2 주 차 교 육

ToBig's 10기 황이은

KNN

K - Nearest Neighborhood

Contents

Unit 01 | KNN 원리

K

K 개의

N

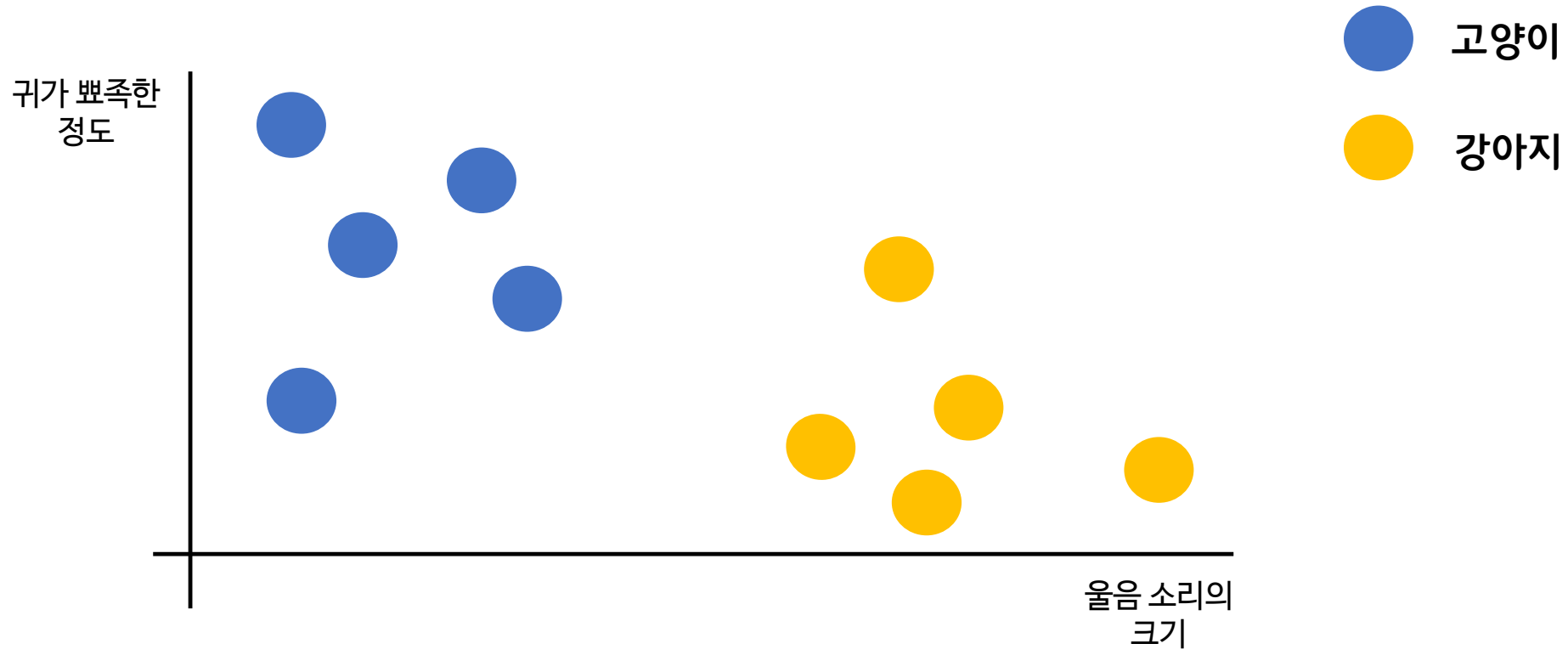
Nearest
가까운

