**[시험용 정리] Big Data Analytics : Data Mining**

**- 연세대 김현중 교수님 자료-**

목차

[1. 데이터 마이닝 4](#_Toc344994330)

[ 데이터 마이닝이란 4](#_Toc344994331)

[ KDD(Knowledge Discovery in Database) 프로세스 4](#_Toc344994332)

[ 데이터 마이닝 모델 타입 5](#_Toc344994333)

[2. Linear Regression 5](#_Toc344994334)

[ 회귀 분석 5](#_Toc344994335)

[ Simple Linear Regression 5](#_Toc344994336)

[ ANOVA 테이블 6](#_Toc344994337)

[ 가설 검증(F-Statistic) 6](#_Toc344994338)

[ 통계적 추론(Statistical Inference) 6](#_Toc344994339)

[ Linear Regression의 해석 6](#_Toc344994340)

[ 변수선택 알고리즘 7](#_Toc344994341)

[ All Subsets 7](#_Toc344994342)

[ Backward Elimination 8](#_Toc344994343)

[ Forward Selection 8](#_Toc344994344)

[ Stepwise Selection 9](#_Toc344994345)

[3. Logistic Regression 9](#_Toc344994346)

[ 분류(classification)란 9](#_Toc344994347)

[ 분류 모델의 종류 9](#_Toc344994348)

[ 1변수에서의 Logistic Regression 10](#_Toc344994349)

[ 2변수에서의 Logistic Regression 10](#_Toc344994350)

[ [중요] 새로운 사례에 대한 스코어 계산하기 11](#_Toc344994351)

[ Logistic Regression의 해석 11](#_Toc344994352)

[ OR(Odds Ratio) 12](#_Toc344994353)

[4. Logistic Regression과 OR 13](#_Toc344994354)

[5. 판별 분석(Discriminant Analysis) 13](#_Toc344994355)

[ LDA(Linear Discriminant Analysis) 13](#_Toc344994356)

[ QDA(Quadratic Discriminant Analysis) 14](#_Toc344994357)

[6. K-nearest neighbor 15](#_Toc344994358)

[ Nearest neighbor의 개념 15](#_Toc344994359)

[ K-nearest neighbor 15](#_Toc344994360)

[ 5-nearest neighbor 16](#_Toc344994361)

[7. Evaluation 기법 16](#_Toc344994362)

[ 데이터의 역할 16](#_Toc344994363)

[ 모델 비교 방법 17](#_Toc344994364)

[ Confusion Matrix 17](#_Toc344994365)

[ 에러율과 정확도의 한계 18](#_Toc344994366)

[ 민감도(sensitivity)와 특이도(specificity) 18](#_Toc344994367)

[ 민감도(sensitivity) 18](#_Toc344994368)

[ 특이도( specificity) 18](#_Toc344994369)

[8. 차트 19](#_Toc344994370)

[ Lift(Gain) 차트 만들기 19](#_Toc344994371)

[ Lift를 통한 모델 비교하기 사례 20](#_Toc344994372)

[ ROC 커브 21](#_Toc344994373)

[ AUROC(Area Under ROC) 22](#_Toc344994374)

[ [중요] AUROC 계산하기 22](#_Toc344994375)

[9. 의사결정 트리(Decision Tree) 23](#_Toc344994376)

[ Gini Impurity 23](#_Toc344994377)

[ 트리를 만들 때 고려사항 23](#_Toc344994378)

[ CART 알고리즘의 특징 24](#_Toc344994379)

[ CART의 방식: 분할할 기준(split)을 어떻게 찾을 것인가? 24](#_Toc344994380)

[ CART의 방식: Split이 많다면, 어느 split을 선택해야 하나? 24](#_Toc344994381)

[ CART의 방식: 언제 분할을 멈출 것인가? 25](#_Toc344994382)

[ CART의 방식: 트리의 일부 가지를 쳐낼 것인가(prunig)? 25](#_Toc344994383)

[ Surrogate split 25](#_Toc344994384)

[ Decision Tree를 사용한 변수 선택 25](#_Toc344994385)

[10. 신경망(Neural Networks) 26](#_Toc344994386)

[ 아키텍처 26](#_Toc344994387)

[ 전형적인 선형 모델의 구조 26](#_Toc344994388)

[ 신경망 모델의 구조 27](#_Toc344994389)

[ [중요] 신경망 노드의 출력값 계산 27](#_Toc344994390)

[ 유의사항 28](#_Toc344994391)

[11. Support Vector Machine 28](#_Toc344994392)

[12. 앙상블(Ensemble) 기법 28](#_Toc344994393)

[ Bagging 28](#_Toc344994394)

[ Boosting 29](#_Toc344994395)

[ Adaboost의 사례 30](#_Toc344994396)

[13. 비교사 학습(Unsupervised Learning) 30](#_Toc344994397)

[ 클러스터링(Clustering) 30](#_Toc344994398)

[ 거리(Distance) 30](#_Toc344994399)

[ 계층적 군집 분석(Hierarchical Cluster Method) 31](#_Toc344994400)

[ 병합 계층 군집화(Agglomerative Hierarchical Method) 31](#_Toc344994401)

[ 분할 계층 군집화(Divisive Hierarchical Method) 31](#_Toc344994402)

[ 거리 계산 방법 32](#_Toc344994403)

[ [중요] Single Linkage를 사용한 군집화 33](#_Toc344994404)

[14. K-means clustering 34](#_Toc344994405)

[ 특징 34](#_Toc344994406)

[ 알고리즘 34](#_Toc344994407)

[ 유의사항 34](#_Toc344994408)

[15. 연관성 분석(Association Analysis) 34](#_Toc344994409)

[ 연관성 분석이란 34](#_Toc344994410)

[ 연관 규칙 35](#_Toc344994411)

[ 동시 구매표 35](#_Toc344994412)

[ 지지도(Support)와 신뢰도(Confidence) 35](#_Toc344994413)

[ 지지도(support) 36](#_Toc344994414)

[ 신뢰도(confidence) 36](#_Toc344994415)

[ [중요] 지지와 신뢰도 계산 예제 36](#_Toc344994416)

[ 향상도(Lift) 37](#_Toc344994417)

[ 지지도와 신뢰도의 한계 37](#_Toc344994418)

[ 향상도 37](#_Toc344994419)

# 데이터 마이닝

## 데이터 마이닝이란

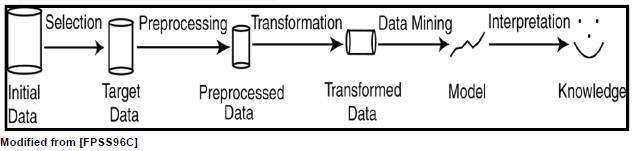
* 다양한 데이터 분석 도구를 사용하여 데이터간의 패턴과 관계를 찾는 프로세스
* 사람이 최소한으로 개입하여, 컴퓨터 도구를 활용하는 탐색적 데이터 분석(exploratory data analysis)
* 과거의 데이터를 기반으로 예측 모형(predictive model) 을 만들어서 미래의 데이터에 모델을 적용



통계학에서는 과거의 데이터에 대해 인과관계를 발견하는 한편, 분석에서는 예측 모형을 만들어서 미래의 데이터를 예측한다.

## KDD(Knowledge Discovery in Database) 프로세스

* 데이터로부터 유용한 정보와 패턴을 찾는 프로세스
* 데이터 마이닝은 KDD 프로세스의 한 단계



- 수집(selection) : 다양한 데이터 정보원으로부터 데이터를 수집

- 전처리(preprocessing) : 데이터 cleansing

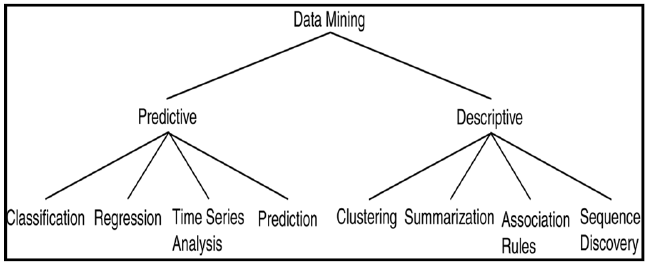
- 변환(transformation) : 표준 형식으로 데이터 변환

- 데이터 마이닝 : 의미 있는 결과를 도출

- 해석 및 평가 : 사용자에게 분석 결과를 유의미한 형태로 전달

* 전처리는 전체 KDD 프로세스에서 약 80% 정도의 시간을 소요한다.

## 데이터 마이닝 모델 타입



* 회귀(regression) : 연속형 값에 대한 예측 모델링
* 분류(classification) : 범주형 또는 이산형 값에 대한 예측 모델링
* 클러스터링(clustering) : 데이터를 다수의 그룹으로 분할

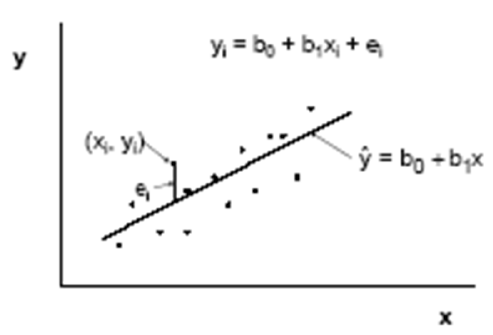
# Linear Regression

## 회귀 분석

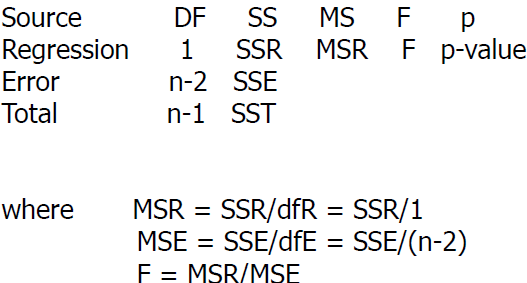
* 독립 변수(x1, x2, …., xk)와 연속형 종속 변수(y) 사이의 관계를 평가하기 위한 통계적인 기법
* Y = f(x1, x2, …, xk)를 만족하는 함수 f를 찾는 방법
* 이를 통해 y를 예측하거나 설명하기에 가장 중요한 독립변수를 결정하는 과정

## Simple Linear Regression

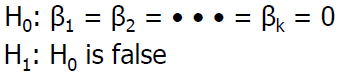
* x, y가 각각 1개인 경우



## ANOVA 테이블



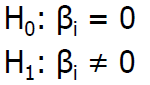
## 가설 검증(F-Statistic)

귀무가설 (반드시 =이 있어야 함)

대립가설 (반드시 =이 있어선 안됨)

* 기본적으로 귀무가설이 참이라고 본다.
* p-value가 작을수록 더 강한 증거이며, 이 경우에는 대립가설이 참이 된다.  
  (일반적으로 p-value < 0.05)

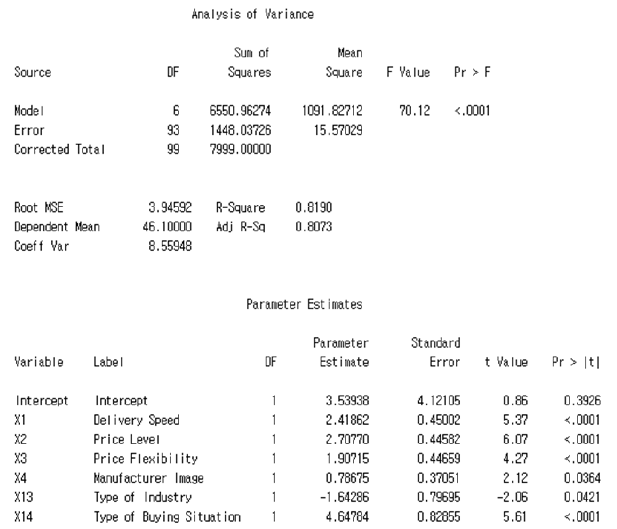
## 통계적 추론(Statistical Inference)

* 다변량 회귀에서는 각각의 파라미터에 대해 가설을 검증할 수 있다.

