

Clustering, Evaluating, Improving

D조

김용규 배윤성 홍예지
서주희 이예림 김강민

목 차

1. 머신러닝
2. Knn 알고리즘
3. Clustering
4. Confusion matrix, F1 score, ROC curve
5. Cross validation(교차검증)
6. Grid search (그리드 서치)

1. 머신러닝

머신러닝이란?

인공지능의 한 연구분파로 데이터에 내재된 패턴, 규칙, 의미 등을 컴퓨터가 스스로 학습할 수 있는 알고리즘을 연구하는 분야



1. 머신러닝- 머신러닝 알고리즘의 종류

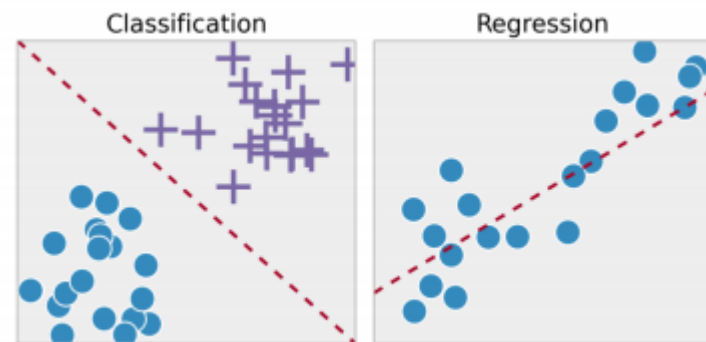
Types	Tasks	Algorithms
지도학습 (Supervised Learning)	분류 (Classification)	<ul style="list-style-type: none"> • KNN : k Nearest Neighbor • SVM : Support Vector Machine • Decision Tree (의사결정 나무) • Logistic Regression
	예측 (Prediction)	<ul style="list-style-type: none"> • Linear Regression (선형 회귀)
비지도학습 (Unsupervised Learning)	군집 (Clustering)	<ul style="list-style-type: none"> • K-Means Clustering • DBSCAN Clustering • Hierarchical Clustering (계층형 군집)
강화학습 (Reinforcement Learning)		<ul style="list-style-type: none"> • MDP : Markov Decision Process

1. 머신러닝- 지도학습

: 주어진 데이터와 레이블(정답)을 이용해서 미지의 상태를 예측(회귀)하거나 분류 하는 학습방법

▪ 분류와 회귀의 비교

	분류 (Classification)	회귀 (Regression)
결과	학습데이터의 레이블 중 하나를 예측 (discrete)	연속된 값을 예측 (Continuous)
예제	학습데이터가 A, B, C 인 경우 결과는 A, B, C 중 하나다. 예) 스팸메일 필터	결과 값이 어떠한 값도 나올 수 있다. 예) 주가 분석 예측



1. 머신러닝- 비지도학습

: 데이터 자체에서 유용한 패턴을 찾아내는 학습방법

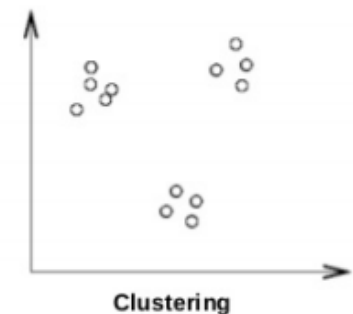
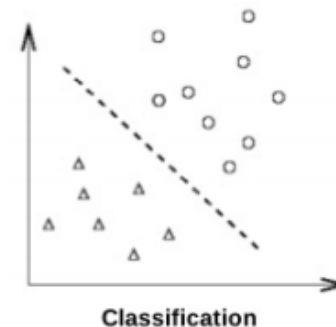
Ex) 군집화, 이상추출, 데이터 분포 추측

- 지도학습 vs 비지도학습

특정 값을 계산하는 함수로 추측 vs 데이터의 성질 직접 추측

▪ 분류와 군집의 비교

	분류 (Classification)	군집 (Clustering)
공통점	입력된 데이터들이 어떤 형태로 그룹을 형성하는지가 관심사	
차이점	레이블이 있다.	레이블이 없다. 예) 의학 임상실험 환자군 구별 예) 구매자 유형 분류



Knn 알고리즘

Knn 알고리즘 : K – nearest neighbors

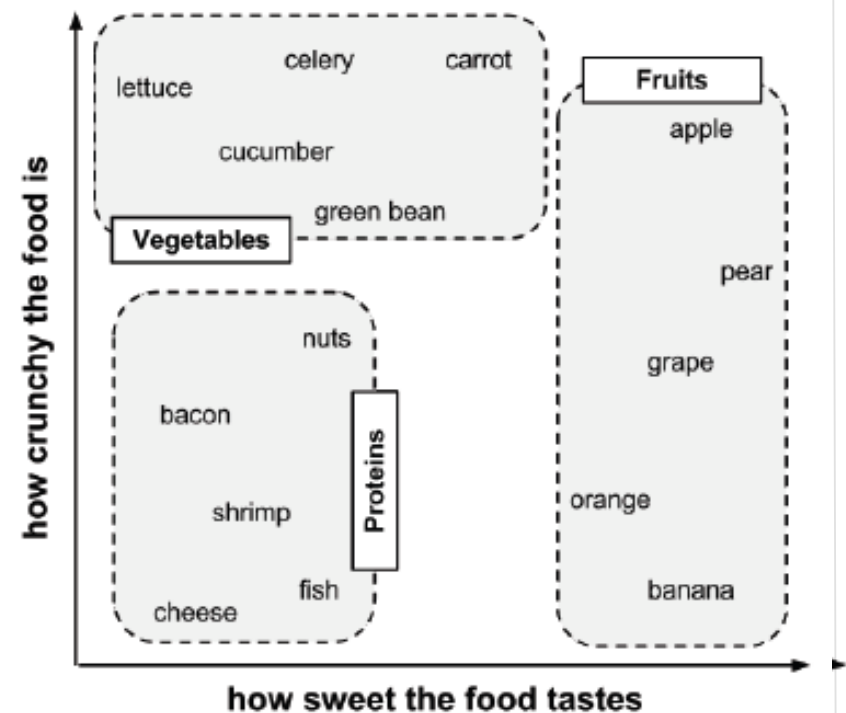
- 머신러닝 – 지도학습 - 분류

- 새로운 데이터가 어느 그룹에 속하는지 분류하기 위해 그 데이터와 가장 가까이 있는 데이터를 알아보는 알고리즘

Knn 알고리즘 : 최근접 이웃을 사용한 분류

[눈 가린 식사]

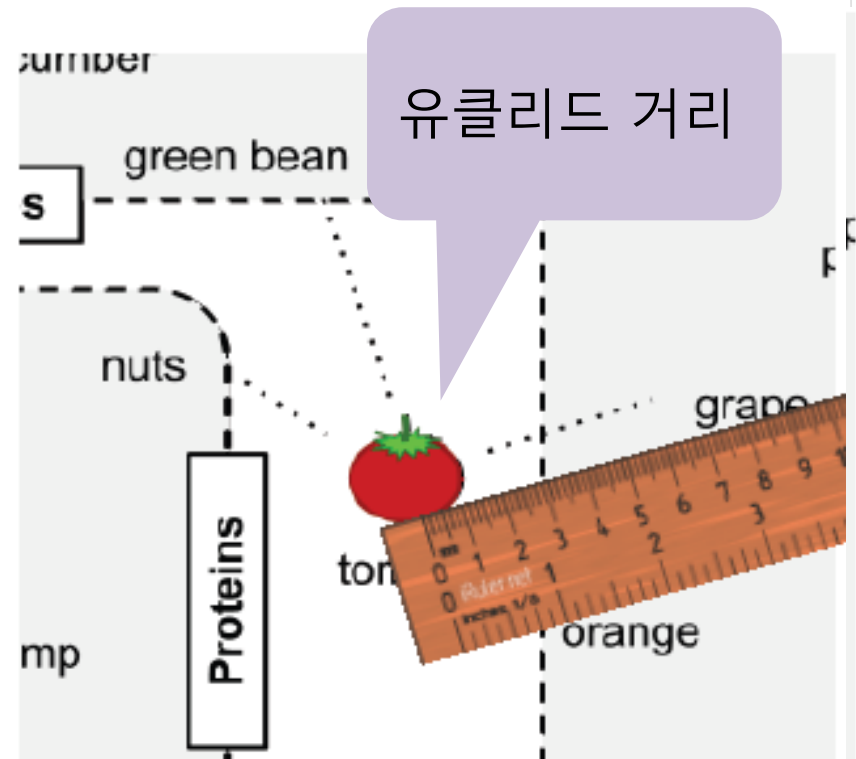
재료	단맛	아삭거림의 정도	음식 종류
apple	10	9	과일
bacon	1	4	단백질
banana	10	1	과일
carrot	7	10	야채
celery	3	10	야채
cheese	1	1	단백질



Knn 알고리즘 : 최근접 이웃을 사용한 분류

[눈 가린 식사]

재료	단맛	아삭거림의 정도	음식 종류
apple	10	9	과일
bacon	1	4	단백질
banana	10	1	과일
carrot	7	10	야채
celery	3	10	야채
cheese	1	1	단백질



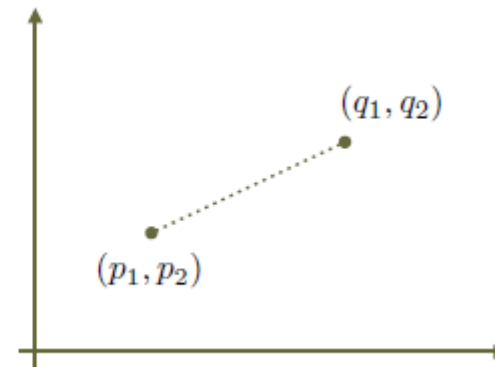
Tomato 는 과일인가? 야채인가?

