

# 범주형자료분석팀

**2팀**

김찬영  
이혜인  
김서윤  
심은주  
진수정

# INDEX

---

0. 지난 주 리뷰

1. Confusion Matrix

2. ROC & AUC

3. Sampling

4. Encoding

0

지난 주 리뷰

# GLM

일반화 선형모형

범주형 반응변수에 대한  
비선형 관계

연속형 반응변수에 대한  
선형 관계

ex) 회귀모형, 분산분석 모형

→ 범주형 반응변수에 대한 모형까지 포함하는  
광범위한 모형의 집합

# 유의성 검정

- 모형의 모수 추정 값이 유의한지 검정
- 축소 모형의 적합도가 좋은지에 대한 검정

- 가설

$$g(\mu) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_k = 0$$

$H_1$ : 적어도 하나의  $\beta$ 는 0이 아니다

- 종류

wald 검정	스코어 검정	가능도비 검정
---------	--------	---------

# 로지스틱 회귀모형

: 반응변수  $Y$ 가 이항반응변수일 때 사용

예시

학점에 따른 합격 여부를 확인하는 문제

- $Y \sim \text{Bernoulli}(\pi)$  이항분포 사용

일반 선형 모델( $\pi(x) = \alpha + \beta x$ )로  
적합한다면?

# 포아송 회귀모형

: 반응변수 Y가 **도수자료**일 때 사용

예시

교통사건 건수를 예측하는 문제

- $Y \sim \text{Poisson}(\mu)$  포아송 분포 사용

**Poisson Regression** : log를 link function으로 사용

$$\log(\mu) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

# 로그 선형 모형

1. 범주형 자료들이 각 칸도수에 어떤 영향을 미치는 지 확인할 수 있음
2. 설명변수와 반응변수 구분이 없음
3. 포아송 분포를 사용함

내포 모형 VS 포화모형



분할표를 가장 잘 설명하는 모형을 찾는 것



1

# Confusion Matrix

# "Confusion Matrix" 혼동 행렬

: 예측 성능 측정을 위해, 학습을 통해 도출한 예측값과 실제 관측값을 비교한 표

- 분류 알고리즘의 성능을 시각화한 표

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 ( $Y$ )	$Y=1$	TP	FN
	$Y=0$	FP	TN

- T(True)와 F(False):  
실제와 예측이 같은지 다른지 여부
- P(Positive) N(Negative):  
예측을 긍정 혹은 부정이라 했는지 여부

# "Confusion Matrix" 혼동 행렬

: 예측 성능 측정을 위해, 학습을 통해 도출한 예측값과 실제 관측값을 비교한 표

- 분류 알고리즘의 성능을 시각화한 표

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 ( $Y$ )	$Y=1$	TP	FN
	$Y=0$	FP	TN

- T(True)와 F(False):  
실제와 예측이 같은지 다른지 여부
- P(Positive) N(Negative):  
예측을 긍정 혹은 부정이라 했는지 여부



# 혼동행렬(Confusion Matrix) 해석하기!

## "Confusion Matrix"

### Step.1

예측이 **긍정(P)**인가? **부정(N)**인가?

- 분류 알고리즘의 성능을 시각화한 표!

### Step.2

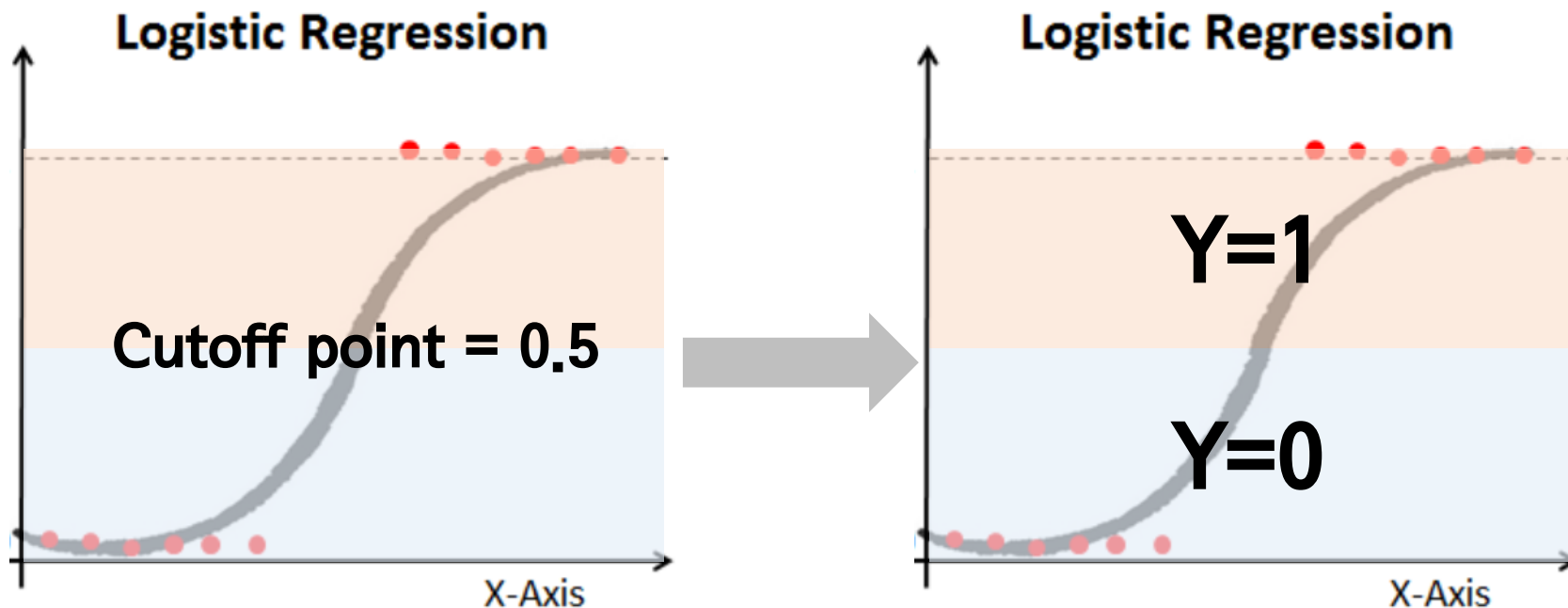
예측이 실제로 **맞았나?(T)** **틀렸나?(F)**

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 ( $Y$ )	$Y=1$	TP	FN
	$Y=0$	FP	TN

EX) 예측은 맞다고 했는데 (P: 긍정 예측), 실제로는 틀린 경우 (F: 거짓)

→ FP

# "Cut-off point" : 분류 임계값



로지스틱 회귀분석의 결과 = 확률값

→ Y=1(성공) or Y=0(실패) 나눌 cut-off point 필요

## 한계

1. 연속적인 예측값인 확률 $[\hat{\pi}]$ 을 이항변수  $[0,1]$ 로 묶어버림
2. cut-off point인  $\pi_0$ 값의 선택이 임의적
3.  $Y$ 값의 상대적 비율에 따라 민감하게 나타남

Confusion  
Matrix

Cut off point에 의존적

정보의 손실 발생

스포하자면... 이러한 이유로 다음에 <ROC & AUC>이 등장한다!

**"ACC"** : Accuracy, 정분류율

$$\text{ACC} = (\text{TP} + \text{TN}) / (\text{전체}) = 1 - \text{Error Rate}$$

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 ( $Y$ )	$Y=1$	TP	FN
	$Y=0$	FP	TN

- 실제와 예측이 맞은 경우의 비율
- 1에 가까울수록 좋은 모형
- Unbalanced Data 모형 평가 시 문제 발생

**"TPR"** : True Positive Rate

*Sensitivity 민감도 / Recall 재현도*

$$\text{TPR} = \text{TP} / (\text{TP} + \text{FN}) = 1 - \text{FNR}$$

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 ( $Y$ )	$Y=1$	TP	FN
	$Y=0$	FP	TN

- 민감도  $P(\hat{Y}=1|Y=1)$   
: 실제 성공을 얼마나 잘 예측했는가?에 대한 답
- 1에 가까울수록 좋다
- ROC곡선의 Y축 값



**"TNR"** : True Negative Rate

Specificity 특이도

$$\text{TNR} = \text{TN} / (\text{FP} + \text{TN}) = 1 - \text{Error Rate}$$

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 ( $Y$ )	$Y=1$	TP	FN
	$Y=0$	FP	TN

- 특이도  $P(\hat{Y}=0|Y=0)$   
: 실제 실패를 얼마나 잘 예측했는가?에 대한 답
- 1에 가까울수록 좋다
- FPR의 정확히 반대 개념



예시를 통해 알아보자!

"TNR"

"민감도"

: 진-짜 성공한 애들 중 모델이 성공했다고 예측한 애들이 얼마나 되는지!

내가 코로나에 걸렸을 때(실제),

검사 결과(예측)가 양성으로 나올 확률

$$TNR = TN / (FP + TN) = 1 - Error Rate$$

$$P(\hat{Y}=1|Y=1)$$

"특이도"

: 진-짜 실패한 애들 중 모델이 실패했다고 예측한 애들이 얼마나 되는지!

내가 코로나에 걸리지 않았을 때(실제),

검사 결과(예측)가 음성으로 나올 확률

$$P(\hat{Y}=0|Y=0)$$

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 (Y)	Y=1		
	Y=0	FP	TN

• 특이도  $P(\hat{Y}=0|Y=0)$

• 실제 실패한 애들 중 모델이 실패했다고 예측한 애들이 얼마나 되는지!

• 실제 실패한 애들 중 모델이 실패했다고 예측한 애들이 얼마나 되는지!

