기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 10. 성능 평가

정 정 민

programmers

Chapter 23. 교차 검증

1. 교차 검증이 뭐고, 이걸 왜 할까요?
2. 다양한 교차 검증 방법
3. 과거 모델을 교차 검증 하기

교차 검증이 뭐고, 이걸 왜 할까요?

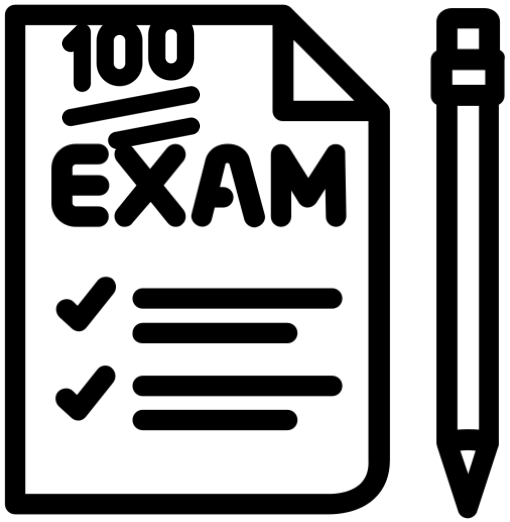
검증에 대한 고찰과 우려

- ‘검증 (validation)’이라는 단어를 언제 사용했을까요? >> 검증 데이터 (validation data)

- 검증이란,
  - 모델의 학습이 잘 진행되었는지(일반화 능력)를 판단하는 평가 과정
  - 학습이 잘 되고있는지 혹은 과적합이 진행되는지를 판단
  - 모델 학습의 최종 의사 결정 과정에서 사용되는 중요한 역할

- 검증은 수험생에게 모의고사와 비슷한 역할
- 하지만 모의고사 너무 쉽다면??
  - 학습의 과정과 결과를 명확히 확인할 수 없고
  - 최종 시험에서 좋은 결과를 얻을 수 없음

- 이런 비슷한 일이 머신 러닝 모델 학습 과정에서 일어날 수 있음
  - 우연히도 너무 쉬운 데이터가 검증 데이터로 구성될 수 있음
  - 이는 많은 데이터를 제공하면 되지만.. 그럴 만큼 데이터가 적다면???

