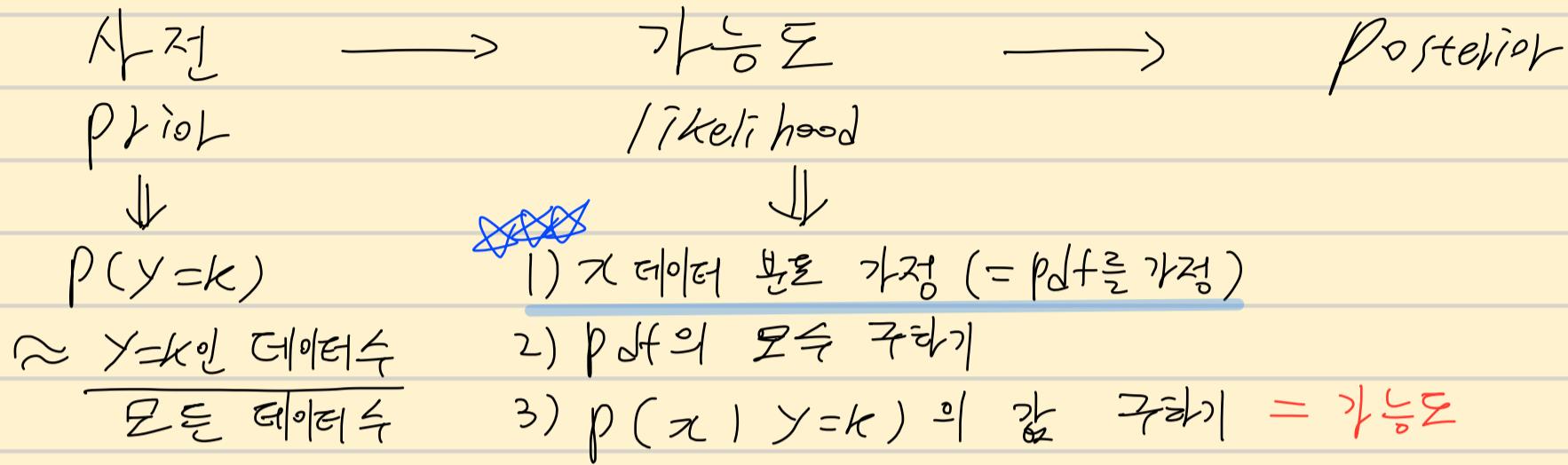


- 생성모형
 - LDA / QDA
 - Naive - Bayes



1) 가이더 분포 가정 모델 주요 가정
 = 가능도의 분포에 의미가 큼
 (분포의 모수는 클래스마다 모두 다르다)

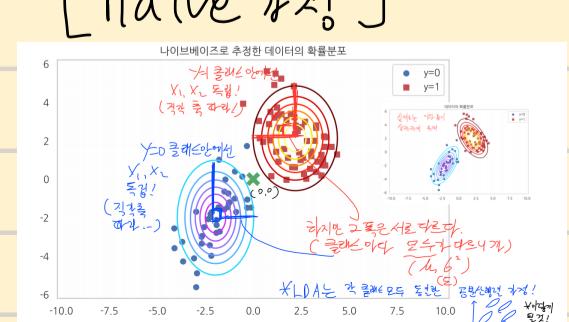
- x = 실수, 단변수 정규분포 라 “가정” QDA
- x = 모든 클래스가 공통된 공분산행렬
- x = 실수, 클래스마다 특별한 값 주변에서 발생 N.B. (가능도=정규분포) · 나이브 가정

- $x = (x_1, \dots, x_d)$, x_d 가 0 과 1 만 가능 N.B (가능도=베르누이) · 나이브 가정
 $* x_d = 1$ 번째 등전 단위에서 나온 결과 이진클래스
 $(= 각 클래스 내 feature들이 특징) = 조건부특징!$
- $x = (x_1, \dots, x_d)$, x_d 가 $1 \sim K$ 가 가능 N.B. (가능도=다항분포) · 나이브 가정
 $* x_d = 1$ 번째 주사위 단위에서 ‘ d ’가 나온 횟수

QDA : 공분산행렬 서로다름

LDA: 공분산행렬 공통

Naive 가정 : 클래스 내 독립



- N.B (베르누이, 다항분포) 이해.

$$1) x = \text{어}|\text{으}|\text{느}|\text{으}|\text{로}$$

$$P(Y=k | X_1, \dots, X_m) = \frac{1}{\binom{n}{k}}$$

$$p(x_1 \dots x_d | y=k) \propto 1$$

$$(\mu_{1,K} \dots \mu_{d,K})$$

$$P(X_1, \dots, X_d | Y=k) = \prod_{d=1}^D \mu_{d,k}^{X_d} (1 - \mu_{d,k})^{1-X_d}$$

$$= P(X=1 \mid Y=k)$$

$\mu_1 \ \mu_2 \ \mu_3 \ \mu_4$ = k 세트가 지목되었을 때, 그게 $x_i \sim \text{기ing}$ 동전세트

$y=0$ ○ ○ ○ ○ 일 가능성도.

