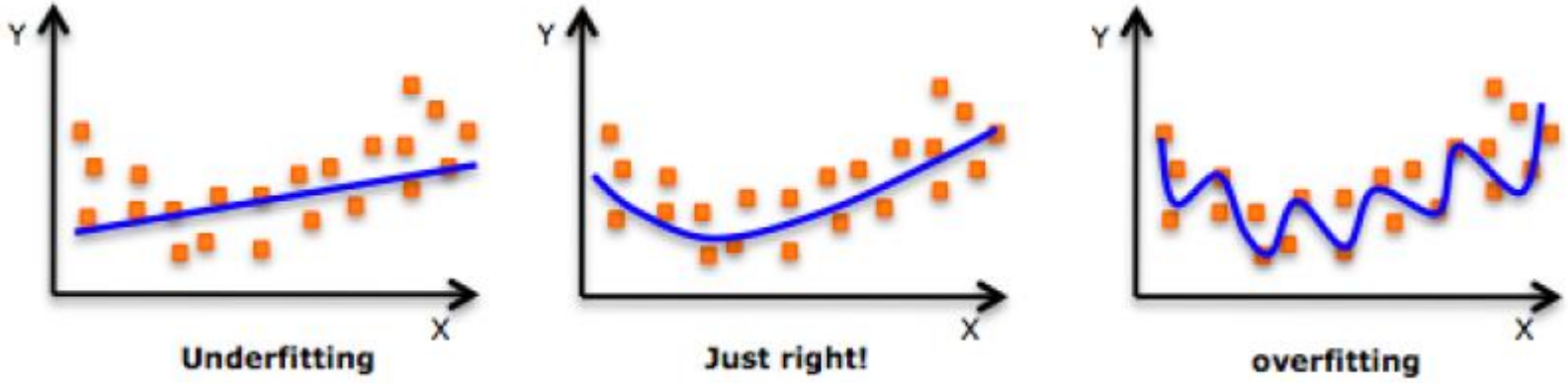
K-fold 교차검증 R로 쉽게 이해하기!

