• 데이터마이닝

- 개요 : 데이터마이닝은 대용량 데이터에서 의미있는 패턴을 파악하거나 예측하여 의사결정에 활용하는 방법이다.

- 통계분석과의 차이점 : 통계분석은 가설이나 가정에 따른 분석이나 검증을 하지만, 데이터마이닝은 다양한 수리 알고리즘을 이용해 데이터베이스의 데이터로부터 의미있는 정보를 찾아내는 방법을 통칭한다.

- 사용분야

1. 병원에서 환자 데이터를 이용해서 해당 환자에게 발생 가능성이 높은 병을 예측
2. 기존 환자가 응급실에 왔을 때, 어떤 조치를 먼저 해야 하는지를 결정(해당 환자의 전 진료 내역과 전 진료 내역과 비슷한 환자들의 데이터를 활용)
3. 고객 데이터를 이용해 해당 고객의 우량/불량을 예측해 대출적격 여부 판단
4. 세관 검사에서 입국자의 이력과 데이터를 이용해 관세물품 반입 여부를 예측

- 데이터마이닝의 최근 환경

1. 데이터마이닝 도구가 다양하고 체계화되어 있고, 알고리즘에 대한 깊은 이해가 없어도 분석에 큰 어려움이 없다.
2. 분석 결과의 품질은 분석가의 경험과 역량에 따라 찾이가 나기 때문에, 분석 과제의 복잡성이나 중요도가 높으면 풍부한 경험을 가진 전문가에게 의뢰할 필요가 있다.
3. **국내에서 데이터마이닝이 적용된 시기는 1990년대 중반이다.**
4. **2000년대**에 비즈니스 관점에서 **데이터마이닝이 CRM의 중요한 요소로 부각**되었다.

- 데이터마이닝의 분석 방법

1. **지도학습** : 의사결정나무, 인공신경망, GLM(로지스틱 회귀분석), 선형 회귀분석, 사례기반 추론, **최근접 이웃법**(KNN : 라벨이 있는 데이터를 학습한 모델에 라벨이 없는 새로운 데이터를 입력시켜 해당 데이터의 라벨을 결정짓는 알고리즘이다. ‘K’는 새로운 데이터와 인접한 데이터들의 개수를 의미한다. 보통 이진분류에 사용하지만 다중분류에도 사용할 수 있다.)
2. **비지도학습** : 군집분석(모집단을 동질성을 지닌 그룹으로 세분화 하는 것), 연관분석, SOM

- 분석 목적에 따른 데이터마이닝 기법 분류

1. 예측 : **분류 규칙**(가장 많이 사용되는 작업으로 과거의 데이터로부터 고객특성을 찾아내어 분류모형을 만들고 이를 토대로 새로운 데이터의 결과값을 예측하는 것, ‘회귀분석, 신경망, 의사결정나무’가 이에 해당)
2. 설명 : **연관규칙**(데이터 안에 존재하는 항목간의 종속관계를 찾아내는 작업, ‘동시발생 매트릭스’가 이에 해당), **연속규칙**(연관규칙에 시관관련 데이터가 포함된 형태로 고객의 구매이력 속성이 반드시 필요함, ‘동시발생 매트릭스’가 이에 해당), **데이터 군집화**(고객 데이터들을 유사한 특성을 지닌 몇 개의 소그룹으로 분할하는 작업으로 레이블이 주어지지 않음, K-Means Clustering이 이에 해당)

- 데이터마이닝 추진단계

1. **목적 설정** : 데이터마이닝을 통해 무엇을 왜 하는지 명확한 목적을 설정한다.
2. **데이터 준비** : 다양하고 충분한 양의 데이터를 확보한다.
3. **가공** : 모델링 목적에 따라 종속 변수를 정의하고, 모델링에 적용할 수 있도록 데이터들을 가공(정제)한다.
4. **기법 적용** : 1단계(목적 설정)에 맞게 데이터마이닝 기법을 선택하여 적용한다.
5. **검증** : 데이터마이닝으로 추출된 정보를 검증한다. 검증을 통해 최적의 모델을 선정하고, 검증이 완료되면 IT부서와 협의해 상시 데이터마이닝 결과를 업무에 적용하고 보고서를 작성하여 추가수익과 투자대비성과 등으로 기대효과를 전파한다.

- 데이터마이닝을 위한 데이터 분할

1. train data : 50%
2. **validation data** : 30%, overfitting과 underfitting을 확인하고 이를 미세조정 하는데 활용한다.
3. **test data** : 20%, 최종적으로 모델의 성능을 검증(측정)하는데 활용한다.

- 모델 검증 방법

1. **hold out** : 주어진 데이터셋을 랜덤하게 두 개의 데이터로 구분(train data와 test data)하여 사용하는 방법이다.
2. **Cross validation** : 주어진 데이터셋을 k개의 하부집단으로 구분하여, k-1개의 집단을 학습용으로 나머지는 하부집단으로 검증으로 설정하여 학습한다. k번 반복 측정한 결과를 평균낸 값을 최종값으로 사용한다.

- 성과분석 (classification 모델의 성능 평가 지표)

1) 정분류율(Accuracy) ← Deep learning의 분류 모델에서 사용되는 평가지표.

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

2) 오분류율(Error Rate)

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

3) 특이도(Specificity)

$$Specificity = \frac{TN}{TN + FP}$$ (TNR : True ~~Native~~ Negative Rate)

4) 민감도(Sensitivity)

$$Sensitivity = \frac{TP}{TP + FN}$$ (TPR : True Positive Rate)

5) 정확도(Precision) ← 'p'에 초점을 맞출라

$$Precision = \frac{TP}{TP + FP}$$

6) 재현율(Recall) : 민감도와 같음

$$Recall = \frac{TP}{TP + FN}$$

7) F1 Score ← 프로그램 상에서 가장 많이 사용되는 평가지표.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

True Positive, TP: 새로운 진단 방법에 의하여 양성(질병)으로 진단되었고 실제로도 양성인 경우.
True Negative, TN: 새로운 진단 방법에 의하여 음성(정상)으로 진단되었고 실제로도 음성인 경우.
False Positive, FP: 새로운 진단 방법은 양성(질병)으로 진단하였으나 실제로는 음성인 경우.
False Negative, FN: 새로운 진단 방법은 음성(정상)으로 진단하였으나 실제로는 양성인 경우.

