

Model measurement

PinkLAB Edu

4 November, 2025

Table of Contents

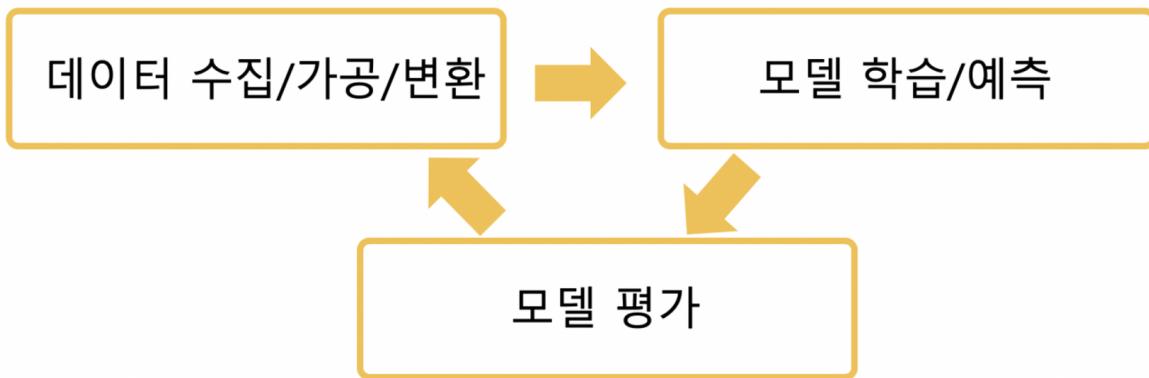
1 모델 평가의 개념	4
1.1 여전히 우리가 만든 모델은 얼마나 좋은 것일까?	4
1.2 회귀모델들은 실제 값과의 에러치를 가지고 계산	4
1.3 분류 모델의 평가 항목이 조금 많음	5
1.4 이진 분류 모델의 평가.....	5
1.5 이진 분류에서 predict vs real	6
1.6 그러면 더 이해가 잘 될까요?.....	6
1.7 Accuracy.....	7
1.8 Precision.....	7
1.9 RECALL (TPR TRUE POSITIVE RATIO).....	7
1.10 FALL-OUT (FPR FALSE POSITION RATIO).....	8
1.11 분류모델은 그 결과를 속할 비율(확률)을 반환한다.....	8
1.12 그 threshold를 변경해 가면서 모델 평가 지표들을 관찰해 보자	8
1.13 Threshold = 0.3	9
1.14 Threshold = 0.4	9
1.15 Threshold = 0.5	10
1.16 Threshold = 0.6	10
1.17 Threshold = 0.8	11
1.18 Threshold = 0.9	11
1.19 정리~.....	12
1.20 F1-Score.....	12
1.21 F1-Score의 계산	13
2 ROC와 AUC	14
2.1 ROC 곡선	14
2.2 만약 완벽하게 분류했다면	14
2.3 적당히 잘 했다면	15
2.4 분류 성능이 나쁘다면	15
2.5 방금 전 예제에서의 ROC 곡선.....	16
2.6 AUC	16

3 ROC 커브 그려보기	17
3.1 다시 데이터 읽어보기	17
3.2 간단히 결정나무 적용해보기	17
3.3 각 수치 구해보기	18
3.4 ROC 커브 그리기	18
3.5 ROC curve	19

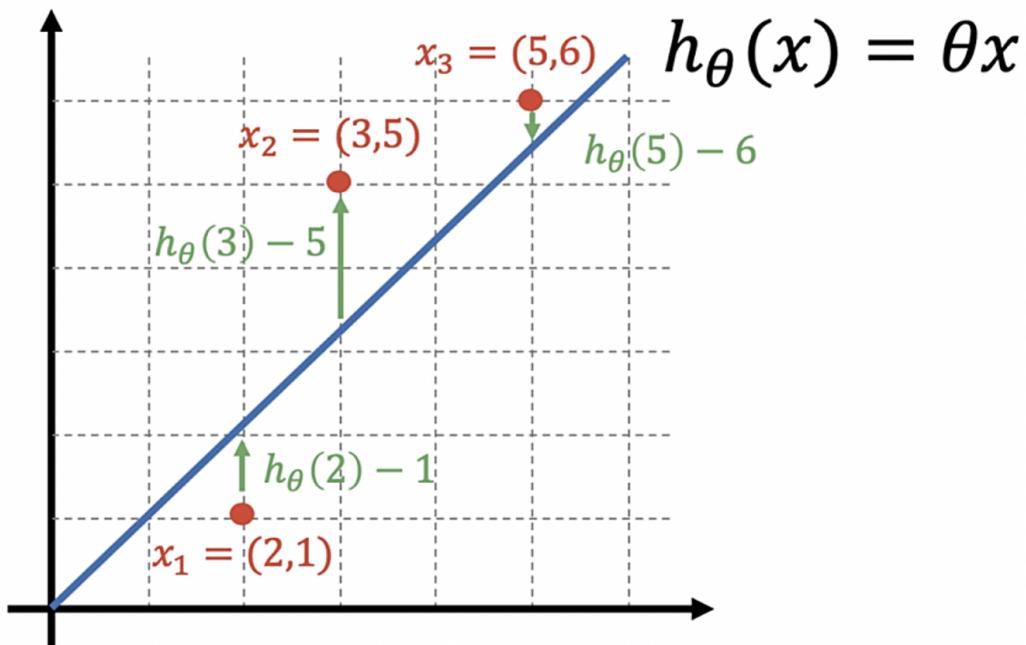
1 모델 평가의 개념

1.1 여전히 우리가 만든 모델은 얼마나 좋은 것일까?

