분류 분석 I

빅데이터 분석

오차 행렬

- 행렬을 사용해 이진 분류의 예측 오류를 나타내는 지표
- 사이킷런에서는 오차 행렬을 구하기 위해 confusion_matrix 함수를 제공

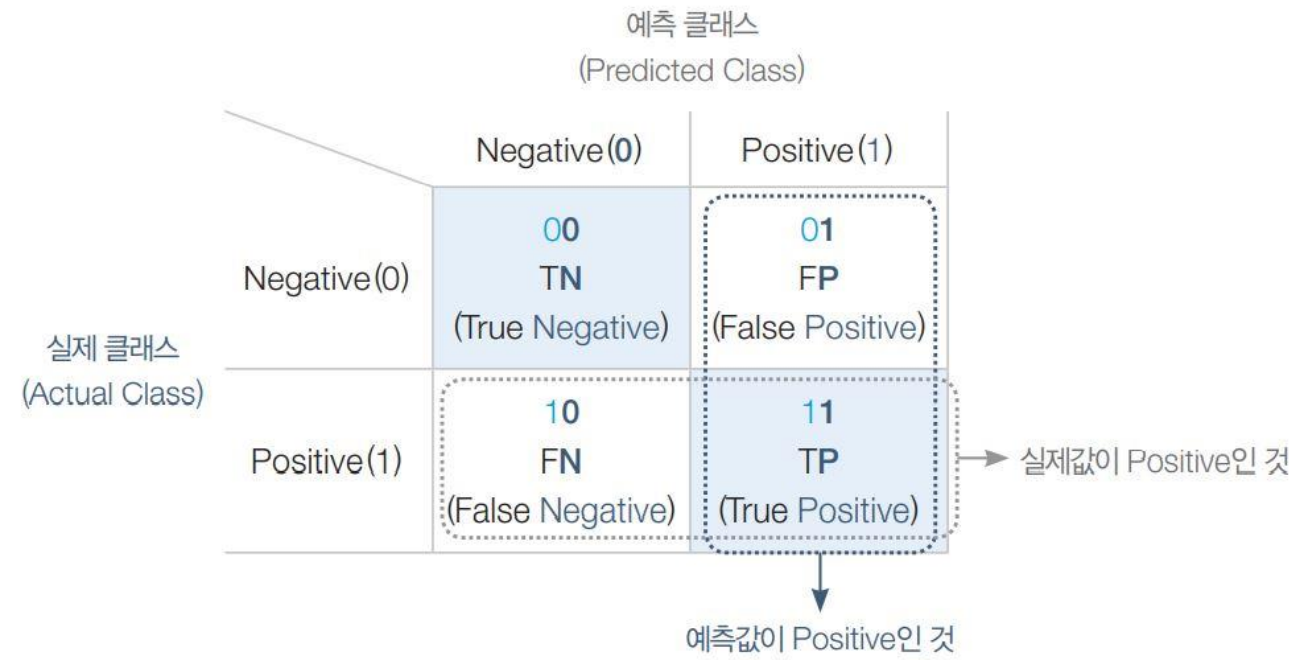


그림 11-2 오차 행렬

- 정확도 $= \frac{\text{예측 결과와 실제값이 동일한 건수}}{\text{전체 데이터 수}} = \frac{(TN+TP)}{(TN+FP+FN+TP)}$

정밀도

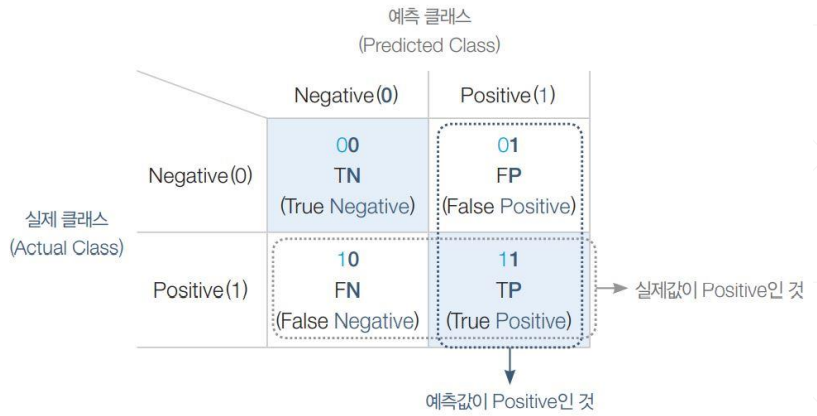


그림 11-2 오차 행렬

① 정밀도(Precision)

