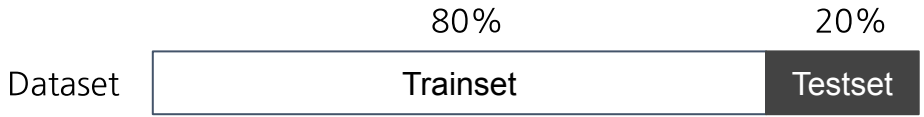
# 교차 검증 (Cross Validation)

# 교차 검증

- 학습과 검증 사이 - Trainset vs Testset
  - ➡️ 학습할 때 보지 않았던 데이터를 이용해 검증한다.
  - ➡️ 일반적으로 학습 데이터와 테스트 데이터의 비율은 8:2

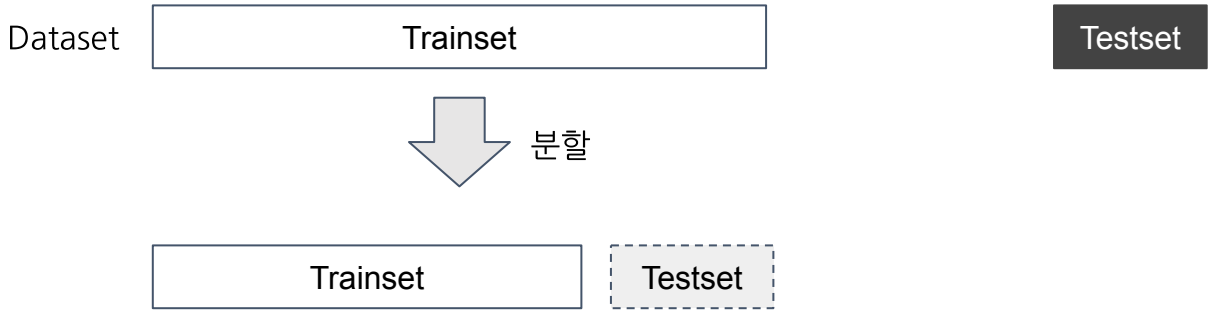



- 고정된 test set을 가지고 모델의 성능을 확인하고 파라미터를 수정하고, 이 과정을 반복하면 결국 내가 만든 모델은 test set에만 잘 동작하는 모델이 된다. 이 경우 test set에 과적합되는 결과.

