3. 평가

머신러닝은 데이터 가공/변환, 모델 학습/예측, 그리고 평가(Evaluation)의 프로세스로 구성

분류

- 결정 클래스 값 종류의 유형에 따라 긍정/부정과 같은 2개의 결괏값만을 가지는 이진 분류
- 여러 개의 결정 클래스 값을 가지는 멀티 분류

분류(모델)의 성능 평가 지표(이진 분류에서 더욱 중요하게 강조하는 지표)

- 정확도(Accuracy)
- 오차행렬(Confusion Matrix)
- 정밀도(Precision)
- 재현율(Recall)
- F1 스코어
- ROC AUC

1. 정확도(Accuracy)

정확도(Accuracy) = 예측 결과가 동일한 데이터 건수 / 전체 예측 데이터 건수

이진분류 의 경우 데이터의 구성에 따라 ML 모델의 성능을 왜곡할 수 있기 때문에 정확도 수치 하나만 가지고 성능을 평가하지 않음
ex) 타이타닉 데이터에서 무조건 성별이 여자인 경우 생존으로, 남자인 경우 사망으로 예측 결과를 예측해도 이와 비슷한 수치가 나올 수 있음

*불균형한(imbalanced) 레이블 값 분포*에서 ML 모델의 성능을 판단할 경우, 적합한 평가 지표가 아님
ex) 100개의 데이터가 있고 이 중에 90개의 데이터 레이블이 0, 단 10개의 데이터 레이블이 1이라고 한다면 무조건 0으로 예측 결과를 반환하는 ML 모델의 경우라도 정확도가 90%가 됨

2. 오차 행렬(confusion matrix, 혼동행렬)

학습된 분류 모델 이 예측을 수행하면서 얼마나 헷갈리고(confused) 있는지도 함께 보여주는 지표

이진 분류의 예측 오류가 얼마인지와 더불어 어떠한 유형의 예측 오류가 발생하고 있는지를 함께 나타내는 지표

4분면 행렬에서, 4분면의 왼쪽, 오른쪽을 예측된 클래스 값 기준으로 Negative와 Positive로 분류
4분면의 위, 아래를 실제 클래스 값 기준으로 Negative와 Positive로 분류

<예측 클래스>
TN | FP
FN | TP

TN, FP, FN, TP는 예측 클래스와 실제 클래스의 Positive 결정 값 (값 1) 과 Negative 결정 값 (값 0) 의 결합에 따라 결정
앞 문자 True/False는 예측값과 실제값이 '같은가/틀린가'를 의미

뒤 문자 Negative/Positive는 예측 결과 값이 부정 (0) /긍정 (1) 을 의미
ex) TN: 예측값을 맞혀서 True, 예측 값 Negative 값 0.=> 실제 값도 Negative.

TP, TN, FP, TN 값을 조합해 Classifier의 성능을 측정할 수 있는 주요 지표인 정확도(Accuracy), 정밀도(Precision), 재현율(Recall) 값을 알 수 있음

정확도 = 예측 결과와 실제 값이 동일한 건수/전체 데이터 수 = (TN + TP)/(TN + FP + FN + TP)

불균형한 이진 분류 데이터 세트에서는 Positive 데이터 건수가 매우 작음
=> TN 매우 커짐, TP 매우 작아짐. FN 매우 작아짐, FP 매우 작아짐(positive로 예측을 잘 안 함)
=> Positive에 대한 예측 정확도를 판단하지 못한 채 Negative에 대한 예측 정확도만으로도 분류의 정확도가 매우 높게 나타나는 수치적인 판단 오류 일어남

3. 정밀도와 재현율

Positive 데이터 세트의 예측 성능에 좀 더 초점을 맞춘 평가 지표

<정밀도>
:예측을 Positive로 한 대상 중에 예측과 실제 값이 Positive로 일치한 데이터의 비율
= TP / (FP + TP) 공식의 분모인 FP + TP: 예측을 Positive로 한 모든 데이터 건수
공식의 분자인 TP: 예측과 실제 값이 Positive로 일치한 데이터 건수

정밀도가 중요한 경우:

- 실제 Negative 데이터를 Positive로 분류하면 안되는 경우
- 스팸메일 여부를 판단하는 모델

TP를 높이고, FP를 낮추는 데 초점

사이킷런은 정밀도 계산을 위해 precision_score() 제공

<재현율>
실제 값이 Positive인 대상 중에 예측과 실제 값이 Positive로 일치한 데이터의 비율
= TP / (FN + TP)
공식의 분모인 FN + TP: 실제 값이 Positive인 모든 데이터 건수
공식의 분자인 TP: 예측과 실제 값이 Positive로 일치한 데이터 건수

재현율이 중요한 지표인 경우:

- 실제 Positive 양성 데이터를 Negative로 잘못 판단하게 되면 업무상 큰 영향이 발생하는 경우
- 보험 사기와 같은 금융 사기 적발 모델

TP를 높이고, FN을 낮추는 데 초점

사이킷런은 재현율 계산을 위해 recall_score() 제공

정밀도/재현율 트레이드오프

정밀도와 재현율은 상호 보완적인 평가 지표이기 때문에 어느 한 쪽을 강제로 높이면 다른 하나의 수치는 떨어지기 쉬움

predict_proba( ) (사이킷런 제공)

