DATA MINING

B.Tech (CSE)

Notes

Prepared By: TOPPERWORLD LEARN & GROW

UNIT-I

INTRODUCTION TO DATA MINING

1. Introduction
2. What is data Mining
3. Definition
4. KDD
5. challenges
6. Data Mining tasks (Pending)
7. Data Preprocessing
8. Data cleaning
9. Missing Data
10. Dimensionality Reduction
11. Feature Subset selection
12. Discretization and Binarization
13. Data Transformation
14. Measures of Similarity and Dissimilarity - Basics.

Introduction :-

Data Mining is defined as the procedure of extracting information from huge sets of data. In other words we can say that data mining is mining knowledge from data.

→ There is a huge amount of data available in the information industry. This data is of no use until it is converted into useful information. It is necessary to analyze this huge amount of data and extract useful information from it.

→ Extraction of information is not only the process we need to perform: Data mining also involves other processes such as data cleaning, Data integration, Data transformation, Pattern Evaluation and Data presentation. Once all these processes are over, we would be able to use this information in many applications such as

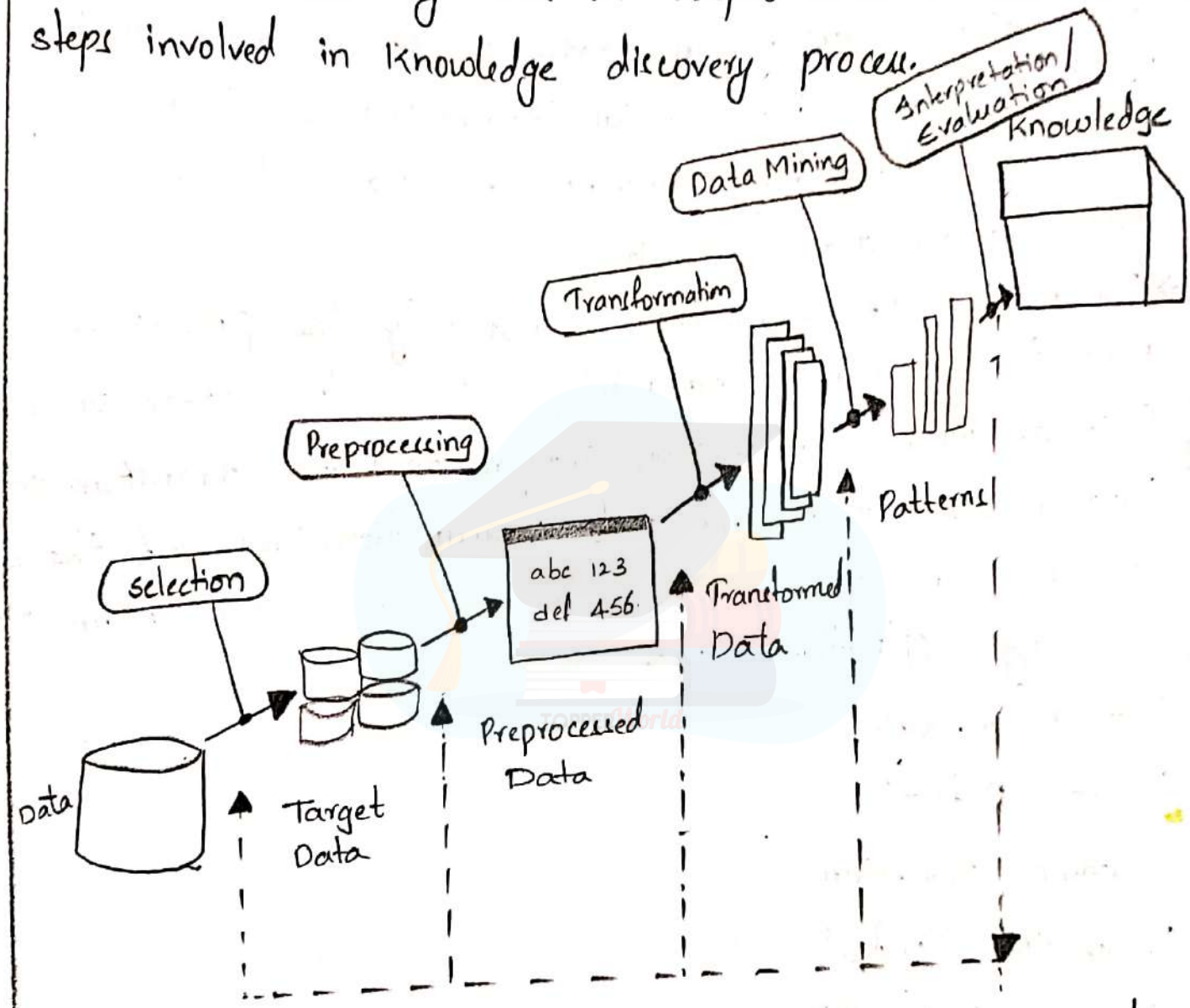
Market Analysis (M)
Fraud Detection (F)
Customer Retention (C)
Production Control (P)
Science Exploration. (S)

* WHAT IS DATA MINING :-

Data Mining is defined as extracting information from huge sets of data. The information or knowledge extracted can be used for many applications.

* **KDD:-**

KDD stands for Knowledge Discovery in Database. Data mining is an essential step in the process of knowledge discovery. There are seven different stages in KDD process. This process takes raw data as input and provides useful information desired by users as output. Here is the list of steps involved in knowledge discovery process.



- Preprocessing of database consists of Data cleaning and Data integration.
- KDD is an iterative process.

DATA MINING CHALLENGES:-

The challenges are

1. Complex Heterogeneous data.
2. Distributed data.
3. Scalability
4. Non-Traditional Analysis
5. High Dimensionality

1. Complex Heterogeneous data:-

The growth in various fields such as Science, medical and finance produced large complex heterogeneous and non-traditional data. Some of such data includes semi-structured text, unstructured text and multimedia. This type of data cannot be handled by classical data analysis techniques.

2. Distributed data:-

Data needed for analysis in certain circumstances does not belong to single owner or stored in single geographic location. This distributed data analysis needs new techniques.

(i) Techniques to minimize resources needed for distributed computing.

(ii) Integration of data mining results from heterogeneous sources.

(iii) Handling Data Security.

3. Scalability:-

Data mining algorithms must be capable to handle and incorporate huge volumes of data. These algorithms can be made more scalable by

(i) Sampling data

(ii) Implementing data structure

(iii) Developing distributed and parallel algorithms.

4. Non-Traditional Analysis:-

This analysis methods are based on hypothesis and testing. This method needs high volumes of resources which becomes difficult in present data mining scenario.

5. High Dimensionality:-

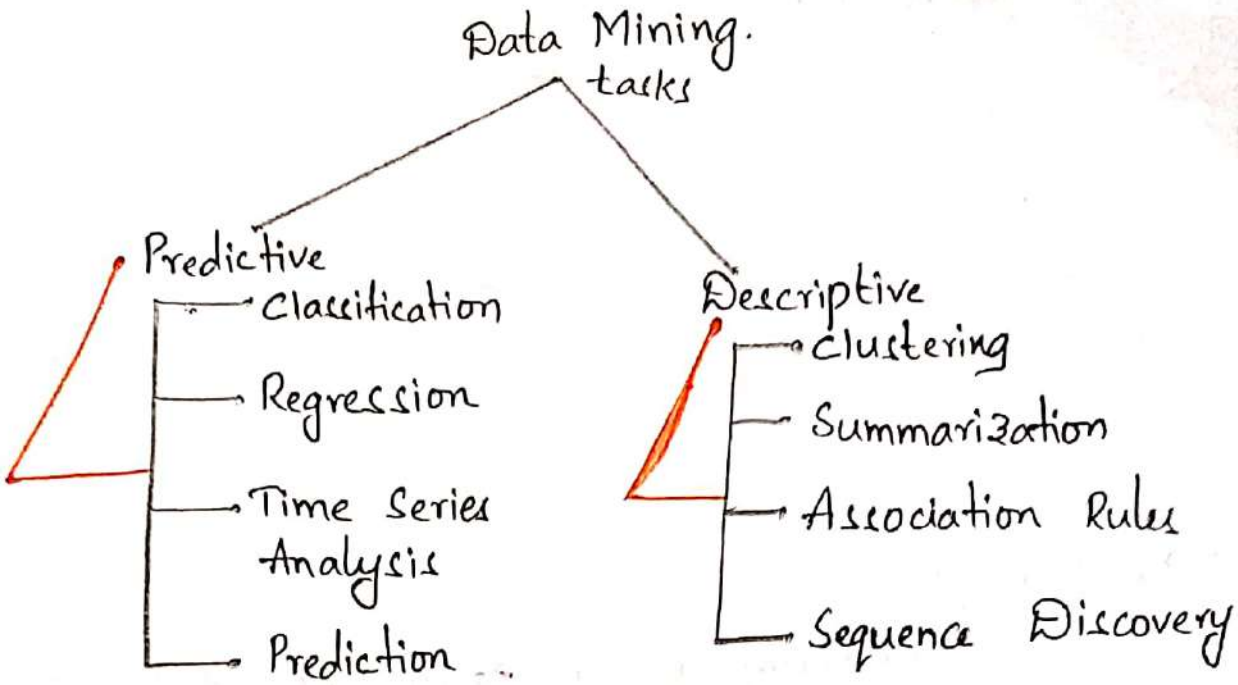
Data sets of present era has several hundreds of attributes. There are number of dimensions that grow with technology advancement.

* DATA MINING TASKS:-

The data mining tasks can be classified generally into two types. Those two categories are

① Predictive Tasks

② Descriptive tasks.



Predictive:- It makes prediction about values of data using known results from different data or based on historical data.

Descriptive:- It identifies patterns or relationship in data, it serves as a way to explore properties of data.

→ Classification:- discovery of a function that classifies a data item into one of several predefined classes.

Given a collection of records.

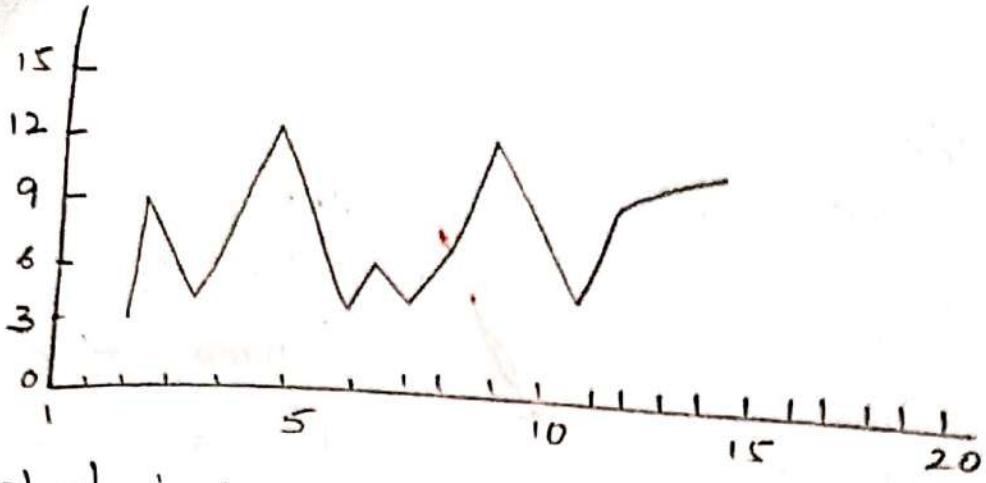
Each record contains a set of attributes, one of the attributes is class.

Ex:- Pattern Recognition.

→ Time series analysis:-

- The value of attribute is examined as it varies over time.
- A time series plot is used to visualize time series.

Ex:- Stock Exchange.



→ Clustering:-

Clustering is a task of segmenting a diverse group into a number of similar subgroups or clusters. Most similar data are grouped in clusters

Ex:- Bank Customer.

DATA PREPROCESSING:-

Data Preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real world data is often incomplete, inconsistent and is likely to contain many errors. Data Preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

* Data preprocessing is used database driven application such as customer relationship management and rule-based applications.

