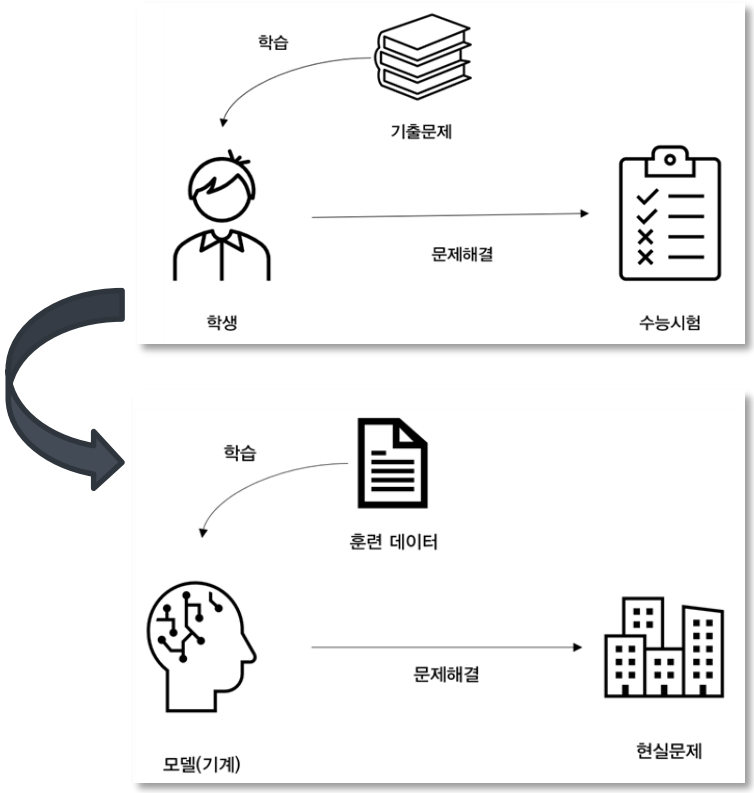
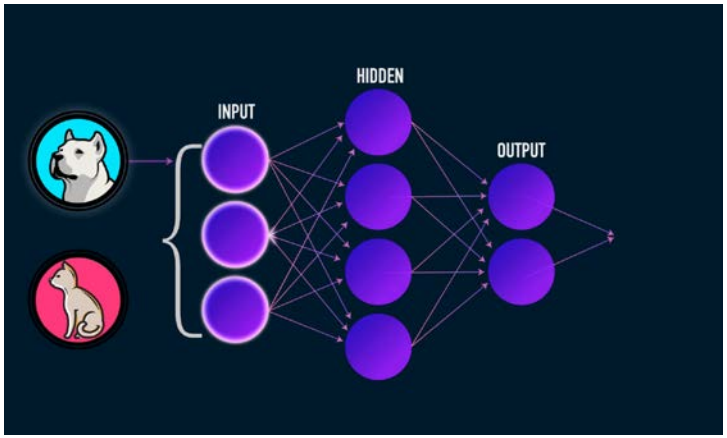
모델의 평가 및 선택
(Test and Model Selection)

한국공학대학교 전자공학부
채승호 교수

기계학습의 목표와 모델

- 머신러닝의 목표
  - 입력에 따른 출력이 실측값과 동일한 결과를 만드는 전달함수(모델) 도출
  - 특정 데이터로 학습하여 일반적인 문제에 적용하는 것
    - 모델의 **일반화 성능**을 극대화하는 것





훈련 오차와 일반화 오차

- 훈련 오차 (Training error)
  - 경험 오차 (Empirical error)와 같은 의미
  - 훈련 데이터에서 발생하는 오차

- 일반화 오차 (Generalization error)
  - 훈련하지 않은 새로운 데이터에서 발생하는 오차
  - 최종 목표는 일반화 오차를 최소화하는 것
    - 일반화 성능을 최대화하는 것과 같은 의미

머신러닝 모델에서의 정확도 문제

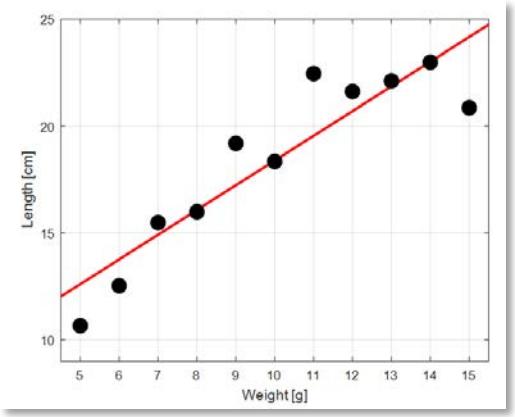
- 훈련 오차 vs 일반화 오차
  - 일반화 오차를 줄이는 것이 궁극적 목표임
  - 그러나, 훈련 과정에서는 훈련 데이터만 경험할 수 있음
    - 실전 데이터를 경험할 수 없음
  - 훈련 데이터의 학습을 통해 일반화 성능을 최대화해야 함
    - 머신러닝의 근본적 문제의 원인
  - 두 가지 중 하나의 문제에 빠질 위험이 있음
    - 학습 능력이 부족함 → 과소적합의 문제 (underfitting)
    - 학습이 과함 → 과적합의 문제 (overfitting)

