과적합

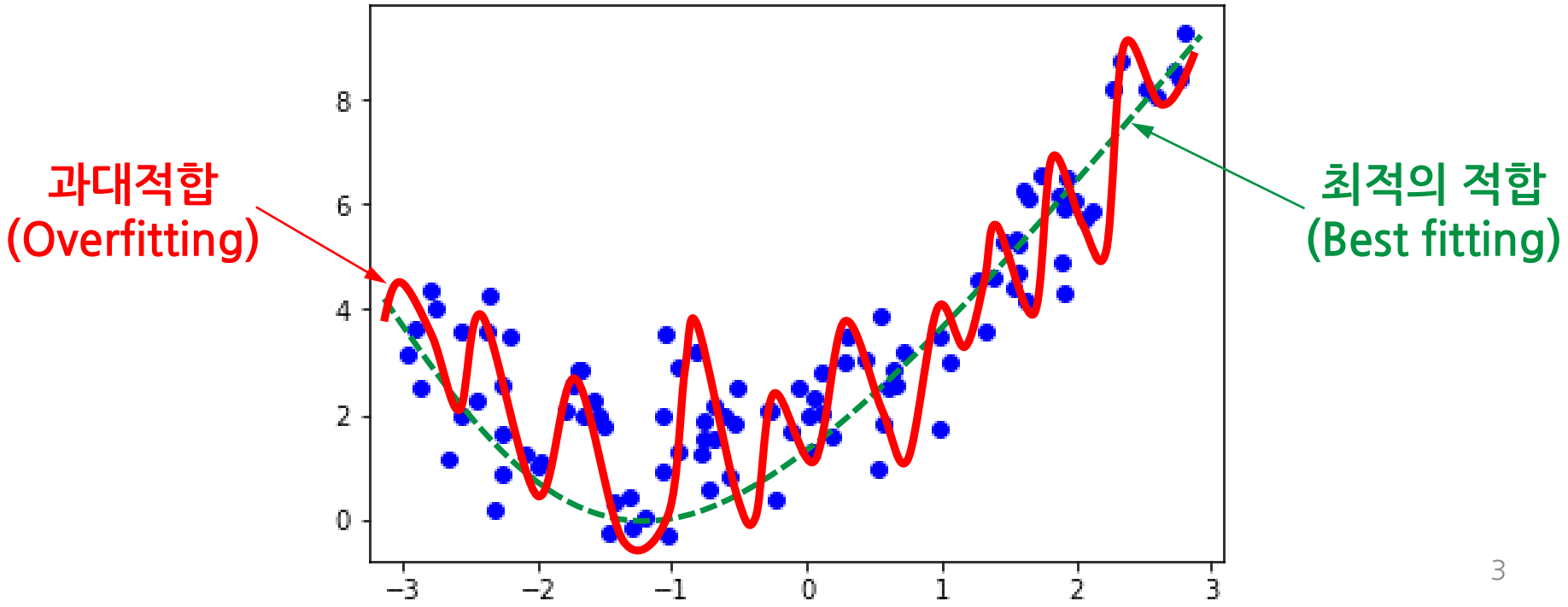
과적합

- 훈련 모형의 적합성
  - 훈련 데이터를 이용하여 학습을 수행할 때, 모형은 당연히 그 훈련 데이터에 대해서는 적합한 결과를 도출할 것이다.
  - 이 모형이 검증 데이터에 대해서도 정확하게 예측해야만 일반화(generalization)가 잘 되었다고 판단할 수 있다.

  - 또는, 훈련 데이터에 대해서 적합한 결과를 도출할 수 있는 정도의 학습이 이루어지지 않을 수도 있다.
  - 설령 검증 데이터에 대해서 정확하게 예측한다고 하더라도 애초에 그 모형의 신뢰성은 높지 않을 것이다.

과적합

- 과대적합 (Overfitting)
  - 훈련 데이터를 너무 지나치게 학습하여 학습 모형이 일반화 되지 않은 상태
  - 훈련 데이터에 대해서만 너무 잘 맞춰져 있기 때문에, 학습 되지 않은 다른 데이터에 대해서는 잘 대응되지 않는다.