* Data preprocessing is required because

⟹ Real world data are generally

- Incomplete:- Missing attribute values.
- Noisy:- Containing errors or outliers.
- Inconsistent:- Containing discrepencies in codes or names

Need for preprocessing the data:-

1) Attributes of interest may not be available always.
2) Relevant data may not be recorded due to misunderstanding or because of equipment malfunctions.
3) Data that is inconsistent with other recorded data might be deleted.
4) Missing data, particularly for tuples with missing values for some attributes, may need to be inferred.
5) The data collection instruments used may be faulty

6) There may be human or computer errors occuring at entry.

7) Errors in data transmissions can also occur.

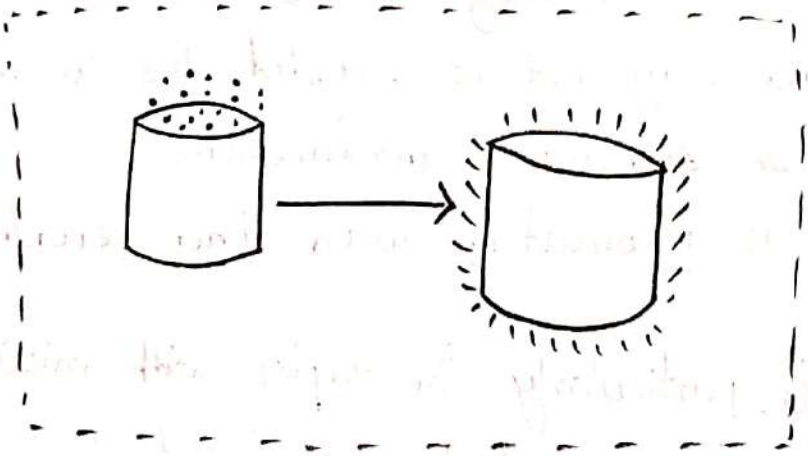
8) There may be technical limitations such as limited buffer size for coordinality synchronized data transfer and consumption.

* To overcome the above problems the following data preprocessing techniques are required.

1. Data cleaning
2. Data Integration
3. Data Transformation
4. Data Reduction.

1. Data cleaning:-

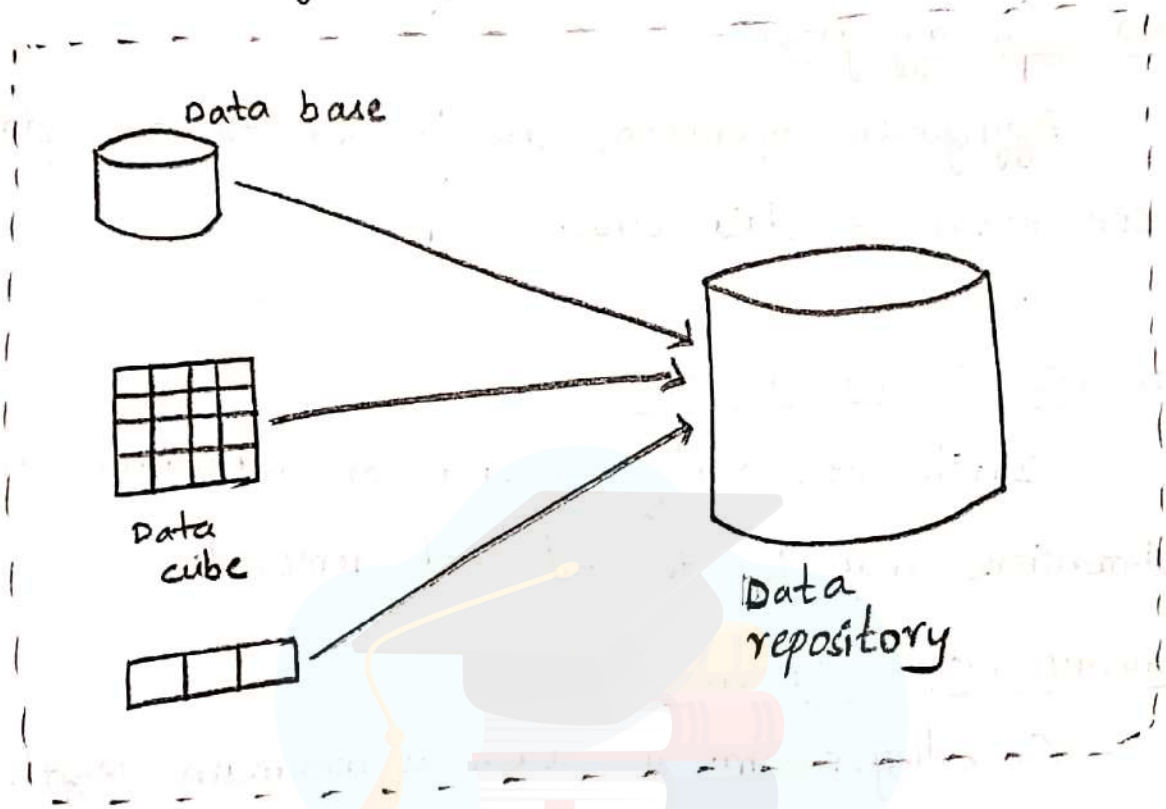
This routine work is to 'clean' the data by filling in missing values, smoothing noisy data, identifying or removing outliers and resolving inconsistencies.



Data cleaning.

Data Integration:-

This is the process of integrating multiple databases, data cubes, or files. Yet some attributes representing a given concept may have different names in different databases, causing inconsistencies and redundancies.



3) Data Transformation:-

This is a kind of operation in which we use normalization and aggregation.

- $-4, 29, 100, 40, 80 \rightarrow -0.04, 0.29, 1.00, 0.40, 0.80$
- $1, 20, 50, 100 \rightarrow 0.01, 0.20, 0.50, 1.0$

4) Data reduction :-

This is the reduced representation of the data set that is much smaller in volume, yet produces the same analytical results.

* Following are different data reduction strategies:

(a) Data cube aggregation:-

Aggregation operations are applied to the data in the construction of data cube.

→ sum, count, add, Min, Max

(b) Attribute subset selection:-

Irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed.

(c) Dimensionality reduction:-

Encoding mechanisms such as minimum length encoding or wavelets are used to reduce the data set size.

(d) Numerosity reduction:-

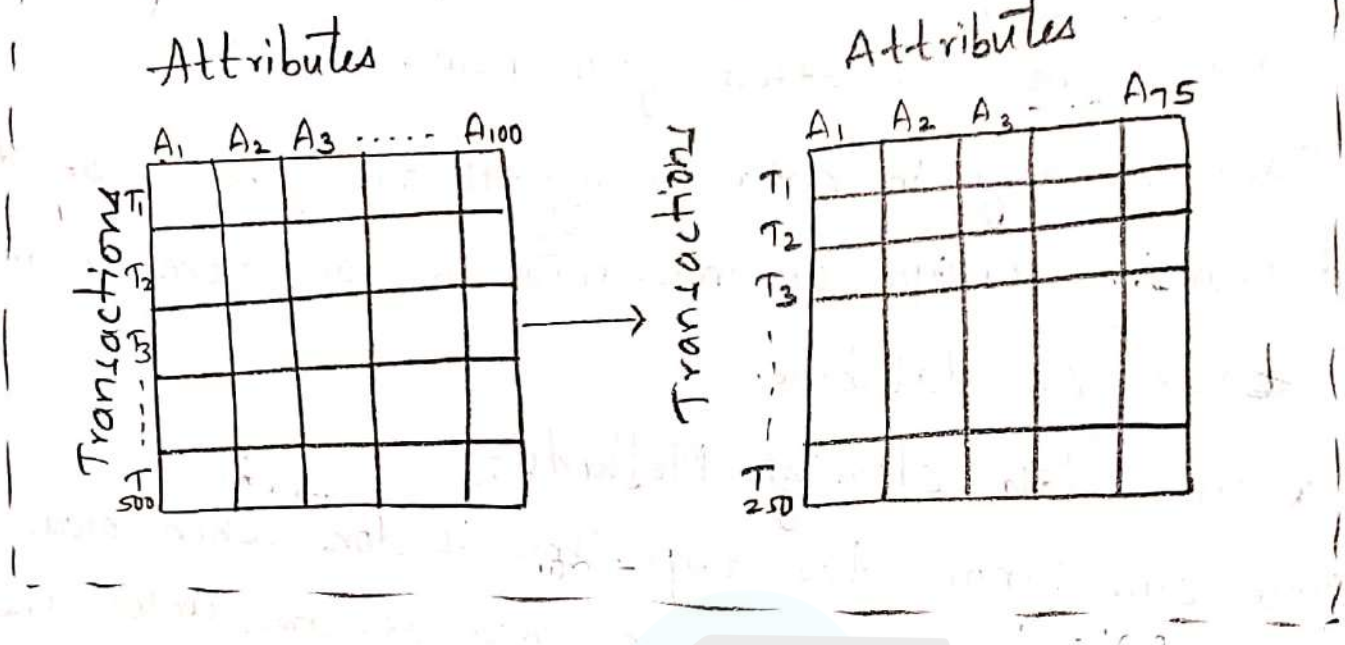
The data is replaced or estimated by alternative, smaller data representations such as clusters or parameters models, histograms, Sampling.

(e) Data discretization and concept hierarchy generation:-

Generalization:-

Raw data values for attributes are replaced by ranges or higher-level concepts. For Example, raw values for age may be replaced by higher-level concepts, such as youth

adult or senior. Automatic generation of concept hierarchies from numerical data.



Data Preprocessing

- Data cleaning
- Integration
- Data selection
- Data Transformation

DATA CLEANING :-

Quality of your data is critical in getting to final analysis. Any data which tend to be incomplete, noisy and inconsistent can effect your result.

* Data cleaning in data mining is the process of detecting and removing corrupt or inaccurate records from a record set, table or database.

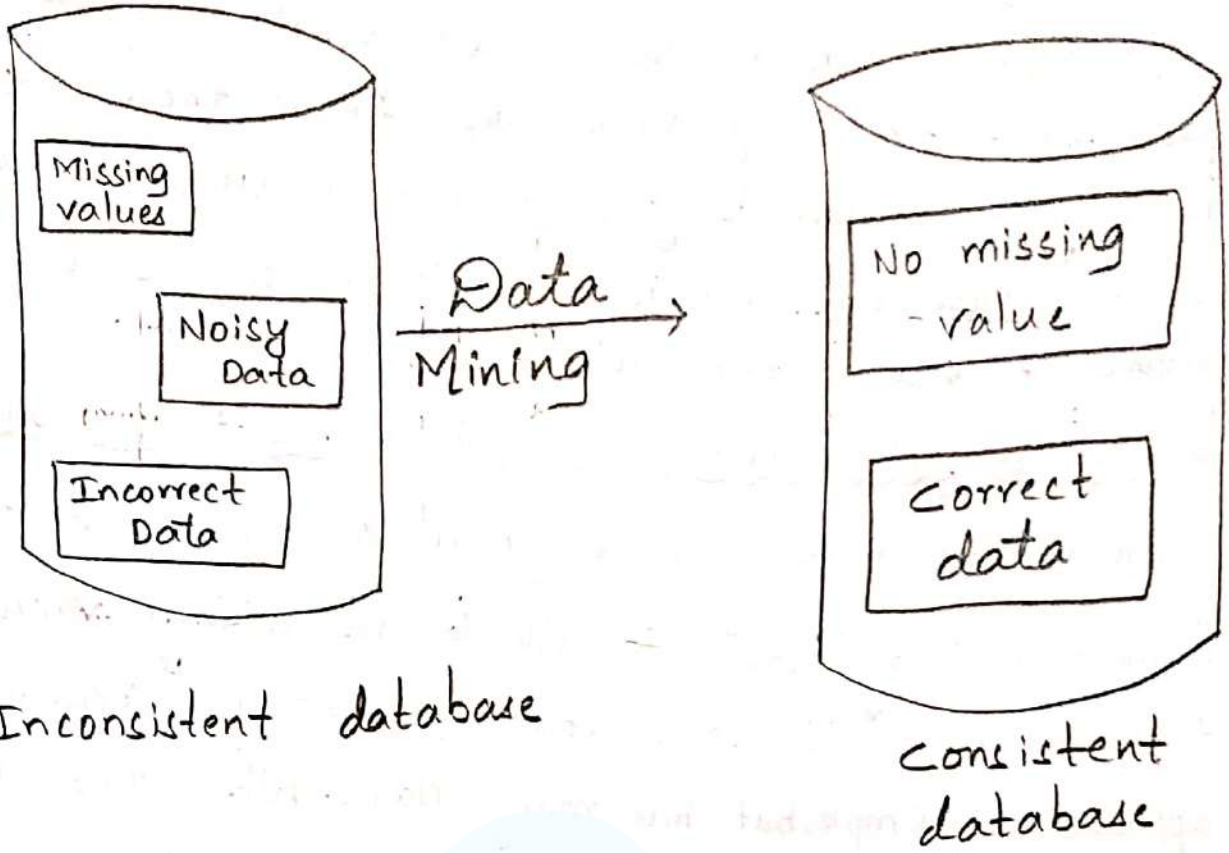
* Some data cleaning Methods :-

1) You can ignore the tuple. This is done when class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values.

