2주차 강의 – Multiple features

박소정

TNT - ML팀

TRAIN AND TEST

Contents

Multiple features

다중 선형회귀

Vectorization

벡터화

실습

TRAIN AND TEST

Multiple features

-다중 선형회귀 -

Contents

2주차: 여러 입력 변수를 사용한 회귀 분석

동영상 49 분 남음　　3개의 평가가 남음

이번 주에는 선형 회귀를 확장하여 여러 입력 특징을 처리하는 방법을 알아봅니다. 또한 벡터화, 특징 스케일링, 특징 엔지니어링, 다항식 회귀 등 모델의 훈련과 성능을 개선하기 위한 몇 가지 방법도 배웁니다. 마지막 주에는 코드에서 선형 회귀를 구현하는 연습을 하게 됩니다.

학습 목표 표시

TRAIN AND TEST

다중 선형 회귀 - multiple features (variables)

- 여러 입력 변수를 사용한 회귀 분석
- ex) 주택 가격 예측 > 침대가 하나 추가 될때마다 4만달러

Model:

Previously: $f_{w,b}(x) = wx + b$

$$f_{w,b}(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

example

$$f_{w,b}(x) = 0.1 x_1 + 4 x_2 + 10 x_3 + -2 x_4 + 80$$

x_1: size, x_2: #bedrooms, x_3: #floors, x_4: years, 80: base price

TRAIN AND TEST

단순(일차) 선형회귀 vs 다중 선형 회귀 vs 다항식 회귀

Simple + 직선 ☞ 원인 1개
(Simple Linear Regression)

$y = ax + b$

- y: 종속변수 (Target, 결과)
- a: 가중치
- x: 독립변수 (Feature, 원인)
- b: y절편

Multiple + 직선 ☞ 원인 n개
(Multiple Linear Regression)

$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n$

- y: 종속변수 (Target)
- a_0: y절편
- $a_1, a_2, a_3, \ldots, a_n$: 가중치
- $x_1, x_2, x_3, \ldots, x_n$: 독립변수 (Feature, 원인 n개)

비선형??? 원인변수의 차수가 1차가 아닌 항이 있으면...!!

<비선형!?>

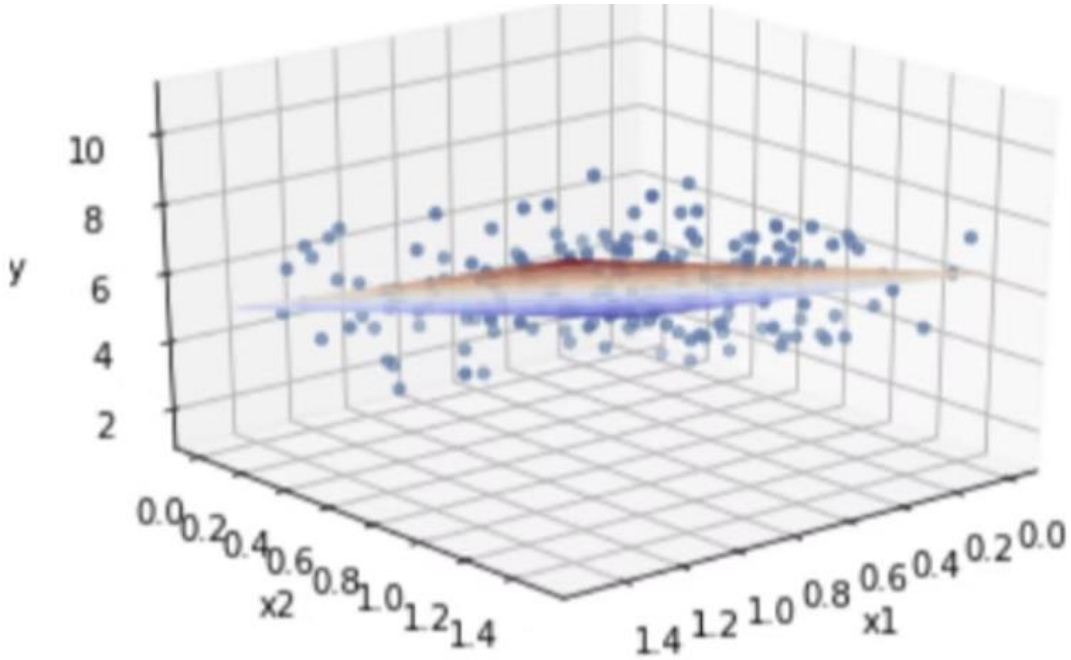
$y = a_0 + a_1x_1^2 + a_2a_2x_2^3 + a_3x_3 + \cdots + a_nx_n$

