Regression Analysis:- (gives approx. values)

Regression lines:- ① y on x line → This line used to find y when x is given
② x on y line → " " " " " " x " y "

① Regression line y on x defined as

$$y - \bar{y} = b_{yx}(x - \bar{x})$$

② The regression line x on y defined as

$$x - \bar{x} = b_{xy}(y - \bar{y})$$

where,

b_{xy} = Regression coefficient x on y
b_{yx} = " " y on x

$$\bar{x} = \frac{\Sigma x}{n}, \quad \bar{y} = \frac{\Sigma y}{n}$$

x	y
x_1	y_1
x_2	y_2
⋮	⋮
x_n	y_n

Regression coefficient

$$b_{yx} = \rho \frac{\sigma_y}{\sigma_x} = \frac{\Sigma xy - \frac{\Sigma x \Sigma y}{n}}{\Sigma x^2 - \frac{(\Sigma x)^2}{n}}$$

where $\sigma_y^2 = \frac{\Sigma (y - \bar{y})^2}{n}$, $\sigma_x^2 = \frac{\Sigma (x - \bar{x})^2}{n}$

ρ = coefficient of correlation

$$b_{xy} = \rho \frac{\sigma_x}{\sigma_y} = \frac{\Sigma xy - \frac{\Sigma x \Sigma y}{n}}{\Sigma y^2 - \frac{(\Sigma y)^2}{n}}$$

If $u = x - A$ where A = assumed mean of x
$v = y - B$ B = assumed mean of y

Then, $b_{xy} = \frac{\Sigma uv - \frac{\Sigma u \Sigma v}{n}}{\Sigma v^2 - \frac{(\Sigma v)^2}{n}}$, $b_{yx} = \frac{\Sigma uv - \frac{\Sigma u \Sigma v}{n}}{\Sigma u^2 - \frac{(\Sigma u)^2}{n}}$

Properties of regression lines

1) The regression lines always intersect at the point $(\bar{x}, \bar{y})$

2) If two regression lines are given and θ be the acute angle between them, then show that

$$\tan\theta = \left|\frac{1-\rho^2}{\rho}\right| \left|\frac{\sigma_x \cdot \sigma_y}{\sigma_x^2 + \sigma_y^2}\right|$$

where ρ = coefficient of correlation

Q) Find the two lines of regression from the following data

Age of husband: 25 22 28 26 35 20 22 40 20 18

Age of wife: 18 15 20 17 22 14 16 21 15 14

Estimate (i) the age of husband when age of wife is 19 and (ii) age of wife, when age of husband is 30 (iii) correlation coefficient

Let x = age of husband, y = age of wife and $u = x - A = x - 26$,

$v = y - B = y - 17$ where 26 and 17 are assumed mean for x and y, and $n = 10$

Here, $\bar{x} = \Sigma x/n = 256/10 = 25.6$, $\bar{y} = \Sigma y/n = 172/10 = 17.2$

Now regression coefficient, $b_{yx} = 0.385$

$b_{xy} = 2.23$

Here the line of regression

y on $x \Rightarrow y - \bar{y} = b_{yx}(x - \bar{x})$

$y - 17.2 = 2.23(x - 25.6)$

or, $\boxed{y = 0.385x + 7.34}$ — (1)

Regression line x on y

$x - \bar{x} = b_{xy}(y - \bar{y})$

$\Rightarrow \boxed{x = 2.23y - 12.76}$ — (2)

(i) When $y = 19$ (age of husband)

Then, $x = 29.6 \approx 30$

(ii) When $x = 30$ (age of husband)

Then $y = 18.89 \approx 19$

(iii) Now the coefficient of correlation

$\rho = \pm\sqrt{b_{yx} \cdot b_{xy}}$

$= 0.927$ $\because$ b_{yx} & b_{xy} are +ve

Chapter - 6

1. Linear Regression & Correlation:

x	-1	1	2	4	6	7
y	-1	2	3	3	5	8

Sol^n:-

x	y	x^2	y^2	xy
-1	-1	1	1	1
1	2	1	4	2
2	3	4	9	6
4	3	16	9	12
6	5	36	25	30
7	8	49	64	56

$\Sigma x = 19$ $\Sigma y = 20$ $\Sigma x^2 = 107$ $\Sigma y^2 = 112$ $\Sigma xy = 107$

We have,

y-intercept (a) of regression;

$$a = \frac{\Sigma x \Sigma xy - \Sigma y \Sigma x^2}{(\Sigma x)^2 - n\Sigma x^2} = \frac{19 \times 107 - 20 \times 107}{(19)^2 - 6 \times 107} = 0.3808$$

slope (b) of regression;

$$b = \frac{(\Sigma x)(\Sigma y) - n\Sigma xy}{(\Sigma x)^2 - n\Sigma x^2} = \frac{19 \times 20 - 6 \times 107}{(19)^2 - 6 \times 107} = 0.9324$$

Now,

$y = a + bx$ [$\because$ For $x = -1$ & $y = -1$]

or, $-1 = 0.38 + 0.93(-1)$

$\therefore y \simeq 0.9324x + 0.3808$

x within range - intrapolation
otherwise extrapolation (r & a changes)

$$S_x = \sqrt{\frac{\Sigma(x-\bar{x})^2}{n-1}} = \sqrt{\frac{\Sigma x^2 - \frac{1}{n}(\Sigma x)^2}{n-1}}$$

Correlation of: (r)

$$r = \left(\frac{S_x}{S_y}\right) b$$

OR;

$$r = \frac{\Sigma xy - 1/n\, \Sigma x \Sigma y}{(n-1)\, S_x S_y}$$

Now;

$$S_x = \sqrt{\frac{\Sigma x^2 - 1/n(\Sigma x)^2}{n-1}} = \sqrt{\frac{107 - 1/6\,(19)^2}{6-1}} = 3.0605$$

$$S_y = \sqrt{\frac{\Sigma y^2 - 1/n(\Sigma y)^2}{n-1}} = \sqrt{\frac{112 - 1/6\,(20)^2}{6-1}} = 3.0110$$

$$\therefore r = \left(\frac{3.0605}{3.0110}\right) 0.9324 = 0.9477$$

If $y = 4.5$, then

$$\frac{4.5 - 0.3808}{0.9324} = x$$

$$\therefore x = 4.4178 \text{ (intrapolation)}$$

2. Multiple Linear Regression:

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$$

where,

α_i = Regression coefficient
x_i = independent variables
y = dependent variables
n = attributes, features

3. Regularization:

Ridge Regression
Lasso Regression

i) Ridge Regression:

$$y = 0.9 + 1.2x_1 + 20x_2 + 39x_3$$

(scale down)

$$\therefore y = 0.9 + 0.7x_1 + 2x_2 + 5x_3$$

ii) Lasso Regression:

$$y = 0.9 + 0x_1 + 0x_2 + 5x_3$$

3. Logistic Regression:

Decision Tree Example:

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	weak	NO
2	Sunny	Hot	High	strong	NO
3	Overcast	Hot	High	weak	Yes
4	Rain	Mild	High	weak	Yes
5	Rain	Cool	Normal	weak	Yes
6	Rain	Cool	Normal	strong	NO
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	weak	NO
9	Sunny	Cool	Normal	weak	Yes
10	Rain	Mild	Normal	weak	Yes
11	Sunny	Mild	Normal	strong	Yes
12	Overcast	Mild	High	strong	Yes
13	Overcast	Hot	Normal	weak	Yes
14	Rain	Mild	High	strong	NO

Decision Tree:

1. ID3 (Iterative

$$Entropy(S) = \Sigma - p(I) \cdot \log_2 P(I)$$

$$Gain(S, A) = Entropy(S) - \Sigma [P(S|A) \cdot Entropy(S|A)]$$

where;

S = decision

A = attributes

Here;

9 decisions labelled Yes

5 decisions labelled NO

Total Decision = 14

$$\begin{aligned} Entropy(Decision) &= -p(Yes) * \log_2 P(Yes) - p(NO) * \log_2 P(NO) \\ &= -(9/14) * \log_2 (9/14) - (5/14) * \log_2 (5/14) \\ &= 0.6428 * 0.6374 + 0.3571 * 1.4854 \\ &= 0.9420 \end{aligned}$$

Wind Factor on Decision:

$$Gain(D, W) = Entropy(D) - \Sigma [P(D|W) * Entropy(D|W)]$$

$$\begin{aligned} Gain(D, W) = Entropy(D) &- [P(D|W = Weak) * Entropy(D|W = Weak)] \\ &- [P(D|W = strong) * Entropy(D|W = strong)] \end{aligned}$$

Weak Wind Factor Decision:

$$\text{Entropy}(D|W = \text{Weak}) = -p(NO) * \log_2 p(NO) - p(Yes) * \log_2 p(Yes)$$
$$= -(2/8) * \log_2(2/8) - (6/8) * \log_2(6/8)$$
$$= 0.8112$$

Strong Wind Factor Decision:

$$\text{Entropy}(D|W = \text{strong}) = -p(NO) * \log_2 P(NO) - p(Yes) * \log_2 P(Yes)$$
$$= -(3/6) * \log_2(3/6) - (3/6) * \log_2(3/6)$$
$$= 1$$