2) You can fill in the missing value manually. This approach is effective on small data set with some missing values.

3) You can replace all missing attribute values with global constant, such as a label like 'unknown' or minus infinity.

4) You can use the attribute mean to fill in the missing value.



* MISSING DATA:-

In data warehouses data is stored in relational format i.e in the form of rows and columns. It any of the attribute value is mistakenly not recorded for any tuple, then it will lead to inaccurate results. In order to handle these missing values, the following methods are employed by data cleaning process.

① Ignoring the tuple with missing data values:-

This approach is beneficial when more number of attribute values are missing within the same tuple. when the number of missing values varies significally.

2) Manually filling the missing data values:-

In this method, the user himself should try to find out the tuples with missing values and fill in those tuples manually. This method is generally not advantageous as it consumes more time and is not suitable to use when massive volume of data contains missing values.

3) Using a global constant to fill in the missing value:-

This method makes use of global constant such as "unknown" or "∞" to fill in the missing value. All tuples with missing value contain identical constant. This approach is simple, but has many flaws. When data analysis is being performed these values are unintentionally treated as special value by the data mining process → inaccurate result.

4) Using attribute average value:-

For Example in student database the missing value in marks attribute can be filled by calculating the average marks of all the students i.e all missing value tuples will contain identical mean value.

5) Use the most probable value to fill in the missing value:-

Eg:- Using the other customer attributes in your data set you may construct a decision tree to predict the missing values for income.

* Missing value may not always result in an error.

DIMENSIONALITY REDUCTION:-

Dimensionality Reduction is the process of reducing the number of random variables or attributes under consideration by using different encoding schemas.

* Dimensionality Reduction methods include

a) Wavelet transforms

b) Principal Component analysis. (PCA)

a) Wavelet transforms:-

The discrete wavelet transform (DWT) is a linear signal processing technique. It transforms a vector into a numerically different vector. (D to D') of wavelet coefficients

→When applying this technique we consider each tuple as an n-dimensional data vector depicting 'n' measurements made on the tuple from 'n' different database attributes.

→Wavelet transform data can be truncated

→A small compressed approximation of the data can be retained by storing only a small ~~function~~ fraction of the strongest wavelength co-efficient.

→Removes noisy without smoothing out the main feature of data making effective for data cleaning.

→Wavelet transforms can be applied to multidimensional data such as data cubes. wavelet transforms have many real world applications including the compression of finger print images, computer vision and analysis of time-series data.

Principle Component Analysis:- (PCA)

(Korhonen – leone - KI method)

Data to be reduced consists of tuples or data vectors described by n-attributes or dimensions.

→ PCA searches for k-dimensional orthogonal vectors that can be best used to represent the data where c<=k.

→ The original data are thus projected onto a much smaller space

→ The basic procedure is as follows:-

i) The input data are normalized, so that each attribute falls within the same range. This step helps to ensure that attributes with larger domains will not dominate attributes with smaller domains.

ii) PCA computes orthonormal vectors to provide a basis for the normalized input data. These are unit vectors that point in a direction perpendicular to others. These vectors are referred to as the principal components.

→ Principal components may be used as input to multiple regression and cluster analysis.

→ PCA handles better the sparse data.

→ PCA is computationally inexpensive and it can be ordered or unordered.

FEATURE SUBSET SELECTION :-

Dimensions can be reduced through feature subset selection.

→ A database consists of massive volumes of data set which inturn are the collection of the records. Each record consists of numerous attributes. Out of these attributes set many of the attributes are duplicate, inconsistent and irrelevant. It is very time consuming if data analysis is performed on all these attributes.

→ It is difficult for a data analyst to select the relevant attributes when the characteristics of the data is unknown. The selection of irrelevant attribute can lead to poor quality pattern, confusion and degradation in performance of mining process.

→ In order to eliminate the usage of irrelevant attributes a strategy called "Attribute subset selection" is used. This strategy compresses the actual size of data set by deleting those attributes that are redundant and irrelevant. The advantage of using this strategy is that, the discovered patterns can be easily understood.

Feature subset selection has three approaches.

a) Embedded
b) Filter
c) Wrapper

a) Embedded:-

Data algorithms during their operations decide whether to apply a particular attribute or not. Decision tree classifier algorithm operates on this principle.

b) Filter:-

characteristics of data set are selected prior to applying datamining algorithms through an independent technique.

c) Wrapper:-

Target datamining are applied on black box to search for best subset of attributes. The following steps are followed to select subset features.

(i) Meaures to evaluate a subset:

The current features subsets must be compared against new features generated. This comparision needs an evaluation criteria to know subsets attributes for data mining operations such as clustering or classification. In wrapper method subset evaluates data mining results whereas filter method (applies) attempts to determine the performance of data mining algorithm on a set of attributes.

ii) Strategy to control new subset Generation:

In this method, all possible feature subsets are searched to select features. These can be selected using various search strategies. however the selected strategy should find optimal or near optimal feature set.

iii) Criteria to stop Feature Generation:-

The subsets can be generated in huge volume which cannot be examined individually therefore a stopping criteria is needed. This criteria is based on certain conditions such as

- The number of iterations.
- The value that measures subset whether it is met with the optimal level or not.
- Whether any improvement can be made using search strategy.
- Whether creation size of subset is obtained or not.
- Whethe evaluation criteria and simultaneous size is obtained or not.

iv) Validation Procedure:-

The result produced by data mining algorithm on target subsets need to be validated. This can be done by running the algorithm with entire features and then comparing the results against the result obtained from feature subset.

* DISCRETIZATION AND BINARYZATION - DATA DISCRETIZATION:

Data discretization converts a large number of data values into smaller ones, so that data evaluation and data management becomes very easy.

(or)

Discretization is the process of putting values into buc so that there are a limited no. of possible states. The buckets themselves are treated as ordered and discrete values.

→ There are several methods that you can use to discrete data. If your data mining solution uses relational data, you can control the number of buckets for grouping data by setting the value of the Discretization Bucket Count property. The default no. of buckets is 5.

Example:-

We have an attribute age with following values:

Age	10, 11, 12, 13, 14, 17, 19, 30, 31, 32, 38, 40, 42, 70, 72, 73, 75

Table : Before discretization

Age	10, 11, 13, 14, 17, 19	30, 31, 32, 38, 40, 42
	70, 72, 73, 75	

Table: How to discretization

Young	old	Mature

Age	Young	Mature

Table: After discretization

*There are different methods which are used for performing data discretization.

a) Supervised Discretization:

It data is discretized using class information then it is refered as supervised or organized discretization.

b) Unsupervised Discretization:

It data values are reduced by substituting them by limited interval description but without using class Information then it is refered to as unsupervised discretization

c) Top - Down Discretization:

It the process starts by first finding one or a few points to split the entire attribute range, and then repeats this recursively on the resulting intervals, then it is called top-down discretization or splitting.

d) Bottom-up discretization:

If the process starts by considering all of the continuous values as potential split-points, removes some by merging neighbour-hood values to form intervals then it is called bottom-up discretization or merging.

Techniques of Data Discretization:

1. Histogram analysis
2. Binning
3. correlation analysis

4) Clustering analysis
5) Decision tree analysis
6) Equal width partitioning
7) Equal depth partitioning
8) Entropy based discretization

1) Histogram analysis:-

Histogram analysis does not use class information so it is an unsupervised discretization technique. Histograms partition the values for an attribute into disjoint ranges called buckets.

2) Binning:

Binning is a top-down splitting technique based on a specified number of bins. Binning is an unsupervised discretization technique.

a) Equal-width binning
b) Equal-depth binning

a) Equal-width binning:-

Given a range of values [min, max] we divide in intervals of approxiamately same width; either we set the width arbitarily to w, or we set the desired number of bins to n, in this case w is calculated as.

$$w = max - min / n$$

Eg:-

If the range is [0,100] and we want 4 bins, each bin will have a width of

$$100 - 0/4$$
$$= 25$$

the bins will be [0,24], [25,49], [50,74], [75,100]

b) Equal-depth binning:-

Given a range of values [min,max], we place approximately the same number of instances in each bin by dividing the total number of samples nb by the desired number of samples in each bin(depth) d, in that case number of bins are calculated as:

$$n = nb/d$$

Eg:-

If the range is [0,100] for 100 samples of different values (for eg 99 is missing), we want 20 samples in each bin, the no. of bins will be

$$100/20$$
$$= 5$$

the bins will be [0,19], [20,39], [40,59], [60,79], [80,100]

Advantage:-

*Equal width binning is more simple however very sensitive to outliers in the data.

* Equal-depth binning scales well by keeping the distribution of data however the bin values may be more difficult to interpret.

3) correlation analysis:-

correlation is often used as a preliminary technique to discover relationships between variables. More precisely, the correlation is a measure of the linear relationship between two variables.

4) clustering analysis:-

clustering is the process of making a group of abstract objects into classes of similar objects

→ cluster analysis is a popular data discretization method. A clustering algorithm can be applied to discrete a numerical attribute of A by partitioning the values of A into clusters or groups.

→ Each initial cluster or partition may be further decomposed into several subcultures, forming a lower level of the hierarchy.

5) Decision tree analysis:-

A decision tree is a structure that includes a root node, branches and leaf nodes. Each internal node denotes a test on a attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is root node.

6) Entropy based discretization:-

Here entropy means lack of order or prediction.

→ Uses the concept called information gain.

→ It is supervised top-down splitting. Each value of 'A' can be considered as a potential interval boundary or splitting point to partition the range 'A' i.e a splitting for 'A' can partition the tuple in 'B', into 2 subsets satisfying $A \leq$ splitting point or $A >$ splitting point respectively. Hereby creating a binning discretization.

tuples:

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

Classes:

$$\text{Entropy}(D_1) = \sum_{i=1}^{m} P_i \log_2(P_i);$$

* DATA TRANSFORMATION:

In data transformation process data are transformed from one format to other format, that is more appropriate for data mining.

In normalization we have 3 methods

1) Min-max normalization
2) Z-score normalization
3) Decimal scaling normalization.

1) Min-Max normalization:-

Performs a linear transformation on the original data. Suppose that min A and max A are the minimum and maximum values of an attribute 'A'.

min-max normalization maps a value v of A to v' in the range [new-min A, new-max A]

$$v' = \frac{v - \min A}{\max A - \min A} (new\_max_A - new\_min_A) + new\_min_A$$

Example:-

Q) Suppose the minimum & maximum values for an attribute income are 12,000 and 98,000 respectively. We would like to map income range [0,1] by min-max normalization a value of 73,600 for income transformed.

Sol:- $$v' = \frac{v - \min_A}{\max_A - \min_A} (new\text{-}max_A - new\_min_A) + new\_min_A$$

$$v' = \frac{73,600 - 12000}{98000 - 12000} (1-0) + 0$$

$$= 0.716$$

* MEASURES OF SIMILARITY AND DISSIMILARITY

Distance or Similarity measures are essential to solve many pattern recognition problems such as classification and clustering. various distance/similarity measures are available in literature to compare two data distributions. As the names suggest, a similarity measures how close two distributions are

1. Similarity measure
2. Dissimilarity measure.

Similarity measure:-

The Similarity between the two objects is a numerical measure of the degree to which the two objects are alike. Consequently similarities are higher for pairs of objects that are more alike. Similarities are usually non-negative and are often between $[0, 1]$

Dissimilarity Measure:-

The dissimilarity between two objects is the numerical measure of the degree to which the two objects are different. Dissimilarity is lower for more similar pair object.

