

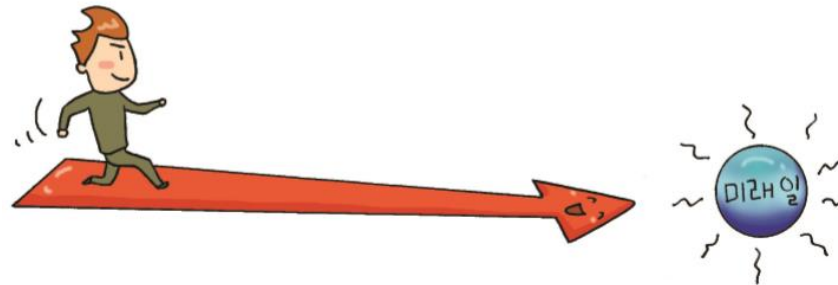
Lecture 15

선형 회귀 (Linear Regression)

1. 선형 회귀

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 딥러닝은 여러가지 통계의 결과들이 무수히 얹히고설켜 이루어지는 복잡한 연산의 결정체
- 딥러닝을 이해하려면 딥러닝의 가장 말단에서 이루어지는 가장 기본적인 두 가지 계산 원리, **선형 회귀**(Linear Regression) 와 **로지스틱 회귀**(Logistic Regression)를 알아야 함
- 가장 훌륭한 예측선 긋기란 통계학 용어인 **선형 회귀**(Linear Regression)를 쉽게 풀어서 쓴 것
- 머신러닝은 제대로 된 선을 긋는 작업부터 시작됨
- 선의 방향을 잘 정하면 그 선을 따라가는 것만으로도 지금은 보이지 않는 미래의 것을 예측할 수 있기 때문



1. 선형 회귀

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

"학생들의 중간고사 성적이 다 다르다."

- 위 문장이 나타낼 수 있는 정보는 너무 제한적
- 학급의 학생마다 제각각 성적이 다르다는 당연한 사실 외에는 알 수 있는 게 없음

"학생들의 중간고사 성적이 []에 따라 다 다르다."

- 이 문장은 정보가 담길 여지를 열어 놓고 있음
- [] 부분에 시험 성적을 좌우할 만한 여러 가지 것이 들어간다면 좀 더 많은 사실을 전달할 수 있음
- 예를 들면 공부한 시간, 시험 당일의 컨디션, 사교육비 지출액 등이 들어갈 수 있음
- 무엇이 들어가든지 해당 성적의 이유를 나름대로 타당하게 설명할 수 있음
- 따라서 이 문장이 중간고사 성적의 차이와 이유를 나타낼 때 더욱 효과적

1. 선형 회귀

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

"학생들의 중간고사 성적이 []에 따라 다 다르다."

- 여기서 []에 들어갈 내용을 '정보'라고 함
- 머신러닝과 딥러닝은 이 정보가 필요함

→ 많은 정보가 더 정확한 예측을 가능하게하며 이때의 '많은 정보'가 곧 '빅데이터'

1. 선형 회귀

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

선형 회귀(Linear Regression)의 정의

■ 독립 변수

x 값이 변함에 따라 y 값도 변한다는 정의 안에서, 독립적으로 변할 수 있는 x 값

■ 종속 변수

x 값이 변함에 따라 y 값도 변한다는 정의 안에서, 독립 변수 x 에 따라 종속적으로 변하는 y 값

■ 선형 회귀

x 값이 변함에 따라 y 값도 변한다는 정의 안에서,

독립 변수 x 를 사용해 종속 변수 y 의 움직임을 예측하고 설명하는 작업을 말함

1. 선형 회귀

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

선형 회귀(Linear Regression)의 종류

- 단순 선형 회귀(Simple Linear Regression)
하나의 x 값 만으로도 y 값을 설명할 수 있을 때
- 다중 선형 회귀(Multiple Linear Regression)
여러 개의 x 값으로 y 값을 설명할 수 있을 때

2. 최고의 예측선 구하기

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 우선 독립 변수가 하나뿐인 **단순 선형 회귀**의 예를 살펴보자.

