머신러닝

데이터를 이용해서 명시적으로 정의되지 않은 패턴을 컴퓨터로 학습하여 결과를 만들어내는 학문 분야.

| Types | Tasks | Algorithms |
|---|---|---|
| **지도학습 (Supervised Learning)** | 분류 (Classification) | • KNN : k Nearest Neighbor<br>• SVM : Support Vector Machine<br>• Decision Tree (의사결정 나무)<br>• Logistic Regression |
| | 예측 (Prediction) | • Linear Regression (선형 회귀) |
| **비지도학습 (Unsupervised Learning)** | 군집 (Clustering) | • K-Means Clustering<br>• DBSCAN Clustering<br>• Hierarchical Clustering (계층형 군집) |
| **강화학습 (Reinforcement Learning)** | | • MDP : Markov Decision Process |

머신러닝의 분류

1) 지도학습

주어진 데어터와 레이블(정답)을 이용해서 미지의 상태나 값을 예측(regression), 분류(classfication)하는 학습 방법

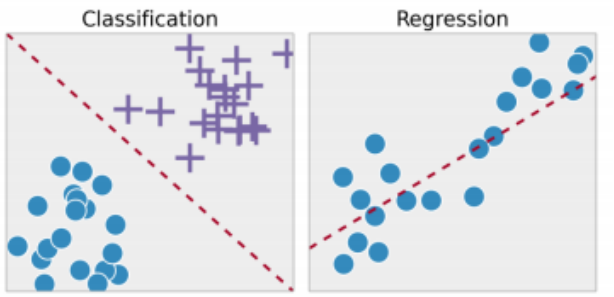
Ex) 과거 주식시장을 보고 내일 주식시장 변화 예측

문서에 사용된 단어 보고 카테고리 분류

사용자가 구매한 상품 토대로 다음에 구입할 상품 예측

▪ **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.<br>예) 스팸메일 필터 | 결과 값이 어떠한 값도 나올 수 있다.<br>예) 주가 분석 예측 |




2) 비지도학습

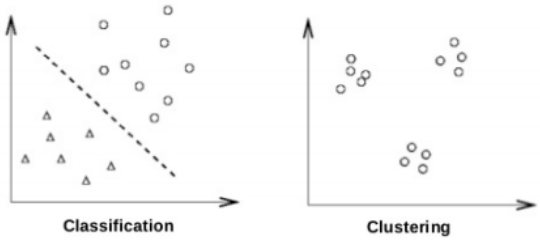
데이터 자체에서 유용한 패턴을 찾아내는 학습 방법

Ex) 비슷한 데이터끼리 묶는 군집화(clustering)

데이터에서 이상한 점을 찾아내는 이상 검출

▪ **분류와 군집의 비교**

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




** 지도학습 vs 비지도학습

➔ 데이터가 주어졌을 때 특정 값을 계산하는 함수, 시스템구축 vs

데이터의 성질 직접적으로 추측, 패턴 추출

3) 강화학습

기계가 환경과의 상호작용(선택과 피드백의 반복)을 통해 장기적으로 얻는 이득을 최대화하도록 하는 학습방법

Ex) 알파고

Knn(K-Nearest Neighbors)

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

**- 머신러닝 – 지도학습 - 분류**

-입력 값과 K개의 가까운 점들이 있다고 생각하고 그 점들이 어떤 라벨과 가장 비슷한 지를 판단하는 것입니다. 비슷하다는 것의 정의는 역시 오차(거리)로 판단합니다.