- 모델을 좋다, SOSO, 나쁘다 등으로 평가할 방법은 없다
- 대부분 다양한 모델, 다양한 파라미터를 두고, 상대적으로 비교한다



1.2 회귀모델들은 실제 값과의 에러치를 가지고 계산

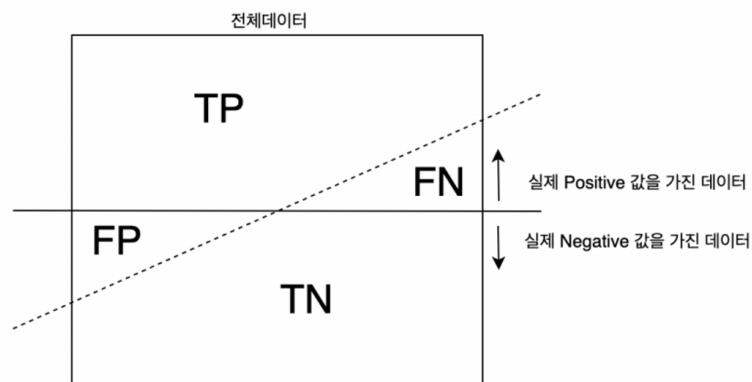


- 사실 회귀모델의 평가는 간단한 편이다

1.3 분류 모델의 평가 항목이 조금 많음

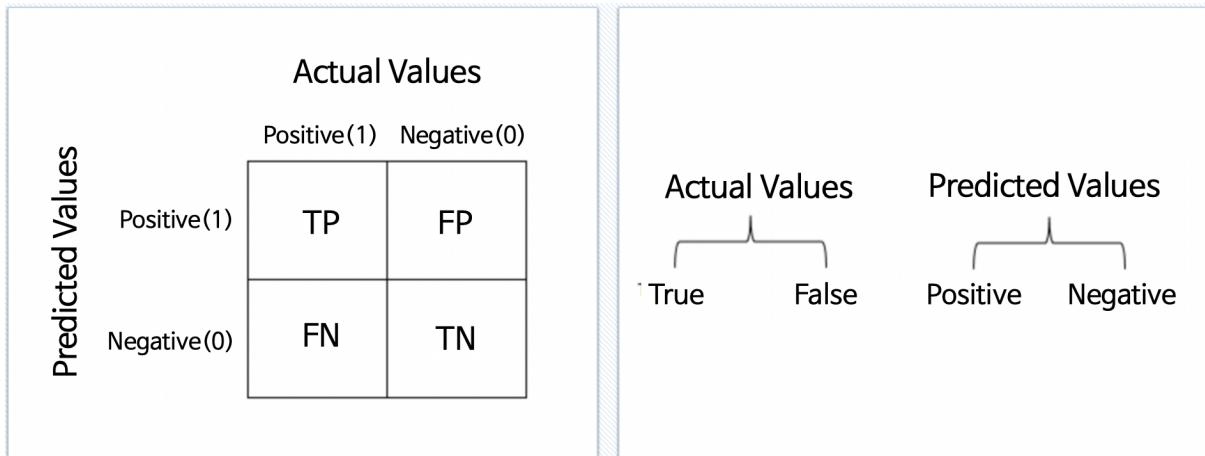
정확도	Accuracy
오차행렬	Confusion Matrix
정밀도	Precision
재현율	Recall
F1 score	-
ROC AUC	-

1.4 이진 분류 모델의 평가

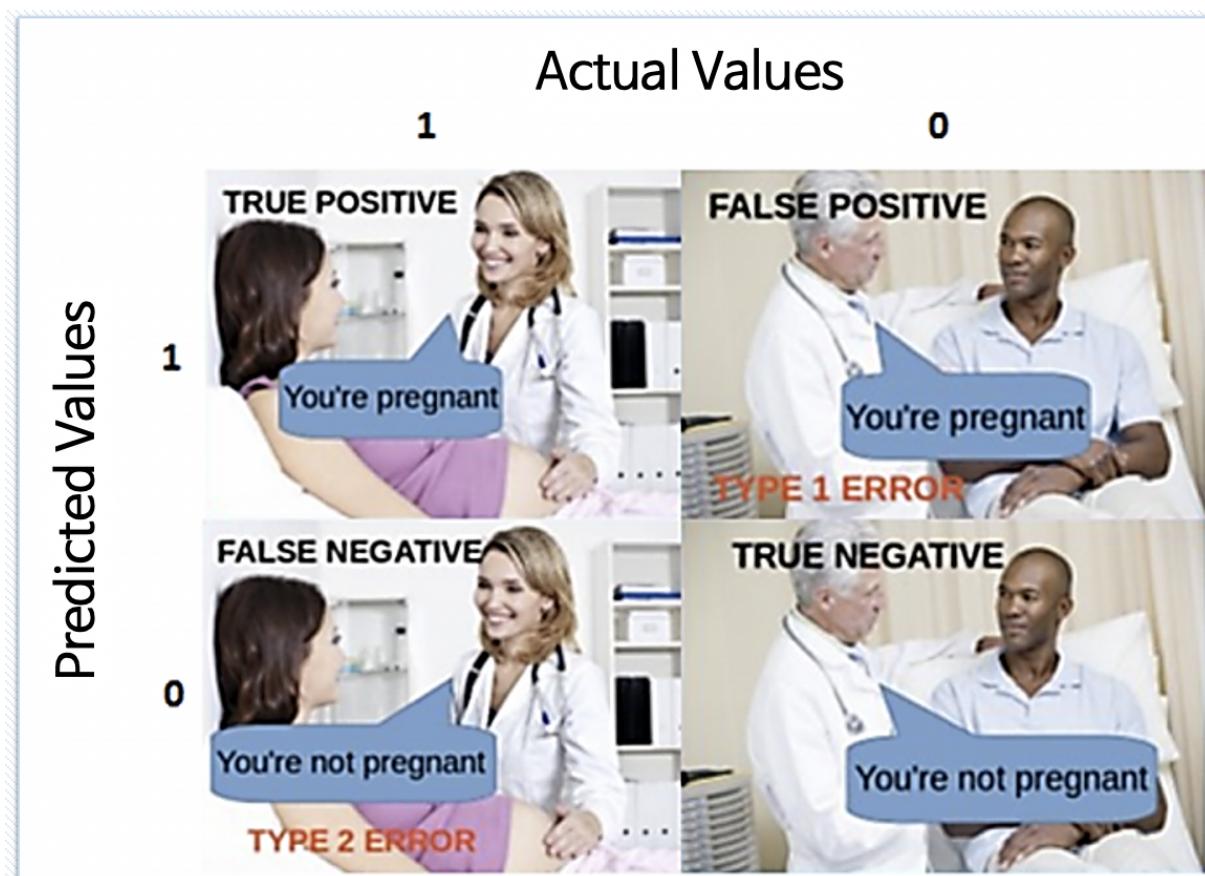


TP True Positive	실제 Positive를 Positive라고 맞춘 경우
FN False Negative	실제 Positive를 Negative라고 틀리게 예측한 경우
TN True Negative	실제 Negative를 Negative라고 맞춘 경우
FP False Positive	실제 Negative를 Positive라고 틀리게 예측한 경우