2) Z-score normalization:-

This is also known as zero-score normalization. The values for an attribute A are normalized based on the mean and standard deviation of 'A'. A value of 'A' is normalized to v' by computing.

$$v' = \frac{v - \bar{A}}{\sigma A}$$

Example:-

a) Suppose the mean and standard deviation of the value for the attribute income are 54,000 & 16,000 respectively. with z-score normalization a value of 73,600 is transformed to

$$v' = \frac{73,600 - 54,000}{16,000}$$

$$= 1.225$$

3) Decimal scaling normalization:-

By moving the decimal point of values of attributes 'A' the number of decimal point moved depends on maximum obsolute value of 'A'. A value 'v' of attribute 'A' is normalized to v' by computing

$$v' = \frac{v}{10^I}$$

where I is smallest integer such that $max(|v'|) < 1$

UNIT-II

Association rule mining

⇒ Association rule mining is a a step process.

1) finding frequent patterns
2) forming strong association rules. → For this we use pattern evaluation methods.

& Algorithms

1) Apriar : Algorithm
2) Fp growth Algorithm.

Apriar Algorithm:- R. Agarwal and R. Srikanth developed the algorithm in 1994.

⇒ Main purpose of the algorithm is to find frequent pattern and forming boolean rules.

⇒ This algorithm uses prior knowledge of item set properties

⇒ It uses level wise search

⇒ It is a a step process

1) join step
2) prune step.

Step 1:- Join step :- To find L_k a set of candidate k-items $[L_k(k\text{-items})]$ is generated by joining L_{k-1} with itself. This set of candidates is denoted by C_k

Step 2:- Prune step :- C_k is a superset of L_k i.e its members may (or) may not be frequent but all of the

K-frequent item sets are included in C_k to reduce the size of C_k. The Apriori property is used.

Eg:- By using apriori algorithm, find the items that are frequently purchased by the customers

TID	List of item - IDs
T100	I_1, I_2, I_5
T200	I_2, I_4
T300	I_2, I_3
T400	I_1, I_2, I_4
T500	I_1, I_3
T600	I_2, I_3
T700	I_1, I_3
T800	I_1, I_2, I_3, I_5
T900	I_1, I_2, I_3

Minimum support count = 2

Scan 'D' for count of each candidate.

C_1 (candidate itemset 1)

Itemset	Suppcount
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

compare candidate support count with minimum support count

L_1

Itemset	Support count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

Generate C_2 candidates from L_1

C_2 (candidates of 2)

Itemset
$\{I_1, I_2\}$
$\{I_1, I_3\}$
$\{I_1, I_4\}$
$\{I_1, I_5\}$
$\{I_2, I_3\}$
$\{I_2, I_4\}$
$\{I_2, I_5\}$
$\{I_3, I_4\}$
$\{I_3, I_5\}$
$\{I_4, I_5\}$

Scan 'D' for count of each candidate

Itemset	Support count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_4\}$	1
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2
$\{I_3, I_4\}$	0
$\{I_3, I_5\}$	1
$\{I_4, I_5\}$	0

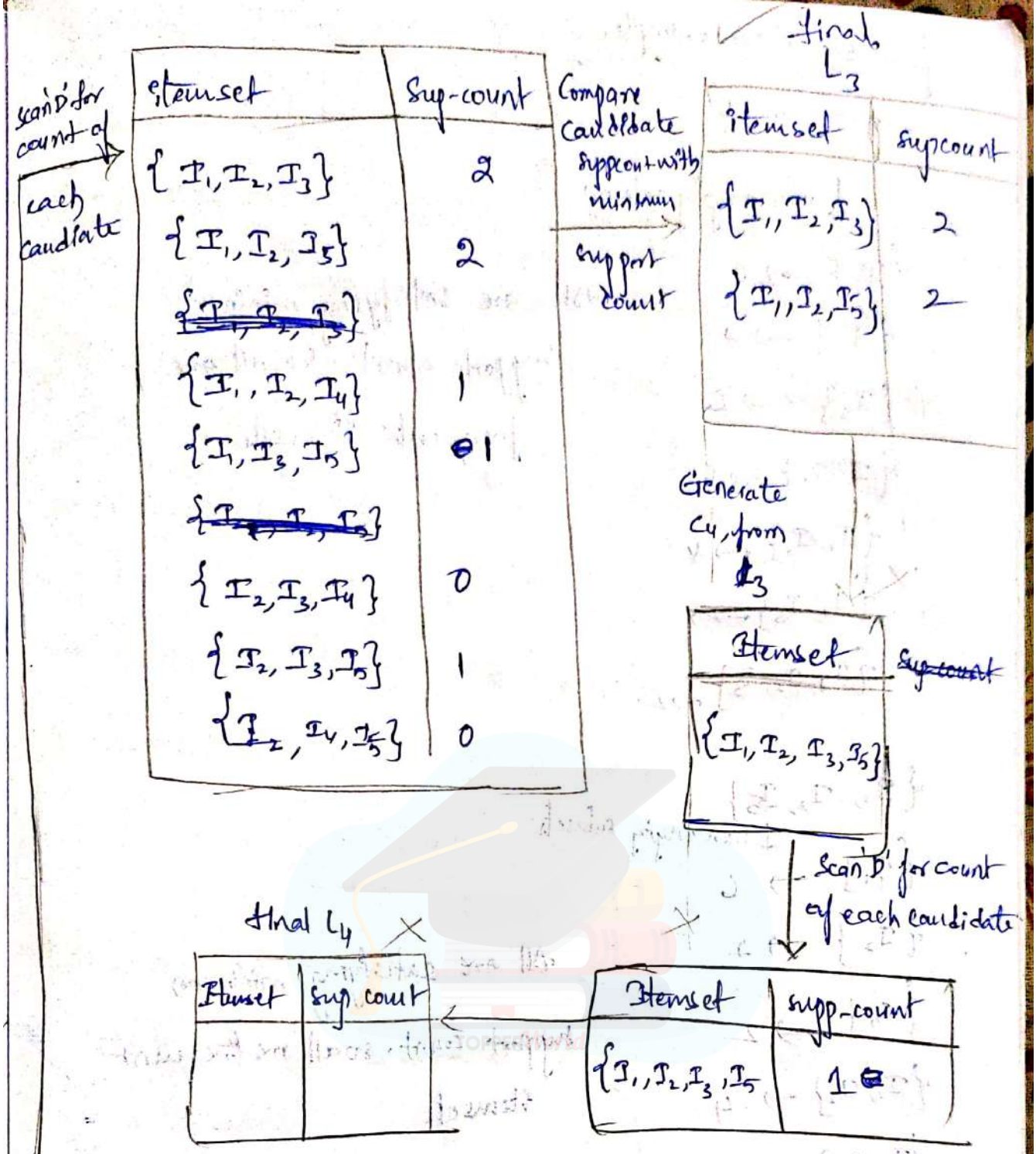
Find L_2 ↓ Compare candidate support count with minimum sup count

Itemset	Sup count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2

Generate C_3 Candidates from L_2 →

C_3

Itemset
$\{I_1, I_2, I_3\}$
$\{I_1, I_2, I_5\}$
~~$\{I_1, I_3, I_5\}$~~
$\{I_1, I_2, I_4\}$
$\{I_1, I_3, I_5\}$
~~$\{I_1, I_4, I_5\}$~~
$\{I_2, I_3, I_4\}$
$\{I_2, I_3, I_5\}$
$\{I_2, I_4, I_5\}$



Frequent itemsets are. $\{I_1, I_2, I_3\}$ and $\{I_1, I_2, I_5\}$

Apriori property :-

All non empty subsets of frequent itemsets must also be frequent

from the above example.

$\{I_1, I_2, I_3\}$

↓ non empty subsets

$\{I_1\} \rightarrow 6$

$\{I_2\} \rightarrow 7$

$\{I_3\} \rightarrow 6$

$\{I_1, I_2\} \rightarrow 4$

$\{I_1, I_3\} \rightarrow 4$

$\{I_2, I_3\} \rightarrow 4$

$\{I_1, I_2, I_3\} \rightarrow 2$

All are satisfying minimum support count, so all are frequent itemsets.

$\{I_1, I_2, I_5\}$

↓ non empty subsets

$\{I_1\} \rightarrow 6$

$\{I_2\} \rightarrow 7$

$\{I_5\} \rightarrow 2$

$\{I_1, I_2\} \rightarrow 4$

$\{I_1, I_5\} \rightarrow 2$

$\{I_2, I_5\} \rightarrow 2$

$\{I_1, I_2, I_5\} \rightarrow 2$

All are satisfying minimum support count, so all are frequent itemsets

Generate strong Association Rules from frequent Itemset (10m) :-

Association rule :- Eg :- 1) milk ⇒ Bread (antecedent ⇒ consequent)

2) buys (onion, potatoes) ⇒ buys (Tomatoes) (antecedent ⇒ consequent)

Def :- Association rules are if-then statements. They

It is used for analysing and predicting customer behaviour.

⇒ In data mining association rules are useful in uncovering relationship btw unrelated data.

⇒ Association rules contain 2 parts

1) Antecedent

2) Consequent

⇒ Antecedent :- can be found alone

⇒ Consequent will be found with combination of antecedent

Eg 3 :- computer ⇒ Antivirous software [support = 2%, confidence = 60%]

⇒ A support of 2% for above rule means that 2% of all transaction under analysis shows that computer and antivirous s/w are purchased together.

⇒ A Confidence of 60% means that 60% of customers who purchased a computer also purchased antivirous software

⇒ Typically association rules are considered interesting (or strong if they satisfy both a minimum support threshold and minimum confidence threshold.

⇒ These thresholds can be set by users. (or domain experts.

Generating strong Association rules from Frequent Itemset (IOM):-

$$\text{Support}(A \Rightarrow B) = P(A \cup B)$$

$$\text{Confidence}(A \Rightarrow B) = P(B/A)$$

$$= \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

$$= \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$$

⇒ Support count of $A \cup B$ = no. of transactions containing the items A and B.

⇒ Support count of A = no. of transactions containing the item A.

⇒ We can find strong association rules by using the confidence

⇒ for each frequent itemset 'l' generate all non empty subsets of 'l'.

⇒ for every non-empty subset 's' of 'l' output the rule "s ⇒ (l-s)" if support count of 'l'

if $\frac{\text{support count of } l}{\text{support count of } s} \geq$ minimum confidence.

Eg :- $\{I_1, I_2, I_5\} \rightarrow$ 'l'

↓ non empty subsets

"$s \Rightarrow (l-s)$" Confidence = support count $\{I_1 \cup I_2, I_5\}$ / support count $\{I_1\}$

R_1	's' → $\{I_1\}$	⟶ $\{I_2, I_5\}$	⇒ 2/6 * 100 = 33.33%.
R_2	$\{I_2\}$	⟶ $\{I_1, I_5\}$	⇒ 2/7 x 100 = 28.57%.
R_3	$\{I_5\}$	⟶ $\{I_1, I_2\}$	⇒ 2/2 * 100 = 100%.
R_4	$\{I_1, I_2\}$	⟶ $\{I_5\}$	⇒ 2/4 * 100 = 50%.
R_5	$\{I_1, I_5\}$	⟶ $\{I_2\}$	⇒ 2/2 * 100 = 100%.
R_6	$\{I_2, I_5\}$	⟶ $\{I_1\}$	⇒ 2/2 x 100 = 100%.
R_7	$\{I_1, I_2, I_5\}$	⟶ { } (or) ϕ	

minimum confidence = 60%

R_5, R_6, R_3 are strong association rules.

FR - Growth Algorithm :- It is a 2 step process.

1) FP - tree
2) Conditional databases

FP - tree construction / Generation :-

step 1 :- Write list of itemsets with support count

step 2 :- Write the items in descending order of their support-count.

step 3 :- Start construction of FP tree with null as root node (Arrange items of every transaction in descending order of supp-count)

step 4 :- Link to nodes.

