Data split

Taehoon Ko (taehoonko@snu.ac.kr)

목표

- **다음 세 데이터셋의 역할을 이해한다.**
  - 학습 데이터셋 (training set)
  - 검증/개발 데이터셋 (validation / development set)
  - 테스트 데이터셋 (test set)

- **여러 데이터 분할 방법에 대해 이해한다.**
  - 2-way and 3-way holdout
  - k-fold cross-validation(*)

Sebastian Raschka's great posts (translated by 박해선)

- I strongly recommend to read Sebastian Raschka's posts.

  ➔ "Model evaluation, model selection, and algorithm selection in machine learning"

  - Part I - The basics
    - https://sebastianraschka.com/blog/2016/model-evaluation-selection-part1.html
    - https://tensorflow.blog/머신-러닝의-모델-평가와-모델-선택-알고리즘-선택-1/
  - Part II - Bootstrapping and uncertainties
    - https://sebastianraschka.com/blog/2016/model-evaluation-selection-part2.html
    - https://tensorflow.blog/머신-러닝의-모델-평가와-모델-선택-알고리즘-선택-2/
  - Part III - Cross-validation and hyperparameter tuning
    - https://sebastianraschka.com/blog/2016/model-evaluation-selection-part3.html
    - https://tensorflow.blog/머신-러닝의-모델-평가와-모델-선택-알고리즘-선택-3/

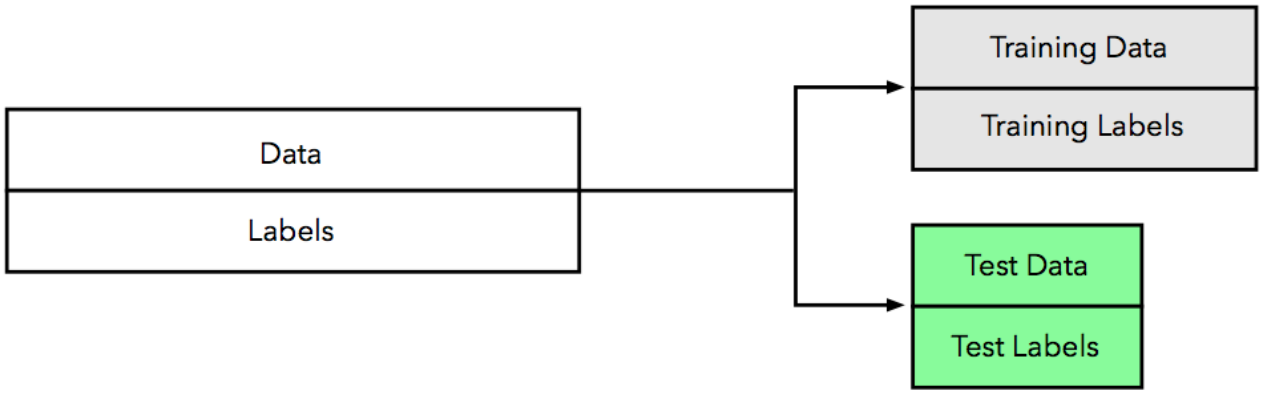
Training / Validation / Test set

- Training set
  - 모델 생성, 학습에 이용
- Validation set (Development set)
  - 모델의 오버피팅 방지
  - 모델의 복잡도 축소
  - 모델의 파라미터 탐색
- Test set
  - 모델의 예측 성능 (predictive performance) 평가

