[시험용 정리] Big Data Analytics : Data Mining

- 연세대 김현중 교수님 자료-

목차

1. 데이터 마이닝

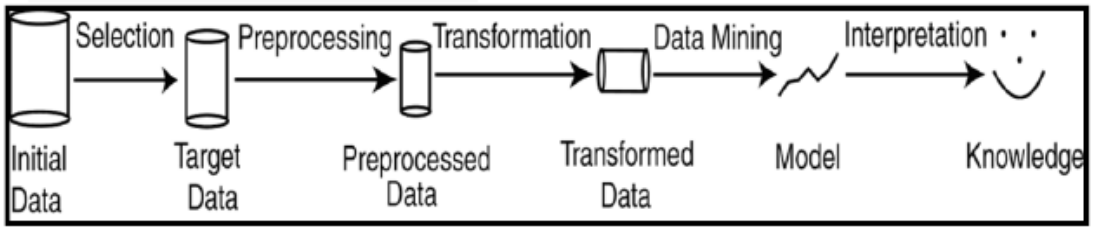
● 데이터 마이닝이란

- 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

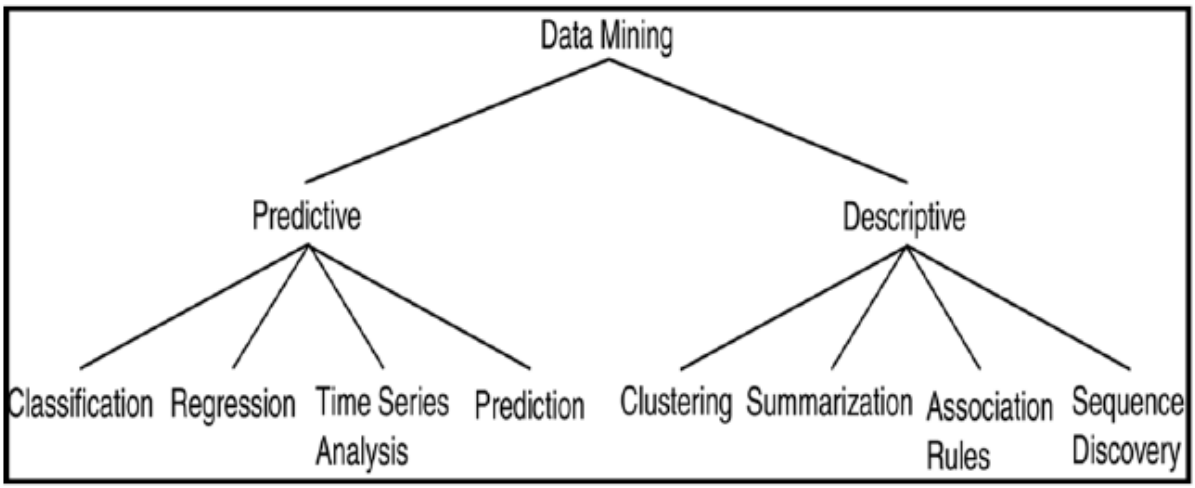


Modified from [FPSS96C]

- 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

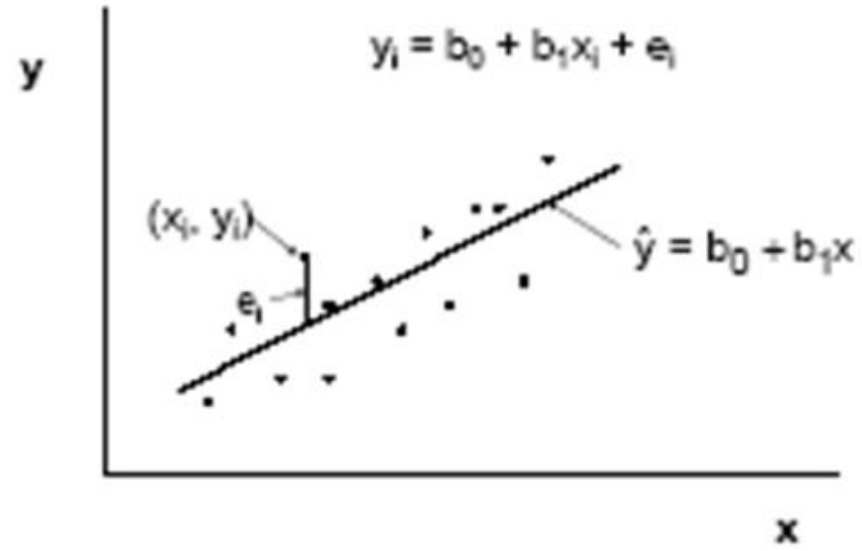
2. Linear Regression

- **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**

Source	DF	SS	MS	F	p
Regression	1	SSR	MSR	F	p-value
Error	n-2	SSE			
Total	n-1	SST			

where MSR = SSR/dfR = SSR/1
MSE = SSE/dfE = SSE/(n-2)
F = MSR/MSE

- **가설 검증(F-Statistic)**

귀무가설 $H_0: \beta_1 = \beta_2 = \bullet\bullet\bullet = \beta_k = 0$ (반드시 =이 있어야 함)
대립가설 $H_1: H_0$ is false (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.

$H_0: \beta_i = 0$ => xi 변수가 중요하지 않다는 의미
$H_1: \beta_i \neq 0$ => xi 변수가 유의미하다는 의미

- **Linear Regression의 해석**

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	6550.96274	1091.82712	70.12	<.0001
Error	93	1448.03726	15.57029		
Corrected Total	99	7999.00000			

Root MSE	3.94592	R-Square	0.8190
Dependent Mean	46.10000	Adj R-Sq	0.8073
Coeff Var	8.55948		

Parameter Estimates

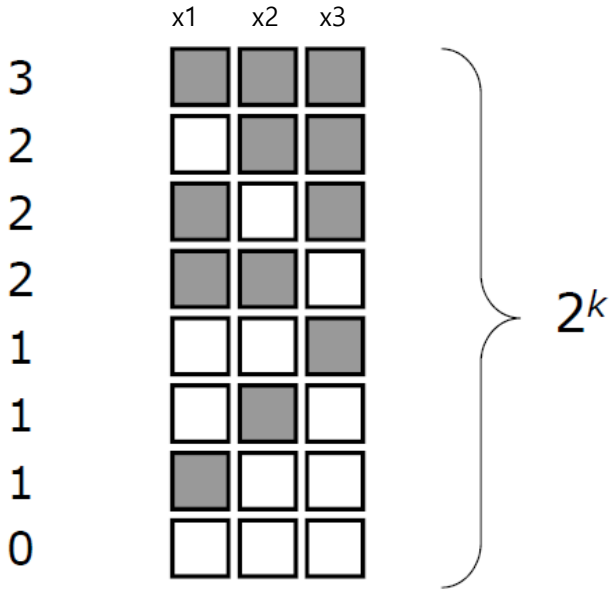
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	3.53938	4.12105	0.86	0.3926
X1	Delivery Speed	1	2.41862	0.45002	5.37	<.0001
X2	Price Level	1	2.70770	0.44582	6.07	<.0001
X3	Price Flexibility	1	1.90715	0.44659	4.27	<.0001
X4	Manufacturer Image	1	0.78675	0.37051	2.12	0.0364
X13	Type of Industry	1	-1.64286	0.79695	-2.06	0.0421
X14	Type of Buying Situation	1	4.64784	0.82855	5.61	<.0001

개별적인 x 변수의 유의성

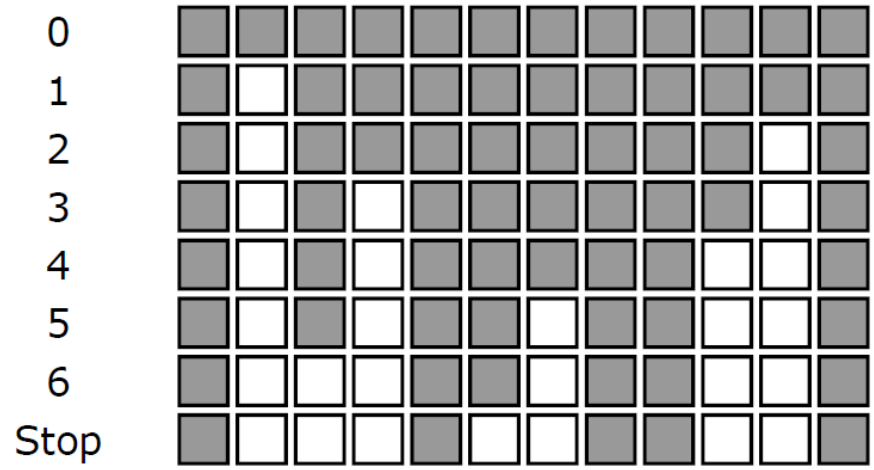
p-value가 작을수록 중요한 변수

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

- **변수선택 알고리즘**
  - ✓ **All Subsets**



✓ **Backward Elimination**

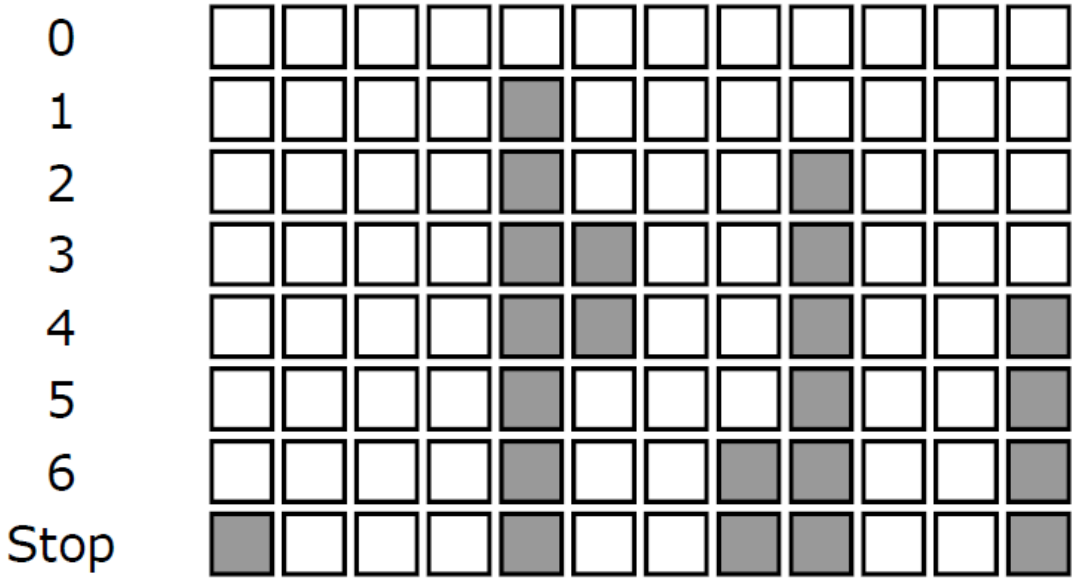


- 모든 변수를 포함하는 모델로부터 시작한다.
- 가장 높은 p-value를 가지는 변수를 제거한다.
- 남아있는 변수가 모두 유의할 때까지 작업을 계속한다.
- 빅데이터에 가장 적합한 방식이다.