Knn 알고리즘 : 최근접 이웃을 사용한 분류

[거리 계산]

유클리드 거리: 가장 짧은 직접적인 경로를 나타내는 '일직선으로' 측정Euclidean distance
2-dimension

$$D(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$



• N-dimension

$$D(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

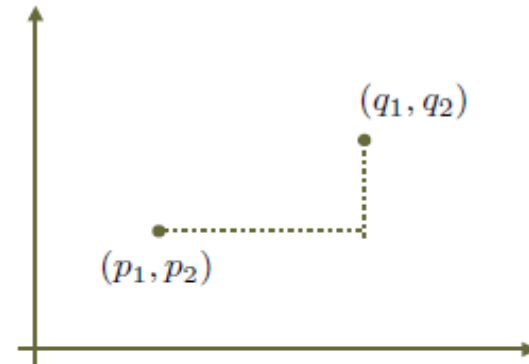
Knn 알고리즘 : 최근접 이웃을 사용한 분류

[거리 계산]

맨하튼 거리: 보행자가 도시의 블록을 걸어 다니는 경로를 바탕으로

- Euclidean distance
2-dimension

$$D(p, q) = |p_1 - q_1| + |p_2 - q_2|$$



- N-dimension

$$D(p, q) = |p_1 - q_1| + |p_2 - q_2| + \cdots + |p_n - q_n|$$

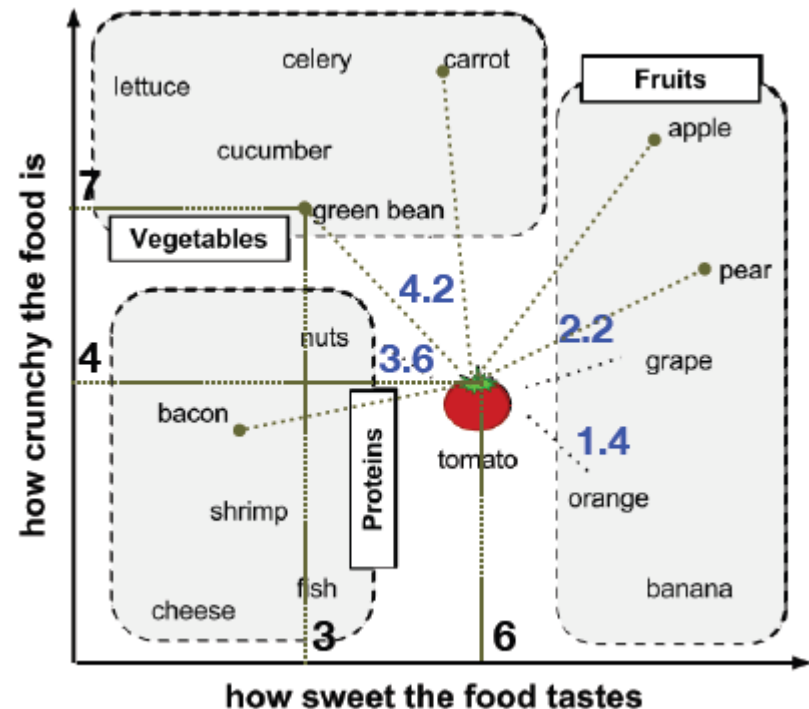
Knn 알고리즘 : 최근접 이웃을 사용한 분류

[눈 가린 식사]

Tomato 는 과일인가? 야채인가?

$$D(\text{tomato}, \text{greenbean}) = \sqrt{(6 - 3)^2 + (4 - 7)^2} = 4.2$$

	sweetness	crunchiness
tomato	6	4
green bean	3	7



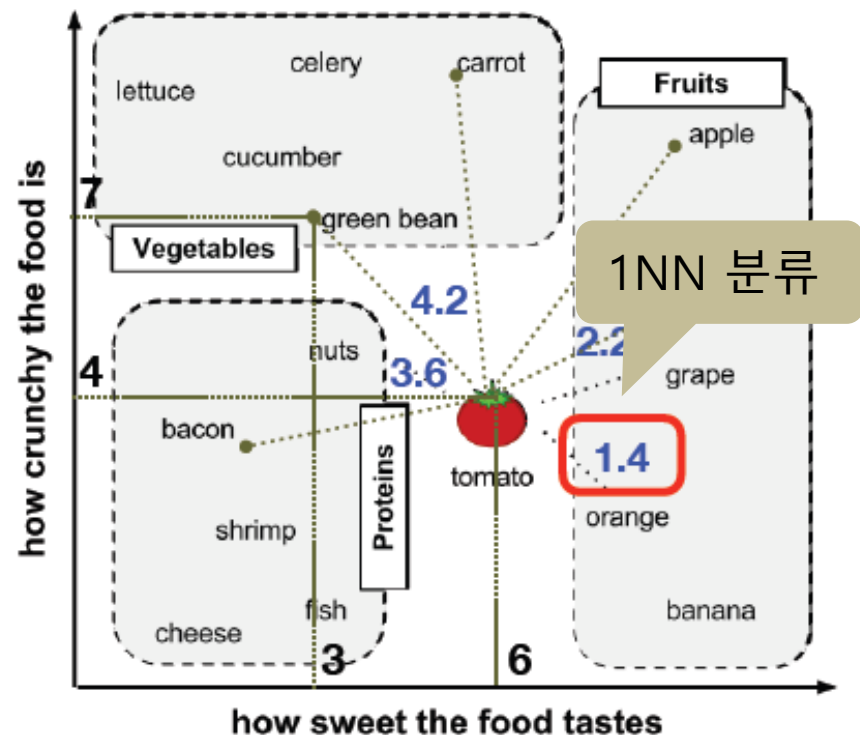
Knn 알고리즘 : 최근접 이웃을 사용한 분류

[눈 가린 식사]

Tomato 는 과일인가? 야채인가?

$$D(\text{tomato}, \text{greenbean}) = \sqrt{(6 - 3)^2 + (4 - 7)^2} = 4.2$$

	sweetness	crunchiness
tomato	6	4
green bean	3	7



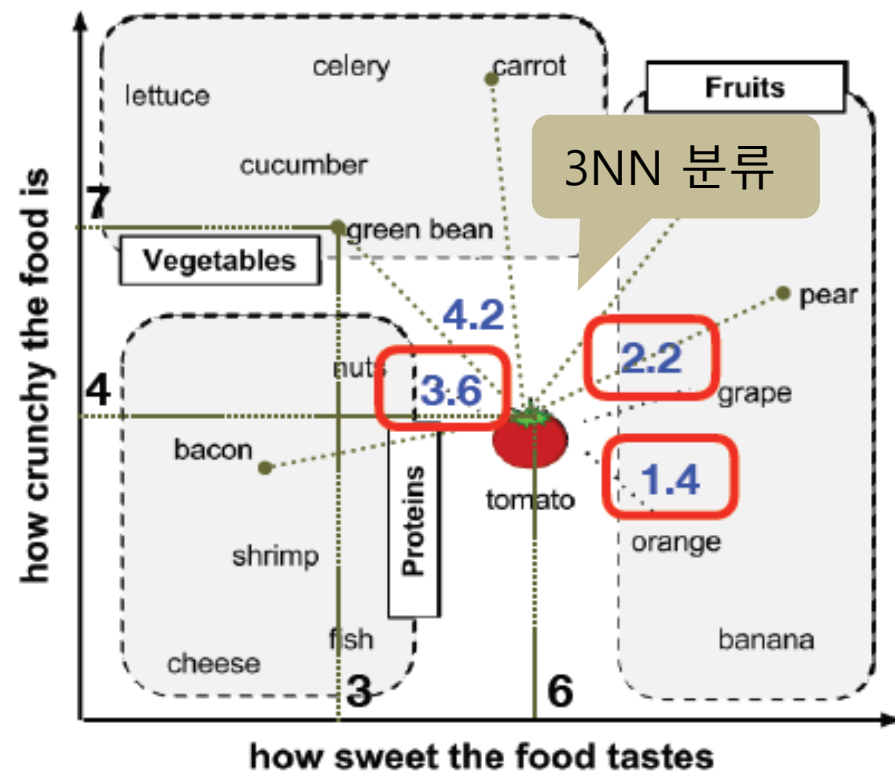
Knn 알고리즘 : 최근접 이웃을 사용한 분류

[눈 가린 식사]

Tomato 는 과일인가? 야채인가?

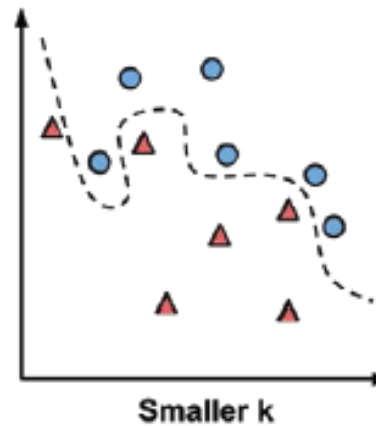
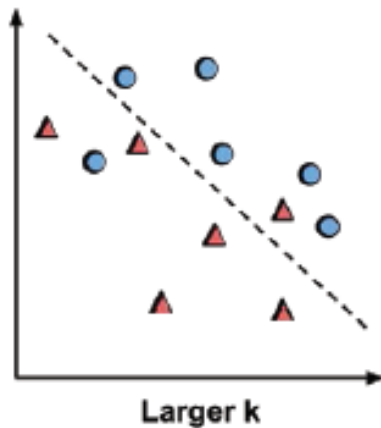
$$D(\text{tomato}, \text{greenbean}) = \sqrt{(6 - 3)^2 + (4 - 7)^2} = 4.2$$

	sweetness	crunchiness
tomato	6	4
green bean	3	7



Knn 알고리즘 : 최근접 이웃을 사용한 분류

[적당한 k의 선택]



- 보통 k는 3과 10 사이에서 결정
- 일반적으로 훈련 데이터의 개수에 제곱근으로 설정
ex) 데이터수 = 15개 15의 제곱근은 3.87 => k=4
- hyper-parameter tuning- 검증

Knn 알고리즘 : 최근접 이웃을 사용한 분류

[kNN 알고리즘은 왜 게으른가?]