Holdout method: Training / Test set

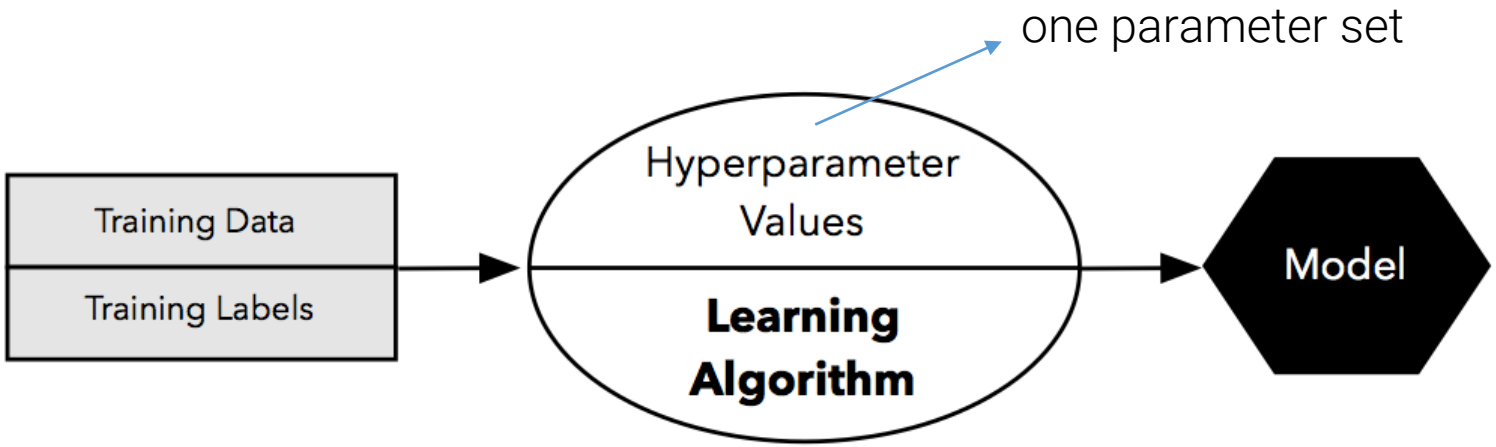
- 데이터를 training set과 test set 두 개로 분리

1



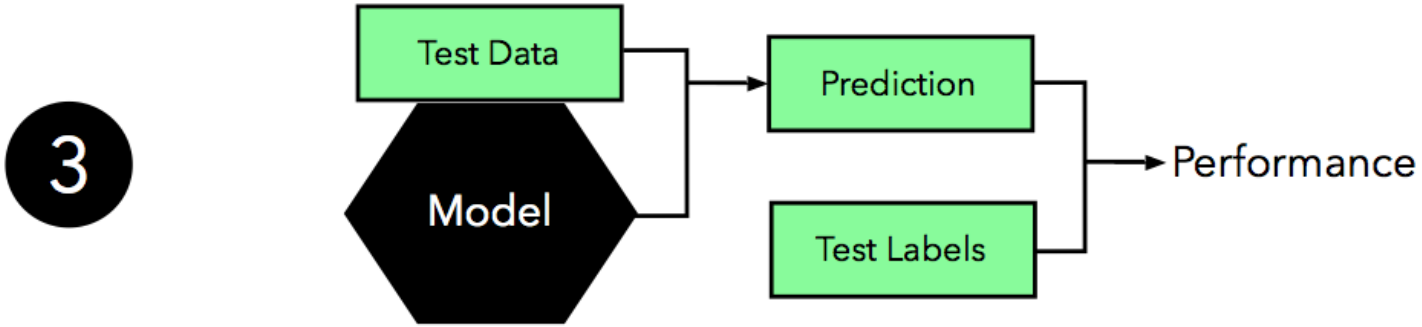
- Training set으로 모델을 학습

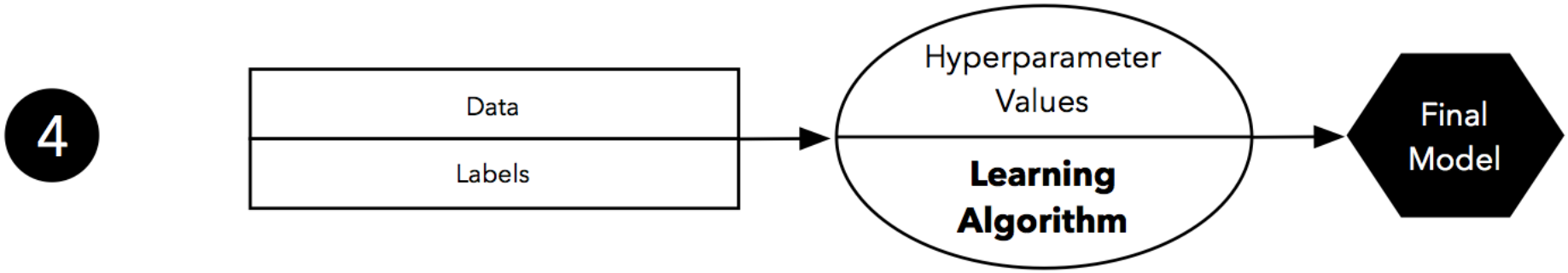
2



Holdout method: Training / Test set

- Test set으로 모델 평가





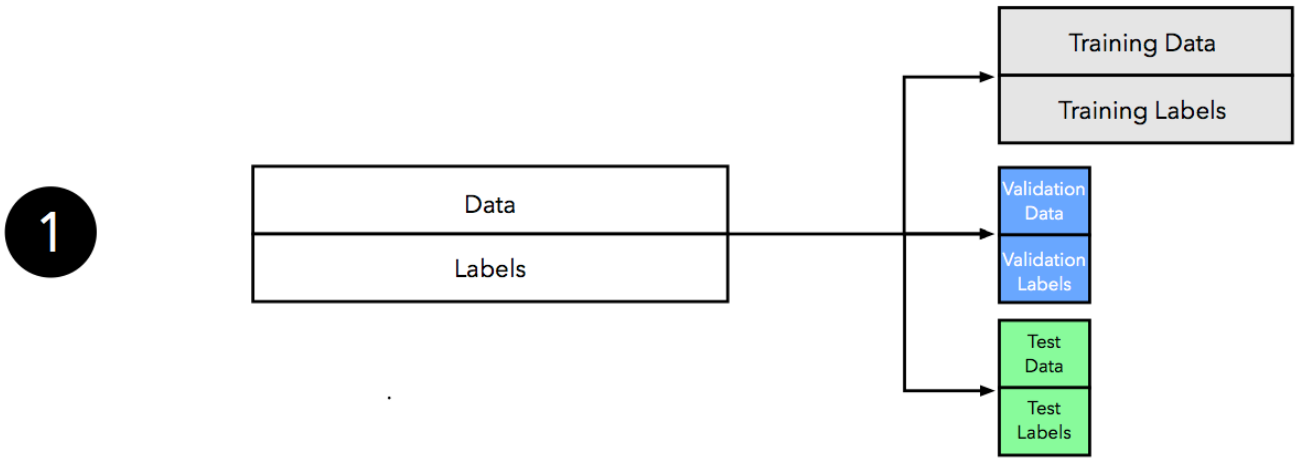
Holdout method: Training / Test set

- Pros
  - 모델 생성에 오랜 시간이 소요되지 않음