✓ **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.

3. Logistic Regression

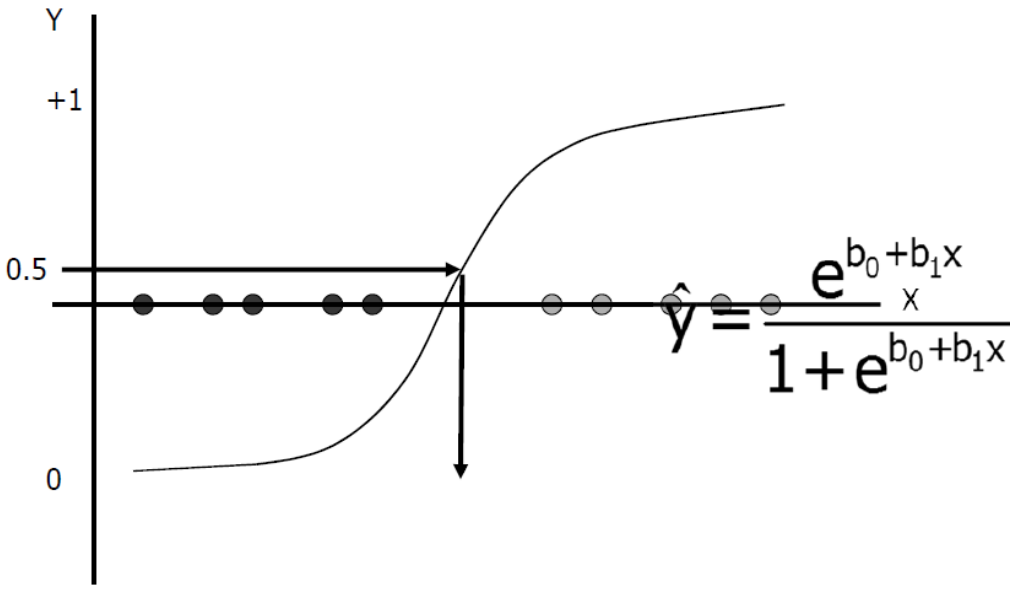
● **분류(classification)란**

- 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**

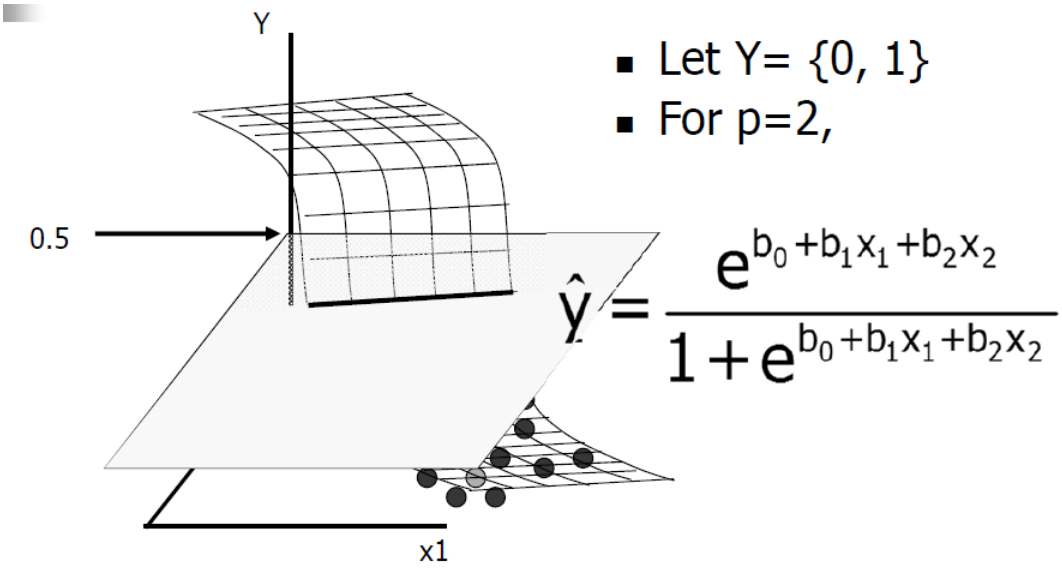
- Logistic Regression
- Discriminant Analysis
- K-nearest neighbors
- Decision Tree
- Nural Networks
- Support Vector Machine

- **1변수에서의 Logistic Regression**

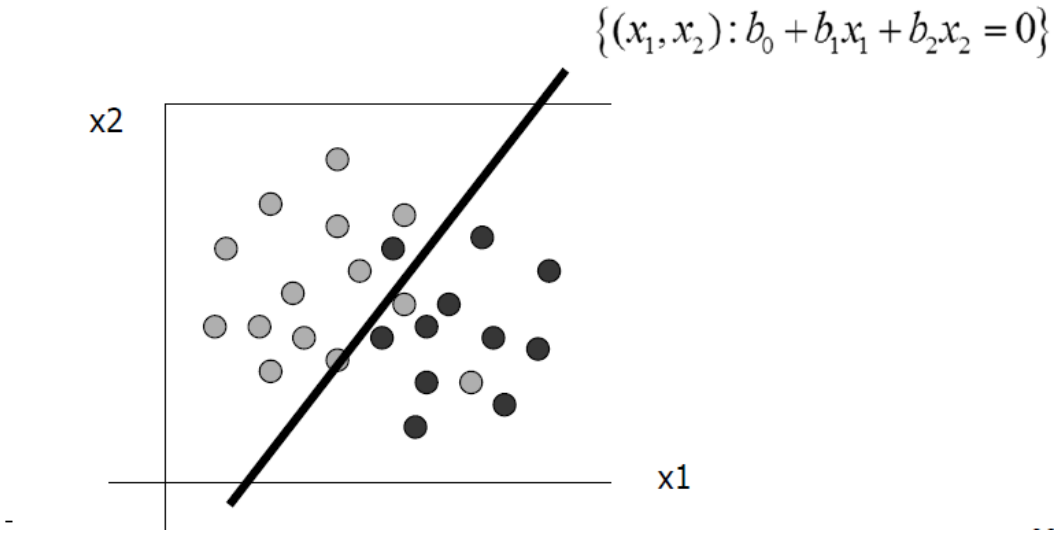


- $\text{logit}(\hat{y}) = b_0 + b_1x = \log\left(\frac{\hat{y}}{1-\hat{y}}\right)$
- $\hat{y} = \frac{e^{b_0+ b_1x}}{1+ e^{b_0+ b_1x}}$

