Multiple Features

Size in feet	Number of bedrooms	Number of floors	Age of home in years	price $1000's
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

x_j = j^{th} feature

n = number of features

$\vec{x}^{(i)}$ = features of i^{th} training example

$x_j^{(i)}$ = Value of feature j in i^{th} training example

$\vec{x}^{(2)} = [1416 \;\; 3 \;\; 2 \;\; 40]$

$x_3^{(2)} = 2$

Model,

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

example)

$$f_{w,b}(x) = \underset{\text{size}}{0{,}1\, x_1} + \underset{\text{bedroms}}{4\, x_2} + \underset{\text{floors}}{10\, x_3} - \underset{\text{years}}{2\, x_4} + \underset{\text{base price}}{80}$$

$$f_{w,b}(x) = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

$$\vec{w} = [w_1 \quad w_2 \quad w_3 \quad w_4]$$

$$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4]$$

$$f_{\vec{w},b}(\vec{x}) = \vec{x} \cdot \vec{w} + b = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n + b$$

multiple linear regression (not multivariate regression)

Vectorization (Vektörleştirme)

→ Kodun daha kısa olmasını ve daha hızlı çalışmasını sağlar.

parameters and features

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

b is number

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

☆

$$\begin{array}{ccc} w[0] & w[1] & w[2] \end{array}$$

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

$$\begin{array}{ccc} x[0] & x[1] & x[2] \end{array}$$

without vectorization

$$f_{w,b}(\vec{x}) = w[0]*x[0] + w[1]*x[1] + w[2]*x[2] + b$$

$$f_{w,b}(\vec{x}) = \left(\sum_{j=1}^{n} w_jx_j\right) + b$$

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

Vectorization

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

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

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[0] * X[0]

⋮

t_{15} f + w[15] * X[15]

Vectorization

```
np.dot[w,x]
```

t_0

W[0]	W[1]	...	W[15]
*	*	*	*
X[0]	X[1]	...	X[15]

t_1

W[0] * X[0] + W[1] * X[0] + --- + W[15] * X[15]

Gradient descant

$\vec{w} = (w_1 \quad w_2 \quad \dots\dots \quad w_{16})$

$\vec{d} = (d_1 \quad d_2 \quad \dots\dots \quad d_{16})$

```
w = np.array([0.5, 1.3, .... 3.4])
d = np.array([0.3, 0.2, .... 0.4])
```

Compute $w_j = w_j - 0.1 d_j$ for $j = 1\ 2\ 3\ 4\ 5 \dots 16$

Without vectorization

$w_1 = w_1 - 0.1 d_1$

$w_2 = w_2 - 0.1 d_2$

⋮

$w_{16} = w_{16} - 0.1 d_1$

```
for j in range (0,16):
    w[j] = w[j] - 0.1 * d[j]
```

with vectorization

$\vec{w} = \vec{w} - 0.1 \vec{d}$

```
w = w - 0.1 * d
```

Çoklu doğrusal Regresyon için dereceli Azalma

	Previous notation	Vector notation
Parameters	$w_1, w_2, w_3, \dots w_n$ b	$\vec{w} = [w_1 \ w_2 \dots w_n$ b
Model	$f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$	$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$ dot product
Cost function	$J(w_1, w_2, \dots w_n, b)$	$J(\vec{w}, b)$

Gradient descent

repeat{

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} (w_1, w_2, \dots w_n, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} (w_1, w_2, \dots w_n, b)$$

}

repeat{

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

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

Gradient Descent

One feature

repeat{

$$w = w - \alpha \underbrace{\boxed{\frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)}}}_{\frac{\partial}{\partial w} J(w,b)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})$$

Simultaneously update w, b

}

n features $(n \geq 2)$

repeat {

$$w_1 = w_1 - \alpha \underbrace{\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) \vec{x}_1^{(i)}}_{\frac{\partial}{\partial w_1} J(\vec{w}, b)}$$

$\vdots$

$$w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_n^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})$$

Simultaneously update

w_j (for $j = 1, \dots n$) and b

}

An Alternative to gradient descent

→ Normal denklemi

- Sadece lineer regresyonda
- Yineleme olmadan w ve b çözer

Dezavantajları

Diğer öğrenme algoritmalarda çalışmaz

Özellikler büyük olursa çalışmaz. (>10000)

Pratikte dereceli Azalma

Feature Scaling (özellik ölçeklendirme)

Feature and Parameter values

Price = $w_1 x_2 + w_2 x_2 + b$

$w_1 x_2$ ↓ size, $w_2 x_2$ ↓ bedrooms

size	bedrooms
0-2000	0-5
large	small

House = $x_1 = 2000$ $x_2 = 5$ price = 500k

Size of parameters w_1, w_2?