- 학습과 모델을 만드는 과정(추상화 & 일반화 과정) 생략.
- kNN 알고리즘은 어떤 것도 학습하지 않고 훈련 데이터를 그대로 저장할 뿐이다.
- 훈련 데이터에 심한 의존성

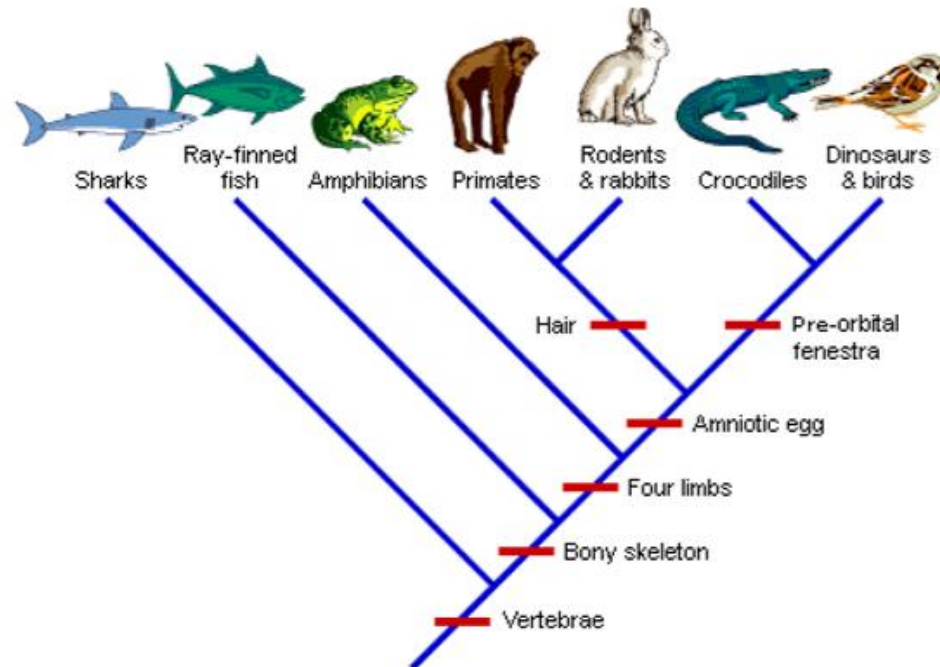
*** kNN을 배경지식으로 간직 -> 로지스틱 회귀 or 딥러닝

Clustering

- 머신러닝 - 비지도학습 - 클러스터링

- 많은 개체(object)들을 일정한 속성에 따라 몇 개의 군집(cluster)으로 분류하여, 같은 군집에 속한 개체들의 유사성과 다른 군집에 속한 개체 간의 상이성을 규명하고자 하는 통계 분석 방법

3. Clustering



Reading phylogenetic trees: A quick review
(Adapted from evolution.berkeley.edu)

3. Clustering - 군집분석 절차

- 분석하고자 하는 개체들의 여러가지 특성들을 유사성 거리로 환산하여 유사성 거리가 상대적으로 가까운 개체들을 동질한 집단으로 군집화 하는 것
- ① 어떠한 특성들을 비교할 것인가? (설명변수 선정)
- ② 어떻게 유사성 거리를 측정할 것인가? (유사성 거리 측정방법)
- ③ 어떻게 동질적인 집단으로 분류할 것인가? (군집화 방법)

3. Clustering - 설명변수 선정

- 군집분석에서 설명변수의 선택에 의해 유사성 거리가 달라지므로 **설명변수의 선택에 주의**
 - 회귀분석이나 판별분석 등 에서와 같이 통계적으로 의미 없는 설명변수를 제외 또는 추가하는 방법이 없음
 - 선정된 설명변수들이 모두 **동일한 비중**으로 유사성 거리의 측정에 반영됨
 - 최종결과에 대한 통계적 유의성 검정의 방법이 없음
- ⇒ 설명변수의 선정에 유의하여야 하며 경우에 따라 요인분석을 통하여 사전에 변수들 간의 중복 부분을 제외한 **순수요인(요인점수)**를 도출하여 이용하기도 함

3. Clustering – 유사성 거리 측정법

① 유클리드 거리(Euclidean distances)

- 변수값 차이를 제곱하여 합산한 거리로 다차원 공간에서의 직선 최단거리
- 가장 일반적인 측정방법
- 별도의 언급이 없는 경우 '거리'라고 하면 일반적으로 유클리드 거리를 의미

$$d_{ij} = [\sum_{k=1}^p (x_{ik} - x_{jk})^2]^{1/2}$$

② 유클리드 제곱거리(Squared Euclidean distances)

- 유클리드 거리를 제곱한 거리

$$d_{ij}^2 = \sum_{k=1}^p (x_{ik} - x_{jk})^2$$

③ 도시-블록, 맨해튼 거리(City-Block distances, Manhattan distances)

- 변수값 차이의 절대값을 합한 거리

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

④ 코사인 거리(Cosine distances)

- 변수값의 코사인 벡터로 환산한 거리

⑤ 체비셰프 거리(Tchebychev distances)

- 변수값 차이의 절대값 중 가장 큰 값을 환산한 거리

⑥ 민코우스키 거리(Minkowski distances)

- 변수값 차이의 p 제곱합의 1/p 누승근으로 환산한 거리

⑦ 피어슨 상관계수(Pearson's correlation coefficient)

$$\rho_{ij} = \frac{\sum_{k=1}^p (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^p (x_{ik} - \bar{x}_i)^2 (x_{jk} - \bar{x}_j)^2}}$$

때에 따라 필요한 거리 측정법 사용

3. Clustering – 변수의 표준화

※ 반드시 변수값들을 표준화 해야 함, why?

>>> 설명변수들의 단위가 서로 다르면, 설명변수들의 중요도가 서로 다르게 계산됨

예) 자동차의 배기량과 가격을 기준으로 자동차를 군집화

- 배기량은 1,000-2,000cc 정도의 편차가 있을 수 있지만, 가격은 1,000,000 원 이상의 편차가 있을 수 있음
- 배기량에서는 1,000 이란 차이가 매우 큰 차이지만, 가격에 있어서 1,000 이란 차이는 매우 적은 차이임
- 유사성 거리의 측정에 있어서 배기량이 차이가 거의 무시되며 가격의 차이가 개체 간의 거리를 완전히 좌우함

∴ 선정된 설명변수들의 단위가 서로 다를 경우에는 표준화를 통하여 설명변수들의 중요도를 동일하게 해야만 한다.

3. Clustering – 군집화 방법

- 1) 비계층적 방법(Nonhierarchical method)
군집이 형성된 이후에도 일정 기준에 따라 개체들이 이합집산과정을 되풀이
 - ① 클러스터 중심(centroid) 또는 평균 기반 클러스터링 k-means
 - ② 빈도수가 많은 중간점(medoid)기반 클러스터링 k-medoids
 - ③ 밀도 기반 클러스터링
- 2) 계층적 방법(Hierarchical method)
군집의 형성에 계층이 있어서 일단 한 군집에 속하게 된 두 개체는 다시 흩어지지 않음
 - ① 계층적(hierarchical) 클러스터링

3. Clustering - K-means Clustering

- 1) 비계층적 방법(Nonhierarchical method) 군집의 개수 k 를 미리 정해두어서, 그 군집의 수에 맞게 군집화 하는 알고리즘
- 2) 데이터와 데이터가 속한 군집의 중심점과의 거리의 제곱합을 최소화시키며 k 개의 군집을 구함
- 3) 반복연산으로 제일 좋은 군집을 찾는 방법

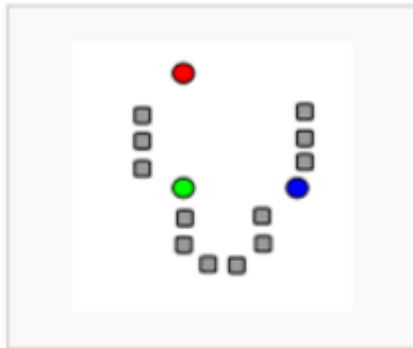


k-medoids clustering

- k-means clustering과 유사하다.
- 차이점은 cluster의 중심을 임의의 점이 아니라 데이터 세트 중 하나를 선정한다.
- 장점 : 실 데이터를 중심으로 함으로 노이즈 처리가 우수
- 단점 : k-means에 비해 계산량이 많다.

