Undergraduate Research Internship in Affective AI LAB.

혼자 공부하는 머신러닝 & 딥러닝

3주차 : ch04. 다양한 분류 알고리즘

컴퓨터공학과 12201682 공지윤

https://affctiv.ai/

목차

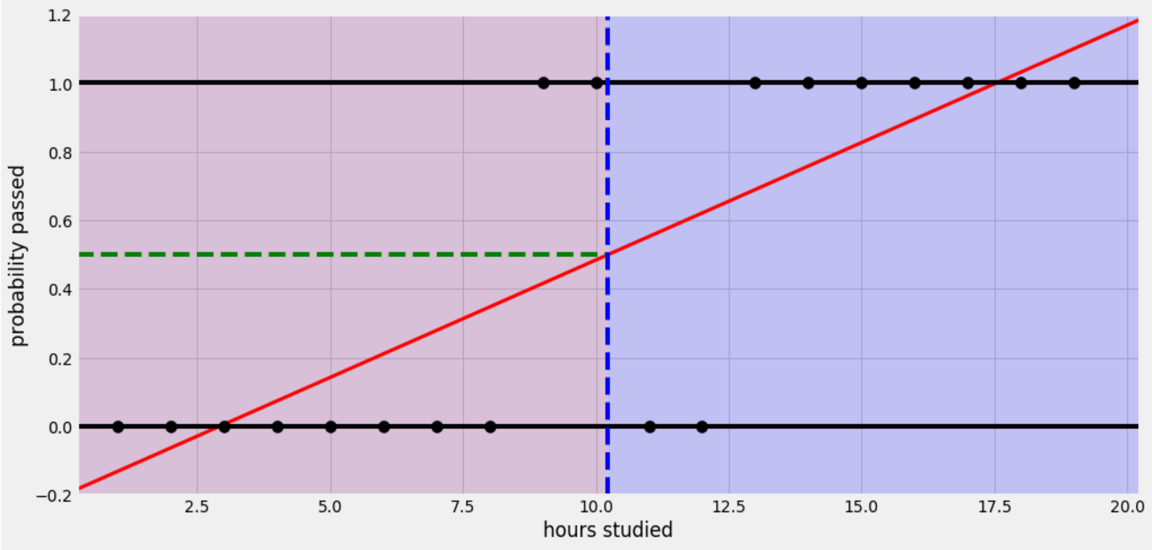
https://affctiv.ai/

로지스틱 회귀

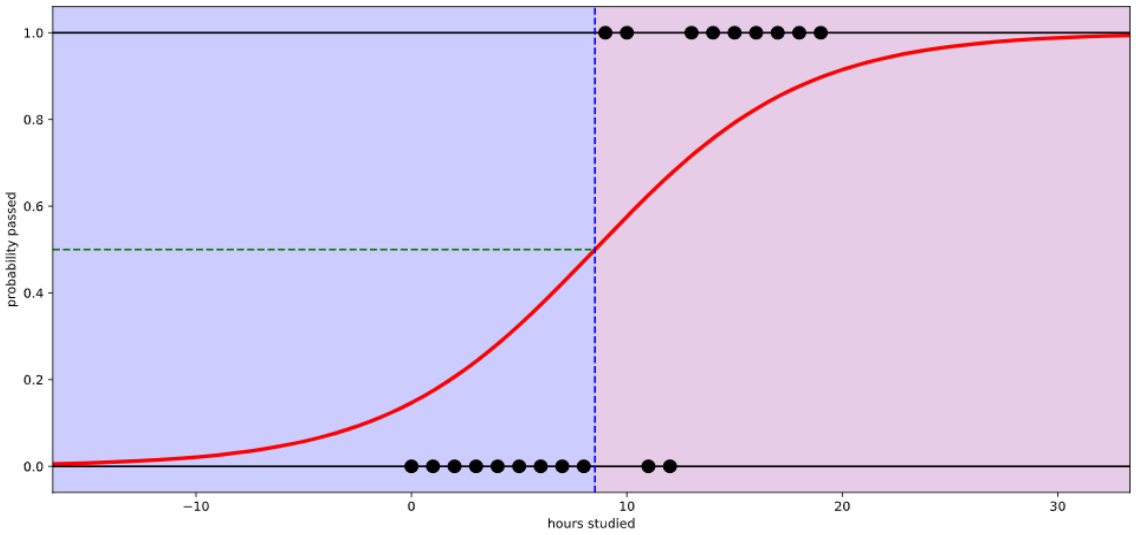
로지스틱 회귀

: 데이터가 어떤 범주에 속할 확률을 예측하고, 확률에 따라 해당 범주에 속하는 것으로 분류해주는 지도 학습 알고리즘

예) 스팸 메일 분류기 – 스팸일 확률이 0.5보다 크다면? 스팸 메일로 분류



선형 회귀 함수값 범위 : -∞ ~ ∞

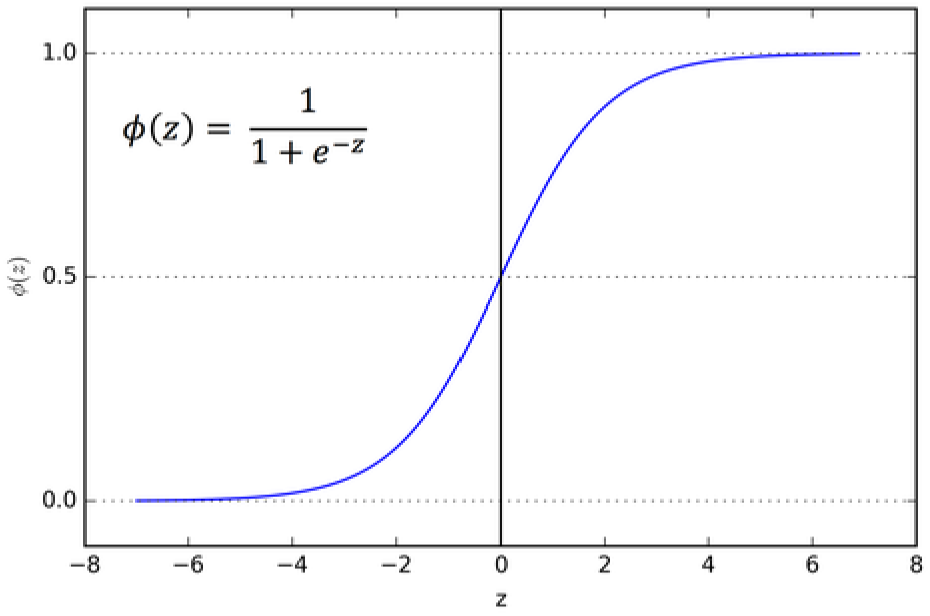


로지스틱 회귀 함수값 범위 : 0 ~ 1

로지스틱 회귀

로지스틱 회귀 예측

- 로지스틱 회귀에서 예측값은 예측 확률을 의미한다.
  - 예측값이 0.5 이상이면 1, 0.5 이하이면 0으로 예측
- 로지스틱 회귀의 예측 확률은 Sigmoid 함수의 출력값으로 계산된다.



Sigmoid 함수 (로지스틱 함수)

→ 오즈 (Odds)와 로짓 (Logit)

https://affctiv.ai/

로지스틱 회귀

오즈 (Odds)

- 실패할 확률 대비 성공할 확률 $\frac{p}{1-p}$

예) 15번 시행 중 5번 성공했을 때의 오즈 : $\frac{5}{10}$

성공 확률 (p) : $\frac{5}{15}$ / 실패 확률 (1-p) : $\frac{10}{15}$

오즈 : $\frac{\frac{5}{15}}{\frac{10}{15}} = \frac{5}{10}$

로짓 (Logit)

- 오즈에 자연로그를 씌운 것
- Logit = log + odds

$$L = \ln(\frac{p}{1-p})$$

Odds 결정 기준 : 값이 1보다 큰지 작은지

Logit 결정 기준 : 값이 0보다 큰지 작은지

https://affctiv.ai/

로지스틱 회귀

(A로 분류될 확률) = (n개의 종속변수를 가지는 선형회귀)

$$p(Y = A) = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

(범위 : 0 ~ 1) (범위 : -∞ ~ ∞)

$$\frac{P(Y=A)}{1 - P(Y=A)} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

(범위 : 0 ~ ∞) (범위 : -∞ ~ ∞)

$$\log\left(\frac{P(Y=A)}{1 - P(Y=A)}\right) = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