$w_1 = 50$ $w_2 = 0{,}1$ $b = 50$

$\widehat{price} = 50 \times 2000 + 5.0{,}1 + 50$

100.000k   0,5k   50k

= 100,050,500 $

$w_1 = 0.1$ $w_2 = 50$ $b = 50$

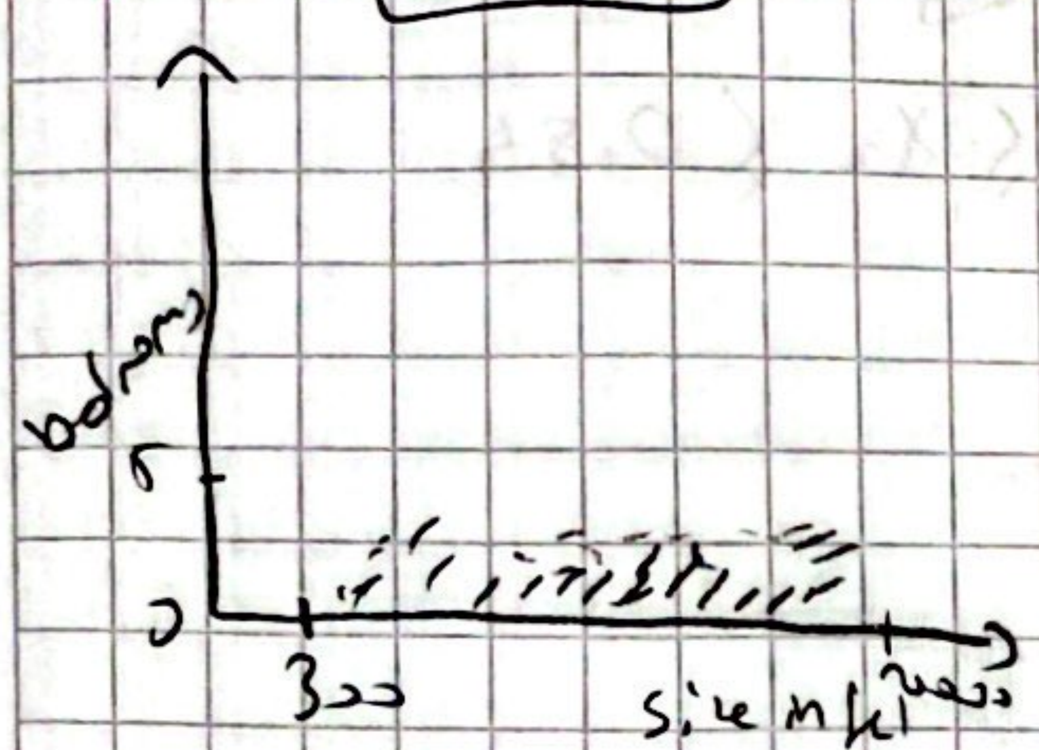
$\widehat{price} = 0.1 \times 2000 + 5.50 + 50$