Now;

$$\text{Gain}(D, W) = 0.94 - (8/14) * 0.8112 - (6/14) * 1$$
$$= 0.048$$

Again;

Outlook Factor on Decision:

S = 5, O = 4, R = 5

Now;

$$\text{Entropy}(D|O = \text{sunny}) = -(3/5) * \log_2(3/5) - (2/5) * \log_2(2/5)$$
$$= 0.9709$$

$$\text{Entropy}(D|O = \text{overcast}) = -(0/4) * \log_2(0/4) - (4/4) * \log_2(4/4)$$
$$= 0$$

$$\text{Entropy}(D|O = \text{Rainy}) = -(2/5) * \log_2(2/5) - (3/5) * \log_2(3/5)$$
$$= 0.9709$$

$$\text{Gain}(D, O) = 0.9402 - (5/14) * 0.9709 - (4/14) * 0 - (5/14) * 0.9709$$
$$= 0.2467$$

Humidity Factor on Decision: H = 7 , N = 7

Entropy (D|H = High) = $-(4/7) * \log_2(4/7) - (3/7) * \log_2(3/7)$
= 0.9852

Entropy (D|H = Normal) = $-(1/7) * \log_2(1/7) - (6/7) * \log_2(6/7)$
= 0.5916

Gain (D, H) = 0.9402 - (7/14) * 0.9852 - (7/14) * 0.5916
= 0.1518

Temperature Factor on Decision: H =

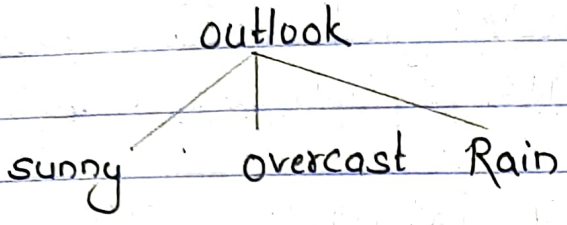
Entropy (D|T = High) =

Entropy (D|T = Mild) =

Entropy (D|T = cool) =

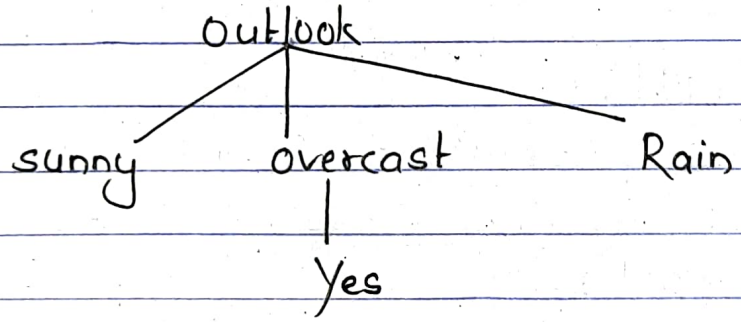
Gain (D, T) =
= 0.029

G(D, W) = 0.048
G(D, H) = 0.151
G(D, O) = 0.246 Highest
G(D, T) = 0.029



- outlook factor on decision produces the highest one, so outlook decision will appear in the root node.

Overcast Outlook on decision:
- Decision will always be Yes.



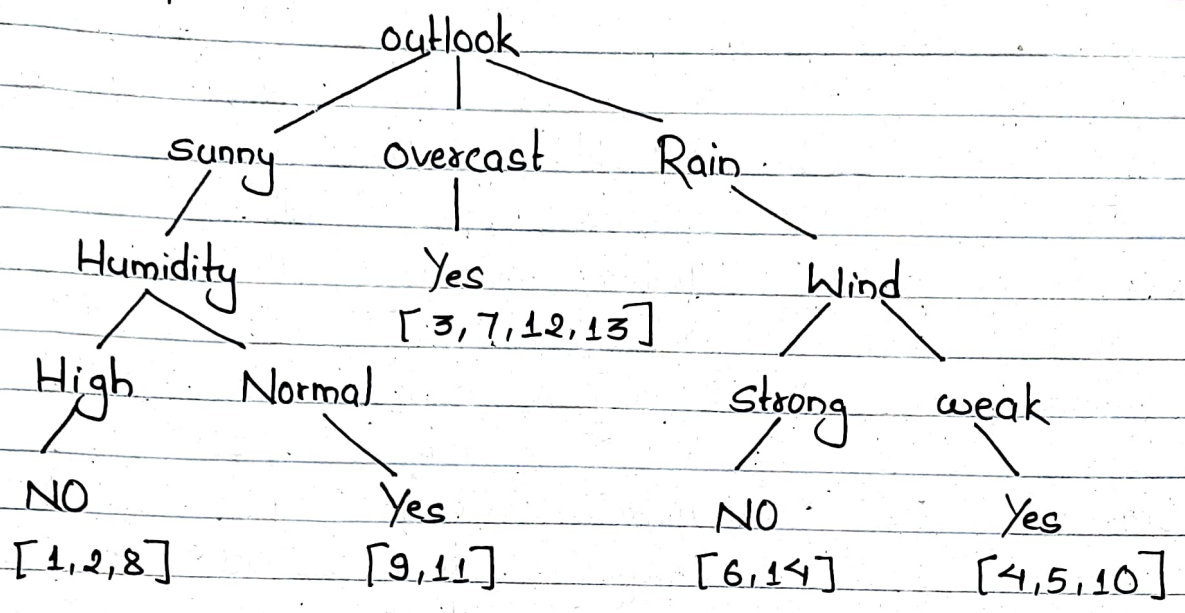
Sunny outlook on decision:

- (outlook = sunny | Temperature) gain = 0.570
  (outlook = sunny | Humidity) gain = 0.970
  (outlook = sunny | Wind) gain = 0.019

- Humidity is the decision.

CART, C4.5 & C5.0 (Gini index)

Complete Decision Tree:



Gain (outlook = Rain | Temperature)
Gain (outlook = Rain | Humidity)
Gain (outlook = Rain | Wind)

- Wind has the highest gain.

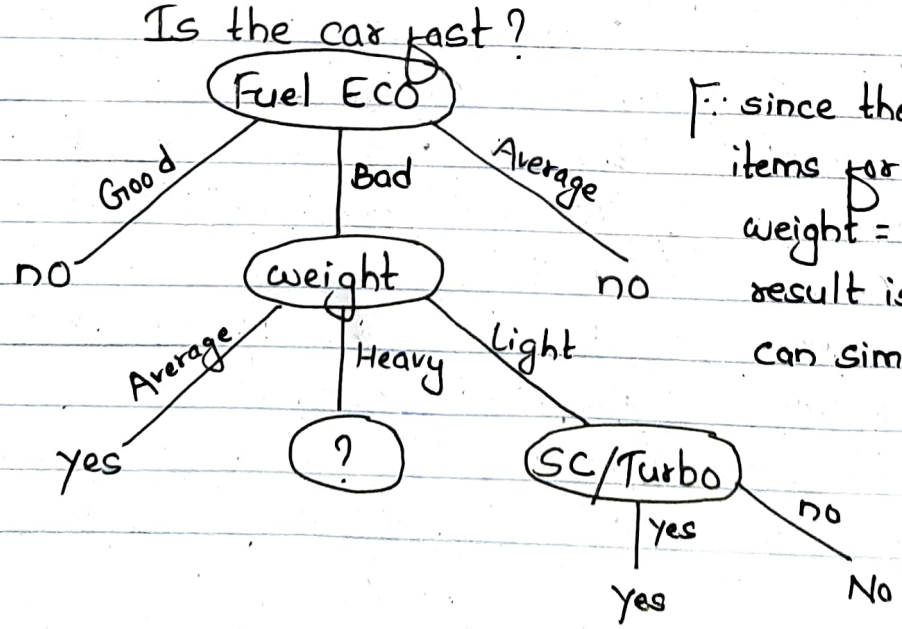
C4.5 Algorithm

- The C4.5 algorithm introduces a number of improvements over the original ID3 algorithm
- The C4.5 algorithm can handle missing data
- If the training records contain unknown attribute values, the C4.5 evaluates the gain for an attribute by considering only the records where the attribute is defined.
- Both categorical and continuous attributes are supported by C4.5
- Values of a continuous variable are sorted and partitioned.
- For the corresponding records of each partition, the gain is calculated, and the partition that maximizes the gain is chosen for the next split.
- The ID3 algorithm may construct a deep & complex tree, which would cause overfitting
- The C4.5 algorithm addresses the overfitting problem in ID3 by using a bottom-up technology called pruning to simplify the tree by removing the ~~least~~ least visited nodes & branches.

slideplayer.com

2. ID3 Example

Model	Engine	SC/Turbo	Weight	Fuel ECO	Fast
Prius	small	no	average	good	no
Civic	small	no	light	average	no
WRX STI	small	yes	average	bad	Yes
M3	medium	no	heavy	bad	Yes
RS4	large	no	average	bad	Yes
GTI	medium	no	light	bad	no
XJR	large	yes	heavy	bad	no
S500	large	no	heavy	bad	no
911	medium	yes	light	bad	Yes
Corvette	large	no	average	bad	Yes
Insight	small	no	light	good	no
RSX	small	no	average	average	no
IS350	medium	no	heavy	bad	no
MR2	small	yes	average	average	no
E320	medium	no	heavy	bad	no



∴ since there are only two items for SC/Turbo where weight = light, and the result is consistent, we can simplify the weight = Light path.]