    - 파라미터 고민 없이 모델을 생성하고 그 모델의 특성을 파악하기에 적합

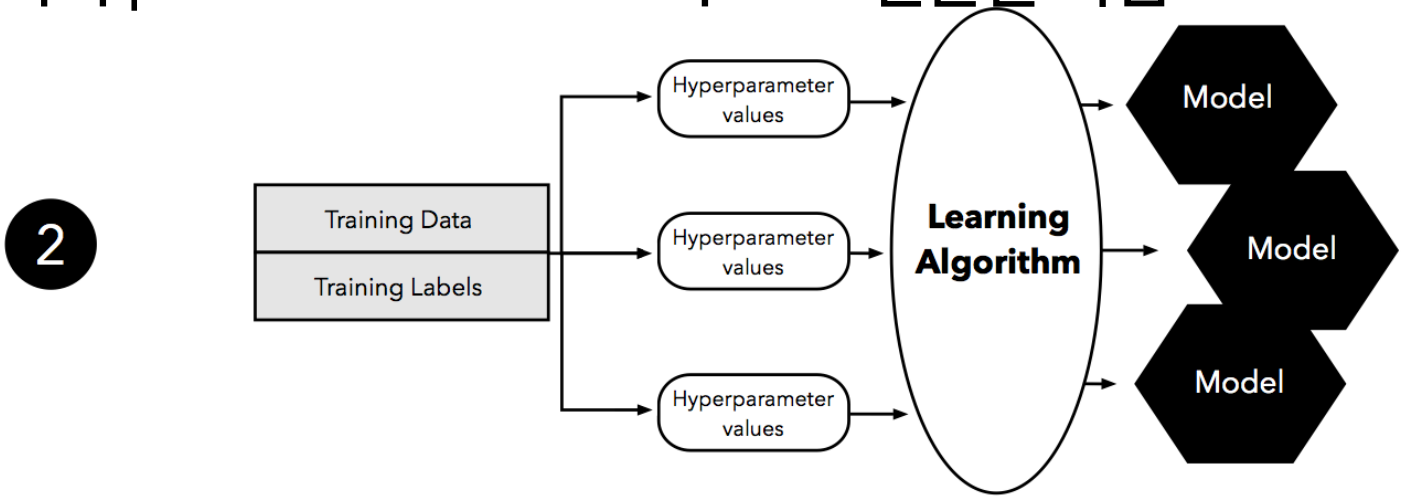
- Cons
  - 모델 파라미터 셋을 데이터를 이용하여 탐색할 수 있는 기회가 없음
  - 만약 파라미터 셋을 테스트 데이터에서의 성능을 이용하여 탐색한다면?
    (test set이 본연의 목적과 다르게 validation set의 목적으로 사용되는 것)
    - Training set is used to learn the model.
    - Test set is used to tuning parameters of the model.
    - Then, how can we calculate the predictive performance (or generalization performance) of the model?

