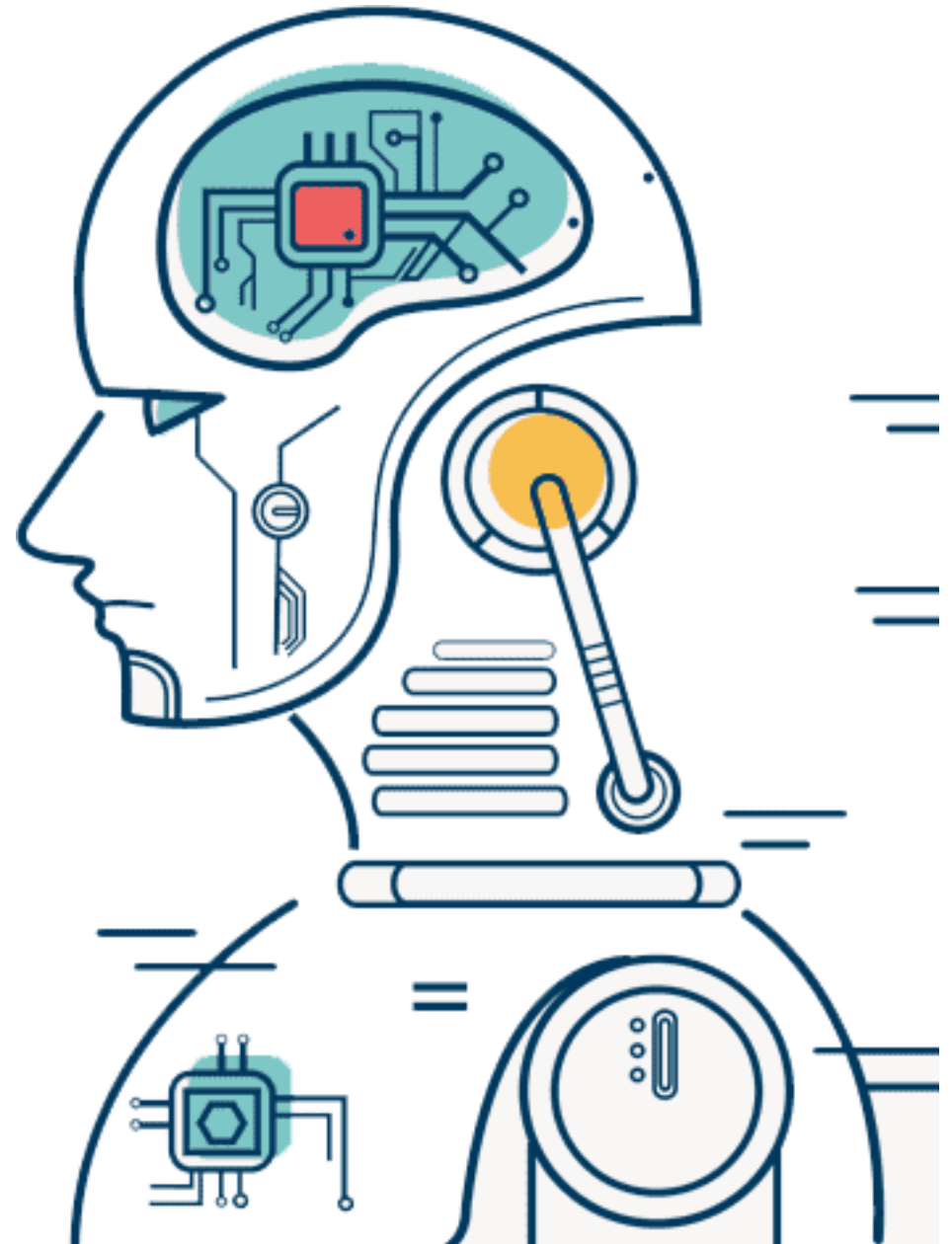


# Machine Learning

## Chapter 2 지도학습

(일반화, 과대적합, 과소적합, KNN, Iris 실습)



