Eg:-

x_0	x_1	x_2	y
1	1	2	3
1	2	1	4
1	3	3	5

$j \rightarrow$

① Compute prediction
② Compute gradient
③ Update Parameters

$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2$

$\theta_0 = \theta_1 = \theta_2 = 0$, $\alpha = 0.1$

$J(\theta) = 25$

⇒ First iteration

① $h_\theta(x^{(1)}) = \theta_0 x_0^1 + \theta_1 x_1^1 + \theta_2 x_2^1 = 0 \times 1 + 0 \times 1 + 0 \times 2 = 0$

$h_\theta(x^{(2)}) = \theta_0 x_0^2 + \text{''} \quad \text{''} = 0 \times 1 + 0 \times 2 + 0 \times 1 = 0$

$h_\theta(x^{(3)}) = \theta_0 x_0^3 + \text{''} \quad \text{''} = 0 \times 1 + 0 \times 3 + 0 \times 3 = 0$

② Compute gradients

i) Compute errors

$e^{(1)} = h_\theta(x^1) - y$	$e^{(2)} = h_\theta(x^2) - y$	$e^{(3)} = h_\theta(x^3) - y$
$= 0 - 3$	$= 0 - 4$	$= 0 - 5$
$= -3$	$= -4$	$= -5$

ii) Calculate Gradient

$$\frac{\partial J}{\partial \theta_0} = \sum_{i=1}^{3} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$= -3 * 1 + -4 * 1 + -5 * 1$

$= -12$

$$\frac{\partial J}{\partial \theta_1} = \sum_{i=1}^{3} (h_\theta(x^i) - y^{(i)})\, x_1^{(i)}$$

$$= -3 \times 1 + -4 \times 2 + -5 \times 3$$

$$= -26$$

$$\frac{\partial J}{\partial \theta_2} = \sum_{i=1}^{3} (h_\theta(x^i) - y^{(i)})\, x_2^{(i)}$$

$$= -3 \times 2 + -4 \times 1 + -5 \times 3$$

$$= -25$$

③ Update Parameters

$$\theta_0 = \theta_0 - \alpha \frac{\partial J}{\partial \theta_0} = 0 - 0.1(-12) = 1.2$$

$$\theta_1 = \theta_1 - \alpha \frac{\partial J}{\partial \theta} = 0 - 0.1(-26) = 2.6$$

$$\theta_2 = \theta_2 - \alpha \frac{\partial J}{\partial \theta} = 0 - 0.1(-25) = 2.5$$

$\theta_0 = 1.2$ $\theta_1 = 2.6$ $\theta_2 = 2.5$

$J = 94.95$ # This may be because we took large steps

Iteration - 2 :-

$\theta_0 = -1.02$, $\theta_1 = -2.41$, $\theta_2 = -2.6$

$J = 25,678.56$

If : $\theta_0 = \theta_1 = \theta_2 = 0$, $\alpha = 0.01$ $J = 10.81$

After iteration 1

$\theta_0 = 0.19$, $\theta_1 = 0.43$, $\theta_2 = 0.41$ $J = 5.11$

After iteration 2

$\theta_0 = 0.25$, $\theta_1 = 0.55$, $\theta_2 = 0.53$ $J = 3$

① Defining the distribution

$$L(\theta) = P(x_{1:n} | \theta)$$

$$= \prod_{i=1}^{n} P(x_i | \theta) \quad \text{\# since IID is assumed}$$

② Computing the log-likelihood :-

$$l(\theta) = \log L(\theta)$$

$$= \log \prod_{i=1}^{n} P(x_i | \theta)$$

$$= \log \left[\theta^m (1-\theta)^{n-m}\right] \quad \log(ab) = \log(a) + \log(b)$$

$$= m \log\theta + (n-m)\log(1-\theta)$$

③ Differentiate and set to '0' to find estimate of θ :-

$$l(\theta) = m\log\theta + (n-m)\log(1-\theta)$$

$$\frac{d(l(\theta))}{d\theta} = \frac{d}{d\theta}\left[m\log\theta + (n-m)\log(1-\theta)\right]$$

$$= \frac{m}{\theta} + (n-m)(-1)\frac{1}{1-\theta}$$

$$= \frac{m(1-\theta) + (m-n)\theta}{\theta(1-\theta)}$$

⇒ Equate to zero

$$0 * \theta(1-\theta) = m(1-\theta) + (m-n)\theta$$

$$0 = m[1-\theta] + (m-n)\theta$$

$$= m - m\theta + m\theta - n\theta$$

$$\theta = \frac{m}{n}$$

$$\boxed{\theta = \frac{m}{n}}$$

⇒ One-Hot Encoding :- [part of EDA]

Colour = {Red, Blue, green}

$$\begin{bmatrix}1\\0\\0\end{bmatrix}_{3\times1} \quad \begin{bmatrix}0\\1\\0\end{bmatrix}_{3\times1} \quad \begin{bmatrix}0\\0\\1\end{bmatrix}_{3\times1}$$

* Vector of size of the set is created and each bit is assigned for each variable respectively
* In EDA we will determine that colour feature must be converted using One-Hot Encoding
* But in target variable we do not use One-hot encoding
  Instead ; Y = 1 N = 0
* ∴ When we assign a vector for a categorical data it is like creating multiple columns within the column

Eg:- $\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \ldots \theta_d x_d$

* Disadv:- The number of features increases

Green	Blue	Red	
1	0	0	} → for green colour

x_1	x_2	y
Red	35.2	Y
blue	38.0	N

⟹

x_1-0	x_1-1	x_2	y
0	1	35.2	Y
1	0	38	N

$h_\theta x^1 = \hat{y} = \theta_0 x_0 + \theta_1 x_1 - 0 + \theta_2 x_2 + \theta_3 x_2$