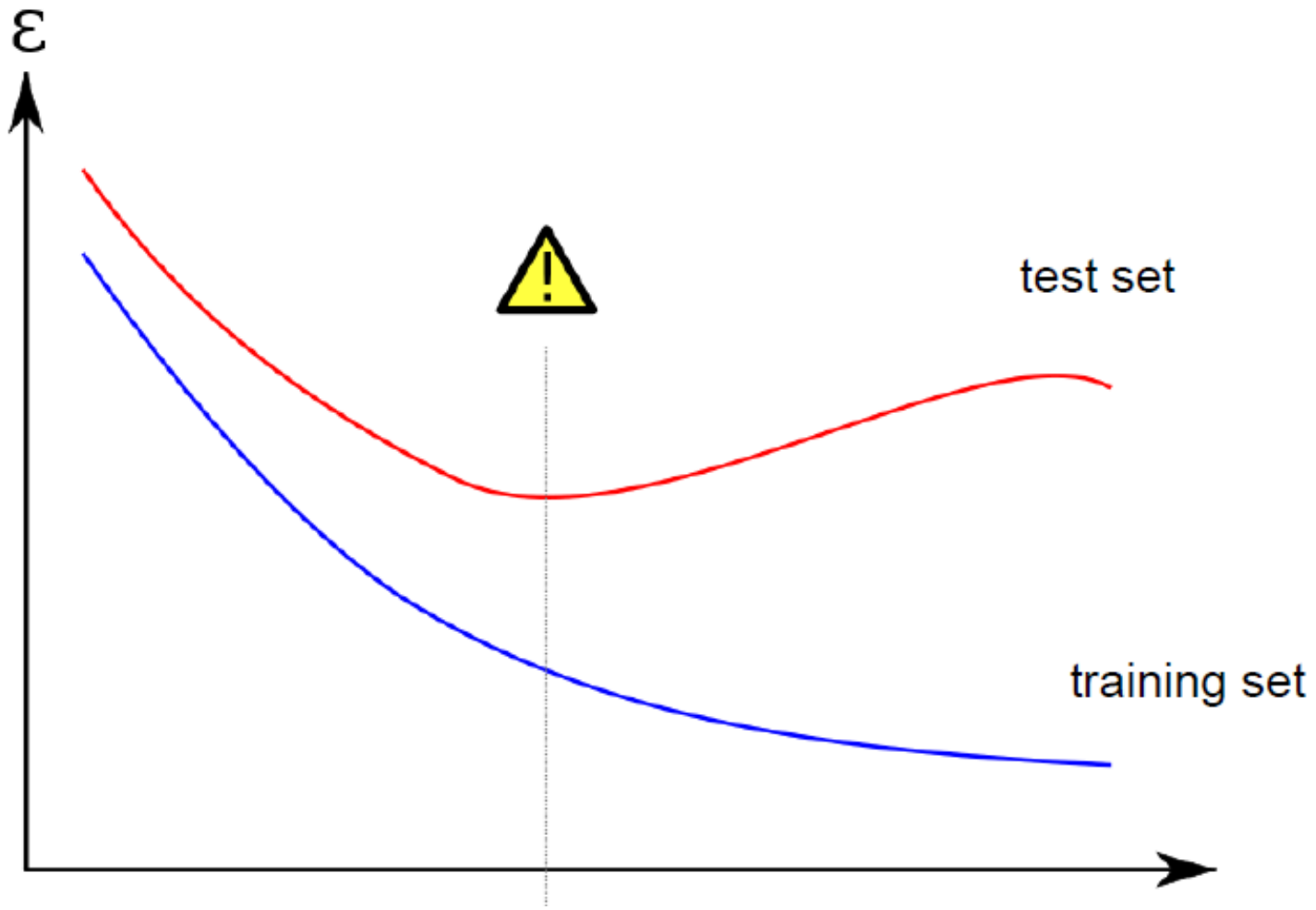
조인식 조교

우리들의 숙적, Overfitting

- Overfitting이랑 문자 그대로 너무 과도하게 데이터에 대해 모델을 learning을 한 경우를 나타낸다.
- 주어진 데이터에 지나치게 꼭 맞는 모델을 생성할 경우 모델은 train데이터는 훌륭한 성능을 낼 수 있을지언정 Test데이터와 미래의 데이터에 대해서는 오히려 성능악화를 초래한다.



(http://sanghyukchun.github.io/59/)


ε
test set
training set

우리들의 숙적, Overfitting

- Variance & Bias : 모델의 에러를 일으키는 요소

- Variance of model: 훈련자료가 바뀌었을 때 모델이 바뀌는 정도
- Bias of model: 개발된 모형과 실제 모형간의 차이
- Bias of data : 데이터의 편향상태, 데이터가 한쪽 레이블에 과도하게 치우쳐진 상태

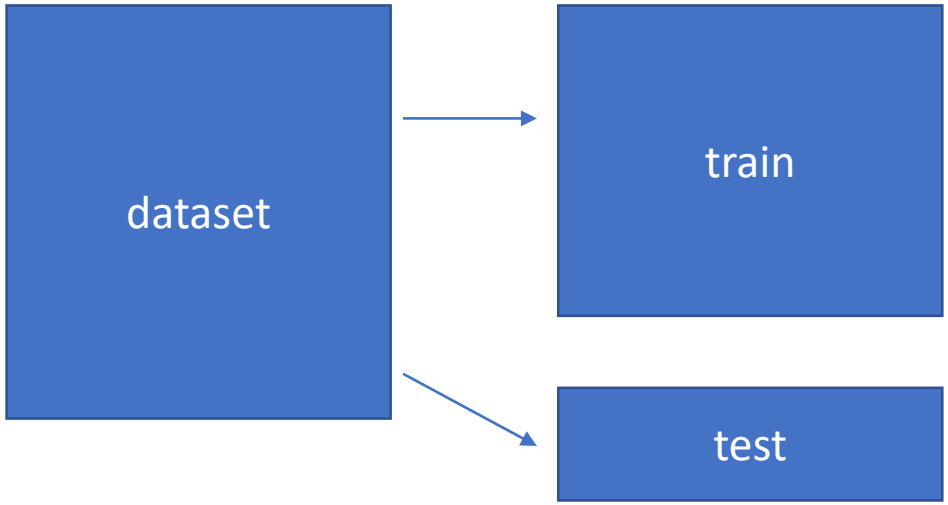
- Bias – Variance Tradeoff
  모형에 대한 가정이 약할수록 bias는 작아지고 variance는 커지는 반면
  모형에 대한 가정이 강할수록 bias는 커지고 variance는 강해진다.

