

# Classifier 성능 평가지표

📌 과제	필수
# 주차	3

Classifier의 성능을 평가하고자 한다면, Confusion Matrix를 기반으로 측정되는 **Accuracy**, **Precision**, **Recall**, **F1 score** 등을 활용할 수 있음

## ▼ Table of Contents

[Confusion Matrix](#)

[Accuracy](#)

[Precision](#)

[Recall](#)

[F1 score](#)

[ROC, Receiver Operating Characteristic](#)

[AUC, Area Under Curve](#)

[AIC, Akaike Information Criterion](#)

[Exercise5](#)

## Confusion Matrix

actual \ predicted	0	1
0	TN	FP
1	FN	TP

- T, True: 예측이 정답
- F, False: 예측이 오답
- P, Positive: 모델은 positive라고 예측
- N, Negative: 모델은 negative라고 예측

## Accuracy

모델이 정답을 맞춘 비율, 1에 가까울수록 정답을 많이 맞춘 것

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

하지만 데이터가 불균형할 때는 Accuracy만으로 모델의 성능을 평가하기 어려움: ex.

positive:negative = 9:1일 때 어떤 값이든 positive로만 예측하는 모델을 Accuracy만으로 평가한다면 성능이 좋다고 평가하게 될 것

## Precision

모델이 positive라고 예측한 것들 중 정답을 맞춘 비율, 1에 가까울수록 정답을 많이 맞춘 것

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

Precision은 FP를 낮추는 것이 중요하기 때문에, 정답이 negative인 데이터를 positive라고 잘못 예측하면 안되는 경우에는 중요한 지표가 될 수 있음: ex. 스팸검출

## Recall

정답이 positive인 것들 중 모델이 positive라고 예측한 비율, 1에 가까울수록 정답을 많이 맞춘 것

$$Recall = \frac{TP}{TP+FN}$$

Recall은 FN을 낮추는 것이 중요하기 때문에, 정답이 positive인 데이터를 negative라고 잘못 예측하면 안되는 경우에는 중요한 지표가 될 수 있음: ex. 암 검출, 금융사기 검출

## F1 score

Precision과 Recall의 조화평균, 1에 가까울수록 모델의 성능이 좋은 것

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Precision과 Recall이 상호보완적인 평가지표임을 활용한 것!

둘의 trade-off를 결정하는 수치는 **Threshold**:

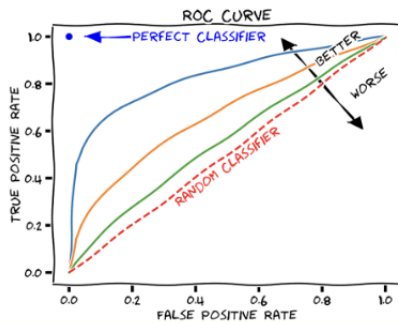
- Threshold는 positive 예측값을 결정하는 확률로, 보통 이진분류의 경우 확률이 0.5 초과이면 positive로 0.5 이하이면 negative로 예측하는데 이때 Threshold는 0.5
- Threshold를 낮추면: positive로 예측하는 값이 많아지면서 Precision 감소 Recall 증가
- Threshold를 높이면: positive로 예측하는 값이 적어지면서 Precision 증가 Recall 감소

## ROC, Receiver Operating Characteristic

ROC curve는 **FPR False Positive Rate**이 변할 때 **TPR True Positive Rate**이 어떻게 변하는지 나타내는 곡선, FPR을 x축으로 TPR을 y축으로 함

- $TPR = Recall$
- $FPR = 1 - TNR = 1 - \frac{TN}{FP+TN} = \frac{FP}{FP+TN}$

랜덤 수준 이진분류에서의 ROC curve를 나타낸 직선에서 멀어질수록 성능이 뛰어난 것:



Threshold를 이용해 FPR 값을 변경시킬 수 있음:

- FPR이 1이려면, TN을 0으로  
negative로 예측한 게 없으면 되므로, Threshold = 0
- FPR이 0이려면, FP를 0으로  
positive로 예측한 게 없으면 되므로, Threshold = 1

```
from sklearn.metrics import roc_curve

#label이 1일 때 예측확률 추출
pred_pos_label = model.predict_proba(x_test)[: , 1]

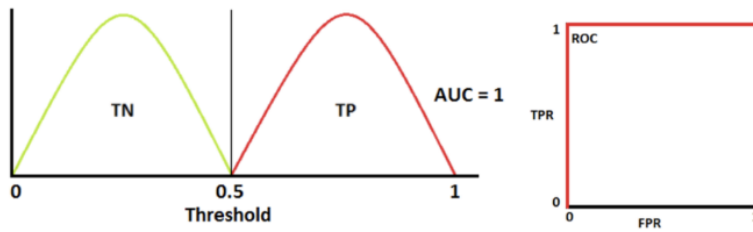
#roc_curve는 실제값, 예측확률값을 넣으면 FPR, TPR, Threshold를 return
fprs, tprs, thresholds = roc_curve(y_test, pred_pos_label)

import matplotlib.pyplot as plt
plt.figure(figsize=(15, 5))
plt.plot([0, 1], [0, 1], label='STR') #random classifier 직선
plt.plot(fprs, tprs, label='ROC')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid()
plt.show()
```

## AUC, Area Under Curve

ROC curve의 하단 면적, 1에 가까울수록 모형의 적합도가 높은 것

TN과 TP 두개의 곡선이 전혀 겹치지 않는 경우 이상적인 분류 성능을 보임:



## AIC, Akaike Information Criterion

최대 우도에 독립변수의 개수에 대한 penalty를 반영하는 목적으로 모형과 데이터의 확률분포 차이를 측정한 것, 값이 낮을수록 모형의 적합도가 높은 것

$$AIC = 2k - 2 \ln(\hat{L})$$

$k$ : number of estimated parameters in the model

$\hat{L}$ : maximum value of the likelihood function for the model

- 우도/가능도 likelihood: 고정된 관측값이 어떠한 확률분포에서 어느정도의 확률로 나타나는지에 대한 확률

### ▼ Exercise5

#### ✓ CASE 1. Class 불균형인 경우

- Confusion Matrix

actual \ predicted	0	1
0	TN = 16	FP = 9
1	FN = 3	TP = 2

- Accuracy

$$Accuracy = \frac{16+2}{16+9+3+2} = \frac{18}{30} = \frac{3}{5}$$

- Precision

$$Precision = \frac{2}{9+2} = \frac{2}{11}$$

- Recall

$$Recall = \frac{2}{3+2} = \frac{2}{5}$$

- F1 score

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2}{\frac{1}{\frac{2}{11}} + \frac{1}{\frac{2}{5}}} = \frac{2}{\frac{11}{2} + \frac{5}{2}} = \frac{2}{8} = \frac{1}{4}$$

#### ✓ CASE 2. Class 균형인 경우

- Confusion Matrix

actual \ predicted	0	1
0	TN = 5	FP = 1
1	FN = 3	TP = 3

- Accuracy

$$Accuracy = \frac{5+2}{5+1+3+3} = \frac{8}{12} = \frac{2}{3}$$

- Precision

$$Precision = \frac{3}{1+3} = \frac{3}{4}$$

- Recall

$$Recall = \frac{3}{3+3} = \frac{1}{2}$$

- F1 score

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2}{\frac{4}{3} + 2} = \frac{6}{10} = \frac{3}{5}$$

---

출처

- <https://white-joy.tistory.com/9>
- <https://driip.me/3ef36050-f5a3-41ea-9f23-874afe665342>
- <https://hajunyoo.oopy.io/c9d4cfe8-ef80-4473-80e3-3a96a34db19b>