3.

이제 민감도를 앞에서 나온 개념들로 표현해 보자. 민감도는 "이미 정답을 알고 있는 사람 중 실제 환자 (즉, 기존에 정답으로 사용되는 진단 방법에 의하여 환자로 판명된 사람) 에 대하여 새로운 진단 방법이 환자라고 판명한 비율"이다. 다시 말하면,

민감도 = 새로운 진단 방법이 환자라고 판명한 사람 중 실제로 환자인 사람 / 실제로 환자인 사람의 수

TP TP+FN

이다. 이 때, 실제로 환자인 사람의 수는 TP+FN 인데, 왜냐 하면,

True Positive, TP: 새로운 진단 방법에 의하여 양성(질병)으로 진단되었고 실제로도 양성인 경우.
False Negative, FN: 새로운 진단 방법은 음성(정상)으로 진단하였으나 실제로는 양성인 경우.

이제 특이도를 살펴 보자. 특이도는 "이미 정답을 알고 있는 사람 중 실제로 정상 (즉, 기존에 정답으로 사용되는 진단 방법에 의하여 정상으로 판명된 사람) 에 대하여 새로운 진단 방법이 정상이라고 판명한 비율"이다. 다시 말하면,

특이도 = 새로운 진단 방법이 정상이라고 판명한 사람 중 실제로 정상인 사람 / 실제로 정상인 사람의 수

이다. 이 때, 실제로 정상인 사람의 수는 TN+FP 인데, 왜냐 하면,

True Negative, TN: 새로운 진단 방법에 의하여 음성(정상)으로 진단되었고 실제로도 음성인 경우.
False Positive, FP: 새로운 진단 방법은 양성(질병)으로 진단하였으나 실제로는 음성인 경우.

으로 정의되었기 때문이다. 즉, 실제로 음성인 사람은 새로운 진단 방법에 의해 음성(N)으로 진단되거나 양성(P)으로 진단이 될텐데 음성인 경우에는 맞게(true) 판단한 것이므로 True Negative, 양성으로 판단한 경우에는 원래는 음성인데 양성으로 잘못(false) 판단한 것이므로 False Positive 인 것이다. 따라서 실제로 음성인 사람의 수는 이 두 경우에 해당하는 수의 합인 것이다. 이제 특이도를 다르게 표현해 보면,

특이도 = TN/(TN+FP)

이 되는 것이다.

- ROC Curve

1. FPR(False Positive Rate, 1 - 특이도)값을 가로축으로, TPR(민감도)값을 세로축으로 두어 시각화한 그래프이다.
2. 2진 분류 모형의 성능을 평가하기 위해 사용된다.
3. AUROC(Area Under ROC)의 값이 크면 클수록 모형의 성능이 좋다고 평가한다.

- 이익도표

예측 모델이 positive (+1) 이라고 예측할 확률.

데이터셋의 각 데이터는 각각 예측 확률을 가진다.
① 전체 데이터를 예측 확률을 기준으로 내림차순 정렬한다.

② 맨 위에서부터 10%씩 구간을 자른다

- 전체 5000명 중에 950명이 실제로 구매
Baseline Lift = 950 / 5000 = 0.19 = 19 %

각 구간 내 총 인원

- 예측 확률 상위 10% 500명 중 435명 구매
반응률(Response) = 435 / 500 = 87 %
반응검출률(Captured Response) = 435 / 950 = 45.79 %

- 예측 확률 상위 10%의 Lift
Lift = Response / Baseline lift = 87 / 19 = 4.58
좋은 모델이라면 Lift 가 빠른 속도록 감소해야 한다.

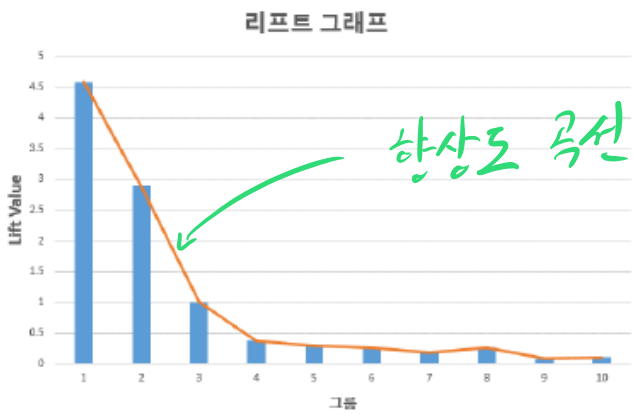
* 전체 5000 명을 10개 구간으로 500명씩 구분

등급	실구매자	Captured Response %	Response %	Lift
1	435	45.79	87	4.58
2	275	28.95	55	2.89
3	95	10	19	1
4	35	3.68	7	0.37
5	27	2.84	5.4	0.28
6	25	2.63	5	0.26
7	18	1.89	3.6	0.19
8	24	2.53	4.8	0.25
9	12	1.26	2.4	0.13
10	4	0.42	0.8	0.04
	950			

- 이익도표에서의 통계량

- %Captured Response $= \frac{\text{해당 등급에서 } y=1\text{인 빈도}}{\text{전체 자료에서 } y=1\text{인 빈도}} \times 100\%$
- %Response $= \frac{\text{해당 등급에서 } y=1\text{인 빈도}}{\text{해당 등급의 자료의 수}} \times 100\%$
- Baseline Lift $= \frac{\text{전체 자료에서 } y=1\text{인 자료의 수}}{\text{전체 자료의 수}} \times 100\%$

Lift $= \frac{\text{해당 등급의 \%Response}}{\text{Baseline Lift}}$



- 각 등급은 사후확률에 따라 매겨진 순위이므로 좋은 예측모형이라면 상위 등급에서는 더 높은 반응률, 하위등급에서는 더 낮은 반응률을 보여야 한다.

인공신경망

- 뉴런은 가중치가 있는 링크들로 연결되어 있다. 그리고 뉴런은 여러 입력 신호를 받지만 출력 신호는 오직 하나만 생성한다. 즉 입력 링크에서 여러 신호를 받아서 새로운 활성화 수준을 계산하고, 출력 링크로 출력 신호를 보낸다.

