

# 범주형자료분석팀

2팀  
이지연  
심예진  
조장희  
조혜현  
진효주

# INDEX

---

0. 2주차 REVIEW

1. 혼동행렬

2. ROC 곡선과 AUC

3. Sampling

4. Encoding

0

2주차 REVIEW

## GLM(일반화선형모형, Generalized Linear Model)

반응변수가 **범주형 자료/count data**인 경우  
일반선형회귀모형 사용 불가



확장된 **GLM** 사용

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



GLM 구성 성분		
랜덤 성분	체계적 성분	연결 함수
$\mu (= E(Y))$	$\alpha + \beta_1 x_1 + \cdots + \beta_k x_k$	$g()$

## 유의성 검정

모형의 **모수 추정값이 유의한지**에 대한 검정  
**축소 모형의 적합도**가 좋은지에 대한 검정



- ✓  $H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0$
- ✓  $H_1 : \text{적어도 하나의 } \beta \text{ 는 } 0 \text{ 이 아니다.}$

## 가능도비 검정 (Likelihood Ratio Test)

귀무가설 하에서 계산되는 **가능도 함수  $l_0$**  와  
 MLE에 의해 계산되는 **가능도 함수  $l_1$**  의 차이 이용

검정통계량 :  $-2 \log \left( \frac{l_0}{l_1} \right) = -2(L_0 - L_1) \sim X^2$

내 얘기 하는 거야?

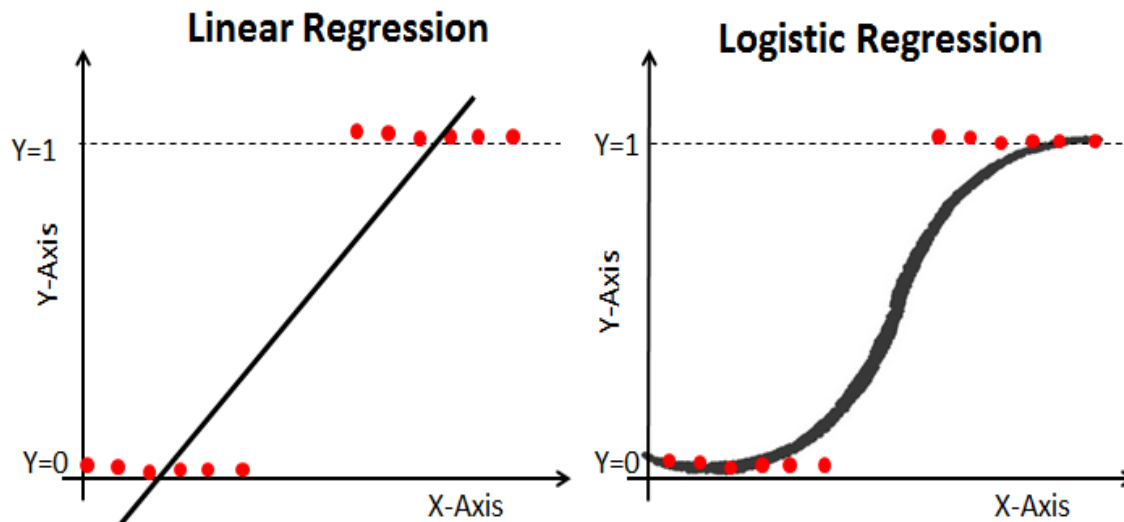


## 로지스틱 회귀 모형

반응변수 Y가 성공 혹은 실패를 나타내는 이항 자료인 회귀모형



$$\text{logit}[\pi(x)] = \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$



## 다범주 로짓 모형 & 포아송 회귀 모형



기준 범주 로짓 모형



누적 로짓 모형



$$\log\left(\frac{\pi_j}{\pi_J}\right) = \log\left(\frac{P(Y = j|X = x)}{P(Y = J|X = x)}\right)$$

$$= \alpha_j + \beta_j^A x_1 + \cdots + \beta_j^K x_K$$

반응변수가 명목형 다항자료일 때 사용

$$\text{logit}[P(Y \leq j|X = x)]$$

$$= \alpha_j + \beta_1 x_1 + \cdots + \beta_p x_p$$

반응변수가 순서형 다항자료일 때 사용

포아송 회귀 모형



$$\log(\mu) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

반응변수가 도수자료(count data) 일 때 사용

1

혼동행렬



## 혼동 행렬



분류 모델의 성능을 평가할 때 사용되는 지표

예측값( $\hat{Y}$ )이 실제 관측값( $Y$ )을 얼마나 정확히 예측했는지 보여주는 행렬

		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN



T(True)와 F(false): 실제와 예측이 같은지 혹은 다른지

P(Positive)와 N(Negative): 예측을 긍정 혹은 부정이라 했는지에 대한 여부

## 혼동 행렬



분류 모델의 성능을 평가할 때 사용되는 지표

예측값( $\hat{Y}$ )이 실제 관측값( $Y$ )을 얼마나 정확히 예측했는지 보여주는 행렬

		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

T(True)와 F(false): 실제와 예측이 같은지 혹은 다른지

✓ P(Positive)와 N(Negative): 예측을 긍정 혹은 부정이라 했는지에 대한 여부

## 혼동 행렬 해석

		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

예측이 **긍정(P)**? 부정(N)?

예측이 **실제로 맞았나(T)**?  
틀렸나(F)?

TP(True Positive)는 효주가 발표할 것이라고 예측했는데 실제 발표한 경우  
 FN(False Negative)는 효주가 발표하지 않을 것이라고 예측했는데 실제 발표한 경우

## 혼동 행렬 해석

		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

예측이 긍정(P)? 부정(N)?

예측이 실제로 맞았나(T)?  
틀렸나(F)?

TP(True Positive)는 효주가 발표할 것이라고 예측했는데 실제 발표한 경우  
FN(False Negative)는 효주가 발표하지 않을 것이라고 예측했는데 실제 발표한 경우

## 혼동 행렬 해석

## 혼동행렬의 한계



		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	이항변수(0 또는 1)로 범주화		FN
	$\hat{Y}=0$	FP	TN

1. 모형은 연속적인 예측값인 확률( $\hat{y}$ )을 결과로 반환하지만, 예측은 cut-off point를 기준으로

2. cut-off point인 값도 임의로 지정되기에 객관적 X

cut-off point가 달라지면 혼동행렬도 달라짐

예측이 긍정(P)? 부정(N)?

예측이 실제로 맞았나(T)? 틀렸나(F)?

클래스 불균형이 심한 경우 혼동행렬이 크게 바뀜

TP(True Positive)는 효주가 발표할 것이라고 예측했는데 실제로 발표한 경우 스포하자면...이러한 이유로 ROC 곡선이 등장한다!

FN(False Negative)는 효주가 발표하지 않을 것이라고 예측했는데 실제 발표한 경우

## 분류 평가지표



**범주형 자료분석**은 데이터마이닝 또는 머신러닝의 관점에서 **분류모델**

이번 파트에서는 분류 모델의 다양한 성능 평가지표에 대해 알아볼 예정!

경우에 따라 사용해야 하는 평가지표가 달라지므로 적절한 사용이 중요!



정확도  
(Accuracy)



정밀도  
(precision)



민감도  
(Sensitivity)



특이도  
(Specificity)



F1-score



MCC  
(매튜 상관계수)

## 정확도 (Accuracy/ACC/정분류율)



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

전체 경우에서 실제값과 예측값이 같은 경우의 비율 

즉, 예측이 실제 정답과 얼마나 정확히 일치하는지 나타내는 지표


		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

- ✓ 직관적이라 자주 쓰이는 지표
- ✓ 1에 가까울수록 좋은 모형
- ✓ Imbalanced data에서 모형 평가하는 경우 문제가 발생

## 정밀도 (Precision/PPV/Positive Predictive Value)



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

True라고 분류한 것 중에 실제로 True인 것의 비율   
즉, 예측한 성공 중 실제 성공은 얼마인지를 나타내는 지표

		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN



민감도 (Sensitivity/TPR/True Positive Rate) 또는 재현율(Recall)



$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

실제 Positive 중 예측과 실제 값이 Positive로 일치하는 비율 

즉, 실제로  $Y = 1$ 인 값 중에서  $\hat{Y} = 1$ 이라고 예측된 값의 비율

		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

## 정밀도와 민감도가 중요한 지표가 되는 경우



### 정밀도

Ex) 스팸 메일 판단

실제로 Positive인 스팸 메일을  
Negative인 일반 메일로 분류한다면  
사용자는 불편할 뿐이지만,  
만약 일반 메일을 스팸 메일로 분류하게 되면  
중요한 메일을 받지 못하게 되는 경우 발생



**False Positive**가  
더 critical한 경우에 사용



### 민감도

Ex) 질병 진단

건강한 사람을 양성이라고 예측한다면  
재검사를 하는 비용이 소모,  
암 환자를 음성이라고 예측한다면  
생명을 앗아가는 심각한 문제 발생



**False Negative**가  
더 critical한 경우에 사용

## 특이도 (Specificity/TNR/True Negative Rate)



$$\text{Specificity} = \frac{TN}{TN + FP}$$

실제 Negative 중 예측과 실제 값이 Negative로 일치하는 비율 

즉, 실제로  $Y = 0$ 인 값 중에서  $\hat{Y} = 0$ 이라고 예측된 값의 비율

		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

✓ 1에 가까울수록 좋음

## FPR(False Positive Rate, fall-out)



$$FPR = \frac{FP}{TN + FP} = 1 - Specificity$$

실제 Negative 중 Positive라고 예측된 비율 

		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

- ✓ TNR의 반대 개념
- ✓ FPR은 0에 가까울수록 좋음
- ✓ ROC 곡선의 X축

## F1-score



$$\frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FN + FP}$$

Precision과 Recall의 조화평균 

		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

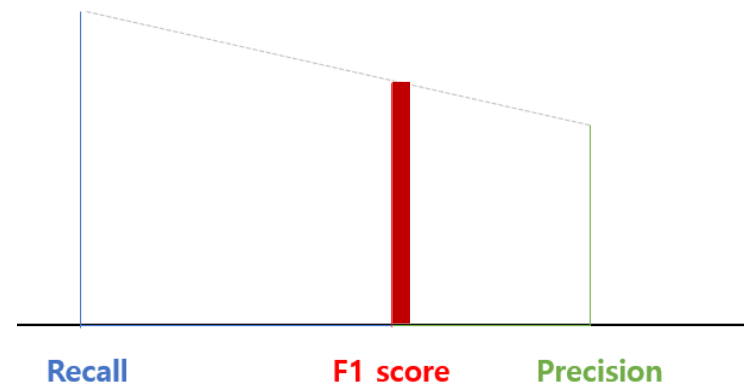
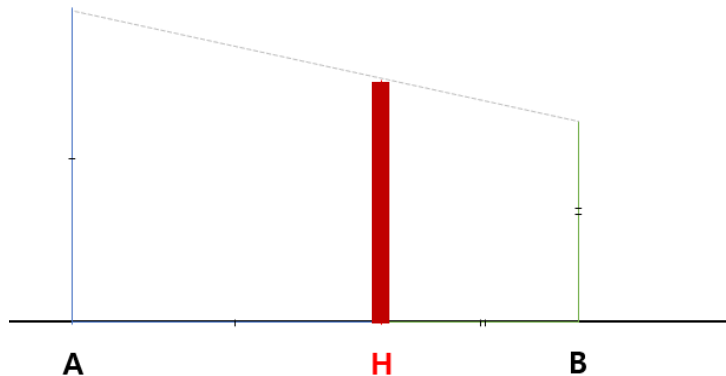
Precision

Recall(Sensitivity)

왜 산술평균이 아니고 조화평균을 구하는가?