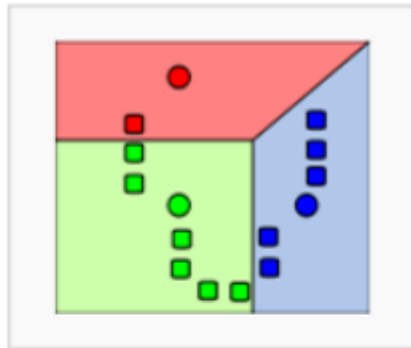
3. Clustering - K-means Clustering

■ 알고리즘 수행 절차



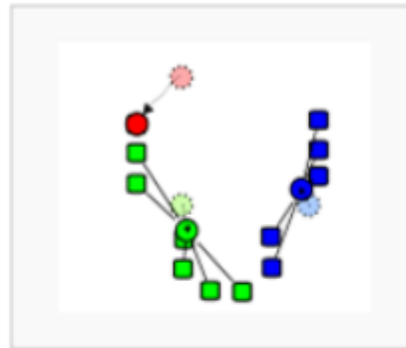
Step 1

- Cluster 수인 k를 정의
- 초기 k개 군집 중심 임의 지정 (initial centroid)
- 위 그림에서는 k=3



Step 2

- 모든 데이터들의 거리 계산 후 가장 가까운 Centroid로 Clustering



Step 3

- 각 Cluster마다 계산하여 새로운 중심 계산



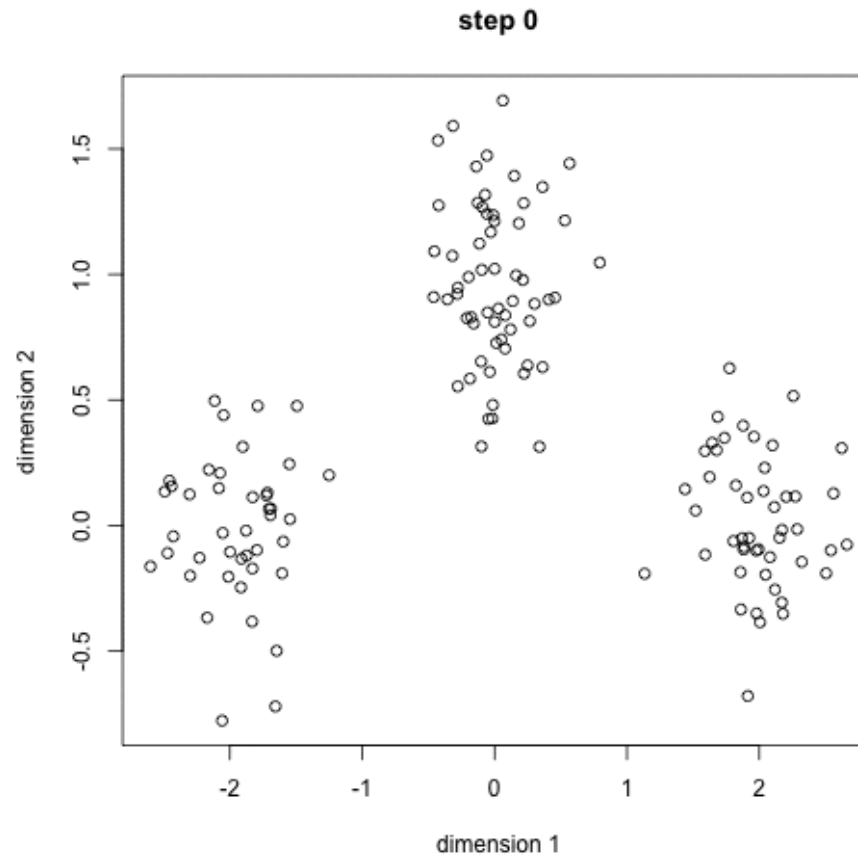
Step 4

- Step 2, 3 을 반복
- 데이터가 자신이 속하는 Cluster를 변경하지 않으면 학습 완료

—centroid :클러스터 중심

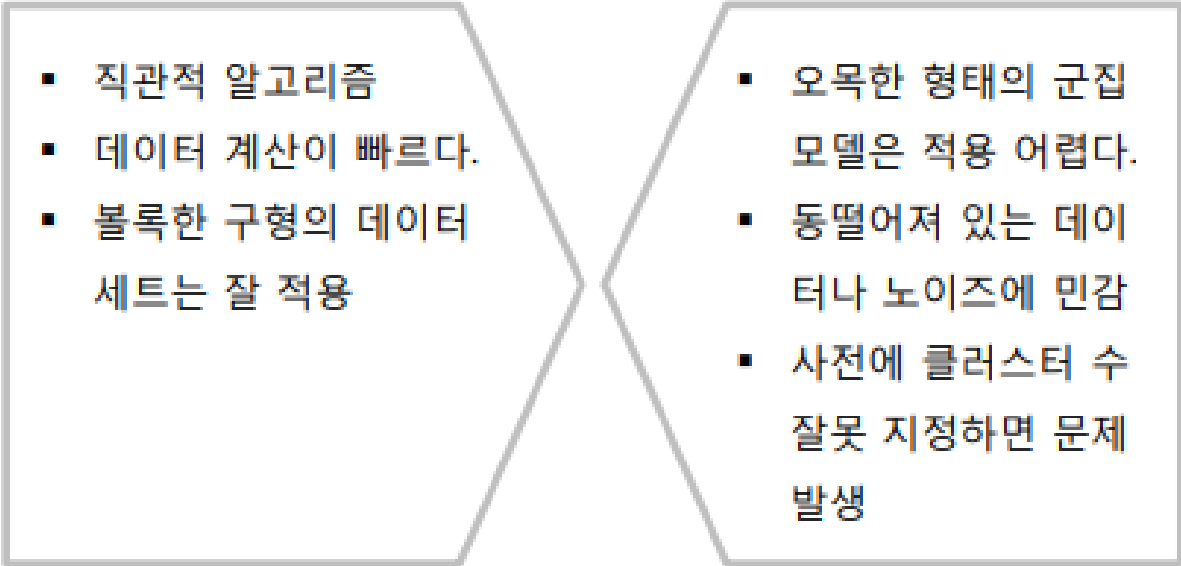
3. Clustering - K-means Clustering

- 알고리즘 수행 절차



3. Clustering - K-means Clustering

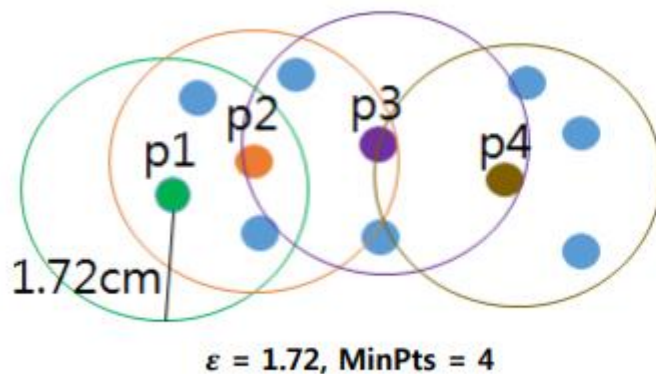
▪ 장/단점

- 
- A diagram consisting of two gray-outlined hexagons pointing towards each other, forming a larger hexagonal shape. The left hexagon contains three bullet points representing advantages, and the right hexagon contains three bullet points representing disadvantages.
- 직관적 알고리즘
 - 데이터 계산이 빠르다.
 - 불룩한 구형의 데이터 세트는 잘 적용
- 오목한 형태의 군집 모델은 적용 어렵다.
 - 동떨어져 있는 데이터나 노이즈에 민감
 - 사전에 클러스터 수 잘못 지정하면 문제 발생

3. Clustering – 밀도 기반 클러스터링과 원리

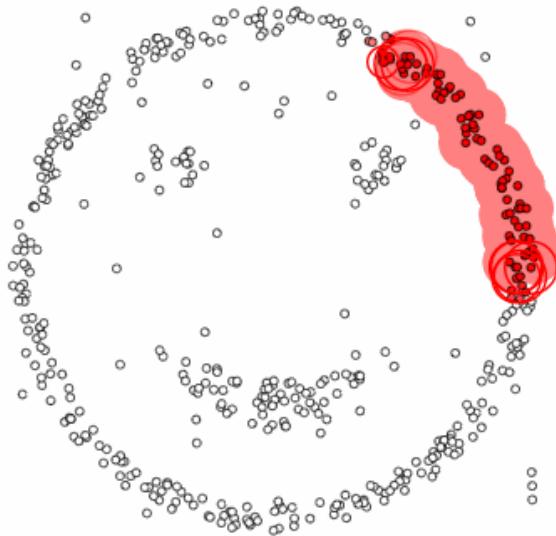
DBSCAN: Density Based Spatial Clustering of Application with Noise

1) DBSCAN에서 밀도 있게 연결돼 있는 데이터 집합은 동일한 Cluster이다.



- 먼저 좌표 공간에 학습 데이터를 표시
- 밀도 : 반경 (ϵ , Epsilon) 안에 있는 다른 좌표 점의 수
- MinPts : 어떤 좌표점이 Cluster를 형성할 수 있는 최소 좌표점의 개수
- A 점의 밀도가 MinPts 이상이면 Core, 미만이면 Noise로 정의
- Cluster 구성 후 이웃 점을 차례로 방문하면서 Core인지를 판단 (p1 → p2 → p3 → p4, 즉 p1과 p4는 같은 Cluster이다)

3. Clustering – 밀도 기반 클러스터링과 원리



장/단점

- 노이즈 식별에 강하다
- 군집의 수를 미리 정할 필요가 없다.

- 밀도 반경(ϵ)과 최소 이웃 수(MinPts)가 민감하게 작용
- Cluster별 밀도가 서로 다른 경우 적용 어려움

epsilon = 1.00
minPoints = 4

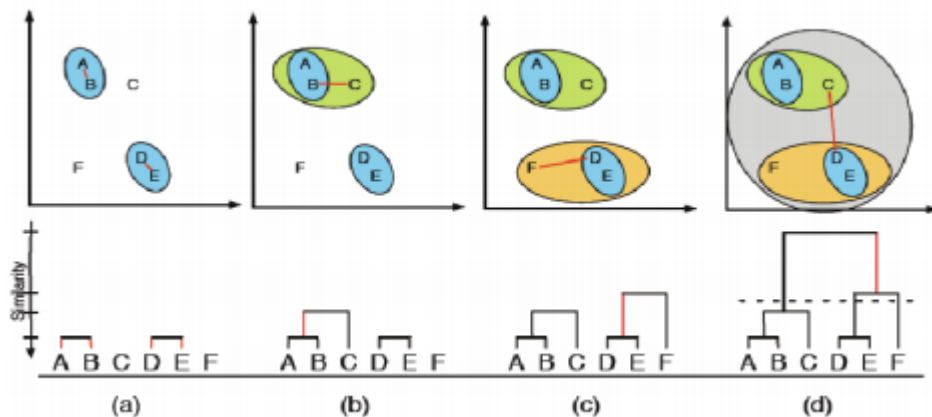
Restart

Pause

3. Clustering - Hierarchical Clustering(계층적 군집)

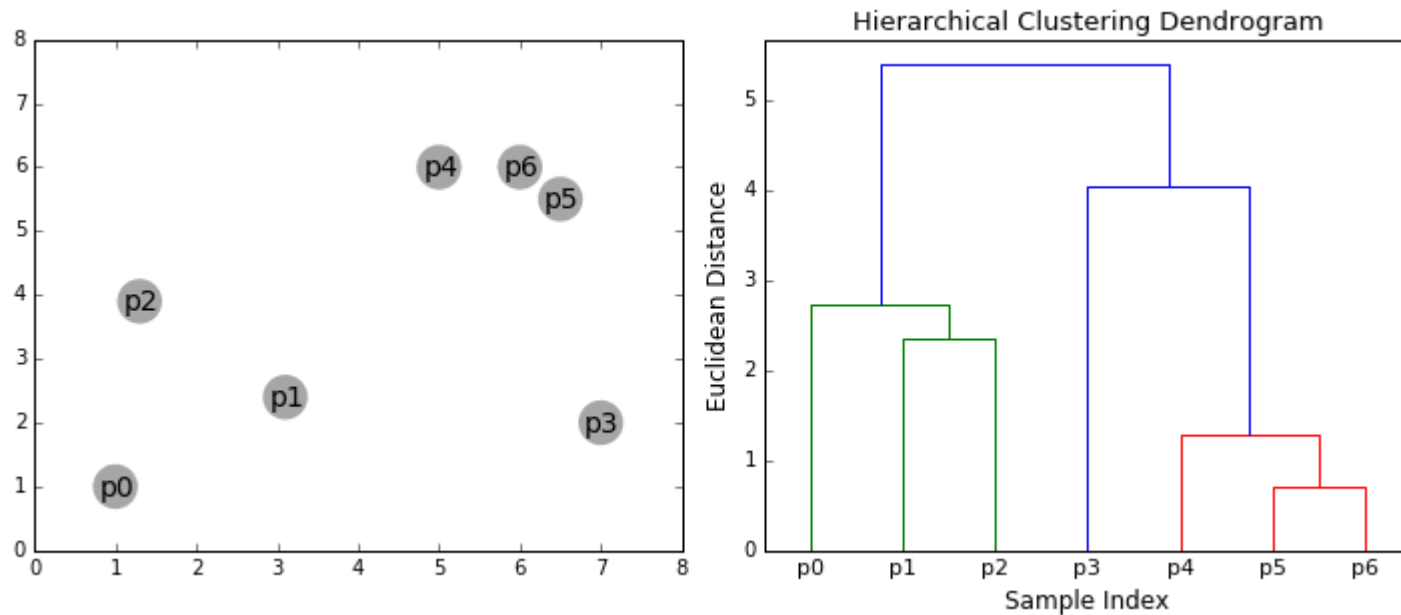
- 특정 알고리즘에 의해 데이터들을 연결하여 계층적으로 클러스터를 구성해 나가는 방법

** 쉽게 말하면 처음에 데이터 세트의 모든 점을 군집의 원점으로 시작해 유사한 Cluster로 합쳐 나간다



- **Step 1** : 모든 데이터를 단일 클러스터로 정의 한다.
- **Step 2** : 각 클러스터간 유사성을 계산한다.
- **Step 3** : 유사성이 높은 두 개의 Cluster를 합한다.
- **Step 4** : 2, 3단계를 전체 Cluster 수가 1이 될 때 까지 반복한다.

