강의계획표

주	해당 장	주제
1	1장 2장, 3장	머신러닝이란
2		머신러닝을 위한 기초지식, 구현을 위한 도구
3	4장	선형 회귀로 이해하는 지도학습
4	5장	분류와 군집화로 이해하는 지도 학습과 비지도 학습
5	6장	다양한 머신러닝 기법들 다항 회귀, 결정 트리, SVM
6		
7	7장	인공 신경망 기초 - 문제와 돌파구
8		중간고사
9	8장	고급 인공 신경망 구현
10	9장	신경망 부흥의 시작, 합성곱 신경망
11	10장	순환 신경망
12	11장	차원축소와 매니폴드 학습
13	12장	오토인코더와 잠재표현 학습
14	13장	인공지능의 현재와 미래
15		보강주
16		기말고사

5장 지도학습-분류
비지도학습-군집화

- K-NN을 이용한 분류
- 성능 평가지표
- K-means를 이용한 군집화
- Scikit-learn 라이브러리를 이용한 프로그램

1. K-NN을 이용한 분류 (1)

❖ 분류classification :

- 소속 집단의 정보를 이미 알고 있는 상태에서 동일 집단으로 묶는 방법
- 예) 닥스훈트와 사모예드 분류: 몸통의 길이와 높이로 구분

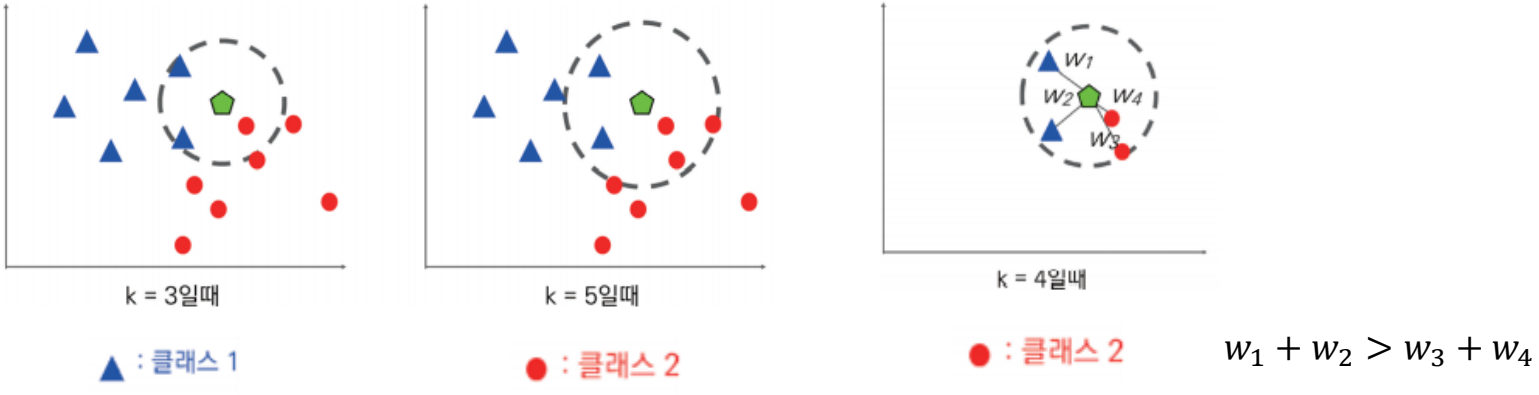
❖ 군집화clustering :

- 소속 집단의 정보가 없는 상태에서 비슷한 집단으로 묶는 방법

1. K-NN을 이용한 분류 (2)

❖ k-최근접 이웃k-Nearest Neighbor :

- 특징 공간에 분포하는 데이터에 대하여 k개의 가장 가까운 이웃을 살펴보고 다수결 방식으로 데이터의 레이블을 할당 하는 분류방식
- K: 홀수
- K: 짝수



$w_1 + w_2 > w_3 + w_4$

- 특징 공간에 있는 모든 데이터에 대한 정보가 필요
  - 단점: 데이터 인스턴스, 클래스, 특징의 요소들의 개수가 많다면, 많은 메모리 공간과 계산 시간이 필요
  - 거리 계산: Euclidian distance
  - **장점: 사전 학습이나 특별한 준비 시간이 필요 없다**

1. K-NN을 이용한 분류 (3)

- Scikit-learn: KNeighborsClassifier()
- 프로그램1 : 닥스훈트와 사모예드 분류

닥스훈트 8마리의 길이와 높이								
길이	77	78	85	83	73	77	73	80
높이	25	28	29	30	21	22	17	35
사모예드 8마리의 길이와 높이								
길이	75	77	86	86	79	83	83	88
높이	56	57	50	53	60	53	49	61

- 프로그램2 : iris(붓꽃) 분류

- https://colab.research.google.com/drive/1XQe7_FCAHIbnig44Lplew2Bildufn7oF

2. 성능 측정을 위한 평가지표(1)

❖ 분류기의 성능 평가

- 정확도accuracy = (**옳게 분류한 데이터 개수**)/(**전체 데이터 개수**)
- 표집편향sampling bias
- 예) 10,000개의 sample 중 class1 9,900개, class2가 100개이면, 무조건 class1로 분류해도 99% 정확도

❖ 혼동행렬confusion matrix (오차행렬)

- 이진 분류의 정답 class에 대해서 분류기가 얼마나 정확하게 예측했는지는 나타내는 행렬

표 4.2 이진 분류기의 혼동행렬

		예 측	
		양성	음성
실제	양성	진양성(True Positive) *TP*	위음성(False Negative) *FN*
	음성	위양성(False Positive) *FP*	진음성(True Negative) *TN*

- 음성, 양성은 분류 분야에 따라 정의됨(예, 암진단, 불량품검출)

2. 성능 측정을 위한 평가지표(2)

- 민감도(sensitivity), 재현율(recall), 진양성률

$$\text{민감도} = \frac{TP}{TP+FN}$$

- 특이도(specificity), 진음성율(true negative rate)

$$\text{특이도} = \frac{TN}{FP+TN}$$

- 정밀도(precision) $\quad \text{정밀도} = \frac{TP}{TP+FP}$
- 음성 예측도 $\quad \text{음성 예측도} = \frac{TN}{TN+FN}$
- 위양성율 $\quad \text{위양성율} = \frac{FP}{FP+TN} = 1 - \text{특이도}$
- 위발견율 $\quad \text{위발견율} = \frac{FP}{TP+FP} = 1 - \text{정밀도}$
- 정확도(accuracy) $\quad \text{정확도} = \frac{TP+TN}{TP+FP+TN+FN}$
- F1 측도 $\quad F1 = 2\frac{(\text{정밀도}) \cdot (\text{재현율})}{(\text{정밀도}) + (\text{재현율})}$ $\quad F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{FN+FP}{2}}$

표 4.2 이진 분류기의 혼동행렬

		예 측	
		양성	음성
실제	양성	진양성(True Positive) TP	위음성(False Negative) FN
	음성	위양성(False Positive) FP	진음성(True Negative) TN

- 의사의 환자 진료: 특이도, 민감도
  - 특이도= $\frac{\text{정상예측}(TN)}{\text{실제정상}(FP+TN)}$
  - 민감도= $\frac{\text{환자예측}(TP)}{\text{실제환자}(TP+FN)}$
- 정보검색, 물체검출: 정밀도, 재현률
  - 정밀도= $\frac{\text{긍정검색}(TP)}{\text{전체(긍정)검색결과}(TP+FP)}$
  - 재현율= $\frac{\text{긍정검색}(TP)}{\text{실제데이터}(TP+FN)}$

2. 성능 측정을 위한 평가지표(3)

❖ 혼동행렬 예1

		KoKIT22의 예측값 (검사결과)					
		음성			양성		
환자의 실제 상태값		N			P		
음성 (COVID 안걸림)	N	89 TN	T 일치	N 예측	11 FP	F 불일치	P 예측
양성 (COVID 걸림)	P	5 FN	F 불일치	N 예측	95 TP	T 일치	P 예측

- 정확률 $Acc = \frac{TP+TN}{FP+FN+TP+TN} = \frac{95+89}{11+5+95+89} = 0.92$

- 재현율recall, 진양비율True Positive Rate:TPR, 민감도sensitivity

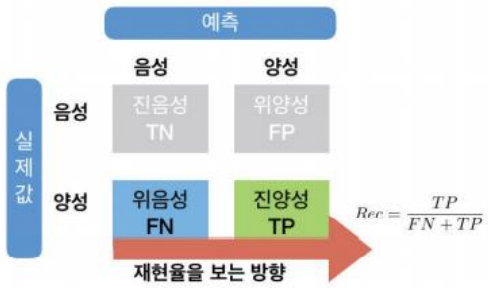
$$TPR = Rec = \frac{TP}{P} = \frac{TP}{FN+TP} = \frac{95}{100} = 0.95$$

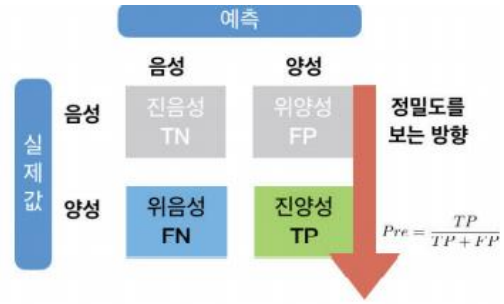
- 정밀도precision

$$Pre = \frac{TP}{TP+FP} = \frac{95}{106} = 0.896$$

- 조화 평균harmonic mean, 다이스 유사도 계수Dice similarity coe

$$F_1 = \frac{2}{\frac{1}{Pre} + \frac{1}{Rec}} = 2\frac{Pre \times Rec}{Pre + Rec}$$





