보통은 train set으로 모델을 훈련하고, test set으로 모델을 검증한다. 이때, 고정된 test set을 통해 모델의 성능을 검증하고 수정하는 과정을 반복하면 결국 이 모델은 test set에만 잘 동작하는 모델이 된다. 즉, test set에 overfitting하게 되는데 이를 해결하고자 하는 것이 cross validation이다. cross validation은 train set을 train set + validation set으로 분리한 뒤, validation set을 사용해 검증하는 방식이다.

cross validation은 모든 데이터셋을 훈련에 활용할 수 있다. 정확도를 높이며, 데이터 부족으로 인한 underfitting을 방지할 수 있다. 또한, 모든 데이터셋을 평가에 활용할 수 있다. 평가에 사용되는 데이터 편중을 막을 수 있고, 평과 결과에 따라 좀 더 일반화된 모델을 만들 수 있다. 그러나, iteration 횟수가 많기 때문에, 모델 훈련 및 평가 시간이 오래걸린다는 단점이 있다.

K-Fold는 가장 일반적으로 사용되는 cross validation 방법이다. K는 폴드의 수를 나타내고 보통 5 또는 10을 사용한다. 과정은 다음과 같다.

1. 전체 데이터셋을 training set과 test set으로 나눈다.

2. Training set를 training set + validation set으로 사용하기 위해 k개의 폴드로 나눈다.

3. 첫번째 폴드를 validation set으로 사용하고 나머지 폴드들을 training set으로 사용한다.

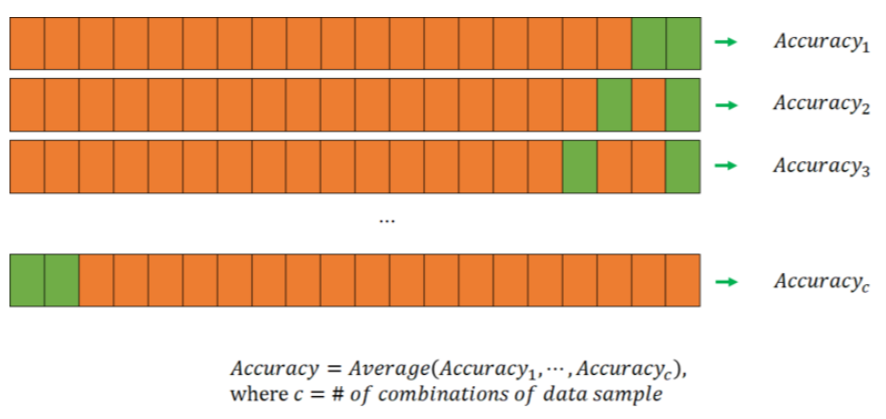
4. 모델을 training한 뒤, 첫 번째 validation set으로 평가한다.

5. 차례대로 다음 폴드를 validation set으로 사용하며 3번을 반복한다.

6. 총 k개의 성능 결과가 나오며, 이 k개의 평균을 해당 학습 모델의 성능이라고 한다.



Leave-p-out은 전체 데이터 중에서 p개의 샘플을 선택하여 그것을 모델 검증에 사용하는 방법이다. 따라서 validation set을 구성할 수 있는 경우의 수는 다음과 같다. k-fold방법과 마찬가지로, 각 데이터 폴드 세트에 대해서 나온 검증 결과들을 평균내어 도출하는 것이 일반적이다.

****

Leve-one-out은 줄여서 LOOCV라고 불리우며, Leave-p-out에서 p=1일 때의 경우를 말한다. 폴드 하나에 샘플 하나만 들어있는 k-fold 검증으로 생각할 수 있다. Leave-p-out보다 계산 시간에 대한 부담은 줄어들고, 더 좋은 결과를 얻을 수 있기 때문에 선호된다. 검증에 사용되는 validation set의 개수가 적은 만큼 모델 훈련에 사용되는 데이터의 개수는 늘어난다. 모델 검증에 희생되는 데이터의 개수가 단 하나이기 때문에, 나머지 모든 데이터를 모델 훈련에 사용할 수 있다는 것이 장점이다.

