분류 기법

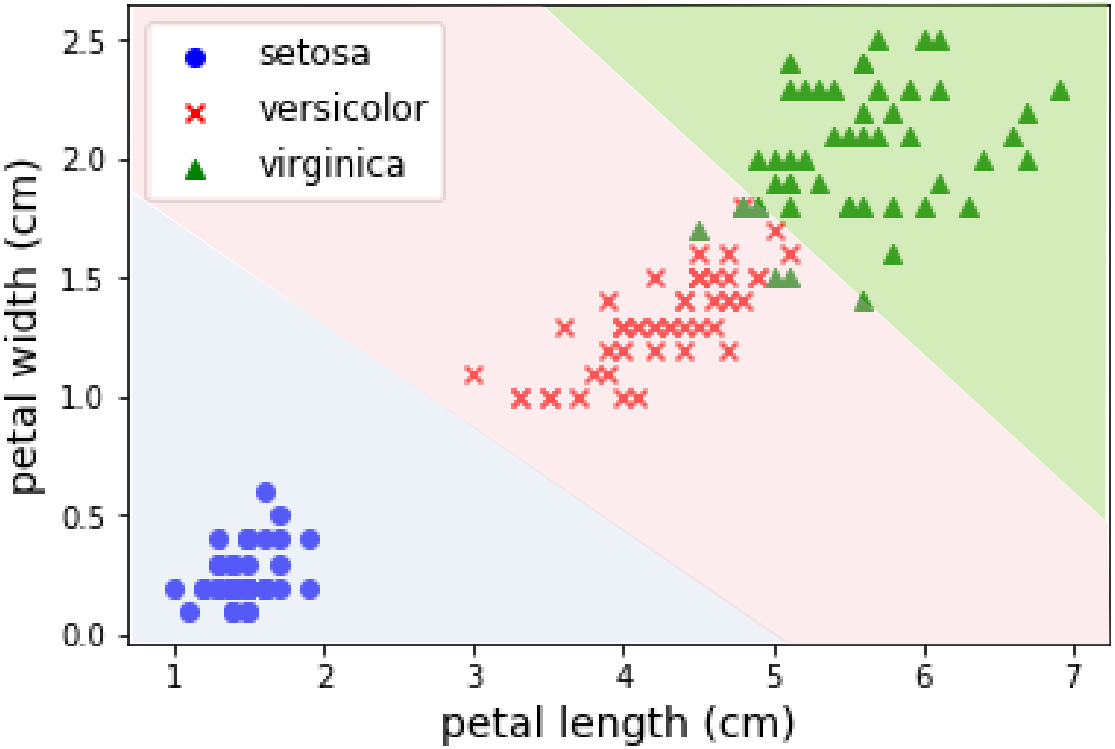
분류 기법

- 분류 (Classification)
  - 관측 데이터가 어떤 그룹(레이블, 클래스)에 속하는지 분석하는 행위
  - 이러한 분류 작업을 수행하는 기법 또는 학습 모형을 분류기(classifier)라고 한다.

※ 그림 출처: https://kr.123rf.com/profile_alazur

분류 기법

- 분류
  - 분류를 하기 위해서는 클래스에 대한 정보가 주어져야 한다.
  - 클래스는 상호 배타적인 그룹 명칭이며, 범주형 데이터 또는 이산형 데이터여야 한다.



데이터의 유형

- 범주형 (Categorical) 데이터
  - 여러 개의 범주로 나뉘어져 있는 데이터
  - 정성적(질적; qualitative) 자료라고 한다.

① 명목형 (nominal) 데이터
- 순서를 매길 수 없고, 단순히 분류를 할 수 있는 데이터
  예) 성별, 종교, 색상, 혈액형

② 순서형 (ordinal) 데이터
- 범주 간의 순서 또는 서열이 존재하는 데이터
- 일반적으로 수치 연산이 무의미하거나 불가능하다.
  예) 등급, 학점, 선호도

데이터의 유형

- 수치형 (Numeric) 데이터
  - 수치의 형태로 표현되어 측정이 가능한 데이터
  - 정량적(양적; quantitative) 자료라고 한다.

