범주형자료분석팀

2팀
김보현
송승현
최용원
김동희
오주원

INDEX

1

혼동행렬

혼동행렬

혼동행렬(Confusion Matrix)

분류 모델의 성능을 평가하기 위한 지표

예측값($\hat{Y}$)과 **실제값**(Y) 을 비교

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

T(True)/ F(False) : 실제 값과 예측 값의 일치 여부

P(Positive)/ N(Negative) : 모델의 긍정 혹은 부정 예측 여부

혼동행렬

TP (True Positive)

긍정($\hat{Y} = 1$)으로 예측하였으며 실제 관측값도 긍정($Y = 1$)인 경우

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

혼동행렬

TN (True Negative)

부정($\hat{Y} = 0$)으로 **예측**하였으며 **실제 관측값**도 **부정**($Y = 0$)인 경우

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

혼동행렬

FP (False Positive)

긍정($\hat{Y} = 1$)으로 **예측**하였으나 **실제 관측값**은 **부정**($Y = 0$)인 경우

1종 오류와 동일!

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

혼동행렬

FN (False Negative)

부정($\hat{Y} = 0$)으로 **예측**하였으나 **실제 관측값**은 **긍정**($Y = 1$)인 경우

2종 오류와 동일!

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

분류평가지표

정확도 (Accuracy)

전체 경우에서 **예측값**과 **실제값**이 **일치**하는 비율

$$Accuracy = \frac{TP + FN}{TP + FP + FN + TN}$$

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

분류평가지표

정확도 (Accuracy)

전체 경우에서 **예측값**과 **실제값**이 **일치**하는 비율

$$Accuracy = \frac{TP + FN}{TP + FP + FN + TN}$$

1에 가까울수록 좋은 성능, 직관적으로 이해하기 쉽고 단순

불균형 데이터(Imbalanced Data)에서는 관측치가 많은 수준(Class)에 의존해

정확한 성능 파악 불가

분류평가지표

정밀도 (Precision)

긍정($\hat{Y} = 1$)으로 **예측**한 값들 중 **실제 관측치 역시 긍정**($Y = 1$)인 비율

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

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

분류평가지표

정밀도 (Precision)

긍정($\hat{Y} = 1$)으로 **예측**한 값들 중 **실제 관측치 역시 긍정**($Y = 1$)인 비율

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

1에 가까울수록 좋은 성능

FP가 치명적일 때 주로 사용

ex) 유죄($Y = 1$)를 무죄로 선고($\hat{Y} = 0$)하는 것보다

무죄($Y = 0$)를 유죄로 선고($\hat{Y} = 1$)하는 것이 더 위험

분류평가지표

민감도 (Sensitivity) / 재현율 (Recall)

실제 긍정($Y = 1$)인 관측값 중 긍정($\hat{Y} = 1$)으로 예측한 비율

$$Sensitivity(Recall) = \frac{TP}{TP + FN}$$

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

분류평가지표

민감도 (Sensitivity) / 재현율 (Recall)

실제 긍정$(Y = 1)$인 관측값 중 **긍정**$(\hat{Y} = 1)$으로 **예측**한 비율

$$Sensitivity(Recall) = \frac{TP}{TP + FN}$$

1에 가까울수록 좋은 성능, ROC 곡선의 Y축

FN이 치명적일 때 주로 사용

ex) 암세포가 없는 사람$(Y = 0)$에게 암이라고 진단$(\hat{Y} = 1)$하는 것보다

실제로 암세포가 있는 사람$(Y = 1)$에게 건강하다$(\hat{Y} = 0)$고 진단하는 것이 훨씬 위험

분류평가지표

특이도 (Specificity)

실제 부정$(Y = 0)$인 관측값 중 **부정**$(\hat{Y} = 0)$으로 **예측**한 비율

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

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

분류평가지표

특이도 (Specificity)

실제 부정($Y = 0$)인 관측값 중 부정($\hat{Y} = 0$)으로 예측한 비율

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

1에 가까울수록 좋은 성능

분류평가지표

FPR(False Positive Rate)

실제 부정$(Y = 0)$인 관측값 중 **긍정**$(\hat{Y} = 1)$으로 **예측**한 비율

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

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

분류평가지표

FPR(False Positive Rate)

실제 부정$(Y = 0)$인 관측값 중 **긍정**$(\hat{Y} = 1)$으로 **예측**한 비율

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

잘못 예측한 비율이므로!

0에 가까울수록 좋은 성능

ROC 곡선의 X축

분류평가지표

F1-Score

정밀도(Precision)와 **재현율**(Recall, Sensitivity)의 **조화평균**

$$F1\ score$$

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

1에 가까울 수록 **모델의 성능**이 좋다고 판단

분류평가지표

F1-Score

정밀도(Precision)와 재현율(Recall, Sensitivity)의 조화평균

조화평균을 사용하는 이유

불균형 데이터에 대해 정확도의 한계점을 보완

상충관계(Trade-off)에 있는 정밀도와 재현율을 모두 균형있게 반영

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FN + FP}$$

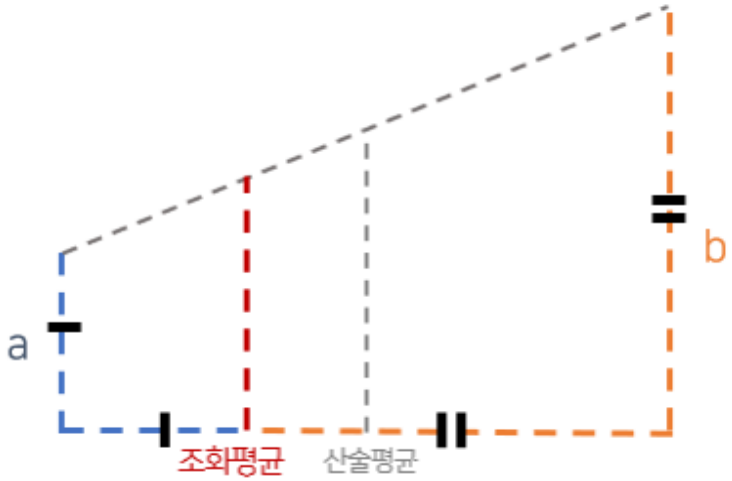
1에 가까울 수록 모델의 성능이 좋다고 판단

분류평가지표

조화평균을 사용하는 이유

불균형 데이터에서 **정확도(Accuracy)**의 한계 보완

상충관계(Trade-off)에 있는 **정밀도**와 **재현율**을 모두 **균형**있게 반영



관측치가 더 많은 클래스에 **패널티**를 줌

해당 클래스에 대한 의존성을 줄여

두 클래스를 **균형**있게 반영

분류평가지표

조화평균을 사용하는 이유

불균형 데이터에서 정확도(Accuracy)의 한계 보완

상충관계(Trade-off)에 있는 정밀도와 재현율을 모두 균형있게 반영

정밀도나 재현율 중 한 지표만을 이용하여 성능을 평가하지 않고

정밀도와 재현율을 모두 고려하여 더 좋은 모델 성능 지표를 찾을 수 있음

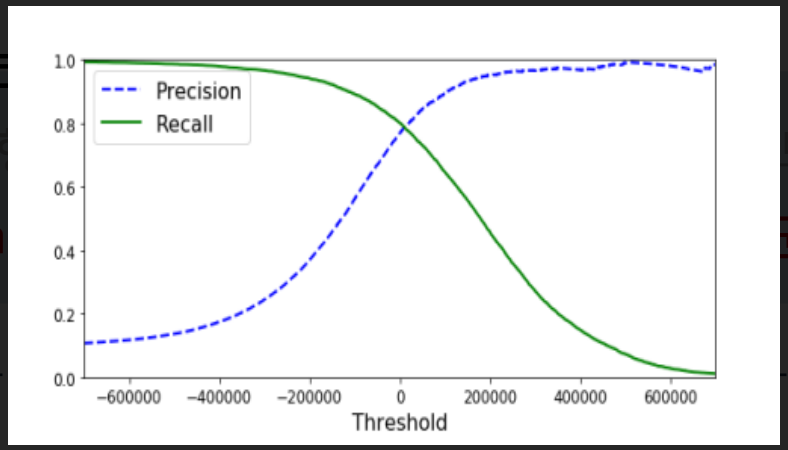
1 혼동행렬

분류평가지표

조화평균을 사용하는 불균형 보완

상충관계(Trade-off)

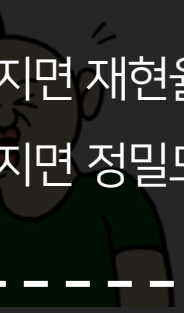
정밀도와 재현율의 상충관계(Trade-off)



정밀도나 재현율 중 한 지표만을 이용하여 성능을 평가하지 않고

정밀도와 재현율은 동시에 큰 값을 지닐 수 없음

정밀도와 재현율을 모두 고려하여 더 좋은 모델 성능 지표를 찾을 수 있음



정밀도가 높아지면 재현율이 낮아짐

재현율이 높아지면 정밀도가 낮아짐

분류평가지표

F1-Score의 한계

F1-Score는 TN(True Negative) 수치를 반영하지 않는다는 한계를 가짐

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	26	27
	$\hat{Y}=0$	24	22

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	26	27
	$\hat{Y}=0$	24	101

$$F1\ Score = \frac{2 \times 26}{2 \times 26 + 27 + 24} = 0.505$$

같은 값을 가짐

분류평가지표

F1-Score의 한계

F1-Score는 TN(True Negative) 수치를 반영하지 않는다는 한계를 가짐

TN(True Negative)의 수치를 반영하지 않는 F1-Score의 한계점

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	26	27
	$\hat{Y}=0$	24	22

MCC 지표를 사용해 보완!

