기계학습실습

2025.10. 13.

SEJONG UNIVERSITY

공지사항

- **중간고사 (20점) : Kaggle 기반 문제풀이 (1문제, 20점)**
  - [001분반] 10월 20일 (월) 13:30~15:00 (13:20분까지 착석 및 준비)
  - [002분반] 10월 20일 (월) 15:00~16:30 (14:50분까지 착석 및 준비)
  - Baseline에 따른 구간별 최종순위를 점수화
    - Baseline1: 실습과제 에서와 유사한 수준
    - Baseline2: 기본적인 구현을 했을 경우 달성할 수 있는 수준
    - [구간1] Baseline 1 이상 [15~20점]
    - [구간2] Baseline 2 이상 ~ Baseline 1 미만 [10~14점]
    - [구간3] Baseline 2 미만 [0~9점]
  - 최종 학점은 절대평가로 부여

주간 계획 (수업계획서)

- 주간 계획에 따른 예습과 복습을 철저히!

주차	이론 (월)	실습 (수)	수업 일
1	수업 소개: 평가기준, 주간계획 등	Kaggle 소개	9/1, 9/3
2	인공지능과 머신러닝, 딥러닝	Kaggle Notebook, Google Colab, Pycharm 소개	9/8, 9/10
3	Pandas 라이브러리 소개	테스트 문제 풀이, 실습과제0: Kaggle 테스트코드 제출	9/15, 9/17
4	마켓과 머신러닝, 훈련 세트와 테스트 세트	실습과제1: ML 문제풀이 (평가)	9/22, 9/24
5	데이터 전처리1	실습과제2: ML 문제풀이 (평가)	9/29, 10/1
6	추석연휴	추석연휴	10/6, 10/8
7	회귀 (KNN회귀, 선형회귀)	실습과제3: ML 문제풀이 (평가)	10/13, 10/15
8	중간고사: 10월 20일(월)	(중간고사기간)	10/20
9	로지스틱 회귀	실습과제4: ML 문제풀이 (평가)	10/27, 10/29
10	확률적 경사하강법, 특성공학과 규제	실습과제5: ML 문제풀이 (평가)	11/3, 11/5
11	데이터 전처리2	실습과제6: ML 문제풀이 (평가)	11/10, 11/12
12	결정트리	실습과제7: ML 문제풀이 (평가)	11/17, 11/19
13	앙상블1	실습과제8: ML 문제풀이 (평가)	11/24, 11/26
14	앙상블2	실습과제9: ML 문제풀이 (평가)	12/1, 12/3
15	교차검증, 그리드서치, 시계열데이터처리	실습과제10: ML 문제풀이 (평가)	12/8, 12/10
16	기말고사: 12월 15일 (월)	(기말고사기간)	12/15

7-1주차

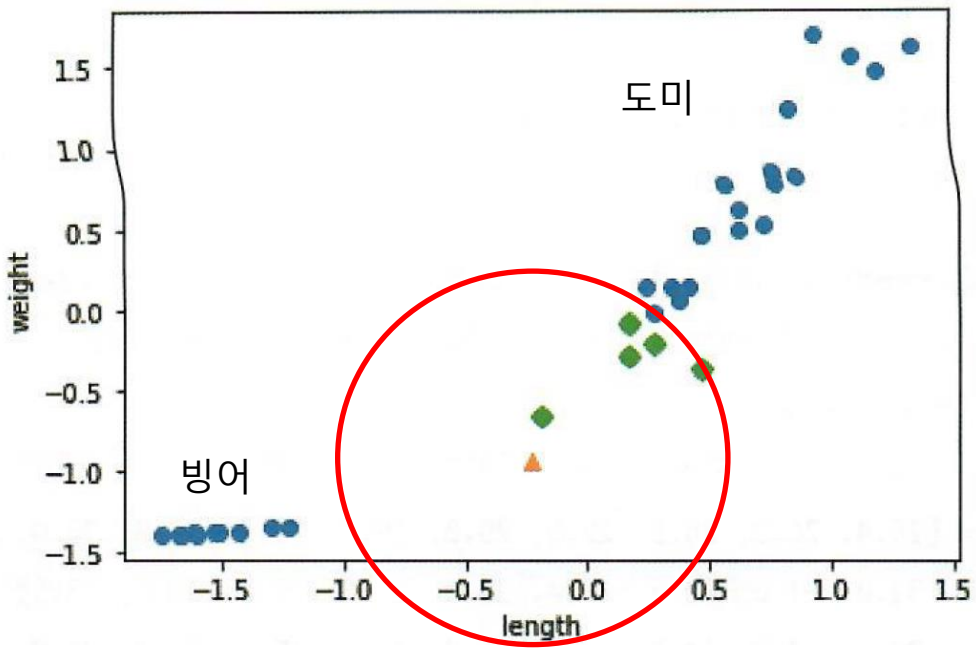
- 지난시간 문제복습 (실습과제2)
- k-최근접 이웃 회귀
- 선형 회귀