- 일반화, 과대적합, 과소적합을 이해 할 수 있다.
- KNN 알고리즘을 이해 할 수 있다.
- 하이퍼파라미터 튜닝을 할 수 있다.
- KNN 회귀에 대해 이해할 수 있다



아이에게 공이 무엇인지 알려주자

## 공이라는 것은..

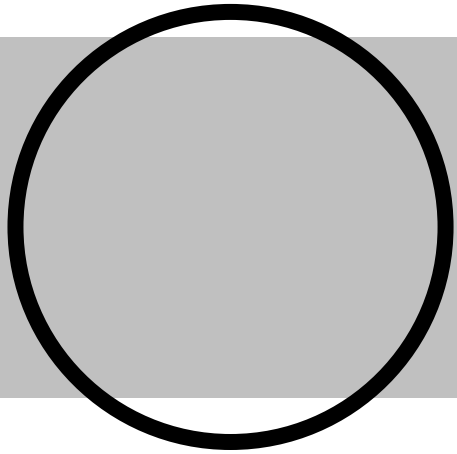


둥글게 생겼다.  
오각형이 여러 개 붙어있다.  
검은색과 흰색으로 구성된다.  
반짝반짝 광이 난다.

# 과대적합

공이라는 것은..

둥글게 생겼다.



**과소적합**

## 일반화 (Generalization)

- 훈련 세트로 학습한 모델이 테스트 세트에 대해 정확히 예측하도록 하는 것.

