



3. EVALUATION

〈개념 위주로〉

@m.joon_ixix

1. 정확도 (ACCURACY)

- 항상 써오던 `accuracy_score`
- 전체 예측 중에 얼마나 맞았는지.
- Ex) 100 predictions. 85 correct. \rightarrow Accuracy = 0.85

2. 오차 행렬 (CONFUSION MATRIX)

		Prediction	
		0 (Negative)	1 (Positive)
Actual	0 (Negative)	TN	FP
	1 (Positive)	FN	TP

False로 시작
→ “내가 어떻게 predict 했는지”로 이름 결정

오차 행렬 →

TN	FP
FN	TP

각각의 칸에는
해당하는 데이터의 개수 (레코드 수)

* 정밀도, 재현율 → Positive 에 중점을 두자!

3. 정밀도 (PRECISION)

- “내가 얼마나 정밀하게 했는지?” → “내가 예측한 positive들이 얼마나 맞는지?”

- 정밀도 = $\frac{TP}{TP+FP}$

- 정밀도가 중요한 경우 → FP가 적어야 함

→ 실제로 negative인데, positive라고 잘못 판단하면 안됨 → (예시) 스팸 메일 판단

4. 재현율 (RECALL)

- “얼마나 재현을 잘했는지?” → “실제로 positive인 것들을 얼마나 정확하게 잡아냈는지?”

- 재현율 = $\frac{TP}{TP+FN}$

- 재현율이 중요한 경우 → FN가 적어야 함

→ 실제로 positive인데, negative라고 잘못 판단하면 안됨 → (예시) 암 진단

* 정밀도, 재현율은 서로 Trade-off 관계 (참고: 1종 오류와 2종 오류의 trade-off → FP vs FN)

Selected thresholds:	[0.1039 0.1213 0.1404 0.1861 0.2806 0.4034 0.5634 0.6667 0.8218 0.9472]
Precisions matching:	[0.3885 0.4403 0.4661 0.5392 0.6471 0.7286 0.8364 0.9487 0.9583 1.]
Recalls matching:	[1. 0.9672 0.9016 0.9016 0.9016 0.8361 0.7541 0.6066 0.377 0.1475]

*Prediction에 사용되는 Threshold (임계값) 조절을 통해
FP와 FN를 적절히 조절하여 정밀도와 재현율을 조절할 수 있다!*

5. F1-SCORE

$$F1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 * \frac{precision * recall}{precision + recall}$$

- precision & recall 이 치우치지 않고 균등할수록 F1 score는 높게 나옴

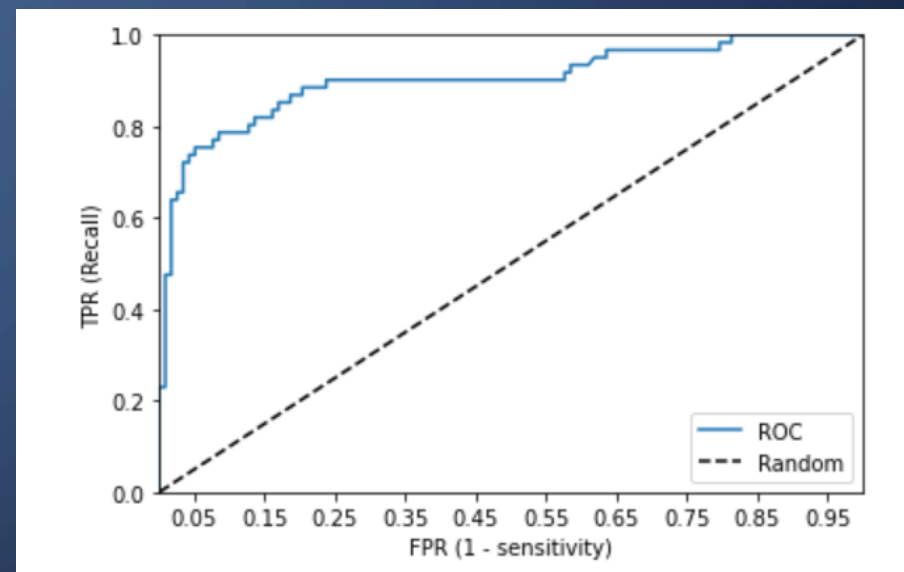
6. ROC, AUC

- ROC curve

$$TPR = \frac{TP}{TP+FN} \quad (\text{실제로 Positive인 것 중에 얼마나 TP가 나왔는지})$$

$$FPR = \frac{FP}{TN+FP} \quad (\text{실제로 Negative인 것 중에 얼마나 FP가 나왔는지})$$

- 곡선이 직선(random 상황)에서 멀어질 수록 모델의 예측 성능이 좋은 것



- AUC score : ROC curve에서 곡선 아래 부분의 넓이 (大大익선)

분석은 코드 보면서 ☺

m-joon-ixix / 2021Winter-MachineLearningStudy

<> Code ! Issues 🔑 Pull requests ▶ Actions 📁 Projects 📖 Wiki ⓘ

main 2021Winter-MachineLearningStudy / PythonMLguidebook / chapter3 /

m-joon-ixix chapter3 update

..

accuracy_and_matrix.ipynb	chapter3 update
f1_and_roc_auc.ipynb	chapter3 update
precision_and_recall.ipynb	chapter3 update