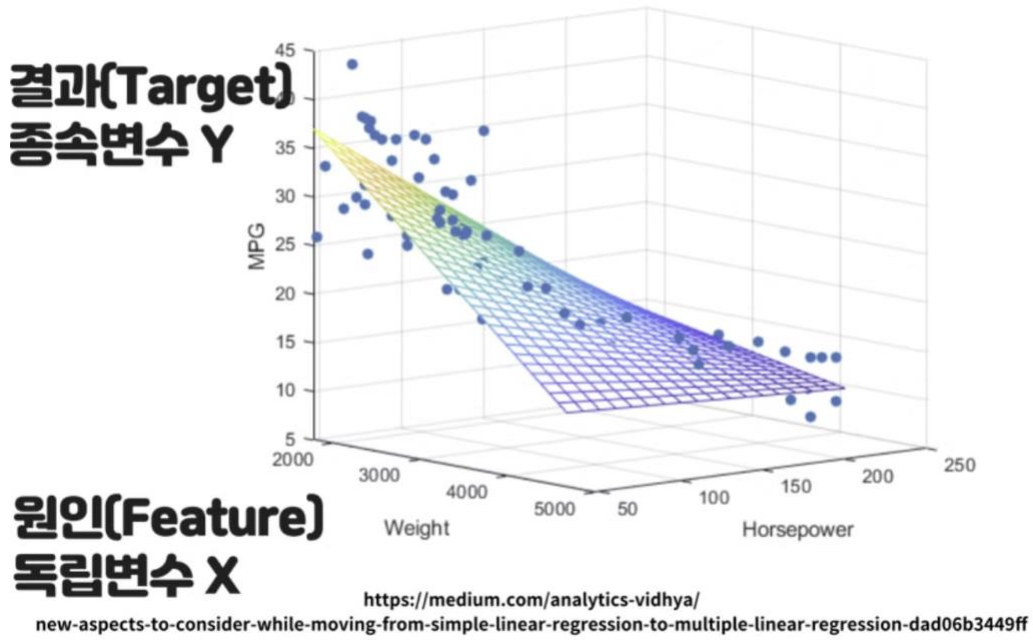
* 차수가 1차가 아닌 경우

-> 독립변수의 개수, 차수 차이

TRAIN AND TEST

다중 선형 회귀 vs 다항식 회귀





https://medium.com/analytics-vidhya/new-aspects-to-consider-while-moving-from-simple-linear-regression-to-multiple-linear-regression-dad06b3449ff

- 독립변수가 2개일때 예시 (평면 vs 곡면)
- 다중 선형 회귀는 모든 독립 변수에대해서 선형이다. 즉 일차 항 계수를 갖는다.

