zero-base /

Life Changing Education

Chapter 16. Precision and Recall

정밀도와 재현율의 트레이드오프

zero-base /

다시 와인데이터

```
import pandas as pd

wine_url = 'https://raw.githubusercontent.com/PinkWink/ML_tutorial/'+\
                                                'master/dataset/wine.csv'
wine = pd.read_csv(wine_url, index_col=0)
wine['taste'] = [1. if grade>5 else 0. for grade in wine['quality']]

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

zero-base /

데이터 분리

```
from sklearn.model_selection import train_test_split

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

zero-base /

간단한 로지스틱 회귀 적용

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

lr = LogisticRegression(solver='liblinear', random_state=13)
lr.fit(X_train, y_train)

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

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

```
Train Acc :  0.7425437752549547
Test Acc :  0.7438461538461538
```

zero-base /

classification_report

```
from sklearn.metrics import classification_report

print(classification_report(y_test, lr.predict(X_test)))
```

```
              precision    recall  f1-score   support

         0.0       0.68      0.58      0.62       477
         1.0       0.77      0.84      0.81       823

    accuracy                           0.74      1300
   macro avg       0.73      0.71      0.71      1300
weighted avg       0.74      0.74      0.74      1300
```

zero-base /

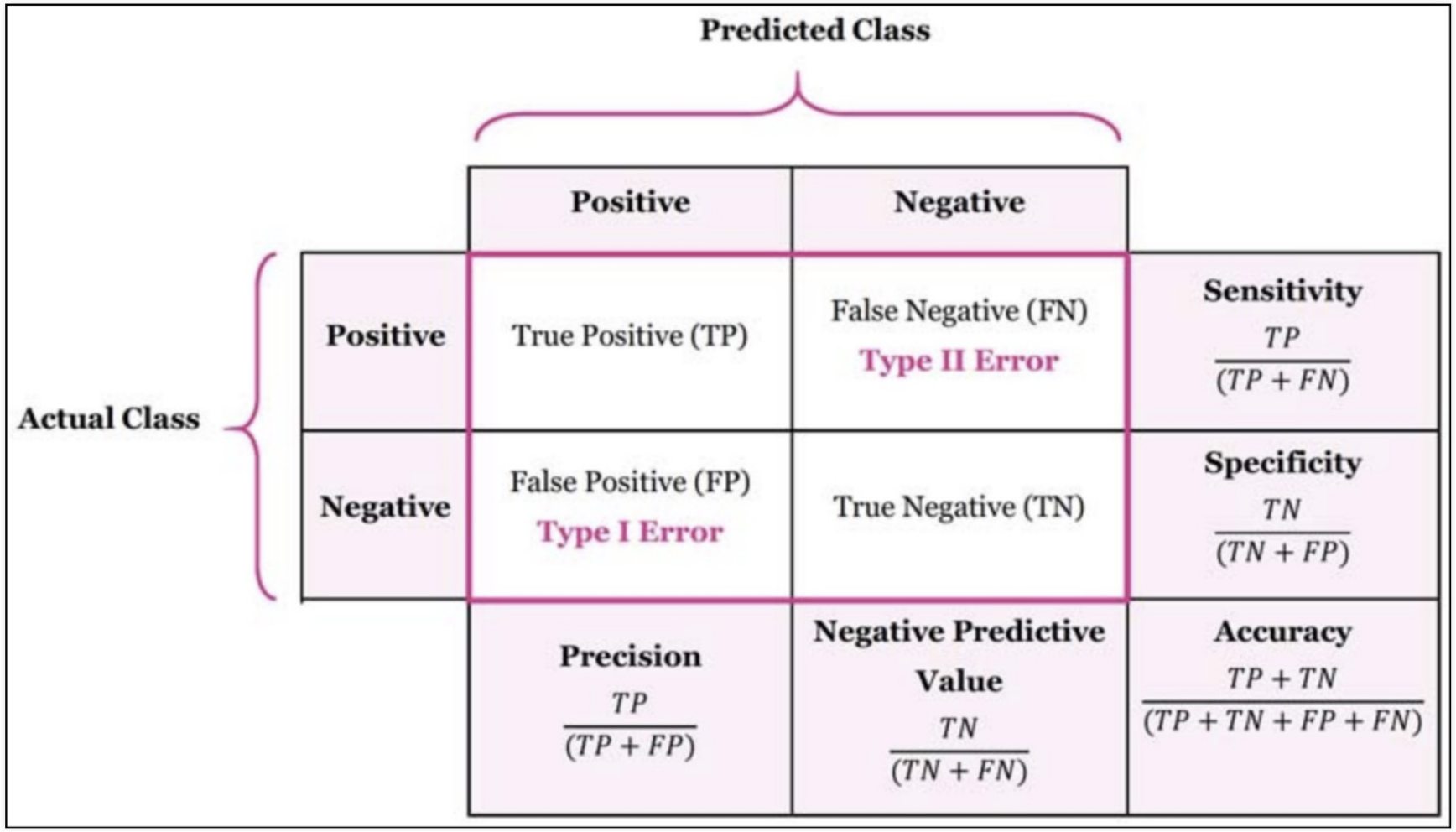
confusion matrix

```
from sklearn.metrics import confusion_matrix

confusion_matrix(y_test, lr.predict(X_test))
```

```
array([[275, 202],
       [131, 692]])
```