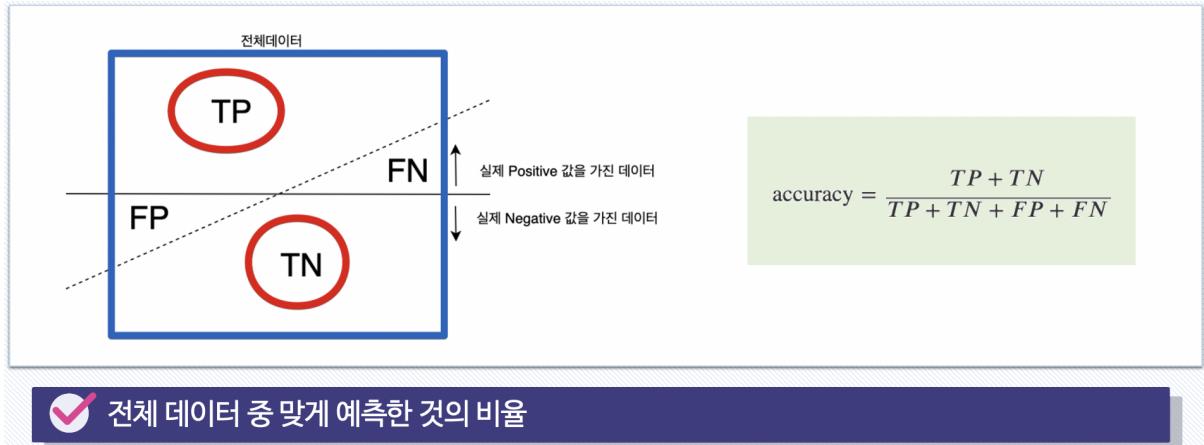
1.5 이진 분류에서 predict vs real



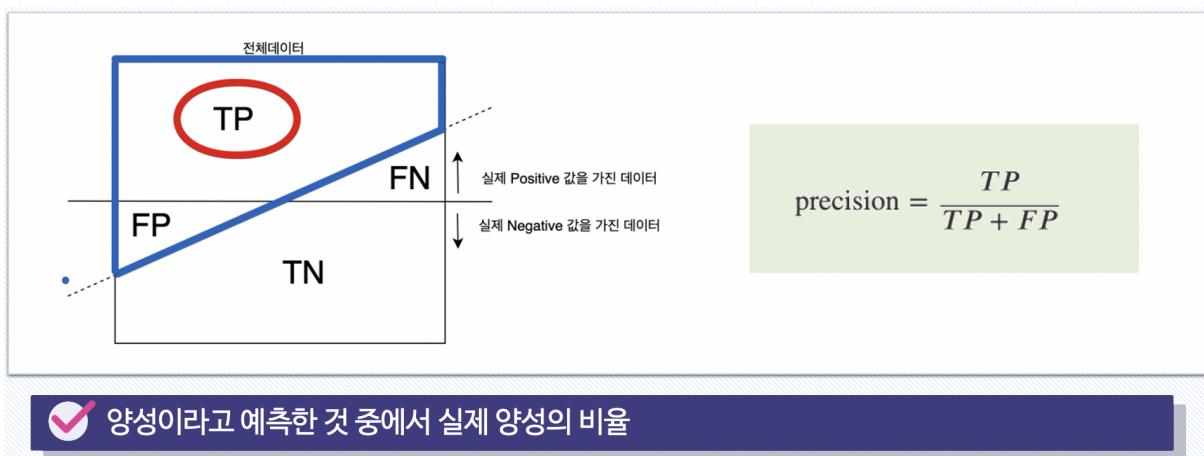
1.6 그러면 더 이해가 잘 될까요?