조화평균 

변 AB에서 각 변의 길이가 만나는 지점까지의 거리



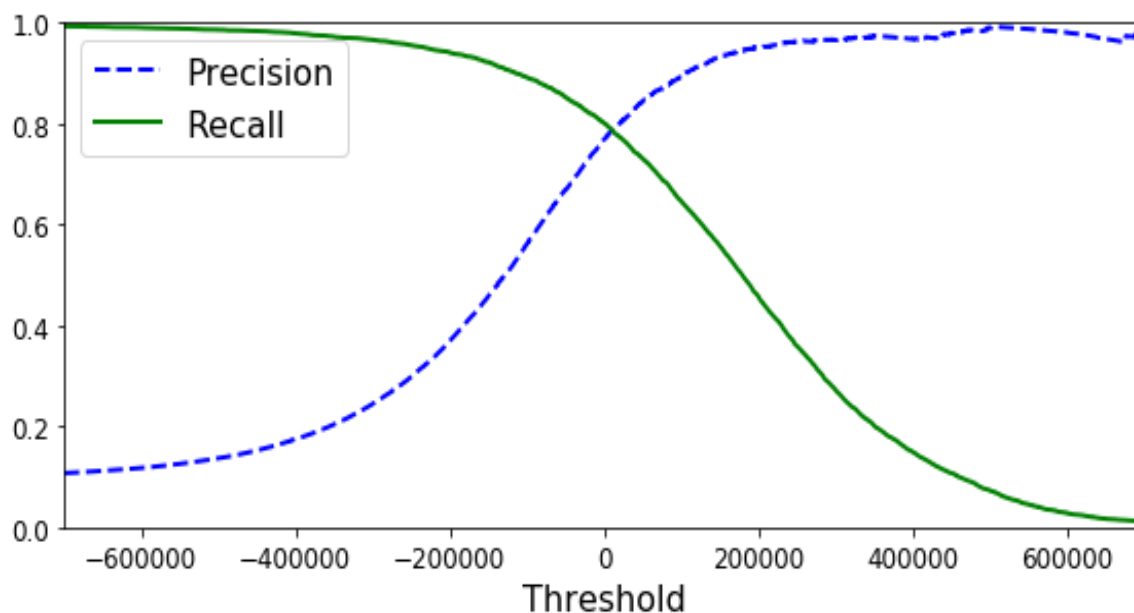
단순히 평균을 구하는 것이 아니라,

큰 값이 있다면 페널티를 주어 **작은 것에 가까운 평균**을 구함

Imbalanced data에서 **큰 값을 가지는 클래스에 대해 페널티를 줄 수 있음!**

## 왜 Precision과 Recall을 고려하는가?

✓ Precision과 Recall은 **Trade-off 관계**이므로 둘 다 최댓값을 가질 수 없기 때문!



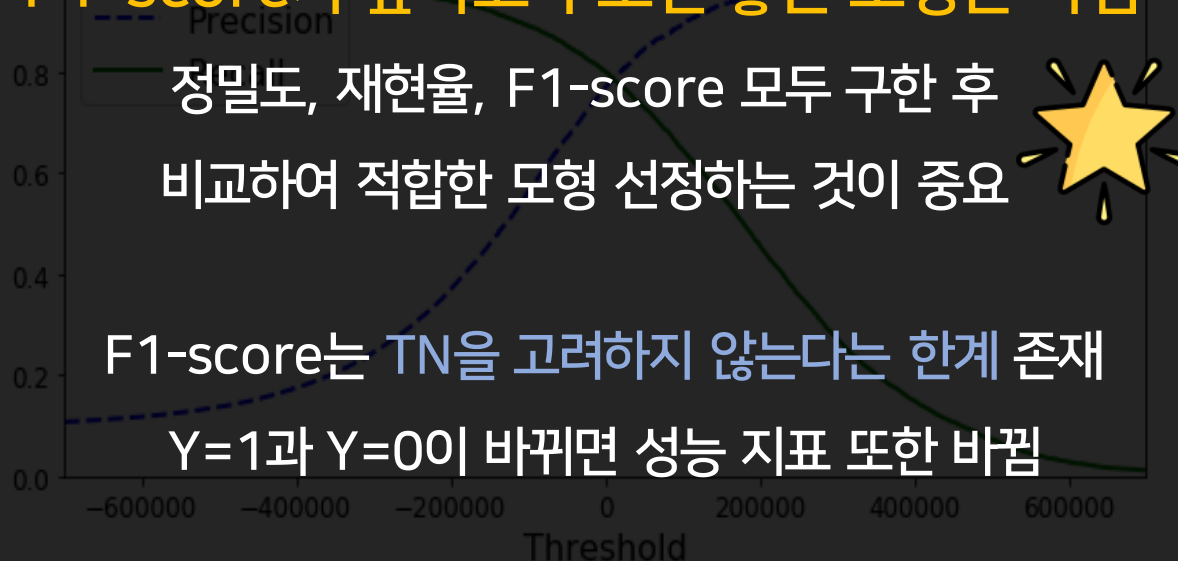
☞ Cut-off point (임계값)을 낮출수록 재현율은 올라가고 정밀도는 떨어짐

임계값을 0.5에서 0.4로 낮추면 그만큼 positive 예측을 너그럽게 해서  
양성을 음성으로 예측하는 횟수도 줄어들게 됨

## 왜 Precision과 Recall을 고려하는가?

✓ F1-score도 1에 가까울수록 좋지만,  
Precision과 Recall은 Trade-off 관계이므로 둘 다 최댓값을 가질 수 없기 때문!

**F1-score가 높다고 무조건 좋은 모형은 아님**



☞ Cut-off point (임계값)을 낮출수록 재현율은 올라가고 정밀도는 낮아진다. Case 예시에서 더 자세히 살펴보자!

임계값을 0.5에서 0.4로 낮추면 그만큼 positive 예측을 너그럽게 해서  
양성을 음성으로 예측하는 횟수도 줄어들게 됨



## MCC (Matthews Correlation Coefficient/매튜 상관계수)



$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

상관계수의 형식



		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FP
	$\hat{Y}=0$	FN	TN

- ✓ 1에 가까울수록 **완전 예측**
- ✓ -1에 가까울수록 **완전 역예측**
- ✓ 0에 가까울수록 **랜덤 예측**

## MCC (Matthews Correlation Coefficient/매튜 상관계수)



혼동행렬의 모든 부분(TN/TP/FN/FP)을 사용하여 만들어져서

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

가장 균형 잡힌 척도로 알려짐

Imbalanced data에서도 유용하게 사용할 수 있음

상관계수의 형식

이제 여러 상황의 예시를 통해 각 지표의 장단점과 MCC가 왜 좋은 지표인지 알아보자!

		관측값(y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	TP	FN
	$\hat{Y}=0$	FP	TN

- ✓ 1에 가까울수록 완전 예측
- ✓ -1에 가까울수록 완전 역예측
- ✓ 0에 가까울수록 랜덤 예측

## Case 1

밸런스가 깨져 있는 95:5의 imbalanced data에서  
단순히 모두  $Y = 1$  이라고 예측하는 모델을 적합했다고 가정




		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	95	5
	$\hat{Y}=0$	0	0

✓  $\text{Accuracy} = \frac{95}{100} = 95\%$

✓  $\text{F1-score} = \frac{2 \cdot 95}{2 \cdot 95 + 5 + 0} = 97.44\%$

모델의 성능이 높게 평가되었으니 이대로 사용해도 될까?

## Case 1



		관측값( $Y$ )	
		$Y=1$	$Y=0$
예측값( $\hat{Y}$ )	$\hat{Y}=1$	95	5
	$\hat{Y}=0$	0	0



하지만 MCC를 구해보면 분모에 0이 들어가기 때문에  
랜덤으로 예측하는 것과 다름 없음

이렇게 모든 결과 값을 하나로 올인해버리는 모델의 경우  
랜덤 예측이나 다를 바 없으므로 잘못되고 있음을 직감해야함...

## Case 2

이번엔 조금 현실적인 경우를 생각해보자.



95:5의 imbalanced data지만, 예측 모델이 조금 달라졌다.



		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	90	4
	$\hat{Y}=0$	5	1

$$\checkmark \text{Accuracy} = \frac{90+1}{100} = 91\%$$

$$\checkmark \text{F1-score} = \frac{2 \cdot 90}{2 \cdot 90 + 4 + 5} = 95.24\%$$



$$\text{MCC} = \frac{(90 \cdot 1) - (5 \cdot 4)}{\sqrt{(90 \cdot 5)(90 + 4)(1 + 5)(1 + 4)}} = 0.14$$

## Case 2



MCC가 0에 가까운 결론에 이르게 됨

이번엔 조금 현실적인 경우를 생각해보자.

95:5의 imbalanced data지만, 예측 모델이 조금 달라졌다.

F1-score는 TN을 고려하지 않는 반면,

MCC는 혼동행렬의 모든 항목에 영향을 받음

		실측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	90	4
	$\hat{Y}=0$	5	1

Negative와 Positive 모두 잘 작동하는 경우에만

MCC가 높은 결과가 나옴

✓ Accuracy =  $\frac{90+1}{100} = 91\%$

✓ F1-score =  $\frac{2 \cdot 90 + 4 + 5}{2 \cdot 90 + 4 + 5} = 95.24\%$

☞ MCC =  $\frac{(90 \cdot 1) - (5 \cdot 4)}{\sqrt{(90 \cdot 5)(90 + 4)(1 + 5)(1 + 4)}} = 0.14$

## Case 3

이번엔 Case2 예시에서

$Y = 1$  과  $Y = 0$  이 바뀐 경우를 생각해보자.




		관측값(Y)	
		Y=1	Y=0
예측값( $\hat{Y}$ )	$\hat{Y}=1$	1	5
	$\hat{Y}=0$	4	90

$$\checkmark F1\text{-score} = \frac{2 \cdot 1}{2 \cdot 1 + 4 + 5} = 18.18\%$$

☞ F1-score는 **TN을 활용하지 않기 때문에**

$Y = 1$ 과  $Y = 0$ 이 바뀌게 되면 성능 지표 또한 바뀌게 됨

## F1-score VS MCC

그렇다면 F1-score는 MCC보다 좋지 않은 지표일까?