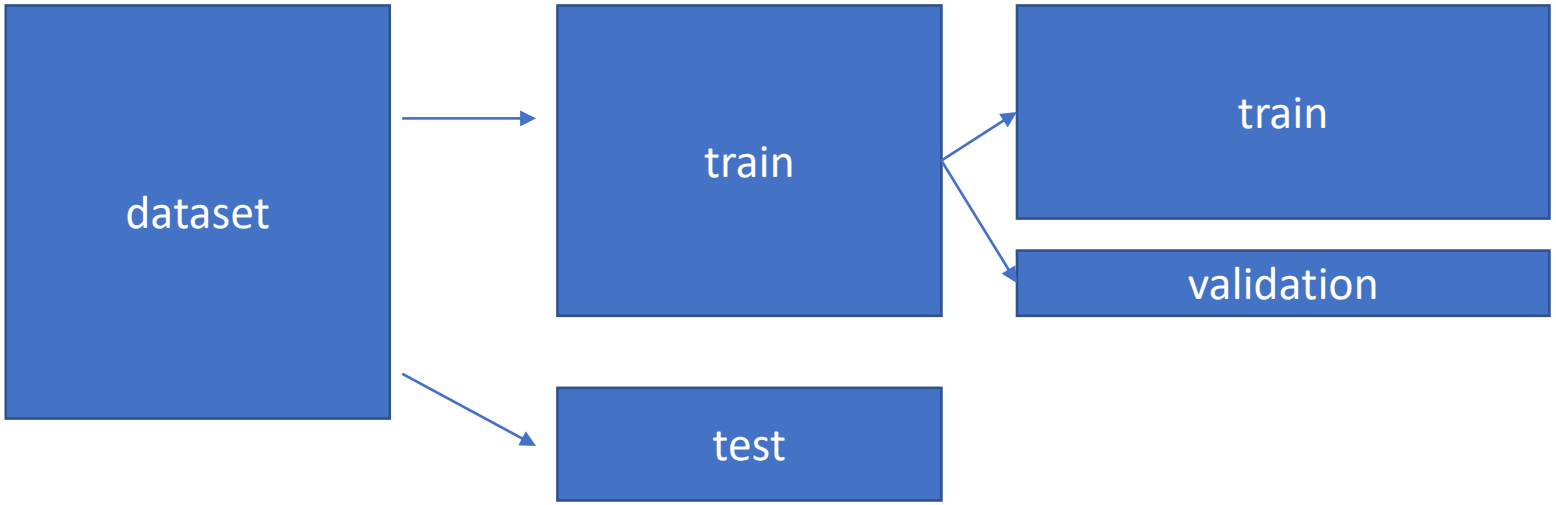
- Variance와 bias를 잘 조절해서 최적의 성능을 가지는 모델을 만드는 것이 machine learning의 과제

