Data mining:

Data mining is defined as produce extracting of information from huge set of data or large amount of data

or

Data mining refers to extracting or mining knowledge

* It is three types, they are.

1. Data base
2. Data ware housing
3. Data translation

Data mining is also called as knowledge discovery, knowledge extension, data analaysis etc....

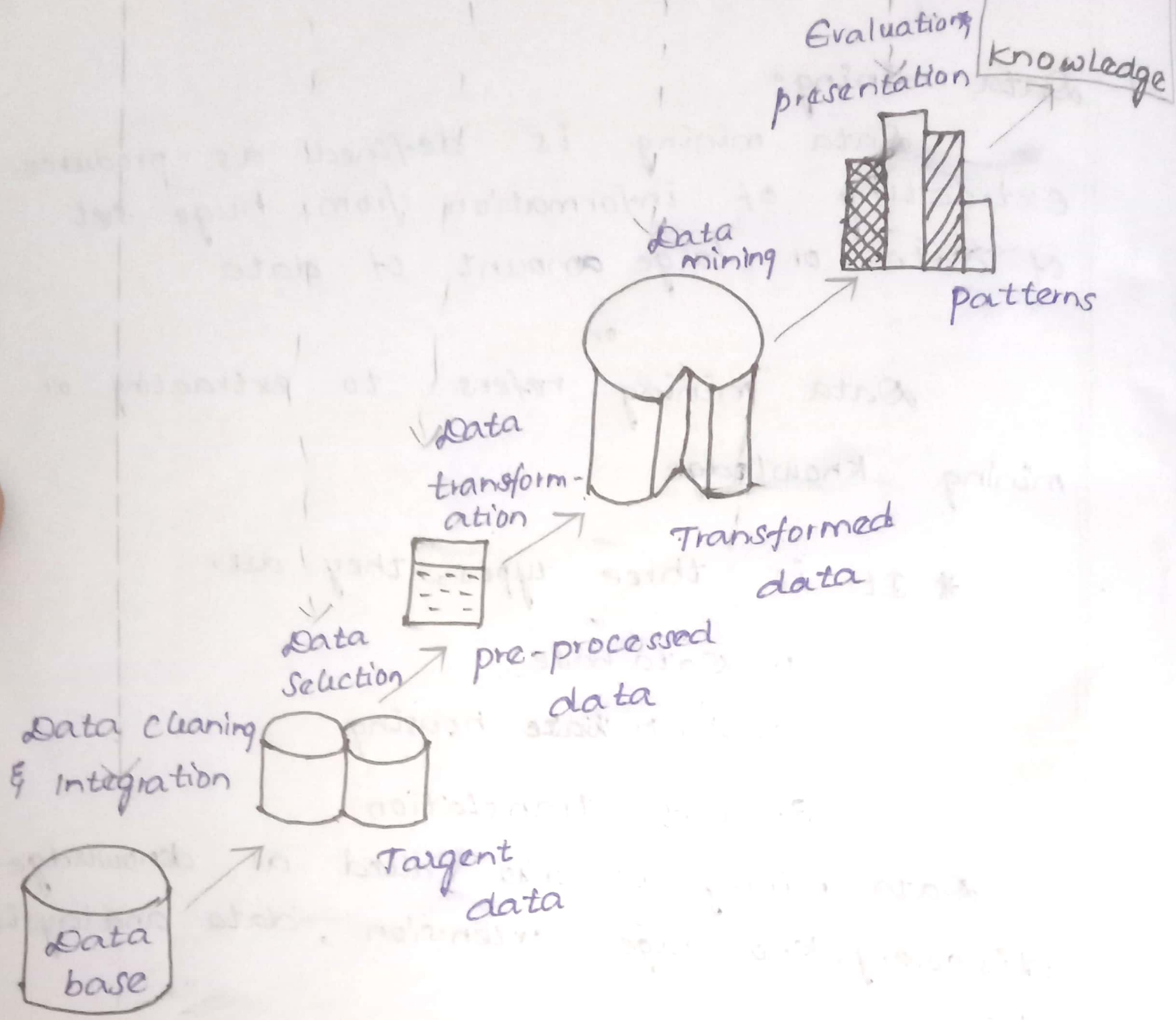
* uses in data mining
    * Banking sector
    * marketing
    * Medicine
    * Television / radio
    * Retial

10m

Data mining is treated KDD

knowledge discovery from data

Drive-D
↓
weka
↓
weka 3.0
↓
Choose
↓
Replace missing value
↓
Apply



Data cleaning:

To Remove Noisdy and unnessary data, null data

Data integration:

where multiple data source may be combined

Data selection;

where data revates to the analaysis task or retrived

Data transformation

where data are transformed into forms approch for mining by performing agrications operation

Data Mining

An essential process where individual intelligent method ar applied in order to extracted

patterns

TO identifies the truly interst pattern representing knowledge based on same interst measure

knowledge presentation:

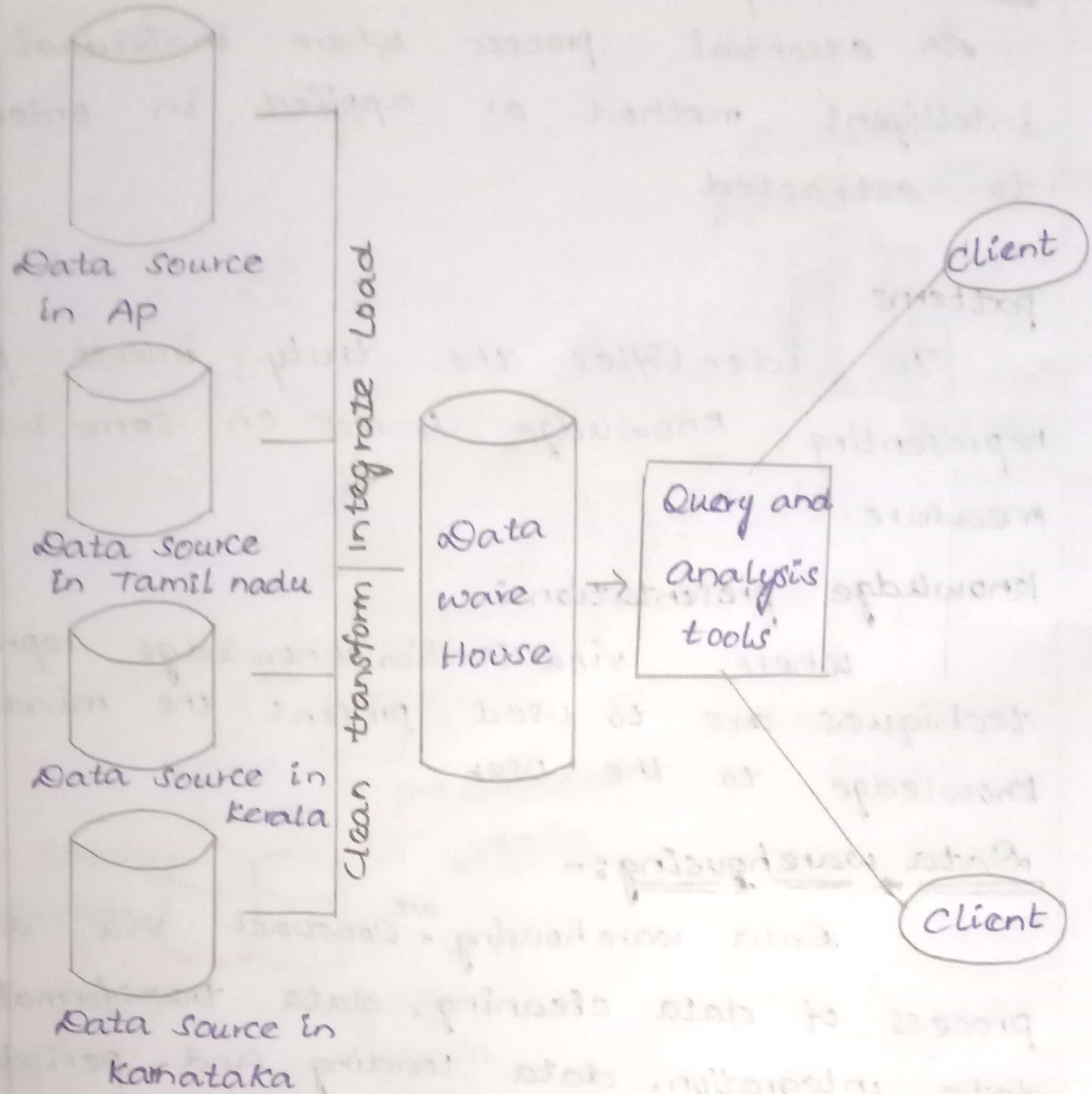
where viewlization knowledge representati techiques are to used present the minded knowledge to the user

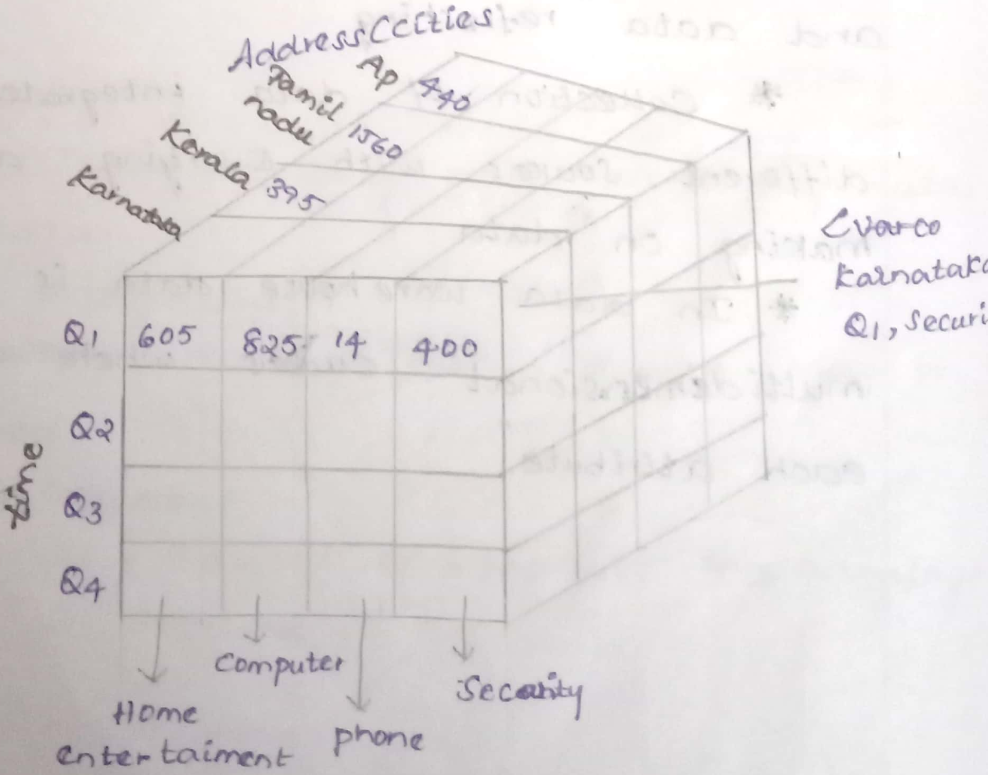
<u>Data warehousing</u>:-

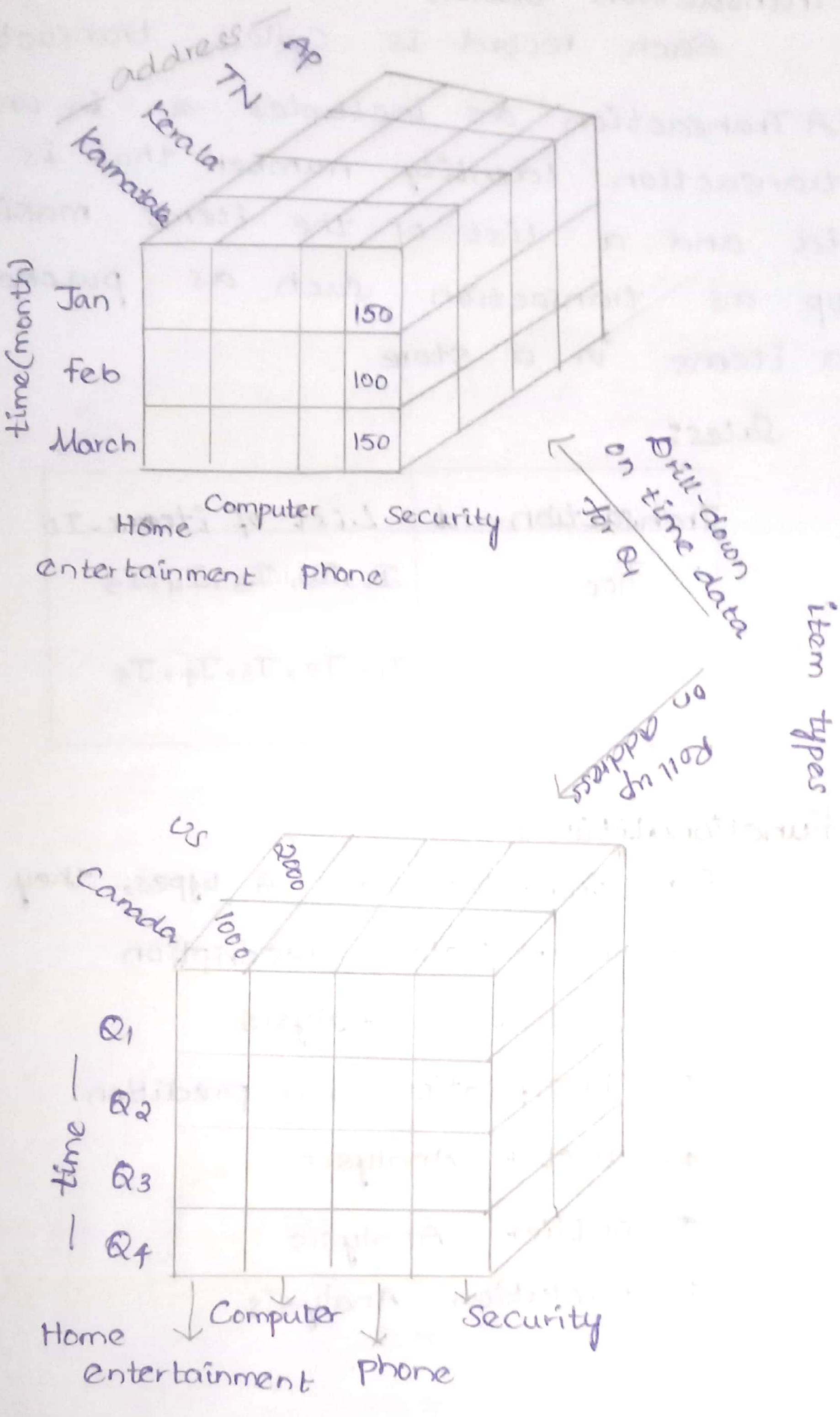
Data warehousing are Constract via a process of data cleaning, data transformation, data integration, data loading and periodic and data refreshing

* Collection of data integrated from different sources with Querying and decision making on data

* In data warehouse data is stored in multidimensional Structure data cubes where each dimensional each attribute


Data source in Ap
Data source in Tamil nadu
Data source in Kerala
Data source in Karnataka
Clean transform Integrate Load
Data ware House
Query and analysis tools
Client
Client


Address(cities)
Ap 440
Tamil nadu 1560
Kerala 395
Karnataka
Q1 605 825 14 400
Q2
Q3
Q4
time
Home entertaiment
Computer
phone
Security



Relation database:- Tables

A database consist of Collection of inter relation data is known as relation data base

identify by a unique key and described by a set of attributes values

A relation database is a Collection of tables each of which is asigned a uniqule name. each table Consists of set of attributes/ Column, and usally stored as large set of Tuples is called ... represents object

Transaction data:

Each record is called transaction data

A Transaction as includes a in unique transaction identity number that is transaction id and a list of the items making up as transaction such as purchased in a items in a store

Sales:

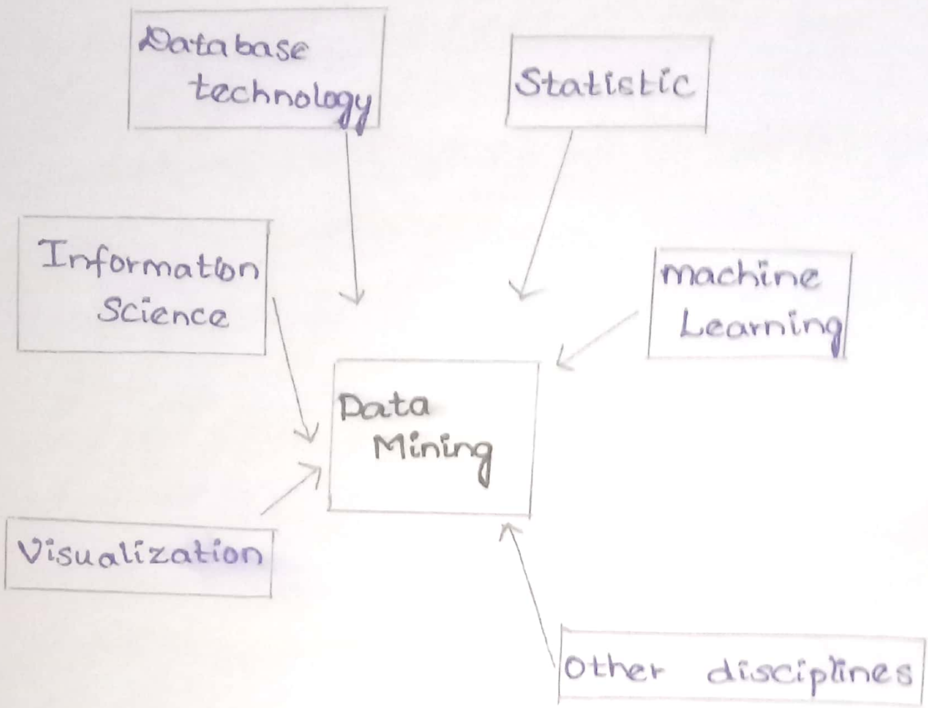
Transaction-id	List of items-Id
T_{100}	I_1, I_2, I_3, I_4, I_5
T_{200}	J_1, J_2, J_3, J_4, J_5

Functionalities:

Functionalities are 6 types, they are

1. Concept / class description
2. Association analysis
3. Classification and predition
4. Cluster Analysis
5. Outlier Analysis
6. Evolution Analysis

5m Classification of data mining system



1. Classification according to the kinds of database mined
2. Knowledge mined
3. techniques mined
4. Application adapted.

* Classification according to the kinds of Database mined.
    * Data base
    * dataware housing
    * transaction

1. Concept / class Description
2. Association analysis
3. Classification and predition
4 Cluster analysis

1. Mining different kinds of knowledge in database

Different uses may be intersted in different kinds of knowledge

Therefore it is neccessary for data mining to cover a wided range of knowledge discovery task, including

* concepts / class description
* Association analysis
* Classification and predition
* cluster Analysis
* outlier Analysis
* evolution Analysis

Eg:-

Supermarket

2. interacting mining at knowledge at multiple levels of abstraction

It is difficult to know exactly what can be discover within a database

The data mining process should be interactive

Interactive mining allow users to focus to search for pattern providing and difining data base request based on written return results

Eg:

college database

3. Incorporation of background knowledge

background knowledge is used to guide the discovery process and express the discovery patterns

Eg: electric bike

4. Data mining Query Language and ad hoc data mining.

Relation Query Language allow user to pose adhoc Queries for data retrive

Data mining Query language need to be developed to allows user. to discribe adhoc data task by speci of the relavent set of data for analysis The domain language of lands of knowledge to be and the conditions and constraints to the enforce mind and the conditions on the discovery patterns

5. Handling noisy and incomplete data

The data cleaning methods are required to handling the noisy and incomplete object wide mining the

When data mining regularity, the objects may confuse the process, the knowledge constracted to over fit of data model

As a result the accuracy of the discover pattern can be poor

6. pattern eluvoution:-

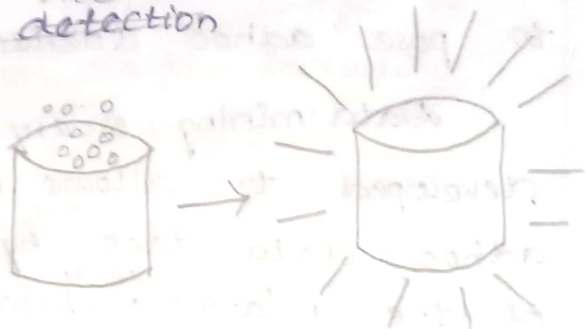
A data mining system can uncover thousands of pattern many of patterns discover may be uninteresting to the given users. representating common knowledge.

Data preprocessing

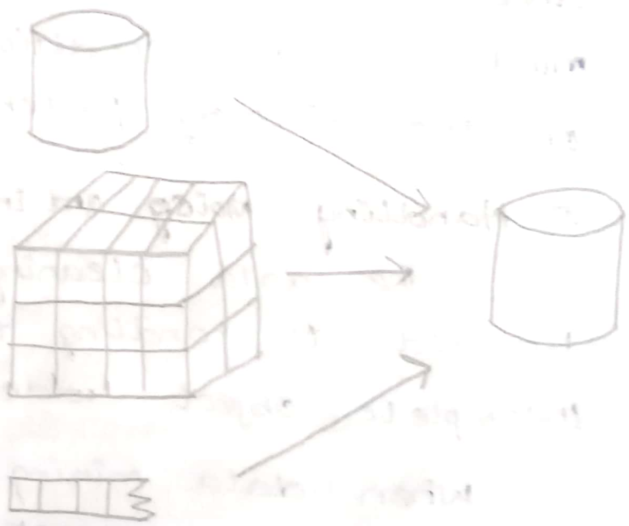
The process of transforming raw data into can understandable formate in data preprocessing

* Data cleaning
* Data integration
* Data transformation
* Data reduction detection

Data Cleaning: ~~process~~ of

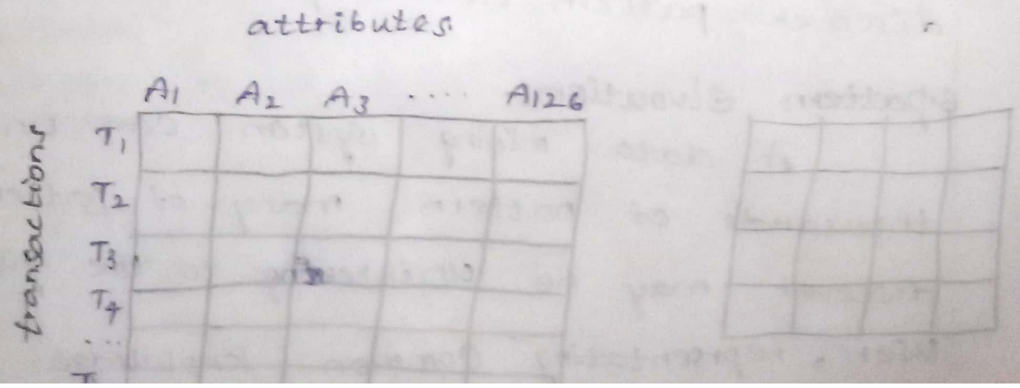


Data Integration



Data transformation -2, 32, 100, 59, 48 -0.02, 0.32, 1.0 0.59, 0.48

Data reduction



process of removel of incorrect, incomplete , inaccurate data also replaces of missing value , noisy data

In data cleaning two methods they are

* missing values
* noisy data

Missing values:-

In place of missing value we can replace not applicable [NA]

Fill in the missing values manually:

This approach is time consuming and may not be feasible given a large data set with many missing values.

~~Use the most probable value to f~~

Ignore the tuple:

This is usually done when the class label is missing

This method is not very effective, unless the tuple contains several attributes with missing values.