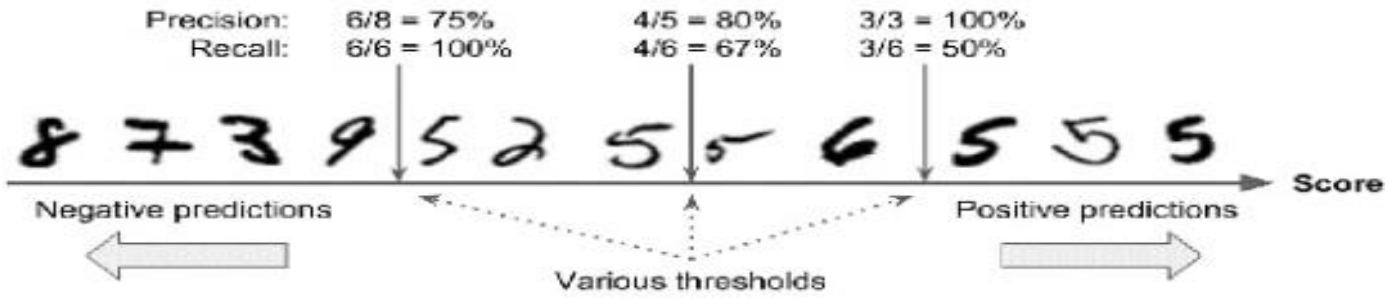
2. 성능 측정을 위한 평가지표(4)

❖ 혼동행렬 예2

- True labels = [2, 0, 0, 2, 4, 4, 1, 0, 3, 3, 3]
- Predict labels = [2, 1, 0, 2, 4, 3, 1, 0, 1, 3, 3]

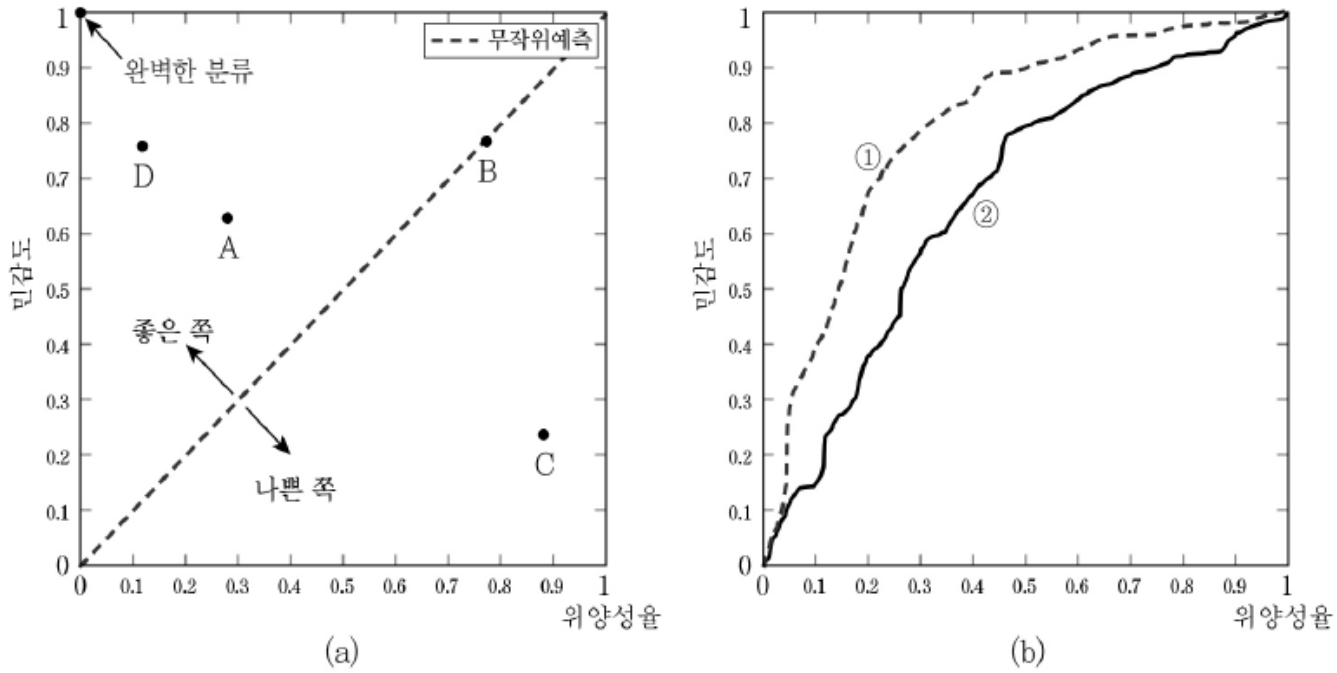
Class	True	predict	precision	recall	F1
0	3	2	2/2=1.00	2/3=0.67	0.80
1	1	3	1/3=0.33	1/1=1.00	0.50
2	2	2	2/2=1.00	2/2=1.00	1.00
3	3	3	2/3=0.67	2/3-0.67	0.67
4	2	1	1/1=1.00	1/2=0.50	0.67

❖ '5'에 대한 평가 결과 (정확률 vs. 재현률의 tradeoff)



2. 성능 측정을 위한 평가지표(5)

- **ROC 곡선**
  - 부류 판정 임계값에 따른 (위양성율, 진양성률/민감도) 그래프
- **AUC**(Area Under the Curve)
  - ROC 곡선에서 곡선 아래 부분의 면적, 클수록 좋은 분류



3. 군집화 (1)

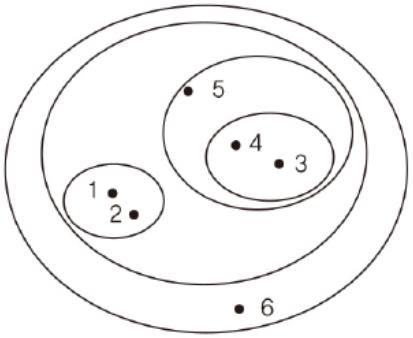
❖ **군집화(clustering) 알고리즘**
- 데이터를 유사한 것들끼리 모우는 것
- **군집 간**의 **유사도(similarity)**는 **크게**, **군집 내**의 **유사도**는 **작게**

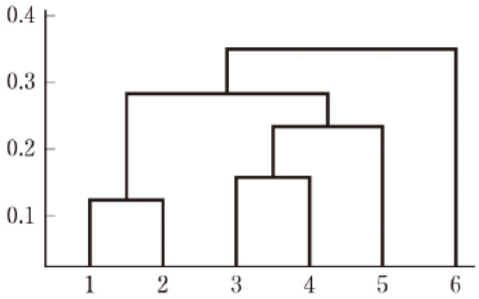
❖ **계층적 군집화 (hierarchical clustering)**
- 군집화의 결과가 군집들이 계층적인 구조를 갖도록 하는 것
- **병합형(agglomerative) 계층적 군집화**
  - 각 데이터가 하나의 군집을 구성하는 상태에서 시작하여, 가까이에 있는 군집들을 결합하는 과정을 반복하여 계층적인 군집 형성
- **분리형(divisive) 계층적 군집화**
  - 모든 데이터를 포함한 군집에서 시작하여 유사성을 바탕으로 군집을 분리하여 점차 계층적인 구조를 갖도록 구성

❖ **분할 군집화 (partitioning clustering)**
- 계층적 구조를 만들지 않고 전체 데이터를 유사한 것들끼리 나누어서 묶는 것
- 예. k-means **알고리즘**

3. 군집화 (2)

❖ 계층적 군집화와 덴드로그램(dendrogram)



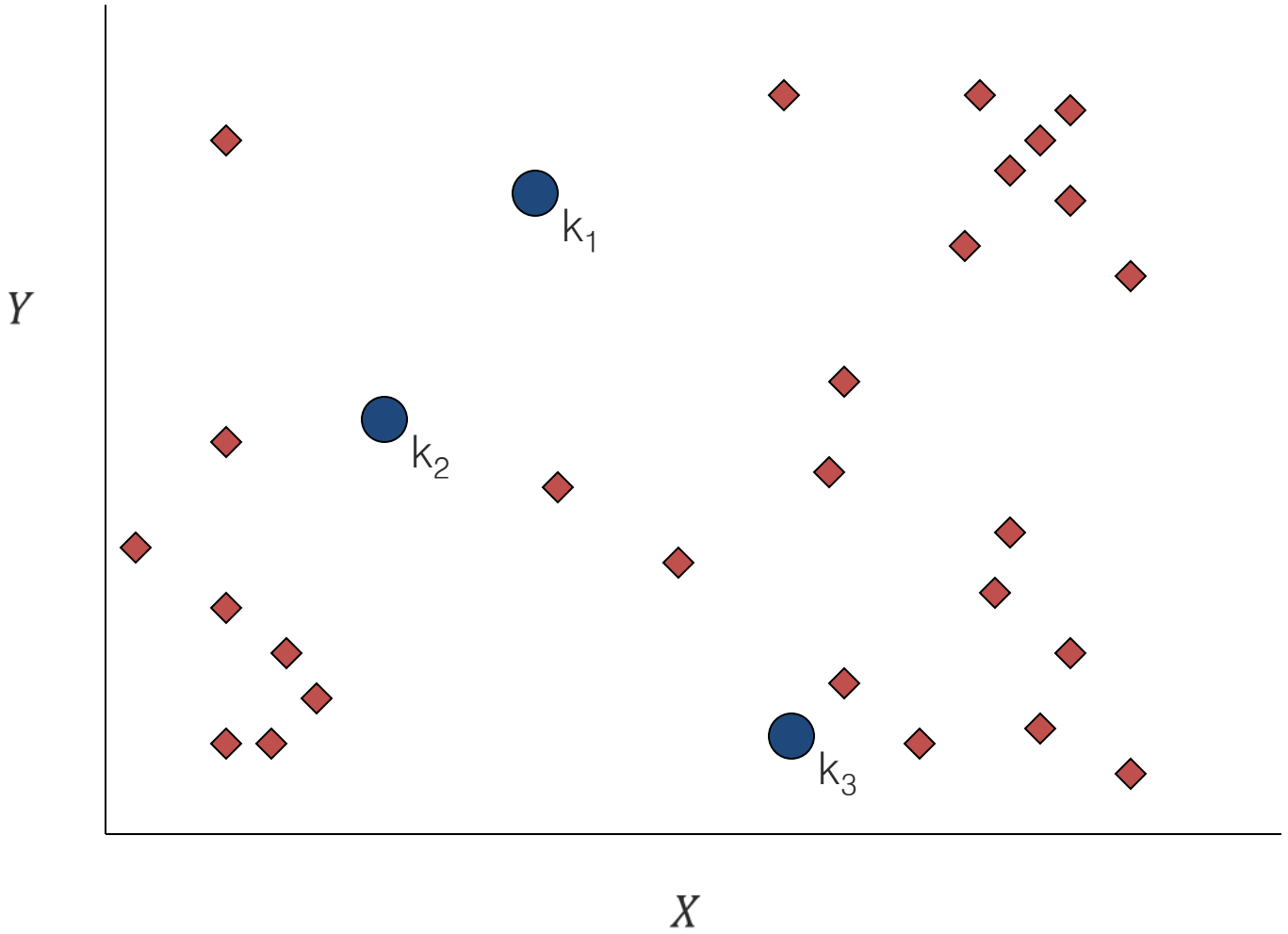


❖ 분할 군집화

- **k-means 알고리즘**
- **군집화 과정**
  1. **군집**의 **중심 위치** 선정
  2. 군집 중심을 기준으로 **군집 재구성**
  3. 군집별 **평균 위치** 결정
  4. 군집 평균 위치로 **군집 중심 조정**
  5. 수렴할 때까지 2-4 과정 **반복**