• FPR의 정확히 반대 개념

**"FPR"** : False Positive Rate

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN}) = 1 - \text{TNR}$$

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 (Y)	Y=1	TP	FN
	Y=0	FP	TN

- 실제 Y=0인 값 중 Y=1이라 예측된 값( $\hat{Y}=1$ )의 비율
- 0에 가까울수록 좋다
- TNR의 반대 개념
- ROC 곡선의 X축 값

**"PPV"** : Positive Predictive Value

Precision 정밀도

$$PPV = TP / (TP + FP) = 1 - FPR$$

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 (Y)	Y=1	TP	FN
	Y=0	FP	TN

- 예측된  $\hat{Y}=1$  중 실제  $Y=1$ 인 것의 비율  
: 예측한 성공 중 실제 성공은 얼마나 되는가?에 대한 답
- 1에 가까울수록 좋다
- F1-Score와 연관

# "F1-Score"

$$\begin{aligned} & 2TP / (2TP + FN + FP) \\ & = 2(\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \end{aligned}$$

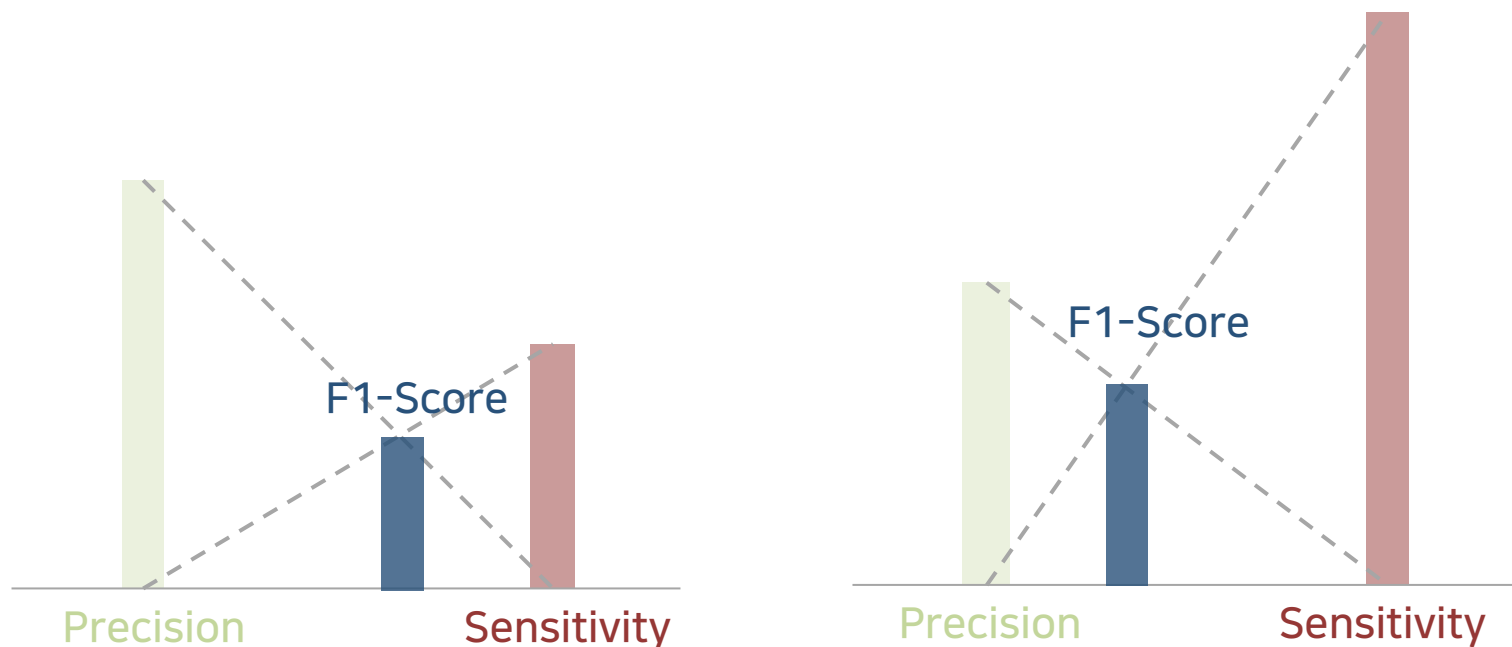
		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 (Y)	Y=1	TP	FN
	Y=0	FP	TN

Sensitivity

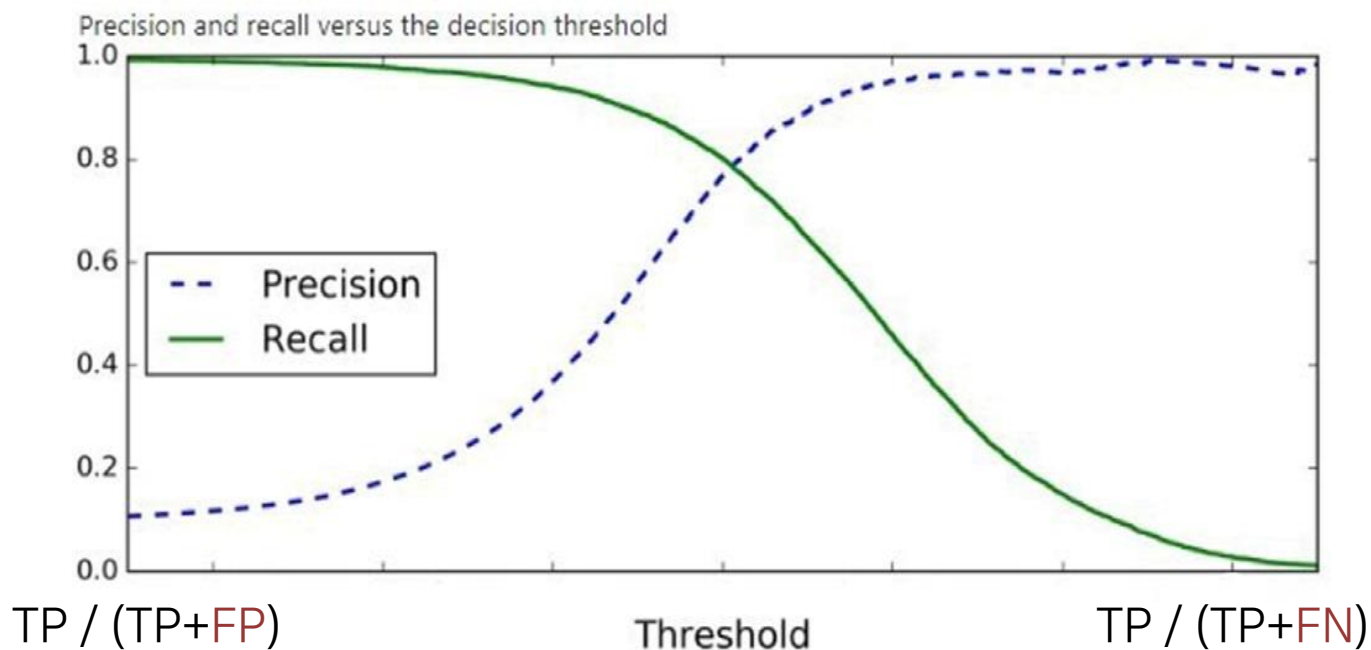
Precision

# "F1-Score"

- Precision과 Recall(=Sensitivity)의 **조화평균**  
: 두 지표의 **밸런스를 고려**해 정확도 측정



# "F1-Score"



일종의 Trade-off 관계임을 알 수 있다!

## "F1-Score"

### 장점

두 개의 False 상황(FN, FP)을 고려하는 지표

➡ Unbalanced data 평가 지표로도 좋다!

### 한계

FN과 FP는 사용해도, TN은 사용하지 않는다



# "MCC"

이름에서 알 수 있듯 상관계수 감성이다!  
: Matthews Correlation Coefficient

$$\frac{(TP \times FP) - (FN \times TN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 (Y)	Y=1	TP	FN
	Y=0	FP	TN

- 모든 부분을 사용 → unbalanced data에도 유용하다!

# "MCC" : Matthews Correlation Coefficient

## 특징

### 1. 범위: $-1 \leq MCC \leq 1$

- 1: 완벽하게 예측!
- 0: 랜덤 예측과 같다 (예측한 의미가 없다...)
- -1: 완벽하게 예측 실패!

### 2. Confusion Matrix를 설명하는 가장 좋은 지표





왜 MCC가 가장 좋은 지표일까?

: 다른 지표들의 한계부터 살펴보자!

“MCC”  
“ACC”

Matthews Correlation Coefficient

특징

1. 범위:  $-1 \leq MCC \leq 1$  Unbalanced data 일 경우 그리 유용하지 X

- 1: 완벽하게 예측!

“F1-score”

- -1: 완벽하게 예측 실패!

TN을 사용하지 X

2. Confusion Matrix를 만들 경우  $Y=1$ 과  $Y=0$  값이 바뀌면 성능 지표도 바뀐다

이제 왜 그런지 예시를 통해 본격적으로 알아보자!

# Case 1. *밸런스 왜장창.. 95: 5 Unbalanced data*

단순히 모두  $Y=1$ 이라고 예측하는 모델을 적합했다고 해보자

CASE1		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 (Y)	Y=1	95	0
	Y=0	5	0



(잘 적합 된 것 같은데...?)

- Accuracy:  $[95/100 = 95\%]$
- F1-score:  $[(2*95)/(2*95+5+0) = 97.44\%]$

# Case 1. 밸런스 왜장창.. 95: 5 Unbalanced data

CASE1		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 (Y)	Y=1	95	0
	Y=0	5	0

• MCC: 
$$\frac{(TP \times FP) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

→ 분모에 0을 포함하게 되어.. 랜덤 예측과 다를 바 없다!



## Case 2. 너무 극단적이지 않은, 조금은 현실적인 모델

CASE2		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 (Y)	Y=1	90	5
	Y=0	4	1



(다시 한번 의심해보자...)

- Accuracy:  $[(90+1)/100 = 91\%]$
- F1-score:  $[(2*90)/(2*95+4+5) = 95.24\%]$

## Case 2. 너무 극단적이지 않은, 조금은 현실적인 모델

CASE2		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 (Y)	Y=1	90	5
	Y=0	4	1

돌다리도 두드려보고 건너라는  
선조 분들의 말씀이 괜히 있는 게 아니다!

- $MCC = 0.14 \rightarrow$  거의 0인데...?  $\rightarrow$  랜덤예측이나 다름없는데..?

## Case 3. *F1-score의 한계: Y=1과 Y=0이 바뀐 경우*

CASE3		예측( $\hat{Y}$ )	
		$\hat{Y}=0$	$\hat{Y}=1$
실제 (Y)	Y=0	1	5
	Y=1	4	90

- F1-score:  $[(2*1)/(2*1+4+5)] = 18.18\%$
- Case2에선 95.24%로 모델 성능이 좋음을 보였으나, TN을 고려하지 않기 때문에 Y=1과 Y=0이 바뀌면 성능 지표 또한 바뀐다



## Case 4.

<TN을 더 적게 맞췄을 때>

VS

<TN을 더 많이 맞췄을 때>

MODEL1		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 (Y)	Y=1	95	20
	Y=0	10	5

MODEL2		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 (Y)	Y=1	95	20
	Y=0	10	99

*F1-score*

TN 고려X → 두 모델의 평가지표가 86.36%로 동일

: 두 모델 성능이 같다고 평가해버린다!