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	26	27
	$\hat{Y}=0$	24	101

$$F1\ Score = \frac{2 \times 26}{2 \times 26 + 27 + 24} = 0.505$$

같은 값을 가짐

분류평가지표

MCC(Matthews Correlation Coefficient)

혼동행렬의 **모든 구성요소**를 활용하여 계산

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

상관계수 값이기 때문에 **-1**과 **1**사이의 값을 가짐

1에 가까울 수록 **완전 예측**

0에 가까울 수록 **랜덤 예측**

-1에 가까울 수록 **역예측**

분류평가지표

MCC(Matthews Correlation Coefficient)

혼동행렬의 **모든 구성요소**를 활용하여 계산

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

상관계수 값이기 때문에 **-1**과 **1**사이의 값을 가짐

1에 가까울 수록 **완전 예측**

0에 가까울 수록 **랜덤 예측**

-1에 가까울 수록 **역예측**

MCC

		관측값(Y)	
		$Y = 1$	$Y = 0$
예측값 ($\hat{Y}$)	$\hat{Y} = 1$	92	4
	$\hat{Y} = 0$	3	1

		관측값(Y)	
		$Y = 1$	$Y = 0$
예측값 ($\hat{Y}$)	$\hat{Y} = 1$	1	3
	$\hat{Y} = 0$	4	92

TP 관측치가 매우 큰 불균형 데이터

$$F1\ Score = \frac{2 \times 92}{2 \times 92 + 4 + 3} = 0.96\ ,\ MCC = \frac{(92 \times 1) - (4 \times 3)}{\sqrt{(92+4)(92+3)(1+4)(1+3)}} = 0.18$$

MCC

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	92	4
	$\hat{Y}=0$	3	1

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	1	3
	$\hat{Y}=0$	4	92

TN 관측치가 매우 큰 불균형 데이터

$$F1\ Score = \frac{2 \times 1}{2 \times 1 + 3 + 4} = 0.22\ ,\ MCC = \frac{(1 \times 92) - (3 \times 4)}{\sqrt{(1+3)(1+4)(92+3)(92+4)}} = 0.18$$

MCC

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	92	4
	$\hat{Y}=0$	3	1

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	1	3
	$\hat{Y}=0$	4	92

F1-Score는 큰 차이가 나타남

$$F1\ Score = \frac{2 \times 1}{2 \times 1 + 3 + 4} = 0.22$$

$$F1\ Score = \frac{2 \times 92}{2 \times 92 + 4 + 3} = 0.96$$

MCC

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	92	4
	$\hat{Y}=0$	3	1

		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	1	3
	$\hat{Y}=0$	4	92

MCC는 0.18로 같음!

$$MCC = \frac{(92 \times 1) - (4 \times 3)}{\sqrt{(92+4)(92+3)(1+4)(1+3)}} = 0.18$$

$$MCC = \frac{(1 \times 92) - (3 \times 4)}{\sqrt{(1+3)(1+4)(92+3)(92+4)}} = 0.18$$

혼동행렬의 분류평가지표

MCC

F1-Score는 TN값을 반영하지 않기 때문에
TN값의 차이가 클 수록 F1-Score의 값에 차이가 발생

F1-Score 한가지만 가지고 모델의 성능을 판단하는 것은 매우 위험

예측값 ($\hat{Y}$)		관측값(Y)	
		$Y=1$	$Y=0$
	$\hat{Y}=1$	92	4
	$\hat{Y}=0$	3	1

예측값 ($\hat{Y}$)		관측값(Y)	
		$Y=1$	$Y=0$
	$\hat{Y}=1$	1	3
	$\hat{Y}=0$	4	92

MCC는 0.18로 같음!

$$MCC = \frac{(92 \times 1) - (4 \times 3)}{\sqrt{(92+4)(92+3)(1+4)(1+3)}} = 0.18$$

$$MCC = \frac{(1 \times 92) - (3 \times 4)}{\sqrt{(1+3)(1+4)(92+3)(92+4)}} = 0.18$$

1 혼동행렬

혼동행렬의 분류평가지표

MCC가 항상 F1-Score보다 좋은 지표일까?

MCC

예측값 $(\hat{Y})$	관측값(Y)	
	$Y=1$	$Y=0$
$\hat{Y}=0$	3	1

예측값 $(\hat{Y})$	관측값(Y)	
	$Y=1$	$Y=0$
$\hat{Y}=0$	4	92

분석 목적이 모든 클래스에 대한 **균형적인 평가**라면 MCC

희귀질환과 같이 중요하지만 관측치가 적은 경우, 이를 **positive**로 두고 F1-Score

MCC는 0.18로 같음!

목적에 맞게 **지표**를 선택하는 것이 중요!

$$MCC = \frac{(92 \times 1) - (4 \times 3)}{\sqrt{(92+4)(92+3)(1+4)(1+3)}} = 0.18$$

$$MCC = \frac{(1 \times 92) - (3 \times 4)}{\sqrt{(1+3)(1+4)(92+3)(92+4)}} = 0.18$$

혼동행렬의 한계점

정보의 손실

임의적인 cut-off point 설정

임의의 **cut-off point**에 따라
이항변수에 맞게 **범주화**

연속적인 확률 값을 **이항의 값**으로
변환시키는 과정에서
숫자가 갖는 정보를 잃게 됨

cut-off point를 분석자가 **임의**로 설정

분석의 객관성을 떨어트림
혼동행렬이 크게 **바뀔 수 있음**

혼동행렬의 한계점

정보의 손실

임의적인 cut-off point 설정

임의의 cut-off point에 따라
이항변수에 맞게 **범주화**

연속적인 확률 값을 **이항의 값**으로
변환시키는 과정에서
숫자가 갖는 정보를 잃게 됨

cut-off point를 분석자가 **임의**로 설정

분석의 객관성을 떨어트림
혼동행렬이 크게 **바뀔 수 있음**

1 혼동행렬

혼동행렬의 한계점

혼동행렬은 특정 **cut-off point**를 기준으로
관측값과 예측값을 분류하여 나열한 도표이기 때문에
cut-off point가 변화함에 따라 **검정력**이 어떻게 **변화**하는지 파악하기 어려움

cut-off point를 분석자가 **임의**로 설정

ROC 곡선을 이용해 **한계**를 보완! **객관성**을 떨어트림

혼동행렬이 크게 **바뀔 수 있음**

2

ROC 곡선

ROC 곡선

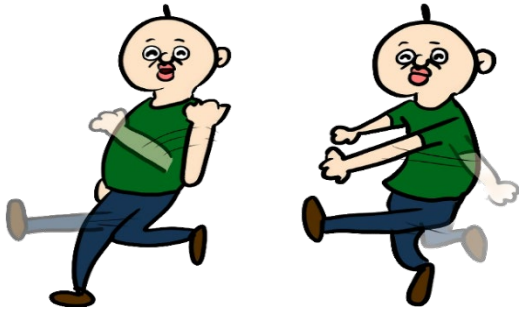
0 ~ 1 범위의 **모든 cut-off point**에 대해

재현율(Y축)과 **1-특이도**(FPR, X축)의 함수로 나타낸 곡선

ROC 곡선의 장점

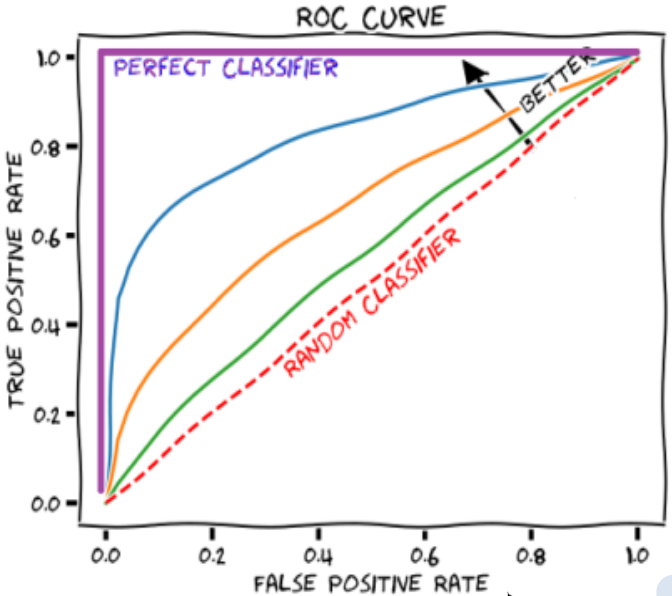
혼동행렬보다 **더 많은 정보**를 가짐

주어진 모형에서 **가장 적합한** cut-off point를 찾을 수 있음



ROC 곡선

재현율(TPR)



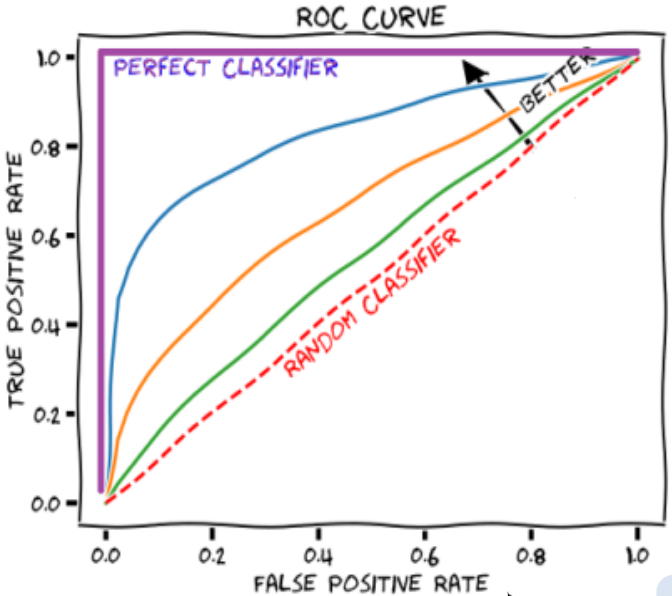
1 – 특이도(FPR)

cut-off point가 0에 가까워짐 → 대부분을 $\hat{Y} = 1$ 로 예측 → TP & FP 증가

→ TN & FN 감소 → TPR & FPR (1,1)에 가까워짐

ROC 곡선

재현율(TPR)