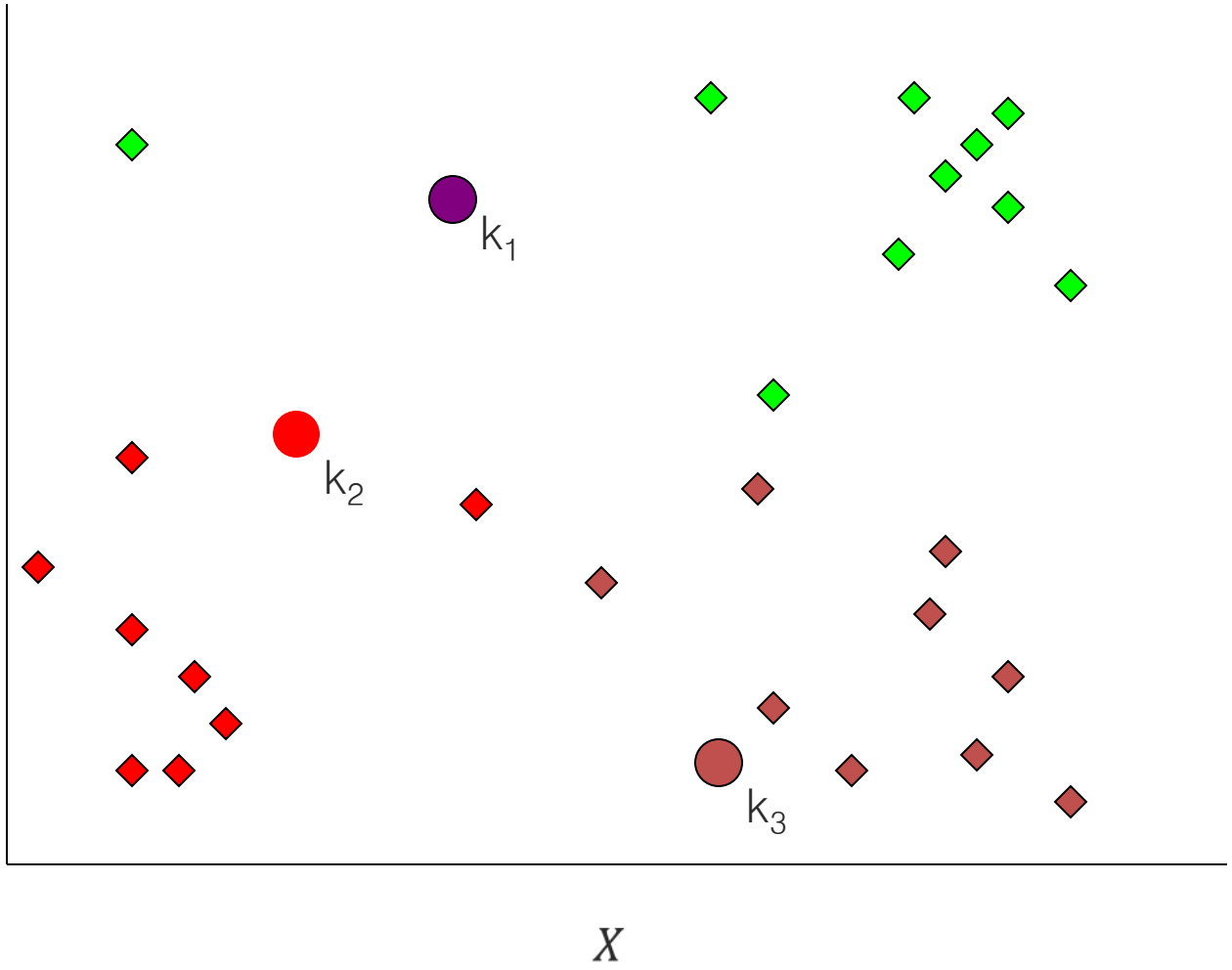
K-means Algorithm (1)

무작위로 군집 중심위치 3개를 선택



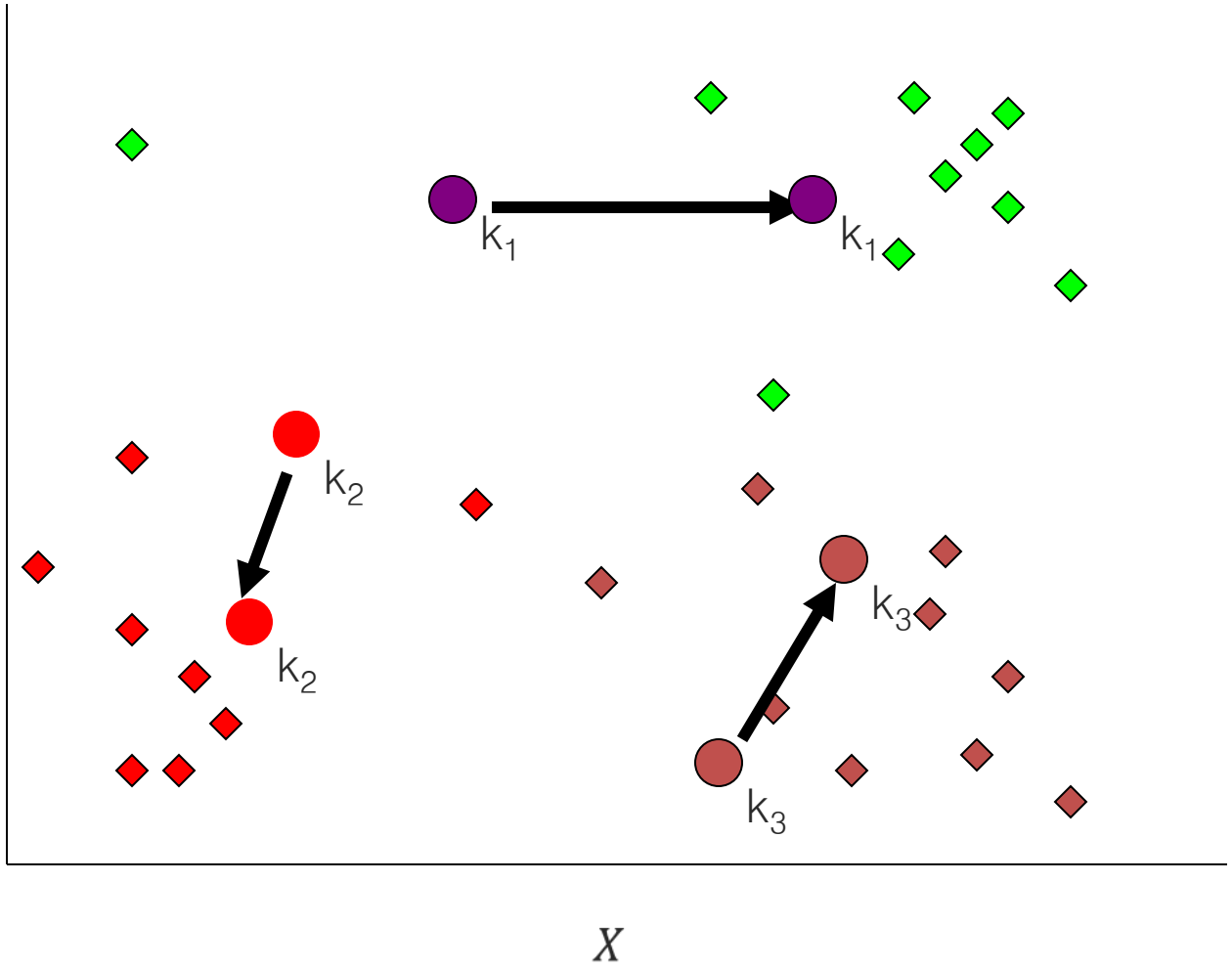
K-means Algorithm (2)