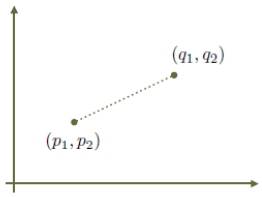
Knn기법 예시는 ppt을 참고하세요

1) 거리 계산

유클리드 거리: 가장 짧은 직접적인 경로를 나타내는 '일직선으로' 측정

Euclidean distance
2-dimension

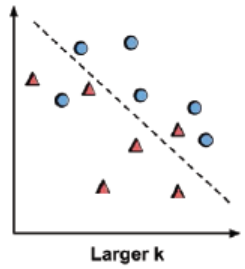
$$D(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

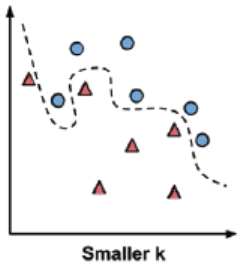



- N-dimension

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

2) 적당한 k의 선택




보통 k는 3과 10 사이에서 결정

일반적으로 훈련 데이터의 개수에 제곱근으로 설정

ex) 데이터수 =15개 15의 제곱근은 3.87 => k=4

KNN 알고리즘의 한계 – 게으른 학습과 속도

- 학습과 모델을 만드는 과정(추상화 & 일반화 과정) 생략.

- kNN 알고리즘은 어떤 것도 학습하지 않고 훈련 데이터를 그대로 저장할 뿐이다.

- 훈련 데이터에 심한 의존

Clustering(군집화)

- 머신러닝 – 비지도학습 – 클러스터링(군집화)

- 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

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-medoids0

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

K-means Clustering

1) 비계층적 방법(Nonhierarchical method) 군집의 개수 k를 미리 정해두어서, 그 군집의 수에 맞게 군집화 하는 알고리즘

2) 데이터와 데이터가 속한 군집의 중심점과의 거리의 제곱합을 최소화 시키며 k개의 군집을 구함

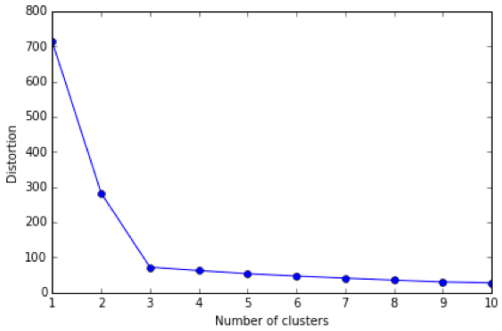
3) 반복연산으로 제일 좋은 군집을 찾는 방법

알고리즘 수행 절차는 ppt을 참고하세요!

K값 설정하는 법

1) 엘보우 기법(Elbow Method

K 값의 변화에 따른 SSE(k값에 대해 중심점과 각 데이터 포인트 사이의 거리의 제곱 합)를 관찰, 왜곡이 가장 빨리 증가하는 시점의 k를 식별




2) 실루엣 분석

클러스터링의 성능자체를 수량화 하는 것

군집 내의 샘플이 얼마나 결합력 있게 그룹핑 되었는지를 수량화하여 평가한다.

실루엣 값은 한 클러스터 안의 데이터들이 다른 클러스터와 비교해서 얼마나 비슷한가를 나타낸다.

1. 클러스터 안의 거리가 짧을 수록 좋고(cohesion), 다른 클러스터와의 거리는 멀수록 좋다(separation)

2. 실루엣은 -1 부터 1사이의 값을 가진다. (실루엣이 1일수록 잘 부합하는 거고, -1일수록 필요없는 데이터)

3. 높을 수록 좋다.

- 구하는 법

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

어떤 클러스터링 알고리즘이든지 상관없다.

* i번째 데이터 포인트에 대해서,

1. a(i) : 같은 클러스터 안에 있는 다른 데이터 포인트와의 평균거리(dissimilarity)

a(i)는 i번째 데이터 포인트가 클러스터에 얼마나 잘 맞는지 측정한다.

낮을수록 더 잘 속해있는 것

2. b(i)는 i가 속하지 않은 다른 클러스터와의 평균 거리 중 가장 작은 거리이다.