- 입력 파라미터: predict( ) 메서드와 동일하게 보통 테스트 피처 데이터 세트를 입력
- 반환 값:
  개별 클래스의 예측 확률을 ndarray m x n (m : 입력값의 레코드 수, n : 클래스 값 유형) 형태로 반환
  입력 테스트 데이터 세트의 표본 개수가 100개이고 예측 클래스 값 유형이 2개(이진 분류)라면 반환값은 100 x 2 ndarray임.
  각 열은 개별 클래스의 예측 확률. 이진 분류에서 첫 번째 칼럼은 0 Negative의 확률, 두 번째 칼럼은 1 Positive의 확률

사이킷런의 predict( )는 predict_proba( ) 메서드가 반환하는 확률 값을 가진 ndarray에서 정해진 임곗값을 만족하는 ndarray의 칼럼 위치를 최종 예측 클래스로 결정

Binarizer 클래스
: threshold 변수를 특정 값으로 설정하고 Binarizer 클래스를 객체로 생성
생성된 Binarizer 객체의 fit_transform() 메서드를 이용해 넘파이 ndarray를 입력하면 입력된 ndarray의 값을 지정된 threshold(분류 결정 임곗값)보다 같거나 작으면 0값으로, 크면 1값으로 변환해 반환

분류 결정 임곗값은 Positive 예측값을 결정하는 확률의 기준이 됨.
=> 임곗값을 변화시켜보며 정확도, 정밀도, 재현율 비교

precision_recall_curve() (사이킷런 제공)

- 입력 파라미터 y_true : 실제 클래스값 배열 (배열 크기= [데이터 건수])
  probas_pred : Positive 칼럼의 예측 확률 배열 (배열 크기= [데이터 건수])
- 반환값 정밀도: 임곗값별 정밀도 값을 배열로 반환 재현율: 임곗값별 재현율 값을 배열로 반환

임곗값이 낮을수록 많은 수의 양성 예측으로 인해 재현율 값이 극도로 높아지고 정밀도 값이 극도로 낮아지고, 임곗값을 계속 증가시킬수록 재현율 값이 낮아지고 정밀도 값이 높아짐

정밀도와 재현율이 맹점

Positive 예측의 임곗값을 변경함에 따라 정밀도와 재현율의 수치가 변경됨
임곗값의 이러한 변경은 업무 환경에 맞게 두 개의 수치를 상호 보완할 수 있는 수준에서 적용돼야 함

4. F1 스코어

정밀도와 재현율을 결합한 지표

정밀도와 재현율이 어느 한 쪽으로 치우치지 않는 수치를 나타낼 때 상대적으로 높은 값을 가짐

공식:
F1 = 2 / (1/recall + 1/precision) = 2 * (precision * recall) / (precision + recall)

f1_score() (사이킷런 제공)

5. ROC곡선과 AUC

ROC 곡선과 이에 기반한 AUC 스코어는 이진 분류의 예측 성능 측정에서 중요하게 사용되는 지표

ROC 곡선: FPR(False Positive Rate) 이 변할 때 TPR(True Positive Rate)이 어떻게 변하는지를 나타내는 곡선
FPR을 X 축으로, TPR을 Y 축으로 잡으면 FPR의 변화에 따른 TPR의 변화가 곡선 형태로 나타남

TPR(True Positive Rate) : 재현율(민감도)
= TP / (FN + TP)

실제값 Positive(양성)가 정확히 예측돼야 하는 수준을 나타냄(질병이 있는 사람은 질병이 있는 것으로 양성 판정)

TNR(True Negative Rate): 특이성(Specificity)
= TN / (FP + TN)

실제값 Negative(음성)가 정확히 예측돼야 하는 수준을 나타냄(질병이 없는 건강한 사람은 질병이 없는 것으로 음성 판정)

FPR = FP / (FP + TN) = 1 - TNR = 1- 특이성

ROC 곡선은 FPR을 0부터 1까지 변경하면서 TPR의 변화 값을 구함
=> 분류 결정 임곗값을 변경하여 FPR 변경
FPR을 0으로 만들려면 임곗값을 1로 지정
FPR을 1로 만들려면 임곗값을 0으로 지정하여 TN을 0으로 만들기

roc_curve( ) (사이킷런 제공)

입력파라미터

- y_true : 실제 클래스 값 array (array shape = [데이터 건수])
- y_score : predict_proba()의 반환 값 array에서 Positive 칼럼의 예측 확률이 보통 사용됨. array.shape = [n_samples]

반환 값

- fpr : fpr 값을 array로 반환
- tpr : tpr 값을 array로 반환
- thresholds : threshold 값 array

AUC(Area Under Curve)

일반적으로 ROC 곡선 자체는 FPR과 TPR의 변화 값을 보는 데 이용하며,
분류의 성능 지표로 사용되는 것은 *ROC 곡선 면적에 기반한 AUC 값*으로 결정

ROC 곡선 밑의 면적을 구한 것으로서 일반적으로 1에 가까울수록 좋은 수치

AUC 수치가 커지려면 FPR이 작은 상태에서 얼마나 큰 TPR을 얻을 수 있느냐가 관건