각 점을 최근접 군집 중심위치에 할당



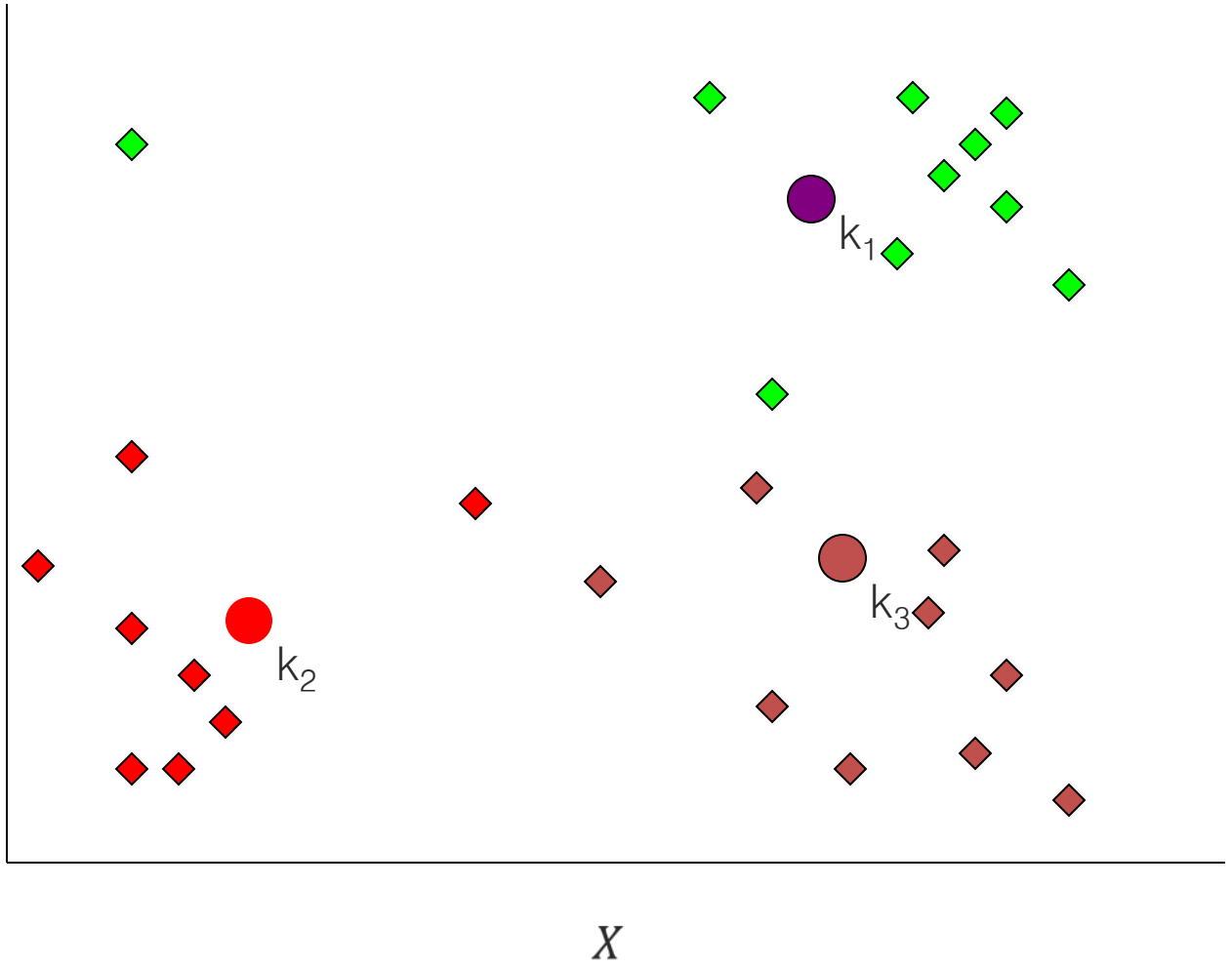
K-means Algorithm (3)

군집 중심 위치를 군집의 평균 위치로 이동

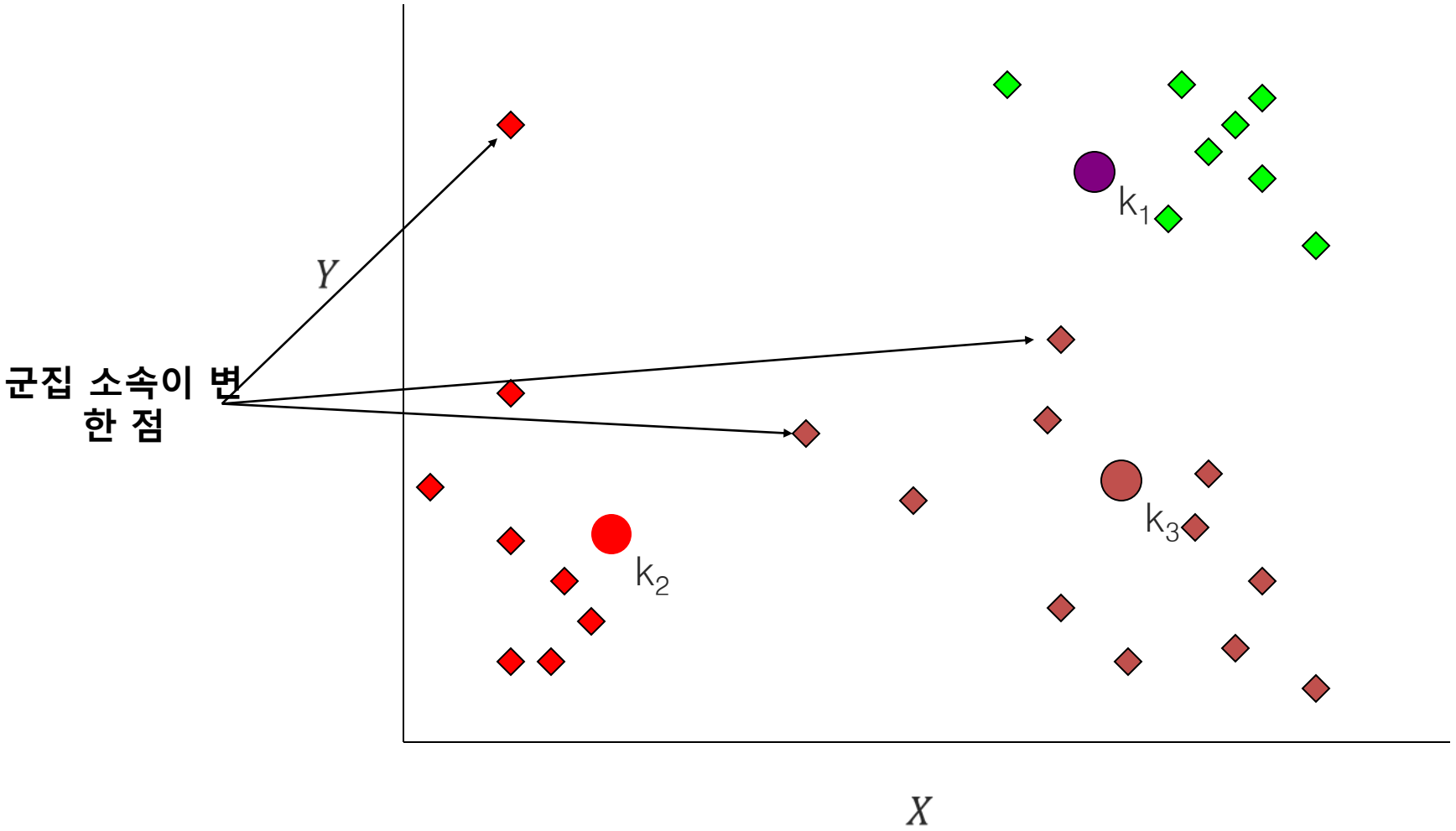


K-means Algorithm (4)

새로운 군집 중심을 기준으로 각 점의 소속을 재할당

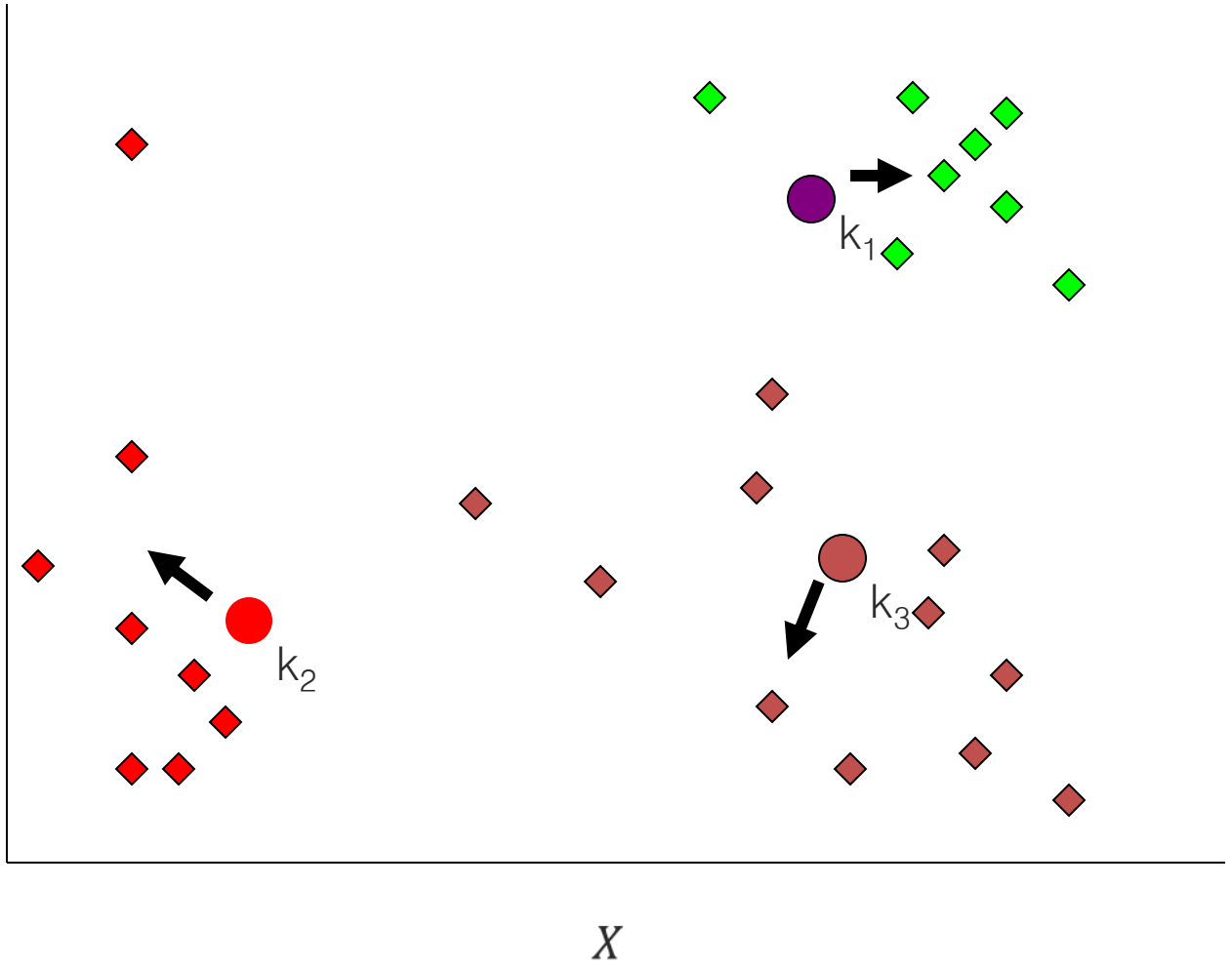


K-means Algorithm (5)