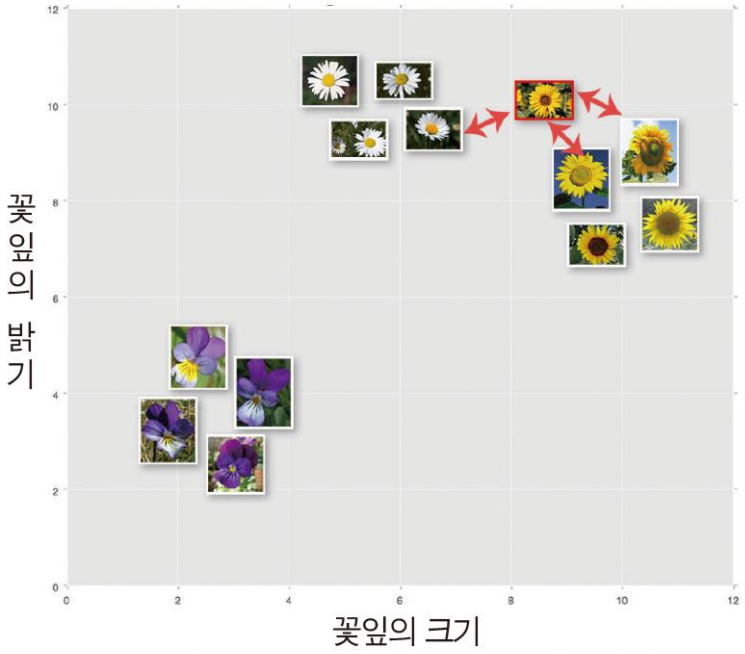
지난시간 문제복습 (실습과제1)

- 코드리뷰 (baseline)
  - 0.72361
- 코드 리뷰 (1위)
  - 전체1위: 0.78391 (이종태)

Chap. 3-1 k-최근접 이웃 회귀

- [복습] K-최근접 이웃 분류
  - K = 5
  - 테스트 샘플과 가장 가까운 K개의 샘플을 선택 → 다수결로 결정



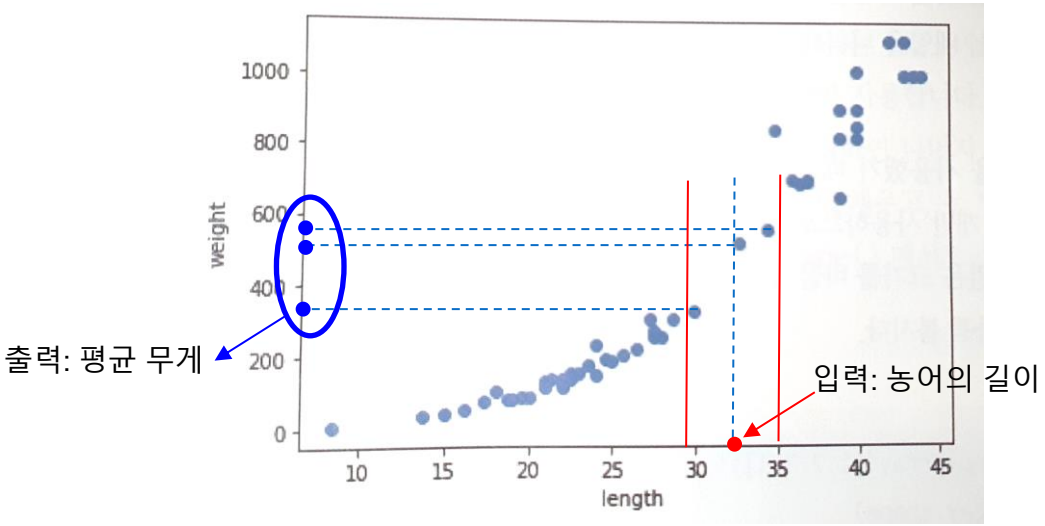


[그림 8.25] 꽃잎의 크기와 밝기에 따른 K-NN 분류

Chap. 3-1 k-최근접 이웃 회귀

- K-최근접 이웃 회귀
  - Ex) 농어의 길이 데이터로 무게를 예측
  - 테스트 샘플과 가장 가까운(length) K개의 샘플을 선택

    → 무게를 평균 내어 예측 값으로 사용



Chap. 3-1 k-최근접 이웃 회귀

- `sklearn.neighbors.KNeighborsRegressor`

  `sklearn.neighbors.KNeighborsClassifier`

- 주요 파라미터
  - n_neighbors : *int, default=5*
    - 고려대상인 이웃 값의 수
    - 값이 큰 경우 일반적인 패턴을 따라가고 작은 경우 국지적인 패턴에 민감
  - Weights : *{'uniform', 'distance'} or callable, default='uniform'*
    - 고려대상인 샘플에 대한 가중치 함수 ('uniform':일정하게, 'distance': 1/distance)
  - Algorithm*{'auto', 'ball_tree', 'kd_tree', 'brute'}, default='auto'*
    - 이웃 값을 찾는 알고리즘
  - P : *int, default=2*
    - *Minkowski distance (P=2* 인 경우 *Euclidean distance)*
  - Metric : *str or callable, default='minkowski'*
    - 거리를 계산하는 알고리즘 선택

Chap. 3-1 k-최근접 이웃 회귀

- 회귀에서의 performance metric
  - 결정계수 (coefficient of determination) or R^2
  - MAE (mean absolute error)
  - MSE (mean squared error) 등
- $R^2 = 1 - SSR / SST$

