

Session 3

Scaling Data



۸: چرا بهتر از قبل از اجرای Gradient Descent داده‌ها را Scale کنیم؟

	A	B	C	D	E
1	area	bedrooms	balcony	age	price
2	1200	2	0	2	500000
3	2300	3	2	5	620000
4	2500	4	2	1	122500
5	3650	5	3	3	6000000
6	1800	3	1	5	2122000
7	3000	3	1	4	120000
8	1222	1	0	2	450000
9	4600	5	3	1	6500000
10	2050	2	2	2	1530000
11	1450	2	2	3	1563330

دیتاستی مثل House Price را در نظر بگیرید، محدوده متغیر feature ها این

Area (Size): 100 - 2000 بسیار متفاوت است.

Bedrooms: 0 - 10

Balcony: 0 - 5

Age: 1 - 30

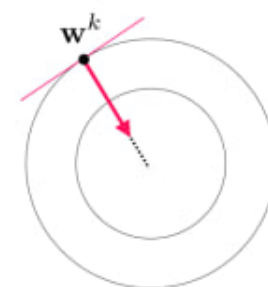
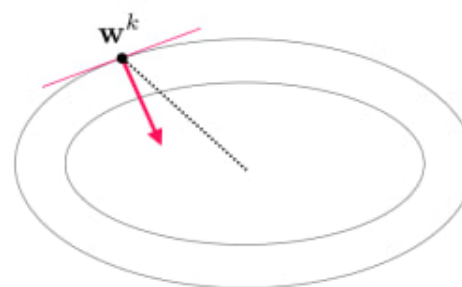
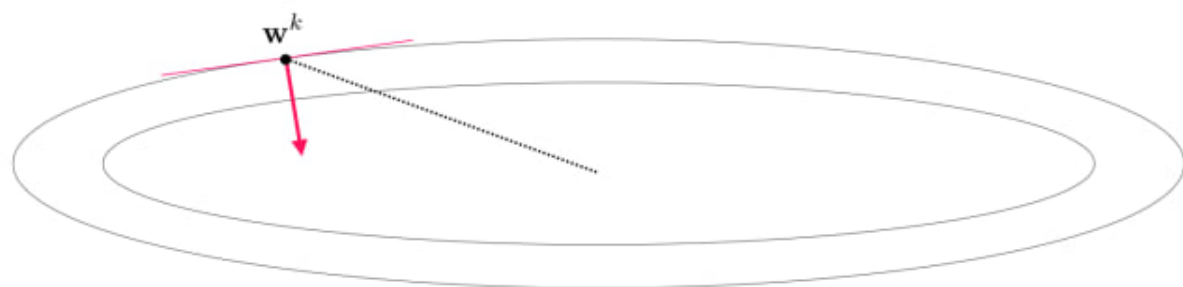
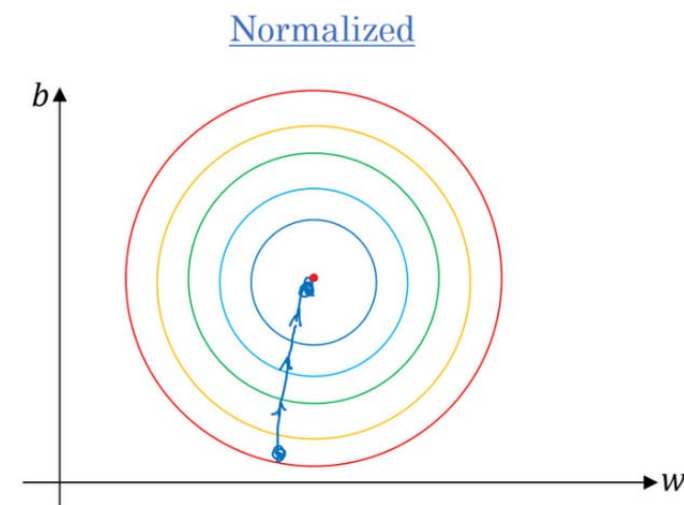
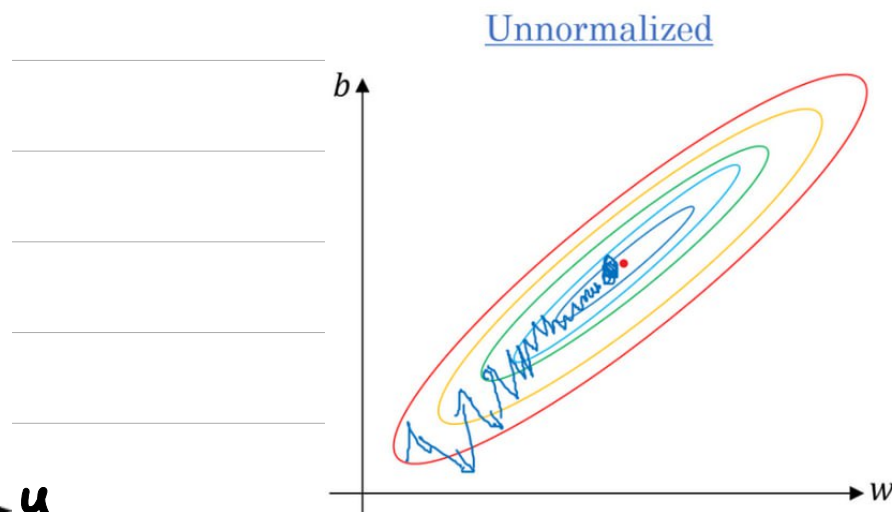
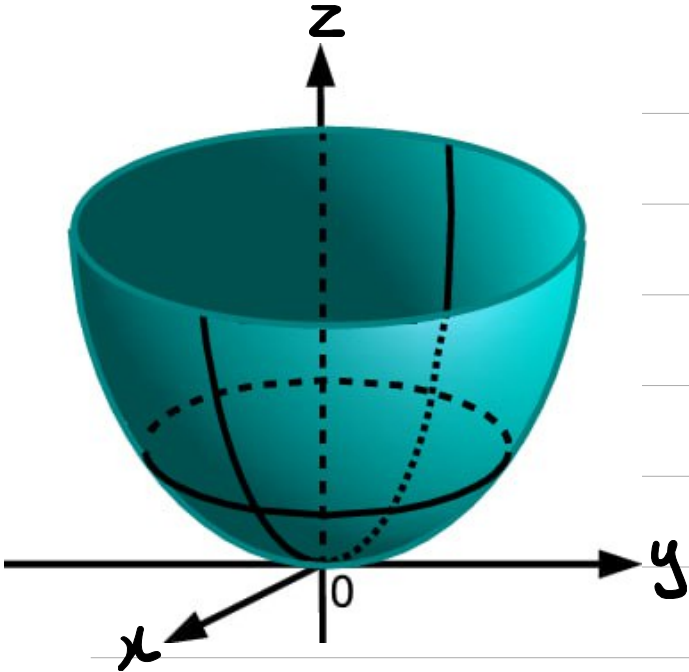
● محدوده عددی area در مقایسه با سایر feature ها بسیار بزرگتر است، این یعنی