(범위 : -∞ ~ ∞) (범위 : -∞ ~ ∞)

https://affctiv.ai/

로지스틱 회귀

$$\log\left(\frac{P(Y=A)}{1-P(Y=A)}\right) = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

구하고자 하는 것 : P(Y = A)

↓

$$P(Y = A) = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n)}}$$

→ logit 함수는 Sigmoid와 역함수 관계

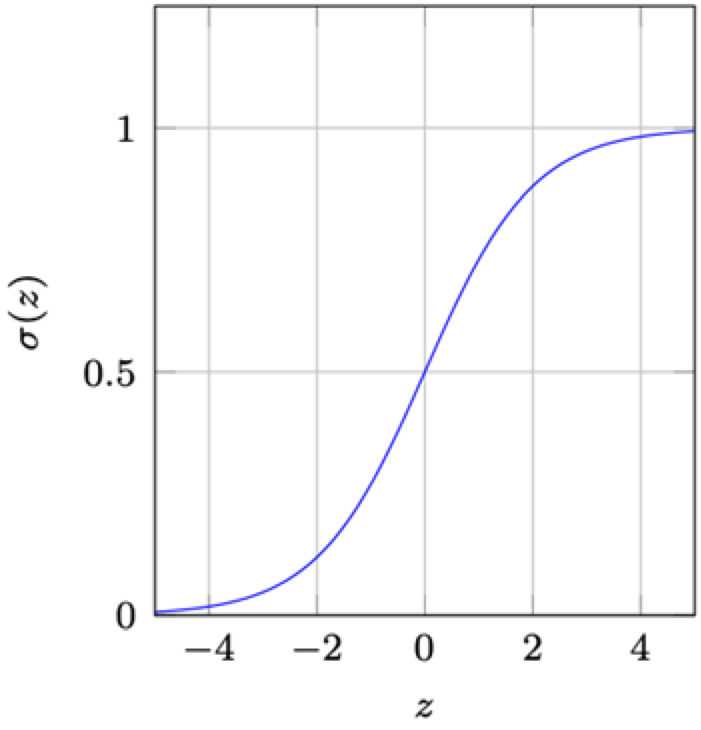
In [19]:
```python
lr.decision_function(train_bream_smelt[:5]) # z값
```

Out[19]:
```
array([-6.02927744,  3.57123907, -5.26568906, -4.24321775, -6.0607117 ])
```

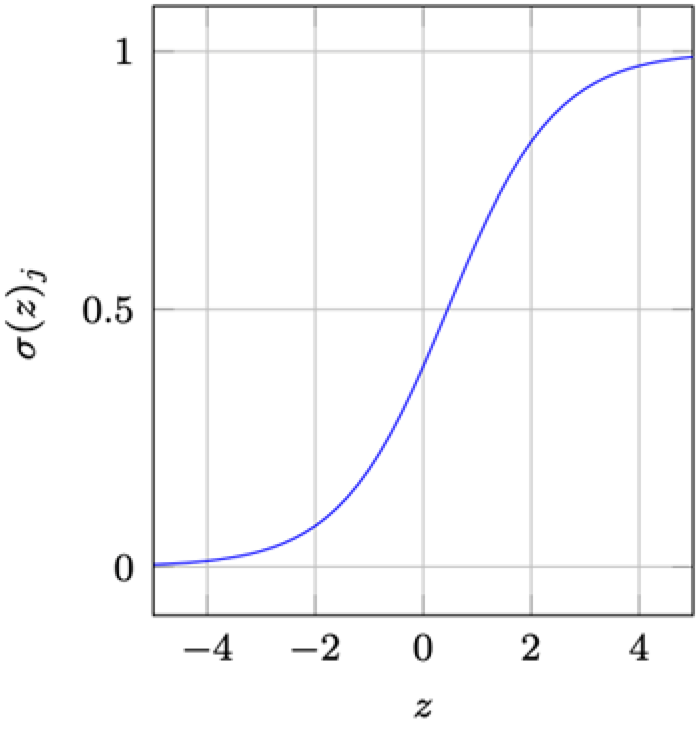
In [20]:
```python
# z값을 sigmoid 함수에 집어넣기!
from scipy.special import expit
print(expit(lr.decision_function(train_bream_smelt[:5]))) # 양성 클래스에 대한 z값
```

```
[0.00240145 0.97264817 0.00513928 0.01415798 0.00232731]
```

다중 로지스틱 회귀



(a) Sigmoid activation function.