머신러닝 모델에서의 정확도 문제

- 머신러닝의 문제
  - 일반화 오차를 줄이는 것이 궁극적 목표임
  - 그러나, 훈련 과정에서는 훈련 데이터만 경험할 수 있음
    - 실전 데이터를 경험할 수 없음
  - 훈련 데이터의 학습을 통해 일반화 성능을 최대화해야 함
    - 머신러닝의 근본적 문제의 원인
  - 두 가지 중 하나의 문제에 빠질 위험이 있음
    - 학습 능력이 부족함 → 과소적합의 문제 (underfitting)
    - 학습이 과함 → 과적합의 문제 (overfitting)

Underfitting(과소적합)

- 학습능력이 부족한 경우
  - 추의 무게와 용수철의 관계
    - 최적 매개변수를 찾았지만, 모델 자체의 학습 능력이 부족함



– 이 모델에 따르면, 추의 무게가 무한하면 길이도 무한해짐

  - 이미지 분류 문제

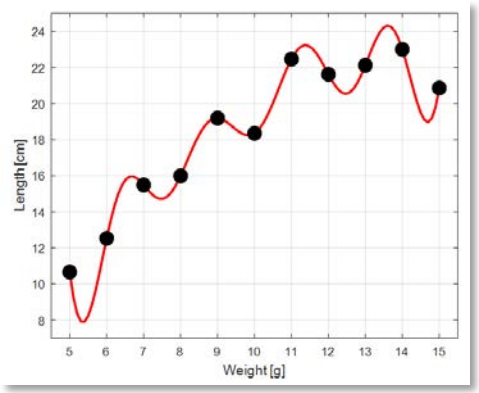
훈련 데이터

새로운 데이터

학습기 예측 결과
→ 이것은 나뭇잎이다.
이유
→ 초록색 둥근 모양은 나뭇잎이다.

Overfitting(과대적합)