$$y = 1 \quad \textcircled{O} \quad / \quad \textcircled{Q} \quad \textcircled{Q} \quad \textcircled{O}$$

$$\begin{array}{cc} \mu_{2,0} & \mu_{3,1} \\ (\mathbf{d}, \mathbf{k}) & (\mathbf{d}, \mathbf{k}) \end{array}$$

$$= P(x_2=1 \mid y=0) = p(x_2=1 \mid y=0)$$

2) $xc =$ 대항분포

$$\sum_{d=1}^D \chi_{d,k} = N$$

$\Rightarrow D$ 면적 주사위

$\Rightarrow k$ 개 주사위

$\Rightarrow N$ 번 면적 나온 결과

$\Rightarrow k$ 개 러지다

$\Rightarrow \sum_{d=1}^D M_{d,k} = 1$

$P(Y=k | \chi_1, \dots, \chi_D) = \text{그리 } N\text{ 번째 주사위일 확률}$

$$p(x_1, \dots, x_d | y = k) \propto 1$$

$$P(x_1, \dots, x_d | y=k) = \prod_{d=1}^D \mu_{d,k}$$

$x_1 \sim x_N$ 로 D면적 주사위 N 번던진 결과일 때,
 그게 K 번째 주사위가 지정되었을 때, 그 때
 $= K$ 번째 주사위

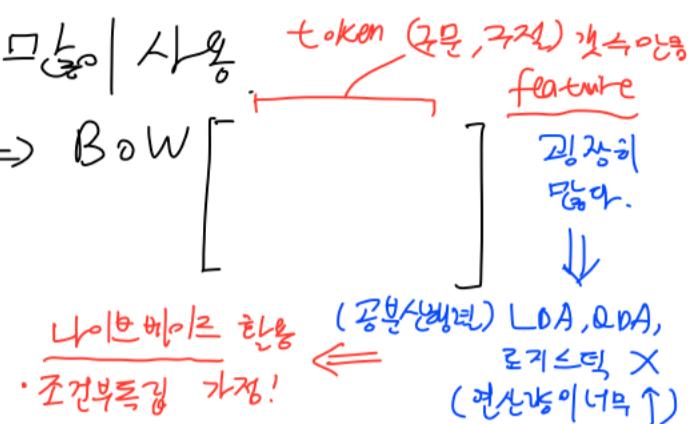
조건부 독립 이해.

7.2 나이브베이즈 분류모형

조건부독립

확률변수 A, B가 독립이면 A, B의 결합확률은 주변확률의 곱과 같다.

$$P(A, B) = P(A)P(B)$$



조건부독립(conditional independence)은 일반적인 독립과 달리 조건이 되는 별개의 확률변수 C가 존재해야 한다. 조건이 되는 확률변수 C에 대한 A, B의 결합조건부확률이 C에 대한 A, B의 조건부확률의 곱과 같으면 A와 B가 C에 대해 조건부독립이라고 한다.

$$P(A, B|C) = P(A|C)P(B|C)$$

C가 주어졌을 때 (데이터) 만
독립!!

$$\underbrace{A \perp\!\!\!\perp B}_{\text{---}} \mid C$$

기호로는 다음과 같이 표기한다.

$$\underbrace{A \perp\!\!\!\perp B}_{\text{---}} \mid \emptyset \quad (\text{무조건부 독립})$$

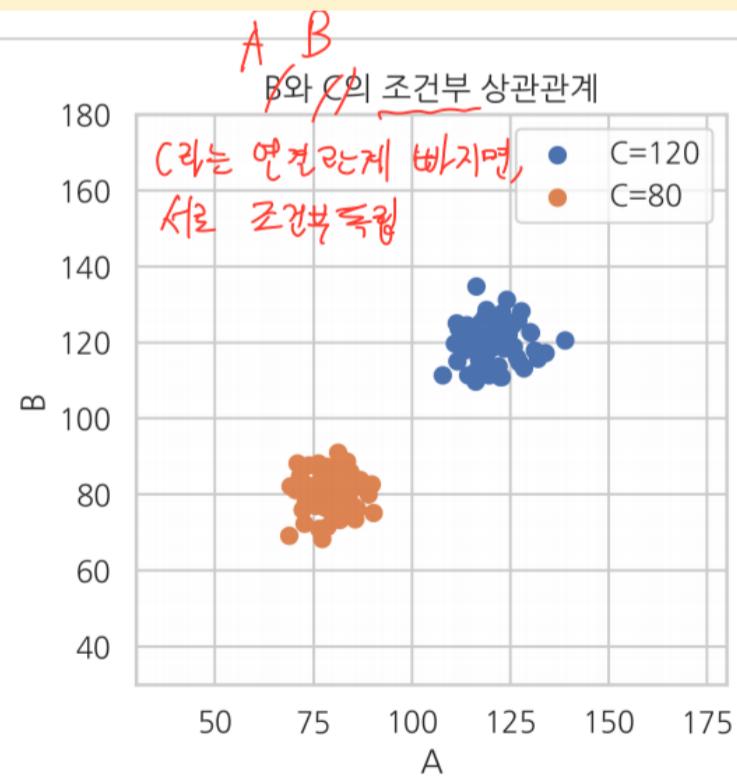
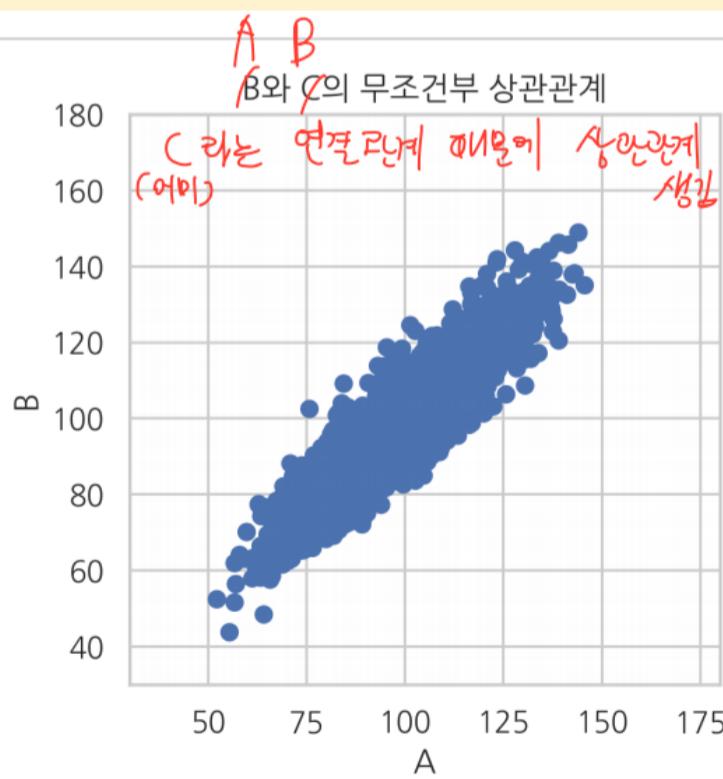
A, B가 C에 대해 조건부독립이면 다음도 만족한다.