(b) Softmax activation function.

- 이진 분류 : Sigmoid 함수를 사용해 0과 1 사이의 값으로 변형
- 다중 분류 : Softmax 함수를 사용해 0과 1 사이의 값으로 변형

https://affctiv.ai/

다중 로지스틱 회귀

Softmax 함수

: 다중 클래스 분류 모델을 만들 때 사용한다.

- 지수함수를 사용하는 이유 : 미분이 가능하도록 하며, 입력값 중 큰 값은 더 크게, 작은 값은 더 작게 만들어서 입력 데이터의 구분이 더 잘되도록 한다.

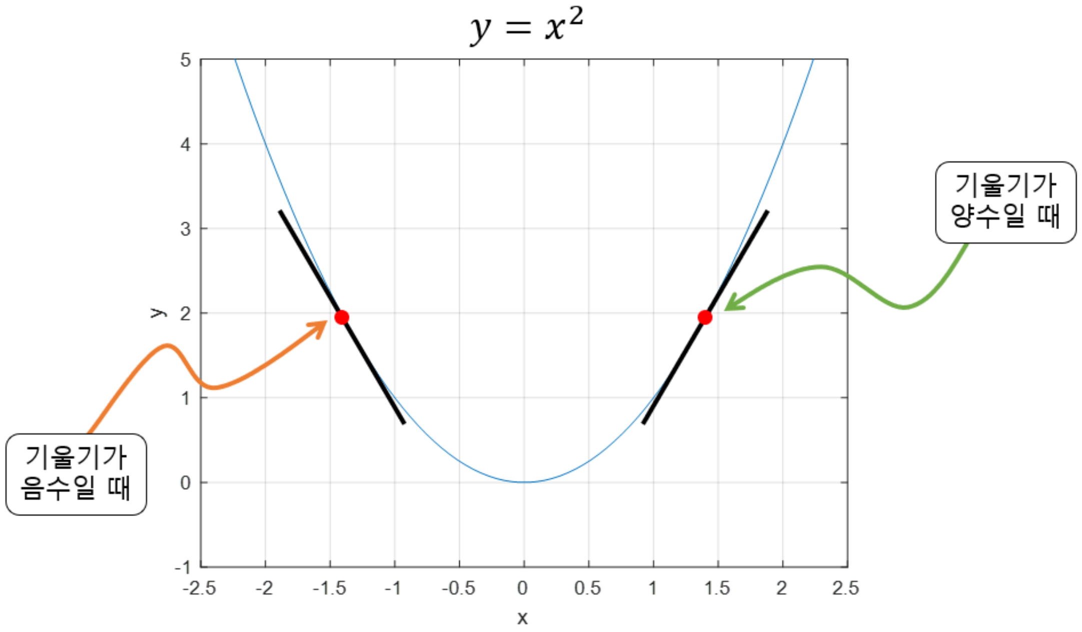
$$p_i = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}} \quad (i = 1, 2, 3, \ldots, k)$$

- Softmax 함수에서 K=2로 두면 (2개의 class를 대상으로 정의) sigmoid 함수로 환원이 되고, 반대로 sigmoid 함수를 K개의 class로 일반화하면 softmax 함수가 유도된다.

경사하강법

경사하강법 (Gradient Descent)

: 비용함수를 최소화하는 가중치를 구하는 알고리즘으로, 기울기가 감소하는 방향으로 점진적인 갱신을 한다.



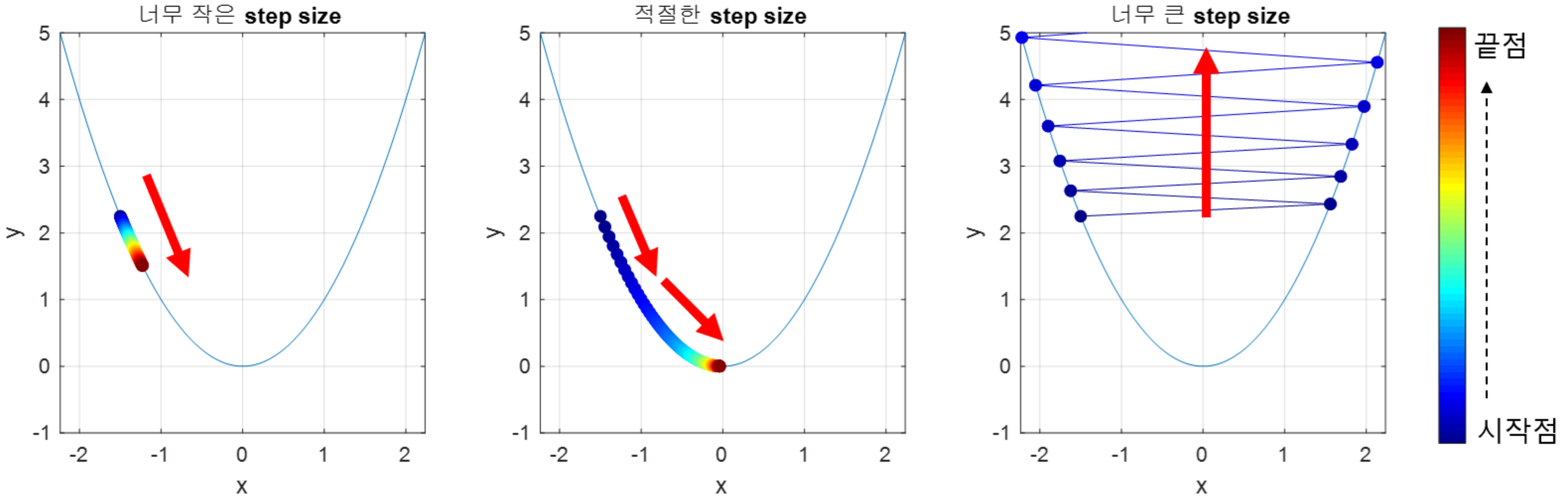
$$x_{i+1} = x_i - (learning\_rate) \times (gradient)$$

경사하강법

Learning rate (학습률)

: 하이퍼파라미터로, 너무 작지도, 너무 크지도 않은 값으로 설정한다.

- 너무 큰 경우 : 가중치 값이 반대쪽을 오가면서 매우 큰 거리를 이동하게 되고, 최솟값에서 점점 멀어지게 된다.
- 너무 작은 경우 : 최솟값에 도달하기 위해 많은 연산이 요구된다.
- 속도에만 영향을 주는 것이 아니다! 학습률 없이 미분 값으로만 갱신 시 최솟값에 도달하지 않고 제자리에서 진동하게 될 수도 있다.



https://affctiv.ai/

경사하강법

확률적 경사하강법

: 추출된 데이터 한 개에 대해서 gradient를 계산하고 알고리즘을 적용한다. 전체 데이터를 사용하는 것이 아니라, 랜덤하게 추출한 일부 데이터를 사용하는 것이다.

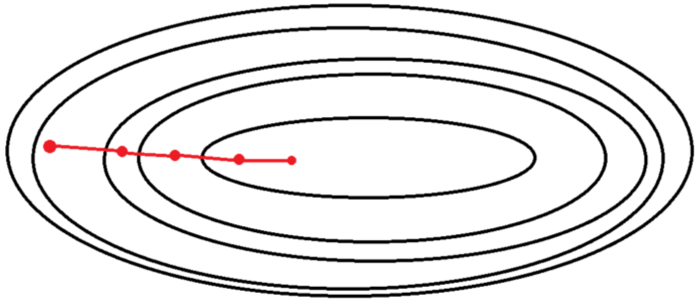
배치 경사하강법

: 경사 하강법의 손실 함수의 기울기 계산에 전체 학습 데이터셋에 대한 에러를 구한 뒤 기울기를 한 번만 계산하여 모델의 파라미터를 업데이트하는 방식이다.

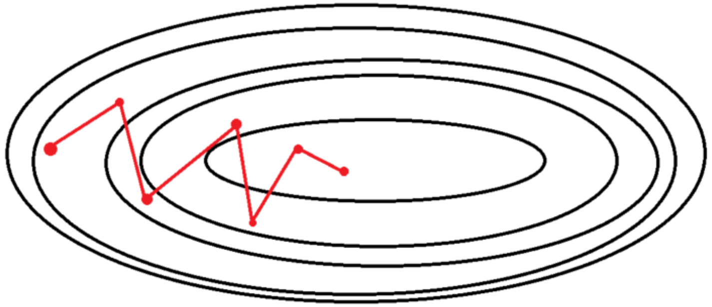
미니배치 경사하강법

: SGD와 BGD의 절충안으로 배치 크기를 줄여 확률적 경사 하강법을 이용하는 방법이다. 전체 데이터를 batch_size개씩 나눠 배치로 학습 시키며, 배치 크기는 사용자가 지정한다.

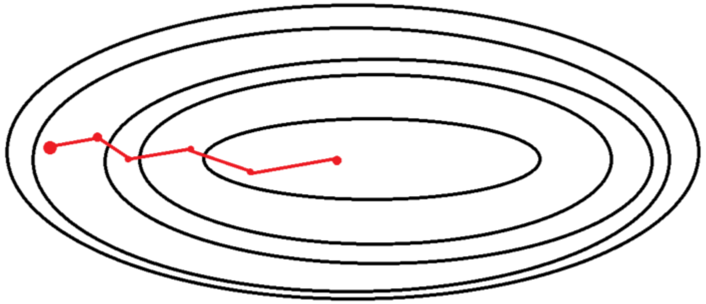
경사하강법



배치 경사하강법(Batch gradient descent)



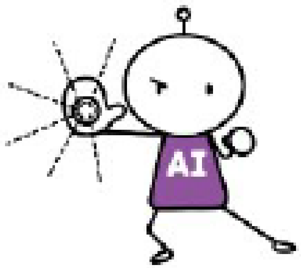
확률적 경사 하강법(SGD)



미니배치 경사 하강법

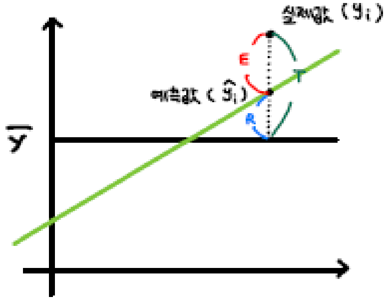
Undergraduate Research Internship in Affective AI LAB.

감사합니다 :>



컴퓨터공학과 12201682 공지윤

별첨) SST = SSR + SSE 식이 성립하는 이유 증명



$E = y_i - \hat{y}_i$ (실제값과 예측값의 차이) 회귀식으로 설명 불가능 (오차)

$R = \hat{y}_i - \bar{y}$ (예측값과 평균의 차이) 회귀식으로 설명 가능
(x 값에 따라 y값이 기울기 따라 얼마만큼 변화가 가능한지 예측 가능)

$T = y_i - \bar{y}$ (실제값과 평균의 차이)

$$T = R + E$$
$$(y_i - \bar{y}) = (\hat{y}_i - \bar{y}) + (y_i - \hat{y}_i)$$

오차 제곱 → 오차의 합이 0 이 되기 때문에 제곱 처리해주기!

SST = SSR + SSE (단, $\hat{y}_i$ 이 최소제곱법으로 구한 선형회귀모델의 y_i 의 예측값일 때)

Proof) SST - SSR - SSE

$$= \Sigma (y_i - \bar{y})^2 - \Sigma (y_i - \hat{y}_i)^2 - \Sigma (\hat{y}_i - \bar{y})^2$$
$$= \Sigma (y_i - \bar{y})^2 - (y_i - \hat{y}_i)^2 - (\hat{y}_i - \bar{y})^2$$
$$= \Sigma \{ (\cancel{y_i^2} + \cancel{\bar{y}^2} - 2y_i\bar{y}) - (\cancel{y_i^2} - 2y_i\hat{y}_i + \hat{y}_i^2) - (\hat{y}_i^2 - 2\hat{y}_i\bar{y} + \cancel{\bar{y}^2}) \}$$
$$= \Sigma (-2\hat{y}_i^2 - 2y_i\bar{y} + 2y_i\hat{y}_i + 2\hat{y}_i\bar{y}) \xrightarrow[\div(-2)]{} \Sigma (\hat{y}_i^2 + y_i\bar{y} - y_i\hat{y}_i - \hat{y}_i\bar{y})$$
$$= \Sigma (\hat{y}_i^2 - y_i\hat{y}_i + y_i\bar{y} - \hat{y}_i\bar{y})$$
$$= \Sigma \{ \hat{y}_i(\hat{y}_i - y_i) + \bar{y}(y_i - \hat{y}_i) \}$$
$$= \Sigma \hat{y}_i(\hat{y}_i - y_i) + \Sigma \bar{y}(y_i - \hat{y}_i)$$
$$= \underline{\Sigma \hat{y}_i(\hat{y}_i - y_i)} + \underline{\bar{y}\,\Sigma(y_i - \hat{y}_i)}$$

↳ $\frac{\Sigma y_i}{n} = \frac{\Sigma \hat{y}_i}{n}$
(실제값의 평균 = 예측값의 평균)

∴ 0

$\hat{y}_i - y_i = e_i$

$\Sigma \hat{y}_i e_i$

.... $\hat{y}_i = \beta_0 + \beta_1 x_i$

$$\Sigma \hat{y}_i e_i = \Sigma(\beta_0 + \beta_1 x_i)e_i$$
$$= \Sigma(\beta_0 e_i + \beta_1 x_i e_i)$$
$$= \cancel{\beta_0 \Sigma e_i}^{0} + \beta_1 \Sigma x_i e_i$$
$$= \beta_1 \Sigma x_i e_i = \beta_1 \Sigma x_i (\hat{y}_i - y_i)$$
$$= \beta_1 \Sigma x_i (\beta_0 + \beta_1 x_i - y_i)$$
$$= \beta_1 \Sigma \beta_0 x_i + \beta_1 x_i^2 - x_i y_i$$
$$= -\beta_1 \Sigma (x_i y_i - \beta_0 x_i - \beta_1 x_i^2)$$

∴ 0

최소제곱법 → $\Sigma (y_i - \hat{y}_i)^2$ 이 최소가 되는 β_0 과 β_1 찾기

$$\frac{d}{d\beta_0} \Sigma (y_i - (\beta_1 x_i + \beta_0))^2$$
$$= \frac{d}{d\beta_0} \Sigma (y_i^2 - 2y_i(\beta_1 x_i + \beta_0) + (\beta_1 x_i + \beta_0)^2)$$
$$= \frac{d}{d\beta_0} \Sigma (y_i^2 - 2y_i\beta_1 x_i - 2\beta_0 y_i + (\beta_1 x_i + \beta_0)^2)$$
$$= \Sigma(-2y_i + 2(\beta_1 x_i + \beta_0)) = 0$$

$$\therefore \boxed{\Sigma y_i = \Sigma \beta_1 x_i + \beta_0 = \Sigma \hat{y}_i}$$

$$\frac{d}{d\beta_1} \Sigma (y_i - (\beta_1 x_i + \beta_0))^2$$
$$= \frac{d}{d\beta_1} \Sigma (y_i^2 - 2y_i\beta_1 x_i - 2\beta_0 y_i + (\beta_1 x_i + \beta_0)^2)$$
$$= \Sigma \{ -2y_i x_i + 2(\beta_1 x_i + \beta_0)x_i \}$$
$$= -2\Sigma \{ x_i y_i - (\beta_1 x_i + \beta_0)x_i \}$$
$$= -2\Sigma (x_i y_i - \beta_1 x_i^2 - \beta_0 x_i) = 0$$

$$\boxed{\Sigma (x_i y_i - \beta_1 x_i^2 - \beta_0 x_i) = 0}$$