- 학습이 과도한 경우
  - 추의 무게와 용수철의 관계
    - 훈련 데이터를 완벽하게 학습한 모델



– 이 모델은 훈련 데이터만 정확하게 맞춤
– 훈련 오차는 최소이나, 일반화 오차는 커짐

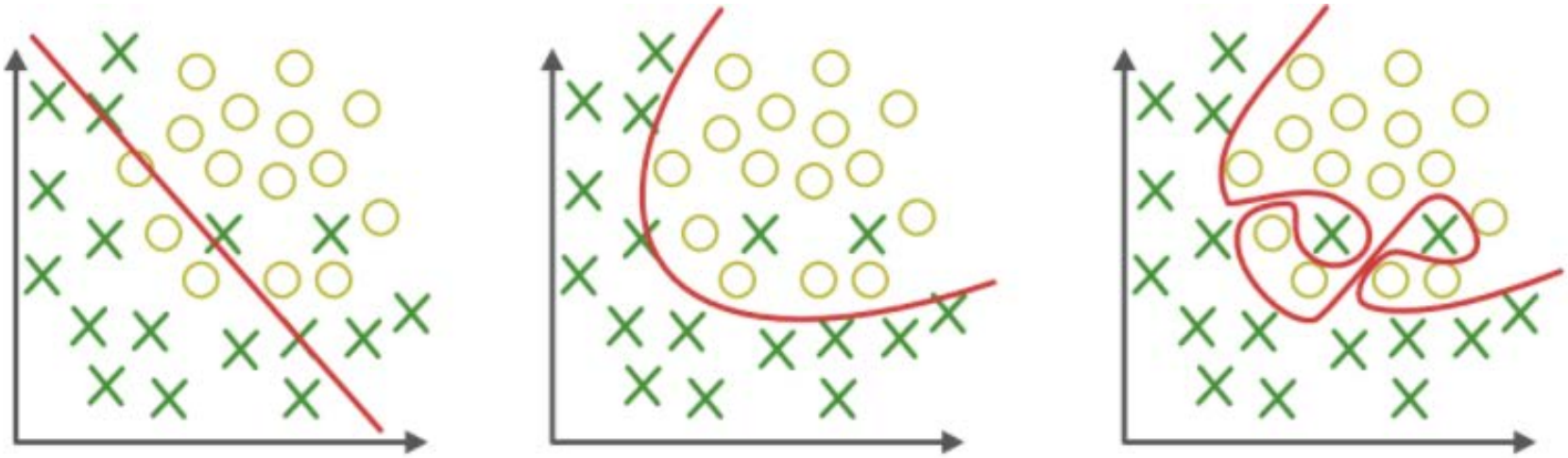
  - 이미지 분류 문제

훈련 데이터 새로운 데이터

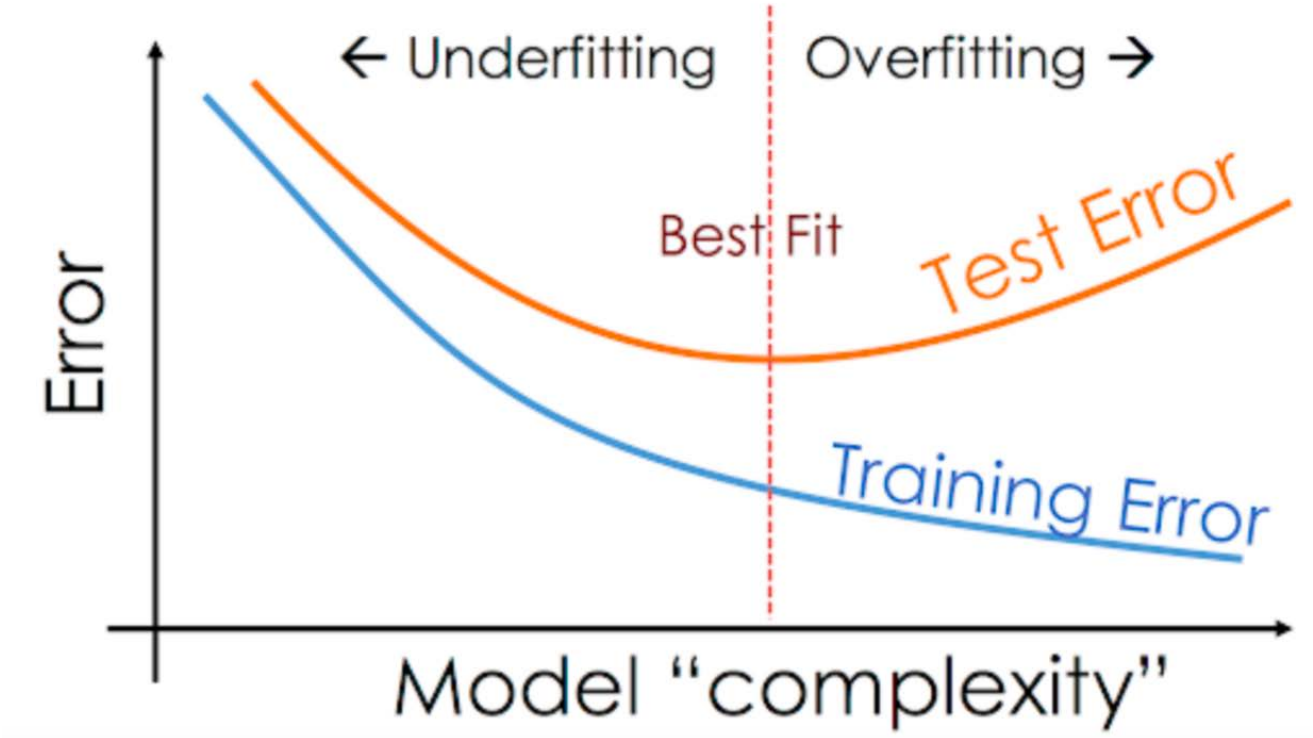
학습기 예측 결과
→ 이것은 나뭇잎이 아니다.
이유
→ 나뭇잎의 테두리는 톱니 모양이다.

Underfitting vs Overfitting

- 다음 중 바람직한(일반화된) fitting line은?



모델을 어떻게 검증하며 학습할 것인가?