Noise data:

Noise is a random error in a measured variable

The following are the data smoothing technique

1. Binning
2. Regression
3. Clustering

Binning:

1. Smoothing by bin means
2. Smoothing by bin medians
3. Smoothing by bin boundaries

because bining methods consult the neighbourhood values they perform there local smoothing

1. Smoothing by bin means:
Each value in a bin is replaced by the mean value of the bin

2. Smoothing by bin medians:
In each bin value is replaced by the bin median

3. Smoothing by bin boundaries:
In min and max value in a given bin are identifies the bin boundaries each bin value is replaced by the closest boundary value

Clustering:-
Outlier may be detect by the clustering where similar values are organise into groups or cluster

Regression:-
Data can be smoothing by fitting the data to a function such as with regression

Linear regression involves find the best line to fit two variable so that one variable is used to credit the other

Data integration
where multiple data sources combin into a single dataset

In data integration two methods are there

1. tight coupling:-

into

2. Loosly Coupling:-

Only a interfacing is created and data is combine through the interface & also access though the interface

Data transformation:-

where data are transformed into forms approch for mining by performing agrication operations.

Data transformation can be involues the following methods

1. Normalization:-

where the attributes are data or scale so as to fall with in a small specified range such as -1.0 to 1.0 (or) 0.0 to 1.0

2. Attribute selection:

where new attributes are constructed and added from the given set of attributes to help the mining process

Generalization:-

where low level data are replaced by higher level concepts through the use of concept hieracial

Eg: city, country

school - college

younger age - older age

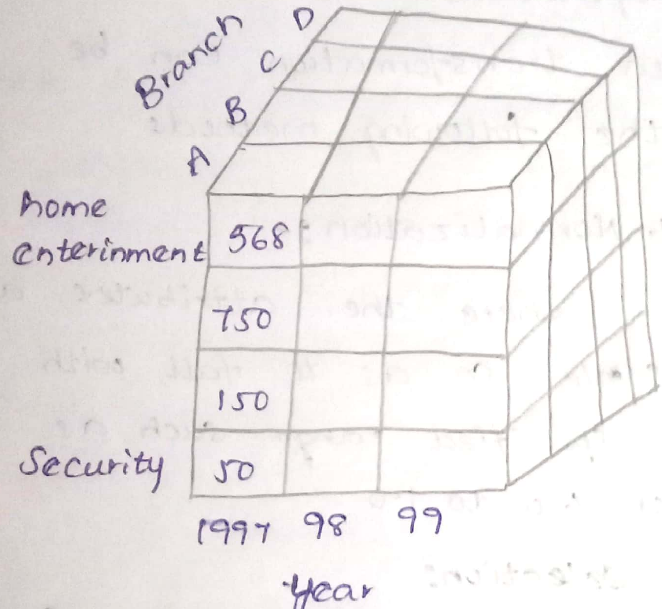
Data reduction:-

Data reduction techiques can be applied to obtain are reduce representation ... that is much smaller in

At get cloase maintain the integrity of the original data

Data cube agrication:-

where agrication operation are applied to the data in the construction of data cube



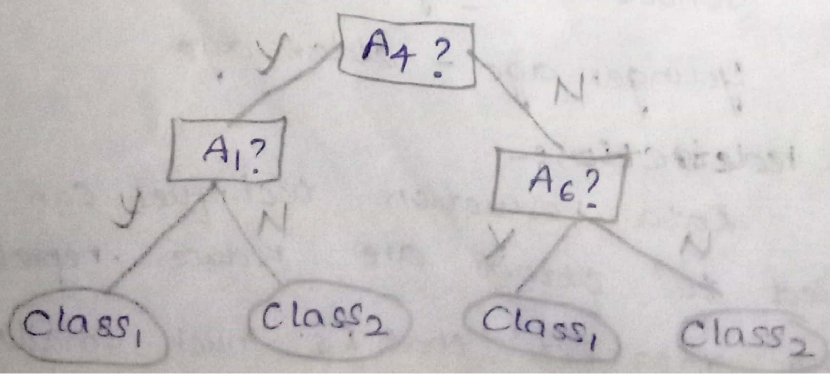
where with rellalant, weekly revalant, weak redateant attributes are dimensiona may be detected and removed

Initial attributes set:

$\{A_1, A_2, A_3, A_4, A_5, A_6\}$

Reduced attributes set:

$\{A_1, A_4, A_6\}$



premitives

we can specificer the the data mining in the from data mining Query,

In this Query is the input to the System

A data mining Query is define in terms of data mining task premitives

this premitive allows to Communicate in an interactive manner with the data mining system

Task relavent data:

1. the first premitive in the Specification of data on which mining is to be perfomed

2. A user is intersted in only a Subset of database

3. In relation database the set of task relavent data can be collected via relation Query involves are select, join, update, and aggregation

The Kind of knowledge to be mined:-

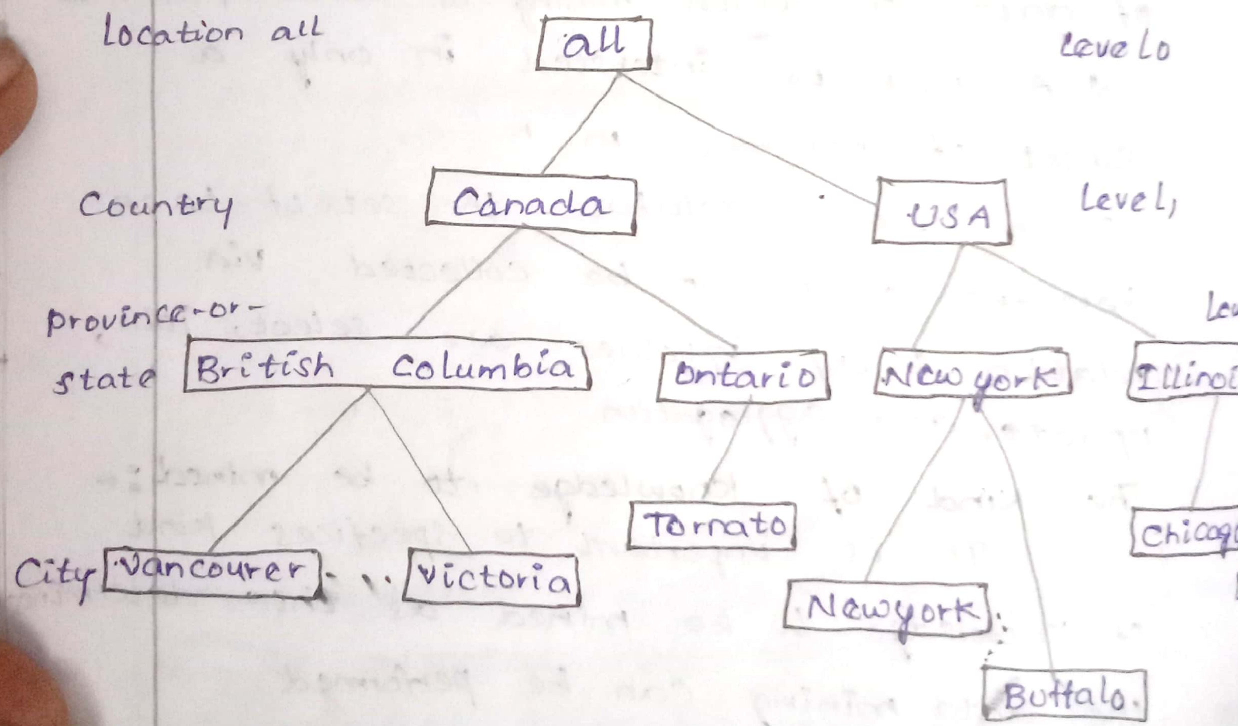
It is important to specfices kind of knowledge to be mined as this determines the data mining can be performed

1. class description/ concept
2. data characterization
3. data Comparasation
4. Associated analysis
5. Classication and predition
6. Cluster analysis
7. Outlier analysis
8. Evluation analysis.

Background Knowledge :-
(Concept hiercial)

Background knowledge is information about the domain to be mined that can be useful in the discovery the process we focus our attention on a simple form of background knowledge known as Concept hiercary

Concept hiercary allow the discovery of knowledge at multiple levels of abstraction



Concept hiercary is represented as set of notes organisation in a set of trees

In a tree, where each node, itself represents a Concept

In Specification of task relavant data and the kind of knowledge to be mind may reduce the number of patterns generated, a data mining process may still generate a large no. of patterns. only small fraction of this pattern will be actually be interest to the given user

presentation and view of discovered patterns:

The uses of concept hiercary place a important role in identity the user to view discovery pattern.

Mining with concept hiercary allows the representation of discovery ~~high~~ in knowledge highlevel concept which may be more understandable to user then rules expressed in premitive data such as functionlities dependency rules or integrity constant.

Rules:

age (x, "young") and income (x, "high")
=> class (x, "A")

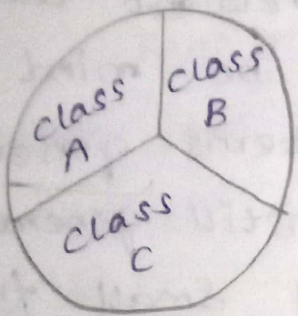
age (x, "young") and income (x, "low") => class(x, "B")

age (x, "old") => class (x, "c")

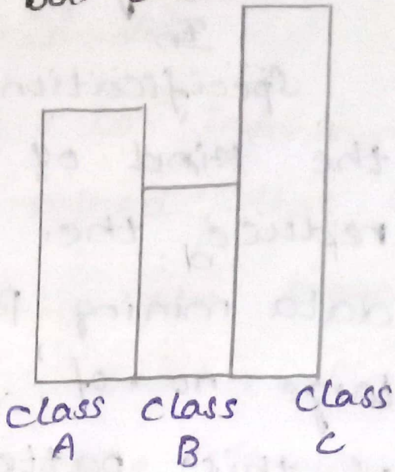
Table:

age	income	class	count
young	high	A	1,402
young	low	B	1,036
old	high	C	786
old	[illegible]	C	1374

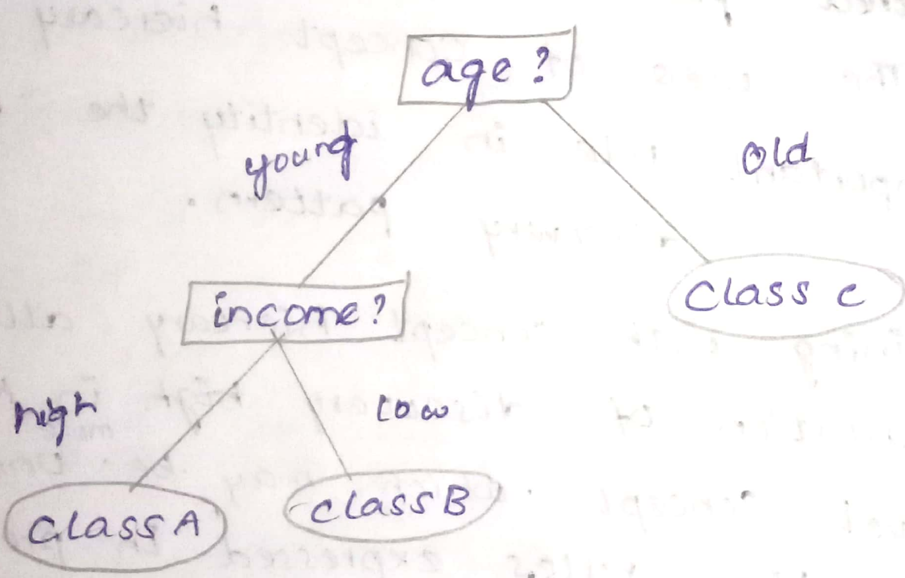
pie chart:



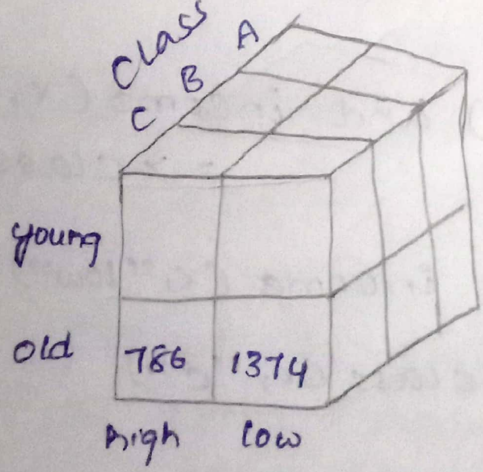
Bar chart:

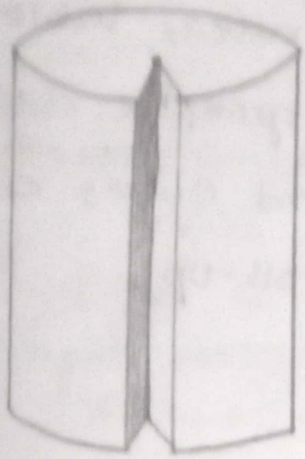


Decision tree:



Data cub:

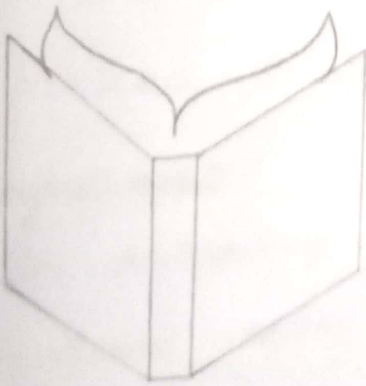




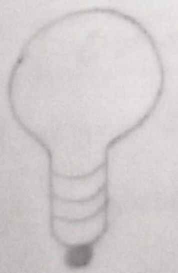
* Task-relavent data, database or
Data warehouse name DB tables
or data warehouse cubes
Conditions for data selection
Relevant attributes or dimensions
data grouping criteria

Kind of knowledges:

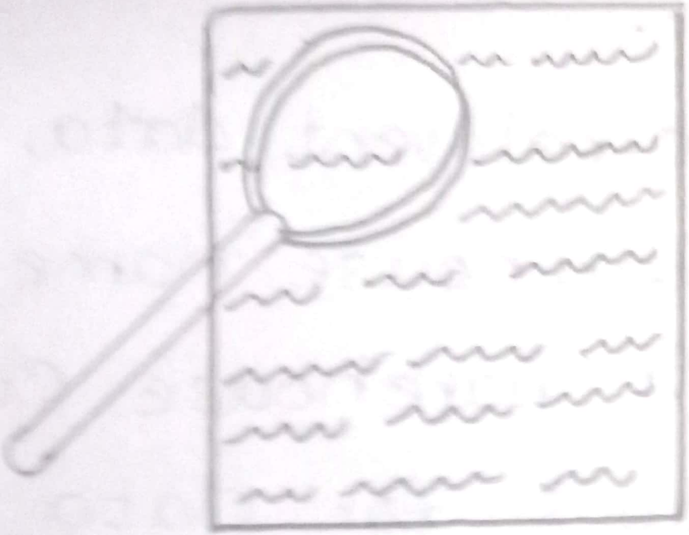
1. Data characterization
2. Data discrimination
3. Classification & prediction
4. Cluster analysis
5. Association analysis



Background knowledge
Concept hierarchies user
beliefs about relationships
in the data



pattern interestingness
measures simplicity
Certainty (eg: confidence)
utility (eg: support)
Novelty



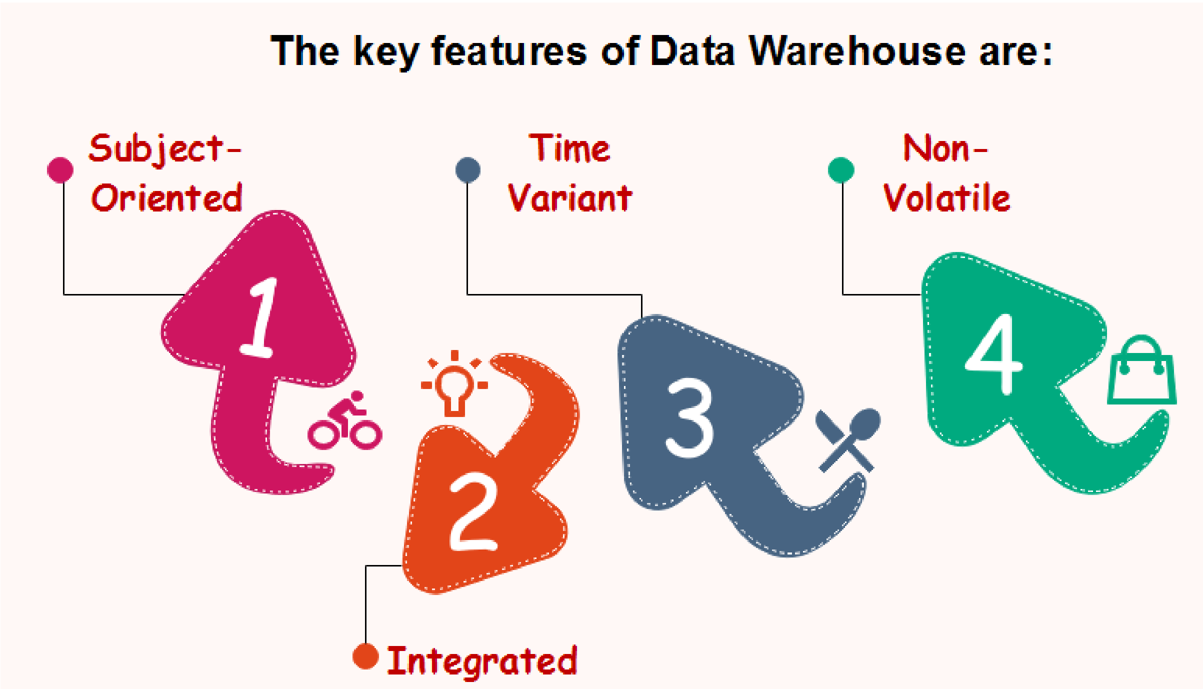
visualization of discovered pattern rules, tables, report Charts, graphs decision trees and Cubes drill down and roll-up

What is a Data Warehouse?

- A Data Warehouse (DW) is a relational database that is designed for query and analysis rather than transaction processing. It includes historical data derived from transaction data from single and multiple sources.
- A Data Warehouse provides integrated, enterprise-wide, historical data and focuses on providing support for decision-makers for data modelling and analysis.
- A Data Warehouse is a group of data specific to the entire organization, not only to a particular group of users.

"Data Warehouse is a subject-oriented, integrated, and time-variant collection of information or data in support of management's decisions."

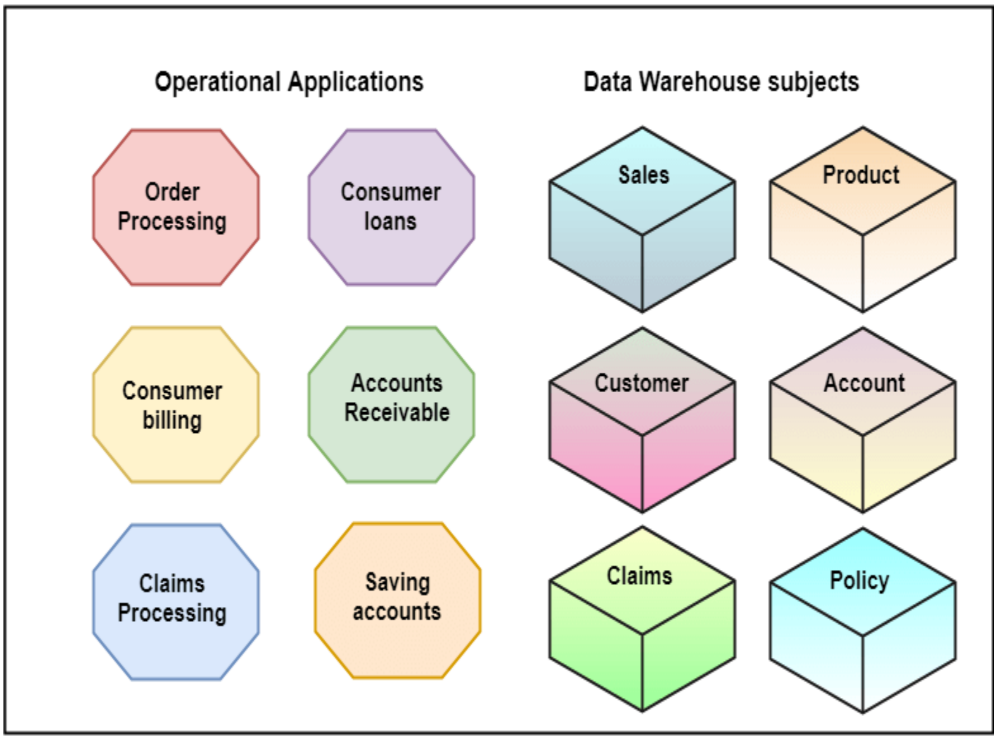
Features of Data Warehouse



Subject-Oriented

- A data warehouse target on the modelling and analysis of data for decision-makers.
- Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations.
- This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.

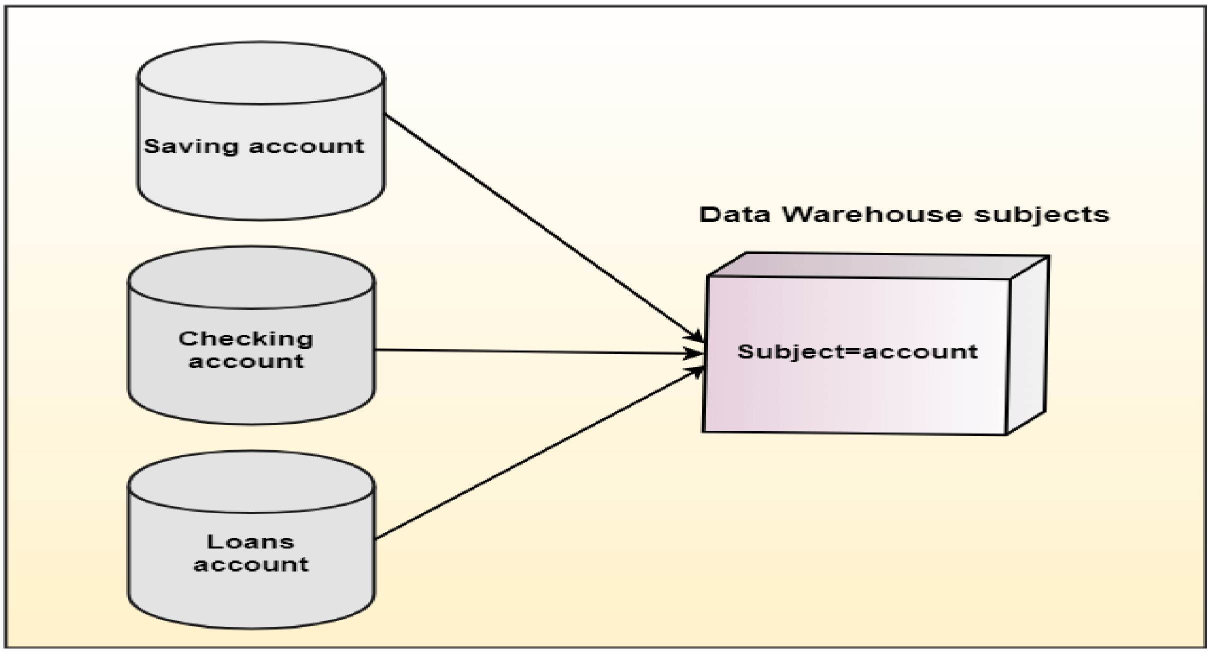
Data Warehouse is Subject-Oriented



Integrated

- A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records.
- It requires performing data cleaning and integration during data warehousing to ensure consistency in naming conventions, attributes types, etc., among different data sources.