## Case 4.

<TN을 더 적게 맞췄을 때>

VS

<TN을 더 많이 맞췄을 때>

MODEL1		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 (Y)	Y=1	95	20
	Y=0	10	5

MODEL2		예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실 제 (Y)	Y=1	95	20
	Y=0	10	99

### MCC

- Model1: 0.1292 → 0에 가깝다 → 의미 없는 모델
- Model2: 0.7355 → 좋은 성능의 모델



# 2

## ROC & AUC

## "ROC Curve"

: 모든 cut-off point에 대해 TPR(민감도)와 FPR(1- 특이도)을 나타낸 그림

- Confusion Matrix의 한계

Confusion  
Matrix

Cut-off point에 의존적

정보의 손실 발생

## "ROC Curve"

: 모든 cut-off point에 대해 TPR(민감도)와 FPR(1- 특이도)을 나타낸 그림

- Confusion Matrix의 한계

Confusion  
Matrix

Cut-off point에 의존적

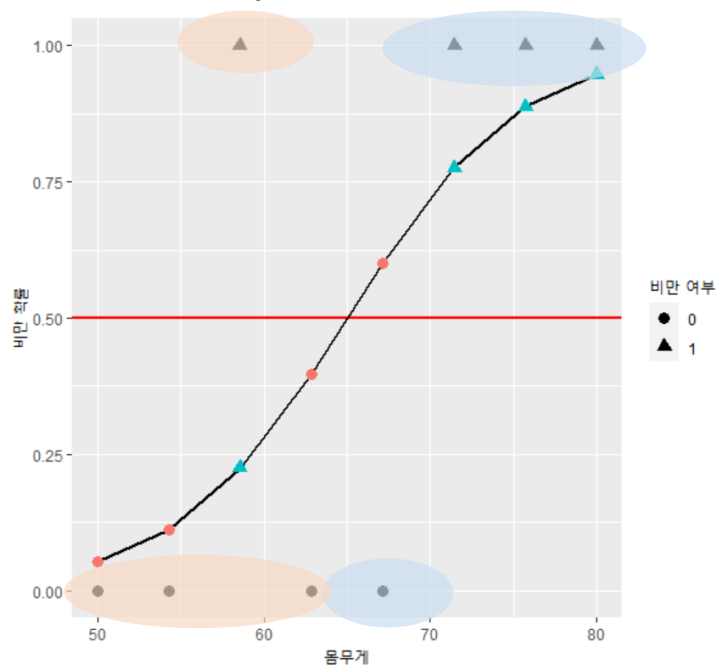
정보의 손실 발생

- Cut-off point에 의존적

*Example*

8명의 몸무게에 따른 비만 여부를 예측하는 문제

<Cut-off point = 0.5 일 때>



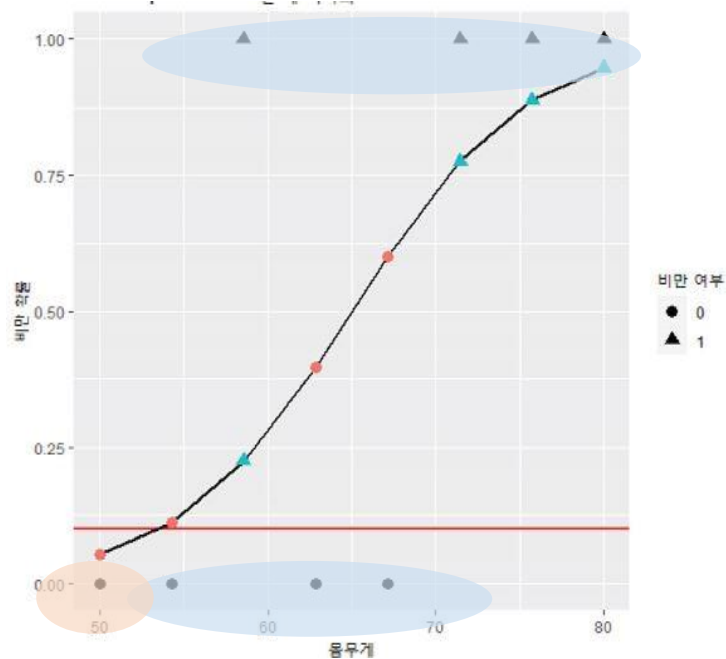
Cutoff point = 0.5		비만 예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 비만 (Y)	Y=1	3	1
	Y=0	1	3

- Cut-off point에 의존적

*Example*

8명의 몸무게에 따른 비만 여부를 예측하는 문제

<Cut-off point = 0.1 일 때>



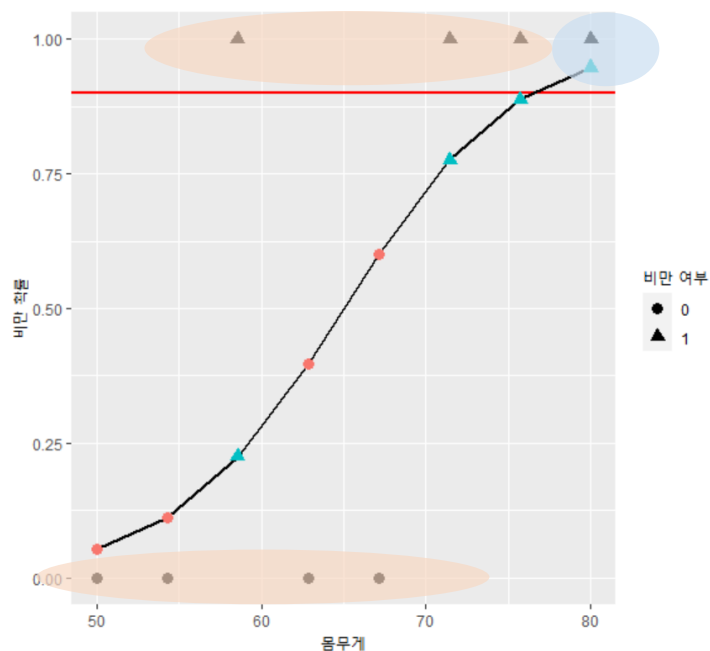
Cutoff point = 0.1		비만 예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 비만 (Y)	Y=1	4	1
	Y=0	3	0

- Cut-off point에 의존적

*Example*

8명의 몸무게에 따른 비만 여부를 예측하는 문제

<Cut-off point = 0.9 일 때>



Cutoff point = 0.9		비만 예측( $\hat{Y}$ )	
		$\hat{Y}=1$	$\hat{Y}=0$
실제 비만 (Y)	Y=1	1	3
	Y=0	0	4



# "ROC Curve"

: 모든 cut-off point에 대해 TPR (민감도)와 FPR(1- 특이도)을 나타낸 그림

- Confusion Matrix의 한계

Confusion  
Matrix

Cutoff point에 의존적

정보의 손실 발생

관측자가 선택한 cut-off point에 대한 정보만 보여주기 때문!

- ROC Curve 의 특징

1. 많은 정보를 포함

: 모든 cut-off point에 대해 예측 검정력을 구하기 때문

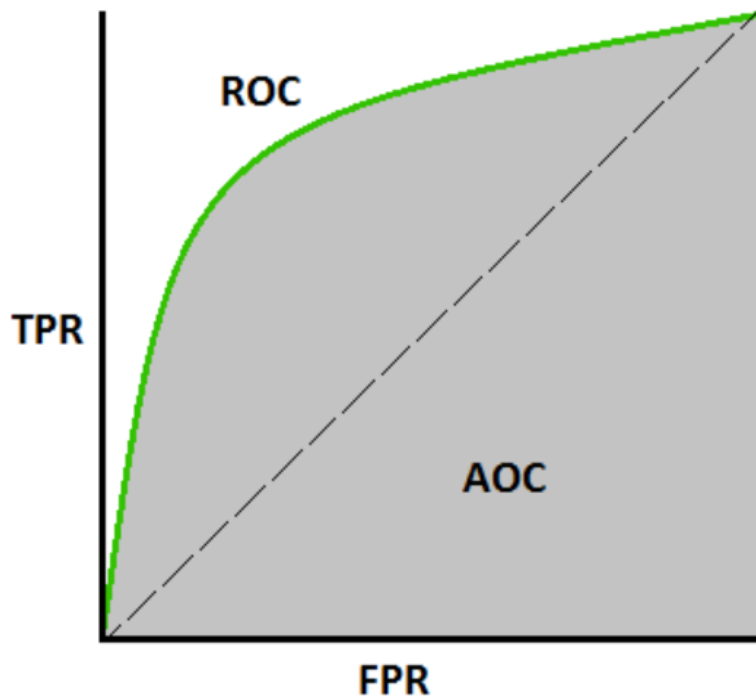
2. 가장 적합한 cut-off point를 찾을 수 있다

ROC 커브가 뭐길래?



- ROC Curve 형태

: 우상향하는 위로 볼록한 곡선 혹은 직선



- Y축:  $TPR = \frac{TP}{TP+FN}$

→ 예측과 실체가 일치

→ Y값이 클수록 좋음

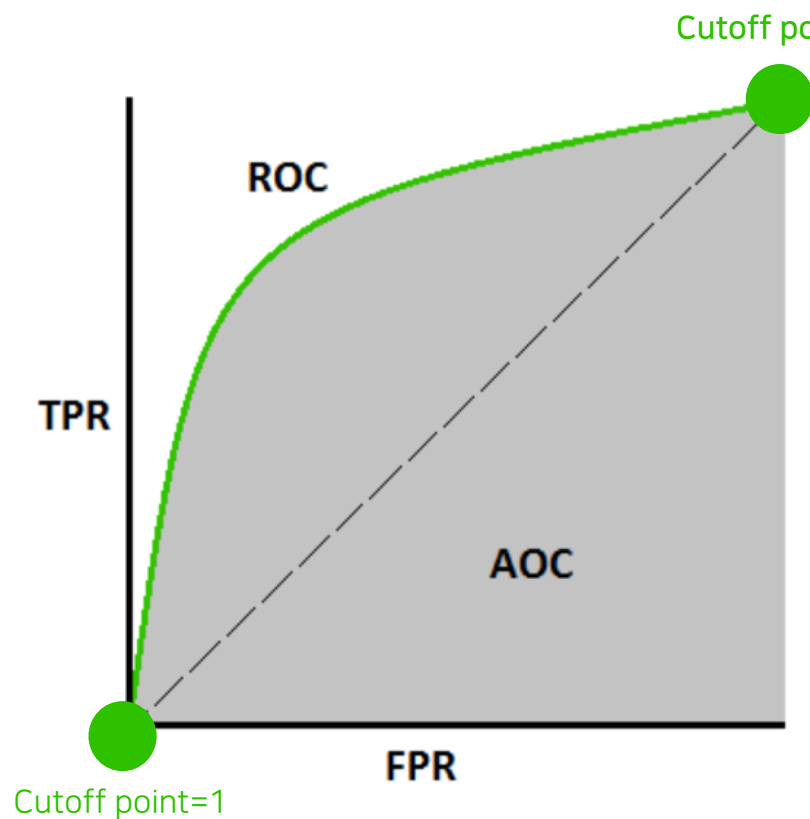
- X축:  $FPR = \frac{FP}{TN+FP}$

→ 예측과 실체가 불일치

→ X값이 작을수록 좋음

- ROC Curve 형태

: 우상향하는 위로 볼록한 곡선 혹은 직선



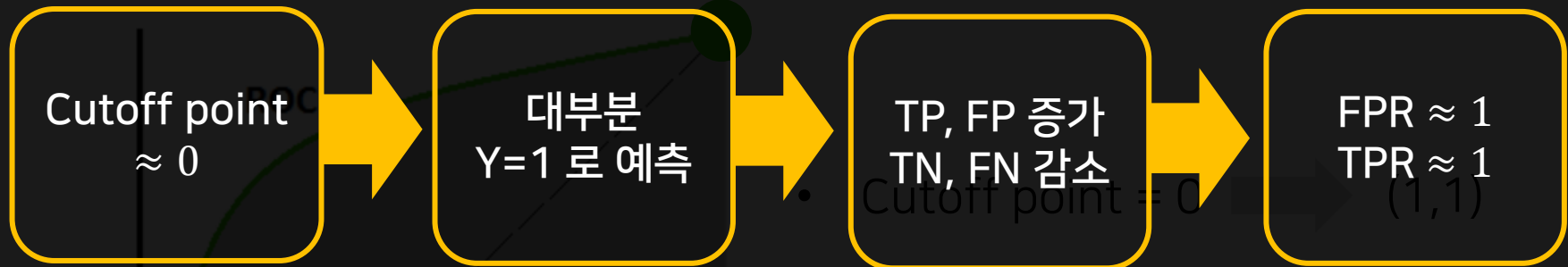
- Cutoff point  $\approx 0$   $\rightarrow$  (1,1)
- Cutoff point  $\approx 1$   $\rightarrow$  (0,0)

다음 장에서 자세히 알아보자!