**그렇지 않다!**

Type 1 error와 Type 2 error의 경중을 따지는 문제와 관련 있음  
실제로 분류 예측의 많은 경우 **FN으로 예측했을 때**와  
**FP로 예측했을 때**의 cost가 다름



일반적으로 F1-score는 비대칭 데이터를 효과적으로 평가하기 위해  
**적은 범주를 positive**로 두고 계산 *Ex) 암 예시*



# F1-score VS MCC

## 방세로 돌아가보자!



그렇다면 F1-score는 MCC보다 좋지 않은 지표일까?

트럭의 air pressure system의

failure를 예측하는 문제에서,

**고장이 아니라고 예측했는데 실제로 고장인 경우**

**훨씬 더 큰 페널티를 부여함**으로써,

cost metric을 모델 평가 지표로 삼았음



**즉, 상황에 따라 적절한 평가지표를 사용하는 것은 매우 중요함!**

적은 범주를 positive로 두고 계산함 Ex) 암 예시

# 2

ROC 곡선과 AUC

## Confusion Matrix의 한계

cutoff point에 의존적 

- ✓ cutoff point를 무엇에 정하는지에 따라  
각각 다른 confusion matrix 생성

정보의 손실 

- ✓ 연속형인 확률을 cutoff point를 기준으로  
두 범주로 나누는 과정에서 정보의 손실 발생



cutoff point

정보의 손실

int에 의존적

무엇에 정하는지에 따라

usion matrix 생성

**ROC curve로**

의 손실

t off point를 기준으로

정에서 정보의 손실 발생



ROC curve

**두 문제 모두 해결 가능!**

## ROC curve란?

모든 cut-off point에 대해 

TPR(민감도)와 FPR(1-특이도)를 나타낸 곡선

Confusion  
matrix의 한계 



ROC curve로 해결 

✓ cutoff point에 의존적



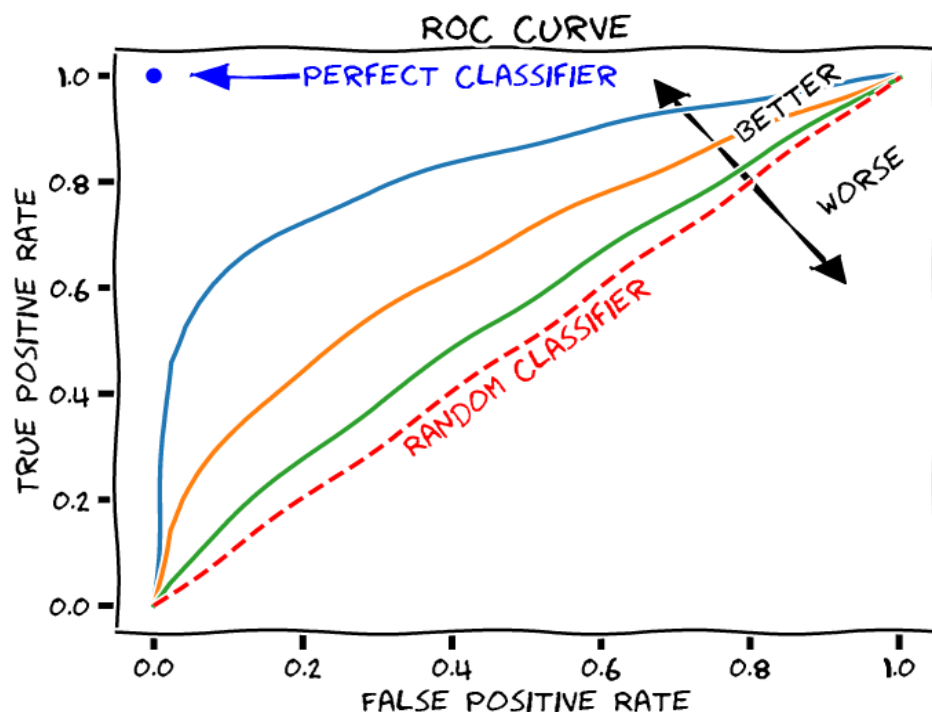
모든 cut-off point를 고려하여 의존하지 않는다.

✓ 정보의 손실



모든 예측 검정력을 구하기 때문에 더 많은 정보를 갖는다.

## ROC 곡선의 형태



우상향하는 위로 볼록한 곡선 

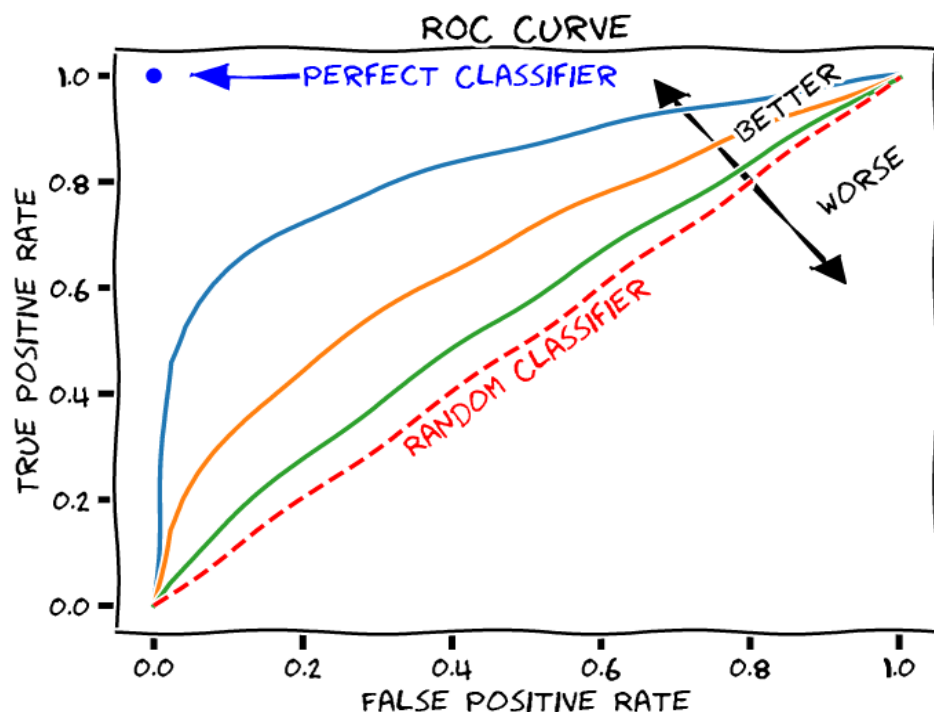
X축 : FPR(1-특이도) *잡인데 맞다고 하는 비율*

Y축 : TPR(민감도) *찐인데 맞다고 하는 비율*

X는 작을수록, Y는 클수록 좋음

X, Y 둘 다 0~1 사이

## ROC 곡선의 형태



Cutoff point가



0에 가까워질수록  $\rightarrow (1,1)$

1에 가까워질수록  $\rightarrow (0,0)$

왜? 왜? 왜?  
왜? 왜? 왜?  
왜? 왜? 왜?  
왜? 왜? 왜?

## Cutoff point가 0에 가까워질 때

기준점이 낮아짐

## ROC 곡선의 형태



대부분  
Y=1로 예측

웬만하면 다 맞다고 할테니까,,



TP & FP 증가  
TN & FN 감소  
숫아라 긍정의 힘!



FPR  $\approx 1$   
TPR  $\approx 1$

짹, 짹 구별없이 무조건  
맞다고 하는 비율 증가

0에 가까워질수록  $\rightarrow (1,1)$ 

## Cutoff point가 1에 가까워질 때

기준점이 높아짐



대부분  
Y=0로 예측  
무척이나 까다로워짐



TP & FP 감소  
TN & FN 증가  
아닌데 아닌데 아닌데



FPR  $\approx 0$   
TPR  $\approx 0$

짹, 짹 구별없이 무조건  
아니라고 하는 비율 증가

왜..?