200k + 250k + 50k

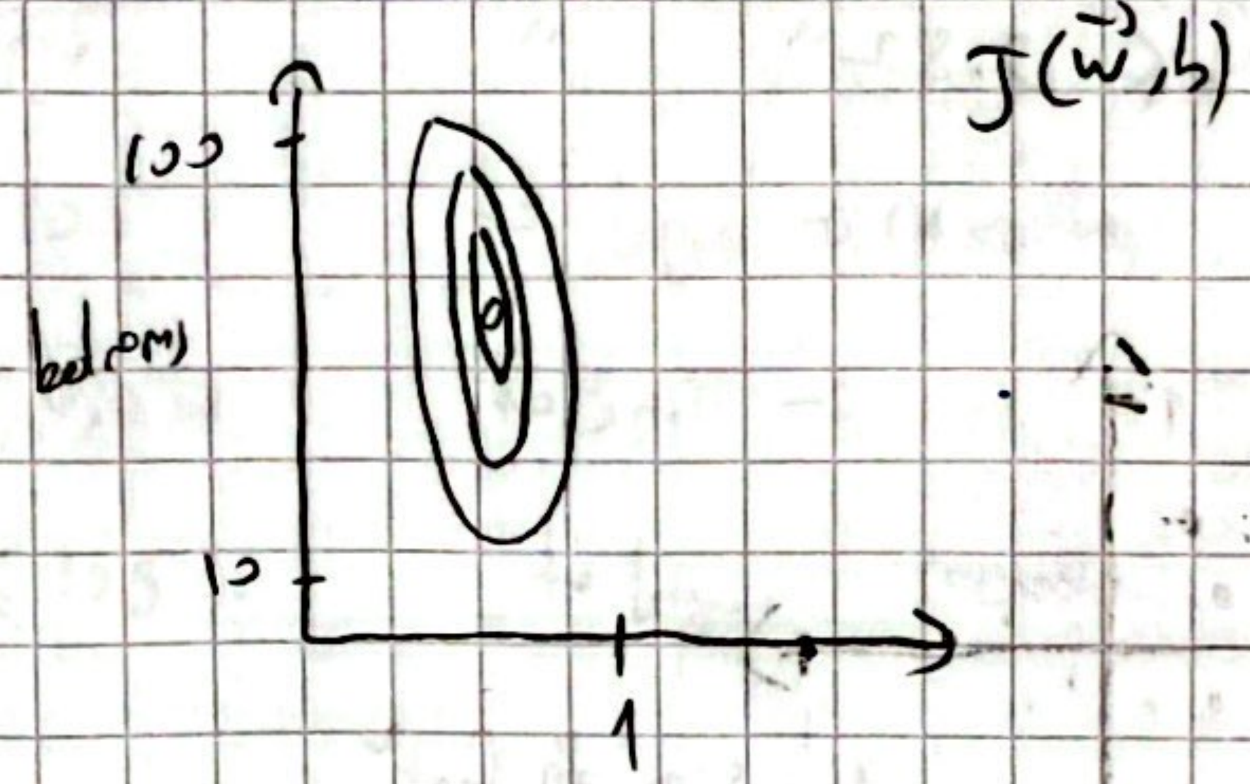
= 500k ✓

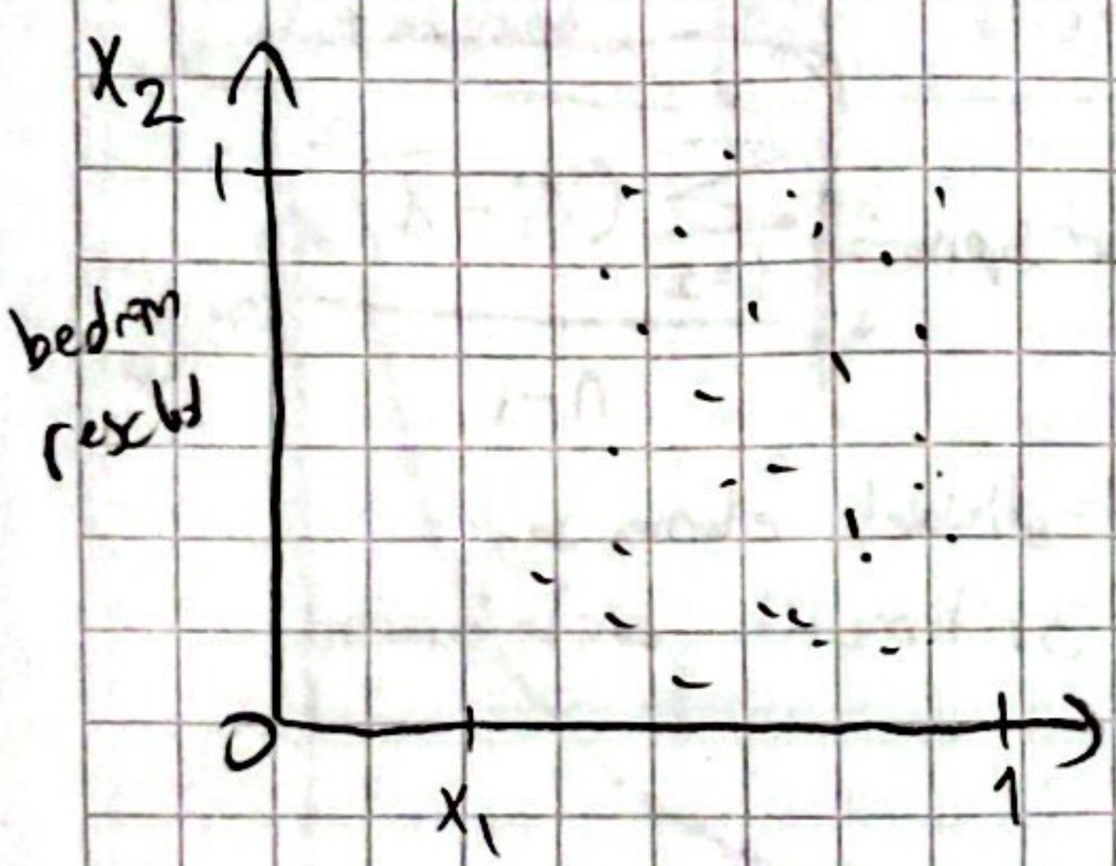
	Size of feature	Size of parameter
Size in feet^2	⟵⟶	⟷
#bedrooms	⟷	⟵⟶