- 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율

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

날씨 예측 모델이 맑다고 예측했는데, 실제 날씨가 맑았는지를 살펴보는 지표

재현율

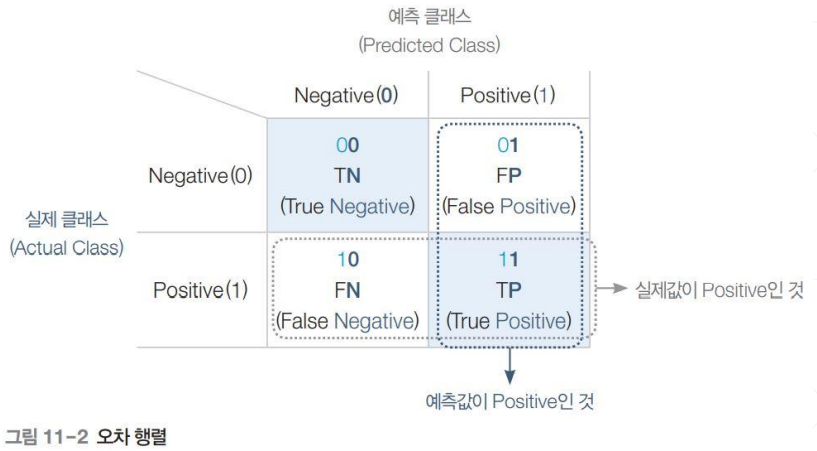


그림 11-2 오차 행렬

② 재현율(Recall)

- 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율

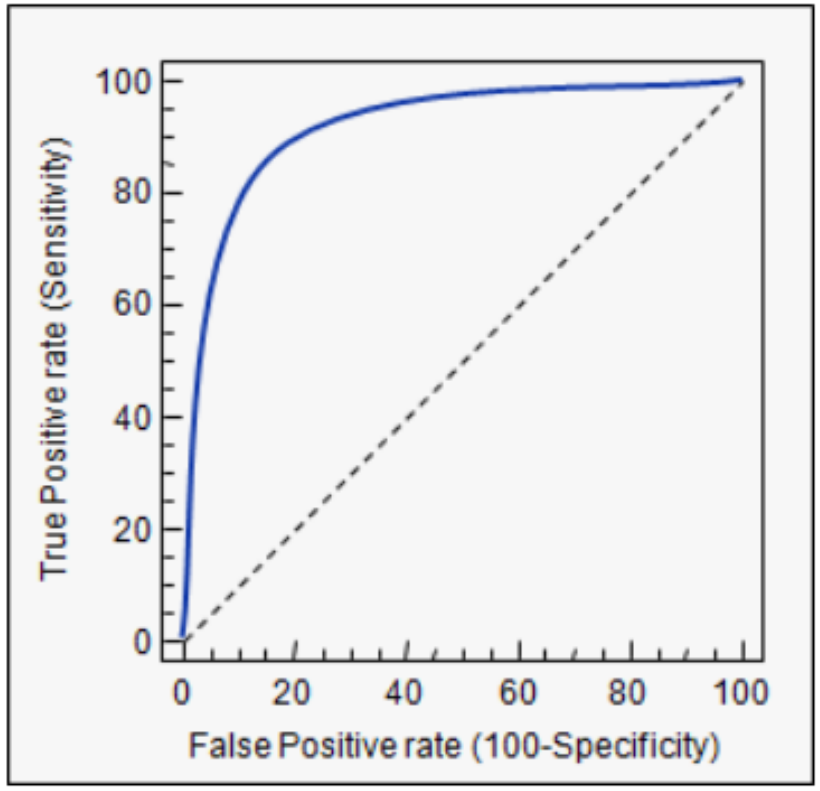
$$(Recall) = \frac{TP}{TP + FN}$$

실제 날씨가 맑은 날 중에서 모델이 맑다고 예측한 비율을 나타낸 지표

Precision은 **모델의 입장**에서, 그리고 Recall은 **실제 데이터의 입장**에서 바라보는 지표

"확실히 맑은 날을 예측할 수 있다면 해당하는 날에만 맑은 날이라고 예측하면 어떨까?"
(예를 들어, 한 달 30일 동안 맑은 날이 20일이었는데, 확실한 2일만 맑다고 예측한다면 이상적인 모델이 될 것인가?)