Data Warehouse is Integrated

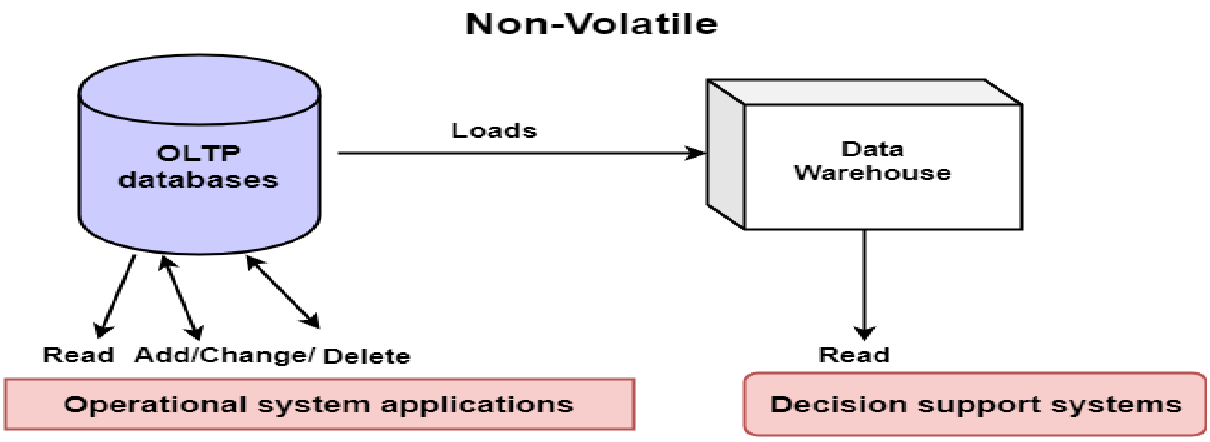


Time-Variant

- Historical information is kept in a data warehouse.
- For example, one can retrieve files from 3 months, 6 months, 12 months, or even previous data from a data warehouse.
- These variations with a transactions system, where often only the most current file is kept.

Non-Volatile

- The data warehouse is a physically separate data storage, which is transformed from the source operational RDBMS.
- The operational updates of data do not occur in the data warehouse, i.e., update, insert, and delete operations are not performed.
- It usually requires only two procedures in data accessing: Initial loading of data and access to data.
- Therefore, the DW does not require transaction processing, recovery, and concurrency capabilities, which allows for substantial speedup of data retrieval.
- Non-Volatile defines that once entered into the warehouse, and data should not change



Multi-Dimensional Data Model

- A multidimensional model views data in the form of a data-cube. A data cube enables data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.
- The dimensions are the perspectives or entities concerning which an organization keeps records.
- For example, a shop may create a sales data warehouse to keep records of the store's sales for the dimension time, item, and location.
- These dimensions allow the save to keep track of things, for example, monthly sales of items and the locations at which the items were sold.
- Each dimension has a table related to it, called a dimensional table, which describes the dimension further. For example, a dimensional table for an item may contain the attributes item name, brand, and type.
- A multidimensional data model is organized around a central theme, for example, sales.
- This theme is represented by a fact table. Facts are numerical measures. The fact table contains the names of the facts or measures of the related dimensional tables.

In the 2-D representation, we will look at the All-Electronics sales data for items sold per quarter in the city of Vancouver. The measured display in dollars sold (in thousands).

2-D view of Sales Data

location ="Vancouver"				
	item (type)			
time (quarter)	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q3	927	1038	38	580

3-Dimensional Cuboids

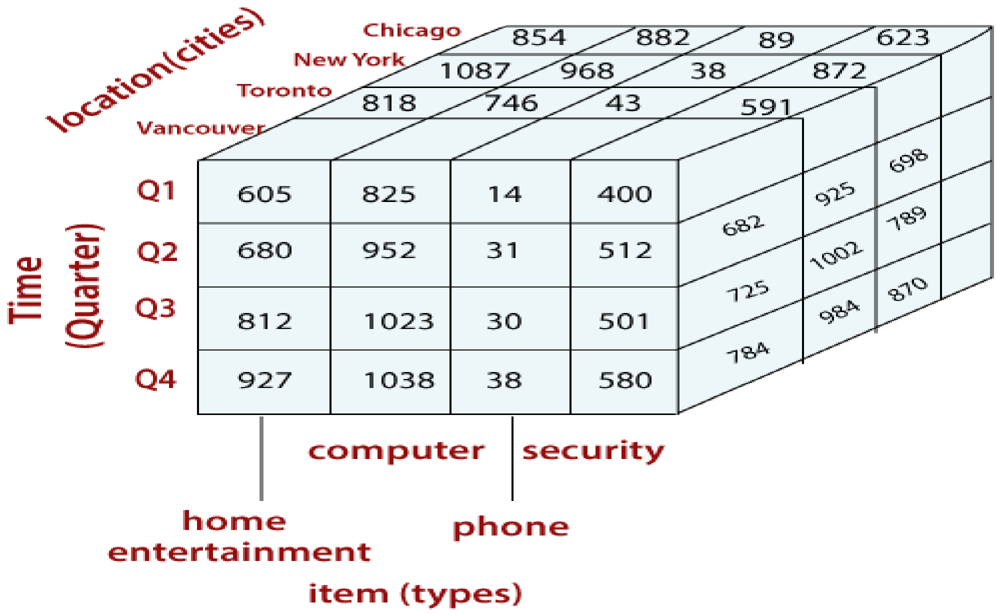
- Now suppose we would like to view the sales data with a third dimension.
- For example, suppose we would like to view the data according to time, item as well as the location for the cities Chicago, New York, Toronto, and Vancouver.
- The measured display in dollars sold (in thousands). These 3-D data are shown in the table.

3-D view of Sales Data

	location ="Chicago"				location ="New York"				location ="Toronto"			
	item				item				item			
time	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784

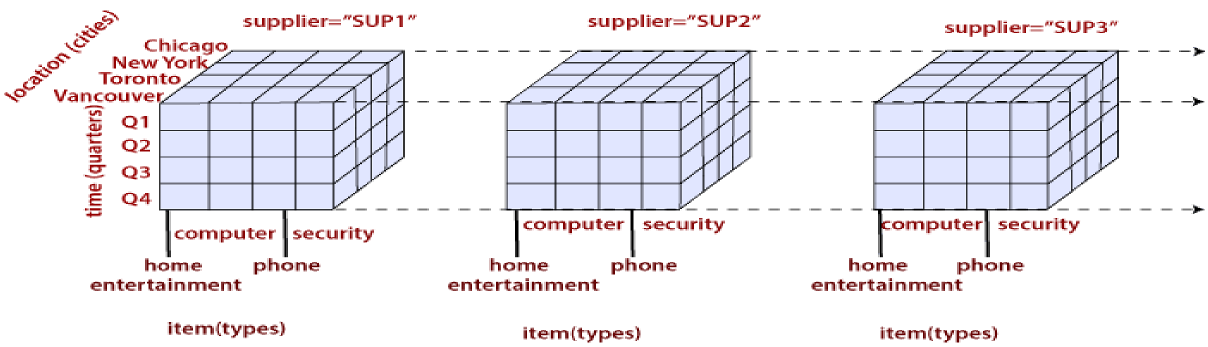
we may represent the same data in the form of 3-D data cubes, as shown in fig:

3-D Data Cube



4-D cuboid

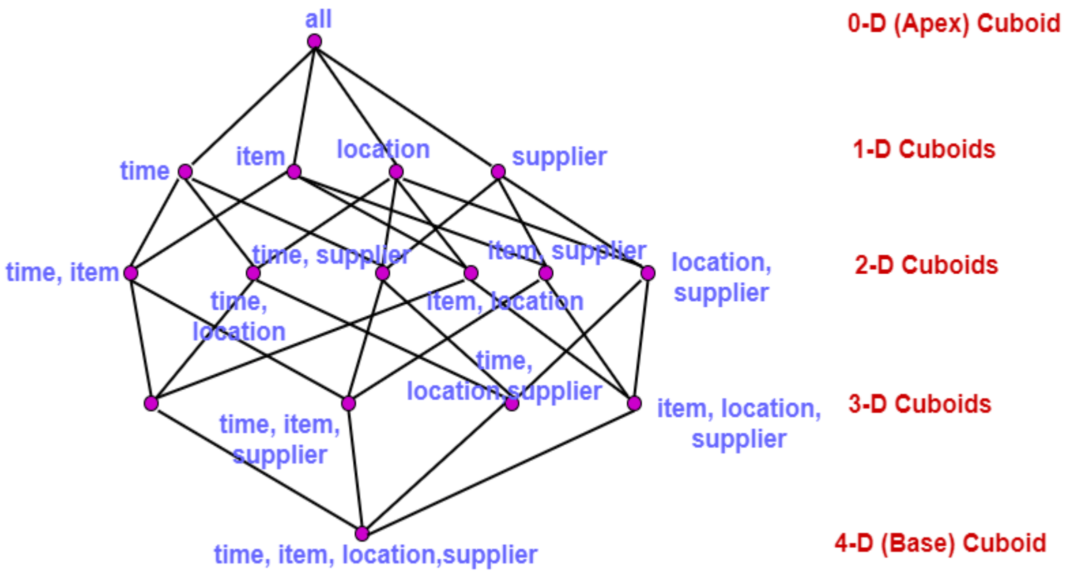
Let us suppose that we would like to view our sales data with an additional fourth dimension, such as a supplier. Viewing things in 4-D becomes tricky.



A 4-D data cube representation of sales data, according to the dimensions time, item, location, and supplier. The measure displayed is dollars sold (in thousands).

The topmost 0-D cuboid, which holds the highest level of summarization, is known as the apex cuboid. In this example, this is the total sales, or dollars sold, summarized over all four dimensions. The apex cuboid is typically denoted by **All**.

The lattice of cuboid forms a data cube. The lattice of cuboids creating 4-D data cubes for the dimension time, item, location, and supplier. Each cuboid represents a different degree of summarization.



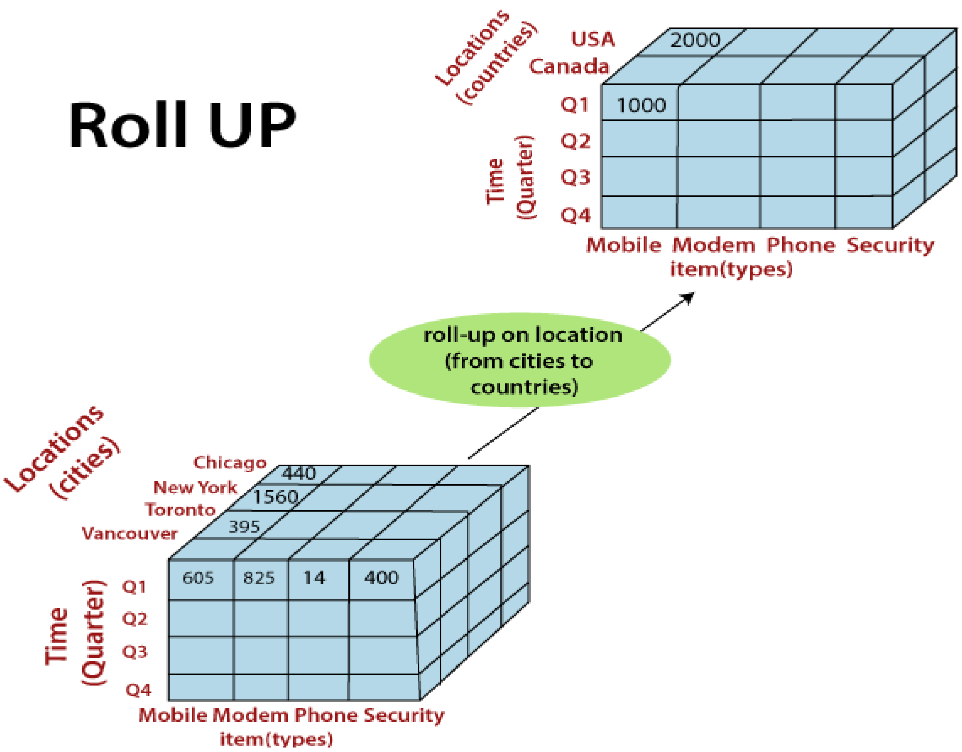
OLAP Operations in the Multidimensional Data Model

- In the multidimensional model, the records are organized into various dimensions, and each dimension includes multiple levels of abstraction described by concept hierarchies.
- This organization support users with the flexibility to view data from various perspectives.
- A number of OLAP data cube operation exist to demonstrate these different views, allowing interactive queries and search of the record at hand.
- Hence, OLAP supports a user-friendly environment for interactive data analysis.

Roll-Up

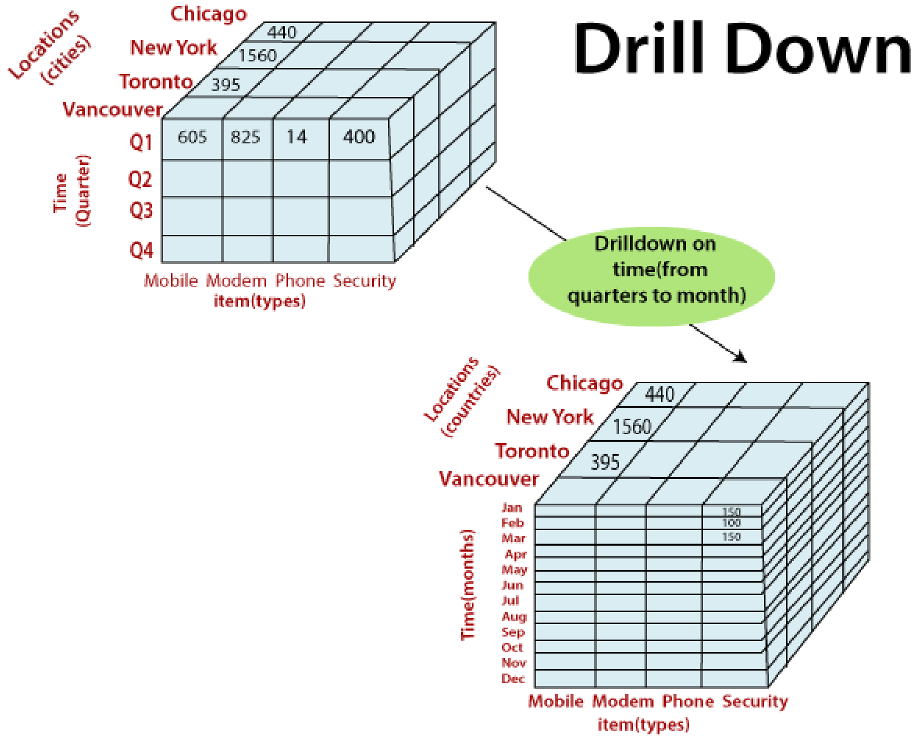
- The roll-up operation (also known as drill-up or aggregation operation) performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction.
- Roll-up is like zooming-out on the data cubes. Figure shows the result of roll-up operations performed on the dimension location.

- The hierarchy for the location is defined as the Order Street, city, province, or state, country. The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country.
- When a roll-up is performed by dimensions reduction, one or more dimensions are removed from the cube.
- For example, consider a sales data cube having two dimensions, location and time. Roll-up may be performed by removing, the time dimensions, appearing in an aggregation of the total sales by location, relatively than by location and by time.



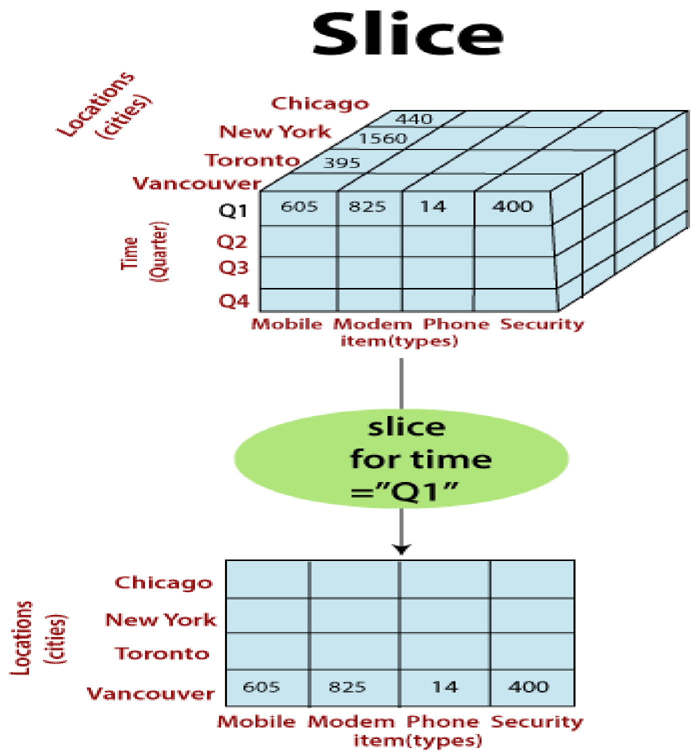
Drill-Down

- The drill-down operation **(also called roll-down)** is the reverse operation of **roll-up**.
- Drill-down is like **zooming-in** on the data cube. It navigates from less detailed record to more detailed data.
- Drill-down can be performed by either **stepping down** a concept hierarchy for a dimension or adding additional dimensions.
- Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year.
- Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.
- Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube.
- For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.



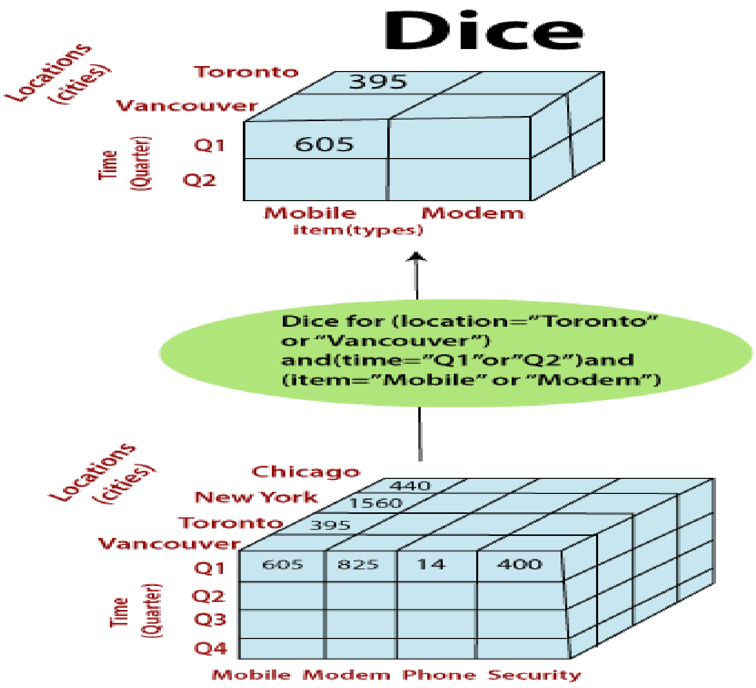
Slice

➢ The Slice operations perform a selection on one dimension of the given cube, thus resulting in a sub cube.



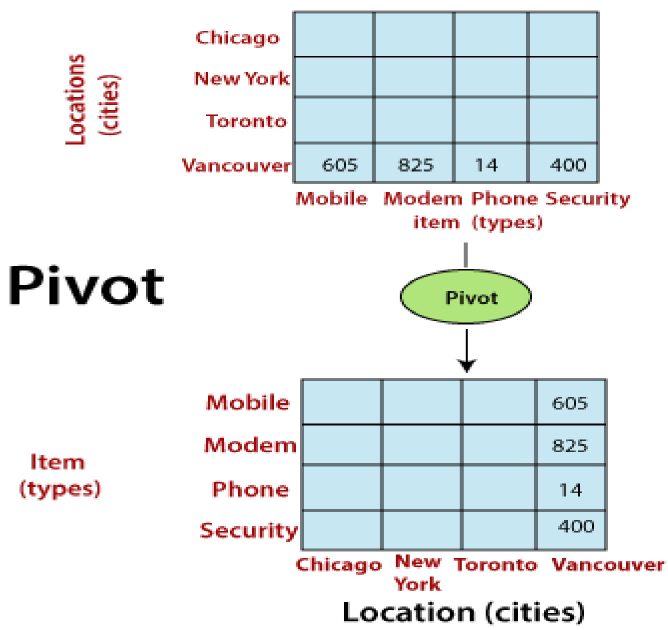
Dice

- ➢ The dice operation describes a subcube by operating a selection on two or more dimension.



Pivot

- ➢ The pivot operation is also called a rotation. Pivot is a visualization operation which rotates the data axes in view to provide an alternative presentation of the data.
- ➢ It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.



Data Warehouse Architecture

Data Warehouses usually have a three-level (tier) architecture that includes:

- Bottom Tier (Data Warehouse Server)
- Middle Tier (OLAP Server)
- Top Tier (Front end Tools).

Bottom Tier

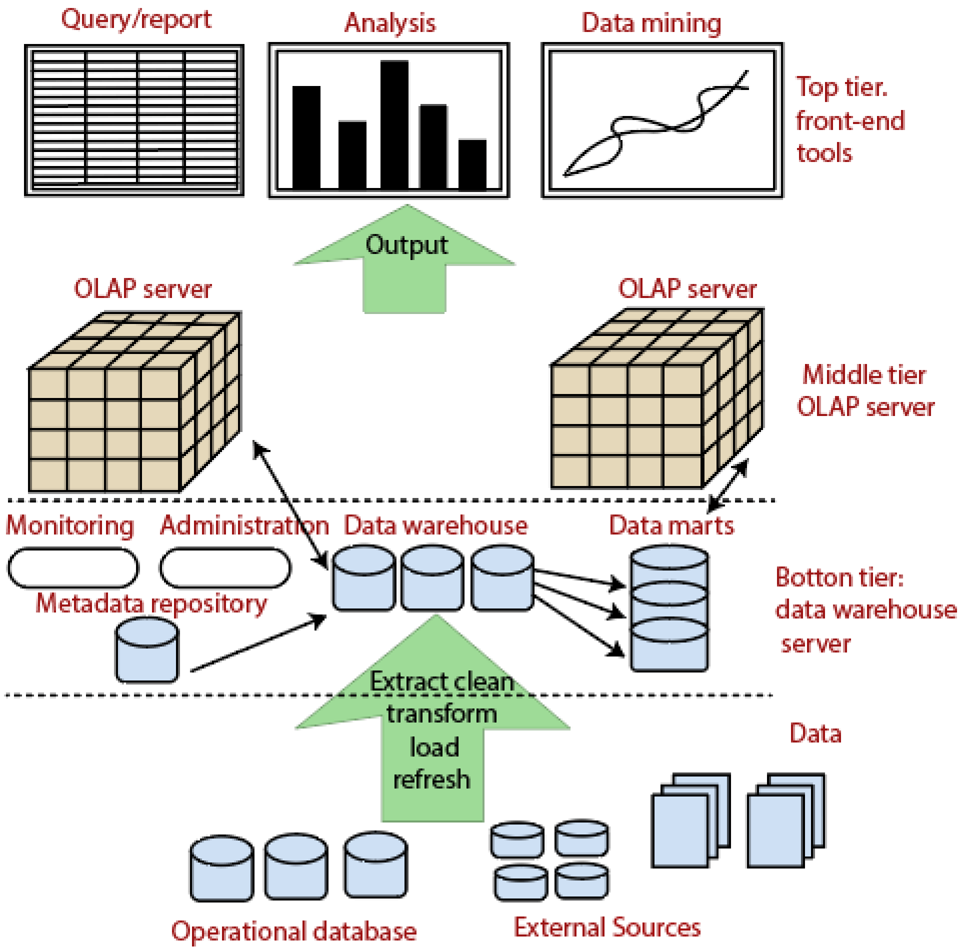
- A bottom-tier that consists of the Data Warehouse server, which is almost always an RDBMS. It may include several specialized data marts and a metadata repository.
- Data from operational databases and external sources (such as user profile data provided by external consultants) are extracted using application program interfaces called a gateway.
- A gateway is provided by the underlying DBMS and allows customer programs to generate SQL code to be executed at a server.
- Examples of gateways contain ODBC (Open Database Connection) and OLE-DB (Open-Linking and Embedding for Databases), by Microsoft, and JDBC (Java Database Connection).