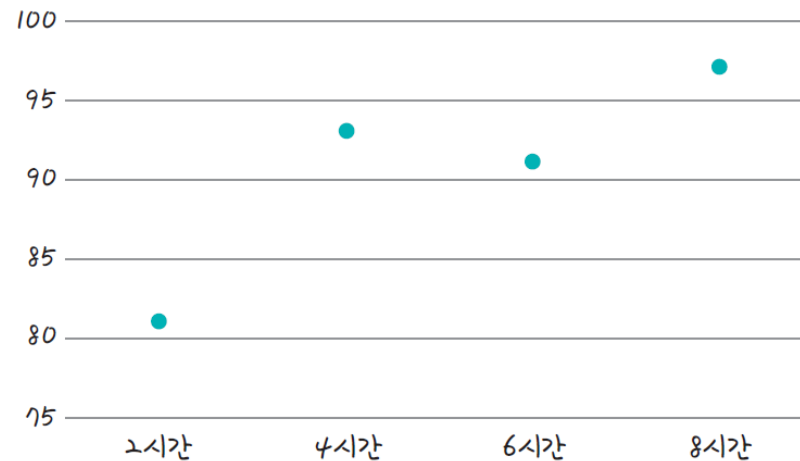
공부한 시간	2시간	4시간	6시간	8시간
성적	81점	93점	91점	97점

공부한 시간과 중간고사 성적 데이터

- 여기서 공부한 시간을 x , 성적을 y 라고 할 때 집합 X 와 집합 Y 를 다음과 같이 표현할 수 있음

$$X = \{2, 4, 6, 8\}$$

$$Y = \{81, 93, 91, 97\}$$



2. 최고의 예측선 구하기

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- **선형 회귀**를 공부하는 과정은 이 점들의 특징을 가장 잘 나타내는 선을 그리는 과정과 일치함
- 여기에서 선은 직선이므로 곧 **일차 함수 그래프**임

$$y = ax + b$$

- 여기서 x 값은 **독립 변수**이고 y 값은 **종속 변수**임
- 즉, x 값에 따라 y 값은 반드시 달라짐
- 다만, 정확하게 계산하려면 상수 a 와 b 값을 알아야 함
- 이 직선을 훌륭하게 그으려면 직선의 기울기 a 값과 절편 b 값을 정확히 예측해야 함

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- **최소 제곱법(Method of Least Squares)**이라는 공식을 알고 적용한다면,
이를 통해 **일차 함수의 기울기** a 와 **절편** b 를 바로 구할 수 있음
- **x값**(입력 값, 여기서는 '공부한 시간')과 **y값**(출력 값, 여기서는 '성적')일 때
이를 이용해 **기울기** a 를 구하는 방법은 다음과 같음

$$y = ax + b \quad a = \frac{(x - x \text{ 평균})(y - y \text{ 평균}) \text{의 합}}{(x - x \text{ 평균})^2 \text{의 합}}$$

- 이것이 바로 **최소 제곱법**
- 근사적으로 구하려는 해와 실제 해의 오차의 제곱의 합이 최소가 되는 해를 구하는 방법
- **최소 제곱법**은 x 가 지금처럼 **공부한 시간에 해당하는 하나의 입력**일 때 적용가능하며,
여러 개의 x 가 주어질 경우에는 '**경사 하강법**'을 사용하게 됨

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

$$y = ax + b$$

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

기울기 a 를 구하는 방법

- x 의 편차(각 값과 평균과의 차이)를 제곱해서 합한 값을 분모로 놓고,
 x 와 y 의 편차를 곱해서 합한 값을 분자로 놓으면 기울기가 나온다는 뜻