TRAIN AND TEST

다중 선형 회귀 vs 다항식 회귀 예시 설명

대학생 스트레스 수준 =a1×과제양 + a2×시험준비시간 +
a3×수면시간+a4×진로불확실성 + a5×경제적부담 +a6×인간관계

- 대학생의 스트레스 수준이 시험 준비 시간이 감소함에 따라 선형적으로 증가
- Vs
- 점점 더 빠르게 증가

TRAIN AND TEST

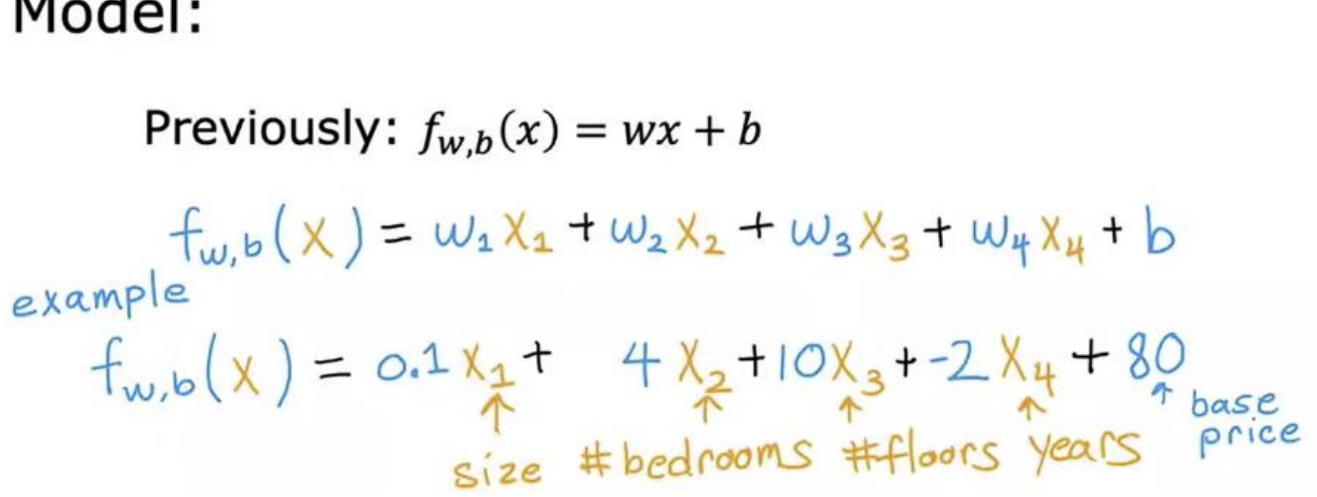
변수의 중요도를 파악할때 - 회귀 계수 값을 비교x

회귀 계수는 각 독립 변수와 종속 변수와의 관계를 나타낸다.
결과적으로 계수가 큰 변수가 보다 큰 변화를 나타내므로 더 중요하다고 생각하기 쉽지만,

각 변수는 단위가 다르므로 직접적인 비교는 불가능하다.