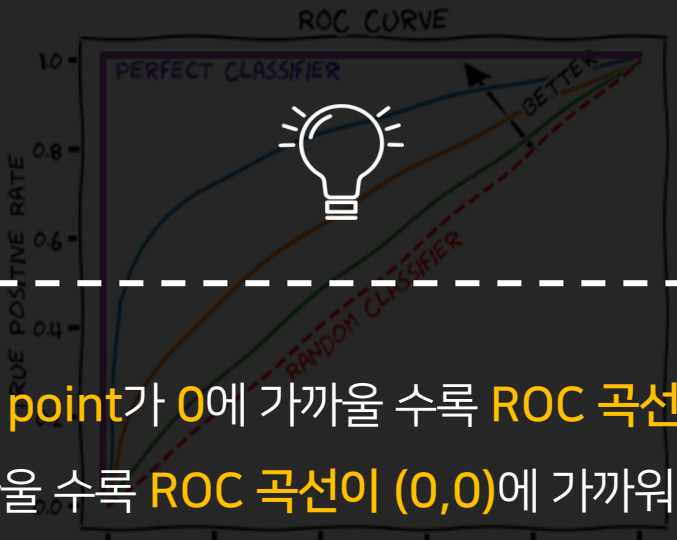
1 – 특이도(FPR)

cut-off point가 1에 가까워짐 → 대부분을 $\hat{Y} = 0$ 로 예측 → TP & FP 감소

→ TN & FN 증가 → TPR & FPR (0,0)에 가까워짐

2 ROC 곡선

ROC 곡선



재현율(TPR)

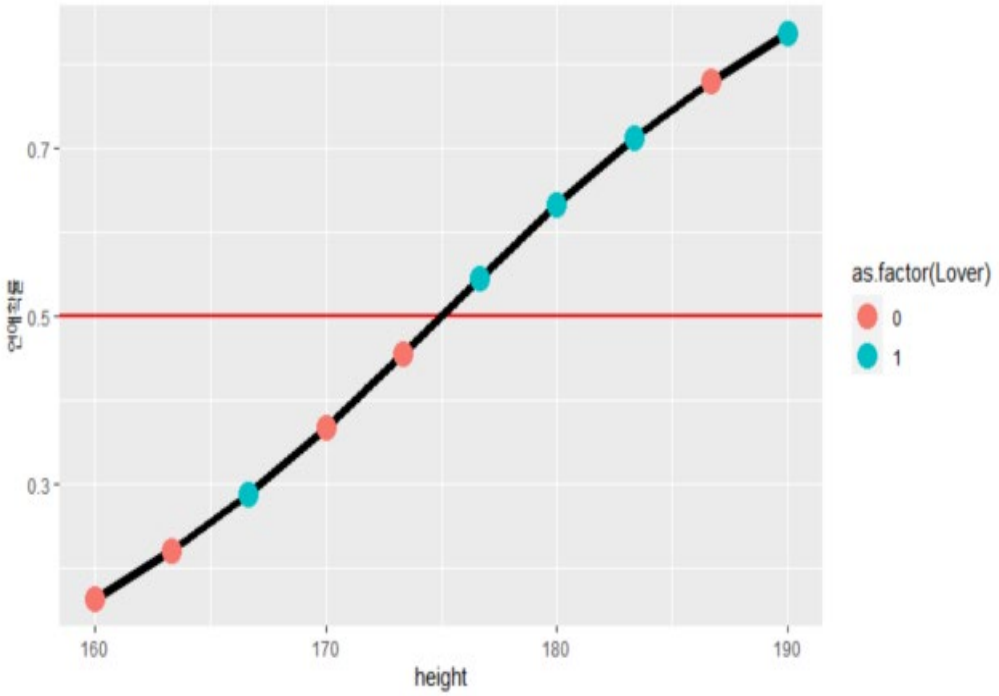
cut-off point가 0에 가까울 수록 ROC 곡선이 (1,1)에,
1에 가까울 수록 ROC 곡선이 (0,0)에 가까워지기 때문에
ROC 곡선이 우상향 곡선의 형태를 띄게 됨

1 – 특이도(FPR)

cut-off point가 1에 가까워짐 → 대부분을 $\hat{Y} = 0$ 로 예측 → TP & FP 감소
→ TN & FN 증가 → TPR & FPR (0,0)에 가까워짐

2 ROC 곡선

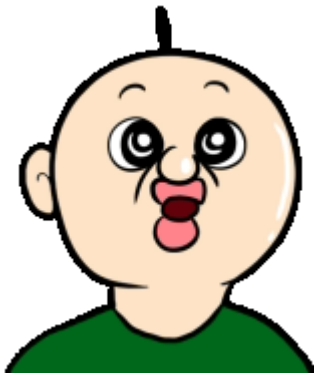
최적의 cut-off point 찾기



키에 따른 연애 여부를 나타낸 그래프

파란색 원은 연애 중인 관측치

빨간색 원은 연애를 하고 있지 않은 관측치



$$logit[\pi(x)] = -19 + 0.1x$$

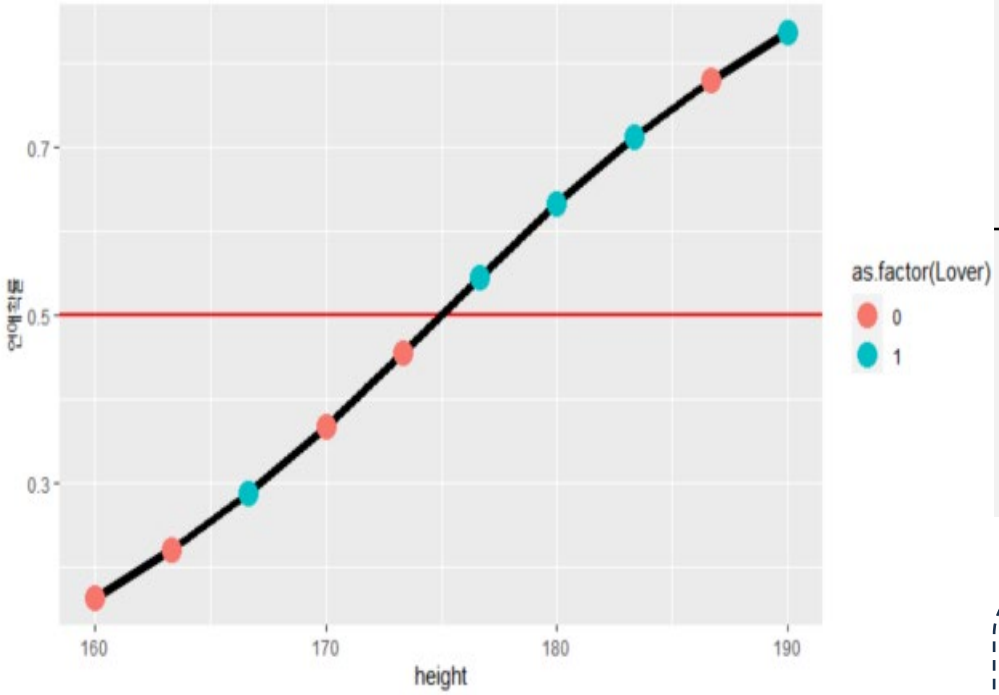
cut-off point가 0.5라면

0.5보다 큰 관측치는 $\hat{Y} = 1$로 예측,

0.5보다 작은 값은 $\hat{Y} = 0$로 예측

2 ROC 곡선

최적의 cut-off point 찾기



키에 따른 연애 여부를 나타낸 그래프

파란색 원은 연애 중인 관측치

빨간색 원은 연애를 하고 있지 않은 관측치

모든 cut-off point에 대해 혼동행렬을 구해 TPR, FPR을 구함		관측값(Y)	
		$Y=1$	$Y=0$
예측값 ($\hat{Y}$)	$\hat{Y}=1$	4	1
	$\hat{Y}=0$	1	4