과적합

- 과대적합

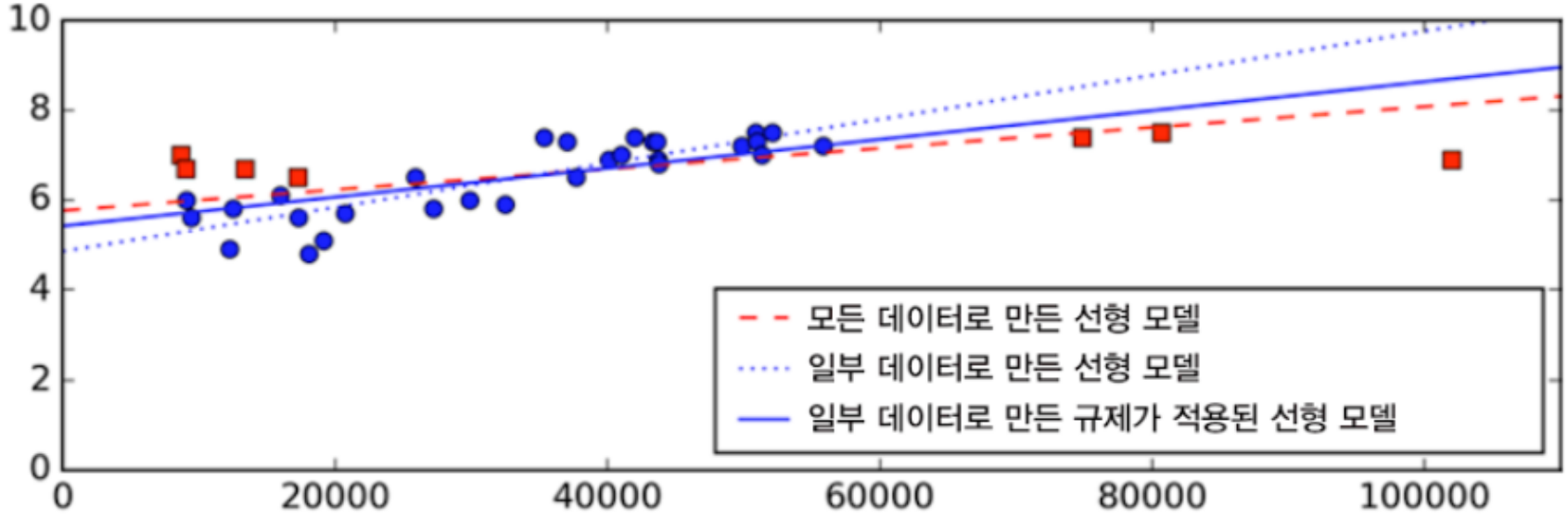
과대적합의 원인	해결 방안
훈련 데이터의 특성에 비해 학습 모형이 너무 복잡한 경우	매개변수의 수가 적은 모형을 선택하거나 모형에 규제(regularization)를 적용하여 단순화시킨다.
훈련 데이터의 특성이 특정 값으로 편중된 경우	훈련 데이터의 특성 값을 다양화한다.
훈련 데이터의 개수가 너무 적은 경우	더 많은 훈련 데이터를 확보한다.
훈련 데이터에 노이즈가 많은 경우	오류 데이터를 수정/삭제하거나 이상치(outlier)를 제거한다.

과적합

- 과대적합에 대한 규제 (Regularization)
  - 과대적합을 피하기 위해서 학습 모형을 단순화하고 제약을 가하는 것을 규제(regularization)라고 한다.
  - 학습하는 동안 적용하는 규제의 수준은 하이퍼파라미터(hyperparameter)를 기반으로 결정되며, 학습을 수행하기 전에 미리 지정되어 훈련하는 동안에는 상수로 동작한다.
  - 하이퍼파라미터의 값을 크게 지정할수록 복잡도가 낮은 모형을 추정하게 된다.
    - 과대적합의 위험성을 감소시킬 수 있다.
    - 반면, 결과 모형의 설명력도 낮아질 수 있다.