Bayesian Classification:

Bayes Theorem:

$$P(H/X) = \frac{P(X/H)\,P(H)}{P(X)}$$

→ $P(H/X)$ - posterior probability

Naive Bayesian Classification:

$$P(C_i/X) = \frac{P(X/C_i)\,P(C_i)}{P(X)}$$

Q.1

RID	age	income	student	credit_rating	Class: buy_computer
1	Youth	high	no	fair	no
2	Youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	Youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	Youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

X = (age = youth, income = medium, student = yes, credit-rating = fair)
buy-computer = ?

sol^n:- Here, To compute $P(X|C_i)$, for $i = 1, 2$

P(buy_computer = yes) = 9/14 = 0.643
P(buy-computer = NO) = 5/14 = 0.357

P(age = youth | buys_computer = yes) = 2/9 = 0.222
P(age = youth | buys_computer = NO) = 3/5 = 0.600

P(income = medium | buys_computer = yes) = 4/9 = 0.444
P(income = medium | buys-computer = NO) = 2/5 = 0.400

P(student = yes | buys_computer = yes) = 6/9 = 0.667
P(student = yes | buys_computer = NO) = 1/5 = 0.200

P(credit-rating = fair | buys_computer = yes) = 6/9 = 0.667
P(credit_rating = fair | buys_computer = NO) = 2/5 = 0.400

Now;

$P(X|buys\_computer = yes) = P(age = youth | buys\_computer = yes)$
$* P(income = medium | buys\_computer = yes)$
$* P(student = yes | buys\_computer = yes)$
$* P(credit\_rating = fair | buys\_computer = yes)$
$= 0.222 * 0.444 * 0.667 * 0.667$
$= 0.044$

$P(X|buys\_computer = no) = 0.600 * 0.400 * 0.200 * 0.400$
$= 0.019$

- $P(X \mid buys\_computer = yes) * P(buys\_computer = yes)$

$= 0.044 * 0.643$

$= 0.028$

$P(X \mid buys\_computer = no) * P(buys\_computer = no)$

$= 0.019 * 0.357$

$= 0.007$

Therefore, naive bayesian classifier predicts "buys_computer = yes" for tuple X.

Q.2 (Bayesian Classifier)

Fruit	Yellow	Sweet	Long	Total
Mango	350	450	0	650
Banana	400	300	350	400
others	50	100	50	150
Total	800	850	400	1200

Fruit = { Yellow, sweet, long }

Soln:- Here;

$P(Fruit: Mango) = 650/1200 = 0.541$

$P(X \mid Mango) = P(Y \mid M) * P(S \mid M) * P(L \mid M)$

$$P(\text{Yellow} \mid \text{Mango}) = \frac{P(\text{Mango}/\text{Yellow}) \cdot P(\text{Yellow})}{P(\text{Mango})}$$

$$= \frac{350/800 * 800/1200}{650/1200}$$

$$= \frac{350}{650}$$

$$= 0.53$$

i) Mango

$$P(X \mid \text{Mango}) = P(Y/M) \cdot P(S/M) \cdot P(L/M)$$

$$= 0.53 * \frac{450/850 * \frac{850}{1200}}{\frac{650}{1200}} * \frac{\frac{400}{1200} * 0}{650/1200}$$

$$= 0.53 * 0.692 * 0$$

$$= 0$$

$\therefore P(S/M) = 0.692$

$\therefore P(L/M) = 0$

$\therefore P(X \mid \text{Mango}) = 0$

ii) Banana

$$P(\text{Banana}) = 400/1200 = 0.333$$

$$P(Y|B) = \frac{P(B/Y) * P(Y)}{P(B)} = \frac{400/800 * 800/1200}{400/1200}$$

$$= 1$$

$$P(S|B) = \frac{P(B|S) * P(S)}{P(B)}$$

$$= 0.75$$

$P(L|B) = 0.875$, then

$P(X|B) = P(Y|B) * P(S|B) * P(L|B)$

$\therefore P(X|B) = 0.65$

iii) others:

$P(Y|O) = 0.33$

$P(S|O) = 0.66$

$P(L|O) = 0.33$ then;

$P(X|O) = P(Y|O) * P(S|O) * P(L|O)$

$\therefore P(X|O) = 0.072$

re; $P(X|B)$ has higher probability value.

$\therefore$ Fruit = { Yellow, sweet, long } = Banana //

Association Rule:

- Support
- Confidence

$$\text{support} = \frac{(X \cup Y) \cdot \text{count}}{n} \qquad \text{Support} = \frac{n \cdot X}{n}$$

$$\text{confidence} = \frac{(X \cup Y) \cdot \text{count}}{X \cdot \text{count}}$$

$$\text{Confidence} = \frac{\text{support } (X \cup Y)}{\text{support } (X)}$$

Support = $\frac{2}{9} \times 100\% = 22.22\%$

Apriori Algorithm:

1. For the following given transaction dataset generate association rules using Apriori Algorithm. Consider the values as minimum support = 2 and minimum confidence = 70%.

Transaction ID	List of Items
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

soln:- Here,

C_1 (Candidates)

Item set	Support Count (frequency)
{A}	6
{B}	7
{C}	6
{D}	2
{E}	2

L_1 (support $\Rightarrow$ 2)

Item set	Support count
{A}	6
{B}	7
{C}	6
{D}	2
{E}	2

C_2 (Candidates pair)

Item set	support count
{A, B}	4
{A, C}	4
{A, D}	1
{A, E}	2
{B, C}	4
{B, D}	2
{B, E}	2
{C, D}	0
{C, E}	1
{D, E}	0

L_2 (Support $\Rightarrow$ 2)

Item set	Support Count
{A, B}	4
{A, C}	4
{A, E}	2
{B, C}	4
{B, D}	2
{B, E}	2

C_3

Item set	Support Count
{A,B,C}	2
{A,B,E}	2
{A,C,E}	1
{B,C,D}	0
{B,C,E}	1
{B,D,E}	0
{A,B,D}	1

L_3 (support $\Rightarrow$ 2)

Item set	Support count
{A,B,C}	2
{A,B,E}	2

C_4

Item set	Support Count
{A,B,C,E}	1
~~{A,B,E,D}~~	
~~{B,C,E,D}~~	

L_4 (support $\Rightarrow$ 2) $= \phi$

Since $L_4 = \phi$, we take selected combination i.e {A,B,C} and {A,B,E} i.e we consider rule for L_3.

For Combination {A,B,C}:,

X → Y	Confidence
{A,B} → {C}	AUBUC / AUB = 2/4 = 50%
{A,C} → {B}	2/4 = 50%
{B,C} → {A}	2/4 = 50%
{A} → {B,C}	2/6 = 33.33%
{B} → {A,C}	2/7 = 28.57%
{C} → {A,B}	2/6 = 33.33%

i.e $\frac{\text{support}(X \cup Y)}{\text{support}(X)}$

Since all the rules ~~do~~ are less than 70% of confidence, we don't accept these rules.

For the confidence of combination {A, B, E}

	Confidence
{A, B} → {E}	AUBUE / AUB = 2/4 = 50%
{A, E} → {B}	2/2 = 100%
{B, E} → {A}	2/2 = 100%
{A} → {B, E}	2/6 = 33.33%
{B} → {A, E}	2/7 = 28.57%
{E} → {A, B}	2/2 = 100%

Final Association Rules; from above calculations, we get following rules: (as their confidence is greater than 70%)

{A, E} → {B}
{B, E} → {A}
{E} → {A, B} //

Apriori property → All non-empty subsets of a frequent item sets must also be frequent.

2. Generate Association Rules using Apriori Algorithm.

Minimum support = 3

Minimum Confidence = 70%

Transaction ID	List of Items
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, Or, K, I, E}

Soln:- Here;

C_1 (Candidates)

Item Set	Support Count
{M}	3
{O}	3
{N}	2
{K}	5
{E}	4
{Y}	3
{D}	1
{A}	1
{U}	1
{C}	2
{Or}	1
{I}	1

L_1 (Support ⇒ 3)

Item Set	Support Count
{M}	3
{O}	3
{K}	5
{E}	4
{Y}	3

C_2 (Candidate Pairs)

Item set	Support Count
{M,O}	1
{M,K}	3
{M,E}	2
{M,Y}	2
{O,K}	3
{O,E}	3
{O,Y}	2
{K,E}	4
{K,Y}	3
{E,Y}	2

L_2 (support $\Rightarrow$ 3)

Item set	Support Count
{M,K}	3
{O,K}	3
{O,E}	3
{K,E}	4
{K,Y}	3

C_3 (Candidates)

Item set	Support Count
{M,O,K}	1
{K,O,E}	3
{K,E,Y}	2
{M,K,E}	2
{M,K,Y}	2
{O,K,Y}	2

L_3 (support $\Rightarrow$ 3)

Item Set	Support count
{K,O,E}	3

We select combination {K,O,E} for association rule.

