# 3장 분류 (2부)

## 감사의 글

자료를 공개한 저자 오렐리앙 제롱과 강의자료를 지원한 한빛아카데미에게 진심어린 감사를 전합니다.

## 주요 내용

- 다중 클래스 분류
- 에러 분석
- 다중 레이블 분류
- 다중 출력 분류

# 3.4 다중 클래스 분류

## 다중 클래스 분류기(multiclass classifier)

- 세 개 이상의 클래스로 샘플을 분류하는 예측기
- 다항 분류기(multinomial classifier)라고도 부름
- 예를 들어, 손글씨 숫자 분류의 경우 0부터 9까지 10개의 클래스로 분류해야 함

## 다중 클래스 분류 지원 분류기

- SGD 분류기
- 랜덤 포레스트 분류기
- 나이브 베이즈(naive Bayes) 분류기

## 이진 분류만 지원하는 분류기

- 로지스틱 회귀
- 서포트 벡터 머신
    - 사이킷런의 `SVC()` 모델은 다중 클래스 분류도 특별한 기법을 적용하여 지원함.

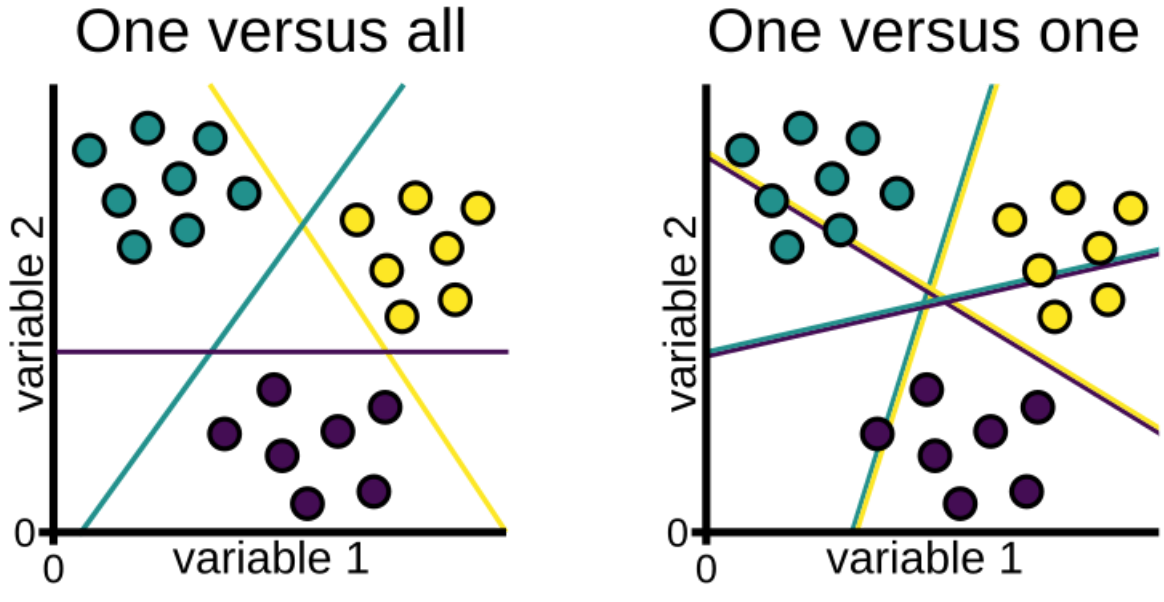
## 이진 분류기 활용

- 이진 분류기를 활용하여 다중 클래스 분류 가능
    - 일대다(OvR 또는 OvA)
    - 일대일(OvO)

- 일대다 방식 활용 예제
    - 숫자 5 예측하기에서 사용했던 이진 분류 방식을 동일하게 모든 숫자에 대해서 실행
    - 각 샘플에 대해 총 10번 각기 다른 이진 분류기를 실행
    - 이후 각 분류기의 결정 점수 중에서 가장 높은 점수를 받은 클래스를 해당 샘플의 클래스로 선택

- 일대일 방식 활용 예제
    - 조합 가능한 모든 클래스 일대일 분류 방식을 진행하여 가장 많은 결투(duell)를 이긴 숫자를 선택
    - MNIST의 경우, 0과 1 구별, 0과 2 구별, ..., 1과 2 구별, 1과 3 구별, ..., 8과 9 구별 등 (9+8+...+1 = 45)개의 결투를 판별하는 45개의 분류기 활용. 단, 각 결투에 해당되는 데이터 샘플만 훈련에 사용됨.
    - 각각의 훈련 샘플에 대해 가장 많은 결투를 이긴 숫자의 클래스를 예측값으로 사용함. 예를 들어, 숫자 1이 9번의 결투를 모두 이기면 숫자 1을 예측값으로 지정함.