N

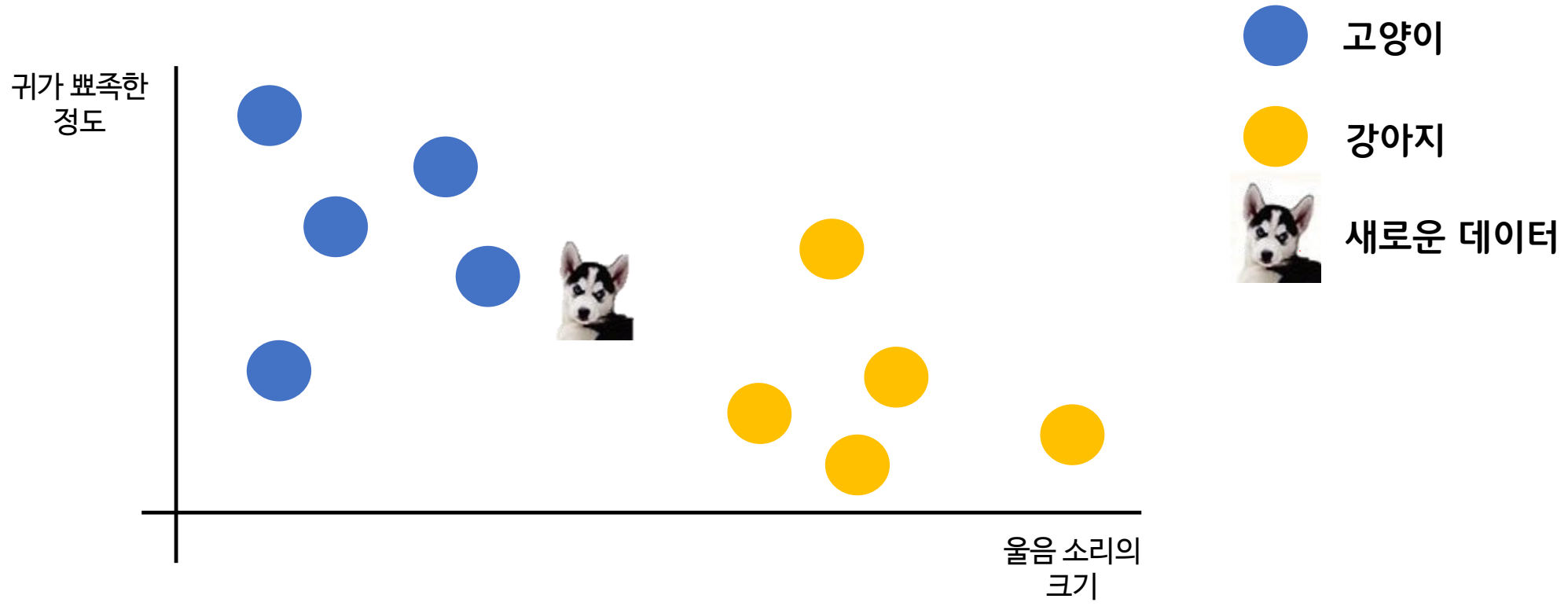
Neighborhood
이웃

Unit 01 | KNN 원리

기존 데이터

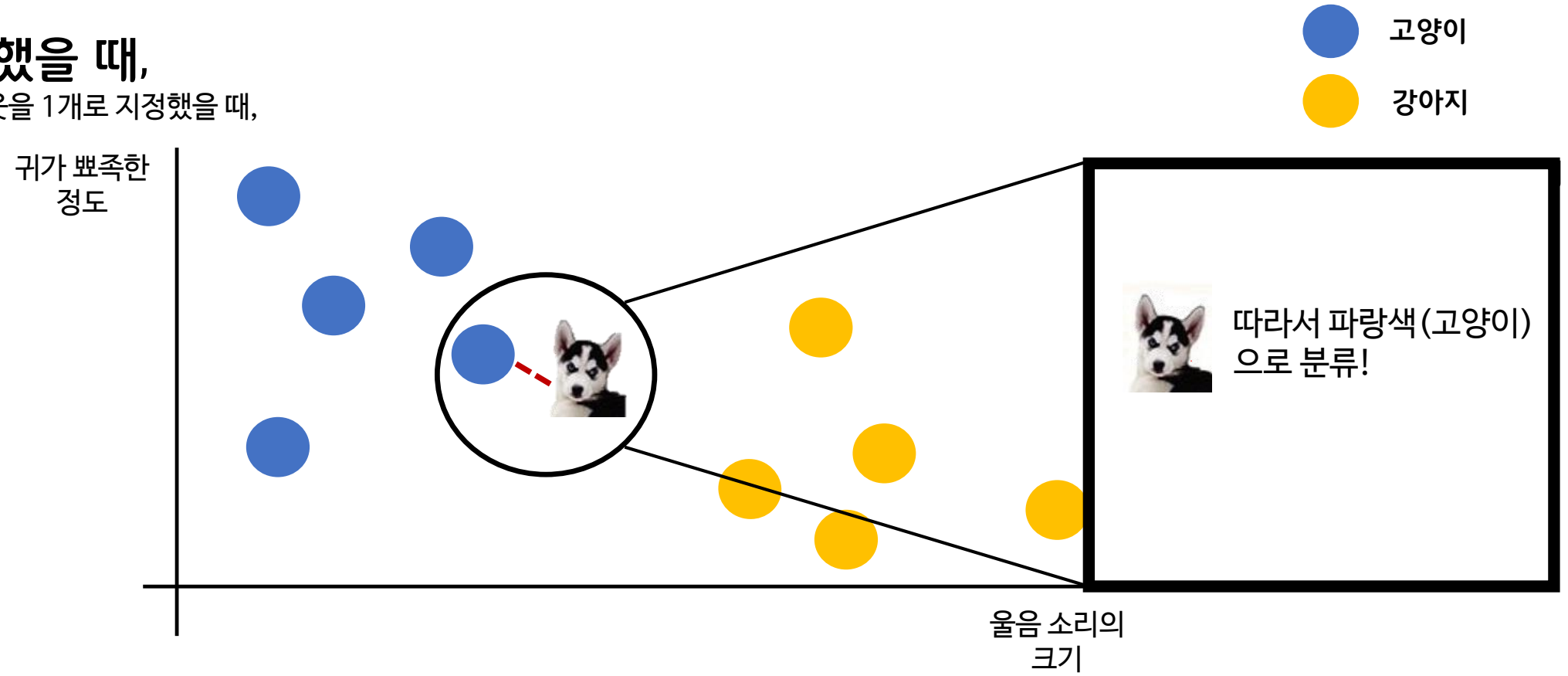


새로운 데이터 유입



