⇒ Cross validation :

* If dataset size is small .

⇒ Model :

* System that perform task .
* It is mathematical Representation of System .

Model : (x, y)
↓
True label

$X \rightarrow \boxed{f(x)} \rightarrow \hat{y}$
↓ predicated label

⇒ $\boxed{\hat{y} = f(x)}$ This relation is called as "model"
↓ model .

⇒ Regression :-

① Simple linear Regression .

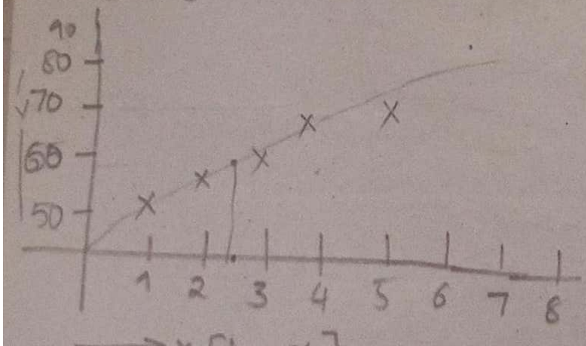
* predicts continuos outcomes .

X Y —— Actual

S.NO	Hours	Marks
1.	1	50
2.	2	55
3.	3	60
4.	4	65
5.	5	70
6.	2.5	?

(rows 1–5) Train
(row 6) J. validation

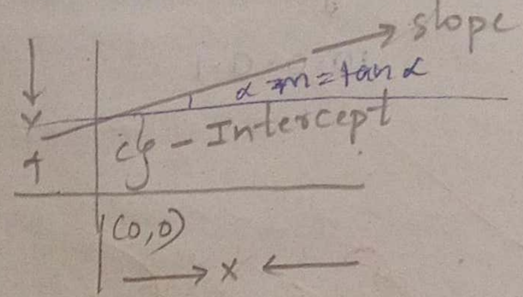
↓
By using we draw Scatter plot to table



We want 2.5 then the line will give the data

⇒ line is the model that capture r/s b/w x & Y

$X \rightarrow \boxed{f(x)} \rightarrow \hat{y}$



Formula for slope :-

$y = mx + c \rightarrow$ Intercept
↙ ↓ ↘
output slope Input

m - slope
C - Intercept

⇒ Data processing :

① Missing value
↳ Anything is missing it will replace .

② Encoding :
Converting the values
Categorical — nominal
— Ordinal

colors (nominal)	T-shirt size — ordinal
Red	L
Blue	M
Green	S
red	XL

↓
we can sort the data

XL > L > M > S

③ Feature Scaling

⇒ Converting the higher range to Range in Small .

Height	Weight (0-1)
50	0.5
60	0.6
70	0.7
80	0.8
100	[illegible]

④ Train val - train Split

one hot encoding · If it categorial and nominal.

Red - 0
Green - 1
Blue - 2

	R	G	B
R	1	0	0
G	0	1	0
B	0	0	1

⇒ preprocessing

① missing value
- fillna()
- Imputer()

② Encoding
- map()
- labelencoder
- oneHotEncoder

③ Scaling
- Standard Scaler
- MinMax Scaler

④ Split

Missing:
- fillna()
- mean
- median
- mode

Imputer

dropna()
↓
If no element is required then we drop that.

⇒ Encoding.
- Dictonary, map
- label encoder
- one hot encoder (nominal)

⇒ Standarization:

$z = \frac{x-\mu}{\sigma}$ — mean, — variance → normal distribution

mean = 0
variane = 1

↓
standard normal distribution

Why mean = 0

Ex:

x	x-μ
2	-4
4	-2
6	0
8	2
10	4
x=6	x=0

↓
mean

⇒ Data :

no. of study (x)	Marks (y)	Model ($\hat{y}$)	$E_i = y-\hat{y}$
1	50	15	
2	55	20	
3	65	25	
4	70	30	
5	80	35	

⇒ Min - Max Normalization.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

min = 2
Max = 8

x	$x_{min} - x_{max}$
2	$\frac{2-2}{8} = 0$
4	$\frac{4-2}{6} = 0.25$
8	$\frac{8-2}{6} = 0.75$
10	$\frac{10-2}{6} = 1$

Model

Data:

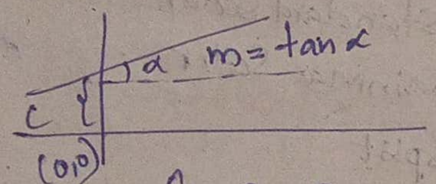
$40 + 2x$

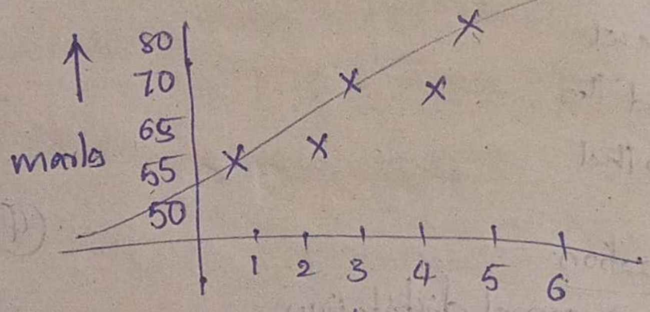
$\hat{y} = 41.5 + 7.5x$

No. of study (x)	Marks (y)	Model (Y)	$E_i = y - \hat{y}$	$\hat{y}$	E_i $(y-\bar{y})$	x_iy_i	Model-3 $\hat{y}$	E_i
1	50	15	35	42	8 (50-52)	50	49	1
2	55	20	35	44	11	110	56.5	1.5
3	65	25	40	46	19	195	64	1
4	70	30	40	48	22	280	71.5	1.5
5	80	35	45	50	30	400	79	1
x_i = 15	320					715		Totals 6

$\hat{y} = 10 + 5x$ — total error = 195

$\hat{y} = 40 + 2x$ — " " = 90





$\hat{y} = \theta_0 + \theta_1 x$ → every neural n/w contains this formula.

parameters θ_0, θ_1

Error $= y - \hat{y}$

possible error:

Error $= |y_i - \hat{y}_i|$

Error $= (y_i - y_i)^2$ — Amplication the error.

Training — Finding the parameters that give the minimum error

Find θ_0 and θ_1 Standard error is minimum :-

① using formula:

↓ By similar Linear Regression

① ML has Model → parameters

② Error

③ Train - find parameters such that errors minimized

$$\boxed{\begin{aligned}\theta_0 &= \bar{y} - \theta_1\bar{x} \\ \theta_1 &= \frac{n\sum x_iy_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}\end{aligned}}$$

$$= \frac{5(715) - (15)(320)}{5(3)^2 - (15)^2} \Rightarrow \frac{3575 - 4800}{45 - 225} = \frac{-1225}{-180}$$

6.80

$\theta_0 = 41.5$

$\theta_1 = 7.5$

$\hat{y} = 41.5 + 7.5x$.

⇒ Regression:-

$Y \rightarrow$ continuos variable.

Independent	Y
	50
1	55
2	65
3	70
4	80
5	

$Y \rightarrow$ Dependent

Error:

$|y_i - \hat{y}_i|$

$(y_i - \hat{y}_i)^2$

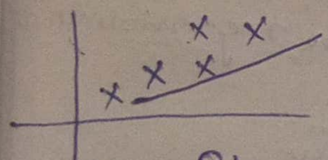
$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

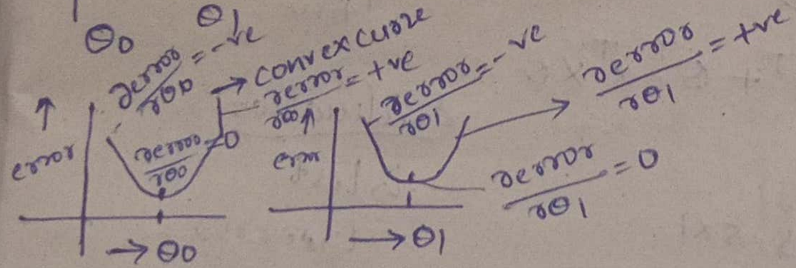
⇒ Formula:

$\theta_0 = y - \theta_1 \bar{x}$

$\theta_1 = \dfrac{n\Sigma xy \cdot \Sigma x - \Sigma y}{n\Sigma x^2 - (\Sigma x)^2}$

⇒ Minimizing the error:





$\boxed{\frac{dy}{dx}}$ ⇒ derivative

↓

launch to minimize the error.