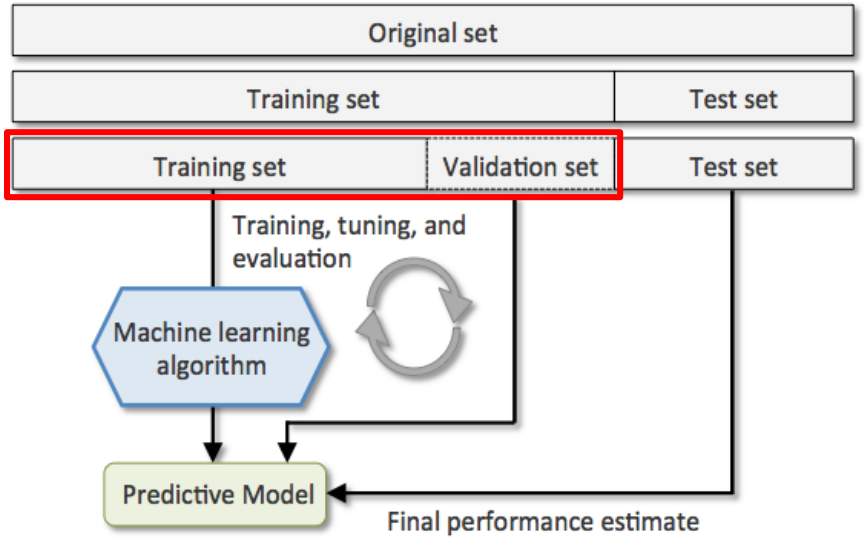
Cross validation

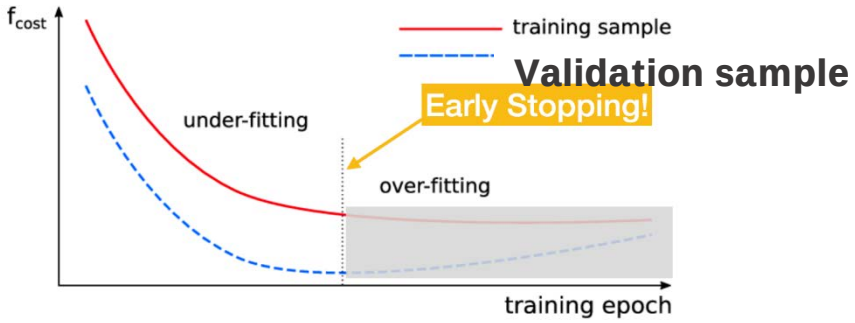
- 교차 검증
  - Why? 데이터 수가 적기 때문
  - 통계적인 평가 방법 / 얼마나 잘 일반화 되어 있는지를 평가
  - 데이터를 여러 번 반복해서 나누고 여러 모델을 학습
- 장점
  - 모든 데이터셋을 훈련에 활용 가능 & 모든 데이터 셋을 평가에 활용 가능
  - 정확도 향상
  - 특정 데이터 셋에 대한 과적합 방지 / 데이터 셋 규모가 적을 시 과소적합 방지
- 단점
  - 모델 훈련 및 평가 소요 시간 증가
- 교차검증 기법 종류
  - K-Fold Cross Validation (k겹 교차 검증)
  - Hold-Out Cross Validation
  - Leave-One-Out Cross Validation(LOOCV)
  - Stratified K-Fold Cross Validation (계층별 k겹 교차 검증)

Hold-Out Cross Validation(검증)

- 전체 데이터를 1) 학습 데이터 2) 검증 데이터 3) 테스트 데이터로 나눔
- 학습 데이터는 모델 학습에, 검증데이터는 하이퍼파라미터 튜닝, 테스트 데이터는 성능 추정에 사용
  - 테스트 데이터 기반 튜닝을 시도하는 것은 올바른 방법이 아님!

Training set을 한번 더 나누자
모델 검증을 위해!!

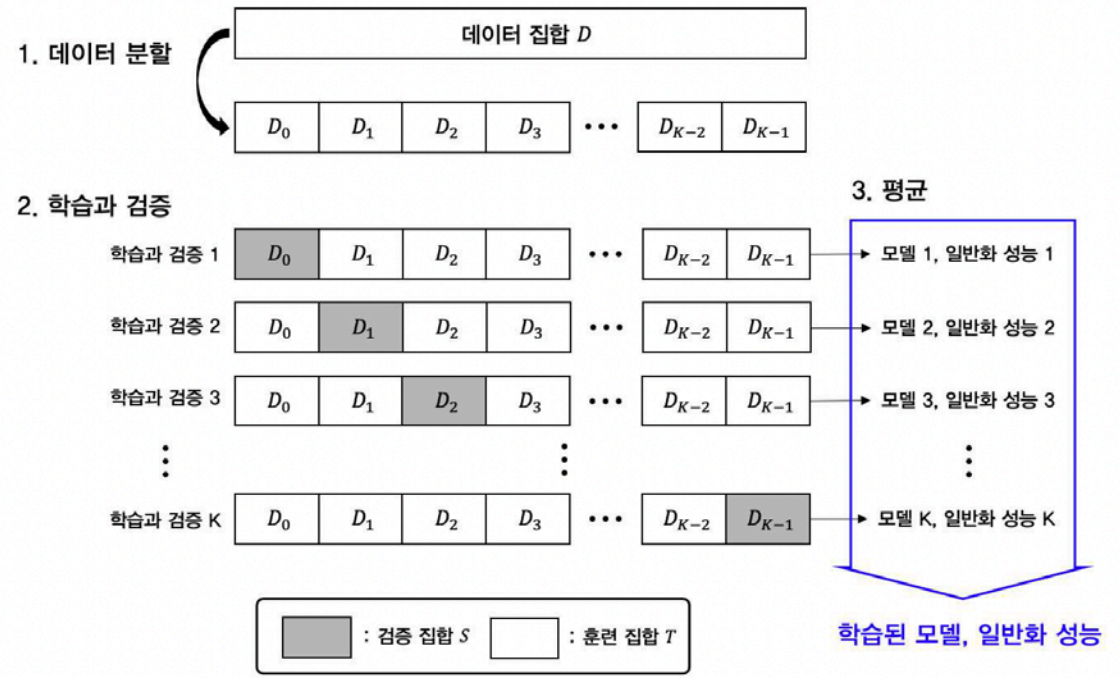




데이터를 어떻게 분할하느냐에 따라 성능이 좌우됨

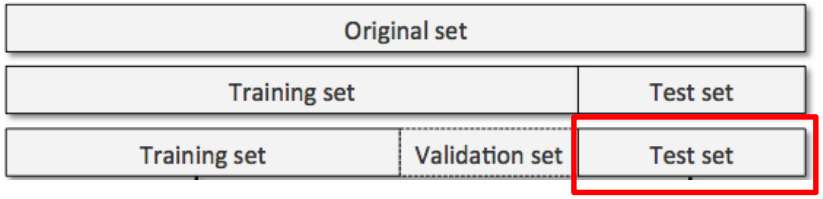
K-fold Cross validation

- 중복없이 훈련 데이터를 K겹으로 랜덤하게 나눔
  - 중복을 허락하지 않으므로 모든 샘플이 검증에 딱 1번 사용
  - 데이터를 고정 분할 또는 매번(학습과 검증 #마다) 데이터 랜덤 셔플링한 후 사용
- 홀드아웃 보다 데이터 분할에 덜 예민한 성능 평가 가능



