ML

25/1/25

Given a dataset for performing linear regression with one variable.

Example:- Used car price prediction

	Features		o/p
I/p → Miles Driven	Engine Capacity		Price

But for simplicity purpose we are taking only one i/p.

Miles Driven	Price
1000	2.5
2520	3.6
⋮	⋮

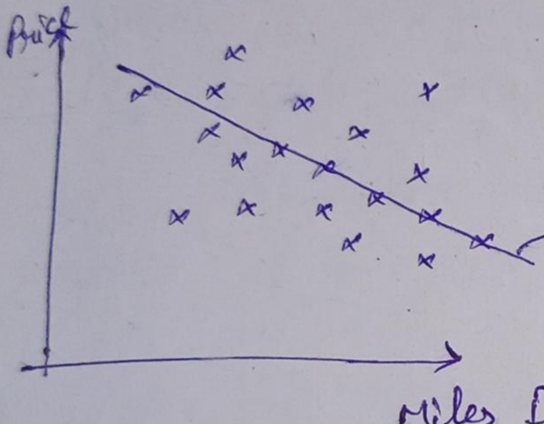
let, 2D matrix

(1000, 2.5)

(2520, 3.6)

$y = mx + c$

$= \theta_1 x + \theta_0$



→ hypothesis

$h_\theta(x) = \underbrace{\theta_0}_{\text{Find}} + \underbrace{\theta_1}_{\text{Find}} x$

CSV (Comma Separate Vector) can store huge no. of datas in tabular form.

25/1/25

1) What is machine learning?

2) What is learning?

3) What are the different types of m/l? → Supervised Learning, → Unsupervised Learning, → Semisupervised learning

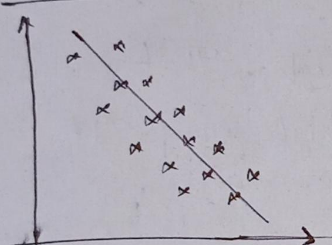
4) What is Supervised Learning?

5) What type of problem we can solve using Supervised learning?

↓ Regression ↓ Classification

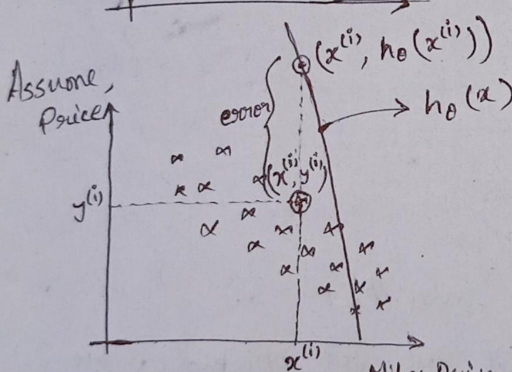
6) What is Regression problem explain with a real-life example?

- Linear Regression with one variable



$$h_\theta(x) = \theta_0 + \theta_1 x$$

Initially choose θ_0 and θ_1 randomly.



→ $h_\theta(x) = \theta_0 + \theta_1 x$

$$\text{error} = (h_\theta(x^{(i)}) - y^{(i)})^2$$

~~We only need +ve values~~

The distance can be -ve but we only need magnitude, but we can't do mod(| |). Because, for later we have to do derivative but mod can make problem at then. So, we can do square.

Now, assume there are m data samples.

So, total no. of errors $= \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2}_{\text{mean squared error (MSE)}}$ → Loss function

Minimize Error

$$\underset{\theta_0, \theta_1}{\text{Minimize}}\ J(\theta_0, \theta_1)$$

$$J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

When we have more than one variable then we need to do partial derivative.

$y = mx + c$ $\Downarrow$ slope

$$m = \frac{dy}{dx}$$

$$\frac{d\,J(\theta_0, \theta_1)}{d\theta_0} \qquad \frac{d\,J(\theta_0, \theta_2)}{d\theta_1}$$

Updating the value of θ_0, θ_1 !

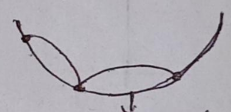
$$\theta_0 = \theta_0 - \alpha \underbrace{\frac{dJ(\theta_0, \theta_1)}{d\theta_0}}_{\text{gradiant}}$$

$$\theta_1 = \theta_1 - \alpha \cdot \frac{dJ(\theta_0, \theta_1)}{d\theta_1}$$

This is called Gradiant Decent Algorithm for learning.