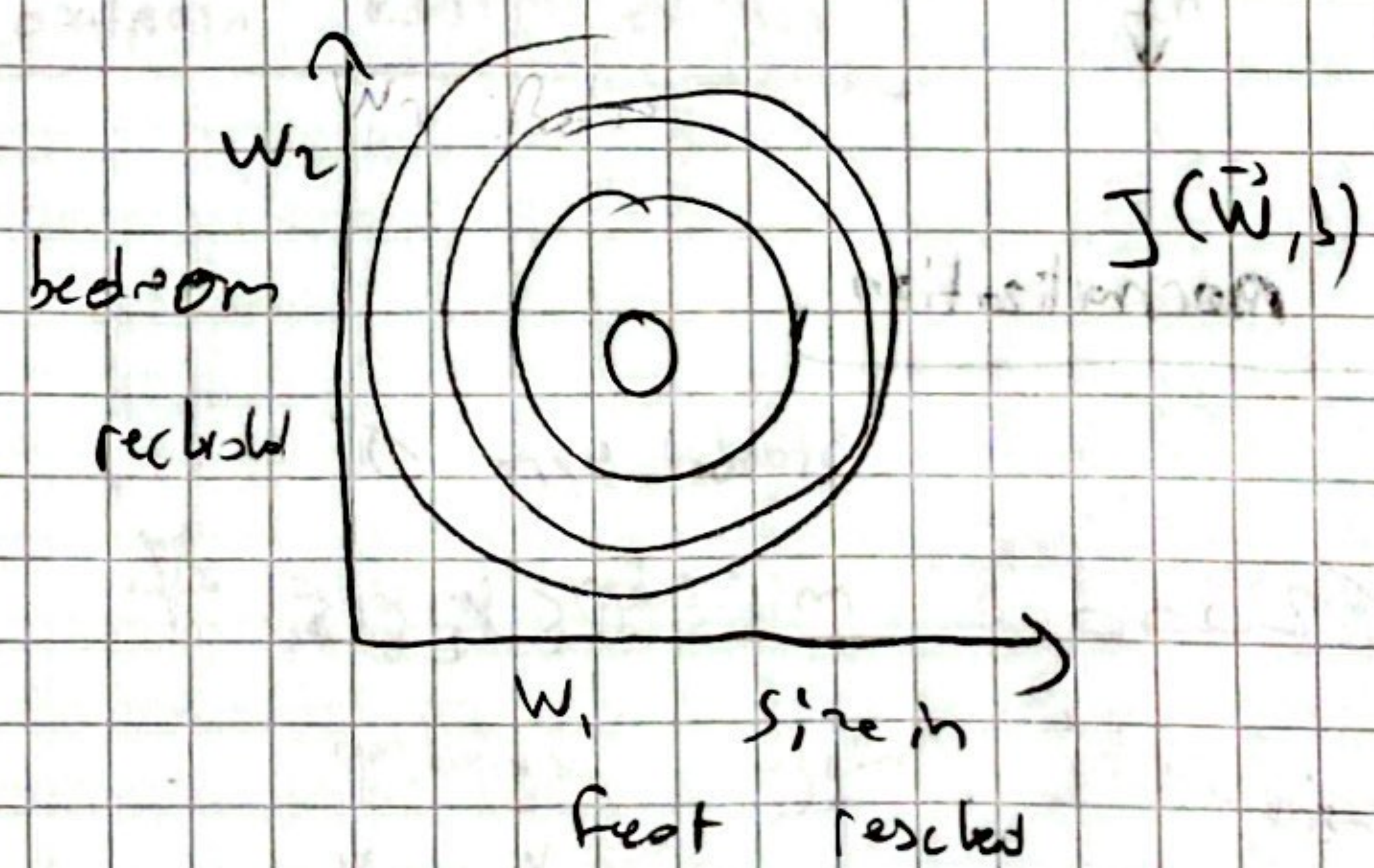
Features



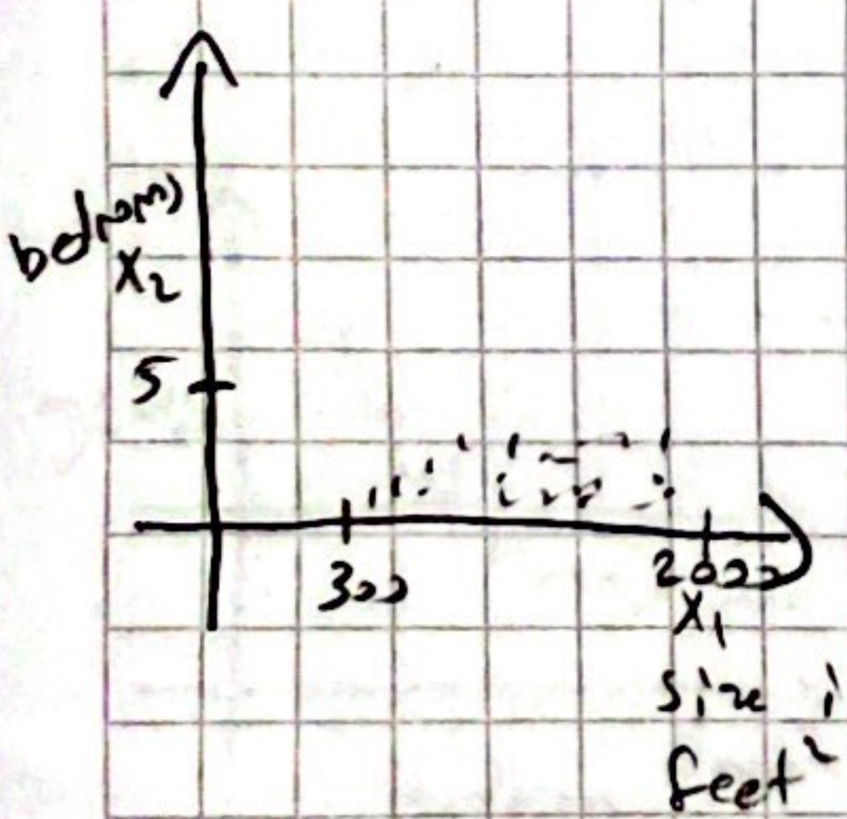
Parameters







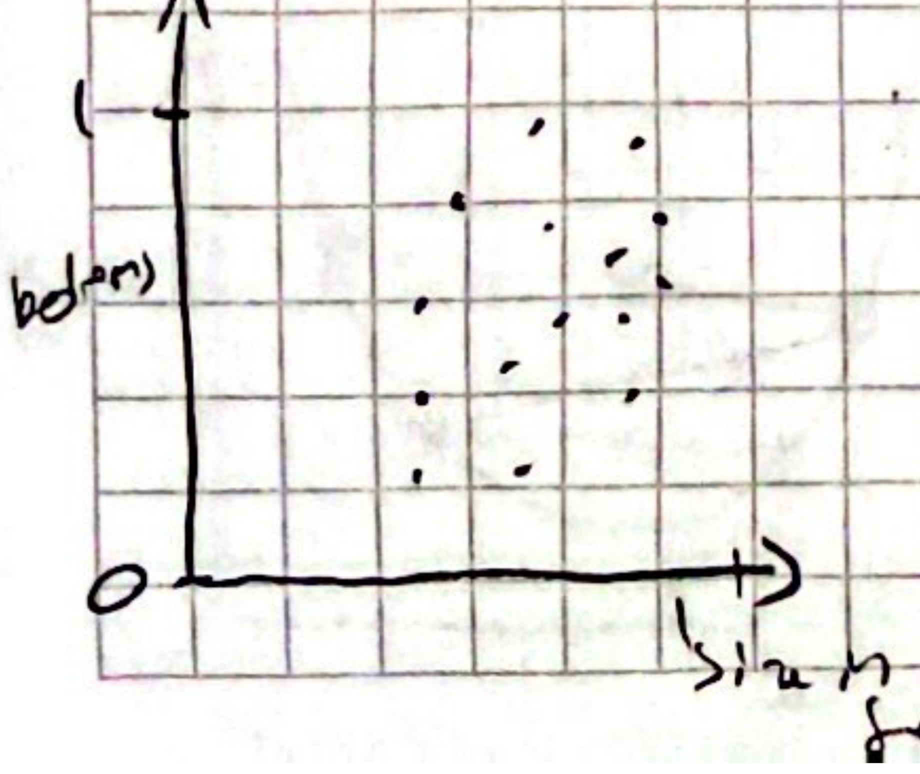
Feature Scaling



$$300 < X_1 < 2000 \qquad 0 < X_2 < 5$$

$$X_{1,scaled} = \frac{X_1}{\underbrace{2000}_{max}} \qquad X_{2,scaled} = \frac{X_2}{5}$$