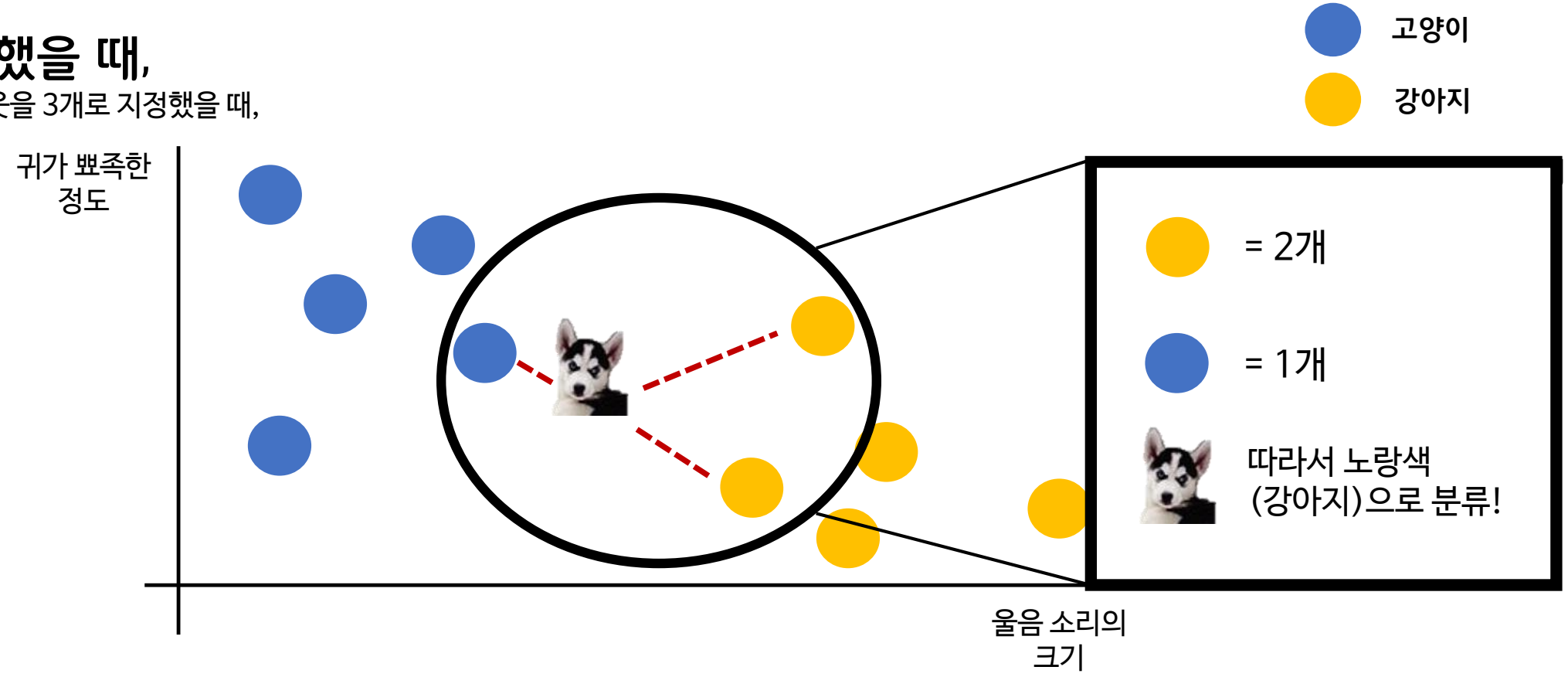
K = 1으로 했을 때,

즉, 가장 가까운 이웃을 1개로 지정했을 때,



K = 3으로 했을 때,

즉, 가장 가까운 이웃을 3개로 지정했을 때,



Unit 01 | KNN 원리

*기억!!!학습이 아님