Eg :-

T ID	List of items IO's
T_{100}	I_1, I_2, I_5
T_{200}	I_2, I_4
T_{300}	I_2, I_3
T_{400}	I_1, I_2, I_4
T_{500}	I_1, I_3
T_{600}	I_2, I_3
T_{700}	I_1, I_3
T_{800}	I_1, I_2, I_3, I_5
T_{900}	I_1, I_2, I_3

minimum support count = 2

min support confidence = 60%

step 1:-

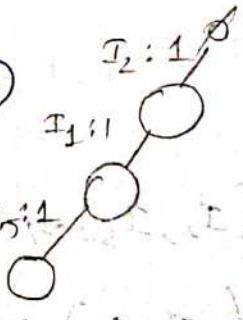
Itemset	support count
I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

step 2:-

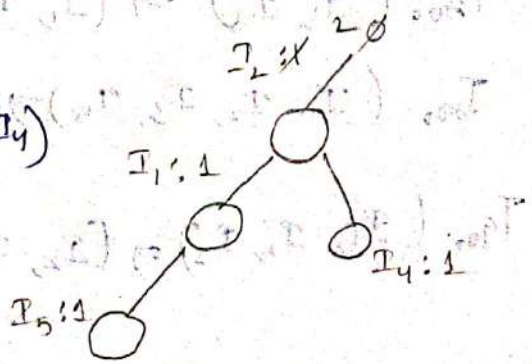
Itemset	support count
I_2	7
I_1	6
I_3	6
I_4	2
I_5	2

step 3:- $T_{100}(I_1, I_2, I_5) \Rightarrow (I_2, I_1, I_5)$

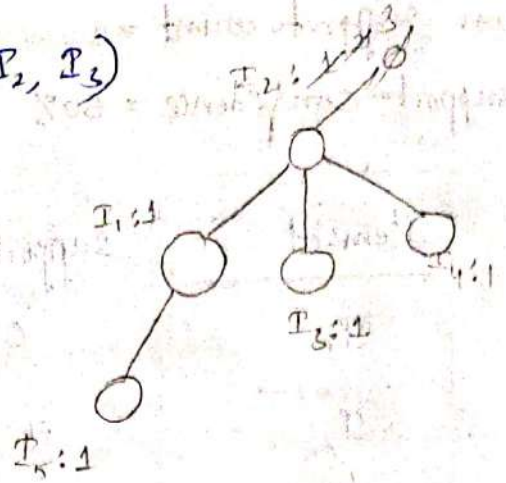
(start every transaction from ϕ)



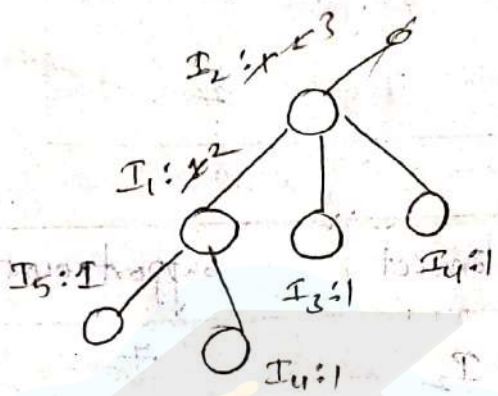
$T_{200}(I_2, I_4) \Rightarrow (I_2, I_4)$



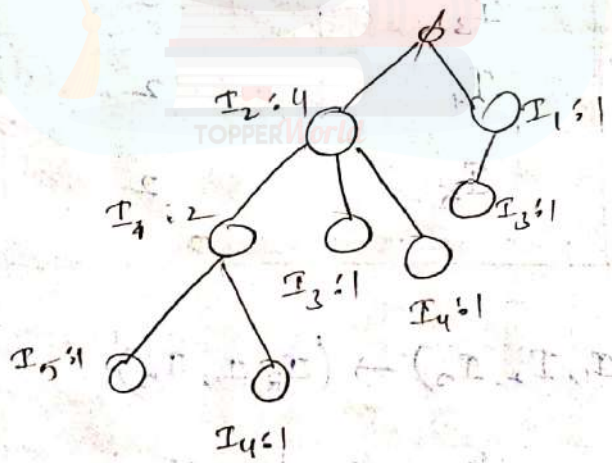
$T_{300}(I_1, I_3) \Rightarrow (I_2, I_3)$



$T_{400}(I_1, I_2, I_4) = (I_2, I_1, I_4)$



$T_{500}(I_1, I_3) \Rightarrow (I_1, I_3)$

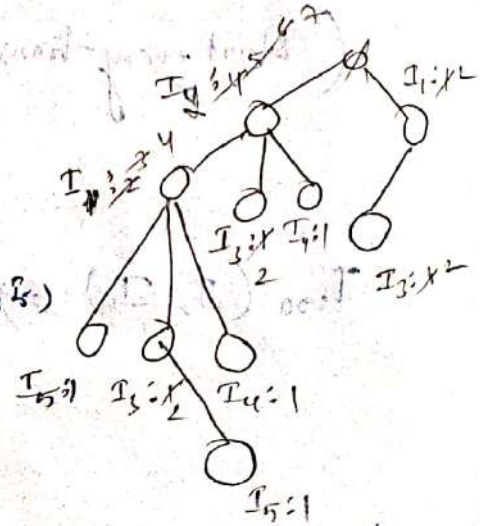


$T_{600}(I_2, I_3) \Rightarrow (I_2, I_3)$

$T_{700}(I_1, I_3) \Rightarrow (I_1, I_3)$

$T_{800}(I_1, I_2, I_3, I_5) \Rightarrow (I_2, I_1, I_3, I_5)$

$T_{900}(I_1, I_2, I_3) \Rightarrow (I_2, I_1, I_3)$



Step 4:-

Item ID	support count	Node link
I_2	7	
I_1	6	
I_3	6	
I_4	2	
I_5	2	

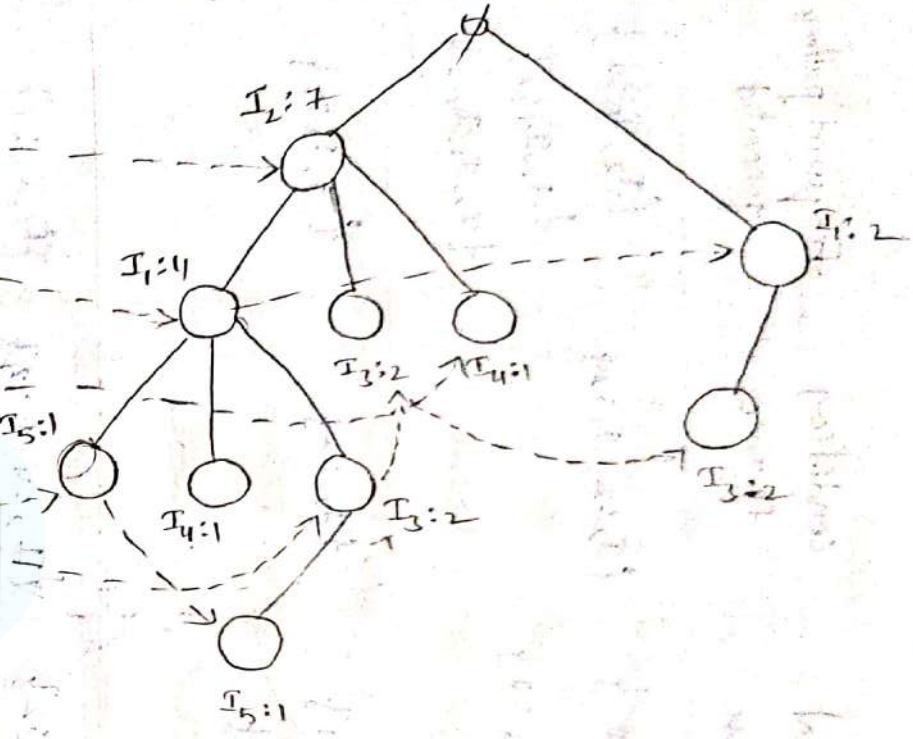


Fig:- An FI-Tree registers compressed frequent patterns information ..

Conditional databases:-

Item	Conditional pattern base (path in the tree)	Conditional FP-Tree (entire all paths)	Frequent patterns generated.
I_5 (least sup. count)	$\{I_2, I_1 : 1\}$, $\{I_2, I_1, I_3 : 1\}$ ($I_3 : 1 < 2$) X (1 = count a node)	$\{I_2 : 2, I_1 : 2\}$	$\{I_2, I_5 : 2\}$, $\{I_1, I_5 : 2\}$, $\{I_2, I_1, I_5 : 2\}$
I_4 (next least sup. count)	$\{I_2, I_1 : 1\}$, $\{I_2 : 1\}$, $I_1 : 1 < 2$ X	$\{I_2 : 2\}$	$\{I_4, I_2 : 2\}$
I_3	$\{I_2, I_1 : 2\}$ (left), $\{I_2 : 2\}$, $\{I_1 : 2\}$ (right to I_3)	$\{I_2 : 4, I_1 : 4\}$ ~~$\{I_1 : 2\}$~~	$\{I_2, I_3 : 4\}$, $\{I_1, I_3 : 4\}$, $\{I_2, I_1, I_3 : 4\}$
I_1	$\{I_2 : 4\}$ X (no intermediate node)	$\{I_2 : 4\}$	$\{I_1, I_2 : 4\}$
I_2	No intermediate node		

for a given transactional data set generate rules by using apriar algorithm. Consider the values as support = 50%, confidence = 75%.

Transaction ID	Items purchased.
1	Bread, cheese, egg, juice
2	Bread, cheese, juice.
3	Bread, milk, yogurt
4.	Bread, juice, milk
5	cheese, juice, milk.

Item	frequency	support
Bread	4	$\frac{4}{5} \times 100 = 80\%$
cheese	3	$\frac{3}{5} \times 100 = 60\%$
egg	1	$\frac{1}{5} \times 100 = 20\%$
juice	4	$\frac{4}{5} \times 100 = 80\%$
milk	3	$\frac{3}{5} \times 100 = 60\%$
yogurt	1	$\frac{1}{5} \times 100 = 20\%$

egg, yogurt not have the support value i.e 50% to remove both, i.e support of egg and yogurt is less than 50%.

Item	Frequency	Support
{Bread, cheese}	2	$\frac{2}{5} \times 100 = 40\%$ ✗
{Bread, juice}	3	$\frac{3}{5} \times 100 = 60\%$ ✓
{Bread, milk}	2	$\frac{2}{5} \times 100 = 40\%$ ✗
{cheese, juice}	3	$\frac{3}{5} \times 100 = 60\%$ ✓
{cheese, milk}	1	$\frac{1}{5} \times 100 = 20\%$ ✗
{juice, milk}	2	$\frac{2}{5} \times 100 = 40\%$ ✗

only {Bread, juice} and {cheen, juice} satisifies the support count. so consider only these two sets and remove the remaining sets.

Item	Frequency	support
{Bread, juice, cheese}	2	$\frac{2}{5} \times 100 = 40\%$

As the set {Bread, juice, cheen} donot satisfy the support count. so consider the previous table frequencies which satisfy the support i.e {Bread, juice} and {cheen, juice}

1. {Bread, juice}

Rule 1:- Bread ⇒ juice → Confidence

$$= \frac{\text{Support count}(B \cup J)}{\text{Support count}(B)}$$

$$= \frac{3/5}{4/5} = \frac{3}{4} = 75\%$$

Rule 2:- Juice → Bread = $\frac{\text{Support count } (J \cup B)}{\text{Support } (J)}$

$= \frac{3/5}{4/5}$

$= 75\%$

2) {cheese, juice} Confidence

Rule 3:- {cheese ⇒ juice = $\frac{3/5}{3/5} = 100\%$

Rule 4: juice ⇒ cheese = $\frac{3/5}{4/5} = 75\%$

All Rules i.e Rule 1, 2, 3, 4. are satisfying the minimum confidence threshold i.e 75%

Hence all rules are strong association rules.

Eg:- 3

For the following given transaction data set generate rules using apriar algorithm. Consider values as support = 22% and confidence = 70%

Transaction ID	Items purchased.
1	I_1, I_2, I_5
2	I_2, I_4
3	I_2, I_3
4	I_1, I_2, I_4
5	I_1, I_3
6	I_2, I_3
7	I_1, I_3
8	I_1, I_2, I_3, I_5
9	I_1, I_2, I_3

Item	Frequency	Support
I_1	6	$\frac{6}{9} \times 100 = 66.6 = 66$
I_2	7	$\frac{7}{9} \times 100 = 77.7 = 77$
I_3	6	$\frac{6}{9} \times 100 = 66.6 = 66$
I_4	2	$\frac{2}{9} \times 100 = 22.2 = 22$
I_5	2	$\frac{2}{9} \times 100 = 22.2 = 22$

→ All items are ~~support~~ satisfying support count.