=> xi 변수가 중요하지 않다는 의미

=> xi 변수가 유의미하다는 의미

## Linear Regression의 해석



p-value가 작을수록 중요한 변수

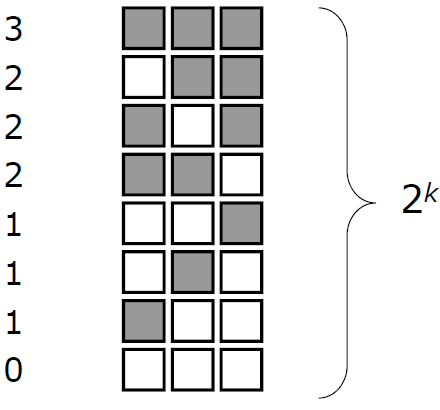
개별적인 x 변수의 유의성

* 일반적으로 변수의 p-value가 5%보다 작으면 그 변수를 유의하다고 본다.

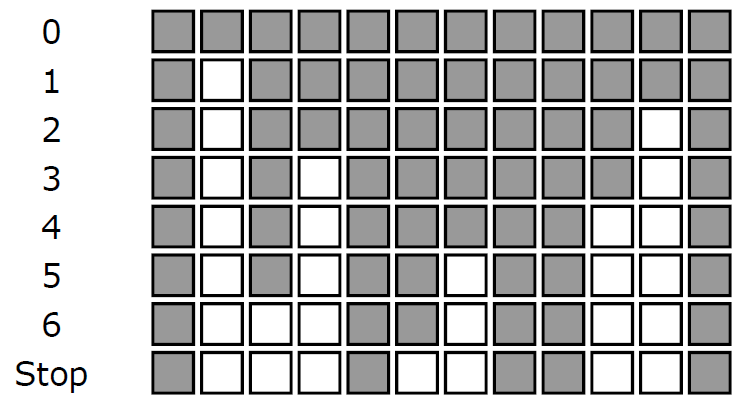
## 변수선택 알고리즘

### All Subsets

x1 x2 x3



### Backward Elimination

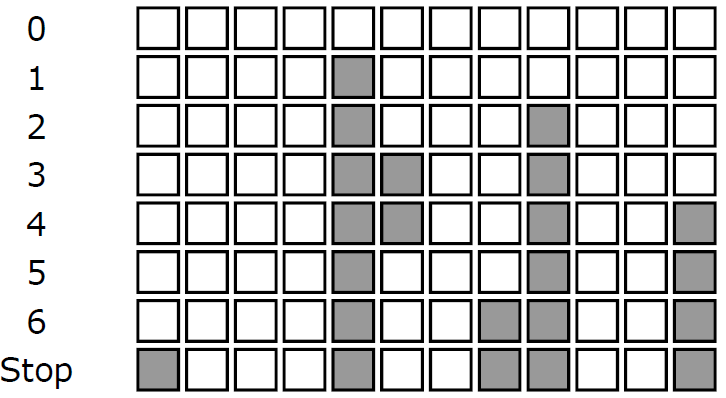


* 모든 변수를 포함하는 모델로부터 시작한다.
* 가장 높은 p-value를 가지는 변수를 제거한다.
* 남아있는 변수가 모두 유의할 때까지 작업을 계속한다.
* 빅데이터에 가장 적합한 방식이다.

### Forward Selection

* 하나 또는 그 이상의 변수를 사용한 모델로부터 시작한다.
* 남아있는 각 변수를 차례대로 추가해서 가장 작은 p-value를 가지는 변수를 추가한다.
* 유의미한 변수를 추가할 수 없을 때까지 작업을 계속한다.

### Stepwise Selection



* 하나 또는 그 이상의 변수를 사용한 모델로부터 시작한다.
* 남아있는 각 변수를 차례대로 추가해서 가장 작은 p-value를 가지는 변수를 추가한다.
* 만약 변수를 추가해서 기존 변수의 p-value가 기준치보다 큰 경우에는 해당 변수를 제거한다..
* 유의미한 변수를 추가/삭제할 수 없을 때까지 작업을 계속한다.

# Logistic Regression

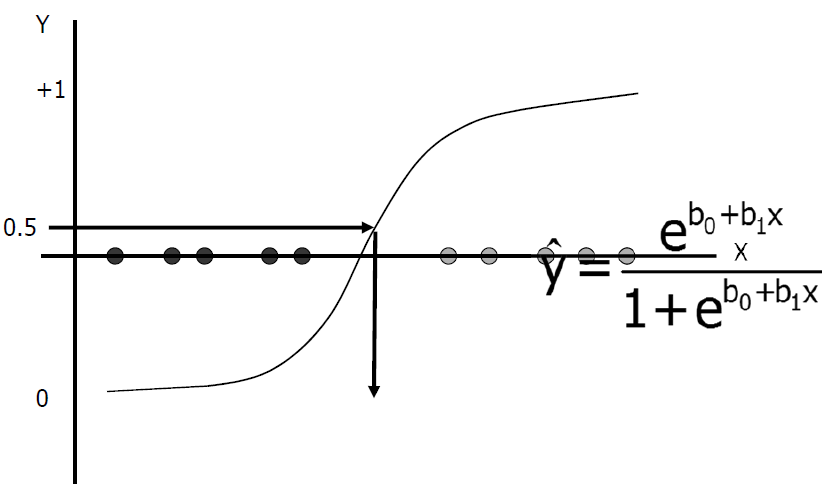
## 분류(classification)란

* 범주형 또는 이산형 값에 대한 예측 모델링
* 분류는 교사 학습(supervised learning)
* 학습 데이터(.training data)를 사용해서 모델을 분류 모델을 만든다.

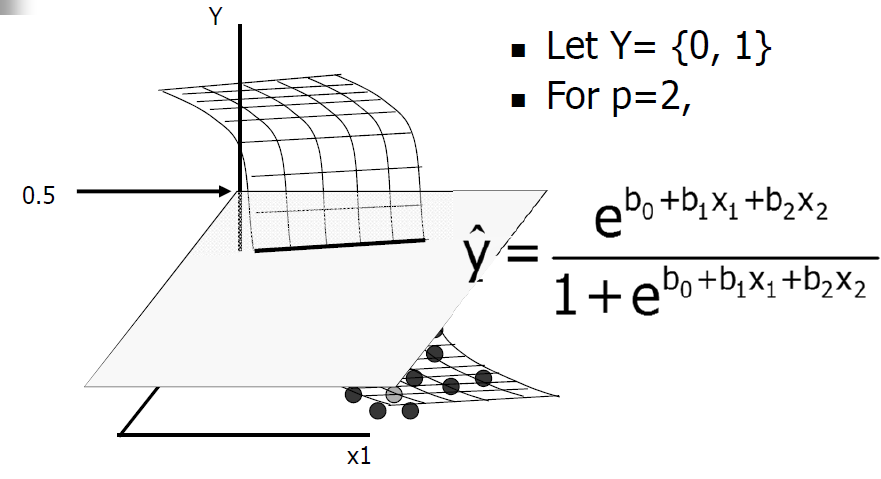
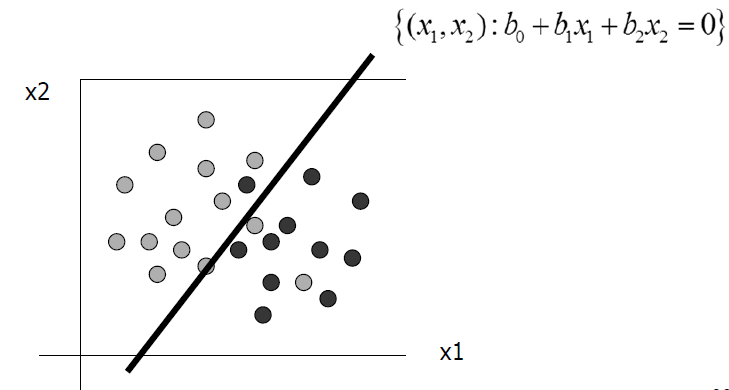
## 분류 모델의 종류

* Logistic Regression
* Discriminant Analysis
* K-nearest neighbors
* Decision Tree
* Nural Networks
* Support Vector Machine