For Confidence of Combination {K, O, E}

	Confidence
{K, O} → {E}	KUOUE / KUO = 3/3 = 100%
{K, E} → {O}	3/4 = 75%
{O, E} → {K}	3/3 = 100%
{K} → {O, E}	3/5 = 60%
{O} → {K, E}	3/3 = 100%
{E} → {K, O}	3/4 = 75%

Hence the required final association rule from above calculations are as follows:

{K, O} → {E} = 100%
{K, E} → {O} = 75%
{O, E} → {K} = 100%
{O} → {K, E} = 100%
{E} → {K, O} = 75% //

3. Support = 50%, Confidence = 75%

TID	Items Purchased
1	Bread, Cheese, Egg, Juice
2	Bread, Cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	Cheese, Juice, Milk

Soln:- Here; C_1 (Candidates)

Item set	Support Count	Support
Bread	4	4/5 = 80%
Cheese	3	3/5 = 60%
Egg	1	1/5 = 20%
Juice	4	4/5 = 80%
Yogurt	1	1/5 = 20%
Milk	3	3/5 = 60%

L_1 (support => 50%)

Item set	Support Count
Bread	80%
cheese	60%
Juice	80%
Milk	60%

C_2 (Candidate pairs)

Item set	Support Count	Support
{B, C}	2	2/5 = 40%
{B, J}	3	3/5 = 60%
{B, M}	2	2/5 = 40%
{C, J}	3	3/5 = 60%
{C, M}	1	1/5 = 20%
{J, M}	2	2/5 = 40%

L_2 (support => 50%)

Item set	Support
{B, J}	60%
{C, J}	60%

C_3

Itemset	Support Count	Support
{B, C, J}	1	1/5 = 20%

L_3 (support $\Rightarrow$ 50%) $= \phi$

Since $L_3 = \phi$, we take selected combination i.e {B, J} and {C, J} i.e rule. L_2.

For confidence of combination {B, J}

	confidence
{B} $\rightarrow$ {J}	BUJ/B = 3/4 = 75%

For confidence of combination {C, J}

	confidence
{C} $\rightarrow$ {J}	CUJ/C = 3/3 = 100%

Hence the required final association rule from the above calculation are as follows:

{B} $\longrightarrow$ {J} = 75%
{C} $\longrightarrow$ {J} = 100%

//

4. Minimum Support = 30%

TID	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

Soln:- Here,

C_1 (candidates)

Item set	Support Count	Support
{A}	5	5/8 = 62.5%
{B}	6	6/8 = 75%
{C}	3	3/8 = 37.5%
{D}	6	6/8 = 75%
{E}	4	4/8 = 50%

L_1 (support = 30%)

Item set	Support
{A}	62.5%
{B}	75%
{C}	37.5%
{D}	75%
{E}	50%

C_2 (Candidate Pairs)

~~{A~~ Item set	Support count	support
{A, B}	4	4/8 = 50%
{A, C}	2	2/8 = 25%
{A, D}	4	4/8 = 50%
{A, E}	4	4/8 = 50%
{B, C}	3	3/8 = 37.5%

C_2 (Candidate Pairs)

Item set	Support count	support
{B, D}	4	4/8 = 50%
{B, E}	3	3/8 = 37.8%
{C, D}	1	1/8 = 12.5%
{C, E}	2	2/8 = 25%
{D, E}	3	3/8 = 37.8%

L_2 (support ⇒ 30%)

Item set	support
{A, B}	50%
{A, D}	50%
{A, E}	50%
{B, C}	37.5%
{B, D}	50%
{B, E}	37.8%
{D, E}	37.8%

C_3 (Candidates)

Item set	Support count	support
{A, B, D}	3	3/8 = 37.5%
{A, D, E}	3	3/8 = 37.5%
{A, B, C}	2	2/8 = 25%
{B, C, D}	1	1/8 = 12.5%
{D, B, E}	2	2/8 = 25%
{B, C, E}	2	2/8 = 25%
{A, B, E}	3	3/8 = 37.5%

L_3 (support ⇒ 30%)

Item set	Support
{A, B, D}	37.5%
{A, D, E}	37.5%
{A, B, E}	37.5%

C_4 (Candidates)

Item set	Support count	support
{A, B, D, E}	2	2/8 = 25%

$L_4 = \phi$

Since $L_4 = \phi$, we take selected combination i.e {A, B, D}, {A, D, E} and {A, B, E} i.e we consider rule for L_3.

FP Growth:

1.

TID	Items
T100	M, O, N, K, E, Y
T200	D, O, N, K, E, Y
T300	M, A, K, E
T400	M, U, C, K, Y
T500	C, O, K, I, E

Minimum Support = 3

Soln:- Here;

C1 (candidates)

Item set	Support count
{M}	3
{O}	3
{N}	2
{K}	5
{E}	4
{Y}	3
{D}	1
{A}	1
{U}	1
{C}	2
{I}	1

L1 (Support ⇒ 3)

Item set	Support count
{M}	3
{O}	3
{K}	5
{Y}	3
{E}	4

Items	Ordered items with high frequency	Priority
{K}	5	1
{E}	4	2
{M}	3	3
{O}	3	4
{Y}	3	5

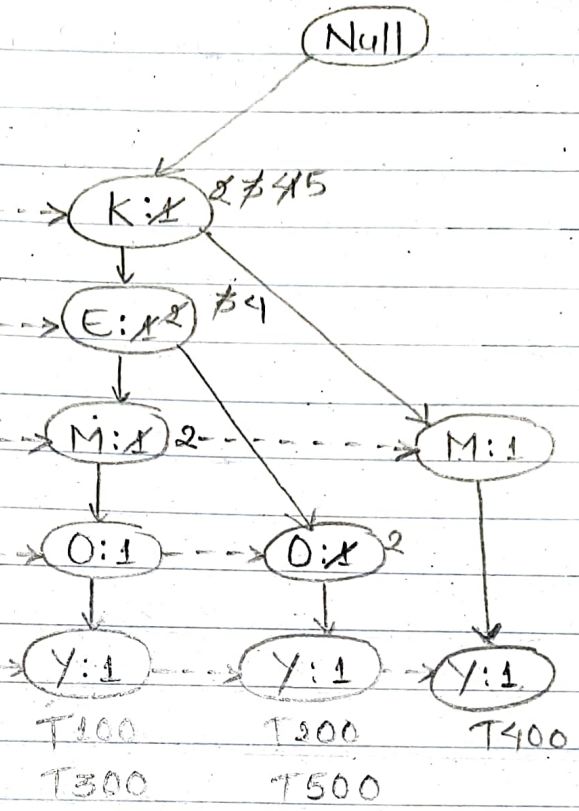
TID	Items	ordered items with priority
T100	{M,O,N,K,E,Y}	{K, E, M, O, Y}
T200	{D,O,N,K,E,Y}	{K, E, O, Y}
T300	{M,A,K,E}	{K, E, M}
T400	{M,U,C,K,Y}	{K, M, Y}
T500	{C,O,K,I,E}	{K, E, O}

Generating FP Tree:

Freq. Item	Support count
{K}	5
{E}	4
{M}	3
{O}	3
{Y}	3

FP Tree:

Freq. Item	Support count
{K}	5
{E}	4
{M}	3
{O}	3
{Y}	3



Mining FP Tree:

Items (low priority)	Conditional Pattern Base	Conditional FP Tree	Frequent Pattern Generated
{Y}	{{K,E,M,O:1}, {K,E,O:1}, {K,M:1}}	<K:3>	{K,Y:3}
{O}	{{K,E,M:1}, {K,E:2}}	<K:3, E:3>	{K,O:3},{E,O:3}{K,E,O:3}
{M}	{{K,E:2}, {K:1}}	<K:3>	{K,M:3}
{E}	{K:4}	<K:4>	{K,E:4}
{K}	-		

2. Minimum Support = 30% = $\frac{30}{100} \times 8 = 2.4 \approx 3$

TID	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

Soln:- Here;

C1 (candidates)

Item set	Support count
{A}	5
{B}	6
{C}	3
{D}	6
{E}	4

L1 (support ⇒ 3)

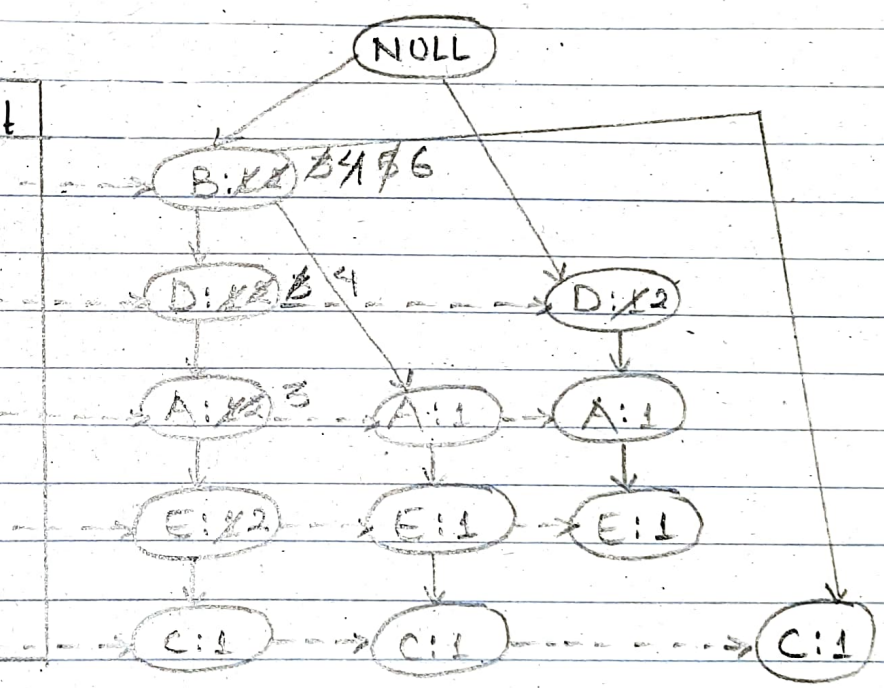
Item set	Support
{A}	5
{B}	6
{C}	3
{D}	6
{E}	4