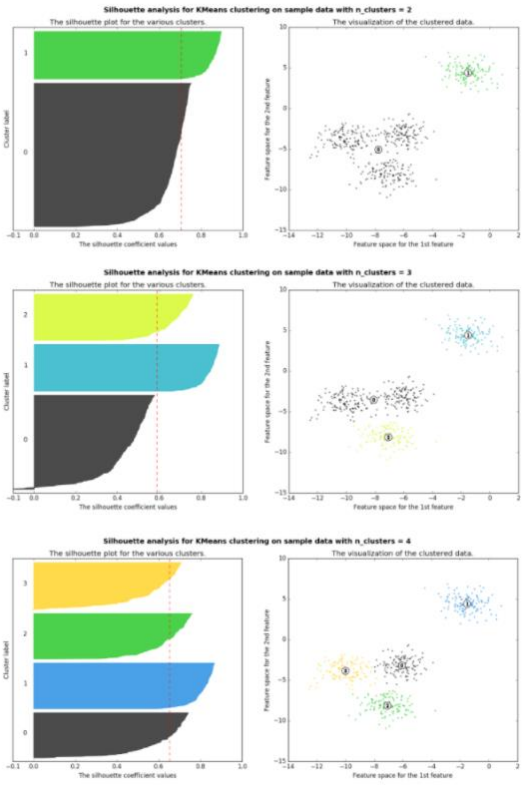
가장 작은 거리라는 건, 내가 속하지 않은 가장 가까운 클러스터, 즉 이웃 클러스터(neighbouring cluster)라는 뜻

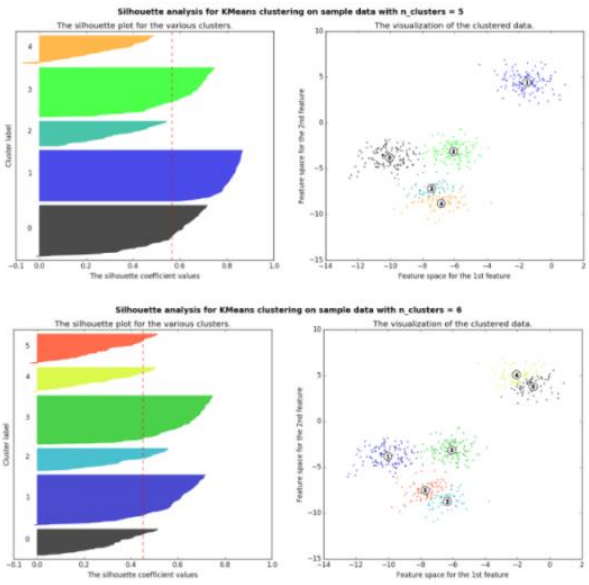
값 해석

1. b가 a보다 많이 클수록 1에 가까워지고 좋다.

2. 0이면 지금 클러스터나 이웃 클러스터나 어디 있든 상관 없다.

-실루엣 그래프 해석 법




- 빨간 점선은 score의 평균

- n_cluster = 2일 경우, 한쪽 클러스터에 데이터가  몰려있다. (0에 더 몰려있다.)

0번 클러스터는 여기에 있어도 되나 싶은 애들이 많이 들어 있다. 0에 너무 몰려있으니까 K를 1 늘리자!

그래도 여전히 0번 클러스터에 필요없는 애들이 많다. 0번 클러스터에 데이터가 많이 몰려있다.

- 꺾이는 부분이 평균보다 낮다 => 필요없는 애들이 더 몰려있다.

꺾이는 부분이 평균보다 높다 => 더 뭉쳐있다.

K-means clustering 장/단점

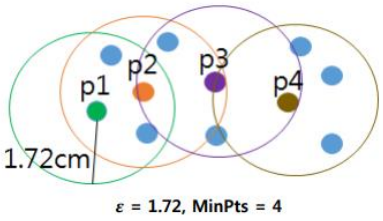
- **장/단점**

| 장점 | 단점 |
| --- | --- |
| ▪ 직관적 알고리즘<br>▪ 데이터 계산이 빠르다.<br>▪ 볼록한 구형의 데이터 세트는 잘 적용 | ▪ 오목한 형태의 군집 모델은 적용 어렵다.<br>▪ 동떨어져 있는 데이터나 노이즈에 민감<br>▪ 사전에 클러스터 수 잘못 지정하면 문제 발생 |