$$\cancel{P(A|B, C) = P(A|C)} \Rightarrow B는 A의 독립이며, B에 대해서도 A는 독립
P(B|A, C) = P(B|C) \Rightarrow A는 B의 독립$$

주의할 점은 조건부독립과 무조건부독립은 관계가 없다는 점이다. 즉, 두 확률변수가 독립이라고 항상 조건부독립이 되는 것도 아니고 조건부독립이라고 꼭 독립이 되는 것도 아니다.

$$P(A, B) = P(A)P(B) \cancel{\Rightarrow} P(A, B|C) = P(A|C)P(B|C)$$

$$P(A, B|C) = P(A|C)P(B|C) \cancel{\Rightarrow} P(A, B) = P(A)P(B)$$



- LDA 가정의 이해

“모든 클래스가 공통된 공유된 행렬 사용”

↓
Pdf가 거의 선형 결합학!

(클래스에 따라 pdf가 달라지만, m_k 만 변함)

⇒ 학별 경계선이
‘직선’인 원인!

경계선 = 직선

선형판별분석법에서는 각 Y 클래스에 대한 독립변수 X 의 조건부확률분포가 공통된 공분산 행렬을 가지는 다변수 정규분포 (multivariate Gaussian normal distribution)이라고 가정한다. 즉

$\Sigma_k = \Sigma$ for all k 기능도 \Rightarrow 각 계산은 순차 결합으로 정리됨 (클래스의 디자인)

이다. $P(Y=k|X)$ 의 정리 (공분산행렬 공통사용/ 경계선 직선 기억) 이해하는 줄어드는 것과 같은 것

확률분포를 다음과 같이 정리할 수 있다.

$$\log p(x | y = k) = \log \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} - \frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k)$$

$$= C_0 - \frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k)$$

K가 빠진다는
상수

$$= C_0 - \frac{1}{2} (x^T \Sigma^{-1} x - 2\mu_k^T \Sigma^{-1} x + \mu_k^T \Sigma^{-1} \mu_k)$$

$$= C(x) + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k$$

K가 있는
상수

여기서 가능도가 선형으로 ..

/ogn화면전

다시 지속화면화해주세요

$$p(x | y = k) = \underline{C'(x)} \exp(\underline{w_k^T x} + w_{k0})$$

이 식에서 $C'(x) = \exp C(x)$ 이다. 가능도 상수 선형 결합 (K 에 따라 가중치변화)

$$\begin{aligned}
 P(y = k \mid x) &= \frac{p(x \mid y = k)P(y = k)}{\sum_l p(x \mid y = l)P(y = l)} \\
 &= \frac{C'(x) \exp(w_k^T x + w_{k0})P(y = k)}{\sum_l C'(x) \exp(w_l^T x + w_{l0})P(y = l)} \\
 &= \frac{C'(x) \exp(w_k^T x + w_{k0})P(y = k)}{C'(x) \sum_l \exp(w_l^T x + w_{l0})P(y = l)} \\
 &= \frac{P(y = k) \exp(w_k^T x + w_{k0})}{\sum_l P(y = l) \exp(w_l^T x + w_{k0})} \\
 &= \frac{P(y = k) \exp(w_k^T x + w_{k0})}{P(x)}
 \end{aligned}$$

이 식에서 $P(x)$ 는 y 클래스값에 영향을 받지 않는다.

따라서

$$\log P(y = k \mid x) = \log P(y = k) + w_k^T x + w_{k0} - \log P(x) = w_k^T x + C''_k$$

모든 클래스 k 에 대해 위와 같은 식이 성립하므로 클래스 k_1 과 클래스 k_2 의 경계선, 즉 두 클래스에 대한 확률값이 같아지는 x 위치를 찾으면 다음과 같다.

$$\begin{aligned} w_{k_1}^T x + C''_{k_1} &= w_{k_2}^T x + C''_{k_2} \\ (w_{k_1} - w_{k_2})^T x + (C''_{k_1} - C''_{k_2}) &= 0 \\ w_{k_1}^T x + C &= 0 \end{aligned}$$

(결론)

즉, 판별함수가 x 에 대한 선형방정식이 되고 경계선의 모양이 직선이 된다. \Rightarrow 그래서 LDA

• 스무딩 (Add-one, 라플라스 스무딩)

$$1. \text{ 분래 } \hat{\mu}_k = \frac{N_{d,k}}{N_k}$$

($k=0 \text{ or } 1, d=1 \sim D$)

$\rightarrow N_{d,k}$
= k 클래스 내
 \downarrow 동전 아님 주사위면
'성공(=1)' 횟수

2. 데이터 수 $\downarrow \Rightarrow N_{d,k} = 0$ 가능, then, $\hat{\mu}_k = 0$ 되는 극단적 상황 발생

3. $\therefore N_{d,k} \neq 0$ 하기 위해 $N_{d,k}$ 에 +1
씩 해준다.