$\Rightarrow \hat{y} = \theta_0 + \theta_1 x$

find θ_0 and θ_1 such that

$\frac{\partial error}{\partial \theta_0} = 0$ and $\frac{\partial error}{\partial \theta_1} = 0$

$\boxed{SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$ — ①

Substitute $\hat{y}$ in equ — ①

$= \Sigma(y_i - \theta_0 - \theta_1 x_i)^2$

$\frac{\partial SSE}{\partial \theta_0} = 2\Sigma(y_i - \theta_0 - \theta_1 x_i)(-1)$

$= -2\Sigma(y_i - \theta_0 - \theta_1 x_i)$

$-2\Sigma(y_i - \theta_0 - \theta_1 x_i) = 0$

$\Rightarrow \Sigma(y_i - \theta_0 - \theta_1 x_i) = 0$

$\Rightarrow \Sigma y_i - \Sigma\theta_0 - \Sigma\theta_1 x_i = 0$

$\Rightarrow \Sigma y_i - n\theta_0 - \theta_1 \Sigma x_i = 0$

$\Rightarrow n\theta_0 = \Sigma y_i - \theta_1 \Sigma x_i$

$\theta_0 = \dfrac{\Sigma y_i - \theta_1 \Sigma x_i}{n}$

$= \dfrac{\Sigma y_i}{n} - \theta_1 \dfrac{\Sigma x_i}{n}$

$\boxed{\theta_0 = \bar{y} - \theta_1 \bar{x}}$ — ①

$\frac{\partial SSE}{\partial \theta_1} = 2\Sigma(y_i - \theta_0 - \theta_1 x_i)(-x_i)$

$= -2\Sigma x_i(y_i - \theta_0 - \theta_1 x_i)$

$= -2\Sigma x_i(y_i - \theta_0 - \theta_1 x_i) = 0$

$\Rightarrow \Sigma x_i(y_i - \theta_0 - \theta_1 x_i) = 0$

$\Rightarrow \Sigma x_i y_i - \theta_0 \Sigma x_i - \theta_1 \Sigma x_i^2 = 0$

$\Rightarrow \Sigma x_i y_i - \left(\dfrac{\Sigma y_i - \theta_1 \Sigma x_i}{n}\right)\Sigma x_i - \theta_1 \Sigma x_i^2 = 0$

$\Rightarrow n\Sigma x_i y_i - \Sigma y_i \Sigma x_i + \theta_1 (\Sigma x_i)^2 - n\theta_1 \Sigma x_i^2 = 0$

$\Rightarrow n\Sigma x_i y_i - \Sigma x_i \Sigma y_i = \theta_1 (n\Sigma x_i^2 - (\Sigma x_i)^2)$

$$\boxed{\Rightarrow \theta_1 = \frac{n\Sigma x_i y_i - \Sigma x_i \Sigma y_i}{n\Sigma x_i^2 - (\Sigma x_i)^2}} \quad — ②$$

⇒ **Normal form Equation:**

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}_{2\times1}$$

$$y = \begin{bmatrix} 50 \\ 55 \\ 65 \\ 70 \\ 80 \end{bmatrix}$$

$$x = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix}_{5\times2} \Rightarrow \hat{y} = \theta_0 \cdot 1 + \theta_1 \cdot x$$

$1 \times \theta_0 + 1 \times \theta_1 1$

$1 \times \theta_0 + 2 \times \theta_1$

$$\boxed{\theta = (x^T x)^{-1} \cdot x^T y}$$

$$\frac{\underset{2\times2}{2\times5 \cdot 5\times2} \quad \underset{2\times1}{2\times5 \cdot 5\times1}}{2\times1}$$

$$\Rightarrow SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \quad \hat{y} = x\theta$$

$$= (y - \hat{y})^T \cdot (y - \hat{y})$$

$$= (y - x\theta)^T \cdot (y - x\theta)$$

$$= y^T y - y^T \cdot x\theta - \theta^T x^T + \theta^T \cdot x^T x \theta$$

$$= y^T x \theta = \theta^T x^T y$$

$1\times5 \cdot 5\times2 \cdot 2\times1$ | $1\times2 \cdot 2\times5 \cdot 5\times1$

$= 1\times1$ | $= 1\times1$

$$= \underbrace{y^T y - 2\theta^T x^T y}_{a^2 - 2ab} + \underbrace{\theta^T x^T x \theta}_{b^2} \quad (a+b)^2 =$$

$$\frac{\partial SSE}{\partial \theta} = 0 - 2 \cdot x^T y + 2 \cdot x^T \cdot x \cdot \theta$$

$$= 2 \cdot x^T \cdot x\theta = 2 \cdot x^T y$$

$$= \cancel{2} x^T x \theta = \cancel{2} \cdot x^T y$$

$$= \theta = \frac{x^T y}{x^T x}$$

$$\boxed{\theta = (x^T x)^{-1} \cdot x^T y}$$

```
import numpy as np
x = np.array([[1,1], [1,2], [1,3],
              [1,4], [1,5]])
y = np.array([50, 55, 65, 70, 80])
theta = np.linalg.inv(x.T @ x)
        @ x.T @ y
print(theta)
```

#output:

$\theta_0 = 41.5$

$\theta_1 = 7.5$

⇒ **Regression:-**

⇒ **polynomial Regression,**

PLR: $\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2$ ↑ Polynomial feature

Non-linear regression.

Ex: temp / time

Evaluation
↳ test
↳ error
↓
→ MAE
→ MSE
→ RMSE
→ R^2
} metrics

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$$

⇒ Evaluation Metrices,

⇒ $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

⇒ $MSE = \frac{SSE}{n}$

⇒ $RMSE = \sqrt{MSE}$

In training case

In testing case we use evaluation

↳ R^2

↳ Adjusted R^2.

⇒ R^2 :-

variance explained by model

$$R^2 = \frac{\Sigma (y_i - \hat{y})^2}{\Sigma (y_i - \bar{y})^2}$$

↓ variance of data → how the data is distributed

1 → good model / perfect fit

⋮

0 → Bad model / not fitting.

Eg:

X (feature)	y	$\hat{y}$
	50	49
	55	56.5
	65	64
	70	71.5
	80	79

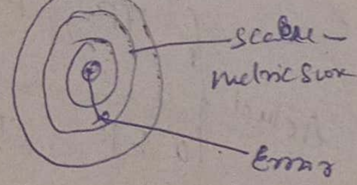
$R^2 = 0.986$

⇒ Adjusted R^2:

$$= 1 - \frac{(1-R^2)(n-1)}{n-k-1}$$

n = no. of data points = 5

k = no. of features = 1



⇒ Classification: Binary classification

Email → f(x) → Y = {spam, not spam}
(spam = 1, not spam = 0)

	feature X	y (Actual label)	$\hat{y}$ (predicted label)	
TN		0	0	✓
TP		1	1	✓
FN		1	0	✗
FP		0	1	✗
TN		0	0	✓
TP		1	1	✓
FP		0	1	✗
FN		1	0	✗
TN		0	0	✓

Accuracy = $\frac{5}{9}$

⇒ Confusion Matrix :-

Actual ↓ / predicted →	1	0
1	TP 2	FN 2
0	FP 2	TN 3

Spam and marked on Spam (TP)

spam but marked as not spam (FN)

TP = True positive (True +ve)

TN = True (-ve)

FN = False -ve

FP = False +ve

Not spam but marked as Spam (FP)

Not Spam and not marked as Spam (TN)

Eg-2:

feature → Cancer prediction → {yes, no}
(yes = 1, no = 0)

feature ↓ Dimension

		Predicted 1	Predicted 0
Actual	1	TP	FN
	0	FP	TN

FN → dangerous — Have cancer but not predicted

TN — no cancer and predicted no cancer.

FP — no cancer but predicted as Caner

TP — Have cancer and predicted correctly.

⇒ Evaluation Metrices : [classification].

X	Y	$\hat{Y}$
	0	0
	0	0
	1	0
	0	1
	1	1
	1	1
	0	1
	1	0
	0	0
	1	1

$\hat{Y}$ ← Binary classification
↳ Two labels
{1, 0}
↙ ↘
+ve −ve

Spam / Not Spam

Cancer / not cancer.

- Accuracy
- precision
- Recall
- F1 score.

⇒ confusion Matrix.

		1	0
Actual	1	3 TP	2 FN
	0	2 FP	TN 3

predicted — 1, 0

FN — All levels are positive. Type 2 Error.

TN — Actual −ve predicted −ve.

FP → Type 1 Error. predicted +ve

$$\text{Accuracy} = \frac{\text{Total correct prediction}}{\text{Total predication}} = \frac{6}{10} = \frac{3}{5} = 0.6 = 60\%$$

$$= \frac{TP + TN}{TP + TN + FP + FN} = \frac{6}{10} = 0.6 = 60\%$$

$$P = \frac{\text{No. of correct positive predictions}}{\text{No. of positive predictions}} = \frac{TP}{TP+FP} = \frac{3}{5} = 0.6\%.$$

$$R = \frac{\text{No. of correct +ve prediction}}{\text{No. of actual positive accus}} = \frac{TP}{TP+FN} = \frac{3}{5} = 0.6\%.$$

$$F1 = \frac{2 * P * R}{P+R} \Rightarrow \frac{2 * 0.6 * 0.6}{0.6+0.6} \Rightarrow 0.6.$$

Accuracy is near to 1 ⇒ Good model.
" " near to 0 ⇒ Bad Model.

⇒ **Multi class classification** :- (More than 2 levels)

Y	$\hat{Y}$
0	0
0	0
0	2
0	0
0	2
1	1
1	1
1	1
1	0
1	1
2	1
2	1
2	2
2	2
2	2

0 – Regular Email
1 – Social Email.
2 – Spam Email.

predicted

Actual	0	1	2
0	3	0	2
1	1	4	0
2	0	2	3

FN
TN
FP

$$\text{Accuracy} = \frac{10}{15} = 0.66.$$

Class 0 :- TP = 3, TN = 9, FP = 1, FN = 2

P = 0.75
R = 0.6
F1 = 0.66

Macro P = 0.67

$\cancel{\frac{P+R+F1}{3}}$ = .

Macro R = $\frac{2}{3}$ = 0.66

Macro F1 = $\frac{1.98}{3}$ = 0.66

class 1 :-

TP = 4, TN = 8 FP = 2 FN = 1

P = 0.66 R = 0.8 F1 = 0.72

Class 2 :

TP = 3 TN = 8 FP = 2 FN = 2

P = 0.6, R = 0.6, F1 = 0.6

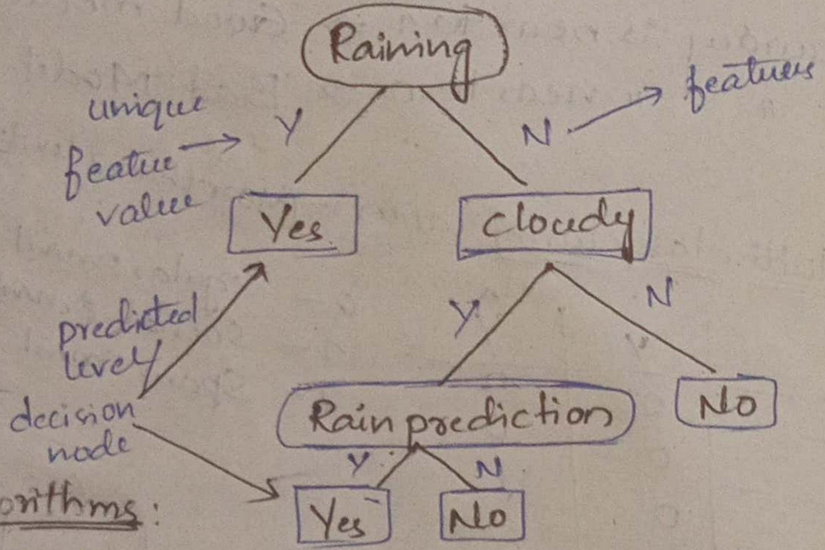
⇒ **Try:**

x	y	$\hat{y}$
	0	2
	0	2
	0	0
	0	0
	0	1
	1	1
	1	1
	1	2
	1	0
	1	1
	2	0
	2	0
	2	0
	2	2
	2	2

// code:

```
if raining
  then yes
else if cloudy
else if rain prediction
  then Yes
else No
```

Decision tree: (model)



⇒ **Classification Algorithms:**

① **Decision tree:**

- {x, y}
  ↓
  discrete
  {spam, Not Spam}

Cloudy, Weather prediction, raining → [Will you take umbrella] → Yes/No

⇒ ① **uncertainity and entropy:—**

① will the students wear uniform to

C1	C2	C3	C4
Y	Y	Y	Y
Y	Y	N	Y
Y	Y	N	N
Y	N	N	N

C1 → more certainity (entropy is less)

C4 ← less certinity ↓ entropy is more

Based on raining

cloudy	Weather prediction	raining	umbrella:	$\hat{y}$
N	N	N	N	N
Y	N	N	N	N
Y	N	N	Y	Y
Y	Y	Y	Y	Y

(cloudy, weather prediction, raining = x; umbrella = y)

Accuracy = 100 %

C1:—	C2:—	C3:—	C4:
More certinity (entropy is less) / uncertainity	$P(U) = 3/4$	$P(U) = 1/4$	$P(U) = 2/4 = 1/2$
$P(U) = 4/4 = 1$	$P(N,U) = 1/4$	$P(NU) = 3/4$	$P(N,U) = 2/4 = 1/2$
$P(N,U) = 0/4 = 0$	$E(C2) = 0.81$	$E(C3) = 0.81$	$E(C4) = 1$
$E(C1) = 0$			

⇒ **Entropy**: (If all become Yes it 0, if all become N then 1)

$$\Rightarrow -P_1 \log_2 P_1 - P_2 \log_2 P_2$$

$$\{P_1, P_2\} \Rightarrow \{U, NU\}$$

$$E(C1) = -1 \log_2 1 - 0 \log_2 0 = 0$$

$$E(C4) = -1/2 \log_2 1/2 - 1/2 \log_2 1/2 = 1 \rightarrow \left[\begin{array}{l} -\log_2 1/2 \\ = -(\log_2 1 - \log_2 2) \\ = (0-1) = 1 \end{array}\right]$$

$$E(C2) = -3/4 \log_2 3/4 - 1/4 \log_2 1/4 = 0.81$$

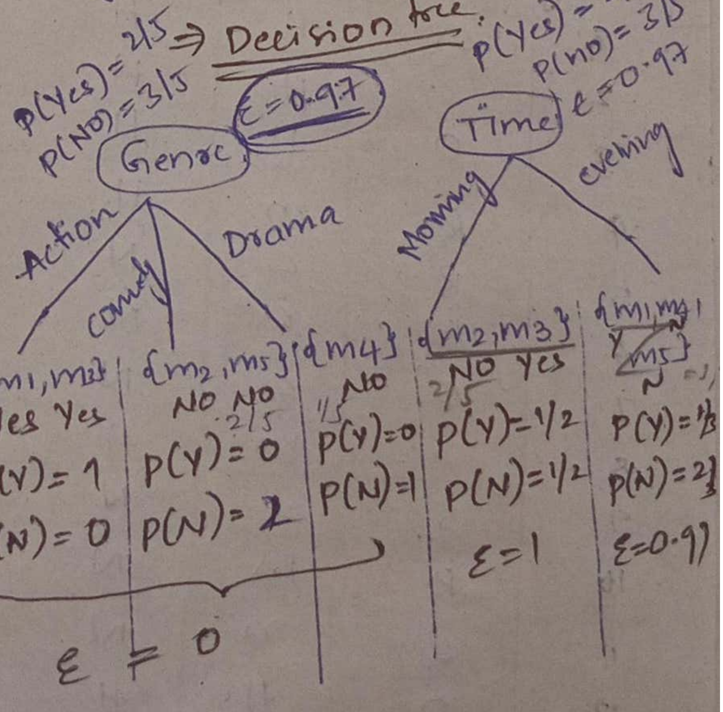
$$E(C3) = -3/4 \log_2 3/4 - 1/4 \log_2 1/4 = 0.81$$

⇒ uncertainity is more ⇒ Entropy is also more.

⇒ **Watch Movie**:

Movie	Genre	Time	Watch
M1	Action	Evg	Yes
M2	Comedy	mrng	No
M3	Action	mrng	Yes
M4	Drama	eveg	No
M5	Comedy	evg	No

⇒ Decision tree.



$$\text{Avg entropy}_{\text{in children}} = 2/5 * 0 + 2/5 * 0 + 1/5 * 0 = 0$$

$$\text{Genre} \Rightarrow \{Y: 2, N: 3\} \Rightarrow E(\text{root}) = -2/5 \log_2 2/5 - 3/5 \log_2 3/5$$

$$= 0.97$$

⇒ Information Gain :- (IG)

IG(Genre) = Entropy in root − Avg entropy in children

= 0.97 − 0 = 0.97 ✓

Time :-

Avg Entropy of children = 2/5 * 1 + 3/5 * 0.91

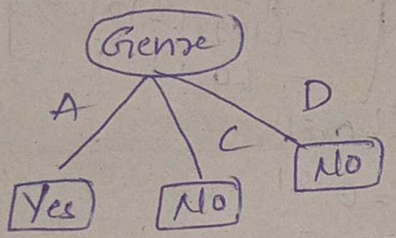
= 0.94 .

Information Gain

IG(time) = 0.97 − 0.94 = 0.03

Select the attribute with highest IG .

Final Decision tree .



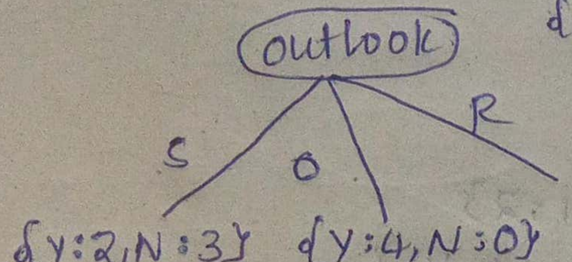
② Day	outlook	Temp	Humidity	wind	play Tennis .
1	S	H	H	W	N
2	S	H	H	S	N
3	O	H	H	W	Y
4	R	m	H	W	Y
5	R	C	N	W	Y
6	R	C	N	S	N
7	O	C	N	S	Y
8	S	m	H	W	N
9	S	C	N	W	Y
10	R	m	N	W	Y
11	S	m	N	S	Y
12	O	H	H	S	Y
13	O	H	N	W	Y
14	R	m	H	S	N
	Sunny overheat Rainy	Hot mid cool	High normal	Weak strong	Yes NO

Day = {O, C, H, S} will you play tennis ?

{Y:9, N:5}

$E(\text{total data}) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14}$

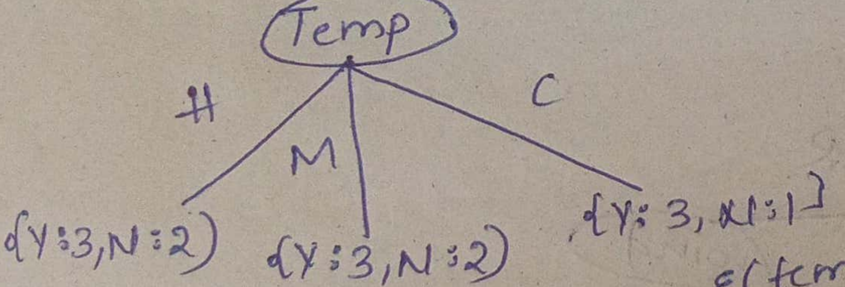
$= \underline{0.94}$



E(outlook S) E(outlook O) E(outlook R)
= 0.97 = 0 = 0.97

Avg = 5/14 * 0.97 + ~~4/14~~ * 0 + 5/14 * 0.97 = 0.692

IG(outlook) = 0.94 − 0.69 = 0.25

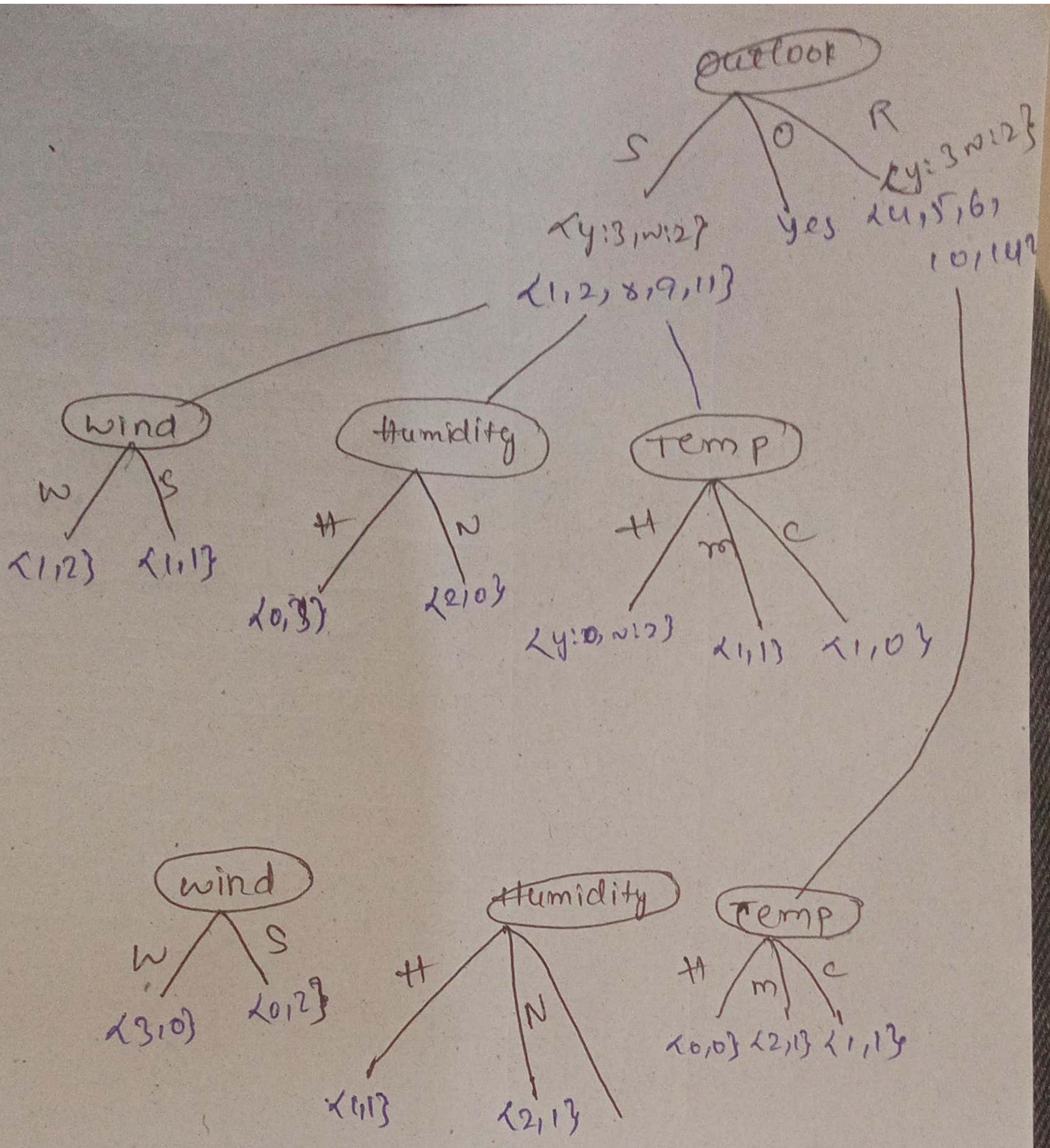


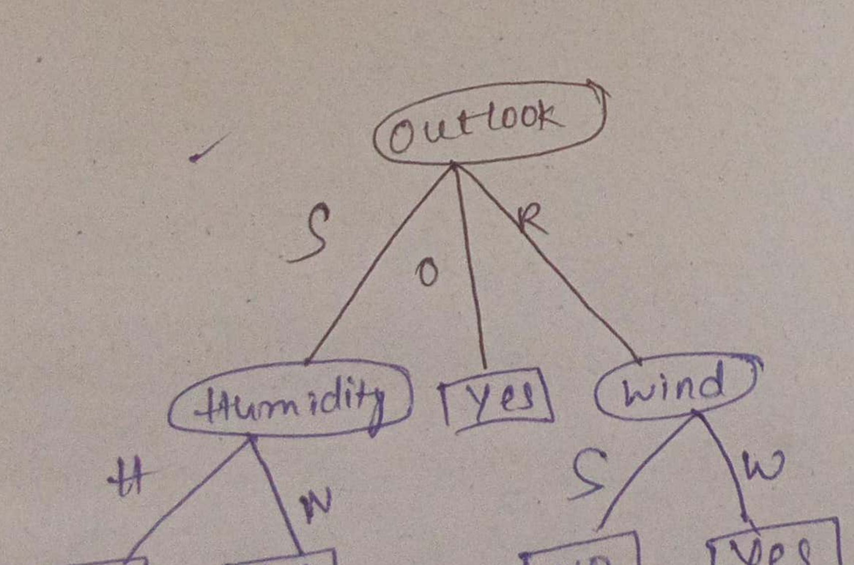
E(temp) = 0.97 E(temp m) = 0.97 E(temp c) = 0.81

Avg = 5/14 * 0.97 + 5/14 * 0.97 + 4/14 * 0.81 = 0.92

IG(temp) ⇒ 0.94 − 0.92 = 0.02

IG(Humidity) ⇒ 0.16 IG(wind) = 0.05


Outlook
S
O
R
{y:3,n:2}
{1,2,8,9,11}
yes
{y:3 n:2}
{4,5,6,10,14}
wind
W
S
{1,2}
{1,1}
Humidity
H
N
{0,3}
{2,0}
Temp
H
m
C
{y:0, n:2}
{1,1}
{1,0}
wind
W
S
{3,0}
{0,2}
Humidity
H
N
{1,1}
{2,1}
Temp
H
m
C
{0,0}
{2,1}
{1,1}


Outlook
S
O
R
Humidity
Yes
wind
H
N
S
W
NO
Yes
NO
Yes