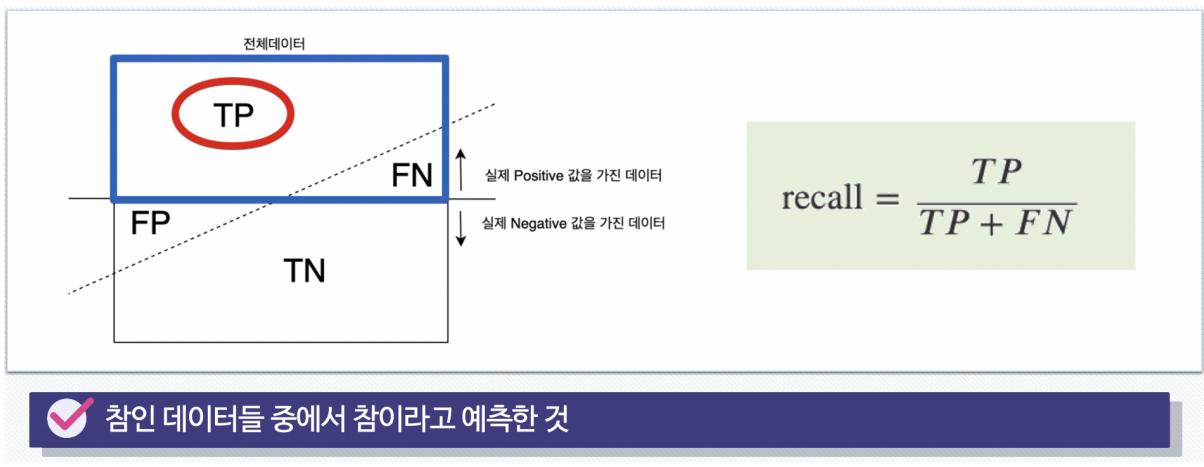
1.7 Accuracy



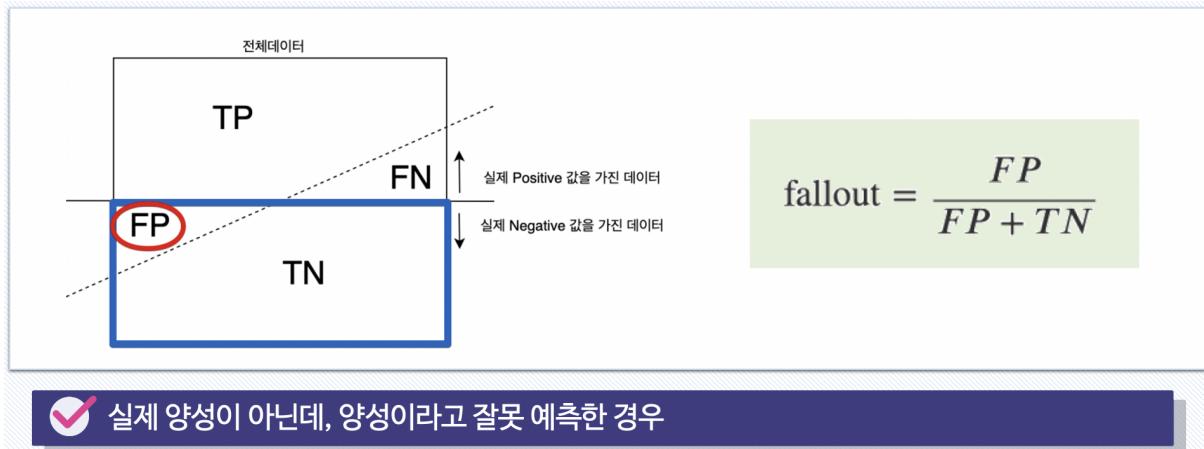
1.8 Precision



1.9 RECALL (TPR TRUE POSITIVE RATIO)



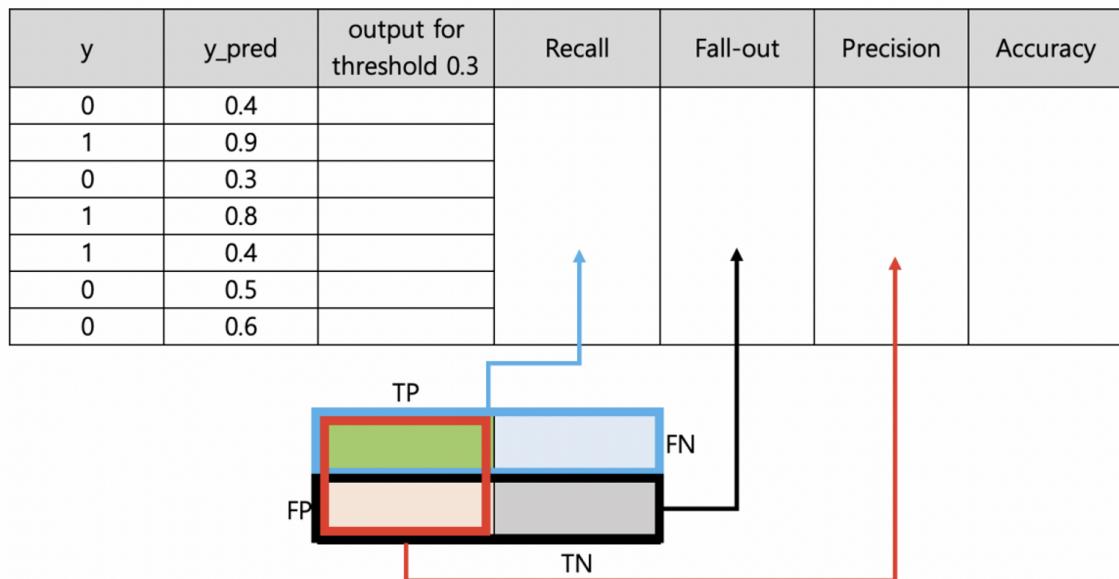
1.10 FALL-OUT (FPR FALSE POSITION RATIO)



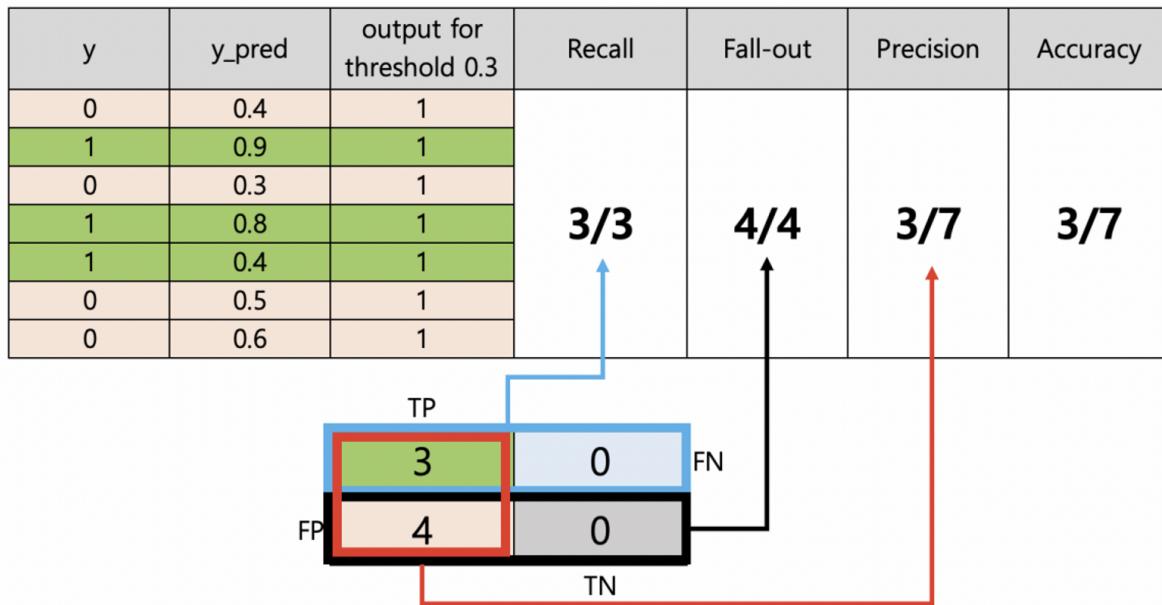
1.11 분류모델은 그 결과를 속할 비율(확률)을 반환한다

- 지금까지는 그 비율에서 threshold를 0.5라고 하고 0, 1로 결과를 반영했다 (if 이진분류)
- iris의 경우 가장 높은 확률값이 있는 클래스를 해당 값이라고 했다