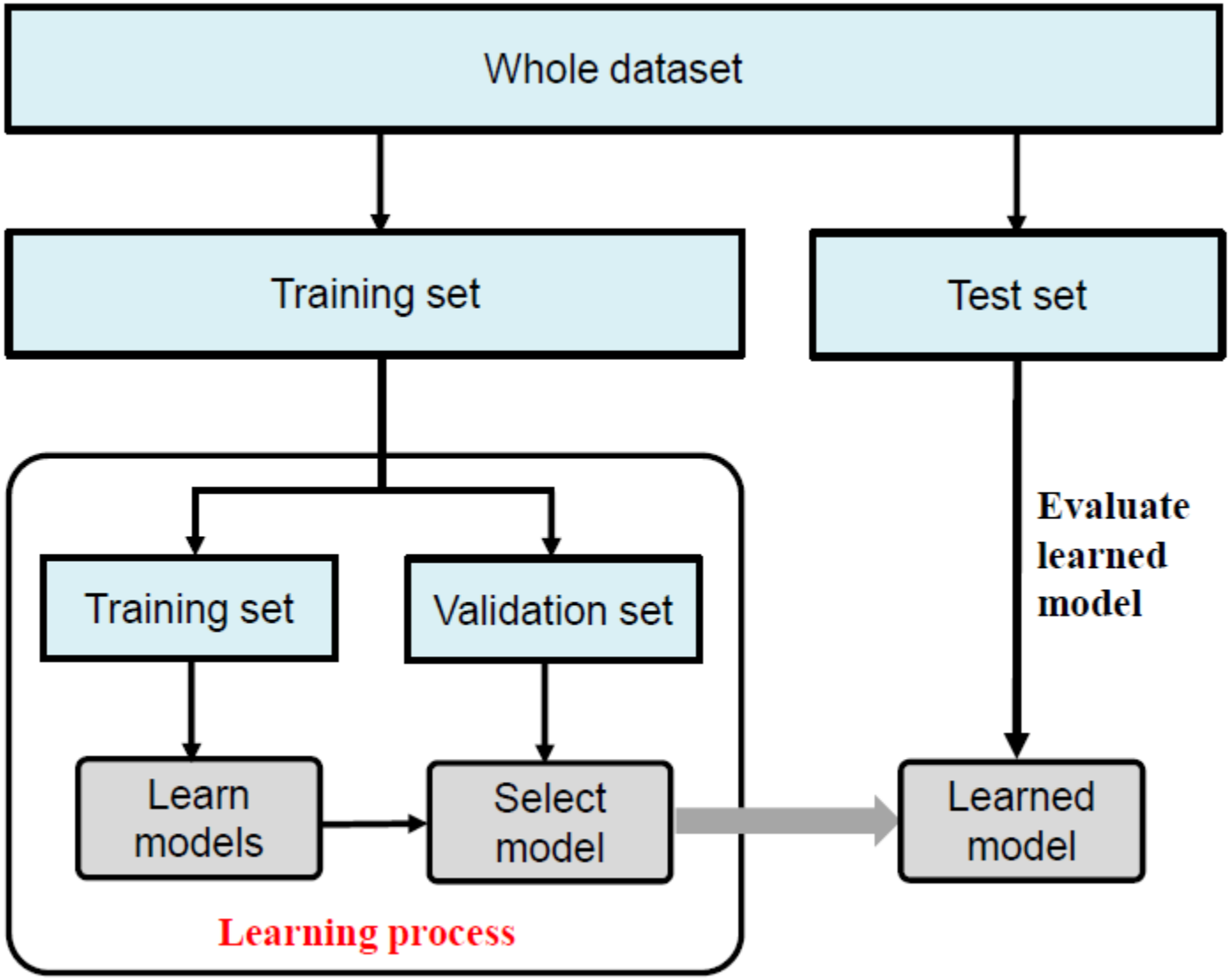
- 이를 조절하기위하여 우리는 검증과정을 거쳐야 한다.

우리가 지금까지 해온 방법… Hold-out 기법

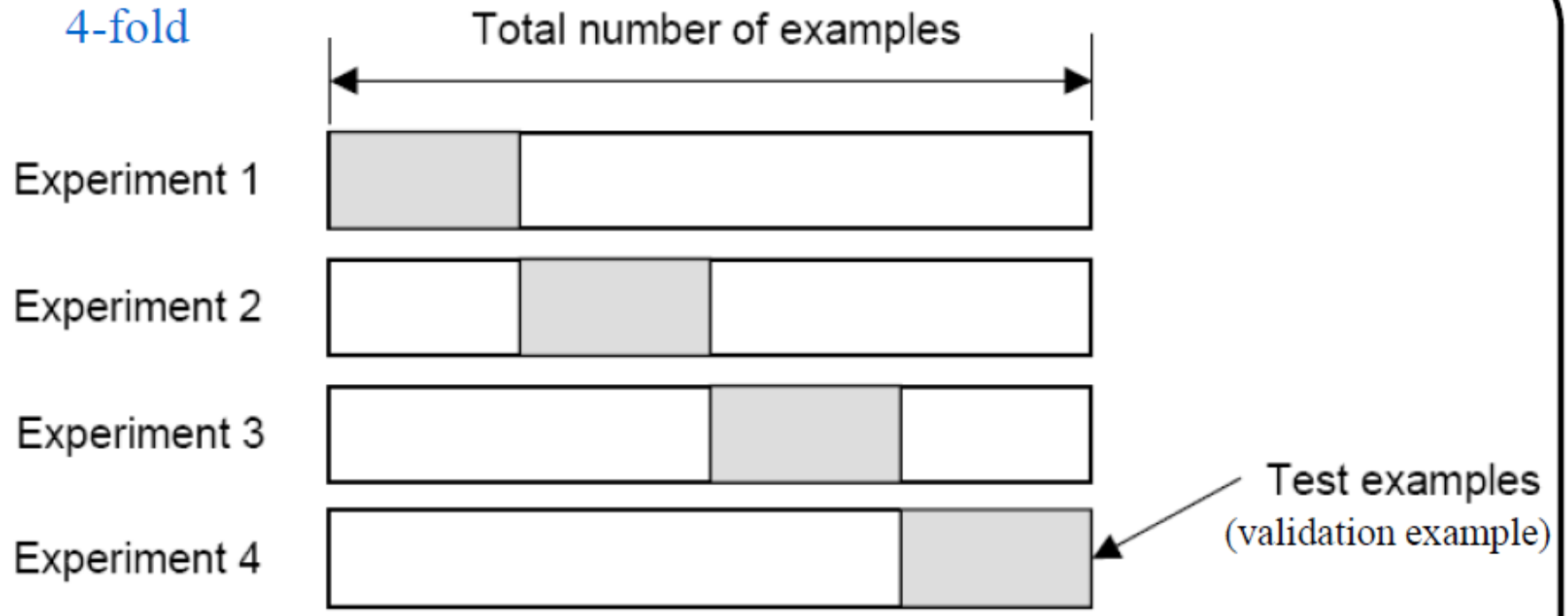
```
Idx   = sample(1:150, 100)
Train = iris[Idx, ]
Test  = iris[-Idx, ]
```