Item	Frequency	Support
$\{I_1, I_2\}$	4	$\frac{4}{9} \times 100 = 44$
$\{I_1, I_3\}$	4	$\frac{4}{9} \times 100 = 44$
$\{I_1, I_4\}$	1	$\frac{1}{9} \times 100 = 11.11 \times$
$\{I_1, I_5\}$	2	$\frac{2}{9} \times 100 = 22$
$\{I_2, I_3\}$	4	$\frac{4}{9} = 44$
$\{I_2, I_4\}$	2	$\frac{2}{9} = 22$
$\{I_2, I_5\}$	2	$\frac{2}{9} = 22$
$\{I_3, I_4\}$	0	$\frac{0}{9} = 0 \times$
$\{I_3, I_5\}$	1	$\frac{1}{9} = 11.11 \times$
$\{I_4, I_5\}$	0	$\frac{0}{9} = 0 \times$

Item	frequency	support
$\{I_1, I_2\}$	4	$\frac{4}{9} = 44$
$\{I_1, I_4\}$	4	$\frac{4}{9} = 44$
$\{I_1, I_5\}$	2	$\frac{2}{9} = 22$
$\{I_2, I_3\}$	4	$\frac{4}{9} = 44$
$\{I_2, I_4\}$	2	$\frac{2}{9} = 22$
$\{I_2, I_5\}$	2	$\frac{2}{9} = 22$

Item	frequency	support
$\{I_1, I_2, I_4\}$	2	$\frac{2}{9} \times 100 = 22$
$\{I_1, I_2, I_5\}$	2	$\frac{2}{9} = 22$
$\{I_1, I_2, I_4\}$	1	$\frac{1}{9} = 11.11$ ×
$\{I_1, I_3, I_5\}$	1	$\frac{1}{9} = 11.11$ ×
~~$\{I_1, I_2, I_3\}$~~		
$\{I_2, I_3, I_4\}$	0	$\frac{0}{9} = 0$ ×
$\{I_2, I_3, I_5\}$	1	$\frac{1}{9} = 11.11$ ×
$\{I_2, I_4, I_5\}$	0	$\frac{0}{9} = 0$ ×

Item	frequency	support
$\{I_1, I_2, I_3\}$	2	$\frac{2}{9} = 22$
$\{I_1, I_2, I_5\}$	2	$\frac{2}{9} = 22$

Item	frequency	Support
$\{I_1, I_2, I_3, I_5\}$	1	$\frac{1}{9} = 4.4$ ✗

It does not ~~support~~ satisfy support count. Consider the previous frequency sets

2
9
6
9

1). $\{I_1, I_2, I_3\}$

2) $\{I_1, I_2, I_5\}$

Confidence

1) $\{I_1, I_2, I_3\}$

Subset ⇒ supp-count $I_1 \cup I_2 \cup I_3$ / sup-(I_1)

non empty sets

R_1 $\{I_1\} \longrightarrow \{I_2, I_3\} \rightarrow \frac{2/9}{6/9} \times 100 = 33.33$ ✗

R_2 $\{I_2\} \rightarrow \{I_1, I_3\} \rightarrow \frac{2/9}{7/9} \times 100 = 28.57$ ✗

R_3 $\{I_3\} \rightarrow \{I_1, I_2\} \rightarrow \frac{2/9}{6/9} \times 100 = 33.33$ ✗

R_4 $\{I_1, I_2\} \rightarrow \{I_3\} \rightarrow \frac{2/9}{4/9} \times 100 = 50$ ✗

R_5 $\{I_1, I_3\} \rightarrow \{I_2\} \rightarrow \frac{2/9}{4/9} \times 100 = 50$ ✗

R_6 $\{I_2, I_3\} \rightarrow \{I_1\} \rightarrow \frac{2/9}{4/9} \times 100 = 50$ ✗

R_7 $\{I_1, I_2, I_3\} \rightarrow \phi \rightarrow \phi$

2) $\{I_1, I_2, I_5\}$

non empty sets

confidence

R_8 $\{I_1\} \rightarrow \{I_2, I_5\} \rightarrow \frac{2/9}{6/9} \times 100 = 33.33$ ✗

R_9 $\{I_2\} \rightarrow \{I_1, I_5\} \rightarrow \frac{2/9}{7/9} \times 100 = 28.57$ ✗

R_{10} $\{I_5\} \rightarrow \{I_1, I_2\} \rightarrow \frac{2/9}{2/9} \times 100 = 100$ ✓

R_{11} $\{I_1, I_2\} \rightarrow \{I_5\} \rightarrow \frac{2/9}{4/9} \times 100 = 50$ ✗

R_{12} $\{I_1, I_5\} \rightarrow \{I_2\} \rightarrow \frac{2/9}{2/9} \times 100 = 100$ ✓

R_{13} $\{I_2, I_5\} \rightarrow \{I_1\} \rightarrow \frac{2/9}{2/9} \times 100 = 100$

R_{14} $\{I_1, I_2, I_5\} \rightarrow \emptyset \rightarrow \emptyset$

R_{10}, R_{12}, R_{13} satisfy the minimum confidence threshold so these are strong association rules

Eg:
Generate FP tree for the following transaction Dataset.

minimum support = 30%

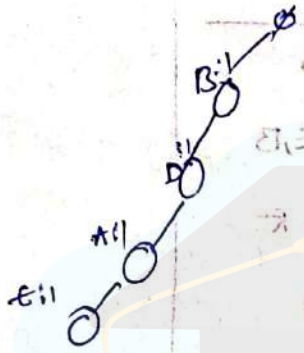
Tr. ID	Items.
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

Item set	frequency	Support count
A	5	5/8 = 62
B	6	6/8 = 75
C	3	3/8 = 37
D	6	6/8 = 75
E	4	4/8 = 50%

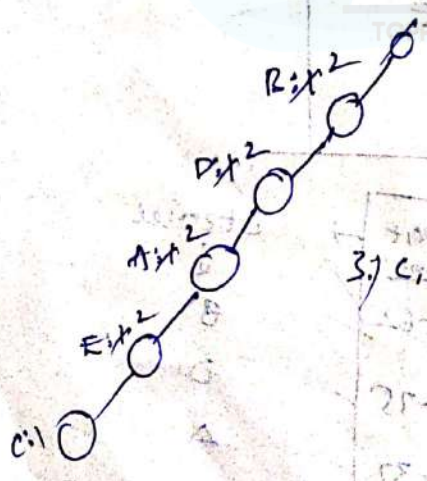
Itemset
B
D
A
E
C

Itemset	frequency	Support count
B	6	$\frac{6}{8} = 75$
D	6	$\frac{6}{8} = 75$
A	5	$\frac{5}{8} = 50$
E	4	$\frac{4}{8} = 62$
C	3	$\frac{3}{8} = 37$

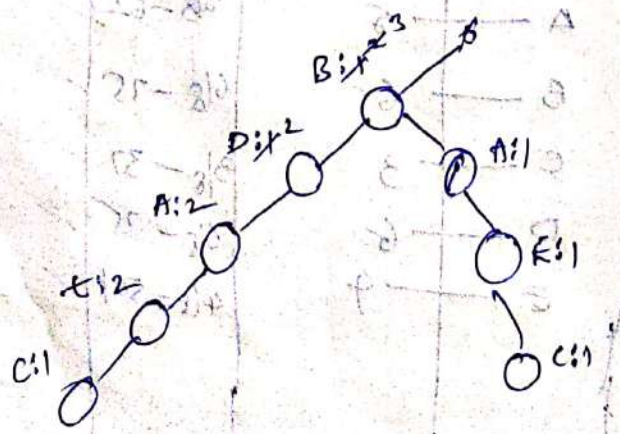
1. E, A, D, B → B D A E



2. D, A, C, E, B → B, D, A, E, C

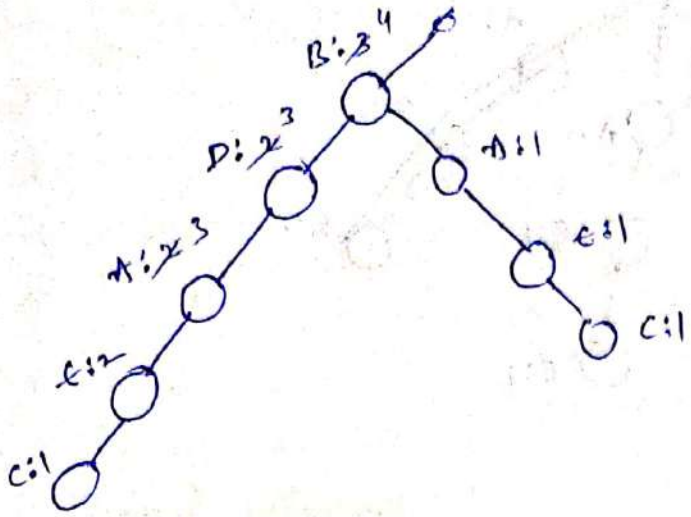


3. C, A, B, E → B, A, E, C

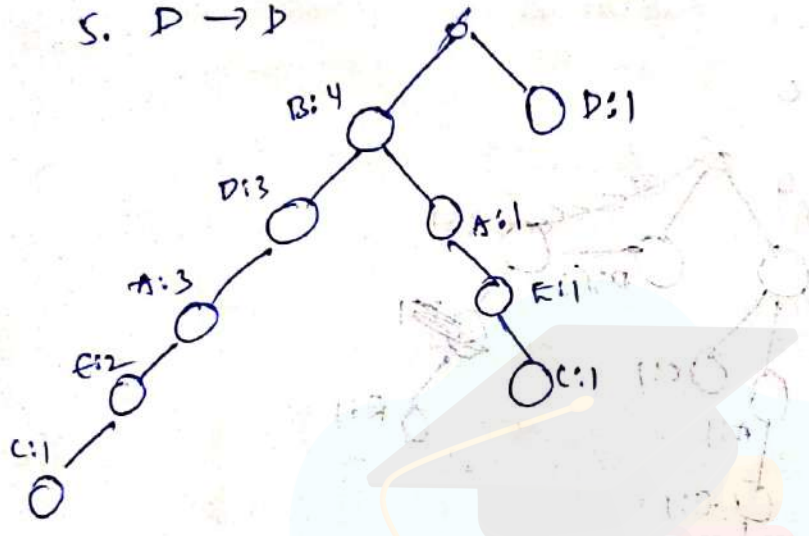


Q.