lazy learner.

*train data

KNN : 기억하고 있는 학습 데이터 중
k개의 가장 가까운 사례를 사용하여 수치 예측 및 분류

*Test data를

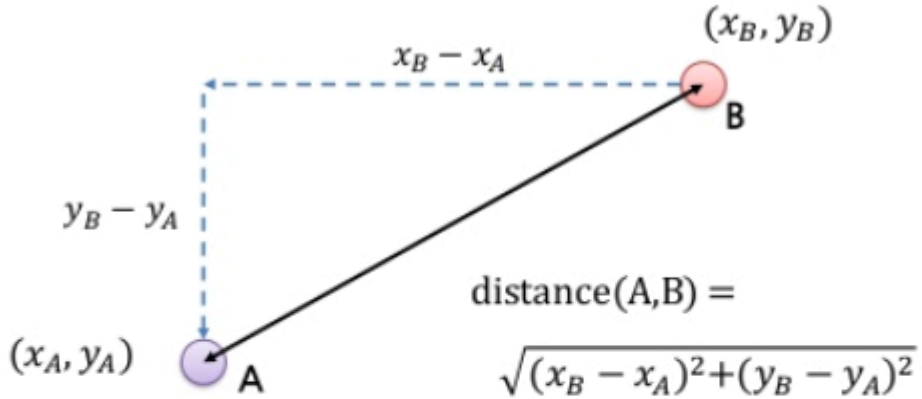
*회귀도 가능

거리
(distance)

