1.
(a.)

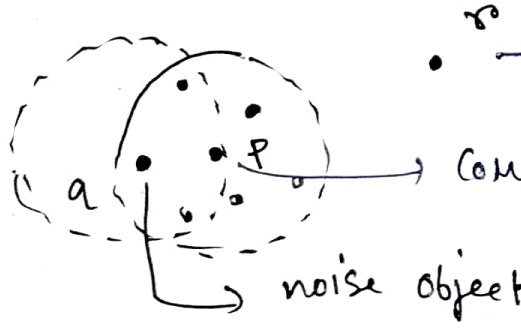
(i) Core object :- An object in an object- database is defined to be core object- if there exist at least MinPts (ϵ) number of points within its ϵ- neighborhood.

where, MinPts and ϵ are the parameters (taken as input) in DBSCAN clustering algorithm.

(ii) Border object :- An object in an object- database is defined to be border object if there are less than MinPts number of ~~others~~ objects within its ϵ-neighborhood.

(iii) Noise object :→ In DBSCAN clustering an object- is defined as noise object- if neither it is a core object- nor a border object-.

the core object-, border object- and Noise objects are shown below: (w.r.t MinPts and ϵ).



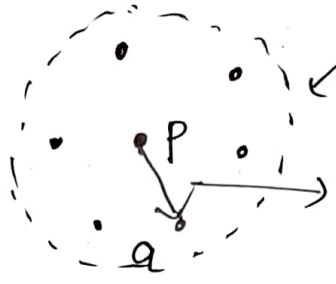
MinPts = 5.

(ii) Density reachability :—

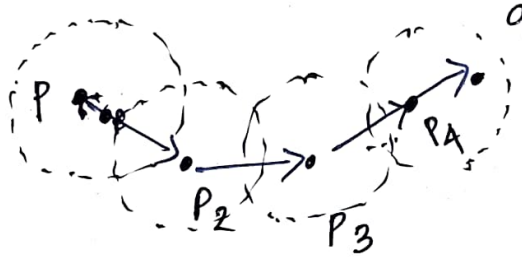
In DBSCAN clustering Density reachability is defined ~~as how~~ in terms of reaching from an object (p) to another object (q) directly or via some other paths (indirectly).

It can be divided in two parts ——

(i) Directly density reachable :— A point q is said to be directly density reachable from another point p if p is a core object and q is within the ε-neighborhood of p.



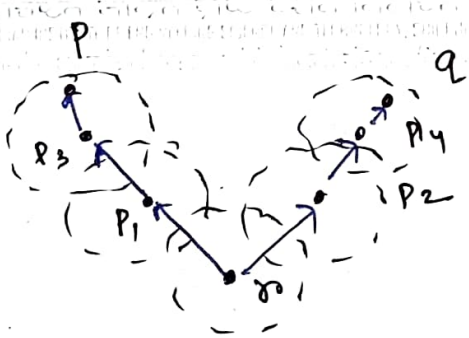
(ii) Density reachable :- A point q is said to be density reachable from another point p, w.r.t ε and minPts if there exist a chain of points $p_1, p_2, \ldots p_n$ such that $p_1 = p$ and $p_n = q$ and p_{i+1} is directly density reachable from p_i.



q is density reachable from p.

4.2

☒ Density connectivity :- An object q is density connected to another object P_1 w.r.t ε and minpts if there exist another object r such that both p and q are density reachable from r.



p and q are density connected to r.

b. Maximality condition :- The maximality condition in DBSCAN algorithm states that if there exist an object p which belongs to some cluster C, that is $p \in C$, then all other points that are density reachable from p also belongs to C, that is if q is in ε-neighborhood of p, then $q \in C$.

(ii) Connectivity condition :- If there exist a cluster C, such that two object- p and q is in C, that is, $p, q \in C$ then p and q are density connected. (3)

(c) Parameters of DBSCAN algorithm :-

(a) → In order to determine the parameters of the DBSCAN algorithm, we will use the concept of K^{th} nearest neigenber. It is assumed that the K^{th} nearest neighbor of every points within a cluster are at a same distance. let's call it K-dist. (4)

If so, K-dist : K^{th} nearest neighbor distance of any points.

(i) First, ~~Calculate~~ select a K value.

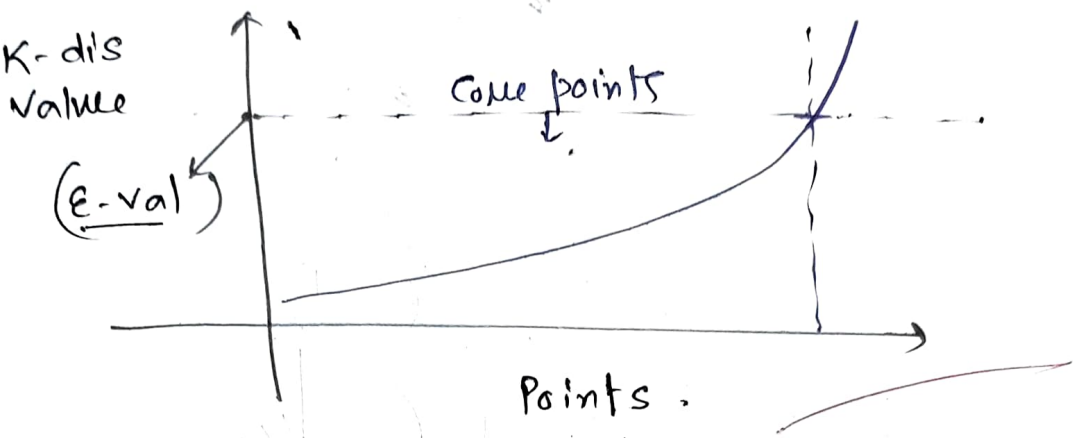
(ii) Find the K^{th} nearest neighbor distance of all points.

(iii) Sort these points based on the K^{th} nearest- neighbor distance.

(iv) Plot these distances.

(v) Carefully, examine the point where K-dist- value drastically changes.

(vi) The K-dist value of that point will be considered as the Eps (ϵ) value and the initial K value will be considered as MinPts.



(2)

(i) **Purity :—** The purity of a cluster is defined as the ratio of the number of points in dominant class to the total number of points.

Mathematically the purity can be defined,

$$\text{Purity}(C_i) = \frac{1}{n_i} \max(n_{ji})$$

↓

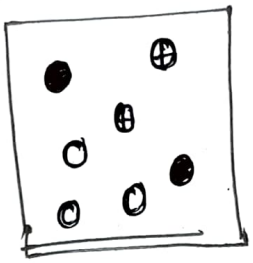
Purity of the ith cluster.

where, n_i → total number of point in the cluster C_i

n_{ji} → number of points that belongs to jth class in the C_ith cluster.

For example :-

Consider the following cluster,



$$\text{Purity}(C) = \frac{1}{7} \max(2,2,3)$$

$$= \frac{3}{7}.$$

(ii) **Rand Index :-** The Rand Index is defined as the ratio of the number of points in the correct cluster with the total no of points in the cluster.

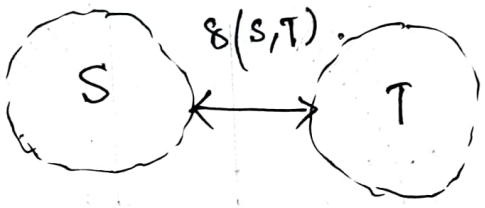
It is denoted by the following matrix,

Number of Points	Points that belongs to same cluster	Points that belongs to different cluster.
Points that show same class labels	A	C
Points that show diff class labels	B	D.

$$\therefore \text{Rand Index} = \frac{A+D}{A+B+C+D}.$$

(b.) Interclusters distance :- The interclusters distance is the distance between any two objects belongs to two different clusters. High value of the distance means better clusters.
As shown below,

Suppose, there exist two clusters S and T the distance between these two clusters are called interclusters distance denoted by $\delta(S,T)$.



→ Intracluster distance :- The intracluster distance is the distance of the cluster itself. It denotes the compactness of a cluster, the less value of the distance indicates more compactness hence good quality cluster.

$\Delta(S)$

→ In both the cases, the distance that are typically used are given below —

(i) Euclidean distance
(ii) Chebyshev distance
(iii) Manhattan distance etc.

Two intercluster distances :—

(i) **single linkage distance** :— The single linkage distance is the minimum distance between any two objects belonging two different clusters.

The equation is as follows,

$$\text{Single linkage distance, } (\delta_1(S,T)) = \min \left\{ \begin{matrix} d(x,y) \\ x \in S \\ y \in T \end{matrix} \right\}.$$

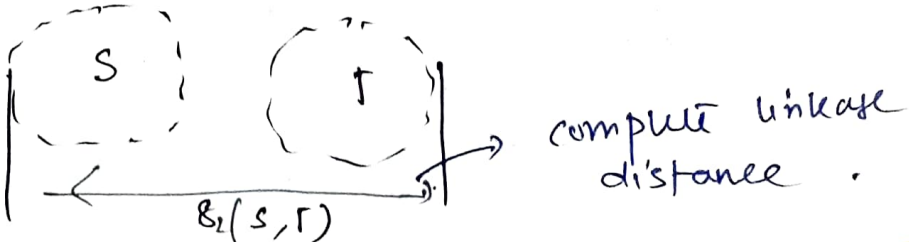
where, S, T → two different clusters.

$d(x,y)$ → distance from x to y.

$x \in S$

$y \in T$.

(ii) **Compute linkage distance** :— The complete linkage distance is the maximum distance of any two objects belonging two different clusters.



The equation is as follows,

$$\delta_2(S,T) = \max \left\{ \begin{matrix} d(x,y) \\ x \in S \\ y \in T \end{matrix} \right\}$$

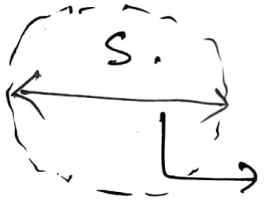
↙ distance between x and y objects.

↓ compute linkage distance.

☒ Two intracluster distances :—

(i) Complete diameter distance :— The complete diameter distance is defined as the maximum distance of any two objects belonging to the same cluster.

$$\Delta_1(S) = \max \left\{ \begin{matrix} d(x,y) \\ x,y \in S \end{matrix} \right\}.$$



$\Delta(S)$
(Compute diameter distance).

(ii) Average diameter distance :— The average diameter distance is the average distance of all the data points belonging to the same cluster.

It is denoted by,

$$\Delta_2(S) = \frac{1}{|S| \cdot (|S|-1)} \cdot \sum_{x,y \in S} d(x,y)$$

(c) Dunn's cluster validation index :→

The Dunn's cluster validation index is define by the following equation,

$$DIndex(U) = \min_{1 \le i \le c} \left\{ \min_{\substack{1 \le j \le c, \\ j \ne i}} \left\{ \frac{\delta(x_i, x_j)}{\max_{1 \le K \le c} (\Delta(x_K))} \right\} \right\}$$

where,

U → ~~any data points~~ (set of cluster)

x_i, x_j → different cluster, (i^{th} cluster, j^{th} cluster)

c → number of clusters.

$\delta(x_i, x_j)$ → inter cluster distance between x_i and x_j.

$\Delta(x_K)$ → intra cluster distance of cluster x_K.

33

☒ Usefulness of Dunn's Index in cluster evaluation :-

(i) Dunn's validation index can be used to evaluate the cluster quality.