정밀도와 재현율의 트레이드오프



zero-base /

precision_recall curve

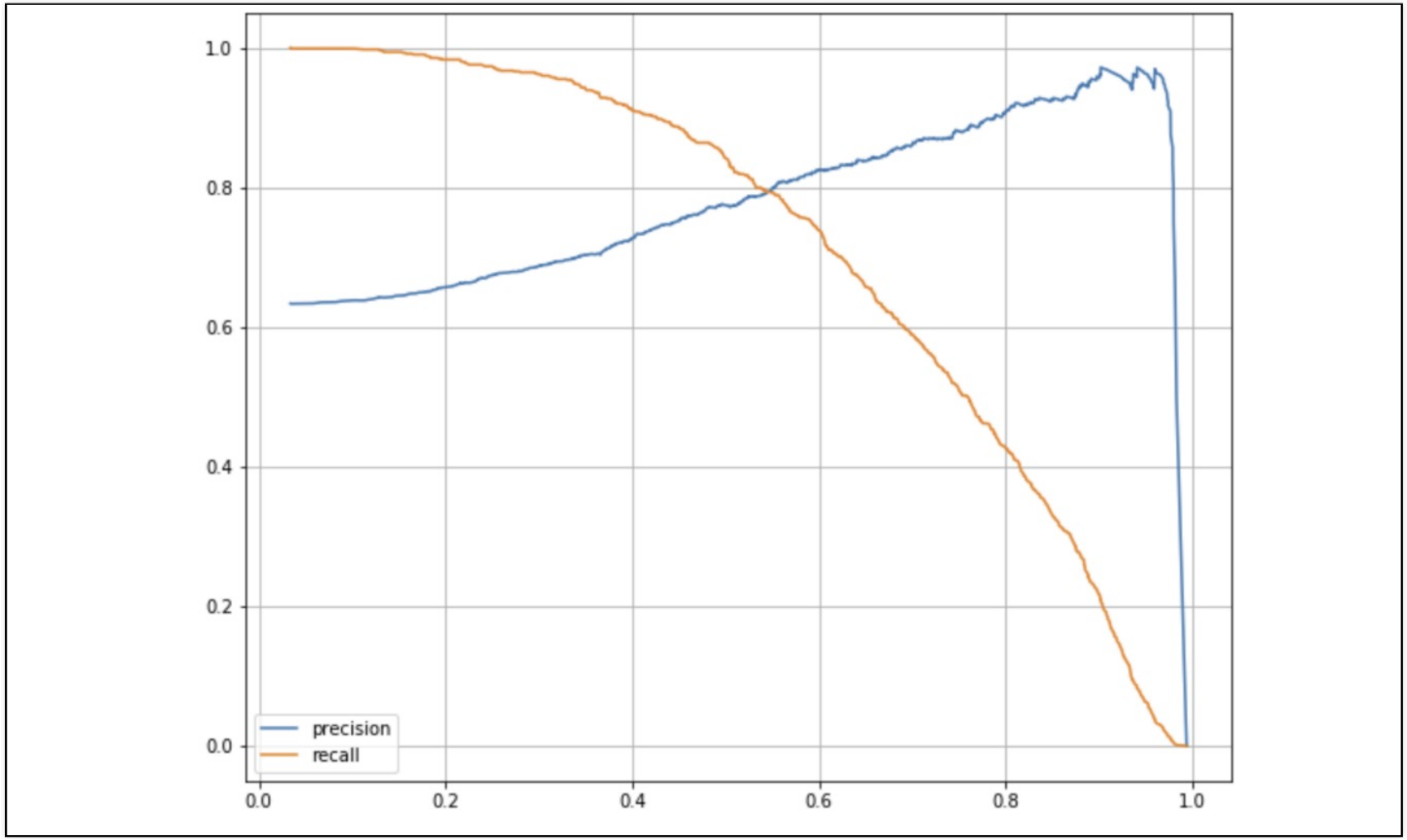
```
import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_curve
%matplotlib inline

plt.figure(figsize=(10,8))
pred = lr.predict_proba(X_test)[:, 1]
precisions, recalls, thresholds = precision_recall_curve(y_test, pred)
plt.plot(thresholds, precisions[:len(thresholds)], label="precision")
plt.plot(thresholds, recalls[:len(thresholds)], label="recall")
plt.grid(); plt.legend(); plt.show()
```

Chapter 16

정밀도와 재현율의 트레이드오프

zero-base /



zero-base /

threshlod = 0.5

```
pred_proba = lr.predict_proba(X_test)
pred_proba[:3]
```

```
array([[0.40526122, 0.59473878],
       [0.50960227, 0.49039773],
       [0.10217325, 0.89782675]])
```

zero-base /

간단히 확인해보기

```
import numpy as np

np.concatenate([pred_proba, y_pred_test.reshape(-1,1)], axis=1)
```

```
array([[0.40526122, 0.59473878, 1.        ],
       [0.50960227, 0.49039773, 0.        ],
       [0.10217325, 0.89782675, 1.        ],
       ...,
       [0.22545775, 0.77454225, 1.        ],
       [0.67370045, 0.32629955, 0.        ],
       [0.31439642, 0.68560358, 1.        ]])
```

zero-base /

threshold 바꿔 보기 - Binarizer

```
from sklearn.preprocessing import Binarizer

binarizer = Binarizer(threshold=0.6).fit(pred_proba)
pred_bin = binarizer.transform(pred_proba)[:, 1]
pred_bin
```

```
array([0., 0., 1., ..., 1., 0., 1.])
```

zero-base /

다시 classification report

```
print(classification_report(y_test, pred_bin))
```

```
              precision    recall  f1-score   support

         0.0       0.62      0.73      0.67       477
         1.0       0.82      0.74      0.78       823

    accuracy                           0.73      1300
   macro avg       0.72      0.73      0.72      1300
weighted avg       0.75      0.73      0.74      1300
```

zero-base /

그리고 confusion matrix

```
confusion_matrix(y_test, pred_bin)
```

```
array([[348, 129],
       [216, 607]])
```