یک تغییر کوچک در area تأثیر بسیار بیشتری بر Cost دارد نسبت به بقیه feature ها.

● زمانی که گرادینان یا قواعد Δ را آپدیت کنید باید برای آپدیت Δ مربوط به area بروز رسانی بزرگتری نسبت به سایر feature ها انجام دهید.

● محدوده متفاوت بروز رسانی ها باعث نوسان های نامنظم GD می شود، که باعث می شود عملکردی سخت، زمان بر یا ناممکن شود.

● Learning Rate ← انتخاب یک نرخ یادگیری مناسب چالش برانگیز است.



حالا چه سود؟

1. سرعت همگرایی بیشتری شود.

2. Feature در یادگیری کم یکسانی دارد و یادگیری Stable است.

3. پارامترها و ضرایب پارامترها آسان تر تنظیم می شوند.

4. خطاهای عددی ناشی از مقادیر بسیار بزرگ و بسیار کوچک را کاهش می دهد.

حال برای رفع این مشکل چه کنیم؟

ما رویم سراغ Scaling داده‌ها، برای این کار روشهای مختلفی داریم مثل:

1. Min-Max Scaler

2. Standard Scaler (Z-Score Normalization)

•
•
•

1. Min-Max Scaler

$$X_{\text{Scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

f_1	f_2
1	10
2	20
3	30
4	40

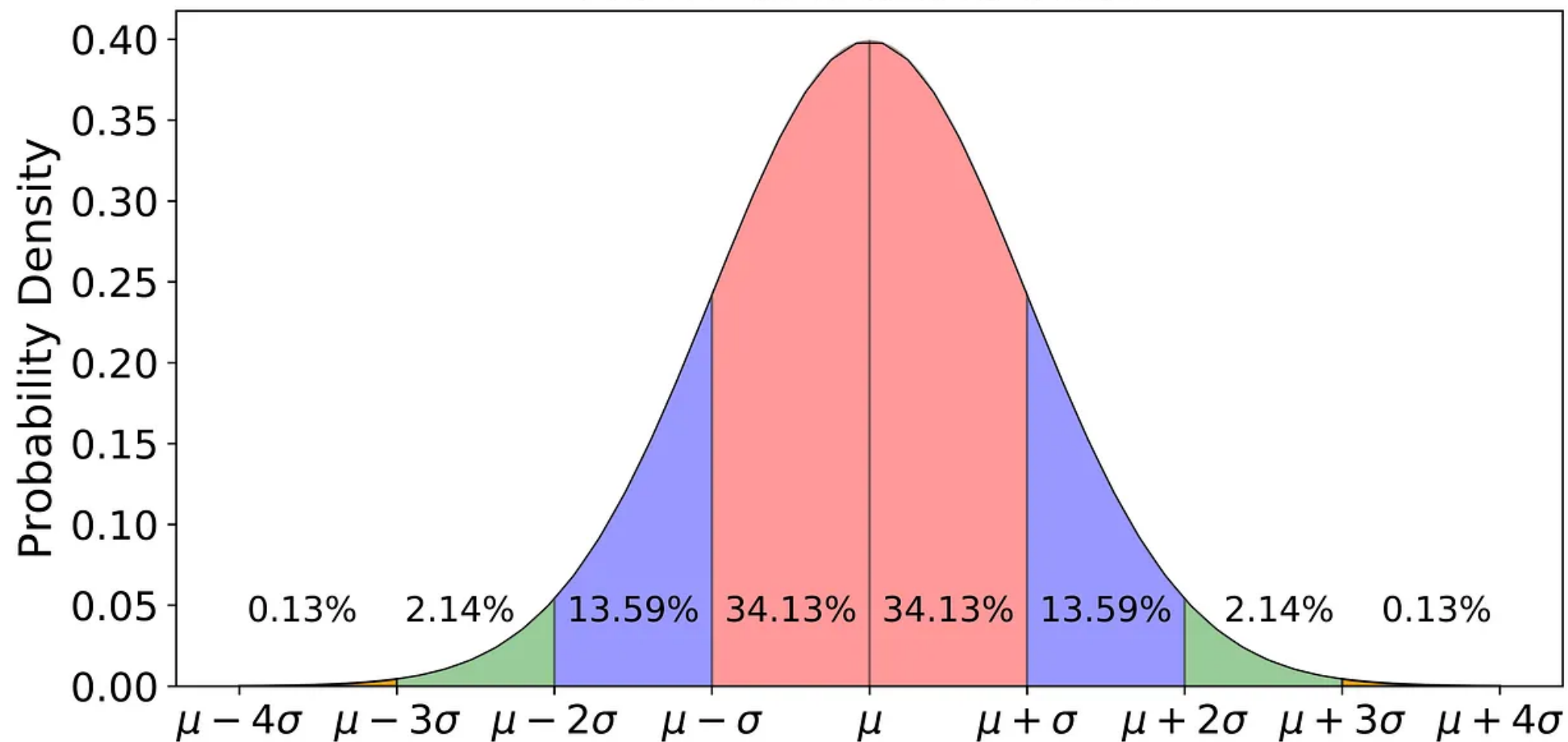
$$f_1: X_{\min}=1, X_{\max}=4 \leadsto X_{\max}-X_{\min}=3$$

$$\begin{bmatrix} (1-1)/3 \\ (2-1)/3 \\ (3-1)/3 \\ (4-1)/3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.33 \\ 0.66 \\ 1 \end{bmatrix}$$

$$f_2: X_{\min}=10, X_{\max}=40 \leadsto X_{\max}-X_{\min}=30$$

$$\begin{bmatrix} (10-10)/30 \\ (20-10)/30 \\ (30-10)/30 \\ (40-10)/30 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.33 \\ 0.66 \\ 1 \end{bmatrix}$$

Normal Distribution



2. Standard Scaler (Z-Score Normalization)

$$Z = \frac{X - \mu}{\sigma} \rightarrow \mu = \frac{\sum_{i=1}^m X_i}{m} \quad \sigma = \sqrt{\frac{\sum_{i=1}^m (X_i - \mu)^2}{m}}$$