Items	ordered items with high freq.	Priority
B	6	1
D	6	2
A	5	3
E	4	4
C	3	5

TID	Items	ordered items with priority
1	{E, A, D, B}	{B, D, A, E}
2	{D, A, C, E, B}	{B, D, A, E, C}
3	{C, A, B, E}	{B, A, E, C}
4	{B, A, D}	{B, D, A}
5	{D}	{D}
6	{D, B}	{B, D}
7	{A, D, E}	{D, A, E}
8	{B, C}	{B, C}

Generating FP Tree:

Frequent Item	Support count
B	6
D	6
A	5
E	4
C	3



Mining FP Tree:

Item (Low priority ↓)	Conditional Pattern Base	Conditional FP Tree	Frequent Pattern Generated
{C}	{{B,D,A,E:1}, {B,A,E:1}, {B:1}}	<B:3>	{B,C:3}
{E}	{{B,D,A:2}, {B,A:1}, {D,A:1}}	<B:3, D:3, A:4>	
{A}	{{B,D:3}, {B:1}, {D:1}}	<B:4, D:4>	{B,A:4}, {D,A:4} {B,D,A:3}
{D}	{B:4}	<B:4>	{B,D:4}
{B}	-		

K-mean: (One point)

1. Using K-mean clustering algorithm, divide the given dataset into 2 Clusters($K=2$).

Data	$m_1 = 4$ (|4-2|=2)	$m_2 = 12$ (|12-2|=10)	|m-d|
2	2	10	m_1
3	1	9	m_1
(4)	0	8	m_1
10	6	2	m_2
11	7	1	m_2
(12)	8	0	m_2
20	16	8	m_2
25	21	13	m_2
30	26	18	m_2

soln:- Let $m_1 = 2$ & $m_2 = 12$.

Then;

Clusters formed are;

$K_1 = \{2, 3, 4\}$ & $K_2 = \{10, 11, 12, 20, 25, 30\}$

$$m_1(new) = \frac{2+3+4}{3} = 3 \qquad m_2 = \frac{10+11+12+20+25+30}{6} = 18$$

Iteration 2:

$K_1 = \{2, 3, 4, 10\}$ $\qquad$ $K_2 = \{11, 12, 20, 25, 30\}$

$m_1(new) = 4.75 \approx 5$ $\qquad$ $m_2 = 19.6 \approx 20$

Iteration 3:

data	$m_1 = 5$	$m_2 = 20$	
2	3	18	m_1
3	2	17	m_1
4	1	16	m_1
10	5	10	m_1
11	6	9	m_1
12	7	8	m_1
20	15	0	m_2
25	20	5	m_2
30	25	10	m_2

$K_1 = \{2, 3, 4, 10, 11, 12\}$ $K_2 = \{20, 25, 30\}$

$m_1(new) = 7$ $m_2(new) = 25$

Iteration 4:

Data	$m_1 = 7$	$m_2 = 25$	
2	5	23	m_1
3	4	22	m_1
4	3	21	m_1
10	3	15	m_1
11	4	14	m_1
12	5	13	m_1
20	13	5	m_2
25	18	0	m_2
30	23	5	m_2

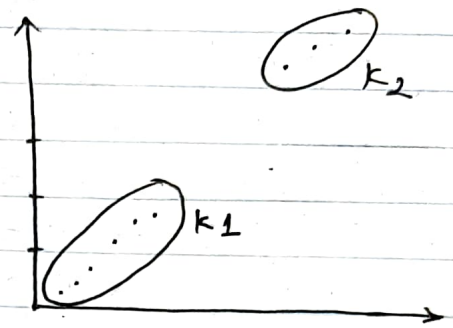
$K_1 = \{2, 3, 4, 10, 11, 12\}$ $K_2 = \{20, 25, 30\}$

Since the clusters k_1 & k_2 formed in iteration 4 are same as of iteration 3, so we stop here.

∴ Final clusters are;

$k_1 = \{2, 3, 4, 10, 11, 12\}$

$k_2 = \{20, 25, 30\}$



2. Use k-mean algorithm;

Data = {15, 15, 16, 19, 19, 20, 20, 21, 22, 28, 35, 40, 41, 42, 43, 44, 60, 61, 65}

K = 2 and centroid ($c_1 = 16$, $c_2 = 22$)

Soln:- Here;

$K = 2$

$c_1 = m_1 = 16$

$c_2 = m_2 = 22$

Iteration 1:

Data	$\lvert m_1 - d \rvert$	$\lvert m_2 - d \rvert$	
15	1	7	m_1
15	1	7	m_1
(16)	0	6	m_1
19	3	3	m_1
19	3	3	m_1
20	4	2	m_2
20	4	2	m_2
21	5	1	m_2
(22)	6	0	m_2
28	12	6	m_2
35	19	13	m_2
40	24	18	m_2
41	25	19	m_2
42	26	20	m_2
43	27	21	m_2
44	28	22	m_2
60	44	38	m_2
61	45	39	m_2
65	49	43	m_2

$K_1 = \{15, 15, 16, 19, 19\}$

$K_2 = \{20, 20, 21, 22, 28, 35, 40, 41, 42, 43, 44, 60, 61, 65\}$

$m_1(new) =$

$m_2(new) =$

Minkowski Distance:

$$d(i,j) = q\sqrt{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{in} - x_{jn}|^q}$$

If $q=1$, d is Manhattam distance,

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{in} - x_{jn}|$$

If $q=2$, d is Eucidean distance,

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{in} - x_{jn}|}$$

Euclidean Distance:

(X, Y) & $C_1(x, y)$

Euclidean Distance $= \sqrt{(X_x - x_1)^2 + (Y_y - y_1)^2}$

X_x: observed value

x_1: Actual value

Two Point K-mean:

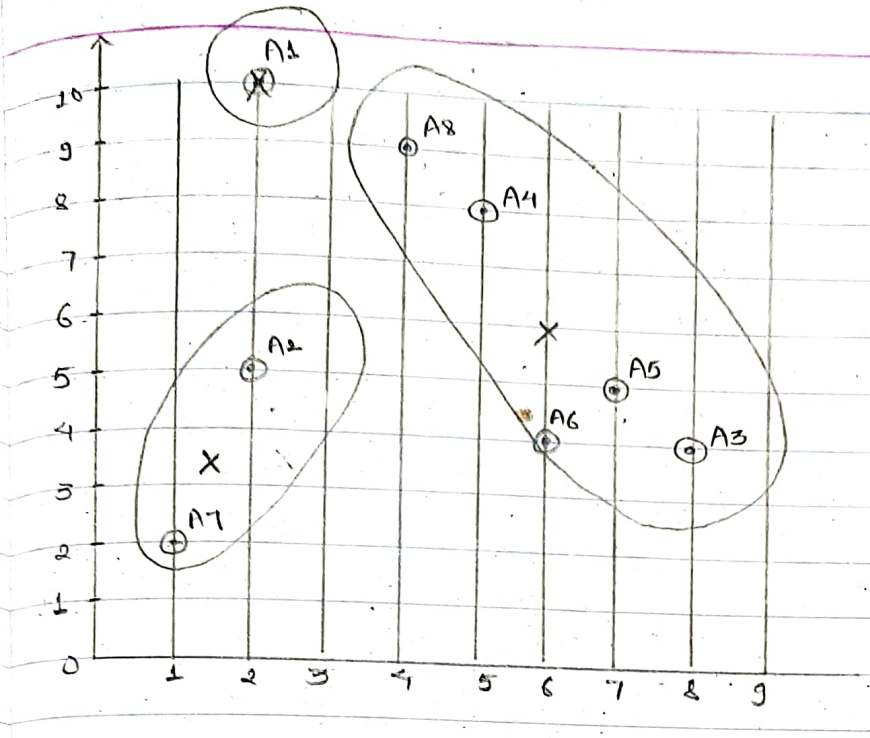
1. Cluster the following 8 points (with (x,y) representing locations) into 3 clusters $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$ and $A_8(4,9)$. Initial cluster centers are: $A_1(2,10)$, $A_4(5,8)$ and $A_7(1,2)$. The distance function between two points $a=(x_1,y_1)$ and $b(x_2,y_2)$ is defined as: $f(a,b) = |x_2-x_1| + |y_2-y_1|$. Use k-means algorithm to find the three cluster centers.