## 1변수에서의 Logistic Regression

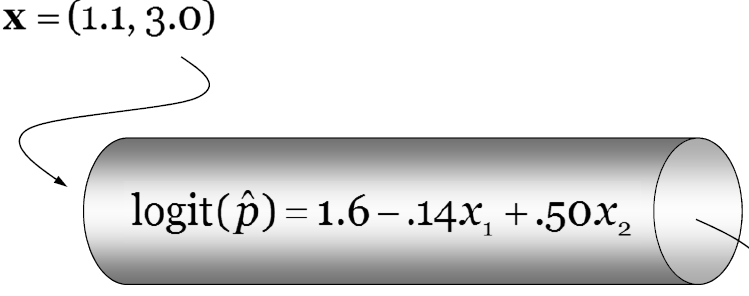


## 2변수에서의 Logistic Regression

* 
* Hyperplane at y = 0.5
* 

## [중요] 새로운 사례에 대한 스코어 계산하기

아래와 같이 x와 logistic 함수가 주어진 경우, p 값을 구하라



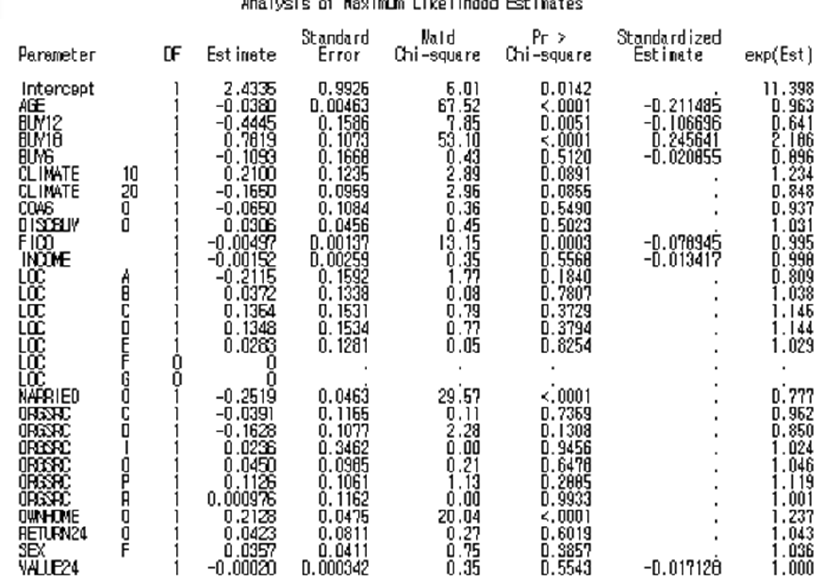
<풀이>

## Logistic Regression의 해석

나이가 1 증가하면, 구매율은 0.963배 증가한다

나이가 많아지면 구매율은 떨어진다

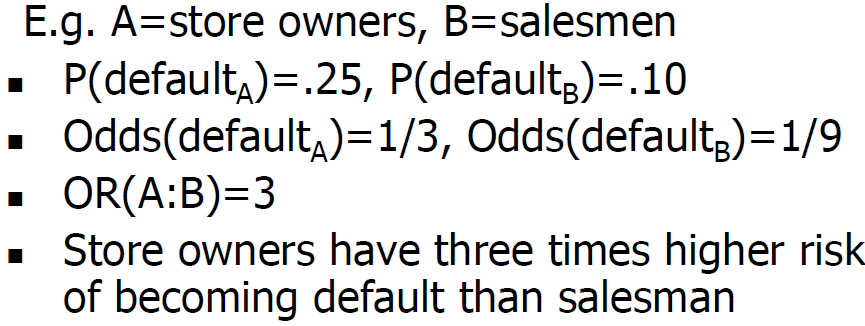
odds ratio



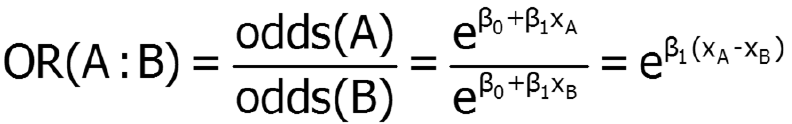
Bi

여성이 남성이 되면, 구매율이 1.036배 증가한다.

## OR(Odds Ratio)

* odds ratio의 예
* 

# Logistic Regression과 OR

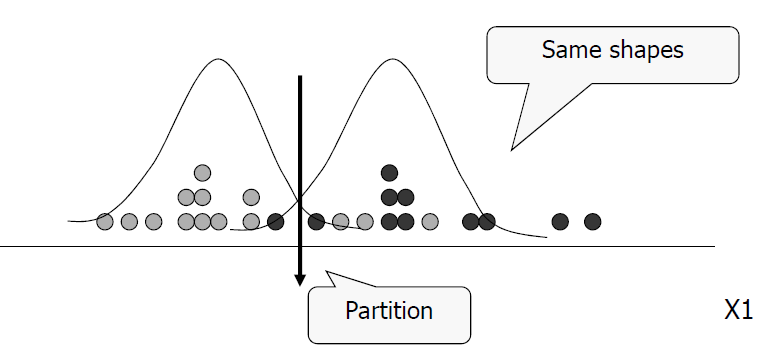


* 만약 x 가 나이라면, 나이가 1증가할 때 마다 odds는 만큼 증가한다.
* 또한 odds ratio가 1이라는 뜻은, b1이 0이라는 뜻이므로, x 변수는 독립변수 y와 무관하다는 뜻이다.

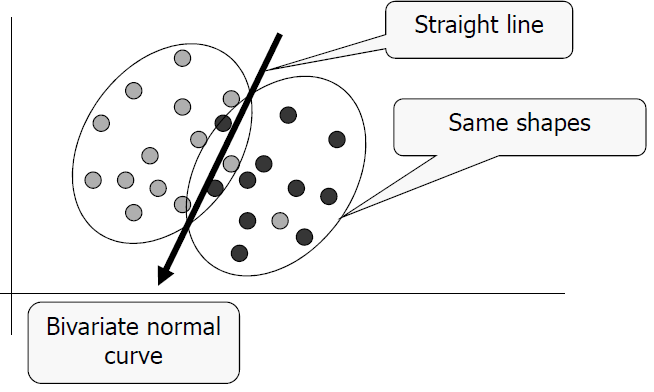
# 판별 분석(Discriminant Analysis)

* 분류 방법 중 하나로, 정규분포를 가정한다.

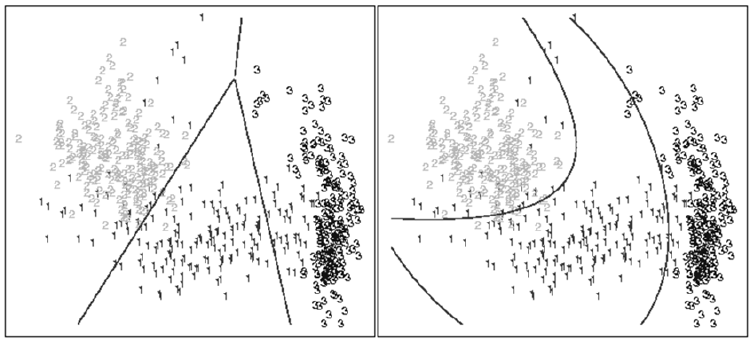
## LDA(Linear Discriminant Analysis)

* 동일한 평균과 분산을 가지는 정규 분포를 가정
* 1차원,
* 

hyperplane

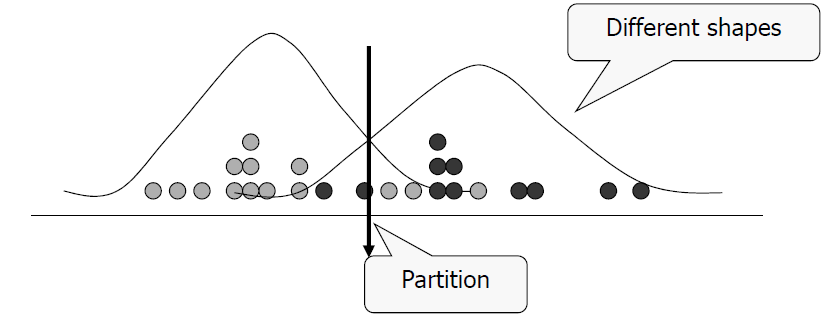
* 2차원
* 

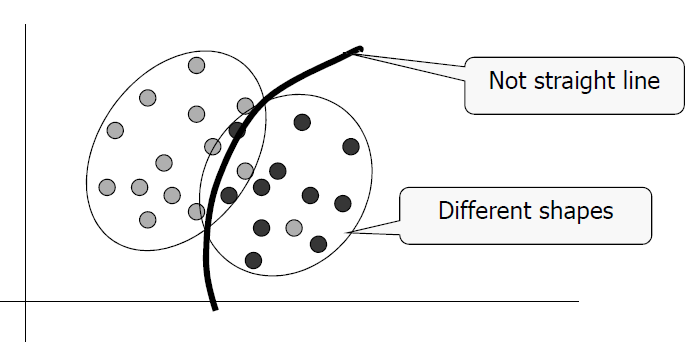
hyperplane

* LDA에서는 hyperplane이 직선(또는 평면)이다.
* 따라서 LDA에서는 오분류가 발생할 빈도가 높아진다. QDA의 hyperplane은 곡선이므로 오분류가 발생할 빈도가 낮아진다.
* 

LDA QDA

## QDA(Quadratic Discriminant Analysis)

* 서로 다른 평균과 분산을 가지는 정규 분포를 가정
* 1차원
* 
* 2차원,



# K-nearest neighbor

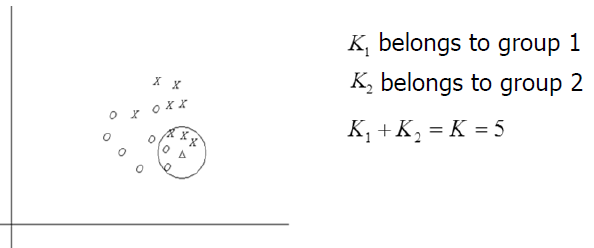
## Nearest neighbor의 개념

* 정규분포를 가정하지 않는다.
* 새로운 사례에 대해, 학습 데이터 중 해당 사례가 가장 가까운 데이터들을 찾는다.  
  (Mahalanobis distance, Eucledian Distance 등을 사용해서)
* 새로운 사례를 가장 근접한 이웃 그룹에 할당한다.

## K-nearest neighbor

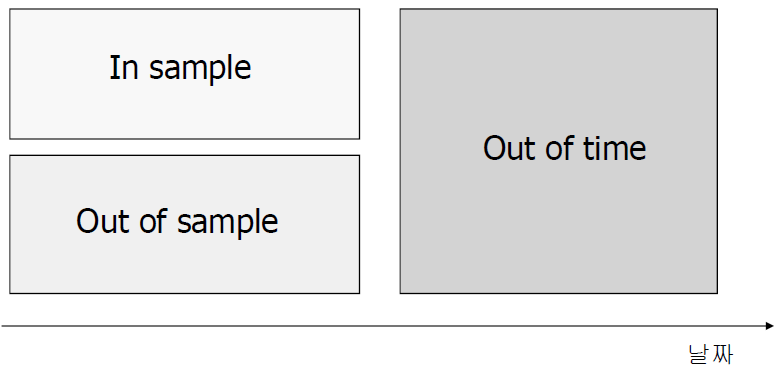
* 가장 가까운 K개의 데이터를 찾는다.