Middle Tier

- A middle-tier which consists of an OLAP server for fast querying of the data warehouse.
- The OLAP server is implemented using either
- A Relational OLAP (ROLAP) model, i.e., an extended relational DBMS that maps functions on multidimensional data to standard relational operations.
- A Multidimensional OLAP (MOLAP) model, i.e., a particular purpose server that directly implements multidimensional information and operations.

Top Tier

- A top-tier that contains front-end tools for displaying results provided by OLAP, as well as additional tools for data mining of the OLAP-generated data.



Three-Tier Data Warehouse Architecture

- A description of the DW structure, including the warehouse schema, dimension, hierarchies, data mart locations, and contents, etc.
- Operational metadata, which usually describes the currency level of the stored data, i.e., active, archived or purged, and warehouse monitoring information, i.e., usage statistics, error reports, audit, etc.
- System performance data, which includes indices, used to improve data access and retrieval performance.
- Information about the mapping from operational databases, which provides source RDBMSs and their contents, cleaning and transformation rules, etc.

Data Warehouse Implementation

Data warehouses contains huge volumes of data. OLAP servers demanded that decisions support queries be answered in the order of seconds.

It is crucial for data warehouse systems to support highly efficient cube computation techniques access methods and query processing techniques.

We will see some methods for the efficient implementation of data warehouse systems. They are

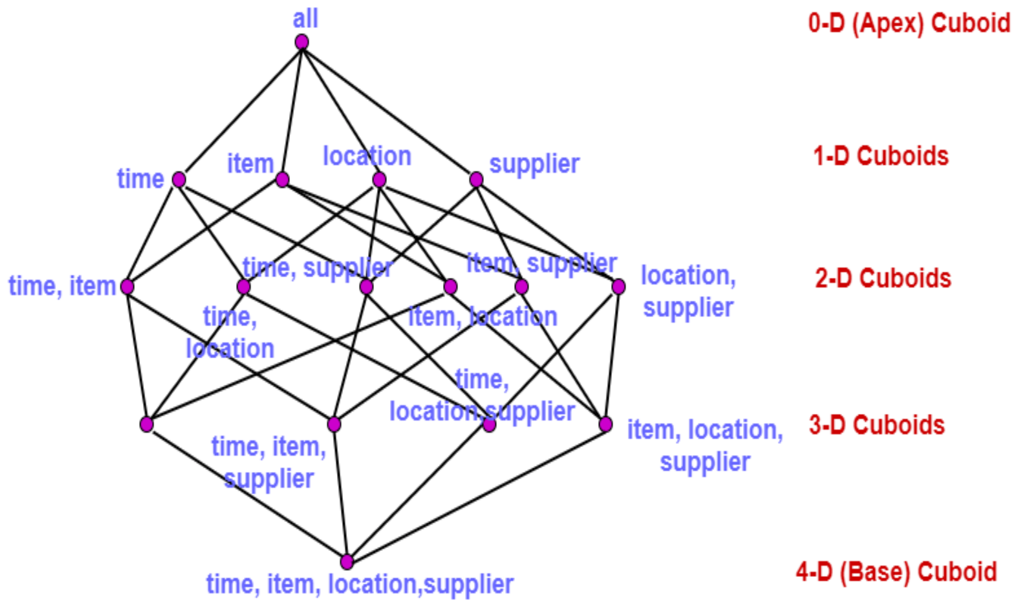
1.Efficient computation of Data Cubes

It contains 2 types those are

- Compute cube operator
- Materialization of data cube

Compute cube operator

- Data cube can be viewed as a lattice of cuboids.
- The bottom-most cuboid is the base cuboid.
- The top-most cuboid (apex)contains only one cell.
- The total number of cuboid or group by's can be computed for data cube is (2 power n).
- Example: If dimensions given as item, city and year, (2 power 4 =16)
- Compute cube operator computes aggregates overall subsets of dimensions specified in the operation.
- The possible group by's are {(time, item, location, supplier)
  (time, item, location), (item, location, supplier), (location, supplier, time),
  (time, item, supplier), (time, item), (item, location), (time), (item), (location), (supplier),()}
- Let consider the diagram given below :



Materialization of data cube

There are three choices for data cube materialization.

- No materialization
- Full materialization
- Partial materialization

No materialization :

- Do not pre-compute any of the "no base" cuboids .This leads to computing expensive multidimensions aggregates on the fly which can be extremely slow.

Full materialization :

- Pre-compute all of the cuboids .The resulting lattice of computed cuboids is referred to as the dull case .This choice typically requires huge amount of memory space.

Partial materialization :

- The third choice-presents an interesting trade-of between storage space and response time.
- The partial materialization of cuboids should consider three factors:
  1. Identify the subset of cuboids to materialize.
  2. Exploit the materialized cuboids during query processing.
  3. Efficiency updates the materialize cuboids during bad refresh.

2.Indexing OLAP Data:

- To facilitate efficient data accessing, most data warehouse systems support index structures and materialized views (using cuboids).
- Indexing can derived into 2 types.
  - Bitmap indexing
  - Join indexing

Bitmap Indexing:

- This method is popular in OLAP products because it allows quick searching in data cubes.
- The bitmap index is an alternative representation of the record _id (RID) list.
- In this bitmap index foe s given attribute, there is a distinct bit vector, Bv, for each value v in the domain of the attribute.
- If the domain of a given attribute consist of n values, then n bits are needed for each entry in the bitmap index.

Let consider the example for bitmap indexing

Base Table

RID	ITEM	CITY
R1	H	V
R2	C	V
R3	P	V
R4	S	V
R5	H	T
R6	C	T
R7	P	T
R8	S	T

item bitmap index table

RID	H	C	P	S
R1	1	0	0	0
R2	0	1	0	0
R3	0	0	1	0
R4	0	0	0	1
R5	1	0	0	0
R6	0	1	0	0
R7	0	0	1	0
R8	0	0	0	1

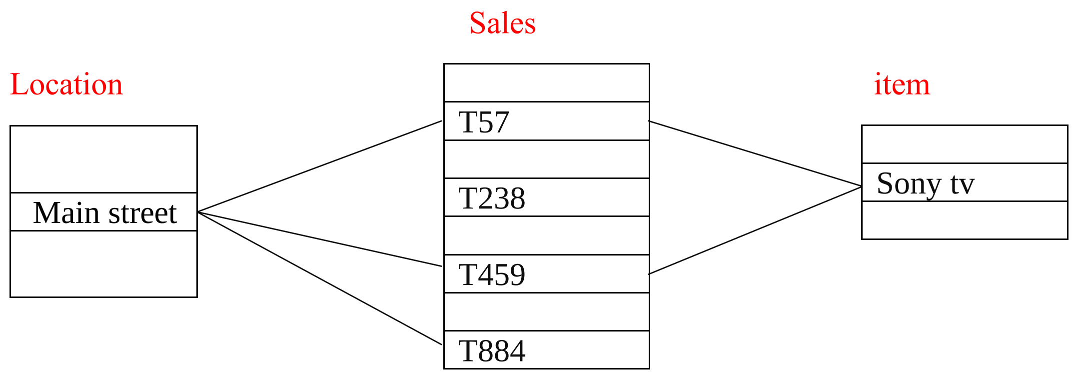
city bitmap index table

RID	V	T
R1	1	0
R2	1	0
R3	1	0
R4	1	0
R5	0	1
R6	0	1
R7	0	1
R8	0	1

Note : H for “home entertainment,” C for Computer,” P for Phone,” Sfor Security,” V for” Vancouver,” T for “Toronto.”

Join Indexing:

- Join indexing registers the joinable rows of two relations from a relational database.
- For example , if two relations R(RID,A)and s(B,SID) join on the attributes A and B, then the join index record contains the pair (RID,SID),where RID and SID are record identifiers from the R and S relations , respectively
- Hence, the join index records can identify joinable tuples without performing costly join operations.
- Join indexing maintains relationships between attribute values of a dimension (e.g., within a dimension table) and the corresponding rows in the fact table.
- Let consider the example for Join indexing



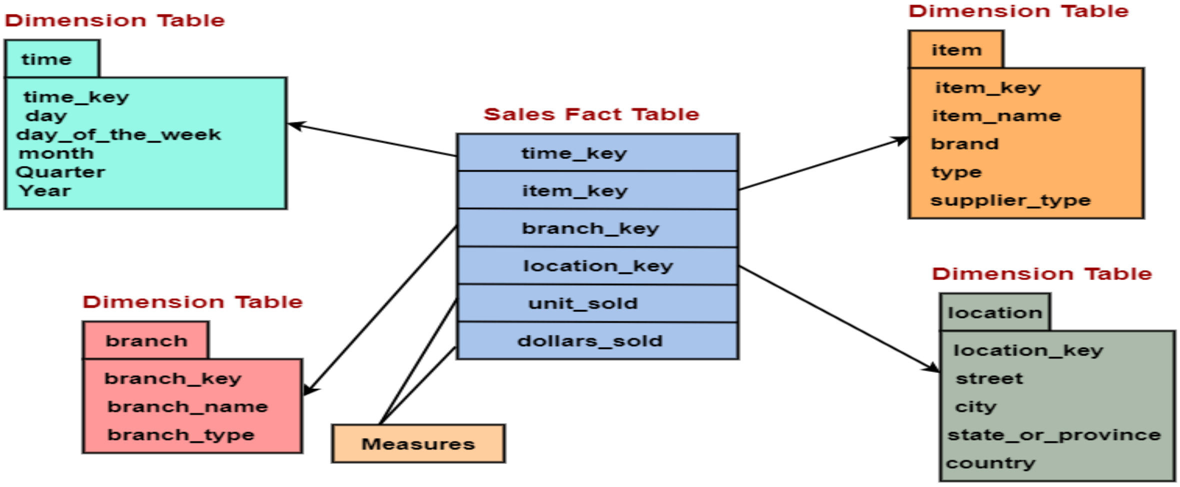
SCHEMAS FOR MULTIDIMENSIONAL DATABASES

- The Entity–Relationship data model is commonly used in the design of Relational Databases. Where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for online transaction processing.
- A data warehouse, however, requires a concise, subject-oriented schema that facilitates online data analysis.
- The most popular data model for a data warehouse is a multidimensional model. Which can exist in the form of a Star schema, a Snowflake schema, or a Fact constellation schema.

Star schema:

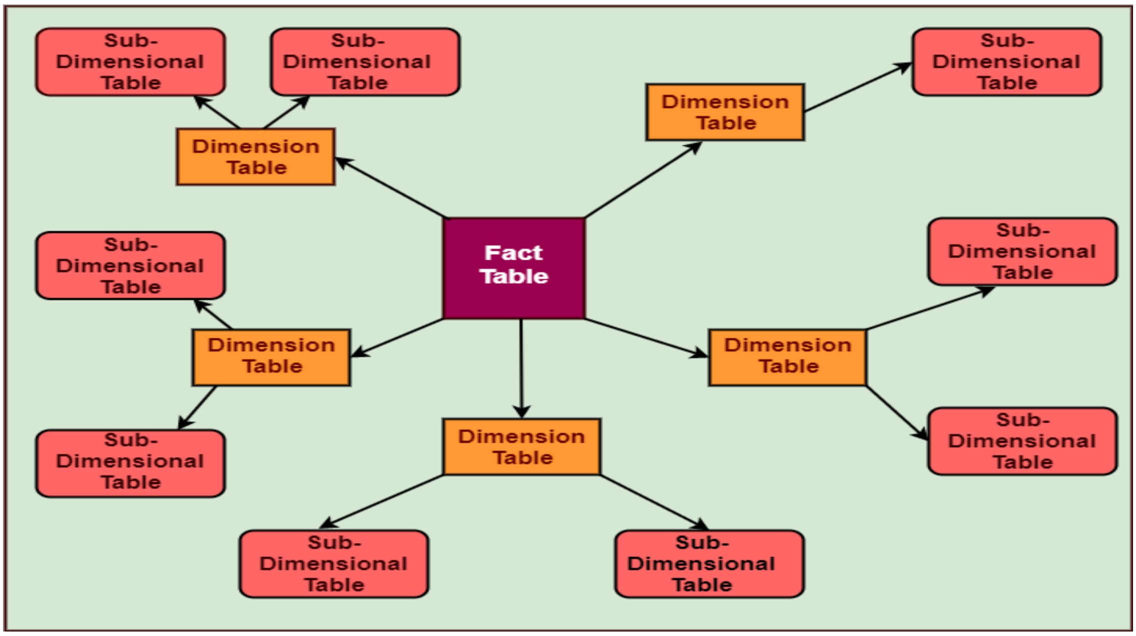
- The most common modelling paradigm is the Star schema, in which the data warehouse contains:
- a large central table (fact table) containing the bulk of the data, with no redundancy.
- A set of smaller attendant tables (dimension tables), one for each dimension. The schema graph resembles a star burst, with the dimension tables displayed in a radial pattern around the central fact table.
- Example: Sales are considered along four dimensions namely time, item, branch and location.

Star schema of Sales data warehouse



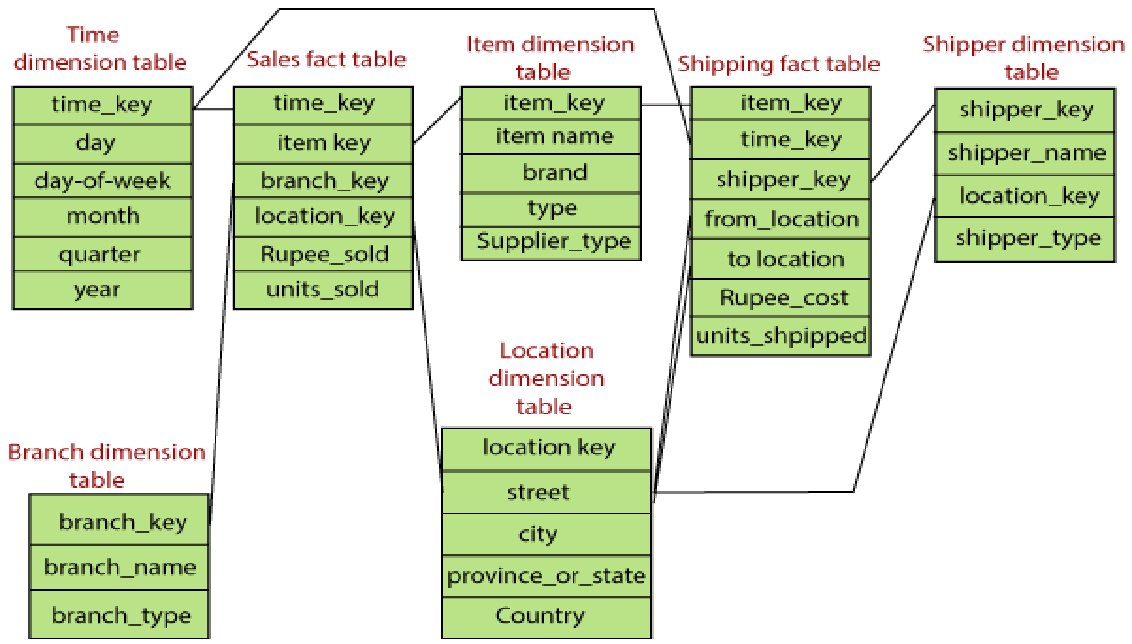
Snowflake Schema:

- The Snowflake schema is a variant of the star schema model, where some dimension tables are normalized.
- The resulting schema graph forms a shape similar to a snowflake. The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies.
- The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables.
- The item dimension table now contains the attributes item_ key, item _name, brand, type, and supplier key, supplier_ key is linked to the supplier dimension table, containing supplier key, and supplier _type information.
- similarly, the single dimension table located for the star schema can be normalized into two new tables: location and city. The city ss_key in the new location table links to the city dimension.



Fact constellation:

- Sophisticated applications may require multiple fact tables to share dimensions tables.
- This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.
- This schema specifies two fact table, sales and shipping. The sales table definition is identical to that of the star schema.
- The shipping table has five dimensions, item_ key, time_ key, shipper_ key, from _location, and to location, and two measures: dollars_ cost and units_shipped.
- A fact constellation schema allows dimension tables to be shared between fact table.



From data warehousing to data mining

Data warehouse usage:

There are three kind of data warehouse application:

- Information processing
- Analytical processing
- Data mining

Information processing:

Supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts, or graphs. A current trend in data warehouse information processing is to construct low_cost web_based accessing tools that are then integrated with web browsers.

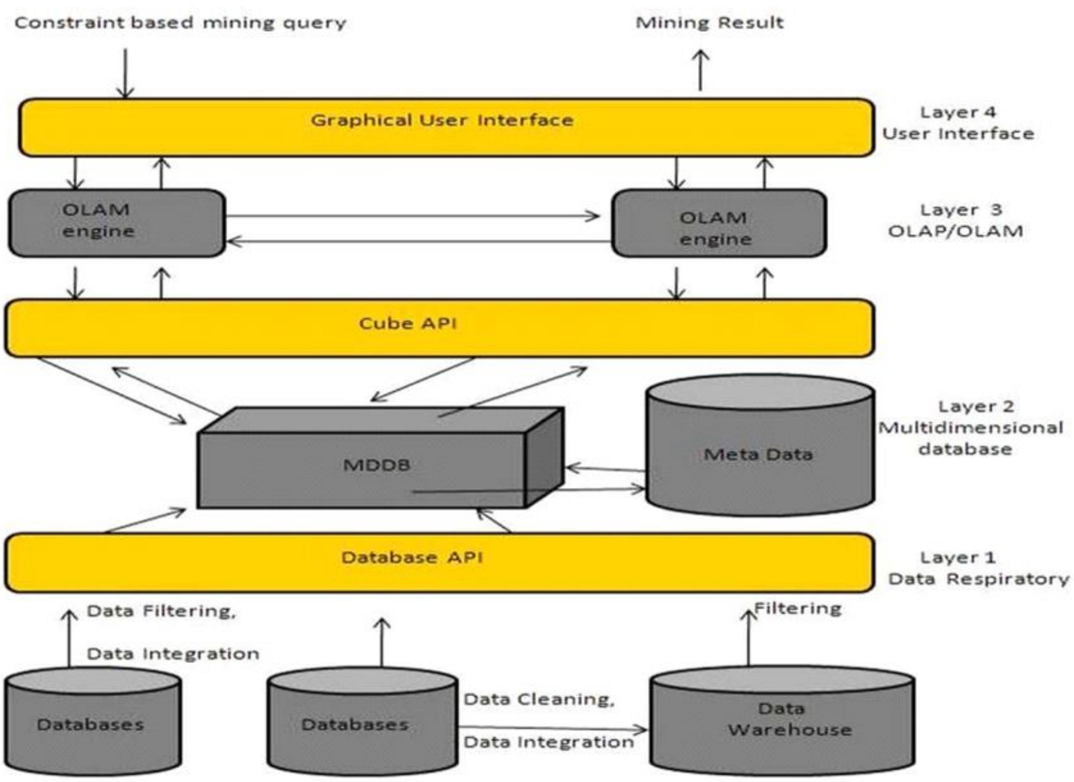
Analytical processing:

- Supports basic OLAP operations, including slice and dice, drill-down, roll-up, and pivoting.
- It generally operates on historical data in both summarized and detailed forms.
- The major OLAP over information processing is the multidimensional data analysis of data warehouse.

Data mining:

Supports knowledge discovery by finding hidden patterns and as sociations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

From OLAP to OLAM:



unit-3

Mining frequent patterns Association and Correlation.

*) Transactional database :- (TDB)

TDB Consists of set of transactions in which each transaction is associated with each item.

Eg:- TDB = {T_1, T_2, T_3 .....}

T_1 = {R, D, S}

T_2 = {m, S, J}

T_3 = {R, O, C}

TDB	set of items
T_1	Rice, Dall, suger
T_2	milk, suger, Jam
T_3	Rice, oil, cake
.	
.	
.	

Mining frequent items are also types they are

*) Mining frequent pattern with Candidate

*) Mining frequent pattern without Candidate

*) Item set :-

A set of item is said to be an item set it is denoted by 'I'.

Items are lowercase letters $I = \{i_1, i_2, i_3 \ldots i_k\}$

Eg:- i_1 = {Rice, Dall, suger}

K-item set :-

An item set consisting of k-no.of items set is called k-item set it is denoted by "IK"

$I_k = \{i_1, i_2 \ldots i_k\}$

imp

*) frequent item set :-

An item set 'i' is said to frequent item set if its satisfies pre-specified minimum support threshold value.

*) The set of frequent k-item set is denoted by skript 'Lk'.

$Lk = \{l_1, l_2, l_3\}$

Association rule :-

An association rule is an implecation of the form $A \rightarrow B$.

where $A \subset i$, $B \subset i$ & $A \cap B = \phi$

with support 's' and confidence 'c'

Eg :-

$D = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8\}$

I = {milk, Coke, pepsi, Juice, Soap}

$T_1 = \{M, C, S\}$ $T_5 = \{M, S, P\}$

$T_2 = \{M, P, J\}$ $T_6 = \{M, C, S, J\}$

$T_3 = \{M, S\}$ $T_7 = \{C, S, J\}$

$T_4 = \{C, J\}$ $T_8 = \{S, C\}$

TID	set of items
T_1	M, C, S
T_2	M, P, J
T_3	M, S
T_4	C, J
T_5	M, S, P
T_6	M, C, S, J
T_7	C, S, J
T_8	S, C

Sol: Minimum support = 3

Frequent 1 - Itemset :-

{{m}, {c}, {J}, {s}}

Frequent 2 - Itemset :-

{{m, s}, {c, J} {c, s}}

Frequent 3 - Itemset :-

{ϕ}

*) Market basket Analysis :-

TID	set of items
B_1	M, C, S
B_2	M, P, J
B_3	M, S
B_4	C, J
B_5	M, S, P
B_6	M, C, S, J
B_7	C, S, J
B_8	S, C

$D = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8\}$

I = {milk, Coke, pepsi, Juice, Soap}

$B_1 = \{M, C, S\}$ $B_5 = \{M, S, P\}$

$B_2 = \{M, P, J\}$ $B_6 = \{M, C, S, J\}$

$B_3 = \{M, S\}$ $B_7 = \{C, S, J\}$

$B_4 = \{C, J\}$ $B_8 = \{S, C\}$

Sol:

Support (milk) $= \frac{5}{8} \times 100 = 62.5$

support (coke) $= \frac{5}{8} \times 100 = 62.5$

(pepsi) $= \frac{2}{8} \times 100 = 25$

(Juice) $= \frac{4}{8} \times 100 = 50$

(soap) $= \frac{6}{8} \times 100 = 75$

~~Frequent 2 - Itemset :-~~

$\frac{500}{8}$

~~support (milk, soop) = $\frac{4}{8}\times100$ = 50~~
~~support (coke, Juice) = $\frac{3}{8}\times100 = 37$~~
~~support (coke, soop) = $\frac{4}{8}\times100 = 50$~~

support (milk, coke) = $\frac{2}{8}\times100 = 25$
support (milk, pepsi) = $\frac{2}{8}\times100 = 25$
support (milk, Juice) = $\frac{2}{8}\times100 = 25$
support (milk, soop) = $\frac{4}{8}\times100 = 50$

(coke, pepsi) = 0
(coke, Juice) = $\frac{3}{8}\times100 = 37$
(coke, soop) = $\frac{4}{8}\times100 = 50$
(pepsi, Juice) = $\frac{1}{8}\times100 = 12.5$
(pepsi, soop) = $\frac{1}{8}\times100 = 12.5$
(Juice, soop) = $\frac{2}{8}\times100 = 25$

$\Rightarrow$
(M, C, P) = 0
(M, C, J) = 0
(M, C, S) = $\frac{2}{8}\times100 = 25$
(C, P, J) = 0
(C, P, S) = 0
(P, J, S) = 0

(M, C, P, J) = 0
(M, C, P, S) = 0
(C, P, J, S) = 0
(M, C, S, J) = $\frac{1}{8}\times100$
= 12.5

$\Rightarrow$

3) Let us consider a transaction database D is given below.