K

거리 지표 – 유클리드 거리

두 점 사이의 직선거리



$$\text{distance(A,B)} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

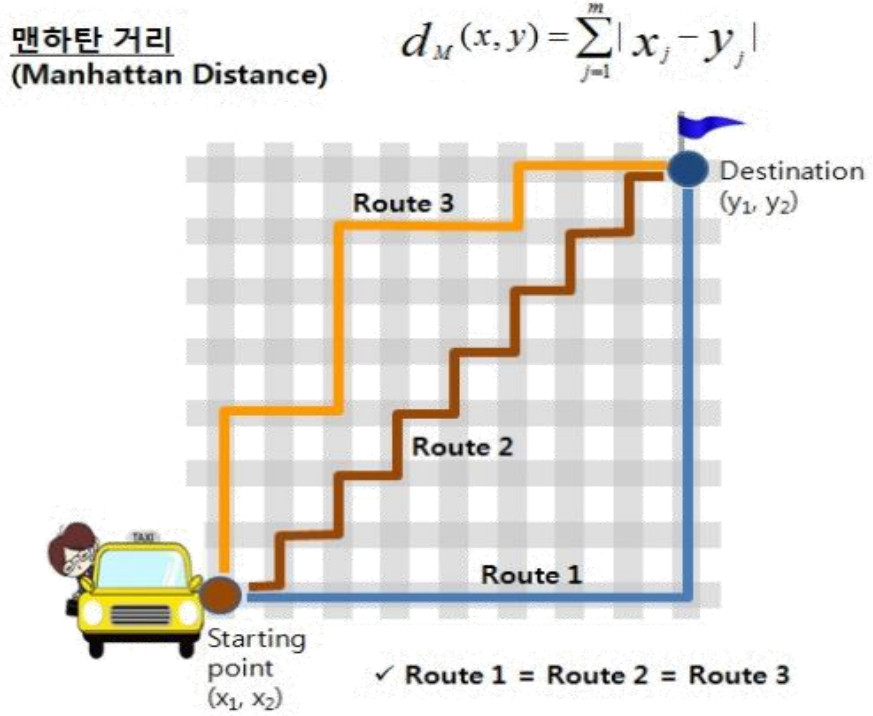
m차원 공간

$$\text{distance(A,B)} = \sqrt{(d_{1,A} - d_{1,B})^2 + (d_{2,A} - d_{2,B})^2 + (d_{3,A} - d_{3,B})^2 + \cdots + (d_{m,A} - d_{m,B})^2}$$

가장 많이 쓰임(디폴트 값)

거리 지표 – 맨하튼 거리

한 축 방향으로만 움직이는 것



점과 점 사이의 이동 시간으로 근접성을 따질 때 좋은 지표가 됨

Unit 02 | 거리 지표

	Manhattan Distance	Euclidean Distance
k=1	78.42% (exponent=10)	81.86% (exponent=10)
k=3	86.00% (exponent=10)	86.57% (exponent=9)
k=5	86.42% (exponent=10)	86.57% (exponent=5)

TABLE II. WEIGHTED KNN CLASSIFICATION RESULTS

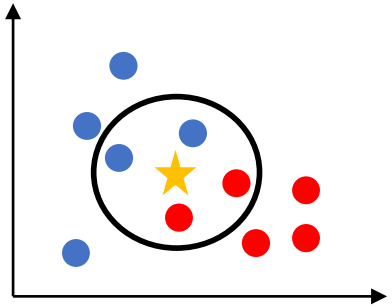
거리에 따라서 성능의 차이가 존재

K 선택하기

K(이웃의 갯수)를 어떻게 설정할지

보통 K는 홀수로 지정

왜? 짝수면 2 대 2의(동률) 상황이 발생할 수 있기 때문!



K의 수?? 데이터에 따라 다르다!

노이즈가 없는 데이터 - 1이 좋음 (EX. 손글씨 데이터MNIST)

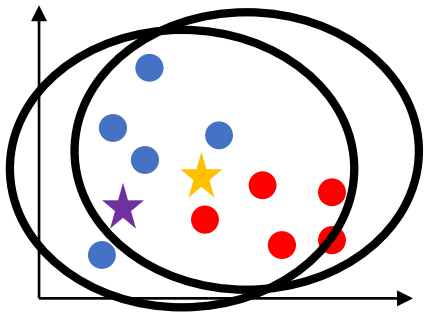
노이즈가 많은 데이터 - K가 클 수록 좋음(EX. 고객 대출 데이터)

K 선택하기

K를 어떻게 설정할지

K =1000000(엄청 큰 숫자)