## 5-nearest neighbor

* .
* 세모로부터 가장 가까운 5개의 데이터를 찾는다.
* 그 중에서 가장 많은 데이터를 가지는 이웃그룹에 세모를 할당한다.

# Evaluation 기법

## 데이터의 역할

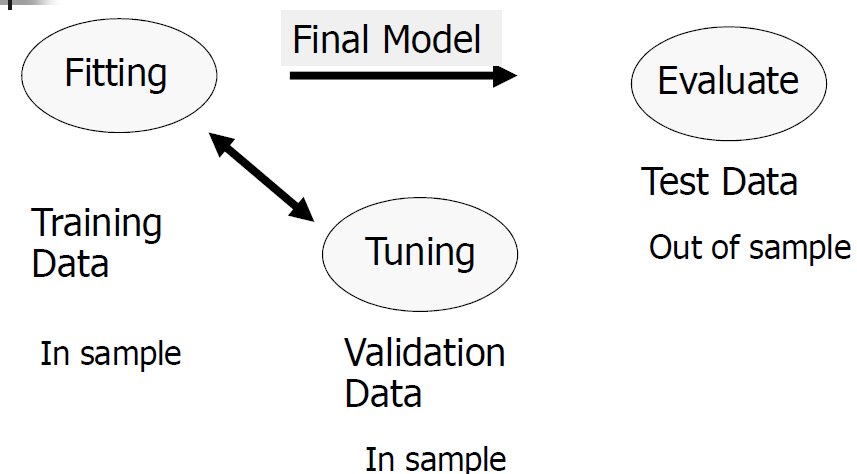


현재

* In sample : 모델 개발에 사용된 데이터

In sampe 데이터 중 70%는 training용으로, 30%는 validation용으로 사용

* Out of sample : 모델 개발에 사용되지 않은 동시점의 데이터
* Out of time : 개발 후 관찰된 데이터(미래 데이터)



최종 후보 선택

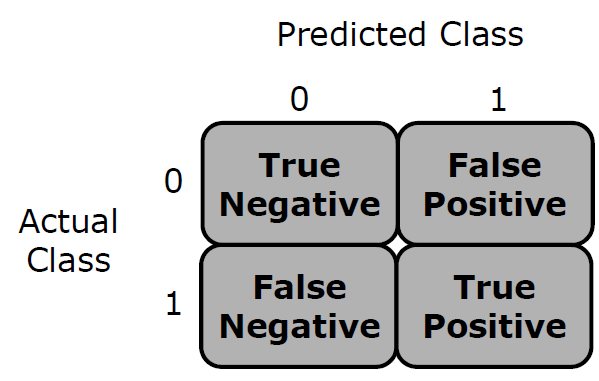
여러 후보 모형을 개발

* 층화 추출(stratified sampling) : 모집단에서 랜덤으로 표본 추출을 할 경우, 사례가 적은 Bad는 모델링을 할 수 있을 정도로 충분히 추출하지 못할 수도 있다. 이 경우 전체 training data에서의 Good : Bad의 비율을 그대로 유지한 채 추출해야 한다.
* validation data는 모델을 최적화하거나, overfitting을 방지하기 위한 용도로 활용한다.

## 모델 비교 방법

* 정확도(accuracy) 또는 에러율
* Lift 차트
* ROC 커브
* KS 통계

## Confusion Matrix

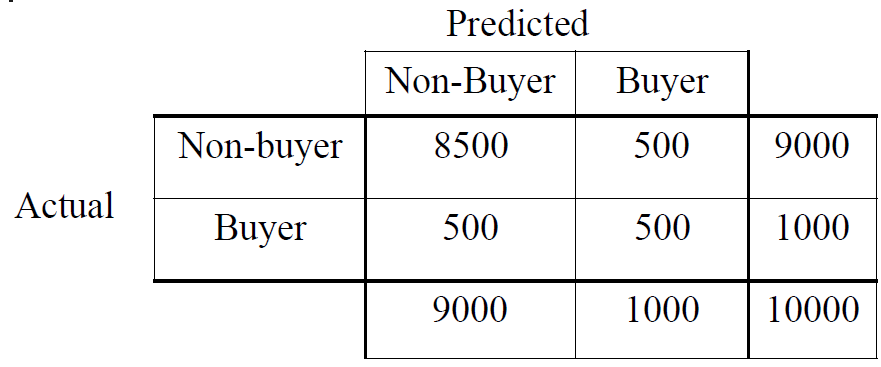


올바른 예측(정확도)

거짓 예측(에러율)

* 에러율(error rate) : 전체 데이터 중 거짓 예측한 데이터의 비율
* Redistribution error : training data에 대한 에러율
* Validation data error : validation data에 대한 에러율
* Redistribution error는 항상 Validation data error보다 낮다. 에러율을 측정할 때는 Validation data error를 사용해야 한다.

## 에러율과 정확도의 한계

* 
* 위의 confusion matrix에서 정확도는 90%이며, 에러율은 10%다. 하지만 원하는 결과가 실제로 구매할 고객을 찾는 것이라면, 구매할 것이라고 예측한 고객 1000명 중에서 실제로 구매한 사람은 500명, 즉 50%정도밖에 예측하지 못한다.

## 민감도(sensitivity)와 특이도(specificity)

### 민감도(sensitivity)

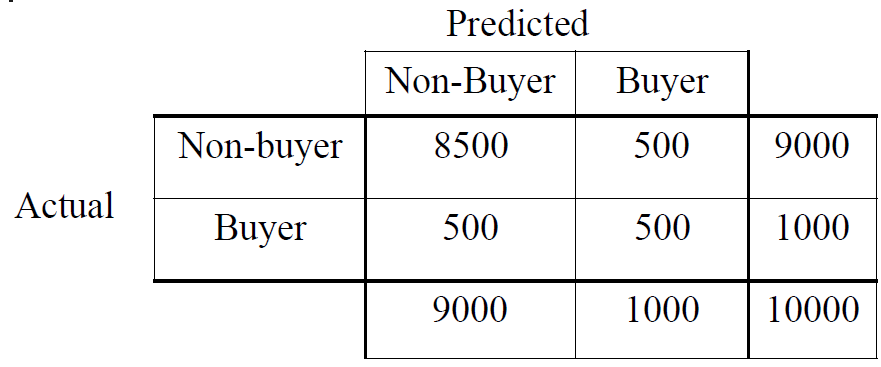
* 실제 Positive 중 Positive로 예측한 비율

민감도 = TP / P = 500 / 1000 = 50%

### 특이도( specificity)

* 실제 Negative 중 Negative로 예측한 비율

특이도 = TN / N = 8500 / 9000 = 94.4%



N

P

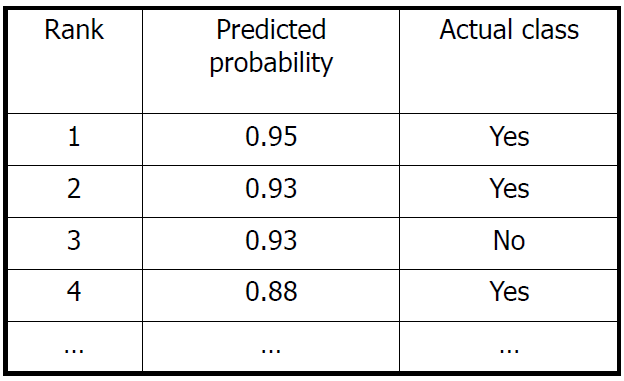
TN

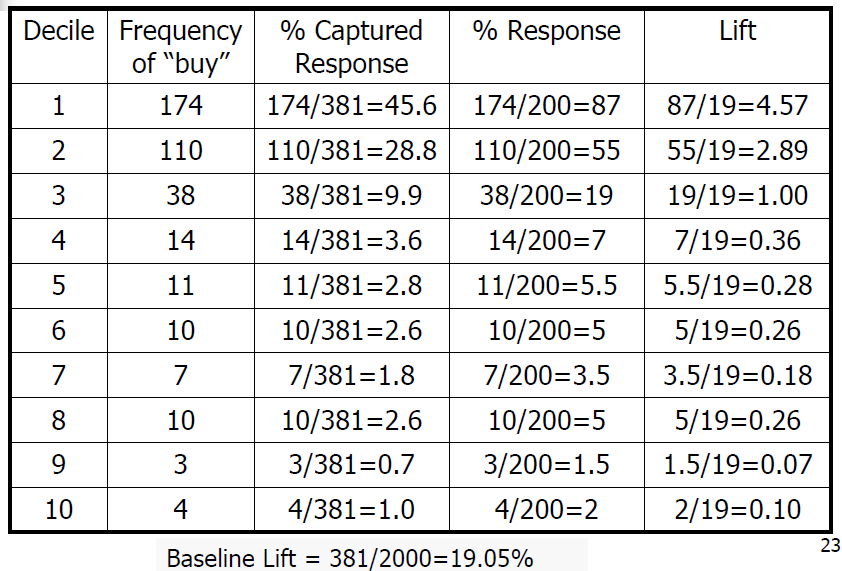
TPP

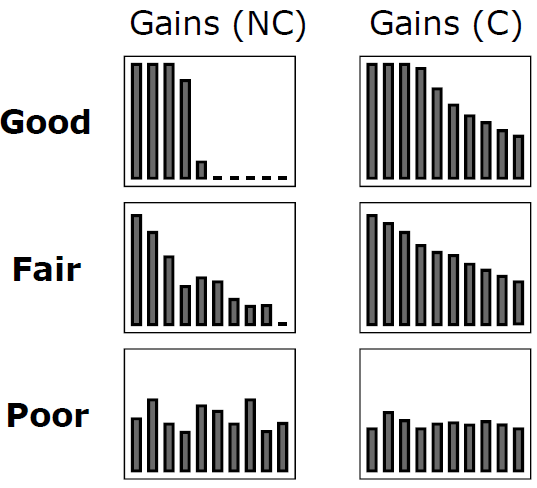
# 차트

## Lift(Gain) 차트 만들기

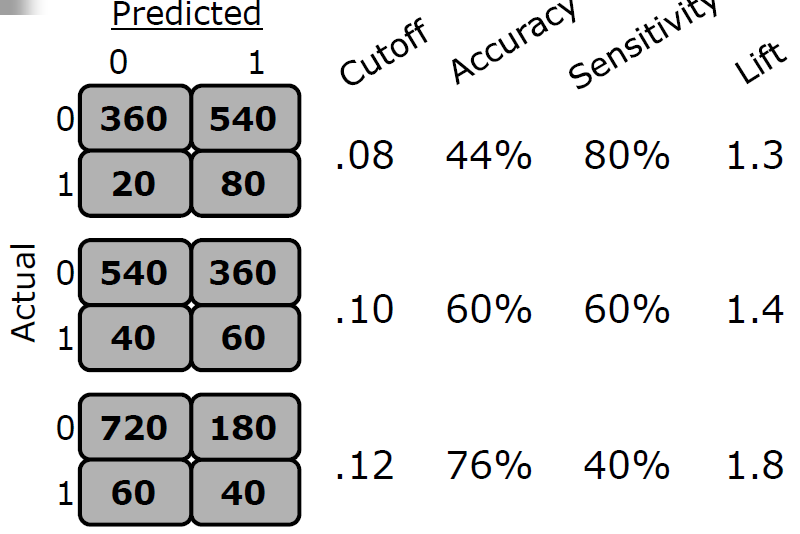
* 데이터셋의 각 데이터는 예측확률을 가진다.
* 전체 데이터셋을 예측확률에 대해 내림차순으로 정렬한다.