$= \theta_0 x_0 + \theta_1 x_1 - 0 + \theta_2 x_1 - 1 + \theta_3 x_2$

Eg :- k-fold cross validation

1
2
.
.
.
k

x_1	x_2	y
2	-1	1
0.5	1.2	0
1	2	1
-3	-2	1
4	0.1	0

F1 $\{O_1, O_2\}$ = {1.8, 2.8}

F2 $\{O_1, O_2\}$ = {2.1, 3.1}

F3 $\{O_1, O_2\}$ = {1.9, 4}

- Perform 3 fold CV

split must always remain same ie do not change the folds

⇒ Fold 1 :-

Training Set

x_1	x_2	y
2	-1	1
0.5	1.2	0
1	2	1
-3	-2	1

Test set

x_1	x_2	y	g(z)
4	0.1	0	0

Accuracy

$\theta^T x^5 = -1.8 \times 4 + 2.8 \times 0.1 = -6.92$

$g(z) = \frac{1}{(1 + e^{-z})}$

$= 0.001$

$0.001 < 0.5$

∴ Accuracy = 100%

⇒ Fold 2 :-

Training set

x_1	x_2	y
1	2	1
-3	-2	1
4	0.1	0

Test set

x_1	x_2	y	g(z)
2	-1	1	1
0.5	1.2	0	1

$\theta^T x^1 = 2.1 \times 2 + 3.1 \times -1 = 1.1$

$\theta^T x^2 = 2.1 \times 0.5 + 1.2 \times 3.1 = 4.77$

$g(z)^1 = 0.75$ $g(z)^2 = 0.991$

∴ Accuracy = 50%

⇒ Fold 3 :-

Training Set

x_1	x_2	y
2	-1	1
0.5	1.2	0
4	0.1	0

Test set

x_1	x_2	y	g(z)
1	2	1	1
-3	-2	1	0

$\theta^T x^3 = 1 \times 1.9 + 2 \times 4 = 9.9$

$\theta^T x^4 = -3 \times 1.9 + -2 \times 4 = -13.7$

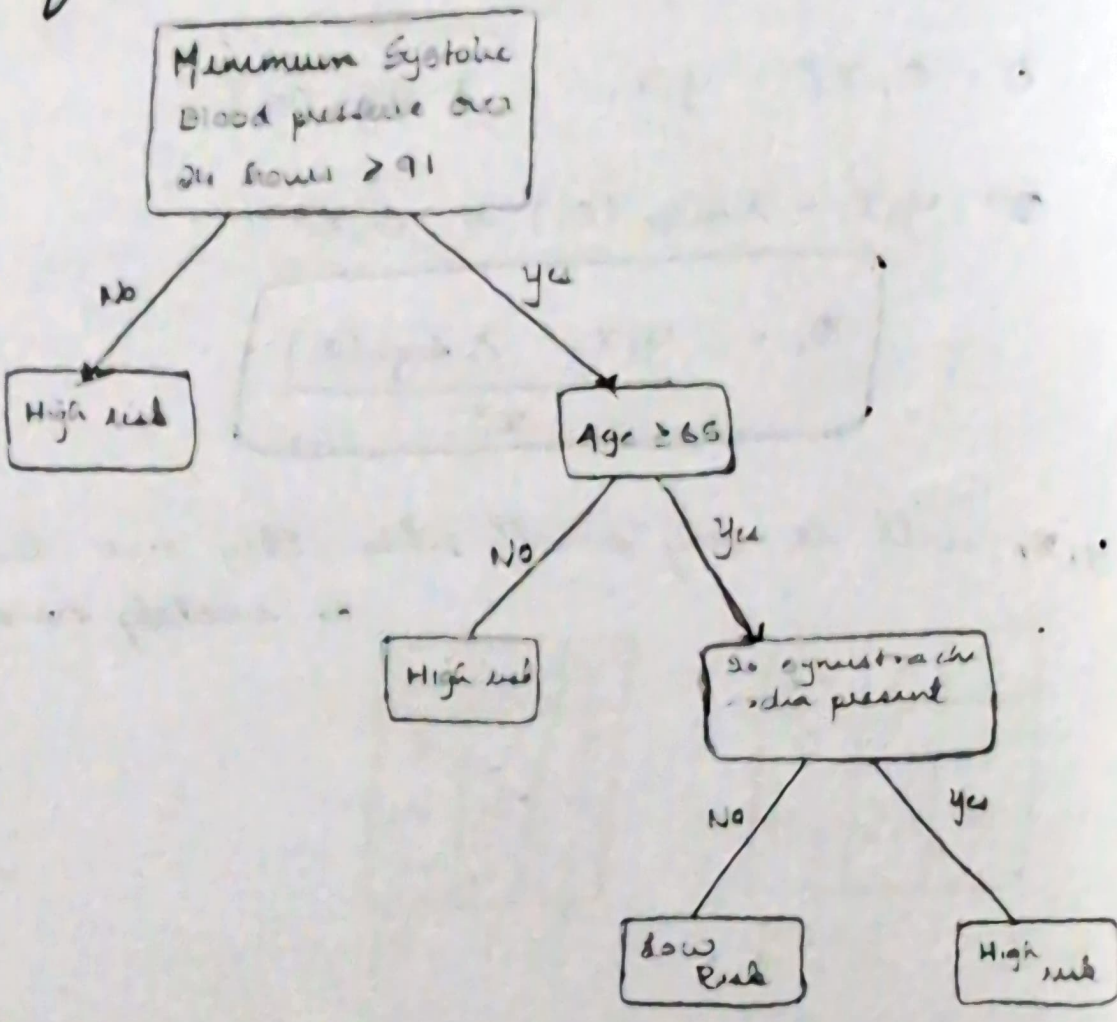
$g(z)^3 = 0.999$

$g(z)^4 = 0.000001$

∴ Accuracy = 50%

⇒ Decision Tree Algorithm :-

1) classification Tree :-



* Here, the tree can be considered as the hypothesis function
* Internal nodes can be initialised at random
* For every internal node the parameters are feature & threshold
  - feature → split dimension
  - threshold → split value

Eg:-

Features :-

x_1 :- date

x_2 :- age

x_3 :- height

x_4 :- weight

x_5 :- Sinus Tachycardia

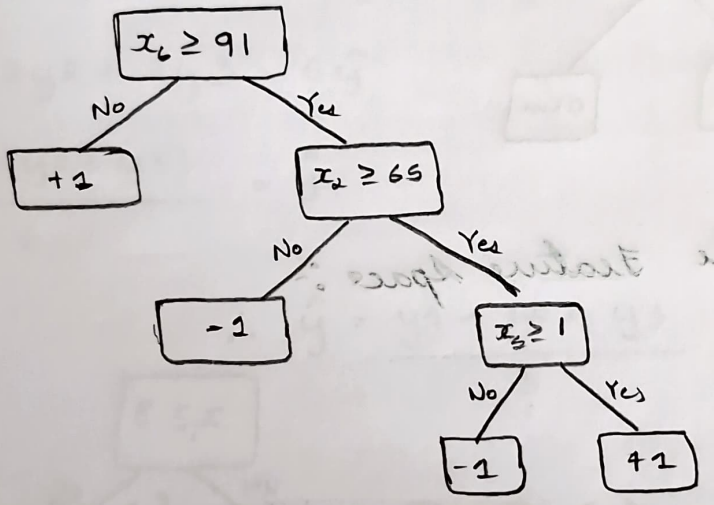
x_6 :- maximum systolic bp 24hrs

x_7 :- latest diastolic bp.