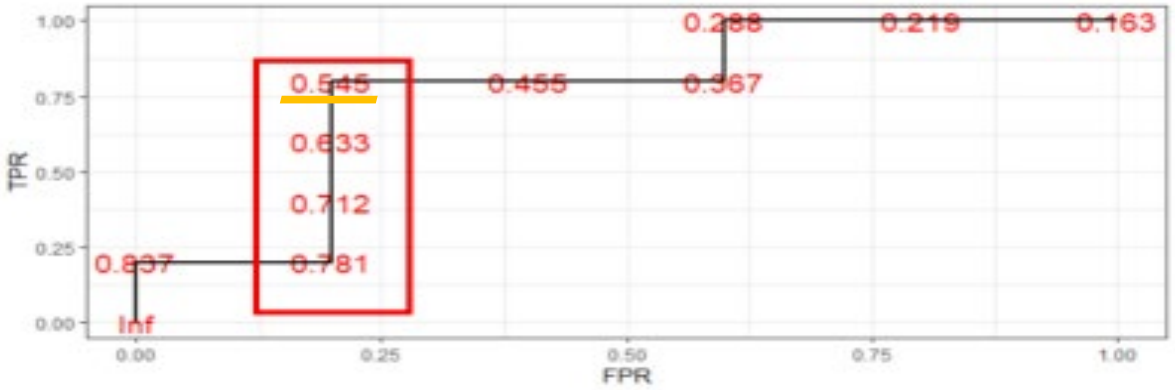
$$\text{TPR} = \frac{TP}{TP+FN} = \frac{4}{4+1} = 0.8$$

$$\text{FPR} = \frac{FP}{FP+TN} = \frac{1}{1+4} = 0.2$$

2 ROC 곡선

최적의 cut-off point 찾기

ROC 곡선의 빨간 글씨는 각 cut-off point를 나타냄



TPR이 1에 가까울 수록, FPR이 0에 가까울 수록 좋음을 이용

Y값(TPR)이 같을 때, **X값(FPR)**이 더 **작을수록** 좋은 cut-off point

X값(FPR)이 같을 때, **Y값(TPR)**이 더 **클수록** 좋은 cut-off point

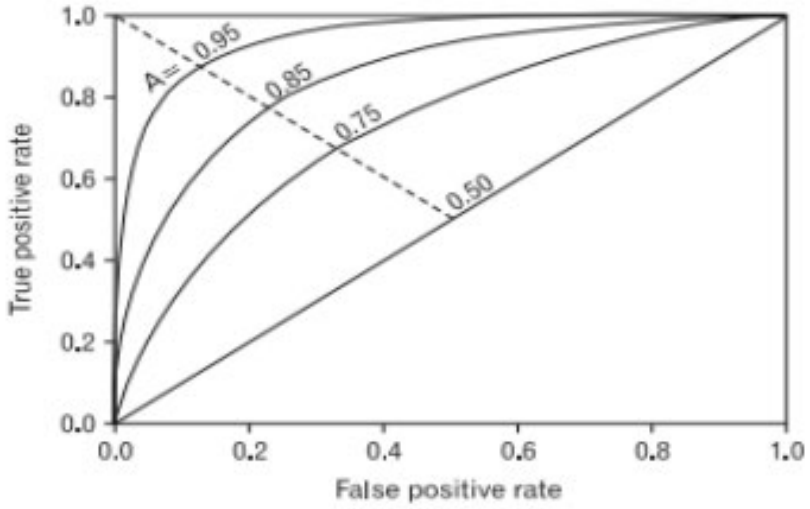
0.545가 최적의 cut-off point!

2 ROC 곡선

AUC

AUC

ROC 곡선 아래의 면적을 의미



AUC (Area Under Curve)

- 0 ~ 1 사이의 값을 가짐
- **AUC 값이 클수록** 모델 성능이 좋음

→ 0.95의 높은 AUC 값을 지닌

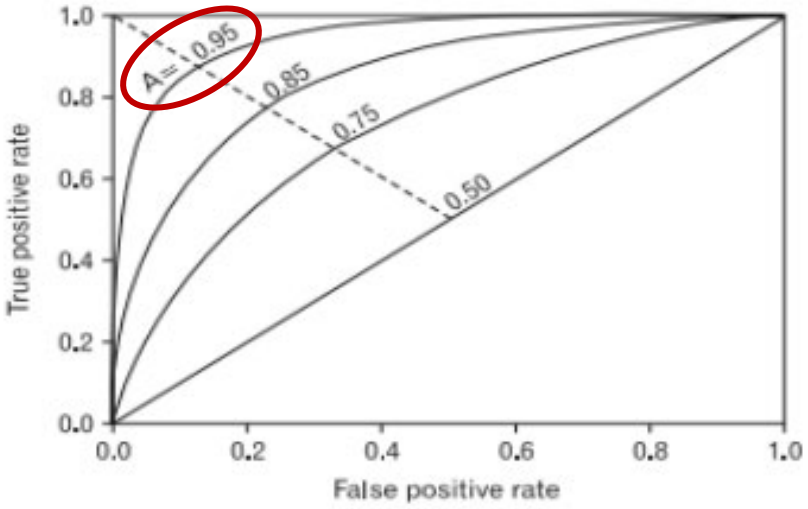
첫 번째 곡선의 모델 성능이 가장 우수!

2 ROC 곡선

AUC

AUC

ROC 곡선 아래의 면적을 의미

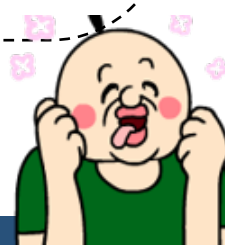


AUC (Area Under Curve)

- 0 ~ 1 사이의 값을 가짐
- **AUC 값이 클수록** 모델 성능이 좋음

→ 0.95의 높은 AUC 값을 지닌

첫 번째 곡선의 모델 성능이 가장 우수!



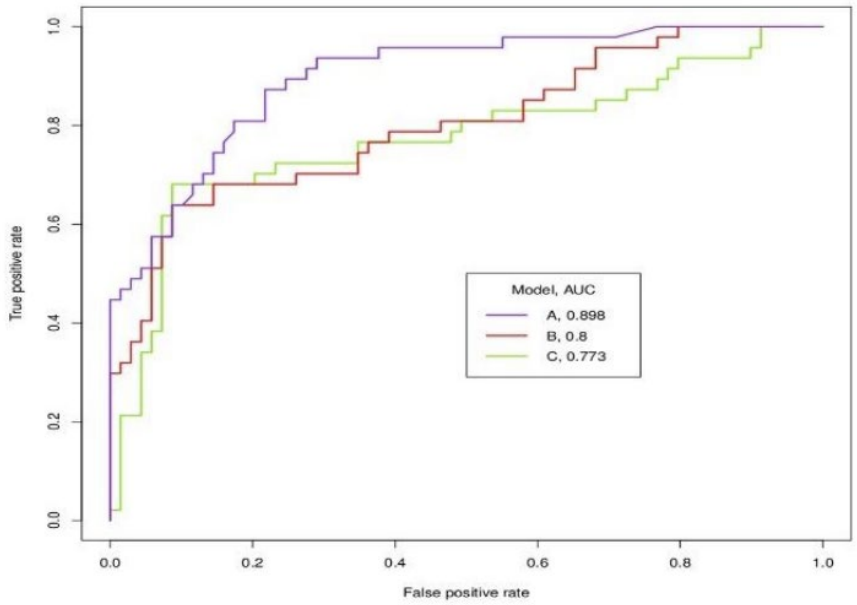
2 ROC 곡선

AUC

AUC

ROC 곡선 아래의 면적을 의미

3개 모델의 ROC 곡선



각 AUC를 계산해 보았을 때

가장 볼록한 형태의 A 모델이 0.898로

가장 높음

A 모델의 성능이 가장 좋음!