- 공부한 시간(x) 평균: $(2 + 4 + 6 + 8) \div 4 = 5$

- 성적(y) 평균: $(81 + 93 + 91 + 97) \div 4 = 90.5$

- 이를 식에 대입하면 다음과 같음

$$a = \frac{(x - x \text{ 평균})(y - y \text{ 평균}) \text{의 합}}{(x - x \text{ 평균})^2 \text{의 합}}$$

$$\begin{aligned}
 a &= \frac{(2 - 5)(81 - 90.5) + (4 - 5)(93 - 90.5) + (6 - 5)(91 - 90.5) + (8 - 5)(97 - 90.5)}{(2 - 5)^2 + (4 - 5)^2 + (6 - 5)^2 + (8 - 5)^2} \\
 &= \frac{46}{20} \\
 &= 2.3
 \end{aligned}$$

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

$$y = ax + b$$

y 절편인 b 를 구하는 방법

$$b = y \text{의 평균} - (x \text{의 평균} \times \text{기울기 } a)$$

- 즉, y 의 평균에서 x 의 평균과 기울기 a 의 곱을 빼면 b 의 값이 나온다는 의미

$$\begin{aligned} b &= 90.5 - (5 \times 2.3) \\ &= 79 \end{aligned}$$

- 이제 다음과 같이 예측 값을 구하기 위한 직선의 방정식이 완성됨

$$y = 2.3x + 79$$

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 예측 값 : x 를 대입했을 때 나오는 y 값

최소 제곱법(Method of Least Squares) 공식으로 구한 성적 예측 값

$$y = 2.3x + 79$$

공부한 시간	2	4	6	8
성적	81	93	91	97
예측 값	83.6	88.2	92.8	97.4

$$y = ax + b$$

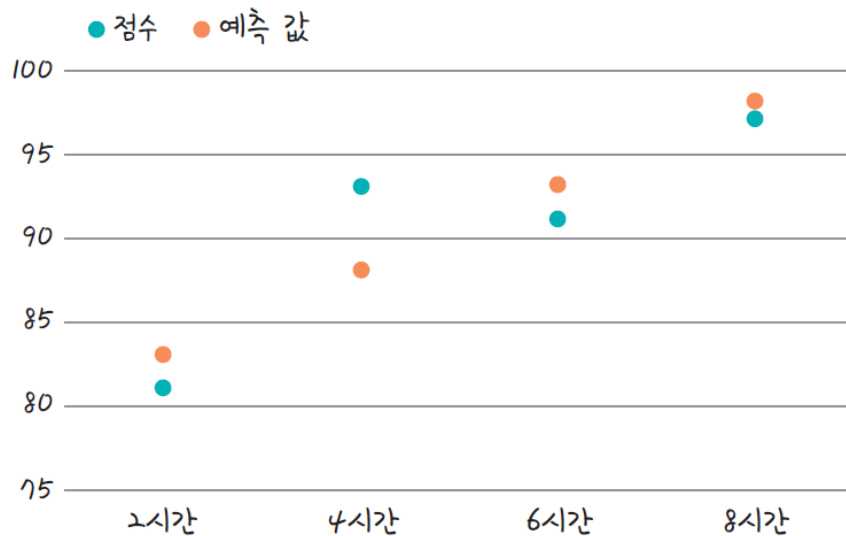
1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