- 입력신호는 미가공 데이터(입력 데이터) 또는 다른 뉴련의 출력이 될 수 있다.

- 신경망은 가중치를 반복적으로 조정하며 학습한다.

- **퍼셉트론의 뉴런 계산 과정** : 활성화 함수(**step function[계단 함수] or sign function[부호 함수]**)를 이용해 출력을 결정하며, 입력신호의 가중치 합을 계산하여 임계값과 비교한다. **sign function**에 의해 가중치 합이 임계값보다 출력은 -1, 같거나 크면 +1을 출력한다.

- 신경망 모형 구축시 입력변수 고려사항

1. 여러 개의 연속형 변수들이 입력변수로 사용될 때는 해당 연속형 변수들의 scale 차이가 많이 나면 안된다.
2. 범주형 변수를 입력변수로 사용할 때는 해당 **범주형 변수를 dummy variable**로 변환해야 한다.

- 역전파 알고리즘은 가중치 초기값에 따라 결과가 많이 달라지므로, 가중치 초기값의 선택은 매우 중요한 문제이다.

- **일반적으로 가중치 초기값은 0 근처로 랜덤하게 선택**하므로 초기 모형은 선형모형에 가깝고, 가중치 값이 증가할수록 비선형모형이 된다.

- 학습모드

1. **온라인 학습 모드** : train data를 한 개씩 학습시킨다.
2. 확률적 학습 모드 : mini batch들을 한 개씩 학습시키되, mini batch들이 학습되는 순서는 확률적(랜덤)이다.
3. 배치 학습 모드 : train data들을 한꺼번에 학습시킨다.

- 은닉층과 은닉노드가 많으면 가중치가 많아지게 되고, 이로써 과대 적합 문제가 발생한다. 반대로 은닉층과 은닉노드가 저긍면 과소적합 문제가 발생한다.

- 은닉층 수가 하나인 신경망은 범용 근사자(universal approximator)이므로, 모든 매끄러운 함수를 근사적으로 표현할 수 있다. 그러므로 ANN에선 가능하면 은닉층은 하나로 선정한다.

- 학습 조기 종료, 가중치 감소 기법(regularization)으로 과대 적합 문제를 해결할 수 있다.

분류분석

- 분류분석 정의 : 종속변수가 범주형이고, 데이터가 어떤 그룹에 속하는지 예측하는데 사용되는 기법이다. 학습데이터마다 레이블이 주어져 있기 때문에, 지도학습에 해당한다.

- 분류분석과 예측분석 비교

1. 공통점 : 지도학습에 해당한다
2. 차이점 : 분류분석의 종속변수는 범주형이고 예측분석의 종속변수는 연속형이다.

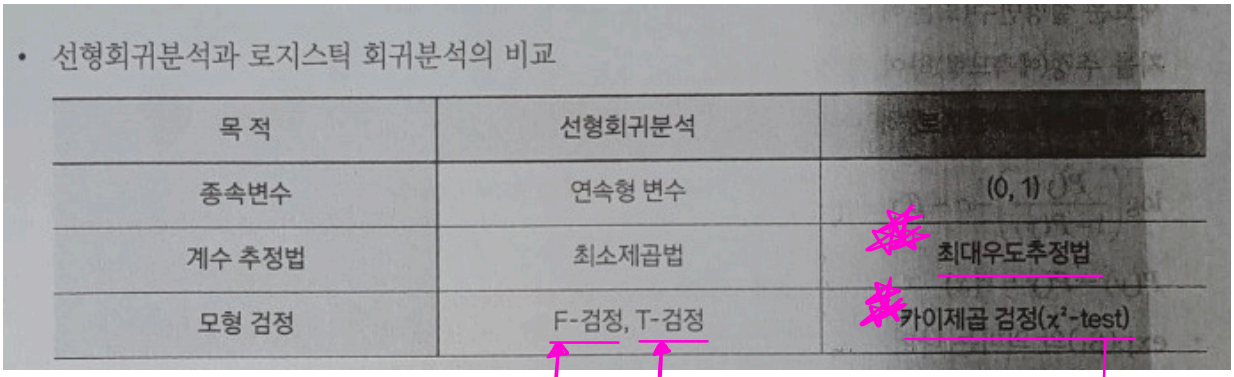
- **분류분석에 사용되는 대표적인 알고리즘** : 로지스틱 회귀분석, 의사결정나무, 베이지안 분류, 딥러닝, 서포트벡터머신, **k 최근접 이웃(k-nearest neighborhood)**

- **분류의 대표적인 모델링** : 신용평가모형(우량, 불량), 고객세분화(VVIP, VIP, GOLD, SILVER, BRONZE), 사기방지모형(사기, 정상), 이탈모형(이탈, 유지)

- 로지스틱 회귀분석

1. 종속변수가 범주형인 경우에 적용되는 회귀분석모형이다.
2. 새로운 독립변수가 주어질 때 종속변수의 각 범주에 속할 확률이 얼마인지를 추정하여, 추정 확률을 기준치에 따라 분류하는 목적(분류모형)으로 활용된다.
3. 로지스틱 회귀분석 함수 식 : **logit function = b0 + b1*x1 + b2*x2 + ...**
   or **Y = 1/(1 + exp[-(b0 + b1*x1 + b2*x2 + ...)]**