dataset
train
train
validation
test


Whole dataset
Training set
Test set
Training set
Validation set
Learn models
Select model
Learning process
Evaluate learned model
Learned model

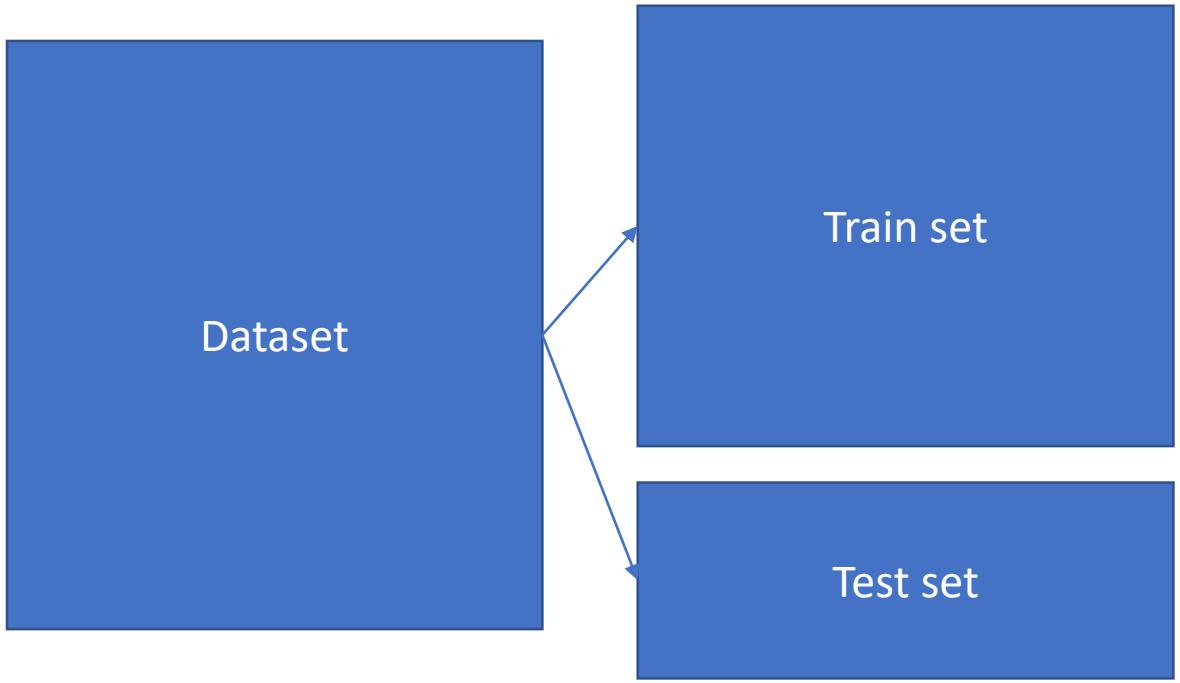
데이터를 k개의 fold로 만들어서 각 fold들이 test/train 역할을
분리해가며 교차검증을 실시



