정규 교육 세미나

ToBig's 11기 정혜인

K N N

K - Nearest Neighbor

Contents

Unit 01 | KNN

Model-based Learning

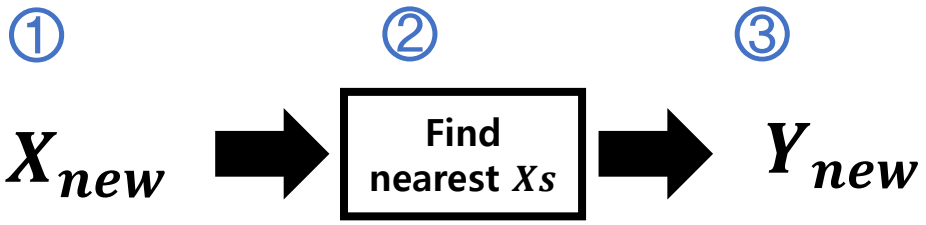
- 선형/비선형모델 (ex. Linear regression, logistic regression)
- Neural network
- Decision tree
- Support Vector Machine

→ **데이터로부터 모델을 생성하여 분류/예측 진행**

Instance-based Learning

- K-nearest neighbor
- Locally weighted regression

→ **별도의 모델 생성 없이 인접데이터를 분류/예측에 사용**



Unit 01 | KNN

K **N** **N**

K 개의

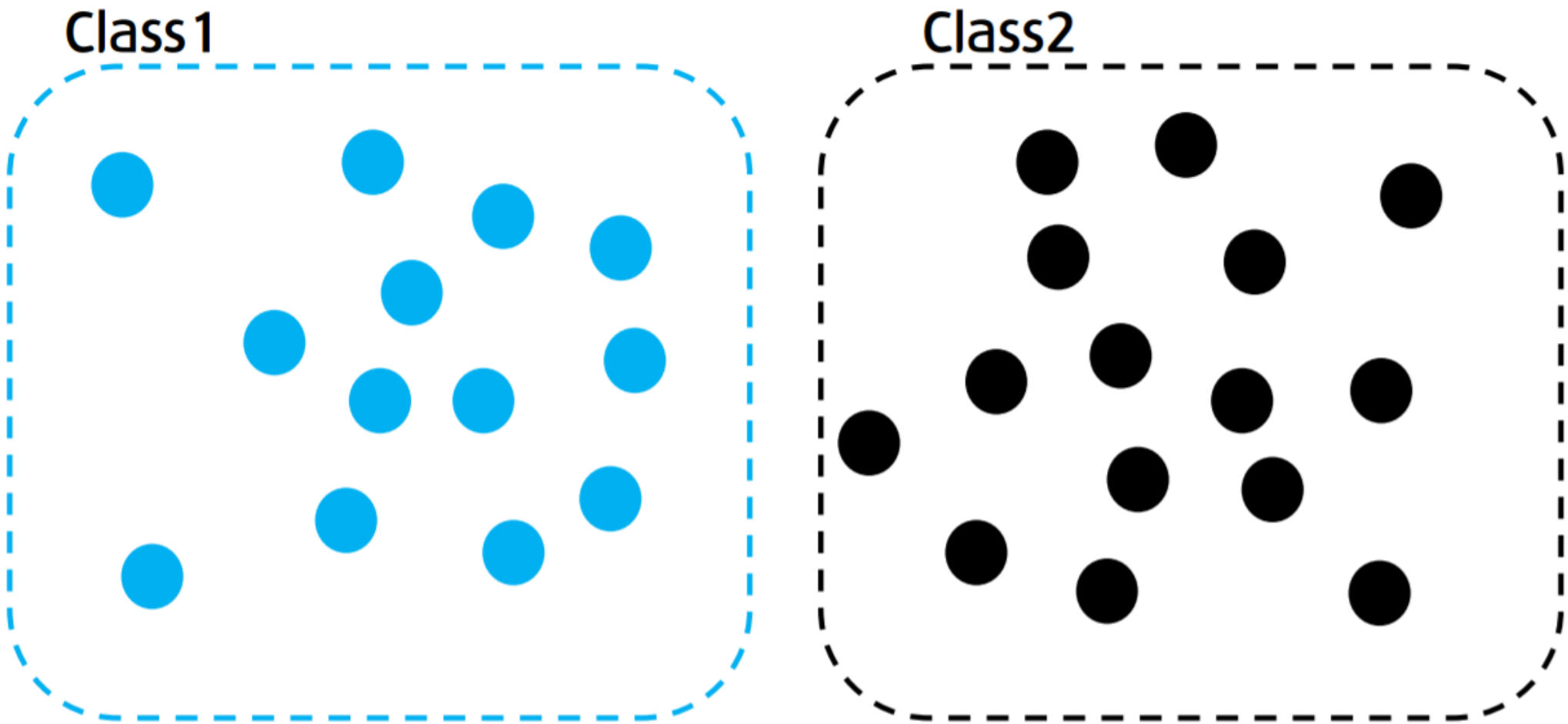
Nearest

가까운

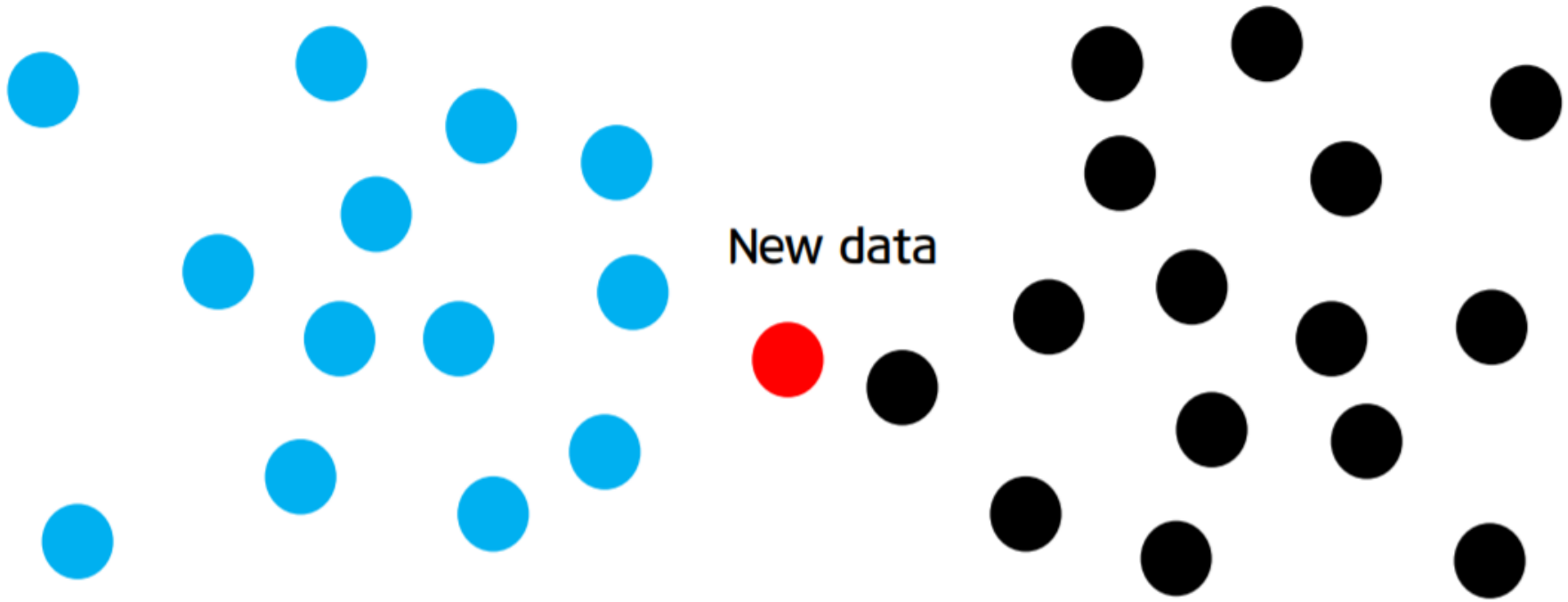
Neighbor

이웃

Unit 01 | KNN

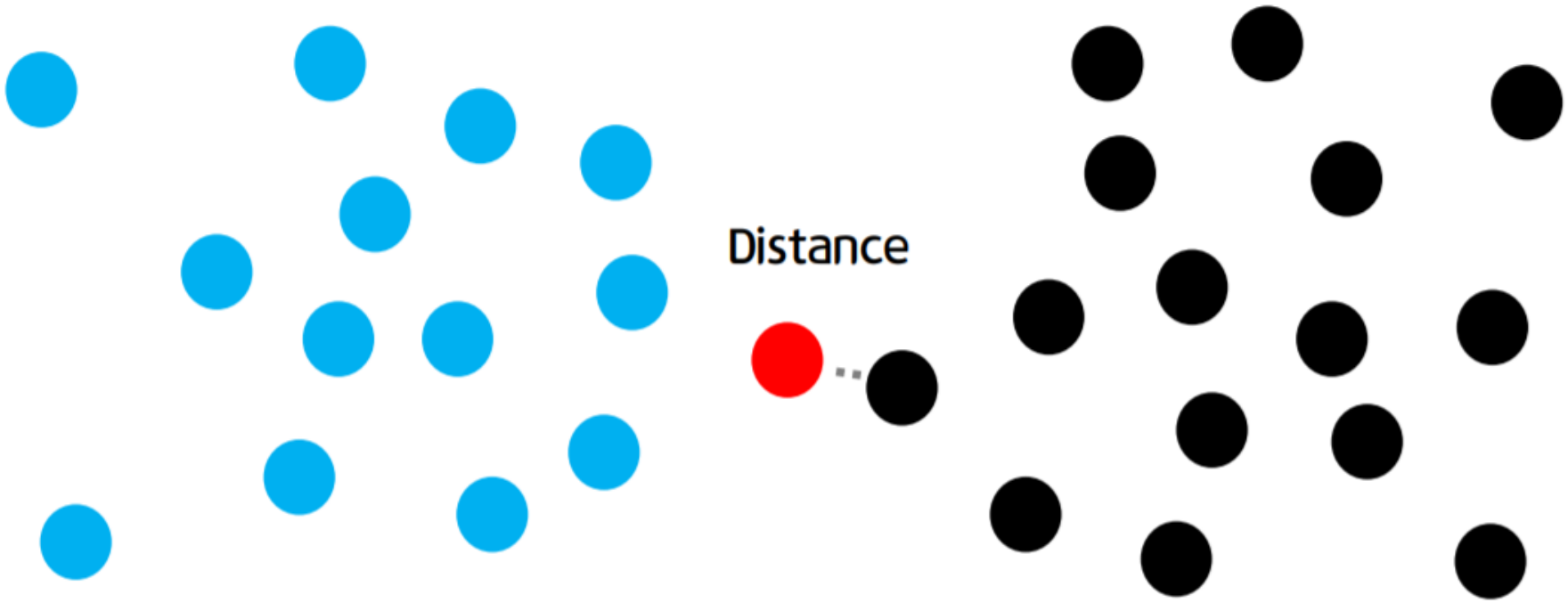


Unit 01 | KNN