3-way holdout method: Training / validation / test set

- 데이터를 training set, validation set, test set으로 분리



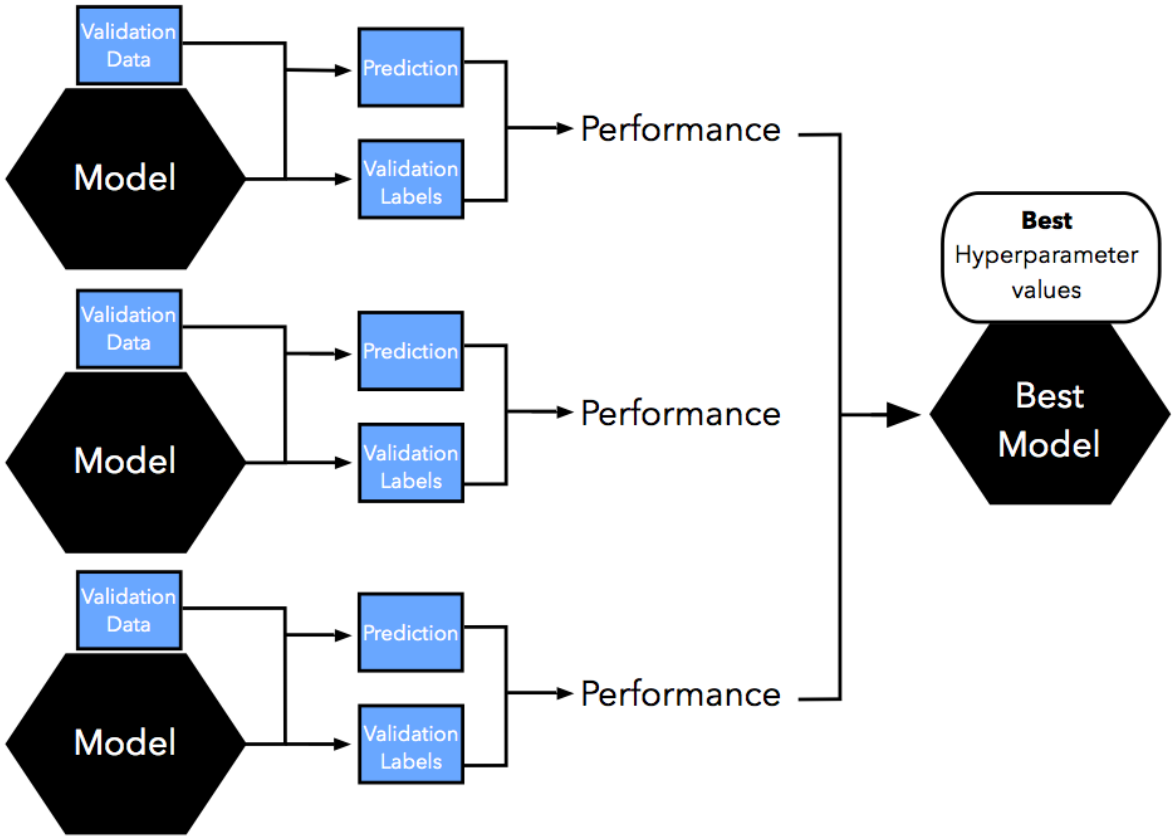
- 여러 parameter sets으로 후보 모델들을 학습



3-way holdout method: Training / validation / test set

- Validation set을 이용하여 모델들을 평가하고, 최적의 parameter set을

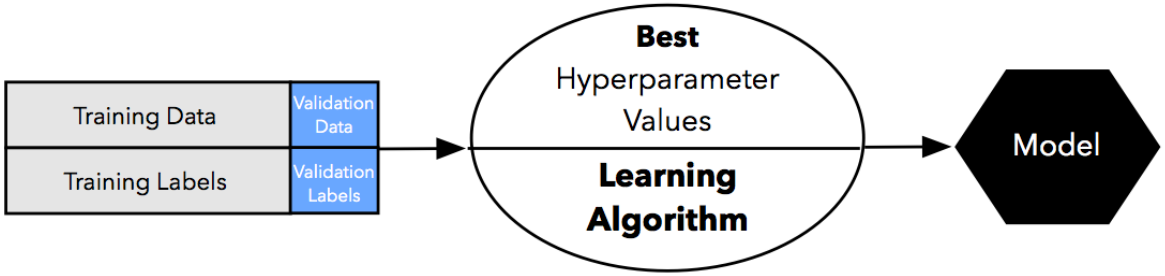
3



3-way holdout method: Training / validation / test set

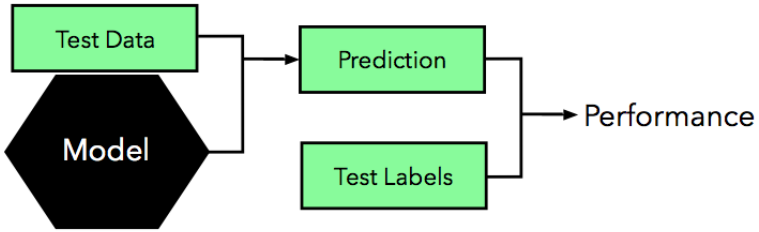
- Training set과 validation set을 합쳐서, 앞서 도출된 best parameter set으로 모델을 재학습