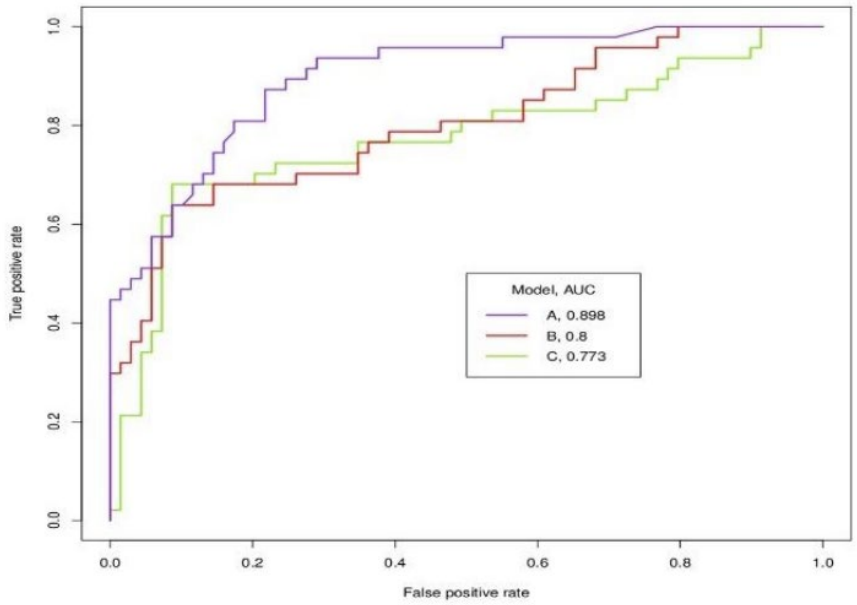
2 ROC 곡선

AUC

AUC

ROC 곡선 아래의 면적을 의미

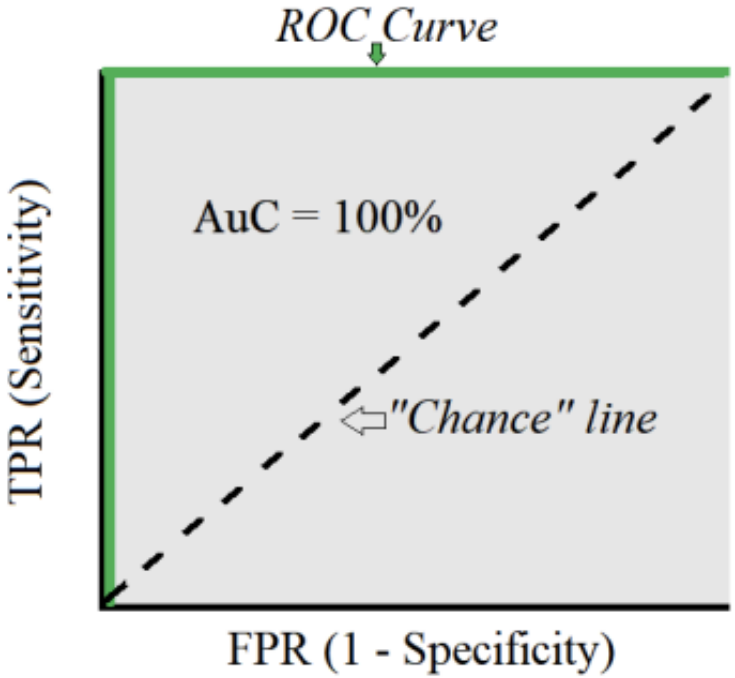
3개 모델의 ROC 곡선



ROC curve가 모든 cut-off point를 고려했기 때문에 AUC도 cut-off point와 상관없이 모델의 성능 측정 가능

2 ROC 곡선

AUC의 해석



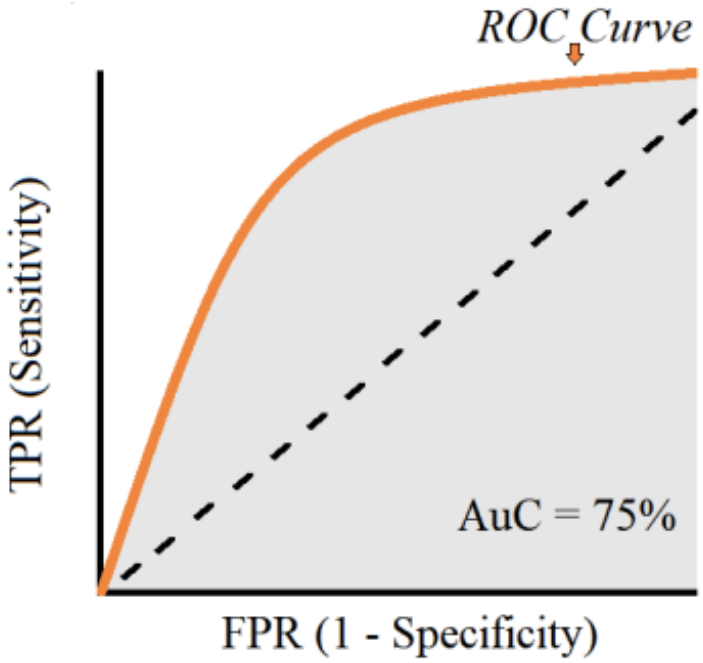
AUC = 1인 경우

모델의 100% **정확히 예측**했다는 의미

모델이 **과적합**된 것은 아닌지 확인 필요

2 ROC 곡선

AUC의 해석



AUC = 0.75인 경우

모델이 실제 값을 **75% 수준**으로

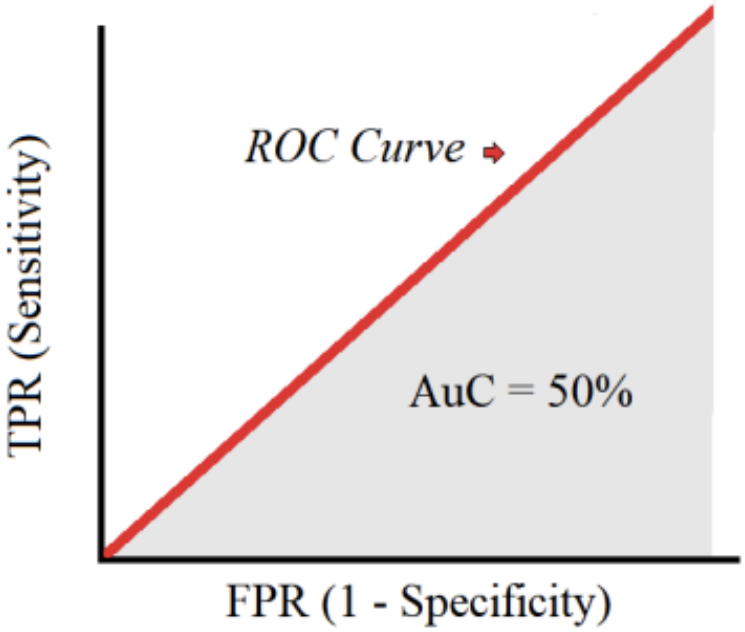
맞췄다는 의미

→ 일반적으로 **AUC가 0.8 이상**이라면

성능이 우수하다고 함

2 ROC 곡선

AUC의 해석



AUC = 0.5인 경우

모델이 실제 값을 50%, **절반**만 맞췄다는 의미

→ 무작위로 예측한 것과 다름이 없음

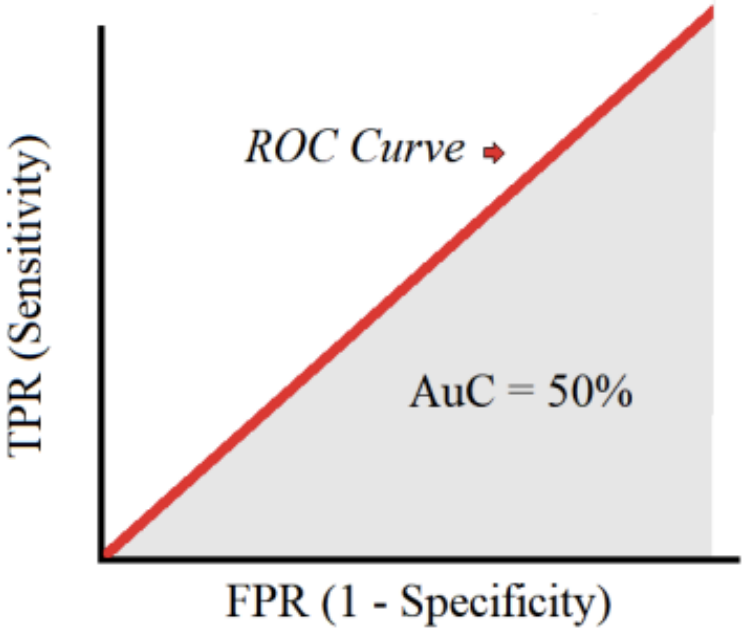
보통 **0.5 이상의 값**을 보여야 정상

0.5보다 낮은 값이 나왔다면 **분류를 반대로 진행**했을 가능성이 큼

즉, Y=1과 Y=0을 **거꾸로** 예측한 것!

2 ROC 곡선

AUC의 해석



AUC = 0.5인 경우

모델이 실제 값을 50%, **절반**만 맞췄다는 의미

→ 무작위로 예측한 것과 다름이 없음

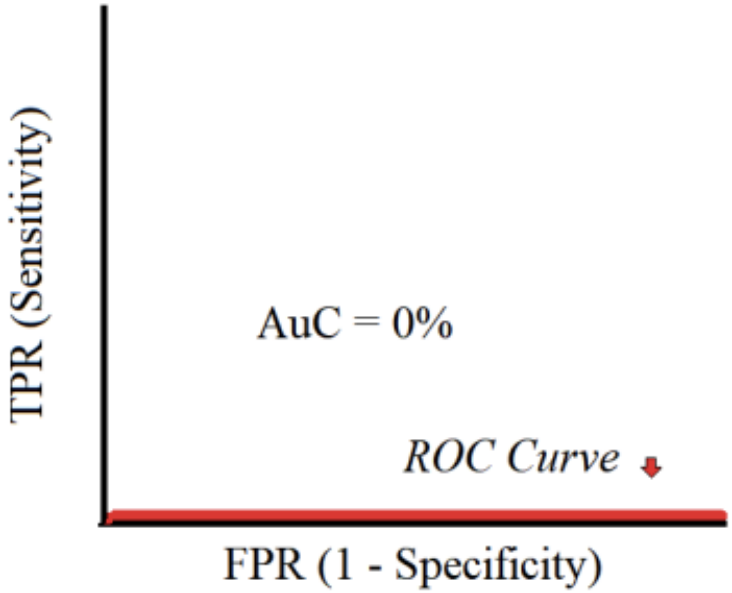
보통 **0.5 이상의 값**을 보여야 정상

0.5보다 낮은 값이 나왔다면 **분류를 반대로 진행**했을 가능성이 큼

즉, **Y=1과 Y=0을 거꾸로** 예측한 것!

