Starting of Gradient descent :→

1) Batch
2) Stochastic
3) Mini Batch

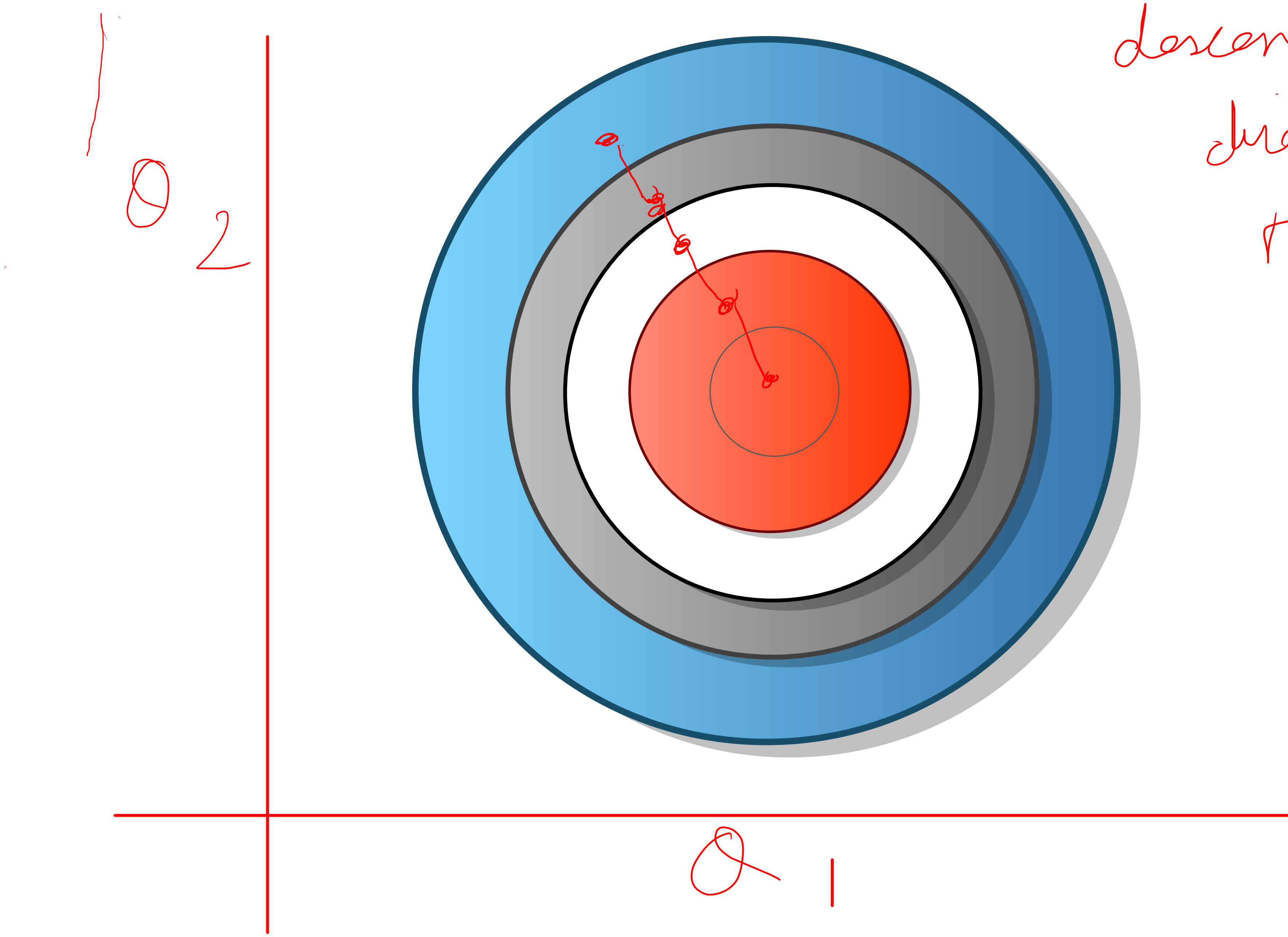
Why we use MSE as our cost function in Linear regression.