- **2변수에서의 Logistic Regression**



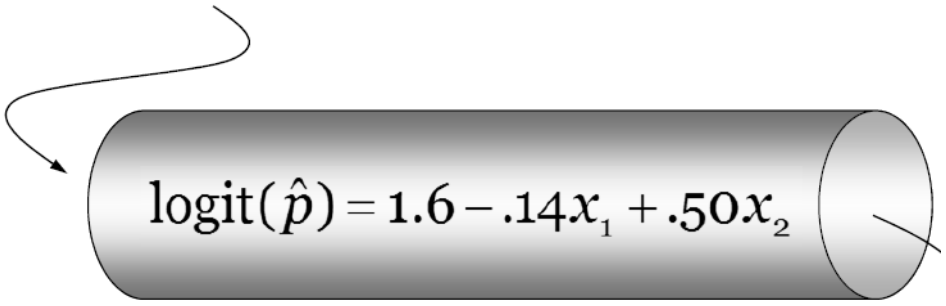
-
- Hyperplane at y = 0.5



- **[중요] 새로운 사례에 대한 스코어 계산하기**

아래와 같이 x와 logistic 함수가 주어진 경우, p 값을 구하라

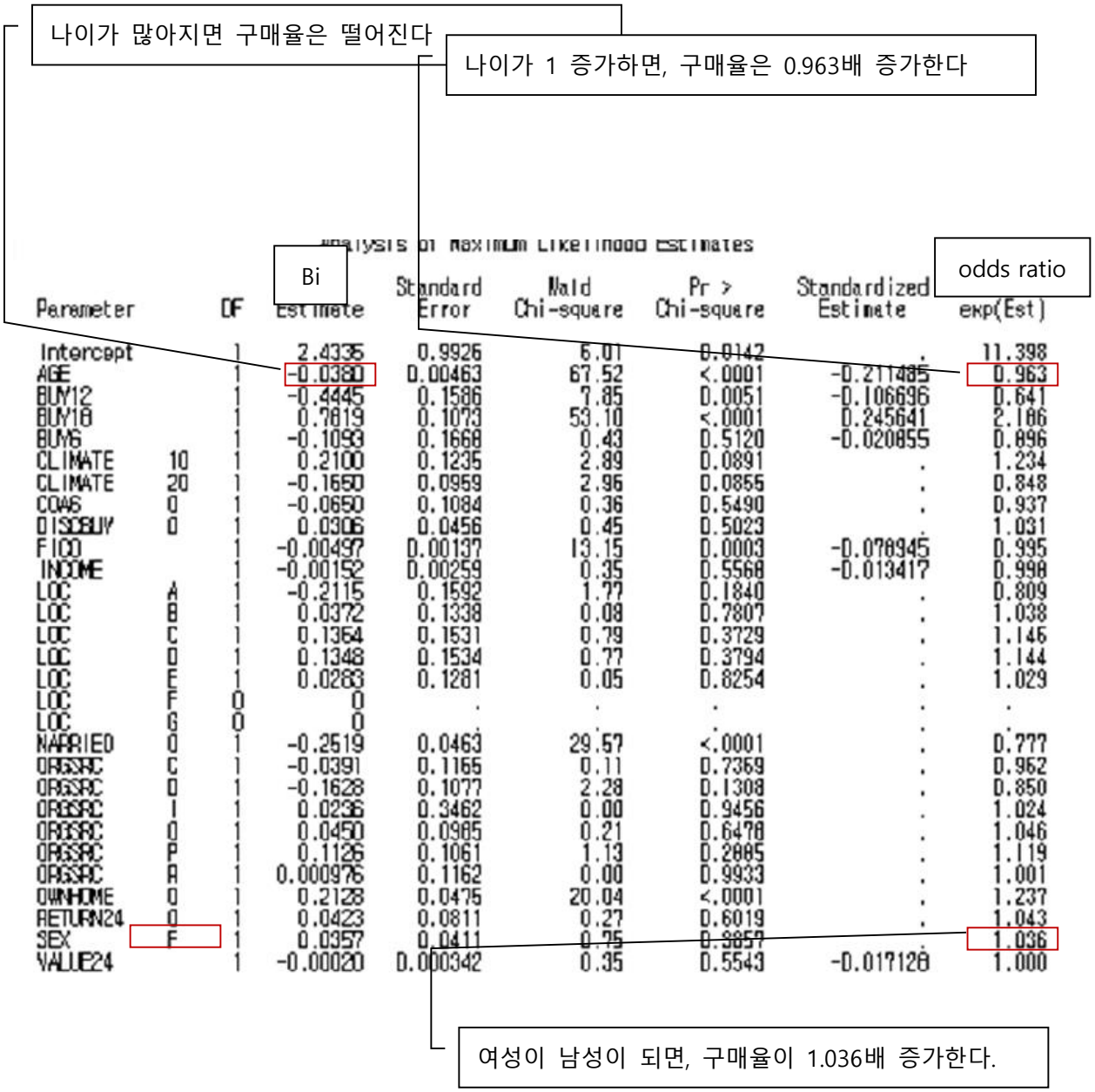
$$\mathbf{x} = (1.1, 3.0)$$



<풀이>

$$\hat{p} = \frac{e^{\text{logit}(\hat{p})}}{1 + e^{\text{logit}(\hat{p})}}$$

- **Logistic Regression의 해석**



Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate	exp(Est)
Intercept		1	2.4335	0.9926	6.01	0.0142	.	11.398
AGE		1	-0.0380	0.00463	67.52	<.0001	-0.211485	0.963
BUY12		1	-0.4445	0.1586	7.85	0.0051	-0.106696	0.641
BUY18		1	0.7619	0.1073	53.10	<.0001	0.245641	2.186
BUY6		1	-0.1093	0.1668	0.43	0.5120	-0.020855	0.896
CLIMATE	10	1	0.2100	0.1235	2.89	0.0891	.	1.234
CLIMATE	20	1	-0.1650	0.0959	2.96	0.0855	.	0.848
COA6	0	1	-0.0650	0.1084	0.36	0.5490	.	0.937
DISCBUY	0	1	0.0306	0.0456	0.45	0.5023	.	1.031
FICO		1	-0.00497	0.00137	13.15	0.0003	-0.078945	0.995
INCOME		1	-0.00152	0.00259	0.35	0.5568	-0.013417	0.998
LOC	A	1	-0.2115	0.1592	1.77	0.1840	.	0.809
LOC	B	1	0.0372	0.1338	0.08	0.7807	.	1.038
LOC	C	1	0.1364	0.1531	0.79	0.3729	.	1.146
LOC	D	1	0.1348	0.1534	0.77	0.3794	.	1.144
LOC	E	1	0.0283	0.1281	0.05	0.8254	.	1.029
LOC	F	0	0	.	.	.	.	.
LOC	G	0	0	.	.	.	.	.
MARRIED	0	1	-0.2519	0.0463	29.57	<.0001	.	0.777
ORGSRC	C	1	-0.0391	0.1165	0.11	0.7369	.	0.962
ORGSRC	D	1	-0.1628	0.1077	2.28	0.1308	.	0.850
ORGSRC	I	1	0.0236	0.3462	0.00	0.9456	.	1.024
ORGSRC	O	1	0.0450	0.0985	0.21	0.6478	.	1.046
ORGSRC	P	1	0.1126	0.1061	1.13	0.2885	.	1.119
ORGSRC	R	1	0.000976	0.1162	0.00	0.9933	.	1.001
OWNHOME	0	1	0.2128	0.0475	20.04	<.0001	.	1.237
RETURN24	0	1	0.0423	0.0811	0.27	0.6019	.	1.043
SEX	F	1	0.0357	0.0411	0.75	0.3857	.	1.036
VALUE24		1	-0.00020	0.000342	0.35	0.5543	-0.017128	1.000

- **OR(Odds Ratio)**
- odds = $\frac{\text{probability that some event will occur}}{\text{probablility that some event will not occur}}$
- odds ratio의 예

E.g. A=store owners, B=salesmen

- $P(default_A)=.25$, $P(default_B)=.10$
- $Odds(default_A)=1/3$, $Odds(default_B)=1/9$
- OR(A:B)=3
- Store owners have three times higher risk of becoming default than salesman

-

4. Logistic Regression과 OR

$$OR(A:B) = \frac{odds(A)}{odds(B)} = \frac{e^{\beta_0+\beta_1 x_A}}{e^{\beta_0+\beta_1 x_B}} = e^{\beta_1(x_A - x_B)}$$

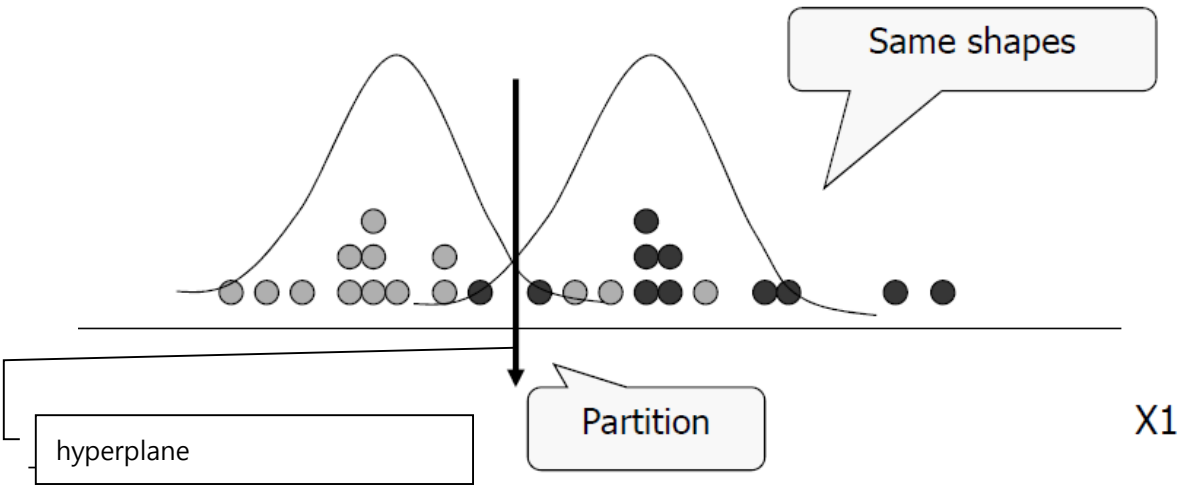
- $\log\left(\frac{p}{1-p}\right) = \log(odds) = \quad b0 + b1x$
- $odds = \quad e^{b0+b1x}$
- 만약 x 가 나이라면, 나이가 1증가할 때 마다 odds는 e^{b_1} 만큼 증가한다.
- 또한 odds ratio가 1이라는 뜻은, b1이 0이라는 뜻이므로, x 변수는 독립변수 y와 무관하다는 뜻이다.

5. 판별 분석(Discriminant Analysis)

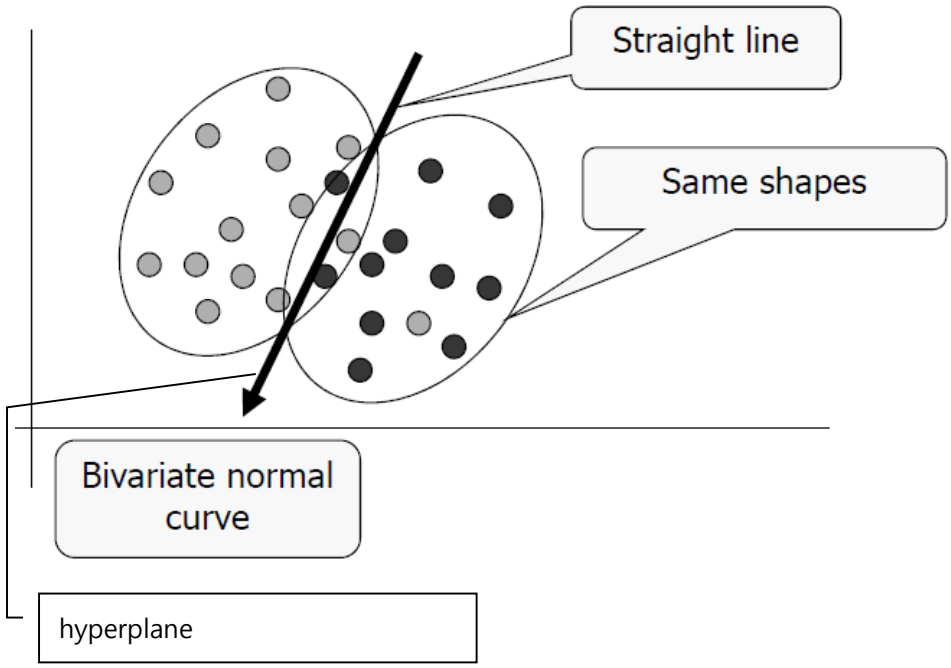
- 분류 방법 중 하나로, 정규분포를 가정한다.