❖ Leave one out cross validation(LOOCV): K와 데이터 수가 같을 경우를 지칭

머신러닝 모델 성능 평가



- 모델의 일반화 성능 평가의 기준: 성능 측정
- 오차(Error)
  - 전체 데이터에 대해, 실측값 – 모델의 예측값
  - 다양한 측정 방법이 가능하며 일반적으로 [0,1] 사이의 값을 갖도록 설계
- 평균 제곱 오차 (Mean squared error, MSE)
  - 회귀 분석의 대표적인 성능 측정 도구
  - 데이터 세트 $D = \{(\boldsymbol{x}_0, y_0), (\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_{N-1}, y_{N-1})\}$
  - 학습기 f의 이상적인 성능은 $f(\boldsymbol{x}) = y$, 즉 예측값이 실제값과 동일, 실제로는 오차 발생
  - 정의

$$\epsilon_{MSE}(f; D) = \frac{1}{N}\sum_{n=0}^{N-1}(f(\boldsymbol{x}_n) - y_n)^2$$

모델의 평가

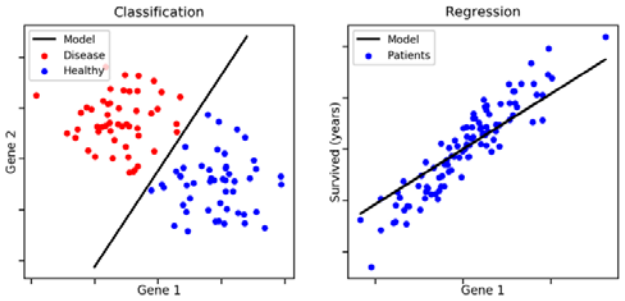
- Training set
  - 모델을 학습(Training)하기 위한 데이터 set
    - Weight Update

$$w_0[t+1] = w_0[t] - \alpha \frac{\partial}{\partial w_0} \epsilon_{MSE}(w_0, w_1)$$

(New: $w_0[t+1]$, Old: $w_0[t]$, Learning Rate: α, Old: w_0, w_1)

- Test set (미래의 data)
  - 모델을 평가(Test)하기 위한 데이터 set → 결정된 weight로 모델 평가

성능 측정(Performance Measure)



- 분류 문제에서의 오차율과 정확도
  - 분류 (classification)
    - 이진분류: 출력이 둘 중 하나 (예, pass or fail)
    - 다중분류: 출력이 여러 개 중 한 (예, A, B, C, D, F)
  - 오차율
    - 모든 데이터 중 잘못 분류한 데이터의 비율

$$P_{err}(f;D) = \frac{1}{N}\sum_{n=0}^{N-1} \boldsymbol{I}(f(x_n) \neq y_n)$$

  - 정확도
    - 모든 데이터 중 성공적으로 분류한 데이터의 비율

$$P_{acc}(f;D) = \frac{1}{N}\sum_{n=0}^{N-1} \boldsymbol{I}(f(x_n) = y_n) = 1 - P_{err}(f;D)$$

성능 측정(Performance Measure)

- Precision(정밀도) & Recall(재현율)

confusion matrix

		모델의 예측 결과	
		양성 (Positive)	음성 (Negative)
데이터의 실제 결과	양성 (Positive)	진짜 양성 (TP, True Positive)	가짜 음성 (FN, False Negative)
	음성 (Negative)	가짜 양성 (FP, False Positive)	진짜 음성 (TN, True Negative)

전체 데이터의 수 N = TP+FP+TN+FN

- 정확도 = $\frac{TP+TN}{N}$
- 오차율 = 1 - $\frac{TP+TN}{N}$ = $\frac{FP+FN}{N}$

- TP(True Positive): 실제 양성을 머신러닝 모델이 양성으로 판별
  - 실제 암에 걸렸는데 암에 걸렸다고 판단한 경우
- FP(False Positive): 실제 음성을 머신러닝 모델이 양성으로 판별
  - 실제 암에 걸리지 않았는데 암에 걸렸다고 잘못 판단한 경우
- FN(False Negative): 실제 양성을 머신러닝 모델이 음성으로 판별
- TN(True Negative): 실제 음성을 머신러닝 모델이 음성으로 판별

코로나 검사자 10명 중에, 정확하게 진단한 7명을 가려내면 70%의 정확도를 보임

그렇다면, 이 진단키트가 정확하다 말할 수 있을까? (--> 이것만 가지고 얘기하기에는 정확하지 않음)

성능 측정(Performance Measure)

- Precision(정밀도) & Recall(재현율)

		모델의 예측 결과	
		양성 (Positive)	음성 (Negative)
데이터의 실제 결과	양성 (Positive)	진짜 양성 (TP, True Positive)	가짜 음성 (FN, False Negative)
	음성 (Negative)	가짜 양성 (FP, False Positive)	진짜 음성 (TN, True Negative)

  - Recall(재현율) = $\frac{TP}{TP+FN}$
    - 실제 양성(True)인 샘플 중 모델이 양성으로 판별한 비율
    - 실제로 코로나 양성인 사람을 확진자로 분류할 확률
  - Precision(정밀도) = $\frac{TP}{TP+FP}$
    - 모델이 양성(True)으로 판단한 것 중에 실제 양성 비율
    - 진단키트가 확진자로 판정한 사람들 중 실제 코로나 양성인 사람일 확률