* 만약 모델이 제대로 만들어졌다면, 예측확률이 높은 사람들이 실제로 Y일 경우가 많을 것이다라는 가정에 기반한다.
* 전체 데이터를 십분위로 분할하여 아래와 같이 만든다.
* 
* Frequency of “buy” : 200명 중 실제로 구매한 사람
* % Captured Response : 반웅검출율  
  = 해당 등급의 실제 구매자 / 전체 구매자
* % response : 반웅률  
  = 해당 등급의 실제 구매자 / 200명
* Lift: 향상도  
  = 반응률 / 기본 향상도
* 좋은 모델이라면 Lift가 빠른 속도로 감소해야 한다.

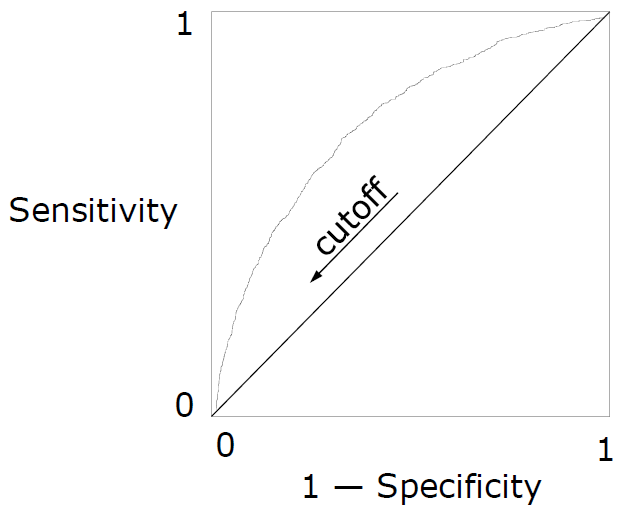


## Lift를 통한 모델 비교하기 사례



* 첫 번째의 경우,  
  정확도 : 정확환 예측 / 전체 = (360 + 80) / 1000 = 44%  
  민감도 : Positive / 실제 Positive = 80 / 100 = 80%

## ROC 커브



false positive rate

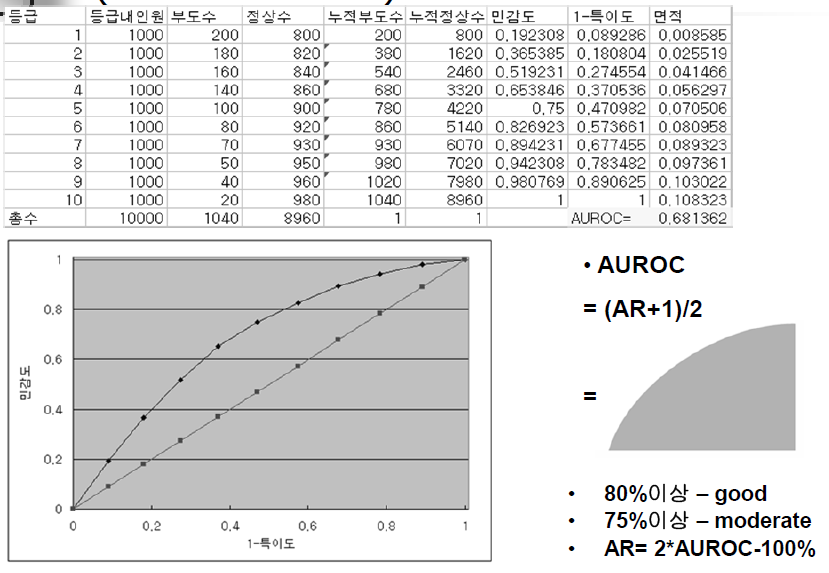
실제 0, 예측 1

true positive rate

실제 1, 예측 1

* 오류(1-특이도)가 같은 상황이라면, 곡선이 위로 갈수록 제대로 맞출 확률이 높다.

## AUROC(Area Under ROC)



* AUROC가 80%이상일 때 good

## [중요] AUROC 계산하기

1등급,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 예측 | | |  |
| 실제 |  | 0 | 1 |  |
| 0 |  | 800 | 8960 |
| 1 |  | 200 | 1040 |
|  |  |  |  |  |

민감도 = 200 / 1040

1- 특이도 = 800 / 8960

2등급,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 예측 | | |  |
| 실제 |  | 0 | 1 |  |
| 0 |  | 1620 | 8960 |
| 1 |  | 380 | 1040 |
|  |  |  |  |  |

민감도 = 380 / 1040

1- 특이도 = 1620 / 8960

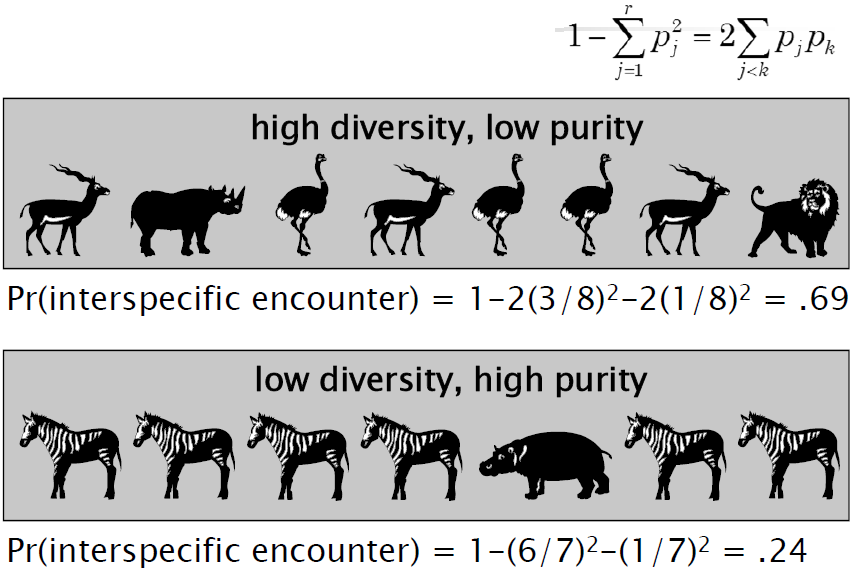
각 등급별로 계산 후에, 각 구간별로 위의 사다리꼴 넓이에서 아래 사다리꼴의 넓이를 뺀다.

# 의사결정 트리(Decision Tree)

* Good 또는 Bad가 많이 모여 있는 부분을 찾는 방법
* 유사도(similarity)를 최대화할 수 있는 방향으로 입력 변수를 분할한다.
* 장점 : 해석이 쉽다 / 범주형 변수를 가변수화 하지 않고 그대로 사용할 수 있다.
* 단점 : 정확도가 떨어진다.

## Gini Impurity

그룹이 유사한 정도를 측정하는 방법



* Gini Impurity가 높다는 말은 유사하지 않은 아이템이 많이 섞여 있다는 의미
* Gini Impurity가 낮은 방향으로 분할해야 한다.

## 트리를 만들 때 고려사항

* 분할할 기준(split)을 어떻게 찾을 것인가?
* Split이 많다면, 어느 split을 선택해야 하나?
* 언제 분할을 멈출 것인가?
* 트리의 일부 가지를 쳐낼 것인가(prunig)?

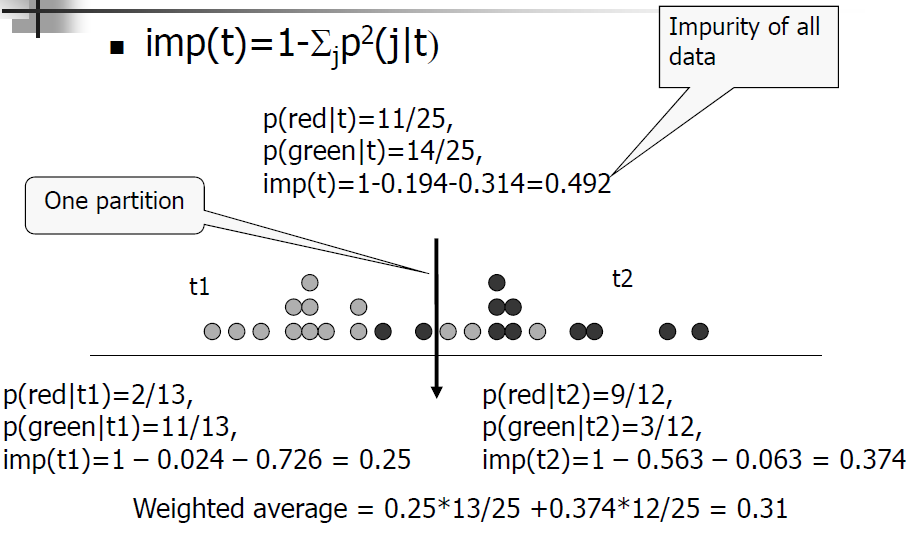
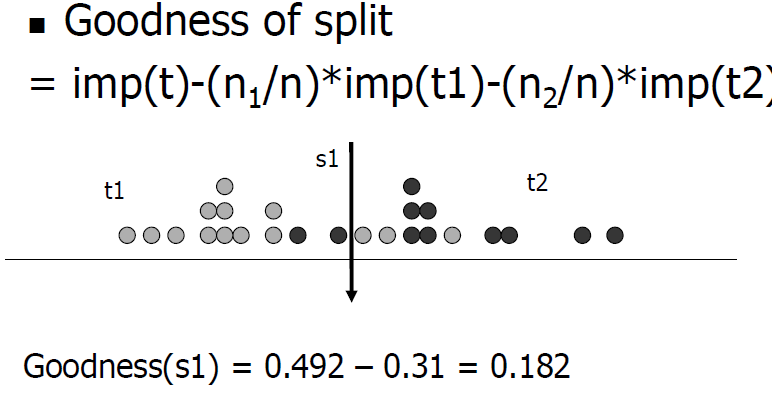
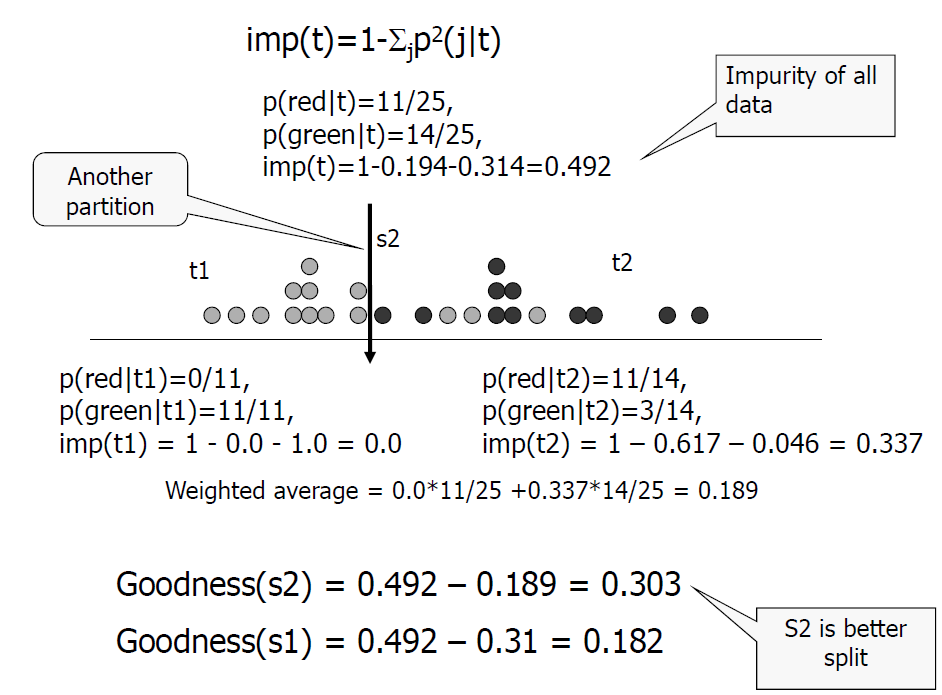
## CART 알고리즘의 특징