Labels :-

high risk :- +1

low risk :- -1



Eg

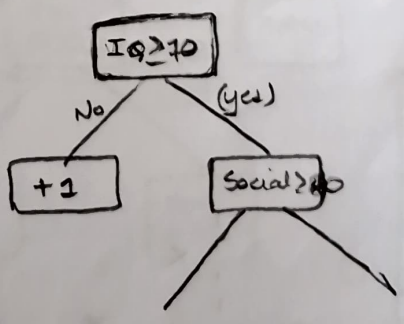
check for this data :

$x^{(i)} = \{2020/11/17, 49,$

Eg:-

Features

IQ	Social	Verbal	Risk for autism
100	79	86	Low
30	40	70	High
80	85	65	Low
90	82	45	High



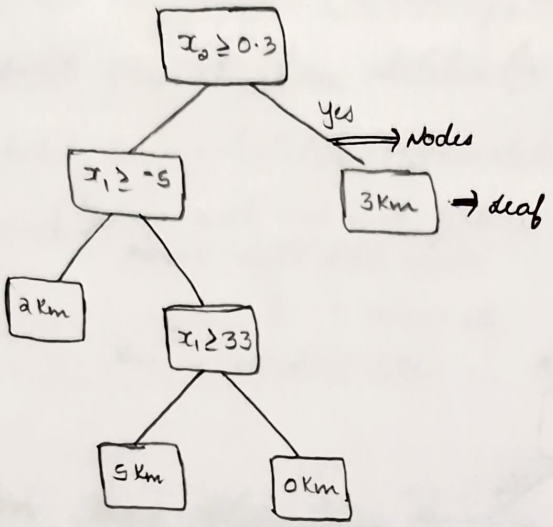
⇒ Regression Tree :-

Eg:-

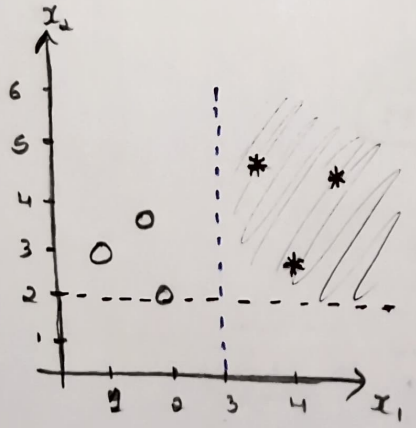
x_1 = Temperature

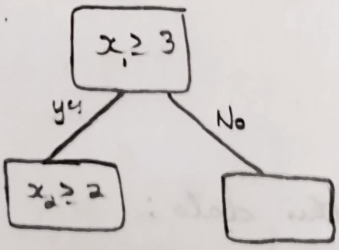
x_2 = precipitation

y : 1 Km , 2 Km , 3 Km

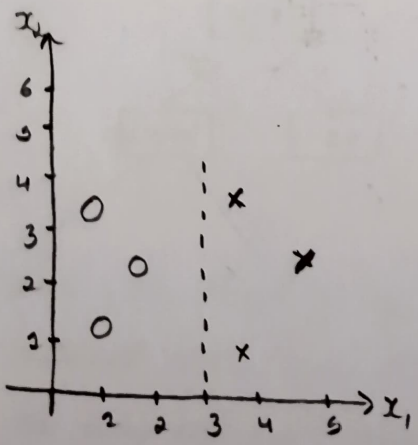


⇒ Partitioning the Feature Space :-

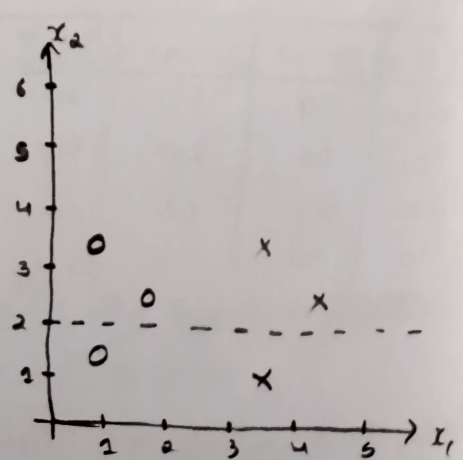




* every node will allow us to split the space



$x_1 \geq 3$



Q) which split is best?

$x_2 \geq 2$

Sol x_1 coz

$y = \begin{bmatrix} -2 \\ 3 \\ 4 \end{bmatrix} \begin{matrix} \to y_1 \\ \to y_2 \\ \to y_3 \end{matrix}$ $\quad \hat{y} = ? \quad (\min) \quad b = (\hat{y}-y_1)^2 + (\hat{y}-y_2)^2 + (\hat{y}-y_3)^2$