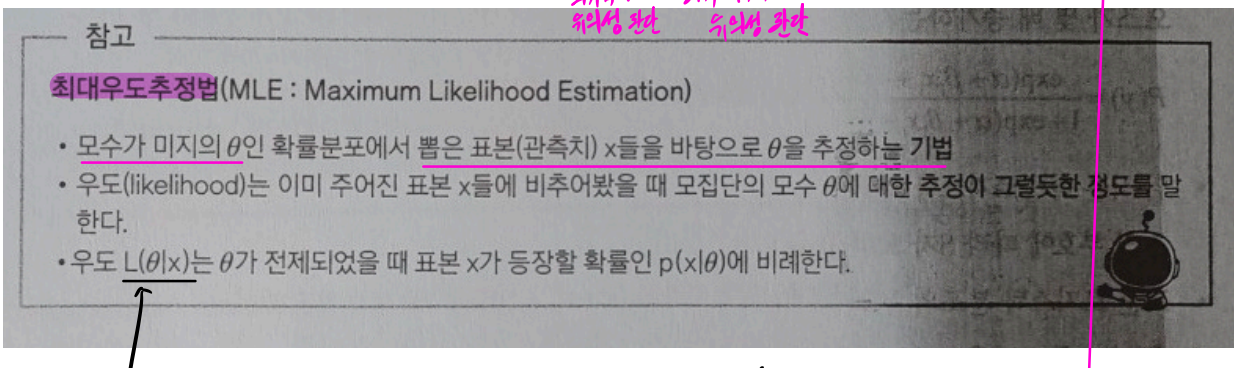
- 선형회귀분석과 로지스틱 회귀분석 비교



• 선형회귀분석과 로지스틱 회귀분석의 비교

목 적	선형회귀분석	로
종속변수	연속형 변수	(0, 1)
계수 추정법	최소제곱법	최대우도추정법
모형 검정	F-검정, T-검정	카이제곱 검정(x^2-test)

회귀식의 유의성 판단 / 회귀계수의 유의성 판단



참고

최대우도추정법(MLE : Maximum Likelihood Estimation)

- 모수가 미지의 θ인 확률분포에서 뽑은 표본(관측치) x들을 바탕으로 θ을 추정하는 기법
- 우도(likelihood)는 이미 주어진 표본 x들에 비추어봤을 때 모집단의 모수 θ에 대한 추정이 그럴듯한 정도를 말한다.
- 우도 $L(\theta|x)$는 θ가 전제되었을 때 표본 x가 등장할 확률인 $p(x|\theta)$에 비례한다.

L. likelihood function : $P(x|\theta) = \prod_{k=1}^{n} p(x_k|\theta)$

- 보통은 likelihood function을 대체하여, log likelihood function을 사용함.
- log likelihood function : $L(\theta|x) = \log P(x|\theta)$

$$= \log \prod_{k=1}^{n} p(x_k|\theta)$$

$$= \sum_{k=1}^{n} \log p(x_k|\theta)$$

- '$L(\theta|x)$'이 최대가 될 때의 'θ'을 찾으면 됨.
- '$\frac{\partial}{\partial\theta} L(\theta|x) = 0$'을 통해 '$\theta$'를 찾으면 됨.

- glm() 호출 결과 해석

1. 로지스틱 회귀모형의 회기계수는 회기계수의 변수가 한 단위 증가했을 때 log(odds)의 증가량으로 해석할 수 있다.
2.

결과를 보면 일반적인 **다중회귀분석**과 비슷한듯 다른 점이 많다. 각 변수에 대한 회귀계수를 보는 법은 똑같지만 **F검정**이 없어졌고 **이탈도deviance**라는 것이 생겼다.

이탈도라는 것은 로지스틱 회귀모형이 얼마나 데이터를 못 설명하는지에 대한 척도로 보아도 무방하다. **영이탈도Null deviance**는 어떠한 변수 없이 상수항만 있을 때의 이탈도로써 데이터가 전혀 없는 최악의 상황이라 할 수 있다. 이렇게 얻어진 **잔차이탈도Residual deviance**는 작으면 작을수록 좋고, 카이제곱분포를 따르기 때문에 카이제곱 적합도 검정을 통해 모형이 적합한지 확인할 수 있다.

↳ 모든 독립 변수가 다 추가 되었을 때의 이탈도.

의사결정나무(자세한 내용은 별도의 노트에 필기되어 있음)

- 의사결정나무는 분류함수를 의사결정 규칙으로 이뤄진 나무 모양으로 그리는 방법이다.

- 나무구조는 연속적으로 발생하는 의사결정 문제를 시각화해 의사결정이 이뤄지는 시점과 성과를 한눈에 볼 수 있게 한다. 그러므로 의사결정나무는 성능이 다른 알고리즘들보다 안 좋다고 할지라도, 고객에게 설명하기 매우 용이하다.

- 의사결정나무는 분류나무와 회귀나무로 구성되어 있다. 그러므로 의사결정나무는 supervised learning에 해당한다.

- 의사결정나무의 구성요소

1. root node : 시작되는 node
2. child node : 하나의 node로부터 분리되어 나온 2개 이상의 node들
3. parent node : 상위 node
4. **terminal node** : 맨 끝에 위치하여 자식node가 없는 node
5. internal node : root node와 terminal node를 제외한 node, 즉 parent node와 child node를 다 갖춘 node
6. **branch** : root node로부터 terminal node까지 연결된 node들
7. **depth** : root node부터 terminal node까지의 중간 node들의 수

- 예측력과 해석력

1. 기대 집단의 사람들 중 가장 많은 반응을 보일 고객의 유치방안을 예측하고자 하는 경우에는 예측력에 치중한다.
2. 신용평가에서는 심사 결과 부적격 판정이 나온 경우, 고객에게 부적격 이유를 설명해야하므로 해석력에 치중한다.

- 의사결정나무 활용

1. **세분화** : 데이터를 비슷한 특성을 갖는 몇 개의 그룹으로 분할해 그룹별 특성을 발견하는 것이다.
2. 분류 : 새로운 데이터가 어느 범주에 속하는지 분류함
3. 예측 : 자료에서 규칙을 찾아내고 이를 이용해 미래의 사건을 예측하고자 하는 경우이다.
4. **차원축소 및 변수선택** : 매우 많은 수의 예측변수 중에서 목표변수에 큰 영향을 미치는 변수들을 골라내고자 하는 경우에 사용하는 기법이다.
5. **교호작용효과의 파악** : 두 개 이상의 변수가 결합하여 목표 변수에 어떻게 영향을 주는지를 쉽게 알 수 있다.

X 의사결정나무는 '다중공선성'과 '이상값'에 민감하지 않다.

- 의사결정나무 특징(장점)

1. 결과 설명이 용이하다.
2. 모형을 쉽게 만들 수 있다.
3. 모델을 만드는 속도가 빠르다.
4. **비정상 잡음 데이터에 대해서도 민감함이 없이 분류할 수 있다.(즉 전처리를 안해도 된다.)**
5. 다중공선성에 크게 영향을 받지 않는다.
6. 독립변수와 종속변수에 연속형 변수와 범주형 변수가 사용될 수 있다.