3. Clustering - Hierarchical Clustering(계층적 군집)



장/단점

- 초기에 Cluster 개수를 정할 필요가 없다.
- 직관적 이해가 편하다

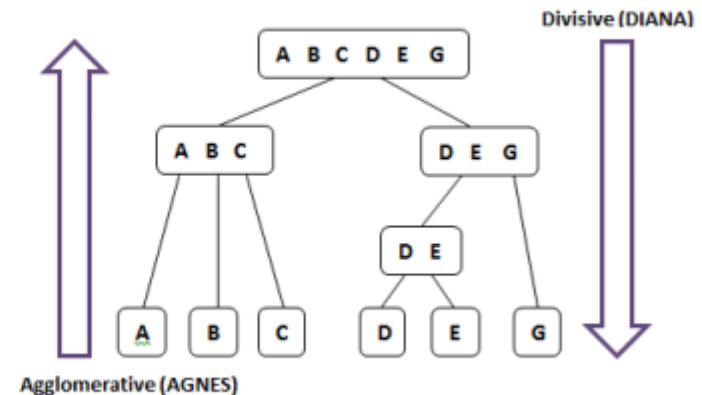
- 자료가 크기가 크면 복잡해져 적용하기 어렵다.

3. Clustering - Hierarchical Clustering(계층적 군집)



divisive clustering

- 위에서 설명한 것은 Hierarchical Clustering 중 Agglomerative Clustering (병합적 군집) 이다
- Divisive Clustering (분할적 군집) 은 위 방법과 반대로 전체를 하나로 묶은 후 유사성이 낮은 Cluster로 분리하는 방법이다.
- 분할적 접근은 잘못된 결정이 하위 클러스터로 파급되는 영향도가 크다는 단점이 있다.



학습 목표

- 각 모델들을 학습 시킬 때 최적의 매개변수 값을 찾는 방법을 익힌다.
- 더 나은 성능을 가진 최종 모델을 만드는 방법을 익힌다.

4. Confusion matrix

혼동 행렬

모델에서 구한 분류의 예측값과 데이터의 실제 분류인 실제값의 발생 빈도를 나열한 그림

		Diagnosis	
		No cancer	Cancer
True state	No cancer	<i>TN</i>	<i>FP</i>
	Cancer	<i>FN</i>	<i>TP</i>

* 관심범주의 예측 관계 판단

- 참 긍정 (TP : True Positive)
실제 YES 를 YES 로 예측
- 참 부정 (TN : True Negative)
실제 NO 를 NO 로 예측
- 거짓 긍정 (FP : False Positive)
실제 NO 를 YES 로 예측
- 거짓 부정 (FN : False Negative)
실제 YES 를 NO 로 예측

4. Confusion matrix

Accuracy (정확도)

전체 예측에서 (예측이 Y이든 N이든 무관하게) **옳은 예측**의 비율

		Diagnosis	
		No cancer	Cancer
True state	No cancer	TN	FP
	Cancer	FN	TP

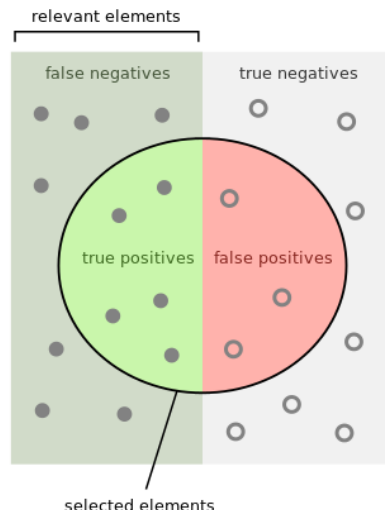
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \text{Accuracy}$$

* 95% 정확도인 모델 => 5%가 잘못 분류됨

$$\text{정확도} = \frac{\text{대각원소의 합}}{\text{혼동행렬의 모든 원소의 합}} \times 100\%$$

4. Confusion matrix, F1 score, ROC curve



How many selected items are relevant?

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Precision and Recall (출처 : wiki)

$$\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False Positive})$$

Precision을 스팸 메일의 예로 설명해보겠습니다.

Precision = 스팸 메일인데 스팸 메일이라 맞춘 수(TP) / (스팸 메일인데 스팸 메일이라 맞춘 수(TP) + 스팸 메일이 아닌데 스팸 메일이라고 맞춘 수(FP))

$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative})$$

Recall = 암 환자인데 암 환자라 맞춘 수(TP) / (암 환자인데 암 환자라 맞춘 수(TP) + 암 환자인데 암 환자가 아니라고 맞춘 수(FN))

4. Confusion matrix, F1 score, ROC curve

F – measure?

precision 과 recall을 하나로 합하는 모델의 성능 측정

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{recall} + \text{precision}} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

장점: 결과가 하나의 단일값이므로, 나란히 모델들을 비교하기에 편리
단점: 정밀도와 재현율에 같은 가중치를 부여함. (유효하지 않은 가정)

G-measure : 기하평균을 활용한 모델 성능 측정 (F measure: 조화평균 이용)

$$G = \sqrt{\text{precision} \cdot \text{recall}}$$

4. Confusion matrix, F1 score, ROC curve – ROC curve

ROC (Receiver Operating Characteristics; 수신자 조작 특성) Curve

어떤 검사의 판단결과(binary classifier)의 performance를 보여주는 그래프

Y축 : TPR (True Positive Rate)

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

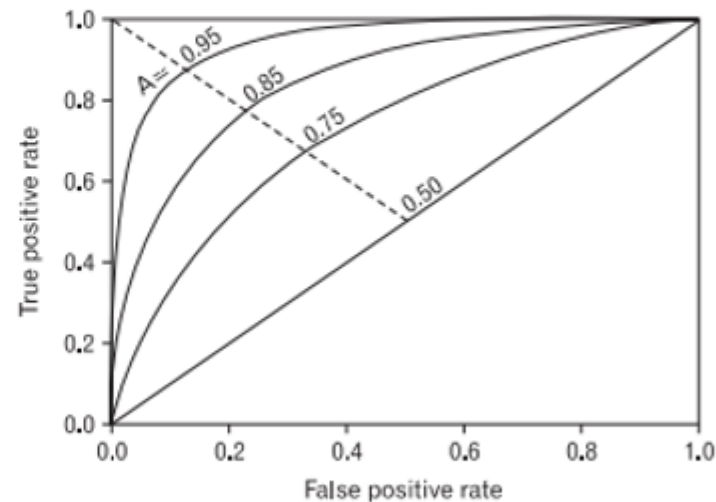
1인 케이스에 대해 1로 예측한 비율

X축 : FPR (False Positive Rate)

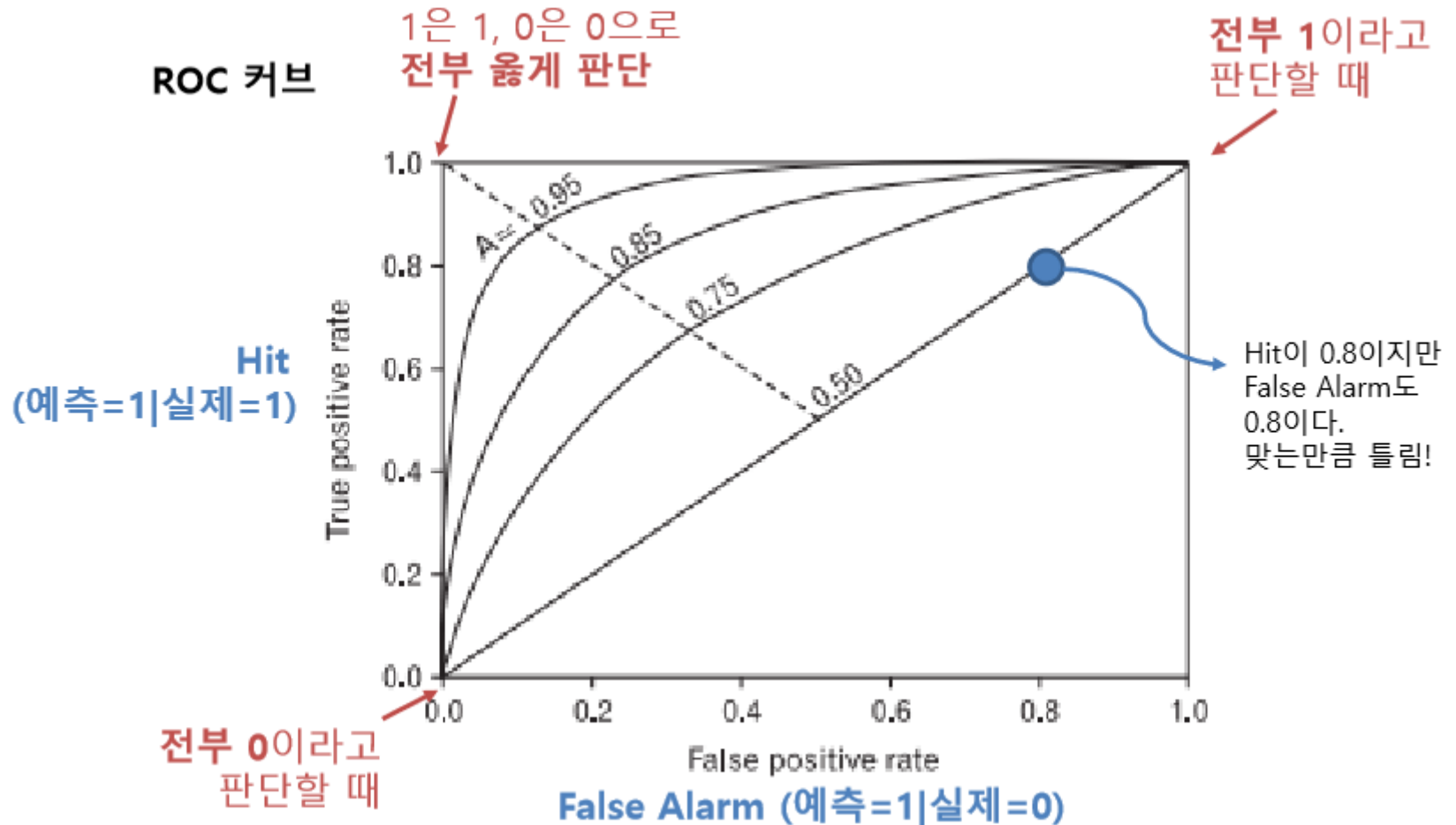
= 1 – Specificity(특이도)