* 분할 정복
* 한번에 하나의 변수만 사용한다.
* 이진 트리
* Greedy search : 가능한 모든 조건을 검색한다

## CART의 방식: 분할할 기준(split)을 어떻게 찾을 것인가?

* Greedy Search

## CART의 방식: Split이 많다면, 어느 split을 선택해야 하나?

* Gini Impurity가 가장 낮은 split을 선택한다.
* 
* 
* 

## CART의 방식: 언제 분할을 멈출 것인가?

* 트리의 높이가 특정 크기보다 클 때
* 하위 트리의 노드 개수가 특정 개수보다 작을 때
* Impurity가 감소 크기가 정해진 크기 보다 작을 때 등

## CART의 방식: 트리의 일부 가지를 쳐낼 것인가(prunig)?

* Training data에 대해 모델을 만들 때 overfitting을 하게 되면, 해당 모델은 validation data에서는 예측율이 떨어진다.
* Decision Tree를 만들 때 overfitting을 막기 위해, 오류율이 높은 가지를 쳐내야 한다.

## Surrogate split

* 미래의 데이터에 missing value가 있더라도 decision tree를 사용하기 위해, 추가적인 split을 만들어둔다.

## Decision Tree를 사용한 변수 선택

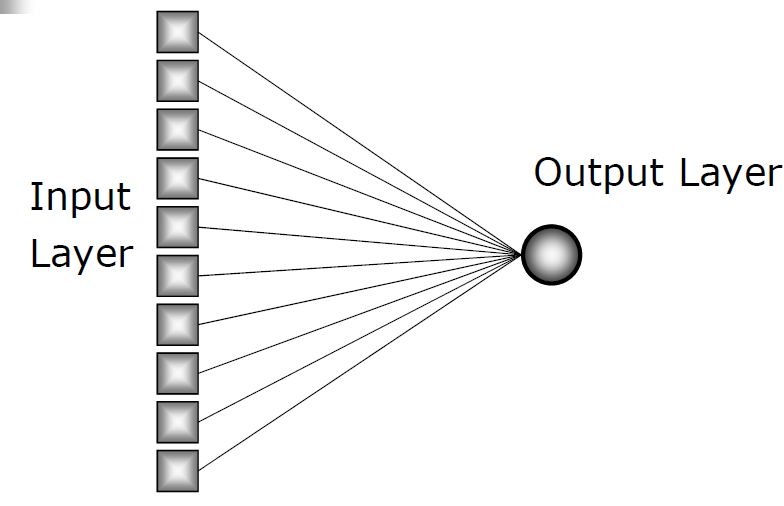
* 중요하지 않은 변수를 필터링하기 위한 목적으로 Decision Tree를 사용할 수 있다.
* Decision Tree의 split 정보를 기준으로 변수를 선택할 수 있다.

# 신경망(Neural Networks)

* 비선형 통계 모델
* Universal approximator
* 장점 : 정확도가 높다
* 단점 : 해석력이 낮다.  
  (Decision Tree의 정반대)

## 아키텍처

### 전형적인 선형 모델의 구조

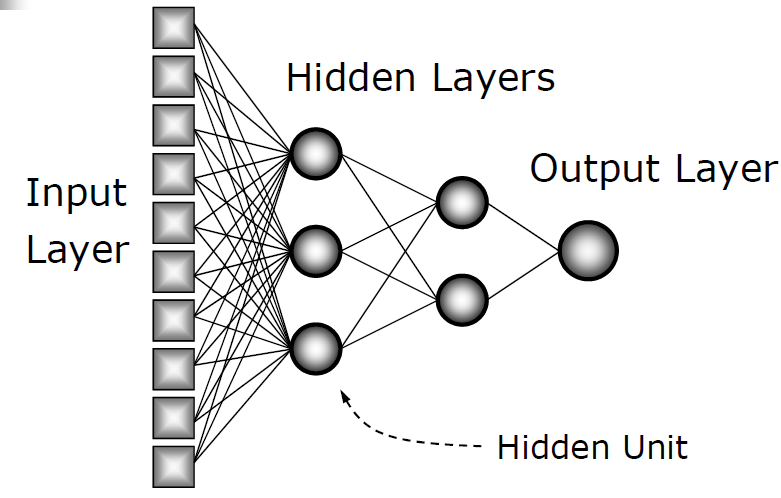
* 

X10

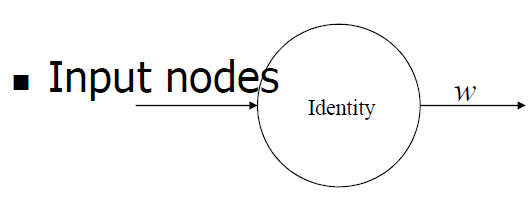
X2

X1

### 신경망 모델의 구조

* 

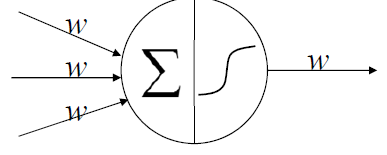
## [중요] 신경망 노드의 출력값 계산

* 입력 노드의 경우  
  

X

출력 = w \* x

* Hidden Layer 노드의 경우,



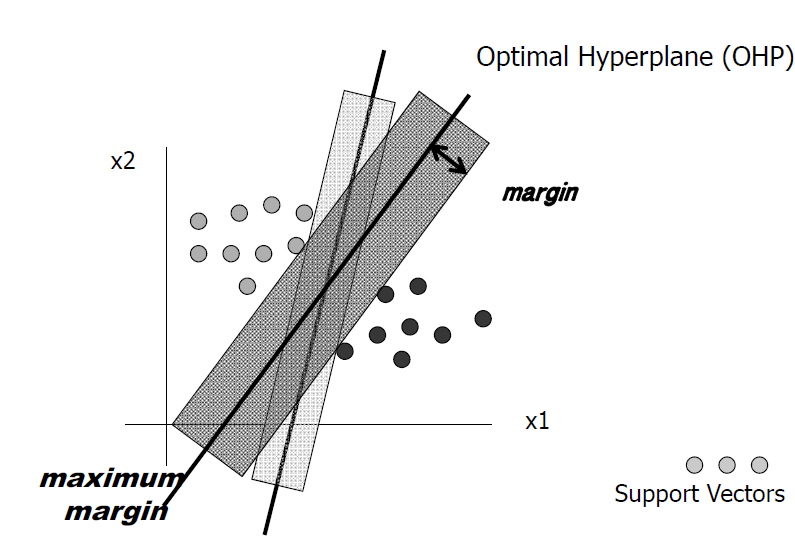
Left 출력 =

Right 출력에서는 활성화 함수(Activation Function)의 적용: Sigmoid =   
Right 출력 =

## 유의사항

* 분산이 적은 연속형 데이터, 또는 범주별로 빈도수가 비슷한 데이터에 가장 효과적이다.
* Hidden layer/node가 증가할수록 overfitting할 가능성이 높아지므로
* 먼저 hidden layer가 하나도 없는 상태에서 시작한 후, validation dataset에 대해 에러율이 낮아질 때까지 hidden layer/node를 추가시켜 나가는 점진적인 방식으로 적용해 본다.