4



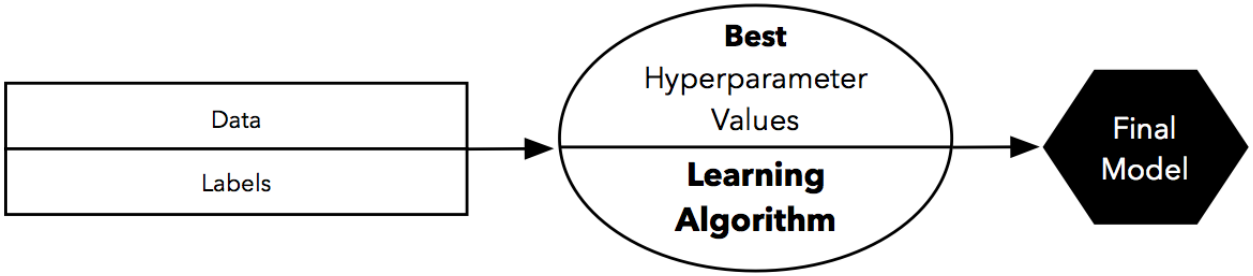
- Test set으로 모델 평가

5



- 최종 모델 학습

6

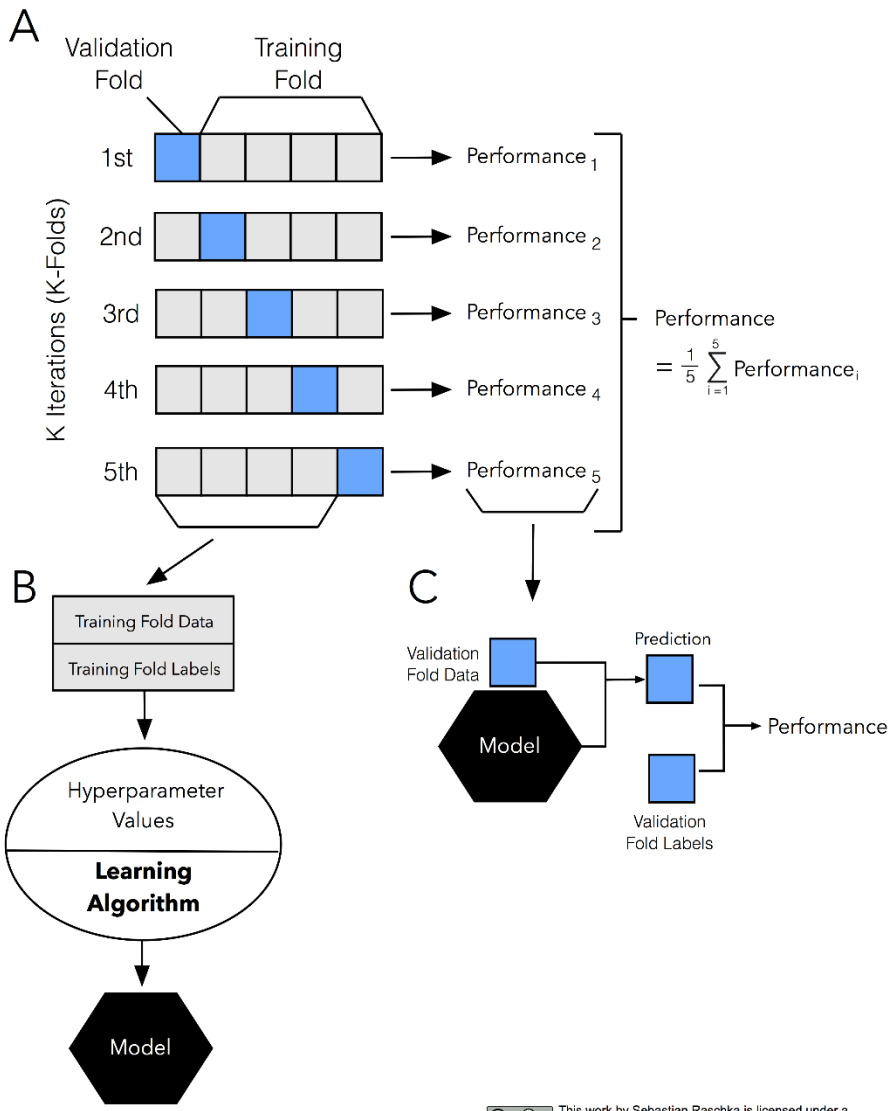


3-way holdout

- Pros
  - 2-way holdout와 비교했을 때, parameter search를 통해 찾은 best parameter set으로 학습한 모델의 일반화 성능 (미래 예측 성능) 을 test set 으로 찾을 수 있음
  - k-fold cv에 비해 모델 생성에 걸리는 시간이 짧음
- Cons
  - Training set의 사이즈가 데이터에 비해 작음