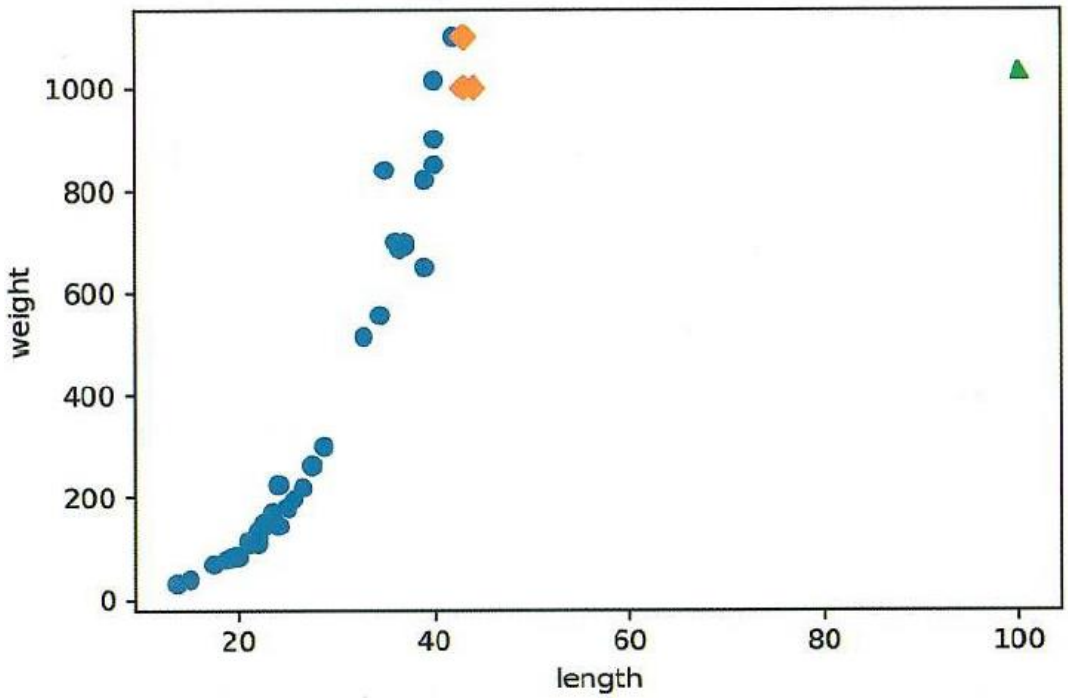
$$SSR = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2$$

$$SST = \sum_{i=1}^{n} \left(y_i - \overline{y}\right)^2$$

  - SSR (Residual Sum of Squares) : (타깃 – 예측)^2 의 합
  - SST (Total Sum of Squares) : (타깃 – 평균)^2 의 합
    - 타깃: 예측해야 하는 값. 즉 정답을 의미
    - Ex) 모델이 타깃의 평균 정도를 예측하는 수준 → R^2 = 0 에 가까워 짐
    - 예측이 타깃에 매우 가까운 값 → R^2 = 1 에 가까워 짐
  - Ex) KNeighborsRegressor.score(test_target, test_prediction)
- MAE
  - 타깃과 예측의 절대값 오차를 평균
  - Ex) sklearn.metrics.mean_absolute_error (y_true, y_pred)
- MSE
  - 타깃과 예측의 오차 제곱을 평균
  - Ex) sklearn.metrics.mean_squared_error(y_true, y_pred)

[주 교재] 선형회귀

- K-Nearest Neighbor (KNN) vs. Linear Regression (LR)
  - 실제 길이 100 cm, 무게 7kg 농어 → KNN 모델을 사용하여 정확히 7kg을 예측해 내는지 확인해 보자!
  - 길이 100 → KNN 모델(k=3) → 약 1.1 kg 으로 예측
  - → 모델 개선이 필요 (이 상황에서는 선형회귀가 더 유용 함)



[주 교재] 선형회귀

- 선형회귀의 예: 공부한 시간과 성적사이의 상관관계 (선형방정식) 구하기

공부한 시간	2시간	4시간	6시간	8시간
성적	81점	93점	91점	97점

표 3-1 공부한 시간과 중간고사 성적 데이터

- 여기서 공부한 시간을 x라 하고 성적을 y라 할 때 집합 X와 집합 Y를 다음과 같이 표현할 수 있음
  - x = {2, 4, 6, 8}
  - y = {81, 93, 91, 97}