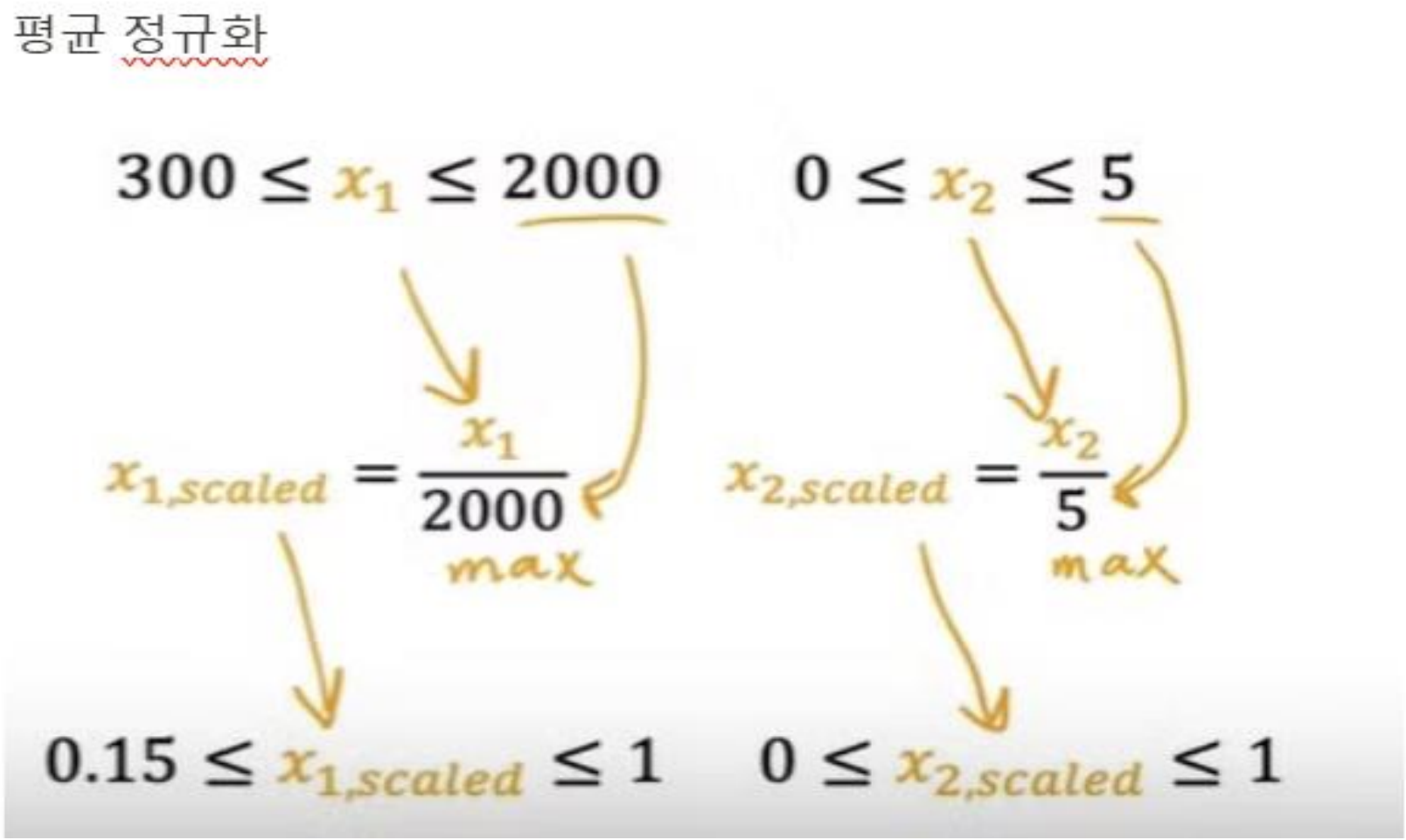
- >

표준화 진행 (feature scaling)



대학생 스트레스 수준 =a1×과제양 + a2×시험준비시간 +
a3×수면시간+a4×진로불확실성 + a5×경제적부담 +a6×인간관계

feature scaling
(1) mean - normalization



feature scaling

(2) Z –score normalization

$$300 \le x_1 \le 2000 \qquad 0 \le x_2 \le 5$$

$$x_1 = \frac{x_1 - \mu_1}{\sigma_1} \qquad x_2 = \frac{x_2 - \mu_2}{\sigma_2}$$

$$-0.67 \le x_1 \le 3.1 \qquad -1.6 \le x_2 \le 1.9$$

다중선형회귀분석 시 유의할 점

- 독립변수(설명변수)들끼리의 상관관계가 높으면 다중공선성 문제**(multicollinearity)**
- 다중 공선성이란, 사용해야 될 독립변수들끼리 서로 밀접한 상관관계가 있어서 다중선형회귀모델에서 각각의 요인들의 효과를 파악하기 어려워지는 것을 말합니다.

다중 공선성의 해결책

- 제거 또는 새로운 변수를 만들어서 해결한다.

TRAIN AND TEST

Feature engineering

Feature engineering

$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + b$$

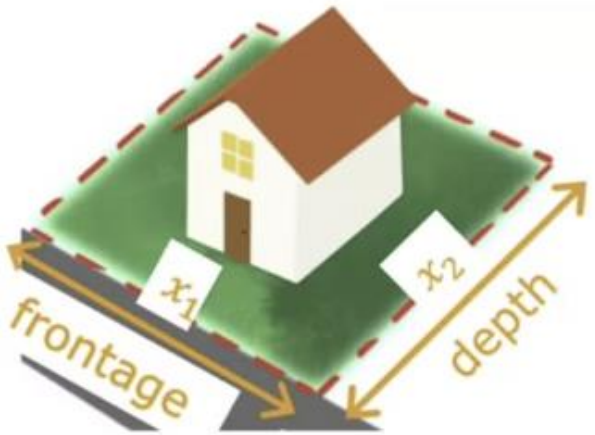
frontage depth

$$area = frontage \times depth$$

$$x_3 = x_1 x_2$$

new feature

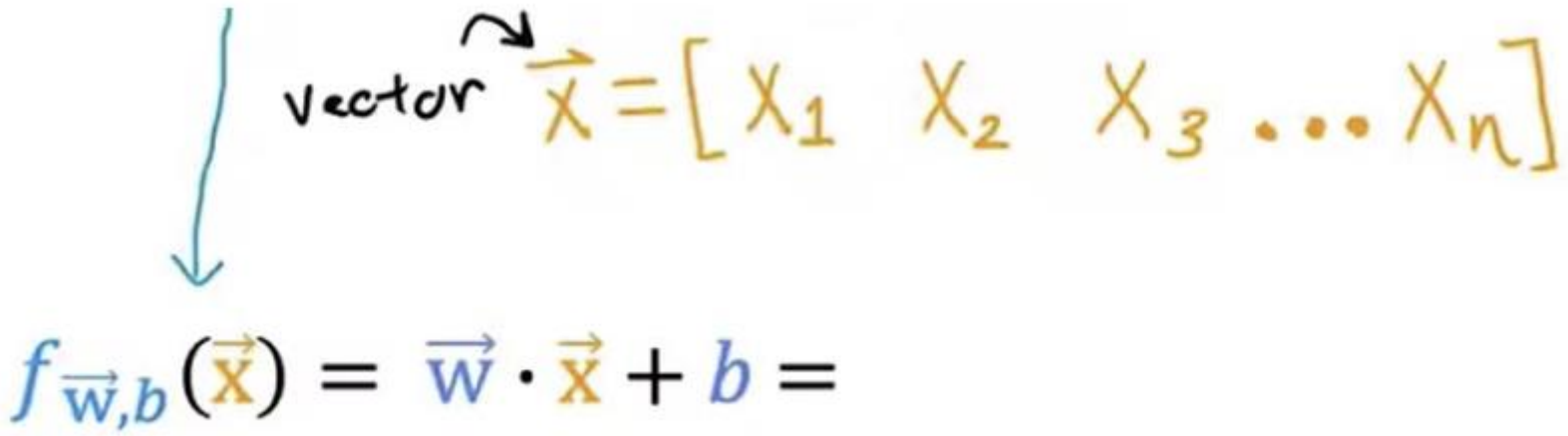
$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$



TRAIN AND TEST

벡터화

벡터화



TRAIN AND TEST

Parameters and features

$\vec{w} = [w_1 \quad w_2 \quad w_3]$ $n=3$

b is a number

$\vec{x} = [x_1 \quad x_2 \quad x_3]$

linear algebra: count from 1

NumPy

$w[0] \quad w[1] \quad w[2]$

```
w = np.array([1.0,2.5,-3.3])
b = 4
x = np.array([10,20,30])
```

$x[0] \quad x[1] \quad x[2]$

code: count from 0

Without vectorization

$$f_{\vec{w},b}(\vec{x}) = \left(\sum_{j=1}^{n} w_j x_j\right) + b \qquad \sum_{j=1}^{n} \rightarrow j = 1 \ldots n \quad 1,2,3$$

range(0,n) → j = 0...n-1

```
f = 0          range(n)
for j in range(0,n):
    f = f + w[j] * x[j]
f = f + b
```

Without vectorization $n = 100,000$

$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

```
f = w[0] * x[0] +
    w[1] * x[1] +
    w[2] * x[2] + b
```

Vectorization

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

```
f = np.dot(w,x) + b
```