$$= 1 - \frac{TN}{TN + FP}$$

0인 케이스에 대해 1로 잘못 예측한 비율



4. Confusion matrix, F1 score, ROC curve – ROC curve

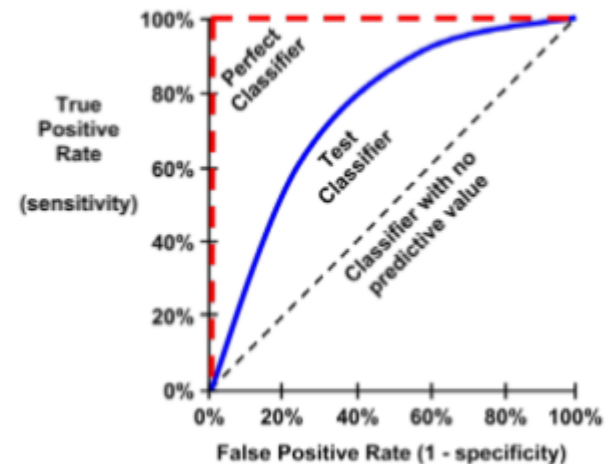


4. Confusion matrix, F1 score, ROC curve – ROC curve

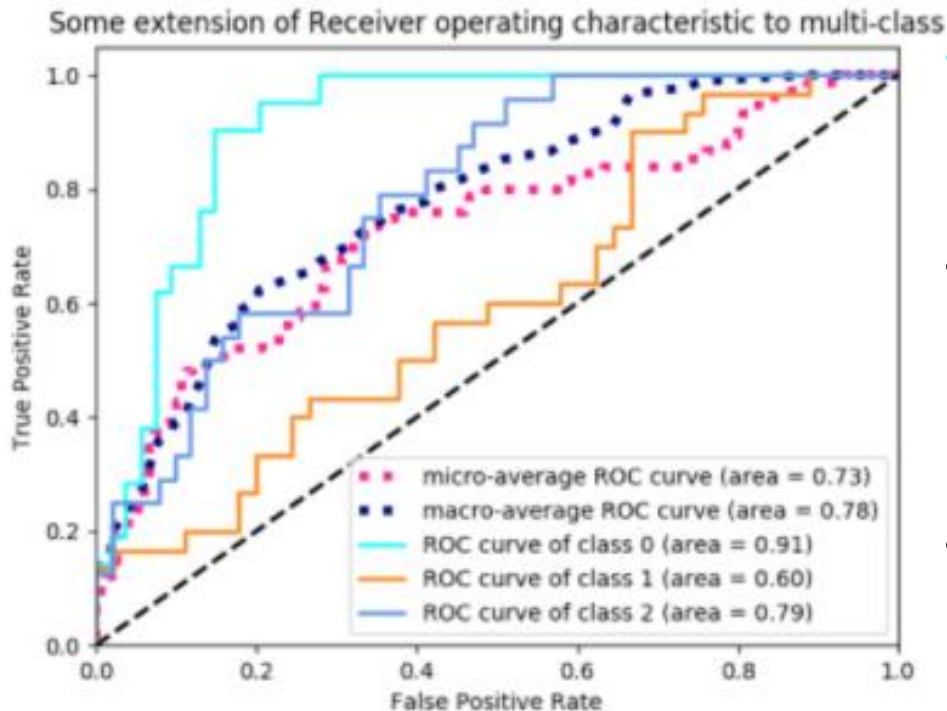
커브가 **왼쪽 위 꼭지점**에 가까울수록 좋은 분류기!

- ① no predicted value 에 가까울수록 ↓
Perfect classifier (빨간 선) 에 가까울수록 ↑
- ② **AUC** (ROC 커브 밑 면적; Area Under the Curve)
0.5 < AUC < 1 에서 1에 가까울수록 ↑

- 0.9 - 1.0 = A (outstanding)
- 0.8 - 0.9 = B (excellent/good)
- 0.7 - 0.8 = C (acceptable/fair)
- 0.6 - 0.7 = D (poor)
- 0.5 - 0.6 = F (no discrimination)



4. Confusion matrix, F1 score, ROC curve – ROC curve



- Class0 – Class2 - Class1 순서로 잘 분류한다는 의미
- ROC 는 그래프이기 때문에, 모델의 정확도를 하나의 숫자로 나타내기 어려움
- AUC(Area Under Curve), 즉 그래프 아래 면적을 이용하여 정확도를 측정

카파 통계

- 우연히 정확한 예측을 할 확률 (0과 1 사이의 값)
- 카테고리 정보에 대한 2명의 평가자의 일치도 측정하는 통계적 지표

$$k = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

$\Pr(a)$: 예측 정확도

$\Pr(e)$: $P(\text{실제 햄}) * P(\text{예측 햄}) + P(\text{실제 스팸}) * P(\text{예측 스팸})$

(우연으로 예측된 결과와 실제 결과가 일치할 확률)

- 형편없는 일치도 (Poor agreement) 0.20 이하
- 적당한 일치도 (Fair agreement) 0.20 ~ 0.40
- 보통의 일치도 (Moderate agreement) 0.40 ~ 0.60
- 괜찮은 일치도 (Good agreement) 0.60 ~ 0.80
- 훌륭한 일치도 (Very Good agreement) 0.80 ~ 1.00

카파 통계

- 데이터에 의존적이다.

예시 1

pred		
act	A	B
A	45	5
B	5	45

Accuracy : 0.9
 95% CI : (0.8238, 0.951)
 No Information Rate : 0.5
 P-Value [Acc > NIR] : <2e-16

Kappa : 0.8

예시 2

pred		
act	A	B
A	85	5
B	5	5

Accuracy : 0.9
 95% CI : (0.8238, 0.951)
 No Information Rate : 0.9
 P-Value [Acc > NIR] : 0.5832

Kappa : 0.4444

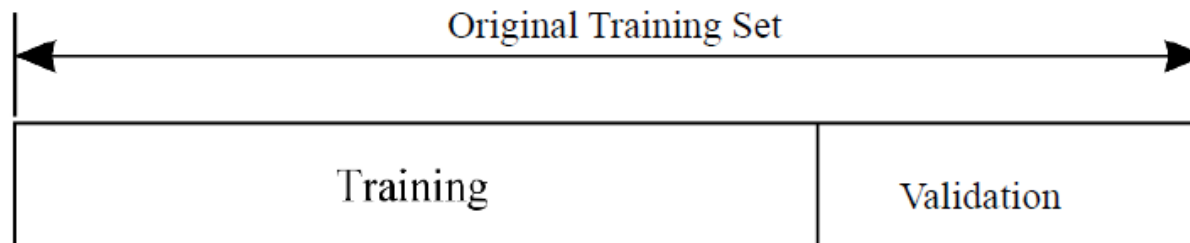
정확도는 0.9로 같은데 카파 값은 크게 차이가 난다.

평가 통계량은 하나로는 완벽하지 않음 $\pi\pi$

5. Cross validation(교차검증)

■ 교차 검증(Cross validation, CV)

: 주어진 데이터를 **일부는 학습을 시켜** 모델을 만드는데 사용하고,
일부는 모델을 검증(학습하지 않은 데이터)하는데 사용하는 것



쉽게 말해,

주어진 데이터를 가지고 반복 측정하여 더 정확한 prediction을 하기 위한 기법

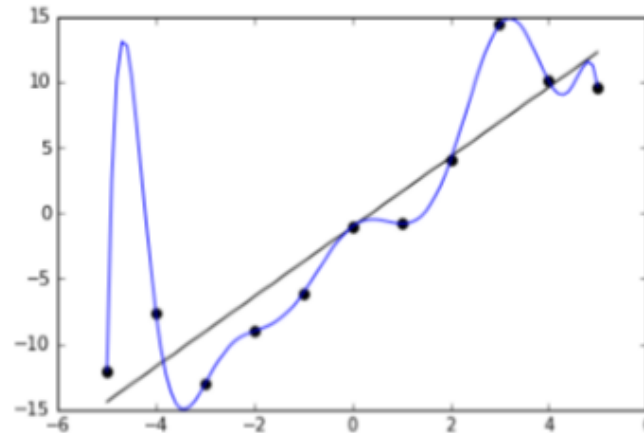
5. Cross validation(교차검증)

왜 교차 검증을 하는가?

: **과적합**을 피하기 위해

** 과적합(overfitting)

1. 학습 데이터에 대해서 지나치게 잘 학습된 상태
2. 학습 데이터에 대해서 높은 성능을 보이더라도,
학습되지 않는 데이터에 대해 좋지 않은 성능을 보일 수 있다.



5. Cross validation(교차검증) - Exhaustive VS Non-Exhaustive

1) Exhaustive CV

: 주어진 데이터를 training 과 validation set으로 나누는데
기준에 따라 가능한 모든 방법으로 나누고 반복 측정

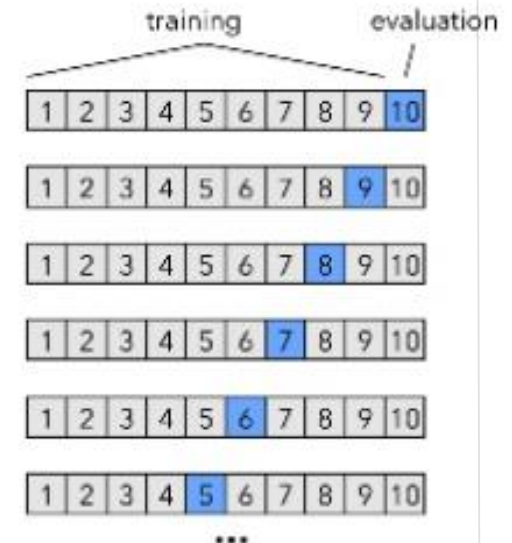
* Leave-p-out CV

: p개의 관측치를 validation으로 두고, 나머지는 training set으로 둬

만약, 총 관측치가 n개 라면, $\binom{N}{p}$ 만큼의 반복측정을 함
(n이 너무 커진다면 사실상 연산은 어려워 짐)

* Leave-one-out CV (LOOCV) : p=1인 경우

LOOCV/leave-one-out cross-validation



5. Cross validation(교차검증) - Exhaustive VS Non-Exhaustive

2) Non-Exhaustive CV

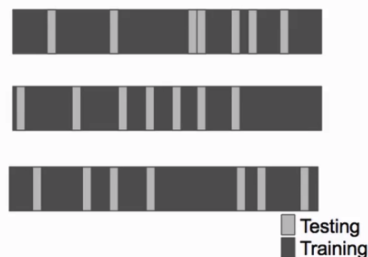
* Repeated random sub-sampling validation