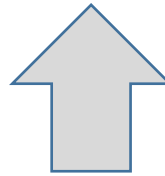
## 과대적합 (Overfitting)

- 훈련 세트에 너무 맞추어져 있어 테스트 세트의 성능 저하.

## 과소적합 (Underfitting)

- 훈련 세트를 충분히 반영하지 못해 훈련 세트, 테스트 세트에서 모두 성능이 저하.

일반화 성능이 최대화 되는 모델을 찾는 것이 목표



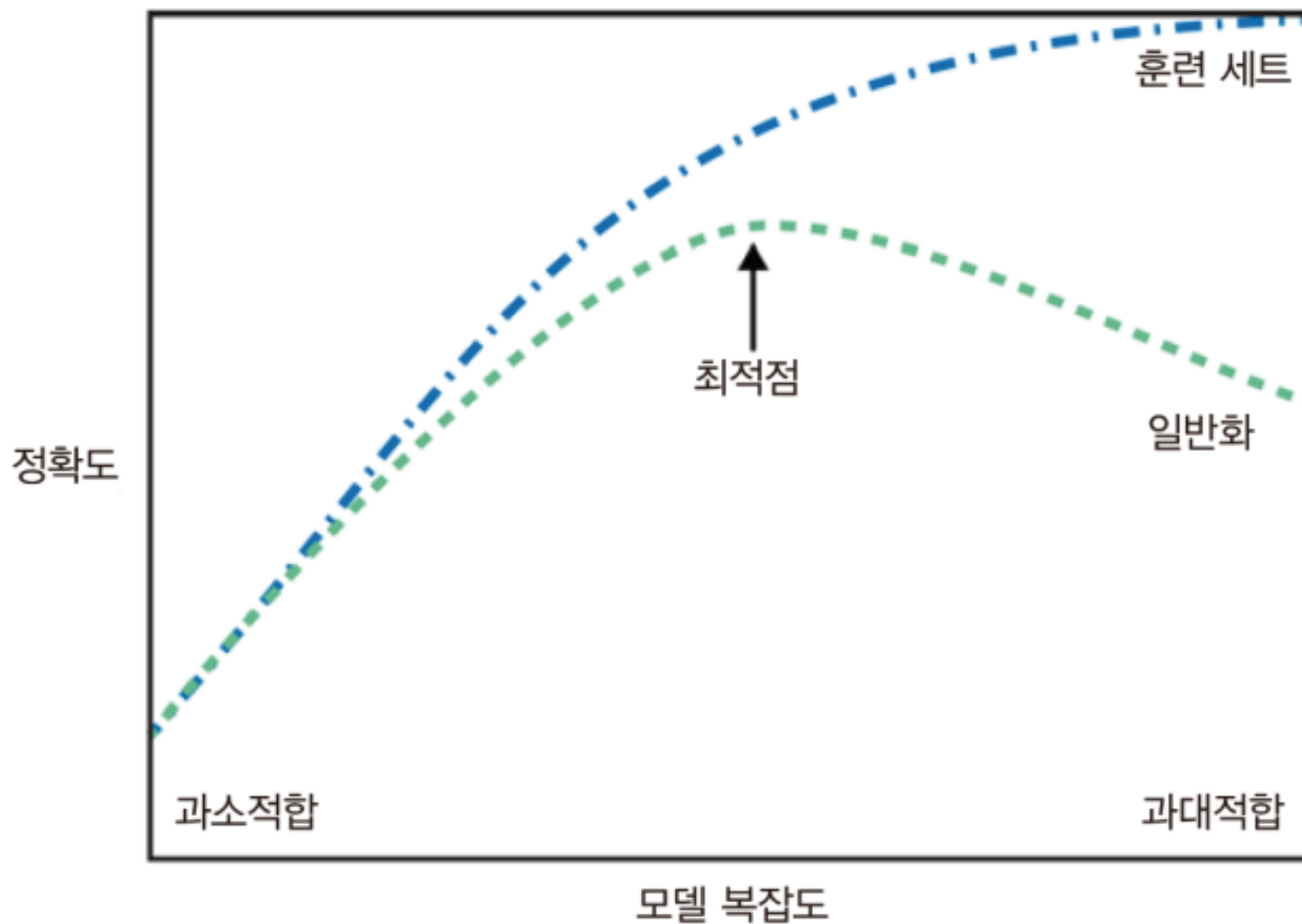
## 과대적합 (Overfitting)

