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
⋮	⋮

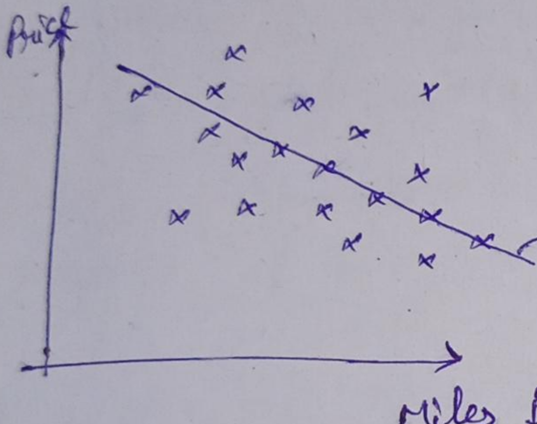
let, 2D matrix

(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?

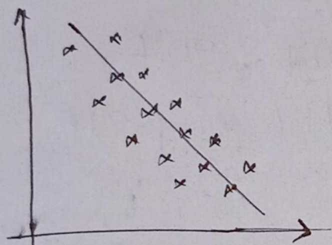
3) What are the different types of m/l? → Supervised Learning → Unsupervised Learning → Semisupervised learning

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?

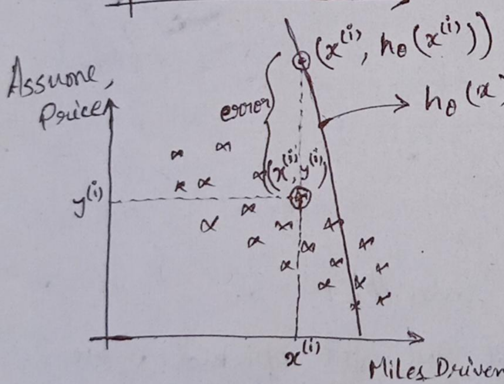
- Linear Regression with one variable



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

Initially choose θ_0 and θ_1 randomly.

Assume,



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

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

~~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 ~~make~~ can make problem at then. So, we can do square.

Now, assume there are m data samples.

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

Minimize Error

Minimize $J(\theta_0, \theta_1)$
θ_0, θ_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 $\quad m = \frac{dy}{dx}$

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

Updating the value of θ_0, θ_1 :

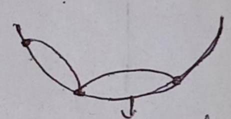
$$\theta_0 = \theta_0 - \alpha \underbrace{\frac{d\,J(\theta_0, \theta_1)}{d\theta_0}}_{\text{gradiant}}$$

$$\theta_1 = \theta_1 - \alpha . \frac{d\,J(\theta_0, \theta_1)}{d\theta_1}$$

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.



Steps :

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

Performing Partial Derivative:

For θ_0

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

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

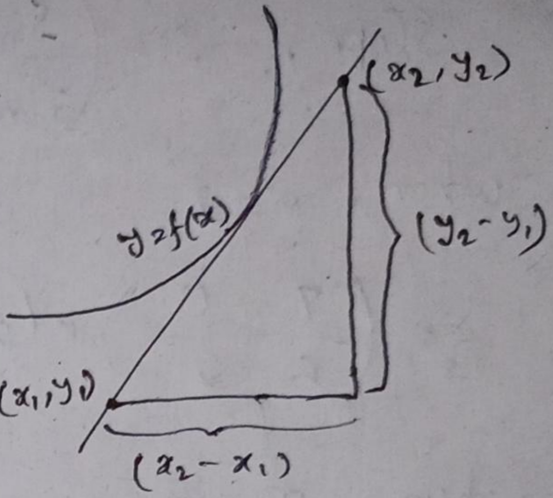
$$\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \frac{d}{d\theta_0} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

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

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

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

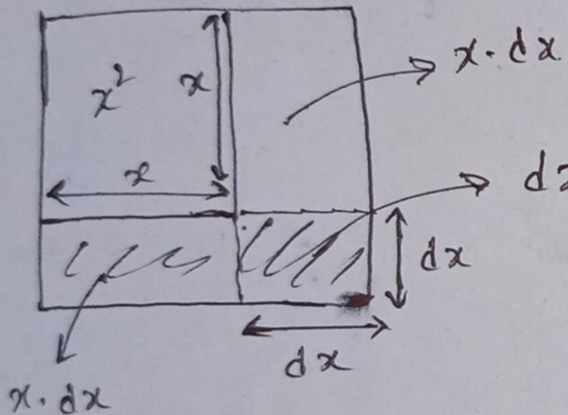
$$\boxed{\frac{d}{d\theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)}$$