- 의사결정나무 특징(단점)

1. overfitting이 발생할 가능성이 높다.(즉 새로운 데이터에 대한 예측 성능이 다른 모델에 비해 좋지 못하다.)
2. 의사결정나무는 연속형 변수를 비연속형 변수로 여기기 때문에, 분류 경계선 부근의 자료값에 대해서 오차가 크다.
3. 설명변수 간의 중요도를 판단하기 쉽지 않다.

- 의사결정나무의 분석 과정

1. 의사결정나무의 형성과정은 **성장, 가지치기, 타당성 평가, 해석 및 예측**으로 이루어진다.
2. **성장 단계** : 각 node에서 적절한 최적의 **분리규칙을 찾아서 나무를 성장**시키는 과정으로, 적절한 **정지규칙을 만족하면 중단**한다.
3. **가지치기 단계** : overfitting을 방지하기 위해, 불필요한 가지를 제거하는 단계
4. **타당성 평가 단계** : validation data를 이용하여 의사결정나무를 평가하는 단계이다.
5. **해석 및 예측 단계** : 구축된 나무모형을 해석하고 예측모형을 설정한 후 예측에 적용하는 단계이다.

- **분리규칙**

1. 분리 변수(입력 변수)가 연속형인 경우 : 'A = x_j <= s'으로 나눌 수 있다.
2. 분리 변수(입력 변수)가 범주형 {1, 2, 3, 4}인 경우 : 'A = 1, 2, 4와 A_c = 3'로 나눌 수 있다.
3. 최적 분할의 결정은 불순도 감소량을 가장 크게 하는 분할이다.

- **분리기준**

1. 종속변수가 이산형 변수일 때 : 카이제곱 통계량 p값(p값이 가장 작은 예측변수와 그 때의 최적분리에 의해서 노드를 형성), 지니 지수, 엔트로피 지수
2. 종속변수가 연속형 변수일 때 : 분산분석에서 F 통계량 p값, 분산의 감소량

- **정지규칙** : 더 이상 분리가 일어나지 않고, 현재의 마디가 끝마디가 되도록 하는 규칙이다.

- **정지기준(사전가지치기를 수행할 때)** : 의사결정나무의 깊이를 지정, terminal node 내 데이터 수의 최소 개수를 지정한다.(사후가지치기를 수행할 때는 해당 정지기준들을 설정할 필요가 없다.)

- 나무의 가지치기

1. branch의 깊이가 깊어질수록 overfitting이 일어날 가능성이 높아지고, 깊이가 얕아지수록 underfitting이 일어날 가능성이 높아진다.
2. 나무의 크기를 모형의 복잡도로 볼 수 있다.
3. 사전가지치기는 보통 node 내 데이터 개수가 일정 수 이하일 때 분리를 정지하고, 사후가지치기는 cost-

- complexity를 이용하여 full tree에 가지치기를 실시한다.

- R에서 의사결정나무 모델링

1. ‘party’패키지의 ‘ctree()’를 통해 의사결정나무를 모델링 할 수 있다.
2. ctree(종속변수 ~ 입력변수, data)

앙상블

- 정의 : 주어진 자료로부터 여러 개의 예측모형들을 만든 후 예측모형들을 조합하여 하나의 최종 예측 모형을 만드는 방법으로 다중 모델 조합, 분류기 조합이 있다.

- train data의 작은 변화(train dataset 내 한 개의 데이터의 수치가 변경되었을 때에 의해 예측모델이 크게 변하는 경우, 해당 모델은 불안정한 것이다. 대표적으로 가장 불안정한 예측 모델은 **의사결정나무**이다.

- 앙상블 기법의 종류로는 **배깅, 부스팅, 랜덤 포레스트**가 있다.

군집분석

- 개요

1. **각 객체(데이터)의 유사성을 측정하여 유사성이 높은 대상 집단을 분류**하고, 군집에 속한 객체들의 유사성과 서로 다른 군집에 속한 객체간의 상이성을 규명하는 분석 방법이다.
2. **특성에 따라 고객을 여러 개의 배타적인 집단으로 나누는 것**이다.
3. 군집의 개수나 구조에 대한 가정 없이 데이터들 사이의 거리를 기준으로 군집화를 유도한다.
4. 대표적인 예로 마케팅 조사에서 소비자군을 분류하여 시장 전략 수립하는 것에 활용된다.
5. 군집분석에서는 **관측 데이터 간 유사성이나 근접성을 측정해 어느 군집으로 묶을 수 있는지 판단**해야 한다.

- 계층적 군집분석과 비계층적 군집분석의 차이점 : **계층적 군집분석은 군집의 개수를 제일 나중에 선정**하지만, **비계층적 군집분석인 K-means 군집분석의 경우**는 모양도 계층적이지 않지만 **군집의 개수를 제일 먼저 선정**하고 모형을 개발한다.

- 계층적 군집분석 예시 : 최단연결법, 최장연결법, 평균연결법, 와드연결

- 비계층적 군집분석 예시 : K-means, 혼합 분포 군집, SOM

- 데이터 간의 거리(=비유사성) 측정법(연속형 변수의 경우) - **데이터 간의 거리가 작을수록 해당 데이터끼리 유사성을 가진다고 할 수 있다.**

1. **유클리디안 거리** : 데이터간의 유사성을 측정할 때 많이 사용하는 거리
2. 표준화 거리 : 모든 데이터 값을 해당 변수의 표준편차로 척도 변환한 후, 유클리디안 거리를 계산하는 방법
3. **마할라노비스** : 변수의 표준화와 변수 간의 상관성을 동시에 고려한 통계적 거리 척도 방법
4. **맨하탄 거리** : 유클리디안 거리와 함께 가장 많이 사용되는 거리로, 맨하탄 도시에서 건물에서 건물을 가기 위한 최단 거리를 구하기 위해 고안된 거리이다.
5. **민코우스키 거리** : 맨하탄 거리와 유클리디안 거리를 한번에 표현한 공식으로, L1거리(맨하탄 거리), L2거리(유클리디안 거리)라 불리고 있다

- 데이터 간의 거리(=비유사성) 측정법(범주형 변수의 경우) - **데이터 간의 거리가 작을수록 해당 데이터끼리 유사성을 가진다고 할 수 있다.**

1. 자카드 거리 - 비교 대상의 두 객체를 **특징들의 집합**으로 간주한다.
2. 코사인 거리 - **문서**를 유사도를 기준으로 분류 혹은 그룹핑할 때 사용한다.