Soln:- Here;

Iteration 1:

	Point	Distance mean 1 (2,10)	Distance mean 2 (5,8)	Distance mean 3 (1,2)	Cluster
A_1	(2,10)	0	5	9	m_1 / 1
A_2	(2,5)	5	6	4	m_3 / 3
A_3	(8,4)	12	7	9	m_2
A_4	(5,8)	5	0	10	m_2
A_5	(7,5)	10	5	9	m_2
A_6	(6,4)	10	5	7	m_2
A_7	(1,2)	9	10	0	m_3
A_8	(4,9)	3	2	10	m_2

cluster 1	cluster 2	cluster 3
$A_1(2,10)$	$A_3(8,4)$	$A_2(2,5)$
	$A_4(5,8)$	$A_7(1,2)$
	$A_5(7,5)$	
	$A_6(6,4)$	
	$A_8(4,9)$	
new (2,10)	new (6,6)	new (1.5,3.5)



Iteration 2:

	Point	mean1 (2,10)	mean2 (6,6)	mean3 (1.5,3.5)	cluster
A1	(2,10)	0	8	7	m_1
A2	(2,5)	5	5	2	m_3
A3	(8,4)	12	4	7	m_2
A4	(5,8)	5	3	8	m_2
A5	(7,5)	10	2	7	m_2
A6	(6,4)	10	2	5	m_2
A7	(1,2)	9	9	2	m_3
A8	(4,9)	3	5	8	m_1

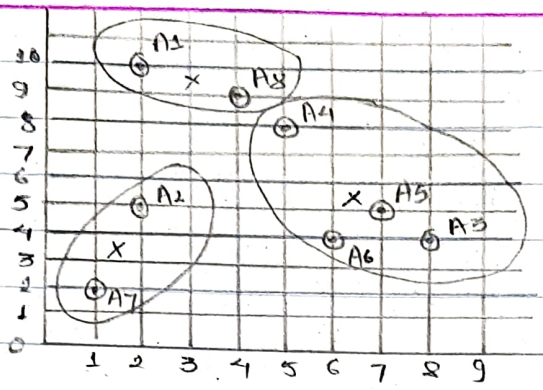
Cluster 1	cluster 2	cluster 3
A1(2,10)	A3(8,4)	A2(2,5)
A8(4,9)	A4(5,8)	A7(1,2)
	A5(7,5)	
	A6(6,4)	

New mean:

cluster 1 = (3, 9.5)

cluster 2 = (6.5, 5.25)

cluster 3 = (1.5, 3.5)



Iteration 3:

	Point	mean 1 (3,9.5)	mean 2 (6.5,5.25)	mean 3 (1.5, 3.5)	Cluster
A1	(2,10)	1.5	9.25	7	m1
A2	(2,5)	5.5	4.75	2	m3
A3	(8,4)	10.5	2.75	7	m2
A4	(5,8)	3.5	4.25	8	m1
A5	(7,5)	8.5	0.75	7	m2
A6	(6,4)	8.5	1.75	5	m2
A7	(1,2)	9.5	8.75	2	m3
A8	(4,9)	1.5	6.25	8	m1

~~A9~~

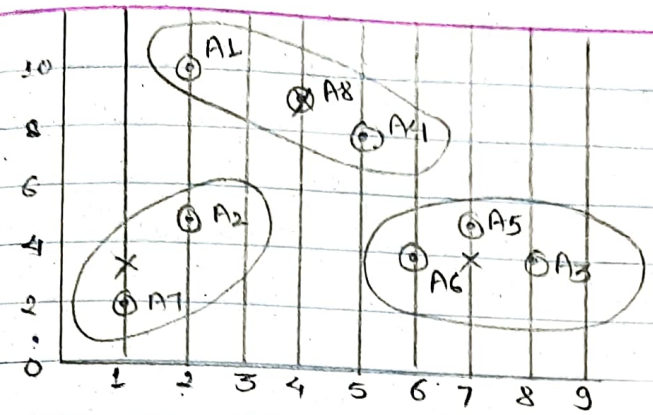
cluster 1	cluster 2	cluster 3
A1(2,10)	A3(8,4)	A2 (2,5)
A4 (5,8)	A5(7,5)	~~A6 (6,4)~~
A8(4,9)	A6 (6,4)	A7 (1,2)

New mean:

mean 1 = (3.67, 9) ≈ (4, 9)

mean 2 = (7 , 4.33) ≈ (7, 4)

mean 3 = (1.5, 3.5)



Iteration 4:

	Point	mean 1 (4,9)	mean 2 (7,4)	mean 3 (1.5, 3.5)	Cluster
A1	(2,10)	3	11	7	m1
A2	(2,5)	6	6	2	m3
A3	(8,4)	9	1	7	m2
A4	(5,8)	2	6	8	m1
A5	(7,5)	7	1	7	m2
A6	(6,4)	7	1	5	m2
A7	(1,2)	10	8	2	m3
A8	(4,9)	0	8	8	m1

Cluster 1	cluster 2	cluster 3
A1(2,10)	A3(8,4)	A2(2,5)
A4(5,8)	A5(7,5)	A7(1,2)
A8(4,9)	A6(6,4)	

mean 1 = (4,9)

mean 2 = (7,4)

mean 3 = (1.5, 3.5)

Since the clusters formed in iteration 3 & 4 are same, we stop here.

2. Using K-mean clustering algorithm, divide the given dataset into 2 clusters. Use Euclidean distance.

ID	X	Y
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

$$E.C.D = \sqrt{|x_2 - x_1|^2 + |y_2 - y_1|^2}$$

Soln:- Here, Iteration 1:

ID	X	Y	ECD1 (3,4)	ECD2 (4.5,5)	Cluster
1	1	1	3.60	5.31	ECD1
2	1.5	2	2.5	4.24	ECD1
3	3	4	0	1.80	ECD1
4	5	7	3.6	2.06	ECD2
5	3.5	5	1.11	1	ECD1
6	4.5	5	1.80	0	ECD2
7	3.5	4.5	0.70	1.11	ECD1

Cluster 1	cluster 2
1 (1,1)	4 (5,7)
2 (1.5,2)	6 (4.5,5)
3 (3,4)	
5 (3.5,5)	
7 (3.5,4.5)	

new Mean1 = (2.5, 3.3)
new Mean2 = (4.5, 6)

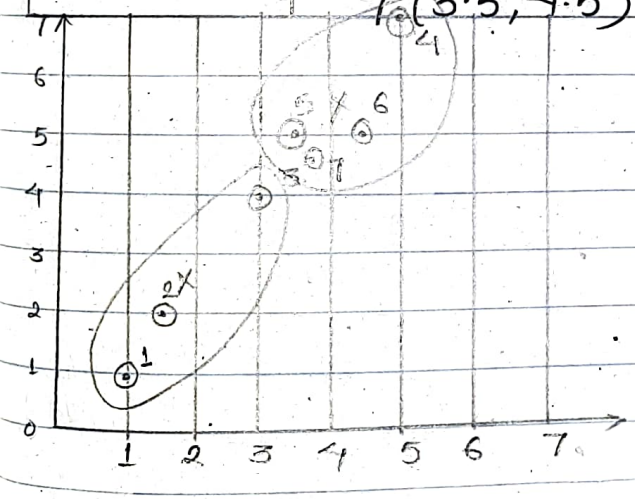
Iteration 2:

ID	X	Y	ECD1 (2.5, 3.3)	ECD2 (4.5, 5)	Cluster
1	1	1	2.74	5.31	~~m2~~ ECD1
2	1.5	2	1.64	4.24	ECD1
3	3	4	0.86	1.80	ECD1
4	5	7	4.46	2.06	ECD2
5	3.5	5	1.97	1	ECD2
6	4.5	5	2.62	0	ECD2
7	3.5	4.5	1.56	1.11	ECD2

Cluster 1	Cluster 2
1 (1, 1)	4 (5, 7)
2 (1.5, 2)	5 (3.5, 5)
3 (3, 4)	6 (4.5, 5)
	7 (3.5, 4.5)

new Mean1 = (1.8, 2.3)

new Mean2 = (4.25, 5.37)



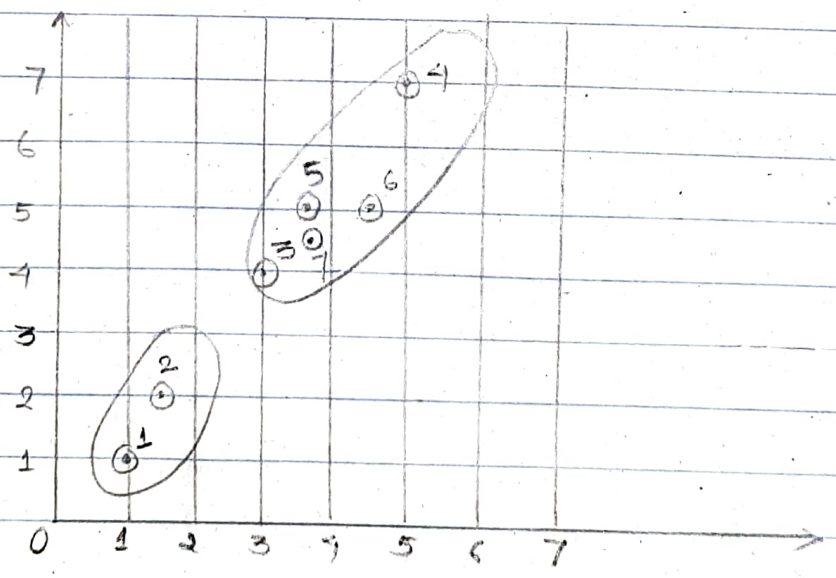
Iteration 3:

ID	X	Y	ECD1 (1.8, 2.3)	ECD2 (4.12, 5.37)	Cluster
1	1	1	1.5	5.4	ECD1
2	1.5	2	0.4	4.3	ECD1
3	3	4	2.1	1.8	ECD2
4	5	7	5.6	1.9	ECD2
5	3.5	5	3.2	0.7	ECD2
6	4.5	5	3.8	0.5	ECD2
7	3.5	4.5	2.8	1.1	ECD2

Cluster 1	Cluster 2
1(1, 1)	3(3,4)
2(1.5, 2)	4(5,7)
	5(3.5,5)
	6(4.5,5)
	7(3.5,4.5)

New mean1: (1.25, 1.5)

New mean2 = (3.9, 5.1)



Iteration 4:

ID	X	Y	ECD1 (1.25, 1.5)	ECD2 (3.9, 5.1)	Cluster
1	1	1	0.5	5	ECD_1
2	1.5	2	0.5	3.9	ECD_1
3	3	4	3.1	1.4	ECD_2
4	5	7	6.7	2.2	ECD_2
5	3.5	5	4.2	0.4	ECD_2
6	4.5	5	4.8	0.6	ECD_2
7	3.5	4.5	3.8	0.7	ECD_2

Cluster 1	cluster 2
1 (1, 1)	3 (3,4)
2 (1.5, 2)	4 (5,7)
	5 (3.5, 5)
	6 (4.5, 5)
	7 (3.5, 4.5)

Since the clusters formed in iteration 3 & 4 are same, we stop here.

3. Using K-means clustering algorithm, divide the given dataset into 2 clusters. Use Euclidean distance.

Height	Weight
185	72
170	56
168	60
179	68
182	72
188	77
180	71
180	8070
183	84
180	88
180	67
177	76

K-Medoid:

i	x	y
x_1	2	6
x_2	3	4
x_3	3	8
x_4	4	7
x_5	6	2
x_6	6	4
x_7	7	3
x_8	7	4
x_9	8	5
x_{10}	7	6

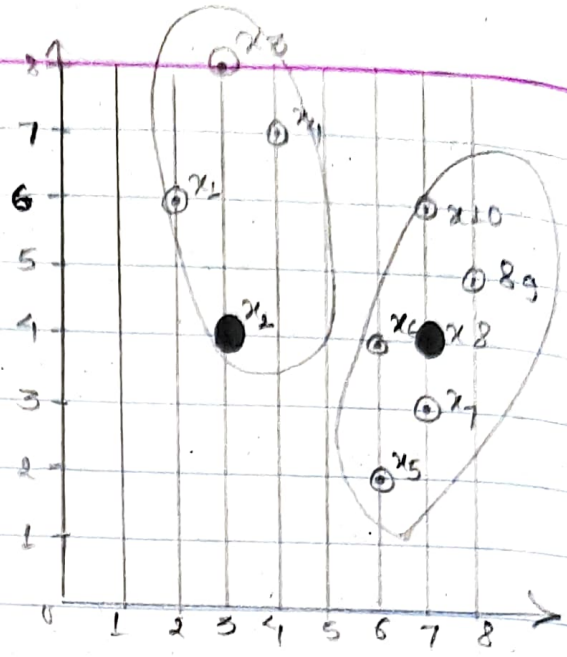
Take k=2 and use Manhattan distance.

soln:- Here, Step 1:

Iteration 1:

i	x	y	Distance for me1 (3,4)	Distance for me2 (7,4)	clusters
x_1	2	6	3	7	me1
x_2	3	4	0	4	me1
x_3	3	8	4	8	me1
x_4	4	7	4	6	me1
x_5	6	2	5	3	me2
x_6	6	4	3	1	me2
x_7	7	3	5	1	me2
x_8	7	4	4	0	me2
x_9	8	5	6	2	me2
x_{10}	7	6	5	2	me2

Cluster 1	Cluster 2
x_1 (2,6)	x_5 (6,2)
x_2 (3,4)	x_6 (6,4)
x_3 (3,8)	x_7 (7,3)
x_4 (4,7)	x_8 (7,4)
	x_9 (8,5)
	x_{10} (7,6)



step 2:

Calculate the total cost, $Tcost(x,c) = \sum_{x=i}^{d} |x_i - c_i|$

Current Total cost: cost{(2,6),(3,4)} + cost{(3,4),(3,8)} +
cost{(3,4),(4,7)} + cost{(7,4),(6,2)} +
cost{(7,4),(6,4)} + cost{(7,4),(7,3)} +
cost{(7,4),(8,5)} + cost{(7,4),(7,6)}

$= (3+4+4) + (3+1+1+2+2)$

$= 20$

step 3:

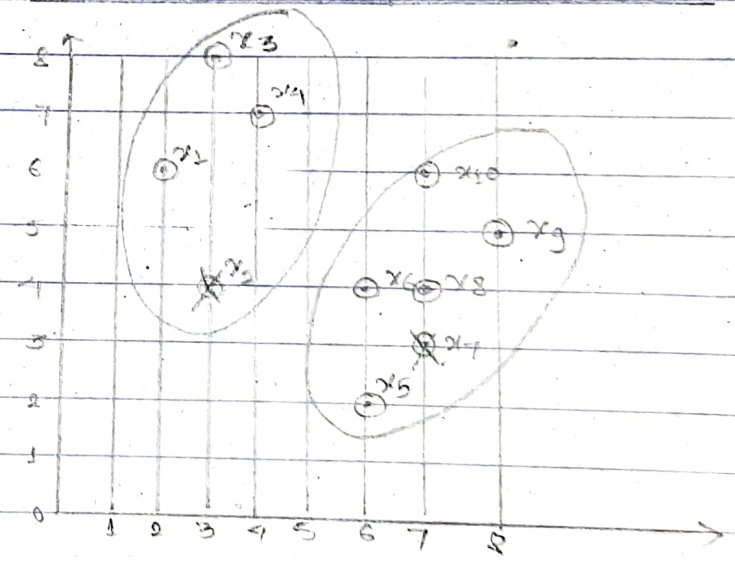
Select one of non-medoid O' and let O' be (7,3).

Step 4:

Iteration 2:

i	x	y	Distance for me1 (3,4)	Distance for me2 (7,3)	clusters
x_1	2	6	3	8	me1
x_2	3	4	0	5	me1
x_3	3	8	4	9	me1
x_4	4	7	4	7	me1
x_5	6	2	5	2	me2
x_6	6	4	3	2	me2
x_7	7	3	5	0	me2
x_8	7	4	4	1	me2
x_9	8	5	6	3	me2
x_{10}	7	6	6	3	me2

Cluster 1	cluster 2
x_1 (2,6)	x_5 (6,2)
x_2 (3,4)	x_6 (6,4)
x_3 (3,8)	x_7 (7,3)
x_4 (4,7)	x_8 (7,4)
	x_9 (8,5)
	x_{10} (7,6)



Step 5:

Current Total Cost = (3+4+4) + cost {(7,3), (6,2)} +
cost {(7,3), (6,4)} + cost {(7,3), (7,4)} +
cost {(7,3), (8,5)} + cost {(7,3), (7,6)}

= 11 + (2+2+1+3+3)

= 22

Step 6:

Now, cost of swapping medoid me_2 to O';

S = total cost (me_1, O') - total cost (me_1, me_2)

$= 22 - 20$

$= 2 > 0$

So moving O' would be a bad idea as $S > 0$. Hence previous choice was good and the final clusters with medoids (3,4) and (7,4) we taken.

2. Take k=2 and use Manhattan distance for following dataset:
{(8,7), (3,7), (4,9), (9,6), (8,5), (5,8), (7,3), (8,4), (7,5), (4,5)}.
Use K-Medoid Method / algorithm.

Example of K-Medoids:

1. PAM (Partitioning Around Medoids) (for small datasets)
2. CLARA (Clustering LArge Applications) } for large dataset
3. CLARANS (Clustering LARge Applications based upon Randomized Search) }

K-Mean Vs. k-Medoid.

K-Medoid	K-Mean
1. The complexity of each iteration is; $O(K(n-k)^2)$.	The complexity of each iteration is; $O(K(n-k)^2)$.
For large values of n & k, such computation becomes very costly.	For large values of n & k, such computation becomes very costly.
2. K-Medoid method is more robust than k-Means in the presence of noise and outliers.	K-Mean is less robust than K-Medoid.
3. K-Medoid is more costly.	K-Mean is less costly.
4. Like k-Means, K-Medoids require the user to specify k.	K-Mean requires the user to specify k.

K-Medoids Vs K-Means:

- The complexity of each iteration is $O(k(n-k)^2)$.
- For large values of n and k, such computations becomes very costly.

Advantages:

- K-Medoid method is more robust than k-Means in the presence of noise and outliers.

Disadvantages:

- K-Medoid is more costly than the k-Mean method.
- Like k-Mean, k-Medoid requires the user to specify k.
- It does not scale well for large data sets.