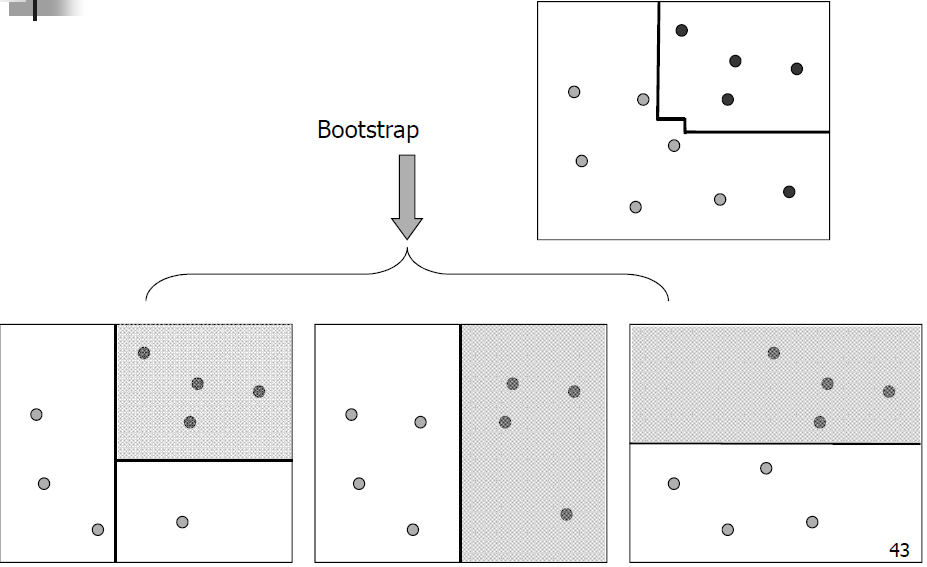
# Support Vector Machine



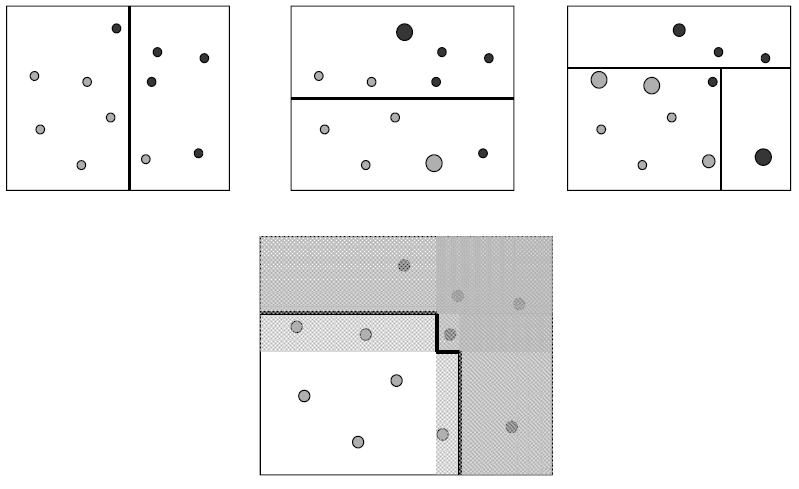
# 앙상블(Ensemble) 기법

* 다수의 방법을 사용하여 나온 결과를 다수결에 따라 결정하는 방법
* 앙상블에 사용할 각 방법이 정확도가 낮은 경우에만 효과를 볼 수 있음.
* 만약 이미 정확도가 높은 방법이라면, 앙상블을 쓰더라도 효과가 크지 않음
* 일반적으로 Decision Tree를 활용함
* 주요 기법에는 Bagging, Boosting이 있음

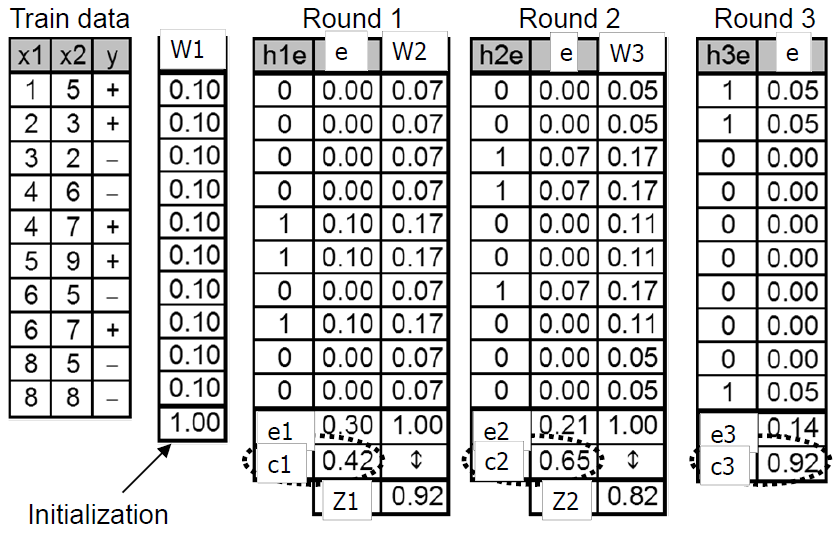
## Bagging

* 복원 추출(bootstrap)
* 
* 3개의 영역에서 2개의 영역이 겹치는 부분을 선택

## Boosting

* 가중치를 적용
* 오분류 항목에 대해서 높은 가중치를, 나머지 항목에 대해서는 낮은 가중치를 적용
* 

## Adaboost의 사례

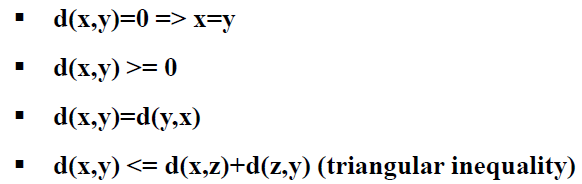
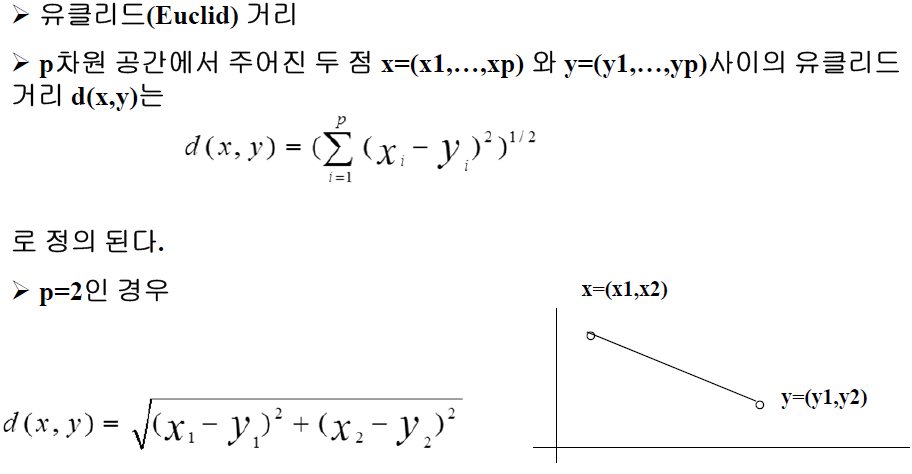
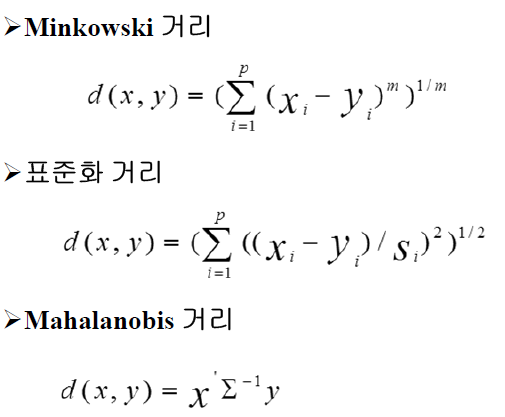
* 

# 비교사 학습(Unsupervised Learning)

## 클러스터링(Clustering)

* 동일한 군집ㅂ에 속하는 개체는 여러 속성이 서로 비슷하고, 서로 다른 군집에 속한 개체는 그렇지 않도록 군집을 구성한다.

## 거리(Distance)

* 유사도를 측정하는 척도
* 거리의 정의  
  
* 거리의 종류
* 
* 

## 계층적 군집 분석(Hierarchical Cluster Method)

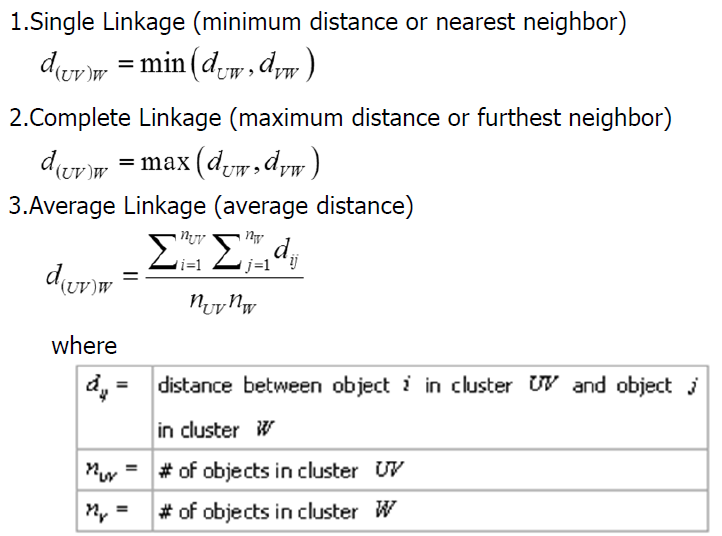
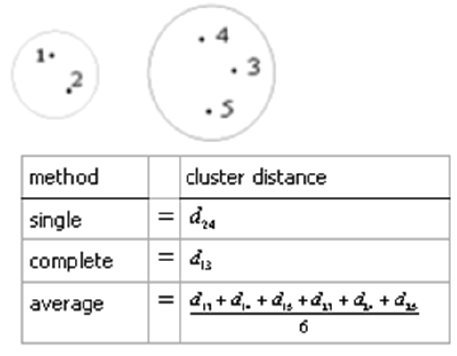
### 병합 계층 군집화(Agglomerative Hierarchical Method)

* 단일 개체로부터 시작하여 서로 유사한 개체끼리 병합하는 방법

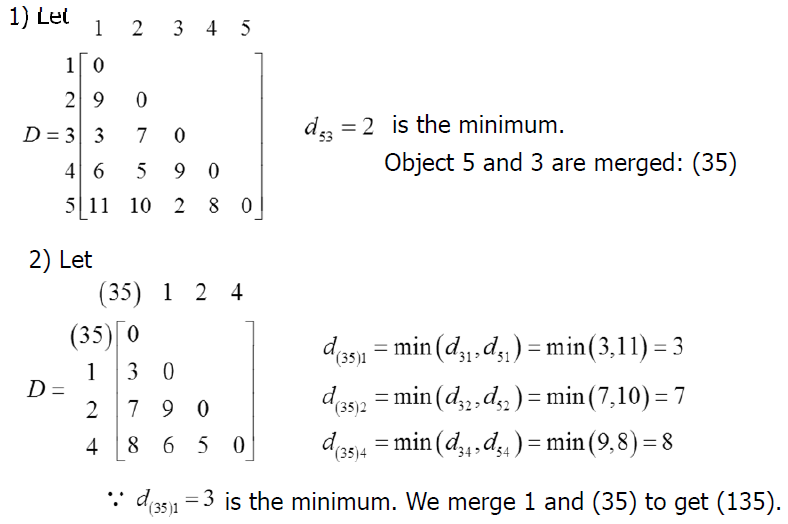
### 분할 계층 군집화(Divisive Hierarchical Method)