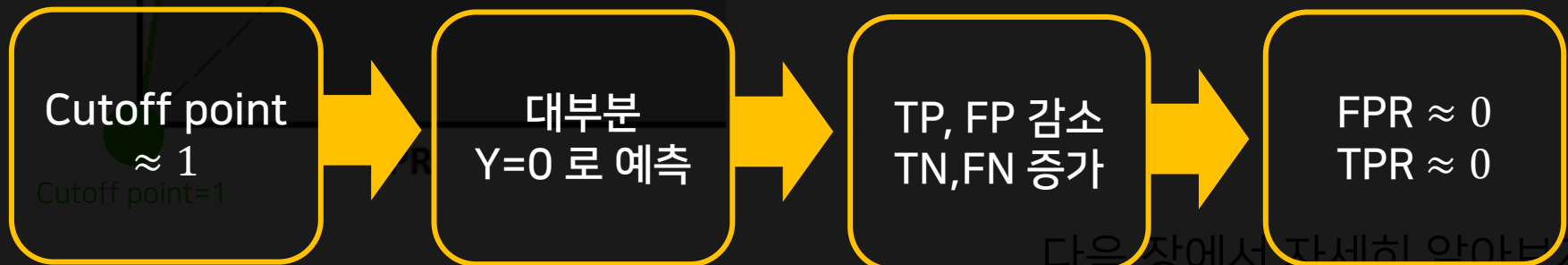


- ROC Curve
  - X축:  $FPR = \frac{FP}{TN+FP}$
  - Y축:  $TPR = \frac{TP}{TP+FN}$

Cutoff point  $\approx 0$  일 때 (1,1) 로 가는 알고리즘



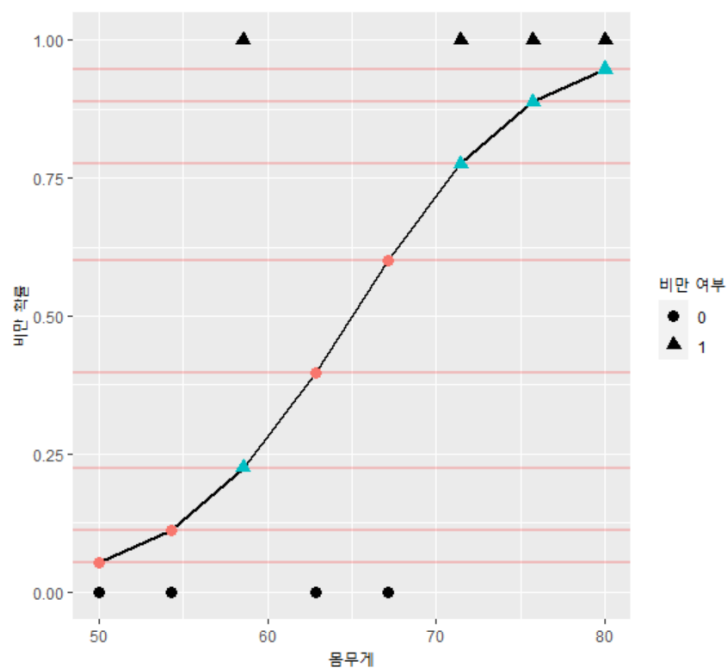
Cutoff point  $\approx 1$  일 때 (0,0) 로 가는 알고리즘



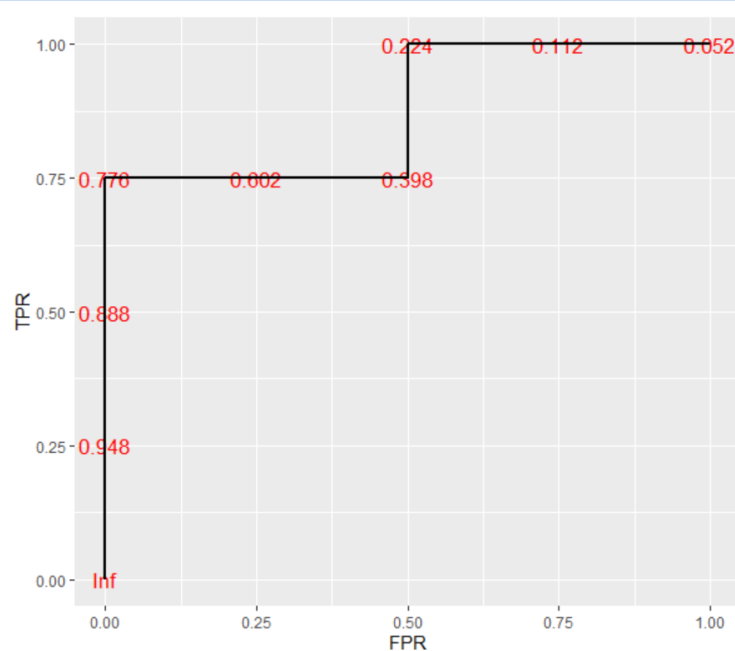
다음 장에서 자세히 알아보자!

- ROC Curve로 적합한 Cut-off point 찾기

### ① Cut-off point 선택

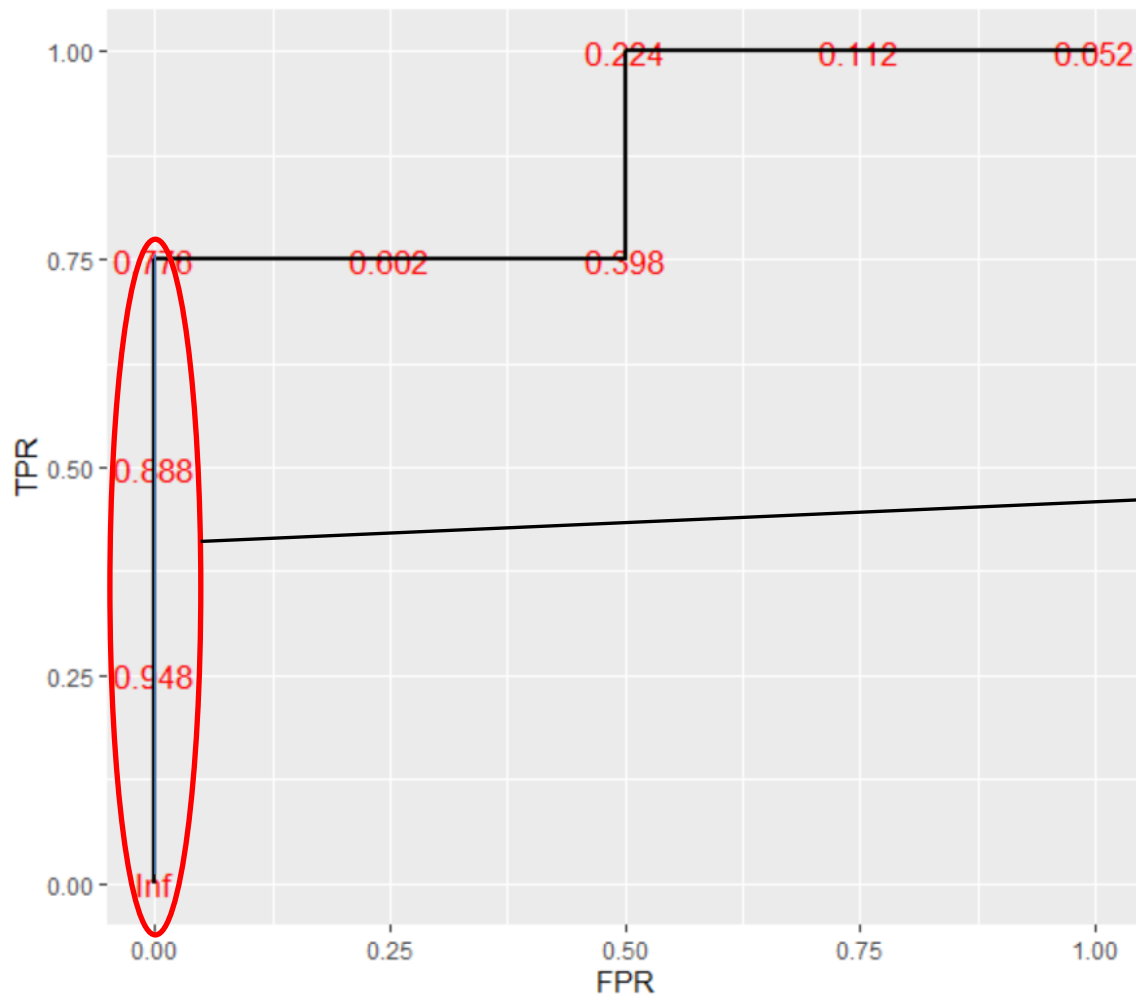


### ② Confusion Matrix 계산 후 ROC 커브 그린 결과



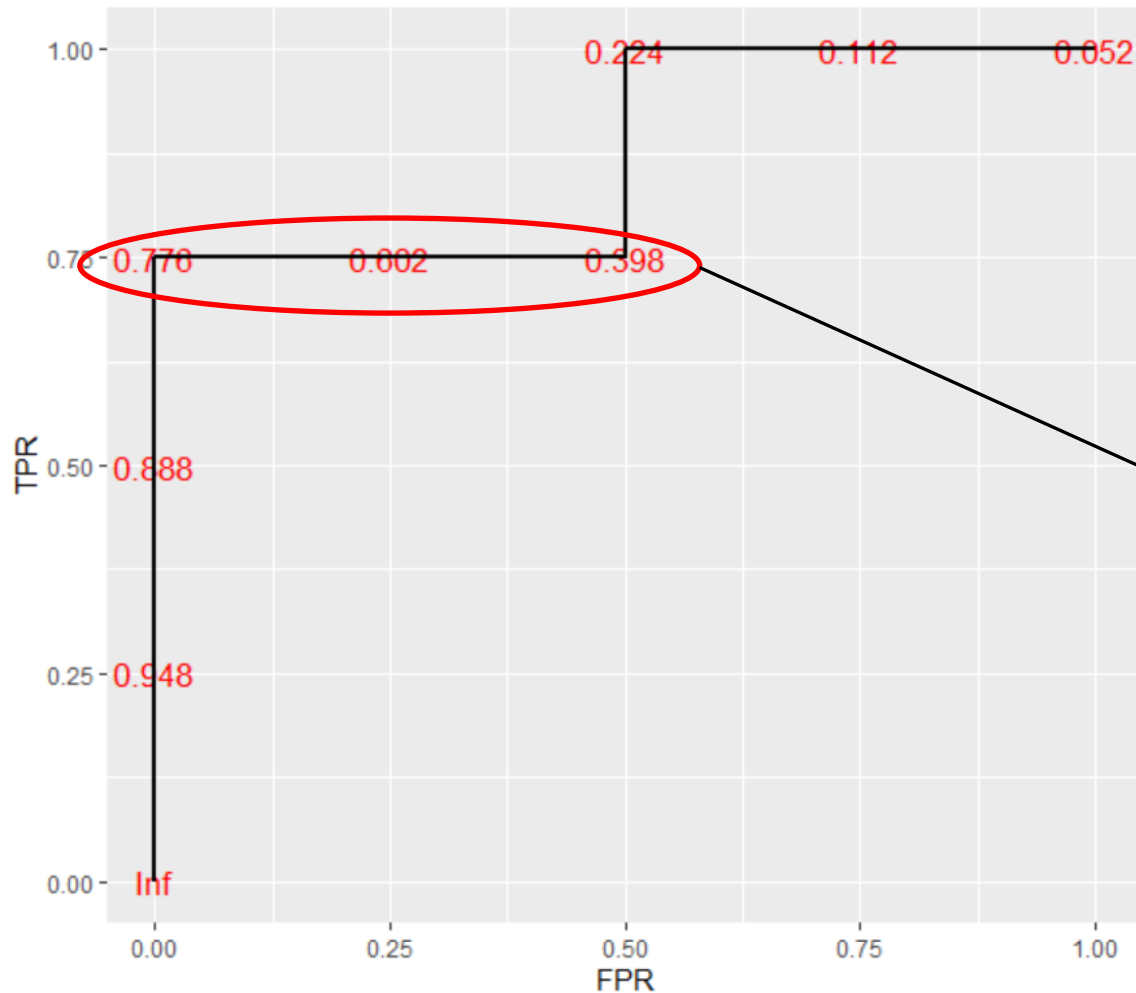
가장 적합한 Cut-off point 는?