$$0{,}15 < X_{1,scaled} < 1 \qquad 0 < X_{2,scaled} < 1$$



Mean normalization

$300 \leq X_1 \leq 2000$

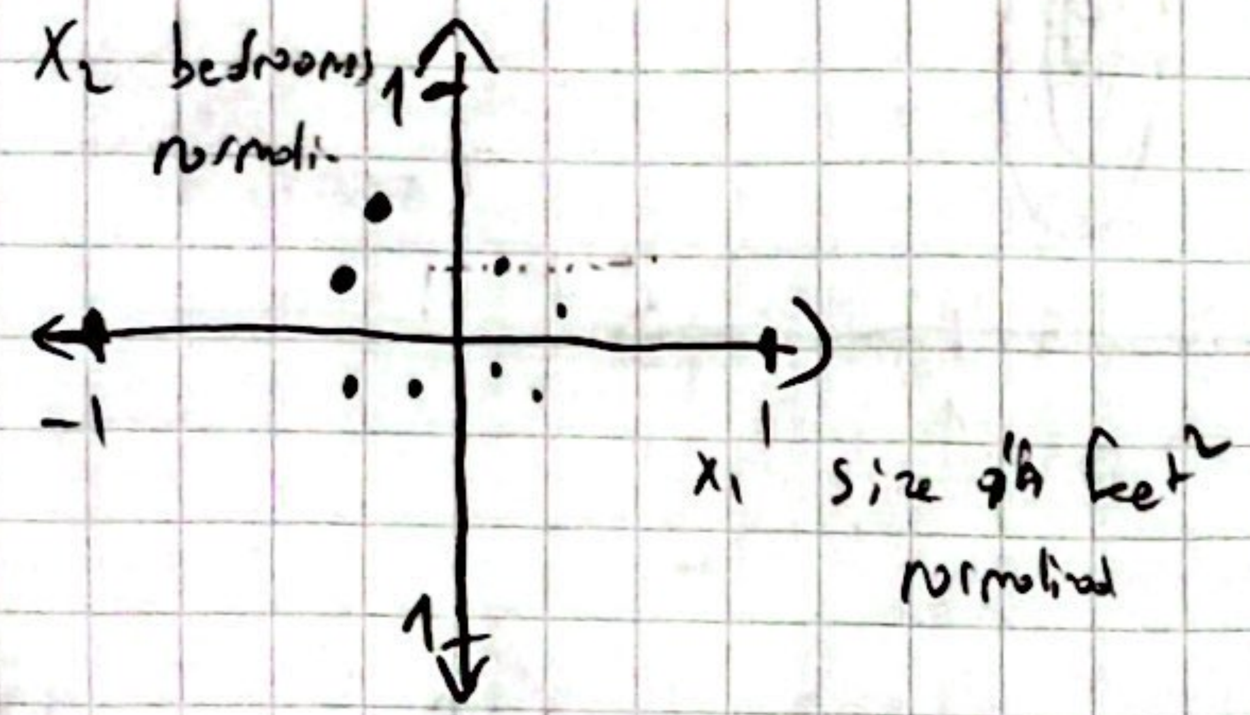
$$X_{1,\text{normalize}} = \frac{X_1 - \mu_1^{\to \text{mean}}}{2000-300}$$

$-0.18 \leq X_1 \leq 0.82$

$0 \leq X_2 \leq 5$

$$X_2 = \frac{X_2 - \mu_2}{5-0}$$

$-0.46 \leq X_2 \leq 0.54$



$$\text{Standart sapma: } \sqrt{\frac{\sum_{i=1}^{N} (X_i - \bar{X})^2}{n-1}}$$

N = dizideki eleman sayısı
$\bar{X}$ aritmetik ortalama

Z-Score normalization

Standart sapma σ

$300 \leq X_1 \leq 2000$

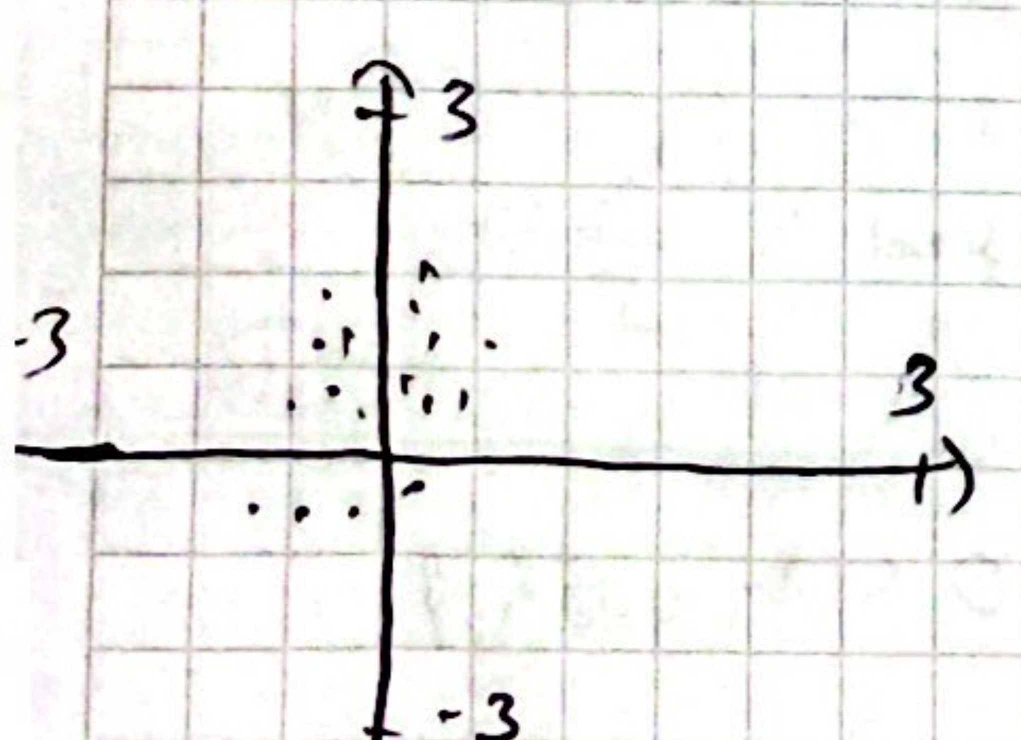
$$X_1 = \frac{X_1 - \mu_1^{\to \text{mean}}}{\sigma_1}$$