① 이산형 (discrete) 데이터
- 정수 등 특정한 형태로 표현되어 계수할 수 있는 데이터
  예) 인원 수, 진료 횟수, 사고 건수

② 연속형 (continuous) 데이터
- 논리적으로 계량할 수 있는 데이터
- 실수의 형태로서, 수치 연산의 산술적 의미가 있다.
  예) 키, 몸무게, 온도, 점수

분류 기법의 형태

- 이진 분류 (Binary Classification)
  - 두 개의 클래스를 구별하는 분류 기법이다.
  - 대표적으로 로지스틱 회귀, 서포트 벡터 머신(SVM; Support Vector Machine), 인공 신경망 학습 등이 있다.

- 다중 분류 (Multiclass Classification)
  - 세 개 이상의 클래스들을 구분하는 분류 기법으로서, 다항(multinomial) 분류라고도 한다.
  - 다항 로지스틱 회귀, 랜덤 포레스트, 나이브 베이즈 분류기 등이 있다.
  - 이진 분류 기법을 여러 개 조합하여 분류할 수도 있다.

분류의 성능 평가 지표

성능 평가 지표

- 정확도 (Accuracy)
  - 말 그대로, 예측한 결과가 실제 결과와 얼마나 동일한지를 나타내는 지표

$$\text{정확도} = \frac{\text{실제 결과와 동일한 예측 결과의 수}}{\text{전체 예측 결과의 수}}$$

실제 결과

발송인	검사 결과
홍길동	스팸
성춘향	정상
김철수	정상
이순신	스팸
유관순	스팸

예측 결과

발송인	검사 결과
홍길동	정상
성춘향	스팸
김철수	정상
이순신	스팸
유관순	정상

정확도 = 40%

성능 평가 지표

- 오차 행렬 (Confusion Matrix)
  - 4분면 행렬을 이용하여 실제 결과 값과 예측 결과 값이 얼마나 매칭되었는지를 표현한다.

실제 결과 \ 예측 결과	Positive	Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative

- TP : 실제로 Positive이고 예측도 Positive로 했다.
- FP : 실제로 Negative인데 예측을 Positive로 했다.
- FN : 실제로 Positive인데 예측을 Negative로 했다.
- TN : 실제로 Negative이고 예측도 Negative로 했다.

성능 평가 지표

- 오차 행렬 (Confusion Matrix)
  - 4분면 행렬을 이용하여 실제 결과 값과 예측 결과 값이 얼마나 매칭되었는지를 표현한다.

실제 결과

발송인	검사 결과
홍길동	스팸
성춘향	정상
김철수	정상
이순신	스팸
유관순	스팸

예측 결과

발송인	검사 결과
홍길동	정상
성춘향	스팸
김철수	정상
이순신	스팸
유관순	정상

		예측 결과: 스팸	예측 결과: 정상
실제 결과	스팸	TP 1개	FN 2개
실제 결과	정상	FP 1개	TN 1개

성능 평가 지표

- 정밀도 (Precision)
  - Positive로 예측한 결과들 중 실제로 Positive인 결과들의 비율

$$\text{정밀도} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

  - “정답으로 찾아낸 결과들 중에서 실제로 정답이 얼마나 많이 있는가?”

		예측 결과: 스팸	예측 결과: 정상
실제 결과	스팸	TP 1개	FN 2개
	정상	FP 1개	TN 1개

정밀도 = 50%

성능 평가 지표

- 재현율 (Recall)
  - 실제로 Positive인 결과들 중 Positive로 예측한 결과들의 비율

$$\text{재현율} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

  - "원래의 정답들 중에서
    실제로 정답으로 찾아낸 결과가 얼마나 많이 있는가?"