(ii) We know that, the condition for a good cluster system is —

(a) inter cluster distance should be more.

(b) intra cluster distance should be less.

Now, In the equation of Dunn's validation index, in the numerator we have inter cluster distance which is supposed to be more where as in the denominator we have intracluster distance which is supposed to be less.

Hence, Higher value of Dunn's index signify better quality cluster.

(4)

(a.) Let, D = following labelled database.

Here, class variable = Play Tennis.

→ Number of tuples having value "yes" = 6.
→ Number of tuples having value "no" = 4.
→ Total number of tuples = 10.

$$\therefore \text{Info}(D) = -\frac{6}{10}\log_2\left(\frac{6}{10}\right) - \frac{4}{10}\log_2\left(\frac{4}{10}\right)$$

$$= -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)$$

$$= -\frac{3}{5}\left[\log_2 3 - \log_2 5\right] - \frac{2}{5}\left[\log_2 2^2 - \log_2 5\right]$$

$$= -\frac{3}{5}\left[1.6 - 2.3\right] - \frac{2}{5}\left[1 - 2.3\right]$$

$$= 0.94$$

→ ☒ Calculating Gain of Outlook :-

$$\text{Gain}(\text{outlook}) = \text{info}(D) - \text{info}_{\text{outlook}}(D) \quad \text{——(i)}$$

Now, $\text{info}_{\text{outlook}}(D)$ so

outlook	Count	Yes	No.
Sunny	4	2	2
Rain	4	2	2
overcast.	2	2	0

$$\therefore \text{info}_{\text{outlook}}(D) = \frac{4}{10}\text{info}_{\text{outlook}}(D_{\text{sunny}}) + \frac{4}{10}\text{info}_{\text{outlook}}(D_{\text{Rain}}) + \frac{2}{10}\text{info}_{\text{outlook}}(D_{\text{overcast}}).$$

$$= \frac{4}{10}\left[-\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right] + \frac{4}{10}\left[-\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right] + \frac{2}{10}\left[-\frac{2}{2}\log_2\left(\frac{2}{2}\right) - \frac{0}{2}\log_2\left(\frac{0}{2}\right)\right].$$

$$= \frac{4}{10}\left[\cancel{0.92} - 2.\frac{1}{2}\left(\log_2 1 - \log_2 2\right)\right] \times 2 + \frac{2}{10}[0]$$

$$= \frac{4}{10}[+1] \times 2$$

$$= \cancel{\frac{8}{10}}\ \frac{8}{10}$$

$$= \cancel{0.8}\ 0.8$$

∴ from (i),

Gain (outlook) = 0.94 − 0.~~8~~8

= ~~0.94~~ 0.14

→ Calculating Gain of Humidity :—

$$\text{Gain}(\text{Humidity}) = \text{info}(D) - \text{info}_{\text{Humidity}}(D).$$

Humidity	count	yes	no
Normal	5	4	1
High	5	2	3

∴ $\text{info}_{\text{Humidity}}(D) = \frac{5}{10}\left[\text{info}_{\text{Humidity}}(D_{\text{Normal}}) + \frac{5}{10}\,\text{info}_{\text{Humidity}}(D_{\text{High}})\right.$

$$= \frac{5}{10}\left[-\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right)\right] +$$

$$\frac{5}{10}\left[-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right]$$

$$= \frac{5}{10}\left[-\frac{1}{5}(\log_2 4 - \log_2 5) - \frac{1}{5}(\log_2 1 - \log_2 5)\right] +$$

$$\frac{5}{10}\left[-\frac{2}{5}(\log_2 2 - \log_2 5) - \frac{3}{5}(\log 2^3 - \log_2$$

$$= \frac{5}{10}\left[-\frac{4}{5} \times (2 - 2.3) - \frac{1}{5}(0 - 2.3)\right] +$$

$$\frac{5}{10}\left[-\frac{2}{5}(1 - 2.3) - \frac{3}{5}(1.6 - 2.3)\right]$$

$$= 0.35 + 0.47$$

$$= 0.82$$

$\therefore$ Gain (Humidity) $= 0.94 - 0.82$

$= 0.12$

→ Calculating Gain For Wind :—

Gain (wind) = info (D) − info$_{\text{wind}}$ (D).