일대다 방식 vs. 일대일 방식




<그림 출처: SVM with the mlr package>

## 예제: 서포트 벡터 머신

- 훈련 세트의 크기에 민감하여 작은 훈련 세트에서 많은 분류기를 훈련시키는 쪽이 훨씬 빠름. 따라서 다중 클래스 분류에 일대일 전략을 사용함.
- 대부분의 이진 분류기는 일대다 전략을 선호

## 일대일 또는 일대다 전략 선택

- 이진 분류기를 일대일 전략 또는 일대다 전략으로 지정해서 학습하도록 만들 수 있음.
- 사이킷런의 경우: `OneVsOneClassifier` 또는 `OneVsRestClassifier` 사용
- 예를 들어, SVC 모델을 일대다 전략으로 훈련시키려면 `OneVsRestClassifier` 활용

```
from sklearn.multiclass import OneVsRestClassifier
  ovr_clf = OneVsRestClassifier(SVC())
  ovr_clf.fi(X_train, y_train)
```

## 다중 클래스 지원 분류기

- `SGDClassifier` 또는 `RandomForestClassifier`는 다중 클래스 분류를 직접 지원함.
- 따라서 사이킷런의 OvR, OvO 등을 적용할 필요 없음

## 다중 클래스 분류기 성능 측정

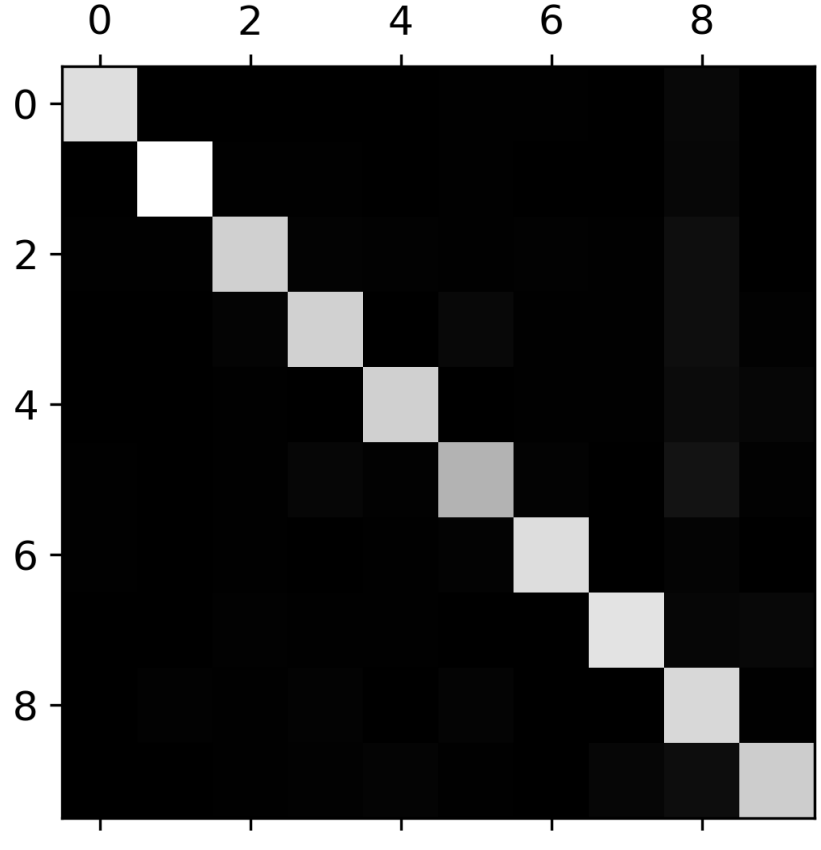
- 다중 클래스 분류기의 성능 평가는 교차검증을 이용하여 정확도를 측정
- MNIST의 경우 0부터 9까지 숫자가 균형 있게 분포되어 있어서 데이터 불균형의 문제가 발생하지 않음.