α = Learning rate

If $\alpha = 1$ then it means α is too high. That means it is seeing the slope and making a high jump. It can make problem.



reqd value

Steps :

1. (i) Take Random points
2. (ii) Calculate errors
3. (iii) Calculate gradiants
4. (iv) Calculate α - value
5. (v) Calculate new value for θ_0 and θ_1
6. (vi) Repeat from step (i) until we get optimal value

Performing Partial Derivative:

For θ_0/

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 \qquad h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

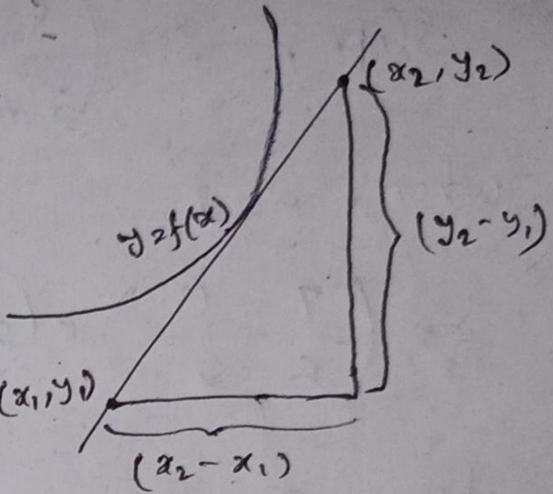
$$\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \frac{d}{d\theta_0} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} \frac{d(\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2}{d(\theta_0 + \theta_1 x^{(i)} - y^{(i)})} \times \frac{d(\theta_0 + \theta_1 x^{(i)} - y^{(i)})}{d\theta_0}$$

$$= \frac{1}{2m} \sum_{i=1}^{m} 2\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)(1 + 0 - 0)$$

$$= \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})$$

$$\boxed{\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})}$$

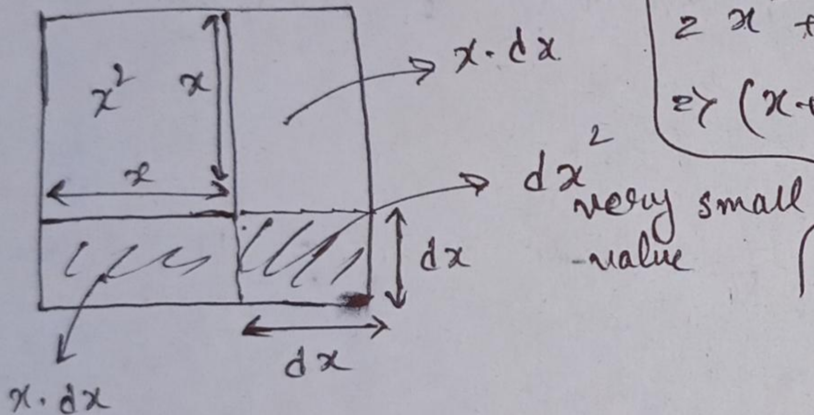


Now, for θ_1

$$\frac{d J(\theta_0, \theta_1)}{d\theta_1} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

$$(x + dx)^2$$
$$= x^2 + x\,dx + x\,dx + dx^2$$
$$\Rightarrow (x+dx)^2 - x^2 = 2x\,dx$$
$$\Rightarrow \frac{(x+dx)^2 - x^2}{dx} = 2x$$

$$\tan\theta = \frac{\text{height}}{\text{base}} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{dy}{dx}$$



30/1/2025

ML

Total Error $\rightarrow \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Avg. err

Mean Square Error $\rightarrow \boxed{\frac{1}{m}\sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2} \rightarrow$ Loss function

(*) Gradiant Descent Algorithm

Linear Regression with multiple features:

Input / output

Miles Driven (x_1)	Engine Capacity (x_2)	Fuel Type (x_3)	Price (y)

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

Fix $x_1 = 1$

$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$

$= \theta^T x$

$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$

$\underset{(1\times n)}{\begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix}} \underset{(n\times 1)}{\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}}$

Repeat Control Convergent:

$$\theta_j = \theta_j - \alpha \frac{d}{d\theta_j} J(\theta) \qquad J = 0 \text{ to } n$$

ONE HOT ENCODING:

- Non-numerical Feature $\xrightarrow{\text{to}}$ Vector

$$\begin{array}{cccc} C & P & D & E \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{array}$$

$[0 \; 0 \; 0 \; 1]$

$[0 \; 1 \; 0 \; 0]$

Fuel Type

→ Convert to

C	P	D	E

Normalization of Features: / Feature Scaling

[0 - 1]

min (age)

max (age)

Age	Salary
31	80000
32	80001
40	240000

$$\frac{x - min}{max} \rightarrow [0 - 1]$$

ML

Miles Driven	Engine Capacity	Year	No. of Pre owners	Fuel Types
2000	[-1 to 1]	[-1 to 1]	[-1 to 1]	CNG Des. Pet. Bat.

C	D	P	B
0	1	0	0
1	0	0	0

$\square \leq 0.5 \longrightarrow$ Class 0

$\square > 0.5 \longrightarrow$ Class 1

0.5 is threshold value. If the probability value is 0.5 then the machine will think that there has 50% chances for both.

Qn:) What is the role of the sigmoid / logistic fn to determine the class?

- Cost fn and loss fn are same.

Regularization : Overfitting Problem

Under fit, Perfect fit, Overfit