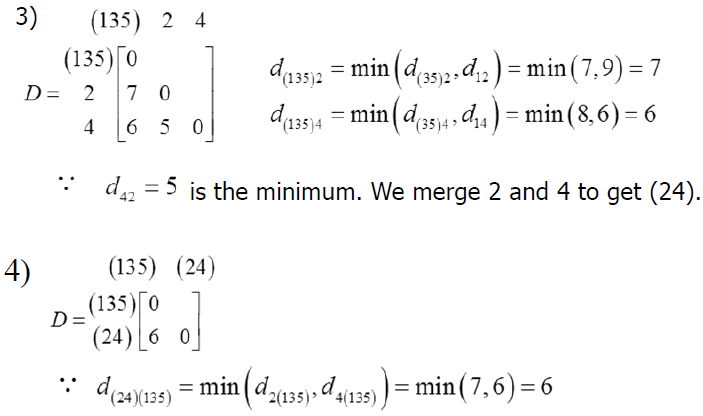
* 단일 그룹에서 시작하여, 두 개의 하위 그룹으로 분할하는 방법

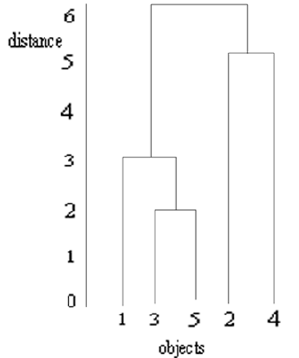
## 거리 계산 방법

* 
* 예제
* 

## [중요] Single Linkage를 사용한 군집화





* dendrogram으로 시각화
* 

# K-means clustering

## 특징

* 사전에 결정된 군집수 k에 기반하여 전체 데이터를 유사한 k개의 군집으로 분할한다.
* 평균만 계산하기 때문에, 대용량 데이터를 빠르게 처리할 수 있다.
* 군집수 K는 반복적으로 K를 달리 사용해 봐서 가장 적합한 K를 결정한다.

## 알고리즘

* 군집수 k를 선택한다.
* 초기 K개 군집의 중심을 선택한다.
* 각 개체를 K개의 중심 중 가장 가까운 거리에 있는 군집에 할당한다.
* 중심을 새로 계산하여, 새로운 중심과 기존의 중심이 차이가 없을 때까지 위의 과정을 반복한다.

## 유의사항

* 군집 분석은 자료 사이의 거리를 이용하기 때문에, 각 자료의 단위가 결과에 큰 영향을 미친다. 따라서 각 변수의 단위를 표준화하여야 한다.
* 좋은 결과란, 각 군집 안에서 분산이 최소화 되는 것.
* 군집화의 목적은 해당 군집이 어떤 변수에 의해서 형성되었는지를 파악하기 위해서다.
* 찾아진 군집이 무엇을 의미하는지를 데이터만 이용해서는 해석이 어렵다.

# 연관성 분석(Association Analysis)

## 연관성 분석이란

* 데이터 안에 존재하는 항목간의 연관 규칙을 발견하는 과정
* 장바구니 분석이라고도 부른다.

## 연관 규칙

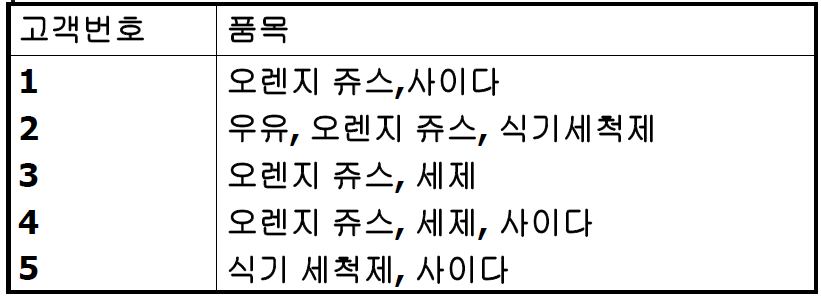
* If A, then B 형태
* 모든 연관 규칙이 유용한 것은 아니다.

자명한 규칙 : 대다수의 사람들이 이미 알고 있는 규칙으로 효용성이 없다.

설명이 불가능한 규칙 : 세밀한 조사가 필요

## 동시 구매표

* 각 품목별로 동시에 구매한 품목들에 대한 매트릭스
* 모든 품목에 대해 동시 구매표를 만들게 되면 생성이 오래 걸리므로, 관심 품목을 한정해서 동시 구매표를 작성해야 한다.
* 동시구매표의 예

<거래내역>  


<동시 구매표>



* 두 상품이 몇번이나 함께 팔렸는지 확인 가능
* 대각선상의 숫자는 해당 품목을 포함하는 총 거래수를 의미. 즉 오렌지 주스를 산 총 거래수는 4다.

## 지지도(Support)와 신뢰도(Confidence)

규칙이 유용하기 위해서는 일정 이상의 지지도와 신뢰로를 만족해야 한다.

If A, then B 의 규칙에서,

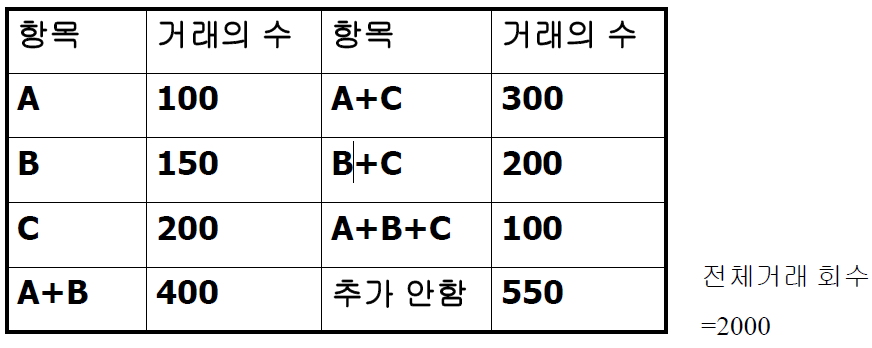
### 지지도(support)

* 지지도(A,B) = 전체 거래 중, 품목 A와 품목 B를 동시에 포함하는 거래의 비율  
   = 품목 A와 B를 동시에 포함하는 거래 수 / 전체 거래수  
   = P(A^B)

### 신뢰도(confidence)

* 신뢰도(A,B) = A를 포함하는 거래 중, 품목 A와 품목 B를 동시에 포함하는 거래의 비율  
   = 품목 A와 B를 동시에 포함하는 거래 수 / 품목 A를 포함하는 거래 수
* = P(A^B) / P(A) = P(B | A)

## [중요] 지지와 신뢰도 계산 예제

* 동시 거래 내역이 다음과 같을 때,
* 

C

B

A

100

200

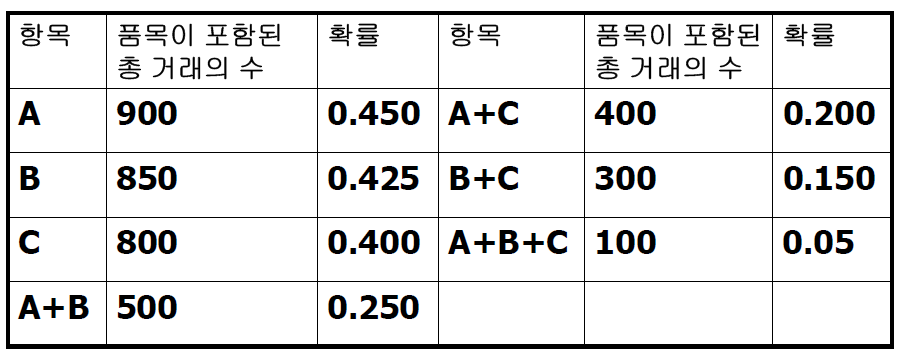
300

400

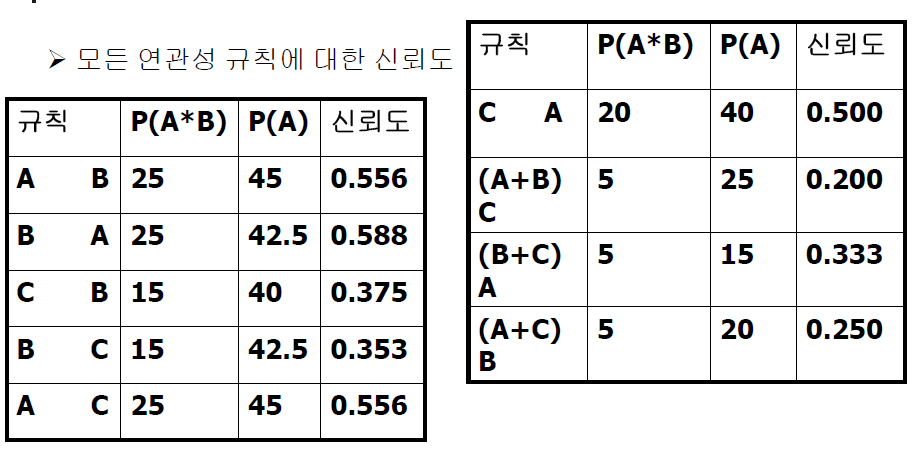
200

150

100

* 지지도 계산
* 

지지도(A+B) = (400 + 100) / 2000 = 0.25

* 신뢰도 계산  
  

A, then B의 신뢰도 = P(A^B) / P(A) = 0.25 / 0.45 = 0.556

(B + C), then A의 신뢰도 = P(B^C^A) / P(B^C) = 0.05 / 0.15 = 0.333

## 향상도(Lift)

### 지지도와 신뢰도의 한계

* If (B + C), then A의 신뢰도는 0.333이다. 하지만 이 규칙은 유용하지 못한데, 전체 거래에서 A가 일어날 확률(즉 조건 자체가 없더라도 A를 구매할 확률)\_)이 0.45로 더 크기 때문이다.
* 이처럼 연관성 규칙의 유의미성을 파악하려면 해당 규칙이 조건이 없을 때에 비해 얼마나 향상시킬 수 있는지를 측정해야 한다.

### 향상도

* If A, then B 규칙의 향상도 = P(B | A) / P(B)
* 향상도가 클수록, 품목 A의 구매 여부가 품목 B의 구매 여부에 큰 영향을 미치게 된다.
* 향상도가 1이면, P(B | A) = P(B), 즉 A의 구매 여부가 B의 구매여부에 영향을 전혀 미치지 않는다는 뜻이다.
* 따라서 향상도가 1보다 큰 값을 가지는 규칙만이 유의미하다.
* 