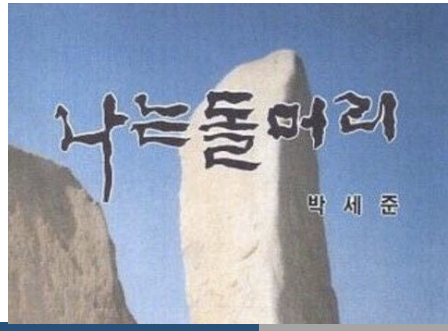
2 ROC 곡선

AUC의 해석



AUC = 0인 경우

모델이 100% 반대로 예측했다는 의미



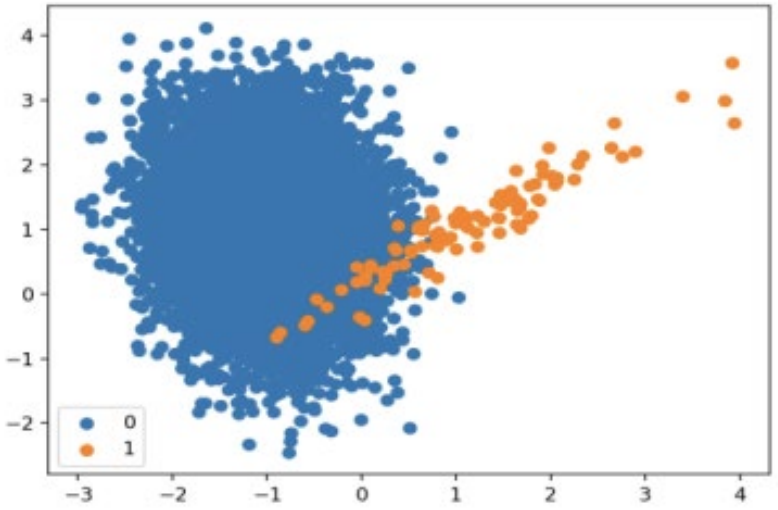
3

샘플링

클래스 불균형

클래스 불균형

각 수준(클래스)에 따라 관측치의 차이가 큰 경우



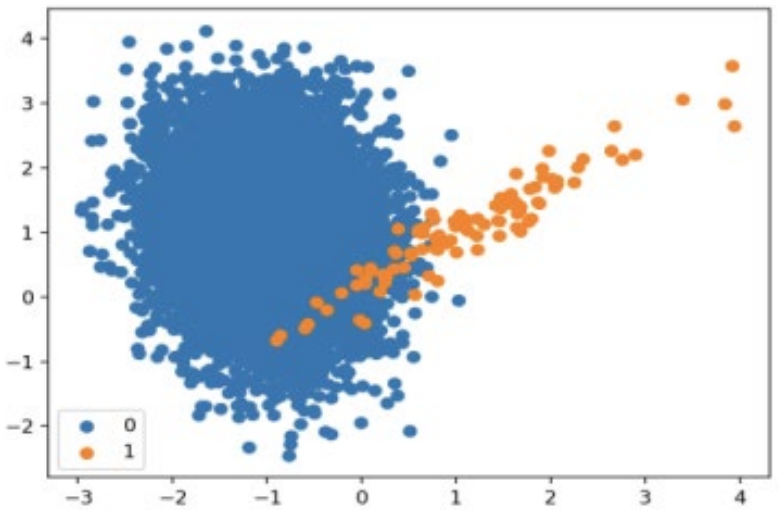
0인 클래스가 1인 클래스에 비해
훨씬 많이 관측됨

Ex) 올림픽 메달이 있는 사람과 없는 사람

클래스 불균형

클래스 불균형

각 수준(클래스)에 따라 관측치의 차이가 큰 경우



샘플링을 통해 해결 가능!

샘플링의 필요성

		Y	
		$Y=1$	$Y=0$
$\hat{Y}$	$\hat{Y}=1$	60	5
	$\hat{Y}=0$	40	5
	클래스 별 정확도	0.6	0.5

전체 정확도 : 0.59

		Y	
		$Y=1$	$Y=0$
$\hat{Y}$	$\hat{Y}=1$	50	4
	$\hat{Y}=0$	50	6
	클래스 별 정확도	0.5	0.6

전체 정확도 : 0.509

Y=1에서 **관측치가 많은** 왼쪽 혼동행렬의 정확도가 높게 나타남

샘플링의 필요성

		Y	
		$Y=1$	$Y=0$
$\hat{Y}$	$\hat{Y}=1$	60	5
	$\hat{Y}=0$	40	5
	클래스 별 정확도	0.6	0.5

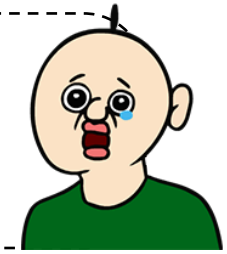
전체 정확도 : 0.59

		Y	
		$Y=1$	$Y=0$
$\hat{Y}$	$\hat{Y}=1$	50	4
	$\hat{Y}=0$	50	6
	클래스 별 정확도	0.5	0.6

전체 정확도 : 0.509

정확도는 **관측치가 많은 수준의 영향**을 받으므로

모델의 성능을 올바르게 평가하지 못할 가능성 존재



3 샘플링

샘플링의 필요성

따라서 소수의 클래스에 특별히 더 관심이 있는 경우

샘플링을 통한 클래스 불균형 해소가 필수적!

클래스 간 불균형을 무시하고 모델을 적합하게 된다면

특정 클래스에 **과적합**될 수 있으니 조심해야 함

		Y	
		$Y=1$	$Y=0$
$\hat{Y}$	$\hat{Y}=1$	60	5
	$\hat{Y}=0$	40	5
	클래스 별 정확도	0.6	0.5

전체 정확도 : 0.58

		Y	
		$Y=1$	$Y=0$
$\hat{Y}$	$\hat{Y}=1$	50	4
	$\hat{Y}=0$	50	6
	클래스 별 정확도	0.5	0.6

전체 정확도 : 0.509

소수의 관측치를 지니는 클래스에 대해

정확도가 떨어질 수밖에 없기 때문!

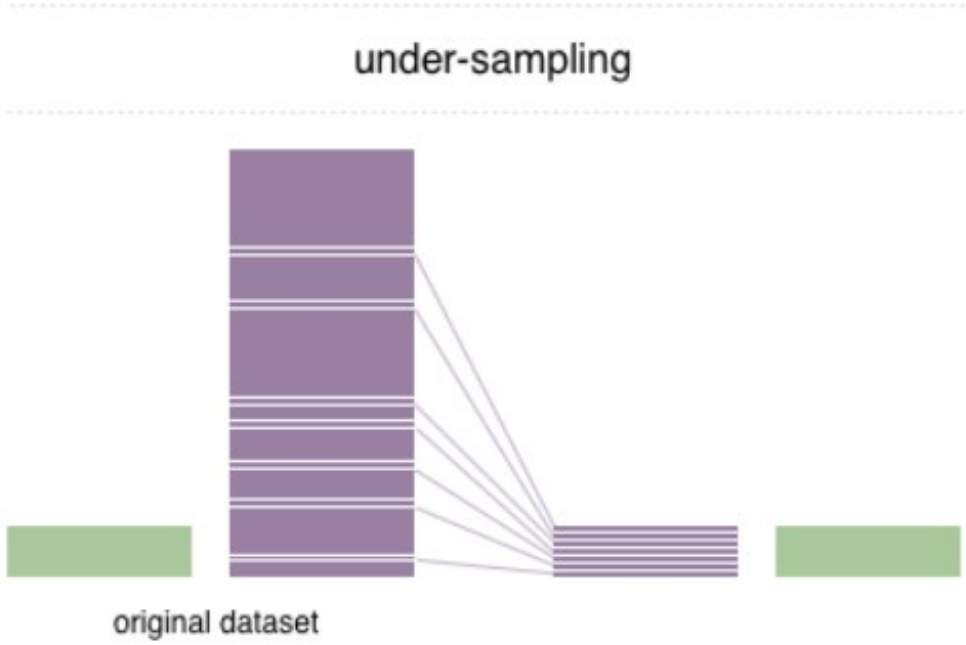
정확도는 **관측치가 많은 수준의 영향**을 받으므로

모델의 성능을 올바르게 평가하지 못할 가능성 존재

언더 샘플링

언더 샘플링 (Under Sampling)

다수의 클래스를 소수의 클래스에 맞추어 관측치를 감소시키는 방법



랜덤 언더 샘플링

랜덤 언더 샘플링 (Random Under Sampling)

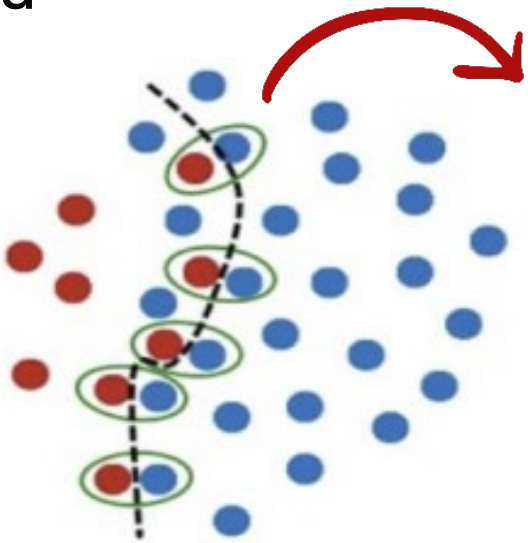
랜덤으로 다수의 클래스에 해당하는 데이터를 제거해 관측치의 수를 줄이는 방법