wind	count	yes	no.
strong	5	2	3
weak.	5	4	1

$$\therefore \text{info}_{\text{wind}}(D) = \frac{5}{10}\left[-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right] +$$

$$\frac{5}{10}\left[-\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right)\right]$$

$$= \frac{5}{10}\left[-\frac{2}{5}\left(\log_2 2 - \log_2 5\right) - \frac{3}{5}\left(\log_2 3 - \log_2 5\right)\right]$$

$$+ \frac{5}{10}\left[-\frac{4}{5}\left(\log_2 4 - \log_2 5\right) - \frac{1}{5}\left(\log_2 1 - \log_2 5\right)\right]$$

$$= \frac{5}{10}\left[-\frac{2}{5}(1 - 2.3) - \frac{3}{5}(1.6 - 2.3)\right] +$$

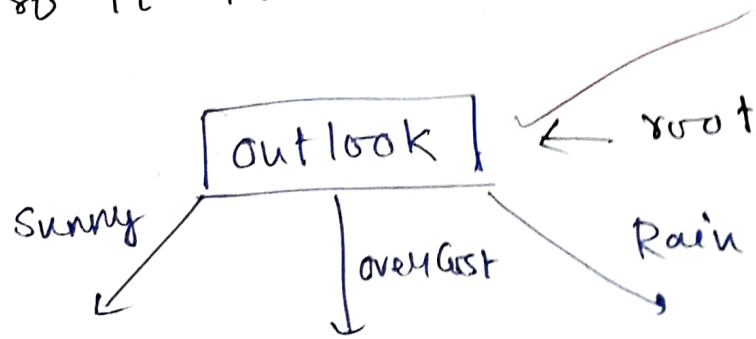
$$\frac{5}{10}\left[-\frac{4}{5}(2 - 2.3) - \frac{1}{5}(0 - 2.3)\right]$$

$$= 0.47 + 0.35$$

$$= 0.82$$

$$\therefore \text{Gain}(\text{wind}) = 0.94 - 0.82 = 0.12.$$

→ Hence, the maximum Gain, we get for the attribute outlook, so it is the root node of the decision tree.



(b.) Terminating criteria of a decision tree :-

There are two possible terminating criteria of a decision tree —

(i) If the attribute list is empty then we assign the class labels to the corresponding node by using majority voting.

(ii) If all the tuples of a given attribute belongs to the same class labels, then we make it a leaf node by assign the corresponding class label to it.

(c) Causes of model overfitting :-

The term 'overfitting' in case of Data mining is defined as below, if a model performs well on the training dataset but perform poorly on a test (unknown) dataset then the model is said to be overfitted, Here are some of the possible reason of model overfitting —

(i) Anomalies or noise in the dataset :- If there exist anomalies or noisy data within the training dataset the model will be overfitted.

(ii) Inconsistent data :- If there exist an Inconsistent tuple in the training dataset itself, it causes overfitting. For inconsistent data, the leaf of the decision tree assigns probability to each class labels.

(iii) Not generalized training sample :- If training sample are not generalized well, it can causes overfitting.

Solution of overfitting in Decision Tree :-

There are two method to solve the over-fitting problem —

(i) Pre pruning :- In this method, during the decision tree construction itself, we remove some of the branches of the decision tree.
→ The removal of branch depends upon the initial threshold value.

→ Each splitting criteria measures the goodness score which is below threshold, causes the decision tree to be pruned.

→ and the pruned attribute is set by the majority class in that subtree.

(ii) Post Pruning :— In this method, we prune the tree after construction is done.