Now, for θ_1

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

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



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

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:

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

$\underbrace{\text{Miles Driven, Engine Capacity, Fuel Type}}_{\text{Input}}$; Price $\leftarrow$ output

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

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

Fix $x_1 = 1$

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

$= \theta^T x$

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

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

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

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

Fuel Type

$\rightarrow$ 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

ML

8/2/25

Classification using Decision-Tree

Feautes

Age, Income, Student? Creadit Rating

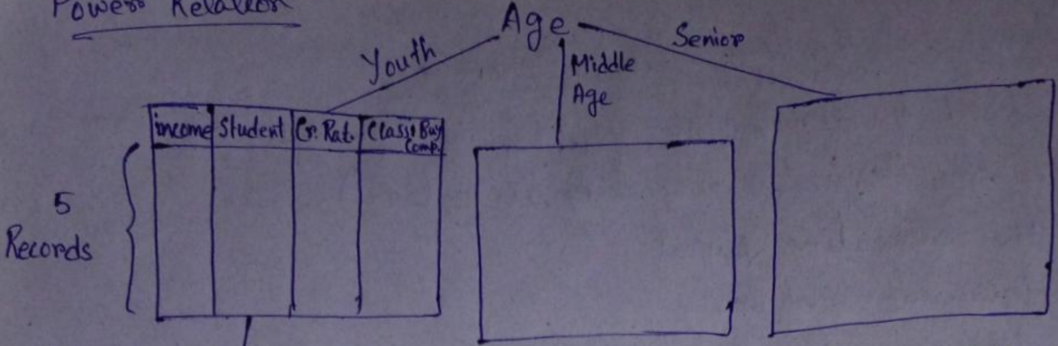
The prediction is basically buy computer or not.

Example :

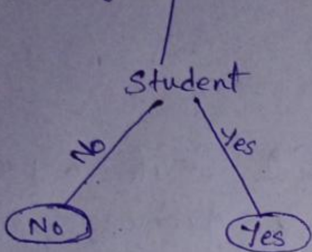
Age	Income	Student	Credit Rating	~~By~~ buy Comp.
Youth	High	No	Fare	No

Age
- Senior
- Middle Age
- Youth

Power Relation



Biased data / So, again splitting



① if age = 'middle age' then
Buy PC = Yes

② if age = 'youth' and
student = 'no'
Buy PC = No

If after splitting all the attributes, there left no attribute then go for majority voting. If the majority is 'yes' then all will be 'yes' and vice-versa.

Attribute Selection for Splitting

Information gain / Entropy

$Info(D) = - \sum_{i=1}^{m} p_i \log_2(p_i)$ ∴ m = no. of classes

- The unit of entropy is bits.

For our case,

m = 2 → Buy computer = Yes &

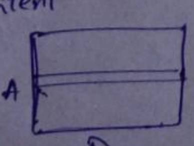
p_1 → " " = No

p_2

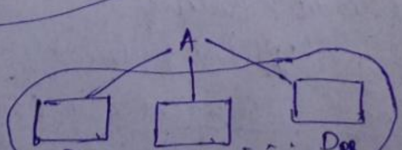
$IC \propto \frac{1}{P(A)}$

↓

Information Content



Info(D) |D|



$Info(D_1)$ $Info(D_2)$ .... $Info(D_v)$

$|D_1|$ $\boxed{Info_A(D)}$

$$\text{Info}_A(D) = \frac{|D_1|}{|D|}\text{Info}(D_1) + \frac{|D_2|}{|D|}\text{Info}(D_2) + \dots + \frac{|D_v|}{|D|}\text{Info}(D_v)$$

$$= \sum_{j=1}^{v} \frac{|D_j|}{|D|}\text{Info}(D_j)$$

$$\boxed{\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)}$$

$$\text{Gain}(\text{Age}) = \text{Info}(D) - \text{Info}_{Age}(D)$$

$$\text{Info}(D) = -\sum_{i=1}^{2}(p_i \times \log_2(p_i))$$

$$= -p_1\log_2(D_1) - p_2\log_2(D_2)$$

$$= -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.94$$

$$\cancel{= -0.64\log_2(0.64) - 0.36\log_2(0.36)}$$

$$\cancel{= (-0.64)\times(-0.15) - (0.36)\times(-1.02)}$$

$$\cancel{= 0.288 + 0.3672}$$

$$\cancel{= 0.28}$$

$$\cancel{\text{Info}_{Age}(D) = \frac{5}{14}\left[-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right] + \frac{5}{14}\left[-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right] + \frac{4}{14}\left\{0 - \frac{4}{4}\log_2\left(\frac{4}{4}\right)\right]}$$