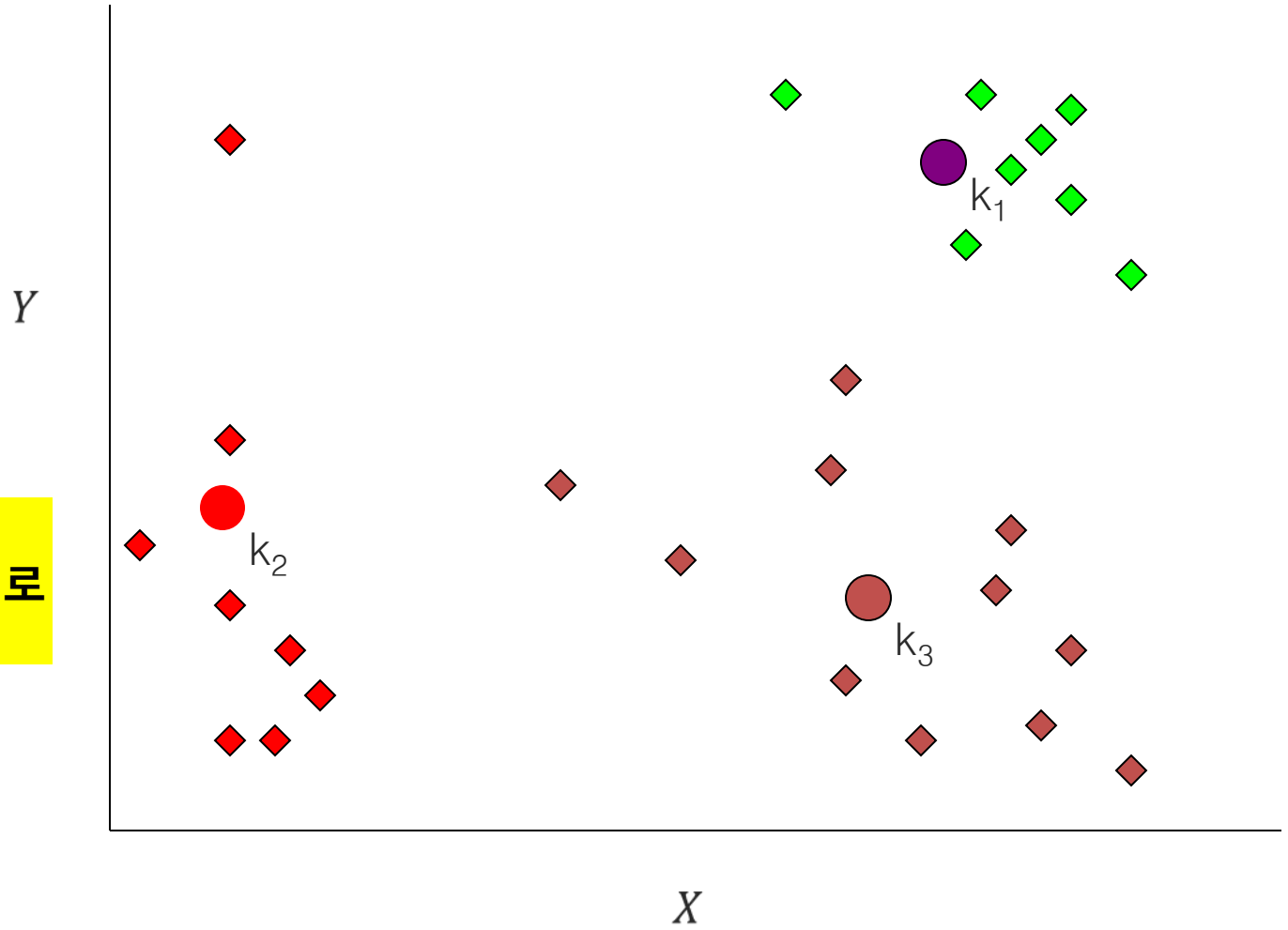
K-means Algorithm (6)

군집 평균 재계산



K-means Algorithm (7)

군집 중심을 군집 평균위치로 변경



K-means Algorithm (8)

❖ **k-means 알고리즘**

- i 번째 클러스터의 **중심**을 μ_i, 클러스터에 속하는 **점의 집합** S_i을 라고 할 때, **전체 분산**

$$V = \sum_{i=1}^{k} \sum_{j \in S_i} |x_j - \mu_i|^2$$

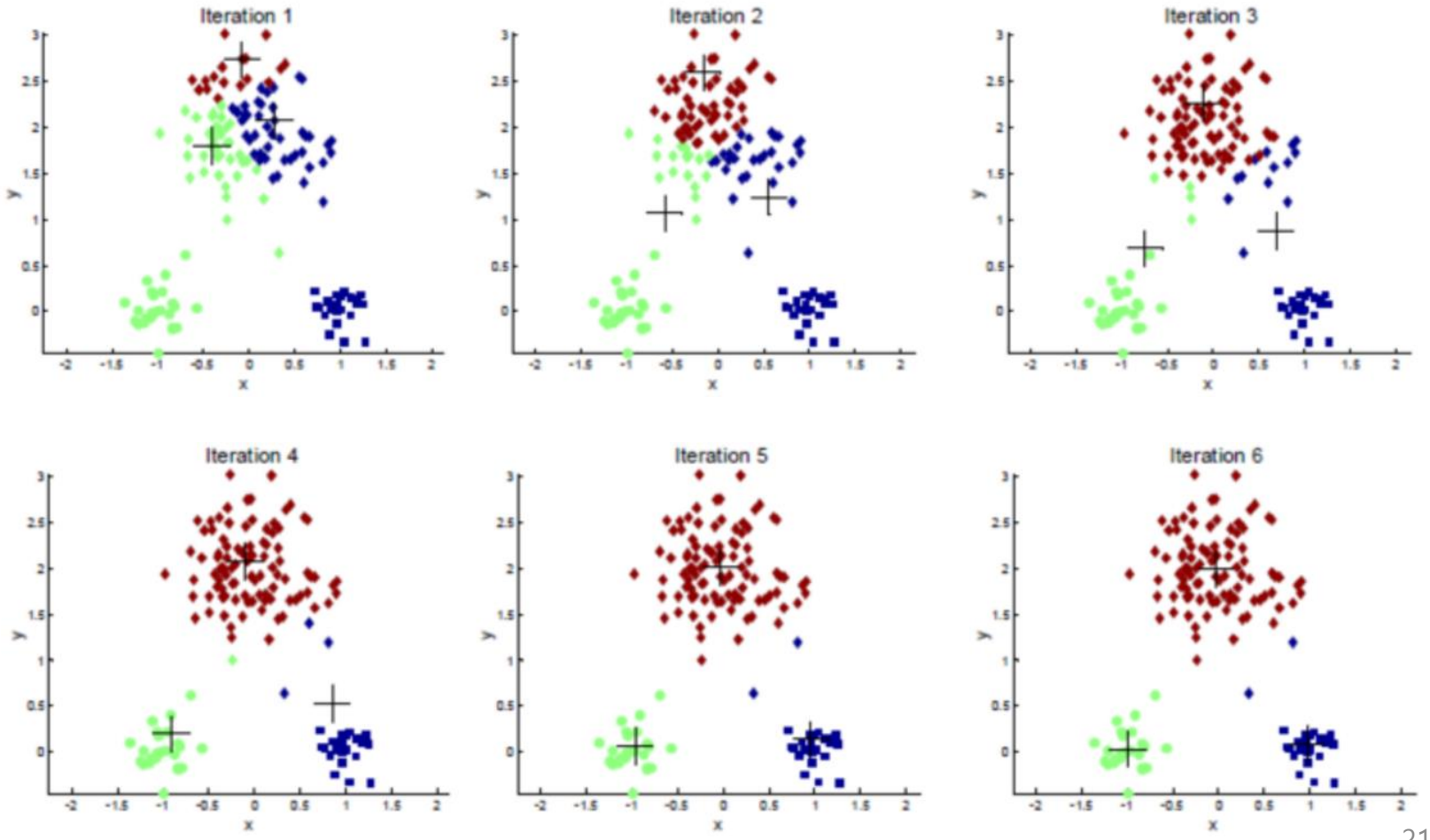
- **분산값 V**을 **최소화**하는 S_i를 찾는 것이 알고리즘의 목표

- **과정**
  1. 우선 초기의 μ_i 를 임의로 설정
  2. 다음 두 단계를 클러스터가 변하지 않을 때까지 반복
     - I. **클러스터 설정**: 각 점에 대해, 그 점에서 가장 가까운 클러스터를 찾아 배당한다.
     - II. **클러스터 중심 재조정**: μ_i를 각 클러스터에 있는 점들의 평균값으로 재설정해준다.