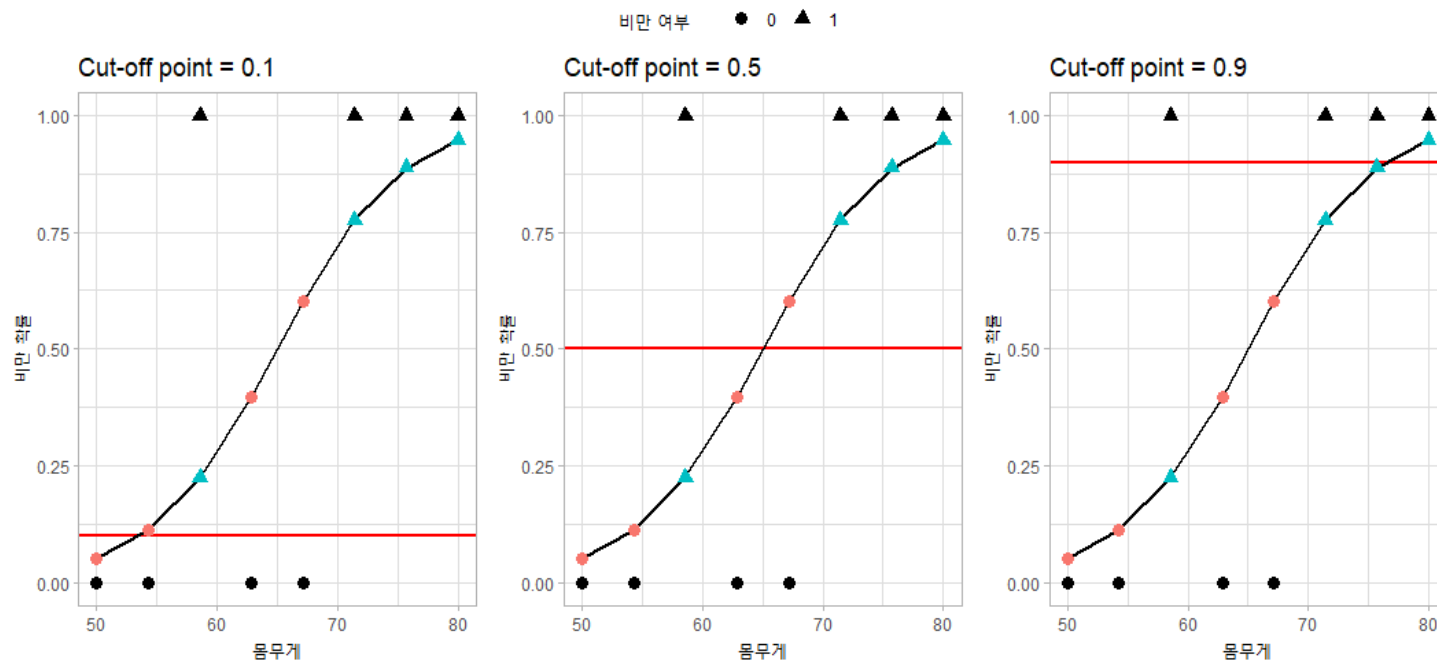




## ROC 곡선으로 적합한 Cutoff point 찾기

### ① 모든 Cutoff point에 대한 Confusion Matrix 생성

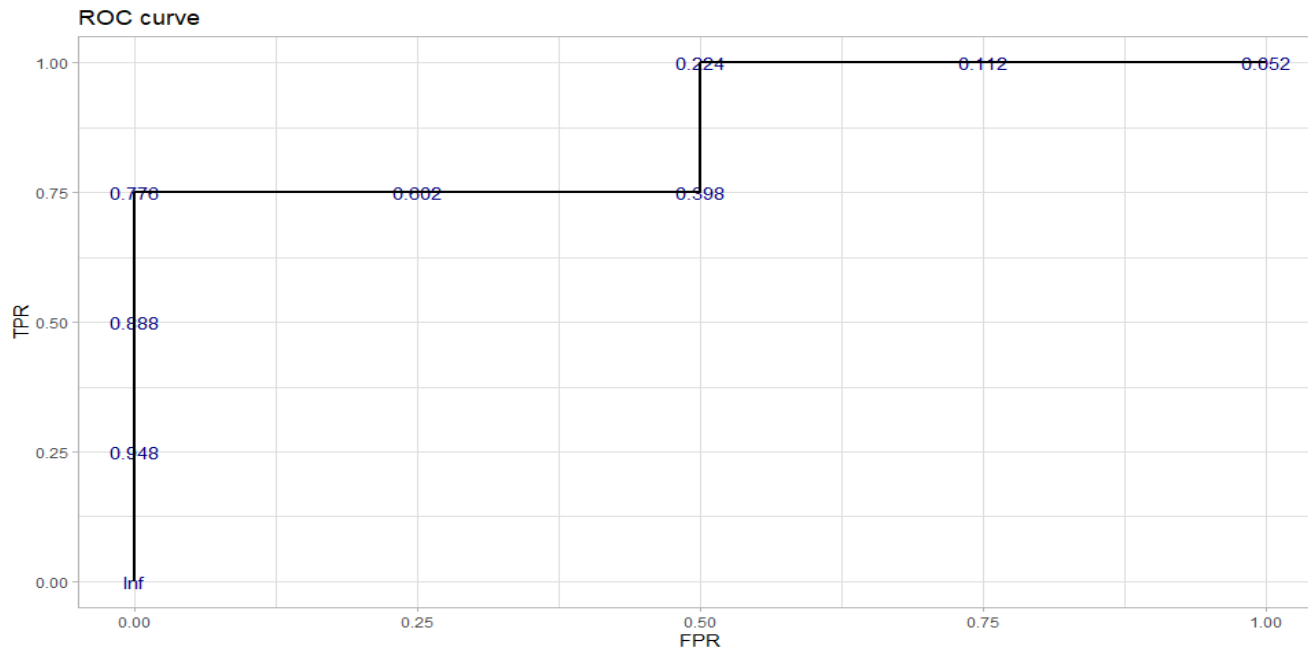
: Cut off point가 달라질 때마다 예측이 다르게 됨 -> 각각 다른 TPR & FPR값



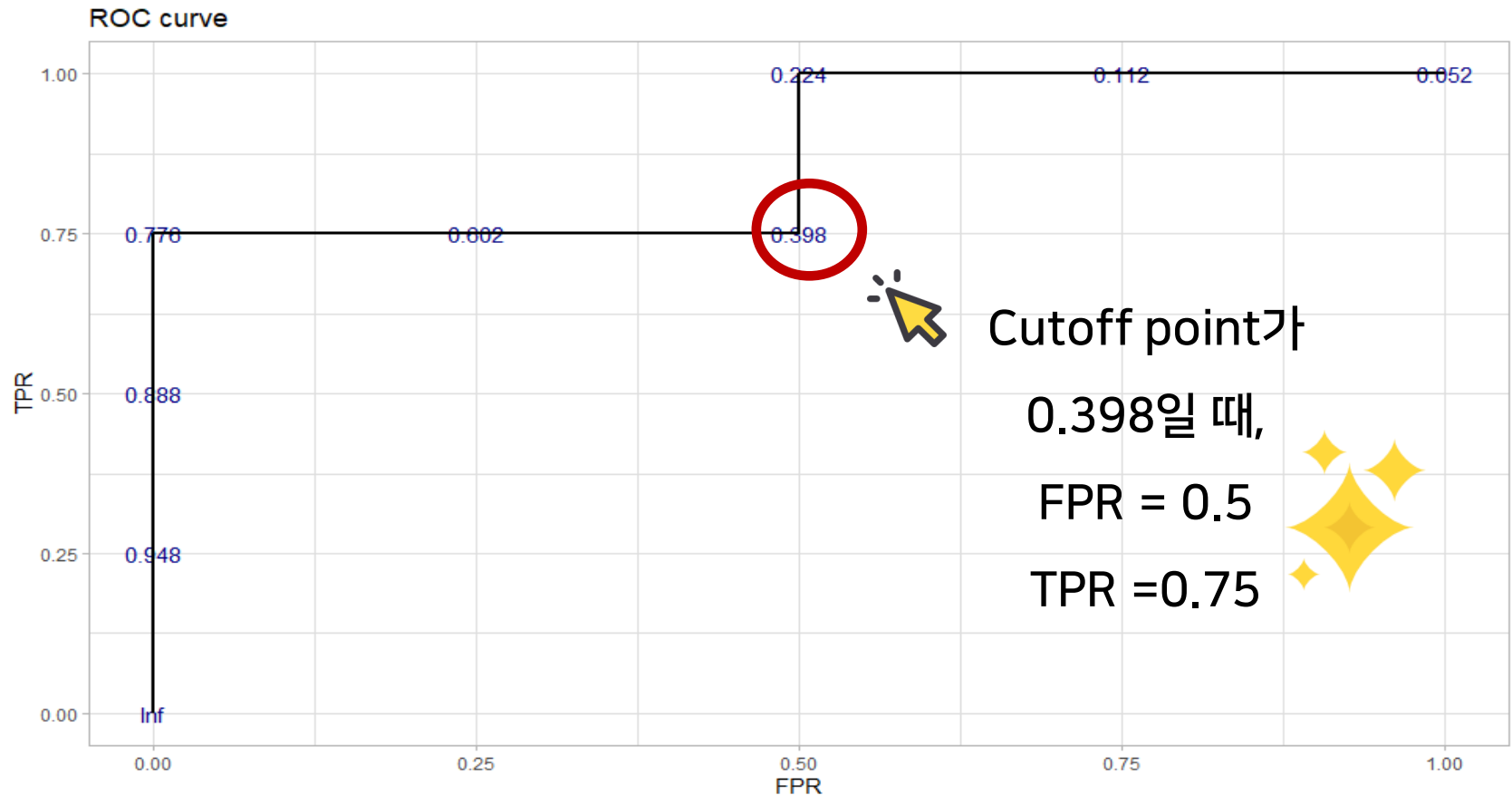
## ROC 곡선으로 적합한 Cutoff point 찾기

### ② 각각 다른 TPR & FPR값으로 ROC 곡선 그리기

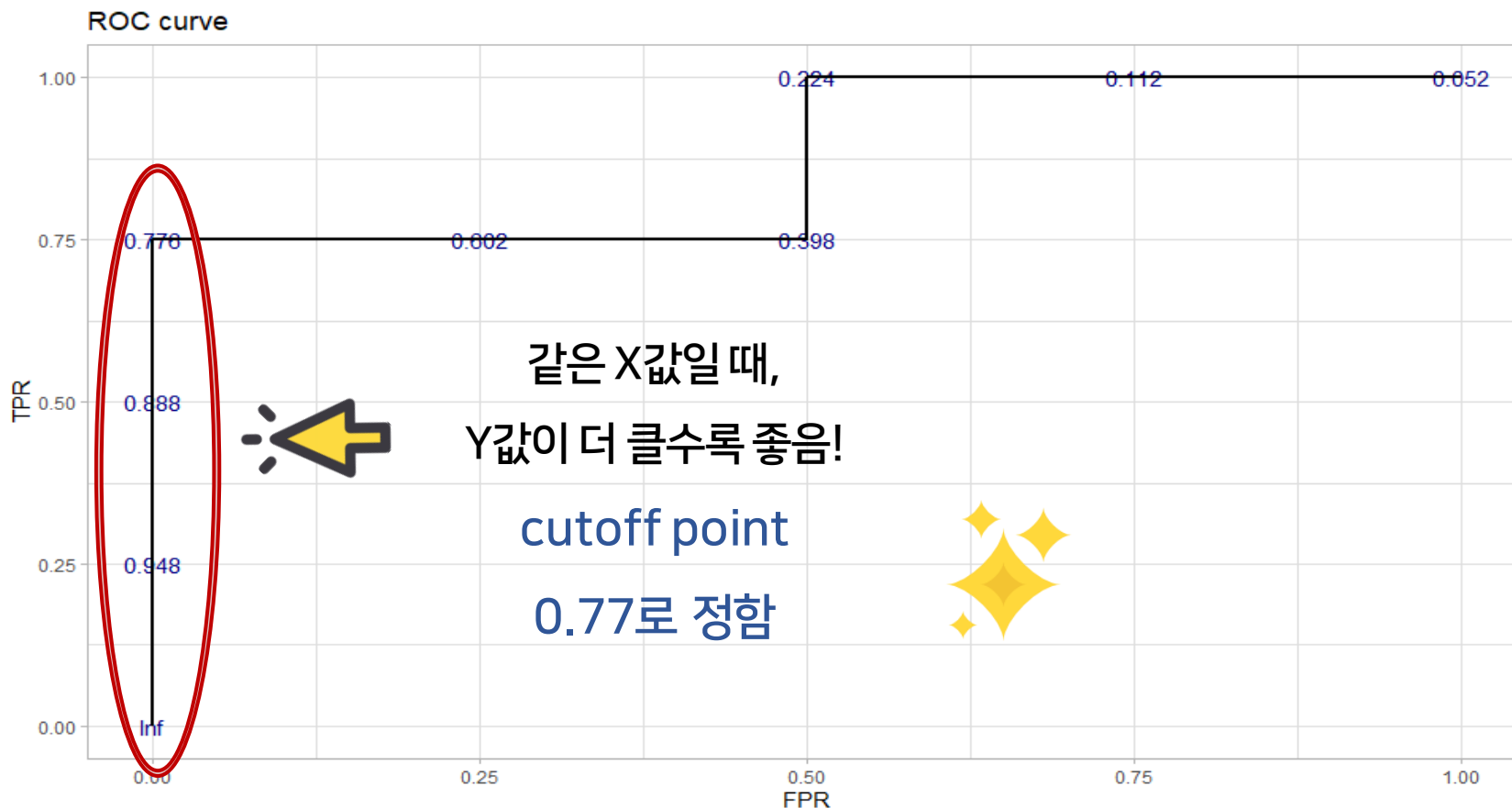
: 수없이 많은 TPR & FPR 점들을 이어 ROC 곡선 생성!