## 3.5 에러 분석

가능성이 높은 모델을 하나 찾았을 때 에러 분석을 통해 모델의 성능을 향상시킬 방법을 찾아볼 수 있음.

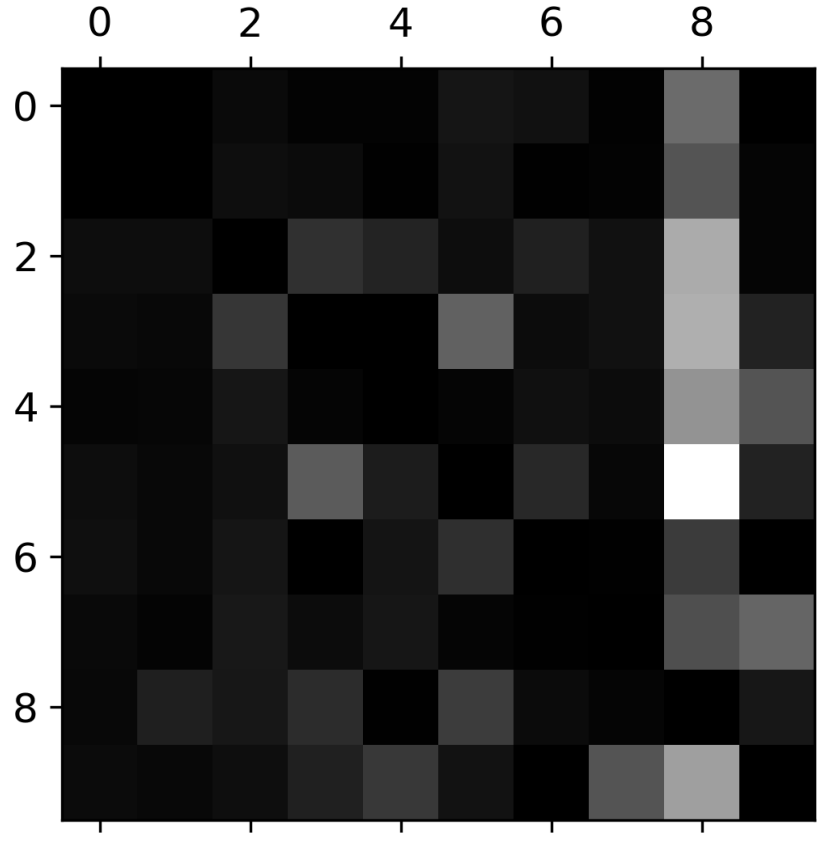
# 오차 행렬 활용

- 손글씨 클래스 분류 모델의 오차 행렬을 이미지로 표현 가능
- 대체로 잘 분류됨: 대각선이 밝음.
- 5행은 좀 어두움. 숫자 5의 분류 정확도가 상대적으로 낮음

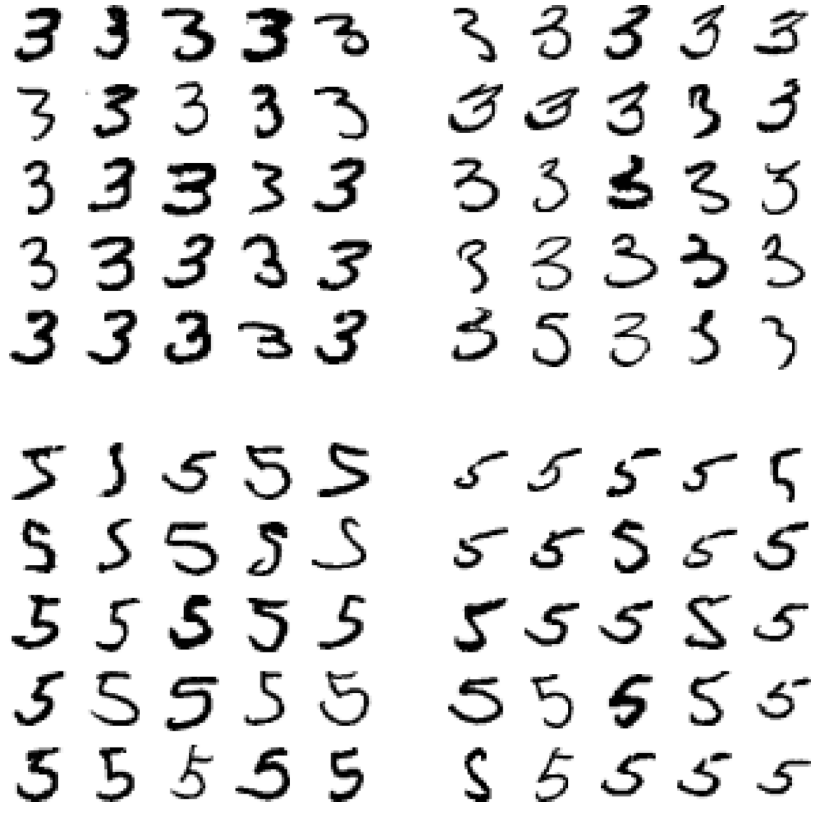


## 오차율 이미지

- 8행이 전반적으로 어두움. 즉, 8은 잘 분류되었음.
- (3, 5)와 (5,3)의 위치가 상대적으로 밝음. 즉, 3과 5가 서로 많이 혼동됨.