    - Training set의 전체 데이터의 분포, 내재된 정보를 제대로 포함하지 못한 상태라면 내가 만든 모델의 성능이 좋아지기 어려움 ➔ *Pessimistic bias*

k-fold cross-validation

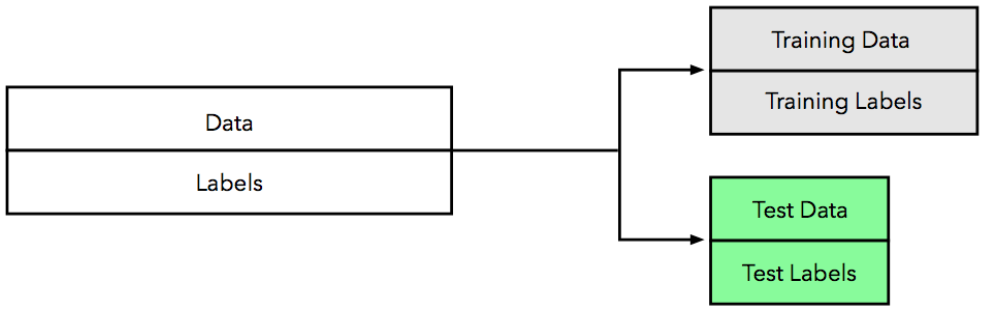
- 데이터를 k개의 겹치지 않는 folds로 분리
- k개의 folds 중 하나를 validation set, 나머지를 training set으로 사용
- 하나의 파라미터 셋에 대해 k번 모델을 생성하여 모델 성능 평가
  - 겹치지 않는 validation set을 이용하므로, 모델 성능의 variance를 측정하기에 매우 용이함.