2) 밀도 기반 클러스터링

DBSCAN: Density Based Spatial Clustering of Application with Noise

밀도 있게 연결돼 있는 데이터 집합은 동일한 Cluster




- 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 Noise 로 정의
- Cluster 구성 후 이웃 점을 차례로 방문하면서 Core인지를 판단 (p1 → p2 → p3 → p4, 즉 p1과 p4는 같은 Cluster이다)

  - 클러스터링 과정 ppt의 gif 참고하세요

밀도 기반 클러스터링 장/단점

- **장/단점**

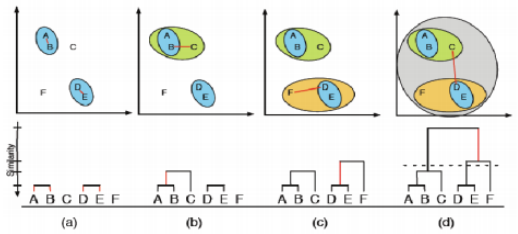
| 장점 | 단점 |
| --- | --- |
| ▪ 노이즈 식별에 강하다<br>▪ 군집의 수를 미리 정할 필요가 없다. | ▪ 밀도 반경(ε)과 최소 이웃 수(MinPts)가 민감하게 작용<br>▪ Cluster별 밀도가 서로 다른 경우 적용 어려움 |

3) 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 Cluster-로 합쳐 나간다

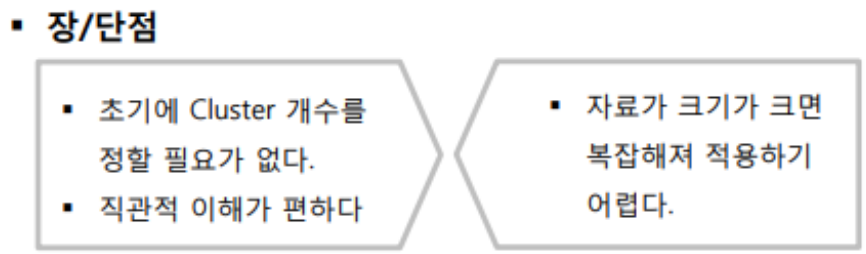
- 계층적 군집의 알고리즘




- **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.

- 클러스터링 과정 ppt의 gif 참고하세요

- 계층적 군집 의 장단점




* 모델 성능 평가(Evaluating Model)

-혼돈행렬 만들어서 여러가지 측정 지표(정확도, precision, recall)

-평가 지표(Roc curve, 통계량, 카파통계량)

**분류를 위한 성능 측정**

하나의 범주에 대단히 많은 데이터가 속한 이런 범주 불균형 문제의 결과로 예측 비율을 총 예측 개수로 나눠 정확도를 측정하는 것이 유용하지 않을 수 있다. 분류기 성능에 대한 최적의 측정 방법은 분류기가 의도한 목적에 부합하는가다.

- 분류기를 평가하기 위해 사용되는 세 가지 종류의 데이터가 있다.

1. 실제 범주 값 : 테스트 데이터의 목적 속성

2. 예측된 범주 값 : 모델을 통해 얻은 속성

3. 예측에 대한 추정된 확률 : 내부 예측 확률

* 혼돈 매트릭스 심층 학습

- 혼돈 매트릭스 : 예측 값이 실제 값과 일치하는지 분류하는 표

**혼동 행렬**

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

*** 관심범주의 예측 관계 판단**

| | | Diagnosis | |
|---|---|---|---|
| | | No cancer | Cancer |
| True state | No cancer | *TN* | *FP* |
| | Cancer | *FN* | *TP* |

- **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

*** 성능을 측정하기 위한 혼돈 매트릭스 사용**

예측 정확도 : $$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

오차 비율 : $$\text{error rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = 1 - \text{accuracy}$$

*** 정확도를 넘어 : 다른 성능 측정**

1. 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(실제 햄)*P(예측 햄) + P(실제 스팸)*P(예측 스팸)