4. Non-informative 하기 위해 분모에도 " $N_{d,k}$ 경우의 수"를 더해준다.

[베르누이 2
다항분포 6 (주사위면)]

$$\mu = \frac{1}{2}$$

$$\hat{\mu}_k = \frac{N_{d,k} + 1}{N_k + 2}$$

스무딩

표본 데이터의 수가 적은 경우에는 베르누이 모수가 0 또는 1이라는 극단적인 모수 추정값이 나올 수도 있다. 하지만 현실적으로는 실제 모수값이 이런 극단적인 값이 나올 가능성성이 적다. 따라서 베르누이 모수가 0.5인 가장 일반적인 경우를 가정하여 0이나 1이 나오는 경우와 같은 경우, 두 개의 가상 표본 데이터를 추가한다. 그러면 0이나 1과 같은 극단적인 추정값이 0.5에 가까운 다음과 같은 값으로 변한다. 이를 라플라스 스무딩(Laplace smoothing) 또는 애드원(Add-One) 스무딩이라고 한다.

$$\hat{\mu}_{d,k} = \frac{N_{d,k} + \alpha}{N_k + 2\alpha}$$

가중치 α 를 사용하여 스무딩의 정도를 조절할 수도 있다. 가중치 α 는 정수가 아니라도 괜찮다. 가중치가 1인 경우는 무정보 사전확률을 사용한 베이즈 모수추정의 결과와 같다.

[베르누이 베이즈 모형] \Rightarrow 손으로 직접!
* 주어진 데이터 $x=(0, 1, 1, 1) \Rightarrow$ 스팸? 정상? 분류!

In [15]:

```
x = np.array([
    [0, 1, 1, 0],
    [1, 1, 1, 1],
    [1, 1, 1, 0],
    [0, 1, 0, 0],
    [0, 0, 0, 1],
    [0, 1, 1, 0],
    [0, 1, 1, 1],
    [1, 0, 1, 0],
    [1, 0, 1, 1],
    [0, 1, 1, 0]])
y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1])
```

↳ 예상 결과 (T:스팸, 0:정상)

이 데이터는 4개의 키워드를 사용하여 정상 메일 4개와 스팸 메일 6개를 BOW 인코딩한 행렬이다. 예를 들어 첫번째 메일은 정상 메일이고 1번, 4번 키워드는 포함하지 않지만 2번, 3번 키워드를 포함한다고 볼 수 있다.

< 베르누이 가능도 모형의 모수추정식 >

$$\hat{\mu}_k = \left(\frac{N_{d,k} + 1\alpha}{N_k + 2\alpha} \right)$$

< 다항분포 //

$$\hat{\mu}_k = \frac{N_{d,k} + 1\alpha}{N_k + 2\alpha}$$

다항분포 나이브베이즈 모형

다항분포 나이브베이즈 모형 클래스 MultinomialNB 는 가능도 추정과 관련하여 다음 속성을 가진다.

- feature_count_ : 각 클래스 k 에서 d 번째 면이 나온 횟수 $N_{d,k}$
- feature_log_prob_ : 다항분포의 모수의 로그

$$\log \mu_k = (\log \mu_{1,k}, \dots, \log \mu_{D,k}) = \left(\log \frac{N_{1,k}}{N_k}, \dots, \log \frac{N_{D,k}}{N_k} \right)$$

여기에서 N_k 은 클래스 k 에 대해 주사위를 던진 총 횟수를 뜻한다.

스무딩 (가상 데이터 추가)

스무딩 공식은 60번자 주사위 5번던지기

$$(1, 1, 0, 2, 1, 0) \quad \hat{\mu}_{d,k} = \frac{N_{d,k} + \alpha}{N_k + D\alpha}$$

이다. $\Rightarrow \hat{\mu}_k = \left(\frac{1}{6}, \frac{1}{6}, \frac{0}{6}, \frac{2}{6}, \frac{1}{6}, \frac{0}{6} \right)$

(베르누이 $\hat{\mu}_k = \frac{N_{d,k}}{N_k}$) * $\hat{\mu}_k = \frac{N_{d,k}}{N_k}$ 너무 극단적 같으므로 스무딩 필요! (베르누이는

$$\frac{1+1}{5+6}, \frac{1+1}{5+6}, \frac{0+1}{5+6}, \frac{2+1}{5+6}, \frac{1+1}{5+6}, \frac{0+1}{5+6}$$

$$+(1, 1, 1, 1, 1, 1)$$

$$(1, 1, 0, 2, 1, 0)$$

$$\frac{2+1}{2+2+1+3+2+1}, \frac{1+1}{2+2+1+3+2+1}$$

$$\frac{+1\alpha}{+2\alpha}, \text{ 다항분포는 } \frac{+1\alpha}{+D\alpha}$$

Prior \rightarrow 가능도 \rightarrow Posterior

- 1) 분트 가정
 - 2) 모수 추정
 - 3) PDF, L¹ 양성
 - 4) 각 클래스별 가능도 계산

$$o=11\overline{2}$$

< 허브누이 >

[22]

$$\cdot p(y=k | x_1, \dots, x_d) \propto p(x_1, \dots, x_d | y=k) p(y=k)$$

데이터 : 4개의 키워드

1 or 0 ~~**~~ 10개 메일 BOW 인코딩 데이터
= 키워드 4개 정상 6개 스팸
출현 여부 2 = $N_{r,m}$

$X = \text{np.array}([$
 $[0, 1, 1, 0],$
 $[1, 1, 1, 1],$
 $[1, 1, 1, 0],$
 $[0, 1, 0, 0],$
 $[0, 0, 0, 1],$
 $[0, 1, 1, 0],$
 $[0, 1, 1, 1],$
 $[1, 0, 1, 0],$
 $[1, 0, 1, 1],$
 $[0, 1, 1, 0]]))$
 $y = \text{np.array}([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])$

W
X

$p(y=1 | x_1=0, x_2=1, x_3=1, x_4=1)$
 vs
 $p(y=0) \quad //$
 $\therefore \text{feature_log_Prob_}$

[학습데이터]
 0개수 4번 $y=0$ $\frac{1}{2+1}$ $\frac{1}{4+2}$
 1개수 6번 $y=1$ $\frac{2+1}{4+2}$ $\frac{3+1}{6+2}$
 (\triangle BB)