과적합

- 과대적합에 대한 규제

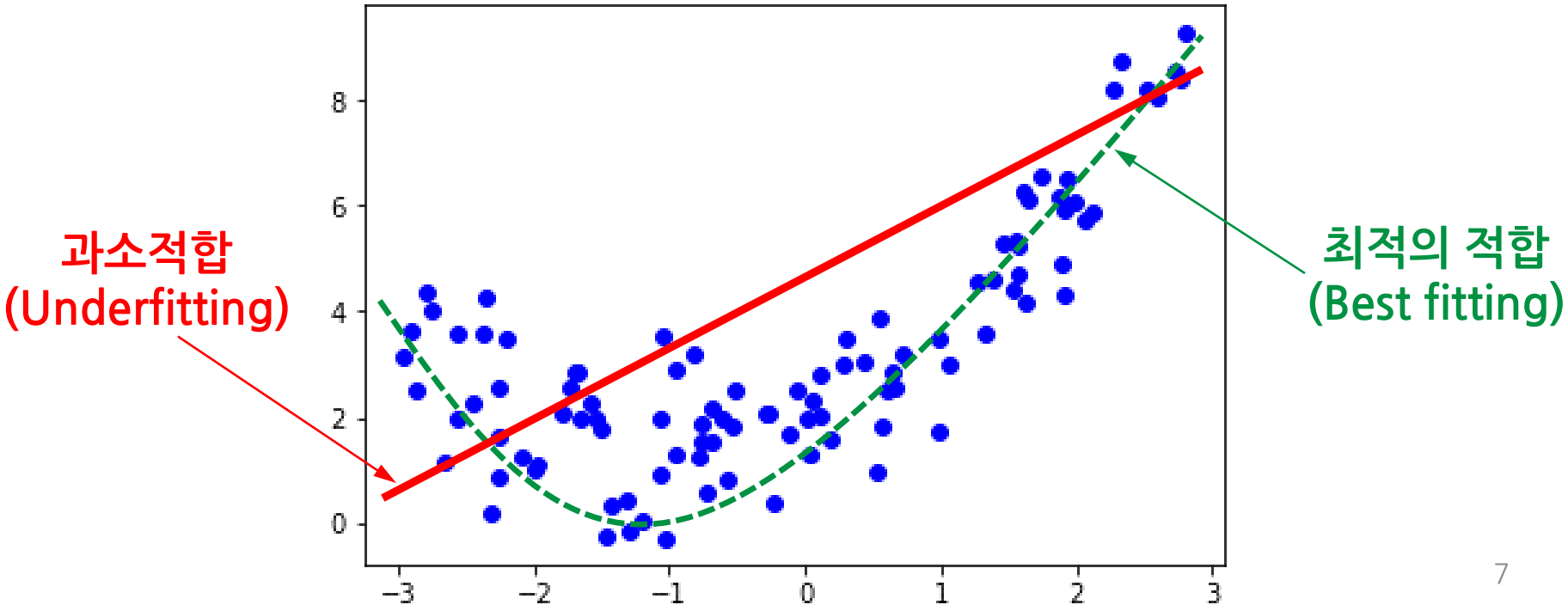


  - 훈련 데이터의 수를 늘릴수록 더 잘 일반화된다.
  - 규제를 적용하면 더 잘 일반화된다.

※ 그림 출처: 핸즈온 머신러닝, p61, 한빛미디어

과적합

- 과소적합 (Underfitting)
  - 훈련 모형이 너무 단순하여 훈련 데이터의 특징을 잘 학습하지 못한 상태
  - 훈련 데이터조차 제대로 반영하지 못 하여 설명력이 낮기 때문에, 다른 데이터의 예측에 대한 신뢰성이 없다.



과적합

- 과소적합

과소적합의 원인	해결 방안
훈련 데이터의 특성에 비해 학습 모형이 너무 단순한 경우	매개변수의 수가 더 많은 복잡한 모형을 선택하거나 새로운 특성을 추가한다.
학습 모형에 지나치게 규제가 많이 적용된 경우	모형에 적용되어 있는 제약을 줄인다.
훈련 데이터 자체를 충분히 학습시키지 않은 경우	과대적합이 되기 전까지 충분히 오랫동안 (또는 반복하여) 학습시킨다.