**(우연으로 예측된 결과와 실제 결과가 일치할 확률)**

2. 재현율 : 실제 긍정일 때 정확하게 예측한 비율

- 결과가 어떻게 마무리되는지 측정

- 높은 재현율을 가진 모델은 넓은 폭을 의미하는 긍정 예제의 비율이 높다

- ex) 병이 걸린 환자에게 얼마나 정확하게 진단했는가

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

3. 정밀도(긍정적인 예측값) :

긍정으로 예측했을 때의 실제 긍정의 비율

Ex) 스팸메일이라고 판단했을 때 얼마나 제대로 판별했는가?

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

4. F측정

: 정밀도와 재현율을 하나로 합하는 모델의 성능 측정 (조합평균 사용)

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

- 같은 가중치로 정밀도와 재현율을 지정했다고 가정

*** 성능 균형의 시각화**

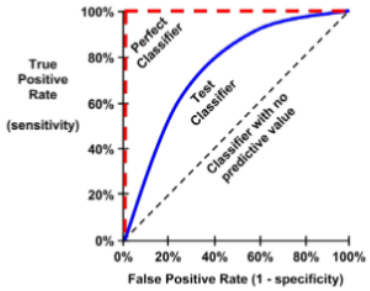
1. ROC 커브 : 거짓 긍정을 배제하고 참 긍정을 식별하는 동안의 균형을 살펴보는 데 사용

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)




- x축 : 1-특이도 y축 : 민감도

** 민감도와 특이도

- 결정을 내릴 때 전반적으로 너무 긍정이거나 너무 부정도 아닌 균형이 필요하다. 일부 측정 방법으로 이를 잡아낼 수 있는데 이 때 민감도, 특이도를 사용.

민감도 (참 긍정) : 정확하게 분류한 긍정의 비율을 측정 $\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$

특이도 (참 부정) : 정확하게 분류한 부정의 비율 측정 $\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$

- 민감도 특이도 모두 0과 1 사이이며 1에 가까울수록 바람직

- 완벽한 분류기에 가까울수록 도표의 상단 좌측 공간에 위치하게 된다.

2. AUC : 2차원 사각형으로 ROC 다이어그램을 다루며 ROC커브의 총면적 계산

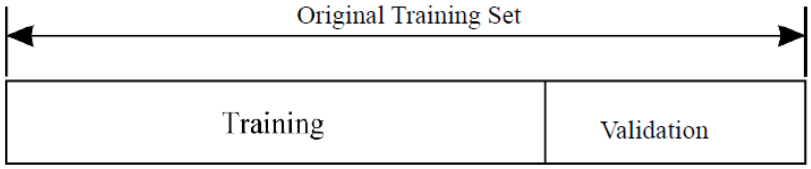
- 그래프 설명은 ppt 참고하세요

모델 평가를 위한 검증법

- **교차 검증**(Cross validation, CV)

  주어진 데이터를 일부는 학습을 시켜 모델을 만드는데 사용하고,

  일부는 모델을 검증(학습하지 않은 데이터)하는데 사용하는 것




  왜 **교차 검증**을 하는가

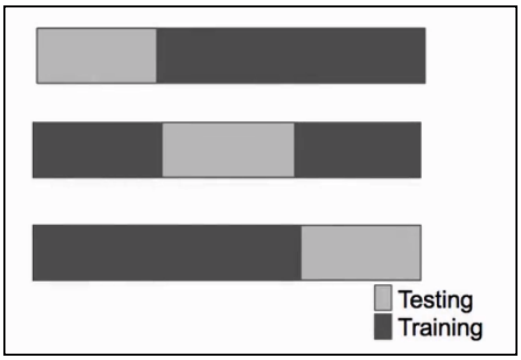
  - **과적합, 즉 학습 데이터에 대해서 높은 성능을 보이더라도,**

    **학습되지 않는 데이터에 대해 좋지 않은 성능을 보이는 것을** 피하기 위해

- K-Fold CV (k-겹 교차검증)