공부한 시간	2	4	6	8
성적	81	93	91	97
예측 값	83.6	88.2	92.8	97.4

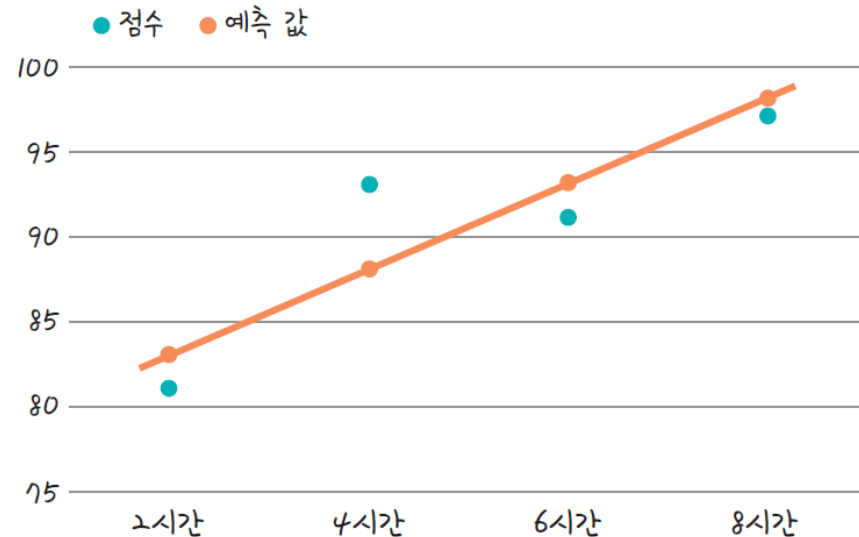
최소 제곱법 공식으로 구한 성적 예측 값

$$y = 2.3x + 79$$

- 좌표 평면에 이 예측 값을 찍어 보자

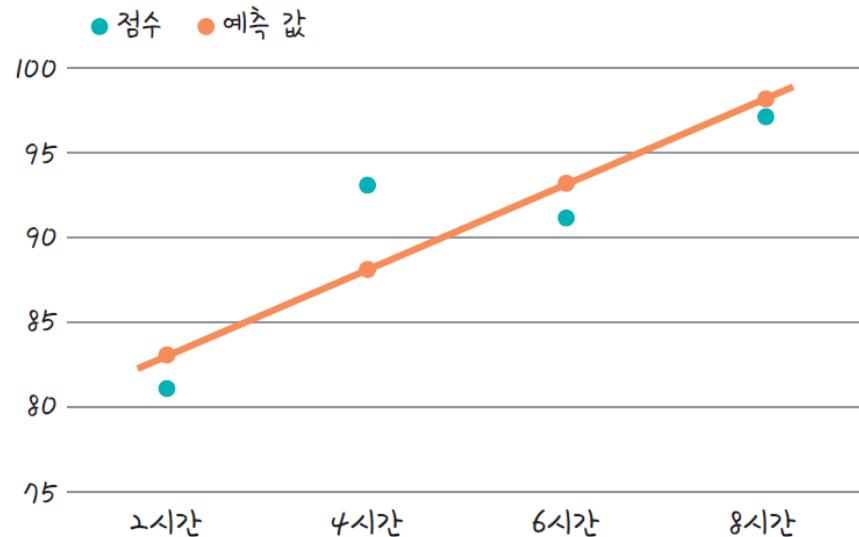


공부한 시간과 성적, 예측 값을 좌표로 표현



오차가 최저가 되는 직선의 완성

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)



오차가 최저가 되는 직선의 완성

- 이것이 바로 오차가 가장 작은, 주어진 좌표의 특성을 가장 잘 나타내는 직선임
- 우리가 원하는 예측 직선임
- 이 직선에 우리는 다른 x 값(공부한 시간)을 넣어서 '공부량에 따른 성적을 예측' 할 수 있음

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

최소 제곱법(Method of Least Squares)을 이용하여
일차 함수의 기울기 a 와 절편 b 를 코딩으로 구현해 보자.

15_(15 page) 강의용 최소 제곱법 (Method of Least Squares).ipynb

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 예측 직선과 실제 값 사이의 **차이 값**이 발생할 수 있다.
이러한 **차이 값을 조금씩 줄여가면서** 정확한 예측 직선을 찾아가야 한다.
- 이를 위해 주어진 예측선의 오차를 평가하는 **오차 평가 알고리즘**이 필요
→ 가장 많이 사용되는 방법 : **평균 제곱 오차(MSE: Mean Squared Error)**
또는 평균 제곱근 오차(RMSE: Root Mean Square Error)

단순 선형 회귀(Simple Linear Regression)

다중 선형 회귀(Multiple Linear Regression)

- **입력 x 값이 공부한 시간 하나**가 아니라 **여러 개일 경우도** 발생한다.
입력한 x 값이 여러 개인 경우는 다음 장에서 학습하고,
우선 계속해서 **공부한 시간 하나**로 **평균 제곱근 오차**에 대입해보자.

