인공지능 딥러닝 FAQ

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 예측 | |
| P | N |
| 실제 | P | TP | FN |
| N | FP | TN |

**[추가]**

1. F1 score가 효과적인 상황
   1. 최적의 Threshold를 이미 알고 있을 때나 Threshold가 고정되어 있을 때
   2. P를 맞추는 것, N을 맞추는 것 둘 모두가 중요할 때
2. AUC, ROC는 최적의 Threshold를 모를 때 효과적.

**1. F1 score란?**

AI 모델을 평가하기 위한 지표.

일반적으로 사용하는 정확도(accuracy)는 데이터가 불균형할 때 모델의 정확한 성능을 알기 힘듦. 예를 들어, 답이 1인 데이터가 100개, 0인 데이터가 10000개라면, 1 쪽을 전부 틀렸더라도 0 쪽을 전부 맞혔다면 정확도가 10000/10100 => 99%가 나옴.

이를 보완하기 위해 **Precision**(정밀도: 모델이 긍정이라고 예측한 것 중, 실제로 긍정인(맞춘) 비율)와 **Recall**(재현율: 실제로 긍정인 것 중 모델이 긍정이라고 예측한(맞춘) 비율) 둘을 이용해 만든 성능 지표가 F1 score.

**공식: F1 = 2 × { (precision × recall) / (precision + recall) }**

\* 공식에 precision × recall이 있기 때문에 둘 중 하나가 0에 가까우면 전체 점수가 크게 깎임 => **두 값 모두 어느 정도 높아야** F1 score가 높아짐 => 그러므로 모델이 모든 레이블 값에 대해 균형있는 성능을 가졌는지 확인 가능

**2. Recall이란?**

재현율. 실제로 긍정인 것 중(TP + FN), 모델이 긍정이라고 예측한(맞춘) 비율(TP).

공식: TP / (TP + FN)

**3. Precision이란?**

정밀도. 모델이 긍정이라고 예측한 것 중(TP + FP), 실제로 긍정인(맞춘) 비율(TP).

공식: TP / (TP + FP)

**4. 민감도(Sensitivity)란?**

민감도 = Recall. 특이도의 반대 개념으로 얘기할 때 민감도라는 용어를 사용. 특히 감염 검사에서, 실제로 양성인 사람들 중 모델이 양성임을 찾아낸 비율을 말함. 쉽게 말하면, 진짜 감염된 사람을 놓치지 않고 찾아낸 비율.

공식: TP / (TP + FN)

**5. 특이도(Specificity)란?**

민감도의 반대 개념. 특히 감염 검사에서, 실제로 음성인 사람들 중 모델이 음성임을 찾아낸 비율을 말함. 쉽게 말하면, 건강한 사람을 괜히 감염됐다고 오진하지 않은 비율.

공식: TN / (TN + FP)

**6. AUC, ROC란?(이거 더 써야 함. 아직 이해 못함!!)**

**텍스트, 라인, 도표, 그래프이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.**그림 출처: https://angeloyeo.github.io/2020/08/05/ROC.html

**TPR**(True Positive Ratio): 실제로 양성인 데이터를 제대로 양성이라고 맞춘 비율. 민감도와 같음. 공식: TP / (TP + FN)

**FPR**(False Positive Ratio): 실제로 음성인 데이터를 양성이라고 예측한(틀린) 비율. (1 – 특이도)와 같음. 공식: FP / (FP + TN)

**Threshold**: 일반적으로 모델은 어떤 데이터에 대해 해당 데이터의 답이 긍정(1)일 ‘확률’을 뱉는데, 여기서 확률이 몇 이상이면 데이터의 답을 긍정(1)으로 판단할지 기준이 Threshold.

**ROC**(Receiver Operating Characteristic curve):

x축을 FPR(= 1 – 특이도),

y축을 TPR(= 민감도)로 한 좌표평면에,

모델이 한 단계에서 예측한 결과를 놓고,

각 threshold 값마다 FPR, TPR 을 각각 구해 점을 찍어 이은 곡선 그래프.

\* ROC는 특정 threshold 값에 대해 FPR, TPR 을 알기 위한 곡선은 아니다.

\* ROC는 혼동행렬같은 모델 성능 측정 방법 중 하나.

\* 데이터가 제대로 분류되지 않아 특정 threshold에서 실제 긍정, 부정 데이터가 제대로 분류되지 못하고 섞인 상태면 ROC 곡선이 점점 내려온다.

1. ROC 곡선이 내려오지 않고 위에 딱 붙어 90도로 꺾인 모양이면, 완벽히 분류된 것.
2. ROC 곡선이 위로 볼록하게 꺾여 있으면, 완벽하진 않더라도 어느 정도 분류해내고는 있는 것.
3. ROC 곡선이 꺾이지 않은 채 45도 대각선 직선이면, 데이터가 완벽하게 반반 섞여 있는 것. (TP, FP 둘이 값이 같고, TN, FN 둘이 값이 같음)
4. ROC 곡선이 아래로 볼록하게 꺾여 있으면, 모델이 데이터를 반대로(긍정은 부정으로, 부정은 긍정으로) 분류해버린 것. (TP 보다 FP 가, TN 보다 FN 가 더 많음)

**AUC**(Area Under the Curve): ROC 곡선 아래 공간의 면적. 이 값이 클수록 좋은 모델.

* AUC = 1.0 -> 완벽한 모델
* AUC = 0.5 -> 아무것도 못 맞힌, 그냥 답을 랜덤으로 내는 모델
* AUC < 0.5 -> 원하는 답과 반대로 예측하는 모델

[ROC를 그리는 과정]

1. 모델이 어떤 데이터에 대해 해당 데이터의 답이 ‘긍정일 확률’을 뱉어냄.

2. threshold를 기준으로 모델의 예측을 긍정(1), 부정(0)으로 나눈 뒤 TP, FP, TN, FN을 세고 민감도, 특이도를 계산.

3. 3에서 구한 **민감도가 y값(TPR)**, **(1 – 특이도)가 x값(FPR)**이 되어 좌표평면에 한 점이 찍힘.

=> 여기서 **threshold를 0~1까지 쭉 바꿔가며(0.01, 0.02, …)** 그때마다 TPR, FPR 을 계산해 좌표평면에 점을 찍어 쭉 이으면 이 그래프가 ROC.

참조

노마드분석가, ROC 곡선 아주 쉽게 이해하기, <https://diseny.tistory.com/entry/ROC-%EA%B3%A1%EC%84%A0-%EC%95%84%EC%A3%BC-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0>, 2022.3.21, 글쓴이가 계산은 잘못했지만, 설명은 아주 쉽게 해 줌.