As it is a Convex function (which means that it has only 1 minima)

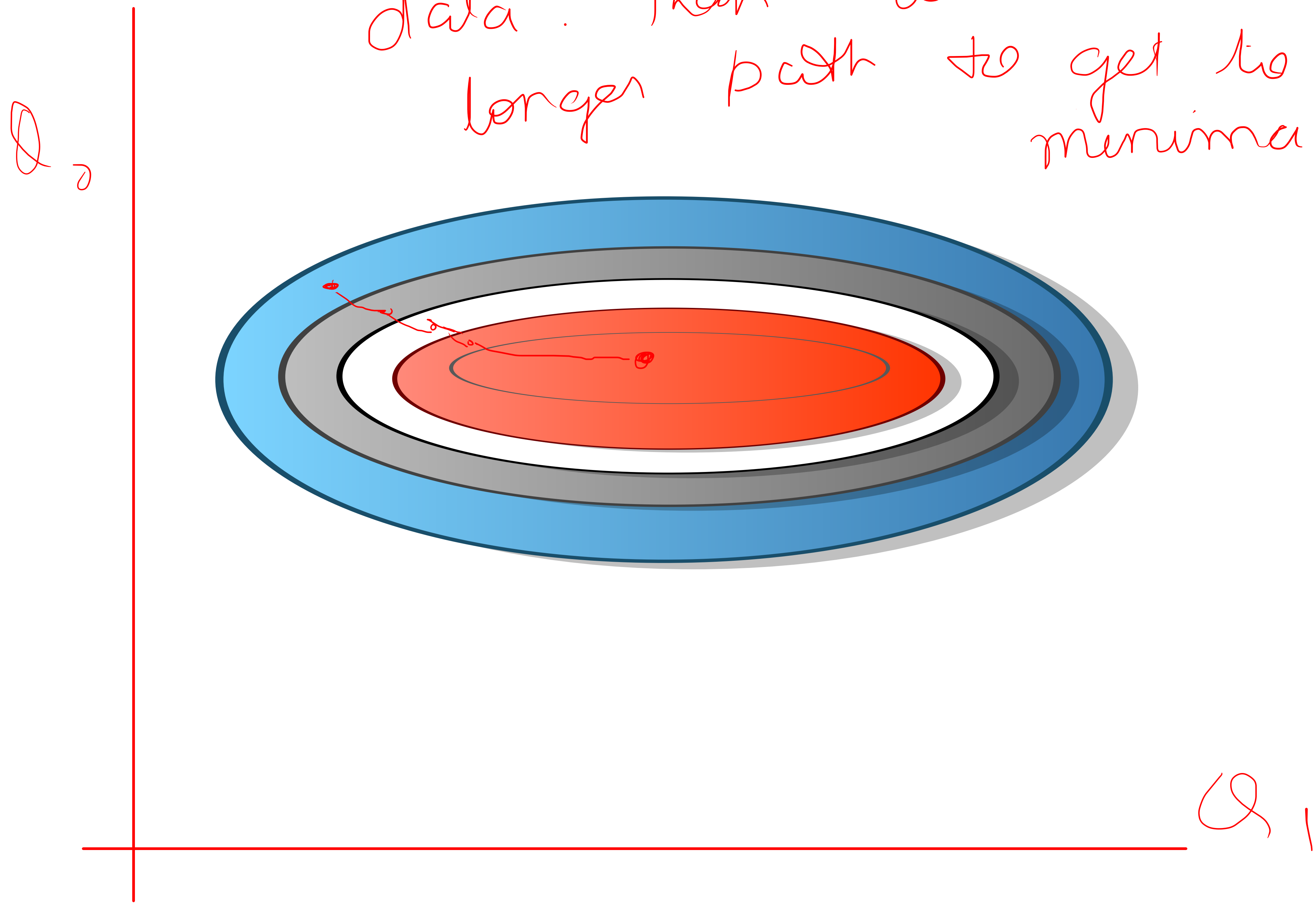
<u>V. V. Imp:</u>

While using Gradient descent our data must be scaled.