편향-분산 트레이드오프

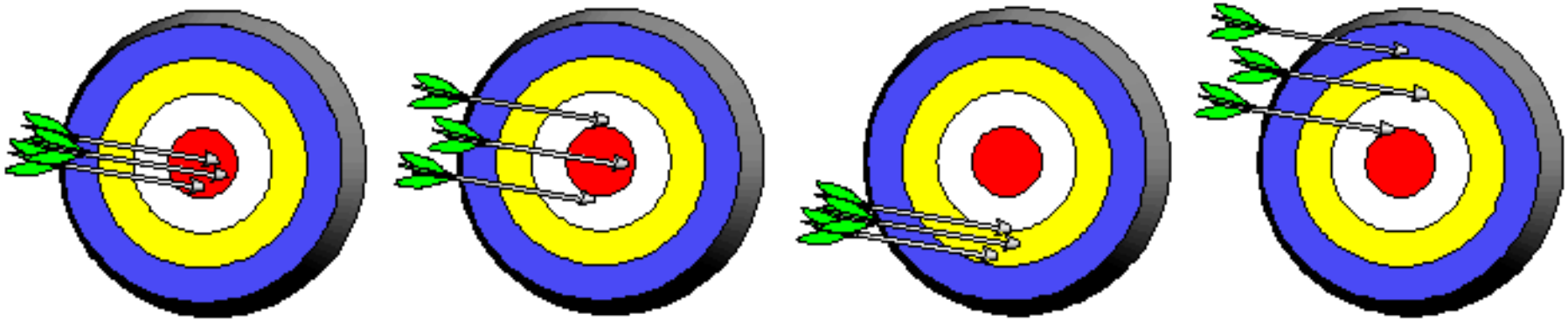
편향-분산 트레이드오프

- 편향 (Bias)
  - 데이터 또는 학습 모형이 얼마나 특정한 방향으로 치우쳐져 있는가를 의미한다.
  - 학습 모형에서 예측값이 정답과 멀리 있는 쪽으로 치우친 경향이 있을 때 고편향(high bias)되어 있다고 표현한다.

- 분산 (Variance)
  - 데이터 또는 학습 모형이 얼마나 넓은 범위에 걸쳐 분포되어 있는가를 의미한다.
  - 학습 모형에서 예측값이 서로 넓게 흩어져 있어서 변동성이 높을 때 고분산(high variance)되어 있다고 표현한다.

편향-분산 트레이드오프

- 편향과 분산의 표현



low bias
low variance

최적의 적합
(Best fitting)

low bias
high variance

과대적합
(Overfitting)