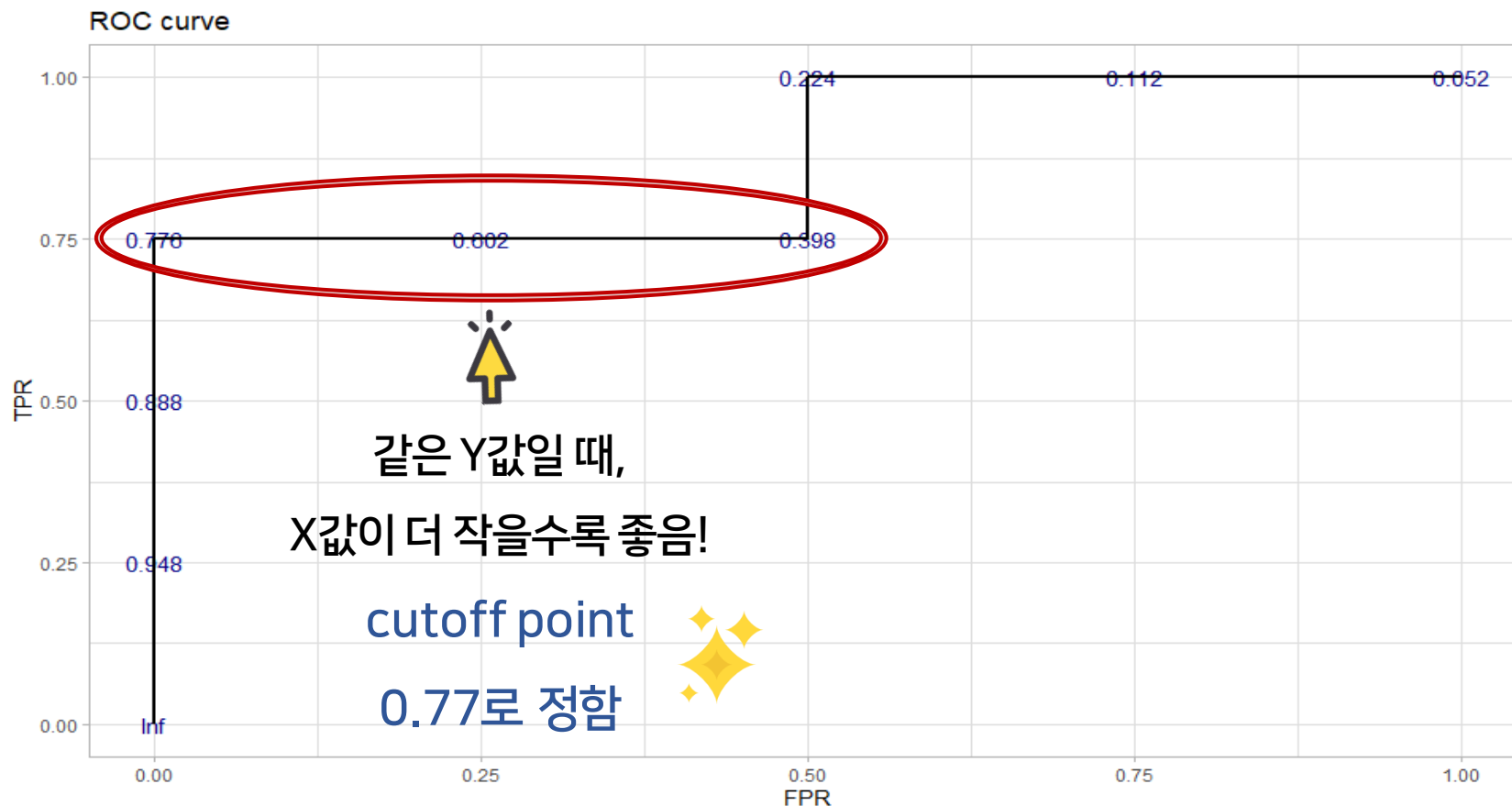
## ROC 곡선으로 적합한 Cutoff point 찾기



## ROC 곡선으로 적합한 Cutoff point 찾기



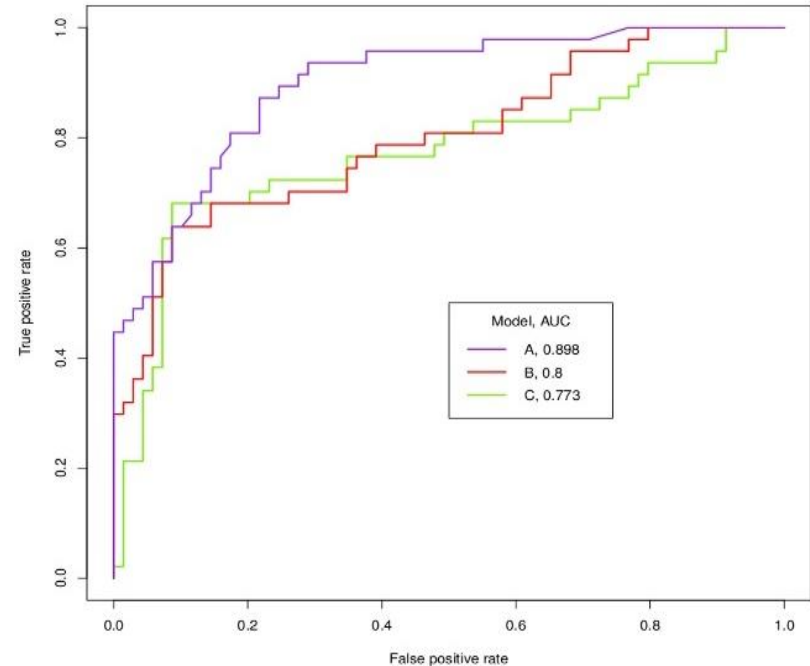
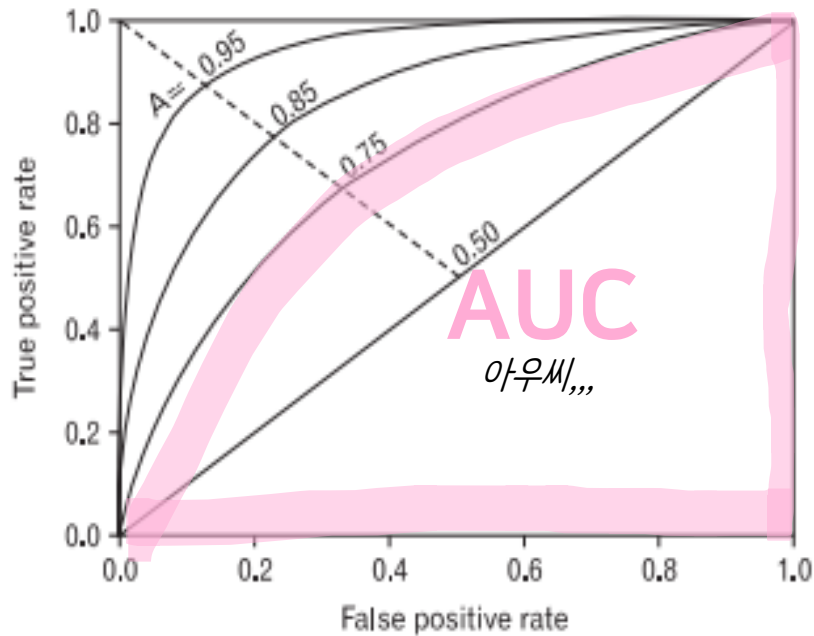
## ROC 곡선으로 적합한 Cutoff point 찾기



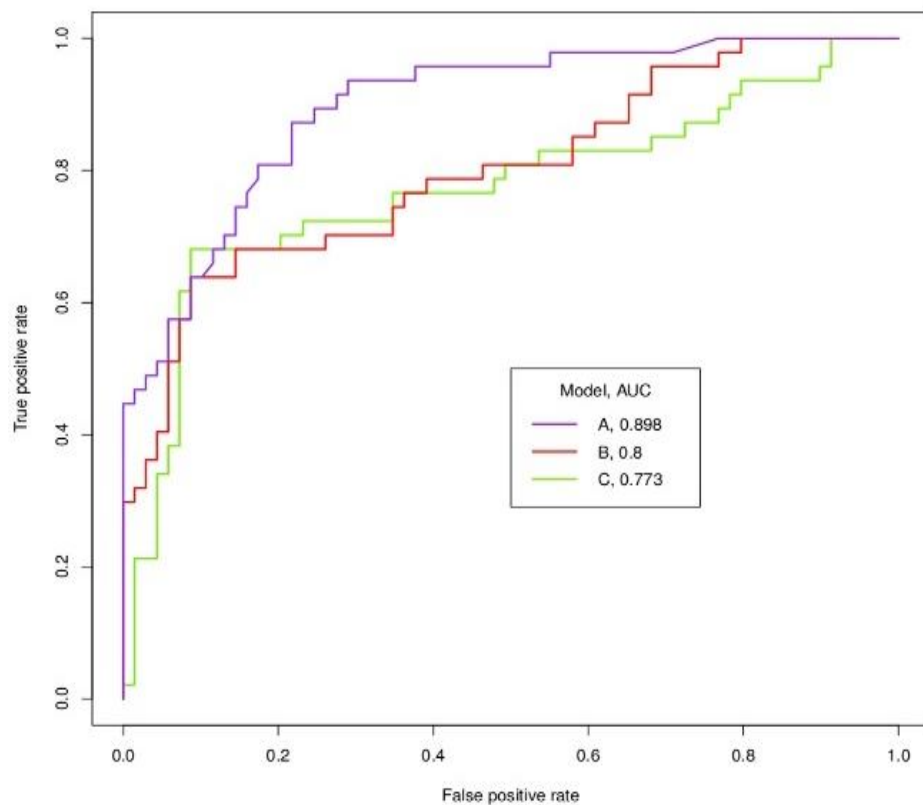
## AUC(Area Under the Curve)란?