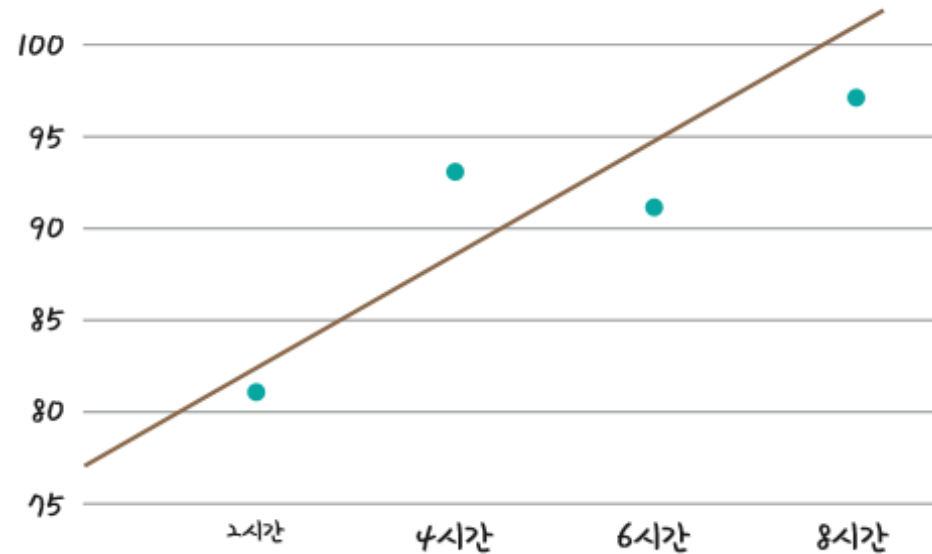
1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

‘일단 선을 그려보고, 그 다음에 조금씩 수정해 나가기’

- 가설을 하나 세운 뒤 이 값이 주어진 요건을 충족하는지를 판단하여 조금씩 변화를 주고, 이 변화가 긍정적이면 오차가 최소가 될 때까지 이 과정을 계속 반복하는 방법
- 나중에 그린 선이 먼저 그린 선보다 더 좋은지 나쁜지를 판단하기 위해 필요한 것은?
 - 각 예측선의 오차를 계산할 수 있어야 한다.
 - 이 오차가 작은 쪽으로 바꾸는 알고리즘이 필요하다.

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

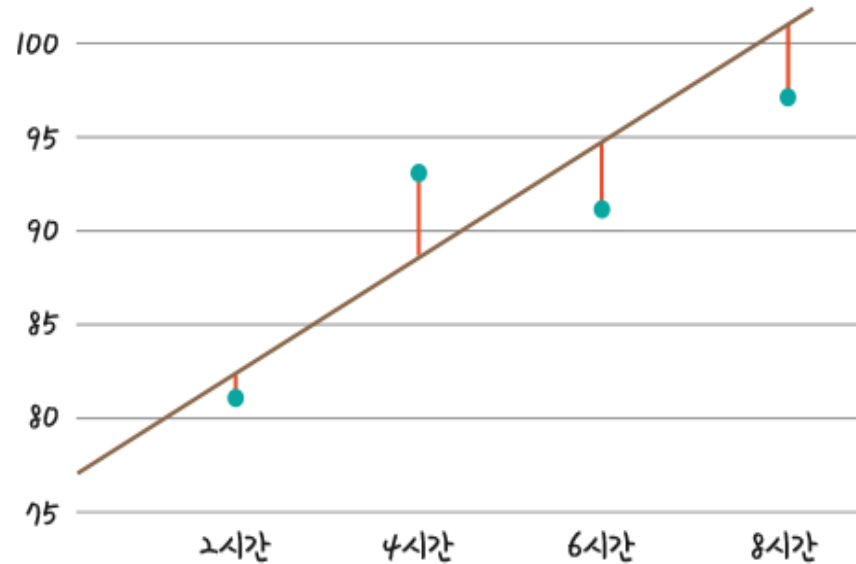
- 지금부터 오차를 계산하는 방법을 알아보자
- 가령 대강의 선을 긋기 위해 기울기 a 와 y 절편 b 를 임의의 수 3 과 76 이라고 가정해 본다면 $y = 3x + 76$ 인 선을 그려야함