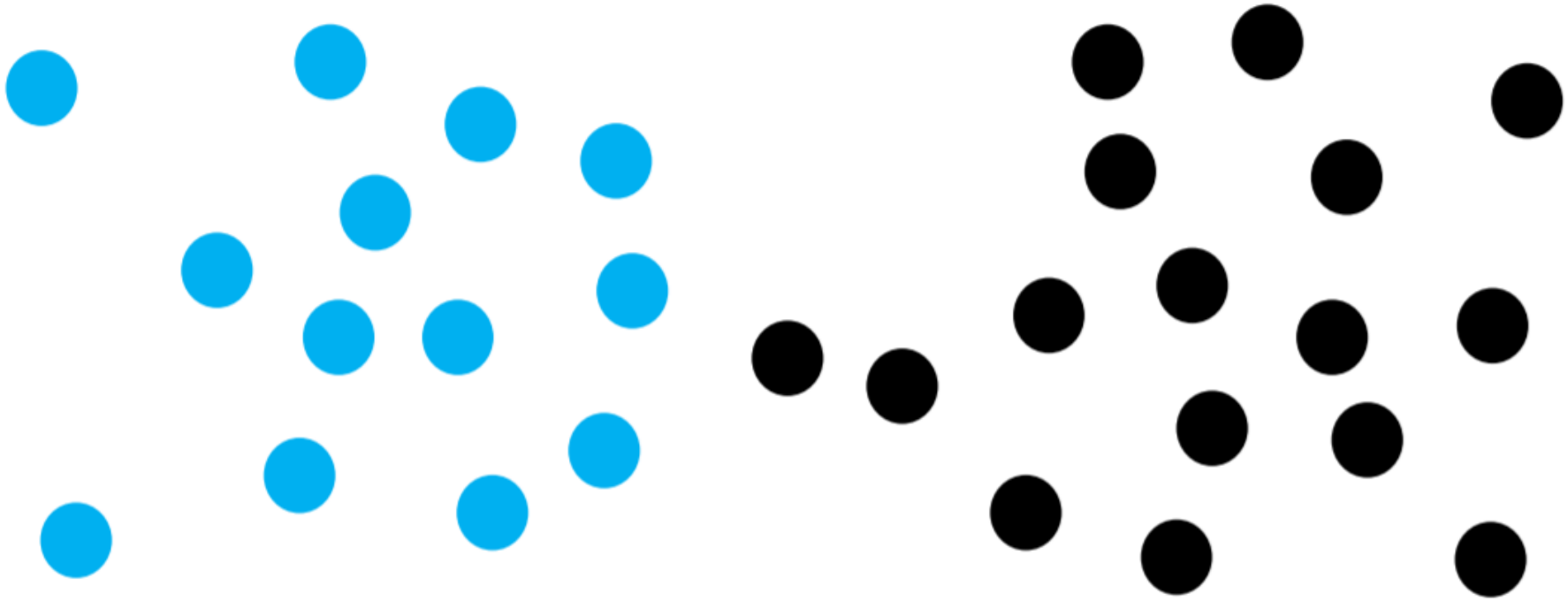
Unit 01 | KNN

K = 1



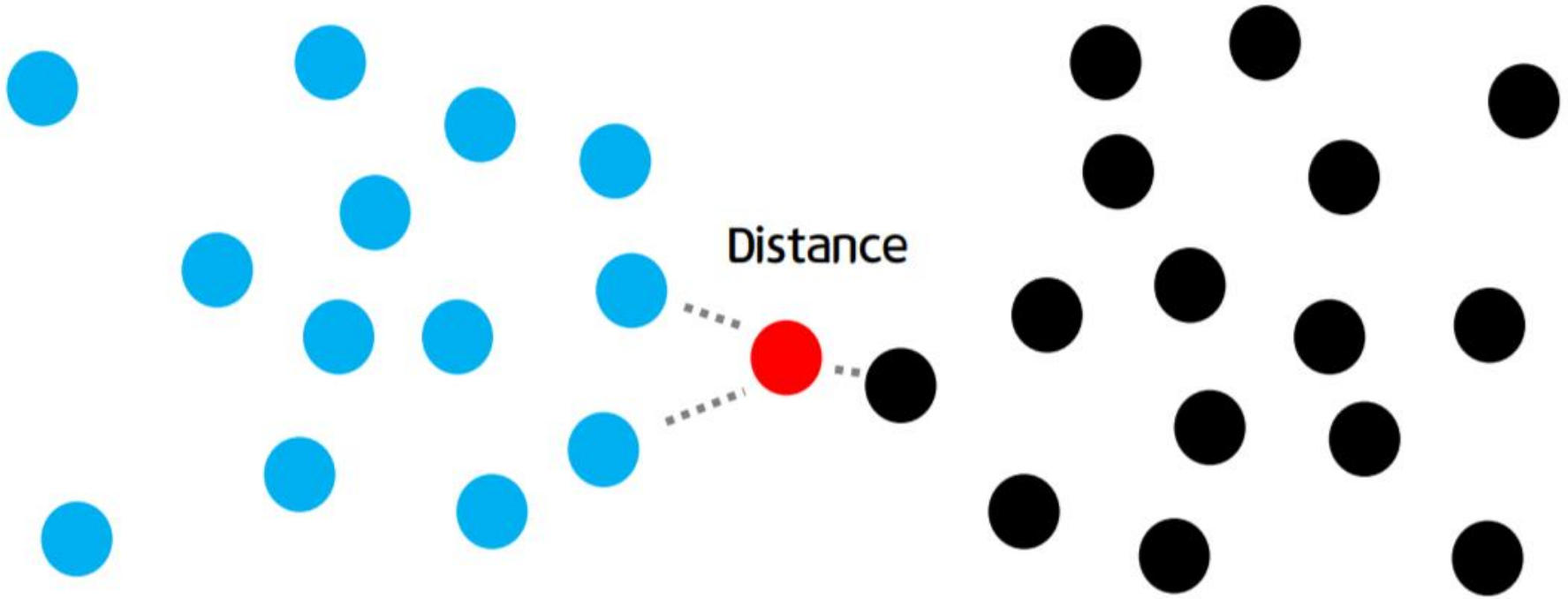
Unit 01 | KNN

K = 1



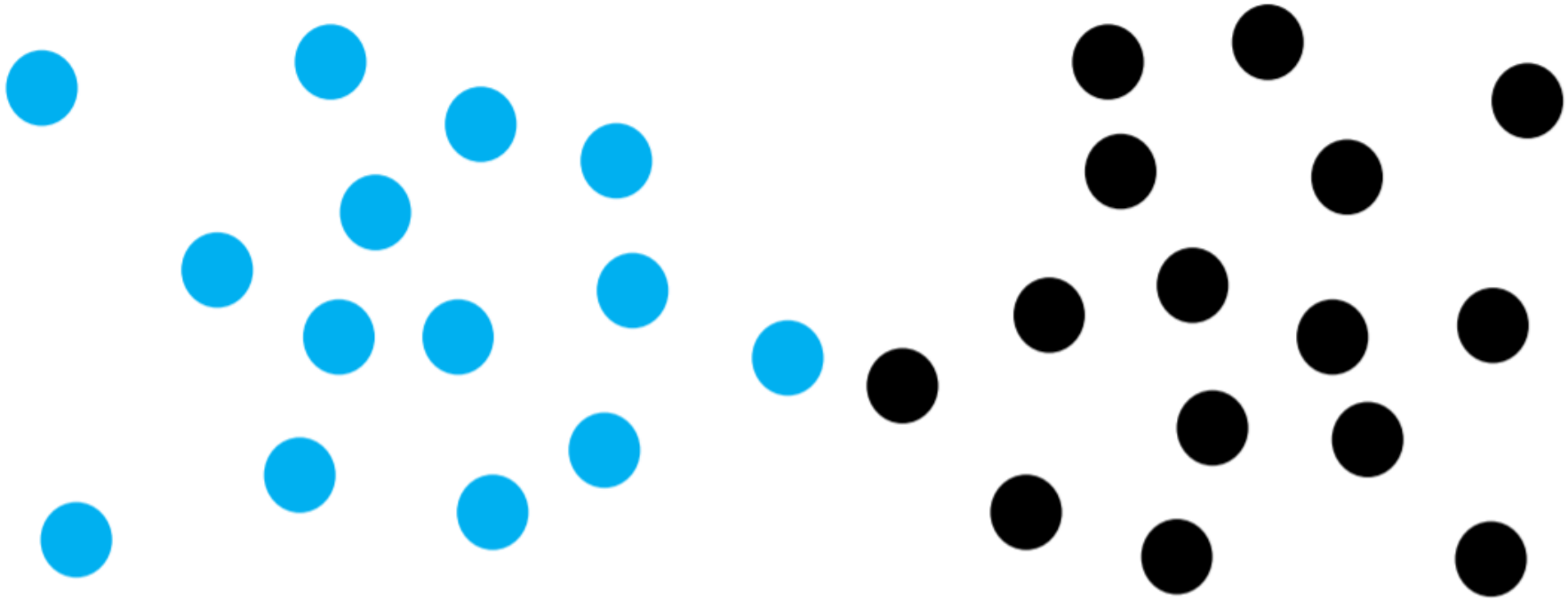
Unit 01 | KNN

K = 3



Unit 01 | KNN

K = 3



Unit 01 | KNN

기억하고 있는 학습 데이터 중

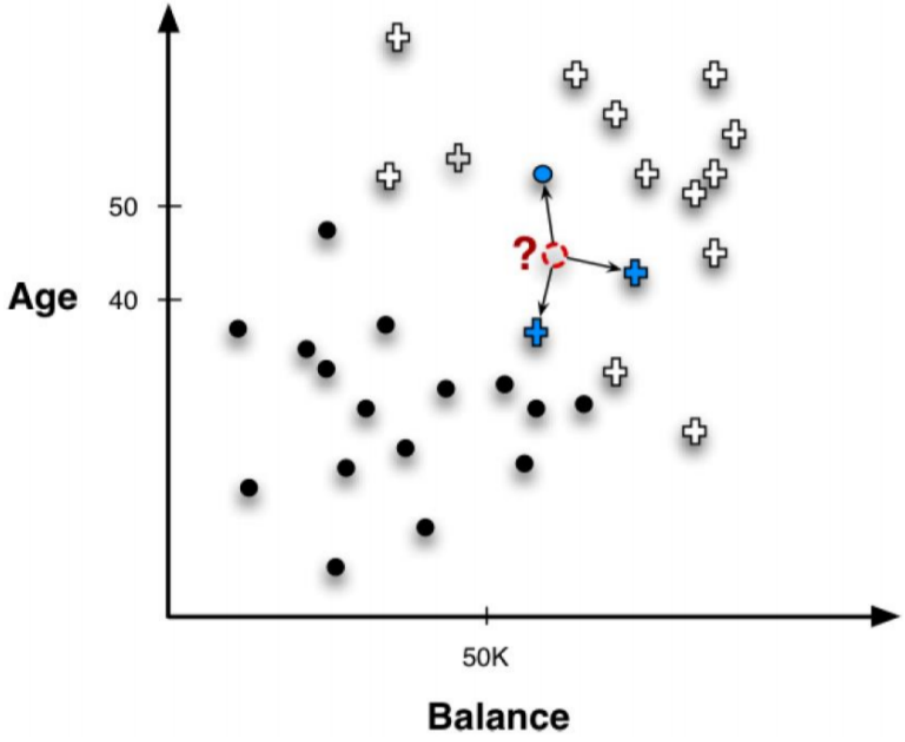
k개의 가장 가까운 사례를 사용하여 **수치 예측** 및 **분류**

Instance-based Learning	각각의 관측치(instance)만을 이용하여 새로운 data에 대한 예측을 진행
Memory-based Learning	모든 학습 data를 메모리에 저장한 후, 이를 바탕으로 예측 시도
Lazy Learning	Delay the process of modeling the training data until it is needed to classify the test instances

Unit 01 | KNN

Example 1 : **Classification** Using Nearest Neighbors

Want to predict the class (label) of a new example



- Basic procedure
  - Retrieve k nearest neighbors
    - (ex) $k = 3$ in this example
  - Consult their target variables
    - (ex) majority vote → "+"

Unit 01 | KNN

Example 2 : **Regression** Using Nearest Neighbors

Want to predict the value of a new example (David's Income)

Customer	Age	Income (1000s)	Cards
David	*37*	*?*	*2*
John	35	35	3
Rachael	22	50	2
Ruth	63	200	1
Jefferson	59	170	1
Norah	25	40	4

- Basic procedure
  - Retrieve k nearest neighbors
    - (ex) $k = 3$ in this example
  - Compute the average (or median) of their target values
    - (ex) $(35 + 50 + 40)/3 \approx 42$