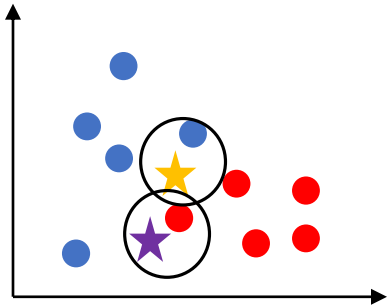
K가 클 때
Underfitting



적절한 K를 찾는게 중요

K =1

K가 작을 때
Overfitting

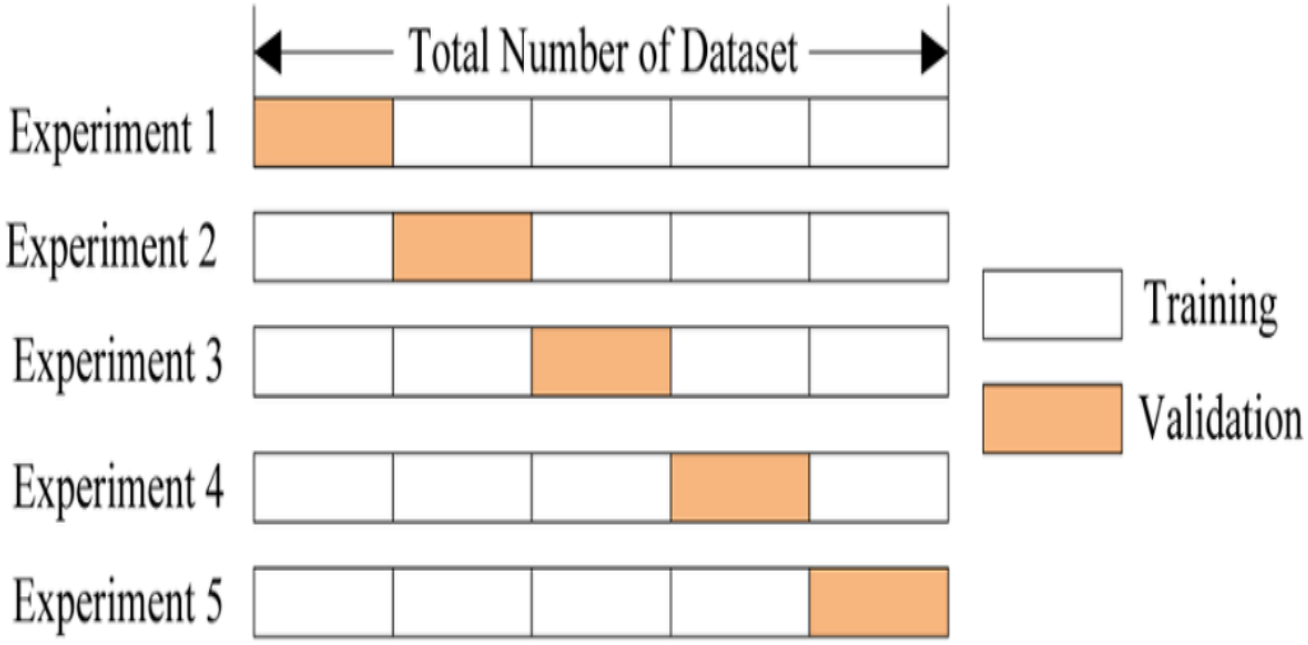


Validation set의 정확도를 보고 적절한 k를 선택하는 경우가 많음

*KNN 뿐만 아니라 머신러닝에서 하이퍼파라미터를 조정할 때 많이 쓰이는 방법

K-fold cross-validation

K개의 폴드로 교차타당성 검사 하겠다.



순서

1. Train set을 k개의 fold로 나눈다
2. 위는 5개의 fold로 나눴을 때의 모습
3. 한 개의 fold에 있는 데이터를 다시 k개로 쪼갠 다음, k-1개는 train으로 나머지 1개는 validation set으로 지정한다.
4. 모델을 생성하고 예측을 진행하여, 이에 대한 에러값을 추출한다.
5. 다음 fold에서는 validation set을 바꿔서 지정하고, 이전 fold에서 validation역할을 했던 set을 train set에 넣는다.
6. 이를 k번 반복한다.

Unit 04 | 주의해야 할 점 및 KNN의 장단점

주의해야 할 점

1. Scaling
2. one-hot 인코딩
3. wknn/회귀/분류
4. 가중치(확률)

Unit 04 | 주의해야 할 점 및 KNN의 장단점

1. Scaling

Knn은 거리기반! 같은 뜻이여도 값이 달라질 수 있다!

X1	X2($)
1	5
2	6
4	4

$$Distance(1,3) = \sqrt{(1-4)^2 + (5-4)^2}$$
$$= 3.162278$$

$$Distance(2,3) = \sqrt{(2-4)^2 + (6-4)^2}$$
$$= 2.828427$$

X1	X2(₩)
1	5000
2	6000
4	4000

$$Distance(1,3) = \sqrt{(1-4)^2 + (5000-4000)^2}$$
$$= 1000.004$$

$$Distance(2,3) = \sqrt{(2-4)^2 + (6000-4000)^2}$$
$$= 2000.001$$

Unit 04 | 주의해야 할 점 및 KNN의 장단점

1. Scaling

Knn은 거리기반! 같은 뜻이여도 값이 달라질 수 있다!

