지도 학습 모델 평가 방법

회귀 모델 평가 지표

* 회귀 평가를 위한 지표는 **실제 값과 회귀 예측값의 차이 값을 기반**으로한 지표가 중심이다.
* 실제 값과 예측 값의 차이를 그냥 더하면 +와 -가 섞여서 오류가 상쇄된다. 때문에 오류의 절대값 평균이나 제곱, 또는 제곱한 뒤 다시 루트를 씌워 평균값을 구한다.
* MAE( Mean Absolute Error )

실제 값과 예측 값의 차이를 절대값으로 변환해 평균한 것이다.

MAE = 1n∑i=1n|yi−y^i|

* MSE( Mean Squared Error )

실제 값과 예측 값의 차이를 제곱해 평균한 것이다.

MSE = 1n∑i=1n(yi−y^i)2

* RMSE( Root Mean Squared Error )

MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운것이 RMSE이다.

RMSE = root ( 1n∑i=1n(yi−y^i)2 )

* R2

분산 기반으로 예측 성능을 평가한다. 실제 값의 분산 대비 예측 값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높다.

R2 = 예측값Variance / 실제값Variance

* MSE나 RMSE에 로그를 적용한 MSLE( Mean Sequared Log Error )와 RMSLE( Root Mean Squared Log Error ) 도 사용한다.
* scikit-learn 제공 평가 지표 API

| 평가방법 | scikit-learn 제공 평가 지표 API | scoring 함수 적용 값 |
| --- | --- | --- |
| MAE | metrics.mean\_absolute\_error() | 'neg\_mean\_absolute\_error' |
| MSE | metrics.mean\_squared\_error() | 'neg\_mean\_squared\_error' |
| R^2 | metrics.r2\_score() | 'r2' |

* scikit-learn에서는 RMSE를 지원하지 않는다. RMSE를 구하기 위해서는 MSE에 제곱근을 씌워서 계산하는 함수를 직접 작성해야 한다.
* scikit-learn의 cross\_val\_score() 함수와 같은 scoring 함수에 회귀 평가 지표를 적용하여 평가한다.

분류 모델 예측력 평가 지표

* 모델이 예측하는 값에는 두 가지( True / False )가 있고, 각 예측값은 실제로 True이거나 False일 수 있다.
* 모델의 예측값과 실제값을 각각 축으로 2 X 2 매트릭스로 표현한 것을 혼동 행렬( Confusion Matrix )라 한다.
* 정확도( Accuracy )
* 정확도는 실제 데이터에서 예측 데이터가 얼마나 같은지를 판단하는 지표

정확도( Accuracy ) =

예측 결과가 동일한 데이터 건수 / 전체 예측 데이터 건수

* 정확도는 직관적으로 모델 예측 성능을 나타내는 평가 지표
* scikit-learn은 오차 행렬을 구하기 위해 accuracy\_score()
* 혼동 행렬( Confusion Matrix )
* 모델이 예측하는 값에는 두 가지( True / False )가 있고, 각 예측값은 실제로 True이거나 False일 수 있다.
* 모델의 예측값과 실제값을 각각 축으로 2 X 2 매트릭스로 표현한 것을 혼동 행렬( Confusion Matrix )라 한다.

| **예측값\실제값** | **T** | **F** |
| --- | --- | --- |
| T | TP( True Positive ) | FP( False Positive ) |
| F | FN( False Negative ) | TN( True Negative ) |

* scikit-learn은 오차 행렬을 구하기 위해 confusion\_matrix()
* 정밀도( Precision )
* True로 예측한 분석 대상중에서 실제 값이 True인 비율을 말하며, **모델의 정확성**을 나타내는 지표가 된다.
* 정밀도가 높다는 것은 False Positive( 실제 False를 True로 잘못 예측 ) 오류가 작다는 뜻이다.

Precision = TP / TP+FP

* scikit-learn은 정밀도 계산을 위한 precision\_score()를 API로 제공
* 재현율( Recall )
* 실제 값이 True인 분석 대상중에서 True로 예측하여 모델이 적중한 비율을 말하며, **모형의 안전성**을 나타내는 지표이다.
* 재현율이 높다는 것은 False Negative( 실제 True를 False로 잘못 예측 ) 오류가 낮다는 뜻이다.

𝑅𝑒𝑐𝑎𝑙𝑙= 𝑇𝑃 / 𝑇𝑃 + 𝐹𝑁

* scikit-learn은 재현율 계산을 위한 recall\_score()를 API로 제공
* F1 지표( F1-score )
* **정확도와 재현율이 균등하게 반영될 수 있도록 정확도와 재현율의 조화 평균을 계산한 값**으로, 모델의 예측력을 종합적으로 평가한 지표
* 값이 높을수록 분류 모델의 예측력이 좋다고 말할 수 있다.

𝐹1−𝑠𝑐𝑜𝑟𝑒 = 2∗(𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛∗𝑅𝑒𝑐𝑎𝑙𝑙) / 𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛+𝑅𝑒𝑐𝑎𝑙𝑙

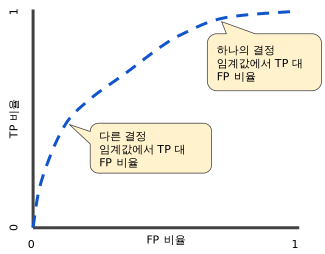
* scikit-learn은 F1 스코어를 구하기 위해 f1\_score()라는 API를 제공
* ROC/AUC
* ROC
* ROC 곡선( 수신자 조작 특성 곡선 ) : 모든 분류 임계값에서 분류 모델의 성능을 보여주는 그래프. 이 곡선은 두 매개변수를 표시한다.
  + 참 양성 비율( TPR )
  + 허위 양성 비율( FPR )
* 참 양성 비율( TPR )은 재현율의 동의어이다.

𝑇𝑃𝑅 = 𝑇𝑃 / 𝑇𝑃 + 𝐹𝑁

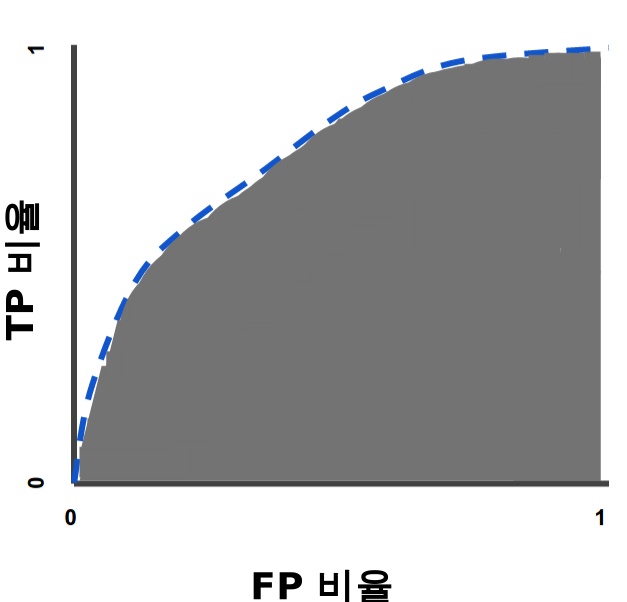
* 허위 양성 비율( FPR )

𝐹𝑃𝑅 = 𝐹𝑃 / 𝐹𝑃 + 𝑇𝑁

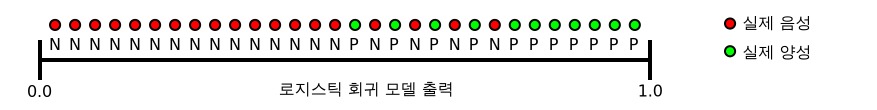
* ROC 곡선은 다양한 분류 임계값의 TPR 및 FPR을 나타낸다. 분류 임계값을 낮추면 더 많은 항목이 양성으로 분류되므로 거짓양성과 참양성이 모두 증가한다.



* ROC 곡선의 점을 계산하기 위해 분류 임계값이 다른 로지스틱 회귀 모델을 여러 번 평가할 수 있지만 이 방법은 효율적이지 않다. 다행이 이 정보를 제공할 수 있는 효율적인 정렬 기반 알고리즘이 AUC 이다.
* scikit-learn은 ROC 곡선을 구하기 위해 roc\_curve() API를 제공
* AUC
* 'ROC 곡선 아래 영역'을 의미. 즉 AUC는 (0,0)에서 (1,1)까지 전체 ROC 곡선 아래에 있는 전체 2차원 영역을 측정한다.( 적분 )



* AUC는 가능한 모든 분류 임계값에서 성능의 집계 측정값을 제공한다.
* AUC를 해석하는 한 가지 방법은 모델이 임의 양성 예제를 임의 음성 예제보다 더 높게 평가할 확률이다.
* 예로 로지스틱 회귀 예측의 오름차순으로 왼쪽에서 오른쪽으로 정렬되어 있다.



* AUC는 임의의 양성( 초록색 ) 예제가 임의의 음성( 빨간색 ) 예제의 오른쪽에 배치되는 확률을 나타낸다.
* AUC 값의 범위는 0 ~ 1이다. 예측이 100% 잘못된 모델의 AUC는 0.0이고 예측이 100% 정확한 모델의 AUC는 1.0이다.
* AUC는 두 가지 이유로 이상적이다.
  + AUC는 **척도 불변**이다. AUC는 절대값이 아니라 예측이 얼마나 잘 평가되는지 측정한다.
  + AUC는 **분류 임계값 불변**이다. AUC는 어떤 분류 임계값이 선택되었는지와 상관없이 모델의 예측 품질을 측정한다.
* 위의 두 이유는 특정 사용 사례에서 AUC의 유용성을 제한할 수 있다는 단점이 있다.
  + **척도 불변이 항상 이상적인 것은 아니다.** 예로 잘 보정된 확률 결과가 필요한 경우가 있는데 AUC로는 이정보를 알 수 없다.
  + **분류 임계값 불변이 항상 이상적인 것은 아니다.** 허위 음성(FN) 비용과 허위 양성( FP ) 비용에 큰 차이가 있는 경우 한 가지 유형의 분류 오류를 최소화하는 것은 위험할 수 있다. 예로 이메일 스팸 감지를 실행할 때 허위 양성(FP)의 최소화로 인해 허위 음성(FN)이 크게 증가한다고 해도 허위 양성(FP) 최소화를 우선시하고 싶을 수 있다. AUC는 이런 유형의 최적화에 유용한 측정항목이 아니다.
* scikit-learn은 AUC 곡선을 구하기 위해 roc\_auc\_score() API를 제공