k-fold cross-validation

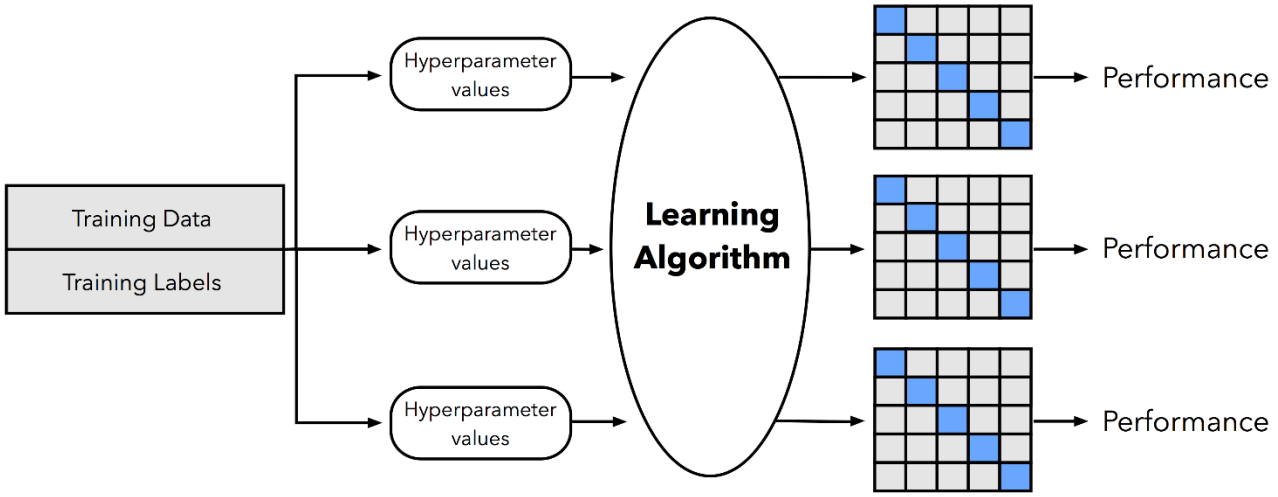
- 데이터를 training set, test set으로 분리

1



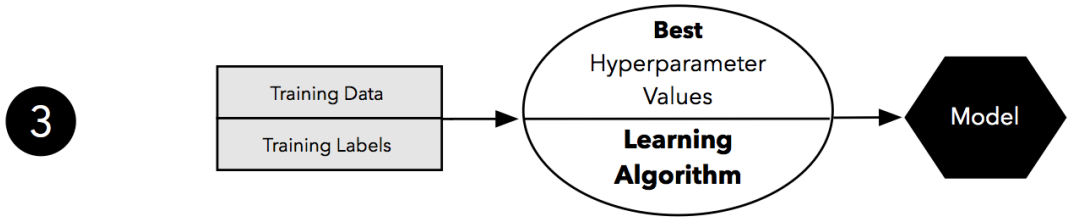
- Training set을 이용하여 여러 개의 parameter sets에 대해 k-fold cross-validation 수행

2

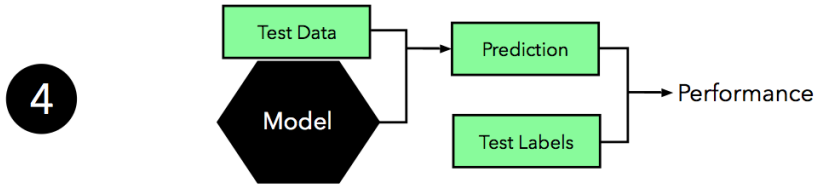


k-fold cross-validation

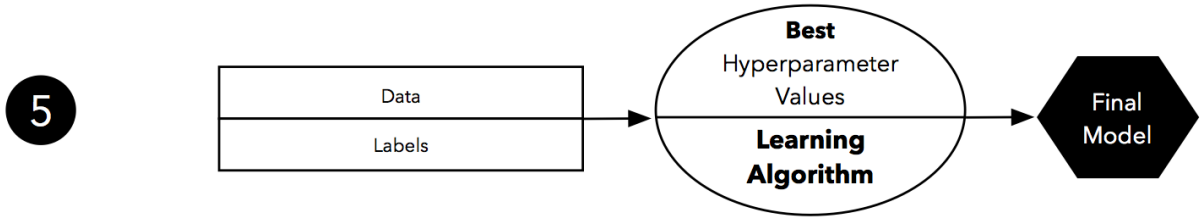
- k-fold cross-validation 결과 가장 좋은 parameter set을 선택하고, 이를 이용해 training set에 모델을 학습