$p(y=1 | x= \dots) \propto p(x_i= \dots)$
 vs
 $p(y=0 | x= \dots) \propto$

$$e) \text{ prior} = \frac{N_1}{N} = \frac{6}{10}$$

$$\approx p(y=1)$$

```
In [17]:  
model_bern.classes_  
  
Out[17]:  
array([0, 1])  
  
In [18]:  
model_bern.class_count_  
  
Out[18]:  
array([4., 6.])  
  
In [19]:  
np.exp(model_bern.class_log_prior_)  
  
Out[19]:  
array([0.4, 0.6])
```

1) 분포가정

1) χ 아래의 표를 가정 모형 주제

$$= \text{주제 } \chi_1, \chi_2, \dots, \chi_n \text{에 따른 } Y_{ij} \text{의 확률}$$

$$(Y_{ij} \text{는 } \chi_1, \chi_2, \dots, \chi_n \text{에 따른 } Y_{ij} \text{의 확률})$$

- $\chi =$ 성별, 대학원 학과별 “가장” QDA
- $\chi =$ LDA
- $\chi =$ 성별, 출생연도(4년 단위) NB ((χ_1, χ_2))
- $\chi = (\chi_1, \dots, \chi_n), \chi_1 \neq 0, 1, \dots, k$ NB ((χ_1, \dots, χ_n))
 - $\chi_1 =$ 성별, 출생연도(4년 단위)
 - $\chi_2 =$ 출생연도(4년 단위)
 - $\chi_3 =$ 출생연도(4년 단위)
 - \vdots
 - $\chi_k =$ 출생연도(4년 단위)
- $\chi = (\chi_1, \dots, \chi_n), \chi_1 \neq 0, 1, \dots, k$ NB ((χ_1, \dots, χ_n))
 - $\chi_1 =$ 성별, 출생연도(4년 단위)
 - $\chi_2 =$ 출생연도(4년 단위)
 - $\chi_3 =$ 출생연도(4년 단위)
 - \vdots
 - $\chi_k =$ 출생연도(4년 단위)

```
from sklearn.naive_bayes import BernoulliNB  
model_bern = BernoulliNB().fit(X, y)
```

2) 모수적정 (해답)

$$\hat{\mu}_{d,\kappa} = \frac{N_{d,\kappa} + \alpha}{N_k + 2\alpha}$$

클래스 κ 에 대한 확률

① 각 클래스 별 Count

	$N_{d,k}$				N_k	동전던전 횟수
$y=0$	2	4	3	1	4	\langle 동전 던자기 \rangle
$y=1$	2	3	5	3	6	$\frac{1000000}{\square} =$ 동전 1개 \hookrightarrow (번 던져서 나온 결과)

```
fc = model_bern.feature_count_
fc
```

Out[20]:

```
array([[2., 4., 3., 1.],
       [2., 3., 5., 3.]])
```

③ $\hat{\mu}_{s,k}$ 은 s 에
(+ 브루닝)

	$M_{1,k}$	$M_{2,k}$	$M_{3,k}$	$M_{4,k}$
$y=0$	$\frac{2+1}{4+2}$	$\frac{4+1}{4+2}$	$\frac{3+1}{4+2}$	$\frac{1+1}{4+2}$
$x=1$	$\frac{2+1}{6+2}$	$\frac{3+1}{6+2}$	$\frac{5+1}{6+2}$	$\frac{3+1}{6+2}$

```
theta = np.exp(model_bern.feature_log_prob_)
theta
```

Out[23]: $\mu_1, \mu_2, \mu_3, \mu_4$

```
array([[ 0.5      ,  0.83333333,  0.66666667,  0.33333333],
       [ 0.375    ,  0.5      ,  0.75     ,  0.5      ]])
```

3) PDF, L 썸네일

$$PDF = L = p(x_1 \dots x_d | Y=k)$$

$$= \prod_{d=1}^D M_{d,k}^{x_{d,k}} (1 - M_{d,k})^{1-x_{d,k}}$$

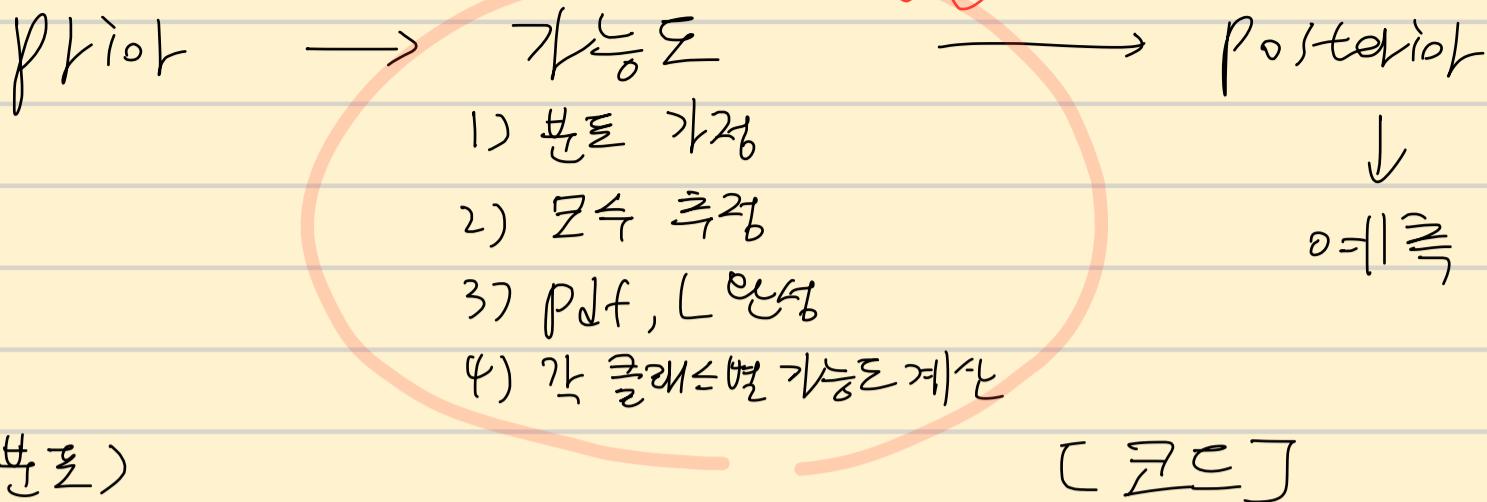
4) 각 클래스별 가능도 계산
 \Rightarrow posterior = 예 / 총