- 3과 5의 오차행렬 그려보기
    - 음성: 3으로 판정
    - 양성: 5로 판정

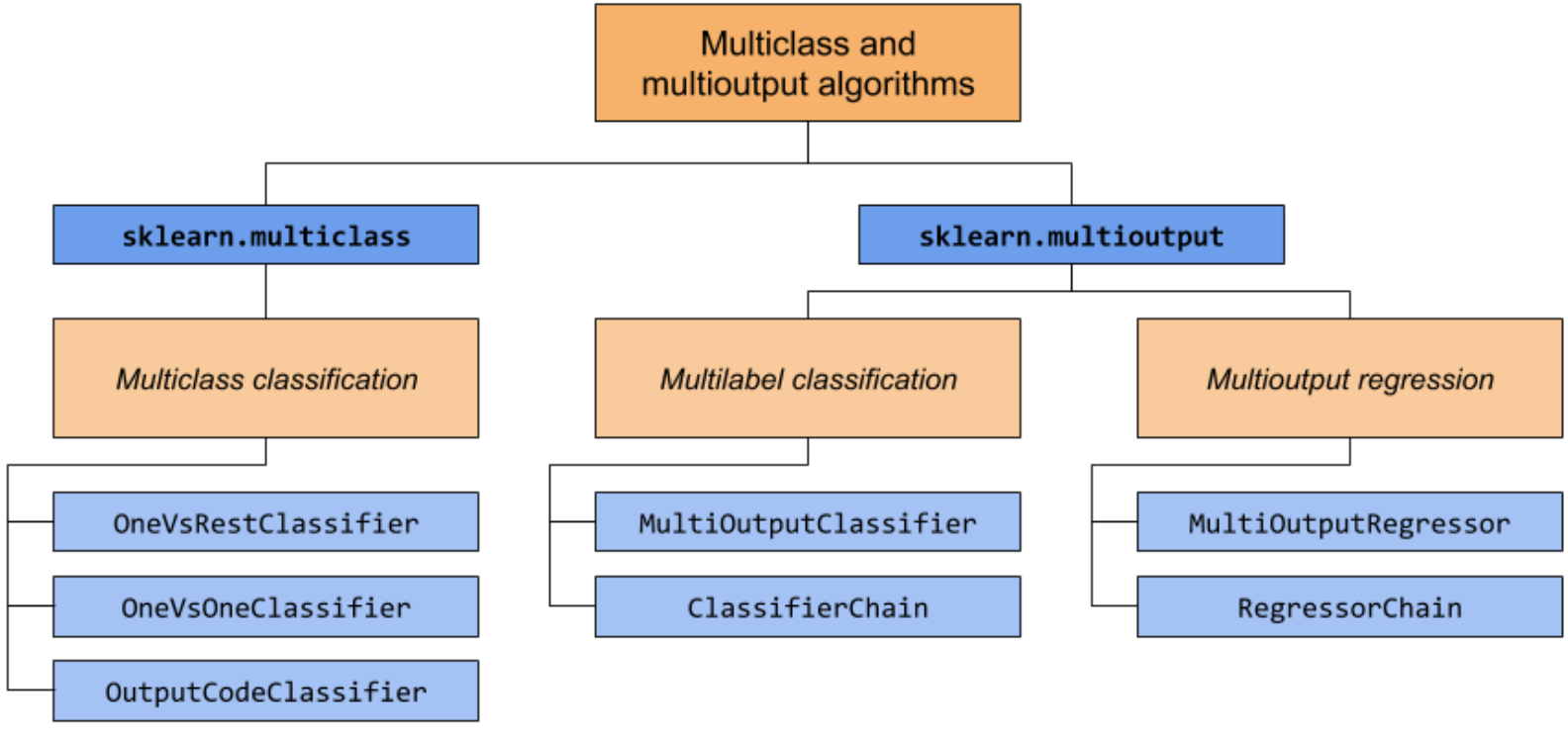


- 3과 5의 구분이 어려운 이유
    - 선형 모델인 SGD 분류기를 사용했기 때문
    - SGD 모델은 단순히 픽셀 강도에만 의존함.
- 이미지 분류기의 한계
    - 이미지의 위치나 회전 방향에 민감함
    - 이미지를 중앙에 위치시키고 회전되지 않도록 전처리하거나, 8은 동그라미가 두 개 있다는 등 각 숫자의 특성을 추가하면 더 좋은 성능의 모델 구현 가능함.

## 다중 클래스 분류 일반화

- 다중 레이블 분류(multilabel classification)
- 다중 출력 분류(multioutput classification)

사이킷런의 다중 클래스와 다중 출력 알고리즘




<이미지 출처: 사이킷런: 다중 클래스와 다중 출력 알고리즘>

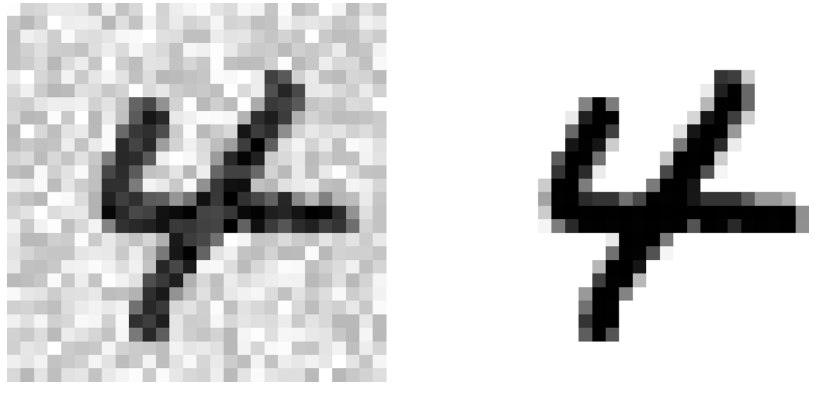
# 3.6 다중 레이블 분류

- 샘플마다 여러 개의 클래스 출력
- 예제: 얼굴 인식 분류기
    - 한 사진에 여러 사람이 포함된 경우, 인식된 사람마다 하나씩 꼬리표(tag)를 붙여야 함.
    - 앨리스, 밥, 찰리의 포함여부를 확인 할 때: 밥이 없는 경우 `[True, False, True]` 출력
- 다중 레이블 분류기를 평가하는 방법은 다양함
    - 예를들어, 각 레이블의 $F_1$ 점수를 구하고 레이블에 대한 가중치를 적용한 평균 점수 계산
    - 가중치 예제: 타깃 레이블에 속한 샘플 수를 가중치로 사용 가능. 즉, 샘플 수가 많은 클래스의 가중치를 보다 크게 줄 수 있음.

# 3.7 다중 출력 분류

- 다중 출력 다중 클래스 분류라고도 불림
- 다중 레이블 분류에서 한 레이블이 이진 클래스가 아닌 다중 클래스를 대상으로 예측하는 분류

- 예제: 이미지에서 잡음을 제거하는 시스템
    - 다중 레이블: 각각의 픽셀에 대해 레이블 예측해야 함.
    - 다중 클래스: 각각의 픽셀에서 예측하는 레이블이 0부터 255 중에 하나임.



- 아래 사진: 분류기가 예측한 이미지