- 너무 상세하고 복잡한 모델링을 하여 훈련데이터에만 과도하게 정확히 동작하는 모델.

## 과소적합 (Underfitting)

- 모델링을 너무 간단하게 하여 성능이 제대로 나오지 않는 모델.

## 모델 복잡도 곡선





## 해결방법

- 주어진 훈련데이터의 다양성 보장 → 다양한 데이터포인트를 골고루 나타내야 한다.
- 일반적으로 데이터 양이 많으면 일반화에 도움이 된다.
- 하지만 편중된 데이터를 많이 모으는 것은 도움이 되지 않는다.
- 규제(Regularization)을 통해 모델의 복잡도를 적정선으로 설정한다.

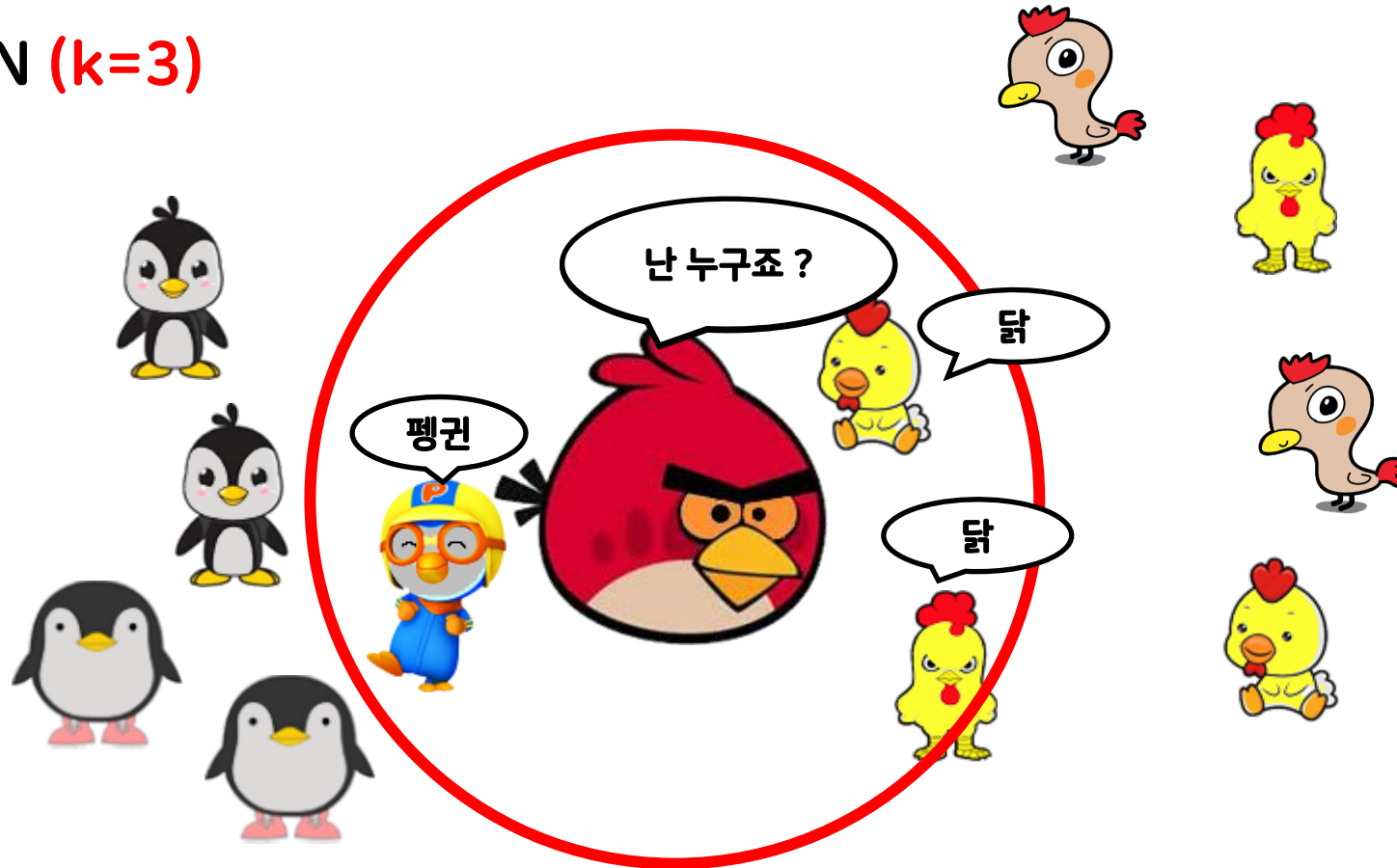
**BMI 데이터에서 성별 특성을 추가하여  
과소적합을 해소해보자**

## k-최근접 이웃 알고리즘

- 새로운 데이터 포인트와 가장 가까운 훈련 데이터셋의 데이터 포인트를 찾아 예측
- k 값에 따라 가까운 이웃의 수가 결정
- 분류와 회귀에 모두 사용 가능