클래스별 posterior 히고!

```
x_new = np.array([0, 1, 1, 1])
In [25]: model_bern.predict_proba([x_new])
Out[25]:
array([[0.34501348, 0.65498652]])
```

$P(Y=0|x) \propto P(x_1=0|Y=0) \times P(x_2=1|Y=0) \times P(x_3=1|Y=0) \times P(x_4=1|Y=0) \times P(Y=0)$

$$P(x_1, \dots, x_d | Y=1) \Rightarrow P(Y=1 | x_1, \dots, x_d) \quad \text{ヒ고!}$$
$$P(x_1, \dots, x_d | Y=0) \Rightarrow P(Y=0 | x_1, \dots, x_d) \quad \text{ヒ고!}$$



$$P(Y=k | x_1, \dots, x_d) \propto P(x_1, \dots, x_d | Y=k) P(Y=k)$$

데이터 : 4개의 키워드

$1 \sim 10$ 개의 메일 BOW 인코딩 데이터

= 키워드

“출현빈도”

4개 정상 6개 스팸

$$N_k = 4 \times 10$$

= k 클래스에 대처
4번 주사위 던진 횟수!

In [30]:
 $X = np.array([1, 4, 1, 2], [3, 5, 1, 1], [3, 3, 0, 4], [3, 4, 1, 2], [1, 2, 1, 4], [0, 0, 5, 3], [1, 2, 4, 1], [1, 1, 4, 2], [0, 1, 2, 5], [2, 1, 2, 3]])$
 $y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])$

$R=0$ $Y=0$ $\frac{12+1}{40+4}$
 $R=1$ $Y=1$ $\frac{5+1}{48+4}$
 $K=0$ $K=1$

10번 주사위 던지는 세트
4세트에서 주사위 104번 던짐
 $\mu_1 = \frac{12+1}{40+4}$
 $\mu_2 = \frac{5+1}{48+4}$

$$\text{1) prior} = \frac{N_1}{N} = \frac{6}{10}$$

$\approx P(Y=1)$

$$N_{1,0} = 12$$

$$N_{2,0} = 16$$

```
In [32]: model_mult.classes_
Out[32]: array([0, 1])

In [33]: model_mult.class_count_
Out[33]: array([4., 6.])

In [34]: np.exp(model_mult.class_log_prior_)
Out[34]: array([0.4, 0.6])
```

1) 분포 가정

1) x에 대한 확률 가정
 $P(X_i = x_i | Y=k)$
 $x_i = 0, 1, 2, \dots, 10$
 $x_i = (x_1, \dots, x_d), x_i \in \Omega_{d,k}$
 $x_i = (x_1, \dots, x_d), x_i \in \Omega_{d,k}$

```
from sklearn.naive_bayes import MultinomialNB
model_mult = MultinomialNB().fit(X, y)
```

2) 모수 추정

(해당)
① 각 클래스별 Count

	$N_{d,k}$				N_k
$Y=0$	12	16	3	9	40
$Y=1$	5	7	18	18	60

$$\hat{\mu}_{d,k} = \frac{N_{d,k} + \alpha}{N_k + 2\alpha}$$

N_k 은 클래스 k 에 대해 주사위를 던진 총 횟수를 뜻한다.

```
In [35]:
fc = model_mult.feature_count_
fc
Out[35]:
array([[12., 16., 3., 9.],
       [5., 7., 18., 18.]])
```

② $\hat{\mu}_{d,k}$ 계산
(+ 스무딩)

$$\begin{array}{l} Y=0 \quad \mu_1 = \frac{12+1}{40+4} = 0.29545455 \\ Y=1 \quad \mu_2 = \frac{5+1}{48+4} = 0.11538462 \end{array}$$

$$\begin{array}{l} Y=0 \quad \mu_1 = \frac{12+1}{40+4} = 0.29545455 \\ Y=1 \quad \mu_2 = \frac{5+1}{48+4} = 0.11538462 \end{array}$$

$$\hat{\mu}_{2,1} = P(Y=1 | x_2=1)$$

```
In [39]:
theta = np.exp(model_mult.feature_log_prob_)
theta
Out[39]:
array([0.29545455, 0.38636364, 0.09090909, 0.22727273,
       0.11538462, 0.15384615, 0.36538462, 0.36538462])
```

3) PDF, L 쿠션

$$PDF = L = p(x_1 \dots x_d | y=k)$$

$$= \prod_{d=1}^D M_{d,k}^{x_{d,k}}$$

4) 각 클래스별 가능도 계산
 $\Rightarrow posterior \Rightarrow 예측$

클래스별 posterior 예고!

In [40]:

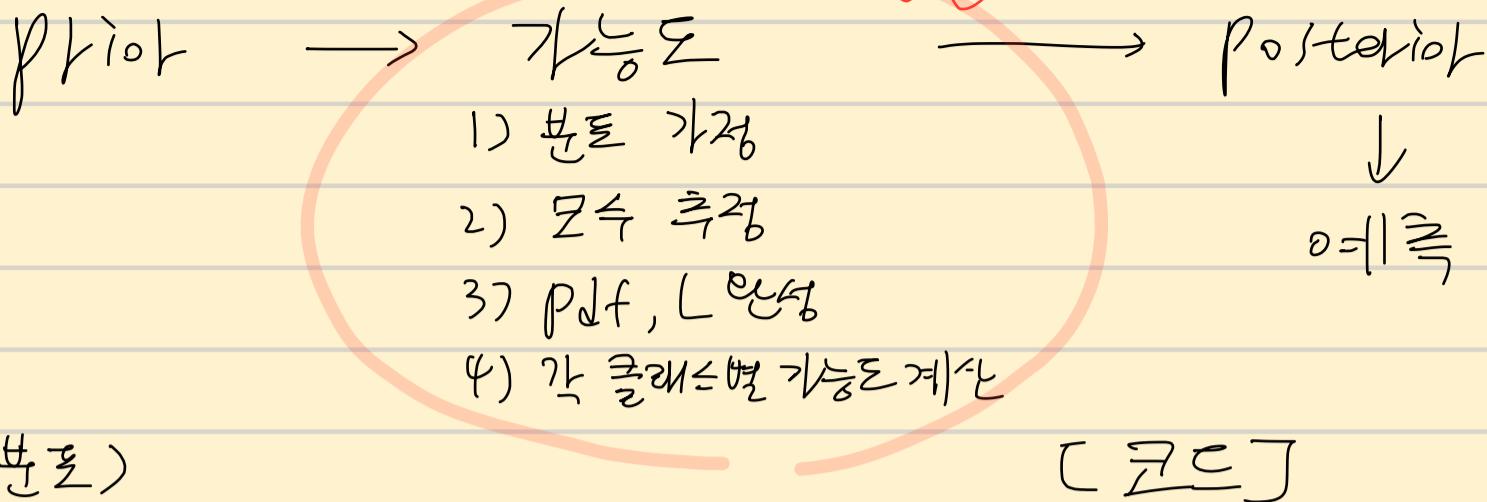
```
x_new = np.array([10, 10, 10, 10])
model_mult.predict_proba([x_new])
```

Out[40]:

```
array([[0.38848858, 0.61151142]])
```

$$P(x_1, \dots, x_d | y=1) \Rightarrow P(y=1 | x_1, \dots, x_d) \quad J 예고!$$

$$P(x_1, \dots, x_d | y=0) \Rightarrow P(y=0 | x_1, \dots, x_d)$$



$$P(Y=k | x_1, \dots, x_d) \propto P(x_1, \dots, x_d | Y=k) P(Y=k)$$

실수인 두 개의 독립변수 x_1, x_2 와 두 종류의 클래스 $y = 0, 1$ 을 가지는 분류문제가 있다.

두 독립변수의 분포는 정규분포이고 y 의 클래스에 따라 다음처럼 모수가 달라진다.

$$\mu_0 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \Sigma_0 = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 2 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1.2 & -0.8 \\ -0.8 & 2 \end{bmatrix}$$

데이터는 $y = 0$ 인 데이터가 40개, $y = 1$ 인 데이터가 60개 주어졌다. 이 데이터를 시각화하면 다음과 같다.

```
np.random.seed(0)
rv0 = sp.stats.multivariate_normal([-2, -2], [[1, 0.9], [0.9, 2]])
rv1 = sp.stats.multivariate_normal([2, 2], [[1.2, -0.8], [-0.8, 2]])
X0 = rv0.rvs(40)
X1 = rv1.rvs(60)
```

$$\text{1) prior} = \frac{N_1}{N} = \frac{6}{10}$$

$$\approx P(Y=1)$$

```
In [5]: model_norm.classes_
Out[5]: array([0., 1.])

In [6]: model_norm.class_count_
Out[6]: 0 클래스 데이터 40개
array([40., 60.]) 1 클래스 데이터 60개

In [7]: model_norm.class_prior_
Out[7]: "prior는 클래스 비율"
array([0.4, 0.6])
```

1) 분포 가정

1) x에 대한 분포 가정	모형	주요 가정
= 정규분포 가정 (나이브 가정은 정규분포 가정)	0.0A	
$x = \dots$	LDA	모든 클래스가 같은 확률로 발생
$x = \dots$, 평균, 편차, 공분산 행렬	NB (Multinomial)	나이브 가정
$x = (x_1, \dots, x_d), x_i \sim N(\mu_i, \Sigma_i)$	NB (Multinomial)	($\Sigma_i = \text{단위 행렬} \Rightarrow \text{나이브 가정}$)
$x = (x_1, \dots, x_d), x_i \sim N(\mu_i, \Sigma_i)$	NB (Multinomial)	($\Sigma_i = \text{단위 행렬} \Rightarrow \text{나이브 가정}$)
↑ 00 ↑ 00 ↑ 00	[Naive Bayes]	

```
In [4]:
from sklearn.naive_bayes import GaussianNB
model_norm = GaussianNB().fit(X, y)
```

2) 모수 추정 (MLE)

$$\text{train data} \Rightarrow \hat{\mu}, \Sigma = \hat{\sigma}^2$$

(표본) $= \bar{x}$ $= s^2$

$$* MLE = \frac{\partial LL}{\partial \mu} - \frac{\partial LL}{\partial \Sigma}$$

각 클래스에 따라 x 가 이루는 확률분포의 모수를 계산하면 다음과 같다. 나이브 가정에 따라 x_1, x_2 는 독립이므로 상관관계를 구하지 않았다.

In [8]: \rightarrow 평균而已 \rightarrow 분산而已 (원래는 공분산행렬)

model_norm.theta_[0], model_norm.sigma_[0]

Out[8]:

(array([-1.96197643, -2.00597903]), array([1.02398854, 2.31390497]))

In [9]:

model_norm.theta_[1], model_norm.sigma_[1]

Out[9]:

(array([2.19130701, 2.12626716]), array([1.25429371, 1.93742544]))

$\Sigma_0 = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 2 \end{bmatrix}$ 비대각성분은 0으로 가정 후 (나이브 가정)
 $\Sigma_1 = \begin{bmatrix} 1.2 & -0.8 \\ -0.8 & 2 \end{bmatrix}$ 일은 모두 대각성분 부분)

3) PDF, L 쿠션

$$PDF = L = p(x_1 \dots x_d | y=k)$$

$$= \prod_{d=1}^D M_{d,k}^{x_{d,k}}$$

4) 각 클래스별 가능도 계산
 \Rightarrow posterior = 예측

클래스별 posterior 이고!

