Machine Learning

10장 Regression Analysis

고려대학교 통계학과
박유성

Contents

01 선형회귀 모형

- 분류와 더불어 지도학습의 중요부분.
- 클래스(이산형) y를 예측 → 분류
- 연속형 y를 예측 → 회귀
- 해석이 용이하고 모수 추정이 쉬워 가장 널리 사용됨.

- 고전적 선형 회귀 모형.

$$y_i = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_d x_{d,i} + \varepsilon_i \qquad (10.1)$$

이때 $i = 1, 2, \cdots, n$: 관측치의 개수 $x_1, \cdots, x_d$: d개의 특성변수.

$\varepsilon_i \sim iid\,(0, \sigma^2)$

01 선형회귀 모형-선형성의 특징

- y와 x와의 관계가 선형 또는 비선형 상관없음.
- Y와 β 와의 관계가 선형일때 → 선형모형.
- 예 $x_1,\ x_2$ 두 개의 특성변수가 있을 때

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{1,i}^2 + \beta_3 x_{1,i} x_{2,i} + \beta_4 e^{x_{2,i}} + \varepsilon_i \qquad (10.2)$$

이때, $x_{1,i}^2 = x_{2,i}^*$, $x_{1,i}x_{2,i} = x_{3,i}^*$, $e^{x_{2,i}} = x_{4,i}^*$ 로 놓으면 선형모형으로 변환가능.

- 특성변수가 제곱의 형태 또는 지수의 형태 등 관계없음.
- Y와 특성 변수 각각에 대한 Scatter plot을 그린 후 특성변수의 변환.

01 선형회귀 모형-모형의 추정

- $$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_d x_{d,i} + \varepsilon_i \quad , i = 1,2,\cdots,n \qquad (10.3)$$
  $$= \boldsymbol{\beta}^T \boldsymbol{x}_i + \varepsilon_i$$

  이때, $\boldsymbol{x}_i = (1, x_{1,i}, \cdots, x_{d,i})^T$이고 $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_d)^T$

- 오차항의 y_i에 대한 영향력을 최소화
  $$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2 \qquad (10.4)$$

- 식(10.4)를 최소화 하는 β를 최소제곱추정치 또는 OLS라고 함.

- 머신러닝에서는 β 의 추정치를 기울기 하강법(Ch.2.3)을 이용하여 추정.

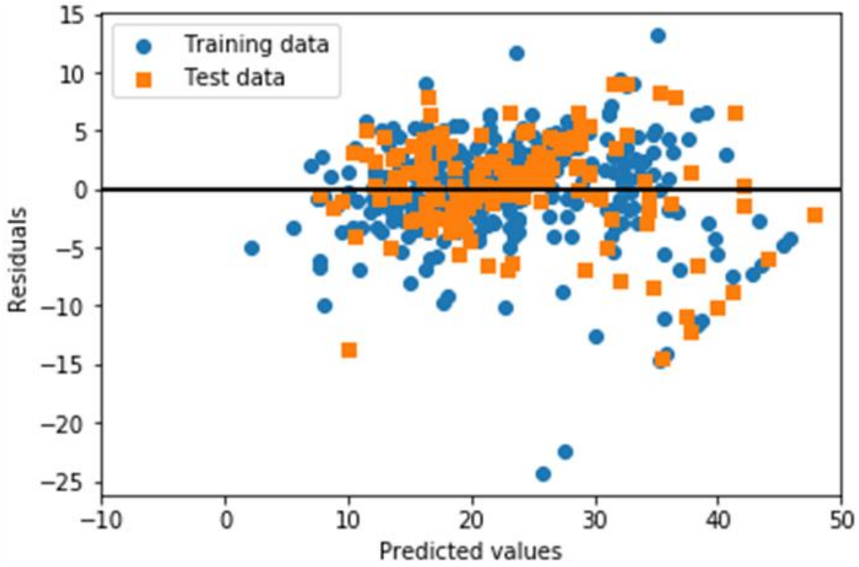
- 직접 β를 추정하면,
  $$\hat{\beta} = (X^T X)^{-1} X^T \boldsymbol{y}$$

- 특성 변수들이 선형적으로 독립이어야함. 즉 특성변수간 상관관계=0

- 특성 변수들 간의 상관 관계를 사전에 점검 하여야 함.

01 선형회귀 모형-모형의 진단

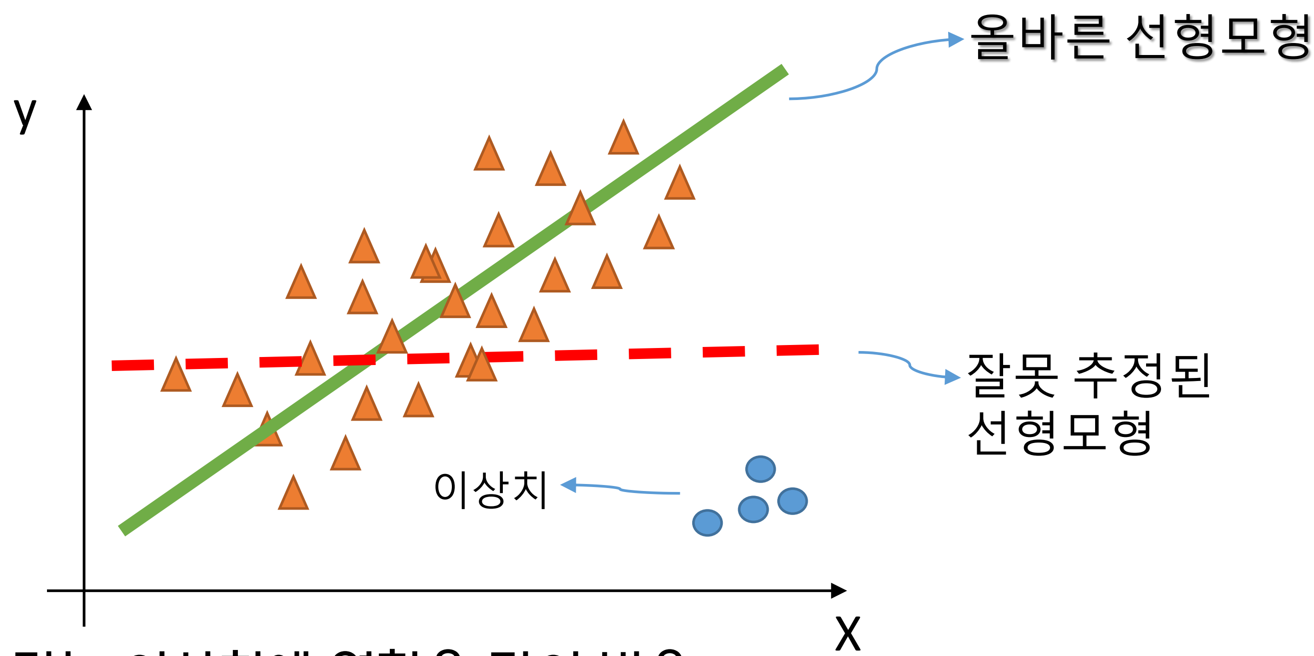
- 모형의 진단은 잔차 plot를 그려 보는 것으로 부터 출발.
- Y의 예측치 : $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \cdots + \hat{\beta}_d x_{d,i}$ 이고, 잔차 : $r_i = y_i - \hat{y}_i$ 일때,
- X축→ $\hat{y}_i$, y축 → r_i 인 plot



- r_i 는 식(10.1)의 ε_i 의 추정치 이므로, 모형이 잘 추정됨→ $r_i \sim iid\,(0, \sigma^2)$ 만족.
- r_i 가 추세 또는 특수한 형태를 가지면 모형을 재추정 해야함.

02 로버스트회귀(Robust regression model)

- 선형회귀모형은 이상치의 영향을 크게 받음. (∵ 오차항의 제곱합 최소화)



- 큰 오차를 갖는 이상치에 영향을 많이 받음.
- 따라서, 로버스트 회귀 방법을 사용해야함.
- 통계학 에서의 로버스트 회귀 : M-estimator, LTS, DPM 등.
- 머신러닝 에서의 로버스트 회귀 : RANSAC(random sample concensus)

02 로버스트회귀-RANSAC

- 매우 직관적이지만 이상치에 매우 로버스트함.

- RANSAC의 절차.

1. 학습데이터에서 작은 크기의 임의 표본을 뽑음.(특성변수 총수 + 20~30개의 표본)
2. 표본에 OLS추정치를 구함. 이후
3. 추정된 모형에 전체 학습데이터를 적용하여 잔차를 구함.
4. 잔차의 중위수(median)을 구한 후 각 잔차의 MAD ($|r_i - median(r_i)|$)를 구함.
5. MAD가 d (보통 4~5)보다 작은 관측치만 모음.→콘센서스셋(concensus set)
6. 콘센서스셋에 OLS를 재적합.
7. step1~step6를 M번 반복한다.

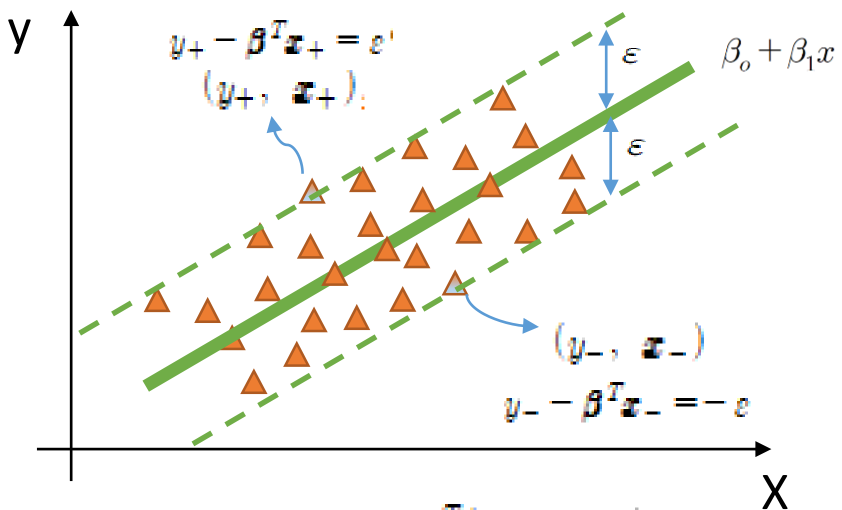
- 가장 큰 그기의 콘센서스셋으로 계산된 OLS 추정치→ RANSAC

- MAD대신 절대값 등으로 대체 가능.

- 초모수 : 반복수 M, 임의의 표본수, 임계값 d (일종의 SVM으로 볼수 있음.)

03 SVM회귀 & 커널 SVM회귀 1

- SVM회귀의 목표 : 자료(y_i, x_i), $i = 1, 2, \cdots, n$가 $|y_i - \beta_0 - \beta_1 x_i| \le \varepsilon_i$인 β_0와 β_1 찾기.



- 이때 두 직선 사이 거리 : $y_+ - y_- + \boldsymbol{\beta}^T(\boldsymbol{x}_- - \boldsymbol{x}_+) = 2\varepsilon$

- 양변을 $\|\boldsymbol{\beta}\| = \sqrt{\sum_{i=0}^{d} \beta_i^2}$ 으로 나누면($\boldsymbol{\beta}^T$의 유일해)

$$\frac{(y_+ - y_-) + \boldsymbol{\beta}^T(\boldsymbol{x}_- - \boldsymbol{x}_+)}{\|\boldsymbol{\beta}\|} = \frac{2\varepsilon}{\|\boldsymbol{\beta}\|} \qquad (10.5)$$

- 식(10.5)의 표준화 거리를 최대화→ $(y_+, \boldsymbol{x}_+)$와 $(y_-, \boldsymbol{x}_-)$ 사이의 거리를 최대화 하는 선형평면.→ 즉 . $(y_+, \boldsymbol{x}_+)$와 $(y_-, \boldsymbol{x}_-)$ 가 서포트 벡터가 됨.

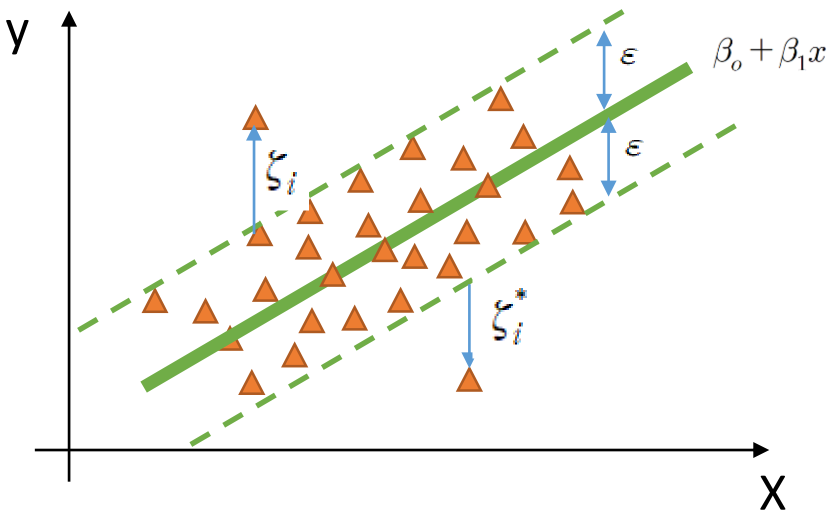
03 SVM회귀 & 커널 SVM회귀 2

- 표준화 거리의 최대화= $\|\boldsymbol{\beta}\|$ 의 최소화= $\|\boldsymbol{\beta}\|^2$ 의 최소화.
- 즉, $i=1,2,\cdots,n$에 대해 $|y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}| \le \varepsilon$ 을 만족하는 $\|\boldsymbol{\beta}\|^2$ 을 최소화 문제.
- SVM 분류와 개념적으로 동일.