- **특성**
  - **군집의 개수** k는 **미리 지정**
  - **초기 군집 위치**에 **민감**

K-means Algorithm (9)

❖ 초기 중심값에 대해 민감한 군집화 결과

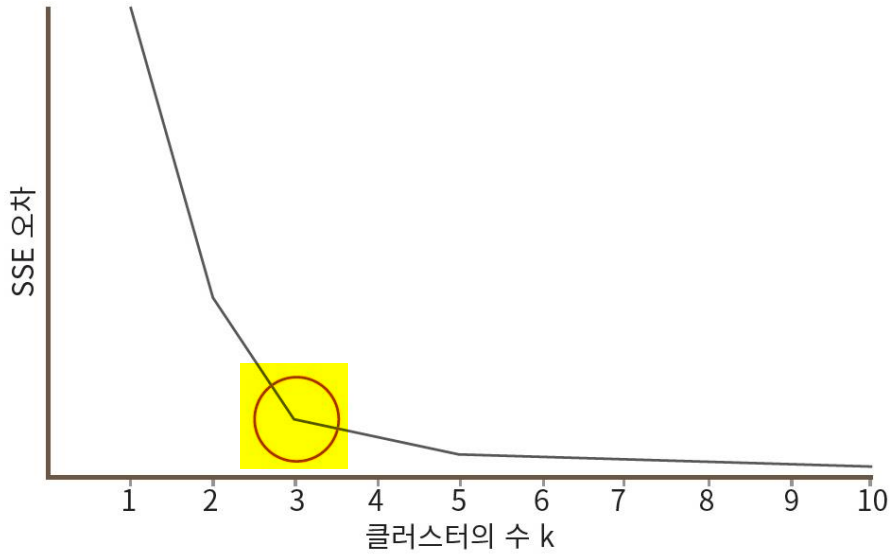


https://sites.google.com/site/myecodriving/k-means-ju-lei-fen-xi

K-means Algorithm (10)

- k를 결정하는 방법
  - "팔꿈치" 방법(elbow method)에서는 k를 1부터 증가시키면서 K-means 클러스터링을 수행한다. 각 k의 값에 대하여 SSE(sum of squared errors)의 값을 계산한다.

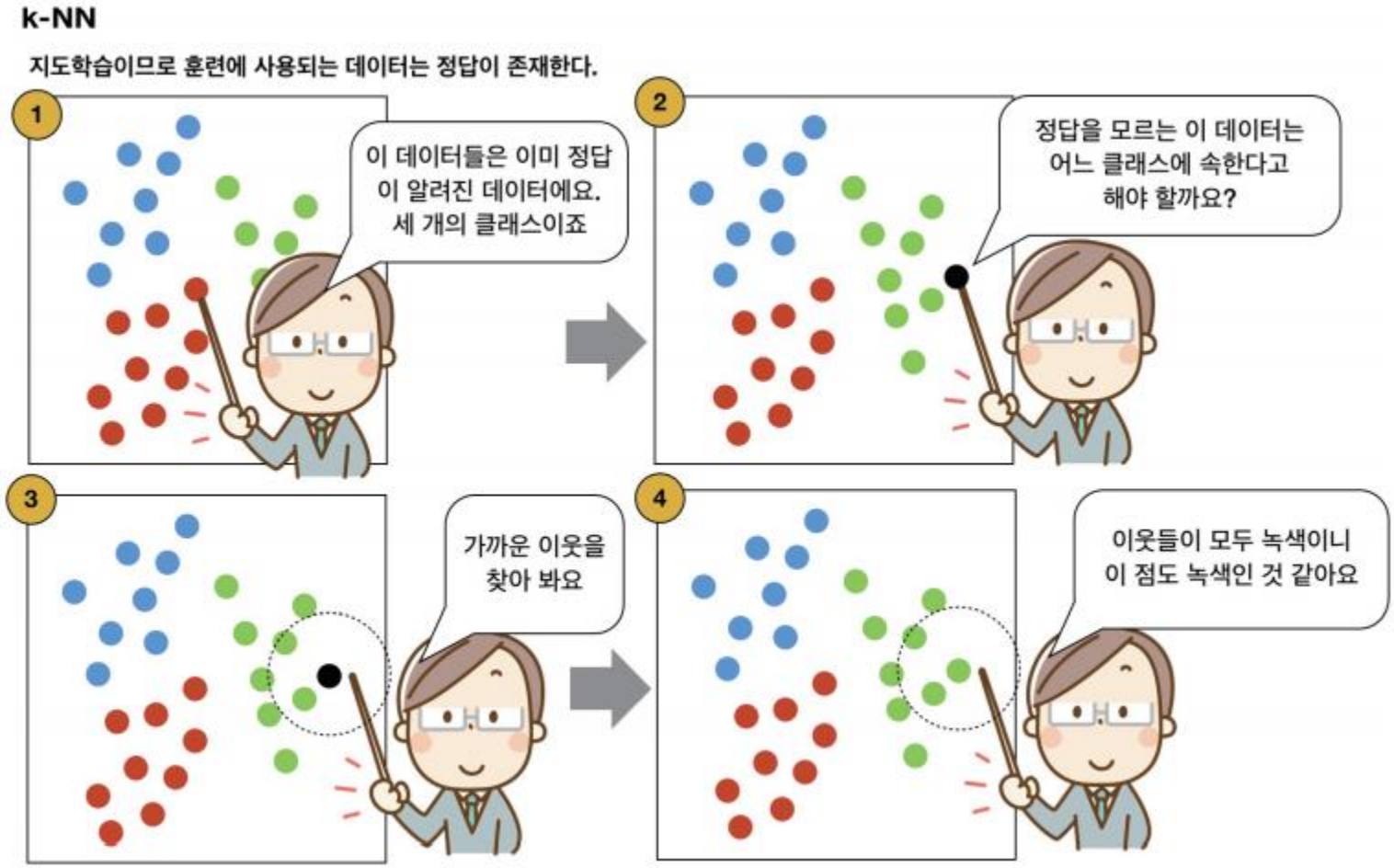
```
var sse = {};
for (var k = 1; k <= maxK; ++k) {
    sse[k] = 0;
    clusters = kmeans(dataset, k);
    clusters.forEach(function(cluster) {
        mean = clusterMean(cluster);
        cluster.forEach(function(datapoint) {
            sse[k] += Math.pow(datapoint - mean, 2);
        });
    });
}
```



https://sites.google.com/site/myecodriving/k-means-ju-lei-fen-xi

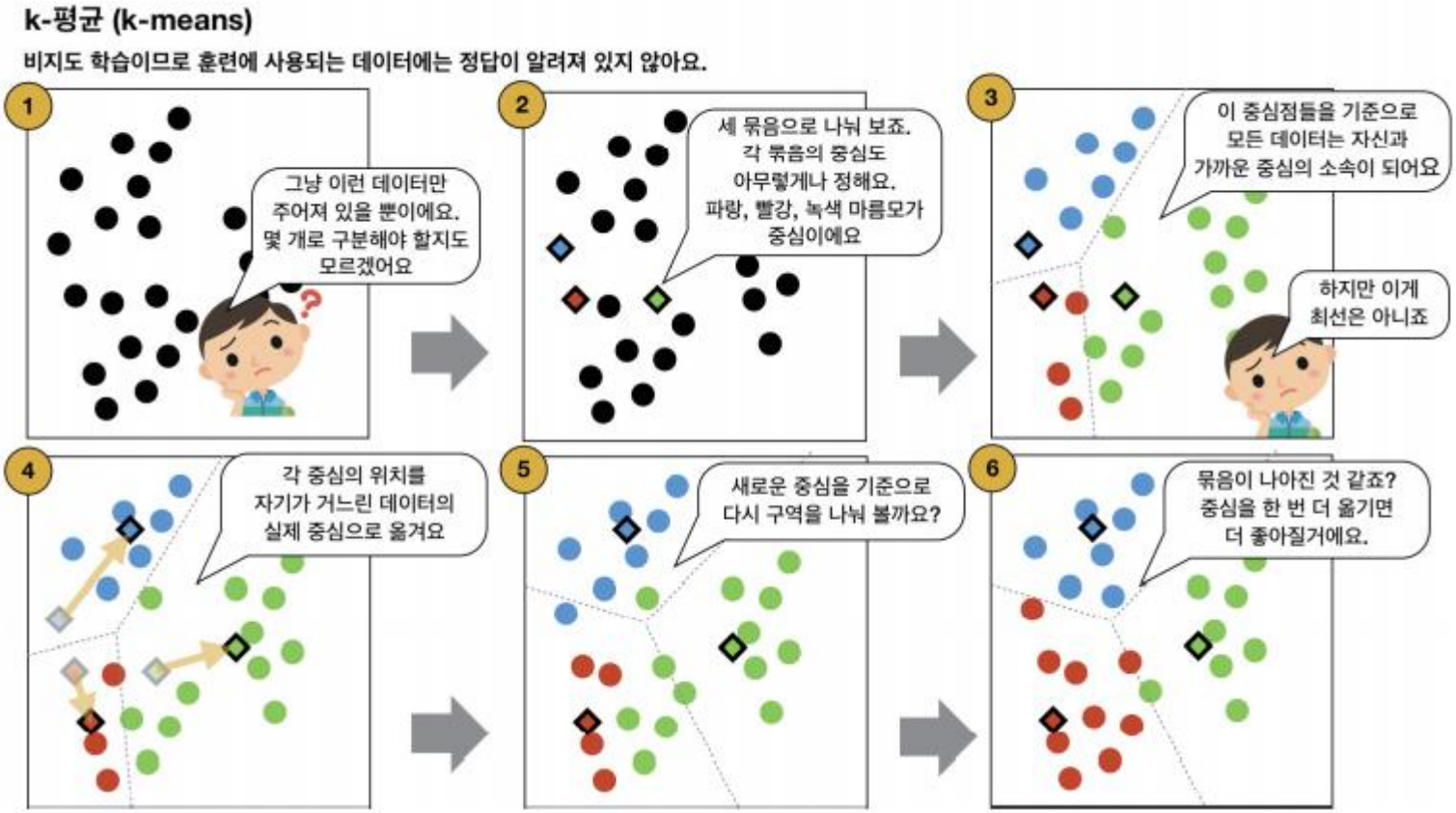
4. k-NN vs. k-means (1)

❖ **K-NN**



4. k-NN vs. k-means (2)

❖ **K-means**



5. 거리와 유사도 (1)