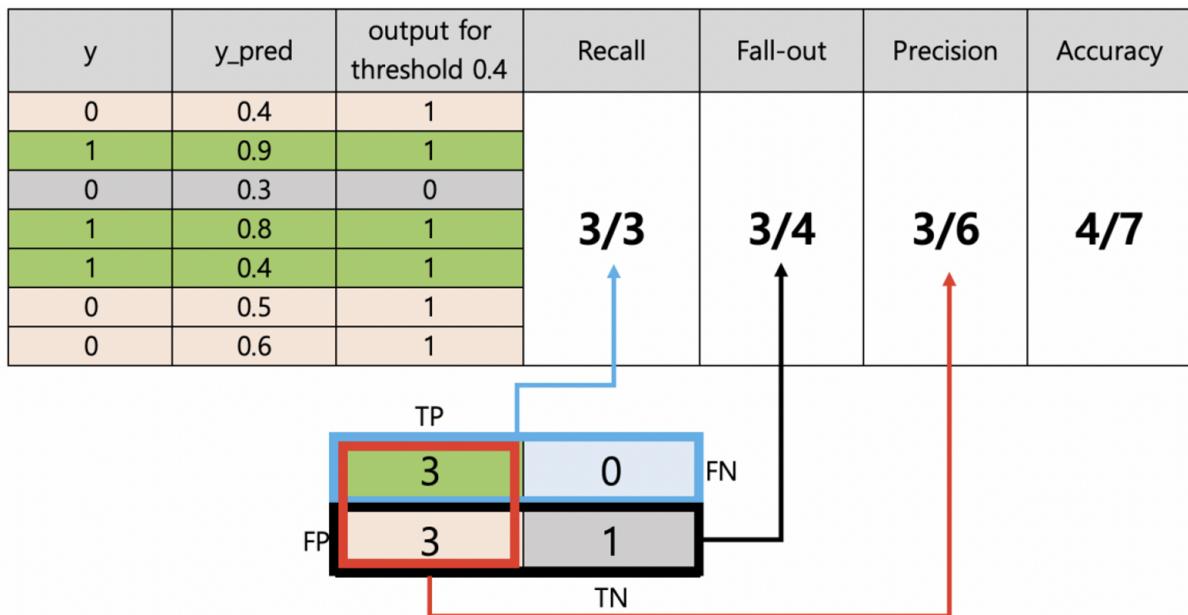
1.12 그 threshold를 변경해 가면서 모델 평가 지표들을 관찰해 보자