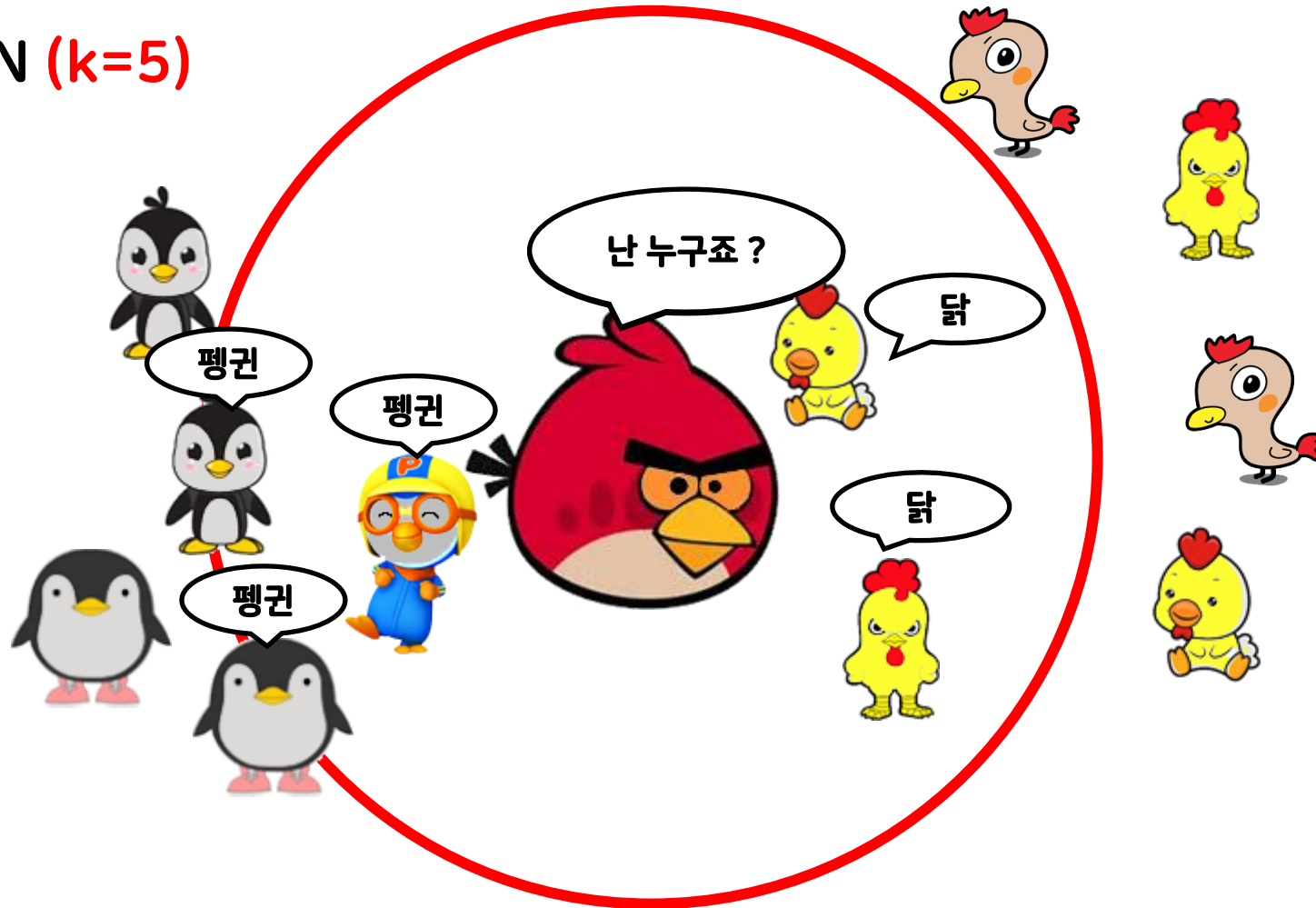
## k-최근접 이웃 알고리즘

KNN (k=3)