- ROC Curve로 적합한 Cut-off point 찾기



동일한 X값에서  
Y값이 클수록 좋음  
↓  
Cutoff point = 0.77

- ROC Curve로 적합한 Cut-off point 찾기

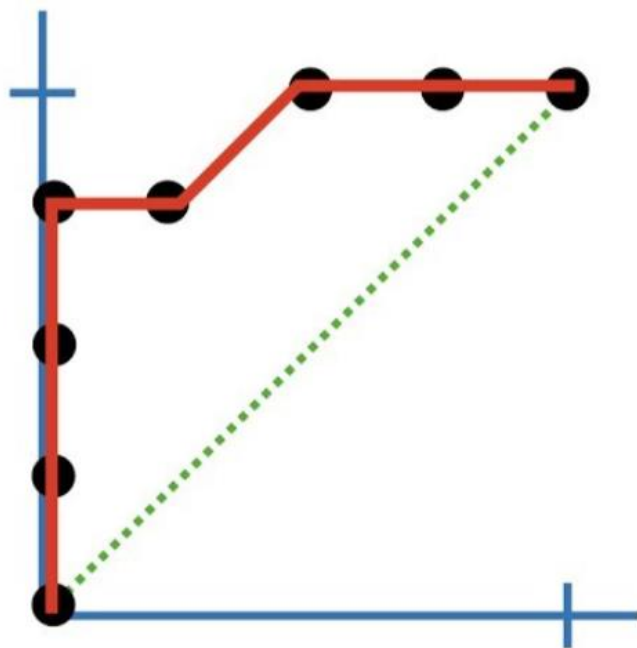


동일한 Y값에서  
X값이 작을수록 좋음  
↓  
Cutoff point = 0.77

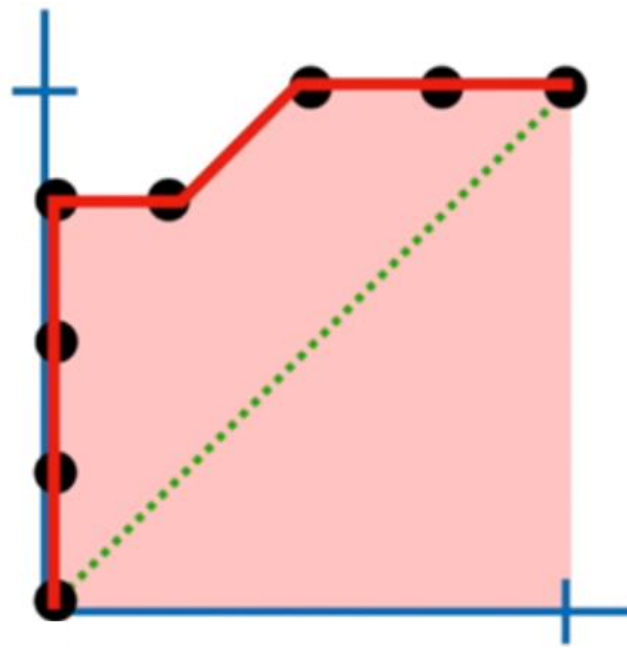


# "AUC (Area Under the Curve)"

: ROC Curve 아래의 면적



ROC



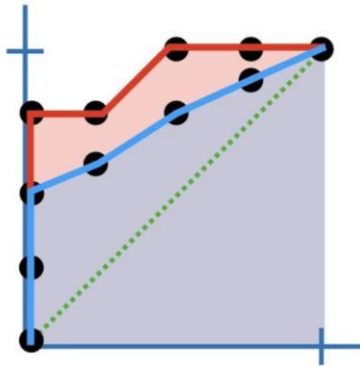
AUC

## AUC 의 특징

- 모델의 성능을 비교하는 지표

Cut-off point와 상관없이 모델의 성능 측정 가능

- $0 \leq AUC \leq 1$
- AUC가 1에 가까워질수록 모델의 성능이 좋음



$$AUC_{Blue} < AUC_{Red} < 1$$

→ 빨간색 모델의 성능이 더 좋음



## 왜 AUC가 1에 가까울수록 좋은가?

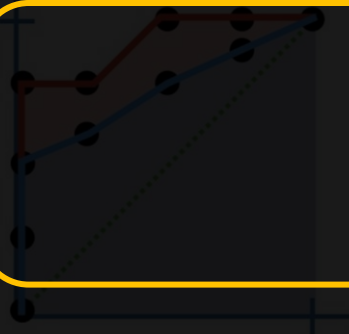
### AUC의 특징

- $0 \leq AUC \leq 1$   
X값이 고정되었을 때, Y값이 클수록 (위로 볼록할수록) 좋은 모델
- 모델의 성능을 비교하는 지표

Cut-off point와 상관없이 성능 측정 가능

- AUC가 1에 가까워질수록 성능이 좋음

### "AUC"

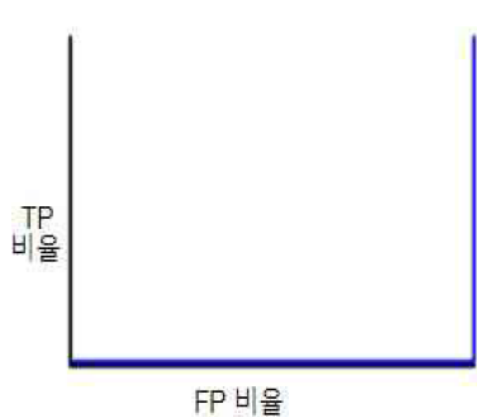


$$AUC_{Blue} < AUC_{Red} < 1$$

AUC 값이 클수록 좋은 모델

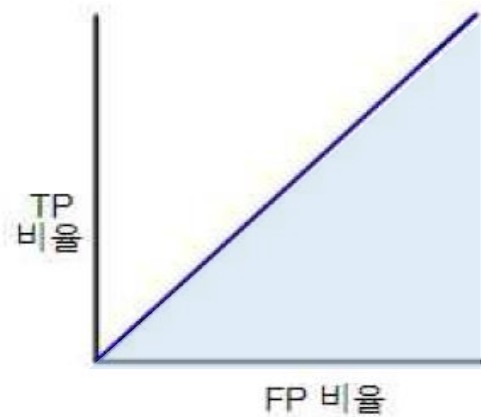
→ 파란색 모델의 성능이 더 좋음

- AUC 비교



AUC = 0

100%  
반대로 예측



AUC = 0.5

무작위 추측과  
같은 성능



AUC = 1

100%  
정확히 예측  
Overfitting  
가능성

- AUC 비교

AUC 값	모델 성능
0.9 ~ 1	Excellent
0.8 ~ 0.9	Good
0.7 ~ 0.8	Normal
0.6 ~ 0.7	Poor
0.5 ~ 0.6	Fail

절대적인 기준은 아니니 참고만...!



3

Sampling

## "비대칭 데이터의 문제"

: Y변수의 **클래스 비율의 차이가 클 때** 나타나는 문제

*비대칭 데이터에서는...*

- 단순히 우세한 클래스를 택하는 모델의 정확도가 높게 나타남  
→ **성능 판별 어려움**
- 소수 클래스의 재현율이 낮아짐

 **샘플링으로 해결!**



## Sampling 종류

### "비대칭 데이터의 문제"

#### 1. 언더 샘플링(Under-Sampling)

- 랜덤 언더 샘플링(Random Under-Sampling)

비대칭 데이터에서는...

#### 2. 오버 샘플링(Over-Sampling)

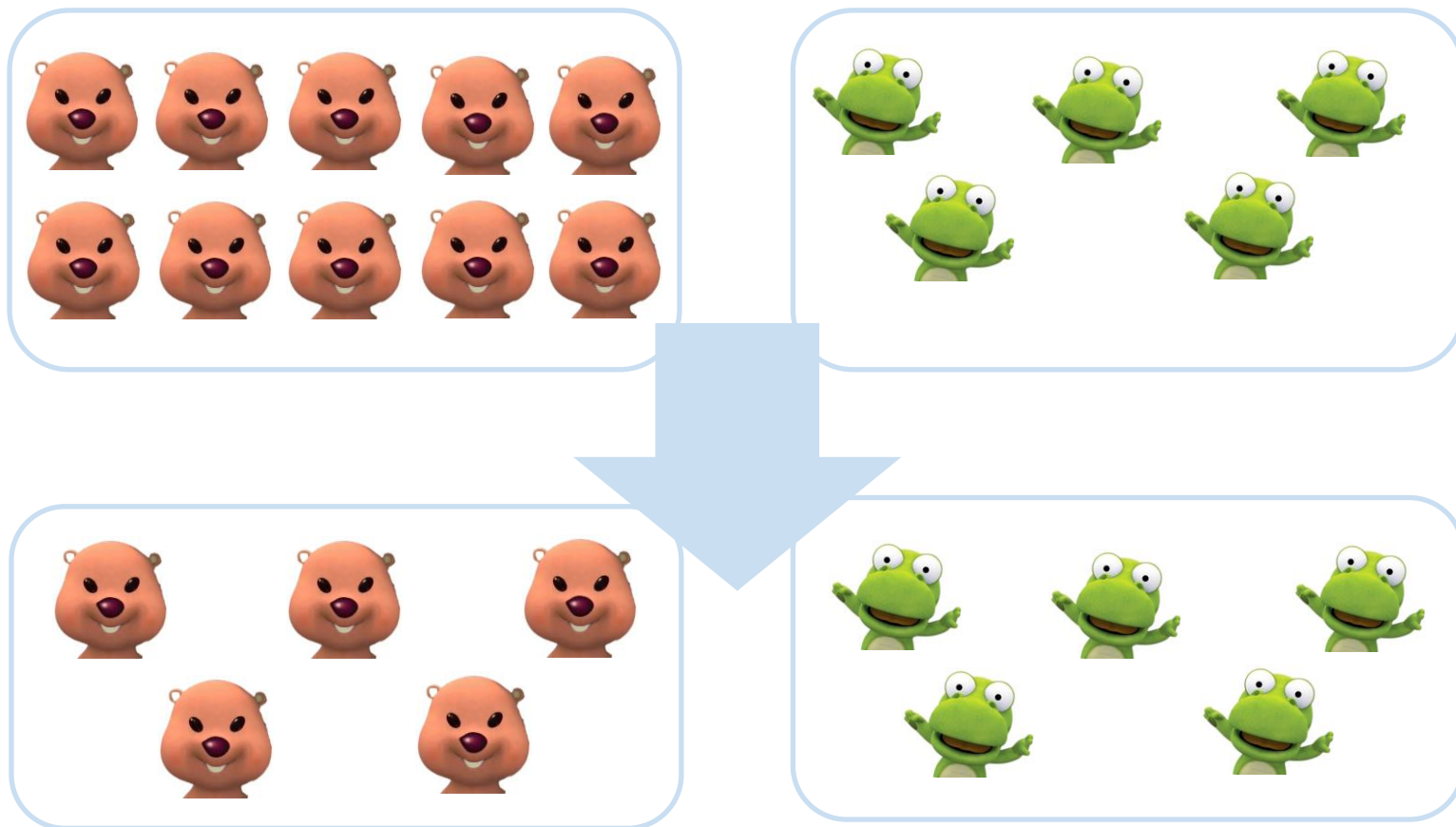
- 랜덤 오버 샘플링(Random Over-Sampling)
- SMOTE
- Modified SMOTE