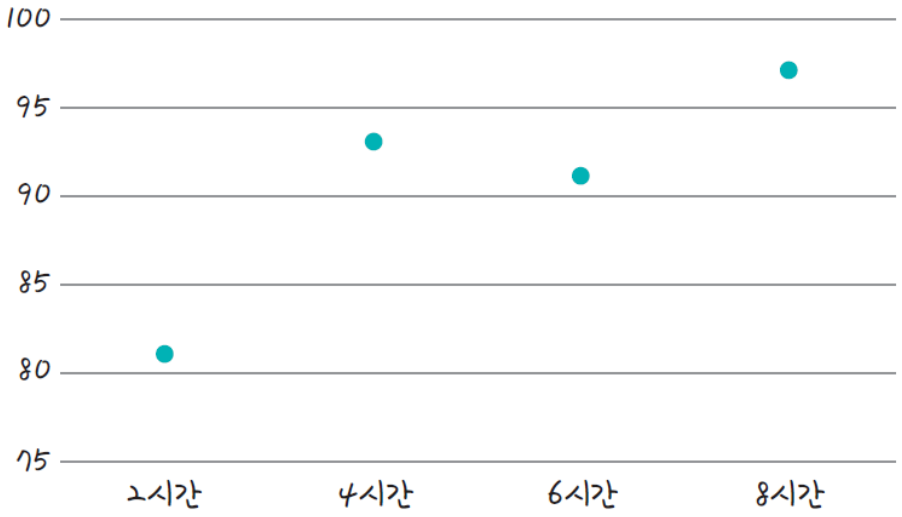
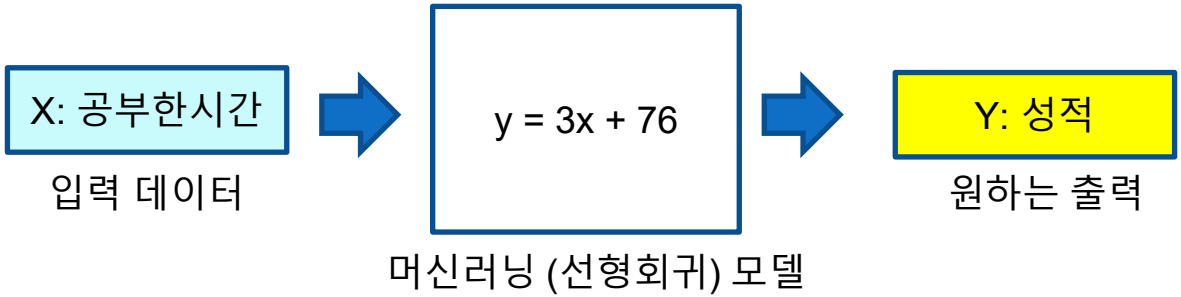
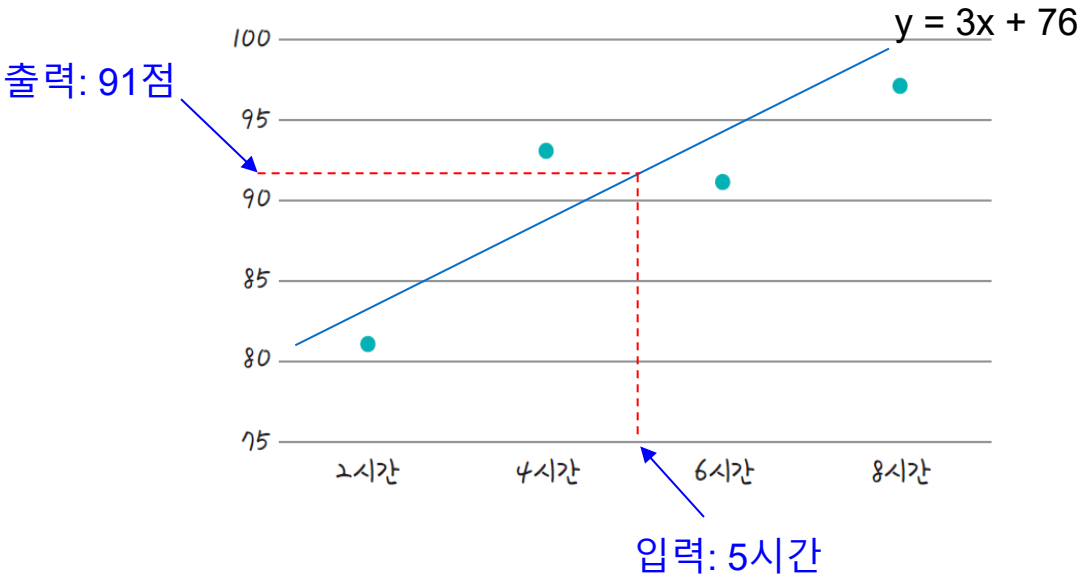


그림 3-1 공부한 시간과 성적을 좌표로 표현

[주 교재] 선형회귀

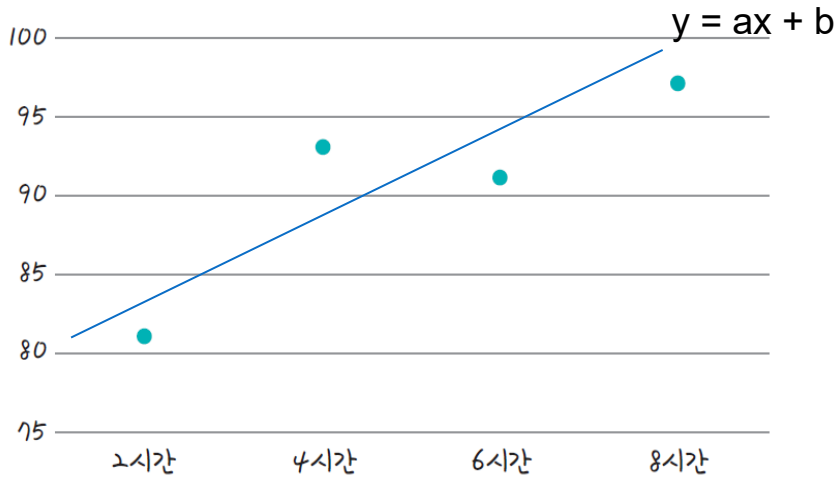
- 선형회귀 모델의 활용
  - a = 3, b = 76 이라고 가정
  - 5시간 공부하면 성적이 어떨까? → 주어진 선형 모델에 입력 → 3 x 5 + 76 = 91 : 91점 으로 예측