- **LDA(Linear Discriminant Analysis)**

- 동일한 평균과 분산을 가지는 정규 분포를 가정
- 1차원,

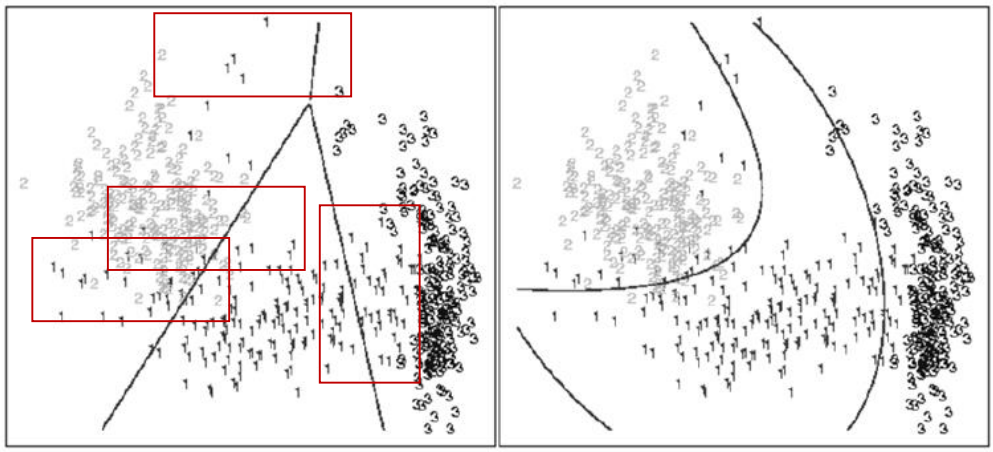


- 2차원



-

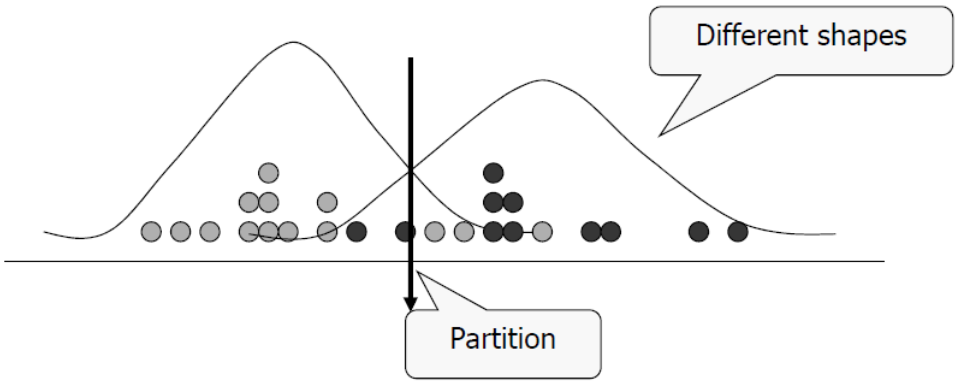
- LDA에서는 hyperplane이 직선(또는 평면)이다.
- 따라서 LDA에서는 오분류가 발생할 빈도가 높아진다. QDA의 hyperplane은 곡선이므로 오분류가 발생할 빈도가 낮아진다.



-

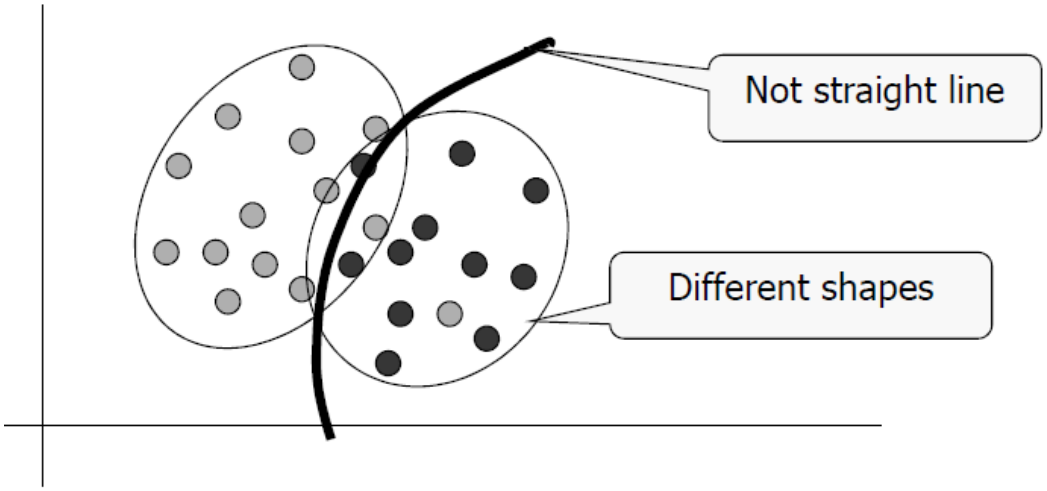
LDA QDA

- **QDA(Quadratic Discriminant Analysis)**
- 서로 다른 평균과 분산을 가지는 정규 분포를 가정
- 1차원



-

- 2차원,



-

6. K-nearest neighbor

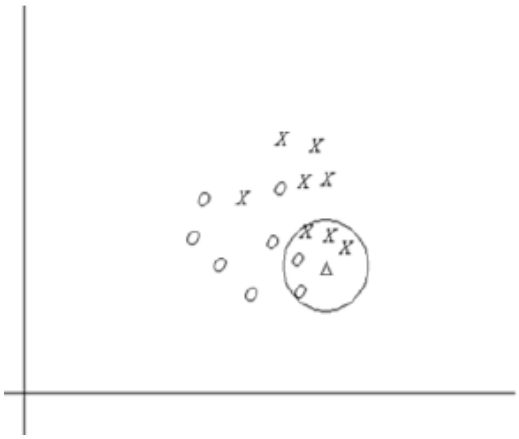
- **Nearest neighbor의 개념**

- 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**



K_1 belongs to group 1

K_2 belongs to group 2

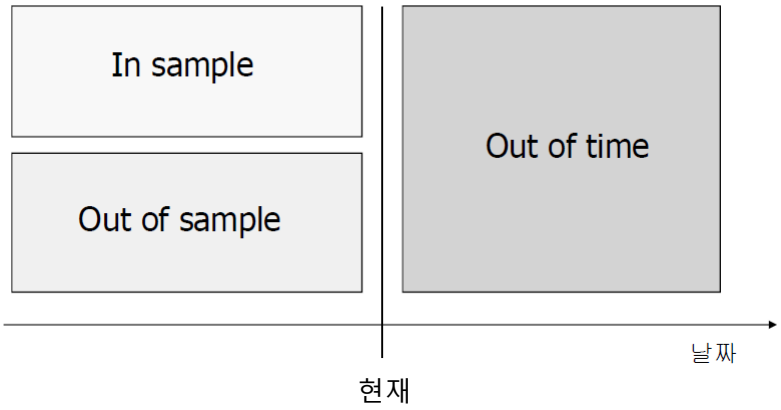
$K_1 + K_2 = K = 5$

-

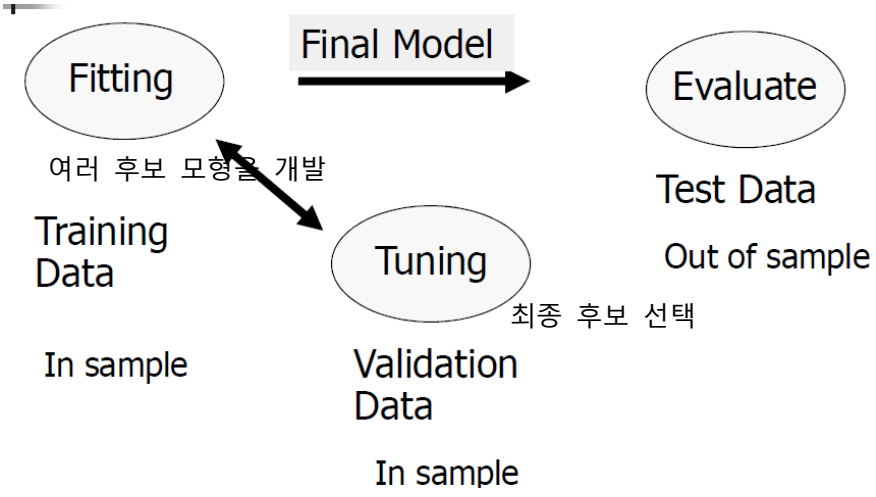
- 세모로부터 가장 가까운 5개의 데이터를 찾는다.
- 그 중에서 가장 많은 데이터를 가지는 이웃그룹에 세모를 할당한다.