$D = \{T_1, T_2, T_3, T_4, T_5, T_6, \ldots\}$
$T_1 = \{NN, CC, TC, CG\}$
$T_2 = \{CC, DB, CG\}$
$T_3 = \{NN, CC, TC, CG\}$
$T_4 = \{NN, CC, DB, CG\}$
$T_5 = \{NN, CC, DB, TC, CG\}$
$T_6 = \{CC, DB, TC\}$

Itemset = {NN, CC, TC, CG, DB}
Minimum support = 33%

Sol) Normalize convert $\Rightarrow \frac{x}{6} = \frac{33}{100}$
$\Rightarrow 100x = 33\times6$
$\Rightarrow x = \frac{33\times6}{100}$
$\Rightarrow \boxed{x = 1.98 \simeq 2}$

frequent 1 Item set :-
{{NN}, {CC}, {TC}, {CG}, {DB}}

frequent 2 Itemset :-
{{NN, CC}, {NN, TC}, {NN, CG}, {NN, DB},
{CC, TC}, {CC, CG}, {CC, DB}, {TC, CG},

{TC, DB}, {CG, DB}}

frequent 3 ~~frequent 1~~ item set :-

{{NN, TC, CC}, {NN, CC, CG}, {NN, CC, DB},

{TC, CG, DB} {CC, TC, CG}}

frequent 4 ~~frequent 1~~ item set :-

{{NN, TC, CC, CG}}

frequent 5 itemset :-

{∅}

frequent 1:

support (NN) = $\frac{4}{6}\times 100$
= 66.6

support (CC) = $\frac{6}{6}\times 100$
= 100

support (TC) = $\frac{4}{6}\times 100$
= 66.6

support (CG) = $\frac{5}{6}\times 100$
= 83.3

support (DB) = $\frac{4}{6}\times 100$
= 66.6

frequent 2 :

(NN, CC) = $\frac{4}{6}\times 100 = 66.6$

(NN, TC) = $\frac{3}{6}\times 100 = 50$

(NN, CG) = $\frac{4}{6}\times 100 = 66.6$

(NN, DB) = $\frac{2}{6}\times 100 = 33.3$

(CC, TC) = $\frac{4}{6}\times 100 = 66.6$

(CC, CG) = $\frac{5}{6}\times 100 = 83.3$

(CC, DB) = $\frac{4}{6}\times 100 = 66.6$

(TC, CG) = $\frac{3}{6}\times 100 = 50$

(TC, DB) = $\frac{2}{6}\times 100 = 33.3$

(CG, DB) = $\frac{3}{6}\times 100 = 50$

frequent 3 :-

(NN, TC, CC) = $\frac{3}{6}\times 100$
= 50

(NN, CC, CG) = $\frac{4}{6}\times 100$
= 66.6

(NN, CC, DB) = $\frac{2}{6}\times 100$
= 33.3

(TC, CG, DB) = $\frac{1}{6}\times 100$ = 16.6

(CC, TC, CG) = $\frac{3}{6}\times 100$ = 50

frequent 4 :-

(NN, TC, CC, CG) = $\frac{3}{6}\times 100$ = 50

*) support :-

The support S is the percentage of transaction in D that contains A∪B.

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

Confidence :-

The confidence c is the percentage of transaction [to] that containing A that also containing B.

$$\text{Con}(A \Rightarrow B) = \frac{\text{support-Count}(A \Rightarrow B)}{\text{support}(A)} \quad \text{(or)} \; P(B|A)$$

*) The efficient and scalable frequent item set mining methods :-

Apriori algorithm :-

Apriori is an influential algorithm for mining frequent item sets for boolean association tools.

The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent item set properties.

Apriori algorithm are 2 types

*) The join step

*) prime step

1. The join step :-

*) To find L_k; a set of candidat k-item set is generated by joing L_{k-1} with it self.

*) This set of candidate is denoted by C_k let l_1 and l_2 be itemsets in L_{k-1}

*) By convention Apriori assumes that items within a transaction itemsets are sorted in lexicographic order.

* The prime step :-

*) C_k is a support of L_k, that is its members may (or) may not be present, but all of the frequent k-itemsets are include in C_k.

*) A scan of the database to determine the count of each candidate in C_k would result in the determine of L_k.

~~Apro~~ Apriori algorithm :-

Find frequent itemsets using an iterative level wise approach based on candiate generate generation.

Input: Database, D. of transactions, minimum support threshold, min-~~step~~ sup.

O/P : L, frequent itemsets in D.

Method :-

~~1. third method~~ 1. L_1 = find-frequent-1-item set(D).

2. for($k=0$; $L_{k-1} \neq \phi$; k++) {

3. C_k = apriori-gen(L_{k-1}, min-sup);

4. for each transaction $t \in D$ { //scan D for counts

5. C_t = subset (C_k, t); // get the subsets of t that are candidates

6. for each Candidate $c \in C_t$

7. c.count ++;

8. }

9. L_k = $\{c \in C_k \mid count \geq min\text{-}\boxed{\cancel{step}}\ sup\}$

10. }

11. return $L = \cup_k L_k$;

Procedure apriori-gen (L_{k-1} : frequent (k-1)-itemsets ; min-sup : minimum support thres-hold)

1. for each itemset $l_1 \in L_{k-1}$
2. for each itemset $l_2 \in l_{k-1}$
3. if $(l_1[1] = l_2[1] \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge$ $(l_1(k-2) = l_2(k-2) \cap l_1(k-1) < (l_2(k-1))$ then
4. $c = l_1 \bowtie l_2$; // join step : generate Candidates.
5. if has-infrequent-subset (c, $k-1$) then
6. ~~data~~ delete c; // prune step : remove unfruitful candidate.
7. else add c to C_k
8. }
9. return C_k;

Procedure has-infrequent-subset (c : candidate k-itemset ; L_{k-1} : frequent (k-1)-itemset :) //use prior knowledge

1. for each (k-1)-subsets of c
2. If $s \notin L_{k-1}$ then
3. return TRUE.
4. return FALSE.

Problem :-

1) Let us consider a transaltional database D with a transaction table

10M

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 = 2

Find all frequent itemset using apriori algorithm.

Itemset

scan D for candiate count →

Item set	support count
i_1	6
i_2	7
i_3	6
i_4	2
i_5	2

Compare candidate support with minimum count 2 →

Item set	Support count
i_1	6
i_2	7
i_3	6
i_4	2
i_5	2

Generate C_2 candidates from → $C_2 = L_1 \bowtie L_2$

item set
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 candidate support scan →

C_2

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

Compare candidate support count with minimum support count →

C_2

Item set	Support 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 candidate from → $C_3 = L_2 \bowtie L_2$

C_3

Item set
$\{i_1, i_2, i_3\}$
$\{i_1, i_2, i_5\}$
$\{i_1, i_3, i_5\}$
$\{i_2, i_3, i_4\}$
$\{i_2, i_3, i_5\}$
$\{i_2, i_4, i_5\}$

support D for candidate support count →

C_3

Itemset	Support Count
$\{i_1, i_2, i_3\}$	2
$\{i_1, i_2, i_5\}$	2
$\{i_1, i_3, i_5\}$	1
$\{i_2, i_3, i_4\}$	0
$\{i_2, i_3, i_5\}$	1
$\{i_2, i_4, i_5\}$	0

Compare candidate support count with minimum support count →

C_3

Itemset	Support Count
$\{i_1, i_2, i_3\}$	2
$\{i_1, i_2, i_5\}$	2

Generate C_4 candidate from → $C_4 = L_3 \bowtie L_3$

C_4

Item set
$\{i_1, i_2, i_3, i_5\}$

support D for candidate support count →

C_4

Itemset	Support Count
$\{i_1, i_2, i_3, i_5\}$	1

Compare candidate support count with minimum support count →

Itemset	Support
$\{i_1, i_2, i_3, i_5\}$	$\emptyset$

The set of one frequent item set :

$L_1 = i_1, i_2, i_3, i_4, i_5$

$L_2 = \{ \{i_1, i_2\}, \{i_1, i_3\}, \cancel{\{i_1, i_4\}}, \{i_1, i_5\}$
$\{i_2, i_3\}, \{i_2, i_4\}, \{i_2, i_5\}, \cancel{\{i_3, i_4\}}, \cancel{\{i_3, i_5\}}$
$\cancel{\{i_4, i_5\}} \}$

$L_3 = \{ \{i_1, i_2, i_3\}, \{i_1, i_2, i_5\}, \cancel{\{i_1, i_3, i_5\}},$
$\cancel{\{i_2, i_3, i_4\}}, \cancel{\{i_2, i_3, i_5\}}, \cancel{\{i_2, i_4, i_5\}} \}$

*** AA 10M

$L_4 = \{i_1, i_2, i_3, i_5\}$

2

TID	set 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, O, K, I, E}

Minimum support = 60 %

Find all frequent item set using apriori algorithm

~~Item set~~ ~~support count~~

~~Scan D for candidate count~~

Convert Minimum value

$\Rightarrow \frac{x}{5} = \frac{60}{100}$

$\Rightarrow 10x = 30$

$\Rightarrow x = \frac{30}{10}$

$\Rightarrow \boxed{x = 3}$

$\xrightarrow{\text{Scan D for Candidate Count}}$

set of items	support count
M	3
O	4
N	2
K	5
E	4
Y	3
D	1
A	1
U	1
C	2
I	1

$\xrightarrow{\text{Compare Candidate support with minimum Count 3}}$

set of items	support count
M	3
O	4
K	5
E	4
Y	3

$\xrightarrow{\text{Generate Candidates from}}$ $C_2 = L_1 \bowtie L_2$

C_2

Itemset
(M, O)
(M, K)
(M, E)
(M, Y)
(O, K)
(O, E)
(O, Y)
(K, E)
(K, Y)
(E, Y)

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

C_2

Scan D for candidate $\rightarrow$ support scan

Compare candidate support with minimum count 3 $\rightarrow$

C_2

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

Generate C_3 candidate from $\rightarrow$ $C_3 = L_2 \bowtie L_2$

C_3

Item set
{O,K,E}
{K,E,Y}
{M,O,K}
{M,K,E}

Scan D for candidate $\rightarrow$ support scan

C_3

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

Compare candidate support with minimum count 3 $\rightarrow$

C_3

Itemset	Support Count
{O,K,E}	3

→ The set of one frequent item set

L_1 = {{M},{O},{N},{K},{E},{Y},{O}, {A},{U},{C},{I}}

L_2 = {{M,O}, ~~{M,K}~~, ~~{M,E}~~, ~~{M,Y}~~, {O,K}, {O,E}, ~~{O,Y}~~, {K,E}, {K,Y}, ~~{E,Y}~~}

L_3 = {{O,K,E}, ~~{K,E,Y}~~, ~~{M,O,K}~~, ~~{M,K,E}~~}

~~L_4 = {O,K,E}~~

*) Generating association rules from frequent itemsets :-

→ Once the frequent item sets from the transaction in database 'D' have been found it is straight forward to generate strong association rule from them (where strong associations rules satisfies both minimum support and minimum confidence). This can be done using the following equation are confidence. where the conditional prob.

For the expressed in terms of item sets

Support Count

$$\boxed{\text{Confidence}(A \Rightarrow B) = P(B/A) = \frac{\text{support\_Count}(A \cup B)}{\text{support\_Count}(A)}}$$

where support_Count (AUB) is the no.of transaction containing in the item sets AUB. and support count A is the no.of transaction containing the item sets A. Based on the this equation Association rules ~~from the~~ ~~generatic~~ can be generated rules.

*) For each frequent item set is 'l' generate add non-empty subset of 'l' for every non-empty subset of 'l' output they rule.

$$\boxed{S \Rightarrow (l-S) \text{ if } \frac{\text{support\_count}(l)}{\text{support\_Count}(S)} \geq \text{min.Confidence}}$$

where min. confidence is the minimum confidence threshold.

Since the rules are generated from frequent item sets. Each one automatically satisfies minimum support.

(1) Let us consider transactional database 'D' with a transaction given the following data

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_3
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 = 2
Minimum confidence = 70%

(Apriori algorithm)

The set of one frequent item sets

$L_1 = i_1, i_2, i_3, i_4, i_5$

$L_2 = \{\{i_1, i_2\}, \{i_1, i_3\}, \cancel{\{i_1, i_4\}}, \{i_1, i_5\} \{i_2, i_3\}$
$\{i_2, i_4\}, \{i_2, i_5\}, \cancel{\{i_3, i_4\}}, \cancel{\{i_3, i_5\}}$
$\cancel{\{i_4, i_5\}}$

$L_3 = \{\{i_1, i_2, i_3\}, \{i_1, i_2, i_5\}, \cancel{\{i_1, i_3, i_5\}}$

$l = \{i_1, i_2, i_5\}$

The set of all non-empty subset of l are

$l_1 = \{\{i_1\}\{i_2\}, \{i_5\}\}$

$l_2 = \{\{i_1, i_2\}, \{i_1, i_5\}, \{i_2, i_5\}\}$

$l_3 = \{i_1, i_2, i_5\}$

Association rules.

$s = i_1$

$S = l - s$

$S = \{i_2, i_5\}$

Confidence

$i_1 \Rightarrow i_2 \cap i_5$

Confidence $= \frac{l}{s} \cancel{\times 100}$

$= \frac{(i_1, i_2, i_5)}{i_1} \times 100$

$= \frac{2}{6} \times 100$

$= 33\%$

find table

$\{i_1, i_2, i_5 = 2$

$i_1 = 6\}$

Association rules :-

$s = i_2$

$S = l - s$

$S = \{i_1, i_5\}$

Confidence $= \frac{l}{s}$

$= \frac{(i_1, i_2, i_5)}{i_2} \times 100$ $(\because i_2 = 7)$

$= \frac{2}{7} \times 100$

$= 29\%$

Association rules :-

$s = i_5$

$S = l - s$

$S = \{i_1, i_2\}$ $(\because i_5 = 2)$

Confidence $= \frac{l}{s}$

$= \frac{(i_1, i_2, i_5)}{i_5} \times 100$

$= \frac{2}{2} \times 100 = 100\%$

$\Rightarrow \{i_1, i_2\}, \{i_1, i_5\}, \{i_2, i_5\}$

Association rules :-

$s = \{i_1, i_2\}$

$S = l - s$ $(\because i_1, i_2 = 4)$

$S = i_5$

Confidence $= \frac{l}{s} = \frac{(i_1, i_2, i_5)}{(i_1, i_2)}$

$= \frac{2}{4} \times 100$

$= \frac{1}{2} \times 100$

$= 50\%$

Association rule

$s = \{i_1, i_5\}$

$S = l - s$ $(\because i_1, i_5 = 2)$

$S = i_2$

Confidence $= \frac{l}{s}$

$= \frac{(i_1, i_2, i_5)}{(i_1, i_5)}$

$= \frac{2}{2} \times 100$

$= 100\%$

Association rule.

$s = \{i_2, i_5\}$

$S = l - s$

$(\because i_2, i_5) = 2$

$S = i_1$

Confidence $= \frac{l}{s}$

$= \frac{\{i_1, i_2, i_5\}}{(i_2, i_5\}} = \frac{2}{2} \times 100 = 100\%$

If the minimum confidence is threshold then only is 70%. then only that is $\{i_2, i_5\}, \{i_5\}, \{i_1, i_5\}$ these are the output

$\therefore$ since these are the only once generated that are strong.

$\Rightarrow \{i_1, i_2, i_3\}$

The set of frequent subset of non-empty set of l are

$l = \{i_1, i_2, i_3\}$

$= \{i_1\}, \{i_2\}, \{i_3\}$

$= \{i_1, i_2\}, \{i_2, i_3\}, \{i_1, i_3\}$

$= \{i_1, i_2, i_3\}$

$l = \{i_1, i_2, i_3\}$

Association rule:

$S = i_1$

$S = l - S$

$S = \{i_2, i_3\}$

($\because i_1, i_2, i_3 = 2$, $i_1 = 6$)

Confidence $= \frac{l}{S}$

$= \frac{\{i_1, i_2, i_3\}}{i_1} = \frac{2}{6} = \frac{1}{3} \times 100 = 37\%$

Association rule :-

$S = i_2$

$S = l - S$

$S = \{i_1, i_3\}$

Confidence $= \frac{l}{S}$ ($\because i_2 = 7$)

$= \frac{2}{i_2}$

$= \frac{2}{7} \times 100$

$= 29\%$

Association rule :-

$S = i_3$

$S = l - S$

$S = \{i_1, i_2\}$ ($\because i_3 = 6$)

Confidence $= \frac{l}{3}$

$= \frac{2}{6}$

$= \frac{1}{3} \times 100$

$= 37\%$

$\Rightarrow \{i_1, i_2\}, \{i_2, i_3\}, \{i_1, i_3\}$

Association rule

$S = \{i_1, i_2\}$

$S = l - S$ ($\because i_1, i_2 = 4$)

$S = \{i_3\}$

Confidence $= \frac{l}{S}$

$= \frac{2}{(i_1, i_2)}$

$= \frac{2}{4} = \frac{1}{2} \times 100 = 50\%$

Association rule

$S = \{i_2, i_3\}$

$S = l - S$ ($\because i_2, i_3 = 6$)

$S = i_1$

Confidence $= \frac{l}{S}$

$= \frac{2}{6}$

$= \frac{1}{3} \times 100 = 37\%$

Association rule

$S = \{i_1, i_3\}$

$S = l - S$

$S = i_2$

($\because i_1, i_3 = 4$)

Confidence $= \frac{l}{S}$

$= \frac{2}{4}$

$= \frac{1}{2} \times 100$

$= 50\%$

②

TID	set of items
T_{100}	M, O, N, K, E, Y
T_{200}	D, O, N, K, E, Y
T_{300}	M, A, K, E
T_{400}	M, U, C, K, Y
T_{500}	C, O, O, K, I, E

The set of ~~frequent~~ item set

$L_1 = \{M, O, K, E, Y\}$

$L_2 = \{M, K\}, \{O, K\}, \{O, E\}, \{K, E\}, \{K, Y\}$

$L_3 = \{O, K, E\}$

$l = \{O, K, E\}$

The set of all non-empty subset of l are

$l_1 = \{O\}, \{K\}, \{E\}$

$l_2 = \{O, K\}, \{O, E\}, \{K, E\}$

Association rule :-

$S = O$

$s = l - s$

$s = \{K, E\}$ $(O, K, E = 3)$

$(\because O = 4)$

Confidence $= \frac{l}{s}$

$= \frac{\{O, K, E\}}{O}$

$= \frac{3}{4} \times 100 = 75\%$

Association rule

$S = K$

$s = l - s$ $(\because K = 5)$

$s = \{O, E\}$

Confidence $= \frac{l}{s}$

$= \frac{\{O, K, E\}}{K}$

$= \frac{3}{5} \times 100 = 60\%$

$S = E$

$s = l - s$

$s = \{K, O\}$ $(\because E = 4)$

Confidence $= \frac{l}{s}$

$= \frac{(O, K, E)}{E}$

$= \frac{3}{4} \times 100$

$= 75\%$

$\Rightarrow \{O, K\}, \{O, E\}, \{K, E\}$

Association rule :-

$S = \{O, K\}$

$s = l - s$ $(\because O, K = 3)$

$s = E$

Confidence $= \frac{l}{s}$

$= \frac{\{O, K, E\}}{\{O, K\}}$

$= \frac{3}{3} \times 100$

$= 100\%$

Association rule :-

$S = \{O, E\}$

$s = l - s$ $(\because O, E = 3)$

$s = K$

Confidence $= \frac{l}{s}$

$= \frac{\{O, K, E\}}{\{O, E\}}$

$= \frac{3}{3} \times 100$

$= 100\%$

Association rule :-

$S = \{K, E\}$

$s = l - s$

$s = K$

Confidence $= \frac{l}{s}$

$= \frac{\{O, K, E\}}{\{K, E\}}$

$= \frac{3}{3} \times 100$

$= 100\%$

*) mining frequent item set without candidate Generation :- (or) frequent pattern algorithm

① Let us consider the transactional database D with 9 transactional than as given in the following table.

TID	List of Item IDs
T100	i1, i2, i5
T200	i2, i4
T300	i2, i3
T400	i1, i2, i4
T500	i1, i3
T600	i2, i3
T700	i1, i3
T800	i1, i2, i3, i5
T900	i1, i2, i3

Minimum Support = 2

Mine all frequent item set using frequent pattern algorithm.

Sol. 1. Construction of frequent pattern tree

⊛ Scan database to set support count of each itemset.

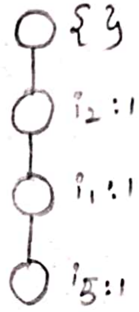
Itemset	Support Count
i1	6
i2	7
i3	6
i4	2
i5	2

L-order :-

L = {{i2 : 7}, {i1 : 6}, {i3 : ...

T100 = {i1, i2, i5}

⇒ L-order = {i2, i1, i5}

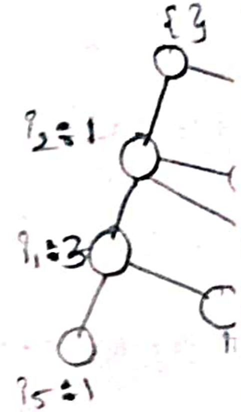


T300 = {i2, i3}

⇒ L-order = {i2, i3}

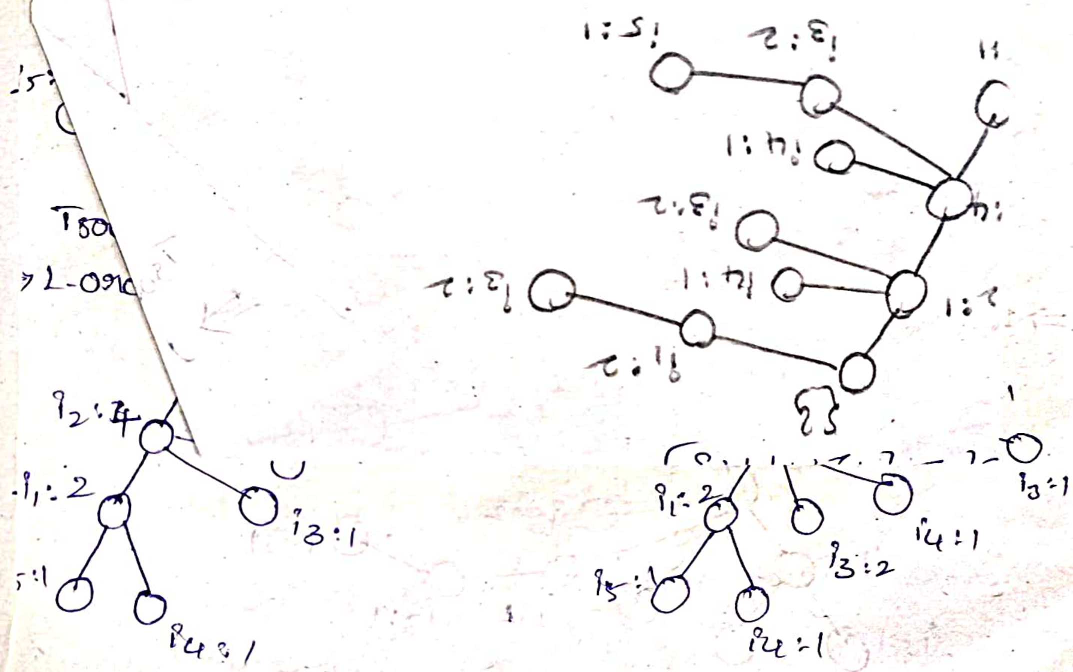
T700 = {i1, ...