: 랜덤하게 training set과 validation set을 나누는 방법

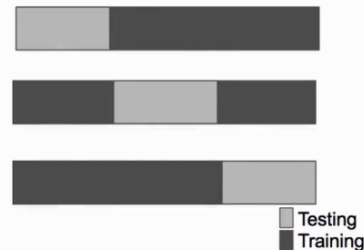
* **k-fold CV**

: 전체 데이터를 랜덤하게 나누어 K 등분을 하고,
K개 중 1번 째 데이터를 test 셋으로, 나머지 전부를 training 셋으로 사용

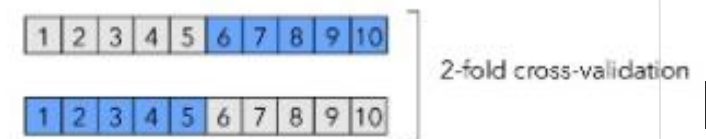
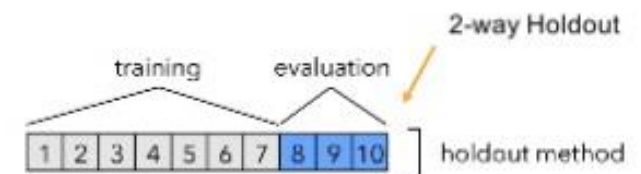
Random subsampling



K-fold



홀드아웃 Holdout



5. Cross validation(교차검증)

LOOCV vs K-fold CV

*분산	LOOCV > K-fold CV ($K < n$)
**계산 시간	LOOCV > K-fold CV

* $k < n$ 인 k-fold에서는 training set사이에 겹치는 부분이 적어,
상관성이 작음 : 상대적으로 상관성이 낮을수록 분산이 작아짐

** LOOCV는 하나의 샘플을 N번씩 반복하므로 계산량이 많다.

5. Cross validation(교차검증) - k-fold cross validation

k-fold cross validation (k-겹 교차검증)

: 교차검증 중 제일 좋다고 알려짐

**** 장점**

1. 데이터에서 많은 부분을 데이터 모델로 학습 가능
2. 최적의 비율로 test data를 나눌 수 있음
3. 학습과 검증을 여러 번 할 수 있음
4. 적은 data set일 경우 사용 가능

5. Cross validation(교차검증) - k-fold cross validation

(ex) K = 3 인 경우,

1. 전체 데이터를 3개로 나눈다

sub 1	sub 2	sub 3
-------	-------	-------

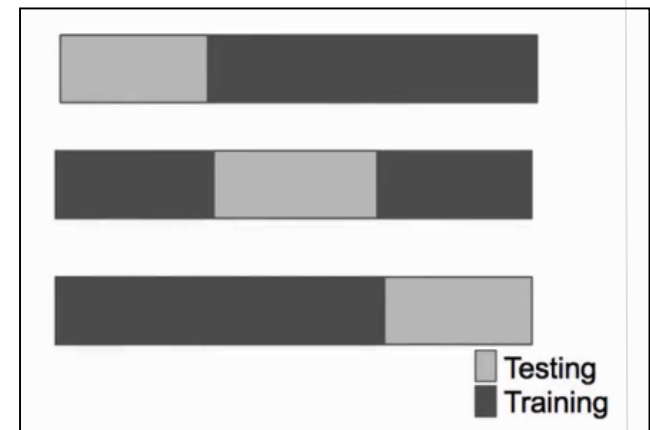
2. ① train data(**sub2 + sub3**) & test data(sub1)로 구분
② train data로 model을 만들고, test data로 검증을 한 후,
error 기록

*** error대신 정확도, ROC커브면적, RMSE 등을 기록할 수 있다

5. Cross validation(교차검증) - k-fold cross validation

3. ① train data(**sub1 + sub3**) & test data(sub2)로 구분
② train data로 model을 만들고, test data로 검증을 한 후, error 기록
4. ① train data(**sub1 + sub2**) & test data(sub3)로 구분
② train data로 model을 만들고, test data로 검증을 한 후, error 기록

*** 3개의 error의 평균을 구하여 지표로 사용



5. Cross validation(교차검증) - k-fold cross validation

5. 여러 모델의 지표를 비교하여 최적의 모델을 선택
6. 최적의 모델을 찾으면 전체 데이터를 가지고 model을 만들어 사용

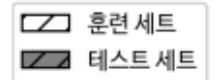
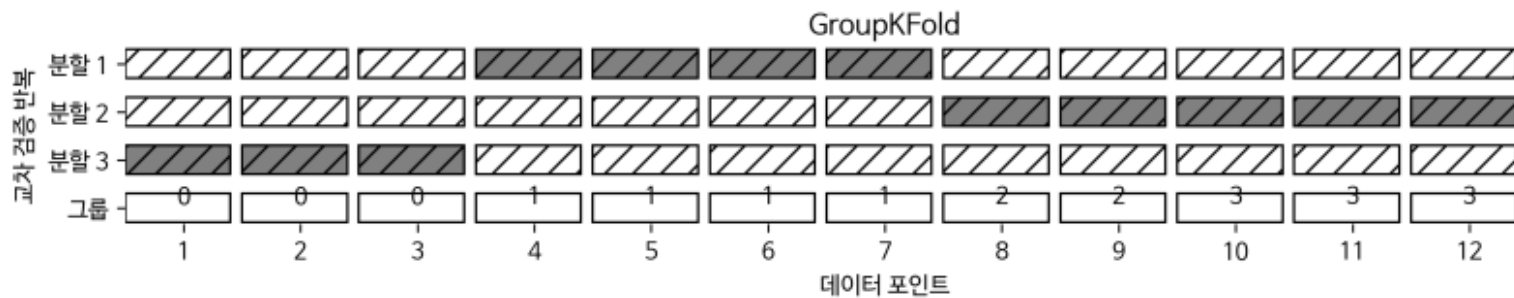
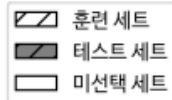
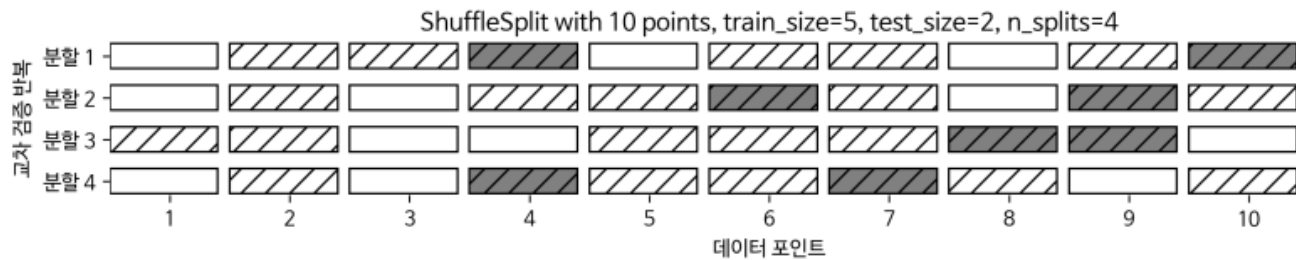
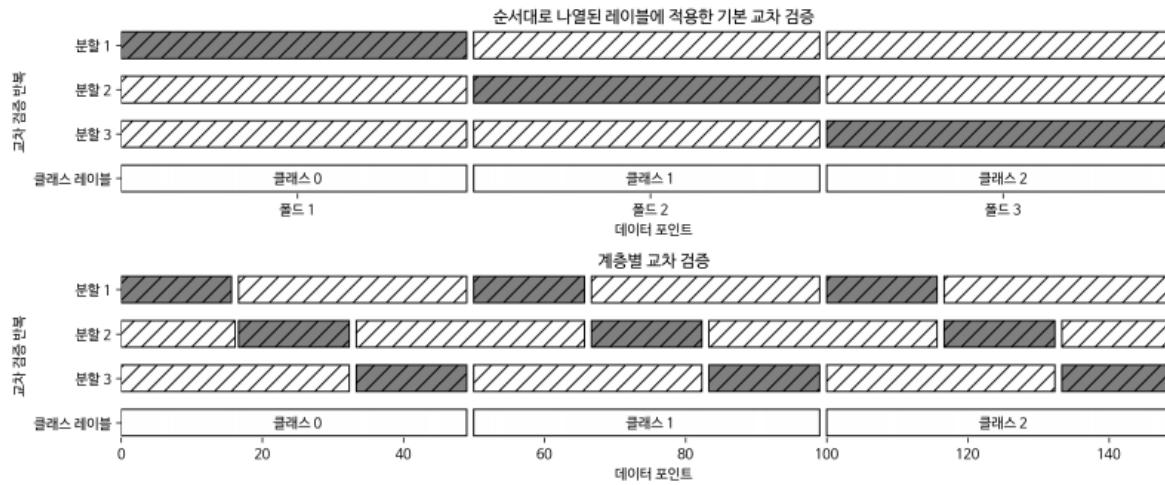
**** k값의 설정 : 보통 k는 10을 많이 사용**

5. Cross validation(교차검증) - k-fold cross validation

K-FOLD 교차 검증 의 문제점

검증 데이터를 반복 사용한다.

검증 데이터 역시 다른 훈련 데이터에 지나지 않게 되어
실제성과 다르게 더 좋게 나올 가능성이 있다.



Bootstrap Sampling?

큰 데이터 특성을 추정하기 위해 데이터의 무작위 샘플을 사용하는 통계적 기법.
무작위 샘플링을 한다.(중복 허용)

How to..?

1. N 개의 샘플을 가진 X 에서 pN 개의 샘플을 임의로 복원추출방식으로 뽑아 훈련데이터를 생성한다.
2. 샘플의 집합으로 성능을 측정하며 이 과정을 독립적으로 r 번 수행하고 그 결과를 평균한 값을 최종 성능으로 취한다.

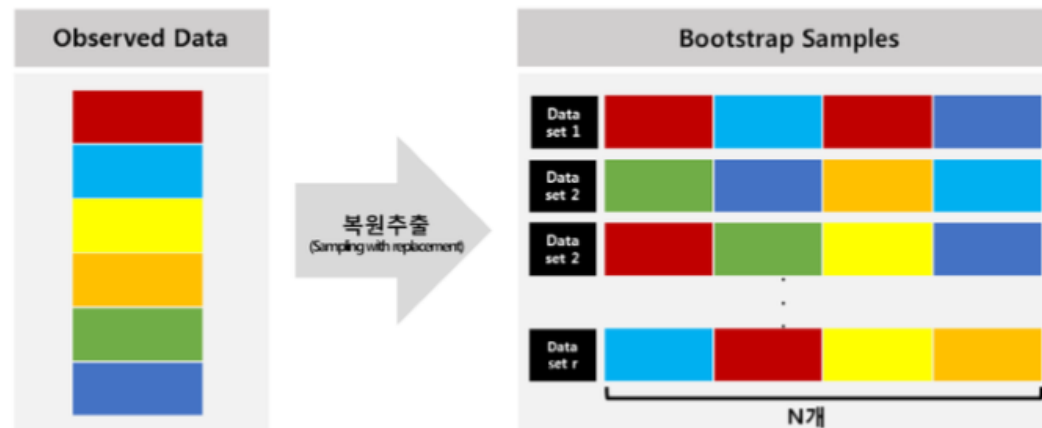
Bootstrap Sampling의 장단점

장점: 매우 작은 데이터에 적합하다.