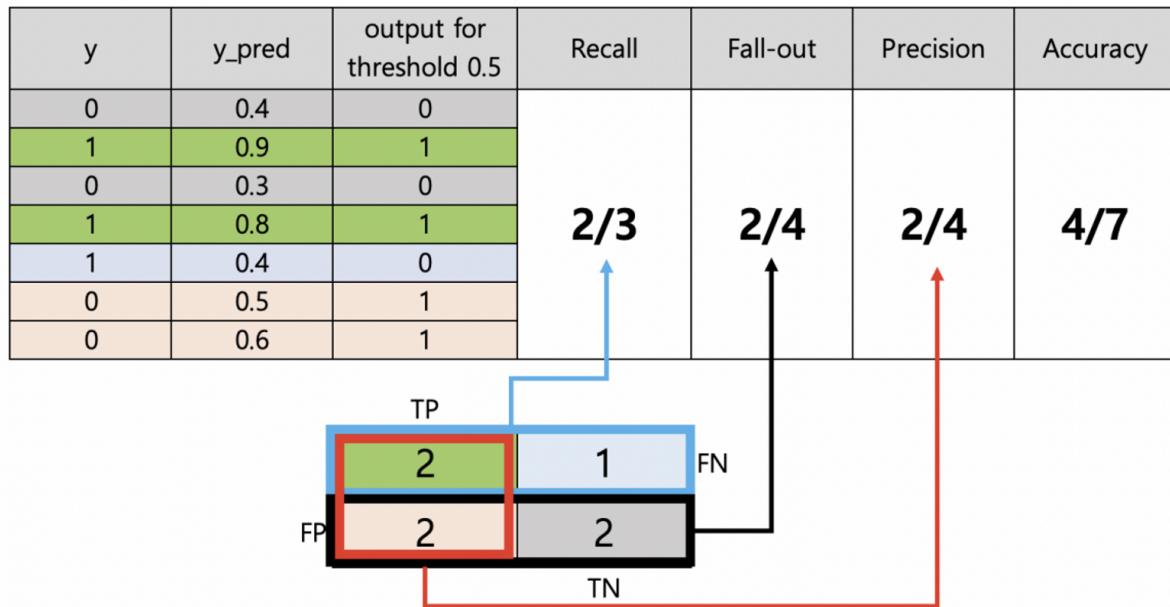
1.13 Threshold = 0.3



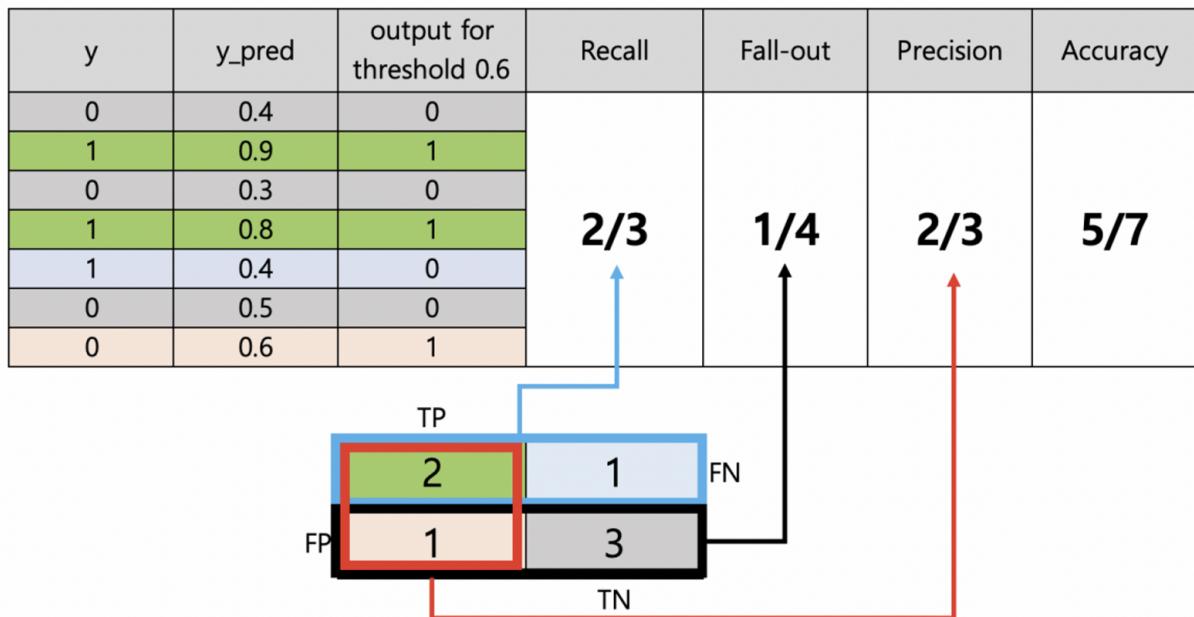
1.14 Threshold = 0.4



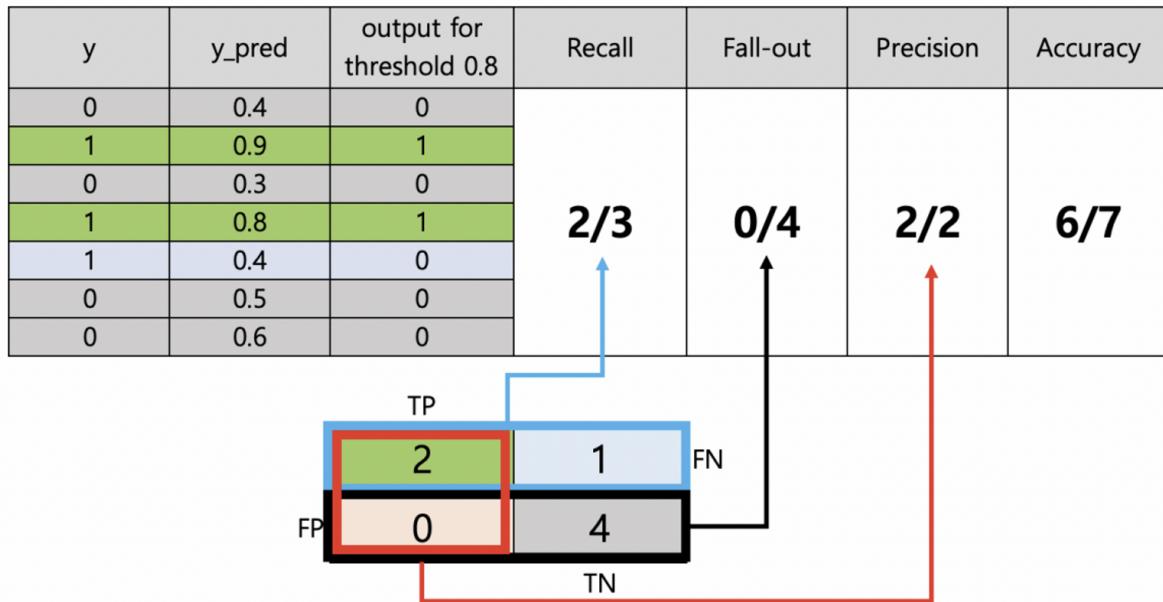
1.15 Threshold = 0.5



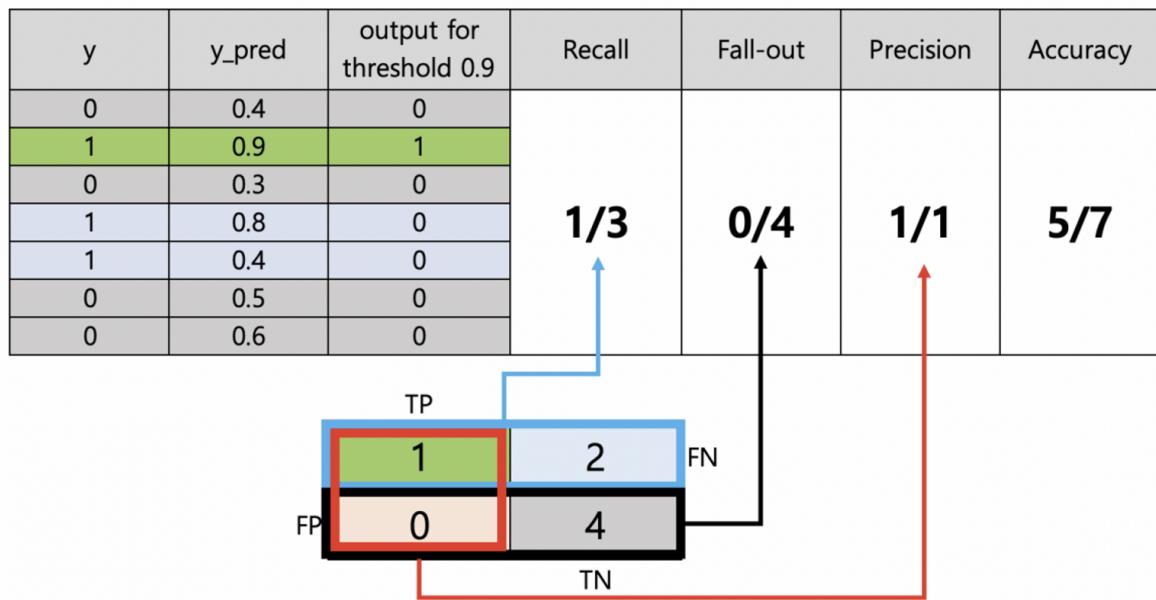
1.16 Threshold = 0.6



1.17 Threshold = 0.8



1.18 Threshold = 0.9



1.19 정리~

	Threshold					
	0.3	0.4	0.5	0.6	0.8	0.9
Recall	1.00	1.00	0.67	0.67	0.67	0.33
Precision	0.43	0.50	0.50	0.67	1.00	1.00
Fall-out	1.00	0.75	0.50	0.25	0.00	0.00
Accuracy	0.43	0.57	0.57	0.71	0.86	0.71

- Recall은 참인 데이터 중에서 참이라고 예측한 데이터의 비율
- Precision은 참이라고 예측한 것 중에서 실제 참인 데이터의 비율
- 실제 양성인 데이터를 음성이라고 판단하면 안되는 경우라면, Recall이 중요하고, 이 경우는 Threshold를 0.3 혹은 0.4로 선정해야 함 (eg. 암환자 판별)
- 실제 음성인 데이터를 양성이라고 판단하면 안되는 경우라면, Precision이 중요하고, 이 경우는 Threshold를 0.8 혹은 0.9로 적합할 수 있음 (eg. 스팸메일)
- 그러나 Recall과 Precision은 서로 영향을 주기 때문에 한 쪽을 극단적으로 높게 설정해서는 안됨

1.20 F1-Score

$$F_{\beta} = (1 + \beta^2) (\text{precision} \times \text{recall}) / (\beta^2 \text{precision} + \text{recall})$$