가장 많이 쓰이는 건 min-max normalization이다.

$$X_{new} = \frac{X - min(X)}{max(X) - min(X)}$$

로 표현된다. 0~1 사이의 값을 가지게 된다.

z-score normalization 또한 많이 쓰이며

$$X_{new} = \frac{X - \mu}{\sigma}$$

로 표현된다. 통계에서 자주 보는 방법이다.

2. one-hot 인코딩

Knn은 거리기반! Input에 int형이 들어가야 한다!

그렇다면 범주형은 어떻게 할까?? - 더미변수 만들어주기! (one-hot 인코딩)

	A	B	C	D	E	F	G	H	I
1	Original data:			One-hot encoding format:					
2	id	Color		id	White	Red	Black	Purple	Gold
3	1	White		1	1	0	0	0	0
4	2	Red		2	0	1	0	0	0
5	3	Black		3	0	0	1	0	0
6	4	Purple		4	0	0	0	1	0
7	5	Gold		5	0	0	0	0	1
8									

흰색에 해당하는 것만 1을 넣고 나머지는 0으로 채운다!

Unit 04 | 주의해야 할 점 및 KNN의 장단점

3. wknn회귀 / 분류

Weighted K-NN

단순히 평균, 다수결로 값을 결정하지 않고
거리에 따라서 영향력을 달리 주고 싶을 때 사용

$$\text{유사도} = \frac{1}{\text{거리}} \qquad \text{가중치} = \frac{\text{유사도}}{\text{모든 이웃의 유사도 합}}$$

Unit 04 | 주의해야 할 점 및 KNN의 장단점

3. wknn/회귀 / 분류

Knn으로 회귀(regression)를 할 수 있다!

평균 값으로 계산되어 나옴

$$\text{유사도} = \frac{1}{\text{거리}} \qquad \text{가중치} = \frac{\text{유사도}}{\text{모든 이웃의 유사도 합}}$$

New Data
(K=4)

이웃	체지방률	거리	유사도	가중치
N1	15.4	1	1	0.48
N2	17.2	2	0.5	0.24
N3	12.3	3	0.33	0.16
N4	11.5	4	0.25	0.12

KNN

= (15.4+17.2+12.3+11.5)/4

= 14.1

W-KNN

= (15.4*0.48+17.2*0.24+12.3*0.16+11.5*0.12)

= 14.868

3. wknn/회귀 / 분류

Knn / w-knn 분류 비교

$$\text{유사도} = \frac{1}{\text{거리}} \qquad \text{가중치} = \frac{\text{유사도}}{\text{모든 이웃의 유사도 합}}$$

New Data
(K=4)

클래스	이웃	특성1	특성2	거리	유사도	가중치
A	N1	0.012	0	1	1	0.48
B	N2	0.179	1	2	0.5	0.24
C	N3	0	0.147	3	0.33	0.16
B	N4	1	0.237	4	0.25	0.12

KNN

A:1, B:2, C:1

B로 분류

W-KNN

A:1*0.48, B:1*0.24+1*0.12, C:1*0.16

A로 분류

KNN의 장단점

장점

알고리즘이 간단하여 구현하기 쉽다.

수치 기반 데이터 분류 작업에서 성능이 좋다.

직관적이다.

단점

학습 데이터의 양이 많으면 분류 속도가 느려진다.

차원(벡터)의 크기가 크면 계산량이 많아진다.

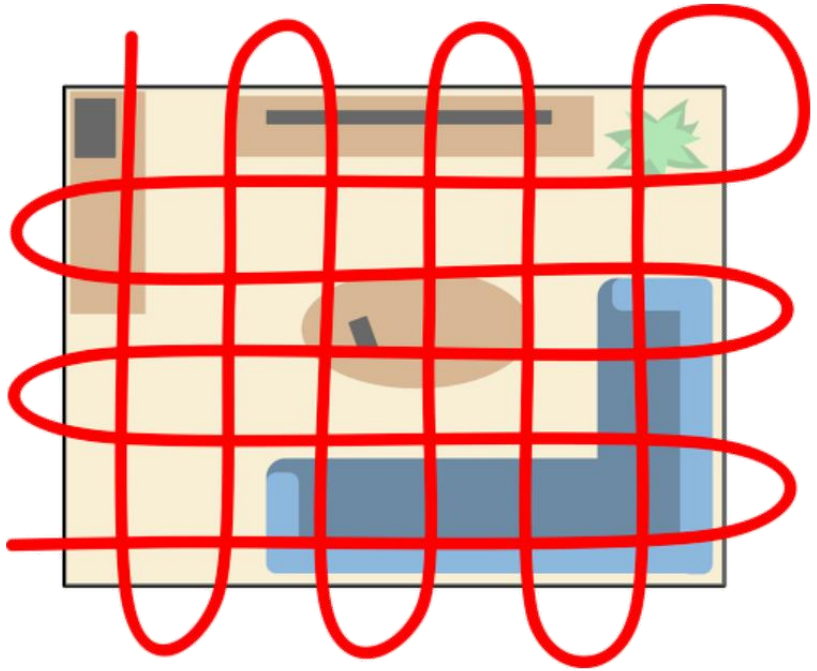
비모수적이다.