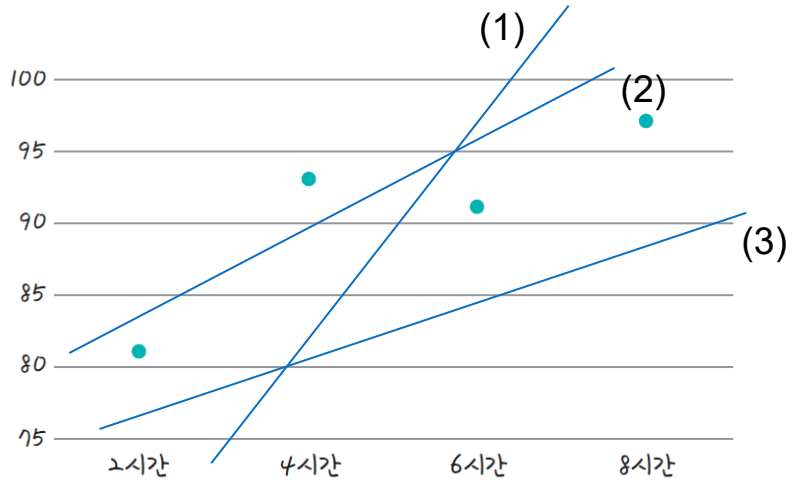
[주 교재] 선형회귀

- **선형회귀 모델 학습**
  - 입력 x 와 y 사이의 선형관계를 찾는 것
  - a: 기울기, b: y절편 를 찾는 것
  - Ex) a = 3, b = 76 라는 값을 찾는 것
  - 필요한 것들
    - Loss (Cost) 함수
    - 최적화: a와 b의 최적 값을 찾는 방법 (loss 함수 최소화)



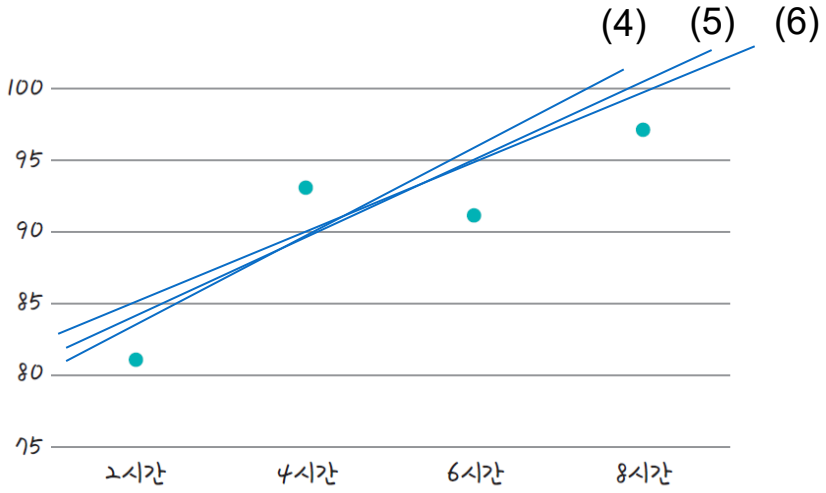
[주 교재] 선형회귀

- 예측의 정확도
  - 공부한시간에 따른 성적을 잘 예측하기 위해 주어진 데이터를 잘 표현하는/대표하는 직선을 찾아야 한다.
  - 어떤 직선이 정확한 직선일까? → 셋 중 (2) 직선이 가장 정확함



[주 교재] 선형회귀

- **정확한 모델**
  - 다음의 경우는 어떤 직선이 가장 정확한 직선일까?
  - → 정량적인 판단기준이 필요함
  - (4) (5) (6) 직선을 구별하는 것은 기울기 a와 y절편 b
  - → a 및 b와 관련된 정확한 모델링의 기준이 필요 함



[주 교재] 선형회귀

- 먼저, 임의의 선을 그어보자
  - 기울기 a 와 y절편 b 를 임의의 수 3과 76이라고 가정 → y = 3 x + 76
  - 그림 3-5와 같은 임의의 직선이 어느 정도의 차이(오차)가 있는지를 확인하려면 각 점과 그래프 사이의 거리를 잰다
  - 이러한 차이(오차) 를 정확한 모델링을위한 지표로 둔다.
  - 즉, 오차가 가장 작도록 a, b를 찾는다
    - → 모델 정확도의 지표로 합리적임