→ F-Score에서 beta를 1로 두면 F1-score임

$$F_1 = 2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$$

✓ F1-score는 Recall과 Precision을 결합한 지표

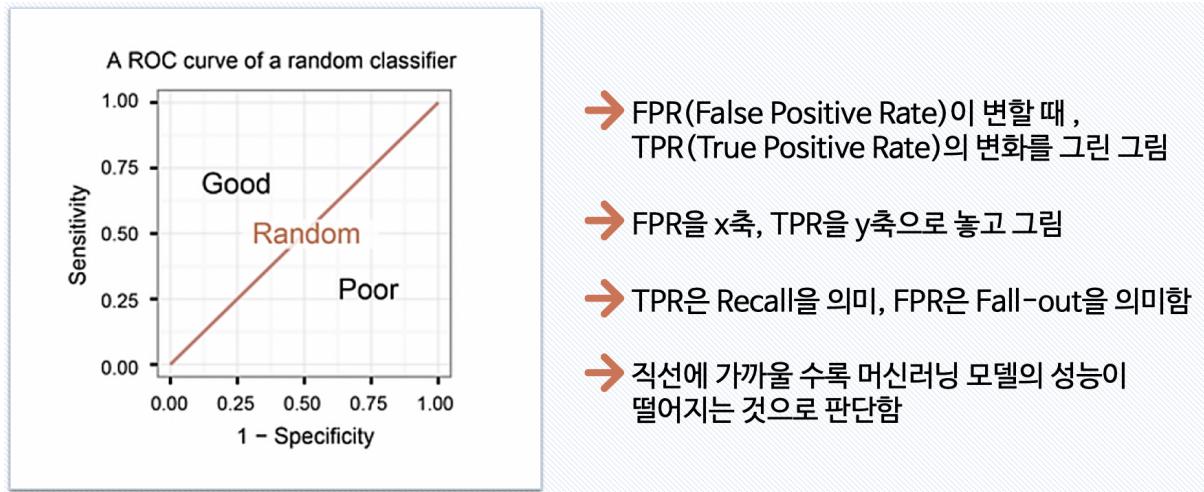
✓ Recall과 Precision이 어느 한쪽으로 치우치지 않고 둘다 높은 값을 가질 수록 높은 값을 가짐

1.21 F1-Score의 계산

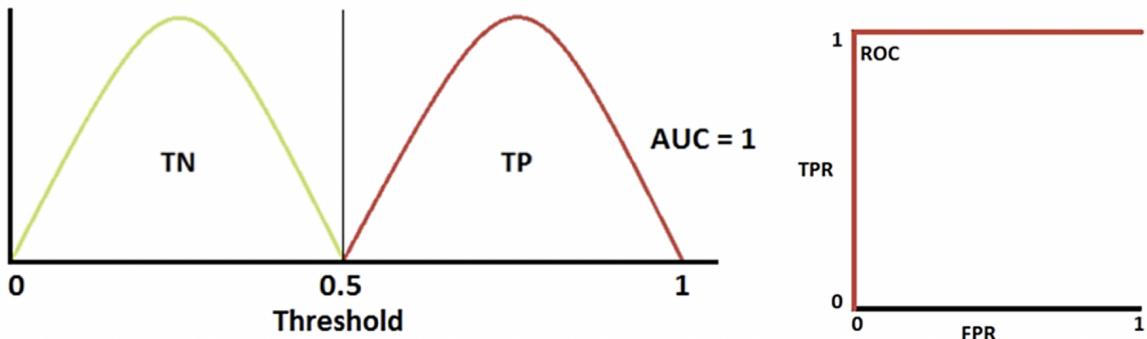
	Threshold					
	0.3	0.4	0.5	0.6	0.8	0.9
Recall	1.00	1.00	0.67	0.67	0.67	0.33
Precision	0.43	0.50	0.50	0.67	1.00	1.00
Fall-out	1.00	0.75	0.50	0.25	0.00	0.00
Accuracy	0.43	0.57	0.57	0.71	0.86	0.71
F1-score	0.60	0.67	0.57	0.67	0.80	0.50

2 ROC와 AUC

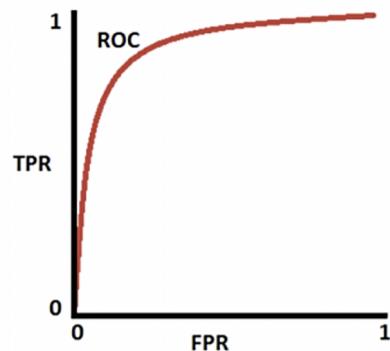
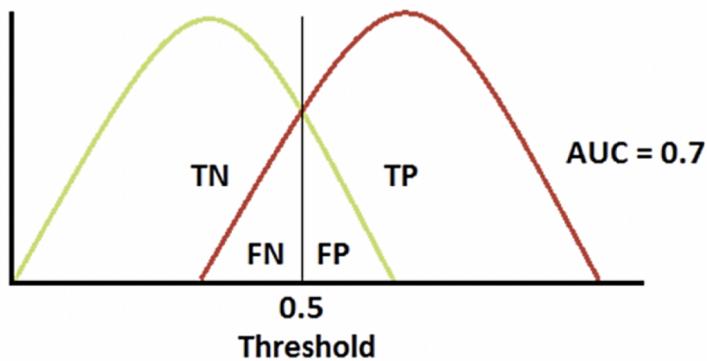
2.1 ROC 곡선



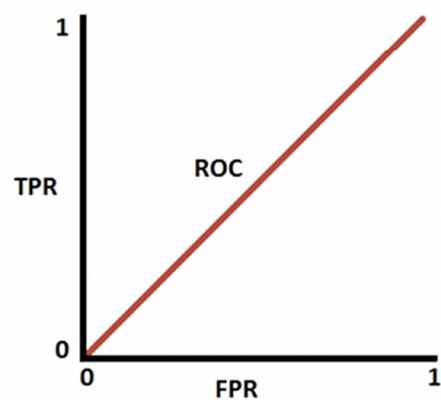
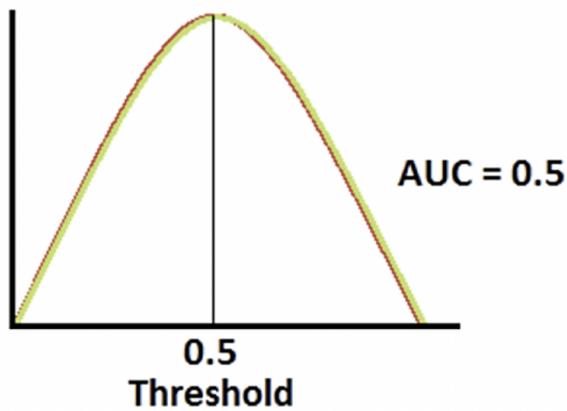
2.2 만약 완벽하게 분류했다면