# 교차 검증

학습 데이터를 다시 분할하여 학습 데이터와 학습된 모델의 성능을 일차 평가하는 검증 데이터로 나눔

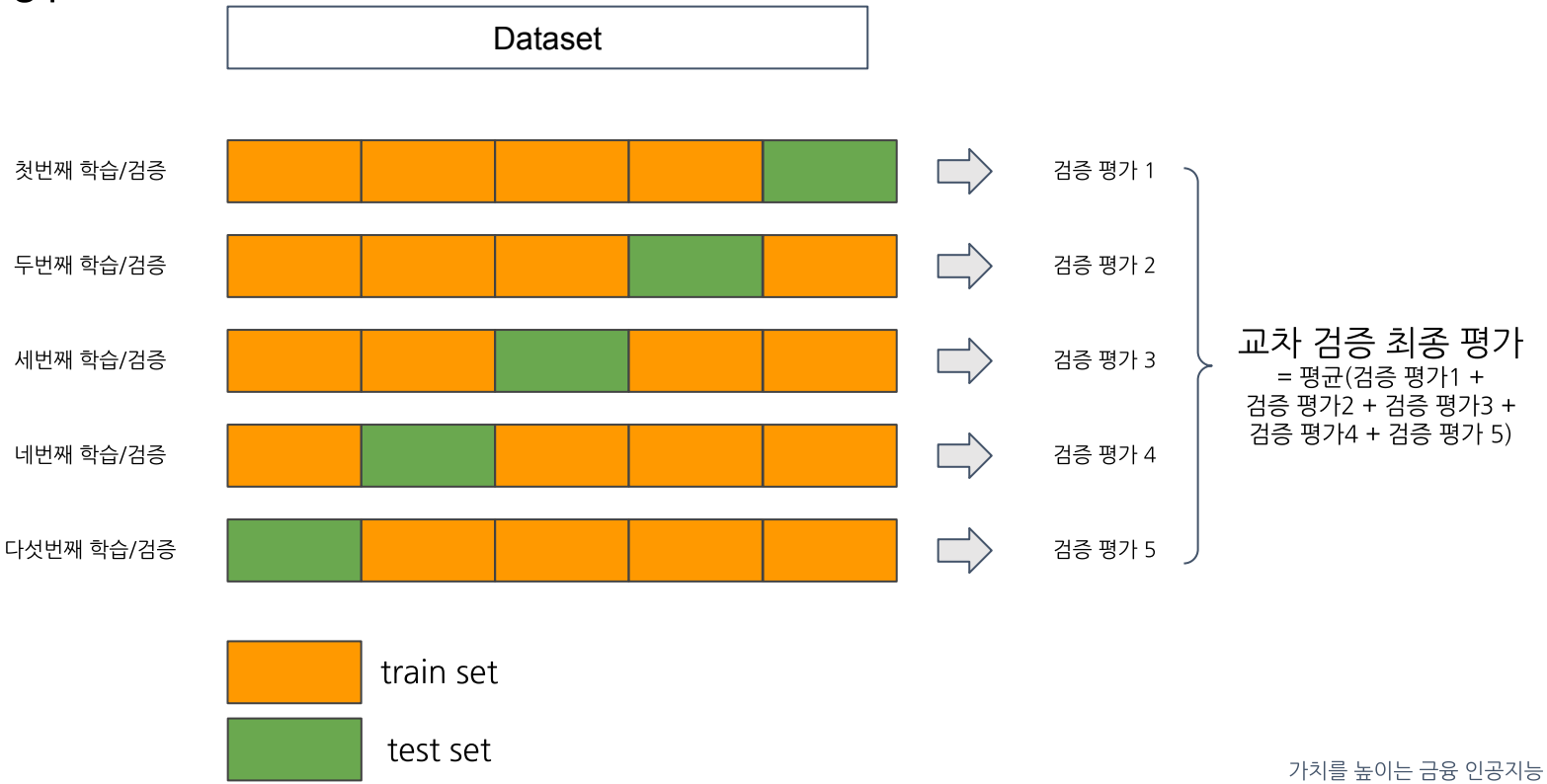
모든 학습/검증 과정이 완료된 후 최종적으로 성능을 평가하기 위한 데이터 세트




- But, 학습데이터가 편중되어 있다면?

# K 폴드 교차 검증

K = 5인 경우



# K 폴드 교차 검증

- 일반 K 폴드

- Stratified K 폴드
  - 불균형한 (imbalanced) 분포도를 가진 레이블 데이터 집합을 위한 K 폴드 방식

    예) 대출 사기 데이터
    대출 사기 건수가 전체에 비해 아주 작은 확률로 분포되어 있는데
    한 곳에 몰리면?
    ➡️ 사기 건수가 한 건도 없는 학습/테스트 데이터 세트 존재 가능

  - 학습 데이터와 검증 데이터 세트가 가지는 레이블 분포도가 유사하도록 검증 데이터 추출

# 교차 검증(Cross Validation)

## 실습

k fold

# 사이킷런 교차 검증

- KFold 클래스를 이용한 교차 검증 방법

1. 폴드 세트 설정
2. For 루프에서 반복적으로 학습/검증 데이터 추출 및 학습과 예측 수행
3. 폴드 세트 별로 예측 성능을 평균하여 최종 성능 평가

cross_val_score()

폴드 세트 추출, 학습/예측, 평가를 한번에 수행

cross_val_score(estimator, X, y=None, scoring=None, cv=None, n_jobs=1, verbose=0, fit_params=None, pre_dispatch='2*n_jobs')

# GridSearchCV

- 교차 검증과 최적 하이퍼 파라미터 튜닝을 동시에
  - : Classifier나 Regressor와 같은 알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력하면서 최적의 파라미터를 도출할 수 있다

```
grid_parameters = { 'max_depth':[1, 2, 3], 'min_sample_spilt:[2,3] }
```

총 파라미터 수 : 3 * 2 = 6

```
cv =3
```

총 학습/검증 총 수행 횟수 6 * 3 = **18**

# 교차 검증(Cross Validation)

## 실습

cross_val_score,
GridSearchCV