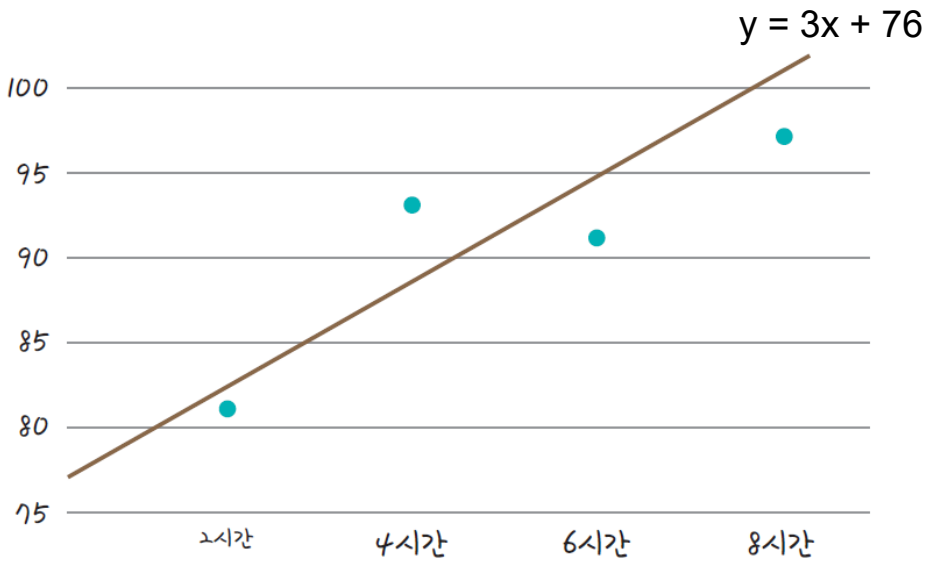


그림 3-5 임의의 직선 그려보기

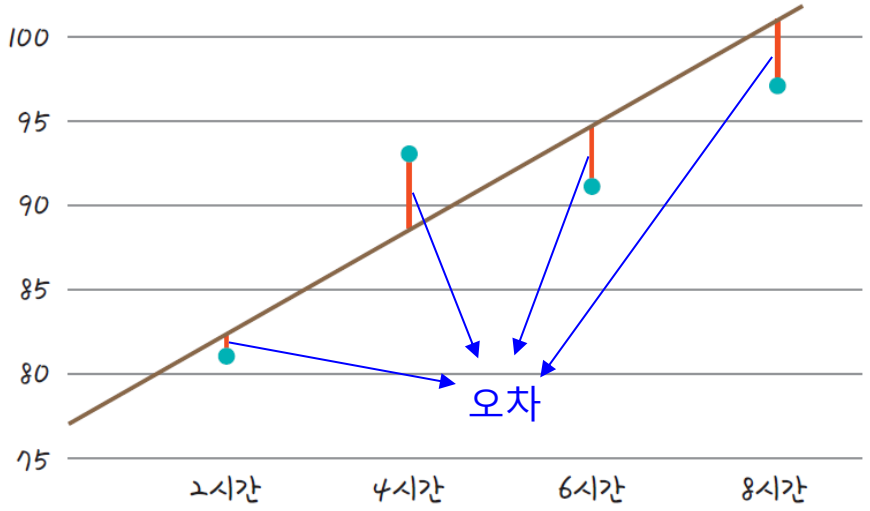


그림 3-6 임의의 직선과 실제 값 사이의 거리

[주 교재] 선형회귀

- 모델 정확도와 오차
  - 그림 3-7, 그림 3-8 에서 볼 수 있는 빨간색 선은 직선이 잘 그어졌는지를 나타냄
  - 이 직선들의 합이 작을수록 잘 그어진 직선이고, 이 직선들의 합이 클수록 잘못 그어진 직선이 됨