두 지표가 둘다 높은 모델이 일반적으로 좋은 모델 (항상은 아님)

  - 재현율이 중요한 지표:
    - 실제 positive 데이터를 negative로 판단하면 업무상 큰 영향을 미치는 경우
      - 실제 positive 암 환자를 negative로 잘못 판단하는 경우
      - 코로나 확진자를 비확진자로 잘못 분류한 경우 (슈퍼 전파자)
  - 정밀도가 중요한 지표:
    - 실제 negative 데이터를 positive로 잘못 판단하면 업무상 큰 영향이 발생하는 경우
      - 실제 negative 일반 메일을 positive인 스팸메일로 잘못 분류 → 이메일수신X(업무x)

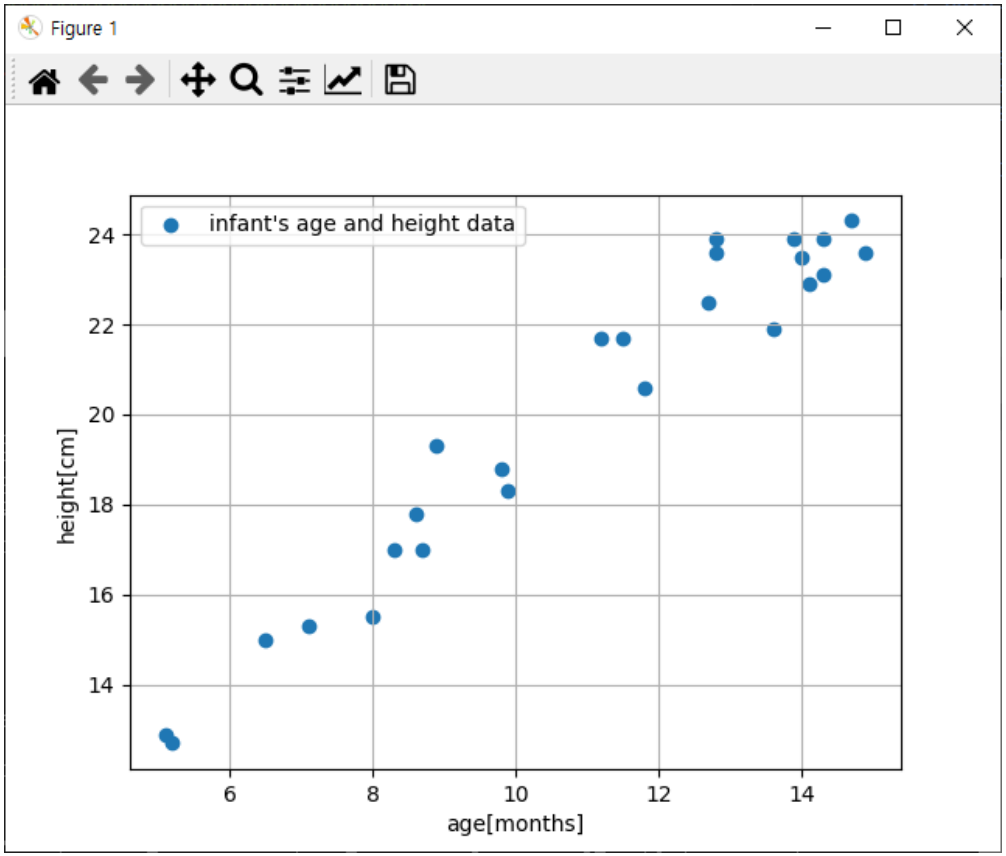
실 습

실습 #1

제공된 데이터 파일을 불러들여 x축은 나이, y축은 키를 나타내는 2차원 평면에 각 데이터의 위치를 점으로 표시하라. (lin_regression_data_03.csv: 총 25개의 data)