ROC curve 밑의 면적



## AUC의 특징



AUC로 모델의 성능 비교 가능

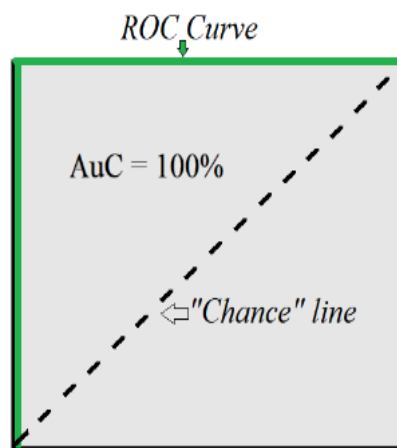
**AUC 값이 클수록 더 좋은 예측 모델!**

$$0 \leq \text{AUC} \leq 1$$

**즉, AUC가 1에 가까울수록 좋은 모델!**

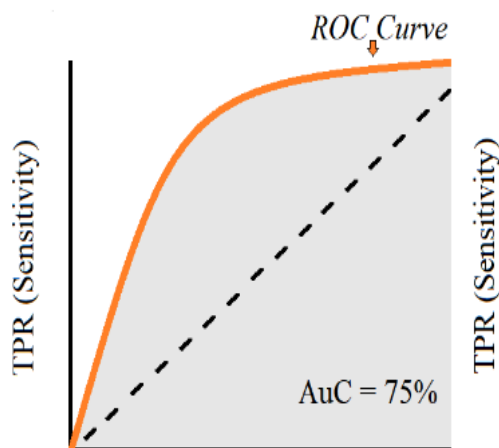
+ 한글 자판으로 하면 “몇” 이다

## AUC의 특징



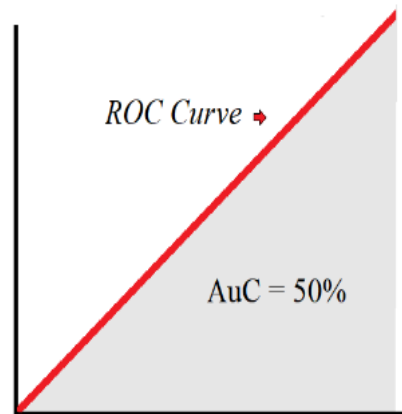
✓ AUC = 1

100% 완벽 예측  
(Overfitting 의심)



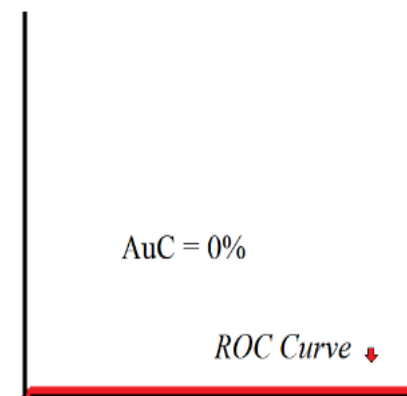
✓ AUC = 0.75

흠 나쁘지 않네



✓ AUC = 0.5

50% 예측  
강 찍은거나 다름없,,



✓ AUC = 0

100% 반대 예측  
시험문제 0점은  
사실 다 맞은거나 다름 없다,,



# 3

## Sampling

## 클래스 불균형(Class Imbalance)

클래스 

: 범주형 반응변수의 수준(level)

클래스 불균형 

: 각 클래스가 갖고 있는 데이터의 양에 큰 차이가 있는 경우  
즉, 데이터 양이 비대칭인 경우!

Ex) 질병이 있는 사람은 질병이 없는 사람에 비해 현저히 적다

## 클래스 불균형(Class Imbalance)

### 클래스 불균형

단순히 우세한 클래스를 택하는 모형의 정확도가 높게 나타남

→ 모델의 성능 판별 불가

소수 클래스의 재현율(TPR)이 낮아짐



주어진 데이터가 불균형이네..

그치만.. 난 클래스 균형이 필요하단 말이야  $\pi\_ \pi$

## 클래스 불균형(Class Imbalance)

클래스 균형



소수의 클래스에 특별히 더 관심이 있는 경우에 필요함  
Sampling을 통해 클래스 불균형을 해소할 수 있음



Garbage In! Garbage Out!

좋은 모델을 만들려면 좋은 train set이 필요하다!

**Sampling**을 통해 비대칭 문제를 해결하자!

## Sampling의 종류

언더 샘플링  
(Under Sampling)



랜덤 언더 샘플링  
(Random Under Sampling)

오버 샘플링  
(Over Sampling)



랜덤 오버 샘플링  
(Random Over Sampling)

SMOTE

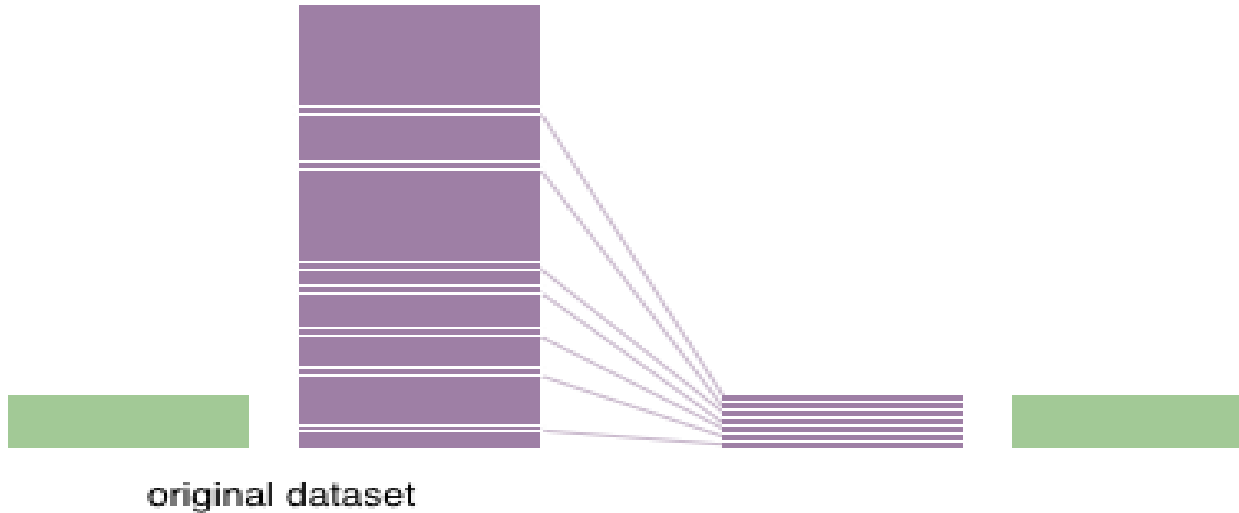
*MSMOTE (Modified SMOTE)*

## 언더 샘플링(Under Sampling)

: 다수 클래스의 데이터를 소수 클래스에 맞추어 감소



under-sampling



## 언더 샘플링(Under Sampling)

장점



메모리 사용, 처리 속도 측면에서 유리

단점



관측치의 손실 → 정보의 누락 문제 발생



## Random Under Sampling



랜덤으로 다수의 샘플 수 줄이는 방법

만약 샘플링으로 얻은 표본이 대표성을 띠지 못하면

부정확한 결과 초래

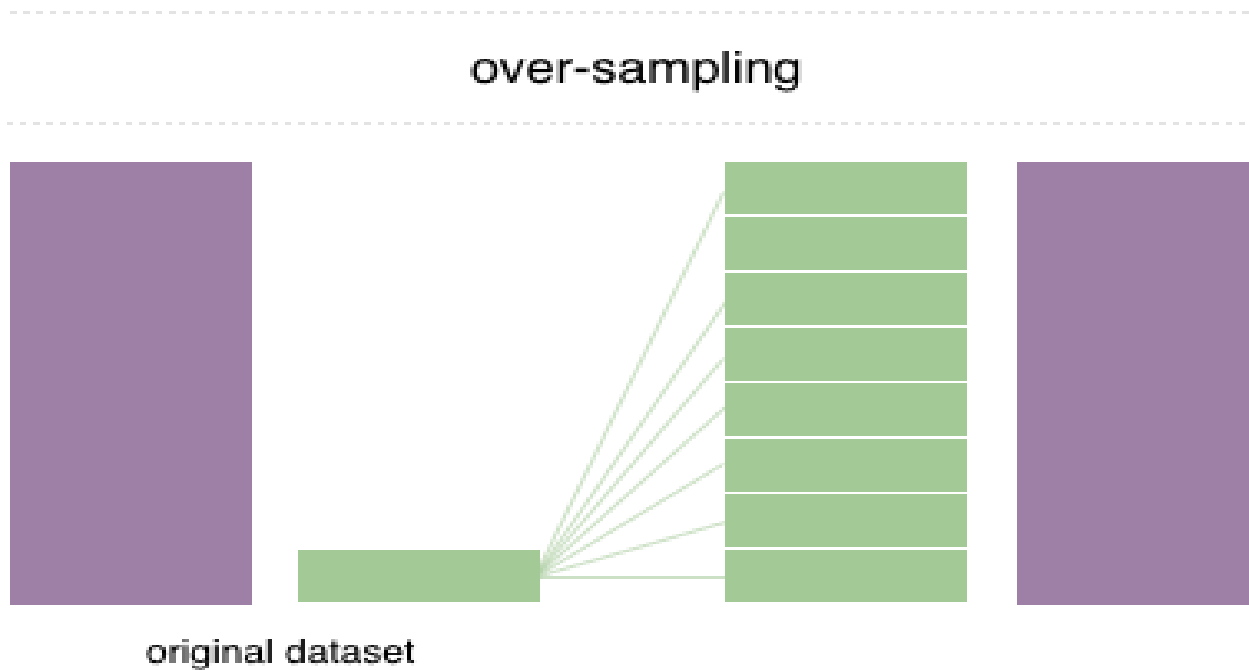


정보의 누락 문제 때문에  
보통 오버 샘플링 사용



## 오버 샘플링(Over Sampling)

: 소수 클래스의 데이터를 다수 클래스에 맞추어 증가





## 오버 샘플링(Over Sampling)

장점



정보 손실이 없음

단점



데이터 수가 늘어나서 메모리 사용,  
처리속도 측면에서 불리

## Random Over Sampling



무작위로 소수 클래스의 데이터 복제  
데이터 복제하기 때문에 **Overfitting** 가능성



실제로  
**SMOTE** 많이 사용!

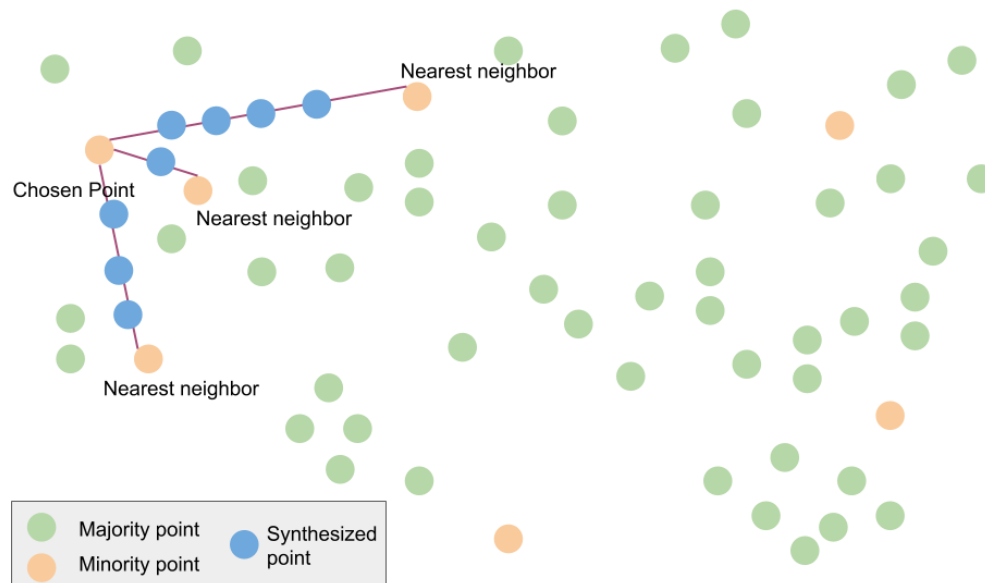


## 오버 샘플링(Over Sampling)의 종류

SMOTE 

Synthetic Minority Over Sampling Technique

가상의 데이터를 생성함으로써 소수 클래스의 데이터를 늘림



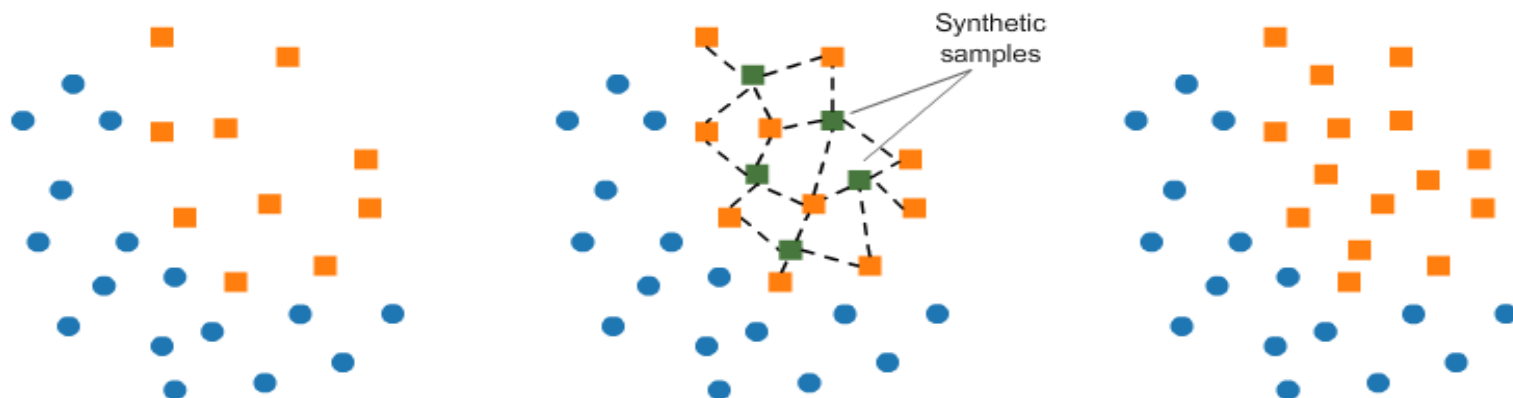
## SMOTE



알고리즘



1. 소수 클래스의 데이터 중 하나를 선택
2. 선택한 데이터와 가장 가까운 소수 클래스의 데이터 중에서 무작위로  $k$ 개의 데이터 선택
3. 선택한 데이터와  $k$ 개의 데이터 사이의 직선 상에 가상의 소수 클래스 데이터 생성



## SMOTE

장점



가상의 데이터를 생성하기 때문에,  
Overfitting 발생 가능성 감소



단점



다수 클래스 데이터의 위치 고려하지 않음  
서로 다른 클래스의 데이터끼리 **겹칠 수 있음**  
고차원 데이터에서 비효율적

*짱구도 좋아하는 SMOTE*

4

Encoding

## 인코딩(encoding)이란?

컴퓨터 공학 관점 

문자나 기호들의 집합을  
컴퓨터에서 표현하는 방법  
↓  
컴퓨터가 이용할 수 있는  
신호로 만드는 것

데이터 분석 관점 

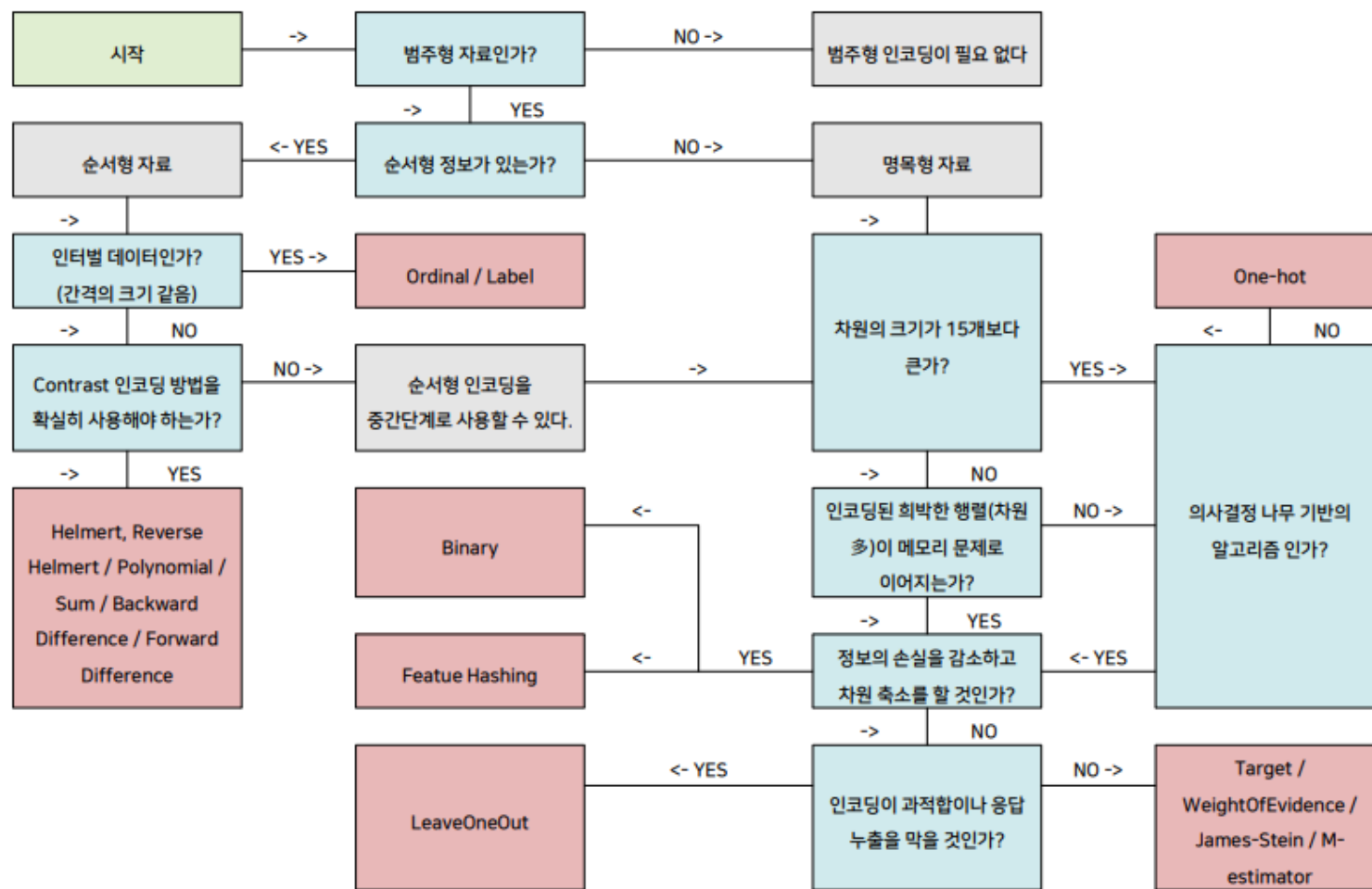
범주형 자료를 컴퓨터가  
읽을 수 있도록 수치화한 것  
↓  
**범주형 인코딩 방법**

필요성 

범주형 자료를 수치화하는 인코딩 방식을 거치면

**수치형 변수만을 설명변수**로 갖는 다양한 분석기법(ex.회귀 계열의 모형들) 사용 가능

## 인코딩 알고리즘



## 인코딩의 종류

Classic	Contrast	Bayesian	기타
Ordinal	Simple	Mean(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	



## One-Hot Encoding(Dummy Encoding)

가변수(dummy variable)를 만들어주는 인코딩 방법

펜트하우스	오윤희	천서진	주단태	나애교
오윤희	1	0	0	0
천서진	0	1	0	0
주단태	0	0	1	0
나애교	0	0	0	1



해당 범주에는 1, 그 외에는 0 입력

## One-Hot Encoding(Dummy Encoding)

가변수(dummy variable)를 만들어주는 인코딩 방법

펜트하우스	오윤희	천서진	주단태	나애교
오윤희	1	0	0	0
천서진	0	1	0	0
주단태	0	0	1	0
나애교	0	0	0	1



기준이 되는 범주의 열 삭제

J-1개의 더미 변수로 J개의 level을 갖는 인자들 충분히 설명 가능

모든 더미 변수가 0이라면 기준 범주일 경우를 뜻하기 때문

## One-hot Encoding (Dummy Encoding)



가변수(dummy variable)를 만들어주는 인코딩 방법

**Tree 기반 모델을 사용할 경우?**

기준 범주를 지우지 않고 N개의 가변수를 생성해야 함	편지	편지	편지	편지
오윤희	1	0	0	0
천서진	0	1	0	0
주단태	0	0	1	0
나예교	0	0	0	1

만약 삭제된 기준 범주가 트리를 생성하는데  
매우 중요한 요소였다면, **트리 모델이 잘못 학습** 될 수 있음



기준이 되는 범주의 열 삭제

J-1개의 더미 변수로 J개의 level을 갖는 인자들 충분히 설명 가능

모든 더미 변수가 0이라면 기준 범주일 경우를 뜻하기 때문

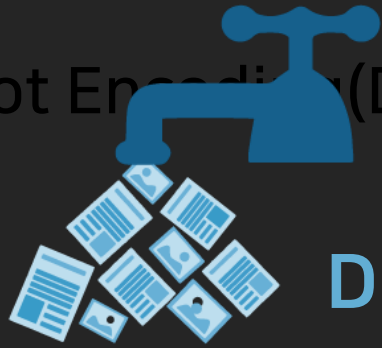
## One-Hot Encoding(Dummy Encoding)

### 원-핫 인코딩의 장점

1. 해석이 용이하다.
2. 명목형 변수 값들을 가장 잘 반영한다.
3. 다중공선성을 해결한다.  
해당 범주만 1이고 나머지는 0으로 표현되기 때문에,  
한 변수가 다른 변수들로 설명되는 것을 방지해줌.
4. 지도학습의 경우 Data Leakage가 발생하지 않는다.



## One-Hot Encoding (Dummy Encoding)



원-핫 인코딩의 장점  
**Data Leakage**란?



1. 해석이 용이하다.  
정답이 존재하는 지도학습에서 **반응변수 Y**에 대한 정보가  
2. 명목형 변수 값들을 가장 잘 반영한다.  
모델 학습 시 **설명변수 X**에 들어가는 것

3. 다중공선성을 해결한다.