7. Evaluation 기법

- **데이터의 역할**



- 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)

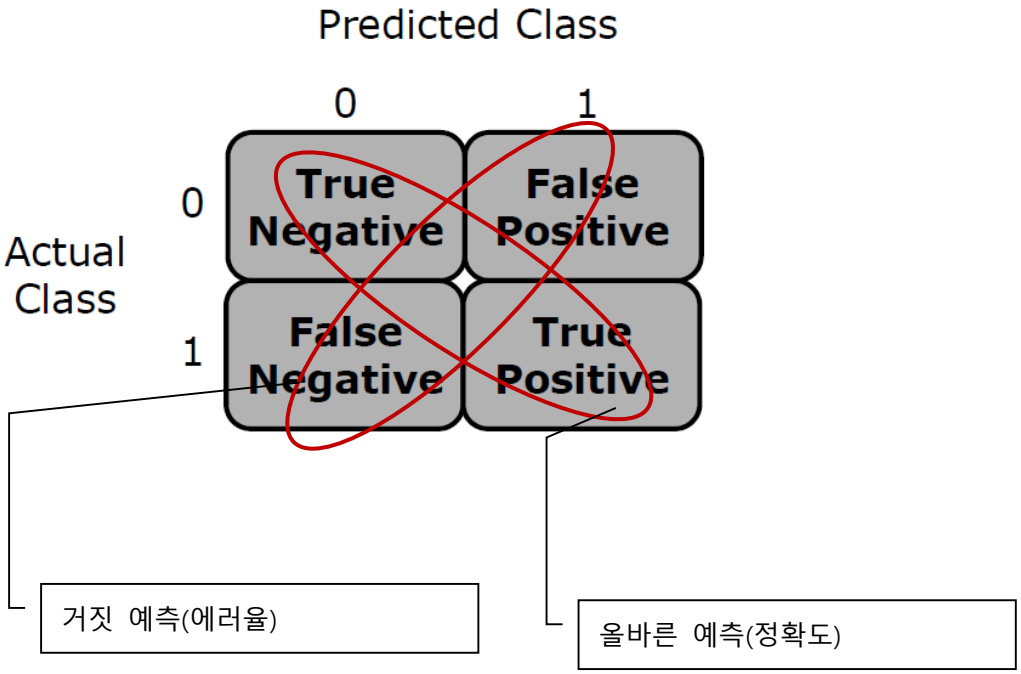


- 층화 추출(stratified sampling) : 모집단에서 랜덤으로 표본 추출을 할 경우, 사례가 적은 Bad는 모델링을 할 수 있을 정도로 충분히 추출하지 못할 수도 있다. 이 경우 전체 training data에서의 Good : Bad의 비율을 그대로 유지한 채 추출해야 한다.
- validation data는 모델을 최적화하거나, overfitting을 방지하기 위한 용도로 활용한다.

- **모델 비교 방법**

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

- **Confusion Matrix**



- 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

		Predicted		
		Non-Buyer	Buyer	
Actual	Non-buyer	8500	500	9000
	Buyer	500	500	1000
		9000	1000	10000

- 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%

		Predicted		
		Non-Buyer	Buyer	
Actual	Non-buyer	8500 TN	500	9000 N
	Buyer	500	500 TP	1000 P
		9000	1000	10000

8. 차트

- **Lift(Gain) 차트 만들기**

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

Rank	Predicted probability	Actual class
1	0.95	Yes
2	0.93	Yes
3	0.93	No
4	0.88	Yes
…	…	…

- 만약 모델이 제대로 만들어졌다면, 예측확률이 높은 사람들이 실제로 Y일 경우가 많을 것이다라는 가정에 기반한다.
- 전체 데이터를 십분위로 분할하여 아래와 같이 만든다.

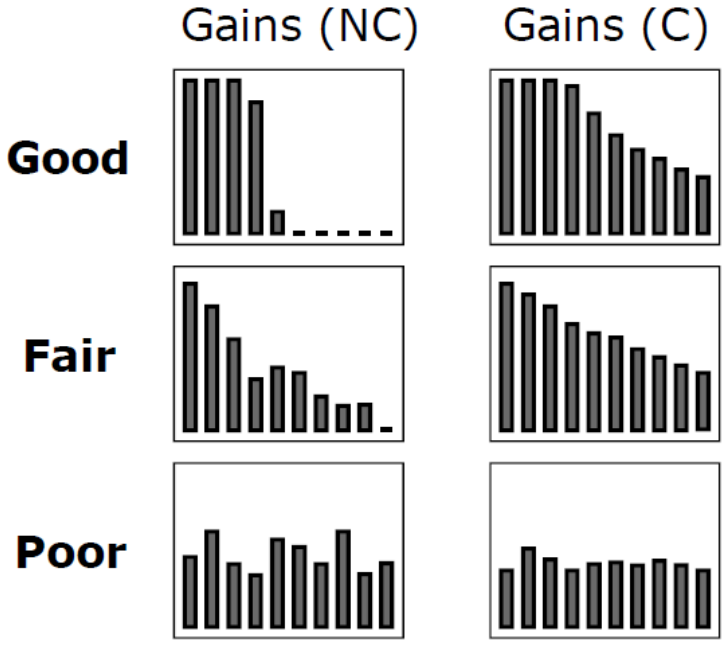
Decile	Frequency of "buy"	% Captured Response	% Response	Lift
1	174	174/381=45.6	174/200=87	87/19=4.57
2	110	110/381=28.8	110/200=55	55/19=2.89
3	38	38/381=9.9	38/200=19	19/19=1.00
4	14	14/381=3.6	14/200=7	7/19=0.36
5	11	11/381=2.8	11/200=5.5	5.5/19=0.28
6	10	10/381=2.6	10/200=5	5/19=0.26
7	7	7/381=1.8	7/200=3.5	3.5/19=0.18
8	10	10/381=2.6	10/200=5	5/19=0.26
9	3	3/381=0.7	3/200=1.5	1.5/19=0.07
10	4	4/381=1.0	4/200=2	2/19=0.10

- Baseline Lift = 381/2000=19.05%

- Frequency of "buy" : 200명 중 실제로 구매한 사람

- % Captured Response : 반응검출율
  = 해당 등급의 실제 구매자 / 전체 구매자

- % response : 반응률
  = 해당 등급의 실제 구매자 / 200명
- Lift: 향상도
  = 반응률 / 기본 향상도
- 좋은 모델이라면 Lift가 빠른 속도로 감소해야 한다.

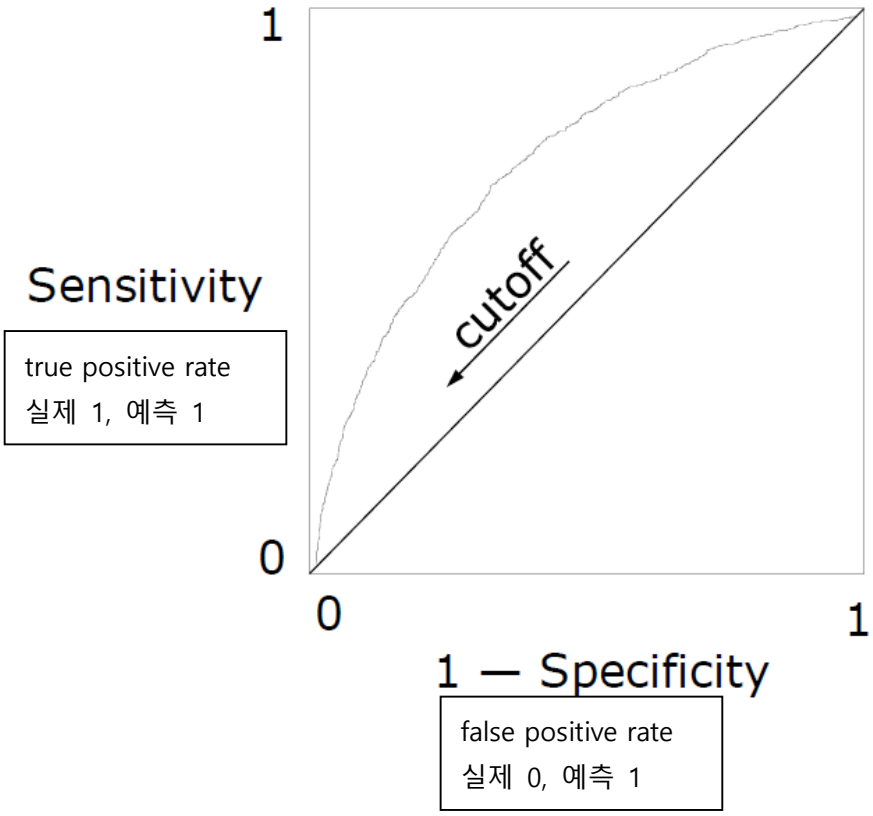


- **Lift를 통한 모델 비교하기 사례**

		Predicted 0	Predicted 1	Cutoff	Accuracy	Sensitivity	Lift
Actual	0	360	540	.08	44%	80%	1.3
	1	20	80				
Actual	0	540	360	.10	60%	60%	1.4
	1	40	60				
Actual	0	720	180	.12	76%	40%	1.8
	1	60	40				

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

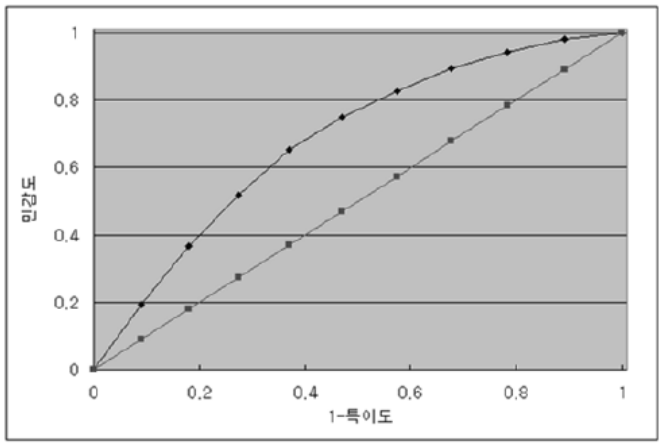
- **ROC 커브**



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

- **AUROC(Area Under ROC)**

등급	등급내인원	부도수	정상수	누적부도수	누적정상수	민감도	1-특이도	면적
1	1000	200	800	200	800	0.192308	0.089286	0.008585
2	1000	180	820	380	1620	0.365385	0.180804	0.025519
3	1000	160	840	540	2460	0.519231	0.274554	0.041466
4	1000	140	860	680	3320	0.653846	0.370536	0.056297
5	1000	100	900	780	4220	0.75	0.470982	0.070506
6	1000	80	920	860	5140	0.826923	0.573661	0.080958
7	1000	70	930	930	6070	0.894231	0.677455	0.089323
8	1000	50	950	980	7020	0.942308	0.783482	0.097361
9	1000	40	960	1020	7980	0.980769	0.890625	0.103022
10	1000	20	980	1040	8960	1	1	0.108323
총수	10000	1040	8960	1	1		AUROC=	0.681362



• **AUROC**

= (AR+1)/2

=

- **80%이상 – good**
- **75%이상 – moderate**
- **AR= 2*AUROC-100%**

- 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.