단점: 63.2%의 훈련 데이터로 훈련된 모델이기 때문에 좀 더 큰 데이터로 훈련된 모델보다 수행이 좋지 않다.

부트스트랩의 성능 추정은 모델을 전체 데이터로 훈련한 것보다 상당히 낮음

Bootstrap Sampling



(교차검증과의 차이점)

교차 검증은 데이터를 구별된 분할로 나누고, 각 예제는 각 분할에 하나만 있지만

부트스트랩은 복원 추출을 통해 예제를 여러 번 선택할 수 있게 한다.

*복원 추출법을 사용해 훈련 데이터에 인스턴스가 들어갈 확률은 63.2%.

각 데이터가 부트스트랩 표본으로 추출될 확률이 $1 - (1 - \frac{1}{N})^N$ 이기 때문에 $1 - e^{-1} = 0.632$

매개변수 튜닝

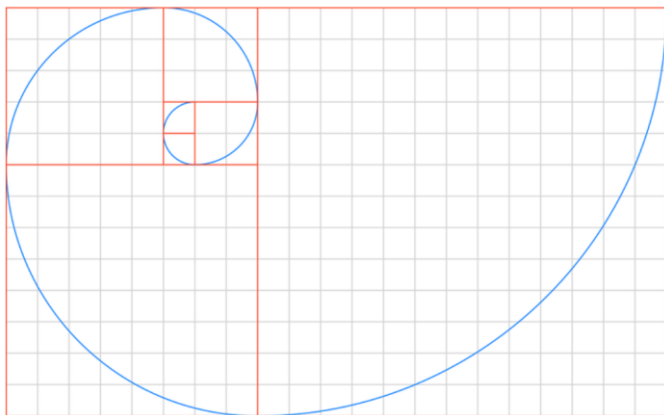
매개변수를 튜닝하여 모델의 일반화 성능(교차 검증)을 증가시킵니다.

Scikit-Learn에는 GridSearchCV와 RandomizedSearchCV가 있습니다.

RBF 커널 SVM의 여러가지 매개변수 조합을 테스트합니다.

	C = 0.001	C = 0.01	...	C = 10
gamma=0.001	SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	...	SVC(C=10, gamma=0.001)
gamma=0.01	SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	...	SVC(C=10, gamma=0.01)
...
gamma=100	SVC(C=0.001, gamma=100)	SVC(C=0.01, gamma=100)	...	SVC(C=10, gamma=100)

6. Grid search (그리드 서치)



그리드 서치란?

: 모델 튜닝을 수행할 때 **하이퍼파라미터** 값들의 좋은 조합을 찾아 낼 때까지, 수동으로 하이퍼파라미터를 사용.

어떤 하이퍼파라미터를 실험하고 싶은지, 어떤 값들을 시험해 보아야 하는지를 알려주는 것

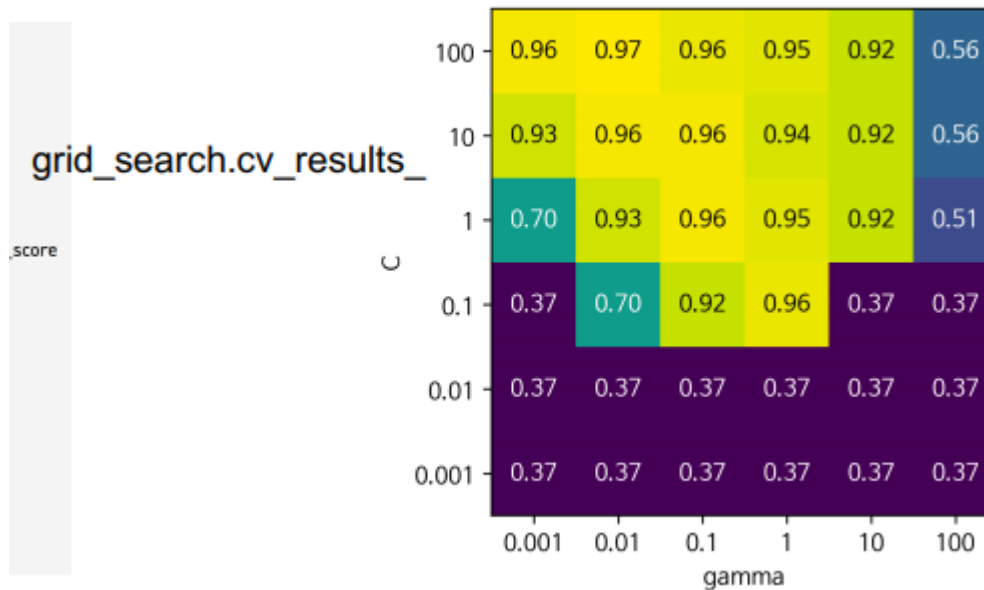
*하이퍼파라미터(초매개변수)란?

신경망을 트레이닝시켜 정확도를 높일 때 실험이나 경험을 통해서 결정하는 값

예) K-nn의 k값

RBF 커널 SVM의 여러가지 매개변수 조합을 테스트합니다.

	C = 0.001	C = 0.01	...	C = 10
gamma=0.001	SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	...	SVC(C=10, gamma=0.001)
gamma=0.01	SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	...	SVC(C=10, gamma=0.01)
...
gamma=100	SVC(C=0.001, gamma=100)	SVC(C=0.01, gamma=100)	...	SVC(C=10, gamma=100)



검증 세트 validation set

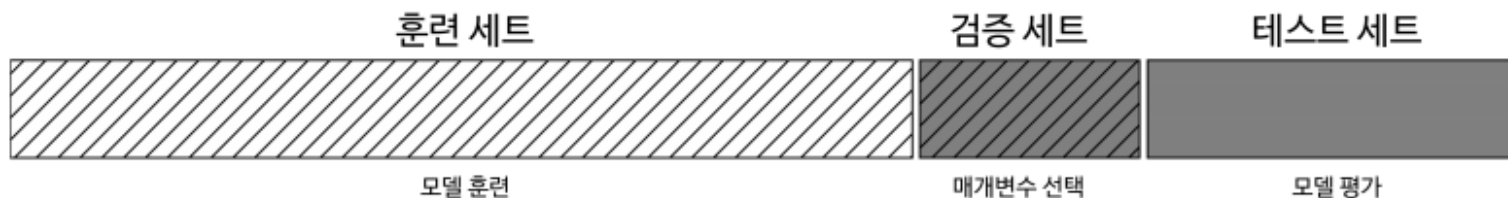
최고 점수: 0.97

최적 파라미터: {'gamma': 0.001, 'C': 100}

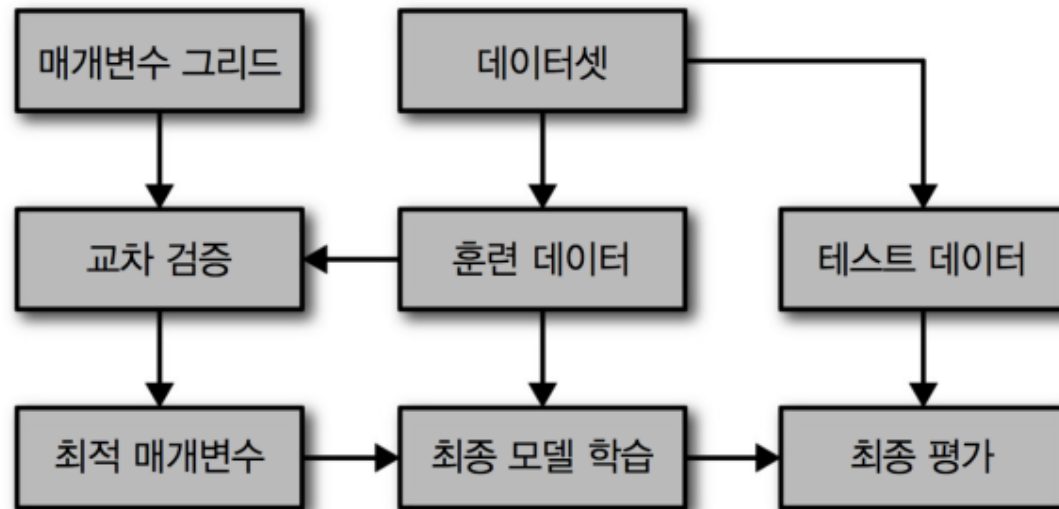
테스트 세트로 **여러가지** 매개변수 조합에 대해 평가했다면 이 모델이 새로운 데이터도 동일한 성능을 낸다고 생각하는 것은 매우 낙관적인 추정입니다.

최종 평가를 위해서는 독립된 데이터 세트가 필요합니다.

검증 세트 validation set 혹은 개발 세트 dev set



매개변수 탐색의 전체 과정



매개변수 그리드 > 교차검증 > 최적 매개변수 > 최종평가

중요한 주의 사항

교차 검증을 해야 합니다.

훈련 데이터: 모델 학습

검증 데이터: 모델과 매개 변수 선택

테스트 데이터: 모델 평가 (마지막에 딱 한번만!)



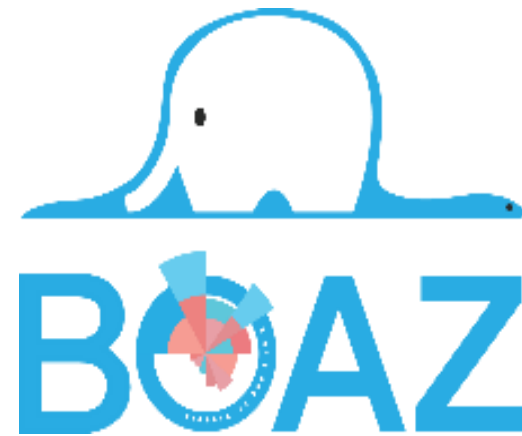
그리드서치

모델 선택과 평가에 적절한 지표를 사용합니다.

높은 정확도를 가진 모델이 아니라 비즈니스 목표에 맞는 모델이 되어야 합니다.

현실에서는 불균형한 데이터셋이 아주 많습니다.

거짓 양성(FP)과 거짓 음성(FN)이 큰 영향을 미치므로 올바른 평가 지표 선택 필요



감사합니다