TRAIN AND TEST

-> 벡터화된 코드 작성시 더 빠르게 계산 가능함

Without vectorization

```
for j in range(0,16):
    f = f + w[j] * x[j]
```

t_0

```
f + w[0] * x[0]
```

t_1

```
f + w[1] * x[1]
```

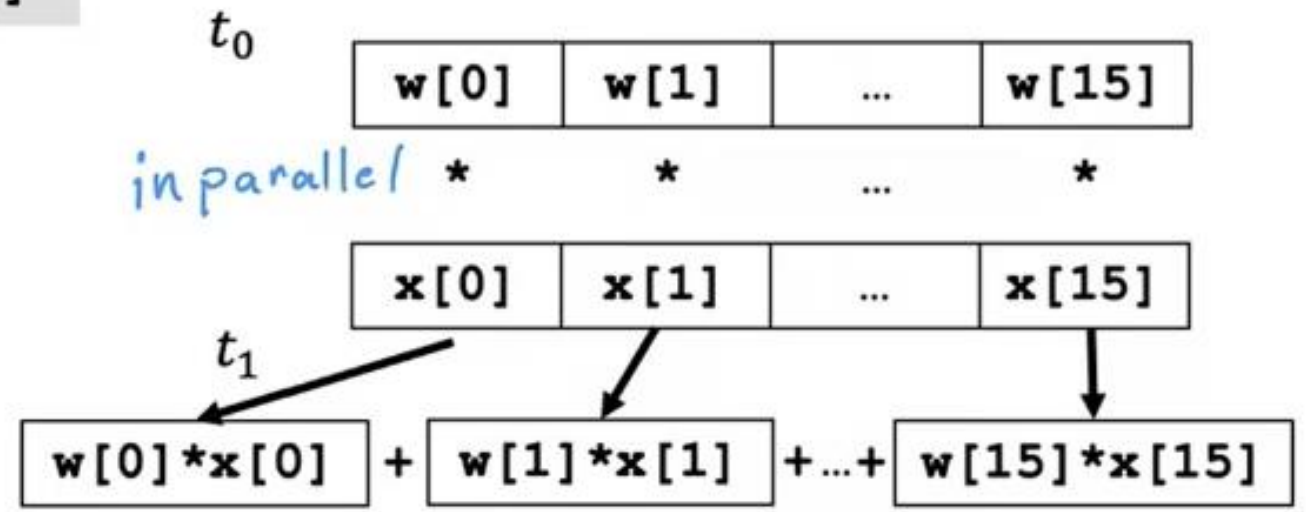
...

t_{15}

```
f + w[15] * x[15]
```

Vectorization

```
np.dot(w,x)
```



직관적으로 이게 왜 컴터에서 계산이 훨 빠른지?

각 쌍을 동시에 병렬로 곱한다. > 차례로 별개의 덧셈을 수행하지 않아도 되서 시간이 훨빠름
대규모 학습 많기 때문에 필수적임

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i \tag{1}$$

여기서 아래첨자 y_i는 i번째 관측치를 의미하고, ϵ_i는 이때의 오차항을 나타낸다. 우리가 추정하고자하는 값, 즉 회귀계수는 β_j $(0 \leq j \leq k)$이고, 독립변수 x_{ij}는 known value이다. 식 (1)을 N개의 샘플에 대하여 확장한 후, vector-matrix 형태로 표기하면 다음과 같다.

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nk} \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}}_{\boldsymbol{\epsilon}} \tag{2}$$

여기서 $\mathbf{e} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_N)$.

TRAIN AND TEST

- 경사 하강법에서 수렴과 학습률

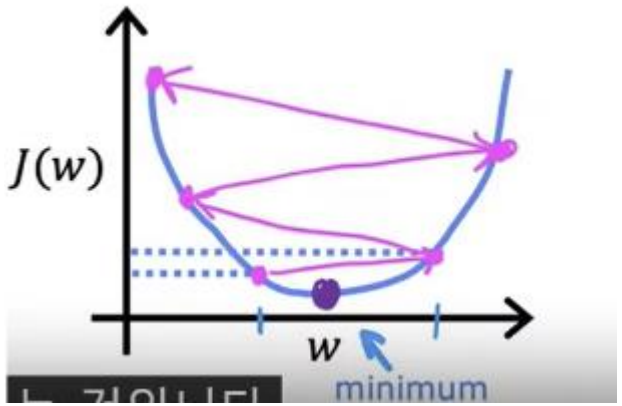
learning rate

$$w = w - \alpha \frac{\partial}{\partial w} J(w,b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w,b)$$

• w= w- (여기서 =은 같다가 아니라, 할당한다는뜻. 업데이트 한다는 뜻 !)