To find min, we must compute derivative and set it to 0

$$\frac{\partial b}{\partial \hat{y}} = 2(\hat{y}-y_1)\cdot 1 + 2(\hat{y}-y_2)\cdot 1 + 2(\hat{y}-y_3)\cdot 1$$

$$0 = 2(\hat{y}-y_1) + 2(\hat{y}-y_2) + 2(\hat{y}-y_3)$$

$$= 2\hat{y} - 2y_1 + 2\hat{y} - 2y_2 + 2\hat{y} - 2y_3$$

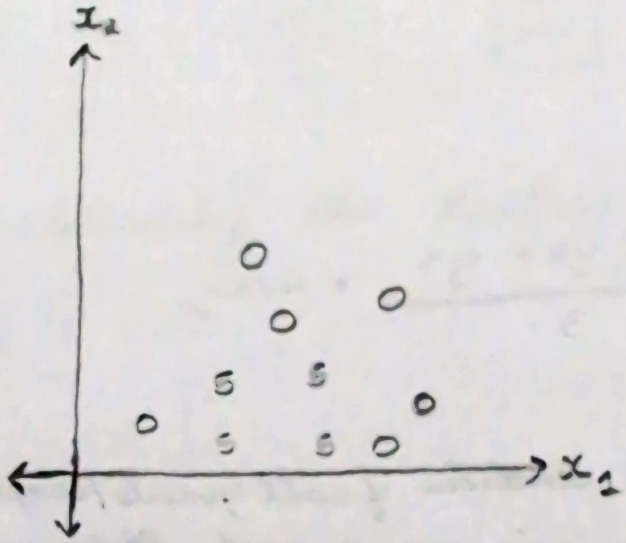
$$= 6\hat{y} - 2y_1 - 2y_2 - 2y_3$$

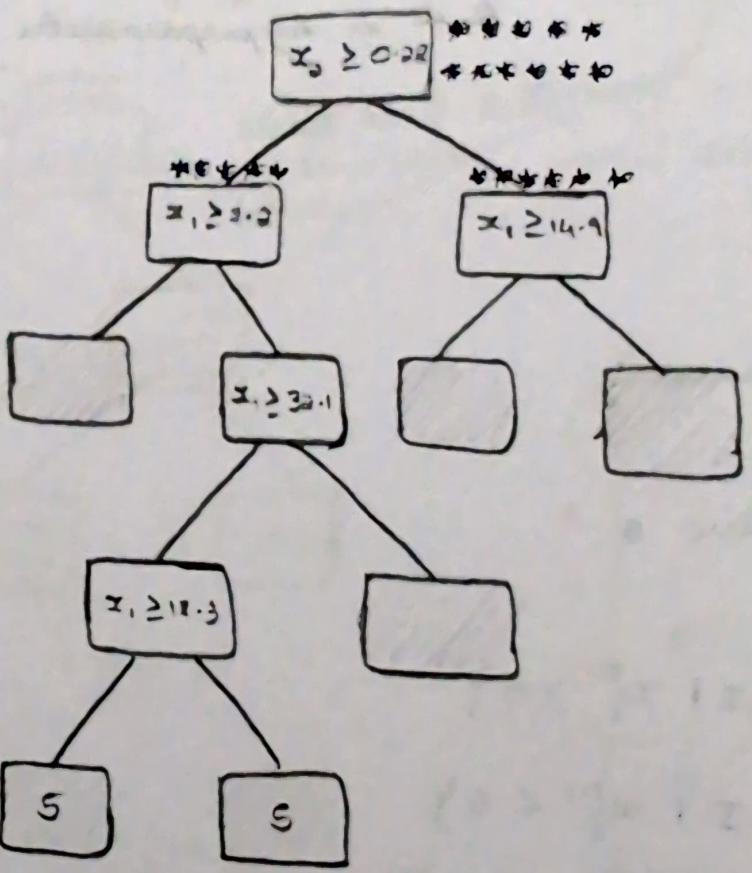
$$2y_1 + 2y_2 + 2y_3 = 6\hat{y}$$

$$\frac{2(y_1+y_2+y_3)}{6} = \hat{y}$$

$$\therefore \hat{y} = \frac{y_1+y_2+y_3}{3} = \text{mean}$$

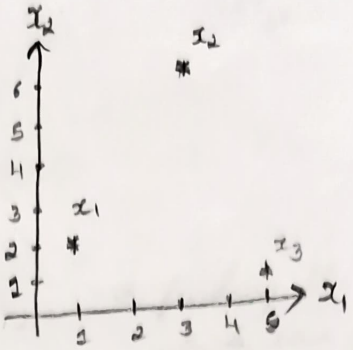
eg :-





⇒ Example question (decision tree)

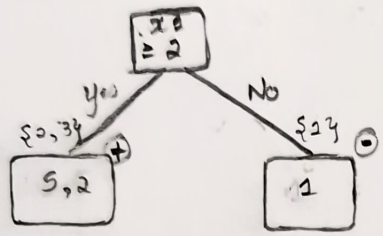
	x_1	x_2	y
→	1	2	1
→	3	6	5
→	5	1	2



for x_1, s = {2, 4}
x_2, s = {

$I = \{1, 2, 3\}$

$k = 2$



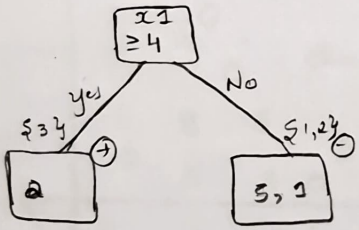
$\hat{y}^+ = \frac{5+2}{2} = \frac{7}{2} = 3.5$

$\hat{y}^- = \frac{1}{1} = 1$

$E^+ = (5-3.5)^2 + (2-3.5)^2 = 4.5$

$E^- = (1-1)^2 = 0$

$E = 4.5$



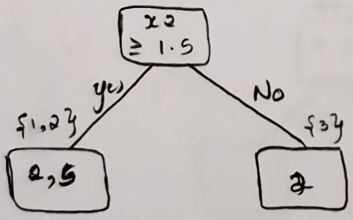
$\hat{y}^+ = \frac{2}{1} = 2$

$\hat{y}^- = \frac{5+1}{2} = \frac{6}{2} = 3$

$E^+ = (2-2)^2 = 0$

$E^- = (3-5)^2 + (1-3)^2 = 4+4 = 8$

$E = 8 - 0 = 8$



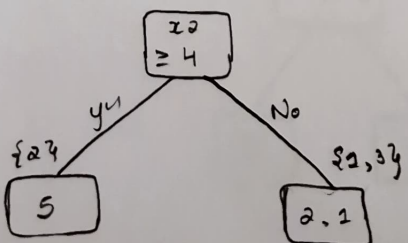
$\hat{y}^+ = \frac{1+5}{2} = \frac{6}{2} = 3$