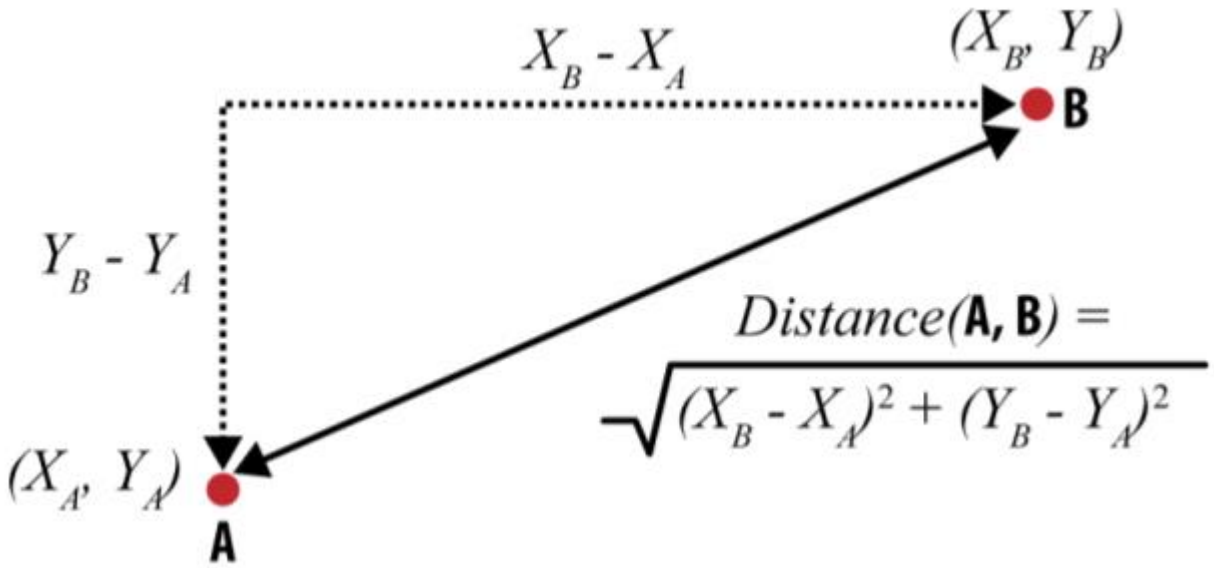
기억하고 있는 학습 데이터 중

k개의 가장 가까운 사례를 사용하여 **수치 예측** 및 **분류**

Hyperparameter : K & Distance Measures

Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 1 : **Euclidean Distance**



- 가장 흔히 사용
- 두 관측치 사이의 직선거리

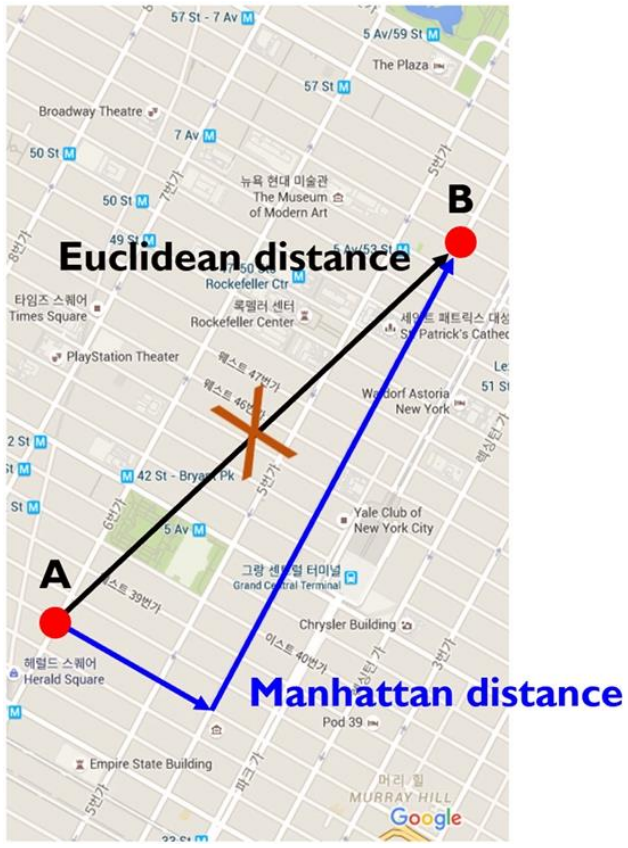
$$d_{\text{Euclidean}}(\mathbf{X}, \mathbf{Y}) = \| \mathbf{X} - \mathbf{Y} \|_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots}$$

Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 2 : **Manhattan Distance**

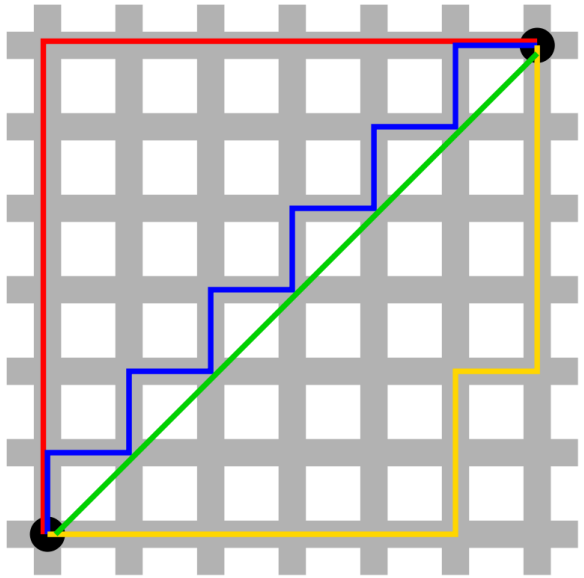
Unit 02 | Hyperparameter 1 – Distance measure

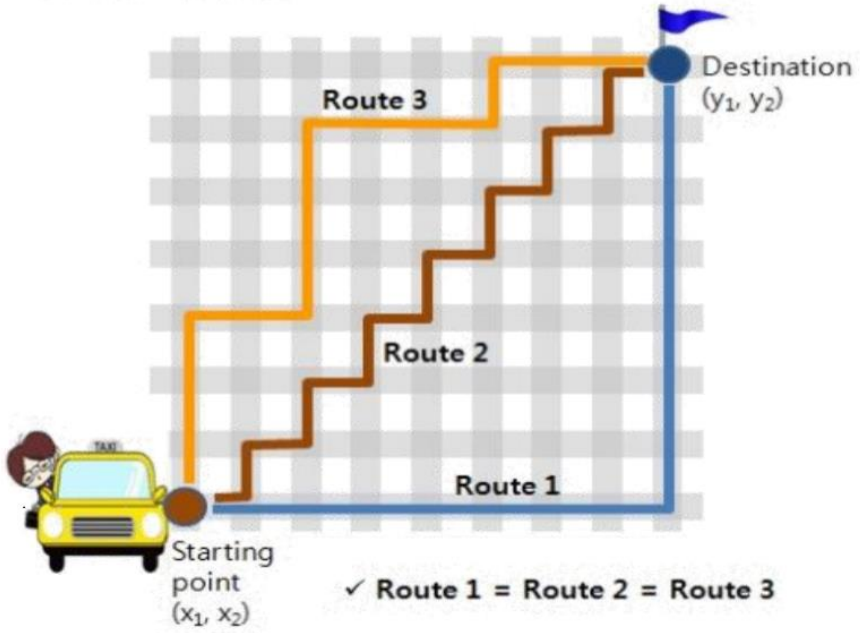
Distance Measure 2 : **Manhattan Distance**



Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 2 : **Manhattan Distance**





$$d_{\text{Manhattan}}(\mathbf{X}, \mathbf{Y}) = \| \mathbf{X} - \mathbf{Y} \|_1 = | x_1 - y_1 | + | x_2 - y_2 | + \cdots$$

Unit 02 | Hyperparameter 1 – Distance measure

Generalization : **Minkowski distance**

$$d(x,y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

r = 1 : **Manhattan Distance** (L_1 norm)

r = 2 : **Euclidean Distance** (L_2 norm)

r = ∞ : **Supremum Distance** (L_∞ norm or L_{max} norm)

$$d(x,\ y) = \lim_{r \to \infty} \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r} = \max_k (|x_k - y_k|)$$

Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 3 : **Mahalanobis Distance**

$$d_{Mahalanobis\,(X,Y)} = \sqrt{(X-Y)^T\Sigma^{-1}(X-Y)},$$

Σ^{-1}: inverse of covariance matrix

- 변수 내 분산, 변수 간 공분산을 모두 반영하여 거리를 계산하는 방식
- Data의 covariance matrix 가 identity matrix인 경우는 Euclidean Distance와 동일함.

Distance Measure 3 : **Mahalanobis Distance**

$\sqrt{(X-Y)^T\Sigma^{-1}(X-Y)} = c$ (c is Mahalanobis distance)

$\rightarrow (X-Y)^T\Sigma^{-1}(X-Y) = c^2$

Let $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} s_{11}^{-1} & s_{12}^{-1} \\ s_{21}^{-1} & s_{22}^{-1} \end{pmatrix}$, then

$\rightarrow (x_1-y_1)^2 s_{11}^{-1} + 2(x_1-y_1)(x_2-y_2)s_{12}^{-1} + (x_2-y_2)^2 s_{22}^{-1} = c^2 \ (\because s_{12}^{-1} = s_{21}^{-1})$

It can be considered as the squared Mahalanobis distance between a certain point X, and the fixed point Y.

Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 3 : **Mahalanobis Distance**

Let $Y = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, then

$$\rightarrow x_1^2 s_{11}^{-1} + 2x_1 x_2 s_{12}^{-1} + x_2^2 s_{22}^{-1} = c^2$$

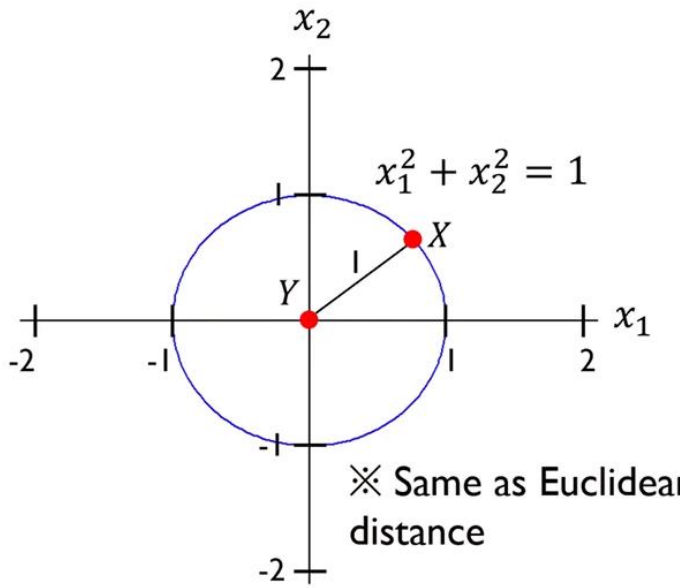
which is a general equation of the ellipse.

Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 3 : **Mahalanobis Distance**

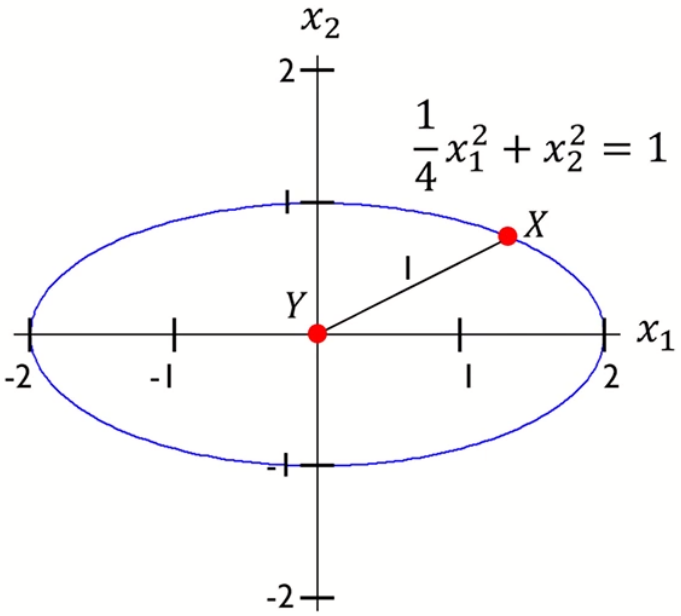
$$\Sigma = \Sigma^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(identity matrix)



※ Same as Euclidean distance

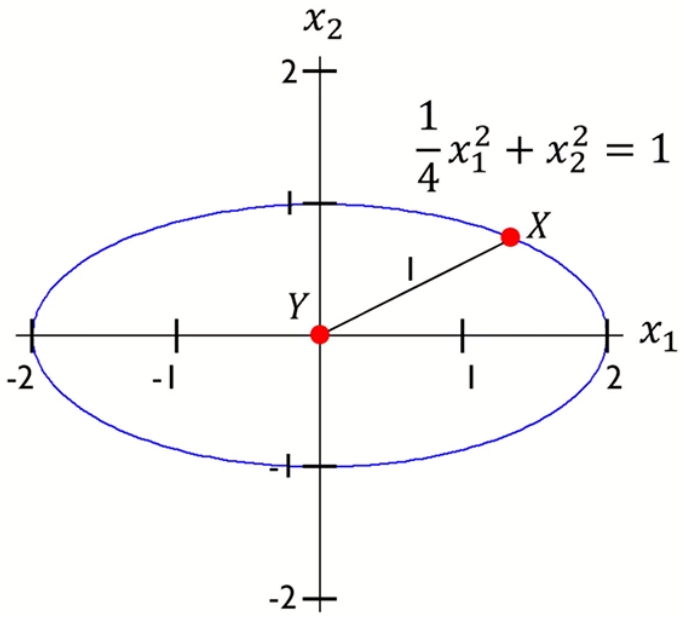
$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/4 & 0 \\ 0 & 1 \end{pmatrix}$$



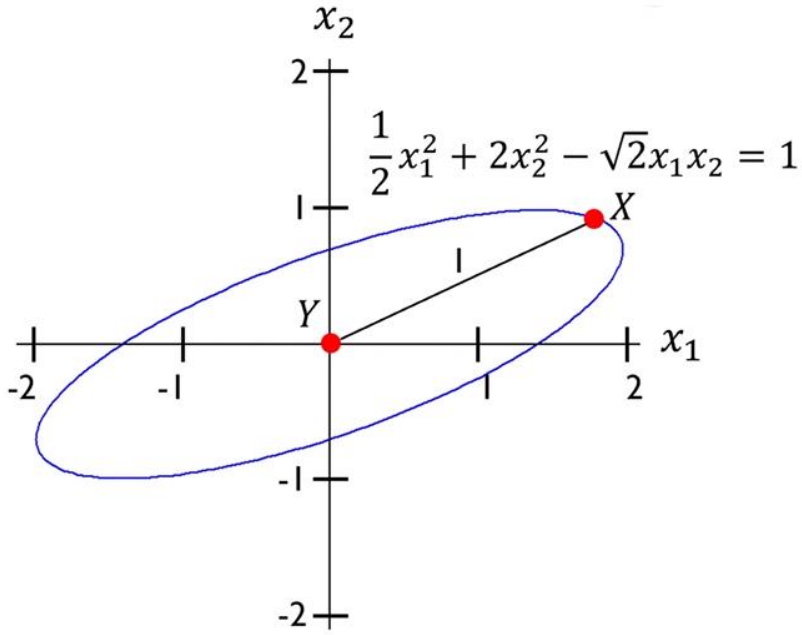
Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 3 : **Mahalanobis Distance**

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/4 & 0 \\ 0 & 1 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 4 & \sqrt{2} \\ \sqrt{2} & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/2 & -\sqrt{1/2} \\ -\sqrt{1/2} & 2 \end{pmatrix}$$



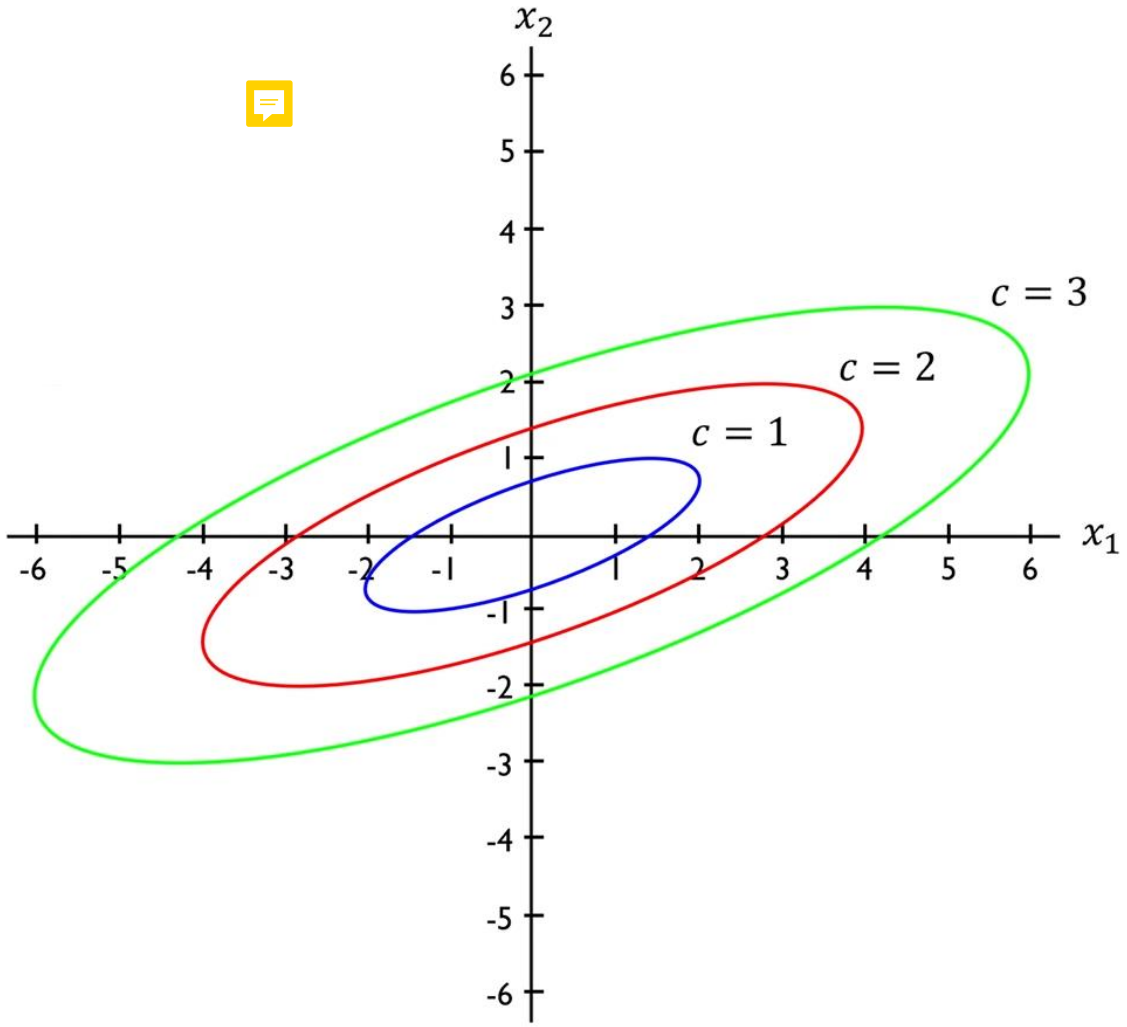
Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 3 : **Mahalanobis Distance**