*(유클리디안 거리: 서로 각각 거리를 구하게 됨)

정리

1. KNN은 K개의 가까운 이웃이다.
2. 자기와 가장 가까운 이웃의 개수만큼 인풋의 라벨이 결정된다.
3. KNN은 분류와 회귀에서 쓰인다.
4. 단순히 vote뿐만 아니라, 거리 가중치를 줄 수 있고 확률값도 반환해 줄 수 있다.
5. KNN의 하이퍼파라메터는 K와 거리이다.
6. KNN은 거리기반이기 때문에 scaling이 중요하다!

그리드 서치



격자무늬로 하이퍼파라메터를 탐색하는 것

모든 parameter의 경우의 수에 대해 cross-validation 결과가
가장 좋은 parameter를 고르는 방법

주어진 공간 내에서 가장 좋은 결과를 얻을 수 있다는 장점이 있지만,
시간이 정말 정말 오래 걸린다는 단점이 존재

코드 실습

실습: iris 데이터

Train, test set 입력 종속변수 K의 개수

```
library(class)
model.knn <- knn(train[-5],test[-5], train$Species, k=5)
confusionMatrix(model.knn, test$Species) #Accuracy : 0.9556
```

```
## min-max 스케일링
normalize <- function(x){
  return( (x-min(x))/(max(x)-min(x)) )
}

iris_normal <- as.data.frame(lapply(iris[-5], normalize))
summary(iris_normal)
```

함수를 만들어 줌

Iris데이터에서 5번째 열(species-종속변수)을 제외하고
normalize 함수를 적용해라. 결과값은 데이터프레임 형태로 나타내라

코드 실습

```
> summary(iris_normal)
  Sepal.Length     Sepal.Width      Petal.Length     Petal.Width
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333
 Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000
 Mean   :0.4287   Mean   :0.4406   Mean   :0.4675   Mean   :0.45806
 3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

모든 값이 0~1로 조정됨!
*종속변수는 다시 붙여주기!

코드 실습

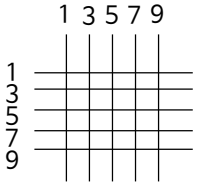
```
#그리드 서치 cv
cv <- trainControl(method = "cv", number = 5, verbose = T)

knn.grid = expand.grid(
  .k = c(1,3,5,7,9)
)

train.knn <- train(Species~.,train_n, method = "knn",trControl = cv,
                   tuneGrid = knn.grid)
train.knn$results
train.knn$bestTune #9
predict.knn <- predict(train.knn, test_n)
confusionMatrix(predict.knn, test_n$Species) #0.9778
```

Cross-validation으로!

K의 개수

= 즉 5-fold cross validation으로 하겠다.

그리드 만들기.
K의 파라미터 벡터 스페이스는 1,3,5,7,9로 한정



종속변수

데이터-정규화 해준 것을 7:3으로 나눈 train데이터

위에서 정의한 5-fold cross validation으로!

위에서 정의한 그리드로!

참고자료

knn
https://www.youtube.com/watch?v=lDCWX6vCLFA

https://www.youtube.com/watch?v=09mb78oiPkA

투빅스 9기 김영제 강의자료

k-fold validtaion
https://nonmeyet.tistory.com/entry/KFold-Cross-Validation%EA%B5%90%EC%B0%A8%EA%B2%80%EC%A6%9D-%EC%A0%95%EC%9D%98-%EB%B0%8F-%EC%84%A4%EB%AA%85

그리드 서치
http://sanghyukchun.github.io/99/

들어주셔서 감사합니다.