임의의 직선 그려보기

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

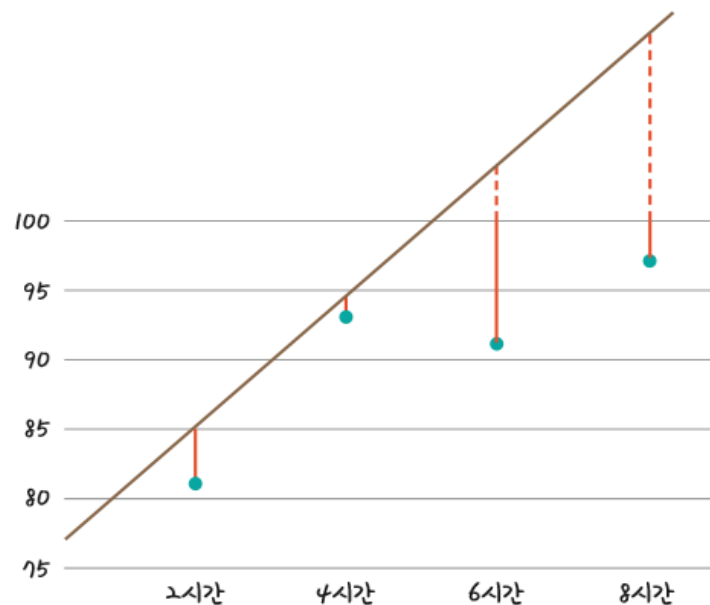
- 임의의 직선이 어느 정도의 오차가 있는지를 확인하려면 각 점과 그래프 사이의 거리를 재면 됨



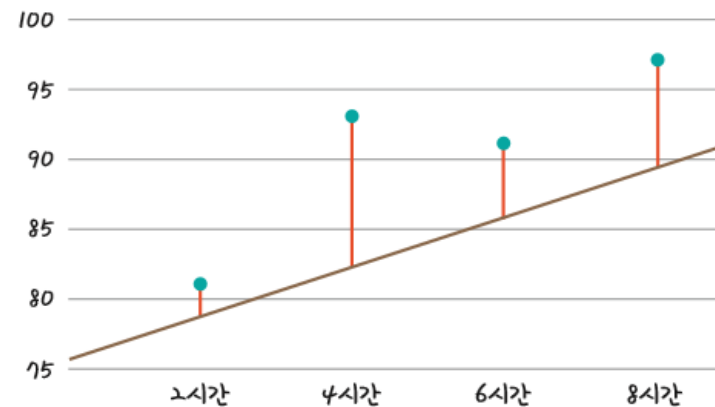
임의의 직선과 실제 값 사이의 거리

- 이 거리들의 합이 작을수록 잘 그려진 직선이고, 이 직선들의 합이 클수록 잘못 그려진 직선이 됨

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)



기울기를 너무 크게 잡았을 때의 오차



기울기를 너무 작게 잡았을 때의 오차

- 그래프의 기울기가 잘못 되었을 수록 빨간색 선의 거리의 합,
즉 오차의 합도 커짐
- ➔ 만약 기울기가 무한대로 커지면 오차도 무한대로 커지는 상관관계가 있다!

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 빨간색 선의 거리의 합을 실제로 계산해 보자

- 거리는 입력 데이터에 나와 있는 '실제 y 값'과

$y = 3x + 76$ 의 식에 대입해서 나오는 '예측 y 값'과의 차이를 통해 구할 수 있음

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

- 예를 들어, 2시간 공부했을 때의 실제 나온 점수(81점)와
그래프 $y = 3x + 76$ 식에 $x = 2$ 를 대입했을 때(82점)의 차이가 곧 오차
→ 오차를 구하는 방정식