⇒ L-order =



T800 = {...}

⇒ L-order

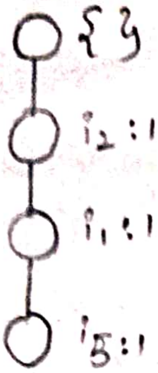


L-order :-

$L = \{\{i_2 : 7\}, \{i_1 : 6\}, \{i_3 : 6\}, \{i_4 : 2\}, \{i_5 : 2\}\}$

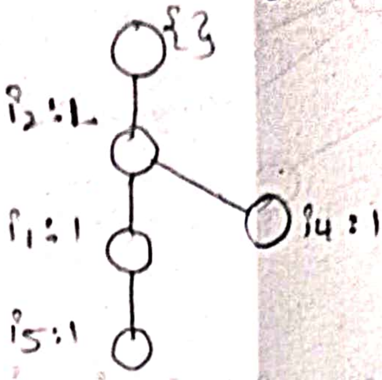
$T100 = \{i_1, i_2, i_5\}$

$\Rightarrow$ L-order $= \{i_2, i_1, i_5\}$



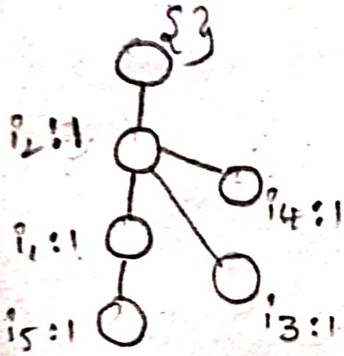
$T200 = \{i_2, i_4\}$

$\Rightarrow$ L-order $= \{i_2, i_4\}$



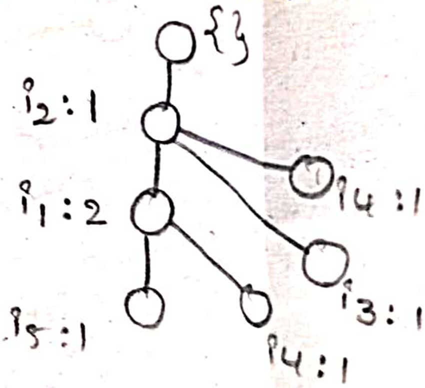
$T300 = \{i_2, i_3\}$

$\Rightarrow$ L-order $= \{i_2, i_3\}$



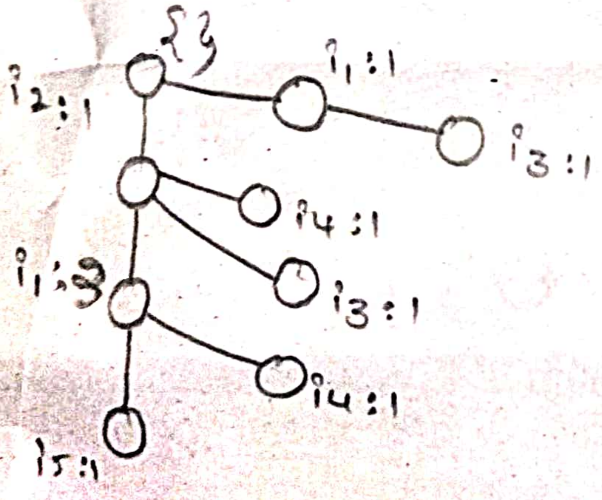
$T400 = \{i_1, i_2, i_4\}$

$\Rightarrow$ L-order $= \{i_2, i_1, i_4\}$



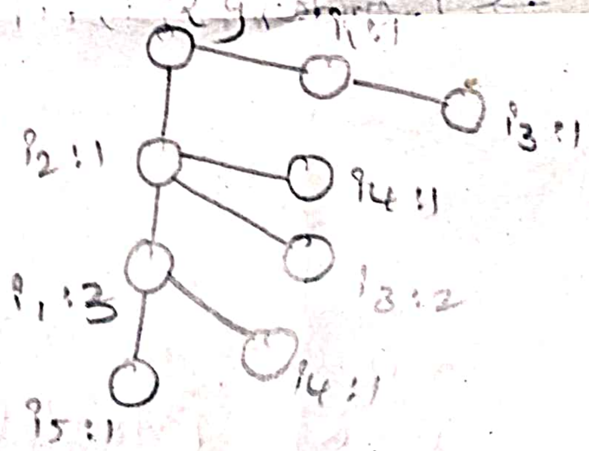
$T500 = \{i_1, i_3\}$

$\Rightarrow$ L-order $= \{i_1, i_3\}$



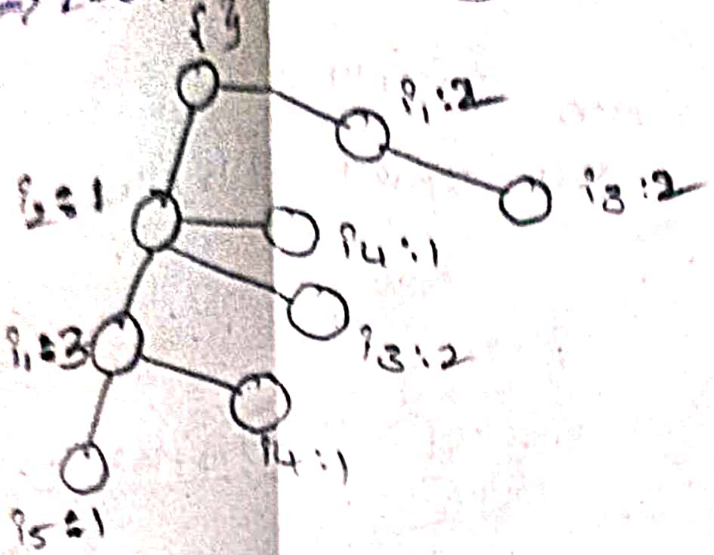
$T600 = \{i_2, i_3\}$

$\Rightarrow$ L-order $= \{i_2, i_3\}$



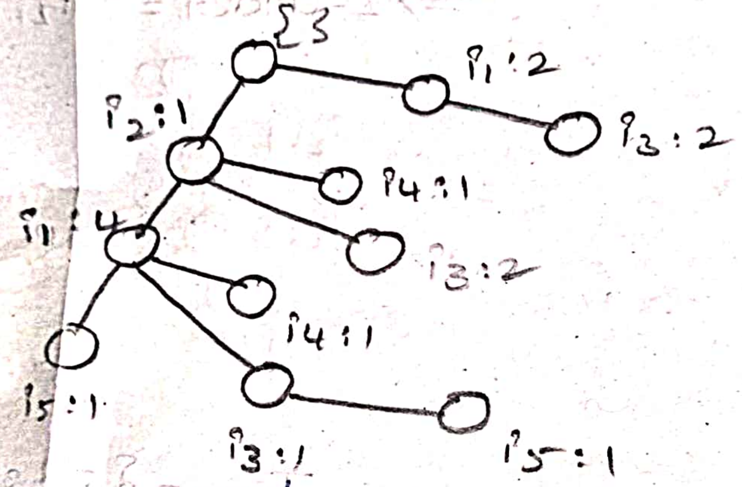
$T_{700} = \{i_1, i_3\}$

$\Rightarrow$ L-order $= \{i_1, i_3\}$



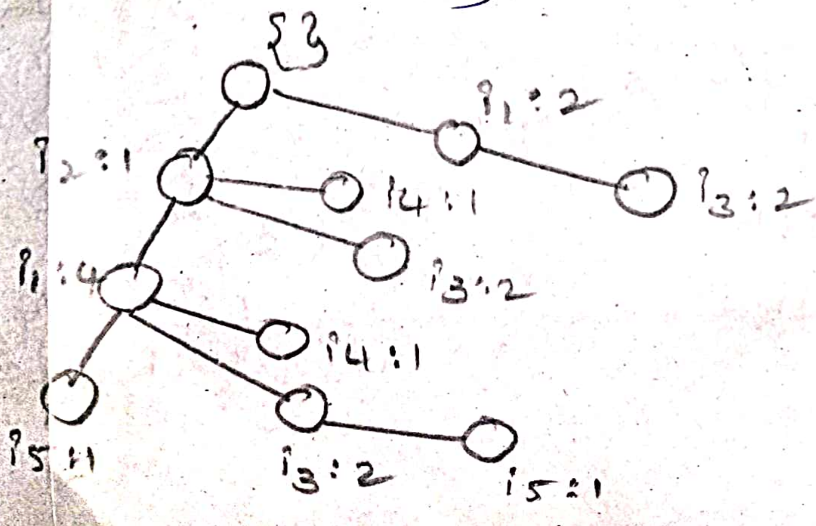
$T_{800} = \{i_1, i_2, i_3, i_5\}$

$\Rightarrow$ L-order $= \{i_2, i_1, i_3, i_5\}$



$T_{900} = \{i_1, i_2, i_3\}$

$\Rightarrow$ L-order $= \{i_2, i_1, i_3\}$



Mining frequent pattern lists Tree :-

Item set	Conditional pattern base	Conditional fp tree	Frequent pattern Generation
I_5	$\{\{I_1, I_2\}:1\}$ $\{\{I_1, I_2, I_3\}:1\}$	$\{i_1, i_2 : 2\}$	$\{i_1, i_5 : 2\}$ $\{i_2, i_5 : 2\}$ $\{i_1, i_2, i_5 : 2\}$
i_4	$\{i_1, i_2 : 1\}$ $\{i_2 : 1\}$	$\{i_2 : 2\}$	$\{i_2, i_4 : 2\}$
i_3	$\{i_1, i_2 : 2\}$ $\{i_2 : 2\}$ $\{i_1 : 2\}$	$\{i_1, i_2 : 4\}$ $i_1 : 2$	$\{i_1, i_3 : 4\}$ $\{i_2, i_3 : 4\}$ $\{i_1, i_2, i_3 : 2\}$
i_2	—	—	—
i_1	$\{i_2 : 4\}$	$\{i_2 : 4\}$	$\{\{i_2, i_1\} : 4\}$

The set of all frequent 1-items sets

$L_1 = \{i_1, i_2, i_3, i_4, i_5\}$

The set of all two frequent sets.

$L_2 = \{i_1, i_5\}\{i_2, i_5\}\{i_1, i_3\}\{i_2, i_3\}\{i_2, i_1\}, \{i_2, i_4\}$

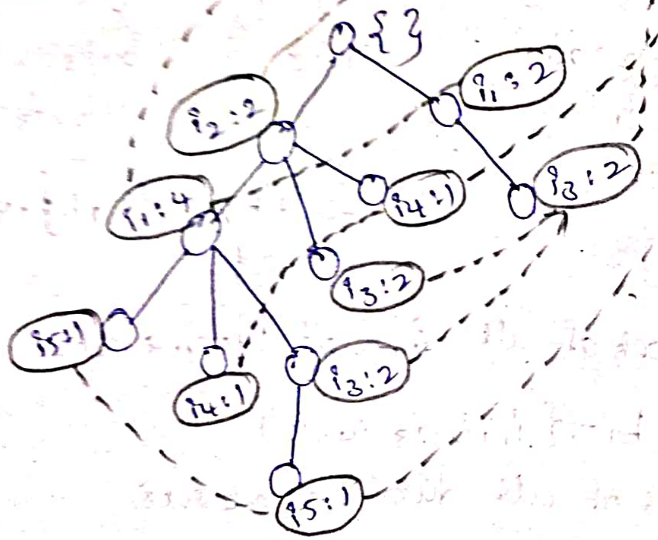
The set of all three frequent itemsets

$L_3 = \{i_1, i_2, i_5\}\{i_1, i_2, i_3\}$

The set of all four frequent itemsets

$L_4 = \{\ \}$

itemid	support count	Node link
i_2	7	
i_1	6	
i_3	6	
i_4	2	
i_5	2	



frequent pattern growth algorithm :-

Algorithm :- fp-growth mine frequent pattern using an fp-tree by pattern fragment growth.

Input :- A transaction database, D, minimum support threshold union-sup.

output :- The complete set of frequent patterns.

1. The fp-tree is constructed in the following steps:

a. scan the transaction database D once. collect the set of frequent item F and their supports sort F in support descending order as L, the list of frequent items.

b. create the root of an fp-tree & label is an "null" for each transaction trans in D do the following.

Select and sort the frequent items in trans according to the order of L. Let the sorted frequent item list in Trans be [p|P] where p is the first element and P is the remaining list Call insert-tree [p|P] which is performed as fallows.

If T has a child N such that N item-name p item-name, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T and its node-link to the nodes with the same item_ name via the node-link structure if is non-empty, call insert-tree(P,N) ~~respe-~~ recursively.

2. Mining of an fp-tree is performed by calling fp-growth (fp-tree, null) which is implemented as follows.

Procedure fp-growth (Tree, α) :-

→ If tree contains a single path p then
→ for each combination (denotes as β) of the node in the path p.
→ Generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in β.
→ else for each a_i in the header of Tree
{
→ Generate pattern $\beta = a_i \cup \alpha$ with support = a_i.support;
→ Construct β's conditional pattern base and then β's conditional fp-tree Treeβ;
→ If Tree $\beta \neq \emptyset$ then
→ call f-growth (Tree β, β);
}

*) Mining various kinds of association rules :-

1) mining multilevel association rules
2) " multi Dimensional "
3) " Quantitative "

① mining multilevel association rules :-

It is difficult to find strong associations among data item at low levels of abstraction due to the sparsity of data at those levels strong association discovered at high level of abstractions.

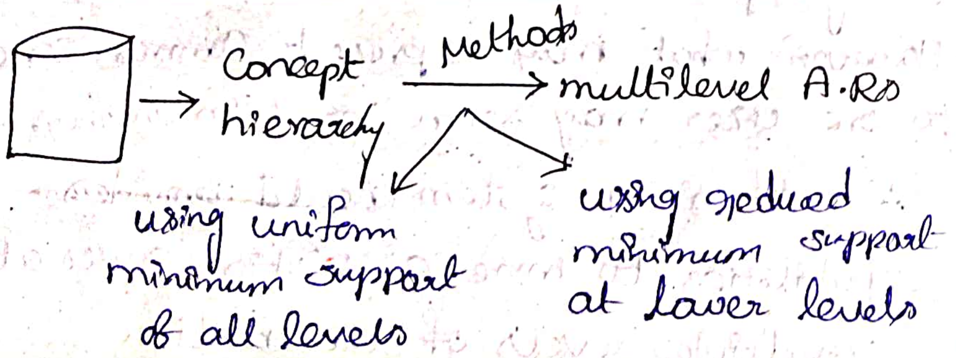
may represents common sence knowledge. However, what may represents common sence to one user may seem novel to another.

∴ Data mining system should provides capabilities to mine association rules at multiple levels of abstraction.

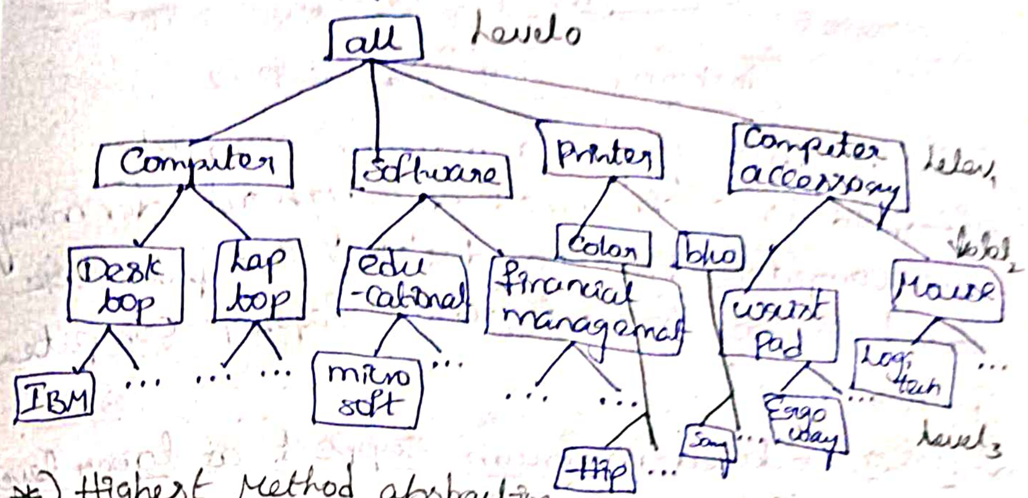
Definition :- Rules generated from association rules mining with Concept hierarchys are Called multiple level or multi level association rules.

Eg :-

TID	Items purchased
T_1	IBM desktop Computer, Sony blue printer
T_2	microsoft educational software microsoft financial management software.
T_3	logitech mouse Compiler accessory Ergoway wrist pad Computer accessory
T_4	IBM desktop Computer, microsoft financial management software
T_5	IBM desktop Computer
⋮	⋮



Concept Hierarchy :-



*) Highest Method abstraction

*) Individual elements

*) we take level, level, with differernat element will take the hierarchy

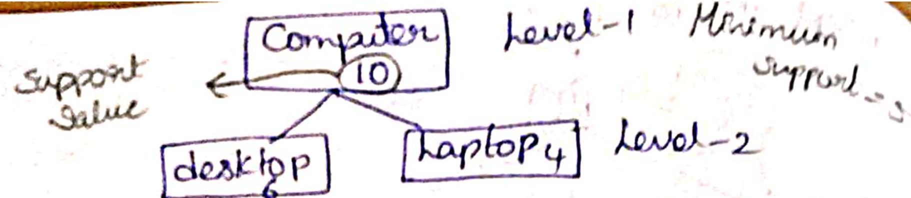
Two Methods :-

1. using uniform minimum support of all levels
2. using reduced minimum support at lower levels.

Data Can be Generalized by replacing low level concept with in the data. by their High level concept from a Concept hierarchy.

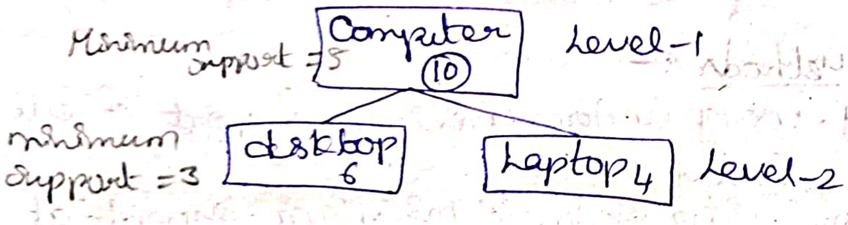
1. using uniform minimum support of all levels

The same minimum support threshold is used when mining at each level of abstraction



In the above figure a minimum support threshold is 5 it used throughout. for mining from computer down to desktop computer are found to be frequent while laptop computer is not found.

2. using reduced minimum support at lower levels
Each level of abstraction as its own minimum support threshold. the lower the abstraction level,



In the above figure a the minimum support threshold for level 1 & level 2 are 5 & 3 respectively.

In this way computer, laptop ~~Computer~~, desktop are found with minimum support.

② Mining multidimensional association rules from relational database and datawarehouse

Definition :-

we have studied association rules that simply a single predicate.

i.e, predicate buys.

In mining our all electronic database, we may discover the boolean association rule.

IBM desktop computer ⇒ sony b/w printer

single dimensional :-

buys (x, "IBM desktop computer") ⇒ buys (x, "sony b/w printer")

As a single dimensional contains a single distinct predicate (buys) with multiple occurrences in the predicate occurs more than once within the rule.

multi dimensional (or) hybrid-dimension :-

age (x, "20 ... 29") ∧ occupation (x, "student") ⇒ buys (x, "laptop")

Association rules that involves two or more dimension or predicate can be referred to as multi dimensional association rule.

~~→ we say that~~

→ each predicate which occurs only once in the rule. Hence it has "no pre ~~r~~ repeated predicates."

→ No repeated predicates are called Interdimension association rule.

$$\boxed{\text{age}(X, \text{"20...29"}) \wedge \text{buys}(X, \text{"laptop"}) \Rightarrow \text{buys}(X, \text{"b/w printer"})}$$

(3) Mining Quantitative Association rules :-

Quantitative association rules are multidimensional association rules in which the numeric attributes are dynamically discretized during the mining process.

2-D quantitative association :-

$$\text{age}(X, \text{"30....39"}) \wedge \text{income}(X, \text{"42k... 48k"}) \Rightarrow \text{buys}(X, \text{"high resolution TV"})$$

The following steps are involved in ARCS (Association rule clustering system)

Binning :-

→ Quantitative attributes can have a very wide range of values defining their domain.

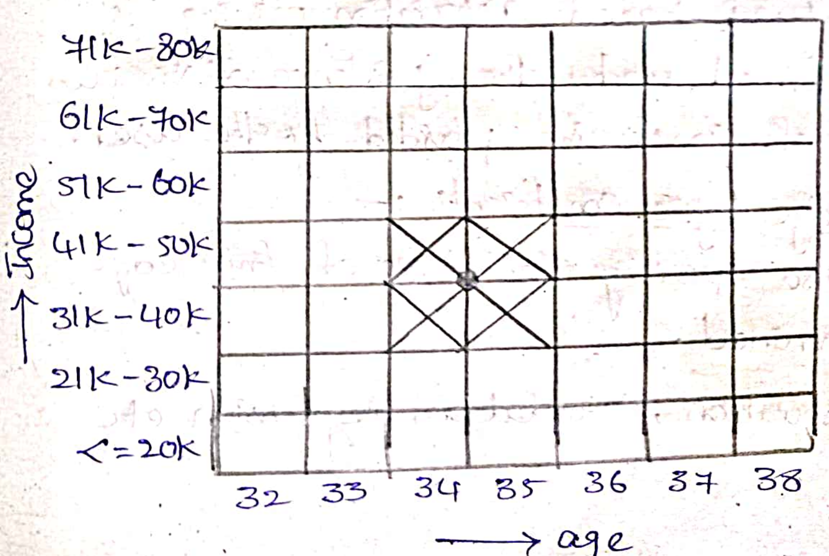
→ The partitioning process is reffered to as binning.

i.e, where the intervals are considered "bins". binning are classified into three strategies they are.

1. equiwidth binning :- where the interval size of each bins is the same.
2. equidepth binning :- where each bin has approximately the same number of tuples assigned.
3. Homogeneity-based binning :- where bin size is determined so that the tuples in each bin uniformly distributed

Finding frequent predicate sets :-

Once the 2-D array containing the count distribution for each category is set up, this can be scanned in order to find the frequent predicate sets.



Clustering the association rules :-

age(X, 34) ∧ income(X, "31K...40K") ⟹
buys(X, "high resolution TV") ~~(6.15)~~

age(X, 35) ∧ income(X, "31K.....40K") ⟹
buys(X, "high resolution TV")

age(X, 34) ∧ income(X, "41K...50K") ⟹
buys(X, "high resolution TV")

age(X, 35) ∧ income(X, "41K...50K") ⟹
buys(X, "high resolution TV")

*) Constraint-Based Association mining :-

Constraint - Condition

→ Association rules are generated based on conditions.

→ In Constraint-based association mining is performed under the guidence of various kinds of Constraint provided by the user.

Knowledge type Constraints :-

→ These specify the type of knowledge to be mined.

→ association, Correlation, regression etc

Data Constraints :-

→ specify the set of task-relevant data

Dimensional / level :-

→ specify the dimension (→ attribute) of the data or level of concept hierarchies

→ dimension which generate to the rules

→ level which generate to the rules

→ used to concept hierarchies with used to levels

Interestingness Constraints :- pickup into

→ The particular data ^how much data will interesting (or) thresholds or statistical measures support, confidence are used to identity

Rule Constraints :-

→ specifies the form of rules to be mined

mined they are :-

1. Metarules guided mining → syntatic forms of rules is specified
   (→ rules about rules)

2. Constraint pushing

↓

variables Cost increasing / sales decreasing

relationship

Classification & prediction.