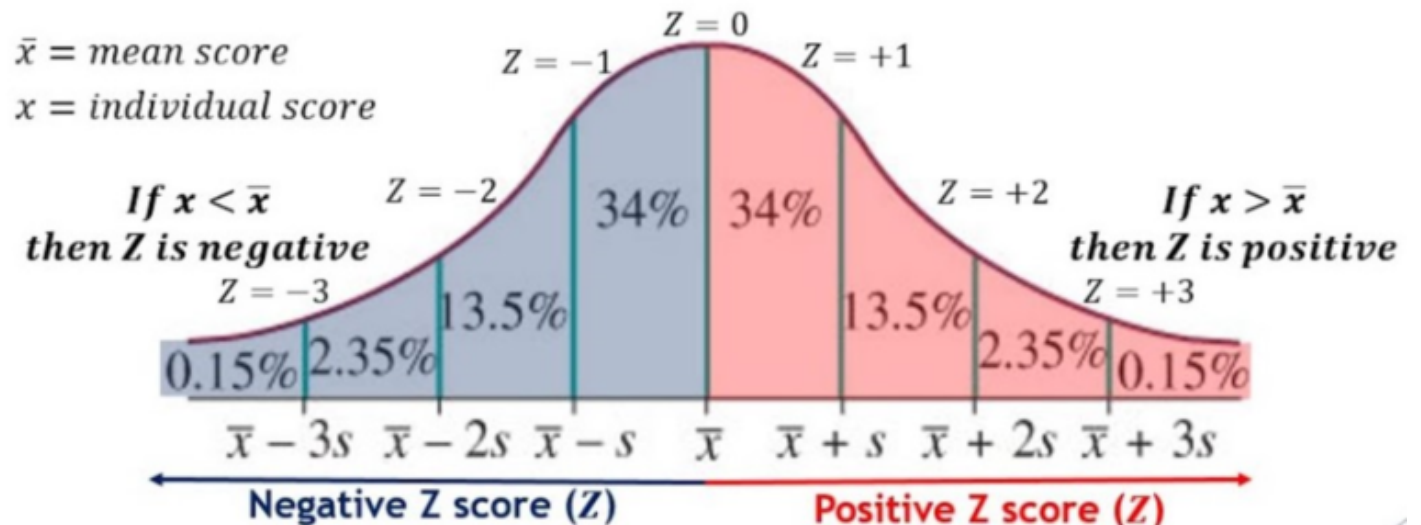
$$\text{f.i. } \mu_1 = \frac{1+2+3+4}{4} = \frac{10}{4} = 2.5 \quad \sigma = \sqrt{\frac{(1-2.5)^2 + (2-2.5)^2 + (3-2.5)^2 + (4-2.5)^2}{4}} = 1.29$$

$$\begin{bmatrix} (1-2.5)/1.29 \\ (2-2.5)/1.29 \\ (3-2.5)/1.29 \\ (4-2.5)/1.29 \end{bmatrix} = \begin{bmatrix} -1.16 \\ -0.38 \\ -0.38 \\ 1.16 \end{bmatrix}$$

Z Score Formula



$$Z = \frac{X - \mu}{\sigma}$$



❓ آیا باید Scaling بروی y یا $target$ هم انجام شود؟ چرا؟

خیر، چون:

① اگر مقیاس متغیر y تغییر کند تفسیر پذیری نتایج را دچار مشکل می کند. اگر y را $Scale$ کنیم باید برای تفسیر نتایج در واحد صبح جدا آن را به واحد اصلی برگردانیم که این کار اضافی است و ممکن زمینه ایجاد خطا در تفسیر نتایج را ایجاد کند.

② از طرفی $Scaling$ متغیر y تأثیری بر عملکرد مدل ندارد چون ما با استفاده از رابطه بین $feature$ ها یا X ها متغیر y را پیش بینی می کنیم.

③ هدف اصلی از $Scaling$ این است که مطمئن شویم هر $feature$ سهم مساوی در تصمیم گیری مدل دارد، این در حالی است که مدل سعی در پیش بینی y دارد و y نقشی در یادگیری ندارد.

④ ممکن است این مورد در برخی از مسائل خاص و تحت شرایطی رخ دهد اما غیر معمول است.