임의적으로 제거한 **데이터의 정보가 누락**되거나

추출된 샘플들이 **기존 데이터에 대해 대표성**을 띄지 못한다면

부정확한 결과 초래 가능

Tomek Links Method



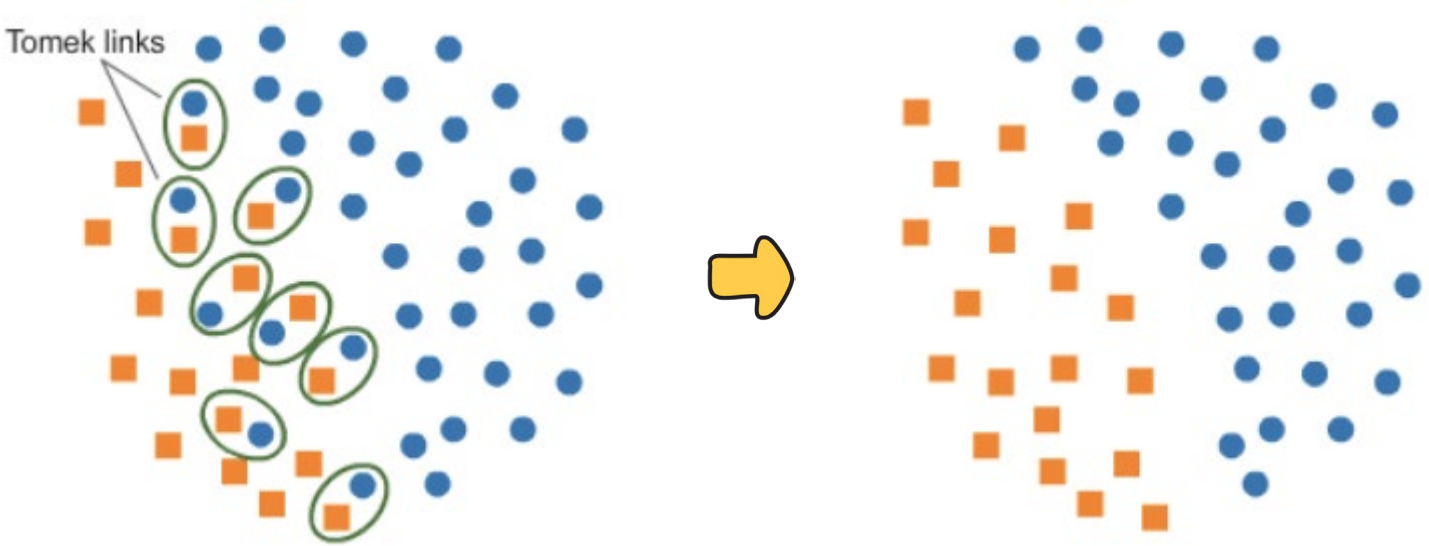
초록색 동그라미는
Tomek Links가 있는 점들을
묶어 놓은 것

임의로 **서로 다른 클래스**의 데이터 **두 점을 선택**하고 **연결**

해당 거리가 주위에 있는 다른 데이터와 서로 연결한 거리보다 짧다면

두 점 간 Tomek Link가 있다고 함

Tomek Links Method

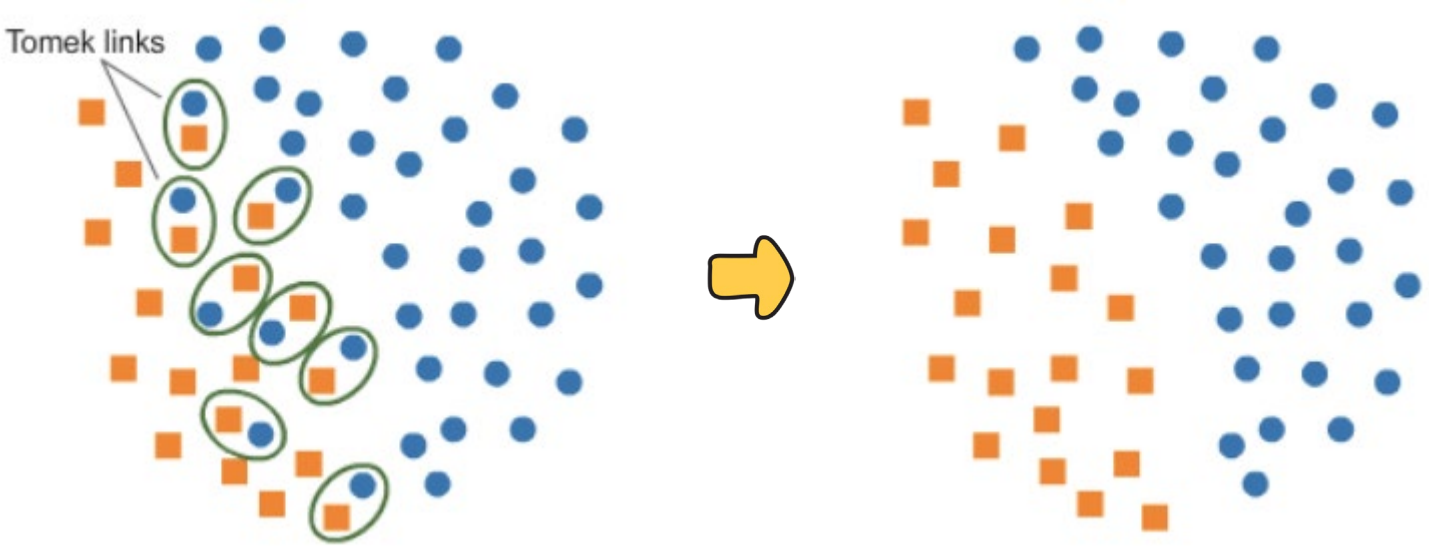


초록색 동그라미로 **묶인 데이터 쌍**에서 **다수의 클래스에 속한 데이터**들을

삭제함으로써 데이터들의 크기를 줄이는 방법

→ 두 클래스 간 **경계에 있는 노이즈 데이터를 제거**하는 방식

Tomek Links Method



분포가 높은 클래스의 중심분포는 어느 정도 유지하며 경계선을 조정하기 때문에
랜덤 언더 샘플링보다 **정보의 유실을 크게 방지**할 수 있지만
묶이는 값이 한정적이므로 언더 샘플링의 효과가 크지 않음

언더 샘플링

언더 샘플링의 장점

데이터의 사이즈가 줄어들어 **메모리 사용**이나 **처리 속도** 측면에서 **유리**

언더 샘플링의 단점

관측치 손실로 인해 **정보가 누락**되는 문제 발생

언더 샘플링

언더 샘플링의 장점

데이터의 사이즈가 줄어들어 **메모리 사용**이나 **처리 속도** 측면에서 **유리**

언더 샘플링의 단점

관측치 손실로 인해 **정보가 누락**되는 문제 발생

3 샘플링

언더 샘플링

언더 샘플링의 장점

데이터의 사이즈가 줄어들어 **메모리 사용**이나 **처리 속도** 측면에서 **유리**

언더 샘플링은 다수 클래스의 데이터를 삭제하는 방법이므로 **정보의 누락** 발생 가능

언더 샘플링의 단점

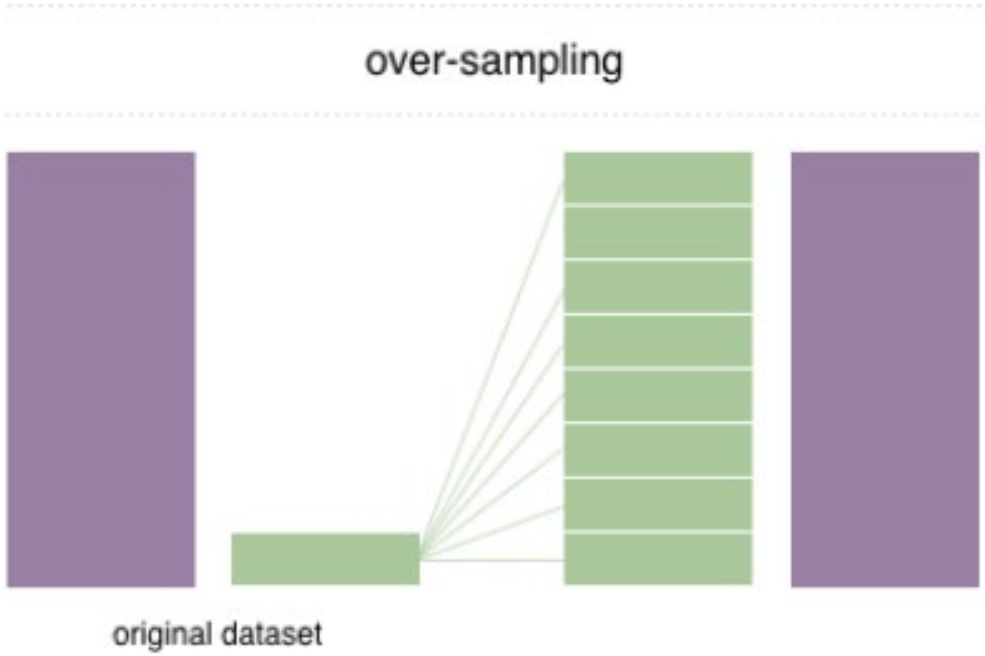
일반적으로 **오버 샘플링**을 사용!

관측치 손실로 인해 **정보가 누락**되는 문제 발생

오버 샘플링

오버 샘플링 (Over Sampling)

소수의 클래스를 다수의 클래스에 맞추어 관측치를 증가시키는 방법



랜덤 오버 샘플링

랜덤 오버 샘플링 (Random Over Sampling)

