Index

Module 1: DWH and Data Mining

1. What is DWH? Explain DWH characteristics.

- A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process.
- A data warehouse is a large collection of business data used to help an organization make decisions.
- A system used for reporting and data analysis; a core component of business intelligence.
- DWs are central repositories of integrated data from one or more disparate sources. They store current and historical data in one single place .

- **Advantages**:
  - Data warehouse allows users to access critical data from the number of sources in a single place. Therefore, it saves user's time of retrieving data from multiple sources.
  - Data warehouse provides consistent information on various cross-functional activities. It also supports ad-hoc reporting and query.
  - Data Warehouse helps to integrate many sources of data to reduce stress on the production system.
  - Data warehouse helps to reduce total turnaround time for analysis and reporting.
  - Restructuring and Integration make it easier for the user to use for reporting and analysis.
  - Data warehouse stores a large amount of historical data. This helps users to analyze different time periods and trends to make future predictions.
  - Immediate information delivery
  - Integration of data from within and outside the organization
  - Provides an insight into the future

- **Characteristics**:

1. **Subject-oriented**:
   - A data warehouse is organized around major subjects .
   - Subject-oriented means data is organized by business topic, not by the business process.
   - A data warehouse focuses on the modeling and analysis of data for decision makers.
   - Hence, data warehouses typically provide a simple and concise view of particular subject issues.

2. **Integrated**:
   - A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and online transaction records.
   - Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.

3. **Time-variant:**
   - Time variant means that the time dimension is explicitly included in the data so that trends and changes over time can be studied.
   - It simply means that the data available was identified on a particular period.
   - Data is stored to provide information from an historic perspective (e.g., the past 5–10 years).
   - Every key structure in the data warehouse contains, either implicitly or explicitly, a time element.
   - Time is a critical factor for comparative analysis.
   - Advantages: 1. Allows for analysis of past; 2. Relates information to the present; 3. Enables forecasts for the future

4. **Nonvolatile:**
   - The data in the data warehouse is read-only, which means it cannot be updated or deleted (unless there is a regulatory or statutory obligation to do so).

2. What are the advantages and applications of DWH?

(Basic introduction and advantages are same as above answer)

Sector	Usage
Airline	It is used for airline system management operations like crew assignment, analysis of route, frequent flyer program discount schemes for passenger, etc.
Banking	It is used in the banking sector to manage the resources available on the desk effectively.
Healthcare sector	Data warehouse used to strategize and predict outcomes, create patient's treatment reports, etc. Advanced machine learning, big data enable data warehouse systems to predict illness.
Insurance sector	Data warehouses are widely used to analyze data patterns, customer trends, and to track market movements quickly.
Retain chain	It helps you to track items, identify the buying pattern of the customer, promotions and also used for determining pricing policy.
Telecommunication	In this sector, data warehouse is used for product promotions, sales decisions and to make distribution decisions.

3. Why is the ER model not suitable for DWH? What are the steps in dimensional modeling?

- A dimensional model contains same information as ER model but packages the data in a symmetric format whose design goals are easy understandability, query performance, and resilience to change.
- A dimensional model in data warehouse is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse.

- In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System (OLTP).
- These dimensional and relational models have their unique way of data storage that has specific advantages.
- ER modelling aims to optimize performance for transaction processing. It is also hard to query ER models because of the complexity; therefore ER models are not suitable for high performance retrieval of data.

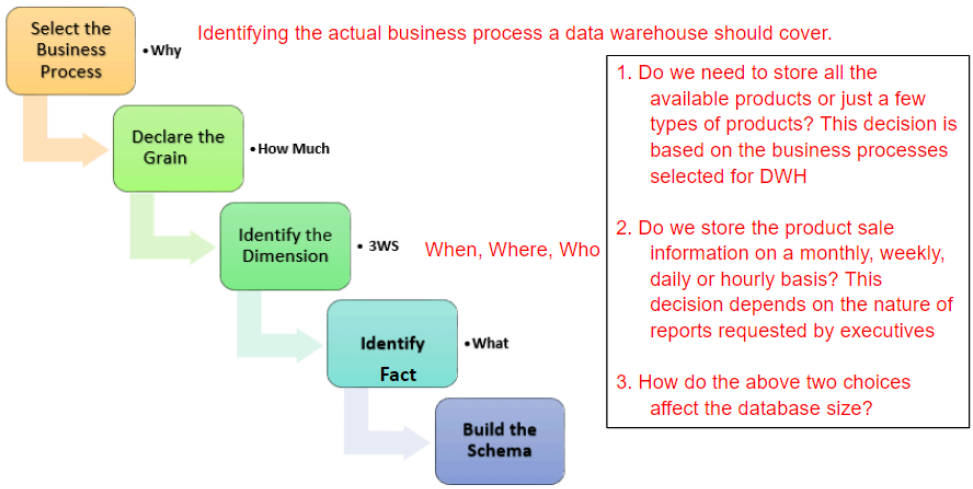
OR

(co-pilot AI)

1. **Complexity and Query Performance**:
    - ER models are optimized for transaction processing , where data is frequently added, updated, or deleted. In such systems, ER models work well.
    - However, data warehouses serve a different purpose. They are designed for analytical queries and reporting, where retrieving large amounts of data efficiently is crucial.
    - ER models can become complex due to the need for many joins between tables.
    - These joins can significantly impact query performance, making ER models less suitable for high-speed data retrieval in data warehouses
2. **Dimensional Modeling vs. ER Modeling**:
    - Dimensional modeling (DM) is specifically tailored for data warehouses.
    - DM aims to optimize the database for faster retrieval of data. It arranges data in a way that facilitates analysis and reporting.
    - In contrast, ER models focus on normalization and reducing redundancy, which is essential for transactional systems but not ideal for data warehouses.
3. **Advantages of Dimensional Modeling**:
    - Efficient Retrieval: DM allows faster retrieval of summarized data, making it suitable for reporting and analytics.
    - Simplicity: DM simplifies complex relationships and reduces the need for joins.
    - Aggregation: Fact tables in DM already contain aggregated data, reducing the need for additional calculations during queries.

In summary, while ER models excel in transactional systems, dimensional modeling is the preferred choice for designing data warehouses due to its focus on efficient data retrieval and analytical needs.

Steps of dimensional modeling



Step 1. **Define the business process**:

- Define the business process you want to track, which could be something as simple as sales data or something more complicated such as inventory data.
- Since it is the most important step of Data Modelling, the selection of business objectives also depends on the quality of data available for that process.

Step 2. **Declare the grain**:

- Declare the grain, the smallest data unit you want to track. For example, if you are tracking sales data, the grain might be a single sale.
- Granularity is the lowest level of information stored in the table. The level of detail for business problems and its solution is described by Grain.

Step 3. **Identify the dimension tables**:

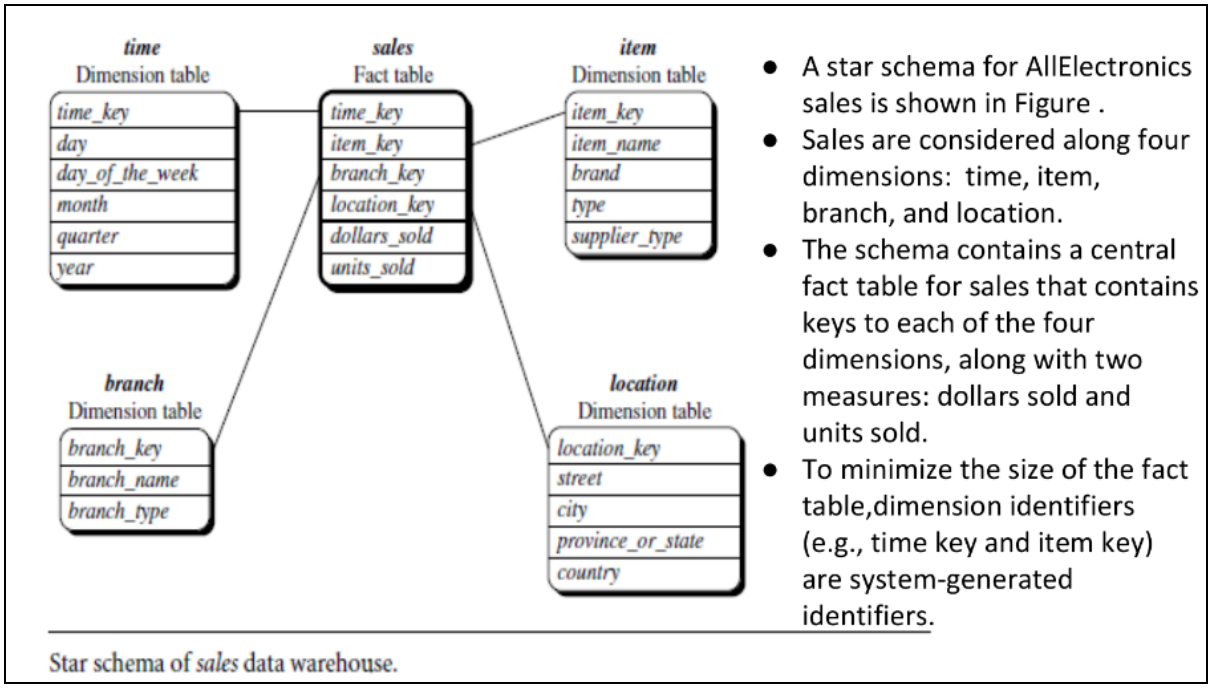
- Dimensions categorize and describe data warehouse facts and measures in a way that supports meaningful answers to business questions. A data warehouse organizes descriptive attributes as columns in dimension tables.
- Identify the dimension tables containing information about the entities involved in the business process. For example, a dimension table for sales data might have information about customers, products, and employees.

Step 4. **Identify the facts**:

- The measurable data is held by the fact table.
- Identify the facts and numerical data you want to track. For example, in a sales data set, the facts might be the quantity of products sold and the total sales price

4. Define dimension, fact , fact table and dimension table with example.

- A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts/measures.
- Dimensions are the perspectives or entities with respect to which an organization wants to keep records.
- Each dimension may have a table associated with it, called a dimension table, which further describes the dimension.
- A multidimensional data model is typically organized around a central theme, such as sales. This theme is represented by a fact table.
- Facts are numeric measures (quantities by which we want to analyze relationships between dimensions).
- The fact table contains the names of the facts, or measures, as well as keys to each of the related dimension tables.



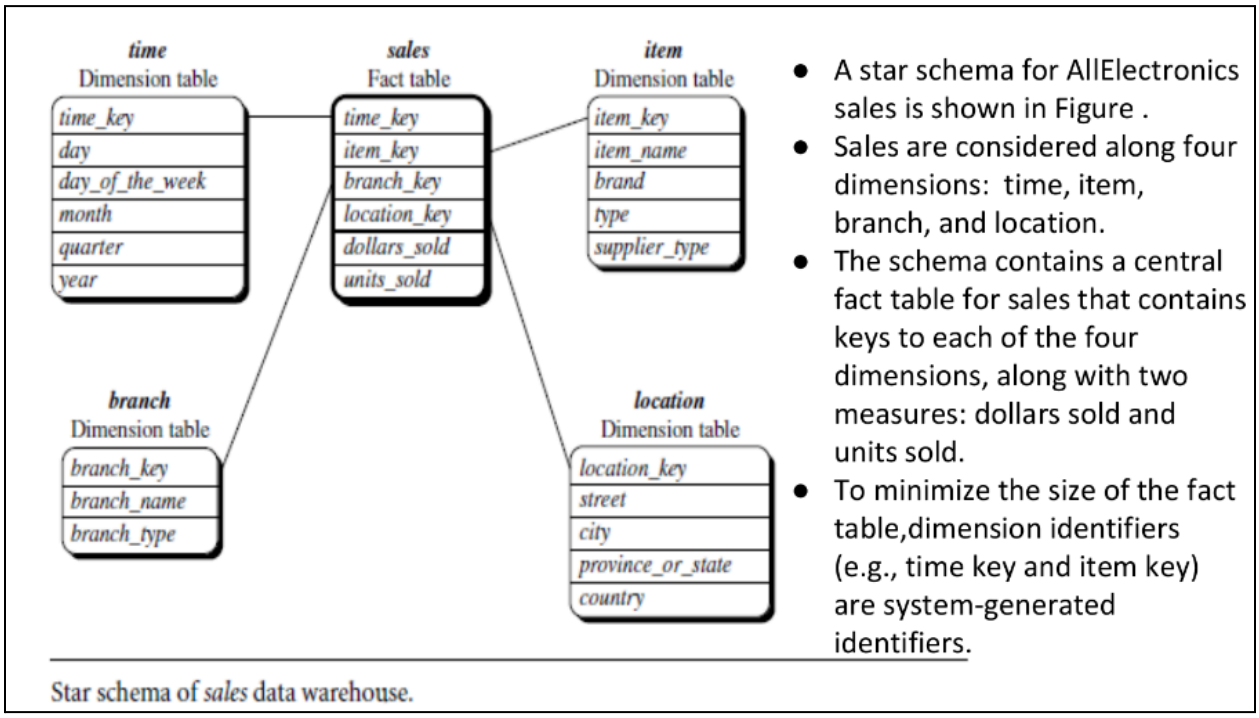
(same as Q6)

5. Difference between star and snowflake schema.

Star Schema	Snowflake Schema
Hierarchies for the dimensions are stored in the dimensional table.	Hierarchies are divided into separate tables.
It contains a fact table surrounded by dimension tables.	One fact table surrounded by dimension table which are in turn surrounded by dimension table
In a star schema, only single join creates the relationship between the fact table and any dimension tables.	A snowflake schema requires many joins to fetch the data.
Simple DB Design.	Very Complex DB Design.
Denormalized Data structure and query also run faster.	Normalized Data Structure.
High level of Data redundancy	Very low-level data redundancy
Single Dimension table contains aggregated data.	Data Split into different Dimension Tables.
Cube processing is faster.	Cube processing might be slow because of the complex join.
Offers higher performing queries using Star Join Query Optimization. Tables may be connected with multiple dimensions.	The Snowflake schema is represented by centralized fact table which is unlikely connected with multiple dimensions.

6. Design star and snowflake schema for given system.

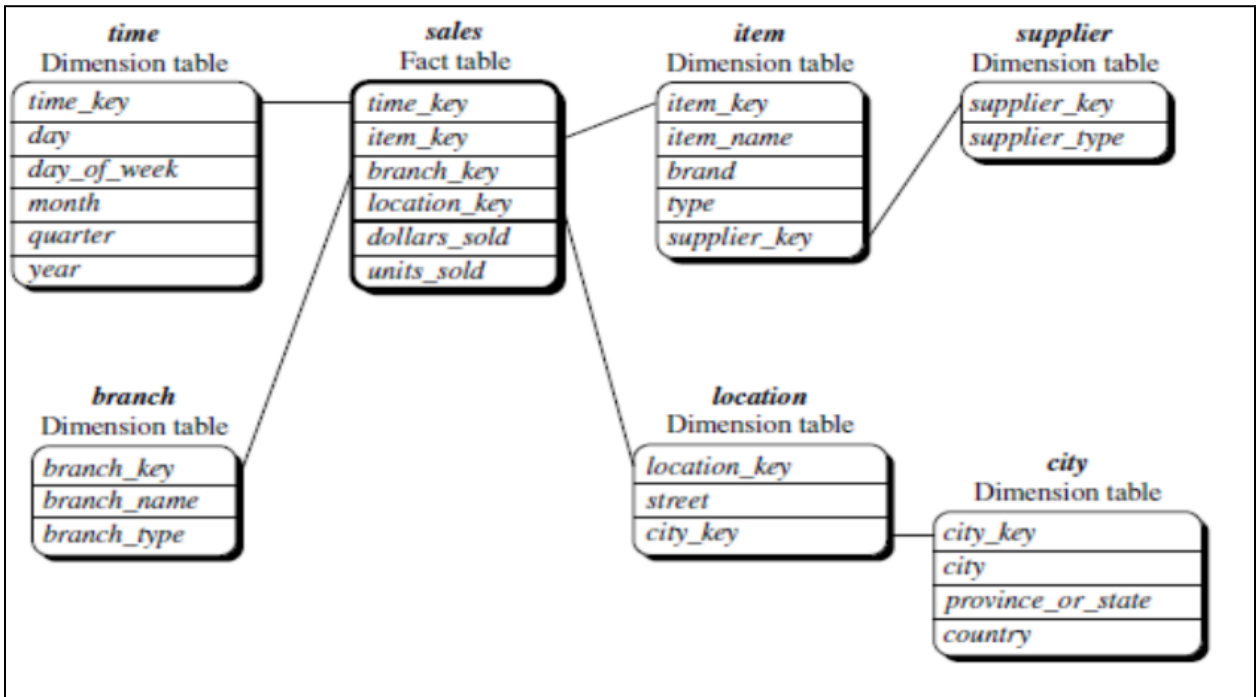
a. Star Schema Example of *Sales Data Warehouse*-



- A star schema for AllElectronics sales is shown in Figure .
- Sales are considered along four dimensions: time, item, branch, and location.
- The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold.
- To minimize the size of the fact table,dimension identifiers (e.g., time key and item key) are system-generated identifiers.

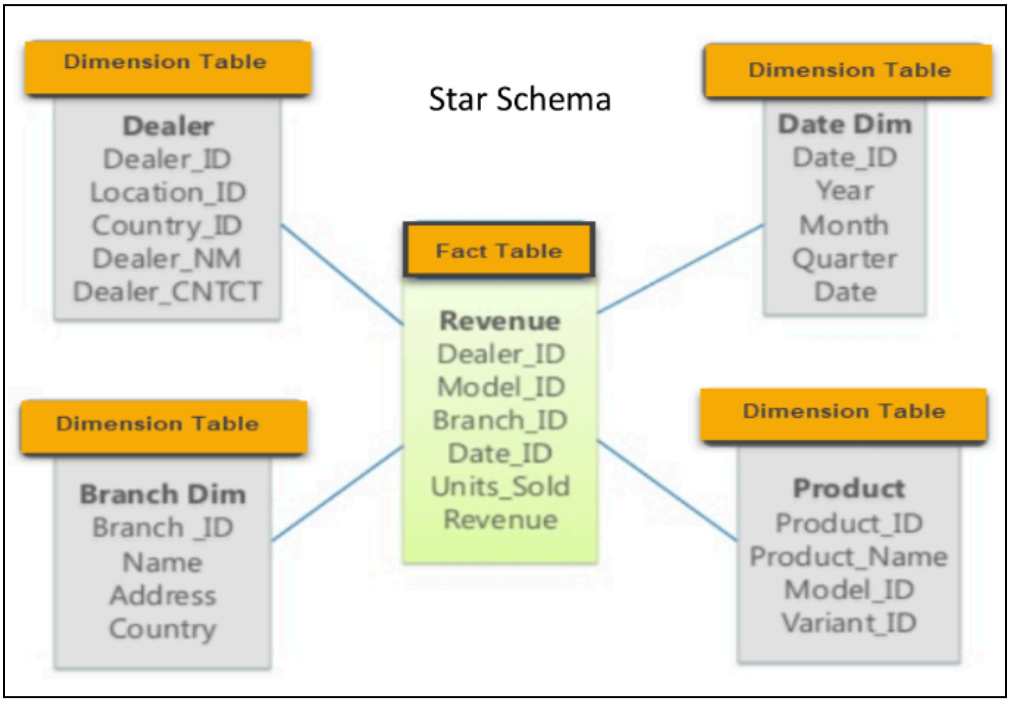
Star schema of *sales* data warehouse.

Snowflake Schema Example of *Sales Data Warehouse*-

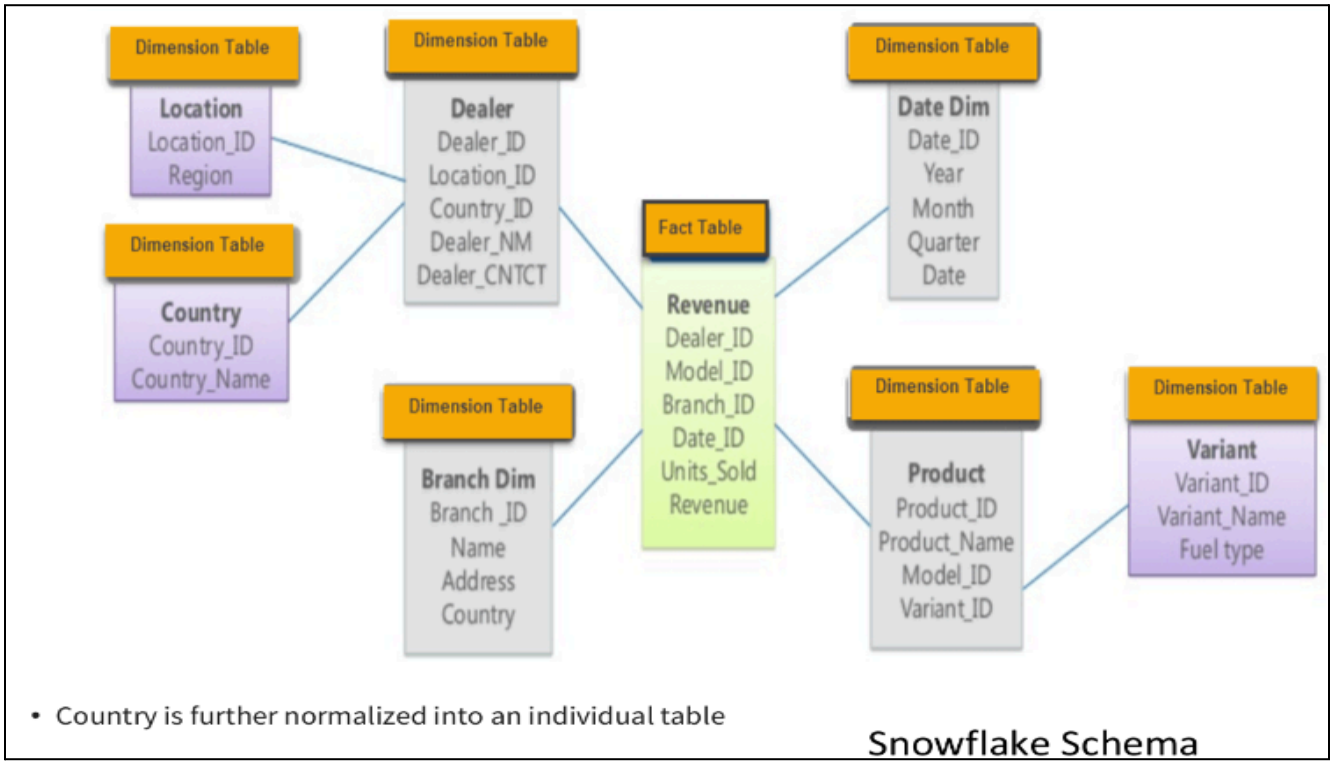


Snowflake schema of a *sales* data warehouse.

b. Star Schema Example of *Revenue Data Warehouse*-



Snowflake Schema Example of *Revenue Data Warehouse*-



c. MU question-

The Mumbai university wants you to help design a star schema to record grades for course completed by students. There are four dimensional tables namely course_section, professor, student, period with attributes as follows :

Course_section Attributes: Course_Id, Section_number, Course_name, Units, Room_id, Roomcapacity. During a given semester the college offers an average of 500 course sections

Professor Attributes: Prof_id, Prof_Name, Title, Department_id, department_name

Student Attributes: Student_id, Student_name, Major. Each Course section has an average of 60 students

Period Attributes: Semester_id, Year. The database will contain Data for 30 months periods. The only fact that is to be recorded in the fact table is course Grade

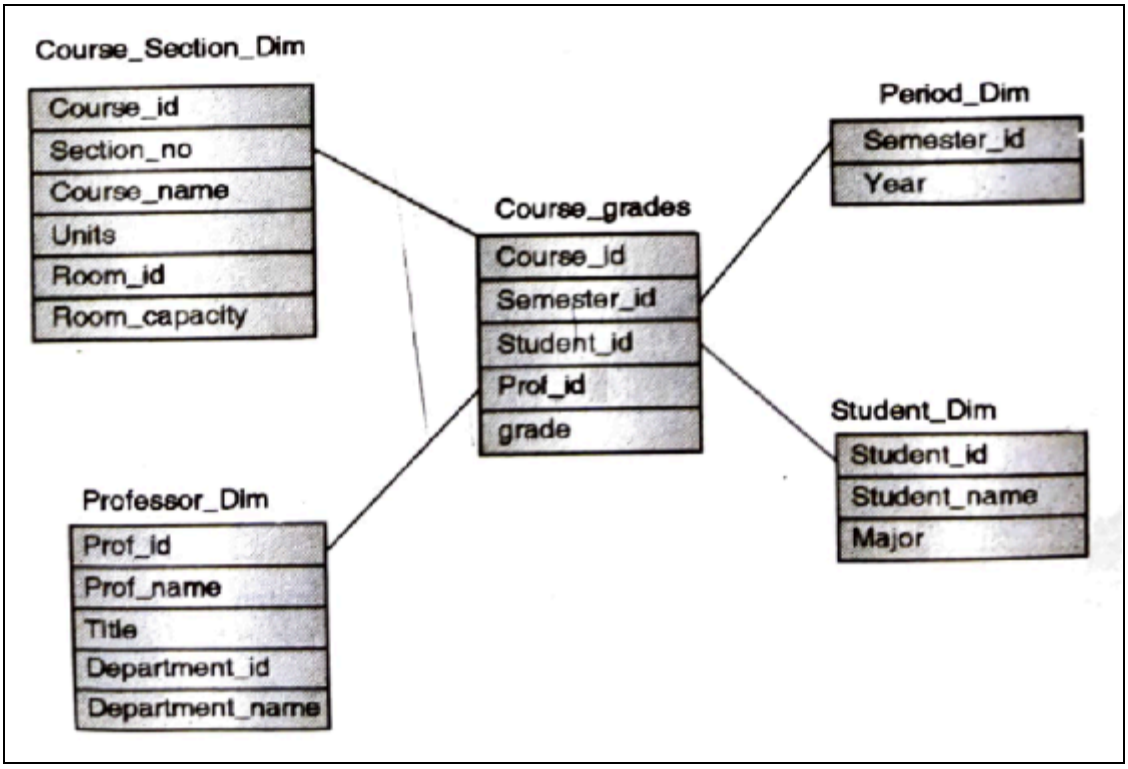
Answer the following Questions

(a) Design the star schema for this problem

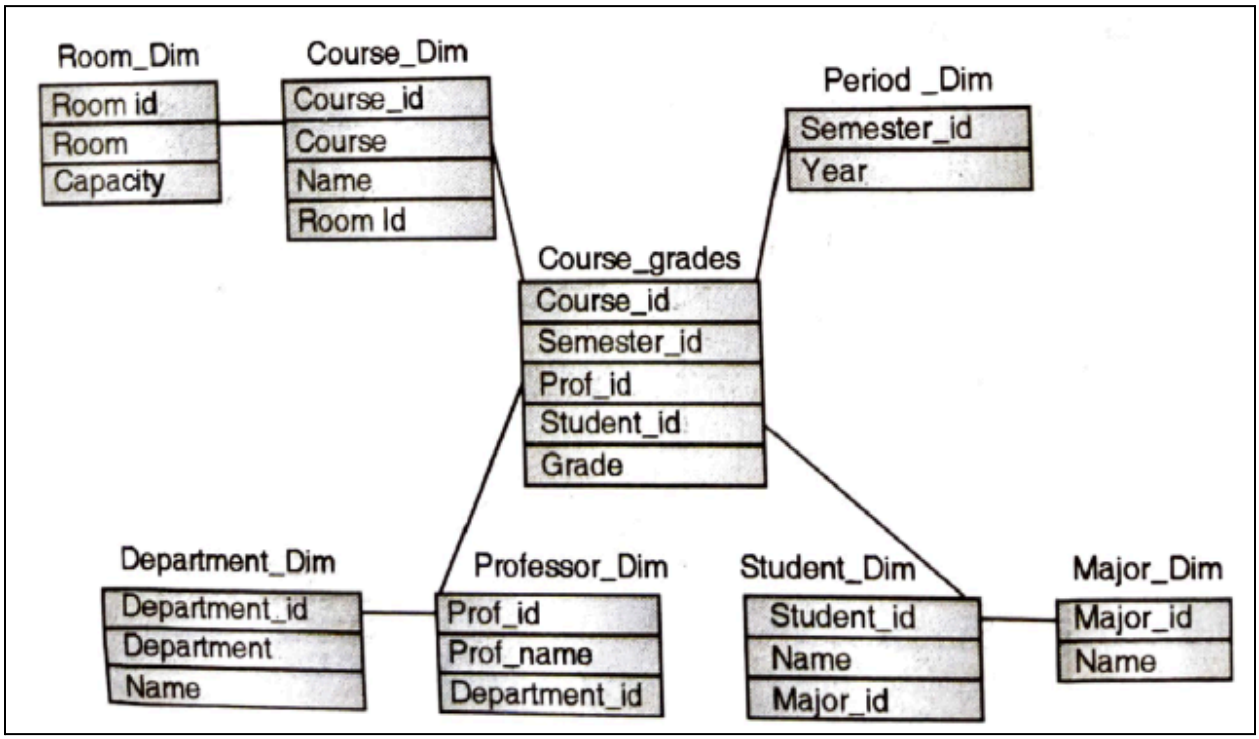
(b) Estimate the number of rows in the fact table, using the assumptions stated above and also estimate the total size of the fact table (in bytes) assuming that each field has an average of 5 bytes.

(c) Can you convert this star schema to a snowflake schema ? Justify your answer and design a snowflake schema if it is possible.

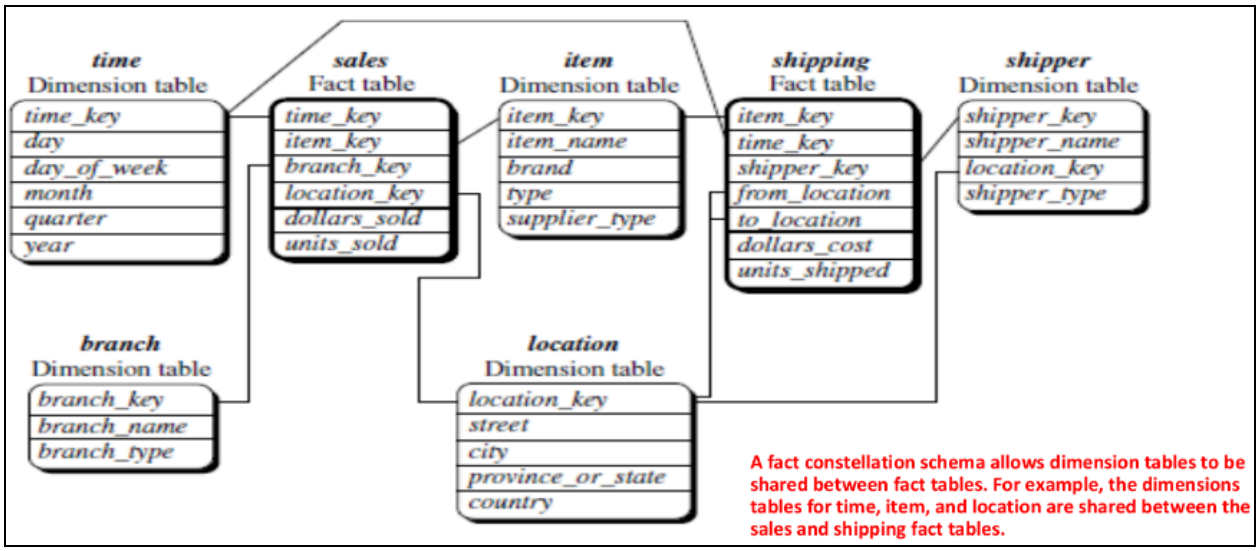
Star Schema-



Snowflake Schema-



d. Extra- Fact Constellation Schema Example of *Sales and Shipping*-



7. Difference between OLTP and OLAP.

OLTP	OLAP
OLTP is an Online Transaction Process.	OLAP is an Online Analysis (and data retrieving) Process.
It is characterized by a large number of short online transactions.	It is characterized by a large volume of data.
OLTP is an online database modifying system.	OLAP is an online database query management system.
OLTP uses traditional DBMS.	OLAP uses the data warehouse.
Insert, Update and Delete information from the database.	Mostly select operations.
OLTP and its transactions are the sources of data.	Different OLTP databases become the source of data for OLAP.
OLTP database must maintain data integrity constraints.	OLAP database does not get frequently modified. Hence data integrity is not an issue.
It's response time is millisecond.	Response time in seconds to minutes.
The data in the OLTP database is	The data in the OLAP process might

always detailed and organized.	not be organized.
Allow read/write operations.	Only read and rarely write.
It is a customer-oriented process.	It is a market-oriented process.
Queries in this process are standardized and simple.	Complex queries involving aggregations.
Complete backup of the data combined with incremental backups.	OLAP only need a backup from time to time. Backup is not important compared to OLTP.
DB design is an application-oriented example: Database design changes with the industry like retail, airline, banking, etc.	DB design is subject-oriented. Example: Database design changes with subjects like marketing, sales, purchasing, etc.
It is used by Data critical users like clerk, DBA and Database professionals.	It is used by Data knowledge users like workers, managers and CEO.
It is designed for real time business operations.	It is designed for analyis of business measures by category and attributes.
Transaction throughput is the performance metric.	Query throughput is the performance metric.
This kind of database allows thousands of users.	This kind of database allows only hundreds of users.
It helps to Increase user's self-service and productivity.	Help to increase the productivity of business analysts.
It provides a fast result for daily used data.	It ensures that response to the query is quicker consistently.
It is easy to create and maintain.	It lets the user create a view with the help of a spreadsheet.

8. What are different OLAP operations?Explain with example.

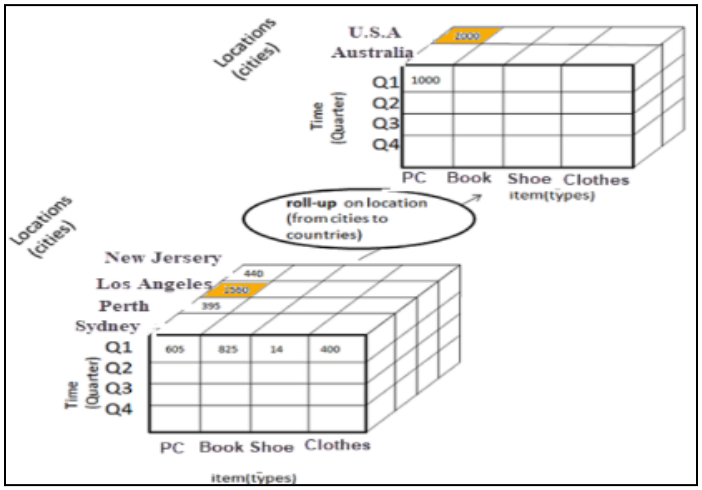
- In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies.
- Thus organization provides users with the flexibility to view data from different perspectives.

- A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand.

Hence, OLAP provides a user-friendly environment for interactive data analysis-

1. Roll-up

- Roll-up is also known as “consolidation” or “aggregation.”
- The Roll-up operation can be performed in 2 ways
  - Reducing dimensions
  - Climbing up concept hierarchy.
- In the roll-up process at least one or more dimensions need to be removed.

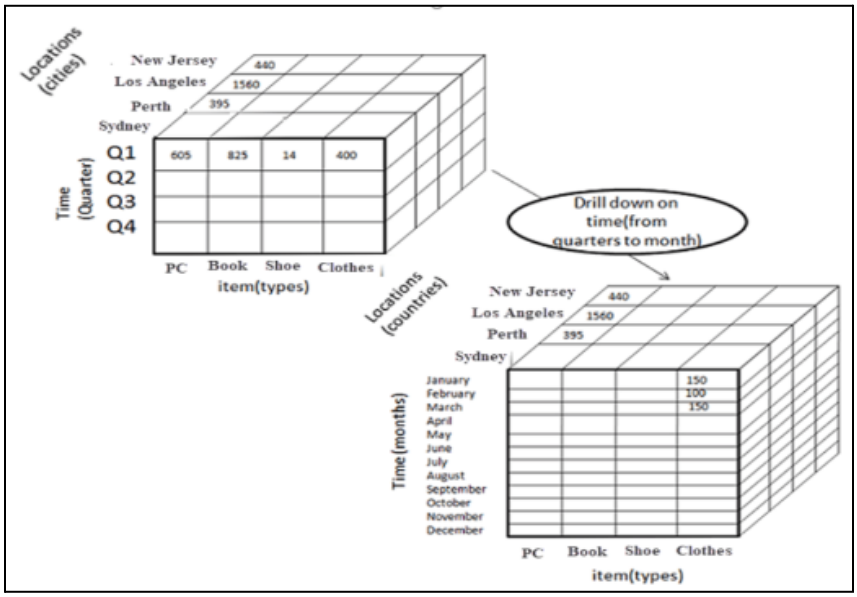


1. In this example, cities New jersey and Los Angels are rolled up into country USA.
2. The sales figure of New Jersey and Los Angels are 440 and 1560 respectively. They become 2000 after roll-up.
3. In this aggregation process, data in location hierarchy moves up from city to the country.
4. In this example, Cities dimension is removed.

2. Drill-down

In drill-down, data is fragmented into smaller parts. It is the opposite of the rollup process. It can be done via-

- Moving down the concept hierarchy
- Increasing a dimension

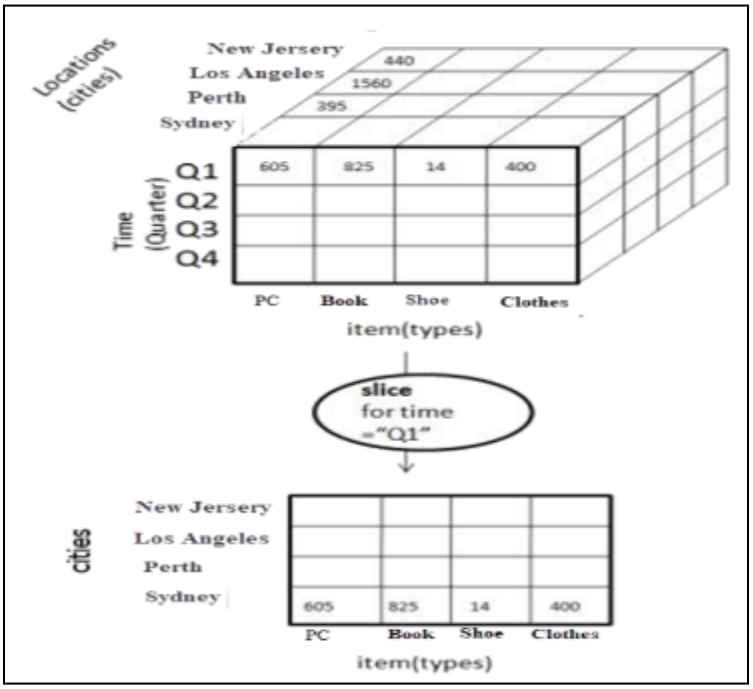


In this Example,
Quarter Q1 is drilled down to months

Here dimension Months is added.

3. Slice

Here, one dimension is selected, and a new sub-cube is created.

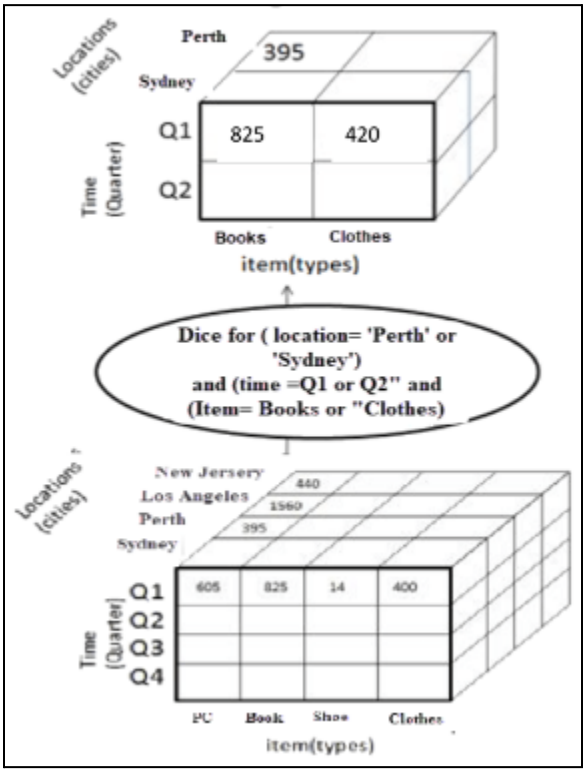


In this Example,
Dimension Time is sliced with Q1 as the filter.

A new cube is created altogether.

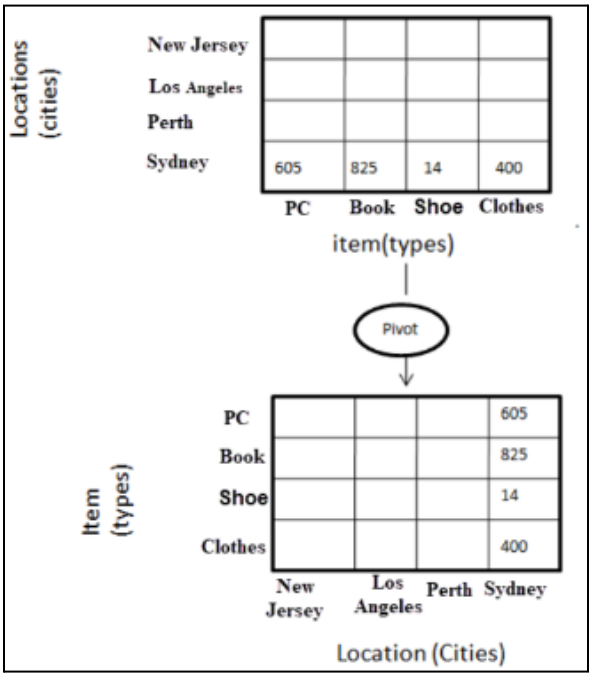
4. Dice

This operation is similar to a slice. The difference in dice is you select 2 or more dimensions that result in the creation of a sub-cube.



5. Pivot(rotate)

In Pivot, you rotate the data axes to provide a substitute presentation of data.



Examples-

Consider a data warehouse for a hospital where there are three dimension

(a) Doctor (b) Patient (c) Time

And two measures i) count ii) charge where charge is the fee that the doctor charges a patient for a visit.

Using the above example describe the following OLAP operations

1) Slice 2) Dice 3) Rollup 4) Drill down 5) Pivot

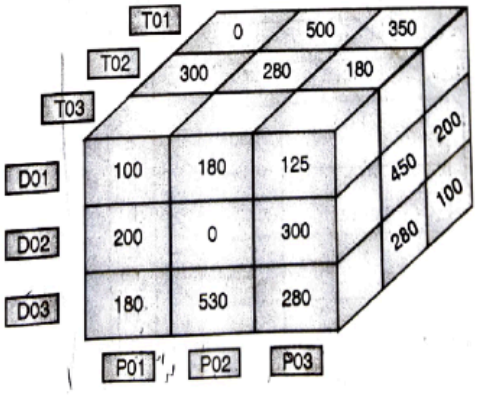
There are four tables, out of 3 dimension tables and 1 fact tabl

Dimension tables :

1. Doctor (DID, name, phone, location,pin,specialisation)
2. Patient (PID,name, phone , state, city,location, pin)
3. Time (TID,day, month, quarter , year)

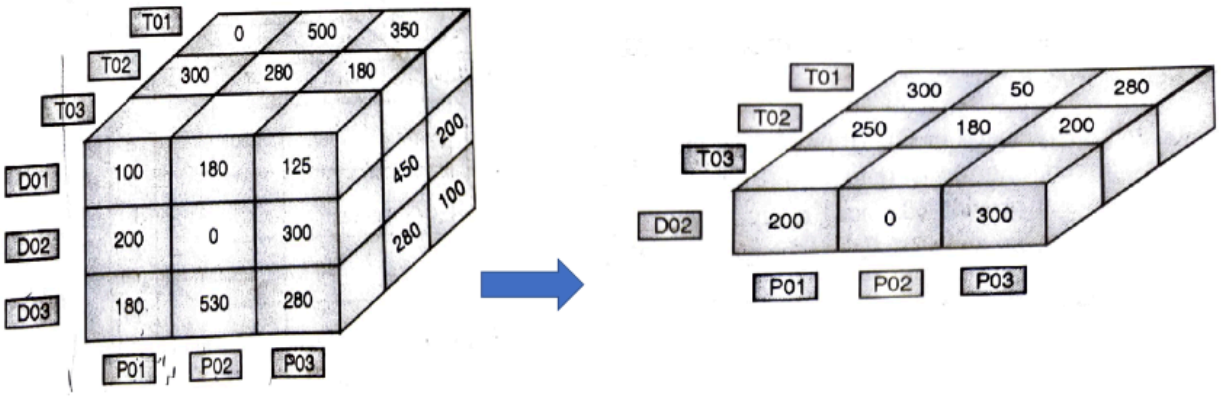
Fact Table :

Fact_table (DID,PID,TID, count, charge)

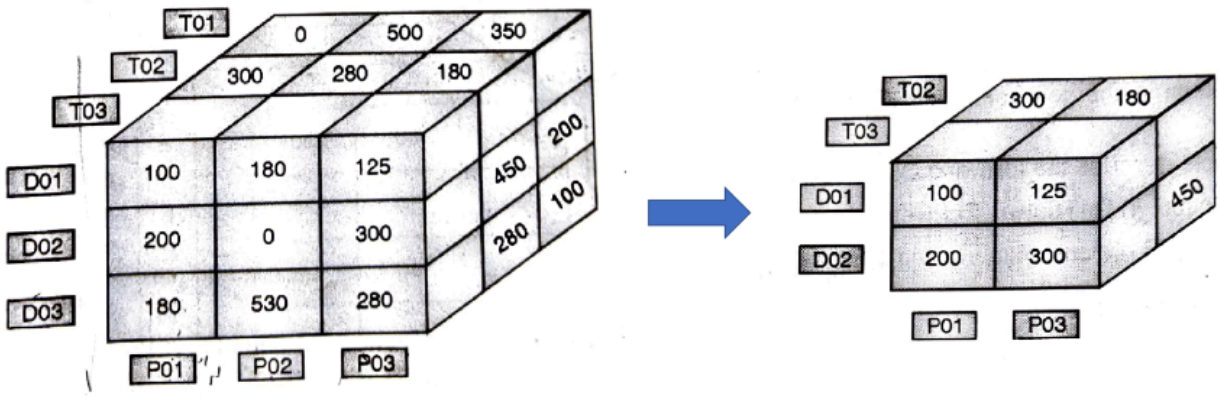


Operations :

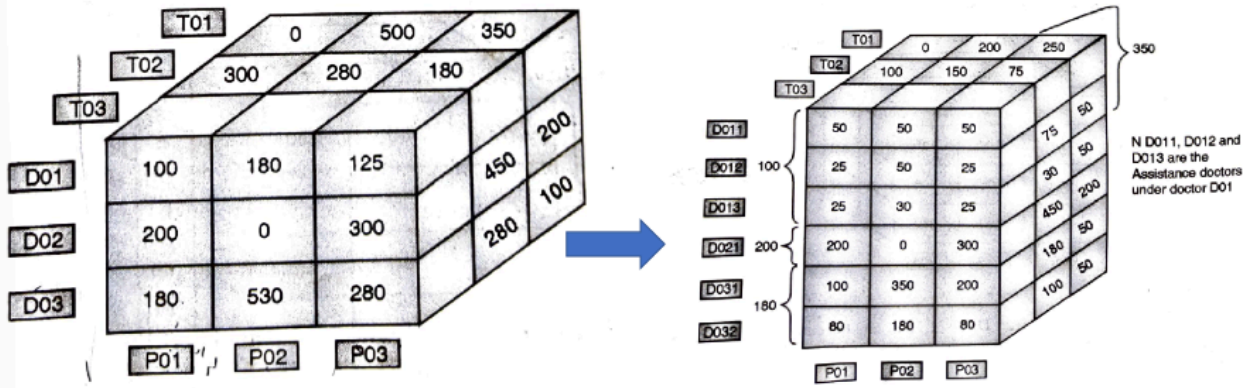
1. **Slice :** Slice on fact table with DID = 2 , this cuts the cube at DID = 2 along the time and patient axis thus it will display a slice of cube, in which time on x and patient on y axis.



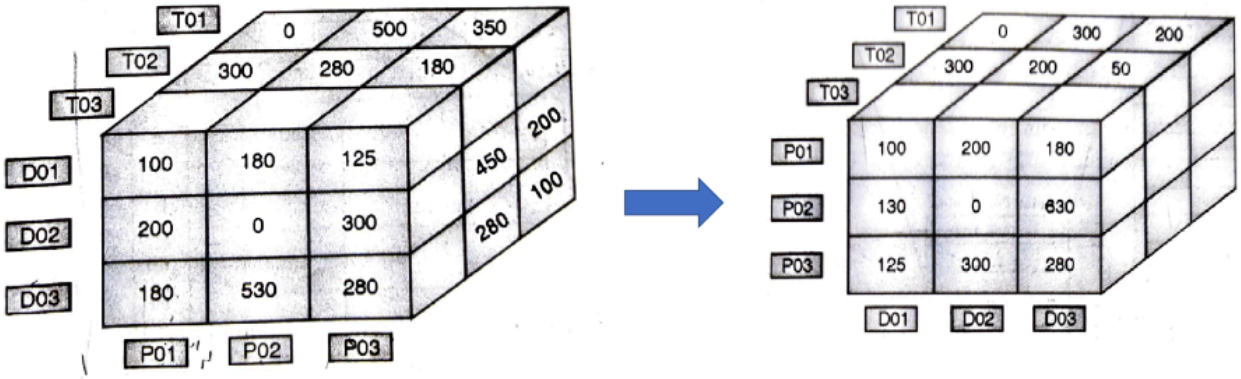
Dice : It is a sub cube of main cube. Thus it cuts the cube with more than predicate like dice on cube with DID = 2, and DID = 01 and PID = 01 PID = 03 and TID = 02, 03



Roll up : it gives summary based on concept hierarchies. Assuming there e concept hierarchy in patient table as state->city->location. Then roll up summarise the charges or count in terms of city or further roll up will charges for a particular state etc.



Drill down : it is opposite to roll up that means if currently cube is summarised with respect to city then drill down will also show summarisation with respect to location



9. Problems on writing a sequence of OLAP operations for the given query.

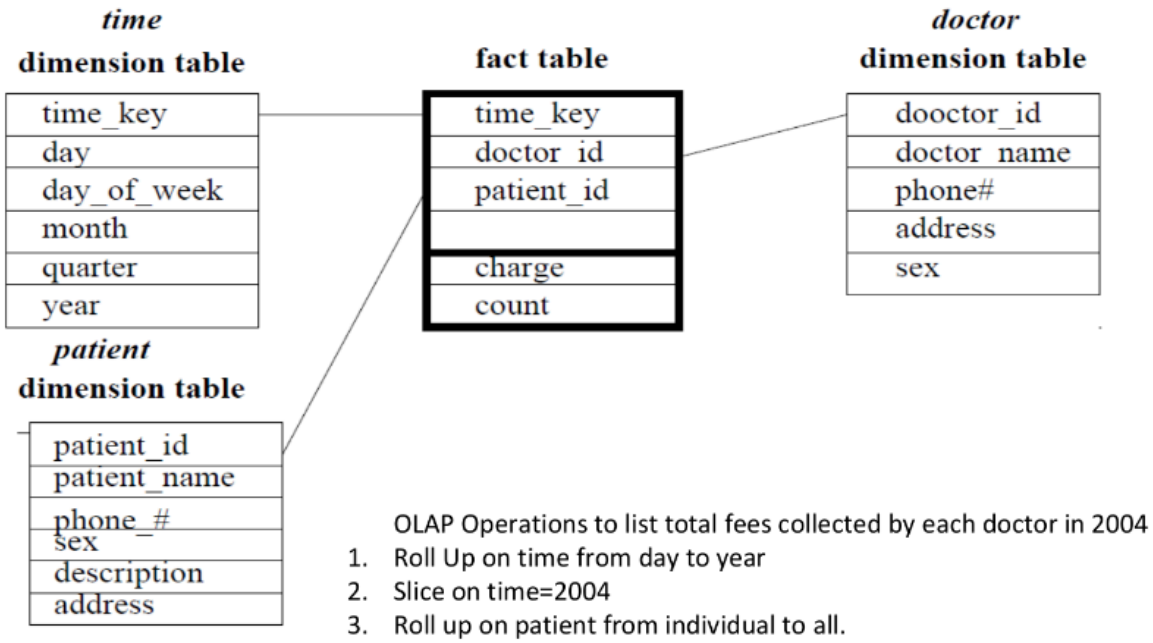
Example 1-

Suppose that a data warehouse consists of the three dimensions *time*, *doctor*, and *patient*, and the two measures *count* and *charge*, where *charge* is the fee that a doctor charges a patient for a visit.

(a) Enumerate three classes of schemas that are popularly used for modeling data warehouses.

(b) Draw a schema diagram for the above data warehouse using one of the schema classes listed in (a).

(c) Starting with the base cuboid [*day*, *doctor*, *patient*], what specific *OLAP operations* should be performed in order to list the total fee collected by each doctor in 2004?



OLAP Operations to list total fees collected by each doctor in 2004

1. Roll Up on time from day to year
2. Slice on time=2004
3. Roll up on patient from individual to all.

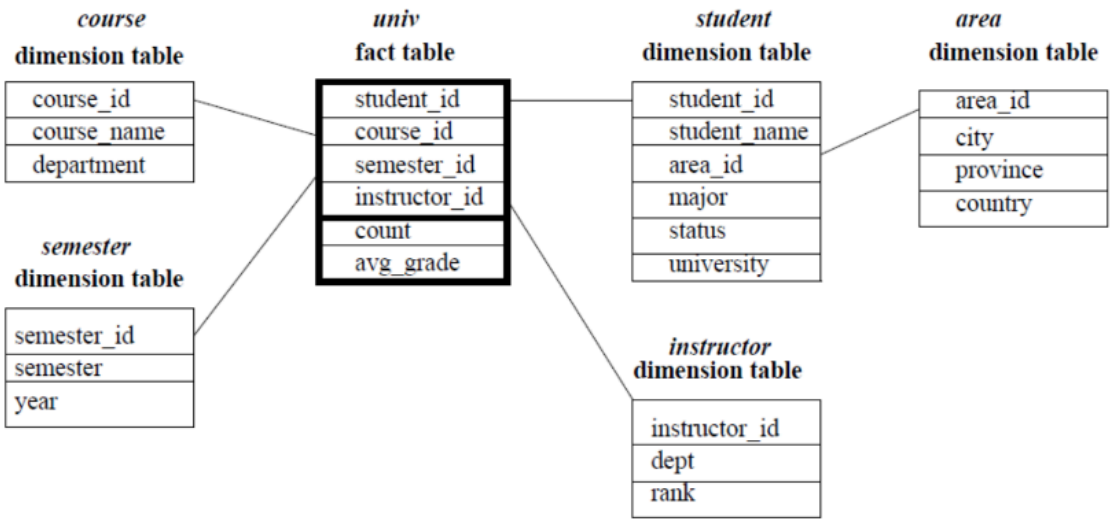
Example 2-

Suppose that a data warehouse for *Big-University* consists of the following four dimensions: *student*, *course*, *semester*, and *instructor*, and two measures *count* and *avg_grade*. When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the *avg_grade* measure stores the actual course grade of the student. At higher conceptual levels, *avg_grade* stores the average grade for the given combination.

(a) Draw a *snowflake schema* diagram for the data warehouse.

(b) Starting with the base cuboid [*student*, *course*, *semester*, *instructor*], what specific *OLAP operations* (e.g., roll-up from *semester* to *year*) should one perform in order to list the average grade of *CS* courses for each *Big-University* student.

(c) If each dimension has five levels (including all), such as "*student* < *major* < *status* < *university* < all", how many cuboids will this cube contain (including the base and apex cuboids)?



Starting with the base cuboid [student, course, semester, instructor], what specific OLAP operations (e.g., roll-up from semester to year) should one perform in order to list the average grade of CS courses for each Big-University student.

The specific OLAP operations to be performed are:

- Roll-up on course from course id to department.
- Roll-up on semester from semester id to all.
- Slice for course="CS" .

(c) If each dimension has five levels (including all), such as student < major < status < university < all, how many cuboids will this cube contain (including the base and apex cuboids)?

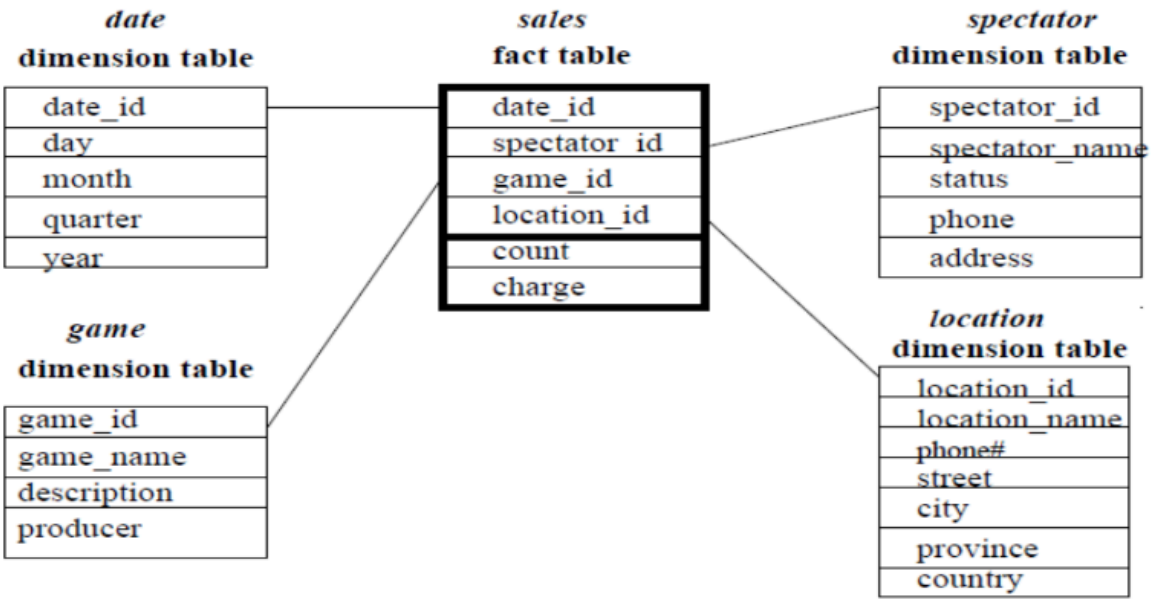
This cube will contain 5^4 = 625 cuboids.

Example 3-

Suppose that a data warehouse consists of the four dimensions, *date*, *spectator*, *location*, and *game*, and the two measures, *count* and *charge*, where *charge* is the fare that a spectator pays when watching a game on a given date. Spectators may be students, adults, or seniors, with each category having its own charge rate.

(a) Draw a *star schema* diagram for the data warehouse.

(b) Starting with the base cuboid [*date*, *spectator*, *location*, *game*], what specific *OLAP operations* should one perform in order to list the total charge paid by student spectators at GM_Place in 2010?



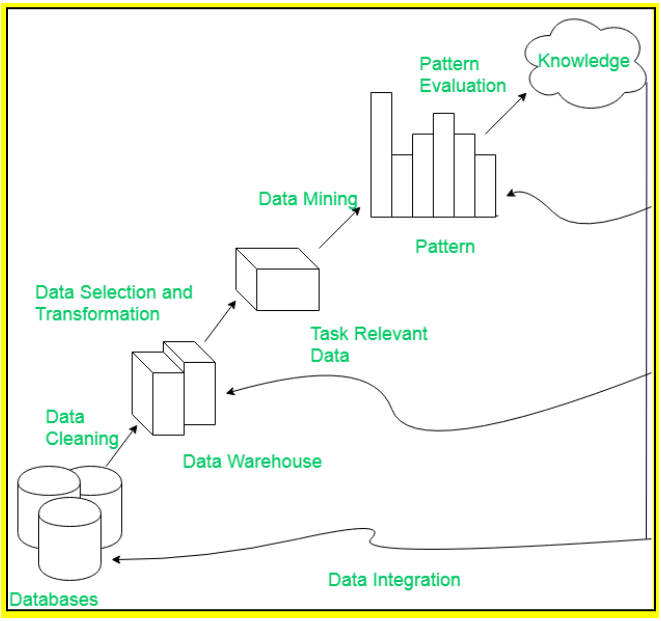
Que: Starting with the base cuboid [date, spectator, location, game], what specific OLAP operations should one perform in order to list the total charge paid by student spectators at GM Place in 2010?

Ans:

The specific OLAP operations to be performed are:

- Roll-up on date from date id to year.
- Roll-up on game from game id to all.
- Roll-up on location from location id to location name.
- Roll-up on spectator from spectator id to status.
- Dice with status="students", location name="GM Place", and year = 2010.

10. Explain steps of KDD (Knowledge discovery in DB)



- Understanding the Domain and Goals:
  - Define the problem and understand the domain in which the data resides.
  - Identify the goals of the knowledge discovery process.
- Data Selection:
  - Select the relevant data from the available datasets based on the defined problem and goals.
  - Ensure that the data is representative and suitable for analysis.
- Data Preprocessing:
  - Clean the data to handle missing values, outliers, and errors.
  - Transform the data into a suitable format for analysis.
  - Feature selection and extraction may be performed to reduce dimensionality.
- Data Transformation:
  - Normalize or standardize numerical data to ensure consistency.
  - Encode categorical variables if necessary.
  - Perform any other transformations needed to prepare the data for analysis.
- Data Mining:
  - Apply data mining techniques to extract patterns, associations, relationships, or knowledge from the prepared dataset.
  - Common data mining techniques include clustering, classification, regression, association rule mining, and more.
- Pattern Evaluation:
  - Evaluate the patterns or models generated by the data mining algorithms.
  - Assess the quality and significance of the discovered patterns in relation to the goals of the knowledge discovery process.

- Knowledge Representation:
    - Represent the discovered patterns or models in a form that is understandable and interpretable by domain experts.
    - This step involves transforming the results into a format that can be used to make decisions or gain insights.
- Interpretation and Evaluation:
    - Interpret the results in the context of the problem domain.
    - Evaluate the usefulness of the discovered knowledge and assess its impact on the decision-making process.

11. State any 2 decision making activities for which organizations are using data in DWH.

- repositioning products and managing product portfolios by **comparing the performance of sales** by quarter, by year, and by geographic regions in order to fine-tune production strategies;
- increasing customer focus, which includes the **analysis of customer buying patterns** (such as buying preference, buying time, budget cycles, and appetites for spending);

12. What is concept hierarchy, partial and total order concept hierarchy? Explain with an example.

- A **concept hierarchy** refers to the arrangement of concepts or categories in a structured and hierarchical manner.
- It helps organize information in a way that reflects the relationships between different concepts.
- There are two main types of concept hierarchies: partial order concept hierarchy and total order concept hierarchy.

  Partial Order Concept Hierarchy:
  In a partial order concept hierarchy, not every pair of concepts has a defined relationship in terms of order.

  E.g:

  Total Order Concept Hierarchy:
  In a total order concept hierarchy, every pair of concepts has a defined relationship in terms of order.

  E.g:

13. What is data mining? State applications of data mining.

- Data mining is the process of converting data into information and then into knowledge.
- Data mining is a process that involves using statistical, mathematical, and artificial intelligence techniques and algorithms to extract and identify useful information and subsequent knowledge (or patterns) from large sets of data.
- Applications:

1. Marketing and CRM:
    - To identify most likely buyers of new products
    - To identify root causes of customer attrition(reduction) so as to improve customer retention
    - To discover time variant associations between products and services to maximize sales and find most profitable customers.
2. Banking and Finance:
    - To detect fraudulent credit card and online banking transactions
    - To optimize the cash return by forecasting cash flow on banking entities
    - To streamline and automate the processing of loan applications by accurately predicting most probable defaulters.
    - To maximize the customer value by identifying and selling the products and services that customers are most likely to buy.
3. Retailing and Logistics:
    - To identify accurate sales volume at specific retail locations in order to determine correct inventory levels.
    - To do MBA to improve store layout and optimize sales promotions
    - To forecast consumption levels for different product types.
    - To discover interesting patterns in the movement of products in a supply chain by analyzing sensory and RFID data.
4. Manufacturing:
    - To predict machine failures using sensory data
    - To discover novel patterns to identify and improve product quality.
5. Brokerages and Security Tradings:
    - To predict when and how much certain stock / bond prices will change.
    - To forecast range of market fluctuations,direction of fluctuations
    - To assess effect of particular issues/events on market movements.
    - To identify and prevent fraudulent activities in security trading.
6. Insurance:
    - To predict which customers will buy new policies
    - Identify fraudulent behavior of customers
    - Prevent incorrect claim payments
7. Computer Hardware and Software:
    - Predict disk failure
    - To identify and filter unwanted web contents and email messages
    - To identify potentially unsecured software products
8. Government and Defense:
    - To forecast the cost of moving military personnel and equipments.
    - To predict resource consumption for better planning and budgeting

9. Travel and Lodging:
    - To predict sales of different services to optimally price these services.
    - To forecast demand at different locations to better allocate limited organizational
    - resources. .
    - To identify most profitable customers and provide them with personalised services.
    - To retain valuable employees by identifying and acting on the root causes for attrition

10. Health and Healthcare:
    - To identify successful medical therapies for different illnesses.
    - To identify people without health insurance and reasons behind it.
    - To forecast the time of demand at different service locations to optimally allocate organizational resources.
    - To retain valuable employees by identifying root causes for attrition

11. Entertainment:
    - To analyze viewer data to determine which programs to show during prime time.
    - To decide where to insert advertisements so as to maximize the returns.
    - To predict financial success of the movies before they are produced.

12. Sports:
    - To improve performance of NBA teams in US
    - To increase the chances of winning.

14. What are the different types of patterns that can be mined?

Data mining functionalities are used to specify the kinds of patterns to be found in data mining tasks.

- **Descriptive mining tasks** : Deals with the General characteristics and converts them into relevant and useful information

1. Class/Concept Description:

- Data entries can be associated with the classes or concepts.
- These descriptions can be derived using

(1) data characterization, by summarizing the data of the class under study (often called the target class) in general terms,
Example: At an electronic store a Customer relationship manager asks to Summarize the characteristics of customers who spend more than Rs.10000 a year at the store.
or
(2) data discrimination, by comparison of the target class with one or a set of comparative classes (often called the contrasting classes),
Example: A customer relationship manager at Electronics store may want to compare two groups of customers—those who shop for computer products regularly (e.g., more than twice a month) and those who rarely shop for such products (e.g. less than three times a year).
or
(3) both data characterization and discrimination.

2. Mining of frequent patterns:

- Patterns that occur frequently in data.
- It includes-

a. Frequent itemset : refers to a set of items that often appear together in a transactional data set

b. Frequent subsequences (also known as sequential patterns): A frequently occurring subsequence like laptop□digital camera□ memory card

c. Frequent substructures:

A substructure can refer to different structural forms (e.g., graphs, trees, or lattices) that may be combined with itemsets or subsequences.
If a substructure occurs frequently, it is called a (frequent) structured pattern.

- Mining frequent patterns leads to the discovery of interesting associations and correlations within data.

3. Association Analysis

- Defines relationships between the data and predefined association rules.
- Suppose that, as a marketing manager at Electronics store, you want to know which items are frequently purchased together

A rule, $buys(X, \text{“}computer\text{”}) \Rightarrow buys(X, \text{“}software\text{”})\ [support = 1\%, confidence = 50\%]$

- Association rules that contain a single predicate are referred to as single-dimensional association rules.
- Suppose, instead, that we are given the Electronics relational database related to purchases. A data mining system may find association rules like

$$age(X, \text{“}20..29\text{”}) \wedge income(X, \text{“}40K..49K\text{”}) \Rightarrow buys(X, \text{“}laptop\text{”})$$
$$[support = 2\%, confidence = 60\%].$$

- Association rules that contain more than one predicate/attributes are referred to as multi-dimensional association rules.

4. Clustering :

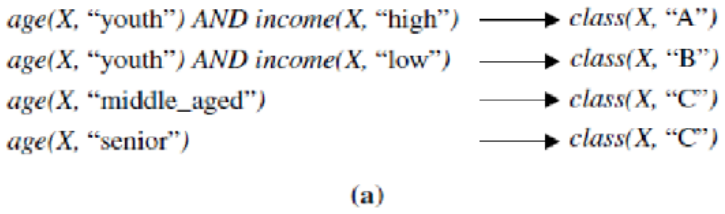
- It can be used to generate class labels for a group of data.
- The objects are clustered or grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. i.e. clusters of objects are formed so that objects within a cluster have high similarity , but are rather dissimilar to objects in other clusters.
- Clustering can also facilitate taxonomy formation □ Organization of observations into a hierarchy of classes that group similar events together.

Example: Cluster analysis can be performed on Electronics store customer data to identify homogeneous subpopulations of customers. These clusters may represent individual target groups for marketing

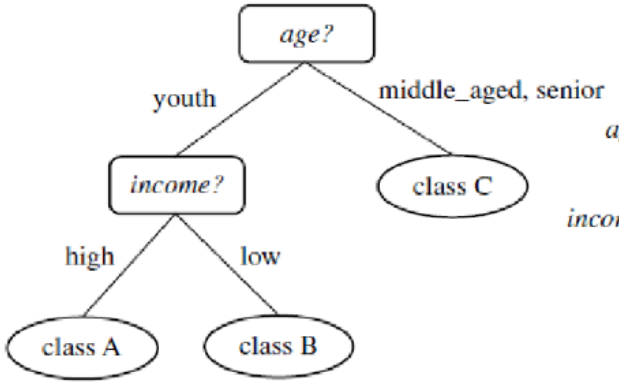
- **Predictive mining tasks:** Predicts future values by analyzing data patterns and their outcomes based on past data.

1. Classification: is the process of finding a model (or function) that describes and distinguishes data classes or concepts.

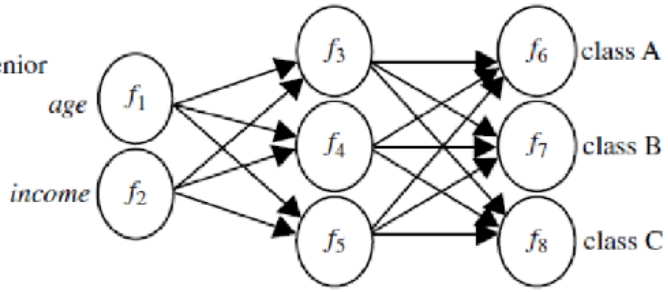
- The models are derived based on the analysis of a set of training data (i.e., data objects for which the class labels are known).
- The model is used to predict the class label of objects for which the class label is unknown.
- The derived model may be represented in various forms, such as classification rules (i.e., IF-THEN rules), decision trees, mathematical formulae, or neural networks



(a)



(b)



(c)

2. Regression:

- Whereas classification predicts categorical (discrete, unordered) labels, regression models continuous-valued functions. That is, regression is used to predict missing or unavailable numerical data values rather than (discrete) class labels.
- The term prediction refers to both numeric prediction and class label prediction.
- Regression analysis is a statistical methodology that is most often used for numeric prediction.
- Regression also encompasses the identification of distribution trends based on the available data.

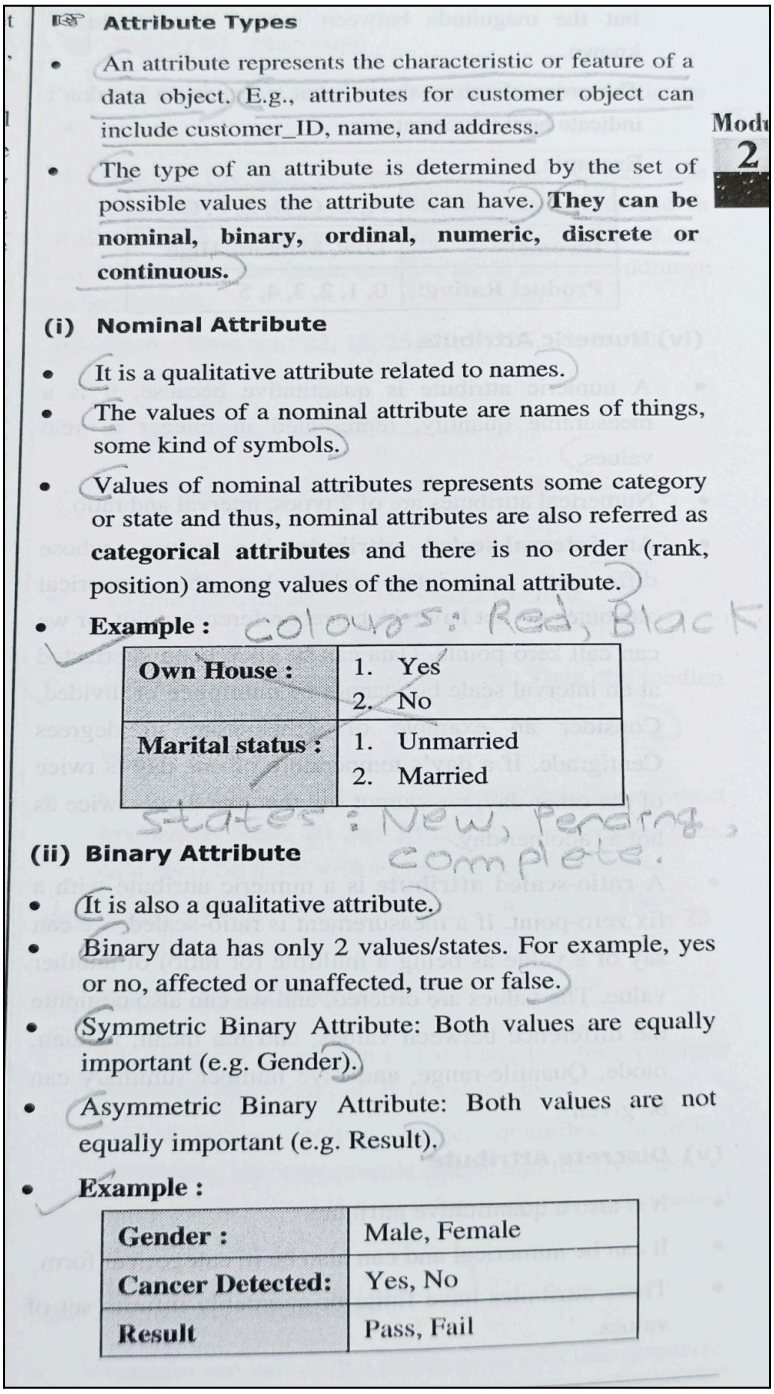
3. Outlier Analysis :
   - Outlier: A data object that does not comply with the general behavior of the data
   - Many data mining methods discard outliers as noise or exceptions. However, in some applications (e.g., fraud detection), the rare events can be more interesting than the more regularly occurring ones.
   - Outliers may be detected using statistical tests that assume a distribution or probability model for the data, or using distance measures where objects that are remote from any other cluster are considered outliers.
   - Example: Outlier analysis may uncover fraudulent usage of credit cards by detecting purchases of unusually large amounts for a given account number in comparison to regular charges incurred by the same account.

Outlier values may also be detected with respect to the locations and types of purchase, or the purchase frequency.

- It is used in observing the change in trends of buying patterns of a customer.

Module 2: Preprocessing

1. What are the different types of attributes? Explain with examples



☞ **Attribute Types**

- An attribute represents the characteristic or feature of a data object. E.g., attributes for customer object can include customer_ID, name, and address.
- The type of an attribute is determined by the set of possible values the attribute can have. **They can be nominal, binary, ordinal, numeric, discrete or continuous.**

(i) Nominal Attribute

- It is a qualitative attribute related to names.
- The values of a nominal attribute are names of things, some kind of symbols.
- Values of nominal attributes represents some category or state and thus, nominal attributes are also referred as **categorical attributes** and there is no order (rank, position) among values of the nominal attribute.
- **Example :**

Own House :	1. Yes 2. No
Marital status :	1. Unmarried 2. Married

(ii) Binary Attribute

- It is also a qualitative attribute.
- Binary data has only 2 values/states. For example, yes or no, affected or unaffected, true or false.
- Symmetric Binary Attribute: Both values are equally important (e.g. Gender).
- Asymmetric Binary Attribute: Both values are not equally important (e.g. Result).
- **Example :**

Gender :	Male, Female
Cancer Detected:	Yes, No
Result	Pass, Fail

(iii) Ordinal Attribute

- It is also a qualitative attribute.
- The Ordinal Attributes contains values that have a meaningful sequence or ranking(order) between them, but the magnitude between values is not actually known.
- The order of values shows what is important but don't indicate how important it is.
- **Example :**

Grade:	A, B, C, D, E, F, O
Income:	Low, Medium, High
Product Rating:	0, 1, 2, 3, 4, 5

(iv) Numeric Attribute

- A numeric attribute is quantitative because, it is a measurable quantity, represented in integer or real values.
- Numerical attributes are of 2 types, interval and ratio.
- An **interval-scaled attribute** has values, whose differences are interpretable, but the numerical attributes do not have the correct reference point, or we can call zero points. Data can be added and subtracted at an interval scale but cannot be multiplied or divided. Consider an example of temperature in degrees Centigrade. If a day's temperature of one day is twice of the other day, we cannot say that one day is twice as hot as another day.
- A **ratio-scaled attribute** is a numeric attribute with a fix zero-point. If a measurement is ratio-scaled, we can say of a value as being a multiple (or ratio) of another value. The values are ordered, and we can also compute the difference between values, and the mean, median, mode, Quantile-range, and Five number summary can be given.

(v) Discrete Attribute

- It is also a quantitative attribute.
- It can be numerical and can also be in categorical form.
- These attributes have finite or countably infinite set of values.

- **Example :**

Profession:	Principal, Teacher, Clerk, Peon
Zipcode:	400050, 400051, 400052

(vi) Continuous Attribute

- It is also a quantitative attribute.
- It can take any value between two specified values.
- **Example :**

Height:	5.2, 5.4, 5.6, ...
Weight:	50.33, ...

- **Attribute** (or dimensions, features, variables): A data field, representing a characteristic or feature of a data object.
    - e.g., customer _ID, name, address
- **Observations**: observed values for a given attribute
- **Attribute vector/feature vector**: A set of attributes that define an object.
- **Types**:
    - Nominal
    - Ordinal
    - Binary
    - Numeric

1. **Nominal**: categories, states, or "names of things"
    - Hair_color = {auburn, black, blond, brown, grey, red, white}
    - marital status, occupation, ID numbers, zip codes
- In the cases of nominal attributes with numeric values e.g. Cust_ID, the numbers are not intended to be used quantitatively.
- Also in case of numeric nominal attributes, values do not have any meaningful order about them.

2. **Binary attributes:** Nominal attribute with only 2 categories/states (0 or 1)
    - 0: attribute is absent
    - 1: attribute is present
- Symmetric binary: both outcomes equally important
    - e.g., gender
- Asymmetric binary: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
- Convention: assign 1 to most important outcome (e.g., HIV positive)
- If two states are True and False, then called as Boolean Attribute

3. **Ordinal Attributes:** Values have a meaningful order or a ranking among them but magnitude between successive values is not known.
    - Ex: Size = {small, medium, large}, grades, professor rankings
- Useful for registering subjective assessments of qualities that cannot be measured objectively; thus often used in surveys for ratings.
    - E.g Customer satisfaction survey
- We can compute mean and median but not mode for the ordinal attributes .
- Note:nominal, binary, and ordinal attributes are qualitative attributes.

4. **Numeric Attributes /Quantitative Attributes:** Represents measurable quantity in integer or real values.

- Types of numeric attributes:

4.1 Interval scaled attributes

    - Measured on a scale of equal-sized units
    - Values have order and can be positive, 0, or negative

- E.g., temperature in C˚or F˚, calendar dates
- We can obtain a ranking of objects by ordering the values.
- Also allow us to compare and quantify the difference between values.
- No true zero-point -We can not speak of values in terms of ratio.
- e.g without a true zero point, we can't say that 10 C˚ is twice as warm as 5C˚.
- Mean, Median and Mode

4.2 Ratio scaled attributes
- Inherent zero-point (fixed zero-point)
- The values are ordered, and we can also compute the difference between values, as well as the mean, median,and mode
- Examples: Count attributes such as years of experience and number of words attribute for a document
- attributes to measure age, weight, height and monetary quantities (e.g., you are 100 times richer with $100 than with $1).

5. **Discrete Attribute:** Has only a finite or countably infinite set of values which may or may not be represented as integers.
   - E.g., zip codes, profession, or the set of words in a collection of documents

- Note: Binary attributes are a special case of discrete attributes

6. **Continuous Attribute:** Has real numbers as attribute values
   - E.g., temperature, height, or weight

- Practically, real values can only be measured and represented using a finite number of digits
- Continuous attributes are typically represented as floating-point variables

2. Problems on basic statistical descriptions of data like finding mean, median, midrange standard deviation, variance,modes for given data.Drawing boxplot for given data to identify outliers.

Suppose we have the following values for *salary* (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.
Claculate variance and standard deviation.

i] Mean $= \frac{1}{N} \sum_{i=1}^{N} x_i$

$= \frac{30 + 36 + 47 + 50 + 52 + 52 + 56 + 60 + 63 + 70 + 70 + 110}{12}$

Mean = 58

ii] Median :

Since the data set has even number of data values, median is the ~~aveo~~ average of the $\frac{n}{2}^{th}$ and $\left(\frac{n}{2}+1\right)^{th}$ data values.

$\therefore$ Median $\Rightarrow \frac{x_6 + x_7}{2} = \frac{52 + 56}{2} = 54$

$\therefore$ **Median = 54**

iii] Modes : It is Bimodal. Modes are **52 & 70**

iv] Midrange : Avg. of largest & smallest value

$= \frac{30 + 110}{2} = 70$

$\therefore$ **Midrange = 70**

Date:

v] Variance $\Rightarrow \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2$

$= \frac{1}{12} [(30-58)^2 + (36-58)^2 + (47-58)^2 + (50-58)^2 +$
$(52-58)^2 + (52-58)^2 + (56-58)^2 + (60-58)^2 +$
$(63-58)^2 + (70-58)^2 + (70-58)^2 + (110-58)^2]$

$= \frac{1}{12} [1453 + 80 + 3017]$

$= \boxed{379.17 = \text{Variance}} = \sigma^2$

vi] Std. deviation $= \sigma = \sqrt{\sigma^2}$

$= \sqrt{379.17}$

$\boxed{\sigma = 19.47}$

UEx. 2.2.2 **MU - May 2019, Dec. 2019**

Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order):

13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

The numbers are sorted in ascending order.

i] Mean $= \frac{1}{N} \sum_{i=1}^{N} x_i$

$= \frac{13+15+16+16+19+20+20+21+22+22+25+25+25+25+30+33+33+35+35+35+35+36+40+45+46+52+70}{27}$

$= \frac{809}{27} = 29.9629$

Median ⇒ Middle value of the ordered set.
$= 25$.

ii] The modes are 25 and 35
∴ Bimodal.

iii] Midrange ⇒ Avg of minimum & maximum values.
$= \frac{13+70}{2} = 41.5$

iv] Five-point summary ⇒
$\{13, Q_1, 25, Q_2, 70\}$.

Q_1 ⇒ First Quartile i.e. value corresponding to 25th percentile.
$= 20$

Q_3 ⇒ Third Quartile i.e. value corresponding to 75th percentile
$= 35$

∴ Five point summary ⇒
$\{13, 20, 25, 35, 70\}$

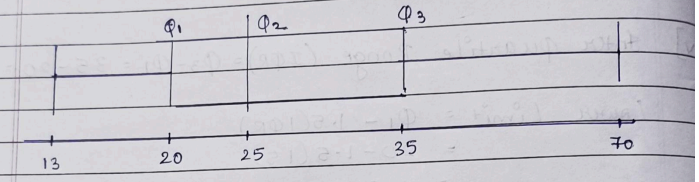
v] Inter Quartile Range (IQR) $= Q_3 - Q_1 = 35 - 20 = 15$.

Lower Limit $= Q_1 - 1.5(IQR)$
$= 20 - 1.5(15)$
$= -2.5$

Upper Limit $= Q_3 + 1.5(IQR)$
$= 35 + 1.5(15)$
$= 57.5$

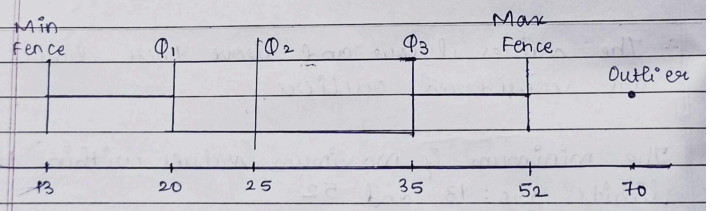
The values below & above these limits are outliers.
∴ 70 is an outlier.

Box-plot for 5-point summary:



Modified Box-plot to show outliers:

- The minimum & maximum values within [-2.5, 57.5] are 13 and 52.
- 70 is an outlier.



3. What is a five number summary of data?

2.2.2 (C) Five Number Summary

- The five number summary gives you a rough idea about what your data set looks like.
- It includes five items: the minimum value, the first quartile (Q_1), the median, the third quartile (Q_3), the maximum value.
- In order for the five numbers to exist, your data set must meet these two requirements :

a. Your data must be **univariate**. In other words, the data must be a single variable. For example, this list of weights is one variable: 120, 100, 130, 145. If you have a list of ages and you want to compare the ages to weights, it becomes bivariate data (two variables). For example: age 1 (25 pounds), 5 (60 pounds), 15 (129 pounds). The matching pairs makes it impossible to find a five number summary.

b. Your data must be **ordinal**, **interval**, or **ratio**.

- Steps to Find a Five-Number Summary

▶ **Step 1 :** Put your numbers in ascending order (from smallest to largest).

For example, consider the data set in order as: 1, 2, 5, 6, 7, 9, 12, 15, 18, 19, 27.

▶ **Step 2:** Find the minimum and maximum for your data set.

In the example in step 1, the minimum (the smallest number) is 1 and the maximum (the largest number) is 27.

▶ **Step 3 :** Find the median. The median is the middle number.

▶ ***Step 4 :*** *Place parentheses around the numbers above and below the median. (This is not technically necessary, but it makes Q_1 and Q_3 easier to find).*

(1, 2, 5, 6, 7), 9, (12, 15, 18, 19, 27).

▶ **Step 5 :** Find Q_1 and Q_3. Q_1 can be thought of as a median in the lower half of the data, and Q_3 can be thought of as a median for the upper half of data.

(1, 2, 5, 6, 7), 9, (12,15,18,19,27).

▶ **Step 6 :** Write down your summary found in the above steps.

minimum = 1, Q_1 = 5, median = 9, Q_3 = 18 and maximum = 27.

Five-number summary of a distribution (Minimum, Q1, Median, Q3, Maximum)

- It is more informative to also provide the two quartiles Q1 and Q3, along with the median
- A common rule of thumb for identifying suspected outliers is to single out values falling at least 1.5IQR above the third quartile or below the first quartile.
- Because Q1, the median, and Q3 together contain no information about the endpoints (e.g., tails) of the data, a fuller summary of the shape of a distribution can be obtained by providing the lowest and highest data values as well. This is known as the five-number summary.
- The five-number summary of a distribution consists of the median (Q2), the quartiles Q1 and Q3, and the smallest and largest individual observations, written in the order of Minimum, Q1, Median, Q3, Maximum.

4. How can we compute dissimilarity between two binary attributes?

how can we compute the dissimilarity between two binary attributes?

One approach involves computing a dissimilarity matrix from the given binary data. If all binary attributes are thought of as having the same weight, we have the 2 x 2 contingency table as shown below

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

Where q = the number of attributes that equal 1 for both objects *i* and *j*,

r = the number of attributes that equal 1 for object *i* but equal 0 for object *j*,

s = the number of attributes that equal 0 for object *i* but equal 1 for object *j*,

t = the number of attributes that equal 0 for both objects *i* and *j*.

p = The total number of attributes = $q + r + s + t$.

35

Proximity Measure for Binary Attributes

A contingency table for binary data

Object *j*

Object *i*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

Distance measure for symmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

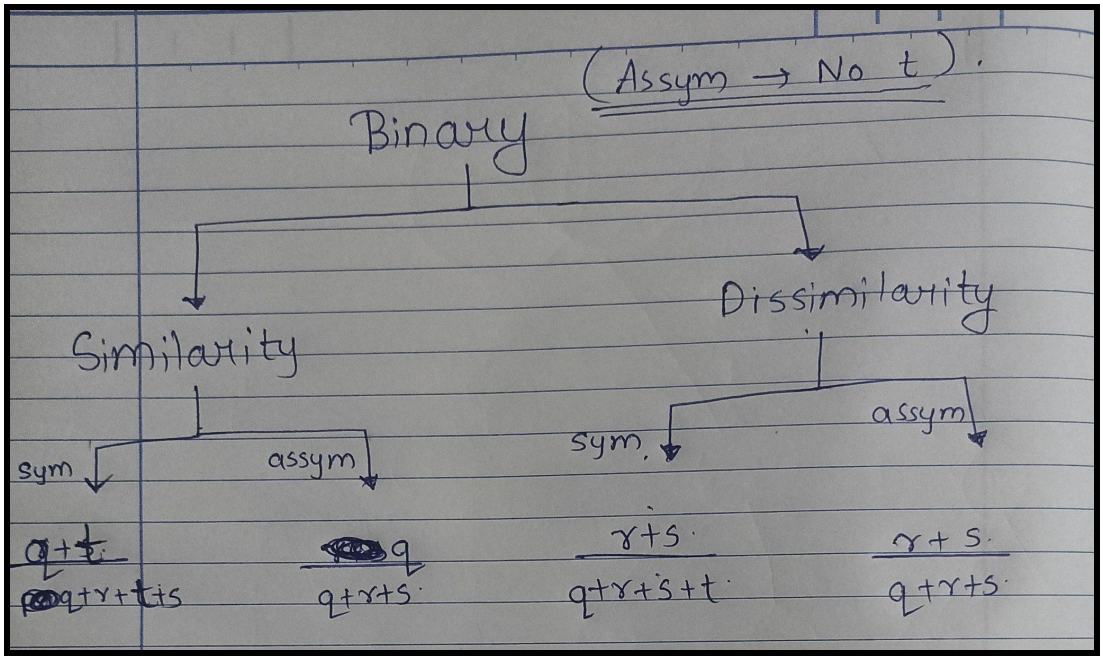
Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s}$$

Jaccard coefficient (similarity measure for asymmetric binary variables):

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

= 1- d(i,j)



Dissimilarity between Binary Attributes

Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- *Gender is a symmetric attribute*
- *The remaining attributes are asymmetric binary*
- *Let the values Y and P be 1, and the value N be 0*

Suppose distance is computed based on only asymmetric attributes

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

$$d(i, j) = \frac{r+s}{q+r+s}$$

- These measurements suggest that Jim and Mary are unlikely to have a similar disease because they have the highest dissimilarity value among the three pairs.

Of the three patients, Jack and Mary are most likely to have a similar disease.

37

5. What is Euclidean distance, Manhattan distance, Minkowski distance? Problems on computing these distances between given objects.

Dissimilarity of Numeric Data:

Euclidean distance: The most popular distance measure

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

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two objects described by numeric attributes.

Manhattan (or city block) distance: named so because it is the distance in blocks between any two points in a city (such as 2 blocks down and 3 blocks over for a total of 5 blocks).

- The distance between two points measured along axes at right angles

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

38

Dissimilarity of Numeric Data: Minkowski Distance

The Euclidean and the Manhattan distance satisfy the following mathematical properties:

- **Non-negativity:** *d*(*i*, *j*) >= 0: Distance is a non-negative number.
- **Identity of indiscernibles:** *d*(*i*, *j*) = 0:The distance of an object to itself is 0.
- **Symmetry:** *d*(*i*, *j*) = *d*(*j*, *i*) : Distance is a symmetric function.
- **Triangle inequality:** *d*(*i*, *j*) <=*d*(*i*, *k*)+*d*(*k*,*j*): Going directly from object *i* to object *j* in space is no more than making a detour over any other object *k*.

A measure that satisfies these conditions is known as **metric**.

Example: Euclidean and Manhattan distance

Euclidean distance and Manhattan distance. Let $x_1 = (1, 2)$ and $x_2 = (3, 5)$ represent two objects The Euclidean distance between the two is $\sqrt{2^2 + 3^2} = 3.61$. The Manhattan distance between the two is $2 + 3 = 5$.

Dissimilarity of Numeric Data: Minkowski Distance

Minkowski distance: A generalization of Euclidean and Manhattan distances

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

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two objects described by p numeric attributes and h is a real number such that h >= 1.

- Also called as L_p norm where p refers to h.

Special Cases of Minkowski Distance

- *h = 1: Manhattan (city block, L_1 norm) distance*

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

- *h = 2: (L_2 norm) Euclidean distance*

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

- *$h \to \infty$. "supremum" (L_{max} norm, L_∞ norm, Chebyshev distance) distance.*

To compute it, we find the attribute *f* that gives the maximum difference in values between the two objects.

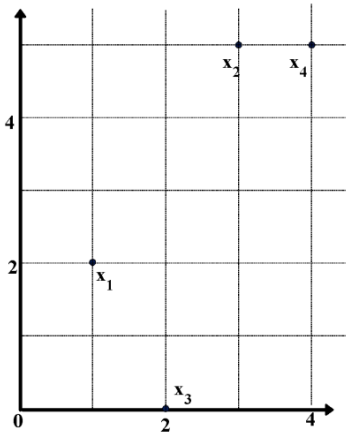
This difference is the supremum distance, defined more formally as:

$$d(i, j) = \lim_{h\to\infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_{f}^{p} |x_{if} - x_{jf}|$$

Example: **Supremum distance.** Let's use the two objects, $\boldsymbol{x}_1 = (1, 2)$ and $\boldsymbol{x}_2 = (3, 5)$ The second attribute gives the greatest difference between values for the objects, which is $5 - 2 = 3$. This is the supremum distance between both objects.

Example: Minkowski Distance

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Dissimilarity Matrices

Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

L_∞	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

Problem 1

Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the *Euclidean distance* between the two objects.

(b) Compute the *Manhattan distance* between the two objects.

(c) Compute the *Minkowski distance* between the two objects, using $h = 3$.

(d) Compute the *supremum distance* between the two objects.

Solution-Problem 1

(a) Compute the *Euclidean distance* between the two objects.

The Euclidean distance is computed using Equation (2.6).

Therefore, we have $\sqrt{(22-20)^2 + (1-0)^2 + (42-36)^2 + (10-8)^2} = \sqrt{45} = 6.7082$.

(b) Compute the *Manhattan distance* between the two objects.

The Manhattan distance is computed using Equation (2.7). Therefore, we have $|22-20| + |1-0| + |42-36| + |10-8| = 11$.

(c) Compute the *Minkowski distance* between the two objects, using $h = 3$.

The Minkowski disance is

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdot + |x_{ip} - x_{jp}|^h} \tag{2.10}$$

where h is a real number such that $h \geq 1$.

Therefore, with $h = 3$, we have $\sqrt[3]{|22-20|^3 + |1-0|^3 + |42-36|^3 + |10-8|^3} = \sqrt[3]{233} = 6.1534$.

(d) Compute the *supremum distance* between the two objects.

The supremum distance is computed using Equation (2.8). Therefore, we have a supremum distance of 6.

Problem 2

It is important to define or select similarity measures in data analysis. However, there is no commonly-accepted subjective similarity measure. Results can vary depending on the similarity measures used. Nonetheless, seemingly different similarity measures may be equivalent after some transformation.

Suppose we have the following two-dimensional data set:

	A_1	A_2
$\boldsymbol{x_1}$	1.5	1.7
$\boldsymbol{x_2}$	2	1.9
$\boldsymbol{x_3}$	1.6	1.8
$\boldsymbol{x_4}$	1.2	1.5
$\boldsymbol{x_5}$	1.5	1.0

(a) Consider the data as two-dimensional data points. Given a new data point, $\boldsymbol{x} = (1.4, 1.6)$ as a query, rank the database points based on similarity with the query using Euclidean distance, Manhattan distance, supremum distance, and cosine similarity.

(b) Normalize the data set to make the norm of each data point equal to 1. Use Euclidean distance on the transformed data to rank the data points.

Solution –Problem2(a)

a)

	Euclidean dist.	Manhattan dist.	supremum dist.	cosine sim.
$\boldsymbol{x_1}$	0.1414	0.2	0.1	0.99999
$\boldsymbol{x_2}$	0.6708	0.9	0.6	0.99575
$\boldsymbol{x_3}$	0.2828	0.4	0.2	0.99997
$\boldsymbol{x_4}$	0.2236	0.3	0.2	0.99903
$\boldsymbol{x_5}$	0.6083	0.7	0.6	0.96536

These values produce the following rankings of the data points based on similarity:

Euclidean distance: x_1, x_4, x_3, x_5, x_2

Manhattan distance: x_1, x_4, x_3, x_5, x_2

Supremum distance: x_1, x_4, x_3, x_5, x_2

Cosine similarity: x_1, x_3, x_4, x_2, x_5

Solution-Problem 2(b)

(b) The normalized query is (0.65850, 0.75258). The normalized data set is given by the following table

	A_1	A_2
x_1	0.66162	0.74984
x_2	0.72500	0.68875
x_3	0.66436	0.74741
x_4	0.62470	0.78087
x_5	0.83205	0.55470

Recomputing the Euclidean distances as before yields

	Euclidean dist.
x_1	0.00415
x_2	0.09217
x_3	0.00781
x_4	0.04409
x_5	0.26320

which results in the final ranking of the transformed data points: x_1, x_3, x_4, x_2, x_5

6. What is cosine similarity?Problems on finding similarity between given documents.

Cosine Similarity

- A **document** can be represented by thousands of attributes, each recording the frequency of a particular word (such as keywords) or phrase in the document.

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- Applications: information retrieval, biologic taxonomy, gene feature mapping, ...
- Cosine measure: If x and y are two vectors (e.g., term-frequency vectors), then

  cos(x, y) = (x • y) /||x|| ||y|| ,

  where • indicates vector dot product, ||x||: the Euclidean norm of vector x, defined as

  $\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$. **= length of vector x**

- The Cosine measure computes the cosine of the angle between vectors x and y.
- A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match.
- The closer the cosine value to 1, the smaller the angle and the greater the match between vectors.

Example: Cosine Similarity

cos(x, y) = = (x • y) / ||x|| ||y|| ,
where • indicates vector dot product, ||d|: the length of vector d

Ex: Find the **similarity** between documents 1 and 2.

x = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)
y = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

x•y = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25

$||x|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

$||y|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$

cos(x, y) = 0.94 -----Quite similar

7. Problems based on finding dissimilarity matrices between nominal,binary and ordinal attributes .

Data Matrix and Dissimilarity Matrix

Data matrix *n*-by-*p* matrix (*n* objects *p* attributes) : 2 mode matrix.

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

Data Matrix and Dissimilarity Matrix

- **Dissimilarity matrix : n X n Matrix**
  - In general, $d(i, j)$ is a non-negative number that is close to 0 when objects i and j are highly similar or "near" each other, and becomes larger the more they differ.
  - Note that $d(i, i) = 0$; i.e. the difference between an object and itself is 0. Furthermore, $d(i, j) = d(j, i)$

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

  - Measures of similarity can often be expressed as a function of measures of dissimilarity. For example, for nominal data, $sim(i, j) = 1 - d(i, j)$ where $sim(i, j)$ is the similarity between objects i and j.
- Also called as **one-mode** matrix

Proximity Measure for Nominal Attributes

- A nominal attribute Can take 2 or more states,
- "How is dissimilarity computed between objects described by nominal attributes?"

Method 1: Simple matching :The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

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

Where

- p is the total number of attributes describing the objects.
- m is the number of matches (i.e., the number of attributes for which i and j are in the same state)

Proximity Measure for Nominal Attributes Example

: Dissimilarity between nominal attributes.

Object Identifier	test-1 (nominal)
1	code A
2	code B
3	code C
4	code A

Here p= 1
d(*i*, *j*) evaluates to 0 if objects *i* and *j* match, and 1 if the objects differ.

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

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}.$$

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

From this, we see that all objects are dissimilar except objects 1 and 4 (i.e.,d(4,1)= 0.

Proximity Measure for Nominal Attributes Example

: Dissimilarity between nominal attributes.

Alternatively, similarity can be computed as

$$sim(i,j)=1-d(i,j)=\frac{m}{p}.$$

Method 2: Use a large number of binary attributes

creating a new binary attribute for each of the M nominal states

how can we compute the dissimilarity between two binary attributes?

One approach involves computing a dissimilarity matrix from the given binary data. If all binary attributes are thought of as having the same weight, we have the 2 x 2 contingency table as shown below

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

Where q = the number of attributes that equal 1 for both objects *i* and *j*,

r = the number of attributes that equal 1 for object *i* but equal 0 for object *j*,

s = the number of attributes that equal 0 for object *i* but equal 1 for object *j*,

t = the number of attributes that equal 0 for both objects *i* and *j*.

p = The total number of attributes = $q + r + s + t$.

35

Proximity Measure for Binary Attributes

A contingency table for binary data

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

Distance measure for symmetric binary variables:

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

Distance measure for asymmetric binary variables:

$$d(i, j) = \frac{r + s}{q + r + s}$$

Jaccard coefficient (similarity measure for asymmetric binary variables):

$$sim_{Jaccard}(i, j) = \frac{q}{q + r + s}$$

= 1- d(i,j)

36

Dissimilarity between Binary Attributes

Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- *Gender is a symmetric attribute*
- *The remaining attributes are asymmetric binary*
- *Let the values Y and P be 1, and the value N be 0*

Suppose distance is computed based on only asymmetric attributes

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

$$d(i, j) = \frac{r+s}{q+r+s}$$

- These measurements suggest that Jim and Mary are unlikely to have a similar disease because they have the highest dissimilarity value among the three pairs.

Of the three patients, Jack and Mary are most likely to have a similar disease.

Example: Dissimilarity for Ordinal Variables

Object Identifier	test-2 (ordinal)
1	excellent
2	fair
3	good
4	excellent

Step 1: There are three states for test-2: fair, good, excellent, that is, Mf = 3.
Replace each value for test-2 by its rank, So objects 1 to 4 will be assigned ranks 3, 1, 2, and 3

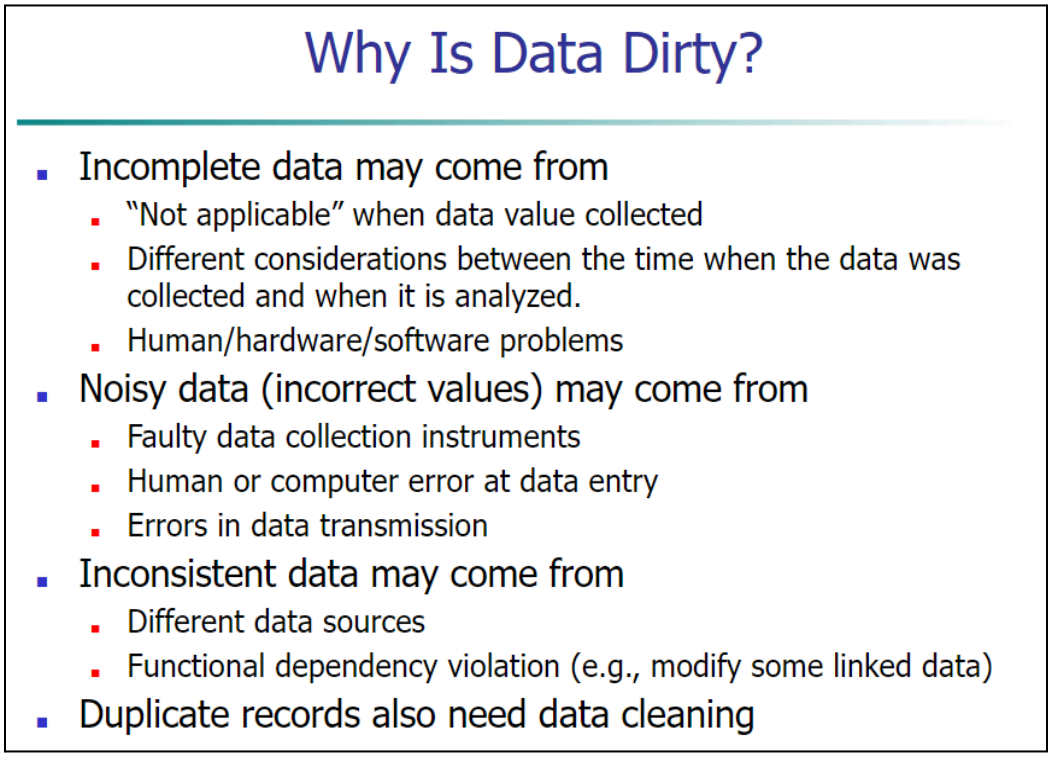
Step 2: Normalize the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0.

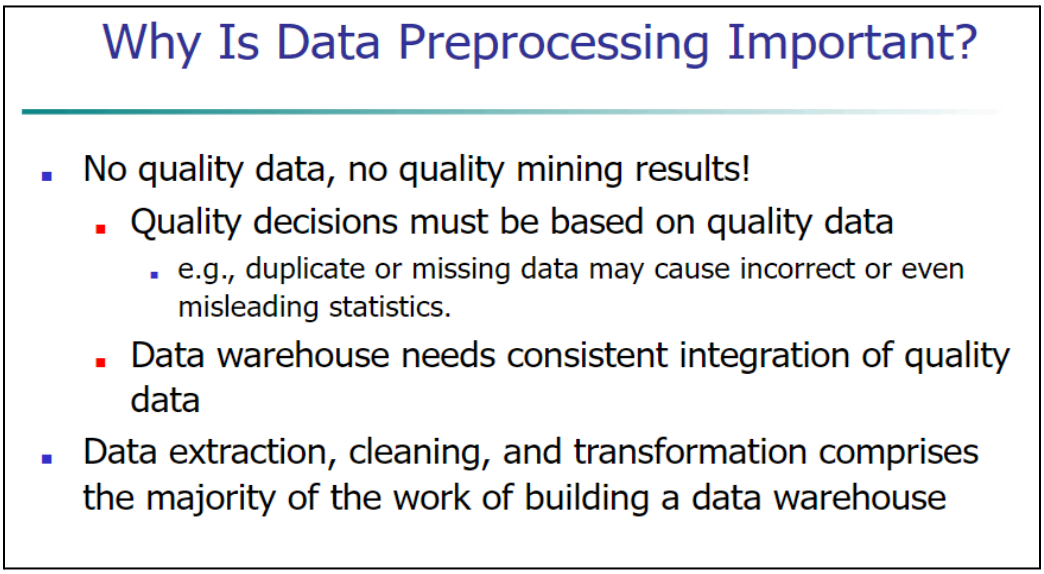
Step 3: we can use, say, the Euclidean distance , which results in the following dissimilarity matrix:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix} \end{array}$$

- Objects 1 and 2 are the most dissimilar, as are objects 2 and 4 (i.e., d(2,1)=1.0 and d(4,2)= 1.0).
- sim(i, j)= 1 - d(i,,j).

8. **Explain in brief the major tasks in data preprocessing.**
(Guys me extra bhi daal ke rakh rahi hu answer me, agar alag koi Q aaya to: Extra is in image form for this question)



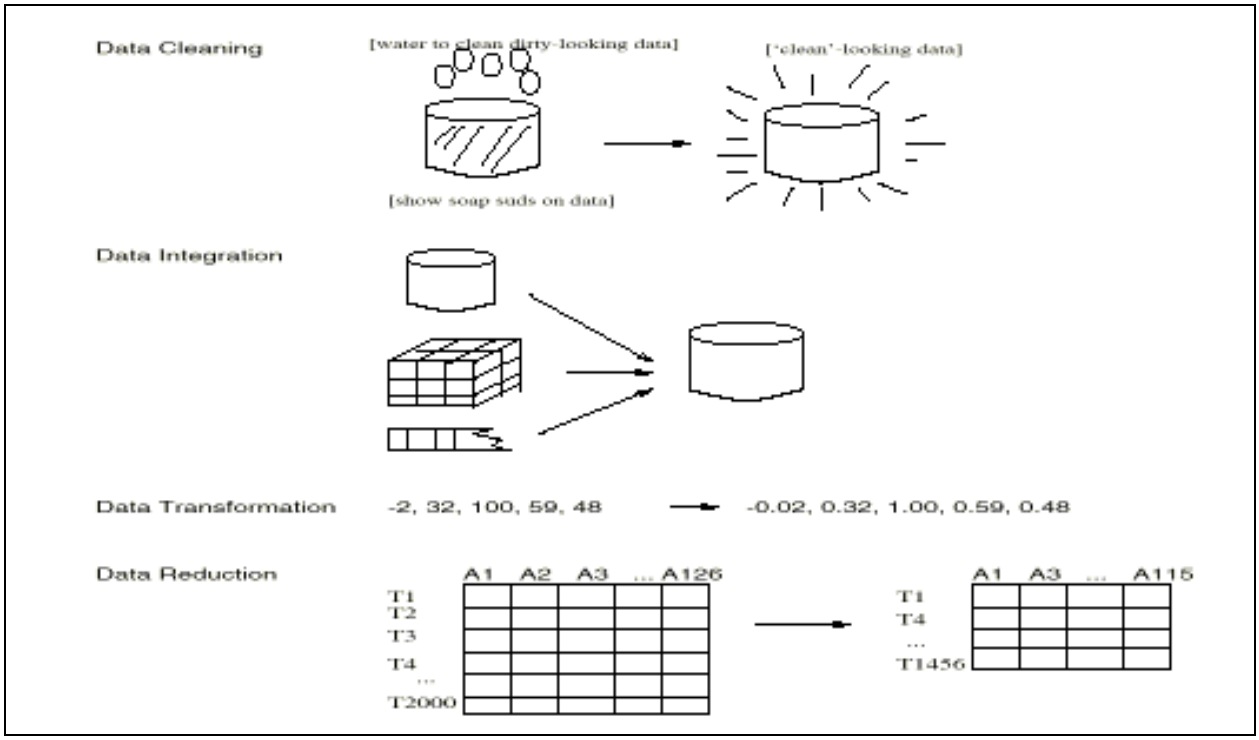


Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
  - Accuracy: degree to which information reflects an object
  - Completeness: when it fulfils expectations of comprehensives
  - Consistency: information stored at different places matches.
  - Timeliness: If information is available when we need it
  - Believability: How much data are trusted by users
  - Interpretability: How easily data are understood

Major Tasks in Data Preprocessing

- Data cleaning
  - Fill in missing values, smooth out the noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data



DATA CLEANING:

Data cleaning tasks-

1. **Fill in missing values**

   Data is not always available.

   E.g., many tuples have no recorded value for several attributes, such as customer income in sales data

   Missing data may be due to

   - Equipment malfunction
   - Inconsistent with other recorded data and thus deleted
   - Data not entered due to misunderstanding.
   - Certain data may not be considered important at the time of entry.

   Missing data may need to be inferred.

   Hande Missing Data-

   - Ignore the tuple: usually done when class label is missing assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
   - Fill in the missing value manually: (tedious + infeasible)
   - Use a global constant to fill in the missing value: Replace all missing attribute values by the same constant such as a label like “Unknown”
   - Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value.
   - Use the attribute mean or median for all samples belonging to the same class as the given tuple.

- Use The most probable value: inference-based such as Bayesian formula or decision tree

2. **Identify outliers and smooth out noisy data**

Noise: random error or variance in a measured variable.
Meaningless data that can not be interpreted by machines.
Noisy data (incorrect values) may come from-

- Faulty data collection instruments
- Human or computer error at data entry
- Errors in data transmission

Handle Noisy Data-

a. Binning-

Binning is a technique where we sort the data and then partition the data into equal frequency bins. Then you may either replace the noisy data with the bin mean bin median or the bin boundary.

There are three methods for smoothing data in the bin-

- Smoothing by bin mean method: In this method, the values in the bin are replaced by the mean value of the bin.
- Smoothing by bin median: In this method, the values in the bin are replaced by the median value.
- Smoothing by bin boundary: In this method, the minimum and maximum values of the bin values are taken, and the closest boundary value replaces the values.

Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (equi-depth) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by bin boundaries:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

b. Regression-

This is used to smooth the data and will help to handle data when unnecessary data is present.

For the analysis, purpose regression helps to decide the variable which is suitable for our analysis.

- Linear regression refers to finding the best line to fit between two variables so that one can be used to predict the other.
- Multiple linear regression involves more than two variables.

Using regression to find a mathematical equation to fit into the data helps to smooth out the noise.

c. Outlier Analysis-

Outliers may be detected by clustering, where similar or close values are organized into the same groups or clusters. Outliers are extreme values that deviate from other observations on data. Outliers can be the following kinds, such as:

- **Univariate outliers** can be found when looking at a distribution of values in a single feature space.
- **Multivariate outliers** can be found in an n-dimensional space (of n-features). Looking at distributions in n-dimensional spaces can be very difficult for the human brain. That is why we need to train a model to do it for us.
- **Point outliers** are single data points that lay far from the rest of the distribution.
- **Contextual outliers** can be noise in data, such as punctuation symbols when realizing text analysis or background noise signal when doing speech recognition.
- **Collective outliers** can be subsets of novelties in data, such as a signal that may indicate the discovery of new phenomena.

3. **Correct inconsistent data**
4. **Resolve redundancy caused by data integration**

Redundant data occur often while integrating multiple databases.

Unimportant data that are no longer required are referred to as redundant data.

- Object identification: The same attribute or object may have different names in different databases
- Derivable data: One attribute may be a “derived” attribute in another table, e.g., annual revenue,age
- Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

Some redundancies can be detected by **correlation analysis**.

Suppose we have a data set that has three attributes - pizza_name, is_veg, is_nonveg

1. Is_veg is 1; if the selecting pizza is veg else, it is 0.
2. Is_nonveg is 1; if the selecting pizza is nonveg else, it is 0.

On analyzing the above table, we have found that if a pizza is not veg (i.e., is_veg is 0 selecting the pizza_name), the pizza is surely non-veg (Since there are only two values in the pizza_name output class- Veg and Nonveg). Hence, one of these attributes became redundant. It means that the two attributes are very much related to each other, and one attribute can find the other. So, you can drop either the first or second attribute without any loss of information.

Age and DoB attribute.

Age can be derived from DoB.

DATA INTEGRATION:

Combines data from multiple sources into a coherent store to retain and provide a unified perspective of the data.

Data Integration Challenges/Issues: The semantic heterogeneity and structure of data.

1. Schema integration:

Integrate metadata from different sources.

e.g., A.cust-id ≡ B.cust-#

Analyzing metadata statistics will prevent you from making errors during schema integration.

2. Entity identification problem:

The problem of identifying object instances from different databases that correspond to the same real-world entity.

e.g., Bill Clinton = William Clinton.

3. Structural Integration:

When matching attributes from one database to another during integration, special attention must be paid to the structure of the data i.e. ensure that any attribute functional dependencies and referential constraints in the source system match those in the target system.

For example, you were given client data from specialized statistics sites. Customer identity is assigned to an entity from one statistics supply, while a customer range is assigned to an entity from another statistics supply. Analyzing such metadata statistics will prevent you from making errors during schema integration.

4. Redundancy and Correlation Analysis

Redundant data occur often while integrating multiple databases.

Some redundancies can be detected by correlation analysis.

Correlation analysis-

Given two attributes , correlation analysis can measure how strongly one attribute implies the other, based on the available data.

- For nominal data, we use the x2 (chi-square) test.
- For numeric attributes, we can use the correlation coefficient and covariance, both of which assess how one attribute values vary from those of another.

5. Tuple Duplication

In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level (e.g., where there are two or more identical tuples for a given unique data entry case).

DATA REDUCTION:

Why data reduction?

- A database/data warehouse may store terabytes of data
- Complex data analysis/mining may take a very long time to run on the complete data set

Data reduction
Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results

Data reduction strategies

- Dimensionality reduction — e.g., remove unimportant attributes
    - Wavelet Transform-
      The discrete wavelet transform (DW
      T) is a linear signal processing technique that, when applied to a data vector X, transforms it to a numerically different vector, X' of wavelet coefficients.
      The compressed data is obtained by retaining the smallest fragment of the strongest wavelet coefficients. Wavelet transform can be applied to data cubes, sparse data, or skewed data.

    - PCA
      Suppose we have a data set to be analyzed that has tuples with n attributes. The principal component analysis identifies k independent tuples with n attributes that can represent the data set.
      In this way, the original data can be cast on a much smaller space, and dimensionality reduction can be achieved. Principal component analysis can be applied to sparse and skewed data.

    - Attribute subset selection
      The attribute subset selection reduces the volume of data by eliminating redundant and irrelevant attributes.
      The most suitable subset of attributes are selected by using techniques like forward selection, backward elimination, decision tree induction or a combination of forward selection and backward elimination.
      The attribute subset selection ensures that we get a good subset of original attributes even after eliminating the unwanted attributes. The resulting probability of data distribution is as close as possible to the original data distribution using all the attributes.

- Numerosity reduction — e.g., fit data into models
  Reduce data volume by choosing alternative, smaller forms of data representation
    - Parametric methods: Regression, log-linear models
      Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
      Example: Log-linear models—obtain value at a point in m-D space as the product on appropriate marginal subspaces; Linear and multiple regression
    - Non parametric methods: histograms, clustering, sampling and data cube aggregation
      Do not assume models
      Methods- Clustering, Sampling, Histograms

- Data Compression-
    - String compression
      There are extensive theories and well-tuned algorithms
      Typically lossless
      But only limited manipulation is possible without expansion
    - Audio/video compression
      Typically lossy compression, with progressive refinement
      Sometimes small fragments of signal can be reconstructed without reconstructing the whole
    - Time sequence is not audio
      Typically short and vary slowly with time

DATA TRANSFORMATION:

- Data transformation is a technique used to convert the raw data into a suitable format that efficiently eases data mining and retrieves strategic information.
- Data transformation includes data cleaning techniques and a data reduction technique to convert the data into the appropriate form.
- Provide patterns that are easier to understand.
- Data transformation changes the format, structure, or values of the data and converts them into clean, usable data.
- Data Transformation Techniques-

1. Data Smoothing-
    - Data smoothing is a process that is used to remove noise from the dataset using some algorithms.
    - It allows for highlighting important features present in the dataset. It helps in predicting the patterns.
    - When collecting data, it can be manipulated to eliminate or reduce any variance or any other noise form.
    - The concept behind data smoothing is that it will be able to identify simple changes to help predict different trends and patterns. This serves as a help to analysts or traders who need to look at a lot of data which can often be difficult to digest for finding patterns that they wouldn't see otherwise.
    - The noise is removed from the data using the techniques such as binning, regression, clustering.
2. Data Aggregation-
    - Data collection or aggregation is the method of storing and presenting data in a summary format.
    - This is a crucial step since the accuracy of data analysis insights is highly dependent on the quantity and quality of the data used.
    - Gathering accurate data of high quality
3. Data Generalization-

- It converts low-level data attributes to high-level data attributes using concept hierarchy.
- This conversion from a lower level to a higher conceptual level is useful to get a clearer picture of the data. Data generalization can be divided into two approaches:

    - Data cube process (OLAP) approach.
    - Attribute-oriented induction (AOI) approach.
- For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).

4. Data Transformation (by Normalization)-

Data scaled to fall within a small, specified range.

It can be performed by three methods-

a. Min-max Normalization
b. Z-score Normalization
c. Decimal scaling Normalization

5. Attribute Construction-

- In the attribute construction method, the new attributes consult the existing attributes to construct a new data set that eases data mining.
- New attributes are created and applied to assist the mining process from the given attributes. This simplifies the original data and makes the mining more efficient.
- For example, suppose we have a data set referring to measurements of different plots, i.e., we may have the height and width of each plot. So here, we can construct a new attribute 'area' from attributes 'height' and 'weight'.
- Attribute construction also helps understand the relations among the attributes in a data set.

DATA DISCRETIZATION:

- This is a process of converting continuous data into a set of data intervals. Continuous attribute values are substituted by small interval labels.
- This makes the data easier to study and analyze.
- If a data mining task handles a continuous attribute, then its discrete values can be replaced by constant quality attributes. This improves the efficiency of the task.
- This method is also called a data reduction mechanism as it transforms a large dataset into a set of categorical data.
- Discretization also uses decision tree-based algorithms to produce short, compact, and accurate results when using discrete values.
- For example, the values for the age attribute can be replaced by the interval labels such as (0-10, 11-20…) or (kid, youth, adult, senior).
- Data discretization methods-

1. Binning-

    - Binning is a top-down splitting technique based on a specified number of bins.
    - These methods are also used as discretization methods for data reduction and concept hierarchy generation.
    - For example, attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median, as in smoothing by bin means or smoothing by bin medians, respectively. These techniques can be applied recursively to the resulting partitions to generate concept hierarchies.
    - It is unsupervised discretization technique.

2. Histogram Analysis-
   - Like binning, histogram analysis is an unsupervised discretization technique.
   - Various partitioning rules(equal width/equal size) can be used to define histograms.
   - The histogram analysis algorithm can be applied recursively to each partition in histogram in order to automatically generate a multilevel concept hierarchy, with the procedure terminating once a prespecified number of concept levels has been reached.
   - A minimum interval size can also be used per level to control the recursive procedure. This specifies the minimum width of a partition, or the minimum number of values for each partition at each level.

3. Clustering-
   - Clustering, decision tree analysis, and correlation analysis can be used for data discretization.
   - A clustering algorithm can be applied to discretize a numeric attribute, A, by partitioning the values of A into clusters or groups. Clustering takes the distribution of A into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results.
   - Clustering can be used to generate a concept hierarchy for A by following either a top-down splitting strategy or a bottom-up merging strategy, where each cluster forms node of the concept hierarchy.

4. Decision Tree-
   - Techniques to generate decision trees for classification can be applied to discretization.
   - These techniques employ a top-down splitting approach.
   - Decision tree approaches to discretization are supervised.

5. Correlation Analysis-
   - Measures of correlation can be used for discretization.
   - ChiMerge is a x2-based discretization method which employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively.
   - Supervised method
   - The basic notion is that for accurate discretization, the relative class frequencies should be fairly consistent within an interval. Therefore, if two adjacent intervals have a very similar distribution of classes, then the intervals can be merged. Otherwise, they should remain separate.

9. Problems based on finding correlation between attributes (Chi Square test, Pearson correlation coefficient , covariance etc…)

CHI SQUARE TEST:

x^2 Correlation Test for Nominal Data

Suppose attribute *A* has *c* distinct values, namely $a_1, a_2, \ldots a_c$.
Attribute B has *r* distinct values, namely $b_1, b_2, \ldots b_r$.
Let (A_i, Bj) denote the joint event that attribute *A* takes on value a_i and attribute *B* takes on value b_j

The x^2 value (also known as the *Pearson* x^2 *statistic*) is computed as:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

where o_{ij} is the *observed frequency* (i.e., actual count) of the joint event (A_i, B_j) and e_{ij} is the *expected frequency* of (A_i, B_j), which can be computed as $e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n}$,

where n is the number of data tuples, $count(A = a_i)$ is the number of tuples having value a_i for A, and $count(B = b_j)$ is the number of tuples having value b_i for B.

x^2 Correlation Test for Nominal Data

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}, \qquad e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n},$$

The x^2 statistic tests the hypothesis that *A* and *B* are *independent*, that is, there is no correlation between them.

The test is based on a significance level, with (*r*-1)x(*c-1*) degrees of freedom.

If the sample statistic x^2 > tabulated statistics $x^2_{\alpha,dof}$, the null hypothesis that A and B are independent is rejected, So, we say that *A* and *B* are statistically correlated.

- **DOF: No of values in the final calculation of the statistic that are free to vary.**
- **Level of significance: Prob of rejecting the null hypothesis when it is true.**

	P										
DF	0.995	0.975	0.2	0.1	0.05	0.025	0.02	0.01	0.005	0.002	0.001
1	.0004	.00016	1.642	2.706	3.841	5.024	5.412	6.635	7.879	9.55	10.828
2	0.01	0.0506	3.219	4.605	5.991	7.378	7.824	9.21	10.597	12.429	13.816
3	0.0717	0.216	4.642	6.251	7.815	9.348	9.837	11.345	12.838	14.796	16.266
4	0.207	0.484	5.989	7.779	9.488	11.143	11.668	13.277	14.86	16.924	18.467
5	0.412	0.831	7.289	9.236	11.07	12.833	13.388	15.086	16.75	18.907	20.515
6	0.676	1.237	8.558	10.645	12.592	14.449	15.033	16.812	18.548	20.791	22.458
7	0.989	1.69	9.803	12.017	14.067	16.013	16.622	18.475	20.278	22.601	24.322
8	1.344	2.18	11.03	13.362	15.507	17.535	18.168	20.09	21.955	24.352	26.124
9	1.735	2.7	12.242	14.684	16.919	19.023	19.679	21.666	23.589	26.056	27.877
10	2.156	3.247	13.442	15.987	18.307	20.483	21.161	23.209	25.188	27.722	29.588
11	2.603	3.816	14.631	17.275	19.675	21.92	22.618	24.725	26.757	29.354	31.264
12	3.074	4.404	15.812	18.549	21.026	23.337	24.054	26.217	28.3	30.957	32.909
13	3.565	5.009	16.985	19.812	22.362	24.736	25.472	27.688	29.819	32.535	34.528
14	4.075	5.629	18.151	21.064	23.685	26.119	26.873	29.141	31.319	34.091	36.123
15	4.601	6.262	19.311	22.307	24.996	27.488	28.259	30.578	32.801	35.628	37.697
16	5.142	6.908	20.465	23.542	26.296	28.845	29.633	32	34.267	37.146	39.252
17	5.697	7.564	21.615	24.769	27.587	30.191	30.995	33.409	35.718	38.648	40.79
18	6.265	8.231	22.76	25.989	28.869	31.526	32.346	34.805	37.156	40.136	42.312
19	6.844	8.907	23.9	27.204	30.144	32.852	33.687	36.191	38.582	41.61	43.82
20	7.434	9.591	25.038	28.412	31.41	34.17	35.02	37.566	39.997	43.072	45.315

Example1: Correlation Analysis of Nominal attributes using x^2

2 x 2 Contingency Table Data

	male	female	Total
fiction	250	200	450
non_fiction	50	1000	1050
Total	300	1200	1500

Note: Are gender and preferred_reading correlated?

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n},$$

$$e_{11} = \frac{count(male) \times count(fiction)}{n} = \frac{300 \times 450}{1500} = 90,$$

2 x 2 Contingency Table Data

	male	female	Total
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Note: Are gender and preferred_reading correlated?

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93.$$

- For this 2x2 table, the degrees of freedom are (2-1)(2-1)= 1.
- For 1 degree of freedom , the x^2 value needed to reject the hypothesis at the 0.001 significance level is **10.828** (From Chi square distribution table)
- Since our computed value is greater than tabulated value, we can reject the hypothesis that *gender* and *preferred reading* are independent and conclude that the two attributes are (strongly) correlated for the given group of people.

Example 2

A food services manager for a baseball park wants to know if there is a relationship between gender (male or female) and the preferred condiment on a hot dog. The following table summarizes the results. Test the hypothesis with a significance level of 10%.

		Condiment			
		Ketchup	Mustard	Relish	Total
Gender	Male	15	23	10	48
	Female	25	19	8	52
	Total	40	42	18	100

Step 1: The hypotheses are:

H0: Gender and condiments are independent

- H1 : Gender and condiments are not independent

Example 2

A food services manager for a baseball park wants to know if there is a relationship between gender (male or female) and the preferred condiment on a hot dog. The following table summarizes the results. Test the hypothesis with a significance level of 10%.

		Condiment			
		Ketchup	Mustard	Relish	Total
Gender	Male	15	23	10	48
	Female	25	19	8	52
	Total	40	42	18	100

Step 2: Find expected frequencies table

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n},$$

		Condiment			
		Ketchup	Mustard	Relish	Total
Gender	Male	15 (19.2)	23 (20.16)	10 (8.64)	48
	Female	25 (20.8)	19 (21.84)	8 (9.36)	52
	Total	40	42	18	100

Step 3: Calculate Chi square statistic

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

$$\chi^{2*} = \frac{(15-19.2)^2}{19.2} + \frac{(23-20.16)^2}{20.16} + \ldots + \frac{(8-9.36)^2}{9.36} = 2.95$$

Step 4: DoF: (c-1)(r-1) =3 x 1 =3

Step 5: Compare calculated test statistic with tabulated one.
Her Tabulated statistic(with significance level =0.01) = 11.345 > 2.95 .
Hence H0 is accepted. i.e attributes are independent.

Correlation Analysis - Hypotheses

- H_0 : Null Hypothesis - attributes are not related (ie Independent)
- H_1 : Alternate Hypothesis - attributes are related (ie dependent)

- Degree of freedom = (#rows-1 *#columns -1)
- Given df and chi :
  - Find the p-value in Chi squared distribution table depending on the prob and chi value
  - The p-value *100 gives the % of likelihood
- Given the df and p-value
  - Get the chi value from the table
  - Reject the null hypothesis if the value of chi is higher than the chi value obtained form tale
  - Else accept the null hypothesis if the obtained value of chi square is less than the tabular value.

	Right-Tail Probability				
df	0.250	0.100	0.050	0.025	0.010
1	1.32	2.71	3.84	5.02	6.63
2	2.77	4.61	5.99	7.38	9.21
3	4.11	6.25	7.81	9.35	11.34
4	5.39	7.78	9.49	11.14	13.28
5	6.63	9.24	11.07	12.83	15.09

Chi-Square Calculation: Example 1

	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- H_0: Science fiction is not associated with playing chess, and
- H_1: Science fiction is associated with playing chess

- df = (2-1)*(2-1) = 1
- X^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- P-value comes out to be 0 i.e there is 0 % likelihood of null hypothesis to be true.
- It shows that like_science_fiction and play_chess are correlated in the group

Chi-Square Calculation: Example 3

A group of students were classified in terms of personality (introvert or extrovert) and in terms of color preference (red, yellow, green or blue) with the purpose of seeing whether there is an association (relationship) between personality and color preference. Data was collected from 400 students and presented in the 2 (rows) x 4 (cols) contingency table below:

(Observed counts)	Colors				
	Red	**Yellow**	**Green**	**Blue**	**Totals**
Introvert personality	20	6	30	44	100
Extrovert personality	180	34	50	36	300
Totals	200	40	80	80	400

Suitable null and alternative hypotheses might be:

- H_0: Color preference is not associated with personality, and$X2 = 71.20$
- H_1: Color preference is associated with personality

To perform a chi-squared test, the number of students expected in each cell of the table if the null hypothesis is true, is calculated.

3

Chi-Square Calculation: Example2

A group of students were classified in terms of personality (introvert or extrovert) and in terms of color preference (red, yellow, green or blue) with the purpose of seeing whether there is an association (relationship) between personality and color preference. Data was collected from 400 students and presented in the 2 (rows) x 4 (cols) contingency table below. Check whether color and personality attributes are correlated.

(Observed counts)	Colors				
	Red	**Yellow**	**Green**	**Blue**	**Totals**
Introvert personality	20	6	30	44	100
Extrovert personality	180	34	50	36	300
Totals	200	40	80	80	400

X2 = 71.20

Correlation Analysis (Numerical Data)

- For numeric attributes, we can evaluate the correlation between two attributes, *A* and *B*,by computing the **correlation coefficient** (also known as **Pearson's product moment coefficient**

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B},$$

where n is the number of tuples, a_i and b_i are the respective values of A and B in tuple i, $\bar{A}$ and $\bar{B}$ are the respective mean values of A and B, σ_A and σ_B are the respective standard deviations of A and B and $\Sigma(a_i b_i)$ is the sum of the AB

Note that $-1 \le r_{A,B} \le +1$.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher the value, the stronger correlation. Hence, a higher value may indicate that *A* (or *B*) may be removed as a redundancy.
- $r_{A,B} = 0$: independent(no correlation);
- $r_{A,B} < 0$: negatively correlated (A's values increase as B's decrease). This means that each attribute discourages the other

Covariance for Numerical Data

In probability theory and statistics, correlation and covariance are two similar measures for assessing how much two attributes change together.

Consider two numeric attributes *A* and *B*, and a set of *n* observations {(*a*1,*b*1),(a2,b2)…, (*an*,*bn)}*

The mean values of *A* and *B*, respectively, are also known as the **expected values** on *A* and *B*, that is, $E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n}$ and $E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}.$

The **covariance** between *A* and *B* is defined as

$$Cov(A,B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n}.$$

Also it can be shown that

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}.$$

For two attributes *A* and *B* that tend to change together, if *A* is larger than $\bar{A}$ (the expected value of *A*), then *B* is likely to be larger than $\bar{B}$ (the expected value of *B*).
Therefore, the covariance between *A* and *B* is *positive*.

On the other hand, if one of the attributes tends to be above its expected value when the other attribute is below its expected value, then the covariance of *A* and *B* is *negative*.

If *A* and *B* are *independent* (i.e., they do not have correlation), then $E(A.B) = E(A).E(B)$
Therefore, the covariance is $Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B} = E(A) \cdot E(B) - \bar{A}\bar{B} = 0.$

However, the converse is not true. Some pairs of random variables (attributes) may have a covariance of 0 but are not independent

Example: Covariance for Numerical Data

Stock Prices for *AllElectronics* and *HighTech*

Time point	AllElectronics	HighTech
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

if the stocks are affected by the same industry trends will their prices rise or fall together?

$$E(AllElectronics) = \frac{6+5+4+3+2}{5} = \frac{20}{5} = \$4$$

and

$$E(HighTech) = \frac{20+10+14+5+5}{5} = \frac{54}{5} = \$10.80.$$

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}.$$

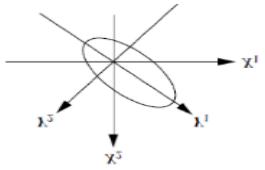
$$Cov(AllElectroncis, HighTech) = \frac{6 \times 20 + 5 \times 10 + 4 \times 14 + 3 \times 5 + 2 \times 5}{5} - 4 \times 10.80$$

$$= 50.2 - 43.2 = 7.$$

- Therefore, given the positive covariance we can say that stock prices for both companies rise together.

Dimensionality Reduction: Principal Component Analysis (PCA)

- Given *N* data vectors from *n*-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
- Steps
  - Normalize input data: Each attribute falls within the same range
  - Compute *k* orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the *k* principal component vectors
  - The principal components are sorted in order of decreasing "significance" or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the weak components, i.e., those with low variance. (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data
- Works for numeric data only
- Used when the number of dimensions is large



Dimensionality Reduction: Principal Component Analysis (PCA)-Example

Given the following data, use PCA to reduce dimensions from 2 to 1

Feature	Ex1	Ex2	Ex3	Ex4
X	4	8	13	7
Y	11	4	5	14

Step1: For given dataset, n=No. of features=2
N: No. of samples =4

Step2: Compute mean of variables, $\bar{X} = 8$ $\bar{Y}$ =8.5

Step3.1: Find covariance of all ordered pairs (x,x),(x,y),(y,x) and (y,y)

The **covariance** between A and B is defined as

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}.$$

Cov(x,x)=(1/(4-1)) *(4-8)(4-8)+(8-8)(8-8)+(13-8)(13-8)+(7-8)(7-8)=14
Cov(x,y)=(1/(4-1)) *(4-8)(11-8.5)+(8-8)(4-8.5)+(13-8)(5-8.5)+(7-8)(14-8.5)=-11
Cov(y,x)=Cov(x,y)= -11
Cov(y,y)=23

PCA Example (Cntd...)

Step 3.2 Find Covariance matrix n x n i.e 2 x 2

$$S = \begin{bmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{bmatrix} = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step 4: : Find Eigen Value , Eigen Vector and Normalized Eigen Vector

1) Find Eigen Value λ : $\det(S-\lambda I)=0$

$\lambda = 30.3849, 6.6151$. Hence $\lambda_1 = 30.3849$, $\lambda_2 = 6.6151$

[Longest Eigen value is first principal component]

2) Find Eigen Vector of λ_1 (U1) : $(S-\lambda_1 I)U_1 = 0$

$$\begin{bmatrix} 14-\lambda_1 & -11 \\ -11 & 23-\lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0.$$

Solving, U1= $\begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$

3) Normalize the eigen vector U1 $e_1 = \begin{bmatrix} 11/\sqrt{11^2 + (-16.3849)^2} \\ -16.3849/\sqrt{11^2 + (-16.3849)^2} \end{bmatrix} = \begin{bmatrix} 0.5371 \\ -0.8433 \end{bmatrix}$

PCA Example (Cntd...)

- Similarly find Eigen vector U2 and normalized eigen vector e2

$$e_2 = \begin{bmatrix} 0.8305 \\ 0.5574 \end{bmatrix}$$

- Step 5: Derive new dataset

	Ex1	Ex2	Ex3	Ex4
First Principal Component PC1	P11	P12	P13	P14

- P11= $e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix} = -4.3052$
- P12=3.7361
- P13=5.6928
- P14=-5.1238
- Dataset with 1 feature is:

	Ex1	Ex2	Ex3	Ex4
First Principal Component PC1	-4.3052	3.7361	5.6928	5.1238

Module 3: Classification

1. Write Decision Tree algorithm :ID3,C4.5 and CART algorithms

Algorithm for Decision Tree Induction

1. The tree starts as a single node, *N*, representing the training tuples in *D*
2. If the tuples in *D* are all of the same class, then node *N* becomes a leaf and is labeled with that class.
3. Otherwise, the algorithm calls *Attribute selection method* to determine the **splitting criterion**.
   - The splitting criterion tells us which attribute to test at node *N* by determining the "best" way to separate or partition the tuples in *D* into individual classes.
   - The splitting criterion also tells us which branches to grow from node *N* with respect to the outcomes of the chosen test.
   - More specifically, the splitting criterion indicates the **splitting attribute** and may also indicate either a **split-point** or a **splitting subset**.
   - The splitting criterion is determined so that, ideally, the resulting partitions at each branch are as "pure" as possible.
   - A partition is **pure** if all the tuples in it belong to the same class.

Attribute Selection Measures

(Also known as Splitting Rules)

- An **attribute selection measure** is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D* , of class-labeled training tuples into individual classes.
- The attribute selection measure provides a ranking for each attribute describing the given training tuples.
- The three popular attribute selection measures—

1. *information gain*
2. *gain ratio*, and
3. *Gini index*

Attribute Selection Measures: Information Gain (ID3 algorithm)

This measure is based on pioneering work by Claude Shannon on information theory, which studied the **value or "information content"** of messages.

The attribute with the highest information gain is chosen as the splitting attribute for node *N*.

This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions.

Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found.

Attribute Selection Measure: Information Gain (ID3)

- Select the attribute with the highest information gain.
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

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

- Information needed (after using A to split D into v partitions) to classify D:

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

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Highest Information gain

Information Gain (ID3)

Let D, the data partition, be a training set of class-labeled tuples. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for $i = 1, \ldots, m$). Let $C_{i,D}$ be the set of tuples of class C_i in D. Let $|D|$ and $|C_{i,D}|$ denote the number of tuples in D and $C_{i,D}$, respectively.

Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by

$$p_i = |C_{i,D}|/|D|$$

Expected information (entropy) needed to classify a tuple in D:

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

Information needed (after using A to split D into v partitions) to classify D:

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

Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

- Select the attribute with the highest information gain.

Computing Information-Gain for Continuous-Value Attributes

- Let attribute A be a continuous-valued attribute
- Must determine the ***best split point*** for A
  - Sort the values of A in increasing order.
  - Typically, the midpoint between each pair of adjacent values where there is a change in classification is considered as a possible *split point*
    - $(a_i + a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
  - The point with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
  - D1 is the set of tuples in D satisfying A ≤ split-point, and D2 is the set of tuples in D satisfying A > split-point

Gain Ratio for Attribute Selection (C4.5)

- 'Information gain' measure is biased towards attributes with a large number of values(e.g. Unique ID attribute like Product_ID)
- C4.5 uses an extension to information gain known as ***gain ratio***, which attempts to overcome this bias.
- It applies a kind of normalization to information gain using a "split information" value defined analogously with *Info(D)* as

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- This value represents the potential information generated by splitting the training data set, *D*, into *v* partitions, corresponding to the *v* outcomes of a test on attribute *A*.

- The gain ratio is defined as
  - GainRatio(A) = Gain(A)/$SplitInfo_A$(D)
- The attribute with the maximum gain ratio is selected as the splitting attribute

Gini index (CART)

- Gini index measures the impurity of *D*, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

where p_i is the probability that a tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. The sum is computed over m classes.

- The Gini index considers a binary split for each attribute.

- **Case 1: A is discrete-valued:** To determine the best binary split on discrete valued attribute *A*, we examine all the possible subsets of the known values of *A*.
- Each subset, S_A, can be considered as a binary test for attribute *A* of the form "$A \in S_A$?"
- If *A* has *v* possible values, then there are 2^v possible subsets.
- Out of these, there are 2^v -2 possible ways to form two partitions of the data, *D*, based on a binary split on *A*.

Gini index (CART)

- When considering a binary split, we compute a weighted sum of the impurity of each resulting partition. For example, if a binary split on A partitions D into D_1 and D_2, the Gini index of D given that partitioning is

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2).$$

- For each attribute, each of the possible binary splits is considered.
- For a discrete-valued attribute, the subset that gives the **minimum Gini index** for that attribute is selected as its splitting subset.
- The reduction in impurity that would be incurred by a binary split on a discrete- or continuous-valued attribute A is

$$\Delta Gini(A) = Gini(D) - Gini_A(D).$$

- The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute. This attribute and its splitting subset (for a discrete-valued splitting attribute) together form the splitting criterion.

Gini index (Continuous-valued attribute)

- For continuous-valued attributes, each possible split-point must be considered.
- The strategy is to take the midpoint between each pair of (sorted) adjacent values as a possible split-point.
- The point giving the minimum Gini index for a given (continuous-valued) attribute is taken as the split-point of that attribute.
- Recall that for a possible split-point of *A*, *D*1 is the set of tuples in *D* satisfying *A* <= *split point*, and *D*2 is the set of tuples in *D* satisfying *A* > *split point*
- The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute. This attribute and its split-point (for a continuous-valued splitting attribute) together form the splitting criterion.

Comparing Attribute Selection Measures

- Information gain:
  - biased towards multivalued attributes
- Gain ratio:
  - tends to prefer unbalanced splits in which one partition is much smaller than the others
- Gini index:
  - biased to multivalued attributes
  - has difficulty when # of classes is large.
  - tends to favor tests that result in equal-sized partitions and purity in both partitions
- Although biased, these measures give reasonably good results in practice.

2. Explain Attribute selection measures (Information Gain, Gain Ratio, Gini Index)

(Refer the above answer)

Information Gain

- Information gain measures the reduction in entropy or variance that results from splitting a dataset based on a specific property.
- It is used in decision tree algorithms to determine the usefulness of a feature by partitioning the dataset into more homogeneous subsets with respect to the class labels or target variable.
- The higher the information gain, the more valuable the feature is in predicting the target variable.

The information gain of an attribute A, with respect to a dataset D, is calculated as follows:

Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by

$$p_i = |C_{i,D}|/|D|$$

Expected information (entropy) needed to classify a tuple in D:

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

Information needed (after using A to split D into v partitions) to classify D:

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

Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

- Information gain measures the reduction in entropy or variance achieved by partitioning the dataset on attribute A.
- The attribute that maximizes information gain is chosen as the splitting criterion for building the decision tree.

Information gain is used in both classification and regression decision trees.

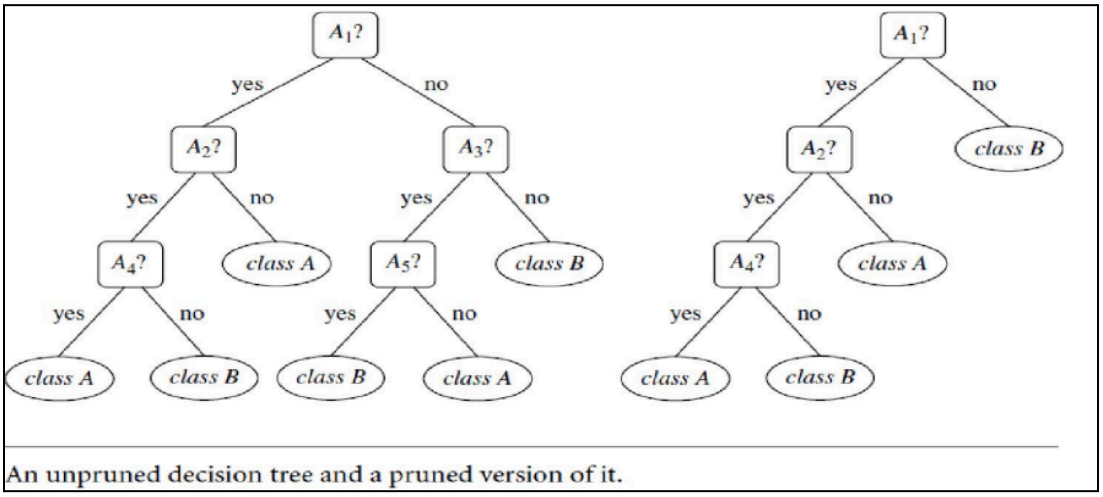
In classification, entropy is used as a measure of impurity, while in regression, variance is used as a measure of impurity. The information gain calculation remains the same in both cases, except that entropy or variance is used instead of entropy in the formula.

3. What is Overfitting and Tree Pruning?

Overfitting

- Overfitting is a common problem that needs to be handled while training a decision tree model.
- Overfitting occurs when a model fits too closely to the training data and may become less accurate when encountering new data or predicting future outcomes.
- In an overfit condition, a model memorizes the noise of the training data and fails to capture essential patterns.

- In decision trees, In order to fit the data (even noisy data), the model keeps generating new nodes and ultimately the tree becomes too complex to interpret. The decision tree predicts well for the training data but can be inaccurate for new data. If a decision tree model is allowed to train to its full potential, it can overfit the training data.
- Tree pruning methods address this problem of overfitting the data.



An unpruned decision tree and a pruned version of it.

Tree Pruning

Pruning is a technique that removes parts of the decision tree and prevents it from growing to its full depth. Pruning removes those parts of the decision tree that do not have the power to classify instances. Pruning can be of two types — Pre-Pruning and Post-Pruning.

Two approaches to avoid overfitting : **Prepruning and Postpruning**

- **Prepruning:**
  - Halt tree construction early (e.g., by deciding not to further split or partition the subset of training tuples at a given node).
  - Upon halting, the node becomes a leaf.
  - The leaf may hold the most frequent class among the subset tuples or the probability distribution of those tuples.
    - Difficult to choose an appropriate threshold
- **Postpruning:**
  - Remove branches from a "fully grown" tree—A subtree at a given node is pruned by removing its branches and replacing it with a leaf.
    - The leaf is labeled with the most frequent class among the subtree being replaced.

4. State Bayes Theorem.

Bayes' Theorem

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}$$

Where X and Y are the events and P (Y) ≠ 0

Where X and Y are the events and P (Y) ≠ 0

P(X/Y) is a conditional probability that describes the occurrence of event X is given that Y is true.
P(Y/X) is a conditional probability that describes the occurrence of event Y is given that X is true.
P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the marginal probability.

5. Explain Naïve Bayesian Classification Algorithm with example .

Naïve Bayesian Classification Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Naïve Bayesian Classification

1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, $A_1, A_2, \ldots, A_n$.

2. Suppose that there are m classes, $C_1, C_2, \ldots, C_m$. Given a tuple, $\boldsymbol{X}$, the classifier will predict that $\boldsymbol{X}$ belongs to the class having the highest posterior probability, conditioned on $\boldsymbol{X}$. That is, the naïve Bayesian classifier predicts that tuple $\boldsymbol{X}$ belongs to the class C_i if and only if

$$P(C_i|\boldsymbol{X}) > P(C_j|\boldsymbol{X}) \quad \text{for } 1 \le j \le m, j \ne i.$$

Thus, we maximize $P(C_i|\boldsymbol{X})$. The class C_i for which $P(C_i|\boldsymbol{X})$ is maximized is called the *maximum posteriori hypothesis*. By Bayes' theorem

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})}$$

3. As $P(\boldsymbol{X})$ is constant for all classes, only $P(\boldsymbol{X}|C_i)P(C_i)$ needs to be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = \cdots = P(C_m)$, and we would therefore maximize $P(\boldsymbol{X}|C_i)$. Otherwise, we maximize $P(\boldsymbol{X}|C_i)P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i) = |C_{i,D}|/|D|$, where $|C_{i,D}|$ is the number of training tuples of class C_i in D.

Naïve Bayesian Classification(Cntd..

4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(\boldsymbol{X}|C_i)$. To reduce computation in evaluating $P(\boldsymbol{X}|C_i)$, the naïve assumption of **class-conditional independence** is made. This presumes that the attributes' values are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$
$$= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i).$$

(a) If A_k is categorical, then $P(x_k|C_i)$ is the number of tuples of class C_i in D having the value x_k for A_k, divided by $|C_{i,D}|$, the number of tuples of class C_i in D.

(b) If A_k is continuous-valued, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ, defined by

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

so that

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

Naïve Bayesian Classification(Cntd..

5. To predict the class label of X, $P(X|C_i)P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of tuple X is the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \leq j \leq m, j \neq i.$$

In other words, the predicted class label is the class C_i for which $P(X|C_i)P(C_i)$ is the maximum.

Example

Suppose we have a dataset of weather conditions and corresponding target variable "Play". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

Problem: If the weather is sunny, then the Player should play or not?

Solution: To solve this, first consider the below dataset:

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Frequency Table of Weather Condition

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Applying Bayes'theorem

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)

P(Sunny|Yes)= 3/10= 0.3
P(Sunny)= 0.35
P(Yes)=0.71
So P(Yes|Sunny) = 0.3*0.71/0.35= 0.60
P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)
P(Sunny|NO)= 2/4=0.5
P(No)= 0.29
P(Sunny)= 0.35
So P(No|Sunny)= 0.5*0.29/0.35 = 0.41

So as we can see from the above calculation that P(Yes|Sunny)>P(No|Sunny)
Hence on a Sunny day, Player can play the game.

6. State advantages and disadvantages of Naive Bayes Algorithm

Advantages

- **Less complex:** Compared to other classifiers, Naïve Bayes is considered a simpler classifier since the parameters are easier to estimate. As a result, it's one of the first algorithms learned within data science and machine learning courses.
- **Scales well:** Compared to logistic regression, Naïve Bayes is considered a fast and efficient classifier that is fairly accurate when the conditional independence assumption holds. It also has low storage requirements.
- **Can handle high-dimensional data:** Use cases, such document classification, can have a high number of dimensions, which can be difficult for other classifiers to manage.

Disadvantages:

- **Subject to Zero frequency:** Zero frequency occurs when a categorical variable does not exist within the training set. For example, imagine that we're trying to find the maximum likelihood estimator for the word, "sir" given class "spam", but the word, "sir" doesn't exist in the training data. The probability in this case would be zero, and since this classifier multiplies all the conditional probabilities together, this also means that posterior probability will be zero. To avoid this issue, laplace smoothing can be leveraged.
- **Unrealistic core assumption:** While the conditional independence assumption overall performs well, the assumption does not always hold, leading to incorrect classifications.

7. What are the different Metrics for Evaluating Classifier Performance(Accuracy, Precision, Recall, F1 score,Specificity, Sensitivity)

There are 6 terms

Metrics for Evaluating Classifier Performance

- Terminologies:
  - **Positive Tuples:** tuples of the main class of interest
  - **Negative Tuples:** All other tuples.
    (e.g. Given two classes , the positive tuples may be buys computer = yes while the negative tuples are buys computer = no.)
  - **True Positives(TP):**These refer to the positive tuples that were correctly labeled by the classifier.
  - **True Negatives(TN):** These are the negative tuples that were correctly labeled by the classifier.
  - **False Positives(FP):**These are the negative tuples that were incorrectly labeled as positive
    (e.g., tuples of class buys computer = no for which the classifier predicted buys computer = yes).
  - **False Negatives(FN):**These are the positive tuples that were mislabeled as negative (e.g., tuples of class buys computer = yes for which the classifier predicted buys computer = no).

  These terms are summarized in a **Confusion Matrix**

1

Confusion Matrix (CM): The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of different classes.

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Now the metrics for evaluating classifier performance

1) Accuracy of a classifier M, acc(M):

Percentage of test set tuples that are correctly classified by the mode

$$accuracy = \frac{TP + TN}{P + N}.$$

2) Precision:

It is a measure of exactness (i.e., what percentage of tuples labeled / predicted as positive are actually positive)

$$precision = \frac{TP}{TP + FP}$$

3) Recall

It is a measure of completeness (what percentage of positive tuples are labeled/predicted as such). It is similar to sensitivity.

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P}.$$

4) F1-Measure

It provides a way to combine both precision and recall into a single measure that captures both properties.

F1-Score = (2 * Precision * Recall) / (Precision + Recall)

5) Specificity = True Negative Rate(TNR) = TN/N

True negative recognition rate .(the proportion of negative tuples that are correctly identified)

6) Sensitivity = True Positive Rate(TPR) = TP/P

True positive recognition rate:(i.e., the proportion of positive tuples that are correctly identified)

ALL THE FORMULAE

Measure	*Formula*
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
F, F_1, F-score, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β, where β is a non-negative real number	$\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

QUESTION

Cancer Classes	Yes	No	Total
Yes	90	210	300
No	140	9560	9700
Total	230	9770	10000

1)

Metrics for Evaluating Classifier

Cancer	Yes	No	Total
Yes	90	210	300
No	140	9560	9700
Total	230	9770	10000

Actual \ Predicted			
	TP	FN	P.
	FP	TN	N
	P'	N'	

$\therefore$ TP = 90 $\quad$ P = TP+FN = 300

FN = 210 $\quad$ N = FP+TN = 9700

FP = 140

TN = 9560

i) Accuracy = $\frac{TP+TN}{P+N} = \frac{90+9560}{10000}$

$= \frac{9650}{10000}$

$= 0.965$

96.5%

ii) Error = $100 - \text{Accuracy}$ (96.5%)

$= 3.5\%$

(2)

2) Sensitivity $= \frac{TP}{P}$

$= \frac{3}{\frac{30}{10}}$

$= 0.3 \quad (30\%)$

3) Specificity or Recall $= \frac{TN}{N}$

$= \frac{9560}{9700}$

$= 0.9856 \quad (98.56\%)$

4) Precision $= \frac{TP}{TP+FP}$

$= \frac{90}{90+140}$

$= \frac{90}{230}$

$= 0.3913 \quad (39.13\%)$

5) F-score $= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

$= \frac{2 \times 39.13 \times 98.56}{39.13 + 98.56} = 56.02\%$

8. Problems based on Decision tree and Naive Bayes algorithm.

Naïve Bayesian Classification Example

Problem: If whether is sunny,then player should play or not?

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Create Frequency table for weather conditions

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Applying Bayes'theorem:

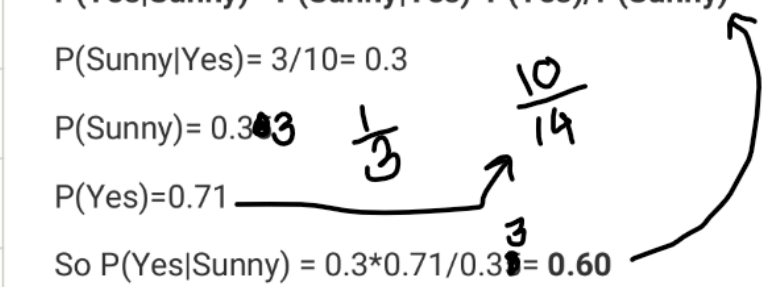
P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)

P(Sunny|Yes)= 3/10= 0.3

P(Sunny)= 0.3

P(Yes)=0.71

So P(Yes|Sunny) = 0.3*0.71/0.3 = **0.60**



P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)

P(Sunny|NO)= 2/4=0.5

P(No)= 0.29

P(Sunny)= 0.3

So P(No|Sunny)= 0.5*0.29/0.3 = **0.41**

So as we can see from the above calculation that **P(Yes|Sunny)>P(No|Sunny)**

Hence on a Sunny day, Player can play the game

Module IV: Clustering

1. What is clustering? State the Applications of clustering algorithms.

- **Cluster**: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- **Cluster analysis**
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes

Clustering Applications

- **Business Intelligence:** To organize a large no of customers into groups.
- **Image Pattern Recognition:** handwritten character recognition systems to improve overall recognition accuracy using multiple models based on multiple subclasses/clusters.
- **Web Search:** To organize the relevant web pages in a concise and easily accessible way.
- **Biology:** It can be used for classification among different species of plants and animals.
- **Information Retrieval:** cluster documents into topics
- **Marketing:** It can be used to characterize & discover customer segments for marketing purposes.

- **Libraries:** It is used in clustering different books on the basis of topics and information.
- **Insurance:** Identifying groups of motor insurance policy holders with a high average claim cost and identifying the frauds.
- As a DM function, clustering can be used
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms
- Outlier detection

2. What are the requirements of any clustering algorithms

- Scalability:
  - Clustering on only a small sample of a large Dataset can lead to biased results.
- Ability to deal with different types of attributes:
  - Recently many applications need clustering on complex data types like graph, sequences, images and documents.
- Discovery of clusters with arbitrary shape
  - Clusters based on ED and MD are spherical in shape.
- Requirements to provide domain knowledge in the form of input parameters:
  - Hard to determine.(high dimensionality DS and no deep understanding)
  - makes quality of the clustering difficult to control.
  - Clustering results are sensitive to those parameters.

- Able to deal with noisy data
  - Clustering algorithms are sensitive to noise and may produce low quality clusters.
- Incremental clustering and insensitivity to input data order.
  - Incorporate incremental updates into existing clustering structures
- Capability of clustering high-dimensional data
  - High dimensional data can be very sparse and highly skewed.
- Incorporation of user-specified constraints
  - Clustering algorithms should be able to find clusters that satisfy the specified constraints.
- Interpretability and usability

3. What are the different approaches to clustering?

Partitioning approach:

- Given a set of *n* objects, a partitioning method constructs *k* partitions of the data, where each partition represents a cluster and *k* <= *n*.
- The basic partitioning methods typically adopt ***exclusive cluster separation***
- Most partitioning methods are distance-based.
- Given k, this approach creates initial partitioning and then uses Iterative relocation technique to improve it.
- The general criterion of a good partitioning is that objects in the same cluster are "close" or related to each other, whereas objects in different clusters are "far apart" or very different .
- Typical methods: k-means, k-medoids, CLARANS

Hierarchical Clustering

- A **hierarchical clustering method** works by partitioning data objects into groups at different levels like a hierarchy or "tree" of clusters.
- Representing data objects in the form of a hierarchy is useful for data summarization and visualization.
- If necessary, hierarchical partitioning can be continued recursively until a desired granularity is reached.
- A hierarchical clustering method can be either *agglomerative* or *divisive*, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or top-down (splitting) fashion.
- An **agglomerative hierarchical clustering method :**
  - The hierarchical decomposition is formed in a bottom-up (merging)way
- A **divisive hierarchical clustering method**
  - The hierarchical decomposition is formed in a top-down (splitting)way
- In either agglomerative or divisive hierarchical clustering, a user can specify the desired number of clusters as a termination condition.
- Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone. So these techniques cannot correct erroneous decisions.

- Density-based approach:
  - Based on connectivity and density functions.
  - The strategy used is to model the dense clusters separated by sparse clusters.
  - The general idea is to continue growing a given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold.
  - Such a method can be used to filter out noise or outliers and discover clusters of arbitrary shape.
    - Major features:
      - Discover clusters of arbitrary shape
      - Handle noise
      - One scan
      - Need density parameters as termination condition

❑ Typical methods: DBSCAN, OPTICS, DenClue

4. Explain the Partitioning approach to clustering (K means and K medoid method)

The *K-Means* Clustering Method

- A centroid-based partitioning technique uses the ***centroid*** of a cluster, *Ci* , to represent that cluster.
- The quality of cluster *Ci* can be measured by the **within cluster variation**, which is the sum of *squared error* between all objects in *Ci* and the centroid ***ci***, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(\boldsymbol{p}, \boldsymbol{c_i})^2,$$

where *E* is the sum of the squared error for all objects in the data set; ***p*** is the point in space representing a given object; and ***ci*** is the centroid of cluster *Ci* (both ***p*** and ***ci*** are multidimensional).

- This objective function tries to make the resulting k clusters as compact and as separate as possible.
- Optimizing the within-cluster variation is computationally challenging.
- To overcome the prohibitive computational cost for the exact solution, greedy approaches are often used in practice. e.g k-means algorithm

- Given *k*, the *k-means* algorithm is implemented as below:
  - First, it randomly selects *k* of the objects in *D*, each of which initially represents a cluster mean or cluster center.
  - For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean.
  - The *k*-means algorithm then iteratively improves the within-cluster variation.
    - For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration.
    - All the objects are then reassigned using the updated means as the new cluster centers.
    - The iterations continue until the assignment is stable, that is, the clusters formed in the current round are the same as those formed in the previous round.

Example and Solution:

- Suppose that the data mining task is to cluster points (with (x, y) representing location) into three clusters, where the points are $A1(2, 10), A2(2, 5), A3(8, 4), B1(5, 8), B2(7, 5), B3(6, 4), C1(1, 2), C2(4, 9)$. The distance function is Euclidean distance. Suppose initially we assign $A1$, $B1$, and $C1$ as the center of each cluster, respectively. Use the *k-means* algorithm to show *only*

  (a) the three cluster centers after the first round of execution.

  Answer:

  After the first round, the three new clusters are: (1) $\{A1\}$, (2) $\{B1, A3, B2, B3, C2\}$, (3) $\{C1, A2\}$, and their centers are (1) (2, 10), (2) (6, 6), (3) (1.5, 3.5).

  (b) the final three clusters.

  Answer:

  The final three clusters are: (1) $\{A1, C2, B1\}$, (2) $\{A3, B2, B3\}$, (3) $\{C1, A2\}$.

- **K-Medoids:** Instead of taking the mean value of the objects in a cluster as a reference point, we can pick actual objects to represent the clusters, using one representative object per cluster(Medoid).

- Each remaining object is assigned to the cluster of which the representative object is the most similar.

- The partitioning method is then performed based on the principle of **minimizing the sum of the dissimilarities** between each object ***p*** and its corresponding representative object.
- An **absolute-error criterion** is defined as $E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, o_i)$,

where *E* is the sum of the absolute error for all objects ***p*** in the data set, and ***oi*** is the representative object of *Ci* .

This is the basis for the ***k*-medoids method**, which groups *n* objects into *k* clusters by minimizing the absolute error

- A medoid of a finite dataset is a data point from this set, whose average dissimilarity to all the data points is minimal i.e. it is the most centrally located point in the set.

Algorithm: *k*-**medoids.** PAM, a *k*-medoids algorithm for partitioning based on medoid or central objects.

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 in *D* as the initial representative objects or seeds;
(2) **repeat**
(3) assign each remaining object to the cluster with the nearest representative object;
(4) randomly select a nonrepresentative object, o_{random};
(5) compute the total cost, *S*, of swapping representative object, o_j, with o_{random};
(6) **if** $S < 0$ **then** swap o_j with o_{random} to form the new set of *k* representative objects;
(7) **until** no change;

PAM, a *k*-medoids partitioning algorithm.

Example: K-medoids,PAM

Given D={1,2,6,7,8,10,15,17,20} Form 3 clusters using k-medoid algorithm. Assume initial medoids as 6,7 8 respectively for K1,K2,and K3.

What Is the Problem with PAM?

- Adv: PAM is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean

- Disadv:PAM works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration : very costly computation for large n and k where n is # of data objects and ,k is # of clusters

- To deal with large data sets,a sampling based method called CLARA can be used.

5. **Explain the Hierarchical Approach to clustering (Agglomerative and Divisive clustering)**

Hierarchical Clustering

- A **hierarchical clustering method** works by partitioning data objects into groups at different levels like a hierarchy or "tree" of clusters.

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

- If necessary, hierarchical partitioning can be continued recursively until a desired granularity is reached.

Hierarchical Clustering
(Agglomerative versus Divisive Hierarchical Clustering)

- A hierarchical clustering method can be either *agglomerative* or *divisive*, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or top-down (splitting) fashion.

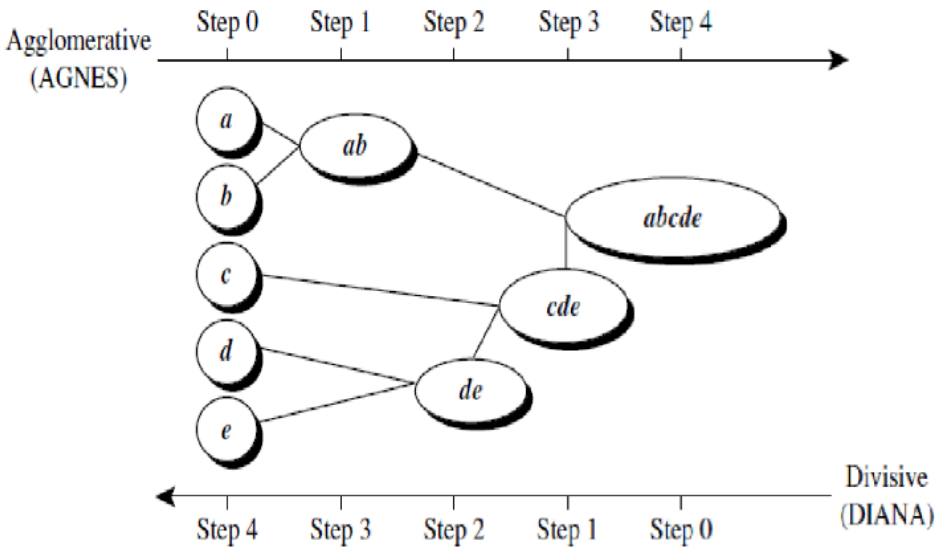
- An **agglomerative hierarchical clustering method :**
  - The hierarchical decomposition is formed in a bottom-up (merging)way

- A **divisive hierarchical clustering method**
  - The hierarchical decomposition is formed in a top-down (splitting)way

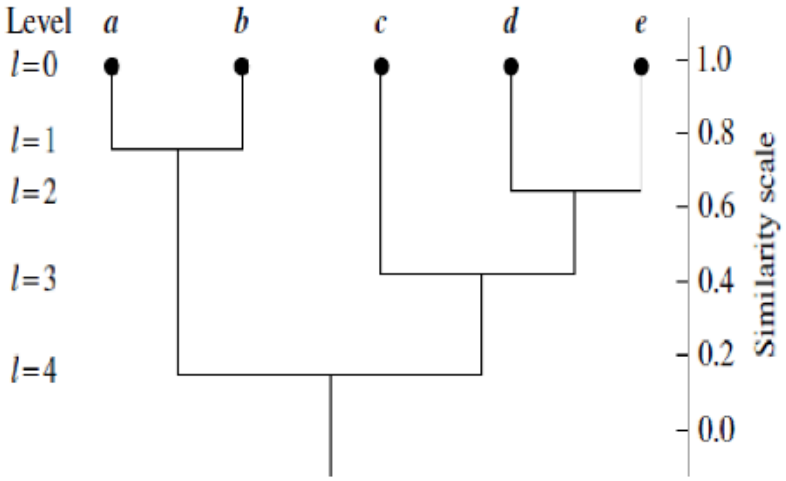
- In either agglomerative or divisive hierarchical clustering, a user can specify the desired number of clusters as a termination condition.

- Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone. So these techniques cannot correct erroneous decisions.

Example : Application of AGNES(AGglomerative NESting and DIANA(DIvisible ANAlysis)



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



Dendrogram representation for hierarchical clustering of data objects $\{a, b, c, d, e\}$.

Divisive Method Challenges

- How to partition a large cluster into smaller ones is a challenge.
  - $(2^{n-1}-1)$ ways to partition n objects into 2 exclusive clusters.
  - For large n, it is computationally inhibitive to examine all possibilities .
- Also a Divisive method typically uses heuristics in partitioning which can lead to inaccurate results.
- For the sake of efficiency divisive methods typically do not backtrack on the partitioning decisions that have been made.
- Due to these challenges, Agglomerative clustering is mostly used than divisive clustering.

The **Agglomerative hierarchical clustering** algorithm:

- It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together.
- It does this until all the clusters are merged into a single cluster that contains all the datasets.
- This hierarchy of clusters is represented in the form of the dendrogram.

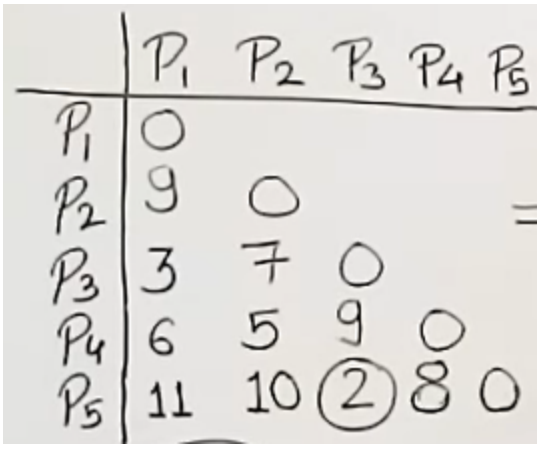
1. Bottom up (Hierarchical Agglomerative Clustering, HAC):

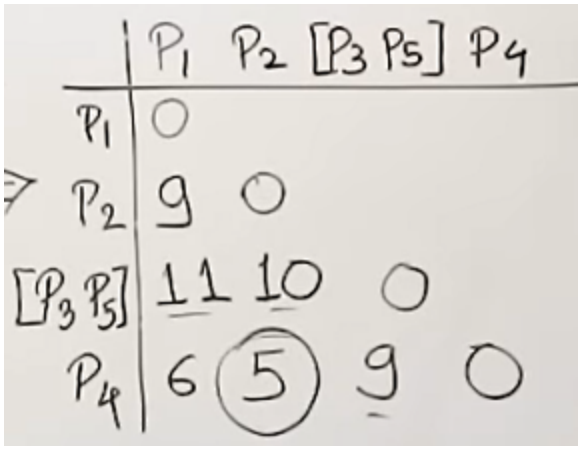
1. Treat each document as a single cluster at the beginning of the algorithm.
2. Merged(agglomerate) two items at a time into a new cluster. *How* the pairs merge involves calculating a dissimilarity between each merged pair and the other samples. There are *many* ways to do this. Popular options:
   1. **Complete linkage**: similarity of the farthest pair. One drawback is that outliers can cause merging of close groups later than is optimal.
   2. **Single-linkage**: similarity of the closest pair. This can cause premature merging of groups with close pairs, even if those groups are quite dissimilar overall.
   3. **Group average**: similarity between groups.
   4. **Centroid similarity: each iteration merges the clusters with the most similar central point.**
3. The pairing process continues until all items merge into a single cluster.

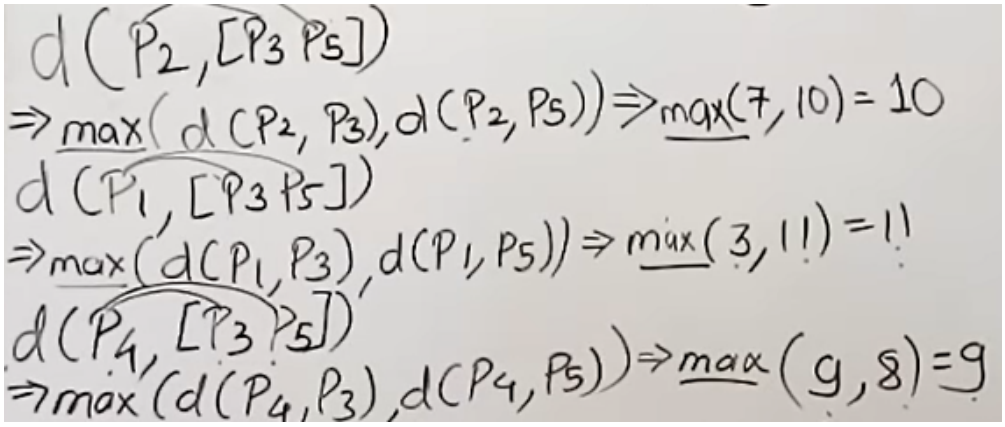
Distance Measures(*linkage measures*) in Algorithmic Methods

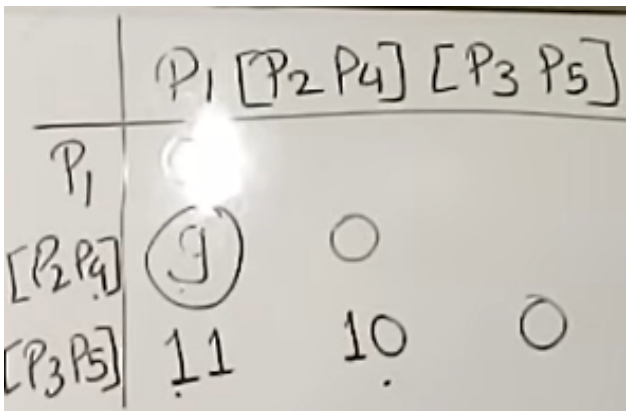
- When an algorithm uses the *minimum distance*, $d_{min}(Ci, Cj)$, to measure the distance between clusters, it is sometimes called a **nearest-neighbor clustering algorithm**.
  - Here if the clustering process is terminated when the minimum distance between nearest clusters exceeds a user-defined threshold, it is called a **single-linkage algorithm**.

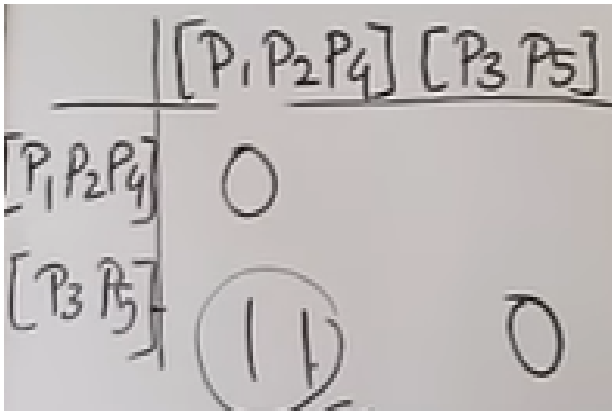
- When an algorithm uses the *maximum distance*, $d_{max}(Ci, Cj)$, to measure the distance between clusters, it is sometimes called a **farthest-neighbor clustering algorithm**.
  - Here,If the clustering process is terminated when the maximum distance between nearest clusters exceeds a user-defined threshold, it is called a **complete-linkage algorithm**

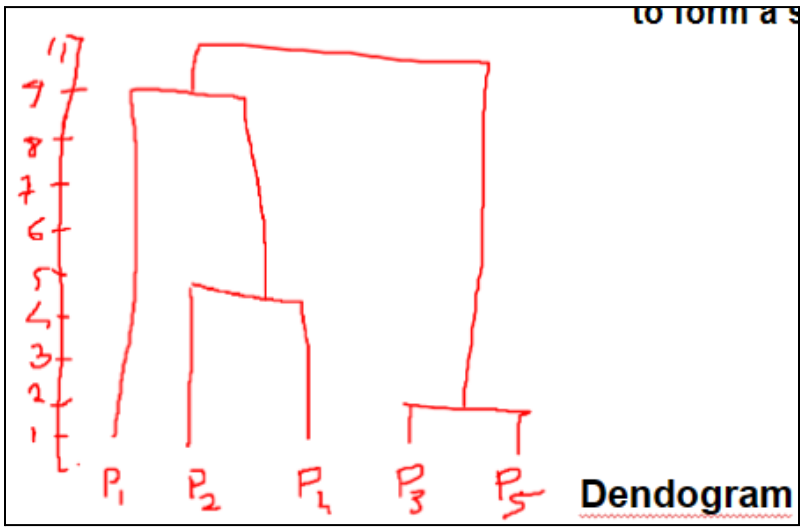












6. **What is single linkage and complete linkage agglomerative clustering?** **(refer above)**

The **Agglomerative hierarchical clustering** algorithm:

- To group the datasets into clusters, it follows the bottom-up approach.
- It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together.

- It does this until all the clusters are merged into a single cluster that contains all the datasets.
- This hierarchy of clusters is represented in the form of the dendrogram.

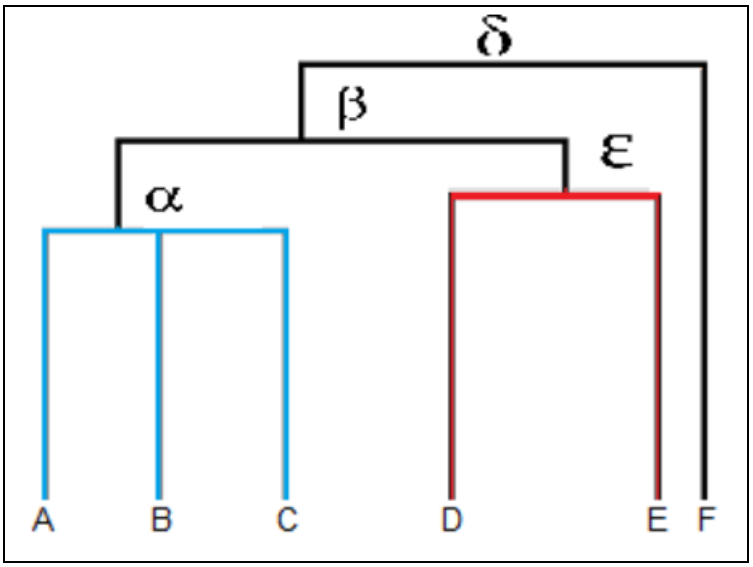
Distance Measures(*linkage measures*) in Algorithmic Methods

- When an algorithm uses the *minimum distance*, $d_{min}(Ci, Cj)$, to measure the distance between clusters, it is sometimes called a **nearest-neighbor clustering algorithm**.
  - Here if the clustering process is terminated when the minimum distance between nearest clusters exceeds a user-defined threshold, it is called a **single-linkage algorithm**.

- When an algorithm uses the *maximum distance*, $d_{max}(Ci, Cj)$, to measure the distance between clusters, it is sometimes called a **farthest-neighbor clustering algorithm**.
  - Here,If the clustering process is terminated when the maximum distance between nearest clusters exceeds a user-defined threshold, it is called a **complete-linkage algorithm**

7. What is a dendrogram?

- A dendrogram is a diagram that shows the hierarchical relationship between objects.
- A dendrogram is a type of tree diagram showing hierarchical clustering relationships between similar sets of data.
- Hierarchical clustering is where you build a cluster tree (a dendrogram) to represent data, where each group (or “node”) links to two or more successor groups.
- It is most commonly created as an output from hierarchical clustering.
- The main use of a dendrogram is to work out the best way to allocate objects to clusters.
  - The basic graph comprises of the parts:
    - The *clade* is the branch. Usually labeled with Greek letters from left to right (e.g. α β, δ…)
    - Each clade has one or more *leaves*. The leaves in the below image are:
      - Single (simplicifolius): F
      - Double (bifolius): D E
      - Triple (trifolious): A B C



- The clades are arranged according to how similar (or dissimilar) they are.
- Clades that are close to the same height are similar to each other; clades with different heights are dissimilar — the greater the difference in height, the more dissimilarity.
    - Leaves A, B, and C are more similar to each other than they are to leaves D, E, or F.
    - Leaves D and E are more similar to each other than they are to leaves A, B, C, or F.
    - Leaf F is substantially different from all of the other leaves.

8. What is DBSCAN? Two parameters used in DBSCAN algo.

- Density-based approach:
    - Based on connectivity and density functions.
    - The strategy used is to model the dense clusters separated by sparse clusters.
    - The general idea is to continue growing a given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold.
    - Such a method can be used to filter out noise or outliers and discover clusters of arbitrary shape.
        - Major features:
            - Discover clusters of arbitrary shape
            - Handle noise
            - One scan
            - Need density parameters as termination condition

- Typical methods: DBSCAN, OPTICS, DenClue

DBSCAN(Density Based Spatial Clustering of Application with Noise)

- Density-Based Clustering Algorithm Based on Connected Regions with High Density.

- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

- **Why DBSCAN?**
  - Real life data may contain irregularities, like:
    1. Clusters can be of arbitrary shape.
    2. Data may contain noise.

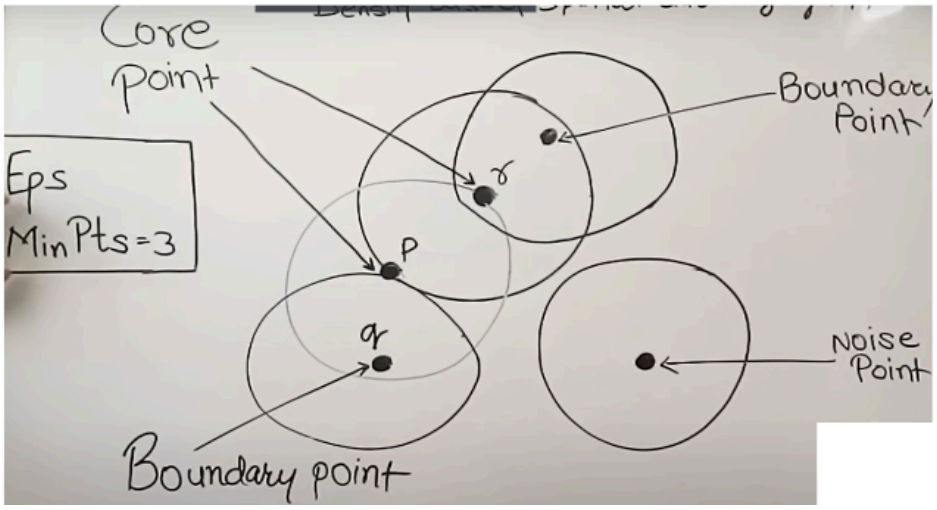
- **DBSCAN algorithm requires two parameters:**

1. **eps** : It defines the neighborhood around a data point
   - if the distance between two points is <= 'eps' then they are considered as neighbors.
   - If the eps value is chosen too small then large part of the data will be considered as outliers.
   - If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters.
   - One way to find the eps value is based on the ***k-distance graph***.

2. **MinPts**: Minimum number of neighbors (data points) within eps radius.
   - Larger the dataset, the larger value of MinPts must be chosen.
   - As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1.
   - The minimum value of MinPts must be chosen as at least 3.

- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) finds *core objects*, that is, objects that have dense neighborhoods. It connects core objects and their neighborhoods to form dense regions as clusters.

- **In this algorithm, we have 3 types of data points.**
  - **Core Point**: A point is a core point if it has more than MinPts points within eps.
  - **Border Point**: A point which has fewer than MinPts within eps(i.e it is not a core point) but it is in the neighborhood of a core point.
  - **Noise or outlier**: A point which is not a core point or border point.



MinPts=3

DBSCAN: The Algorithm

- Arbitrarily select a point *p*
- Retrieve all points that are density-reachable from *p* w.r.t. *Eps* and *MinPts*.
- If *p* is a core point, a cluster is formed.
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

Density-Based Clustering vs Partitioning-Based and Hierarchical Clustering

Density-based clustering is a type of clustering method that is based on the notion of density. It groups together data points that are close to each other in the data space and have a sufficient number of neighbors. The most common example of density-based clustering is DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

Partitioning-based clustering, such as K-means, divides the data into non-overlapping subsets (clusters) without any cluster-internal structure. Hierarchical clustering, on the other hand, creates a tree of clusters, allowing you to visualize the data at different levels of granularity.

Conditions Favoring Density-Based Clustering

1. **Noise and Outliers**: Density-based clustering is more robust to noise and outliers compared to partitioning-based and hierarchical clustering. In DBSCAN, for example, outliers are classified as noise.
2. **Arbitrary Shapes**: Density-based clustering can find clusters of arbitrary shapes, unlike partitioning-based clustering which assumes spherical clusters.
3. **No Need to Specify Number of Clusters**: Unlike partitioning-based clustering (e.g., K-means), density-based clustering does not require the user to specify the number of clusters in advance.

Application Examples

1. **Image Processing**: Density-based clustering can be used in image processing to identify objects of interest in an image, which can have arbitrary shapes.
2. **Anomaly Detection**: In cybersecurity, density-based clustering can be used to detect unusual patterns or anomalies, which are treated as noise.
3. **Spatial Data Analysis**: In geography and related fields, density-based clustering can be used to identify areas of high density, such as urban areas in a population density map.

Module V: Frequent Pattern Mining

1. **What do you mean by frequent itemset, frequent subsequence and frequent substructure? State one example for each.**
   Frequent itemset, frequent subsequence, and frequent substructure are concepts commonly used in data mining and pattern recognition to identify recurring patterns in data.

   a) **Frequent Itemset**: In data mining, a frequent itemset refers to a set of items that frequently appear together in a dataset. The frequency of an itemset is measured by its support, which is the proportion of transactions in which the itemset appears.

Frequent Itemset: A **frequent itemset** is an itemset whose support is greater than some user-specified minimum support

Eg - Suppose support of itemset = 3

Milk	Milk
Milk	Butter
Bread	Bread
Biscuit	Egg

In this Case Milk is bought 3 times which is greater than or equal to support value
Hence here Milk is an Frequent Itemset

b) **Frequent Subsequence**: A frequent subsequence is a sequence of items that occurs frequently in a sequence database. Unlike itemsets, subsequences maintain the order of items.
For example, It is a sequence of customer transactions, a frequent subsequence could be
{Bread —> Butter —> Milk —> Biscuit}: Almost 25%
{eggs—> cheese —> bread} : Almost 20%
if many customers follow this sequence of actions.

c) **Frequent Substructure**: In data mining, a frequent substructure refers to a recurring itemsets that appears frequently in a dataset. A substructure can refer to different structural forms, such as subgraphs, subtrees, or

sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (frequent) structured pattern. For example,In a frequent itemset, a frequent substructure could
(Bread - Butter) : Almost 15 %
(Milk - Egg) : Almost 25%

Let's consider a dataset of chemical compounds represented as graphs, where atoms are nodes and chemical bonds are edges. We want to find frequent substructures in these compounds.

Suppose we have the following simplified dataset:

1. Compound A: CH3-CH2-CH2-OH
2. Compound B: CH3-CH2-CH2-Cl
3. Compound C: CH3-CH2-NH2
4. Compound D: CH3-CH2-CH2-CH3

In this dataset, the substructure "CH3-CH2-CH2-" appears frequently across multiple compounds. It's a common backbone in organic molecules. So, "CH3-CH2-CH2-" is a frequent substructure in this dataset.

2. **What is Market Basket Analysis ? What are the applications of market basket analysis?**
   Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings and products that are likely to be purchased together.

   One example is the Shopping Basket Analysis tool in Microsoft Excel, which analyzes transaction data contained in a spreadsheet and performs market basket analysis. A transaction ID must relate to the items to be analyzed. The Shopping Basket Analysis tool then creates two worksheets:

   1) The Shopping Basket Item Groups worksheet, which lists items that are frequently purchased together,
   2) And the Shopping Basket Rules worksheet shows how items are related (For example, purchasers of Computer are likely to buy AntiVirus).

Market Basket Analysis is modelled on Association rule mining, i.e., the IF {}, THEN {} construct. For example, IF a customer buys bread, THEN he is likely to buy butter as well.

Association rules are usually represented as: {Bread} -> {Butter}

Applications:

- **Retail**: The most well-known MBA case study is Amazon.com. Whenever you view a product on Amazon, the product page automatically recommends, "Items bought together frequently."
- Another example, you are almost always likely to find shampoo and conditioner placed very close to each other at the grocery store.
- **Telecom**: With the ever-increasing competition in the telecom sector, companies are paying close attention to customers' services. For example, Telecom has now started to bundle TV and Internet packages apart from other discounted online services to reduce churn.
- **IBFS**: Tracing credit card history is a hugely advantageous MBA opportunity for IBFS organizations. For example, Citibank frequently employs sales personnel at large malls to lure potential customers with attractive discounts on the go. They also associate with apps like Swiggy and Zomato to show customers many offers they can avail of via purchasing through credit cards.
- **Medicine**: Basket analysis is used to determine comorbid conditions (two or more medical conditions) and symptom analysis in the medical field. It can also help identify which genes or traits are hereditary and which are associated with local environmental effects.

Advantages

- **Increasing market share**: Once a company hits peak growth, it becomes challenging to determine new ways of increasing market share. Market Basket Analysis can be used to put together demographic and gentrification data to determine the location of new stores or geo-targeted ads.
- **Behaviour analysis**: Understanding customer behaviour patterns is a primal stone in the foundations of marketing. MBA can be used anywhere from a simple catalogue design to UI/UX.
- **Optimization of in-store operations**: MBA is not only helpful in determining what goes on the shelves but also behind the store. Geographical patterns play a key role in determining the popularity or strength of certain products, and therefore, MBA has been increasingly used to optimize inventory for each store or warehouse.

- **Campaigns and promotions**: Not only is MBA used to determine which products go together but also about which products form keystones in their product line.
- **Recommendations**: OTT platforms like Netflix and Amazon Prime benefit from MBA by understanding what kind of movies people tend to watch frequently.

3. Define the terms support,support count,confidence, Frequent itemset, closed frequent itemset,maximal frequent itemset with an example

Frequent Itemset: A **frequent itemset** is an itemset whose support is greater than some user-specified minimum support

Closed Frequent Itemset: An itemset is closed if none of its immediate supersets has the same support as that of the itemset.

Maximal Frequent Itemset: An itemset is maximal frequent if none of its immediate supersets is frequent.

Itemset: A collection of one or more items.
e.g. {Milk, Bread, Butter}

1	*milk, bread*
2	*bread, butter*
3	*beer*
4	*milk, bread, butter*
5	*bread*
6	*milk, bread, butter*

Support Count(σ): It represents frequency of occurrence of an itemset. **Absolute support**
e.g. σ{Milk, Bread}=3

Support (s): Fraction of transactions that contain an itemset. **Relative support**
e.g s{Milk, Bread}=3/6

Frequent Itemset: An itemset whose support is greater than or equal to a minimum support threshold.
If minimum support threshold=3 then {Milk, Bread} are frequent
If minimum support threshold=4 then {Milk, Bread} are not frequent

Association Rule
An implication expression of the form X→Y where X and Y are itemsets
E.g. {Milk, Bread}→Butter
Rule Evaluation Metrics
Support(S): Fraction of transaction that contain both X and Y.
Confidence(c): Measures how often items in Y appear in transactions that contain X.

A. **Support:** Support is the proportion of transactions in a dataset that contain a particular itemset. It indicates how frequently an itemset appears in the dataset.
   Example: If there are 100 transactions and 50 of them contain the itemset {milk, bread}, then the support of {milk, bread} is 50%.
B. **Support Count:** Support count is the number of transactions in a dataset that contain a particular itemset.
   Example: If the support count of {milk, bread} is 50, it means that this itemset appears in 50 transactions out of the total.
C. **Confidence:** Confidence is the probability that a transaction containing one itemset also contains another itemset. It measures the strength of the association between two itemsets.
   Example: If the confidence of {milk} -> {bread} is 70%, it means that 70% of transactions containing {milk} also contain {bread}.
D. **Frequent Itemset:** A frequent itemset is an itemset whose support is greater than or equal to a specified minimum support threshold.
   Example: If the minimum support threshold is 30%, then {milk, bread} is a frequent itemset because its support is 50%.
E. **Closed Frequent Itemset:** A closed frequent itemset is a frequent itemset for which there is no superset with the same support.
   Example: If {milk, bread} is a closed frequent itemset and {milk, bread, butter} is also frequent but has the same support as {milk, bread}, then {milk, bread} is a closed frequent itemset.
F. **Maximal Frequent Itemset:** A maximal frequent itemset is a frequent itemset that is not a subset of any other frequent itemset.
   Example: If {milk, bread} is a maximal frequent itemset, it means that there is no other frequent itemset that contains {milk, bread} as a subset.

Milk,Bread
Milk,Bread
Milk
Milk
Butter

Lets suppose Support value = 3

- Support of Milk = 80%
- Support count of milk = 4

- Confidence value of {Milk,Bread} = 50%
- Support value of milk is 4 which is more than given value 3 hence it is a frequent itemset
- Milk is Closed Frequent Itemset because Its just superset is {Milk,Bread} having support count 2 which is not equal to support count of Milk i.e. 4
- Milk is Maximal Frequent itemset because its immediate superset is {Milk,Bread} having support count 2 which is less than given support value that is 3 hence is not a frequent item set

4. Explain an Apriori Algorithm for frequent itemset mining.

Apriori algorithm refers to an algorithm that is used in mining frequent products sets and relevant association rules. Generally, the apriori algorithm operates on a database containing a huge number of transactions.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Step-1: K=1
(I) Create a table containing support count of each item present in dataset – Called C1(candidate set)

Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

C1

(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.

Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

L1

Step-2: K=2

- Generate candidate set C2 using L1 (this is called join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common.
- Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
- Now find support count of these itemsets by searching in dataset.

C2

Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

(II) compare candidate (C2) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.

Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I2,I5	2

L2

- **Step-3:**
  - Generate candidate set C3 using L2 (join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common. So here, for L2, first element should match.
    So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5}{I2, I3, I4}{I2, I4, I5}{I2, I3, I5}
  - Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
  - find support count of these remaining itemset by searching in dataset.

Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

(II) Compare candidate (C3) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.

Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

- **Step-4:**
  - Generate candidate set C4 using L3 (join step). Condition of joining L_{k-1} and L_{k-1} (K=4) is that, they should have (K-2)

elements in common. So here, for L3, first 2 elements (items) should match.

- Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4
- We stop here because no frequent itemsets are found further

Thus, we have discovered all the frequent item-sets. Now generation of strong association rule comes into picture. For that we need to calculate confidence of each rule.

Confidence –

A confidence of 60% means that 60% of the customers, who purchased milk and bread also bought butter.

Confidence(A->B)=Support_count(A ∪ B)/Support_count(A)

So here, by taking an example of any frequent itemset, we will show the rule generation.

Itemset {I1, I2, I3} //from L3

SO rules can be

[I1^I2]=>[I3] //confidence = sup(I1^I2^I3)/sup(I1^I2) = 2/4*100=50%

[I1^I3]=>[I2] //confidence = sup(I1^I2^I3)/sup(I1^I3) = 2/4*100=50%

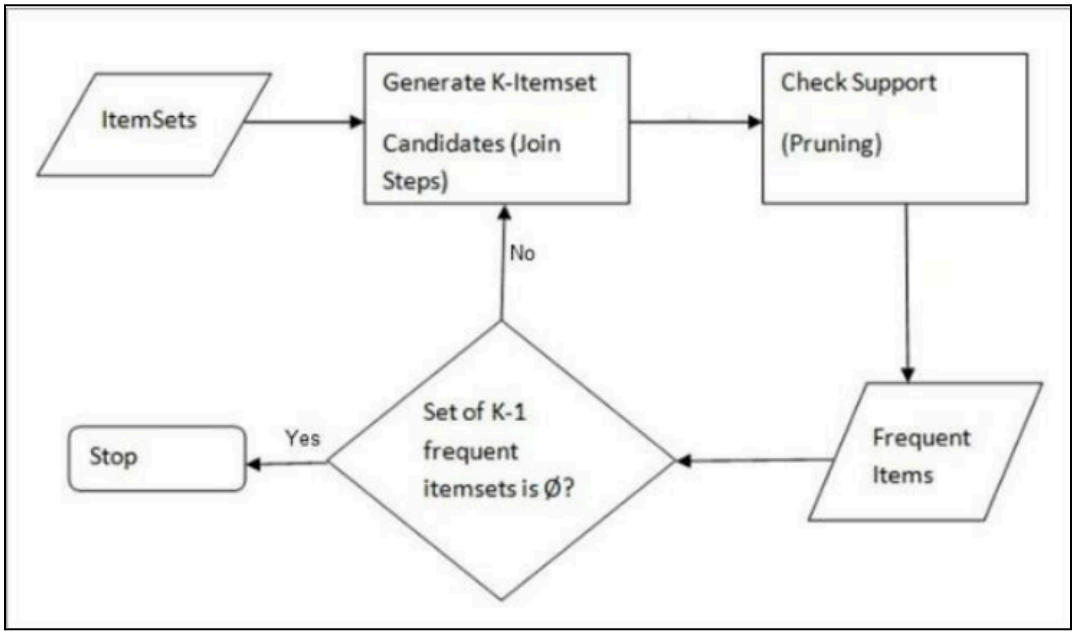
[I2^I3]=>[I1] //confidence = sup(I1^I2^I3)/sup(I2^I3) = 2/4*100=50%

[I1]=>[I2^I3] //confidence = sup(I1^I2^I3)/sup(I1) = 2/6*100=33%

[I2]=>[I1^I3] //confidence = sup(I1^I2^I3)/sup(I2) = 2/7*100=28%

[I3]=>[I1^I2] //confidence = sup(I1^I2^I3)/sup(I3) = 2/6*100=33%

So if minimum confidence is 50%, then first 3 rules can be considered as strong association rules.



5. Explain the Join and Prune step of Apriori algorithm with an example.

1. **Generating Candidate Itemsets**: Initially, you start with individual items as your candidate itemsets. Then, you combine them to generate larger itemsets.
2. **Joining**: In the joining step, you take the frequent (k-1)-itemsets (sets of items that occur frequently in the dataset) and join them together to form candidate k-itemsets.
3. **Pruning**: After joining, you prune the candidate itemsets to eliminate those that cannot be frequent. This is typically done by checking if all the (k-1)-subsets of a candidate k-itemset are frequent. If any subset is not frequent, then the candidate k-itemset is not frequent and can be discarded.
4. **Repeat**: The process continues iteratively to generate candidate itemsets of larger sizes until no new frequent itemsets can be generated.

So, in short, the "join" step in the Apriori algorithm involves combining frequent itemsets of size k-1 to generate candidate itemsets of size k.

Prune Step:

- In the prune step, the algorithm scans the count of each item in the database.
- If a candidate itemset does not meet the minimum support threshold, it is considered infrequent and removed from further consideration.
- This step helps reduce the size of the candidate itemsets and focuses computation on potentially meaningful itemsets.

Continuing with our example, let's assume the minimum support threshold is set to 2 (meaning an itemset must appear in at least 2 transactions to be considered frequent).
After generating potential 2-itemsets in the join step, we count the occurrences of each candidate itemset in the database:

- {bread, milk}: 2
- {bread, eggs}: 2
- {milk, eggs}: 1 (not meeting minimum support, so it's pruned)
- ...

In this case, the itemset {milk, eggs} does not meet the minimum support threshold and is pruned from further consideration.
The join and prune steps are repeated iteratively to generate larger itemsets until no new frequent itemsets can be found or until the desired itemset size is reached. This process forms the basis of the Apriori algorithm for frequent itemset mining.

6. Advantages and disadvantages of Apriori Algorithm

Advantages:

1. **Calculation of Large Itemsets:** The primary advantage of the Apriori algorithm is its ability to efficiently calculate large itemsets. It uses a "bottom-up" approach,

starting from itemsets of size 1 and iteratively finding larger itemsets by combining smaller ones. This allows it to handle datasets with a large number of transactions and items effectively.

2. **Simple to Understand and Apply:** Another significant advantage of Apriori is its simplicity. The algorithm's concept is straightforward and intuitive, making it easy to understand and implement, even for those new to data mining and machine learning. This simplicity contributes to its popularity and widespread use in both academia and industry.

Disadvantages:

- **Computational Expense:** One of the main drawbacks of the Apriori algorithm is its computational expense, particularly in terms of calculating support. Support refers to the frequency of occurrence of an itemset in the dataset. Since Apriori needs to scan the entire database to calculate support for each itemset, it can be time-consuming and resource-intensive for large datasets.
- **Large Number of Candidate Rules:** In some cases, the Apriori algorithm may generate a large number of candidate rules, especially when dealing with datasets containing a vast number of items or transactions. These candidate rules need to be evaluated to identify meaningful associations, which can significantly increase the computational complexity of the algorithm. As a result, processing such datasets can become computationally expensive and may require substantial computational resources.

7. What are the different methods to improve efficiency of Apriori Algorithm?

Here are some of the methods how to improve efficiency of apriori algorithm -
Hash-Based Technique:

a) Hash Based Techniques

- Hash table is used as data structure.
- First iteration is required to count support of each itemset.
- From second iteration, efforts are made to enhance execution of Apriori by utilizing hash table concept.
- Hash table minimizes the number of itemset generated in second iteration.
- In second iteration i.e. 2-itemset generation, for every combination of two item, we map them into the diverse bucket of hash table structure and increment the bucket count.
- If count of bucket is less than min. sup. count, we remove them from candidate sets.

This method uses a hash-based structure called a hash table for generating the k-itemsets and their corresponding count. It uses a hash function for generating the table.

Bucket address	0	1	2	3	4	5	6
Bucket Count	2	2	4	2	2	4	4
Bucket Contents	{I1-I4}-1 {I3-I5}-1	{I1-I5}-2	{I2-I3}-4	{I2-I4}-2	{I2-I5}-2	{I1-I2}-4	{I1-I3}-4
L2	No	No	Yes	No	No	Yes	Yes

Advantages:

- Reduce the number of scans
- Remove the large candidates that cause high Input/output cost

Transaction Reduction:

Transaction that does not contain any frequent k-itemsets cannot contain any frequent (k+1)-itemsets. Therefore, such a transaction can be marked or removed from further consideration

This method reduces the number of transactions scanned in iterations. The transactions which do not contain frequent items are marked or removed.

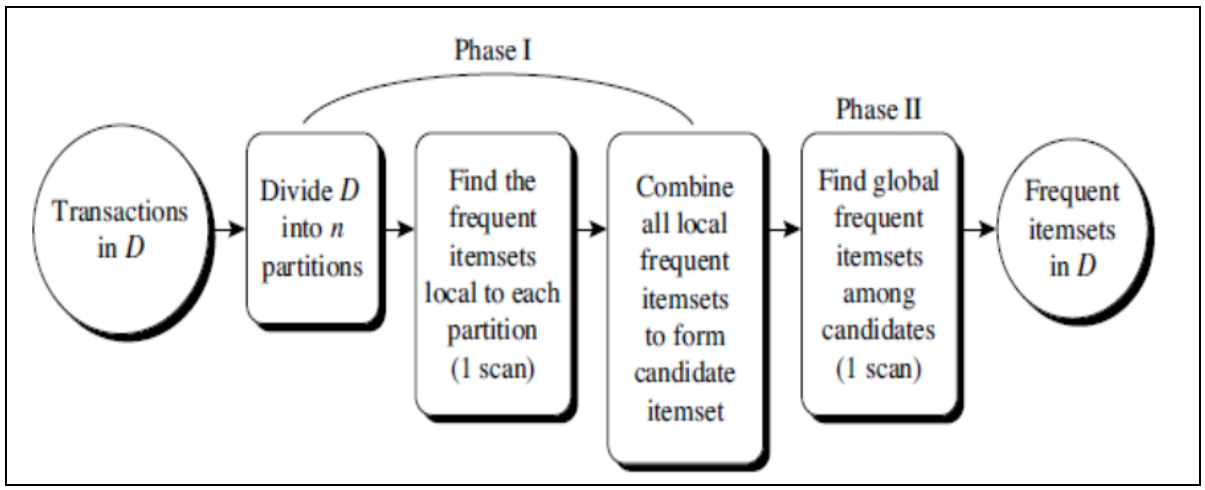
Example: Min. support = 2

	I1	I2	I2	I4	I5
T1	1	1	0	0	1
T2	0	1	1	1	0
T3	0	0	1	1	0
T4	1	1	1	1	0

	I1,I2	I1,I3	I1,I4	I2,I3	I2,I4	I3,I4
T1	1	0	0	0	0	0
T2	0	0	0	1	1	1
T3	0	0	0	0	0	1
T4	1	1	1	1	1	1

Partitioning:

Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB (2 DB Scan)



This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database.

Trans.	Itemset	First Scan	Second Scan	Shortlisted
		Support =20% Min. sup=1	Support =20% Min. sup=2	
T1	I1, I5	I1-1, I2-1, I4-1, I5-1	I1-1, I2-3	I2-3, I2-2
T2	I2, I4	{I1,I5}-1 {I2, I4}-1	I3-2. I4-3	I4-3 ,I5-3
T3	I4, I5	I2-1, I3-1, I4-1, I5-1	I5-3, {I1, I5}-1	{I2, I4}-2
T4	I2, I3	{I4,I5}-1,{I2, I3}-1	{I2, I4}-2, {I4, I5}-1	{I2, I3}-2
T5	I5	I2-1, I3-1,I4-1, I5-1	{I2, I3}-2, {I3,I4}-1	
T6	I2, I3, I4	{I2, I3}-1, {I2,I4}-1	{I2, I3, I4}-1	
		{I3, I4}-1 {I2, I3, I4}-1		

Sampling:

- The fundamental idea of the sampling approach is to select a random sample S of the given data D, and then search for frequent itemsets in S rather than D.

- It may be possible to lose the global frequent itemset. This can be reduced by lowering the minimum support.

- The sample size of S is such that the search for frequent itemsets in S can be completed in main memory, and therefore only one scan of the transactions in S is needed overall.

- Because we are searching for frequent itemsets in S rather than in D, it is possible that we will miss some of the global frequent itemsets

This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min_sup.

Dynamic Itemset Counting:

It is an algorithm which reduces the number of passes made over the data while keeping the number of itemsets which are counted in any pass relatively low.

This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database.

This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database.

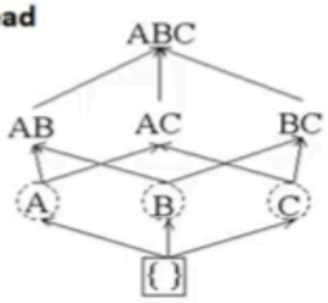
Solid box: ☐ Confirmed frequent itemset - an itemset we have finished counting and exceeds the support threshold *minsupp*

Solid circle: ◯ Confirmed infrequent itemset - we have finished counting and it is below *minsupp*

Dashed box: ⬚ suspected frequent itemset - an itemset we are still counting that exceeds *minsupp*

Dashed circle: ◌ suspected infrequent itemset - an itemset we are still counting that is below *minsupp*

Itemset lattice before any transactions are read



Empty itemset is marked with a **solid box.**
All 1-itemsets are marked with **dashed circles.**

Counters: A = 0, B = 0, C = 0

8. State applications of Apriori Algorithm

Apriori Algorithm has picked up a pace in recent years and is used in different industries for data mining and handling.

Some fields where Apriori is used:

1. Medical

Hospitals are generally trashed with data every day and need to retrieve a lot of past data for existing patients. Apriori algorithm help hospitals to manage the database of patients without jinxing it with other patients.

2. Education

The educational institute can use the Apriori algorithm to store and monitor students' data like age, gender, traits, characteristics, parent's details, etc.

3. Forestry

On the same line as the education and medical industry, forestry can also use the Apriori algorithm to store, analyze and manage details of every flora and fauna of the given territory.

4. New Tech Firms

Tech firms use the Apriori algorithm to maintain the record of various items of products that are purchased by various customers for recommender systems.

5. Mobile Commerce

Big data can help mobile e-commerce companies to deliver an easy, convenient and personalized shopping experience. With the Apriori algorithm, the real-time product recommendation accuracy increases, which creates an excellent customer experience and increases sales for the company.

6. **Offices**

One of the most efficient uses of this computer science technique is in the offices where they have to record a large number of day to day transactions related to sale and purchase of various good and services, like recording the transactions of creditors, sales and purchases so there is need of analysis of all such transactions so that there should not be any kind of confusion.

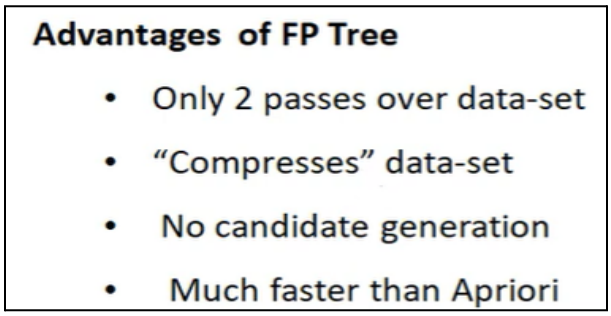
9. Explain Frequent pattern algorithm. State advantages of it over Apriori algorithm

- It is pattern growth approach for mining frequent itemsets.
- It's concept is basically based on divide-and-conquer strategy.
- First it compresses the frequent item database into frequent pattern tree(FP tree) which captures itemset association information.
- Set of conditional databases is obtained from FP tree. Conditional pattern base for each node represent various pattern fragment and we extract frequent pattern fragment or pattern fragment who satisfies minimum confidence criteria.
- Therefore, this approach substantially reduce the size of the data sets to be searched and also given out various frequent pattern segments

Algo:

Step 1: Calculate minimum support count.
Step 2: Find frequency of occurrence.
Step 3: Filter out itemsets which does not support Min. support count. Prioritize the items.
Step 4: Order the items according to the priority.
Step 5: Build FP-Tree.
Step 6: Compute Conditional pattern base for each item.

Step 7: Build Conditional FP Tree for each item.
Step 8: Generate Frequent Patterns.
Step 9: Write the frequent patterns.

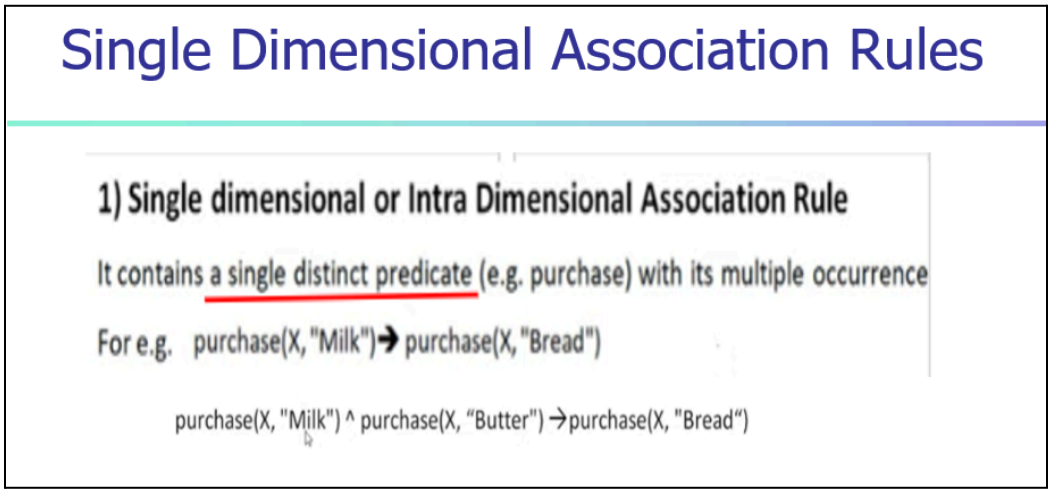


Detailed advantages:

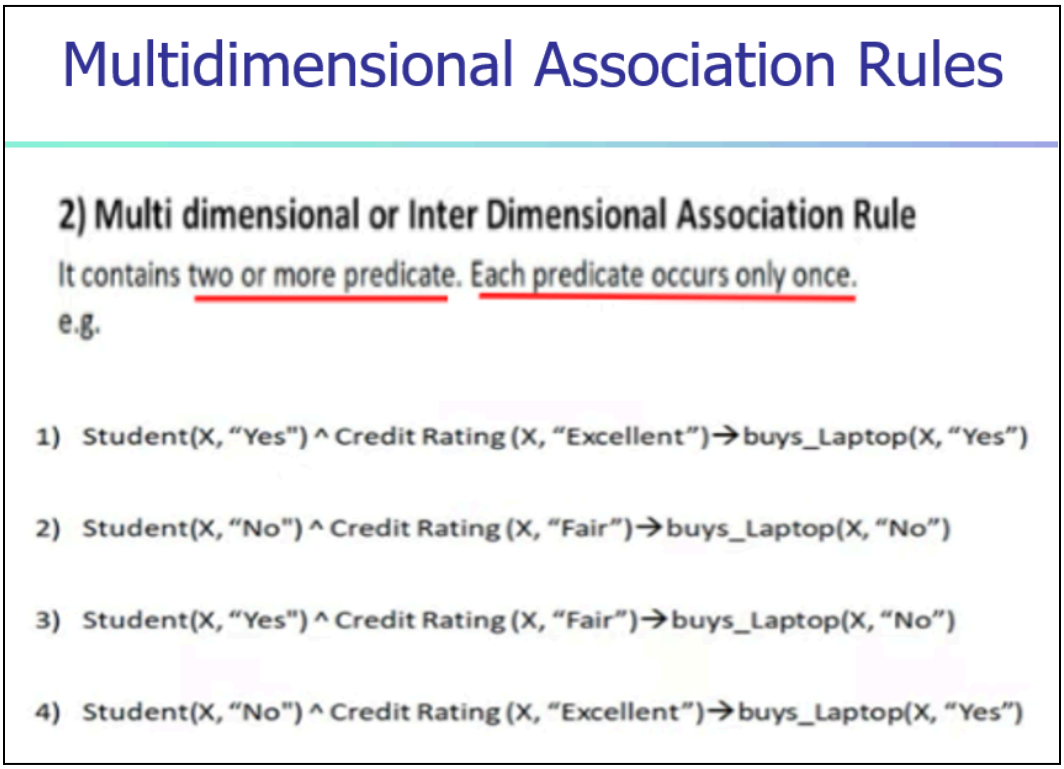
1. **Efficiency:**
    - FP-growth typically outperforms Apriori in terms of runtime, especially on large datasets. This is because it only requires two passes over the dataset regardless of the number of items, whereas Apriori can require multiple passes.
    - The FP-tree structure allows for efficient pattern mining by compressing the dataset into a compact form.
2. **Reduced Candidate Generation:**
    - Apriori generates candidate itemsets at each iteration, which can lead to an exponential increase in the number of candidates, especially when dealing with large datasets or long patterns.
    - FP-growth does not need to generate candidate itemsets explicitly, which reduces the computational overhead significantly.
3. **Less Disk I/O:**
    - FP-growth typically requires less disk I/O compared to Apriori since it constructs a compact FP-tree structure during the initial database scan and uses it for subsequent pattern mining. This reduces the overhead of accessing the disk repeatedly.
4. **Better Memory Usage:**
    - FP-growth often uses less memory compared to Apriori, especially when dealing with large datasets. This is because it only needs to store the FP-tree and a few counters, whereas Apriori needs to maintain candidate itemsets and their support counts.

10. What is the single dimensional and multidimensional association rule?

Single-Dimensional Association Rule: This refers to association rules that involve only one dimension or attribute. For example, in a retail setting, a single-dimensional association rule might indicate that customers who buy bread are also likely to buy butter.



Multidimensional Association Rule: This refers to association rules that involve multiple dimensions or attributes. In the same retail setting, a multidimensional association rule might indicate that customers who buy bread and milk are also likely to buy eggs. This type of association rule takes into account combinations of multiple items across different dimensions.



Module VI: Business Intelligence

1. What is BI?BI Applications

The primary objective of BI is to support data-driven decision-making within organizations by providing insights derived from various data sources.

Key components of BI include:

- Data Integration: BI systems gather data from multiple sources, such as databases, spreadsheets, and enterprise applications, and integrate them into a centralized repository.
- Data Warehousing: Data warehouses are specialized databases designed for storing and managing large volumes of structured and unstructured data. They serve as a foundation for BI systems, providing a single source of truth for analytics.
- Data Analysis: BI tools enable users to analyze data to identify trends, patterns, and correlations. This analysis can involve querying, reporting, data mining, and statistical analysis.
- Data Visualization: BI platforms offer visualization tools to represent data in interactive charts, graphs, dashboards, and maps. Visualization enhances data comprehension and enables users to communicate insights effectively.
- Reporting and Dashboards: BI systems generate reports and dashboards that summarize key performance indicators (KPIs) and metrics. These reports provide stakeholders with timely and actionable information for monitoring business performance.
- Predictive Analytics: BI incorporates predictive modeling and forecasting techniques to anticipate future outcomes based on historical data. Predictive analytics enables organizations to make proactive decisions and plan for future scenarios.
- Self-Service BI: Modern BI solutions empower users to access and analyze data independently without relying on IT or data analysts. Self-service BI tools offer intuitive interfaces and pre-built templates for ad-hoc analysis and reporting.
- Business Performance Management: BI supports performance management processes by setting goals, tracking progress, and evaluating outcomes. Performance management features enable organizations to align strategies with objectives and measure success.

Applications of BI encompass various business functions, including:

- Financial Analysis: BI helps in financial reporting, budgeting, forecasting, and profitability analysis.
- Marketing and Sales: BI supports customer segmentation, campaign analysis, sales forecasting, and lead management.
- Supply Chain Management: BI facilitates inventory optimization, demand forecasting, supplier management, and logistics optimization.
- Human Resources: BI aids in workforce planning, talent management, employee engagement, and HR analytics.

- Operations Management: BI assists in process optimization, resource allocation, quality management, and operational efficiency.
- Risk Management: BI enables organizations to identify, assess, and mitigate risks through risk analysis, compliance monitoring, and fraud detection.

2. What are the advantages and disadvantages of BI system? Advantages of BI Systems:

Advantages of BI Systems:

- Data-Driven Decision Making: BI systems provide access to accurate, timely, and relevant data, enabling organizations to make informed decisions based on insights derived from data analysis.
- Improved Business Performance: By analyzing key performance indicators (KPIs) and metrics, BI systems help organizations identify areas of improvement, optimize processes, and enhance overall business performance.
- Competitive Advantage: BI enables organizations to gain competitive advantages by identifying market trends, understanding customer behavior, and responding quickly to changing business conditions.
- Enhanced Operational Efficiency: BI systems streamline data management processes, reduce manual effort, and automate reporting tasks, leading to greater operational efficiency and productivity.
- Increased Revenue and Profitability: By identifying new opportunities, optimizing pricing strategies, and targeting high-value customers, BI systems contribute to revenue growth and improved profitability.
- Better Customer Insights: BI enables organizations to analyze customer data, preferences, and purchasing patterns, leading to more personalized marketing campaigns, enhanced customer service, and increased customer satisfaction.
- Risk Management: BI systems help organizations identify and mitigate risks by monitoring compliance, detecting anomalies, and predicting potential issues before they escalate.
- Strategic Planning: BI facilitates strategic planning by providing insights into market trends, competitor analysis, and performance benchmarks, enabling organizations to develop long-term strategies and business plans.

Disadvantages of BI Systems:

- Cost: Implementing and maintaining a BI system can be expensive, involving significant upfront investments in software licenses, hardware infrastructure, and ongoing maintenance and support costs.
- Complexity: BI systems can be complex to implement and manage, requiring specialized skills and expertise in data integration, data modeling, and analytics. Complexity may also lead to longer implementation timelines and higher risk of project failure.
- Data Quality Issues: BI systems depend on accurate and reliable data for effective analysis. Poor data quality, inconsistent data sources, and data integration challenges can lead to inaccurate insights and flawed decision-making.

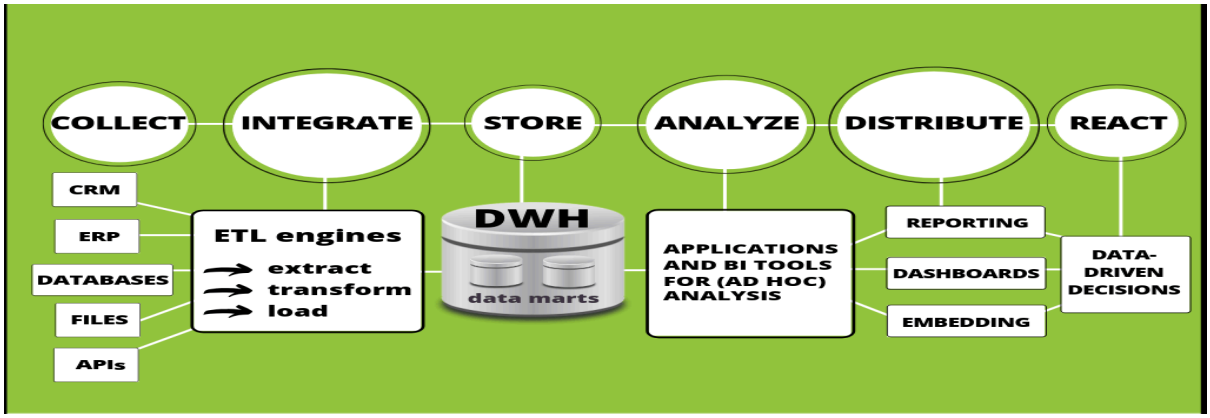
- Security and Privacy Concerns: BI systems involve handling sensitive business data, raising concerns about data security, privacy, and compliance with regulatory requirements. Unauthorized access, data breaches, and compliance violations are potential risks associated with BI systems.
- User Adoption Challenges: BI systems may face resistance from users who are unfamiliar with data analysis tools or skeptical about the value of data-driven decision-making. Training and change management efforts may be necessary to promote user adoption and acceptance.
- Scalability Issues: As data volumes and user demands grow, scalability becomes a concern for BI systems. Scaling up infrastructure, optimizing performance, and ensuring high availability may pose challenges for large-scale deployments.
- Dependency on IT Resources: BI systems often rely on IT support for data integration, system administration, and technical troubleshooting. This dependence on IT resources can create bottlenecks and slow down the responsiveness of BI initiatives.
- Overemphasis on Technology: Organizations may become overly focused on technology and tools, neglecting the importance of people and processes in BI initiatives. Successful BI implementations require a holistic approach that addresses cultural, organizational, and change management factors.

3. What are the components of BI architecture

There are various components and layers that business intelligence architecture consists of.

A solid BI architecture framework consists of:

1. Collection of data
2. Data integration
3. Storage of data
4. Data analysis
5. Distribution of data
6. Reaction based on insights



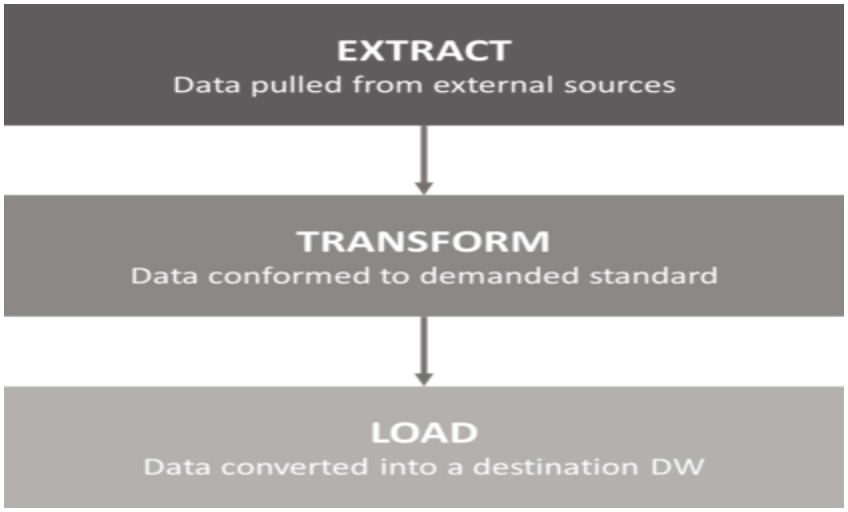
In above BI architecture diagram, we can see how the process flows through various layers, and now we will focus on each.

1. Collection of data

The first step in creating a stable architecture starts in gathering data from various data sources such as CRM, ERP, databases, files or APIs, depending on the requirements and resources of a company. Modern BI tools offer a lot of different, fast and easy data connectors to make this process smooth and easy by using smart ETL engines in the background. They enable communication between scattered departments and systems that would otherwise stay disparate. From a business point of view, this is a crucial element in creating a successful data-driven decision culture that can eliminate errors, increase productivity, and streamline operations. You have to collect data in order to be able to manipulate with it.

2. Data integration

When data is collected through scattered systems, the next step continues in extracting data and loading it to a data warehouse. This process is called ETL (Extract-Transform-Load).



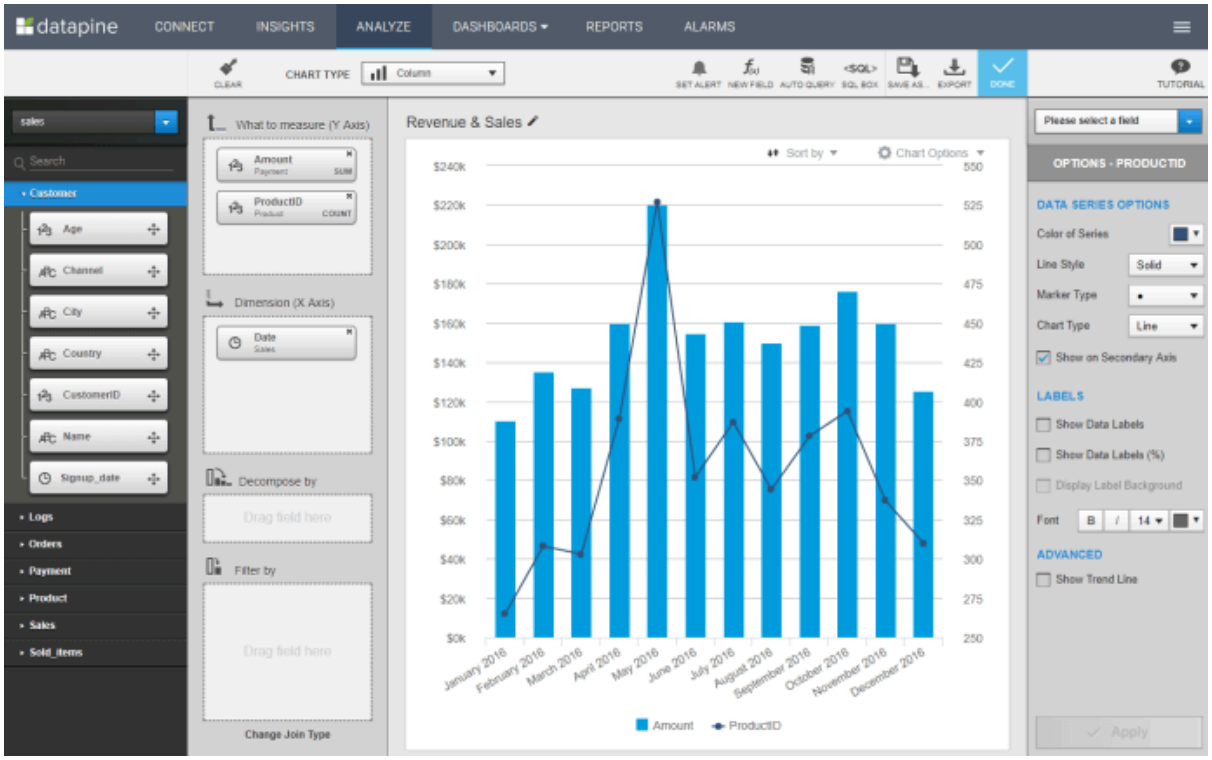
With an increasing amount of data generated today and the overload on IT departments and professionals, ETL as a service comes as a natural answer to solve complex data requests in various industries. The process is simple; data is pulled from external sources (from our step 1) while ensuring that these sources aren't negatively impacted with the performance or other issues. Secondly, data is conformed to the demanded standard. In other words, this (transform) step ensures data is clean and prepared to the final stage: loading into a data warehouse.

3. Data storage

Store data in DWH.

4. Analysis of data

In this step of our compact BI architecture, we will focus on the analysis of data after it's handled, processed, and cleaned in former steps with the help of data warehouse(s). The ubiquitous need for successful analysis for empowering businesses of all sizes to grow and profit is done through BI application tools. Especially when it comes to ad hoc analysis that enables freedom, usability, and flexibility in performing analysis and helping answer critical business questions swiftly and accurately.



This visual above represents the power of a modern, easy-to-use BI user interface. Modern BI tools empower business users to create queries via drag and drop, and build stunning data visualizations with a few clicks, even without profound technological knowledge. This simplifies the process of creating business dashboards, or an analytical report, and generate actionable insights needed for improving the operational and strategic efficiency of a business. The data warehouse works behind this process and makes the overall architecture possible.

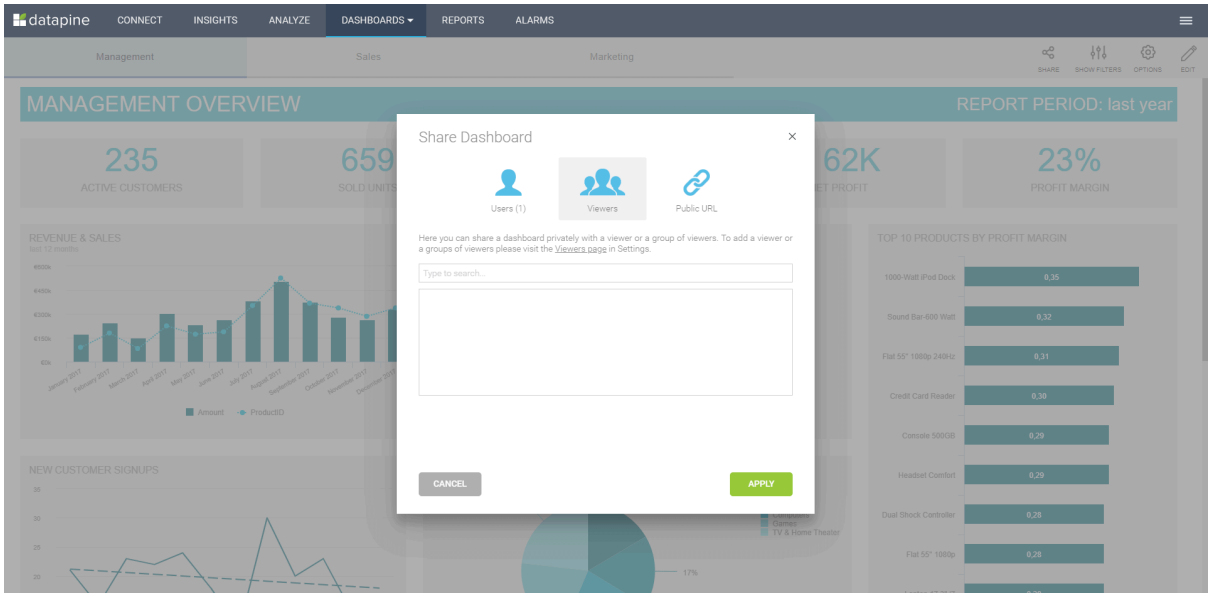
5. Data distribution

Data distribution comes as one of the most important processes when it comes to sharing information and providing stakeholders with indispensable insights to obtain sustainable business development. Distribution is usually performed in 3 ways:

a) Reporting via automated e-mails: Created reports can be shared with selected recipients on a defined schedule. The dashboards will be automatically updated on a daily, weekly or monthly basis which eliminates manual work and enables up to date information.

b) Dashboarding: Another reporting option is to directly share a dashboard in a secure viewer environment. The users you share with cannot make edits or change the content but can use assigned filters to manipulate data and interact with the dashboard.

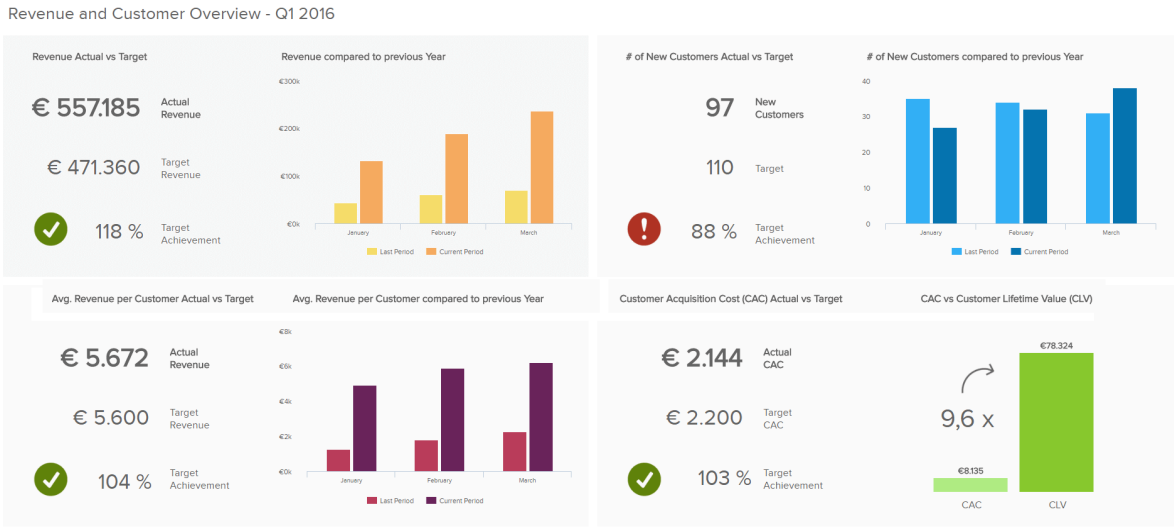
Another option is to share via public URL that enables users to access the dashboards even if they're outside of your organization, as shown in the picture below:



c) Embedding: This form of data distribution is enabled through embedded BI. Your own application can use dashboards as a mean of analytics and reporting without the need for labelling the BI tool in external applications or intranets.

6. Reactions based on generated insights

The final stage where the BI architecture expounds its power is the fundamental part of any business: creating data-driven decisions. Without the backbones of data warehousing and business intelligence, the final stage wouldn't be possible and businesses won't be able to progress. CEOs, managers, professionals, coworkers, and all the interested stakeholders can have the power of data to generate valid, accurate, data-based decisions that will help them move forward. Let's see this through one of our dashboard examples: the management KPI dashboard.



This dashboard is the final product on how data warehouse and business intelligence work together. The processes behind this visualization include the whole architecture which we have described, but it would not be possible to achieve without a firm data warehouse solution. Ultimately, this enables a high-level manager to get a comprehension of the strategic development and potential decisions for creating and maintaining a stable business.

On this particular dashboard, you can see the total revenue, as well as on a customer level, adding also the costs. The targets are also set so that the dashboard immediately calculates if they have been met or additional adjustments are needed from a management point of view. As revenue is one of the most important factors when evaluating if the business is growing, this management dashboard ensures all the essential data is visualized and the user can easily interact with each section, on a continual basis, making the decision processes more cohesive and, ultimately, more profitable.

4. What is a decision support system (DSS)? Examples https://www.tutorialspoint.com/management_information_system/decision_support_system.htm

Decision support systems (DSS) are interactive software-based systems intended to help managers in decision-making by accessing large volumes of information generated from various related information systems involved in organizational business processes, such as office automation system, transaction processing system, etc.

DSS uses the summary information, exceptions, patterns, and trends using the analytical models. A decision support system helps in decision-making but does not necessarily give a decision itself. The decision makers compile useful information from raw data, documents, personal knowledge, and/or business models to identify and solve problems and make decisions.

Example #1

Consider a retail company called ABC, aiming to optimize its inventory management. Using a Decision Support System (DSS), the company can analyze historical sales data, current market trends, and supplier information. The DSS employs predictive analytics to forecast demand for different products in various seasons. It also considers lead times, production costs, and storage expenses.
With this information, the company's decision-makers can model different scenarios, such as adjusting order quantities, reorder points, and safety stock levels. By simulating these scenarios and their potential outcomes, the DSS aids the company in making informed decisions about inventory levels, minimizing stockouts, reducing excess inventory costs, and ultimately improving overall operational efficiency and customer satisfaction.

Example #2

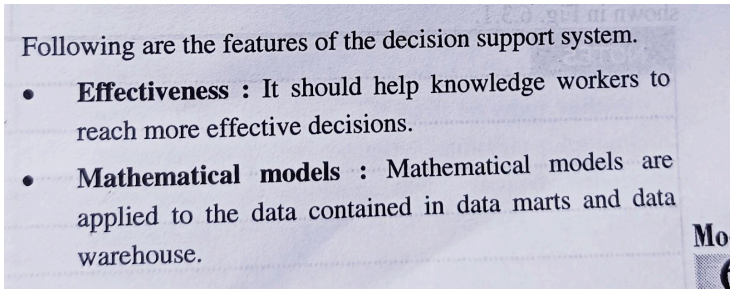
Consider a healthcare organization, XYZ, looking to optimize its patient scheduling process. With a Decision Support System (DSS), the organization can integrate data from patient appointments, physician availability, and treatment requirements. The DSS utilizes algorithms to identify scheduling patterns, peak hours, and resource constraints. By inputting patient preferences and medical priorities, the system generates optimized schedules that minimize wait times, maximize resource utilization, and improve patient flow. Decision-makers can then explore different scheduling scenarios, considering factors like appointment duration and required equipment. The DSS assists in creating efficient schedules, enhancing patient satisfaction, and streamlining the use of medical resources, ultimately leading to better patient care and operational effectiveness.

5. What are the characteristics of DSS?

- Support for decision-makers in semi-structured and unstructured problems.
- Support for managers at various managerial levels, ranging from top executive to line managers.
- Support for individuals and groups. Less structured problems often require the involvement of several individuals from different departments and organization levels.
- Support for interdependent or sequential decisions.
- Support for intelligence, design, choice, and implementation.
- Support for variety of decision processes and styles.
- DSSs are adaptive over time.

6. What are the advantages and disadvantages of DSS?

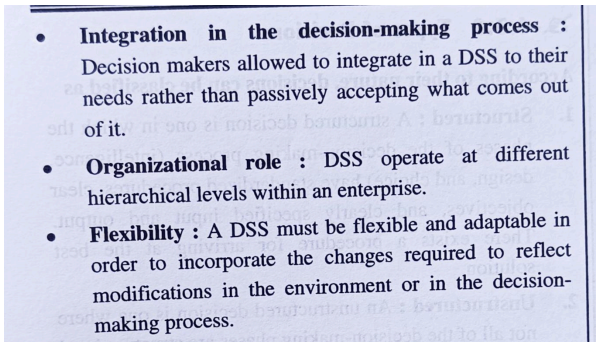
ADVANTAGES:



Following are the features of the decision support system.

- **Effectiveness :** It should help knowledge workers to reach more effective decisions.
- **Mathematical models :** Mathematical models are applied to the data contained in data marts and data warehouse.

Mo



- **Integration in the decision-making process :** Decision makers allowed to integrate in a DSS to their needs rather than passively accepting what comes out of it.
- **Organizational role :** DSS operate at different hierarchical levels within an enterprise.
- **Flexibility :** A DSS must be flexible and adaptable in order to incorporate the changes required to reflect modifications in the environment or in the decision-making process.

- Support for decision-makers in semi-structured and unstructured problems.
- Support for managers at various managerial levels, ranging from top executive to line managers.
- Support for individuals and groups. Less structured problems often require the involvement of several individuals from different departments and organization levels.
- Support for interdependent or sequential decisions.
- Support for intelligence, design, choice, and implementation.
- Support for variety of decision processes and styles.
- DSSs are adaptive over time.

DISADVANTAGES:

1. Cost and Complexity: Implementing and maintaining a DSS can be costly and complex. It requires significant investment in terms of hardware, software, training, and ongoing support. Additionally, designing and developing a DSS to meet the specific needs of an organization can be a complex and time-consuming process.
2. Data Quality and Reliability: The effectiveness of a DSS depends heavily on the quality and reliability of the underlying data. Poor data quality, incomplete data, or inaccuracies can undermine the reliability of the insights generated by the system and lead to erroneous decisions.
3. Privacy and Security Concerns: DSS often involves the processing and analysis of sensitive and confidential data, raising concerns about privacy and security. Ensuring the confidentiality, integrity, and availability of data within a DSS environment is crucial to prevent unauthorized access, data breaches, and compliance violations.

7. What are the types of DSS?

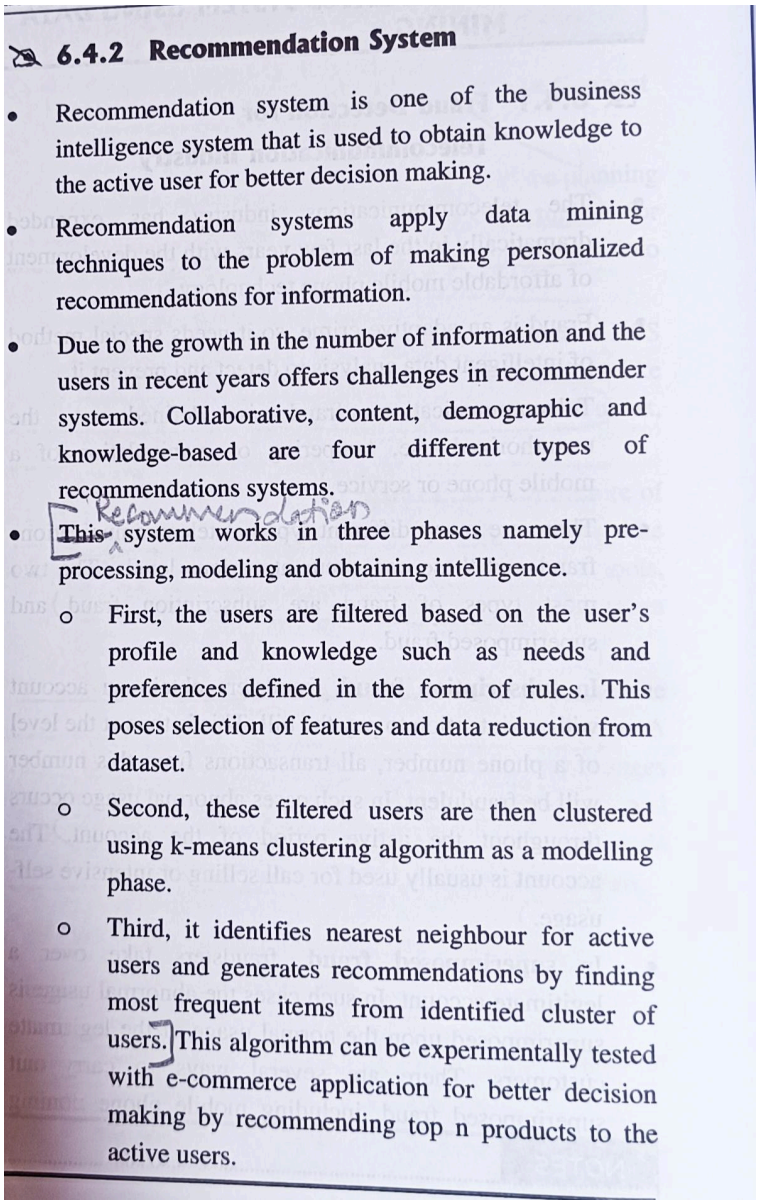
6.3.1 Types of DSS

There are several types of DSSs as discussed below.

1. **Communication-driven DSS** which enables cooperation, supporting more than one person working on a shared task; examples include integrated tools like Google Docs or Microsoft Groove.
2. **Document-driven DSS** which manages, retrieves, and manipulates unstructured information in a variety of electronic formats.
3. **Knowledge-driven DSS** provides specialized problem-solving expertise stored as facts, rules, procedures, or in similar structures
4. **Model-driven DSS** emphasizes access to and manipulation of a statistical, financial, optimization, or simulation model. Model-driven DSS use data and parameters provided by users to assist decision makers in analyzing a situation; they are not necessarily data-intensive.
5. **Data-driven DSS** (or data-oriented DSS) emphasizes access to and manipulation of a time series of internal company data and, sometimes, external data. A data-driven DSS, which we will focus on, emphasizes access to and manipulation of a time series of internal company data and sometimes external data. Simple file systems accessed by query and retrieval tools provide the most elementary level of functionality. Data warehouse systems that allow the manipulation of data by computerized tools tailored to a specific task and setting or by more general tools and operators provide additional functionality. Data-driven DSS with online analytical processing (OLAP) provide the highest level of functionality.

8. Development of a business intelligence system using Data Mining for business Applications like Fraud Detection, Recommendation, Retail etc.

RECOMMENDATION SYSTEM



6.4.2 Recommendation System

- Recommendation system is one of the business intelligence system that is used to obtain knowledge to the active user for better decision making.
- Recommendation systems apply data mining techniques to the problem of making personalized recommendations for information.
- Due to the growth in the number of information and the users in recent years offers challenges in recommender systems. Collaborative, content, demographic and knowledge-based are four different types of recommendations systems.
- This system works in three phases namely pre-processing, modeling and obtaining intelligence.
    - First, the users are filtered based on the user's profile and knowledge such as needs and preferences defined in the form of rules. This poses selection of features and data reduction from dataset.
    - Second, these filtered users are then clustered using k-means clustering algorithm as a modelling phase.
    - Third, it identifies nearest neighbour for active users and generates recommendations by finding most frequent items from identified cluster of users. This algorithm can be experimentally tested with e-commerce application for better decision making by recommending top n products to the active users.

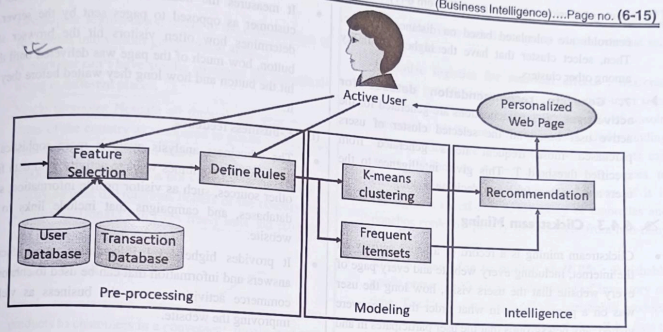


Fig. 6.4.2 : Recommendation System

The steps involved in the Recommendation System are given below :

1. Identifying the dataset
2. Choose the consideration columns/features
3. Filtering objects by defining the rules
4. Identifying frequent items
5. Cluster objects using k-means clustering
6. Find nearest neighbour of active user
7. Generate recommendation dataset for active user

1. Identifying the dataset : To maintain the data systematically and efficiently, database and data warehouse technologies are used. The data warehouse not only deals with the business activities but also contains the information about the user that deals with the business.

2. Choose the consideration columns /features : Once the dataset D has been identified, the next step of the system is to choose the consideration ~~column or filtering~~ columns/features. That is, from the whole dataset, the columns/subset of features to be considered for our work are chosen. This includes the elimination of the irrelevant column in the dataset. The irrelevant column/feature may be the one which provide less information about the dataset.

3. Filtering objects by defining the rules :

From the consideration dataset, the objects can be grouped under stated conditions that are defined in terms of rules. That is, for each column that is considered, specify the rule to extract the necessary domain from the original dataset. This rule is considered to be the threshold value T. The domain can be chosen by identifying the frequent items from the dataset.

4. Identifying frequent items : The frequent items can be identified by analyzing the repeated value in the consideration column satisfying the support count and the confidence threshold. This will create a new dataset D'.

5. Cluster objects using k-means clustering :

Upon forming the new dataset D', the objects in D' are clustered based on similarity of objects using k-means clustering. k-means clustering is a method of classifying or grouping objects into k clusters (where k is the number of clusters). The clustering is performed by minimizing the sum of squared distances between the objects and the corresponding centroid. The result consists of cluster of objects with their labels/classes.

6. Find nearest neighbour of active user :

In order to find the nearest neighbours of the active user, similarity of the active user between cluster

centroids are calculated based on distance measure. Then, select cluster that have the highest similarity among other clusters.

7. Generate recommendation dataset for active user : Recommendations are generated for the active user based on the selected cluster of users purchased most frequent items generated from specified threshold T. This gives intelligence to the users and business for better decision making.

FRAUD DETECTION SYSTEM

6.4.1 Fraud Detection for Telecommunication Industry

- The telecommunications industry has expanded dramatically in the last few years with the development of affordable mobile phone technology.
- Fraud is an adaptive crime, so it needs special method of intelligent data analysis to detect and prevent it.
- Telecommunication fraud is defined as the unauthorized use, tampering or manipulation of a mobile phone or service.
- There are many different types of telecommunications fraud and these can occur at various levels. The two most types of fraud are subscription fraud and superimposed fraud.
- In **subscription fraud**, fraudsters obtain an account without intention to pay the bill. This is thus at the level of a phone number, all transactions from this number will be fraudulent. In such cases abnormal usage occurs throughout the active period of the account. The account is usually used for call selling or intensive self-usage.
- In **superimposed fraud**, fraudsters take over a legitimate account. In such cases the abnormal usage is superimposed upon the normal usage of the legitimate customers. There are several ways to carry out superimposed fraud, including mobile phone cloning

and obtaining calling card authorization details. Examples of such cases include cellular cloning, calling card theft and cellular handset theft. Superimposed fraud will generally occur at the level of individual calls; the fraudulent calls will be mixed in with the justified ones.

- Other types of telecommunications fraud include ghosting (technology that tricks the network in order to obtain free calls) and insider fraud where telecommunication company employees sell information to criminals that can be explained for fraudulent gain.
- These method exists in the areas of Knowledge Discovery in Databases (KDD), Data Mining, Machine Learning and Statistics. They offer applicable and successful solutions in different areas of fraud crimes.
- At a low level, simple rule-based detection systems use rules such as the apparent use of the same phone in two very distant geographical locations in quick succession, calls which appear to overlap in time and very high value and very long calls.
- At a higher level, statistical summaries of call distributions (often called profiles or signature at the user level) are compared with thresholds determined either by experts or by application of supervised learning methods to known fraud/non-fraud cases.
- Some forensic accountants specialize in forensic analytics which is the procurement and analysis of electronic data to reconstruct, detect, and otherwise support a claim of financial fraud.
- The main steps in forensic analytics are (a)data collection, (b)data preparation, (c)data analysis, and(d)reporting.
- For example, forensic analytics may be used to review an employees' purchasing card activity to assess whether any of the purchases were diverted or divertible for personal use.

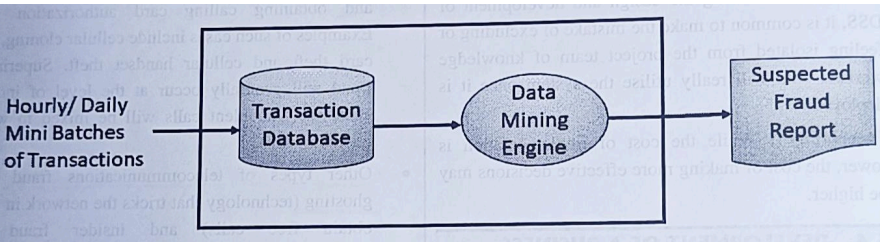


Fig. 6.4.1 : Fraud Detection

- Techniques used for fraud detection fall into two primary classes: Statistical techniques and Artificial intelligence.
- Examples of Statistical data analysis techniques are :

1. Data pre-processing techniques for detection, validation, error correction, and filling up of missing or incorrect data.
2. Calculation of various statistical parameters such as averages, performance metrics. For example, the average may include average length of call, average number of calls per month.
3. Computing user profiles.
4. Time-series analysis of time-dependent data.
5. Clustering and classification to find patterns and association among groups of data.

Examples of AI techniques are :

1. Data mining to classify, cluster, and segment the data and automatically find associations and rules in the data that may signify interesting patterns, including those related to fraud.
2. Expert systems to encode expertise for detecting fraud in the form of rules.
3. Pattern recognition to detect approximate classes, clusters, or patterns of suspicious behavior either automatically or to match given inputs.
4. Machine learning techniques to automatically identify characteristics of fraud.

RETAIL INDUSTRY

6.4.5 Retail Industry

- Retail organizations thrive by providing quality products to customers in a convenient, timely, and cost-effective manner.
- Understanding emerging customer shopping patterns can assist retailers in organising their products, inventory, store layout, and web presence in order to delight their customers, thereby increasing revenue and profits.
- Retailers generate a lot of transaction and logistics data that can be used to solve problems.
- **Optimize inventory levels at different locations :** Retailers must carefully manage their inventories. Carrying too much inventory incurs carrying costs, whereas carrying too little inventory can result in stock-outs and missed sales opportunities. Dynamic sales trend prediction can assist retailers in moving inventory to where it is most in demand. Online retailers can provide their suppliers with real-time information about their items' sales, allowing the suppliers to deliver their product to the right locations and reduce stock-outs.
- **Improve store layout and sales promotions :** Using a market basket analysis, you can create predictive models of which products frequently sell together. This understanding of product affinities can assist retailers in co-locating those products. Alternatively, those affinity products could be placed further apart in order to force the customer to walk the length and breadth of the store, exposing them to other products. Promotional discounted product bundles can be created to promote a

non-selling item and also a group of products that sell well together.

- **Optimize logistics for seasonal effects :** Seasonal products provide extremely profitable short-term sales opportunities, but they also pose the risk of unsold inventories at the end of the season. Understanding which products are in season in which markets can assist retailers in dynamically managing prices to ensure inventory is sold during the season. If it is raining in a specific area, inventory of umbrellas and ponchos could be quickly moved there from non-rainy areas to help increase sales.
- **Reduce losses due to limited shelf life :** Perishable goods present difficulties in disposing of inventory on time. Tracking sales trends allows perishable products that are at risk of not selling before their sell-by date to be appropriately discounted and promoted.

NEW EXTRA QUESTIONS

1. What are the different ways to handle missing data?

1. Ignore the tuple: usually done when class label is missing assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
2. Fill in the missing value manually: (tedious + infeasible)
3. Use a global constant to fill in the missing value: Replace all missing attribute values by the same constant such as a label like "Unknown"
4. Filling the value with the help of mean , If there is numeric type of that than the missing values can be filled using their combined mean
   Eg - Year: 2018,____,2020
   Missing value will be 2019
5. Another way is Filling the value with the help of Median , If there is numeric type of that than the missing values can be filled using their combined median
   Eg - Year: 2018,____,2020,2021,2022
   So missing value would be 2020 as it is median
6. Filling the value with the help of mode, If there is categorical type of that than the missing values can be filled using their combined mode
   Eg - Class: D15A,D15A,D15A,_____,D15B
   Here missing value would be D15A
7. Also interpolation can be used to fill up the missing values interpolation basically calculates the value that can be inserted at the missing place based on their neighbouring values
   Eg - Year : 2018,____,2020,2021
   Here based on interpolation 2019 will be filled
8. Forward Filling can also be used to fill the missing values ,missing value is filled based on the corresponding value in the previous row.
   Eg - Year: 2018,____,2020
   Here Value will be 2018
9. Backward Filling can also be used to fill the missing values ,missing value is filled based on the corresponding value in the next row.

Eg - Year: 2018,_____,2020
Here Value will be 2020

10. Use the attribute mean or median for all samples belonging to the same class as the given tuple.
11. Use The most probable value: inference-based such as Bayesian formula or decision tree

2. What are the different ways to handle noisy data?

Handle Noisy Data-

a. Binning-

Binning is a technique where we sort the data and then partition the data into equal frequency bins. Then you may either replace the noisy data with the bin mean bin median or the bin boundary.

There are three methods for smoothing data in the bin-

Smoothing by bin mean method: In this method, the values in the bin are replaced by the mean value of the bin.

Smoothing by bin median: In this method, the values in the bin are replaced by the median value.

Smoothing by bin boundary: In this method, the minimum and maximum values of the bin values are taken, and the closest boundary value replaces the values.

Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (equi-depth) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by bin boundaries:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

b. Regression-

This is used to smooth the data and will help to handle data when unnecessary data is present.
For the analysis, purpose regression helps to decide the variable which is suitable for our analysis.

- Linear regression refers to finding the best line to fit between two variables so that one can be used to predict the other.
- Multiple linear regression involves more than two variables.

Using regression to find a mathematical equation to fit into the data helps to smooth out the noise.

c. Outlier Analysis-

Outliers may be detected by clustering, where similar or close values are organized into the same groups or clusters. Outliers are extreme values that deviate from other observations on data. Outliers can be the following kinds, such as:

- Univariate outliers can be found when looking at a distribution of values in a single feature space.
- Multivariate outliers can be found in an n-dimensional space (of n-features). Looking at distributions in n-dimensional spaces can be very difficult for the human brain. That is why we need to train a model to do it for us.
- Point outliers are single data points that lay far from the rest of the distribution.
- Contextual outliers can be noise in data, such as punctuation symbols when realizing text analysis or background noise signal when doing speech recognition.
- Collective outliers can be subsets of novelties in data, such as a signal that may indicate the discovery of new phenomena.

3. What are the different data transformation strategies?

DATA TRANSFORMATION:

- Data transformation is a technique used to convert the raw data into a suitable format that efficiently eases data mining and retrieves strategic information.
- Data transformation includes data cleaning techniques and a data reduction technique to convert the data into the appropriate form.
- Provide patterns that are easier to understand.
- Data transformation changes the format, structure, or values of the data and converts them into clean, usable data.
- Data Transformation Techniques-

  1. Data Smoothing-

- Data smoothing is a process that is used to remove noise from the dataset using some algorithms.
- It allows for highlighting important features present in the dataset. It helps in predicting the patterns.
- When collecting data, it can be manipulated to eliminate or reduce any variance or any other noise form.
- The concept behind data smoothing is that it will be able to identify simple changes to help predict different trends and patterns. This serves as a help to analysts or traders who need to look at a lot of data which can often be difficult to digest for finding patterns that they wouldn't see otherwise.
- The noise is removed from the data using the techniques such as binning, regression, clustering.

2. Data Aggregation-
   - Data collection or aggregation is the method of storing and presenting data in a summary format.
   - This is a crucial step since the accuracy of data analysis insights is highly dependent on the quantity and quality of the data used.
   - Gathering accurate data of high quality

3. Data Generalization-

- It converts low-level data attributes to high-level data attributes using concept hierarchy.
- This conversion from a lower level to a higher conceptual level is useful to get a clearer picture of the data. Data generalization can be divided into two approaches:
  - Data cube process (OLAP) approach.
  - Attribute-oriented induction (AOI) approach.
- For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).

4. Data Transformation (by Normalization)-

Data scaled to fall within a small, specified range.

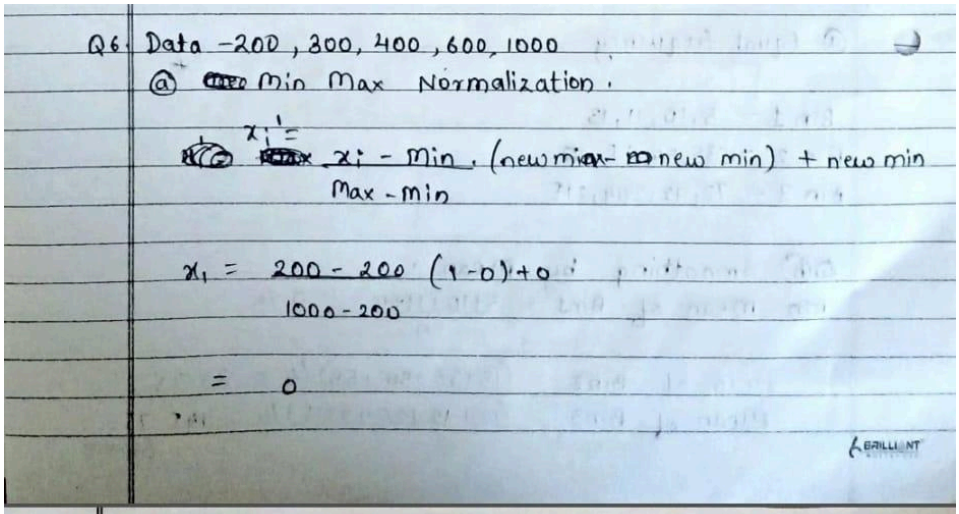
It can be performed by three methods-

a. Min-max Normalization
b. Z-score Normalization
c. Decimal scaling Normalization

5. Attribute Construction-

- In the attribute construction method, the new attributes consult the existing attributes to construct a new data set that eases data mining.
- New attributes are created and applied to assist the mining process from the given attributes. This simplifies the original data and makes the mining more efficient.
- For example, suppose we have a data set referring to measurements of different plots, i.e., we may have the height and width of each plot. So here, we can construct a new attribute 'area' from attributes 'height' and 'weight'.
- Attribute construction also helps understand the relations among the attributes in a data set.

4. Problems on min max ,z score and decimal scaling normalization.



Q6. Data -200, 300, 400, 600, 1000

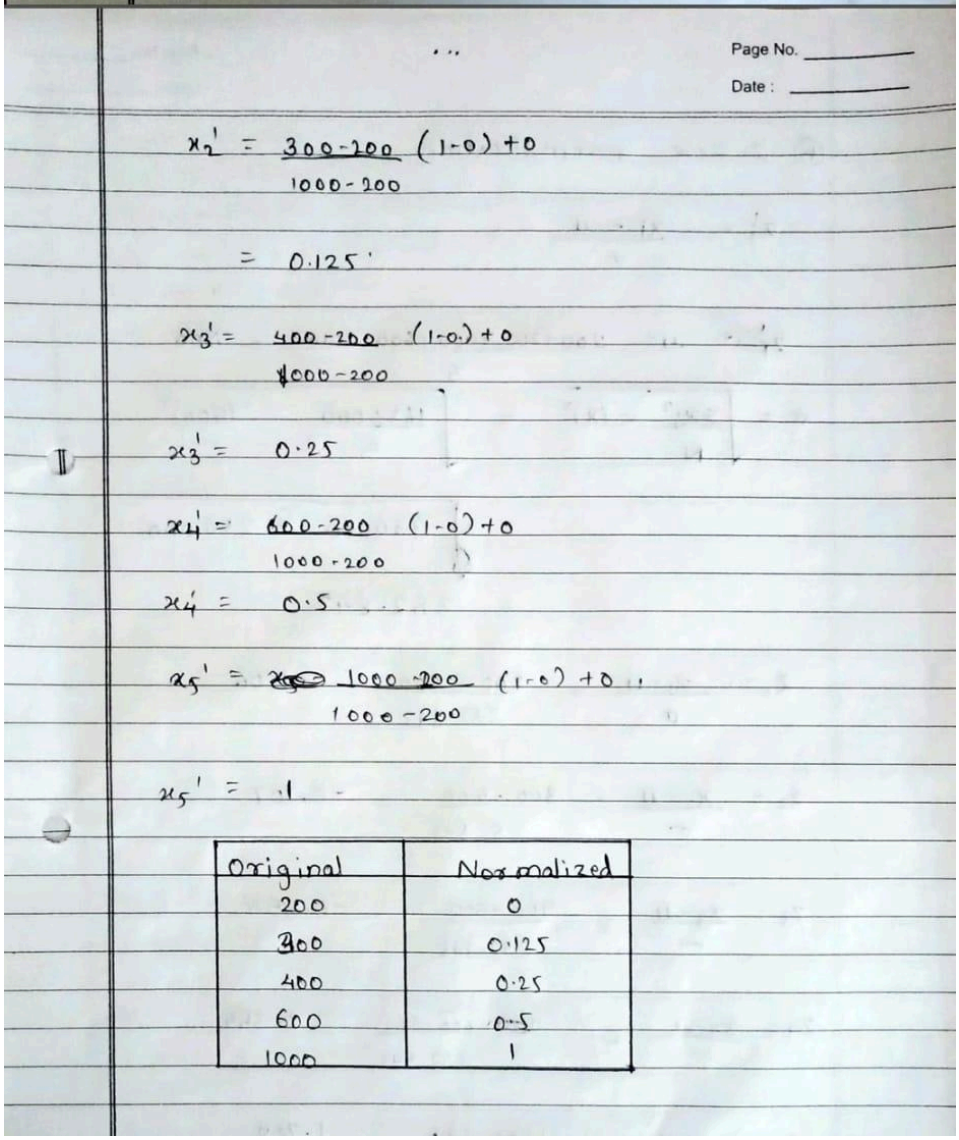
(a) Min Max Normalization.

$$x_i' = \frac{x_i - min}{max - min}(new\ max - new\ min) + new\ min$$

$$x_1 = \frac{200 - 200}{1000 - 200}(1-0) + 0$$

$$= 0$$

BRILLIANT



Page No.

Date :

$$x_2' = \frac{300 - 200}{1000 - 200}(1-0) + 0$$

$$= 0.125$$

$$x_3' = \frac{400 - 200}{1000 - 200}(1-0) + 0$$

$$x_3' = 0.25$$

$$x_4' = \frac{600 - 200}{1000 - 200}(1-0) + 0$$

$$x_4' = 0.5$$

$$x_5' = \frac{1000 - 200}{1000 - 200}(1-0) + 0$$

$$x_5' = 1$$

Original	Normalized
200	0
300	0.125
400	0.25
600	0.5
1000	1

Page No. _____

Date : _____

(b) Z-score normalization

$$z_i' = \frac{x_i - \mu}{\sigma}$$

$$\mu = \frac{200+300+400+600+1000}{5} = 500$$

$$\sigma = \sqrt{\frac{\Sigma x_i^2}{N} - (\bar{x})^2} = \sqrt{\frac{1650000}{5} - (500)^2}$$

$$= \sqrt{330000 - 250000}$$

$$= 282.843$$

$$z_1 = \frac{x_1 - \mu}{\sigma} = \frac{200 - 500}{282.843} = -1.06$$

$$z_2 = \frac{x_2 - \mu}{\sigma} = \frac{300 - 500}{282.843} = -0.707$$

$$z_3 = \frac{x_3 - \mu}{\sigma} = \frac{400 - 500}{282.843} = -0.354$$

$$z_4 = \frac{x_4 - \mu}{\sigma} = \frac{600 - 500}{282.843} = 0.354$$

$$z_5 = \frac{x_5 - \mu}{\sigma} = \frac{1000 - 500}{282.843} = 1.768$$

BRILLIANT

Page No. _____

Date : _____

© Z-score normalization :
Mean Absolute Deviation

$$Z = \frac{x - \mu}{A}$$

$$A = \frac{|200-500| + |300-500| + |400-500| + |600-500| + |1000-500|}{5}$$

$$A = 240$$

$$Z_1 = \frac{200-500}{240} = -1.25$$

$$Z_2 = \frac{300-500}{240} = -0.83$$

$$Z_3 = \frac{400-500}{240} = -0.42$$

$$Z_4 = \frac{600-500}{240} = 0.42$$

$$Z_5 = \frac{1000-500}{240} = 2.083$$

Page No.

Page No.

Date :

(d) Decimal Scaling.

smallest integer such that $\max\left(\frac{x_i}{10^j}\right) \leq 1$

$\therefore j = 3$

$z_1 = \frac{200}{10^3} = 0.2$

$z_2 = \frac{300}{10^3} = 0.3$

$z_3 = \frac{400}{10^3} = 0.4$

$z_4 = \frac{600}{10^3} = 0.6$

$z_5 = \frac{1000}{10^3} = 1$

5. State different data reduction strategies.

DATA REDUCTION:

Why data reduction?

- A database/data warehouse may store terabytes of data
- Complex data analysis/mining may take a very long time to run on the complete data set

Data reduction
Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results

Data reduction strategies

- Dimensionality reduction — e.g., remove unimportant attributes
    - Wavelet Transform-
      The discrete wavelet transform (DW
      T) is a linear signal processing technique that, when applied to a data vector X, transforms it to a numerically different vector, X' of wavelet coefficients.
      The compressed data is obtained by retaining the smallest fragment of the strongest wavelet coefficients. Wavelet transform can be applied to data cubes, sparse data, or skewed data.

    - PCA
      Suppose we have a data set to be analyzed that has tuples with n attributes. The principal component analysis identifies k independent tuples with n attributes that can represent the data set.
      In this way, the original data can be cast on a much smaller space, and dimensionality reduction can be achieved. Principal component analysis can be applied to sparse and skewed data.

    - Attribute subset selection
      The attribute subset selection reduces the volume of data by eliminating redundant and irrelevant attributes.
      The most suitable subset of attributes are selected by using techniques like forward selection, backward elimination, decision tree induction or a combination of forward selection and backward elimination.
      The attribute subset selection ensures that we get a good subset of original attributes even after eliminating the unwanted attributes. The resulting probability of data distribution is as close as possible to the original data distribution using all the attributes.

- Numerosity reduction — e.g., fit data into models
  Reduce data volume by choosing alternative, smaller forms of data representation
    - Parametric methods: Regression, log-linear models

Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
Example: Log-linear models—obtain value at a point in m-D space as the product on appropriate marginal subspaces; Linear and multiple regression

- Non parametric methods: histograms, clustering, sampling and data cube aggregation
  Do not assume models
  Methods- Clustering, Sampling, Histograms

- Data Compression-
  - String compression
    There are extensive theories and well-tuned algorithms
    Typically lossless
    But only limited manipulation is possible without expansion
  - Audio/video compression
    Typically lossy compression, with progressive refinement
    Sometimes small fragments of signal can be reconstructed without reconstructing the whole
  - Time sequence is not audio
    Typically short and vary slowly with time

6. Binning different types and problems bases on binning

Q5. Data - 5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215.

→

(a) Equal frequency.

Bin 1 - 5, 10, 11, 13.
Bin 2 - 15, 35, 50, 55.
Bin 3 - 72, 92, 204, 215

(b) Smoothing by mean.

Mean of Bin1 = $\frac{5+10+11+13}{4} = 9.75$

Mean of Bin2 = $(15+35+50+55)/4 = 38.75$
Mean of Bin3 = $(72+92+204+215)/4 = 145.75$

BRILLIANT

Page No. _____
Date : _____

Bin 1 - 9.75, 9.75, 9.75, 9.75
Bin 2 - 38.75, 38.75, 38.75, 38.75
Bin 3 - 145.75, 145.75, 145.75, 145.75

(c) Equal width

$$\text{width} = \frac{\text{Max} - \text{Min.}}{\text{No. of Bins}} = \frac{215 - 5}{3} = 70$$

Range of Bin1 = (5), (70+5)
Bin2 = 75, (75+70)
Bin3 = 145, (145+70)

∴ Bin1 = 5, 10, 11, 13, 15, 35, 50, 55, 72.
Bin2 = 92,
Bin3 = 204, 215

7. **What is noise? Explain data smoothing methods as noise removal technique to divide given data into bins of size 3**
   Refer Q1 and Q6
8. **In real-world data, tuples with missing values for some attributes are a common occurrence. Describe various methods for handling this problem**
   Refer Q1
9. **What is the Class Imbalance problem? Explain with Example. (GPT)(Better ans milta iska to plej replace crow :))**
   The Class Imbalance problem occurs when the distribution of classes in a dataset is highly skewed, meaning that one class is significantly more prevalent than the others. This imbalance can lead to challenges in training machine learning models, particularly when the minority class (the less frequent class) is of particular interest but may be overlooked due to its scarcity.

   Here's an example to illustrate the Class Imbalance problem: Imagine you're working on a project to develop a model for credit card fraud detection. In your dataset, you have information about thousands of credit card transactions, where the majority of transactions are legitimate (non-fraudulent), but only a tiny fraction are actually fraudulent.

   Let's say out of 10,000 transactions, only 50 are fraudulent. This means that the fraud class constitutes just 0.5% of the dataset, while the non-fraud class makes up the remaining 99.5%.

   Now, if you were to train a machine learning model on this dataset without addressing the class imbalance, the model might simply learn to always predict the majority class (non-fraud) because it achieves high accuracy by doing so. As a result, the model may completely fail to identify instances of fraud, which is the critical task in this scenario.

Or

The main class of interest is rare. That is, the data set distribution reflects a
significant majority of the negative class and a minority positive class.

For example, in fraud detection applications, the class of interest (or
positive class) is “fraud,” which occurs much less frequently than the
negative “nonfraudulant” class.

In medical data, there may be a rare class, such as “cancer.”

Suppose that you have trained a classifier to classify medical data tuples,
where the class label attribute is “cancer” and the possible class values are
“yes” and “no.”

An accuracy rate of, say, 97% may make the classifier seem quite accurate, but what if only, say, 3% of the training tuples are actually
cancer? Clearly, an accuracy rate of 97% may not be acceptable—the
classifier could be correctly labeling only the noncancer tuples, for instance,
and misclassifying all the cancer tuples. Instead, we need other measures,
which assessess how well the classifier can recognize the positive tuples

(cancer = yes) and how well it can recognize the negative tuples (cancer =
no).

The main class of interest is rare.

e.g In medical data, there may be a rare class, such as “cancer.”

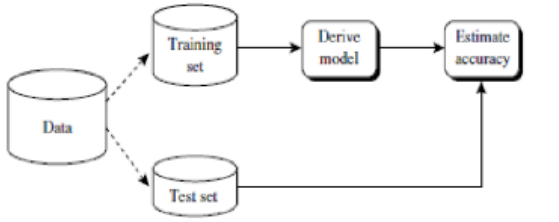
For Class Imbalance problems , we need other measures, which assesses
how well the classifier can recognize the positive tuples (e.g. cancer =
yes) and how well it can recognize the negative tuples (e.g. cancer = no).

10. What are the different Methods for evaluating accuracy of the classifier(Holdout method ,Random Subsampling,Cross Validation, Bootstrap method)

Holdout Method:

- In the Holdout method, the dataset is divided into two parts: a training set and a testing set.
- The classifier is trained on the training set and then evaluated on the separate testing set.
- The evaluation metrics, such as accuracy, are calculated based on the performance of the classifier on the testing set.
- Typically, a common split ratio is 70% training data and 30% testing data, but this can vary depending on the size and nature of the dataset.

- Holdout method
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation ---Pessimistic estimate



Estimating accuracy with the holdout method.

  - Random subsampling: a variation of holdout
    - Repeat holdout method k times, accuracy = avg. of the accuracies obtained from each iteration

Random Subsampling:

- Similar to the Holdout method, but instead of performing a single split of the dataset into training and testing sets, Random Subsampling involves repeating this process multiple times.
- Each time, a random subset of the data is selected as the training set, and the remaining data is used as the testing set.
- The evaluation metrics are then averaged over all iterations to obtain a more stable estimate of the classifier's performance.

Cross-Validation:

- Cross-Validation is a more robust method compared to Holdout and Random Subsampling.
- It involves partitioning the dataset into k equal-sized folds (or subsets).
- The classifier is trained on k-1 folds and tested on the remaining fold, iteratively for k times (each fold serves as the testing set once).

- The evaluation metrics are averaged over all iterations to obtain a comprehensive assessment of the classifier's performance.
- Common variants include k-fold cross-validation and stratified k-fold cross-validation, where class distributions are preserved in each fold.

Cross-validation (*k*-fold, where k = 10 is most popular)

- Randomly partition the initial data into *k mutually exclusive* subsets or folds,D1 to Dk, each of approximately equal size.
- Perform training and testing k times.
- At *i*-th iteration, use D_i as test set and others collectively as training set
- Unlike the holdout and random subsampling methods, here each sample is used the same number of times for training and once for testing.
- The results of each iteration are averaged, to find accuracy which is used as a performance metric to compare the efficiency of different models.
- The k-fold cross-validation technique generally produces less biased models as every data point from the original dataset will appear in both the training and testing set.
- This method is optimal if you have a limited amount of data.

Leave-one-out: A special case of k folds where k is set to number of initial tuples i.e only one sample is left out at a time for test set.

Stratified cross-validation: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data.

Bootstrap Method:

- Bootstrap is a resampling technique that involves generating multiple datasets of the same size as the original dataset by randomly sampling with replacement.
- Each bootstrap sample is used to train and test the classifier.
- The evaluation metrics are then averaged over all bootstrap samples to obtain a robust estimate of the classifier's performance.
- Bootstrap is particularly useful when the dataset is limited in size or when the distribution of the data is complex and not easily represented by a simple parametric model.

- **Bootstrap**
  - Works well with small data sets
  - Samples the given training tuples uniformly *with replacement*
    - i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
- Several bootstrap methods, and a common one is **.632 bootstrap**
  - Suppose we are given a data set of d tuples. The data set is sampled d times, with replacement, resulting in a training set of d samples. The data tuples that did not make it into the training set end up forming the test set. On an average 63.2% of the original data will end up in the bootstrap sample, and the remaining 36.8% will form the test set (since $(1 - 1/d)^d \approx e^{-1} = 0.368$)
  - Repeat the sampling procedue k times, overall accuracy of the model:

$$Acc(M) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set}),$$

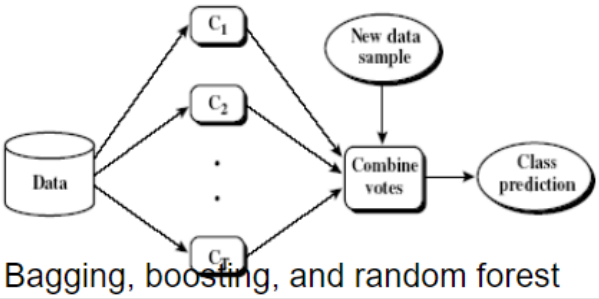
where $Acc(M_i)_{test\_set}$ is the accuracy of the model obtained with bootstrap sample i when it is applied to test set i. $Acc(M_i)_{train\_set}$ is the accuracy of the model obtained with bootstrap sample i when it is applied to the original set of data tuples.

4

11. Explain the Ensemble Methods for Improving the Accuracy of classifier(Bagging, boosting, and random forest)

Ensemble methods are techniques that combine multiple base classifiers to improve the overall performance and accuracy of the model. Three popular ensemble methods are Bagging, Boosting, and Random Forest. Let's delve into each of them:

- Ensemble methods
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, …, M_k, with the aim of creating an improved model M*
  - A given data set, *D*, is used to create *k* training sets, *D*1, *D*2, : : : , *Dk*, where *Di(1<=i<=k-1)* is used to generate classifier *Mi* .
  - Given a new data tuple to classify, the base classifiers each vote by returning a class prediction. The ensemble returns a class prediction based on the votes of the base classifiers.
  - Ensembles yield better results when there is significant diversity among the models.



  - Examples are Bagging, boosting, and random forest

5

Bagging (Bootstrap Aggregating):

- Bagging involves creating multiple subsets of the original dataset by sampling with replacement (bootstrap sampling).
- Each subset is used to train a base classifier independently.
- Predictions from all base classifiers are then aggregated, typically by averaging (for regression) or voting (for classification).
- Bagging helps to reduce variance and overfitting by creating diverse base classifiers, each trained on a slightly different subset of the data.
- The most famous example of Bagging is the Random Forest algorithm.

Bagging: Bootstrap Aggregation

- Analogy: Diagnosis based on multiple doctors' majority vote
- Training
  - Given a set D of *d* tuples, at each iteration *i*, a training set D_i of *d* tuples is sampled with replacement from D (i.e., bootstrap)
  - A classifier model M_i is learned for each training set D_i
- To classify an unknown sample **X** ,
  - Each classifier M_i returns its class prediction which counts as one vote.
  - The bagged classifier M*, counts the votes and assigns the class with the most votes to **X**
- Bagging can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple
- Accuracy
  - Often significant accuracy than a single classifier derived from D
  - It will not be considerably worse and is more robust to the effects of noisy data and overfitting.
  - The increased accuracy occurs because the composite model reduces the variance of the individual classifiers

6

Bagging: Bootstrap Aggregation

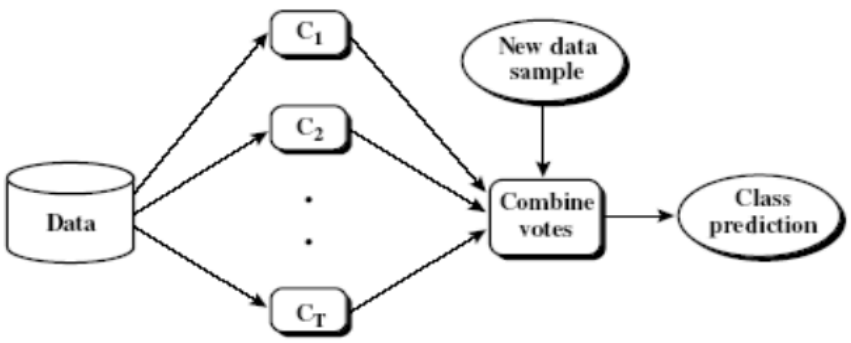


Figure: Increasing classifier accuracy: Ensemble methods generate a set of classification models, M1, M2, : : : , Mk. Given a new data tuple to classify, each classifier "votes" for the class label of that tuple. The ensemble combines the votes to return a class prediction.

Algorithm: Bagging. The bagging algorithm—create an ensemble of classification models for a learning scheme where each model gives an equally weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a classification learning scheme (decision tree algorithm, naïve Bayesian, etc.).

Output: The ensemble—a composite model, $M*$.

Method:

(1) **for** $i = 1$ to k **do** // create k models:
(2) create bootstrap sample, D_i, by sampling D with replacement;
(3) use D_i and the learning scheme to derive a model, M_i;
(4) **endfor**

To use the ensemble to classify a tuple, X:

let each of the k models classify $\boldsymbol{X}$ and return the majority vote;

Boosting:

- Boosting is an iterative ensemble method where base classifiers are trained sequentially, and each subsequent classifier focuses on the instances that were misclassified by the previous ones.
- At each iteration, the weights of misclassified instances are adjusted to emphasize the importance of those instances in subsequent iterations.
- Boosting algorithms aim to improve model performance by giving more weight to difficult-to-classify instances, effectively reducing bias and increasing overall accuracy.
- Popular boosting algorithms include AdaBoost (Adaptive Boosting), Gradient Boosting Machines (GBM), and XGBoost.

Boosting

- How boosting works?
  - Weights are assigned to each training tuple
  - A series of k classifiers is iteratively learned
  - After a classifier M_i is learned, the weights are updated to allow the subsequent classifier, M_{i+1}, to pay more attention to the training tuples that were misclassified by M_i
  - The final boosted classifier M* combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy

S.NO	Bagging	Boosting
	The simplest way of combining predictions that belong to the same type.	A way of combining predictions that belong to the different types.
	Aim to decrease variance, not bias.	Aim to decrease bias, not variance.

	Each model receives equal weight.	Models are weighted according to their performance.
	Each model is built independently.	New models are influenced by the performance of previously built models.
	Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset.	Every new subset contains the elements that were misclassified by previous models.
	Bagging tries to solve the over-fitting problem.	Boosting tries to reduce bias.

	If the classifier is unstable (high variance), then apply bagging.	If the classifier is stable and simple (high bias) the apply boosting.
	In this base classifiers are trained parallelly.	In this base classifiers are trained sequentially.
	Example: The Random forest model uses Bagging.	Example: The AdaBoost uses Boosting techniques

Random Forest:

- Random Forest is an ensemble learning method that combines the concepts of Bagging and decision trees.
- It creates an ensemble of decision trees where each tree is trained on a random subset of the features and a random subset of the data (using bootstrap sampling).
- During tree construction, at each node, the best split is chosen from a random subset of features, leading to greater diversity among individual trees.
- The final prediction is made by averaging (for regression) or voting (for classification) over all trees in the forest.
- Random Forest is highly effective due to its ability to handle high-dimensional data, reduce overfitting, and provide robust predictions.

12. What is Simple Linear Regression and multiple linear regression? Examples .

Parameter	Linear (Simple) Regression	Multiple Regression
Definition	Models the relationship between one dependent and one independent variable.	Models the relationship between one dependent and two or more independent variables.
Equation	$Y = C_0 + C_1X + e$	$Y = C_0 + C_1X_1 + C_2X_2 + C_3X_3 + ... + C_nX_n + e$
Complexity	Simpler dealing with one relationship.	More complex due to multiple relationships.
Use Cases	Suitable when there is one clear predictor.	Suitable when multiple factors affect the outcome.
Assumptions	Linearity, Independence, Homoscedasticity, Normality	Same as linear regression, with the added concern of multicollinearity.
Visualization	Typically visualized with a 2D scatter plot and a line of best fit.	Requires 3D or multi-dimensional space, often represented using partial regression plots.
Risk of Overfitting	Lower, as it deals with only one predictor.	Higher, especially if too many predictors are used without adequate data.
Multicollinearity Concern	Not applicable, as there's only one predictor.	A primary concern; having correlated predictors can affect the model's accuracy and interpretation.
Applications	Basic research, simple predictions, understanding a singular relationship.	Complex research, multifactorial predictions, studying interrelated systems.

13. Supervised and unsupervised learning

	Supervised Learning	Unsupervised Learning
Input Data	Uses Known and Labeled Data as input	Uses Unknown Data as input

Computational Complexity	Less Computational Complexity	More Computational Complex
Real-Time	Uses off-line analysis	Uses Real-Time Analysis of Data
Number of Classes	The number of Classes is known	The number of Classes is not known
Accuracy of Results	Accurate and Reliable Results	Moderate Accurate and Reliable Results
Output data	The desired output is given.	The desired, output is not given.

Model	In supervised learning it is not possible to learn larger and more complex models than in unsupervised learning	In unsupervised learning it is possible to learn larger and more complex models than in supervised learning
Training data	In supervised learning training data is used to infer model	In unsupervised learning training data is not used.
Another name	Supervised learning is also called classification.	Unsupervised learning is also called clustering.
Test of model	We can test our model.	We can not test our model.

Example	Optical Character Recognition	Find a face in an image.

14. What is classification? Classification applications

- A process of finding a model that describes and distinguishes the data classes.
- Predicts categorical class labels (discrete or nominal)

Applications:

- Credit approval
- Target marketing
- Medical diagnosis
- Fraud detection

15. Classification model building phases

Developing the Classifier or model creation: This level is the learning stage or the learning process. The classification algorithms construct the classifier in this stage. A classifier is constructed from a training set composed of the records of databases and their corresponding class names. Each category that makes up the training set is referred to as a category or class. We may also refer to these records as samples, objects, or data points.

Applying classifier for classification: The classifier is used for classification at this level. The test data are used here to estimate the accuracy of the classification algorithm. If the consistency is deemed sufficient, the classification rules can be expanded to cover new data records. It includes:

- **Sentiment Analysis:** Sentiment analysis is highly helpful in social media monitoring. We can use it to extract social media insights. We can build sentiment analysis models to read and analyze misspelled words with advanced machine learning algorithms. The accurate trained models provide consistently accurate outcomes and result in a fraction of the time.
- **Document Classification:** We can use document classification to organize the documents into sections according to the content. Document classification refers to text classification; we can classify the words in the entire document. And with the help of machine learning classification algorithms, we can execute it automatically.
- **Image Classification:** Image classification is used for the trained categories of an image. These could be the caption of the image, a statistical value, a theme. You can tag images to train your model for relevant categories by applying supervised learning algorithms.
- **Machine Learning Classification:** It uses the statistically demonstrable algorithm rules to execute analytical tasks that would take humans hundreds of more hours to perform.

  Data Classification Process: The data classification process can be categorized into five steps:

- Create the goals of data classification, strategy, workflows, and architecture of data classification.
- Classify confidential details that we store.
- Using marks by data labelling.
- To improve protection and obedience, use effects.
- Data is complex, and a continuous method is a classification.

16. Based on the following data determine the gender of a person having height 6 ft., weight 130 lbs. and foot size 8 in. (use Naive Bayes algorithm).

person	height (feet)	weight (lbs)	foot size (inches)
male	6.00	180	10
male	6.00	180	10
male	5.50	170	8
male	6.00	170	10
female	5.00	130	8
female	5.50	150	6
female	5.00	130	6
female	6.00	150	8

17. What are the weaknesses of hierarchical clustering?

18. Compare k-means with k-medoids algorithms for clustering.

Aspect	K-Means Clustering	K-Medoids Clustering
Objective	Minimize the sum of squared distances	Minimize the sum of dissimilarities/dissimilarity sum
Type of Centroid	Mean of points in each cluster	Actual data point in each cluster
Sensitivity to Outliers	Sensitive, outliers can greatly affect centroids	Less sensitive, as medoids are actual data points
Scalability	Scales well with a large number of samples	Can be computationally expensive for large datasets
Initialization	Random initialization of centroids	Medoids are typically initialized using k-means
Robustness	Prone to converge to local minima	More robust to outliers and noise
Complexity	O(k * n * i * t) where k is the number of clusters, n is the number of samples, i is the number of iterations, and t is the number of attributes	O(k * (n-k)^2) where k is the number of clusters and n is the number of samples
Applicability	Works well with data of equal variance and density	Suitable for data with uneven density and varying variances
Distance Metric	Typically uses Euclidean distance	Can use various distance metrics like Manhattan, cosine similarity, etc.

Factors	K means	K medoids
Reference point	Mean of points in cluster	Actual object in cluster
Efficiency	More efficient	Less efficient
Outliers	Sensitive to outliers	Not sensitive to outliers
Datasets	Efficient for large datasets	Efficient for small datasets
Complexity	O(nkt), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.	$O(k(n-k)^2)$ for each iteration : very costly computation for large n and k where n is # of data objects and ,k is # of clusters

Scalability	More Scalable	Less scalable

19. Compare Agglomerative (AGNES) and Divisive (DIANA) algorithm)

S.No.	Parameters	Agglomerative Clustering	Divisive Clustering
1.	**Category**	Bottom-up approach	Top-down approach
2.	**Approach**	each data point starts in its own cluster, and the algorithm recursively merges the closest pairs of clusters until a single cluster containing all the data points is obtained.	all data points start in a single cluster, and the algorithm recursively splits the cluster into smaller sub-clusters until each data point is in its own cluster.
3.	**Complexity level**	Agglomerative clustering is generally more computationally expensive, especially for large datasets as this approach requires the calculation of all pairwise distances between data points, which can be computationally expensive.	Comparatively less expensive as divisive clustering only requires the calculation of distances between sub-clusters, which can reduce the computational burden.
4.	**Outliers**	Agglomerative clustering can handle outliers better than divisive clustering since outliers can be absorbed into larger clusters	divisive clustering may create sub-clusters around outliers, leading to suboptimal clustering results.
5.	**Interpretability**	Agglomerative clustering tends to produce more interpretable results since the dendrogram shows the merging process of the clusters, and the user can choose the number of clusters based on the desired level of granularity.	divisive clustering can be more difficult to interpret since the dendrogram shows the splitting process of the clusters, and the user must choose a stopping criterion to determine the number of clusters.
6.	**Implementation**	Scikit-learn provides multiple linkage methods for agglomerative clustering, such as “ward,” “complete,” “average,” and “single,”	divisive clustering is not currently implemented in Scikit-learn.
7.	**Example**	Here are some of the applications in which Agglomerative Clustering is used : Image segmentation, Customer segmentation, Social network analysis, Document clustering, Genetics, genomics, etc., and many more.	Here are some of the applications in which Divisive Clustering is used : Market segmentation, Anomaly detection, Biological classification, Natural language processing, etc.

20.

Suppose that the data mining task is to cluster points (with (x, y) representing location) into three clusters, where the points are: A_1 (2, 10), A_2 (2, 5), A_3 (8, 4), B_1 (5, 8), B_2 (7, 5), B_3 (6, 4), C_1 (1, 2), C_2 (4, 9).
The distance function is Euclidean distance. Suppose initially we assign A_1, B_1, and C_1 as the center of each cluster, respectively. Use the *k-means* algorithm to show only (i) The three cluster centers after the first round of execution (ii) The final three clusters.

Differentiate between simple linkage, average linkage and complete linkage algorithms.
Use complete linkage algorithm to find the clusters from the following dataset.

X	4	8	15	24	24
Y	4	4	8	4	12

21. **Consider the transaction database given below. Set minimum support count as 2 and minimum confidence threshold as 70%. Generate strong association rule**

Transaction ID	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

22. **How to compute confidence for an association rule X ◊ Y?**
23. **Find all frequent item sets using Apriori algorithm. List all the strong association rules.**

24. **The transaction details are given in the following table, what is the confidence and support of the association rule {Diapers} ⇒ {Coffee, Nuts}?Find all frequent itemsets using Apriori algorithm. List all the strong association rules.**

T_id	Items bought
10	Beer, Nuts, Diapers
20	Beer, Coffee, Diapers, Nuts
30	Beer, Diapers, Eggs
40	Beer, Nuts, Eggs, Milk
50	Nuts, Coffee, Diapers, Eggs, Milk

25.

1)Suppose we have data on a few individuals randomly surveyed. The data gives the responses towards interests to promotional offers made in the areas of Finanace, Travel, Reading, and Health. Sex is the output attribute to be predicted. Apply Naïve Bayesian classification algorithm to classify the new instance (Finance = No,Travel = Yes, Reading = Yes, Health = No).

2)Build Decision Tree from Following Dataset where Sex is target/Output attribute,

Finance	Travel	Reading	Health	Sex
Yes	No	Yes	No	Male
Yes	Yes	No	No	Male
No	Yes	Yes	Yes	Female
No	Yes	No	Yes	Male
Yes	Yes	Yes	Yes	Female
No	No	Yes	No	Female
Yes	No	No	No	Male
Yes	Yes	No	No	Male
No	No	No	Yes	Female
Yes	No	No	No	Male

26. The following table shows the midterm and final exam grades obtained for students in a database course.

Use the method of least squares to find an equation for the prediction of a student's final exam grade based on the student's midterm grade in the course.

Predict the final exam grade of a student who received 86 marks on the midterm exam with the abo

x(Mid-term Exam)	Y(Final Exam)
72	84
50	63
81	77
74	78
94	90
86	75
59	49
83	79
65	77
33	52
88	74
81	90