Jaccard Index & Jaccard Distance

Jaccard index
(Intersection over Union,
Jaccard **similarity** coefficient,
Jaccard coefficient)

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}, \quad 0 \leq J(X,Y) \leq 1$$

Jaccard distance
(Jaccard **dissimilarity** coefficient)

$$d_{jaccard}(X,Y) = 1 - J(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cup Y| - |X \cap Y|}{|X \cup Y|}$$

코사인 유사도 (Cosine Similarity) vs. 코사인 거리 (Cosine Distance)

Cosine similarity

$$cosine\ \ similarity = \frac{X \cdot Y}{\|X\|_2 \cdot \|Y\|_2}$$

Cosine distance

$$d_{cosine}(X,Y) = 1 - \frac{X \cdot Y}{\|X\|_2 \cdot \|Y\|_2}$$

, where $\|X\|_2$ is the L2 norm

→ **문서 분류, 군집화 (Text Classification, Clustering) 에 활용**

- 계층적 군집분석

1. n개의 군집으로 시작해 점차 군집의 개수를 줄여 나가는 방법이다.
2. 계층적 군집을 형성하는 방법에는 **합병형 방법**과 **분리형 방법**이 있다.

- 최단연결법

1. 거리행렬에서 거리가 가장 가까운 데이터를 묶어서 군집을 형성한다.
2. 군집과 군집 또는 군집과 데이터의 거리를 계산할 때, **최단거리(min)**를 거리로 계산하여 거리행렬 수정을 진행한다.
3. 수정된 거리행렬에서 거리가 가까운 데이터 또는 군집을 새로운 군집으로 형성한다.

- 최장연결법

1. 군집과 군집 또는 군집과 데이터의 거리를 계산할 때, **최장거리(max)**를 거릴로 계산하여 거리행렬을 수정하는 방법이다.
   (나머지는 최단연결법과 동일)

- 평균연결법

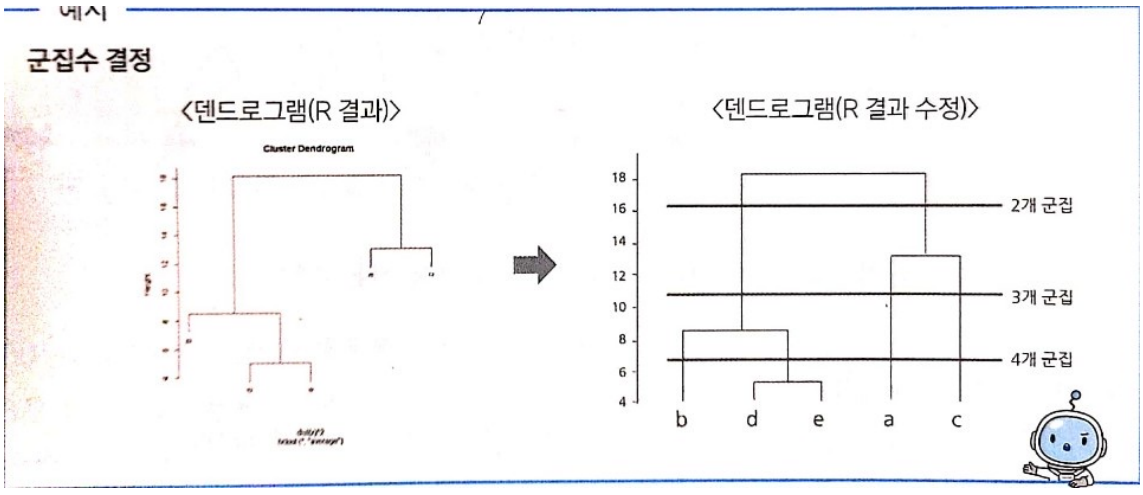
1. 군집과 군집 또는 군집과 데이터의 거리를 계산할 때, **평균(mean)**를 거릴로 계산하여 거리행렬을 수정하는 방법이다.
   (나머지는 최단연결법, 최장연결법과 동일)

- 군집화

1. 거리행렬을 통해 가장 가까운 거리의 객체들간의 관계를 규명하고 덴드로그램을 그린다.
2. 덴드로그램을 보고 군집의 개수를 변화해 가면서 적절한 군집수를 선정한다.
3. 군집의 수는 분석 목적에 따라 선정할 수 있지만, **대부분 5개 이상의 군집은 잘 활용하지 않는다.**

- 군집화 단계

1. 거리행렬을 기준으로 덴드로그램을 그린다.
2. 덴드로그램의 최상단부터 세로축의 개수에 따라 가로선을 그어 군집의 개수를 선택한다.
3. 각 객체들의 구성을 고려해서 적절한 군집수를 선정한다.



- 비계층적 군집분석 : n개의 데이터를 g개의 군집으로 나눌 수 있는 모든 가능한 방법을 점검해 최적화한 군집을 형성하는 것이다. **g개의 군집을 먼저 설정하고**, 해당 알고리즘을 실시한다.

- K-means clustering

1. 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작한다.
2. 거리계산을 통해 군집화가 이뤄지므로, **연속형 변수에만 활용이 가능하다.**
3. **K개의 초기 중심값은 랜덤으로 선택 가능**하며, 가급적이면 서로 멀리 떨어지는 것이 바람직하다. 또한 초기 중심값이 일렬로(좌우, 상하)로 선택하면 군집 혼합되지 않고 층으로 나눠질 수 있어 주의해야한다. 초기 중심값의 선정에 따라 결과가 달라질 수 있다.
4. 초기 중심으로부터 오차 제곱합을 최소화하는 방향으로 군집이 형성되는 **탐욕적 알고리즘**이므로, 안정된 군집은 보장하나 **최적이라는 보장은 없다.**

KC는 대표적인 분리형 군집화 알고리즘 가운데 하나입니다. 각 군집은 하나의 중심(centroid)을 가집니다. 각 개체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 형성합니다. 사용자가 사전에 군집 수(k)가 정해야 알고리즘을 실행할 수 있습니다. k가 하이퍼파라메터(hyperparameter)라는 이야기입니다. 이를 수식으로 적으면 아래와 같습니다.