  전체 데이터를 랜덤하게 나누어 K 등분을 하고,

  K개 중 1번 째 데이터를 test 셋으로, 나머지 전부를 training 셋으로 사용




  - 검증 과정

  - **(ex) K =3 인 경우,**

    1. 전체 데이터를 3개로 나눈다

    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

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. 여러 모델의 지표를 비교하여 최적의 모델을 선택

6. 최적의 모델을 찾으면 전체 데이터를 가지고 model을 만들어 사용

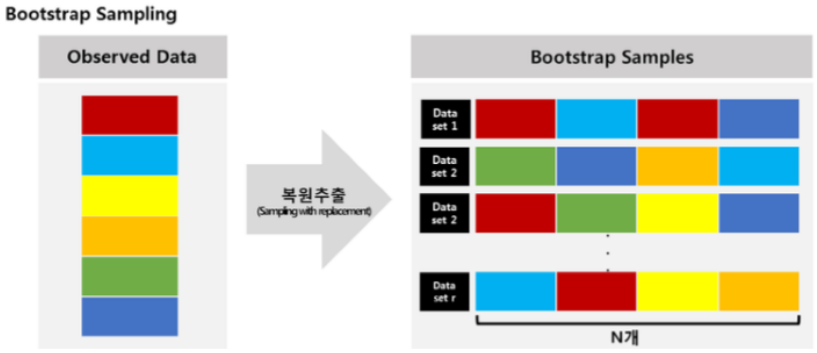
- K-Fold 교차검증의 문제점

1. 검증 데이터를 반복 사용
2. 검증 데이터 역시 다른 훈련데이터에 지나지 않아 실제 성능과 다르게 더 좋게 나올 가능성 있음

■ Bootstrap sampling

무작위 샘플링

큰 데이터의 특성을 추정하기 위해 데이터의 무작위 샘플을 사용하는 통계적기법




(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 $1-e^{-1}=0.632$

모델 튜닝(최적화)

유망한 모델을 가졌다고 가정했을 때, 모델을 튜닝하고 최적화 시키는 것이 필요

- 임의로 지정해 놓은 하이퍼파라미터 최적화 혹은

  다양한 모델을 활용하여 결과를 조합(앙상블)

  **하이퍼파라미터란?

  학습 모델을 구축 할 때 모델을 튜닝하기 위한 알고리즘의 옵션들

  하이퍼파라미터 최적화 방법

1. 사람이 직접 지정(경험)
2. 그리드 탐색(Grid Search)
3. 무작위 탐색(Random Search for Hyper-Parameter Optimization)
4. 베이지안 최적화와 같은 모형기반 방법

■ 그리드 탐색(Grid Search)

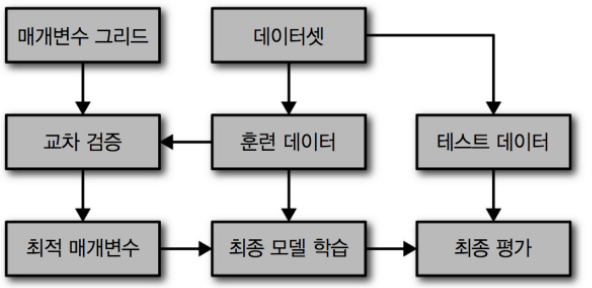
하이퍼파라미터 공간에서 수동으로 지정한 하위 집합을 단순하게 모든 조합을 다 탐색하는 것

Ex)

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) |

## 매개변수 탐색의 전체 과정




매개변수를 그리드를 한 후 훈련세트를 이용해 교차 검증을 한 후 최적의 매개변수를 찾습니다. 그 후 최종모델 학습을 시킨 다음 테스트 데이터 세트를 통해 최종평가를 내립니다.

훈련 데이터에서도 동일한 성능을 낸다고 추정할 수 없기에 최종평가가 필요합니다.

최종평가를 위해서는 독립된 데이터 세트가 필요합니다.

## 검증 세트validation 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 세트validation set 혹은 개발 세트dev set