$\hat{y}^- = 2/2 = 2$

$E^+ = (1-3)^2 + (5-3)^2 = 4+4 = 8$

$E^- = 2 - 2 = 0$

$E = 8 - 0 = 8$



$\hat{y}^+ = \frac{5}{1} = 5$ $\quad E^+ = (5-5)^2 = 0$

$\hat{y}^- = \frac{2+1}{2} = \frac{3}{2}$ $\quad E^- = \left(2-\frac{3}{2}\right)^2 + \left(1-\frac{3}{2}\right)^2$

$= \frac{1}{4} + \frac{1}{4}$

$= 2/4 = 1/2$

$E = 1/2$

$j^*, s^* = \min\{E_{(x_1,2)}, E_{(x_1,4)}, E_{(x_2,1.5)}, E_{(x_2,4)}\} \Rightarrow \arg\min(\text{expand}, s)$

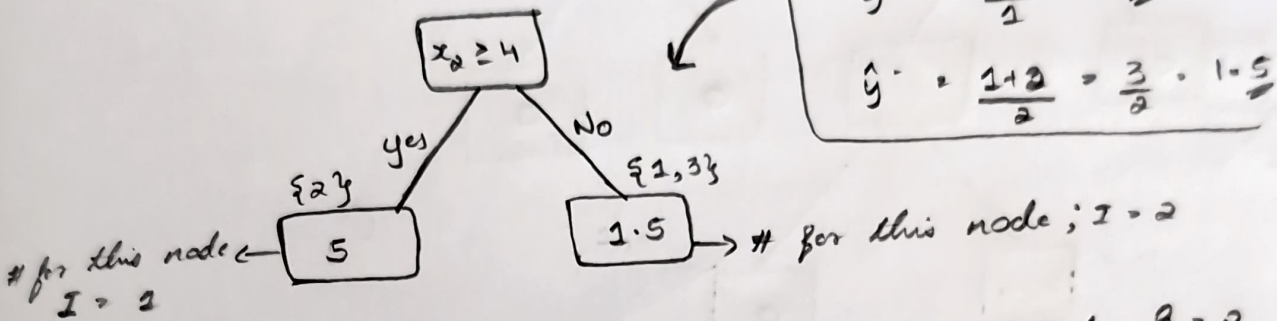
$j^*, s^* = x_{(2,4)}$

$j^* s^* = [x_2, 4]$

If block condition

$\hat{y}^+ = \frac{5}{1} = 5$

$\hat{y}^- = \frac{1+2}{2} = \frac{3}{2} = 1.5$

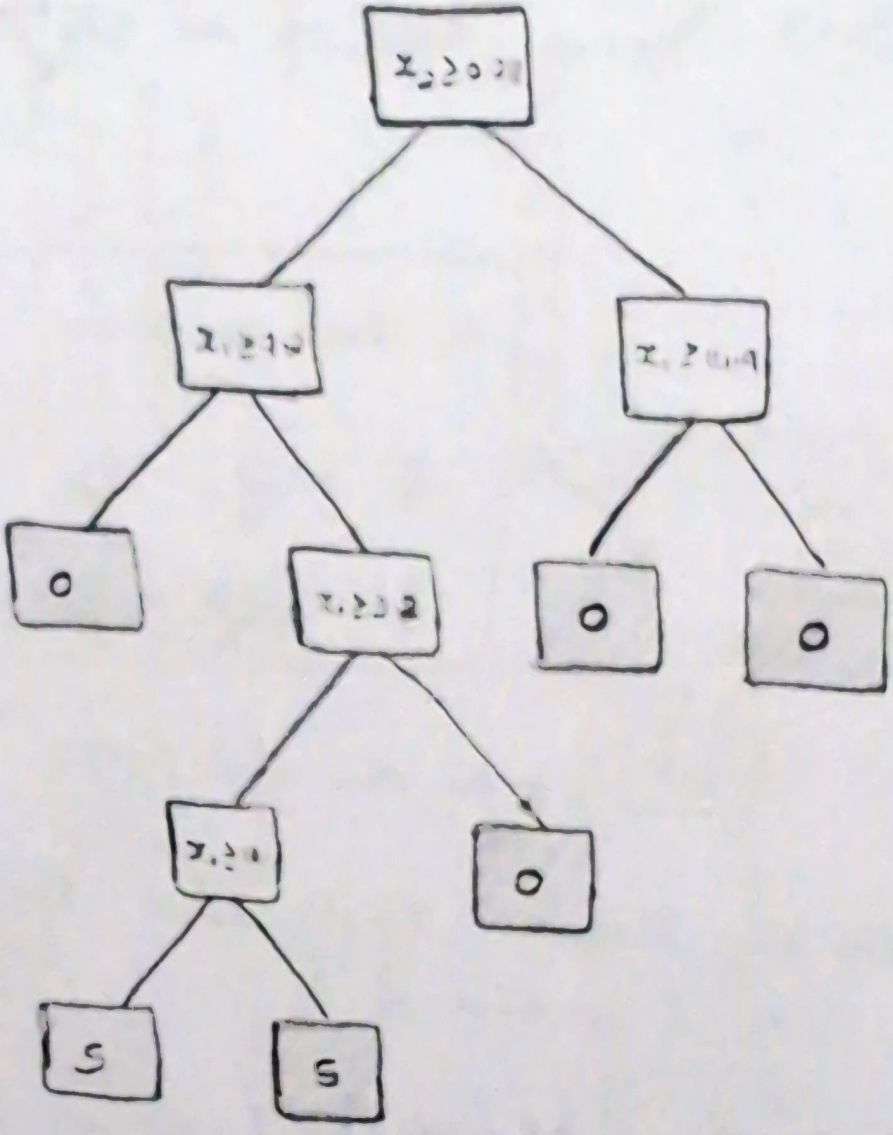


stop here because I = 2 and k = 2
ie I is now $\leq k$

* When we invoke build_tree for right node ['No'], we see that I = 2 which satisfies the if condition ∴ doesn't enter else loop