$-0.67 \leq X_1 \leq 3.1$

$0 \leq X_2 \leq 5$

$$X_2 = \frac{X_2 - \mu_2}{\sigma_2}$$

$-1.6 \leq X_2 \leq 1.9$



Amaç: Her değeri

$$\left.\begin{array}{l} -1 \leq x_j \leq 1 \\ -3 \leq x_j \leq 3 \\ -0,3 \leq x_j \leq 0,3 \end{array}\right\} \text{acceptable ranges}$$

$0 \leq x_1 \leq 3$ okay, no rescalling

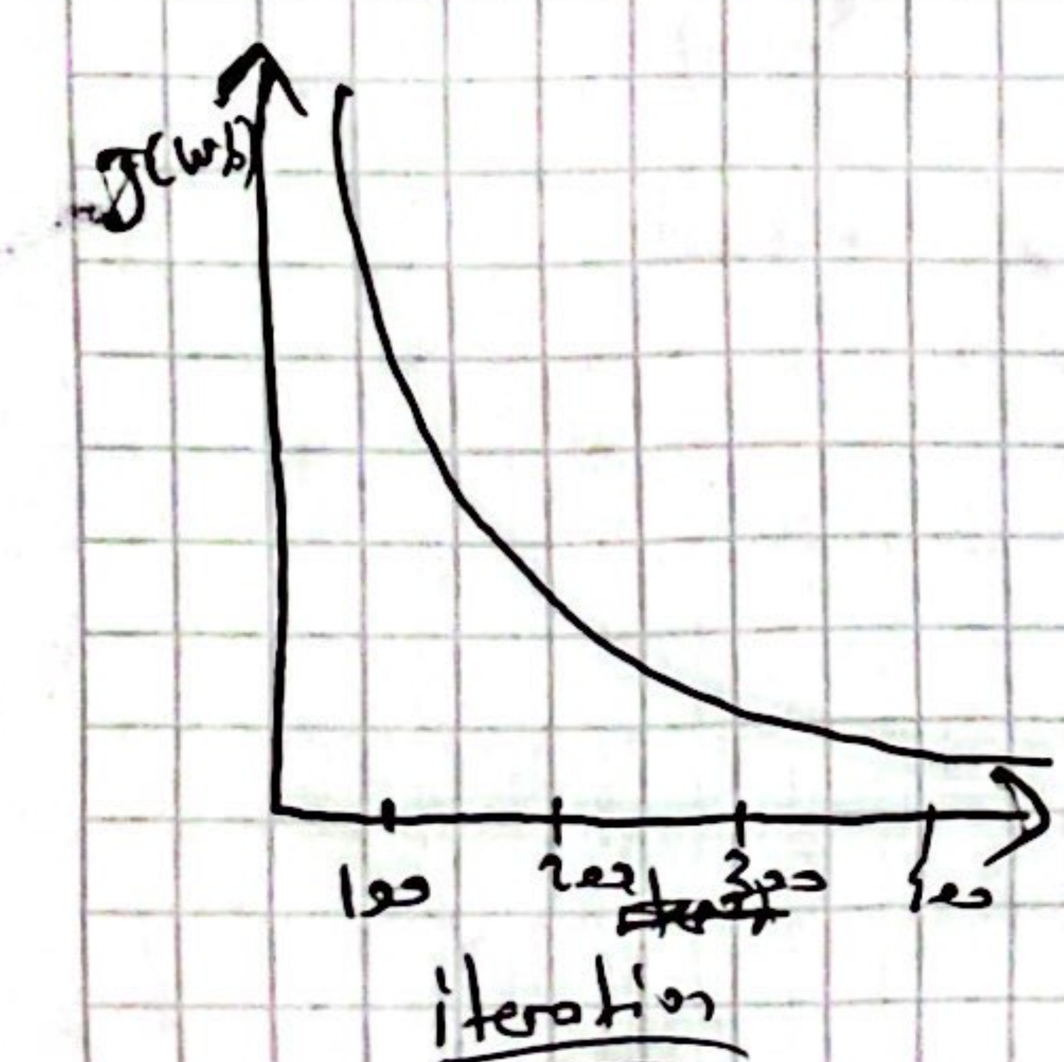
$-2 \leq x_2 \leq 0.5$ " "

$-100 \leq x_3 \leq 100$ to large → rescalling

$-0,001 \leq x_4 \leq 0,001$ to small → "

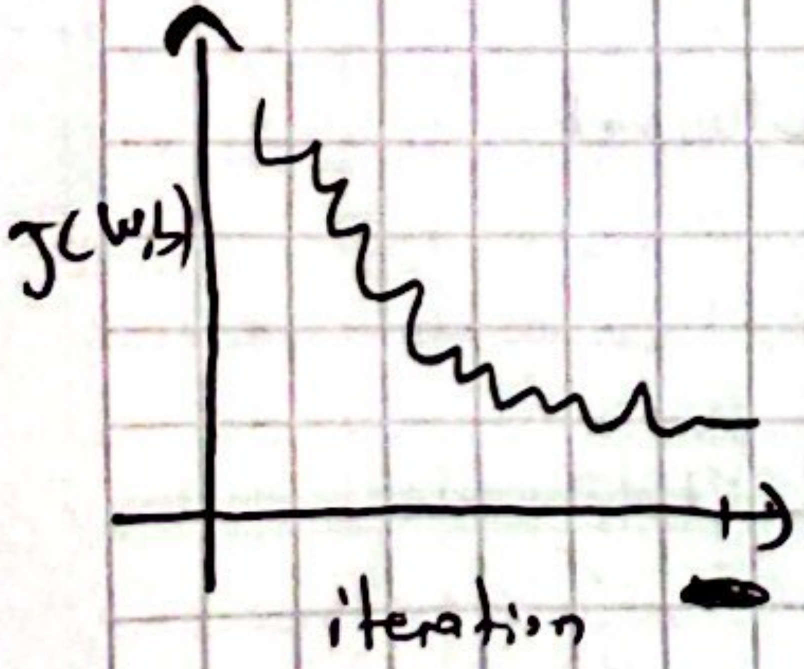
$98,5 \leq x_5 \leq 105$ to large rescalling