#### 3. 복합 샘플링(Combining Over and Under-Sampling)



# "언더 샘플링(Under-Sampling) "

: 다수 클래스의 데이터를 소수 클래스에 맞추어 감소시킴



# "언더 샘플링(Under-Sampling) "

: 다수 클래스의 데이터를 소수 클래스에 맞추어 감소시킴

## 장점

메모리 사용, 처리속도 측면에서 유리

## 단점

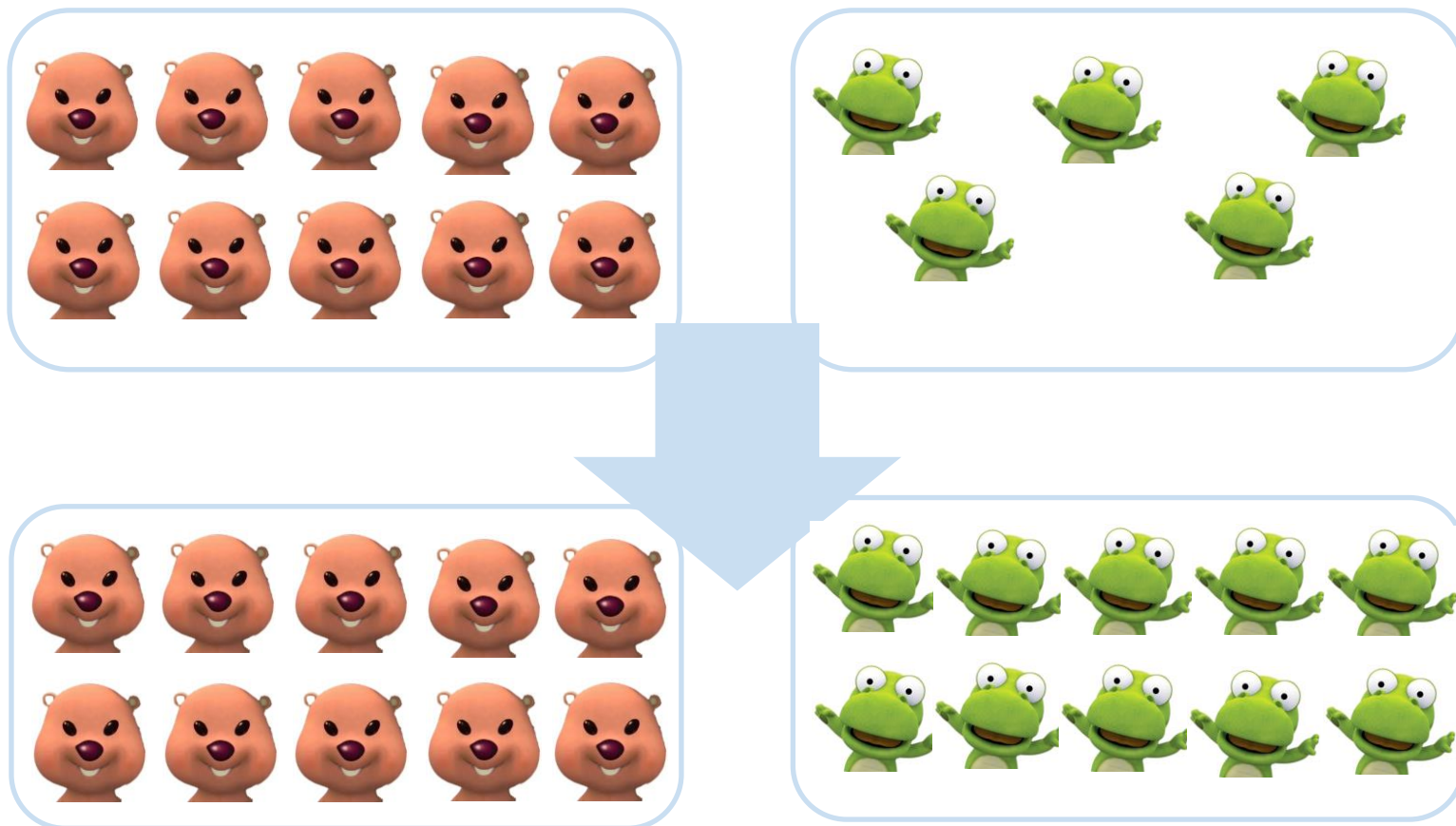
데이터 손실로 인한 정보 누락 가능성



보통 정보를 누락시키지 않는 오버 샘플링 많이 사용

# "오버 샘플링(Over-Sampling)"

: 소수 클래스의 데이터를 다수 클래스에 맞추어 증가시킴



# "오버 샘플링(Over-Sampling) "

: 소수 클래스의 데이터를 다수 클래스에 맞추어 증가시킴

## 장점

정보의 손실이 없기 때문에, under-sampling에 비해 성능이 좋다

## 단점

메모리 사용, 처리속도 측면에서 불리

## "Random Under-Sampling"

: 무작위로 다수 클래스의 데이터를 줄임

주의!

샘플링으로 얻은 표본이 정확한 대표성을 가지지 못하면 부정확한 결과 초래

## "Random Over-Sampling"

: 무작위로 소수 클래스의 데이터를 복제

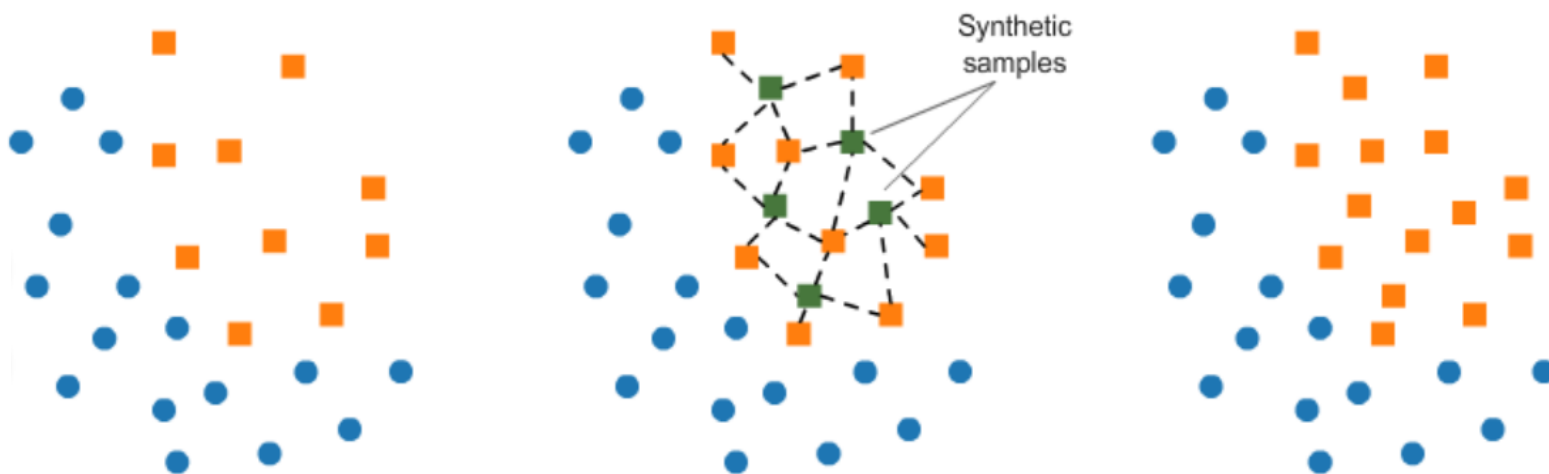
주의!

임의로 데이터를 복제하므로 과적합 가능성

# "SMOTE (Synthetic Minority Over-sampling Technique)"

## 알고리즘

1. 소수 클래스의 데이터 하나를 선택
2. 선택된 데이터와 가까운 소수 클래스 데이터에서 랜덤하게  $k$ 개 선택
3. 선택된 데이터와  $k$ 개의 데이터 사이의 가상의 직선 상에 소수 클래스 데이터 생성



## "SMOTE (Synthetic Minority Over-sampling Technique)"

### 장점

- 데이터를 복제하는 대신 가상의 데이터를 생성하므로, Overfitting 가능성 줄어듦
- 정보 손실 우려 없음

### 단점

- 데이터들의 위치는 고려하지 않음
  - 서로 다른 클래스가 겹치거나 노이즈 생성될 수 있음
- 고차원 데이터에 효율적이지 않음

# "MSMOTE" (Modified Synthetic Minority Over-sampling Technique)

: 소수 클래스의 분포와 잠재적인 노이즈 고려

## 특징

소수 클래스 데이터 간의 거리를 기준으로 세 가지 그룹으로 분류

- Security/safe samples: 분류 모델의 성능을 높이는 데이터
- Latent noise samples: 분류 모델의 성능을 낮추는 데이터
- Border samples: 두 그룹에 속하지 않는 데이터