Hierarchial Method:

1. Agglomerative
2. Divisive

1. Perform agglomerative algorithm on the following data and plot a dendogram using single link approach. The given data indicates the distances between elements.

Elements	E	A	C	B	D
E	0				
A	1	0			
C	2	2	0		
B	2	5	1	0	
D	3	3	6	3	0

Pair (E, A)
⇒ (1)

Soln:- Here;

	(E,A)	C	B	D
(E,A)	0			
C	2	0		
B	2	1	0	
D	3	6	3	0

Pair (B, C)
⇒ (1)

	(E,A)	(B,C)	D
(E,A)	0		
(B,C)	2	0	
D	3	3	0

Pair ((E,A), (B,C))
⇒ (2)

	(E,A,B,C)	D
(E,A,B,C)	0	
D	3	0

Plotting into Dendogram;

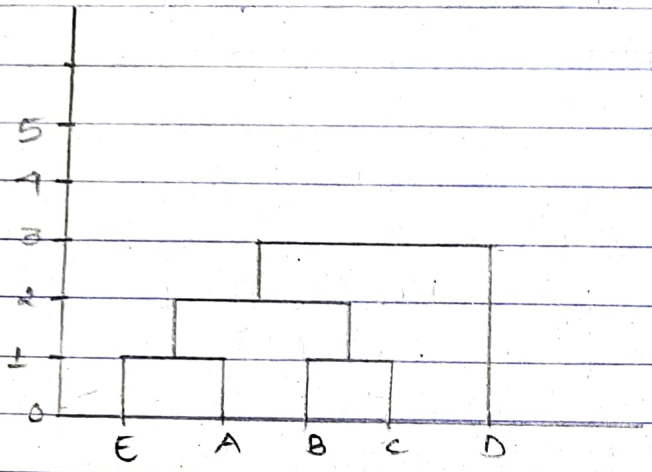


Fig: Dendogram

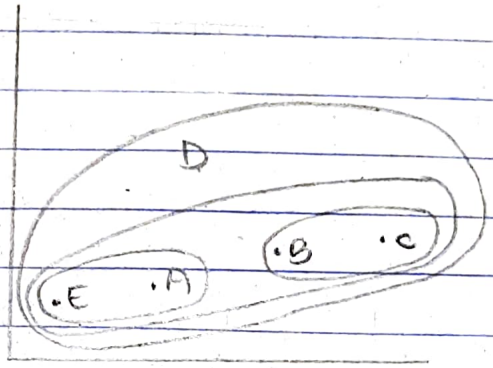


Fig: Clusters

2.

Item	A	B	C	D	E
A	0				
B	9	0			
C	3	7	0		
D	6	5	9	0	
E	11	10	(2)	8	0

Pair (C,E)
⇒ (2)

soln:- Here;

Items	(C,E)	A	B	D
(C,E)	0			
A	3	0		
B	7	9	0	
D	8	6	5	0

Pair (C,E,A)
⇒ (3)

Items	(C,E,A)	B	D
(C,E,A)	0		
B	7	0	
D	6	5	0

Pair (B,D)
⇒ (5)

Items	(A,C,E)	(B,D)
(A,C,E)	0	
(B,D)	6	0

$\min((A,B),(C,B),(E,B),(A,D),(C,D),(E,D))$
$= \min(9,7,0,6,9,8)$
$= \min(6)$

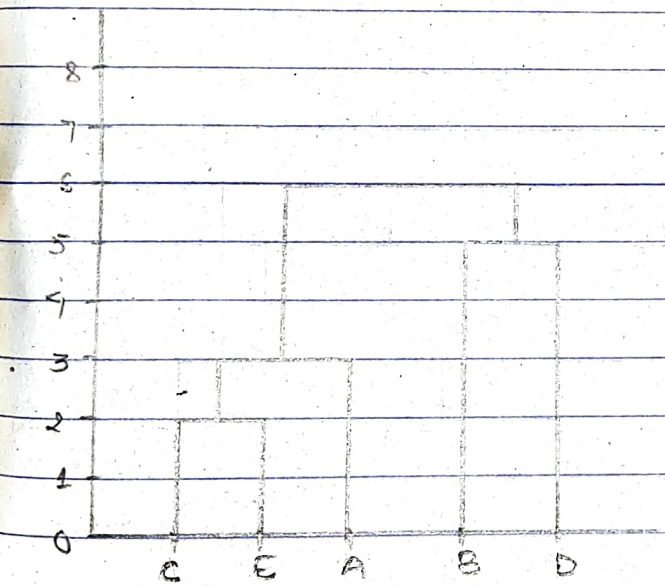


Fig: Dendogram

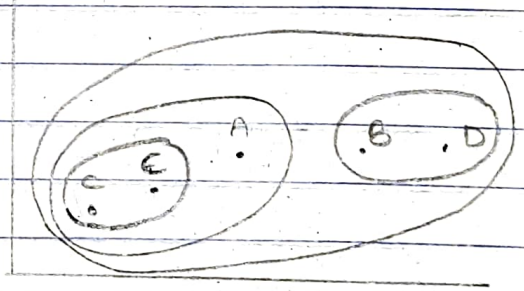


Fig: Cluster

3.

Elements	A	B	C	D	E	F
A	0					
B	0·71	0				
C	5·66	4·95	0			
D	3·61	2·92	2·24	0		
E	4·24	3·54	1·41	1·00	0	
F	3·20	2·50	2·50	0·50	1·12	0

Soln:- Here,

Since pair (D,F) has minimum value 0·50, we select (D,F) as pair.

⇓ Pair (D,F)
(0·50)

Elements	(D,F)	A	B	C	E
(D,F)	0				
A	3·20	0			
B	2·50	0·71	0		
C	2·24	5·66	4·95	0	
E	1·00	4·24	3·54	1·41	0

For (D,F) & (D,F);
= min((D,D),(D,F),(F,D),(F,F))
= min(0, 0·50, 0)
= 0

For (D,F) & A;
= min((D,A),(F,A))
= (3·61, 3·20)
= 3·20

The pair (B,A) has the minimum value of 0.71.

Elements	(B,A)	(D,F)	C	E
(B,A)	0			
(D,F)	2.50	0		
C	4.95	2.24	0	
E	3.54	1.00	1.41	0

⇓ pair (D,E,F)
(1.00)

Elements	(A,B)	C	(D,E F)
(A,B)	0		
C	4.95	0	
(D,E,F)	2.50	1.41	0

⇓ pair (C,D,E,F)
(1.41)

Elements	(A,B)	(C,D,E,F)
(A,B)	0	
(C,D,E,F)	2.50	0

Plotting into Dendogram;

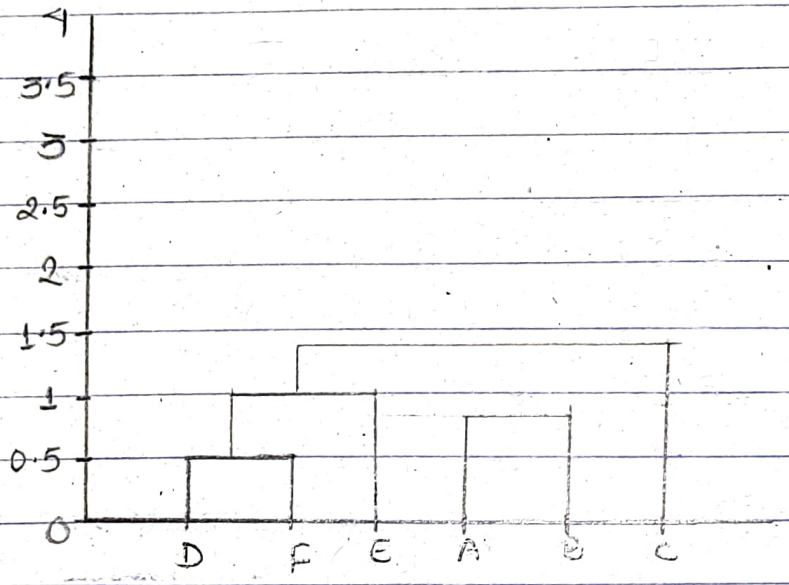


Fig: Dendogram

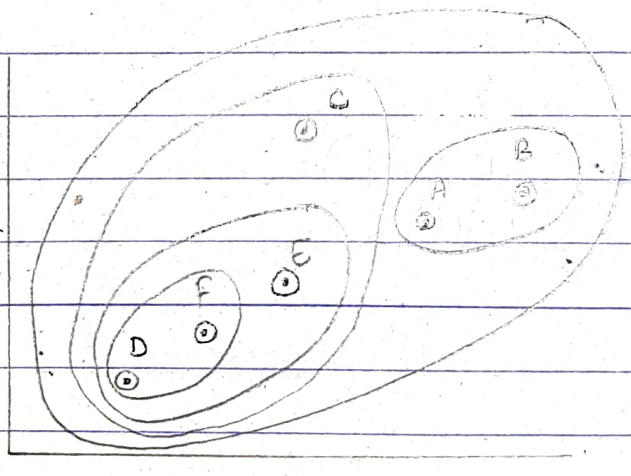


Fig: Clusters