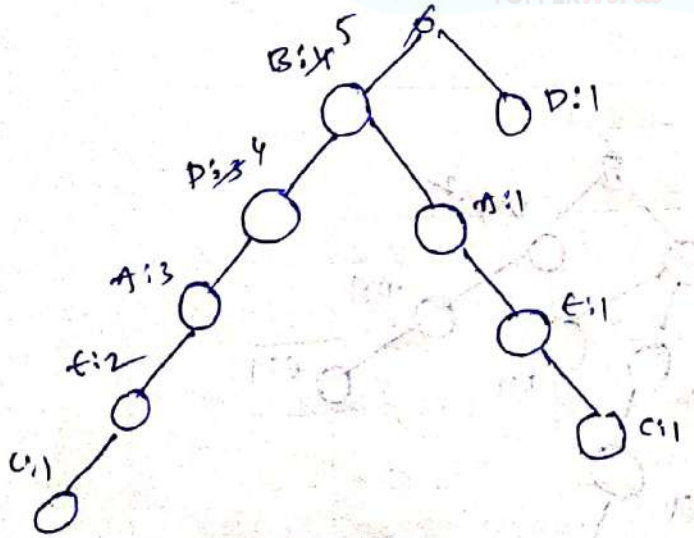
4. B, A, D → B, D, A



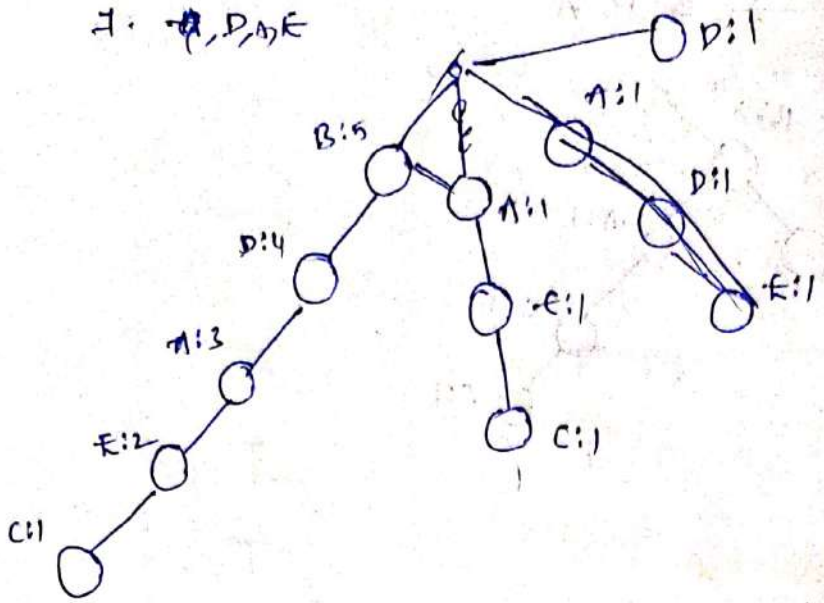
5. D → D



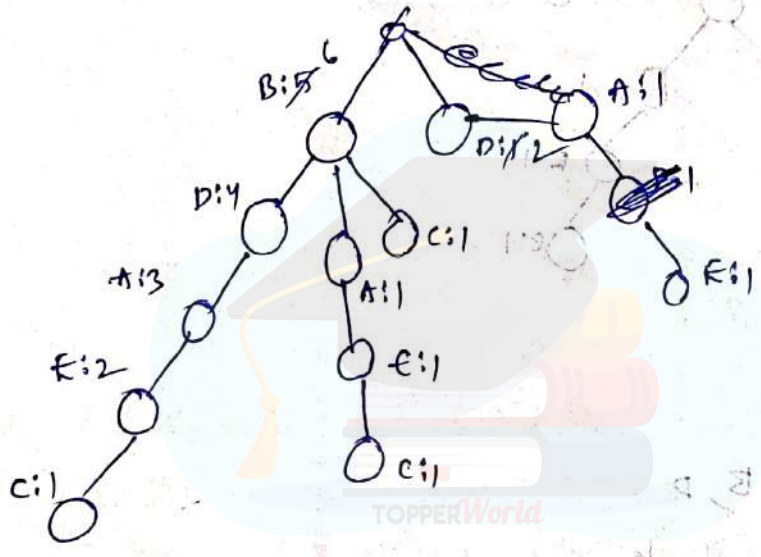
6. D, B → B, D



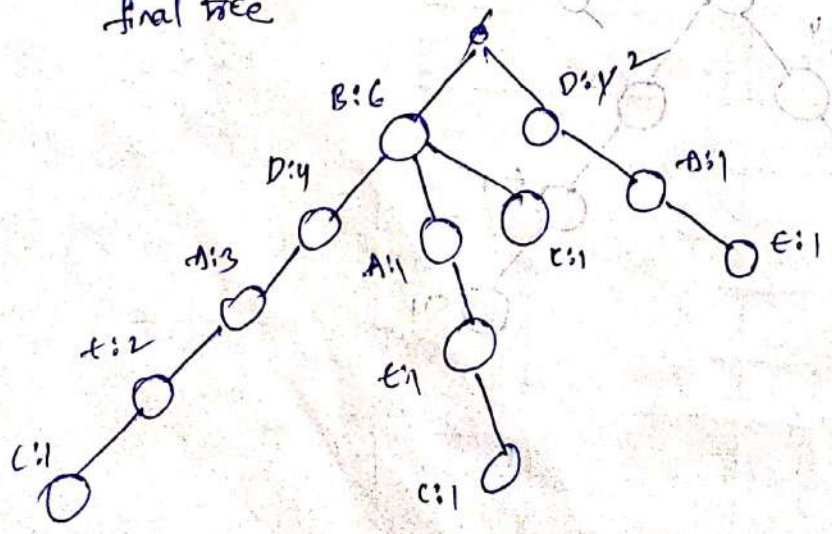
7. A, D, A, E



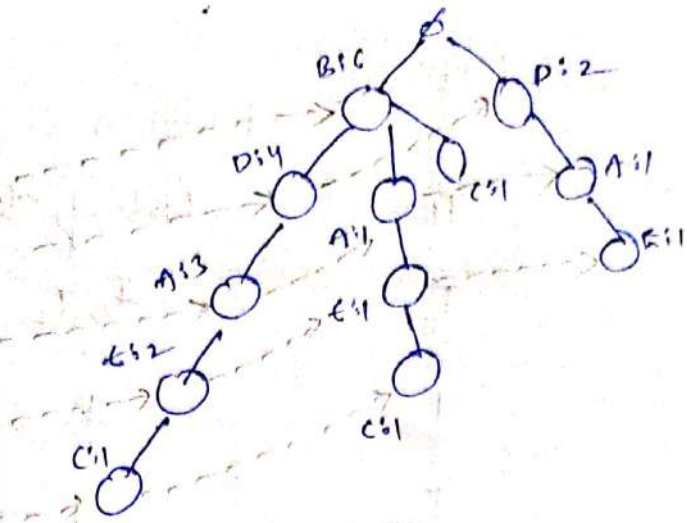
8. B, C → B, C



final tree



Item	support count frequency	node link
B	6	•
D	6	•
A	5	•
E	4	•
C	3	•



Conditional database:-

Item	Conditional pattern base	condition FP tree	Frequent patterns generated
C	{B, D, A, E:1} {B, A, E:1} {B:1}	{B:3, }	{B, C:3}
E	{D, A:1} {B, D, A:1} {B, A:1}	{A:4, B:3, D:3,}	{A, E:3} {B, E:3} {D, B:3} {A, B

Eclat Algorithm:-

Minimum support = 2

TID	List of Item IDs
T_{100}	I_1, I_2, I_5
T_{200}	I_2, I_4
T_{300}	I_2, I_3
T_{400}	I_1, I_2, I_4
T_{500}	I_1, I_3
T_{600}	I_2, I_3
T_{700}	I_1, I_3
T_{800}	I_1, I_2, I_3, I_5
T_{900}	I_1, I_2, I_3

ECLAT - Equivalent class transformation.

Vertical dataformat :- {item : TID -set}

Horizontal data format : { TID : Itemset}

Vertical data format:

Item set	Tid-set
I_1	$\{T_{100}, T_{400}, T_{500}, T_{700}, T_{800}, T_{900}\}$
I_2	$\{T_{100}, T_{200}, T_{300}, T_{400}, T_{600}, T_{800}, T_{900}\}$
I_3	$\{T_{300}, T_{500}, T_{600}, T_{700}, T_{800}, T_{900}\}$
I_4	$\{T_{200}, T_{400},\}$
I_5	$\{T_{100}, T_{800}\}$

Finding 2 item set in vertical data format:

Itemset	TID-set
$\{I_1, I_2\}$	$\{T_{100}, T_{400}, T_{800}, T_{900}\}$
$\{I_1, I_3\}$	$\{T_{500}, T_{700}, T_{800}, T_{900}\}$
$\{I_1, I_4\}$	$\{T_{400}\}$ X
$\{I_1, I_5\}$	$\{T_{100}, T_{800}\}$
$\{I_2, I_3\}$	$\{T_{300}, T_{600}, T_{800}, T_{900}\}$
$\{I_2, I_4\}$	$\{T_{200}, T_{400}\}$
$\{I_2, I_5\}$	$\{T_{100}, T_{800}\}$
$\{I_3, I_4\}$	ϕ X
$\{I_3, I_5\}$	$\{T_{800}\}$ X
$\{I_4, I_5\}$	ϕ X

Find 3 item set in vertical data format

Itemset	TID-set
$\{I_1, I_2, I_3\}$	$\{T_{800}, T_{900}\}$
$\{I_1, I_2, I_4\}$	$\{T_{400}\}$ X
$\{I_1, I_3, I_5\}$	$\{T_{800}\}$ X
$\{I_1, I_2, I_5\}$	$\{T_{100}, T_{800}\}$
$\{I_2, I_3, I_4\}$	ϕ X
$\{I_2, I_3, I_5\}$	$\{T_{800}\}$ X
$\{I_2, I_4, I_5\}$	ϕ X

Find 4 item set in vertical data format

Item set	Tid-set
$\{I_1, I_2, I_3, I_5\}$	$\{T800\}$ X

CLUSTERING

Definition :-

Clustering is the process of partitioning a set of data objects into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. A cluster of data objects can be treated as one group.

- The process of partitioning data objects into subclasses is called as cluster.
- Clustering is also called as data Segmentation, because it partitions large data sets into groups according to their Similarity.

Applications :-

- Business Intelligence
- Image Pattern recognition
- Web search
- Biology
- Security.

In marketing field clustering helps to find group of customers with similar behaviour from a given dataset customer record.

In biology classification of plants and animal according to their features.

In library clustering is very useful in book ordering

- clustering is sometimes called automatic classification.
- clustering is known as unsupervised learning because the class label information is not present.

Why?

clustering is very much important as it determines the intrinsic grouping among the unlabeled data present.

Requirements:-

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Requirements for domain knowledge to determine input parameters
- Ability to deal with noisy data.
- Incremental clustering and insensitivity to input order
- Capability of clustering high-dimentional data.
- Constraint based clustering
- Interpretability and usability.

→ we need highly scalable clustering algorithms to deal with large databases.

→ Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical and binary data.

→ clustering algorithm should be capable of detecting cluster of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.

→ High dimencionality :- The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.

→ Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to ~~to~~uch data and may lead to poor quality clusters.

Clustering Methods :-

Clustering methods can be clasified into following categories.

- Partitioning method
- Hierarchical method
- Density-based method
- Grid based method

Partitioning method:-

Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of each data. Each partition will represent a cluster and $k \leq n$. It means that it classifies the data into k groups, which satisfy the following requirements

It conducts one-level partitioning on dataset. The basic partitioning methods typically adopt exclusive cluster seperation.

- Each group contains atleast one object
- Each object must belong to exactly one group.
- For a given number of partitions (say k) the partitioning method will create an initial partitioning
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

Hierarchical Methods:-

This method creates a hierarchical decomposition of the given set of data objects. There are two approaches

- Agglomerative approach (Bottom-up approach)
- Divisive approach (Top-down)

Agglomerative approach:-

This apprach is also known as bottom-up approach. In this we start with ~~all~~ each of the objects in the same cluster. In the continuous iteration forming seperate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive Approach:-

This approach is also known as top-down approach. In this we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid i.e once a merging or splitting is done, it can never be undone.

Approaches to improve Quality of Hierarchical clustering:-

Two approaches

① Perform careful analysis of object linkages at each hierarchical partitioning.

② Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

Density-based Method:-

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighbourhood exceeds some threshold i.e for each data point within a given cluster, the radius of a given cluster has to contain atleast a minimum number of points.

Grid-based Method:-

In this method, a model is hypothesized for each cluster to find

In this the objects together form a grid. The object space is quantized to finite no. of cells that form a grid structure.

Advantages:-

- The major advantage is fast processing time.
- It is dependent only on the numbers of cells in each dimensions in the quantized space.

Method	General characteristics
Partitioning Method	- Find mutually exclusive clusters of spherical shape - Distance-based - May use mean (or) medoid to represent cluster center - Effective for small-to-medium size data sets
Hierarchical Method	- clustering is hierarchical decomposition (i.e multiple levels) - cannot correct erroneous merges or splits - May incorporate other techniques like microclustering or consider object "linkages"
Density-based methods	- can find arbitrarily shaped clusters - clusters are dense regions of objects in space that are seperated by low-density regions - cluster density - May filter out outliers.
Grid-based methods	- use a multiresolution grid data structure - fast processing time.

Partitioning Methods

① K-Means

② K- Medoids

Hierarchical Methods

① Diana

② Agnes, BIRCH, ROCK, CHAMELEON

Density - based

① DBSCAN

② OPTICS

③ DenClue

Grid based

① DB SCAN

② OPTICS

③ Denclue.

K-Means clustering algorithm:

K-Means performs division of objects into clusters. The term K is basically a number. If K=2, we have two clusters if we have K=3 then three clusters.

We have to produce K clusters.

objects $X = \{x_1, x_2, \ldots, x_m\}$

m objects

Each objects is described in terms of n features.

$$x_i = (x_{i1}, x_{i2}, x_{i3} \cdots x_{in})$$

we should get output as k clusters.

$$S = (S_1, S_2, \ldots, S_K)$$

S_i is represented by cluster center u_i;

Steps:-

① Take mean value

② Find the nearest number to mean and put it in the cluster.