$$\cancel{= \frac{10}{14}(0.97 + 0.97)}$$

$$\cancel{= 0.69285}$$

$$\cancel{\text{gain(age)} = 0.97 - 0.69 = 0.2465}$$

$$\text{Info}_{Age}(D) = \sum_{j=1}^{v}\frac{|D_j|}{|D|}\text{Info}(D_j)$$

$$= \sum_{j=1}^{3}\frac{|D_j|}{|D|}\text{Info}(D_j)$$

$$= \frac{|D_1|}{|D|}\text{Info}(D_1) + \frac{|D_2|}{|D|}\text{Info}(D_2) + \frac{|D_3|}{|D|}\text{Info}(D_3)$$

$$= \frac{5}{14}\left[-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log\frac{3}{5}\right] + \frac{4}{14}\left[-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right]$$

$$+ \frac{5}{14}\left[-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right]$$

$$= 0.6928$$

$$\text{gain(age)} = 0.97 - 0.69 = 0.28$$

Gain of Income = 0.029 bits
Gain of Student = 0.151 bits
Credit Rating = 0.048 bits

ML

13/2/2025

Entropy

$$\text{Info}(D) = -\sum_{i=1}^{m} P_i \log_2(P_i)$$

m: no. of classes

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \cdot \text{Info}(D_j)$$

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

$$\text{Split Info}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|}$$

According to Income attribute

$$= -\sum_{j=1}^{3} \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|}$$

$$= -\frac{|D_1|}{|D|} \times \log_2 \frac{|D_1|}{|D|} - \frac{|D_2|}{|D|} \times \log_2 \frac{|D_2|}{|D|} - \frac{|D_3|}{|D|} \times \log_2 \frac{|D_3|}{|D|}$$

$$= -\frac{4}{14} \log_2 \frac{4}{14} - \frac{6}{14} \log_2 \frac{6}{14} - \frac{4}{14} \log_2 \frac{4}{14}$$

$$\cancel{= 1.5564} = 1.5564$$

$$\cancel{= 0.155465 + 0.15770 + 0.155345}$$

$$\text{Gain Ratio}(A) = \frac{\text{Gain}(A)}{\text{Split Info}(A)}$$

$$= \frac{0.28}{1.5564} = 0.1799 \text{ Ans}$$

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} p_i^2$$

$$\text{Gain}_A(D) = \frac{|D_1|}{|D|} \text{Gini}(D_1) + \frac{|D_2|}{|D|} \text{Gini}(D_2)$$

$$\Delta \text{Gini}(A) = \text{Gini}(D) - \text{Gini}_A(D)$$

$$\sum_{i=1}^{m} \underbrace{y^{(i)}}_{\text{Actual class}} \log_2 \underbrace{h_\theta(x^{(i)})}_{\text{Predicted class}}$$

Binary cross entropy loss

$$\Big(\underbrace{\sum_{i=1}^{m} y^{(i)} \log_2 h_\theta(x^{(i)})}_{\text{Binary cross entropy for +ve class}} + \underbrace{\sum_{i=1}^{m} (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))}_{\text{Binary class entropy for -ve class}}\Big)$$

ML

Receiver Operating Characteristics (ROC) Curve

if $g(\theta^T x) \geq 0.5 \Rightarrow$ Class 1

$g(\theta^T x) < 0.5 \Rightarrow$ Class 0

$h_\theta(x) = g(\theta^T x)$

$0 \leq h_\theta(x) \leq 1$

Consider T_1 as thresold value

$T_1 \longrightarrow S_1$ → TPR (True Positive Rate)

→ FPR (False Positive Rate)

ML

20/2/2025

SVM

Class 1 ≥ 1

Class 0 < -1

Support Vector Machine :

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \longrightarrow 2D$$

(1) $$\left(\sum_{i=1}^{n} \left(x_i^{(2)} - x_i^{(1)}\right)^2 \right)^{1/2} \longrightarrow nD \longrightarrow \text{Eucledian Distance/(L2)}$$

(2) Manhattan Distance/(L1) Distance/City Block Distance

$$L_1 = |x_2 - x_1| + |y_2 - y_1| \longrightarrow 2D$$

nD $\rightarrow$

$$L_1 = |x_1^{(2)} - x_1^{(1)}| + |x_2^{(2)} - x_1^{(1)}| + \dots + |x_n^{(2)} - x_n^{(1)}|$$

$$= \sum_{i=1}^{n} |x_i^{(2)} - x_i^{(1)}|$$

KNN

Lazy Learner $\rightarrow$ no learning in historical data, but learn in prediction data.

Solve Classification problem using KNN:

Solve regression problem using KNN: