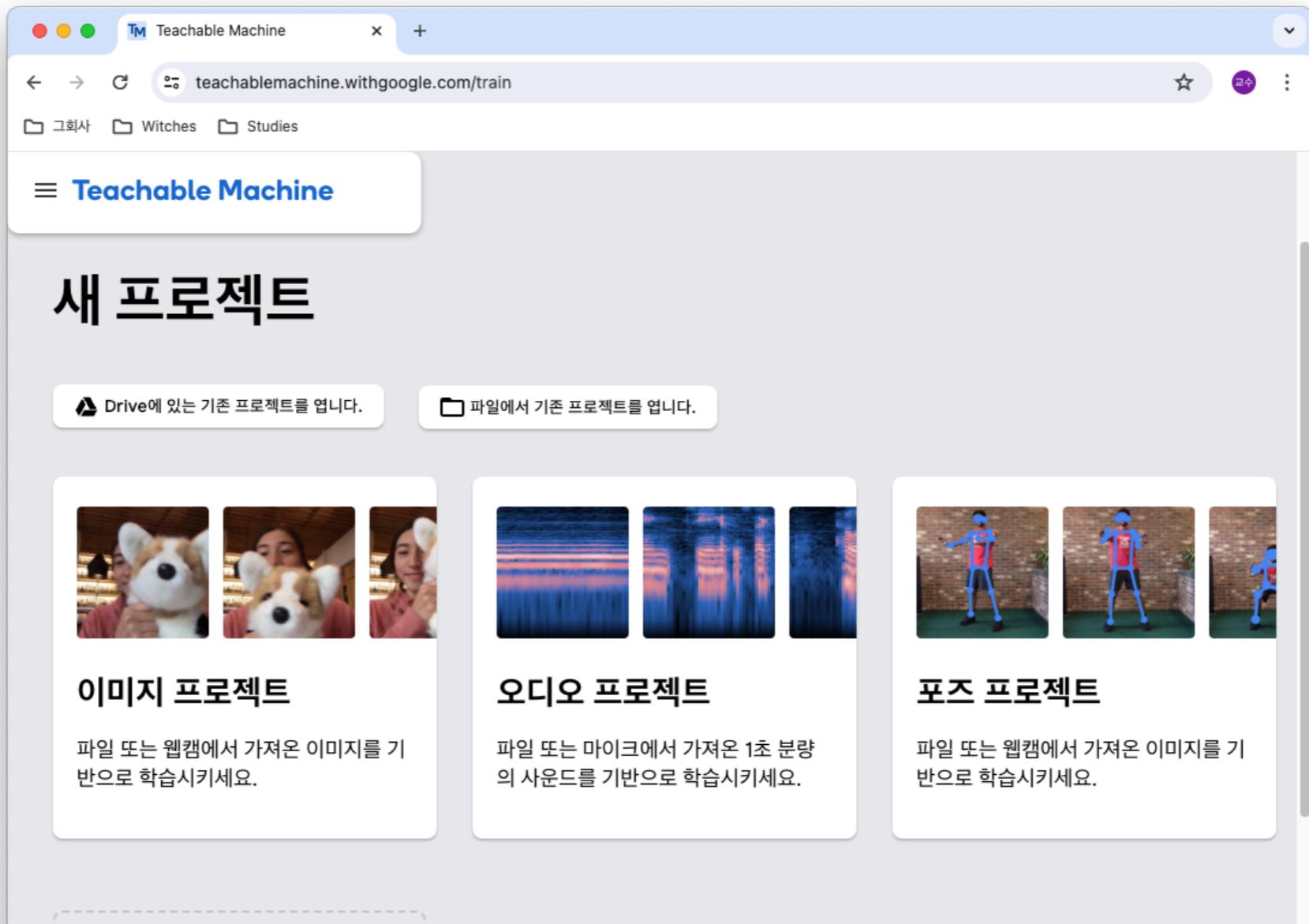


## 2. 지도학습

### ○ 회귀와 분류

#### ■ 티처블머신([teachablemachine.withgoogle.com](https://teachablemachine.withgoogle.com))

- 사자와 호랑이 분류 학습

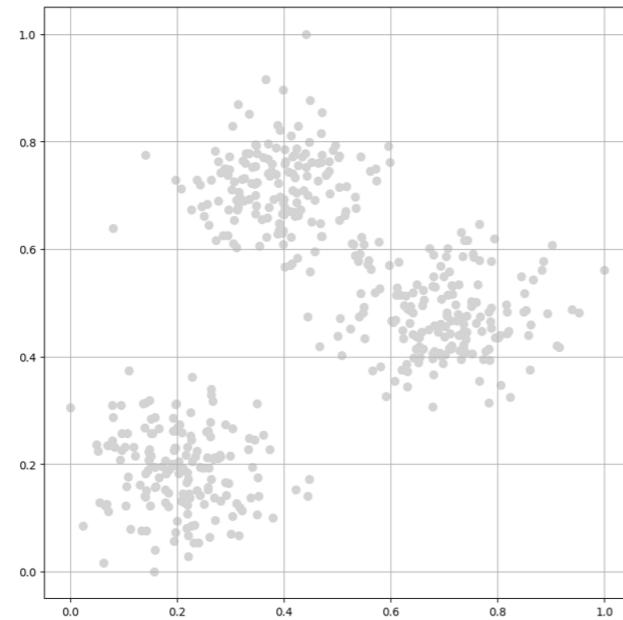


# 3. 비지도학습

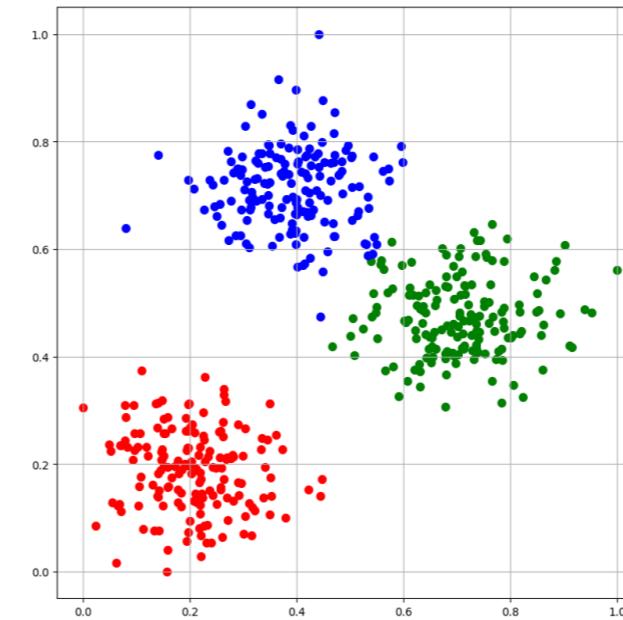
## ○ 비지도 학습

- 컴퓨터가 스스로 입력들을 분류
  - $y = f(x)$ 의 레이블  $y$ 가 주어지지 않음
  - 데이터 상관도를 분석하여 유사 데이터를 모음

K-means 클러스터링



Data



Clustering

## 4. 강화학습

---

### ○ 강화 학습

- 인간은 처한 환경의 상태를 보고 유리한 행동을 결정하고 실행
  - “행동 → 상태 변화 → 보상”의 사이클 속에서 학습
  - 컴퓨터가 행동을 취할 때마다 외부에서 처벌이나 보상이 주어짐
- 바둑, 게임, 자율주행 등에 활용
  - 알파고 : 2016년 3월 딥마인드 챌린지 매치(with 이세돌)
  - 알파스타 : 2019.10 Starcraft II 상위 0.2% 수준
  - 로봇, 자율주행차 제어등에 널리 사용
- 자연 영상 처리 등 특정 분야에서는 성능을 발휘하지 못함
  - 보상 지정을 자율로 할 수 없음

# 5. 머신러닝의 과정

## ○ 머신러닝 과정

- 사과 등급을 분류하는 인공지능 시스템 사례
  - 목표 : 사과를 상/중/하로 분류하는 인공지능 기계

### 1. 데이터의 특징 결정

- 예) 사과의 지름(크기), 익은 정도(색깔), 표면의 균일도
- 특징 벡터와 레이블 준비
  - 특징 벡터 : [지름 크기(cm), 붉은색의 균일성(1~10 척도), 껍질의 윤기(1~10)]
  - 레이블 : 상 = 1, 중 = 2, 하 = 3

### 2. 데이터 수집

- 상/중/하 비율이 비슷하도록 수천개의 사과를 수집
- 여러 농장에서 골고루 수집 : 데이터의 다양성 확보 필요(데이터 편향을 피하기) → 인식률 향상
- 학습 데이터 정제하기 : 중복 제거, 정규화, 오류 수정 등



# 5. 머신러닝의 과정

## ○ 머신러닝 과정

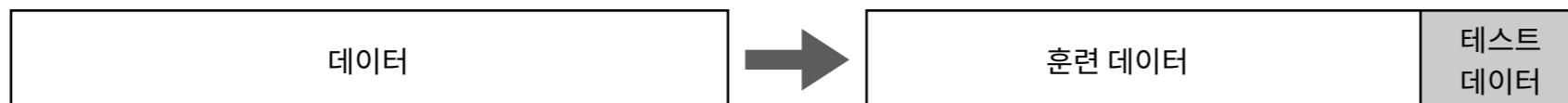
### ■ 사과 등급을 분류하는 인공지능 시스템 사례

#### 2. 데이터 수집

##### ○ 훈련 데이터(학습 데이터) 특징 추출

| 데이터 연번 | 특징 #1 | 특징 #2 | 특징 #3 | 레이블 |
|--------|-------|-------|-------|-----|
| 1      | 7.9   | 8     | 7     | 2   |
| 2      | 8.0   | 9     | 8     | 1   |
| 3      | 6.5   | 6     | 4     | 3   |
| ...    | ...   | ...   | ...   | ... |

##### ○ 테스트 데이터 : 인공지능 모델 평가용



# 5. 머신러닝의 과정

---

## ○ 머신러닝 과정

### ■ 사과 등급을 분류하는 인공지능 시스템 사례

#### 3. 모델 선택(model)

- 훈련 데이터의 종류와 학습 목표에 따라 다양한 모델이 연구되어 있음
  - 주어진 데이터에 가장 알맞는 모델은 선택(SVM, k-NN, 결정 트리, 신경망 등)

#### 4. 학습(fit)

- 훈련 데이터가 인공 지능 모델에 의해 만들어진 결과가 레이블과 일치하도록 학습
  - 오류가 적은 방향으로 인공지능 모델을 업데이트(모델을 구성하는 가중치값(weight)를 변경)

#### 5. 평가(evaluate)

- 학습이 완료된 모델의 평가
  - 테스트 데이터를 이용하여 성능 평가

# 5. 머신러닝의 과정

---

## ○ 머신러닝 과정

### ■ 사과 등급을 분류하는 인공지능 시스템 사례

#### 6. 예측(predict)

○ 완성된 모델에 질문(데이터)을 하면 결과를 제출하는 활용 단계

- 주어진 데이터에 가장 알맞는 모델은 선택

# 5. 머신러닝의 과정

## ○ 머신러닝 과정

### ■ SVM을 이용한 필기 숫자 인식

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** Handwrite\_Digits.ipynb
- Toolbar:** 파일, 수정, 보기, 삽입, 런타임, 도구, 도움말, 모든 변경사항이 저장됨
- Code Cells:**
  - Cell 1: from sklearn import datasets  
import matplotlib.pyplot as plt
  - Cell 2: # 데이터 읽기  
digit = datasets.load\_digits()
  - Cell 3: # 데이터 확인  
plt.figure(figsize=(5,5))  
plt.imshow(digit.images[0], cmap=plt.cm.gray\_r, interpolation='nearest')  
plt.show()
  - Cell 4: print(digit.data[0])  
print('이 숫자는 ', digit.target[0], '입니다.')

# 5. 머신러닝의 과정

## ○ 머신러닝 과정

- SVM을 이용한 필기 숫자 인식

```
0초   ▶ from sklearn import svm

# 모델 선택 : SVM(Support Vector Machine) 사용(분류 모델인 SVC)
s = svm.SVC(gamma=0.1, C=10)
# 하이퍼 파라미터 : gamma 값이 클수록 근접데이터 가중치를 높임, C값이 클수록 제약조건 완화

# 모델 학습
s.fit(digit.data, digit.target)

# 예측
new_d = [digit.data[0], digit.data[1], digit.data[2]]
res = s.predict(new_d)
print('예측값은 ', res)
print('참값은 ', digit.target[0], digit.target[1], digit.target[2])
```

# 6. 머신러닝의 알고리즘 평가

## ○ 성능 평가 기준

### ■ 혼동 행렬(confusion matrix)

- 부류(클래스)별로 옳은 분류와 틀린 분류의 개수를 기록한 행렬

|           |     | 참값(그라운드 트루스) |    |     |    |     |    |
|-----------|-----|--------------|----|-----|----|-----|----|
|           |     | 0            | 1  | ... | 5  | ... | 9  |
| 예측한<br>부류 | 0   | 69           | 2  |     | 1  |     | 0  |
|           | 1   | 0            | 69 |     | 0  |     | 2  |
|           | ... |              |    |     |    |     |    |
|           | 5   | 0            | 0  |     | 65 |     | 1  |
|           | ... |              |    |     |    |     |    |
|           | 6   | 0            | 0  |     | 1  |     | 66 |
|           |     |              |    |     |    |     |    |

필기 숫자의 경우

|     |    | 그라운드 트루스 |    |
|-----|----|----------|----|
|     |    | 긍정       | 부정 |
| 예측값 | 긍정 | TP       | FP |
|     | 부정 | FN       | TN |

부류가 2개인 경우

TP : True Positive(참 긍정)

FN : False Negative(거짓 부정)

FP : False Positive(거짓 긍정)

TN : True Negative(참 부정)

# 6. 머신러닝의 알고리즘 평가

## ○ 성능 평가 기준

### ■ 정확도(Accuracy)

- 대표적인 성능 측정 기준

$$\text{정확도} = \frac{\text{맞힌 샘플 수}}{\text{전체 샘플 수}} = \frac{\text{대각선 샘플 수}}{\text{전체 샘플 수}}$$

- 2부류 문제(긍부정)에서 한쪽 샘플 자체가 적은 경우 한계

○ 예) 암환자 진단 : 200명 중 1명 꼴의 암환자 → 의사의 200명 진단 소견 모두 정상 = 정확도 99.5%

- 특이도 : 정상을 올바르게 정상이라고 진단하는 비율

- 민감도 : 질병이 있는 사람을 환자라고 올바르게 진단하는 비율

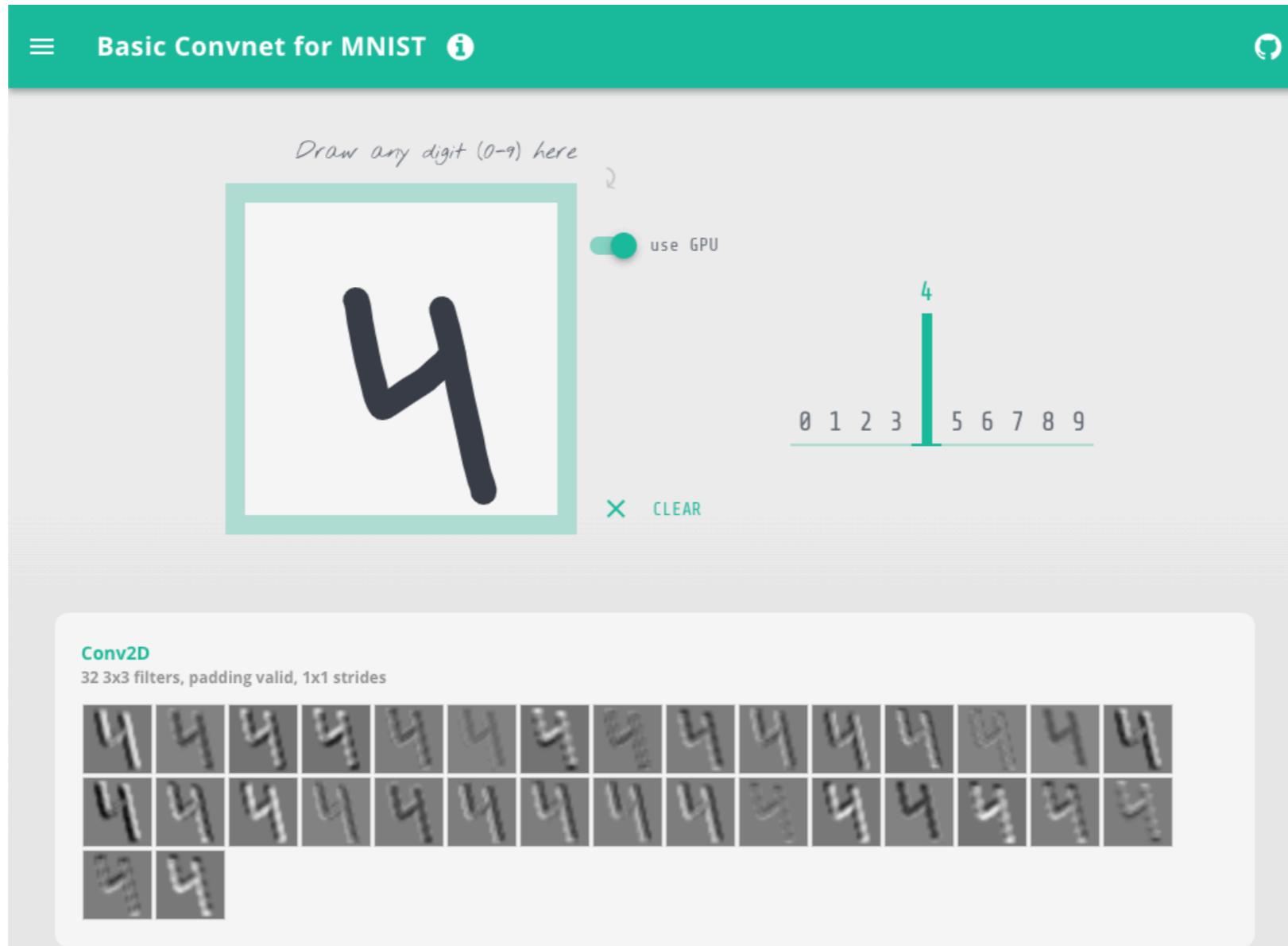
$$\text{특이도} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{민감도} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

# 6. 머신러닝의 알고리즘 평가

## ○ 머신 러닝 체험

- <https://transcranial.github.io/keras-js/#/>



# 7. 머신러닝의 용도

---

## ○ 이용 분야

### ■ 빅데이터 부상과 함께 머신 러닝의 중요성 증가

- 영상인식, 음성 인식 : 규칙과 공식이 복잡하여 프로그램하기 어려운 분야
- 시스템 침입 탐지, 신용카드 거래 사기 등 작업 규칙이 지속적으로 바뀌는 상황
- 주식거래, 에너지 수요 예측, 쇼핑 추세 예측 처럼 데이터 특징이 계속 바뀌는 상황
- 구매자가 클릭할 확률이 가장 높은 광고를 알아내는 시스템(구글 Ads)
- 넷플릭스 비디오 추천 시스템, 아마존 상품추천 시스템
- 이미지 인식 시스템(문자 인식)
- 자율주행 자동차
- 텍스트 자동 인식