필수요소: x축, y축 이름, grid, legend

결과물: 코드, 그래프

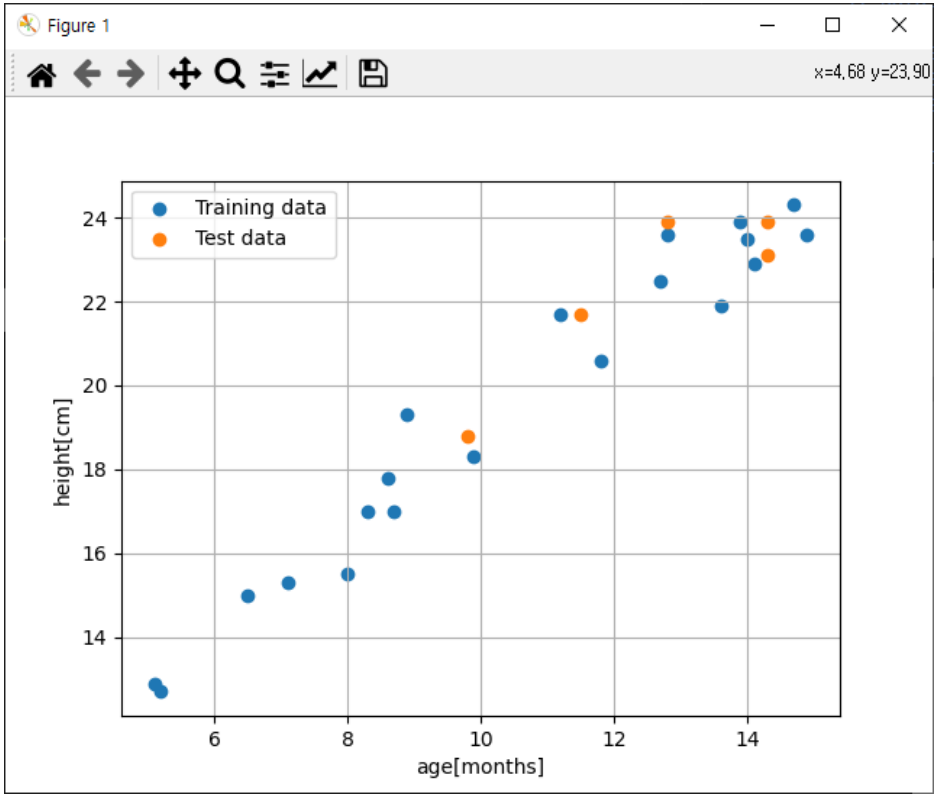


실습 #2

전체 데이터 중 첫 20개(1번~20번)를 훈련 집합(S)으로 후반 5개(21번~25번)를 테스트 집합(T)으로 나누고 각 집합의 데이터를 그래프로 나타내어라.

(주의: 데이터의 순서를 바꾸지 말 것)

결과물: 코드, 그래프



실습 #3

실습 #2에서 만든 훈련 집합을 적용해 K=6, 7, 8, 9, 10, 11, 12, 13일 때의 가우스 함수를 이용한 선형 기저함수 모델의 최적해를 구하라. (K는 기저함수의 개수를 의미함)

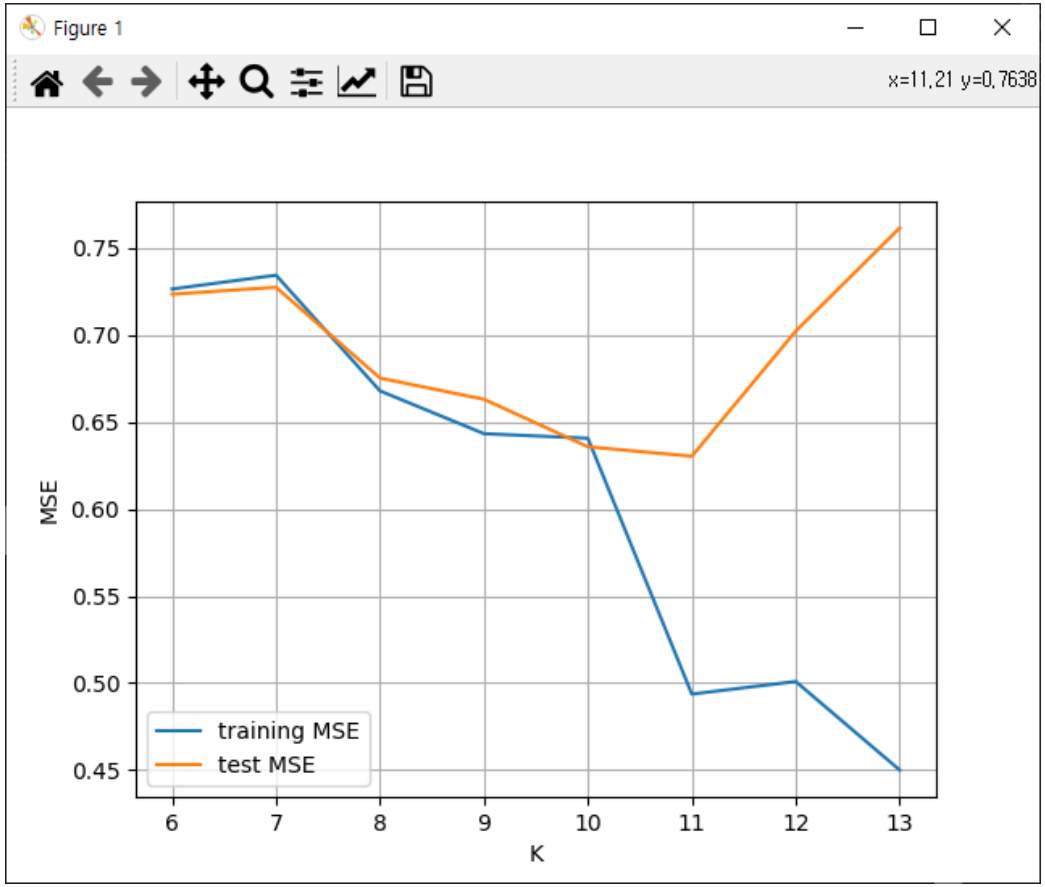
결과물: 코드, 최적해

```
k=6의 최적해 : [-7.63501681  2.53206263 -4.28859078  3.87117628 -1.61529661  4.74444806
 19.51782248]
k=7의 최적해 : [-8.13201556  0.62082535 -3.30574573 -0.14898306  0.61214225 -0.20214456
  2.65018211 21.12653219]
k=8의 최적해 : [ 0.87106964  7.97408664 -4.72343518 14.68212759 -6.4467811  17.61878216
 -5.06058052 17.13554107  7.44399784]
k=9의 최적해 : [-21.89483269   7.31964336 -21.0321158    5.87378214 -14.76465843
   1.47883791  -8.6565154    0.10144946  -7.92520385  33.05474   ]
k=10의 최적해 : [-13.21203484   7.78936264 -13.44715153   5.27237485  -6.80893693
   2.12747233  -3.23277825   2.46104792  -2.16263329   2.04727205
  22.95801065]
k=11의 최적해 : [ 24.61161006  -1.06569796  25.78720171  -4.01290625  31.86053366
  -6.60958517  35.39312291  -6.72730469  36.18333242  -6.25729531
  37.47028692 -14.89027091]
k=12의 최적해 : [19.25456643 -5.93405982 24.32761741 -8.21672434 25.16701736 -2.97774617
 18.52463791  9.38362867  6.99961857 20.21783312 -0.4588178  27.2270837
 -6.18124615]
k=13의 최적해 : [-10.1789781    3.40280306  -8.84164941   4.55002185 -12.7582108
  13.59730883 -19.6761914   19.7626584  -18.61359426  14.76012034
 -10.20718941   6.56356208  -1.07841873  22.05507319]
```