1

$$\text{오차} = \text{실제 값} - \text{예측 값}$$

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 이 식에 주어진 데이터를 대입하여 얻을 수 있는 모든 오차의 값을 정리하면

$$\text{오차} = \text{실제 값} - \text{예측 값}$$

주어진 데이터에서 오차 구하기

공부한 시간(x)	2	4	6	8
성적(실제 값, y)	81	93	91	97
예측 값	82	88	94	100
오차	1	-5	3	3

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 부호를 없애야 정확한 오차를 구할 수 있음

따라서 오차의 합을 구할 때는 각 오차의 값을 **제곱**해 준다

$$2 \quad \text{오차의 합} = \sum_{i=1}^n (p_i - y_i)^2$$

- 여기서 i 는 x 가 나오는 순서를, n 은 x 원소의 총 개수를 의미
- p_i 는 x_i 가 대입되었을 때 직선의 방정식(여기서는 $y = 3x + 76$)이 만드는 '예측 값'
- y_i 는 x_i 에 대응하는 '실제 값'

공부한 시간(x)	2	4	6	8
성적(실제 값, y)	81	93	91	97
예측 값	82	88	94	100
오차	1	-5	3	3

- 오차의 값을 **제곱**해서 다시 오차의 **합**을 계산하면 $1 + 25 + 9 + 9 = 44$

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 오차의 합을 n 으로 나누면 오차 합의 **평균**을 구할 수 있음

→ **평균 제곱 오차(MSE: Mean Squared Error)**

3

$$\text{평균 제곱 오차(MSE)} = \frac{1}{n} \sum_{i=1}^n (p_i - y_i)^2$$

- 위 식은 앞으로 머신러닝과 딥러닝을 공부할 때 자주 등장하는 중요한 식!
- 위 식에 따라 우리가 앞서 그은 임의의 직선은 $44/4 = 11$ 의 **평균 제곱 오차**를 갖는 직선이라고 말할 수 있음
- **평균 제곱 오차(MSE)** 값은 오차의 제곱을 구하므로 실제 오차 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 **평균 제곱근 오차(RMSE)** 값을 사용할 수 있다.

5. 평균 제곱 오차 / 평균 제곱근 오차

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 여기에 다시 **제곱근**을 씌워 주면,

평균 제곱근 오차(RMSE: Root Mean Squared Error)라고 함

4

$$\text{평균 제곱근 오차(RMSE)} = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - y_i)^2}$$

- 앞서 그은 직선의 **평균 제곱근 오차**는 $\sqrt{11}$ **3.3166...**이 된다.
- **평균 제곱 오차(MSE)** 또는 **평균 제곱근 오차(RMSE)**는 오차를 계산해서 앞선 추론이 잘 되었는지 평가하는 대표적인 공식

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

- 잘못 그은 선 바로잡기는

곧 '평균 제곱근 오차(RMSE)'의 계산 결과가 가장 작은 선을 찾는 작업

- 선형 회귀

x 값이 변함에 따라 y 값도 변한다는 정의 안에서,

독립 변수 x 를 사용해 종속 변수 y 의 움직임을 예측하고 설명하는 작업을 말함

$$y = ax + b$$
 일차 함수 그래프

→ '최소 제곱법'으로 임의의 직선을 그어 이에 대한 평균 제곱근 오차(RMSE)를 구하고

→ 이 값을 가장 작게 만들어 주는 a 와 b 값을 찾아가는 작업

1. 선형 회귀
(Linear Regression)
2. 최고의 예측선 구하기
3. 최소 제곱법
(Method of Least Squares)
4. 코딩으로 확인하는
최소 제곱법 (Method of
Least Squares)
5. 평균 제곱 오차
MSE(Mean Squared Error)
/ 평균 제곱근 오차
RMSE(Root Mean Square
Error)
6. 코딩으로 확인하는
평균 제곱근 오차
RMSE(Root Mean Square
Error)

평균 제곱 오차(MSE: Mean Squared Error)와

평균 제곱근 오차(RMSE: Root Mean Squared Error) 를 코딩으로 구현해 보자.

15_(27 page) 강의용 평균 제곱근 오차 (RMSE).ipynb