9. 의사결정 트리(Decision Tree)

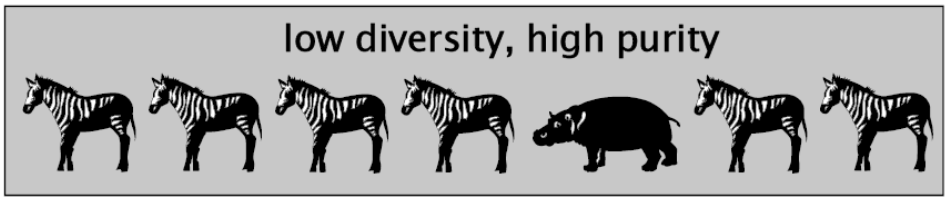
- Good 또는 Bad가 많이 모여 있는 부분을 찾는 방법
- 유사도(similarity)를 최대화할 수 있는 방향으로 입력 변수를 분할한다.
- 장점 : 해석이 쉽다 / 범주형 변수를 가변수화 하지 않고 그대로 사용할 수 있다.
- 단점 : 정확도가 떨어진다.

● Gini Impurity

그룹이 유사한 정도를 측정하는 방법

$$1-\sum_{j=1}^{r} p_j^2 = 2\sum_{j<k} p_j p_k$$

Pr(interspecific encounter) = $1-2(3/8)^2-2(1/8)^2 = .69$



Pr(interspecific encounter) = $1-(6/7)^2-(1/7)^2 = .24$

- 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**

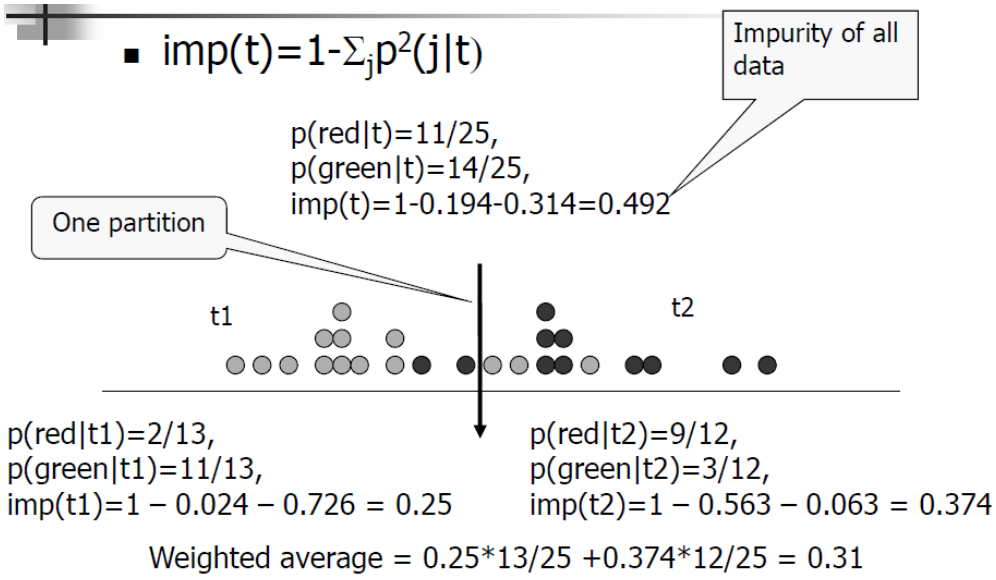
- 분할 정복
- 한번에 하나의 변수만 사용한다.
- 이진 트리
- Greedy search : 가능한 모든 조건을 검색한다

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

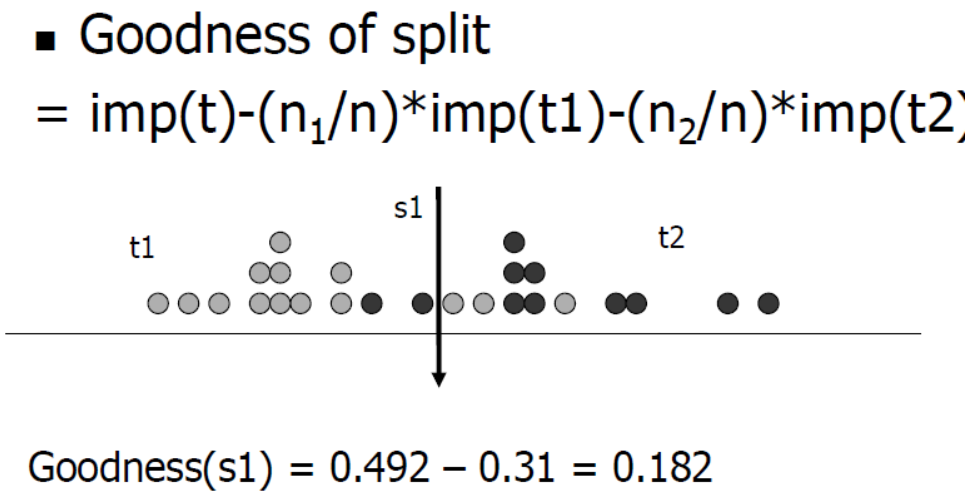
- Greedy Search

- **CART의 방식: Split이 많다면, 어느 split을 선택해야 하나?**

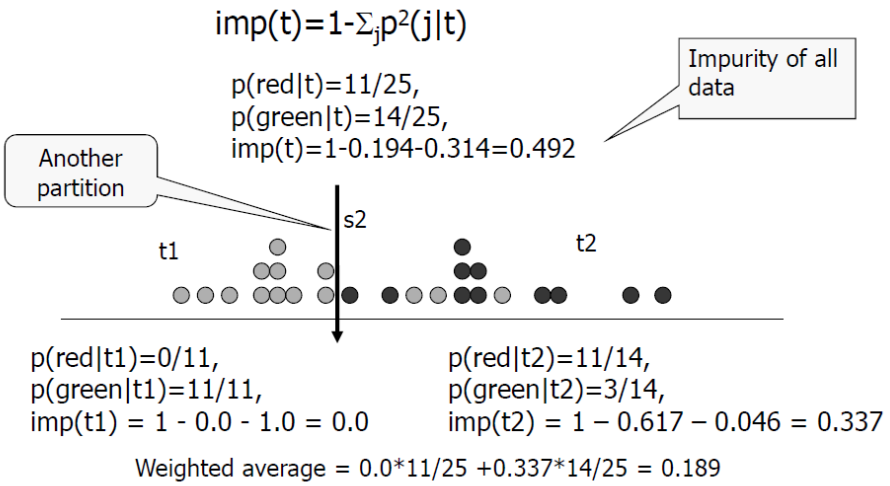
- Gini Impurity가 가장 낮은 split을 선택한다.

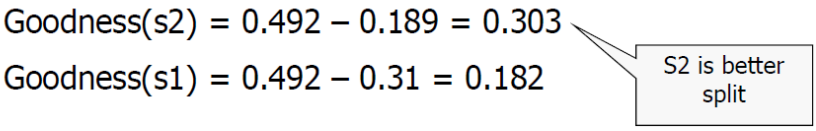


-
-



-





-

- **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**

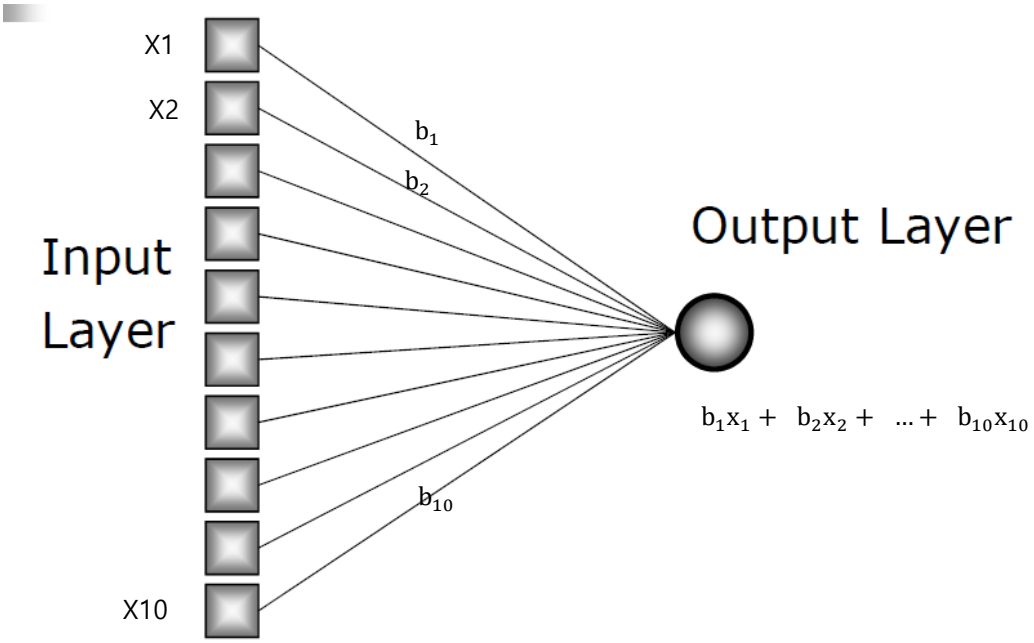
- 중요하지 않은 변수를 필터링하기 위한 목적으로 Decision Tree를 사용할 수 있다.
- Decision Tree의 split 정보를 기준으로 변수를 선택할 수 있다.