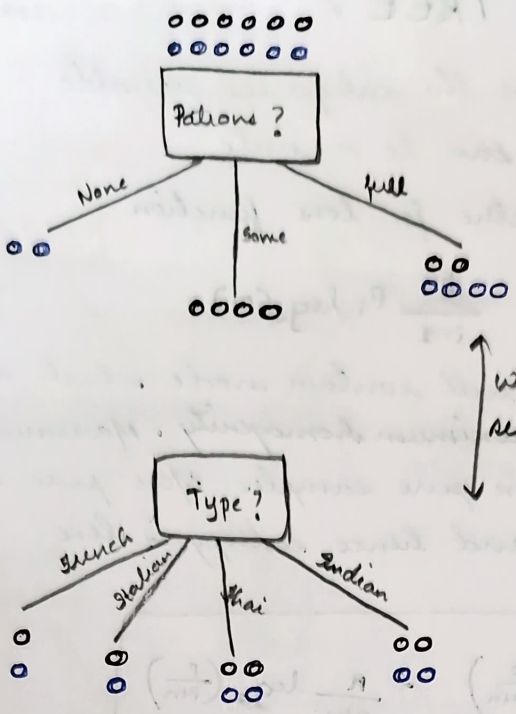
* When we invoke build_tree for left node ['Yes'], we see that I = 1, which satisfies if condition, ∴ doesn't enter the loop.

* The leaf contains predicted $\hat{y}$ values for the corresponding data-set ie $\hat{y}$ predicted for 1st and 3rd dataset is 1.5 and 2 dataset is 5



* When the tree is potentially large and/or has very high number of leaves there is very high chance that there can be overfitting of the model

* Hence we do regularization to avoid this where we add a penalty term



● :- wait
● :- wont wait

$E(A) = 1$

which one would you select ?

∴ :- Patrons because higher information gain

$$EH(A) = \sum_{i=1}^{k} \frac{p_i + n_i}{p+n} H\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

Entropy

p :- ●
n :- ○

$$I(A) = H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - EH(A)$$

Information gain

$H(\text{patrons}) - EH(\text{patrons})$

* Pick value of entropy which gives highest information gain

(parent) $H(\text{patrons}) = H\left(\frac{p}{p+n}, \frac{n}{p+n}\right)$

$= H\left(\frac{6}{12}, \frac{6}{12}\right)$

$= 1$

(all 3 children) $EH(\text{patrons}) = \sum_{i=1}^{k} \frac{p_i+n_i}{(p+n)} H\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$

$= \left[\frac{p_1+n_1}{12} H\left(\frac{0}{2}, \frac{2}{2}\right) + \frac{p_2+n_2}{12}\left(H\left(\frac{4}{4}, \frac{0}{4}\right)\right.\right.$

$\left.\left. + \frac{p_3+n_3}{12} H\left(\frac{2}{6}, \frac{4}{6}\right)\right)\right]$

$$H\left(\frac{0}{2},\frac{2}{2}\right) = \frac{-p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

$$= -\frac{0}{2}\log_2\left(\frac{0}{2}\right) - \frac{2}{2}\log_2\left(\frac{2}{2}\right)$$

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

$$EH(\text{patrons}) = \frac{2}{12}(-0\log_2(0) - 1\log_2(1)$$

$$+$$

$$\frac{4}{12}(-1\log_2(1) - 0\log_2(0)$$

$$+$$

$$\frac{6}{12}\left(-\frac{2}{6}\log_2\left(\frac{2}{6}\right) - \frac{4}{6}\left(\log_2(4/6)\right)\right)$$

$$= \frac{1}{6}(0-0) + \frac{1}{3}(0-0) + \frac{6}{12}\left(-\frac{2}{6}(-1.585 - \frac{4}{6}\right.$$

$(-0.585))$

$E[H]$

$+0+\frac{6}{12}(0.5)-(0.92)$ $= 0.455$

$0.5+0$

$\therefore I(A) = 1 - 0.455$

$I(A) = 0.55$

$I(\text{patrons}) = 0.55$

using similar steps

Gini index

$I(\text{types}) = 0.00$

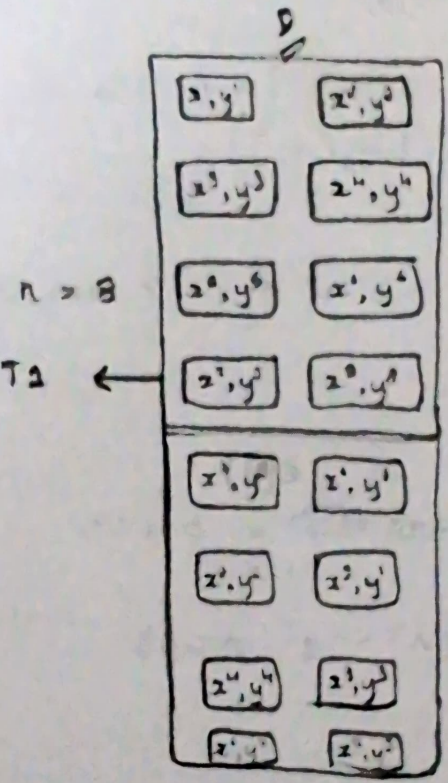
$\therefore$ we select patrons as it have more information gain.

⇒ Ensemble Methods :-

Ensemble Algorithms

- Bagging Algorithm (Bootstrapping aggregation)
- Boosting Algorithm

Ensemble Algorithms :-



* Randomly pick one dataset
* Then again randomly pick another data set (with replacement)
* Continue this for n times
* Bootstrapping :-

D1 (Bag 1)

D2 (Bag 2)

D1, D2 . . . DB
T1, T2 . . . TB

* Like this B datasets can be constructed.
* Each data set will be a tree
* ∴ we will get B number of trees
* decision tree algorithm will be applied for each bag.

→ Inference :-

$x \longrightarrow \hat{y}$

$x \longrightarrow T_1 \longrightarrow \hat{y}_1$

$x \longrightarrow T_2 \longrightarrow \hat{y}_2$

$x \longrightarrow T_3 \longrightarrow \hat{y}_3$

$$\hat{y} = \frac{1}{B} \sum_{i=1}^{B} \hat{y}_i$$

* Aggregation :-

→ Why is ensemble better than decision tree ?
→ What learns ?