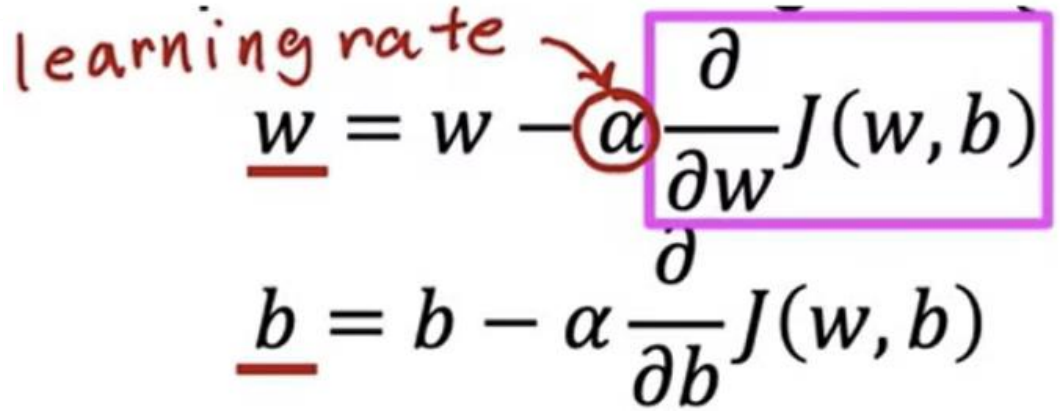
선형회귀 일때 알파는 학습률 (항상 양수)
너무 작으면 느리고
너무 크면 M/m 찾지 못하고,
즉 수렴하지 못하고, 갈라질 수 있다.



적합한 학습률 찾는법

먼저 작은 숫자로 시도하고 3배 , 10배 등으로 조금씩 조정한다.