$$\Sigma = \begin{pmatrix} 4 & \sqrt{2} \\ \sqrt{2} & 1 \end{pmatrix}, \Sigma^{-1} = \begin{pmatrix} 1/2 & -\sqrt{1/2} \\ -\sqrt{1/2} & 2 \end{pmatrix}$$

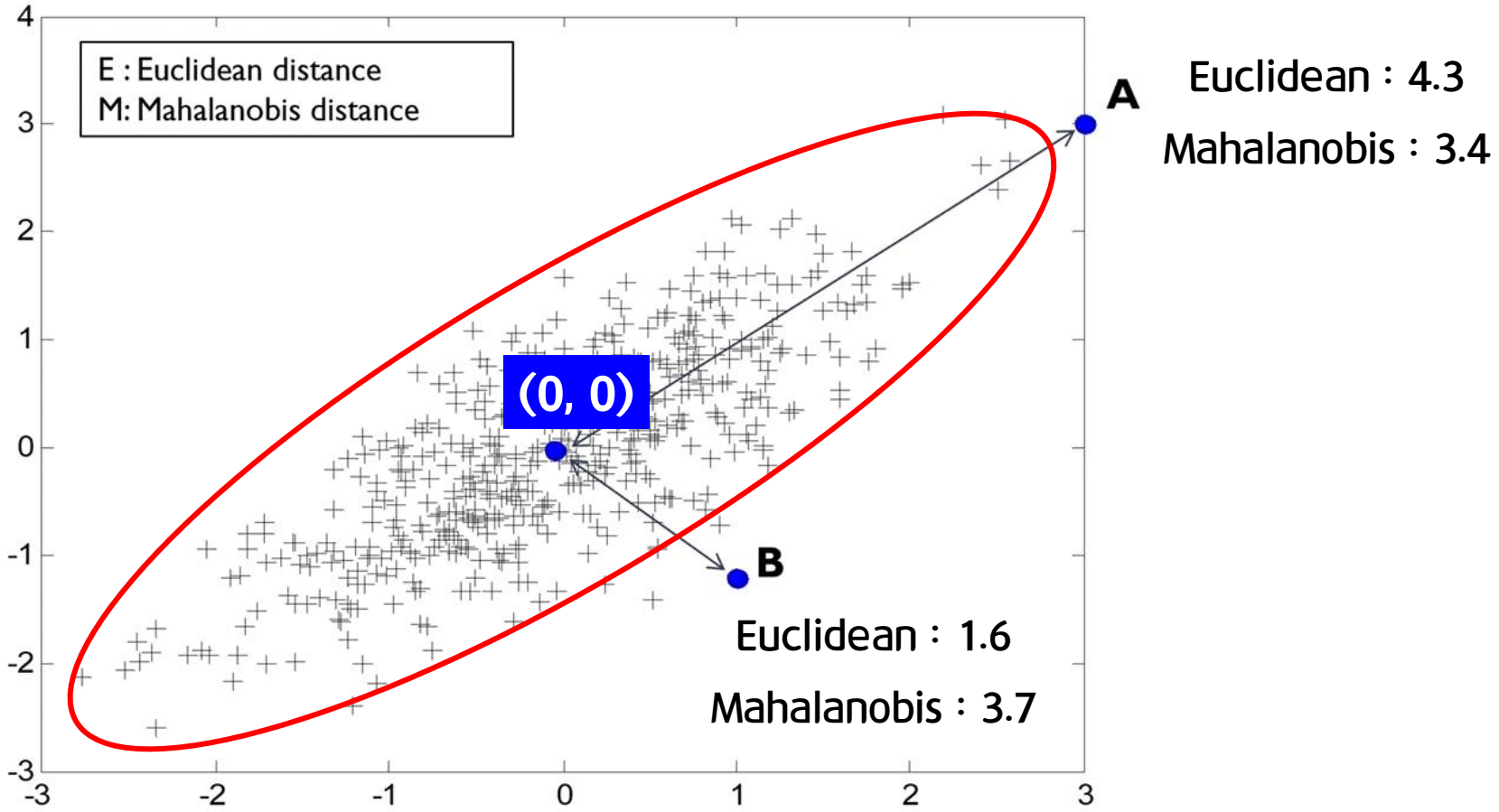
Equation of Ellipse

$$\frac{1}{2}x_1^2 + 2x_2^2 - \sqrt{2}x_1x_2 = c^2$$



Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 3 : **Mahalanobis Distance**



Unit 02 | Hyperparameter 1 – Distance measure

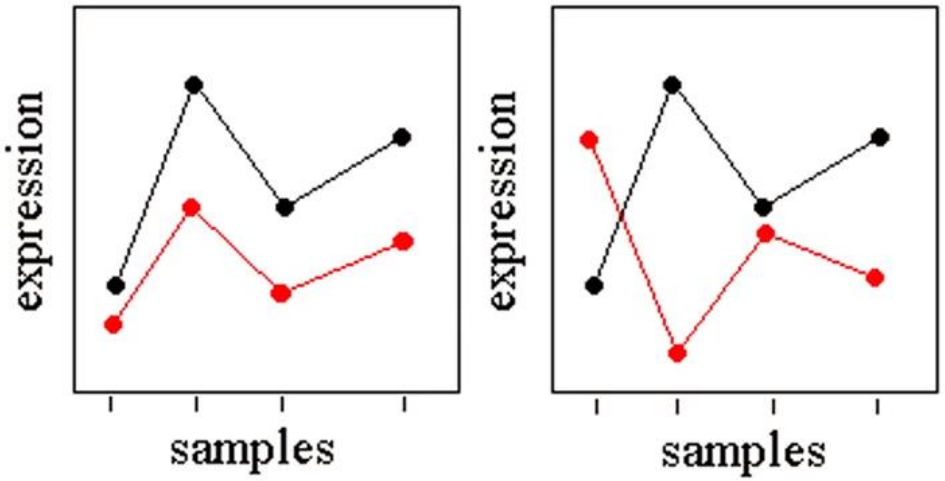
Distance Measure 4 : **Correlation Distance**

$$d_{Corr(X,Y)} = 1 - r$$

$$where\ r = \sigma_{XY}$$

$$-1 \leq r \leq 1$$

$$0 \leq d_{Corr(X,Y)} \leq 2$$



- 데이터 간 Pearson correlation을 거리 측도로 사용하는 방식
- 데이터 패턴의 유사도를 반영할 수 있음

Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 5 : **Spearman Rank Correlation Distance**

$$d_{Spearman(X,Y)} = 1 - \rho,$$

$$where\ \rho = 1 - \frac{6\sum_{i=1}^{n}(rank(x_i) - rank(y_i))^2}{n(n^2-1)}$$

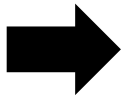
- p를 Spearman correlation이라 함.
- 데이터의 rank를 이용하여 correlation의 distance를 계산하는 방식
- p의 범위는 -1에서 1로 Pearson correlation과 동일