❖ 군집화에서의 거리 개념

- 유사도와 반대 개념
- 특징 값의 성격

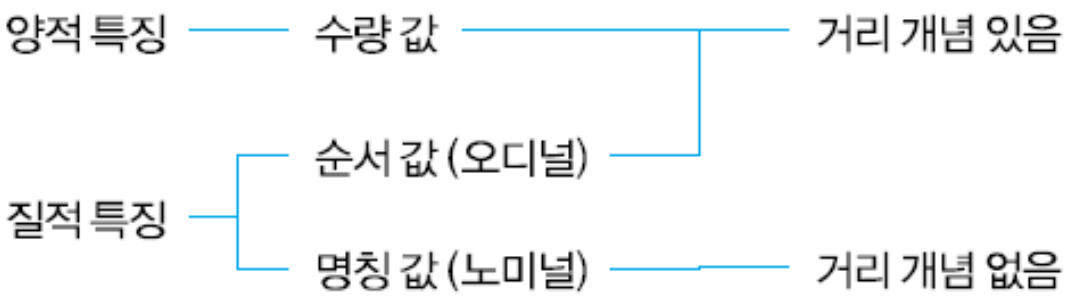


그림 10.3 특징의 유형

5. 거리와 유사도 (2)

❖ Minkowski 거리

- 두 점 $x_i=(x_{i1},\ldots,x_{id})^T$와 $x_j=(x_{j1},\ldots,x_{jd})^T$간의 거리 척도
- $d_{ij} = (\sum_{k=1}^{d} |x_{ik} - x_{jk}|^p)^{1/p}$
  - 유클리디언Euclidean 거리 (p=2) $d_{ij} = \sqrt{\sum_{k=1}^{d} |x_{ik} - x_{jk}|^2}$
  - 맨하탄Manhattan거리(p=1) $d_{ij} = \sum_{k=1}^{d} |x_{ik} - x_{jk}|$

❖ Mahalanobis 거리

- 두 점간의 거리계산에서 그들이 속한 분포를 고려
- $((x-\mu_i)^T \sum^{-1}(x-\mu_i))^{1/2}$
- 예) ω_1=(1,2)^T, (3,1)^T, (5,2)^T, (3,3)^T
  ω_2=(7,6)^T, (8,4)^T, (9,6)^T, (8,8)^T
  x = (8,2)^T
- x 에서 μ_1 유클리디언 거리: 5.0
- x 에서 μ_2 유클리디언 거리: 4.0

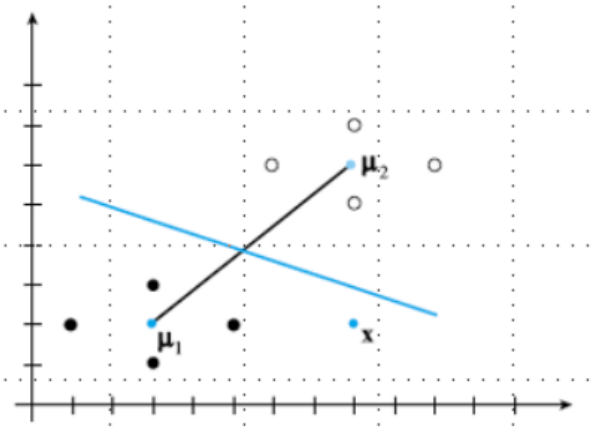


그림 2.15 두 통계 분포까지의 마할라노비스 거리

$$\mu_1\text{까지의 마할라노비스 거리} = \left(\begin{pmatrix} 8-3 & 2-2 \end{pmatrix} \begin{pmatrix} 3/8 & 0 \\ 0 & 3/2 \end{pmatrix} \begin{pmatrix} 8-3 \\ 2-2 \end{pmatrix} \right)^{1/2} = 3.062$$

$$\mu_2\text{까지의 마할라노비스 거리} = \left(\begin{pmatrix} 8-8 & 2-6 \end{pmatrix} \begin{pmatrix} 3/8 & 0 \\ 0 & 3/2 \end{pmatrix} \begin{pmatrix} 8-8 \\ 2-6 \end{pmatrix} \right)^{1/2} = 4.899$$

5. 거리와 유사도 (3)

❖ 코사인 유사도
- 문서 검색 응용에서 주로 사용 (단어의 출현 빈도가 특징 값)
- $s_{ij} = cos\theta = \frac{x_i^T x_j}{||x_i||||x_j||}$

❖ 이진 특징 벡터의 유사도
- $s_{ij} = \frac{n_{00}+n_{11}}{n_{00}+n_{11}+n_{01}+n_{10}}$
- $s_{ij} = \frac{n_{11}}{n_{11}+n_{01}+n_{10}}$

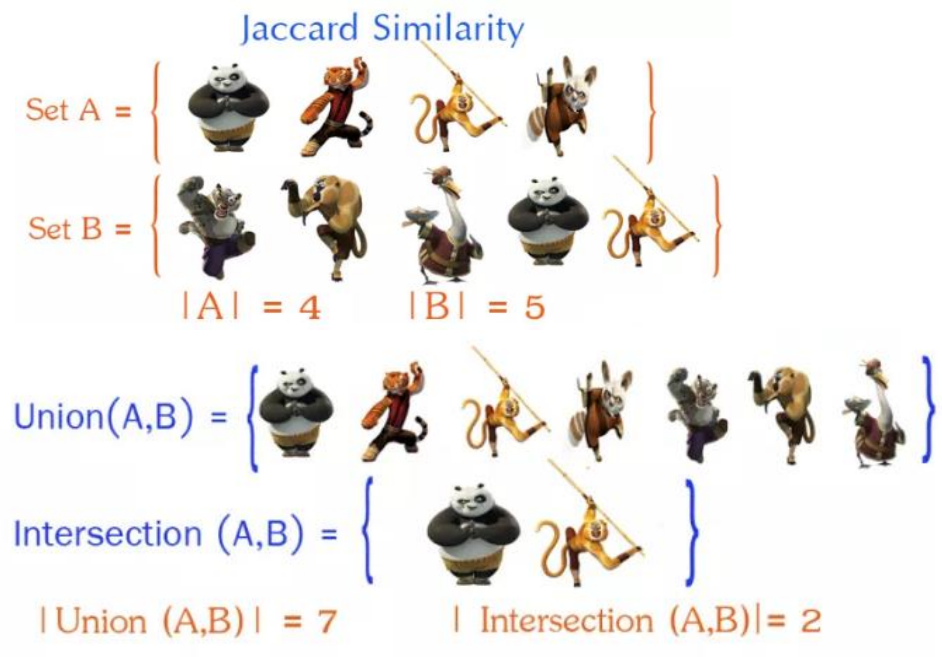
n_{00}: 둘다 0, n_{11}: 둘다 1
n_{01}: x_i는 0, x_j는 1
n_{10}: x_i는 1,x_j는 0

❖ $d_{ij} = s_{max} - s_{ij}$

5. 거리와 유사도 (5)

❖ Jaccard 유사도

– The Jaccard similarity measures the similarity between finite sample sets and is defined as the cardinality of the intersection of sets divided by the cardinality of the union of the sample sets. Suppose you want to find Jaccard similarity between two sets A and B it is the ration of cardinality of A ∩ B and A ∪ B



• Jaccard Similarity J(A,B) = $\frac{|A \cap B|}{|A \cup B|} = \frac{2}{7} = 0.286$

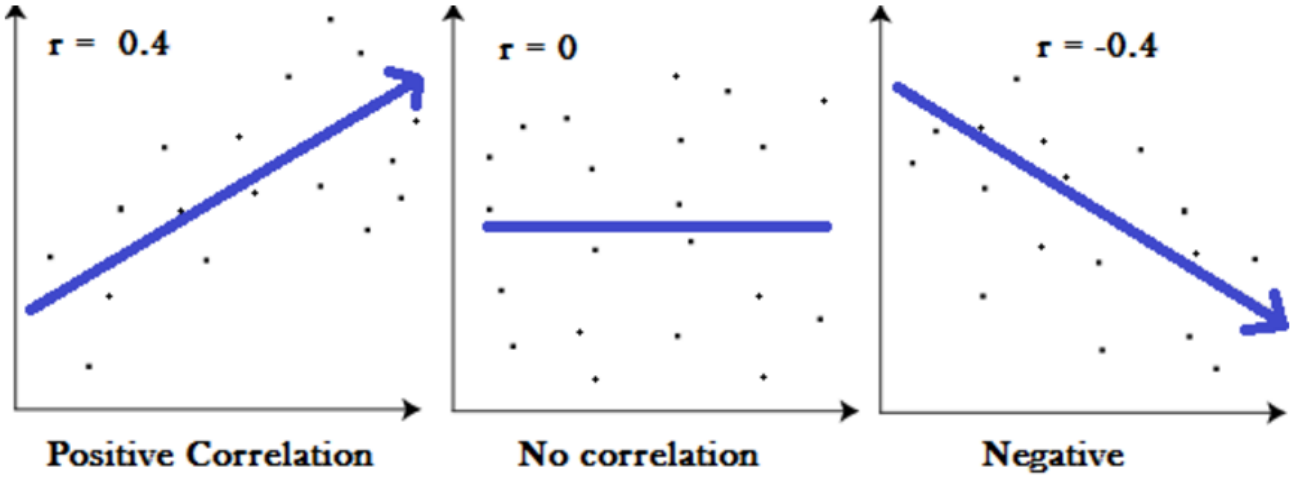
5. 거리와 유사도 (6)

❖ Pearson 상관계수

- 상관분석은 확률론과 통계학에서 두 변수간에 어떤 선형적 관계를 갖고 있는지를 분석하는 방법
- 회귀분석: 두 변수간에 원인과 결과의 인과관계 파악(방향, 정도와 수학적 모델)
- $r = \frac{\text{X와 Y가 함께 변하는 정도}}{\text{X와 Y가 따로 변하는 정도}}$

$$r = \frac{\Sigma XY - \frac{\Sigma X \Sigma Y}{n}}{\sqrt{[\Sigma X^2 - \frac{(\Sigma X)^2}{n}][\Sigma Y^2 - \frac{(\Sigma Y)^2}{n}]}}$$

- High correlation: .5 to 1.0 or -0.5 to 1.
  Medium correlation: .3 to .5 or -0.3 to .5.
  Low correlation: .1 to .3 or -0.1 to -0.3.