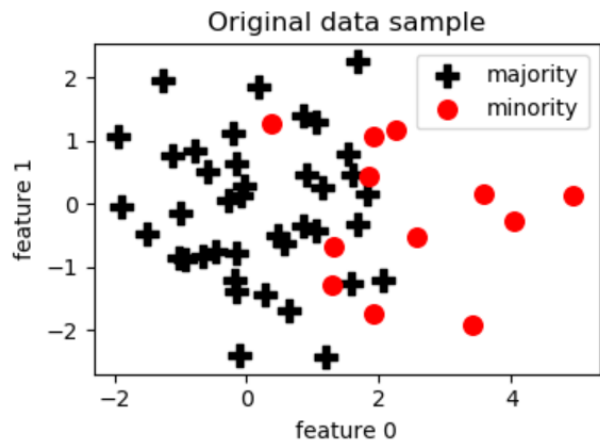


Security/safe samples 위주로 데이터 생성

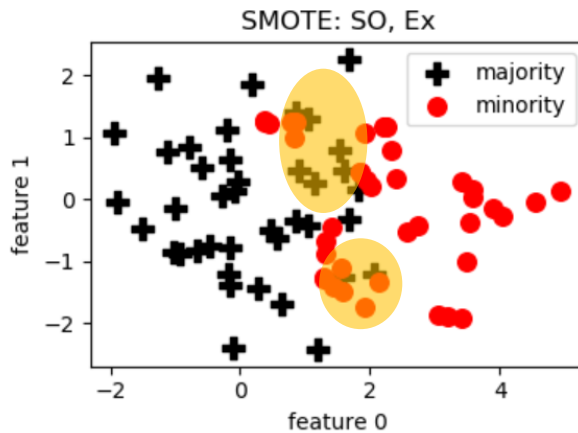
Latent noise sample에 대해서는 데이터 생성 X



# SMOTE vs MSMOTE



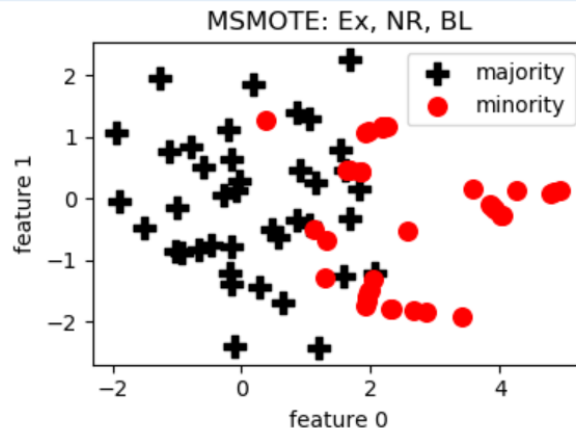
S  
M  
O  
T  
E



노이즈  
&  
클래스  
중복



M  
S  
M  
O  
T  
E

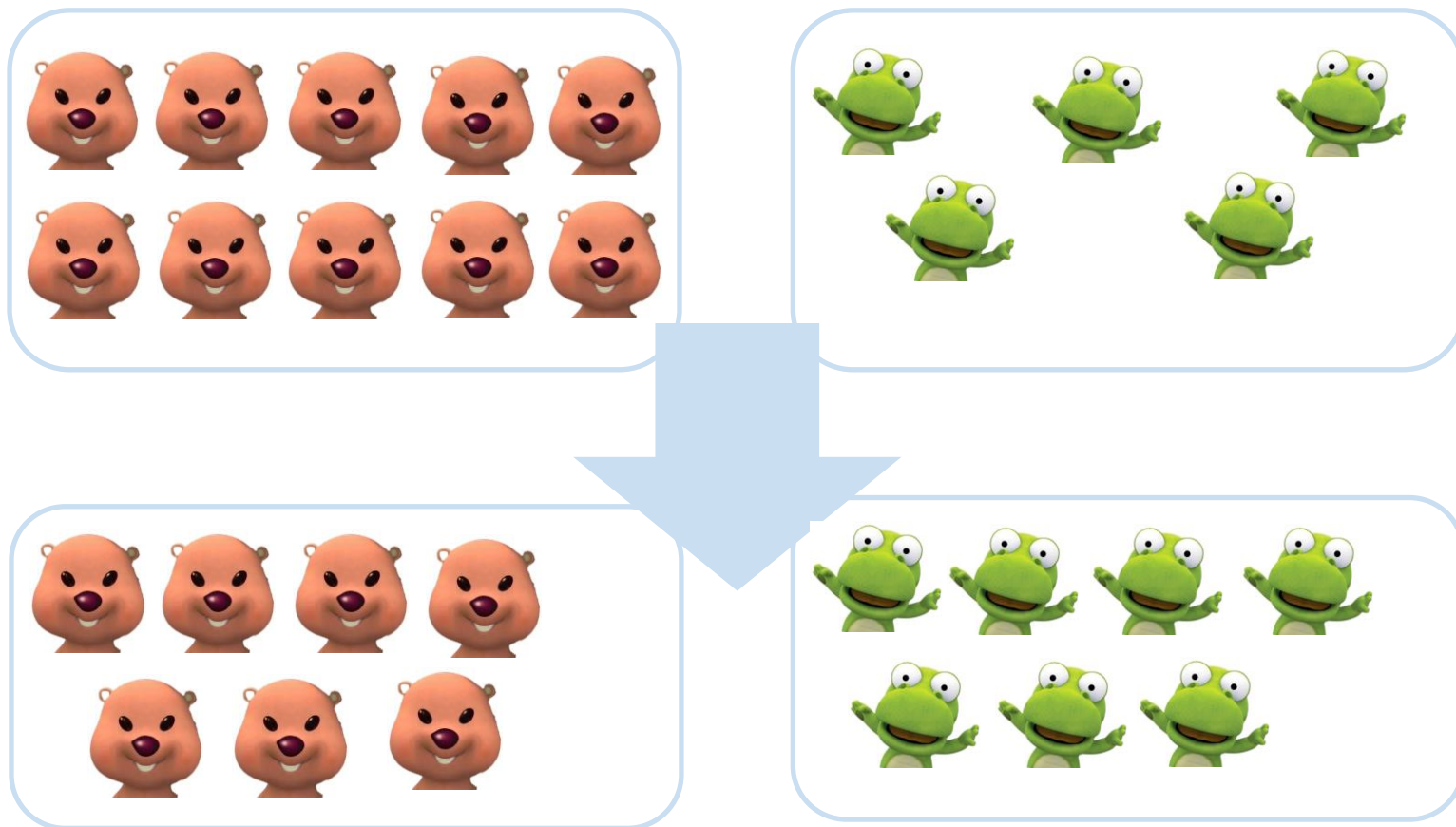


노이즈  
&  
클래스  
중복



## “복합 샘플링”

: 언더 샘플링과 오버 샘플링을 함께 진행 → 장단점도 반-반!



4

Encoding

# Encoding

인코딩

## 컴퓨터 공학적

사용자가 입력한  
문자나 기호  
↓  
컴퓨터가  
이용할 수 있는 신호

## 데이터 분석

범주형 변수 값을 **수치화**  
↓  
컴퓨터가  
읽을 수 있는 값

# Encoding 인코딩

- 필요성

1. 수치형 변수보다 범주형 변수가 더 많은 경우가 대부분

2. 수치형 변수만을 설명변수로 받는 분석기법 사용 가능

→ 이번 주 패키지 과제에 있는 XGBoost랄까..?

→ 다양한 회귀계열 모델도 사용할 수 있게 된다!

- 2-0번. XGboost 기본 세팅

참고) xgboost 패키지를 사용하세요.

- 2-0-1번. Train, Test에 있는 범주형 변수들을 one-hot-encoding 해주세요.

(뜨거웠던 패키지와의 기억들 생생하쥬..?)

참고) Xgboost는 numeric 변수만 받으므로 필수적으로 범주형 변수에 대해 encoding을 해야 합니다.

- Encoding의 종류

Classic	Contrast	Baysian	기타
Ordinal	Simple	Target	Frequency
One-hot	Sum	Leave One Out	
Label	Helmert	Weight of Evidence	
Binary	Reverse Helmert	Probability Ratio	
BaseN	Forward Difference	James Stein	
Hashing	Backward Difference	M-estimator	
	Orthogonal Polynomial	Ordered Target Encoding	

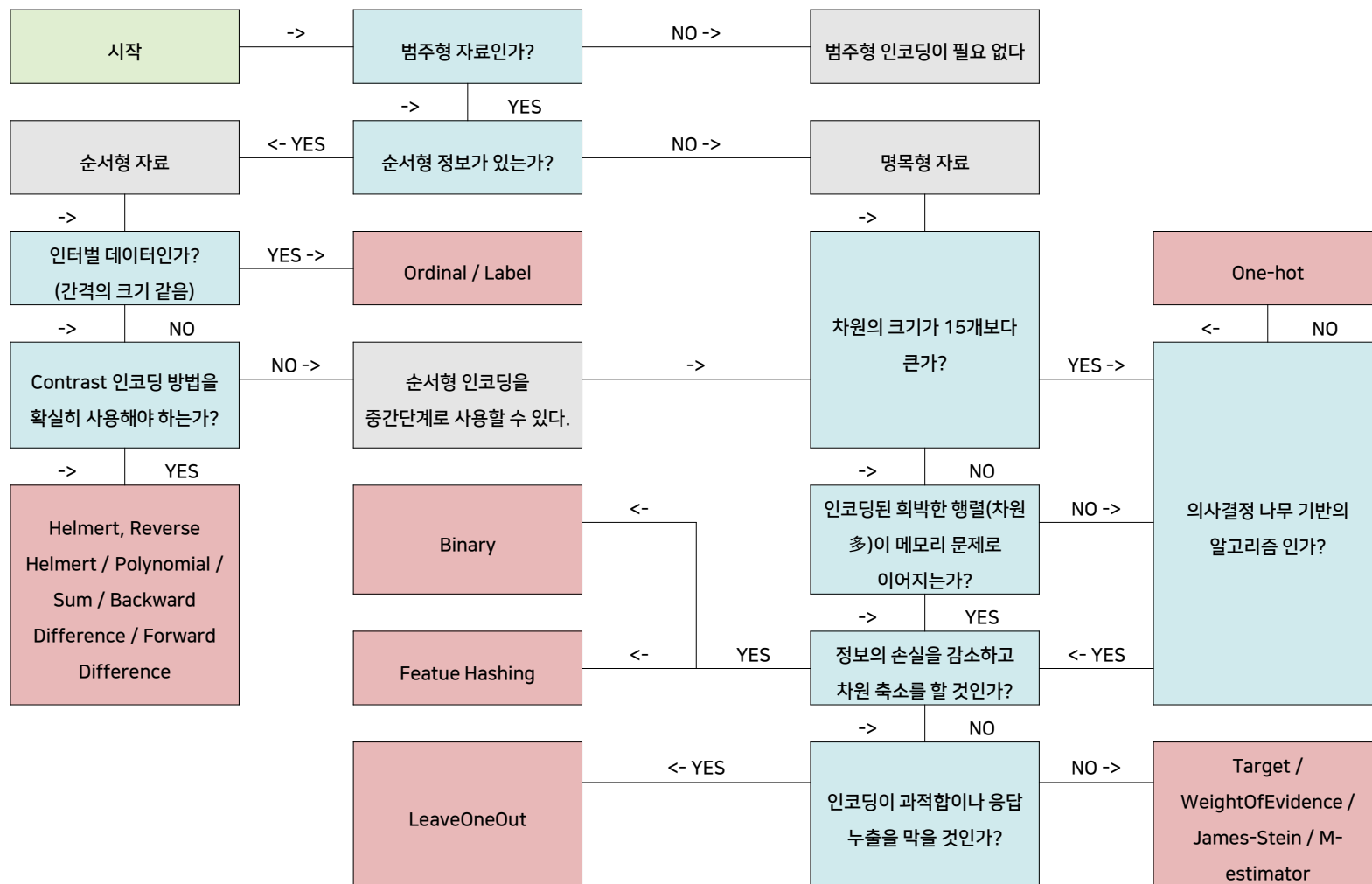
- Encoding의 종류

Classic	Contrast	Baysian	기타
Ordinal	Simple	Target	Frequency
One-hot	Sum	Leave One Out	
Label	Helmert	Weight of Evidence	
Binary	Reverse Helmert	Probabilistic	
Base	Forward Difference	Bayesian	
Hashing	Backward Difference	M-estimator	
	Orthogonal Polynomial	Ordered Target Encoding	

왜 이리 종류가 많은지.. ^^  
선택과 집-중 감성으로 가자!