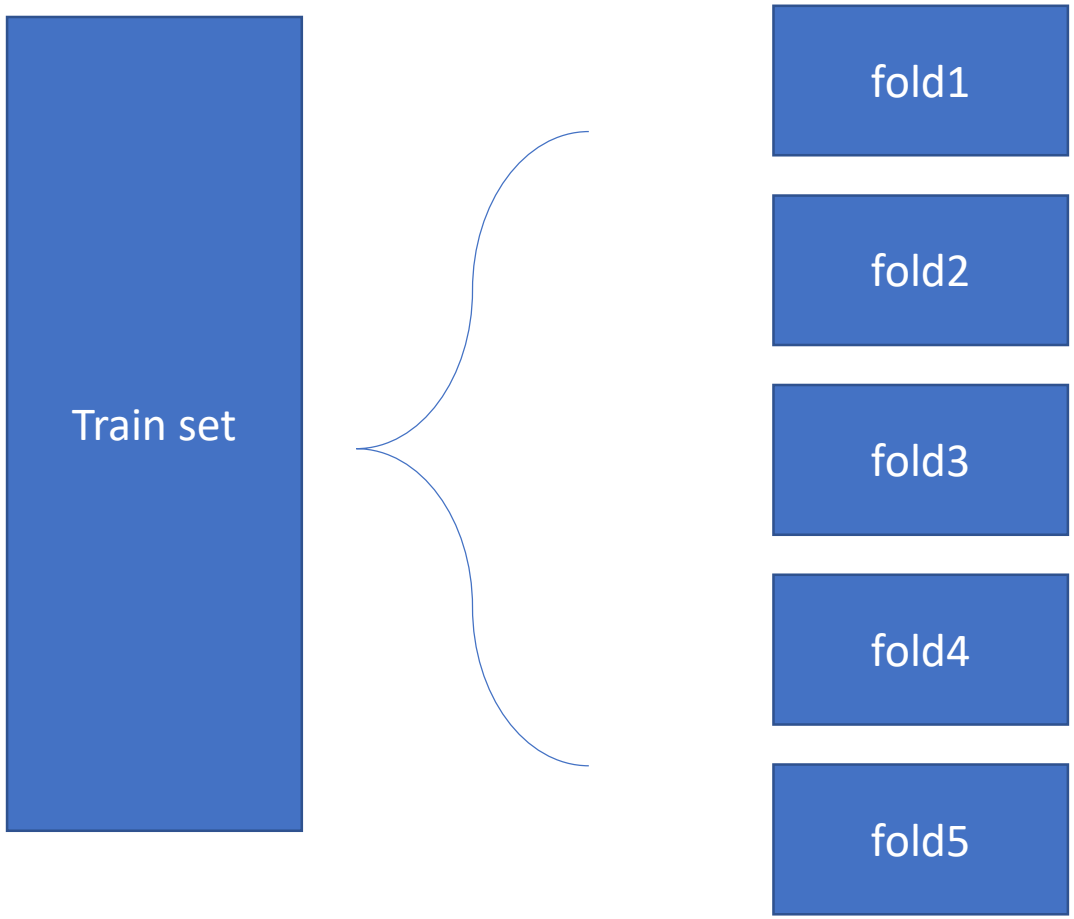
$$\text{Model error } E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Learning process