5. Distance 함수

- 유클리디안 거리(2차 노름) $d_E(x,y) = \|x-y\|_2 = \sqrt{(x-y)^T(x-y)} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$
- 1차 노름 $d_1(x,y) = \|x-y\|_1 = \sum_{i=1}^{n}|x_i-y_i|$
- ***p***차 노름 $d_p(x,y) = \sqrt[p]{\sum_{i=1}^{n}(x_i-y_i)^p}$
- 내적 $d_{IN}(x,y) = x \cdot y = \sum_{i=1}^{n} x_i y_i$
- 코사인 거리 $d_{\cos}(x,y) = 1 - \frac{x \cdot y}{\|x\| \, \|y\|}$
- 정규화된 유클리디안 거리 $d_{NE}(x,y) = \sqrt{\sum_{i=1}^{n}\frac{(x_i-y_i)^2}{\sigma_i^2}}$ (σ_i^2는 데이터의 분산)
- 마할라노비스 거리 (mahalanobis) $d_M(x,y) = \sqrt{(x-y)^T\Sigma^{-1}(x-y)}$ (Σ는 데이터의 공분산행렬)
- Pearson 상관계수 $r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$

Python program

❖ Iris data를 이용한 clustering

실습 목표

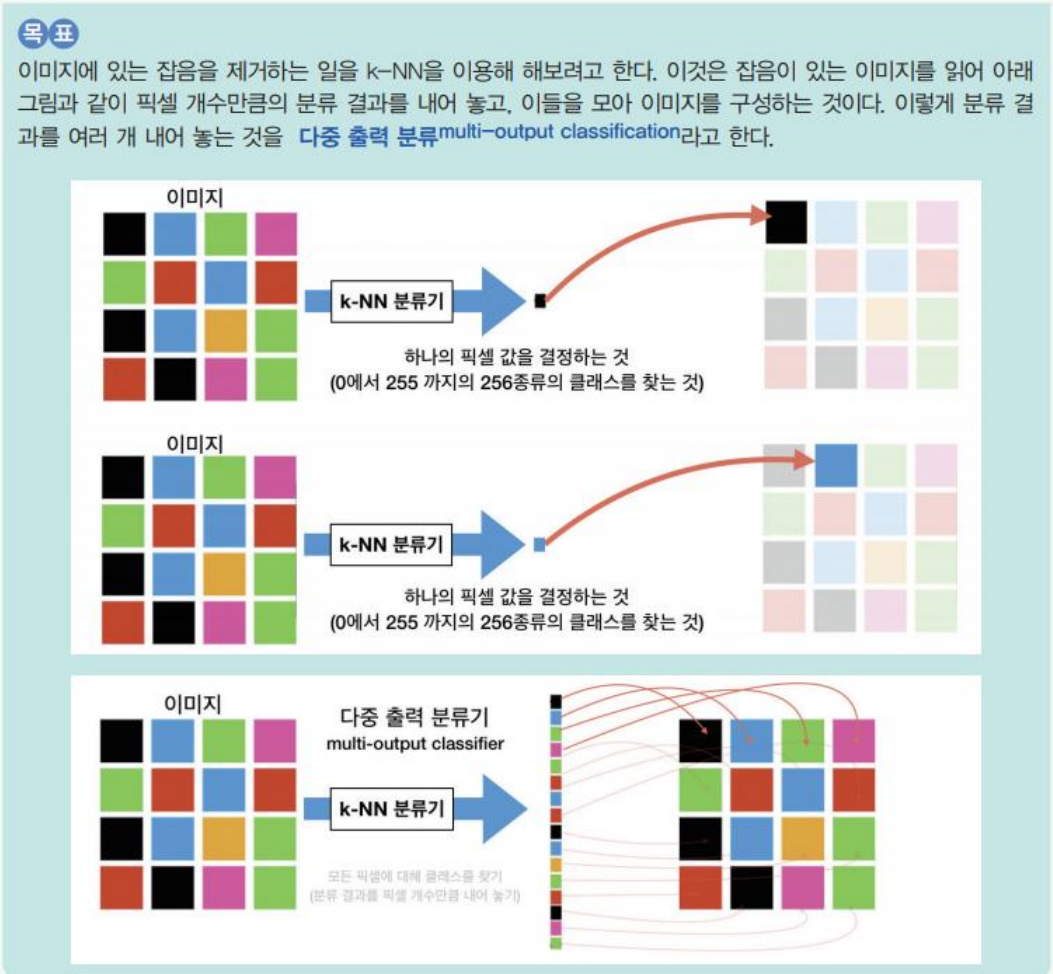
iris 데이터에 대하여 sklearn에서 제공하는 k-평균 군집화 알고리즘을 적용하자. 이때 target 정보는 이용하지 말고, 4개의 특성값만을 이용하여 3개의 군집으로 나누어 보도록 한다. 다음으로 k-평균 군집화 알고리즘을 이용하여 군집화된 데이터의 레이블 정보를 출력하면 다음 출력과 같이 [1, 1, 1, ..., 0, 0, 0, ..., 2, 2, 2, ..]로 레이블링 될 수 있다. 이와 같이 출력되는 이유는 군집화 알고리즘에서는 각각의 군집에 대한 레이블만을 출력할 뿐 어느 군집이 setosa(0), versicolor(1) virginica(2)에 속하는지에 대한 target 정보가 없기 때문이다. 이제 이 정보를 바탕으로 다시 레이블링을 하여 new_label을 만들어 보자. 마지막으로, 이 군집화 결과값과 원래 iris 데이터의 target과의 차이를 비교하여 정확도를 다음과 같이 출력하도록 하자.

- https://colab.research.google.com/drive/1XQe7_FCAHIbnig44Lplew2Bildufn7oF

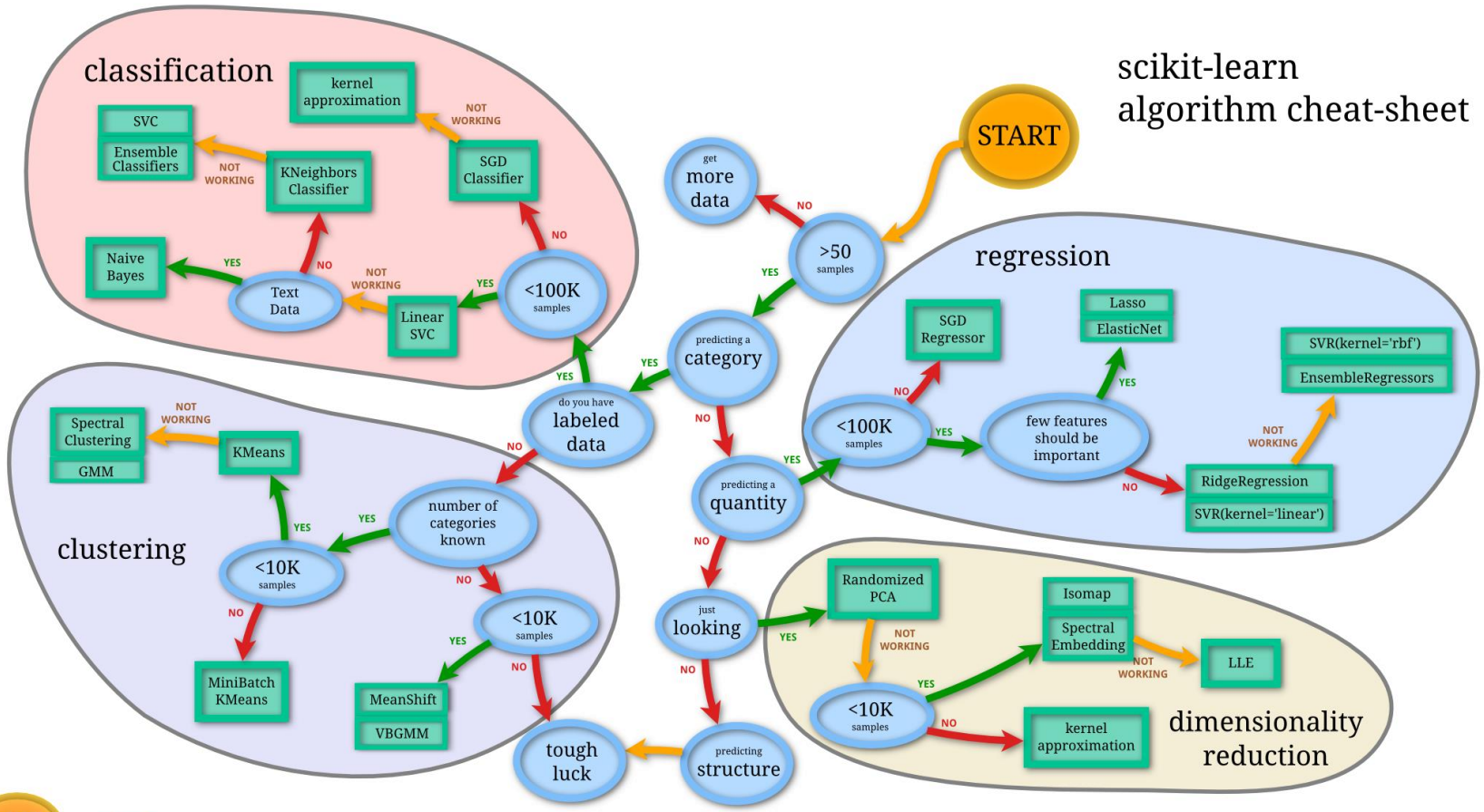
Mini Project

❖ 잡음제거: k-NN의 활용 (과제2)

- https://colab.research.google.com/drive/1X45XKyEHgOpR0Mq56eZvEGZPpoA4vBVq



Scikit-Learn (1)



Scikit-Learn (2)