4

→ The ~~nodes would be pruned~~ branch that would be pruned, will be replaced by the majority class present in the pruned subtree of the decision tree.

(5) (a) Applying Naive Bayes classification algorithm :—

Total number of tuples = 10

~~Total number of~~

Number of classes = 2.

Let, c_1 = when (PlayTennis = "yes").

c_2 = when (playTennis = "no").

Number of tuples with, $c_1 = 6$.

Number of tuples with $c_2 = 4$.

$\therefore \quad P(C_1) = \frac{6}{10} = 0.6$

$P(C_2) = \frac{4}{10} = 0.4$.

Now,

let, $X = \langle$ outlook = Overcast, Humidity = High, Wind = Weak $\rangle$.

⇒ To predict the class label of X, we need to use the, maximum posteriory,

$$P(C/X) = \frac{P(X/C) \cdot P(C)}{P(X)}.$$

As, we need to maximize $P(C/X)$ we only consider the numerator as denominator is constant over all class labels.

→ For, $C = C_1$ (PlayTennis = "yes").

Maximum posteriory,

$P(X/C_1) \cdot P(C_1)$ —(i).

Now, let, $P(X/C_1) = P(\text{outlook} = \text{Overcast} / \text{playTennis} = \text{'yes'})$

$= 2/6 = 0.34$

Let $P(x_2/c_1) = P(\text{Humidity} = \text{"High"} / \text{playTennis} = \text{"yes"})$.

$= 2/6 = 0.34$

Let, $P(x_3/c_1) = P(\text{Wind} = \text{"Weak"} / \text{playTennis} = \text{"yes"})$

$= \frac{4}{6}$

$= 0.67$.

$\therefore P(X/c_1) = P(x_1/c_1) \cdot P(x_2/c_2) \cdot P(x_3/c_3)$

$= 0.34 \times 0.34 \times 0.67$

$= 0.077$

from (i).

$\therefore P(X/c_1) \cdot P(c_1) = 0.077 \times 0.6$

$= 0.046$ —— (2).

That, is $P(X/\text{playTennis} = \text{"yes"}) = 0.046$. —— (2).

Now, For $c = c_2$ [playTennis = "NO"],

let,

$P(x_1/c_2) = P(\text{outlook} = \text{"over cast"} / \text{playTennis} = \text{"no"})$

$= 0$.

As, one term is zero,

hence,

$$P(X_1/C_2) = 0.$$

and, $P(X/C_2) \cdot P(C_2)$ is also zero.

So, The Naive Bayes' classification algorithm will predict the class label as playTennis = "yes".

(b.) In the above, Naive Bayes' classifier algorithm when we tried to calculate the maximum value of $P(X/\text{playTennis} = \text{"no"})$, we got one probability value as zero, hence it causes the whole value to be equal to zero. This is an error. [Zero probability error].

→ The error can be corrected using; Laplacian estimator which states that if there is a zero probability problem, adding 1 to all the occurances of the discrete values of that attributes causes the probability to get some value closer to the actual value.

→ Applying this correction to the above Naïve Bayes' classification,

→ For, $c = c_2$ [playTennis = "no"],

let, $P(x_1/c_2) = P(\text{outlook} = \text{"overcast"} / \text{playTennis} = \text{"no"})$

$= \frac{0+1}{4}$

$= 0.25.$

(4) let, $P(x_2/c_2) = P(\text{humidity} = \text{"high"} / \text{playTennis} = \text{"no"})$

$= \frac{3+1}{4}$

$= 1$

let, $P(x_3/c_2) = P(\text{wind} = \text{"weak"} / \text{playTennis} = \text{"no"})$

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

$= 0.5.$

$\therefore P(x/c_2) = 0.25 \times 1 \times 0.5 = 0.125$

$\therefore P(x/c_2) \cdot P(c_2) = 0.125 \times 0.4 = 0.05.$

→ Hence after correction, the actual class label for x is, (playTennis = "no").