Techniques for classification

* decision tree classification
* Bayesian "
* Bayesian belief network classification
* rule Based classification model
* support vector machine.
* Neural network classification
* Back propagation classification
* K-nearest neighbour classification.
* case based reasoning classification
* genetic algorithm
* fuzzy logic based classification.

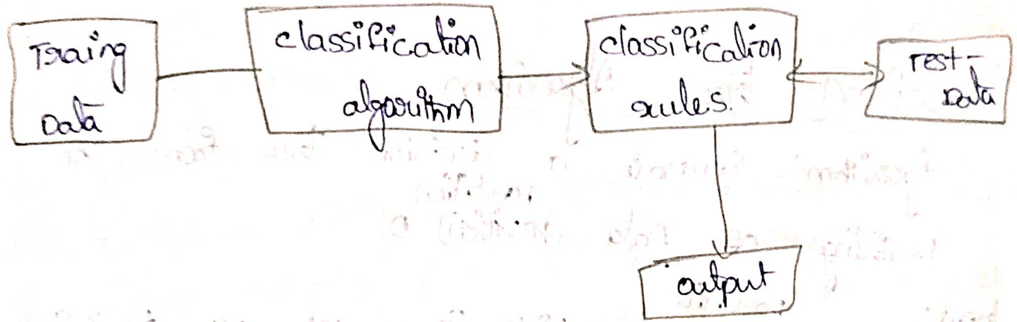
Prediction Techniques :-

→ linear regression
→ multi linear Regression
→ Non-linear regression.

issues regarding classification & prediction

* Data cleaning.
* Data integration.
* Data reduction.
* Data transformation.

Classification & prediction:-

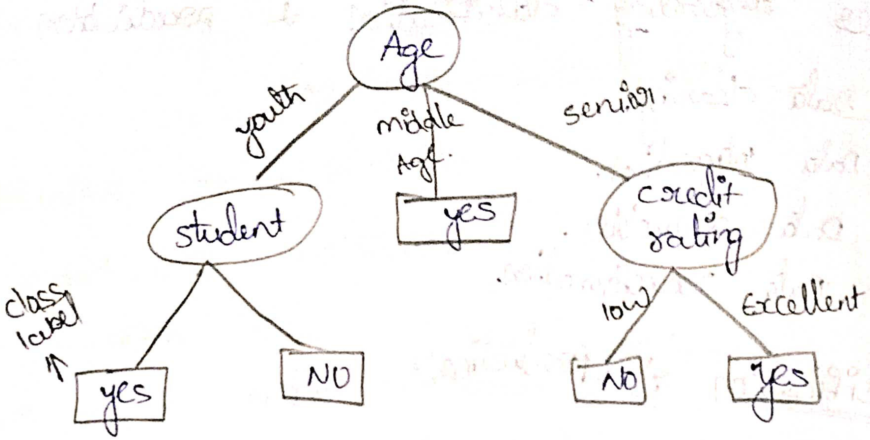


Classification by decision Tree Induction :-

* Decision tree Induction is learning of decision tree from class label training data tuples(set)

* Decision tree :- is flow chart like tree structure. where each internal node denotes a test an attribute and each branch represents the outcome of the test. and each leaf node having the class lables.

A decision tree



Imp **

Decision tree Algorithm

(15) Algorithm:- Generate a decision tree from the training of Data partition (position) D.

Input:- Data partition (position) D, which is a set of training tuples and their associated class labels.

* attribute list the set of candidate attributes.
* Attribute - selection method, a procedure to determin the spliting criterian that "best" partitions the data tuples into individual classes. this criterian consists of a spliting-attribute and possibly either a split pointer spliting subset.

output:- A Decision Tree

(1) create a Node N;

(2) if tuples in D are all of the same class, c then

(3) return N as a leaf node labeled with the classes.

(4) if attribute-list is empty then

(5) return N as a leaf node labeled with the majority class in D; // majority voting.

(6) apply attribute-selection-method (D, attribute-list] to find the "best" spliting-criterian

(7) label node N with splitting-criterian

(8) if splitting-attribute is discrete-valued and multi-way splits allowed then

(9) attribute-list ← attribute-list - splitting-attribute

(10) for each outcome j of splitting - criterian

(11) let Dj be the set of data tuples in D satisfying outcome j;

(12) if Dj is empty then

(13) attach a leaf labeled with the majority class in D to Node N;

(14) Else attach the node returned by generate decision-tree (Dj, attribute-list) to node N;

(15) End for

(16) return N;

Problem:-

let us consider a training data 'D' with 14 data tuples and two class label same: {yes, n has given the following table.

RID	Age	Income	student	credit rating	[class label] buys computer
1	youth	high	NO	Avg	NO
2	youth	high	NO	Excellent	NO
3	middle	high	NO	Avg	yes
4.	senior	medium	NO	Avg	yes
5.	senior	low	yes	Avg	yes
6.	senior	low	yes	excellent	NO
7.	middle	low	yes	Excellent	yes
8.	youth	medium	NO	Avg	NO
9.	youth	low	yes	Avg	yes
10.	senior	medium	yes	Avg	yes
11.	youth	medium	yes	excellent	yes
12.	middle	medium	NO	excellent	yes
13.	middle	high	yes	Avg	yes
14.	senior	medium	NO	excellent	NO.

let us construct a decision tree for the above training Dataset using decision tree algorithm.

Formules :-

① Entire Data set:-

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

where m = no. of. class lables.

p_i = probability out i^{th} class.

② one attribute:-

$$Info(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

③ Information Gain (D) = $Info(D) - Info_A(D)$

* Total problem based on class label in last column.

* How many possibilities are there in column.

① Entire data set:

$$Info(D) = -\sum_{i=1}^{2} P_i \log_2 P_i$$

$$= -\left[P_1 \log_2(P_1) + P_2 \log_2(P_2)\right]$$

$$= -\left[\frac{9}{14}\log_2\left(\frac{9}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right]$$

$$= -(-0.94)$$

$$= 0.94$$

② one attribute :- Age.

age: y $\langle$ 2, 3; m $\langle$ 4, 0; s $\langle$ 3, 2

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$= -\frac{5}{14}\left[\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right]$$

$$= -\frac{4}{14}\left[\frac{4}{4}\log_2\left(\frac{4}{4}\right) + \frac{0}{4}\log_2\left(0/4\right)\right]$$

$$-\frac{5}{14}\left[\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right]$$

$$= 0.694$$

Information gain = Info(D) − InfoA(D)

$$= 0.94 - 0.694$$

$$= \cancel{0.503}.$$

$$= 0.246$$

one attribute for income

income: high $\langle$ 2(y), 2(N); medium $\langle$ 4(y), 2(N); low $\langle$ 3(y), 1(N)

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$= -\frac{4}{14}\left[\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right]$$

$$-\frac{6}{14}\left[\frac{4}{6}\log_2\left(\frac{4}{6}\right) + \frac{2}{6}\log_2\left(\frac{2}{6}\right)\right].$$

$$-\frac{4}{14}\left[\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right]$$

$$= 0.285 + 0.393 + 0.231$$

$$= 0.91$$

Information gain (D) = Info(D) − InfoA(D)

$$= 0.94 - 0.91$$

$$= 0.03.$$

one attribute for student:

stu: yes $\langle$ 6, 6; NO $\langle$ 4, 3

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$= -\frac{7}{14}\left(\frac{4}{7}\log_2\left(\frac{4}{7}\right) + \frac{3}{7}\log_2\left(\frac{3}{7}\right)\right)$$

$$-\frac{7}{14}\left[\frac{1}{7}\log_2\left(\frac{1}{7}\right) + \frac{6}{7}\log_2\left(\frac{6}{9}\right)\right]$$

$$= 0.492 + 0.295$$

$$= 0.787$$

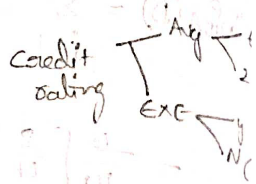
In gain (D) = Info(D) − InfoA(D)

$$= 0.94 - 0.78$$

$$= 0.16.$$

one - attribute for credit rating.

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$



$$= \frac{-8}{14}\left[\frac{6}{8}\log_2\left(\frac{6}{8}\right) + \frac{2}{8}\log_2\left(\frac{2}{8}\right)\right]$$

$$-\frac{6}{14}\left[\frac{3}{6}\log_2\left(\frac{3}{6}\right) + \frac{3}{6}\log_2\left(\frac{3}{6}\right)\right]$$

$$= 0.463 + 0.428$$

$$= 0.891$$

Information gain(D) = Info(D) − $Info_A$(D).

$= 0.94 - 089$

$= 0.05$

Imp

** **Bayesian classification:-**

Bayesian classifiers are ~~stelical~~ stastical classifier

They can predict class membership probabilities such as the probability that a given sample belongs to a particular class.

Bayeisan classification Based on Bayes theorem

Bayes theorem:

$$P(c|x) = \frac{P(x/c).P(c)}{P(x)}$$

It finds posterior probability and priority probability.

* It find the posterior probability of a class conditional.

Algorithm:-

step 1:-

* let a 'D' be a training dataset of Data tuple associated with class lables.

step 2:-

* each tuple is represented by an n-dimensional vector $X = (x_1, x_2 --- x_n)$ where $x_1, x_2 --- x_n$ are values of attributes.

* Suppose that there are m no. of classes $C_1, C_2 --- C_m$ given a data tuple x the classifier is predict that $x \in$ the class having the highest posterior probability condition on X.

* This can be written mathematically where $j = 1, 2, --- m$ and $i \neq j$.

i.e., we maximize $P(C_i|x)$

Bayes theorem $P(C_i|x) = \frac{P(x/C_i)P(C_i)}{P(x)}$

step 3:-

As $P(x)$ is constant for all classes it is enough to maximize only that is numerical function has $P(x|C_i)$.

* If the class probability is not known
$P(C_1) = P(C_2) = \cdots = P(C_m)$
so it is enough to maximize only $P(x/C_i)$

step 4:-

$$P(x/C_i) = \prod_{k=1}^{m} P(x_k / C_i)$$

$$= P(x_1/C_i) \times P(x_2/C_i) \times \cdots \times P(x_m/C_i)$$

(i) If A_k categorical $P(x_k/C_i)$ is number of tuples of class C_i in 'D' having the value small S_k for A_k / no. of tuples of C_i in D.

(ii) A_k is continuous $g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\,\sigma C_i} \times e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

Where $g(x, \mu, \sigma)$ is the Gaussian (normal) density function for attribute A_k while μ and σ are the mean and standard deviation, respectively given the values for attributes A_k for training samples of class C.

step 5:-

The classifier predict the class lable is C_i if and only if $P(x/C_i)\,P(C_i) > P(x/C_j) \cdot P(C_j)$ for $1 \le j \le m$, $j \ne i$

RID	Age	Income	student	credit rating	buys Computer
1	youth	high	NO	Avg	NO
2	"	"	"	Exce	"
3	middle	"	"	Avg	yes
4	senior	medium	"	"	"
5	senior	low	yes	"	"
6	senior	"	yes	Exce	NO
7	middle	"	yes	"	yes
8	youth	medium	~~yes~~ NO	Avg	NO
9	youth	low	~~NO~~ yes	"	yes
10	senior	medium	yes	"	"
11	youth	"	"	Exce	"
12	middle	"	NO	Exce	"
13	middle	high	yes	Avg	yes
14	senior	medium	NO	Exce	NO

X = (age = "<=30", income = "medium", student = "yes", credit rating = "~~fair~~ avg.") predict the class labeled data tuple 'x' using bayesian classification algorithm.

Sol:- The prior probability of each class.

$P(\text{buys\_computer} = \text{"yes"}) = \frac{9}{14} = 0.6428$

$P(\text{buys\_computer} = \text{"NO"}) = \frac{5}{14} = 0.3571$

young:-

$P(\text{age} <= 30 \mid \text{buys.computer} = \text{"yes"}) = \frac{2}{9} = 0.222$

$P(\text{age} <= 30 \mid \text{buys computer} = \text{"NO"}) = \frac{3}{5} = 0.600$

medium :-

$P(\text{income} = \text{"medium"} \mid \text{buys computer} = \text{"yes"}) = \frac{4}{9} = 0.444$

$P(\text{income} = \text{"medium"} \mid \text{buys computer} = \text{"NO"}) = \frac{2}{5} = 0.400$

student:-

$P(\text{student} = \text{"yes"} \mid \text{buys computer} = \text{"yes"}) = \frac{6}{9} = 0.666$

$P(\text{student} = \text{"yes"} \mid \text{buys computer} = \text{"NO"}) = \frac{1}{5} = 0.200$

credit - rating:-

$P(\text{credit\_rating} = \text{"avg"} \mid \text{buys computer} = \text{"yes"}) = \frac{6}{9} = 0.666$

$P(\quad " \quad = \text{"avg"} \mid \text{buys computer} = \text{"NO"}) = \frac{2}{5} = 0.400$

Using the above probabilities :-

$P(X \mid \text{buys.computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.666 \times 0.666$

$= 0.0437$

$P(X \mid \text{buys.computer} = \text{"NO"}) = 0.600 \times 0.400 \times 0.200 \times 0.400$

$= 0.019$

$P(X \mid \text{buys.computer} = \text{"yes"})\, P(\text{buys\_computer} = \text{"yes"})$

"yes" $= 0.0437 \times 0.6428 = 0.028$

"NO" $= 0.019 \times 0.3571 = 0.0069$

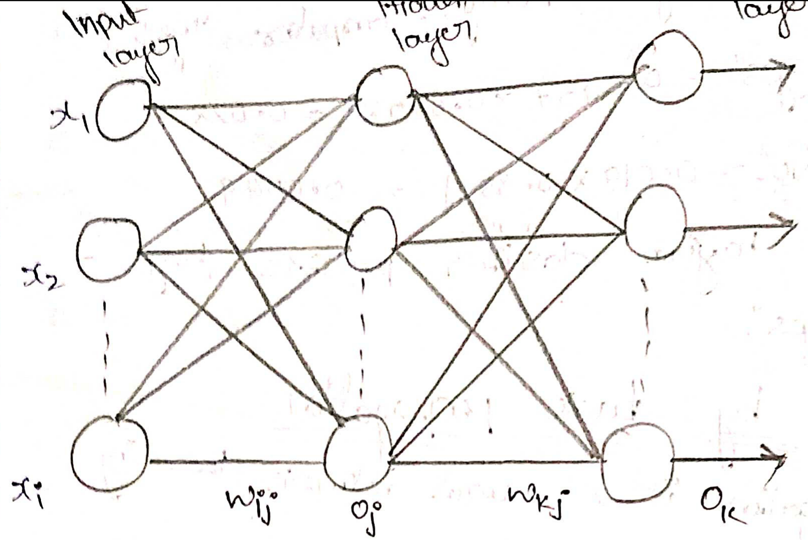
Therefore the bayesian classifier predicts buys - computer = "yes".

Classification by back propagation. → neural networks

* Back propagation is a neural network learning algorithm
* The field of neural network was original developed by psychologist and neurologist.
* A neural network is a set of connected i/p & o/p units, where each connection as a weight associated with it.
* during the learning phase the network learns by adjusting the weights so as to be able to predict the correct class lable of the input.

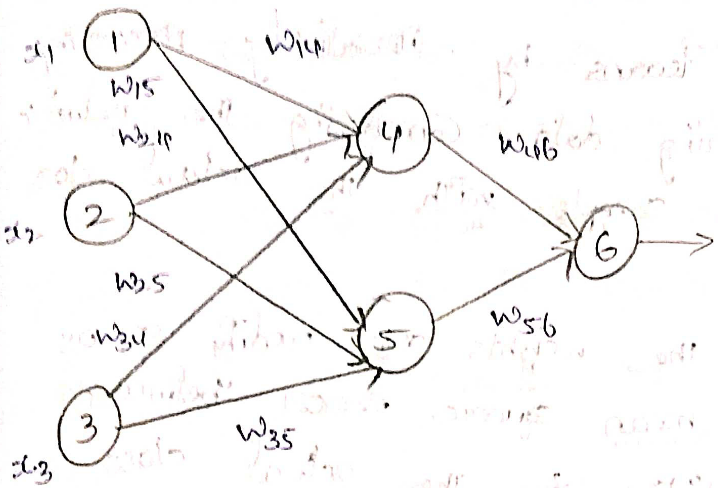
The multilayer feed forward Neural networks.

The backpropagation algorithm performs learning on a multilayer feed forward neural networks.



The units in the hidden layer and output layer are some times refered as ~~neu sene~~ neurodes.

* The multilayer neural network has two layer of output layer units. Therefore we say that it is a two layer neural network.
* Similarly a network containing two hidden layer is called a Three layer neural network.
* The network is feed forward in that non of the weights cycles back to an input units. or To an o/p unit of a previous layer
* It is fully connected in that each unit provides input to each unit in the next forward layer.



$x = (x_1, x_2, \ldots x_n)$

$I_1 = x_1 w_{11} + x_2 w_{21} + \ldots + x_i w_{i1} + \ldots + x_n w_{n1}$

$I_2 = x_1 w_{12} + x_2 w_{22} + \ldots + x_i w_{i2} + \ldots + x_n w_{n2}$

$I_3 = x_1 w_{13} + x_2 w_{23} + \ldots + x_i w_{i3} + \ldots + x_n w_{n3}$.

Hidden layer:-

$I_j = x_1 w_{1j} + x_2 w_{2j} + \ldots + x_n w_{nj}$

$I_j = \sum_{i=1}^{n} x_i w_{ij}$

output layer:-

$I_j = \sum_i^n o_i w_{ij}, \; j = 1, 2, \ldots$

$o_j = f(I_j) = \frac{1}{1+e^{-I_j}}$

Error = target o/p − actual o/p.

Backpropagation process:-

Backpropagation learns by iteratively processing a set of training data. Comparing the network prediction for each sample with the actual class label.

* For each training data the weights are modify so as to minimize the mean square error between the network prediction & the actual class.
* These modifications are made in the Backward direction. i.e. from the o/p layer through each hidden layer down to the first hidden layer or input layer. Hence the name is called backpropagation.

Backpropagation Algorithm:-

Neural network learning for classification, using the backpropagation algorithm.

Input:- The training samples; The learning rate, l; a multilayer feed forward network.

Output:- A neural Network trained to classify the samples.

Method:-

1. Initialize all the weights and biases in network
2. while terminating condition is not satisfied{
3. for each training sample X in samples{
4. // propagate the inputs forward:
5. for each hidden or output layer unit j{
6. $I_j = \sum_i w_{ij} O_i + \theta_j$; // compute the net input of unit j with respect to the previous layer, i.
7. $O_j = \frac{1}{1+e^{-I_j}}$; } // Compute the o/p of each unit j.
8. // Backpropagate the errors;
9. for each unit j in the output layer.
10. $Err_j = O_j(1-O_j)(T_j - O_j)$; // compute the error.
11. for each unit j in the hidden layers, from the last to the first hidden layer.
12. $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer k.
13. for each weight w_{ij} in network{
14. $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
15. $w_{ij} = w_{ij} + \Delta w_{ij}$ } // weight update.
16. for each bias θ_j in network{
17. $\Delta \theta_j = (l) Err_j$; // bias increment
18. $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
19. }}

Rule based classification model

If rule based classifier the trained model is represented as set of "IF - Then rules".

Rule based classification model $= \{R_1, R_2, \ldots R_n\}$

* if each R_i is if - Then rules.
* The general formate of an if - Then rule is
* if condition Then conclusion.

If part (or) Antecedent (or) precondition : Then part (or) consequent (or) post - condition

Ex:-

R: If age = youth And Income = high Then buys Computer = yes.

[or]

R: (age = youth) $\cap$ (income = high) $\Rightarrow$ buys computer = yes.

Measures:-

1. Coverage Rule
2. Accuracy rule

formules :-

* $\text{Coverage}(R) = \dfrac{n_{cover}}{|D|}$
* $\text{Accuracy}(R) = \dfrac{n_{correct}}{n_{covers}}$

Where $|D|$ = ~~Rarrav~~ no. of tuples in the Dataset D.

n_{cover} = no. of tuples covered by R

$n_{correct}$ = no. of tuples corrected by R.

Eg:- R: If age = youth And student = yes then buys computer = yes.

* $\text{Coverage}(R) = \dfrac{n_{cover}}{|D|} = \dfrac{2}{14} = 0.1428 = 14.28\%$
* $\text{Accuracy}(R) = \dfrac{n_{correct}}{n_{covers}} = \dfrac{2}{2} = 1 = 100\%$

Rule Based classification Algorithm: [OR] Rule Induction Algorithm

Input:- A Dataset D consisting of Tuples and They associated with class labels [training data set]

O/P:- A set of IF- Then rules.

Method:-

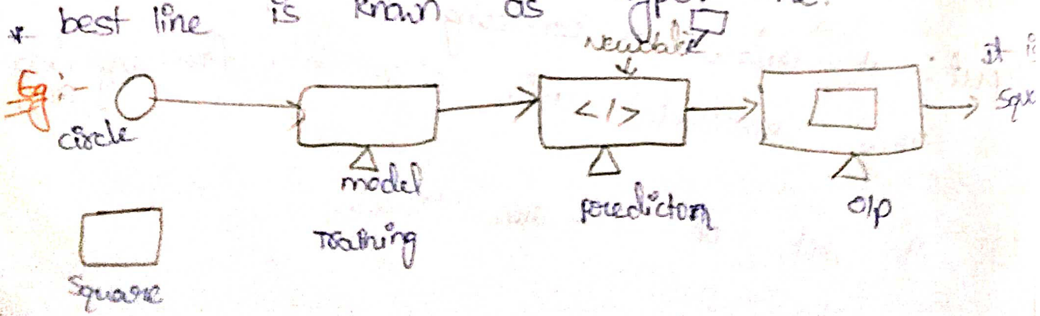
1. Initialize Rule set = { }
2. for each class 'c' do
3. Repeat
4. Rule = learn - one - Rule (D, attribute value , c)
5. Remove the tuple covered by rule from D.
6. until terminating condition
7. Rule set = Rule set + Rule
8. Return Rule set.

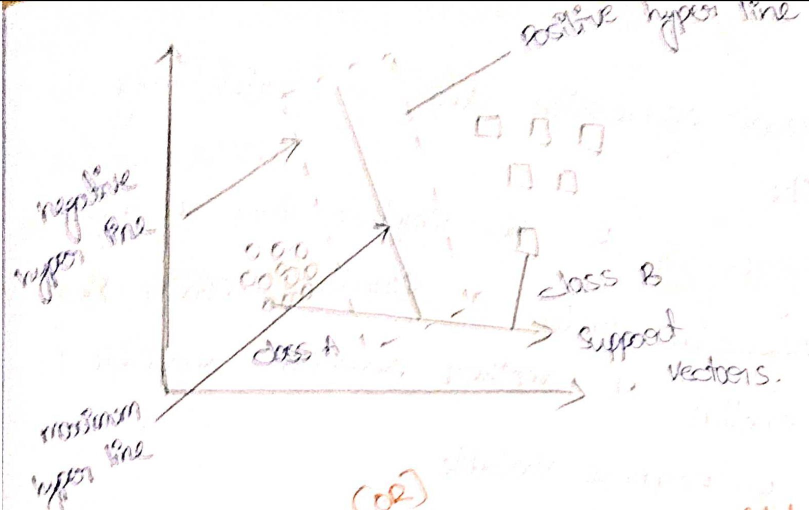
Support vector mechine :- (SVM)

SVM is the one of the most popular super-visor learning algorithm.

* SVM is used for classification as well as regression problem.
* SVM mainly used for classification problems.
* the main goal of the SVM algorithm is to create the best line
* best line is known as hyper line.

Eg:-





(OR)

* lazy learner:- learning from neighbours.

learning is classified into two types

① eager learning
② lazy learning.

* lazy learnering:-

Simply stores training data and wait until it get's a test tuple.

* It works only when it gets a new example

eager	lazy learning
* less predict time	* less training time in lazy
* more training time.	* more predict time
	* KNM - K-nearest neighbour model

Linear Regression :-

In linear regression data ∞ model using a straight line.

* linear regression is the simplest form of Regression
* linear regression model a random variable Y, as a linear function of another random variable X.

(*) where y = Response Variable

x = predict variable.

$$\boxed{Y = \alpha + \beta X}$$

where α, β are regression coefficients.

(*) These coefficients can be solved for by the method of least squares. which minimize the error b/w the actual data & the estimate of the line.

$$\beta = \frac{\sum_{i=1}^{s} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s} (x_i - \bar{x})^2}$$

$$\alpha = \bar{Y} - \beta \bar{X}$$

where $\bar{x}$ is the avg of x and $\bar{y}$ is the avg of y.

Problem: linear regression using the method of least square.

Salary data:

x years experience		y salary (in \$ 1000s)
3	x_1	30
8	x_2	57
9	x_3	64
13	x_4	72
3	x_5	36
6	x_6	43
11	x_7	59
21	x_8	90
1		20
16		83

Now find the $\bar{x}$

$x = 10 \Rightarrow y$

$\bar{x}$ means avg of x

$$\bar{x} = \frac{3+8+9+13+3+6+11+21+1+16}{10}$$

$\bar{x} = 9.1$

$\bar{y} = 55.4$

$\alpha = 23.6$

$\beta = 3.5$

$$\bar{x} = \frac{91}{10}$$

$$\bar{x} = 9.1$$

findout The $\bar{y}$

$$\bar{y} = \frac{30+57+64+72+36+43+59+90+20+83}{10}$$

$$\bar{y} = \frac{554}{10}$$

$$\bar{y} = 55.4$$

Now findout The β value

$$\beta = \frac{\sum_{i=1}^{s} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s} (x_i - \bar{x})^2}$$

step -1

x_1 x_2 x_3

$$= \frac{(3-9.1)(30-55.4)}{(3-9.1)^2} + \frac{(8-9.1)(57-55.4)}{(8-9.1)^2} + \frac{(9-9.1)(64-55.4)}{(9-9.1)^2}$$

$$+ \frac{(13-9.1)(72-55.4)}{(13-9.1)^2} + \frac{(3-9.1)(36-55.4)}{(3-9.1)^2} + \frac{(6-9.1)(43-55.4)}{(6-9.1)^2}$$

$$+ \frac{(11-9.1)(59-55.4)}{(11-9.1)^2} + \frac{(21-9.1)(90-55.4)}{(21-9.1)^2} + \frac{(1-9.1)(20-55.4)}{(1-9.1)^2}$$

$$\frac{(16-9.1)(83-55.4)}{(16-9.1)^2}$$

step :-2

$$= \frac{(-6.1)(-25.4)}{\underbrace{(3-9.1)^2}_{\text{value } 37.21}} + \frac{(-1.1)(1.6)}{1.21} + \frac{(-0.1)(8.6)}{0.01}$$

$$+ \frac{(3.9)(16.6)}{15.21} + \frac{(-6.1)(-19.4)}{37.21} + \frac{(-3.1)(-12.4)}{9.61} +$$

$$\frac{(1.9)(3.6)}{3.61} + \frac{(11.9)(34.6)}{141.61} + \frac{(-8.1)(-35.4)}{65.6} +$$

$$\frac{(6.9)(27.6)}{47.61}$$

step 3:

$$= \frac{154.94 + (-1.76) + (-0.86) + 64.74 + 118.34}{37.21 + 1.21 + 0.01 + 15.21 + 37.21}$$

$$= \frac{38.44 + 6.84 + 411.74 + 286.74 + 190.44}{9.61 + 3.61 + 141.61 + 65.6 + 47.61}$$

$$= \frac{1269.6}{358.9}$$

$$\beta = 3.5375 = 3.5$$

$$\alpha = \bar{y} - \beta\bar{x}$$

$$= 55.4 - (3.5)(9.1)$$

$$= 55.4 - 31.85$$

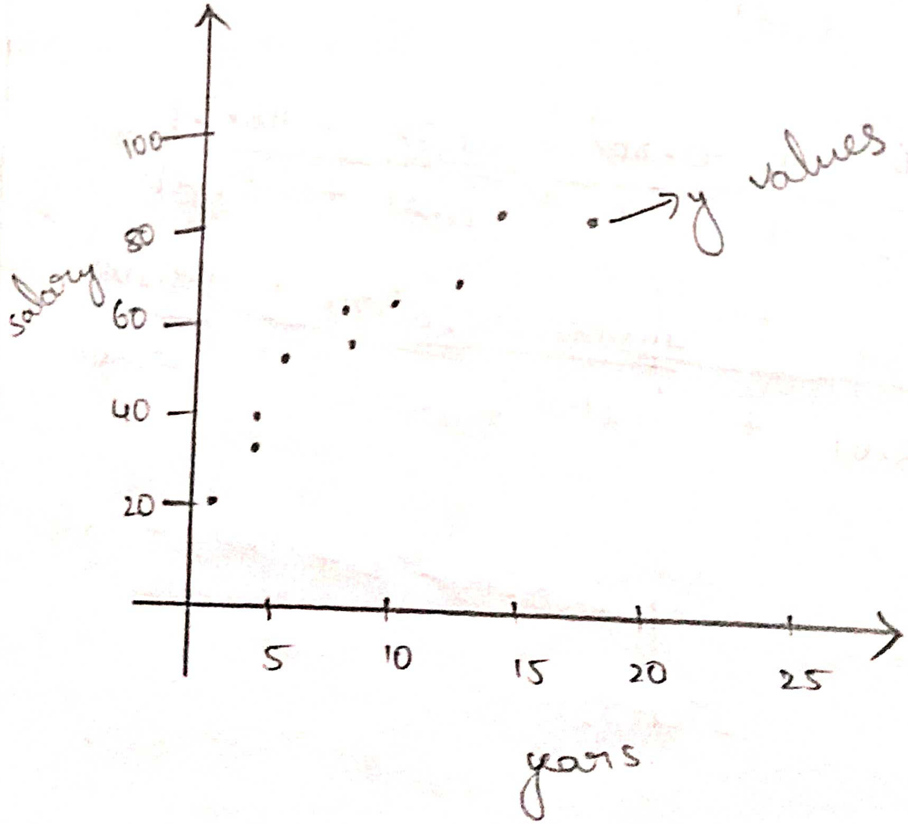
$$\alpha = 23.55$$

$$\alpha = 23.6.$$

$y = 23.6 + (3.5)(10)$

$y = 23.6 + 35$

$y = 58.6.$



Differences b/w classification and prediction :-

classification	prediction.
Accuracy :- The accuracy of the classifier refers to the ability of a given classifier to correctly predict the class lable of new data.	Accuracy :- The accuracy of a predi-ctor refers to How well a given predictor can guess The value of the predicted attribute for new data
speed :- This refers to The computational cost involved in generating and using The given classifier	This refers to The computional cost involved in generating and using The given predictor.
Scalability :- This refers to The ability of construct The classifier efficiently given large amount of data.	This refers to The ability of construct the Predictor efficiently given large amount of data.
Interopertability :- This refers to The levels of understanding and insite That is provided by The classifier.	This refers to The levels of understanding and insite That is provided by The predictor.

Unit -5

What is cluster Analysis

- The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering.
- A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.
- A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.
- Cluster analysis tools based on k-means, k-medoids, and several methods have also been built into many statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS.

Requirements Of Clustering In Data Mining:

Scalability:

Many clustering algorithms work well on small data sets containing fewer than several hundred data objects; however, a large database may contain millions of objects. Clustering on a sample of a given large data set may lead to biased results.

Highly scalable clustering algorithms are needed.

Ability to deal with different types of attributes:

Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.

Discovery of clusters with arbitrary shape:

Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density.

However, a cluster could be of any shape. It is important to develop algorithms thatcan detect clusters of arbitrary shape.

Minimal requirements for domain knowledge to determine input parameters:

Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). The clustering results can be quite sensitive to input parameters. Parameters are often difficult to determine, especially for data sets containing high dimensional objects. This not only burdens users, but it also makes the quality of clustering difficult to control.

Ability to deal with noisy data:

Most real-world databases contain outliers or missing, unknown, or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

High dimensionality:

A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions. Human eyes are good at judging the quality of clustering for up to three dimensions. Finding clusters of data objects in high dimensional space is challenging, especially considering that such data can be sparse and highly skewed.

Types of data in cluster analysis:

Imp (10M) Types of Data in cluster Analysis:

Data structure:-

Data structures are two types.

① Data matrix:- This represents n objects, such as persons, with p variables such as age, height, weight, gender, race, and so on. The structure is in the form of a relational table, or n-by-p matrix (n objects × p variables):

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Dissimilarity matrix :-

This stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n by n table:

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ \vdots & \vdots & \vdots & \\ d(n,1) & d(n,2) & \cdots & 0 \end{bmatrix}$$

where $d(i,j)$ is the measured difference or dissimilarity b/w objects i and j. In general, $d(i,j)$ is a nonnegative number that is close to 0 when objects i and j are highly similar or "near".

Interval - Scaled variables :-

Interval - Scaled variables are continuous measurements of a roughly linear scale. Typical examples include weight and height, latitude and longitude coordinates (Eg:- when clustering houses), and weather temperature.

① Calculate the mean absolute deviation :

$$S_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|\right)$$

where $x_{1f}, \ldots x_{nf}$ are n measurements of f, and m_f is the mean value of f, that is,

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \cdots + x_{nf}\right)$$

2. Calculate the standardized measurement, (or) z - Score.

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

The mean absolute deviation, s_f, is more robust to outliers than the standard deviation, σ_f. When computing the mean absolute deviation the deviations from the mean (i.e., $|x_{if} - m_f|$) are not squared.

The most popular distance measure is Euclidean distance which is defined as

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2}$$

where $i = (x_{i1}, x_{i2} \cdots x_{ip})$ and $j = (x_{j1}, x_{j2}, \cdots, x_{jp})$ are two p-dimensional data objects.

Another well-known metric is Manhattan distance defined as

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|.$$

Both the Euclidean distance & manhattan distance satisfy the following mathematic requirements of a distance function.

① $d(i,j) \geq 0$: Distance is a non-negative number.

② $d(i,i) = 0$: The distance of an object to itself is 0

③ $d(i,j) = d(j,i)$: distance is a symmetric function

iii. $d(i,j) \leq d(i,h) + d(h,j)$: Going directly from object i to object j in space is no more than other object h.

Minkowski distance:-

MD is a generalization of both Euclidean distance and Manhattan distance. It is defined as

$$d(i,j) = (|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q)^{1/q}$$

where q is a positive integer. It represents the Manhattan distance when q = 1, and Euclidean distance when q = 2.

Binary Variables:-

A Binary variable has only two states. The variable is 0 or 1, where 0 means that it is absent and 1 means that it is present. Given the variable smoker describing a patient, for instance, 1 indicates that the patient smokes while 0 indicates that the patient does not. Treating binary variables as if they are interval-scaled can lead to misleading distorting results.

I) Symmetric:-

A binary variable is symmetric if both of its states are equally valuable and carry the same weight.

② asymmetric:-

A binary variable is asymmetric if the outcomes of the states are not equally important, such as the positive & negative outcomes of a disease test.

③ Nominal, ordinal variables:

Nominal variables:-

A Nominal variable is a generalization of the binary variable in that it can take on more than two states.

Eg:- map color is a nominal variable that may have, say, five states: red, yellow, green, pink & blue.

The dissimilarity b/w two objects i and j can be computed using the simple matching approach

$$d(i,j) = \frac{p-m}{p}$$

∴ m is the no. of matches

p is the total number of variables

Ordinal variables:-

A discrete ordinal variable resembles a nominal variable, except that the M states of the ordinal value are ordered in a meaningful sequence.

ordinal variables are very useful for registering subjective assessments of qualities that cannot ... ... objectively.

A continuous ordinal variable looks like a set of continuous data of an unknown scale.

Variables of Mixed Types.

Suppose that the data set contains p variables of mixed type. The dissimilarity $d(i,j)$ b/w objects i and j is defined as

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

The Indicator $\delta_{ij}^{(f)} = 0$ if either ① x_{if} or x_{jf} is missing. ② $x_{if} = x_{jf} = 0$ and variable f is asymmetric binary.

① If f is binary or nominal:

$d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise $d_{ij}^{(f)} = 1$

* If f is interval-based: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$

where h runs over all non missing objects for variable f.

* If f is ordinal or ratio-scaled: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as interval-scaled

Minkowshi distance:

$$d(i,j) = \left(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \cdots + |x_{ip} - x_{jp}|^q\right)^{1/q}$$

where q is a +ve integer. It represents the manhatton distance when $q=1$ & Euclidean distance when $q=2$.

Problem:-

given Two objects represented by tuples (22, 1, 42, 10) and (20, 0, 36, 8).

(i) Compute The Eucliden distance b/w Two objects

(ii) Compute The manhattan " " " "

(iii) Compute The minkowshi distance b/w Two objects using $q=3$.

let

$$X = (22, 1, 42, 10)$$

$$Y = (20, 0, 36, 8)$$

$$\text{Euclidean}(i,j) = \sqrt{|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \cdots + |x_{ip} - x_{jp}|^2}$$

$$= \sqrt{|22-20|^2 + |1-0|^2 + |42-36|^2 + |10-8|^2}$$

$$= \sqrt{|2|^2 + |1|^2 + |6|^2 + |2|^2}$$

$$= \sqrt{4+1+36+4}$$

$= \sqrt{45}$

$= 6.708.$

Manhattan $= |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$

$= |22-20| + |1-0| + |42-36| + |10-8|$

$= |2| + |1| + |6| + |2|$

$= 2+1+6+2$

$= 11$

Minkowski $= \left(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q\right)^{1/q}$

Here $q=3.$

$= \left(|22-20|^3 + |1-0|^3 + |42-36|^3 + |10-8|^3\right)^{1/3}$

$= \left(|2|^3 + |1|^3 + |6|^3 + |2|^3\right)^{1/3}$

$= (8+1+216+8)^{1/3}$

$= (233)^{1/3}$

$= 6.153.$

Binary Variables:-

Binary variables are. Two types.

1. Symmetric Variable.

does not change The values.

Eg:- Covid $<$ positive / Negative.

Partitioning Methods

- A partitioning method constructs *k* partitions of the data, where each partition represents a cluster and *k* <= *n*. That is, it classifies the data into *k* groups, which together satisfy the following requirements:

- Each group must contain at least one object, and
- Each object must belong to exactly one group.

- A partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another.

- The general criterion of a good partitioning is that objects in the same cluster are close or related to each other, whereas objects of different clusters are far apart or very different.

The *k*-Means Method

- The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low.
- Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity.

The *k*-means algorithm proceeds as follows.

- First, it randomly selects *k* of the objects, each of which initially represents a cluster mean or center.
- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
- It then computes the new mean for each cluster.
- This process iterates until the criterion function converges.

Typically, the square-error criterion is used, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2,$$

Where E is the sum of the square error for all objects in the data set p is the point in space representing a given object M_i is the mean of cluster C_i

The k-means partitioning algorithm:

The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

k: the number of clusters,

D: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
(2) **repeat**
(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) **until** no change;

Hierarchical Methods:

- A hierarchical clustering method works by grouping data objects into a tree of clusters.
- The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed. That is, if a particular merge or split decision later turns out to have been a poor choice, the method cannot backtrack and correct it.
- Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion.

Agglomerative hierarchical clustering:

- This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.
- Most hierarchical clustering methods belong to this category. They differ only in their definition of inter cluster similarity.

Divisive hierarchical clustering:

- This top-down strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster.
- It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster

on its own or until it satisfies certain termination conditions, such as a desired number of clusters is obtained or the diameter of each cluster is within a certain threshold.

Density-based methods:

- Most partitioning methods cluster objects based on the distance between objects.
- Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes.
- Other clustering methods have been developed based on the notion of density.
- Their general idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
- Such a method can be used to filter out noise (outliers)and discover clusters of arbitrary shape.
- DBSCAN and its extension, OPTICS, are typical density-based methods that grow clusters according to a density-based connectivity analysis.
- DENCLUE is a method that clusters objects based on the analysis of the value distributions of density functions.

DBSCAN: Density-Based Spatial Clustering of Applications with Noise

The DBSCAN algorithm uses two parameters:

Eps: It is considered as the maximum radius of the neighbourhood.

MinPts: Minimum number of data points insides the circle

These parameters can be understood if we explore two concepts called Density Reachability and Density Connectivity.

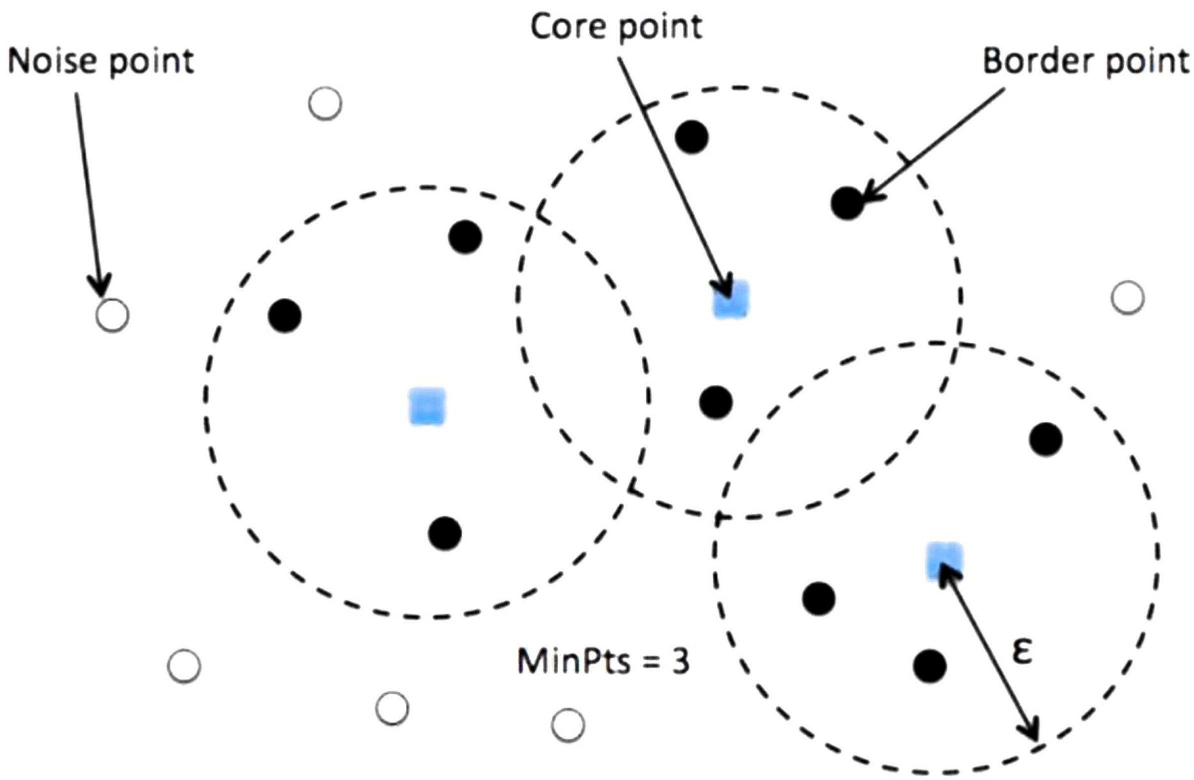
- **Reachability** in terms of density establishes a point to be reachable from another if it lies within a particular distance (eps) from it.
- **Connectivity**, on the other hand, involves a transitivity-based chaining-approach to determine whether points are located in a particular cluster. For example, p and q points could be connected if p->r->s->t->q, where a->b means b is in the neighborhood of a.

There are three types of points after the DBSCAN clustering is complete:

Core — This is a point that has at least m points within distance n from itself.

Border — This is a point that has at least one Core point at a distance n.

Noise — This is a point that is neither a Core nor a Border. And it has less than m points within distance n from itself.

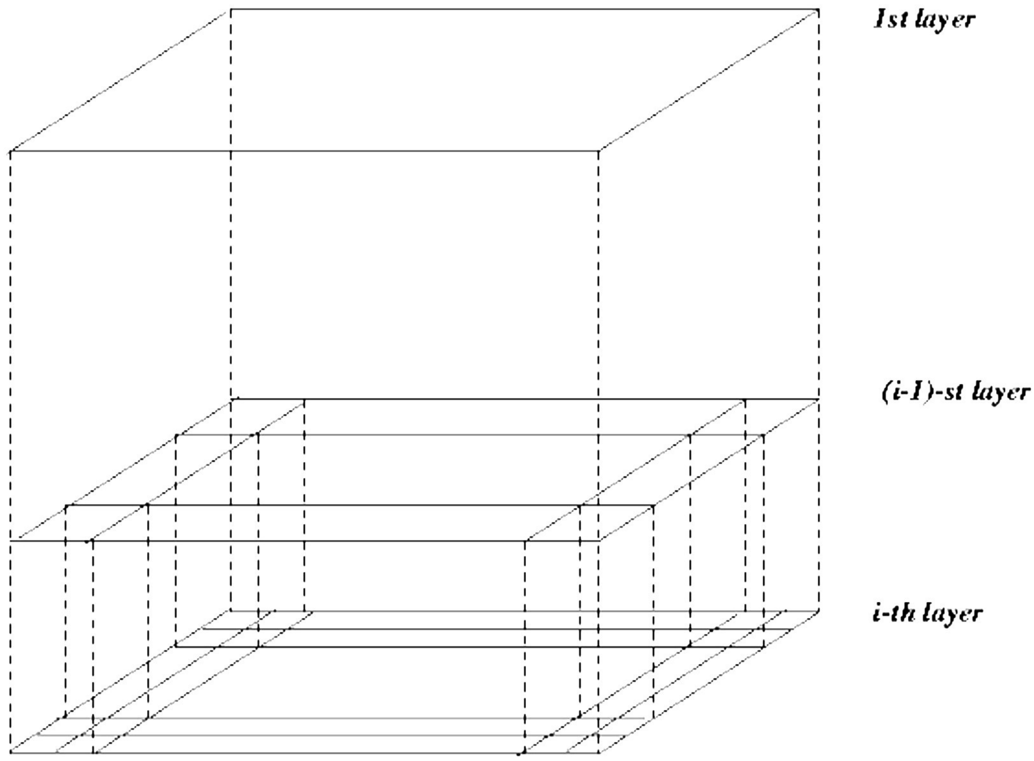


Grid-Based Methods:

- Grid-based methods quantize the object space into a finite number of cells that form a grid structure.
- All of the clustering operations are performed on the grid structure i.e., on the quantized space. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.
- STING is a typical example of a grid-based method. Wave Cluster applies wavelet transformation for clustering analysis and is both grid-based and density-based.

STING - A Statistical Information Grid

- Spatial area is divided into rectangular cells. There are several levels of cells corresponding to different levels of resolution, and these cells are forming a hierarchical structure (or) tree structure.



- For each cell, the high level is partitioned into several smaller cells in the next lower level.
- The statistical info of each cell is calculated and stored beforehand and is used to answer queries.
- The parameters of higher-level cells can be easily calculated from parameters of lower-level cell
  - Count, mean, s, min, max
  - Type of distribution—normal, uniform, etc.
- Calculation of these parameters should starts at root and go down till bottom layer.