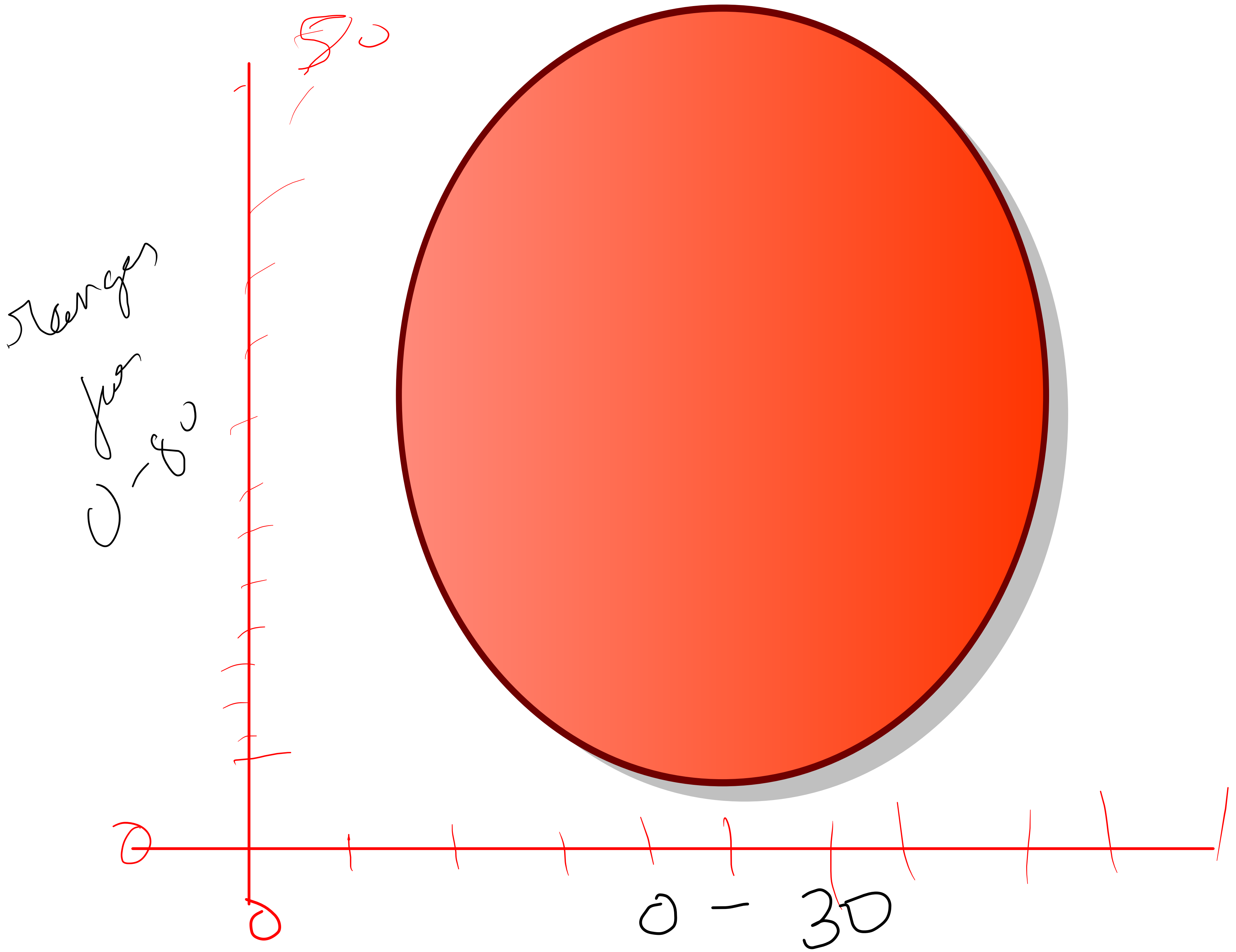
If the features are having same scale. The gradient descent will directly travel toward minima