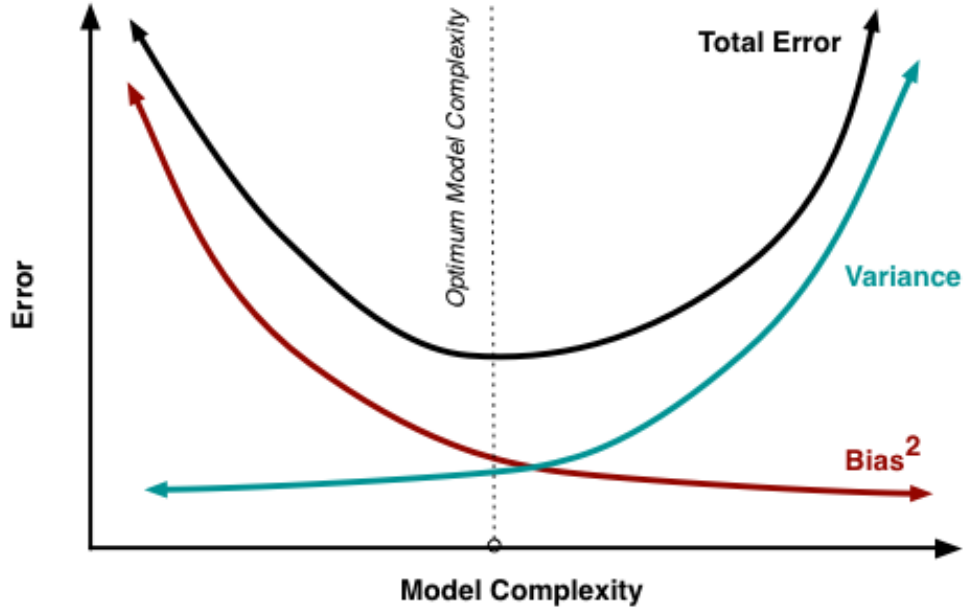
high bias
low variance

과소적합
(Underfitting)

high bias
high variance

편향-분산 트레이드오프

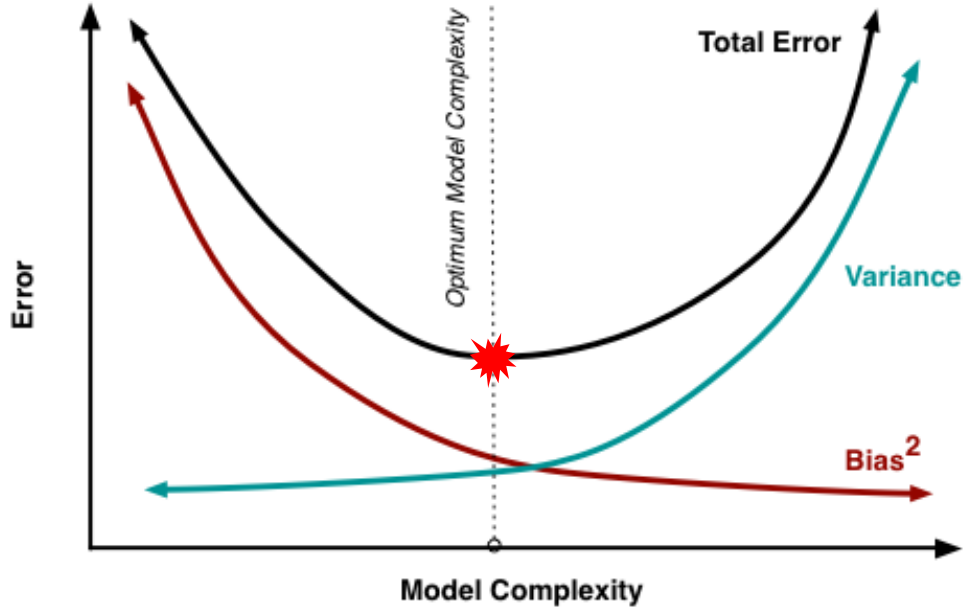
- 편향-분산 트레이드오프 (Bias-Variance Trade-off)
  - 일반적으로 편향과 분산 간에는 한 쪽이 높으면 나머지 한 쪽이 낮아지는 경향이 있다.
  - 즉, 편향이 높으면 분산이 낮아진다. (과소적합)
  - 반대로, 분산이 높으면 편향이 낮아진다. (과대적합)



※ 그림 출처: http://scott.fortmann-roe.com/docs/BiasVariance.html

편향-분산 트레이드오프

- 편향-분산 트레이드오프 (Bias-Variance Trade-off)
  - 편향과 분산이 둘 다 동시에 최소화 되는 것은 일반적으로 불가능하며, 이것을 편향-분산 트레이드오프라고 한다.
  - 학습을 통해서 전체 오류가 최소화 되는 지점을 찾으면 그 시점에서의 추정된 모형이 가장 최적의 모형이 된다.



※ 그림 출처: http://scott.fortmann-roe.com/docs/BiasVariance.html

교차 검증

교차 검증

- 교차 검증 (Cross Validation)
  - 학습 모형의 성능을 확인하기 위해서는 검증 데이터가 필요하므로, 전체 원본 데이터 중 일부를 골라서 검증 데이터로 사용하게 된다.
  - 이 검증 데이터를 어떻게 선택하는가에 따라서 학습 모형의 성능이 달라질 수 있다.
  - 따라서 검증 데이터의 편중을 막기 위해서 검증 데이터를 1종만 사용하지 않고 각각 다른 여러 종류의 검증 데이터를 선택하여 검증을 여러 차례 실시한다.
  - 이러한 과정을 교차 검증(cross validation)이라고 한다.