In [11]:

```
model_norm.predict_proba([x_new])
```

Out[11]:

```
array([[0.48475244, 0.51524756]])
```

$$P(x_1, \dots, x_d | y=1) \Rightarrow P(y=1 | x_1, \dots, x_d) \quad \text{J 이고!}$$

$$P(x_1, \dots, x_d | y=0) \Rightarrow P(y=0 | x_1, \dots, x_d)$$

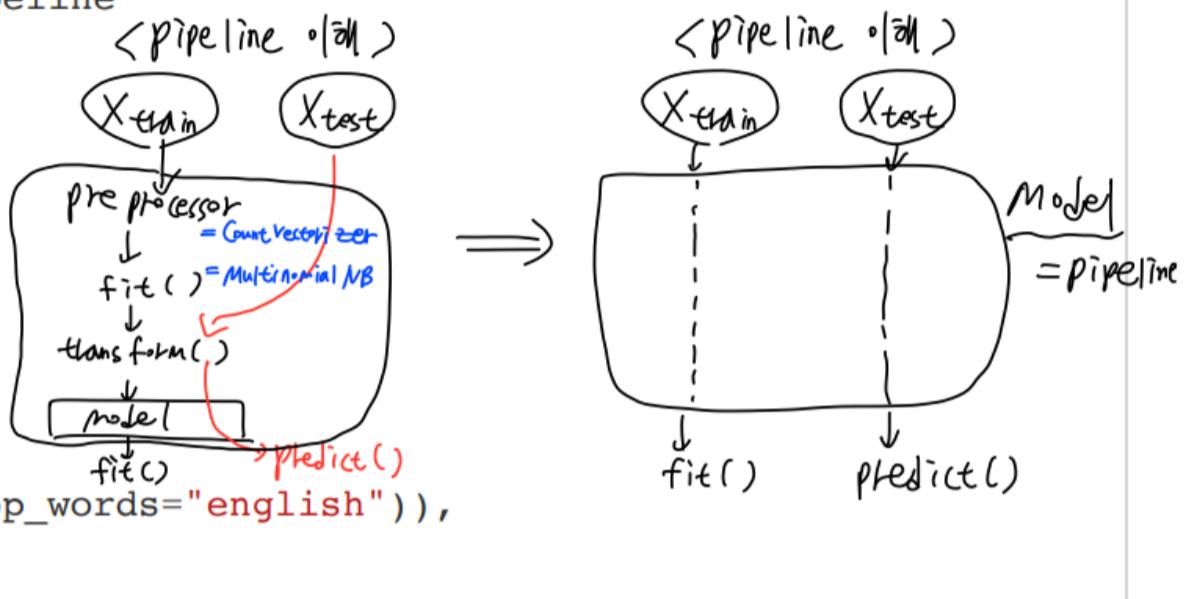
[Pipeline 퀄리티 고하기]

```

from sklearn.feature_extraction.text import TfidfVectorizer, HashingVectorizer,
CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline

model1 = Pipeline([
    ('vect', CountVectorizer()),
    ('model', MultinomialNB()),
])
model2 = Pipeline([
    ('vect', TfidfVectorizer()),
    ('model', MultinomialNB()),
])
model3 = Pipeline([
    ('vect', TfidfVectorizer(stop_words="english")),
    ('model', MultinomialNB()),
])
model4 = Pipeline([
    ('vect', TfidfVectorizer(stop_words="english",
                           token_pattern=r"\b[a-zA-Z_\\-\\.]+[a-zA-Z][a-zA-Z_\\-\\.]+\b")),
    ('model', MultinomialNB()),
])

```



In [43]:

```

%%time
from sklearn.model_selection import cross_val_score, KFold

for i, model in enumerate([model1, model2, model3, model4]):
    scores = cross_val_score(model, X, y, cv=5)
    print("Model{}: Mean score: {:.3f}".format(i + 1, np.mean(scores)))

```

```

Model1: Mean score: 0.855
Model2: Mean score: 0.856
Model3: Mean score: 0.883
Model4: Mean score: 0.888
CPU times: user 2min 15s, sys: 3.48 s, total: 2min 18s
Wall time: 1min 56s

```

작고 빠른 모델이 좋다
NB가 더 좋다

X

[헌법 분포 차이와의 posterior, 예측 분류기]

Wall time: 1min 56s

만약,

연습 문제 5

X		
$x_1 - x_{10}$	$x_{11} - x_{20}$	$x_{21} - x_{30}$
$\begin{array}{ c c c }\hline GNB & \begin{array}{ c c }\hline 1 & 0 \\ \hline 1 & 1 \\ \hline \end{array} & \begin{array}{ c c }\hline 1 & 1 \\ \hline \end{array} \\ \hline \end{array}$	$y=0$	$y=1$
$\begin{array}{ c c c }\hline & & \\ \hline \end{array}$	$y=0$	$y=1$

X 분포가 정규, 베르, 다항이 섞여 있다면?

(1) 만약 독립변수로 실수 변수, 0 또는 1 값을 가지는 변수, 자연수 값을 가지는 변수가 섞여있다면 사이킷런에서 제공하는 나이브베이즈 클래스를 사용하여 풀 수 있는가?

(2) 사이킷런에서 제공하는 분류문제 예제 중 숲의 수종을 예측하는 covtype 분류문제는 연속확률분포 특징과 베르누이확률 분포 특징이 섞여있다. 이 문제를 사이킷런에서 제공하는 나이브베이즈 클래스를 사용하여 풀어라.

"predict_proba"에서

$$p(y|x) = p(x_1|y) p(y)$$

$$= p(x_{1:10}|y) p(x_{11:20}|y) p(x_{21:30}|y) \times p(y)$$

$$\frac{p(y|x_{1:10})}{p(x)} = p(x_{1:10}|y)$$