해당 범주만 1이고 나머지는 0으로 표현되기 때문에,  
∴ **overfitting** 발생 가능  
한 변수가 다른 변수들로 설명되는 것을 방지해줌.

4. 지도학습의 경우 Data Leakage가 발생하지 않는다.



## One-Hot Encoding(Dummy Encoding)

원-핫 인코딩의 단점 

범주형 변수의 Level/  
범주형 변수가 많음



많은 가변수 생성



차원이 늘어남



학습속도가 느려지고  
상당한 computer power 요구

## Label Encoding

각 범주를 나누어 주기 위해 단순히 점수를 할당하는 인코딩 방법  
(명목형 자료에 많이 사용)

펜트하우스	점수
오윤희	1
천서진	2
주단태	3
나애교	4

요리보고 저리봐도 알 수 없는  
인코딩 인코딩~



할당된 점수의 숫자에는 어떠한 순서나 연관성이 없음



## Label Encoding

장점



✓  
원-핫 인코딩과 다르게  
차원이 늘어나지 않는다

단점



라벨 간의 순서나 연관성이 존재한다고  
학습되어 정보 왜곡의 가능성 생김  
✓




## Ordinal Encoding

순서형 정보에 대응하는 점수를 할당하는 인코딩 방법

매움 정도	점수
	1
	2
	3
	4

<고추 “매움정도” 표시방법>

구 분				
매움 정도	맵지 않음	약간 매움	<b>보통 매움</b>	매우 매움
캡사이신 함량 (ppm)	100 미만	100~800	800~2,000	2,000이상

농산물 표준규격기준 고추 맵기 표시

☞ 점수를 할당할 때는 보통 1부터 부여  
할당된 점수들은 순서나 연관성이 있음

## Ordinal Encoding

장점



차원이 늘어나지 않기에 빠른 학습 가능  
순서형 정보 표현 가능

단점



순서형 정보가 있을 때만 사용 가능  
Level 간의 차이가 어느 정도인지를  
고려하기 쉽지 않음

Ex) 보통매움과 매우매움의 정도의 차이가  
1일 수도 있고, 3일 수도 있음

## Mean Encoding(Target Encoding)

범주형 변수의 각 수준에 대하여

반응변수(타겟변수)Y의 평균으로 점수를 할당하는 인코딩 방법

[Y] 토익 점수	[X] 팀	Target Encoding
780	선대	855
930	선대	855
850	범주	820
870	범주	820
810	범주	820
750	범주	820
660	데마	863.33
980	데마	863.33
950	데마	863.33

$$\frac{850+870+810+750}{4}$$

=820 (범주팀 토익 평균)

실제로 맞는지 아닌지 모름 ...  
물어보지 마셈 ...

## Mean Encoding(Target Encoding)

장점



✓  
차원이 늘어나지 않음  
할당된 점수에 **당위성** 존재  
(≠Label Encoding)


단점



- ✓
- Data Leakage가 일어나기 때문에 overfitting 될 가능성이 높음
  - Training set에 없던 **새로운 범주**가 Test set에 **등장**하면, 점수를 할당할 수 없게 되는 문제 발생

## Leave One Out Encoding(LOO Encoding)

현재 행을 제외하고 평균을 구한 뒤 이를 점수로 할당하는 인코딩 방법



[Y] 합격	[X] 팀	Target Encoding	LOO Encoding
1	선대	66.7%	50%
1	선대	66.7%	50%
0	선대	66.7%	100%
1	범주	50%	33.33%
1	범주	50%	33.33%
0	범주	50%	66.7%
0	범주	50%	66.7%
0	데마	33.3%	50%
0	데마	33.3%	50%
1	데마	33.3%	0%

## Leave One Out Encoding(LOO Encoding)

### 장점



- ✓ 차원이 늘어나지 않음
- ✓ Outlier의 영향을 적게 받음
- ✓ **Direct** Data Leakage를 방지할 수 있음
- ✓ 현재 행을 제외하고 계산했기 때문!

### 단점



- ✓ Data Leakage가 일어나기 때문에 overfitting 될 가능성이 높음
- ✓ Training set에 없던 **새로운 범주**가 Test set에 **등장**하면, 점수를 매길 수 없음

## Ordered Target Encoding(CatBoost Encoding)

현재 행 이전의 값들을 사용하여 평균을 구하고  
이를 점수로 할당하는 인코딩 방법

- ✓ Target Encoding(Mean Encoding)과 매우 유사  
Ordered Target Encoding은 같은 범주라도 다른 점수 할당 가능
- ✓ CatBoost(부스팅 모델 중 하나)에서 사용되는 인코딩 방식  
장단점은 L00 Encoding과 동일



## Ordered Target Encoding(CatBoost Encoding)

[Y] 합격	[X] 팀	Target Encoding	LOO Encoding
1	선대	66.7%	100%
1	선대	66.7%	100%
0	선대	66.7%	100%
1	범주	50%	100%
1	범주	50%	100%
0	범주	50%	100%
0	범주	50%	66.67%
0	데마	33.3%	0%
0	데마	33.3%	0%
1	데마	33.3%	0%

[1|1|0]

$$\frac{2}{3} = 0.6667$$

각 수준에서 첫번째 행은 그냥 해당 행만 가지고 평균을 구함



# 5

부록

## CatBoost

범주형 변수가 많을 때 사용하기 좋은 부스팅 모델

1. 자동으로 CatBoost Encoding을 해주기 때문에 편리함

2. CatBoost의 `max_ctr_complexity` 파라미터를 통해 categorical feature combinations를 모델 내에서 진행

Country	Hair color	Class_label
India	Black	1
India	Black	1
Russia	White	0
Russia	White	0

Country만 알면 Hair color는 알 필요 X

알아서 상관관계가 높은 변수끼리 묶음  
(`max_ctr_complexity`를 늘릴수록 모델링 속도는 느려짐)

## 우리 지연이는 이런 것도 해줍니다 ... ^^ (너네 팀은 지연이 없지? ㅎㅎ)

### 5. 맺음말 | 3주의 클린업을 달려오신 짱 멋진 범주팀에게 :)

안녕하세요. 짱 웃기고 짱 귀여운 범주팀 여러분 ^^ 범주팀장이 되고부터 클린업 동안 여러분들에게 부족하지 않은 지식을 전달해드리고 싶어서 방학동안 걱정도 많이 하고 준비도 많이 한다고 했는데 이렇게 열심히 공부하고 잘 따라와주어서 정말 감사할 따름이에요... ‘범주형자료분석’이라는 과목이 통계학과라면 꼭 배워야 하는 과목 중 하나임에도 불구하고 무엇을 배우는지 제대로 알지 못하고 지나치는 경우가 많은 것 같아요. 데이터를 다루는데 있어서 머신러닝이나 딥러닝처럼 말그대로 fancy한 방법들이 넘쳐나는 시기이지만, 통계학과라면 적어도 가장 근본적인 통계 지식들을 알고 있어야 한다고 생각해서 범주를 선택했습니다. 교안을 쓰면서 제대로 이해하지 못하고 지나쳤던 지식들을 하나하나 다시 공부할 수 있어서 저로서도 정말 배우는 게 많았던 3주였습니다.

특히 범주는 회귀분석을 재미있게 들은 사람이라면 더 쉽게 받아들일 수 밖에 없는 과목이라고 생각해요. 첫 스터디 날 직관적이면서 매력적인 과목이라고 말했던 이유는 통계학원론과 수리통계학에서 배운 확률분포, 추정과 가설검정의 원리들을 적용시킴으로써 회귀분석에서 배운 내용을 더 확장시키는 과목이라고 느껴졌거든요. 여러분들도 배우면서 그런 점들을 느낄 수 있도록 교안을 만들고 싶었는데, 잘 전달되었을까요? 1,2주차에는 분할표와 독립성 검정, 로지스틱 회귀를 포함한 GLM 등 CDA의 기본에 대해 배웠다면 3주차에는 여러분들이 분석에 활용할 수 있는 정말 실용적인 정보들을 담았어요. 특히 분류 모델의 평가지표들에 대해 여러분들이 잘 알아 두신다면, 앞으로 분류 예측을 할 때 정말 수월할 것이라고 생각해요... 좋은 모델에는 정답이 없다고 하죠. 어떤 평가지표를 사용해서 모델을 평가할 것인지도 결국 분석자가 정해야 하는 문제인데요. 결국 분석자가 주어진 문제상황에 대해 이해하는 것(도메인 지식), 어떤 방법이 최선인지 고민하는 것(분석자의 주관)이 분석에서 가장 중요할 수 밖에 없는 것 같습니다.

여러분들이 앞으로 실제 데이터를 다루고 분석할 때 도움이 되는 내용들로 구성된 교안을 만들고 싶었어요. 방학동안 오산시 어린이 교통사고 위험지역 선정이라는 주제의 공모전을 준비하면서, 교통사고를 예측하는 모델을 만들고자 했습니다. 교통사고 수 데이터는 count data라서 GLM을 써야 하는데, 문제는 데이터가 ‘10년치’의 ‘어린이’ 교통사고 수였기 때문에 대부분이 0인 데이터였던 거예요. 그래서 이름만 알았던 ZIP 회귀모형을 써보자 했는데, 제대로 모델의 원리를 이해하지 않고 코드만 구글링해서 쓰다 보니 오류 해결이 어려워져 결국 포기했던 경험이 있어요. 그래서 모델을 사용할 때 어떤 경우에 사용하고 어떤 원리로 fitting하는지를 아는 것이 정말 중요하다는 생각을 했습니다. 이번 교안을 쓸 때도 이런 것들을 최대한 담으려고 했구요. 주제분석이나 또는 그 이후에 여러분들이 하시게 될 공모전이나 다른 데이터 분석 프로젝트에서도 이번 클린업에서 배운 지식들이 도움이 된다면 참 뿌듯할 것 같아요.

3주 클린업 시간이 정말 빨리 지나간 것 같아요. 하루에 짧은 시간이라도 매일 무엇을 하면 고수가 된다고 하던데 학회를 하는 동안 통계공부도, 코딩 실력도, 분석에 도움되는 지식들도 하다못해 PPT 만드는 스킬이나 발표 같은 것들도 많이 배웠던 것 같아요. 매주 교안 공부하고 피피티 만들고 패키지 과제까지 하면서 바쁜 하루하루 보내셨을 텐데, 그만큼 고수가 되셨을겁니다... 여러분들은 저보다 웃기고 멋진 사람들이니까 더 많이 배워 가실 거라고 믿어요. 거리두기 단계로 다섯명이 다같이 밥 한 번 아직 못 먹어봤는데, 그럼에도 이렇게 어려운 상황 속에서 열심히 해 주셔서 감사합니다. 범주팀과 만날 때마다 즐거운 에너지가 생겨서 정말 행복해요. 중간고사 잘 보시고 재밌고 멋진 주제 정해서 4,5월의 주제분석도 파이팅해봅시다! :)



**THANK YOU**