If we are not scaling the data. Then we will travel longer path to get to minima

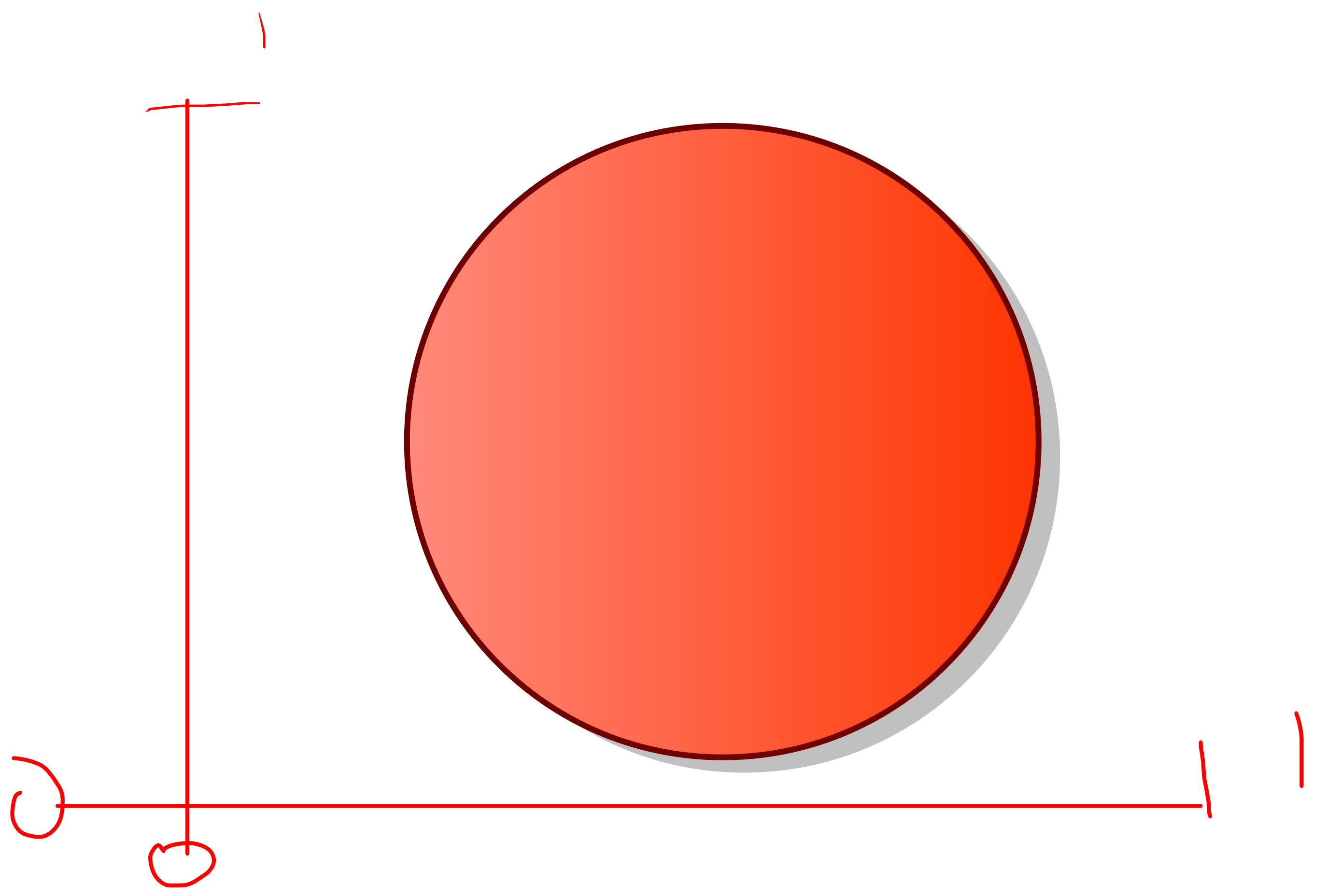


Age of Student	Class of Student
6	1
16	10
30	18
50	20
70. 80.	Phd



50
Ranges for 0-80
0
0
0 - 30

What if we put any b/w 0 – 1 & class also b/w 0 & 1. Then



Q) How do we scale the data?

Ans) There are 2 scaling methods.

1) Standard scaling

2) Min - max scaling.

Standard Scaling :-

Properties :- It scales the data such that $\boxed{\text{mean is zero}}$.

2) All the data lies b/w 1 standard deviation. (b/w 1σ)

$$x_{i_{new}} = \frac{x_i - \mu}{\sigma}$$

{ μ = mean

σ = standard deviation)

$$x_{i\,new} = \frac{x_i - \mu}{\sigma}$$

Where $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}$$

Also called :- Standardization.

Min - max Scaling :-

Also called Normalization.

$$x_{i\,new} = \frac{x_i - min}{max - min}$$

The $x_{i\,new}$ will always be b/w 0 & 1

Training a model means to find the best possible parameters. To minimize cost function like MSE. If we have more features. Than algorithm will have to find more parameters and the search becomes complex.

1) Batch Gradient Descent.

2) Stochastic Gradient descent.

3) Mini-Batch. " "

Partial Derivative -

1) We initialize all parameters randomly.

2) Then we change 1 weight while keeping all others fixed & calculate change in cost function.

$$\frac{\partial}{\partial \theta_j} (MSE) = \frac{2}{m} \left(\theta^T \cdot x^{(i)} - y^{(i)}\right) * x_j^{(i)}$$

$$\nabla_\theta MSE(\theta) = \begin{pmatrix} \frac{\partial}{\partial \theta_0} MSE(\theta) \\ \frac{\partial}{\partial \theta_1} MSE(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} MSE(\theta) \end{pmatrix} = \frac{2}{m} X^T \cdot (X \cdot \theta - y)$$

$$\nabla_{\theta} MSE(\theta) = \frac{2}{m} X^T (X \cdot \theta - y)$$

1) Batch gradient descent:
will load all the data in every step.

So Batch gradient descent is very slow on large datasets.

But in

$$\theta^{\text{next step}} = \theta - \eta \nabla_{\theta} MSE(\theta)$$

Learning rate.