- 그러나, 실제 문제에서 ε 이 매우 크지 않는 한 $|y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}| \le \varepsilon$ 만족 어려움.
- SVM에서와 같이 완화변수 (slack variable: ζ_i와 ζ_i^*)도입 필요.

03 SVM회귀 & 커널 SVM회귀 3

- 완화 변수(slack variable)



$$y_i - \boldsymbol{x}_i^T\boldsymbol{\beta} \le \varepsilon + \zeta_i$$

$$\boldsymbol{x}_i^T\boldsymbol{\beta} - y_i \le \varepsilon + \zeta_i, \quad \text{이때 } \zeta_i, \zeta_i^* \ge 0$$

- SVM 회귀의 최적화

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \,\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}(\zeta_i + \zeta_i^*)$$

$$y_i - \boldsymbol{x}_i^T\boldsymbol{\beta} \le \varepsilon + \zeta_i \text{와 } \boldsymbol{x}_i^T\boldsymbol{\beta} - y_i \le \varepsilon + \zeta_i \qquad (10.6)$$

- 즉, ε-insensitive 손실함수

$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} L_\varepsilon(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|^2$ 과 동일. Where $L_\varepsilon(r) = \begin{cases} 0 \text{ if } |r| < \varepsilon \\ |r| - \varepsilon \quad 0/w \end{cases}$ $\quad \lambda = \frac{1}{C}$

- 즉, 튜브안($|y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}| < \varepsilon$)이면 0을 부여, 튜브 밖의 관측치에만 손실을 부여.

03 SVM회귀 & 커널 SVM회귀 4

- 라그랑지 승수 α_i α_i^*, η_i 와 η_i^* 를 이용하여 손실함수 정의.

$$L_p = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}(\zeta_i^* + \zeta_i) + \sum_{i=1}^{n}\alpha_i(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \varepsilon - \zeta_i)$$
$$+ \sum_{i=1}^{n}\alpha_i^*(\boldsymbol{x}_i^T\boldsymbol{\beta} - y_i - \varepsilon - \zeta_i^*) - \sum_{i=1}^{n}(\eta_i\zeta_i + \eta_i^*\zeta_i^*) \qquad (10.7)$$

- L_p를 최소화 하는 β, ζ_i와 ζ_i^*대해 미분.

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i = \boldsymbol{\beta},\ \ \alpha_i = C - \eta_i,\ \ \alpha_i^* = C - \eta_i^*$$ 이므로,

- 이를 식(10.7)에 대입.

$$L_p = \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{n}y_i(\alpha_i - \alpha_i^*) + \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\boldsymbol{x}_i^T\boldsymbol{x}_j \qquad (10.8)$$

- L_p최소화하는 α_i와 α_i^* 는 아래와 같은 조건에서 구함.

$$\alpha_i(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \varepsilon - \zeta_i) = 0,\ \ \alpha_i^*(\boldsymbol{x}_i^T\boldsymbol{\beta} - y_i - \varepsilon - \zeta_i^*) = 0,\ \ (C - \alpha_i)\zeta_i = 0,\ \ (C - \alpha_i^*)\zeta_i^* = 0$$

- 이에 대한 해는 convex quadratic 프로그램 문제. [계속]

03 SVM회귀 & 커널 SVM회귀 5

- ε -튜브안의 모든 관측치는 $\zeta_i = \zeta_i^* = 0$ 이고 $|y_i - \boldsymbol{x}_i^T\beta| < \varepsilon$ 임.
- 따라서 조건 $\alpha_i(y_i - \boldsymbol{x}_i^T\beta - \varepsilon - \zeta_i) = 0$ 와 $\alpha_i^*(\boldsymbol{x}_i^T\beta - y_i - \varepsilon - \zeta_i^*) = 0$ 에 의해 ε - 튜브안의 모든 관측치에 대한 $\alpha_i = \alpha_i^* = 0$ 됨.
- 이는 $\beta = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i$ 이므로 , ε -튜브안의 관측치는 β 에 기여하지 못함.
- 즉 ε -튜브 경계선 및 밖의 관측치만 β 를 구하는데 기여함.(즉, 써포트 벡터)
- 새로운 특성변수 $\boldsymbol{x}$ 에 대한 y 예측치 Where : $y_i = \boldsymbol{x}_i^T\beta$ 이고 $\hat{\beta} = \sum_{i=1}^{n}(\hat{\alpha}_i - \hat{\alpha}_i^*)\boldsymbol{x}_i$