③ Repeat ① & ② until we get same mean.

Eg:-

$S = \{2, 3, 4, 10, 11, 12, 20, 25, 30\}$.

If $K = 2$

Randomly we take

$m_1 = 4$ $\quad$ $m_2 = 12$

By following step ①

$K_1 = \{2, 3, 4\}$ $\quad m_1 = \frac{2+3+4}{3} = 3$.

$K_2 = \{10, 11, 12, 20, 25, 30\}$ $\quad m_2 = \frac{108}{6} = 18$

$m_1 = 3$ $\quad$ $m_2 = 18$

again nearest values to 3 & 18

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

$m_1 = \frac{19}{4} = 4.75$ $m_2 = 19.6$

$m_1 = 5$ $m_2 = 20$.

Again find out nearest cluters

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

$\frac{25+30}{2}$

$m_1 = 7$ $m_2 = 25$

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

$m_1 = 7$ 1st group

$m_2 = 25$ 2nd group

K- Medoid algorithm :- (PAM - Partitioning Around Method)

→ Arrange values in increasing order and take middle values as medoid.

→ when your data set is $\{1, 2, 4\}$ then 2 is middle one

when it is $\{1, 2, 3, 4\}$ then take average of 2 & 3 i.e $\frac{2+3}{2} = 2.5$

→ In single dimensions it is ok to arrange points in increasing order. In multi dimensional ordering is complex. Definition to higher dimension we use medoid.

$S = \{x_1, x_2 \cdots, x_n\} \subseteq R^M$ ↳ m-dimensions.

↓ data set

d : euclidean distance

K - medoid is also known as partitioning around medoid. This was proposed by kaufman and Rousseeuw. A medoid can be defined as point in a cluster.

Medoid (c_i)

object (p_i)

$$c = \sum_{c_i} \sum_{p_i \in c_i} |p_i - c_i|$$

Algorithm:-

1. Initialize:
   select k random points out of n data points as the medoids.
2. Associate each data point to the closest medoid by using any common distance metric methods.
3. While the cost decreases
   For each medoid m, for each data point o which is not a medoid
   (i) swap m and o, associate each data point to the closest medoid, recompute the cost.
   (ii) If the total cost is more than that in previous step undo the swap

Example:-

We use Manhattan distance i.e

$$(x_1, y_1) \& (x_2, y_2)$$
$$= |x_1 - x_2| + |y_1 - y_2|$$

S.No	x	y	Distance from c_1 (3,4)	Distance from c_2 (7,4)
1	2	6	$\|2-3\|+\|6-4\| = 3$	$\|2-7\|+\|6-4\| = 7$
2	3	4		
3	3	8	$\|3-3\|+\|8-4\| = 4$	$\|3-7\|+\|8-4\| = 8$
4	4	7	$\|4-3\|+\|7-4\| = 4$	$\|4-7\|+\|7-4\| = 6$
5	6	2	$\|6-3\|+\|2-4\| = 5$	$\|6-7\|+\|2-4\| = 3$
6	6	4	$\|6-3\|+\|4-4\| = 5$	$\|6-7\|+\|4-4\| = 1$
7	7	3	$\|7-3\|+\|3-4\| = 5$	$\|7-7\|+\|3-4\| = 1$
8	7	4		
9	8	5	$\|8-3\|+\|5-4\| = 6$	$\|8-7\|+\|5-4\| = 2$
10	7	6	$\|7-3\|+\|6-4\| = 6$	$\|7-7\|+\|6-4\| = 2$

Step 1:- We first select 2 medoids (3,4) & (7,4)

Step 2:- We calculate the distance between the rest of data points and both medoids.

Step 3:- We calculate the total cost involved in forming the cluster using these medoid.

Step 4:- We again choose some other medoids & repeat step 1 to step 2. If we don't get better cost we will stop.

→ min_point

Minimum distance is considered

Total cost = 3 + 4 + 4 + 3 + 1 + 1 + 2 + 2

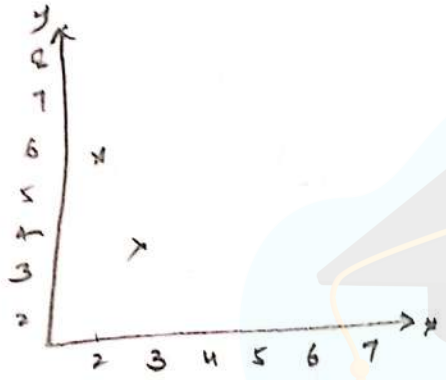
= 20

clusters with medoid {3, 4} are

{{3,4} {2,6} {4,7} {3,8}}

{7, 4} are

{{7,4} {6,2} {6,4} {7,3} {8,5} {7,6}}



Hierarchical Method:-

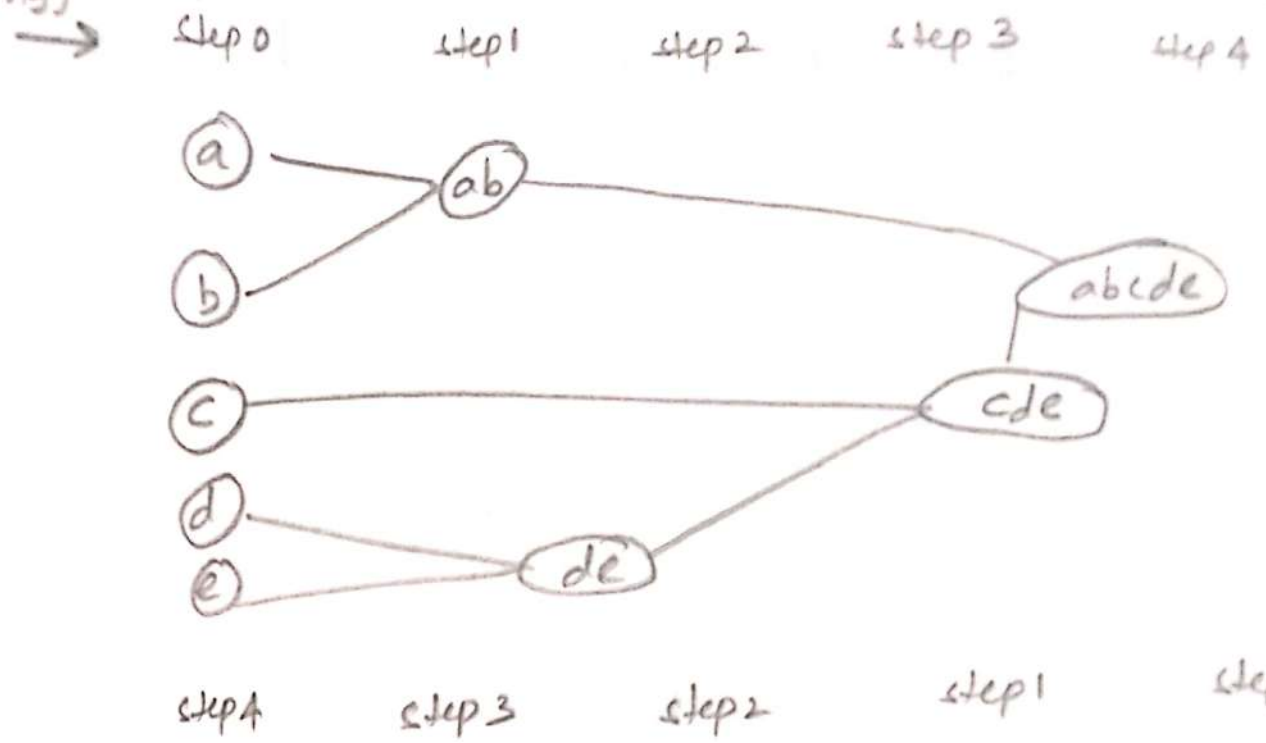
A hierarchical clustering method works by grouping data objects into a hierarchy or tree structure.

Representing data objects in the form of a hierarchy is useful for summarization and visualization.

eg:- Manager of human resources at All electronics may organize your employees into major groups such as executives, managers and staff.

senior officers — officers — trainees.

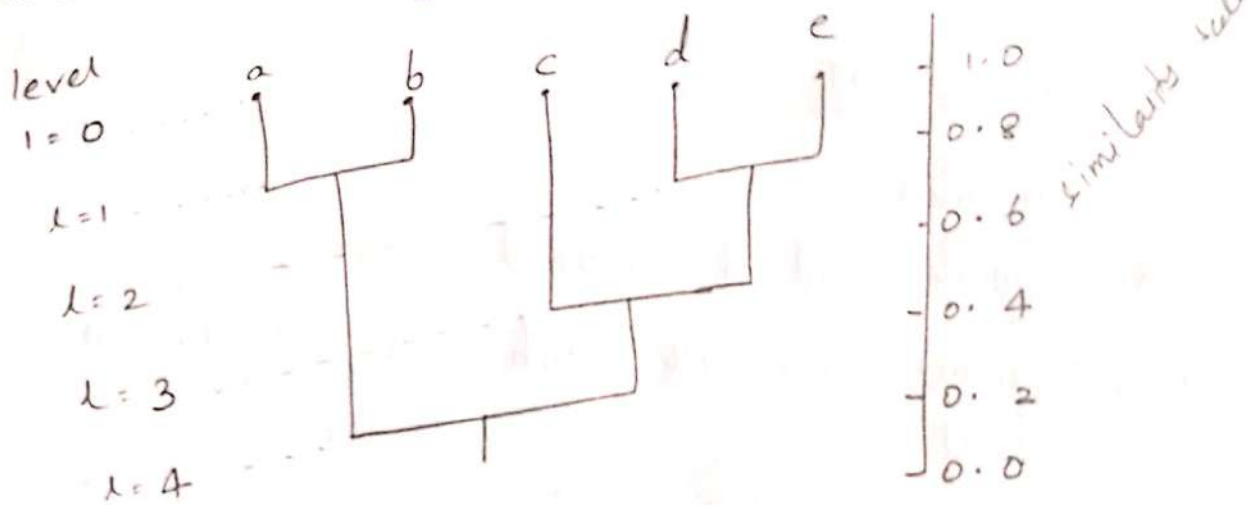
Agglomerative



Agglomerative and divisive hierarchical clustering on data objects {a, b, c, d, e}.

Dendrogram:

It is used to represent the process of hierarchical clustering.



At $l = 0$ five objects as singleton clusters.

At $l = 1$ a & b are grouped together.

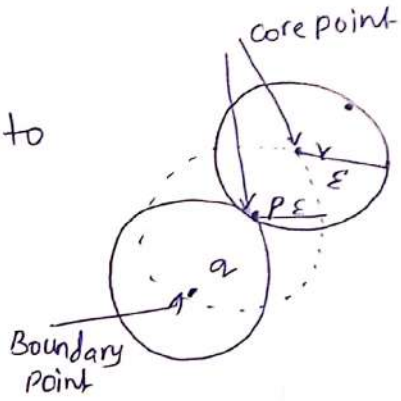
DBSCAN:

Density Based Spatial clustering of Applications with noise.

Eps.

Minimum points = 3.

→ Two points belong to same region

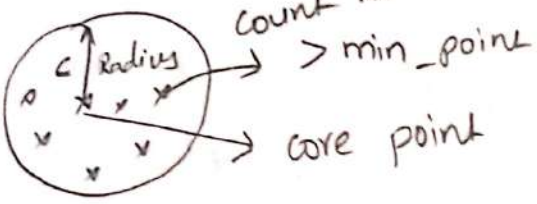


→ Density is lower.

Two parameters

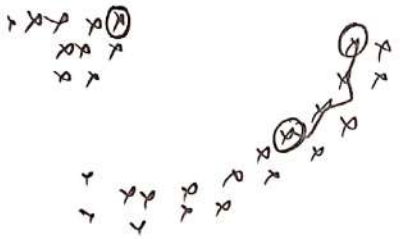
→ Min points : kind of threshold value

→ ε (epsilon) : Area over which you will perform count. (radius)



Core point is a point lies in high density.

A point is Density reachable by traversing only through core point.



Density connected

i & j are density connected if there exists core point K from which both of them are density reachable. then i & j are density connected.

→ i & j are in same cluster if and only if density connected

→ DBSCan computation is signitant

→ All kinds of arbitraral clustering.

Optics

ordering data such points such that different points of

- Neighbourhood - ε
- Min - point
- core point
- Border point
- Noise point