Yakınsama için Dereceli azalmanın kontrol edilmesi



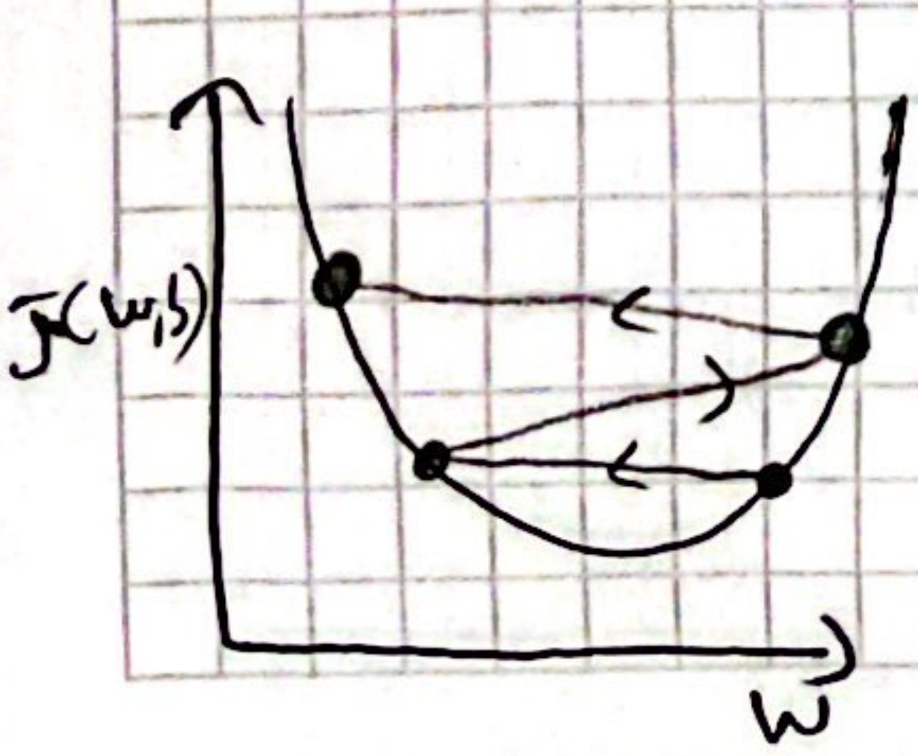
★ J(w,b) değişim $\varepsilon = 10^{-3}$ den az olmaya başladıysa uygun J(w,b) bulunmuş demektir

Chasing the Learning Rate

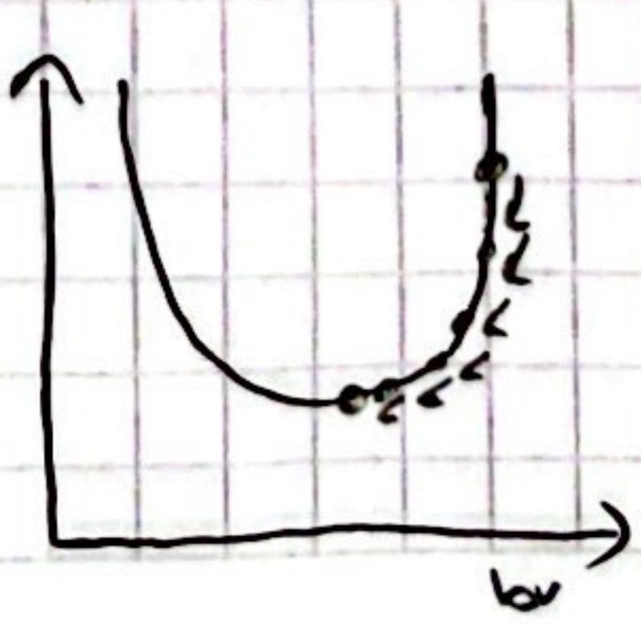


⇒ Kodda bir hata olabilir

⇒ öğrenme oranı büyük olabilir.