ROC 기반 AUC 스코어



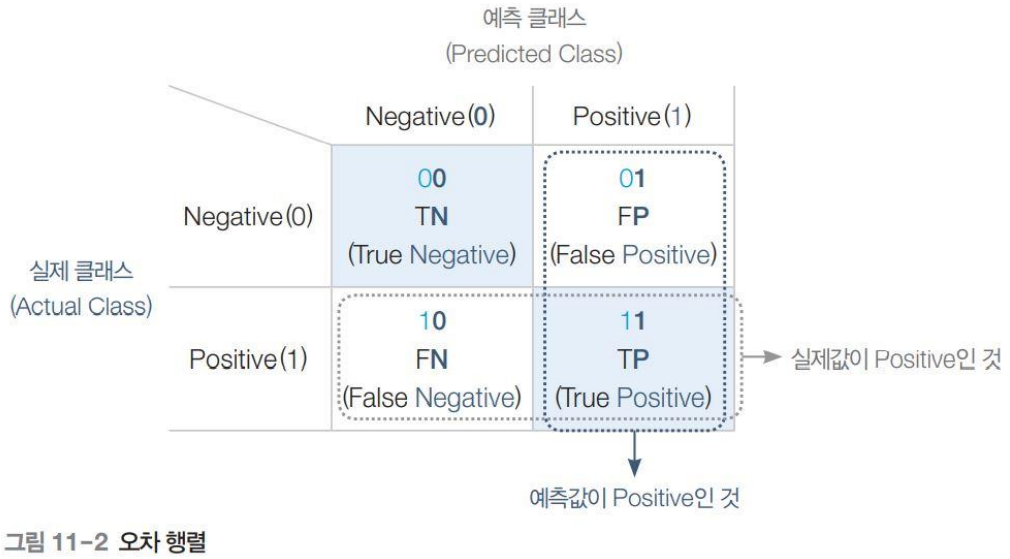


그림 11-2 오차 행렬

- X축(FPR) : 음성인 케이스를 양성으로 잘못 예측한 비율
- Y축(TPR) : 양성인 케이스를 양성으로 제대로 예측한 비율

$$False\ Positive\ Rate(FPR) = \frac{False\ Positive(FP)}{False\ Positive(FP) + True\ Negative(TN)}$$

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

특정 데이터로 유방암 진단하기

1. 데이터 준비하기

In [1]:	import numpy as np import pandas as pd from sklearn.datasets import load_breast_cancer
In [2]:	b_cancer = load_breast_cancer()

2. 데이터 탐색하기

In [3]:	print(b_cancer.DESCR)
In [4]:	b_cancer_df = pd.DataFrame(b_cancer.data, columns = b_cancer.feature_names)
In [5]:	b_cancer_df['diagnosis']= b_cancer.target
In [6]:	b_cancer_df.head()

특정 데이터로 유방암 진단하기 (cont'd)

3. 데이터셋의 크기와 독립 변수 X가 되는 피처에 대한 정보를 확인

In [7]:	print('유방암 진단 데이터셋 크기: ', b_cancer_df.shape)
In [8]:	b_cancer_df.info()

4. 로지스틱 회귀 분석에 피처로 사용할 데이터를 정규 분포 형태로 맞춤

In [9]:	from sklearn.preprocessing import StandardScaler scaler = StandardScaler()
In [10]:	b_cancer_scaled = scaler.fit_transform(b_cancer.data)
In [11]:	print(b_cancer.data[0])
In [12]:	print(b_cancer_scaled[0])

특정 데이터로 유방암 진단하기 (cont'd)

5. 로지스틱 회귀를 이용하여 분석 모델 구축하기

In [13]:	from sklearn.linear_model import LogisticRegression from sklearn.model_selection import train_test_split

In [14]:	# X, Y 설정하기
	Y = b_cancer_df['diagnosis']
	X = b_cancer_scaled

In [15]:	# 훈련용 데이터와 평가용 데이터 분할하기 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state = 0)

In [16]:	# 로지스틱 회귀 분석: (1) 모델 생성
	lr_b_cancer = LogisticRegression()

In [17]:	# 로지스틱 회귀 분석: (2) 모델 훈련
	lr_b_cancer.fit(X_train, Y_train)

In [18]:	# 로지스틱 회귀 분석: (3) 평가 데이터에 대한 예측 수행 -> 예측 결과 Y_predict 구하기
	Y_predict = lr_b_cancer.predict(X_test)

특정 데이터로 유방암 진단하기 (cont'd)

6. 생성한 모델의 성능 확인하기

In [19]:	from sklearn.metrics import confusion_matrix, accuracy_score from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score

In [20]:	# 오차 행렬
	confusion_matrix(Y_test, Y_predict)

In [21]:	acccuracy = accuracy_score(Y_test, Y_predict) precision = precision_score(Y_test, Y_predict) recall = recall_score(Y_test, Y_predict) roc_auc = roc_auc_score(Y_test, Y_predict)
In [22]:	print('정확도: {0:.3f}, 정밀도: {1:.3f}, 재현율: {2:.3f}'.format(acccuracy,precision,recall))
In [23]:	print('ROC_AUC: {0:.3f}'.format(roc_auc))