		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

<Fig1. Confusion matrix>

이제 각 case별로 살펴보겠습니다.

- True Positive(TP) : 실제 True인 정답을 True라고 예측 (정답)
- False Positive(FP) : 실제 False인 정답을 True라고 예측 (오답)
- False Negative(FN) : 실제 True인 정답을 False라고 예측 (오답)
- True Negative(TN) : 실제 False인 정답을 False라고 예측 (정답)

2. 분류 성능 평가 지표

2.1 Precision (정밀도)

- 모델의 입장에서, 모델이 True 라고 예측한 것 중에서 실제 True 인 것의 비율
- TP / (TP+FP)

- 한 달 30일 동안 맑은 날이 20일 이었는데, 확실한 2일 만 맑다고 예측한다면, 당연히 맑다고 한 날 중에 실제 맑은 날(Precision)은 100%가 나오게 됩니다. 하지만 과연, 이러한 모델이 이상적인 모델일까요?

2.2 Recall (재현율)

- 민감도와 동일
- 실제 정답(data)의 입장에서, 실제 True 인 것 중에서 모델이 True 라고 예측한 것의 비율
- TP / (TP+FN)

=> Precision이나 Recall은 모두 실제 True인 정답을 모델이 True라고 예측한 경우에 관심이 있으나, 바라보고자 하는 관점만 다르다.

2.3 Precision-Recall Trade-off

2.4 Accuracy (정확도)

- 전체 예측 (True, False) 한 것 중 맞게 예측한 비율
- (TP+TN) / (TP+FN+FP+TN)

- 문제점 : 불균형 데이터의 경우 정확한 평가 지표가 될 수 없다
- 만약 우리가 예측하고자 하는 한 달 동안이 특정 기후에 부합하여 비 오는 날이 흔치 않다고 생각해 보자. 이 경우에는 해당 data의 domain이 불균형 하게 되므로 맑은 것을 예측하는 성능은 높지만, 비가 오는 것을 예측하는 성능은 매우 낮을 수 밖에 없습니다. 따라서 이를 보완할 지표가 필요합니다.

2.5 F1 score

- 데이터가 불균형 할 때, accuracy가 아닌 f1 score 사용
- F1 score : precision과 recall의 조화 평균
- 평균(산술 평균)을 쓰지 않고 조화 평균을 구하는 이유는 recall, precision 둘 중 하나가 0에 가깝게 낮을 때 지표에 그것이 잘 반영되도록, 다시 말해 두 지표를 모두 균형 있게 반영하여 모델의 성능이 좋지 않다는 것을 잘 확인하기 위함이다.

2.6 ROC

- FPR의 변화에 따른 TPR의 변화 / trade-off
  - FPR(False Positive Rate) : 1-특이도(TN), 실제 0을 1이라 예측한 비율
    작으면 작을수록 좋음
  - TPR(True Positive Rate) : 재현율, 실제 1을 1이라 예측한 비율
    크면 클수록 좋음

ex) 모든 값을 1이라고 예측하면 재현율은 1이된다. 실제 1인 값을 모두 1이라고 예측했기 때문이다. 하지만 특이도는 0이 된다. 실제 0을 모두 1이라고 예측했기 때문이다.

- 계단처럼 각지게 되어있는 영역 : 임계값을 0에서 1로 단계적으로 변화시키면서 나타나는 현상
- 그래서, Tree 계열의 알고리즘은 임계값에 따라 변하는 node가 달라지기에 더 각진 모양의 roc 커브가 나온다

2.7 AUC

- roc curve 그래프 아래의 면적값
- AUC가 1에 가까울 수록 좋은 모델 (0.7이상이면 성능이 좋다고 평가)

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_auc_score

def print_metrics(y, pred_y, title=None):
    print(title)
    print('정확도(accuracy) :', accuracy_score(y, pred_y)
    print('confusion_matrix :')
    print(confunsion_matrix(y.values.argmax(axis=1), pred_y.argmax(axis=1))
    print('정밀도(precision), 재현율(recall), f1-score :')
    print(classification_report(y, pred_y, target_names=list(y.columns)
    print('AUC 점수 :', roc_auc_score(y, pred_y))
```

2.8 MCC (매튜 상관 계수)

$$F1\ \text{score} = \frac{2TP}{2TP + FN + FP}$$

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

FP와 FN은 바람직하지 않기 때문에 입력 할 때 빨간색으로 표시됩니다.

- 불균형 데이터 세트의 분류 성능 평가 지표
- MCC 범위 : -1에서 1