-

10.신경망(Neural Networks)

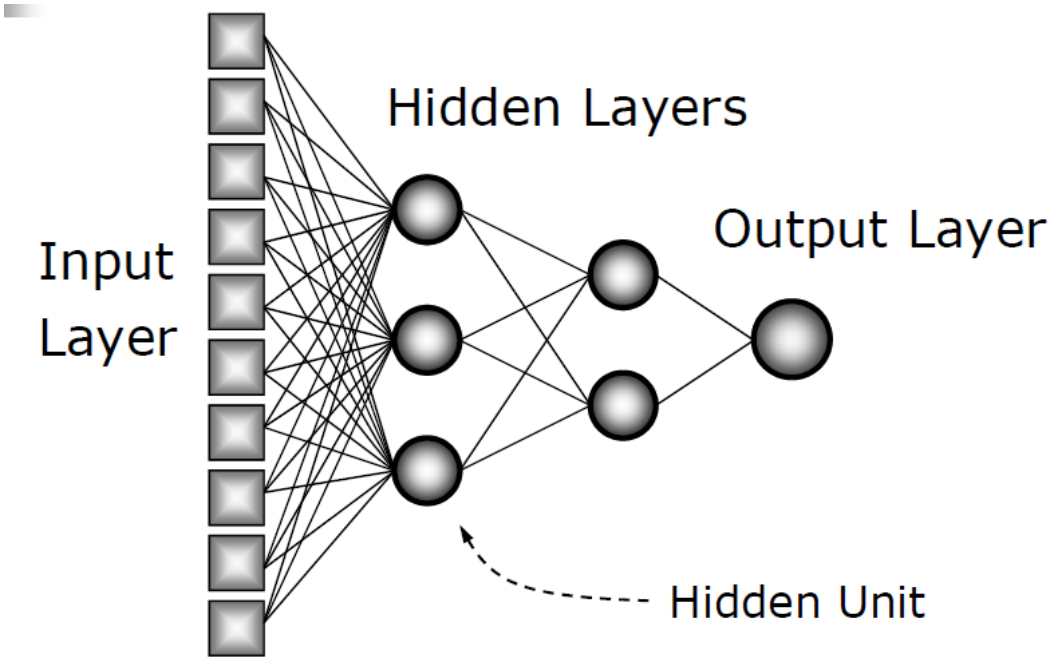
- 비선형 통계 모델
- Universal approximator
- 장점 : 정확도가 높다
- 단점 : 해석력이 낮다.
  (Decision Tree의 정반대)

- **아키텍처**
  - ✓ **전형적인 선형 모델의 구조**



-

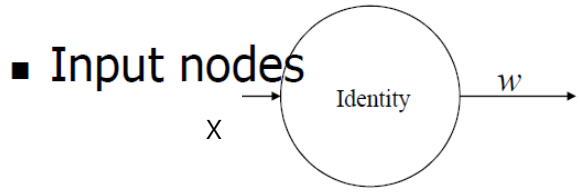
✓ **신경망 모델의 구조**



-

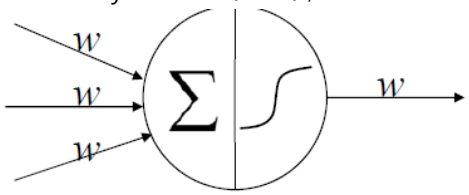
- **[중요] 신경망 노드의 출력값 계산**

- 입력 노드의 경우



출력 = w * x

- Hidden Layer 노드의 경우,



Left 출력 = $w_1x_1 + w_2x_2 + w_3x_3$

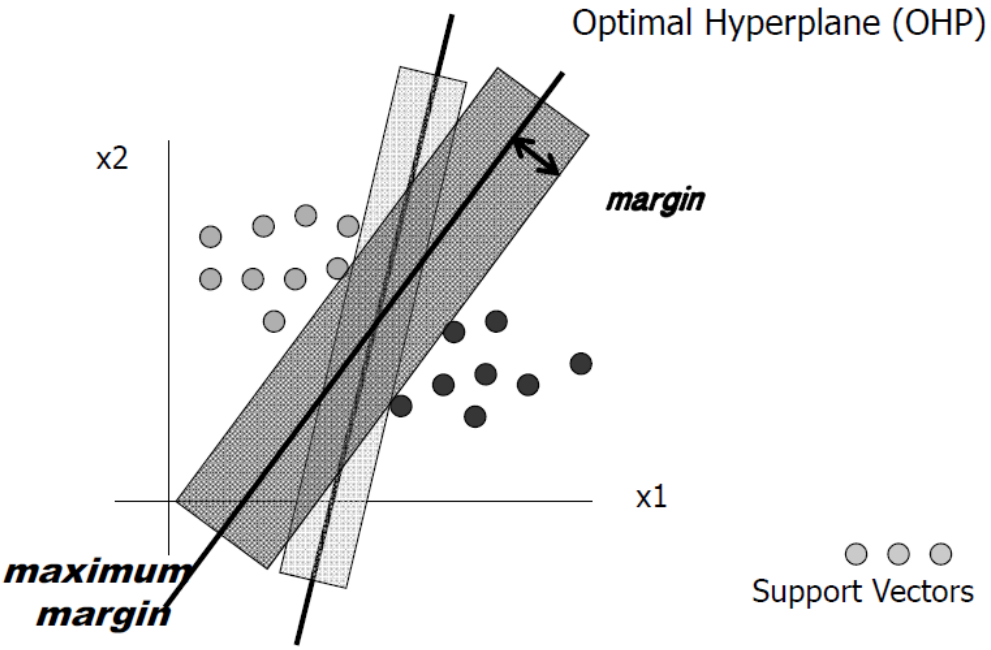
Right 출력에서는 활성화 함수(Activation Function)의 적용: Sigmoid = $\frac{e^x}{1+ e^x}$

Right 출력 = $\frac{e^{w_1x_1+ w_2x_2+ w_3x_3}}{1+ e^{w_1x_1+ w_2x_2+ w_3x_3}}$

- **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.

11.Support Vector Machine

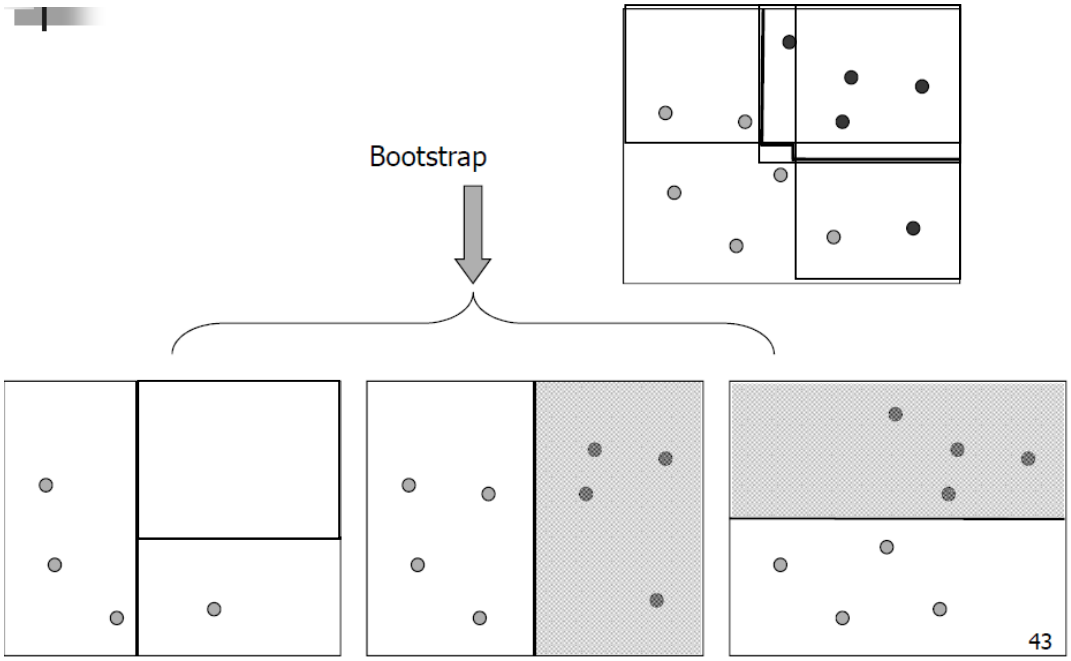


12.앙상블(Ensemble) 기법

- 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

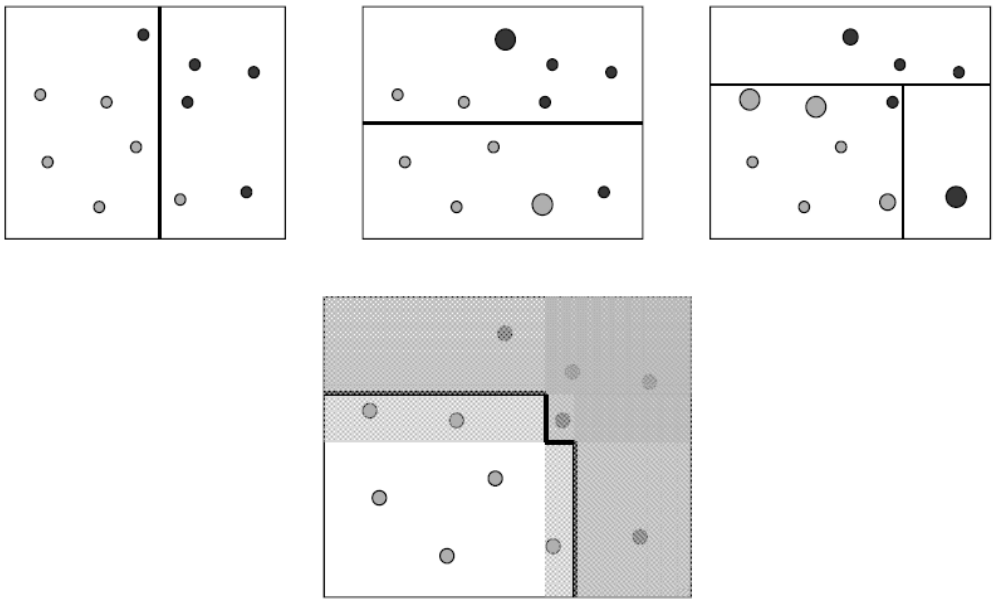
- **Bagging**

- 복원 추출(bootstrap)