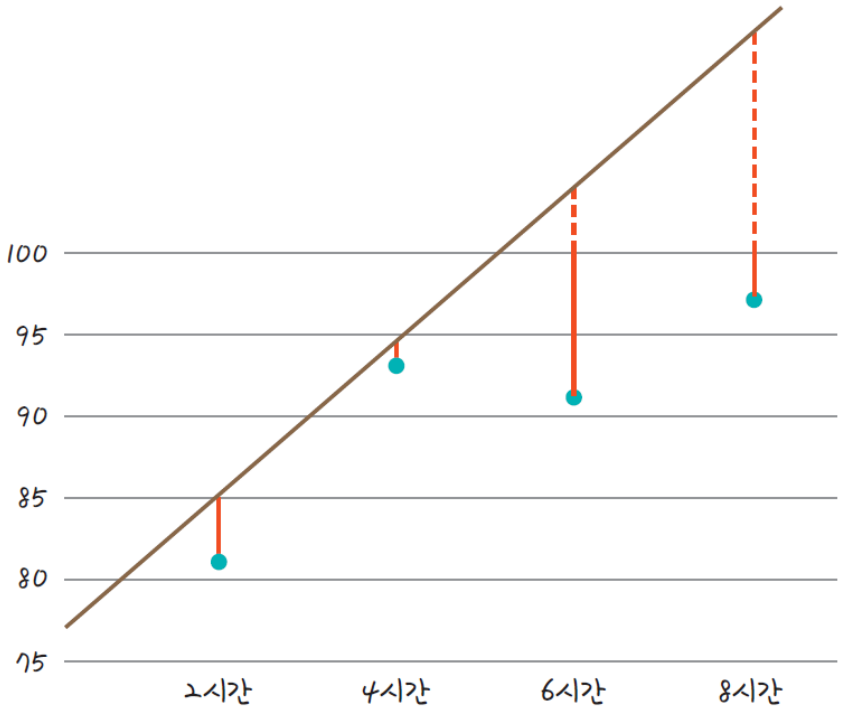


그림 3-7 기울기 a 를 너무 크게 잡았을 때의 오차

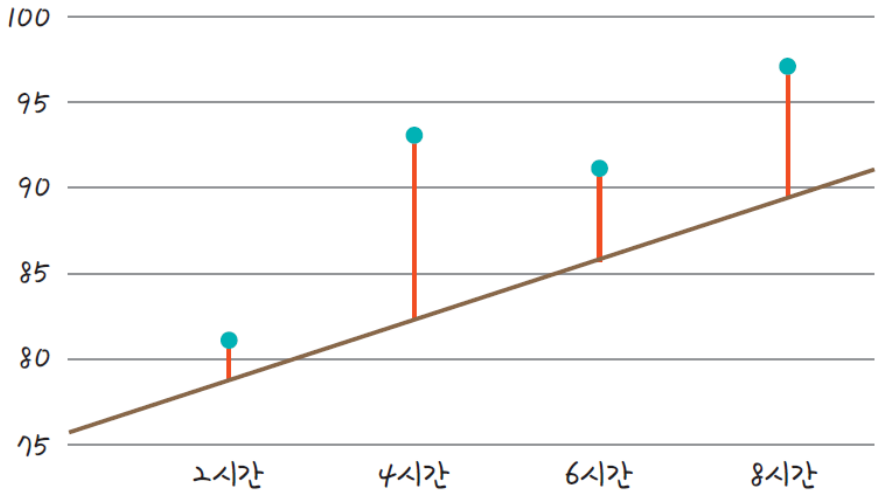


그림 3-8 기울기 a 를 너무 작게 잡았을 때의 오차

[주 교재] 선형회귀

- **모델 정확도와 오차**
  - 그래프의 기울기 a와 y절편 b가 잘못 되었을수록 빨간색 선의 거리의 합, 즉 오차의 합도 커짐
    - Ex) 기울기가 무한대로 커지면 오차도 무한대로 커지는 상관관계가 있는 것을 알 수 있음
  - 거리(오차)는 입력 데이터에 나와 있는 y의 '실제 값'과 x를 나오는 '예측 값'과의 차이를 통해 구할 수 있음 오차 = 예측 값 – 실제 값 식에 대입해서
    - Ex) a = 3, b = 76인 경우, 오차계산의 예

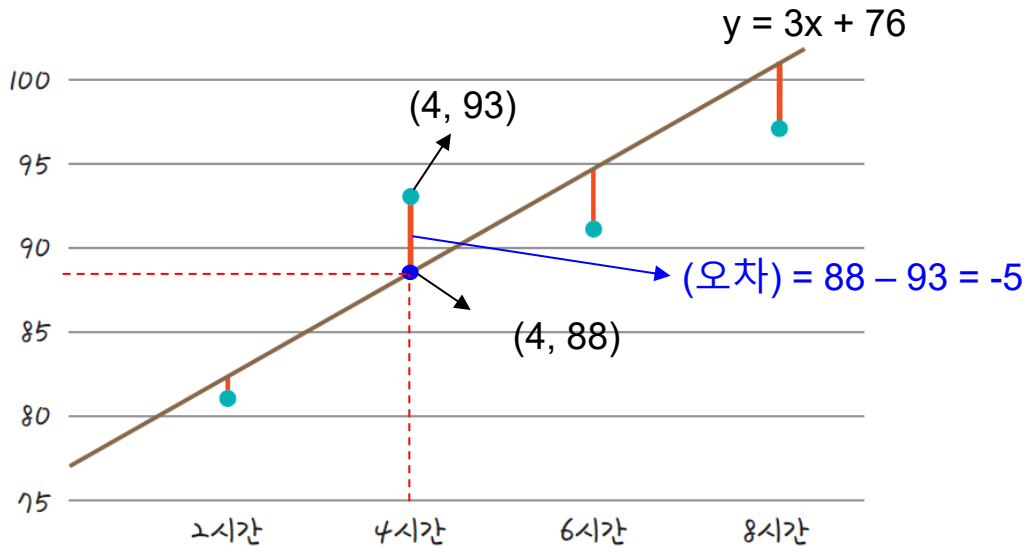


그림 3-6 임의의 직선과 실제 값 사이의 거리

[주 교재] 선형회귀

- 오차 계산

공부한 시간(x)	2	4	6	8
성적(실제 값, y)	81	93	91	97
예측 값 $\hat{y}_i$	82	88	94	100
오차	1	−5	3	3

표 3-3 주어진 데이터에서 오차 구하기

- 이렇게 해서 구한 오차를 모두 더하면 1 + (-5) + 3 + 3 = 2가 됨
- 이 값은 오차가 실제로 얼마나 큰지를 가늠하기에는 적합하지 않음
- 오차에 양수와 음수가 섞여 있어서 오차를 단순히 더해 버리면 합이 0이 될 수도 있기 때문임
- 부호를 없애야 정확한 오차를 구할 수 있음

➔ 오차의 제곱 (squared error) 사용!