- 생성된 모델을 test set을 이용하여 평가



- 전체 데이터에 최종 모델을 학습



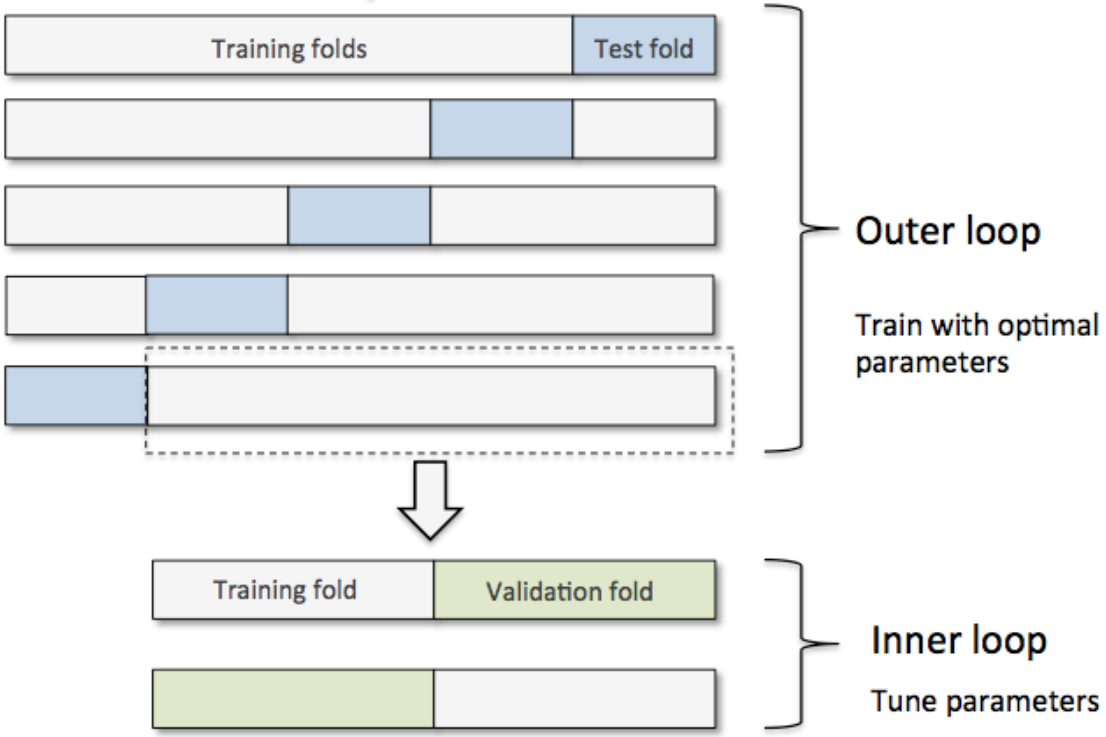
k-fold cross-validation

- Pros
  - 모델의 편향성 (bias) 와 분산성 (variance) 을 측정하기에 용이함
    - 하나의 parameter set으로 k번 모델 생성이 가능함
      - 생성된 k개의 모델의 validation error를 이용하여 모델의 예측 성능이 얼마나 잘 나오는지 (bias) 를 계산할 수 있고,
      - k개 모델로 계산한 validation error의 분포, 표준편차 등을 통해 해당 모델이 학습 데이터의 변동에 따른 성능 변화 (variance) 를 볼 수 있음.

- Cons
  - 여러 번 모델 학습이 이루어지므로 시간이 많이 소요됨

Nested k-fold cross-validation

- 앞에서는 데이터를 training set과 test set으로 나눈 후, training set에 대해 k-fold CV를 수행
- Nested k-fold CV는 실제로 두 번의 CV를 수행
  - Outer CV: Training set과 test set을 여러 번 나눔
  - Inner CV: Outer CV에서 생성된 training set 안에서 k-fold CV를 수행



Data split

- Candidates
  - Training 70% / Test 30%
  - Training 50% / Validation 30% / Test 20%
  - Training 70% / Validation 20% / Test 10%
  - Training 2018.01.01~2018.09.30 / Test 2018.10.01~2018.12.31
  - ...
- **고려해야 할 사항?**
  - 데이터 포인트 수가 충분한가?
  - Train / Validation / Test의 데이터 분포가 동일한가?
    - Independent and identically distributed (i.i.d.) samples