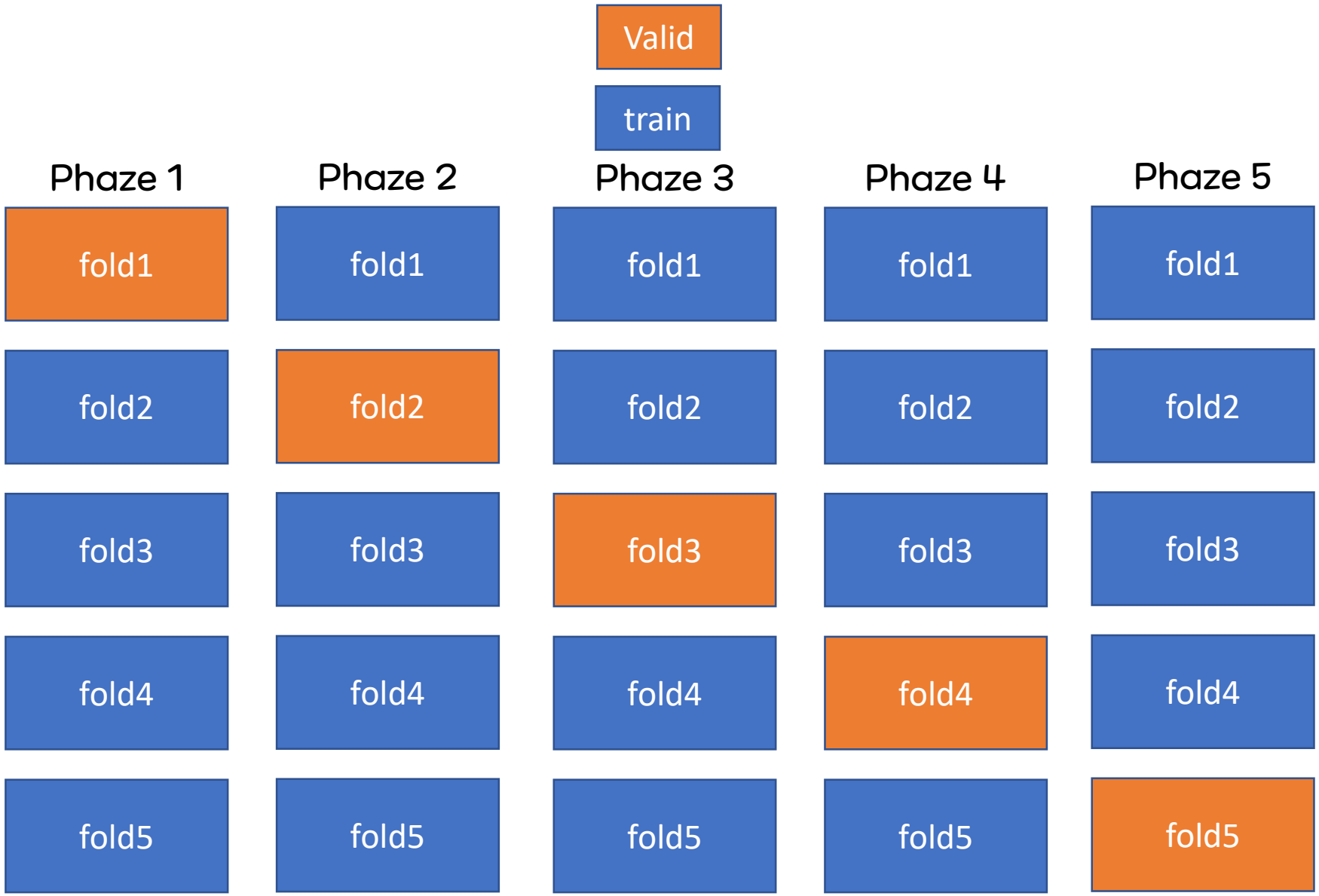
교차검증의 과정



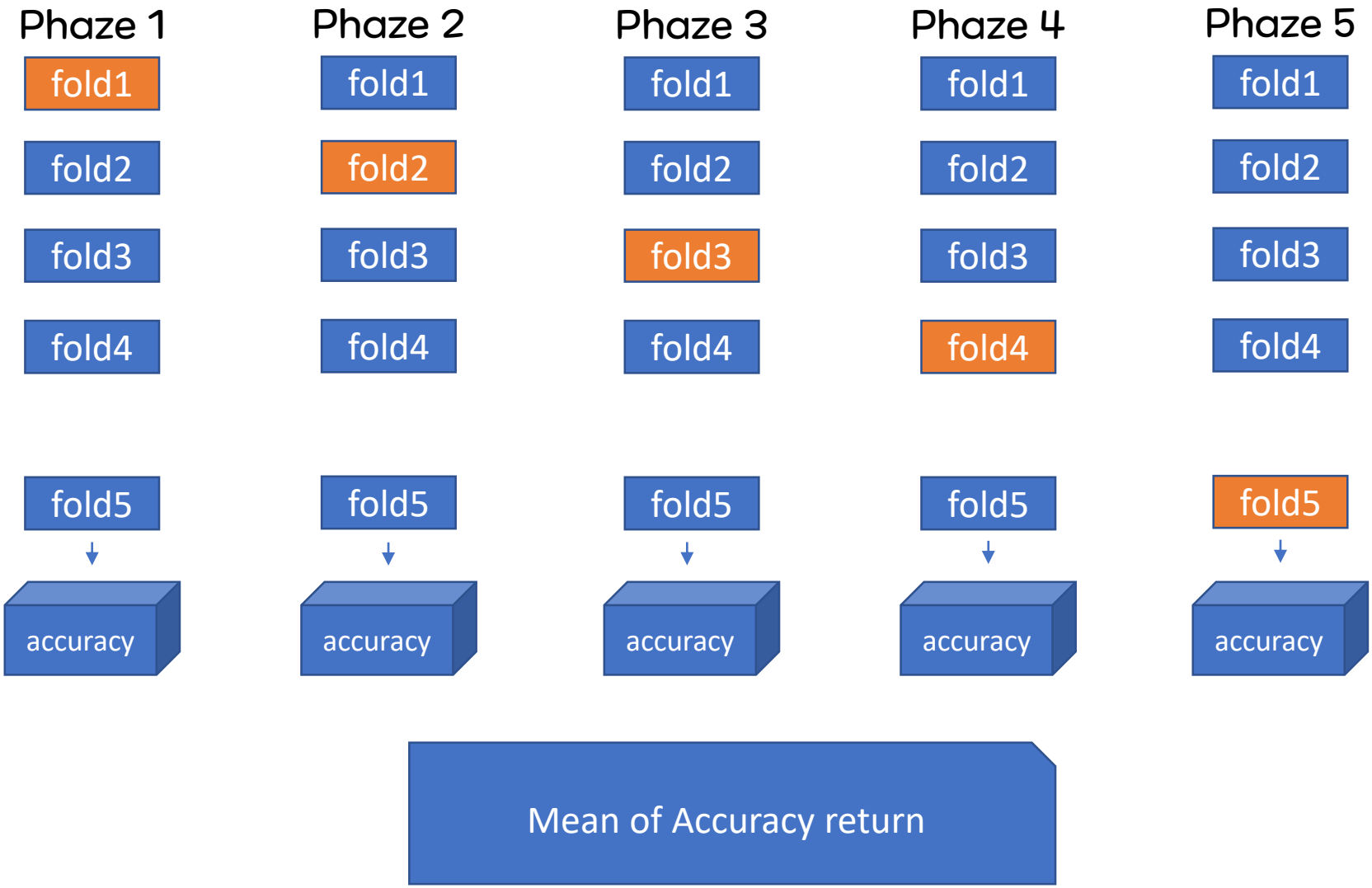
교차검증의 과정



교차검증의 과정


Valid
train
Phaze 1
Phaze 2
Phaze 3
Phaze 4
Phaze 5
fold1
fold1
fold1
fold1
fold1
fold2
fold2
fold2
fold2
fold2
fold3
fold3
fold3
fold3
fold3
fold4
fold4
fold4
fold4
fold4
fold5
fold5
fold5
fold5
fold5

교차검증의 과정


Phaze 1
fold1
fold2
fold3
fold4
fold5
accuracy
Phaze 2
fold1
fold2
fold3
fold4
fold5
accuracy
Phaze 3
fold1
fold2
fold3
fold4
fold5
accuracy
Phaze 4
fold1
fold2
fold3
fold4
fold5
accuracy
Phaze 5
fold1
fold2
fold3
fold4
fold5
accuracy
Mean of Accuracy return

마지막 실습문제

1주에 배웠던 내용과 이번주에 배운 내용, 그리고 K-fold 교차검증을 이용해서 다음 함수를 구축해보자.
K-fold Cross validation을 하는 함수를 구현해본다.
K-fold Cross Validation으로 RandomForest 를 돌리고 이에 대한 결과로 평균 Accuracy를 리턴하는 함수를 만들어보자.

〈함수구조〉

> Kfold.rf(ds = "x인자", cl = "y인자", fold = 10)

```
Kfold.rf = function(ds, cl, fold = 10){
        …(함수 구축)
}
```

〈실행 예〉

```
>Kfold.rf(ds = iris[, -5], cl = iris[, 5], fold = 10)
[1]0.9533333
```

힌트 : fold를 나누는 함수 = caret 패키지의 createFolds(벡터, 폴드수)