[주 교재] 선형회귀

- 오차계산

	i=1	i=2	i=3	i=4	n=4
공부한 시간(x)	2	4	6	8	
성적(실제 값, y)	81	93	91	97	
예측 값 $\hat{y}_i$	82	88	94	100	
오차의 제곱	1	25	9	9	

표 3-3 주어진 데이터에서 오차 구하기

- 오차의 합을 구할 때는 각 오차의 값을 제곱해 준다 $\text{오차의 합} = \sum_{i}^{n} (\hat{y}_i - y_i)^2$
- y_i 는 x_i에 대응하는 '실제 값' 이고 $\hat{y}_i$는 x_i가 대입되었을 때 직선의 방정식 (여기서는 y = 3 x + 76) 이 만드는 '예측 값' 임
- 이 식으로 오차의 합을 다시 계산하면 1 + 25 + 9 + 9 = 44임
- 데이터가 많아질 수록 (n 값이 클 수록) 오차의 합이 커짐 → 오차의 크고 작은 정도를 파악하기 불편

➔ 평균 (Mean) 사용!

[주 교재] 선형회귀

- 오차계산

총 데이터 개수 $n=4$

	$i=1$	$i=2$	$i=3$	$i=4$
공부한 시간(x)	2	4	6	8
성적(실제 값, y)	81	93	91	97
예측 값 $\hat{y}_i$	82	88	94	100
오차의 제곱	1	25	9	9

(평균) = 11

표 3-3 주어진 데이터에서 오차 구하기

- 평균자승오차 (Mean Squared Error, MSE)

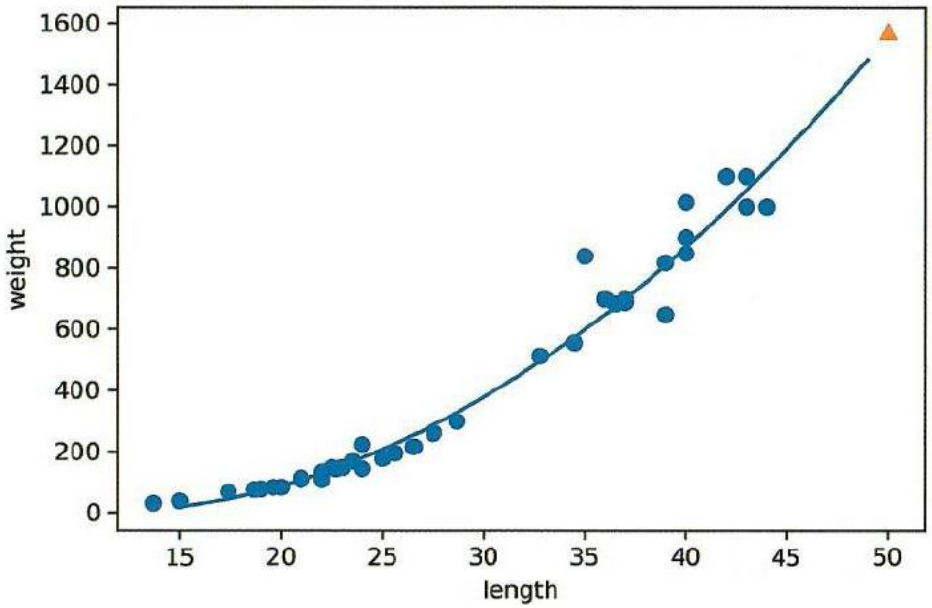
$$\text{평균 제곱 오차(MSE)} = \frac{1}{n}\sum (\hat{y}_i - y_i)^2$$

  - (MSE) = 44 / 4 = 11
  - 즉, 하나의 데이터에 대해 평균적으로 11 만큼의 제곱오차가 발생했다는 뜻
- 이러한 MSE를 모델 정확도의 정량지표로 활용 → MSE를 cost (loss) 함수라고 한다

Cost 함수(MSE)를 최소화하도록 모델파라미터 (a, b 값)를 찾는 것이 바로 머신러닝(선형회귀) 이다

[주 교재] 선형회귀

- (참고) 다항회귀 (Polynomial Regression)
  - 회귀곡선: $y = ax^2+bx+c$ 의 형태
    - $y = az + bx + c$ 로 보면 선형회귀
    - Ex) $y = ax^2$ 의 형태 → weight 와 $(length)^2$ 의 선형회귀



[주 교재] 선형회귀

- **sklearn.linear_model.LinearRegression**
  - https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
  - Parameters
    - fit_intercept: y절편(상수항) 고려 여부, default = True
    - n_jobs: 계산에 활용할 CPU 코어 개수, -1은 최대 사용, default = None
  - Methods
    - score: Return the coefficient of determination (R^2) of the prediction
      - 모델이 타깃의 평균 정도를 예측하는 수준 → R^2 = 0 에 가까워 짐
      - 예측이 타깃에 매우 가까운 값 → R^2 = 1 에 가까워 짐