교차 검증

- K 폴드 (K-fold) 교차 검증
  - 데이터 집합을 K개의 부분집합(폴드)로 분리하여, (K-1)개의 폴드를 이용하여 학습을 수행하고 나머지 1개의 폴드를 이용하여 검증을 수행하는 과정을 K회 반복하는 검증 방식

교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (1)
  ① model_selection 모듈에 있는 KFold를 이용하여 폴드를 분리할 객체를 생성한다.
  - 첫 번째 매개변수 또는 n_splits는 분리할 폴드의 개수이다. 기본값은 3이다.
  - 매개변수 shuffle은 데이터를 섞어서 분리할 것인지의 여부를 결정한다. 기본값은 False이다.
  - 매개변수 random_state는 분리할 때마다 동일한 집합으로 생성할 것인지 결정하는 정수 값이다.

```
1  import sklearn.model_selection as ms
2
3  kfold = ms.KFold(5)
```

교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (1)

  ② 데이터를 준비하고 회귀 모형 객체를 생성한다.

```
1  import sklearn.datasets as d
2  import sklearn.linear_model as lm
3
4  diab = d.load_diabetes()
5  X = diab.data
6  y = diab.target
7
8  lr = lm.LinearRegression()
```

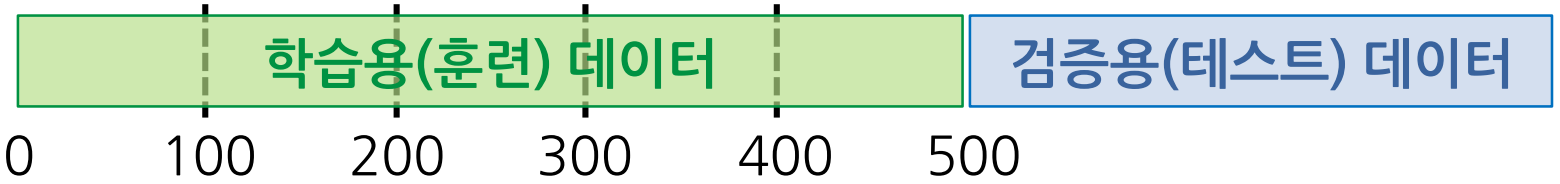
교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (1)

  ③ KFold 객체의 **split** 함수를 호출하면 폴드 별 훈련 데이터 및 검증 데이터의 레코드 번호(즉, 행의 인덱스)들을 배열 형태로 반환한다. 이를 이용하여 폴드 별로 학습을 수행한다.

  - 매개변수는 분리 대상인 전체 데이터이다.

K=5인 경우



  - 위의 경우, KFold 객체의 split 함수 결과는 다음과 같다.

```
[0 1 ... 399]  [400 401 ... 499]
[0 1 ... 299 400 401 ... 499]  [300 301 ... 399]
[0 1 ... 199 300 301 ... 499]  [200 201 ... 299]
[0 1 ... 99 200 201 ... 499]  [100 101 ... 199]
[100 101 ... 499]  [0 1 ... 99]
```

교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (1)

  ③ KFold 객체의 **split** 함수를 호출하면 폴드 별 훈련 데이터 및 검증 데이터의 레코드 번호(즉, 행의 인덱스)들을 배열 형태로 반환한다. 이를 이용하여 폴드 별로 학습을 수행한다.

```
import sklearn.metrics as mt

n_iter = 0
r2_scores = []

for train_num, test_num in kfold.split(X):
    X_train, X_test = X[train_num], X[test_num]
    y_train, y_test = y[train_num], y[test_num]

    reg = lr.fit(X_train, y_train)
    y_pred = reg.predict(X_test)

    n_iter += 1
    r2_scores.append(mt.r2_score(y_test, y_pred))
```