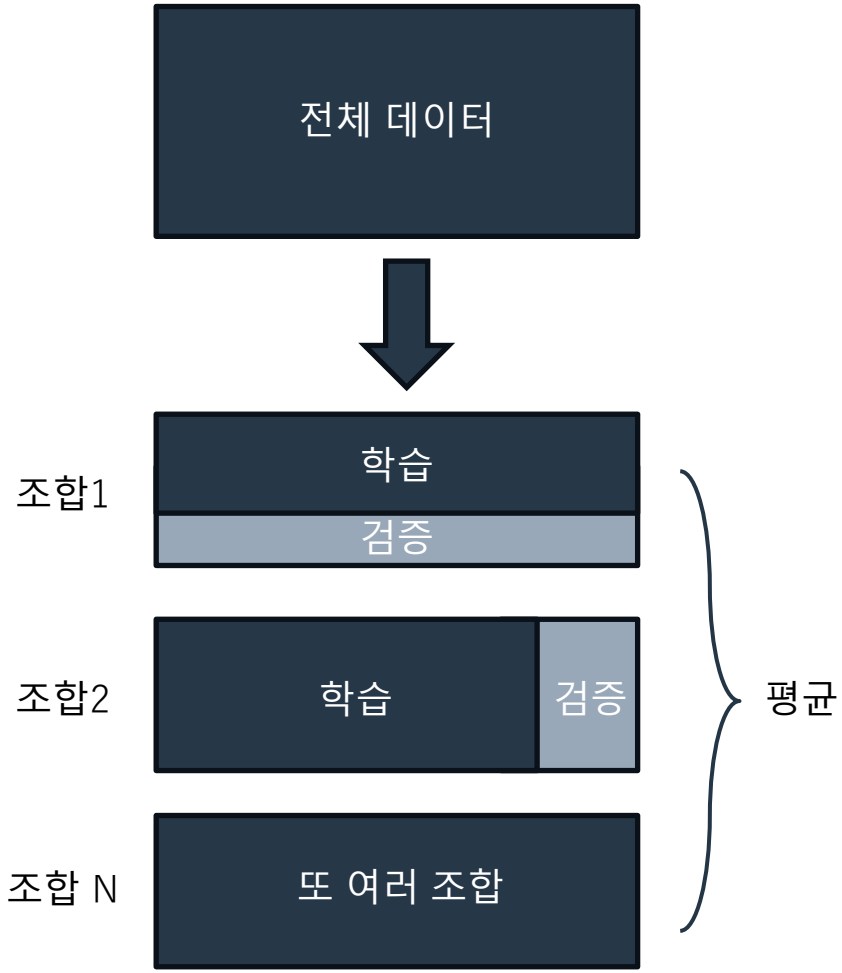


교차 검증

- 앞선 문제를 회피 혹은 감수 하면서도 검증의 원래 의미를 살리는 평가를 진행하는 방법 : 교차 검증 (Cross Validation)
  - 앞선 문제
    - 쉬운 데이터로의 편향
    - 전체적인 데이터 양의 부족
  - 검증의 원래 의미 : 일반화 능력 판단

- 교차 검증이란,
  - 전체 데이터를 여러 개의 하위 데이터로 나누고
  - 이 하위 세트들의 조합은 서로 다른 방법으로 훈련과 검증에 사용해
  - 모델의 일반화 능력을 충분히 측정하는 것을 의미

- 교차 검증은 시간 복잡도 측면을 제외하고
  일반적인 랜덤 검증 데이터 활용 검증 방법보다 좋은 방법
  - 일반화 능력 추정, 데이터 활용 최대화, 과적합 방지



다양한 교차 검증 방법

다양한 교차 검증(CV) 방법

- 학습 데이터와 검증 데이터를 구성하는 방법에 따라 여러 방법이 존재

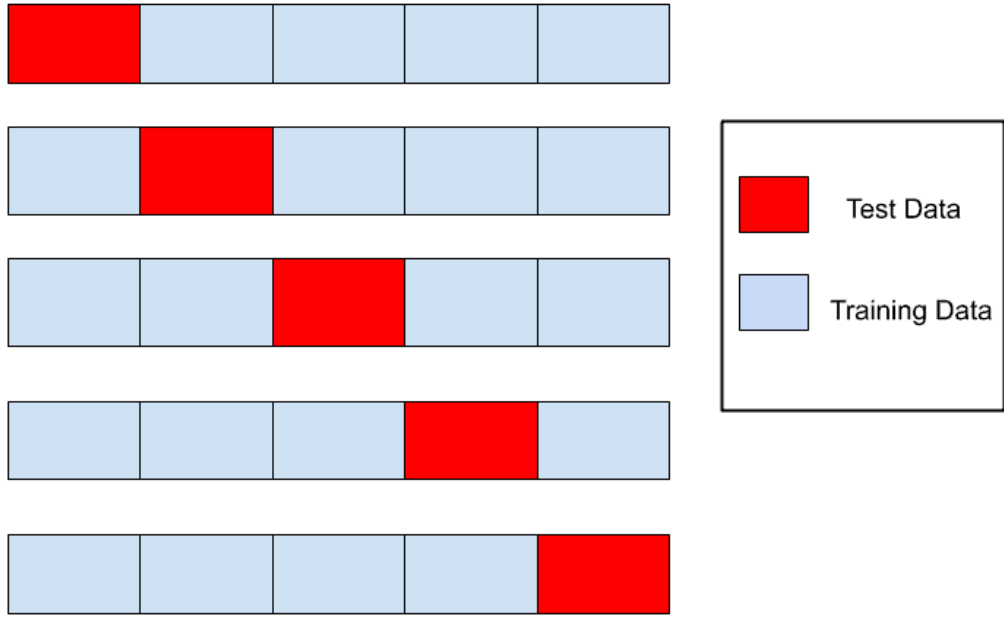
- K-Fold CV
- 계층적 교차 검증 (Stratified Cross-Validation)
- LOOCV (Leave-One-Out Cross Validation)

K-Fold CV

- 전체 데이터 세트를 총 K개의 덩어리(폴드, Fold)로 나누고, 각 덩어리(폴드)를 순차적으로 검증 데이터로 사용하는 방법

- 과정
  - 데이터 세트를 K개의 폴드로 나눔 → K 개의 조합 생성
  - 하나의 조합 당
    - 하나의 폴드만 Validation data로 사용
    - 나머지 K-1개의 폴드를 Train data로 사용
  - 총 K번의 학습 및 평가 과정이 반복