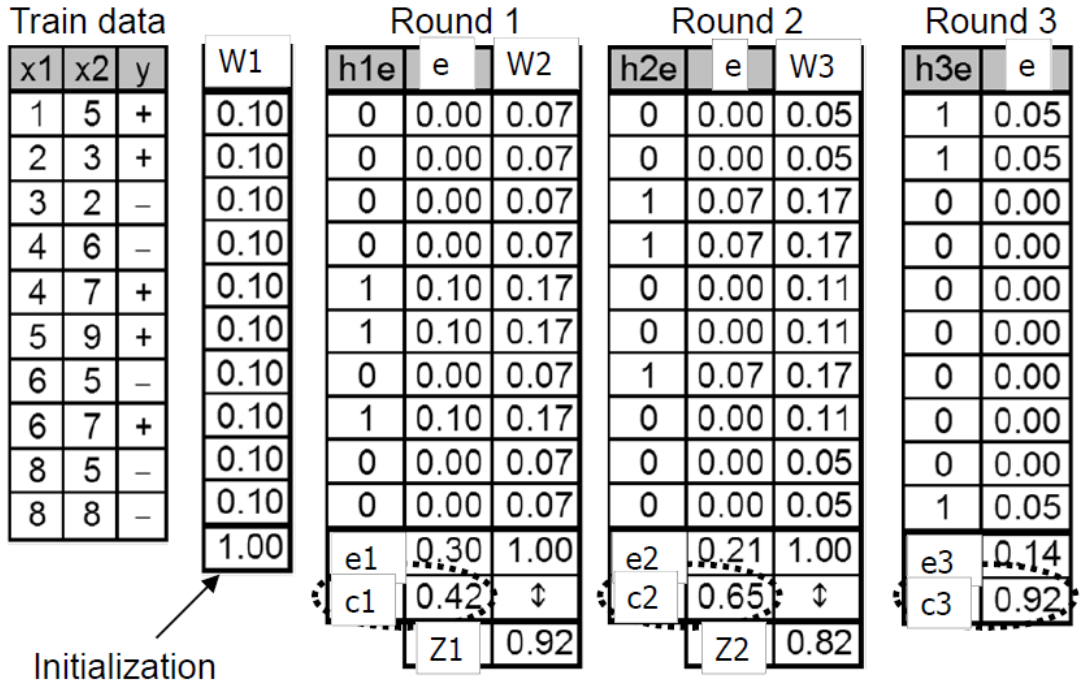
- 

- 3개의 영역에서 2개의 영역이 겹치는 부분을 선택

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

- 

- **Adaboost의 사례**



-

13.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
  - **d(x,y)=0 => x=y**
  - **d(x,y) >= 0**
  - **d(x,y)=d(y,x)**
  - **d(x,y) <= d(x,z)+d(z,y) (triangular inequality)**
- 거리의 종류

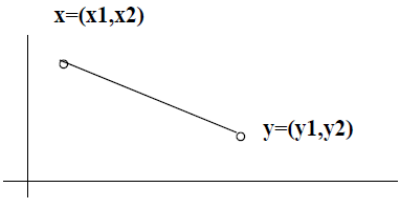
➢ 유클리드(Euclid) 거리

➢ p차원 공간에서 주어진 두 점 x=(x1,…,xp) 와 y=(y1,…,yp)사이의 유클리드 거리 d(x,y)는

$$d(x,y) = (\sum_{i=1}^{p} (x_i - y_i)^2)^{1/2}$$

로 정의 된다.

➢ p=2인 경우



$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

-

➢ **Minkowski** 거리

$$d(x,y) = (\sum_{i=1}^{p} (x_i - y_i)^m)^{1/m}$$

➢ 표준화 거리

$$d(x,y) = (\sum_{i=1}^{p} ((x_i - y_i)/s_i)^2)^{1/2}$$

➢ **Mahalanobis** 거리

$$d(x,y) = x^{'} \Sigma^{-1} y$$

-

- **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**

  1.Single Linkage (minimum distance or nearest neighbor)

  $$d_{(UV)W} = \min\left(d_{UW}, d_{VW}\right)$$

  2.Complete Linkage (maximum distance or furthest neighbor)

  $$d_{(UV)W} = \max\left(d_{UW}, d_{VW}\right)$$

  3.Average Linkage (average distance)

  $$d_{(UV)W} = \frac{\sum_{i=1}^{n_{UV}} \sum_{j=1}^{n_W} d_{ij}}{n_{UV} n_W}$$

  where

d_{ij} =	distance between object i in cluster UV and object j in cluster W
n_{UV} =	# of objects in cluster UV
n_V =	# of objects in cluster W

-
- 예제

method		cluster distance
single	=	d_{24}
complete	=	d_{13}
average	=	$\frac{d_{11} + d_{14} + d_{15} + d_{21} + d_{24} + d_{25}}{6}$

-

- **[중요] Single Linkage를 사용한 군집화**

1) Let

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix} \end{array}$$

$d_{53} = 2$ is the minimum.

Object 5 and 3 are merged: (35)

2) Let

$$D = \begin{array}{c} \\ (35) \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} (35) & 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{bmatrix} \end{array}$$

$$d_{(35)1} = \min(d_{31}, d_{51}) = \min(3,11) = 3$$
$$d_{(35)2} = \min(d_{32}, d_{52}) = \min(7,10) = 7$$
$$d_{(35)4} = \min(d_{34}, d_{54}) = \min(9,8) = 8$$

$\because$ $d_{(35)1} = 3$ is the minimum. We merge 1 and (35) to get (135).

3)

$$D = \begin{array}{c} \\ (135) \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} (135) & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 6 & 5 & 0 \end{bmatrix} \end{array}$$

$$d_{(135)2} = \min\left(d_{(35)2}, d_{12}\right) = \min(7,9) = 7$$
$$d_{(135)4} = \min\left(d_{(35)4}, d_{14}\right) = \min(8,6) = 6$$

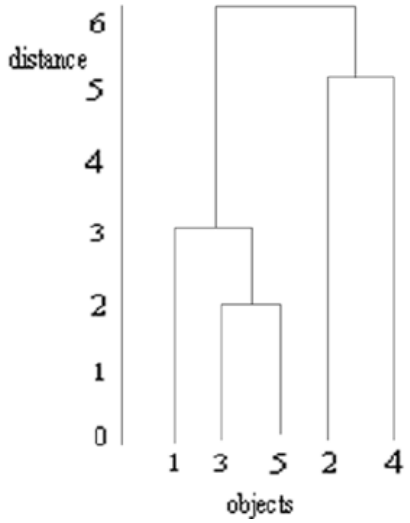
$\because$ $d_{42} = 5$ is the minimum. We merge 2 and 4 to get (24).

4)

$$D = \begin{array}{c} \\ (135) \\ (24) \end{array} \begin{array}{c} \begin{array}{cc} (135) & (24) \end{array} \\ \begin{bmatrix} 0 & \\ 6 & 0 \end{bmatrix} \end{array}$$

$$\because \quad d_{(24)(135)} = \min\left(d_{2(135)}, d_{4(135)}\right) = \min(7,6) = 6$$

- dendrogram으로 시각화

- 

14.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.

15.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>

고객번호	품목
1	오렌지 쥬스,사이다
2	우유, 오렌지 쥬스, 식기세척제
3	오렌지 쥬스, 세제
4	오렌지 쥬스, 세제, 사이다
5	식기 세척제, 사이다

<동시 구매표>

	오렌지 쥬스	식기 세척제	우유	사이다	세제
오렌지 쥬스	**4**	**1**	**1**	**2**	**2**
식기 세척제	**1**	**2**	**1**	**1**	**0**
우유	**1**	**1**	**1**	**0**	**0**
사이다	**2**	**1**	**0**	**3**	**1**
세제	**2**	**0**	**0**	**1**	**2**

- 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

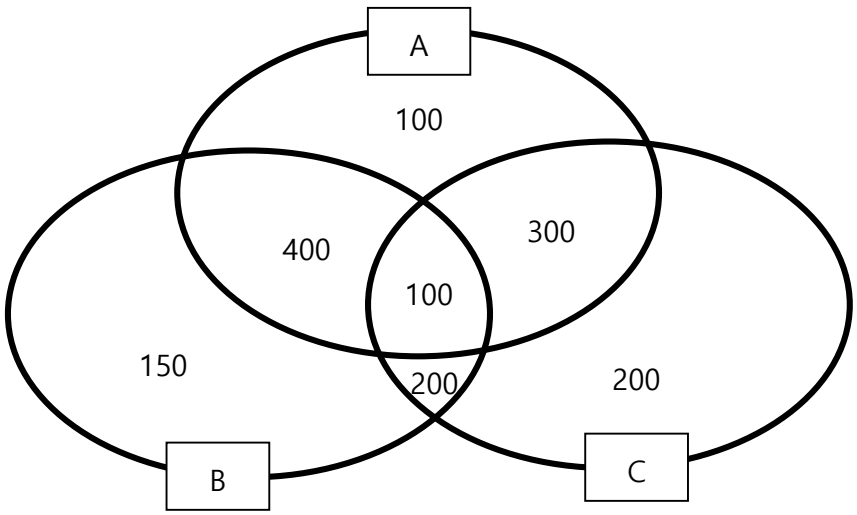
- 동시 거래 내역이 다음과 같을 때,

항목	거래의 수	항목	거래의 수
A	**100**	**A+C**	**300**
B	**150**	**B+C**	**200**
C	**200**	**A+B+C**	**100**
A+B	**400**	추가 안함	**550**

전체거래 회수
=2000

-

-



- 지지도 계산

항목	품목이 포함된 총 거래의 수	확률	항목	품목이 포함된 총 거래의 수	확률
A	**900**	**0.450**	**A+C**	**400**	**0.200**
B	**850**	**0.425**	**B+C**	**300**	**0.150**
C	**800**	**0.400**	**A+B+C**	**100**	**0.05**
A+B	**500**	**0.250**			

-

지지도(A+B) = (400 + 100) / 2000 = 0.25

- 신뢰도 계산

➢ 모든 연관성 규칙에 대한 신뢰도

규칙	P(A*B)	P(A)	신뢰도
A B	**25**	**45**	**0.556**
B A	**25**	**42.5**	**0.588**
C B	**15**	**40**	**0.375**
B C	**15**	**42.5**	**0.353**
A C	**25**	**45**	**0.556**

규칙	P(A*B)	P(A)	신뢰도
C A	**20**	**40**	**0.500**
(A+B) C	**5**	**25**	**0.200**
(B+C) A	**5**	**15**	**0.333**
(A+C) B	**5**	**20**	**0.250**

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.
-

향상도	의미
1	두 품목이 독립적인 관계
< 1	두 품목이 서로 음의 상관 관계
> 1	두 품목이 서로 양의 상관 관계