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}(\hat{\alpha}_i - \hat{\alpha}_i^*)\boldsymbol{x}_i^T\boldsymbol{x} \tag{10.9}$$

- 비선형 커널 SVM회귀 : $\boldsymbol{x}_i^T\boldsymbol{x}_i$ 대신에 적절한 kernel 함수 $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 를 대입.
- α_i 와 α_i^* 를 구한 후 비선형 커널 SVM회귀의 예측치는.

$$\hat{y}_i = \sum_{i=1}^{n}(\hat{\alpha}_i - \hat{\alpha}_i^*)K(\boldsymbol{x}_i, \boldsymbol{x})$$

04 규제화된 선형회귀모형과 기타 회귀모형 1

- 과적합(over fitting)의 문제 고려 하여야함.
- 리지(Ridge) 회귀.
- L_2 규제화를 살펴보면

$$\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{d}\beta_i^2 \qquad (10.10)$$

- 이때 L_2규제화 에서의 β 임. 식(10.10)을 행렬식으로 표현하면,

$$L(\beta) = (\boldsymbol{y} - X\boldsymbol{\beta})^T(\boldsymbol{y} - X\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}$$ 을 미분하면,

$$\frac{\partial L(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} = -2X^T\boldsymbol{y} + 2X^TX\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta} = 0$$ 을 풀면, $$\hat{\boldsymbol{\beta}}_{Ridge} = (X^TX + \lambda I)^{-1}X^T\boldsymbol{y}$$

- 통계학 : X^TX 의 역행렬이 존재하지 않을 때 적용.
- 해석의 문제가 발생하여 거의 쓰지 않는 방법
- L_2 규제화로 해석이 가능해짐. → 회귀모형의 과대적합 해결방안으로 유용.

04 규제화된 선형회귀모형과 기타 회귀모형 2

- **LASSO**(Least absolute shrinkage and selection operator)
- L_1 규제화를 살펴보면,

  $\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{d}|\beta_j|$ 를 최소화 하는 $\boldsymbol{\beta}$ 를 구함.
- 이때, λ 가 클수록 설명력이 작은 순서대로 β_j→0 수렴.
- 즉 ,특성변수를 선택하는 모형임.

- **Elastic Net** : L_1 규제화와 L_2 규제화를 결합한 선형회귀모형.

  $\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \lambda_1\sum_{j=1}^{d}|\beta_j| + \lambda_2\sum_{j=1}^{d}\beta_j^2$ 를 최소화 하는 $\boldsymbol{\beta}$ 를 구함.

- 기타 회귀모형: KNN, 의사결정나무(회귀나무), Boosting을 이용한 회귀.

04 규제화된 선형회귀모형과 기타 회귀모형 3

- Scikit learn을 이용한 Ridge, LASSO, Elastic Net.

1.리지(Ridge)회귀.

```
from sklearn,linear_model import Ridge
rd=Ridge(alpha=1.0)
```

alpha는 L_2 규제의 λ

2.LASSO

```
from sklearn.linear_model import Lasso
lss=Lasso(alpha=1.0)
```

3. Elastic Net.

```
from sklearn.linear_model import ElasticNet
elt=ElasticNet(alpha=1.0 l1_ratio=0.5)
```

alpha는 L_1과 L_2 규제하의 $\lambda_1 + \lambda_2$

$l1$_ratio는 alpha$\times l1$_ratio로 L_1 규제하의 λ_1

- 즉, $l1\_ratio = 1$ → LASSO & $l1\_ratio = 0$ → Ridge

04 규제화된 선형회귀모형과 기타 회귀모형 2

- **LASSO**(Least absolute shrinkage and selection operator)
- L_1 규제화를 살펴보면,

  $\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{d}|\beta_j|$ 를 최소화 하는 $\boldsymbol{\beta}$ 를 구함.
- 이때, λ 가 클수록 설명력이 작은 순서대로 β_j→0 수렴.
- 즉 ,특성변수를 선택하는 모형임.

- **Elastic Net** : L_1 규제화와 L_2 규제화를 결합한 선형회귀모형.

  $\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \lambda_1\sum_{j=1}^{d}|\beta_j| + \lambda_2\sum_{j=1}^{d}\beta_j^2$ 를 최소화 하는 $\boldsymbol{\beta}$ 를 구함.

- 기타 회귀모형: KNN, 의사결정나무(회귀나무), Boosting을 이용한 회귀.

Q & A