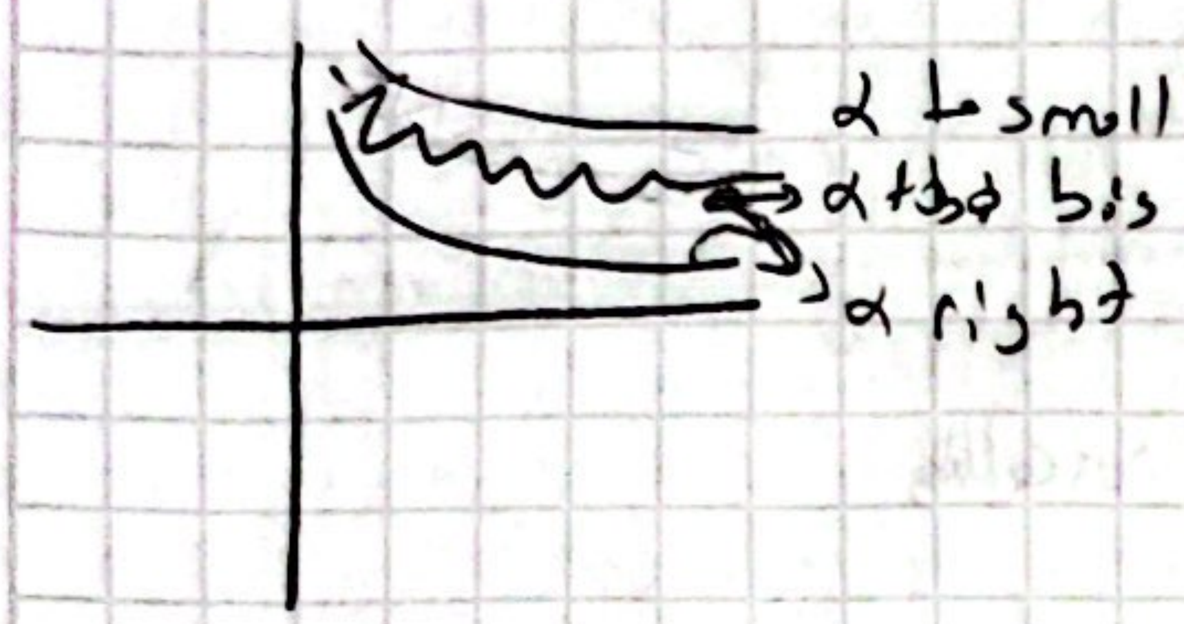


α is too big ⇒ use smaller α



α küçük ise çok fazla adım gerekir

α büyük ise uygun $J(w,b)$ değerini tutturamaz



Feature Engineering

$$f_{\vec{w},b}(\vec{X}) = w_1 \underbrace{x_1}_{\text{frontage}} + w_2 \underbrace{x_2}_{\text{depth}} + b$$

area = frontage X depth

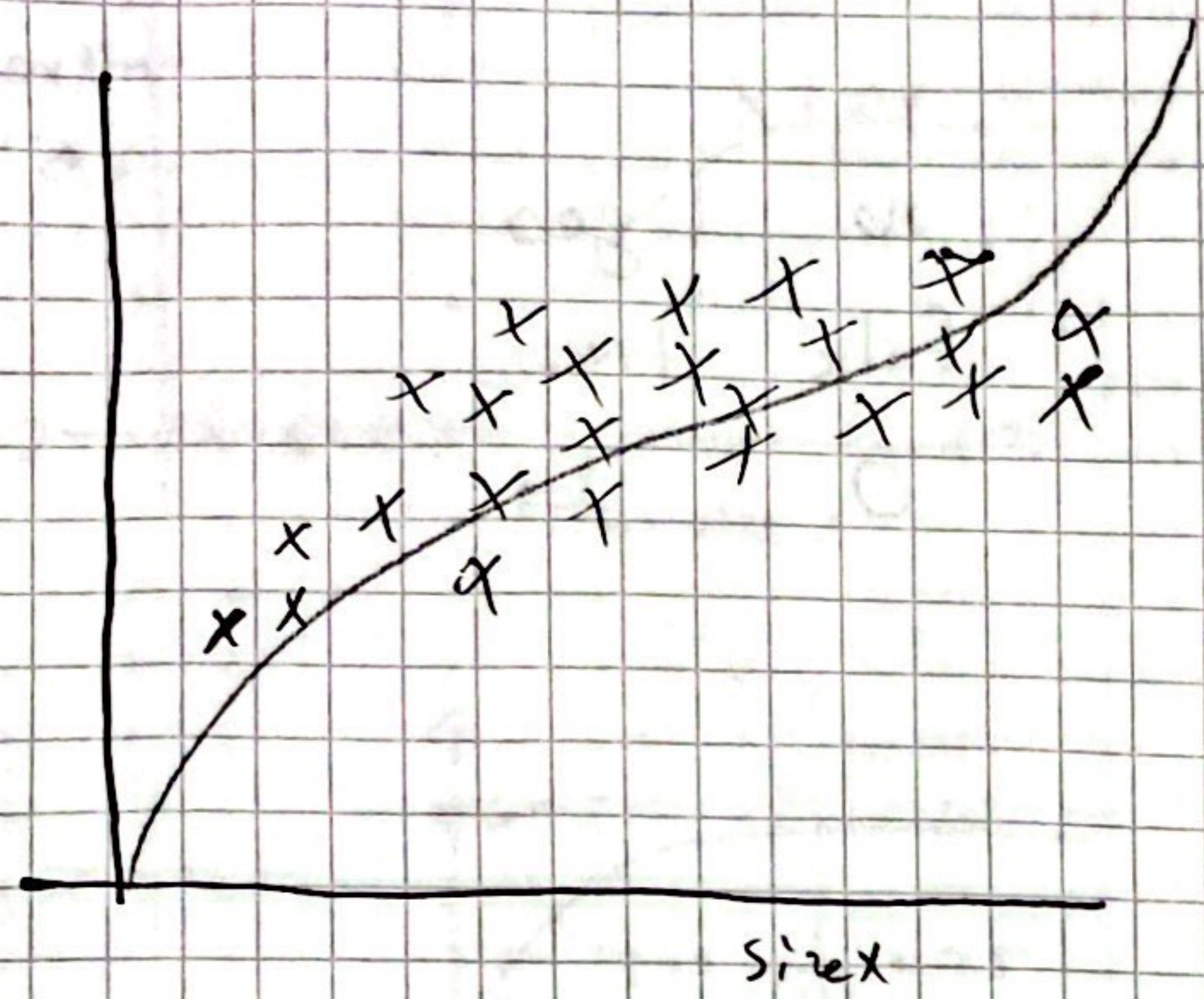
$x_3 = x_1 x_2$

new feature

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

özellik mühendisliği : Orjinal özellikleri dönüştürerek veya birleştirerek yeni özellikler tasarlamak için sezgiyi kullanmak

Polinom Regresyon



$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + b$$

$\uparrow$ size — $\uparrow$ $(size)^2$ — $\uparrow$ $(size)^3$

$1-10^3$ — $1-10^6$ — $1-10^9$

feature scaling

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

$\uparrow$ size — $\uparrow$ $\sqrt{size}$