## 인코딩 알고리즘






# "One-hot Encoding"

*Treatment Encoding, Dummy Encoding*

MBTI	
ESFJ	
ISFJ	
ENTP	
INTP	



ESFJ	ISFJ	ENTP	INTP
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

: 가변수(dummy variable)를 만들어 주는 것

# "One-hot Encoding"

*Treatment Encoding, Dummy Encoding*

MBTI	ESFJ (기준 범주)	ISFJ	ENTP	INTP
ESFJ	1	0	0	0
ISFJ	0	1	0	0
ENTP	0	0	1	0
INTP	0	0	0	1

해당 범주에는 1, 그 외에는 0을 입력

# "One-hot Encoding"

*Treatment Encoding, Dummy Encoding*

MBTI	ESFJ (기준 범주)	ISFJ	ENTP	INTP
ESFJ	1	0	0	0
ISFJ	0	1	0	0
ENTP	0	0	1	0
INTP	0	0	0	1

한 열(기준 범주)을 삭제! → 더미 변수 간 다중공선성 해결!

: (J-1)개의 더미변수로 J개의 수준을 갖는 인자 표현 가능



Tree 기반 모델의 경우,  
N개의 가변수를 생성하자!

"One-hot Encoding"

Treatment Encoding, Dummy Encoding

WHY?

MBTI	ESFJ	ISFJ	ENTP	INTP
ESFJ	1	0	0	0
ISFJ	0	1	0	0
ENTP	0	0	1	0
INTP	0	0	0	1

Tree 기반 모델은

사용 가능한 모든 부분을 활용해 트리 생성

삭제되는 기준 범주가 트리를 생성하는 데에 매우 중요한  
요소였다면, 트리 모델이 잘못 학습될 수 있다!

한 열(기준 범주)을 삭제!

: (J-1)개의 더미변수로 J개의 수준을 갖는 인자 표현 가능

# "One-hot Encoding"

*Treatment Encoding, Dummy Encoding*

- 장점
  1. 해석의 편의성
    - 기준 범주를 기준으로 해석
  2. 명목형 변수 값들의 특성을 가장 잘 반영
  3. 지도학습의 경우, Data Leakage 발생 X

*Data Leakage는 나가 있어..*





# Data Leakage란?

*"One-hot Encoding"*

: 데이터가 누출된 것 atment Encoding, Dummy Encoding

- 장점

1. 해석의 편의성

즉, 정답이 존재하는 지도학습에서

: 기준 범주를 기준으로 해석

반응변수 Y에 대한 정보가

2. 다중공선성 해결

모델 학습 시 사용한 설명변수 X에 들어가는 것

: 한 변수가 나머지로 설명되는 것을 방지

3. 명목형 변수 값들의 특성을 가장 잘 반영



4. 지도학습의 경우, Data Leakage 발생 X

**과적합 야기!!**

# "One-hot Encoding" Treatment Encoding, Dummy Encoding

- 단점

범주형 변수의 level이 높거나 범주형 변수가 많은 데이터의 경우,  
너무 많은 열(차원/가변수)이 생긴다.

(MBTI 총 16개니까 다 하면 15개 변수 생성...)



학습속도 ↓

&

상당한 Computing Power 필요

## *“Label Encoding”*

MBTI
ESFJ
ISFJ
ENTP
INTP



MBTI	점수
ESFJ	1
ISFJ	2
ENTP	3
INTP	4

- 단순히 **점수를 할당** → 어떤 수를 부여하든 상관 X
- 명목형 자료에 많이 사용
- 할당된 점수들 간의 순서나 연관성 X



# "Label Encoding"

- 장점
  - 차원이 늘어나지 않는다!
- 단점
  - Label 간의 순서나 연관성이 존재한다고 학습될 수 있음
  - 즉, 정보왜곡의 가능성 多

(넌 그저 라벨인데 왜..)



## *“Ordinal Encoding”*

행복정도
매우 불행
불행
행복
매우 행복



행복 정도	점수
매우 불행	1
불행	2
행복	3
매우 행복	4

- **순서형 정보에 대응되는 점수를 할당** → 대체로 1부터 부여
- 순서형 자료에 사용
- Label Encoding과 달리 **할당된 점수들 간의 순서나 연관성 0**

## *“Ordinal Encoding”*

- 장점
  1. 차원이 늘어나지 않는다
  2. **순서형 정보** 이용
- 단점
  1. 순서형 정보가 있을 때만 사용 가능
  2. 범주 간의 순서에 따라 점수를 할당하는 데 있어,  
**어느 정도의 차이를 둘지 고려하기 어려움**  
: ‘행복’과 ‘매우 행복’의 차이는 2일 수도, 3일 수도 있다!  
→ **도메인 지식(선행연구나 기타 코드북)을 활용**

# "Target Encoding" Mean Encoding

범주형 변수의 각 수준에 대해, **반응 변수 Y의 평균**으로 점수 할당

[Y] 통학 시간 (분)	[X] 팀	Target Encoding
100	범주	58.3
70	범주	58.3
5	범주	58.3
15	회귀	32.5
50	회귀	32.5
20	선대	35
50	선대	35



$$\frac{100+70+5}{3} = \text{범주팀 통학시간의 평균}$$

# "Target Encoding"

*Mean Encoding*

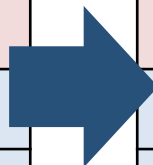
- 장점
  1. 차원이 늘어나지 않는다
  2. Label Encoding과 달리, 할당된 점수에 당위성이 존재
- 단점
  1. 반응변수 Y값을 사용하므로, Data Leakage 발생 가능성 ↑
    - 과적합이 일어날 가능성 ↑
  2. Train set에는 없는 범주가 Test set에 존재할 경우,  
어떻게 점수를 할당해야 할지 애매

# *“Leave One Out Encoding”* L00 Encoding

- 현재 행을 제외하고 평균을 구해 이를 점수로 할당하는 방식
  - Outlier의 영향을 줄일 수 있음
- Target Encoding (Mean Encoding)과 매우 유사한 방법
  - L00 Encoding은 같은 범주더라도 다른 점수를 할당할 수 있음  
즉, 다양한 라벨링 가능!

# "Leave One Out Encoding" LOO Encoding

[Y] 자취여부	[X] 팀	Target Encoding
0	범주	25%
0	범주	25%
0	범주	25%
1	범주	25%
1	회귀	66.7%
1	회귀	66.7%
0	회귀	66.7%
0	선대	33.3%
0	선대	33.3%
1	선대	33.3%



[Y] 자취여부	[X] 팀	LOO Encoding
0	범주	33.3%
0	범주	33.3%
0	범주	33.3%
1	범주	0%
1	회귀	50%
1	회귀	50%
0	회귀	100%
0	선대	50%
0	선대	50%
1	선대	0%

# “Leave One Out Encoding” LOO Encoding

- 장점
  1. 차원이 늘어나지 않는다
  2. Outlier의 영향 ↓
  3. **Direct** Response Leakage 방지
    - 현재 행을 제외하고 평균을 계산하기 때문에, **Direct**한 Data Leakage는 발생 X
- 단점
  1. 여전히 **Data Leakage** 발생 가능성 존재
  2. Train set에는 없는 범주가 Test set에 존재할 경우,  
어떻게 점수를 할당해야 할지 애매



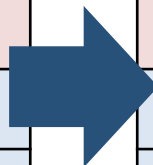
# "Ordered Target Encoding" CATBOOST Encoding

- 현재 행 이전의 값들을 사용하여 구한 평균을 점수로 할당하는 방법
- Target Encoding (Mean Encoding)과 매우 유사한 방법
  - Ordered Target Encoding은  
같은 범주더라도 다른 점수를 할당할 수 있음
- 부스팅 모델 중 하나인 CATBOOST에서 사용하는 인코딩 방법

(지난학기 데마팀 클린업에 CATBOOST에 대한 설명이 있다는 사실...!!)

# "Ordered Target Encoding" CATBOOST Encoding

[Y] 자취여부	[X] 팀	Target Encoding
0	범주	25%
0	범주	25%
1	범주	25%
0	범주	25%
1	회귀	66.7%
1	회귀	66.7%
0	회귀	66.7%
1	선대	33.3%
0	선대	33.3%
0	선대	33.3%



[Y] 자취여부	[X] 팀	CATBOOST Encoding
0	범주	0%
0	범주	0%
1	범주	0%
0	범주	33.3%
1	회귀	100%
1	회귀	100%
0	회귀	100%
1	선대	100%
0	선대	100%
0	선대	50%

# “Ordered Target Encoding” CATBOOST Encoding

- 장점
  1. 차원이 늘어나지 않는다
  2. Outlier의 영향 ↓
  3. Direct Response Leakage 방지
- 단점
  1. 여전히 Data Leakage 발생 가능성 존재
  2. Train set에는 없는 범주가 Test set에 존재할 경우,  
어떻게 점수를 할당해야 할지 애매

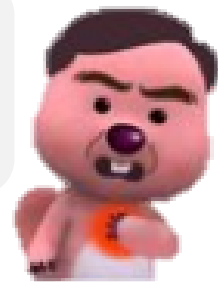


(LOO Encoding과 똑같다!)

**Q1.** Train set에는 없던 새로운 범주가  
Test set에 존재한다면 어떻게 하나요???

**Q2.** Test set에 아예  
반응 변수 Y가 없으면 어떡하죠???

**Q3.** 그래서 제일 좋은 인코딩 방법이 뭔데요???



**A1.** 해당 변수를 **아예 제거**하거나  
**재범주화**를 통해 해결!

**Q2.** Test set에 아예  
반응 변수 Y가 없으면 어떡하죠???

**Q3.** 그래서 제일 좋은 인코딩 방법이 뭔데요???

**A1.** 해당 변수를 **아예 제거**하거나  
**재범주화**를 통해 해결!

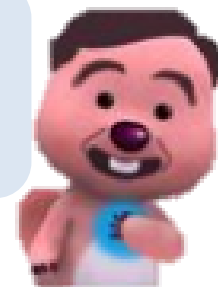
**A2.** 범주의 수준 별로 Encoding을 진행하면서  
할당했던 점수들을 그-대로 맵핑

**Q3.** 그래서 제일 좋은 인코딩 방법이 뭔데요???

**A1.** 해당 변수를 **아예 제거**하거나  
**재범주화**를 통해 해결!

**A2.** 범주의 수준 별로 Encoding을 진행하면서  
할당했던 점수들을 그-대로 맵핑

**A3.** 다양한 시도를 통해  
각 데이터마다 적합한 인코딩 방식을 찾자!





**THANK YOU**