Unit 02 | Hyperparameter 1 – Distance measure

Distance Measure 5 : **Spearman Rank Correlation Distance**

계절 평균 낮 최고 기온

지역	봄	여름	가을	겨울
서울	17.06	28.43	19.07	3.50
뉴욕	16.32	28.22	18.37	5.43
시드니	22.23	17.03	21.90	25.63



지역 별 계절 기온 순위

지역	봄	여름	가을	겨울
서울	3	1	2	4
뉴욕	3	1	2	4
시드니	2	4	3	1

서울 – 뉴욕 간
Spearman rank correlation distance

$$\rho = 1 - \frac{6\{(3-3)^2 + (1-1)^2 + (2-2)^2 + (4-4)^2\}}{4(4^2-1)} = 1 \rightarrow d_{(서울,뉴욕)} = 1 - 1 = 0$$

서울 – 시드니 간
Spearman rank correlation distance

$$\rho = 1 - \frac{6\{(3-2)^2 + (1-4)^2 + (2-3)^2 + (4-1)^2\}}{4(4^2-1)} = -1 \rightarrow d_{(서울,시드니)} = 1 - (-1) = 2$$

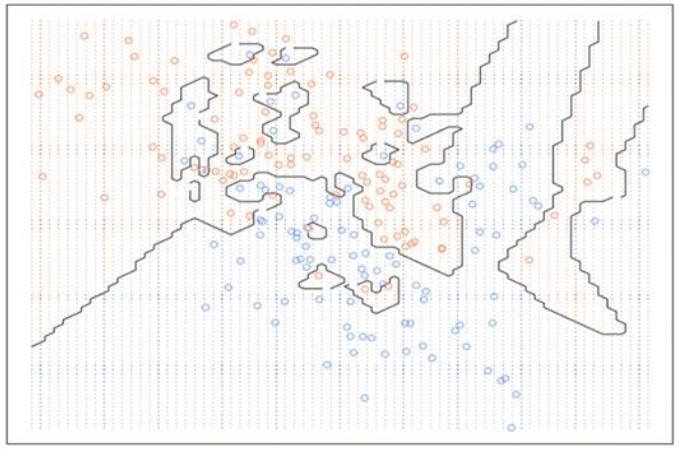
Unit 02 | Hyperparameter 1 – Distance measure

	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)

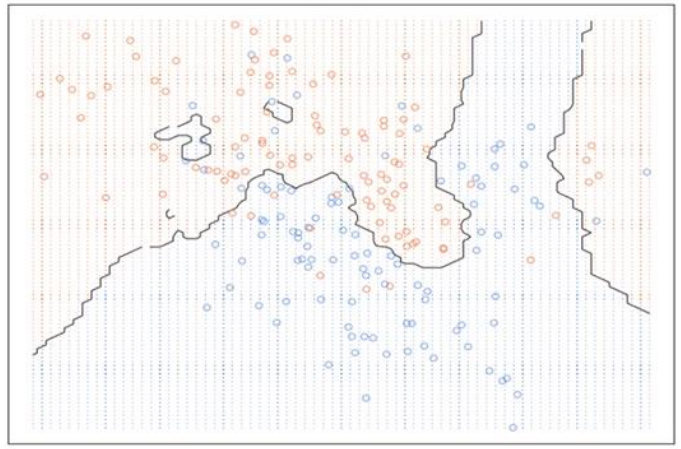
TABLE II. WEIGHTED KNN CLASSIFICATION RESULTS

Distance Measure에 따른 성능 차이 존재

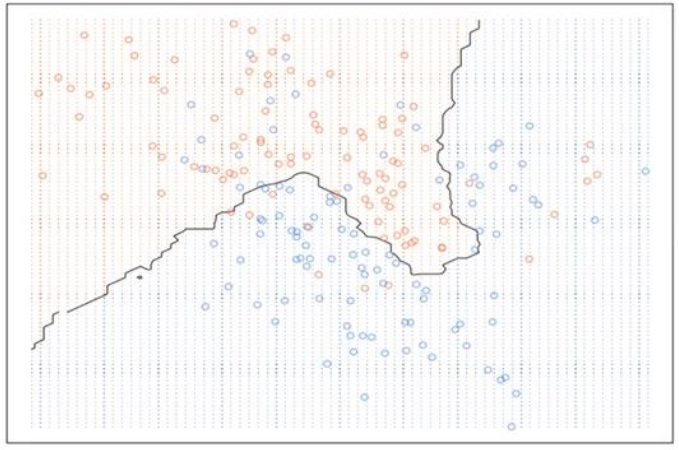
Unit 03 | Hyperparameter 2 – K



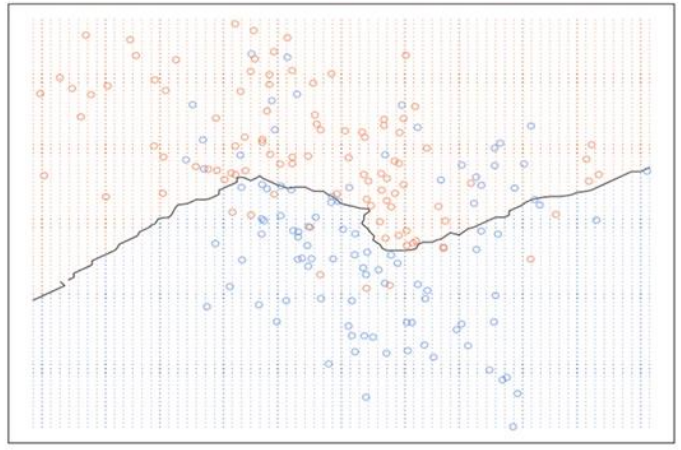
1-nearest neighbor



5-nearest neighbor



15-nearest neighbor



50-nearest neighbor

Unit 03 | Hyperparameter 2 – K

보통 K는 홀수로 지정

→ 짝수의 경우 동률이 발생할 수 있기 때문

K = 1

K가 작을 때

Overfitting

적절한 K를 찾는 게 중요 !