Bias and Variance to cross validation

Random forest Eg:-

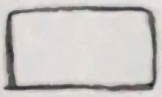
E.2

	Age	Salary	Spend Pattern	Buy Product
1	16	0	High	Yes
2	30	90,000	Low	No
3	31	90,000	High	No
4	55	3,00,000	Medium	No

b1	b0
2, 3, 3, 4 (Random pattern)	1, 1, 3, 1 (Random pattern

Age	Sal	Spend	Buy
30	90	High	No
31	90	Low	No
31	90	High	No
55	300	Med	No

Tree 1

 → Selecting first node

for feature in (Age, Salary)

{

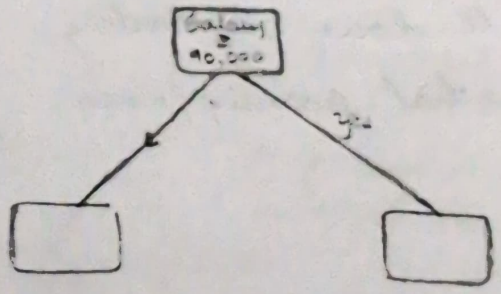
for each value in feature

{

IG = (IG of value) append

IG of Age = 1.2

IG of Salary = 1.2 ✓



for feature in (Spend pattern, Salary)

{

for each value in feature

{

IG = (IG of value) appnd

IG of SP = 0.2

IG of Salary = 1 ✓ Appropte to the tree

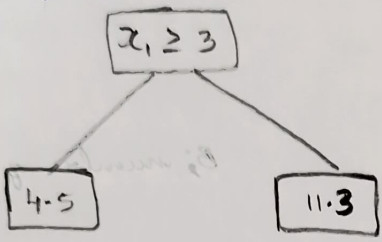
Eg : 1

x_1	y	r
1	3	3
2	6	6
3	8	8
4	11	11
5	15	15

Step 1

$r = y$

a) fit a model



b, c)

$\lambda = 0.1$

x_1	r	$\hat{y}$	
1	3	3.45	→ 3 + 0.1(4.5)
2	6	6.45	
3	8	9.13	→ 8 + 0.1(11.3)
4	11	10.13	
5	15	16.13	

c)

x_1	r	$\hat{y}$	r.new
1	3	3.45	-0.45
2	6	6.45	-0.45
3	8	9.13	-1.13
4	11	10.13	-1.13
5	16	16.13	-1.13

Inference :

$x_1 = 5$ → testing input

$$= \sum_{i=1}^{B} \lambda f_b(x)$$

$$= \lambda f_1(5) + \lambda f_2(5) + \lambda f_3(5) \dots \lambda f_{(4)} 5$$

$$= 0.8$$

1) Boosted Regression Tree (AdaBoost Regression)

2) Boosted classification Tree (AdaBoost classification)

$$F = f_0(x) + \lambda f_1(x) + \lambda f_2(x) + \cdots \lambda f_0(x)$$

Eg:- 2 :-

x1	x2	y
1	2	4
2	3	5
3	4	6
4	5	7

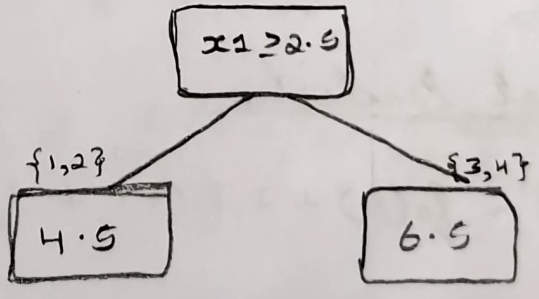
⇒ Iteration - 1

* $r = y$ Step 1

x1	x2	r
1	2	4
2	3	5
3	4	6
4	5	7

Step 2 :-

a) Fit a model $f_1(x) \rightarrow r$



b) update the model ($\lambda = 0.1$) → c) Update the residuals

$$F(x) = f_0(x) + \lambda f_1(x)$$

$$r^1 = r^1 - \lambda f_1(x^0)$$
$$r^2 = r^{(0)} - \lambda f_1(x^2)$$
$$r^3 = r^{(3)} - \lambda f_1(x^3)$$
$$r^4 = r^{(4)} - \lambda f_1(x^4)$$

c)

$r^{(1)} = 4 - [0.1 * 4.5] = 3.55$

$r^{(2)} = 5 - [0.1 * 4.5] = 4.55$

$r^{(3)} = 6 - [0.1 * 6.5] = 5.35$

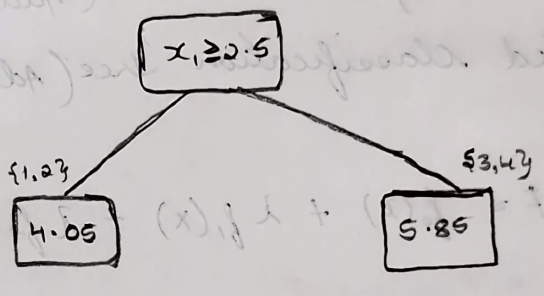
$r^{(4)} = 7 - [0.1 * 6.5] = 6.35$

$\Rightarrow$ Iteration 2

x1	x2	r
1	2	3.55
2	3	4.55
3	4	5.35
4	5	6.35

Step 2

a) Fit a model



b)

c) Update residuals

$r^1 = r^1 - \lambda f_2(x^1) = 3.55 - [0.1 * 4.05] = 3.145$

$r^2 = r^2 - \lambda f_2(x^2) = 4.55 - [0.1 * 4.05] = 4.145$

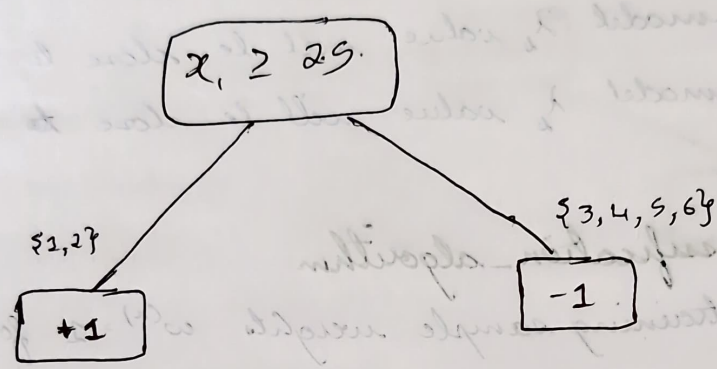
$r^3 = r^3 - \lambda f_3(x^3) = 5.35 - [0.1 * 5.85] = 4.765$