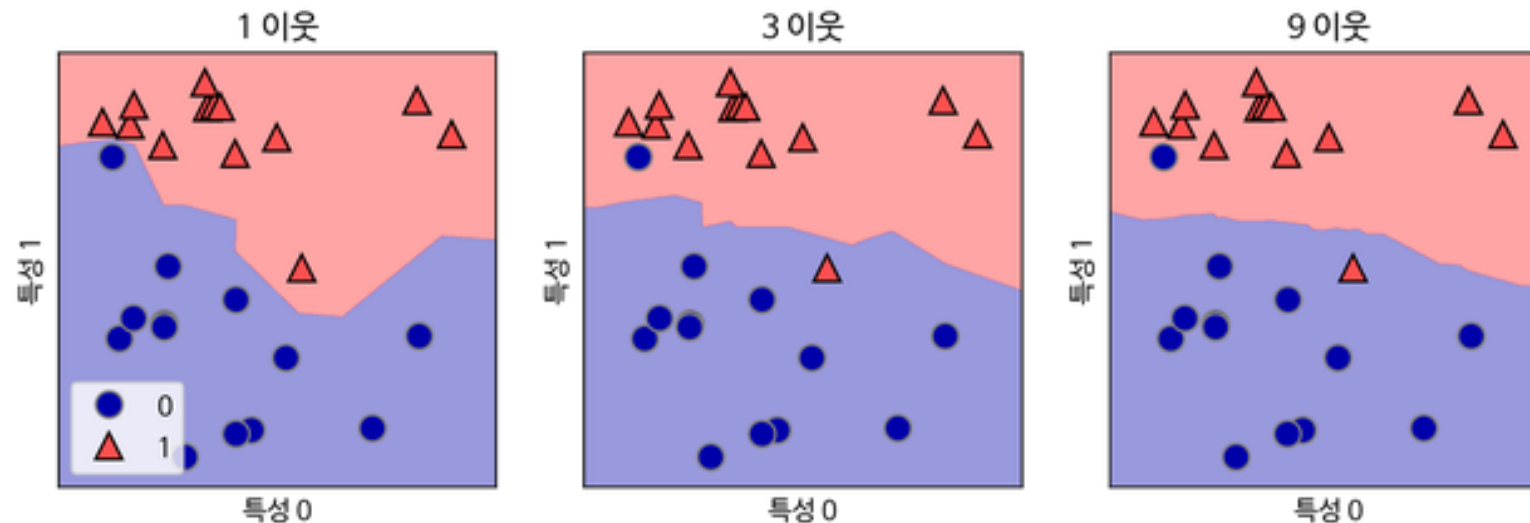
## k-최근접 이웃 알고리즘

KNN (k=5)



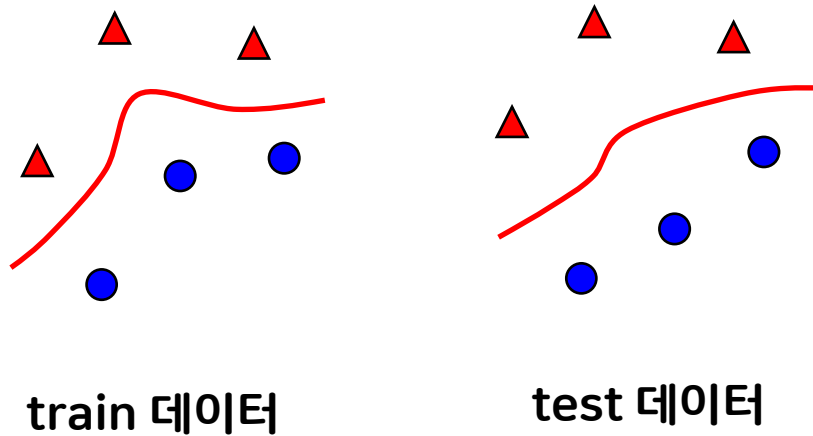
## k-최근접 이웃 알고리즘

- **결정경계 (Decision Boundary)** : 클래스 분류하는 경계
  - 이웃이 적을수록 모델의 복잡도 상승 → **과대적합**
  - 이웃이 전체 데이터 개수와 같아지면 항상 가장 많은 클래스로 예측 → **과소적합**