		예측 결과: 스팸	예측 결과: 정상
실제 결과	스팸	TP 1개	FN 2개
	정상	FP 1개	TN 1개

재현율 = 33.3%

성능 평가 지표

- F1 스코어 (F1 Score)
  - 정밀도와 재현율을 결합한 지표로서, 두 수치의 조화 평균

$$F1 = \frac{2}{\frac{1}{\text{정밀도}} + \frac{1}{\text{재현율}}} = 2 \times \frac{\text{정밀도} \times \text{재현율}}{\text{정밀도} + \text{재현율}}$$

		예측 결과: 스팸	예측 결과: 정상
실제 결과	스팸	TP 1개	FN 2개
	정상	FP 1개	TN 1개

정밀도 = 50%

재현율 = 33.3%

F1 = 40%

성능 평가 지표

- 사이킷런으로 성능 평가 지표 확인
  - metrics 모듈에 있는 함수들을 이용하여 성능 평가 지표를 도출한다.
    - 모든 함수들의 첫 번째 매개변수는 클래스의 실제값이고, 두 번째 매개변수는 클래스 예측값이다.

함수명	설명
accuracy_score	정확도를 계산한다.
confusion_matrix	오차 행렬을 도출한다.
precision_score	정밀도를 계산한다.
recall_score	재현율을 계산한다.
f1_score	F1 스코어를 계산한다.
classfication_report	정밀도, 재현율, F1 스코어를 함께 보여준다.

성능 평가 지표

- 사이킷런으로 성능 평가 지표 확인
  - 실제값과 예측값 데이터를 준비한다.

실제 결과

발송인	검사 결과
홍길동	스팸
성춘향	정상
김철수	정상
이순신	스팸
유관순	스팸

예측 결과

발송인	검사 결과
홍길동	정상
성춘향	스팸
김철수	정상
이순신	스팸
유관순	정상

```
1  actual_result = [1, 0, 0, 1, 1]
2  predicted_result = [0, 1, 0, 1, 0]
```

'스팸'이면 양성 (클래스1)
'정상'이면 음성 (클래스0)

성능 평가 지표

- 사이킷런으로 성능 평가 지표 확인
  - 함수들을 호출하여 성능 평가 지표를 구한다.

```
1  import sklearn.metrics as mt
2
3  accuracy = mt.accuracy_score(actual_result, predicted_result)
4
5  matrix = mt.confusion_matrix(actual_result, predicted_result)
6
7  precision = mt.precision_score(actual_result, predicted_result)
8
9  recall = mt.recall_score(actual_result, predicted_result)
10
11 flscore = mt.f1_score(actual_result, predicted_result)
12
13 scores = mt.classification_report(actual_result, predicted_result)
```

성능 평가 지표

- 사이킷런으로 성능 평가 지표 확인
  - 성능 평가 지표 값들을 확인한다.

```
1 print("정확도:", accuracy, "₩n")
2 print("오차행렬₩n", matrix, "₩n")
3 print("정밀도:", precision, "₩n")
4 print("재현율:", round(recall, 3), "₩n")
5 print("F1스코어:", f1score, "₩n")
```

```
정확도: 0.4

오차행렬
[[1 1]
[2 1]]

정밀도: 0.5

재현율: 0.333

F1스코어: 0.4
```

오차 행렬의 행과 열은 클래스 번호 순서대로 표시된다.
즉, 첫 번째 행은 클래스0이고 두 번째 행은 클래스1이다.
마찬가지로 첫 번째 열은 클래스0이고 두 번째 열은 클래스1이다.

성능 평가 지표

- 사이킷런으로 성능 평가 지표 확인
  - 성능 평가 지표 값들을 확인한다.

```
1  print("결과₩n", scores)
```

결과

	precision	recall	f1-score	support
0	0.33	0.50	0.40	2
1	0.50	0.33	0.40	3
micro avg	0.40	0.40	0.40	5
macro avg	0.42	0.42	0.40	5
weighted avg	0.43	0.40	0.40	5