K = n

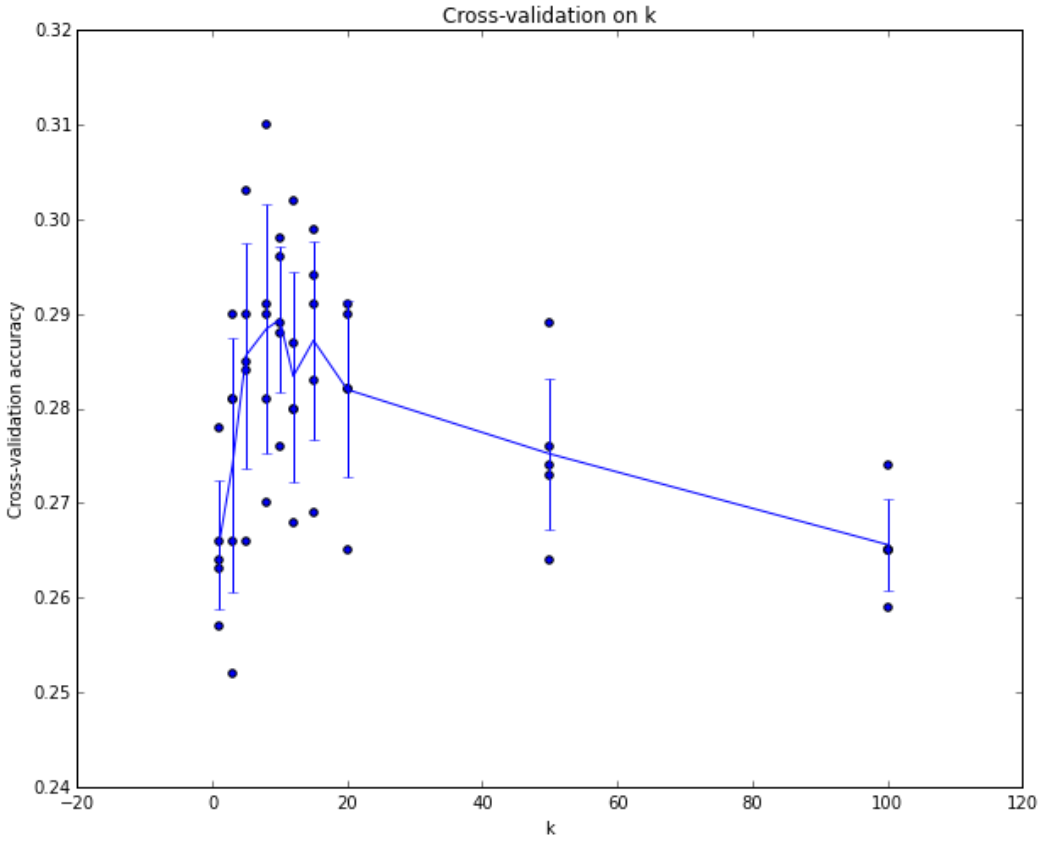
K가 클 때

Underfitting

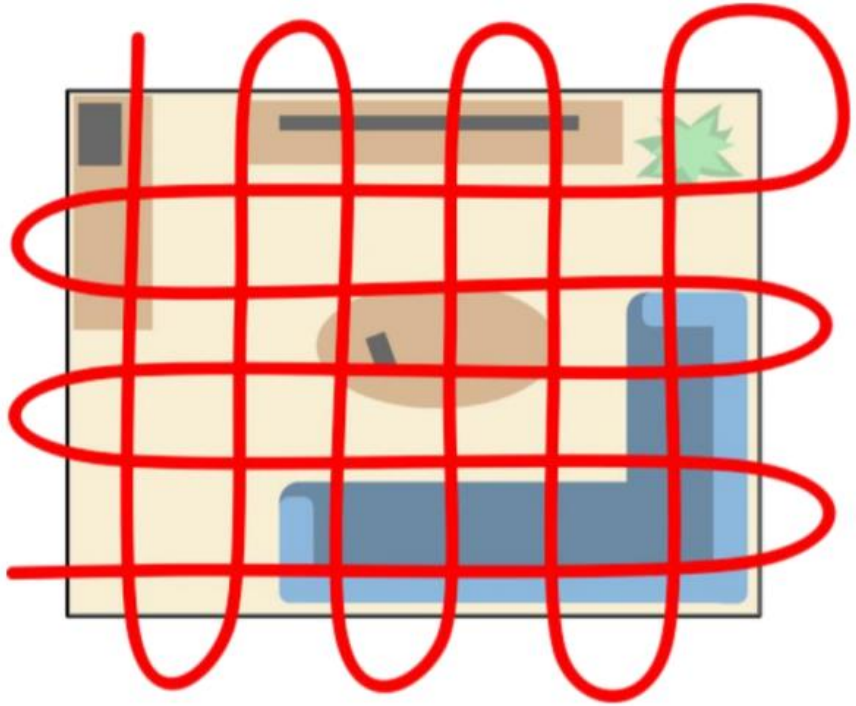
Unit 03 | Hyperparameter 2 – K

Validation set의 정확도를 보고 적절한 k를 선택하는 경우가 많음. * KNN 뿐만 아니라 ML에서 Hyperparameter를 조정할 때 많이 쓰이는 방법

K-fold cross-validation
(5-fold cross-validation)



그리드 서치 (Grid Search)



격자(Grid)무늬로 Hyperparameter를 탐색(Search)

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

- 장점 : 주어진 공간 내에서 가장 좋은 결과를 얻을 수 있다
- 단점 : 시간이 정말 오래 걸린다

Unit 04 | KNN 고려사항

1. Distance 기반 알고리즘

- 변수들의 단위(scale)에 민감 **Feature Scaling**
- categorical은? **One-hot encoding**

2. Majority Voting, average 보다 더 좋은 방법? **Weighted KNN**

Unit 04 | KNN 고려사항

1. Feature Scaling

	X1	X2 ($)
A	1	5
B	2	6
c	4	4

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

= 3.162278

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

= **2.828427**

	X1	X2 (₩)
A	1	5000
B	2	6000
c	4	4000

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

= **1000.004**

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

= 2000.001

Unit 04 | KNN 고려사항

1. Feature Scaling

min-max normalization

- 0 ~ 1 사이의 값을 가짐.

$$X = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Z-score normalization

- 통계에서 자주 보는 방법

$$X = \frac{x - x_{mean}}{x_{std}}$$

https://datascienceschool.net/view-notebook/f43be7d6515b48c0beb909826993c856/

Unit 04 | KNN 고려사항

2. One-hot encoding : 단 하나의 값만 True이고 나머지는 모두 False인 인코딩

즉, 1개만 Hot(True)이고 나머지는 Cold(False)

- KNN은 거리 기반 → Input에 numerical 와야 함.
- Categorical은 numerical로 바꿔줘야 함. 더미변수 만들기 (one-hot encoding)

	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									

Unit 04 | KNN 고려사항

3. Weighted KNN

- Example 1 : Problem with majority voting (if k=4)

Customer	Age	Income (1000s)	Cards	Response (target)	Distance from David	
David	*37*	*50*	*2*	*?*	0	
John	35	35	3	Yes	$\sqrt{(35-37)^2+(35-50)^2+(3-2)^2}=15.16$	2nd
Rachael	22	50	2	No	$\sqrt{(22-37)^2+(50-50)^2+(2-2)^2}=15$	1st
Ruth	63	200	1	No	$\sqrt{(63-37)^2+(200-50)^2+(1-2)^2}=152.23$	
Jefferson	59	170	1	No	$\sqrt{(59-37)^2+(170-50)^2+(1-2)^2}=122$	4th
Norah	25	40	4	Yes	$\sqrt{(25-37)^2+(40-50)^2+(4-2)^2}=15.74$	3rd

Unit 04 | KNN 고려사항

3. Weighted KNN

- 단순히 평균, 다수결로 값을 결정하지 않고 **거리에 따라서 영향력을 달리 주고 싶을 때** 사용
- Weighted average or Weighted voting

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

Unit 04 | KNN 고려사항

3. Weighted KNN

$$\text{유사도} = \frac{1}{\text{거리}}$$

$$\text{가중치} = \frac{\text{유사도}}{\text{모든 유사도의 합}}$$

- Example 2 : Regression → Weighted Average

  K=4 일 때, new data의 체지방률?

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

① KNN

(15.4 + 17.2 + 12.3 + 11.5) / 4

= 14.1

② Weighted KNN

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

= 14.868

Unit 04 | KNN 고려사항

3. Weighted KNN

$$\text{유사도} = \frac{1}{\text{거리}}$$

$$\text{가중치} = \frac{\text{유사도}}{\text{모든 유사도의 합}}$$

- Example 3 : Classification → Weighted Voting

  K=4 일 때, new data의 클래스?

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

① KNN

A : 1, B : 2, C : 1

→ **B로 분류**

② Weighted KNN

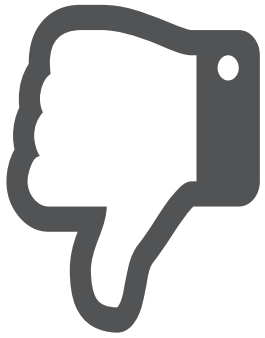
A : 1 * 0.48, B : 1 * 0.24 + 1 * 0.12, C : 1 * 0.16

→ **A로 분류**

Unit 05 | KNN 장·단점

1. 학습 과정이 없다
2. 직관적인 이해 용이
3. 데이터가 충분히 많으면 좋은 성능을 보인다

Unit 05 | KNN 장·단점



1. 메모리가 많이 필요하다
2. 지나치게 기존 사례에 의존적
3. 데이터가 많을 수록 예측/분류 시간이 길어진다
4. 회귀의 경우, 주변부에서 왜곡 현상이 나타날 수 있다
   - 기존 사례의 최대치를 넘어서는 새로운 사례 : 과소하게 예측
   - 기존 사례의 최소치를 밑도는 새로운 사례 : 과다하게 예측

과 제

KNN classifier 구현 : Iris_knn_Assignment_example.ipynb 참고

1) Preprocessing / EDA

2) KNN classifier

- K
- Distance Measure : Euclidean? Manhattan?
- Weighted voting? Majority voting?

3) Evaluation

https://www.kaggle.com/uciml/iris

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

참고자료

- 투빅스 10기 황이은 강의자료
- 투빅스 9기 최영제 강의자료
- https://www.youtube.com/watch?v=W-DNu8nardo
- https://ratsgo.github.io/machine%20learning/2017/04/17/KNN/
- Foster Provost and Tom Fawcett, "Data Science for Business," O'Reilly, 2013.
- http://aikorea.org/cs231n/classification/

들어주셔서 감사합니다.