- 의미
  - 모든 데이터가 학습 및 평가로 사용됨
    - 데이터 활용의 극대화
    - 과적합 방지
    - 논리적인 일반화 평가 진행



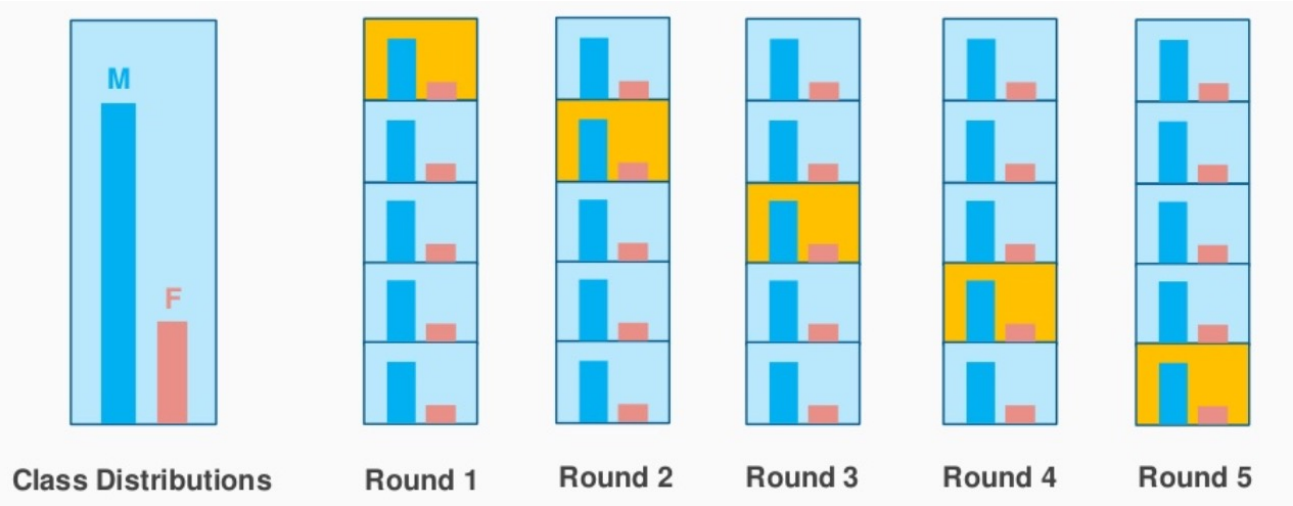
https://www.mltut.com/k-fold-cross-validation-in-machine-learning-how-does-k-fold-work/

계층적 교차 검증 (Stratified Cross-Validation)

- K-Fold CV와 유사하지만, 각 폴드에서 클래스의 비율을 원본 데이터셋의 클래스 비율과 유사하게 유지

- 과정
  - 클래스 별로 데이터를 분할
  - 각 클래스 데이터를 K 개의 폴드로 나눔
  - 각 클래스에 존재하는 K 개의 폴드를 하나씩 조합 → K 개의 조합 생성
  - 이후 과정은 K-Fold CV 와 동일

- 의미
  - K-Fold의 장점과 더불어
  - 클래스 사이의 불균형이 있는 경우의 편향까지 고려



https://stats.stackexchange.com/questions/49540/understanding-stratified-cross-validation

LOOCV (Leave-One-Out Cross Validation)

- 한 번에 하나의 데이터 포인트만을 검증 데이터로 사용
- 과정
  - 극단적인 K-Fold CV의 경우로
  - 전체 데이터 수 만큼의 K를 활용
- 의미
  - 매우 정확한 검증 방식
  - 하지만 데이터의 크기가 크다면 매우 많은 시간이 소요
  - 작은 데이터셋에 유용

과거 모델을 교차 검증 하기

강의 실습 코드 : 링크

선형 분류 모델을 이용해 교차 검증하기

- Logistic 회귀 모델을 활용한 선형 분류 모델을 이용해
- 교차 검증 진행
  - K-Fold CV, Stratified CV, LOOCV

```
from sklearn.model_selection import cross_val_score,
                                    StratifiedKFold,
                                    LeaveOneOut
from sklearn.metrics import accuracy_score

kfold_score = np.mean(cross_val_score(logistic_reg, X, y, cv=5))
strat_score = np.mean(cross_val_score(logistic_reg, X, y, cv=StratifiedKFold(5)))
loocv_score = np.mean(cross_val_score(logistic_reg, X, y, cv=LeaveOneOut()))
```

E.O.D