(c.) For any feature, if it is contineous valued attribute,

Then calculation of the likelihood is done by using the Gaussian's distribution as shown below,

$$P(\underset{(x_K)}{\text{Humidity value}} / c) = g(\underset{(x_K)}{\text{Humidity val}}, \mu_c, \sigma_c)$$

3

where, μ_c = mean value of the class label.

σ_c^2 = variance of the class label.

Now, $$g(\underset{(x_K)}{\text{Humidity}}, \mu_c, \sigma_c) = \frac{1}{\sqrt{2\pi\sigma_{c_K}^2}} \cdot e^{-\frac{1}{2}\left(\frac{x_K - \mu_{c_K}}{\sigma_{c_K}}\right)^2}$$

In the above algorithm, we need to replace, the given attribute value probability using the above procedure.

P.T.O

Extra

(3.) support :- The support of an association rule $(X \to Y)$ is defined as the ratio of the number of transaction the $(X \cup Y)$ item set occurs with the total number of transactions,

$$\text{Support}(X \to Y) = \frac{(X \cup Y).\text{Count}}{n}$$

Confidence :- The confidence of an association rule $(X \to Y)$ is defined as $P(Y/X)$.

$$\text{Confidence}(X \to Y) = P(Y/X) = \frac{P(X \cup Y)}{P(X)}$$

→ An itemset is referred to as the frequent itemset if, its support has at least equal of the minimum support.

→ An association rule is referred to as the important rule, if its support ≥ minsup and confidence ≥ minconf.

☑ given, minsup = 30/100 = 0.3
minConf = 80/100 = 0.8

→ Using Apriori Algorithm

let, Bread = I_1 , Jelly = I_4
Butter = I_2 , Coke = I_5
Milk = I_3 ,

The transaction database is,

Transactions	items
T_1	I_1, I_2
T_2	I_1, I_3, I_2
T_3	I_1, I_4, I_2
T_4	I_1, I_5
T_5	I_1, I_3
T_6	I_3, I_5.

C_1:

items	support count
I_1	$5/6 = 0.84$ ✓
I_2	$3/6 = 0.5$ ✓
I_3	$3/6 = 0.5$ ✓
I_4	$1/6 = 0.17$ ×
I_5	$2/6 = 0.34$ ✓

frequent itemset →

f_1:

items	support
I_1	0.84
I_2	0.5
I_3	0.5
I_5	0.34

C_2:

items	support
(I_1, I_2)	$3/6 = 0.5$ ✓
(I_1, I_3)	$2/6 = 0.34$ ✓
(I_1, I_5)	$1/6 = 0.17$ ×
(I_2, I_3)	$1/6 = 0.17$ ×
(I_2, I_5)	$0/6 = 0$ ×
(I_3, I_5)	$1/6 = 0.17$ ×

frequent itemset →

f_2:

items	support
(I_1, I_2)	0.5
(I_1, I_3)	0.34.

C_3 : NULL $\quad f_3$ = NULL

hence, frequent itemset, $f_2 = \{(I_1, I_2), (I_1, I_3)\}$.

→ Association rules :—

For, I_1, I_2

rules	confidence
$I_1 \rightarrow I_2$	support $(I_1 \cup I_2)$ / support (I_1) = 0.5/0.84 = 0.59 ×
$I_2 \rightarrow I_1$	support $(I_1 \cup I_2)$ / support (I_2) = 0.5/0.5 = 1 ✓

For, I_1, I_3,

Rule	Confidence
$I_1 \rightarrow I_3$	0.34/0.84 = 0.40 ×
$I_3 \rightarrow I_1$	0.34/0.5 = 0.68 ×

So, important association rules, $I_2 \rightarrow I_1$.

☑ Drawbacks of a-priori algorithm :—

(i) The time complexity is exponential.
(ii) Space requirement is huge.
(iii) Not efficient enough.