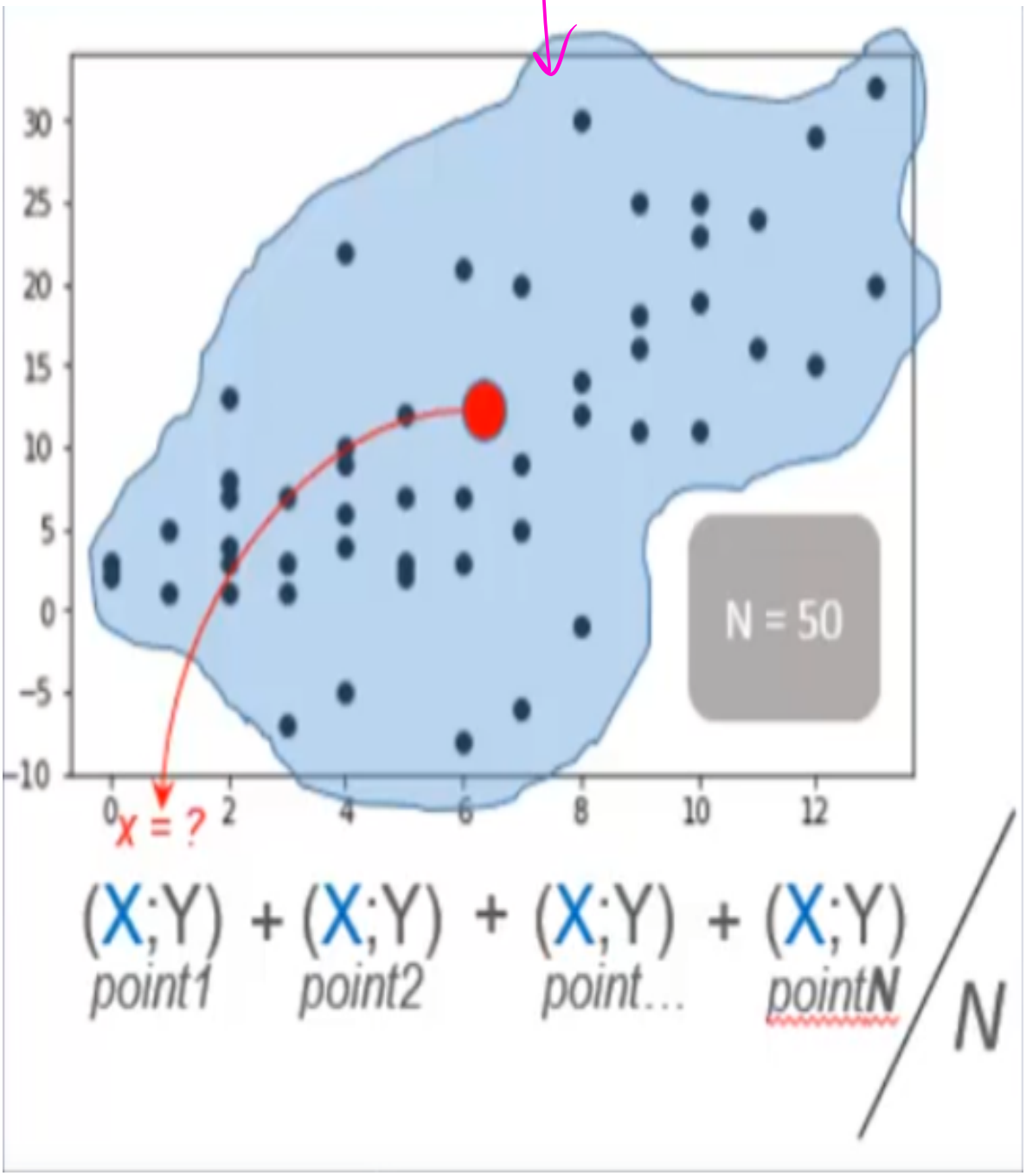
$$X = C_1 \cup C_2 \ldots \cup C_K, \quad C_i \cap C_j = \phi$$

$$argmin_C \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - c_i\|^2$$

↳ 해당 식의 결과가 최적화되는 클러스터 집합 ($C = \{C_1, C_2, \cdots, C_K\}$)을 찾는 것이다.

- K-means steps

1. 연속형 데이터셋 준비
2. 군집의 개수를 결정
3. 초기 **centroid(클러스터의 중심)** 결정(randomly select centroid로 결정)
4. 각 데이터 포인트를 제일 가까운 centroid에 연결(centroid에 연결된 데이터들이 하나의 클러스트를 형성)
5. centroid를 클러스트 내 중심으로 위치를 옮김
6. 클러스트의 변화가 없을 때까지 4, 5번 과정을 반복



- K-means clustering의 장점

1. 알고리즘이 단순하고 빠르게 수행된다.
2. 많은 양의 데이터를 다룰 수 있다.
3. 내부 구조에 대한 사전정보가 없어도 의미있는 자료구조를 찾을 수 있다.