교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (1)
  ④ 개별 회차별 결정 계수와 전체 평균 결정 계수 값을 출력하여 검증 결과를 확인한다.

```
1  import numpy as np
2
3  for i in range(n_iter):
4      print("[{}회차] R2: {:.3f}".format(i+1, r2_scores[i]))
5  print("평균 R2:", np.round(np.mean(r2_scores), 3))
```

```
[1회차] R2: 0.430
[2회차] R2: 0.523
[3회차] R2: 0.483
[4회차] R2: 0.427
[5회차] R2: 0.550
평균 R2: 0.482
```

교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (2)
  - model_selection 모듈에 있는 cross_val_score 함수를 이용하여 K 폴드 교차 검증을 수행한다.
    - 첫 번째 매개변수는 학습을 수행하는 객체이다.
    - 두 번째 매개변수는 데이터의 특성(독립변수) 집합이다.
    - 세 번째 매개변수는 데이터의 레이블(종속변수) 집합이다.
    - 매개변수 scoring은 성능 평가 지표 값을 적용할 함수 이름이다. 기본적으로는 명시할 필요 없다.
    - 매개변수 cv는 교차 검증을 수행할 폴드의 개수 또는 KFold 객체이다. 기본값은 폴드의 개수 3이다.
    - 반환 결과는 성능 평가 지표 값들이 들어 있는 배열이다.

교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (2)
  - model_selection 모듈에 있는 cross_val_score 함수를 이용하여 K 폴드 교차 검증을 수행한다.

```
1  import sklearn.datasets as d
2  import sklearn.linear_model as lm
3  import sklearn.model_selection as ms
4  import numpy as np
5
6  diab = d.load_diabetes()
7  X = diab.data
8  y = diab.target
9  lr = lm.LinearRegression()
10
11 r2_scores = ms.cross_val_score(lr, X, y, cv=5)
12
13 print("교차 검증 회차별 R2:", np.round(r2_scores, 3))
14 print("평균 R2:", np.round(np.mean(r2_scores), 3))
```

```
교차 검증 회차별 R2: [0.43  0.523 0.483 0.427 0.55 ]
평균 R2: 0.482
```

교차 검증

- 사이킷런에서 K 폴드 교차 검증 수행 (2)
  - 폴드 분리를 어떻게 하는가에 따라 검증 결과가 달라지고, 이에 따라 성능 지표 역시 변화하는 것을 확인할 수 있다.

```
1  import sklearn.datasets as d
2  import sklearn.linear_model as lm
3  import sklearn.model_selection as ms
4  import numpy as np
5
6  diab = d.load_diabetes()
7  lr = lm.LinearRegression()
8
9  kfold = ms.KFold(3, shuffle=True, random_state=0)
10 r2_scores = ms.cross_val_score(lr, diab.data, diab.target, cv=kfold)
11
12 print("교차 검증 회차별 R2:", np.round(r2_scores, 3))
13 print("평균 R2:", np.round(np.mean(r2_scores), 3))
```

```
교차 검증 회차별 R2: [0.404 0.521 0.544]
평균 R2: 0.49
```

참고 : 매개변수 scoring

매개변수 scoring

- 매개변수 scoring
  - 함수 cross_val_score의 매개변수 scoring은 기본값으로 각 추정자 객체에 대해서 대표적인 성능 지표 함수가 자동적으로 호출된다.
    - 예를 들어, LinearRegression의 경우 scoring의 기본값은 ‘r2’이며 metrics.r2_score를 호출한다.
  - 이 함수를 다른 것으로 적용하고자 하는 경우에는 원하는 함수명을 직접 명시해 주면 된다.

매개변수 scoring

- 매개변수 scoring
  - 전체 scoring 이름과 대응하는 함수의 목록들은 사이킷런 웹사이트의 [Documentation]-[User Guide]-[3.3. Model evaluation: quantifying the quality of predictions] 의 [3.3.1 The scoring parameter] 에서 확인할 수 있다.

    ※ 링크 : https://scikit-learn.org/stable/modules/model_evaluation.html#the-scoring-parameter-defining-model-evaluation-rules