실습 #4

실습 #3에서 구한 선형 기저함수 모델의 평균제곱오차(MSE)를 훈련 집합과 테스트 집합에 대해 각각 구하고 그래프를 그려라.

결과물: 코드, 그래프

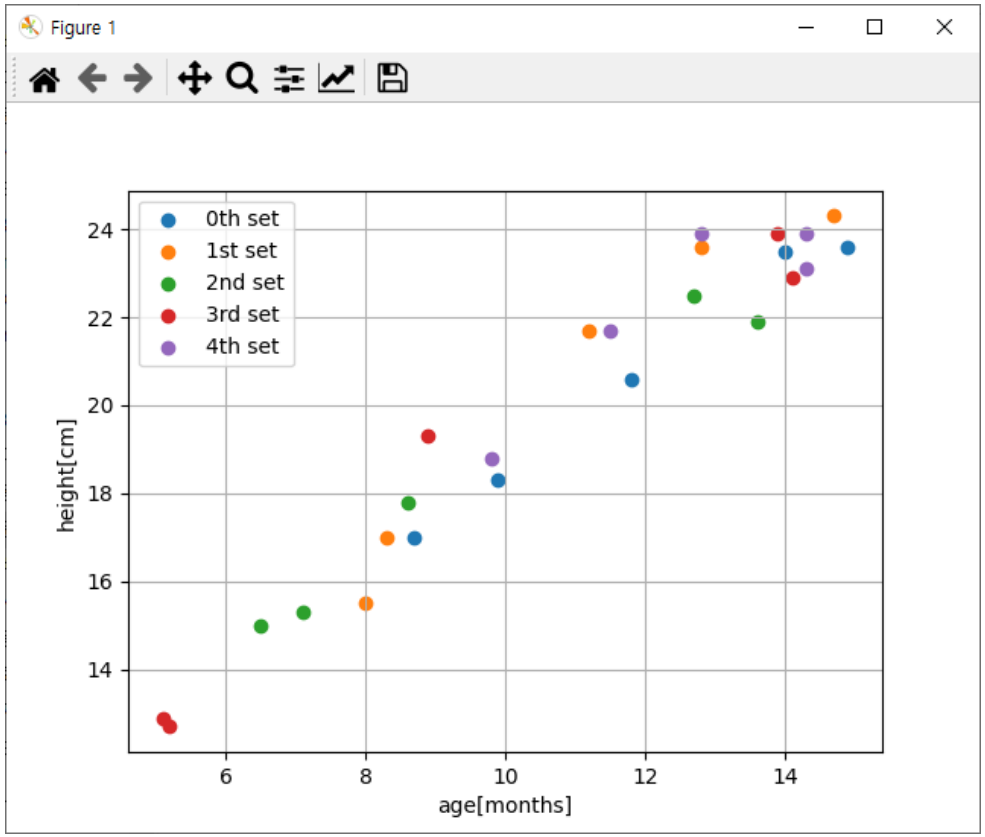


실습 #5

전체 데이터를 차례로 5등분하여 5개의 부분집합으로 나누고, 각 집합의 데이터를 x축은 나이, y축은 키를 나타내는 2차원 평면에 서로 다른 모양의 마커로 표시하라.

(k=5인 교차검증을 위한 준비 작업)

결과물: 코드, 그래프 (범례와 함께)



실습 #6

실습 #5에서 만든 다섯 개의 데이터 집합을 이용해 5겹 교차검증을 구현하려고 한다. 모델은 K=9일때의 가우스 함수를 이용한 선형 기저함수 모델을 사용.

이를 위해 5개의 홀드아웃 검증을 설계하고 각 홀드아웃의 결과물(매개변수, 일반화 오차)을 구하라.

결과물: 코드, 매개변수, 일반화 오차

실습 #7

실습 #6에서 각 홀드아웃의 결과로 생성된 선형 기저함수 모델을 각각의 훈련데이터, 검증데이터와 함께 그래프에 표시하라.

결과물: 코드, 그래프