랜덤으로 소수의 클래스의 데이터를 복제하여 관측치의 수를 늘리는 방법

기존의 데이터를 그대로 복제하는 방식이기 때문에
동일한 데이터의 수가 늘어나 과적합될 가능성이 큼

3 샘플링

SMOTE

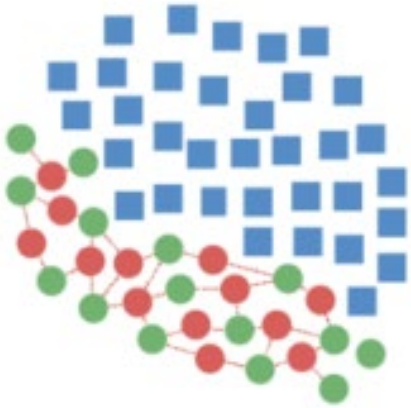
① **소수 클래스** 중 **무작위**로 하나를 **선택**

② 선택한 데이터를 기준으로 KNN 알고리즘을 활용해
K개의 가장 가까운 데이터 탐색 후 선택

③ 선택한 데이터와 K개의 데이터 사이에 직선을 그리고
그 직선 상에 가상의 소수 클래스 데이터 생성

3 샘플링

SMOTE



① **소수 클래스** 중 **무작위**로 하나를 **선택**

② 선택한 데이터를 기준으로 **KNN 알고리즘**을 활용해
K개의 가장 가까운 데이터 탐색 후 **선택**

③ 선택한 데이터와 K개의 데이터 사이에 직선을 그리고
그 직선 상에 가상의 소수 클래스 데이터 생성

3 샘플링

SMOTE

① **소수 클래스** 중 **무작위**로 하나를 **선택**

② 선택한 데이터를 기준으로 KNN 알고리즘을 활용해
K개의 가장 가까운 데이터 탐색 후 **선택**

③ 선택한 데이터와 K개의 데이터 사이에 직선을 그리고
그 직선 상에 **가상의 소수 클래스 데이터** 생성

3 샘플링

SMOTE

SMOTE는 가상의 데이터를 생성하기 때문에

데이터를 단순 복사하는 랜덤 오버 샘플링보다 **과적합이 발생할 위험이 적지만**

소수 클래스의 데이터들 간 거리만 고려해 새롭게 생성된 데이터가

다른 클래스의 데이터와 **겹치거나 노이즈가 발생할 수 있음**

② 선택한 데이터를 기준으로 **KNN 알고리즘**을 활용해

K개의 **가장 가까운 데이터** 탐색 후 선택

고차원 데이터에선 SMOTE가 **효율적이지 않음!**

③ 선택한 데이터와 K개의 데이터 사이에 직선을 그리고

그 직선 상에 **가상의 소수 클래스 데이터** 생성

오버 샘플링

오버 샘플링의 장점

정보의 손실이 발생하지 않아 일반적으로 언더 샘플링보다 성능이 좋음

오버 샘플링의 단점

데이터가 커지기 때문에 **메모리 사용**이나 **처리 속도** 측면에서 상대적으로 **불리**

오버 샘플링

오버 샘플링의 장점

정보의 손실이 발생하지 않아 일반적으로 언더 샘플링보다 성능이 좋음

오버 샘플링의 단점

데이터가 커지기 때문에 **메모리 사용**이나 **처리 속도** 측면에서 상대적으로 **불리**

4

인코딩

인코딩

인코딩 (Encoding)

문자열 혹은 기호로 표현되어 있는 범주형 데이터를

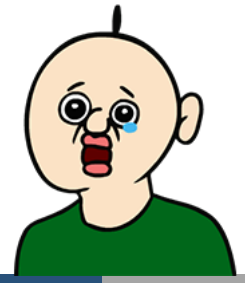
수치화하는 과정

회귀모델과 같은 **수치형** 변수들만 설명변수로 갖는 모델 적용 가능

4 인코딩

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	



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

One-Hot Encoding (Dummy Encoding)

데이터에 **가변수**를 추가하여 인코딩을 진행하는 기법

설명변수를 0과 1로 변환한 변수

호그와트 친구들	→	해리	헤르미온느	론	지니	네빌
해리		1	0	0	0	0
헤르미온느		0	1	0	0	0
론		0	0	1	0	0
지니		0	0	0	1	0
네빌		0	0	0	0	1

클래스의 개수만큼 열 추가하여 해당 범주에는 1을, 그 외 값에는 0 부여

One-Hot Encoding

기준 범주 해리를 삭제해도 데이터 유지하여

J-1개의 가변수로 J개의 수준을 갖는 범주형 변수 표현 가능

회귀분석에서의 자유도 반영!

호그와트 친구들		해리	헤르미온느	론	지니	네빌
해리	→	1	0	0	0	0
헤르미온느		0	1	0	0	0
론		0	0	1	0	0
지니		0	0	0	1	0
네빌		0	0	0	0	1

인코딩의 종류

트리 기반 모델의 경우,

기준 범주 해리를 삭제해도 데이터 유지하여

삭제하는 가변수 없이 J개의 가변수 생성 필요

J-1개의 가변수로 J개의 수준을 갖는 범주형 변수 표현 가능

사용 가능한 모든 부분을 활용해 트리 생성하기 때문에

삭제된 기준 범주가 트리 생성에 중요한 요소일 경우

트리가 잘못 학습될 가능성

호그와트 친구들	→	해리	헤르미온느	론	지니	네빌
해리		1	0	0	0	0
헤르미온느		0	0	0	0	0
론		0	0	1	0	0
지니		0	0	0	1	0
네빌		0	0	0	0	1

회귀모델은 J-1개의 가변수,

트리 기반 분류 모델은 J개의 가변수로 인코딩

One-Hot Encoding의 장점

해석 용이

기준 범주를 기준으로 모델 해석 가능

명목형 변수 값을 가장 잘 반영하는 방법

해당 수준에 속하는 경우만 1로 표현되므로

한 변수가 다른 변수들로 설명되는 **다중공선성 문제 해결**

4 인코딩

One-Hot Encoding의 장점

주어진 범주형 변수의 수준이 많거나
데이터 상에서 범주형 변수가 지나치게 많을 경우

명목형 변수 값을 가장 잘 반영하는 방법

너무 많은 가변수 생성으로 인해 **차원이 늘어나는 문제** 발생

모델의 학습 속도 저하와 많은 계산 요구

Label Encoding

명목형 자료가 주어졌을 때

범주형 변수의 **각 수준에 점수를 할당**하는 방법

혈액형	점수
A형	1
B형	2
O형	3
AB형	4

각 수준에 부여한 숫자들 사이의
의미나 연관성 존재하지 않음

Label Encoding

One-Hot Encoding과 달리 가변수를 생성하지 않아
차원이 늘어나지 않으므로 모델의 학습 속도 빠름

할당된 점수에 순서나 연관성이 있다고 잘못 판단하여 정보 왜곡할 가능성

Ordinal Encoding

순서형 자료가 주어졌을 때

각 순서에 대응하는 점수를 **차등적으로 할당**하는 방식

만족도	점수
매우 별로	1
별로	2
보통	3
좋음	4
매우 좋음	5

Label Encoding과 달리, 할당된 점수들 간 순서나 연관성 존재

Ordinal Encoding

차원이 늘어나지 않으므로

모델의 데이터 처리 속도 빠름

범주 내 수준 간 차이를 정확히 반영하기 어려움

Ordinal Encoding

해당 데이터에 대한 **도메인** 명확히 파악해야함

범주 내 수준 간 차이를 정확히 반영하기 어려움

Target Encoding

타겟으로부터 도출된 수치를 활용해 범주를 인코딩하는 방식

범주형 변수의 각 수준을 구별할 뿐만 아니라

해당 변수와 반응변수 간의 수치적인 관계 반영

Mean Encoding

범주형 변수의 각 수준에서 도출된 **반응변수의 평균**으로
수준별 점수를 할당하는 인코딩 방식

키(cm) [Y]	학과 [X]	Mean Encoding [X]
168	경영	172
180	경영	172
168	경영	172
174	통계	166
156	통계	166
163	통계	166
171	통계	166
165	경제	171.66
180	경제	171.66
170	경제	171.66

클래스별 반응변수에 대한 평균

Mean Encoding의 장점

당위성

설명변수와 **반응변수** 간의 **관계** 고려하여 점수 할당

빠른 학습 속도

One-Hot Encoding과 달리 차원이 증가하지 않아

학습 속도가 빠름

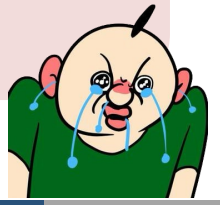
Mean Encoding의 한계점

Train set에 없던 **새로운 수준**이
Test set에 등장하면 활용 어려움

평균은 **이상치에 취약**
반응변수가 지나치게 크거나 작을 경우
이상치의 영향을 많이 받음

반응변수 값 활용한 설명변수 인코딩으로
Data Leakage 문제
모델 학습 시 **과적합** 발생 위험

관측치 값이 **적은** 범주의 경우
모델링 시 부정확한 결과 도출 가능성



Leave-One-Out Encoding (LOO Encoding)

인코딩하고자 하는 **현재 행을 제외한**

나머지 행들의 평균을 점수로 할당하는 방식

키(cm) [Y]	학과 [X]	Mean Encoding [X]
168	경영	174
180	경영	168
168	경영	174

$$\frac{180 + 168}{2}$$

$$\frac{168 + 168}{2}$$

$$\frac{168 + 180}{2}$$

이상치의 영향을 줄여 Mean Encoding의 한계점 극복

Leave-One-Out Encoding (LOO Encoding)

과적합의 가능성이 Mean Encoding보다 **낮음**

모든 반응변수의 정보를 다 반영하지는 않기 때문

Mean Encoding**과 동일**한 한계점을 지님

Ordered Target Encoding (CatBoost Encoding)

같은 수준에 속한 행들 중 이전 행 값들의 평균을

점수로 할당하는 방식

키(cm) [Y]	학과 [X]	Mean Encoding	Ordered Target Encoding
168	경영	172	169.5
174	통계	166	169.5
165	경제	171.66	169.5
156	통계	166	174
180	경영	172	168
163	통계	166	165
180	경제	171.66	165
170	경제	171.66	172.5
168	경영	172	174
171	경제	166	164.33

Ordered Target Encoding (CatBoost Encoding)

키(cm) [Y]	학과 [X]	Mean Encoding	Ordered Target Encoding
168	경영	172	169.5
174	통계	166	169.5
165	경제	171.66	169.5
156	통계	166	174
180	경영	172	168
163	통계	166	165
180	경제	171.66	165
170	경제	171.66	172.5
168	경영	172	174
171	경제	166	164.33

이전 행이 없으므로 전체 데이터의 평균 할당

Ordered Target Encoding (CatBoost Encoding)

키(cm) [Y]	학과 [X]	Mean Encoding	Ordered Target Encoding
168	경영	172	169.5
174	통계	166	169.5
165	경제	171.66	169.5
156	통계	166	174
180	경영	172	168
163	통계	166	165
180	경제	171.66	165
170	경제	171.66	172.5
168	경영	172	174
171	경제	166	164.33

앞선 두 통계학과 값의 평균

$$\frac{174 + 156}{2}$$

Mean Encoding보다 더 다양한 값을 각 행에 할당

THANK YOU

클린업 끝~~
주분에서 만나요…