$r^4 = r^4 - \lambda f_4(x^4) = 6.35 - [0.1 * 5.85] = 5.765$

③ $\Rightarrow$ Final Prediction

$F(x;\theta) = f_0(x) + \lambda_1 f_1(x) + \lambda_2 f_2(x) + \dots \lambda f_0(x)$

x_1	x_2	y	w
1	2	1	1
2	3	1	1
3	3	1	1
4	5	-1	1
5	5	-1	1
6	6	-1	1



$$E = \frac{1}{6} \Rightarrow \frac{\text{mismatch}}{\text{total}} = 0.1667$$

$$\lambda = \frac{1}{2} \log_e \left[\frac{1 - 0.1667}{0.1667} \right]$$

$$= 0.808$$

$$w^{(1)} = 1 \cdot e^{-0.805} \quad \text{(correct)} = 0.447$$

$$w^{(-1)} = 1 \cdot e^{0.805} = 2.236$$

x_1	x_2	y	w
1	2	+1	0.447
2	3	1	0.447
3	3	1	2.236
4	5	-1	0.447
5	5	-1	0.447
6	6	-1	0.447

⇒ Kernel :

$K(x,y) = \langle \phi(x), \phi(y) \rangle$, $\langle\rangle$ = dot product

$x = (x_1, x_2, x_3)$ * x & y are just 2 random vectors, not features & ground truths

$y = (y_1, y_2, y_3)$

Eg

$\phi x = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}$

$x = [1, 2, 3]$

$y = [4, 5, 6]$

$\langle \phi(x), \phi(y) \rangle$

$\phi y = \begin{bmatrix} y_1y_1 \\ y_1y_2 \\ \\ \\ y_3y_3 \end{bmatrix}$

$\phi(x) \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 4 \\ 6 \\ 3 \\ 6 \\ 9 \end{bmatrix}$ $\phi(y) \begin{bmatrix} 16 \\ 20 \\ 24 \\ 20 \\ 25 \\ 30 \\ 24 \\ 30 \\ 36 \end{bmatrix}$

$\langle \phi x \quad \phi y \rangle = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324$

$= 1024$

* can we do something on orginal space which is equivalent to higher space?

* This is where kernel helps

$K(x,y) = (\langle x, y \rangle)^2$

$= (32)^2$

$= 1024$

* We are able to mimic what happend in higher dimension in lower dimensional space using kernel function

Eg:

x_1	x_2	y
1	2	0
3	4	0

$$\phi(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}$$

$X = [x_1, x_2]^T$ # 2 dimensional space

$X = \mathbb{R}^2$ $\quad \phi(x) \in \mathbb{R}^3$ $\quad \theta \in \mathbb{R}^3 \Rightarrow [\theta_1, \theta_2, \theta_3]^T$

$$\theta = \sum_{i=1}^{n} \beta_i \phi(x^i)$$

$$\theta = \sum_{i=1}^{2} \beta_i \phi(x^i)$$

=

① Compute each sample to higher dimensional space

$$\phi(x^1) = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}_{3\times 1}$$

$$\phi(x^2) = \begin{bmatrix} 9 \\ 16 \\ 12 \end{bmatrix}_{3\times 1}$$

② Compute the θ

$$\theta = \beta_1 \phi(x^1) + \beta_2 \phi(x^2)$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \beta_1 + 9\beta_2 \\ 4\beta_1 + 16\beta_2 \\ 2\beta_1 + 12\beta_2 \end{bmatrix}$$

⇒ SVM

1) Maximal Margin classifier (Constrained optimization problem)

Maximize M

$\beta_0, \beta_1, \beta_2 \ldots \beta_p, M$ —— ①

Subject to

$\sum_{j=1}^{p} \beta_j^2 = 1$ —— ② * similar to l_2 regularization

$y^i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots \beta_p x_p) \geq M$ — ③

for all $i = 1$ to n

$$d = \frac{|\beta_0 + \beta_1 x_1 + \beta_2 x_2|}{\sqrt{\beta_1^2 + \beta_2^2}}$$

Eg: Three lines (hyperplane)

$2x_1 + 3x_2 - 5 = 0$

$-x_1 + 4x_2 + 7 = 0$

$9x_1 - 12x_2 + 10 = 0$

Dataset

x_1	x_2	y
3	4	1
2	3	1
1	-1	-1
-2	1	-1

M2 = $\frac{-5 + 2(3) + 3(4)}{\sqrt{2^2 + 3^2}}$

M1 = $\frac{|-5 + 2(1) + 3(4)|}{\sqrt{2^2 + 3^2}}$

= $\frac{13}{\sqrt{13}}$

M2 = $\frac{-5 + 2(0) + 3(3)}{\sqrt{4 + 9}}$

= $\frac{-5 + 4 + 9}{\sqrt{13}}$

= $\frac{8}{\sqrt{13}}$

M3 = $\left|\frac{-5 + 2(1) + 3(-1)}{\sqrt{13}}\right|$

= $\left|\frac{-5 + 2 - 3}{\sqrt{13}}\right|$

= $\frac{+6}{\sqrt{13}}$

M4 = $\frac{-5 + 2(-2) + 3(0)}{\sqrt{13}}$

= $-5 - 4 + 3$

= $\frac{-6}{\sqrt{13}}$

min = $6/\sqrt{13}$ = 1.664

⇒ Hyperplane - 2

M1 = $\frac{+7 - 1(3) + 4(4)}{\sqrt{1^2 + 4^2}}$

= $\frac{7 - 3 + 16}{\sqrt{17}}$

= $\frac{20}{\sqrt{17}}$

M2 = $\frac{+7 - 1(2) + 4(3)}{\sqrt{17}}$

= $\frac{7 - 2 + 12}{\sqrt{17}}$

= $\frac{17}{\sqrt{17}}$

M3 = $\frac{+7 - 1(1) + 4(-1)}{\sqrt{17}}$

= $\frac{+7 - 1 - 4}{\sqrt{17}}$ = $\frac{2}{\sqrt{17}}$

M4 = $\frac{+7 - 1(-2) + 4(1)}{\sqrt{17}}$

= $\frac{7 + 2 + 4}{\sqrt{17}}$ = $\frac{13}{\sqrt{17}}$