- 0.001의 학습률을 시도하고
0.01과 0.1 등과 같이 10배 더 큰 학습률을 시도

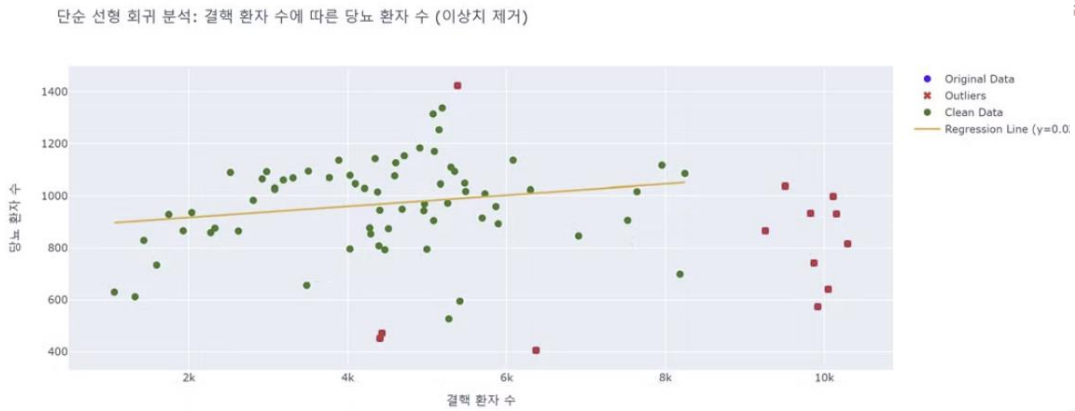


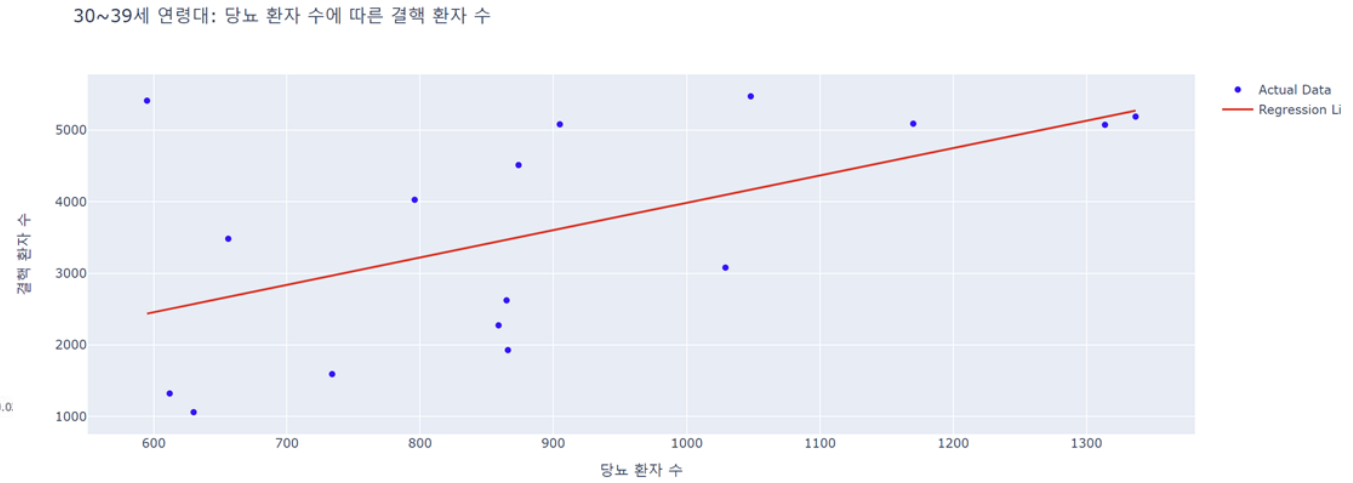
실습

- MSE (Mean Squared Error) = $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- MAE (Mean absolute error) = $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- RMSE (Root Mean Squared Error) = $\sqrt{MSE}$
- R-squared (Coefficient of determination) = $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} = 1 - \frac{SSE}{SST} = \frac{SSR}{SST}$

```
((y-y.mean())**2).sum()
```

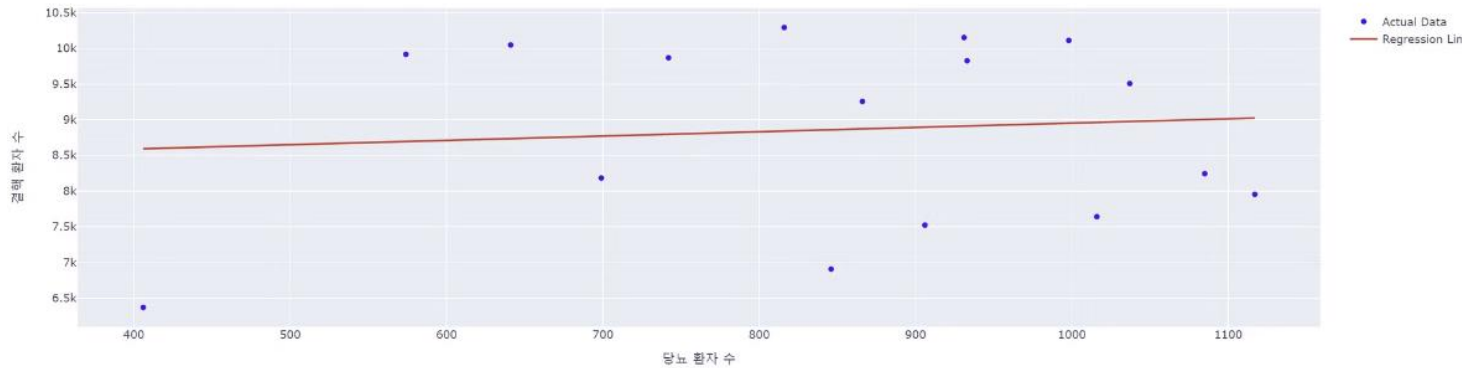
당뇨환자 수와 결핵환자 수 회귀 분석 실습





30~39세 연령대: 당뇨 환자 수에 따른 결핵 환자 수

70세 이상 연령대: 당뇨 환자 수에 따른 결핵 환자 수



뉴욕 맨해튼 의 주택 임대료 데이터 실습

manhattan.csv
git clone https://github.com/Codecademy/datasets.git head
/home/ubuntu/workspace/datasets/streeteasy/manhattan.csv

정답
https://velog.io/@hyesoup/다준선형회귀Multiple-Linear-Regression-예제
https://recipesds.tistory.com/entry/맨하탄집값-분석-다중회귀-예측까지-실습