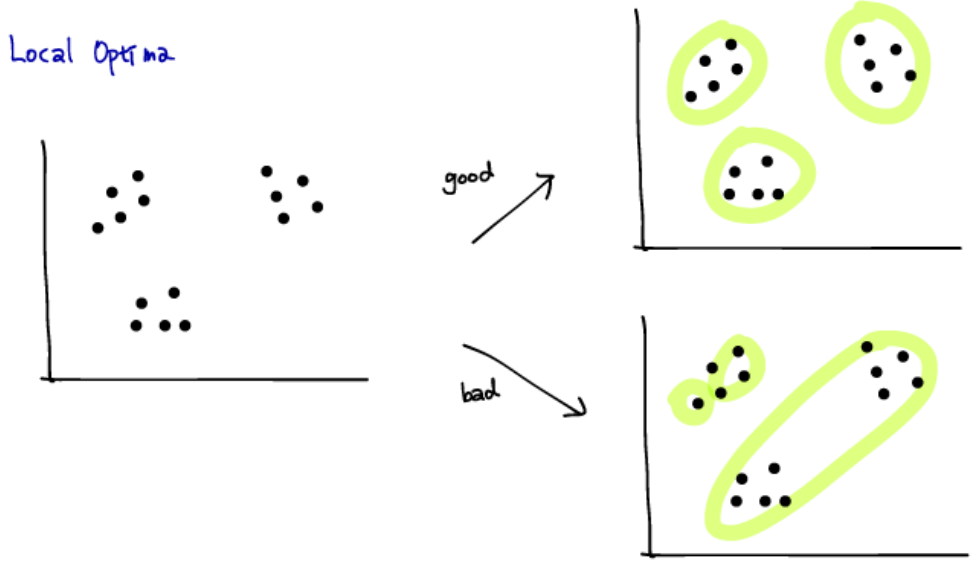
- **K-means clustering의 단점**

1. 가중치와 거리 정의가 어렵다.
2. 초기 군집수 결정이 어렵다. 초기 설정 클러스터의 수가 적합하지 않으면, 결과가 좋지 못하다.
3. 사전에 주어진 목적이 없으므로 결과 해석이 어렵다.
4. 잡음이나 이상값의 영향을 많이 받는다. → 해당 취약점 보완 군집분석 : PAM.
5. 데이터 분포가 특이한 케이스에 대해 군집이 잘 이뤄지지 않는다.

운이 좋다면 적당한 centroid를 고르게 되겠지만, 어떤 경우에는 제대로 clustering이 되지 못하는 초기화를 하게 된다

K-means 알고리즘은 초기화에 따라 다른 결과가 나타날 수 있다. 나쁜 경우, local optima에 빠진다.

Local Optima



- **혼합 분포 군집**

1. 데이터가 K개의 모수적 모형(흔히 **정규분포** 또는 **다변량 정규분포**)의 가중합으로 표현되는 모집단 모형으로부터 나왔다는 가정하에서 모수와 함께 가중치를 자료로부터 추정하는 방법을 사용한다.
2. K개의 각 모형은 군집을 의미하며(즉, K개의 군집이 각각 정규분포의 형태를 띠고있음), 각 데이터는 추정된 K개의 모형 중 어느 모형으로부터 나왔을 확률이 높은지에 따라 군집의 분류가 이루어진다.
3. 흔히 혼합모형에서의 모수와 가중치의 추정(최대가능도추정)에는 **EM 알고리즘**이 사용된다.

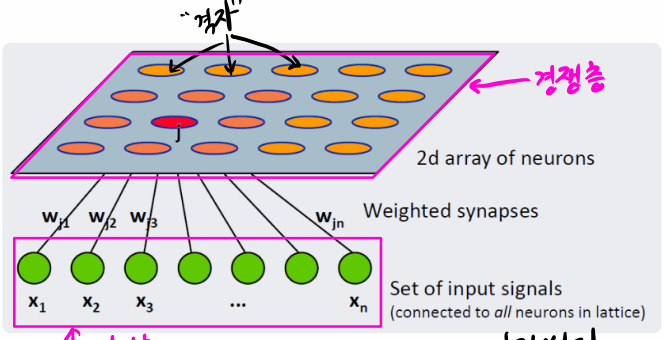
- 혼합 분포 군집의 특징

1. **확률분포**(흔하 정규분포 또는 다변량 정규분포)**를 도입**하여 군집을 수행한다.
2. 군집을 몇 개의 모수(정규분포)로 표현할 수 있으며, 서로 다른 크기나 모양의 군집을 찾을 수 있다.
3. 군집의 크기가 너무 작으면 추정의 정도가 떨어지거나 어려울 수 있다.
4. **이상치 자료에 민감**하므로 사전에 조치가 필요하다.

- SOM(Self Organizing Map)

1. 인공신경망을 활용한 비지도 학습 모델이다.

SOM 아키텍처의 핵심은 대략 아래 그림과 같습니다.



입력벡터 = 레이어 한 개에 대한 벡터

위 그림에서 초록색 노드(x_i)는 n차원 입력벡터의 각 요소를 뜻합니다. 주황색 노드(w_j)는 2차원 격자입니다.

저차원 격자 하나에는 여러 개의 입력벡터들이 속할 수 있습니다. 여기에 속한 입력벡터들끼리는 서로 위치적인 유사도를 가집니다(=가까운 곳에 있음).

그럼 임의의 입력벡터가 주어졌을 때 2차원상 어떤 격자에 속하는지 어떻게 알 수 있을까요? 위 그림 기준으로 j번째 격자는 원데이터 공간에 존재하는 n차원 벡터 $[w_{j1}, w_{j2}, \ldots, w_{jn}]$에 대응됩니다.

다시 말해 2차원상 격자가 위 그림처럼 20개라면 그에 해당하는 n차원 크기의 격자벡터도 20개 있다는 이야기이죠.

"경쟁학습방법"

① 임의의 n차원 입력벡터가 들어왔을 때 가장 가까운 격자벡터를 찾습니다. 이것을 Winning node라고 합니다. 이 벡터에 대응되는 2차원상 격자에 해당 입력벡터를 할당하면 이것이 바로 군집화가 되는 것입니다.

② Winning node와 그 근처의 node 요소들이 업데이트된다.

↑ 격자벡터(=노드)들 중 하나가 'winning node'로 선정된다.

같은 격자에 할당된 입력벡터라 하더라도 Winning node와의 거리가 제각기 다를 수 있습니다. 이러한 멀고 가까움 또한 표시를 하게 되면 고차원 공간의 원데이터를 2차원 내지 3차원으로 차원을 축소하는 효과까지 낼 수 있습니다.

2. 고차원의 데이터를 저차원의 지도 형태(2차원)로 형상화 하기 때문에, 시각적으로 이해하기 쉽다.
3. 입력 변수의 위치 관계를 그대로 보존하기 때문에, 실제 데이터가 유사하면 지도상에서 가깝게 표현된다.
4. 역전파 알고리즘 등을 이용하는 인공신경망과 달리 단 하나의 전방 패스를 사용함으로써 속도가 매우 빠르다.
   따라서, 실시간 학습처리를 할 수 있는 모형이다.