CHAPTER3. 분류

- 성능 측정
- 다중 분류
- 다중 레이블 분류

3.1 성능 측정

3.1.1 Cross Validation (CV, 교차검증)

- 모델을 만들다 보면 성능향상을 위해 많은 일들을 진행합니다. ex) 모수 개수 늘리기, 비선형 모델 사용 등
- 그렇게 하다보면 Overfitting(과최적화) 문제가 발생합니다. 이를 방지하기 위해 교차검증을 시행합니다.
- 교차검증은 모델에 사용된 데이터를 Train/Test로 나눠 검증을 실시하고 평균 성능과 평균 분산을 구하는 것을 의미합니다.
- 대표적인 교차검증에는 K-Fold 교차검증이 있습니다.
- K-Fold는 K개의 Fold(Train/Test)를 만들어서 각 Fold별로 성능을 측정하는 방법입니다.
- K개의 데이터에 대한 성능이 비슷해야 Overfitting이 없는 모델이라 할 수 있습니다.

3.1.2 Confusion Matrix (오차 행렬)

- 실제 y값과 모델이 예측한 y값이 서로 일치하는지 개수를 파악하는 것을 의미합니다.

	0이라 예측	**1이라 예측**	**2라고 예측**
실제값이 0	실제값이 0인데 0이라 예측	실제값이 0인데 1이라 예측	실제값이 0인데 2라고 예측
실제값이 1	실제값이 1인데 0이라 예측	실제값이 1인데 1이라 예측	실제값이 1인데 2라고 예측
실제값이 2	실제값이 2인데 0이라 예측	실제값이 2인데 1이라 예측	실제값이 2인데 2라고 예측

3.1.3 Classification Report

: Classification은 일반적으로 Binary Confusion Matrix를 활용합니다.

	Positive라고 예측	**Negative라고 예측**
실제 Positive	True Positive (TP)	False Negative (FN)
실제 Negative	False Positive (FP)	True Negative (TN)

1) Accuracy (정확도)

- 전체 샘플 중 맞게 예측한 샘플의 비율입니다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2) Precision (정밀도)

- Positive라고 예측한 샘플 중 실제값이 Positive인 샘플의 비율을 의미합니다.

$$Precision = \frac{TP}{TP + FP}$$

3) Recall (재현율)

- 실제값이 Positive인 샘플 중 Positive라고 예측한 비율입니다.
- 분류기가 정확히 감지한 양성 샘플의 비율로 Sensitivity(민감도) 또는 TPR(True Positive Rate, 진짜 양성비율)이라고도 합니다.

$$Recall = \frac{TP}{TP + FN}$$

4) Specificity (특이도)

- 실제값이 Negative인 샘플 중 Negative라고 예측한 비율입니다.
- Specificity = 1 - Fall-out (위양성율)

$$Specificity = \frac{TN}{FP + TN}$$

5) Fall-Out (위양성율)

- 실제값이 Negative인 샘풀 중 Positive라고 예측한 비율(FPR, False Positive Rate)입니다.
- Fall-out = 1 - Specificity (특이도)

$$Fall - out = \frac{FP}{FP + TN}$$

6) F(beta) Score

- Precision(정밀도)와 재현율(Recall)의 조화평균입니다.

$$F_b = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall}$$

- F1 Score는 beta에 1을 대입한 스코어입니다. 다음과 같은 수식을 얻을 수 있습니다.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- 정밀도와 재현율이 비슷한 분류기에서는 F1점수가 높습니다. 하지만 이게 항상 바람직한것은 아닙니다.
- 예를 들어 어린아이에게 안전한 동영상을 걸러내는 분류기를 훈련한다고 했을 때, 좋은 동영상이 많이 제외되더라도(낮은 재현율) 안전한 것들만 보여주는(높은 정밀도) 분류기를 선호할 것입니다.
- 다른 예로 감시 카메라로 도둑을 잡아내는 분류기를 훈련한다고 했을 때, 분류기의 재현율이 99%라면 정확도가 30%만 되더라고 괜찮을 겁니다. (경비원들이 고생하겠지만요..)

7) 관계 파악

- Precision(정밀도)과 Recall(재현율) & Fall-out(위양성율)은 대략적으로 음의 상관관계가 있습니다.
- 정밀도를 높이기 위해 판단기준을 엄격하게 할수록 Recall(재현율)이나 Fall-out(위양성율)이 감소하는 경향을 띕니다.
- 반면 Recall(재현율)과 Fall-out(위양성율)은 양의 상관관계가 있습니다.
- Recall(재현율)을 높이기 위해선 Positive로 판단하는 기준을 낮춰 조금만 조건에 충족하면 양성으로 판단하면 됩니다.
- 이러면 음성임에도 양성으로 판단되는 데이터가 같이 증가하고 이는 Fall-out(위양성율)을 증가시키는 결과를 초래합니다.

3.1.4 정밀도(Precision) / 재현율(Recall) 트레이드오프

- 모델은 결정 함수(Decision Function)를 사용해 각 샘플의 점수를 계산합니다.
- 이 점수가 임곗값(Threshold)보다 크면 양성 클래스에, 그렇지 않으면 음성 클래스에 할당합니다.
- 일반적으로 임곗값을 높이면 정밀도가 높아지고 재현율은 낮아집니다.
- 반대로 임곗값을 낮추면 정밀도가 낮아지고 재현율은 높아집니다.
- 재현율에 대한 정밀도 곡선을 그리고 프로젝트에 맞는 정밀도/재현율 트레이드오프를 선택합니다.

3.1.5 ROC Curve (Receiver Operator Characteristic)

- ROC Curve는 클래스 판별 기준값의 변화에 따른 Recall(재현율)과 Fall-ou(위양성율)의 변화를 시각화한 것입니다.

f	y_hat	y
2.167	1	1
1.447	1	1
1.234	1	1
0.452	1	1
-0.152	0	1
-0.784	0	0
-1.235	0	0

1) ROC Curve가 작성되는 과정

- 현재는 0을 기준값(Threshold)으로 구분해 판별함수값(f)이 0보다 크면 양성(1), 작으면 음성(0)이 됩니다.

- 데이터가 분류가 다르게 되도록 기준값을 증가 or 감소시킵니다.
- 기준값을 여러가지 방법으로 증가 or 감소시키면 여러가지 다른 기준값에 대해 분류결과가 달라지고 Recall, Fall-out 등의 성능 평가 점수도 달라집니다.
- Scikit-Learn의 roc_corve는 위 과정을 자동화해줍니다.

2) 특징

- 다중 클래스의 경우는 정밀도, 재현율, 위양성율을 구하거나 ROC Curve를 그릴 수 없습니다.
- 해결하는 방법은 각각의 클래스에 대해 OvR문제를 가정해 각각의 OvR에 대한 값을 구합니다.

3.1.6 AUC (Area Under the Curve)

- AUC는 ROC Curve의 면적을 의미합니다.
- Fall-out(위양성율) 대비 Recall(재현율)값이 클수록 1에 가깝고 민감한 모형입니다.

3.2 다중 분류

3.2.1 OvR (One versus Rest)

- 특정 숫자 하나만 구분하는 숫자별 이진 분류기 N개를 훈련시켜 클래스가 N개인 숫자 이미지 분류 시스템을 만듭니다.
- 이미지를 분류할 때 각 분류기의 결정 점수 중에서 가장 높은 것을 클래스로 선택하는 방법입니다.

3.2.2 OvO (Onn versus One)

- 0과 1구별, 0과 2구별, 1과 2구별 등과 같이 각 숫자의 조합마다 이진분류기를 훈련시키는 방법입니다.
- 클래스가 N개면 분류기는 N(N-1)/2 개가 필요합니다.
- 투표를 통해 가장 많은 표를 얻은 클래스가 선택되는데 정규화된 판결 기준값을 이용합니다.

3.3 다중 레이블 분류

- 분류기가 샘플마다 여러 개의 클래스를 출력해야 할 때도 있습니다.
- 얼굴 인식 분류기에서 같은 사진에 여러 사람이 등장하면 어떻게 해야 할까요?
- 인식된 사람마다 레이블을 하나씩 할당해야 합니다. [1, 0, 1] (있음, 없음, 있음)
- 클래스별 가중치를 조절하고 싶으면 클래스의 지지도(Support)를 가중치로 주면 됩니다.