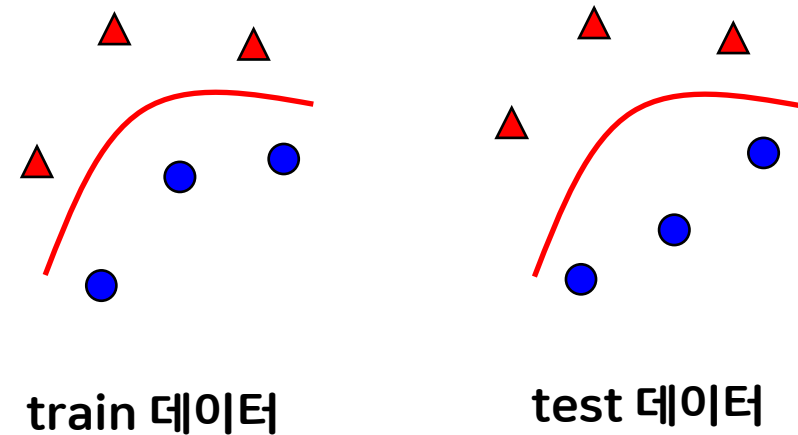
## k-최근접 이웃 알고리즘

적은 이웃 → 과대 적합



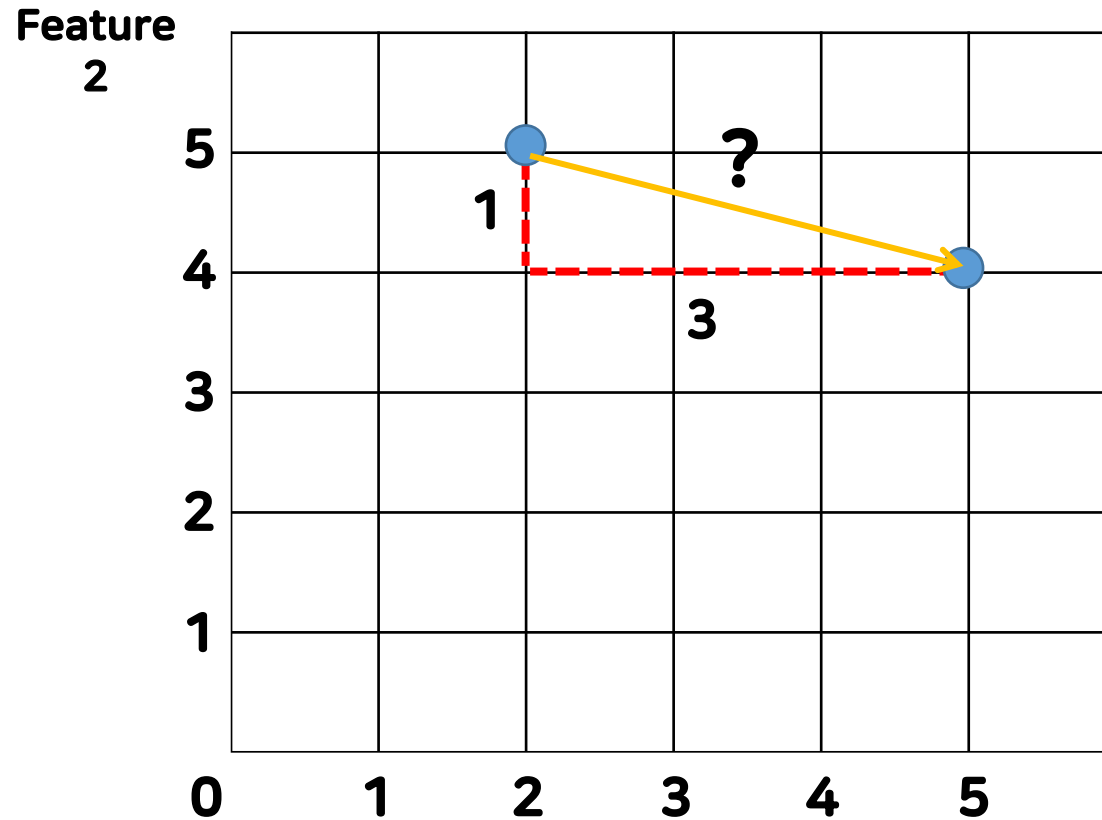
데이터가 살짝만 달라져도 결정 경계가  
달라져서 정확도가 떨어짐

많은 이웃 → 과소 적합

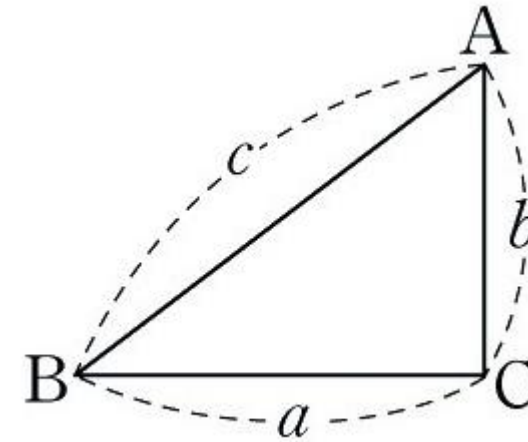


데이터가 달라져도 결정 경계가  
변하지 않아서 정확도가 올라감

## 데이터 포인트(sample) 사이 거리 값 측정 방법



$$a^2 + b^2 = c^2 \text{ (피타고라스의 정리)}$$





## 데이터 포인트(sample) 사이 거리 값 측정 방법

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

**유클리디언 거리공식 (Euclidean Distance)**

## 주요 매개변수(Hyperparameter)

- 거리측정 방법, 이웃의 수, 가중치 함수

**KNeighborsClassifier(n\_neighbors = ?, P = ?, weight = ?)**

P : 거리 계산 방법

- 1 : 맨하튼 방식
- 2 : 유클리디언 방식

n\_neighbors : 이웃의 수

weight : 가중치 함수

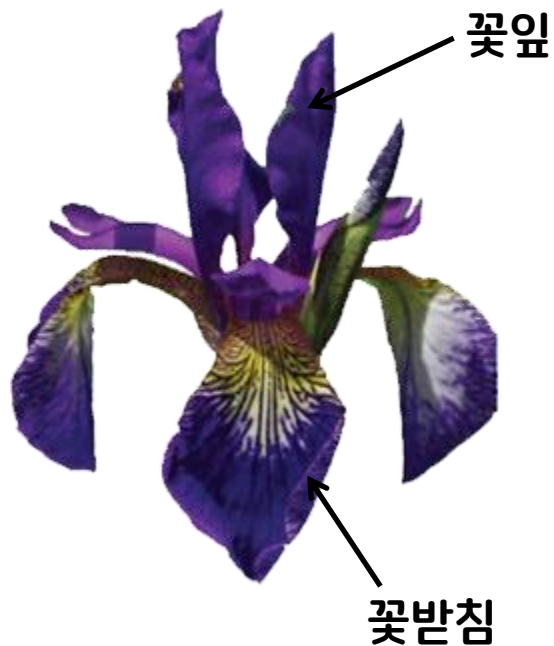
- uniform : 가중치를 동등하게 설정.
- distance : 가중치를 거리에 반비례하도록 설정

## 장단점

- 이해하기 매우 쉽고 조정 없이도 좋은 성능을 발휘하는 모델
- 훈련 데이터 세트가 크면(특성, 샘플의 수) 예측이 느려진다
- 수백 개 이상의 많은 특성을 가진 데이터 세트와 특성 값 대부분이 0인 희소(sparse)한 데이터 세트에는 잘 동작하지 않는다
- 거리를 측정하기 때문에 같은 scale을 같도록 정규화 필요
- 전처리 과정이 중요, 잘 쓰이지는 않음

## iris 데이터를 이용한 KNN 분류 실습

## 붓꽃(iris) 데이터셋



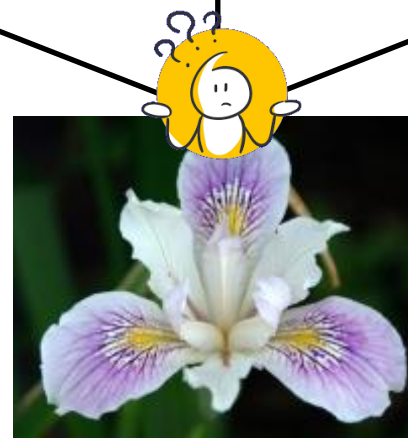
setosa



virginica



versicolor



## 붓꽃(iris) 데이터셋

- 150개의 데이터
- 4개의 특성과 1개의 클래스(3개의 품종)로 구성

	sepal_length	sepal_width	petal_length	petal_width	species
	꽃받침 길이	꽃받침 넓이	꽃잎 길이	꽃잎 넓이	품종
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	
3	4.7	3.2	1.3	0.2	
...					
150	5.9	3.0	5.1	1.8	Iris-virginica

## train\_test\_split() 함수

- 데이터 셋에서 훈련데이터와 테스트데이터로 분리하는 기능

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,  
                                                    test_size = 0.3,  
                                                    random_state=0)
```

X : 특성 데이터

y : 라벨 데이터

test\_size : 테스트 셋의 비율

random\_state : 선택할 데이터 시드