2.3 적당히 잘 했다면

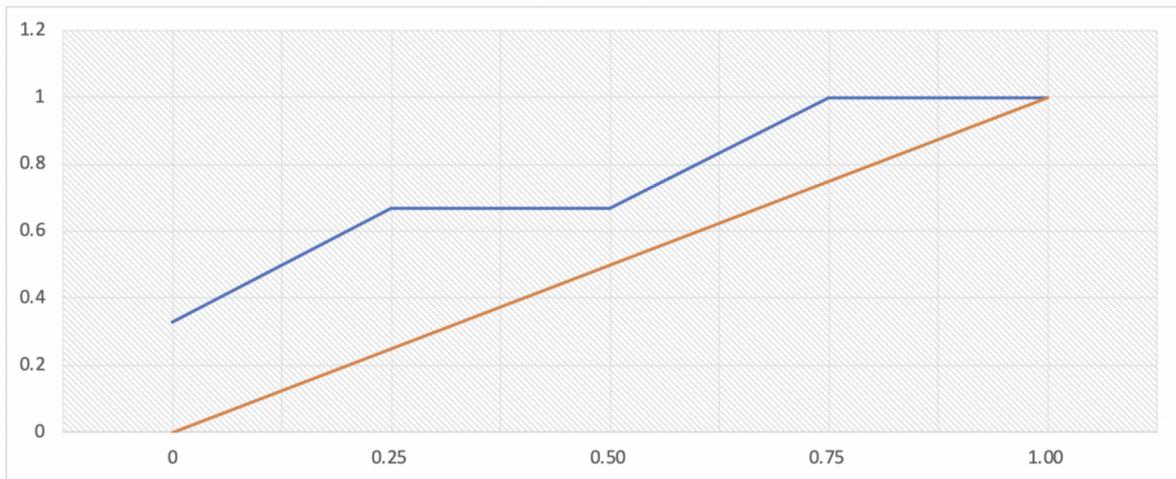


2.4 분류 성능이 나쁘다면

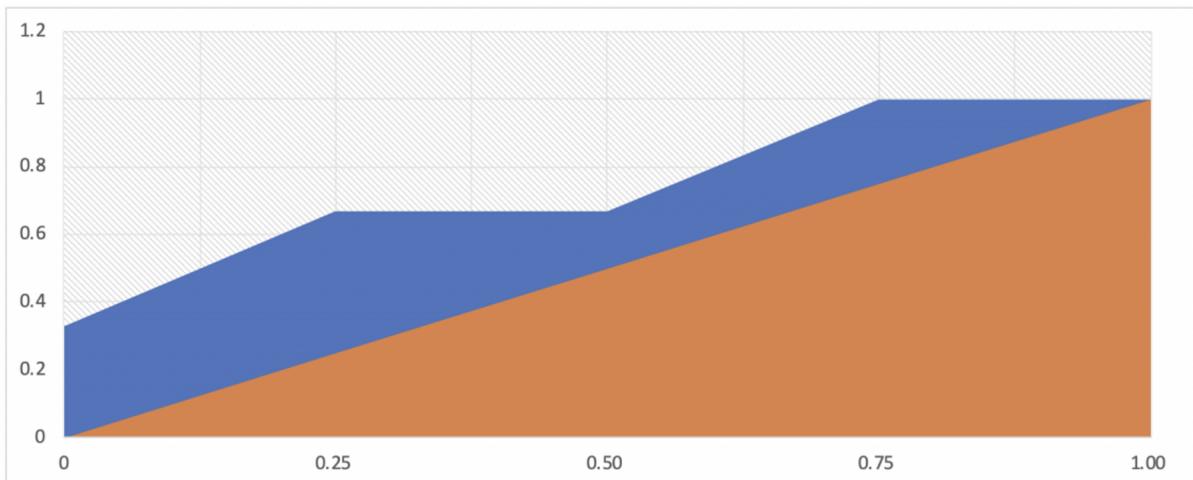


2.5 방금 전 예제에서의 ROC 곡선

Recall	1.00	1.00	0.67	0.67	0.67	0.33
Fall-out	1.00	0.75	0.50	0.25	0.00	0.00



2.6 AUC



- ROC 곡선 아래의 면적
- 일반적으로 1에 가까울 수록 좋은 수치
- 기울기가 1인 직선 아래의 면적이 0.5 → AUC는 0.5보다 커야 함

3 ROC 커브 그려보기

3.1 다시 데이터 읽어보기

```
import pandas as pd

red_url = 'https://raw.githubusercontent.com/PinkWink/ML_tutorial' +\
           '/master/dataset/winequality-red.csv'
white_url = 'https://raw.githubusercontent.com/PinkWink/ML_tutorial' +\
           '/master/dataset/winequality-white.csv'

red_wine = pd.read_csv(red_url, sep = ';')
white_wine = pd.read_csv(white_url, sep = ";")

red_wine['color'] = 1.
white_wine['color']= 0.

wine = pd.concat([red_wine, white_wine])
wine['taste'] = [1. if grade > 5 else 0. for grade in wine['quality']]

X = wine.drop(['taste', 'quality'], axis=1)
y = wine['taste']

✓ 1.3s
```

Python

3.2 간단히 결정나무 적용해보기

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
                                                    random_state = 13)

wine_tree = DecisionTreeClassifier(max_depth = 2 , random_state = 13)
wine_tree.fit(X_train, y_train)

y_pred_tr = wine_tree.predict(X_train)
y_pred_test = wine_tree.predict(X_test)

print('Train Acc : ' , accuracy_score(y_train, y_pred_tr))
print("Test Acc : " , accuracy_score(y_test, y_pred_test))

✓ 0.7s
```

Python

Train Acc : 0.7294593034442948
 Test Acc : 0.7161538461538461

3.3 각 수치 구해보기

```
from sklearn.metrics import (accuracy_score, precision_score, recall_score,
| | | | | | | | | f1_score, roc_auc_score, roc_curve)

print('Accuracy : ', accuracy_score(y_test, y_pred_test))
print('Recall : ', recall_score(y_test, y_pred_test))
print('Precision : ', precision_score(y_test, y_pred_test))
print('AUC Score : ', roc_auc_score(y_test, y_pred_test))
print('F1 Score : ', f1_score(y_test, y_pred_test))
```

✓ 0.0s

Python

Accuracy : 0.7161538461538461
 Recall : 0.7314702308626975
 Precision : 0.8026666666666666
 AUC Score : 0.7105988470875331
 F1 Score : 0.7654164017800381

3.4 ROC 커브 그리기

```
import matplotlib.pyplot as plt

pred_proba = wine_tree.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, pred_proba)

plt.figure(figsize=(10,8))
plt.plot([0,1],[0,1])
plt.plot(fpr, tpr)
plt.grid()
plt.show()
```

✓ 0.4s

Python

3.5 ROC curve

